

Eastern
Economy
Edition

DATA COMMUNICATIONS AND COMPUTER NETWORKS



PRAKASH C. GUPTA

**DATA COMMUNICATIONS
AND
COMPUTER NETWORKS**

Second Edition

PRAKASH C. GUPTA

Formerly Head
Department of Information Technology
Maharashtra Institute of Technology
Pune

PHI Learning Private Limited

Delhi-110092

2014

DATA COMMUNICATIONS AND COMPUTER NETWORKS, Second Edition Prakash C. Gupta © 2014 by PHI Learning Private Limited, Delhi. All rights reserved. No part of this book may be reproduced in any form, by mimeograph or any other means, without permission in writing from the publisher.

ISBN-978-81-203-4864-6

The export rights of this book are vested solely with the publisher.

Eighth Printing (Second Edition) **January,**
2014

Published by Asoke K. Ghosh, PHI Learning Private Limited, Rimjhim House, 111, Patparganj Industrial Estate, Delhi-110092 and Printed by Rajkamal Electric Press, Plot No. 2, Phase IV, HSIDC, Kundli-131028, Sonapat, Haryana.

Contents [Preface](#) [xxv](#) [1 Data Communication](#)
[Concepts and Terminology](#) 1–34

1.1 Basic Model of Data Communication System	1
1.2 Data Representation	2
1.2.1 ASCII—American Standard Code for Information Interchange	3
1.2.2 Byte	5
1.3 Data Transmission	5
1.3.1 Parallel Transmission	5
1.3.2 Serial Transmission	6
1.3.3 Bit Rate	7
1.3.4 Receiving Data Bits	7
1.4 Modes of Data Transmission	8
1.4.1 Asynchronous Transmission	9
1.4.2 Synchronous Transmission	10
1.5 Digital Signal Encoding	11
1.6 Unipolar and Polar Line Codes	12
1.6.1 Non Return to Zero (NRZ) Codes	12
1.6.2 Return to Zero (RZ) Codes	13
1.7 Bipolar Line Codes	14
1.7.1 Alternate Mark Inversion (AMI)	14
1.7.2 High Density Bipolar-3 Zeroes (HDB3)	14
1.7.3 Bipolar with 8 Zeroes Substitution (B8ZS) Code	15
1.8 Block Codes	15
1.9 Frequency Spectrum	16
1.9.1 Fourier Series	17
1.10 Transmission Channel	19
1.10.1 Bauds	20
1.10.2 Baseband Transmission	21
1.10.3 Modem	21
1.11 Data Compression	21
1.11.1 Information	22
1.11.2 Entropy	23
1.11.3 Redundancy	23
1.11.4 Encoding for Compression	25
1.11.5 Shannon-Fano Code	26
1.11.6 Huffman Code	26
1.12 Data Communication	29
1.12.1 Synchronous Communication	29
1.12.2 Asynchronous Communication	29
1.13 Directional Capabilities of Data Exchange	30

1.14 Line Configurations	30
Summary	31
Exercises	32

2 Transmission Media 35–64

2.1 Transmission Line Characteristics	35
2.1.1 Primary Parameters	35
2.1.2 Secondary Parameters	36
2.1.3 Phase Velocity and Phase Delay	36
2.1.4 Frequency Dependence of Secondary Parameters	37
2.2 Linear Distortions	38
2.2.1 Group Delay	39
2.2.2 Frequency Domain Equalizers	39
2.3 Characteristics of Transmission Line in Time Domain	40
2.4 Crosstalk	41
2.5 Logarithmic Units of Power Level Measurements	42
2.6 Metallic Transmission Media	43
2.6.1 Balanced Pair	43
2.6.2 Balanced Pair Cables	43
2.6.3 Loading of Balanced Pairs	44
2.6.4 Balanced Pair for Data Networks	45
2.6.5 Coaxial Cable	46
2.7 Optical Fibre	47
2.7.1 Multimode Fibres	47
2.7.2 Modal Dispersion	48
2.7.3 Monomode Fibre	49
2.7.4 Graded Index Fibres	49
2.7.5 Chromatic Dispersion	50
2.7.6 Total Dispersion	50
2.7.7 Fibre Attenuation	50
2.7.8 Advantages of Optical Fibre	51
2.8 Radio Media	51
2.8.1 Electromagnetic Spectrum	51
2.9 Baseband Transmission of Data Signals	52
2.9.1 First Nyquist Criterion	52
2.9.2 Second Nyquist Criterion	54
2.9.3 Channel Characteristic for Finite Duration Pulses	56
2.10 Equalization	56
2.10.1 Transversal Filter Equalizer	57
2.10.2 Adaptive Equalization	59
2.11 Clocked Regenerative Receiver	60
2.12 Eye Pattern	61
Summary	63

Exercises 63

3 Telephone Network 65–101

3.1 Telephone Network	65
3.1.1 Network Topology	65
3.1.2 Single Exchange Area	66
3.1.3 Multiple Exchanges	67
3.1.4 Trunk Automatic Exchanges	68
3.1.5 International Transit Exchange	69
3.2 Transport Network	69
3.2.1 Hybrids	70
3.2.2 Multiplexing	71
3.2.3 Pulse Code Modulation (PCM)	74
3.2.4 30-Channel PCM Signal	75
3.2.5 Plesiochronous Digital Hierarchy (PDH)	76
3.3 Synchronous Digital Hierarchy (SDH)	77
3.3.1 STM-1 Frame Structure	77
3.3.2 Virtual Container (VC)	79
3.3.3 VC-4	79
3.3.4 VC-12	80
3.3.5 Mapping of Data Signals on STM-1	81
3.3.6 Higher Order SDH Signals	82
3.4 Transmission Systems for Long Distance Network	83
3.5 Echo in Transmission Systems	84
3.6 Noise in Transmission Systems	85
3.6.1 Intermodulation Noise	86
3.6.2 Thermal and Shot Noise	87
3.6.3 Psophometric Weighting	88
3.6.4 Signal to Noise Ratio	88
3.6.5 Companders	89
3.7 Signal Impairments in the Telephone Network	91
3.7.1 Impulse Noise	91
3.7.2 Gain Hits and Dropouts	91
3.7.3 Phase Hits	92
3.7.4 Phase Jitter	92
3.7.5 Single Frequency Interference	92
3.7.6 Frequency Shift	92
3.8 Integrated Services Digital Network (ISDN)	93
3.8.1 ISDN Interface	93
3.8.2 ISDN Devices	94
3.8.3 Reference Interfaces	95
3.8.4 BRA Frame Structure	96
3.8.5 Access Mechanism for D Channel in BRA	97
3.8.6 Data Transmission Mechanisms of ISDN	97
3.9 Data Communications on Telephone Network	97
3.9.1 300–3400 Hz Voice Channel Bandwidth	98

3.9.2 ISDN Services	98
3.9.3 Digital Point-to-Point Links	98
3.9.4 ITU-T Recommendations for Voice Band Leased Circuits	99
Summary	100
Exercises	101

4 Data Line Devices 102–152

4.1 Digital Modulation Methods	102
4.1.1 Amplitude Shift Keying (ASK)	103
4.1.2 Frequency Shift Keying (FSK)	104
4.1.3 Phase Shift Keying (PSK)	105
4.2 Multilevel Modulation	106
4.2.1 Gray Code	106
4.2.2 4 PSK Modulator	107
4.2.3 4 PSK Demodulator	108
4.3 Differential PSK	109
4.3.1 Differential BPSK	109
4.3.2 Differential 4 PSK	111
4.3.3 16 Quadrature Amplitude Modulation (QAM)	112
4.4 Modem	114
4.4.1 Types of Modems	115
4.4.2 Scrambler and Descrambler	117
4.4.3 Block Schematic of a Modem	119
4.4.4 Additional Modem Features	122
4.5 Standard Modems	125
4.5.1 ITU-T V.21 Modem	125
4.5.2 ITU-T V.22 Modem	126
4.5.3 ITU-T V.22bis Modem	126
4.5.4 ITU-T V.23 Modem	128
4.5.5 ITU-T V.26 Modem	128
4.5.6 ITU-T V.26bis Modem	128
4.5.7 ITU-T V.26ter Modem	129
4.5.8 ITU-T V.27 Modem	129
4.5.9 ITU-T V.27bis Modem	130
4.5.10 ITU-T V.27ter Modem	130
4.5.11 ITU-T V.29 Modem	130
4.5.12 ITU-T V.32 Modem	132
4.5.13 ITU-T V.33 Modem	134
4.5.14 ITU-T V.34 Modem	135
4.5.15 ITU-T V.90 Modem	135
4.6 Other Modems and Line Drivers	135
4.6.1 Limited Distance Modems	135
4.6.2 Baseband Modems	136
4.6.3 Line Drivers	136
4.6.4 Group Band Modems	136
4.7 Digital Subscriber Line (DSL)	136

4.7.1 Asymmetric Digital Subscriber Line (ADSL)	137
4.7.2 Modulation	138
4.7.3 Distance Limitations	138
4.7.4 ITU-T Recommendations for ADSL	139
4.7.5 xDSL Technologies	139
4.8 Data Multiplexers	140
4.8.1 Types of Data Multiplexers	141
4.8.2 Frequency Division Multiplexers (FDMs)	142
4.8.3 Time Division Multiplexers (TDMs)	143
4.9 Statistical Time Division Multiplexers	144
4.9.1 Buffer	144
4.9.2 Protocol	145
4.9.3 Bit Map Statistical Multiplexing	145
4.9.4 Multiple-Character Statistical Multiplexing	145
4.9.5 Multiplexed Data Frame for Error Control Capability	146
4.9.6 Line Utilization Efficiency	147
4.9.7 Comparison of Data Multiplexing Techniques	150
Summary	150
Exercises	151

5 Error Control 153–185

5.1 Transmission Errors	153
5.1.1 Content Errors	153
5.1.2 Flow Integrity Errors	154
5.1.3 Methods of Error Control	155
5.2 Coding for Detection and Correction of Content Errors	156
5.2.1 Error Detection	156
5.2.2 Error Correction	157
5.2.3 Perfect Error Correcting Code	158
5.2.4 Systematic Code	159
5.2.5 Bit Error Rate (BER)	159
5.3 Error Detection Methods	159
5.3.1 Parity Checking	160
5.3.2 Checksum Error Detection	161
5.3.3 Cyclic Redundancy Check (CRC)	165
5.4 Forward Error Correction Methods	172
5.4.1 Block Parity	172
5.4.2 Hamming Code	174
5.4.3 Interleaved Codes	176
5.4.4 Convolutional Codes	177
5.5 Reverse Error Correction	181
5.5.1 Stop and Wait	181
5.5.2 Go-Back- <i>N</i>	182
5.5.3 Selective Retransmission	182
Summary	183

Exercises 183

6 Network Architecture 186–212

6.1 Topology of a Computer Network	186
6.2 Elements of Meaningful Communication	187
6.3 Transport-Oriented Functions	189
6.3.1 Interaction with the Subnetwork	189
6.3.2 Quality of Transport Service	189
6.3.3 Conversion of Signals	189
6.3.4 Error Control	189
6.4 Components of a Computer Network	190
6.5 Architecture of a Computer Network	190
6.5.1 Network Architecture Models	190
6.5.2 Partitioning of a System	191
6.5.3 Features of a Partitioned Structure	191
6.6 Layered Architecture of a Computer Network	192
6.6.1 Need for Standardization of Network Architecture	193
6.7 Open System Interconnection	193
6.8 Layered Architecture of the OSI Reference Model	194
6.8.1 Application Layer	195
6.8.2 Presentation Layer	195
6.8.3 Session Layer	196
6.8.4 Transport Layer	196
6.8.5 Network Layer	196
6.8.6 Data Link Layer	197
6.8.7 Physical Layer	197
6.9 Functionality of the Layered Architecture	198
6.9.1 Hierarchical Communication	198
6.9.2 Peer-to-Peer Communication	199
6.10 OSI Terminology	200
6.11 Service Interface	201
6.11.1 Service Interface Primitives and Parameters	202
6.11.2 Types of Services	203
6.12 Data Transfer Modes	204
6.12.1 Connection-Oriented Mode of Data Transfer	204
6.12.2 Connectionless Mode of Data Transfer	205
6.13 Supplementary Functions	206
6.13.1 Multiplexing of Connections	206
6.13.2 Segmenting, Blocking, and Concatenation of Data Units	207
6.14 Other Layered Architectures	208
6.14.1 TCP/IP	209
6.14.2 Systems Network Architecture (SNA)	209
6.14.3 Digital Network Architecture (DNA)	209
6.15 Standards Making Organizations	209

Summary	211
Exercises	211

7 The Physical Layer **213–241**

7.1 The Physical Layer	213
7.1.1 Physical Connection	214
7.1.2 Service Provided to the Data Link Layer	214
7.2 Functions within the Physical Layer	215
7.3 Relaying Function in the Physical Layer	216
7.4 Physical Interface	217
7.5 Physical Layer Standards	218
7.6 EIA-232-D Digital Interface	219
7.6.1 DTE/DCE Interface	219
7.6.2 DTE and DCE Ports	220
7.6.3 DCE-DCE Connection	220
7.7 EIA-232-D Interface Specifications	221
7.7.1 Mechanical Specifications	221
7.7.2 Electrical Specifications	222
7.7.3 Functional Specifications	222
7.7.4 Procedural Specifications	226
7.8 Common Configurations of EIA-232-D Interface	229
7.8.1 Three-Wire Interconnection	230
7.8.2 Three-Wire Interconnection with Loopback	230
7.9 Null Modem	231
7.9.1 Null Modem with Loopback and Multiple Crossovers	232
7.10 Limitations of EIA-232-D	233
7.11 EIA-449 Interface	233
7.11.1 Mechanical Specifications	233
7.11.2 Electrical Specifications	235
7.11.3 Functional Specifications	236
7.12 EIA-530	237
7.13 ITU-T X.21 Recommendation	237
7.13.1 Mechanical Specifications	237
7.13.2 Electrical Specifications	237
7.13.3 Functional Specifications	238
7.13.4 Procedural Specifications	239
7.13.5 X.21bis Recommendation	239
Summary	239
Exercises	240

8 The Data Link Layer **242–280**

8.1 Need for Data Link Control	242
--------------------------------	-----

8.2 Data Link Layer	243
8.2.1 Service Provided by the Data Link Layer	244
8.2.2 Data Link Protocols	245
8.3 Frame Design Considerations	246
8.3.1 Types of Frame Formats	246
8.3.2 Transparency	248
8.3.3 Bit-Oriented and Byte-Oriented Data Link Protocols	248
8.4 Flow Control Mechanisms	248
8.4.1 Stop-and-Wait Flow Control	249
8.4.2 Sliding Window Flow Control	251
8.5 Data Link Error Control	257
8.6 Error Control in Stop-and-Wait Mechanism	258
8.6.1 Stop-and-Wait Using Timeout	258
8.6.2 Stop-and-Wait Using Negative Acknowledgement (NAK)	259
8.6.3 Error Control Using Numbered Frames	260
8.6.4 Link Utilization in Presence of Errors	261
8.7 Error Control in Sliding Window Mechanism	263
8.7.1 Error Control Using Selective Retransmission	264
8.7.2 Error Control Using Go-Back-N	265
8.7.3 Link Utilization in Presence of Errors in Sliding Window Flow Control	267
8.8 Sequence Numbering of the Frames in Sliding Window Flow Control	272
8.9 Piggybacking Acknowledgements	273
8.10 Data Link Management	274
8.11 Application Environment of Data Link Protocols	275
Appendix	277
Data Link Service Primitives	277
Summary	278
Exercises	278

9 Data Link Protocols 281–321

9.1 Binary Synchronous Communication Data Link Protocol (BISYNC)	281
9.1.1 Communication Modes	282
9.2 Transmission Frame	282
9.2.1 Frame Format	282
9.2.2 Control Characters	284
9.2.3 Error and Flow Control	285
9.2.4 Transparency	286
9.3 Protocol Operation	287
9.3.1 Point-to-Point Communication	287
9.3.2 Point-to-Multipoint Communication	288

9.3.3	Limitations of BISYNC Protocol	290	
9.4	High Level Data Link Control (HDLC)	290	290
9.4.1	Types of Stations	291	
9.4.2	Modes of Operation	292	
9.5	Flow and Error Control in HDLC	293	293
9.6	Framing in HDLC	294	294
9.6.1	Frame Formats	294	
9.6.2	Control Field of HDLC Frames	296	
9.6.3	Poll/Final (P/F) Bit	299	
9.7	Transparency in HDLC	300	300
9.8	HDLC Protocol Operation	301	301
9.8.1	Normal Response Mode, Point-to-Point	302	
9.8.2	Normal Response Mode, Point-to-Multipoint	305	305
9.8.3	Asynchronous Response Mode (ARM)	305	
9.8.4	Asynchronous Balanced Mode (ABM)	308	
9.9	Additional Features	308	308
9.9.1	Extended Addressing	308	
9.9.2	Extended Control Field	309	
9.10	Comparison of BISYNC and HDLC Features	309	309
9.11	Link Access Procedure-Balanced (LAP-B)	310	310
9.12	Multilink Procedure (MLP)	311	311
9.13	Link Access Procedures for Modems (LAP-M)	312	312
9.14	Link Access ProcedureD (LAP-D)	313	313
9.14.1	Frame Format	313	
9.14.2	Procedures	315	
	Summary	317	317
	Exercises	317	317

10 Local Area Networks 322–340

10.1	Need for Local Area Networks	323	323
10.1.1	LAN Attributes	323	
10.1.2	LAN Environment in an Organization	323	323
10.2	Lan Topologies	324	324
10.2.1	Bus Topology	324	
10.2.2	Ring Topology	326	
10.2.3	Star Topology	327	
10.2.4	Logical Topology	328	
10.3	Media Access Control	328	328
10.4	Layered Architecture of LAN	329	329
10.5	IEEE Standards	330	330
10.6	Logical Link Control (LLC) Sublayer	331	331
10.6.1	LLC Service	331	
10.6.2	LLC Protocol	333	

10.6.3 LLC Procedures	335	
10.7 Media Access Control (MAC) Sublayer		336
10.7.1 MAC Service	336	
10.7.2 MAC Protocol	336	
10.8 Transmission Media for Local Area Networks		337
10.8.1 Twisted Copper Pair Cable	338	
10.8.2 Coaxial Cables	338	
10.8.3 Optical Fibre Cable	338	
Summary	339	
Exercises	339	

11 IEEE 802.3 Ethernets 341–379

11.1 Contention Access		341
11.1.1 Pure ALOHA	341	
11.1.2 Throughput of Pure ALOHA Channel		342
11.1.3 Slotted ALOHA	344	
11.2 Carrier Sense Multiple Access (CSMA)		345
11.2.1 Non-Persistent CSMA	346	
11.2.2 1-Persistent CSMA	346	
11.2.3 <i>p</i> -Persistent CSMA	346	
11.3 CSMA/CD		347
11.3.1 Media Access Control in CSMA/CD		347
11.3.2 Maximum Cable Segment	349	
11.3.3 MAC Frame Format (IEEE 802.3)		350
11.3.4 Format of Ethernet (DIX) Frame		351
11.3.5 Truncated Binary Exponential Back Off		353
11.4 Physical Topology of Ethernet LAN		353
11.4.1 Bus Topology	353	
11.4.2 Point-to-Point Topology	354	
11.4.3 Star Topology	354	
11.5 Ethernet Repeater		355
11.5.1 Collisions in a Repeater	355	
11.5.2 Link Segments	355	
11.5.3 Ethernet Hubs	356	
11.6 Types of Ethernets		356
11.6.1 Physical Layer of Ethernet LANs		357
11.7 10 Mbps Ethernets		358
11.7.1 10Base5 (Thick Ethernet)	358	
11.7.2 10Base2 (Thin Ethernet)	359	
11.7.3 10Broad36	359	
11.7.4 10BaseT	359	
11.7.5 10BaseF	360	
11.8 Fast Ethernet		360
11.8.1 Additional Functions Required for 100 Mbps LANs		361
11.8.2 Physical Layer of Fast Ethernets		361

11.8.3 100BaseT4	362
11.8.4 100BaseT2	366
11.8.5 100BaseX	367
11.9 Flow Control	369
11.10 Auto-Negotiation	370
11.10.1 Transport Mechanism for Auto-Negotiation	370
11.10.2 FLP Burst Encoding	371
11.10.3 Ability Negotiation Mechanism	372
11.10.4 Parallel Detection	373
11.11 Gigabit Ethernet	373
11.11.1 Gigabit Carrier Extension	374
11.11.2 Frame Bursting	375
11.11.3 1000BaseT	375
11.11.4 1000BaseX	376
11.11.5 Auto-Negotiation in Gigabit Ethernets	377
Summary	377
Exercises	378

12 Token Passing Local Area Networks 380–411

12.1 Token Ring Local Area Network	380
12.2 Media Access Control in Token Ring LAN	381
12.2.1 Token Holding Time	383
12.2.2 Early Token Release	383
12.3 Ring Size	383
12.3.1 Bypass Relay	383
12.3.2 Multi-Station Attachment Unit	384
12.4 Standards for Token Ring LAN	385
12.4.1 IEEE 802.5 MAC Frame Format	385
12.5 MAC Addresses (DA/SA) in Token Ring LAN	388
12.5.1 Functional Address	388
12.6 Priority Management in Token Ring LAN	389
12.6.1 Stacking Station	390
12.7 Ring Management in Token Ring LAN	391
12.7.1 Active Monitor Station Selection	392
12.7.2 Upstream Neighbour Determination	392
12.7.3 Token Management	393
12.7.4 Initialization Process for a New Station	393
12.7.5 Persistent Circulating Frames	393
12.7.6 Master Clock Generation	394
12.7.7 Beacons	394
12.8 Token Bus LAN	395
12.8.1 Media Access Control in Token Bus LAN	395
12.8.2 Frame Structure of Token Bus LAN	396
12.8.3 Response Window	397
12.8.4 Token Bus Management	397

12.8.5 Priority Operation in Token Bus	399
12.8.6 Physical Specifications	401
12.9 Fibre Distributed Data Interface (FDDI)	402
12.9.1 Physical Topology	402
12.9.2 Types of FDDI Stations	403
12.9.3 Types of Services	404
12.10 Media Access Control in FDDI	405
12.10.1 Frame Fragmentation	405
12.10.2 Fragment Removal	406
12.10.3 Priority Management in FDDI	406
12.11 MAC Frame Format in FDDI	407
12.12 Physical Specifications of FDDI	409
12.12.1 Ring Size and Number of Stations	409
Summary	409
Exercises	410

13 Wireless Local Area Networks 412–439

13.1 Wireless Local Area Network	412
13.1.1 Wireless Local Area Network Configuration	413
13.1.2 Communication Modes	414
13.2 Layered Architecture of Wireless Local Area Network	415
13.2.1 Functions of MAC Sublayer in IEEE 802.11	416
13.3 Media Access Control in Wireless Local Area Network	417
13.3.1 Inter-Frame Spaces in IEEE 802.11	419
13.4 Distributed Coordination Function (DCF)	419
13.4.1 DCF with RTS/CTS	421
13.4.2 Binary Exponential Back-off	423
13.4.3 Fragmentation	424
13.5 Point Coordination Function (PCF)	424
13.5.1 Communication during Contention Free Period (CFP) Using PCF	425
13.6 MAC Frames of the IEEE 802.11	426
13.6.1 Control Frames	426
13.6.2 Data Frames	427
13.6.3 Format of MAC Frames of IEEE 802.11	428
13.7 Transmission Technologies of IEEE 802.11	430
13.7.1 Spread Spectrum Transmission Systems	430
13.7.2 Infrared Transmission Systems	433
13.7.3 Orthogonal Frequency Division Multiplexing (OFDM)	435
13.8 Physical Layer of IEEE 802.11	435
13.8.1 Original IEEE 802.11 Physical Layer	436
13.8.2 IEEE 802.11a	437
13.8.3 IEEE 802.11b	437
Summary	437

Exercises 438

14 Bridges and Layer-2 Switches 440–468

14.1 Motivation for Using Lan Bridges	440
14.2 LAN Bridge	441
14.2.1 Bridge Architecture	441
14.2.2 Types of Bridges	442
14.3 Transparent Bridges	443
14.3.1 Frame Filtering and Forwarding	443
14.3.2 Learning Addresses	444
14.3.3 Multiple Paths	446
14.4 Spanning Tree Algorithm	447
14.4.1 Bridge Protocol Data Unit (BPDU)	448
14.4.2 Constructing the Spanning Tree	449
14.4.3 Error Situations and Limitations of Transparent Bridge	452
14.5 Source Routing Bridges	453
14.5.1 Frame Structure	453
14.5.2 Routing Directives	455
14.6 Route Discovery in Source Routing	455
14.6.1 Route Discovery Using All-Route-Broadcast (ARB)	455
14.6.2 Route Discovery Using Single-Route-Broadcast (SRB)	458
14.7 Source Routing Bridge versus Transparent Bridge	462
14.8 Remote Bridges	462
14.9 Layer 2 Ethernet Switches	463
14.9.1 Motivation behind Ethernet Layer-2 Switches	463
14.9.2 Latency in Ethernet Layer-2 Switch	465
14.9.3 Basic Features of Ethernet Layer-2 Switch	465
Summary	466
Exercises	466

15 Network Layer 469–495

15.1 Wide Area Networks	469
15.1.1 Switched Data Networks	470
15.1.2 Types of Switched Data Networks	471
15.2 Circuit Switching	471
15.2.1 Operational Phases in Circuit Switching	471
15.2.2 Delays in Circuit Switched Data Network	472
15.3 Store-and-Forward Data Networks	473
15.3.1 Message Switching	474
15.3.2 Packet Switching	475
15.4 Types of Packet Switched Data Networks	477
15.4.1 Datagram Switching Network	477
15.4.2 Virtual Circuit Packet Switching	481
15.5 Purpose of the Network Layer	485

15.5.1 The End System to Access Node Link	485	
15.5.2 Node-to-Node Link	486	
15.5.3 End System-to-End System Layered Network Architecture		486
15.6 Network Service	487	
15.6.1 Connection-Oriented Network Service (CONS)	488	
15.6.2 Connectionless-Mode Network Service (CLNS)	488	
15.6.3 Basic Features of Network Service	488	
15.7 Functions of the Network Layer	488	
15.8 Internetworking	489	
Summary	491	
Exercises	491	

16 Virtual Circuit Packet Switching Network 496–544

16.1 X.25 Interface	496	
16.1.1 Scope of X.25 Interface	497	
16.2 X.25 Services	498	
16.3 Logical Channels	498	
16.3.1 Grouping of Logical Channels	499	
16.4 General Packet Format	499	
16.5 Procedures for Switched Virtual Circuits		501
16.5.1 Call Establishment Phase	502	
16.5.2 Data Transfer Phase	503	
16.5.3 Call Clearing Phase	508	
16.5.4 Call Collision	509	
16.5.5 Virtual Circuit Reset	510	
16.5.6 Restart	511	
16.5.7 Error Recovery by Timers	511	
16.5.8 Procedures for Permanent Virtual Circuit (PVC)		512
16.6 User Facilities in X.25	512	
16.6.1 Fast Select	512	
16.6.2 Reverse Charging and Reverse Charge Acceptance		513
16.6.3 Closed User Groups	513	
16.6.4 Flow Control Parameter Negotiation	514	
16.7 Addressing in X.25	515	
16.8 Packet Assembler and Disassembler (PAD)		516
16.8.1 PAD Operation	517	
16.9 Frame Relay	518	
16.9.1 Frame Relay Network Topology	518	
16.9.2 Frame Relay Connection	519	
16.9.3 Frame Relay Services	520	
16.9.4 Layered Architecture of Frame Relay Network		520
16.10 Congestion Control in Frame Relay		521
16.10.1 Parameters for Congestion Control	521	
16.10.2 Congestion Control Using FECN, BECN, and DE Bits		523

16.11	Frame Format in Frame Relay	524
16.11.1	Reserved Data Link Connection Identifiers	525
16.11.2	Basic Operation of LAP-F	525
16.11.3	IP Encapsulation	526
16.12	Asynchronous Transfer Mode (ATM)	526
16.12.1	UNI and NNI	528
16.12.2	ATM Virtual Channel Connection (VCC)	528
16.12.3	Virtual Path Connection (VPC)	529
16.13	Layered Architecture in ATM	530
16.13.1	Physical Layer	530
16.13.2	ATM Layer	532
16.14	ATM Adaptation Layer (AAL)	534
16.14.1	Traffic Classification	535
16.14.2	Structure of the Adaptation Layer	536
16.14.3	ATM Adaptation Layer 1 (AAL 1)	537
16.14.4	ATM Adaptation Layer 2 (AAL 2)	538
16.14.5	ATM Adaptation Layer 3/4 (AAL 3/4)	538
16.14.6	ATM Adaptation Layer 5 (AAL 5)	540
	Summary	541
	Exercises	542

17 Internet Protocol (IP) 545–593

17.1	Connectionless-Mode Switched Data Networks	545
17.2	Internet Protocol (IP)	546
17.2.1	Associated Protocols	547
17.3	IPv4 Packet Format	547
17.3.1	Fragmentation	550
17.3.2	Options	551
17.4	Hierarchical Addressing	552
17.4.1	Addressing Scheme of IPv4	553
17.4.2	Classful IP Addressing	554
17.4.3	Number of Networks and Hosts	556
17.4.4	Special Addresses	556
17.5	Subnetting	557
17.5.1	Classical Subnetting	558
17.5.2	Route Advertisement	560
17.5.3	Variable Length Subnet Mask (VLSM)	562
17.5.4	Classless Addressing and Supernetting	563
17.6	Address Resolution Protocol (ARP)	564
17.6.1	Layered Architecture for ARP	565
17.6.2	ARP Operation	565
17.6.3	Format of ARP Packet	567
17.6.4	Complete Picture of IP Packet Delivery	569
17.7	Internet Control Message Protocol (ICMP)	570
17.7.1	ICMP Message	571

17.8 IPv6 Internet Protocol	574
17.8.1 Format of IPv6 Packet	574
17.8.2 Extension Headers	575
17.8.3 Address Notation in IPv6	577
17.8.4 Comparison of IPv6 and IPv4 Headers	577
17.9 ISO Connectionless-Mode Network Protocol (CLNP)	577
17.9.1 Types of IPDU	578
17.10 Format of ISO 8473 IPDU	578
17.10.1 Fixed Part	579
17.10.2 Address Part	580
17.10.3 Segmentation Part	580
17.10.4 Options Part	581
17.11 Point-to-Point Protocol (PPP)	582
17.11.1 Layered Architecture of PPP	582
17.11.2 Physical Layer for PPP	582
17.11.3 PPP Frame Format	583
17.11.4 PPP Operation	584
17.12 Link Control Protocol (LCP)	585
17.12.1 LCP Packets	586
17.13 Authentication Protocols	587
17.13.1 PAP	587
17.13.2 CHAP	587
17.14 Network Control Protocol (NCP)	588
Summary	588
Exercises	589

18 Routing Protocols 592–639

18.1 Routing	592
18.1.1 Administrative and Routing Domains	593
18.2 Static Routing	594
18.2.1 Default Routes	595
18.3 Dynamic Routing	596
18.4 Distance Vector Routing Algorithm	596
18.4.1 Slow Convergence to Steady State	599
18.4.2 Count-to-Infinity	599
18.4.3 Split Horizon	600
18.4.4 Hold-Down Timer	601
18.4.5 Path Vector	601
18.5 Routing Information Protocol (RIP)	602
18.5.1 Format of RIPv1 Packet	602
18.5.2 Types of RIP Packets	604
18.5.3 Forwarding Table	604
18.5.4 Limitations of RIPv1	605
18.5.5 Routing Information Protocol (Version 2)	606

18.5.6	Format of RIPv2 Packet	606	
18.5.7	Next-Hop Field in RIPv2	607	
18.5.8	Authentication	608	
18.6	Link State Routing	608	
18.6.1	Basic Operation	608	
18.6.2	Dijkstra's Algorithm	609	
18.7	Open Shortest Path First (OSPF) Routing Protocol		613
18.7.1	Hierarchical Routing	614	
18.7.2	OSPF Packets	615	
18.8	Formation of Adjacencies in OSPF		617
18.8.1	Hello Protocol	617	
18.8.2	Database Synchronization	620	
18.8.3	Format of Database Description Packet	621	
18.9	Link State Updates in OSPF		622
18.9.1	Controlled Flooding	622	
18.9.2	Types of LSAs	624	
18.9.3	Format of LSA	626	
18.9.4	Advantages of Link State Routing over Distance Vector Routing		627
18.10	OSI Routing Protocols		627
18.10.1	OSI Routing Terminology	628	
18.11	Interior and Exterior Gateway Protocols		629
18.11.1	Interior Gateway Protocols	629	
18.11.2	Exterior Gateway Protocols	629	
18.12	Border Gateway Protocol (BGP)		630
18.12.1	Basic Operation	631	
18.12.2	BGP Messages	631	
18.12.3	Path Attributes	632	
18.12.4	Choosing the Active Route	635	
Summary		636	
Exercises		636	

19 Multicasting and Multiprotocol Label Switching (MPLS) 640–666

19.1	Multicasting	640	
19.1.1	Multicast Group	641	
19.2	Multicast Routing Principles		642
19.2.1	Multicast Using Controlled Flooding	642	
19.2.2	Multicast Using Spanning Tree	643	
19.3	Reverse Path Forwarding (RPF)		644
19.3.1	Improved RPF	645	
19.3.2	Pruning	647	
19.3.3	Grafting	648	
19.4	Core-Based Trees (CBT)		648
19.5	Multicasting Protocols		650

19.5.1 Protocol Independent Multicast (PIM)	650	
19.5.2 DVMRP and MOSPF	652	
19.6 Addressing in IP Multicast	653	
19.6.1 Multicast on LAN Segments	654	
19.7 Internet Group Management Protocol (IGMP)		655
19.7.1 IGMP Version 1	655	
19.7.2 IGMP Version 2	656	
19.7.3 IGMP Version 3	657	
19.8 Multiprotocol Label Switching		657
19.8.1 Basic Approach of MPLS	658	
19.8.2 MPLS Header	660	
19.8.3 Forwarding Equivalence Class (FEC)	661	
19.8.4 Label Distribution Protocol (LDP)	661	
19.8.5 Other Methods of Creating LSPs	662	
19.8.6 MPLS for Traffic Engineering	662	
19.8.7 MPLS Tunnels	663	
19.8.8 Label Stacking	664	
Summary	664	
Exercises	665	

20 Transport Layer 667–694

20.1 Transport Layer	667	
20.1.1 Purpose of Transport Layer	668	
20.1.2 Types of Transport Service	669	
20.1.3 Functions within Transport Layer	669	
20.2 Transmission Control Protocol (TCP)		671
20.3 TCP Ports and Connections	671	
20.4 Format of TCP Segment	673	
20.4.1 Maximum Segment Size (MSS)	675	
20.4.2 Pseudo IP Header	675	
20.4.3 Forced Data Delivery	676	
20.4.4 Urgent Data	676	
20.5 TCP Operation	676	
20.5.1 TCP Connection Establishment Phase	676	
20.5.2 TCP Data Transfer Phase	678	
20.5.3 TCP Disconnection Phase	681	
20.5.4 TCP Connection Reset	682	
20.6 Flow Control in TCP	682	
20.6.1 Window Size	683	
20.6.2 Sliding Window Mechanism	683	
20.6.3 Silly Window Syndrome	685	
20.7 Estimation of Retransmission Timeout in TCP		685
20.7.1 Methods of Estimating RTT and RTO	686	
20.8 Congestion Avoidance in TCP	687	
20.8.1 Slow Start	687	

20.8.2 Congestion Avoidance	689
20.8.3 Fast Recovery	690
20.9 User Datagram Protocol (UDP)	691
20.9.1 Format of UDP Datagram	692
20.9.2 UDP Operation	693
Summary	693
Exercises	693

21 Network Security 695–725

21.1 Security Requirements	695
21.2 Cryptography Algorithms	696
21.3 Algorithms for Confidentiality	697
21.3.1 Secret Key Encryption Algorithms	697
21.3.2 Exchange of Secret Key	699
21.3.3 Public Key Encryption Algorithms	700
21.4 Algorithms for Integrity	702
21.4.1 MD5 Algorithm	703
21.4.2 DES Cipher Block Chaining (DES-CBC)	703
21.4.3 Keyed MD5	704
21.5 Basic Authentication Mechanisms	705
21.5.1 Authentication Using Secret Key	705
21.5.2 Authentication Using Secret Key with Third Party	707
21.5.3 Authentication Using Message Digest	708
21.5.4 Authentication Using Public Key	708
21.6 Mechanisms for Ensuring Message Integrity	709
21.7 Digital Signature	709
21.7.1 Digital Signature Using Private Key	709
21.7.2 Digital Signature Using Private Key and Message Digest	710
21.7.3 Digital Signature Using Third Party and Secret Keys	711
21.8 Management of Public Keys Through Third Parties	712
21.8.1 Digital Certificate	712
21.8.2 X.509	713
21.8.3 Certification Authority Hierarchy	713
21.8.4 Revocation of Certificates	714
21.9 Transport Layer Security	714
21.9.1 Secure Socket Layer (SSL)	714
21.9.2 SSL Architecture	715
21.9.3 SSL Record Protocol	716
21.9.4 Handshake Protocol	717
21.9.5 Change Cipher Spec Protocol	718
21.9.6 Alert Protocol	718
21.10 IP Security (IPSec)	718
21.10.1 Components of IPSec	719
21.10.2 Security Association	719
21.10.3 Authentication Header (AH)	720

21.10.4 Encapsulating Security Payload (ESP)	721
21.11 Firewalls	722
21.11.1 Filter-Based Firewalls	722
21.11.2 Proxy-Based Firewalls	723
Summary	723
Exercises	724

22 Application Layer 726–760

22.1 TCP/IP Application Protocols	726
22.1.1 Client-Server Model	727
22.2 Domain Name System (DNS)	728
22.2.1 Hierarchical Naming System	728
22.2.2 Internet Naming System	729
22.2.3 DNS Protocol	731
22.3 Bootstrapping Protocol (BOOTP)	732
22.4 Dynamic Host Configuration Protocol (DHCP)	735
22.4.1 Configurable Parameters	735
22.4.2 Message Format	735
22.4.3 Message Types	735
22.5 Trivial File Transfer Protocol (TFTP)	736
22.5.1 TFTP Message Formats	737
22.5.2 TFTP Operation	738
22.6 Telnet	738
22.6.1 Character Set of NVT	739
22.6.2 Telnet Commands	739
22.6.3 Standard User Commands	740
22.7 File Transfer Protocol (FTP)	740
22.7.1 Basic Features of FTP	741
22.7.2 FTP Operation	741
22.8 Electronic Mail	742
22.8.1 Basic Components	742
22.8.2 Mail Addresses	743
22.8.3 Mail Format	744
22.8.4 Simple Mail Transfer Protocol (SMTP)	744
22.8.5 Multipurpose Internet Mail Extension (MIME)	746
22.8.6 Post Office Protocol (POP3) and Internet Message Access Protocol (IMAP4)	748
22.9 Simple Network Management Protocol (SNMP)	748
22.9.1 Basic Elements	749
22.9.2 SNMP Architecture	750
22.9.3 Management Information Base (MIB)	751
22.9.4 Structure of Management Information (SMI)	752
22.9.5 SNMP Operation	752
22.10 World Wide Web (WWW)	754
22.10.1 Hypertext Markup Language (HTML)	754

22.10.2 Uniform Resource Locator (URL)	755	
22.11 Hypertext Transfer Protocol (HTTP)		756
22.11.1 HTTP Request Messages	756	
22.11.2 HTTP Response Messages	757	
Summary	758	
Exercises	760	

23 Quality of Service 761–809

23.1 Motivation for Quality of Service (QOS)		761
23.1.1 QOS in IP Networks	762	
23.1.2 Contractual Agreements for QOS	763	
23.2 QOS Parameters	763	
23.2.1 Bandwidth	764	
23.2.2 Delay	764	
23.2.3 Jitter	765	
23.2.4 Packet Loss	766	
23.3 Functions Required for Supporting QOS		767
23.3.1 Traffic Classification and Scheduling	767	
23.3.2 Traffic Control	767	
23.3.3 Congestion Management	767	
23.4 Queuing System	768	
23.4.1 Single Queue	769	
23.4.2 Priority Queuing	770	
23.4.3 Weighted Round-Robin (WRR) Queuing	771	
23.4.4 Fair Queuing	773	
23.4.5 Weighted Fair Queuing	776	
23.5 Traffic Control	777	
23.5.1 Traffic Characterization Parameters	777	
23.5.2 Basic Traffic Control Functions	778	
23.6 Leaky Bucket Algorithm	779	
23.6.1 Shaping Using Leaky Bucket Regulator	779	
23.6.2 Policing Using Leaky Bucket Regulator	780	
23.7 Token Bucket Algorithm	780	
23.7.1 Maximum Burst Size	781	
23.7.2 Queuing Delay of Shaped Traffic	782	
23.8 Queue Buffer Management	783	
23.8.1 Tail Drop	783	
23.8.2 Random Early Detection (RED)	784	
23.8.3 Weighted RED (WRED)	786	
23.9 Explicit Congestion Notification (ECN)		786
23.9.1 Enhancements at IP Layer for ECN	787	
23.9.2 Enhancements at TCP Layer for ECN	787	
23.9.3 ECN Operation	788	
23.9.4 Benefits and Limitations of ECN	790	
23.10 Frameworks for Implementing QOS		790

23.11 Integrated Service	791	
23.11.1 Classes of Integrated Service	791	
23.11.2 Tspec and Rspec	792	
23.11.3 Components of Integrated Service	792	
23.12 Resource Reservation Protocol (RSVP)		793
23.12.1 RSVP Messages	793	
23.12.2 Format of RSVP Messages	794	
23.12.3 Basic Operation of RSVP	796	
23.12.4 Resource Reservation for Multicasting		798
23.12.5 Reservation Styles	799	
23.12.6 Limitations of RSVP	799	
23.13 Differentiated Service	799	
23.13.1 Differentiated Service Code Point (DSCP) Field		800
23.13.2 Per-Hop-Behaviour (PHB)	800	
23.13.3 Differentiated Service (DS) Domain	803	
23.14 Differentiated Service Support in MPLS		804
23.14.1 Limitations of Differentiated Service Framework		805
Summary	805	
Exercises	806	

Bibliography **811–812**

Answers to Selected Exercises **813–826**

List of Acronyms **827–831**

Index **833–848**

Preface

Data communications and computer networks are the two perspectives of a multidimensional field that encompasses interests of computing industry, telecommunications industry, Internet service providers, and the vast base of user groups. The last decade saw enormous standardization work and technological advancement in this field. One of the offshoots of this advancement was manifold increase in the demand for the courses on data communications and computer networks. Apart from the slight difference in the focus, these courses refer to the same subject. Communications engineers view this subject as study of *network infrastructure* that supports data transport. Computer network engineers, on the other hand, follow a top-down approach that focusses on *end-to-end* secure communication between applications *using* the network infrastructure. The content of this book has been designed keeping in mind the requirements of both these user groups. The book should also appeal to the network professionals who need to keep themselves updated on the network architectural changes as demanded by the new service requirements.

The book lays emphasis on the underlying principles of communication and networking as applied to data networks. Standard protocols that define network architecture are covered in sufficient detail to equip the students for taking up networking assignments after completing the course.

Organization of the Book The text is organized into an integrated sequence of topics ensuring that there are no knowledge-gaps. Wherever required, concepts borrowed from other related disciplines are used for filling the gaps. The twenty-three chapters of the book are organized as follows: **Chapters 1–5.** Basic principles of data transmission, error control, data compression, transmission media and data line devices (modems, multiplexers).

Chapters 6–9. Introduction to layered network; architectures and protocols for physical transport of bits.

Chapters 10–14. Layer-2 wired and wireless networks, layer-2 devices (bridges and switches).

Chapters 15–19. Layer-3 networks based on virtual circuit switching and datagram approaches, routing protocols, MPLS and multicasting.

Chapters 20–22. End-to-end data transport, congestion control, network

security and network applications.

Chapter 23. Quality of Service The book contains the most recent developments of the networking technology. The following are some of the important topics included in the book

- **Access and transport:** ADSL, SDH, ISDN.
- **Data link protocols:** HDLC, PPP, LAP-B, LAP-D, LAP-F, LAP-M, PPP.
- **Local area networks:** Fast ethernets, Gigabit ethernets, auto-negotiation, token passing LANs, wireless LANs based on DSSS and FHSS.
- **Layer-2 internetworking:** Transparent and source routing bridges, layer-2 switches.
- **Wide area networks:** X.25, ATM, frame relay, congestion control.
- **Internet:** IPv4, IPv6, QOS, MPLS, LDP, ARP, RARP.
- **Multicasting:** RPF, CBT, IGMP, PIM (Dense and sparse modes), DVMRP, MOSPF.
- **Routing protocols:** RIPv2, OSPF, BGP.
- **Transport layer:** TCP, UDP, congestion control.
- **Network security:** Secret-key/public key encryption, DES, RSA, authentication, MD5, digital signature, digital certificate, IPsec, firewalls, SSL.
- **Network applications:** SNMP, SMTP, Telnet, FTP, TFTP, BOOTP, DHCP, WWW, DNS, HTML, URL, HTTP

Each chapter is structured into sections and subsections to provide flexibility in curriculum design. The text contains *numerous examples and illustrations* to bring clarity to the subject and enhance its understanding. The exercises with some twist are given at the end of each chapter. These exercises will help the teaching faculty to test the understanding of the subject by their students. Answers to selected exercises are included in the book. The book contains list of frequently used acronyms, bibliography, and index at the end.

Course Design The content of the book is designed for a two-semester course spanning over 120 hours. It can meet the requirements of the following courses:

- Fundamentals of Data Communications
- Data Networks
- Computer Networks.

Fundamentals of Data Communications. The curriculum of the first course on Data Communications focusses on fundamentals of data transmission and basic knowledge of data networks. Its content can be based on Chapters 1–10 and Chapter 15.

Course on Data Networks. This course is meant for the students who have the basic knowledge of Data Communications. This can be the second course for the students of Electronics and Communication stream. The emphasis of the proposed curriculum is on data networks. Chapter 11 to Chapter 23 provide sufficient material for a one-semester course. A quick recap of the topics covered in Chapters 1, 6, 8, 9, and 10 will make understanding of the subsequent advanced network topics.

Course on Computer Networks. The curriculum of the Computer Network course stresses on networks and applications. The prerequisite for the Computer Networks course is basic course on communications. Selected topics from the chapters indicated in brackets will provide the needed foundation for the course.

- (Chapters 1, 5, 8, 9, and 10), Chapters 6, 11, 13, 15, and 17 to 23.

This book would not have been possible without the help from many people. I would like to thank them for their support in bringing this book out. First, I thank Nutan and Priyanka who helped me in preparing the manuscript and proof reading. I would also like to thank the editorial and production teams of PHI Learning for their excellent and professional approach to the publication of the book.

Prakash C. Gupta

1

Data Communication Concepts and Terminology

Communication, whether between human beings or computer systems, involves transfer of information from a sender to a receiver. Data communication refers to exchange of information between two devices capable of generating, processing and interpreting data. In this chapter, we examine some of the basic concepts and terminology relating to data communication. Data representation, modes of data transmission, and line codes are discussed first. We then proceed to examine applications of some theoretical concepts of data transmission. These include Fourier series, Nyquist's and Shannon's theorems that relate bandwidth, data rate, and signal to noise ratio. Data compression techniques are introduced next. We close the chapter with distinction between data transmission and data communication. These terms are used interchangeably in the literature on data communication and are the cause of much confusion and frustration.

1.1 BASIC MODEL OF DATA COMMUNICATION SYSTEM

The purpose of any type of communication is to exchange information between two agents. The terms 'information' and 'exchange' connote meaningful transfer of the symbols which constitute a message. Data communication, in particular, refers to meaningful exchange of information between two entities which can be a human operator on a computer terminal and a computer program or two computer programs (Figure 1.1). The information to be transmitted is processed to ensure its reliability, integrity, and intelligibility during transfer. Some of required processing functions are representation of information in form of binary data, dialogue management, error control, addressing, and sequencing. The

transmitter converts the binary data into a time varying signal $s(t)$ having characteristics suitable for the transmission medium being used. The signal $s(t)$ presented to the transmission medium is subjected to a number of impairments before it reaches the receiver. However, it is ensured that the received signal $r(t)$ has good likeness of the transmitted signal $s(t)$. The received signal is converted back into binary data which is processed

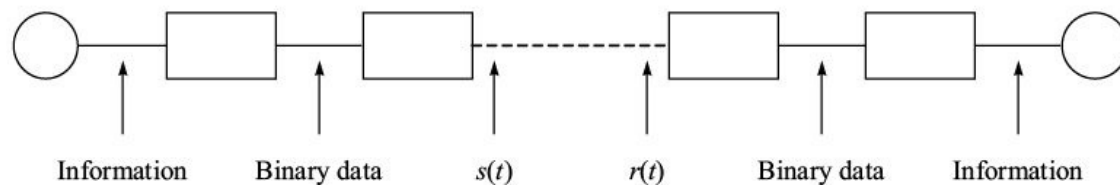


FIGURE 1.1 Simplified model of a communication system.

to get the information. The information is displayed on the terminal, stored on a hard disk or executed as a command.

The above model of data communication is a simplistic picture of a very complex system. It highlights two important issues which we address in rest of this chapter. These are:

- Representation of data in binary form.
- Transmission of data as a time varying signal.

We will examine a complete model of data communication system in Chapter 6, Network Architecture.

1.2 DATA REPRESENTATION

In data communications, we deal with exchange of information represented in binary form. A binary digit, called *bit*, has only two values 0 and 1, and can represent only two symbols. However, the simplest form of communication between two terminal devices requires much larger set of symbols. For example,

- 52 upper and lower case letters of English alphabet.
- 10 numerals from 0 to 9.
- Punctuation marks and other special symbols.
- Terminal control characters, carriage return (CR), line feed (LF).

Therefore, a group of bits is used as a code to represent a symbol. The code is usually 5 to 8 bits long. 5-bit code can have $2^5 = 32$ combinations and can, therefore, represent 32 symbols. Similarly, an 8-bit code can represent $2^8 = 256$ symbols. A code set is set of these codes representing symbols. There have been several code sets. Some code sets are application-specific and some are the proprietary code sets of computer manufactures. The following two code sets have been very common in the industry.

- American Code for Information Interchange (ASCII) of ANSI.
- Extended Binary Coded Decimal Interchange Code (EBCDIC) of IBM.

ASCII is a 7-bit code and is used worldwide. EBCDIC is an 8-bit code primarily used in large IBM computers. BCDIC and Baudot Teletype codes are other codes which are not of much significance to data processing community today, but were used one time or the other. BCDIC is 6-bit code with 64 symbols. Its code set does not include lower case letters. Baudot Teletype code, also called International Telegraph Alphabet Number 2 (ITA2), is a 5-bit code and was used in electromechanical teletype machines.

1.2.1 ASCII—American Standard Code for Information Interchange

ASCII is defined by American National Standards Institute (ANSI) in ANSI X.3.4. The corresponding ITU-T recommendation is T.50 (International Alphabet No. 5 or IA5) and ISO specification is ISO 646. It is 7-bit code and all the possible 128 codes have defined meanings (Table 1.1). The code set consists of the following symbols.

- 96 graphic symbols (columns 2 to 7), comprising 94 printable characters, SPACE, and DELeTe characters.
- 32 control symbols (columns 0 and 1).

Binary representation of a symbol is readily determined from its hexadecimal coordinates. For example, coordinates of symbol K are (4, B) and, therefore, its binary code is 100 1011. Note that bit position count starts from least significant bit (LSB) position.

TABLE 1.1 ASCII Code Set

Bit position	7	6	5	4	3	2	1	0	
	0	0	0	0	1	1	1	1	
	0	0	1	1	0	0	1	1	
	0	1	0	1	0	1	0	1	
	0	1	2	3	4	5	6	7	
	SPACE								
	!								
	"								
	#								
	0	P	p						
	1	Q	q						
	2	R	r						
	3	S	s						
0000	0	NUL	DLE	%	4	@	T	'	t
0001	1	SOH	DC1		5	U	a	u	
0010	2	STX	DC2		6	V	b	v	
0011	3	ETX	DC3	&	7	A	W	c	w
0100	4	EOT	DC4	,	8	B	X	d	x
0101	5	ENQ	NAK		9	C	Y	e	y
0110	6	ACK	SYN	(D	Z	f	z
0111	7	BEL	ETB			E		g	
1000	8	BS	CAN)	:	F		h	{
1001	9	HT	EM			G	[i	}
1010	A	LF	SUB			H		j	
1011	B	VT	ESC	*	;	I	\	k	
1100	C	FF	FS			J		l	
1101	D	CR	GS		<	K		m	
1110	E	SO	RS	+		L]	n	}
1111	F	SI	US	=		M		o	
				,		N			
						O	^	~	
					>				
				-	?	-		DEL	
				.					
				/					

ASCII is often used with an eighth bit called the *parity bit*. It is added in the most significant bit (MSB) position. This bit is utilized for detecting errors which occur during transmission. We will examine the use of parity bit in detail in Chapter 4, Error Control.

EXAMPLE 1.1 Represent the message 3P.bat in ASCII code. The parity bit position may be kept as 0.

Solution

Bit positions	8	7	6	5	4	3	2	1
3	0	0	1	1	0	0	1	1
P	0	1	0	1	0	0	0	0
.	0	0	1	0	1	1	1	0
b	0	1	1	0	0	0	1	0
a	0	1	1	0	0	0	0	1
t	0	1	1	1	0	1	0	0

The control symbols are code reserved for special functions. Table 1.2 lists the control symbols. Some of the important functions associated with the control symbols are given below.

TABLE 1.2 ASCII Control Symbols			
ACK	Acknowledgement	FF	Form Feed
BEL	Bell	FS	File Separator
BS	Backspace	GS	Group Separator
CAN	Cancel	HT	Horizontal Tabulation
CR	Carriage Return	LF	Line Feed
DC1	Device Control 1	NAK	Negative Acknowledgement
DC2	Device Control 2	NUL	Null
DC3	Device Control 3	RS	Record Separator
DC4	Device Control 4	SI	Shift-In
DEL	Delete	SO	Shift-Out
DLE	Data Line Escape	SOH	Start of Heading
EM	End of Medium	STX	Start of Text
ENQ	Enquiry	SUB	Substitute Character
EOT	End of Transmission	SYN	Synchronous Idle
ESC	Escape	US	Unit Separator
ETB	End of Transmission Block	VT	Vertical Tabulation
ETX	End of Text		

- Functions relating to operation of terminal device, *e.g.* a printer or a video display unit (VDU)

- Carriage return (CR)
- Line feed (LF)
- Functions relating to error control
 - Acknowledgement (ACK)
 - Negative acknowledgement (NAK)
- Functions relating to blocking (grouping) of data characters
 - Start of text (STX)
 - End of text (ETX)
- User definable functions
 - DC1, DC2, DC3, and DC4

DC1 and DC3 are generally used as X-ON and X-OFF for switching the transmitter.

1.2.2 Byte

Byte is a group of bits which is considered as a single unit during processing. It is usually eight bits long though its length may be different. A byte may be an element of a standard code set. But it is not necessary. It may consist of any combination of bits.

7-bit ASCII code is appended with an additional bit that makes it eight bits long. The additional bit can have significance as parity bit or it may not be of any significance than filling the vacant eighth bit position. ASCII character K can be written as an 8-bit byte 01001011.

1.3 DATA TRANSMISSION

There is always need to exchange data, commands, and other control information between a computer and its terminals or between two computers. This information, as we saw in the previous section, is in the form of bits. Data transmission refers to movement of the bits over some physical medium connecting two or more digital devices. There are two options of transmitting the bits, namely, parallel transmission, or serial transmission.

1.3.1 Parallel Transmission

In *parallel transmission*, all the bits of a byte are transmitted simultaneously on separate wires as shown in Figure 1.2. Multiple circuits interconnecting the two

devices are, therefore, required. It is practical only if the two devices, *e.g.* a computer and its associated printer, are close to each other.

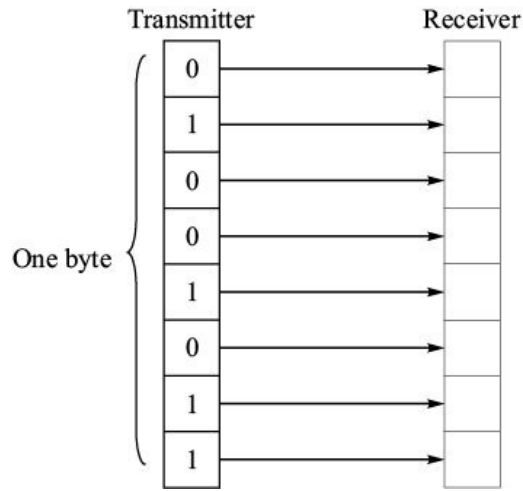


Figure 1.2 Parallel transmission.

1.3.2 Serial Transmission

In *serial transmission*, bits are transmitted serially one after the other (Figure 1.3). The Least Significant Bit (LSB) is usually transmitted first. Serial transmission requires only one circuit interconnecting the two devices. Therefore, serial transmission is suitable for interconnecting devices as a network or for transmission over long distances.

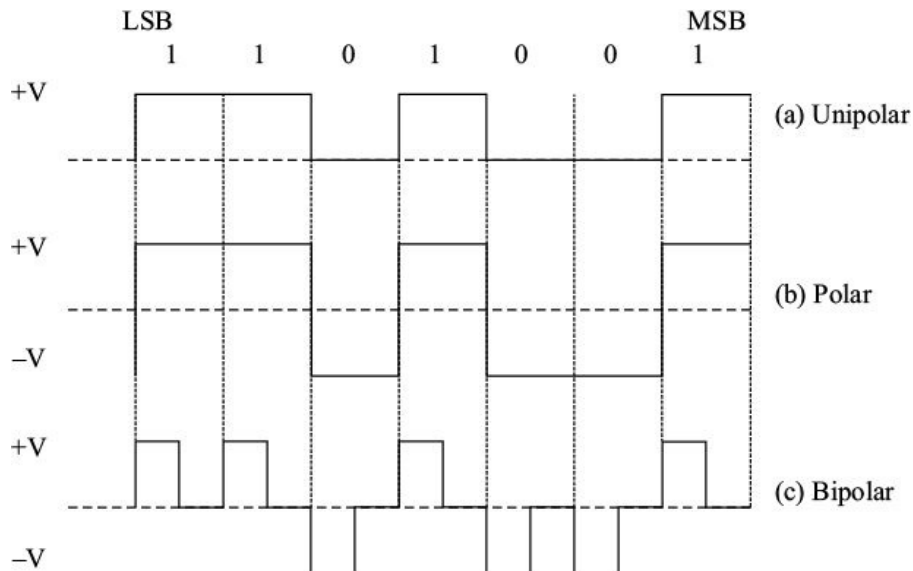


Figure 1.3 Serial transmission.

EXAMPLE 1.2 Write bit transmission sequence of the message given in Example 1.1.

Solution

3	P	.	b	a	t
11001100	00001010	01110100	01000110	10000110	00101110

Bits are transmitted as electrical signals over the interconnecting wires. The two binary states, 1 and 0, are represented by two voltage levels. Digital electrical signals can be categorized in three types based on the way the bits are represented in the signal.

- Unipolar
- Polar
- Bipolar

Unipolar. As the name suggests, *unipolar* signal has only one polarity. Usually positive polarity is used for binary 1 and zero level for binary 0 (Figure 1.3a).

Polar. A *polar* signal has two levels of opposite polarity (Figure 1.3b). Note that the signal never comes back to zero voltage level. It is always in one of the two states, $+V$ or $-V$.

Bipolar. A *bipolar* signal has three level states $+V$, $-V$ and zero level (Figure 1.3c). Bipolar signal is also termed as pseudo-ternary because it has three level states but uses two of these to represent the two binary digits, 0 and 1.

1.3.3 Bit Rate

Bit rate is simply the number of bits which can be transmitted in a second. If t_p is the duration of a bit, the bit rate R will be $1/t_p$. It must be noted that bit duration is not necessarily the pulse duration. In Figure 1.3a the first pulse is of two-bit duration. We will, later, come across signal formats in which the pulse duration is only half the bit duration.

1.3.4 Receiving Data Bits

The signal received at the other end of the transmitting medium is never identical to the transmitted signal as the transmission medium distorts the signal to some extent (Figure 1.4b). There may be additional pulses due to noise spikes. As a result, the receiver has to put in considerable effort in identifying the bits. The receiver must know the time instant at which it should look for a bit. Therefore, the receiver must have synchronized clock pulses which mark the

location of the bits (Figure 1.4c). The received signal is sampled using the clock pulses, and depending on the polarity of a sample, the corresponding bit is identified (Figure 1.4e). Sampling serves another purpose. Unwanted noise spikes that are added during transmission are removed unless a spike occurs at the sampling instant.

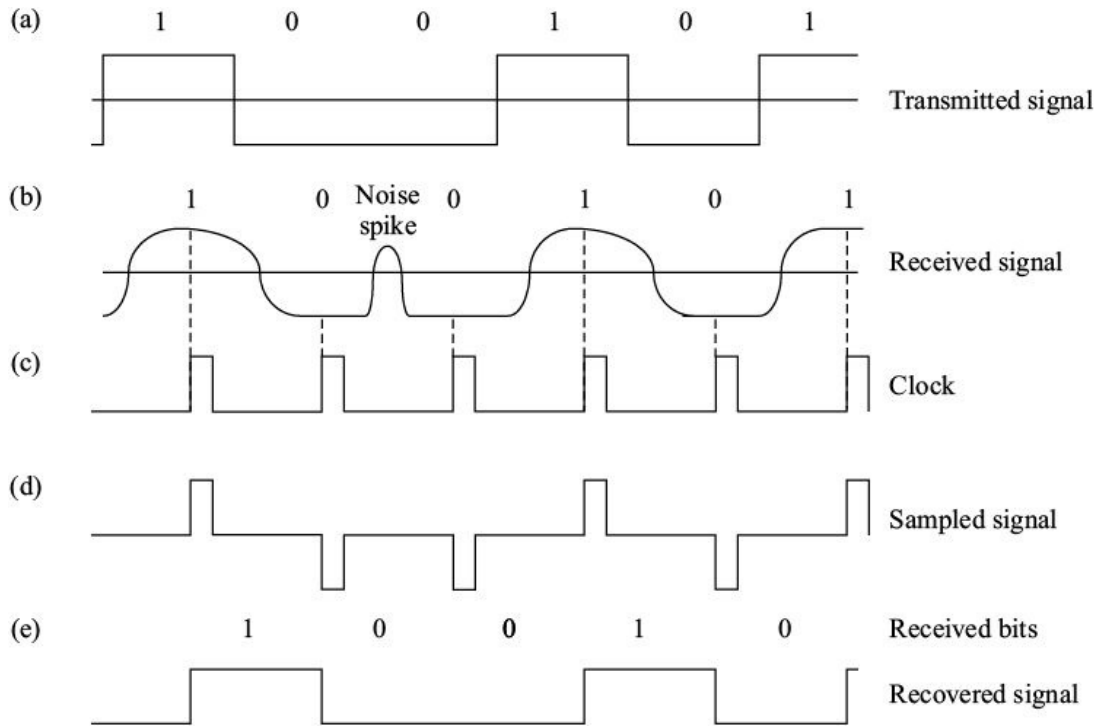


Figure 1.4 Bit recovery.

It is essential that the received signal is sampled at the right instants else it could be misinterpreted. Therefore, the clock frequency should be exactly the same as the transmission bit rate. Even a small difference in frequency will build up as timing error and eventually result in sampling at wrong instants. Figure 1.5 shows two situations when the receiver clock frequency is slightly faster and slightly slower than the bit rate. When clock frequency is faster, a bit may be sampled twice. A bit may be missed when the clock frequency is slower.

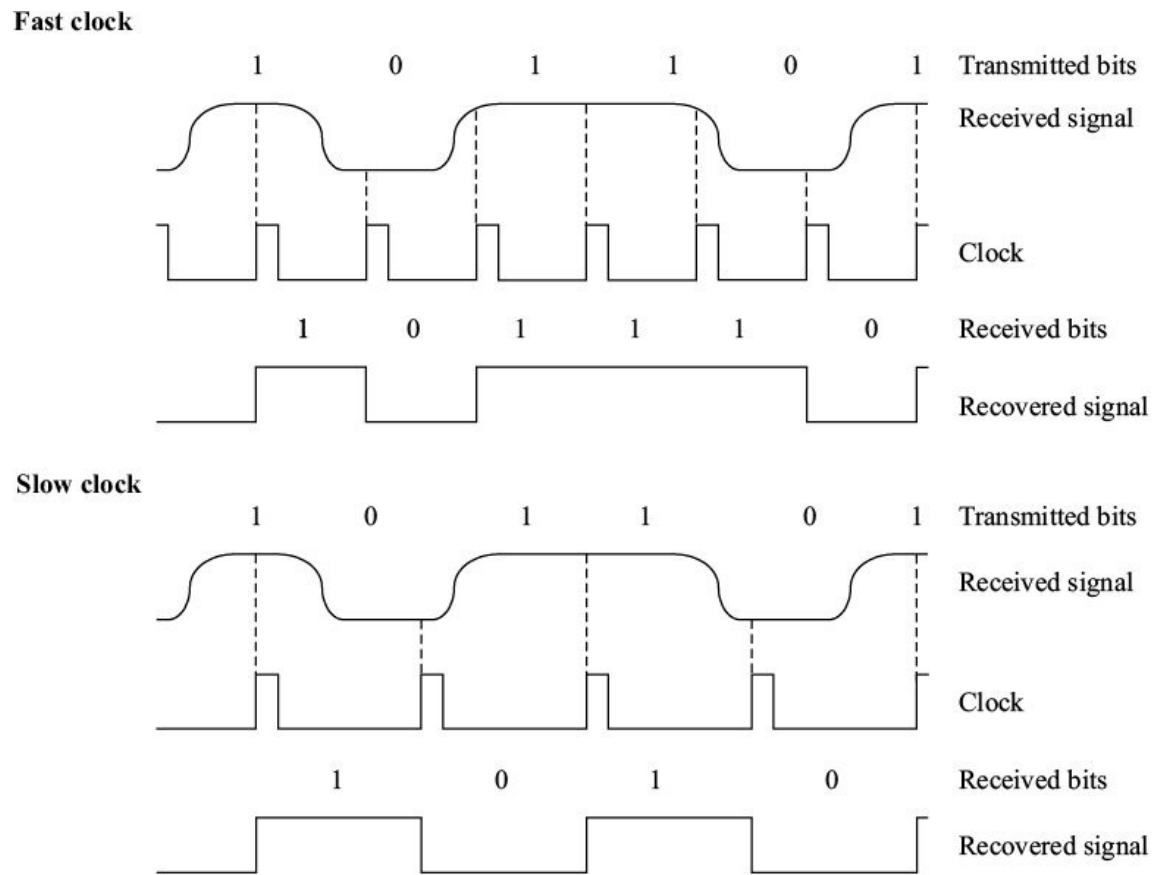


Figure 1.5 Errors due to inaccurate timing.

1.4 MODES OF DATA TRANSMISSION

There are two modes of data transmission:

- Asynchronous transmission
- Synchronous transmission.

We call an action *asynchronous* when the agent performing the action does so whenever it wishes. A *synchronous* action, on the other hand, is performed under control. Asynchronous and synchronous transmissions refer to the modes in which bytes are exchanged between two devices.

1.4.1 Asynchronous Transmission

Asynchronous transmission refers to the case when the sending end commences transmission of bytes at any instant of time. Only one byte is sent at a time and

there is no time relation between consecutive bytes, *i.e.* after sending a byte, the next byte can be sent after arbitrary delay (Figure 1.6). The signal level during the idle state, when no byte is being transmitted, corresponds to 1 as per the accepted practice.

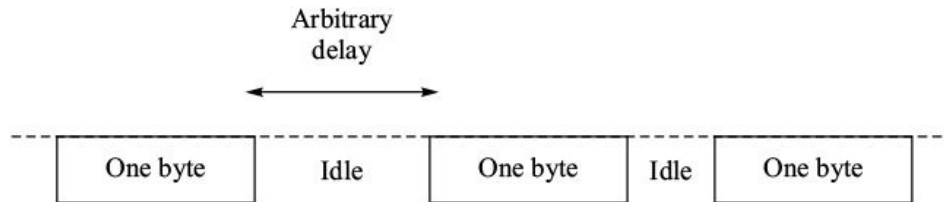


FIGURE 1.6 Asynchronous transmission.

Due to the arbitrary delay between consecutive bytes, the time occurrences of the clock pulses at the receiving end need to be synchronized repeatedly for each byte. This is achieved by providing two extra bits, a *start bit* at the beginning and a *stop bit* at the end of a byte.

Start bit. The *start bit* is always 0 and is prefixed to each byte. At the onset of transmission of a byte, it ensures that the electrical signal changes from idle state 1 to 0 and remains at 0 for one bit duration. The leading edge of the start bit is used as a time reference for generating the clock pulses at the required sampling instants (Figure 1.7). Thus, each onset of a byte results in resynchronization of the receiver clock.

Stop bit. To ensure that the transition from 1 to 0 is always present at the beginning of a byte, it is necessary that polarity of the electrical signal should correspond to 1 before occurrence of the start bit. That is why the idle state is kept at 1. But consider the case when there are two bytes, one immediately following the other and the last bit of the first byte is 0. Since the line does not go in the idle state between the bytes, the transition from 1 to 0 will not occur at the commencement of second byte. To avoid such situations, a *stop bit* is also suffixed to each byte (Figure 1.7). It is always 1 and its duration is set to 1, 1.5, or 2 bits.

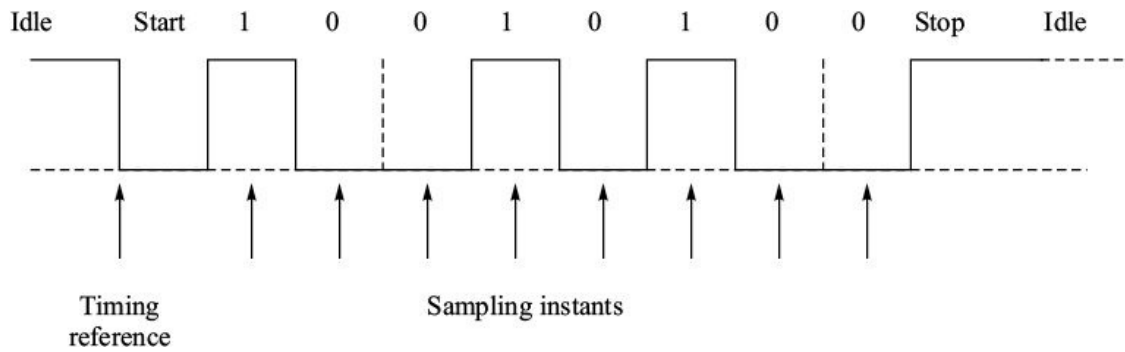


Figure 1.7 Start and stop bits.

EXAMPLE 1.3 Sketch the logic levels for the message ‘HT’ when it is transmitted in asynchronous mode with stop bit equal to one bit. Use ASCII code with parity bit 0 (Figure 1.8).

Solution

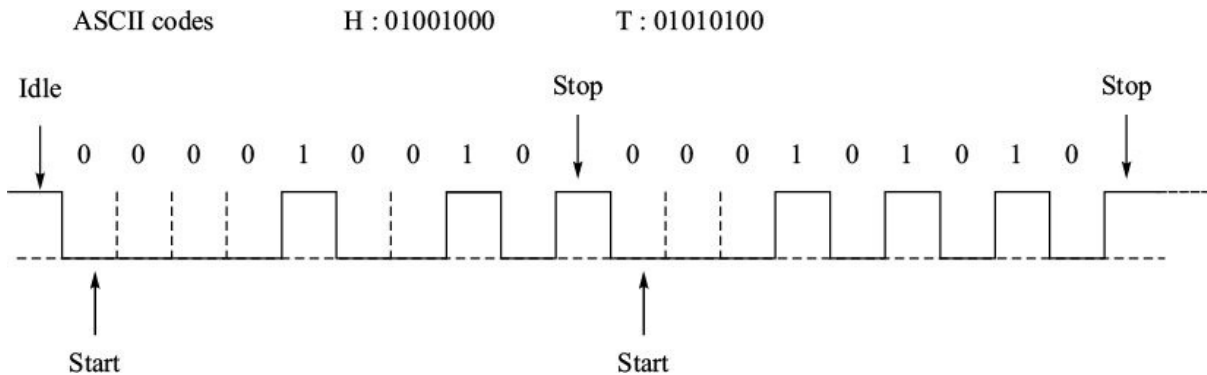


Figure 1.8 Asynchronous transmission of ‘HT’.

1.4.2 Synchronous Transmission

A synchronous action, unlike on asynchronous action, is carried out under the control of a timing source. In synchronous transmission, all the bits are always synchronized to a reference clock irrespective of the bytes they belong to. There are no start or stop bits. Data bytes are transmitted as a block in a continuous stream of bits (Figure 1.9). Even the inter-block idle time is filled with idle characters.

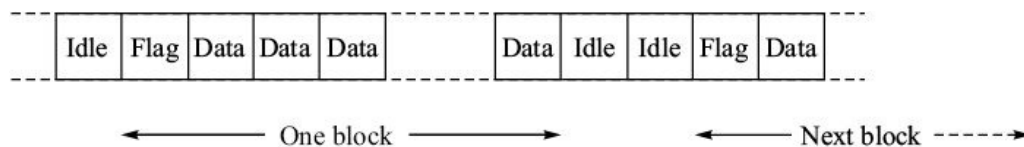


Figure 1.9 Synchronous transmission.

Continuous transmission of bits enables the receiver to extract the clock from

the incoming electrical signals (Figure 1.10). As this clock is inherently synchronized to the bits, the job of the receiver becomes simpler.

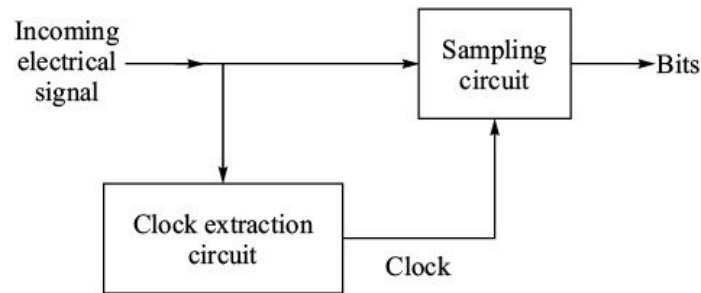


Figure 1.10 Bit recovery in synchronous transmission.

There is, however, still one problem. The data bytes lose their identity and therefore their boundaries need to be identified. A unique sequence of fixed number of bits, called *flag*, is prefixed to each block (Figure 1.9). The flag identifies the start of a block. Receiver first detects the flag and then identifies the boundaries of different data bytes using a counter. Just after the flag, there is first bit of the first byte.

A more common term for data block is frame. A frame contains many other fields in addition to the flag and data fields. It is also possible that the length of the data field may not necessarily be a multiple of bytes. We will discuss frame structures later in Chapter 8, Data Link Layer.

1.5 DIGITAL SIGNAL ENCODING

Signal encoding refers to mapping from the data bits into elements of the digital signal. In Figure 1.3, binary digits 0 and 1 were simply mapped two signal levels. In this section, we will study a variety of other signal encoding schemes. These schemes were devised to overcome problems associated with transmission of the digital signals. In section 1.3.4, we examined the basic tasks to be performed by the receiver for recovering data bits from the received signal. The design of the receiver can be simplified by adopting robust signal encoding scheme. The basic required characteristics of an encoded digital signal are as follows:

- The digital signal should have sufficient level transitions for the clock extraction circuit in the receiver to work properly.
- There should not be any ambiguity in recognizing the binary states of the received signal.

- A signal code should not impose any restriction on the content of bit sequence. In other words, it should be possible to transmit any message.
- The interconnecting signaling links do not provide DC connectivity. Transformers or capacitors are used to block the DC component from transmitter to receiver. Therefore, the line code must ensure that the average signal level (DC component) is zero.
- We will show that a signal can be represented as sum of sine waves of different frequencies and phases. Description of a signal in terms of its constituent frequencies is called its *frequency spectrum*. Various signal encoding schemes have different spectral characteristics, *i.e.* their frequency components and amplitudes of the frequency components are different. These characteristics should suit the transmission medium that carries the digital signal. The signal gets distorted otherwise.

There are several signal encoding schemes which can be divided into three broad classes:

- Unipolar and polar line codes
- Bipolar line codes
 - Alternate Mark Inversion (AMI)
 - High Density Bipolar-3 (HDB-3)
- Block codes.

These codes were devised for various signal transmission applications. Type of transmission media (copper, optical fibre, radio), bit rate, susceptibility to noise are some of the considerations that determine feasibility of a code for a particular application. We will come across these codes in the chapters that follow.

1.6 UNIPOLAR AND POLAR LINE CODES

We introduced unipolar and polar signals while explaining the concept of serial transmission. These signals operate on two levels. Line codes that use unipolar and polar signals are of two kinds:

- Non Return to Zero (NRZ) codes

- Return to Zero (RZ) codes.

1.6.1 Non Return to Zero (NRZ) Codes In this class of codes, the signal level remains constant during a bit duration. There are several types of NRZ codes. Figure 1.11 shows a bit sequence 00101110 represented in various types of NRZ codes.

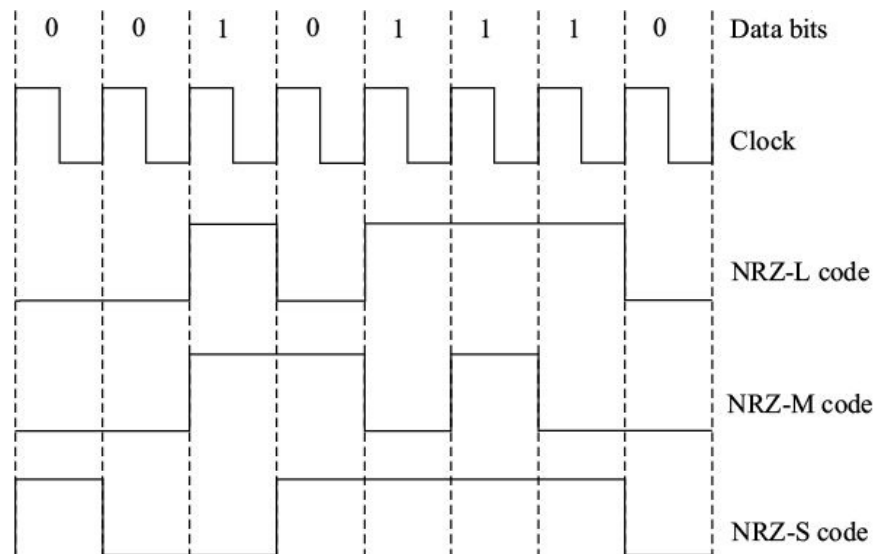


FIGURE 1.11 NRZ signal codes.

NRZ-L. In NRZ-L (Non Return to Zero-level) code, a bit is represented by a voltage level which remains constant during the bit duration.

NRZ-M and NRZ-S. In NRZ-M (Non Return to Zero-Mark), and NRZ-S (Non Return to Zero-Space) codes, it is change in signal level which corresponds to one of the two binary values. Absence of change in level corresponds to the other binary value. The nomenclature ‘Mark’ and ‘Space’ is taken from telegraph code and represents binary 1 and 0 respectively.

In NRZ-M code, signal level changes state on every occurrence of 1. Occurrence of 0 has no effect on the signal level. NRZ-M is also known NRZ-I, Non Return to Zero-Invert on 1s. In NRZ-S, every occurrence of 0 changes the state of signal level.

NRZ-M and NRZ-S are examples of differential encoding. In differential encoding, the signal is decoded by comparing the polarity of adjacent signal elements. Presence and absence of transition of polarity determines the bit. It is easier to detect transitions than absolute levels that determine the binary state, particularly when the received signal level is not steady. Another benefit of

differential encoding is its insensitivity to polarity of the signal. It is very difficult to maintain the polarity of wires interconnecting the two communicating devices. If the leads of a twisted pair are accidentally reversed, all 1s will become 0s and vice versa in NRZ-L encoding.

The NRZ codes are easy to implement but they inherently lack timing information. If a data message consisting of continuous stream of 0s or 1s is encoded using one of the NRZ codes, the resulting digital signal will be without any level transition. Such signal will be as good as no signal for the clock extraction circuit and therefore the receiver will not function properly.

1.6.2 Return to Zero (RZ) Codes

Return to zero (RZ) codes overcome the limitation of NRZ codes in regard to their lack of timing information. RZ codes ensure that the signal has sufficient transitions for any bit pattern. These codes are essentially a combination of NRZ-L and the clock signal. Figure 1.12 shows some examples of RZ codes.

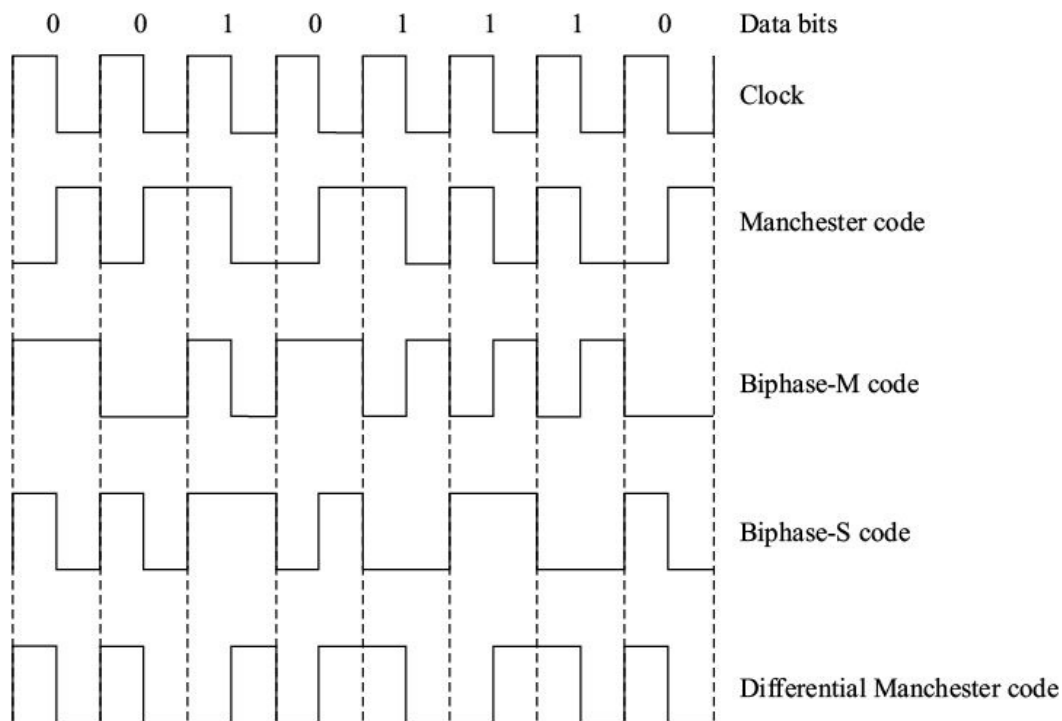


Figure 1.12 RZ signal codes.

Manchester code. In this code, 1 is represented as logical AND of binary 1 and the clock. This produces one clock cycle. For binary 0, the clock cycle is inverted. Note that whatever the bit sequence be, each bit period will have one transition in the middle. Thus the clock extraction circuit of the receiver never

faces a dearth of transitions. Manchester code is widely used in local area networks. It is also called Biphase-L code.

Biphase-M code. In this code there is always a transition at the beginning of a bit interval. Binary 1 has another transition in the middle of a bit interval.

Biphase-S code. In this code also there is always a transition at the beginning of a bit interval. Binary 0 has another transition in the middle of a bit interval.

Differential Manchester code. In this code there is always a transition in the middle of a bit interval. Binary 0 has additional transition at the beginning of the bit interval.

Biphase-M, Biphase-S and Differential Manchester are differential codes and therefore insensitive to the polarity of the received signal.

1.7 BIPOLAR LINE CODES

Bipolar signal codes are used on the metallic transmission media. These codes operate on three signal levels (+V, 0, -V). The middle level (0) represents binary 0 (Space). The other two levels (+V and -V) are used for binary 1 (Mark). We will shortly see how two levels are used for one binary symbol. A bipolar signal code when used for transmitting binary data, is often referred to as pseudo-ternary because it represents only two symbols 1 and 0 instead of three. Three widely used bipolar codes are:

- Alternate Mark Inversion (AMI)
- High Density Bipolar—3 Zeroes (HDB3)
- Bipolar with 8 Zeroes Substitution (B8ZS).

1.7.1 Alternate Mark Inversion (AMI) In AMI code, binary 1 is represented alternately by positive and negative pulses (Figure 1.13a). Alternation of pulse polarity also ensures that the coded signal does not contain any DC component. Each binary 1 introduces a transition, and therefore long string of 1s does not cause loss of synchronization in the receiver. A long string of binary 0s would still be problem.

Some degree of error monitoring can be implemented in AMI using its

polarity alternation property. Any isolated error that deletes a pulse or adds a pulse would cause violation of AMI coding rule. These violations can be used for error monitoring.

1.7.2 High Density Bipolar-3 Zeroes (HDB3) Another widely used bipolar signal code is HDB3 (High Density Bipolar-3 zeroes) code. It is a modification of AMI code and overcomes the problem of long string of binary 0s. If there are more than three consecutive zeroes, a violation pulse (V) is substituted for the fourth zero Figure 1.13b. The violation pulse has the same polarity as the last pulse and therefore it is easily identified at the receiving end. The receiver considers a V pulse as binary 0.

However, this simple scheme has an inherent problem. If there is a long string of zeroes, every fourth pulse will be a V pulse and all the V pulses in the string will be of same polarity. The DC component of such a signal will not average to zero and therefore the signal will suffer distortion during transmission. The problem can be overcome by making the successive V pulses to have alternating polarity. But then we will not be able to identify the V pulses. This is overcome by introduction of an additional bipolar (B) pulse to enable detection of V pulses. The consecutive four zeroes (0000) are, thus, substituted either by 000V or by B00V sequence. Figure 1.13b illustrates the use of HDB3 substitution rules. Note that the B pulses along with 1s of the data sequence follow the polarity alternation rule. The V pulses follow this rule among themselves. Thus the DC value of the signal averages to zero. Like AMI code, HDB3 has error monitoring capability.

1.7.3 Bipolar with 8 Zeroes Substitution (B8ZS) Code B8ZS is commonly used in USA. It is also a modification of AMI code. It overcomes the problem of long string of zeroes by substituting eight consecutive zeroes by 000VB0VB; the first violation pulse (V) is of the same polarity as the last pulse (Figure 1.13c). B pulse then follows the inverse polarity rule. The following V pulse is of the same polarity as preceding B pulse. The last B pulse is of inverse polarity. The receiver recognizes the pattern and interprets the octet as consisting of

all zeroes. B8ZS code also has error monitoring capability like AMI and HDB3 codes.

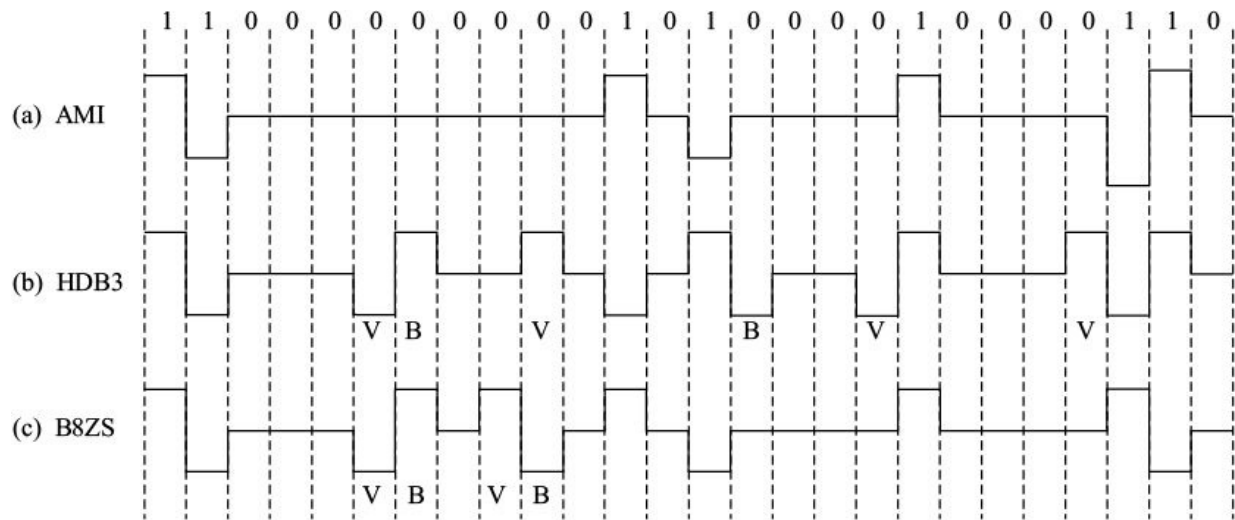


Figure 1.13 Bipolar AMI, HDB3, B8ZS signal codes.

1.8 BLOCK CODES

In *block codes*, the string of bits to be transmitted is divided into blocks of n bits. Each block of n bits is mapped to an m -bit code, where m is greater than n . Since m -bit code set has larger set of combinations (2^m), some of the m -bit combinations will be left as spare when 2^n combinations of n -bit blocks are mapped onto it. This gives some flexibility in choosing the right set of m -bit code words for 2^n data blocks. This flexibility is utilized in two ways:

- Those m -bit combinations are not selected which are likely to result in zero strings of length more than what the receiver can tolerate.
- As far as possible, minimum Hamming distance is maintained between the selected codes. Hamming distance between two code words is simply the number of bit positions that are different in the code words. For example, Hamming distance between 1001 and 1100 is 2. Hamming distance determines the vulnerability of a code to errors. We will discuss this in Chapter 5, Error Control.

The block codes with two signal states are referred to as nB/mB , e.g. 4B/5B, 5B/6B, 8B/10B etc. nB/mB codes are generally used on optical fibre media

because the optical signal can have only two states, ON and OFF. We will describe some of these codes in the later chapters.

Metallic transmission media make it possible to have multiple electrical signal levels. 2B/1Q (2 bits coded into 1 quaternary digit), 4B/3T (4 bits coded into 3 ternary digits) are some of the examples of these codes. An example of 4B/3T mapping is shown in Table 1.3. Figure 1.14 shows a block of four bits 1001 coded into 3 ternary digits (+, -, 0).

TABLE 1.3 4B/3T Code			
Binary word	Ternary code	Binary word	Ternary code
	0 - +		0 + -
	- + 0		+ - 0
	- 0 +		+ 0 -
0000		1000	+ 0 0
0001		1001	+ 0 +
0010	+ - +	1010	+ + 0
0011		1011	+ + +
0100	0 + +	1100	
0101	0 + 0	1101	+ + -
0110	0 0 +	1110	
0111		1111	+ + +
	- + +		

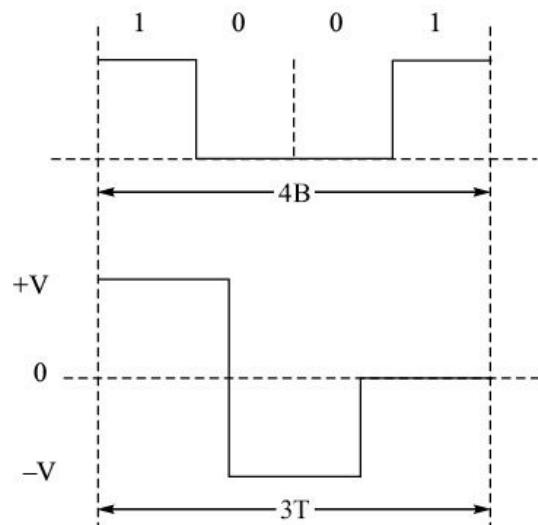


Figure 1.14 Mapping of 1001 (4B) to + - 0 (3T) code.

1.9 FREQUENCY SPECTRUM

Before we proceed any further with data transmission, it is essential to understand the spectral characteristics of a data signal. We have been representing a data signal as series of high and low levels in time domain. When we analyze this signal in frequency domain, we find that it consists of many frequency components. Spectral characteristics of a signal are important because when the signal is transported over a transmission medium, it is to be ensured that the spectral characteristics are not unduly impaired by the medium. We will examine the transmission media and impairments caused by it in the next chapter. We will restrict ourselves to the signal characteristics in this chapter.

1.9.1 Fourier Series

It was shown by Fourier that any periodic signal can be expressed as the sum of infinite series of sine waves having different frequencies. Let us consider a periodic signal $v(t)$, having time period T . Fourier showed that $v(t)$ can be expressed as infinite trigonometric series.

$$v(t) = a_0 + \sum_{n=1}^{\infty} [a_n \cos (2pnft) + b_n \sin (2pnft)]$$

where

$$f = \frac{1}{T}$$

$$a_0 = \frac{1}{T} \int_{-T/2}^{T/2} v(t) dt \quad a_n = \frac{1}{T} \int_{-T/2}^{T/2} v(t) \cos (2pnft) dt \quad b_n = \frac{1}{T} \int_{-T/2}^{T/2} v(t) \sin (2pnft) dt$$

Thus, the frequency spectrum of $v(t)$ consists of a DC component having value a_0 and frequency components that are harmonics of the fundamental frequency f . Amplitude c_n and the phase q_n of the n th harmonic are given by $c_n = (a_n^2 + b_n^2)^{1/2}$

$q_n = -\tan^{-1} (b_n/a_n)$ We can use the above equations to find the frequency components and their amplitudes of a data signal. Let us consider a unipolar data signal consisting of alternating 1s and 0s. If the bit duration is t_p , the resulting signal is a square wave having time period $T = 2t_p$ (Figure 1.15).

Using the above equations, we get

$$a_0 = \frac{A}{2}, \quad a_n = \frac{A \sin (n\pi/2)}{n\pi/2}, \quad b_n = 0, \quad f_n = \frac{n}{2t_p} = \frac{nR}{2}.$$

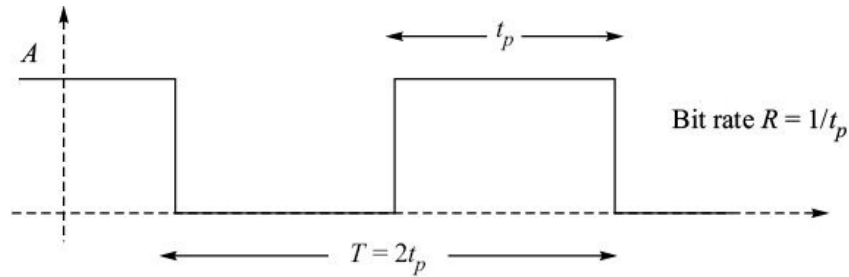


Figure 1.15 Periodic data signal.

Thus, a repetitive 1 and 0 bit pattern has the fundamental frequency (f) which is half the bit rate and odd harmonics of the fundamental frequency. Figure 1.16 shows the amplitude plot of the frequency components. It appears at first glance that transmitting this signal would require a channel with infinite bandwidth but it is to be noted that the amplitudes of the frequency components decrease as we go up in the spectrum. Therefore, a good likeness of the signal can be obtained by significant first few frequency components of the spectrum.

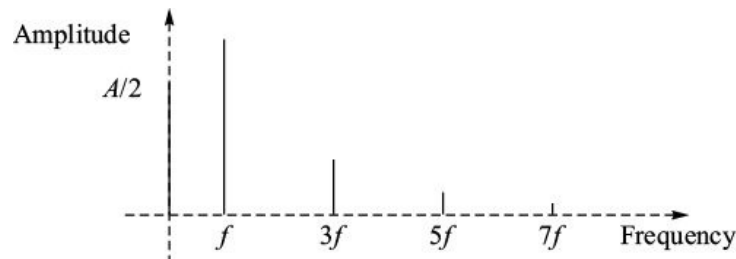


Figure 1.16 Spectral components of repetitive 1 and 0 bit stream.

Note that the above signal contains DC component. The transmission media do not allow DC component to pass. Therefore we use polar signals in place of unipolar signals.

The data signal with alternating 1s and 0s is the most frequently changing data signal. If the data signal consists of random sequences of 1s and 0s, the spectrum becomes continuous and extends up to the origin (Figure 1.17). In the continuous spectrum, we represent power spectral density $S(f)$ instead of amplitudes on the Y-axis. Power spectral density, when integrated over a frequency band, gives the combined power of the frequency components contained in that band.

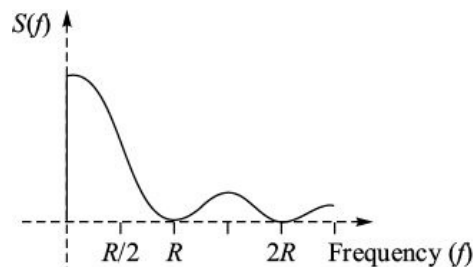


Figure 1.17 Frequency spectrum of random bit pattern.

Figure 1.18 shows the plot of power spectral density of various signal codes. The frequency axis has been normalized to the bit rate. As can be seen, the NRZ

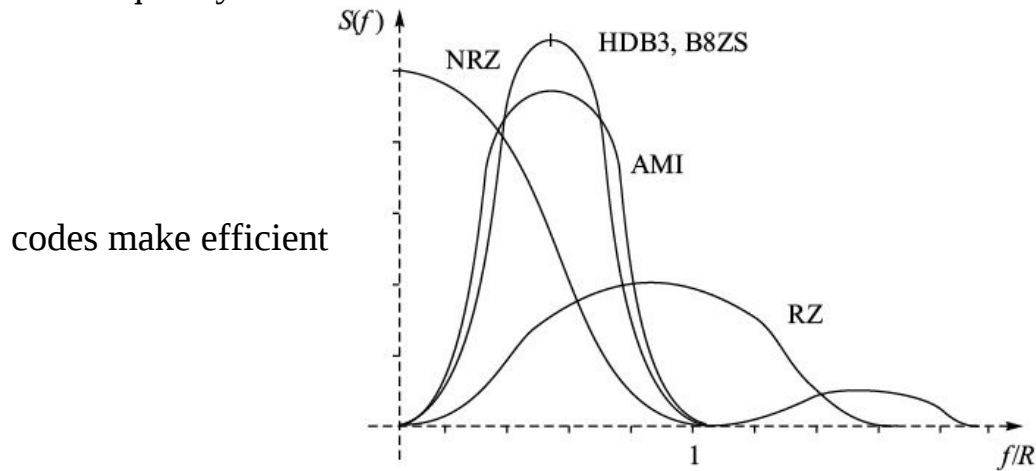


Figure 1.18 Power spectral densities of signal encoding schemes.

use of the bandwidth since most of the signal power in NRZ signals is concentrated below half the bit rate. Little concentration of signal power at the clock frequency indicates these codes inherently lack timing information.

Power spectral density plot of the RZ codes shows that bulk of the signal power is in the frequency range 0.5–1. Significant amount of signal power is present at the bit rate frequency indicating presence of timing information in the encoded signal.

The AMI and HDB3 codes have no DC component and very small low frequency content. The bandwidth requirement is equal to the bit rate. Most of the energy is concentrated in relatively small spectrum around mid of the frequency band.

1.10 TRANSMISSION CHANNEL

A *transmission channel* transports the electrical signals from the transmitter to the receiver. It is characterized by two basic parameters, bandwidth and signal-to-noise ratio. These parameters determine the ultimate information-carrying capacity of the channel. Nyquist derived the limit of bit rate considering a perfectly noiseless channel. Nyquist's theorem states that if B is the bandwidth of a transmission channel which carries a signal having L levels, the maximum bit rate R is given by $R = 2B \log_2 L$

Polar signal shown in Figure 1.3b has two levels. Thus $R = 2B$. In other words, a polar signal requires transmission channel bandwidth which is half the bit rate. The number of levels (L) can be more than two, as we shall see shortly. It appears from this theorem that a limited bandwidth channel can carry however high bit rate by increasing the number of levels. This is not entirely true if the effect of noise present in the channels is taken into account. Shannon extended Nyquist's work to include the effect of noise. If signal-to-noise ratio (signal power divided by noise power) is S/N , the maximum bit rate is given by $R = B \log_2 \left(1 + \frac{S}{N} \right)$

Signal-to-noise ratio is usually expressed in dB (decibels).

Signal-to-noise ratio (dB) = $10 \log_{10} (S/N)$ Shannon's equation puts a limit on the number of levels L . If bandwidth is 3000 Hz and signal-to-noise ratio is 30 dB (30 dB = 1000), then the maximum bit rate can be $R = 3000 \log_2 (1 + 1000)$ @ 30,000 bits/s Number of levels (L) required for the bit rate of 30,000 bits per second, can be computed from Nyquist's theorem.

$$30,000 = 2 \log_2 L$$

$$L = 32$$

Very sophisticated equipment is required for achieving such high bit rates on a voice grade telephony channel that has bandwidth of 3100 Hz.

1.10.1 Bauds

When bits are transmitted as an electrical signal having two levels, the bit rate and the modulation rate of the electrical signal are the same (Figure 1.19). Modulation rate is the rate at which the electrical signal changes its levels. It is expressed in *bauds* ('per second' is implied). Note that there is one to one correspondence between bits and electrical levels.

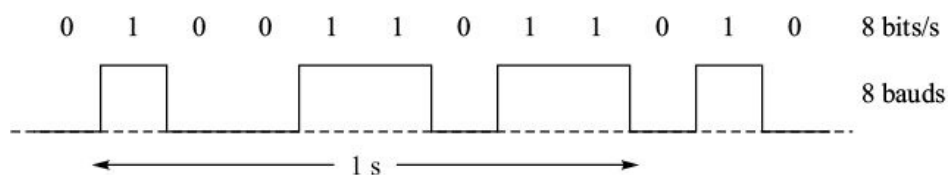


Figure 1.19 Baud rate for two-level modulation.

It is possible to associate more than one bit to one electrical level. For example, if the electrical signal has four distinct levels (quaternary signal), two

bits can be associated with one level of the quaternary electrical signal (Figure 1.20). The four voltage levels are so chosen that there is equal spacing between the levels. Note that the changes in electrical signal take place at

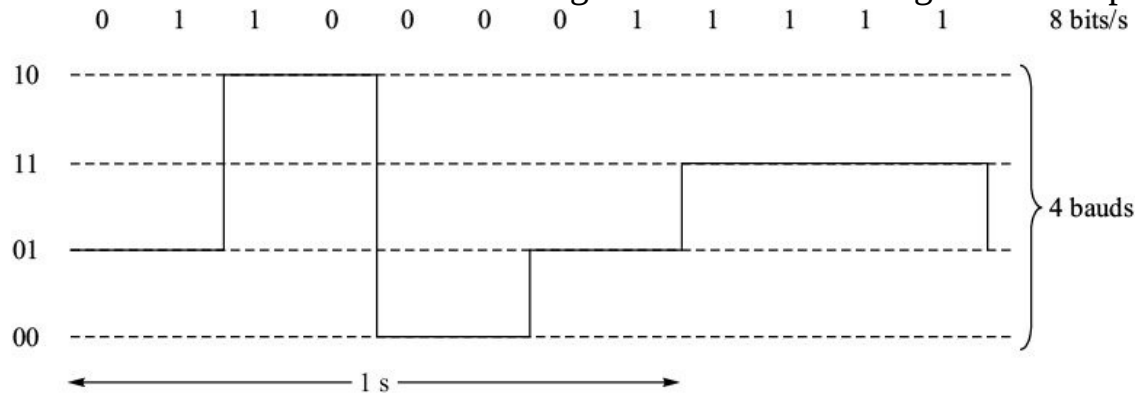


Figure 1.20 Baud rate for four-level modulation.

half of the bit rate. Therefore, the baud rate is half of the bit rate. The modulation scheme shown in the figure is used in ISDN (Integrated Services Digital Network) interface and is referred to 2B1Q, where B stands for binary bits and Q for quaternary digit.

If the electrical signal is allowed to have L levels, we can associate $n = \log_2 L$ bits with each level. Therefore, baud rate (r) is given by $r = R/n = R/\log_2 L$

If we substitute the value of R from the Nyquist theorem, we get $r = 2B$. Thus for any band limited channel having bandwidth B , we can achieve maximum baud rate of $2B$ bauds. By increasing the number of levels, we can increase the bit rate.

EXAMPLE 1.4 What is the maximum possible baud rate of a voice channel having bandwidth of 3100 Hz?

Solution Baud rate $r = 2 \times 3100 = 6200$ bauds.

1.10.2 Baseband Transmission

When a digital signal is transmitted on the medium using one of the signal codes discussed earlier, it is called *baseband transmission*. In baseband transmission, DC signal is modulated using one of the line coding schemes. Multiple levels are used to increase the bit rate over a band limited channel.

1.10.3 Modem

Instead of baseband transmission using a DC signal for level modulation, we can use a sinusoidal signal for modulation. A sinusoidal has three basic attributes,

amplitude, frequency, and phase. The binary signal modulates one or more of these signal attributes. Phase modulation in particular allows large number of phase states. Each phase state can be associated with several bits and bit rate can be very high even 56 kbps on a voice grade line.

The sine wave carries the information and is, therefore termed as ‘carrier’. The device which performs modulation is called a *modulator* and the device which recovers the binary signal from the modulated carrier is called a *demodulator*. In data transmission, we usually come across devices which perform both modulation as well as demodulation functions and these devices are called *modems*. Modems are required when data is to be transmitted over long distances. In a modem, the input digital signal modulates a carrier which is transmitted to the distant end. At the distant end, another modem demodulates the received carrier to get the digital signal. A pair of modems is, thus, always required. We discuss modems in detail in Chapter 4, Data Line Devices.

1.11 DATA COMPRESSION

Data compression refers to reducing the number of bits that need to be transmitted for exchanging a given volume of information. Data compression techniques are also used for reducing the data storage requirements. Virtually all forms of data contain redundancy, *i.e.* information content is less than what the data representation is capable of. By making use of more efficient data representation methods, redundancy can be reduced. Representing English alphabets in 7-bit ASCII is one way of representing data. It would appear at the first glance that seven is the minimum number of bits required to represent 128 symbols and control characters, and there is no redundancy in this code. It is not entirely true. To understand this, we need to define information and then examine how it can be coded in an efficient manner.

1.11.1 Information

Typically one thinks of information as having to do with knowledge. Gaining information signifies acquiring knowledge that was not there earlier. A quantitative measure of amount of information is required to deal with it mathematically. We can develop a measure of amount of information intuitively as explained below.

Gaining information is equivalent to reducing uncertainty. If outcome of an

event is known with certainty (say with probability equal to unity), it does not add to the knowledge. On the other hand, if the outcome is uncertain, its occurrence adds to the knowledge. Therefore, amount of information $I(x)$ is inverse to probability $P(x)$ of occurrence of event x . If we define $I(x) = 1/P(x)$, receipt of a bit having equal probability of occurrence of 1 or 0 will convey information equal to 2 units ($= 1/0.5$).

We need to examine if this definition holds good for incremental information. With this definition of information, receipt of every additional bit adds two units of information. Let us analyze this definition from another angle. If we consider the first two bits together, there are four possible outcomes each having probability of 0.25. Thus the amount of information acquired when two bits are received is 4 units ($=1/0.25$). If we take three bits together, there are eight equally probable outcomes, and the accumulated information will be 8 units ($=1/0.125$), not 6 as we concluded above.

Suppose we define information $I(x) = \log_2 1/P(x)$. Then, Accumulated information on receipt of first bit $I(x) = \log_2 (1/0.5) = 1$

Accumulated information on receipt of second bit $I(x) = \log_2 (1/0.25) = 2$

Accumulated information on receipt of third bit $I(x) = \log_2 (1/0.125) = 3$

Accumulated information on receipt of fourth bit $I(x) = \log_2 (1/0.0625) = 4$

In this case every additional bit generates one unit of information. This definition of information meets our requirements. We have taken logarithmic base as 2 for convenience. The unit of information for base 2 is called *bit*. We could have used any other logarithmic base. It would have amounted to multiplying the measure of information by a constant factor and changing the unit of information measure.

Bit as unit of information has direct correspondence with number of physical bits required to represent the information. It assumes, however, that 0 and 1 have equal probability of occurrence. Let us take an example where the probabilities of occurrence of 1 and 0 are not equal and estimate the average information of a bit.

$$P(1) = 0.2$$

$$I(1) = \log_2 (1/0.2) = 2.322$$

$$P(0) = 0.8$$

$$I(0) = \log_2 (1/0.8) = 0.322$$

Average information per bit $P(1) I(1) + P(0) I(0) = 0.722$.

In this case, 0 occurs more frequently but it conveys less information than 1. Therefore, average information per bit is less than one. We will need more than one bit to represent one *bit* of information in such situations. In the following sections we will assume equal probability of occurrence for 1 and 0. We will not indicate the logarithmic base but it will always be 2.

1.11.2 Entropy

Concept of average information content per outcome as defined above for binary outcomes 1 and 0 can be generalized to coded messages. Consider an information source that generates messages in form of symbols taken from a source alphabet (a_1, a_2, \dots, a_n) . A message can be in form of words consisting of string of symbols. But we will consider each symbol as a separate message for the sake of simplicity. If P_i is probability of occurrence of symbol a_i , information content of the message a_i is $1/\log P_i$. We can determine average information (H) contained in a message generated by a source by multiplying the information of each message by its probability of occurrence and taking summation over the entire alphabet set.

Average information per message H is termed as *entropy*. Entropy of a source serves a very important purpose in binary encoding process. It imposes lower bound on the average number of bits required to encode a source alphabet using binary 1 and 0. The alphabet of the following example requires average number of 2.893 bits per symbol since each bit can carry information content of one unit.

EXAMPLE 1.5 A source generates messages from alphabet set (a, b, c, d, e, f, g, h) . Calculate entropy of the source for the probability of occurrence of the symbols indicated within the brackets.

$a (0.48), b (0.08), c (0.12), d (0.02), e (0.12), f (0.04), g (0.06), h (0.08)$ **Solution**

The entropy (H) of the source $H = 0.48 \log (1/0.48) + 0.08 \log (1/0.08) + 0.12 \log (1/0.12) + 0.02 \log (1/0.02) + 0.12 \log (1/0.12) + 0.04 \log (1/0.04) + 0.06 \log (1/0.06) + 0.08 \log (1/0.08) = 2.367$

1.11.3 Redundancy

If we wish to encode the alphabet set of the above example, one obvious way is to assign a fixed length 3-bit code to each of its eight symbols. But we will not be utilizing full information carrying capability of the code since entropy of the source is 2.367. We can reduce the average number of bits required to encode

the alphabet by using variable length code instead of fixed 3-bit code. Some of the symbols can be assigned fewer than three bits so that average code length is reduced. Average code length L is the expected value as given below.

$L = \sum_{i=1}^n P_i L_i$ where P_i is the probability of occurrence of symbol a_i and L_i is the length of its code word. The frequently occurring symbols are assigned shorter code words to reduce the average code length. Unutilized capability of the code is called *redundancy* (R) and is defined as below.

$R = L - H = \sum_{i=1}^n P_i L_i - \sum_{i=1}^n P_i \log (1/P_i)$ We would like redundancy to be as low as possible. It can be shown that for an optimum code redundancy R is less than 1. Let us see how.

For a code word to be optimal, its length L_i should be equal to its information content I_i rounded to the next higher integer. In other words, $L_i - I_i < 1$

If probability of occurrence of the code word is P_i , then on multiplying the above inequality by P_i , we get $P_i L_i - P_i I_i < P_i$ Taking summation over the entire alphabet set, we get $\sum_1^n P_i L_i - \sum_1^n P_i I_i < \sum_1^n P_i$ But

$$\sum_1^n P_i = 1, \sum_1^n P_i L_i = L, \sum_1^n P_i I_i = H$$

Therefore,

$$L - H = R < 1.$$

EXAMPLE 1.6 A source generates messages from alphabet set (a, b, c, d, e, f, g, h) with probabilities as indicated below. Calculate average code length and redundancy.

Symbol	Probability	Code
a	0.48	1
b	0.08	0000
c	0.12	001
d	0.02	01111
e	0.12	010
f	0.04	01110
	0.06	

g	0.08	0110
h		0001

Solution The entropy H of the source is: $H = 0.48 \log (1/0.48) + 0.08 \log (1/0.08) + 0.12 \log (1/0.12) + 0.02 \log (1/0.02) + 0.12 \log (1/0.12) + 0.04 \log (1/0.04) + 0.06 \log (1/0.06) + 0.08 \log (1/0.08) = 2.367$

The average code length (L) is $L = 0.48 \cdot 4 + 0.08 \cdot 3 + 0.12 \cdot 5 + 0.12 \cdot 3 + 0.04 \cdot 5 + 0.06 \cdot 4 + 0.08 \cdot 4 = 2.38$

The redundancy (R) is $R = 2.38 - 2.367 = 0.013$.

1.11.4 Encoding for Compression

Information, entropy, and average length of code provide insights into the design and evaluation of data compression methods. We will look at two data compression methods, Shannon-Fano algorithm and Huffman algorithm. It is believed that the Huffman algorithm is most optimal and cannot be improved upon. It provides a benchmark to which other data compression methods can be compared. The terminology and assumptions associated with data compression are described first to ease the understanding.

Distinct code. A code is distinct if each code word is distinguishable from every other code word, *i.e.* each symbol is mapped to a distinct code word.

Uniquely decodable code. A distinct code is uniquely decodable if every code word is identifiable when immersed in a string of code words.

A distinct code may not be uniquely decodable. Consider a code set (0, 1, 10, 11) of source alphabet (a, b, c, d). This code is distinct but it is not uniquely decodable because coded message 11 could be decoded as c or bb. Distinct and uniquely decodable are two desirable features of any code.

Prefix code. A uniquely decodable code is a prefix code if no code word is a proper prefix of any other code word. The code in the example above is not a prefix code because 1 is proper prefix of code words 10 and 11, and it is also a code word. All uniquely decodable codes are not prefix codes, *e.g.* (0, 011111, 11) is uniquely decodable but is not a prefix code. String 110111110011 can be interpreted as parsed into codes 11, 011111, 0,0, and 11 only. Note that to parse the string into code words required look-ahead. Prefix codes are instantaneously decodable and therefore do not require look-ahead.

Minimal prefix-code. A minimal prefix code is a prefix code such that if x is a

proper prefix of some code word, then string x_0 and x_1 are either code words or proper prefix of a code word. (00, 01, 10) is not minimal prefix code since 1 being proper prefix of 10, minimal prefix criterion requires that 11 should either be a proper prefix or a code word.

Minimal prefix codes do not have code words which are longer than necessary. In the above example, we can use a shorter code word 1 in place of longer code word 10.

1.11.5 Shannon-Fano Code

Shannon-Fano code has advantage of its simplicity. The message symbols are listed in descending order of their probabilities. This list is then divided in such a way as to form two groups of as nearly equal probabilities as possible. Each symbol in the first group is assigned 0 as the first bit. The symbols of the second group are assigned 1 as the first bit. Each of these groups is then further divided and assigned second bit in the same manner. The process is continued until only one symbol is left in each of the subdivided groups (Table 1.4).

TABLE 1.4 Shannon-Fano Code

	I	II	III	IV	V
	0.48				
	0.12				
<i>a</i>	0.12	0	0	0	0
<i>c</i>	0.08	1	10	100	100
<i>e</i>	0.08	1	10	101	101
<i>b</i>	0.06	1	11	110	1100
<i>h</i>	0.04	1	11	110	1101
<i>g</i>		1	11	111	1110
<i>f</i>		1	11	111	1111
<i>d</i>	0.02	1	11	111	1111

EXAMPLE 1.7 Determine Shannon-Fano code for the following alphabet. The probability of occurrence of the symbols is indicated within the brackets. What is the average code length?

a (0.48), *b* (0.08), *c* (0.12), *d* (0.02), *e* (0.12), *f* (0.04), *g* (0.06), *h* (0.08) **Solution**
 Average code length (L) $L = 0.48 + 0.08 \cdot 4 + 0.12 \cdot 3 + 0.02 \cdot 5 + 0.12 \cdot 3 + 0.04 \cdot 5 + 0.06 \cdot 4 + 0.08 \cdot 4 = 2.38$

Shannon-Fano algorithm yields minimal prefix code. Average length of the code meets the criterion $H \leq L < H + 1$. Shannon-Fano code is not guaranteed to

produce an optimal code.

1.11.6 Huffman Code

Huffman code is generated by drawing a binary tree consisting of leaf nodes and branch nodes. Each leaf node corresponds to a symbol and has weight equal to probability of its occurrence. A branch node is formed by merging two nodes and has weight equal to sum of the weights of the merged nodes.

The binary tree is drawn in steps by merging two nodes having the lowest weights into a branch node. To understand the algorithm, we will generate Huffman code for the following alphabet set. The probability of occurrence of the symbols is indicated within the brackets.

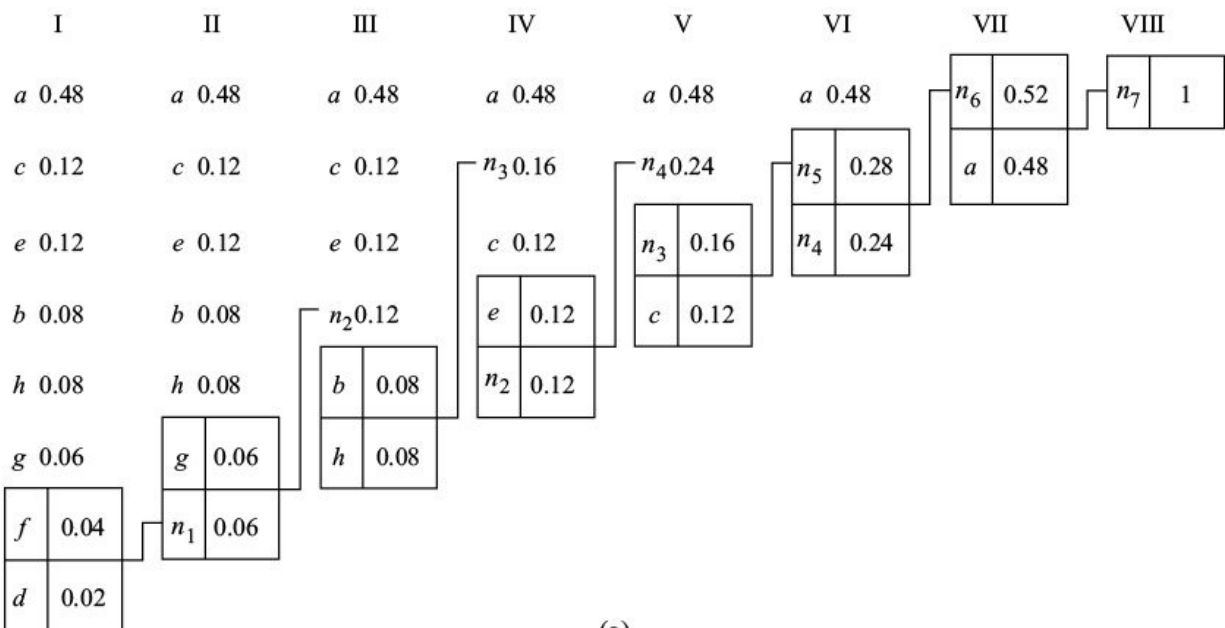
$a (0.48), b (0.08), c (0.12), d (0.02), e (0.12), f (0.04), g (0.06), h (0.08)$

	The symbols are arranged in descending order their probabilities (Figure 1.21a).
<i>Step I:</i>	Symbols having equal probability can be kept in any order. Symbols f and d have the lowest weights and are merged into branch node n_1 having weight 0.6 (Figure 1.21b).
<i>Step II:</i>	We are left with symbols a, b, c, e, g, h , and branch node n_1 . These are rearranged in descending order of weights. g and n_1 have the lowest weights and are merged into branch node n_2 having weight 0.12.
<i>Step III:</i>	We are left with symbols a, b, c, e, h , and branch node n_2 . These are rearranged in descending order of weights. b and h have the lowest weights and are merged into branch node n_3 having weight 0.16.
<i>Step IV:</i>	We are left with symbols a, c, e , and branch nodes n_2 and n_3 . These are rearranged in descending order of weights. e and n_2 have the lowest weights and are merged into branch node n_4 having weight 0.24.
<i>Step V:</i>	We are left with symbols a, c , and branch nodes n_3 and n_4 . These are rearranged in descending order of weights. c and n_3 are the next to be merged into n_5 having weight 0.28.
<i>Step VI:</i>	We are left with symbol a and branch nodes n_4 and n_5 . These are rearranged in descending order of weights. Next to be merged are nodes n_4 and n_5 . Branch node n_6 so formed has weight 0.52.
<i>Step VII:</i>	We are left with symbol a and the branch node n_6 . These merge into root node of weight 1.

Having drawn the tree (Figure 1.21b), it is time to assign binary codes to the symbols. Length of code word for a symbol is determined by its distance from root node. Note that the least probable symbols are farthest from the root node and therefore will have longest code words.

There can be many alternatives for specifying the actual bits; it is necessary only that the resulting code has prefix property. The usual practice is to assign 1 to the right side branch of a branch node and 0 to the left side branch. Huffman code for a symbol is derived by tracing the path from the root node to the respective leaf node and writing down in sequence the 1s and 0s that are encountered along the path.

Symbol	Probability	Code
	0.48	
<i>a</i>	0.08	1
<i>b</i>	0.12	0000
<i>c</i>	0.02	001
<i>d</i>	0.12	01111
<i>e</i>	0.04	010
<i>f</i>	0.06	01110
<i>g</i>		0110
<i>h</i>	0.08	0001



(a)

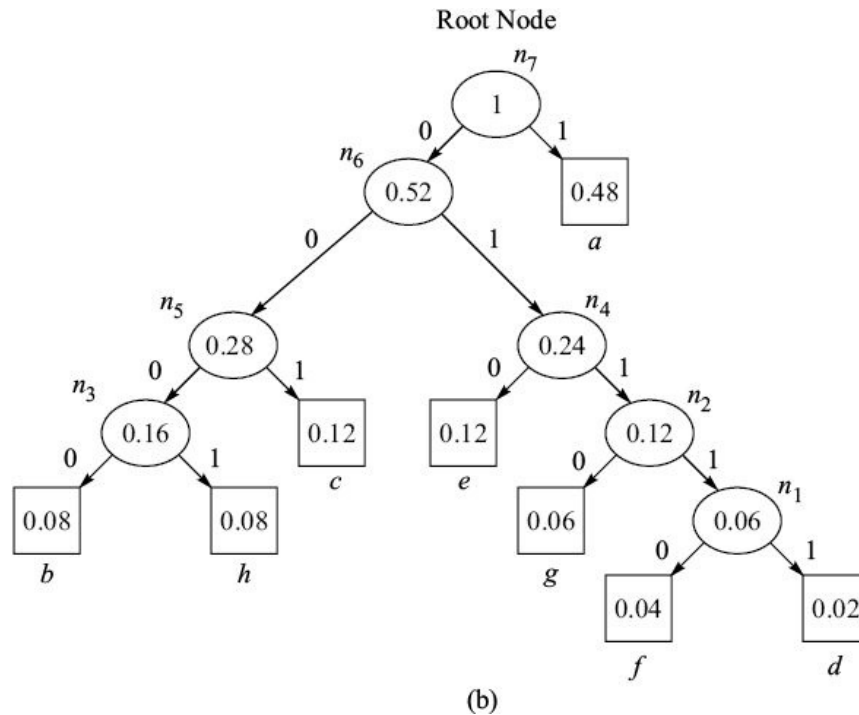


Figure 1.21 Huffman code.

This is the same code for which we calculated entropy as 2.367, average code length 2.38, and redundancy 0.013 in Example 1.6. Huffman algorithm yields minimal prefix code. It can be shown that the code has minimum redundancy and thus is optimal.

1.12 DATA COMMUNICATION

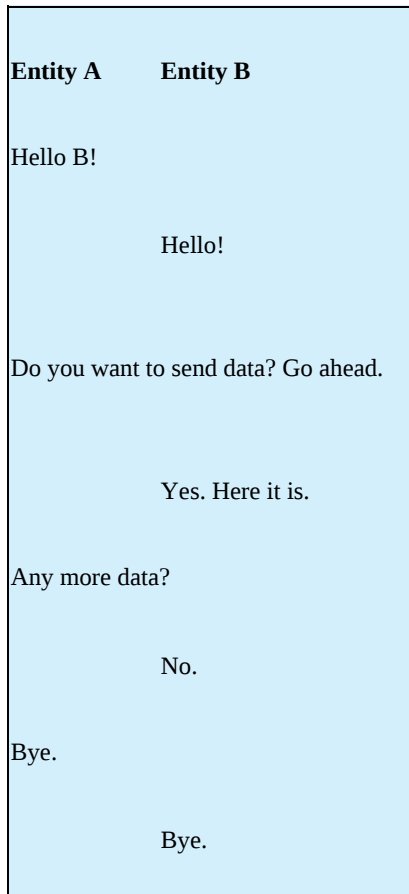
Communication and *transmission* terms are often interchangeably used, but it is necessary to understand the distinction between the two activities. Transmission is physical movement of information and concerns issues like bit polarity, synchronization, clock, electrical characteristics of signals, modulation, demodulation, *etc.* We have been examining these data transmission issues.

Communication has a much wider connotation than transmission. It refers to meaningful exchange of information between the communicating entities. Therefore, in *data communications* we are concerned with all the issues relating to exchange of information in the form of a dialogue, *e.g.* dialogue discipline, interpretation of messages, and acknowledgements.

Communication can be synchronous and asynchronous.

1.12.1 Synchronous Communication

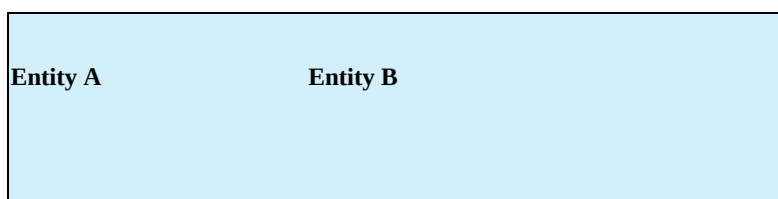
In *synchronous* mode of the communication, the communicating entities exchange messages in a disciplined manner. An entity can send a message when it is permitted to do so.



The dialogue between the entities A and B is synchronized in the sense that each message of the dialogue is a command or response. Physical transmission of data bytes corresponding to the characters of these messages could be in synchronous or asynchronous mode.

1.12.2 Asynchronous Communication

Asynchronous communication, on the other hand, is less disciplined. A communicating entity can send message whenever it wishes to.



Hello B!

Hello! Here is some data.

Here is some data. Here is more data.

Did you receive what I sent? Yes. Here is more data. Please acknowledge.

Acknowledged. Bye.

Bye.

Note the lack of discipline in the dialogue. The communicating entities send message whenever they please. Here again, physical transmission of bytes of the messages can be in synchronous or asynchronous mode.

We will come across many examples of synchronous and asynchronous communication in this book when we discuss protocols. Protocols are the rules and procedures for communication.

1.13 DIRECTIONAL CAPABILITIES OF DATA EXCHANGE

There are three possibilities of data exchange:

- Transfer in both directions at the same time.
- Transfer in both directions, but only in one direction at a time.
- Transfer in one direction only.

Terminology used for specifying the directional capabilities is different for data transmission and for data communication (Table 1.5). In most of the literature, however, the terminology is used interchangeably.

TABLE 1.5 Terminology for Directional Capabilities		
Directional capability	Transmission	Communication

One direction only	Simplex (SX)	One Way (OW)
One direction at a time	Half Duplex (HDX)	Two-Way Alternate (TWA)
Both direction at the same time	Full Duplex (FDX)	Two-Way Simultaneous (TWS)

1.14 LINE CONFIGURATIONS

The communicating devices can be interconnected in variety of ways. Traditionally, there are two basic configurations, *point-to-point* and *point-to-multipoint* (Figure 1.22).

Communication between two directly interconnected devices is referred to as point-to-point communication. This configuration is used for communication between two computers or between a terminal and a computer. Mode of communication in point-to-point configuration can be synchronous or asynchronous. In synchronous mode, one of the two computers controls the dialogue and it is called primary or master station (Figure 1.22a). The corresponding names of the other station are secondary or slave station. The secondary station is permitted to transmit only when it is invited to transmit by the primary station. Asynchronous mode of point-to-point communication can be used between two computers having primary status. They do not require permission of the other station to send a message.

In point-to-multipoint configuration, there is one host and several tributary stations (Figure 1.22b). All the tributary stations share a common transmission media. The host decides which tributary station will send or receive messages. All messages are sent by or to the host. The advantage of point-to-multipoint configuration is that only one port of the host is used for communicating with several stations.

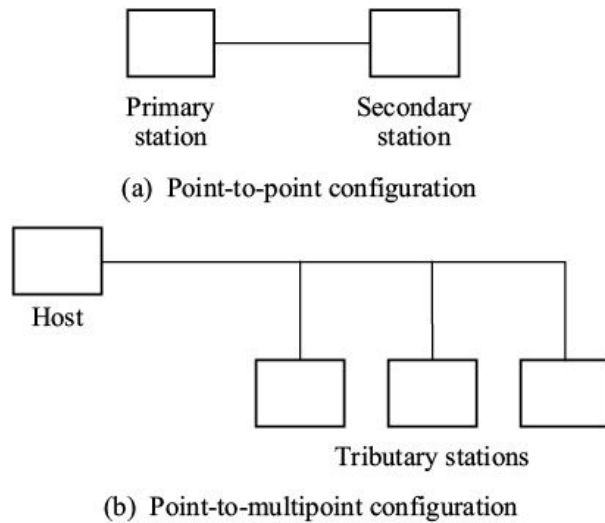


Figure 1.22 Line configurations.

SUMMARY

This chapter is the first step towards understanding the characteristics of information that we deal in data communications. Binary codes are used for representing the data symbols for computer communications. ASCII is the most common code set used worldwide.

Transmission of bits can be asynchronous or synchronous. Asynchronous transmission is byte-by-byte transmission with start/stop bits appended to each byte. In synchronous transmission, data bytes are transmitted as a group in the form of frame that has flags to identify its start and end. Clock is required in synchronous transmission.

Transmission of bits as electrical signals requires their encoding as RZ (Return to Zero) or NRZ (Non Return to Zero) codes. These codes are designed for enabling clock extraction from the received signal. The electrical signals representing data bits can be polar, unipolar or bipolar. Polar or bipolar signals are used for serial transmission as they do not have the DC component.

A communication channel is limited in its information-carrying capacity by its bandwidth and signal to noise ratio. To make best use of this limited capacity of the channel, sophisticated data compression and carrier-modulation methods are used. Modems are the devices that carry out the modulation and demodulation functions. Data compression involves reducing the redundancy from the data representation without sacrificing the information content.

There is need to understand distinction between data transmission and data communication. While transmission refers to transport of information,

communication implies meaningful exchange of information. Asynchronous and synchronous in context of communication refer to non-disciplined and disciplined exchange of messages respectively.

EXERCISES

1. (a) Write ASCII code for the word 'Data'. Assume parity bit is 0.
 (b) Write the bit transmission sequence for the above word.
 (c) Draw the signal waveform if the word 'Data' is transmitted in asynchronous mode using stop bit of one bit duration.
2. Draw the signal waveforms when 00110101 is transmitted using the following codes:
 - (a) NRZ-L
 - (b) NRZ-M
 - (c) NRZ-S
 - (d) Manchester code
 - (e) Biphas-M
 - (f) Biphas-S
 - (g) Differential Manchester.
3. The signals encoded in Exercise 2 are received inverted due to wrong wiring. Write bit sequence generated by the receiver in each case. Which codes are insensitive to inversion of polarity of wires?
4. (a) The circuit shown in Figure E1.23 carries out signal encoding. Analyze its operation for input bit sequence 10110011 and determine the line code it generates. t is one bit delay.
 (b) Draw the decoder implementation for this encoder using an EXOR gate and one bit delay circuit.

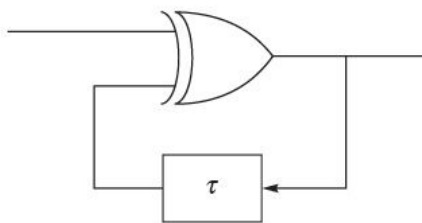


FIGURE E1.23.

5. Write the message conveyed by the following received signal (Figure E1.24) if the line code used is
 - (a) NRZ-M

- (b) NRZ-S
- (c) NRZ-L

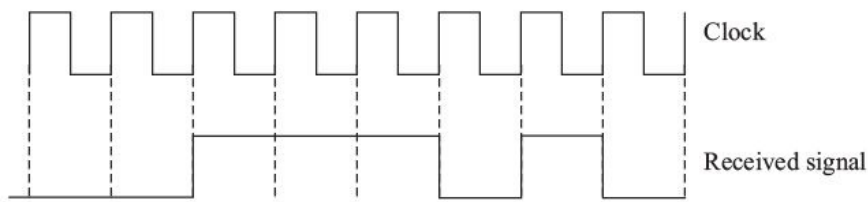


FIGURE E1.24.

6. Write the message conveyed by the following received signal (Figure E1.25) if the line code used is
- (a) Manchester
 - (b) Differential Manchester.

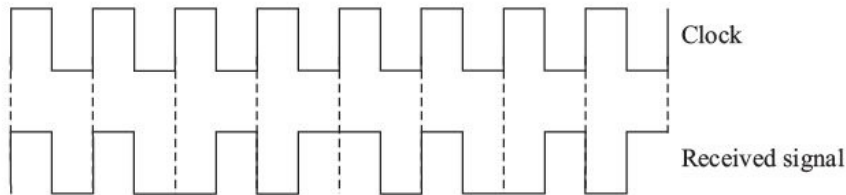


FIGURE E1.25.

7. Write the message conveyed by the following received signal (Figure E1.26) if the line code used is
- (a) Biphas-M
 - (b) Biphas-S.

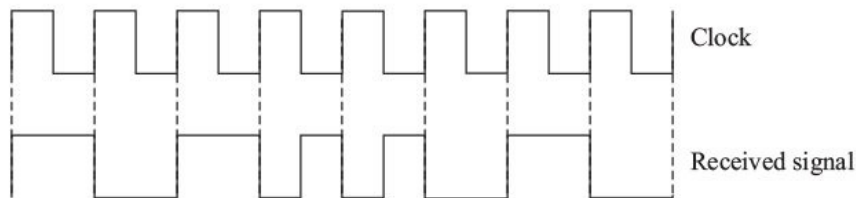


FIGURE E1.26.

8. Decode the following bipolar (+V, 0, -V) sequence if it coded in HDB3.
 +V -V 0 +V 0 0 0 +V 0 -V +V -V 0 0 -V
9. Decode the following bipolar (+V, 0, -V) sequence if it coded in B8ZS.
 0 +V 0 0 0 0 0 0 -V +V 0 0 0 +V -V 0 -V +V 0 0 0 +V -V 0 -V +V 0
10. Each signaling element of a digital signal encodes 8-bit words. If the bit rate is 9600 bps, what is the required bandwidth of noiseless transmission channel for carrying this signal?

11. A channel is required to carry a signal at 32 Mbps. The bandwidth of the channel is 4 MHz. What is the required signal to noise ratio of the channel in order to achieve this capacity?
12. Is the following code (a) a distinct code, (b) uniquely decodable?
13. A:00, B:010, C:100, D:1
14. Is the following code (a) a uniquely decodable, (b) prefix code, (c) instantaneously decodable?
(1, 100000, 00)
15. Decode the message 1000000001 of the code set (a:1, b: 100000, c: 00).
16. (a) A channel has S/N of 20 dB and bandwidth of 3100 Hz. What is the maximum bit rate it can transmit?
(b) What is the number of signaling levels required for achieving the maximum bit rate?
(c) If each level is assigned a binary code of fixed length, calculate the maximum achievable bit rate.
17. (a) Calculate entropy of the following alphabet. Figures within brackets indicate the probability of their occurrence.
A(1/2), B(1/6), C(1/12), D(1/12), E(1/24), F(1/24), G(1/24), H(1/24)
(b) If the above alphabet is encoded using a 3-bit fixed length code, what is the redundancy?
(c) If the above alphabet is encoded as indicated in the brackets, what is the average length of the code?
A(1), B(001), C(010), D(0000), E(0110), F(0001), G(01110), H(01111)
(d) What is the redundancy in the above code?
18. (a) Calculate entropy of the source that generates the following symbols with probabilities indicated in the brackets.
 $a (8/40)$, $b (3/40)$, $c (6/40)$, $d (5/40)$, $e (4/40)$, $f (7/40)$, $g (2/40)$, $h (5/40)$
(b) Generate Huffman code, calculate average code length and redundancy.
(c) Generate Shannon-Fano code, calculate average code length and redundancy.
19. (a) Calculate entropy of the source that generates the following symbols with probability of occurrence as indicated in the brackets.
 $a (20/40)$, $b (12/40)$, $c (4/40)$, $d (2/40)$, $e (1/40)$, $f (1/40)$
(b) Generate Huffman code, calculate average code length and redundancy.
(c) Generate Shannon-Fano code, calculate average code length and redundancy.

2

Transmission Media

In the last chapter, we studied the characteristics of digital signals that carry data. In this chapter, we will examine the characteristics of the transmission media that carry the digital signals. First we develop the concepts of transmission line theory and examine characteristics of metallic and nonmetallic transmission lines, namely, balanced-pair, coaxial pair and optical fibres. We discuss briefly radio media used for communications before proceeding to principles of data transmission over bandwidth limited channels. We then discuss Nyquist criteria for minimum intersymbol interference. Before close of the chapter, we study equalizers required for compensating for the transmission line characteristics and examine transversal equalizer used in modems.

2.1 TRANSMISSION LINE CHARACTERISTICS

Classic theory of transmission line is based on two parallel metallic conductors, placed side by side or coaxially. Characteristics of a transmission line are described in terms of primary and secondary parameters.

2.1.1 Primary Parameters The elementary section of a transmission line of length x can be modelled as shown in Figure 2.1. This model is based on four primary parameters of the transmission line:

- The series resistance per unit length (R) of the two conductors.
- The inductance per unit length (L) of the conductors.
- The capacitance per unit length (C) between the two conductors.

- The leakage conductance per unit length (G), which primarily accounts for the dielectric losses. The insulation losses are generally negligible.

The primary parameters of a transmission line enable us to describe the secondary parameters.

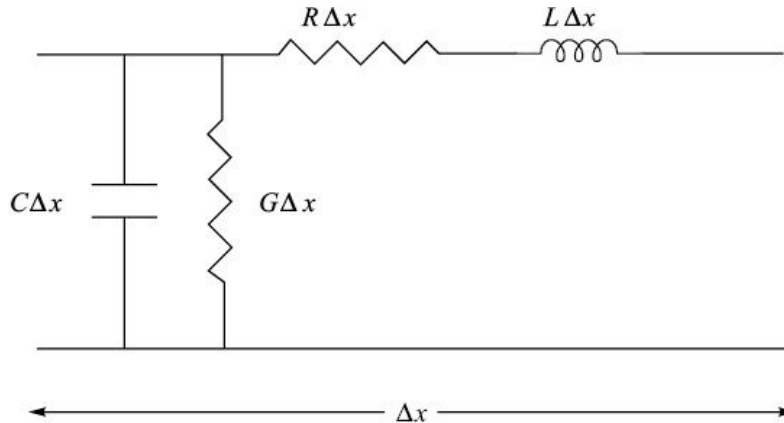


Figure 2.1 Primary parameters of a transmission line.

2.1.2 Secondary Parameters Secondary parameters describe the behaviour of a transmission line and can be easily measured. The two secondary parameters are *characteristic impedance* and *propagation constant*.

Characteristic impedance (Z_C). It is the input impedance of an infinitely long transmission line. It is given by $Z_C = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$.

If a transmission line is terminated in its characteristic impedance, its input impedance is Z_C irrespective of its length.

Propagation constant (g). Propagation constant g determines the attenuation and the phase change in a sinusoidal wave travelling along the transmission line. Propagation constant g is given by $g = a + jb = \sqrt{(R + j\omega L)(G + j\omega C)}$ where a is attenuation of the signal having frequency w over a unit distance, and b is the phase change in the signal when it travels a unit distance. a and b are called attenuation constant and phase constant of the transmission line respectively.

2.1.3 Phase Velocity and Phase Delay To describe the changes in phase relationships of various frequency components of a

signal that travels on a transmission line, we define two more terms, phase velocity and phase delay. If f is the phase change between points x and $x + \Delta x$, then $f = \Delta x / b$. If the same phase change f occurs at point $x + \Delta x$ in time t , then $f = \omega t$. Thus, if a constant phase point is observed along the transmission line, the speed of propagation or the phase velocity is given by $v = \Delta x / t = \omega / b$. The time required to travel unit distance is called phase delay per unit length. It is given by $t = 1/v = b/\omega$.

2.1.4 Frequency Dependence of Secondary Parameters Asymptotic behaviour at low frequencies. At low frequencies, we can assume $\omega L \ll R$ and if $G = 0$, the secondary parameters of the

transmission line are given by $Z_c @ \sqrt{\frac{R}{j\omega C}} = \sqrt{\frac{R}{\omega C}} e^{-j\pi/4}$

$$g = a + jb @ \sqrt{j\omega RC} = \sqrt{\omega RC/2} + j\sqrt{\omega RC/2}$$

In conclusion, when $\omega L \ll R$,

- the characteristic impedance is complex and is inversely proportional to the square root of the frequency; and
- the attenuation and phase constants are proportional to the square root of the frequency.

Asymptotic behaviour at high frequencies. At high frequency we can assume $\omega L \gg R$, and if $G @ 0$, we can write the secondary parameters of the transmission line as $Z_c = \sqrt{L/C}$

$$g = a + jb @ \sqrt{-\omega^2 LC + j\omega RC} = j\omega\sqrt{LC} \left(1 - j\frac{R}{2\omega L} \right)$$

$$a = \frac{R}{2}\sqrt{L/C}$$

$$b = \omega\sqrt{LC}$$

In conclusion, when $\omega L \gg R$,

- the characteristic impedance is real and independent of frequency;
- the attenuation constant is proportional to R (and therefore, if the skin effect is taken into account, a will vary as ω); and

- the phase constant increases linearly with the frequency.

Figure 2.2 summarizes the results of the preceding sections. Note that a is proportional to the square root of the frequency at low and high frequencies due to entirely different reasons.

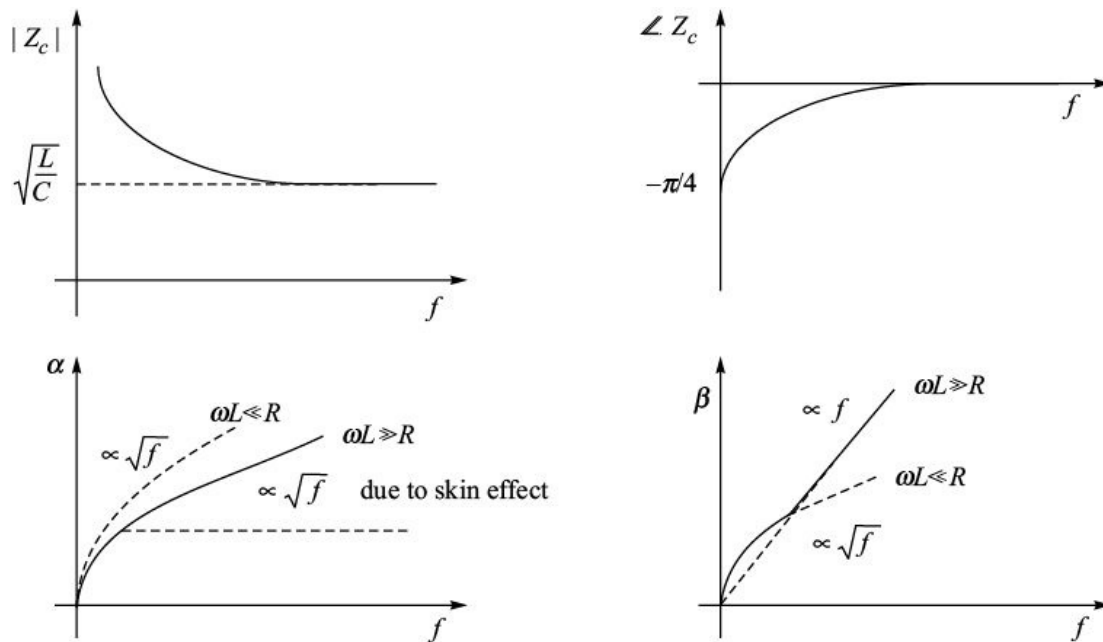


Figure 2.2 Asymptotic behaviour of the secondary parameters.

2.2 LINEAR DISTORTIONS

An ideal transmission system should not distort the signal in any manner. It can, at most, cause the received signal to differ from the transmitted signal in having been multiplied by a constant factor and delayed by a certain time. This implies that for distortionless transmission of a signal that is expressed as a series of sinusoidal signals,

- amplitudes of all its frequency components are multiplied by the same factor; and
- all its frequency components are delayed by the same amount when they are transmitted.

In other words, the attenuation constant a and the phase delay t of the transmission line must be independent of frequency. t being equal to b/w , the

phase constant b should increase linearly with frequency. Figure 2.3 shows the plots of a and b for distortionless transmission. These conditions for distortionless transmission cannot be perfectly satisfied in practice. As a result, the transmitted signal always gets distorted. A transmission line introduces,

- attenuation distortion if its attenuation constant a varies with frequency; and
- phase distortion if the phase constant b is not a linear function of the frequency.

These distortions are called linear distortions because the transmission line presents a linear system in which the principle of superposition remains valid. A sinusoidal signal remains a sinusoid and no new spectral components appear at the other end of the transmission line. Non-linear distortions introduce additional frequency components as we shall see later.

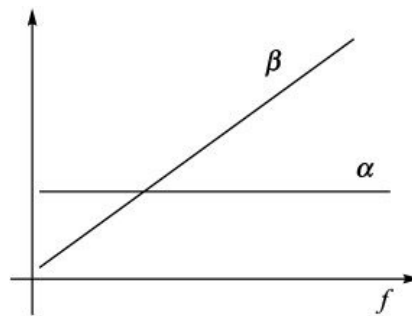


FIGURE 2.3 Attenuation and phase characteristics for distortionless transmission.

2.2.1 Group Delay

The condition for no phase distortion is in fact very stringent and difficult to implement. Considering that most of the transmitted signals are modulated carriers, we can relax the constant phase delay requirement. It can be shown that phase distortion does not occur in the modulating signal if the slope db/dw of phase characteristic is constant. The slope db/dw is called *group delay* and it can be easily measured for any transmission channel.

2.2.2 Frequency Domain Equalizers A transmission line always introduces linear distortions because the attenuation constant a is proportional to the square root of the frequency and the phase constant b is not a linear function of frequency. These distortions are corrected by using equalizers. As the name

suggests, equalizers make up for the transmission characteristics and minimize the attenuation and phase distortions.

Equalizer is usually provided at the receiving end of the transmission line. There are separate equalizers for correcting attenuation distortion and phase distortion. The attenuation equalizer has a loss characteristic which is inverse of the attenuation characteristic of the transmission line so that the net result is a flat amplitude frequency response (Figure 2.4a).

As regards the phase distortion, the transmission line is equalized for group delay. The group delay characteristic is made flat using an equalizer with complementary group delay response (Figure 2.4b).

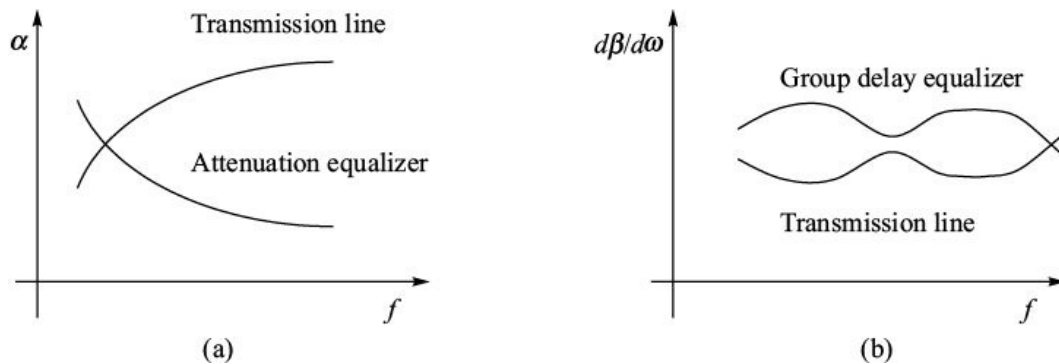


FIGURE 2.4 Equalizer characteristics.

2.3 CHARACTERISTICS OF TRANSMISSION LINE IN TIME DOMAIN

The attenuation $A(\omega) = \alpha l$ and the phase change $B(\omega) = \beta l$ describe the behaviour of the transmission line of length l in the frequency domain, *i.e.* they enable us to calculate the amplitude and phase of the signal obtained at the distant end of a transmission line when a sinusoidal signal of angular frequency ω is applied to it. $A(\omega)$ and $B(\omega)$ together define the transfer function $H(\omega)$ of the line. Inverse Fourier transform of $H(\omega)$ gives the impulse response $h(t)$ of the transmission line. The impulse response $h(t)$ completely describes the behaviour of the transmission line in time domain. Figure 2.5 shows the received signal at the output of the line when a pulse of time duration T is applied at its input. The shape of the received signal depends on the impulse response of the transmission line. Without going into mathematical details, we can quote the result of time domain analysis. A transmission line is said to be short with respect to T when

its length l is such that $\frac{2T}{RCl^2} \gg 10$

When the line is short, the pulse shape is slightly deformed and the pulse duration is retained (Figure 2.5b). On the other hand, for a long transmission line (when the above inequality is reversed) the pulse is considerably deformed and its duration also gets stretched (Figure 2.5c). It may even interfere with the subsequent neighbouring pulses. This type of interference is termed as Intersymbol Interference (ISI). We will come back to intersymbol interference later and discuss how it can be minimized.

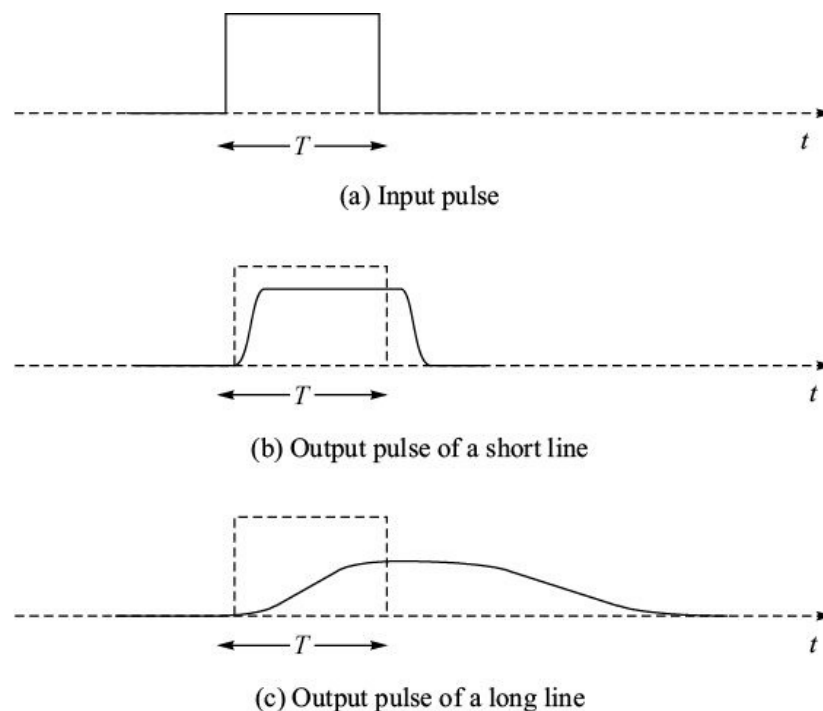


FIGURE 2.5 Pulse distortion in time domain.

2.4 CROSSTALK

When two transmission lines are very close, they interfere with each other and it results in *crosstalk*, *i.e.* signals of one line cross over to the other. Crosstalk occurs due to three types of mutual coupling between the lines:

- Galvanic coupling which is due to a common resistance of the two lines. This phenomenon is noticeable in lines having common return conductor

(Figure 2.6a).

- Capacitive coupling which is due to the capacitance between the conductors of the lines (Figure 2.6b).
- Inductive coupling which is due to the mutual inductance of the transmission lines (Figure 2.6c).

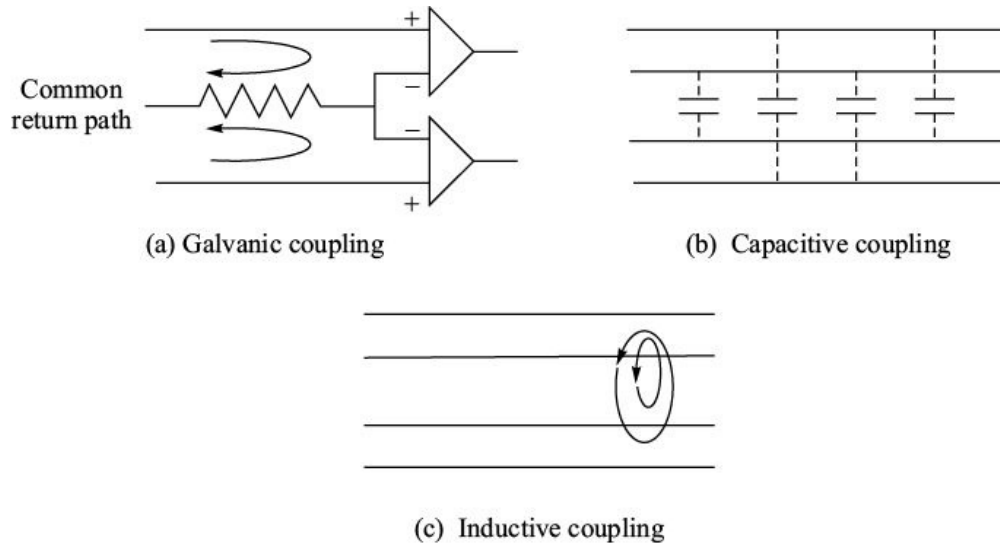


FIGURE 2.6 Types of coupling in transmission lines.

The extent of coupling depends on the geometric configuration of the conductors and the proximity of the transmission lines. The signals coupled to a transmission line due to crosstalk progress towards the far end as well as back to the near end (Figure 2.7). The crosstalk which appears at the near end is called *near-end crosstalk* (NEXT), and the crosstalk which appears at the distant end is called *far-end crosstalk* (FEXT). NEXT is relatively independent of the length of the transmission line as the first few metres of the transmission line are responsible for the bulk of it. On the other hand, the effect of FEXT increases with the length of the transmission line.

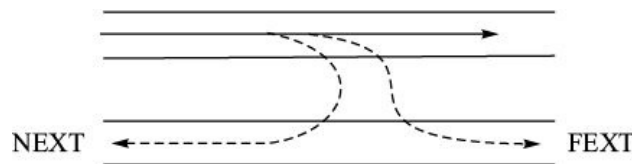


FIGURE 2.7 Near-end crosstalk (NEXT) and far-end crosstalk (FEXT).

2.5 LOGARITHMIC UNITS OF POWER LEVEL MEASUREMENTS

Logarithmic units of power levels simplify network analysis because multiplication is replaced with addition or subtraction. Thus output of an amplifier can be obtained by simple addition of input power level and gain, if both are expressed in logarithmic units. The basic unit is Bel, but usually we express all the power levels in decibels, which is one tenth of a Bel. Decibel is defined as below.

If we have two power levels, P_2 and P_1 , relative value of P_2 with respect to P_1 in decibels (dB) is given by $\text{dB} = 10 \log_{10} (P_2/P_1)$ If P_2 is ten times P_1 , the decibels calculated as above will be 10 dB. Instead of saying P_2 is ten times P_1 , we say P_2 is 10 dB more than P_1 . We can, thus, express gain of an amplifier and loss of an attenuator in decibels.

EXAMPLE 2.1 A signal with a power of 10 mW is transmitted through a coaxial cable. Signal level measured at the other end of the cable is 5 mW. What is the loss of the cable?

Solution

$$P_2 = 5 \text{ mW}, P_1 = 10 \text{ mW}$$

$$\text{Loss} = 10 \log_{10} (5/10) = -3 \text{ dB}$$

Decibel is a relative unit of power level. To express the absolute values of power levels, we base the computation with reference to a standard unit of power, say 1 mW or 1 W. The corresponding decibel units are dBm and dBW, respectively. For example, power levels of

100 mW and 2W can be expressed in dBm and dBW, respectively as under: $100 \text{ mW} = 10 \log_{10} (100 \text{ mW}/1 \text{ mW}) = 10 \cdot 2 = 20 \text{ dBm}$ $2 \text{ W} = 10 \log_{10} (2 \text{ W}/1 \text{ W}) = 10 \cdot 0.3 = 3 \text{ dBW}$

EXAMPLE 2.2 If the input power level is 1 mW, what is the received signal strength in Figure 2.8?

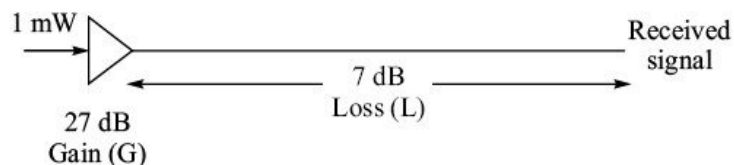


Figure 2.8 Example 2.2.

Solution

Input power level $P_i = 1 \text{ mW} = 10 \log_{10} (1\text{mW}/1\text{mW}) = 0 \text{ dBm}$ $P_o = P_i + G - L = 0 + 27 - 7 = 20 \text{ dBm} = 100 \text{ mW}$.

2.6 METALLIC TRANSMISSION MEDIA

The transmission line concepts developed above are applicable to any type and shape of transmission line. In the industry, we come across two forms of metallic transmission lines: balanced pair, and coaxial pair.

2.6.1 Balanced Pair

A balanced pair is a two-wire transmission line in which the two conductors are identical and have the same capacitance and leakage conductance with respect to the ground. The term 'balanced' implies electrical balance which is intended to reduce galvanic, capacitive, and inductive coupling between the transmission lines. Since both the conductors are identically placed with respect to the ground, coupling from another line or from any other source generates common mode voltages in the conductors. The common mode voltages cancel each other at the receiver (Figure 2.9).

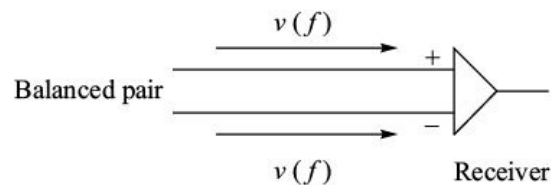


FIGURE 2.9 Common mode voltages in a balanced pair.

The first application of balanced pair was the open-wire balanced transmission line which consisted of two parallel bare conductors supported on poles using insulators. It is now obsolete and has been replaced to a large extent by other transmission media.

2.6.2 Balanced Pair Cables The balanced-pair cable consists of a bunch of pairs of insulated copper wires which are twisted in such a way as to reduce inductive coupling among the pairs. Balanced-pair cables are used in the telecommunication network as junctions for interconnecting telephone exchanges and as local cables to extend the connection from telephone

exchange to customer.

The conductors have diameters ranging from 0.4 to 1.5 mm and the insulating material is paper or polyethylene. The number of pairs in a balanced pair cable can be from 4 to 1200 or even more. The balanced pair used in the telecommunication network works in the frequency range where $\omega L \ll R$. The principal characteristics of the pair in this frequency range are as below:

- a and b are proportional to the square root of the frequency and therefore, attenuation and phase distortions are present.
- The characteristic impedance is complex.

Figure 2.10 shows variation in the characteristic impedance and attenuation parameter with frequency. Note that the characteristic impedance varies from 600 ohms at 1 kHz to 150 ohms at frequencies higher than 10 kHz. The balanced pairs are terminated accordingly depending on frequency of the signal.

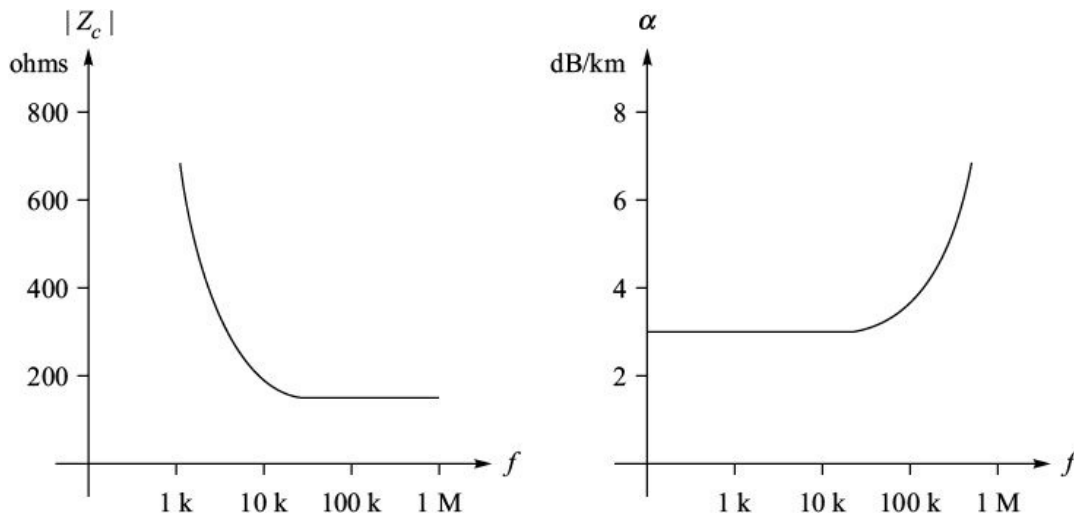


Figure 2.10 Transmission characteristics of the 0.6 mm diameter balanced pair.

2.6.3 Loading of Balanced Pairs

The characteristics of a transmission line are nearly ideal when $\omega L \gg R$, and the skin effect is negligible. We can approach these conditions at low frequencies by artificially increasing the inductance per unit length (L) of the transmission line. When the inductance is increased, the attenuation constant a diminishes and becomes independent of the frequency, and the phase constant b increases and becomes proportional to the frequency as shown below.

$$g = a + jb = \sqrt{(R + j\omega L)(G + j\omega C)}$$

Since $G \ll \omega C$, $g = a + jb = \sqrt{j\omega L \left(1 + \frac{R}{j\omega L}\right) j\omega C} = \frac{R}{2} \left(\frac{C}{L}\right)^{1/2} + j\omega(LC)^{1/2}$

To increase L , lumped inductors called *loading coils* are added at regular intervals in the transmission line (Figure 2.11a). The transmission line behaves like a low-pass filter having reduced constant attenuation in the pass band (Figure 2.11b). The attenuation increases rapidly beyond the cut-off frequency f_c

given by $f_c = \frac{1}{\pi\sqrt{L_p Cd}}$

where L_p is the inductance of the loading coil and d is the separation between adjacent coils. In the telephone network, we generally choose $L_p = 88.5$ mH and $d = 1830$ m. Assuming negligible inductance of the line itself, the cut-off frequency calculated from the above relation comes to 4228 Hz.

A loaded pair (0.6 mm) gives loss of 0.4 dB/km and unloaded pair (0.6 mm) gives loss of 0.8 dB/km. But the loaded pair offers limited bandwidth suitable only for voice.

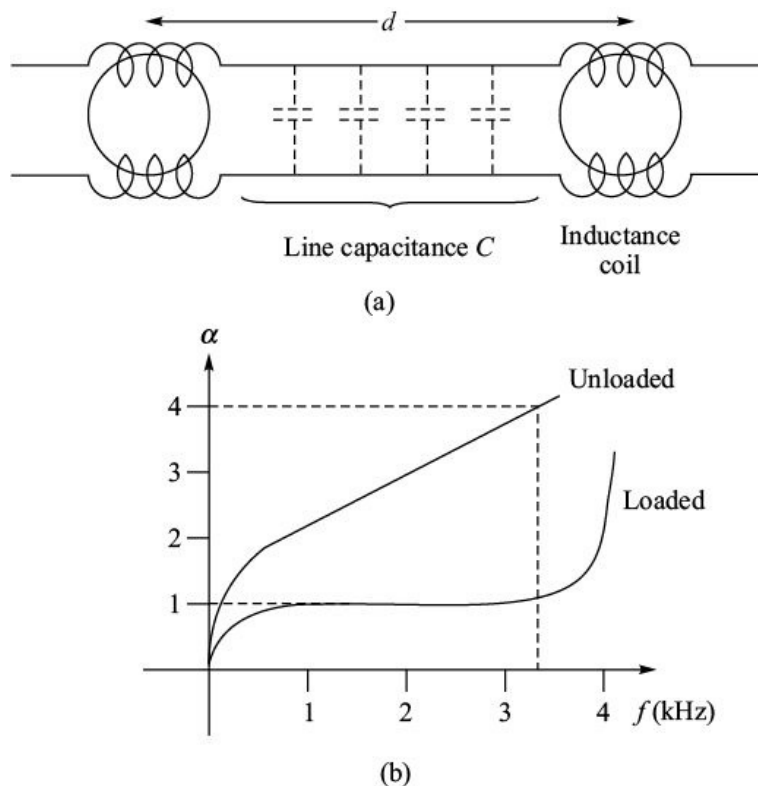


Figure 2.11 Attenuation characteristics of a loaded balanced pair.

2.6.4 Balanced Pair for Data Networks

Balanced pairs are extensively used in the data networks for interconnecting

computers. Two types of balanced pairs are used:

- Unshielded twisted pair (UTP);
- Shielded twisted pair (STP).

Shielded twisted pair carries a metallic braid on each pair to shield against noise and crosstalk. The Electronics Industries Association (EIA) has developed standards for UTP cables. Five categories of UTP cables (CAT 1 to CAT 5) have been standardized. The first two categories are suitable for voice transmission and low speed data (less than 4 Mbps). Table 2.1 lists the attenuation characteristics of CAT 3, CAT 4 and CAT 5 UTP cables.

CAT 3. It is suitable for data networks operating at speed up to 10 Mbps.

CAT 4. It is suitable for data rates up to 16 Mbps with segment length up to 100 metres.

CAT 5. It is specified to work up to 100 Mbps.

CAT 3 and CAT 5 cables have received the most attention for local area networks (LAN). CAT 3 is found in abundance as it was used for voice communications. CAT 5 is found in all new installations. CAT 4 is generally applicable for token passing ring LAN. CAT 5E (CAT 5 Enhanced) is a later version of CAT 5 that offers better transmission characteristics. Standards are under development for CAT 6 and CAT 7 cables that will operate at 200 Mbps and 700 Mbps respectively.

TABLE 2.1 Attenuation in dB of UTP Cables at 20°C over 305 m

Frequency (MHz)	CAT 3	CAT 4	CAT 5
0.064	2.8	2.3	2.2
1	7.8	6.5	6.3
4	17	13	13
8	26	19	18
10	30	22	20

16	40	27	25
20		31	28
25			32

2.6.5 Coaxial Cable

The coaxial pair is composed of two concentric conductors separated by dielectric discs or continuous material (Figure 2.12). The external conductor can be solid copper sheet or a metallic braid. A coaxial cable may contain one or more coaxial pairs.

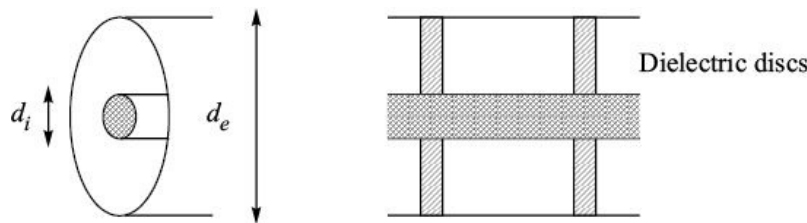


FIGURE 2.12 Coaxial pair.

The secondary parameters of the coaxial pair can be expressed in terms of external diameter (d_i) of central conductor and the internal diameter (d_e) of the external conductor. The principal characteristics of the coaxial pair are summarized below. It is assumed that $wL \gg R$, which is true in the useful frequency range of coaxial pair.

- The characteristic impedance depends on the ratio of diameters d_e/d_i .
- The attenuation constant is inversely proportional to the diameter d_e . For a given d_e , minimum value of a is obtained when $d_e/d_i = 3.6$.
- The attenuation constant is inversely proportional to the square root of the frequency due to skin effect. Thus, there is attenuation distortion.
- The phase constant is a linear function of frequency at frequencies $f > 100$ kHz. Thus, there is no phase distortion.

ITU-T recommendations for the coaxial pairs are given in Table 2.2. The attenuation characteristics are shown in Figure 2.13. Being a logarithmic plot, the slope of the characteristic is 1/2.



TABLE 2.2 ITU-T Recommendations for the Coaxial Pairs

ITU-T Recommendations			
	G.623	G.622	G.621
	2.6 mm	1.2 mm	0.75 mm
	9.5 mm	4.4 mm	2.9 mm
$d_i d_e d_e/d_i Z_c$	3.65	3.67	4.14
	75 1	75 1	75 1

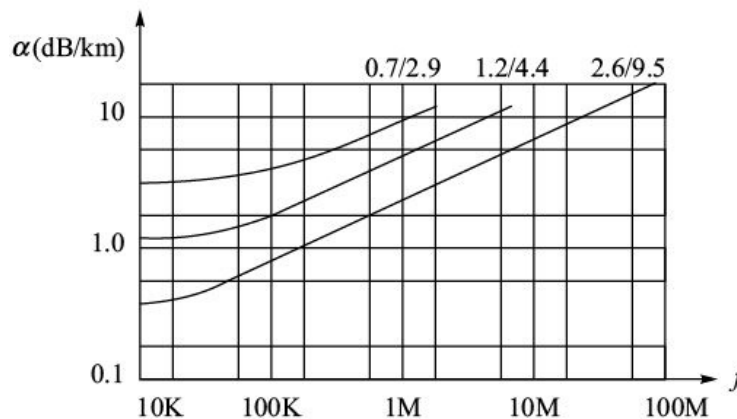


FIGURE 2.13 Attenuation characteristics of coaxial pair.

Coaxial cables are used as the long distance transmission medium in telephone networks. Coaxial cables also find application in cable television networks and local area networks for computer communications. They are not used at frequencies lower than 60 kHz due to their degraded phase and crosstalk properties at low frequencies.

2.7 OPTICAL FIBRE

An optical fibre consists of an inner glass core surrounded by a cladding also of glass but having a lower refractive index. Transmission of digital signals is in the form of intensity-modulated light signal which is trapped in the core. Light is launched into the fibre core using a light source (LED or Laser) and is detected at the other end using a photodetector (Figure 2.14). Early optical systems used

LEDs having wavelength of 870 nm (nanometer) but later 1300 and 1550 nm wavelengths were found more suitable and most of the optical fibre systems use these wavelengths at present.

2.7.1 Multimode Fibres Light propagation in the core is based on the phenomenon of total internal reflection which takes place at the core-cladding interface. The refractive index of the cladding being less than that of the core, an oblique light ray in the core is reflected back if it strikes the interface at an angle greater than the critical angle which is determined by the refractive indices of the core

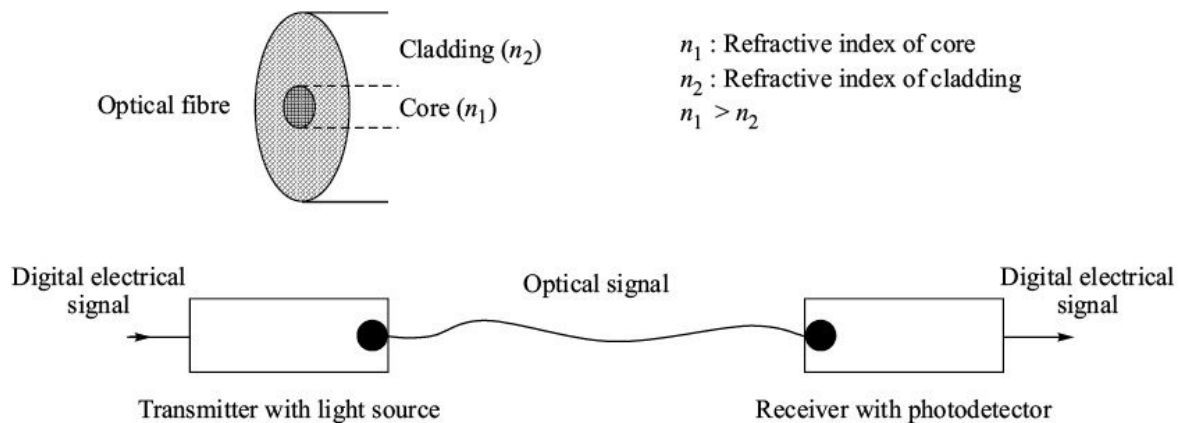


Figure 2.14 Optical fibre as transmission medium.

and cladding (Figure 2.15). It can be shown that in a cylindrical guided medium like optical fibre, a finite number of modes of light propagation can be sustained. Mode refers to the light path which a ray traces depending on its angle of incidence at the interface. The number of the modes depends on the diameter of the core and can be reduced by reducing the core diameter. A multimode step index fibre, as the name suggests, supports several modes and has step change in the refractive index profile at the core-cladding interface. Typical core and cladding diameters of a multimode fibre are 50 mm and 125 mm, respectively.

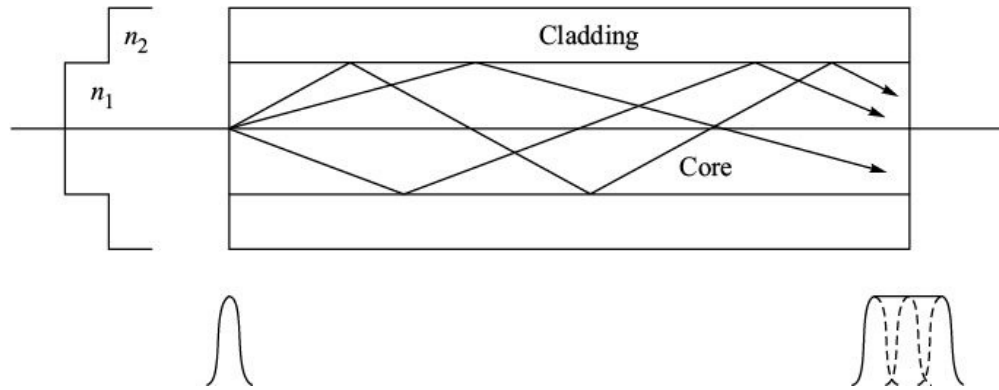


Figure 2.15 Light propagation in step index multimode fibre.

2.7.2 Modal Dispersion When a pulse of light is incident on the fibre end, several modes of light propagation are generated in the core. These modes propagate through the fibre over paths of different lengths. When they arrive at the other end of the fibre, they are staggered in time and result in stretching of the transmitted pulse (Figure 2.14). This phenomenon is called *modal dispersion* and it increases with distance. Dispersion limits the bit rate a length of fibre section can support. Modal dispersion t_m is given by $t_m = \frac{n_1 l}{c n_2} (n_1 - n_2)$ where n_1 and n_2 are the refractive indices of the core and cladding respectively, c is speed of light in vacuum and l is the length of fibre section. Typical value of modal dispersion for multimode fibre is 50 ns/km.

2.7.3 Monomode Fibre

If the diameter of the fibre core is reduced to such an extent that it can sustain only one mode, modal dispersion can be eliminated. Such a fibre is called monomode or single mode fibre (Figure 2.16). Its core diameter is of the order of a few microns. Due to its low dispersion, monomode fibre can support very high bit rates.

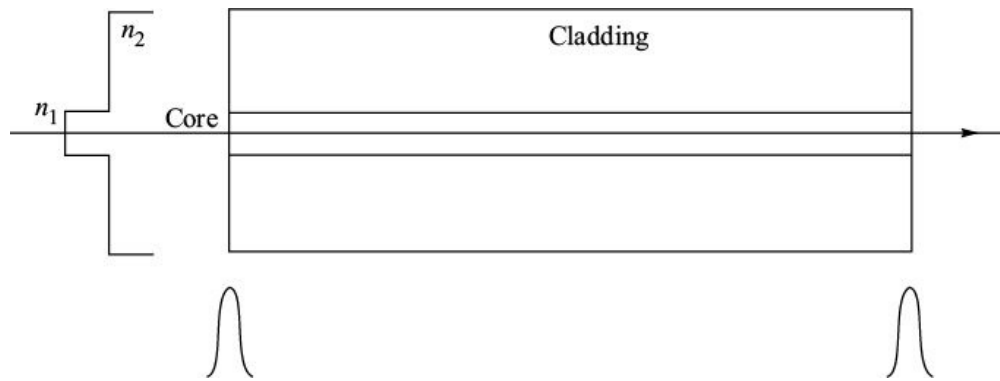


Figure 2.16 Light propagation in monomode fibre.

2.7.4 Graded Index Fibres Multimode and monomode fibres discussed above are called step index fibres as there is step variation of the refractive index profile. There is another type of fibre called graded index fibre in which the refractive index profile of the core approximates a parabolic shape. It can be shown that the modes take different curved paths in the core of the graded index fibre, and periodically converge to common points at the same instant of time. Thus the modes emerge at the other end of fibre almost simultaneously (Figure 2.17). Modal dispersion is therefore, significantly reduced.

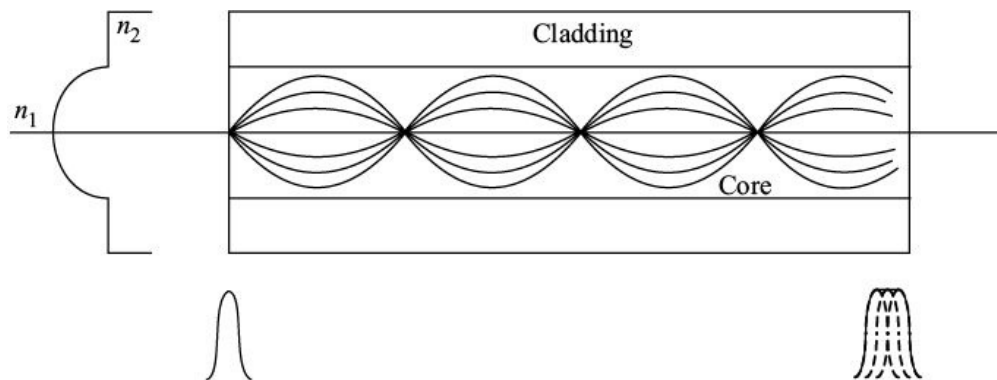


Figure 2.17 Light propagation in graded index fibre.

2.7.5 Chromatic Dispersion

Having reduced the modal dispersion to a very low value in the graded index fibres and having eliminated it in the monomode fibres, further increase in bit rate is limited by another type of dispersion called *chromatic dispersion*. It is due

to different speeds of propagation of different wavelengths emitted by the source. These wavelengths arrive at the other end of the fibre at different times and cause chromatic dispersion.

The light source in the transmitter is not a monochromatic optical source and its spectral width (difference between the highest and the lowest wavelengths emitted) is of the order of 50 nm in case LED and 1 nm in case of solid state laser. Chromatic dispersion depends on the spectral width of the source, composition of the fibre material, and length of the fibre section. It is given by t_c

$$= Ml, M = \frac{d^2n}{d\lambda^2}$$

where l is the spectral width of the source, l is the length of fibre section and n is the refractive index. For silica glass and step-index profile, it turns out that at approximately 1300 nm, M becomes zero. Therefore chromatic dispersion is extremely small, ~ 10 ps/km, at this wavelength. Solid state lasers at 1300 nm along with monomode fibres are used for transmission at very high bit rates.

2.7.6 Total Dispersion Total dispersion in a fibre is obtained by adding the squares of modal and chromatic dispersion.

$$t = \sqrt{\tau_m^2 + \tau_c^2}$$

The optical source and detector also have rise time associated with them. If their rise times are t_s and t_d respectively, system rise time t_r is given by $t_r =$

$$\sqrt{\tau_s^2 + \tau_d^2 + \tau^2}$$

If the bit rate is R , the maximum design value of t_r is kept at $0.7/R$ for NRZ signals and

$0.35/R$ for RZ signals.

2.7.7 Fibre Attenuation Intrinsic attenuation of the fibre is due to absorption and Rayleigh scattering of light. While absorption losses due to impurities can be controlled and minimized, Rayleigh scattering cannot be overcome. In the early seventies, the fibre loss was minimum around $\lambda = 820$ nm and therefore, early optical fibre systems operated at this wavelength. The loss was of the order of 5 dB/km. As the fibre manufacturing technology developed, the fibre loss was brought down to about 0.5 dB/km at 1300 nm. The next wavelength to be used

was 1550 nm at which the loss is less than 0.2 dB/km. Dispersion-shifted fibres which exhibit minimum chromatic dispersion at this wavelength have already been developed. Most of the present day systems operate at 1300 or 1550 nm wavelengths.

2.7.8 Advantages of Optical Fibre Some advantages of optical fibres as a transmission medium are listed below:

- Optical fibres offer very wide bandwidth for transmission of signals. Bit rates in the range of gigabits per second are feasible with the present technology.
- Optical signals are not affected by electromagnetic interference.
- Optical fibres have very low attenuation (0.2 dB/km).
- Copper cables need repeaters¹ at about every four kilometers while this distance is of about 50–70 km in optical fibres.
- Optical transmission is secure as the signals cannot be tapped.
- Optical fibre cables are very light in weight, very small in size, and are easily laid.
- Glass being an insulator, optical fibres are safe when laid along a high tension power line.
- Optical fibres are made of silica which is abundantly available as a natural resource. Copper, on the other hand, is rare metal.
- Optical fibre has very low sensitivity to temperature and environment.

2.8 RADIO MEDIA

The radio systems provide wireless communication links and are useful where distance and terrain render cable media uneconomical. In all the radio systems, the signal to be transmitted modulates a radio frequency (RF) carrier. The transmitting antenna sends the modulated carrier as electromagnetic wave which propagates through free space. The receiving antenna picks up the radio frequency signal, and demodulates the received signal. Radio systems are inherently insecure and prone to interference. Frequency of the carrier determines its propagation characteristics, capacity and immunity to noise.

Radio transmission media found limited applications in data communications till recently. Wireless local area networks have become very popular in the last few years.

2.8.1 Electromagnetic Spectrum The part of electromagnetic spectrum that is used for radio communications extends from 30 kHz to 300 GHz. It is subdivided into number of bands as shown in Table 2.3. Frequencies in the band 1 GHz to 300 GHz are referred to as microwave frequencies.

HF band. In HF band, the radio frequency propagation consists of two waves, sky wave and ground wave. The sky wave bounces back from ionosphere and the ground wave propagates along surface of the earth. Ionospheric transmission of HF wave is used for Amateur Radio. It is highly undependable. Ground wave HF communication is used for Medium Wave radio broadcast.

TABLE 2.3 Electromagnetic Frequency Bands

Band	Frequency	Wavelength
	300 kHz–3 MHz	
	3 MHz–30 MHz	1 km–100 m
	30 MHz–300 MHz	100 m–1 m
Medium frequency (MF)	300 MHz–3 GHz	10 m–1 m
High frequency (HF)	3 GHz–30 GHz	1 m–10 cm
Very high frequency (VHF)	30 GHz–300 GHz	10 cm–1 cm
Ultra high frequency (UHF)		1 cm–1 mm
Super high frequency (SHF)		
Extremely high frequency (EHF)		

VHF and UHF bands. VHF and UHF radio systems are line-of-sight systems. These frequency bands are used for FM and TV broadcasts, and for low capacity telecommunication systems.

SHF band. The SHF band systems are called microwave systems in practice. Microwave systems are also line-of sight systems. There are two categories of microwave systems, terrestrial systems and satellite systems. The terrestrial microwave systems need repeaters at spacing of about 50 kilometers due to curvature of earth. The satellite microwave systems provide extreme flexibility in terms of channel capacity and geographic location. Satellite acts as a repeater in the space and covers wide geographic area.

Satellite communication systems have one disadvantage. The propagation time of radio wave from one earth station to another through a geo-synchronous² satellite is about 250 milliseconds. Such large propagation delay causes distinctly audible echo in voice communications. In satellite based data communications, the acknowledgements for the sent data packets are received only after 500 milliseconds (onward and return propagation time).

2.9 BASEBAND TRANSMISSION OF DATA SIGNALS

Considering that a random binary signal has a frequency spectrum which extends up to infinity, it would appear that we require a transmission channel which meets the conditions for distortionless transmission over the entire frequency band of the baseband signal. A band limited transmission channel would always distort the signals and result in intersymbol interference. However, a closer examination of the data signal receiver, which we discussed in the last chapter, reveals that information is extracted from the received signal at the sampling instants. It is necessary, therefore, that the intersymbol interference be zero only at the sampling instants. We need not really bother about the intersymbol interference at any other instant. Nyquist showed that a band limited channel can achieve zero intersymbol interference at the sampling instants.

2.9.1 First Nyquist Criterion Let us assume that a pulse is located at $t = 0$ (Figure 2.18). If T is the duration between adjacent binary symbols, other symbols would be located at mT , $m = 1, 2, 3, \dots$ and the sampling

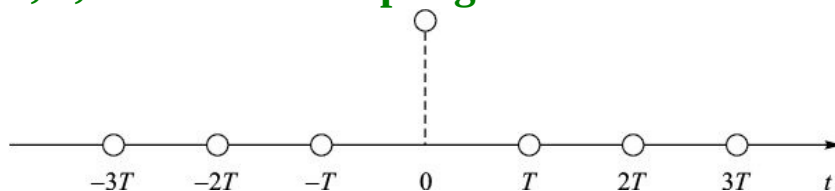


Figure 2.18 Instants of zero intersymbol interference for the first Nyquist criterion.

for them would be done at these instants. We would like to ensure that the pulse at $t = 0$ does not cause any intersymbol interference at these locations. In other words, the received signal corresponding to the pulse at $t = 0$ must pass through encircled points marked in the figure.

One such signal which satisfies this requirement is $\sin(\pi t/T)/(\pi t/T)$ as shown in Figure 2.19a. Figure 2.19b shows the Fourier transform $H(f)$ of the signal. Thus, if the input is an impulse and the transmission channel has the frequency response $H(f)$ as shown in Figure 2.19b, the impulse will generate a response shown in Figure 2.19a and there will not be any intersymbol interference. Figure 2.19c shows the channel response for a data signal in which the data symbols 1 and 0 are represented respectively as impulse and no impulse. Being a linear system, the principle superposition is applicable. The overall response is the sum of individual responses. Thus, it is sufficient to have a transmission channel having ideal characteristics of a low-pass filter with the cut-off frequency equal to half the bit rate Figure 2.19b. We do not require an ideal channel of infinite bandwidth as inferred earlier. This is called the first Nyquist criterion for zero intersymbol interference.

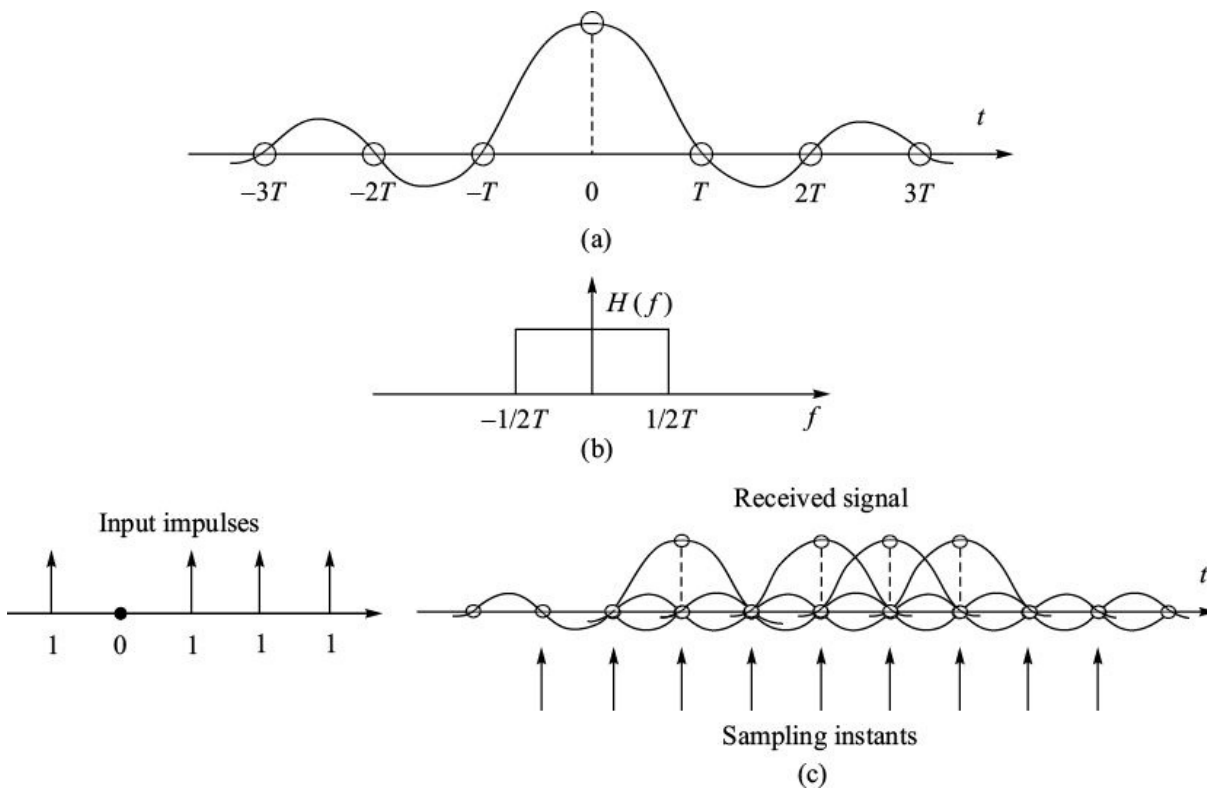


FIGURE 2.19 $\sin(\pi t/T)/(\pi t/T)$ channel response based on first Nyquist criterion.

There are several limitations in realizing the transmission channel defined by the first Nyquist criterion:

- Ideal low-pass filter characteristics of the channel are non-realizable.
- There is significant intersymbol interference around the sampling instants.

Even small errors in the sampling instants result in large amount of intersymbol interference.

2.9.2 Second Nyquist Criterion

We had assumed $\sin(\pi t/T)/(\pi t/T)$ response of the transmission channel to arrive at the first Nyquist criterion. The second Nyquist criterion gives realizable solutions for transmission channel characteristics. We arrive at the second criterion by forcing the channel response to pass through some additional points as shown in Figure 2.20. The channel response curve which passes through the encircled points and its Fourier transform are shown in Figure 2.21. This response curve is called *raised cosine response*.

$$h(t) = \frac{\sin(\pi t/T)}{\pi t/T} \frac{\cos(\pi t/T)}{1 - (2t/T)^2}$$

$$H(f) = T \cos^2(\pi f T/2) \text{ for } |f| \leq 1/2T$$

$$= 0 \text{ for } |f| > 1/2T$$

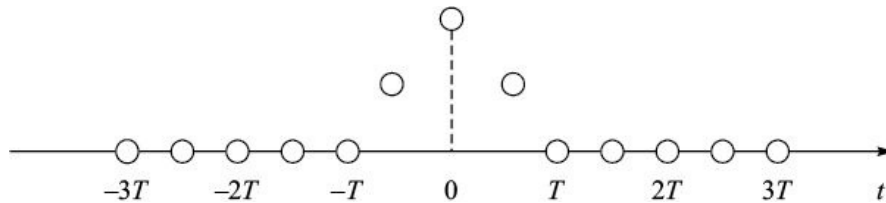


Figure 2.20 Instants of zero intersymbol interference for second Nyquist criterion.

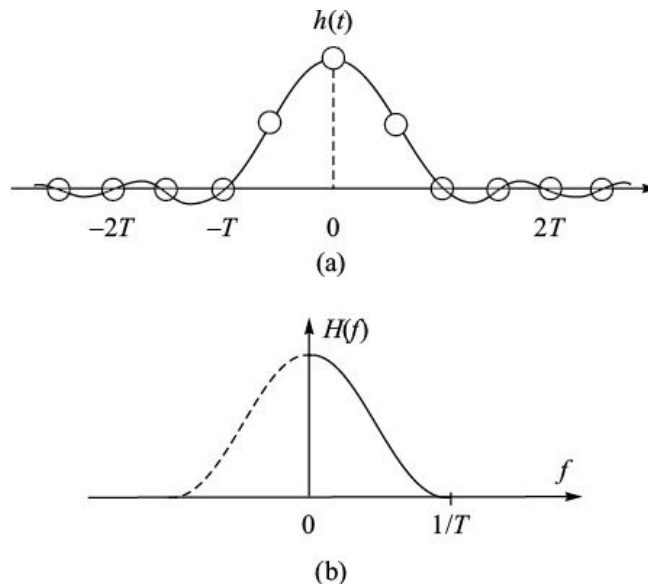


FIGURE 2.21 Raised cosine impulse response and its Fourier transform.

The above channel characteristics define the second Nyquist criterion for no intersymbol interference at the sampling instants. Due to the second power of time in the denominator, intersymbol interference approaches zero much more rapidly than the $\sin(pt/T)/(pt/T)$ response discussed earlier. Therefore, sensitivity to errors in the sampling instants does not severely degrade the performance. But all this is achieved at the expense of doubling the required frequency band ($1/T$ instead of $1/2T$). Although such response curve is not strictly realizable, it can be closely approximated.

EXAMPLE 2.3 A 30 channel PCM system has a bit rate of 2048 kbps. What is the frequency band requirement for the PCM signal as per the first and second Nyquist criteria?

Solution The first Nyquist criterion gives $f = 1/2T = R/2$, where R is the bit rate. Hence, $f = 2048/2 = 1024$ kHz The second Nyquist criterion gives $f = 1/T = R$, where R is the bit rate. Thus, $f = 2048$ kHz Raised cosine and $\sin(pt/T)/(pt/T)$ are the two extreme cases of responses for zero intersymbol interference at the sampling instants. It is possible to define more channel responses that have no intersymbol interference at the sampling instants and have frequency band greater than $1/2T$ and less than $1/T$. Figure 2.22 shows several such channel characteristics. The necessary condition for zero intersymbol interference is that the channel frequency response should have an odd symmetry about $f = 1/2T$. In general, we can say that a binary signal having bit rate $R = 1/T$ would occupy a frequency band of $B = (1+r)R/2$ Hz, where r is the roll-off factor whose value varies from 0 to 1.

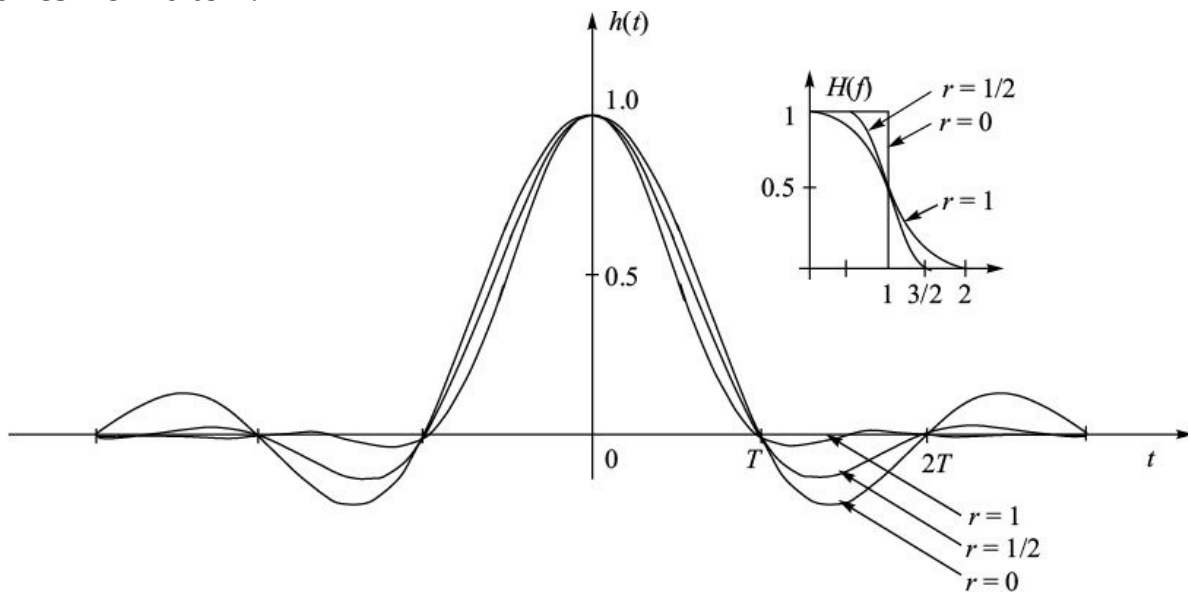


FIGURE 2.22 Generalized raised cosine characteristics.

EXAMPLE 2.4 If the bit rate is 9600 bps and raised cosine pulse spectrum with roll-off factor equal to 0.8 is used, determine the frequency band occupied by the signal.

Solution

$$B = (1 + r) R/2 = (1 + 0.8) 9600/2 = 1.8 \cdot 4800 = 8640 \text{ Hz.}$$

2.9.3 Channel Characteristic for Finite Duration Pulses The raised cosine response is obtained when an impulse is applied to the channel. In practice, we transmit pulses of finite duration T . Therefore, we need to modify the channel characteristic to ensure that we get the same raised cosine response when a finite duration pulse is applied to it.

The Fourier transform of a pulse of duration T is given by $X(f) = T \left[\frac{\sin(\pi f T)}{\pi f T} \right]$

The Fourier transform of the desired output $Y(f)$ is given by $Y(f) = T \cos^2(\pi f T/2)$ for $|f| \leq 1/T$

$$= 0 \text{ for } |f| > 1/T$$

If $H(f)$ is the Fourier transform of the modified channel characteristic, then $H(f)$

$$= \frac{Y(f)}{X(f)} = \frac{\cos^2(\pi f T/2)}{\left[\frac{\sin(\pi f T)}{\pi f T} \right]} \text{ for } |f| \leq 1/T$$

$$= 0 \text{ for } |f| > 1/T$$

Therefore, the raised cosine characteristic of the channel needs to be modified by introducing an additional filter having $(\pi f T)/\sin(\pi f T)$ frequency response characteristic.

2.10 EQUALIZATION

The transmission lines or the channels provided by the telephone network do not meet the Nyquist criterion for zero intersymbol interference. It is necessary to provide an equalizer in the receiver so that overall response characteristic from the data transmitter to the receiver meets the Nyquist criterion (Figure 2.23).

Usually, the maximum attenuation and group delay distortions in the telecommunication channels are guaranteed by providing fixed frequency domain equalizers in the network. The raised cosine equalizers are provided in the interconnecting devices that connect the telecommunication channel to the data terminal equipment. Modem is one such intermediary device and it will be discussed at length in Chapter 4.

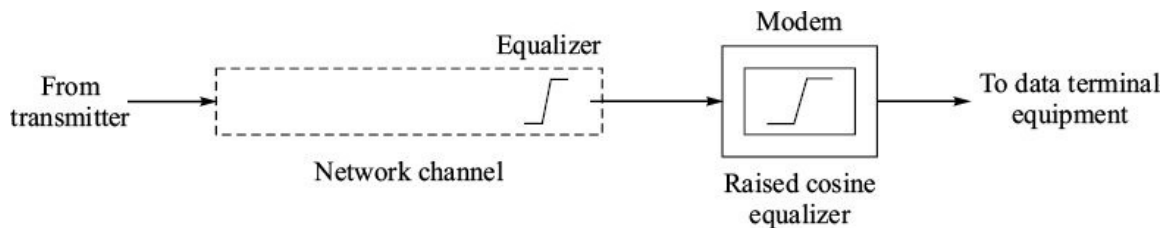


FIGURE 2.23 Equalizers in transmission systems.

The equalizers may be fixed or variable. *Fixed equalizers* are used when the transmission channel characteristics are always within the design limits, while variable equalizers are required when the transmission channel characteristics change with time. For example, if the connection between the transmitter and receiver is established through a telephone exchange, characteristics of the transmission channel change every time a new connection is established.

The *variable equalizers* are of two types, manual and adaptive. *Manual equalizers* are provided with controls to adjust the equalizer for the desired channel response. Most designs use some method of indicating the amount of intersymbol interference which is minimized by adjusting the controls. *Adaptive equalizers*, on the other hand, have the capability of automatic equalizer adjustment for minimum intersymbol interference. Adaptive equalizers are based on the transversal filter equalizer which is described next.

2.10.1 Transversal Filter Equalizer *Transversal filter equalizer* consists of a tapped delay line having $2N + 1$ taps. The delay between successive taps is equal to the pulse sampling interval T (Figure 2.24). Each tap is connected through a variable gain device to a summing amplifier. By adjusting the gain of these devices, it is possible to force intersymbol interference to zero at the sampling instants.

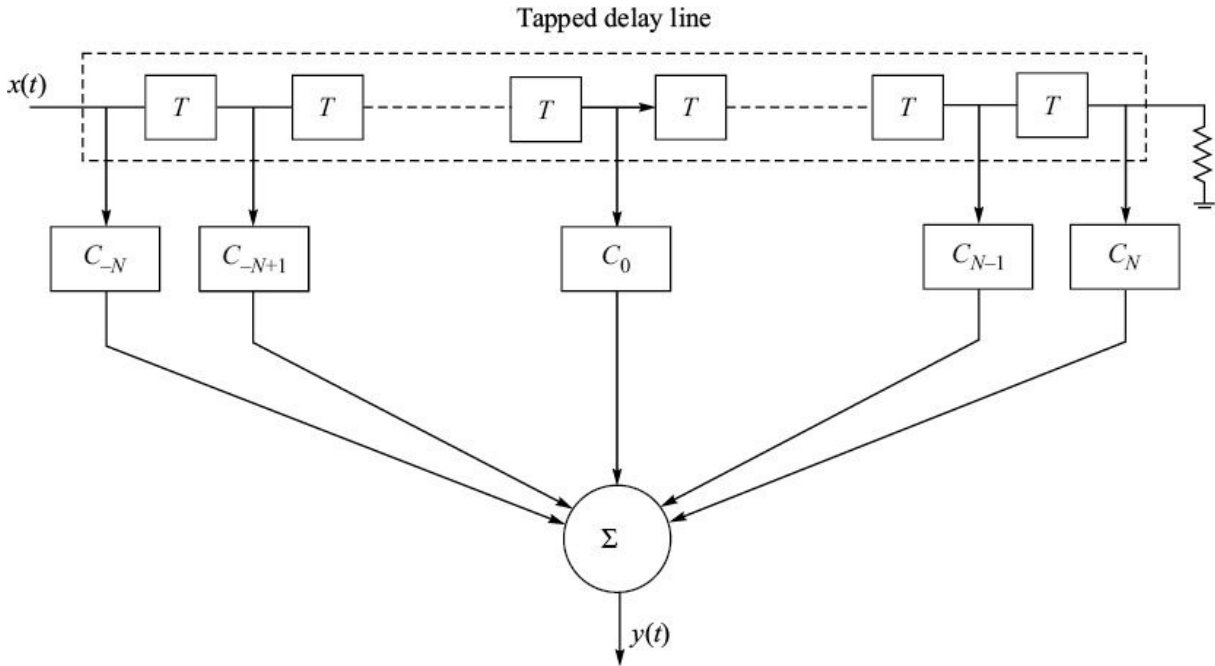


FIGURE 2.24 Transversal filter equalizer.

If the input to the equalizer is $x(t)$, the output $y(t)$ can be written as $y(t) = \sum_{i=-N}^N C_i x [t - (i + N)T]$

where C_i is the gain associated with the i th tap. Since we are interested in the intersymbol interference at the sampling instants only, let us put $t = (k + N)T$, where $k = 0, 1, \dots$ in the above equation.

$$y(k) = \sum_{i=-N}^N C_i x[(k - i)T]$$

where $x(i)$, $-2N \leq i \leq 2N$ are the values of received signal at the sampling instants. Our objective is to adjust the values of the coefficients C_i , so that the intersymbol interference at the sampling instants become zero in the output. In other words, $y(k) = 1$ for $k = 0$

$$= 0 \text{ for } k = 1, 2, 3, \dots, N$$

The quantity $y(k)$, for $k \neq 0$ represents intersymbol interference. Note that we have specified the output at $2N + 1$ points only assuming that at the other points beyond $k = N$, the residual intersymbol interference will be insignificant.

Substituting these values of the output, we get

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} =$$

$$\begin{bmatrix} x(0) & x(-1) & \dots & x(-2N) \\ x(1) & x(0) & \dots & x(-2N+1) \\ \vdots & \vdots & & \vdots \\ x(N-1) & x(N-2) & \dots & x(-N-1) \\ x(N) & x(N-1) & \dots & x(-N) \\ x(N+1) & x(N) & \dots & x(-N+1) \\ \vdots & \vdots & & \vdots \\ x(2N-1) & x(2N-2) & \dots & x(-1) \\ x(2N) & x(2N-1) & \dots & x(0) \end{bmatrix} \begin{bmatrix} C_{-N} \\ C_{-N+1} \\ \vdots \\ C_{-1} \\ C_0 \\ C_1 \\ \vdots \\ C_{N-1} \\ C_N \end{bmatrix}$$

There are $2N + 1$ variables and $2N + 1$ simultaneous equations. We can compute the required values of the coefficients from these equations. Once we set the gain coefficients to the values as computed above, the output $y(t)$ will not have any intersymbol interference.

EXAMPLE 2.5 Determine the gain coefficients of a three-tap equalizer which will reduce the intersymbol interference of the following received signal (Figure 2.25). Draw the equalized signal.

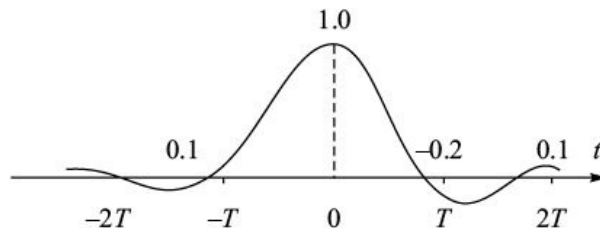


Figure 2.25 Received signal of Example 2.5.

Solution With the three-tap equalizer we can produce one zero crossing on either side of

$t = 0$ in the equalized pulse. The tap gains are given by $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} =$

$$\begin{bmatrix} 1.0 & 0.1 & 0 \\ -0.2 & 1.0 & 0.1 \\ 0.1 & -0.2 & 1.0 \end{bmatrix} \begin{bmatrix} C_{-1} \\ C_0 \\ C_1 \end{bmatrix}$$

Solving for C_{-1} , C_0 and C_1 , we get $\begin{bmatrix} C_{-1} \\ C_0 \\ C_1 \end{bmatrix} = \begin{bmatrix} -0.09606 \\ 0.9606 \\ 0.2017 \end{bmatrix}$

The values of the equalized pulse at the sampling instants can be computed using $y(k) = C_{-1}x(k+1) + C_0x(k) + C_1x(k-1)$.

$$\begin{aligned} y(-3) &= 0.0 \\ y(-2) &= -0.0096 \\ y(-1) &= 0.0 \\ y(0) &= 1.0 \\ y(1) &= 0.0 \\ y(2) &= 0.056 \\ y(3) &= 0.02 \end{aligned}$$

The equalized signal is shown in Figure 2.26.

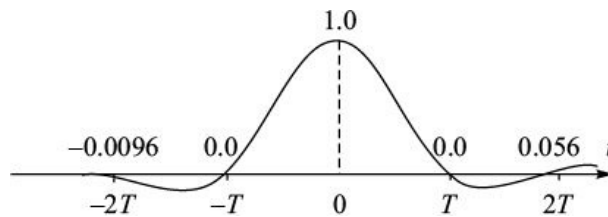


FIGURE 2.26 Equalized signal of Example 2.5.

2.10.2 Adaptive Equalization

Transversal filter equalizer can be configured to adapt to the characteristics of the transmission channel automatically. Adaptation involves the following steps:

1. Prior to data transmission a known test sequence called a *training sequence* is transmitted on the channel.
2. Resulting response sequence $y(k)$ is obtained in the receiver by measuring output of the transversal filter at the sampling instants.
3. An error sequence $y(k)$ is obtained by subtracting the received response sequence from the desired response sequence $d(k)$.

$$e(k) = d(k) - y(k)$$

4. The error sequence $e(k)$ is used to determine the gain coefficients. An

algorithm is used for optimum setting of the coefficients. It is based on minimizing the sum of the squares of the errors, $Se^2(k)$.

5. The duration of the training sequence is so chosen that the adaptive equalizer converges to the optimum setting.

Adaptive equalizers are provided in high speed data modems. We will examine their operation and the training sequences in Chapter 4, Data Line Devices.

2.11 CLOCKED REGENERATIVE RECEIVER

Regeneration is the process of reconstructing the data signal from the received signal which has been attenuated, deformed, and possibly contaminated with noise during transmission. The clocked regenerative receiver removes as much disturbances as possible and then interprets the processed signal to determine the binary states of the received signal at the sampling instants. The sampling instants are decided by the clock signal. The digital signal is, then, regenerated. Figure 2.27 shows functional block schematic of a regenerative receiver. The receiver block schematic which we saw in Chapter 1 was the simplified version of this receiver.

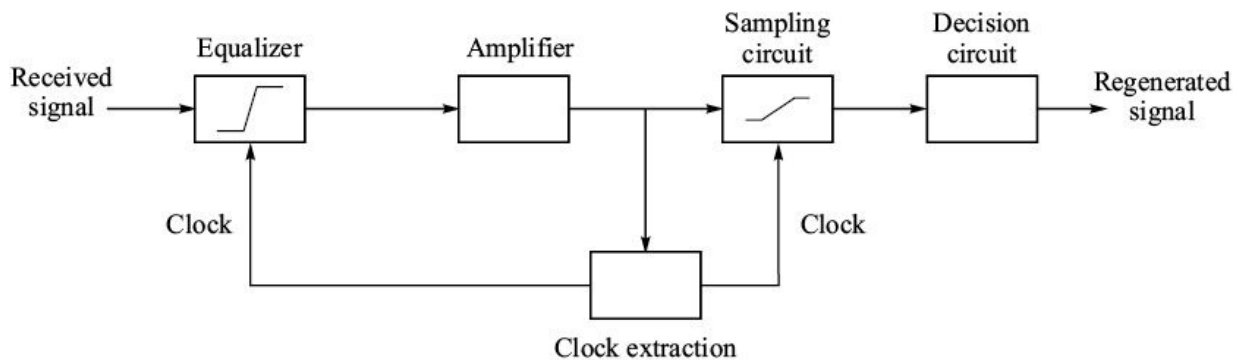


FIGURE 2.27 Clocked regenerative receiver.

The main features of the clocked regenerative receiver are given below:

- Equalization is carried out to remove intersymbol interference.
- The equalized signal is amplified to compensate for the attenuation and restore the signal to the required level.

- Sampling clock is extracted from the signal. This clock is inherently synchronized in frequency and phase to the received signal.
- The equalized and amplified signal is sampled using the clock. The clock determines the precise instants at which the sampling is done.
- The decision circuit maintains a threshold to decide the discrete value of the received signal. It compares the sampled signal with the threshold and generates a reconstructed signal.

Ideally, the sampled value of the signal should depend only on the current binary state of the signal but it may not so because of the residual intersymbol interference. The overall performance of the data transmission system is determined by ability of the decision circuit to discriminate between the binary states using the defined thresholds. The thresholds can be set using eye pattern which is discussed next.

2.12 EYE PATTERN

Eye pattern is a convenient method of displaying the effect of intersymbol interference on an oscilloscope. Figure 2.28 shows an example of a response curve which extends over three clock periods for a binary unipolar signal. Let us assume that it is generated whenever symbol 1 is transmitted on the channel. The response curve has been divided into three parts, namely, A, B, and C. It is obvious from the figure that there is significant intersymbol interference at the adjacent sampling instant following the symbol.

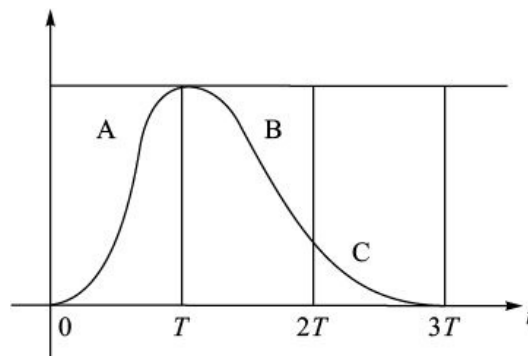


FIGURE 2.28 Response curve.

Since the response curve extends over three clock periods, the received signal at any instant is determined by the last three symbols. Table 2.4 lists all possible

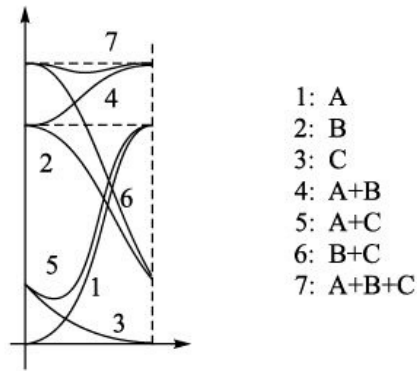
combinations of the three bit

TABLE 2.4 Response Curves for Various 3-Bit Input Sequences

Bit sequence	Response parts to be superimposed
0 0 0	0
0 0 1	A
0 1 0	B
0 1 1	A + B
1 0 0	C
1 0 1	A + C
1 1 0	B + C
1 1 1	A + B + C

sequences. The resulting signal waveform during a clock period is obtained by superimposing the sections A, B, and C of the response curve depending on the bit sequence.

Figure 2.29a shows the waveforms generated by the various bit sequences. If a random data signal is transmitted, all these sequences will be generated and if the received signal is observed in an oscilloscope, the display on the screen will be overlapping waveforms corresponding to each bit sequence (Figure 2.29b). The extracted clock of the data signal is used for horizontal synchronization in the oscilloscope. The display is in the shape of an 'eye' and is, therefore, termed as eye pattern. The opening X is meaningful indicator of the quality of equalization. When equalization is poor, the eye opening is small. A perfect equalization, *i.e.* zero intersymbol interference, appears as maximum opening of the eyes.



(a) Response curves for various input bit sequences

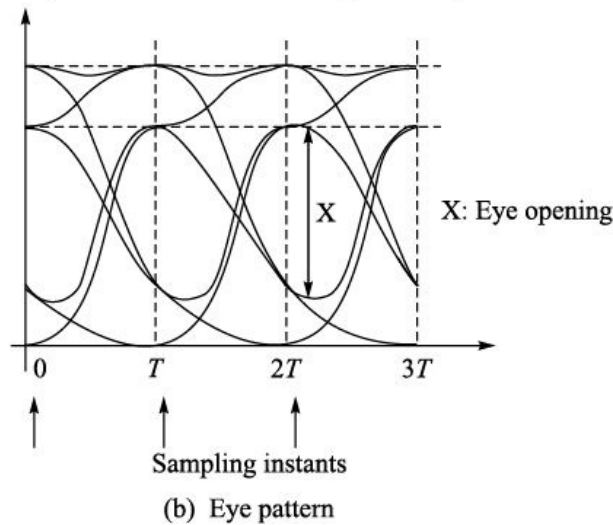


Figure 2.29 Formation of eye pattern.

The eye pattern is of practical value as it allows: (a) fine tuning of the equalizer to get minimum intersymbol interference and (b) adjustment of the phase of the local sampling clock in the regenerative receiver. The phase is so adjusted that the sampling takes place exactly at the maximum eye opening where the discrimination between the discrete levels is maximum.

SUMMARY

Transmission media for carrying digital signals can be of three types—metallic copper pair, optical fibre or radio (wireless). The characteristics of transmission medium determine the bit rate it can support for a given distance between the transmitter and receiver. Group delay and attenuation are the two basic parameters of copper pair. For optical fibre in place of group delay, dispersion parameter is defined. Transmission characteristics of the media introduce linear distortions in the transmitted signals. These distortions in time domain are referred to as InterSymbol Interference (ISI). Nyquist defined a raised cosine

filter characteristics for a transmission channel to have no intersymbol interference. Equalizers are used to correct the characteristics of the transmission media. Transversal filter equalizer is a tapped delay line that minimizes the intersymbol interference at the sampling instants.

Near end crosstalk (NEXT) and far end crosstalk (FEXT) are the two other important impairments of the metallic pairs when they are bundled as a cable. Crosstalk is reduced by using balanced mode of transmission and by shielding the metallic pairs individually using metallic braid or foil.

Unshielded twisted pair (UTP) and shielded twisted pair (STP) cables are the most commonly used transmission media over short distances. Five categories of UTP cables (CAT 1 to CAT 5) have been standardized. CAT 5 cable can be used up to 100 Mbps bit rate.

Coaxial pairs were used extensively in local area networks earlier. But these have been now replaced by CAT 5 cables for short spans, and by optical fibres for longer spans.

Optical fibres do not suffer from crosstalk and electromagnetic interference. Optical fibres are of three types: multimode-step index, multimode graded index, and monomode fibre. Monomode fibre has the lowest dispersion and therefore can support very high bit rates (\sim Giga bit per second) over long spans of the fibre.

Radio transmission medium found limited applications in data communications till recently. Radio medium is electromagnetic interference prone and has limited bandwidth. Wireless local area networks have become very popular in last few years.

EXERCISES

1. Impulse response of a channel is given by

$$h(t) = \sin(t/T)/(t/T)$$

Sketch the output of the channel when impulses corresponding to input data 1011 are applied to the channel. Assume 1 is represented by a unit positive impulse and 0 by a unit negative impulse.

2. A single impulse input to a transmission channel results in received samples of

$$0.2 \ 0.5 \ 1.0 \ 0.5 \ 0.2$$

where the middle sample is in the sampling interval corresponding to the transmitted impulse and the other entries are in the adjacent intervals.

Determine the tap gains of a three tap transversal filter equalizer which will eliminate the intersymbol interference.

3. (a) In a voice channel, the noise level is 0.001 mW. What is the signal to noise ratio if the signal level is 1 mW?
 (b) Express the following levels in dBm:
 - (i) 100 W
 - (ii) 0.1 W
 - (iii) 1.0 mW
 - (iv) 1.0 nW
 - (v) 1.0 pW
4. The refractive indices of the core and the cladding are 1.5 and 1.48 respectively in a multimode fibre. Calculate the modal dispersion if its length is 40 km.
5. Calculate the chromatic dispersion in a fibre of length 10 km when the spectral width and the wavelength of the source are 50 nm and 820 nm respectively. $M = 110$ ps/nm km at 820 nm.
6. A signal consisting of 1 kHz and 2 kHz tones is sent on a transmission line. If 1 kHz component undergoes a phase shift of $p/3$ along the transmission line, what will be the phase shift of 2 kHz component if the signal is not distorted by the transmission line?
7. Figure E2.30 shows unit impulse response of a channel. Draw the eye pattern at the receiving end when a random sequence of unit impulses are applied at the input of the channel.

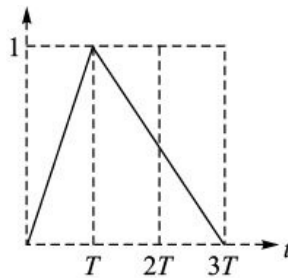


Figure E2.30.

- 1 As the signal travels along a transmission medium, it gets attenuated and distorted. A repeater regenerates the digital signal and makes it as good as the signal transmitted by the source.
- 2 A geo-synchronous satellite takes exactly 24 hours for its one revolution around the earth along its equatorial plane. Since the earth also takes 24 hours for one rotation, the satellite appears stationary with respect to a point on the surface of the earth.

3

Telephone Network

Public Switched Telephone Network (PSTN), or simply the telephone network, provides means of voice communication to any corner of the world. Its ready availability has induced the data processing community to make use of this network for transporting data as well. To make use of the telephone network for data transmission, we need to understand its signal transmission characteristics and the specifications of the service offered by it.

We begin this chapter with the topology of the telephone network and examine its components, access network, telephone exchanges, and the interconnecting trunks. We discuss principles of Frequency Division Multiplexing (FDM) and Time Division Multiplexing (TDM) methods. The standard multiplexing hierarchy used in the telephone network is explained thereafter. Structure of Synchronous Digital Hierarchy (SDH) frame and mapping of lower tributaries to SDH frame are described in detail. Then we move over to various signal impairments and the methods used to overcome them. Before close of the chapter, we discuss in detail Integrated Services Digital Network (ISDN) interfaces for voice and data services.

3.1 TELEPHONE NETWORK

The concept of a network emerges when several sources and several sinks of information are to be interconnected. The telephone network provides interconnection service for voice communications to its subscribers. Human speech signals occupy the frequency band extending from a few tens of Hz to about 8 kHz. Most of the information is contained in the 100–500 Hz frequency range. The higher frequencies serve to give ‘character’ to a voice. As a compromise between high quality of speech which demands full voice bandwidth, and cost which calls for narrow bandwidth, the telephone network

provides voice service over the frequency band 300–3400 Hz.

3.1.1 Network Topology

To interconnect two subscribers, the telephone network assigns transmission channels to them either on a permanent basis or dynamically. In the latter case, switching is necessary. The switching operation interconnects the subscribers when they request for interconnection. Switching function is performed by the network exchange (Figure 3.1).

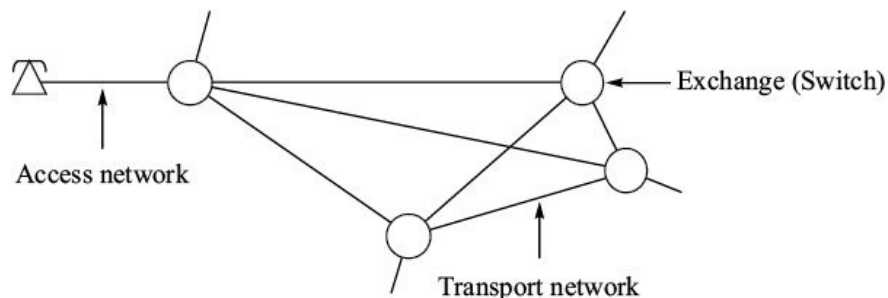


FIGURE 3.1 Telephone network.

We can view the telephone network as consisting of three parts:

- access network that connects telephone instrument to the telephone exchange,
- exchange or switch that performs the switching function, and
- *transport network* that interconnects various telephone exchanges.

We will briefly describe each of these in the following sections. Access and transport networks are very important to us since we share this infrastructure for establishing data networks as well.

3.1.2 Single Exchange Area

Let us examine a simple telephone network consisting of one exchange serving the subscribers located within its service area (Figure 3.2). The service area is usually less than a hundred square kilometers. The network resources consist of access network and a telephone exchange.

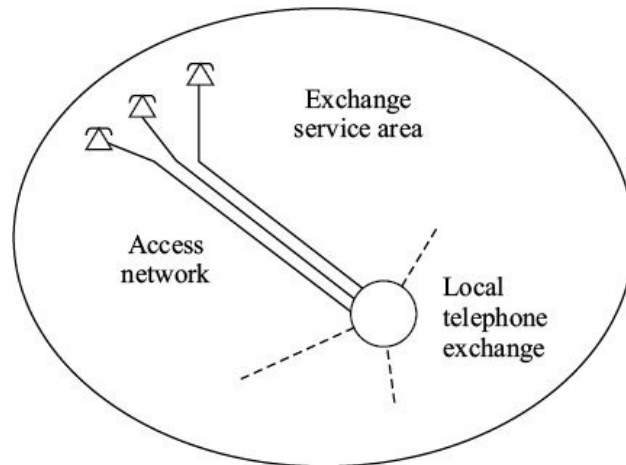


Figure 3.2 Single exchange telephone network.

The access network consists of balanced twisted-pair copper cables from the telephone exchange to the subscribers. Each telephone instrument of the network is connected to the exchange through these cables and is identified by a telephone number. In the exchange, each pair of the local network is terminated on a distribution frame. This two-wire connection from the exchange to the subscriber provides both-way voice transmission capability.

The telephone exchange consists of switching equipment which accepts the dialed digits from a telephone instrument and establishes connection to the dialed telephone number. The switching operation in the exchange involves operation of some relay contacts which connect one subscriber line to another. The connection through the exchange provides both-way AC continuity for speech signals. DC signals are blocked by a transmission bridge which couples the subscriber line to the exchange equipment either through capacitors or through transformers (Figure 3.3). DC feed from exchange to the subscriber's telephone instrument is required to energize it. It was also used till late eighties for signaling (e.g. sending dialed digits) in the form of make and break pulses.

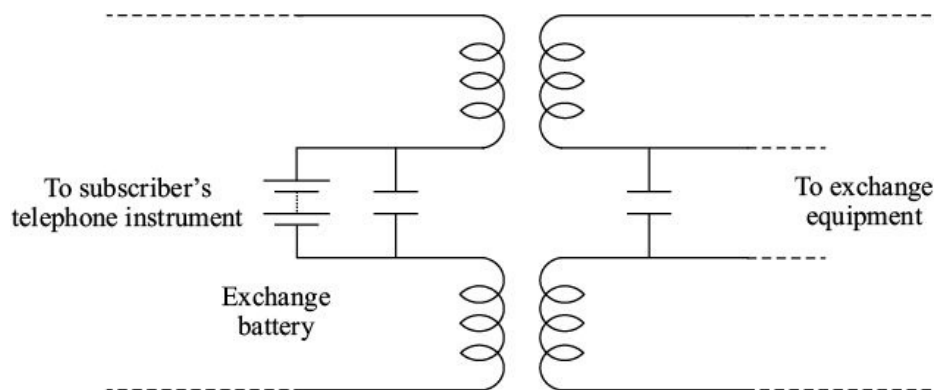


FIGURE 3.3 DC blocking transmission bridge.

Electromechanical telephone exchanges deploying relays have now been replaced by electronic exchanges. In electronic exchanges, the speech signal is first converted into digital signal of 64 kbps and then it is switched. The digital signal is converted back into the analog form before transmission to the subscriber at the other end. For conversion to digital form, the speech signal is band limited to 300–3400 Hz by the exchange equipment.

The access network too is undergoing a radical change. Optical fibre cables are now used in place of the copper cables. Optical fibre cables bring with them the advantage of large bandwidth that can be shared to provide data communications services also.

3.1.3 Multiple Exchanges

A single exchange can serve a limited number of subscribers. It can serve the subscribers located up to a maximum distance of about five kilometers if the access network consists of copper cables. When the number of subscribers and their geographic distribution expand, it becomes more economical to establish additional exchanges. Each exchange has its defined exchange service area and its local access network. It is identified by its unique exchange code (Figure 3.4). The exchange code is two (or three) digit prefix of a telephone number, *e.g.* the telephone number 678 45267 belongs to the telephone exchange having code 678.

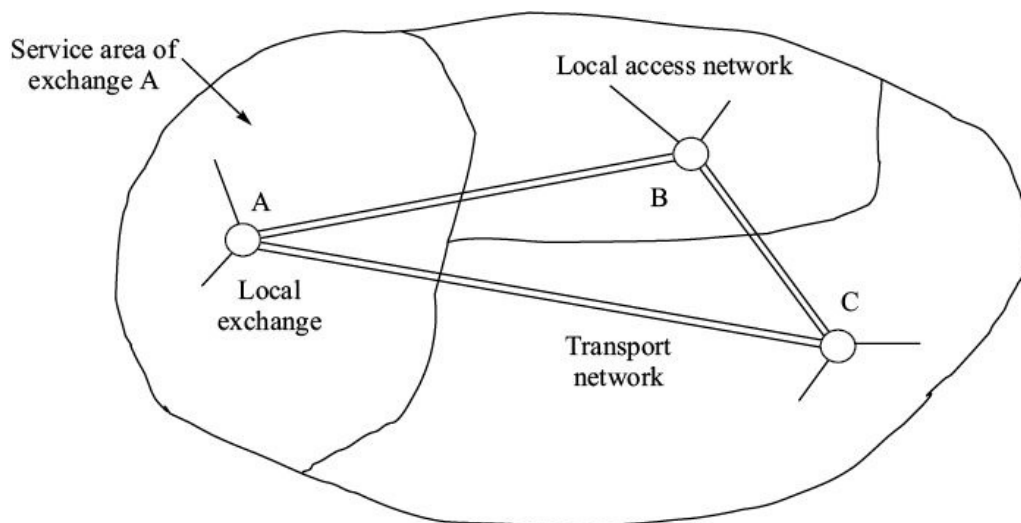


FIGURE 3.4 Multi-exchange telephone network.

The exchanges are interconnected through transport network to carry the signals from subscribers belonging to one exchange to the subscribers belonging to another exchange. When a subscriber dials a telephone number that belongs to

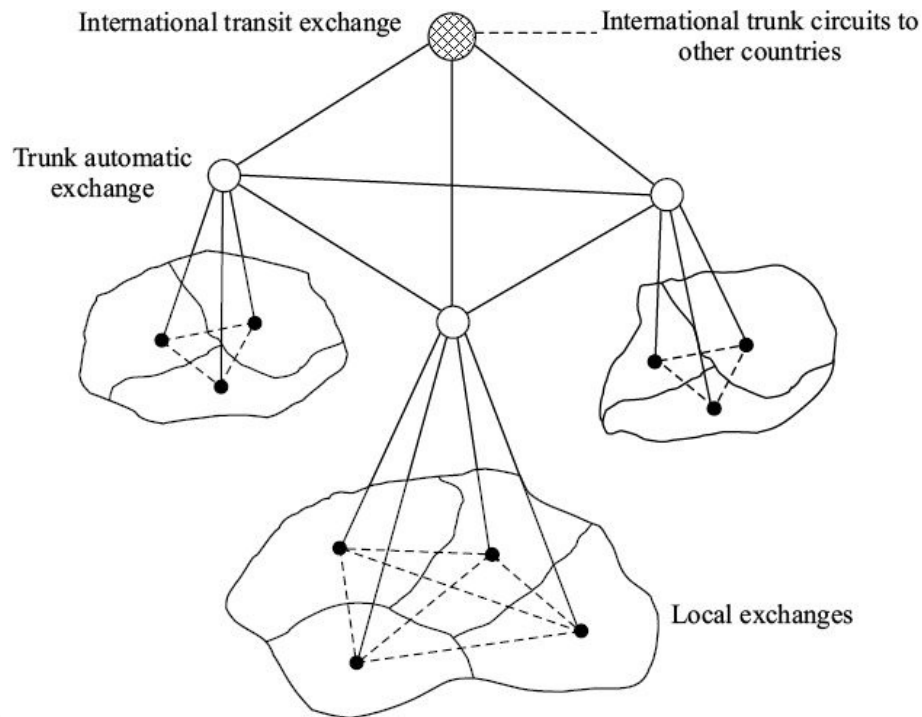
a different exchange as identified by the exchange code, the telephone call is routed by the originating exchange to the terminating exchange through the transport systems that interconnect the two exchanges. Transport systems are based on analog or digital multiplexing of voice channels (described later). Each channel has a bandwidth of 300–3400 Hz. Transport systems are based on microwave or optical fibre cable technologies.

In the era of electromechanical exchanges, the inter-exchange transport system consisted of balanced twisted copper pair cables called junction cables. Junction cables had lower attenuation than local cables. The junction cables were generally loaded, in which case the bandwidth got restricted to about 4 kHz.

3.1.4 Trunk Automatic Exchanges

An average sized city may have several telephone exchanges. The total number of exchanges in a country may be several hundred or even several thousand. Providing interconnectivity among these exchanges poses a problem because each exchange needs to be connected to every other exchange. The problem is resolved by introducing another level of switching, called *trunk automatic exchange* (Figure 3.5). One trunk automatic exchange is established in a city or a group of adjoining cities designated as one area. Each area is given an area code. The trunk exchanges are interconnected using long distance transport network based on microwave, coaxial cable, satellite, and optical fibre technologies.

The network consisting of trunk exchanges provide interconnectivity for carrying telephone traffic from one area to the other. When a customer of an area wants to establish a connection to a customer in another area, he dials the destination area code followed by the telephone number of the distant customer. His call is routed through his local exchange, the local trunk exchange, the distant trunk exchange and the distant local exchange. To distinguish between



the local calls

FIGURE 3.5 Worldwide telephone network.

and inter area calls, 0 is prefixed before the area code. Calls that have 0 prefix are routed through the trunk exchange. Thus the local traffic¹ within an area does not flow through trunk exchanges.

The telephone network may have even more than two levels of hierarchy depending on the number of areas to be interconnected, number of trunk routes, and degree of routing flexibility provided in the network.

3.1.5 International Transit Exchange

The various telephone networks of different countries are interconnected through yet another level in the hierarchy of telephone exchanges. These are called *international transit exchanges* (Figure 3.5). International transit exchanges are interconnected through international trunk circuits. The trunk exchanges of a telephone network have access to one international transit exchange and the subscribers need to dial yet another code to route the call to the international transit exchange of the required country.

3.2 TRANSPORT NETWORK

The circuits which interconnect the local, trunk, and international transit exchanges carry the speech signals over long distances ranging from a few tens to several thousands of kilometers. The speech signals need to be amplified periodically for transmission over such long distances. There are two basic problems in this regard:

- An amplifier is a unidirectional device while a two-wire telephone circuit provides both-ways transmission. Therefore, the transmission paths for outgoing and incoming speech signals must be split, *i.e.* the two-wire circuit must be converted into a four-wire circuit.
- The trunk circuits are point-to-point and very large in number. Each trunk circuit requires its individual amplifiers to compensate for cable attenuation. Further the speech signal needs to be amplified repeatedly over long distances, since signal to noise ratio cannot be allowed to fall below the specified value. Therefore, a very large number of amplifiers are required.

The conversion of a two-wire circuit to a four-wire circuit is carried out using hybrids. As for the second problem, a cost-effective solution is to multiplex the speech signals so that there is one multiplexed outgoing signal and one multiplexed incoming signal. Of course, the multiplexed signal has a wider bandwidth and, therefore, calls for transmission equipment (amplifiers, equalizers, medium, etc.) having wider bandwidth.

3.2.1 Hybrids

Hybrids are used for converting a bidirectional two-wire circuit into a four-wire circuit which has separate pairs for 'go' and, 'return' directions. A hybrid is a four-port balanced transformer which provides isolation of the opposite ports (Figure 3.6). It is designed for 600 ohm terminations at its ports. When properly terminated, a speech signal entering one port gets equally divided between the two side ports and no signal flows to the opposite port.

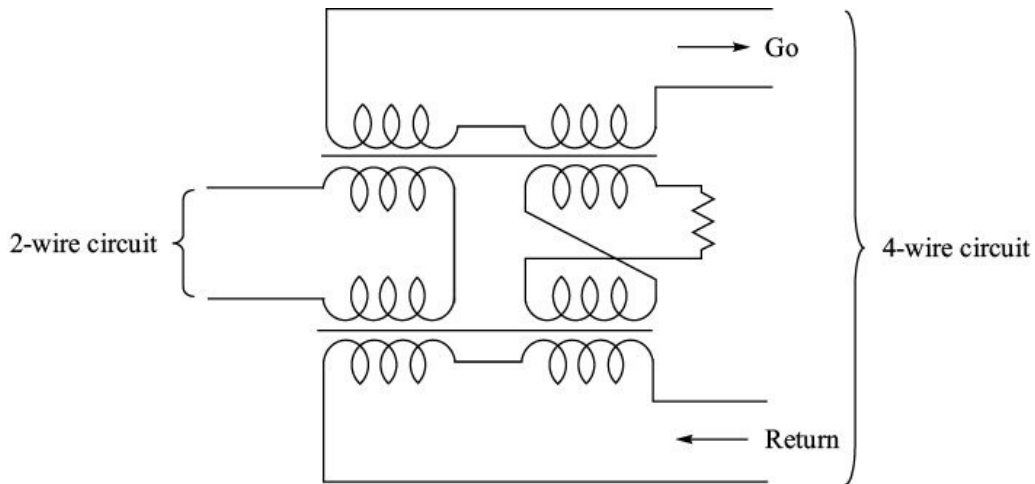


FIGURE 3.6 Hybrid for 2-wire/4-wire conversion.

In a two-wire circuit, two hybrids are required, one at each end. Figure 3.7 shows how two hybrids are connected. The speech signal entering port A of hybrid H_1 is divided equally between ports B and D. Port B is connected to the transmit side of the four wire circuit. The speech signal which appears on port B is transmitted to the other end after amplification. Port D is connected

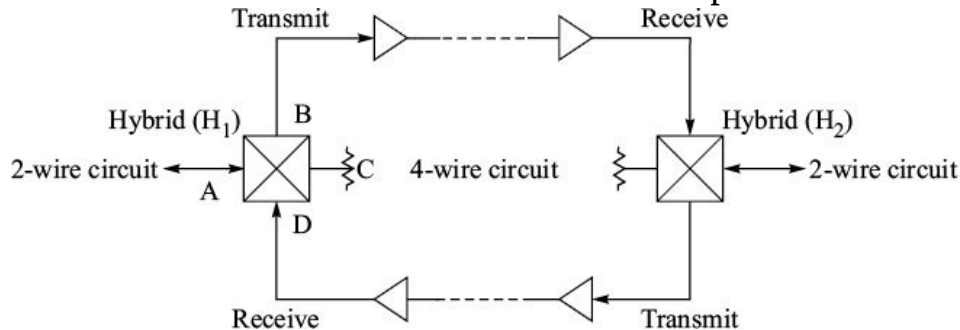


FIGURE 3.7 2-wire end-to-end circuit using hybrids.

to the output of the receive amplifier and, therefore, the speech signal from A appearing at this port is dissipated.

The speech signal received from the distant end appears at port D of the hybrid. It is also equally divided between ports A and C. The signal appearing at port A is transmitted on the two-wire circuit while the signal appearing at port C is dissipated in the resistive termination. If the hybrid is properly terminated, no part of the received signal goes to port B.

Hybrids are provided in a telephone network when the speech signals need to be processed. For example, in electronic exchanges, 2-wire to 4-wire conversion needs to be carried prior to digitalization of analog speech signals. Therefore, hybrids are installed in the

exchange (Figure 3.8).

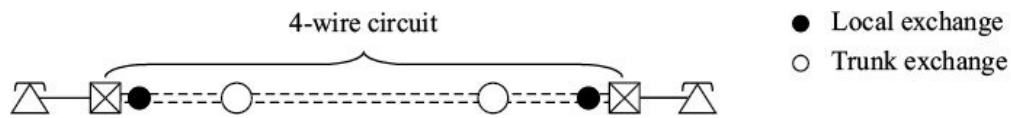


FIGURE 3.8 Use of hybrids in telephone exchanges.

3.2.2 Multiplexing

Multiplexing involves grouping of several channels in such a way as to transmit them simultaneously on the same physical transmission medium (e.g. cable or carrier frequency of a radio link). At the receiving end, *demultiplexing* is performed to separate the channels. In the telephone network, each channel provides a bandwidth of 300–3400 Hz for speech signals. Multiplexing and demultiplexing of ‘go’ and ‘return’ channels is done separately and, therefore, the multiplexing and demultiplexing equipment comes between the two hybrids of a two-wire circuit (Figure 3.9).

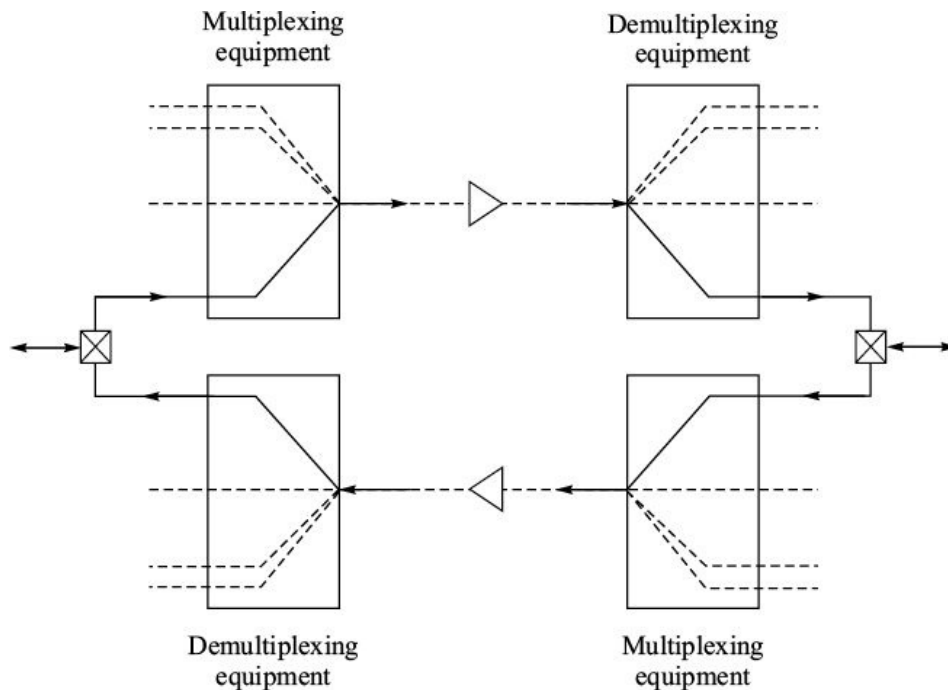


FIGURE 3.9 Multiplexing and demultiplexing of telephone channels.

There are two basic multiplexing technologies in the telephone network, namely, Frequency Division Multiplexing (FDM), and Time Division Multiplexing (TDM). FDM is the older technology based on analog transmission principles and TDM is the new technology based on digital transmission principles.

Frequency division multiplexing (FDM). *Frequency division multiplexing* involves translation of the speech signal from the frequency band 300–3400 Hz to a higher frequency band. Each channel is translated to a different band and then all the channels are combined to form a frequency division multiplexed signal. In FDM, the speech channels are stacked at intervals of 4 kHz to provide a guard band between adjacent channels.

Frequency translation is done by suppressed-carrier amplitude modulation of a carrier (f_c) by the speech signal. Of the two sidebands generated in this process, the upper sideband is separated using a bandpass filter. This process translates the speech channel to frequency band, f_c to $f_c + 4$ kHz. By choosing different carrier frequencies at interval of 4 kHz, we can stack a number of speech channels one after the other. Figure 3.10 shows how three speech channels can be multiplexed. The frequency band of a speech channel is usually represented as a small right-angled triangle. The lower end corresponds to 0 Hz and the upper end corresponds to 4000 Hz. The speech signal lies within this band from 300 to 3400 Hz.

To facilitate interconnection among the different telecommunication system in use worldwide, ITU-T has recommended a standard frequency translation plan. The smallest multiplexed unit consists of 12 channels and is termed as a group. The groups are multiplexed to form supergroups and so on. In other words, each bigger multiplexed unit is composed of several immediately lower multiplexed units. Table 3.1 shows this hierarchy.

Name	Number of channels	Bandwidth	Composition
Group	12	48	12 Speech channels
Supergroup	60	240	5 Groups
Master group	300	1232	5 Supergroups
Supermaster group	900	3872	3 Master groups

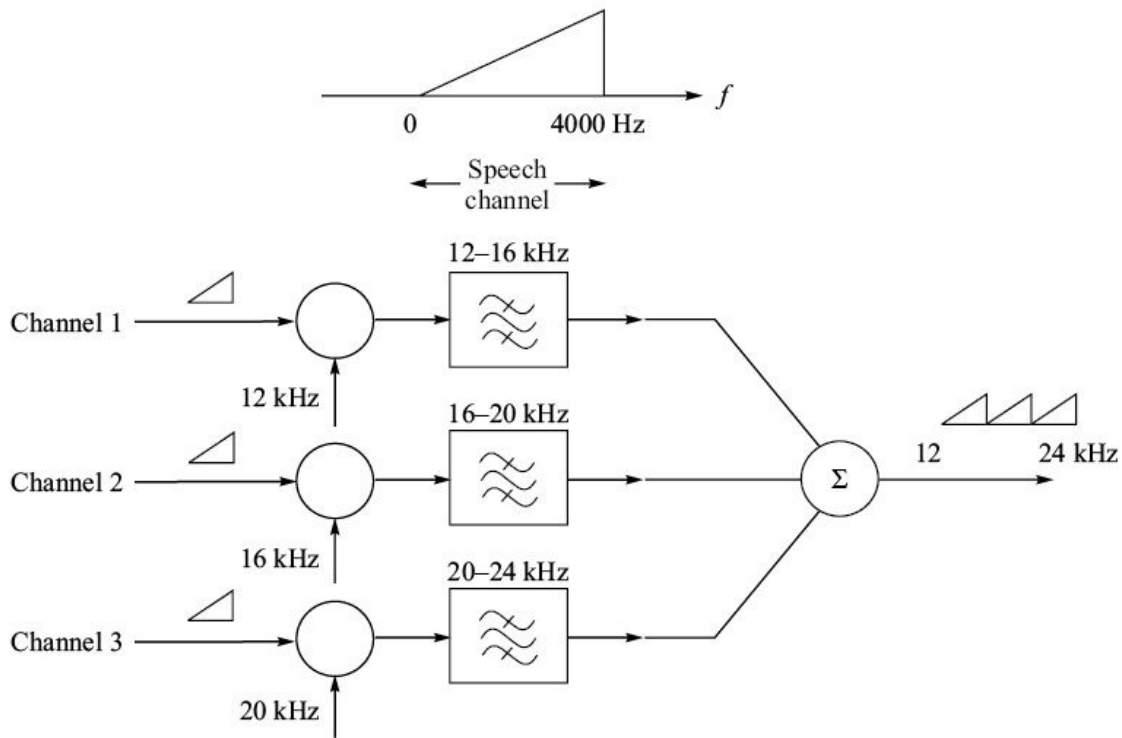


Figure 3.10 Frequency division multiplexing.

At the demultiplexer, the reverse process is carried out. Each higher level is translated to several signals of the next lower level. Finally each group is demultiplexed to 12 speech channels.

Time division multiplexing (TDM). The basis of *time division multiplexing* is the sampling theorem which states that a signal containing frequency components less than f_{\max} can be entirely determined by its equidistant samples taken at the rate f_s , such that $f_s \geq 2 f_{\max}$. The theorem emphasises that an analog signal has high-time redundancy which can be utilized for transmitting additional information. The voice channel has the highest frequency of 3400 Hz. If it is sampled at the rate of more than 6800 samples per second, the speech signal can be entirely reconstructed from the samples. It is to be ensured, before sampling is carried out, that the speech signal does not contain any frequency component higher than 3400 Hz. Therefore, a low-pass filter is always provided just before the sampler.

The standard sampling rate for the speech signal is 8000 samples/second to allow for gradual slope of the filter characteristics. At this rate, the samples are separated by 125 ms (Figure 3.11a). This time can be utilized for sending samples of other speech channels (Figure 3.11b). The sampling clocks of

different channels are synchronized and staggered in time so that the channels generate non-overlapping samples.

Demultiplexing in TDM involves separating the samples of different channels. Since multiplexing has been done sequentially, the demultiplexer can separate the samples of different channels but it faces the problem of identifying the channels. Therefore, a flag is also multiplexed along with the samples. The flag has a unique attribute that can be identified by the demultiplexer.

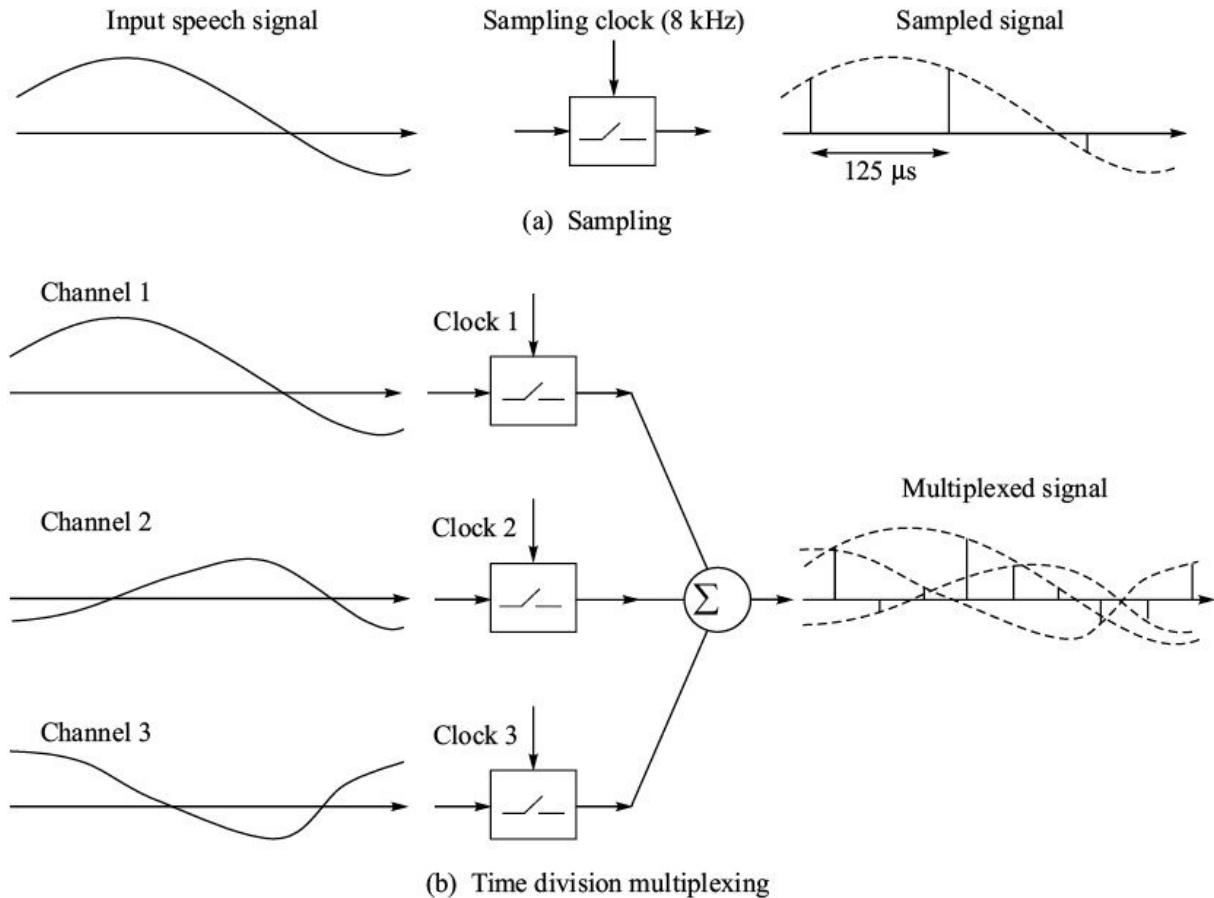


FIGURE 3.11 Sampling and time division multiplexing.

The first sample just after the flag belongs to channel one (Figure 3.12). The flag followed by one complete cycle of sampling is termed a ‘frame’.

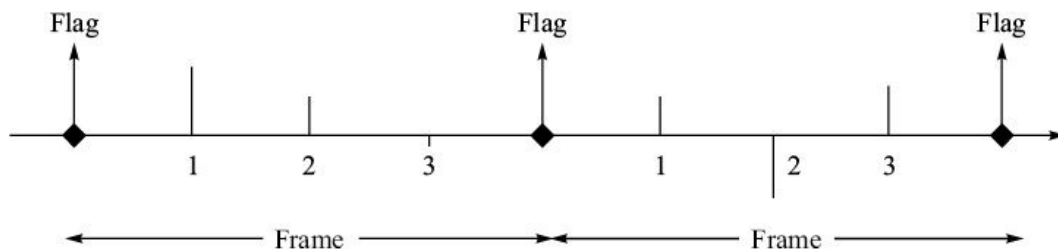


FIGURE 3.12 Identification of samples of a TDM signal using flag.

Once the samples have been separated, the speech signal can be reconstructed by passing the samples through a low-pass filter with the cut-off at 3400 Hz.

3.2.3 Pulse Code Modulation (PCM)

Although in pulse form, the time division multiplexed signal is still an analog signal as the sample levels can have infinite possible values. *Pulse code modulation* is used for converting these analog samples to digital form. This process involves two stages namely, quantization, and coding which we now discuss.

Quantization. *Quantization* is approximation of the level of the sample by the nearest value drawn from a finite assortment of discrete levels. For example, if the set consists of discrete levels 0, 1, 2, ..., 7 volts and the sample level is 3.2 volts, it is approximated by a discrete level of 3 volts. Note that by approximating the level of 3.2 volts by 3 volts, we have introduced some error because the receiver will later generate the sample of 3 volts when it receives the coded sample. This error is called *quantization error*. Quantization error is serious at low levels. An error of 0.1 volts in 5 volts is 2 per cent while the same error in 1 volt is 10 per cent. Therefore, a non-uniform quantization law is adopted in the telephone channels. It ensures equal quantization error percentages at all the levels. There are two quantization laws used worldwide, A-law and m-law. A-law is adopted in India and European countries. m-law is used in Japan and USA.

Coding. *Coding* involves converting the discrete level of the sample after quantization to the binary code of fixed length, *e.g.* 3 volts may be coded as 011. The number of bits in the code is determined by the total number of discrete levels. In telephony, 256 discrete levels are used and they are coded using eight bits. As the sampler generates 8000 samples per second and each sample is coded into eight bits, the bit rate of a digitalized speech channel is $8 \times 8000 = 64$ Kbps.

3.2.4 30-Channel PCM Signal

ITU-T has recommended standards for PCM channels. The basic PCM signal provides for multiplexing of 30 speech channels. Eight bit codes of the samples of these channels are time-division multiplexed. Figure 3.13 shows the format of the 30-channel PCM frame.

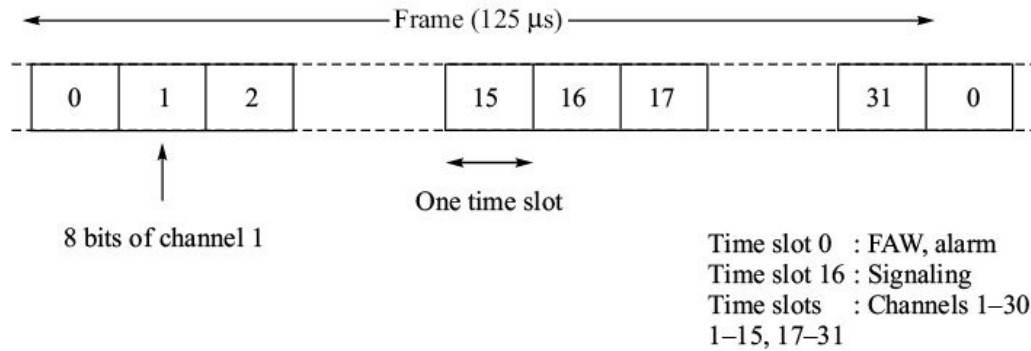


Figure 3.13 30-channel PCM frame.

- The 30-channel PCM frame consists of 32 time slots. Each time slot (TS) contains an 8-bit code. The frame starts with time slot TS-0 which contains the flag called Frame Alignment Word (FAW) and alarm signals in alternative frames.
- TS-1 to TS-15 and TS-17 to TS-31 contain sample codes of the 30 speech channels.
- TS-16 contains the signaling information. Signaling information pertains to the dialed digits and control information that passes between two telephone exchanges for establishing and disconnecting telephone connections. Each TS-16 contains signaling information of two channels. If a TS-16 contains signaling information of channels 1 and 16, the next TS-16 in the next frame will contain signaling information of channels 2 and 17. After fifteen frames the cycle is repeated.

Since the sampling rate is 8000 per second, the frames are generated at the same rate. The number of bits in a frame being $32 \times 8 = 256$, the bit rate of 30-channel PCM signal is $256 \times 8000 = 2.048$ Mbps. 30-channel PCM signal is commonly referred to as E1. In North America and Japan, 24 voice channels are multiplexed into 1.544 Mbps PCM signal. It is commonly referred to as T1.

Note that the voice channels are byte-interleaved, each byte being 8-bits long. The bytes of a voice channel occupy specific time slot in all the frames.

3.2.5 Plesiochronous Digital Hierarchy (PDH)

Just like the analog hierarchy, ITU-T has recommended standards for the digital multiplexing hierarchy. Each higher order multiplexed signal is derived from four immediately lower order multiplexed signals, called tributaries (Table 3.2).

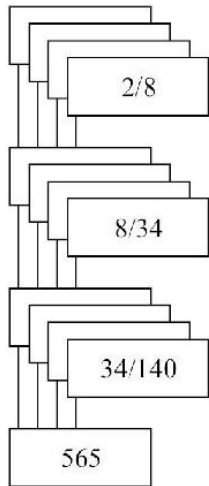


TABLE 3.2 Plesiochronous Digital Hierarchy

Order	Number of channels	Bit rate	Composition
1	30	2.048	30 × 64 kbps
2	120	8.448	4 × 2.048 Mbps
3	480	34.368	4 × 8.448 Mbps
4	1920	139.264	4 × 34.368 Mbps
5	7680	564.992	4 × 139.264 Mbps

Second order and onwards multiplexing is bit-interleaved, as opposed to first order multiplexing (E1) which is byte-interleaved (Figure 3.14). There is frame alignment word to mark the beginning of the frame in all the tributaries. All the tributaries that are multiplexed have same nominal frequency but minor frequency deviation (e.g. 50 parts per million bits in an E1) are permitted. Instead of calling such tributaries synchronous, we term them as *plesiochronous*.

When plesiochronous tributaries are multiplexed, bit by bit, faster tributaries will have some extra leftover bits. To accommodate these bits, justification bit positions are provided in the

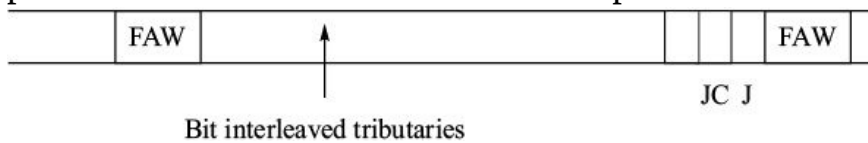


FIGURE 3.14 Multiplexing of plesiochronous signals.

frame. For example if an E1 tributary is running at slightly higher speed, say at 2.048050 MHz, the extra 50 bits that are received every second need to be accommodated in the next higher order frame. These bits are accommodated at justification bit positions (J). Without going into details, it may be mentioned that there is justification control bit (JC) which indicates to the receiving equipment whether the justification bit position (J) is carrying tributary bit or it is vacant. The gross bit rate of each higher order multiplexed signal is not exact multiple of the lower order signal because additional frame alignment word and justification bits are needed to be introduced at each stage of multiplexing.

Bit interleaving and justification bits make the higher order multiplexing structure very rigid. If we want to retrieve an E1 from, say fourth order multiplexed signal, we need to retrace all the demultiplexing steps, *i.e.* from

fourth to third order, third to second order, and lastly second order to E1. This is so because we cannot identify directly the bits of an E1 tributary in a fourth order frame. Synchronous Digital Hierarchy (SDH), discussed next, permits this. SDH is called SONET in US.

3.3 SYNCHRONOUS DIGITAL HIERARCHY (SDH)

Synchronous digital hierarchy follows the byte interleaved frame architecture of E1 even at higher multiplexing levels. The frame structure is so designed that bytes of lower order tributaries can be readily identified at any level of multiplexing. This enables insertion/dropping of lower order tributaries to/from a multiplexed signal without going into intermediate stages of multiplexing or demultiplexing. Figure 3.15 shows a typical application of SDH Add-Drop Multiplexer (ADM) which can insert the required number of E1s in the STM-1 frame and drop them at various destinations.

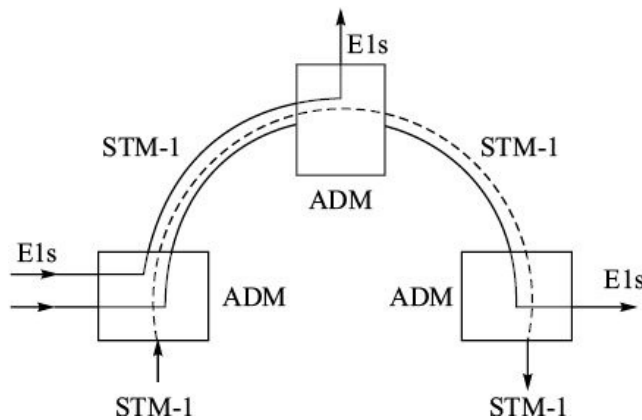


Figure 3.15 SDH add-drop multiplexers.

3.3.1 STM-1 Frame Structure

To understand frame structures in SDH, let us go back to 32-octet E1 frame (Figure 3.13). It can be redrawn as shown in (Figure 3.16). Note that 216 octet frame has two parts, overhead bytes and payload bytes. The overhead octets contain FAW, alarms, signaling information. Payload octets contain digitized speech signals. One E1 frame is transmitted in 125 m sec.

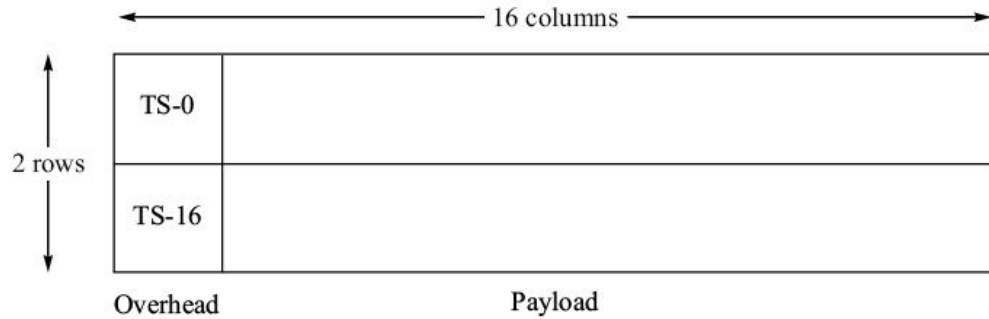


Figure 3.16 32-octet E1 frame.

The octet-oriented frame structure of E1 is extended to the next higher levels of multiplexing in SDH as shown in Figure 3.17. This basic frame is called Synchronous Transport Module-1 (STM-1). It consists of $270 \times 9 = 2430$ octets. The entire STM-1 frame is transmitted in 125 μ sec. The bit rate can be calculated as under: Bits in one frame $2430 \times 8 \text{ bits} = 19440 \text{ bits}$ Time to transmit 125 μ sec.

Bit rate $19440/125 = 155.52 \text{ Mbps}$

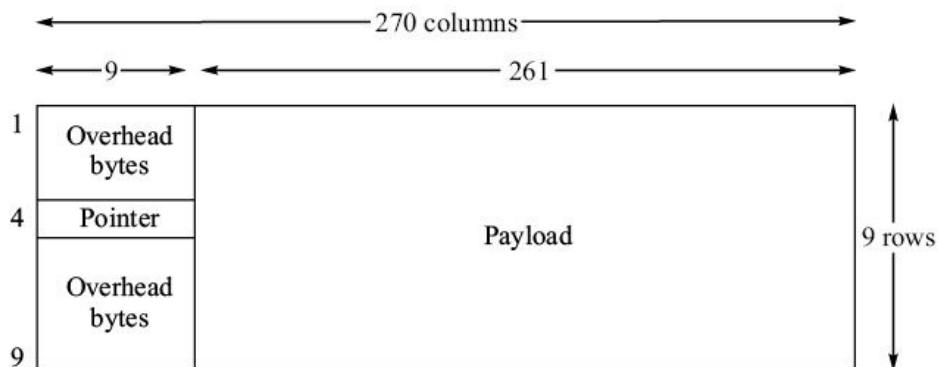


Figure 3.17 STM-1 frame structure.

STM-1 frame consists of three parts:

- Overhead bytes
- Pointer
- Payload.

The first nine columns contain overhead bytes and a pointer. The overhead octets contain several fields. The most important are first two octets of first row which contain frame alignment word (FAW) which marks the start of the STM-1 frame. Other fields in the overhead are for alarms, management of SDH equipment, and for error detection. The payload portion of the frame contains

E1s, 34 Mbps or 140 Mbs PDH signals, or data packets mapped on to the frame. These signals are packed into containers and then mapped to the payload section of STM-1 frame. Containerization enables their adding, dropping or transfer from one STM-1 to another.

3.3.2 Virtual Container (VC)

As mentioned above, bytes of a tributary are packed into containers before they are mapped to the payload of an STM-1 frame. Each container is associated with a *path overhead*. Path overhead identifies what is packed in the container. It also has bytes for error detection, path identifier, *etc.* A container with its path overhead is referred to as Virtual Container (VC). There are several types of virtual containers VC-4, VC-3, VC-2, VC-12, VC-11. VC-4 is the highest in the order. It can contain one 140 Mbps tributary or lower order VCs. VC-12 is for one E1 and VC-11 is for one T1.

The first byte of a virtual container identifies the VC. Knowing the location of the first byte, we can identify all the bytes of the virtual container. Therefore, a pointer is provided in the SDH frame. The pointer points to the first byte of the VC. The pointer shown in Figure 3.17 points to the location of the first byte of VC4 contained in the payload.

3.3.3 VC-4

VC-4 frame consists of 9 261 bytes and its first column contains path overhead, POH (Figure 3.18). The nine bytes of POH take care of error detection, alarm and path trace functions. Rest of VC-4 (9 260 bytes) contains the payload bytes. The bytes important to us are J1, B3, and C2. J1 byte contains the identification of VC-4 path from where it originated. B3 is used for error monitoring and C2 indicates the type of payload and whether the payload is equipped or not. If VC-4 is vacant, C2 is all zeroes. If a data signal is directly mapped on VC-4, C2 will be 15, 16, or 17 (Hexadecimal) depending on the type of the data signal .

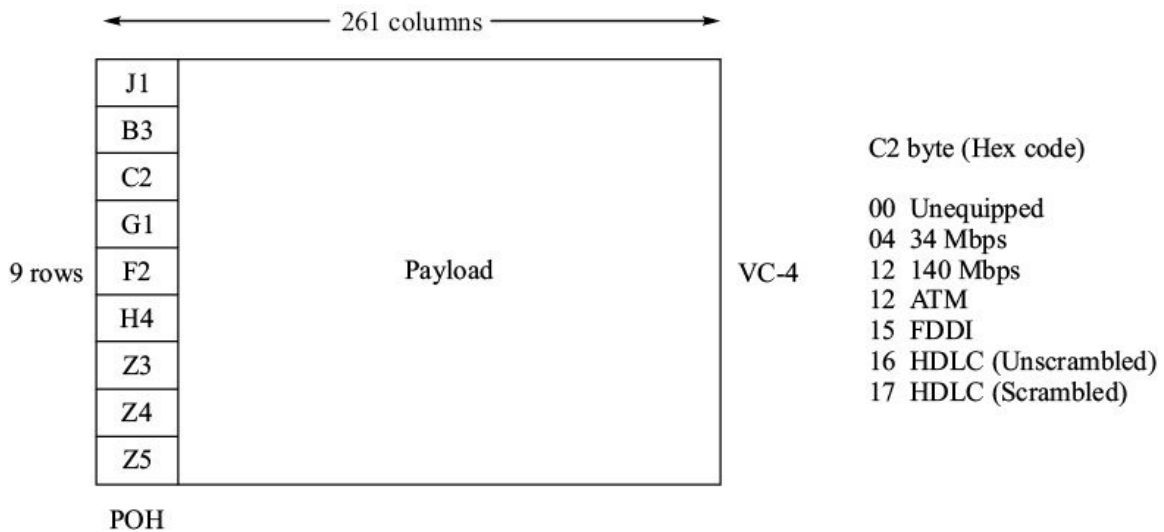


FIGURE 3.18 Virtual container (VC-4).

When a VC-4 is mapped on to an STM-1 frame, its phase will not be generally aligned with payload section of STM-1 frame. The POH bytes in the first column of VC-1 are very important because they are used for identification of the payload, error monitoring, and other functions associated with its path. It is necessary to first identify the POH bytes whenever VC-4 is processed. The pointer byte in STM-1 frame contains the location of first byte (J1) of VC-4 (Figure 3.19). VC-4 obviously overflows to next STM-1 frame, but it does not affect the operation in any way.

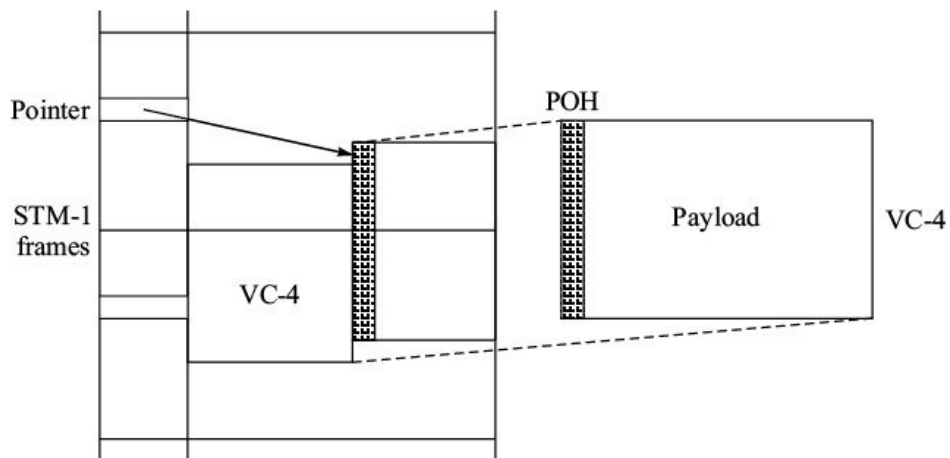


FIGURE 3.19 Mapping of VC-4 onto an STM-1 frame.

A VC-4 can contain one 140 Mbps PDH tributary, or three 34 Mbps PDH tributaries or 63 E1s. Since PDH signals are no longer used, we will concentrate only on mapping of E1s onto a VC-4.

3.3.4 VC-12

Mapping of E1 on to a VC-4 follows a step-by-step roundabout approach (Figure 3.20).

1. 128 bytes of an E1 along with 4 bytes of POH and 8 additional stuff bytes are packed as VC-12 having 136 bytes. The first byte V5 identifies the VC-12. Note that VC-12 containing 128 bytes of E1 can be formed in 500 msec. Therefore, one VC-12 will be spread over 5 STM-1 frames, each containing part of VC-12 (Figure 3.21).
2. VC-12 is mapped on TU-12s (Tributary Unit), each of 36 (9 4) bytes. The TU-12s are part of VC-4. We will shortly see how the TU-12s are packed in a VC-4. V1 and V2 bytes of tributary units comprise pointer that indicates the location of V5 byte of VC-12. Note that a V-12 is spread over five TU-12s.
3. A group of three TU-12s of three different E1s are packed as a TUG-2 (Tributary unit group) of 108 (9 12) bytes. Thus a TUG-2 contains 3 E1s. In SDH, multiplexing is always carried out by interleaving the columns of multiplexed units (Figure 3.22).
4. Seven TUG-2s are further packed into a TUG-3 that has payload of 756 (9 12 7) bytes. Along with overhead of 9 2 bytes, a TUG-3 has 9 rows and 86 columns. A Tug-3 contains 21 E1s.
5. Three TUG-3s are further packed into a VC-4. Thus a VC-4 has 63 E1s. Three TUG-3s occupy 258 (3 86) columns in a VC-4. Out of the balance three columns of the VC-4, the first contains POH and the other two are spare columns.

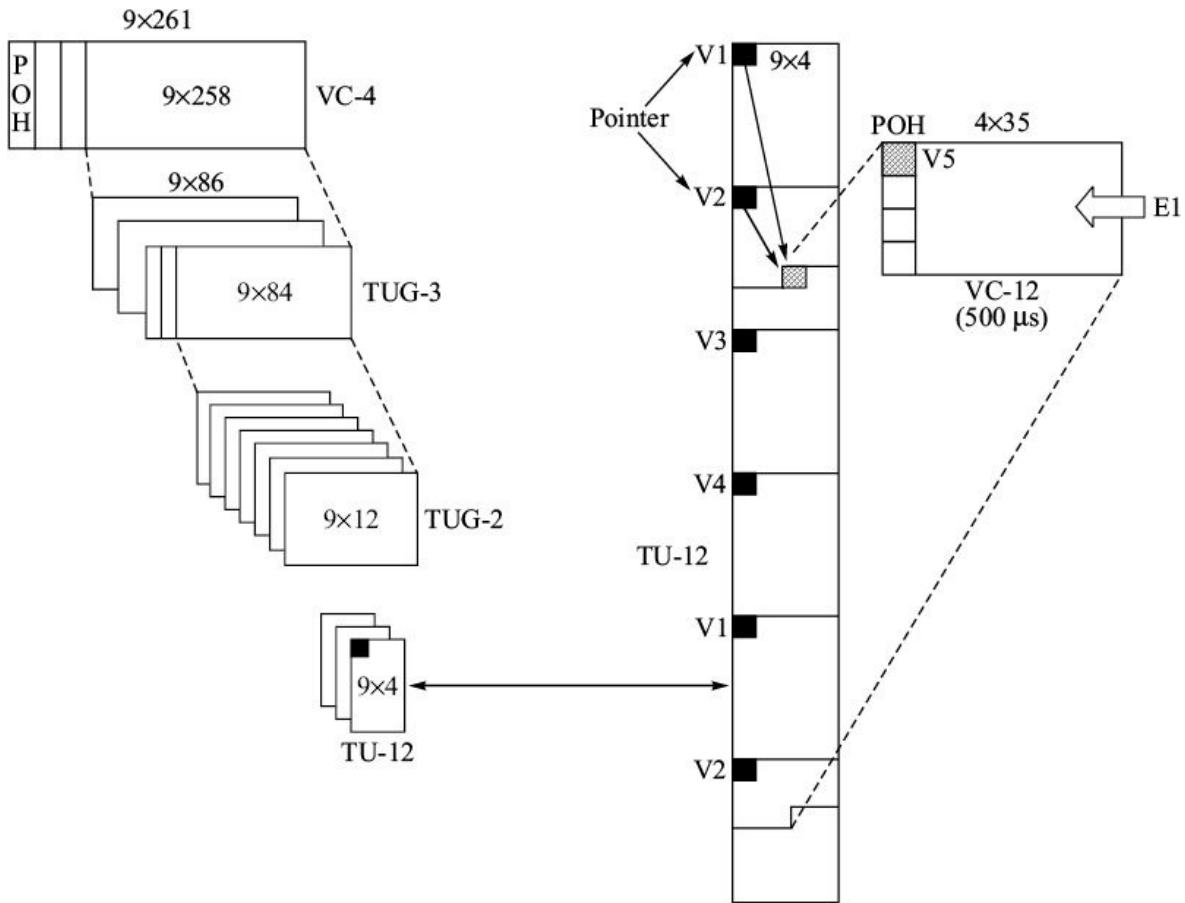


Figure 3.20 Mapping of E1 on VC-4.

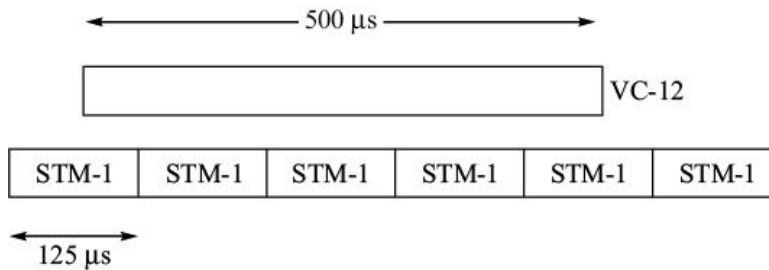


Figure 3.21 Formation of VC-12.

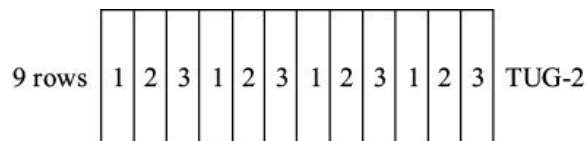


FIGURE 3.22 Multiplexing of TU-12s.

3.3.5 Mapping of Data Signals on STM-1

SDH allows direct mapping of data bytes on its VC-4. Data bytes are grouped as frame with flags to delineate the boundaries of the frame. The data bytes are

scrambled² and the bit stream is mapped on the VC-4 directly (Figure 3.23). An SDH link with data bytes directly mapped on to it is referred to as POS (Packet over SDH) link.

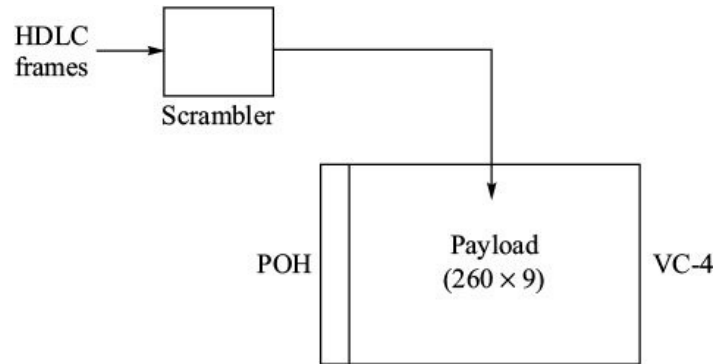


FIGURE 3.23 Mapping of data frames on VC-4.

3.3.6 Higher Order SDH Signals

STM-1 is base SDH signal which can be synchronously multiplexed to higher bit rates in multiples of four. Table 3.3 shows the hierarchy. All the higher order multiplexed signals are formed by column-interleaving as explained earlier (Figure 3.24).

TABLE 3.3 Multiplexing Hierarchy in SDH			
Order	Number of E1s	Bit rate Mbps	Composition
		155.52	
		622.08	
STM-1	63	2488.32	1 STM-1
STM-4	252		4 STM-1
STM-16	1008		16 STM-1
STM-64	4032	9953.28	64 STM-1

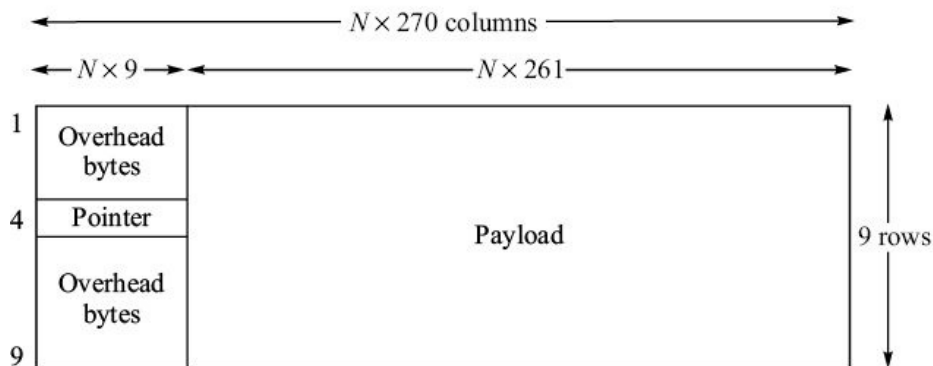


FIGURE 3.24 STM-N frame structure.

3.4 TRANSMISSION SYSTEMS FOR LONG DISTANCE NETWORK

Multiplexed speech channels are transmitted on the long distance network using cable transmission systems or radio transmission systems. The choice of transmission media depends on the terrain, required capacity and the cost.

Cable transmission systems. There are three alternatives for the cable transmission systems:

- Balanced metallic pair cable system
- Coaxial cable system
- Optical fibre system.

Balanced pair systems were used in the past for short distances and for low capacity systems. Balanced pairs were bundled with coaxial pairs in the same cable and used as transmission media for intermediate stations. Coaxial cable has been the most commonly used transmission media for many decades. It provided reliable long distance transmission of both analog and digital signals. During last decade, however, optical fibre systems have overtaken all other transmission systems because they offer very high capacity, scalability,³ and immunity to interference, and they are very cost effective.

Radio systems. The radio systems provide wireless communication links and are useful where the distance and terrain render the cable media uneconomical. In all the radio systems, the multiplexed signal is made to modulate a radio carrier which is transmitted as an electromagnetic wave. The receiving antenna at the other end picks up the radio carrier and the received radio signal is demodulated to get the multiplexed signal.

The radio systems are categorized based on the frequency bands as described in the last chapter. Telecommunication network finds little use of HF systems. HF band is used primarily for medium wave radio broadcast. VHF and UHF frequency bands are for low capacity telecommunication systems having capacity up to thirty speech channels per radio carrier.

SHF systems are referred to as microwave systems. There are two categories of microwave systems in the telecommunication network.

- Terrestrial microwave systems
- Satellite systems.

Terrestrial microwave systems are high capacity systems and provide channel capacities of the order of 300–2700 channels per radio carrier. These are line-of-sight system and need repeaters at spacing of about fifty kilometers due to curvature of the earth.

The satellite microwave systems provide high degree of deployment flexibility and channel-capacity granularity. The satellite acts as a repeater and covers a wide geographic area. Earth stations having capacities as low as single channel per carrier and as high as 1800 channels per carrier are used in the telephone network.

Satellite communication systems, however, have one disadvantage. The propagation time of radio wave from one satellite earth station to another through the satellite is about 250 milliseconds. Such large propagation delay causes distinctly audible echo in the voice channels. Therefore echo suppressors or cancellers must be used. Large propagation time has serious implications in data communications as we shall see in Chapter 8, Data Link Layer.

3.5 ECHO IN TRANSMISSION SYSTEMS

Introduction of hybrids in the transmission systems gives rise to a very serious problem, particularly when the link is through a satellite. Theoretically speaking, a hybrid should not cause any leakage of the signal to the opposite port if it has been properly terminated at all the four ports. In actual practice, the impedance at the port towards the subscriber (port A in Figure 3.7) is highly uncertain. It depends on the distance between location of the customer's telephone instrument with respect to the hybrid. The distance is not fixed and varies from customer to customer. As a result, the hybrid causes leakage of the received signal to the opposite port (Figure 3.25). This signal travels back to the customer who originated the signal and is heard as an *echo* by him.

The echo is present in all types of transmission systems but it is not heard distinctly as the delay of the echo is very small, ~ 20 ms in the terrestrial systems. In case of a satellite link, the delay is of the order of 500 ms and there is a distinct echo in

the channel.

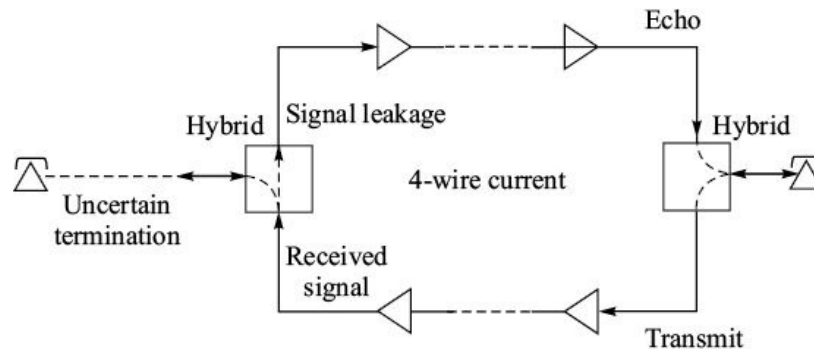


Figure 3.25 Echo generation in transmission systems.

Echo is a nuisance needs to be eliminated from the telephone circuits. Echo suppressors/cancellors are installed in the network for this purpose. In echo suppressors, transmit port of the hybrid is isolated when a signal is being received so that the signal through the hybrid does not travel back (Figure 3.26a). In echo cancellors, the echo is eliminated by subtracting a simulated echo signal from the transmitted signals (Figure 3.26b).

For data transmission on the telephone network, either a four-wire or a two-wire connection is used. When an end-to-end four-wire circuit is extended to the subscriber premises, hybrids do not come into the picture at all so there is no need of echo suppressors. When a two-wire circuit is used, transmit and receive signals have two different frequencies. Echo may be present but the receiver is not tuned to the echo. It can receive only the signal transmitted by the other end. Therefore, echo suppressors are not required for data circuits and are disabled if present in the transmission link.

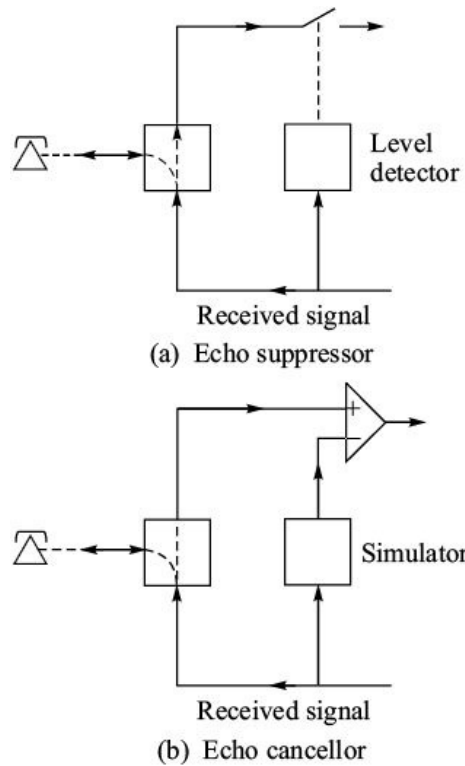


Figure 3.26 Echo suppression/cancellation in transmission systems.

3.6 NOISE IN TRANSMISSION SYSTEMS

Noise can be generally described as any received signal, which, when interpreted by the receiver, delivers incoherent information of no interest to the receiver. Noise gets added to the signal during its transmission and degrades the quality of the signal. Background noise is always present even in the absence of a useful signal. Sources of background noise are:

- Thermal noise
- Intrinsic noise of the electronic devices, *e.g.* shot noise
- Atmospheric disturbances
- Electromagnetic interference
- Cosmic sources.

Lightning discharge and rain attenuation are the two sources of atmospheric noise. Electromagnetic interference is caused by other radio systems, discharges in commutator motors, and spark plugs of vehicles. All celestial bodies generate

electromagnetic interference which is picked up by the radio system antennae. Sources of thermal noise and shot noise are present within the telecommunication equipment.

Signal distortion due to non-linearities in a telecommunication system results in supplementary noise which appears only when the signal is present. Two important causes of such noise are intermodulation and quantization. While discussing PCM, we mentioned quantization error. It appears as noise superimposed on the useful signal.

3.6.1 Intermodulation Noise

Non-linear distortion is caused by the non-linear input/output characteristics. For example, if the input level is very high, an amplifier may be driven into saturation and its output level may no longer remain proportional to the input level. The output can be expressed as a series of powers of the input for a non-linear characteristic.

$$y(t) = a_1x(t) + a_2x^2(t) + a_3x^3(t) \dots$$

If the input is a sinusoidal signal of frequency f , the output of a non-linear stage contains, in addition to the frequency f , its multiples $2f$, $3f$, *etc.* If the input consists of several sinusoidal components, which is normally the case, the non-linearity generates intermodulation products in addition to the harmonics. Intermodulation products are the sum and differences of the input frequencies and of their harmonics. For example, if the input contains two frequencies, f_1 , and f_2 , intermodulation products will consist of $f_1 - f_2$, $2f_1 - f_2$, $2f_2 - f_1$, *etc.* Intermodulation products are categorized as second order, third order, and so on. $f_1 - f_2$, are second order intermodulation products, $2f_1 - f_2$ and $2f_2 - f_1$ are third order intermodulation products.

Figure 3.27 depicts a qualitative picture of the output spectrum when the input signal

has a frequency band from f_1 to f_2 . Note that the second order harmonics and second order intermodulation product can be removed by filtering the output signal, but some of the third order intermodulation products occupy the same frequency band as the original signal. They cannot be removed by filtering the output and they manifest themselves as noise in the original signal.

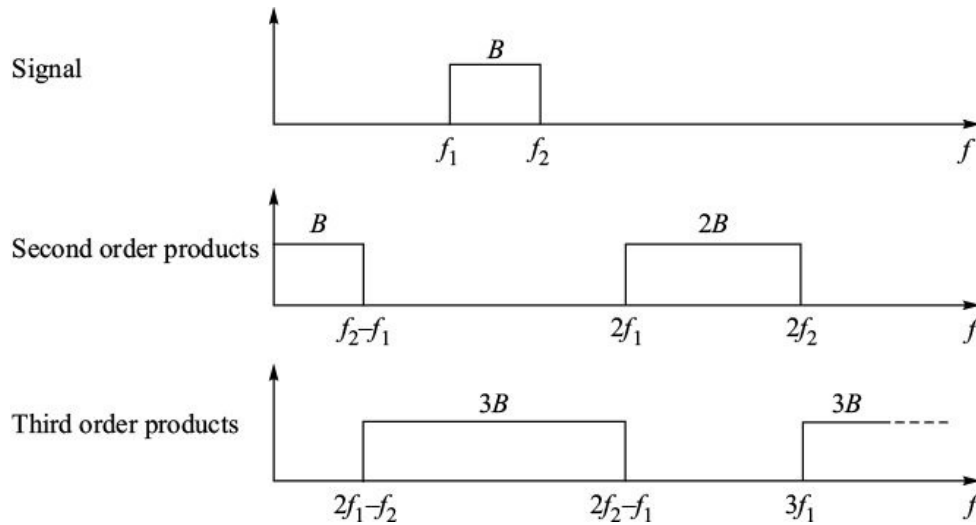


Figure 3.27 Intermodulation products due to non-linear distortion.

Non-linear distortions occur at almost all stages of transmission system except where only passive linear components are involved. Note that intermodulation noise is generated only when signals are actually present and it may also cause crosstalk if its spectral components lie in the frequency band of another channel. This is particularly true for FDM channels. Intermodulation noise can be kept within specified limits by ensuring that signal levels do not exceed the specified level.

EXAMPLE 3.1 An amplifier has the following input/output characteristic: $y(t) = a_1x(t) + a_2x^2(t) + a_3x^3(t)$ Write all the spectral components of the output when two tones $\cos(2pf_1t)$ and $\cos(2pf_2t)$ are applied at the input. $f_1 = 1000$ Hz, $f_2 = 1100$ Hz.

Solution

- | | | | |
|-------------------------------------|---|--|---------|
| 1. Output for the term $a_1x(t)$ | : | $a_1 \cos(2\pi f_1 t) + a_1 \cos(2\pi f_2 t)$ | |
| 2. Frequency components | : | $f_1 = 1000 \text{ Hz}, f_2 = 1100 \text{ Hz}$ | |
| 3. Output for the term $a_2x^2(t)$ | : | $a_2\{\cos(2\pi f_1 t) + \cos(2\pi f_2 t)\}^2$ | |
| | | 4. $= a_2\{\cos^2(2\pi f_1 t) + \cos^2(2\pi f_2 t) + 2 \cos(2\pi f_1 t)$ | |
| | | 5. $\cos(2\pi f_2 t)\}$ | |
| 6. Frequency components | : | 2 f_1 | 2000 Hz |
| | | 7. 2 f_2 | 2200 Hz |
| | | 8. $f_1 + f_2$ | 2100 Hz |
| | | 9. $f_2 - f_1$ | 100 Hz |
| 10. Output for the term $a_3x^3(t)$ | : | $a_3\{\cos(2\pi f_1 t) + \cos(2\pi f_2 t)\}^3$ | |
| | | 11. $a_3\{\cos^3(2\pi f_1 t) + \cos^3(2\pi f_2 t)$ | |
| | | $+ 3\cos^2(2\pi f_1 t) \cos(2\pi f_2 t)$ | |
| | | 12. $+ 3\cos(2\pi f_1 t) \cos^2(2\pi f_2 t)\}$ | |
| 13. Frequency components | : | f_1 | 1000 Hz |
| 14. | | 3 f_1 | 3000 Hz |
| 15. | | f_2 | 1100 Hz |
| 16. | | 3 f_2 | 3300 Hz |
| 17. | | 2 $f_1 + f_2$ | 3100 Hz |
| 18. | | 2 $f_1 - f_2$ | 900 Hz |
| 19. | | 2 $f_2 + f_1$ | 3200 Hz |
| 20. | | 2 $f_2 - f_1$ | 1200 Hz |

3.6.2 Thermal and Shot Noise

All the lossy components and active devices of an electronic circuit generate noise. Lossy components generate thermal noise and active components generate shot noise. Thermal noise constitutes the most important source of noise in telecommunication systems. It is generated by the random motion of electrons in a conductor. Its power spectral density function is flat in the range of frequencies of interest in telecommunications. For this reason, thermal noise is also called 'white noise'. Without going into its mathematical representation and analysis, we will quote the result of the analysis.

"The maximum thermal noise which a resistance R can deliver in the frequency band B Hz at absolute temperature T is given by $P_n = kTB$ watts, where k is the Boltzmann's constant (1.38×10^{-23} joules/ $^\circ\text{K}$)."

The shot noise is generated in the active electronic components due to discrete and random emission of electrons which constitute current in these devices. For

example, shot noise current is generated in a forward biased diode when free electrons from the n -side accelerate towards the p -side. Mean square value of the shot noise current in a forward biased diode is given by $\bar{I}_n^2 = 2q(I + I_S)B$

where q is the electronic charge (1.6×10^{-19} coulomb), I is the forward current in amperes, I_S is the saturation current in amperes, and B is the bandwidth in Hz.

EXAMPLE 3.2 What is the maximum available thermal noise from a resistive termination at ambient temperature of 290°K in the bandwidth 3.1 kHz?

Solution Maximum available thermal noise from the resistive termination is kTB watts. Substituting the values, we get $P_n = 1.38 \times 10^{-23} \times 290 \times 3100$

$$= 1.24 \times 10^{-17} \text{ watt} = -140 + 0.93 = -139.07 \text{ dBm}$$

3.6.3 Psophometric Weighting

The annoying effect of noise on the ear is different at different frequencies. To account for this subjective effect, the background noise is measured using a special voltmeter called a *psophometer*. It incorporates a filter which simulates the sensitivity of the ear to noise at different frequencies. The attenuation characteristic of this filter, called the *psophometric curve*, has been standardized by ITU-T Recommendation P. 53. Noise measured using a psophometric filter is termed as weighted noise and is expressed in dBmp. When a noise with flat power spectral density is measured in the frequency band 300–3400 Hz, the weighted noise level is 2.5 dB less than the unweighted noise level. Thus flat noise level of -40 dBm when measured using psophometric filter will be -42.5 dBmp.

3.6.4 Signal to Noise Ratio

The quality of a signal is determined by its level with respect to the level of the noise which contaminates it. It is expressed as the ratio of signal power (P_s) to noise power (P_n) and is termed the Signal-to-Noise ratio (SNR). If psophometrically weighted noise level is used, the signal-to-noise ratio is called *weighted* SNR. ITU-T has recommended weighted SNR of ≥ 50 dB for speech channels. It includes contribution of all types of noise present in a telecommunication link.

$$\frac{S}{N} = \frac{P_s}{P_n}$$

$\frac{S}{N} = (P_s)_{\text{dBm}} - (P_n)_{\text{dBm}}$ **EXAMPLE 3.3** Calculate the weighted SNR of a voice channel if the signal level is 0 dBm and the noise level is -30 dBm.

Solution

$$\text{Weighted noise level} = -30 - 2.5 = -32.5 \text{ dBm} \quad (S/N)_{\text{weighted}} = 0 - (-32.5) = 32.5 \text{ dB}$$

3.6.5 Companders

Companders are used in the telephone network whenever there is need to improve SNR without actually increasing the signal level. Companders reduce the dynamic range of the telephone signals by using a compressor at the transmitting end and restore it to its original value by using an expander at the receiving end. The word compander is derived from the terms compressor and expander. We will shortly see how the SNR improvement is achieved by reducing the dynamic range.

The usual dynamic range of a telephone signal is from -45 dBm to 5 dBm with the average signal level of -15 dBm. The compressor used in telephone network reduces it to half. It employs a variable gain amplifier. The gain is unity for the nominal input signal level of 0 dBm. For the input level is less than 0 dBm, say -x dBm, the gain is set automatically at x/2 so that the output level becomes -x/2 dBm. Similarly, if the input level is more than 0 dBm, say x dBm, the gain is set at -x/2 so that the output level is x/2 dBm. This is illustrated in Figure 3.28a.

The operation of the expander used at the receiving end is the reverse of the compressor operation. The received signal at 0 dBm is unaffected and passed as it is. The levels above 0 dBm are amplified and below 0 dBm are attenuated to restore them to their original values. For example, if the input to the expander is x dBm, its output will be 2x dBm. If the input is -x dBm, the output will be -2x dBm.

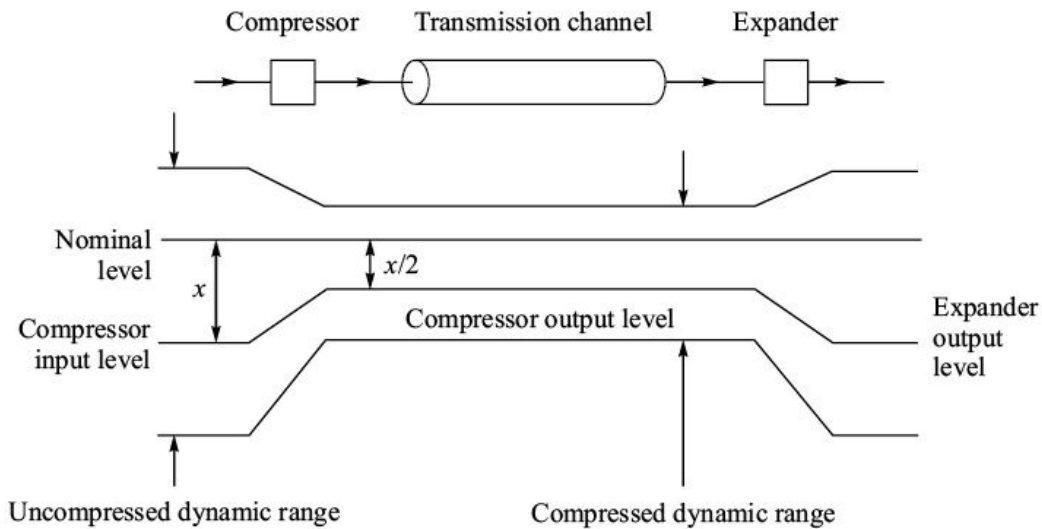
Noise is added to the signal during its transmission through the channel (Figure 3.28b). It is to be noted that it has not passed through the compressor. It is a known fact that noise level is usually much below 0 dBm. Let us assume that it is -P_n dBm. The negative sign has been used to emphasise the fact that the level is below 0 dBm. When the noise passes through the expander circuit, its level is reduced to -2P_n in the absence of any speech signal. Thus, the noise

level is significantly reduced at the receiving end.

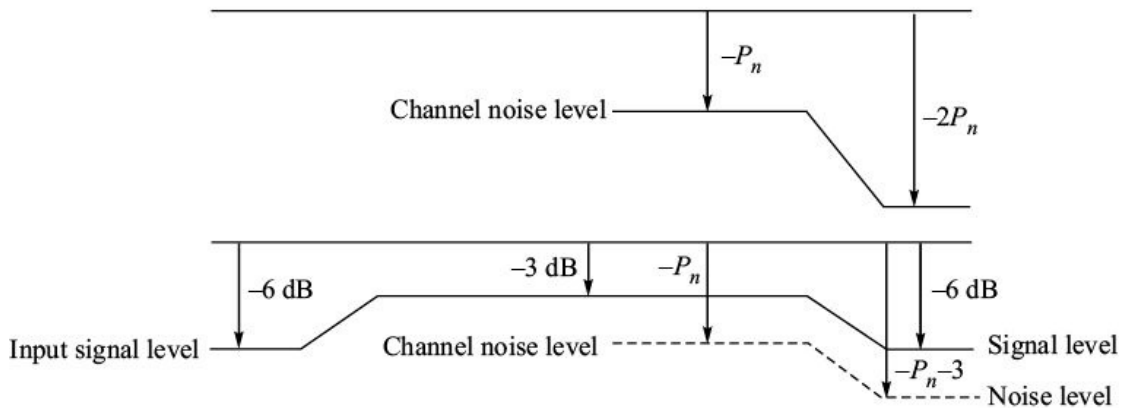
When the speech signal is also present, the noise advantage is not so significant. Let us assume that the signal level is -3 dBm and the noise level is -20 dBm at the expander input. When added together, we get 3 dBm = 0.5 mW

$$-20 \text{ dBm} = 0.01 \text{ mW}$$

$0.5 + 0.01 = 0.51 \text{ mW} = -3 \text{ dBm}$ The expander which would have given the loss of 20 dB had only the noise been present, gives a loss of only 3 dB to restore the signal level. Therefore, the noise level after the expander circuit is -23 dBm, and there is only marginal improvement. Therefore, the compander is more effective in reducing noise levels when the channel is idle.



(a) Dynamic range reduction using compandor



(b) Noise reduction in the received signal

Figure 3.28 Compander.

It has been observed by subjective tests that noise causes more annoyance to the listener when it is present during inter-syllabic pauses and when the channel is idle. In telephone circuits, the speech signal is present on average for 25 per cent of the time in any one direction. The channel is idle for the rest of the time either because the other party is speaking or due to inter-syllabic pauses. Therefore companders can be effective in the telephony channels. Tests have indicated that for speech signals, a compander gives a subjective improvement of 16 dB in SNR.

When a speech channel is used for data transmission, the modems transmit the carrier continuously. Unlike speech signals, there are no inter-syllabic pauses and, therefore, the compander is ineffective. It may, on the other hand, introduce errors because it changes the amplitude of the data carrier. The compander needs to be disabled whenever a speech channel is used for data transmission.

3.7 SIGNAL IMPAIRMENTS IN THE TELEPHONE NETWORK

There are several signal processing stages in the telephone network, *e.g.* switching, filtering, amplification, frequency translation, or quantization, *etc.* At every stage there is some impairment of the signal quality. We have already examined some of the major impairments, *e.g.* linear distortions, noise, *etc.* Other impairments of the signal, described below, are not very critical for transmission of the speech signal but may be the cause of poor bit error rate for a data signal.

3.7.1 Impulse Noise

Impulse noise is characterized by high amplitude peaks (hits) of short duration. It is caused by bad electrical contacts, dial pulses, crosstalk, relay contacts, *etc.* It can also originate from external sources such as power lines and lightning.

Impulse noise is generally not objectionable in voice communication as an impulse hit is never more than 10 ms in duration. Typical duration of the impulse hit is 4 ms (Figure 3.29). It has been empirically determined that an impulse hit will not produce transmission errors in a data signal unless it comes within 6 dB of the signal level.

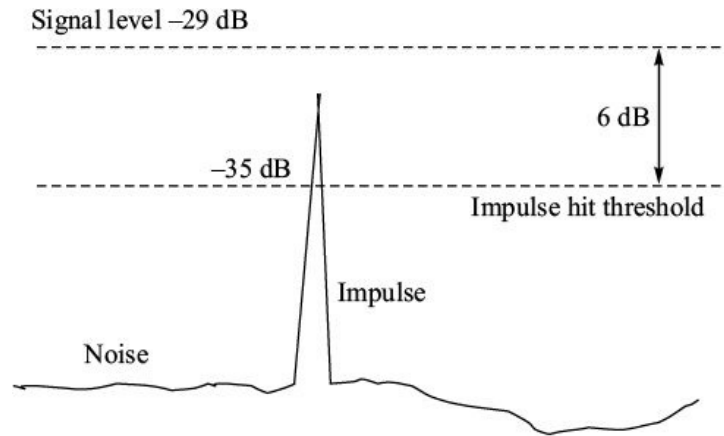


Figure 3.29 Impulse noise.

3.7.2 Gain Hits and Dropouts

A *gain hit* is a sudden random change in the signal level of more than 3 dB and lasting more than 4 ms (Figure 3.30). The signal returns to the original level within 200 ms. Gain hits are due to change in end-to-end gain of the channel which is caused by the transients produced during switch-over of radio channels.

A *dropout* is a decrease in channel gain of more than 12 dB that lasts longer than 4 ms (Figure 3.30). Dropouts are caused by deep fades in the radio channels caused by atmospheric conditions.

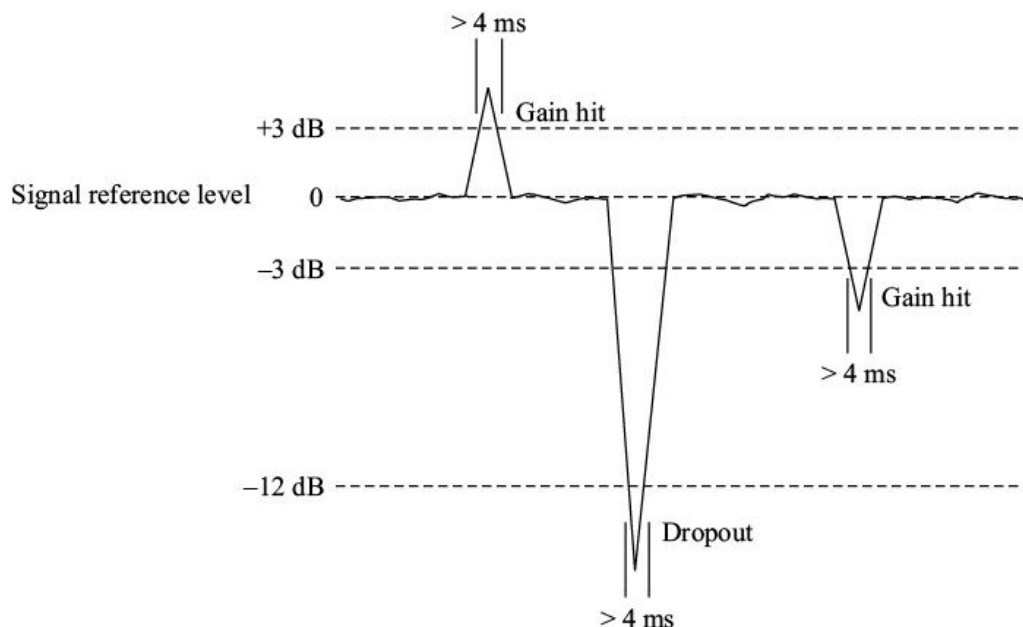


FIGURE 3.30 Gain hit and dropout.

3.7.3 Phase Hits

Phase hits are sudden random changes in the phase of a transmitted signal. The

hits lasting more than 4 ms and greater than 20° peak are recorded for measurement.

3.7.4 Phase Jitter

Phase jitter is a form of incidental phase modulation produced in a tone when transmitted on a speech channel. It is caused by the low frequency ripple in the power supplies of the telecommunication equipment. It is also caused by the jitter present in the carriers used for translating the baseband. Its frequency is generally less than 300 Hz.

3.7.5 Single Frequency Interference

Single frequency interference is the presence of one or more unwanted tones in the speech channel. They are caused by crosstalk and intermodulation.

3.7.6 Frequency Shift

In the frequency division multiplexing equipment, if the carrier frequencies used for translating the frequency bands in the transmit side are not exactly the same as those in the receive side, a tone transmitted on a speech channel will suffer a change in frequency. This change in frequency is termed as *frequency shift*.

3.8 INTEGRATED SERVICES DIGITAL NETWORK (ISDN)

Evolution of the telecommunication network has focussed around voice communications. There were limited requirements for data communications a few decades ago. Data terminal equipment used voice circuits (switched or dedicated) for transmission of data signals. Low speed modems (300 bauds) were used to convert digital signals of data equipment into analog signals of voice band.

If we look at the architecture of today's telecommunication network, analog voice is converted into 64 kbps digital signal in the telephone exchange. The entire transmission path of this signal from originating telephone exchange to terminating telephone exchange is based on PCM technology. The telephone exchange carries out its function by switching 64 kbps digitized voice signals from one time slot of PCM to another. The network, however, offers to its subscribers bandwidth of 300–3400 Hz for analog voice signals.

Integrated Services Digital Network (ISDN) provides digital access to the digital switching technology used in the telephone exchanges. The digital access can be used for voice and data signals.

3.8.1 ISDN Interface

To enable use of ISDN for voice, data, image transmission applications at home and office, two types of access to the network have been defined (Figure 3.31):

- Basic Rate Access (BRA)
- Primary Rate Access (PRA).

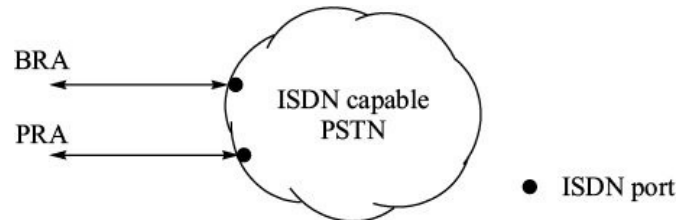


FIGURE 3.31 ISDN interface.

These accesses are defined in terms of three channel types, bearer channel (B), data channel (D), and hybrid channel (H).

B channel. B channel can carry full duplex transmission of digitized voice or data at the rate of 64 kbps.

D channel. D channel is used for signaling during the set-up and clearing of switched connections on B channels. For example, the dialed digits for a telephone connection are sent on D channel. It can be used for carrying user data as well under certain circumstances. Its data rate is 16 kbps for BRA and 64 kbps for PRA.

H channel. There are two types of H channels, H0 (384 kbps) and H1 (1920 kbps). These are used for applications requiring high data rate. Videoconferencing is one such application.

Basic rate Access (BRA) is defined as '2B + D'. It has useable bandwidth of 144 kbps (128 + 16). Multiple devices (up to eight) can be terminated on BRA. The two B channels can be used for one 128 kbps connection or two independent connections on the two channels. The signaling information for the two channels is sent onto D channel. The two B channels and the D channels are multiplexed with overhead bits in form of a frame structure described later. The overall bit rate of BRA is 192 kbps.

Primary rate Access (PRA) is defined as '30B + D' or 'H1 + D'. It has useable bandwidth of 1920 kbps (30B). Reduced bandwidth capacity on PRA is also

available as 'H0 + D' which offers useable bandwidth of 384 kbps (6B). The D channel is of 64 kbps and it can be used only for signaling.

PCM frame as described earlier is used for PRA. D channel occupies TS 16. TS1 to TS15 and TS17 to TS31 are used by 30B channels. 1920 kbps bandwidth can be used in the following manner:

- B channels can be used separately for setting up 30 connections.
- The entire bandwidth of 1920 kbps can be as one H1 channel.
- B and H0 channels can be used in mixed mode $nB + mH0$.

3.8.2 ISDN Devices

ISDN devices can be classified into three types (Figure 3.32):

- Terminal equipment, TE1, and TE2
- Terminal adapter, TA
- Network terminating device, NT1.

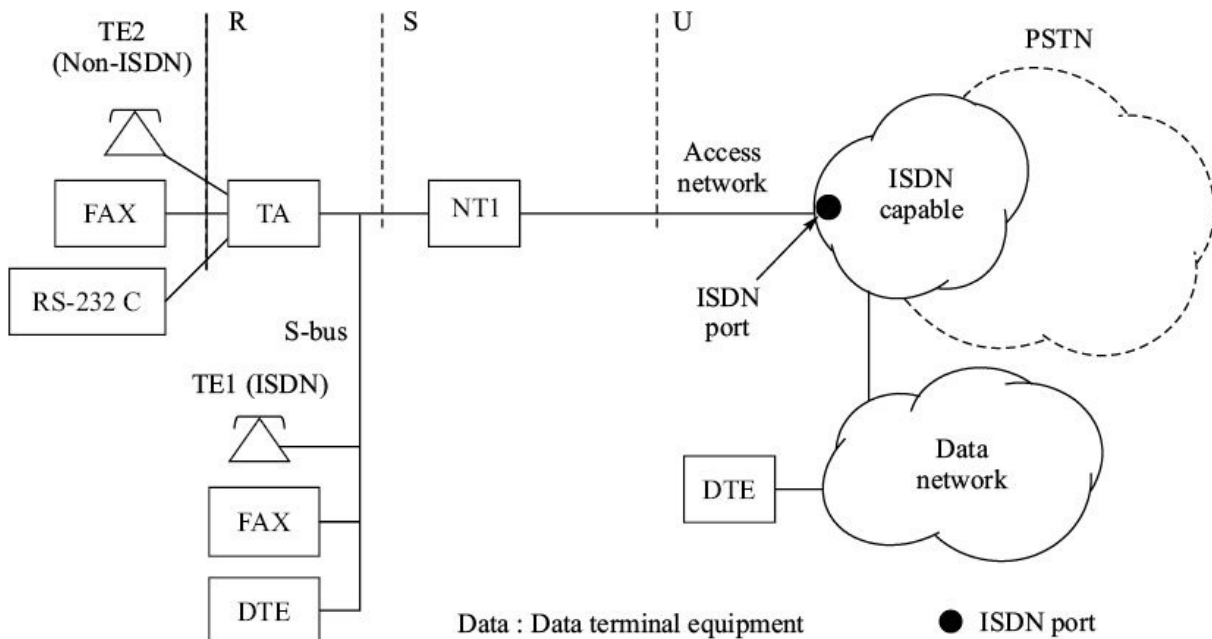


FIGURE 3.32 ISDN devices and interfaces.

TE1. ISDN terminal devices (TE1) include digital telephone instruments, FAX (Group 4), or data terminal equipment. All these devices have S-bus ISDN interface.

TE2. Non-ISDN devices such as ordinary telephone instrument, FAX (Group 3),

modems, *etc.* are referred to as TE2. These devices are connected to ISDN S-bus interface through a terminal adaptor (TA).

TA. Terminal adapter (TA) is another ISDN device that acts as intermediary device for non-ISDN terminal devices (TE2). It converts non-ISDN interface of these devices to ISDN interface. Some versions of TA have RS-232C serial port also.

NT1. Network terminating device (NT1) connects the user ISDN equipment (TA or TE1) to the access network which is usually a balanced copper pair terminated in the exchange on the ISDN port. NT1 serves several purposes.

- The line side (exchange side) of NT1 is 2-wire and the subscriber side (TE side) is 4-wire. NT1 incorporates hybrid to carry out 2-wire/4-wire conversion and provides proper electrical interface (signal levels, line codes, impedances).
- Multiple simultaneous connections can be established on an ISDN port. NT1 multiplexes the information.

NT2. NT2 may be required when multiple devices share a PRA. For example, an ISDN PABX (Private Automatic Branch Exchange) can act as an NT2. NT2 is connected to the terminal devices on one side and to NT1 on the other side (Figure 3.33). The interface towards TE depends on the device that is connected to NT2.

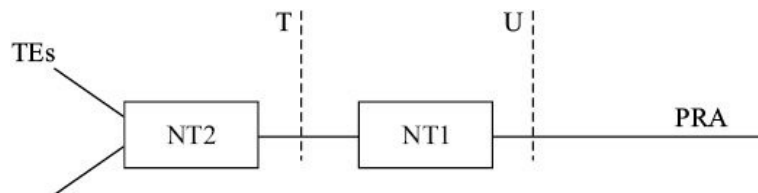


FIGURE 3.33 NT2.

NT2 performs several other functions in addition to providing proper interfaces towards NT1 and TEs. It carries out layer-2 and layer-3 functions as well. These functions pertain to error control, flow control, routing, *etc.* We will describe the concept of layers in Chapter 6.

3.8.3 Reference Interfaces

ITU-T has defined standard interfaces between adjacent devices. The standard interfaces are called *reference points* and are designated as R, S, T, and U

(Figures 3.32 and 3.33).

R interface. R interface is for non-ISDN devices and is therefore not defined in ISDN. It

can be RS-232C, V or X series of ITU-T standards, or ordinary telephone interface with two wires.

S interface. S interface is a 4-wire balanced bus to which up to eight ISDN terminals can be attached. The line code used is pseudo-ternary that we learnt in Chapter 1. The physical connector for S interface on terminals and NT1 is 8-pin RJ-45 modular connector.

U interface. The U interface is the local copper pair of the access network. The same pair is used for full duplex transmission of digital signals. The line code is 2B1Q for BRA and

HDB-3 for PRA. Since NT1 has 2-wire interface on one side and 4-wire interface (S interface) on other side, a hybrid is built into it. Hybrid has associated echo problems and therefore an echo canceller is also provided (Figure 3.34).

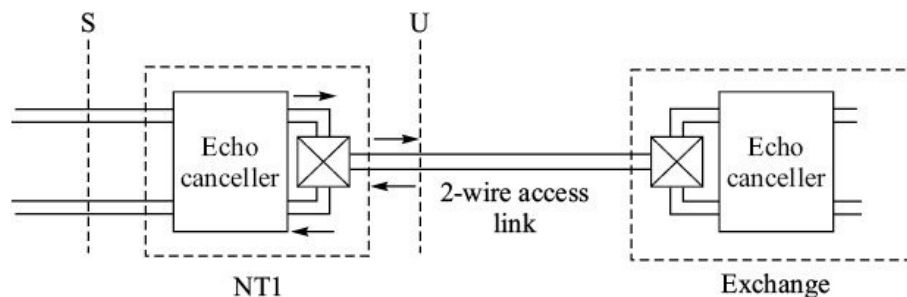


FIGURE 3.34 Hybrids and echo cancellation in NT1.

T interface. T interface is applicable only to PRA. T interface is between NT1 and NT2 (Figure 3.33).

3.8.4 BRA Frame Structure

Basic Rate Access (BRA) has useable bandwidth of 144 kbps (2B + D). The two B channels and the D channels are multiplexed with overhead bits in the form of a frame structure. The overall bit rate of BRA becomes 192 kbps. The overhead bits include framing bits, DC balancing bits, and other bits. We will not go into details of the overhead bits. Figure 3.35 shows the basic structure of the frame.

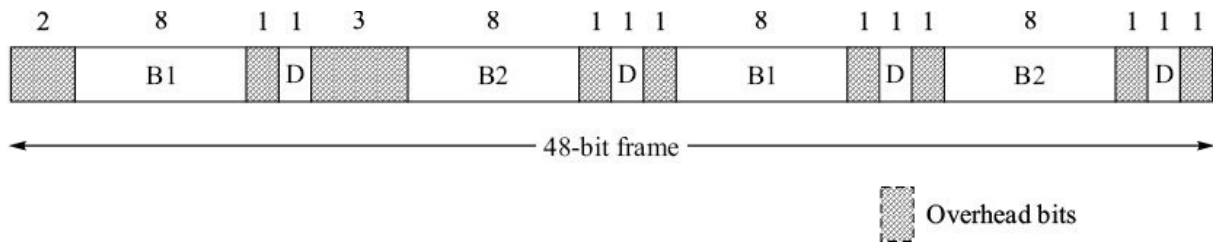


Figure 3.35 Multiplexed 2B + D frame structure.

The 48-bit frame consists of:

- 16 bits of B1 channel
- 16 bits of B2 channel
- 4 bits of D channel
- 12 overhead bits for framing, DC balancing, and other functions.
- The frame is transmitted in 250 msec, which results in the following bit rates:
- Each B channel :.....16/250 msec = 64 kbps
- D channel :.....4/250 msec = 16 kbps
- Overhead :.....12/250 msec = 48 kbps
- Overall bit rate :.....48/250 msec = 192 kbps

3.8.5 Access Mechanism for D Channel in BRA

As mentioned in section 3.8.3, up to eight devices can be terminated on S bus. These devices share the two B channels and one D channel. A device can send data on B channel after the end-to-end connection has been established by the exchange. Therefore, there is no contention for the B channels.

The D channel is shared for signaling purposes by all the eight devices on the S bus. These devices contend for access to the D channel. The contention process is explained below:

- When a TE is idle, it sends all 1s in the D bit positions to the NT.
- When the D channel is idle, the NT sends continuous stream of 1s on the D channel in the direction towards TEs. When D channel is being used there cannot be more than 6 consecutive 1s.⁴
- A device wanting to use B channel, sends an indication to the NT by setting a D bit to zero. After setting the D bit to zero, the device waits for the next D bit from the NT.

- The NT echoes back the received D bit (= 0) in the next available D bit position.
- The echoed zero in the D bit position indicates to all the devices that one of the devices has reserved the D channel for its use.

3.8.6 Data Transmission Mechanisms of ISDN

We have broadly described the physical aspects of ISDN, *i.e.* bits rates, line codes, access mechanisms, various physical interfaces and devices. The data transmission mechanisms of ISDN, frame structure for error, and flow control require understanding of data link protocols, HDLC (High Level Data Link Control) in particular. Therefore, we will defer further discussion on ISDN for Chapter 9, Data Link Protocols.

3.9 DATA COMMUNICATIONS ON TELEPHONE NETWORK

Having considered the characteristics of various transmission media, the required channel characteristics for data transmission, and the services offered by the telephone network, we can now examine how the telephone network can support data communications. Data networks consist of data packet switching devices (e.g. routers) and data terminal equipment (DTE). Computer is one example of a DTE. The existing telephone network infrastructure can be used to integrate these devices.

The services offered by the telephone network, that can be used for data communications are summarized below.

- 300–3400 Hz voice channel bandwidth (switched or point-to-point).
- ISDN services (BRA and PRA).
- n 64 kbps, n E1, n STM-1 digital point-to-point digital links.

3.9.1 300–3400 Hz Voice Channel Bandwidth

300–3400 Hz voice channel, whether switched or a dedicated link through the telephone network, requires a device that converts the digital signals into analog signals using modulation techniques. Such a device is called *modem* (modulator-demodulator). A pair of these devices is always required, one at the each end of

the link (Figure 3.36).



FIGURE 3.36 Modems.

Modems have capability to dial a telephone number if a switched connection service is used. These modes are connected to the telephone exchange one a pair of wires. We discuss modems in detail in the next chapter.

3.9.2 ISDN Services

As discussed in the last section, ISDN provides switched connectivity at bit rates up to 2 Mbps. Instead of a modem we need an NT for accessing ISDN service (Figure 3.32). It is possible for an ISDN enabled telephone exchange to interconnect directly to a data network. BRA and PRA are extensively used for accessing data networks (e.g. Internet).

3.9.3 Digital Point-to-Point Links

The telephone network offers point-to-point digital links of bit rates ranging from 64 kbps to n STM-1. The links having bit rates in the lower range (64 kbps to n E1) are usually for accessing a data network device (e.g. a router) and the links having bit rates in the higher range (E1 to n STM-1) are used for interconnecting the routers (Figure 3.37). The telephone network links act purely as a transport medium for carrying data signals.

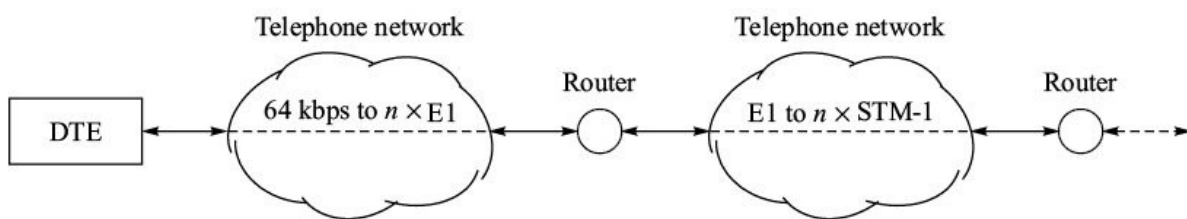


FIGURE 3.37 Point-to-point dedicated links for data networks.

3.9.4 ITU-T Recommendations for Voice Band Leased Circuits

ITU-T recommendations M.1020 and M.1025 are for the point-to-point voice band leased circuits to be used for data transmission. These recommendations specify the transmission characteristics of the circuits.

ITU-T M.1020 recommendation. This recommendation is for leased circuits intended to be used for data transmission using modems that are not equipped with the equalizers. The important parameters of the recommendations are:

- Nominal overall loss : ≤ 28 dB at 800 Hz
- Attenuation distortion : Limits as shown in Figure 3.38a
- Group delay distortion : Limits as shown in Figure 3.38b
- Random circuit noise : ≤ -38 dBm0 for leased circuit $> 10,000$ km
- Impulse noise > -21 dBm0 : ≤ 18 impulses in 15 minutes
- Gain hits $> \pm 2$ dB : 10 in 15 minutes
- Frequency shift : $\leq \pm 5$ Hz
- Non-linear distortion measured with 700 Hz tone at -15 dBm0 : Level of the received harmonics should be 25 dB below the received level of 700 Hz.

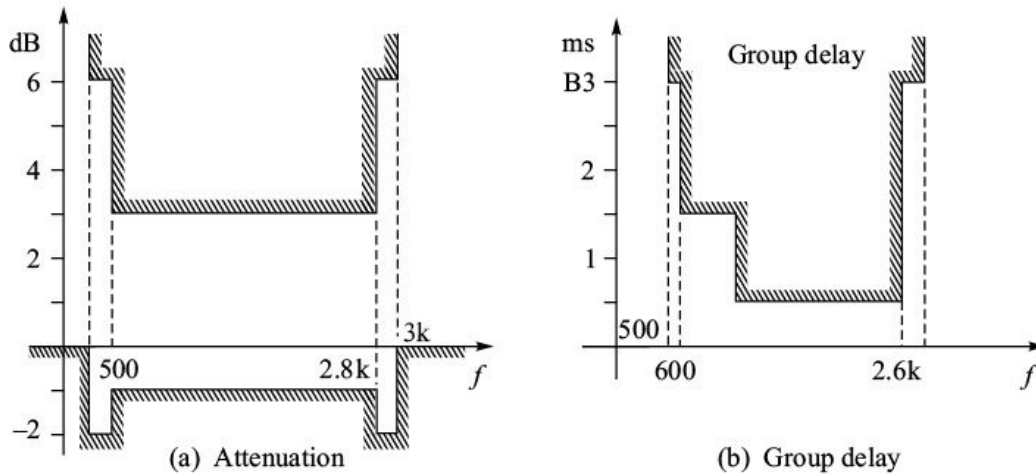


Figure 3.38 ITU-T M.1020 recommendation for attenuation and group delay.

ITU-T M.1025 recommendation. This recommendation is for leased circuits intended to be used for data transmission using modems equipped with equalizers. The limits for all the parameters are the same as M.1020 except those for attenuation and group delay distortions which are more liberal. The figures indicated below are relative to those at 800 Hz:

- Attenuation distortion:

300–500 Hz	12 dB to –2 dB
500–2800 Hz	8 dB to –2 dB
2800–3000 Hz	12 dB to –2 dB
- Group delay distortion:

500–1000 Hz	≤ 3 ms
1000–2600 Hz	≤ 1.5 ms
2600–2800 Hz	≤ 3 ms

SUMMARY

Telecommunication network is designed primarily for carrying voice signals. It consists of:

- access network that connects the subscribers to the telephone exchanges,
- hierarchy of telephone exchanges that switch the voice channels, and
- trunk network that interconnects telephone exchanges.

The available bandwidth of a voice channel is 300 Hz to 3400 Hz on a 2-wire circuit. Digital electronic telephone exchanges convert the analog voice into 64 kbps digital signals and use digital switching principles. In the trunk network, the voice channels are multiplexed using Time Division Multiplexing (TDM). 64 kbps voice channels are multiplexed to a 2048 kbps digital stream called E1. An E1 frame contains 32 time slots, each carrying a 64-kbps channel. E1s are further multiplexed in a hierarchical manner using Synchronous Digital Hierarchy (SDH). In SDH, the bit rates are 155.52 Mbps (STM-1), 622.08 Mbps (STM-4), 2488.32 (SDH-16), and 9953.28 Mbps (STM-64).

The voice channels of telephone network are not clear channels. They incorporate devices like companders to improve signal to noise ratio and echo suppressors to counteract the effect of propagation delay that causes echo.

Integrated Services Digital Network (ISDN) provides direct digital access to the digital switching technology used in the telephone exchanges. The digital access can be used for voice and data signals. ISDN basic rate access (BRA) offers useable bit rate of 144 kbps (2B + D). Multiple devices (up to eight) can be terminated on BRA. Primary Rate Access (PRA) has useable bit rate of 1920 kbps (30B).

Telephone network can offer the following services for data communications:

- Switched or point-point 300–3400 Hz bandwidth.
- Switched BRA (144 kHz) and PRA (2048 kbps) services of ISDN.
- Point-to-point bulk bandwidth (E1 to STM-64) of its trunk network.

EXERCISES

1. If a telephone network had no switches, and every subscriber was connected to every other subscriber, how many lines are required for ten subscribers?
2. A signal having three frequency components f_1 , f_2 and f_3 of equal amplitudes is applied to an amplifier having input-output characteristic defined by

$$y(t) = a_1x(t) + a_2x^2(t) + a_3x^3(t)$$
 Write down all the frequency components of the output. If $f_1 = 1000$ Hz, $f_2 = 2000$ Hz, and $f_3 = 3000$ Hz, are there any intermodulation products in the frequency band 1000–2000 Hz?
3. In a voice channel, the unweighted white noise level is 0.001 mW/kHz. What is the psophometrically weighted noise level in dBm?
4. If noise temperature is 135 K, calculate the thermal noise power for a bandwidth of 36 MHz.
5. Two telephone exchanges of a hypothetical telephone system are interconnected with 100 both way voice channels. The average usage of a telephone is one call of duration 3 minutes in an hour. 10% of the total calls made are between two exchanges. What is the maximum number of telephone connections the telephone system can support?
6. What is the percentage overhead of an E1?
7. How long does a voice sample last on an E1? How long does it last on an STM-1?
8. What bit rate can an E1 support if except the TS0, all the time slots are used for user data?
9. What aggregate bit rate of user data can an STM-1 signal support if it is fully loaded with E1s, each of which carries user data in its 31 time slots?

1 The voice calls that originate and terminate in the same area constitute local traffic.

2 Scrambling is done to ensure that continuous stream of zeroes or ones is avoided. Generating polynomial $x^{43} + 1$ is used for scrambling. We will learn more about scrambling in the next chapter.

3 A scalable system can be upgraded for higher capacity at minimal cost.

4 HDLC protocol that we will study in Chapter 9, ensures that there are not more than six consecutive ones, whatever be the user data pattern.

4

Data Line Devices

In Chapters 2 and 3, we have examined the transmission requirements for data signals and the characteristics of telecommunication media which are primarily designed for voice communications. The first attempts for establishing data networks were based on using the existing telecommunication network for data communications. Intermediary devices were therefore developed for interconnecting the data devices to telecommunication network. These devices also ensured that the transmission characteristics of the data signals matched with the available network characteristics. Modems, DSL equipment, and data multiplexers are three categories of data line devices that are discussed in this chapter.

We begin this chapter by examining various digital modulation methods which are used in modems. Besides modulation and demodulation, there are many additional functions which are performed by the modems. We examine all these functions and familiarize ourselves with the modem terminology. There are number of ITU-T recommendations on the modems. We take a brief look at the features of the ITU-T modems. We next examine the various types of DSL (Digital Subscriber Line) equipment. Asymmetric DSL (ADSL) has found wide spread use and therefore we discuss it in detail. Finally, we move over to data multiplexers and introduce FDM, TDM, and Statistical TDM (STDM) multiplexers.

4.1 DIGITAL MODULATION METHODS

There are three basic types of modulation methods for transmission of digital signals. These methods are based on the three attributes of a sinusoidal signal—amplitude, frequency, and phase. The corresponding modulation methods are called, Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), and

Phase Shift Keying (PSK). In addition, a combination of ASK and PSK is employed at high bit rates. This method is called Quadrature Amplitude Modulation (QAM). We will discuss these modulation methods and later examine their application in the standard modems.

4.1.1 Amplitude Shift Keying (ASK) *Amplitude shift keying is the simplest form of digital modulation. In ASK, the carrier amplitude is multiplied by the binary 1 or 0 (Figure 4.1). The digital input is a unipolar NRZ signal. The amplitude modulated carrier signal can be written as $v(t) = d \sin(2\pi f_c t)$ where f_c is the carrier frequency and d is the data bit variable which can take values 1 or 0, depending on the state of the digital signal.*

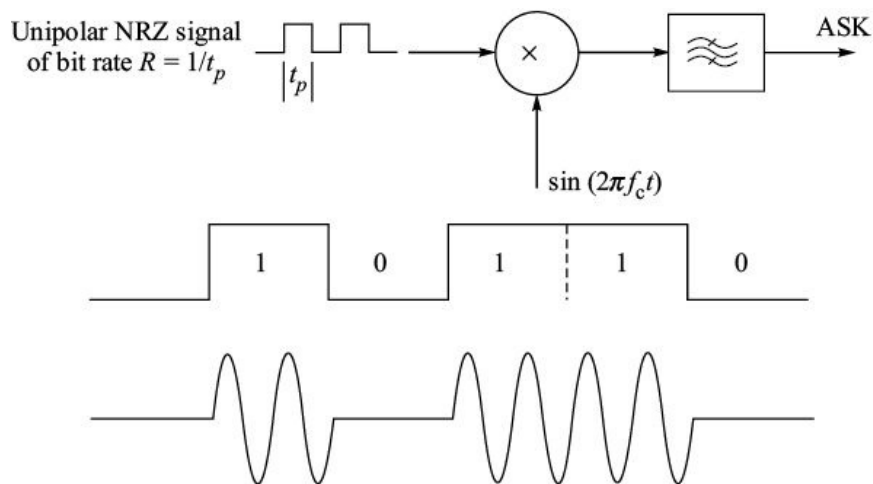


FIGURE 4.1 Amplitude shift keying (ASK).

The frequency spectrum of the ASK signal consists of the carrier frequency with upper and lower side bands (Figure 4.2). For a random unipolar NRZ digital signal having bit rate R , the first zero of the spectrum occurs at R Hz away from the carrier frequency. The transmission bandwidth B of the ASK signal is restricted by using a filter to where r is a factor related to the filter characteristics and its value lies in the range 0–1.

$$B = (1 + r)R$$

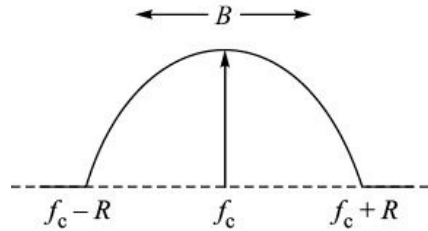


Figure 4.2 Frequency spectrum of ASK signal.

ASK is very sensitive to noise and finds limited application in data transmission. It is used at bit rates less than 100 bps.

4.1.2 Frequency Shift Keying (FSK) In frequency shift keying frequency of the carrier is shifted between two discrete values, one representing binary 1 and the other representing binary 0 (Figure 4.3). The carrier amplitude does not change.

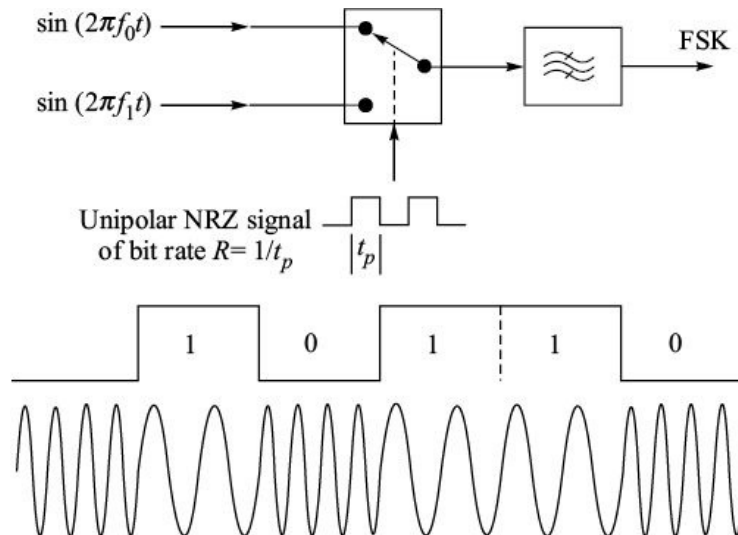


Figure 4.3 Frequency shift keying (FSK).

The FSK signal can be written as

$v(t) = d \sin (2pf_1t) + \bar{a} \sin (2pf_0t)$ where f_1 and f_0 are the frequencies corresponding to binary 1 and 0 respectively and d is the data signal variable as before. \bar{a} is inverse of d . From the above equation, it is obvious that the FSK signal can be considered to be comprising of two ASK signals with carrier frequencies f_1 and f_0 . Therefore the frequency spectrum of the FSK signal can be drawn as shown in Figure 4.4.

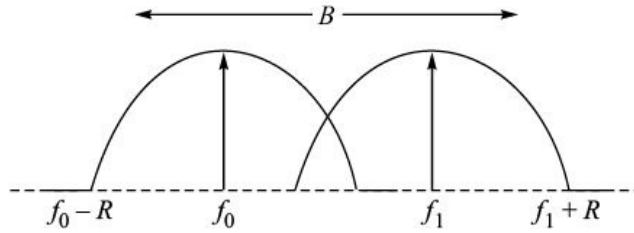


FIGURE 4.4 Frequency spectrum of FSK signal.

To get an estimate of the bandwidth B for the FSK signal, we need to include the separation between f_1 and f_0 and significant portions of the upper side band of carrier f_1 and of the lower side band of carrier f_0 .

$$B = |f_1 - f_0| + (1 + r)R$$

The separation between f_1 and f_0 is kept at least $2R/3$. ITU-T Recommendation V.23 specifies $f_1 = 2100$ Hz and $f_0 = 1300$ Hz for bit rate of 1200 bps. FSK is not very efficient in its use of the available transmission channel bandwidth. It is relatively simple to implement. It is used extensively in low speed modems having bit rates below 1200 bps.

4.1.3 Phase Shift Keying (PSK)

Phase shift keying is the most efficient of the three modulation methods and is used for high bit rates. In PSK, phase of the carrier is modulated to represent the binary values. Figure 4.5 shows the simplest form of PSK called Binary PSK (BPSK). The carrier phase is changed between 0 and π by the polar digital signal. Binary states 1 and 0 are represented by the negative and positive polarities of the digital signal in Figure 4.5.

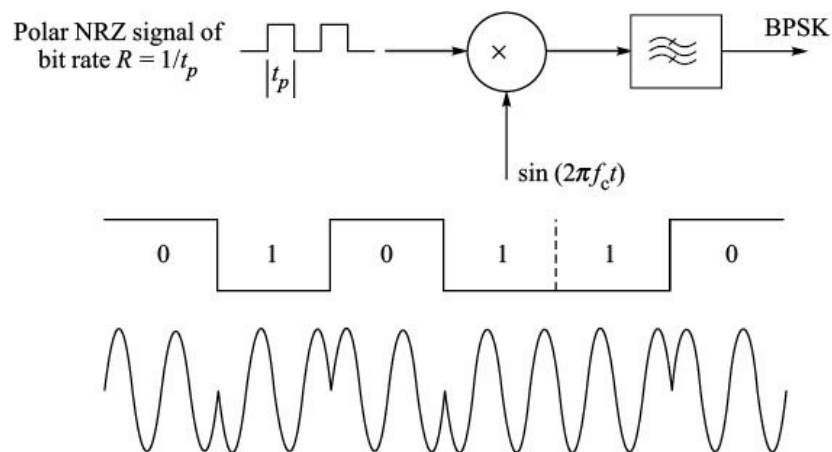


Figure 4.5 Binary phase shift keying (BPSK).

The BPSK signal can be written as

$$v(t) = \sin (2pf_{c}t) \text{ for binary 0}$$

$$v(t) = -\sin (2pf_{c}t) = \sin (2p f_{c}t + p) \text{ for binary 1}$$

In other words,

$$v(t) = d \sin (2p f_{c}t), \text{ where } d = 1$$

Expression for BPSK signal is very similar to the expression for ASK signal except that the data variable d takes the values ± 1 . The carrier gets suppressed due to polar modulation signal. The frequency spectrum of the PSK signal for random NRZ digital modulating signal is shown in Figure 4.6.

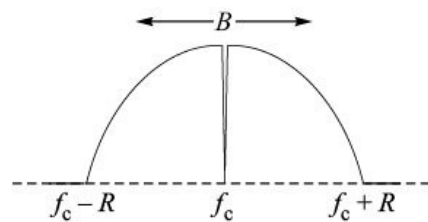


FIGURE 4.6 Frequency spectrum of BPSK signal.

The estimate of bandwidth B of the BPSK signal can be obtained as before.

$$B = (1 + r)R, \text{ where } 0 < r < 1$$

The value of parameter r depends on transmission filter characteristics. The BPSK signal requires smaller bandwidth as compared to the FSK signal.

4.2 MULTILEVEL MODULATION

We have so far considered the schemes for two-level modulation, *i.e.* the bit rate and the baud rate are the same. Due to the limited bandwidth of the telephone voice channel, the maximum bit rate which can be achieved using any of the above two-level modulation methods does not meet the requirements of the data processing community. Keeping the baud rate the same, the bit rate can be increased using multilevel modulation as we saw in Chapter 1. The data bits are divided into groups of two or more bits and each group is assigned a specific state of the sinusoidal signal. Any of the three attributes of the signal, amplitude, frequency or the phase, can be used to represent the groups of the data bits. ASK being very sensitive to noise and FSK being very expensive from the bandwidth point of view, multilevel modulation is used only with PSK. Instead of two phase states, 0 and p as in BPSK, the carrier is allowed to have four or more

phase states (Figure 4.7).

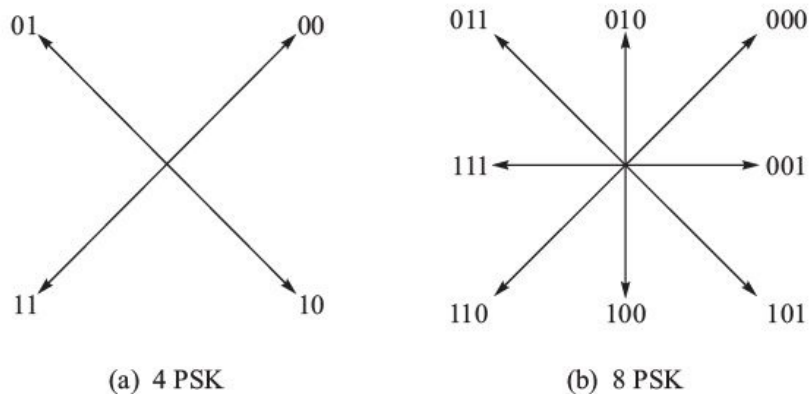


Figure 4.7 Multilevel PSK.

In the four-state PSK (or simply 4 PSK), two bits are associated with each phase state. Therefore, the bit rate is twice the baud rate. Similarly, three bits are associated with each phase state in 8 PSK and, therefore, its bit rate is three times the baud rate.

4.2.1 Gray Code

A very important point to be noted in Figure 4.7 is the sequence of assignment of codes to the phase states. The codes of adjacent states differ in only one bit position and the resulting sequence is not in usual binary count. This sequence is called *Gray code*. When a PSK signal is transmitted, the received signal does not remain in the precisely defined phase states shown in Figure 4.7 due to noise and phase distortion. For example, 000 state may be received at 15° instead of 45° (Figure 4.7b). The receiver has to take a decision whether the received code is 000 or 001. As the received phase is nearer to the adjacent code 001, it decodes the phase as 001. It makes a mistake but introduces only one bit error because the adjacent codes differ in only one bit position. Had we used the binary count sequence, the receiver would have introduced three errors by decoding the received phase as 111. Thus Gray code reduces the impact of phase distortion on the bit error rate.

4.2.2 4 PSK Modulator

Figure 4.8 shows the schematic of a 4 PSK modulator. It consists of two BPSK modulators. The carrier frequency of one of the modulators is phase shifted by $\pi/2$ radians. The data bits are taken in the groups of two bits called *dibits* (Table 4.1) and two bipolar digital signals are generated, one from the first bit of the

dibits and the other from the second bit of the dibits and applied to the BPSK modulators. Outputs of the modulators are added to generate 4 PSK output. Since the carriers are in phase quadrature, vector addition of the respective phasors of the two modulated carriers is carried out to get the resultant phase as shown in Figure 4.8.

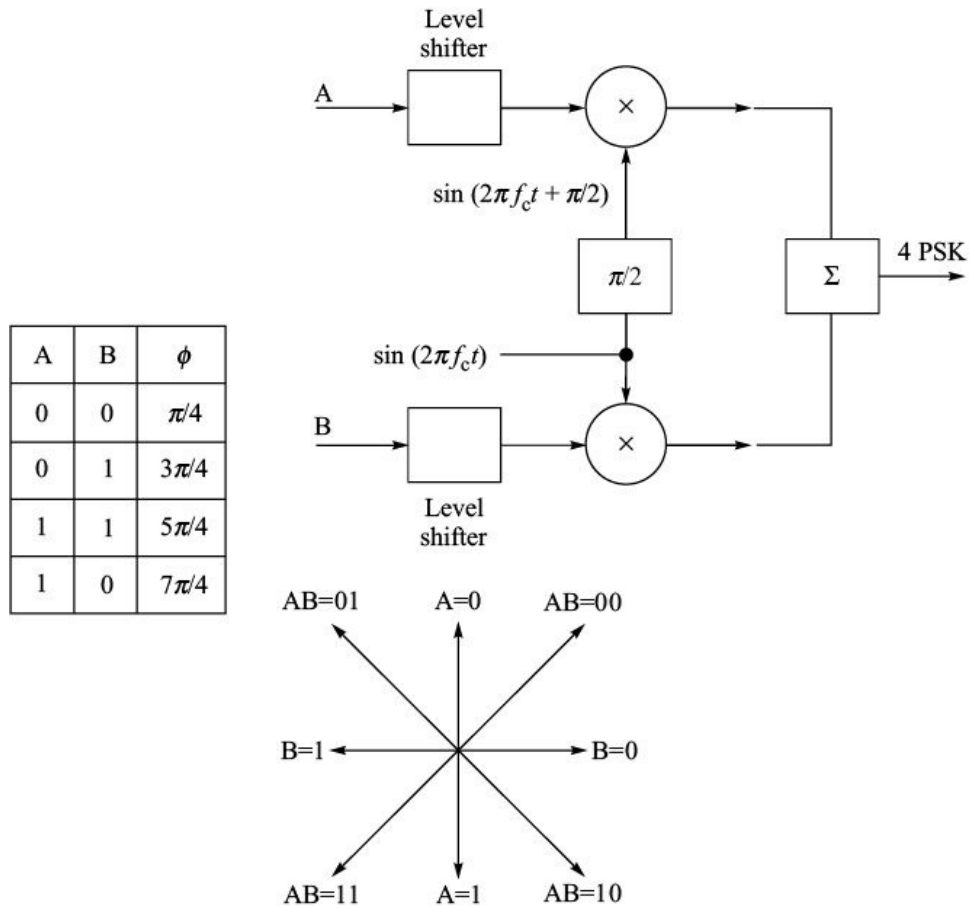


Figure 4.8 4 PSK modulator.

TABLE 4.1 Formation of Dibits	
Data	1 0 0 1 0 1 1 1 0 0
Dibits	A 1 0 0 1 0
	B 0 1 1 1 0

4.2.3 4 PSK Demodulator Figure 4.9 shows a 4 PSK demodulator. The reference carrier is

recovered from the received modulated carrier. As in the modulator, a $p/2$ phase shifted carrier is also generated. When these carriers are multiplied with the received signal, we get $\sin(2p f_c t + f) \sin(2p f_c t) = \frac{1}{2} \cos f - \frac{1}{2} \cos(4p f_c t + f)$ and $\sin(2p f_c t + f) \sin(2p f_c t + p/2) = \frac{1}{2} \cos(f - p/2) - \frac{1}{2} \cos(4p f_c t + f + p/2)$ where f is the phase of the received carrier.

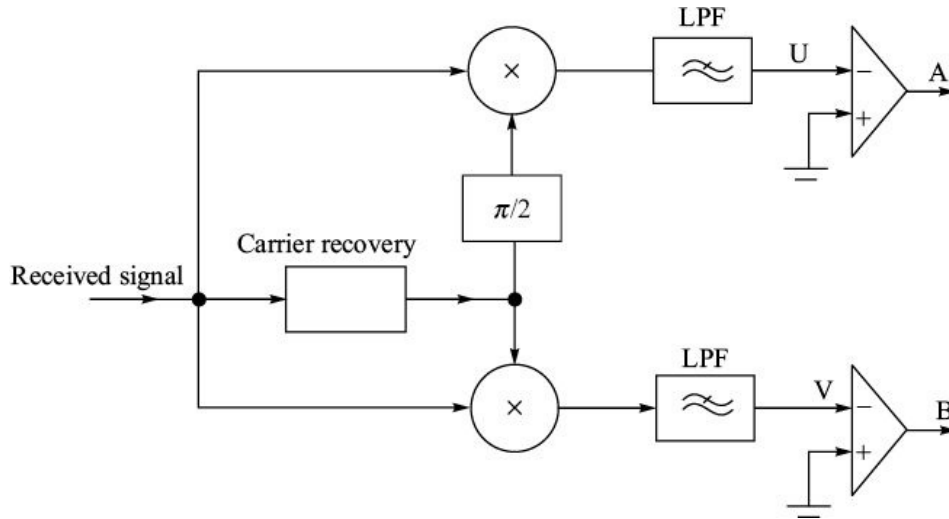


FIGURE 4.9 4 PSK demodulator.

The outputs of the multipliers are passed through low pass filters to remove the $2f_c$ frequency component and applied to the level comparators which generate the dibits. Table 4.2 gives the outputs of low pass filters for various values of input phase f .

In the above demodulation method, we have assumed availability of the phase coherent carrier at the receiving end, *i.e.* the recovered carrier at the receiving end is in phase with the carrier at the transmitting end. But it is quite possible that the phase of the recovered carrier is out by $p/2$ or p . If this happens, the demodulator operation will be upset as demonstrated in Example 4.1.

f	U	V	A	B
	0.35	0.35		
	0.35	-0.35		
$p/4$	-0.35	-0.35	0	0
$3p/4$			0	1
$5p/4$			1	1
$7p/4$	-0.35	0.35	1	0

EXAMPLE 4.1

1. What are the phase states of the carrier when the bit stream 1 0 1 1 1 0 0 1 0 0 is applied to 4 PSK modulator shown in Figure 4.8.
2. If the recovered carrier at the demodulator is out of phase by p radians, what will be the output when the above 4 PSK carrier is applied to the demodulator shown in Figure 4.9?

Solution

1. Modulator input	1	0	1	1	1	0	0	1	0	0
Phase states of the transmitted carrier	$7\pi/4$		$5\pi/4$		$7\pi/4$		$3\pi/4$		$\pi/4$	
2. Relative phase with respect to the recovered carrier	$3\pi/4$		$\pi/4$		$3\pi/4$		$7\pi/4$		$5\pi/4$	
Output of the demodulator (Table 4.2)	0	1	0	0	0	1	1	0	1	1

4.3 DIFFERENTIAL PSK

The problem of generating the coherent carrier with at the receiving end can be circumvented by encoding the digital information as the phase change rather than as the absolute phase. This modulation scheme is called *differential PSK*. If f_{t-1} is the previous phase state and f_t is the new phase state of the carrier when data bits modulate the carrier, the phase change is defined as $f = f_t - f_{t-1}$

f is coded to represent the data bits. The phase space diagrams of Figure 4.7 are still applicable for 4 differential PSK and 8 differential PSK, but now they represent phase change rather than the absolute phase states. For demodulating the differential PSK signal, the carrier phase variations are detected. The absolute value of the carrier phase is no longer important.

4.3.1 Differential BPSK

Differential BPSK modulator is implemented using an encoder before a BPSK modulator (Figure 4.10). The encoder logic is so designed that the desired phase changes are obtained at the modulator output.

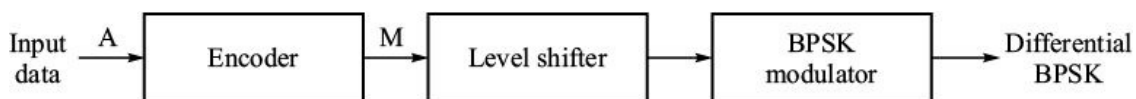


Figure 4.10 Differential BPSK modulator.

Table 4.3 shows the relation between the input data bits and the phase states of the carrier at the modulator output. Knowing that the carrier phase is 0 for binary 0 and p for binary 1 at the modulator input (M), we can write the encoder logic as shown in Table 4.3. It is easily implemented using a JK flip flop in the toggle mode.

TABLE 4.3 Encoder Logic of Differential BPSK Modulator					
A	f	f_{t-1}	f_t	M_{t-1}	M_t
0	0	0	0	0	0
0	0	p	p	1	1
1	p	0	p	0	1
1	p	p	0	1	0

EXAMPLE 4.2 Write the phase states of the differential BPSK carrier for input data stream 100110101. The starting phase of the carrier can be taken as 0.

Solution

A	1	0	0	1	1	0	1	0	1
f	p	0	0	p	p	0	p	0	p
f	0	p	p	0	p	p	0	0	p

Figure 4.11 shows the demodulation scheme for differential BPSK signal. The received signal is delayed by one bit and multiplied by the received signal. In other words, the carrier phase states of the adjacent bits are multiplied.

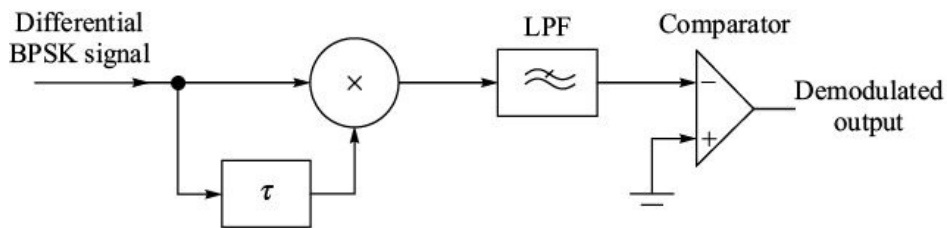


FIGURE 4.11 Differential BPSK demodulator.

$\sin^2 (2pf_c t) = \sin^2 (2pf_c t + p) = \frac{1}{2} - \frac{1}{2} \cos (4pf_c t)$
 $\sin (2pf_c t) \sin (2pf_c t + p) = -\frac{1}{2} + \frac{1}{2} \cos (4pf_c t)$
 The low-pass filter allows only the DC component to pass through. Adjacent phase states may be in phase or p out of phase. If they are in phase, the filtered output of the

multiplier is positive and if they are out of phase, the output is negative. Thus polarity of the signal at the filter output reflects the phase change. The comparator generates the demodulated data signal.

The differential demodulator does not require phase coherent carrier for demodulation. Also, note that there is no decoder corresponding to the encoder in the modulator. If a phase-coherent demodulator is used in place of the differential demodulator, a decoder will be required at the output of the demodulator.

4.3.2 Differential 4 PSK

Just like differential BPSK modulator, differential 4 PSK modulator can also be implemented using an encoder before a 4 PSK modulator as shown in Figure 4.12.

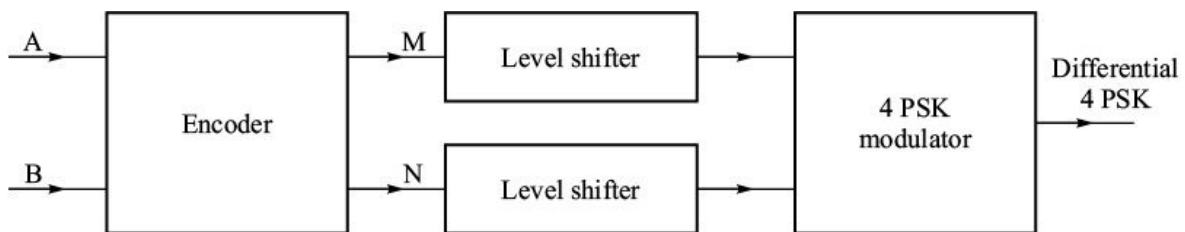


FIGURE 4.12 Differential 4 PSK modulator.

The encoder logic is so designed that its outputs M and N modulate the carrier to produce the required phase changes in the carrier. Table 4.4a shows the relation between the input dibit AB and the phase changes of the modulated carrier. This modulation scheme has been standardized in ITU-T recommendation V.26. Table 4.4b shows the relation between MN bits and the corresponding phase of the modulated carrier. Table 4.4c gives the encoder logic derived from Tables 4.4a and 4.4b. From Table 4.4c, it can be shown that $M_t =$

$$A \times B + \bar{A} \times B \times P + A \times \bar{B} \times \bar{P}$$

$$N_t = A \times B + \bar{A} \times B \times \bar{P} + A \times \bar{B} \times P$$

$$P = M_{t-1} \times \bar{N}_{t-1} + \bar{M}_{t-1} \times N_{t-1}$$

A	B	f
0	0	0
0	1	$p/2$

1	1	p
1	0	$3p/2$

TABLE 4.4(b) Encoder Logic of Differential 4 PSK Modulator				
	$AB = 00, f = 0$	$AB = 01, f = p/2$	$AB = 11, f = p$	$AB = 10, f = 3p/2$
$f_{t-1} (MN)_{t-1}$	$f_t (MN)_t$	$f_t (MN)_t$	$f_t (MN)_t$	$f_t (MN)_t$
$p/4(00) 3p/4(01)$ $5p/4(11) 7p/4(10)$	$p/4(00) 3p/4(01)$ $5p/4(11) 7p/4(10)$	$3p/4(01) 5p/4(11)$ $7p/4(10) p/4(00)$	$5p/4(11) 7p/4(10)$ $p/4(00) 3p/4(01)$	$7p/4(10) p/4(00)$ $3p/4(01) 5p/4(11)$

TABLE 4.4(c) Absolute Phase Changes		
M	N	f
0	0	$p/4$
0	1	$3p/4$
1	1	$5p/4$
1	0	$7p/4$

Implementation of encoder using logic gates and JK flip flops is left as an exercise to the reader.

EXAMPLE 4.3 The following bit stream is applied to the differential 4 PSK modulator described in Table 4.4. Write the carrier phase states taking the initial carrier phase as zero.

1 0 1 1 1 1 0 0 0 1

Solution

Bits stream	1	0	1	1	1	1	0	0	0	1
	f	$3p/2$	p	p	0	$p/2$				
	f	$3p/2$	$p/2$	$3p/2$	$3p/2$	0				

4.3.3 16 Quadrature Amplitude Modulation (QAM)

We can generalize the concept of differential phase shift keying to M equally

spaced phase states. The bit rate will become n (baud rate), where n is such that $2^n = M$. This is called M -ary PSK or simply MPSK. The phase states of the MPSK signal are equidistant from the origin and are separated by $2\pi/M$ radians (Figure 4.13). As M is increased, the phase states come closer and result in degraded error rate performance because of the reduced phase detection margin. In practice, differential PSK is used up to $M = 8$.

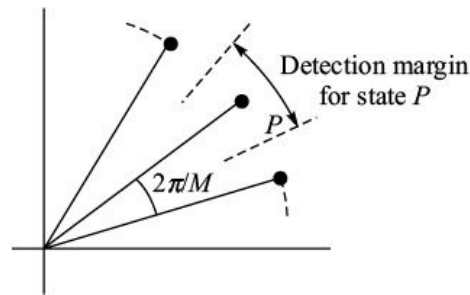


FIGURE 4.13 Phase states of M -ary PSK.

Quadrature amplitude modulation is one approach in which separation of the phase states is increased by utilizing combination of amplitude and phase modulations. Figure 4.14 shows the states of 16 QAM. There are sixteen states and each state corresponds to a group of four bits. Unlike PSK, the states are not equidistant from the origin, indicating the presence of amplitude modulation. Note that each state can be represented as the sum of two carriers in quadrature. These carriers can have four possible amplitudes v_1 and v_2 , determined by the value of odd-bit pair and even-bit pair.

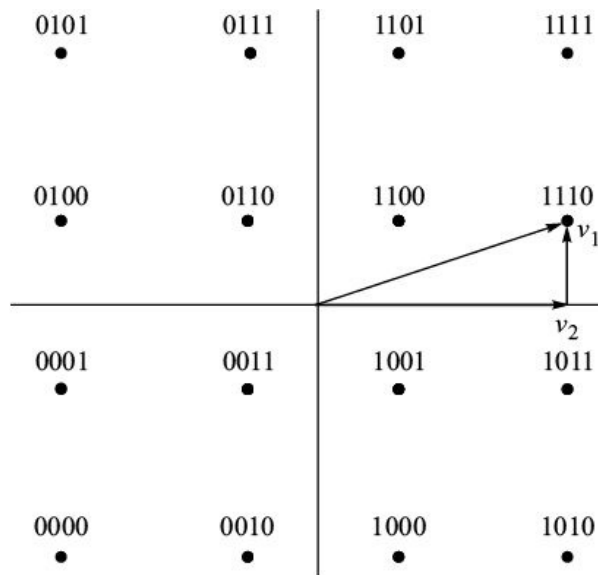


Figure 4.14 Phase states of 16 QAM.

Figure 4.15 shows block schematic of the modulator for 16 QAM. The odd numbered bits at the input are combined in pairs to generate one of the four levels at the D/A (digital to analog converter) output which modulates the carrier. The even numbered bits are combined in a similar manner to modulate the other $p/2$ phase shifted carrier. The modulated carriers are combined to get the 16 QAM output.

It can be shown that 16 QAM gives better performance than 16 PSK. Out of the basic modulation methods PSK comes closest to the Shannon's limit for bit rate which we studied in Chapter 1. QAM displays further improvement over PSK.

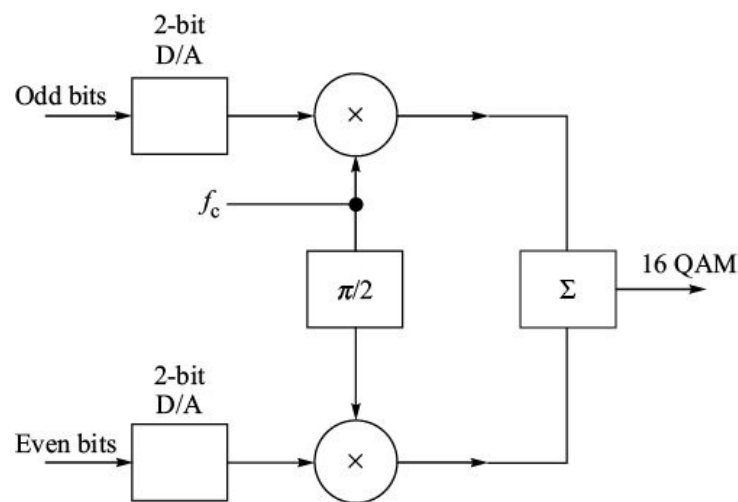


Figure 4.15 16 QAM modulator.

4.4 MODEM

The term 'modem' is derived from the words, MODulator and DEModulator. A *modem* contains a modulator as well as a demodulator. The digital modulation/demodulation schemes discussed above are implemented in the modems. Most of the modems are designed for utilizing the analog voice band service offered by the telecommunication network. Therefore, the modulated carrier generated by a modem fits into the 300–3400 Hz bandwidth of the speech channel.

A typical data connection set up using modems is shown in Figure 4.16. The terminal devices which exchange digital signals are called *data terminal equipments* (DTE). Two modems are always required, one at each end. The modem at the transmitting end converts the digital signal from the DTE into an

analog signal by modulating a carrier. The modem at the receiving end demodulates the carrier and hands over the demodulated digital signal to the DTE.

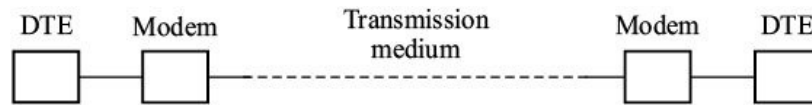


FIGURE 4.16 A data circuit implementation using modems.

The transmission medium between the two modems can be a dedicated circuit or a switched telephone circuit. In the latter case, modems are connected to the local telephone exchanges. Whenever data transmission is required, connection between the modems is established through the telephone exchanges. Modems are also required within a building to connect terminals which are located at distances usually more than 15 meters from the host.

Broadly, a modem is composed of a transmitter, a receiver, and two interfaces (Figure 4.17). The digital interface connects the modem to the DTE which generates and receives the digital signals. The line interface connects the modem to the transmission media for transmitting and receiving the modulated signals. Digital signal to be transmitted is applied to the transmitter through the digital interface. The modulated carrier that is received from the distant end at the line interface is applied to the receiver.

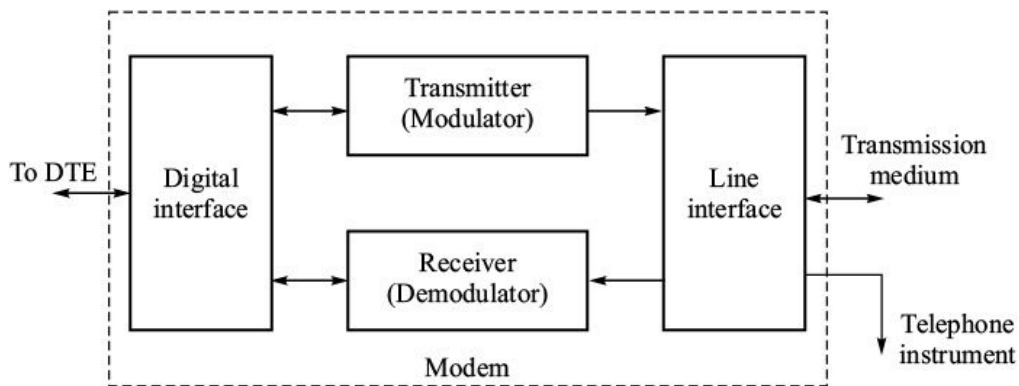


Figure 4.17 Building blocks of a modem.

Modems connected to telephone exchanges have additional provision for connecting a telephone instrument. The transmitter and receiver in a modem consist of several signal processing circuits which include a modulator in the transmitter and a demodulator in the receiver.

4.4.1 Types of Modems

Modems can be of several types and they can be categorized in a number of ways. Categorization is usually based on the following basic modem features:

- Directional capability—half duplex modem and full duplex modem.
- Connection to the line—2-wire modem and 4-wire modem.
- Transmission mode—asynchronous modem and synchronous modem.

Half duplex and full duplex modems. A *half duplex modem* permits transmission in one direction at a time. If a carrier is detected on the line by the modem, it gives an indication of the incoming carrier to the DTE through a control signal of its digital interface (Figure 4.18a). So long as the carrier is being received, the modem does not give clearance to the DTE to transmit.

A *full duplex modem* allows simultaneous transmission in both directions. Thus, there are two carriers on the line, one outgoing and the other incoming (Figure 4.18b).

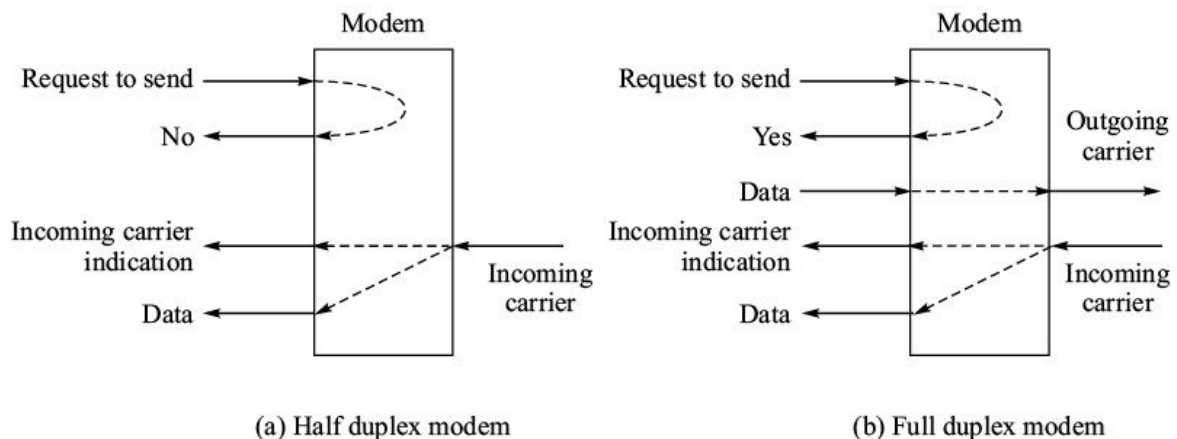


Figure 4.18 Full duplex and half duplex modems.

2-wire and 4-wire modems. The line interface of the modem can have a 2-wire or a 4-wire connection to transmission medium. In a 4-wire connection, one pair of wires is used for the outgoing carrier and the other pair is used for the incoming carrier (Figure 4.19). Full duplex and half duplex modes of data transmission are possible on a 4-wire connection. As the physical transmission path for each direction is separate, the same carrier frequency can be used for both the directions.

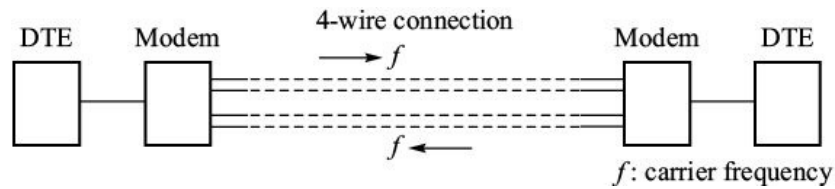


FIGURE 4.19 4-wire modems.

A leased 2-wire connection is cheaper than a 4-wire connection because only one pair of wires is extended to the subscriber's premises. The data connection established through telephone exchange is also a 2-wire connection. Modems with a 2-wire line interface are required for 2-wire connections. Such modems use the same pair of wires for outgoing and incoming carriers. Half duplex mode of transmission using the same frequency for the incoming and outgoing carriers can be easily implemented (Figure 4.20a). The transmit and receive carrier frequencies can be the same because only one of them is present on the line at a time.

For full duplex mode of operation on a 2-wire connection, it is necessary to have two transmission channels, one for the transmit direction and the other for receive direction (Figure 4.20b). This is achieved by frequency division multiplexing of two different carrier frequencies. These carriers are placed within the bandwidth of the speech channel (Figure 4.20c). A modem transmits data on one carrier and receives data from the other end on the other carrier. A hybrid is provided in the 2-wire modem to couple the line to its modulator and demodulator.

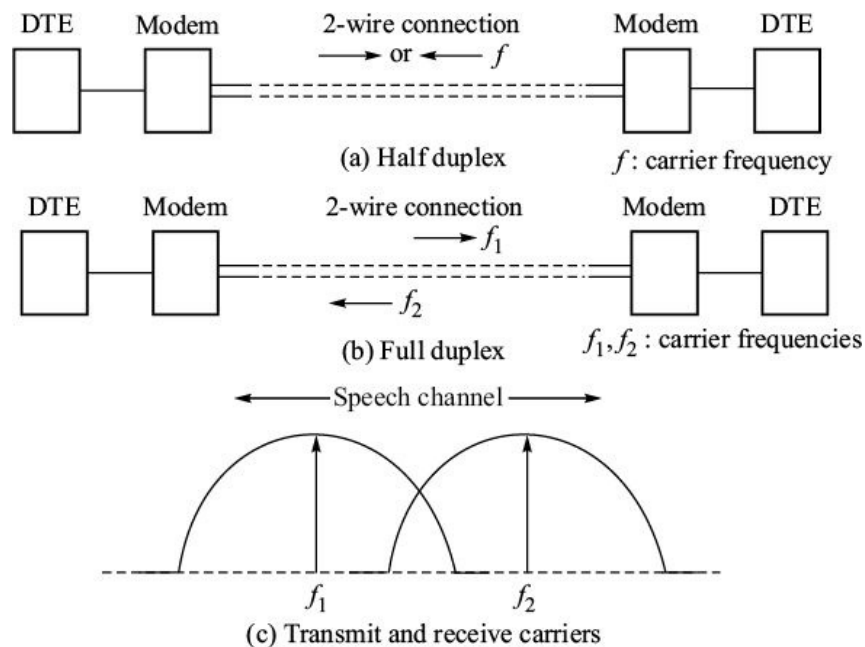


Figure 4.20 2-wire modems

Figure 4.20 2-wire modems.

Note that available bandwidth for each carrier is reduced to half. Therefore, the baud rate is also reduced to half. There is a special technique which allows simultaneous transmission of incoming and outgoing carriers having the same frequency on the 2-wire transmission medium. Full bandwidth of the speech channel is made available to both the carriers simultaneously. This technique is called *echo cancellation technique* and is implemented in high speed 2-wire full duplex modems.

Asynchronous and synchronous modems. Modems for asynchronous and synchronous transmission are of different types. An *asynchronous modem* can only handle data bytes with start and stop bits. There is no separate timing signal or clock between the modem and the DTE (Figure 4.21a). The internal timing pulses are synchronized repeatedly to the leading edge of the start pulse.

A *synchronous modem* can handle a continuous stream of data bits but requires a clock signal (Figure 4.21b). The data bits are always synchronized to the clock signal. There are separate clocks for the data bits being transmitted and received.

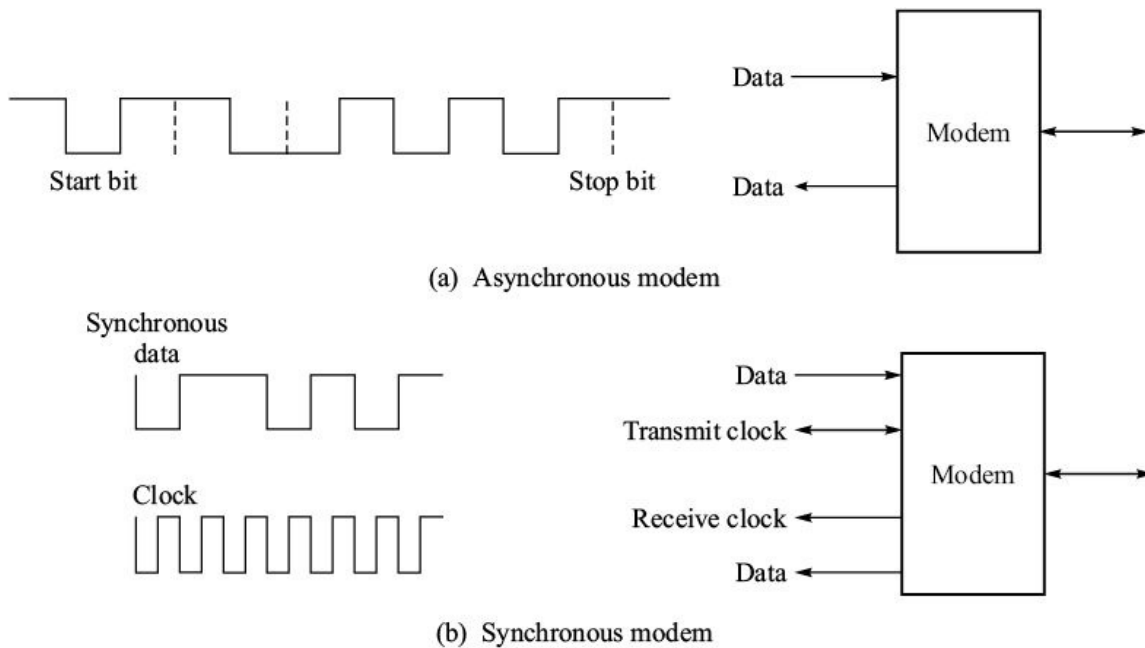


Figure 4.21 Asynchronous and synchronous modems.

For synchronous transmission of data bits, the DTE can use its internal clock and supply the same to the modem. Else, it can take the clock from the modem which recovers the clock signal from the received data signal and supplies it to the DTE. It is, however, necessary that the received data signal contains enough

transitions to ensure that the timing extraction circuit remains in synchronization. High speed modems are equipped with scramblers and descramblers for this purpose.

4.4.2 Scrambler and Descrambler As mentioned above, it is essential to have sufficient transitions in the transmitted data for clock extraction. A scrambler is provided in the transmitter to ensure this. It uses an algorithm to change the data stream received from the terminal in a controlled way so that a continuous stream of zeros or ones is avoided. The scrambled data is descrambled at the receiving end using a complementary algorithm.

There is another reason for using scramblers. It is often seen in data communications that computers transmit 'idle' characters for relatively long periods of time and then there is a sudden burst of data. The effect is seen as repeating errors at the beginning of the data. The reason for these errors is sensitivity of the receiver clock phase to certain data patterns. If the transmission line has poor group delay characteristic in some part of the spectrum and the repeated data pattern concentrates the spectral energy in that part of the spectrum, the recovered clock phase can be offset from its mean position. Drifted clock phase results in errors when the data bits are regenerated. This problem can be overcome by properly equalizing the transmission line. But it may not be possible always. Therefore the data always randomized using scrambler before it is transmitted to avoid errors due to pattern sensitivity of the clock phase.

The scrambler at the transmitter consists of a shift register with some feedback loops and exclusive OR gates. Figure 4.22 shows a scrambler used in the ITU-T V.27 4800 bps modem.

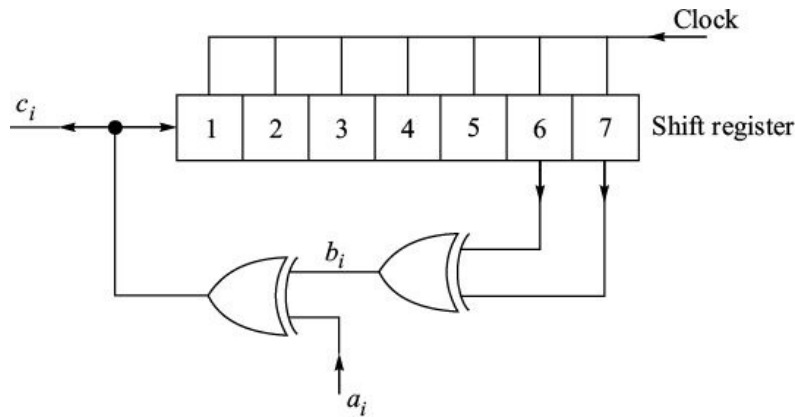


Figure 4.22 Scrambler used in ITU-T V.27 modem.

The output c_i (Figure 4.22) can be written as $c_i = a_i + b_i = a_i + c_{i-6} + c_{i-7}$

If we represent one-bit delay using a delay operator x^{-1} , the above equation can be rewritten as follows: $c_i = a_i + c_i(x^{-6} + x^{-7})$ or

$c_i = a_i / (1 + x^{-6} + x^{-7})$ Note that addition and subtraction operations are the same in modulo-2 arithmetic. Thus, a scrambler effectively divides the input data stream by polynomial $1 + x^{-6} + x^{-7}$. This polynomial is called the *generating polynomial*. By proper choice of the polynomial, it can be assured that undesirable bit sequences are avoided at the output. The generating polynomials recommended by ITU-T for scramblers are given in Table 4.5.

TABLE 4.5 ITU-T Generating Polynomials	
ITU-T recommendations	Generating polynomial
V.22, V.22bis	$1 + x^{-14} + x^{-17}$
V.27	$1 + x^{-6} + x^{-7}$
V.26ter, V.29, V.32	$1 + x^{-18} + x^{-23}, 1 + x^{-5} + x^{-23}$

To get back the data sequence at the receiving end, the scrambled data stream is multiplied by the same generating polynomial. The descrambler is shown in Figure 4.23.

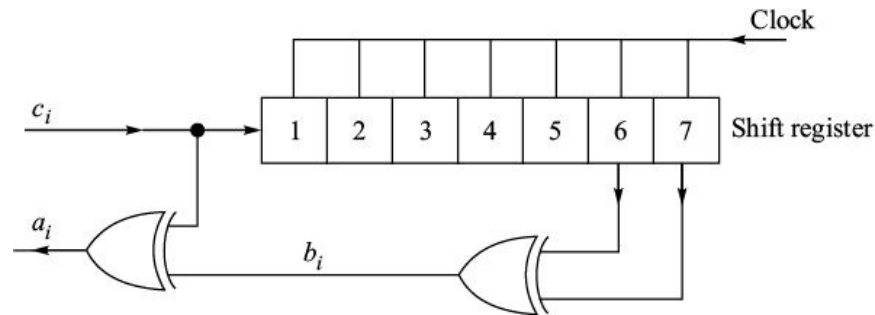


FIGURE 4.23 Descrambler used in ITU-T V.27 modem.

In this case the scrambled bit stream c_i is the input and the descrambled output is a_i , which is obtained by multiplying c_i with the generating polynomial.

$a_i = c_i + b_i = c_i + c_{i-6} + c_{i-7} = c_i (1 + x^{-6} + x^{-7})$ In the above analysis, we have assumed that there was no transmission error. If an error occurs in the scrambled data, it is reflected in three data bits after descrambling. If one of the scrambled bits c_i is received wrong, a_i , a_{i+6} , and a_{i+7} will be affected as c_i moves along the shift register. Therefore, scramblers result in increased error rate but their usefulness outweighs this limitation.

4.4.3 Block Schematic of a Modem With this background, we can now describe the detailed block schematic of a modem. The modem design and complexity vary depending on the bit rate, type of modulation, and other basic features as discussed above. Low speed modems up to 1200 bps are asynchronous and use FSK. Medium speed modems from 2400 to 4800 bps use differential PSK. High speed modems which operate at 9600 bps and above employ QAM and are the most complex. Medium and high speed modems operate in synchronous mode of transmission.

Figure 4.24 shows important components of a typical synchronous differential PSK modem. It must, however, be borne in mind that this design gives the general functional picture of the modem. Actual implementation will vary from vendor to vendor.

Digital interface. The digital interface connects the internal circuits of the modem to the DTE. On the DTE side, it consists of several wires carrying different signals. These signals are either from the DTE or from the modem. The

digital interface contains drivers and receivers for these signals. Brief descriptions of some of the important signals are given below:

- Transmitted data (TD) signal from the DTE to the modem carries data to be transmitted.
- Received data (RD) signal from the modem carries the data received from the other end.
- DTE Ready (DTR) signal from the DTE and indicates readiness of the DTE to transmit and receive data.
- Data Set Ready (DSR) signal from the modem indicates its readiness to transmit and receive data signals.
- Request to Send (RTS) signal from the DTE seeks permission of the modem to transmit data.
- Clear to Send (CTS) signal from the modem gives clearance to the DTE to transmit data. CTS is given as response to the RTS.
- Received line signal detector signal from the modem indicates that the incoming carrier has been detected on the line interface.
- Timing signals are the clock signals from the DTE to the modem and from the modem to the DTE for synchronous transmission.

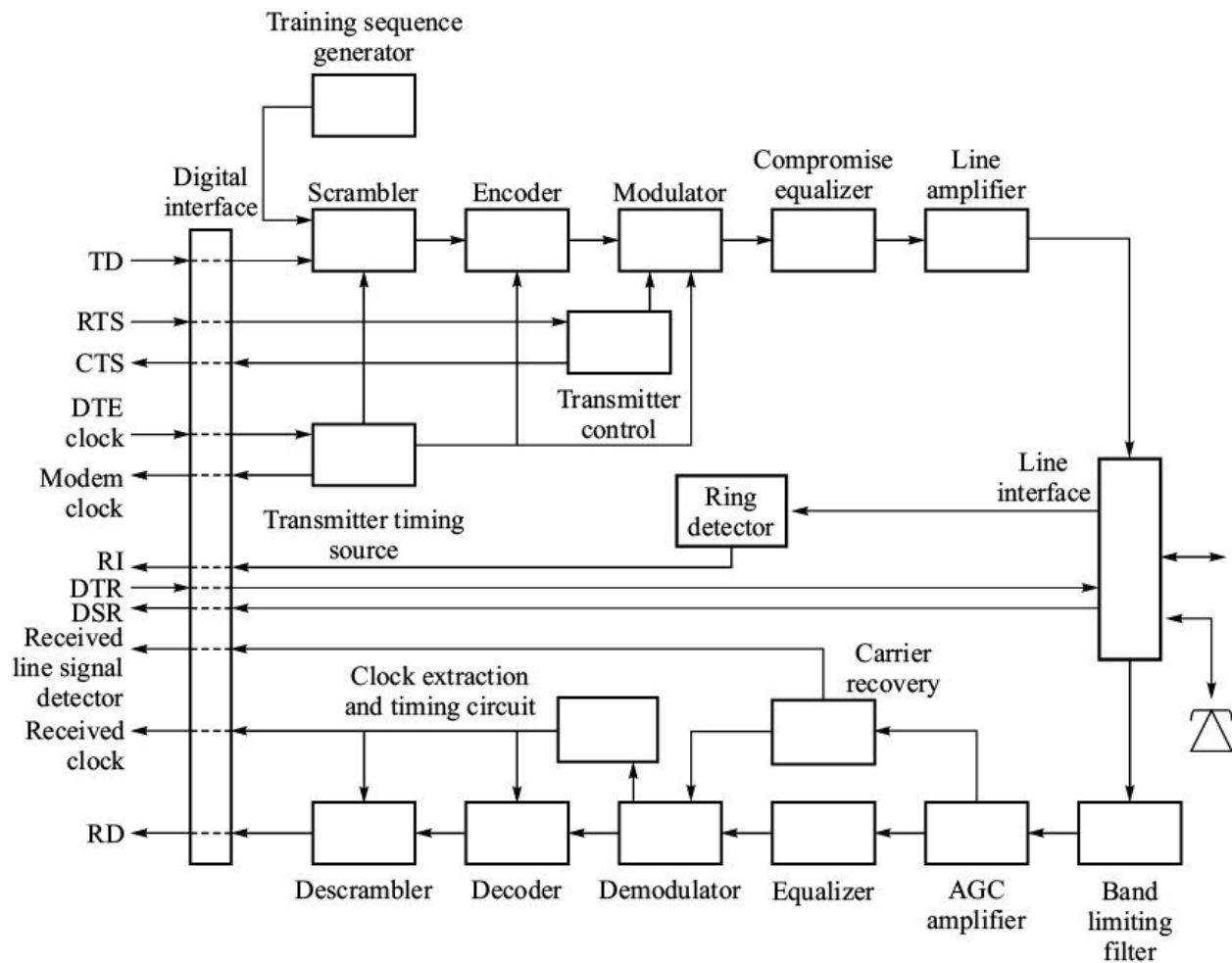


Figure 4.24 Block schematic of a differential PSK modem.

Digital interface has been standardized so that there are no compatibility problems. There are several standards, but the most common standard digital interface is EIA 232 D. There are equivalent ITU-T recommendations also. We will study the digital interface in detail in Chapter 7.

Scrambler. *Scrambler* is incorporated in the modems which operate at data rate of 4800 bps and above. The data stream received from the DTE at the digital interface is applied to the scrambler. The scrambler divides the data stream by the generating polynomial and its output is applied to the encoder.

Encoder. The *encoder* consists of a serial to parallel converter for grouping the serial data bits received from the scrambler, e.g. in a modem employing 4 PSK, dibits are formed. The data bit groups are then encoded for differential PSK.

Modulator. The *modulator* changes the carrier phase as per the output of the encoder. A pulse shaping filter precedes the modulator to reduce the intersymbol

interference. Raised cosine pulse shape is usually used. The modulator output is passed through a band pass filter to restrict the bandwidth of the modulated carrier within the specified frequency band.

Compromise equalizer. *Compromise equalizer* is a fixed equalizer which provides pre-equalization of the anticipated gain and delay characteristics of the line.

Line amplifier. The *line amplifier* is provided to bring the carrier level to the desired transmission level. Output of the line amplifier is coupled to the line through the line interface.

Transmitter timing source. Synchronous modems have an in-built crystal clock source which generates all the timing references required for the operation of the encoder and the modulator. The clock is also supplied to the DTE through the digital interface. The modem has provision to accept the external clock supplied by the DTE.

Transmitter control. This circuit controls transmission of the carrier from the modem. When the RTS is received from the DTE, it switches on the outgoing carrier and sends it on the line. After a brief delay, it sends the CTS signal to the DTE so that it may start transmitting data. In half duplex modems CTS is not given if the modem is receiving a carrier.

Training sequence generator. For reception of the data signals through the modems, it is necessary that the following operational conditions are established in the receiver portion of the modems beforehand:

- The received carrier is detected and recovered. Gain of the AGC amplifier is adjusted and absolute phase reference of the recovered carrier is established.
- The adaptive equalizer is conditioned for the line characteristics.
- The receiver timing clock is synchronized.
- The descrambler is synchronized to the scrambler.

These functions are carried out by sending a training sequence. It is transmitted by a modem when it receives the RTS signal from the DTE. On receipt of RTS from the DTE, the modem transmits a carrier modulated with the training sequence of fixed length and then it gives the CTS signal to the DTE so that it may commence transmission of its data. On receipt of the carrier

modulated with the training sequence, the modem at the receiving end

- recovers the carrier,
- establishes its absolute phase reference,
- conditions its adaptive equalizer, and
- synchronizes its clock and descrambler.

The composition of the training sequence depends on the type of the modem. We will examine some of the training sequences while discussing the modem standards later.

Line interface. The *line interface* provides connection to the transmission line through coupling transformers. The coupling transformers isolate the line for DC signals. The transmission line can provide a 2-wire or 4-wire connection between the two modems. For a 4-wire connection, there are separate transformers for the transmit and receive directions. For a 2-wire connection, the line interface is equipped with a hybrid.

Receive band limiting filter. In the receive direction, the *band limiting filter* selects the received modulated carrier and removes the out-of-band noise.

AGC amplifier. Automatic Gain Control (AGC) amplifier provides variable gain to compensate for carrier-level loss during transmission. The gain depends on the received carrier level.

Equalizer. The *equalizer* section of the receiver corrects the attenuation and group delay distortion introduced by the transmission medium and the band limiting filters. Fixed, manually adjustable or adaptive equalizers are provided depending on speed, line condition, and the application. In high speed dial up modems, an adaptive equalizer is provided because characteristics of the transmission medium change on each instance of call establishment.

Carrier recovery circuit. The carrier is recovered from the AGC amplifier output by this circuit. The recovered carrier is supplied to the demodulator. An indication of the incoming carrier is given at the digital interface.

Demodulator. The *demodulator* recovers the digital signal from the received modulated carrier. The carrier required for demodulation is supplied by the carrier recovery circuit.

Clock extraction circuit. The *clock extraction circuit* recovers the clock from

the received digital signal. The clock is used for regenerating the digital signal and to provide the timing information to the decoder. The receiver clock is also made available to the DTE through the digital interface.

Decoder. The *decoder* performs a function complementary to the encoder. The demodulated data bits are converted into groups of data bits which are serialized by using a parallel to serial converter.

Descrambler. The decoder output is applied to the *descrambler* which multiplies the decoder output by the generating polynomial. The unscrambled data is given to the DTE through the digital interface.

4.4.4 Additional Modem Features As mentioned above, modems vary in design and complexity depending on speed, mode of transmission, modulation methods, and their application. The driving force for the developments in modems has been the high cost of the transmission medium. By more efficient utilization of the available bandwidth and increasing the effective throughput, the high cost of transmission can be neutralized. Echo cancellers and secondary channel are the two additional features of modems in this direction. Other additional features of modems include test loops, compression and error control.

Echo canceller. Full duplex transmission of data on 2-wire leased or dial up connection is implemented by dividing the available frequency band for the two carriers. This effectively reduces the available bandwidth for each carrier to half and limits the data speed to about 2400 to 4800 bps. Echo cancellation makes it possible to use the same carrier frequency and the entire frequency band for both the carriers simultaneously.

When the transmit and receive carrier frequencies are same, the transmitted carrier must be prevented from appearing at the local receiver input. The line-coupling hybrid gives about 15 dB loss across the opposite ports. Thus the transmitted carrier with 15 dB loss appears at the receiver input of the modem. This signal is referred to as *near-end echo* (Figure 4.25). It has high amplitude and very short delay.

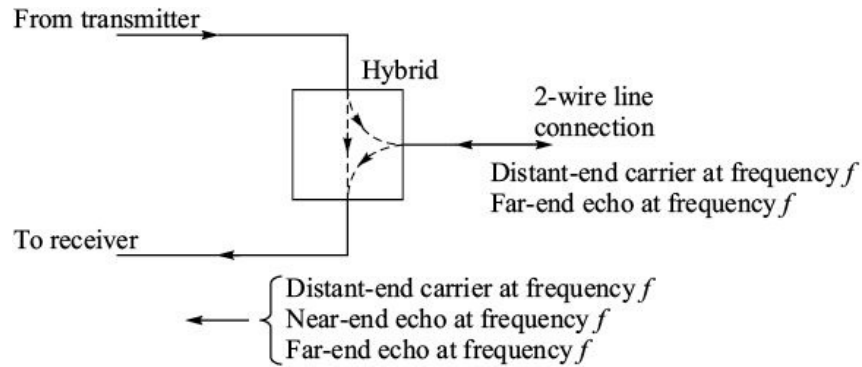


Figure 4.25 Far-end and near-end echoes present in 2-wire full duplex modem.

There is another type of echo which is called the *far-end echo*. Far-end echo is caused by the hybrids present in the interconnecting telecommunication link. It is characterized by low amplitude but long delay. For terrestrial connections, the delay can be of the order 40 ms and for the satellite based connections, it is of the order of half a second.

The echo being at the same carrier frequency as the received carrier, interferes with the demodulation process and needs to be removed. For this purpose, an *echo canceller* is built into the high-speed modems. It generates a copy of the echo from the transmitted carrier and subtracts it from the received signals. The echo canceller circuit consists of a tapped-delay line with a set of coefficients which are adjusted to get the minimum echo at the receiver input. This adjustment is carried out when the training sequence is being transmitted.

Secondary channel. We have seen that a DTE needs to exchange RTS/CTS signals with the modem before it transmits data. On receipt of the RTS signal, the modem gives the CTS after a certain delay. During this period, it transmits the training sequence so that the modem at the other end may detect the carrier, extract the clock, synchronize the descrambler, and condition the equalizers. If the mode of operation is half duplex, each reversal of the direction of transmission involves RTS-CTS delay and thus, reduced the effective throughput. In most of the data communication situations, the receiver sends short acknowledgements for every received data frame and for transmitting these acknowledgements the direction of transmission must be reversed. To avoid frequent reversal of direction of transmission, a low speed secondary channel is provided in the modems (Figure 4.26).

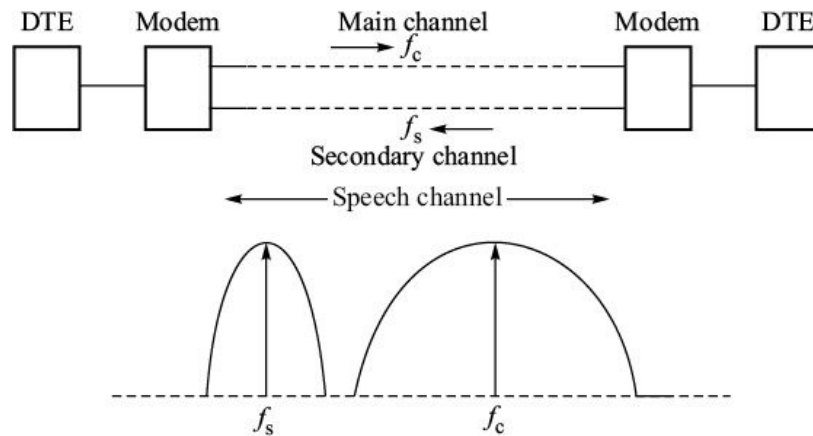


Figure 4.26 Secondary channel.

The *secondary channel* operates at 75 bps and uses FSK. The secondary channel has its own RTS, CTS and other control signals which are available at the digital interface of the modem. It should be noted that the main channel is used in half duplex mode for data transmission and the DTEs are configured to send the acknowledgements on the secondary channel.

Test loops. Modems are provided with the capability for locating faults in the digital connection from DTE to DTE. The testing procedure involves sending a test data and looping it back at various stages of the connection. The test pattern can be generated by the modem internally or it can be applied externally using modem tester. The common test configurations are shown in Figure 4.27.

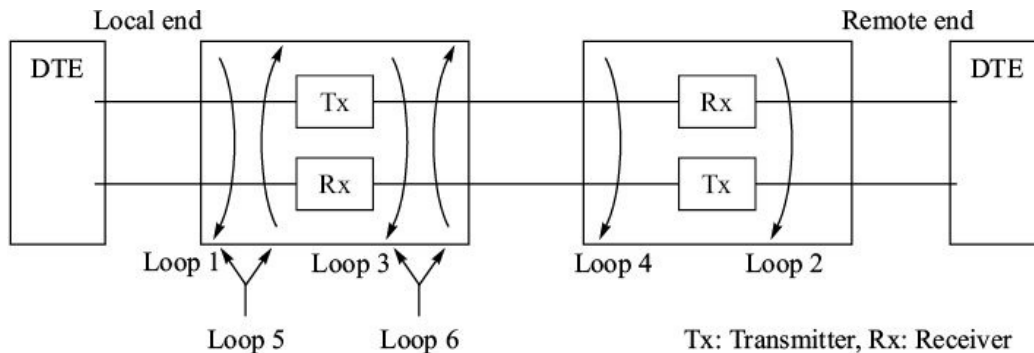


FIGURE 4.27 Test loops in modems.

Loop 1: Digital loopback. This loop is set up as close as possible to the digital interface.

Loop 2: Remote digital loopback. This loop checks the line and the remote modem. It can be used only in full duplex modems.

Loop 3: Local analog loopback. The modulated carrier at the transmitter output of the local modem is looped back to the receiver input. The loopback

may require some attenuators on the line side to adjust the level.

Loop 4: Remote analog loopback. This loop arrangement is applicable for 4-wire line connections only. The two pairs at the distant end are disconnected from the modem and connected to each other.

Loop 5: Local digital loopback and loopforward. In this case, the local digital loopback is provided for the local modem and remote digital loopback is provided for the remote modem.

Loop 6: Local analog loopback and loopforward. In this case, the local modem has analog loopback and the remote modem has remote analog loopback.

The test configurations can be set up by pressing the appropriate switches provided on the modems or using the modem command set. The digital interface also provides some control signals for activating the loop tests. When in the test mode, the modem indicates its test status to the local DTE through a control signal in the digital interface.

All modems do not have provision for all these tests. Test features are specific to the modem type. Test loops 1 to 4 have been standardized by ITU-T in their Recommendation V.54.

Error control and data compression. To meet the high data rate requirements of modems, error control and compression mechanisms are built into the modems. For error control a protocol called *link access procedure* for modems (LAP-M) is used. This protocol is based on retransmission of data frames that are received with errors. The protocol is applicable between two communicating modems. We will discuss LAP-M in Chapter 9, Data Link Protocols. This protocol is specified in ITU-T V.42 recommendation.

To achieve high data rate compression mechanism is also incorporated in some modems. Lempel-Ziv-Welch compression is one of the examples. It gives compression up to 1:4 if a file has not already been compressed by some other mechanism. The ITU-T recommendation for compression in modems is V.42bis.

4.5 STANDARD MODEMS

It is essential that modems conform to international standards because similar modems supplied by different vendors must work with each other. ITU-T has drawn up modem standards which are internationally accepted. We will discuss

some of the important ITU-T modems to illustrate the application of concepts developed earlier. The reader is urged to refer to the ITU-T recommendations for detailed description of all these modems.

4.5.1 ITU-T V.21 Modem

This modem provides full duplex asynchronous transmission over the 2-wire leased line or switched telephone network. It operates at 300 bps. It utilizes FSK over the following two channels:

- Transmit channel frequencies (originating modem)
1180\Hz (space) 980 Hz (mark)
- Receive channel frequencies (originating modem)
1850\Hz (space) 1650 Hz (mark).

4.5.2 ITU-T V.22 Modem

This modem provides full duplex synchronous transmission¹ over 2-wire leased line or switched telephone network. It transmits data at 1200 bps. As an option, it can also operate at 600 bps.

Scrambler. A scrambler and a descrambler having the generating polynomial $1 + x^{-14} + x^{-17}$ are provided in the modem.

Modulation. Differential 4 PSK over two channels is utilized in this modem. The dibits are encoded as phase changes as given in Table 4.6. The carrier frequencies are:

- Low channel 1200 Hz
- High channel 2400 Hz.

A	B	f
0	0	$p/2$
0	1	0
1	1	$3p/2$
1	0	p

At 600 bps, the carrier phase changes are $3p/2$ and $p/2$ for binary 1 and 0 respectively.

Equalizer. Fixed compromise equalizers shared equally between the transmitter and receiver are provided in the modem.

Test loops. Test loops 2 and 3 as defined in Recommendation V.54 are provided in the modem. For self-test, an internally generated binary pattern of alternating 0 and 1 is applied to the scrambler. At the output of the descrambler, an error detector identifies the errors and gives visual indication.

4.5.3 ITU-T V.22bis Modem This modem provides full duplex synchronous transmission on a 2-wire leased line or switched telephone network. The bit rates supported are 2400 or 1200 bps at the modulation rate of 600 bauds.

Scrambler. The modem incorporates a scrambler and a descrambler having the generating polynomial $1 + x^{-14} + x^{-17}$.

Modulation. At 2400 bps, the modem uses 16 QAM having a constellation as shown in Figure 4.28. From the scrambled data stream *quadbits* are formed. The first two bits of the quadbits are coded as quadrant change as given in Table 4.7. The last two bits of the quadbits determine the phase within a quadrant as shown in Figure 4.28. The following two carriers are used for transmit and receive directions. The calling modem uses the low channel to transmit data.

- Low channel carrier 1200 Hz
- High channel carrier 1800 Hz.

At 1200 bps, the dibits are formed from the scrambled data stream and coded as quadrant change. In each quadrant, the phase state corresponding to 01 is transmitted.

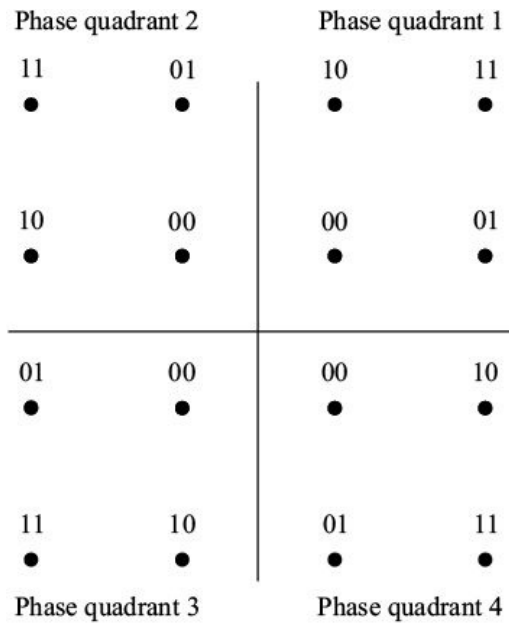


Figure 4.28 Phase states of ITU-T V.22bis 16 QAM modem.

TABLE 4.7 Quadrant Changes (ITU-T V.22bis Modem)

Last quadrant	First two bits of the quadbits			
	00	01	11	10
	Next quadrant			
1	2			
2	3			
3	4	1		
4	1			

	4	3
2	1	4
3	2	1
4	3	2

Equalizer. A fixed compromise equalizer is provided in the modem transmitter. The modem receiver is equipped with an adaptive equalizer.

Test loops. Test loops 2 and 3 as defined in Recommendation V.54 are provided in the modem. For self-test, an internally generated binary pattern of alternating 0 and 1 is applied to the scrambler. At the output of the descrambler, an error detector identifies the errors and gives visual indication.

4.5.4 ITU-T V.23 Modem The modem is designed to operate in full duplex asynchronous transmission mode over a 4-wire leased line. It can also operate in half duplex over a 2-wire leased line and switched telephone network. The modem can operate at two speeds—600 bps and 1200 bps. It is equipped with the secondary channel which operates at 75 bps.

Modulation. The modem employs FSK over two channels. The frequencies are:

- Transmit frequencies (originating modem)
1180 Hz (space) 980 Hz (mark)
- Receive frequencies (originating modem)
1850 Hz (space) 1650 Hz (mark)
- Secondary channel frequencies
450 Hz (space) 390 Hz (mark)

4.5.5 ITU-T V.26 Modem This modem operates in full duplex synchronous mode on a 4-wire leased connection. It operates at 2400 bps. It also includes a secondary channel having bit rate of 75 bps.

Modulation. Differential 4 PSK is employed to transmit data at 2400 bps. The carrier frequency is 1800 Hz. The modulation scheme has two alternatives A and B (Table 4.8). The secondary channel works on FSK with the same frequencies as for V.23.



TABLE 4.8 Modulation Scheme of ITU-T V.26 Modem

Dibit	A	B
	f	f
00	0	$p/4$
01	$p/2$	$3p/4$
11	p	$5p/4$
10	$3p/2$	$7p/4$

4.5.6 ITU-T V.26bis Modem It is a half duplex synchronous modem for use in the switched telephone network. It operates at a nominal speed of 2400 bps or at a reduced speed of 1200 bps. It includes a secondary channel which operates at the speed of 75 bps.

Modulation. The modem uses the differential 4 PSK for transmission at 2400 bps. The modulation scheme is the same as for V.26, alternative B. At 1200 bps, the modem uses differential BPSK with phase changes $p/2$ and $3p/2$ for binary 0 and 1 respectively. The frequencies of the secondary channel are the same as in V.23.

Equalizer. A fixed compromise equalizer is provided in the receiver.

4.5.7 ITU-T V.26ter Modem It is a full duplex synchronous modem for use in 2-wire leased line or switched telephone network. It uses an echo cancellation technique for channel separation. As an option, the modem can accept asynchronous data from the DTE. If asynchronous option is used, the modem converts the asynchronous data suitably for synchronous transmission. The modem operates at a nominal speed of 2400 bps with fallback at 1200 bps.

Modulation. The modem uses differential 4 PSK for transmission at 2400 bps. The carrier frequency is 1800 Hz in both directions. The modulation scheme is the same as for V.26, alternative A. At 1200 bps, differential BPSK is used. The phase changes corresponding to binary 0 and 1 are respectively 0 and p radians respectively.

Equalizer. A fixed compromise equalizer or an adaptive equalizer is provided in

the receiver. No training sequence is provided for convergence of the adaptive equalizer.

Scrambler. The modem incorporates a scrambler and a descrambler. The generating polynomial for the call-originating modem is $1 + x^{-18} + x^{-23}$. The generating polynomial of the answering modem for transmission of its data is $1 + x^{-5} + x^{-23}$.

Test loops. Test loops 2 and 3 as defined in Recommendation V.54 are provided in the modem.

4.5.8 ITU-T V.27 Modem This modem is designed for full duplex/half duplex synchronous transmission over a 4-wire or 2-wire leased connection which is specially conditioned as per M.1020. It operates at the bit rate of 4800 bps with modulation rate of 1600 baud. It includes a secondary channel which operates at 75 bps.

Scrambler. The modem incorporates a scrambler and a descrambler having the generating polynomial $1 + x^{-6} + x^{-7}$.

Modulation. The modem uses differential 8 PSK for transmission at 4800 bps. The modulation scheme is given in Table 4.9. The carrier frequency is 1800 Hz. The secondary channel frequencies are the same as in V.23.

Tribits	Phase change
001	0
000	$p/4$
010	$p/2$
011	$3p/4$
111	p
110	$5p/4$
100	$3p/2$
101	$7p/4$

Equalizer. A manually adjustable equalizer is provided in the receiver. The transmitter sends scrambled continuous binary 1s for the equalizer adjustment.

The modem provides indication of correct adjustment of the equalizer.

4.5.9 ITU-T V.27bis Modem This modem is designed for full duplex/half duplex synchronous transmission over 4-wire/ 2-wire leased connection not necessarily conditioned as per M.1020. Its speed, modulation scheme and other features are the same as in V.27. The principal differences are given below:

- It can operate at a reduced rate of 2400 bps. At 2400 bps, the modem uses differential 4 PSK. The modulation scheme is the same as in V.26, alternative A.
- An automatic adaptive equalizer is provided in the receiver.
- A training sequence generator is incorporated in the transmitter.

The training sequence used in V.27bis modem is shown in Table 4.10. It consists of three segments. Each segment is of defined duration. Duration is expressed in terms of Symbol Intervals (SI). One SI is equal to 1/ baud rate. The figures shown within brackets are for the 2-wire connection and for the 4-wire connection that does not meet M.1020 conditioning requirements.

The first segment consists of continuous phase reversals of the carrier. It enables AGC convergence and carrier recovery. During the second segment, the adaptive equalizer is conditioned. Differential BPSK carrier is transmitted during this interval. The modulating sequence is generated from every third bit of a PRBS (Pseudo-Random Binary Sequence) having the generating polynomial $1 + x^{-6} + x^{-7}$. The phase changes in the carrier are 0 and p radians for binary 0 and 1 respectively. The third segment of the training sequence synchronizes the descrambler. It consists of scrambled binary 1s.

TABLE 4.10 Training Sequence of V.27bis Modem			
	Segment 1	Segment 2	Segment 3
Duration (SI)	14(58)	58(1074)	8
Type of line signal	Continuous 180° phase reversals	Differential BPSK carrier	Differential 8/4 PSK carrier

4.5.10 ITU-T V.27ter Modem This modem is designed for use in the switched telephone network. It is similar to V.27bis modem

in most respects. It incorporates additional circuits for auto-answering, ring indicator, etc.

4.5.11 ITU-T V.29 Modem This modem is designed for point-to-point full duplex/half duplex synchronous operation on 4-wire leased circuits conditioned as per M.1020 or M.1025. It operates at a nominal speed of 9600 bps. The fallback speeds are 7200 and 4800 bps.

Scrambler. The modem incorporates a scrambler and a descrambler having the generating polynomial $1 + x^{-18} + x^{-23}$.

Modulation. The modem employs 16 state QAM with modulation rate of 2400 baud. The carrier frequency is 1700 Hz. The scrambled data at 9600 bps is divided into quadbits. The last three bits are coded to generate differential eight-phase modulation identical to Recommendation V.27. The first bit along with the absolute phase of the carrier determines its amplitude (Figure 4.29a). The absolute phase is established during transmission of the training sequence.

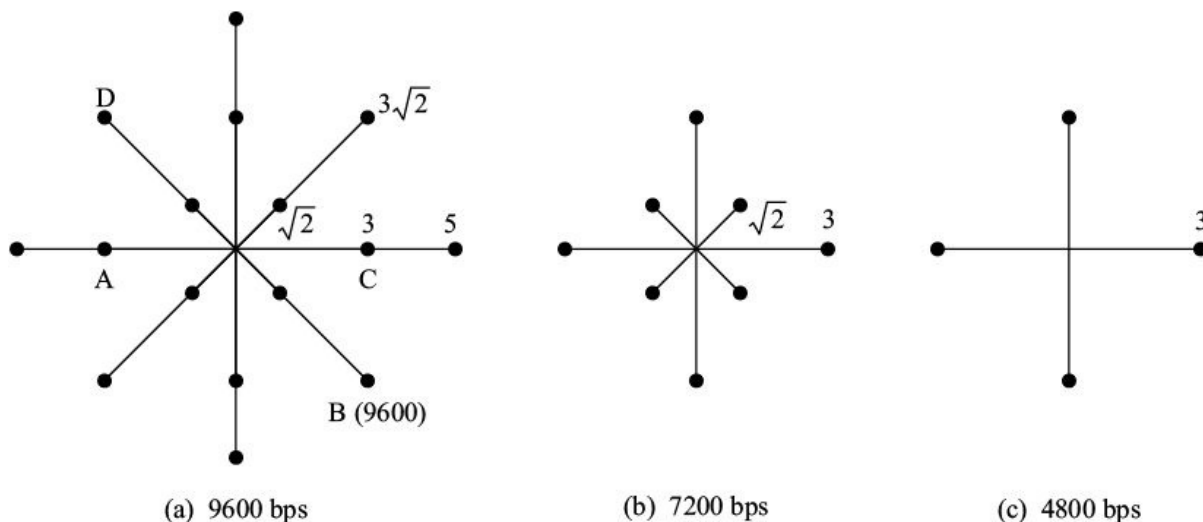


FIGURE 4.29 Phase states of ITU-T V.29 16 QAM modem.

At the fallback rate of 7200 bps, tribits are formed from the scrambled 7200 bps bit stream. Each tribit is prefixed with a zero to make the quadbit. At the fallback rate of 4800 bps, dibits are formed from the scrambled 4800 bps bit stream. These dibits constitute the second and third bits of the quadbits. The first bit of the quadbits is zero as before and the fourth bit is modulo 2 sum of the second and third bits. The phase state diagrams for the modem operation at 7200

and 4800 bps are shown in Figure 4.29.

Equalizer. An adaptive equalizer is provided in the receiver.

Training sequence. The training sequence is shown in Table 4.11. It consists of four segments which provide for clock synchronization, establishment of absolute phase reference for the carrier, equalizer conditioning, and descrambler synchronization.

Segment	Signal type	Duration (Symbol intervals)
1		48
2	No transmitted energy Alternations	128
3	Equalizer conditioning pattern Scrambled binary 1s	384
4		48

The second segment consists of two alternating signal elements A and B (Figure 4.29). This sequence establishes absolute phase of the carrier. The third segment consists of the equalizer conditioning signal which consists of elements C and D (Figure 4.29). Whether C or D is to be transmitted is decided by a pseudo-random binary sequence at 2400 bps generated using the generating polynomial $1 + x^{-6} + x^{-7}$. The element C is transmitted when a 0 occurs in the sequence. The element D is transmitted when a 1 occurs in the sequence. The fourth segment consists of a continuous stream of binary 1s which is scrambled and transmitted. During this period descrambler synchronization is achieved.

4.5.12 ITU-T V.32 Modem This modem is designed for full duplex synchronous transmission on 2-wire leased line or switched telephone network. It can operate at 9600 and 4800 bps. The modulation rate is 2400 bauds.

Scrambler. The modem incorporates a scrambler and a descrambler. The generating polynomial for the call-originating modem is $1 + x^{-18} + x^{-23}$. The generating polynomial of the answering modem for scrambling its data bits is $1 + x^{-5} + x^{-23}$.

Modulation. The carrier frequency is 1800 Hz in both directions of transmission. Echo cancellation technique is employed to separate the two channels. 16 or 32 state QAM is employed for converting the digital signal into the analog signal. There are two alternatives for encoding the 9600 bps

scrambled digital signal, non-redundant coding and trellis coding.

Non-redundant coding. The scrambled digital signal is divided into quadbits. The first two bits of each quadbit Q_{1n} and Q_{2n} are differentially encoded into Y_{1n} and Y_{2n} respectively as per Table 4.12. $Y_{1(n-1)}$, $Y_{2(n-1)}$ are the previous values of the Y bits. The last two bits are taken without any change and the encoded quadbit $Y_{1n}Y_{2n}Q_{3n}Q_{4n}$ is mapped as shown in Figure 4.30.

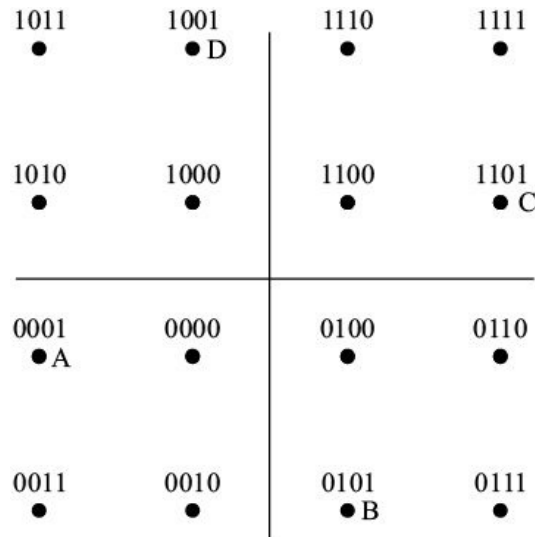


Figure 4.30 Phase states of ITU-T V.32 modem at 9600 (Non-redundant coding).

At 4800 bps, the scrambled data stream is grouped into dibits which are differentially encoded as per Table 4.12 and mapped on a subset ABCD of the phasor states (Figure 4.30).

$Y_{1(n-1)}Y_{2(n-1)}$	$Q_{1n}Q_{2n}$			
	00	01	10	11
	$Y_{1n}Y_{2n}$			
00	01	00	11	10
01	11	01	10	00
10	00	10	01	11
11	10	11	00	01

Trellis coding. *Trellis coding* enables detection and correction of errors which are introduced in the transmission medium. We will study the principles of error control using trellis coding in the next chapter. There are several coding algorithms for error control and trellis coding is one of them. It is implemented using convolution encoders. It is sufficient to mention at this stage that some

additional bits are added to a group of data bits for detecting and correcting the errors. In trellis coded V.32 modem, quadbits formed from the scrambled data stream, are converted into groups of five bits using a convolution encoder. The coding steps are as under:

1. The first two bits Q_{1n} and Q_{2n} of the quadbit are differentially encoded into Y_{1n} and Y_{2n} as given in Table 4.13.
2. From Y_{1n} and Y_{2n} , Y_{0n} is generated using the convolution encoder.
3. Y_{0n} , Y_{1n} , and Y_{2n} form the first three bits of the five bit code. The last two bits of the code are Q_{3n} and Q_{4n} bits of the quadbit.

TABLE 4.13 Differential Encoding for the First Two Bits of the Quadbit (V.32 Trellis Coded Modem)

$Y_{1(n-1)}Y_{2(n-1)}$	$Q_{1n}Q_{2n}$			
	00	01	10	11
	$Y_{1n}Y_{2n}$			
00	00	01	10	11
01	01	00	11	10
10	10	11	01	00
11	11	10	00	01

The phase state diagram of the V.32 trellis coded modem is shown in Figure 4.31.

Equalizer. An adaptive equalizer is provided in the receiver.

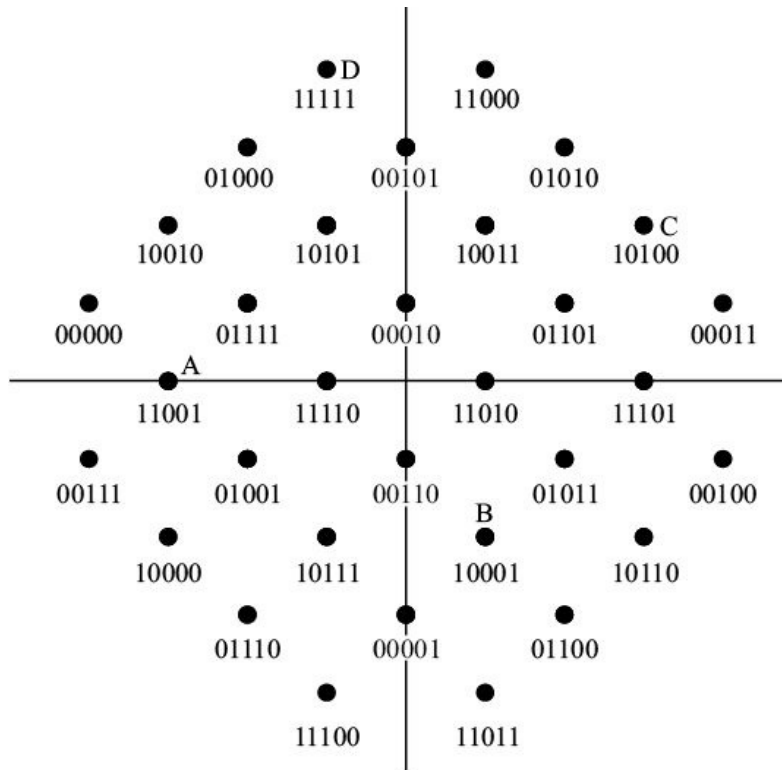


Figure 4.31 Phase states of ITU-T V.32 modem at 9600 bps (Trellis coding).

Training sequence. A training sequence is provided in the modem for adaptive equalization, echo cancellation, data rate selection, and for the other functions described earlier. It consists of the following five segments (Table 4.14). For details of the training sequence, the reader is advised to refer to the ITU-T recommendation.

TABLE 4.14 Training Sequence of ITU-T V.32 Modem		
Segment	Signal type	Duration (Symbol intervals)
1		256
2	Alternations between states A and B (Figure 4.31)	16
3	Alternations between states C and D (Figure 4.31) Equalizer and echo canceller conditioning pattern	1280
4	Data rate indicating sequence	8
5	Scrambled binary 1s	128

Test loops. Test loops 2 and 3 as defined in Recommendation V.54 are provided in the modem.

4.5.13 ITU-T V.33 Modem This modem is designed for full duplex synchronous transmission on 4-wire leased connections

conditioned as per M.1020 or M.1025. It operates at 14,400 bps with modulation rate of 2400 bauds. The fallback speed is 12,000 bps.

Scrambler. The modem incorporates a scrambler and a descrambler. The generating polynomial for the call-originating modem is $1 + x^{-18} + x^{-23}$.

Modulation. The carrier frequency is 1800 Hz in both directions of transmission. 128 state QAM using trellis coding is employed for converting the digital information into an analog signal. The scrambled data bits are divided into groups of six bits. The first two bits of each six-bit group are encoded into three bits using the differential encoder followed by a convolution encoder as described in V.32. Seven bit code words are thus formed and these codes are mapped to 128 phase state diagram similar to Figure 4.31. At the fallback speed of 12,000 bps, five-bit groups are formed instead of six-bit groups. The first two bits of each group are coded into three bits using the same scheme as above. The six-bit codes so generated are mapped to 64 phase states in similar manner as shown in Figure 4.30.

Equalizer. An adaptive equalizer is provided in the receiver.

Training sequence. The training sequence is provided in the modem for adaptive equalization, data rate selection and the other functions described earlier. For details of the training sequence, the reader can refer to the ITU-T recommendation.

4.5.14 ITU-T V.34 Modem This modem is designed for full duplex synchronous/asynchronous transmission over 2-wire dial-up or leased connection. It supports speed from 2400 bps to 33.6 kbps. It implements error correcting protocol V.42 and data compression protocol V.42bis.

4.5.15 ITU-T V.90 Modem This modem is designed for full duplex synchronous/asynchronous transmission over 2-wire dial-up connection. It is asymmetric modem in the sense that its upstream and downstream speeds are different. It supports downstream speed up to 56 kbps and upstream speed up to 33.6 kbps. High downstream speed is achieved by use of digital

connectivity between the telephone exchange and the Internet service provider.

4.6 OTHER MODEMS AND LINE DRIVERS

The ITU-T modems discussed above are designed to operate on the speech channel of 300 to 3400 Hz provided by the telecommunication network. Filters are provided in the network to restrict the bandwidth to this value to multiplex several channels on the transmission media. There are other line devices used as modems that are designed to work beyond this frequency band to achieve higher bit rates.

4.6.1 Limited Distance Modems The copper pair as such provides much wider frequency pass band as we saw in the last chapter. Limited-distance Modems (LDM) are designed for the entire frequency band of the non-loaded copper transmission line. Their application is limited to short distances as the transmission medium distortions and attenuation increase with the distance. The distance limitation is a function of bit rate and cable characteristics. The longer the distance, the slower must the transmission speed be. Some typical figures are 20 kilometers at 1200 bps and 8 kilometers at 19,200 bps on 26-gauge cable. LDMs usually require 4-wire unloaded connection between modems.

4.6.2 Baseband Modems Another class of modems which fall under the category of LDMs are the baseband modems. Baseband modems do not have modulators and demodulators. They utilize digital baseband transmission and incorporate equalizers to compensate for the media characteristics.

4.6.3 Line Drivers Line drivers as modem substitutes provide transmission capabilities usually limited to within buildings

where the terminals are separated from the host at distances which cannot be supported by the digital interface. A line driver converts the digital signal to low-impedance balanced signal which can be transmitted over a twisted pair. For the incoming signals, a line driver also incorporates a balanced line receiver. Line drivers usually require DC continuity of the transmission medium.

4.6.4 Group Band Modems The FDM telecommunication network provides group band service which extends from 60 kHz to 108 kHz. The modems designed to operate over this frequency band are called *group band modems*. ITU-T V.36 group band modem provides synchronous transmission at bit rates 48, 56, 64, and 72 kbps. Single sideband amplitude modulation of carrier at 100 kHz is used. The carrier is also transmitted along with the modulated signal. An optional speech channel occupying the frequency band 104 to 108 kHz is integrated into the modem. The modem incorporates a scrambler and a descrambler. For bit rates higher than 72 kHz, ITU-T has specified the V.37 group band modem. It supports 96 kbps, 112 kbps, 128 kbps, and 144 kbps bit rates.

4.7 DIGITAL SUBSCRIBER LINE (DSL)

Data service through the telephone network using voice band modems serves the purpose of data processing community to limited extent as the data rates achievable on 300–3400 Hz frequency band cannot meet the requirements for video and high speed internet. As we saw in Chapter 3, the limitation of bandwidth is within the core telephone network. The copper cable based access network can support much wider bandwidth. Thus if a broadband core data network is installed, it should be possible to use the existing copper cable based access network for the broadband data services. It can result in large savings as access network cost constitutes 60–70% of the total network cost.

Digital subscriber line systems are deployed between the telephone exchange and customer's premises to provide high speed data access in addition to telephone connection on the same copper pair (Figure 4.32). Broadband data

node can be a router or ATM switch.

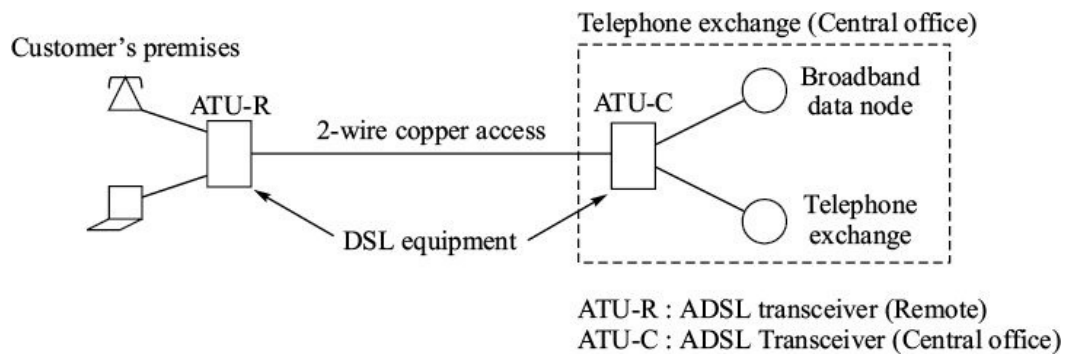


Figure 4.32 Digital subscriber line (DSL) equipment.

There are several DSL technologies that are collectively referred to as xDSL. Asymmetric digital subscriber line (ADSL), is most commonly deployed and described in detail in the next section. Other DSL technologies SDSL, HDSL, and VDSL are described briefly towards the end of the section.

4.7.1 Asymmetric Digital Subscriber Line (ADSL) As the name suggests, *asymmetric digital subscriber line* allows higher down stream data rate from the central office (telephone exchange premises) towards the customer than upstream data rate from the customer end towards the central office. This asymmetry is suitable for internet surfing, video-on-demand and remote LAN access applications. These applications demand higher down stream bit rate than upstream bit rate.

In addition to the connectivity for asymmetric data service, ADSL provides connectivity for standard telephone service on the same copper pair. ADSL uses frequency division multiplexing to separate the voice band, the upstream data, and down stream data (Figure 4.33a). The lowest frequency band is reserved for voice service. The next frequency band is reserved for upstream data from the subscribers. The uppermost band is for downstream data to the subscribers.

The upstream and downstream data bands can be overlapping as shown in Figure 4.33b. In this case, echo-cancellation technique is used for separating the two overlapping frequency bands.

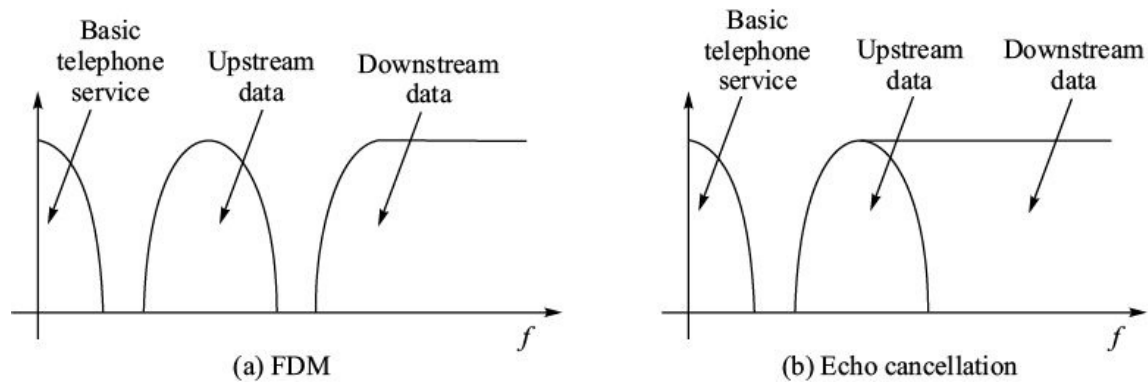


FIGURE 4.33 ADSL channel configuration.

4.7.2 Modulation

There are two modulation techniques used for ADSL systems:

- Carrierless Amplitude and Phase (CAP) modulation
- Discrete Multitone (DMT) modulation.

Carrierless Amplitude and Phase (CAP) modulation is similar to QAM. Unlike DMT, it is single carrier technique that generates wide band modulated signal. Since the cable attenuation and phase characteristics are not uniform over the entire band, wide band adaptive equalizers are required. It is difficult to scale and susceptible to narrow band interference. CAP is not much used in the industry.

Discrete Multitone (DMT) modulation scheme splits the available bandwidth into number of sub-channels having 4 kHz bandwidth (Figure 4.34). Each 4kHz sub-channel consists of a modulated carrier. 64 QAM or QPSK modulation is used depending on the quality of the sub-channel. A sub-channel having poorer transmission characteristics is modulated with QPSK. The basic DMT scheme works as under:

- On initialization, the ADSL modulator sends out a test signal on each sub-channel to determine signal to noise ratio.
- The ADSL modem assigns more bits to sub-channels having better transmission characteristics. Thus each sub-channel carries a different bit rate depending on the quality of the sub-channel. Number of bits per symbol per sub-channel is 2 to 15 bits. It is possible that some of the sub-channels may not carry any data bit due to their poor quality.

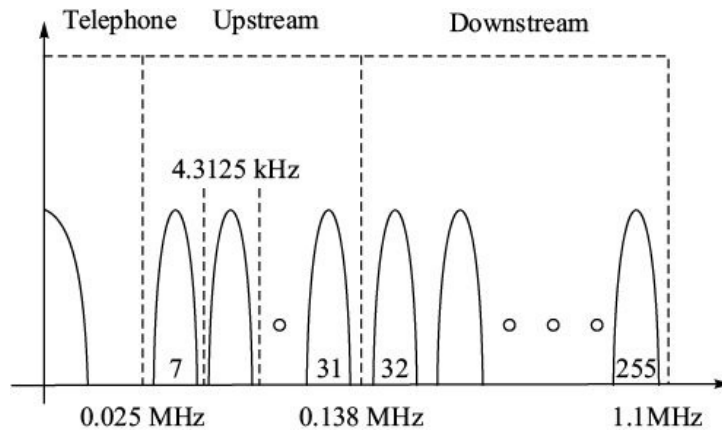


Figure 4.34 DMT sub-channel allocation of G.992.1.

ADSL uses forward error correction (FEC) based on Reed Soloman code. The FEC code is convolutionally interleaved with data to take care of burst errors.

4.7.3 Distance Limitations Distance limitations of ADSL depend on the transmission characteristics of the available copper media. Typical values are shown in Table 4.15.

TABLE 4.15 Distance Limitations of ADSL		
Data rate (Mbps)	Wire (mm)	Distance (km)
2.048	0.5	5.5
2.048	0.4	4.6
6.1	0.5	3.7
6.1	0.4	2.7

4.7.4 ITU-T Recommendations for ADSL

The ITU-T recommendations for ADSL are:

- ADSL G.992.1 G.dmt, 1999
- ADSL G.992.2 G.lite, 1999

G.992.1. The full rate standard of ITU-T, G.992.1 specifies 256 sub-channels in the frequency band 0–1.1 MHz (Figure 4.34). The frequency band up to 25 kHz (corresponding to sub-channels 0–6) is used for telephone service and as guard band. The rest of the sub-channels are used as indicated in Table 4.16. The gross

downstream bit rate with 2 to 15 bits/symbol/sub-channel works out to be from 64 kbps to 8.192 Mbps. The gross upstream bit rate works out to be from 16 kbps to 768 kbps.

G.992.2. The second ADSL standard of ITU-T called G.lite (G.922.2) specifies 128 sub-channels in the frequency band 0–550 kHz. Frequency band of sub-channels 0–6 is used for voice and as guard band as before. The allocation of the rest of the sub-channels of G.lite is given in Table 4.16. G.992.2 uses only 2 to 8 bits per symbol per sub-channel. The gross downstream bit rate with 2 to 8 bits/symbol/sub-channel works out to be from 64 kbps to 1.5 Mbps. The gross upstream bit rate works out to be from 16 kbps to 368 kbps.

New standards of ADSL are under development by ITU-T. G.992.5 ADSL 2 Plus will extend the number of sub-channels to 512 using the copper pair bandwidth up to 2.2 MHz. The gross downstream data rates will be in the range of 20 Mbps on phone lines as long as 1500 metres.

TABLE 4.16 Sub-channel Allocation in ITU-T G.992.1 and G.992.2

ITU Standard	Data stream	Sub-channels	Frequency band
ITU-T G.992.1	Upstream	7–31	25–138 kHz
	Down stream	32–255	0.138–1.1 MHz
	Down stream (Echo-cancellation)	7–255	0.025–1.1 MHz
ITU-T G.992.2	Upstream	7–31	25–138 kHz
	Down stream	32–127	0.138–0.55 MHz

4.7.5 xDSL Technologies

ADSL is one of the several access network technologies developed for providing data service on the existing copper cable infrastructure of the telephone network. These technologies are collectively referred to as xDSL. Table 4.17 summarizes the features of these DSL technologies.

TABLE 4.17 DSL Technologies

	ADSL	HDSL	SDSL	VDSL

Services	Voice, data	Data	Data	Voice, data
Data rate	Asymmetric	Symmetric	Symmetric	Asymmetric
Upstream data rate	768 kbps	E1	E1	~2.3 Mbps
Downstream data rate	8.192 Mbps	E1	E1	< 55 Mbps
Copper pairs	1	2	1	1
Modulation	Analog	Baseband	Baseband	Analog
	(CAP/DMT)	2B1Q	2B1Q	DMT

HDSL (High data rate DSL). HDSL was developed by Bellcore to provide cost effective means of extending 1.544 Mbps T1 service. It was standardized for T1 service by ANSI in US and for E1 service by ETSI in Europe. HDSL uses two pairs of copper wires, each carrying bit rate of 784 kbps in the case of T1, and 1168 kbps in the case of E1. ETSI also specified a three pair version of HDSL for E1 with each pair carrying bit rate of 784 kbps. With adaptive equalization and 2B1Q coding, HDSL can cover a distance of 3.7 km. There is no provision for voice communication in HDSL. Single-pair version HDSL, called HDSL-2 is under development.

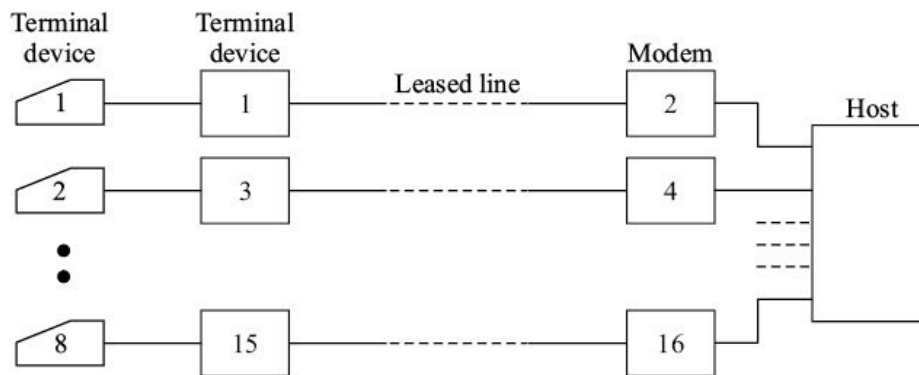
SDSL (Symmetric DSL). SDSL was developed as an alternative to HDSL as HDSL requires two pairs of wires. SDSL uses 2B1Q line coding and maximum bit rate is 1.544 Mbps (T1) or 2.048 Mbps (E1) full-duplex on a single pair of wires. It allows equal data rates in both the directions, upstream from the subscriber and downstream to the subscriber. It supports data service only. Echo-cancellation technique is used for separating the upstream and downstream flows. It is useful in those applications where traffic is symmetrical, *e.g.* web hosting, file transfer, *etc.*

VDSL (Very high data rate DSL). VDSL service is similar to ADSL (Voice plus asymmetrical data rates) but offers very high data rates over short stretches of copper pairs. The standard for VDSL is under development. The maximum

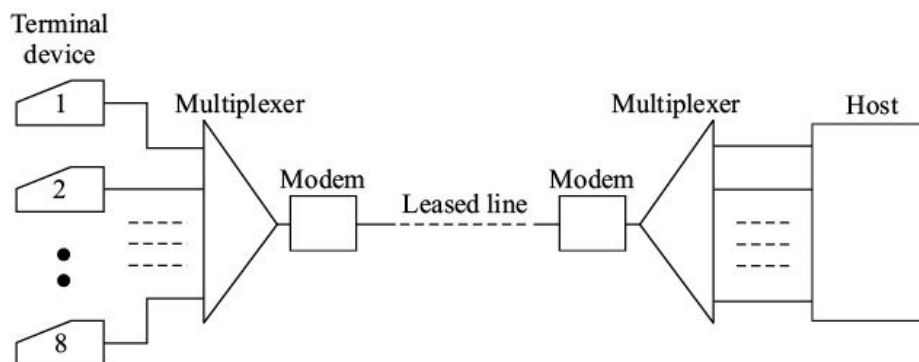
downstream data rate under consideration is between 51 Mbps to 55 Mbps over distance of 300 metres. The upstream data rate will be from 1.6 to 2.3 Mbps.

4.8 DATA MULTIPLEXERS

A *modem* is an intermediary device which is used for interconnecting terminals and computers when the distances involved are large. Another *data transmission* intermediary device is the data multiplexer which allows sharing of the transmission media. Multiplexing is adopted to reduce the cost of transmission media and modems. Figure 4.35 shows a simple application of data multiplexers. In the first option, 16 modems and eight leased lines are required for connecting eight terminals to the host.



(a) Multiple connections to host without multiplexers



(b) Multiple connections to host using multiplexers

FIGURE 4.35 Use of multiplexers for sharing media and modems.

In the second option, the terminals and the host are connected using two data multiplexers. The modem requirement is reduced to two and the leased line requirement is reduced to one. However, there is possibility of catastrophic

failure. If any of the multiplexers or the leased line fails, all the terminals will be cut off from the host.

Besides consideration of economy, the other benefit of multiplexing is centralized monitoring of all the channels. Data multiplexers can be equipped with diagnostic hardware/software for monitoring the performance of individual data channels.

The multiplexer ports which are connected to the terminals are called *terminal ports* and the port connected to the leased line is called the *line port*. A multiplexer has a built-in demultiplexer also for the signals coming from the other end.

4.8.1 Types of Data Multiplexers Like speech channel

multiplexing, data multiplexers use either Frequency Division Multiplexing (FDM) or Time Division Multiplexing (TDM). In FDM, the line frequency band is divided into sub-channels. Each terminal port is assigned one sub-channel for transmission of its data. In TDM, the sub-channels are obtained by assigning time intervals (time slots) to the terminals for use of the line. Time slot allotment to the sub-channels may be fixed or dynamic. A time division multiplexer with dynamic time slot allotment is called Statistical Time Division Multiplexer, (or STDM or statistical multiplexer in short).

In the following sub-sections we will briefly introduce the frequency division and time division multiplexers. Statistical multiplexer is more common than these two types of multiplexers. It is described in detail in the next section.

4.8.2 Frequency Division Multiplexers (FDMs) The leased line usually provides speech channel bandwidth of 300–3400 Hz. Therefore, most of the multiplexers are designed for this band. For frequency division multiplexing, the frequency band is divided into several sub-channels separated by guard bands. The sub-channels utilize frequency shift keying for modulating the carrier. Aggregate of all sub-channels is within the speech channel bandwidth and is an analog signal. Therefore, the

multiplexer does not require any modem to connect it to the line. A four-wire circuit is always required for outgoing and incoming channels.

Bandwidths of the sub-channels depend on the baud rates. Frequency division data multiplexers provide baud rates from 50 to 600 bauds. The number of sub-channels varies from thirty-six to four depending on baud rate (Table 4.18).

TABLE 4.18 Frequency Division Multiplexers		
Data rate (bps)	Number of sub-channels	Aggregate bit rate
50	36	1,800
75	24	1,800
110	18	1,980
150	12	1,800
600	4	2,400

Multidrop operation of the frequency division multiplexer is shown in Figure 4.36. Each remote terminal is connected through a single channel unit which transmits and receives a different frequency. The multiple line unit which is connected to the host separates the signals received on the line. It also carries out frequency division multiplexing of the outgoing signals.

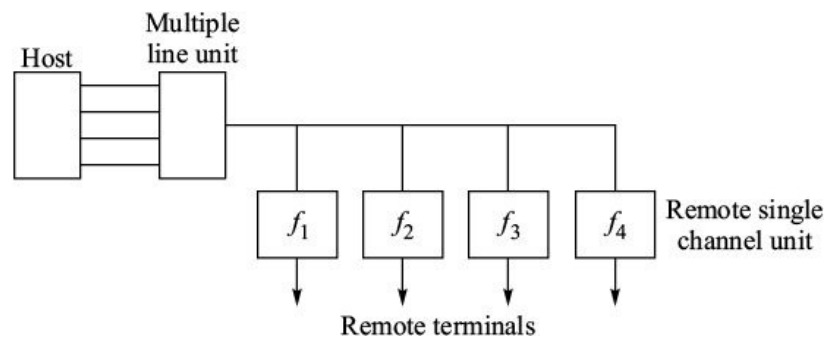


FIGURE 4.36 Multidrop application of frequency division multiplexers.

Frequency division multiplexers are not much in use today. Their major limitations are:

- Their production costs are high because of analog components.
- Total capacity is limited to 2400 bps due to large wasted bandwidth in the guard bands.
- They usually require a conditioned line.
- Most multiplexers do not allow mixing of bit rates of the sub-channels, *i.e.*

all the sub-channels have the same bit rate.

- They are inflexible. If the sub-channel capacity has to be changed, hardware modifications are required.

One advantage of frequency division multiplexers is that they are robust. Failure of one channel does not affect other sub-channels.

4.8.3 Time Division Multiplexers (TDMs) A time division multiplexer uses a fixed assignment of time slots to the sub-channels. One complete cycle of time slots is called a *frame* and the beginning of a frame is marked by a synchronization word (Figure 4.37). The synchronization word enables the demultiplexer to identify the time slots and their boundaries. The first bit of the first time slot follows immediately after the synchronization word.

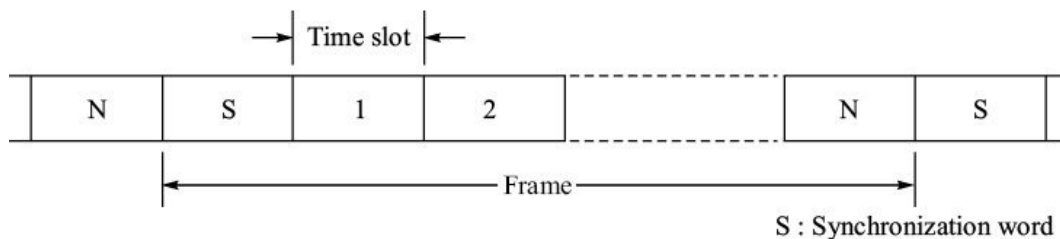


FIGURE 4.37 Frame format of time division multiplexer.

If all the sub-channels have the same bit rates, all the time slots have the same lengths. If the multiplexer permits speed flexibility, the higher speed sub-channels have longer time slots. The frame format and time slot lengths are, however, fixed for any given configuration or number of sub-channels and their rates. Since the frame format is fixed, time slots of all the sub-channels are always transmitted irrespective of the fact that some of the sub-channels may not have any data to send.

Bit and byte interleaved TDM. Time division multiplexers are of two types:

1. Bit interleaved multiplexer
2. Byte interleaved multiplexer.

In the *bit interleaved multiplexer*, each time slot is one bit long. Thus, the user data streams are interleaved taking one bit from each stream. Bit interleaved

multiplexers are totally transparent to the terminals.

In the *byte interleaved multiplexer*, each time slot is one byte long. Therefore, the multiplexed output consists of a series of interleaved characters of successive sub-channels. Usually, a buffer is provided at the input of each of its ports to temporarily store the character received from the terminal. The multiplexer reads the buffers sequentially. The start-stop bits of the characters are stripped during multiplexing and again reinserted after demultiplexing. It is necessary to transmit a special 'idle' character when a terminal is not transmitting.

The bit rate at the output of the multiplexer is slightly greater than the aggregate bit rate of the sub-channels due to the overhead of the synchronization word. The time division multiplexers do not have any provision for detecting or correcting the errors. Time division multiplexers permit the mixing of bit rates of the sub-channels. Their line capacity utilization is also better than frequency division multiplexers.

4.9 STATISTICAL TIME DIVISION MULTIPLEXERS

Statistical time division multiplexer uses dynamic assignment of time slots for transmitting data. If the sub-channel has data waiting to be transmitted, the multiplexer allots it a time slot in the frame (Figure 4.38). Duration of the time slot may be fixed or variable. There is need to identify the time slots and their boundaries. Therefore, some additional control fields are required. When we examine the protocols later, we will see how the time slots are identified.

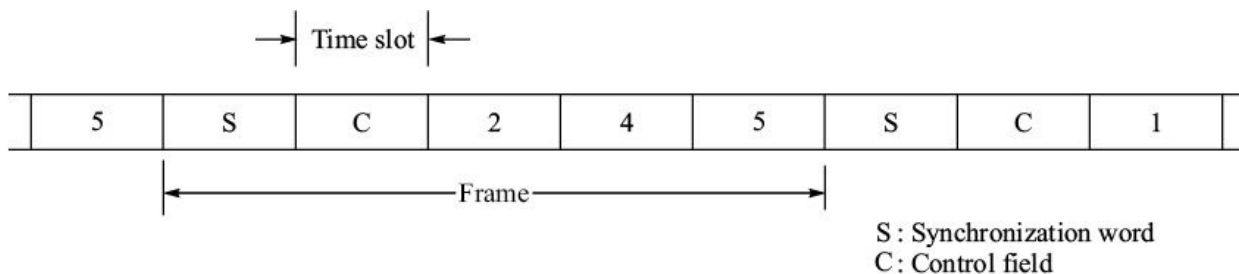


FIGURE 4.38 Frame format of statistical time division multiplexer.

Dynamic assignment allows the aggregate bit rates of the sub-channels to be more than the line speed of the statistical multiplexer considering that all the terminals will not generate traffic all the time. If sufficient aggregate traffic is assured at the input, the statistical multiplexer permits full utilization of the line

capacity. It is not so in the case of time division multiplexers, where the line time is wasted if a time slot is not utilized by a sub-channel, though another sub-channel may have data to send.

4.9.1 Buffer

A statistical multiplexer is configured to handle an aggregate sub-channel bit rate which is more than the line rate. It must have a buffer so that it may absorb the input traffic fluctuations maintaining a constant flow of multiplexed data on the line. Buffer size may vary from vendor to vendor but 64 kB is typical. This buffer is usually shared by both the directions of transmission, *i.e.* by the multiplexer and the demultiplexer portions of a statistical multiplexer. To guard against the overflow, the sub-channel traffic is flow-controlled.

4.9.2 Protocol

Some of the important issues which need to be addressed to have dynamic time slot allotment are:

1. In simple time division multiplexer, the location of time slot with respect to the synchronization word identifies the time slot because fixed frame format is used. But in a statistical multiplexer, the frame has variable format. Therefore, some mechanism for identifying the time slots is required.
2. Lengths of the time slots are variable. There is need to define time slot delimiters.

There are several proprietary statistical multiplexing approaches but none of them is standard. The two common approaches are bit map and multiple-character.

4.9.3 Bit Map Statistical Multiplexing In the *bit map statistical multiplexing*, the multiplexed data frame consists of a map field and several data fields (Figure 4.39). The map field has one bit for each sub-channel. It is two bytes long for the sixteen-port statistical multiplexer. If a bit is 1 in the map field, it indicates that the frame contains data field of the corresponding sub-channel. A zero in the map field of a frame indicates that data

field of the corresponding sub-channel is missing from this particular frame.

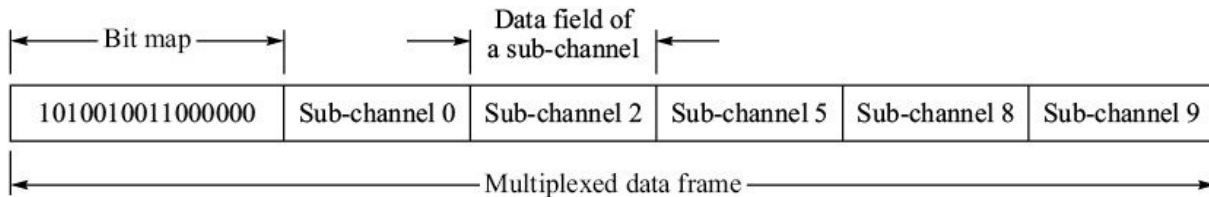


FIGURE 4.39 Bit map multiplexed data frame.

Note that the map field is present in all frames and has fixed length. The size of data field of a channel, if present, is fixed in the frame. It can be set to any value while configuring the statistical multiplexer. Fixed sizes of the data field enable the receiving demultiplexer to identify the boundaries of these fields. For asynchronous terminal ports, the data field size is usually set to one character. The start-stop bits are stripped before multiplexing and are reinserted after demultiplexing.

4.9.4 Multiple-Character Statistical Multiplexing The bit map multiplexing has one limitation. The number of bytes in the data field of a sub-channel cannot be varied from frame to frame. Multiple-character statistical multiplexing overcomes this limitation by including additional fields in the frame for indicating the sizes of the various data fields. The frame format used in multiple-character multiplexing is shown in (Figure 4.40).

The data field of a sub-channel that is present in a frame is identified by the sub-channel identifier of four bits. Thus, there can be a maximum of 16 sub-channels.

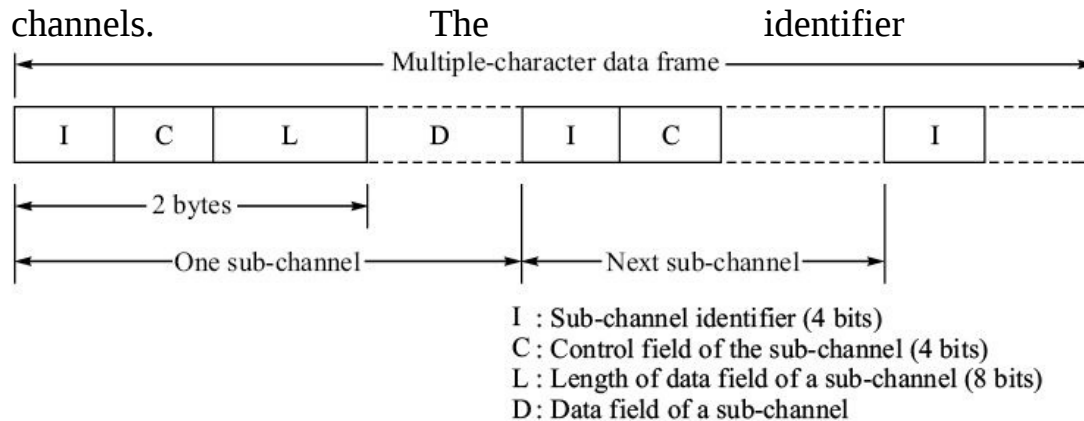


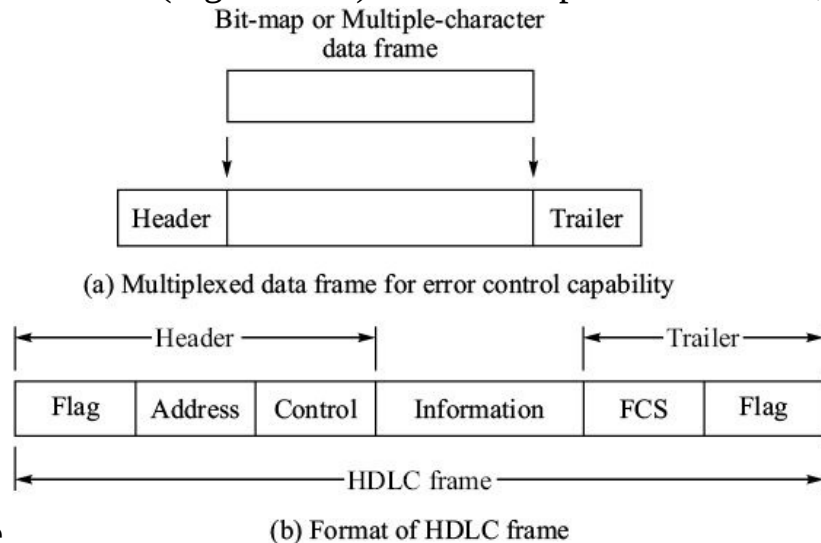
Figure 4.40 Multiple-character statistical multiplexer frame.

is followed by a four-bit sub-channel control field for management purposes. The control field is followed by a length field which indicates the number of bytes in the data field of the sub-channel. The length field is also one byte long and, therefore, there can be maximum 256 bytes per sub-channel per frame. The data field follows immediately after the length field. The format is repeated for each sub-channel in the frame.

4.9.5 Multiplexed Data Frame for Error Control Capability

A statistical multiplexer can have enhanced capability of error control. A commonly implemented protocol for error control is High Level Data Link Control (HDLC). HDLC protocol is reverse error correction method and is described in Chapter 9. It is sufficient to understand at this juncture that HDLC uses synchronous transmission with flags and CRC check bits for error detection.

Multiplexed data frame with error control capability consists of three parts: header, information, and trailer (Figure 4.41). If HDLC protocol is used, the



header consists of three

Figure 4.41 Frame structure for statistical multiplexing with error control.

fields, flag, address, and control fields. Flag identifies start of frame. Flag is followed immediately by the address and control fields. These fields are used for frame numbering, flow control, requesting retransmission of a frame. The trailer contains the CRC check bits for error detection. The information part of the frame contains bit-map or multiple-character sub-frame depending on the type of multiplexing scheme adopted.

4.9.6 Line Utilization Efficiency The frame overheads bits (control bits, header, trailer) do not contain user data and their transmission does not serve any purpose of the user. The maximum line utilization efficiency is, therefore, less than 100%, even if bits are continuously transmitted by the multiplexer and even if there are no transmission errors. Let us calculate the maximum line utilization efficiency for bit-map and multiple-character multiplexing schemes when HDLC framing is used.

Bit-map multiplexing. The HDLC frame contains seven overhead bytes**² (flag-1 byte, address-1 byte, control-1 byte, FCS-2 bytes, bit map-2 bytes). If there are N user data bytes in the frame, the maximum line utilization efficiency E is given by $E = \frac{N}{N+7}$

Multiple-character multiplexing. The HDLC frame in this case contains total overhead byte count of $5 + 2N$ bytes, where N is the number of sub-channels present in the frame. Therefore, the line utilization efficiency E is given by $E = \frac{\sum_N d_i}{5 + 2N + \sum_N d_i}$

where d_i is the number of data bytes in i th sub-channel.

EXAMPLE 4.4 A host is connected to 16 asynchronous terminals through a pair of statistical time division multiplexers utilizing the bit-map multiplexing. The sixteen asynchronous terminal ports operate at 1200 bps. The line port has a bit rate of 9600 bps. The data link control protocol is HDLC.

1. Calculate the maximum line utilization efficiency and throughput.
2. Will there be any queues in the multiplexer,
 - (a) if the average character rate at all the ports is 10 cps?
 - (b) if the host sends full screen display of average 1200 characters to each terminal?
3. How much time will the multiplexer take to clear the queues?

Solution

1. As $N = 16$, the line utilization efficiency is given by
 $E = 16/(7 + 16) = 0.696$
Throughput $T = E \cdot 9600 = 0.696 \cdot 9600 = 6678$ bps
2. (a) Aggregate average input = $16 \cdot 10 = 160$ cps
= $160 \cdot 8$ bps = 1280 bps
Since the throughput is 6678 bps, it is very unlikely there will be queues at the terminal ports.
(b) With start and stop bits, the minimum size of a character is 10 bits. Therefore, at 1200 bps, the host will take 10 seconds to transfer 1200 characters of one screen of a terminal. The multiplexer will get $1200 \cdot 16 = 19200$ characters in 10 seconds from the host.
Throughput = 6678 bps = $6678/8 = 834.75$ cps
Characters transmitted in 10 seconds = 8347.3 characters
Queue size at the end of 10 seconds = $19200 - 8347.5 = 10852.5$ characters
3. The multiplexer will take $10852.5/834.75 = 13$ additional seconds to clear the queue.

EXAMPLE 4.5 A host is connected to 16 asynchronous terminals through a pair of statistical time division multiplexers utilizing the multiple-character multiplexing. The asynchronous terminal ports operate at 1200 bps. The line port has a bit rate of 9600 bps. The data link control protocol is HDLC and the maximum size of the HDLC frame is 261 bytes.

1. Calculate the line utilization efficiency when all the ports generate their maximum traffic. Will queues develop for this load?
2. What is the maximum line utilization efficiency without having the queues?
3. If the host sends full screen display of average 1200 characters to each terminal, will there be any queue? If so, how much time will the statistical multiplexer take to clear the queue?

Assume start and stop bits of one bit duration each.

Solution

1. If all the 16 users simultaneously generate a burst of data, each HDLC frame will contain all the sub-channels. As the HDLC frame size is 261 bytes, each sub-channel will occupy $(261 - 5)/16 = 16$ bytes. The data field of each channel will be $16 - 2 = 14$ bytes. Therefore the line utilization

efficiency E is

$$E = \frac{16 \times 14}{261} = 0.8582$$

$$\text{Time to transmit one frame } t_0 = \frac{261 \times 8}{9600} = 217.5 \text{ ms}$$

Number of characters (n) received at each port in 217.5 ms is

$$n = 0.2175 \times 1200/10 = 26.1 \text{ characters}$$

But out of these only 14 characters are transmitted in each frame; so queues will develop.

2. If there are fewer sub-channels, the overhead of two bytes per sub-channel is reduced. Therefore, the line utilization efficiency may be increased. Let there be N sub-channels in a frame and d data bytes in each sub-channel.

$$\text{Size of the HDLC frame} = 5 + 2N + Nd$$

$$\text{Time to transmit the frame on the line} = \frac{(5 + 2N + Nd) \times 8}{9600} \text{ seconds}$$

Time taken by the terminals to generate d characters is $10d/1200$ seconds. If there are no queues, then

$$\frac{10d}{1200} = \frac{(5 + 2N + Nd) \times 8}{9600}$$

Simplifying, we get

$$d = (5 + 2N)/(10 - N), N \leq 10$$

We need to solve the above equation for integer values of d and N . Substituting the value of d in the equation for line utilization efficiency, we get

$$E = \frac{Nd}{5 + 2N + Nd}$$

$$E = N/10$$

As $N \leq 10$, maximum line utilization efficiency is obtained when $N = 9$.

Therefore,

$$E = 0.9, N = 9, d = 23$$

3. Time required by the host to transfer one screen = $1200 \times 10/1200 = 10$ seconds

$$\text{Number of characters to be transferred in 10 seconds} = 10 \times 1200 = 12,000$$

$$\text{Time taken to transmit one HDLC frame at 9600 bps} = \frac{261 \times 8}{9600}$$

Assuming all the sub-channels are present in the frame, the data character transfer rate per HDLC frame is 224 characters/frame. Therefore, number of data characters transferred in 10 seconds is

$$\frac{224 \times 10}{t_0} = 10298.85 \text{ characters}$$

where $t_0 = 0.2175$ second is the time taken to transmit one HDLC frame.

As the required character transfer rate is higher, there will be queues.

$$\text{Additional time required to clear the queues} = \frac{(19200 - 10298.85) \times 10}{10298.85} = 8.64$$

seconds

4.9.7 Comparison of Data Multiplexing Techniques When compared with other types of data multiplexers, statistical multiplexers offer many advantages. Table 4.19 gives a general comparison of the data multiplexing techniques. The parameters used for comparison are: Line utilization efficiency. It is the potential to effectively utilize the line capacity.

Channel capacity. It is the aggregate capacity of all the sub-channels.

High speed channels. This parameter compares the ability to support high speed data sub-channels.

Flexibility. This parameter compares the ability to change speed of sub-channels.

Error control. This parameter compares the ability to detect and correct transmission errors.

Multidrop capability. This parameter compares the ability to use multidrop techniques on a sub-channel.

Transmission delay. This parameter compares the additional transmission delays introduced by the multiplexer, over and above the propagation delay.

TABLE 4.19 Comparison of Data Multiplexing Techniques

Parameter	FDM	TDM	Stat Mux
-----------	-----	-----	----------

Line efficiency	Channel capacity	High speed sub-channel	Flexibility	Poor	Good	Excellent
Error control	Multidrop capability	Cost		Poor	Good	Excellent
Transmission delay				Very poor	Poor	Excellent
				Very poor	Good	Excellent
				None	None	Possible
				Good	Difficult	Possible
				High	Low	Medium
				None	Low	Random

SUMMARY

Transmission of digital signals using the limited bandwidth of the speech channel of the telephone network necessitates use of digital modulation methods, namely, FSK, PSK, and QAM. Modems and Digital Subscriber Line (DSL) systems use these modulation techniques to send data signals over copper pairs.

A modem has two interfaces, a digital interface, which is connected to the Data Terminal Equipment (DTE) and a line interface which is connected to the line. It consists of several functional blocks besides a modulator and a demodulator. Encoding, scrambling, equalizing, and timing extraction are some of the additional functions carried out in a modem. ITU-T has standardized modems for bit rates starting from 300 bps to 56 kbps for voice channels. These modems are full duplex or half duplex and can work on 2-wire or 4-wire circuits. Limited distance modems, baseband modems and line drivers are designed for copper cable connection between the modems. These modems require the wider bandwidth of the cable and cannot work within the 300–3400 Hz band of the speech channel.

Digital Subscriber Line (DSL) systems are deployed between the telephone

exchange and customer's premises to provide high speed data access on the existing copper pair of the telephone connection. There are several DSL technologies that are collectively referred to as xDSL. ADSL, Asymmetric Digital Subscriber Line is the most widely deployed DSL technology. It uses frequency band of 25 kHz to 138 kHz for upstream data signals and 138 kHz to 1.1 MHz for the downstream data signals. The gross downstream bit rate can be from 64 kbps to 8.192 Mbps and the gross upstream bit rate can be from 16 kbps to 768 kbps. The ITU-T recommendations for ADSL are G.992.1 and G.992.2. G.992.2 is a thinned down version of G.992.1 and it has maximum downstream bit rate of 1.5 Mbps. Its maximum upstream bit rate is 368 kbps.

Data multiplexers are used to economize on lines and modems. Frequency division and time division data multiplexers offer limited capabilities and do not make optimum use of the channel capacity. Statistical time division multiplexers offer a very high potential utilization of channel capacity. They also offer high flexibility of configuring terminal port speeds.

EXERCISES

1. Tick the right answer.
 - (a) Full duplex mode of transmission is possible in
 - (i) 4-wire modems only
 - (ii) 2-wire modems only
 - (iii) 2-wire or 4-wire modems.
 - (b) The secondary channel in a modem uses the following:
 - (i) ASK
 - (ii) FSK
 - (iii) PSK.
 - (c) One of the functions of the training sequence is to
 - (i) test the modems
 - (ii) synchronize the descrambler
 - (iii) test the DTE.
 - (d) High-speed modems for switched telephone connections are equipped with
 - (i) adaptive equalizers
 - (ii) fixed equalizers
 - (iii) manually adjustable equalizers.
 - (e) Echo canceller is required in

- (i) 4-wire full duplex high speed modems
 - (ii) 2-wire half duplex high speed modems
 - (iii) 2-wire full duplex high speed modems.
2. In a differential BPSK modulator, binary 0 results in a phase change of p and binary 1 does not cause any phase change. Write the phase states of the carrier for the following binary sequence.
1 0 1 1 0 0 1
 3. Draw the phasor diagram of the output if the carrier is given a phase shift of $-p/2$ instead of $p/2$ in Figure 4.8.
 4. If each element of the QAM signal shown in Figure 4.14 has a time duration of 1 ms, what are the baud and bit rates?
 5. For the PSK modulator shown below, analog outputs of D/A converters are indicated in the table. Draw the phasor diagram of the output. If the bit rate is 4800 bps, what is the baud rate?

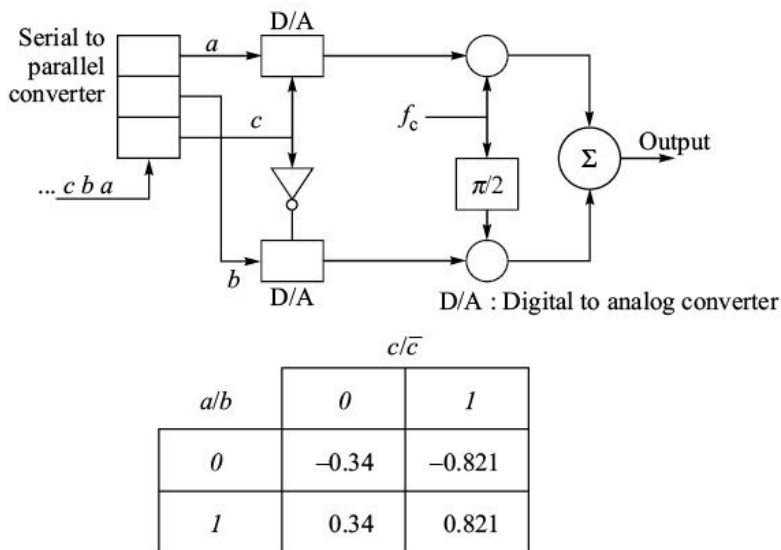


FIGURE E4.42.

6. Solve the bit map statistical multiplexer problem given in the chapter when the line speed is 19.2 kbps and the average number of characters in the full screen is 1920.
7. A statistical multiplexer based on multiple-character multiplexing protocol described in the chapter is connected to 16 asynchronous 1200 bps terminals. The line speed is 9600 bps. At data link layer, the statistical multiplexer utilizes HDLC protocol with 261 bytes frame length. If only one user is active, what is the maximum line utilization efficiency?

1 As an alternative, the recommendation provides for accepting from the terminal asynchronous data which is suitably converted for transmission synchronously. The internal clock of the modem is used for this purpose.

2 For maximum line utilization efficiency, we assume concatenation of HDLC frames so that the trailing flag also acts as leading flag for the next frame.

5

Error Control

In voice communication, the listener can tolerate a good deal of signal distortion and make sense of the received signal. Digital systems, on the other hand, are very sensitive to errors and may malfunction if the data is corrupted. Therefore, error control mechanisms are built into all digital systems. In this chapter, we discuss error control mechanisms required for data communications. We begin with basic concepts and terminology of error detection, error correction, and bit error rate. Then we discuss parity checking, checksum and cyclic redundancy check methods of error detection. We proceed to forward error correction methods which include block codes, Hamming code, and convolution code. Reverse error control mechanisms are extensively used in data communications. These are based on detection of errors in a message and its retransmission if errors are detected. We also introduce these mechanisms in this chapter.

5.1 TRANSMISSION ERRORS

Errors are introduced in the data bits during their transmission across a data network. These errors can be categorized as:

- content errors, and
- flow integrity errors.

5.1.1 Content Errors *Content errors are the errors in the content of a message, e.g. a binary 1 may be received as a binary 0. A block of data may have single bit error or multiple errors. Multiple errors in a data block are referred to as burst error. Length of burst error is defined from the first corrupted bit to*

the last corrupted bit. For example, burst error in Figure 5.1 is six bits long. An error burst may contain uncorrupted bits as well.

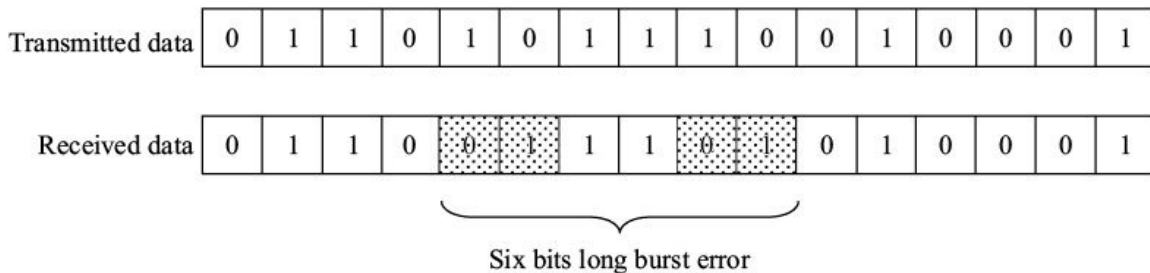


FIGURE 5.1 Burst error.

There can be several causes of content errors.

Signal impairment. Distortion of electrical signal and noise are the main causes of content errors. Thermal noise causes sporadic bit errors. An impulse can last long enough to corrupt several bits and therefore leads to burst errors.

Loss of synchronization. The receiver samples the received signal to extract the digital information by using a clock that is synchronized to the clock of the transmitter. The clock is extracted from the received signal using a filter and a phase lock loop. Too few transitions in the line signal may cause the phase lock loop to drift leading to loss of synchronization, which in turn results in sampling at wrong instants. Errors introduced are in form of burst till the synchronization is reestablished between transmitter and receiver clocks.

Scramblers. Scrambler is used to ensure synchronization of transmitter and receiver clocks. It randomizes the serial data before its transmission. Strings of zeros and ones are thus avoided in the transmitted signal. Scrambling is done using a shift register and exclusive OR gates. Scramblers have unfortunate property of multiplying errors. If an error is received, it gets multiplied as it shifts through the shift register. The errors multiplied depend on the number of taps taken from the shift register.

Transmission channel switching. Transmission channels of telecommunication network are usually protected, *i.e.* if the main path fails due to any reason (cut in fibre, fading of radio signal), the signals are switched over to standby path (alternate fibre route, alternate radio frequency). The switchover may take about 10 to 50 ms, and it causes momentary loss of digital signal. These momentary disturbances are unnoticeable for voice channels but lead to burst errors in data

signals.

5.1.2 Flow Integrity Errors Data is sent as data packets across a data network. It is not necessary that data packets sent by a source to the destination take the same path. In Figure 5.2, packet 1 goes via nodes A, B, and D. Packet 2 goes directly from node A to node D. Packet 3 starts its journey from nodes A to D via nodes B and C. Flow of these data packets from the source to the destination may be affected in several ways.

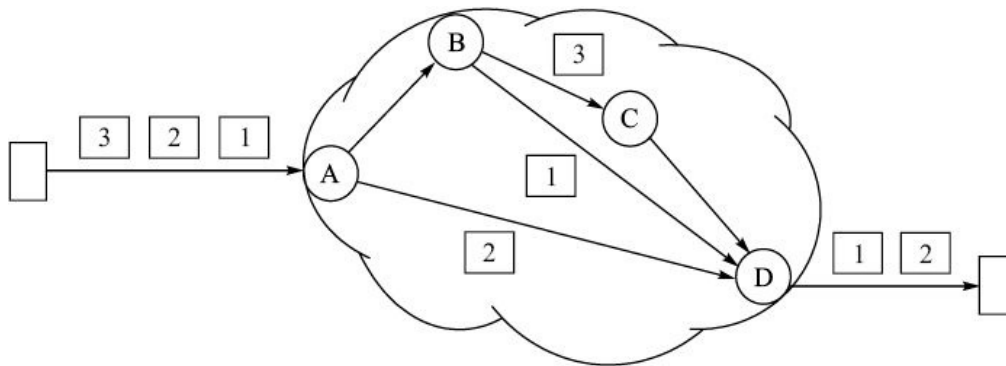


Figure 5.2 Flow integrity errors.

- The packets may be delivered out-of-sequence. Packet 2 takes shorter path and therefore it may arrive at node D ahead of packet 1.
- A data packet may not arrive at the destination at all. Packet 3 reaches up to node C, but is lost thereafter. Reasons of packet loss can be several:
 - A packet may get corrupted due to content errors and may be rejected by a node.
 - It may be delivered to wrong address due to errors that could not be detected.
 - A node may not forward the packet due to congestion in the network. When a node experiences congestion (i.e. it has long queue of packets), it starts discarding data packets.
 - Each data packet has limited lifetime in the network. This is called *time to live* (TTL). TTL can be defined in terms of number of hops. This is done to avoid circulating packets (packets that lose the way). A network node dumps all the TTL expired packets.
- The sender at times expects from the receiver an acknowledgement of having received a packet. If the acknowledge is not received within a

defined time interval, the sender resends the packet. It may happen the receiver sends the acknowledgement, but the acknowledgement from the receiver is lost in the network. In such situation, the receiver will receive duplicate data packets eventually.

Thus a packet may be delivered to a wrong destination, or may be delivered out of sequence, or may not be delivered, or delivered in duplicate. All these are instances of flow integrity errors.

5.1.3 Methods of Error Control Error control involves two basic steps, *error detection* and *error correction*. The content errors are detected by introducing additional check bits in a block of the data bits. The sender encodes the check bits in such a way that the receiver can be able to detect the content errors. There are coding schemes that enable the receiver to correct the error also. Alternatively, the receiver simply requests the sender for retransmission of the data block whenever it detects errors.

Flow integrity errors are dealt by building flow-regulating mechanisms for transport of data packets between two nodes. These mechanisms are implemented at various stages of data communications. We will concentrate on flow integrity errors that occur between two adjacent data nodes in this chapter. We will take up end-to-end flow integrity errors across the network later in the book.

5.2 CODING FOR DETECTION AND CORRECTION OF CONTENT ERRORS

For detection and correction of content errors, we need to add some check bits to a block of data bits. Check bits are so chosen that the resulting bit sequence has a unique ‘characteristic’ which enables error detection. Coding is the process of adding the check bits. Before we proceed further, let us familiarize ourselves with some of the terms relating to coding theory.

- The block of data bits to which check bits are added is called a *data word*.
- The bigger block containing check bits is called the *code word*.
- Hamming distance or simply distance between two code words is the

number of disagreements between them. For example, the distance between the two words given below is 3 (Figure 5.3).

- The weight of a code word is number of 1s in the code word, *e.g.* 11001100 has a weight of 4.
- A code set consists of all valid code words. All the valid code words have a built in ‘characteristic’ of the code set.

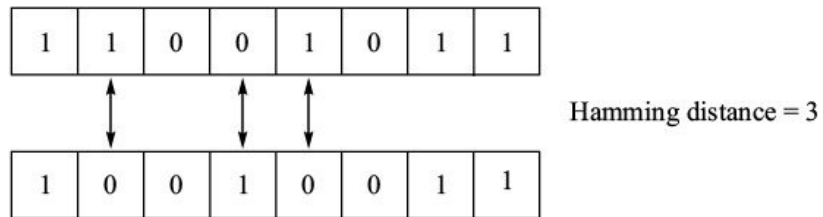


FIGURE 5.3 Hamming distance.

5.2.1 Error Detection When a code word is transmitted, one or more of its bits may be reversed due to signal impairment. The receiver can detect these errors only if the received code word is not one of the valid code words of the code set. If it is, errors cannot be detected. It implies that if length of code words is n bits, not all the 2^n combinations of n bits should be included in the code set.

If data word has length l , and r number of check bits are added to each data word to generate code words of length $n = l + r$, we have 2^l combinations of data words to be mapped to code words from 2^{l+r} possible combinations. By prudent selection of code words, we can ensure that up to certain numbers of bit errors in a code word, we will be able to always detect errors. Let us see how this can be achieved.

When errors occur, the distance between the transmitted and received code words becomes equal to the number of error in the received code word as shown in Table 5.1.

TABLE 5.1 Distance between Transmitted and Received Code Words			
Transmitted code word	Received code word	Number of errors	Distance
11001100	11001110	1	1
10010010	00011010	2	2
10101010	10100100	3	3

Therefore, it follows that the valid code words should have distance more than one among themselves. Otherwise, even a single bit error will generate another valid code word and the error will not be detected.

The number of errors which can be detected depends on the minimum distance between any two valid code words in the code set. For example, if the valid code words are separated by a distance of 4, up to three errors in a code word can be detected. In general, if we want to detect up to E errors, then all the valid code words should be at least $E + 1$ distance apart. By adding a certain number of check bits and properly choosing the algorithm for generating them, we ensure some minimum distance between any two valid code words of a code set.

5.2.2 Error Correction After an error is detected, there are two approaches to correction of errors:

- Reverse Error Correction (REC)
- Forward Error Correction (FEC).

In the first approach, the receiver requests for retransmission of the code word whenever it detects an error. If there is no error, the receiver returns an acknowledgement for the correctly received code words. Reverse error correction methods are extensively used in data networks. But these methods require the following conditions for their efficacy:

- There should be return channel from receiver to sender. The return channel should have low error rate, otherwise the retransmission requests and acknowledgements may be lost.
- High transmission time of onward and return channels imposes memory and throughput constraints as the code words must be stored by the sender till they are acknowledged.

In the second approach, the code is so designed that it is possible for the receiver to detect and correct the errors as well. The receiver locates the errors by analyzing the received code word and reverses the erroneous bits. An alternative way of forward error correction is to search for the most likely correct code word. When an error is detected, the distances of all the valid code words from the received invalid code word are measured. The nearest valid code

word is the most likely correct version of the received word (Figure 5.4). This method is called *maximum likelihood decoding*.

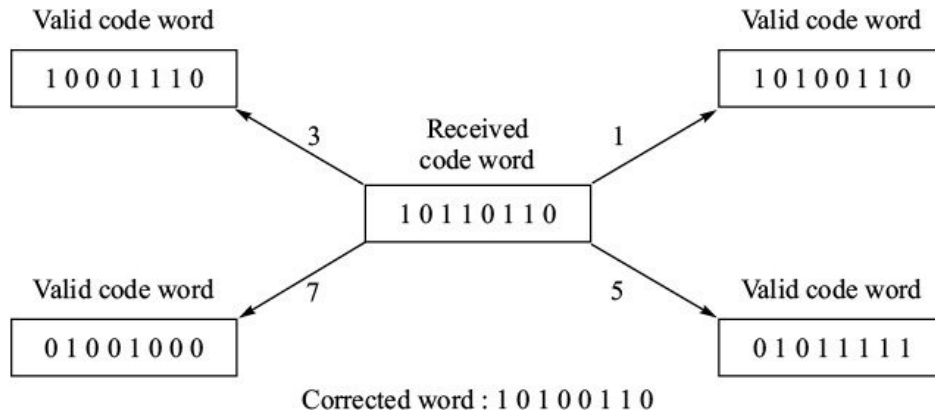


FIGURE 5.4 Error correction based on the maximum likelihood decoding.

If we want to correct up to E errors, the valid code words should be at least $2E + 1$ distance apart. In case of more than E errors, the received code word to be nearer to the wrong valid code word.

5.2.3 Perfect Error Correcting Code We are yet to address the issue of number of check bits required to correct a certain number of errors. Let us determine number of check bits (r) required for a data word of length l to correct a single-bit error in a code word of length $n = l + r$. We need to address this problem from two angles:

- The number of check bits should be sufficient to ensure that single-bit errors do not generate another valid code world. Otherwise we will not be able to detect the error.
- The number of check bits should be sufficient to point to location of the bit in error if there is an error.

Single-bit errors in one n -bit code word can generate n invalid code words. There are 2^l code words. Thus total space of invalid code words consists of $n2^l$ invalid words. The total number of invalid code words and valid code words cannot exceed 2^n combinations possible with n bits. Therefore, $2^n \geq 2^l + n2^l$ or

$$2^{l+r} \geq 2^l + n2^l \text{ or}$$

$$2^r \geq n + 1$$

Let us look at it from another angle. For the code word of length n , a single-bit error can occur in any one of the n bitpositions. The check bits should uniquely indicate which bit position is in error. The check bits should also indicate if there is no error in the code word. The r check bits are therefore required to indicate $n + 1$ states of a received code word, no-error state and n single error locations. There are 2^r check bit combinations, therefore we need r check bits so that $2^r \geq n + 1$.

Note that the inequality is same from both the angles. A perfect error correcting code is the one in which number of check bits is equal to the minimum integer value of r that satisfies the above inequality. For example, if $n = 7$, the minimum value of r is 3, *i.e.* the perfect code will require 3 check bits for coding 4-bit data words. Hamming code, discussed later, is a perfect code.

5.2.4 Systematic Code Error detection and correction involves addition of check bits. A *systematic code* is a code in which each code word includes the unaltered data word followed or preceded by a separate group of check bits.

5.2.5 Bit Error Rate (BER) In analog transmission, signal quality is specified in terms of signal-to-noise ratio (S/N) which is usually expressed in decibels. In digital transmission, the quality of received digital signal is expressed in terms of Bit Error Rate (BER), which is the number of errors in a given number of transmitted bits. A typical error rate on a high quality leased telephone line is as low as 1 error in 10^6 bits or simply $1 \cdot 10^{-6}$.

Just like BER, Character Error Rate (CER) and Frame Error Rate (FER) can be defined. CER is the average number of characters received with at least one error in a large sample of transmitted characters. The probability of having at least one error in a byte calculated in Example 5.1 gives a CER of 8 in 10^5 characters. FER, likewise, refers to the average number of frames received with at least one error in a large sample of transmitted frames. It can also be calculated on the same lines as CER. For low values of BER, the CER and FER

can be calculated from BER as below: $CER = b \text{ BER}$

$$FER = f \text{ BER}$$

where b is the number of bits per character and f is the number of bits per frame.

Whatever be the methods of error control, errors cannot be completely eliminated. There is always some residual error which goes undetected. Residual Error Rate (RER) refers to the error rate in the data bits after error control has been performed.

EXAMPLE 5.1 If the average BER is 1 in 10^5 , what is the probability of having
(a) single bit error,
(b) single bit correct,
(c) at least one error in an eight-bit byte?

Solution

(a) Probability of having single bit error $1/10^5 = 0.00001$

(b) Probability of having single bit correct $1 - 0.00001 = 0.99999$

(c) Probability of having one or more errors in an 8-bit byte $1 - (0.99999)^8 = 0.00008$.

5.3 ERROR DETECTION METHODS

The popular error detection methods are:

- Parity checking
- Checksum error detection
- Cyclic Redundancy Check (CRC).

Error detection is carried out by adding check bits to the data word in each of the above three methods. The check bits are called parity bits, checksum bits or CRC bits depending on the method used to generate them. Each method has its advantages and limitations as we shall see in the following sections.

5.3.1 Parity Checking In *parity checking methods, an additional bit called *parity bit* is added to each data word. The additional bit is so chosen that the weight of the code word thus formed is*

either even (even parity) or odd (odd parity) (Figure 5.5). All the code words of a code set have the same parity (either odd or even) which is decided in advance.

<i>Even parity</i>		<i>Odd parity</i>	
P	Data word	P	Data word
0	1 0 0 1 0 1 1	1	1 0 0 1 0 1 1
1	0 0 1 0 1 1 0	0	0 0 1 0 1 1 0

P : Parity bit

FIGURE 5.5 Even and odd parity bits.

When a single error or an odd number of errors occurs during transmission, the parity of the code word changes (Figure 5.6). Parity of the code word is checked at the receiving end and violation of the parity rule indicates errors somewhere in the code word.

Transmitted code word	1 0 0 1 0 1 1 0	Even parity
Received code word (Single error)	0 0 0 1 0 1 1 0	Odd parity (Error is detected)
Received code word (Double error)	0 0 0 1 1 1 1 0	Even parity (Error is not detected)

Figure 5.6 Error detection by change in parity.

Note that double or any even number of errors will go undetected because the resulting parity of the code word will not change. Thus, a simple parity checking method has its limitations. It is not suitable for multiple errors. To keep the possibility of occurrence of multiple errors low, the size of the data word is usually restricted to a single byte.

Parity checking does not reveal the location of the erroneous bit. The received code word with an error is always at equal distance from two valid code words. Therefore, errors cannot be corrected by the parity checking method. The main advantage of using parity is that we can generate the parity without need for additional storage or computational overhead.

EXAMPLE 5.2 Write the ASCII code of the word ‘HELLO’ using even parity.

	8	7	6	5	4	3	2	1	Bit positions*
H	0	1	0	0	1	0	0	0	
E	1	1	0	0	0	1	0	1	
L	1	1	0	0	1	1	0	0	
L	1	1	0	0	1	1	0	0	
O	1	1	0	0	1	1	1	1	

Solution

* The parity bit is at eighth bit position.

5.3.2 Checksum Error Detection In checksum error detection method, a checksum is transmitted along with every block of data bytes. Eight-bit bytes of a block of data are added in an eight-bit accumulator. Checksum is the resulting sum in the accumulator. Being an eight-bit accumulator, the carries of the most significant bits are ignored.

EXAMPLE 5.3 Find the checksum of the following message. The MSB is on the left-hand side of each byte.

10100101 00100110 11100010 01010101 10101010 11001100 00100

	1	1			1					}	Carries
			1	1	1	1	1	1			
	1	0	1	0	0	1	0	1			
	0	0	1	0	0	1	1	0			
	1	1	1	0	0	0	1	0			
	0	1	0	1	0	1	0	1			
	1	0	1	0	1	0	1	0			
	1	1	0	0	1	1	0	0			
	0	0	1	0	0	1	0	0			
	1	0	0	1	1	1	0	0			Checksum byte

After transmitting the data bytes, the checksum is also transmitted. The checksum is regenerated at the receiving end and errors show up as a different checksum. Further simplification is possible by transmitting the 1's or 2's complement of the checksum in place of the checksum itself. The receiver in this case accumulates all the bytes including the complement of the checksum. If there is

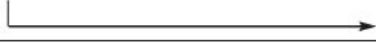
no error, the contents of the accumulator should be zero after accumulation of the complement of the checksum byte.

The advantage of this approach over simple parity checking is that 8-bit addition ‘mixes up’ bits and the checksum is representative of the overall block. Unlike simple parity where even number of errors may not be detected, in checksum there is 255 to 1 chance of detecting random errors. Checksum is suitable even for burst errors. It is easily implemented in software. Internet protocol of TCP/IP suite uses checksum for error detection in IP packets.

Example 5.4 Find the checksum byte using modulo-256 addition of the following data bytes: 10101010 11001100

Solution	1								Carries	
		1	0	1	0	1	0	1	0	Data word
		1	1	0	0	1	1	0	0	Data word
		0	1	1	1	0	1	1	0	Checksum

Example 5.5 The following data bytes received contain checksum byte as the last byte. The checksum was generated using 1’s complement. Is there any error in the bit sequence?

										10101010 11001100 10001000	
	1									Carries	
		1	0	1	0	1	0	1	0	First byte	
		1	1	0	0	1	1	0	0	Second byte	
		1	0	0	0	1	0	0	0	Checksum	
Solution	1	1	1	1	1	1	1	1	0	Sum	
											
		1	1	1	1	1	1	1	1		
		0	0	0	0	0	0	0	0	1’s complement	

There is no error in the received bit stream.

Transport protocol checksum. It may be noted in Example 5.3 that the checksum will remain same even if the bits interchange their position within a column. If such errors occur during transmission, the received checksum will fail to reflect any error. A more powerful method of generating checksum is used in the transport protocol standardized by ISO and ITU-T in IS 8072 and X.224. In

this method, two checksum bytes are generated instead of the usual one. To understand the algorithm, consider that $B_1, B_2, B_3,$ and B_4 are four data bytes, and X and Y are the two checksums. The checksums are so generated that when there are no errors, the following summations S and S are zero at the receiving end.

$$S = B_1 + B_2 + B_3 + B_4 + X + Y = 0$$

$$S = 6B_1 + 5B_2 + 4B_3 + 3B_4 + 2X + Y = 0$$

Therefore,

$$X = -(5B_1 + 4B_2 + 3B_3 + 2B_4) \quad Y = 4B_1 + 3B_2 + 2B_3 + B_4$$

The above formulae can be used at the transmitting end to calculate checksums, X and Y . Alternatively, we can perform summation for S and S at transmitting end with initial values of X and Y taken as zero.

$$S = B_1 + B_2 + B_3 + B_4$$

$$S = 6B_1 + 5B_2 + 4B_3 + 3B_4$$

Now X and Y can be expressed in terms of S and S .

$$X = S - S$$

$$Y = S - 2S$$

The procedure can be generalized for calculating the checksum bytes. It involves the following steps:

1. To start with, assume checksum bytes X and Y are 00000000.
2. Define variables P_i and Q_i such that

$$P_i = P_{i-1} + B_i \quad B_i = \text{ith data byte}$$

$$Q_i = Q_{i-1} + P_i \quad P_0 = Q_0 = 0$$
3. P_i and Q_i are calculated for all the n data bytes in the message and for the initial values of the checksum bytes X and Y . Note that P_{n+2} and Q_{n+2} are S and S respectively.
4. From the resulting values of P_i and Q_i variables, calculate the final values of checksum bytes X and Y as below:

$$X = P_{n+2} - Q_{n+2}$$

$$Y = Q_{n+2} - 2P_{n+2}$$

These checksum bytes are transmitted along with the data bytes.

5. At the receiving end, steps 2 and 3 are carried out again and variables P_{n+2} and Q_{n+2} are generated. If there are no errors in the received data block, the values of these variables will be zero.

EXAMPLE 5.6 Generate the checksum bytes as per the transport protocol for the following data bytes: 10100101 00100110 11100010 01010101

Regenerate variables P_i and Q_i at the receiving end and show that their final values are zero if there are no errors.

Solution

(a) Transmitting end

	P_0	0 0 0 0 0 0 0 0					
1 0 1 0 0 1 0 1	B_1	<u>1 0 1 0 0 1 0 1</u>		Q_0	0 0 0 0 0 0 0 0		
	P_1	1 0 1 0 0 1 0 1	→		<u>1 0 1 0 0 1 0 1</u>		
0 0 1 0 0 1 1 0	B_2	<u>0 0 1 0 0 1 1 0</u>		Q_1	1 0 1 0 0 1 0 1		
	P_2	1 1 0 0 1 0 1 1	→		<u>1 1 0 0 1 0 1 1</u>		
1 1 1 0 0 0 1 0	B_3	<u>1 1 1 0 0 0 1 0</u>		Q_2	0 1 1 1 0 0 0 0		
	P_3	1 0 1 0 1 1 0 1	→		<u>1 0 1 0 1 1 0 1</u>		
0 1 0 1 0 1 0 1	B_4	<u>0 1 0 1 0 1 0 1</u>		Q_3	0 0 0 1 1 1 0 1		
	P_4	0 0 0 0 0 0 1 0	→		<u>0 0 0 0 0 0 1 0</u>		
0 0 0 0 0 0 0 0	X	<u>0 0 0 0 0 0 0 0</u>		Q_4	0 0 0 1 1 1 1 1		
	P_5	0 0 0 0 0 0 1 0	→		<u>0 0 0 0 0 0 1 0</u>		
0 0 0 0 0 0 0 0	Y	<u>0 0 0 0 0 0 0 0</u>		Q_5	0 0 1 0 0 0 0 1		
	P_6	0 0 0 0 0 0 1 0	→		<u>0 0 0 0 0 0 1 0</u>		
				Q_6	0 0 1 0 0 0 1 1		
	P_6	0 0 0 0 0 0 1 0					
	Q_6	<u>0 0 1 0 0 0 1 1</u>					
$X = P_6 - Q_6$	X	1 1 0 1 1 1 1 1					
	Q_6	0 0 1 0 0 0 1 1					
	$2 P_6$	<u>0 0 0 0 0 1 0 0</u>					
$Y = Q_6 - 2P_6$	Y	0 0 0 1 1 1 1 1					

(b) Receiving end Summation up to P_4 and Q_4 will be same as above. The rest of the steps are as follows:

Summation up to P_4 and Q_4 will be same as above. The rest of the steps are as follows:

	P_4	0 0 0 0 0 0 1 0			
1 1 0 1 1 1 1 1	X	<u>1 1 0 1 1 1 1 1</u>	Q_4	0 0 0 1 1 1 1 1	
	P_5	1 1 1 0 0 0 0 1	→	<u>1 1 1 0 0 0 0 1</u>	
0 0 0 1 1 1 1 1	Y	<u>0 0 0 1 1 1 1 1</u>	Q_5	0 0 0 0 0 0 0 0	
	P_6	0 0 0 0 0 0 0 0	→	<u>0 0 0 0 0 0 0 0</u>	
			Q_6	0 0 0 0 0 0 0 0	

Thus P_6 and Q_6 are zero, indicating that there is no error.

In the next example, we consider errors in the third bit position of the second and fourth bytes and try to detect the errors using the method just described. These errors cannot be detected by the usual checksum method as these errors will not change the sum of the third column.

EXAMPLE 5.7 Check whether the following bytes have any error. The last two bytes are the transport protocol checksum bytes.

10100101 00000110 11100010 01110101 11011111 00011111

	P_0	0 0 0 0 0 0 0 0			
1 0 1 0 0 1 0 1	B_1	<u>1 0 1 0 0 1 0 1</u>	Q_0	0 0 0 0 0 0 0 0	
	P_1	1 0 1 0 0 1 0 1	→	<u>1 0 1 0 0 1 0 1</u>	
0 0 0 0 0 1 1 0	B_2	<u>0 0 0 0 0 1 1 0</u>	Q_1	1 0 1 0 0 1 0 1	
	P_2	1 0 1 0 1 0 1 1	→	<u>1 0 1 0 1 0 1 1</u>	
1 1 1 0 0 0 1 0	B_3	<u>1 1 1 0 0 0 1 0</u>	Q_2	0 1 0 1 0 0 0 0	
	P_3	1 0 0 0 1 1 0 1	→	<u>1 0 0 0 1 1 0 1</u>	
0 1 1 1 0 1 0 1	B_4	<u>0 1 1 1 0 1 0 1</u>	Q_3	1 1 0 1 1 1 0 1	
	P_4	0 0 0 0 0 0 1 0	→	<u>0 0 0 0 0 0 1 0</u>	
1 1 0 1 1 1 1 1	X	<u>1 1 0 1 1 1 1 1</u>	Q_4	1 1 0 1 1 1 1 1	
	P_5	1 1 1 0 0 0 0 1	→	<u>1 1 1 0 0 0 0 1</u>	
0 0 0 1 1 1 1 1	Y	<u>0 0 0 1 1 1 1 1</u>	Q_5	1 1 0 0 0 0 0 0	
	P_6	0 0 0 0 0 0 0 0	→	<u>0 0 0 0 0 0 0 0</u>	
			Q_6	1 1 0 0 0 0 0 0	

Since Q_6 is not zero, there are errors in the received bytes.

5.3.3 Cyclic Redundancy Check (CRC) *Cyclic redundancy check codes are very powerful and are now almost universally employed. These codes provide a better measure of protection at the lower level of redundancy and can be fairly easily implemented using shift registers or software.*

A CRC code word of length l with n -bit data word is referred to as (l, n) cyclic code and contains $(l - n)$ check bits. These check bits are generated by modulo-2 division. The dividend is the data word followed by $r = l - n$ zeroes and the divisor is a special binary word of length $r + 1$. The CRC code word is formed by modulo-2 addition of the dividend and the remainder so obtained. In CRC code any cyclic shift of a code word results in another valid code word.

EXAMPLE 5.8 Generate CRC code for the data word 110101010 using the divisor 10101.

Solution

Data word	1 1 0 1 0 1 0 1 0	
Divisor	1 0 1 0 1	
	$ \begin{array}{r} 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1 \\ \hline 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0 \\ \underline{1\ 0\ 1\ 0\ 1} \\ 1\ 1\ 1\ 1\ 1 \\ \underline{1\ 0\ 1\ 0\ 1} \\ 1\ 0\ 1\ 0\ 0 \\ \underline{1\ 0\ 1\ 0\ 1} \\ 1\ 1\ 0\ 0\ 0 \\ \underline{1\ 0\ 1\ 0\ 1} \\ 1\ 1\ 0\ 1\ 0 \\ \underline{1\ 0\ 1\ 0\ 1} \\ 1\ 1\ 1\ 1\ 0 \\ \underline{1\ 0\ 1\ 0\ 1} \\ 1\ 0\ 1\ 1 \end{array} $	Quotient Dividend Remainder
Code word	$ \begin{array}{r} 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0 \\ \hline 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1 \end{array} $	

The code word so generated is completely divisible by the divisor because it is the difference of the dividend and the remainder (Modulo-2 addition and subtraction are equivalent). Thus, when the code word is again divided by the same divisor at the receiving end, a non-zero remainder after so dividing will indicate errors in transmission of the code word.

In the above example, note that the CRC code word consists of the data word followed by the remainder. The appended remainder bits are referred to as CRC check bits.

EXAMPLE 5.9 The code word of Example 5.8 is received as 1100100101011. Check if there are errors in the code word.

Solution Dividing the code word by 10101, we get

$$\begin{array}{r}
 \\
 \\
 \\
 \hline
 \\
 \\
 \hline
 \\
 \\
 \hline
 \\
 \\
 \hline
 \\
 \\
 \hline
 \\
 \\
 \hline
 \text{ Remainder}
 \end{array}$$

Non-zero remainder indicates that there are errors in the received code word.

Algebraic representation of binary code words. For the purpose of analysis, the binary codes are represented using algebraic polynomials. In a polynomial of variable x , coefficients of the powers of x are the bits of the code, the most significant bit being the coefficient of the highest power of x . The data word of Example 5.8 can be represented by a polynomial $M(x)$ as: $M(x) = 1x^8 + 1x^7 + 0x^6 + 1x^5 + 0x^4 + 1x^3 + 0x^2 + 1x^1 + 0x^0$

or

$$M(x) = x^8 + x^7 + x^5 + x^3 + x$$

Note that nine-bit data word is represented by an algebraic polynomial of eight degree. Degree of the polynomial is always less than word length by one because polynomial ends with x^0 .

The polynomial corresponding to the divisor is called the *generating polynomial* $G(x)$. $G(x)$ corresponding to divisor used in the last example would be $G(x) = 1x^4 + 0x^3 + 1x^2 + 0x^1 + 1x^0$

or

$$G(x) = x^4 + x^2 + 1$$

The polynomial $D(x)$ corresponding to the dividend (110101010000) is $D(x) = x^{12} + x^{11} + x^9 + x^7 + x^5 = x^4M(x)$. Note that multiplying $M(x)$ by x^4 is equivalent to appending four zeros to the data word.

EXAMPLE 5.10 Write the data words corresponding to the following polynomials: (a) $x^9 + x^5 + x + 1$

(b) $x^3(x^9 + x^5 + x + 1)$ **Solution** (a) $x^9 + x^5 + x + 1$
 $= 1x^9 + 0x^8 + 0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 0x^2 + 1x^1 + 1x^0$

Thus the data word is 1000100011.

(b) Since the polynomial at (a) is multiplied by x^3 , the data word can be obtained by appending three zeroes at the end of data word at (a). Thus the data word is 1000100011000.

Algebraic analysis of CRC code. Let us now re-examine the CRC error detection using algebraic polynomials. We will use the following nomenclature for discussion on CRC (Table 5.2).

TABLE 5.2 Nomenclature for CRC			
	Polynomial	Degree	Number of bits
Data word (M)	$M(x)$	$n - 1$	n
CRC check bits			r
Generating polynomial	$G(x)$	r	$r + 1$
Remainder	$R(x)$	$r - 1$	r
Quotient	$Q(x)$		
Dividend = $x^r M(x)$	$D(x)$	$n + r - 1$	$n + r$
Transmitted code word (T)	$T(x)$	$n + r - 1$	$n + r$
Received code word (T)	$T(x)$	$n + r - 1$	$n + r$
Error word (E)	$E(x)$	$n + r - 1$	$n + r$

If $Q(x)$ is the quotient and $R(x)$ is remainder when $D(x)$ is divided by $G(x)$, $D(x) = Q(x)G(x) + R(x)$ or

$$D(x) + R(x) = Q(x)G(x) + R(x) + R(x) \text{ or}$$

$$D(x) + R(x) = Q(x)G(x) \text{ or}$$

$T(x) = D(x) + R(x) = Q(x)G(x)$ Note that $R(x) + R(x)$ is zero because we are doing modulo-2 addition. Thus, the CRC code $D(x) + R(x)$ is completely divisible by

$G(x)$. As $D(x) = x^r M(x)$, the last power of x is x^r in $D(x)$. $R(x)$ has the highest power of x as x^{r-1} . Therefore $T(x) = D(x) + R(x)$ consists of terms of $D(x)$ followed by the terms of $R(x)$ and is completely divisible by the generating polynomial $G(x)$. Some of the common generating polynomials used in the industry and their applications are: *ITU-T V.41*. It is used in

HDLC/SDLC/ADCCP protocols.

$$x^{16} + x^{12} + x^5 + 1$$

CRC-12. It is employed in BISYNC protocol with 6-bit characters.

$$x^{12} + x^{11} + x^3 + x^2 + x + 1$$

CRC-16. It is used in BISYNC protocol with 8-bit characters.

$$x^{16} + x^{15} + x^2 + 1$$

CRC-32. It is used with 8-bit characters when very high probability of error detection is required.

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Undetected errors in CRC. Not all the types of errors can be detected by CRC code. The probability of error detection and the types of errors that can be detected depend on the choice and degree of the generating polynomial. We will examine in this section the considerations which determine choice of generating polynomials and their error detection capabilities.

The transmitted code word T consists of n -bit data word followed by r -bit CRC code. During transmission errors occur and some of the bits get inverted. The received word T can be written as $T = T + E$ where E is the error word of same length as T . The addition is modulo-2 as before. E has 1s at those bit positions where error has occurred.

Using the algebraic polynomials in place of binary, we get $T(x) = T(x) + E(x)$ At the receiving end, the received polynomial is divided by the generating polynomial $G(x)$ and the remainder is examined.

$$\frac{T'(x)}{G(x)} = \frac{T(x)}{G(x)} + \frac{E(x)}{G(x)}$$

$$\begin{aligned} \text{Remainder } \left[\frac{T'(x)}{G(x)} \right] &= \text{Remainder } \left[\frac{T(x)}{G(x)} + \frac{E(x)}{G(x)} \right] \\ &= \text{Remainder } \left[\frac{T(x)}{G(x)} \right] + \text{Remainder } \left[\frac{E(x)}{G(x)} \right] = \text{Remainder } \left[\frac{E(x)}{G(x)} \right] \end{aligned}$$

since the transmitted polynomial $T(x)$ is completely divisible by $G(x)$. Transmission errors remain undetected when remainder of division of $E(x)$ by $G(x)$ is zero.

EXAMPLE 5.11 If the transmitted code word is 10001001 and the received code word is 11000101, what is the error word? Express these as polynomials.

Received word (T')	1	1	0	0	0	1	0	1
Solution Transmitted word (T)	1	0	0	0	1	0	0	1
Error word (E)	0	1	0	0	1	1	0	0

The corresponding polynomials are: $T(x) = x^7 + x^6 + x^2 + 1$

$$T(x) = x^7 + x^3 + 1$$

$$E(x) = x^6 + x^3 + x^2 + 1$$

Let us examine the situations when the errors remain undetected.

Single bit errors. For single bit errors, $E(x) = x^i$, when the error occurs at $i + 1$ bit position. If we choose the generating polynomial $G(x)$ having at least two terms, it will never divide single term $E(x)$. Thus all single bit errors are detected when $G(x)$ has at least two terms.

Double bit errors. For double bit errors, $E(x) = x^j + x^i$, when the errors occur at $j + 1$ and

$i + 1$ bit positions. $E(x)$ can be written as $x^i(x^{j-i} + 1)$. If $G(x)$ contains at least three terms, it can be shown that except for very large impractical value of $(j - i)$, factors $x^i(x^{j-i} + 1)$ cannot contain all the factors of $G(x)$, or $G(x)$ itself. Therefore, all double bit errors are detected if $G(x)$ contains three terms.

Odd number of errors. For odd number of bit errors, $E(x)$ will contain any odd number of terms, e.g. $E(x) = x^7 + x^6 + x^2$. It can be easily shown that any such polynomial (with odd number of terms) cannot be divided by $x + 1$. If $G(x)$ is so chosen that $x + 1$ is one of its factors, $G(x)$ will not divide $E(x)$ completely and therefore all the odd number of bit errors will get detected.

Burst error of length r . Let us assume that burst error of length r affects the check bit field (Figure 5.7a). The error polynomial for this error will be of degree $r - 1$. Burst error of smaller length will generate $E(x)$ of degree less than $r - 1$.

$$E(x) = x^{r-1} + x^{r-2} \diamond\diamond\diamond x + 1$$

Since $G(x)$ has degree r , it will never divide $E(x)$ completely. Therefore, all the burst errors of length up to r in the check bits will be detected.

If the burst error occurs any where else in the code word, $E(x)$ in the above equation gets multiplied simply by x^i , where i determines how far the burst is shifted from the right hand side of the code word (Figure 5.7b).

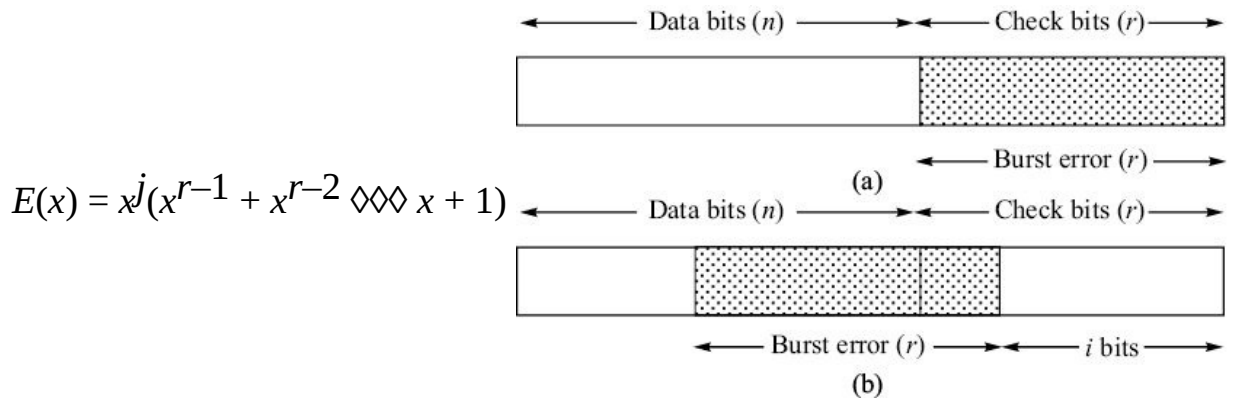


FIGURE 5.7 r -bit long burst error.

If $G(x)$ does not have any power of x (e.g. x^j) as one of its factors, it will never divide the above expression completely since $G(x)$ is of degree r . While choosing $G(x)$, its lowest term is always 1 so that even x is not one of its factors. Therefore, all burst errors of length up to r , irrespective of their location in the data word, will be detected.

Burst error of length $r + 1$. Let us assume that burst error of length $r + 1$ affects right hand side of the code word (Figure 5.8). The error polynomial for this error will be of degree r . $G(x)$ will completely divide the $E(x)$ only when $E(x) = G(x)$, i.e. $G(x)$ and $E(x)$ match term by term.

The most significant term of $E(x)$ corresponding to $(r + 1)$ th bit position is always x^r and the least significant term is always 1 because these two terms delineate the burst error. There are $r - 1$ terms in between in $E(x)$. These $r - 1$ terms have 2^{r-1} combinations and out of these only one combination will match the corresponding terms of $G(x)$. Thus only one out of 2^{r-1} possible $E(x)$

polynomials will be undetected. Probability of not detecting $(r + 1)$ -bit long burst error is, therefore, $1/2^{r-1}$. A little thought will reveal that this result is true for all locations of the $(r + 1)$ -bit long burst errors in the code word.

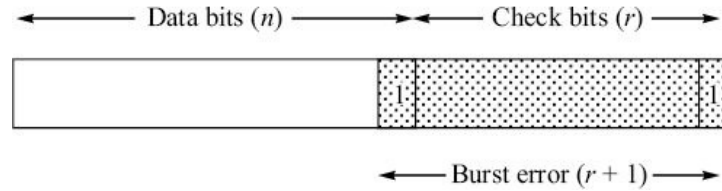


FIGURE 5.8 $(r + 1)$ -bit burst error.

Burst error of length $> r + 1$. In this case degree of $E(x)$ can be $n + r - 1$ when the burst error covers the entire code word. Thus $E(x)$ can be written as below:

$$E(x) = \sum_0^{n+r-1} a_i x^i$$

There are 2^{n+r} possible error polynomials for various combinations of (a_{n+r-1}, \dots, a_0) . All 0s combination can be ignored since it implies that there is no error, we have $2^{n+r}-1$ error polynomials.

For the errors to remain undetected, $G(x)$ should be one of the factors of $E(x)$. Since $G(x)$ has degree r , we can therefore write $E(x)$ as below.

$$E(x) = G(x) \sum_0^{n-1} a_i x^i$$

The number of such error polynomials for various combinations of (a_{n-1}, \dots, a_0) is $2^n - 1$. We have again ignored all 0s combination. Therefore out of $2^{n+r} - 1$ error polynomials of burst errors ($> r + 1$), only $2^n - 1$ error polynomials will be divisible by $G(x)$. In other words, Probability of undetected burst errors of length $> r + 1 = (2^n - 1)/(2^{n+r} - 1)$ If we ignore -1 in the numerator and the denominator, we get Probability of undetected burst errors of length $> r + 1 = 1/2^r$.

To summarize, if the number of check bits in CRC mode is r , probabilities of error detection for various types of errors for suitably chosen generating polynomial are as follows:

- Single errors 100%

- Two bits errors 100%
- Odd number of bits in error 100%
- Burst error of length $< r + 1$ 100%
- Burst error of length $= r + 1$ $1 - (1/2)^{r-1}$
- Burst error of length $> r + 1$ $1 - (1/2)^r$

EXAMPLE 5.12 Show that a polynomial with odd number of terms cannot have $x + 1$ as one of its factors.

Solution If a polynomial $F(x)$ with odd number of terms has $(x + 1)$ as one of the factors, we can write it as $F(x) = (x + 1)P(x)$ Therefore,

$$F(1) = (1 + 1)P(1) = 0$$

$1+1$ in modulo-2 addition is zero. But $F(1)$ cannot be zero since when 1 is substituted as value of x in $F(x)$, each of its terms will become 1. Modulo-2 summation of odd numbers of 1s is always 1. Therefore $(x + 1)$ cannot be a factor of $F(x)$ with odd number of terms.

5.4 FORWARD ERROR CORRECTION METHODS

There are four important forward error correction codes that find applications in digital transmission. They are:

- Block parity
- Hamming code
- Interleaved code
- Convolutional code.

5.4.1 Block Parity The concept of parity checking can be extended to detect and correct single errors. The data block is arranged in a rectangular matrix form as shown in Figure 5.9 and two sets of parity bits are generated, namely,

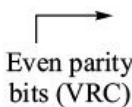
- Longitudinal Redundancy Check (LRC)
- Vertical Redundancy Check (VRC).

VRC is the parity bit associated with the character code and LRC is generated over the rows of bits. LRC is appended to the end of a data block. The eighth bit of the LRC represents the VRC of the other 7 bits of the LRC. In Figure 5.9, even parity is used for the LRC and the VRC.

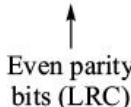
Single bit errors in the block result in failure of longitudinal redundancy check in one of the rows and vertical redundancy check in one of the columns. The bit which is at intersection of the row and the column is in error.

Block code may detect and correct multiple errors depending on the distribution of errors in the block. For example, if two errors occur in the same row, then there will be VRC errors in the

		C	O	M	P	U	T	E	R	
	1	1	1	1	0	1	0	1	0	1
7-bit ASCII codes	2	1	1	0	0	0	0	0	1	1
	3	0	1	1	0	1	1	1	0	1
	4	0	1	1	0	0	0	0	0	0
	5	0	0	0	1	1	1	0	1	0
	6	0	0	0	0	0	0	0	0	0
	7	1	1	1	1	1	1	1	1	0
			1	1	0	0	0	1	1	1



Even parity
bits (VRC)



Even parity
bits (LRC)

Bit transmission sequence:

11000011 11110011 10110010 00001010 10101010 00101011 10100011 01001011 11100001

FIGURE 5.9 Vertical and longitudinal parity check bits.

corresponding columns but there will not be any LRC errors. These errors cannot be corrected.

Block parity is easy to compute and it can correct single bit errors, but to compute LRC bits, we must accumulate at least one column of bits in the memory. Therefore, there is coding delay.

EXAMPLE 5.13 The following bit stream is encoded using VRC, LRC and even parity. Correct the error, if any.

11000011 11110011 10110010 00001010 10111010 00101011 10100011
01001011 11100001

	1	1	1	0	1	0	1	0	1	
	1	1	0	0	0	0	0	1	1	
	0	1	1	0	1	1	1	0	1	
	0	1	1	0	①	0	0	0	0	→ Wrong parity
Solution	0	0	0	1	1	1	0	1	0	
	0	0	0	0	0	0	0	0	0	
	1	1	1	1	1	1	1	1	0	
	1	1	0	0	0	1	1	1	1	
					↑					
										Wrong parity

Fourth bit of the fifth byte is in error. It should be 0.

5.4.2 Hamming Code As mentioned earlier, Hamming code is a perfect error correcting code. It can correct single bit errors. It consists of code words of length n each having r parity bits, where r is the smallest integer satisfying the condition $2^r \geq n + 1$.

The check bits are generated using even or odd parity for a defined set of data bits.

To understand the logic of Hamming code, let us take data word of four bits (Figure 5.10).

For correction of single bit errors, the code word will require three check bits because

$$2^3 \geq (4 + 3 + 1).$$

For correcting an error, we need to know location of the bit that is in error. This location is pointed out by *error syndrome*. Error syndrome is calculated based on the parity checks. It has a value equal to the location of the bit in error. For example, if sixth bit is in error, error syndrome will take binary value 110. For a seven-bit Hamming code, syndrome can take values from 001 to 111 as indicated in the figure.

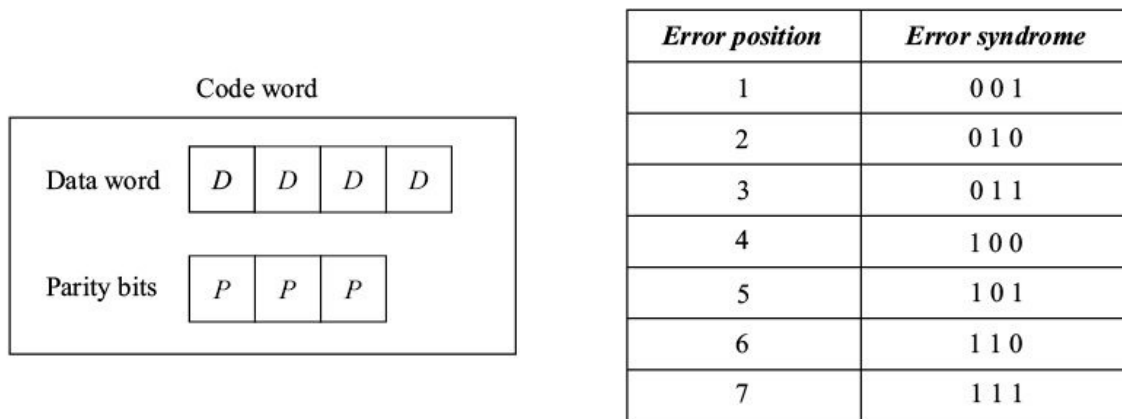


FIGURE 5.10 Hamming code for four bit data word.

LSB of the syndrome is 1 for the bit positions 1, 3, 5, and 7. These bit positions are grouped together. The error in these positions is detected by keeping a parity bit at any one of these four positions. Whenever any of these bit positions is in error, the LSB of the syndrome is made 1.

The middle bit of syndrome is 1 for the bit positions 2, 3, 6, and 7. On the same logic as above, we can place the second parity bit at any one of these four bit positions. If an error occurs in these positions, the second parity bit will detect it and when error is detected, the middle bit of the syndrome is made 1.

The third parity bit is kept at any one of the bit positions 4, 5, 6, or 7 to detect the error in these positions and if there is an error, the MSB of the syndrome is made 1.

Note that the three parity bits together check the entire seven bit code word. We have so designed the groups of bits that whenever any bit is error, three parity checks will generate error syndrome that will point to location of the bit. Let us take an example. If there is error in sixth bit position, first parity check carried out on bits (1, 3, 5, 7) will not detect any error. LSB of the syndrome will be zero. The second parity check on bits (2, 3, 6, 7) will detect an error. Therefore, middle bit of the syndrome will be 1. The third parity check on bits (4, 5, 6, 7) will again detect an error. So MSB of the syndrome will be 1. Thus the syndrome will be 110 indicating that sixth bit is in error.

The parity bits are usually placed at bit positions 1, 2, and 4. The first parity bit is placed at the first bit position, second parity bit at the second bit position, and the third at fourth bit position. These parity bit placements are within their respective groups as indicated earlier. These parity bit positions have weights (1, 2, 4) that correspond to the weights given to their checks in the syndrome.

Note that each data bit is checked by a number of parity bits. Data bit position

expressed as sum of the powers of 2 determines parity bits that check the data bit (Table 5.3). For example, a data bit in position 6 is checked by parity bits P_2 and P_4 ($6 = 2^1 + 2^2$).

TABLE 5.3 Data Bit Positions Checked by the Parity Bits

Data bit positions	Parity bits		
	First (P_1)	Second (P_2)	Third (P_4)
3 = 1 + 2			
5 = 1 + 4			
6 = 2 + 4			
7 = 1 + 2 + 4			

In general, bit positions 1, 2, 4, 8 ..., etc. of the code word are reserved for the parity bits. The other bit positions are for the data bits (Figure 5.11). The MSB of the data word is on the left-hand side and its position is third in Figure 5.11. As usual, the LSB is transmitted first.

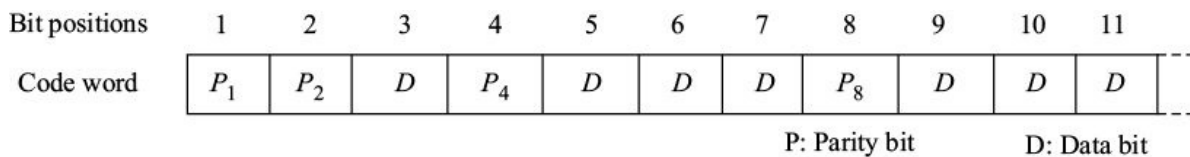


Figure 5.11 Location of parity bits in Hamming code.

Example 5.14 Generate the code word for ASCII character K = 1001011. Assume even parity for the Hamming code.

Solution Four parity bits are required for seven bit data word since $r = 4$ is the minimum number that satisfies the inequality ($2^r \geq 7 + r + 1$). The parity bits are placed at bit positions 1, 2, 4, and 8 (Table 5.4).

TABLE 5.4 Hamming Code of ASCII Character K

	Bit position										
	1	2	3	4	5	6	7	8	9	10	11
	P_1	P_2	1	P_4	0	0	1	P_8	0	1	1
First parity bit	P_1	1	0	1	0	1	$P_1 = 1$				
Second parity bit	P_2	1		0	1		1	1	$P_2 = 0$		
Third parity bit			P_4	0	0	1		$P_4 = 1$			
Fourth parity bit							P_8	0	1	1	$P_8 = 0$
Code word	1	0	1	1	0	0	1	0	0	1	1

Example 5.15 Detect and correct the single error in the received Hamming code word 10110010111. Assume even parity.

Solution

Bit position	Even parity check	Error syndrome
--------------	-------------------	----------------

1	2	3	4	5	6	7	8	9	10	11
P_1	P_2	D	P_4	D	D	D	P_8	D	D	D
Code word	1	0	1	1	0	0	1	0	1	1
First check	1	1	0	1	1	1	1	Fail	---	1
$(P_1, 3, 5, 7, 9, 11)$										
Second check	0	1	0	1	1	1	1	Pass	--	01
$(P_2, 3, 6, 7, 10, 11)$										
Third check	1	0	0	1	1	1	1	Pass	-	001
$(P_4, 5, 6, 7)$										
Fourth check	0	1	1	1	1	1	1	Fail	1001	
$(P_8, 9, 10, 11)$										

Thus the ninth bit position is in error. Correct code word is 10110010011. The data word is 1001011.

5.4.3 Interleaved Codes

Block parity and Hamming codes are not suited for burst errors because multiple errors cannot be detected by these codes. Interleaving is carried out to spread burst errors into single bit errors which can be detected and corrected by these codes. This technique involves writing m consecutive n -bit code words in $m \times n$ matrix and then transmitting the bits column-wise instead of row-wise. Parameter m is called *depth of interleaved code*.

At the receiving end the code word matrix is formed again and error detection is carried on each code word as usual. An error burst of up to m bits will appear as single bit errors in the received code words. Interleaved codes require buffering the input, and therefore add to delay.

EXAMPLE 5.16 Construct interleaved code with depth 3 for HELLO! using even parity.

Solution

Bit positions*	8	7	6	5	4	3	2	1
H	0	1	0	0	1	0	0	0
E	1	1	0	0	0	1	0	1
L	1	1	0	0	1	1	0	0
L	1	1	0	0	1	1	0	0
O	1	1	0	0	1	1	1	1
!	0	0	1	0	0	0	0	1

* Even parity at eight bit position

Since the depth is three, three rows are transmitted at a time as interleaved code. The transmission sequence starting from LSB will be as under: 010 000 011 101 000 111 011 011 010 110 110 000 001 110 110

5.4.4 Convolutional Codes

Unlike block codes in which the check bits are computed for a block of data, convolutional codes are generated over a 'span' of data bits, *e.g.* a convolutional code of constraint length 3 is generated bit by bit using the 'last 3 data bits'. Each data bit is convolved with neighbouring bits so that if it gets corrupted during transmission, enough information is carried by the neighbouring bits to determine the transmitted bit.

Figure 5.12 shows a simple convolutional encoder consisting of a shift register and EXOR gates which generate two output bits for each input bit. It is called a *rate 1/2 convolutional encoder*. The output bits z_1 and z_2 can be written as $z_1 =$

$$x_n \approx x_{n-2}$$

$$z_2 = x_n \approx x_{n-1} \approx x_{n-2}$$

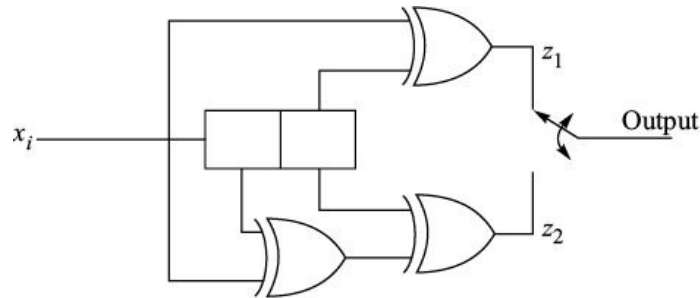


FIGURE 5.12 Half rate convolutional encoder.

The outputs z_1 and z_2 can be written as product of inputs and *generating vectors* [111] and [101].

$$[z_1 \ z_2] = [x_n \ x_{n-1} \ x_{n-2}] \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

State transition diagram of this encoder is shown in Figure 5.13. Each circle in the diagram represents a state of the encoder, which is the content of two leftmost stages of the shift register. There are four possible states 00, 01, 10, 11. The arrows represent the state transitions for the input bit that can be 0 or 1. The label on each arrow shows the input data bit by which the transition is caused and the corresponding output bits. As an example, suppose the initial state of the encoder is 00 and the input data sequence is 1011. The corresponding output sequence of the encoder will be 11010010.

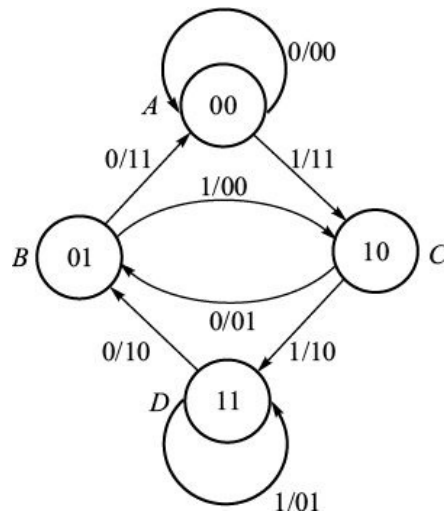


Figure 5.13 State transition diagram.

Trellis diagram. An alternative way of representing the states is by using the trellis diagram (Figure 5.14). Here the four states 00, 01, 11, 10 are represented

as four levels. The arrows represent state transitions as in the state transition diagram. The labels on the arrows indicate the output. By convention, 0 input is always represented as an upward transition and 1 input as a downward transition. The trellis diagram can be derived from the state transition diagram.

EXAMPLE 5.17 Generate the convolutional code using the trellis diagram of Figure 5.14 for the input bit sequence 0101 assuming the encoder is in state A to start with.

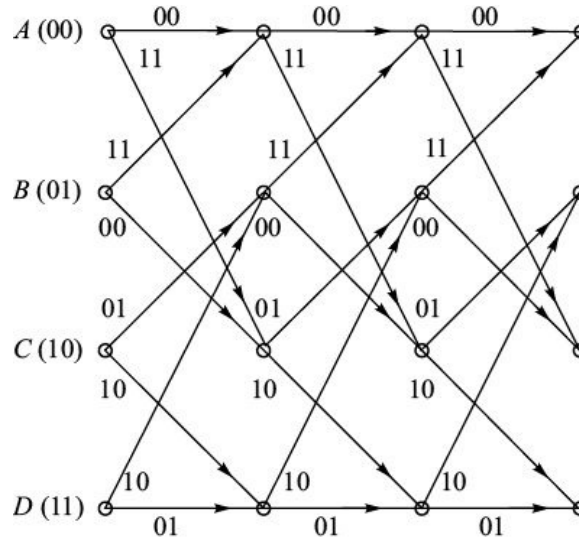


Figure 5.14 Trellis diagram of convolutional encoder shown in Figure 5.12.

Solution Starting from state A at top left corner in Figure 5.14 and tracing the path through the trellis for the input sequence 0101, we get the output bits as shown in the following table:

Present state	Input bit	Next state	Output bits
A	0	A	00
A	1	C	11
C	0	B	01
B	1	C	00

Output bit sequence: 0 0 1 1 0 1 0 0

Decoding algorithm. Decoder for the convolutional code is based on the maximum likelihood principle. Knowing the encoder behaviour and the received sequence of bits, we can find the most likely transmitted sequence by analyzing all the possible paths through the trellis. The path which results in the output sequence which is nearest to the received sequence is chosen and the corresponding input bits are the decoded data bits. The decoding algorithm is

known as Viterbi algorithm.

Let the data bit sequence be 1011 which is encoded as 11010010 using the encoder shown in Figure 5.12. The received sequence is 11110010 having an error in the third bit position.

We need to analyse all possible paths through the trellis and select the path which results in an output sequence that is nearest to the received sequence. Figure 5.15 shows all such paths. We will carry out the analysis of these paths in two steps. After the first step we will be in a position to drop further analysis of some of the paths.

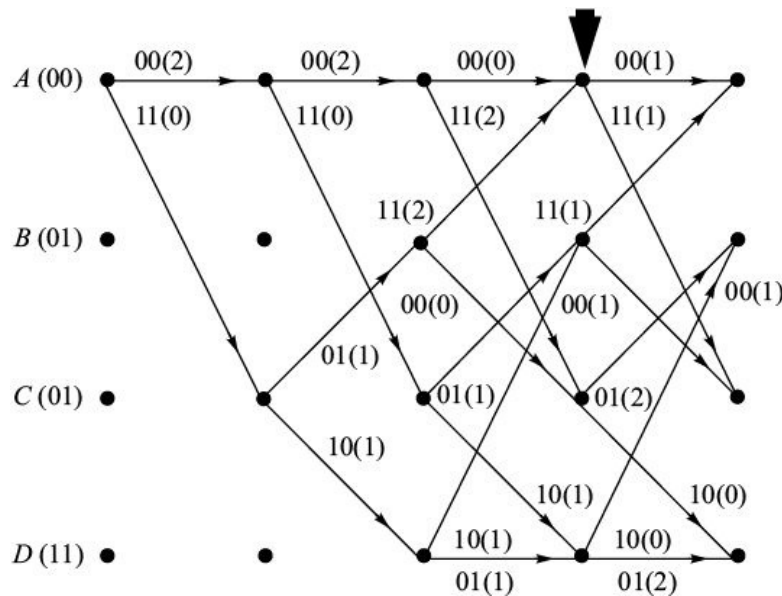


Figure 5.15 Trellis diagram for decoding 11110010.

Step 1. At any point of analysis, when segments of two paths converge on the same state, we chose one of the paths which is nearer to the received sequence up to that point, for further analysis. Note that a pair of paths converge on each state, *e.g.* state A can be reached via AAAA or ACBA. But path AAAA results in output sequence 000000 which is at a distance of 4 from the first six bits of the received sequence. In the case of the other path ACBA, this distance is only 3. Because we are looking for a sequence with the smallest distance, we need not consider the first path for further analysis. We can drop some more paths in similar manner. The selected paths for further analysis are ACBA, ACDB, ACBC, and ACDD.

Step 2. Having considered the first three pairs of bits, let us move further. Transitions from the last state arrived at in the first step, will result in two potential states depending on the next input bit. Distances of the resulting bit

sequences from the received sequence are given in Table 5.5. Note that we have computed the distances for only the selected paths of the first step. The minimum distance is for the path *ACBCD* which corresponds to the correct data bit sequence 1011.

TABLE 5.5 Alternative Paths through the Trellis

Input data bits	Path	First step		Next data bit	Next state	Next step	
		Output sequence	Distance from 111100			Output sequence	Distance from 11110010
000	AAAA	000000	4				
100	ACBA	110111	3	0	A	11011100	4
				1	C	11011111	4
110	ACDB	111010	2	0	A	11101011	3
				1	C	11101000	3
010	AACB	001101	3				
001	AAAC	000011	6				
101	ACBC	110100	1	0	B	11010001	3
				1	D	11010010	1
111	ACDD	111001	2	0	B	11100110	2
				1	D	11100101	4
011	AACD	001110	3				

Example 5.18 What is the message sequence if the received rate 1/2 encoded bit sequence is 00010100? Use the trellis diagram given in Figure 5.14.

Solution Drawing the path through trellis (Figure 5.14), we select the paths *AAAA*, *AACB*, *AAAC*, and *AACD* in the first step as indicated in the following table. The next step leads to the path *AACBC* that gives output sequence nearest to the received code word. Therefore the corrected received sequence is *00110100* and the message is *0101*.

data bits	Path	First step			Next data bit	Next state	Next step	
		Output sequence	Distance from 000101	Output sequence			Distance from 00010100	
000	AAAA	000000	2	0	A	00000000	2	
				1	C	00000011	4	
100	ACBA	110111	3	1	C	11011111	4	
110	ACDB	111010	6					
010	AACB	001101	1	0	A	00110111	3	
				1	C	00110100	1	
001	AAAC	000011	2	0	B	00001101	3	
				1	D	00001110	2	
101	ACBC	110100	3					
111	ACDD	111001	4					
011	AACD	001110	3	0	B	00111010	4	
				1	D	00111001	4	

Convolutional codes suffer from one disadvantage. Decoding can take place when the whole block is received. The block size is usually large and therefore the decoding delay is also large.

5.5 REVERSE ERROR CORRECTION

We have seen some of the methods of forward error correction but reverse error correction is more economical than forward error correction in terms of the number of check bits. There are three basic mechanisms of reverse error correction:

- Stop and wait,
- Go-back- N ,
- Selective retransmission.

In data communications, reverse error correction methods are used extensively. We shall describe these mechanisms in detail in Chapter 8. The underlying principles behind these methods are introduced below.

5.5.1 Stop and Wait

In this scheme, the sending end transmits one block of data at a time and then waits for acknowledgement from the receiver. The data block contains check bits for error detection. If the receiver detects any error in the data block, it sends a request for retransmission in the form of negative acknowledgement. If there is no error, the receiver sends a positive acknowledgement, after receiving which the sending end transmits the next block of data. Figure 5.16 illustrates this mechanism.

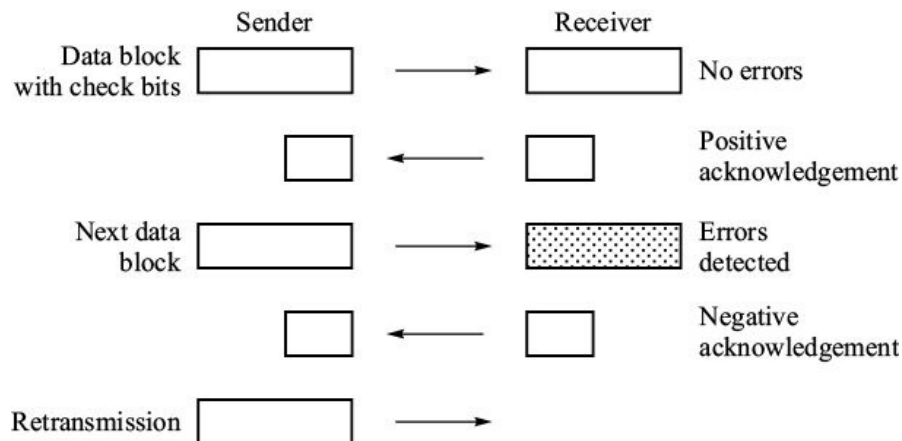


Figure 5.16 Reverse error correction by stop-and-wait mechanism.

5.5.2 Go-Back— N

In this mechanism all the data blocks are numbered and the sending end keeps transmitting the data blocks with check bits. Whenever the receiver detects error in a block, it sends a retransmission request indicating the sequence number of the data block received with errors. The sending end then starts retransmission of all the data blocks from the requested data block onwards (Figure 5.17).

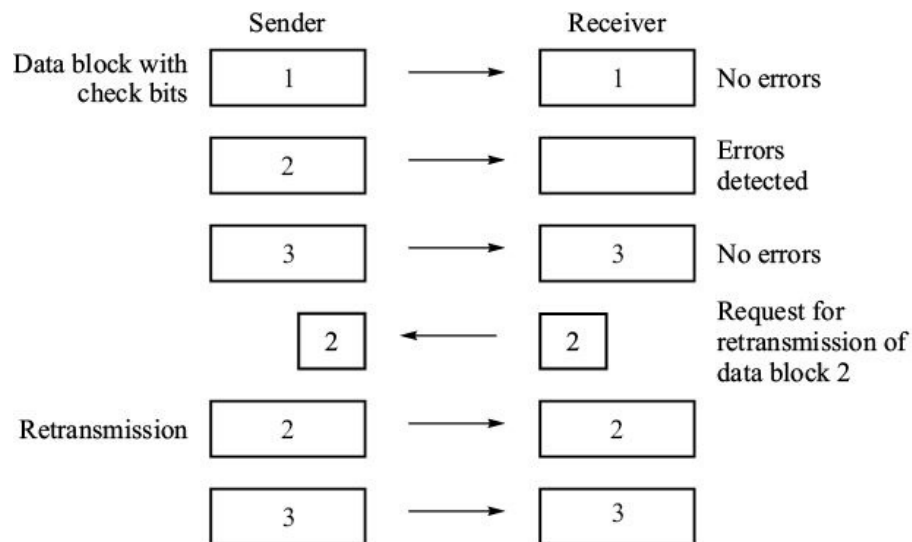


Figure 5.17 Reverse error correction by go-back-N mechanism.

5.5.3 Selective Retransmission

If the receiver is equipped with the capability of putting the received data blocks in sequence, it requests for selective retransmission of the data block containing errors. On receipt of the request, the sending end retransmits the data block but skips the following data blocks already transmitted. It continues with the next data block (Figure 5.18).

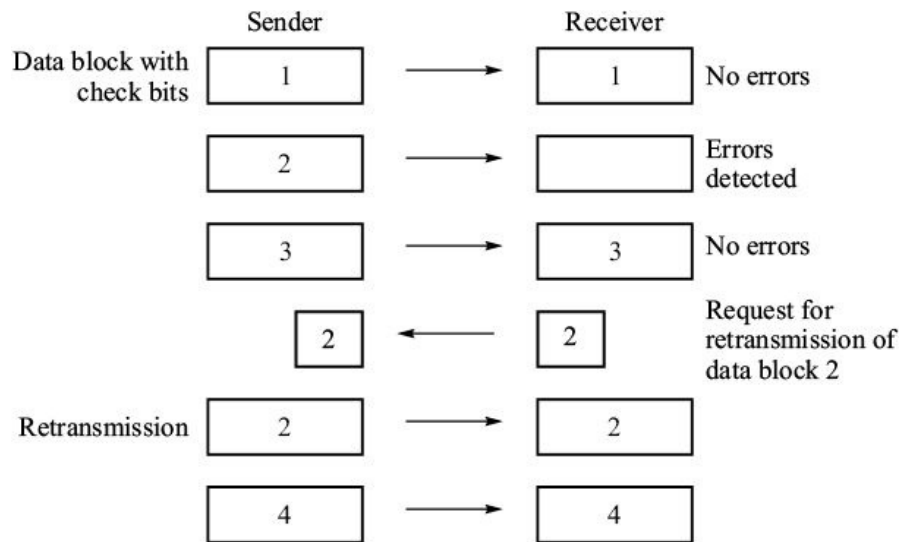


FIGURE 5.18 Reverse error correction by selective retransmission.

SUMMARY

Errors are introduced due to imperfections in the transmission media and in the operation of the network. For error control, we need to detect the errors and then take corrective actions. Errors can be content errors or flow integrity errors. Content errors are detected by adding additional check bits to a block of data. The check bits can be in the form of parity bits, checksum, and Cyclic Redundancy Check (CRC) bits. Out of the three, CRC is the most powerful and widely implemented. It can detect multiple bit errors and burst errors. Flow integrity errors refer to out of sequence, duplicate, missing blocks of data. These are detected by putting a sequence number on each block of data.

Error correction methods include forward error correction or reverse error correction. In forward error correction, the receiver is able to detect and locate the content errors. Correction involves inverting the bit in error. Forward error correction requires additional check bits. Block parity, Hamming code, interleaved code and convolutional code are some of the forward error correction methods. Forward error correction methods are rarely used in data communications.

In reverse error correction methods, the receiver requests retransmission of the blocks of data that are received with errors. Reverse error correction mechanisms can handle both content and flow integrity errors. Reverse error correction mechanisms are stop and wait, go-back-N or selective retransmission.

EXERCISES

1. There are 200 bytes in a data block, each byte being 8 bits. If the error rate is $1 \cdot 10^{-5}$, what is the probability of the block being received in error?
2. (a) Write the bit sequence corresponding to the message 'DECODER'. Assume ASCII code with odd parity bit.
 (b) What is the text of the message corresponding to the following transmitted bit sequence? Assume ASCII code and even parity:
 001000101111001100110011001100111000001001001011
 (c) Detect errors if any in the following message assuming it is in ASCII code with odd parity:
 00001011100000110100111000101010
3. Write the bit transmission stream for the message 'START BIT' which is coded in ASCII with VRC and LRC for error detection. Assume odd parity.
4. The following bit stream is encoded using VRC and LRC. Correct the error if any. What is the transmitted message? Assume ASCII code with even parity in VRC and LRC.
 1100101000010111100001100110011000101110000001011000110110001
5. The following bit stream is encoded using VRC and LRC. Detect the errors present in the message. Can the errors be corrected? Assume even parity.
 1100101000010110100101100110011110010111000000101100011011000
6. (a) Find simple checksum of the following bytes using modulo-256 addition. The MSBs are on the left of each byte.
 10101010 10000001 11011011 01101100 10010101
 (b) The above bit stream is transmitted with 2's complement of the checksum as the last byte. Check whether there are errors in the following received sequence:
 10101010 10010001 11011011 01101110 10010101 11111001
 (c) Repeat (b) for the following received sequence:
 10001010 10000001 11011011 01101100 10110101 11111001
 Why are the transmission errors not detected?
7. (a) Find the two checksum bytes as per the transport protocol for the following data bytes. The MSBs are on the left of each byte.
 10101101 11101110 11111011 00111100 10001001
 (b) Verify that the checksum is corrected by performing the receiver check on the data bytes and the checksum bytes.
8. Generate CRC code for the data word 1010001011 using the divisor 11101.

9. If the CRC code is 10100010111100 and the generating polynomial is $x^4 + x^3 + x^2 + 1$, check if there is any error in the code word.
10. Received Hamming code word is 11110000101. Even parity is used. Locate and correct the bit in error.
11. Generate Hamming code for the following characters using even or odd parity as indicated. The character parity bit is not used in the ASCII code set.
 - (a) "U" [ASCII, parity: Odd]
 - (b) "?" [ASCII, parity: Even]
 - (c) "M" [EBCDIC, parity: Even]
12. What are the characters corresponding to the received Hamming codes given below. The parity used is indicated in parentheses.
 - (a) 00011111110 ASCII (Odd)
 - (b) 11100011010 ASCII (Even)
 - (c) 100010011100 EBCDIC (Odd)
13. Generate convolutional code for message bits 1101 using rate 1/2 encoder shown in Figure 5.12.
14. What are the message bits, if the received rate 1/2 code word is 11100010? Use the trellis diagram given in Figure 5.14.
15. If $G(x) = x^{20} + x^7 + 1$, how many check bits are in the code word?
16. If the transmitted word is 1001111001 and the received word is 1000101101, what is the length of the burst error?
17. If the generating polynomial is $G(x) = x^{20} + x^7 + 1$, what is the probability of detecting burst errors having length more than 21?
18. If size of data block is 63 bits, what is the minimum number of check bits required to correct single bit errors?
19. Which of the following generating polynomials will detect with certainty burst error of length 15?
 - (a) $G(x) = x^{12} + x^{11} + x^3 + x^2 + x + 1$
 - (b) $G(x) = x^{16} + x^{15} + x^2 + 1$
 - (c) $G(x) = x^{10} + x^9 + x^5 + x^4 + x + 1$
20. For the data word $a_3 x^3 + a_2 x^2 + a_1 x + a_0$, find the CRC check bit using $x + 1$ as generating polynomial. How does the CRC check bit compare with

the parity bit?

21. If the code words are generated by multiplying the generating polynomial $x + 1$ with the data words, instead of dividing them, write the code set for all the 3 bit data words. Is it a systematic code?
22. Draw a convolution encoder with code rate $1/2$ and constraint length of 4. The generating vectors are $[1101]$ and $[1111]$. Draw the state transition diagram.
23. Assume that the shift register is initialized to 000 state in Figure E5.19. Write the output sequence for the input sequence of 101110.

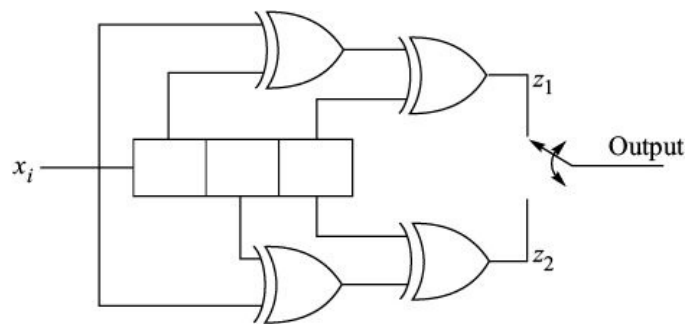


Figure E5.19.

6

Network Architecture

Communication is based on transfer of information having a wider scope. In this chapter, we examine a man-to-man communication analogy to identify the functional requirements for meaningful communication. In the context of these requirements, we develop a model for communication in a computer network. The model is based on the concept of layered architecture. Then we proceed to the concepts of open systems and discuss the reference model for Open System Interconnection (OSI). We also have a look at other layered architectures used in the industry.

Understanding of the concept of layered architecture and of the OSI model is essential prerequisite for systematic grasp over the computer networking. Therefore, it is necessary to cover this chapter before proceeding further with the rest of the book.

6.1 TOPOLOGY OF A COMPUTER NETWORK

A *computer network* consists of end systems which are sources and sinks of information, and which communicate through a *transit system* interconnecting them (Figure 6.1). The transit system is also called an *interconnection subsystem* or simply a *subnetwork*.

An end system consists of computers, terminals, software, and peripherals forming an autonomous whole capable of performing information processing. Each end system has an interconnection point through which it is physically connected to the transmission medium. The interconnection point has an address by which the end system is identified. Each end system hosts one or more application entities. It is due to these application entities that communication

takes place between the end systems. They determine the subject and duration of their communication.

The subnetwork is without any application entity and performs all transmission and switching activities required for transporting messages between the end systems. It consists of nodes that process the messages for routing them towards their destination. Nodes are interconnected to the end systems at the edges of the subnetwork through a transmission medium that carries electrical signals.

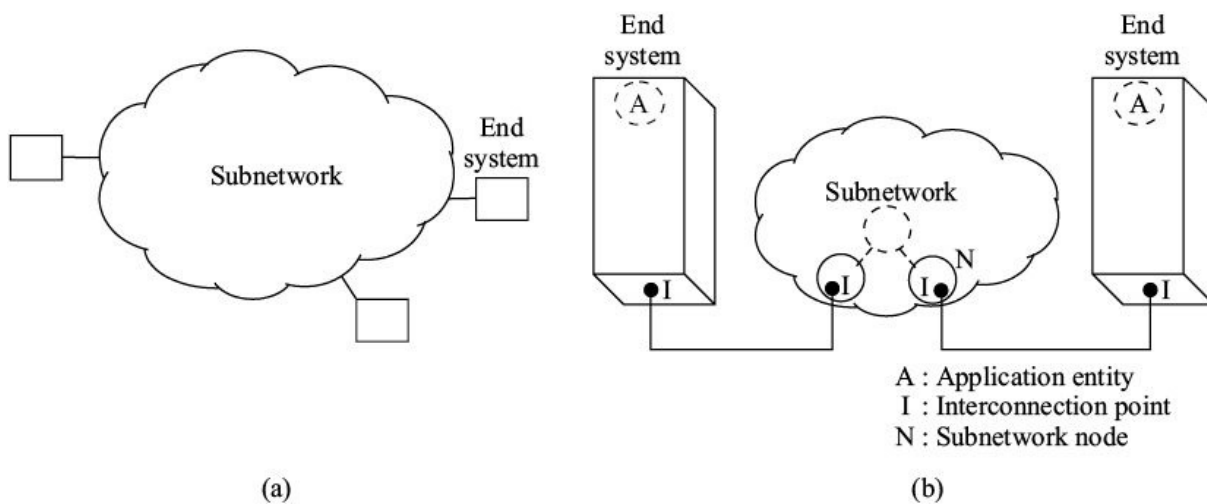


FIGURE 6.1 Computer network.

6.2 ELEMENTS OF MEANINGFUL COMMUNICATION

The purpose of communication between the application entities is not served just by exchanging bits. The communication needs to be meaningful. Meaningful communication is always done with a purpose and aims at enlarging common understanding between communicating entities. Physical transfer of bits is transmission not communication.

There are some basic constituents of the communication process which must be present in any communication to make it meaningful. This is true for any type of communication, between human beings or between computers. We shall take a human communication analogy to understand the elements of the communication process. From this analogy we shall try to derive functional requirements for meaningful communication between two end systems of a

computer network.

Consider that a young Indian, Ravi, would like to tell his friend Mary who lives with her mother in Britain, about her visit to India during vacations. Mary is learning Hindi for her forthcoming visit. Ravi decides to call Mary over the telephone and dials her telephone number. Mary's mother picks up the receiver. The conversation which ensues is as follows:

<i>Ravi</i> : Hello, Ravi speaking.	Authentication
<i>Mother</i> : Hello Ravi, mama here.	Identification of communicating entities.
<i>Ravi</i> : Mama, could I speak to Mary?	
<i>Mother</i> : Please wait. I'll call her.	
. . . (<i>Mother calls Mary</i>)	
<i>Mary</i> : Hello Ravi, Mary here.	
<i>Ravi</i> : Hello, I have made plans for your visit to India.	Common Theme
<i>Mary</i> : Thanks.	Agreement on the common theme.
<i>Ravi</i> : Are your Hindi lessons continuing? May I speak in Hindi?	Common Language
<i>Mary</i> : No. I am still not at ease with Hindi. Please continue in English.	Agreement on the common language.
<i>Ravi</i> : O.K. (<i>Ravi tells her the program.</i>)	
<i>Mary</i> : Yes, fine.	Synchronization (Forward)
	Point of common understanding and indication of willingness to proceed.
(There is some disturbance on the line)	
<i>Mary</i> : Please repeat the dates. I did not hear you clearly.	Error Recovery
<i>Ravi</i> : ... (<i>Ravi repeats</i>) ...	Recovery of the lost messages.
<i>Mary</i> : Yes.	
<i>Mary</i> : Please speak slowly. Let me take it down.	Flow Control
. . . (<i>Ravi slows down</i>)	Control of the flow of messages.
<i>Mary</i> : I could not follow after the visit to Agra.	Synchronization (Backward)
<i>Ravi</i> : . . . (<i>Ravi repeats and continues</i>)	Loss of the point of common understanding. Going back to the last point of common understanding.
<i>Ravi</i> : Good bye, Mary.	
<i>Mary</i> : Bye, Ravi.	

The above communication involved three communicating entities—Ravi, Mary, and her mother. The entire process involved several steps:

- Establishing connection through a telephone network.
- Identification of the communicating entities.
- Understanding on the common theme and common language for communication.
- Disciplined dialogue exchange which required flow control, error control, and synchronization.

Common theme, common language, and an orderly session are the essential elements of any type of meaningful communication, though they may not be explicitly spelled out in every incidence of human communication.

The elements of meaningful communication discussed above are also applicable to a distributed computing system where application entities residing in different end systems communicate. There is need to establish connection between the communicating entities through the subnetwork. Authentication (user password), login, syntax, and orderly exchange of messages with markers for dialogue synchronization are built into an end system. For each incidence of communication, these elements must be decided and agreed upon explicitly by the communicating entities for the communication to be meaningful.

6.3 TRANSPORT-ORIENTED FUNCTIONS

Communication results in generation of messages which are to be truthfully transported between the communicating entities. In the example we have just discussed, this function is carried out by the telephone network. In a computer network, the subnetwork provides means of transporting the messages. But some additional functions, as discussed below, must also be built into the end systems for enabling transport of the messages through the subnetwork without any error.

6.3.1 Interaction with the Subnetwork In the man-to-man

communication situation described above, the communicating entities interacted with the telephone network by way of lifting the handset of the telephone instrument, waiting for the dial tone, dialing the telephone number, and answering the ring. A computer network is somewhat analogous in this respect. The end systems need to interact with the subnetwork for

transporting the messages to the destination. This interaction is in the form of specifying the address of the destination, answering an incoming call, and releasing the connection.

It is also possible that a subnetwork may offer message transport service that is equivalent to the postal service. In this case, an end system merely releases a message in the subnetwork. The message bears source and destination addresses and the subnetwork delivers the message to the destination.

6.3.2 Quality of Transport Service The decision to use the telephone network and not the other available means (e.g. telegram, post) was taken by Ravi on considerations of delivery delay, cost, and reliability. In a computer network, the end system needs to set up an appropriate transport connection of the required quality of service. Quality of Service (QOS) is specified in terms of error rate, transit delay of message delivery, throughput, and, of course, the cost.

6.3.3 Conversion of Signals

The messages generated by the communicating entities in the above example were speech signals which are suitably converted to electrical signals for transmission by the telephone instrument. In digital devices, messages are in the form of bits. The bits need to be converted by the end systems into electrical signals having suitable voltage levels and impedance for the transmission media.

6.3.4 Error Control

Speech signals have so much built-in redundancy that even if there is some corruption of the signal, the communicating entities are usually able to make sense of the received signals. But computer communication is very sensitive to the errors. Errors are introduced due to noise and distortion of the electrical signals during transmission. Some mechanism in the end systems to control these errors is required.

6.4 COMPONENTS OF A COMPUTER NETWORK

From the above analogy of man-to-man communication, we can deduce certain functional capabilities that must be built into an end system for meaningful communication. These capabilities are listed below. This list is not exhaustive but it does indicate broad categorization of the required capabilities.

- Authentication and login
- Common syntax
- Establishment of an orderly exchange of messages with markers for forward and backward synchronization
- Establishing transport connection of required quality, flow control
- Interacting with the subnetwork
- Error control
- Conversion of bits into electrical signals and vice versa.

The above communication functions are implemented using many hardware (physical) and software (logical) components in a computer network. All the components of a computer network function in a coordinated fashion to realize the functional requirements of meaningful communication between the end systems. Design and implementation of such a system is one of the most complex tasks that man has ever tried.

6.5 ARCHITECTURE OF A COMPUTER NETWORK

The architecture of a system, whether it is a building, organization or a computing system, describes how the system has been assembled using various components of the system. It defines the specifications of the components and their interrelationships. The architecture of a computer network, or simply network architecture, specifies a complete set of rules for the connections and interactions of its physical and logical components for providing and utilizing communication services. The network architecture does not encompass the application software which are, in fact, users of the services provided by the network.

6.5.1 Network Architecture Models Designing the architecture of

a complex system requires a model. A model helps in several ways as follows:

- It enables visualizing and understanding the complex structure of the system.
- The model identifies the various components of the systems and defines their interrelationships.
- Unforeseen structural errors can be avoided at the modelling stage so that detailed design effort is not wasted due to these errors.
- Additions and structural changes to accommodate new requirements at a later stage can be coordinated retaining the overall basic structure of the model.

After the model is ready, design specifications of the components and their standardization can be taken up. For developing architecture of a computer network, we follow the same approach. We need an orderly reference model of the computer network.

6.5.2 Partitioning of a System The approach usually adopted for modeling a complex system is to partition it into meaningful functional pieces. After identifying these functional pieces, their interrelationships, interfaces, services, and functionality are so defined that on integration they form a complete model. Partitioning is beneficial in many ways:

- A complex problem is broken down into smaller tasks. Each smaller task can be attended to by a specialist team.
- Partitioning can be done in such a way that some of the tasks which already have an acceptable solution are not attempted again, thus reducing the developmental effort.
- Partitioning also results in a modular structure which permits flexibility of upgrade and reconfiguration.

6.5.3 Features of a Partitioned Structure Partitioning is not a new concept. It is built into every system either intentionally or intuitively. Let us consider example of an organization that has

several geographically distributed offices and each office has several functional levels. For the functionality of the overall organization, procedures for interactions are defined at various levels.

Figure 6.2 shows the sequence of events which take place when a manager communicates with another manager in the organization. In this example, the organizational structure has a vertical partition and several horizontal partitions. Some of the important features of this partitioned structure are:

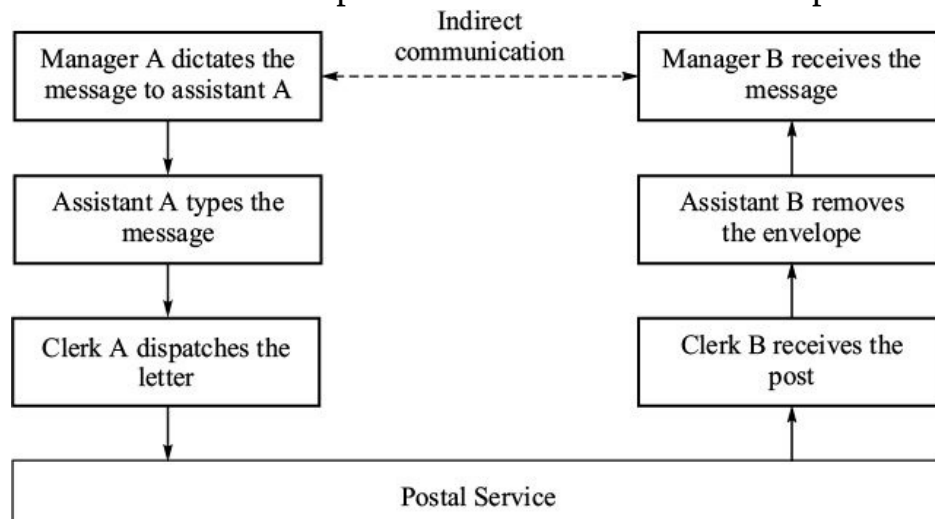


FIGURE 6.2 Partitioned model of communications in an organization.

- Different functions in the organization are separated and each function is distinctly implemented, *e.g.* typing is restricted to the second level only.
- Orderly sequence of functions is ensured in the vertical structure of the organization. Manager interacts with his assistant who in turn interacts with dispatch clerk.
- There is no bypassing of levels. The interaction is between the adjacent levels only, *e.g.* the managers do not directly interact with the clerks.
- To carry out functions assigned to a level, services of the next lower level are needed. For example, manager A needs typing services of the assistant to send the message.
- To provide service to the higher level, the peer at lower layers may coordinate, *e.g.* to trace a lost letter the clerks A and B interact.
- The services are transparent, *i.e.* the lower level does not restrict the higher level in any way. It communicates whatever it receives from the higher level. For example, clerk A dispatches the letter. He is in no way concerned

about its contents. He may sent the letter in two or several envelopes if the pages are many. But he must put some instructions on the envelope for the clerk at the other end to enable recompilation of the letter by him.

- There is indirect interaction between the peer levels. The managers interact with each other through the services provided by their assistants.
- An interconnecting transport medium is needed. Postal department carries out this function in this case.

6.6 LAYERED ARCHITECTURE OF A COMPUTER NETWORK

Decomposition of the organization into offices and each office into hierarchical functional levels and the interaction procedures define the overall organization architecture. A computer network is also partitioned into end systems interconnected using a subnetwork and the communication process is decomposed into hierarchical functional layers (Figure 6.3). Just like in an office, each layer has a distinct identity and a specific set of functions assigned to it. Each layer has an active element, a piece of hardware or software, which carries out the layer functions. It is called *layer entity*.

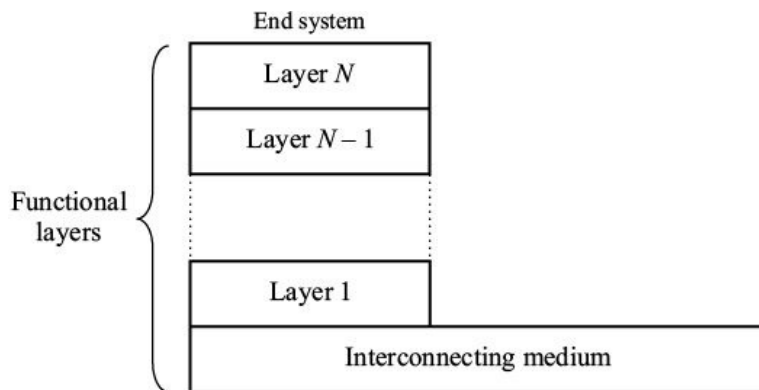


Figure 6.3 Layered architecture of an end system.

The criteria for defining the boundaries of a layer are:

- Each function is distinctly identified and implemented precisely in one layer.
- Functions are carried out in logical sequential manner by proper design of the hierarchy.

- Volume of communication between adjacent layers is minimized by suitably choosing the layer boundaries.
- Boundaries of a layer are defined by considering the existing acceptable implementation.
- The implementation details of a function in a layer are hidden so that any change in the implementation does not affect other layers.

In section 6.4, we examined the basic functional capabilities to be built into an end system for meaningful communication. Each of these functions is assigned to one of the layers of the model. Functions cannot be assigned in any arbitrary order. Sequence in which the functions are carried out must taken into account for proper assignment.

6.6.1 Need for Standardization of Network Architecture The layered architecture concept was built into many computer systems but different vendors defined proprietary protocols and interfaces. The layer partitioning also did not match. As a result, there was total integration incompatibility of architectures developed by different vendors. Standardization of network architecture can solve many problems and save a lot of effort required for developing interfaces for networking different architectures.

There are several network architectures developed by manufacturers and by standardization organizations. Some of the important network architectures are:

- IBM's System Network Architecture (SNA)
- Digital's Digital Network Architecture (DNA)
- Open System Interconnection (OSI) reference model developed by ISO (International Organization for Standardization) and ITU-T.
- Internet architecture.

SNA and DNA are vendor-specific layered architectures. The OSI reference model and Internet are two vendor-independent architectures. Between the two, Internet architecture is more widely deployed. We will go through layer by-layer design of both these architectures in the book.

6.7 OPEN SYSTEM INTERCONNECTION

Open System Interconnection (OSI) represents a generalization of concepts of inter-process communication so that any open system may be technically able to communicate with another open system. Systems achieve openness by following certain architecture and obeying standard protocols. These standards are open for anybody to use and implement unlike proprietary architectures whose implementation details are always either trade secrets or covered by patent rights.

The OSI architecture is the first step towards standardization. It decomposes the communication process into hierarchical functional layers and identifies the standards necessary for open system interconnection. It does not specify the standards but provides a reference model for development of standards. The OSI architecture is, therefore, called *reference model* for open system interconnection.

This model was developed primarily by ISO and was approved as international standard IS 7498 in 1983. ITU-T did parallel work and their Recommendation X.200 is in complete alignment with IS 7498.

6.8 LAYERED ARCHITECTURE OF THE OSI REFERENCE MODEL

In the OSI reference model, the communication functions are divided into a hierarchy of seven layers as shown in Figure 6.4. It is also referred to as the 7-layer model. The transmission medium is not included in the seven layers and, therefore, it can be regarded as the 0th layer.

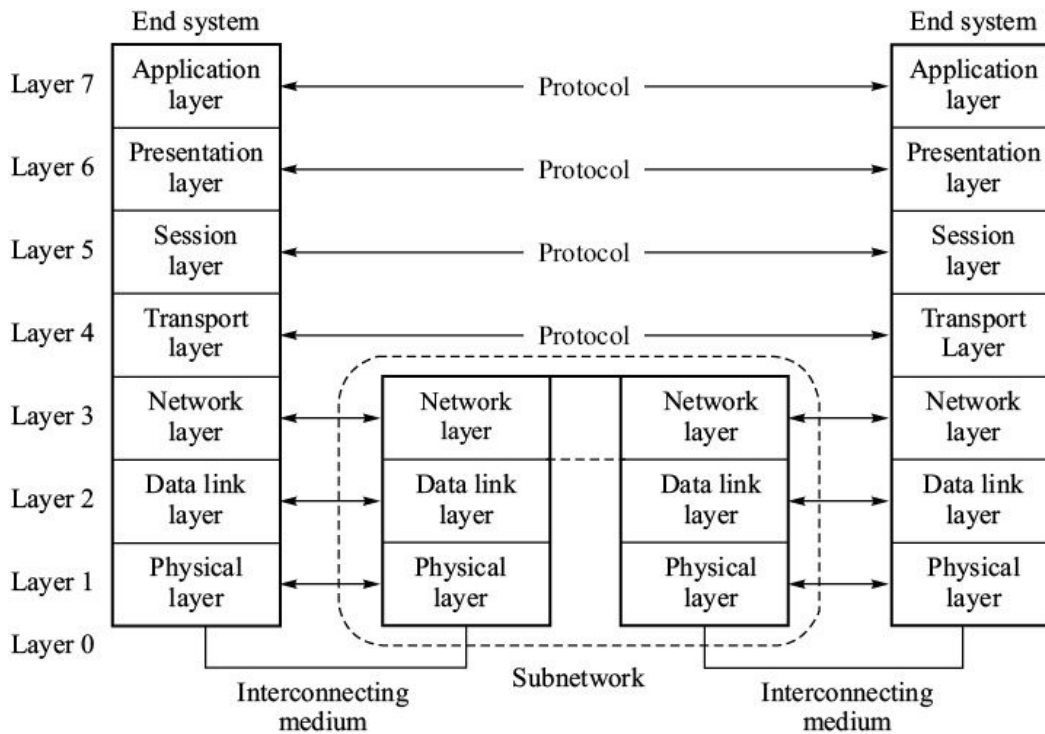


FIGURE 6.4 Layered architecture of the OSI reference model.

The computer network is partitioned into end systems and subnetwork. The end systems are the computer systems that house the applications. The subnetwork has at most three layers as shown in Figure 6.4. These three layers interface with the corresponding peer layers of the end systems to carry out functions relating to the transport of messages from one end system to the other.

Table 6.1 gives a summary of the functions and services provided by the layers of the OSI model. We will take up some of these layers individually and describe them in detail later in the book.

TABLE 6.1 Services Provided by the Various Layers of the OSI Reference Model

Level	Layer	Primary functions	Services provided to next higher layer
7	Application	Support the end user, login, password, file transfer	This is the highest layer and provides user-oriented services.
6	Presentation	Code and format conversion	Freedom from compatibility problems
5	Session	Session management, synchronization	Management of dialogue between two application entities
4	Transport	End-to-end delivery of data units	End-to-end transport connection of the required quality of service
3	Network	Establishing end-to-end network connections, routing	End-to-end transport of data packets.
2	Data link	Error control, flow control	Reliable transfer of bits across the physical

		connection	
1	Physical	Conversion of bits into electrical signals and their transmission	Transmission of bits

6.8.1 Application Layer

As the highest layer in the OSI model, the *application layer* provides services to the user of the OSI environment. Login, password checking, file transfer, *etc.* are some of the functions of the application layer.

6.8.2 Presentation Layer

The purpose of the *presentation layer* is to present the information to the communicating application entities in a way that preserves the meaning while resolving the syntax (code and data format) differences. There are three syntactic versions of data being transferred, the syntax used by the application entity of the originator of the data, the syntax used by the recipient of the data, and the “transfer” syntax used to transfer the data between presentation entities (Figure 6.5).

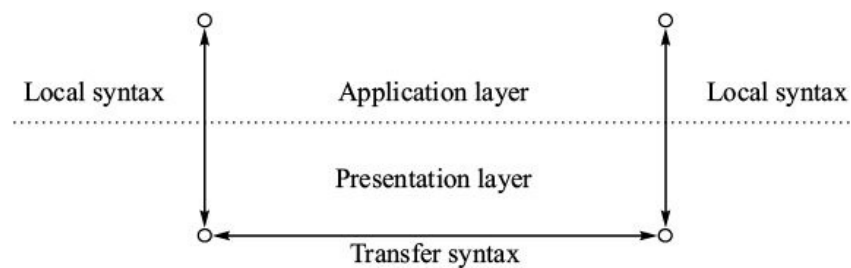


FIGURE 6.5 Local and transfer syntax of the presentation layer.

These syntax may be same or different. When they are not same, the presentation layer contains functions necessary to transform the transfer syntax to the required syntax used by the application entities preserving the meaning. There is no fixed transfer syntax, and is to be negotiated by the presentation entities. Encryption of data, if required, is also carried out by the presentation layer.

6.8.3 Session Layer

The purpose of the *session layer* is to provide the means necessary for the cooperating presentation entities to organize and synchronize their data exchange. It provides functions which are necessary for opening a communication relationship called a *session*, for carrying it out in an orderly fashion and for terminating it. At the time of session termination, the session

entities ensure that there is no data loss unless there is a request from the presentation entities to abort the session.

The session layer provides two-way simultaneous, two-way alternate, and one-way communication services. The session synchronization service enables the presentation entities to mark and acknowledge identifiable ordered synchronization points. During a session, the session entities enable resynchronization if the transport service is disrupted and reestablished.

6.8.4 Transport Layer

The overall function of the *transport layer* is to provide transport service of the quality required by the session entities in a cost-effective manner. The quality of the transport connection is specified in terms of residual error rate, delay, throughput and other quality determining parameters.

For optimum utilization of network resources and for achieving the quality of service, the transport layer may do multiplexing, splitting, blocking, and segmenting. These functions are described in the next section.

It may be seen from Figure 6.4 that the transport layer provides end-to-end connectivity. To meet the quality of service (QOS) requirements, the transport layer may be required to carry out sequencing of the messages and to exercise flow control, if the network service does not meet the QOS requirements.

6.8.5 Network Layer

The *network layer* provides the means to access the subnetwork for routing the messages to the destination end system. The network layer of an end system interacts with the network layer of the subnetwork for this purpose. An access node of a subnetwork facing the end system must support the three lower layers of the OSI model. Within the subnetwork, the nodes may use the same protocols that are used between the end systems and the access nodes of the subnetwork. A message may traverse through several nodes of the subnetwork before it is delivered to its destination (Figure 6.6). Routing decisions at each node are taken by the network layer.

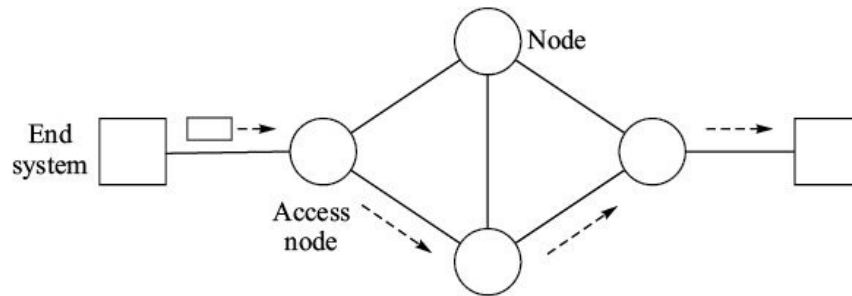


FIGURE 6.6 Routing of packets by the network layer.

6.8.6 Data Link Layer

The primary function of the *data link layer* is to improve the quality of service provided by the physical layer by correcting the errors which are introduced during transmission of electrical signals. It appends error detection bits to a block of data before handing it over to the physical layer for transmission. These bits are used for detecting the errors in the data blocks received by the data link layer at the other end. Usually, reverse error correction mechanisms are employed to correct the errors.

It is necessary that the receiving end be provided with some control to regulate the flow of the incoming frames. Therefore, flow control mechanisms are also an integral part of the error control mechanism in the data link layer.

The data link layer performs another function that is specific to local area networks. In local area networks, terminals share a common transmission media. Media access control function is carried out by the data link layer. Media access control determines which terminal can use the media for its transmission.

6.8.7 Physical Layer

The *physical layer* is primarily concerned with transmission of bits across the interconnecting media. To this end, the physical layer carries out the following functions:

- Conversion of the bits into electrical signals having characteristics suitable for transmission over the media
- Signal encoding, if required
- Relaying of the digital signals using intermediary devices like modems.

The physical layer does not have capability to detect and correct errors that are introduced by the noise and distortion of electrical signals during transmission.

The OSI model provides complete reference for the totality of standards

necessary for open system interconnection. It enables the developers of standards to keep existing standards in perspective, identify areas in existing standards requiring additional development and areas where new standards should be developed.

6.9 FUNCTIONALITY OF THE LAYERED ARCHITECTURE

The layered architecture emphasizes that there is hierarchy of functions. Each layer needs to interact with the peer layer of another end system or the subnetwork to carry out its assigned functions. Since there is no direct communication path between peer layers, they interact using the services of the lower layers. Therefore, two types of communication take place in the layered architecture to make it work properly (Figure 6.7).

- Hierarchical communication.
- Peer-to-peer communication.

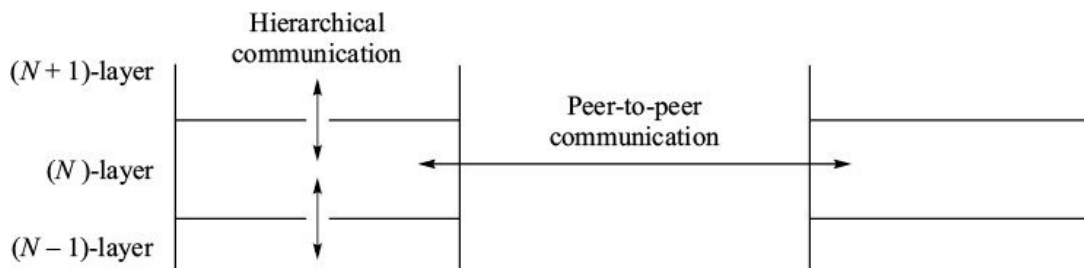


FIGURE 6.7 Hierarchical and peer-to-peer communication.

6.9.1 Hierarchical Communication Hierarchical communication between adjacent layers of a system is for requesting and receiving services from the lower layer. Service interface definition of two adjacent layers specifies rules and procedures for hierarchical communication and description of the services (Figure 6.8). It consists of:

- description of the services provided by the lower layer,
- procedures for requesting and utilizing the services,

- definition of parameters associated with a service which needs to be specified for requesting or utilizing the service.

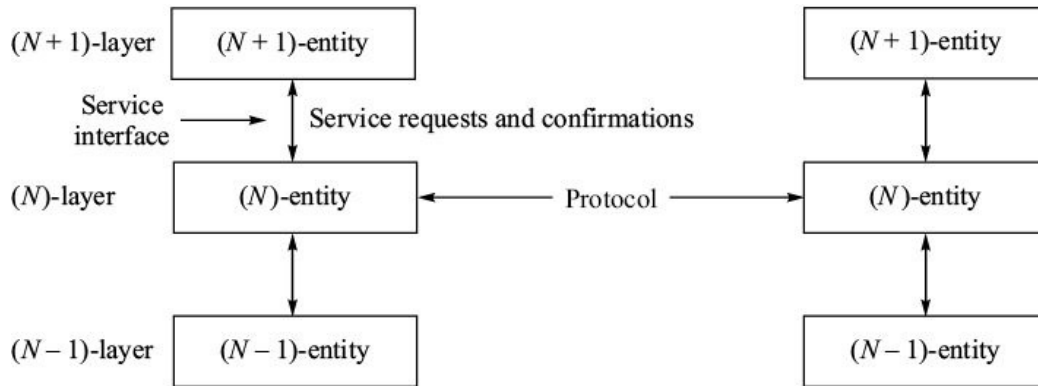


FIGURE 6.8 Service interface.

6.9.2 Peer-to-Peer Communication Peer-to-peer communication is between the peer layers for carrying out an assigned set of functions. Rules and procedures for peer-to-peer communication are called *protocol*. The data units that are exchanged between the peer entities are called Protocol Data Units (PDU). Since there is no direct path between the peer layers, PDUs are exchanged using the services provided by the lower layer. The mechanism commonly utilized is shown in Figure 6.9.

- A layer hands over the PDU to be sent to the servicing layer below it.
- The servicing layer entity in the lower layer attaches a header to the data unit received from the layer above. The header may consist of several fields that contain commands, acknowledgements, sequence numbers, priority, addresses, *etc.* Header is the message to the peer entity. It specifies the action to be taken by the peer entity on the attached data unit. Header with the data unit constitutes the PDU of the layer to be sent to the peer entity. The process is repeated at each layer. Thus each layer generates its PDUs and hands them over to the next lower layer for sending the peer layer.

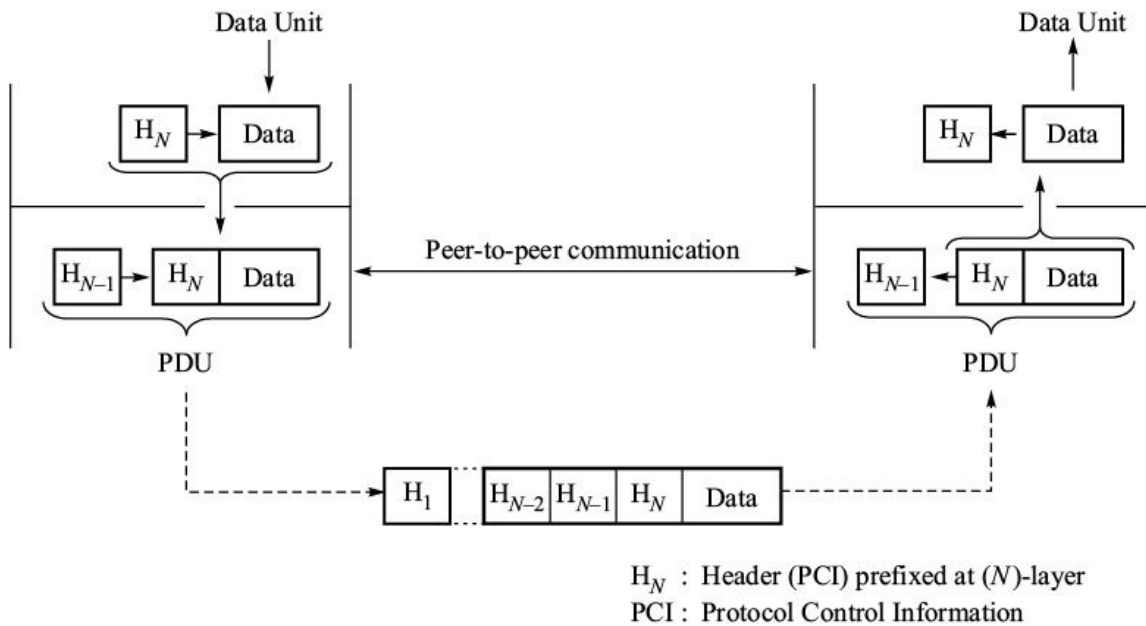


FIGURE 6.9 Peer-to-peer communication.

Header is also referred to as Protocol Control Information (PCI). For the messages to be meaningful to the peer layers, format and content of the PCI fields should be known and agreed by the peer layers in advance. The protocol definition of each layer specifies these attributes of the PCI fields.

- At the other end, each layer receives PDU from the lower layer. The layer entity strips off the header from the received block of data, interprets the header, and undertakes the required action consistent with the protocol.
- The remaining part of the PDU is handed over to the layer above.

It is not necessary that all the PDUs exchanged between the peer entities carry data units received from the upper layer. Some of the PDUs are just commands, responses, acknowledgements or requests for retransmission between peer layers.

6.10 OSI TERMINOLOGY

A very well structured terminology is used in the OSI model to define its functionality. This terminology is extensively used in the definition of the services and protocols.

Layer designation. The following initials are used for specifying the layer to

which an entity, a data unit or a primitive belongs.

Application layer	layer	A	Presentation
Session layer		S	
Transport layer		T	
Network layer		N	
Data link layer		DL	
Physical layer		Ph	

Thus T-PDU implies a PDU of the transport layer. Any entity, data unit or primitive of an unspecified layer prefixed with (N), indicates that it belongs to the Nth layer.

Connection. A connection is the logical association of peer entities for providing services to the next higher layer. In Figure 6.10, (N + 1)-entities communicate over the (N)-connection established by the (N)-entities on the request of (N + 1)-entities. They send their protocol data units transparently over the connection.

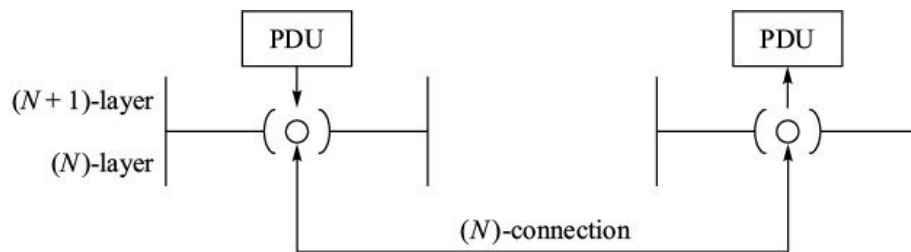


FIGURE 6.10 (N)-connection for providing services to (N + 1)-layer.

Service access point (SAP). For hierarchical communication, the adjacent layer entities interact through a *service access point* which is at the interface between the layers (Figure 6.11). Each service access point supports one communication path.

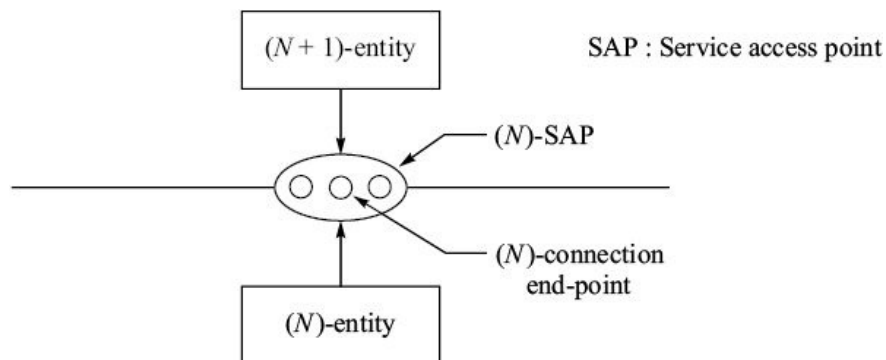


FIGURE 6.11 (*N*)-service access point.

Data units. There are two basic types of data units as follows (Figure 6.12):

- *Protocol Data Unit (PDU)*. (*N* + 1)-PDU is the data unit with header that (*N* + 1)-entity sends to the peer (*N* + 1)-entity using the services of the (*N*)-layer.
- *Service Data Unit (SDU)*. (*N*)-SDU is the data unit received by (*N*)-entity from (*N* + 1)-layer for servicing.

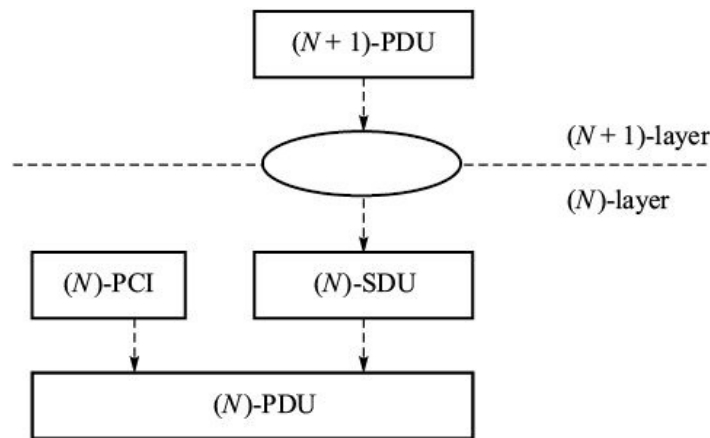


Figure 6.12 Protocol and service data units in the OSI reference model.

6.11 SERVICE INTERFACE

A layer entity carries out its functions using the services provided by the next lower layer. Services are provided across the service access point between the adjacent layers. There can be various types of services at each service interface, which are expressed as

- name of the service provider,
- service name,
- service primitive, and
- associated parameters.

The name initials of layers as indicated in section 6.10 are used to specify the service provider. The name initial is followed by the service name separated by a hyphen. For example, a transport layer service description will start as T-(service

name, primitive, parameters).

Each layer provides a set of services each of which is identified by a name written in upper case. Some of the services and their names are given below:

- Establishing a connection CONNECT
- Transferring a data unit on an established connection DATA
- Disconnecting a connection DISCONNECT
- Transferring a data unit in connectionless-mode of operation, e.g. transferring a datagram. UNITDATA

Thus service description for establishing a transport connection will be T-CONNECT (primitive, parameters). There are many other services but mentioning them at this stage will not serve much purpose. When we describe individual layers in the chapters that follow, we will also examine the services provided by each layer and their names.

6.11.1 Service Interface Primitives and Parameters Service names are associated with service primitives and parameters for providing and using the services. The four basic primitives standardized by ISO and ITU-T are request, indication, response, and confirmation. Figure 6.13 depicts an example of primitives used by $(N + 1)$ -entity to establish (N) -connection. Service primitives are represented in lower case and are written after the service name, leaving a space in between.

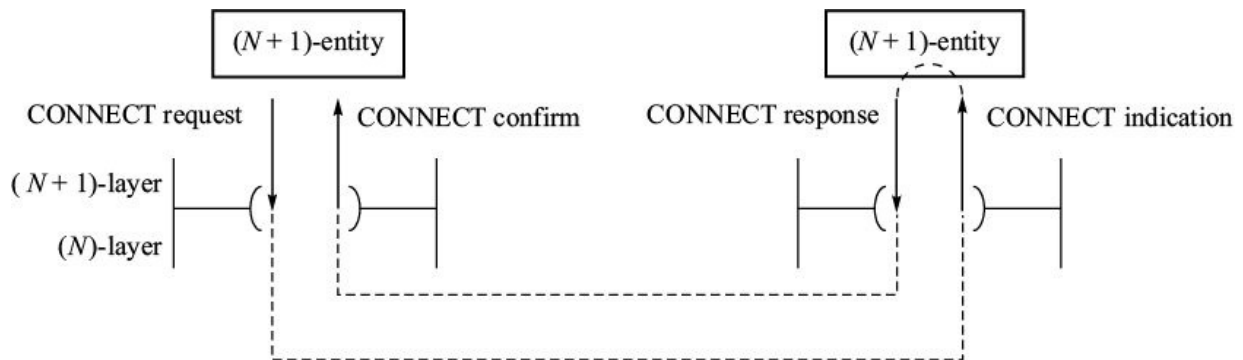


FIGURE 6.13 Service interface primitives for connection establishment.

Request. It is issued by the $(N + 1)$ -entity to request (N) -entity to invoke a particular procedure, e.g. CONNECT request is requested to establish a connection.

Indication. This primitive is issued by (N) -entity to $(N + 1)$ -entity to indicate

that a procedure has been invoked by its peer entity. Thus indication primitive is always an outcome of receipt of request primitive at the peer service provider in the other end system. For example, CONNECT indication is given by (N)-entity to indicate to ($N + 1$)-entity that connection establishment procedure has been invoked by the peer entity at the other end.

Response. It is issued by the ($N + 1$)-entity to (N)-entity in reply to the indication primitive previously received by it. For example, CONNECT response is given by ($N + 1$)-entity in response to CONNECT indication.

Confirm. This primitive is used by the (N)-entity to indicate the completion of some procedure previously invoked by a request primitive by the ($N + 1$)-entity. Usually the confirmation primitive will be given after the response primitive is received by the peer service provider entity in the other end system. For example, if the requested connection is established, (N)-entity confirms this to ($N + 1$)-entity by giving CONNECT confirm.

There is need to indicate service parameters along with service name and primitive. These parameters depend on the service type. The following is a typical example of description of request for invoking the connection-mode service of network layer by transport layer.

N-CONNECT request (calling and called addresses, network connection identification, quality of service).

The above service description includes the:

- name (N) of the service provider which is the network layer;
- name of the service (CONNECT) which indicates that the service relates to establishment of a connection;
- service primitive (request) which indicates the connection establishment service is being invoked; and
- parameters within the parentheses.

The set of parameters associated with the request primitive are:

- the source address required to identify the calling entity to the called entity;
- the called address that enables the network layer entity to establish the connection;
- the network connection identification parameter for future reference (Data units bearing this identification will automatically be transferred across this

- connection); and
- quality of service which is specified using one or more parameters such as transit delay, error rate, *etc.*

6.11.2 Types of Services

There are three types of services:

- Confirmed service
- Non-confirmed service
- Provider initiated service.

Confirmed service. After providing the service, (N)-entity confirms this to the ($N + 1$)-entity using confirm primitive. Figure 6.13 illustrates an example of a confirmed service. All the four service primitives request, indication, response, and confirm are required for confirmed service.

Non-confirmed service. The service requested by the ($N + 1$)-entity is provided by the (N)-entity but confirmation of having provided the service is not given to the requesting ($N + 1$)-entity. The service primitives used in non-confirmed service are request and indication (Figure 6.14).

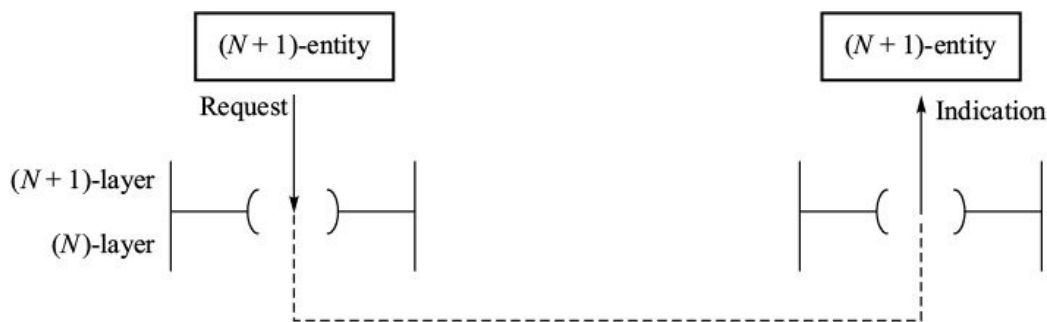


FIGURE 6.14 Non-confirmed service primitives.

Provider initiated service. In this case the (N)-entity initiates and provides the service without any request. The only service primitive required is indication (Figure 6.15). For example, the network layer may disconnect a network connection on its own and report the disconnection using such service to the transport layer.

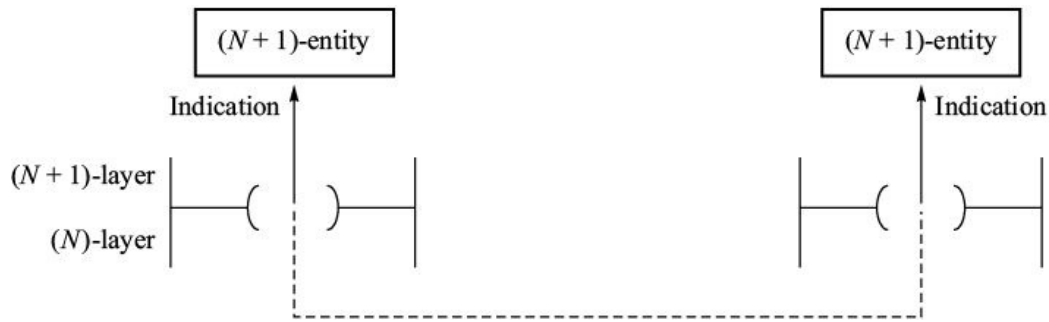


FIGURE 6.15 Provider-initiated service primitives.

6.12 DATA TRANSFER MODES

There can be two alternative modes of transferring PDUs between peer entities:

- Connection-oriented mode of data transfer
- Connectionless mode of data transfer.

6.12.1 Connection-Oriented Mode of Data Transfer A layer entity can transfer data to the peer entity using the connection-oriented mode of data transfer if such service is supported by the next lower layer entity. Connection-oriented mode of data transfer involves three phases:

- Connection establishment phase
- Data transfer phase
- Connection release phase.

Connection is established between the communicating $(N + 1)$ -entities using the service provided by the (N) -entities. Establishment of a connection involves negotiation of quality of service parameters, exchange of source and destination addresses, alerting the distant end entity, acceptance of the incoming call, and confirmation of having established the connection. The connection once established provides sequenced delivery of the data units. After the data transfer has taken place, there is need to disconnect the connection.

Table 6.2 shows the service primitives and the parameters for the three phases of connection-oriented mode of data transfer. Note that DATA service does not have response and confirm primitives. It does not mean that the received PDUs

are not acknowledged. The acknowledgement is sent as part of another PDU or as separate PDU by the receiving entity by invoking DATA service in the reverse direction.

TABLE 6.2 Service Primitives of Connection-oriented Mode of Data Transfer		
Service name	Primitives	Parameters
CONNECT	Request, indication, response, confirm	End point identifiers, quality of service, etc.
DATA	Request, indication	User data (PDU)
DISCONNECT	Request, indication	End point identifier, reason

6.12.2 Connectionless Mode of Data Transfer

Connectionless data transfer is a single self-contained action without establishing, maintaining, and releasing a connection. Connectionless mode service provides transmission of one (N)-SDU from a source (N)-SAP to another (N)-SAP. There is no prior negotiation between the service users and the service providers. There is no assurance of delivery of the data unit. Each instance of (N)-SDU transfer is treated as independent of past such instances.

The connectionless service name is UNITDATA and the associated service primitives are request and indication (Figure 6.16). The parameters associated with the primitives are source (N)-SAP address, destination (N)-SAP address, quality of service parameters, and (N)-user data.



FIGURE 6.16 Primitives for connectionless data transfer.

6.13 SUPPLEMENTARY FUNCTIONS

The layer entities carry out the following supplementary functions to facilitate transport of data units:

- Multiplexing of connections

- Segmenting of data units
- Blocking of data units
- Concatenation of data units.

Multiplexing is a mapping function performed on the connections while segmenting, blocking, and concatenation functions are performed on the data units.

6.13.1 Multiplexing of Connections Multiplexing can be of two types:

- Upward multiplexing
- Downward multiplexing.

Upward multiplexing. In *upward multiplexing*, several (N) -connections are mapped into one $(N - 1)$ -connection (Figure 6.17a). *Demultiplexing* is the reverse process performed at the other

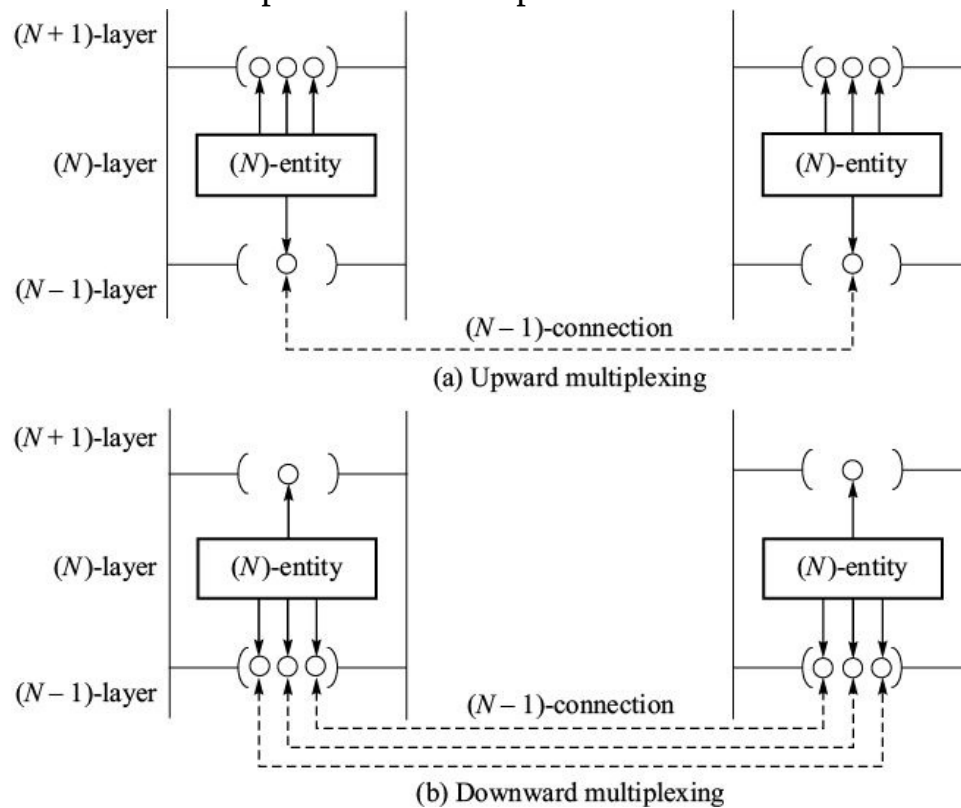


Figure 6.17 Upward and downward multiplexing.

end. Upward multiplexing may be performed by an (N) -entity in order to make

more efficient use of the $(N - 1)$ -connection, and provide several (N) -connections in an environment where only one $(N - 1)$ -connection exists.

Downward multiplexing. In *downward multiplexing*, one (N) -connection is mapped into several $(N - 1)$ -connections (Figure 6.17b). Recombining is the reverse process performed at the other end. Downward multiplexing may be performed by an (N) -entity in order to increase reliability of the (N) -connection by having access to several $(N - 1)$ -connections; and to provide the required grade of performance in terms of increased throughput and reduced delivery delay.

6.13.2 Segmenting, Blocking, and Concatenation of Data Units

Segmenting, blocking, and concatenation are carried out by the layer entities to accommodate incompatible sizes of the data units.

Segmenting.¹ (N) -entity may segment an (N) -SDU into several (N) -PDUs within an (N) -connection (Figure 6.18). At the other end of the connection, the (N) -PDUs are reassembled into one (N) -SDU. An SDU always preserves its identity between the ends of a connection.

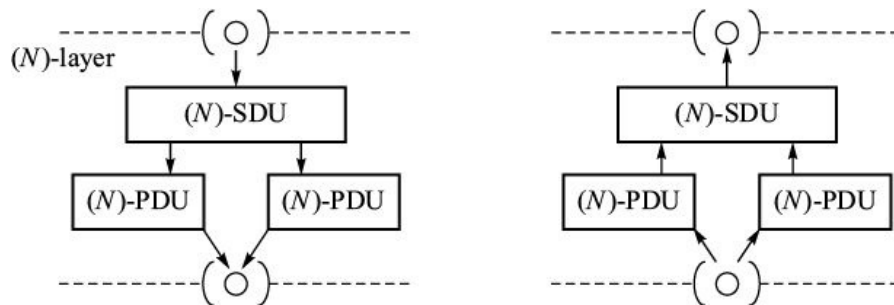


FIGURE 6.18 Segmentation and reassembly.

Blocking. In *blocking*, several (N) -SDUs with their (N) -PCIs are mapped into one (N) -PDU. *Deblocking* is the reverse process carried out at the other end of the connection to get back the (N) -SDUs (Figure 6.19).

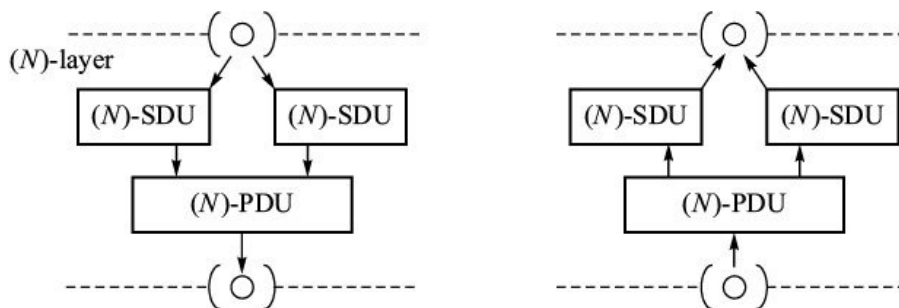


FIGURE 6.19 Blocking and deblocking.

Concatenation. *Concatenation* involves mapping of several (N)-PDUs into a single ($N-1$)-SDU (Figure 6.20). The concatenated PDUs are separated by the receiving peer entity. Enough control information must be conveyed to enable the individual PDUs to be separated. This is accomplished by carrying a length field with each PDU.

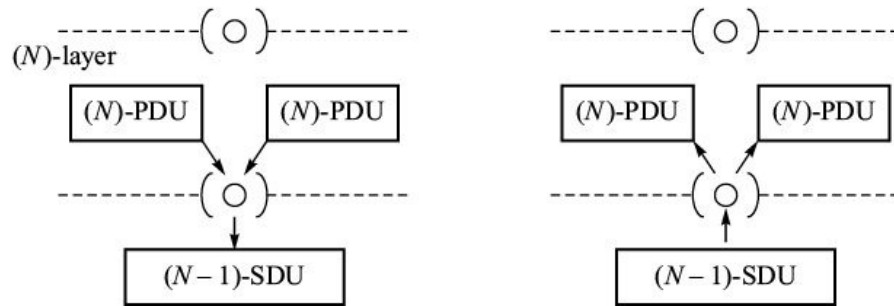


Figure 6.20 Concatenation.

6.14 OTHER LAYERED ARCHITECTURES

As mentioned in section 6.6, there are several network architectures that exist nowadays. Figure 6.21 depicts these architectures with respect to the OSI reference model. The OSI and TCP/IP are open standards. OSI is the most structured reference model but its usage and industry

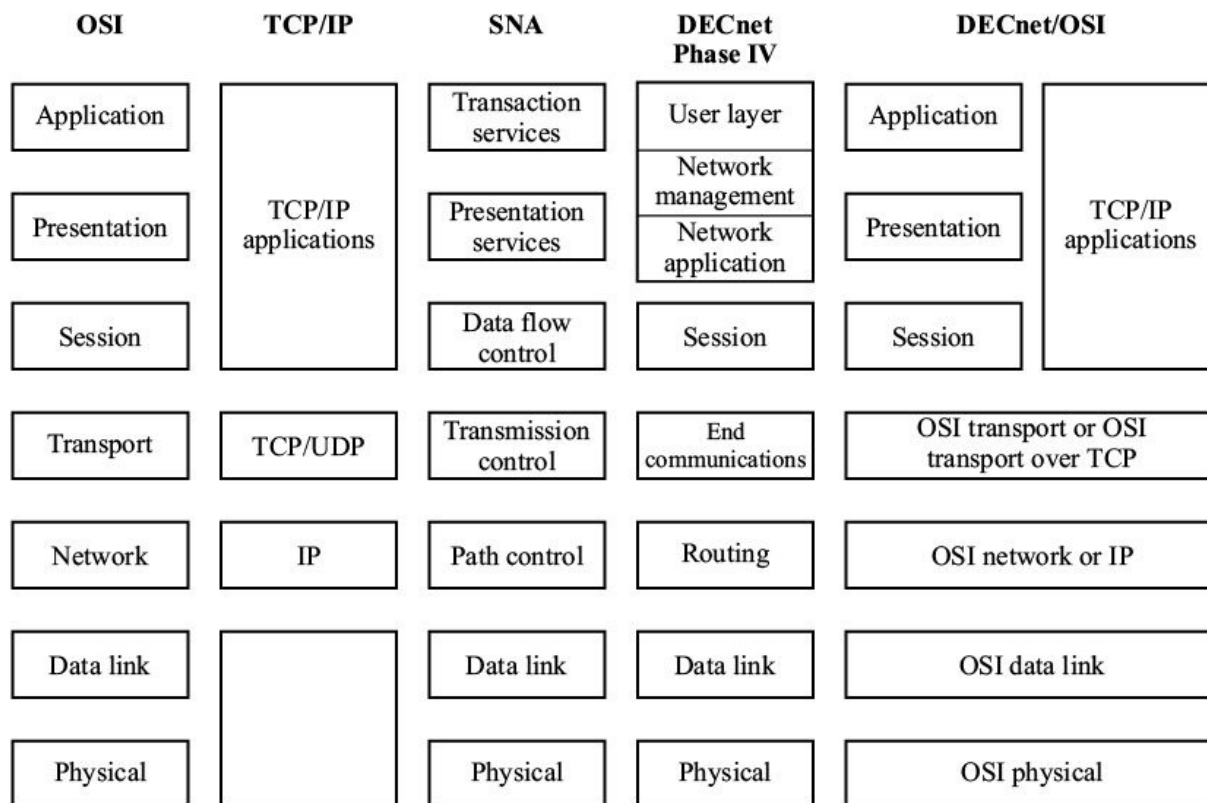


FIGURE 6.21 Network architectures.

acceptance has been low due to sluggish and cumbersome standardization process. TCP/IP, on the other hand, is widely deployed. The rest of the network architectures are vendor-specific and have inter-operability issues.

6.14.1 TCP/IP

TCP/IP protocol suite is the most widely deployed network architecture today. It specifies the architecture of three layers:

- TCP/IP applications layer, which corresponds roughly to the application, session, and presentation layers of the OSI reference model.
- TCP/UDP layer which corresponds to the transport layer of the OSI reference model. TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are the two specified protocols.
- IP layer corresponds to the network layer of the OSI reference model. Internet Protocol (IP) is the specified protocol of this layer.

TCP/IP protocol suite works on host of data link layer types (ATM, Ethernets, FDDI, HDLC, PPP, etc.). This gives TCP/IP suite the required operational

flexibility.

6.14.2 Systems Network Architecture (SNA) IBM's SNA is one of the first network architectures that had the largest implementation base. It was developed in 1970s and is often referred to as legacy architecture. It has seven layers as in OSI architecture. The logical sequence of various functions is also same but some of layer names are different, e.g. path control layer performs the same function as the network layer of OSI reference model.

6.14.3 Digital Network Architecture (DNA) DNA is comprehensive layered architecture developed by Digital Equipment Corporation. It supports several proprietary and standard protocols. Several data networking products known by the generic name DECnet are available based on this architecture. The first DECnet release was in 1975. DECnet Phase IV is proprietary product of Digital with eight layers. The top four layers are user layer, network management layer, network application layer, and session control layer, and correspond roughly to the top three OSI layers.

DECnet/OSI is the most recent version. It supports a subset of OSI protocols and TCP/IP protocols in addition to proprietary protocols of Digital. Compatibility with TCP is achieved by implementing OSI transport over TCP. DECnet/OSI also supports the top four proprietary layers of DECnet Phase IV over its OSI transport layer.

6.15 STANDARDS MAKING ORGANIZATIONS

Tremendous amount of activity is presently underway in standards development. Many standards have been developed, many are still in draft stage and development of some is yet to be attempted. We will examine some of the important standards for services and protocols in the remaining chapters.

Development of standards has been taken up by several organizations. The important standards-making organizations are listed below.

- International Telecom Union-Telecommunication Standards Sector (ITU-T).
- International Organization for Standardization (ISO).
- Institute of Electrical and Electronics Engineers (IEEE).
- Internet Society (ISOC).
- American National Standards Institute (ANSI)
- Electronic Industries Association (EIA).

ITU-T. The International Telecommunication Union-Telecommunication Standards Sector (ITU-T) is an international organization within United Nations, where governments and the private telecommunication sector coordinate global telecom networks and services. It was known as Consultative Committee for International Telegraphy and Telephony (CCITT) till 1993. ITU-T is the standardization organization for all the telecom networks and services. OSI architecture and its standards have been developed by ITU-T along with ISO.

ISO. The International Organization for Standardization (ISO) is a federation of national standards bodies from more than 140 countries. It is a non-government organization for development of standards for goods and services. It has major contributions in development of OSI model and its standards.

IEEE. The Institute of Electrical and Electronics Engineers (IEEE) is the largest professional engineering society in the world and it aims for advancements in all the areas relating to electrical engineering, electronics, and communications. It has special committee IEEE 802 for development of standards for networks based on layer 2, *e.g.* local area networks (LAN), metropolitan area networks (MAN).

Internet society (ISOC). It is a professional society with more than 150 organizational and 6000 individual members. It is the highest body for growth of usage and applications of the Internet. It hosts Internet Architecture Board (IAB) which provides the focus and coordination for much of the research and development of TCP/IP protocols. IAB contains two major groups, Internet Research Task Force (IRTF) and Internet Engineering Task Force (IETF). IRTF coordinates and prioritizes the focus of research from the long term perspective. IETF provides engineering solutions and extensions to the TCP/IP protocol

suite. The standardization process is based on proposals called RFCs (request for comments). IETF then decides which RFCs will be accepted as standards.

ANSI. The American National Standards Institute (ANSI) is a private non-profit organization. ANSI members include professional societies, industry associations, government and regulatory bodies, and consumer groups. ANSI submits its proposals to ITU-T, ETSI, CEPT and is the designated member from US to ISO.

EIA. The Electronics Industries Association (EIA) is a non-profit organization devoted to the profession of electronics manufacturing. Its activities include lobbying for industrial standards development. Its main contribution has been in the development of standards for the physical interfaces.

SUMMARY

Meaningful communication requires more than mere exchange of bits between application entities that reside in computer end systems. It requires error control, flow control, session synchronization, authentication, and other functions besides exchange of data bits. To implement these functions, a standard network architecture is required. The network architecture is based on hierarchical layered model. Each layer uses the service provided by the lower layer to carry out the assigned functions and in turn, provides services to the next higher layer. Protocols are the rules and procedures of interaction between peer layers of different systems.

There have been several network architectures. Of these, Reference Model for Open System Interconnection (OSI) and TCP/IP suite are the most important. OSI specifies a seven-layered architecture. The seven layers are physical layer, data link layer, network layer, transport layer, session layer, and application layer.

TCP/IP suite specifies three top layers—Internet Protocol (IP) layer, Transmission Control Protocol (TCP), and application layer. IP and TCP layers correspond to the network and transport layers of OSI reference model. TCP/IP suite also provides User Datagram Protocol (UDP) at transport layer for connectionless service. The Internet is based on TCP/IP suite and therefore TCP/IP suite has vast deployed base. Development of standards for OSI architecture has been very slow and long process and therefore there are few networks based on OSI networks.

EXERCISES

- Match the following:

(a) Data encryption	(i) Session layer
(b) Bit synchronization	(ii) Network layer
(c) Media access control	(iii) Presentation layer
(d) Routing	(iv) Data link layer
(e) Login	(v) Physical layer
(f) End-to-end connection of required QOS	(vi) Application layer
(g) Synchronization of dialogue	(vii) Transport layer
- Three post offices cooperate to provide postal service. They are interconnected by mail vans which carry the mail. At each post office three processes are carried out in order to provide the postal service.
 - Collection of mail from service users
 - Sorting of mail
 - Packing of mail for the other post offices.Draw a layered model of the postal department. Define the functions and service provided by each layer. Define possible peer layer interactions. Build required security error control measures in the system. Assume that the mail packing department hands over the mail to the mail van. How will a letter posted within the same postal area be processed?
- Two end systems are interconnected using a pair of modems. A modem has only the physical layer and two ports, one towards the end system and the other towards the second modem. Draw the layer model of the configuration.
- In Figure 6.4, error control is carried out by the four data link layers and the two transport layers. Justify the need for error control at so many stages and levels.
- In Figure 6.4, a user at one of the end system wishes to access a database at the other end system. Write the primitives which are exchanged at various interfaces of the OSI reference model for establishing the connection. Assume confirmed service at each interface.
- In a broadcast network, the transmission from one end system is received by all other end systems. Do we need the network layer?
- A network system has N -layer hierarchy. Applications generate message of length M bytes. Layers 2 to N each adds a header of H bytes to the data unit received from the upper layer. What fraction of network bandwidth is filled

with headers? Assume that the physical media is never idle.

1 Segmenting is called 'fragmentation' in other network architectures.

7

The Physical Layer

Transmission of digital information from one device to another is the basic function for the devices to be able to communicate. This chapter describes the first layer of the OSI model, the *physical layer*, which carries out this function. After examining the services it provides to the data link layer, functions of the physical layer are discussed. Relaying through the use of modems is a very important data transmission function carried out at the physical layer level. Various protocols and interfaces which pertain to the relaying functions are put into perspective. We, then, proceed to examine EIA-232 D, a very important interface of the physical layer. We discuss its applications and limitations. We take a look at two other less popular interfaces, namely, EIA-449 and X.21 before close of this chapter.

7.1 THE PHYSICAL LAYER

Let us consider a simple data communication situation shown in Figure 7.1, where two digital devices A and B need to exchange data bits.

For the devices to be able to exchange data bits, the following requirements must be met:

- There should be a physical interconnecting transmission medium which can carry electrical signals between the two devices.

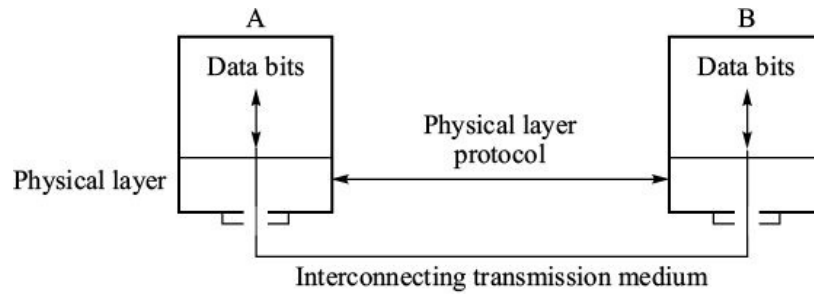


Figure 7.1 Transmission of bits by the physical layer.

- The data bits need to be converted into electrical signals and vice versa.
- The electrical signal should have characteristics (voltage, current, impedance, rise time, etc.) suitable for transmission over the medium.
- The devices should be ready to exchange the electrical signals.

These requirements, which are related purely to the physical aspects of transmission of bits, are met by the physical layer. The rule and procedures for interaction between the physical layers are called physical layer protocols (Figure 7.1).

The physical layer provides its service to the data link layer which is the next higher layer and uses this service. It receives service of the physical interconnection medium for transmitting the electrical signals.

7.1.1 Physical Connection

The physical layer receives the bits to be transmitted from the data link layer (Figure 7.2). At the receiving end, the physical layer hands over these bits to the data link layer. Thus, the physical layers at the two ends provide a bit transport service from one data link layer to the other over a '*physical connection*' activated by them. A physical connection is different from a physical transmission path in the sense that it is at the bit level while the transmission path is at the electrical signal level.

The physical connection shown in Figure 7.2 is point-to-point. Point-to-multipoint physical connection is also possible as shown in Figure 7.3.

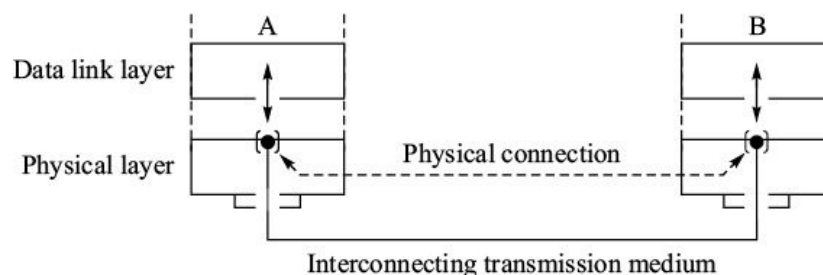


Figure 7.2 Point-to-point physical connection.

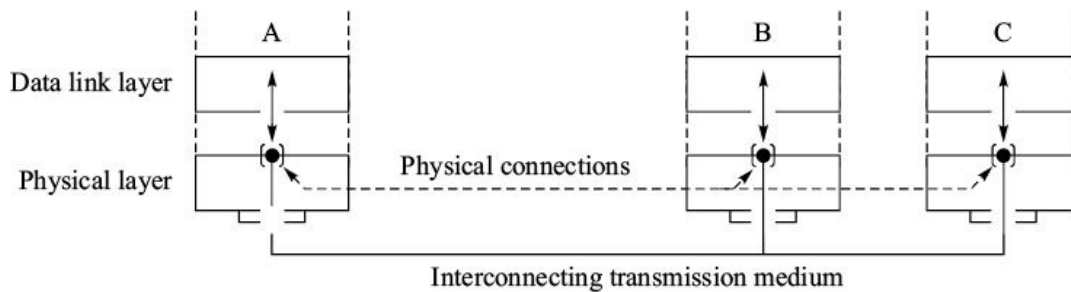


Figure 7.3 Point-to multipoint physical connection.

7.1.2 Service Provided to the Data Link Layer

The service provided by the physical layer to the data link layer is the bit transmission service over the physical connection. The physical layer service is specified in ISO 10022 and ITU-T X.211 documents. Some of the features of this service are now described.

Activation/deactivation of the physical connection. The physical layer, when requested by the data link layer, activates and deactivates a physical connection for transmission of bits. Activation of the physical connection ensures readiness of the physical layer to transport the bits across the connection. The activation and deactivation service is non-confirmed, *i.e.* the data link layer activating or deactivating a connection is not given any feedback of the action having been carried out by the physical layer.

A physical connection may allow full duplex or half duplex transmission of the bits. In half duplex transmission, the data link layers themselves decide which of the two users may transmit. This control is not exercised by the physical layer.

Transparency. The physical layer provides transparent transmission of the bit stream between the data link entities over the physical connection. Transparency implies that any bit sequence can be transmitted without any restriction imposed by the physical layer.

Physical service data units (Ph-SDU). Ph-SDU received from the data link layer consists of one bit in serial transmission and of 'n' bits in parallel transmission.

Sequenced delivery. The physical layer tries to deliver the bits in the same sequence as they were received from the data link layer. It does not carry out any error control. Therefore, it is likely that some of the bits are altered, some are not

delivered at all, and some are duplicated.

Fault condition notification. Data link entities are notified in case of any fault detected in the physical connection.

Service primitives. The physical layer provides a non-confirmed service to the data link layer. The service names and primitives for activation of the physical connection, data transfer and for deactivation of the physical connection are shown in Table 7.1.

TABLE 7.1 Service Primitive of the Physical Layer

Service	Primitive
Connection activation	Ph-ACTIVATE request Ph-ACTIVATE indication
Data transfer	Ph-DATA request Ph-DATA indication
Connection deactivation	Ph-DEACTIVATE request Ph-DEACTIVATE indication

7.2 FUNCTIONS WITHIN THE PHYSICAL LAYER

To provide the services as listed above to the data link layer, the physical layer carries out the following functions:

- It activates and deactivates the physical connection at the request of the data link layer entity. These functions involve interaction of the physical layer entities. The peer physical layer entities exchange control signals for this purpose.
- A physical connection may necessitate the use of a relay at an intermediate point to regenerate the electrical signals (Figure 7.4). Activation and deactivation of the relay is carried out by the physical layer. This function is explained in detail in the next section.

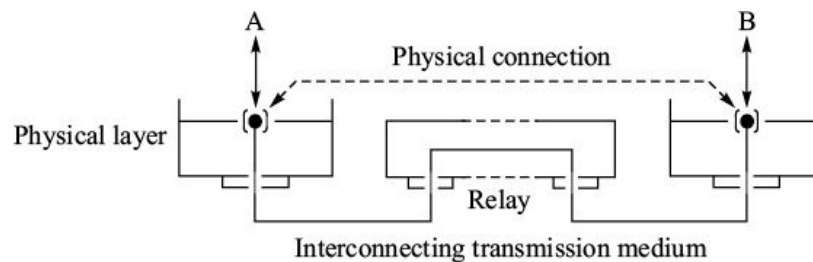


FIGURE 7.4 Relaying function of the physical layer.

- The physical transmission of the bits may be synchronous or asynchronous. The physical layer provides synchronization signals (clock) necessary for transmission of the bits. Character level or frame level synchronization is the responsibility of the data link layer.
- If signal encoding is required, it is carried out by the physical layer. Thus the line coding we learnt in Chapter 1 is implemented in the physical layer.
- The physical layer does not incorporate any error control function.

7.3 RELAYING FUNCTION IN THE PHYSICAL LAYER

It may not always be practical to directly connect two digital devices using a cable if the distance between them is very long. The quality of the received signals gets degraded by noise, attenuation, and phase characteristics of the interconnecting medium. Signal Converting Units (SCUs) are used in the physical interconnecting medium as relays to overcome these problems (Figure 7.5).

SCUs employ one or more of the following methods to ensure acceptable quality of the signal received at the distant end:

- Amplification
- Regeneration
- Equalization of media characteristics
- Modulation.

Examples of SCUs which carry out these functions are: modems, LDMs (Limited Distance Modems), and line drivers. A pair of these devices is always required, one at each end. These two devices together act as a relay. They receive electrical signals representing data bits at one end and deliver the same signals at the other end.

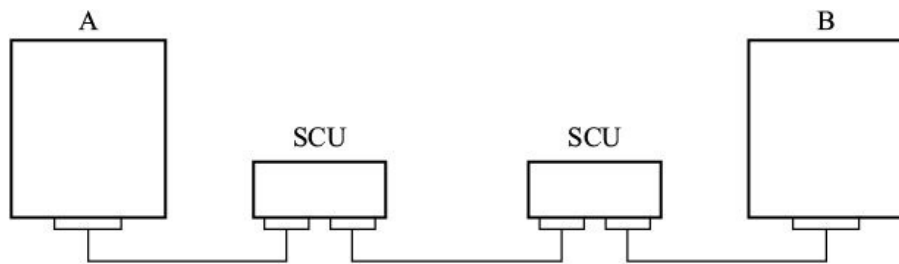
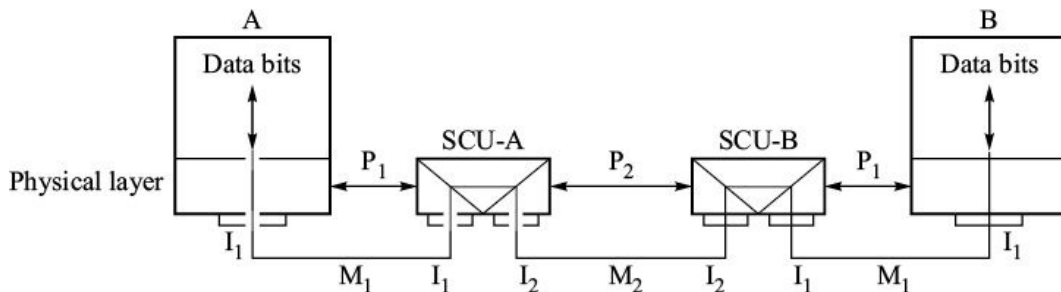


Figure 7.5 Signal converting unit (SCU).

The digital end devices face the SCUs and interact with the SCUs at the physical layer level. This is shown in detail in Figure 7.6. Notice that a number of protocols and interfaces at physical layer level are involved when SCUs are used as relay units.



- M_1 : Transmission medium between end device and SCU.
- M_2 : Transmission medium between SCUs.
- I_1 : Physical interface between end device and SCU.
- I_2 : Physical interface between SCUs.
- P_1 : Physical layer protocol between end device and SCU.
- P_2 : Physical layer protocol between SCUs.

Figure 7.6 Physical interfaces and protocols while using signals converting units.

The transmission media M_1 and M_2 are usually different. M_1 consists of a bunch of copper wires, each carrying data or a control signal. M_2 , on the other

hand, can be a balanced copper pair, telephony channel or even optical fibre. Physical interfaces I_1 and I_2 depend on the type of SCU and the transmission medium used.

As regards the physical layer protocols, note that the physical layer of end device A no longer interacts with the physical layer of end device B. It interacts with the physical layer of SCU-A to carry out the physical layer functions. The two SCUs have a different physical layer protocol between them.

7.4 PHYSICAL INTERFACE

The physical layers need to exchange protocol control information between them. Unlike the other layers which send the protocol control information as a header, the physical layers use the interconnecting medium for sending the protocol control signals. These signals are sent on separate wires as shown in Figure 7.7. Note that the control signals originate and terminate in the physical layers. They have no functional significance beyond the physical layer. This is in conformity with the principles of the layered architecture.

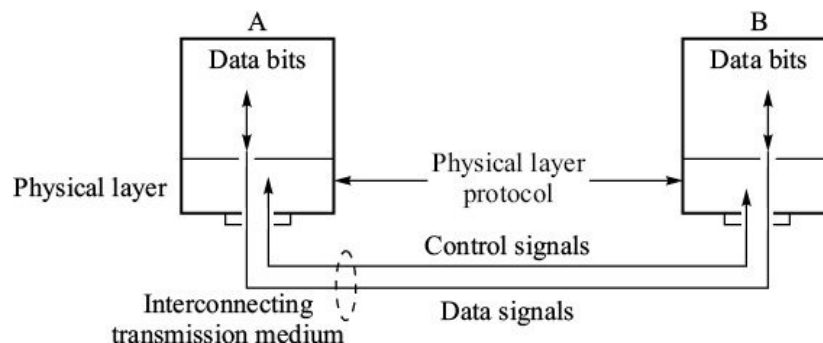


FIGURE 7.7 Transmission of control signals of the physical layer.

The physical interconnecting medium consists of a number of wires carrying data and control signals. It is essential to specify which wire carries which signal. Moreover, the mechanical specifications of the connector, type of the connector (male or female) and the electrical characteristics of the signals need to be specified. Definition of the physical layer interface includes all these specifications.

7.5 PHYSICAL LAYER STANDARDS

Historically, the specifications and standards of the physical media interface have also covered the physical layer protocols. But these specifications have not identified the physical layer protocols as such. Physical layer specifications can be divided into the following four components (Figure 7.8):

- Mechanical specification
- Electrical specification
- Functional specification
- Procedural specification.

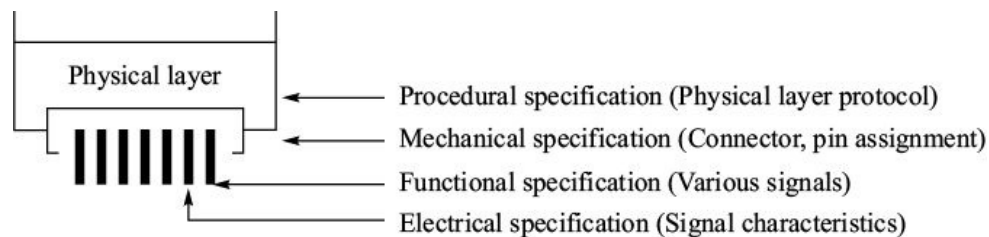


Figure 7.8 Physical layer specifications.

The procedural specification is the physical layer protocol definition and the other three specifications constitute the physical layer interface specifications.

- The mechanical specification gives details of the mechanical dimensions and the type of connectors to be used on the device and the medium. Pin assignments of the connector are also specified.
- The electrical specification defines the permissible limits of the electrical signals appearing at the interface in terms of voltage, current, impedance, rise time, *etc.* The required electrical characteristics of the medium are also specified.
- The functional specification indicates the functions of various control signals.
- The procedural specification indicates the sequence in which the control signals are exchanged between the physical layers for carrying out their functions.

Although there are many standards of the physical layer, only a few are of wide significance. Some examples of physical layer standards are given below:

EIA : EIA-232D, EIA-449, EIA-422A, EIA-423A

ITU-T : V.10, V.11, V.24, V.28, V.35

X.20, X.20bis, X.21, X.21bis

ISO : ISO 2110

Out of the above, the EIA-232-D interface is the most common and is found in almost all computers. ITU-T V-series recommendations indicated above are equivalent to EIA standards. We will examine EIA-232-D in detail in the following sections. Other less important physical layer standards will also be discussed in brief. Local area networks (LAN) have different set of physical layer protocols. We will discuss these in the Chapters 10, 11 and 12.

7.6 EIA-232-D DIGITAL INTERFACE

The EIA-232-D digital interface of Electronics Industries Association (EIA) is the most widely used physical medium interface. It is applicable to the following modes of transmission:

- Serial transmission of data
- Synchronous and asynchronous transmission
- Point-to-point and point-to-multipoint working
- Half duplex and full duplex transmission.

7.6.1 DTE/DCE Interface

EIA-232-D is applicable to the interface between a Data Terminal Equipment (DTE) and a Data Circuit Terminating Equipment (DCE) (Figure 7.9). The computer terminals are usually DTEs. The DTEs are interconnected using two intermediary devices (e.g. modems) which carry out the relaying function. The intermediary devices are categorized as DCEs.

Two types of physical layer interfaces are involved in the above configuration:

- Interface between a DTE and a DCE
- Interface between the DCEs.

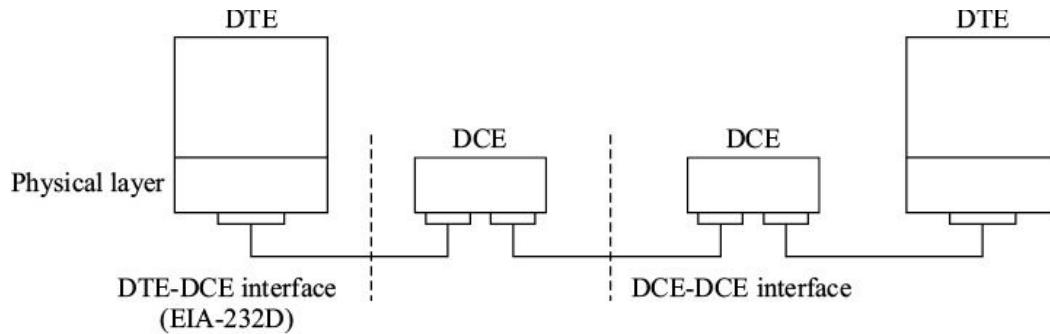


Figure 7.9 DTE/DCE interfaces at the physical layer.

EIA-232-D defines the interface between a DTE and a DCE. The physical medium interconnecting the DTE and the DCE consists of bunch of wires circuits carrying data, control, and timing signals from DTE to DCE and vice versa. The return path for all the signals is a common signaling ground wire.

7.6.2 DTE and DCE Ports

Over the years, use of the term DTE and DCE for classifying two kinds of devices on the basis of their functions has declined. We can have today a terminal equipment which looks like a DCE at its transmission port. The situation becomes even more confusing when we come across a device which has multiple ports of different types (Figure 7.10). Therefore, the use of the terms DTE and DCE at the physical layer level refers to the description of the transmission port of a device rather than the device itself.

EIA-232-D has been designed with modem as DCE, and the terminology which has been used to specify its signals and functions also refers to DCE as modem. We shall, therefore, describe the interface with modem as the DCE.

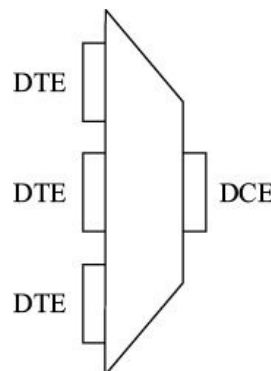


Figure 7.10 Multiplexer with DTE and DCE parts.

7.6.3 DCE-DCE Connection

A DCE has two interfaces, DTE-side interface which is EIA-232-D, and the line-

side interface which interconnects the two DCEs through the transmission medium (Figure 7.11). EIA-232-D is not applicable to DCE-DCE interface. There are other standards for DCE-DCE interface.

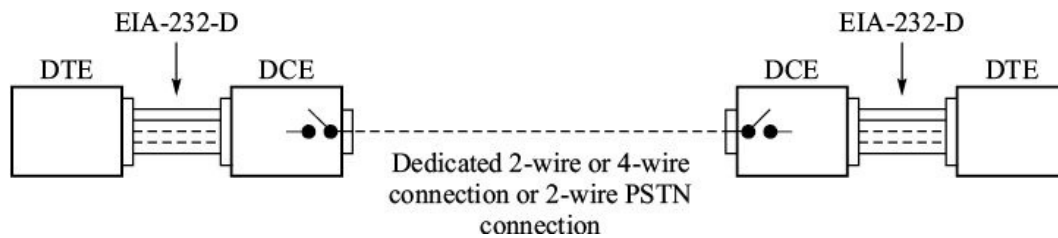


Figure 7.11 Transmission alternatives between two DCEs.

The connection between the two DCEs and mode of transmission between them are controlled by the control signals that flow between the DTE and DCE. There can be several forms of connection and modes of transmission between the DCEs:

- The two DCEs may be connected directly through a dedicated transmission medium.
- The dedicated connection may be on a 2-wire transmission circuit or on a 4-wire transmission circuit.
- The two DCEs may be connected to PSTN (Public Switched Telephone Network).
- The mode of transmission between the DCEs may be either full duplex or half duplex.

Full duplex mode of transmission is easily implemented on a 4-wire dedicated circuit. Full duplex operation on a 2-wire circuit requires two communication channels which are provided on two different carrier frequencies on the same medium. PSTN provides a 2-wire circuit between the DCEs and the circuit needs to be established and released using a standard telephone interface. Note that electronics of the DCE may not be directly connected to the interconnecting transmission circuit. This connection is made on request from the DTE as we shall see later.

7.7 EIA-232-D INTERFACE SPECIFICATIONS

EIA-232-D interface defines all the four sets of specifications for the physical layer interface between a DTE and a DCE. ITU and ISO have also adopted the same specifications. Their corresponding specification references are indicated in the brackets.

- Mechanical specifications, (ISO 2110)
- Electrical specifications, (V.28)
- Functional specifications, (V.24)
- Procedural specifications, (V.24).

7.7.1 Mechanical Specifications

Mechanical specifications include mechanical design of the connectors which are used on the equipment and the interconnecting cables; and the pin assignments of the connectors.

EIA-232-D defines the pin assignments and the connector design is as per ISO 2110 standard. A DB-25 connector having 25 pins is used (Figure 7.12). The male connector is used for the DTE port and the female connector is used for the DCE port.

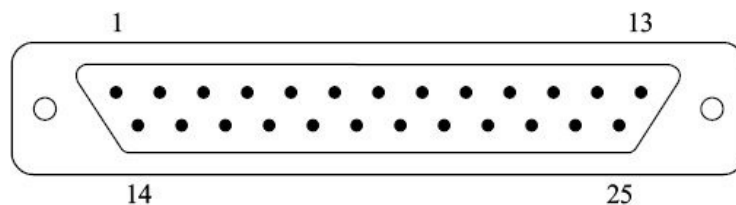


Figure 7.12 25-pin (DB-25) connector of EIA-232-D interface.

7.7.2 Electrical Specifications

The electrical specifications of the EIA-232-D interface specify characteristics of the electrical signals. Line code used is NRZ-L. The limits of voltage levels assigned to the two states of a binary digital signal are shown in Figure 7.13. The nominal voltage level is +12 volts for binary 0 and -12 volts for binary 1. All the voltages are measured with respect to the common ground. The 25-volts limit is the open circuit or no-load voltage. The range from -3 to +3 volts is the transition region and is not assigned any state.

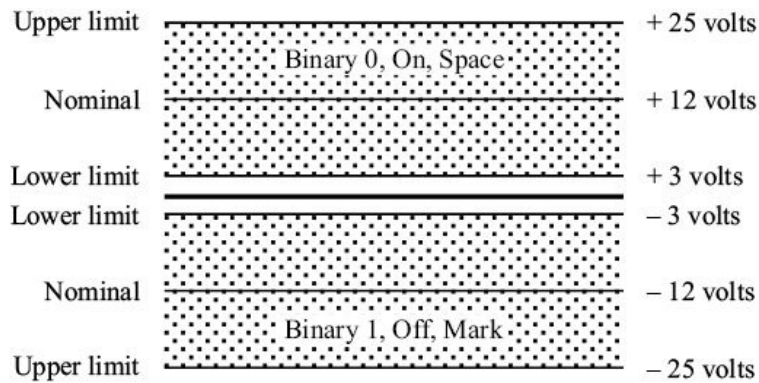


Figure 7.13 Electrical specifications of EIA-232-D interface.

DC resistance of the load impedance is specified to be between 3000 to 7000 ohms with a shunt capacitance less than 2500 pF. The cable interconnecting a DTE and a DCE usually has a capacitance of the order of 150 pF per metre which limits its maximum length to about 16 metres. EIA-232-D specifies the maximum length of the cable as 50 feet (15.3 metres) at the maximum data rate of 20 kbps.

7.7.3 Functional Specifications

Functional specifications describe the various signals which appear on different pins of the EIA-232-D interface. These signals are divided into six categories (Table 7.2):

- Ground or common return
- Data circuits
- Control circuits
- Timing circuits
- Secondary channel circuits
- Test circuits.

TABLE 7.2 EIA-232-D Interchange Circuits			
Pin	Direction		Circuit name (Acronym)
	DTE	DCE	
	Common	Common	ITU-T EIA no. label

1	Shield	101	—
7	Signal Ground	107	
2	Transmitted Data (TD)	103	AB
3	Received Data (RD)	104	BA
4	Request to Send (RTS)	105	BB
5	Clear to Send (CTS)	106	CA
6	DCE Ready (DSR)	107	CB
20	Data Terminal Ready (DTR)	108	CC
22	Ring Indicator (RI)	125	CD
8	Carrier Detect (CD), Received Line Signal Detector	109	CE
21	Signal Quality Detector	110	CF
23	Data Rate Selector from DTE	111	CG
23	Data Rate Selector from DCE	112	CH
24	Transmitter Signal Element Timing (DTE Clock)	113	CI
15	Transmitter Signal Element Timing (DCE Clock)	114	DA
17	Receiver Signal Element Timing (Received Clock)	115	DB
14	Secondary Transmitted Data (S-TD)	118	DD
16	Secondary Received Data (S-RD)	119	SBA
19	Secondary Request to Send (S-RTS)	120	SBB
13	Secondary Clear to Send (S-CTS)	121	SCA
12	Secondary Received Line Signal Detector (S-CD)	122	SCB
18	Local Loopback (LL)	141	SCF
21	Remote Loopback (RL)	140	LL
25	Test Mode (TM)	142	RL
			TM

A circuit implies the wire carrying a particular signal. The return path for all the circuits in both directions (from DTE to DCE and from DCE to DTE) is common. It is provided on pin 7 of the interface. EIA uses a two-or three-letter designation for each circuit. ITU-T labels each circuit with a three-digit number. In day-to-day use, however, acronyms based on the function of individual circuits are more common and we will use these acronyms in the text. Not all the circuits are always wired between a DTE and a DCE. Depending on configuration and application, only essential circuits are wired. Functions of the commonly used circuits are now described.

Signal ground. It is the common earth return for all data and control circuits in both directions. This is one circuit that is always required whatever be the configuration.

Data terminal ready (DTR), DTE DCE. The ON condition of the signal on this circuit informs the DCE that the data terminal equipment, DTE, is ready to operate and the DCE should also connect itself to the transmission medium.

DCE ready (DSR¹), DTE DCE. This circuit is usually turned ON in response to DTR and indicates ready status of the DCE. When this signal is ON, it means that power of the DCE is switched on and it is connected to the transmission medium. If the DCE-to-DCE connection is through PSTN, ON status of the CC implies that the call has been established.

Request to send (RTS), DTE DCE. OFF to ON transition on RTS triggers the local DCE to perform such set-up actions as are necessary to transmit data. These set-up activities include sending a carrier to the remote DCE so that it may further alert the remote DTE and get ready to receive data. Transition of the RTS from ON to OFF asks the DCE to complete transmission of all data and then withdraw the carrier.

Clear to send (CTS), DTE DCE. CTS signal indicates that the DCE is ready to receive data from the DTE on Transmitted Data (TD) circuit. This control signal is changed to the ON state in response to the RTS from the DTE after a predefined delay. This delay is provided to give sufficient time to the remote DCE and DTE to get ready for receiving data. Figure 7.14 illustrates how the RTS signal works with CTS signal to coordinate data transmission between a DTE and a DCE.

Transmitted data (TD), DTE DCE. Data from DTE to DCE is transmitted on

this circuit. When no data is being transmitted, the DTE keeps the signal on this circuit in 1 (OFF) state. Data can be transmitted on this circuit only when the following control signals are ON:

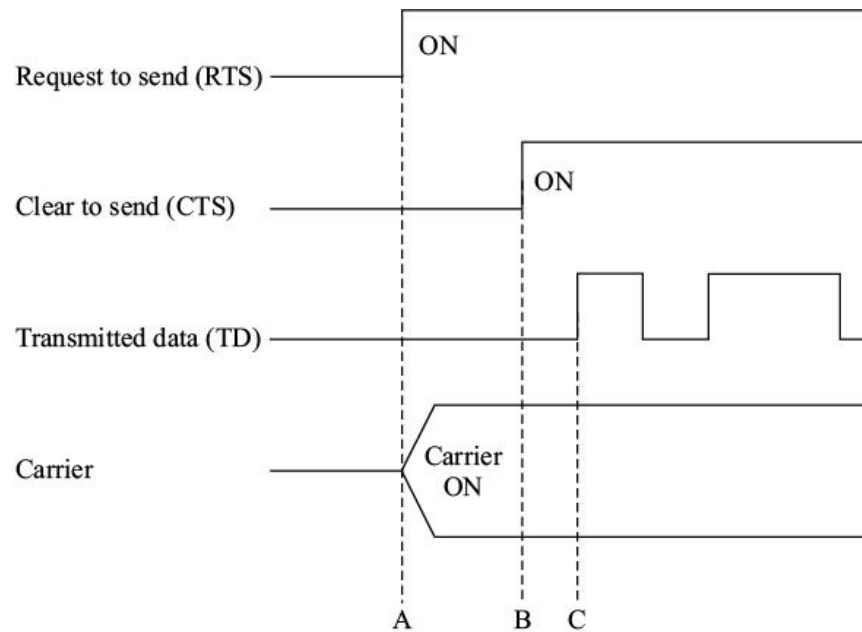
- RTS
- CTS
- DSR
- DTR.

The ON state of these signals ensures that the local DCE is in readiness to transmit data and sufficient opportunity has been given to the remote DCE and DTE to get ready for receiving data.

Received data (RD), DTE DCE. Data from DCE to DTE is received on this circuit. DCE maintains the signal on this circuit in 1 (OFF) state when no data is being received.

Transmitter signal element timing (DTE clock), DTE DCE. When operating in the synchronous mode of transmission, the DTE clock is available to the DCE on this circuit.

Transmitter signal element timing (DCE clock), DTE DCE. When operating in the synchronous mode of transmission, the DCE clock is available to the DTE on this circuit. One of the two clocks, DTE clock or DCE clock, is used as timing reference.



- A : DTE sends RTS indicating its wish to send data.
DCE sends carrier on the line.
- B : DCE indicates its readiness to accept data from DTE.
- C : DCE starts receiving data from DTE.

Figure 7.14 Time sequence of CTS and RTS circuits.

Receiver signal element timing (Received clock), DTE DCE. At the receiving end, this circuit provides the received clock from the DCE to the DTE. This clock is extracted from the received signal by the DCE and is used by the DTE to store the data bits in a shift register.

Figure 7.15 shows two typical methods of configuring the timing circuits. In the first alternative, the DCE supplies clock to the DTE for the transmitted data. At each clock transition, one data bit is pushed out of the DTE. In the second alternative, the DTE supplies clock to the DCE. At the remote end, the clock is extracted from the received data and supplied to the DTE for the received data in both the alternatives.

Carrier detector (CD), Received line signal detector, DTE DCE. On receipt of RTS signal from the DTE, the local DCE sends a carrier to the remote DCE so that it may get ready to receive data. When the remote DCE detects the carrier on the line, it alerts the DTE by turning the CD circuit ON to get ready to receive data.

Ring indicator (RI), DTE DCE. The ON state of this circuit indicates to the DTE that there is an incoming call and the DCE is receiving a ring signal from the telephone exchange. On receipt of this signal the DTE is expected to get

ready and indicate this to the DCE by turning its DTR signal ON.

Secondary channel circuits (S-TD, S-RD, S-RTS, S-CTS, S-CD). These circuits are used when a secondary channel is provided by a DCE. The secondary channel operates at a lower data signaling rate (typically 75 bits/s) than the data channel and is intended to be used for return of supervisory control signals. The control circuits for the secondary channel, S-RTS, S-CTS,

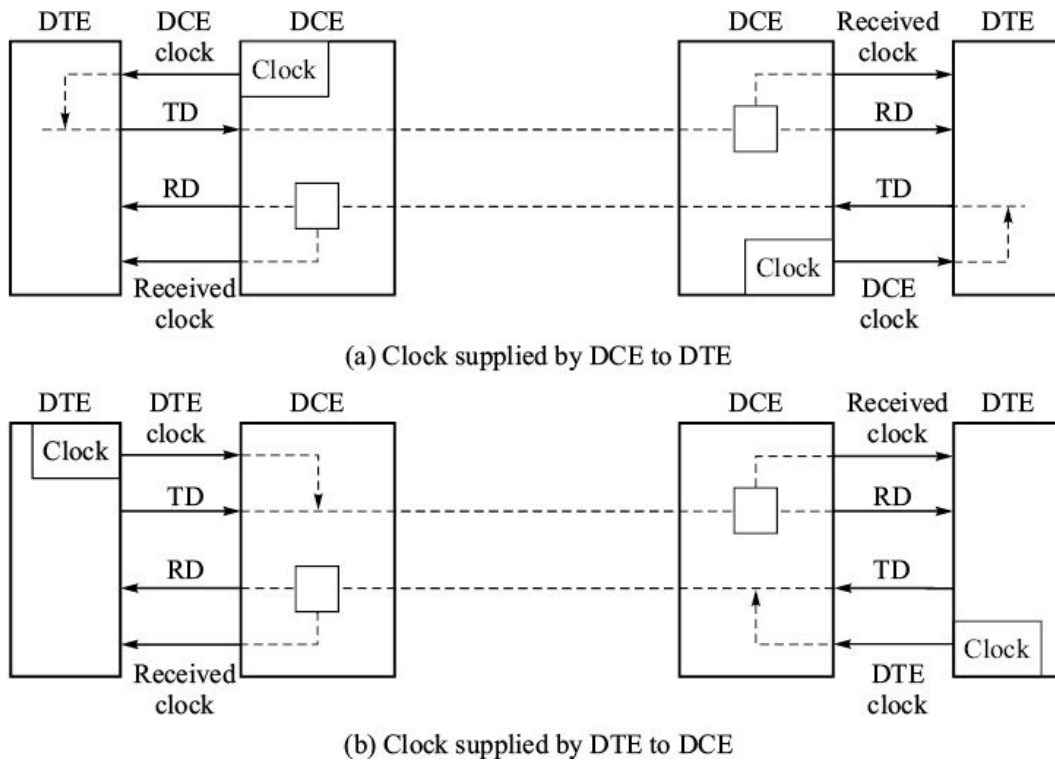


FIGURE 7.15 Clock supply alternatives for synchronous transmission.

and S-CD, are functionally the same as RTS, CTS, and CD except that they are associated with the secondary channel.

Local loopback (LL), DTE DCE. The ON condition of this circuit causes a local loopback at the DCE line output so that the data transmitted on the TD is made available on the received data RD for conducting local tests.

Remote loopback (RL), DTE DCE. The ON condition of this circuit causes loopback at the remote DCE so that the local DCE line and the remote DCE could be tested.

Test mode (TM), DTE DCE. After establishing the loopback condition, the DCE indicates its loopback status to the local DTE by the ON condition of the TM circuit.

7.7.4 Procedural Specifications

Procedural specifications lay down the procedures for the exchange of control signals between a DTE and a DCE. The sequence of events which comprise the complete procedure for data transmission can be divided into the following four phases:

- Equipment readiness phase
- Circuit assurance phase
- Data transfer phase
- Disconnect phase.

Equipment readiness phase. The following functions are carried out during the equipment readiness phase:

- The DTE and DCE are energized.
- Physical connection between the DCEs is established if they are connected to PSTN.
- The transmission medium is connected to the DCE electronics.
- The DTE and DCE exchange signals that indicate their ready state.

We shall consider two simple configurations of connection between the DCEs:

- The DCEs having dedicated transmission medium between them.
- The DCEs having a switched connection through PSTN between them.

Dedicated transmission connection. The DTE that wants to transmit, asserts the DTR signal which causes the DCE electronics to connect to the transmission line. The DCE replies with the DSR signal as shown in Figure 7.16.

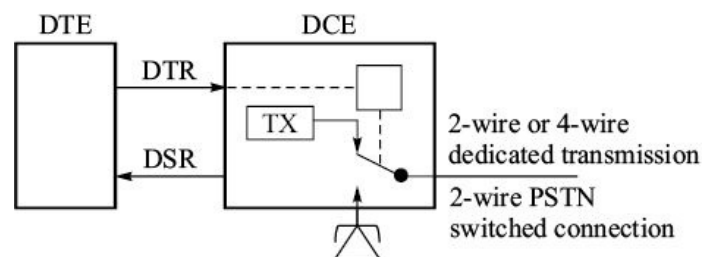


FIGURE 7.16 Equipment readiness phase.

Switched connection. In this case, the physical connection between the DCEs

needs to be established through a switched telephone network. This is done either manually by the operators at both ends or automatically.

In the manual operation, the DCEs are fitted with a telephone instrument and the telephone line is through to the telephone instrument. The operator wishing to establish the connection dials the distant end telephone number and indicates his intent to the distant end operator. The operators then press appropriate switches on their respective DTEs to send the DTR signals to the DCEs. DTR signal causes the transmission medium to changeover from the telephone instrument to the DCE at both ends (Figure 7.17). DCEs indicate their line connected status to DTE on DCE Ready (DSR) circuit.

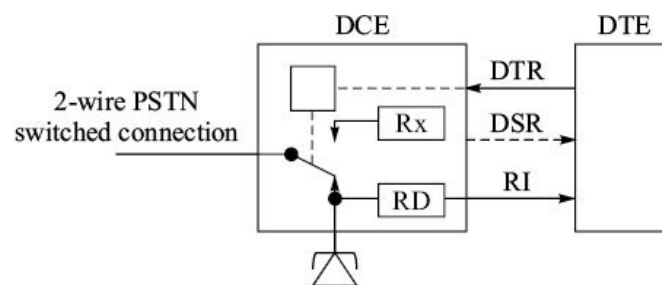


Figure 7.17 Distant end readiness with auto-answering.

If automatic answering facility is used, the incoming call is detected by the Ring Detector (RD) circuit of the call receiving DCE. The incoming call indication is given to the DTE by the Ring Indicator (RI) signal. The DTE sends the DTR signal to the DCE on receipt of RI. DTR causes the connection to changeover from telephone instrument to the DCE receiver electronics. The DCE indicates its readiness status simultaneously to the DTE on the DSR circuit (Figure 7.17).

Thus, at the end of the equipment readiness phase, we have

- ON state of the DTR and DSR signals at both the ends and
- the transmission medium connected to the DCE electronics at both the ends.

Carrier is not yet transmitted on the line.

Circuit assurance phase. In the circuit assurance phase, the DTEs indicate their intent to transmit data to the respective DCEs and the end-to-end (DTE to DTE) data circuit is activated. If the transmission mode is half duplex, only one of the two directions of transmission of the data circuit is activated.

Half duplex mode of transmission. DTE indicates its intent to transmit data by

asserting the RTS signal which activates the transmitter of the DCE and a carrier is sent to the distant end DCE (Figure 7.18). RTS signal also inhibits the receiver of the transmitting end DCE.

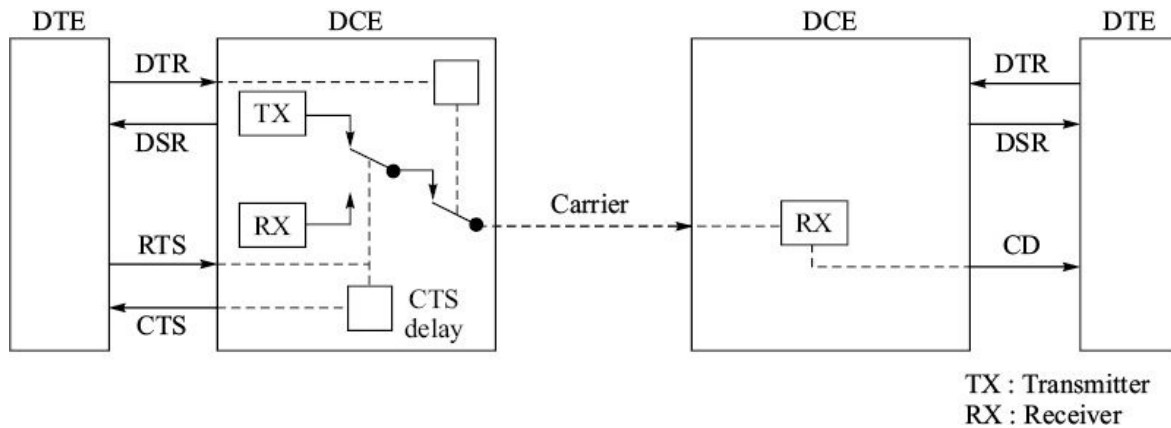


FIGURE 7.18 Circuit assurance phase in half duplex mode of transmission.

After a short interval of time equal to the propagation delay, the carrier appears at the input of the distant end DCE. The DCE detects the incoming carrier and gets ready to demodulate data from the carrier. It also alerts the DTE using the received line signal detector (Carrier detect, CD) circuit (Figure 7.18).

After activating the circuit, the sending end DCE signals the DTE to proceed with data transmission by returning the Clear to Send (CTS) signal after a fixed delay. This delay ensures that sufficient opportunity is given to the distant end to get ready to receive data. With the CTS signal, the equipment readiness and end-to-end data circuit readiness are assured and the sending end DTE can initiate data transmission.

In half duplex operation, the CTS is given in response to RTS only if the local received line signal detector (Carrier detect, CD) circuit is OFF.

Full duplex operation. In full duplex operation, there are separate communication channels for each direction of data transmission so that both the DTEs may transmit and receive simultaneously. The circuit assurance phase is exactly the same in half duplex transmission mode except that both the DTEs can independently assert RTS. In this case, the receivers always remain connected to the receiving side of the communication channel.

Data transfer phase. Once the circuit assurance phase is over, data exchange between DTEs can start. The following circuits are in ON state during this phase.

Transmitting end

- DTR
- DSR
- RTS
- CTS

Receiving end

- DTR
- DSR
- CD

At the transmitting end, the DTE sends data on transmitted data circuit to the DCE which sends a modulated carrier on the transmission medium. The distant end DCE demodulates the carrier and hands over the data to the DTE on received data circuit.

In the half duplex operation, the direction of transmission needs to be reversed every time a DTE completes its transmission and the other DTE wants to transmit. The RTS signal is withdrawn after the transmitting end DTE completes its transmission. The respective DCE withdraws its carrier and switches the communication channel to its receiver. The DCE also inhibits further flow of data from the local DTE by turning off the CTS signal.

When the distant end DCE notices the carrier disappear, it withdraws the Carrier Detect (CD) signal. Noticing that the transmission medium is free, the distant end DTE performs actions of the circuit assurance phase and then transmits data. Thus, a DTE wanting to transmit, checks each time if the channel is free by sensing CD and if it is OFF, it asserts the RTS.

Disconnect phase. After the data transfer phase, disconnection of the transmission media is initiated by a DTE. It withdraws DTR signal. The DCE disconnects from the transmission media and turns off the DSR signal.

7.8 COMMON CONFIGURATIONS OF EIA-232-D INTERFACE

Not all the circuits defined in EIA-232-D specifications are always implemented. Depending on application and communication configuration only a subset of the circuits is implemented. Figure 7.19 shows the circuits commonly implemented in a standard full duplex configuration.

Standard full duplex configuration implementation as shown in Figure 7.19 is required for communication involving modems and telephone network. In practice, however, the following non-standard configurations are also quite often

used.

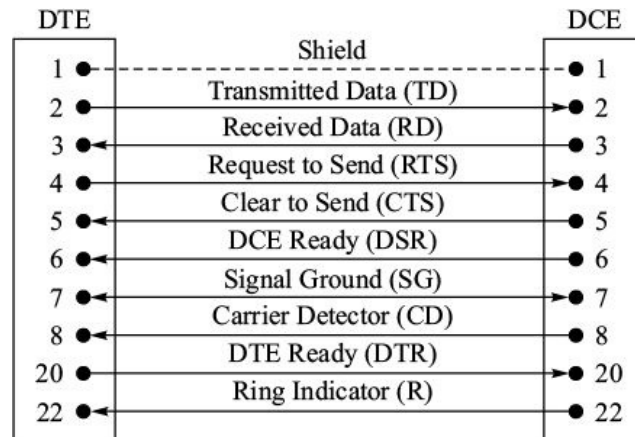


FIGURE 7.19 Commonly implemented circuits in a standard full duplex configuration.

7.8.1 Three-Wire Interconnection

Figure 7.20 depicts a three-wire interconnection which is quite adequate for many interfacing configurations. This interconnection provides a bare minimum number of circuits necessary for full duplex communication. The circuits present are Transmitted Data (TD), Received Data (RD), and Signal Ground (SG). The DTE and DCE must always be in data transfer phase if this configuration is implemented.

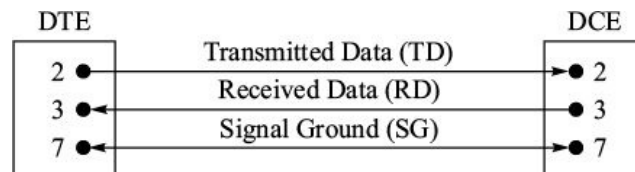


Figure 7.20 Three-wire interconnection for full duplex operation.

7.8.2 Three-Wire Interconnection with Loopback

If Request to Send (RTS) and Clear to Send (CTS) circuits are implemented in a DTE port, the three-wire interconnection shown in Figure 7.20 does not work because the DTE will not transmit data unless it receives the CTS signal. A three-wire interconnection with loopback overcomes this problem (Figure 7.21) by locally generating the signals required for initiating the transmission. The following jumpers are provided:

- RTS circuit is jumpered to CTS and CD circuits
- DTR circuit is jumpered to DSR circuit.

By jumpering the DTR circuit to DSR circuit, the equipment readiness phase is completed as soon as the DTE asserts the DTR signal. Quite often, this occurs when power is applied to the DTE. When the DTE asserts the RTS signal, the circuit assurance phase is immediately completed because it receives immediately the CTS and CD signals.

By providing the loopbacks, the number of interconnecting wires is reduced but it should be kept in mind that certain features of EIA-232-D interface have also been compromised. There are many other configurations, each tailored to a particular requirement and with its own merits and limitations.

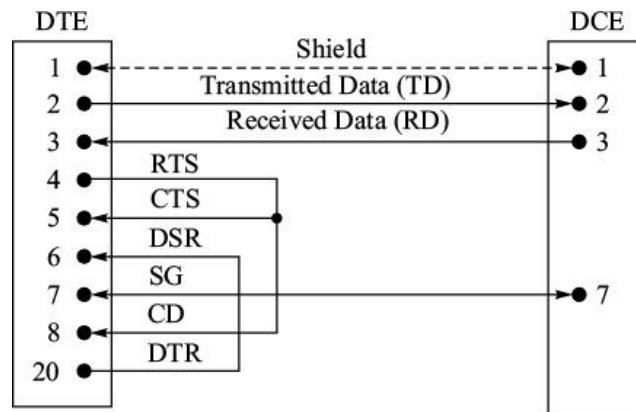


Figure 7.21 Three-wire interconnection with loopbacks.

7.9 NULL MODEM

If we view the EIA-232-D interface standing between the DTE and the DCE, it is seen that a signal which comes out of a particular pin of the DTE port goes towards the DCE on the same pin. In other words, in any pair of corresponding pins of the DTE and DCE ports, one is output pin and the other is input pin. The DCE and DTE ports are, thus, straight connected (Figure 7.22a). *Straight cable* is the one that makes connections between corresponding pins of the two ports, *e.g.* pin 2 of DTE port is connected to pin 2 of DCE port. In other words, if we intend to connect two digital devices having EIA-232-D interfaces using a straight cable, one

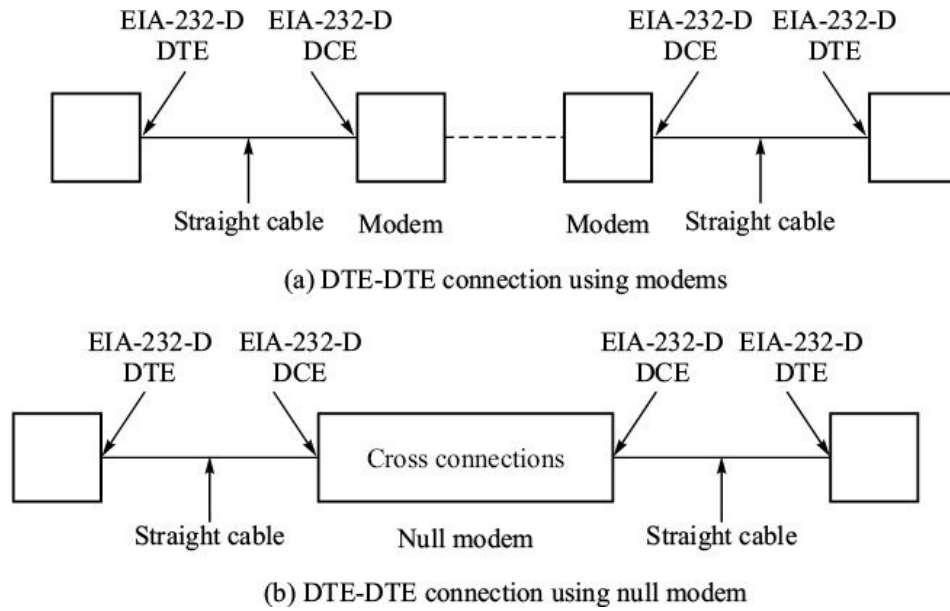


FIGURE 7.22 Null modem.

of the devices should have DCE port and the other device should have DTE port. When we use modems, such is the situation always.

There are situations when we don't need to use modems, *e.g.* when a terminal and the host computer are in vicinity (less than fifteen metres apart), and both the devices to be interconnected have DTE ports. In such situations, we need to introduce an intermediary dummy device which has DCE ports on both sides that face DTEs (Figure 7.22b). The device merely contains cross over connections so that the signals go to the appropriate pins. For example, TD signal available on pin 2 of one DTE is terminated on to RD (Pin3) of the other DTE. The device is called *null modem* because it may not contain any electronics. We will examine a typical crossover configuration of a null modem next.

7.9.1 Null Modem with Loopback and Multiple Crossovers

Figure 7.23 shows a null modem cable having the following jumpers and crossovers:

- Jumpers from
 - Request to Send (RTS) to Clear to Send (CTS)
 - Ring Indicator (RI) to DCE Ready (DSR).
- Crossovers between
 - Transmitted Data (TD) and Received Data (RD)

- Request to Send (RTS) and Carrier Detector (CD)
- Data Terminal Ready (DTR) and Ring Indicator (RI).

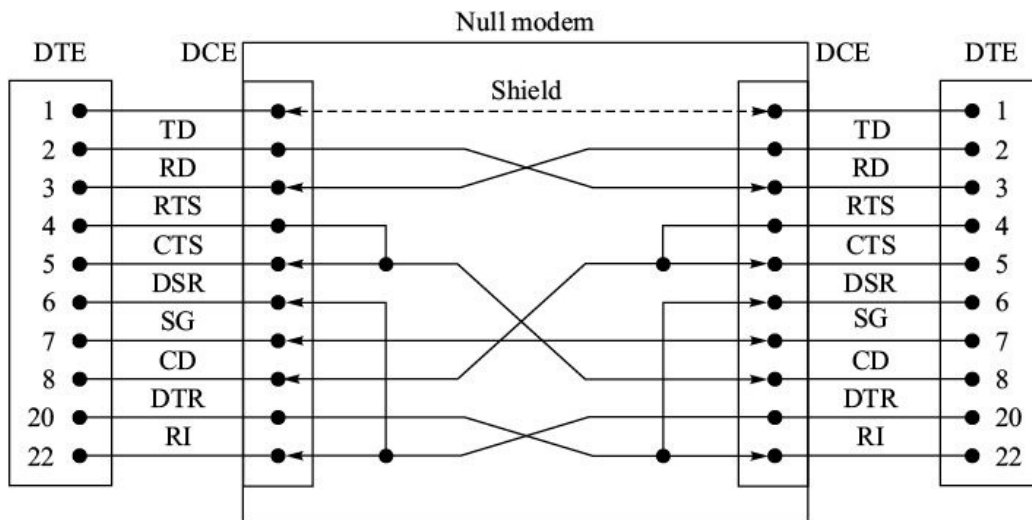


Figure 7.23 Null modem cable.

When a DTE asserts a DTR signal, the other DTE is immediately given a stimulus, the RI, to believe that it has an incoming call. It responds with its DTR which results in the DSR signal at the calling DTE. Thus, the equipment readiness phase is completed. Before transmitting data, the calling DTE asserts the RTS which raises the CD at the other DTE. The RTS signal is looped back at the calling DTE end as CTS. Therefore, the circuit assurance phase is also immediately completed and data transmission can begin.

Note that the crossovers and jumpers can be made within the DB-25 female connectors that connect to the DTE ports. Therefore, a separate box as null modem is not required. Such a cable is called *null modem cable*.

Null modem for synchronous operation. The above discussion applies to the asynchronous mode of operation because we have not considered the clock. If the terminal devices require external clock, the null modem cable will not serve the purpose. A synchronous null modem device with a clock source is required. Else, the internal clock of a DTE can serve the purpose. This clock which is available on pin 24, is wired to pin 17 locally for receive timing, and to pins 15 and 17 of the other device for transmit and receive timings.

7.10 LIMITATIONS OF EIA-232-D

Although EIA-232-D is the most popular physical layer interface, its use in computer networking is limited to low data rates and short distance data transmission applications. The distance between a DTE and DCE is limited to 15 metres beyond which modems are necessary. Even a small industrial plant or an office requires modems between the host and its terminals. As regards the data rate, the EIA-232-D interface has upper limit of 20 kbps which is inadequate for computer networking.

The above limitations of the EIA-232-D interface are due to the following two reasons:

- Unbalanced transmission mode of its signals
- Shared common ground for all signals flowing in both the directions.

Raised ground potential and crosstalk due to these factors result in introduction of errors at higher bit rates and for longer separation between the DTE and the DCE.

7.11 EIA-449 INTERFACE

In the early 1970s, the EIA introduced EIA-422-A, EIA-423-A and EIA-449 interfaces to overcome the limitations of RS-232-C, the earlier version of EIA-232-D. EIA-422-A and EIA-423-A define the electrical specifications. EIA-449 defines the mechanical, functional, and procedural specifications. These specifications are compatible with EIA-232-D so that a device having EIA-232-D interface can be interconnected to a device having the EIA-449 interface. ITU-T also adopted EIA-422-A, EIA-423-A, and EIA-449 subsequently and published recommendations V.10, V.11, and V.54. In the following sections we shall briefly discuss mechanical, electrical, and functional specifications of these standards. Procedural specifications are the same as in EIA-232-D and, therefore, are not described again.

7.11.1 Mechanical Specifications

Since EIA-449 incorporates more than 25 signals, two connectors, one with 37 pins and the other with 9 pins have been specified. Mechanical design of the connectors is as per ISO 4902 standard. All signals associated with the basic operation of the interface appear on the 37-pin connector. The secondary channel

circuits are grouped on the 9-pin connector. Table 7.3 gives a list of the signals defined in the EIA-449 interface. For purposes of comparison, we have included the corresponding EIA-232-D signals also in Table 7.3.

TABLE 7.3 EIA-449 Interface Circuits

37 Pin connector

EIA-449 Circuit name	Pin no.	Direction	EIA-232-D Circuit name
		DTE... DCE	
Shield	1	Common	Shield
Signal Ground (SG)	19	Common	Signal Ground (SG)
Send Common (SC)	37	—	
Receive Common (RC)	20	—	
Terminal in Service (TS)	28	—	
Incoming Call (IC)	15		Ring Indicator (RI)
Terminal Ready (TR)	12, 30		Data Terminal Ready (DTR)
Data Mode (DM)	11, 29		DCE Ready (DSR)
Send Data (SD)	4, 22		Transmitted Data (TD)
Receive Data (RD)	6, 24		Received Data (RD)

Terminal Timing (TT)	17, 35	Transmitter Signal Element Timing(DTE)
Send Timing (ST)	5, 23	Transmitter Signal Element Timing (DCE)
Receive Timing (RT)	8, 26	Receiver Signal Element Timing (DCE)
Request to Send (RS)	7, 25	Request to Send (RTS)
Clear to Send (CS)	9, 27	Clear to Send (CTS)
Receive Ready (RR)	13, 31	Carrier Detector (CD)
Signal Quality (SQ)	33	Signal Quality Detector
New Signal (NS)	34	—
Select Frequency (SF)	16	—
Signal Rate Selector (SR)	16	Data Signal Rate Selector (DTE)
Signal Rate Indication (SI)	2	Data Signal Rate Selector (DCE)
Local Loopback (LL)	10	Local Loopback

Remote Loopback (RL)	14		Remote Loopback
Test Mode (TM)	18		Test Mode
Select Standby (SS)	32		—
Standby Indicator (SB)	36		—
Spare	3, 21		—
9 Pin Connector			
Shield	1	Common	—
Signal Ground	5	Common	Signal ground
Send Common	9		—
Receive Common	6		—
Sec. Send Data	3		Sec. Transmitted Data
Sec. Received Data	4		Sec. Received Data
Sec. Request to Send	7		Sec. Request to Send
Sec. Clear to Send	8		Sec. Clear to Send

The EIA-449 standard also specifies the maximum cable length and the corresponding data rate supported by the cable. Figure 7.24 shows this relationship graphically.

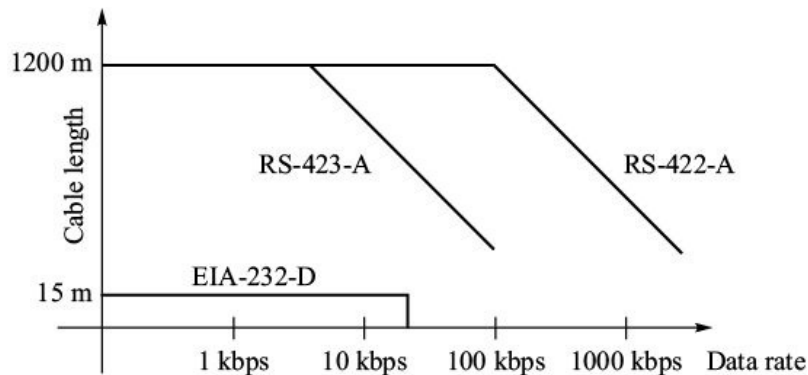


FIGURE 7.24 Data rates supported by EIA-449.

7.11.2 Electrical Specifications

Electrical specifications for EIA-449 interface are defined in EIA-422A and EIA-423-A. EIA-422-A specifies electrical characteristics of the balanced circuits while EIA-423-A specifies electrical characteristics of the unbalanced circuits. Circuits of EIA-449 are divided into two categories. Category I circuits are as follows:

- Send Data (SD)
- Receive Data (RD)
- Terminal Timing (TT)
- Send Timing (ST)
- Receive Timing (RT)
- Request to Send (RS)
- Clear to Send (CS)
- Receive Ready (RR)
- Terminal Ready (TR)
- Data Mode (DM).

The rest of the circuits belong to Category II. Category I circuits are implemented using either EIA-422-A or EIA-423-A for data rates of less than 20 kbps. For higher data rates, balanced EIA-422-A electrical characteristics is

used. Circuits belonging to Category II are always implemented using EIA-423-A characteristics.

Electrical characteristics (EIA-422-A). EIA-422-A specifies electrical characteristics of the balanced circuits. The corresponding ITU-T recommendation is V.11. Figure 7.25a shows the voltage levels corresponding to the two binary states of the electrical signals. Note that the transition region between binary states 1 and 0 on the receiving side is much narrower because of the better crosstalk performance and elimination of earth potential problem. Line code used is NRZ-L.

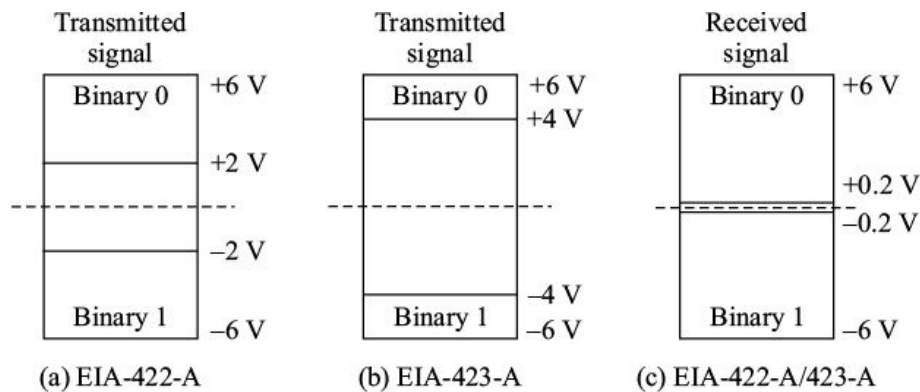


Figure 7.25 EIA-422-A and EIA-423-A electrical specifications.

Electrical characteristics (EIA-423-A). EIA-423-A specifies unbalanced transmission as in EIA-232-D interface but with separate ground return wires for the two directions of transmission. The corresponding ITU-T recommendation is V.10. As with EIA-232-D, NRZ-L line encoding is used. Figure 7.25b depicts the relationship between the electrical voltages and the binary states. The receive side specifications are same as in case of EIA-422-A.

7.11.3 Functional Specifications

Table 7.3 gives complete listing of the EIA-449 circuits and corresponding EIA-232-D circuits where applicable. There are several new signals defined in the EIA-449 standard. Their functions are briefly described below.

Send common (SC). This is the common return path for all unbalanced circuits from DTE to DCE.

Receive common (RC). This is the common return path for all unbalanced circuits from DCE to DTE.

Terminal in service (TS). This circuit indicates to the DCE whether or not the DTE is operational.

New signal (NS). In multipoint polling applications, NS circuit is used by the control DTE (host) to indicate to its DCE the end of a message from one of the remote DTEs, and to alert the DCE to get ready for receiving message from a new DTE.

Select frequency (SF). A DTE selects transmit frequency of a DCE on the communication channel by signaling on the SF circuit.

Select standby (SS). This circuit is used by a DTE to select the standby equipment.

Standby indicator (SI). This circuit is activated in response to SS to indicate whether standby or normal equipment is in use.

7.12 EIA-530

EIA-449 was not readily accepted by the industry because it specified 37-pin connector. 37-pin connector was bigger and investments had already been made in 25-pin connector. EIA-530 was introduced in 1987. It uses the same DB-25, 25-pin connector as specified in EIA-232-D. Electrical specifications are as per EIA-422-A and EIA-423-A. To accommodate the required essential circuits on 25 pins, ring indicator, signal quality detector, data signal rate selector, and the secondary channel circuits are not included in this interface.

7.13 ITU-T X.21 RECOMMENDATION

ITU-T X.21 recommendation is for the interface between a DTE and a DCE for circuit-switched data networks. DCE in this case is a Circuit Switched Public Data Network (CSPDN) node. CSPDN is equivalent to PSTN but it is used for transporting data over switched connections. X.21 specifies protocols for the lowest three layers, namely, the network, data link, and physical layers. The X.25 interface for packet-switched data networks also specifies use of the X.21 physical layer interface for its physical layer. The physical layer of X.21 specifies the protocol and physical interface for synchronous transmission. Its mechanical, electrical, functional, and procedural specifications are given below

in brief.

7.13.1 Mechanical Specifications

X.21 specifies a 15-pin connector (DB-15), which is similar to DB-25 connector shown in Figure 7.12. The mechanical design and the pin assignments are as per the ISO 4903 standard.

7.13.2 Electrical Specifications

The electrical specifications are as per V.10 for unbalanced transmission for data rates up to 9600 bps and V.11 for balanced transmission for rates above 9600 bps. The interface can operate at data rates from 600 to 48000 bps.

7.13.3 Functional Specifications

The important signaling circuits used in X.21 between the DTE and DCE are as under.

Table 7.4 gives X.21 pin allocation of the above circuits for balanced operation.

Pin no.	Direction		Circuit name
	DTE	DCE	
1	Common	Shield	
2,9			Transmit (T)
3,10			Control (C)
4,11			Receive (R)
5,12			Indication (I)
6,13			Signal element timing (S)

7,14		Byte timing (B)
8	Common	Signal ground (G)
15		Reserved

Transmit (T). This circuit is used by the DTE for sending to the DCE, data signals and call control signals during call establishment and call clearing phases.

Receive (R). This circuit is used by the DCE for sending to the DTE, data signals and call control signals sent during call establishment and clearing phases.

Signal element timing (S). This circuit is used by the DTE to transmit the DTE clock to the DCE.

Control (C). This circuit is used by the DTE to indicate condition of the Transmit (T) circuit. It is ON when data is being transmitted by the DTE.

Indication (I). This circuit is used by the DCE to indicate condition of the Receive (R) circuit. It is ON when data is being sent by the DCE to the DTE.

Byte timing (B). This signal is transmitted by the DCE for byte synchronization at the DTE.

Signal ground (G). This is the common ground for the signals when the transmission is as per V.28 recommendation.

DTE common return (Ga). This is the common return for circuits from the DTE and is used by the receivers in the DCE when the transmission is unbalanced as per V.10.

DCE common return (Gb). This is the common return for circuits from the DCE and is used by the receivers in the DTE when the transmission is unbalanced as per V.10.

7.13.4 Procedural Specifications

Unlike the EIA-232-D interface where each control signal has a separate circuit, X.21 defines a sequence of signal combinations and states of the interface for

transfer of data. A typical procedure for establishing and clearing an X.21 connection is summarized in Table 7.5.

TABLE 7.5 Typical X.21 Procedure

DTE		Description of interface states	DCE	
T	C		R	I
1	OFF	Ready	1	OFF
0	ON	Call request	1	OFF
0	ON	Proceed to select	+++...	OFF
Address	ON	DTE transmits address	+++...	OFF
1	ON	Connection establishment in progress	Call progress signals	ON
1	ON	Connection established, ready for data	1	ON
Data	ON	Data transfer	Data	X
0	OFF	Clear request by DTE	X	OFF
0	OFF	Clear confirmation by DCE	0	OFF

7.13.5 X.21bis Recommendation

As most of the commercially available terminal devices do not conform to the X.21 interface, X.21bis has been defined by ITU-T to connect existing DTEs having the V.24 interface. X.21bis is a V.24 compatible interface with some additional signaling procedures for working with switched data networks. The DTE can have the following mechanical and electrical interfaces together with the V.24 interface.

- V.28 with 25-pin connector and pin allocation as per ISO 2110.
- V.10 with 37-pin connector and pin allocation as per ISO 4902.
- V.11 with 37-pin connector and pin allocation as per ISO 4902.

SUMMARY

The physical layer is responsible for transport of bits from one device to the other on the physical connection. It converts the bits into electrical signals having characteristics suitable for transmission over the physical medium. It also supports the relaying function in the transmission medium. The physical medium interface is defined in terms of its mechanical, electrical, functional, and procedural specifications. The procedural specification is the definition of the physical layer protocol. EIA-232-D is the most popular physical medium interface but it has limitations of data rate and distance. The distance between a DTE and DCE is limited to 15 metres. The maximum data rate of the EIA-232-D interface has upper limit of 20 kbps which is inadequate for computer

networking. Therefore, other physical interfaces (EIA-422-A, EIA-423-A, EIA-429) that overcome these limitations have been defined.

ITU-T X.21 recommendation is for the interface between a DTE and a DCE for circuit-switched data networks. DCE in this case is a circuit switched public data network (CSPDN) node. The physical layer of X.21 specifies the protocol and physical interface for synchronous transmission. It supports data rates from 600 to 48000 bps.

EXERCISES

1. The physical service is a non-confirmed service. If some data bits are lost during transmission over the interconnecting media, which layer will detect their loss and take recovery action?
2. Trellis-coded modem carries out forward error correction and improves the error performance of the transmission media. This function is carried out at the physical layer in the modems. Show by a layered model how the end systems are not involved in the error correction process.
3. In Figure 7.3, when point-to-multipoint physical connection is activated, there is possibility of intermixing of electrical signals transmitted by different end systems in the shared media. Can the physical layer avoid such mixing of signals?
4. In Figure E7.26, statistical multiplexers have the first two layers of the OSI reference model. Draw the layered model showing the physical connections which exist in this configuration.



FIGURE E7.26.

5. In Figure 7.1 of this chapter, devices A and B have half duplex physical connection between them. Indicate the service primitives when:
 - ◆ A activates the connection,
 - ◆ A sends one data unit,
 - ◆ A deactivates the connection,
 - ◆ B activates the connection,
 - ◆ B sends one data unit,
 - ◆ B deactivates the connection.

6. State whether true or false:
- (a) In half duplex transmission, the physical layers decide who will transmit.
(True/False)
 - (b) Activation and deactivation of the physical connection is a confirmed service.
(True/False)
 - (c) The protocol control information (PCI) at the physical layer is usually sent on separate wires.
(True/False)
 - (d) If modems are used for relaying electrical signals, the physical layer protocol is between the end system and the modem.
(True/False)
 - (e) The physical layer just converts bits into electrical signals and vice versa. Clock and signal encoding are data link layer functions.
(True/False)
7. If the received line signal detector circuit becomes ON, which circuit at the other end has been raised?
8. If the modem is OFF, which circuit will not be ON on the EIA-232-D interface?
9. A DTE using the EIA-232-D interface sends the ASCII character “K” with odd parity. The mode of transmission is asynchronous. Sketch the transmitted data signal assuming 15 V logic circuits and stop pulse of 1-bit duration.
10. At times, the DTE and DCE ports do not have the specified gender of the connector. It becomes impossible to identify the type of port by its physical attributes. Which are the pins likely to indicate the type of port when the device is energized?
11. If DTE-A is transmitting data to DTE-B over a half duplex transmission line with modems at either end, list the events which takes place when B starts transmitting. The interface between the DTEs and modems is EIA-232-D.
12. If two DTEs having EIA-232-D interface are connected through modems and have full duplex transmission between them, indicate the ON/OFF status of the following circuits at both the ends:

- ◆ Data Terminal Ready
- ◆ DCE Ready
- ◆ Request to Send
- ◆ Clear to Send
- ◆ Received Line Signal Detector.

1 DSR: Data set (modern DCE) ready.

8

The Data Link Layer

The basic service provided by the physical layer is transportation of bits over the physical connection. This service is unreliable in the sense that disturbed line conditions of the transmission medium may introduce errors which are not taken care of by the physical layer. Error control and other associated functions are carried out by the second layer, the *data link layer* of the OSI reference model.

We begin this chapter, with a description of the purpose and functions of the data link layer. After a brief discussion on the data link service, we move over to frame design considerations. Error control and flow control functions using stop-and-wait and sliding window mechanisms are discussed next. We also examine link utilization efficiency of these mechanisms in absence and presence of errors. Application environment of data link protocols in the networking industry is presented at the end of the chapter.

8.1 NEED FOR DATA LINK CONTROL

Let us consider a situation, as shown in Figure 8.1, where two digital devices A and B need to exchange information. These devices could be computers, network nodes or other data terminal equipment.

To exchange digital information between devices A and B we require an interconnecting transmission medium to carry the electrical signals (e.g. copper wire), a standard interface (e.g. EIA-232-D), and the physical layer to convert bits into electrical signals and vice versa.

Together with the transmission medium, physical layer of the devices provide the capability for transparent exchange of bits over the physical connection but this capability has certain limitations:

- If the electrical signal gets impaired due to the noise encountered during

transmission or due to the medium characteristics, error may be introduced in the data bits. There is need to establish mechanisms to control transmission errors.

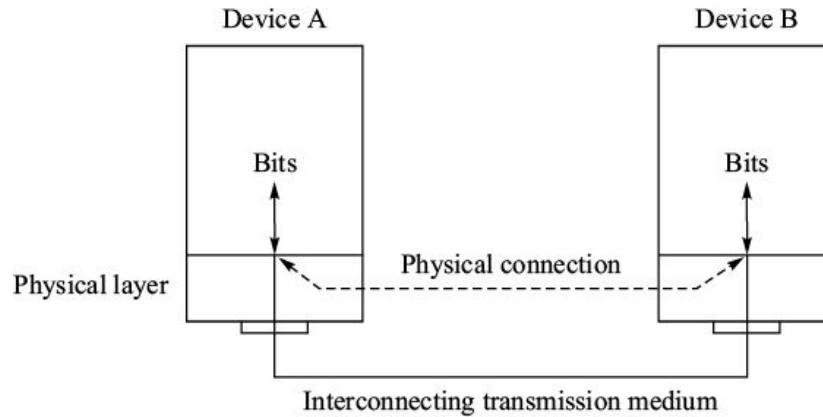


FIGURE 8.1 Exchange of bits over physical connection.

- Errors can also be introduced if the receiving device is not ready for the incoming bits and some of the bits are lost. Therefore, a data flow control mechanism also needs to be implemented.

The physical layer does not meet these requirements. Error and flow control functions are implemented in the data link layer which ensures error-free transfer of bits from one device to the other.

8.2 DATA LINK LAYER

Data link layer constitutes the second layer of the hierarchical OSI model. Data link layer together with the physical layer and the interconnecting medium provide a data link connection for reliable transfer of data bits over an imperfect physical connection (Figure 8.2).

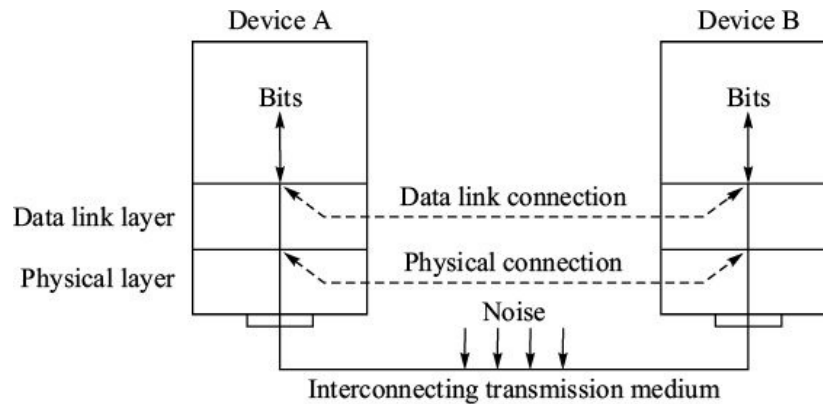


FIGURE 8.2 Reliable transfer of bits over data link connection.

Data link layer incorporates certain processes which carry out error control, flow control, and the associated link management functions. It receives the data to be sent to the other device from the next higher layer and adds some control bits to a block of data bits. The data block along with the control bits is called a *frame*. The frame is handed over to the physical layer. The physical layer converts bits of the frame into an electrical signal for transmission over the interconnecting transmission medium (Figure 8.3).

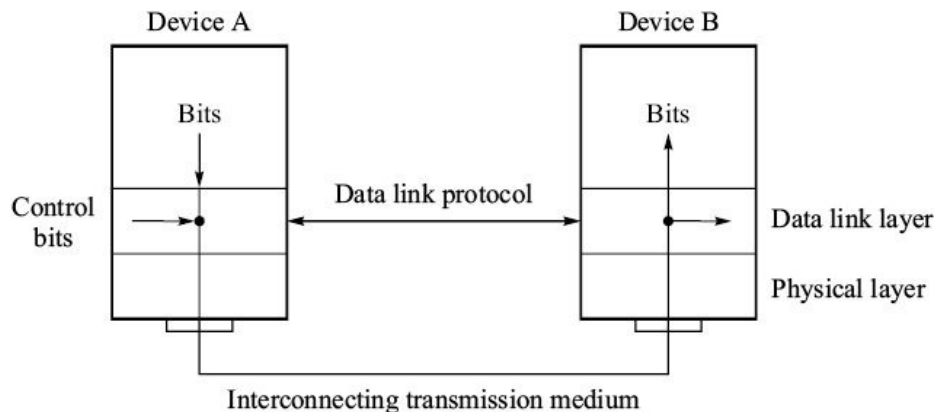


FIGURE 8.3 Formation of frames in data link layer.

At the receiving end, the incoming electrical signal is converted back to bits by the physical layer and the frame is handed over to the data link layer. The data link layer removes the control bits and checks for errors. If there is no error, it hands over the received data bits to the next layer.

The control bits include error-check bits, addresses, sequence numbers, *etc.* These additional bits usually constitute more than one field in a frame and enable error control, flow control, and link management.

8.2.1 Service Provided by the Data Link Layer Data link layer

receives service from the physical layer and provides service to the network layer which is the user of these services. In OSI terminology, the user data unit received from the network layer is called a Data Link Service Data Unit (DL-SDU) and the frame formed by adding control bits to the DL-SDU is called a Data Link Protocol Data Unit (DL-PDU) as shown in Figure 8.4. The control bits constitute a header and a trailer.

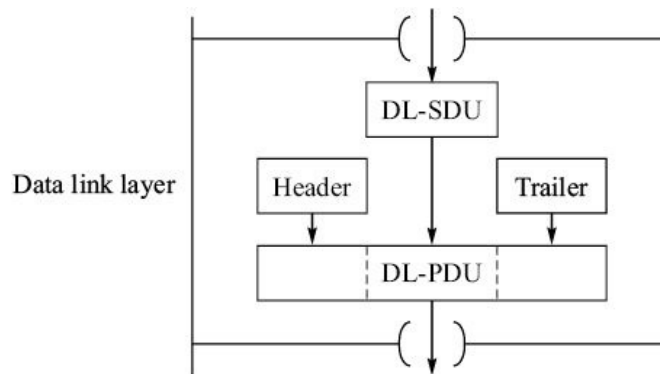


FIGURE 8.4 Data link service data units and protocol data units.

The basic service provided by the data link layer to the network layer is reliable transfer of DL-SDUs over the data link connection which is established, maintained, and released by the data link layer on the request of the network layer. This basic service has the following associated features.

Sequencing. The sequence integrity of the DL-SDUs is maintained.

Error notification. If the data link layer detects an unrecoverable error, it notifies the network layer.

Flow control. The network layer can control the rate at which it receives the DL-SDUs from the data link layer. This control may be reflected in the rate at which the data link layer will accept DL-SDUs at the other end.

Quality of service parameters. The data link layer provides selectable quality of service parameters which include residual error rate, transit delay throughput, *etc.* The selected quality of service is maintained during data link connection.

The data link service, described above, is connection-mode service but it can be connectionless-mode service also. The service primitives for connection-mode and connectionless-mode are given in Appendix A at the end of the

chapter. Connectionless-mode data link service is described in Chapter 10. The complete definition of data link service is given in ISO 8886. The corresponding ITU-T Recommendation is X.212.

8.2.2 Data Link Protocols

It is essential that the structure of the frame is known to both the data link layers so that control bits can be identified. The data link layers should also agree on the set of procedures to be adopted for exchange of control information. The specified set of rules and procedures for carrying out data link control functions is called *data link protocol*. A data link protocol specifies the following:

- Format of the frame, *i.e.* locations and sizes of the various fields.
- Contents of various fields.
- Sequence of messages to be exchanged to carry out the error control, flow control, and data link management functions.

There are many data link protocols developed by various manufacturers and organizations. While all the protocols broadly satisfy the basic functional requirements of the data link layer, the frame formats and contents of various fields are very specific to each protocol. Examples of data link protocols are:

- Binary Synchronous Data Link Control (BISYNC, BSC)
- Synchronous Data Link Control (SDLC)
- High-level Data Link Control (HDLC)
- Advanced Data Communication Control Procedure (ADCCP)
- Point-to-Point Protocol (PPP)

We will discuss BISYNC and HDLC protocols in the next chapter. PPP is discussed in Chapter 17, Internet Protocol.

8.3 FRAME DESIGN CONSIDERATIONS

The first and foremost task of the data link layer is to format the user data as series of frames each having a predefined structure. A frame contains user data and control fields. Each frame is processed as one entity for error and flow control, *i.e.* if an error is detected, the whole frame is retransmitted.

The format of a frame, in general, consists of three components as shown in Figure 8.5.

- Header
- Data
- Trailer.

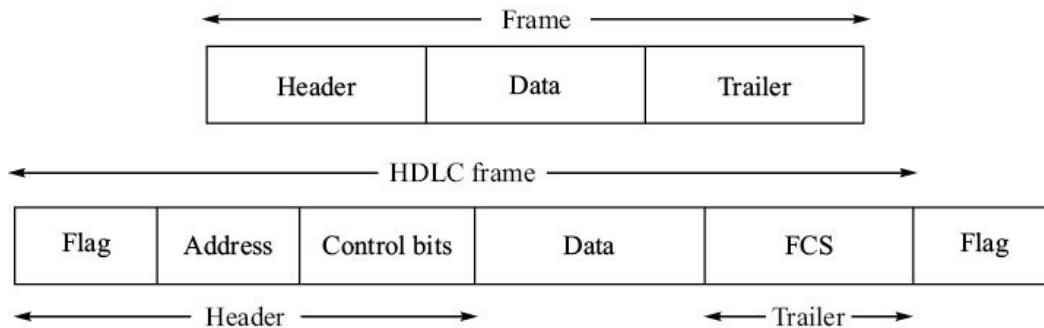


FIGURE 8.5 Data link layer frame formats.

The header and the trailer consist of one or more fields containing data link protocol control information. A typical example of the composition of a frame used for carrying user data in HDLC protocol is shown in Figure 8.5. The frame starts with a flag to identify the start of a frame. The flag is followed by an address field. The control field contains the sequence number of the frame, acknowledgement of the receipt of a frame or other control information. The data field contains the user data received from the network layer. The frame check sequence (FCS) contains CRC bits for error detection. Each HDLC frame is always followed by a flag which acts as a frame delimiter.

We will refer to the frames containing user data as data frames. There can be frames that contain only the protocol control information, *i.e.* there is no user data. There are several types of such frames. One example of such frames is the acknowledgement frame. A receiver sends the acknowledgement frame on receipt of a data frame.

8.3.1 Types of Frame Formats

The frame format is so designed that the receiver is always able to locate the beginning of a frame and its various fields, and is able to separate the data field. To identify a frame and its various fields, field identifiers/delimiters are incorporated in it. These are unique symbols which indicate by their presence the beginning and end of a frame or a specific field. For example, the flags in an HDLC frame indicate the start and end of the frame.

The requirement of field identifiers and delimiters is determined by the frame structure. A data link protocol may adopt fixed or variable formats of the frames.

In fixed format, all the fields are always present in all the frames. In variable format, the presence of any field is optional. The length of a frame may also be fixed or variable. Thus, there are several possibilities:

- Variable format-variable length
- Fixed format-fixed length
- Fixed format-variable length.

A variable format-fixed length frame is not possible and is never used. Let us consider a simple case of a frame having only three fields and try to determine the requirement of the delimiters and identifiers for the above options (Figure 8.6).

Variable format-variable length. Figure 8.6a shows the frame format. All the fields are optional and if a field is present, its size is variable. In all, five delimiters/identifiers are required. *X* is the frame start identifier. The presence of each field is indicated by a field identifier which also acts as delimiter for the previous field. As the size of frame is variable, an end delimiter *Y* is required to indicate the end of the frame.

Fixed format-fixed length. In this case the format of the frame is decided once for all and the field sizes are also fixed in all the frames. The frame format is shown in Figure 8.6b. Only one identifier is required at the beginning of the frame. On receipt of the identifier, the receiver is able to identify all the fields as the format of the frame and the sizes of the various fields are known to the receiver in advance.

Fixed format-variable length. In this case frame start and end identifier/delimiters are required (Figure 8.6c). The identifier for the first field is not required as the frame identifier also identifies the field. As regards other fields, field delimiters are required for each field.



(a) Variable format and variable length



(b) Fixed format and fixed length

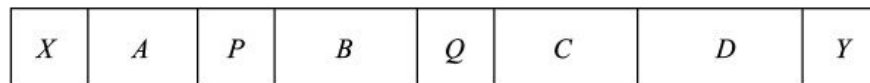


(c) Fixed format and variable length

Figure 8.6 Types of frame structures.

EXAMPLE 8.1 Consider a frame consisting of four fields, *A*, *B*, *C*, and *D*. Field *A* is always present and is of fixed length. Fields *B* and *C* are of variable length and optional. Field *D* is fixed and always present. What are the identifiers and delimiters required?

Solution



X : Frame identifier *P* : Identifier for field *B*
Y : Frame delimiter *Q* : Identifier for field *C*

1. Frame identifier (*X*) is required.
2. Field *A* is fixed and always present, therefore can be located without an identifier/delimiter.
3. Fields *B* and *C* require identifiers and delimiters. They can be delimited by the following identifier.
4. Field *D* can be located by the frame delimiter (*Y*) by counting back the bits.

8.3.2 Transparency

Transparency refers to providing a service to the users wherein no restriction is placed on the contents of the user data. Since any bit pattern can be sent by the user, problems may arise if the data field contains bit patterns similar to the field identifiers/delimiters. For example, if the data field of the HDLC frame shown in Figure 8.5 contains a bit pattern identical to the flag, the receiver may mistake it for the end flag of the frame. Therefore, the field identifiers and delimiters should not be present in any field apart from their predefined locations in the

frame. Different methods are adopted in various data link protocols to achieve transparency as we shall see in the next chapter where specific protocols are discussed.

8.3.3 Bit-Oriented and Byte-Oriented Data Link Protocols The data link protocols can be categorized as bit-oriented and byte-oriented data link protocols. In a bit-oriented data link protocol, control information is coded at bit level and the length of the data field may not be a multiple of bytes. Bit level implies that a control symbol need not to be one full byte. For example, in HDLC protocol which is a bit-oriented protocol, the first bit of the control field of the HDLC frame indicates the type of frame.

Byte-oriented data link protocols define all the control symbols that are at least one byte long. The size of data field is also a multiple of bytes. BISYNC is a byte-oriented protocol.

8.4 FLOW CONTROL MECHANISMS

Flow control mechanisms are incorporated to ensure that the data link layer at the sending end does not send more frames containing data than what the data link layer at receiving end is capable of handling. Therefore, the receiver is provided with a control to regulate the flow of the incoming frames. This control is in the form of an acknowledgement which is sent by the receiver. The acknowledgement serves two purposes:

- It clears the sender to transmit the next data frame.
- It acknowledges receipt of all previous frames.

There are two methods of flow control:

- Stop-and-wait
- Sliding window.

In stop-and-wait mechanism, the sending end sends one data frame at a time

and waits

for an acknowledgement (ACK) from the receiver. In sliding window flow control mechanism, the sending end continues sending the data frames one after the other without waiting for acknowledgements for individual data frames. Let us examine the operation of these mechanisms in detail.

8.4.1 Stop-and-Wait Flow Control

In *stop-and-wait* flow control, the receiver can temporarily stop flow of data frames by withholding the acknowledgement. Alternatively, it can request for temporary suspension of transmission of the data frames by sending ACK and WAIT (WACK). On receipt of WACK, the sending end waits for ACK to recommence transmission of the next data frame. Note that WACK suspends transmission of data frames only. It is not applicable to control frames. Figure 8.7 illustrates this mechanism.

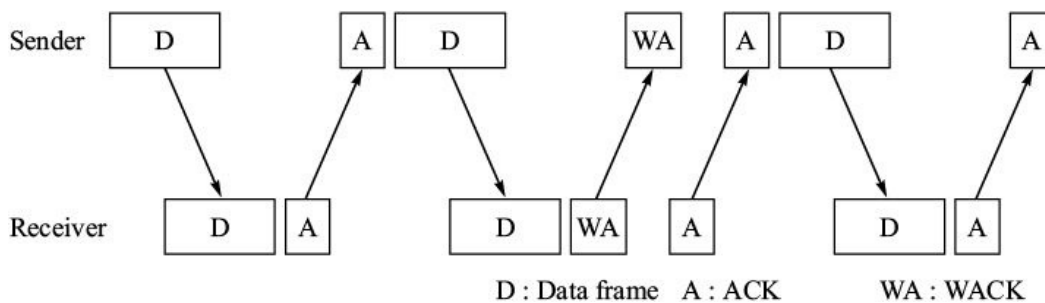


Figure 8.7 Idle RQ (Start-and-stop) flow control mechanism.

Link utilization in stop-and-wait flow control. In the stop-and-wait flow control mechanism, only one data frame is sent at a time and each frame is individually acknowledged. The data frame and the acknowledgement take a certain amount of propagation time to travel across the link. Propagation time can be as large as 270 milliseconds for a satellite communication link or a few milliseconds for a terrestrial link. Large propagation time makes the stop-and-wait mechanism very inefficient from the point of view of link utilization. Let us calculate the link utilization efficiency for the stop-and-wait flow control mechanism.

Figure 8.8 shows the sequence of events and the associated time instants of their occurrence. If average size of a frame is L bits and the data rate is R , sending device (A) shall complete

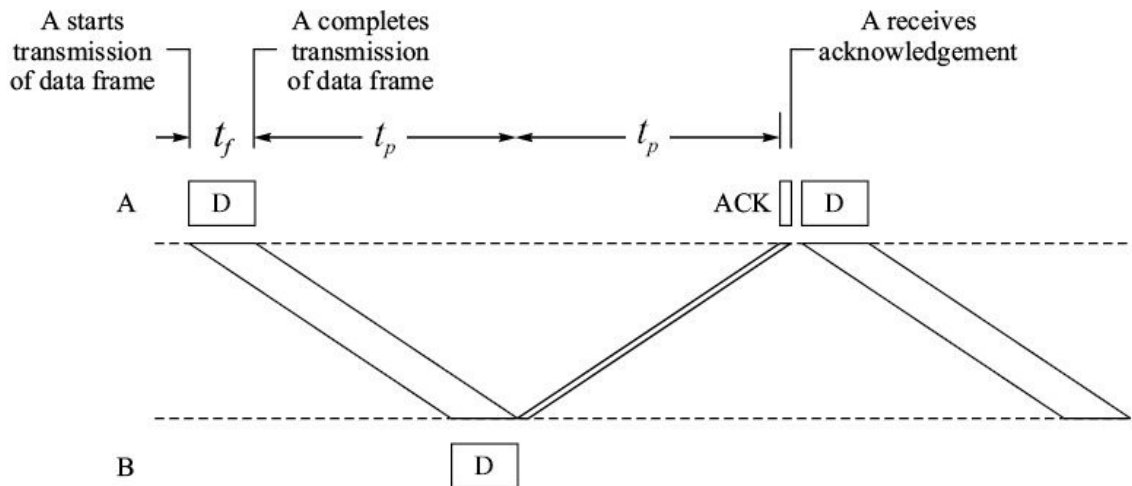


Figure 8.8 Link utilization in stop-and-wait flow control mechanism.

transmission of one frame in time t_f given by $t_f = \frac{L}{R}$

If t_p is the propagation time, the frame is completely received by receiving device (B) after time $t_f + t_p$. To calculate the best possible utilization of the link, let us assume that

- the receiver sends back acknowledgement immediately on receipt of a data frame,
- the size of the acknowledgement frame is very small, and
- there are no errors in the data frame or its acknowledgement.

The sending end (A) will receive the acknowledgement after time $t_f + t_p + t_p$ and can send the next data frame immediately thereafter. Out of the total time $t_f + 2t_p$, A has utilized the link

for time t_f only for transmission of its data frame. Therefore, link utilization

efficiency U is given by $U = \frac{t_f}{t_f + 2t_p} = \frac{1}{1 + 2t_p/t_f} = \frac{1}{1 + 2t_p R/L} = \frac{1}{1 + 2A}$, $A =$

$$\frac{t_p R}{L}$$

Figure 8.9 shows the variation of link utilization (U) with respect to frame size. The link utilization can be improved by keeping large frame size (L). But large frames are more prone to errors and therefore effective link utilization is reduced. We will examine this issue in the next section. Therefore, the stop-and-wait mechanism is suitable only when the propagation time is less than or

comparable to the frame transmission time.

EXAMPLE 8.2 Calculate the maximum link utilization efficiency for stop-and-wait flow control mechanism if the frame size is 2400 bits, bit rate is 4800 bps, and distance between the devices is 2000 km. Speed of propagation over the transmission media can be taken as 200,000 km/s.

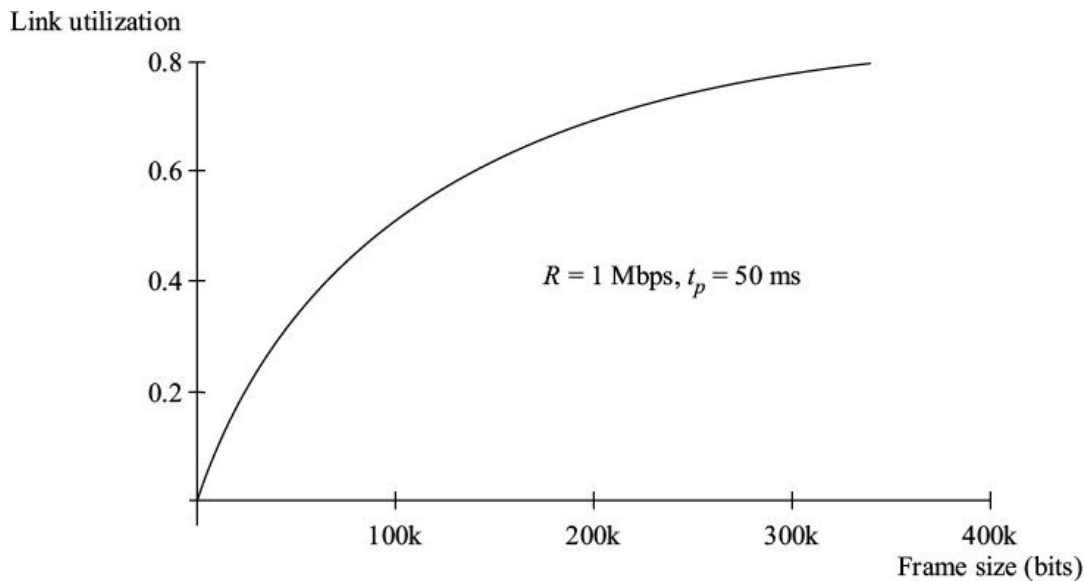


Figure 8.9 Link utilization as a function of frame size in stop-and-wait flow control.

Solution

Frame transmission time (t_f) = 2400/4800 = 0.5 s Propagation time (t_p) = 2000/200000 = 0.01 s $A = 0.01/0.5 = 0.02$

$$U = 1/(1 + 0.04) = 96\%$$

Table 8.1 gives link utilization efficiencies for some typical cases. Speed of propagation in cable media is assumed to be 200,000 km/s. Note that link utilization is very poor in case of a satellite link because the propagation time is very large.

TABLE 8.1 Link Utilization in Stop-and-Wait Mechanism				
Type of media	t_f	t_p	A	U
	0.1	0.00005	~0.0	
	0.1	0.005	0.05	
Local cable link (10 km)				~100%
Coaxial cable link (1000 km)				91%
Satellite link	0.1	0.270	2.70	15.6%

8.4.2 Sliding Window Flow Control

In stop-and-wait flow control mechanism, link utilization efficiency was poor when time to transmit a frame (t_f) was significantly less than the propagation time (t_p). After transmitting a data frame, the sending end remained idle until the acknowledgement was received from the other end. In *sliding window flow control* mechanism, the sending end continues to transmit the next data frame without waiting for acknowledgement for the last frame (Figure 8.10). Compare this figure with Figure 8.8. In this case the sender sends six more data frames before it receives acknowledgement for the first data frame. Therefore, the link does not remain idle and link utilization efficiency is improved.

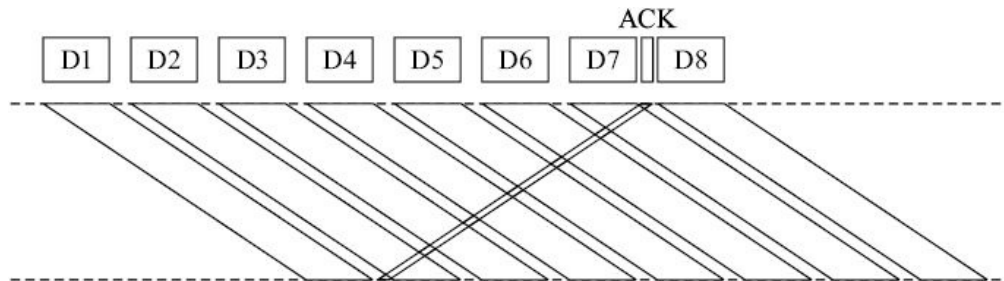


FIGURE 8.10 Link utilization in continuous RQ.

Sliding window flow control is based on the following acknowledgement mechanism:

- Each data frame carries a sequence number for its identification.
- The receiver acknowledges receipt of one or more data frames by sending back a numbered acknowledgement. The acknowledgement frame is written as RR- N (Receive Ready- N). N is the sequence number of the next data frame the receiver expects to receive.
- All previous data frames are assumed acknowledged on receipt of an acknowledgement. Note that all data frames are still acknowledged but may not be individually.

In Figure 8.11, A sends data frames D0 to D5. It receives acknowledgements RR2 and RR5. By sending RR2, the receiver is acknowledging receipt of data frames bearing numbers D0 and D1. RR3 is lost but it does not matter because RR5 acknowledges all previous frames.

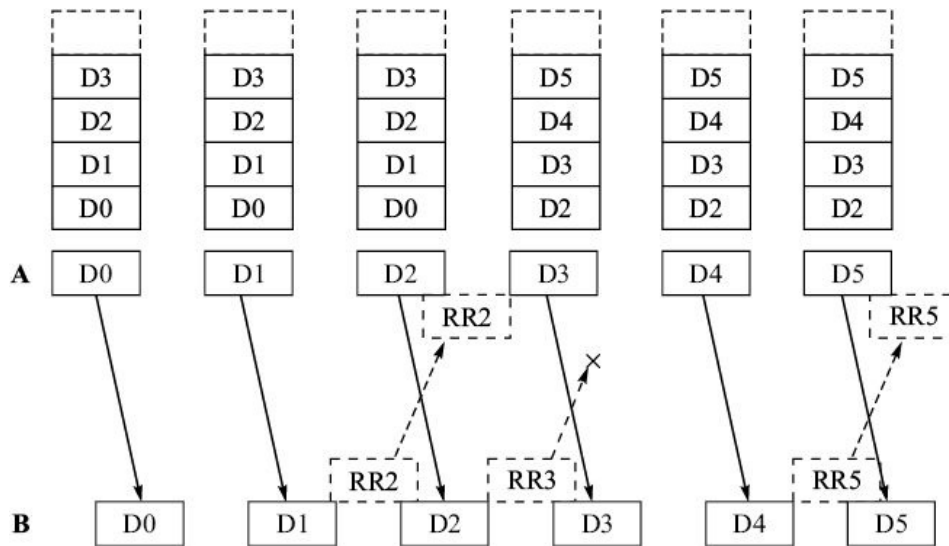


FIGURE 8.11 Flow control in continuous RQ flow control mechanism.

To stop the transmission of data frames temporarily, the receiving end can send another type of acknowledgement, Receive Not Ready (RNR— N). RNR- N is acknowledgement for data frames up to $D(N - 1)$ and a request to stop further transmission temporarily. RNR— N is equivalent to WACK of stop-and-wait mechanism. Transmission of data frames is resumed when RR— N is received from the receiving end. Note that RNR is used for stoppage of transmission of data frames only.

Sliding window. In sliding window flow control, the sender and the receiver need to store the sent data frames temporarily in their respective buffers. The sender needs buffer because it needs to keep copies of all the sent frames for which acknowledgements are yet to be received. The receiver may request for retransmission of a data frame that it received with errors. In Figure 8.11, sending end A keeps copies of D0 and D1 in the memory until it receives RR2.

The buffer size needs to be just sufficient to accommodate as many frames that can be sent in the time required to get the acknowledgement ($t_f + 2t_p$) of a data frame (Figure 8.10). As soon as an acknowledgement for a sent data frame is received, the sender can remove the copy of the frame from its buffer and let the next waiting frame to occupy the buffer.

The receiver stores the received data frames temporarily in its buffer because the frames may be received out of sequence and it must put them in sequence before processing them for retrieval of user data. Data frames are received out of sequence when a frame is lost in transit or received with errors. A frame

received with errors is as good as not received because the error may be in its sequence number. Therefore, the receiver comes to know of a missing frame when it receives next data frame in sequence without any error.

Limited memory availability at sending and receiving ends restricts maximum number of data frames that

- can be sent by the sending end without receiving an acknowledgement from the receiver, and
- can be received out of sequence at the receiving end.

We can consider the sending end buffer as window that contains frames waiting to be transmitted or acknowledged. As the acknowledgements are received, the copies of acknowledged frames are deleted and new frames enter the window and line up for their turn. It is equivalent to sliding the window to accommodate new frames and to expel the acknowledged frames (Figure 8.12).

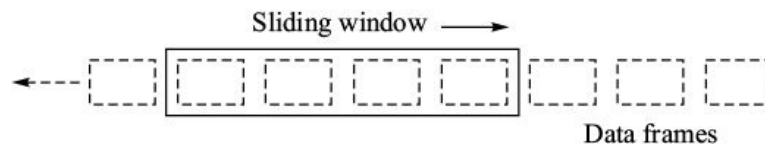


FIGURE 8.12 Sliding window.

Physically the window is defined in terms of three parameters (Figure 8.13).

- Lower window edge (*LWE*),
- Upper window edge (*UWE*), and
- Window size (*W*).

Window at transmitting end. The semantics of window at the transmitting end and the receiving end are somewhat different. At the transmitting end, the window contains

- copies of those data frames that have been transmitted but their acknowledgements are yet to be received, and
- the data frames which are next to be transmitted.

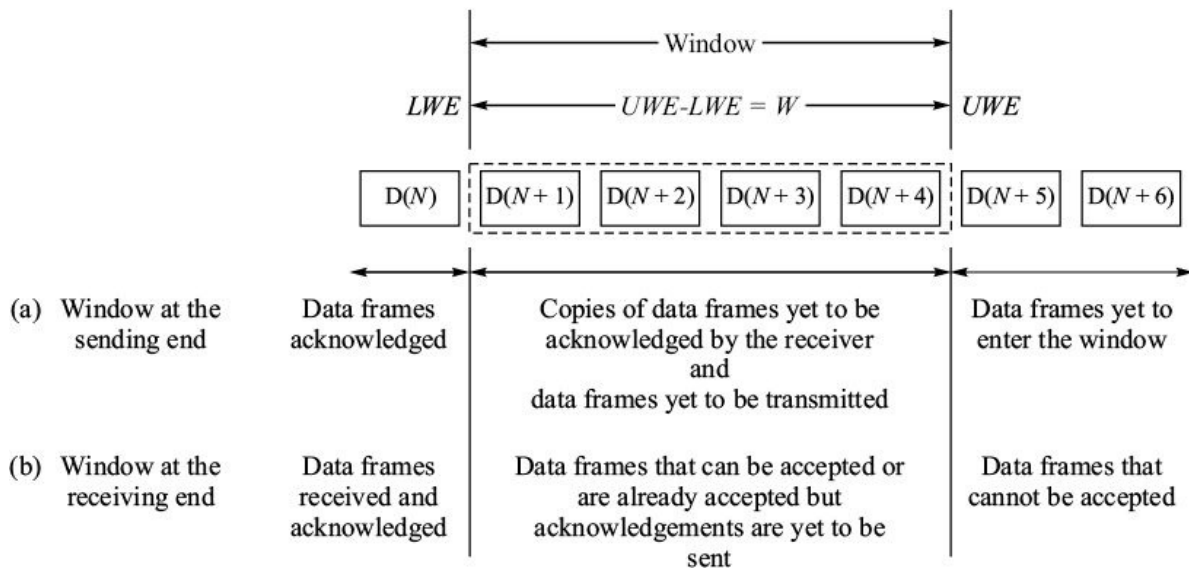


Figure 8.13 Semantics of sliding window at the transmitting and receiving ends.

There can be altogether W such data frames in the window. When an acknowledgement is received, copies of acknowledged data frames are deleted from the window by sliding up the window. Equivalent number of data frames awaiting transmission make entry in the window.

Window at receiving end. At the receiving end, the window contains the sequence numbers of the data frames the receiver is ready to accept. Any received data frame having sequence number which is not in the window is discarded. When an acknowledgement is sent, the window slides up to the sequence number of data frame for which acknowledgement is to be sent next. The receiver does not send acknowledgements on receipt of data frames immediately when it is too busy or there are missing data frames. When the missing frame is received, the receiver can send one acknowledgement for several frames and slide up the window by several steps. We will examine this aspect in detail when we discuss error control.

Flow control mechanism. Figure 8.14 illustrates operation of the sliding window flow control mechanism. Device A is sending frames to device B. Let us assume that the window size is seven and the window is initially located on data frames D_0 to D_6 . A initiates the transmission with its first frame D_0 followed by frames D_1 , D_2 , etc. A can send frames up to D_6 without getting any acknowledgement from B.

While A is in the process of sending D_2 , it receives an acknowledgement RR2 from B. A slides the window by two frames deleting copies of D_0 and D_1 from

the window. Data frames D2 to D8 now occupy the window and frames up to D8 can be sent without waiting for further acknowledgement.

At the receiving end B accepts D0 and D1 data frames. It releases RR2 acknowledging receipt of D0 and D1, and removes these frames from the window. The receive window that was at D0-D6 slides to D2-D8. It next receives D2, D3, and D4 and releases RR5.

Link utilization. Unlike the stop-and-wait mechanism, in the sliding window flow control, the sending end can send a number

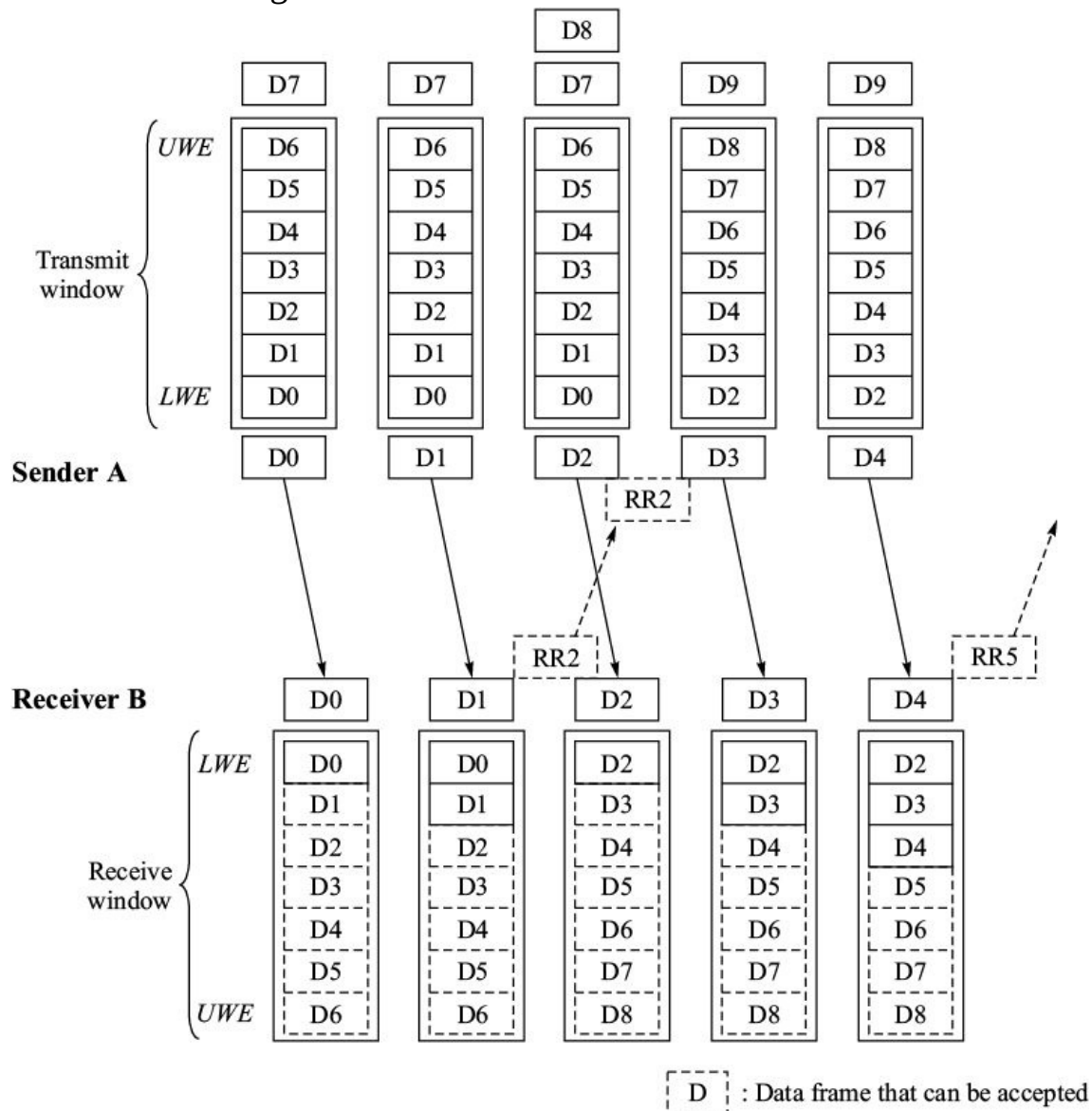


Figure 8.14 Sliding window flow control mechanism.

of frames one after the other without waiting for acknowledgement, which results in better link utilization. To calculate link utilization, let us consider the following two possible situations:

- The sender receives an acknowledgement before it exhausts the window.
- The sender exhausts the window before it receives an acknowledgement.

In the first situation, the sender can keep transmitting data frames without interruption. The link utilization is unity. If the window size is W , this situation will occur when the time required to transmit W frames is more than the earliest possible arrival of an acknowledgement, which is $t_f + 2t_p$. It is assumed that the size of acknowledgement is very small and the receiver responds with acknowledgement immediately on receipt of a data frame.

$$Wt_f \geq t_f + 2t_p \text{ or}$$

$W \geq 1 + 2A$, where $A = t_p/t_f$ In the second situation, the sender transmits W frames in time Wt_f and then suspends further transmission of the frames until an acknowledgement is received. If we assume that the receiver sends the acknowledgement at the earliest opportunity, *i.e.* immediately following the receipt of the first frame, the acknowledgement will be received after time $t_f + 2t_p$. Therefore, the second situation will occur when $Wt_f < t_f + 2t_p$.

or

$W < 1 + 2A$, where $A = t_p/t_f$ In this case the link has been engaged for time $t_f + 2t_p$ while A has utilized it for time Wt_f only. Therefore, link utilization is given

$$\text{by } U = \frac{Wt_f}{t_f + 2t_p} = \frac{W}{1 + 2A}$$

Figure 8.15 shows link utilization efficiency as a function of A . Curves for three window sizes 1, 7, and 127 have been shown. When window size is 1, the sliding window mechanism degenerates to simple stop-and-wait mechanism. Note that the sliding window flow control mechanism permits use of higher values of A without sacrificing link utilization efficiency. Higher value of A means shorter frame size. Short frames have two advantages:

- Probability of occurrence of errors in the frame is reduced. We will examine this issue in the next section.

- Even if an error occurs, wasted time is less.

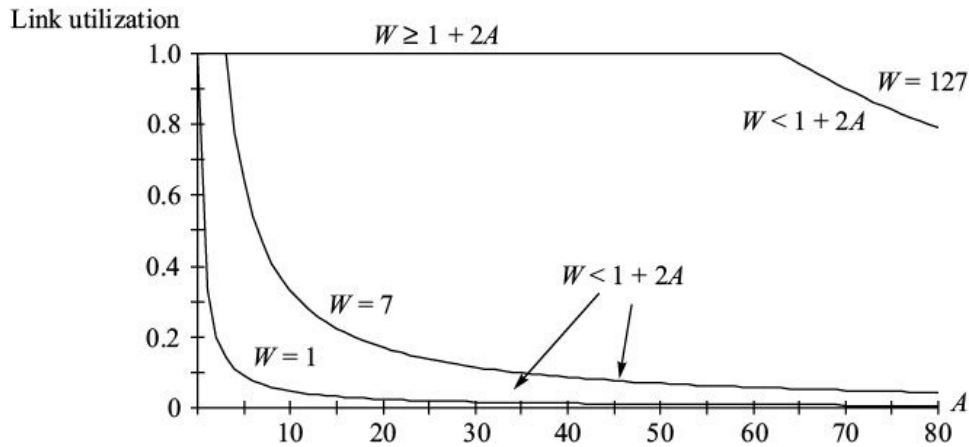


Figure 8.15 Link utilization in sliding window flow control.

EXAMPLE 8.3 Calculate link utilization efficiency if (a) Bit rate (R) = 19.2 kbps, frame size (L) = 960 bits, window size (W) = 3, propagation time (t_p) = 0.06 s.

(b) Bit rate (R) = 19.2 kbps, frame size (L) = 960 bits, window size (W) = 7, propagation time (t_p) = 0.06 s.

What is the minimum window size for 100 per cent link utilization?

Solution

$t_f = 960/19200 = 0.05$ s, $A = 0.06/0.05 = 1.2$, $2A + 1 = 3.4$, (a) $W = 3 < 2A + 1$,
therefore $U = 3/3.4 = 88\%$

(b) $W = 7 > 2A + 1$, therefore $U = 100\%$

Minimum size of the window for $U = 100\%$ can be obtained by setting $W = 2A + 1 = 3.4$

Therefore, the window size should be at least equal to 4 for 100% link utilization efficiency.

8.5 DATA LINK ERROR CONTROL

Two types of errors can occur during transmission of frames from one device to the other:

- Content errors
- Flow integrity errors.

Errors contained in a received frame are termed *content errors*. *Flow integrity errors* refer to the lost or duplicate data frames and acknowledgements. Data link error control takes care of both the types of errors.

Content errors are detected using parity check or cyclic redundancy check bits. The check bits are added as the trailer in a frame at the sending end. Their span of check usually cover all the bits in the frame except the frame identifier (Figure 8.16). If the flag itself is corrupted, the receiver will not recognize the arrival of the frame.

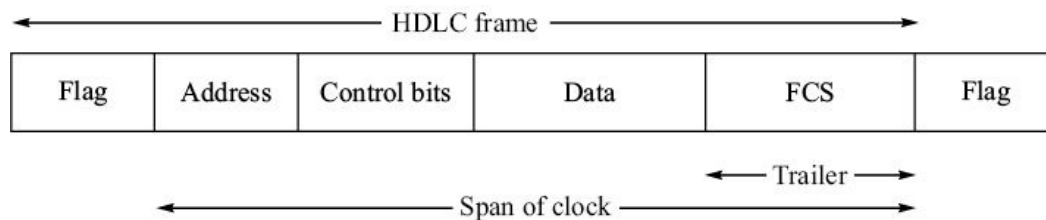


Figure 8.16 Span of error check.

The most common method of content error correction is retransmission of the frames.

The receiver informs the sending end of the error and the sending end retransmits the frame. Note that it is essential for the sending end to retain a copy of the transmitted frame until it is acknowledged by the receiver.

For flow integrity errors, the data link protocols specify the procedures to be adopted to detect and recover the missing/duplicate frames and acknowledgements. These procedures are built into the flow control mechanisms.

It must be remembered that no error control method is 100 per cent effective. There will always be some undetected content and flow integrity errors. Residual Error Rate (RER) refers to the errors that still exist in the data stream after all the error control procedures have been completed.

8.6 ERROR CONTROL IN STOP-AND-WAIT MECHANISM

There are two ways of implementing the content error control in stop-and-wait mechanism.

- Stop-and-wait using timeout

- Stop-and-wait using negative acknowledgement (NAK) and timeout.

We will examine these schemes for various possible instances of content and flow integrity errors.

8.6.1 Stop-and-Wait Using Timeout

In this case, the sender maintains a timer and if the acknowledgement is not received within a predefined time interval, it retransmits the frame. Figure 8.17 illustrates the mechanism. A is the sender of data frames and B is the receiver. B sends acknowledgements.

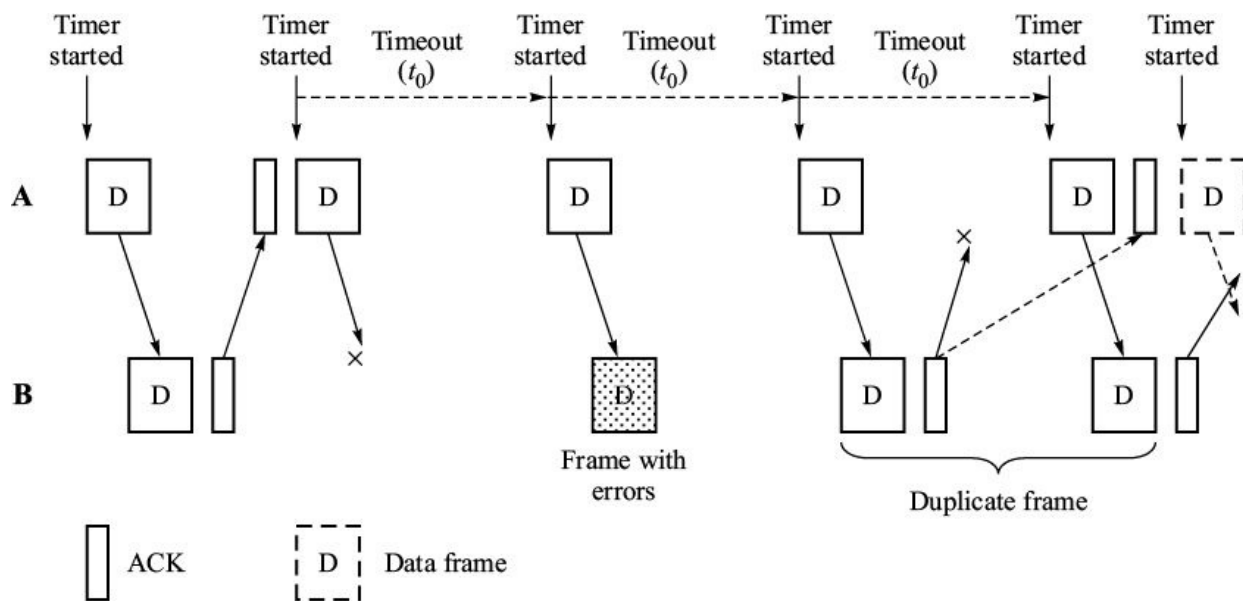


FIGURE 8.17 Error control in stop-and-wait mechanism using timeout.

a. Lost or damaged data frames (Figure 8.17)

- A sends a data frame and starts the timer.
- The frame is lost in transit. B does not receive any frame, or B receives the data frame with content errors. It discards the frame.
- A waits for ACK and after timeout retransmits the frame.

b. Lost or delayed ACK (Figure 8.17)

- B sends ACK for the received data frame.
- The ACK is lost or is delayed. After timeout, A retransmits the frame.
- B receives duplicate data frames but B does not know that the received

frame is duplicate frame.

In case of delayed ACK, A having retransmitted the data frame after timeout, assumes that the received ACK is for the retransmitted data frame and it proceeds with the next data frame.

8.6.2 Stop-and-Wait Using Negative Acknowledgement (NAK) In timeout scheme, if an error is detected by the receiver, this fact is not conveyed back to the sender. The sender always waits for timeout and then retransmits the frame. Timeout interval is kept sufficiently large to account for all possible delays. This leads to inefficient utilization of the link. By implementing a negative acknowledgement (NAK) scheme along with timeout, this shortcoming can be overcome to an extent.

In this scheme, the receiver sends a positive acknowledgement (ACK) if there is no content error in the received data frame; else it responds with a negative acknowledgement (NAK). The sending end continues with the next data frame if it receives an ACK or repeats the previous frame if it receives a NAK. Figure 8.18 illustrates the mechanism.

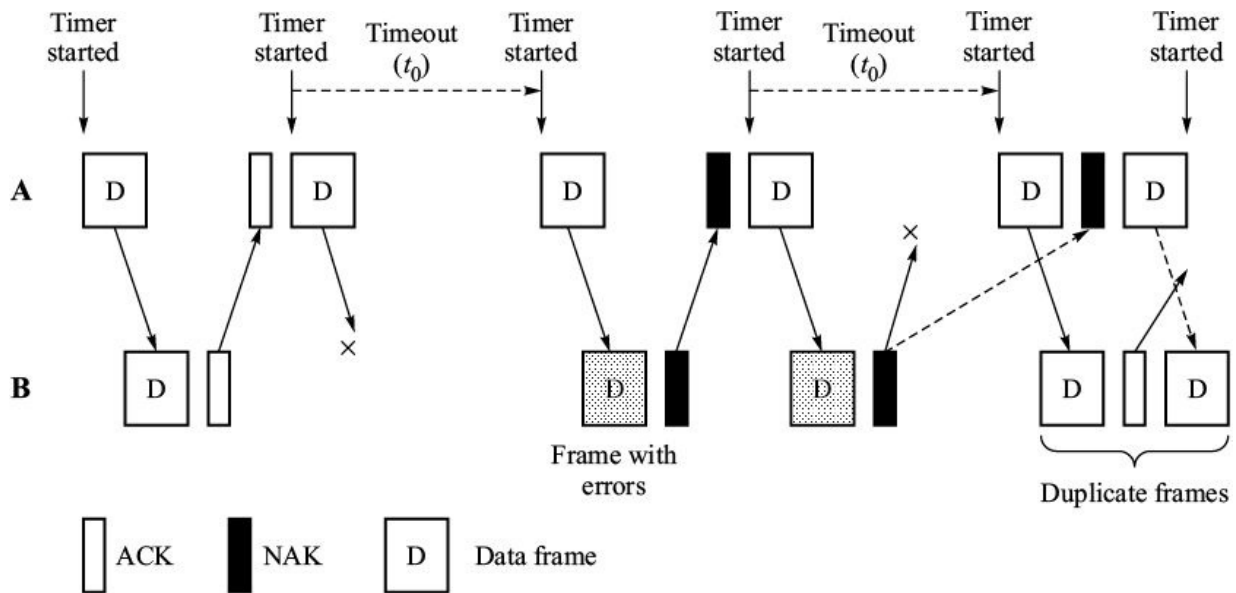


Figure 8.18 Error control in stop-and-wait mechanism using NAK.

a. Lost or damaged data frame (Figure 8.18)

- A sends a data frame and starts the timer.
- The data frame is lost in transit. B does not receive any frame. A does not receive ACK/NAK. After expiry of timer, A retransmits the data frame.
- B receives the data frame with content errors. It replies with a NAK. A receives NAK before timeout. It immediately retransmits the data frame.

b. Lost or delayed NAK (Figure 8.18)

- B sends NAK for the received data frame with errors. The NAK is lost or is delayed. After timeout, A retransmits the data frame in either case.
- In case of delayed NAK, A having retransmitted the data frame, assumes that the received NAK is for the retransmitted data frame and it retransmits the data frame again. B receives duplicate data frame but B does not know that the received frame is duplicate frame.

In both the above schemes, timeout and NAK, there is problem of receiving duplicate frames. The receiver has no method of identifying duplicate data frames. Duplicate frame is an error situation because duplicate frames are individually processed to retrieve the user data, which also gets duplicated.

An alternative scheme overcomes problem of duplicates by using sequence numbers to distinguish between consecutive data frames and acknowledgements. Let us see how the sequence numbers resolve this issue.

8.6.3 Error Control Using Numbered Frames To distinguish between consecutive frames and acknowledgements, a sequence number is attached to them. Since only one frame or one acknowledgement is in transit, only two numbers 0 and 1 are required. The data frames are alternatively assigned numbers 0 and 1. We will call data frames as D0 and D1. The ACKs and NAKs also carry the number. The usual practice for assigning number to ACKs and NAKs is to indicate the number of data frame next wanted, e.g. ACK1 indicates to the sender that D1 is required next. It also implies that previous data frame D0 has been received. NAK1 indicates that D1 is required next because last received data frame D1 had errors.

Figure 8.19 shows the examples of timeout and NAK error control schemes

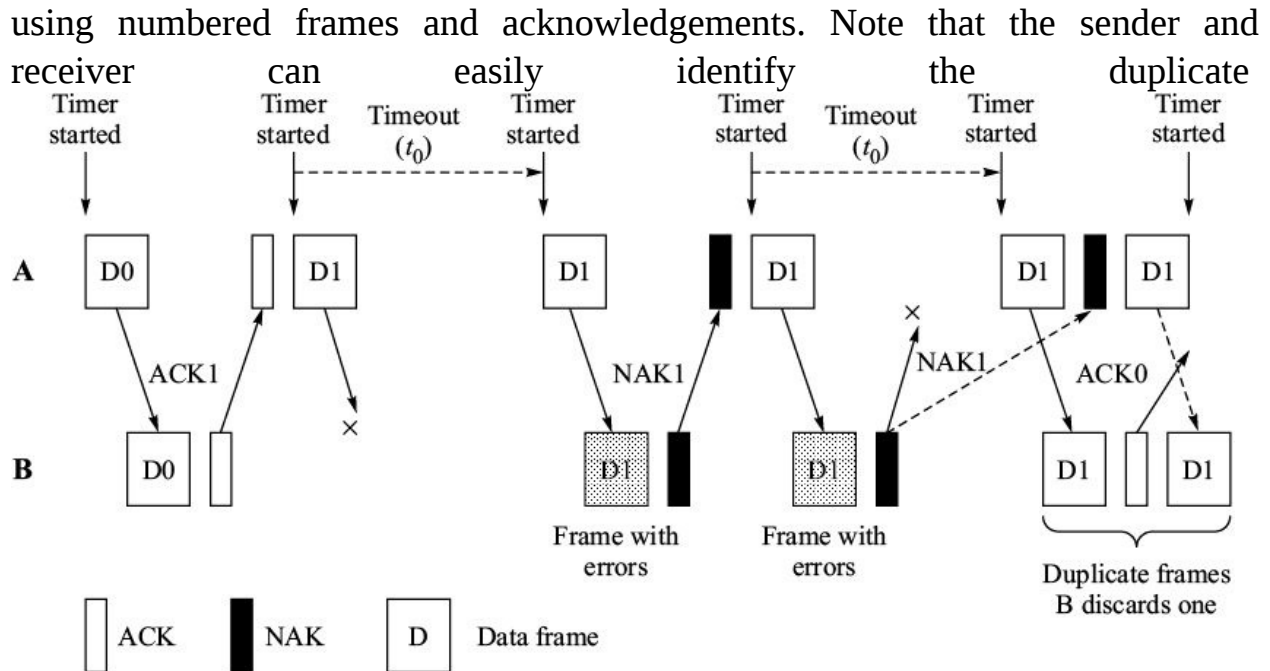


Figure 8.19 Error control in stop-and-wait mechanism using NAK and numbered frames.

frames and acknowledgements. The duplicates when received are discarded. However, the receiver needs to send the acknowledgement for duplicate frames to keep the sender in synchronization.

Note that stop-and-wait mechanism with numbered frames is degenerate case of sliding window flow control mechanism with window size 1. The advantage of the stop-and-wait mechanism is its simplicity, but as discussed earlier, it is inefficient from the link utilization point of view.

8.6.4 Link Utilization in Presence of Errors In section 8.4.2, we calculated link utilization efficiency when there were no errors. Having understood the stop-and-wait error control mechanism, we can now calculate the link utilization efficiency for a given bit error rate.

Average number of transmissions. If p is probability that a bit is corrupted by random error, and L is the number of bits in a frame, probability that a frame is received without any error is $(1 - p)^L$. Thus probability P_f that a frame is received with one or more errors is given by $P_f = 1 - (1 - p)^L$

The probability of transmitting a frame successfully in the i th attempt is

$$P_f^{i-1}(1 - P_f). \text{ We can calculate the average number of transmissions } (N_r) \text{ as } N_r = \sum_{i=0}^{\infty} iP_f^{i-1}(1 - P_f)$$

Using geometric series summation formula, we get $\sum_{i=0}^{\infty} iP_f^{i-1} = \frac{d}{dP_f} \sum_{i=0}^{\infty} P_f^i = \frac{d}{dP_f} \left(\frac{1}{1 - P_f} \right) = \frac{1}{(1 - P_f)^2}$

Therefore, average number of transmissions required to receive a frame correctly is given by $N_r = \frac{1}{(1 - P_f)^2} (1 - P_f) = \frac{1}{1 - P_f} = \frac{1}{(1 - p)^L}$

where P_f is probability of receiving a frame in error, p is probability of receiving a bit in error, and L is length of the frame.

EXAMPLE 8.4 What is the average number of transmissions required to send a frame of length 1000 bytes correctly, if the bit error rate is $1 \cdot 10^{-4}$?

Solution

$$p = 0.0001$$

$$L = 1000 \text{ bytes} = 8000 \text{ bits } N_r = 1/(1 - 0.0001)^{8000} = 2.22$$

Thus, the average number of transmissions required to send the frame correctly is 2.22.

Link utilization in stop-and-wait with timeout. If the average number of transmissions required for sending one data frame correctly is N_r , and timeout interval is t_0 , total time for which the link is engaged is:

- $(N_r - 1)t_0$ for releasing N_r frames on the link,
- $t_f + 2t_p$ for the last retransmission when the data frame is correctly received and acknowledged using ACK.

During this entire interval, the link is utilized effectively only for sending one data frame having transmission time t_f . Therefore, link utilization efficiency (U)

$$\text{is given by } U = \frac{t_f}{(N_r - 1)t_0 + t_f + 2t_p} = \frac{1 - P_f}{(1 + 2A)(1 - P_f) + P_f t_0 / t_f}$$

where $A = t_p/t_f$ and P_f is probability of receiving a frame with one or more errors. As before the following assumptions have been made in calculating link utilization efficiency:

- The receiver sends back ACK immediately on receipt of a data frame.
- The size of ACK frame is very small.
- There is no error in ACK.

Link utilization in stop-and-wait using NAK. To calculate the link utilization efficiency in this case, we assume that there are no flow integrity errors in the data frames or its acknowledgements. In other words, we have only the content errors. This assumption is in addition to the ones listed above.

If the average number of transmissions required to send one data frame correctly is N_r , the total time for which the link is engaged is $N_r (t_f + 2t_p)$, where t_f is frame transmission time and t_p is the propagation time. During this time only one frame is successfully sent; therefore the link utilization efficiency (U) is

$$\text{given by } U = \frac{t_f}{N_r(t_f + 2t_p)} = \frac{1 - P_f}{1 + 2A}$$

where

$$A = \frac{t_p}{t_f} = \frac{t_p R}{L}$$

and P_f is the probability of receiving a frame with one or more errors and is given by $P_f = 1 - (1 - p)^L$

Figure 8.20 shows plot of link utilization with respect to frame size. Note that there is an optimum size of frame (L) that maximizes link utilization efficiency for the given set of link parameters (t_p , R , p).

EXAMPLE 8.5 Calculate the link utilization for stop-and-wait flow control mechanism based on NAK, if $\text{BER} = 1 \cdot 10^{-4}$

$$t_p = 40 \text{ ms } R = 1 \text{ Mbps } L = 1000 \text{ bytes } \textbf{Solution}$$

$$A = t_p/t_f = 40 \cdot 10^{-3}/(8000/10^6) = 5$$

$$p = 10^{-4}$$

$$P_f = 1 - (1 - p)^L = 1 - (1 - 10^{-4})^{8000} = 0.550689$$

$$U = (1 - 0.550689)/(1 + 2 \cdot 5) = 0.040 = 4\%$$

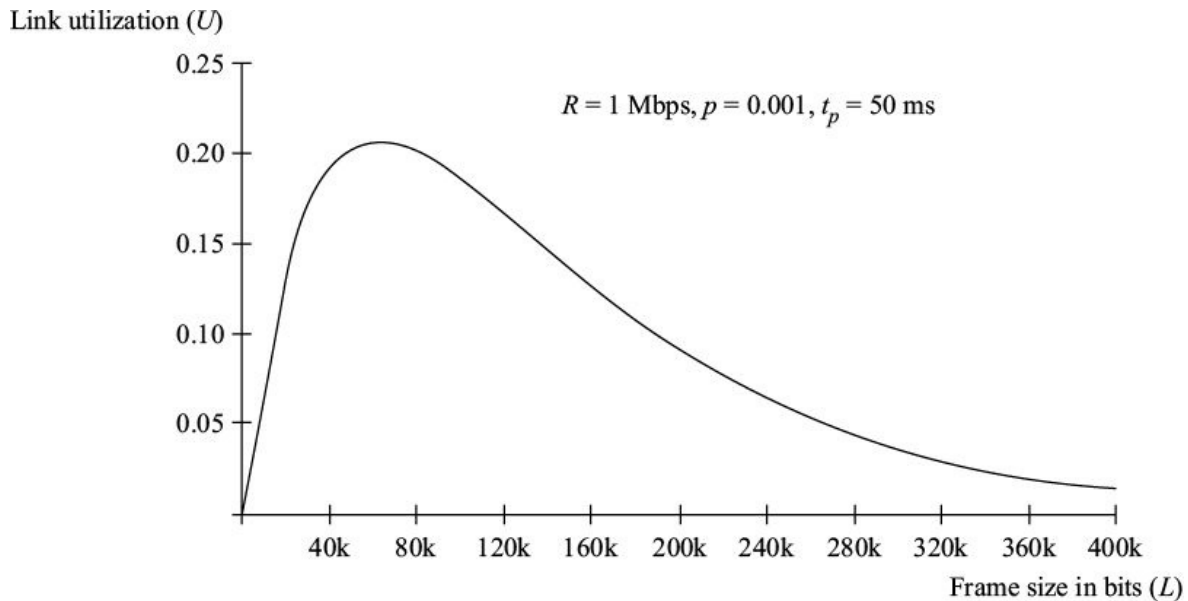


Figure 8.20 Link utilization of stop-and-wait flow control in presence of errors.

8.7 ERROR CONTROL IN SLIDING WINDOW MECHANISM

In the sliding window mechanism, each frame is assigned a sequence number and, therefore, a more elaborate error control scheme is feasible. The receiver keeps track of the sequence numbers of the incoming data frames. If any out-of-sequence frame is received, immediately, a request for retransmission of the missing frame is sent. There are two choices for the retransmission mechanism:

- Selective retransmission
- Go-back- N .

Selective retransmission. In selective retransmission, the receiver maintains a window of data frames that it is ready to accept. The receive window size is usually same as transmit window size. Whenever the receiver detects a missing data frame, it requests retransmission of the missing data frame by sending a Selective Reject (SREJ— N). N is the sequence number of the missing data frame. On receipt of SREJ— N , the sending end retransmits frame N and then it continues with the ongoing transmission. The receiver accepts the succeeding data frames meanwhile if they are received. When the data frame N is received, it arranges all the frames in proper sequence,

retrieves the user data, and hand it over to the next upper layer.

Go-Back— N . In go-back- N retransmission, the receiver maintains a window of size 1. The receiver can, therefore, accept only the next data frame in sequence. Whenever it notices a missing frame, it requests retransmission of the missing data frame by sending Reject (REJ— N). If any other data frame, other than frame N is received before frame N , the receiver discards it. REJ— N indicates request for retransmission of the all data frames starting with the frame N .

In both the above schemes missing frames are detected when an out-of-sequence frame is received. REJ or SREJ are not sent on detection of content error in the received data frame as the error could be in the sequence number of the frame itself. The receiver waits for the next correct frame and then declares the missing frame. Both SREJ— N and REJ— N also acknowledge receipt of data frames up to $N - 1$.

8.7.1 Error Control Using Selective Retransmission

Figure 8.21 illustrates the selective retransmission mechanism for window size of 7 at the sending end (A) and receiving end (B). As before we will consider various error situations, loss of data frame, loss of RR, and loss of SREJ.

a. Loss of a data frame (Figure 8.21)

- A has its transmit window at D0-D6. It sends D0.
- B has its receive window at D0-D6. It receives D0, replies with RR1, and slides its receive window to D1-D2.
- A sends D1 which is lost in transit or is received with errors by B.
- A receives RR1 and it slides window to data frames D1-D7. A sends D2.
- B accepts D2. But it finds missing data frame D1. It sends SREJ1. After sending a SREJ1, B cannot send RR till it receives the missing data frame D1 because sending RR3 would imply receipt of all earlier frames.
- A sends D3. B accepts it.
- A receives SREJ1. It sends D1.
- B accepts D1 and replies with RR4 acknowledging receipt of D1, D2, and D3. B slides its receive window to D4-D10.
- After sending D1, A continues with sending next data frame D4. A receives RR4 and slides its window to D4-D10.
- B accepts D4, releases RR5 and slides its receive window to D5-D11.

A missing data frame is detected by the receiver when the next data frame in sequence is received. If the lost frame was the last data frame in the transmit window, the receiver will never respond with SREJ. Therefore, the sender maintains a timer. The timer is started when the sender sends the last frame in the transmit window. After timeout, it repeats all the frames in the

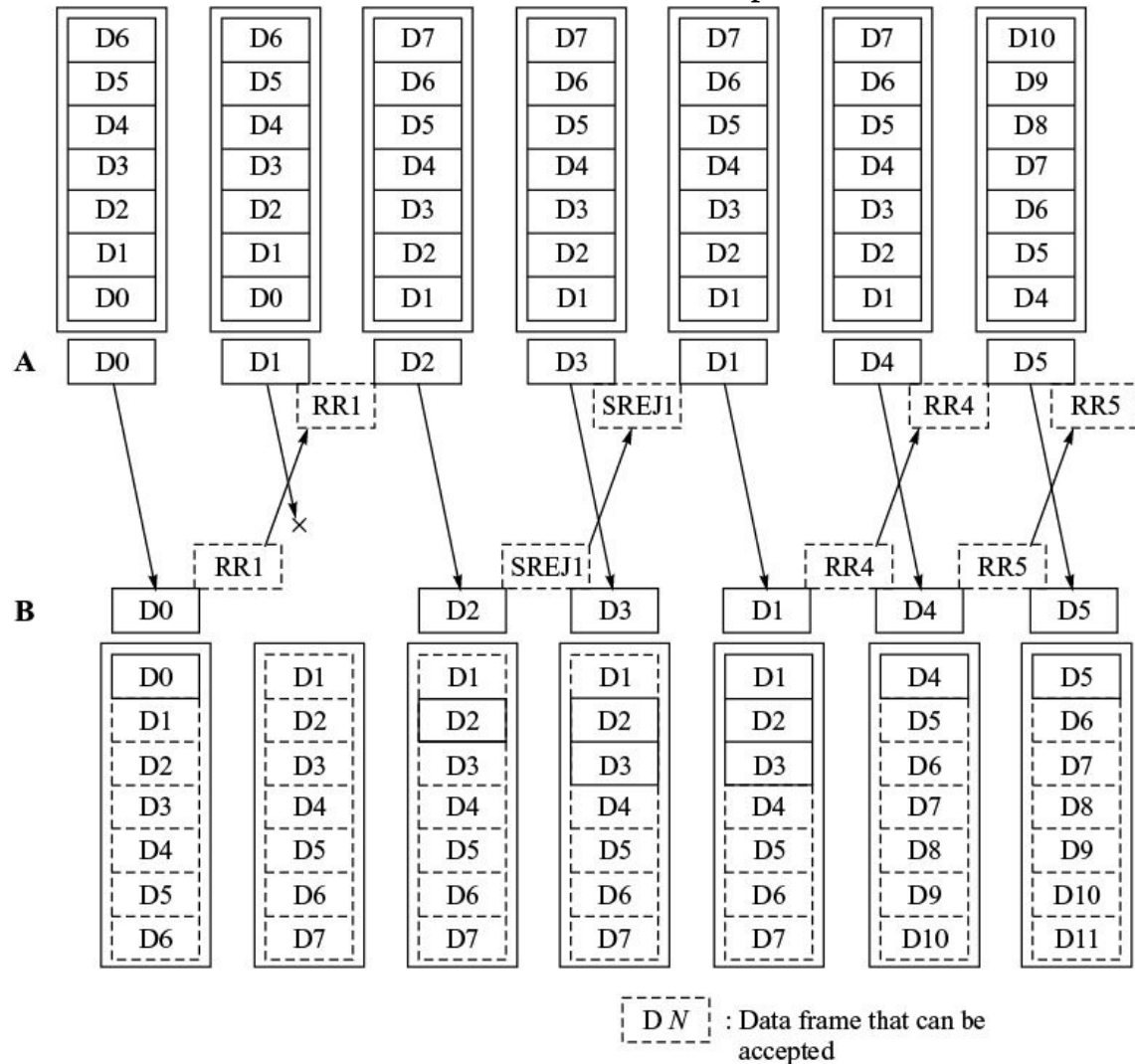


Figure 8.21 Selective retransmission in sliding window flow control mechanism.

window. Alternatively, it solicits response from the receiver by sending a command. In HDLC protocol that we discuss in the next chapter, the later approach is adopted.

b. Loss of RR or SREJ. If an RR is lost or delayed, the next RR released by the receiver acknowledges all the previous data frames. Therefore, the operation is unaffected. If there is no subsequent RR, the sender initiates action as indicated above after the timeout.

To account for the possibility of loss of SREJ, the receiver also maintains a timer. It is started as soon as SREJ is released. If the required data frame is not received within timeout, SREJ is sent again by the receiver.

8.7.2 Error Control Using Go-Back—N

Figure 8.22 illustrates the go-back-N mechanism for window size of seven at the sending end (A). The window size at the receiving end (B) is always 1 for go-back-N mechanism. As before, we will consider various error situations, loss of data frame, loss of RR, and loss of REJ.

a. Loss of a data frame (Figure 8.22)

- A has its transmit window at D0-D6. It sends D0.
- B has its receive window at D0. It receives D0, replies with RR1 and slides its receive window to D1.
- A sends D1 which is lost in transit or is received with errors by B.
- A receives RR1 and it slides window to data frames D1-D7. A sends D2.
- On receipt of D2, B finds missing data frame D1. B discards D2 and sends REJ1.
- A sends D3. B discards D3.
- A receives REJ1. It sends D1.
- B accepts D1 and sends RR2.
- A sends D2. On receipt of RR2, A slides its window to D2-D8. A sends D3.
- B accepts D2 and sends RR3.

As before, the sender maintains a timer to take into account the possibility that the lost data frame may be the last frame in the transmit window. After timeout, either the sender solicits response from the receiver by sending a control frame or repeats all the data frames in the transmit window.

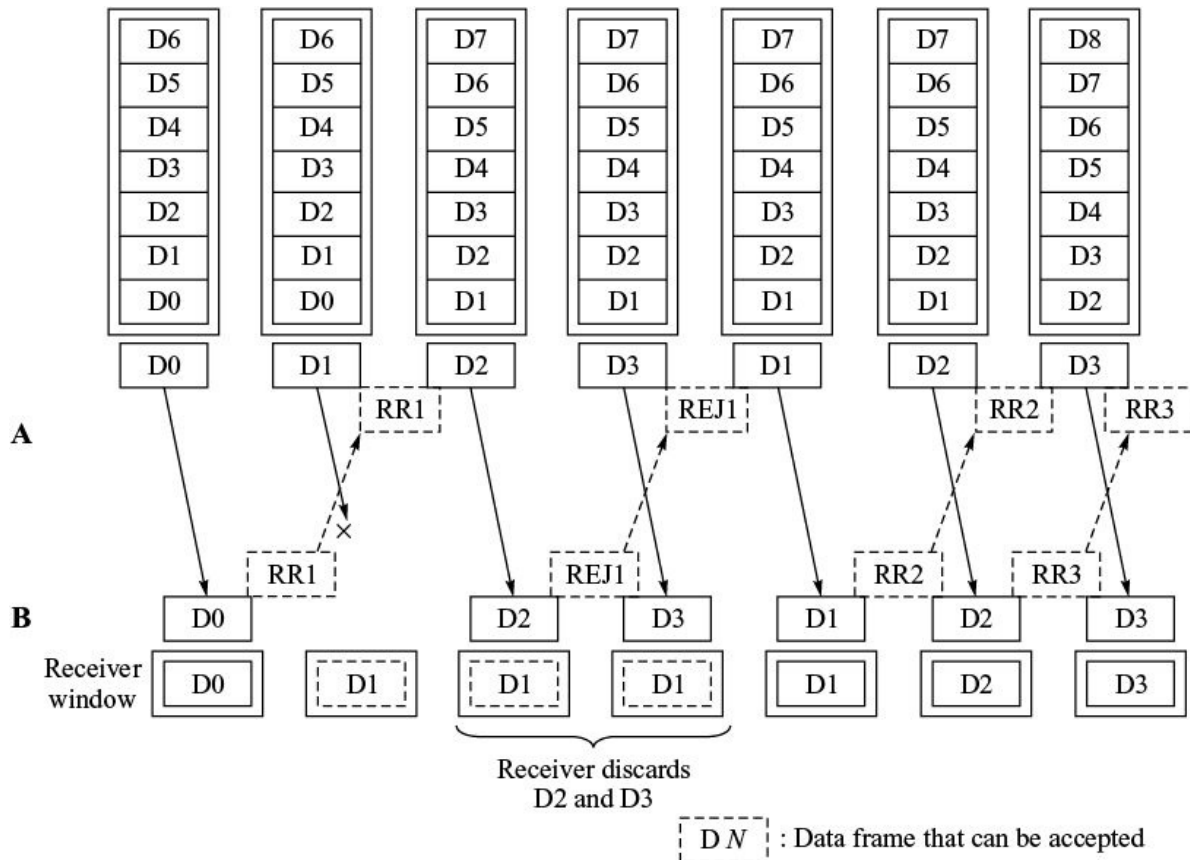


Figure 8.22 Go-back-N error control in sliding window mechanism.

b. Loss of a RR or REJ (Figure 8.22). If an RR is lost or delayed, the next RR released by the receiver acknowledges all the previous data frames. Therefore, the operation is unaffected. If there is no subsequent RR, the sender initiates action as indicated above after the timeout. To account for the possibility of loss of REJ, the receiver maintains a timer and retransmits the REJ after timeout.

8.7.3 Link Utilization in Presence of Errors in Sliding Window Flow Control

As before we consider the following two possible situations for arriving at the expressions for link utilization in presence of errors:

- The sender receives an acknowledgement before it exhausts the window.
- The sender exhausts the window before it receives an acknowledgement.

If the window size is W , the first situation occurs when $W \geq 1 + 2A$, and the second situation occurs when $W < 1 + 2A$, where $A = t_p/t_f$, t_f is the time required

to transmit a frame and t_p is one way propagation time. When there are no errors, the link utilization efficiency (U) for these two situations will be $U = 1$ when $W \geq 1 + 2A$ $U = \frac{W}{1+2A}$ when $W < 1 + 2A$ We will now consider selective retransmission and go-back-N mechanisms separately because their link utilization efficiencies are different.

Link utilization in selective retransmission. When $W \geq 1 + 2A$, the link is continuously busy for transmission of data frame during the round trip time for acknowledgement ($t_f + 2t_p$). If

P_f is the probability of receiving a frame with errors, we can calculate the link utilization in presence of errors as follows: No. of frames transmitted during ($t_f + 2t_p$) = $(t_f + 2t_p)/t_f$ No. of frames received with errors = $P_f (t_f + 2t_p)/t_f$ Link time wasted by the frames in error = $t_f P_f (t_f + 2t_p)/t_f = P_f (t_f + 2t_p)$ Time for which the link is effectively utilized = $(t_f + 2t_p) - P_f (t_f + 2t_p)$ Link utilization efficiency (U) = $[(t_f + 2t_p) - P_f (t_f + 2t_p)]/(t_f + 2t_p) = 1 - P_f$ When $W < 1 + 2A$, the link is used for time Wt_f , out of ($t_f + 2t_p$) for transmission of data frames. For the rest of the period, the link is idle. If P_f is the probability of receiving a frame with errors, we can calculate the link utilization in presence of errors as under.

$$\text{No. of frames transmitted during } (t_f + 2t_p) = W$$

$$\text{No. of frames received with errors} = P_f W$$

$$\text{Link time wasted by the frames in error} = t_f P_f W$$

$$\text{Time for which the link is effectively utilized} = Wt_f - t_f P_f W$$

Link utilization efficiency (U) = $(Wt_f - t_f P_f W)/(t_f + 2t_p) = W(1 - P_f)/(1 + 2A)$ Thus link utilization is reduced by a factor of $(1 - P_f)$ in either case. Figure 8.23 shows the plots of link utilization against the frame size for three cases:

- no errors
- BER = $1 \cdot 10^{-5}$
- BER = $1 \cdot 10^{-6}$.

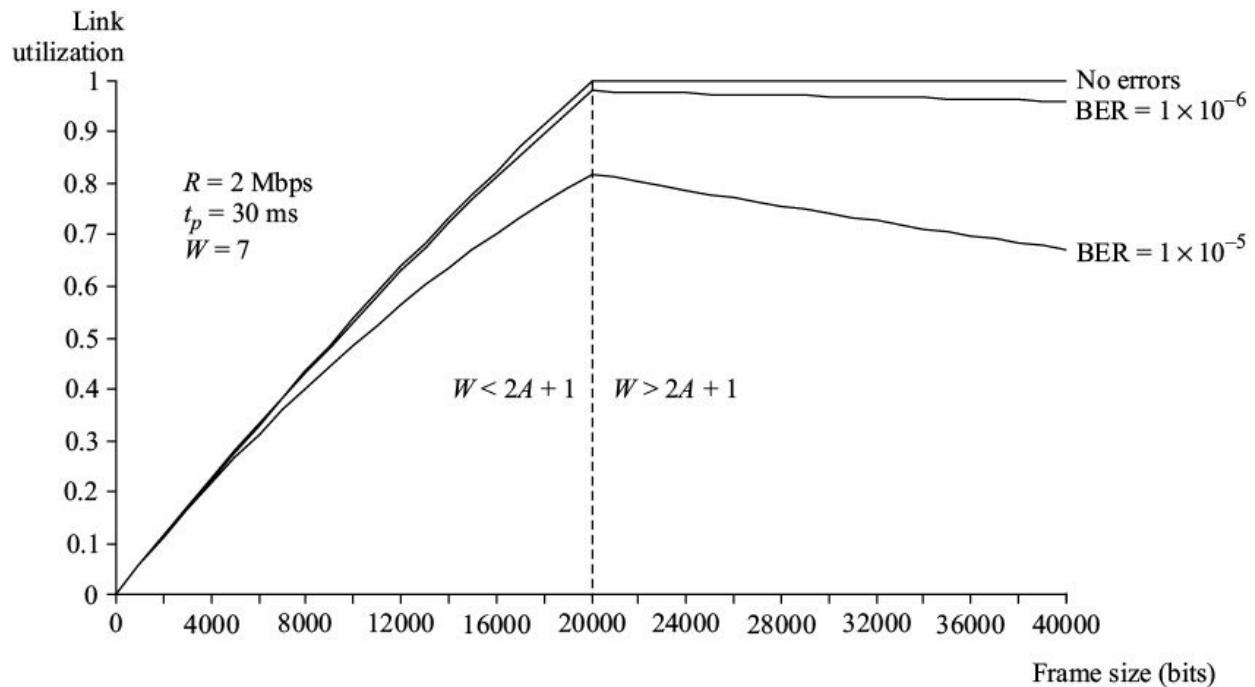


Figure 8.23 Link utilization in selective retransmission in presence of errors.

Some important observations are as follows:

1. When $W < 1 + 2A$ and there are no errors, the link utilization gradually increases as the frame size is increased. This is because the idle time of the link is reduced.
2. When $W < 1 + 2A$ and there are errors, there is still gradual increase in the utilization as the frame size is increased but the increase is at reduced rate. Increase in link utilization is due to reduced idle time of the link. The slope of the utilization curve decreases due to wasted link time when some of the frames are lost because of errors. As the frame size increases, the wasted link time also increases.
3. When $W > 1 + 2A$ and there are no errors, the link utilization is unity irrespective of the frame size.
4. When $W > 1 + 2A$ and there are errors, there is gradual decrease in the utilization as the frame size is increased. This is due to wasted link time when some of the frames are lost because of errors. As the frame size increases, the wasted link time also increases.
5. At higher bit error rate, there is all around reduction in link utilization.

EXAMPLE 8.6 Calculate link utilization for the following link parameters. The

data frame size is 1000 bytes and the flow control mechanism used is selective retransmission.

$$t_p = 40 \text{ ms}, \text{BER} = 1 \cdot 10^{-5}, R = 2 \text{ Mbps} \text{ The window size is (a) } W = 7, \text{ (b) } W = 127$$

Solution

(a) $W = 7$

$$p = 0.00001, P_f = 1 - (1 - 0.00001)^{8000} = 0.076884$$

$$t_f = \frac{8 \times 1000}{2 \times 10^6} = 4 \text{ ms}, A = \frac{40}{4} = 10$$

Therefore, $W < 1 + 2A$, and the link utilization is given by $U = \frac{7 \times (1 - 0.076884)}{2 \times 10 + 1} = 0.307$

(b) $W = 127$

In this case, $W > 2A + 1$. Hence, $U = 1 - 0.076884 = 0.923$

Link utilization in go-back-N. Link utilization for go-back-N, when there are no errors, is given by $U = 1$ for $W \geq 1 + 2A$ $U = \frac{W}{1 + 2A}$

for $W < 1 + 2A$ where W is size of the transmit window and $A = t_p/t_f$. The number of transmissions (N_r) required to send a data frame correctly for given frame error rate (P_f) is given by $N_r = \frac{1}{1 - P_f}$

We need to keep in mind that in go-back-N all the succeeding data frames are discarded by the receiver when a frame is received with errors. On each such occurrence, $2t_p + t_f$ of the link time is lost (Figure 8.24). To simplify the mathematical model, we assume that REJ is sent immediately on receipt of a data frame with errors instead of waiting for the next frame.

With this background, we can now derive an expression for link utilization. As before, we will consider two cases, $W \geq 1 + 2A$ and $W < 1 + 2A$.

a. $W \geq 1 + 2A$. On average a data frame requires N_r transmissions to send it correctly once. Therefore, $(N_r - 1) (2t_p + t_f)$ of link time is wasted. The sender utilizes the link effectively for time equal to t_f for sending the last transmission.

Therefore, link utilization efficiency is given by $U = \frac{t_f}{t_f + (N_r - 1)(2t_p + t_f)} = \frac{1}{1 + (N_r - 1)(2A + 1)} = \frac{(1 - P_f)}{(1 - P_f) + P_f(2A + 1)} = \frac{1 - P_f}{1 + 2P_f A}$

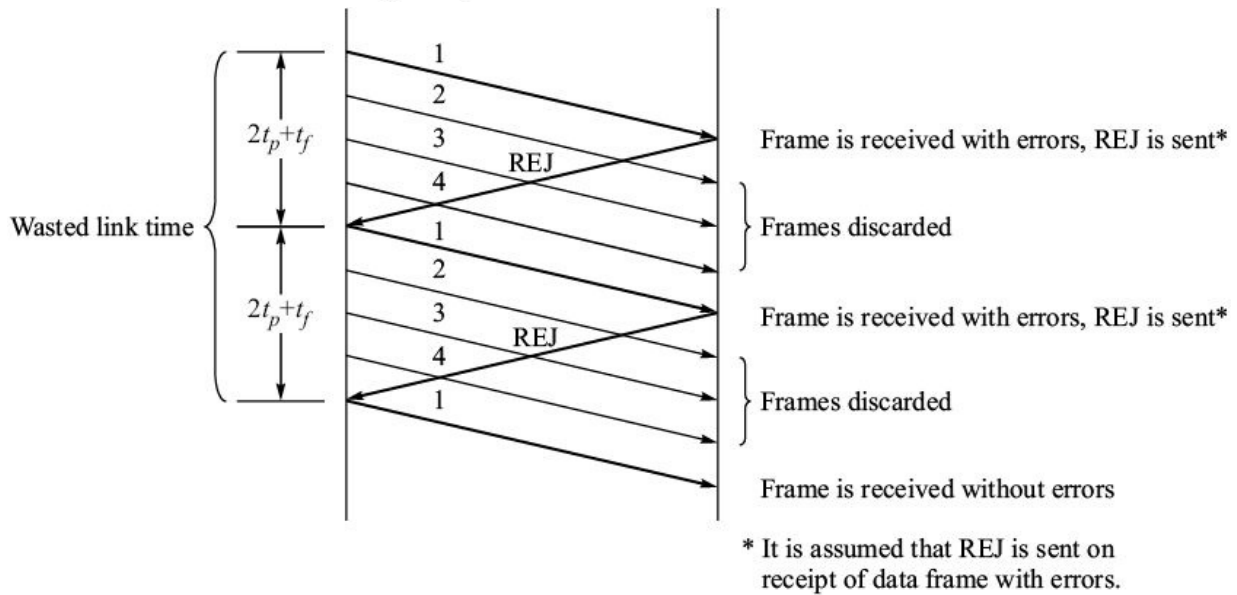


FIGURE 8.24 Link utilization in go-back-N when $W \geq 2A + 1$.

b. $W < 1 + 2A$. In the previous case, the sender was never idle because $W \geq 1 + 2A$. But when $W < 1 + 2A$, the sender sends W data frames and then it remains idle till RR or REJ is received from the receiver. We need to take this factor into account. So long as a frame is received with error, the entire round trip time ($2t_p + t_f$) is wasted. This includes the idle time of the sender. Therefore, $(N_r - 1)(2t_p + t_f)$ of link time is wasted (Figure 8.25).

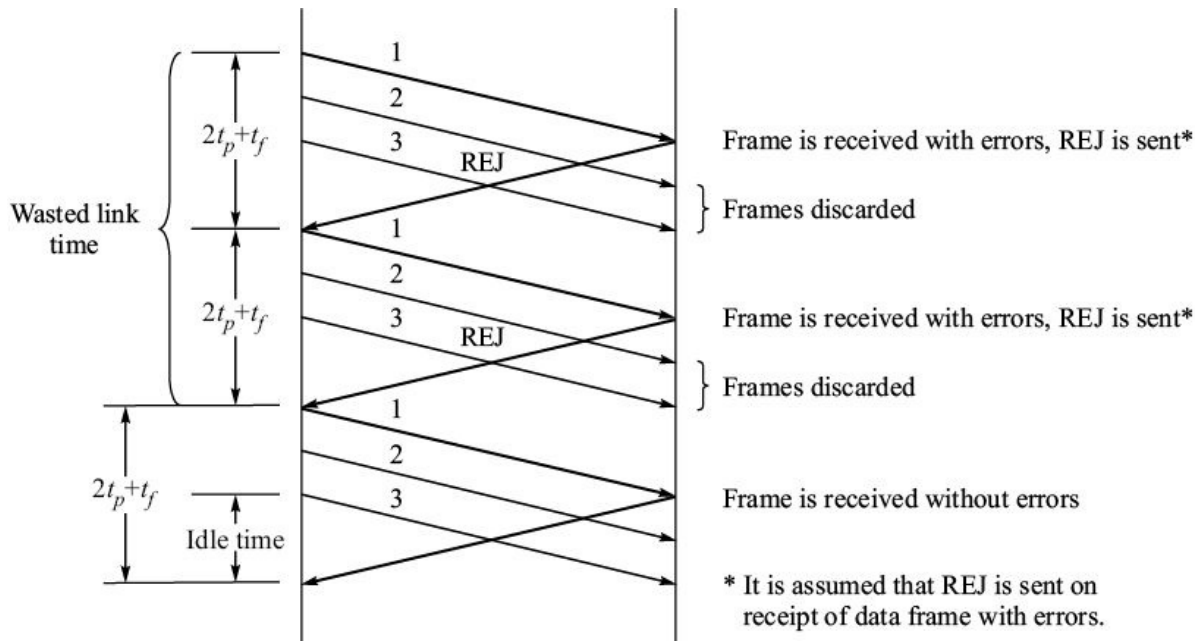


FIGURE 8.25 Link utilization in go-back-N when $W < 2A + 1$.

When the sender sends the data frame N_r th time, the data frame is received correctly. But this cycle of transmission of W data frames includes idle time of the sender also. The link is engaged for time $(2t_p + t_f)$ for sending W data frames. Therefore, on average a data frame engages the link for time equal to $(2t_p + t_f)/W$. The total time for which the link is engaged for a transmitting a data frame correctly is $(N_r - 1)(2t_p + t_f) + (2t_p + t_f)/W$. Therefore, link utilization (U) for go-back-N mechanism, when $W < 1 + 2A$ is given by $U = \frac{t_f}{(N_r - 1)(2t_p + t_f) + (2t_p + t_f)/W} = \frac{W}{(2A + 1)[1 + W(N_r - 1)]} = \frac{W(1 - P_f)}{(2A + 1)(1 - P_f + WP_f)}$

Figure 8.26 shows a plot of the link utilization with respect to the frame size. As before, plots for no error, $BER = 1 \cdot 10^{-5}$, and $BER = 1 \cdot 10^{-6}$ are shown. Note that compared to selective retransmission, go-back-N mechanism shows more sensitivity to errors. At $BER = 1 \cdot 10^{-5}$, the maximum link utilization that can be achieved is less than 0.4. In selective retransmission, maximum link utilization was about 0.8. In go-back-N, error in one frame results in retransmission of several frames and, therefore, more link time is wasted.

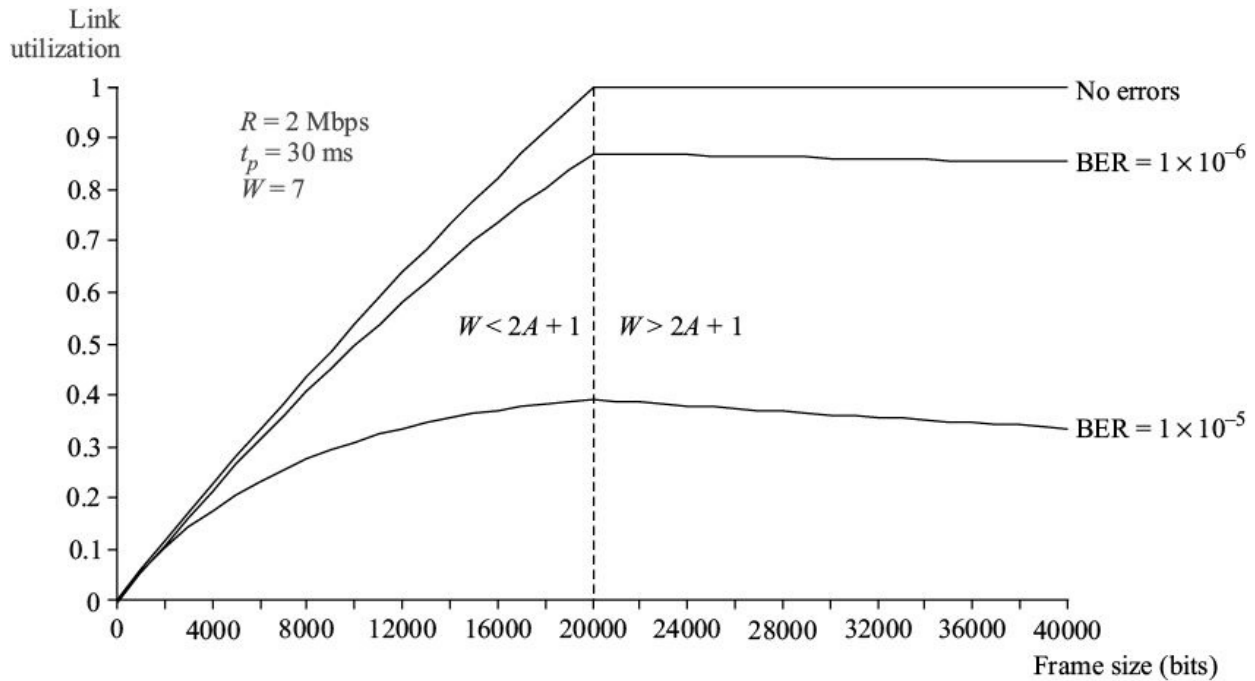


Figure 8.26 Link utilization in go-back-N in presence of errors.

EXAMPLE 8.7 Calculate link utilization for the following link parameters. The data frame size is 1000 bytes and the flow control mechanism used is go-back-N.

$t_p = 40$ ms, $BER = 1 \times 10^{-5}$, $R = 2$ Mbps The window size is (a) $W = 7$, (b) $W = 127$

Solution

(a) $W = 7$

As calculated in Example 8.6,

$$P_f = 0.076884, t_f = 4 \text{ ms}, A = 10$$

Therefore, $W < 1 + 2A$, and the link utilization is given by $U =$

$$\frac{7 \times (1 - 0.076884)}{(2 \times 10 + 1)(1 - 0.076884) + 7 \times 0.076884} = 0.21$$

(b) $W = 127$

In this case, $W > 2A + 1$. Thus, $U = \frac{1 - 0.076554}{1 + 2 \times 10 \times 0.07668} = 0.36$

8.8 SEQUENCE NUMBERING OF THE FRAMES IN SLIDING WINDOW FLOW CONTROL

In the sliding window mechanism, all data frames are given a binary sequence

number having a fixed number of bits. Any numbering scheme of fixed size has a finite count sequence after which it must start all over again from the beginning. If sequence number consists of n bits, the length of the count sequence would be 2^n . For example, the three-bit numbering scheme counts from 0 to 7 and then starts again from 0. If the window size is greater than the count sequence, there will be more than one data frames having same sequence numbers. This creates anomalies in operation. The receiver cannot distinguish the frames with same sequence number. It will discard the frame that is received later. The sender also cannot distinguish between the acknowledgements (RR, REJ, SREJ) for data frames having same sequence numbers. Therefore, the length of the count sequence must be at least equal to the size of the window. But window size equal to the count sequence results in some ambiguous situations as explained below.

Let us suppose window size is four and the two-bit numbering scheme is used. The status of window on receipt of RR2 is shown in Figure 8.27a. On receipt of RR2, A sends all the four data frames in the window and waits for acknowledgement. Suppose all the four data frames are lost in transit. After timeout A challenges¹ B. B repeats the previous acknowledgement (RR2). A assumes that this RR2 is acknowledgement of the data frame D1 it just sent. RR2 also acknowledges D0, D3, and D2. But as we know, B did not receive these four frames. This anomaly is resolved by restricting the window size to three (Figure 8.27b). When B repeats RR2, A immediately realizes the loss of three frames it just sent. Therefore, the maximum window size is restricted to $2^n - 1$ for an n -bit sequence number.

We assumed that after timeout, the sending end solicits acknowledgement from the receiving end. In a variation of this protocol, the sender just repeats all the data frames after timeout if it does not receive any acknowledgement. In this case the count sequence should be at least twice the window size if selective retransmission is used.

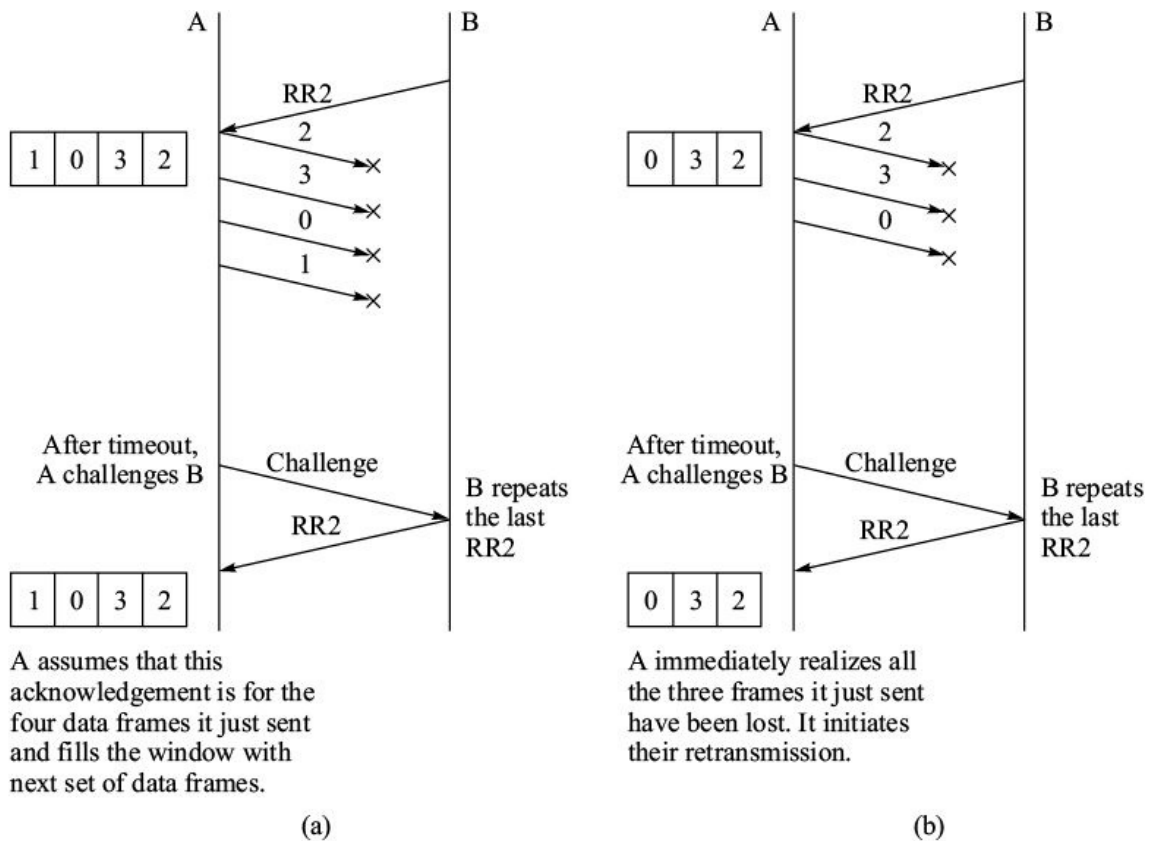


Figure 8.27 Window size in go-back-N.

8.9 PIGGYBACKING ACKNOWLEDGEMENTS

We have so far restricted ourselves to transmission of data frames from one device and the acknowledgements from the other device. In general, both the devices will exchange information and will send data frames and acknowledgements as shown in Figure 8.28. In the sliding window flow control mechanism, the acknowledgements can be sent through special acknowledgement frames or, alternatively, they can be piggybacked on a data frame. Thus, a data frame will have two sequence numbers, one for the data frame and the other for the acknowledgement.

Only RR can be piggybacked on a data frame. If a data frame does not have any new RR to report, it repeats the last RR sequence number. REJ, SREJ, and RNR are always sent as separate frames.

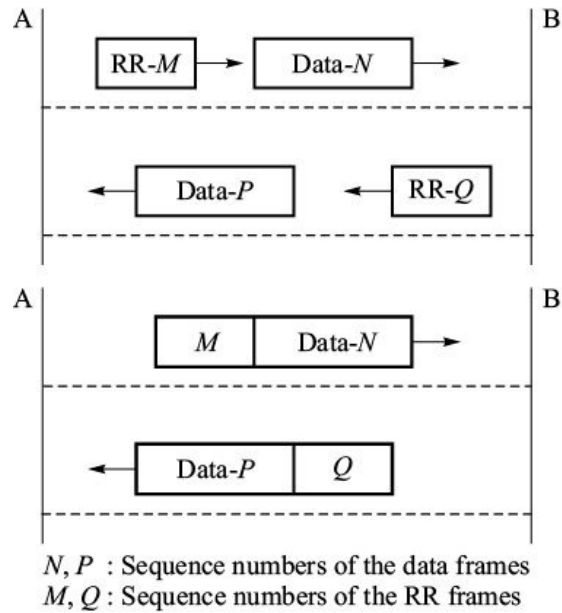


Figure 8.28 Piggybacking of acknowledgement.

8.10 DATA LINK MANAGEMENT

The data transfer process between two devices can be viewed as consisting of the following five phases (Figure 8.29):

1. Connect phase
2. Link establishment phase
3. Data transfer phase
4. Termination phase
5. Clear phase.

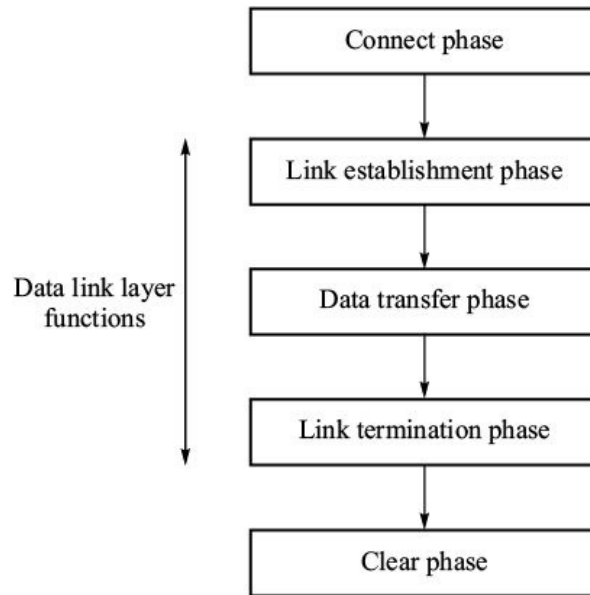


FIGURE 8.29 Link management phases.

The connect and clear phases consist of functions associated with establishing and releasing a physical connection between the two devices. These functions are part of the physical layer protocol. The span of data link protocols covers link establishment, information transfer and termination phases only. The data link management function involves execution of these phases.

Link establishment phase includes processes required to initialize the data link, to call/poll the other end and to set communication mode for data transfer (synchronous/asynchronous, TWA/TWS). The data transfer phase involves exchange of data and acknowledgements. When some abnormal situation arises during data transfer phase, which cannot be corrected by usual error control procedures, recovery procedures are initiated. The link may be required to be re-established. The termination phase consists of processes associated with disconnecting the link.

The data link protocols define set of control symbols and procedures to execute the above-mentioned functions. These symbols and procedures are specific to a data link protocol and, therefore, cannot be generalized.

Data link layer also offers connectionless-mode data transfer service. This mode is applicable to local area networks. In this mode of data transfer, data link need not be established and terminated. The data link entities are always in data transfer mode.

8.11 APPLICATION ENVIRONMENT OF

DATA LINK PROTOCOLS

Data link protocols find application in diverse networking situations. The major application environments of the data link layer protocols are depicted in Figure 8.30.

End system and subnetwork node. The data subnetwork nodes have first three layers of OSI reference model. Data link protocol operates between the node and end system (Figure 8.30a). The subnetwork node can be packet switched (X.25, IP) node or circuit switched (ISDN, X.21) node.

Nodes of a packet switched subnetwork. Error and flow control between the adjacent nodes of a subnetwork is taken care by implementing data link protocol between them. Thus in an end-system to end-system communication through a cascade of subnetwork nodes, each pair of adjacent nodes have a data link protocol operating between them (Figure 8.30b).

End systems connected through modems. Since modems are layer one device, data link protocol operates end to end in this case. The modes may be interconnected on a leased circuit or through a dial-up connection. The switched telephone network (PSTN) provides merely a physical path between the two modems (Figure 8.30c).

Nodes of a local area network. The local area network (LAN) operates in broadcast mode. The physical media interconnects all the stations on the LAN. LAN can take form of a bus or a ring (Figure 8.30d). The data link layer entity in every station communicates with every other data link entity in other stations. In a LAN, the data link layer has another unique function, media access control. Media access control is the discipline established for sharing a common transmission media.

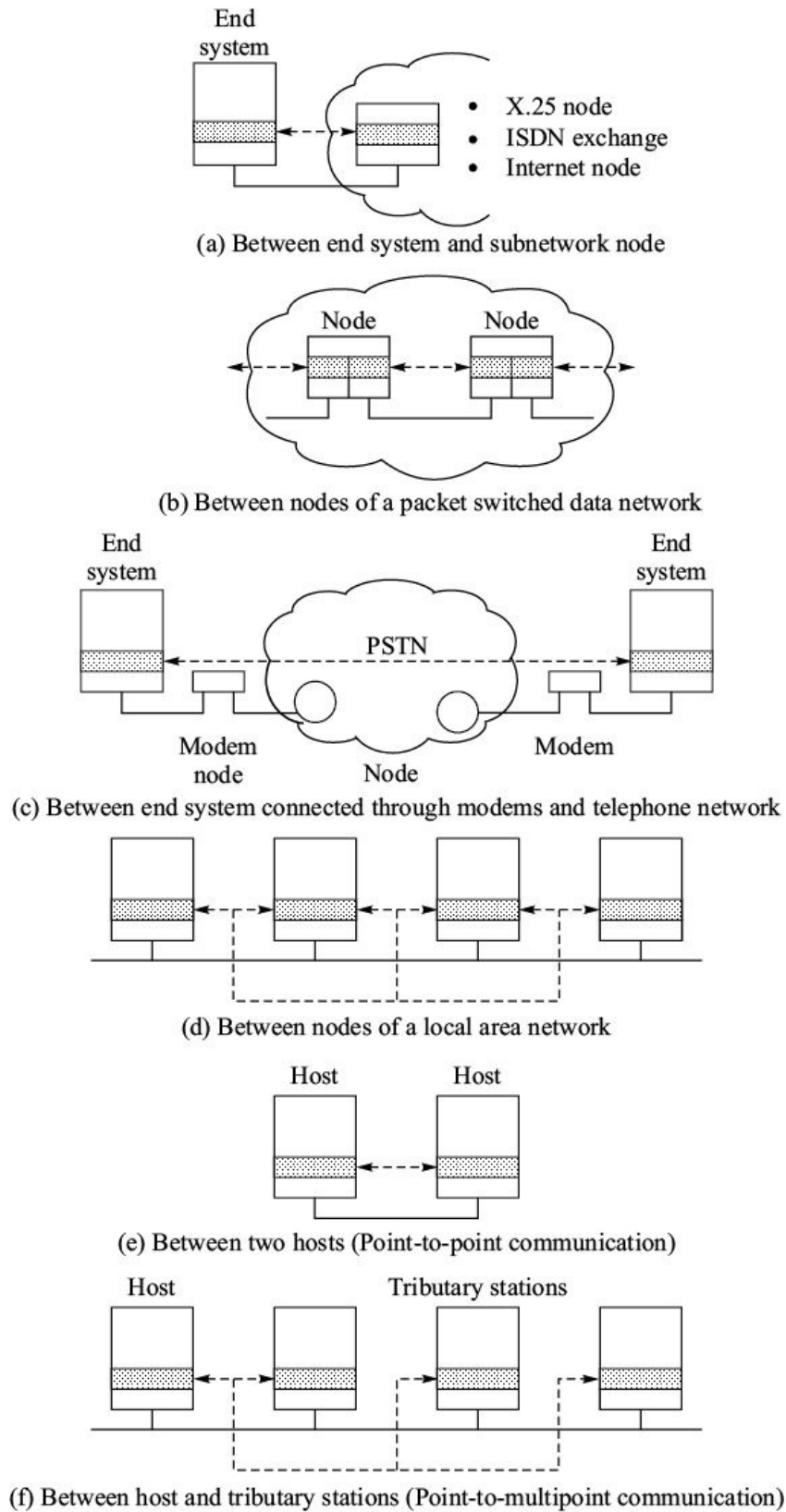


FIGURE 8.30 Application environment of data link protocols.

Point-to-point communication. Point-to-point communication is between two hosts interconnected directly (or through modems) (Figure 8.30e). Data link protocol is implemented between them.

Point-to-multipoint communication. Point-to-multipoint communication is between a host and tributary stations (Figure 8.30a). Data link protocol implemented among them ensures dialogue discipline so that several stations do not send frames simultaneously.

There are several data link protocols as mentioned in the beginning of this chapter. All the above applications are covered by one or the other protocol. We will discuss BISYNC, HDLC, LAPB, LAPD, and LAPM data link protocols in the next chapter. Protocols for local area networks will be discussed in Chapter 10. PPP is another data link protocol used between the routers of an IP network. We will discuss this protocol in Chapter 17.

APPENDIX

DATA LINK SERVICE PRIMITIVES

Data Link service primitives for connection-mode and connectionless-mode of data transfer are given in the following table (Table 8.2). Connection establishment and reset are confirmed services. Expedited data allows transfer of a data unit which is not flow controlled.

TABLE 8.2 Data Link Service Primitives	
Service	Primitive
Connection establishment	DL-CONNECT request
	DL-CONNECT indication
	DL-CONNECT response
	DL-CONNECT confirm
Connection release	DL-DISCONNECT request
	DL-DISCONNECT indication
Normal data transfer	DL-DATA request
	DL-DATA indication
Expedited data transfer	DL-EXPEDITED-DATA request
	DL-EXPEDITED-DATA indication
Connection reset	DL-RESET request
	DL-RESET indication
	DL-RESET response
	DL-RESET confirm
Error reporting	DL-ERROR REPORT indication

Data transfer (Connectionless-mode)	DL-UNITDATA request DL-UNITDATA indication
--	---

SUMMARY

Data link layer is the second layer of the OSI reference model. Together with the physical layer, it provides a link for reliable transfer of data bits over imperfect physical connection. It carries out error control, flow control and data link management functions. In the local area networks, the data link layer has another function—media access control.

The data link layer encapsulates the data bits into frames having a header and a trailer. Data link protocols specify the format of the frames and procedures for exchange of the frames. Data link protocols can be bit oriented or byte oriented.

There are two types of flow control mechanisms used by the data link protocols—stop-and-wait and sliding window. In stop-and-wait mechanism, one data frame is sent at a time and is acknowledged. Stop-and-wait mechanism is inefficient from the link utilization point of view.

In sliding window flow control mechanism, the sending end maintains a window containing several frames and it can send the data frames from the window without waiting for acknowledgements for individual data frames. By appropriately choosing the window size, link utilization of sliding window mechanism can reach up to 100%. Frame size is an important factor in determining the link utilization in presence of error. Large frame size reduces the link utilization notably.

Trailer of a data link frame contains check bits for error detection. For the flow integrity errors, the frames are given sequence numbers. Whenever an error is detected, request for retransmission of the frame is sent. In sliding window mechanism, the receiver can request retransmission of the frame selectively or of all the following frames as well.

Data link layer can provide connection-oriented or connectionless service. In connection-oriented service there are link establishment and link termination phases. In connectionless mode data transfer service, the data link need not be established and terminated. The data link entities are always in data transfer mode.

EXERCISES

1. A channel is operating at 4800 bps and the propagation delay is 20 ms.

What should be the minimum frame size for stop-and-wait flow control to get 50% link utilization efficiency?

2. If the frame size is 960 bits on a satellite channel operating at 960 kbps, what is the maximum link utilization for the following:
 - (a) Stop-and-wait flow control mechanism?
 - (b) Sliding window flow control with window size of 7?
 - (c) Sliding window flow control with window size of 127?
 - (d) Sliding window flow control with window size of 255?Assume propagation delay of 270 ms.
3. If window size is 7, and modulo 8 counting is used in sliding window flow control, show the exchange of frames and the data frames in the window at each of the following steps. Assume A needs to transmit 10 frames to B and TWA mode of communication is used.
 - (a) A sends data frames 0, 1, 2 and 3 to B.
 - (b) B acknowledges frames 0 and 1.
 - (c) A sends data frames 4, 5, and 6.
 - (d) B rejects frame 3 and asks A to retransmit from frame 3 onwards.
 - (e) A transmits frames 3, 4, 5, 6, and 7.
 - (f) B acknowledges frames 3, 4, 5, 6, and 7.
 - (g) A sends data frames 0 and 1.
 - (h) B acknowledges frames 0 and 1.
4. Stations A and B exchange frames using sliding window flow control. The acknowledgements are piggybacked on the data frames. Each station has five data frames to transmit. The window size is 3 and modulo 4 counting is used. Fill in the sequence numbers in the control field of the frames shown in Figure 8.31. Assume that there are no errors.
5. Station A continuously sends data frames to station C through an intermediate station B. A-B and B-C links are TWS. The link parameters are as follows:

Data rate between A and B : 1 Mbps
Frame size : 2500 bits
Propagation delay : 5×10^{-6} s/km
Distance between A and B : 5000 km
Distance between B and C : 25 km

Determine the data rate of the link between B and C such that buffers of B are not flooded, when

(a) the link between A and B uses sliding window protocol (window size 4), and the link between B and C is stop-and-wait.

(b) the link between A and B uses sliding window protocol (window size 127), and the link between B and C is stop-and-wait.

Assume there are no errors, acknowledgement size is negligible and acknowledgement is sent immediately on receipt of a frame.

6. In Figure 8.17, can a frame sent by A be lost without A and B being aware of it?
7. In Figure 8.27, if all the acknowledgements are lost instead of all the data frames what should be maximum window size?

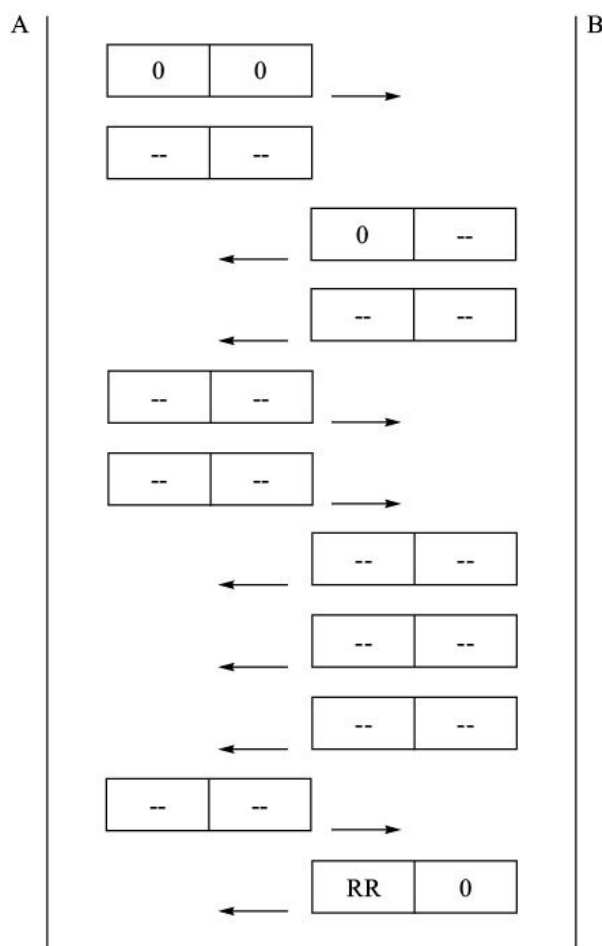


FIGURE E8.31.

8. Show that maximum window size for sliding window flow control using selective reject is limited to 2^{k-1} when k -bit sequence numbers are used.
9. What is the average number of transmissions required to send a frame of length 1200 bytes correctly, if the bit error rate is $1 \cdot 10^{-6}$?

10. If frame size is 1000 bytes and bit rate is 20 Mbps on a link 100 km long, determine the link utilization when the protocol used is:
- (a) Stop-and wait with NAK.
 - (b) Sliding window with selective reject. The window size is 10.
 - (c) Sliding window with go-back- N . The window size is 10.
- Assume velocity of propagation as $2 \cdot 10^8$ meters/s and BER of $4 \cdot 10^{-5}$.

1 We will examine in the next chapter, how a reply can be forced from the other entity.

9

Data Link Protocols

Having understood the flow control and error control as implemented in the data link layer, we can now look at their applications in industry standard data link protocols. We discuss in this chapter BISYNC, HDLC, LAP-B, LAP-D, and LAP-M data link protocols. Basic features, modes of operation, and frame structures of BISYNC, and HDLC are first described in detail. Protocol operation is illustrated with the help of some typical examples of data communication situations. These examples include two-way alternate and two-way simultaneous communication in asynchronous and synchronous modes. LAP-B, LAP-D, and LAP-M protocols are subsets of HDLC and are widely used by the industry. We examine their prime features. LLC protocol used in local area network is also based on HDLC. We leave discussion on LLC for Chapter 10 on local area networks. Certain liberties have been taken in the level of completeness of description of the protocols so that the overall picture is not clouded with too many details.

9.1 BINARY SYNCHRONOUS COMMUNICATION

DATA LINK PROTOCOL (BISYNC) BINARY

SYNCHRONOUS COMMUNICATION, BISYNC OR BSC IN SHORT, IS A DATA LINK LAYER PROTOCOL USED FOR COMMUNICATION BETWEEN IBM COMPUTERS AND TERMINALS. RELATED ISO STANDARDS ARE ISO 1745, ISO 2111, ISO 2628 AND ISO 2629. THE BASIC FEATURES OF BISYNC PROTOCOL ARE:

- It is a byte-oriented protocol.
- It supports three code sets—ASCII, EBCDIC, and transcode.
- It supports synchronous two-way alternate communication.

- It is applicable for point-to-point and point-to-multipoint communication.

There are several variations of BISYNC. The one we describe here uses numbered acknowledgements, unnumbered frames, and enquiry command if there is no reply from the slave.

It must be kept in mind that the communicating entities in the discussion that follows are the data link layer entities of the stations. Thus, when we use the term ‘station’, we mean the ‘data link layer’ of the station.

9.1.1 Communication Modes

BISYNC supports point-to-point and point-to-multipoint modes of communication. In these modes, there is master–slave relationship between the two communicating stations. The station which is to send a message is designated as the master and the station which receives messages and sends acknowledgements is designated as the slave.

Point-to-point communication. Point-to-point communication is between two hosts. These two stations contend for master status whenever they want to transmit a message. Alternatively, one of the stations can be designated as control station which delegates master or slave status to the other depending on which station is to transmit messages.

Point-to-multipoint communication. In point-to-multipoint communication, there is one host and several tributary stations. The host decides who will send or receive messages. All the messages are sent by or to the host.

Polling and selecting. In point-to-multipoint communication, the host invites a station to transmit data by *polling*. The polled station becomes the master station and controls further communication. After satisfactory completion of transmission it returns the control to the host.

If the host wants to transmit data to a station, it alerts the station to receive messages. The process is called *selecting*. The selected station takes over the status of slave station for this communication.

9.2 TRANSMISSION FRAME

In IBM terminology, BISYNC frames are called *blocks* but we will stick to the terminology being used in this book. BISYNC utilizes two categories of frame

types—supervisory and data frames (Figure 9.1). *Supervisory frames* are used for sending control information and are not protected against content errors. *Data frames* contain user data and contain error detection bytes.

9.2.1 Frame Format

Being a byte-oriented protocol, all the fields in BISYNC frames are of multiple bytes. BISYNC employs variable format and variable size frames. Therefore, several field identifiers/delimiters are required to indicate the presence of a field and to mark the end of a field.

Frame identifier. Frame identifier is two-character long and consists of synchronizing characters SYN SYN. It is always present in all types of the frames.

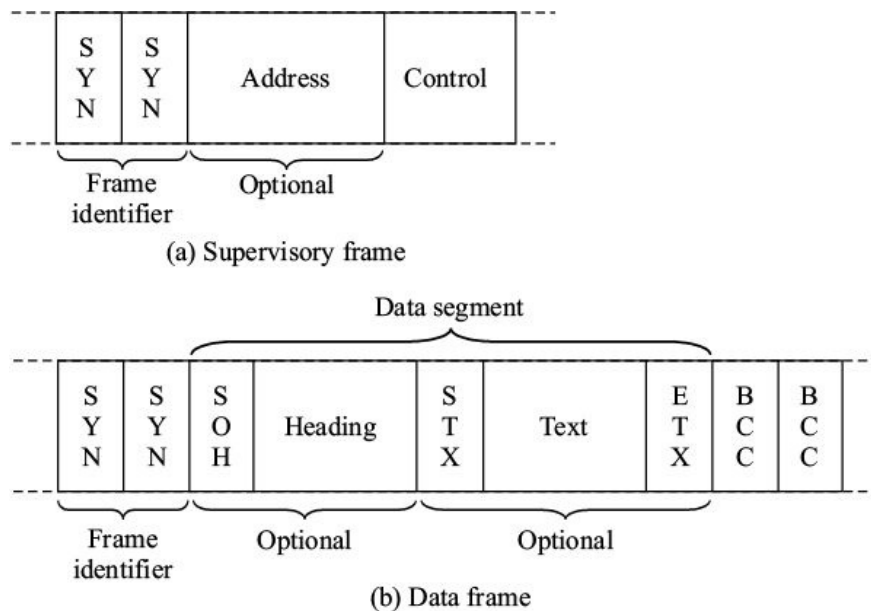


FIGURE 9.1 BISYNC frame formats.

Address field. The address field is optional and is present in the supervisory frames sent for polling and selecting in point-to-multipoint configuration. It is the address of the tributary station. In the poll frame, the address is in upper case; and in a select frame it is in lower case. The address field can be from one to seven bytes long.

Control field. The control field contains control byte(s) for link management, error, and flow control. The control bytes are characters from the character code set being used.

Heading field. The heading field contains information for higher layer functions

such as message identification, routing, device control and priority. It is optional and its presence is indicated by the field identifier SOH (Start of Heading). Length of the heading is variable and is delimited by the field identifier of the following field.

Text field. The text field contains user data bytes. It is of variable size and is optional. A field identifier STX (Start of Text) indicates its presence. A field delimiter ETX (End of Text) or ETB (End of Transmission Block) is provided to mark end of the field.

Block check characters (BCC). BCC field is present in the data frames and is used for content-error detection. For ASCII code, it is one character long. For EBCDIC and transcode, it is two characters long.

EXAMPLE 9.1 Compose the data segment consisting of SOH, heading, STX, Text and ETX for the following heading and text fields. Write the bit transmission sequence of the data segment assuming ASCII code with odd parity.

Heading : 45

Text : BSC

Solution	Data	segment	of	the	frame					
		S	4	5	S	B	S	C	E	
		O			T				T	
		H			X				X	
	LSB	1	1	0	1	0	0	1	1	1
		2	0	0	0	1	1	1	1	1
ASCII codes with odd parity		3	0	1	1	0	0	0	0	0
		4	0	0	0	0	0	0	0	0
		5	0	1	1	0	0	1	0	0
		6	0	1	1	0	0	0	0	0
		7	0	0	0	0	1	1	1	0
	Parity	8	0	0	1	0	1	1	0	1

Bit transmission sequence

0000000 00101100 10101101 01000000 01000011 11001011 11000010

9.2.2 Control Characters

BISYNC uses ten characters from the character code set for link management, acknowledgements and framing. Some two-character control sequences are also

defined as they are not readily available in the code set. Important control characters and their functions are described below.

SYN (Synchronous idle). Two SYN characters are used as frame identifier as mentioned earlier. This character is also used to fill inter-frame idle time.

SOH (Start of heading). It acts as field identifier for the heading field.

STX (Start of text). This control character is used as field identifier for the text field.

ETB (End of transmission block). This control character is used as the field delimiter for text field.

ETX (End of text). This control character is used as the field delimiter for the text field in the last frame of a message which may have been transmitted over several frames.

ENQ (Enquiry). ENQ is used during the link establishment phase to activate the other end. The host polls or selects a tributary station by sending ENQ. For communication between two hosts, ENQ is used for gaining master station status. It is also used during the data transfer phase to challenge the other station after timeout if no acknowledgement is received.

EOT (End of transmission). This control character signifies the end of a transmission. It results in relinquishment of the data link. It is also a negative response to a poll call.

ACK0/ACK1 (Positive acknowledgements). ACK is a positive acknowledgement of the received frame. On receipt of ACK, the sending end can dispatch the next frame. In BISYNC, ACK0 and ACK1 are used alternately to acknowledge the received frames.

NAK (Negative acknowledgement). When a frame is received with errors, NAK is sent back as request for retransmission of the frame.

DLE (Data line escape). This character is used to change the meaning of the following contiguous characters. For example, when DLE is combined with STX as DLE-STX, it indicates a transparent data sequence.

WACK (Wait and acknowledge). WACK is optional and is positive acknowledgement with a request for temporary suspension of further transmission.

9.2.3 Error and Flow Control

BISYNC uses the stop-and-wait mechanism of flow control. Wait-and-acknowledge (WACK) is used to indicate inability of the receiving end to accept more data frames. For error control, BISYNC uses alternating acknowledgements ACK0 and ACK1. Timers are provided for error recovery. Depending on the character code set being used, BISYNC employs block parity check or cyclic redundancy check for error detection.

ASCII. Parity bit (VRC) is used in each byte. In addition, an 8-bit LRC is provided in the one byte long BCC field.

EBCDIC. 16-bit code based on CRC-16 polynomial is used in the two-byte long BCC field.

Transcode. 12-bit code based on CRC-12 polynomial is used in the two 6-bit BCC bytes.

Accumulation of bytes that are covered by the BCC field for error detection, starts at the first SOH or STX character in a frame and all bytes up to (and including) ETX and ETB are covered (Figure 9.2).

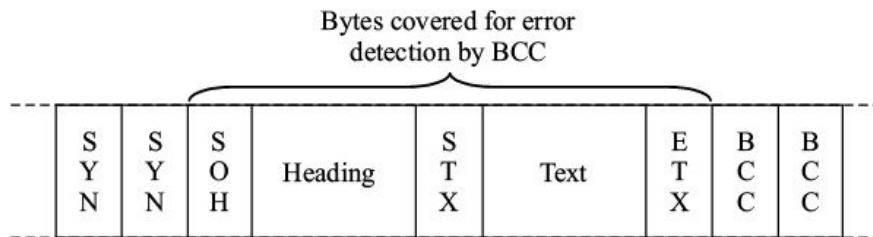


FIGURE 9.2 Span of error detection by BCC field.

EXAMPLE 9.2 For the data segment of Example 9.1, determine the BCC character. Show the complete bit structure of the frame in order of transmission of bits.

Solution Data segment of the frame

		S	S	S	4	5	S	B	S	C	E	B
		Y	Y	O			T				T	C
		N	N	H			X				X	C
LSB	1	0	0	1	0	1	0	0	1	1	1	0
	2	1	1	0	0	0	1	1	1	1	1	0
ASCII codes with odd parity	3	1	1	0	1	1	0	0	0	0	0	1
	4	0	0	0	0	0	0	0	0	0	0	1
	5	1	1	0	1	1	0	0	1	0	0	0
	6	0	0	0	1	1	0	0	0	0	0	1
	7	0	0	0	0	0	0	1	1	1	0	0
VRC	8	0	0	0	0	1	0	1	1	0	1	0

Bit transmission sequence

```
01101000 01101000 10000000 00101100 10101101 01000000 01000011
11001011 11000010 11000001 00110100
```

9.2.4 Transparency

Transparency is achieved in BISYNC by inserting the DLE control character before the text field identifier STX. The DLE STX sequence effectively instructs the receiving end to treat all characters in the text field as data bytes even if they are control characters. The transparent mode of the text field is terminated by DLE ETX (or DLE ETB) as shown in Figure 9.3.

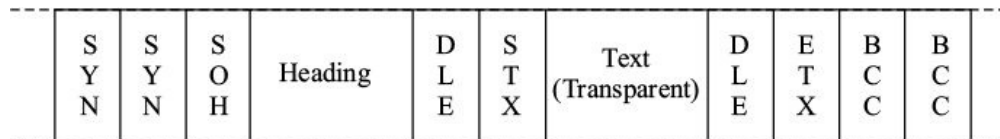


FIGURE 9.3 Transparency in BISYNC.

If a byte representing DLE itself appears in the text field, the sending end stuffs another DLE into the byte stream to indicate to the receiving end that this is not the control character DLE which appears before ETX (or ETB) in the transparent mode of the text field. When the receiver detects two consecutive DLEs, it discards one and considers the other as part of the text field.

EXAMPLE 9.3 Show the structure of a data frame containing user message which is (a) ETB character;

(b) DLE character.

Solution

(a) SYN SYN DLE STX ETB DLE ETX BCC BCC
 (b)
 SYN SYN DLE STX DLE DLE DLE ETX BCC I

9.3 PROTOCOL OPERATION

The protocol operation can be divided into three phases:

- Data link establishment phase
- Information transfer phase
- Data link termination phase.

Establishment of a data link involves exchange of certain supervisory frames between the data link entities to ensure their readiness to transmit and receive frames containing data. During this phase, master or slave status of a station is also decided. After establishing the data link, the master station sends the data frames containing user data bytes. The slave station sends acknowledgements using the supervisory frames. The master station initiates termination of link, and after the link is terminated, the stations lose their master and slave status. We will discuss these operations in detail below with two examples, one for point-to-point communication and the other for point-to-multipoint communication.

9.3.1 Point-to-Point Communication Point-to-point communication involves three phases namely, link establishment, data transfer, and link termination phase.

Link establishment phase. Figure 9.4 shows the supervisory frames exchanged during link establishment phase.

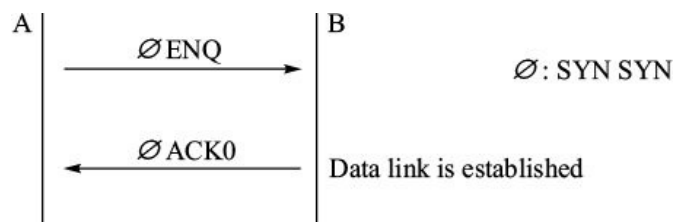


FIGURE 9.4 Link establishment in point-to-point communication.

- A wants to send a message and therefore bids for master status by sending SYN SYN ENQ.
- B replies SYN SYN ACK0 if it is ready to receive. If it is not ready to receive, it replies SYN SYN NAK and in this case A needs to retry.
- If an invalid reply is received or if there is no reply, A transmits SYN SYN EOT to terminate its first attempt and then it retries.
- Since it is possible that both the stations may bid together, contention is resolved by keeping different timeouts for retry.

Data transfer phase. In data transfer phase, data frames are sent by the master station. The slave states reply with

- alternating positive acknowledgements ACK0 and ACK1, or
- negative acknowledgement NAK if errors are detected or
- WACK to temporarily stop flow of frames.

ENQ is used for challenging after timeout. Figure 9.5 illustrates a typical example of exchange of frames during data transfer phase between stations A and B. A is the master station and B is the slave station.

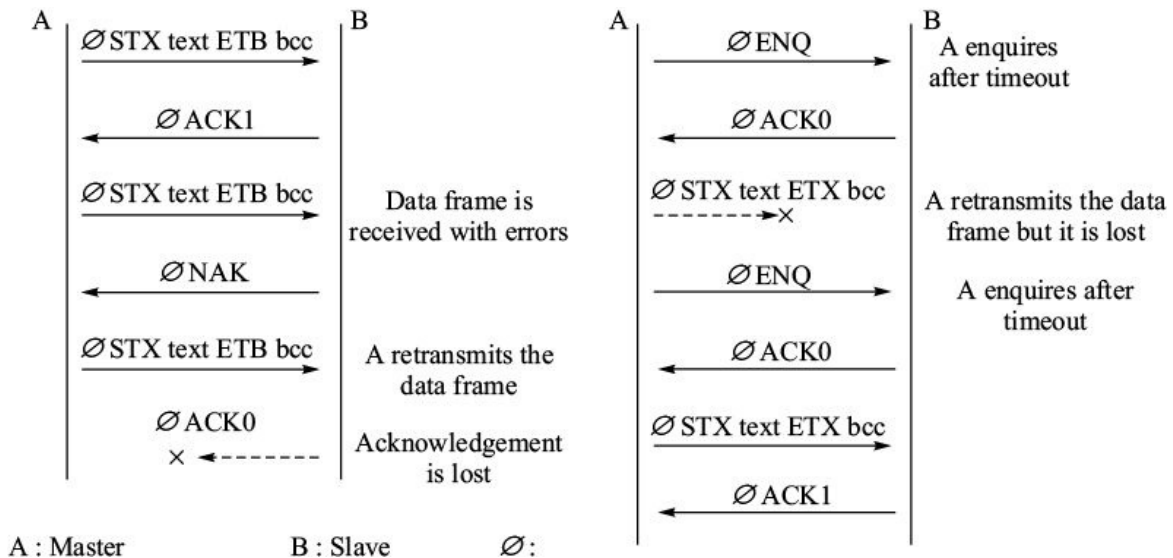


FIGURE 9.5 Data transfer phase of point-to-point communication.

Link termination. Termination is initiated by the master station following reception of a positive acknowledgement to the last data frame. It is effected by sending the supervisory frame SYN SYN EOT. Upon termination of the link, the master station loses its status and control of the link returns to the host. In point-

to-point communication, the link becomes available to both the stations to contend for.

9.3.2 Point-to-Multipoint Communication In point-to-multipoint link, the host activates one tributary station at a time by sending a poll or a select. Figure 9.6 shows the frames which are exchanged during polling and selecting.

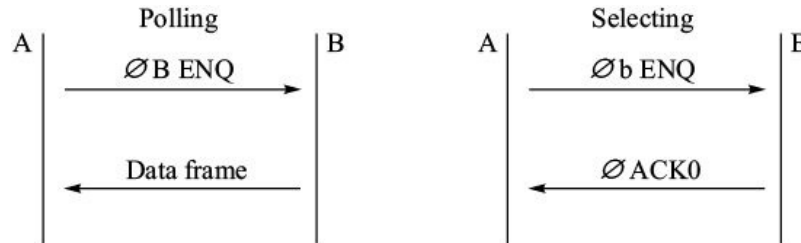


Figure 9.6 Polling and selecting in point-to-multipoint communication.

- The host sends SYN SYN (address) ENQ to the tributary station. The poll/select supervisory frame is received by all the tributary stations but it is responded only by the addressed tributary station.
- Poll call and select call are differentiated by the address field. The address is in upper case for polling and in lower case for selecting.
- If the polled tributary station B has data to send, it replies with a data frame and assumes master status. Else, it terminates the link by sending SYN SYN EOT.
- In the case of select call, reply of the tributary station B is SYN SYN ACK0 if it is ready to receive a data frame, or SYN SYN NAK if it is not ready to receive any data frame.

Figure 9.7 shows an example of data link dialogue on a point-to-multipoint link. In this example, all the three phases of data link operation are illustrated.

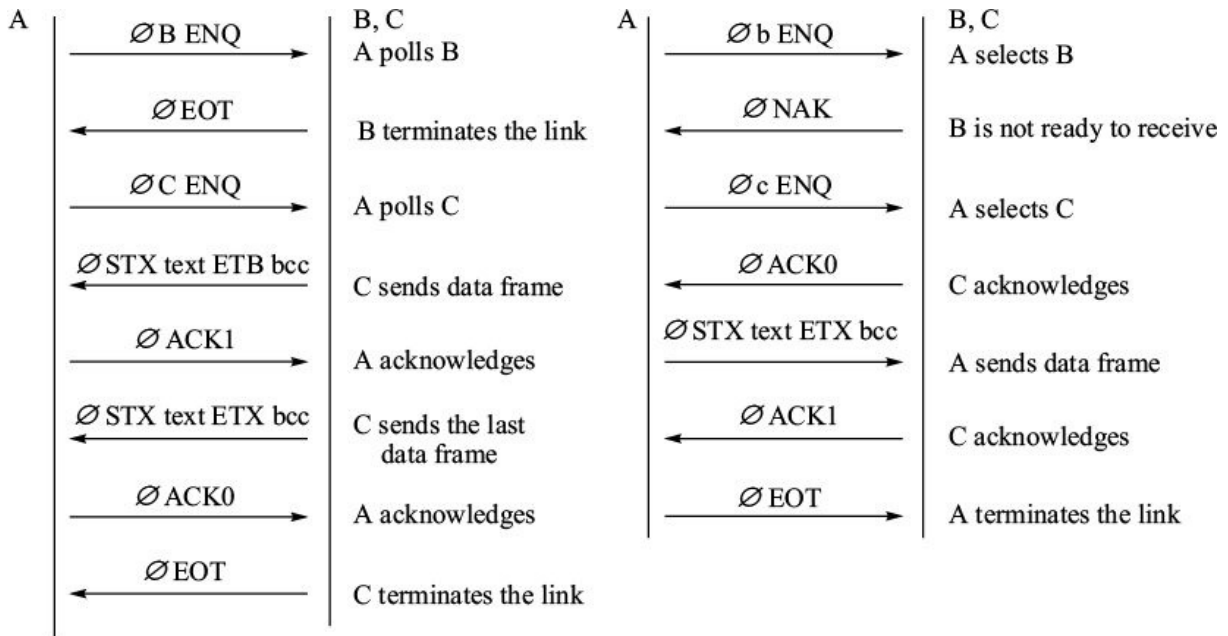


FIGURE 9.7 Point-to-multipoint communication.

EXAMPLE 9.4 The following point-to-point exchange of BISYNC frames takes place without any errors. Only important fields that identify the type of frame have been shown. Rewrite the exchange of frames if the second data frame had (a) failed the error check; (b) been lost during transmission.

A	ENQ		STX-ETB		STX-ETB		EOT
B		ACK0		ACK1		ACK0	

Solution

(a)

A	ENQ		STX-ETB		STX-ETB	
B		ACK0		ACK1		NAK

A	STX-ETB		EOT
B		ACK0	

(b)

A	ENQ		STX-ETB		STX-ETB	Timeout
B		ACK0		ACK1		

A	ENQ		STX-ETB		EOT
B		ACK1		ACK0	

9.3.3 Limitations of BISYNC Protocol Layered architecture concept is based on independence of functions and their distinct implementation. Since BISYNC was not originally designed with hierarchical functional layers in mind, it is awkward at places. For example, the heading field is a higher layer function but the field is defined at the data link layer. As regards data link layer functions, it meets the basic requirements but suffers from some inherent limitations:

- Supervisory frames are not protected against errors.
- As a result of using characters from the code set, natural transparency is impossible. Therefore, transparency is achieved only as a special case by invoking transparent text mode.
- Communication is always two-way alternate even if a full duplex line is used. Its effect pervades all hierarchical layers.
- Link utilization is poor due to inherent limitations of the stop-and-wait flow control mechanism.
- It supports synchronous communication only.

9.4 HIGH LEVEL DATA LINK CONTROL (HDLC)

HDLC is a bit-oriented data link control protocol which satisfies a wide variety of data link control requirements as follows:

- Point-to-point and point-to-multipoint communication.
- Two-way simultaneous communication over full duplex circuits.
- Two-way alternate communication over half duplex or full duplex circuits.
- Synchronous and asynchronous communication.
- Communication between equal stations and between host and remote stations.
- Full data transparency.

The ISO standards for HDLC are ISO 3309, ISO 4335, ISO 6159, and ISO 6256.

9.4.1 Types of Stations

To make HDLC protocol applicable to various possible network configurations, three types of stations have been defined:

- Primary station
- Secondary station
- Combined station.

A primary station has the responsibility of data link management. A secondary station operates under the control of a primary station. A combined station can act both as a primary and a secondary station.

When communication is between a primary station and a secondary station, the primary station has the responsibility of activating, maintaining, and disconnecting the data link. All the frames sent by a primary station are called *commands* and the frames sent by a secondary station are called *responses* (Figure 9.8).

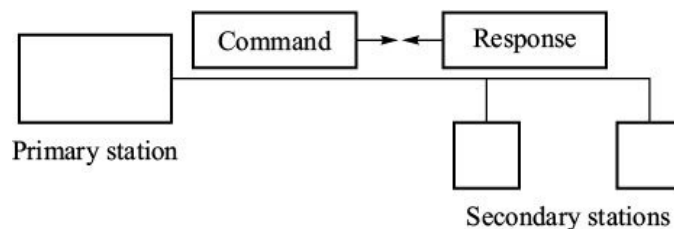


Figure 9.8 Primary and secondary stations.

Communication can be between two logical equal status computers also, in which case they are designated as combined stations and can send and receive both, commands and responses (Figure 9.9). When a combined station sends a command, the other responds.

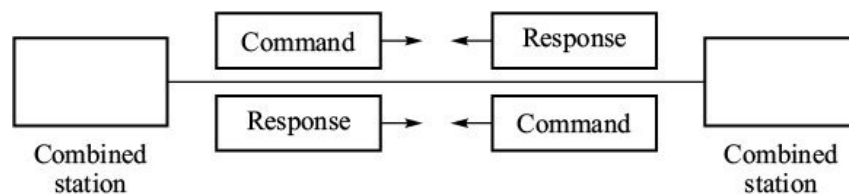


Figure 9.9 Combined stations.

9.4.2 Modes of Operation

There are six modes of operation of HDLC protocol. The first three modes are for data transfer and the last three modes refer to the states before and after the data transfer phase. The modes are as follows:

- Normal Response Mode (NRM)¹
- Asynchronous Response Mode (ARM)
- Asynchronous Balanced Mode (ABM)
- Normal Disconnected Mode (NDM)
- Asynchronous Disconnected Mode (ADM)
- Initialization Mode (IM).

The normal and asynchronous response modes of operation provide an unbalanced type of data transfer capability between logical unequal stations—one primary and other secondary station. The asynchronous balanced mode of operation is for logical equal or combined stations.

Normal response mode (NRM). In the normal response mode the primary station controls the overall link management function. A secondary station can send a frame only as a result of receiving explicit permission to do so from the primary station. It is a synchronous mode of communication. The normal response mode is applicable to point-to-point and point-to-multipoint configurations.

Asynchronous response mode (ARM). It is an asynchronous mode of communication between a primary and a secondary station. The secondary station can send a frame without any explicit permission from the primary station. The link management function is the responsibility of the primary station. The asynchronous response mode is applicable to both point-to-point and point-to-multipoint configurations. In multipoint environment, however, only one secondary station can be active at a time and other secondary stations must be kept in disconnected mode.

Asynchronous balanced mode (ABM). Asynchronous balanced mode is applicable to point-to-point communication between two combined stations. Both the stations are capable of link management function when required. They can issue commands and force a response from the other station if required. Being in asynchronous communication mode, a station can send a frame without any explicit permission from the other station.

Normal disconnected mode (NDM). In the disconnected modes, the stations are logically disconnected. They need to exchange mode-setting commands to come out of the disconnected mode. When in normal disconnected mode, a secondary station is activated by a mode-setting command for normal response

mode from the primary station.

Asynchronous disconnected mode (ADM). In this mode, the stations enter asynchronous response mode or asynchronous balanced mode when the corresponding mode-setting command is exchanged. A secondary station in asynchronous disconnected mode can request for a mode-setting command from the primary station in order to establish data transfer mode.

Initialization mode (IM). In the initialization mode, operational parameters are exchanged. It is invoked when a primary station concludes that the secondary station is operating abnormally and needs its operational parameters corrected. Also, a secondary station can request the primary station for initialization mode if it is unable to function properly.

Transition from one mode to another is effected by the primary station by giving mode-setting commands. We will study these commands shortly. Figure 9.10 shows how transition of the logical modes takes place when appropriate commands are given.

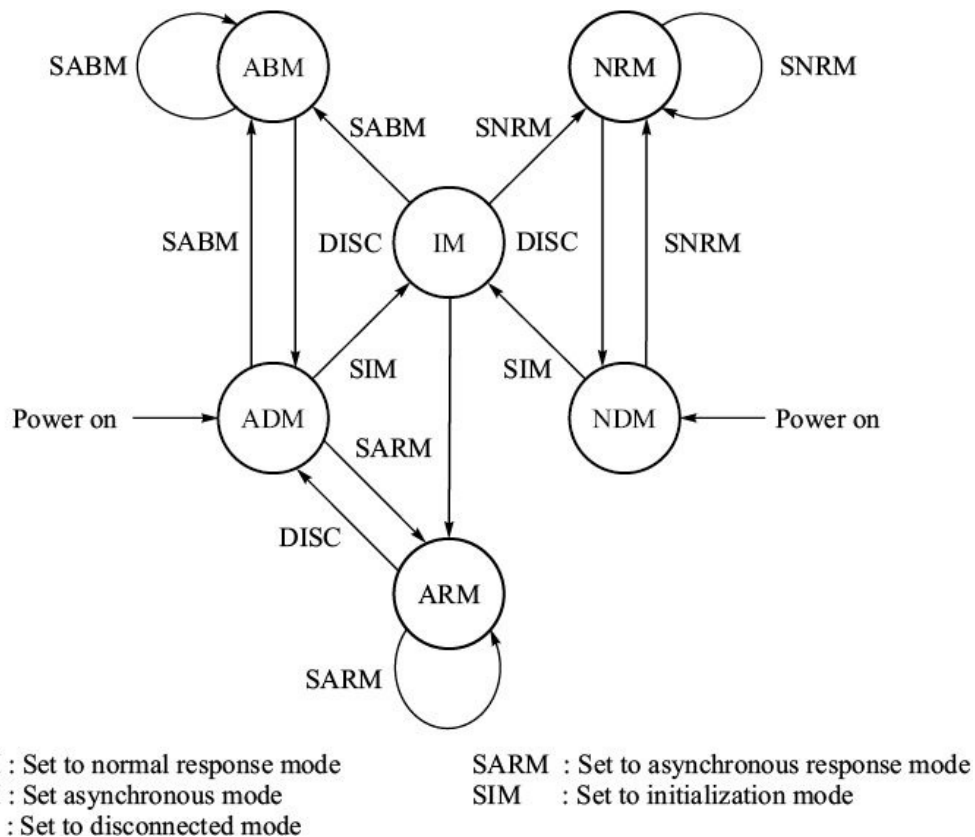


Figure 9.10 Mode transition in HDLC.

Note that, when switched on, a station is in the disconnected mode. When it

receives a mode-setting command from a primary station, it enters the mode corresponding to the command.

9.5 FLOW AND ERROR CONTROL IN HDLC

Flow control. HDLC utilizes the sliding window flow control mechanism. The maximum window size can be either 7 or 127. All the data frames and acknowledgements are numbered as required for the sliding window flow control mechanism. The receiving end sends acknowledgement in the form of RR— N (Receive ready for frame N , frames up to $N - 1$ acknowledged). When the receiver is not ready to receive more data frames, it sends RNR— N (Receive not ready for frame N , frames up to $N - 1$ acknowledged).

Error control. Error control is based on retransmission of frames received with errors. Retransmission is requested by sending a reject (REJ— N) or a selective reject (SREJ— N). Error detection is carried out using a 16-bit CRC code generated using ITU-T V.41 polynomial $x^{16} + x^{12} + x^5 + 1$ (10001000000100001).

For recovery purposes, the following parameters are specified:

- $T1$: Timeout for retransmission of a frame. Its typical value is 3 seconds.
- $T3$: Timeout for completion of link initialization. Its typical value is 90 seconds.
- $N1$: Maximum number of transmissions and retransmissions before declaring link down. Its typical value is 20.

9.6 FRAMING IN HDLC

There are three types of HDLC frames:

- Information transfer frame (I-frame)
- Supervisory frame (S-frame)
- Unnumbered frame (U-frame).

The I-frame is used for transporting user data. It also carries acknowledgement of the received frames. The S-frame does not have a data field and is used for carrying acknowledgements and requests for retransmission. As explained earlier, an acknowledgement (RR) can be sent either on a supervisory frame or piggybacked on an I-frame. On the other hand, RNR, REJ, and SREJ are sent only through a supervisory frame. A U-frame, as the name suggests, does not have any sequence number. It is used for link establishment, termination, mode setting, and other control functions.

9.6.1 Frame Formats

HDLC utilizes two types of frame formats as shown in Figure 9.11. The I-frame has a format as shown in Figure 9.11a. The S-frame and U-frame have formats as shown in Figure 9.11b. The frame format is fixed. Except the information field, all other fields have fixed sizes. Two frame identifier/delimiters are required, one at the start of the frame and the other at the end. The frame is transmitted from left to right and the low-order bit is transmitted first.

Flag. The flag is a unique 8-bit word pattern (01111110) which identifies the start and end of each frame. It is also used for filling the idle time between consecutive frames (Figure 9.12).

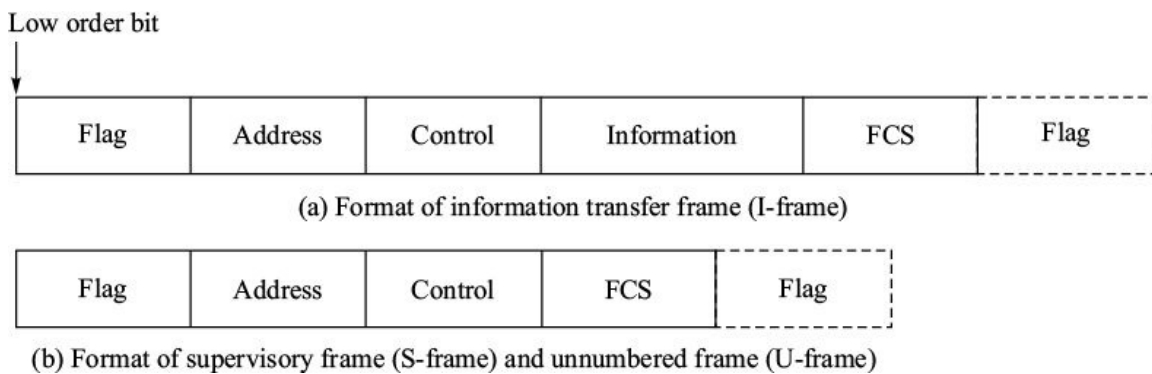


Figure 9.11 Types and formats of HDLC frames.

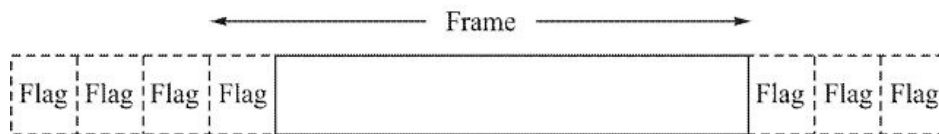


Figure 9.12 Flag transmission to fill idle time between consecutive frames.

Address field. The address field always contains the address of the secondary station whether a frame is being transmitted by a primary or a secondary station (Figure 9.13). The address field identifies whether the frame is a command or

the response. If a frame contains address of the receiver, it implies that the receiver is a secondary station and therefore, the frame is a command. If the frame contains address of the sender, it implies that the sender is a secondary station. Therefore, the frame is a response. A combined station acts both as primary station and secondary station during the dialogue. Depending on whether it is issuing a command or response, it puts the receiver's or its own address in the address field.

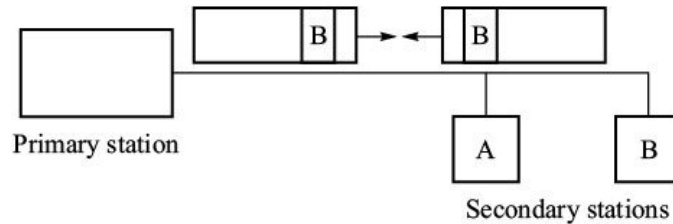


FIGURE 9.13 Address field.

The address field consists of 8 bits, giving it a capability of 256 different addresses. Greater than 256 address capability is also possible as explained later. The all 1s address is specified as global address. It addresses all the secondary stations simultaneously.

Control field. The control field consists of 8 bits. It identifies the type of HDLC frame. It carries the sequence number of the frame, acknowledgements, request for retransmission, and other control commands and responses. It can be extended to 16 bits to accommodate 7 bit sequence numbering scheme.

Information field. The information field has variable size and can consist of any number of bits. The maximum number of the bits in the information field is not specified. It contains the user data and is completely transparent.

Frame check sequence (FCS). Frame check sequence is a 16-bit CRC code for detection of errors in the address, control, and information fields (Figure 9.14).

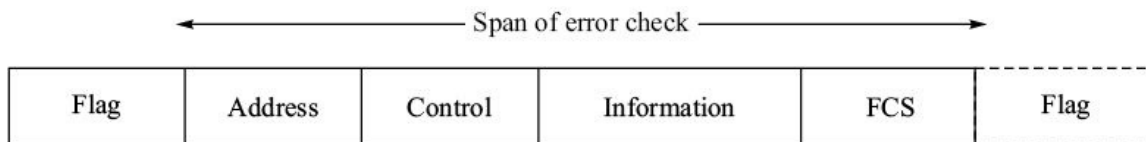


FIGURE 9.14 Span of error check by the FCS field.

EXAMPLE 9.5 Shown below is a bit sequence containing an HDLC frame. Identify various fields of the frame.

```
01111110 01111110 10100011 01100010 01110000 11000011 010
10101011 01111110 01111110
```

Solution To identify the various fields we perform the following steps:

1. Identify the start and end flags. Start flag is just before a non-flag field and the end-flag is just after a non-flag field.
2. After the start flag, we have one octet each of the address and control fields.
3. Before the end flag, we have two octets of the FCS.
4. The remaining bits comprise the information field.

Start flag	Address	Control	← Information →
01111110	10100011	01100010	01110000 11000011 01010101 10101011
End flag			
01111110			

9.6.2 Control Field of HDLC Frames

The control field of the HDLC frames identifies the type of frame, carries acknowledgements, sequence numbers, unnumbered commands/responses, and the P/F bit. Figures 9.15 to 9.17 show the structure of the control field of I-, S- and U-frames.

Control field of I-frame. Figure 9.15 shows the control field of I-frame.

- The first bit of the control field is always 0 in an I-frame.
- The next three bits are the sequence number N(S) of the frame.
- The fifth bit is the Poll/Final (P/F) bit. Its use is explained later.
- The last three bits are the sequence number N(R) of the acknowledgement RR which is piggybacked on the I-frame.

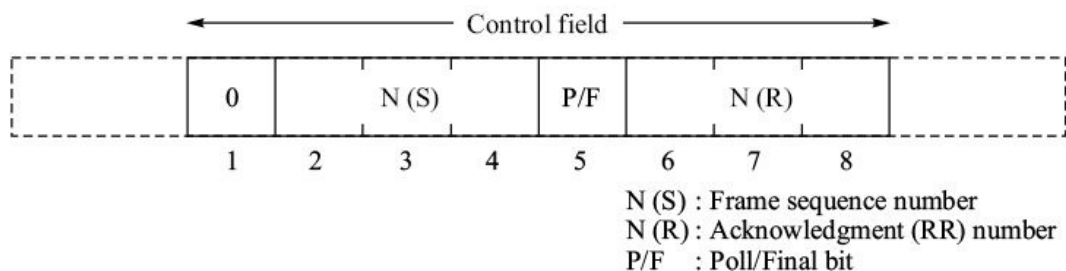


Figure 9.15 Control field of an information frame (I-frame).

Control field of S-frame. Figure 9.16 shows the control field of S-frame.

- S-frame is identified by the first two bits of the control field. These two bits are always 10 in an S-frame.
- The next two bits SS are used for indicating type of acknowledgement, Receive Ready (RR), Receive Not Ready (RNR), Reject (REJ), and Selective Reject (SREJ).
- The fifth bit is the poll/final bit, explained later.
- The last three bits are the sequence number associated with the acknowledgement RR, RNR, REJ, or SREJ as indicated by the SS bits of the control field.

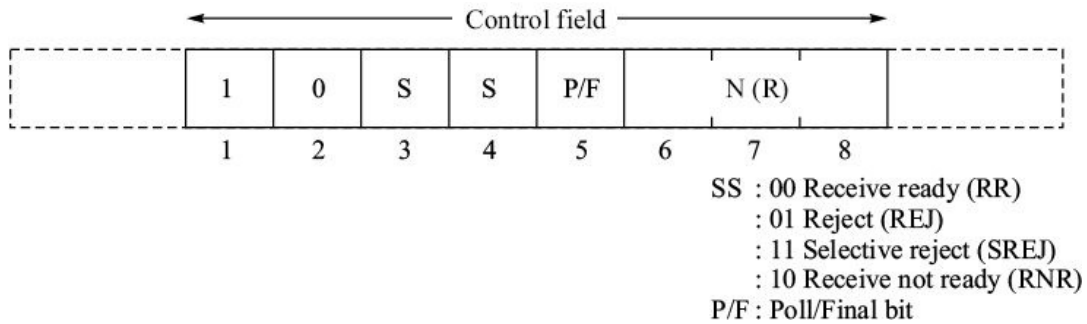


Figure 9.16 Control field of a supervisory frame (S-frame).

Control field of U-frame. Figure 9.17 shows the control field of U-frame.

- The first two bits of the control field are always 11 in an unnumbered frame.
- The fifth bit is the poll/final bit, explained later.
- The remaining five bits are called *modifier bits* (M-bits). They specify the control function. Table 9.1 gives the codes of the control field of a U-frame for the various commands and responses.

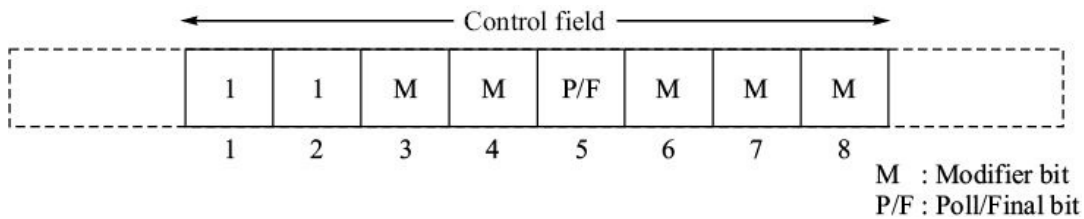


Figure 9.17 Control field of an unnumbered frame (U-frame).

TABLE 9.1 Control Field of U-Frame

Control field

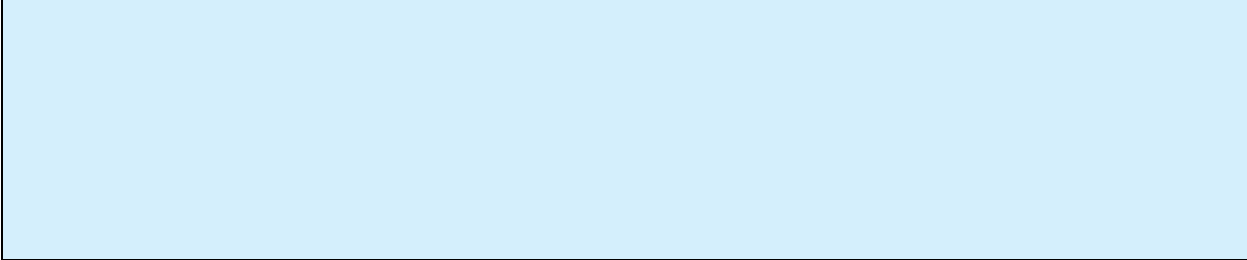
U-Frame commands

1 2 3 4 5 6 7 8

						0	1	
						0	0	
1	1	0	0	P	0	0	0	
1	1	1	1	P	0	0	0	
1	1	1	1	P	1	1	1	Set Normal Response Mode (SNRM) Set Asynchronous Response Mode (SARM) Set
1	1	1	1	P	0	1	0	Asynchronous Balanced Mode (SABM) Set NRM Extended (SNRME)
1	1	1	1	P	0	1	0	Set ARM Extended (SARME)
1	1	1	1	P	1	0	0	Set ABM Extended (SABME)
1	1	1	0	P	0	1	0	Set Initialization Mode (IM) Disconnect (DISC)
1	1	0	0	P	0	0	0	Unnumbered Information (UI)
1	1	0	0	P	0	0	0	Unnumbered Poll (UP)
1	1	0	0	P	1	0	1	Reset (RSET)
1	1	1	1	P	0	0	1	Exchange Identification (XID) Test
1	1	1	1	P	1	1	1	Non-reserved commands
1	1	0	0	P	1			
1	1	0	1	P	0	*	*	

U-Frame responses

						1	0	
						0	0	
1	1	0	0	F	1	0	0	
1	1	1	1	F	0	0	0	Unnumbered Acknowledgement
1	1	1	0	F	0	0	0	Disconnected Mode (DM)
1	1	0	0	F	0	0	1	Request Initialization Mode (RIM) Unnumbered Information (UI)
1	1	1	0	F	0	1	0	Frame Reject Response (FRMR) Exchange Identification (XID) Request Disconnect (RD)
1	1	1	1	F	1	1	1	Test
1	1	0	0	F	0			Non-reserved responses
1	1	0	0	F	1			
1	1	0	1	F	0	*	*	



The primary station sends U-frame commands and the secondary station sends U-frame responses. These commands and responses are described briefly as follows: *Mode-setting commands* (*SNRM*, *SARM*, *SABM*, *SNRME*, *SARME*, *SABME*). *SNRM*, *SARM*, and *SABM* are the mode-setting commands. A secondary or a combined station sets itself in the mode corresponding to the received command. *SNRME*, *SARME*, and *SABME* are used when an extended frame numbering format is used to accommodate a window size of more than 7.

Set initialization mode command (IM). This command is used to establish the initialization mode of operation during which operational parameters are exchanged.

Disconnect command (DISC). It is used to terminate a previously established link and to cause the stations to assume the disconnected mode.

Unnumbered information (UI) command/response. The unnumbered information frames are used to exchange miscellaneous information such as hourly reports, periodic time checks, *etc.* These frames are not acknowledged.

Unnumbered poll (UP) command. The unnumbered poll command is used to solicit a response frame from a station without regard to sequencing.

Reset command (RSET). Reset command is used for recovery purposes and resets sequence numbers *N(S)* and *N(R)* in one direction of transmission to zero.

Exchange identification (XID). *XID* command and response are used to request and report identity of a station and optionally, its operational parameters.

Test command/response. Test command and response are used for testing the data link control.

Unnumbered acknowledgement (UA). The *UA* response is used to acknowledge receipt and execution of a U-frame command.

Disconnected mode (DM) response. It is sent to the primary station in response to a mode-setting command to indicate that mode-setting action has not been executed. It is also used as request to the primary station to send the mode-setting command.

Request initialization mode (RIM). This response is used for requesting the primary station to establish the initialization mode.

Frame reject response (FRMR). FRMR response is used to report a condition which is not correctable by retransmission of frames, *e.g.* receipt of invalid $N(R)$.

Request disconnect (RD). The RD response is used for requesting the primary station to disconnect the link.

EXAMPLE 9.6 Identify the type of frame from the control field given below. Also identify the sub-fields within the control field. The low order bit is on the left hand side.

- (a) 01010111
- (b) 10111010
- (c) 11000000

Solution

- (a) I-frame, $N(S) = 101$, $P/F = 0$, $N(R) = 111$
- (b) S-frame, SREJ, $P/F = 1$, $N(R) = 010$
- (c) U-frame, unnumbered information, $P/F = 0$

9.6.3 Poll/Final (P/F) Bit

Fifth bit of the control field of an HDLC frame is called Poll/Final (P/F) bit. It is called P bit when the frame is a command, *i.e.* the frame is being sent by a primary station. It is called F bit when the frame is a response, *i.e.* the frame is being sent by a secondary station. As explained earlier, a frame is identified as a command or response by the address field.

P/F bit in NRM. In normal response mode, a primary station invites a secondary station to transmit a frame by setting the P of the control field to 1. Having received the invitation to transmit, the secondary station sends frames and finally returns the permission by explicitly marking its last frame. The secondary station utilizes the F bit of the control field in the last frame. It sets this bit to 1.

P/F bit in ARM/ABM. In the asynchronous modes of data transfer, ARM and ABM, the P bit is used to solicit response from a secondary/combined station. When a frame with P set to 1 is received, the receiving station responds with the F bit set to 1 at the earliest opportunity.

Once a command with P bit set to 1 is sent, the primary station awaits a response with F set to 1 and does not send another frame with the P bit set to 1

until it is established that such response will not be forthcoming. This may happen if either the command or the response is lost.

9.7 TRANSPARENCY IN HDLC

In HDLC, transparency is achieved by ensuring that the unique flag sequence (01111110) does not occur in the address, control, information, and FCS fields. A technique called *zero stuffing* is used. At the sending end an extra 0 is inserted after five contiguous 1s that occur anywhere after the opening flag and before the closing flag. At the receiving end, the extra 0 bit following five contiguous 1s is deleted.

The steps involved in assembling an HDLC frame are given below. Note that zero stuffing is performed before the flags are appended to the frame. Therefore, any sequence of bits (including the flag sequence) can be transmitted in the address, control, information, and FCS fields without affecting the data link control operation.

- Build address and control fields and append to the information field
- Generate CRC
- Carry out zero stuffing
- Append flags

At the receiving end, the above steps are carried out in reverse order.

- Identify flags and delete them
- Remove the stuffed zeros
- Compute and check FCS
- Check address and control fields

EXAMPLE 9.7 The following bit stream represents an HDLC frame from the

address field to the FCS field. Construct the full HDLC frame.

10111110111011110111111111111111000000

Solution

1. Inserting an extra zero after every five consecutive ones, we get

```
1 0 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0
          ↑           ↑           ↑           ↑           ↑
0 0 0 0 0
```

2. Constructing the complete frame by adding flags, we have

01111110 1011111010111011111001111101111101111101000000 01111110

EXAMPLE 9.8 Identify various fields of the following HDLC frame.

01111110101111101011101111100111110111110111110100000001111110

Solution

1. Removing the flags, we get

1011111010111011111001111101111101111101000000

2. Removing extra zeros after every five consecutive ones, we obtain

10111111 01110111 110111111 11111111 11000000

3. Address field : 10111111

Control field : 01110111

FCS : 11111111 11000000

Information field : 110111111

9.8 HDLC PROTOCOL OPERATION

Having reviewed the basic features of the HDLC protocol, let us now examine its operation. Typical data communication situations include point-to-point and point-to-multipoint links in various data transfer operating modes, namely, NRM, ARM, and ABM. We will examine operation of the protocol with the help of some examples of these modes of data transfer. These examples serve to illustrate the operation of the protocol in typical situations but it must be noted that illustrated situations do not cover all possibilities.

In these examples, we will consider the following three phases of operation:

- Link establishment phase
- Data transfer phase
- Link disconnection phase.

Link establishment is always initiated by the primary station by sending a mode-setting command with P bit set to 1. The link is established when an unnumbered acknowledgement with F bit set to 1 is received from the secondary station.

Link disconnection is also carried out in the same manner by the primary station. It sends an unnumbered disconnect command with P bit set to 1. The secondary station responds with unnumbered acknowledgement having F bit set to 1. It is ensured by the primary station that all I-frames have been acknowledged and all acknowledgements have reached the destination before the link disconnection is initiated. In asynchronous balanced mode, either of the two combined stations can establish and disconnect the link.

The following five-symbol (ABCDE) code is used for representing various data link frames: A = Address of the secondary station in the address field of HDLC frame.

B = Type of frame—I (Information), S (Supervisory), U (Unnumbered).

C = Sequence number N(S) of the I-frame, if it is an I-frame or type of acknowledgement (RR, RNR, REJ, SREJ) if it is an S-frame or link management command if it is a U-frame.

D = Sequence number N(R) of the acknowledgement if it is an I-or S-frame.

E = Poll/final bit. It is shown only when it is 1. In commands, it is written as P and in responses as F. When it is not shown, it implies that P/F bit is 0.

For example,

B I 2 5 P sent by A I-frame having sequence number 2, sent to secondary station B, piggybacked RR number is 5, the P/F bit is P bit and it is set to 1.

B I 4 2 sent by B I-frame having sequence number 4, sent by secondary station B, piggybacked RR number is 2, the P/F bit is F bit and it is set to 0.

B S RR 1 sent by B S-frame carrying RR1, the P/F bit is F bit and it is set to 0.

9.8.1 Normal Response Mode, Point-to-Point

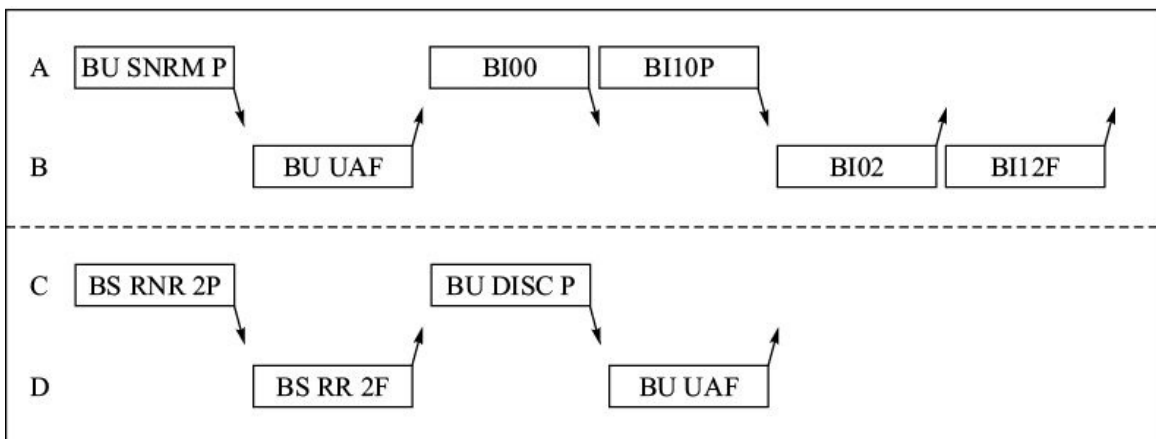
Figures 9.18 to 9.20 illustrate the operation of the protocol in normal response mode for point-to-point communication over Two Way Alternate (TWA) and Two Way Simultaneous (TWS) links.

TWA point-to-point communication without errors. Figure 9.18 shows operation of the protocol over TWA link without errors.

- The primary station A sends mode setting command SNRM with P bit set to 1.
- The mode setting command is acknowledged with UA (Unnumbered Acknowledgement) response and with F bit set to 1.
- Being TWA operation, only one station transmits at a time. B being the secondary station, can initiate transmission only after it receives explicit permission in the form of P bit set to 1 from the primary station A.
- Link is disconnected by A by sending DISC command which is acknowledged with UA response.
- Before the link is disconnected, A ensures that

–all the frames have been acknowledged, –all acknowledgements have been received, and –the secondary station is barred from sending more I-frames.

In Figure 9.18, station A sends RNR2 to acknowledge I-frames received from B and to indicate its unwillingness to accept more I-frames. Station A ensures that this message reaches station B by sending P bit set to 1. The only alternative left for B is to respond with RR2 with F bit set to 1. On receipt of RR2 from B, A sends the disconnect command.

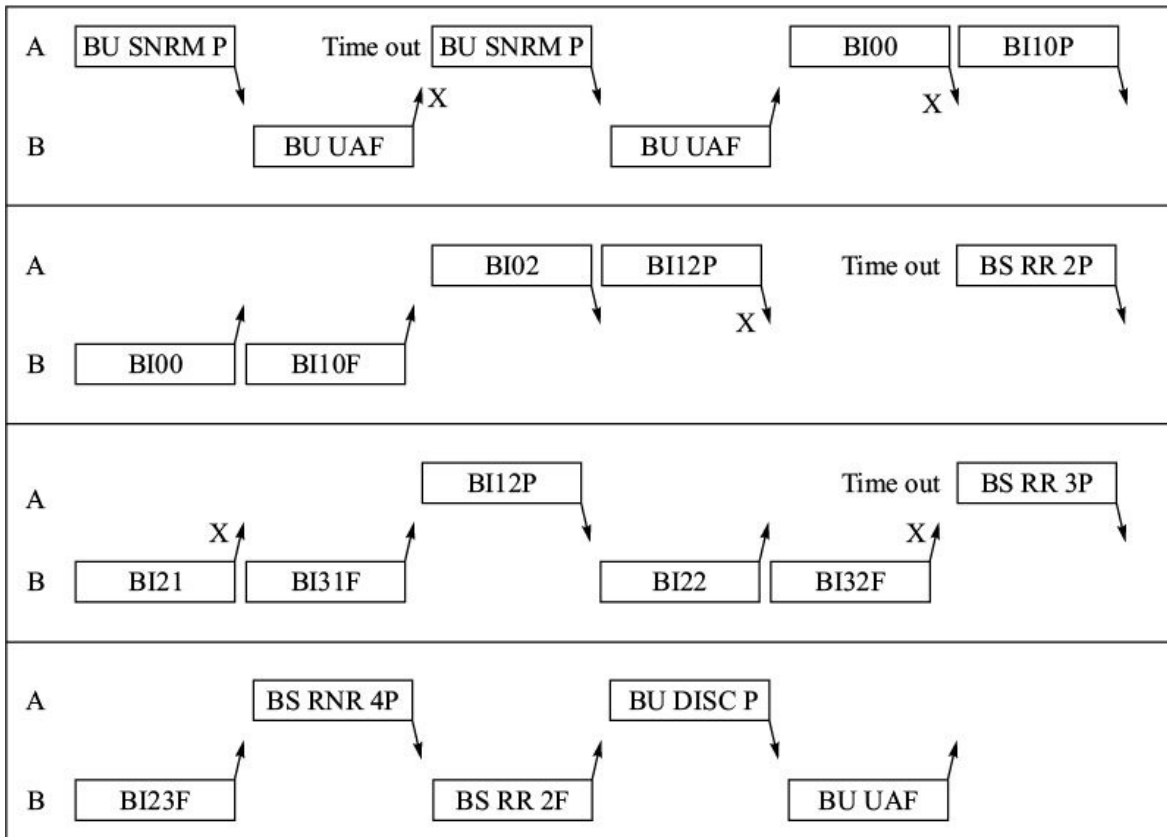


A : Primary, B : Secondary

Figure 9.18 TWA point-to-point communication in normal response mode without errors.

TWA point-to-point communications with errors. Figure 9.19 shows operation of the protocol over TWA link with errors.

- A mode setting command is retransmitted after timeout if it is not acknowledged.
- Loss of a frame is detected when the next frame in the sequence is received. A frame received with errors is also considered as lost because the error may even be in its sequence number.
- Loss of a frame is communicated when its acknowledgement is not received in the frames sent subsequently, *e.g.* BI00 could have been sent after receipt of BI11P of A. As BI00 of B does not acknowledge BI00 of A, BI00 of A must have been lost.
- Loss of a frame with P bit set to 1 is detected when there is no response. After timeout, the primary station A challenges the secondary station B by sending another frame (B S RR 2 P) with P bit set to 1.
- Loss of a frame with F bit set to 1 is detected by the primary station when there is no activity on the link after it sends a frame with P bit set to 1. It challenges the secondary station after timeout by sending another frame (B S RR 3 P) with P bit set to 1.

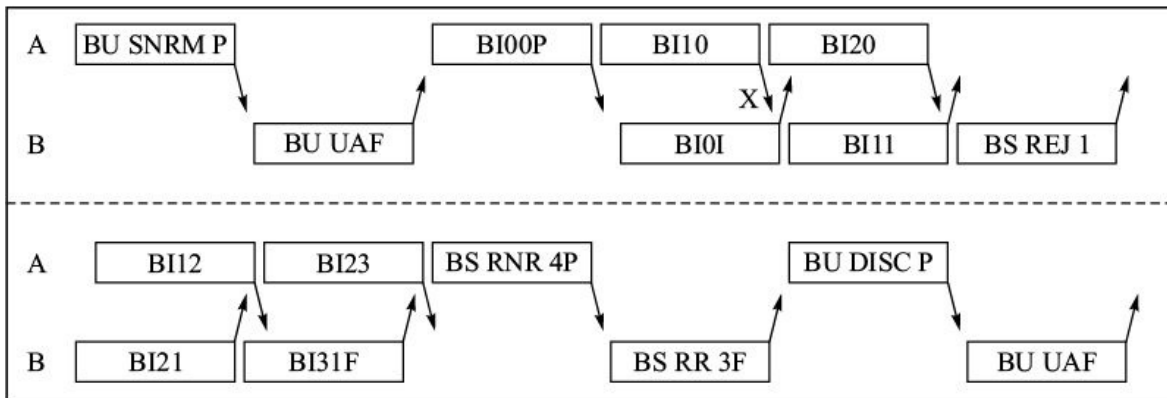


X : Lost frame

A : Primary, B : Secondary

Figure 9.19 TWA point-to-point communication in normal response mode with errors.

TWS point-to-point communication with errors. Figure 9.20 shows operation of the protocol over TWS link with errors.



X : Lost frame

A : Primary, B : Secondary

Figure 9.20 TWS point-to-point communication in normal response mode with errors.

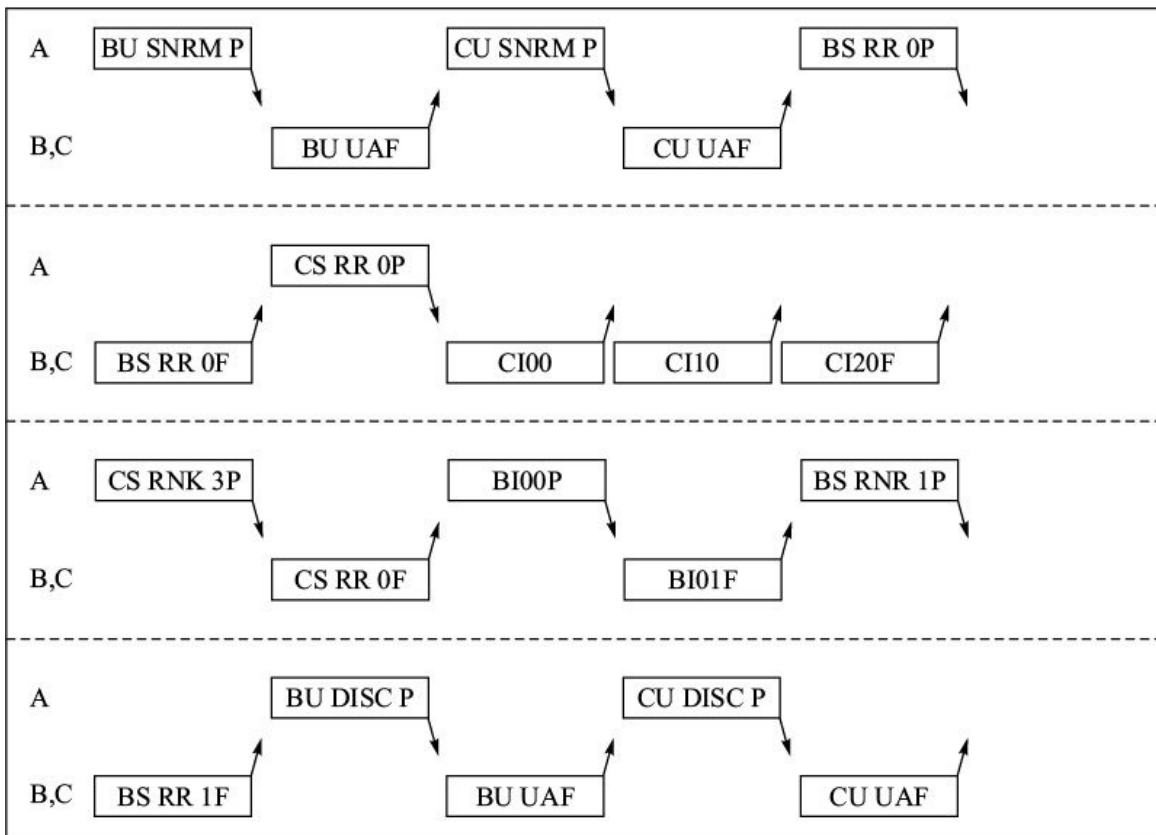
- Both the stations can transmit and receive simultaneously, but the secondary station can send a frame only after it receives permission to

transmit from the primary station.

- When loss of a frame is detected, a supervisory frame with REJ is sent.
- When REJ is received, all the frames from the lost frame onwards are retransmitted.

9.8.2 Normal Response Mode, Point-to-Multipoint In Figure 9.21, communication between a primary station A and two secondary stations B and C is shown. Two way alternate mode of communication is adopted.

- The secondary stations are set to the normal response mode individually at the start by the primary station. At the end of communication, the primary station sets the secondary stations in normal disconnected mode individually.
- The primary station polls the secondary station one at a time. If a secondary station does not have any frame to send, it responds with the receive ready (RR) frame with F bit set to 1.



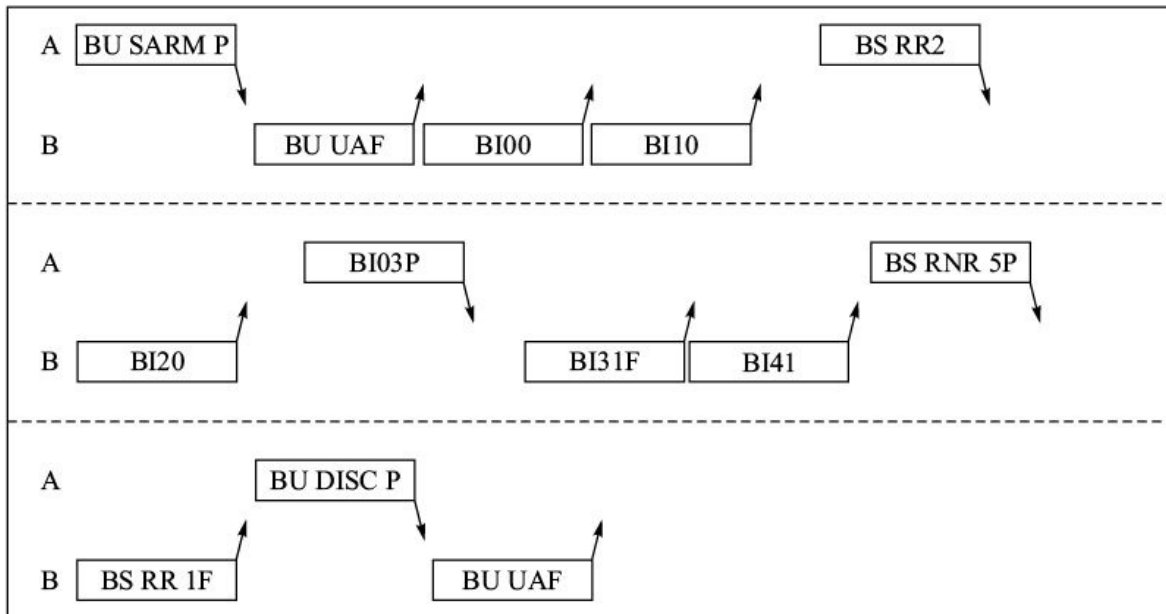
A : Primary, B, C : Secondary

Figure 9.21 TWA point-to-multipoint communication in normal response mode.

9.8.3 Asynchronous Response Mode (ARM) Figures 9.22 to 9.24 illustrate the operation of the protocol in asynchronous response mode over two way alternate and two way simultaneous links.

TWA communication without errors. Figure 9.22 shows the operation of the protocol in ARM over TWA link without errors.

- The secondary station need not wait for the poll from the primary station for sending an I-frame. In Figure 9.22, after receiving the mode setting command from primary station A, the secondary station B sends I-frames BI00 and BI10 to A.
- Since the stations are operating on TWA link, a station sends a frame after it senses no activity on the link.
- P bit is set to 1 only when the primary station wants to solicit an acknowledgement from the secondary station. In Figure 9.22, when A sends an I-frame (BI03P) with P bit set to 1, B responds with I-frame (BI31F) with F bit set to 1, indicating to A that this is the solicited response. This response is sent at the earliest opportunity.



A : Primary, B : Secondary

Figure 9.22 TWA point-to-point communication in asynchronous response mode without errors.

TWA communication with errors. Figure 9.23 shows the operation of the protocol in ARM over TWA link having errors.

- If a frame sent by the secondary station is lost, the secondary station retransmits the frame after timeout. In Figure 9.23, frame BI00 is lost, B waits for acknowledgement, and B retransmits the frame after timeout.
- The primary station can always force an acknowledgement from the secondary station after timeout by sending an S-frame (RR or RNR) with P bit set to 1.

TWS communication with errors. Figure 9.24 shows the operation of the protocol in ARM over TWS link having errors.

- When a frame is lost, an S-frame with REJ is sent. If the other station is in process of transmitting another frame when the REJ is received, the transmission is aborted and the next frame as requested in REJ is sent. In Figure 9.24, when BI11 from A is detected missing, B sends BS REJ 1. On receipt of BS REJ 1, A aborts transmission of BI33 and starts from BI13.

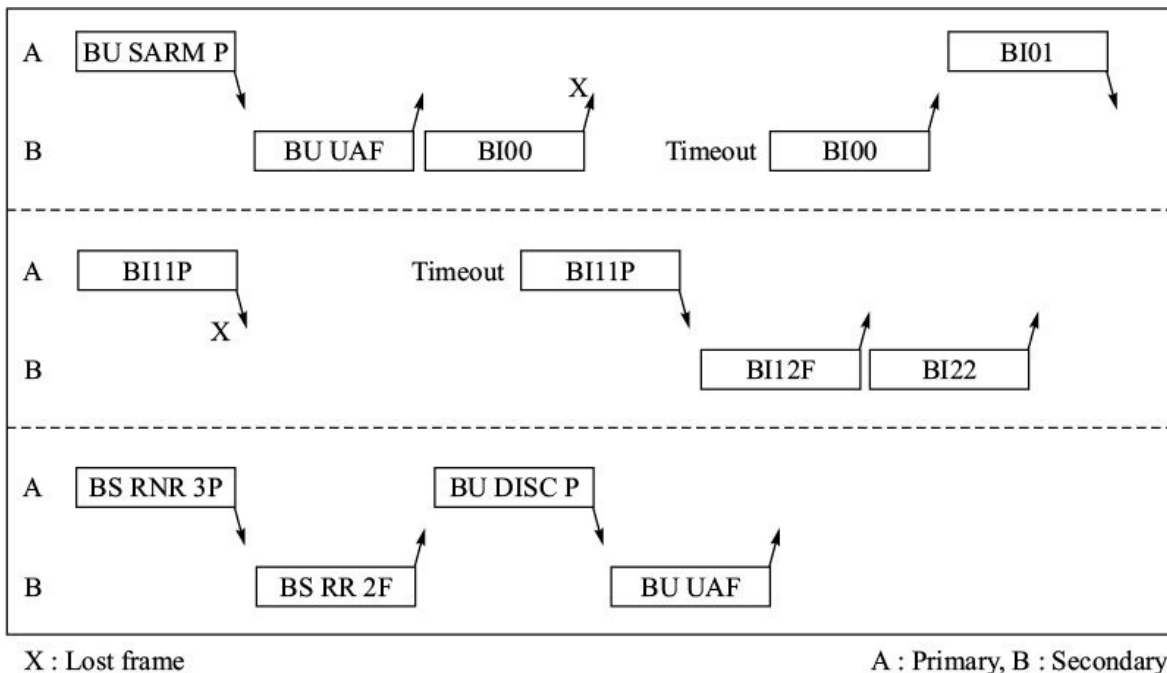


Figure 9.23 TWA point-to-point communication in asynchronous response mode with errors.

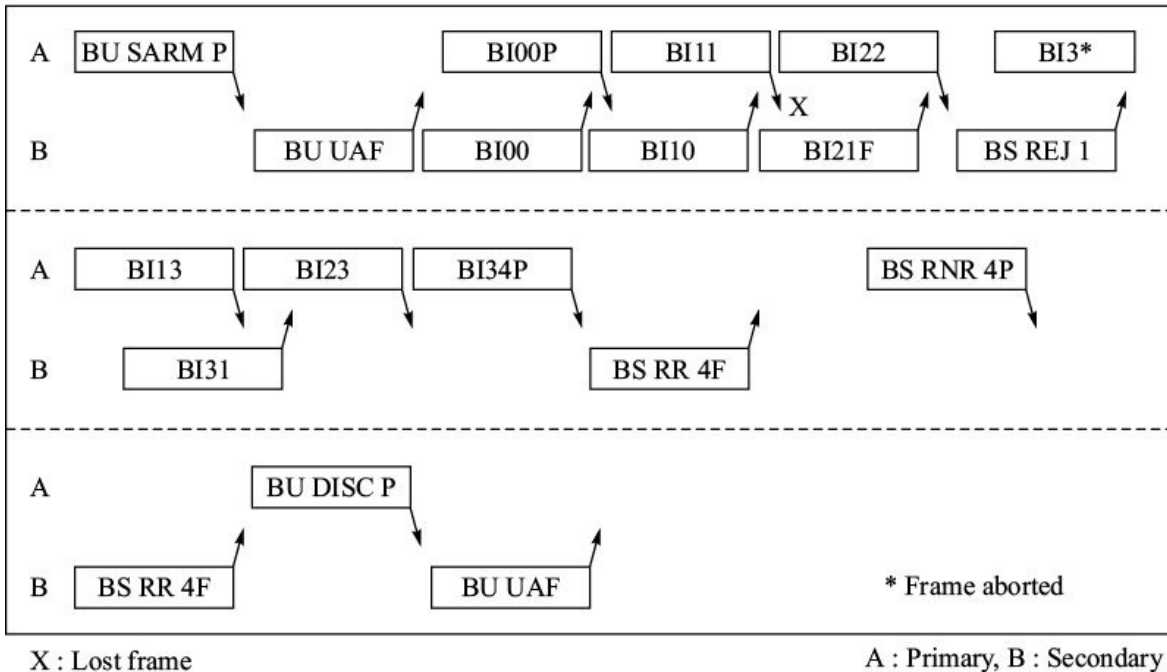
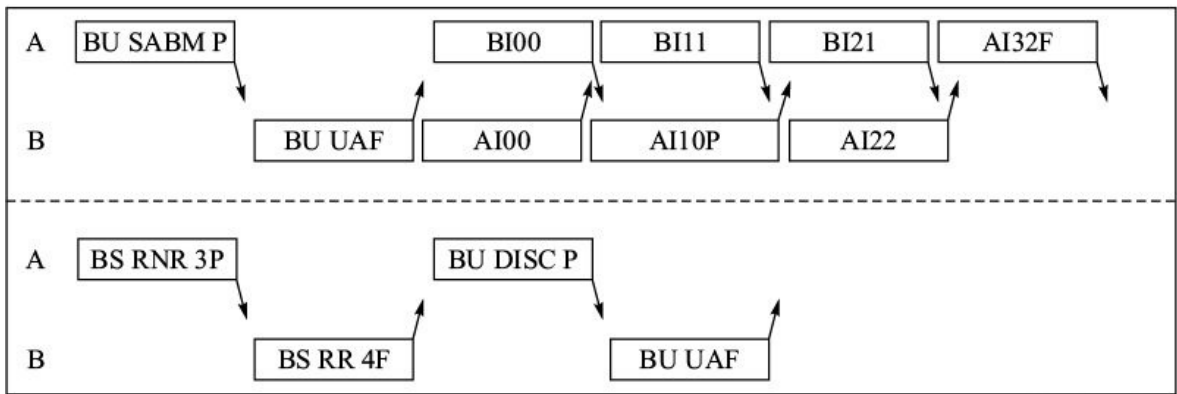


Figure 9.24 TWS point-to-point communication in asynchronous response mode with errors.

9.8.4 Asynchronous Balanced Mode (ABM) Figure 9.25 illustrates an example of asynchronous balanced mode of operation over two way simultaneous data link. A and B are combined stations.

- Either of the two combined stations can set up the link by sending SABM mode setting command.
- A station may send an I-frame as a command or as a response. The address field indicates whether it is a command or response.
- Both the stations can force a response by sending a command with P bit set to 1.



A, B : Combined stations

Figure 9.25 TWS point-to-point communication in asynchronous balanced mode without errors.

9.9 ADDITIONAL FEATURES

HDLC protocol has provision for extension of the address and the control fields. Extended address field is required when more than 256 addresses are to be accommodated. Extended control field is required when the window size is greater than seven. The format of HDLC frame is modified to accommodate the extended fields as described below.

9.9.1 Extended Addressing

Addressing capability of HDLC protocol can be enhanced using *multi-octet addressing scheme*. In case of multi-octet addressing, the address field is recursively extended using the first bit of each octet to indicate the extended format of the address field (Figure 9.26). The first bit of each address octet is set to 0 indicating that the next octet is also to be considered as part of the address

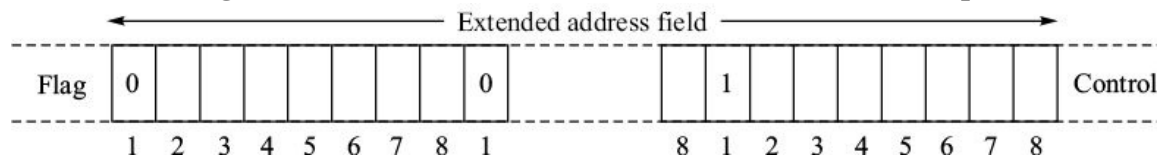


Figure 9.26 Extended addressing.

field. In the final address octet, the first bit is set to 1. The addressing scheme, either single octet or multi-octet is chosen once and thereafter it cannot be dynamically changed.

9.9.2 Extended Control Field

To extend the maximum window size in the HDLC protocol to 127, the sequence number needs to be seven bits long. To accommodate seven-bit frame sequence numbers, the control field of I- and S-frames is extended to two octets as shown in Figure 9.27. The control field of the U-frame remains unchanged as it does not carry the frame sequence number. SNRME, SARME and SABME mode-setting commands are used in place of corresponding SNRM, SARM, and SABM commands, for the extended format of the control field.

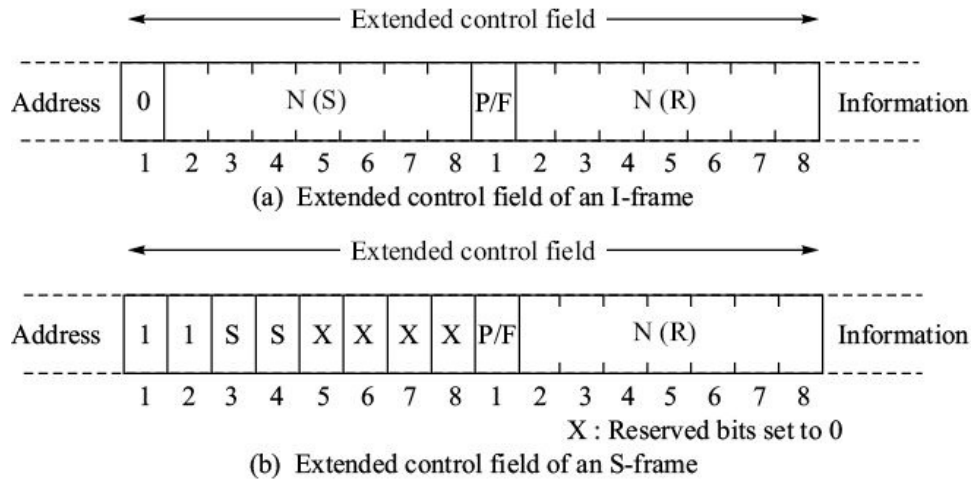


Figure 9.27 Extended control field.

9.10 COMPARISON OF BISYNC AND HDLC FEATURES

Table 9.2 gives comparison of the features of BISYNC and HDLC protocols. The physical layer characteristics required for supporting these protocols are also indicated.

TABLE 9.2 Comparative Features of BISYNC and HDLC Protocols		
Feature	BISYNC	HDLC
Transmission (synchronous/asynchronous)	Synchronous	Synchronous
Communication mode (synchronous/asynchronous)	Synchronous	Synchronous, asynchronous
Directional mode (TWA/TWS)	TWA	TWA, TWS
Configuration (Point-to-point/point-to-multipoint)	Point-to-point, point-to-multipoint	Point-to-point, point-to-multipoint

Flow control	Stop-and-wait	Sliding window
Error detection and correction	LRC, CRC, ACK0/ACK1	CRC, sequence number
Code set	ASCII, EBCDIC, Transcode	Any
Control characters	Many	None
Frame identifier	SYN SYN	Flag
Frame delimiter	ETB/ETX	Flag
Information field	Multiple bytes	Multiple bits
Transparency	DLE stuffing	Zero stuffing

9.11 LINK ACCESS PROCEDURE-BALANCED (LAP-B)

ITU-T Recommendation X.25 defines interface between a DTE and a DCE operating in the packet mode (Figure 9.28). The DCE is an access node of a packet switched data network and the DTE is a terminal equipment owned by the subscriber. A packet is N-PDU generated by the network layer of the DTE and is routed to the destination by the network layer of the packet network. We will study packet switching and X.25 recommendation in detail in Chapters 15 and 16.

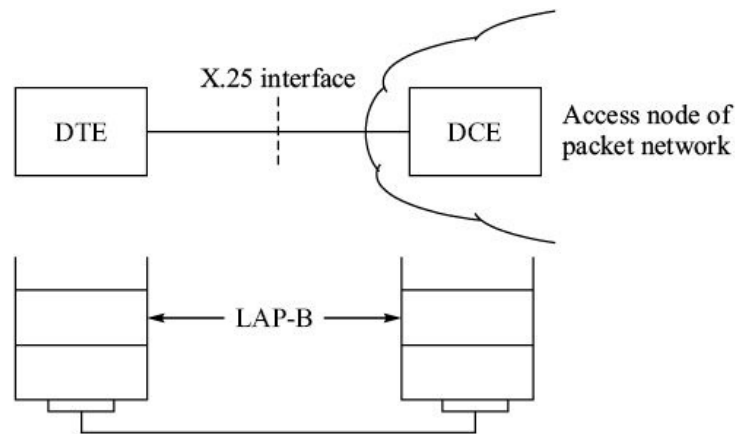


Figure 9.28 Link access procedure-balanced.

X.25 defines the interface for the first three layers. For the data link layer, X.25 specifies the Link Access Procedure-Balanced (LAP-B). LAP-B is an option of the HDLC protocol. Asynchronous balanced mode is used in the LAP-B protocol. The frame structures for I-frames, S-frames and U-frames and the basic protocol operation are the same as described earlier.

LAP-B is used for reliable transfer of data packets between the DTE to the DCE over the physical connection established by the physical layers. The data link layers of the DTE and the DCE are given addresses 00000011 and

00000001 respectively. These addresses enable identification of commands and responses as described earlier.

LAP-B specifies the following commands and responses for the U-frames:

- SABM Set to Asynchronous Balanced Mode command
- DISC Disconnect command
- DM Disconnected Mode response
- UA Unnumbered Acknowledgement response
- FRMR Frame Reject response.

FRMR is sent by the DTE or by the DCE to report a non-recoverable error condition such as receipt of an invalid command or response, receipt of an I-frame whose information field exceeds the maximum established length, and receipt of invalid acknowledgement number, N(R).

The S-frames use RR, RNR, and REJ acknowledgements. The I-frames contain the packets received from the network layer in their information field.

9.12 MULTILINK PROCEDURE (MLP)

ITU-T Recommendation X.25 permits multiple physical connections between the packet mode DTE and the subnetwork access node DCE (Figure 9.29). Multiple physical connections provide increased reliability of operation. The protocol for utilizing the multiple connections is implemented in the data link layer.

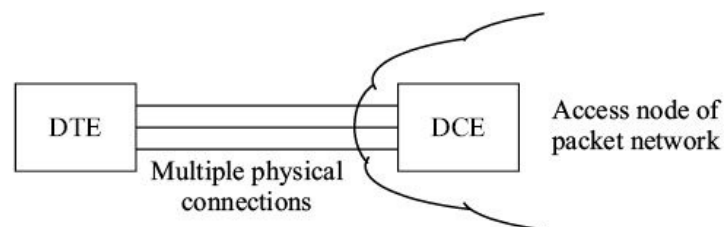


Figure 9.29 Multiple physical connections for increasing transmission reliability.

The data link layer is subdivided into a Multilink Procedure (MLP) sublayer and Single Link Procedure (SLP) sublayer (Figure 9.30). The SLP sublayer is LAP-B and, therefore, each single link connection operates as described in the last section.

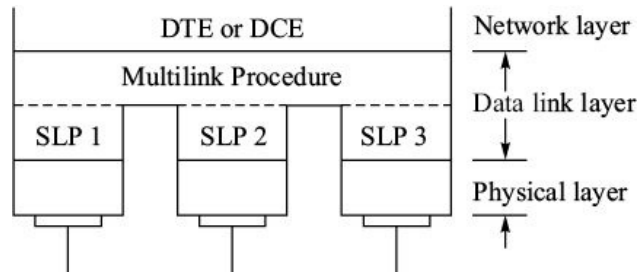
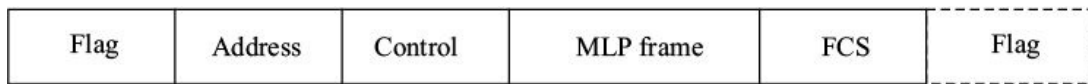
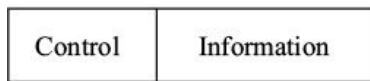


Figure 9.30 Sublayers of the data link layer for multilink procedure.

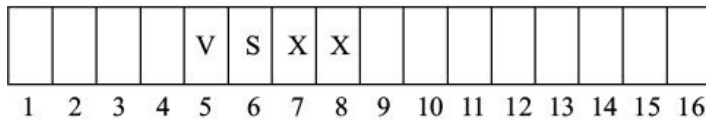
The multilink procedure forms a common sublayer above the SLP sublayers and makes all single links look like one logical link. It provides for optimum load sharing and resequencing of frames received from the different single links. To distribute the data units over different SLP sublayers and to resequence the data units at the other end, the MLP sublayer adds a multilink control field to the data units (Figure 9.31). The multilink frame so formed fits inside the information field of the LAP-B frame.



(a) SLP frame



(b) MLP frame



Bits 1 to 4 and 9 to 16 : MLP sequence number

V : Void sequence bit

S : Sequence check option bit

X : Reserved

Sequence number : 9 10 11 ... 16 1 2 3 4 (Bit positions)

LSB MSB

(c) MLP control field

FIGURE 9.31 Format of SLP frame.

The MLP control field is two octets long and contains 12-bit multilink sequence number, void sequencing bit (V bit), and sequencing check bit (S bit). The multilink sequence number is used for resequencing the received frames and for detecting the duplicate frames. It is not used for acknowledgement. All error control functions are implemented at the SLP level.

V and S bits are used as follows:

- S = 1 Sequence number is not assigned to the frame.
- S = 0 Sequence number is assigned to the frame.
- V = 1 Resequencing of the frames is not required.
- V = 0 Resequencing of the frames is required.

S bit is significant when V bit is 1. When S = 0 and V = 1, the sequence number is used only to detect duplicate frames.

9.13 LINK ACCESS PROCEDURES FOR MODEMS (LAP-M)

The high speed modems (V.32) have inbuilt error correction capability. These modems use HDLC-based error correction mechanism called link access procedure for modems (LAP-M). Figure 9.32 shows the layered architecture. Note that although HDLC is a layer 2 protocol, LAP-M is implemented as a sublayer of layer 1 between the modems. The data link layers of the end systems interact directly without any interaction with the LAP-M layer of the modems.

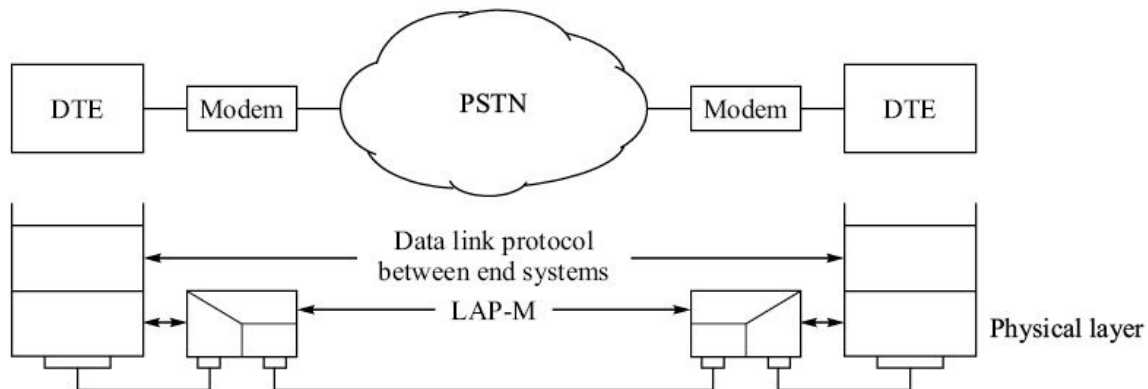


Figure 9.32 Layered architecture of LAP-M based error correcting modems.

LAP-M protocol is implemented on the line side of the modem. The digital interface of the modem towards the DTE is EIA-232-D. The data received at the digital interface is packed into the information field of the I-frame which is transferred across the link using Asynchronous Balanced Mode (ABM) of the HDLC protocol. Link set-up is done using SABM command and UA response. Link disconnection is done using DISC command UA response.

9.14 LINK ACCESS PROCEDURED (LAP-D)

ISDN interface, as we discussed in Chapter 3, consists of B-channels and a D-channel. B-channels carry the bearer traffic, *i.e.* digitized voice or user data packets. The user data packets are based on X.25 protocol (layer three) which we

will discuss later in another chapter. LAP-B data link protocol is used at layer two for carrying X.25 data packets. The X.25 data packets are encapsulated in the information field of the I-frames and sent over the B-channel.

The D-channel is used primarily for carrying signaling information for establishing the connections, *e.g.* for transporting dialed digits. Call control procedure has been defined by ITU-T in its I.451 Recommendations. D-channel can also be used for carrying user X.25 data packets (Figure 9.33). The data link protocol used in the D-channel is based on HDLC and is called Link Access ProcedureD (LAP-D).

9.14.1 Frame Format

The frame format used in LAP-D is shown in Figure 9.34. The basic frame structure is same as in HDLC. LAP-D uses window size up to 127, therefore, the control field of the I- and S-frames is two octets long to accommodate 7-bit sequence numbers. The maximum size of information field is 260 octets. The flag and FCS fields are same as in HDLC.

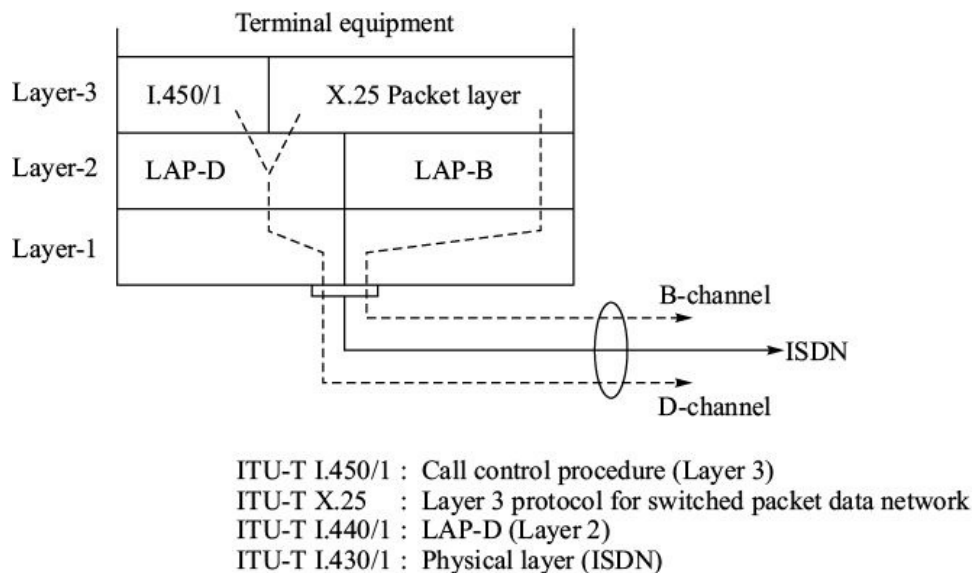


Figure 9.33 Layered architecture of ISDN terminal for data services.

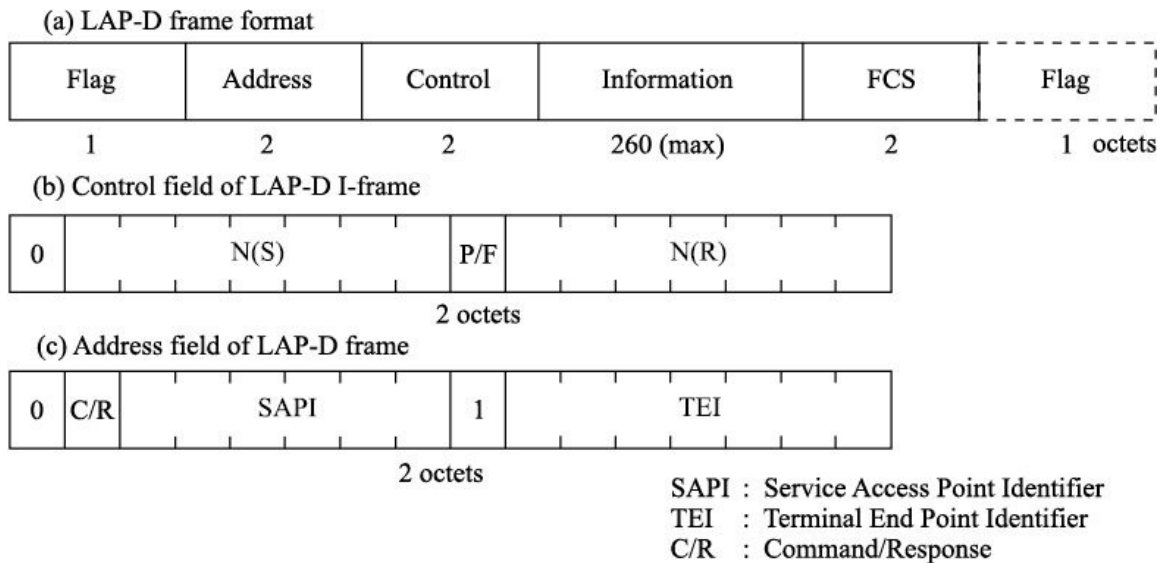


Figure 9.34 Frame format of LAP-D.

The major difference is in the address field which is two bytes long and contains two addresses—SAPI (Service Access Point Identifier) and TEI (Terminal End Point Identifier) as shown in Figure 9.34c. The two octets of address field have first bit 0 and 1 respectively, indicating the extended format of the address field.

As we mentioned in Chapter 3, we can connect up to eight devices on the S interface of ISDN connection. Therefore, a point-to-multipoint communication situation exists. The address of destination device needs to be specified in each frame coming from the exchange side. TEI address serves this purpose. Except the addressed device, other devices discard all the frames not addressed to them. All 1s is used as broadcast address when a frame is meant for all the terminal devices. The outgoing frame from a device contains the TEI of the device itself.

Note that LAP-D layer interfaces with two different entities at the next higher layer—call control and X.25 (Packet layer). Call control entity uses services of LAP-D layer for setting up the circuit switched connections through PSTN and X.25 (Packet layer) uses the services of LAP-D layer for establishing switched packet data connections. The SAPI field in the address identifies the user (Call control or X.25) of the data contained in the information field of an I-frame. The SAPI values are as follows:

- SAPI = 0 Call control procedure
- SAPI = 1 Packet mode communication using I.451 call control procedure
- SAPI = 16 Packet communication conforming to X.25 layer-3 procedure

C/R bit is used for indicating whether the frame is a command or response. In other words, whether the P/F bit is P bit or F bit.

LAP-D uses the following commands and responses of HDLC:

	<i>Commands</i>	<i>Responses</i>
• Supervisory frames:	RR, REJ, RNR	RR, REJ, RNR
• Unnumbered frames:	SABME, UI, DISC, XID	DM, UA, FRMR, XID

9.14.2 Procedures

We will consider a typical example of establishment of a packet switched data connection ISDN interface. Packet switched services can be provided by ISDN using X.25 packet handler in PSTN or by establishing connection to a X.25 packet switched data network (Figure 9.35). The terminal device has X.25 interface with full set of X.25 protocols (layers 1 to 3). Layer-2 protocol between the terminal and TA is LAP-B. It is connected through a TA (Terminal Adapter) to ISDN network. X.25 call establishment procedure (layer 3) is described in detail in Chapter 16.

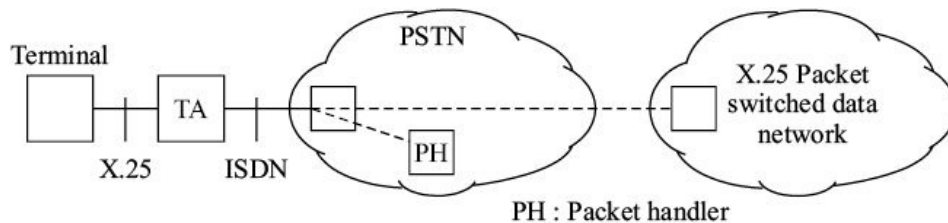


Figure 9.35 X.25 packet switched connection using ISDN.

We discuss here layer-2 procedure as applicable to establishing X.25 connection using B channel for bearer traffic and D channel for call control. Figure 9.36 illustrates the packet call set-up procedure.

- Packet call is originated by the X.25 terminal by sending an I-frame containing X.25 Call Request (CR) packet in its information field. The I-frame has been depicted as I[X.25,CR] in the figure.

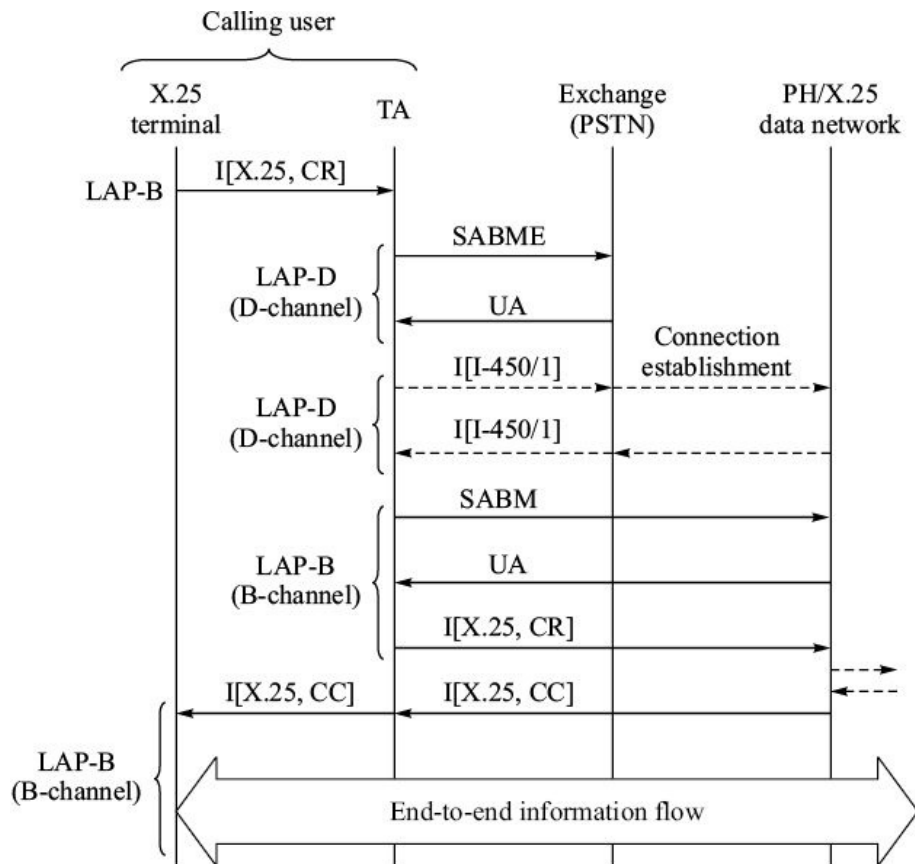


Figure 9.36 Packet call set-up using D-and B-channels.

- The TA initializes the LAP-D link to ISDN exchange by sending SBME command using a LAP-D frame on the D-channel.
- The exchange acknowledges the SBME command with UA response.
- I.451 call connection procedure to packet handler (PH) or to the packet switched data network node is initiated through D channel. LAP-D I-frames are used for this purpose.
- Once the connection is established, the LAP-B procedure on B-channel is invoked. TA sends SABM command to the PH/packet switched data network node to initialize the data link. The command is acknowledged by the PH/packet switched data network node by UA response.
- TA forwards the I[X.25, CR] frame received from the terminal to the PH/packet switched data network node B channel.
- The PH/packet switched data network node after establishing the X.25 connection to the destination terminal responds with Call Connected (CC) packet which is sent encapsulated in an I-frame to the call originating terminal. The I-frame is depicted as I[X.25, CC] in the figure.

- End-to-end information flow then starts.

SUMMARY

In this chapter we examined two data link protocols; Binary Synchronous Communication (BISYNC) and High-level Data Link Control (HDLC). BISYNC is a byte-oriented data link protocol. It supports two-way alternate synchronous communication between a master station and a slave station. It is applicable to point-to-point or point-to-multipoint configurations. It provides full data transparency and uses stop-and-wait mechanism of flow control.

BISYNC has two types of frame—data frames and supervisory frames. Supervisory frames are used for acknowledgements, polling, selecting, and sending enquiry to a station. They are not protected with error detecting bits. The data frames have one or two bytes long trailer for error detection. BISYNC uses several control characters from the code set as field delimiters (e.g. SOH, STX, ETX) and acknowledgements (e.g. ACK, NAK). Transparency is achieved by inserting DLE before the text field identifier STX.

High-level Data Link Control (HDLC) is a bit-oriented data link protocol. It provides two-way alternate or two-way simultaneous communication between two stations. The mode of communication can be synchronous or asynchronous. It is applicable to point-to-point or point-to-multipoint configurations. It provides full data transparency and uses sliding window mechanism of flow control. Error recovery is done using the 16-bit CRC code and timers. HDLC offers flexibility and adaptability for its application in a variety of network configurations. There are several variations of HDLC protocol that are used in various networking environments:

- Link Access Procedure-Balanced (LAP-B). It is used in X.25.
- Link Access Procedure Channel (LAP-D). It is used in ISDN.
- Link Access Procedure-Modem (LAP-M). It is used in V.32 modems.

EXERCISES

1. The heading and text fields of a BISYNC frame are given below: Heading : LQ
Text : D3
Write the data segment of the frame and bit transmission sequence. Assume ASCII code with even parity.

2. Compose BISYNC data frame for the following heading and text fields.

Heading : REPORT

Text : PAYMENT ETX MONTH DLE JUNE

ETX and DLE are the control characters and are part of the text field.

3. Find the BCC field of the following frame. All the bytes are coded using ASCII and even parity is used for VRC and LRC.

SYN SYN STX 90% ETB BCC

4. A is a host connected to three tributary stations B, C, and D. The polling sequence is B, C, D. The select sequence is C, B, D. A wants to send a message to each tributary station. The tributary stations also have a message to send to the host A. Priority is given to the messages from the host. Write the complete sequence of the frames exchanged between the host and the tributary stations. Use abbreviated frame notation as used in Example 9.4 of the chapter.

5. In Exercise 4, some of the frames as indicated below are lost during transmission or are received with errors as indicated: Data frame from A to D is received with error.

Data frame from B to A is lost in transit.

Acknowledgement for the data frame from C is lost in transit.

Write the complete sequence of frame exchange.

6. Shown below is an example of BISYNC point-to-point communication without any errors. Rewrite the exchange of frames (a) if B had sent a WACK for the first data frame.

(b) if the first data frame was received with errors and the acknowledgement of the second frame was lost.

A	ENQ		STX-ETB		STX-ETB		EOT
B		ACK0		ACK1		ACK0	

7. Translate the following dialogue between stations A and B into BISYNC protocol: A : Are you ready to receive?

B : Yes.

A : Here is data frame.

B : Received correctly.

A : Here is the next data frame.

(The frame is lost. There is no response from B.) A : Did you receive my

last data frame?

B : The one I last received was correct and here is the acknowledgement I sent.

A : Here is the data frame again.

B : Received correctly.

A : I am terminating transmission.

B : Are you ready to receive?

A : Yes.

B : Here is data frame.

A : Received correctly but wait.

A : O.K. Go ahead.

B : Here is data frame.

A : There are errors in your data frame.

B : Here is the data frame again.

A : Received correctly.

B : I am terminating the transmission.

8. Locate an HDLC frame from the following bit streams and identify its various fields.

(a)

0111111001111110110011000111001000001110010101010100111110011

(b)

011111101101111100111100110101011111011101010111110111110.

9. Various fields of an HDLC frame are given below. The bits are shown in their transmission order. Compose the HDLC frame.

Address : 00011111

Control : 00110111

Information : 11111000011

FCS : 1000000101110001.

10. Write the control fields of the following frames. Low-order bits have been shown first: (a) I-frame, $N(S) = 010$, $N(R) = 101$, command with P bit equal to 0.

(b) S-frame, RNR, $N(R) = 101$, command with P bit equal to 1.

(c) U-frame, Unnumbered acknowledgement, response with F bit equal to 1.

11. Fill in the blanks. Frame representation as explained in the chapter has

been used. A is the primary station and B the secondary station.

(a) Normal Response Mode, no errors, TWA communication.

A	_ U SNRM _		_ I _ _ _		_ I _ _ P
B		_ _ UA _		_ I _ _ _	

A		_ _ DISC _	
B	_ S RR _ _		_ U _ _

(b) Normal Response Mode, with errors, TWA communication. (*) indicates that the frame is lost.

A	_ _ SNRM _		_ I _ _ P*	Time out	B S _ _ P
B		_ _ _ _ _			

A		_ I _ _ P			_ I _ _ P
B	_ S _ _ _		_ I _ _ _*	_ I _ _ _	

A			_ S RNR _ _		_ U DISC _
B	_ I _ _ _	_ I _ _ _		_ S _ _ _	

A	
B	_ _ UA _

(c) Normal Response Mode, with errors, TWS communication. (*) indicates that the frame is lost.

A	_ U SNRM _		_ I _ _ P	_ I _ _ _	_ I _ _ _
B		_ _ UA _		_ I _ _ _	_ I _ _ F

A	_ S RR _ P			Timeout	_ S RR _ P
B		_ I _ _	_ I _ _ F*		

A		_ S RNR _ P		_ U DISC _	
B	_ I _ _ F		_ S _ _ _		_ _ UA _

12. A is the primary station and B the secondary station. A and B

communicate in TWA mode. Give the sequence of HDLC frames corresponding to the following actions: (a) A sends a command to set B in normal response mode.

(b) B acknowledges.

(c) A polls B to transmit.

(d) B indicates that it has nothing to send.

(e) A sends an information frame and polls B again.

(f) B acknowledges.

(g) A sends disconnect command.

(h) B acknowledges.

13. A is the primary station and B the secondary station. A and B communicate in TWA mode. B is already in normal response mode. Give the sequence of frames corresponding to the following actions: (a) A sends information frames 0 and 1 and polls B.

(b) B acknowledges the frames and sends its frames 0 and 1, indicates nothing more to send.

(c) A sends frames 2 and 3 and polls B. A indicates that it wants frame 0 from B.

(d) B sends frame 0 and 1, acknowledges the frame 1 from A and indicates nothing more to send.

(e) A sends frames 2 and 3, acknowledges the frames 0 and 1 from B and polls B.

(f) B acknowledges the frames from A and indicates that nothing more to send.

(g) A sends disconnect command.

(h) B acknowledges.

Which were the frames lost during transit in steps (b) and (c)?

14. A is the primary station and B the secondary station. A and B communicate in TWA mode. B is already in asynchronous response mode of HDLC. Give the sequence of frames corresponding to the following actions: (a) B sends information frames 0 and 1.

(b) A acknowledges.

(c) B sends information frames 2 and 3.

(d) A sends information frames 0 and 1, acknowledges the frame 2 from B, and polls B.

(e) B responds and sends information frame 3, acknowledges frames 0 and

1 from A.

(f) A acknowledges, indicates that it is not ready for more information frames, and polls B.

(g) B responds with acknowledgement.

(h) A sends disconnect command.

(i) B responds with acknowledgement.

1 Normal mode is synchronous mode of communication. Thus, HDLC permits both synchronous and asynchronous modes of communication. Synchronous and asynchronous terms do not imply 'start-stop/bits' or 'clock' which refer to the physical layer. These terms refer to the dialogue discipline for communications. At physical layer, HDLC requires synchronous transmission.

10

Local Area Networks

In Chapter 9, we considered application of data link protocols for simple network configurations, point-to-point and point-to-multipoint. These configurations are not suitable for any-to-any communications. For example, if we have N computer terminals, we will need $N - 1$ ports on each computer and $N(N - 1)/2$ dedicated links for interconnecting the terminals (Figure 10.1a). Obviously, it is not a scalable solution. We need to create network that provides any terminal to any terminal connectivity with single port to each computer terminal and optimize on the number of interconnecting links. Figures 10.1b and c show two of the several possible network topologies. Note that we require only one port on each terminal, and the interconnecting media resources are shared. These simple networks are categorized as Local Area Networks (LANs). Their geographical coverage is restricted to a building. They are based on the first two layers of the OSI reference model. In this chapter we present an introduction to such networks. This will be the first step towards building complex data networks that span across the globe and that require network layer also.

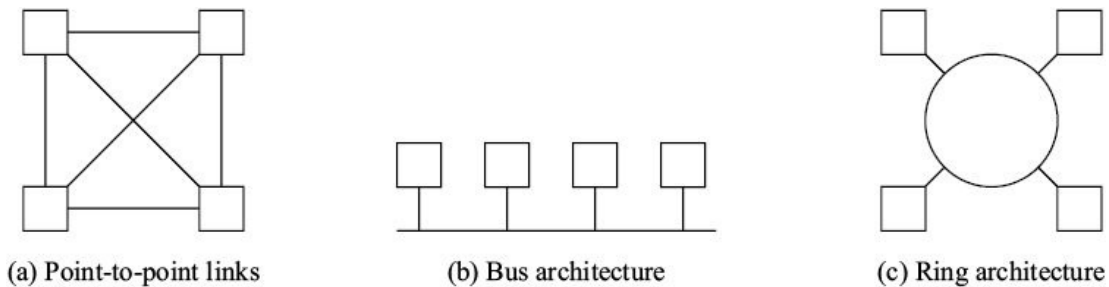


Figure 10.1 Data networks.

We begin this chapter with the examination of local networking requirements of an organization and the basic LAN attributes. Bus, ring, and star LAN topologies are described next. Then we proceed to examine the layered architecture of the local area networks. Media access control and addressing issues are discussed with the help of layered architecture. After a brief look at

IEEE standards for the local area networks, we discuss the services and protocols of the LAN Data Link sublayers. We move next to the transmission media for local area networks and discuss their characteristics and capabilities.

10.1 NEED FOR LOCAL AREA NETWORKS

Local area network, LAN in short, is a generic term for a network facility spread over a relatively small geographical radius. The LAN concept began with the development of distributed processing in the seventies. With the proliferation of microcomputers to the work sites, the need was felt to interconnect the computers so that data, software, and hardware resources within the premises of an organization could be shared.

10.1.1 LAN Attributes

A LAN consists of a number of computers, graphic stations and user terminal stations interconnected through a cabling system. It has the following characteristic attributes:

- Geographic coverage of local area networks is limited to area less than 5 km.
- The data rates exceed 1 Mbps.
- The physical interconnecting medium is privately owned.
- The physical interconnecting medium is usually shared by the stations connected to the LAN.

10.1.2 LAN Environment in an Organization

Figure 10.2 shows the computer networking configuration within an organization. Usually, the various departments of the organization establish their own LANs which interconnect the departmental microcomputers. Each of these LANs may be self-sufficient in its resources and management. However, there is always a need to exchange interdepartmental information and access one or more servers at the organizational level. A high-speed backbone LAN serves the purpose of connecting various departmental LANs, mainframe and mini-systems. The mainframe systems may have their own dedicated back-end LAN

for sharing data and high-speed peripherals.

Multiple LAN environment in an organization has several advantages, such as:

- Better data security can be achieved by restricting the interdepartmental LAN access and access to the servers.
- The traffic is dispersed over several LANs which individually meet the intradepartmental communication requirements.
- Partitioning a big network into several smaller LANs also overcomes the distance and data rate limitations of local area networks.

At the same time, however, internetworking of various LANs becomes a major issue. Bridges, routers, and gateways are required to interconnect the LANs. We will address this issue in Chapter 14.

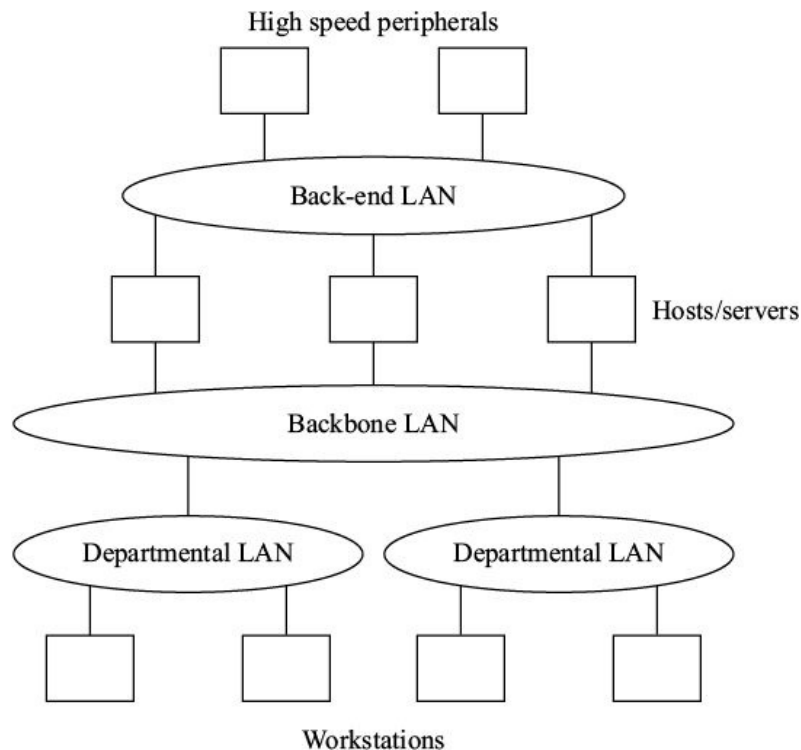


Figure 10.2 Computer networking environment within an organization.

10.2 LAN TOPOLOGIES

The physical topology of a local area network refers to the way in which the stations are physically interconnected. Physical topology of a local area network

should have the following desirable features:

- The topology should be flexible to accommodate changes in physical locations of the stations, increase in the number of stations, and increase in the LAN geographic coverage.
- The cost of physical media and installation should be minimum.
- The network should not have any single point of complete failure.

There are several LAN topologies available in the industry. Bus topology, ring topology, and star topology are common. There can be some other topologies as well such as distributed star, tree, *etc.* These are extensions of the basic topologies, *i.e.* bus, ring, and star.

10.2.1 Bus Topology

In bus topology, a single transmission medium interconnects all the stations which share this medium for transmission of their signals (Figure 10.3). The bus operates in broadcast mode, *i.e.* every station listens to all the transmissions on the bus. Every transmission has source and destination addresses so that stations can pick the data units meant for them and identify their senders. If two stations transmit simultaneously, the collision takes place.

The bus shown in Figure 10.3 is a two way bus, *i.e.* the signals flow in both directions. To avoid signal reflection, the ends of the bus are terminated into matching impedance called the *head end*.

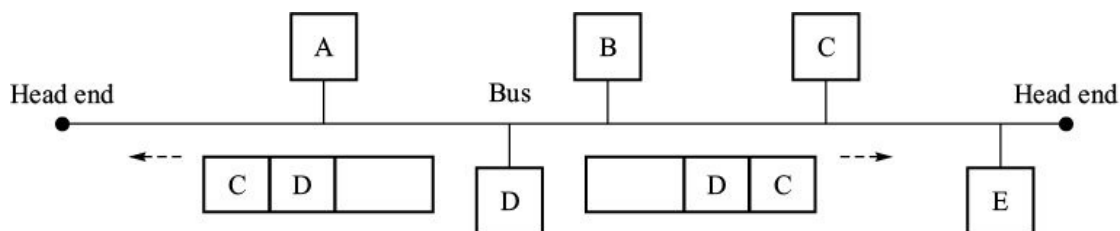


Figure 10.3 Bus topology.

Repeater. If the geographic coverage needs to be expanded, repeaters which interconnect two buses are required (Figure 10.4). A *repeater* regenerates the signals of one bus into the other bus. It is transparent to the rest of the system in the sense that it does not have a buffer and interconnects the two sections of the LAN to make them virtually one section.

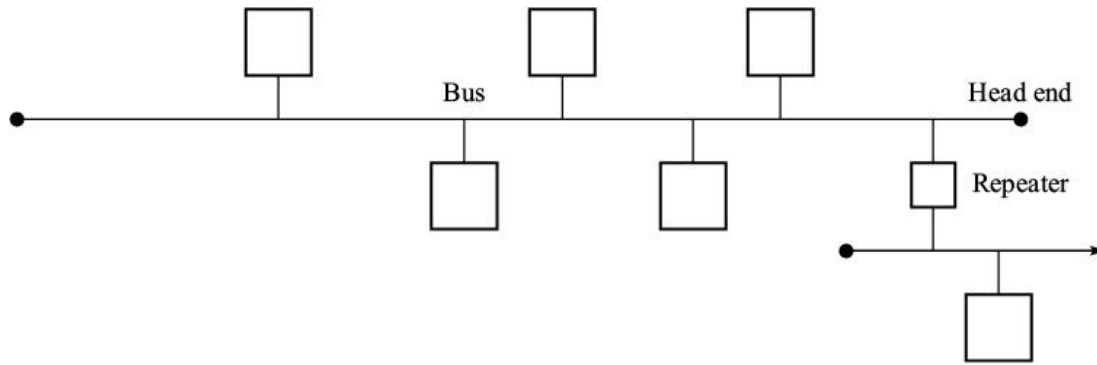


Figure 10.4 Bus with a repeater.

Dual bus. If signals are amplified along the bus, the bus becomes unidirectional. In this case, two separate buses are required—a transmit bus and a receive bus (Figure 10.5). Every station injects signals on the transmit bus and listens on the receive bus. The buses are looped at one of the two ends.

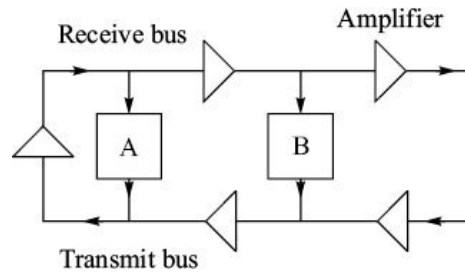


Figure 10.5 Dual bus topology with amplifiers.

Dual bus using FDM. The transmit and receive buses can also be provided on a single bus by frequency division multiplexing of transmission channels (Figure 10.6). The stations send

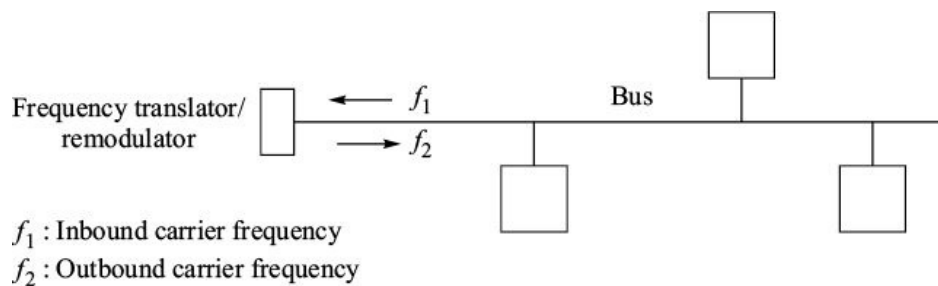


FIGURE 10.6 Dual bus topology using frequency division multiplexing.

digital signals by modulating the transmit carrier and receive their digital signals by demodulating the receive carrier. The head end consists of a frequency translator which changes the inbound carrier frequency to the outbound carrier frequency. A remodulator is also used in place of a simple frequency translator. The remodulator first demodulates the inbound carrier to get the digital signals

and then modulates the outbound carrier.

Some advantages of bus topology are:

- Stations can be connected to the bus using a passive tap.
- Least length of physical transmission medium is used.
- Coverage can be increased by extending the bus through the use of repeaters.
- New stations are easily added by tapping a working bus.

10.2.2 Ring Topology

A ring network consists of a number of transmission links joined together in the form of a

ring through repeaters called Ring Interface Units, RIU (Figure 10.7). The transmission is unidirectional on the ring. Thus, each RIU receives the signals at its input and after regeneration, sends them to the RIU of the next station. Every data unit in the ring contains the source and

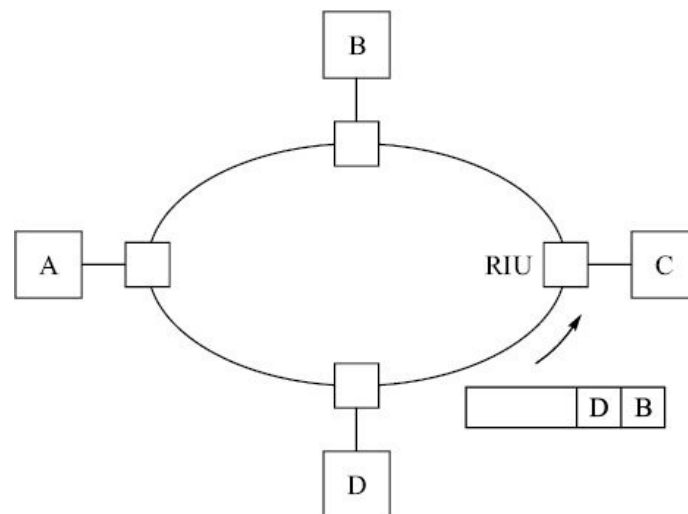


Figure 10.7 Ring topology.

destination addresses. When a circulating data unit passes through an RIU, the station connected to the RIU retains a copy of the data unit if the data unit is meant for it.

Since the ring does not have an end, the data units will circulate continuously unless they are removed from the ring. The responsibility of removing a data unit after it has completed one round is given to the sending station. The destination station is not given this responsibility because it may be out of order

or the destination address may be wrong. The possibility of the sending station going out of order after transmitting a data unit cannot be ruled out and, therefore, a monitoring station is also required. It removes the data units going round a second time.

A ring is not as flexible as a bus because to add a station involves breaking the ring and adding an RIU. Wire centres are provided to improve the flexibility of removing or adding a station and to isolate a faulty section (Figure 10.8). All the stations are connected to the wire centre.

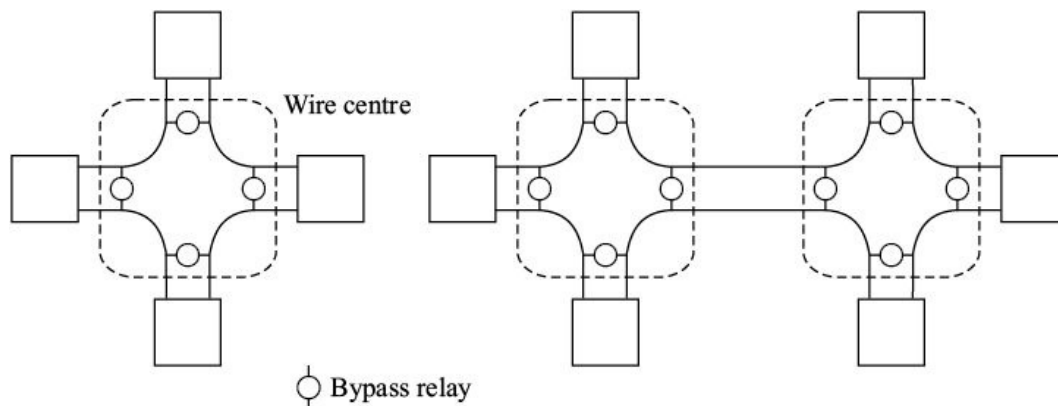


FIGURE 10.8 Wire centres.

If an RIU fails, this can result in total network failure. Therefore, a relay is provided in the wire centre to bypass the failed RIU. Two wire centres can be connected together to increase geographic coverage of the network. As can be seen from the above figure, a ring network does not economize on cables.

10.2.3 Star Topology

A star network consists of dedicated point-to-point links from the stations to the central controller (Figure 10.9). Each interconnection supports two-way communication. There can be two alternative communication approaches:

- The central controller acts as a switch to route the data units from the source to the destination.
- The central node can operate in broadcast mode. A frame from one station is transmitted to all other stations by the central node. In this case the central node is referred to as hub.

The use of star topology was initially restricted to small LAN installations. The high speed LAN technology that was developed in the late nineties operated

on point-to-point links. Today most of the LAN installations are based on star/tree topology using hubs and LAN switches.

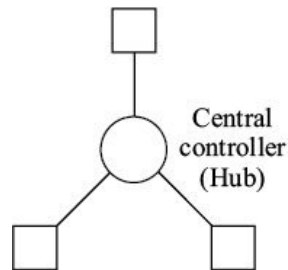


Figure 10.9 Star topology.

10.2.4 Logical Topology

In the past, LANs were categorized on the basis of physical topology because it also determined the way in which the LANs operated. But today we have LANs having the same physical topology but operating in different ways. Therefore, LAN categorization based on physical topology is incomplete.

Logical topology refers to the way the stations are interconnected for the purpose of exchanging data units. Physical topology, discussed above, may be different from the logical topology. For example, the LAN in Figure 10.9 has star topology. If the central node acts as hub and broadcasts a data unit received from a station to all other stations, the logical topology of the network is a bus. A station cannot transmit when transmission from another station is ongoing because two transmissions will collide, just like it happens in a bus.

10.3 MEDIA ACCESS CONTROL

The network of a LAN consists of a physical transmission medium which interconnects various stations of the network. The common transmission medium is shared by the stations connected to it. A discipline is followed by the stations so that every station gets fair opportunity to transmit its data and collisions (two stations simultaneously accessing the media) do not take place. Procedures for accessing the medium for signal transmission are called *media access control methods*.

Media access control and addressing functions are implemented in the data link layer of the stations. These functions are in addition to the error and flow control functions of the data link layer. A discipline is built up among the stations of the LAN so that fair opportunity is given to each station to transmit

its data frames.

For the bus topology, the following two methods have been standardized:

- Carrier Sense Multiple Access/Collision Detection (CSMA/CD)
- token passing.
- For the ring topology also, there are several media access control methods as follows:
 - Token passing
 - Register insertion
 - Empty slot.

Of these media access control methods, CSMA/CD dominates the present day market. Register insertion and empty slot mechanisms are of academic interest only.

For the media access control, the data link layer is divided into two sublayers. The media access control mechanisms are implemented in the media access control sublayer of the data link layer which is described in the next section.

10.4 LAYERED ARCHITECTURE OF LAN

The data link layer in the local area networks is divided into two sublayers. These sublayers are called Logical Link Control (LLC) and Media Access Control (MAC). Figure 10.10 shows the relationship of this division to the OSI reference model.

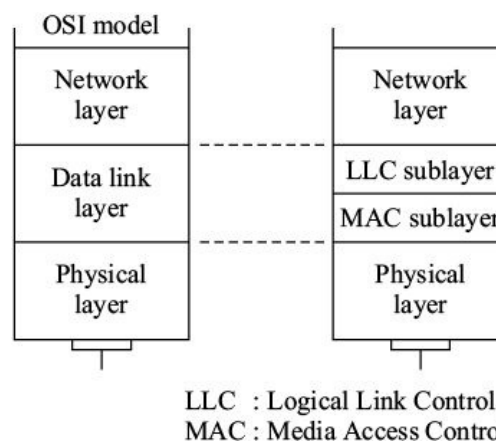


Figure 10.10 Layered architecture of a LAN.

The physical layer transports bits from one station to all the other stations

using suitable signal codes. The data link layer functions are divided between the two sublayers as follows:

- | | |
|--|---|
| <p><i>Media access control (MAC) sublayer</i></p> <ul style="list-style-type: none"> • Control of access to media • Unique addressing of stations directly connected to the LAN • Error detection | <p><i>Logical link control (LLC) sublayer</i></p> <ul style="list-style-type: none"> • Error recovery • Flow control • User addressing |
|--|---|

To carry out these functions, each sublayer appends a header to the user data. The MAC sublayer also adds a trailer containing error check bits (Figure 10.11).

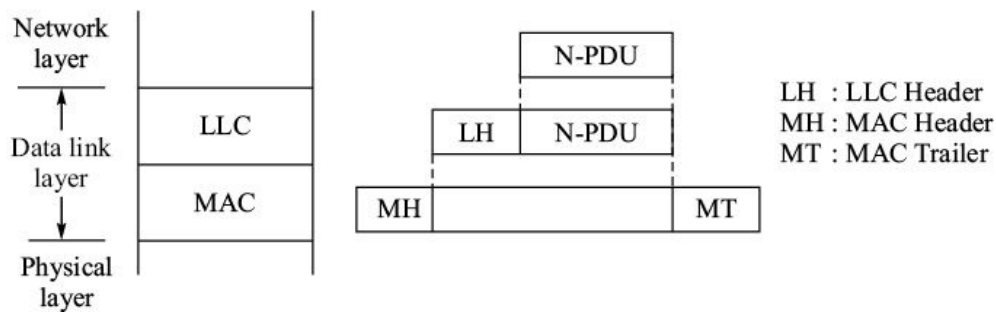


Figure 10.11 LLC and MAC headers of a LAN frame.

The user data (N-PDU) from the network layer is passed to the LLC sublayer which adds the protocol control information in the form of LLC header to it. The LLC protocol data unit so formed is passed to the MAC sublayer which adds MAC header and trailer to the LLC-PDU to form MAC-PDU or the frame. The frame is handed over to the physical layer for transmission.

The LLC header contains a control field and address fields. Its structure is based on HDLC frame. The control field is used for error control, flow control and sequencing of LLC service data units. The address fields identify the sending and receiving network layer entities. For example, destination address (SAP) = 6 identifies IP network layer entity in Figure 10.12. Thus the LLC sublayer hands over N-PDU contained in an LLC frame having LLC destination address 6 to the IP network layer.

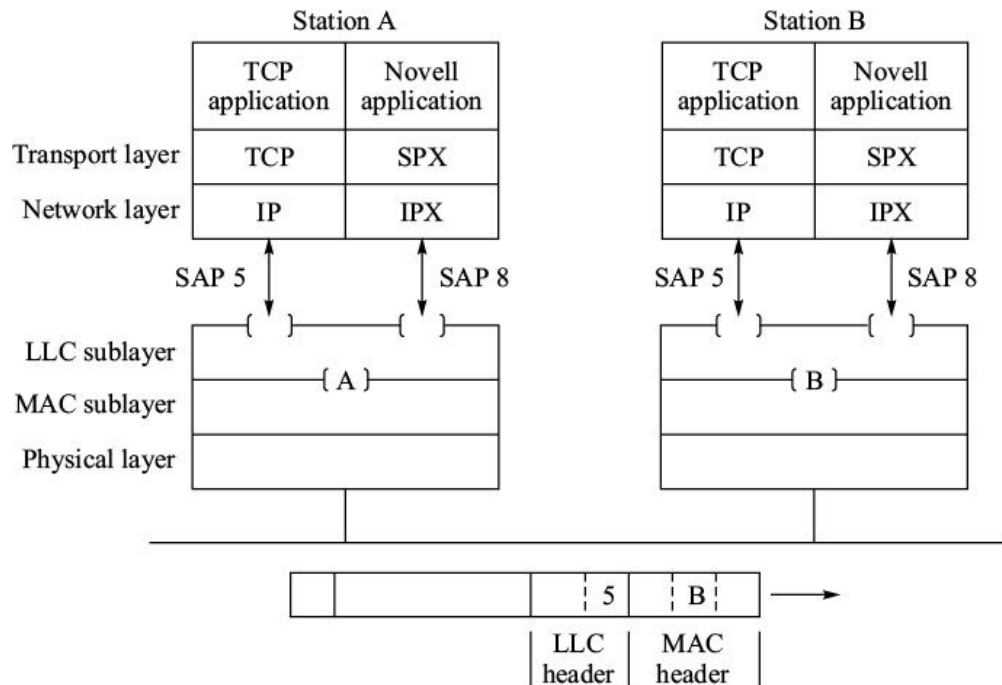


Figure 10.12 Addressing at MAC and LLC sublayers.

The MAC header consists of media access control field and station address fields. The MAC address fields identify the sending and receiving stations. Note the difference in addressing at LLC level and at MAC level (Figure 10.12). The destination address on a MAC frame identifies the station, the frame is meant for. Thus the frame shown in the above figure will be accepted by station B. Other stations will discard this frame.

For error detection, the MAC sublayer appends error check bits as a trailer and carries out content error check at the receiving end. If an error is detected, it does not request for retransmission. It just discards the frame and leaves the LLC sublayer to recover from the error.

10.5 IEEE STANDARDS

The layered architecture and other standards of LANs have been developed by the Institution of Electrical and Electronics Engineers (IEEE) under their project 802 set up in 1980. The following groups were constituted for the purpose:

- Architecture, Management and Internetworking
- Logical Link Control (LLC).
- CSMA/CD

- Token Bus
- Token Ring
- Metropolitan Area Networks (MANs)
- Broadband Technical Advisory Group
- Fibre Optic Technical Advisory Group
- Integrated Data and Voice Network.

For the local area networks, IEEE adopted three mechanisms of media access control, namely, CSMA/CD, token bus, and token ring. The IEEE standards for these media access control schemes and associated physical layers are IEEE 802.3, IEEE 802.4, and IEEE 802.5 respectively (Figure 10.13). The physical layer specifications include signal encoding, data rates, and interface to the physical transmission medium. The logical link control specifications are given in IEEE 802.2. These standards have been adopted by other organizations as well. The corresponding ISO references are 8802/2, 8802/3, 8802/4, and 8802/5.

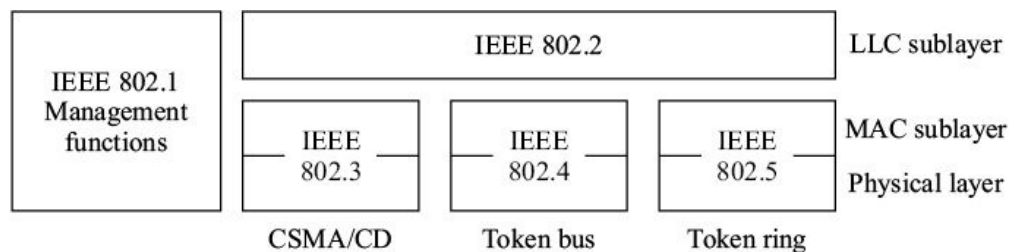


Figure 10.13 IEEE LAN related standards.

10.6 LOGICAL LINK CONTROL (LLC) SUBLAYER

As mentioned earlier, the LLC sublayer carries out error control, flow control, sequencing, and user addressing functions. It provides service to the network layer entities and receives services from the MAC sublayer to carry out the assigned functions.

10.6.1 LLC Service

The LLC sublayer offers the following three types of services to the network layer entity. These services are made available at the data link layer Service Access Point (SAP).

- Type 1—Unacknowledged connectionless-mode service
- Type 2—Connection-mode service
- Type 3—Acknowledged connectionless-mode service.

A station can be capable of providing more than one type of service. Therefore, four classes of stations are defined:

- Class 1 (LLC1)—Type 1
- Class 2 (LLC2)—Types 1 and 2
- Class 3 (LLC3)—Types 1 and 3
- Class 4 (LLC4)—Types 1, 2, and 3.

Type 1—Unacknowledged connectionless-mode service. It is a non-reliable data link service in which there is no guarantee of data delivery. There is no acknowledgement either of delivery or non-delivery of data units. There is no error control, flow control or sequencing of the data units in the data link layer. These functions become responsibility of the higher layers. For example, if TCP layer of TCP/IP suite provides error control, flow control, and sequencing functions, there may not be any point in duplicating these functions at the data link layer.

Only two primitives are specified for this service:

- DL-UNITDATA request (source address, destination address, user data, priority)
- DL-UNITDATA indication (source address, destination address, user data, priority).

The request primitive is used at the transmitting end to pass the user data to the LLC sublayer and the indication primitive is used at the receiving end to pass the received user data to the user. The priority parameter is passed down to MAC sublayer which implements the priority mechanisms.

Type 2—Connection-mode service. In this service, three phases are involved, connection establishment phase, data transfer phase and disconnection phase. Flow control, error control, and sequencing of the data units are the basic functions of this service.

The primitives used during the three phases of connection-mode service are given in Table 10.1.

TABLE 10.1 Service Primitives for Connection-mode Service of LLC Sublayer

Service	Primitives	Parameters
DL-CONNECT	Request	source address, destination address, priority

DL-CONNECT	Indication	source address, destination address, priority
DL-CONNECT	Confirm	source address, destination address, priority
DL-DATA	Request	source address, destination address, data
DL-DATA	Indication	source address, destination address, data
DL-DATA	Confirm	source address, destination address, data, status
DL-FLOW CONTROL	Request	source address, destination address, amount of data
DL-FLOW CONTROL	Indication	source address, destination address, amount of data
DL-RESET	Request	source address, destination address, reason
DL-RESET	Indication	source address, destination address, reason
DL-RESET	Confirm	source address, destination address, reason
DL-DISCONNECT	Request	source address, destination address
DL-DISCONNECT	Indication	source address, destination address, reason
DL-DISCONNECT	Confirm	source address, destination address, status

The following points are to be noted:

- The priority is decided at the time of connection set-up. The data units transferred during the lifetime of the connection have this priority.
- The status parameter associated with DL-DATA service indicates whether the data unit was transferred successfully to the LLC entity at the other end or not.
- The flow control is exercised independently by the LLC and the user entity, *i.e.* there is no correspondence between DL-FLOW CONTROL request and DL-FLOW CONTROL indication primitives. Flow control is exercised by indicating the amount of user data that may be passed.
- Reset causes all the undelivered data units to be discarded. The responsibility of recovery is with the user.
- Except flow control, all the services are confirmed services. The confirmation is based on the acknowledgement from the remote LLC entity as there is no response primitive in LLC service (Figure 10.14).

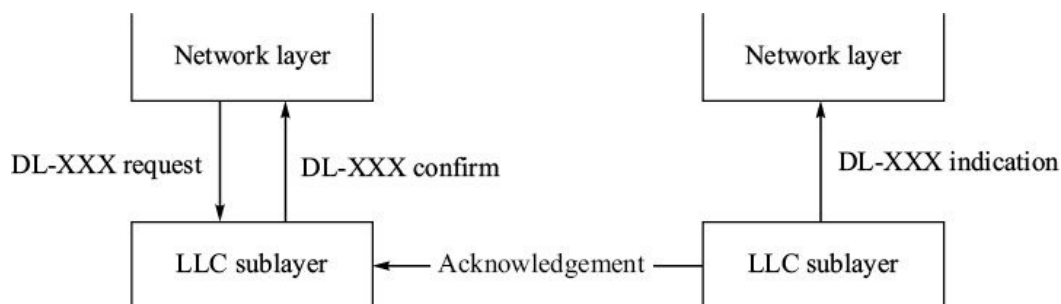


Figure 10.14 Connection-mode service of LLC sublayer.

Type 3—Acknowledged connectionless-mode service. Acknowledged connectionless-mode service allows an LLC user to request an immediate acknowledgement to a transmission. Data transfer is in connectionless mode and

each N-PDU must be acknowledged in the form of status indication across the service interface before the next N-PDU is handed over. The status indication is based on the acknowledgement received by the local LLC entity from the remote LLC entity as there is no response primitive across the service interface. The primitives used for this service are DL-DATA-ACK request, DL-DATA-ACK indication and DL-DATA-ACK-STATUS indication.

Acknowledged connectionless-mode service also provides the poll and response service. A user can request (poll) a PDU from a remote station. The primitives for this service are DL-REPLY request, DL-REPLY indication, and DL-REPLY-STATUS indication.

10.6.2 LLC Protocol

LLC protocol is modelled on the HDLC protocol. Extended asynchronous balanced mode is used in the LLC sublayer. The LLC PDU is shown in Figure 10.15. It contains the addresses of the source data link layer service access point, the destination data link layer service access point, control field, and information field. The information field contains the N-PDU.

Note that the usual frame check sequence field for detection of transmission errors is missing. Error detection function is the responsibility of the MAC sublayer and, therefore, this field is provided in the MAC frame.

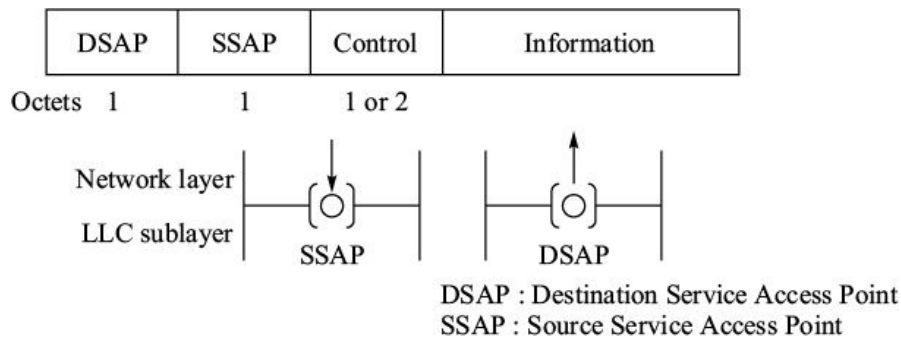


Figure 10.15 Format of LLC PDU.

DSAP/SSAP fields. Destination Service Access Point (DSAP) and Source Service Access Point (SSAP) fields are one octet long and contain respective addresses. Each of these fields is further subdivided (Figure 10.16).

- DSAP field contains U (user) bit and I/G (individual/group) bit in addition to six-bit destination service access point address.
- I/G bit divides address space of 256 addresses into
 - individual addresses (I/G = 0) and

- group addresses (I/G = 1).
- U bit further partitions the address space into two parts
 - addresses for specific network layer protocols reserved by IEEE (U = 1), and
 - addresses for vendor specific network layer protocols (U = 0).
- SSAP field contains U bit and C/R (command/response) bit in addition to six bits of source service access point address. Being source address, group addressing does not make sense and therefore there is no I/G bit. U bit is used as in DSAP.

Recall that address field in HDLC protocol determined whether a frame contained

P bit (for commands) or F bit (for responses). LLC frame contains both source and destination addresses and therefore we cannot adopt the same scheme. In LLC, C/R bit identifies whether a frame is a command (C/R = 0) or response (C/R = 1).

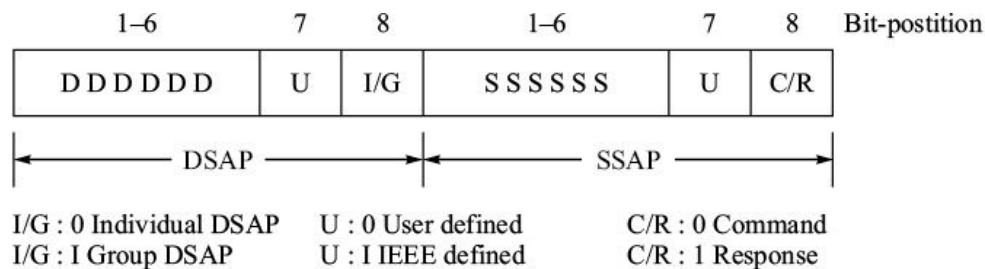


Figure 10.16 Format of DSAP and SSAP fields.

Some examples of DSAP addresses are given below. Note that the addressee is a network layer protocol. I/G bit is shown as 0 in the following examples:

- Hex 06 (0000 0110) : DOD IP (U = 1)
- Hex AA (1010 1010) : TCP/IP SNAP (U = 1)
- Hex FE (1111 1110) : ISO network layer (U = 1)
- Hex 42 (0100 0010) : IEEE 802.1d Spanning tree protocol (U = 1)
- Hex F0 (1111 0000) : NetBios (U = 0)
- Hex E0 (1110 0000) : Novell (U = 0)
- Hex 04 (0000 0100) : SNA path control (U = 0)

Control field. Format of the LLC control field is same as the HDLC control field. It is one or two octets long. In U-frames, it is one octet long and in I-and S-frames it is two octets long (Figure 10.17). The P/F bit is identified (P in

commands and F in responses) by the C/R bit in SSAP field (Figure 10.16).

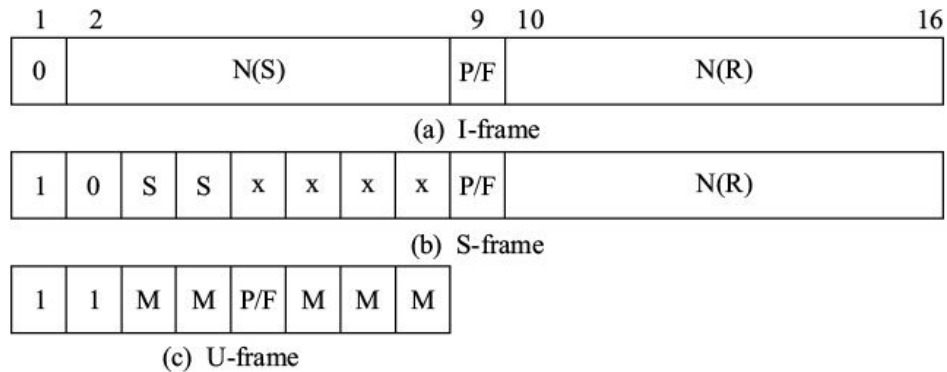


Figure 10.17 Control field of LLC frame.

Various types of frames used by LLC protocol are indicated in Table 10.2. The codes used for identifying the frames are the same as used in HDLC protocol.

TABLE 10.2 LLC Commands and Responses			
Service type	Type 1	Type 2	Type 3
Commands	UI, XID, TEST I, RR, RNR, REJ, SABME, DISC, REJ		AC0, AC1
Responses	XID, TEST I, RR, RNR, UA, FRMR, DM		AC0, AC1

10.6.3 LLC Procedures

Asynchronous balanced mode of data transfer was discussed at length in Chapter 9. The same operation is applicable to the LLC entities. The procedure in brief is given as follows:

- In the connectionless-mode service (Type 1 service), UI commands are used for exchanging information, XID commands/responses are used to exchange operation parameters such as window size, and TEST command is used for loopback testing. The receiving LLC entity returns TEST response as soon as possible.
- For the connection-mode service, SABME and DISC commands are used to

establish and release the connection. Application of other commands and responses was explained in the previous chapter.

- For the acknowledged connectionless-service, two new unnumbered information frames, AC0 and AC1 have been defined. AC stands for Acknowledged Connectionless. AC0 and AC1 are used alternatively as described earlier in stop-and-wait mechanism in Chapter 8. The sender alternates the use of AC0 and AC1 command frames, which the receiver replies with AC0 and AC1 responses respectively. The control fields of AC0 and AC1 frames are 1110(P/F)110 and 1110(P/F)111.

10.7 MEDIA ACCESS CONTROL (MAC) SUBLAYER

The media access control, error detection, and station addressing are the three basic functions of the MAC sublayer. It provides service to the LLC sublayer and receives service from the physical layer below it.

10.7.1 MAC Service

The MAC sublayer provides connectionless-mode service for the transfer of LLC PDUs. The MAC sublayer service primitives are:

- MA-UNITDATA request (source address, destination address, data, service class, priority).
- MA-UNITDATA indication (source address, destination address, data, reception status, service class, priority).
- MA-UNITDATA-STATUS confirm (source address, destination address, transmission status, provided service class, provided priority).

The address parameters are the MAC service access point addresses which also identify the stations on the LAN. MA-UNITDATA-STATUS indicates the transmission status of LLC-PDU to the LLC entity (Figure 10.18).

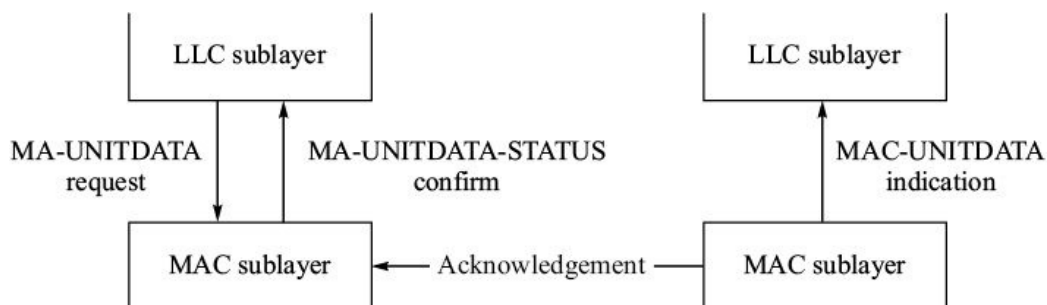


Figure 10.18 Connection-mode service of MAC sublayer.

10.7.2 MAC Protocol

As mentioned earlier, the MAC frame consists of a header, data field and a trailer. The data field contains LLC-PDU. The header specifies source and destination station addresses. It also contains a control field for media access control and a frame delimiter to identify start of the frame. The trailer contains check bits for error detection and an end delimiter to mark end of the frame.

The frame formats, contents of various fields and MAC protocols are different for different media access control mechanisms. IEEE has standardized the following MAC protocols:

- CSMA/CD IEEE 802.3
- Token bus IEEE 802.4
- Token ring IEEE 802.5.

We will discuss these media access control protocols and the physical layer specifications in the next two chapters.

10.8 TRANSMISSION MEDIA FOR LOCAL AREA NETWORKS

In a local area network, the interconnecting transmission medium can be a twisted pair cable or a coaxial cable or an optical fibre cable. Choice of transmission medium depends on several factors:

Bandwidth. Bandwidth of the transmission medium determines the maximum data rate which can be handled by it. The transmission medium bandwidth should not become a bottleneck in achieving the required data rate. It must be kept in mind that bandwidth of transmission medium is function of the length of the medium, *e.g.* it may be possible to achieve very high data rates on a low cost twisted pair but then the maximum length of one transmission segment is limited to not more than a few meters.

Connectivity. Some transmission media are suitable for broadcast mode of operation and point-to-multipoint links, while others are better suited for point-to-point links. For example, optical fibre is suitable for point-to-point links.

Geographic coverage. Geographic coverage of a LAN depends on the characteristics of the transmission medium used for carrying the electrical signals. The transmission parameters of concern are attenuation, group delay and propagation time. Since all these parameters are distance dependent, they determine the limits of geographic coverage of a LAN.

Noise immunity. Ideally, the transmission medium chosen for LAN should be free from any interference from the outside sources but, in practice, it is not possible. The degree of immunity depends on transmission medium to medium. For example, optical fibres have least noise interference. LAN cabling is usually done in ducts that carry power cables also. LANs based on copper cables (twisted pair, coaxial cables) laid in the vicinity of power cables have susceptibility to interference from power cables.

Cost. Cost is the major consideration in choice of transmission medium for LANs. Cost

of transmission medium should be considered along with the associated equipment and its installation cost. Therefore, an overall view of the cost structure is more important than the cost of transmission medium alone.

The alternatives for transmission media for local area networks are as follows:

- Shielded or unshielded twisted copper pair cable
- Baseband or broadband coaxial cable
- Optical fibre cable.

General features of these transmission media are described below. Their actual use in local area network is dependent on LAN technologies. We will discuss their use for these technologies in the next two chapters.

10.8.1 Twisted Copper Pair Cable

Twisted copper pair cable is suitable for point-to-point links and point-to-multipoint links. Tapping a twisted pair for adding a new station is not easy without disturbing a working network. CAT 3 UTP copper cable is the most inexpensive transmission medium. But it has limitation of data rate. It can be used for very small local area networks. CAT 5 UTP cable is higher data rates for a small number of devices. Shielded twisted pair cables support higher data rates over longer LAN segments but are relatively expensive.

10.8.2 Coaxial Cables

Coaxial cable is a widely used transmission medium in local area networks. It has low loss, high bandwidth, and low susceptibility to external noise and cross talk. 50 ohm and 75 ohm CATV coaxial cables are popular in local area networks. The 50 ohm coaxial cable is called *baseband cable* because digital signals are transmitted without any carrier modulation. The outer conductor of the baseband cable is metallic braid. The 75 ohm coaxial cable is called *broadband cable* as it has large bandwidth. It has solid outer conductor. Digital signals are transmitted on this cable as modulated carriers. For bidirectional transmission, the frequency band is divided into inbound and outbound frequency bands. The frequency translation takes place at the head end.

Geographic coverage of the coaxial cable is much larger than the twisted pair cable. Cost-wise, coaxial cable is more expansive than twisted pair cable but the overall cost of the installed coaxial cable is marginally different due to significant cable installation cost in either case.

10.8.3 Optical Fibre Cable

With optical fibre cables it is possible to realize very high data rates (~gigabits per second) over much larger distances than coaxial cables. Cost of optical fibre cable has come down significantly over the last decade, and these cables are increasingly being used in the local area networks. All the topologies, star, ring, and bus are possible with optical fibre cables. Compared to coaxial cables, optical fibre cables offer greater bandwidth, smaller size, lighter weight, lower loss, high noise immunity, and enhanced security.

SUMMARY

Local area networks provide a high speed and high throughput solution to the networking requirements within a small geographic area. The physical topology of a LAN can take the shape of a bus, a ring or a star. In bus topology, a single transmission medium interconnects all the stations. The bus operates in broadcast mode and the stations can pick the data units meant for them.

A ring network consists of a number of transmission links joined together in the form of a ring through ring interface units. The transmission is unidirectional on the ring. When a circulating data unit passes through a station, a copy of the data unit is retained if it is meant for the station.

Local area networks need a media access control mechanism as the

interconnecting transmission medium is shared by all the stations. For the bus topology, the two methods have been standardized, CSMA/CD and token passing. Ring networks use a token passing mechanism for media access control. Media Access Control (MAC) methods are implemented as a sublayer of the data link layer. The MAC sublayer also carries out error detection and station addressing functions. The other sublayer of the data link layer is the Logical Link Control (LLC) sublayer which carries out error control, flow control, sequencing, and user addressing functions. LLC sublayer offers connectionless and connection-mode data link services. IEEE has developed LAN standards which include protocols and services of the LLC and MAC sublayers. These standards also cover the physical layer specifications.

Transmission media of the local area networks can be twisted pair (shielded or unshielded) cable, coaxial cable or optical fibre. Choice of the transmission medium depends on several factors such as geographic coverage, data rate, number of stations, topology, *etc.*

EXERCISES

1. The following terms are associated with one of the three basic LAN topologies, bus, ring, and star. Indicate the topology against each of them:
 - (a) Wire centre
 - (b) Central controller
2.
 - (c) Remodulator
 - (d) Bypass relay.
3. Indicate the LAN topology against the following characteristics:
 - (a) Unidirectional transmission
 - (b) Least length of transmission media
 - (c) Single point of failure.
4. If the ring length is 1000 meters and speed of propagation is 2×10^8 metres,
 - (a) how much time a frame of 1000 bits will take to round the ring? Assume bit rate is 1 Mbps.
 - (b) Will the frame arrive back to the station before the station completes its transmission?
 - (c) What should the minimum size of the ring so that leading edge of the frame does not return back before the station completes its transmission?
5. RIU in ring topology adds a delay of 1 bit. If the bit rate is 1 Mbps, and the

signal propagation speed is $2 \cdot 10^5$ km/s, what is the equivalent length added by each RIU to the ring?

6. A baseband bus of length 1 km operates at 10 Mbps. What is the time required to send a frame of 1000 bits from a station A at one end of the bus to the other end of the bus?
7. Can HDLC protocol is used in place of LLC? If not, what is lacking?
8. Choose the correct answer:
 - (a) Content errors in a frame are detected by
 - (i) LLC sublayer
 - (ii) MAC sublayer
 - (iii) Physical layer.
 - (b) A station on the LAN is identified by its
 - (i) MAC address
 - (ii) LLC address
 - (iii) Network address.
 - (c) When CRC in the trailer of a frame detects an error, the respective sublayer
 - (i) discards the frame
 - (ii) requests for its retransmission
 - (iii) corrects the error.
 - (d) Connectionless (Type 1) LLC services use
 - (i) numbered information frames to send user data
 - (ii) unnumbered information frames to send user data
 - (iii) AC0/AC1 frames to send user data.

11

IEEE 802.3 Ethernets

Ethernet is the most widely used local area network in the industry today. We have ethernet technologies that operate at 10 gigabit/s bit rates. We examine the evolution of ethernet technology from its basics to the current state in this chapter. We begin with ALOHA, the contention access method that was the starting point of ethernet. We examine its throughput and the various back-off mechanisms used for improving the throughput. MAC frame formats of IEEE 802.3 Ethernet and Ethernet (DIX) are examined next. With this as the foundation, we move over to study of various types of ethernet LANs ranging from 10 Mbps to 1 Gbps. We examine the physical layer characteristics of each type of LAN in detail. 100 Mbps and higher bit rate LANs use auto-negotiation mechanism that enables the end stations to negotiate the operational parameters. We close the chapter with discussion on auto-negotiation mechanism.

11.1 CONTENTION ACCESS

In contention access methods there is no scheduled time or sequence for the stations to transmit the data frames on the medium. They contend for the use of the medium. It is, therefore, quite likely that more than one station will transmit simultaneously and the data frames will 'collide'. There are several ways of reducing the likelihood of these collisions. Carrier Sense Multiple Access/Collision Detection (CSMA/CD) is the most commonly used contention access method used in the local area networks. To understand this method, we must start from the first contention access method—Pure ALOHA, and go through its various variants.

11.1.1 Pure ALOHA

ALOHA contention access mechanism derives its name from its first

implementation in ALOHA Packet Radio network that was built in 1970s to connect the various campuses of University of Hawaii. The network was based on a single radio channel for several stations to communicate with each other. There have been several variants of ALOHA subsequently. The original contention access mechanism is, therefore, referred to as Pure ALOHA. The basic scheme is as follows:

- All the stations share a common radio channel for transmitting their data frames.
- A station can transmit its data frame on the radio channel whenever it wants. There is no pre-assigned time or sequence in which the stations transmit.
- When a transmission is in progress, if another station initiates its transmission, collision (overlapping) of the two transmissions occurs. In other words, the two transmissions corrupt each other.
- A mechanism to detect collision is established (e.g. carrier detection). When a transmission is in progress, if another carrier is detected, collision is assumed to have occurred and the data frame is retransmitted.

Figure 11.1 shows transmission of data frames by four stations. Data frames A, B, D, E, and F get corrupted during transmission due to collisions with other frames. But there are instances when there is single transmission on the channel and it reaches the destination without any collision. Data frames C, G, H, and I are successfully transmitted.

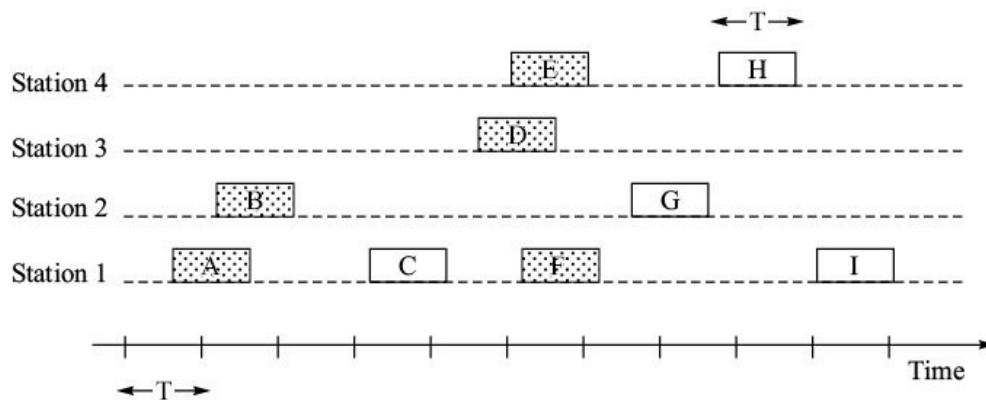


Figure 11.1 Frame transmission in pure ALOHA.

11.1.2 Throughput of Pure ALOHA Channel

Throughput S of a channel is defined as average number of successful

transmission of the data frames on the channel per unit time. It is usually expressed as percentage of carrying capacity of the channel. To calculate the throughput of pure ALOHA channel, let us consider a simple communication model. There are N stations that send data frames to the base station on using a shared communication channel (Figure 11.2).

We assume that

- all the data frames have the same size and each frame takes time T to transmit, and
- each station generates frames independent of other stations.

Transmit time T is frame size divided by bit rate. For the sake of convenience, we take the time to transmit a frame (T) as our unit of time. Clearly, we can send on the channel at the most one frame in time T . Therefore, the channel capacity

is one frame per time unit (T). When

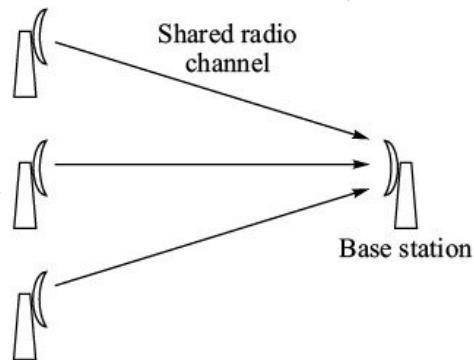


Figure 11.2 Aloha network.

collisions occur, some of the transmissions are lost and part of available channel time is

wasted. This results in the value of throughput S always less than one. For example, throughput

is equal to $4/10 = 0.4$ in Figure 11.1, as only four frames are successfully transmitted in

time $10T$.

The average number of frames generated in the network in time T by the stations is called load (G). For example, in Figure 11.1, G is $9/10 = 0.9$ as nine frames are generated in time

$10T$. G can have any value depending on number of stations and how frequently they generate

the frames. Since there is only one channel having capacity equal to one, there are many

collisions when G is more than 1. When G is low, much less than 1, there are few collisions and $S \approx G$.

Let us assume that probability of generating a data frame by a station in time T is s . Therefore, the average number of frames generated by the N stations in time T is given by $G = sN$.

The channel time is wasted whenever there is overlap of two frames. Overlap of any amount is always fatal and results in wasted channel time. Let us assume that a station generates a frame at time t . For successful transmission of the frame generated by the station at time t , it is necessary that there should not be any other frame after time $t - T$ and up to time $t + T$ (Figure 11.3).

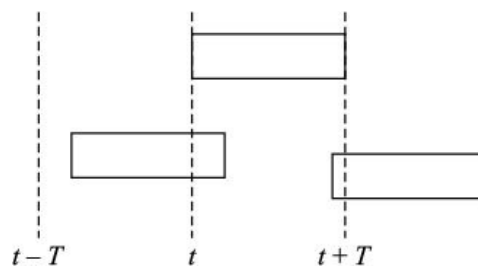


FIGURE 11.3 Collision of frames.

We can calculate the probability of a successful transmission by a station at time t as follows: Probability of no transmission by a station in $t - T$ to $t = (1 - s)$
 Probability of no transmission by $N - 1$ stations in $t - T$ to $t = (1 - s)^{N-1}$

Probability of no transmission by $N - 1$ stations in t to $t + T = (1 - s)^{N-1}$

Probability of no transmission by $N - 1$ stations in $t - T$ to $t + T = (1 - s)^{2(N-1)}$
 1) Probability of a successful transmission by a station = $s(1 - s)^{2(N-1)}$ Since

there are N stations, the throughput (S) is given by $S = sN(1 - s)^{2(N-1)}$ If N is large and s is small, we can rewrite the above expression for throughput as $S = Ge^{-2G}$, $G = sN$

The plot of throughput S with respect to load G is shown in Figure 11.4. The maximum throughput occurs at $G = 0.5$ and is equal to 0.184. This simply means that maximum throughput occurs when all the stations together generate on average one frame in time interval of $2T$. When frames are generated at this rate, only 18.4% of these frames are successfully transmitted.

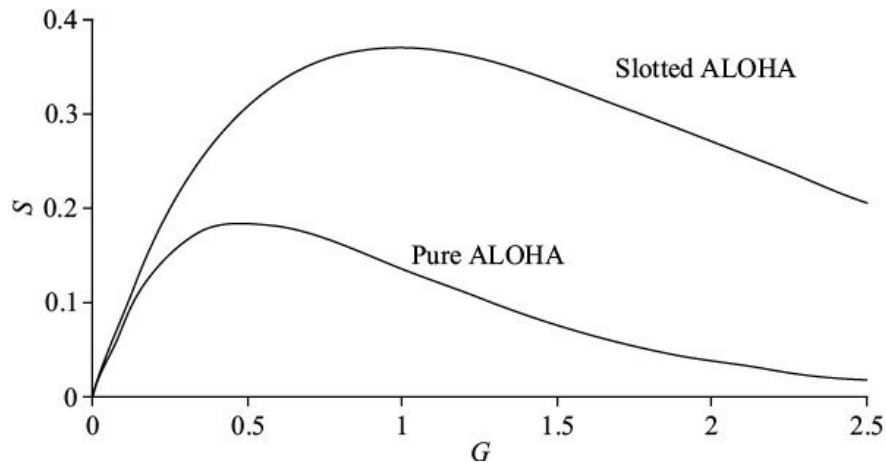


FIGURE 11.4 Throughput of pure and slotted ALOHA.

11.1.3 Slotted ALOHA

The maximum throughput of pure ALOHA access method is somewhat modest, the reason being large wasted time when a collision occurs (Figure 11.5). Note that even though there may have been only few bits of overlap, the whole of two frames is wasted in one collision.

Wasted channel time due to collisions can be reduced if all the transmissions are synchronized. The channel time is divided into time slots equal to time to transmit a frame (T) and the stations are allowed to transmit at specific instants of time so that all transmissions arrive aligned with the time slot boundaries (Figure 11.5). Collisions will still occur but the wasted time channel time is reduced to one time slot.

In slotted ALOHA, collision can take place only if there are more than one frame in a time slot. Since the probability of having a single frame in a time slot is $s(1 - s)^{N-1}$, the throughput S of slotted ALOHA for N stations is given by $S = sN(1 - s)^{N-1}$

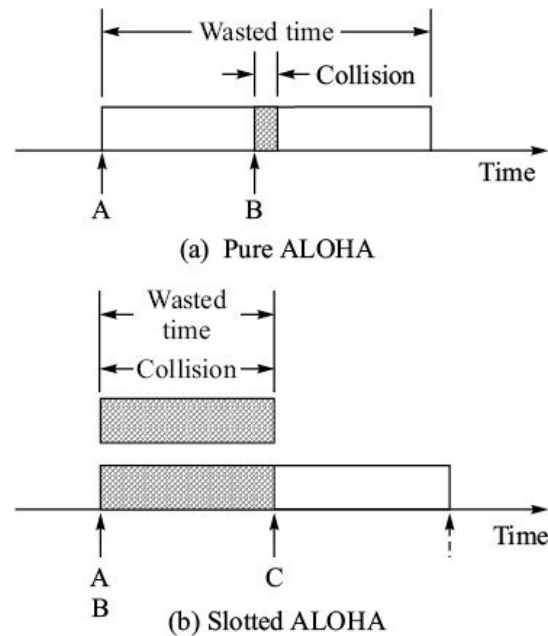


Figure 11.5 Wasted time due to collision.

If N is large and s is small, the throughput can be rewritten as $S = Ge^{-G}$, $G = sN$

Plot of S with respect to G for slotted ALOHA is shown in Figure 11.4. In this case the maximum throughput occurs at $G = 1$ and its value is 0.368.

11.2 CARRIER SENSE MULTIPLE ACCESS (CSMA)

In the ALOHA channel discussed above, the possibility of collision can be reduced if some discipline is built into the totally random access mechanism. If a station checks for the presence of any other carrier on the medium before starting its own transmission, a collision can be avoided. If there is a carrier on the channel, it does not commence its transmission. Carrier Sense Multiple Access (CSMA), as the name suggests, is based on this principle.

CSMA is widely used in local area networks. In the discussion that follows, we will be using the term carrier despite the fact that most of the local area networks use baseband transmission instead of a carrier to transmit data frames. In baseband LANs, the term ‘sensing carrier’ connotes ‘detecting presence of a baseband signal’.

When a station has a frame to send and the transmission medium is free, it starts to transmit its frame. Other stations wanting to transmit their frames soon sense presence of this frame as the signal travels along the medium, and they defer their transmissions to avoid collisions. CSMA does not avoid collisions

altogether. Collision can still occur if a station commences its transmission before the first bit already transmitted by another station is heard by it. This happens because of the propagation delay. Provided propagation delay is low, CSMA is much better than ALOHA as we shall shortly see.

In CSMA, an algorithm is needed to specify when a station can transmit once the channel is found busy because there can be several stations waiting to transmit. There can be several approaches as described below.

11.2.1 Non-Persistent CSMA

In this scheme, a station having a frame to send, checks the medium and, if the medium is busy, it backs off for a random interval of time (Figure 11.6). It checks the medium again after expiry of the interval and if the medium is free, it transmits. There is likelihood of some wasted time when the channel is not in use by any station.

11.2.2 1-Persistent CSMA

In this scheme, when a station wants to transmit its frame, monitors the medium continuously until the medium is free and then it transmits immediately (Figure 11.6). The problem with this strategy is that if two or more stations are waiting to transmit, each station will transmit its frame as soon as it finds the medium free. As a result there will always be multiple frames on the medium and collision will occur. Maximum throughput of 1-persistent CSMA is 0.53.

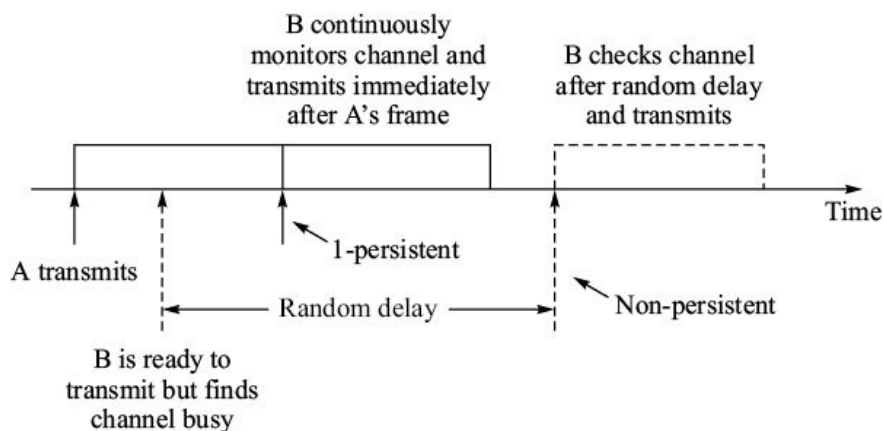


Figure 11.6 CSMA variants.

11.2.3 p-Persistent CSMA To reduce the probability of collision in 1-persistent CSMA, not all the waiting stations are allowed to transmit immediately after the medium is free. A waiting

station transmits with probability p if the medium is free. For example, if $p = 1/6$ and there are six stations waiting to transmit, on average only one of them will transmit and the rest will continue to wait. It is equivalent to throwing a dice and if a station gets six, it transmits. If two stations get six, then both will transmit and collision will take place. Likelihood of such occurrences can be reduced by reducing the transmission probability p . Optimized p -persistent CSMA can give maximum throughput of 0.8–0.9.

Figure 11.7 compares throughputs of various contention access schemes. We see that CSMA is always better than ALOHA.

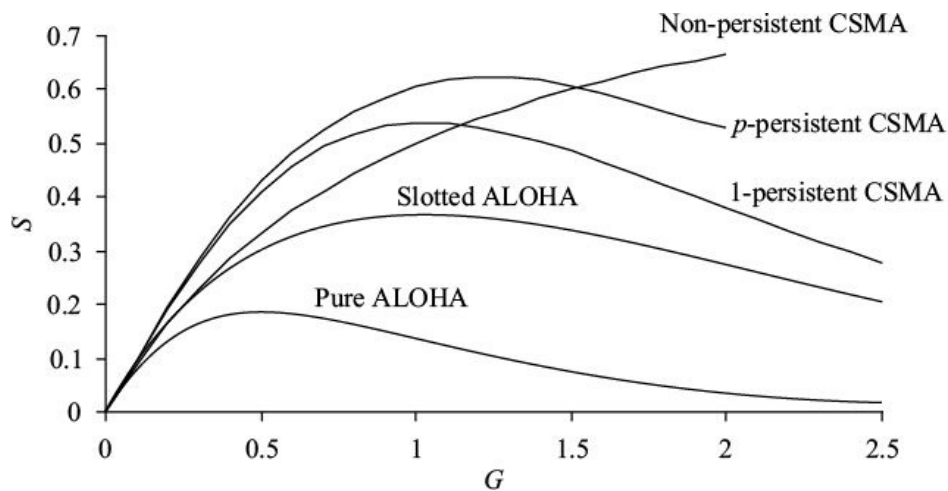


FIGURE 11.7 Throughput of various contention access methods.

11.3 CSMA/CD

The most commonly deployed multiple access technique in the local area networks is

CSMA/CD, where CD stands for Collision Detection. CSMA/CD specifications were developed jointly by DEC, Intel, and Xerox (DIX) in 1980. They called this network *Ethernet*. These specifications were later adopted by IEEE as their standard IEEE 802.3 in 1985. There are some differences in IEEE 802.3 and Ethernet (DIX) specifications as we shall see shortly. These differences make them incompatible to each other.

Throughout the rest of this chapter, the term *Ethernet* refers to the network compatible

to IEEE 802.3. Ethernet developed by DEC, Intel, and Xerox will be referred to as Ethernet (DIX).

11.3.1 Media Access Control in CSMA/CD

In the CSMA technique discussed above, a station continues transmission of a frame until the end of the frame even if a collision occurs. Continuing transmission of a frame that has already suffered collision results in unnecessary wastage of channel time. In CSMA/CD, transmission of a frame is abandoned as soon as collision is detected and a jam signal is appended to the frame to alert the other stations. Figure 11.8 illustrates the basic operation of the scheme. Note that for the scheme to work properly it is necessary that

- a station should still be transmitting its frame when it realizes that collision has occurred,
- the stations do not attempt to transmit again immediately after a collision has occurred. Otherwise, the same frames will collide again.

The stations are given a random back off delay for retry. If collision repeats, back off delay is progressively increased. So the network adapts itself to the traffic. In Ethernet, the random back off delay for retry after collision is doubled on each retry up to 10 retries.

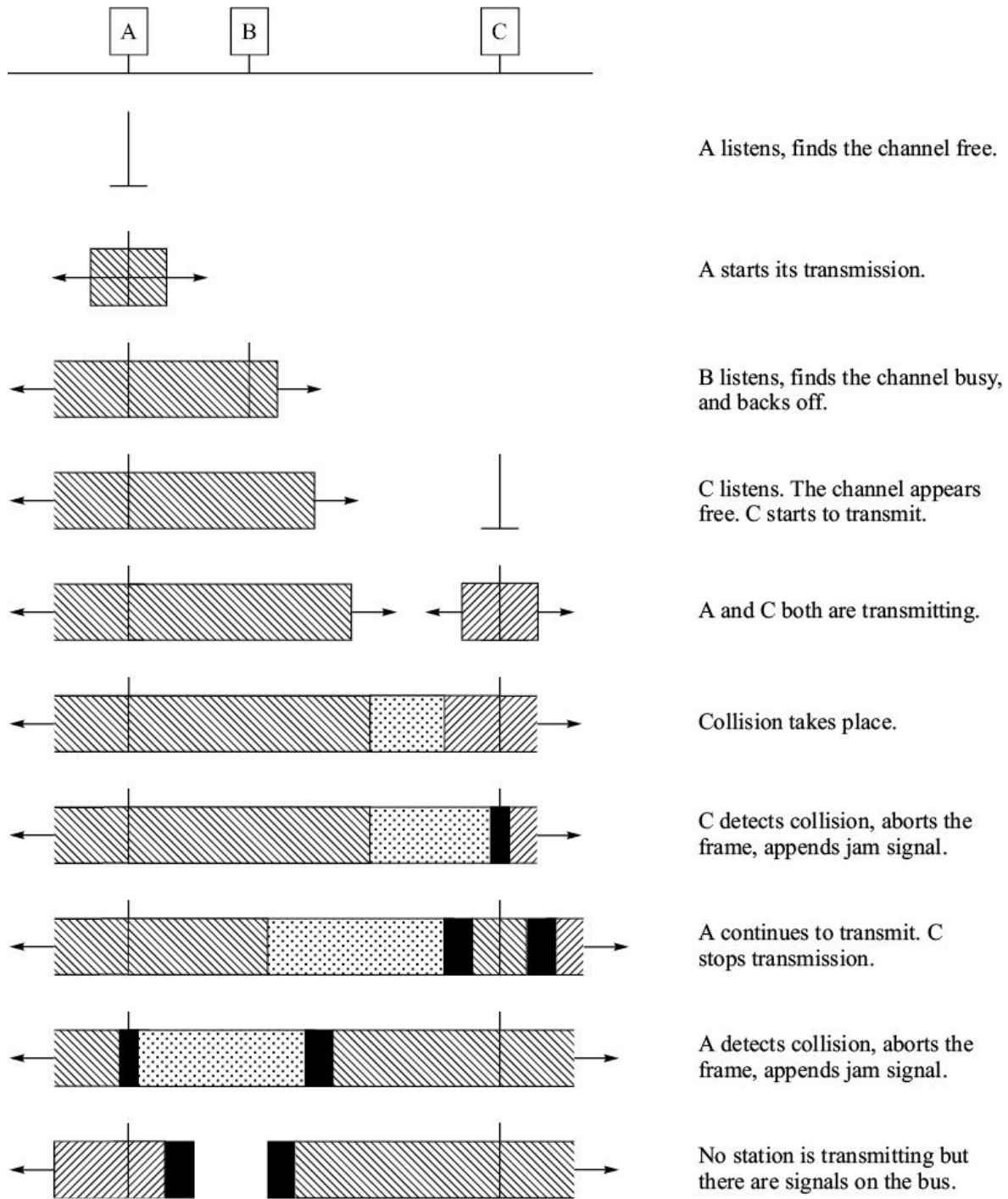


Figure 11.8 Collision detection in CSMA/CD.

By careful design, it is possible to achieve efficiencies of more than 90 per cent using CSMA/CD. Figure 11.9 summarizes the basic steps required for transmitting a frame.

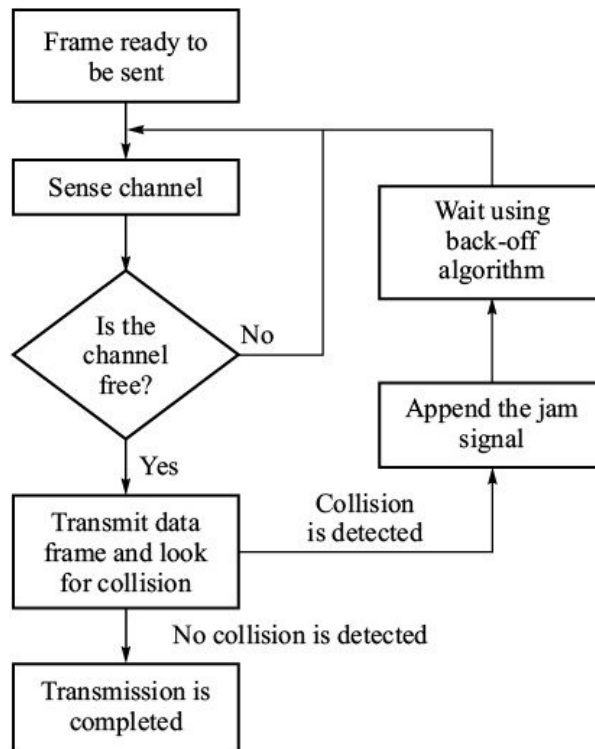


Figure 11.9 Frame transmission in Ethernet.

11.3.2 Maximum Cable Segment

As mentioned earlier, it is necessary that a station be transmitting if it is to detect a collision and append a jam signal to the aborted frame. This condition puts a limit on the minimum size of the frame and maximum end-to-end cable segment. If stations A and C in Figure 11.8 are at the extreme ends of a cable segment, the first indication of occurrence of the collision at C reaches A after time period equal to twice the propagation time of the cable segment. This period is called *collision window*. Note that:

- All collisions will occur within the collision window. Collision should not occur beyond the collision window since all the stations would have by this time sensed the presence of A's frame on the medium.
- If A is to append jam signal to its frame when a collision occurs, A should still be transmitting the frame. Therefore, the frame transmission time should be at least equal to the collision window.

Depending on the bit rate, transmission characteristics of the cable and the minimum frame size, maximum end-to-end cable segment length can be calculated. Ethernet specifies maximum length of cable segment as 2.5 km for

minimum frame size of 64 octets transmitted at 10 Mbps over a standard 50 ohms cable. It takes into account other delays like those associated with regenerative repeaters.

Example 11.1 Calculate maximum end-to-end cable segment length for an Ethernet LAN operating at 10 Mbps having minimum frame size of 64 octets. Assume propagation velocity of the medium as 2×10^5 km/s.

Solution

Frame transmission time = $64 \times 8 / 10^7 = 0.0512$ ms.

Maximum end-to-end propagation time = $0.0512 / 2 = 0.0256$ ms.

Maximum segment length = $2 \times 10^5 \times 0.0256 \times 10^{-3} = 5.12$ km.

11.3.3 MAC Frame Format (IEEE 802.3)

MAC frame format as per IEEE 802.3 standard is shown in Figure 11.10. It consists of the following fields:

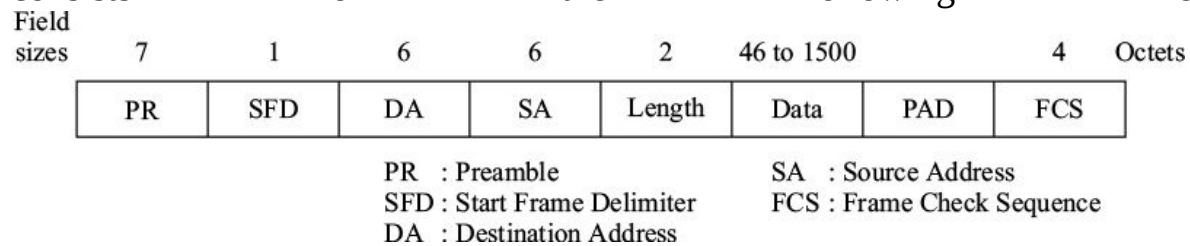


Figure 11.10 Format of IEEE 802.3 CSMA/CD frame.

Preamble. The preamble is of seven octets, each equal binary pattern 10101010. It enables bit synchronization.

Start frame delimiter (SFD). It is a one octet-long unique bit pattern (10101011) that marks the start of the frame.

Destination address (DA). The destination field identifies the station(s) which should receive the frame. The destination address field is 6 octets long (Figure 11.11a). Leftmost bit, I/G bit, indicates whether the address is individual or group address. If it is group address, the frame is accepted by all the members of the group. The second bit from the left, U/L bit, indicates whether the address pertains to universal addressing scheme administered by IEEE or it is a locally administered address. Destination address containing all 1s is broadcast address.

I/G = 0 for individual address

I/G = 1 for group address (multicast)

U/L = 0 for global address administered by IEEE

U/L = 1 for the locally administered address.

IEEE administered addresses consist of two parts. Octets 0, 1, and 2 are assigned by IEEE to each vendor of network components as vendor code. Octets 3, 4, and 5 are used by the vendors for numbering their network components.

Source address (SA). The source address field is 6 octets long (Figure 11.11b). It identifies the sending station. The least significant bit (47th bit) is not used because the source address is always individual. U/L bit has same significance as described above.

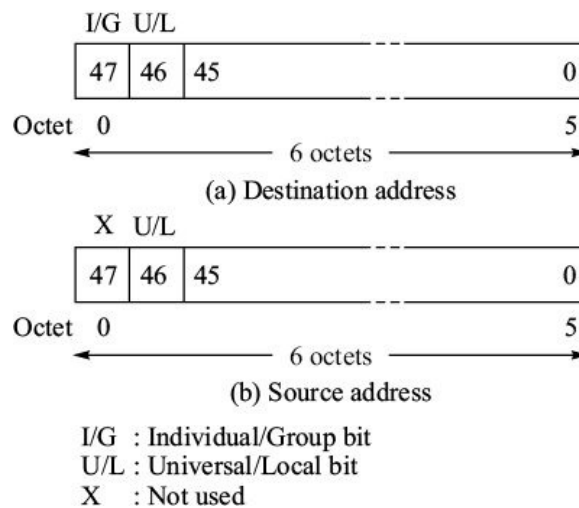


Figure 11.11 The address fields of IEEE 802.3.

Length (L). This field is two octets long and indicates the number of octets in the data field.

Data field. It can have 46 to 1500 octets.

PAD. If size of the data field is less than 46 as indicated by the length field, the PAD field makes up the difference to ensure the minimum size of the frame.

Frame check sequence (FCS). The frame check sequence is 4 octets long and contains the CRC code for error detection. The FCS is generated over DA, SA, length, data, and PAD fields using CRC-32 polynomial.

Jam signal is not part of the frame. It is appended by the transmitting station to a frame that suffers collision. It is a 32 bit binary sequence.

The frame size can be between 64 and 1518 octets (excluding preamble and SFD). Successive frames in Ethernet are separated by a time gap equivalent to 12 octets (9.6 msec. for 10 Mbps Ethernet). This ensures that the medium is free

of all the electrical signals associated with the preceding frame.

11.3.4 Format of Ethernet (DIX) Frame

The Ethernet (DIX) frame format is somewhat different from IEEE 802.3 frame. Ethernet (DIX) frame structure is shown in Figure 11.12. The differences in the two frame structures are as follows:

- The PR field of the Ethernet (DIX) frame is 8 octets long. It is same as PR and SFD fields of IEEE 802.3 taken together.

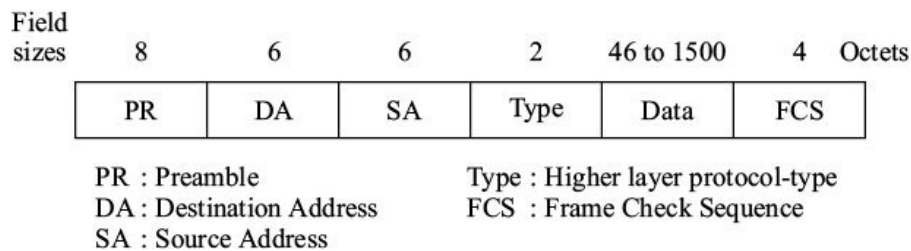


Figure 11.12 Format of Ethernet (DIX) frame.

- There is protocol-type field in Ethernet (DIX) that identifies the higher level protocol type associated with the frame. It determines how the data field is to be interpreted. Some examples for the protocol-type field are:

Novell	0 8138
IP	0 0800
SNMP	0 814C
IPv4	0 0800
IPv6	0 86DD
ARP	0 0806
RARP	0 8035

For protocol identification, IEEE 802.3 relies on Logical Link Control (LLC) sublayer just above it. As explained in the last chapter, SSAP and DSAP fields of LLC frame identify the higher layer protocol type. Ethernet (DIX) directly works with layer 3 protocols. It does not require LLC sublayer above it.

- Since there is no provision for the PAD field in Ethernet (DIX), the length field of IEEE 802.3 is not required.

Because of the above differences, IEEE 803.2 and Ethernet (DIX) stations attached to the same LAN cannot communicate with each other.

If we look at the formats of the Ethernet (DIX) frame and IEEE 802.3 frame, it is the protocol type field of Ethernet (DIX) and length field of IEEE 802.3 that distinguishes the two frames. Both are however two octets long. But the value of the field is limited to 1500 in IEEE 802.3. The assigned values of protocol type field in Ethernet (DIX) are greater than 1500.

Example 11.2 A octets of a frame in hexadecimal are given below. The preamble and start delimiter octets are not included. Identify the various fields. Is it an IEEE 802.3 frame or Ethernet (DIX) frame?

00 00 66 33 B5 49 00 00 A7 12 36 B7 08 00 AA AA 03 00 00 00 08 00 48 45
4C 50

Solution

DA	00 00 66 33 B5 49
SA	00 00 A7 12 36 B7
Type	08 00 Ethernet (DIX) frame because $0\ 0800 > 1500$
FCS	48 45 4C 50
Data	AA AA 03 00 00 00 08 00

11.3.5 Truncated Binary Exponential Back Off

Ethernet uses 1-persistence multiple access method described in the last section. A station waiting to transmit its frame initiates its transmission after the medium becomes free. If there are multiple stations waiting for medium to become free, collisions are bound to occur. Retransmission of frames that have suffered collision is controlled using a back-off mechanism called truncated binary exponential back off. It works as follows: (a) In the first try for a new frame a station sets frame retransmission counter $k = 0$, and transmits the frame.

(b) Every time a collision occurs, it retransmits the frame after a back off equal to nC where C is collision window and n is a random number in the range $[0, 2^{\min(k,10)}]$. The frame retransmission counter is incremented by 1.

(c) If a retransmission is successful, the station resets its retransmission counter to $k = 0$ for the next frame.

(d) When $k = 16$ and collision re-occurs, the station gives up further retransmission attempts and reports the error to the next higher layer.

Thus maximum number of attempts to transmit a frame are limited to 16. The

maximum back off for any retransmission is limited to 10 C, after which it levels off.

11.4 PHYSICAL TOPOLOGY OF ETHERNET LAN

Ethernet was originally conceived as LAN technology based on bus topology. But today it has become a standard interface for interconnecting data networking devices. Therefore, Ethernets have topological configurations that are combination of three basic structures:

- Bus
- Point-to-point
- Star.

11.4.1 Bus Topology

The original Ethernet LANs were implemented with a coaxial bus structure as shown in

Figure 11.13a. The bus configuration offers half-duplex working, *i.e.* a station can either transmit or receive a frame. Segment length of bus were limited due to cable attenuation and distortion characteristics. However, several bus segments could be interconnected with repeaters forming one bigger LAN having one collision domain. New Ethernet LANs are no longer connected in bus configuration.

11.4.2 Point-to-Point Topology

Point-to-point is the simplest basic structure in which there are only two network elements or end stations (Figure 11.13b). Point-to-point Ethernet link is an equivalent of point-to-point data link. The interconnection can be full duplex or half duplex. For full duplex connection two pairs of wires are required, one for each direction. There cannot be any collision in full duplex point-to-point Ethernet link. Physical transmission media used are twisted copper pairs or optical fibres. Optical fibre cable is used in place of twisted pair when distance is large and required data rate is high.

11.4.3 Star Topology

Since the early 1990s, the network configuration of choice has been the star-connected topology (Figure 11.13c). Instead of connecting the stations of a local area network to a bus, the stations converge on a central network element called a hub on point-to-point links. The hub is multipoint repeater which we describe in the next section. The central network element can also be an Ethernet switch which we will discuss in Chapter 13. The transmission medium used is twisted copper pairs or optical fibre cables. Several switches can be further interconnected to form a tree structure.

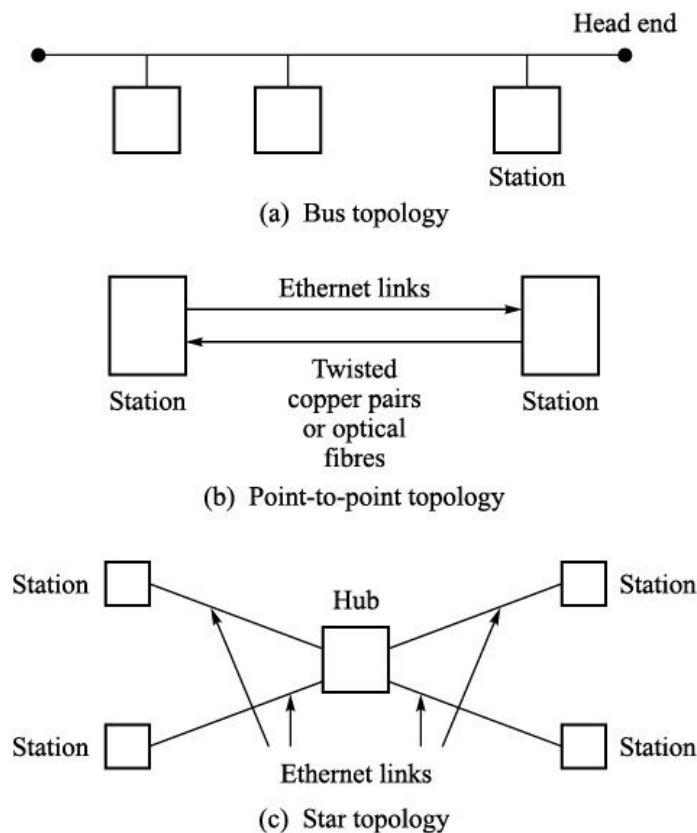


Figure 11.13 Ethernet topologies.

11.5 ETHERNET REPEATER

A repeater is needed to expand the geographic coverage of an ethernet LAN by cascading two cable segments. It can be a two port or multiport device (Figure 11.14). It regenerates the signals received on one of its ports and sends the regenerated signals to the other network segments connected on its other ports.

There is no buffering of the Ethernet frames in the repeater. There is very short delay associated with regeneration of electrical signals.

11.5.1 Collisions in a Repeater

A repeater has a shared media backplane ethernet bus. Collision occurs in the repeater when another frame arrives at the repeater when a frame is being regenerated by it. If collisions occur within the repeater, the repeater appends the jam signal to the frames that have collided. Thus, LANS interconnected through repeaters form one collision domain (Figure 11.14).

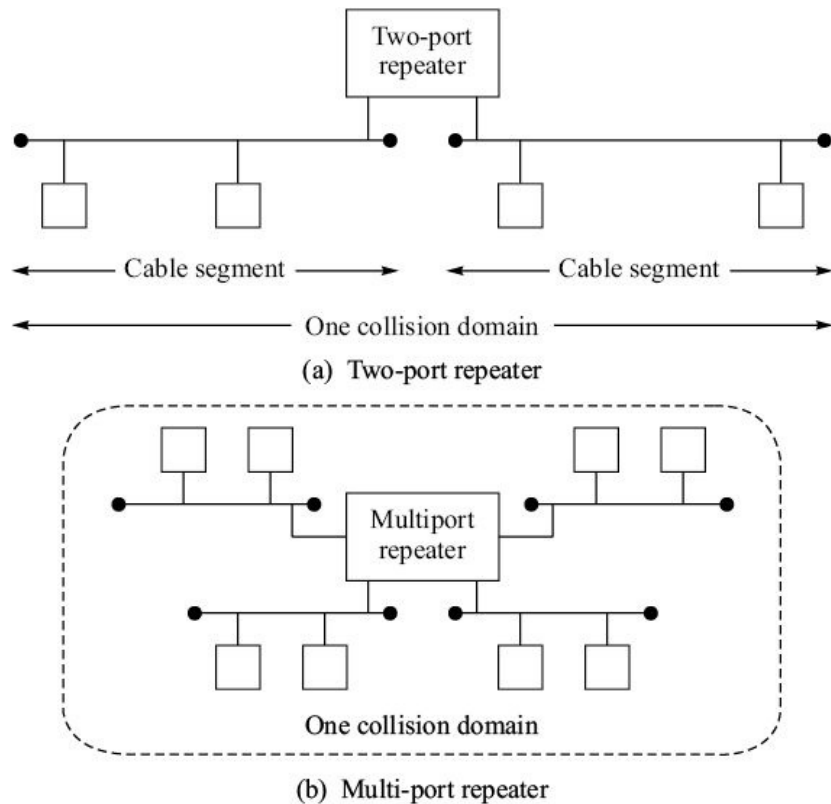


Figure 11.14 Ethernet repeaters.

11.5.2 Link Segments

Repeaters can be local or remote. The repeaters shown in Figure 11.14 are local repeaters. Local repeaters interconnect bus segments. Remote repeaters have ports for point-to-point ethernet links, usually called *link segments* (Figure 11.15). Link segments have either twisted pair copper cables or optical fibre cables. Separate cable pairs or optical fibres are provided for transmit and receive paths.

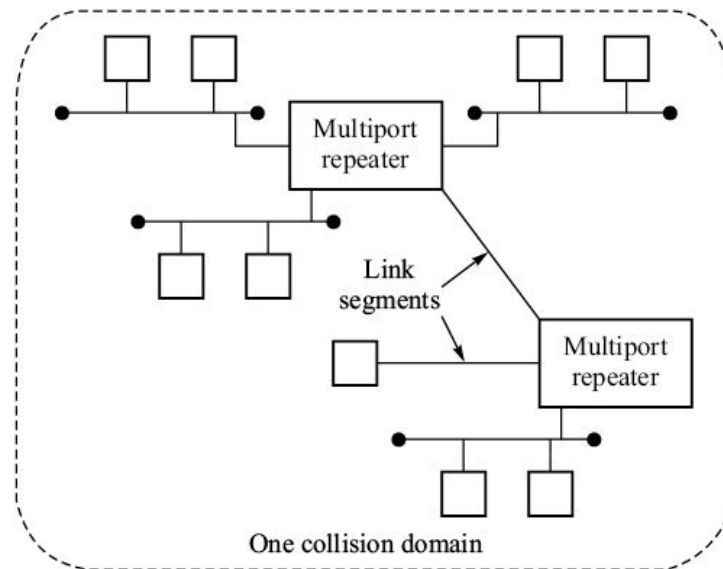


Figure 11.15 Link segments.

11.5.3 Ethernet Hubs

The current Ethernet LANs use multiport repeaters as ‘hubs’ (Figure 11.16). The hubs interconnect with stations and with other hubs directly using point-to-point ethernet link segments that form a star like structure.

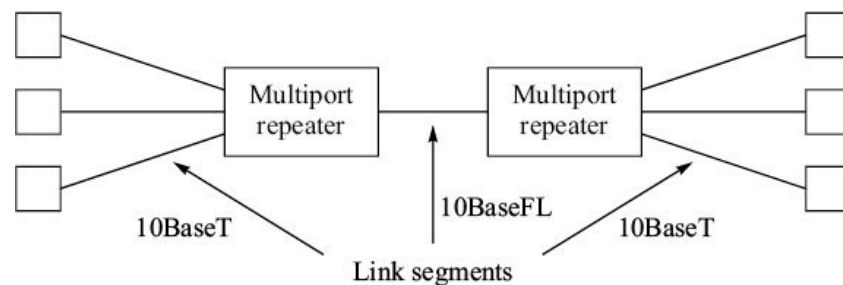


Figure 11.16 Ethernet hubs.

11.6 TYPES OF ETHERNETS

Ethernet is implemented in a variety of ways at the physical layer level. The choice is in terms of:

- Physical topology of the network—bus, point-to-point, star
- Transmission media—twisted pair, coaxial cable, optical fibre
- Bit rates—10, 100, and 1000 Mbps.

There are several Ethernet technologies that differ in terms of bit rate, topology, and media.

- 10 Mbps Ethernet
- 10Base5, 10Base2, 10BaseT, 10BaseFL, 10BaseFP, 10BaseFB, 10Broad36
- 100 Mbps Ethernet (Fast Ethernet)
- 100BaseTX, 100BaseFX, 100BaseT4, 100BaseT2
- 1000 Mbps Ethernet (Gigabit ethernet)
- 1000BaseT, 1000BaseCX, 1000BaseLX, 1000BaseSX

The nomenclature that is used to represent the technology consists of three parts—bit rate, signal type, and physical medium. The signal type is either ‘Base’ for baseband transmission or ‘Broad’ for broadband transmission using differential PSK. The physical medium part either represents maximum length of cable segment in units of 100 metres, or type of physical medium.

For example,

10Base5 : 10 Mbps ethernet, baseband transmission, maximum cable segment 500 m.

10BaseF : 10 Mbps ethernet, baseband transmission, optical fibre transmission medium.

11.6.1 Physical Layer of Ethernet LANs

While the MAC sublayer is same for all these technologies, the physical layer specifications are different. The physical layer of Ethernet is divided into several sublayers.

These sublayers are specific to the data rate, encoding used, and type of physical medium.

- Physical medium independent interface contains reconciliation functions that hide medium dependent differences from the MAC sublayer.
- Medium-Independent Interface (MII) provides separate transmit and receive data paths. These are bit serial for 10 Mbps LAN, nibble-serial (4 bits in parallel) for 100 Mbps LAN, and byte-serial (8 bits in parallel) for 1000 Mbps LAN.
- Physical media-dependent sublayer carries out encoding/decoding, synchronization, serial-parallel data conversion, frame encapsulation,

collision detection, and all other functions relating to signaling. It also incorporates auto-negotiation function which enables the end stations to negotiate half or full duplex operation and data rate so that a 10 Mbps station may work with a 100 Mbps station.

- The physical connector (also called Medium-Dependent Interface, MDI) forms the lowest sublayer and connects to the transmission medium.

Figure 11.17 shows the internal architecture of the physical layer of Ethernet LAN in general.

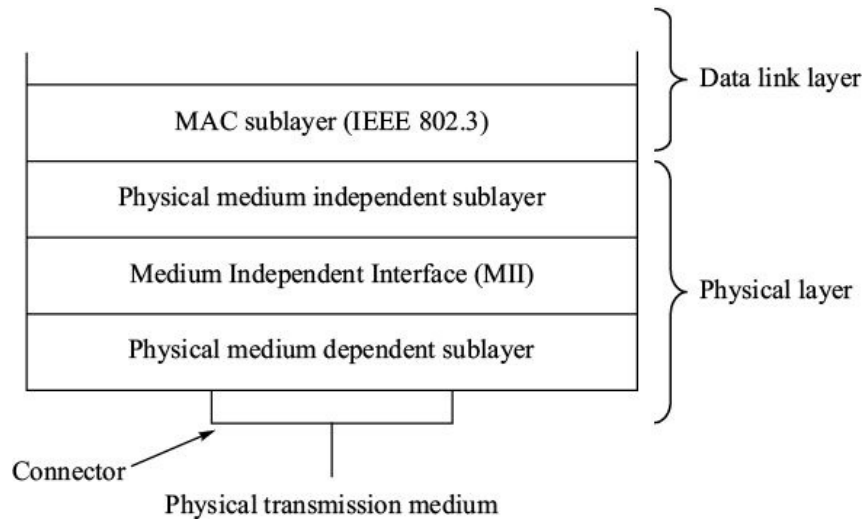


Figure 11.17 Architecture of the physical layer in Ethernet.

11.7 10 MBPS ETHERNETS

There are several types of implementations of 10 Mbps Ethernet. The primary differentiating feature is the transmitting media used for their implementation.

- Coaxial cable : 10Base5, 10Base2, 10Broad36
- Twisted copper pair : 10BaseT
- Optical fibre : 10BaseFL, 10BaseFP, 10BaseFB

The physical characteristics of coaxial cable and twisted copper pair Ethernet are summarized in Table 11.1.

TABLE 11.1 IEEE 802.3 10 Mbps Ethernet				
	10Base5	10Base2	10BaseT	10Broad36

Data rate (Mbps)	10	10	10	10
Signaling	Baseband	Baseband	Baseband	PSK
Max. segment length (m)	500	185	100	1875
Transmission medium	50 ohms	50 ohms	UTP, CAT 3	75 ohms
	coaxial (RG-8)	coaxial (RG-58)	or better	coaxial (RG-59)
Topology	Bus	Bus	Star	Dual bus
Transmission mode	Half duplex	Half duplex	Full duplex	Half duplex
Number of cable pairs	1	1	2	2

11.7.1 10Base5 (Thick Ethernet)

10Base5 is based on bus topology and uses baseband transmission with Manchester line code. Fifty ohms thick coaxial cable (RG-8) is used in the 10Base5 standard. Thick coaxial cable has 10 mm outer diameter and is relatively inflexible compared to thin coaxial cable used in 10Base2 Ethernet. The maximum length of a cable segment without using repeaters is limited to 500 metres (Figure 11.18). A maximum of 100 stations can be hooked on one cable segment of 500 metres. Minimum spacing between the stations is 2.5 metres. Since the collision domain of 10 Mbps Ethernet LAN is limited to 2500 metres diameter, maximum 5 cable segments using four repeaters can be cascaded. But only three segments out of the five can be populated with stations.

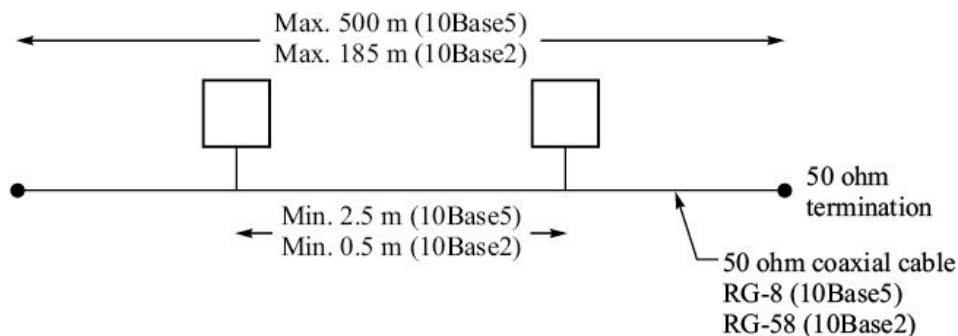


Figure 11.18 10Base5 Ethernet.

11.7.2 10Base2 (Thin Ethernet)

10Base2 is less expensive version of 10Base5 and is sometimes called *cheapernet*. It has bus topology and is based on RG-58 coaxial cable which is thinner, more flexible, and cheaper than the cable used in 10Base5. Baseband transmission with Manchester code is used. The maximum cable segment length

is 185 metres and maximum number of taps can be 30 per segment (Figure 11.18). Minimum spacing between adjacent stations is 0.5 metre.

11.7.3 10Broad36

10Broad36 uses 75 ohms CATV cable (RG-59) as bus with a remodulator at the head end. Differential PSK is used for transmitting data signals on 10Broad36 Ethernet. The maximum cable span is 3750 metres in two segments of 1875 metres from the head end. Other services such as TV and voice can also be integrated on the same cable using frequency division multiplexing. Broadband version of Ethernet was not successful and is not used today.

11.7.4 10BaseT

The 10 Mbps Ethernets discussed so far were based on bus topology, which is no longer used now. 10BaseT Ethernet is based on star topology where stations are connected to a hub (Figure 11.16). 10BaseT is also used as link segments between the repeaters or between stations (Figure 11.13b).

The physical medium is unshielded UTP, CAT 3 or better copper cable. Two pairs of the four pair cable are used with RJ-45 connectors at the ends. Each pair is configured as simplex link where transmission is in one direction. Maximum length of cable segment can be 100 metres. 10BaseT uses baseband transmission with Manchester line code.

10BaseT can support half duplex or full duplex operation. There are no collisions in full duplex transmission as separate pairs are used for transmit and receive and the links are point-to-point. In half duplex mode, the hub or end station can either transmit or receive data frames. Collision is detected when a data frame is received while the hub (or end station) is already transmitting a data frame.

Link test pulses. The 10BaseT standard includes a link test mechanism to ensure health of connection at the physical layer level. Immediately after power up, a station starts sending a 100 ns positive unipolar pulse every 16 8 ms (Figure 11.19). These pulses are called Normal Link Pulses (NLP). If the other end of the link is also powered up, it sends its own NLPs. Link is activated when each station receives valid NLPs from the other station. If a station does not receive NLP from the other end, it continues to send its NLP till it receives a valid NLP from the other end. Link fail condition is detected if the receiver does not receive a frame or link test pulse for more than 50 ms.

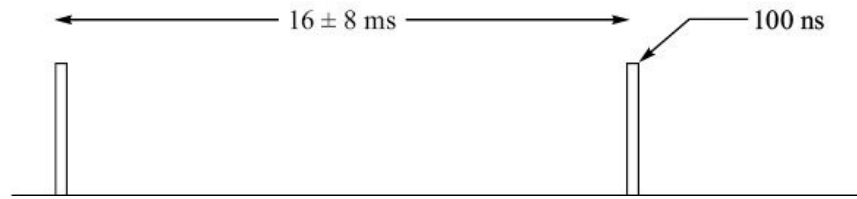


Figure 11.19 Normal link pulses (NLP) in 10BaseT Ethernet.

11.7.5 10BaseF

10BaseF is set of three optical fibre based Ethernets—10BaseFP, 10BaseFB, and 10BaseFL. It is used in either point-to-point or star topology. Two strands of multimode optical fibre cable are required, one for each direction of transmission. 10BaseFP is used with a passive hub in star topology. The maximum length of the segment is 1 km. 10BaseFL and 10BaseFB are used for the link segments. 10BaseFL is for the link between a station and an active hub. 10BaseFB is for backbone link between two active hubs. The maximum segment length is 2 km in either case.

11.8 FAST ETHERNET

Development of technology for high speed LANs was undertaken in 1993. Two sets of technologies for 100 Mbps LANs were standardized, one based on existing IEEE 802.3 and the other was newly developed, IEEE 802.12. The one based on existing IEEE 802.3 was called *fast Ethernet*. Since it was based on twisted pair, it was referred to as 100BaseT. It comprised three types of 100 Mbps LANs—100BaseT4, 100BaseT2, and 100BaseTX. All these LANs are based on twisted pair copper cable as transmission medium.

Later 100BaseTX technology was extended to optical fibres and called 100BaseFX. 100BaseTX and 100BaseFX have same encoding, decoding, clock recovery, and other control functions except that the signal types are different—electrical and optical. Together these two technologies are referred to as 100BaseX.

The new development IEEE 802.12 was called 100VG-AnyLAN. It is 100 Mbps LAN based on Voice Grade (VG) cable. The term ‘AnyLAN’ emphasizes that it can be configured to interconnect with various other LAN types. But 100VG-AnyLAN did not find much acceptance with the industry and failed miserably. We will therefore not discuss this technology in this text.

11.8.1 Additional Functions Required for 100 Mbps LANs

100BaseT uses the existing IEEE 802.3 Ethernet specifications and therefore the frame format at MAC sublayer remains same. It differs from 10BaseT in having some additional functionality at the MAC sublayer and the physical layer. Major differences are as follows:

- At 100 Mbps, flow control at MAC sublayer is also required. Flow control is implemented using pause control function.
- 100BaseT supports dual speeds of 10 Mbps and 100 Mbps. It also supports half and full duplex modes of operation. Therefore, it has inbuilt auto-negotiation function that determines speed (10 Mbps or 100 Mbps) and mode of operation (Full duplex or half duplex).
- More frequency-efficient line coding scheme than Manchester coding is required. The goal is to reduce the baud rate so that the frequency characteristics of the line signal are compatible to the transmission medium.
- For clock and bit stream recovery, the MAC frame is scrambled using random sequences.
- Block codes are used to expand the code space to enable error detection at the physical layer level. Receipt of invalid code words indicates there is an error.
- MAC frame is encapsulated at the physical layer level using start and end of data stream symbols.

At 100 Mbps, the time required to transmit a minimum-length frame of 64 bytes is one tenth of that required at 10 Mbps. This in turn means that the maximum network diameter specified for 10 Mbps needs to be reduced by a factor of ten. It is about 200 metres for 100 Mbps Ethernet LAN.

11.8.2 Physical Layer of Fast Ethernet

The physical layer for various fast Ethernet LANs is divided into sublayers that are independent of media and signal encoding, and sublayers that are media dependent (Figure 11.20).

Media-independent interface (MII) sublayer. MII sublayer provides generic interface between MAC sublayer above it and various physical media-dependent sublayers below it. It provides separate paths for upward-going and downward-

going bit streams. The paths are byte-serial (8-bits in parallel) or nibble-serial (4-bits in parallel) depending on type of the LAN. In 100BaseT, for example, MII interface is 20 pair shielded cable with 68 ohms impedance and having maximum length of 50 cm.

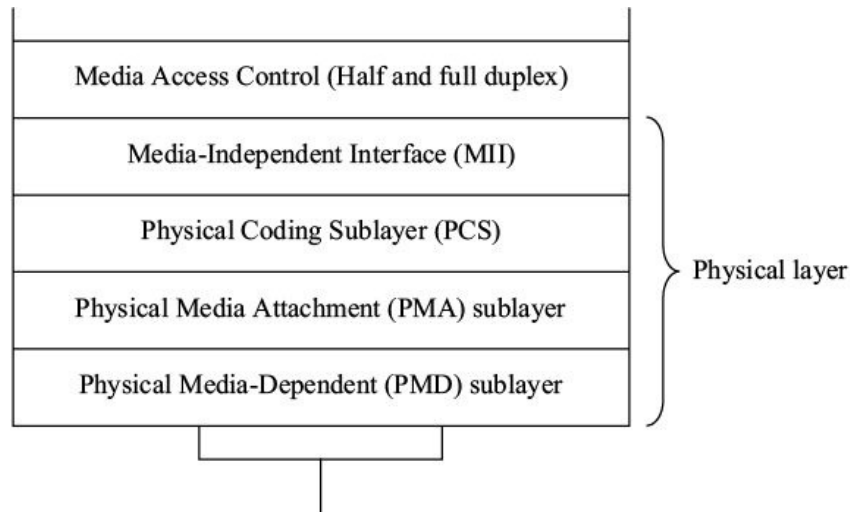


Figure 11.20 Architecture of the physical layer in fast Ethernet.

Physical coding sublayer (PCS). The media-dependent physical coding sublayer provides logic for encoding the bits received from MII sublayer and for decoding the bits received from the PMA sublayer below it.

Physical medium attachment (PMA) sublayer. Physical medium attachment sublayer serializes the code-groups (e.g. 5-bit code-group of 4B/5B block code used in 100BaseTX LAN) received from PCS sublayer above it. Similarly it converts incoming serial bit stream received from PMD sublayer into block of code-groups for decoding.

Physical media-dependent (PMD) sublayer. PMD sublayer contains transmitter, receiver, clock recovery, collision detection and auto-negotiation functions.

11.8.3 100BaseT4

100BaseT4 was developed to allow the 10BaseT networks to be upgraded to 100 Mbps without requiring the upgrade of existing four-pair CAT 3 UTP cable infrastructure. As a CAT 3 UTP pair cannot support transmission of 100 Mbps over a distance of any practical use, three UTP pairs out of the four are used for transmission at 100 Mbps in any one direction (Figure 11.21).

- Two pairs are configured for half duplex transmission.
- One pair is configured for simplex transmission.

The bit rate on each pair is reduced by a factor of three using three pairs of wires. Taken together all the four pair can support 100 Mbps half duplex transmission. Full duplex transmission is not possible. When frame transmission is going on in one direction, the simplex pair of the opposite direction is used for carrier sense and collision detection as we will shortly see.

Line code. If Manchester line code is used for transmission, the bit rate transmitted over each pair will be 33.33 Mbps which exceeds the 30 MHz upper limit set for use of CAT3 cables. Therefore, instead of binary transmission, 3-level ternary transmission is used. 8B/6T

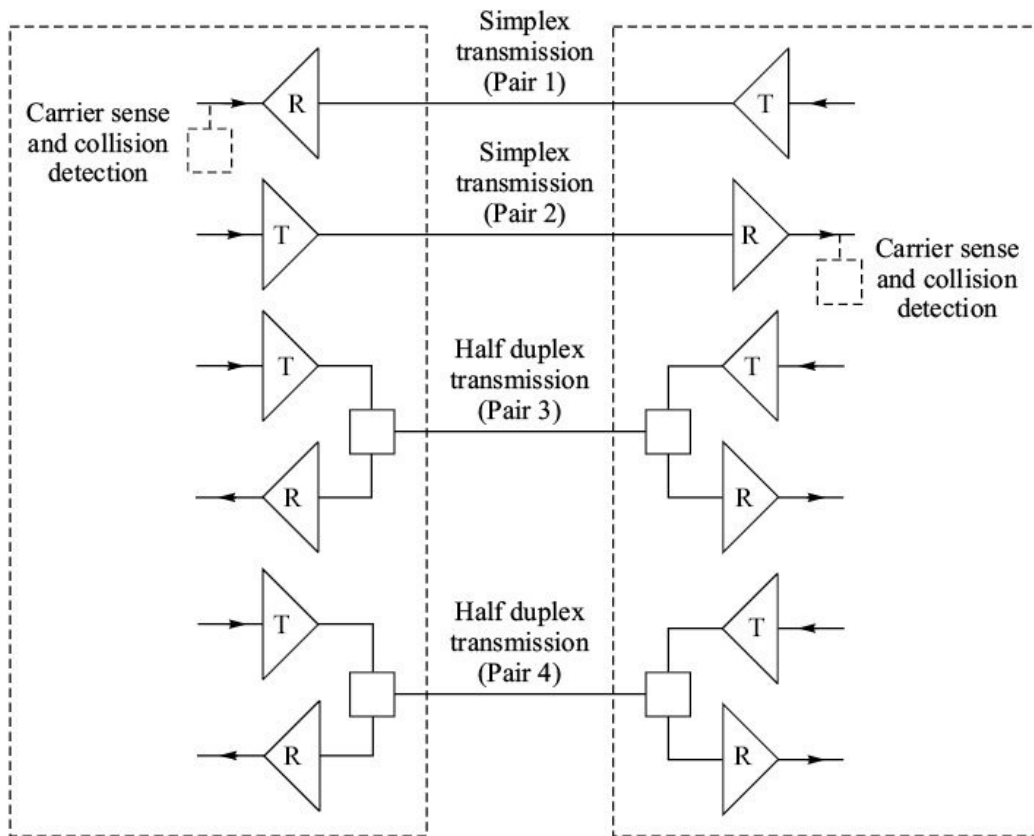


Figure 11.21 100BaseT4 on 4 pair CAT 3 UTP cable.

line encoding is used for this purpose. Each octet is mapped to 6 ternary symbols called 6T code-group (Figure 11.22). The baud rate of 6-ternary signal is 25 Mbaud which is well within 30 MHz limit.

$$\text{Bit rate} = 100 \text{ Mbps}$$

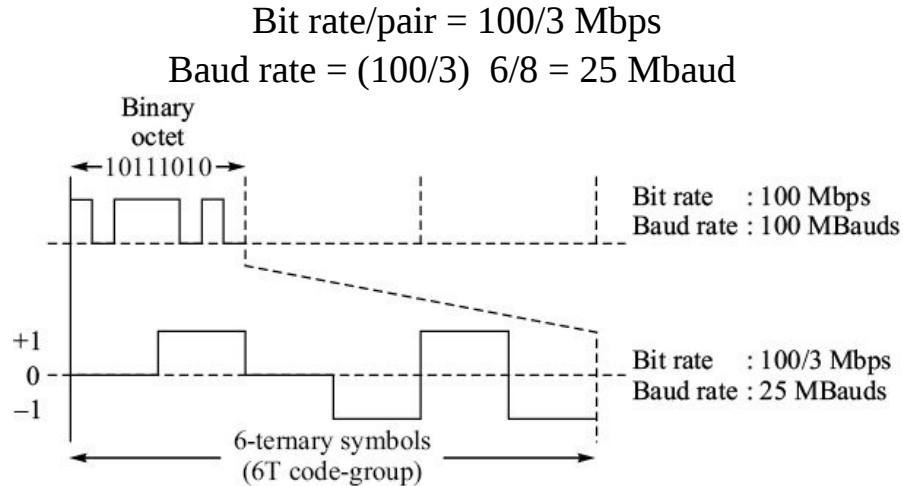


Figure 11.22 Binary to ternary conversion using 8B/6T line code.

Note that each 6T code-group that represents one octet, occupies time equivalent to three octets because the bit rate is reduced to one third. There are $729 (= 3^6)$ possible combinations of 6T code-groups. But we require only $256 (= 2^8)$ valid code-groups to represent complete set of 8-bit binary words. The valid 6T code-groups are selected based on the following considerations:

- To achieve DC balance, 6T code-groups with more +1s than -1s and more -1s than +1s should be avoided.
- To facilitate clock recovery at the receiver, the selected 6T code-groups should have sufficient signal transitions.

To meet the first requirement we select those combinations which have weight¹ of 0 or +1. There are 267 such combinations. To meet the second condition, we select out of 267, those combinations that have at least two signal transitions. We remove the combinations starting or ending with four consecutive zeroes. This leaves 256 6T code-groups required to represent 256 8-bit binary words.

DC balance. DC balance is required since all the copper pairs are inductively coupled through transformers. The transformers block the DC component of the signal. If the average signal level is not zero, the base line of signals on the copper pairs moves away from the zero level (Figure 11.23). This results in misinterpretation of the signal received at the distant end.

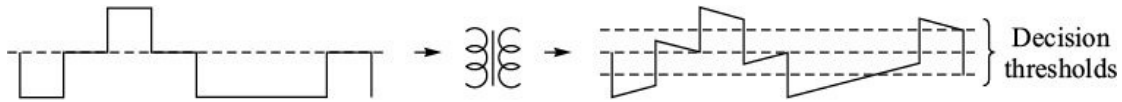


Figure 11.23 Base line wander due to transformer coupling.

The problem is resolved to a large extent by keeping the weight of the selected 6T code-groups 0 or +1. Further DC balance is achieved by maintaining a running sum of the weights of transmitted 6T code-groups. When a code-group with +1 weight is sent, the running sum is incremented by 1. When the next code-group with +1 weight is to be transmitted, the signal polarity is reversed. By reversing the polarity, the weight of transmitted code-group becomes -1 , which restores DC balance. For example, if ternary signal (0 +1 +1 +1 -1 -1) is to be sent when the running sum is +1, the transmitter actually sends code word (0 -1 -1 -1 +1 +1) having weight -1 on the line. The receiving end also maintains running sum counter. It inverts the received signals before decoding when running sum is +1.

Frame transmission. The octets of a frame are transmitted on the three copper pairs in round robin sequence (Figure 11.24). Note that line coding can commence only after one full octet is ready for transmission. Each 6T code-group occupies time slot equivalent to three 100 Mbps octets. The 6T code-groups on the three pairs are staggered in time by duration of one octet, which corresponds to delay equivalent to 2 ternary symbols (2T).

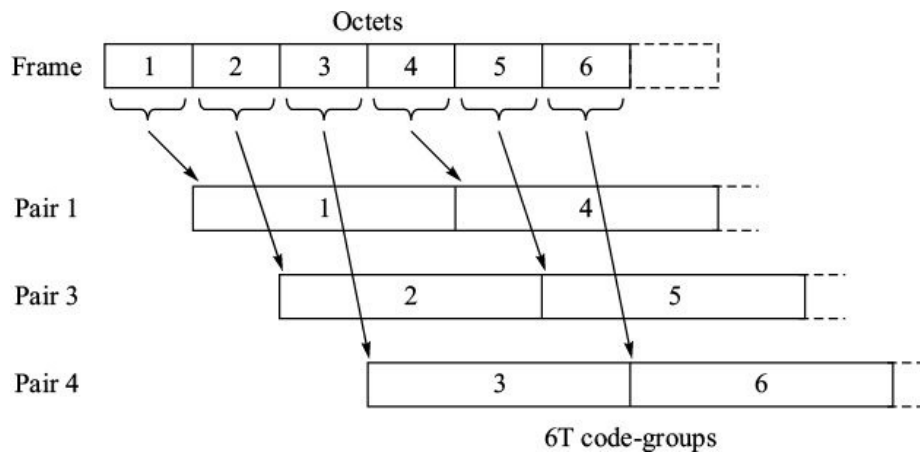


Figure 11.24 Transmission of 6T code groups.

Preamble and start-of-frame delimiter. When a station is ready to send a frame and finds that the Ethernet link is free, it sends Start-of-Stream (SOS) and Start-of-Frame Delimiter (SFD) symbols on the three pairs as shown in Figure 11.25. SOS consists of alternating +1 and -1 levels and enables clock recovery.

SFD is (+1 -1 +1 -1 -1 +1). Preamble of pairs 1 and 3 have two SOSs followed by SFD and preamble of pair 4 has one SOS followed by SFD. The SOSs are staggered in time on the three pairs by time slot equivalent to one octet. The octets of the frame are sent in round robin fashion on the three pairs in the sequence 4, 1, and 3. The first octet of the frame goes on pair 4.

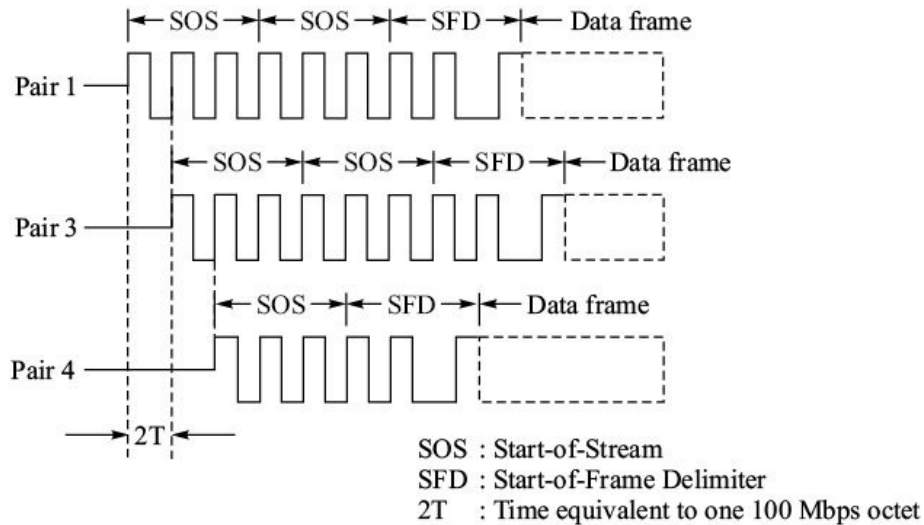


Figure 11.25 Preamble and start-of-frame delimiter.

End-of-stream delimiters. At the end of each frame, *i.e.* after the four FCS octets have been transmitted, End-of-Stream (EOS) symbols are transmitted on each of the three pairs (Figure 11.26). The EOS symbol streams are 2T, 4T or 6T long. The pair that transmits the last octet of the frame, the fourth FCS byte, has 2T long EOS. EOS symbols are readily identified as they are all +1 or -1 symbols. Whether +1 or -1 symbols are to be appended at the end of

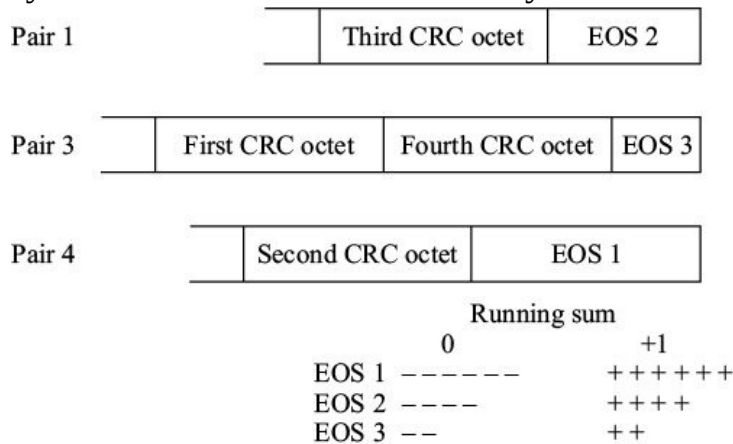


Figure 11.26 EOS signals.

the stream, is decided by the running sum of weights of the code-groups

transmitted on the pair. If the running sum is 1, +1 symbols are appended. If running sum is 0, -1 symbols are appended.

Carrier sense and collision detection. Stations indicate that they are not using the link by sending a special 6T code-group (IDLE) on the simplex pairs. In normal conditions, a station while transmitting a frame, receives IDLE code-group continuously from the other end on the simplex link (Figure 11.21). Receipt of a non-IDLE code-group over this link any time before expiry of collision window indicates that collision has occurred. On detection of collision, the station sends jam sequence and stops further transmission of the frame.

11.8.4 100BaseT2

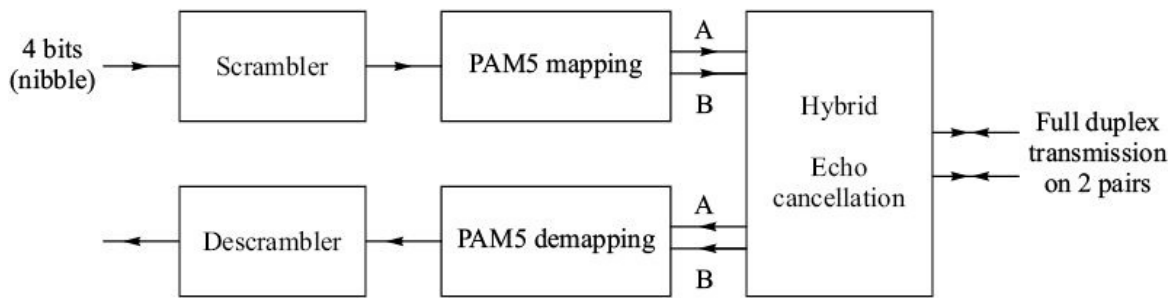
100BaseT4 has two major limitations. It uses four pairs of UTP cable and provides half duplex transmission only. These were overcome in 10BaseT2 specifications which were developed with two goals in mind:

- Provide communications over two pairs of CAT 3 or better cable.
- Support both full duplex and half duplex operations.

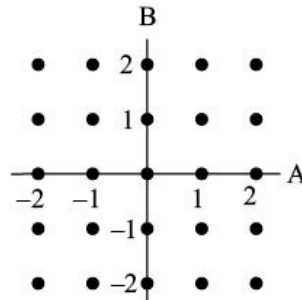
But by the time specifications for 100Base2 were finalized in 1997, the industry had lost interest in 100 Mbps. 100BaseT2 did not even reach the manufacturing stage.

In 100BaseT2, each station sends encoded symbols in both the directions simultaneously on the two pairs of UTP cable (Figure 11.27a). The data frame bits are first scrambled to randomize the bit sequence. Each nibble (group of four bits) is encoded into two five-level (+2, +1, 0, -1, -2) pulse amplitude modulated (PAM5) signals which are sent separately on the two pairs of UTP cable.

Four bits have 2^4 (= 16) combinations. The two PAM5 signals give 25 (= 5^2) combined states of the five levels on the two pairs (Figure 11.27b). Some of the states are used for other signals such as IDLE, frame encapsulation, *etc.* PAM5 encoding on two pairs is referred to as two dimensional PAM5 or PAM55 in some of the texts.



(a) Encoding in 100BaseT2



(b) State space of PAM5x5

Figure 11.27 100BaseT2 signal processing.

Similar signals are received simultaneously from the other end on the same pairs of wires. The polynomial used for scrambling at the other station is different to ensure that the data streams travelling in opposite directions are not correlated. The received signals are separated from the transmitted signal using hybrid and echo canceller. Each transmitted frame is encapsulated. Link synchronization is maintained by sending IDLE signals during interframe gaps.

11.8.5 100BaseX

100BaseX refers to two fast Ethernet specifications 100BaseTX and 100BaseFX. 100BaseTX is for transmission over two pairs of CAT 5 UTP cable or STP cables. 100BaseFX is for transmission over two strands of optical fibres. The encoding procedures are same for both the media. 100BaseX supports point-to-point half and full duplex transmissions.

100BaseX uses 4B/5B encoding wherein each 4-bit nibble is mapped to a 5-bit binary code which is transmitted serially over the link. The 5-bit code set consists of 32 ($= 2^5$) words which are assigned as indicated in Table 11.2.

TABLE 11.2 4B/5B Code Set Used in 100BaseX					
Hex/Name	4-bit nibble	5-bit code	Hex/Name	4-bit nibble	5-bit code
					1011

				1100	
				1101	
				1110	
				1111	
0	0000	11110	B		10111
1	0001	01001	C	—	11010
2	0010	10100	D		11011
3	0011	10101	E		11100
4	0100	01010	F	—	11101
5	0101	01011	I (Idle)		11111
6	0110	01110	J (SSD-1)		11000
7	0111	01111	K (SSD-2)	—	10001
8	1000	10010	T (ESD-1)		01101
9	1001	10011	R (ESD-2)	—	00111
A	1010	10110	H (Error)		00100
				—	
				—	

- 16 codes are used for 16 ($= 2^4$) possible 4-bit nibble combinations.
- Four codes are used as Start-of-Stream Delimiter (SSD) and End-of-Stream Delimiter (ESD). Each MAC frame is encapsulated between these two codes. The first octet of the preamble of MAC frame is replaced with the two SSD codes. Pair of ESD codes are appended to after the FCS field of the MAC frame (Figure 11.28).
- One code is used to represent the IDLE state of the link. This code is continuously transmitted during interframe gap or when the link is idle to maintain the clock synchronization.
- Eleven code groups are invalid codes. Receipt of invalid code results the incoming frame to be treated as an invalid frame.

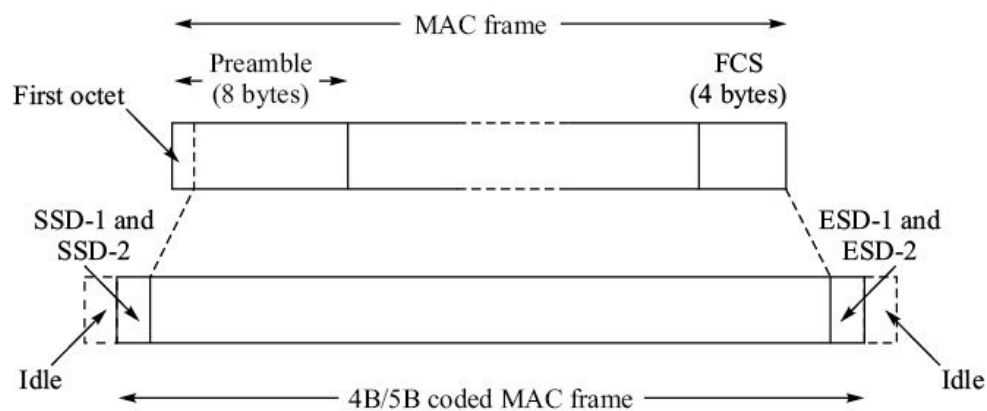


Figure 11.28 Encapsulation of MAC frame in 100BaseX

Figure 11.20 Encapsulation of MAC frame in 100BaseTX.

100BaseTX. 100BaseTX uses two pairs of CAT5 UTP cable or STP cable with RJ-45 connectors. Three level modulation scheme MLT-3 is used (Figure 11.29). In MLT-3, there is level transition for binary 1 and no level transition for binary 0.

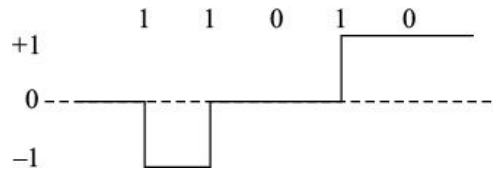


Figure 11.29 MLT-3 signal.

IEEE 802.3u specification of 100BaseTX allows links with a maximum of two repeaters (hubs) and network diameter of approximately 200 metres. Any link segment can be up to 100 metres.

100BaseFX. 100BaseFX uses two strands of optical fibres as transmission medium. NRZI line coding is used for signaling. NRZI encoder generates a transition when binary 1 is being transmitted. No transition occurs when binary 0 is transmitted.

Physical characteristics of 100 Mbps Ethernets are summarized in Table 11.3.

TABLE 11.3 Physical Characteristics of 100 Mbps Ethernets			
	100BaseTX	100BaseFX	100BaseT4
Cable	CAT 5 UTP, Type 1or 2 STP	62.5/125 multimode optical fibre	CAT 3,4 or 5 UTP
Cable pairs/fibre strands	2 pairs	2 strands	4 pairs
Connector	RJ-45 (ISO 8877)	Duplex SC connector	RJ-45 (ISO 8877)
Maximum link segment	100 metres	400 metres	100 metres
Network diameter	200 metres	400 metres	200 metres

11.9 FLOW CONTROL

LANs at high speeds and with full duplex operation require flow control at layer 2. Therefore, an optional flow control mechanism is introduced in the MAC sublayer. It is based on sending a pause command with an indication of time duration for which the transmitting station should stop sending the frames to the station that generated the pause command. The control frame used for this

purpose has the format as shown in Figure 11.30.

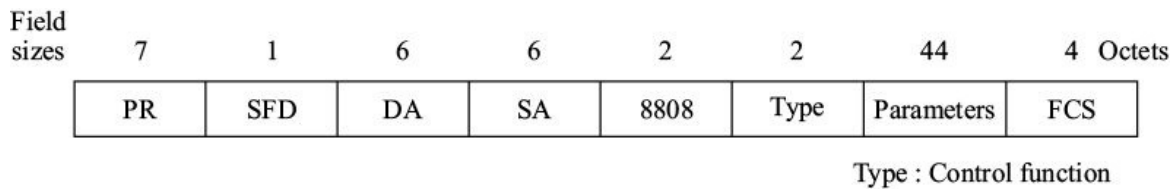


Figure 11.30 Format of IEEE 802.3 control frame.

- The control frame is identified by the length field of the MAC frame. The length field contains 8808 (Hexadecimal) when it is a control frame.
- Type field indicates the type of control function. The only control defined so far is pause function which is coded as 0x0001.
- Parameters field contains the parameters associated with the control function defined in the type field. For the pause function, this field specifies the time duration for which the further transmission of frames must be suspended to the station indicated in the SA field.

Time duration is specified in the multiples of transmission time required for the minimum sized frame (512 bits for 100 Mbps). For example, if the parameter value specified is n , transmission of frames must be suspended for $n \cdot 512 \cdot 10^{-8}$ seconds. For the gigabit Ethernet that operates at 1000 Mbps, the minimum size of the frame is 4096 bits and therefore the pause duration will be $n \cdot 4096 \cdot 10^{-9}$ seconds.

The Destination Address (DA) of the pause frame can be address of one station or broadcast address or multicast address. After expiry of the time specified, these stations can resume transmission of frames to the station indicated in the SA field of the pause command. Transmission of frames can also be resumed if these stations receive another pause command with parameters field set to zero ($n = 0$) from the SA.

11.10 AUTO-NEGOTIATION

100BaseT specification describes a negotiation process that allows the stations of a network to automatically exchange information about their capabilities and then to negotiate and select the most favourable operational mode that they are

capable of supporting. Auto-negotiation is performed totally within the physical layers during link initialization. Auto-negotiation enables two stations to do the following:

- Advertise their Ethernet version and operational capabilities.
- Acknowledge receipt and understanding of agreed operational modes.
- Reject any operational mode that is not agreed to.
- Configure for the highest level operational mode that can be supported.

Auto-negotiation is available only for copper media based fast Ethernets (100BaseTx, 100BaseT4, 100BaseT2), 10BaseT, and 1000BaseT. It is not applicable to 100BaseFX. It is specified as an option for 10BaseT, 100BaseTX, 100BaseT4, but it is required for 100BaseT2 and 1000BaseT implementations.

11.10.1 Transport Mechanism for Auto-Negotiation The auto-negotiation function uses modified form of 10BaseT link integrity pulse sequence. The NLPs are replaced by bursts of Fast Link Pulses (FLP). FLP bursts carry negotiation messages in form of FLP data bits which enable handshaking and negotiating operational modes. Basic features of FLP burst are as follows:

- Each FLP burst is an alternating clock/data sequence of 33 pulse positions.
- The clock pulses are at odd pulse positions. Thus there are 17 clock pulses in a burst. The clock pulses are used for timing and recovery of FLP data bits at even pulse positions.
- The FLP data bits are at even pulse positions. Thus there are sixteen FLP data bit positions in an FLP burst.
- When an FLP data bit is binary 0, no FLP pulse is sent at the corresponding pulse position. If it is binary 1, FLP pulse is sent.
- An FLP burst lasts 2 ms and FLP bursts are generated at interval of 16 ± 8 ms. Each pulse is of 100 ns duration.
- FLP bursts are sent only during the initialization phase of the link. (See Figure 11.31.)

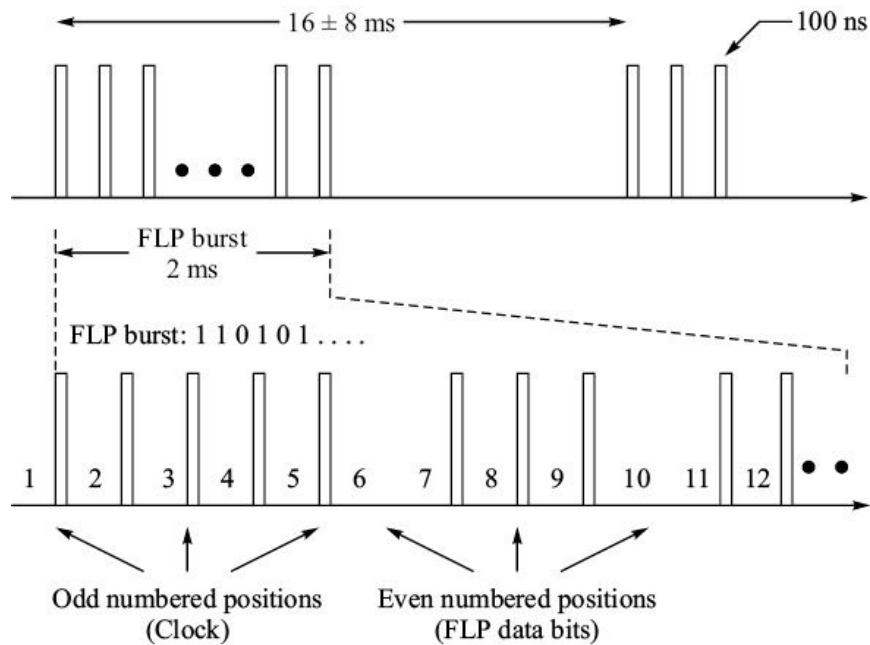


Figure 11.31 FLP bursts.

11.10.2 FLP Burst Encoding

The data bits in an FLP burst contain a 16-bit Link Code Word (LCW). Figure 11.32a shows the format of base LCW. It consists of five fields:

1. Selector field—5 bits
2. Technology Ability Field (TAF)—8 bits
3. Remote Fault (RF)—1 bit
4. Acknowledgement (Ack)—1 bit
5. Next Page (NP)—1 bit.

Selector field. Selector field identifies the technology. There can be 32 different definitions of technologies. At present only two technology codes are defined—10000 for IEEE 802.3 and 01000 for IEEE 802.9.

Technology ability field (TAF). TAF is defined in respect of the selector field. For IEEE 802.3, the eight bits of TAF are encoded as follows to indicate the technology ability of the sending station.

A0	10BaseT		
A1	10BaseT, full duplex	A2	100BaseTX
A3	100BaseTX, full duplex	A4	100BaseT4
A5	Pause operation for full duplex	A6	Reserved
		A7	Reserved

Remote fault (RF). RF bit allows transmission of simple fault indication to the other end. For example, if receive side of the link is not working, a station will set the RF bit to 1 in the base LCW. This will indicate to the other end that remote station has a fault status.

Acknowledgement (Ack). Acknowledgement bit is used to confirm receipt of base LCW to the sending station.

Next Page (NP). NP bit is used to indicate that there is an additional page other than base LCW.

Next pages contain additional information relating to ability, device type, and vendor. Next page function is feasible if it is supported by both the ends. The format of next page is shown in Figure 11.32b. Next page can be message-next-page or unformatted-next-page. Message Page (MP) bit identifies the type of next page. If MP bit is binary 1, it is message-next-page and the message field (11 bits) contains predefined messages as per IEEE 802.3. Unformatted-next-page is identified by MP bit equal to binary 0. The unformatted messages are not predefined.

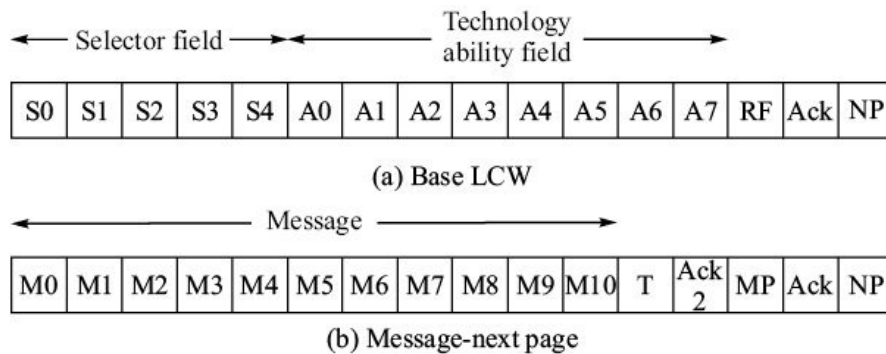


Figure 11.32 Formats of base LCW and next page.

Acknowledgement bit 2 (Ack2) is set by the receiving device to indicate that the device supports the function indicated in the message field. Toggle (T) bit is used for synchronizing the exchange of next pages. Toggle bit is always set to the inverted value of the 11th bit of the last message received. Ack and NP fields work in the same manner as for the base LCW.

11.10.3 Ability Negotiation Mechanism

To understand the basic ability negotiation mechanism, let us consider two auto-negotiation capable devices A and B ready (powered on) to negotiate the operational mode. Negotiation steps from station A's perspective are listed

below. Only base LCWs are used in this example.

- A sends base LCWs with Ack = 0.
- A receives three identical base LCWs with Ack = 0 from B.
- A transmits identical LCWs with Ack = 1 six to eight times to indicate to B that its LCW has been correctly received.
- A receives three identical LCWs with Ack = 1 from B. A knows that its LCWs have been correctly received by B.
- A configures the highest performance configuration.

Both the stations, A and B, compare their ability after receipt of LCWs with Ack = 1, and configure the highest performance operational mode as determined by priority resolution (Table 11.4).

Priority level	Operational mode	Maximum data transfer rate*
1	1000 BaseT, full duplex	2000 Mbps
2	1000 BaseT, half duplex	1000 Mbps
3	100 BaseT2, full duplex	200 Mbps
4	100 BaseTX, full duplex	200 Mbps
5	100 BaseT2, half duplex	100 Mbps
6	100 BaseT4, half duplex	100 Mbps
7	100 BaseTX, half duplex	100 Mbps
8	10 BaseT, full duplex	20 Mbps
9	10 BaseT, half duplex	10 Mbps

* Total data transfer rate for full duplex operation aggregate of data transfer rates of both the directions.

11.10.4 Parallel Detection

If one of the two end devices does not support auto-negotiation, the link is configured based on the signals generated by the device. The end station with auto-negotiation capability has parallel detection capability. It first looks for FLPs on the receive side and if they are absent, it identifies the type of signal being received from the other end. For example, if the received signals are NLPs, it configures itself to 10BaseT.

11.11 GIGABIT ETHERNET

The gigabit Ethernet standards development resulted in two primary specifications—1000BaseT and 1000BaseX (Figure 11.33). IEEE specifications

for 1000BaseT are IEEE 802.3ab. 1000 BaseX comprises three sets of physical interfaces, 1000BaseCX, 1000BaseSX and 1000BaseLX. IEEE specifications for 1000BaseX are IEEE 802.3z.

1000BaseCX uses two shielded twisted copper pairs as transmission medium. 1000BaseSX and 1000BaseLX are based on multimode/monomode optical fibre cables. S stands for short wavelength (850 nm) and L stands for long wavelength (1300 nm) of optical signals used in these interfaces.

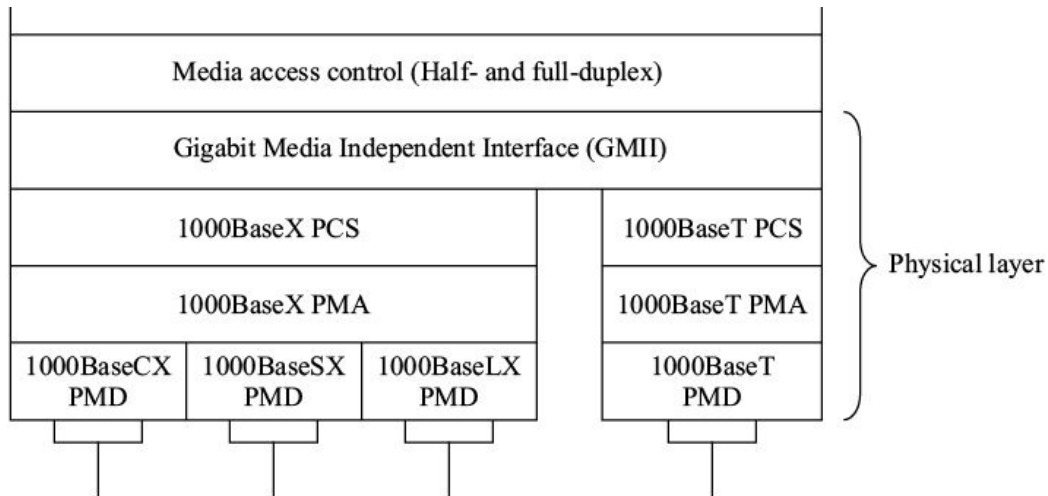


Figure 11.33 Gigabit Ethernet layered architecture.

MAC sublayer remains same as for the other lower rate ethernets. At the physical layer, two new concepts are introduced:

- Gigabit carrier extension
- Frame bursting.

Before going into gigabit Ethernet interfaces (1000BaseT, 1000BaseCX, 1000BaseSX, and 1000BaseLX), let us understand these concepts.

11.11.1 Gigabit Carrier Extension

At 1000 Mbps data rate, frame transmission time becomes one-tenth of that at 100 Mbps. This in turn implies that the maximum diameter of the gigabit Ethernets should be reduced by a factor of ten, to about 20 metres. Gigabit Ethernet with 20 metres of maximum diameter is not of any practical use. Alternative is to increase minimum frame size. But that would make gigabit Ethernet backward incompatible, *i.e.* incompatible to lower speed Ethernets. Therefore, an apparent increase in frame size is made at the physical layer level.

A variable length non-data field is appended by the physical layer to the frames that are shorter than the specified minimum size, 520 bytes for 1000BaseT ethernet and 416 bytes for 1000BaseX² Ethernet. (Figure 11.34).

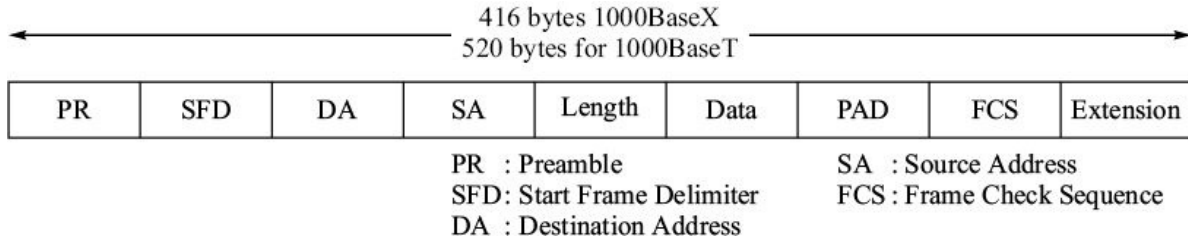


Figure 11.34 MAC frame with gigabit carrier extension.

Because the frame extension is carried out at the physical layer, we call this extension as ‘carrier extension’. The extension field is removed by the physical layer during frame reception. The extension bytes need to be special bytes that do not occur in the MAC frame so that they are readily identified and stripped without having to analyze the entire frame. Therefore, a special control code group is assigned for extension field.

11.11.2 Frame Bursting

Frame burst mode allows a station to send a short sequence (burst) of frames continuously without having to interrupt the transmission after every frame. The medium appears occupied continuously to the other stations during the frame burst. The maximum burst size is limited to about 5.4 times the maximum Ethernet frame size (Figure 11.35).

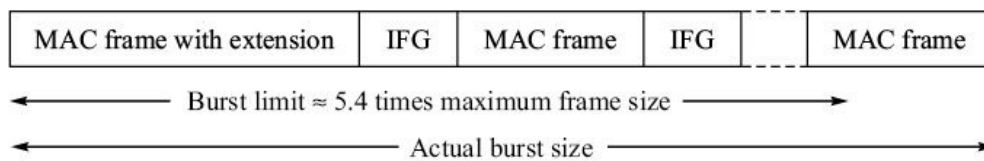


Figure 11.35 Frame bursting in gigabit Ethernet.

The first frame of the burst has at least the prescribed minimum size of the frame, with extension bytes if required. Subsequent frames in the burst may have any size, from 64 bytes to 1518 bytes. In other words, the subsequent frames do not require carrier extension. There can be any number of frames in a burst so long as the maximum burst size is not exceeded. If the burst limit is reached after a frame transmission has begun, the transmission is allowed to continue until the entire frame has been sent. The frames are separated by InterFrame Gaps (IFG) of 12 bytes as before and IFG is filled with the extension bytes.

11.11.3 1000BaseT

1000BaseT is based on the proven design approaches of fast Ethernet. Each frame is encapsulated in start-of-stream and end-of-stream delimiters. IDLE symbols are sent during interframe gaps. The bit stream is scrambled before it is encoded. Separate scramblers at each end generate essentially uncorrelated data streams that travel in opposite directions on the four pairs (Figure 11.36).

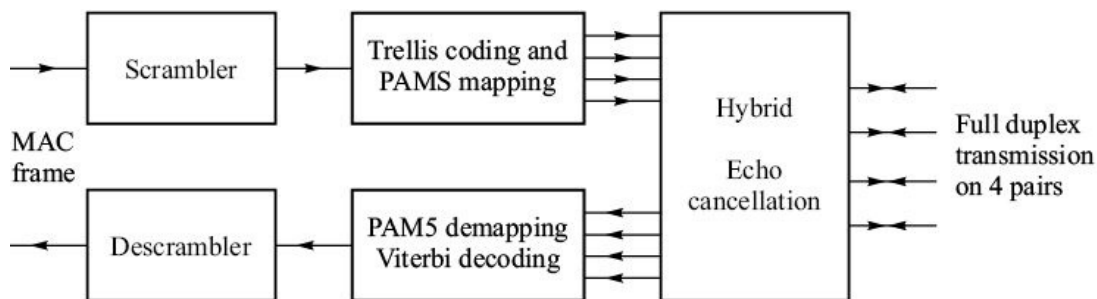


Figure 11.36 1000BaseT Ethernet signal processing.

8-state trellis forward error correction code is used at the physical layer level to correct the errors that may occur during transmission of signals. Trellis coding generates one extra bit for every 8 bits (Figure 11.37). Viterbi decoder is used at the receive side.

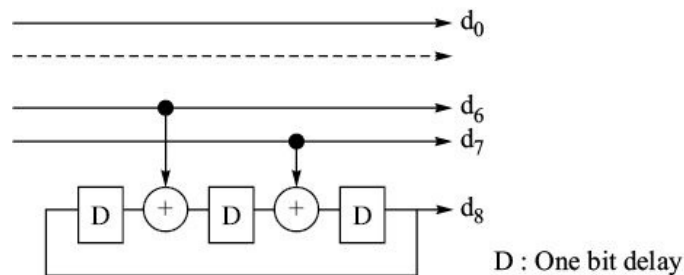


Figure 11.37 Trellis encoder.

Four-dimensional PAM5 is used for signal encoding which has 625 ($= 5^4$) possible states. These states are used to represent 512 ($= 2^9$) combinations of 9 trellis encoded bits. Out of the rest PAM5 states, some are used for control signals and others are invalid codes.

Hybrids are used for combining the outgoing and incoming signals on each pair. Echo cancellation and digital signal processing techniques are used at each end for pulse shaping and removing the transmit signal leakage of the hybrids.

1000BaseT provides half-and full-duplex transmission on four-pair CAT 5 or better UTP copper cable. The maximum length of CAT 5 segment length is 100 metres. In half duplex mode only one repeater is allowed. 1000BaseT supports

auto-negotiation using FLP bursts. It is compatible with 10BaseT and 100BaseT Ethernets.

11.11.4 1000BaseX

1000BaseX Ethernet is based on using two STP copper pairs or two strands of multimode/ monomode optical fibres. It uses 8B/10B block encoding for mapping each 8-bit word into 10-bit code group. Thus the bit rate after encoding is 1250 Mbps. 10-bit code-groups have 1024 ($= 2^{10}$) combinations. These are used as follows:

- There are 256 data words and 12 control words to be coded. Examples of control words are
 - Configuration (C) for auto-negotiation
 - Start of stream (S) delimiter
 - End of stream (T) delimiter.
- Each 8-bit word is assigned a pair of 10-bit code groups. The pair is so selected that it has ten binary 1s and ten binary 0s in all. The pair entities are alternatively sent on each occurrence of respective 8-bit word. This ensures that number of zeroes and ones are transmitted in equal numbers maintaining DC balance.
- The unassigned code-groups are invalid and are enable error detection.

1000BaseX is implemented as 1000BaseCX, 1000BaseSX or 1000BaseLX. 1000BaseCX uses 150 ohms shielded two pair copper cable with DB-9 connector. Maximum link segment length is 25 metres. 1000BaseCX is intended for the patch chord use. 1000BaseSX uses 850 nanometre optical wavelength on multimode optical fibres. The maximum link length can be 275 to 550 metres depending on modal bandwidth of the optical fibre cable. 1000BaseLX is based on 1300 nanometre optical wavelength on multimode or monomode optical fibres. The maximum link length is 550 metres on multimode fibres and 5000 metres on monomode fibres. Table 11.5 summarizes the physical characteristics of gigabit Ethernets.

Characteristics	1000BaseT	1000BaseCX	1000BaseSX	1000BaseLX	
Cable	CAT 5 UTP or better	150 ohms STP copper cable	62.5/125, 50/125 mm multimode optical fibre	62.5/125, 50/125 mm multimode optical fibre	10/125 monomode optical fibre

Wavelength	-	-	850 nm	1300 nm	1300 nm
Cable pairs/ fibre strands	4 pairs	2 pairs	2 strands	2 strands	2 strands
Connector	DB-9	DB-9	SFF MT-RJ or duplex SC connector	SFF MT-RJ or duplex SC connector	SFF MT-RJ or duplex SC connector
Maximum link segment	100 m	25 m	275 to 550 m	550 m	5 km
Operation mode*	HD/FD	HD/FD	HD/FD	HD/FD	HD/FD

* Half duplex (HD), Full duplex (FD).

11.11.5 Auto-Negotiation in Gigabit Ethernet

Both 1000BaseT and 1000BaseX support auto-negotiation function to determine the mode of transmission (half or full duplex) and flow control mode. 1000BaseT also supports speed negotiation and is backward compatible with 10BaseT and 100BaseT Ethernet. FLP bursts are used in 1000BaseT Ethernet.

1000BaseX does not negotiate speed. For the negotiation purpose, 8B/10B code groups are used instead of FLP bursts. Format of the base page LCW is shown in Figure 11.38. Format of the next page is same as shown in Figure 11.32b.

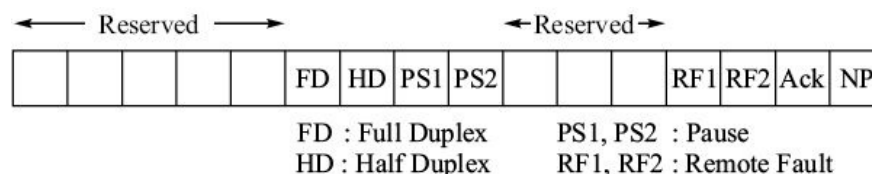


Figure 11.38 Base LCW format in gigabit Ethernet.

SUMMARY

Ethernet LANs are based on CSMA/CD contention access method in which the stations contend for the use of the medium. It is quite likely that more than one station will transmit simultaneously and the data frames will collide. Therefore, a station checks if the transmission medium is free and if the medium is free, it transmits its frames. If collision is detected during transmission, it aborts the frame and then tries again. Retries are made using a back off mechanism called truncated binary exponential back off to avoid repeat collisions.

Ethernet was originally conceived as LAN technology based on bus topology.

But today it has become a standard interface for interconnecting data networking devices and therefore we have point-to-point and star topologies. Star topology uses hubs (multiport repeaters) or layer 2 switches. Ethernet LANs are no longer connected in bus configuration.

There is choice of Ethernet variations at the physical layer level. The choice is in terms of transmission medium and bit rate. Bus topology Ethernet LANs were based on coaxial cable. Point-to-point and star topologies use twisted pair copper cables (CAT3, CAT5, STP) and optical fibres (multimode, monomode). The bits rates can be 10, 100 and 1000 Mbps.

EXERCISES

1. Prove that the maximum throughput of pure ALOHA occurs at $G = 0.5$ and the maximum throughput is 18.4%.
2. Prove that the maximum utilization of slotted ALOHA occurs at $G = 1$ and the maximum throughput is 36.8%.
3. Why the Ethernet MAC frame has length field?
4. Calculate the minimum size of the 10 Mbps Ethernet frame required if the round trip delay of the bus is 480 ms.
5. If the bit rate is increased to 100 Mbps in Exercise 4, what will be the minimum size of the frame? What are the drawbacks of having large frame size?
6. Suppose A and B be two stations attempting to transmit their frames on Ethernet. Suppose their first attempt results in collision. For their second attempt, they pick randomly between numbers 0 and 1. Assume that A wins by getting 0 and transmits its frame. B gets 1 and therefore backs off for one unit of time. After A completes transmission of its first frame, the next frame of A and the first frame of B will collide again. This time B is to pick randomly a number from 0, 1, and 2 and A need to pick randomly a number from 0 and 1.
 - (a) Calculate the probability that A wins the back off race second time also.
 - (b) Calculate the probability that B wins the back off race this time.
7. In Exercise 6, suppose a station that has transmitted its frame successfully is given a minimum back off of 2 units of time. After A has successfully transmitted its frame, while B was waiting, which will be the next likely station to transmit?

8. In Exercise 7, suppose there is third station C that wants to transmit. Will C get an opportunity to transmit if A and B have number of frames to transmit? If not, why not?
9. Octets of an Ethernet frame in hexadecimal are given below. The preamble and start delimiter octets are not included. Identify the various fields. Is it an IEEE 802.3 frame or Ethernet (DIX) frame?
10. 00 00 66 33 B5 49 00 00 A7 12 36 B7 00 60 AA AA 03 00 00 00 08 00 48 45 4C 4C
11. Encode the following message using 4B/5B code and then draw the resulting NRZI signal.
12. 1101 1110 1010 1101
13. If 10BaseT network is upgraded to 100 Mbps, which 100Base version(s) can be used and why?
14. Suppose Ethernet addresses are chosen at random. What is the probability that all the addresses will be different on a LAN having 16 stations?

1 For example, (+1 -1 0 +1 +1 0) has weight of +2.

2 The minimum frame size of 1000BaseX is less because 8B/10B encoding is used in 1000BaseX.

12

Token Passing Local Area Networks

In the last chapter we examined the operation of contention access based local area networks. In this chapter we study the token passing local area networks (LANs). In these LANs, the access to media is controlled by circulating a token among the stations connected to a LAN. A station that holds token is authorized to use the media for transmitting its frame. Token passing LANs can have ring or bus topologies. In this chapter we discuss the following token passing LANs:

- IEEE 802.5 token ring LAN
- IEEE 802.4 token bus LAN
- Fibre Distributed Data Interface (FDDI) LAN.

We discuss their basic operation, MAC sublayer protocol, token management and priority functions. Physical layer of each of these LANs is described with respects to number of stations, maximum size of the LAN, line codes, and other physical aspects.

12.1 TOKEN RING LOCAL AREA NETWORK

A token passing ring consists of number of stations interconnected in the form of a ring through point-to-point links (Figure 12.1). Each station acts as a repeater and regenerates the signals it receives on one link and sends them onto the next link after a delay of at least one bit. The ring is unidirectional.

All the stations share the interconnecting media for exchanging the data frames. Suppose station A in Figure 12.1 wants to send a frame to station D. It

breaks the ring and inserts its frame with the destination (D) and source (A) addresses. The frame passes through the stations B and C which act as repeaters and forward the frame to the next link. They do not copy the

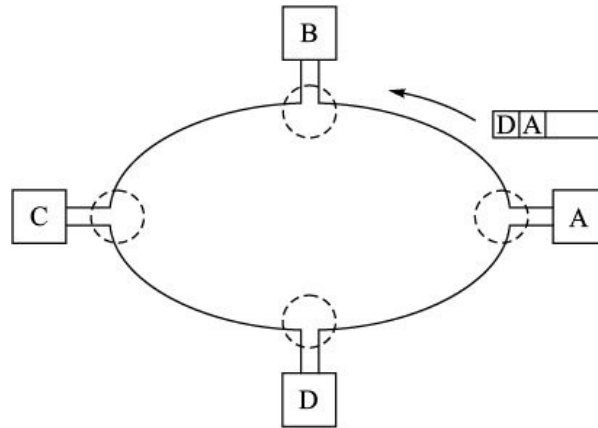


Figure 12.1 Token ring LAN.

frame as the frame is not addressed to them. Station D finds its address on the frame and copies it. The frame continues its journey anti-clockwise and returns back to station A. The frame does not circulate again around the ring as the ring is broken at station A. Station A removes the frame from the ring.

Thus an active station on the ring is always in one of the three modes—repeater mode, insert mode, or copy mode (Figure 12.2).

- In the repeater mode, the received signals are regenerated and transmitted on the outgoing link. There is at least one bit delay in the shift register.
- In the insert mode, the ring is broken. The station sends its own frame on the outgoing link. The incoming signals are received but are not sent on the ring again.
- In the copy mode, the station regenerates the received signals and sends them on the outgoing link. It also copies the received signals for its use.

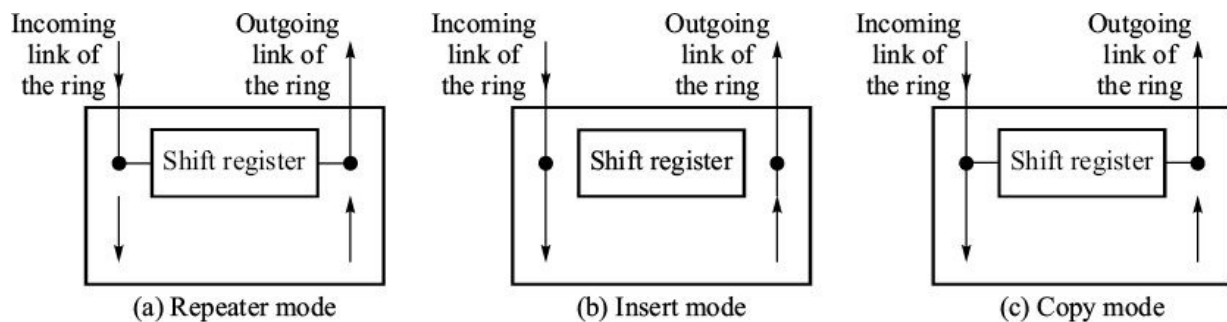


Figure 12.2 Interconnection modes of a token ring station.

12.2 MEDIA ACCESS CONTROL IN TOKEN RING LAN

Access to the ring for transmitting a frame is controlled by use of a token. The token is passed from station to station around the ring. The physical locations of the stations on the ring determine the sequence of passing the token. When a station has frames to transmit, it seizes the token. It holds the token and sends its one or more frames and then releases the token for the next station on the ring. Figure 12.3 shows in detail the sequence of events when station A sends a frame to station C and releases the token which is then picked up by station D.

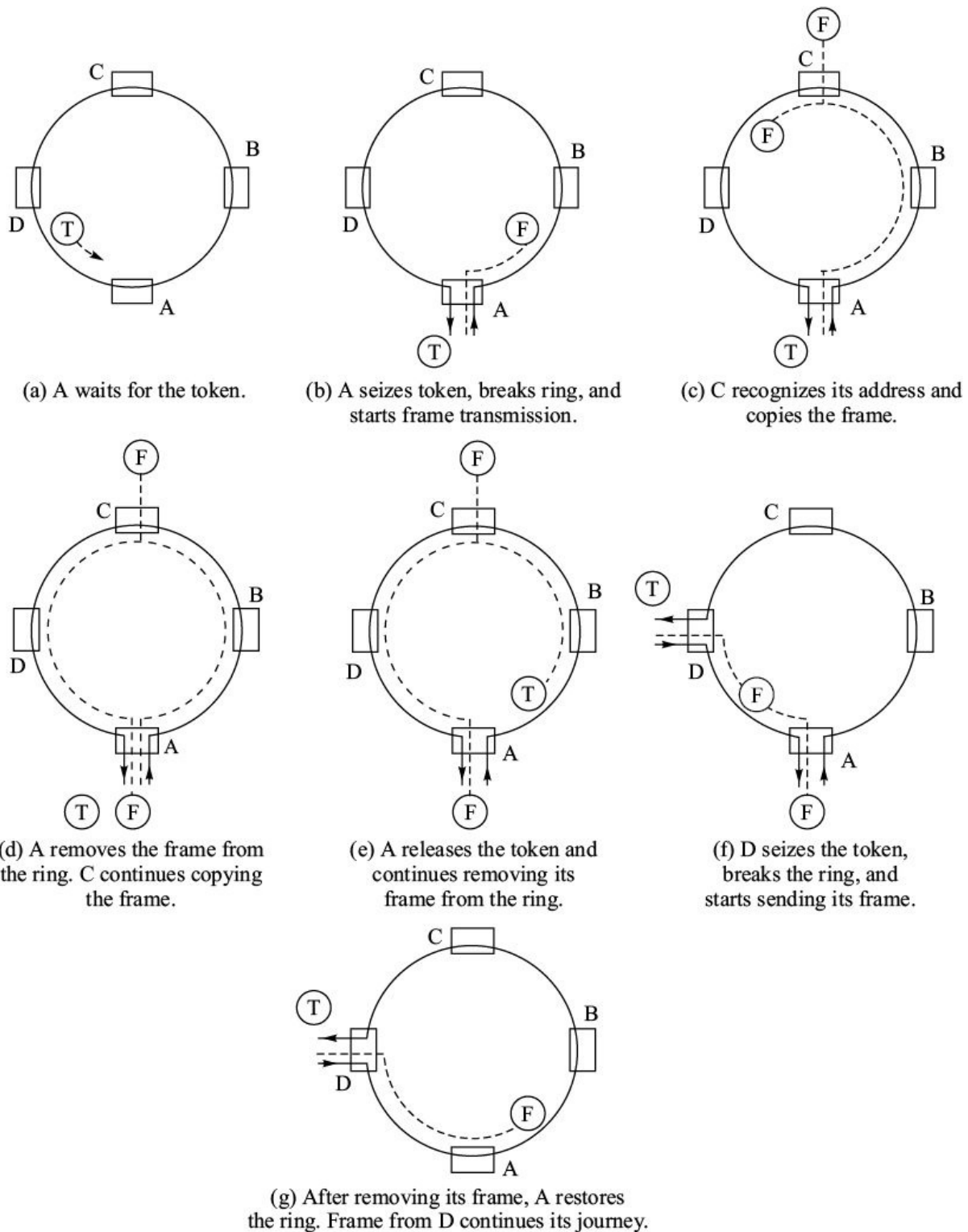


Figure 12.3 Media access control using token.

12.2.1 Token Holding Time

To give fair opportunity to every station, a station can hold the token for a

maximum predefined period called Token Holding Time (THT). Its typical value is 10 ms, which limits the maximum frame size to about 4500 octets at 4 Mbps. A station can always send multiple smaller frames during THT.

12.2.2 Early Token Release

There are two options for release of token. A station can release the token immediately after transmitting the last octet of its data frame. Alternatively, it holds the token till it removes the entire frame from the ring, *i.e.* till the last octet of the frame returns to it. The first option is called early release option and is adopted in 16 Mbps token ring.

12.3 RING SIZE

When none of the stations on the ring has any frame to send, the token circulates in the ring for any station to pick up. In such situation the leading bits of the token frame may corrupt the trailing bits of the frame if the size of the ring is not long enough. Let us calculate the minimum size (circumference) of the ring for a 4 Mbps token ring LAN. The token is 3 octets long as we shall see later.

Time for transmitting a bit at 4 Mbps : 0.25 ms

Time required for transmitting 24 bits of the token frame : 6 ms

Cable length travelled by the leading bit in 6 ms : $198 \times 6 = 1188$ m

(Propagation speed = 1.98×10^8 m/s)

If the ring size is less than 1188 m, the leading bits of the token will hit the trailing bits and corrupt the token frame. For 16 Mbps minimum token ring size comes to about 300 metres. This limitation is overcome by providing a constant 24-bit shift register in the ring. This shift register is introduced in the ring by one of the stations designated as active monitor station. Active monitor station carries out several other functions also. We will examine functions of the active monitor station later.

12.3.1 Bypass Relay

Each station acts as a repeater. If a station is down due to any reason, the entire network will also collapse. Therefore, a passive bypass relay, connected across the station, is provided (Figure 12.4).

The relay is powered by the station and it bypasses the station in its neutral

position. The ring continuity is maintained through the relay contacts when the station is bypassed with no repeater function.

When the station is activated, the relay contacts changeover bringing the station into the network. The relay box is often termed as Ring Interface Unit (RIU) or Trunk Coupling Unit (TCU). It connects the station to the ring. As shown in Figure 12.1, all stations are connected through RIUs in the ring.

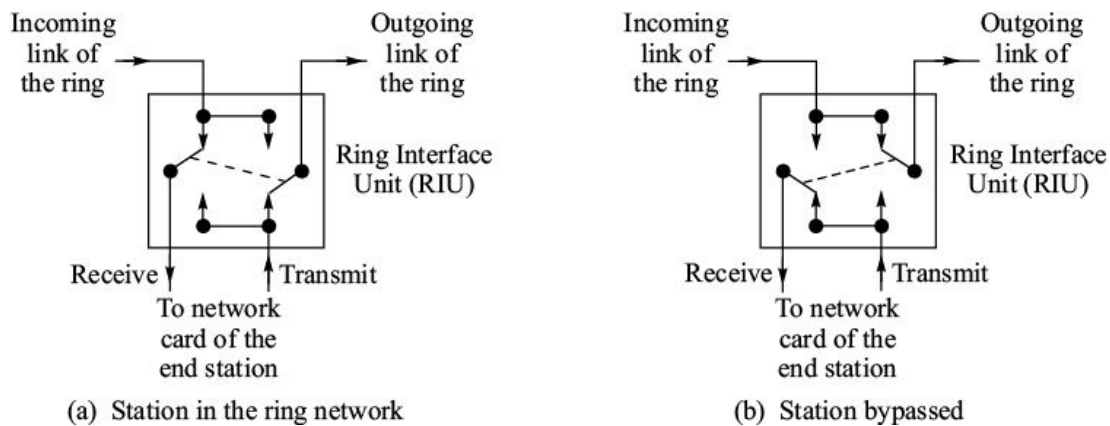


Figure 12.4 Ring interface unit.

12.3.2 Multi-Station Attachment Unit

RIUs impose limitation on ring size. Consider a situation when stations B, C, and D are bypassed (Figure 12.1). The relays in these stations bypass the regeneration function also. Therefore, the entire ring becomes one physical segment. The ring size therefore gets limited to the maximum permissible length of one segment, which is determined by the transmit signal level, the cable characteristics and the receiver sensitivity. Multi-Station Attachment Unit (MSAU) overcomes this limitation.

MSAU packs several RIUs in one box. The stations are connected to an MSAU using two pairs of cables. Since the bypass relays are provided in an MSAU, there is no increase in physical length of a link when a station is bypassed.

Depending on the number of stations to be connected to the ring, several MSAUs can be interconnected among themselves to complete the ring (Figure 12.5). MSAUs provide flexibility in cable layout design and make it possible to use structured cabling for the network.

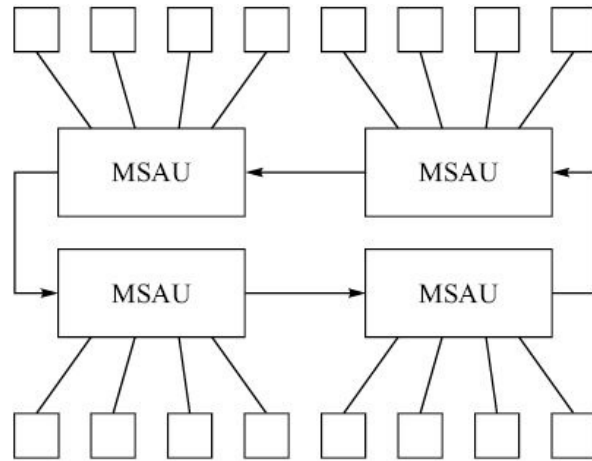


Figure 12.5 Multi-station attachment unit.

12.4 STANDARDS FOR TOKEN RING LAN

Token ring network was originally developed by IBM in 1970s. IEEE adopted the IBM design and framed IEEE 802.5 specifications for token ring local area networks. IEEE specifications are almost identical to IBM's token ring specifications, only having some minor differences.

The IEEE 802.5 MAC sublayer works with the LLC sublayer. At the physical layer, the specified bit rates are 4 and 16 Mbps. The line code used is differential Manchester. IEEE 802.5 does not specify any size of link segments or number of stations on a ring, but minimal required quality of receive signal is specified. Signal quality can be achieved by proper combination of transmit signal levels, receiver sensitivity, and medium quality. Usually STP cable is used for token rings. Typical values maximum segment length between two stations and maximum number of stations in a ring are as follows:

4 Mbps (STP cable) : Segment length 385 m, number of stations 260

16 Mbps (STP cable) : Segment length 173 m, number of stations 136

12.4.1 IEEE 802.5 MAC Frame Format

IEEE 802.5 specifies two basic MAC frame format types (Figure 12.6):

- Token frame
- Data/Control frame.

Token frame is three byte long and consists of start delimiter byte, access

control byte, and end delimiter byte. Data frames vary in size depending on the size of data field. Control frames do not have data field.

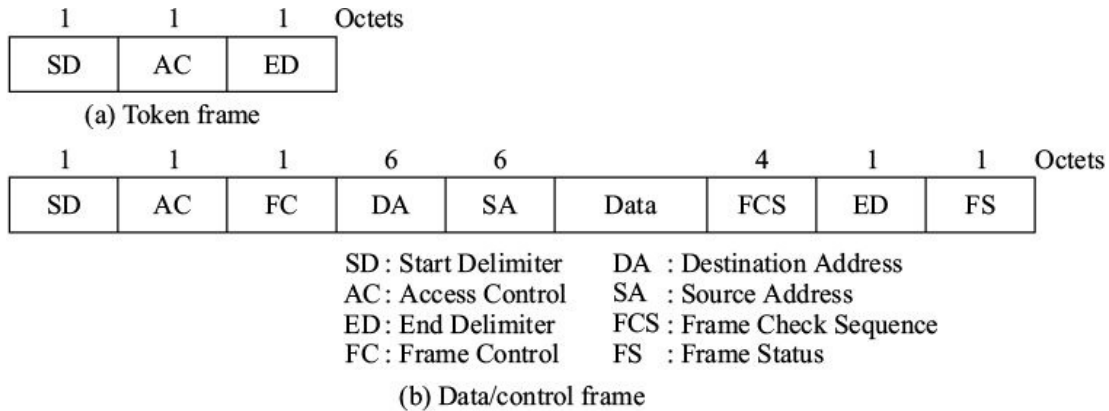


Figure 12.6 Formats of IEEE 802.5 frames.

Note that there is no preamble in these frames. Preamble is not required because a station receives regenerated signals from the neighbour and its internal clock is locked to a master clock of the ring. It is not so in Ethernet or token bus LANs where signals travel end to end without regeneration along the bus and therefore get distorted.

Start delimiter (SD). It is one octet long unique symbol pattern that marks the start of the frame (Figure 12.7a). J and K are special symbols that enable identification of the start delimiter. These symbols do not have usual signal change in the middle of bit and thus violate the differential Manchester code. Code violation property of J and K bits is used for their detection.

Access control (AC). It is one octet long field containing priority bits (P), token bit (T), monitoring bit (M), and reservation bits (R) as shown in Figure 12.7b. Priority bits (P) indicate the current priority level of the data or token frame. Reservation bits (R) are used for reserving the next priority. We will examine priority reservation mechanism shortly.

The token bit (T) distinguishes a token frame from data/control frame. It is 0 in the token frame and 1 in the data/control frame. A station in repeater mode and wanting to send a frame, waits for the token bit. If it finds that token bit is 0, it seizes the token by changing over to insert mode and breaking the ring. Minimum one bit delay ensures that the token bit is not passed to the output port before the ring is broken. The station then inserts 1 in place of 0 at the token bit position and then continues with its transmission of the rest of its data/control frame. Thus the token frame gets converted into data/control frame. Note that the

priority bits (P) remain unchanged in this process.

Frame control (FC). It is one octet long field and distinguishes data and control frames (Figure 12.7c). If FF bits are 01, then it is a data frame that contains LLC frame in the data field. If FF bits are 00, then it is a control frame. The six Z bits indicate the control function. Some of the control functions are listed below.

<i>Z-bits</i>	<i>Control function</i>
000011	Claim Token (CT)
000010	Beacon
000100	Purge
000101	Active Monitor Present (AMP)
000000	Standby Monitor Present (SMP)

Destination address. The destination address field is 6 octets long. The address structure is same as in IEEE 802.3 Ethernet except some special features applicable to token-ring architecture. These are described later.

Source address. The source address field is also 6 octets long address structure same as in IEEE 802.3 Ethernet except some special features applicable to token-ring architecture. The destination and source address fields can be 2 or six octets long as per the standard but 2-octet addressing is seldom used.

Data field. It can have 0 or more octets. There is no maximum size, but a station can hold the token for a limited period. The maximum size of the data frame (and therefore the maximum size of data field) is determined by bit rate and the Token Holding Time (THT). Typical maximum length of data field is 4500 octets for 4 Mbps LAN and 18000 octets for 16 Mbps LAN.

Frame check sequence. The frame check sequence is 4 octets long and contains CRC code. It checks on DA, SA, FC, and data fields.

End delimiter. It is one octet long and contains a unique symbol pattern as below that marks the end of a token or data frame (Figure 12.7d):

- J and K are special symbols that violate the differential Manchester code and identify the end delimiter.
- E-bit is error bit. When a frame passes by a station, the station carries out FCS check on fly and if an error is detected, it sets E-bit to 1.
- I-bit is set to 1 by the sending station if there are more frames to follow. It

is set to 0 in the last frame.

Frame status. This field is one octet long (Figure 12.7e). It contains two address recognized bits (A-bits) and two frame copied bits (C-bits). Every frame is sent with AC = 00. When a frame passes by a station having address same as in the DA field, the station sets A-bit to 1 indicating to the frame originating station that the destination station is alive on the ring. If the destination station is also able to copy the frame, it sets C-bit also to 1. Thus possible combinations of AC bits are as follows:

AC = 00, Addressed station is not on the ring.

AC = 11, Frame has been copied by the addressed station.

AC = 10, Addressed station is on the ring but the frame is not copied.

AC = 01, Invalid value of the AC field.

If a station recognizes DA as his own address and finds that A-bit has already been set to 1 by another station, it implies that there are duplicate addresses on the ring.

Note that FS field cannot be brought in check-span of FCS bytes because the bits of FS field are changed by the frame-receiving station. To check occurrence of errors in the FC field, the AC bits are sent in duplicate (Figure 12.7e). By duplicating the AC bits in the field, errors can be detected if the two AC fields are different.

(a) Start Delimiter (SD) field	J	K	0	J	K	0	0	0
(b) Access Control (AC) field	P	P	P	T	M	R	R	R
(c) Frame Control (FC) field	F	F	Z	Z	Z	Z	Z	Z
(d) End Delimiter (ED) field	J	K	1	J	K	1	I	E
(e) Frame Status (FS) field	A	C	X	X	A	C	X	X

Figure 12.7 Formats of important fields of IEEE 802.5.

12.5 MAC ADDRESSES (DA/SA) IN TOKEN RING LAN

Figure 12.8 shows the formats of 6-octet source and destination address fields.

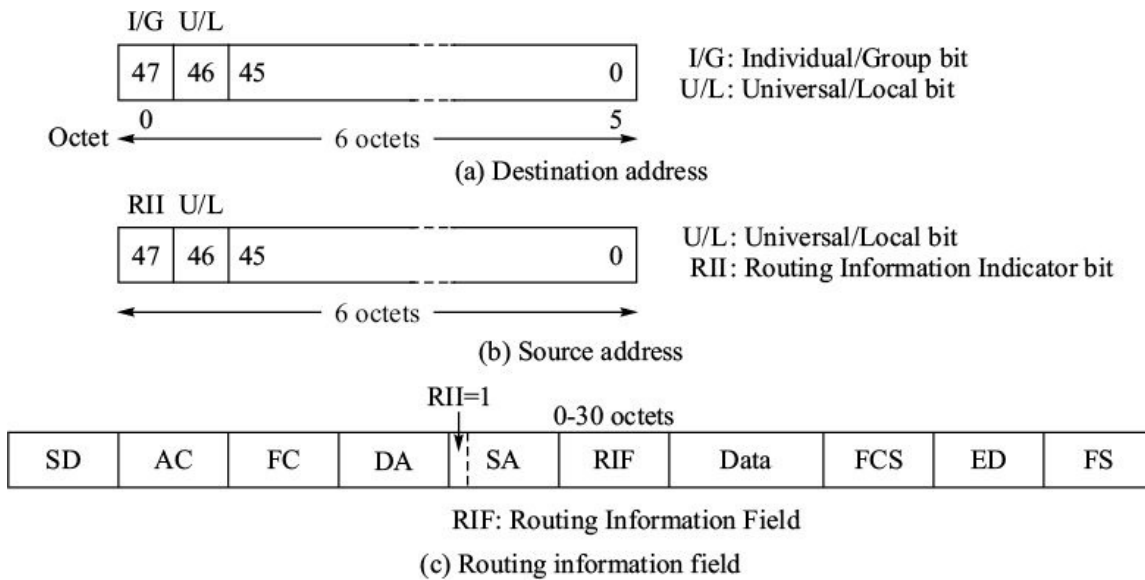


Figure 12.8 The address fields of IEEE 803.5.

- I/G bit of destination address indicates whether the address is individual (I/G = 0) or group address (I/G = 1).
- U/L bit of destination and source addresses indicates whether the address is locally administered (U/L = 1) or globally administered (U/L = 0).
- All 1s have broadcast address.
- All 0s have null address. Null address is used when a station wants to send a frame to itself. It is not recognized by any other station.

Routing information indicator bit (RII). The first bit of source address field is Routing Information Indicator (RII) bit. It is used when the ring has a source routing bridge¹ connected in the ring. If RII = 0, the frame does not contain the routing information.

If RII = 1, it implies that the destination address is not on the ring and it is accessible through the bridge. The routing information required by the bridge is contained in the optional field Routing Information Field (RIF) just after the SA field (Figure 12.8c). The source routing information field is 0 to 30 octets long. The first octet of RIF contains 5-bit length field that indicates the size of RIF field. Using the length information, RIF and data fields can be delineated. We will not go further into the structure of RIF field at this stage.

12.5.1 Functional Address

Some of the stations are assigned special functional tasks. Active monitor is one

such task. These tasks can be assigned to any station. If a station needs to send information regarding a special function to the station that has been assigned that function, it does not need to know the identity of that station. It can send the information on a data frame bearing a functional address in the DA field. The station that has been assigned to carry out the task, will pick up all the frames bearing the functional address. For example, functional address of the active monitor station is C000-0000-0001. A frame transmitted with this functional address in the destination address field will be picked up by the active monitor station, whoever it may be.

Figure 12.9 shows structure of DA field that contains functional address. The first 17 bits of DA are the functional address indicator bits that indicate that this DA is functional address. The rest 31 bits represent the functional address. Some examples of functional addresses are given below:

Active monitor	C000-0000-0001 (Hex)
Ring parameter server	C000-0000-0002 (Hex)
Reserved	C000-0000-0004 (Hex)
Ring error monitor	C000-0000-0008 (Hex)
Configuration report server	C000-0000-0010 (Hex)

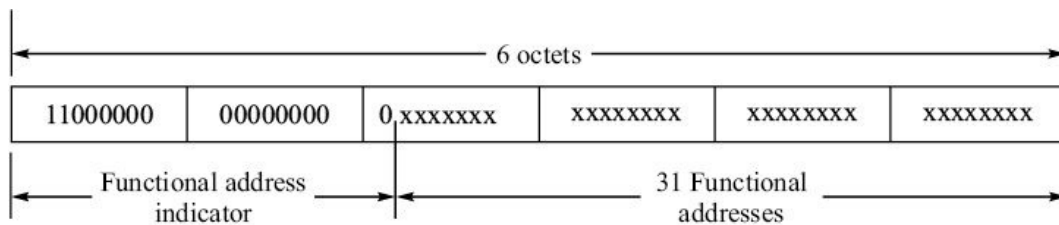


Figure 12.9 Functional addresses.

There are 31 functional addresses. Each address has single binary 1 in one of the 31 bit positions and the rest 30 bits are zeroes. This lends an interesting property to the functional addressing scheme. Several functional addresses can be combined. For example,

Active monitor	C000-0000-0001 (Hex) Last 4 bits 0001
Ring parameter server	C000-0000-0002 (Hex) Last 4 bits 0010
Active monitor and ring parameter server	C000-0000-0003 (Hex) Last 4 bits 0011

12.6 PRIORITY MANAGEMENT IN

TOKEN RING LAN

TOKEN RING LAN

There are eight levels of priority, which are indicated by the P and R bits of the AC field of the token and data frames. The lowest priority level is 000 and the highest priority level is 111. P bits indicate the current priority and the R bits indicate the reserved priority in the token and data frames.

A station can capture the token if it has waiting data frames having priority same or above the current priority level (p) as indicated by the P bits of the token. It sends these data frames till expiry of THT. In these frames it sets reserved priority level (r) as indicated by R bits equal to 0. When these frames pass through the station downstream, each station raises the reserved priority level (r) by changing the R bits if it has higher priority frames waiting in the queue. Since there are eight priority levels, each station maintains eight queues. Only upward revision of reserved priority can be done.

When the frame returns to the originating station, its RRR field indicates the highest priority level (r) demanded by the stations on the ring. When it is time to release the token, the originating station sets new priority level in the PPP field of the token equal to the highest of the following and releases the token.

- Last priority level (p)
- Reserved priority (r)
- Priority (w) of the waiting frames at the station.

The R bits in the new token are set depending on the new value of PPP bits as determined above.

- If $PPP = p$, then $RRR = \max [r, w]$.
- If $PPP = \max [r, w]$, then $RRR = 0$.

12.6.1 Stacking Station

Note that new priority level cannot be less than the last priority level (p). It can be higher than p , if r or w is greater than p . If the priority level is not downgraded, it will rise eventually to its highest level and will remain there. Therefore, the station that releases token with raised new priority level, becomes responsible for downgrading the priority to the previous level. It puts the previous priority value in its memory stack and waits for the token to pass by.

This station is called stacking station. When the station sees the token with the raised priority level passing by, it changes the P and R bits as under bits to restore the priority level to its previous value.

$$\left. \begin{array}{l} \text{PPP} = r, \quad \text{RRR} = 0, \quad \text{if } (s < r) \\ \text{PPP} = s, \quad \text{RRR} = r, \quad \text{if } (s > r) \end{array} \right\} \text{ where } s \text{ is stacked priority level.}$$

It is possible for a station to have several stacked priority levels. After having released token with a higher priority level, a station may get more frames to transmit, that have even higher priority and the station may be required to raise the priority level again. The station, therefore, stacks all the previous values of priority and restores the priority levels whenever there is opportunity as explained in the last paragraph.

EXAMPLE 12.1 Four stations A, B, C, and D are on IEEE 802.5 token ring in the sequence A-B-C-D. Each station has one data frame to transmit. The priority of their data frames is as below.

A : Priority = 0, B : Priority = 2, C : Priority = 4, D : Priority = 4

A receives token with PPP = 0 and RRR = 0. Write the sequence of data and token frames that are transmitted on the ring. Indicate the priority levels (PPP) and reserved priority (RRR) on these frames.

Solution

F(P, R)/T(P, R): Data/token frame with priority P and reserved priority R				
A	B	C	D	Remarks
T(0, 0)				A seizes the token.
F(0, 0)	F(0, 2)	F(0, 4)	F(0, 4)	A releases its frame. B and C reserve their priorities.
T(4, 0)	T(4, 2)	T(4, 2)		A releases token with P = 4. B makes R = 2. C seizes the token.

	F(4, 0)	F(4, 4)		C releases its frame with P = 4. D sets R = 4.
F(4, 4)	F(4, 4)	T(4, 4)	T(4, 4)	C releases token with P = 4. D seizes the token.
		F(4, 0)		D releases its frame with P = 4.
F(4, 0)	F(4, 2)	F(4, 2)	T(4, 2)	B sets R = 2. D releases token with P = 4 and R = 2.
T(2, 0)	T(2, 0)			A downgrades priority P to reserved priority 2.
	F(2, 0)	F(2, 0)	F(2, 0)	B releases its frame with P = 2.
F(2, 0)	T(2, 0)	T(2, 0)	T(2, 0)	B releases token with P = 2.
T(0, 0)				A downgrades priority P to reserved/stacked priority 0.

12.7 RING MANAGEMENT IN TOKEN RING LAN

Token ring local area network requires several management functions to be performed for its proper operation. These functions include:

- Selection of active monitor station
- Identifying the upstream neighbour
- Initialization for entry of a new station in the ring
- Checking for duplicate addresses
- Locating the failed links/RIUs

- Removing the perpetually circulating frames and fragmented frames from the ring
- Purging the ring
- Releasing the new token when the ring is brought up and when the token is lost
- Providing a common clock to all the stations in the ring.

The last four functions are assigned to the active monitor station. The active monitor station is selected based on the highest MAC address. All other stations are called standby monitor station. They keep a watch on the functioning of the active monitor station. If the active monitor fails on any account, the standby monitor stations start active monitor selection process after timeout.

12.7.1 Active Monitor Station Selection

When the stations are powered on, they are in insert mode (Figure 12.2b), and compete for acquiring active monitor status. Any station can be active monitor station depending on its address. At any point of time the station with highest address is selected as the active monitor station. The steps involved in the selection process for active monitor station are as follows:

1. All stations send Claim-Token (CT) frames downstream.
2. When a station receives a CT frame from the upstream neighbour, it checks the source address. If the received CT frame has higher source address than its own address, the station stops sending CT frames, enters repeater mode and lets the received CT frames pass by. If the SA in the received CT frame is lower, it continues sending its own CT frames.
3. When a station receives a CT frame that has its own address, it implies that the CT frame released by it downstream has circulated round the ring and all other stations are in repeater mode. The station assumes the active monitor status.
4. The active monitor station inserts 24-bit buffer in the ring.
5. The active monitor station sends purge-frame in the ring to ensure that there are no other frames in the ring. When the purge-frame is received back by the active monitor station, the ring is successfully purged.

12.7.2 Upstream Neighbour Determination

The token rings are unidirectional. Each station needs to know the address of its adjacent active upstream neighbour. This information is used for identification of the fault domain when an error occurs. The steps involved in determination of upstream active neighbour are as follows:

1. The process is initiated by active monitor station which releases Active-Monitor-Present (AMP) frame downstream with $AC = 00$. The destination address is broadcast address.
2. The downstream neighbour sets $AC = 11$ and repeats the frame downstream. It also takes down the source address of the AMP frame as Upstream-Neighbour-Address (UNA). It resets its AMP timer. Then it sends downstream a Standby-Monitor-Present (SMP) frame with its address in the SA field, broadcast address in the DA field and $AC = 00$.
3. The next downstream station first receives AMP frame with $AC = 11$. It resets its AMP timer and repeats the AMP frame downstream. Next it receives SMP frame from its upstream neighbour. It notes the UNA, sets $AC = 11$ in the SMP frame and repeats the frame downstream. This station also generates its SMP frame in the same manner for the next downstream station.
4. This process is carried out by every station on the ring. Thus, all the stations have their UNA and have AMP timers reset.

AMP/SMP frames are received by their respective originators after going round the ring and are removed by them.

Active monitor station initiates the above process every 7 seconds and ensures that it is completed. Thus, every station expects to receive AMP frame at the interval of every 7 seconds and maintains a timer for the same. If it is not received, active monitor selection procedure (Section 12.7.1) is restarted.

12.7.3 Token Management

Active monitor carries out the following tasks in respect to token management.

Release of first token. After active monitor is selected, the ring is purged and the AMP/SMP frames are circulated, the active monitor station then releases the first token on the ring.

Lost tokens. When the active monitor station does not see token passing by for 10 ms, it carries out ring purge and releases new token.

12.7.4 Initialization Process for a New Station

When a new station is inserted in the ring, it waits for the token and then sends a Duplicate-Address-Test (DAT) control frame with $AC = 00$. The SA and DA fields of the DAT frame contain the address of the station. Every station on the ring compares the DA address on the DAT frame with its own address. If a station finds that the DA is its own address, it sets $AC = 11$. When the originating station receives the DAT frame after it has circulated round the ring, it checks the AC field. If it is 00, then there is no other station on the ring with its address. If duplicate address is found, the station aborts insertion in the ring.

After duplicate address check is successfully over, the station waits for AMP, SMP or ring purge frame. These frames indicate presence of active monitor station. It participates in the upstream neighbour determination process when it receives AMP or SMP frame. If these frames are not received, it starts active monitor selection process.

12.7.5 Persistent Circulating Frames

Normal operation of token ring LAN requires the sending station to remove the frame which it released on the ring after the frame returns back to it. It is possible that the frame is not removed from the ring due to some error condition and it circulates persistently on the ring. Some situations when this happens are given below:

- The sending station goes down immediately after releasing the frame on the ring.
- There is some error in the SA field and therefore the sending station does not recognize its own frame.
- A new station enters the ring and activates its RIU relay when a frame was crossing its RIU. The frames gets damaged and is not removed by the source.

Persistent circulating frames need to be removed from the ring because the ring cannot be used by any station. There is also the need to generate new token. Responsibility of removing such frames is on the active monitor station. M bit of AC field enables the active monitor station to detect and remove the persistent circulating frames (Figure 12.7b). A frame when released on the ring by the originating station has $M = 0$ (Figure 12.10). The active monitor station sets $M = 1$ in the frame when the frames pass by it. If this frame is not removed by the

originating station, the second trip of the frame on the ring is immediately detected by the active monitor station when it finds M bit already set to 1. The active monitor station removes the frame, cleans the ring by sending a purge frame and then, generates a new token.

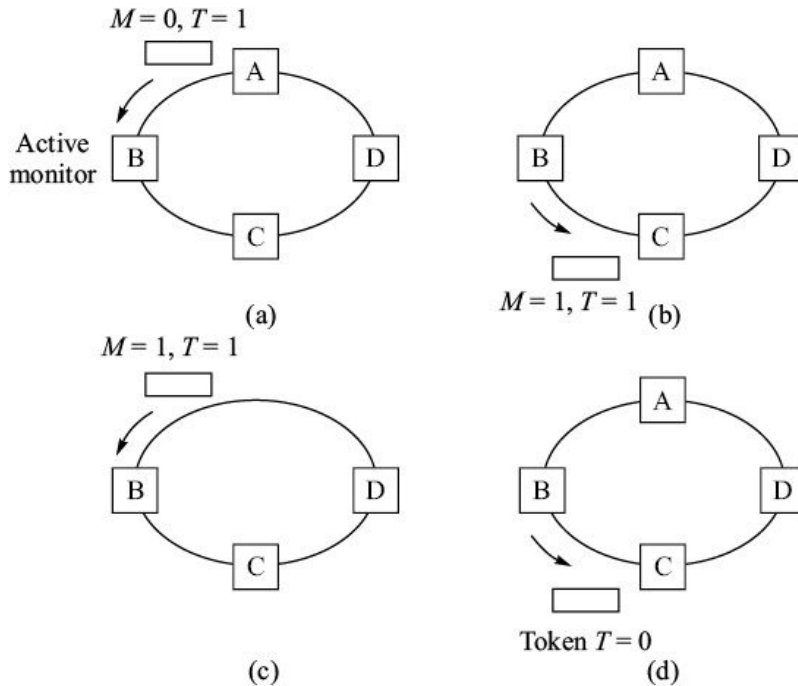


Figure 12.10 Removal of persistent circulating frames by active monitor station.

12.7.6 Master Clock Generation

The active monitor station acts as the master source of clock for all the stations. It sends the frames to the downstream stations using its internal master clock. All other stations on the ring derive their local clock from the incoming bit stream using Phase Lock Loop (PLL). The recovered clock is also used by them for sending data frames to the downstream stations.

Repeated recovery of clock using PLLs results in accumulation of phase jitter. The active monitor station is provided with an elastic buffer (shift register) of 6 bits length. This buffer is in addition to the 24-bit buffer for introducing minimum required delay in the ring. 6-bit buffer can accommodate jitter of 3 bits and can regenerate the signals using its master clock. Regeneration of the signals at the active master station removes the accumulated jitter.

12.7.7 Beacons

If a serious failure (e.g., break in transmission medium, or RIU failure) occurs in any part of the network, a procedure called *beaconing* alerts all the stations on

the ring and recovery procedures are initiated. The network failure is detected on expiry of timers associated with AMP/SMP or token passing. The affected station starts sending Beacon Supervisory (BCN) control frames. BCN frames contain source address and the address of upstream neighbour. They are sent periodically. If the upstream neighbour receives BCN, it removes itself from the ring by deactivating its bypass relay. When subsequent BCN frames arrive their home address, it implies that the failure has been rectified by removal of the upstream neighbour. If BCN frames do not reach home within 16 seconds, the station removes itself from the ring. The ring initialization procedure is carried out to bring up the network again.

12.8 TOKEN BUS LAN

Token bus local area network was designed for production lines with the objective of realizing guaranteed response time. It is very similar to token ring local area network in operation. The physical topology of the network is bus but the stations are connected in a logical ring topology (Figure 12.11). The logical topology follows the address hierarchy with the station with lowest address connected to the station with the highest address. Each station knows the identities of the preceding station and the succeeding station. There is no relation between the physical location of a station on the bus to its address.

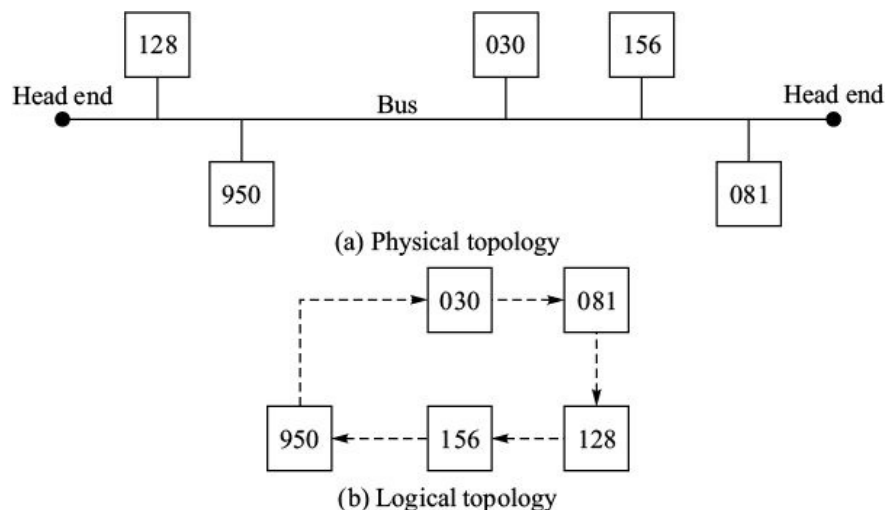


Figure 12.11 Token passing sequence in a bus.

12.8.1 Media Access Control in Token Bus LAN

The basic operation of a token bus LAN is as follows:

- The access to the interconnecting bus is regulated by a token. At any time, only the station that holds the token has the right to transmit its data frames on the bus. Each frame carries source and destination addresses. A station may send one or more frames while it is holding the token.
- All stations are ready to receive frames at any time except when holding a token.
- The token must be released before timeout with the address of the next station in the sequence.
- The released token is taken over by the station whose address is on the token. To maintain continuity of communication, it is necessary for each station to take over the token even if it does not have any frames to send. It can release the token immediately for the next station.
- In one cycle, each station gets one opportunity to transmit. Thus each station gets a fair chance to send its frames in round robin fashion. It is possible to give more than one opportunity to a station in one cycle by assigning it more than one address to it.

12.8.2 Frame Structure of Token Bus LAN

IEEE 802.4 MAC sublayer of token bus operates under LLC sublayer. Figure 12.12 shows the MAC frame structure as specified in IEEE 802.4. It consists of the following fields:

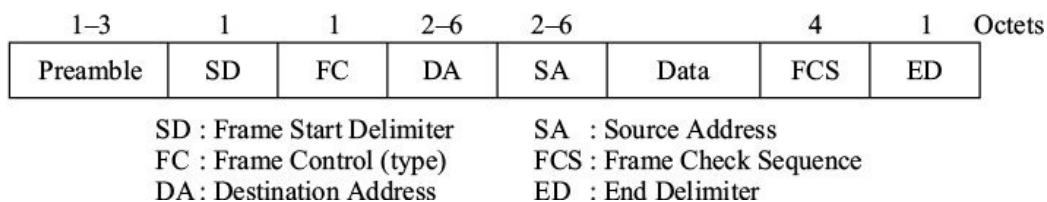


Figure 12.12 Format of IEEE 802.4 frame.

Preamble. The preamble is 1 to 3 octets long pattern. It enables bit synchronization.

Start delimiter (SD). It is one octet long unique bit pattern which marks the start of the frame. As in token ring, SD contains non-data codes for identification.

Frame control (FC). The frame control field indicates type of the frame—data frame or control frame. The token frame is one of the control frames. Figure 12.13 lists the FC field of various types of frames. The functions of various frames are described later. This field is one octet long.

Destination address (DA). The destination address field is 6 octets long.

Source address (SA). The source address field is 6 octets long. As before the destination and source address fields can be 2 octets long but 2-octet long addresses are seldom used.

Data. Data field contains the LLC frame. Maximum size of the MAC frame excluding SD/ED fields should not exceed is 8191 octets.

Frame check sequence (FCS). Frame check sequence is 4 octets long and contains CRC. It checks on DA, SA, FC, and data fields.

End delimiter. It is unique bit pattern which marks the end of the frame. It is one octet long. As in token ring, ED contains non-data codes for identification.

	1	2	3	4	5	6	7	8
(a) Claim-token	0	0	0	0	0	0	0	0
(b) Solicit-successor-1	0	0	0	0	0	0	0	1
(c) Solicit-successor-2	0	0	0	0	0	0	1	0
(d) Who-follows	0	0	0	0	0	0	1	1
(e) Resolve-contention	0	0	0	0	0	1	0	0
(f) Token	0	0	0	0	1	0	0	0
(g) Set-successor	0	0	0	0	1	1	0	0
(h) Data frame	0	1	M	M	M	P	P	P
(i) Station management	1	0	M	M	M	P	P	P

Figure 12.13 FC field of IEEE 802.4.

12.8.3 Response Window

When a control frame is transmitted on the bus by a station, it must ensure that the frame is received and responded by the receiving station. For example, if a station passes token frame to its successor, the successor must respond by releasing a data/control frame on the bus. Else the sending station will assume

that the token has not been passed to its successor successfully. The sending station, therefore, monitors the bus for some response activity after it sends a control frame. The maximum time the sending station waits for the response activity is called response window. If there is no response activity within the response window, it takes a corrective action (e.g., retransmission of the control frame).

The response window is two times the end-to-end propagation time on the bus plus the expected frame processing time required by the receiver to respond.

12.8.4 Token Bus Management

Token bus LAN requires considerable maintenance effort like token ring LAN. The following are the main maintenance functions:

- Bus initialization
- Addition of new stations to the bus
- Deletion of inactive successor.

These functions are performed using control frames. Control and management of the token is distributed among the active stations. Each station can initiate and respond to the control frames such as claim-token, solicit-successor, set-successor, and who-follows frames. We will not go into detailed procedures of these functions. General approach taken for these functions based on IEEE 802.4 standard is given below.

Bus initialization. Each station in the network has a timer known as inactivity timer, which is reset whenever a transmission is heard on the bus. If the token is lost, all the activity on the network comes to halt and the inactivity timers of the stations start expiring depending on when they were reset. A station whose inactivity timer has expired starts bus initialization process. It involves the following steps:

1. It sends a claim-token frame with information field having length that is integer number of response window times. The integer is 0, 2, 4, or 6 depending on the first two address bits of the station.
2. When a station finds a claim-token frame, it responds with claim token frame if its address is higher than the received claim-token frame. Else, it withdraws from trying to become first owner of the token.

3. The process is repeated using other bits of the address till the station with the highest address is identified. The station with the highest address, thus, becomes the first owner of the token.
4. The first owner of the token sends a solicit-successor-1 frame to identify the next station in the logical ring. The solicit-successor-1 frame specifies a range of addresses. The station within this range responds with set-successor frame. If there are more than one responses, contention is resolved using resolve-contention frame.
5. Once the successor is identified, the first owner of the token passes token to its successor which repeats the process to identify its successor. The last station having the lowest address uses set-successor-2 frame to close the ring.

Addition of new stations. Each station sends out solicit-successor frame periodically to enable entry of new stations in the logical ring. Solicit-successor frame is sent with the source address in the SA field and address of the existing successor in the DA field. New station having its address within the range SA-DA replies with set-successor frame. Response from the new station must be given within the response window. On receipt of the set-successor frame, the token is next passed to the new station. If there are multiple responses to the solicit-successor frame, an arbitration procedure using resolve-contention frame is initiated.

Deletion of inactive successor. Token passing will halt if any station in the logical ring becomes inactive. Inactive stations must be removed from the ring. The steps involved in the process are as follows:

1. A station sends token frame to its known successor. It listens to any subsequent activity on the bus. If it hears a frame being transmitted, it assumes that the token has successfully passed to the successor.
2. If there is no subsequent activity for a period equal to response window, it assumes that the token is lost and therefore retransmits the token to the known successor.
3. If there is still no activity on the bus, it assumes that the successor has failed and it proceeds to search the new successor. It broadcasts who-follows frame with the current successor's address in the data field.
4. On receipt of the frame each station compares its current predecessor's

address with the address in data field of the who-follows frame. The station whose predecessor's address matches, replies with set-successor frame. On receipt of the set-successor frame, new successor is established and the token is passed to the new successor.

12.8.5 Priority Operation in Token Bus

An optional priority scheme is specified in the IEEE 802.4 standard. There are four classes of priorities—6, 4, 2, and 0. The priority scheme is implemented using two timers, token rotation timer and token holding timer.

Token rotation timer. Each station keeps a token rotation timer which indicates the time that has expired since it last received the token. Thus, when the station receives the token again, the timer indicates the Token Rotation Time (TRT). TRT is used for setting the token holding timer and then restarted from zero for the next measurement of TRT.

TRT has a defined upper threshold, called Target Token Rotation Time (TTRT). If a station receives the token earlier than TTRT, it can hold the token for duration $(TTRT - TRT)$ and send data frames of any priority during this period. But the frames must be sent in order of priority.

If the token arrives late, *i.e.* the quantity $(TTRT - TRT)$ is negative (or zero), the station must release the token immediately. TTRT can be defined separately for priority 4, 2, and 0.

Token holding timer. Token holding timer maintains the time for which the token is held by a station. It is initialized to $(TTRT - TRT)$ as soon as the token is received and starts counting time downwards. As soon as it reaches zero, the token is released after completing the transmission of the last frame.

Token holding time for priority-6 frames (P-THT). For priority-6 frames, special allocation of token holding time is made. A station can hold the token for a defined duration for transmitting priority-6 frames in the queue. This duration is termed as Token Holding Time (THT) in most of the texts. We will call it Priority Token Holding Time (P-THT) to convey its intended significance. Thus irrespective of the value of $(TTRT - TRT)$, a station can always send its priority-6 frame every time it receives the token for a period equal to P-THT.

There is another timer, TRT-M, for maintenance purpose. If this timer is not expired, a station can send solicit-successor frame to include new stations in the token passing sequence. Figure 12.14 illustrates the use these timers for sending

the frames.

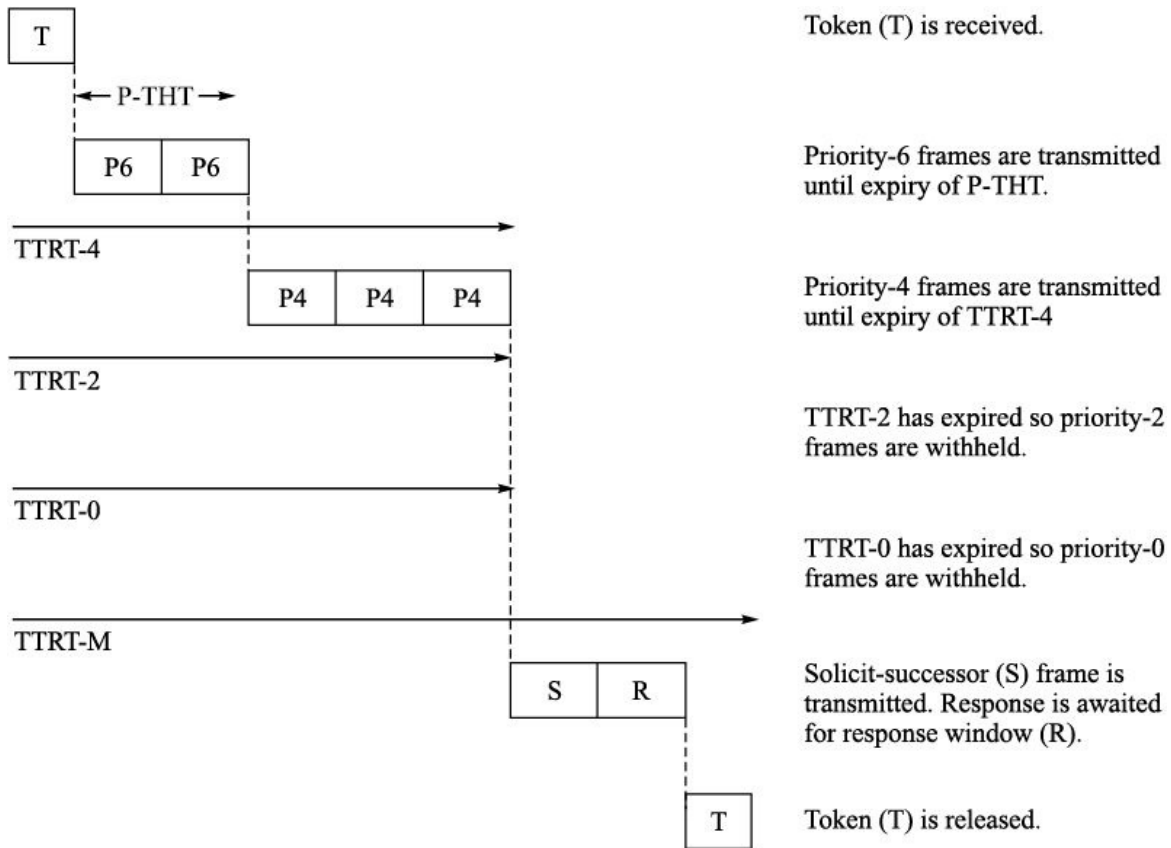


Figure 12.14 Timers for implementing priority.

EXAMPLE 12.2 In Figure 12.15, stations A, B, C, and D always have two priority-6 data frames to send. All the stations also have large number lower priority data frames to send. All the frames have equal sizes and take one unit of time to transmit. The propagation delay on each of the interconnecting links is negligible. If TTRT is 16 and P-THT for priority-6 data frames is 3 units of time, find the number of frames transmitted by each station in four rounds of the token after an initialization round zero in which no data frames are transmitted. Assume that the token size is small and its transmission time can be neglected. The token rotation follows the sequence A B C D A.

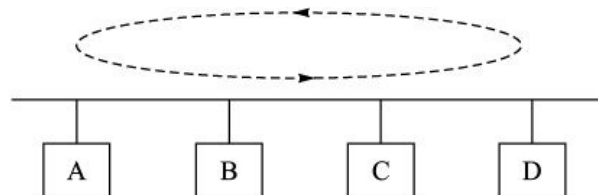


Figure 12.15 Example 12.2.

Solution The following table lists the status of TRT and frames transmitted in each round of token. PF is the number of priority-6 data frames transmitted. NF is the number of the lower priority data frames transmitted. TRT, PF, and NF at each station after each round are calculated as follows:

TRT = Token propagation time on the ring (= 0) + number of frames transmitted after the last reset of TRT.

Number of PF = Since P-THT is 3, the stations always send their two waiting priority-6 data frames. Thus, PF = 2. This is irrespective of the value of TRT.

Number of NF = 0 if $(TTRT - TRT) \leq 2$
 $(TTRT - TRT) - PF$ if $(TTRT - TRT) > 2$

Round	Station A	Station B	Station C	Station D
0	TRT is reset	TRT is reset	TRT is reset	TRT is reset
	TRT = 0	TRT = 2 + 14 = 16	TRT = 2 + 14 + 2 = 18	TRT = 2 + 14 + 2 + 2 = 20
	TTRT - TRT = 16	TTRT - TRT = 0	TTRT - TRT = -2	TTRT - TRT = -4
1	TRT is reset	TRT is reset	TRT is reset	TRT is reset
	PF = 2	PF = 2	PF = 2	PF = 2
	NF = 14	NF = 0	NF = 0	NF = 0
	TRT = 22	TRT = 8	TRT = 14	TRT = 14
	TTRT - TRT = -6	TTRT - TRT = 8	TTRT - TRT = 2	TTRT - TRT = 2
2	TRT is reset	TRT is reset	TRT is reset	TRT is reset
	PF = 2	PF = 2	PF = 2	PF = 2
	NF = 0	NF = 6	NF = 0	NF = 0
	TRT = 14	TRT = 14	TRT = 8	TRT = 14
	TTRT - TRT = 2	TTRT - TRT = 2	TTRT - TRT = 8	TTRT - TRT = 2
3	TRT is reset	TRT is reset	TRT is reset	TRT is reset
	PF = 2	PF = 2	PF = 2	PF = 2
	NF = 0	NF = 0	NF = 6	NF = 0
	TRT = 14	TRT = 14	TRT = 14	TRT = 8
	TTRT - TRT = 2	TTRT - TRT = 2	TTRT - TRT = 2	TTRT - TRT = 8
4	TRT is reset	TRT is reset	TRT is reset	TRT is reset
	PF = 2	PF = 2	PF = 2	PF = 2
	NF = 0	NF = 0	NF = 0	NF = 6

12.8.6 Physical Specifications

Token bus LANs operate at 1, 5 or 10 Mbps using analog signaling over 75 ohms coaxial cable. The following two types of transmission systems are used:

- Carrier band (single channel)
- Broadband (multiple channels).

Both the systems use modulation techniques to reduce the effect of noise present in the manufacturing environment. The carrier band system is based on single channel bidirectional transmission using phase coherent FSK modulation. Frequencies used to represent binary 1 and 0 are in the ratio of 1:2.

Broadband system employs multiple channel unidirectional transmission using combination of phase and amplitude modulation. It uses conventional CATV components and a remodulator at the headend. Separate carriers are used for transmit and receive directions. There can be several transmit and receive carriers and each carrier provides 5 or 10 Mbps data rate. Broadband LAN can cover a span of several kilometres.

12.9 FIBRE DISTRIBUTED DATA INTERFACE (FDDI)

Fibre Distributed Data Interface (FDDI) LAN standard is based on optical fibre as transmission medium. This standard was developed by ANSI and is given in X3.139-1987. The corresponding ISO standard is ISO 9314. It operates under IEEE 802.2 LLC sublayer allowing it to be integrated easily with other IEEE LANs.

FDDI was originally conceived as a back-end network interconnecting several hosts and high speed peripherals. It can also be used as backbone network interconnecting several front-end LANs. As a backbone LAN, FDDI has built-in fault-tolerant features that make it resilient to the faults.

12.9.1 Physical Topology

FDDI implements a fault-tolerant architecture using dual ring topology, which consists of a primary ring and a secondary ring (Figure 12.16a). The MAC sublayer is attached to the primary ring. For the signals flowing on the secondary ring, the station acts only as repeater.

If a station on the ring fails, or a cable section is damaged, the dual ring is automatically wrapped into a single ring. Data continues to be transmitted on the FDDI ring without any performance impact during the wrapped condition of the ring. Figures 12.16b and 12.16c show how

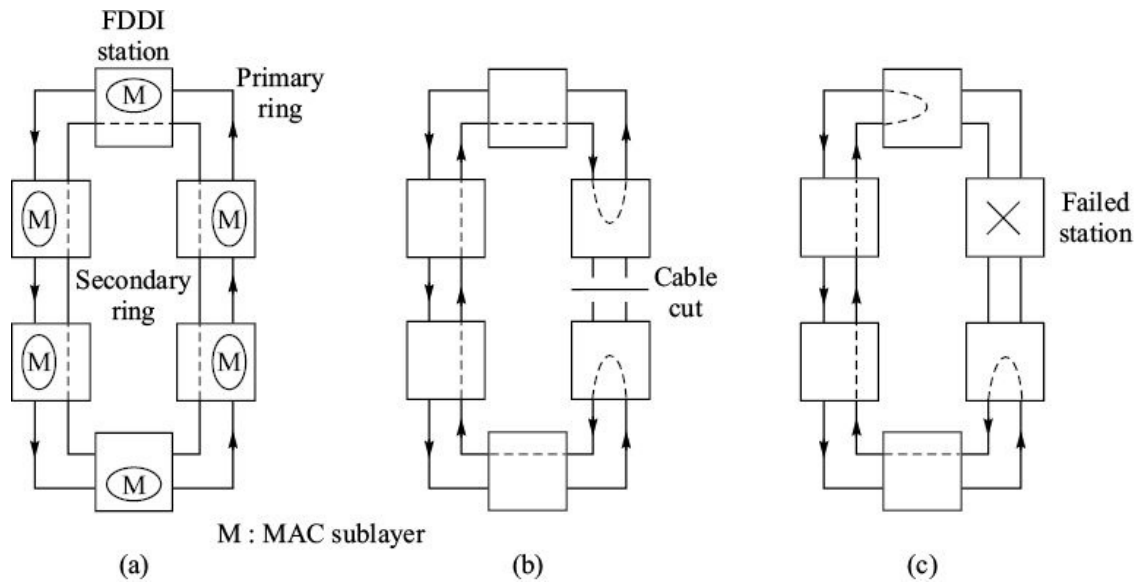


Figure 12.16 Fault-tolerant FDDI ring.

the damaged cable section and the faulty station are bypassed using the secondary ring. It should be noted that fault tolerance in FDDI is against single failure only.

Optical bypass switch is also provided in the interface to connect the incoming optical port directly to the outgoing optical port so that a station is completely bypassed. The optical switch is equivalent of relay in the RIU of token ring. Optical switch consists of micro mirrors that rotate when energized to deflect the incident light to the right port. Benefit of optical switch is that the ring need not be wrapped if a station has been withdrawn for extended period.

12.9.2 Types of FDDI Stations

FDDI defines four types of devices that are connected on an FDDI network (Figure 12.17):

1. Dual Attachment Station (DAS)
2. Single Attachment Station (SAS)
3. Dual Attachment Concentrator (DAC)
4. Single Attachment Concentrator (SAC).

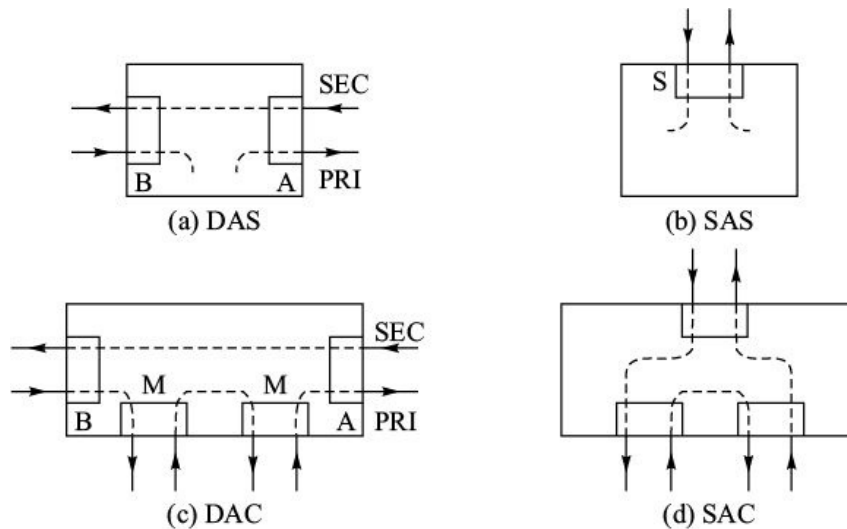


Figure 12.17 FDDI devices.

DAS. A DAS attaches to both the rings, primary and secondary, through its dual media interface couplers, MIC-A and MIC-B (Figure 12.17a).

SAS. An SAS attaches only to the primary ring through a concentrator (DAC or SAC). It has one slave port, called MIC-S that connects the SAS to the MIC-M port of the concentrator (Figure 12.17b). MIC-S is less expensive compared to the dual couplers used in a DAS. A SAS can be readily withdrawn from the ring without affecting the ring operation.

DAC. A DAC attaches to both the rings through its dual couplers, MIC-A and MIC-B (Figure 12.17c). It connects several SASs to the ring through its master coupler ports (MIC-M).

SAC. An SAC is connected to the ring through a DAC. It provides secondary level of concentration (Figure 12.17d). It is less reliable as it connects only to the primary ring.

Figure 12.18 shows typical FDDI LAN set-up using DAC, SAC, DAS, and SAS.

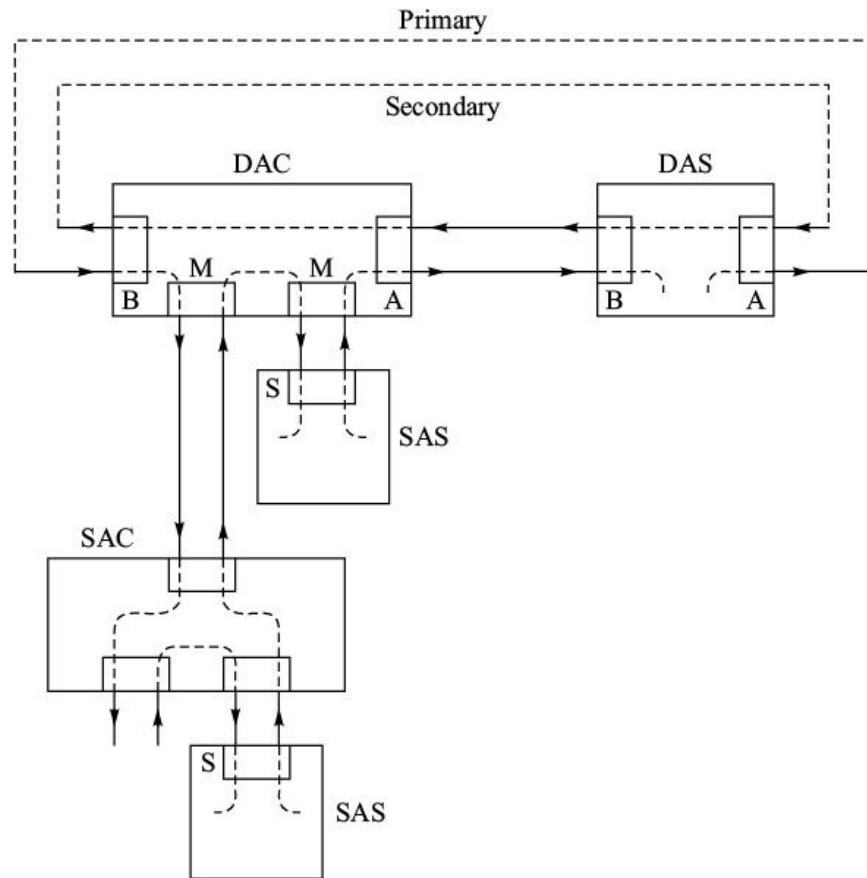


Figure 12.18 FDDI topology using DAC and SAC.

12.9.3 Types of Services

FDDI offers two types of services:

- Synchronous service and
- Asynchronous service.

Synchronous service. Synchronous service is for real time applications for which bandwidth and response time are the two critical parameters and are predictable. In FDDI, each station is assured of token availability and minimum token holding time for transmitting its time-critical data frames. Such frames are referred to as synchronous frames. The allocation of ring bandwidth for synchronous transmission is done mutually by all the stations.

Asynchronous service. Asynchronous service provides dynamic bandwidth and is suitable for usual bursty data traffic. Data frames of asynchronous service are called asynchronous frames.

12.10 MEDIA ACCESS CONTROL IN FDDI

Media access control in FDDI is almost same as in IEEE 802.5 token passing ring. A token is circulated in the ring and a station wishing to transmit its data frames captures the token. It transmits the data frames and then releases the token. Early token release option is used in FDDI. The frame carries source and destination address so that when it passes the destination address, the destination station retains a copy of the frame. The frame is finally removed from the ring by the source when it comes back.

12.10.1 Frame Fragmentation

In IEEE 802.5 token ring, the ring size is kept less than the frame size so that a frame comes back to the source after circulating round the ring while it is still being transmitted by the source. In IEEE 802.5, the source is able to remove the frame completely because it has not yet released the token and the ring is broken at the source.

Unlike IEEE 802.5 token ring, the ring size is bigger than the frame size in FDDI. Therefore, the source completes transmission of a frame before the first bit of the frame returns back to it. The source releases the token immediately after transmitting the frame. The downstream stations can seize the token and transmit their frames also. Therefore, a train of several frames travels on an FDDI ring. As shown in Figure 12.19, station A generates a frame and releases token. Station B repeats the frame, converts the token into its frame and releases the token. Station C repeats first two frames, seizes the token, and transmits its frame also.

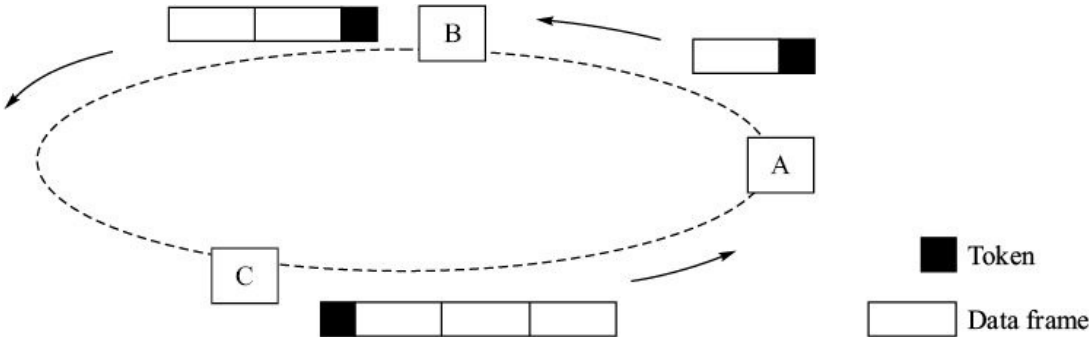


Figure 12.19 Frame generation in FDDI.

Having completed the transmission of its frames and having released the token, a station enters the repeater mode. When the frame released by station A

returns back, the station is already in the repeater mode. The first four fields of the frame (preamble, SD, FC, and DA) do not give any clue to A that this is the same frame it transmitted. Therefore, it repeats the first four fields. Then it comes across SA field and realizes that this was the frame it transmitted. It stops repeating the rest of the bytes of the frame immediately. But by that time, a fragment of the frame is already in the ring. Such frame fragments are continuously generated in the ring and must be removed from the ring.

12.10.2 Fragment Removal

FDDI adopts byte stripping method to purge the ring of such fragments. Each station has one byte buffer between its incoming and outgoing ports. It is similar to one bit buffer in token ring. While repeating a frame, a station keeps looking for End Delimiter (ED). If it does not find ED, it strips the last byte still in the buffer. As the frame fragment that does not have ED field passes through the stations, each station strips the last one byte from it till it is completely removed from the ring.

12.10.3 Priority Management in FDDI

In FDDI, the synchronous data frames are given priority over asynchronous data frames. The transmission of these two categories of frames is prioritized in the same manner as it is done for priority-6 frames in token bus using two timers at each station, token rotation timer and token holding timer. Target Token Rotation Time (TTRT) and Token Holding Time for Priority Frames (P-THT) are defined in the same manner as in token bus. The following example illustrates the application of these parameters in the ring operation.

EXAMPLE 12.3 In Figure 12.20, stations A and C always have only two synchronous data frames to send. Stations B and D never have synchronous data frames, but they always have large number asynchronous data frames to send. All the frames have equal sizes and take one unit of time to transmit. The propagation delay on each of the interconnecting links is one unit of time. If TTRT is 20 and P-THT for synchronous data frames is 2 units of time, find the number of frames transmitted by each station in five rounds of the token after an initialization round zero in which no data frames are transmitted. Assume that the token size is small and its transmission time can be neglected.

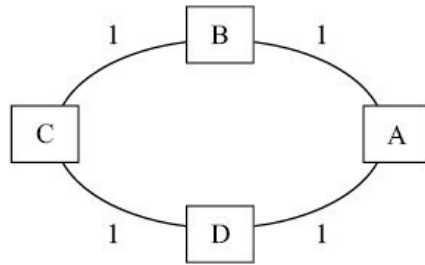


Figure 12.20 Example 12.3.

Solution The following table lists the status of TRT and frames transmitted in each round of token. SF is the number of synchronous data frames transmitted. AF is the number of the asynchronous data frames transmitted. TRT, SF, and AF, at each station after each round are calculated as below:

TRT = Token propagation time on the ring (= 4) + number of frames transmitted after the last reset of TRT.

Number of SF = Since P-THT is 2, stations A and C always send their two waiting synchronous data frames. This is irrespective of the value of TRT.

Number of AF = Stations B and D calculate (TTRT-TRT) and if it is positive number, they transmit as many asynchronous data frames.

Round	Station A	Station B	Station C	Station D
0	TRT is reset	TRT is reset	TRT is reset	TRT is reset
1	TRT = 4 TRT is reset SF = 2	TRT = 4 + 2 TRT is reset = 20 - 6 = 14	TRT = 4 + 2 + 14 = 20 TRT is reset, SF = 2	TRT = 4 + 2 + 14 + 2 = 22 TRT is reset AF = 0
2	TRT = 4 + 2 + 14 + 2 = 22, TRT is reset, SF = 2	TRT = 4 + 14 + 2 + 2 = 22 TRT is reset, AF = 0	TRT = 4 + 2 + 2 = 8, TRT is reset SF = 2	TRT = 4 + 2 + 2 = 8, TRT is reset AF = 20 - 8 = 12
3	TRT = 4 + 2 + 2 + 12 = 20, TRT is reset, SF = 2	TRT = 4 + 2 + 12 + 2 = 20, TRT is reset, AF = 20 - 20 = 0	TRT = 4 + 2 + 12 + 2 = 20, TRT is reset, SF = 2	TRT = 4 + 12 + 2 + 2 = 20, TRT is reset, AF = 20 - 20 = 0
4	TRT = 4 + 2 + 2 = 8, TRT is reset SF = 2	TRT = 4 + 2 + 2 = 8, TRT is reset AF = 20 - 8 = 12	TRT = 4 + 2 + 2 + 12, = 20, TRT is reset, SF = 2	TRT = 4 + 2 + 12 + 2 = 20, TRT is reset, SF = 20 - 20 = 0
5	TRT = 4 + 2 + 12 + 2 = 20, TRT is reset, SF = 2	TRT = 4 + 12 + 2 + 2 = 20, TRT is reset, SF = 20 - 20 = 0	TRT = 4 + 2 + 2 = 8, TRT is reset SF = 2	TRT = 4 + 2 + 2 = 8, TRT is reset SF = 20 - 8 = 12

12.11 MAC FRAME FORMAT IN FDDI

The FDDI frame format is similar to the format of token ring frame. Figure 12.21 shows the formats of the token and data/control frames. In FDDI the field

sizes are specified in terms of symbols. Each symbol is 4B/5B encoded nibble (4 bits).

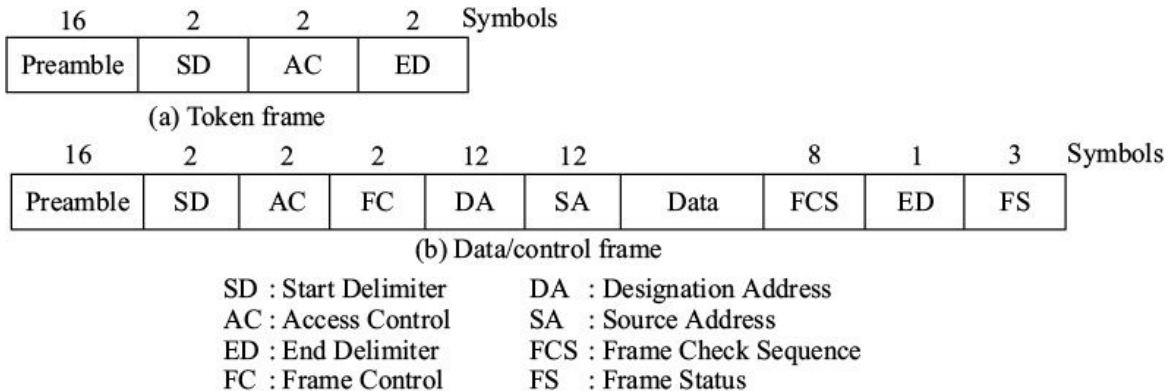


Figure 12.21 Formats of FDDI frames.

Preamble. The preamble is for clock synchronization as there is no master clock source in FDDI. It consists of 16 idle symbols (11111).

Start delimiter (SD). It is the flag indicating start of the frame. It consists of two symbols, J (11000) and K (10001). J and K contain code violations for their identification.

Frame control (FC). It is two symbols long field (Figure 12.22), and contains the following bits.

Class bit (C) : It indicates whether the data field contains synchronous (C = 1) or asynchronous (C = 0) data.

Address length bit (L) : It indicates size of the address field, 4 (L = 0) or 12(L = 1) symbols.

Frame format bits (FF) : These two bits identify type of the frame, data frame or control frame.

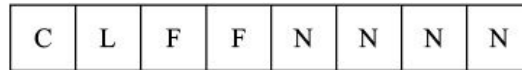
- | | |
|----------------|--|
| FF = 00, C = 0 | SMT ² control frame |
| FF = 00, C = 1 | MAC control frame, e.g. token, claim token, beacon |
| FF = 01 | Data frame with LLC PDU |
| FF = 10, 11 | Reserved |

Control bits (NNNN) : NNNN bits identify the type of control frame. For example,

- | | |
|------|-------------|
| 0010 | Beacon |
| 0011 | Claim token |

0000

Token



C : Class Bit
 L : Address Length Bit
 FF : Frame Format Bits
 NNNN : Control Function

Figure 12.22 Frame control field.

Destination address (DA). This is 4 or 12 symbols long. 4 symbols long addresses are never used, though the standard provides for them.

Source address (SA). This is 4 or 12 symbols long. 4 symbols long addresses are never used, though the standard provides for them.

Data. Data field can contain symbols corresponding to 0 to 4478 octets of data.

Frame check sequence (FCS). It is eight symbols long field and contains CRC sequence for error detection in FC, DA, SA, and data fields.

End delimiter (ED). It marks end of the frame. It is one symbol long and contains violation symbol T.

Frame status (FS). It is similar to the FS field of token ring and contains indicators for address-recognized, frame-copied, and error-detected. S and R symbols are used for this purpose.

12.12 PHYSICAL SPECIFICATIONS OF FDDI

FDDI uses 4B/5B encoding as indicated in Table 12.1. The 5B code set has 16 ($= 2^5 - 2^4$) spare 5-bit codes, eight of which are used as special control symbols (Q, I, H, J, K, T, S, and R). NRZ-I line code is used to modulate the optical signal. FDDI operates at 100 Mbps. Because of 4B/5B coding, the bit rate on the ring is 125 Mbps.

TABLE 12.1 4B/5B Code Set Used in FDDI					
Hex/Name	4-bit nibble	5-bit code	Hex/Name	4-bit nibble	5-bit code
0	0000	11110	C	1100	11010
1	0001	01001	D	1101	11011

2	0010	10100	E	1110	11100
3	0011	10101	F	1111	11101
4	0100	01010	I (Idle)		11111
5	0101	01011	J		11000
6	0110	01110	K		10001
7	0111	01111	T		01101
8	1000	10010	R		00111
9	1001	10011	S		11001
A	1010	10110	H (Halt)		00100
B	1011	10111	Q (Quiet)		00000

12.12.1 Ring Size and Number of Stations

An unbroken FDDI ring can have size of 100 kilometres with 500 dual attached nodes on the ring. The ring design is such that the maximum ring length does not exceed 200 km when the ring wraps. The nodes can have separation up to 2 kilometres on multimode fibre (62.5/125 mm) and up to 10 kilometres on single mode fibre.

SUMMARY

In Ethernet LAN, access to media is based on contention. The token passing LANs use a more disciplined approach for media access control. A token is rotated among stations in a sequence and station holding the token is authorized to transmit its frames. To ensure that fair opportunity is given to all the stations,

parameters like token holding time and token rotation time are defined. A priority mechanism is also implemented so that priority frames do not get delayed. It is possible that token may be lost and therefore mechanism for regenerating the token is required. Also a station need to know identity of its down stream neighbour to pass the token.

There are three token passing LAN technologies, token passing ring (IEEE 802.5), token passing bus (IEEE 802.4) and FDDI. The token ring LAN operates at 4 and 16 Mbps having maximum segment lengths of 385 and 11173 metres respectively. STP cables are used as the transmission medium. The token bus LAN operates at 1, 5, and 10 Mbps. It uses coaxial cable as transmission medium.

Optical fibre as transmission medium offers significant bandwidth advantage. Fibre Distributed Data Interface is an approved ANSI standard and provides a bit rate of 100 Mbps. It provides synchronous and asynchronous services and is suited for real time applications. It has dual ring architecture to make it fault tolerant. An unbroken FDDI ring can have size of 100 kilometres with 500 dual attached nodes on the ring. The nodes can have separation up to 2 kilometres on multimode fibre (62.5/125 mm) and up to 10 kilometres on single mode fibre.

EXERCISES

1. In token ring, AC, ED, and FS field are not covered by FCS for error detection. What could be the reason for sparing these fields from FCS check?
2. List possible scenarios that result in a corrupted frame to circulate round the ring forever. What is the fix built into the token ring to correct such situations?
3. The token frame in IEEE 802.5 does not have CRC. Justify why CRC is not required the way the protocol has been designed.
4. Four stations A, B, C, and D are on IEEE 802.5 token ring in the sequence A-B-C-D. Each station has one data frame to transmit. The priority of their data frames is as follows:
A: Priority = 2, B: Priority = 2, C: Priority = 4, D: Priority = 4
A receives token with PPP = 0 and RRR = 0. Write the sequence of data and token frames that are transmitted on the ring. Indicate the priority levels (PPP) and reserved priority (RRR) on these frames.

5. In Example 12.2, each of the stations A, B, C, and D has three priority-6 data frames and large number lower priority data frames to send. All the frames have equal sizes and take one unit of time to transmit. The propagation delay on each of the interconnecting links is negligible. If TTRT is 16 and P-THT for priority-6 data frames is 3 units of time, find the number of frames transmitted by each station in four rounds of the token after an initialization round zero in which no data frames are transmitted. Assume that the token size is small and its transmission time can be neglected. The token rotation follows the sequence ABCDA.
6. A token ring LAN consists of ten twisted pair segments of 100 metres each. If the speed of propagation is in a twisted pair is 2×10^8 metres/second and each RIU introduces a delay of 1 bit, calculate the time taken by a bit to make a complete round of the ring. Assume that the ring is operating at 4 Mbps.
7. In the token ring LAN, the first bit of the token should not come back to the station where it was produced until the whole token is transmitted. Calculate the minimum length a ring to operate properly at 16 Mbps. Assume that speed of propagation of electrical signals is 2×10^8 metres/second. What is the minimum length of the ring if there are ten stations on the ring and each station introduces one-bit delay?
8. An IEEE 802.5 token ring has five stations and total length of the ring is 230 metres. How many bits of delay must the monitor station insert into the ring? Assume that the ring operates at 16 Mbps and speed of propagation of electrical signals is 2×10^8 metres/second.
9. Explain how the token ring operation will be affected if
 - (a) the bye-pass relay was not provided in the ring.
 - (b) the monitor station did not have 24-bit shift register.
 - (c) if the RIU did not have the minimum one-bit delay.
 - (d) if the FS field of IEEE 802.5 frame was provided before the FCS field.
 - (e) if M-bit was not provided in of IEEE 802.5 frame.

1 We will describe source routing bridges in Chapter 14.

2 SMT : Station management.

13

Wireless Local Area Networks

The local area network types discussed so far in Chapters 11 and 12 use metallic pair (twisted/coaxial pair) or optical fibre as the physical transmission media and are categorized as wired local area networks. In this chapter we introduce another kind of local area network, wireless local area network which is based on wireless transmission medium. The wireless stations can form a local area network of their own or can be adjuncts to a wired local area network. We begin the chapter with configurations and associated definitions of wireless local area networks. Media access control methods, namely Distributed Coordination Function (DCF) and Point Coordination Function (PCF) are described in detail. We examine the layered architecture and IEEE 802.11 MAC frame structure before moving to the physical layer. In the physical layer, we first introduce the various wireless technologies for the physical layer, namely, Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), infrared wireless local area network, and Orthogonal Frequency Division Multiplexing (OFDM). Thereafter, we move over to the IEEE 802.11 physical layer specifications for these technologies.

13.1 WIRELESS LOCAL AREA NETWORK

In the past few years, there has been growing demand to have networks that permit access by wireless stations so that restriction of physical connectivity of the station to the transmission media is overcome. Wireless local area network as the name suggests is a local area network that uses wireless transmission medium. In addition to the flexibility of mobility of the stations, wireless local area network finds applications where cabling is not feasible or practical, e.g historical buildings, open spaces, *etc.* Another application of wireless local area

network is to connect the local area networks of nearby buildings using a point-to-point wireless link.

Before we go into the technology of wireless local area networks, it is necessary to get familiar with the terminology associated with their configurations and various communication modes. We will use the terminology as defined in IEEE 802.11 standard for wireless local area networks.

13.1.1 Wireless Local Area Network Configuration

Configuration of a wireless local area network is defined in terms of Basic Service Set (BSS), Extended Service Set (ESS), Access Point (AP), and Distribution System (DS).

Basic service set (BSS). A wireless local area network consists of stations that communicate with one another using wireless transmission medium (Figure 13.1). Each station has a wireless network interface for this purpose. The wireless stations are typically battery powered laptops but can be fixed workstation with wireless interface. They share the common wireless transmission medium and compete for access to the medium using the same MAC protocol. A set of such interconnected stations is called *basic service set* in IEEE 802.11 standard. A BSS, not connected to any other network, is called Independent BSS (IBSS). IBSS is also referred to as ad hoc network.

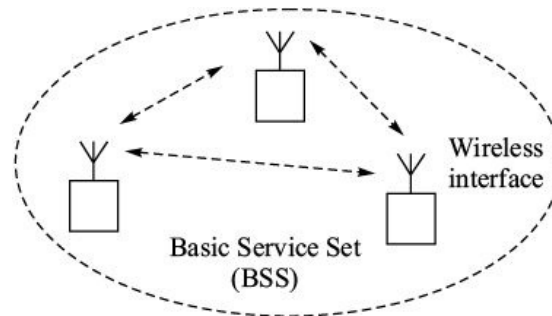


Figure 13.1 Basic service set.

Extended service set (ESS). Several basic service sets can be interconnected through a distribution system (Figure 13.2). Such a network is called *extended service set*. The ESS appears as a single logical local area network to the LLC sublayer.

Access point (AP). One of the stations of a BSS has additional functionality and interface built into it for interconnection to the Distribution System (DS). This station is called *access point*. A BSS with AP is called infrastructure BSS.

Distribution system (DS). *Distribution system* is a backbone network that interconnects the access points. It can be a wired local area network, *e.g.* an Ethernet.

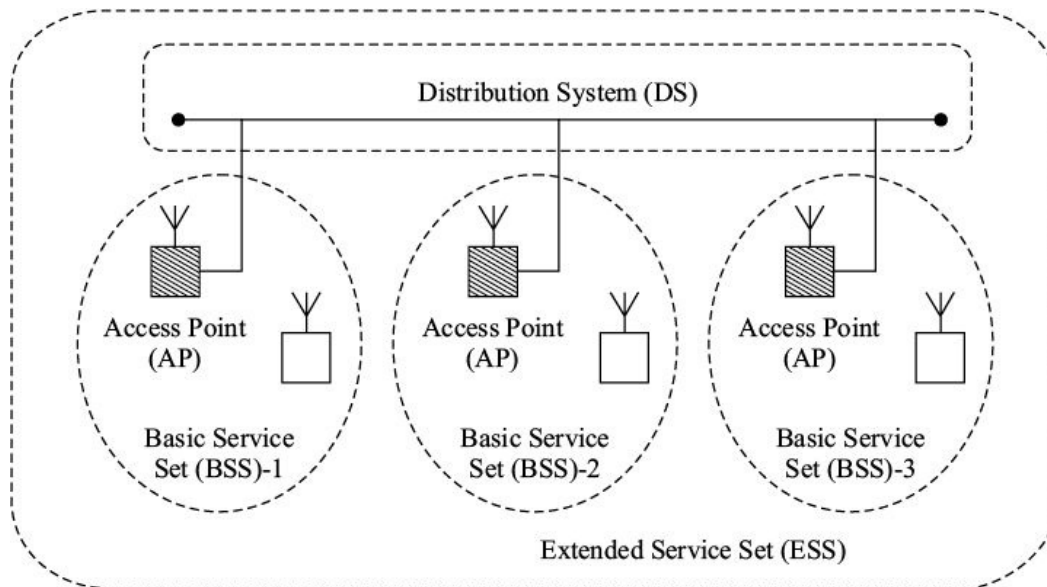


Figure 13.2 Extended service set.

13.1.2 Communication Modes

IEEE 802.11 provides two modes of communication among the wireless stations in a BSS (Figure 13.3).

Distributed coordination function (DCF). Distributed Coordination Function (DCF) permits direct any-to-any wireless communication between two stations. It is based on a contention algorithm similar to CSMA/CD used in Ethernet local area networks.

Point coordination function (PCF). Point Coordination Function (PCF) is a centralized mode of communication, wherein one station designated as Point Coordinator (PC) polls the other wireless stations in round robin fashion and gets their responses. Thus, the wireless stations communicate through the point coordinator in the PCF mode. Since a station can transmit only when it is polled, there is no contention in PCF mode.

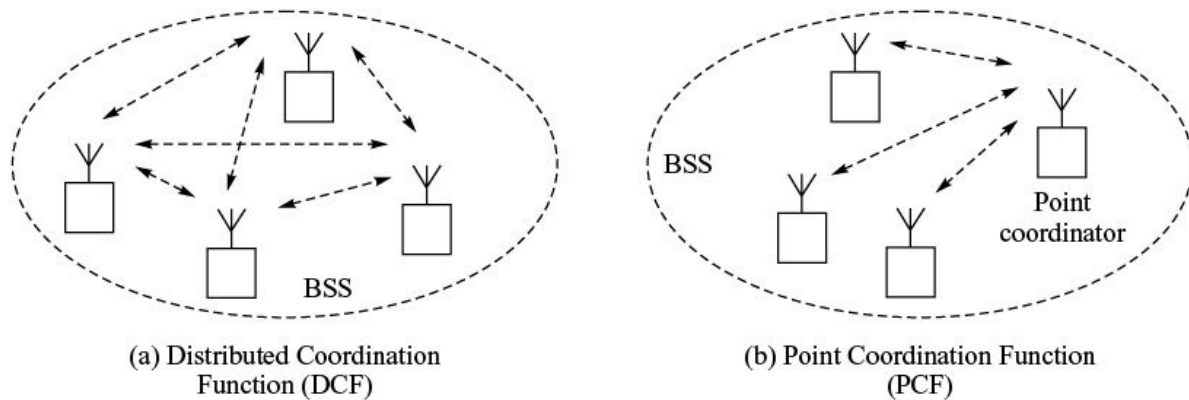


Figure 13.3 Communication modes in IEEE 802.11.

An IEEE 802.11 wireless local area network can support both the modes of communication, DCF and PCF, simultaneously. DCF mode can be used for the applications that are not time sensitive. PCF mode implements a discipline and therefore it is suitable for high-priority and time sensitive data. PCF and DCF modes are implemented on time-sharing basis as we will see later.

13.2 LAYERED ARCHITECTURE OF WIRELESS LOCAL AREA NETWORK

The layered architecture of node of a wireless local area network is shown in Figure 13.4a. It is similar to the architecture of the wired local area networks. The data link layer consists of LLC and MAC sublayers. We studied IEEE 802.2 LLC sublayer in detail in Chapter 10. The MAC sublayer and the physical layer of the wireless local area networks are specified in IEEE 802.11. For the physical layer, IEEE 802.11 specifies several alternative frequency bands and modulation schemes, which we discuss later in the chapter. Functions of IEEE 802.11 MAC sublayer of a wireless local area networks are described in the next sub-section. Figure 13.4b shows the architecture of a wireless local area network having an Ethernet local area network as the distribution system.

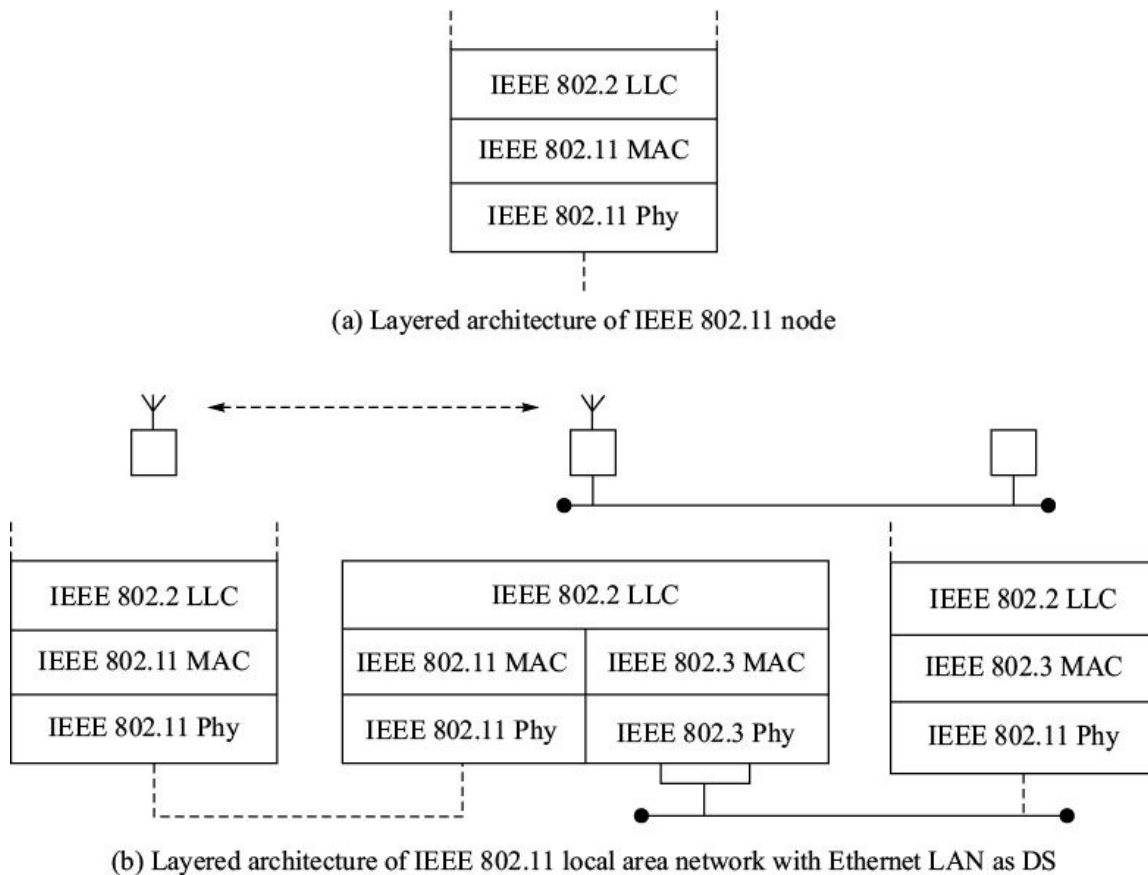


Figure 13.4 Layered architecture of IEEE 802.11 wireless local area network.

13.2.1 Functions of MAC Sublayer in IEEE 802.11

IEEE 802.11 MAC sublayer of wireless local area networks is more complex than the MAC sublayer of wired local area networks. Its basic functions include:

- Media access control
- Reliable delivery of data units
- Management functions
- Authentication and encryption.

Media access control. Media access control is based on distributed control and centralized control as mentioned in the last section. Distributed access control is called Distributed Coordination Function (DCF) and the centralized access control is called Point Coordination Function (PCF). DCF is based on contention access and PCF is contention free.

Reliable delivery of data units. Wireless transmission is not much reliable due to several factors. Therefore, IEEE 802.11 MAC sublayer implements an

acknowledgement mechanism. All MAC frames are individually acknowledged. In addition to the acknowledgement mechanism, an alert mechanism based on Request to Send (RTS) and Clear to Send (CTS) is implemented between transmitting and receiving stations. By exchanging RTS and CTS frames before sending a data frame, it is ensured that the receiving station is ready to accept the data frame, and that other stations do not interrupt their communication.

Management functions. Wireless local area network provides mobility of stations from one BSS to another and from one ESS to another. MAC layer implements the following functions in this regard:

Association. This function relates to establishing initial association between a station and the AP of a BSS. Once the association is established with a station, the AP can communicate this information to other APs of the ESS to facilitate delivery of frames addressed to the station.

Reassociation. This function enables movement of a station from one BSS to another.

Disassociation. This function enables termination of an existing association between a station and an AP.

Authentication and encryption. Since there is no physical connectivity of the stations in a wireless local area network, it is prone to unauthorized access and eavesdropping. While establishing association, a wireless station is required to authenticate its identity. IEEE 802.11 supports several authentication schemes. Also, there is possibility of a third station copying communication between two stations. Therefore, the messages exchanged between two stations may require encryption.

We will describe the medium access control and reliability methods in detail in this chapter. Management and authentication methods of IEEE 802.11 are beyond the scope of this text.

13.3 MEDIA ACCESS CONTROL IN WIRELESS LOCAL AREA NETWORK

Wireless transmission differs from wireline transmission in many ways. Before we go into media access control methods of wireless local area networks, we need to examine the unique features of wireless transmission and their implications on the working of a network.

Unbounded coverage area. There is no well-defined physical boundary of the transmission medium in wireless transmission. The signal strength of transmission reduces in inverse proportion of square of distance between the transmitter and the receiver. There is minimum received signal strength below which the error rate increases sharply. Thus it is possible that a receiving station does not detect the transmitted signals beyond a certain distance from the transmitting station. In other words, each transmitting station has a coverage area around it (Figure 13.5). The coverage area is determined by the transmitted power and the receiver sensitivity (minimum required signal strength for a given error rate).

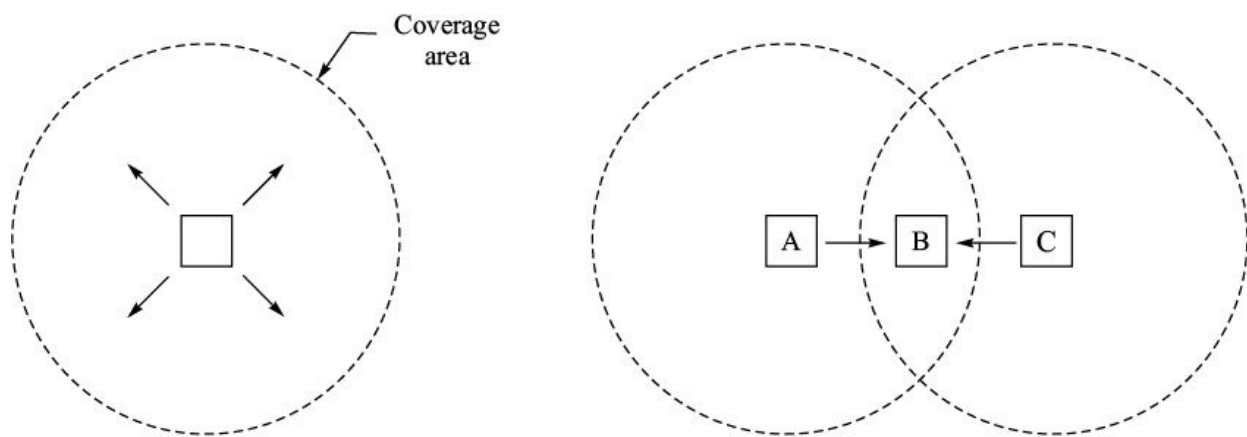


Figure 13.5 Hidden station.

Hidden station problem. Limited coverage area results hidden station problem. The dotted circles in Figure 13.5 represent the coverage area of transmissions from station A and C. Note that A and C do not receive transmission from one another and are therefore hidden from one another. When A is transmitting to B, C assumes that the medium is free and if it also transmits, there will be a collision. The same thing happens when C is transmitting to B and A wrongly concludes that the medium is free.

Exposed station. Partially overlapping coverage areas result in hidden station problem when the receiver in the overlapping area hears two transmissions simultaneously. It is possible to have multiple simultaneous transmissions if the receivers are in the non-overlapping areas. For example in Figure 13.6, when B transmits to A, D does not hear this transmission because it is outside the coverage area of the station B. If C transmits to station D when transmission from B to A is ongoing, there will not be any collision. Thus, C can transmit to D simultaneously but it does not actually transmit as it detects transmission of B.

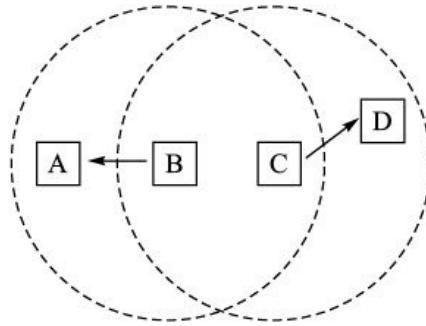


Figure 13.6 Exposed station.

Reliability issues. A wireless local area network usually operates in a confined space where direct and reflected signals reach the receiver with differential delays (Figure 13.7). Multiple overlapping signals cause distortion and may result in errors. Therefore, a station having transmitted a frame, does not really know whether the frame was received correctly by the intended receiver.

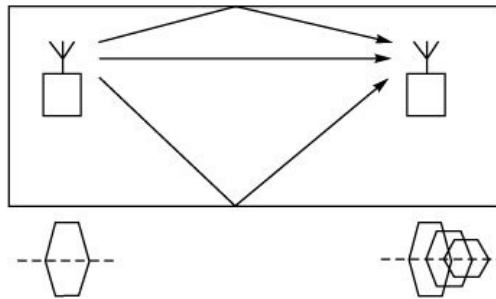


Figure 13.7 Multi-path propagation in confined space.

To increase the reliability of communication, wireless local area networks implement an acknowledgement mechanism at the MAC sublayer. Every data frame is acknowledged by the receiver. If the sender does not receive an acknowledgement before timeout or receives the acknowledgement with errors, it takes necessary steps to retransmit the data frame.

The exchange of a data frame and its acknowledgement between two stations is an atomic operation not to be interrupted by any other transmission from any station. This discipline is ensured by allowing the acknowledgement to be sent after a shorter time gap than that required for sending a data frame (Figure 13.8). The time gap between adjacent frames is called Inter-Frame Space (IFS) in IEEE 802.11. Acknowledgement frames are sent after the shortest time gap called Short IFS (SIFS) as explained in the next section.

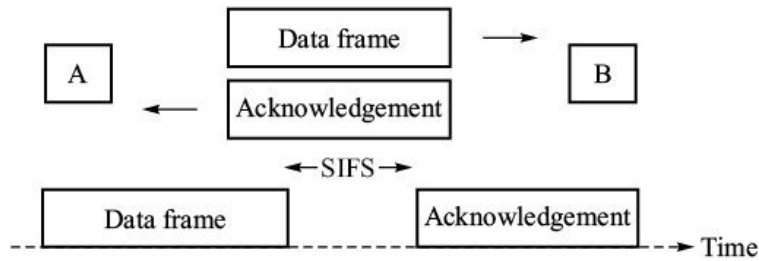


Figure 13.8 Acknowledgement for reliable communication in IEEE 802.11.

13.3.1 Inter-Frame Spaces in IEEE 802.11

In wireless local area networks, MAC frame transmissions are separated by a time gap called Inter-Frame Space (IFS). Three different types of IFSs are defined in IEEE 802.11 (Figure 13.9). Different IFSs enable establishment of a priority mechanism for transmission of various types of frames.

SIFS (Short IFS). It is the shortest IFS used for the highest priority frames such as acknowledgement frame, CTS frame, poll response. It is also used for sending fragments of same data unit.

PIFS (PCF-IFS). PCF-IFS as the name suggests is used by the PCF for polling. After the transmission medium becomes free, the point coordinator can issue a poll after inter-frame gap equal to PIFS. PIFS equals SIFS plus one slot time.¹

DIFS (Distributed-IFS). DIFS is used by DCF for transmitting data frames. It is equal to SIFS plus two slot-times and is the longest inter-frame gap.

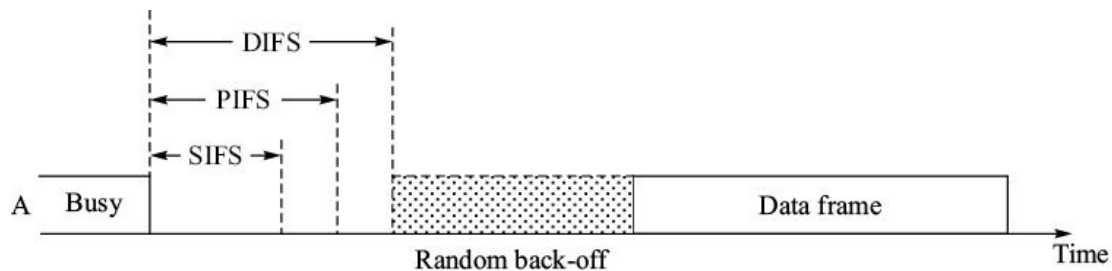


Figure 13.9 Inter-frame spaces in IEEE 802.11.

Typical values of SIFS, PIFS, and DIFS are 10 ms, 30 ms, and 50 ms respectively. It is obvious that the response frames (CTS, ACK, and poll response) get the first priority as SIFS expires first. Thus, if a station has received RTS or data frame or poll addressed to it, it will be able to send the corresponding response. The next priority goes for round robin poll if PCF is implemented. The point coordinator waits for time period equal to PIFS after the transmission media becomes free and sends the poll. The data frames that are

sent under DCF get the last priority as DIFS expires last.

The random back-off shown in Figure 13.9, is applicable to DIFS only. It is applied in DCF mode as there may be several stations contending to transmit their data frames. We will learn more about the random back-off later.

13.4 DISTRIBUTED COORDINATION FUNCTION (DCF)

CSMA/CD, used in the Ethernet local area networks, is based on the following principles:

Carrier sense. Check presence of signal on the transmission medium.

Transmit. If the transmission medium is free, send the frame. If not, continue sensing the medium and as soon as the medium becomes free, transmit the frame.

Collision detect. During transmission of the frame continue monitoring the medium for any other transmission on the medium. If a collision occurs, abort the frame and delay retransmission of frame using an exponential back-off algorithm.

Note that collision is allowed to occur in CSMA/CD before exponential back-off algorithm kicks off. We cannot adopt the above CSMA/CD scheme in wireless local area networks on several accounts:

1. In wired local area networks, all transmissions reach every station and therefore 'carrier sensing' is possible. On the other hand, it is possible to have 'hidden stations' in wireless local area networks. If C is hidden from station A, the transmission from C is never detected by A. Thus the fact that the transmitting station (A) senses the medium free does not necessarily mean the medium is free around the intended receiver (B). The receiver (B) may be receiving transmission from station (C).
2. The stations use the same frequency for transmission. It is not possible for a station, while it is transmitting a frame, to simultaneously monitor the transmissions from other stations at the same frequency. Its own transmission overwhelms its receiver. Therefore, a station while transmitting a frame cannot detect when a collision should it happen.
3. In wireless local area networks, exchange of a data frame and its acknowledgement is one atomic operation (Figure 13.8). This operation cannot be interrupted by transmission from a third station even though the

medium becomes free for a short time interval before the acknowledgement is transmitted.

A safer contention access approach called CSMA/Collision Avoidance or, simply CSMA/CA is adopted for wireless local area networks. IEEE 802.11 has adopted CSMA/CA scheme for wireless local area networks and calls this approach Distributed Coordination Function (DCF). DCF works as follows (Figure 13.10):

- a. A station checks twice whether the transmission media is free. The second check is made after an interval greater than SIFS. This interval is called DIFS and it was introduced in the last section. The transmission medium should be free on both these instances for a station to transmit a frame immediately after the second check.
- b. If the medium is found busy in either of the above checks, the station continues monitoring the medium. As soon as it becomes free, it waits for interval DIFS and checks the transmission medium again. If it is free, the station backs off for time interval as determined by exponential back-off algorithm. Note that unlike CSMA/CD, the back-off algorithm is initiated even though collision has not occurred.

A station continues monitoring the transmission medium during the back-off interval. If the medium remains free, the station transmits its frame. If during the back-off interval, another transmission is detected, the station halts its back-off timer and repeats step (b) above. The remaining back-off interval is used when it gets next opportunity to transmit. This ensures that the station gets higher priority in the next round (Figure 13.11).

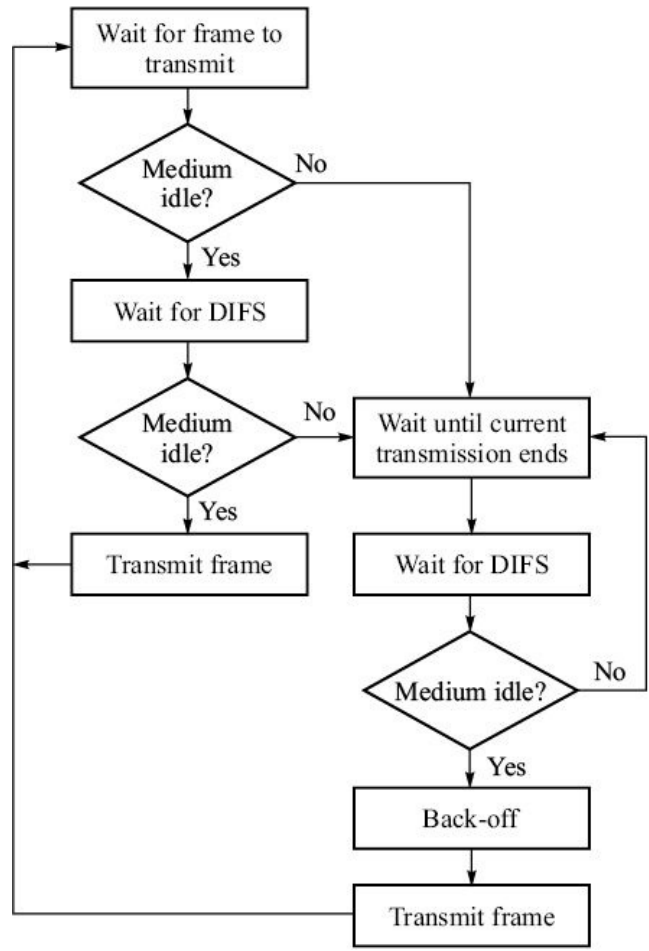


Figure 13.10 CSAMA/collision avoidance (CA) in IEEE 802.11.

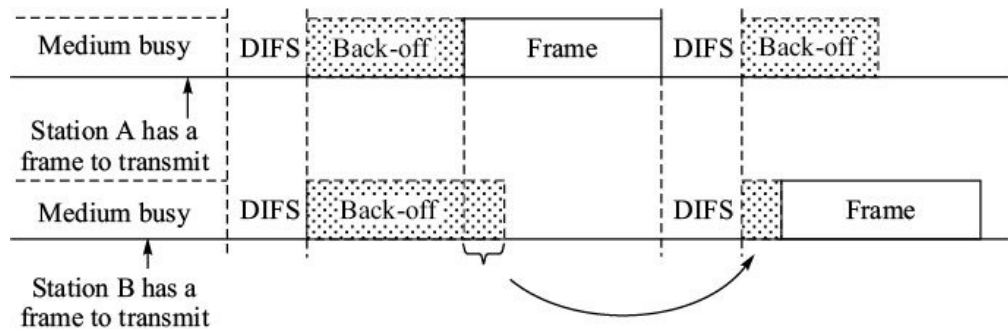


Figure 13.11 Back-off carry over.

13.4.1 DCF with RTS/CTS

To further refine the collision avoidance in wireless networks, IEEE 802.11 specifies an option involving four-frame atomic operation. All the stations in the coverage areas of the sender and receiver are alerted in advance that a frame is going to be sent and they should withhold their transmissions for a specified time. Request to Send (RTS) and Clear to Send (CTS) frames are used for this

purpose (Figure 13.12). These frames have a duration field that indicates the time for which the wireless medium is required by the two stations for their impending exchange of one data frame and its acknowledgement.² In fact, duration field is present in most of the frames used in wireless local area networks.

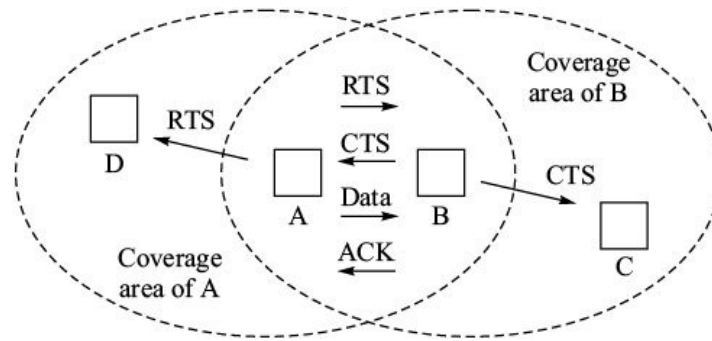


Figure 13.12 Four frame atomic exchange.

Each station has a Network Allocation Vector (NAV) timer that stores the duration field of the received frames. Thus, NAV timer at a station indicates the time for which transmission medium is reserved. A station waits till expiry of the NAV timer. It can thereafter start the process for transmitting its frame. Four-frame atomic exchange operates as under (Figure 13.12):

1. Station A sends RTS to B indicating its intent to transmit a data frame. RTS is received by all the stations (e.g. B and D) in the coverage area of A.
2. On receipt of RTS, the stations in the coverage area of A set their NAV timer to the duration specified in the RTS frame and start down counting. They defer their attempts to transmit till NAV becomes zero.
3. Station B responds to the RTS with a CTS frame after a time gap equal to SIFS. CTS is received by all the stations in the coverage area of B. Even the stations hidden from A (e.g., C) receive CTS. The stations in the coverage area of B set their NAV timer to the duration indicated in the CTS frame and defer their transmissions till the NAV becomes zero. Thus, hidden station problem is taken care of.
4. On receipt of CTS, A sends its data frame to B after an interval equal to SIFS. There is no possibility of any collision for this frame.
5. On receipt of the data frame from A, B responds with acknowledgement after an interval equal to SIFS. There is no possibility of any collision for the acknowledgement frame.

Figure 13.13 shows the complete picture of DCF using RTS/CTS. Stations A, B, C, and D are located as shown in Figure 13.12. Stations C and D defer their attempts to transmit after they receive CTS and RTS respectively with the NAV time durations indicated in these frames.

Note that RTS and CTS frames can collide with RTS/CTS frames from other stations. These frames are kept short to minimize the lost time. RTS/CTS is not a mandatory feature of

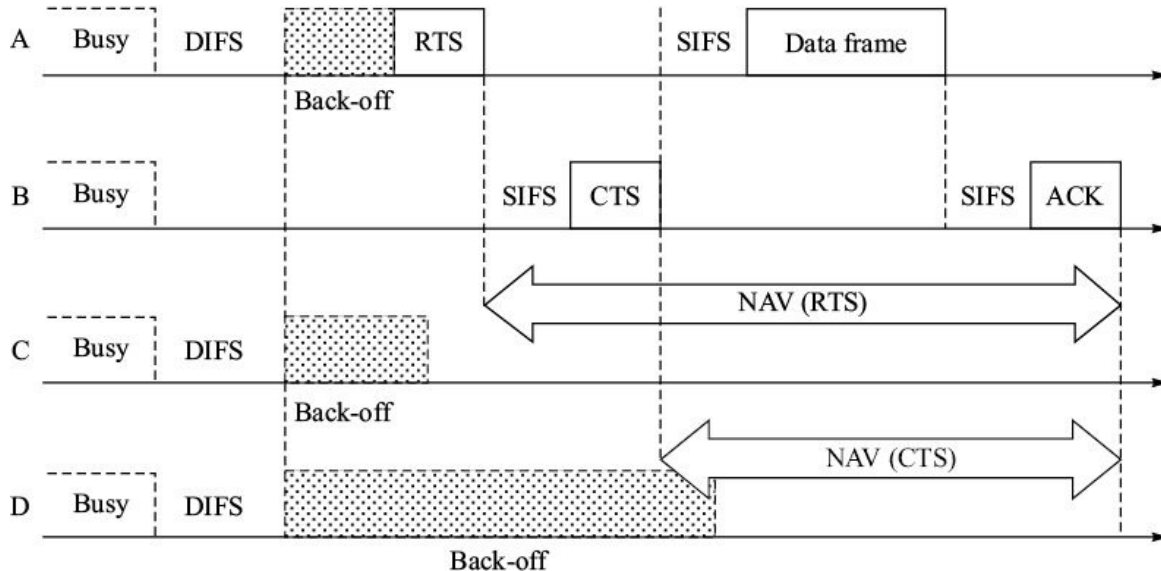


Figure 13.13 DCF with RTS/CTS.

IEEE 802.11. It improves efficiency in terms of reduced collisions. But it results in increased delays.

13.4.2 Binary Exponential Back-off

After expiry of DIFS, the stations waiting to transmit data frames set their back-off timer using binary exponential back-off algorithm that we studied in Chapter 11. Binary exponential back-off algorithm achieves congestion control by dynamically choosing the contention window. It works as follows:

1. Each station draws a random number n from range $(0, C)$, and backs off for duration n slot-time. C is called contention window. Its value depends on the physical layer. For example, $C = 32$ for Direct Sequence Spread Spectrum (DSSS) physical layer.
2. After expiry of the back-off interval, a station transmits its data frame if the medium is free.

3. It is possible that more than one stations get same back-off and therefore collision can take place. If a station does not get acknowledgement for its transmitted data frame within a specified time interval, it assumes that the frame is lost and doubles the value of C for the next retry. Larger value of C implies that n can have larger value and therefore back-off interval can be larger in the next retry.
4. Steps (1) and (2) above are repeated on every occasion of retry. If there are large number of stations trying to access the transmission medium, collisions take place and back-off increases exponentially.
5. A maximum upper limit for C is defined. For DSSS physical layer, the maximum value of C is 1024. When C reaches this value, it remains constant for further retries. There is an upper limit on number of retries, after which the frame is discarded.
6. C is set to its original minimum value when transmission of a data frame is successful as confirmed by receipt of its acknowledgement or when the frame is discarded after maximum retries.

13.4.3 Fragmentation

Fragmentation can be carried out at MAC sublayer. Fragments of a data unit at MAC sublayer are given priority so that all the fragments reach the receiver without any interruption and the receiver is able to reconstruct the whole data unit. Like RTS/CTS frames, the fragments and their acknowledgements also carry the duration field, which other stations use for setting their NAV values. Fragments and their acknowledgements have the inter-fragment-frame gap equal to SIFS and there is no back-off for these frames (Figure 13.14).

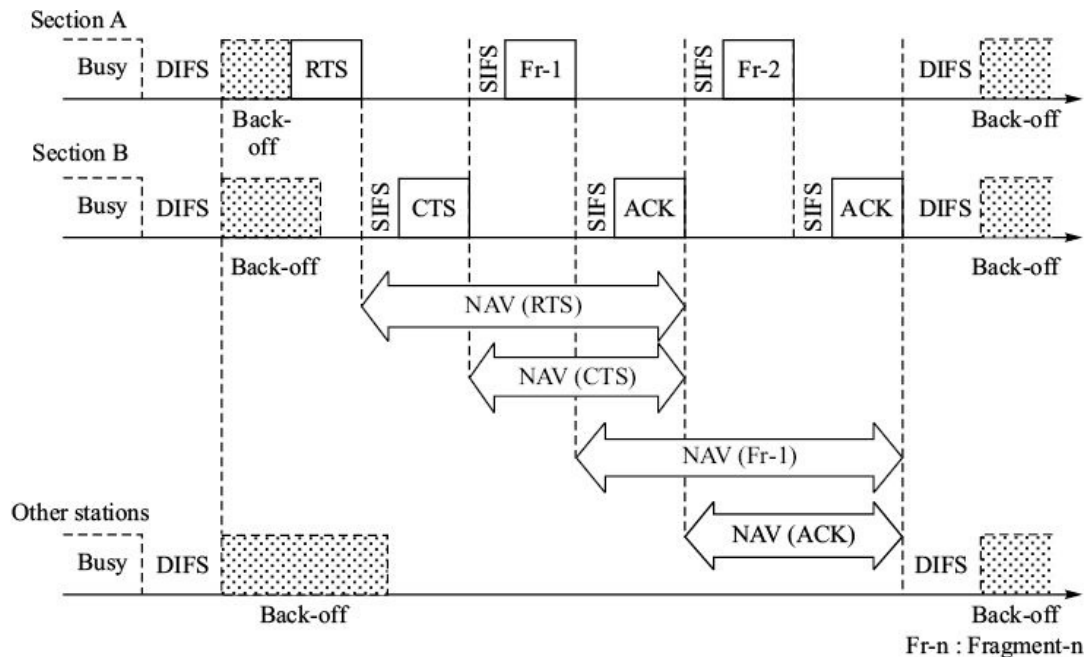


Figure 13.14 Fragmentation.

The MAC frame of each fragment contains More Fragment (MF) bit which is set to 1 to indicate to the receiver that another fragment is to follow. Sequence Control (SC) field of the MAC frame contains fragment sequence number and frame sequence number to facilitate reassembly of the frame.

13.5 POINT COORDINATION FUNCTION (PCF)

Unlike DCF which based on distributed control, Point Coordination Function (PCF) a centralized communication control process. It provides contention free access to the transmission media. The station designated as Point Coordinator (PC) polls other stations for sending and receiving data frames in a round robin method. PCF is required for high priority and time-sensitive data.

For DCF and PCF to coexist in a BSS simultaneously, Contention Free Period (CFP) and Contention Period (CP) are defined (Figure 13.15). CFP and CP alternate in a cyclic manner. One cycle of CFP and CP has a defined nominal length and is called *superframe*.

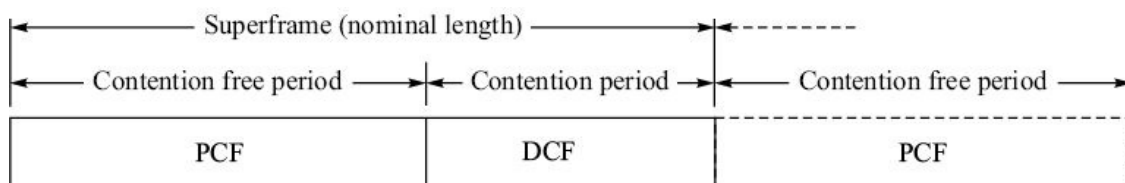


Figure 13.15 Superframe.

Communication is controlled by the point coordinator using PCF during contention free period. The PC sends a beacon signal followed by a round robin poll of the stations covered under PCF. Some of the polled stations may not have data to send. Therefore, CFP may end sooner. After the point coordinator has finished its one complete round of polls, it sends a CF-End frame to signal end of contention free period.

Contention period starts immediately after the contention free period (Figure 13.16). During contention period, communication is carried out using DCF as described earlier. Note that the boundaries of CFP and CP are not rigid. It is possible that a transmission under DCF may still be ongoing at the defined boundary of the superframe. The point coordinator waits till end of current DCF transmission. Thereafter, it defers its transmission of its first frame by PIFS. PIFS being less than DIFS, the point coordinator overtakes other stations which need to defer their transmission by DIFS. The super frame, however, in this process gets fore-shortened.

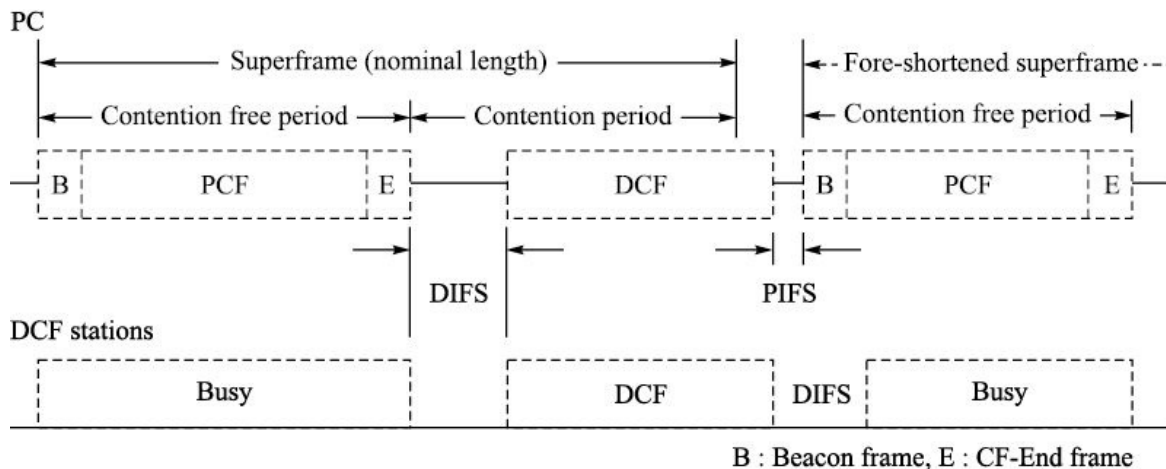


Figure 13.16 Alternation of CFP and CP.

13.5.1 Communication during Contention Free Period (CFP) Using PCF

Figure 13.17 illustrates a typical exchange of frames between the point coordinator and the other stations during contention free period. Communication between point coordinator and other stations during contention free period takes place using various types of frames. We describe these frames in the next section. The entire exchange of frames during contention free period takes place with inter-frame gap equal to SIFS so that other stations may not be able to grab the transmission medium.

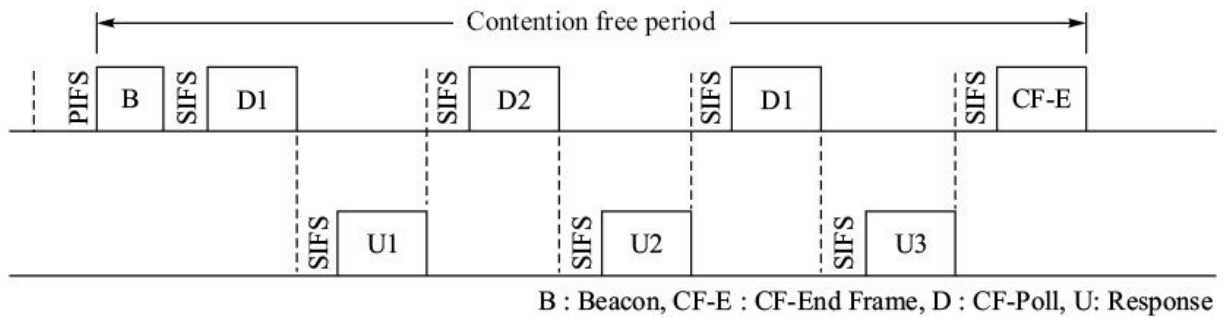


Figure 13.17 Point coordination function.

- Contention free period begins with a beacon frame sent by the point coordinator.
- The point coordinator polls the stations one by one sending a poll frame. The poll frame may also contain data from the point coordinator.
- The polled station responds with a data frame or acknowledgement frame or a combination (Data + acknowledgement) frame.
- The point coordinator ends contention free period with CF-End frame. CF-End frame may also contain the acknowledgement for the last received frame.

13.6 MAC FRAMES OF THE IEEE 802.11

There are three types of the MAC frames in IEEE 802.11:

- Control frames
- Data frames
- Management frames.

Management frames are several and are used to manage communication between stations and APs. The functions include association, re-association, disassociation, beacon and authentication. We will not go into details of these management functions. Control and data frames are described below.

13.6.1 Control Frames

Control frames assist in reliable delivery of data frames. There are six sub-types of control frames.

RTS (Request to send) frame. This is the first frame of the four-frame exchange discussed earlier. It alerts all the stations in the coverage area of its

source indicating the duration for which the transmission medium will remain engaged. It is also the request to the destination station for a response confirming its readiness to accept the frame.

CTS (Clear to send) frame. This is the second frame of the four-frame exchange discussed earlier. It is sent as response to RTS frame by the destination station of RTS. It also alerts all the stations in its coverage area indicating the duration for which the transmission medium will remain engaged.

ACK (Acknowledgement) frame. This frame is sent by the destination station as acknowledgement of having received a data frame, management frame or PS-Poll frame.

CF-end frame. This frame is sent by the point coordinator to announce the end of the contention free period.

Ack + CF-end frame. This frame is sent by the point coordinator to announce the end of the contention free period and to acknowledge the received data.

PS-Poll (Power save poll). This frame is sent by a station to request the point coordinator to transmit the frame that has been buffered for the station while it was in power saving mode.

13.6.2 Data Frames

These frames carry user data. There are eight sub-types of the data frames:

Data frame. This frame is used by a station to send data.

Data + CF-Ack frame. This frame is used by the polled station to send data and to acknowledge the received data.

Data + CF-poll frame. This frame is used by the point-coordinator to send data to a station, inviting the polled station to respond.

Data + CF-Ack + CF-poll frame. This frame is used by the point-coordinator to send data to a station, inviting the polled station to respond. This frame also acknowledges the received data.

CF-Ack frame. This frame is used by the polled station to acknowledge the received data.

CF-poll frame. This frame is used by the point-coordinator to poll a station inviting the station to send its data.

CF-Ack + CF-poll frame. This frame is used by the point-coordinator to poll a station inviting the station to send its data. This frame also acknowledges the received data.

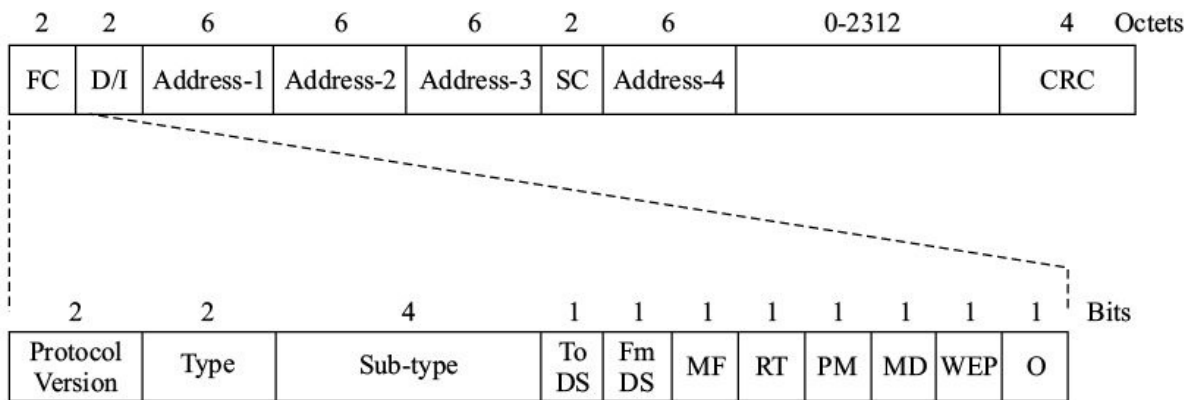
Null function frame. The null function frame is used for carrying power management field. It does not carry data, acknowledgement or poll.

CF-end frame. This frame is sent by the point-coordinator to announce the end of the contention free period.

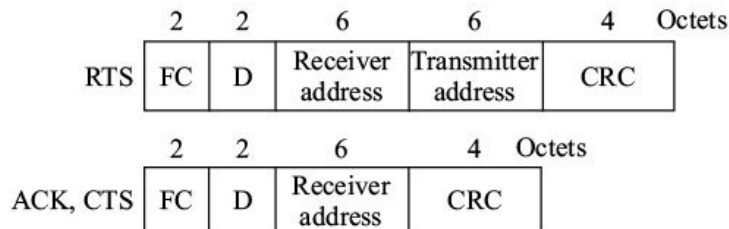
CF-Ack + CF-end frame. This frame is sent by the point coordinator to announce the end of the contention free period and to acknowledge the received data.

13.6.3 Format of MAC Frames of IEEE 802.11

Figure 13.18a shows the general format of the control and data of IEEE 802.11 frames. Formats of RTS, CTS and ACK frames have reduced number of fields as shown in Figure 13.18b.



(a) General frame format



FC : Frame Control D/I : Duration/Connection Id SC : Sequence Control MF : More Fragments
 RT : Retry PM : Power Management MD : More Data
 O : Order WEP : Wired Equivalent Privacy

(b) Formats of RTS, CTS, ACK frames

Figure 13.18 Format of IEEE 802.11 frame.

Frame control (FC, 2 octets). It indicates the type of frame. Some of its important subfields are as under:

Protocol version (2 bits). This field indicates the version of IEEE 802.11 protocol being used.

Type (2 bits). This field indicates the type of frame (control, data, management).

Sub-type (4 bits). This field indicates the sub-type of the frame.

To DS (1 bit). This bit is set to 1 if the frame is destined for DS. Use of ‘To DS’ bit is indicated in Table 13.1.

From DS (1 bit). This bit is set 1 if the frame is coming from DS. Use of ‘From DS’ bit is indicated in Table 13.1.

TABLE 13.1 Address Fields of IEEE 802.11 MAC Frame						
	To DS	From DS	Address 1	Address 2	Address 3	Address 4
Independent BSS	0	0	DA	SA	BSS-Id	—
From AP	0	1	DA	BSS-Id	SA	—
To AP	1	0	BSS-Id	SA	DA	—
Within DS	1	1	RA	TA	DA	—
						SA

MF (More fragment, 1 bit). This bit is set to 1 if more fragments are to follow.

RT (Retry, 1 bit). This bit is set to 1 if this frame is retransmission of previous frame.

PM (Power management, 1 bit). This bit is set to 1 if the transmitting station is in sleep mode.

MD (More data, 1 bit). This bit when set 1 indicates that the transmitting station has more data to send.

WEP (Wired equivalent privacy, 1 bit). This bit is 1 if the wired equivalent privacy protocol is implemented.

Order (1 bit). If the service provided by the MAC sublayer is ‘Strictly Ordered’ service, this bit is set to 1.

D/I (Duration/connection Id, 2 octets). As duration field, it indicates the time in microseconds, the channel is reserved for reliable transmission of a MAC frame and its acknowledgement. As connection-id field, it identifies an association or a connection.

Address fields (6 octets each). There can be up to four address fields. Their number and use depend on the context (Table 13.1). DA and SA are the destination address and source address respectively. The remaining terminology is as follows:

BSS-Id : BSS Identifier

RA : Receiver Address

TA : Transmitter Address

RA and TA refer to addresses of APs within the Distribution System (DS).

Sequence control (SC, 2 octets). It contains 4-bit fragment number subfield which is used for fragmentation and reassembly. The other 12-bit subfield is the sequence number of the frame sent between a given pair of transmitter and receiver.

CRC. It is 32-bit frame CRC sequence for detection of errors.

13.7 TRANSMISSION TECHNOLOGIES OF IEEE 802.11

Wireless local area networks use special transmission coding schemes such spread spectrum (DSSS, FHSS), Pulse Position Modulation (PPM), Orthogonal Frequency Division Multiplexing (OFDM) to make efficient use of the available bandwidth and to work in interference-prone environment. We will examine each of these schemes briefly before going into the description of the physical layer of IEEE 802.11.

13.7.1 Spread Spectrum Transmission Systems

Spread spectrum techniques were originally developed for military applications to counter jamming and unauthorized detection of wireless signals. In spread spectrum techniques, the information signal is spread over wider bandwidth at the transmitter. Bandwidth spreading is done using a Pseudo Random Binary Sequence (PRBS) called spreading sequence (Figure 13.19). At the receiver, the spreading sequence is used again to restore the information signal. It is necessary, however, that the spreading sequences at the transmitter and the receiver are same and synchronized.

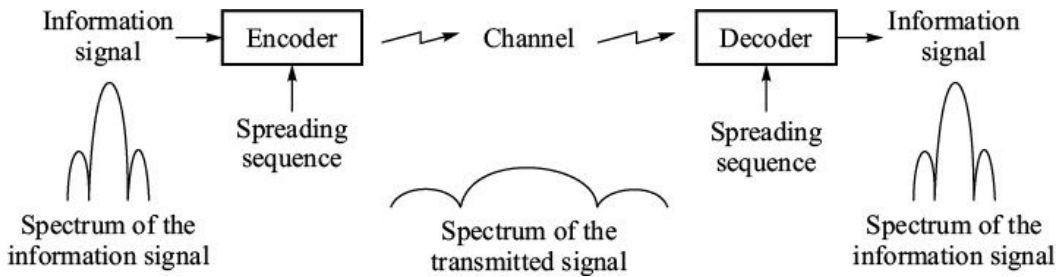


Figure 13.19 Spread spectrum.

The spreading sequence at the receiver compacts the transmitted signal but it spreads the interference and thus reducing its impact (Figure 13.20). The interfering signal can even be a

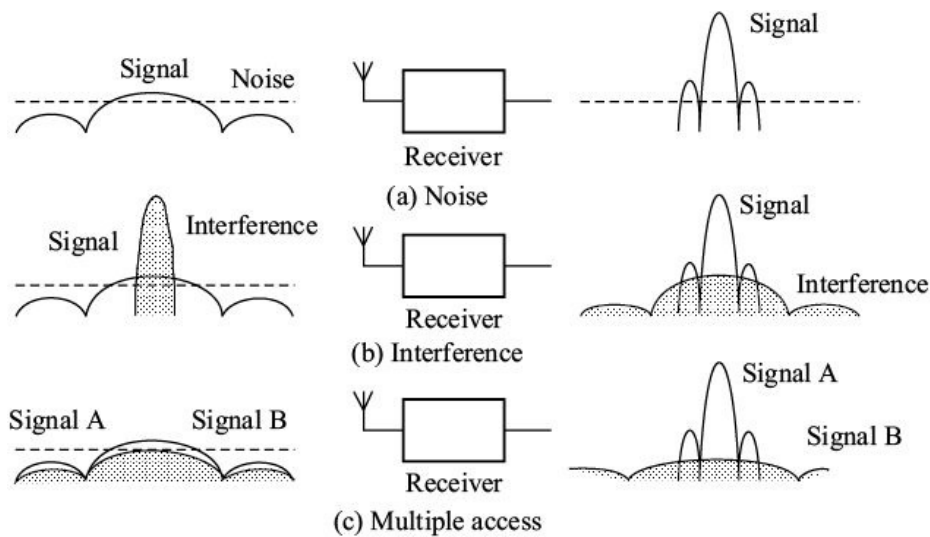


Figure 13.20 Interference suppression in DSSS.

simultaneous transmission from another source at the same frequency. Thus, the spread spectrum techniques can be effectively utilized for the multiple access required in the local area networks.

There are two basic spread spectrum techniques:

- Direct Sequence Spread Spectrum (DSSS)
- Frequency Hopping Spread Spectrum (FHSS).

Direct sequence spread spectrum (DSSS). In Direct Sequence Spread Spectrum (DSSS), each data bit is represented by multiple bits in the transmitted signal. The original signal therefore spreads across a wider frequency band in proportion to the number of bits used to represent one data bit. The data bits are exclusive-ORed with a spreading sequence which is Pseudo Random Binary

Sequence (PRBS). The spreading sequence has much higher bit rate than the data bits. In Figure 13.21, 7-bit spreading sequence 1110010 is used. Note that for each data bit 7 bits are transmitted. Thus, the bandwidth of the transmitted signal expands to seven times.

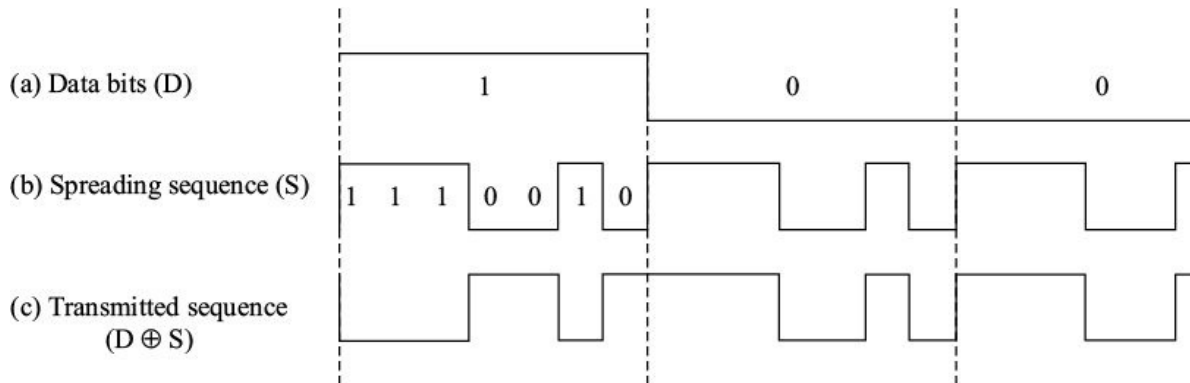


Figure 13.21 Direct sequence spread spectrum at the transmitter.

At the receiving end, the received sequence is again exclusive-ORed with the spreading sequence to get the data bits (Figure 13.22). Note that we need at the receiver spreading sequence which is synchronized with data bit boundaries in addition to being clock-synchronous. A preamble is attached to the data frame to achieve this.

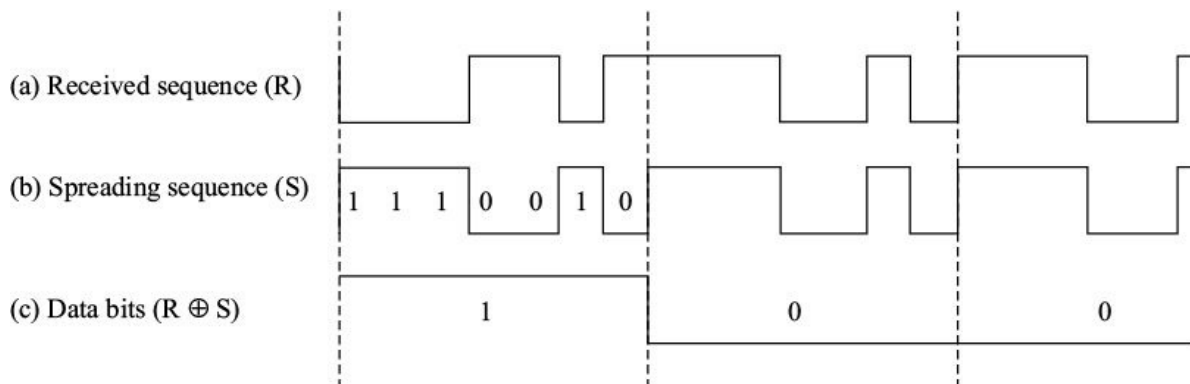


Figure 13.22 Direct sequence spread spectrum at the receiver.

One bit of the spreading sequence is called a chip. Thus, chipping rate is the bit rate of the spreading sequence. The ratio of chipping rate and data rate expressed in decibels is referred to as processing gain.

Frequency hopping spread spectrum (FHSS). In Frequency Hopping Spread Spectrum (FHSS), the transmitted carrier frequency is changed randomly at fixed intervals. In Figure 13.23a, the message signal modulates the carrier which is at f_1 to start with. After a fixed interval T_C , the carrier frequency changes to f_7

and then to f_5 and so on. The receiver is synchronized to the transmitter. It picks up the transmitted carrier frequency to retrieve the information signal by demodulating it.

Figure 13.23b shows the basic block schematic of the FHSS transmitter and receiver. At the transmitter, the data signal modulates the IF (Intermediate Frequency) using FSK/PSK. The modulated IF is up-converted to RF (Radio frequency) carrier using a frequency synthesizer. The frequency synthesizer is driven by a PRBS so that the RF is hopped randomly. At the receiver, the received RF is down-converted the IF using local frequency synthesizer that is driven by the same PRBS synchronized to the PRBS of the transmitter. The IF is demodulated to get back the data signal.

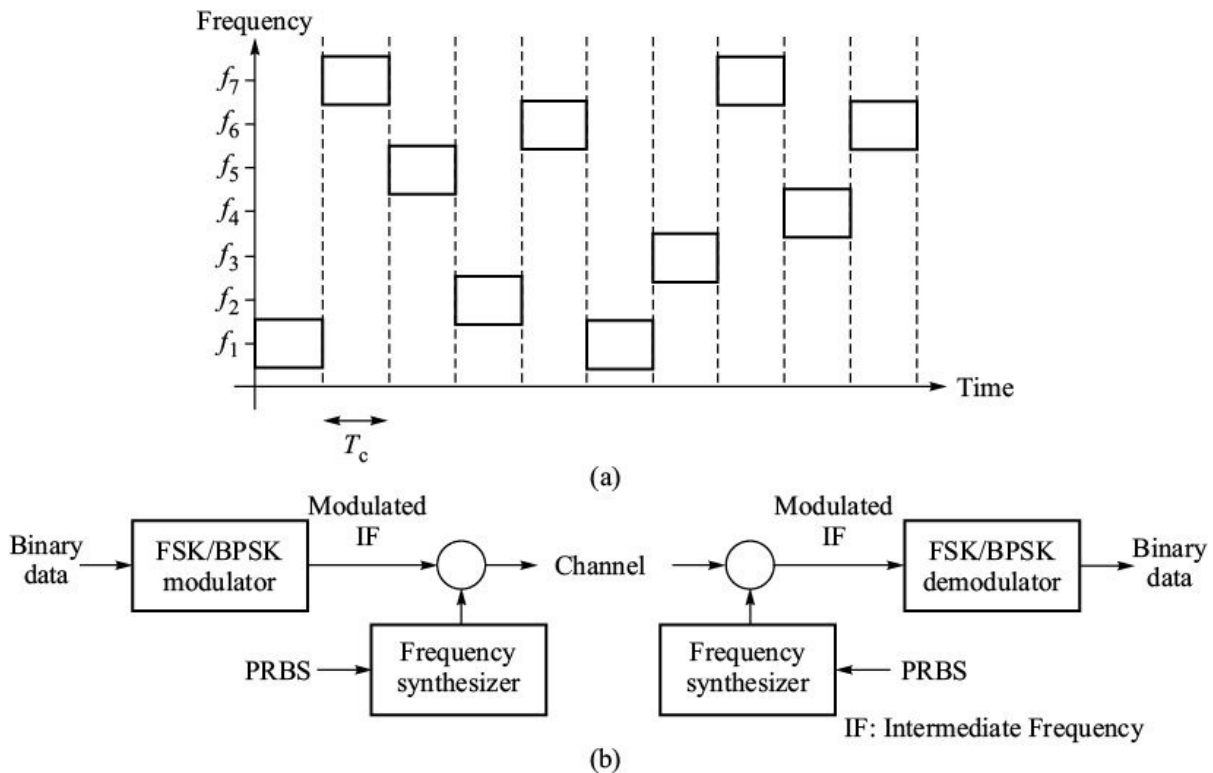


Figure 13.23 Frequency hopping spread spectrum.

If the hopping rate ($=1/T_c$) is greater than the baud rate³ (modulation rate), we call it as the case of fast frequency-hopping spread spectrum. If the hopping rate is less than the baud rate, we refer it to as the case of slow frequency hopping spread spectrum.

In Figure 13.24, the two-state FSK modulation is used. Thus each symbol contains one bit. The higher frequency represents binary 1 and the lower frequency represents binary 0. Figure 13.24a, illustrates example of slow

frequency hopping spread spectrum. Each bit occupies two frequency hops. Figure 13.24b illustrates an example of fast frequency hopping spread spectrum. In this case, each frequency hop contains two bits.

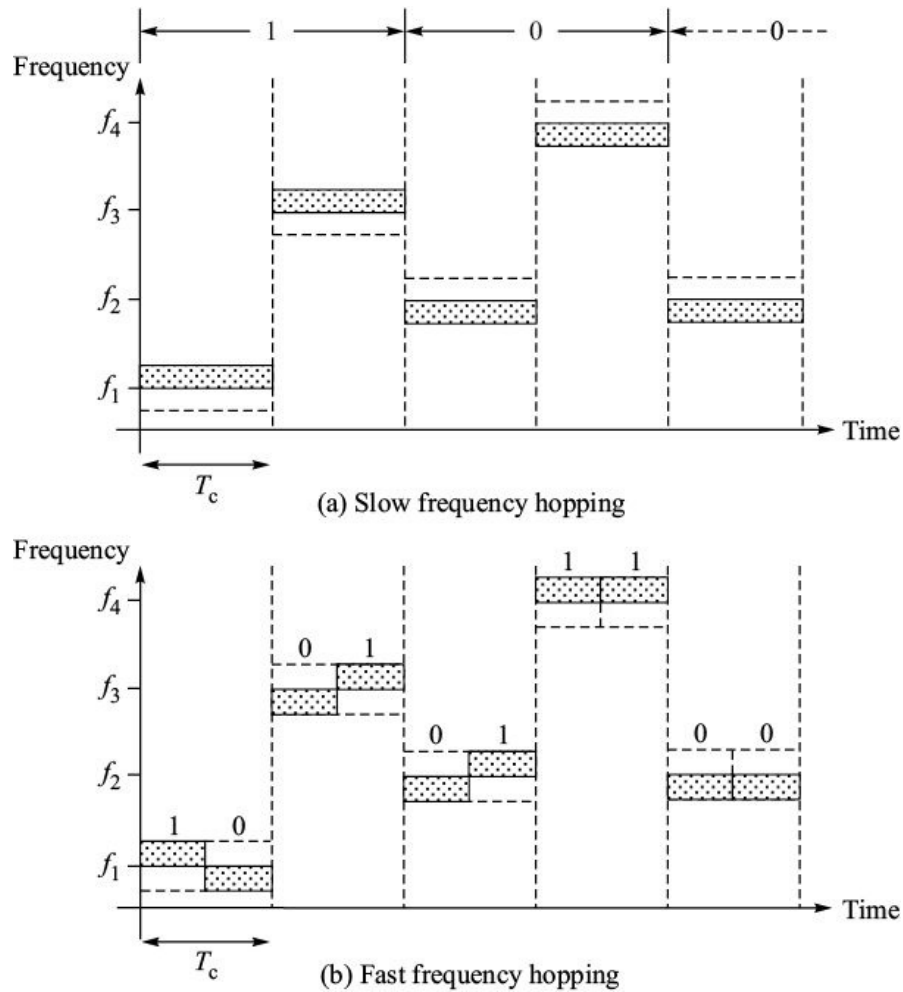


Figure 13.24 Slow and fast frequency hopping.

13.7.2 Infrared Transmission Systems

Infrared local area networks work on infrared signals of wave length in the range 800–950 nm. The infrared (IR) signal is modulated by the data bits using intensity modulation. Since infrared signals cannot penetrate the walls, the coverage area is confined to within walls of a room. To provide connectivity from a station to every other station on the local area network, the infrared beam from the station is optically diffused so that the infrared signal is spread over a wide angular area. This is called diffused mode of operation. Alternative is to point the infrared beam to an active or passive reflector on the roof, which dispersively reflects the beam in all directions (Figure 13.25). Active reflector is

basically a multiport wireless repeater operating at infrared wavelength.

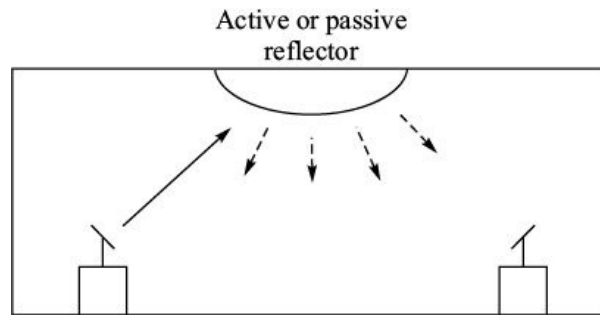


Figure 13.25 Infrared signal dispersion using reflector.

The advantages of infrared local area networks are as follows:

- Infrared spectrum is unregulated worldwide and is readily available for use. On the other hand, availability of radio frequencies in the gigahertz range used in spread spectrum and microwave local area networks is limited and regulated.
- IR technology uses intensity modulation and is relatively inexpensive.
- IR beam is readily reflected/diffused by the physical objects. Thus it is easily spread in a room. Separate IR installation can be operated in every room as the opaque walls block penetration of IR signals through the walls.

The drawback of IR technology is that the level of background noise is relatively high requiring higher radiated power from the IR station. Sunlight, indoor lighting, infrared remote controls, all are sources of background IR noise.

Pulse position modulation (PPM). IEEE 802.11 specifies Pulse Position Modulation (PPM) for infrared wireless local area networks. PPM optimizes the radiated power of the infrared source. In PPM, groups of 2 or 4 bits are formed. 2-bit group can have 4 possible combinations and 4-bit group can have 16 possible combinations. These are mapped to position of a pulse in symbol consisting of 4 or 16 positions respectively. Figure 13.26 shows an example of PPM in which 2-bit grouping is used. Note that each symbol of PPM stream contains three zeroes and single 1. When 4-bit grouping is used, each symbol contains fifteen zeroes and single 1.

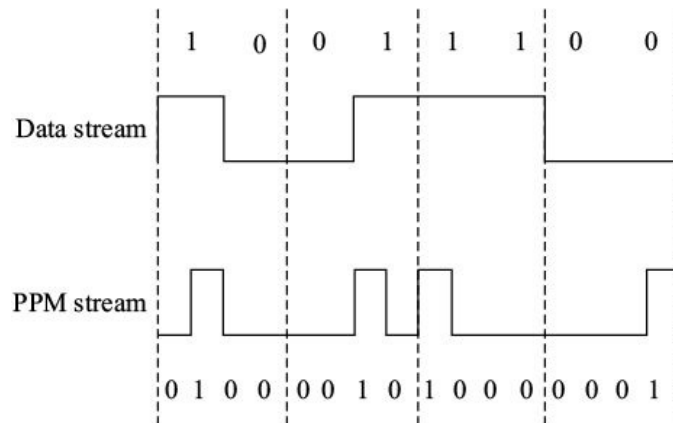


Figure 13.26 Pulse position modulation.

Infrared local area network use intensity modulation which is simply switching the infrared source ON and OFF. On every occurrence of 1 in the PPM stream, the device is switched on and radiates infrared signal.

13.7.3 Orthogonal Frequency Division Multiplexing (OFDM)

It is also known as multi-carrier modulation. In this scheme, the information data signal is divided into several data streams of lower bit rate. Each bit stream is, then, transmitted on a separate narrowband (312.5 kHz) sub-carrier. BPSK, QPSK, 16-QAM or 64-QAM modulation is used depending on the bit rate being transmitted on a carrier. Group of up to 52 such carriers constitutes one 16.6 MHz wide channel (Figure 13.27). The channel separation is 20 MHz.

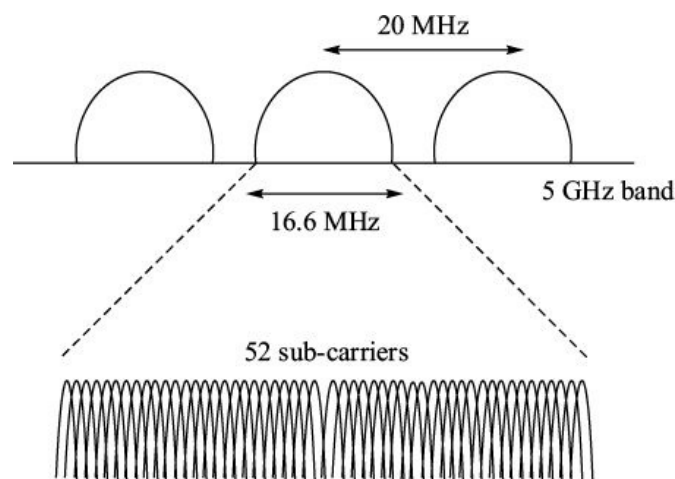


Figure 13.27 Orthogonal frequency division multiplexing.

13.8 PHYSICAL LAYER OF IEEE 802.11

The physical layer of IEEE 802.11 provides several transmission alternatives in

terms of frequency, bit rate, and modulation scheme. These specifications have been issued in several stages as IEEE 802.11, IEEE 802.11a, IEEE 802.11b and IEEE 802.11g. Figure 13.28 shows

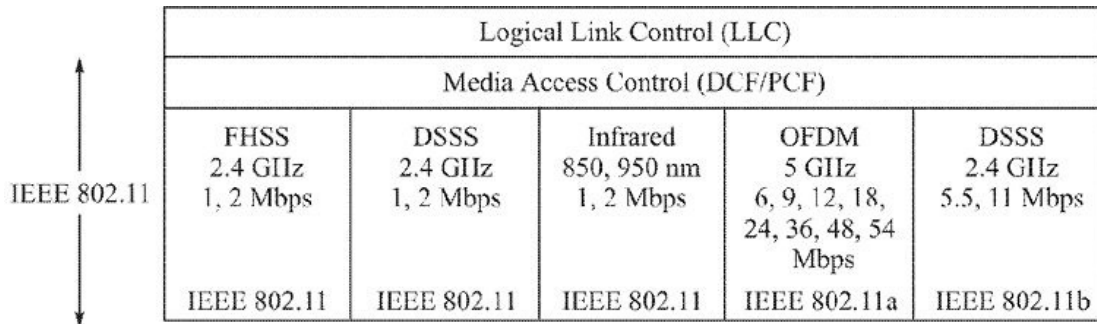


Figure 13.28 The physical layer alternatives in IEEE 802.11 wireless local area network.

the alternative physical layers of IEEE 802.11. IEEE 802.11g is not shown in the figure. It is extension of IEEE 802.11b and extends the bit rates up to 54 Mbps. Table 13.2 summarizes the features of physical layer as specified in these standards.

TABLE 13.2 Transmission Alternatives of IEEE 802.11 Physical Layer

Standard	Frequency	Coding scheme	Data rate (Mbps)	Modulation
IEEE 802.11	2.4 GHz	DSSS	1, 2	DBPSK, DQPSK
	2.4 GHz	FHSS	1, 2	2-level FSK, 4-level FSK
	850 nm,	4-bit PPM,		
	950 nm Infrared	2-bit PPM	1, 2	Intensity modulation
IEEE 802.11a	5 GHz	OFDM, FEC	6, 9, 12,	BPSK, QPSK,

		18, 24, 36, 54	16-QAM, 64-QAM
IEEE 802.11b	2.4 GHz	DSSS, CCK	5.5, 11
			QPSK

13.8.1 Original IEEE 802.11 Physical Layer

The original version of IEEE 802.11 defines the following three physical media. The bit rates specified for all the three media are 1 Mbps and 2 Mbps.

- 2.4 GHz using Direct Sequence Spread Spectrum (DSSS)
- 2.4 GHz using Frequency Hopping Spread Spectrum (FHSS)
- 850/950 nm infrared.

2.4 GHz DSSS. 2.4 GHz DSSS provides bit rate of 1 Mbps and 2 Mbps on one RF (Radio Frequency) channel in 2.4 GHz band. The modulation used for 1 Mbps bit rate is differential BPSK. For 2 Mbps bit rate, differential QPSK is used. The modulated RF carrier bandwidth is 5 MHz. Number of RF channels in 2.4 GHz frequency band depends on the bandwidth allocated by the regulatory agency of a country. IEEE 802.11 uses 11-bit Barker sequence 10110111000 as the spreading sequence. It has processing gain of $10 \log 11 = 10.4$ dB.

2.4 GHz FHSS. In this case, 2-level FSK is used for the 1 Mbps system and 4-level FSK is used for the 2 Mbps system. In 2-level FSK binary 0 and 1 are mapped two frequencies. In 4-level FSK, a group of two adjacent data bits is mapped to one of the four frequencies of the FSK signal. RF frequency hopping is carried out at the rate of 2.5 hops per second. Bandwidth of each RF channel is 1 MHz in 2.4 GHz band. The least hopping distance is 6 MHz in most of the countries. The number of RF channels depends on the allocated bandwidth by the regulatory agency of a country.

850/950 nm infrared. In this case, Pulse Position Modulation (PPM) is used. For 1 Mbps bit rate, a group of four bits is mapped to the position of 1 in string of 16 bits. Rest of the bits in the 16-bit string are zeroes. This 16-bit symbol modulates the infrared signal of wavelength 850 or 950 nm. Intensity modulation is used, *i.e.* the infrared signal is transmitted on each occurrence of 1. For 2 Mbps bit rate, groups of two bits are mapped to 4-bit symbols having three

zeroes and single 1.

13.8.2 IEEE 802.11a

IEEE 802.11a supports bit rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbps. It uses RF frequency band at 5 GHz. Instead of spread spectrum techniques, it specifies Orthogonal Frequency Division Multiplexing (OFDM), as described earlier. IEEE 802.11a uses forward error correction based on convolutional code at the rate of 1/2, 2/3 or 3/4. IEEE 802.11a gives a transmission range of 100 metres outdoor and 10 metres indoor.

13.8.3 IEEE 802.11b

IEEE 802.11b is an extension of IEEE 802.11 DSSS scheme. It provides data rate of 5.5 and 11 Mbps. The chipping rate is 11 Mbps which is same as in the original IEEE 802.11 DSSS scheme. The occupied bandwidth is thus the same. Higher data rate is achieved by using a modulation scheme called Complementary Code Keying (CCK). CCK is a complex modulation scheme and its description is beyond the scope of this book. IEEE 802.11b gives a transmission range of 30 metres outdoor and 10 metres indoor. An extension to IEEE 802.11b has been proposed as IEEE 802.11g standard. It supports bit rates of 6, 9, 12, 18, 24, 36, 48, 54 Mbps.

SUMMARY

Wireless local area networks are based on distributed and centralized access control mechanisms. Distributed access control uses CSMA/Collision Avoidance (CA) contention access method and is implemented as Distributed Control Function (DCF) in the MAC sublayer of IEEE 802.11. Transmission of a data frame in DCF can be 2-frame atomic operation consisting of exchange of the data frame and its acknowledgement. It can optionally be 4-frame atomic operation in which two additional frames, RTS and CTS frames, are exchanged to reserve the transmission media.

Point Coordination Function (PCF) of IEEE 802.11 is a centralized access control mechanism. It is contention free and is based on round robin poll of the stations by a designated station called point coordinator. PCF and DCF can co-exist in a wireless local area network in time-sharing mode.

IEEE 802.11 wireless local area networks use several types of transmission technologies that include Direct Sequence Spread Spectrum (DSSS), Frequency

Hopping Spread Spectrum, Orthogonal Frequency Division Multiplexing (OFDM), and infrared (850/950 nm). The RF frequencies allotted for wireless local area networks are in the frequency band 2.5 GHz and 5 GHz. The bit rates range from 1 Mbps to 54 Mbps.

EXERCISES

1. In DCF with RTS/CTS, the stations will not begin transmitting data frames until they have reserved the channel. Explain why the exponential back-off is still required.
2. Stations A and B use DCF with RTS/CTS. Assume that they are located at the fringe of each other's coverage areas. A has a data frame to send to B. What happened when
 - (a) RTS from A is lost?
 - (b) CTS from B is lost?
 - (c) data frame from A is lost?
 - (d) acknowledgement from B is lost?
3. Two stations A and B of an independent BSS have a frame to transmit to each other. Draw the timing diagram for A and B assuming that the frames arrived when the medium was busy. Assume that the stations communicate using the basic DCF communication mode, there are no collisions and the first data frame transmitted is lost due errors.
4. Complete the timing diagram shown in Figure E13.29. Assume DIFS = 50 ms, SIFS = 10 ms and back-off = 20 ms.

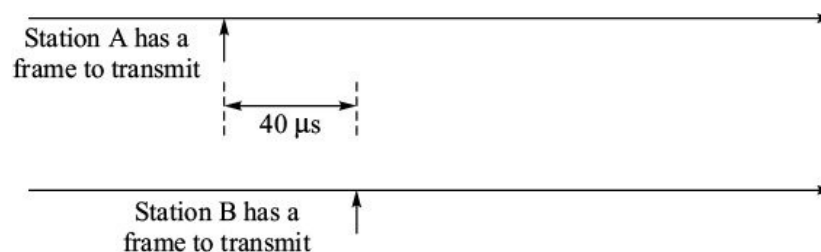


Figure E13.29.

5. In the FHSS system shown in Figure 13.23, the up-converter output at the transmitter is sum of IF and the output frequency of the frequency synthesizer. The output frequency of the synthesizer is determined by the 3-bit binary sequence as indicated below.

3-bit binary input	000	001	010	011	100	101	110	111
Frequency (MHz)	10	20	30	40	50	60	70	80

The 4-level FSK modulator generates 1, 2, 3, and 4 MHz IF output for binary inputs 00, 01, 11, and 10 respectively. If one complete cycle of the PRBS sequence is 001110011001001, fill in the transmitted frequencies for the data sequence shown in the table below. Is it slow or fast FHSS?

Input data sequence	0 1	1 0	0 0	0 0	1 1
Synthesizer binary input	001	110	011	001	001
Transmitted frequency (MHz)					

6. Do the Exercise 5 if the PRBS clock frequency is doubled.
7. A DSSS system uses 4-bit spreading code 0110. What is the transmitted DSSS sequence corresponding to the input data sequence 100110?
8. A DSSS system uses 4-bit spreading code 0110. What is the data sequence corresponding to the received DSSS sequence 011001101001100101101001?

1 Slot time is the time required to emit one data frame of minimum size.

2 The process of reserving the medium by specifying the time is referred to as *virtual carrier sensing*.

3 If the bit rate is R and there are L bits per symbol, the modulation rate will be R/L .

14

Bridges and Layer-2 Switches

Merely putting together a local area network does not meet requirements of an organization. Very soon the number of stations swells up, required geographic coverage goes beyond what is technically feasible and need arises to interconnect several local area networks to share common information base. LAN bridge, layer-2 switch, and router are the devices required to interconnect two or more local area networks. Router is a layer-3 device that we will take up in another chapter later. In this chapter we focus on layer-2 internetworking devices, bridges, and switches.

We start with the layered architecture of a LAN bridge and describe two types of bridges—transparent bridge used for interconnecting Ethernet LANs and source routing bridge used for interconnecting token ring LANs. We discuss in detail bridge protocols used in these bridges. We, then, move over to layer-2 switches. We discuss the motivation that lead to their development and their basic features before closing the chapter.

14.1 MOTIVATION FOR USING LAN BRIDGES

A local area network is a stand-alone network with capability to provide communication service to the end systems. There is, however, always need to communicate with the stations connected on other local area networks. Bridge is intermediary device that interconnects two or more local area networks. One would think that it should be simpler to form a bigger LAN that spans all the stations but it may not always be possible because:

- There is limit on the maximum number of stations that can be connected on

a single LAN.

- The maximum physical size of a LAN is limited.
- Upgrading the existing infrastructure can be very costly and the investments already made are wasted.

Therefore, the solution is to interconnect two LANs as separate entities using a bridge.

At times, bridge installation may be done from a different perspective. There may be reasons to partition an existing LAN into several LANs using bridges. For example, if the performance of a LAN is poor because of too many collisions, it can be improved by partitioning a LAN into several interconnected LANs using bridges. A bridge partitions the collision domain and therefore the performance improves.

A bridge is to be differentiated from a repeater and a router. A repeater merely regenerates electrical signals and expands the physical domain a local area network. It works at the physical layer level. Thus, a network having repeaters still constitutes one single LAN and single collision domain.

Router is a packet switch implemented at the layer-3. The packets are routed and forwarded at layer three. Thus, all physical layer and data link layer protocol differences are taken care of. The addressing scheme at the layer three is structured. A layer-3 address is composed of two parts—network part and host (end system) part. Therefore, unlike local area networks which work in broadcast mode, more intelligent and dynamic routing decisions are possible in layer three networks. We will study such networks later in the book.

14.2 LAN BRIDGE

Bridge is a device that is attached to two or more local area networks. It takes MAC frames from one LAN and sends them across to the other LAN. A MAC frame is transferred across the bridge only if its destination address is in the other LAN. Note the difference between a bridge and a repeater. A repeater does not look at the address field and regenerates all the frames.

The interconnected LANs across a bridge need not be of the same type theoretically. But the differences in the LAN technologies make it impractical to design such bridges for the following reasons:

- Each type of LAN has its own MAC sublayer frame format.
- The maximum size of the MAC frames is different in LAN technologies.
- The data rates are different.
- LAN management and priority management functions are different.

Therefore, bridges generally interconnect LANs of the same type. Interconnection of different LAN types is nowadays dealt with using routers.

14.2.1 Bridge Architecture

Figure 14.1 shows the layered architecture of a two-port bridge. It has the physical layer and the MAC sublayer at its each port. The protocols of the physical layer and the MAC sublayer at each port of the bridge match with the protocols of the respective LAN. The MAC sublayers of the bridge have relay and routing function between them.

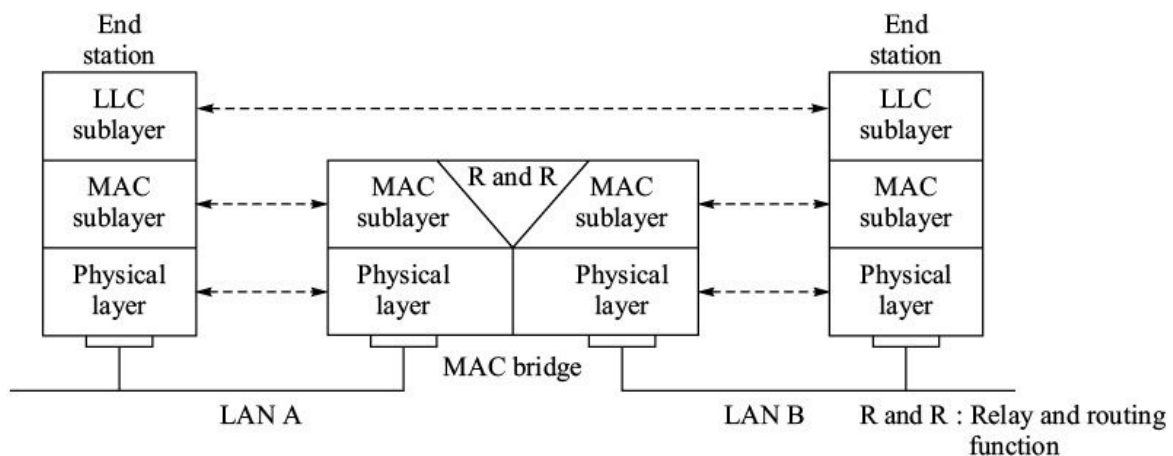


Figure 14.1 Layered architecture of a bridge.

When a MAC frame is received by the bridge at its one port, it examines its destination address. If the bridge decides that the frame should be transferred across, it forwards the frame to the other port. We assume that the interconnected LAN types are same.

The decision whether to forward the frame or not, is either taken by bridge (transparent bridge), or alternatively, the MAC frame itself contains the routing instruction (source routing bridge). Transparent bridge and source bridges are two bridge types, we discuss later.

It is to be noted that:

- The bridged LANs maintain their identity although the users may not

perceive so.

- The bridge stores and forwards frames and therefore the collision domains of Ethernets remain the same. In other words, a bridge does not extend the collision domain of one Ethernet to the other interconnected Ethernet.

14.2.2 Types of Bridges

The most widely deployed bridges are of two types:

- Transparent bridges
- Source routing bridges.

Transparent bridges are also known as spanning tree bridges. They are used in Ethernet environment. The transparent bridge architecture is defined in IEEE 802.1d, MAC bridges. The corresponding ISO standard is 10038. Source routing bridges are used for token ring LANs. Source routing is part of IEEE 802.5 token ring specifications.

Besides the protocol differences, these two types of bridges differ in the way the routing decision is made. A transparent bridge is invisible to the stations in the network. The stations perceive the network as an extended network. The forwarding decisions are made by the bridge. A transparent bridge creates and maintains a comprehensive forwarding table of all destination stations.

In source routing bridges, as the name suggests, the routing is decided by the station sending a frame. The station incorporates the route to be followed to the destination in the frame. The bridge merely follows the routing instructions incorporated by the source. The source routing bridges do not require any routing table for forwarding frames. Transparent bridges, therefore, are more complex than source routing bridges.

14.3 TRANSPARENT BRIDGES

A transparent bridge works in ‘plug and play’ mode, *i.e.* when the bridge is attached to two Ethernet LANs, it starts working. The networking software/configurations of the stations need not be altered in any way. The bridge achieves this result by operating in promiscuous mode, *i.e.* it receives all the frames whether or not they are addressed to it. It builds a forwarding table by looking at the source addresses of the frames that hit its ports. Based on the

table, the bridge can, then, forward the frames from one Ethernet to the other. Thus we can describe the operation of a transparent bridge as consisting of the following basic functions:

- Frame filtering and forwarding the frames based on forwarding table.
- Learning the addresses of the stations and creating forwarding table.

14.3.1 Frame Filtering and Forwarding

As Ethernet LANs operate in broadcast mode, all the frames, irrespective of their destination, appear at the port of the bridge connected to the LAN (Figure 14.2). The bridge maintains a table of the MAC addresses of all the stations on the extended bridged network. The table is maintained port-wise. In other words, by looking in the table, it is possible to find out the port on which a particular destination address is available. When a bridge receives a frame at any of its ports, it takes one of the following actions:

Filtering. If the destination address is available on the same port through which it received the frame, the bridge filters (ignores) the frame.

Forwarding. If the bridge determines that the destination address is available on a different physical port than the one through which the frame was received, it forwards the frame onto that port.

If the destination address on the frame is a global address or multicast address, it sends the frame on all its ports except the one from which it received the frame.

Flooding. If the bridge does not find the address in its forwarding table, it floods (sends) the frame onto all the physical ports except the one from which it received the frame.

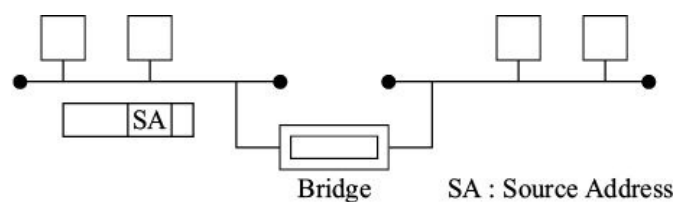


Figure 14.2 Bridged ethernet LANs.

14.3.2 Learning Addresses

When a bridge comes up for the first time, its forwarding table is empty. It builds up the address database by examining the source address field of the

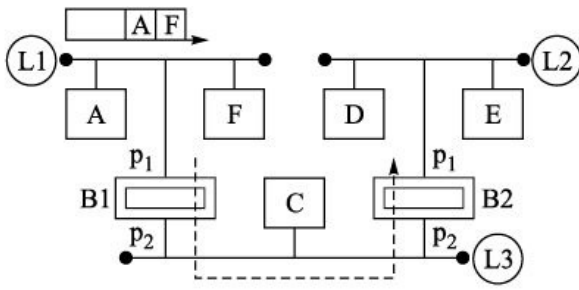
MAC frames that hit its ports. Whenever the bridge finds that the source address is not in its forwarding table, it updates the forwarding table with the source address and the identity of the port at which the frame was seen. Thus as the stations exchange the frames, the bridge gradually builds up its forwarding table. Let us understand the process with an example.

There are three local area networks L1, L2, and L3, two bridges B1 and B2, and five stations, A, C, D, E, and F, on these networks (Figure 14.3):

- To start with, the forwarding tables of bridges B1 and B2 are empty. Station A sends a frame to station F on L1.
- Bridge B1 receives the frame on its port p_1 . It floods the frame through its port p_2 . It also adds updates its forwarding table. It adds station A under port p_1 (Figure 14.3a).
- We use the term ‘flood’ here to emphasize that the bridge sends the frame on all its other ports, which happen to be only one, p_2 , in this case.
- Bridge B2 receives the frame on its port p_2 . It floods the frame through its port p_1 . It also updates its forwarding table. It adds station A under port p_2 (Figure 14.3a).
- Station F replies to A on L1. Bridge B1 filters the frame as station A is available on the same LAN. It also updates its forwarding table. It adds station F under port p_1 (Figure 14.3b).
- Station F sends a frame to station E. Bridge B1 receives the frame on its port p_1 . It floods the frame through its port p_2 (Figure 14.3c).
- Bridge B2 receives the frame on its port p_2 . It floods the frame through its port p_1 . It also updates its forwarding table. It adds station F under port p_2 (Figure 14.3c).
- Station E sends its reply to station F. Bridge B2 receives this frame on its port p_1 . It forwards the frame to L3 through its port p_2 after consulting its forwarding table. It also updates its forwarding table. It adds station E under port p_1 (Figure 14.3d).
- Bridge B1 receives the frame on its port p_2 . It forwards the frame on to L1 through its port p_1 after consulting its forwarding table. It also updates its forwarding table. It adds station E under port p_2 (Figure 14.3d).
- The process continues and the bridges build up their forwarding tables

(Figure 14.3e).

All the entries in the forwarding table have a lifetime. If an entry is not refreshed frequently by arrival of frames from the respective source, it is timed out and deleted from the table. This deletion is required to ensure that network changes are quickly taken care of. If a station is moved from one Ethernet LAN to another, the old entry is deleted and the new entry is rebuild automatically. Similarly, if a station is down, its entry is deleted from the table automatically.

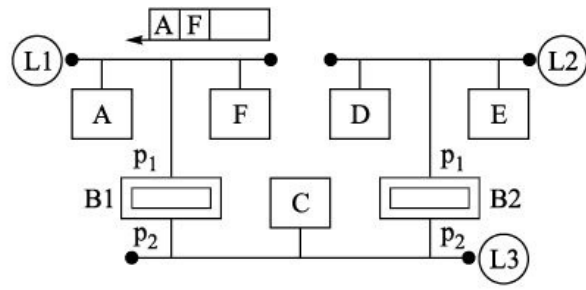


P ₁	P ₂
A	

P ₁	P ₂
	A

Forwarding table of B1 Forwarding table of B2

(a)

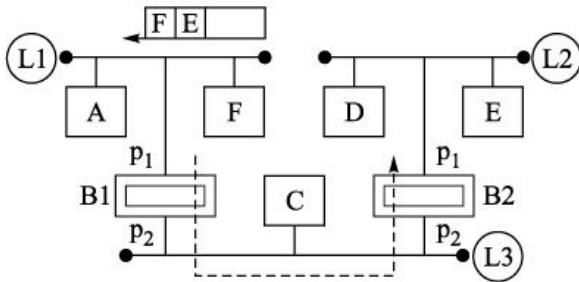


P ₁	P ₂
A	
F	

P ₁	P ₂
	A

Forwarding table of B1 Forwarding table of B2

(b)

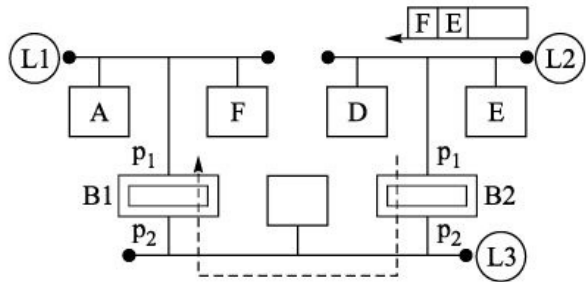


P ₁	P ₂
A	
F	

P ₁	P ₂
	A
	F

Forwarding table of B1 Forwarding table of B2

(c)



P ₁	P ₂
A	
F	

P ₁	P ₂
E	A
	F

Forwarding table of B1 Forwarding table of B2

(d)

P ₁	P ₂
A	E
F	C
	D

P ₁	P ₂
E	A
D	F
	C

Forwarding table of B1 Forwarding table of B2

(e)

Figure 14.3 Learning addresses.

14.3.3 Multiple Paths

If the topology of interconnected Ethernet LANs has loops, there will be more than one path interconnecting two segments of a network. Multiple interconnecting paths can affect network operation in three ways:

- Confusion during address learning
- Endless circulation of frames with unknown destination
- Broadcast storms.

Consider that there are two bridges B1 and B2, interconnecting two Ethernet LANs—L1 and L2 (Figure 14.4). The forwarding tables of the bridges are empty to start with. When station A on LAN L1 transmits a frame addressed to station B on LAN L2, both the bridges will take note that the station A is available on their port p₁. They will individually forward the frame to LAN L2 through their port p₂. These frames will be received by station B. Frame released by one bridge will also be received by the other bridge.

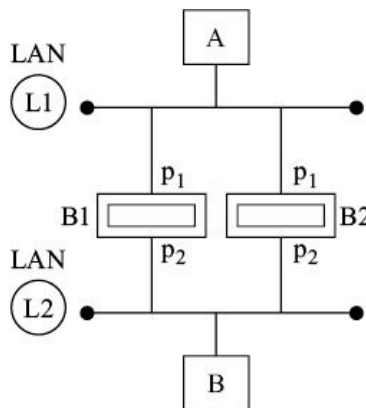


Figure 14.4 Multiple paths affect address learning.

When the frame released by bridge B2 on LAN L2 is received by bridge B1, bridge B1 immediately updates its forwarding table again based on the source address contained in the received frame. It also transmits the frame through its port p₁. The forwarding table now indicates that A is available on port p₂. Bridge B2 also does the same when it receives the frame released by B1 on LAN L2. Although station A is on LAN L1, the forwarding tables of the two bridges do not indicate so. Thus, these two frames keep circulating in the ring and each time update the entries of the forwarding table.

Figure 14.5 shows another situation where a broadcast frame creates a storm of frames. Station P on LAN L1 sends a broadcast frame. Bridge B3 receives

two frames from bridges B1 and B2. Bridge B4 sends these two frames back to LAN L1. Bridges B1 and B2 forward these two frames back to bridge B3 as four frames. Thus the number of frames gets multiplied as they go along the loop anticlockwise. We have not yet considered the storm that will be generated in clockwise direction by the first frame that hits bridge B4 directly.

Multiple paths are built into the network to improve network resilience against failure of segment or a bridge of a network. When one path fails, the alternative path takes over. But

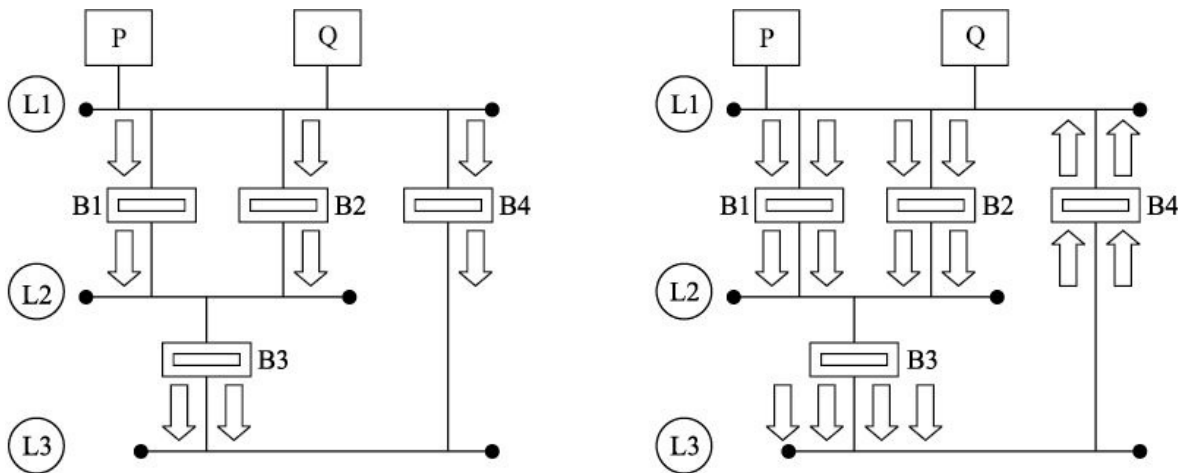


Figure 14.5 Frame storm due to parallel paths.

multiple paths cannot be allowed to coexist. Therefore, what we need to ensure is that there is only one path interconnecting any two segments of the network at any point of time in the network. If there is a failure in any section of the network, an alternative path is automatically built through the redundant network resources. This is achieved using the spanning tree protocol, which we examine next.

14.4 SPANNING TREE ALGORITHM

A spanning tree is a graph structure that includes all the bridges and stations on an extended network but it never has more than one active path connecting any two stations. Therefore, the complications of multiple paths never occur. For example, Figure 14.6b shows the spanning tree topology of the network shown in Figure 14.6a. Note that all the multiple paths have been removed from the network.

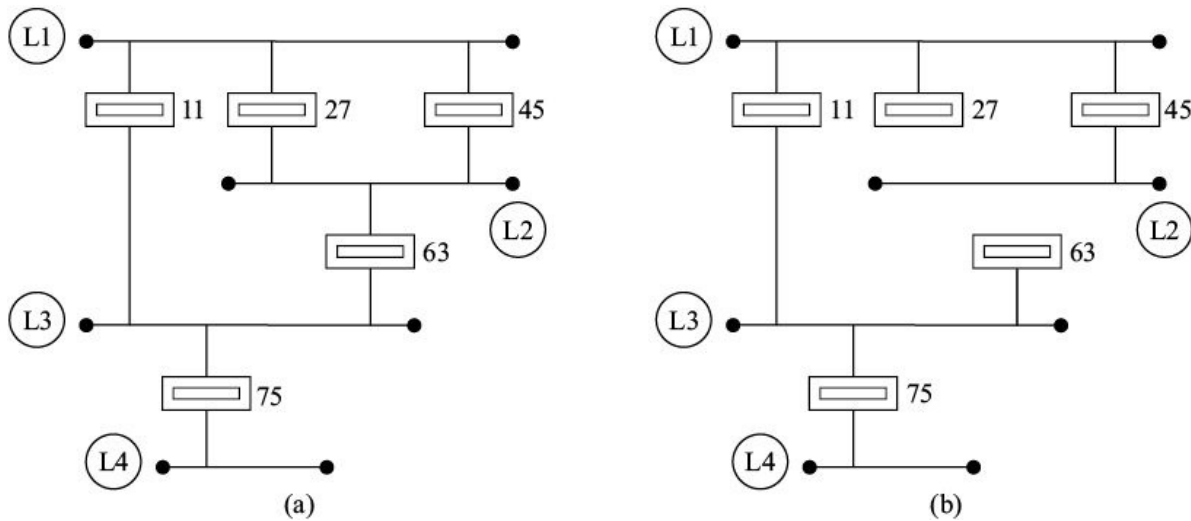


Figure 14.6 Spanning tree topology.

What we need is an algorithm by which the bridges will be able to derive the spanning tree automatically. The algorithm must be such that topological changes in the network due to additions, deletions of stations or faults are automatically discovered and spanning tree is reworked dynamically. Before we go into the algorithm, let us understand some of the definitions associated with the algorithm.

Root bridge. Each bridge has a unique identifier. The bridge with the lowest identifier is called the root bridge.

Port cost. Each port of a bridge has an associated cost parameter, which is the cost of transmitting a frame through the particular port. The default value of the port cost is $1000/(\text{bit rate in Mbps})$. It can be set to a different value by the network administrator. The port cost is associated with the bits transmission through a port. Reception of bits through a port has no cost attached to it.

Root path cost. Root path cost is the cost associated with a path to the root bridge from a port of another bridge. It is sum of all the intervening port cost parameters. Note that cost of the transmitting ports only is added to arrive at the root path cost.

Root port. A bridge may have several paths emanating from its ports to the root bridge. The port having the least root path cost is called root port. If there are more than one such ports, the port with lowest port identification number is chosen as the root port.

Designated bridge. If there are several bridges connected to a LAN segment, one of the bridges is the designated bridge for forwarding the data frames from

that LAN segment. Selection of the designated bridge is based on the root path cost. The bridge that offers least root path cost from the LAN segment is selected as the designated bridge for that LAN segment. If more than one bridges have the same root path cost, the bridge with lower identifier number is chosen as the designated bridge.

Designated port. The port of the designated bridge that connects to the LAN segment is called designated port.

14.4.1 Bridge Protocol Data Unit (BPDU)

To construct the spanning tree and for other management functions, a control frame called configuration Bridge Protocol Data Unit (BPDU) is exchanged by the bridges. Configuration BPDUs are generated by the bridges and have MAC multicast address. A configuration BPDU generated by a bridge contains the following information:

- Bridge Identifier (B-Id)
- Root Identifier (R-Id) as perceived by this bridge
- Root Path Cost (RPC) of the bridge
- Port Identifier (P-Id)
- Other information.

The format of BPDU is shown in Figure 14.7. The protocol identifier field contains 0000 (Hex) for IEEE 802.1d spanning tree protocol. BPDU type contains 00 (Hex) for configuration BPDU. There are several other fields that indicate age of BPDU, time delay before a port is put in forwarding mode, time intervals after which configuration BPDUs are generated, *etc.* We will restrict ourselves to understanding of spanning tree protocol and will not dwell further on other aspects of the IEEE 802.1d standard which describes the protocol in detail.

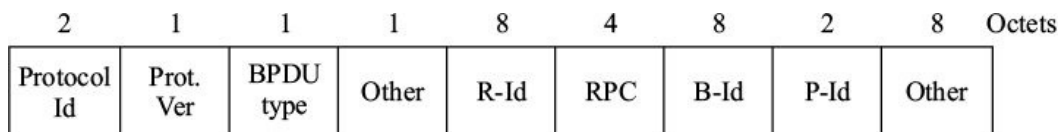


Figure 14.7 Format of BPDU.

14.4.2 Constructing the Spanning Tree

Constructing the spanning tree involves the following steps:

1. Identification of the root bridge
2. Identification of the root ports
3. Selection of the designated bridges for every LAN.

After the above steps are carried out, the root ports and the designated ports of the bridges are set in forwarding mode and all other ports are retained in blocked mode. In forwarding mode, a port becomes active for the data frames. Once this done, the goal of spanning tree algorithm is achieved and there will be only one path from any station to any other station in the network.

To understand spanning tree algorithm, let us consider a network as shown in Figure 14.8. It consists of four LAN segments L1 to L4. There are five bridges having identifiers 11,

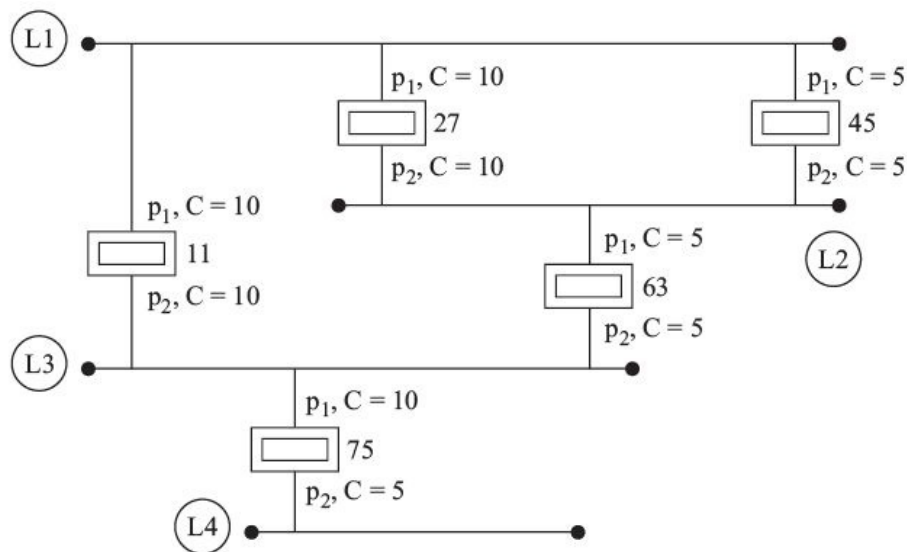


Figure 14.8 Developing spanning tree (1).

27, 45, 63, and 75. The port cost (C) associated with each port is also indicated in the figure.

To begin with, all the ports of all the bridges are in blocked mode. A port in blocked mode does not accept any data frame but it accepts BPDUs.

Selection of the root bridge. The bridge with the lowest bridge identifier is selected as root bridge. Root bridge is identified as follows:

- Each bridge asserts that it is the root bridge by sending a configuration BPDU on all its ports (Figure 14.9). It indicates its identity, root bridge identity which is its own identity, root path cost which is zero and port

identity in the BPDUs.

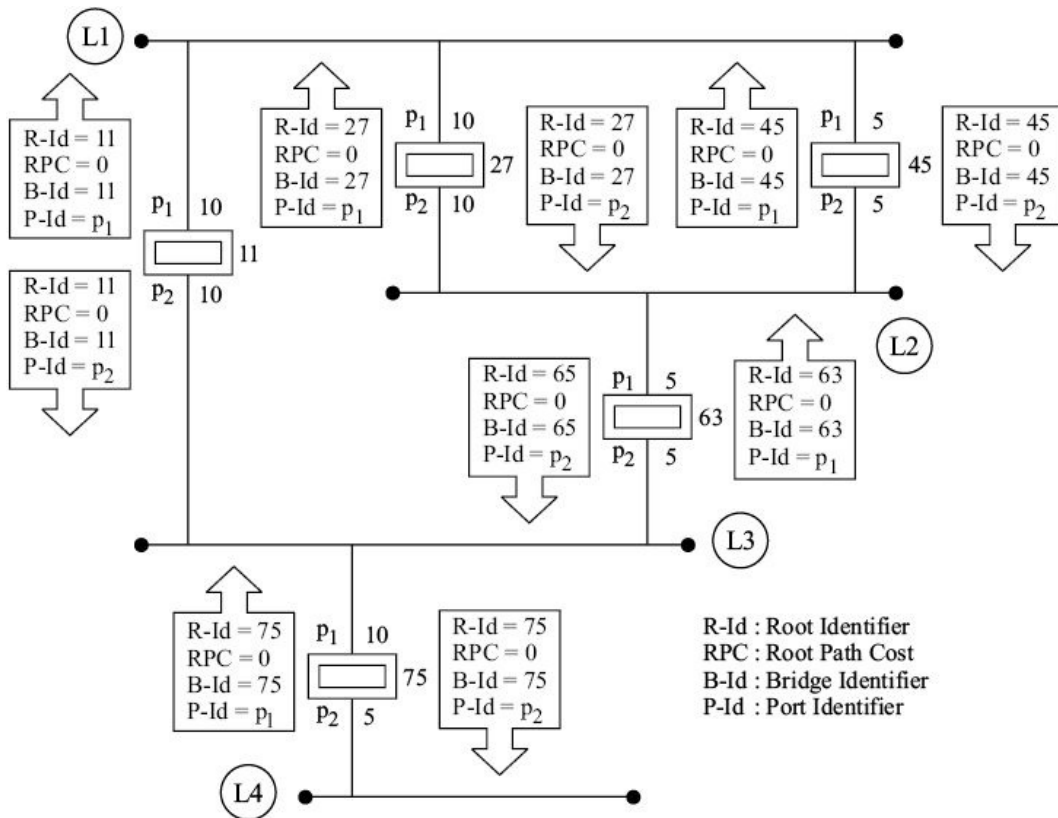


Figure 14.9 Developing spanning tree (2).

- If a bridge receives a configuration BPDUs with lower root bridge identity than its own identity on any of its ports, it stops sending its configuration BPDUs and forwards the received BPDUs on to its other ports. When forwarded by a bridge, a BPDUs undergoes the following changes:
 - The bridge puts its own identity in the B-Id field.
 - The bridge puts its own port identity in the P-Id field.
 - The root path cost is incremented by the cost of the port through which the BPDUs is received because later it will be transmitting through this port.
 - The root bridge identity is retained as it is.
- If a bridge receives a configuration BPDUs with higher root bridge identity than its own identity on a port, it continues sending its configuration BPDUs.

Thus all the bridges except bridge 11 stop sending their configuration BPDUs. Bridge 11 is identified as the root bridge by all the other bridges.

Identification of the root ports. Every bridge other than the root bridge determines its root port, which is the port with lowest root path cost. It involves the following steps:

- The configuration BPDUs generated by the root bridge are forwarded by all the other bridges on their ports after making the necessary changes in the B-Id, P-Id, and RPC fields (Figure 14.10). The root path cost is incremented by the cost of the port through which the BPDU is received.
- Each bridge determines which of its ports has the lowest root path cost and that port is selected as the root port. For example, bridge 27 receives three BPDUs, one from root bridge 11 on port p₁, second from bridge 45 on port p₂, and the third from bridge 63 also on port p₂. It calculates the root path costs as 10, 15, and 15 from these BPDUs respectively. It decides p₁ as its root port. If there is a tie, the port with lower port identifier is selected as root port.
- The root ports are put in the forwarding mode. Other ports remain in the blocked mode.

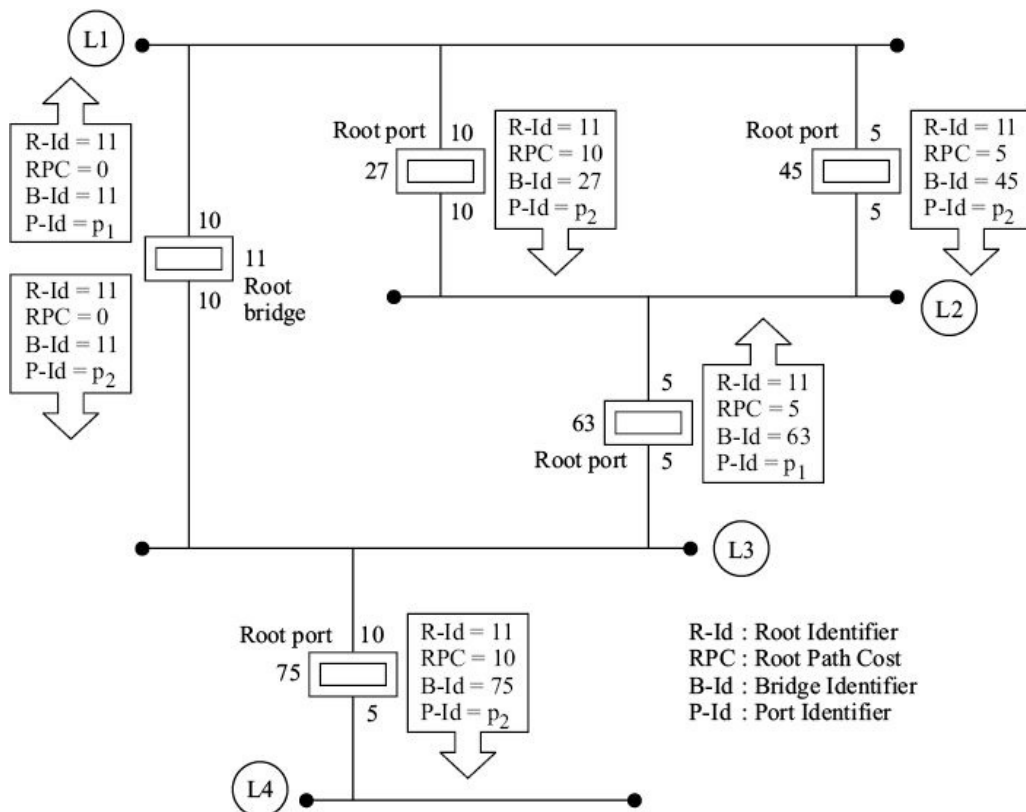


Figure 14.10 Developing spanning tree (3).

Selection of the designated bridges and the designated ports. A bridge is designated for each LAN segment to carry its data frames to the other LAN segments. If there are several bridges connected to a LAN, the bridge that has lowest RPC is selected as the designated bridge. For example, LAN L2 has three bridges, 27, 45, and 63, attached to it. Bridge 45 is the designated bridge for L2. port p₂ of bridge 45 that connects to LAN L2, is called the designated port.

The decision of selecting the designated bridge is made by the bridges themselves. For example, in case of LAN L2, the bridges 27, 45, and 63 decide as follows:

- When bridge 45 receives BPDUs offering RPC of 10 from bridge 27 and RPC of 5 from bridge 63, it decides as below:
 - Bridge 27 has higher RPC and therefore it cannot be the designated bridge for LAN L2.
 - Bridge 63 has the same RPC but it has higher B-Id and therefore it cannot be the designated bridge for LAN L2.Having carried out this exercise, bridge 45 assumes the role of designated bridge and puts its port p₂ (designated port) in the forwarding mode.
- Bridges 27 and 63 carry out similar exercise and decide to keep their ports that connect to LAN L2 in the blocked mode.

The ultimate network scenario is shown in Figure 14.11. The dotted lines indicate the blocked links. Note that the root bridge is the designated bridge for all the LANs connected to it directly. From this moment onwards, normal network operation is possible. Every station will have only one path to any other station on the network.

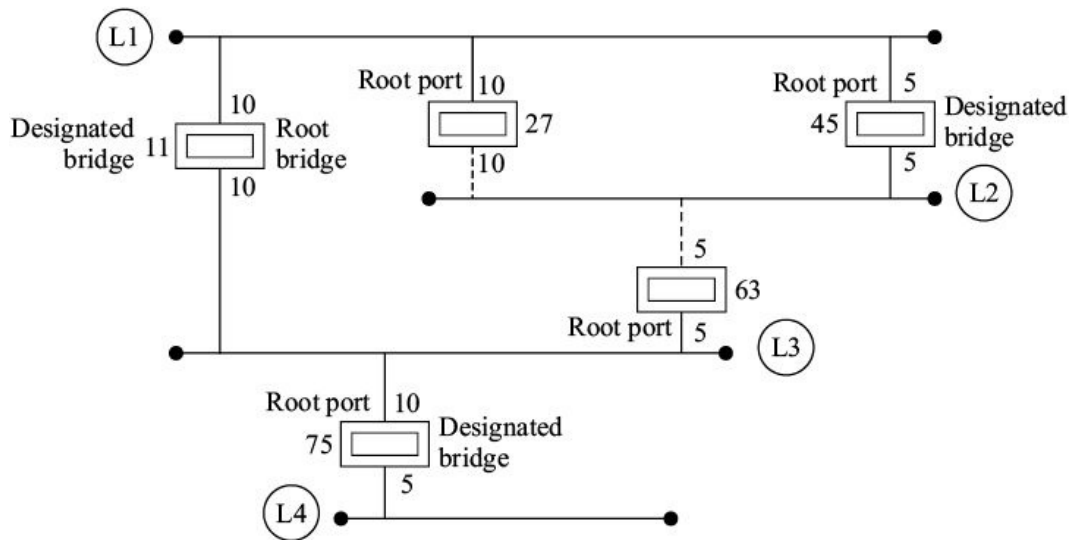


Figure 14.11 Developing spanning tree (4).

14.4.3 Error Situations and Limitations of Transparent Bridge

Normally the root bridge sends a configuration BPDU every 1 to 10 seconds. This BPDU is forwarded to all the ports of all the bridges (even the blocked ports). If the BPDU is not received within specified time, there are two possibilities:

- The root bridge has failed.
- The designated bridge has failed.

In the first case, the whole spanning tree is rebuilt. In the second case, if an alternative bridge is available which can support the affected LAN segment, it will take over as the designated bridge of that LAN segment. Thus, redundancy built into the network comes handy in case of failure of a network element (link or a bridge).

There are two limitations of transparent bridges:

- If there is a spurt in traffic, there is no possibility of the sharing of traffic on the redundant ports because these are blocked.
- The routes defined by the spanning tree are not optimal. For example, a frame from L2 to L3 could have easily gone through bridge 63, but it goes through bridges 45 and 11 (Figure 14.11).

14.5 SOURCE ROUTING BRIDGES

The transparent bridges use the MAC destination address of a frame to direct it. The route is decided by the bridges based on the forwarding tables built by them. In source routing, each source station is expected to know the route over which to send its frames across the bridged network. The routing information is included in the frames and the bridges en route forward the frames to their destinations based on this routing information. Unlike the transparent bridges we discussed earlier, the source routing bridges are not transparent. The stations know the presence of the bridges and indicate the routing information using their identities. The routes are determined by the stations using a route discovery procedure that we will study shortly.

Source routing bridges are simpler in architecture compared to transparent bridges. Source routing allows load sharing on the redundant paths but it has the drawback of generating additional and extensive network traffic because of its route discovery procedures.

Source routing was developed by IBM and adopted by IEEE in their token ring standard IEEE 802.5. Let us look at the frame structure before we proceed to the operation of the source routing algorithm.

14.5.1 Frame Structure

Figure 14.12 shows the format of IEEE 802.5 token ring MAC frame structure. The Routing Information Indicator (RII) field (first bit of source address SA) indicates whether the frame contains Routing Information (RI) field or not. RII = 1 indicates that RI field is present.

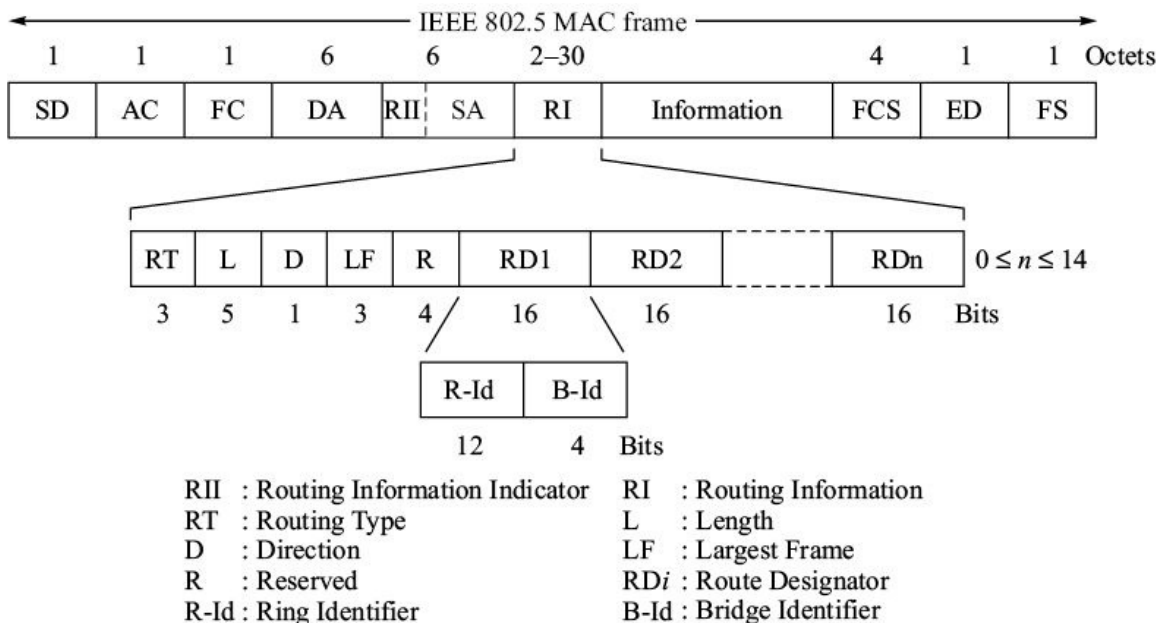


Figure 14.12 Format of routing information (RI) field.

RI field consists of the following subfields:

Routing type (3 bits). This subfield contains routing directive for the bridge. These directives are explained later.

Length (5 bits). This subfield gives the length of RI field in octets. Its range is 2 to 30 octets. RI field can therefore contain up to 14 Route Designators (RD).

Direction (1 bit). This subfield indicates the direction of the route (from RD1 to RDn or the other way). Its use allows the route designator subfields to appear in the same sequence in the to and fro frames between two end stations.

D = 0 Route designators are to be interpreted from left to right.

D = 1 Route designators are to be interpreted from right to left.

Largest frame (3 bits). This subfield gives the maximum size of the information field of the MAC frame that can be handled in the designated route. When a route is being discovered, the bridges en route indicate the largest size of the information field which can be sent on the route. Each bridge updates this field if the current value in the subfield exceeds what the bridge or the adjoining LAN can handle.

000: 516 octets 100: 8191 octets

001: 1500 octets 101: 11407 octets

010: 2052 octets 110: 17800 octets

011: 4471 octets 111: 65535 octets

Reserved (4 bits). This subfield is reserved.

Route designator, RDi:(16 bits). It contains a 12-bit ring identifier and a 4-bit bridge identifier. These identifiers are maintained by the bridges and are inserted by them in the route discovery frames. The route is from RDi to RDn or the other way depending on the direction bit (D).

A route can be defined in terms of maximum 13 bridges as the last route designator has only the ring identifier. There is no bridge identifier.

14.5.2 Routing Directives

Each frame contains one of the following routing directives. These directives are coded in the Routing Type (RT) subfield.

RT = 0 This is Non-Broadcast (NB) directive that instructs a bridge to follow

the route given in the route designator field.

RT = 1 0 This is All-Routes-Broadcast (ARB) directive that instructs a bridge forward the frame to all its port except the one from which the frame was received. In this case the destination will receive multiple copies of the frame from all possible routes.

RT = 1 1 This is Single-Route-Broadcast (SRB) directive that results in appearance of the frame only once in each LAN. This is achieved by sending the frame through a spanning tree. The destination gets only one copy of the frame in this case.

14.6 ROUTE DISCOVERY IN SOURCE ROUTING

Before a source station transmits a data frame to the destination station, it must locate the destination station. It first tries to locate the destination on the local ring by sending a test MAC frame containing destination address and with RII bit 0. The test MAC frame contains LLC-PDU (XID or Test). If the destination station is not located in the local ring, it sends a route discovery frame across the entire network. There are two ways of doing this:

- All-Route-Broadcast (ARB)
- Single-Route-Broadcast (SRB).

14.6.1 Route Discovery Using All-Route-Broadcast (ARB)

All-routes-broadcast works as follows:

- The source station sends a route discovery frame with an all-route-broadcast directive to all the bridges on the local ring in its bid to determine the route to a particular destination station.
- Each bridge en route
 - adds the RD_i subfield in the route discovery frame (Figure 14.13). RD_i subfield indicates identity of the bridge and the identity of the ring in which the frame is released by the bridge,
 - updates the LF subfield, if required,
 - forwards the frame to another ring.

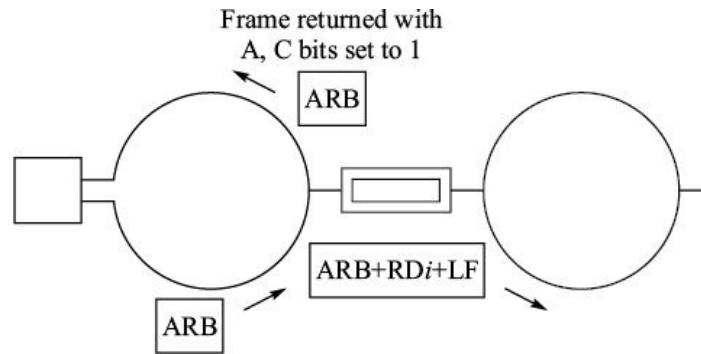


Figure 14.13 All-route-broadcast frame for route discovery.

To prevent multiple visits to the same bridge due to the loops present in the network, each bridge checks the RI field before forwarding a frame to another ring. If the frame has been to the ring already, it does not forward the frame into that ring.

If a bridge forwards the frame to another ring, it must tell the source station (or bridge) that it has copied the frame. It sets A and C bits of the FS field in the MAC frame to 1 indicating the frame has been copied (Figure 14.13).

- Eventually multiple copies of the route discovery frame reach the destination station and each copy indicates a different route. The destination station, then, sends a response frame using a non-broadcast directive on each route.
- The source station receives multiple responses from the destination. Each response indicates a different route to the destination. The source station selects a route to send the MAC data frames.

An aging mechanism is provided to update the routing information periodically. Let us take an example to understand the process. In Figure 14.14, station A on ring R-1 wants to discover route to station B which is on ring R-4.

- Station A sends a route discovery frame in ring R-1 with the following fields (Figure 14.14a):
 DA = B address of the destination station.
 RII = 1 so that bridges on the ring process the frame.
 RT = 1 0 so that ARB mode is selected and the frame gets broadcast all over the network.
 L = 2 since RI field is two octets long.
 D = 0 the RD octets to be appended will be read left to right.

It sets the LF field also to the largest known size of a frame allowed on the network.

- Bridge B-1 forwards the frame onto ring R-4 after adding input ring identifier (R-1), own identity (B-1), and output ring identifier (R-4) as RD1 and RD2 subfields. It also modifies the length field (L) accordingly (Figure 14.14b). The bridge identifier in RD2 is kept as 0. If the bridge B-1 and output ring R-4 support smaller size if data frames, bridge B-1 updates the LF field.
- Bridge B-2 picks up the above frame and forwards it onto ring R-3 after adding its identity in RD2 and adding subfield RD3 with ring identity R-3 (Figure 14.14c). It also modifies LF field, if required.
- The frame is copied by the destination station B. It is also picked up by bridge B4, which forwards it to ring R-2 in the same manner as above. When B-3 receives the frame from R-2, it does not forward it to R-1 since R-1 is already present in the RD field. Thus multiple visits to the same ring are avoided.

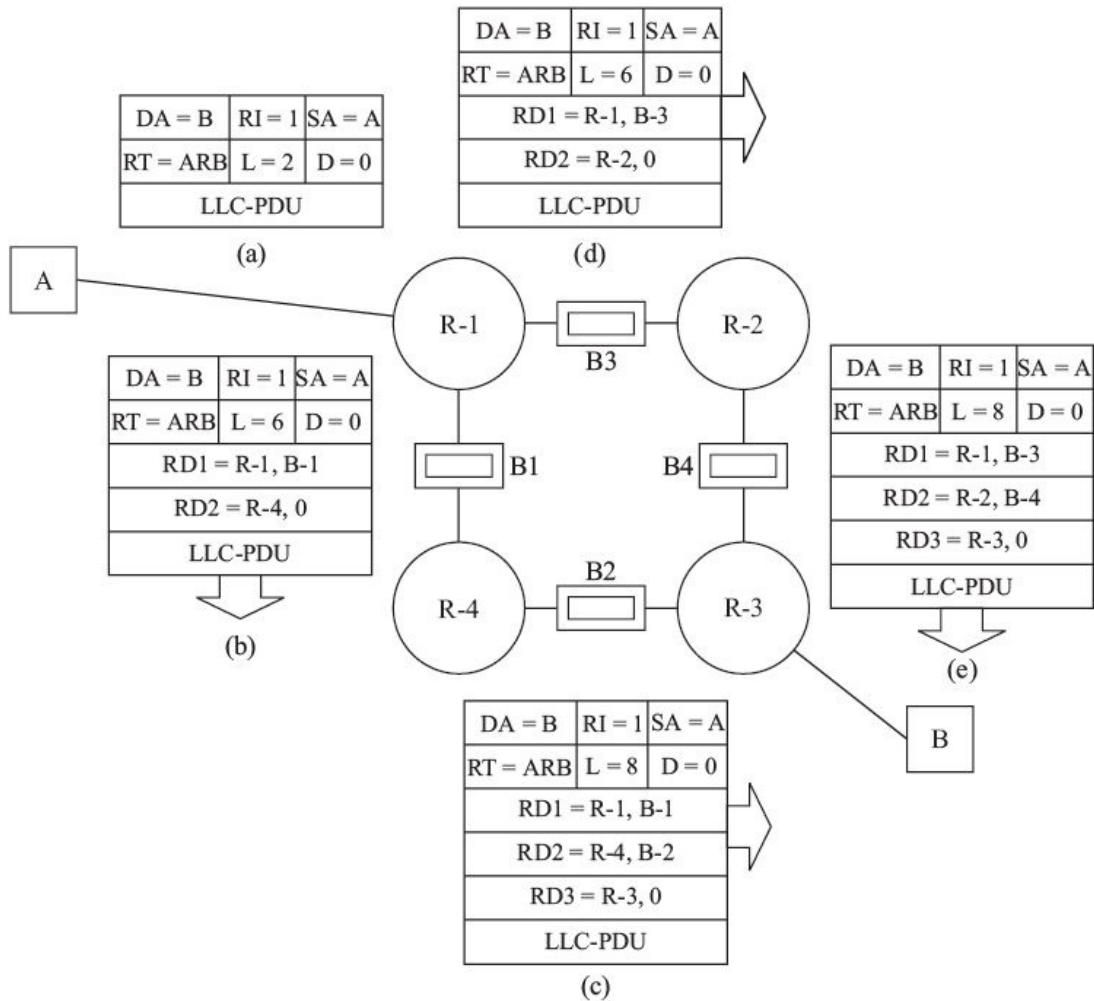


Figure 14.14 Route discovery in ARB.

- Another copy of the route discovery frame travels via R-1, B-3, R-2, B-4, R-3 to the destination station B (Figure 14.14e). This frame travels further to B-2, R-4, and B-1. It is filtered by B-1 to avoid the multiple visits to the same ring.
- Station B, thus receives two route discovery frames having different routes. Both the frames are returned by the destination B to the source station A with RT = 0 (Non-broadcast directive), DA = A and SA = B. In the return frames, station B makes direction (D) bit equal to 1 so that the bridges read the route from the right to the left. These frames take the route as already defined in their RD field.
- The source station A receives the responses from station B and selects one of the received routes to station B for future communication. Usually the route received first is selected.

14.6.2 Route Discovery Using Single-Route-Broadcast (SRB)

Route discovery using all-routes-broadcast generates multiple route discovery frames in the network. If the destination address does not exist, the wasted effort is costly because multiple route discovery frames are generated. A destination station may not exist in the network because the destination address is wrong or the station is down and bypassed by its RIU.

ARB process can be improved if the destination station sends the route discovery frames to the source. The source sends a Single-Route-Broadcast (SRB) to the destination station which responds with all-route-broadcast to discover all the routes from destination to the source station. From the received responses, the source can select a route for future communication with the destination station.

For single-route-broadcast, the spanning tree is required. Spanning tree can be configured by the network administrator. Alternatively, it can be dynamically built using spanning tree protocol that we studied in transparent bridges. The following example illustrates route discovery mechanism using SRB. It is assumed that the spanning tree is available, and there are no loops or multiple paths in the network.

Station A on ring R-1 wants to discover routes to station B which is on ring R-4 (Figure 14.15).

- Station A sends an SRB route discovery frame in ring R-1 with (Figure 14.15a)
DA = B address of the destination station.
RII = 1 so that bridges on the ring process the frame.
RT = 1 1 so that SRB mode is invoked and the frame gets broadcast on the spanning tree.
L = 2 since RI field is two octets long.
D = 0 the RD octets to be appended will be read left to right.
- Bridge B-1 forwards the frame onto ring R-4 after adding input ring identifier (R-1), own identity (B-1), and output ring identifier (R-4) as RD1 and RD2 subfields. It also modifies the length field (L) accordingly (Figure 14.15b). The bridge identifier in RD2 is kept as 0. Further transmission of this frame is stopped as the link through bridge B-2 is not part of the spanning tree.
- Bridge B-3 also forwards the frame received from ring R-1 onto ring R-2 in the same manner (Figure 14.15c). The frame moves to ring R-3 and then to

the destination station B (Figure 14.15d). In this case the destination station receives only one frame.

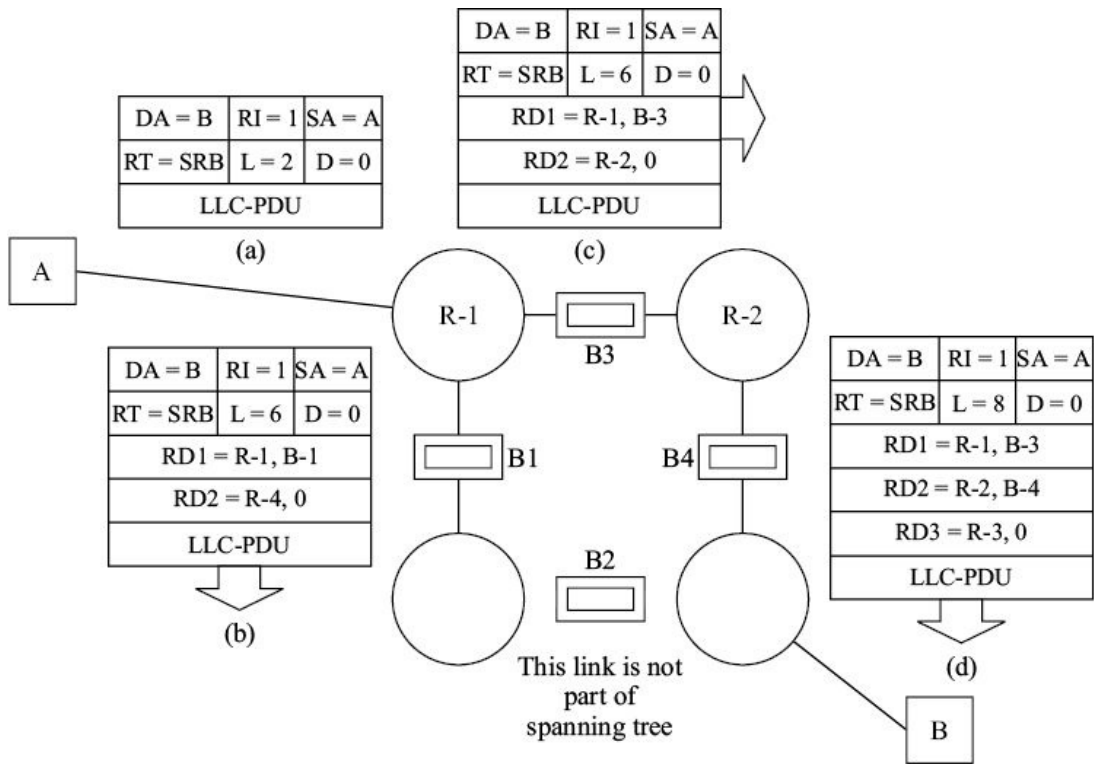


Figure 14.15 SRB on the spanning tree.

- To the received frame, the destination station B replies with ARB frame (Figure 14.16). The ARB process is exactly same as we discussed earlier except that in this case the route discovery frame is being sent by station B to station A. Stations A receives two frames with different routes from B to A, and it selects one for future communication with B.

The advantage of SRB method is that all route broadcast takes place only if the destination station is available.

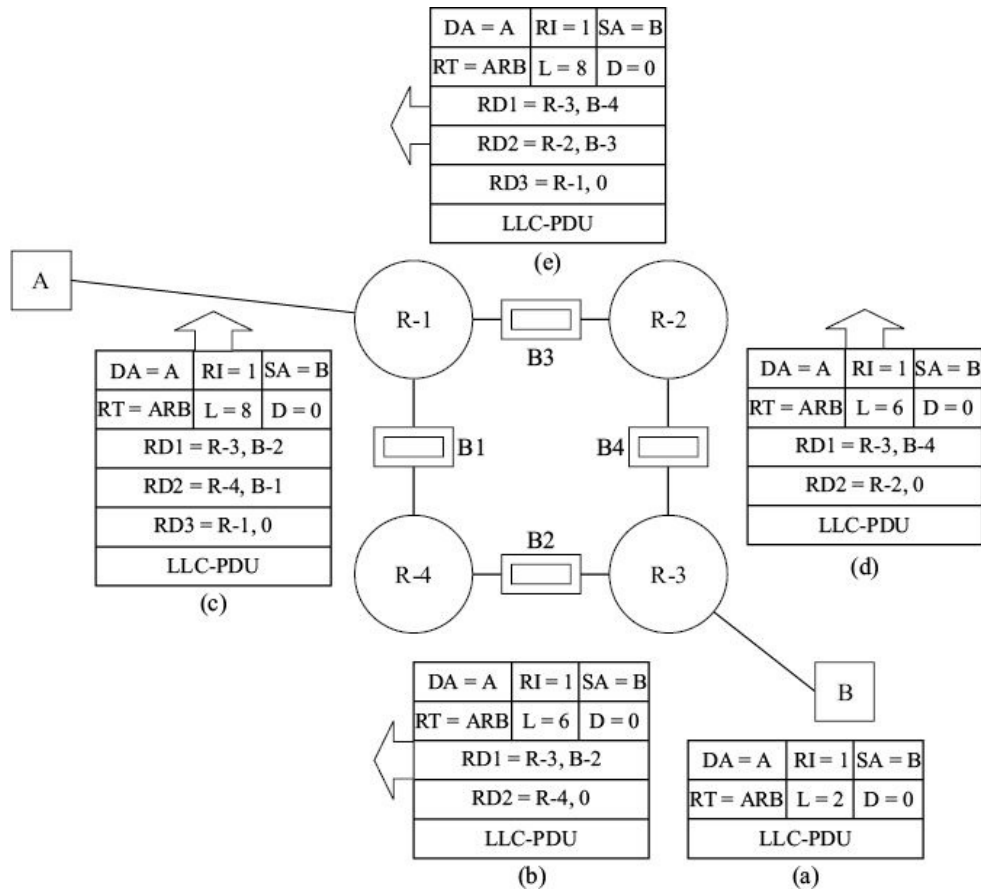


Figure 14.16 ARB by the destination station.

EXAMPLE 14.1 Figure 14.17 shows a bridged token ring network that operates on source routing using single-route-broadcast for route discovery. Assume that all the bridge ports have equal costs. Derive

- the spanning tree.
- all the paths taken by SRB sent by station A for station B.
- all the paths taken by the response from station B to SRB from station A.

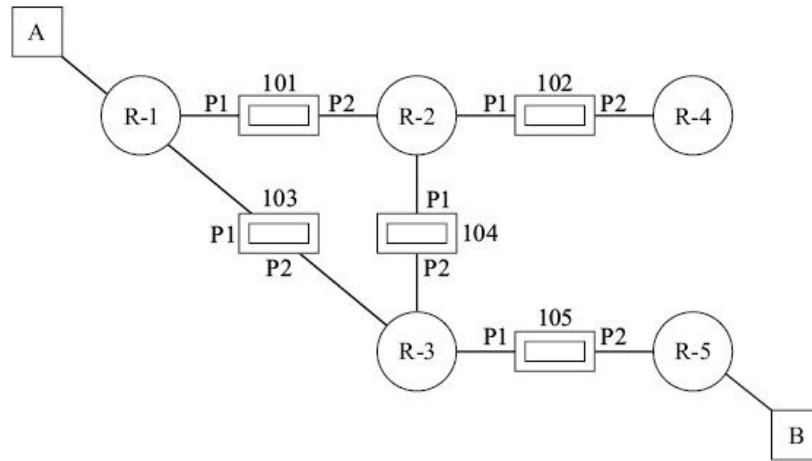


Figure 14.17 Example 14.10.

Solution

(a) Spanning tree is shown in Figure 14.18:

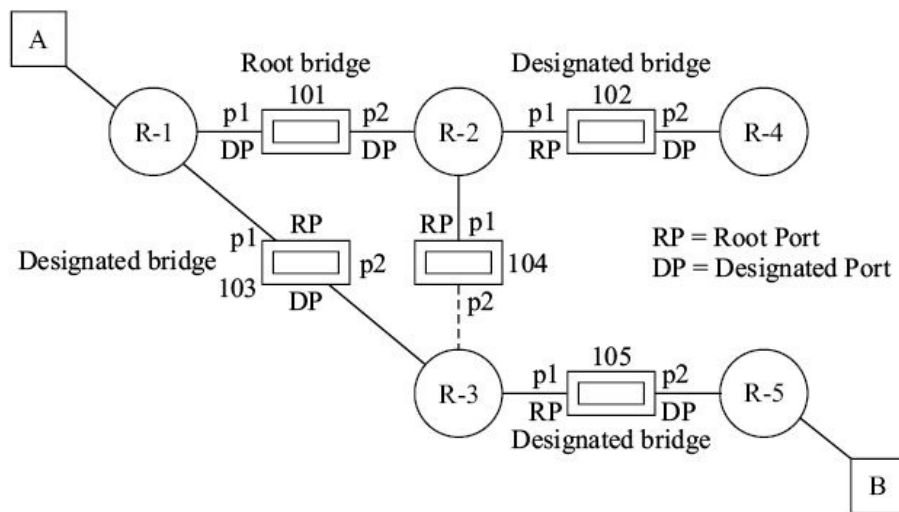


Figure 14.18.

- Root bridge : 101 The bridge with the lowest identifier is chosen.
- Root port of 102 : p1 Root bridge is not accessible through p2.
- Root port of 103 : p1 RPC is lower from this port.
- Root port of 104 : p1 RPC is lower from this port.
- Root port of 105 : p1 Root is not accessible through p2.
- Designated bridge for R-1: 101 R-1 is directly connected to the root bridge.
- Designated bridge for R-2: 101 R-2 is directly connected to the root bridge.
- Designated bridge for R-3: 103 RPC is same for 103 and 104. Therefore, the bridge with lower identifier is chosen as designated bridge.
- Designated bridge for R-4: 102 R-4 is connected to 102 only.
- Designated bridge for R-5: 105 R-5 is connected to 105 only.

- (b) SRB paths from station A to station B.
 - (i) R-1 → 101 → R-2 → 102 → R-4
 - (ii) R-1 → 103 → R-3 → 105 → R-5
- (c) B responds with ARB. The ARB paths from station B to station A are determined considering all the paths without considering the spanning tree.
 - (i) R-5 → 105 → R-3 → 103 → R-1 → A
 - (ii) R-5 → 105 → R-3 → 103 → R-1 → R-2 → 102 → R-4
 - (iii) R-5 → 105 → R-3 → 104 → R-2 → 101 → R-1 → A
 - (iv) R-5 → 105 → R-3 → 104 → R-2 → 102 → R-4

Thus, station A gets two ARB frames from B along with the routes (i) and (iii).

14.7 SOURCE ROUTING BRIDGE VERSUS TRANSPARENT BRIDGE

Table 14.1 compares the features of source routing and transparent bridges.

TABLE 14.1 Features of Source Routing and Transparent Bridges	
Source routing bridge	Transparent bridge
Route is decided by the end stations.	Route is decided by the bridges.
Bridges are not transparent to the end stations.	Bridges are transparent to the end stations.
The source can select an optimum route from a choice of routes.	The spanning tree does not yield optimum routes.
It is possible in theory to do load sharing by judicious choice of routes.	Some of the ports are blocked. Load sharing is not possible through blocked ports.

The frame processing delay is less than that in transparent bridge.

The frame processing delay is more than that in source routing bridge because the forwarding table is to be consulted.

There is considerable route discovery overhead.

The overhead of forwarding a frame to the unknown destination is considerably less as the frame is broadcast on the spanning tree.

In source routing, inactivity timers can be implemented in the end stations so that every end station initiates route discovery procedure after timeout. But this results in considerable traffic overhead.

The periodic update of spanning tree triggered by the root bridge ensures that failed links (ports) are removed from the forwarding tables. Inactivity timer also removes entries from the forwarding table.

14.8 REMOTE BRIDGES

We have so far considered local bridges which interconnect contiguous LANs. Only one bridge is required between two LANs. Consider a situation where two LANs are located at some distance. To interconnect these LANs one bridge will not serve the purpose because the bridge can be located at one of the two LAN locations. The other LAN cannot be extended to the bridge due to

- the distance limitation of the LAN and
- non-availability of transmission link that can support the LAN data rate.

One alternative is to interconnect the LANs using routers and wide area network. We will examine this solution when we study wide area networks. Another possibility is to take a full duplex leased connection from the telephone network operator and connect the LANs using two bridges, one at each end of the leased connection (Figure 14.19). These bridges are called remote bridges and have a port that has data rate and signal levels compatible to the telephone network standard. The usual data rates of this port are multiples of 64 kbps. HDLC (or PPP) protocol is implemented on this port. The other port has the MAC protocol of the local area network. The bridges establish a data link connection between them through the leased circuit and then carry out the bridge operation. HDLC protocol takes care of the transmission errors

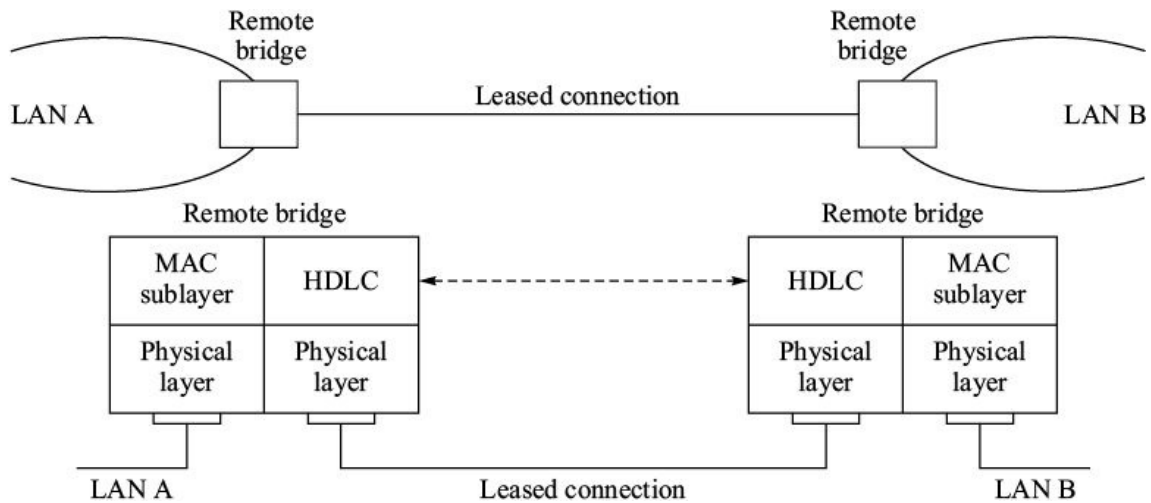


Figure 14.19 Layered architecture of a remote bridge.

of the leased connection. Note that the HDLC protocol implements the MAC frame transport service. The MAC frame is encapsulated in the information field of an HDLC frame using a header and trailer at the transmitting end. At the other end, the MAC frame is delivered after stripping off the HDLC header and trailer.

14.9 LAYER 2 ETHERNET SWITCHES

We have seen three types of Ethernet devices so far—repeater, hub, and transparent bridge. These devices are required when there is need to expand a local area network in terms of the number of stations and their geographic distribution. During 1990s, these Ethernet devices were replaced by Ethernet switches, also called layer-2 switches. We discuss in this section the motivation behind introduction of layer-2 switches in local area networks and their features.

14.9.1 Motivation behind Ethernet Layer-2 Switches

Layer-2 switches were introduced in the network when the required data rate, geographic coverage, and number of stations could not be supported on one Ethernet. Let us understand the issues involved and how layer-2 switches could address these issues.

Collision domain. As shown in Figure 14.20, in case of a repeater or a hub, there is one Ethernet and one collision domain. As the number of stations and transmission rate increase, so does the collisions and eventually a stage comes when the throughput performance deteriorates. At this stage it becomes

necessary to partition the local area network using a bridge.

A bridge has buffer and it uses a store and forward mechanism to send the frames. Store and forward mechanisms does not allow frames on one segment of the network to collide with the frames on the other segment of the network. Therefore, a bridge effectively partitions the network into separate Ethernets, each having its own collision domain. Thus the constraint of supporting large number of stations on one local area network is overcome. But a bridge has

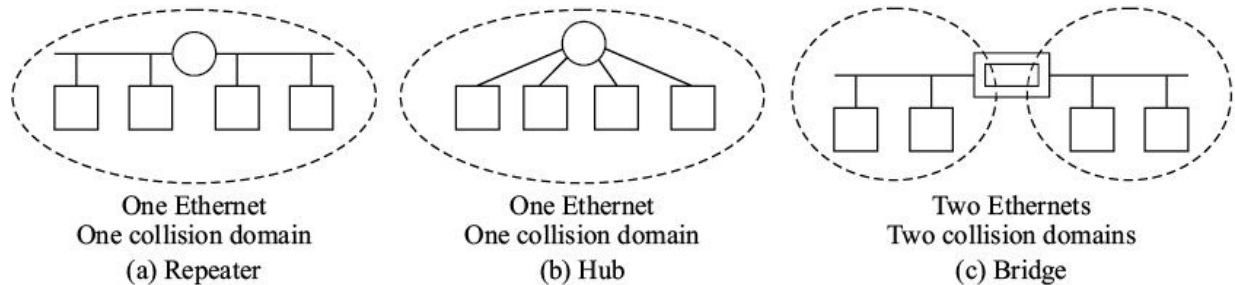


Figure 14.20 Collision domains in a network having a repeater, a hub or a bridge.

limitation of number of ports. It can partition a LAN into a few segments, say two or four, which does not meet the requirement.

Layer-2 switch is functionally a multi-port bridge. It can have number of ports ranging from twelve to a few hundreds. With such large number of ports it is possible to create big collision free networks. Each station is connected to the switch on point-to-point full duplex Ethernet link (Figure 14.21). We saw in the Chapter 11 that such links are collision free. A switch can also support half duplex bus Ethernets on some of its ports.

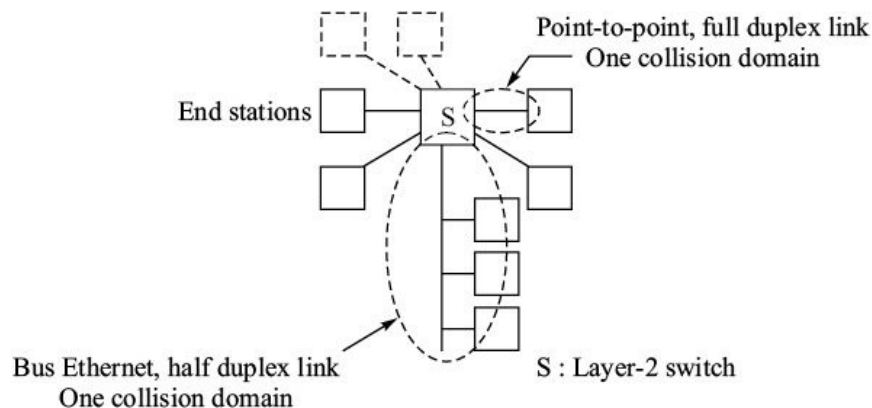


Figure 14.21 Collision domains in a network with layer-2 switch.

Multi-rate adaptation. Repeaters and hubs do not support different speeds at their ports, *i.e.* 10 Mbps Ethernet cannot be interconnected to a 100 Mbps Ethernet using a hub or repeater. A transparent bridge can do this because it has

buffer to store the frames temporarily. But this capability is fully made use of in layer-2 switches. The client machines are connected at 10/100 Mbps ports of the switch (Figure 14.22). The layer-2 switches are interconnected at 100/1000 Mbps. The servers at the back end also connect to the switches at 100/1000 Mbps. All the links are usually full duplex point-to-point and therefore collision free.

Structured cabling. Bus topology of Ethernet was replaced with tree topology using hubs because hubs made structured cabling feasible. All the stations connect to the hub and the hubs are further interconnected to form a local area network. Structured cabling enables minimum cabling changes as the network expands. Layer-2 switches are also based on tree topology and support structured cabling.

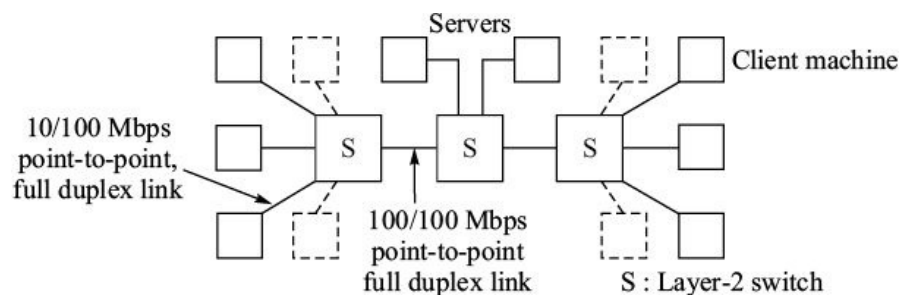


Figure 14.22 Multi-rate adaptation in layer-2 switches.

14.9.2 Latency in Ethernet Layer-2 Switch

When a frame is received on a port of a switch, the source address is added to the forwarding table, if it is not already there. The destination address in the frame is searched in the forwarding table for determining the output port of the switch. The frame is put in the queue for subsequent transmission on that port. Such layer-2 switch is categorized as store and forward switch.

In a store and forward switch, the latency is the time taken from ingress of the frame to its egress from switch. Minimum latency will be when there are no frames in the queue. For a 1500 octets frame at 10 Mbps, the latency comes to over 1 ms. Latency of the switch can be reduced by starting to forward the frame on the output port as soon as the source and the destination addresses have been received. Such layer-2 switch is called cut-through switch and it reduces the switch latency to less than 50 ms.

However, there are several situations where cut-through switching will not work or may not be desirable. For example:

- If the destination port is busy, the frame must be put in the queue, so there is no advantage gained.
- The source and destination ports must be operating at the same data rate for cut-through to work.
- If an error is detected in a frame, a store and forward switch will filter the frame but a cut-through switch will not. Therefore, if error rate is above a defined threshold, the cut-through switches are designed to change over to store and forward mode.

14.9.3 Basic Features of Ethernet Layer-2 Switch

Considering that a layer-2 switch is functionally similar to a multi-port bridge, we can summarize the basic features of layer-2 switches as follows:

- Layer-2 switches operate in the same way as transparent bridges for functions like address learning, filtering, and forwarding frames. They also have forwarding tables like bridges.
- Spanning tree protocol is used to remove multiple paths between two stations.
- Most of the stations are connected to switches on full duplex point-to-point links.
- The network topology is tree like and structured cabling is possible.
- They support rate adaptation, *i.e.* a switch can send a transmission at 100 Mbps on a port of the switch to another port of the switch at 10 Mbps.
- The network is collision free if only point-to-point full duplex links are used.
- A switch can have large number of ports. Several switches can be interconnected to meet the port requirements.
- As compared to a bridge, a switch is faster because several of its functions are implemented in hardware. The current implementation of layer-2 networks use switches instead of bridges.

SUMMARY

LAN bridges are used to interconnect different LANs at MAC sublayer. Bridges are different from repeaters which interconnect two segments of the same LAN at physical layer. There are two types of bridges, transparent bridges and source routing bridges. Transparent bridges are used for Ethernet LANs and source

routing bridges are used for token ring LANs. Interconnecting two different types of LANs is usually done using a router.

A transparent bridge learns the location of a station by looking at the source addresses of the frames that hit its ports and creates a forwarding table. It forwards the frames from one LAN to another based on this forwarding table. To avoid broadcast storms and circulating frames, frames are forwarded along a spanning tree that connects all the LANs to the root bridge. The spanning tree is created and updated dynamically using a control frame called Bridge Protocol Data Unit (BPDU).

Unlike transparent bridges, the source routing bridges merely forward the frame based on the path specified by the source in the body of each frame. Each station discovers path to the desired destination using one of the two approaches, All-Route-Broadcast (ARB) or Single-Route-broadcast (SRB).

A bridge has limitation of the maximum number of ports it can have. A LAN switch or layer-2 switch, functions like a transparent bridge but can have very large number of ports. Layer-2 switch has a forwarding table which is created and updated in the similar manner as in a transparent bridge. Stations are usually directly connected to layer-2 switch through full duplex links. Switches are very fast and support rate adaptation, *i.e.* a switch can send a transmission from a 100 Mbps port to a 10 Mbps port.

EXERCISES

1. Draw the spanning tree for the network consisting of LANs A, B, C, D, and E, bridges 101 to 105 (Figure E14.23). Identify the root and designated ports.

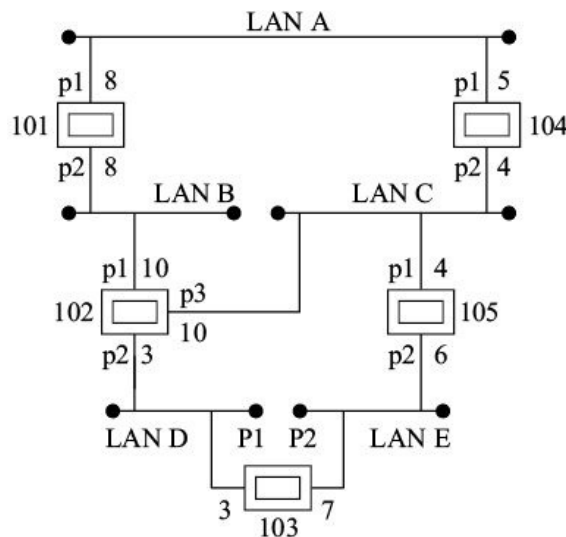


Figure E14.23.

2. Station X sends a route discovery frame with all-rout-broadcast directive for station Y (Figure E14.24). Indicate the route designator subfields of the multiple copies of the frames received by Y. Which is the minimum hops route?

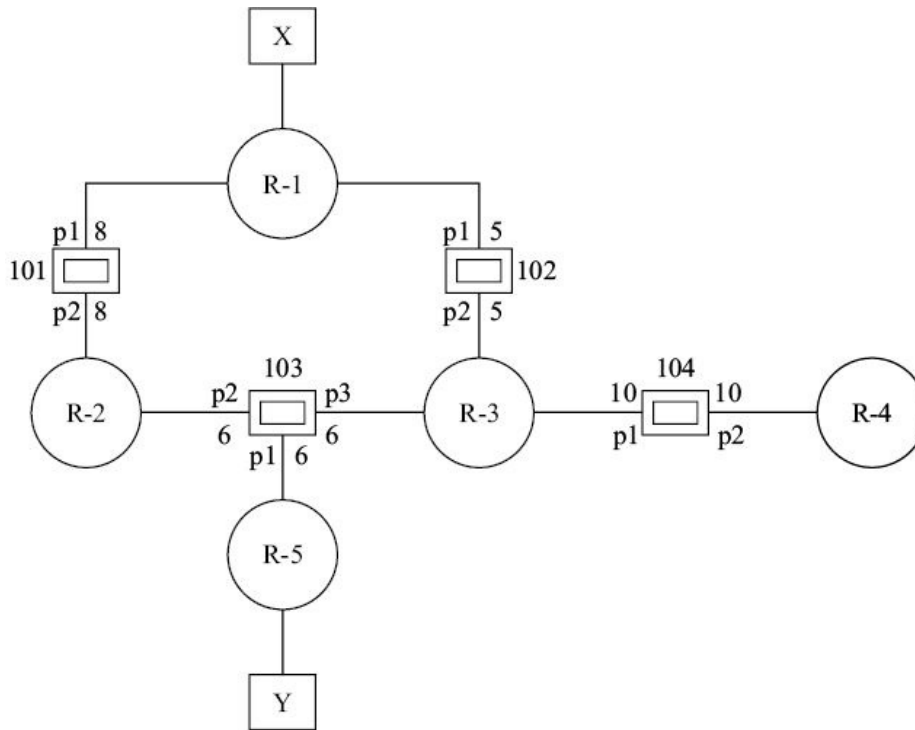


FIGURE E14.24.

3. The bridged token ring network in Exercise 2 operates on source routing using SRB for route discovery. The bridge ports have costs as indicated. Derive
- the spanning tree.
 - all the paths taken by SRB by station X for station Y.
 - all the paths taken by the response from station Y to SRB from station X.
4. Consider the arrangement of transparent bridges as shown in Figure E14.25. Initially all the forwarding tables of the bridges are empty. Write the forwarding table of each bridge after the following transmissions.
- A sends a frame to C.
 - C sends a frame to A.
 - D sends a frame C.

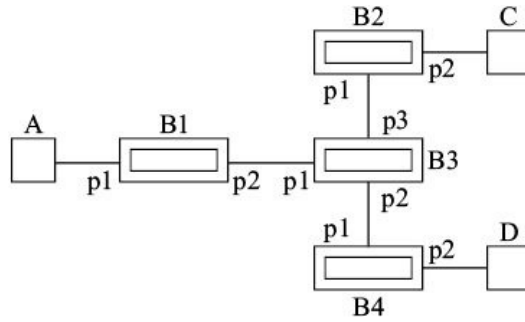


Figure E14.25.

5. Suppose a transparent bridge has two of its ports connected to the same Ethernet LAN. How will the bridge detect and correct this?
6. In Figure E14.26, B1 and B2 are transparent bridges having empty forwarding tables. A sends a frame to itself. What will happen if
 - (a) the bridges first take the actions required for forwarding and then update the source address entry in the forwarding table?
 - (b) the bridges first update the source address entry in the forwarding table and then take the actions required for forwarding?

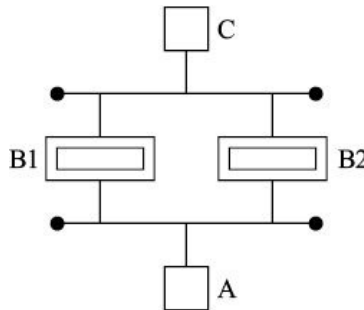


Figure E14.26.

7. Determine the spanning tree of the extended LAN shown in Figure E14.27 consisting of transparent bridges 10, 20, and 30. Assume that the all port costs are equal.

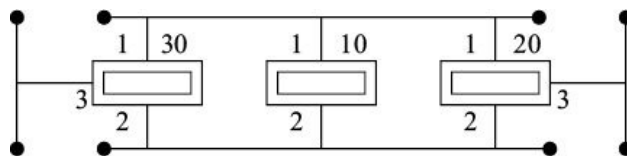


Figure E14.27.

15

Network Layer

Local area networks, as the name suggests have limited geographic coverage and scalability. In this chapter we introduce switched data networks that overcome these limitations. We examine three types of networking technologies—circuit switching, message switching, and packet switching. Packet switching is covered in considerable detail and two very important packet switching concepts, datagram routing and virtual circuit routing, are introduced.

Packet switching function is implemented in the network layer of the OSI reference model. After a brief description of the purpose, functions, and services of the network layer, we look at the end-to-end layered architecture of the packet switching data networks. In this chapter, we lay foundations for the layer 3 protocols which we will discuss in detail in the next two chapters.

15.1 WIDE AREA NETWORKS

The local area networks that we studied in the last three chapters have limited geographic coverage and limited scalability in terms of number of stations. These limitations are due to their inherent technology that is based on broadcast type of communication. It is possible to engineer data networks that can span cities, countries and continents. Such data networks are called Wide Area Networks (WANs). The technology used in these networks does not have distance or scalability limitations.

A wide area network consists of collection of network nodes which provides resources for interconnection of the end-systems (Figure 15.1). The network provides two types of services to the end systems:

- Fixed transport service
- Switched service

In fixed transport service, the network resources are allotted to the end systems for their exclusive use on long term basis. Leased point-to-point circuit of say 64 kbps is a fixed transport service provided by the network.

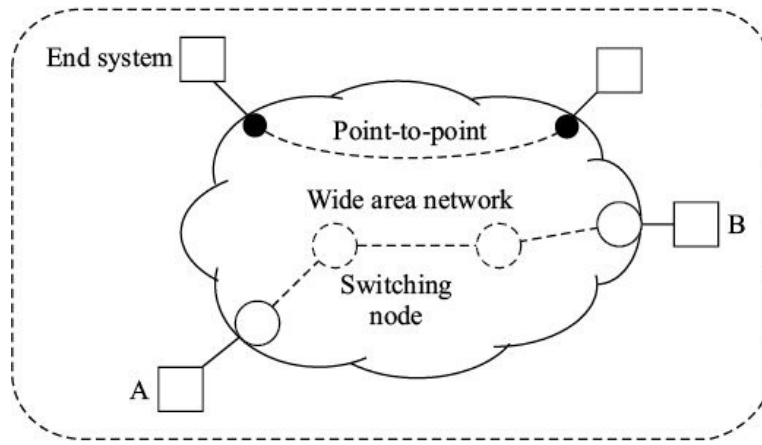


Figure 15.1 Data network.

In switched service, the network resources are dynamically used by the end systems. Telephone service for voice communications is an example of switched service provided by the telephone network.

15.1.1 Switched Data Networks

A switched data network consists of an interconnected collection of nodes. The interconnecting link between two nodes is called *trunk* (Figure 15.2). Data units are transmitted from source to destination by being routed through these nodes. For example, data units from end system 4A intended for 6F are sent to the entry node 4, switched to the transit node 5 and then to the exit node 6. Each end system is identified by a unique address to facilitate routing of the data units. Usually the address of an end system also contains identification of the node to which the end system is attached.

Two major networking requirements that drive use of switched networks are flexible topology and resource sharing.

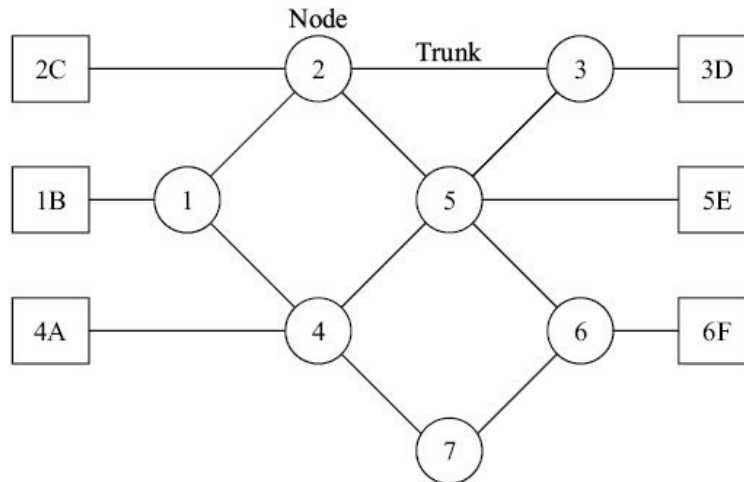


Figure 15.2 Switched data network.

Flexible topology. Switching enables delivery of information presented at one access node of the network to a variety of destinations which can be selected by the users. Thus, switching provides a flexible interconnection topology.

Resource sharing. The network resources are available to all users of the network. As and when a user requires the services of the network, the resources are allocated to it.

15.1.2 Types of Switched Data Networks

There are two basic techniques for switching data units at the nodes:

- Circuit switching
- Store-and-forward switching.

Circuit switching is used primarily for voice communications but it can be used for data networks. The data networks, on the other hand, are based on store-and-forward switching. VOATM (Voice-Over-ATM) and VOIP (Voice-Over-IP) are new technologies that enable even the voice signals to be carried on store-and-forward networks.

15.2 CIRCUIT SWITCHING

The circuit switched data network consists of circuit switching nodes interconnected by trunk circuits. The circuit switching nodes carry out interconnection of the incoming trunk circuits and outgoing trunk circuits to

establish an end-to-end transmission path, called connection. All the data units are sent on the connection so established. The connection is released by the end systems after availing the network services.

Examples of circuit switched data networks are:

- ISDN service of the telephone network provides circuit switched data communication service through the network meant primarily for voice communications.
- The telex network, now obsolete, was circuit switched data network. ITU-T Recommendation X.21 is for the circuit switched data network.

15.2.1 Operational Phases in Circuit Switching

Transfer of data units through a circuit switched network involves the following three operational phases:

- Connection establishment phase
- Data transfer phase
- Connection release phase.

Connection establishment phase. The call originating end system sends a connection request with the destination address to the entry node. The entry node builds up a path by cross connecting one of the outgoing trunk circuits in the direction of the destination end system (Figure 15.3). The address information is transferred to the next node where again a cross-connection between the incoming and outgoing trunk is made. This process is repeated at each intermediate node and at the exit node which serves the destination end system. The exit node sends in incoming call indication to the destination end system which returns a call acceptance. The network confirms establishment of connection to the call originating end system.

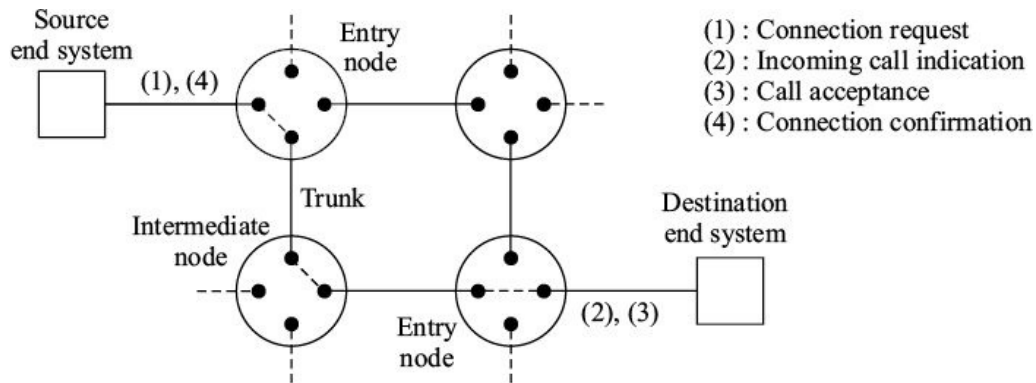


Figure 15.3 Circuit switched data network.

The network resources allocated for the purpose of building up the connection are for the exclusive use of the end systems for transporting their data. Trunks may be real (metallic pairs, FDM channels, PCM channels) or virtual. If they are virtual, they must always be available to their users immediately when the data units are to be transmitted.

Data transfer phase. After the connection confirmation is received, data transfer can begin on the connection. The basic features of the data transfer service are as follows:

- The same connection is used by both the end systems to communicate, *i.e.* the connection is bidirectional.
- The network nodes cannot store, even temporarily, the data units. Therefore, the data rates at the source and destination are the same.
- Address of the destination is specified only once during call set up. All subsequent data units are transmitted on the path already established.
- The nodes do not carry out any form of error control.

Connection release phase. The connection is released at the request of the end-systems and after the release, the network resources that were engaged for setting up of the connection are also released.

15.2.2 Delays in Circuit Switched Data Network

Connection establishment in a circuit switched network involves certain set-up time as shown in Figure 15.4. It includes cross-connection delays at the nodes and connection request propagation delays. Once the connection is set up, user data transfer involves only propagation delay and it is constant. There is almost no delay in the nodes during the data transfer phase.

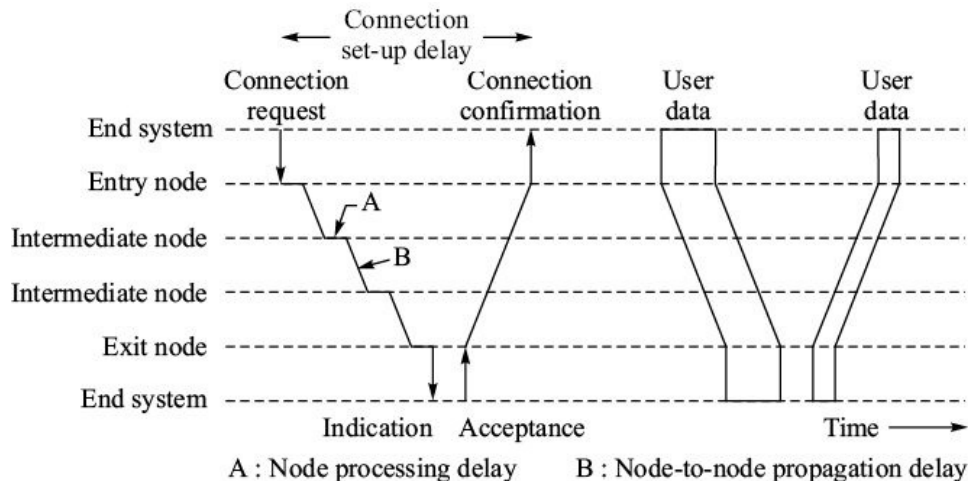


Figure 15.4 Data delivery delays in circuit switched data network.

Data units are transmitted immediately and incrementally as soon as they are presented to the network. Since the delivery delay is constant, the time relationships of the data units and their sequence of transmission are maintained.

During peak traffic hours, connection set-up delay may increase because the network resources may not be free but once the connection is established, there is no increase in delivery delay through the network as the network resources are allotted for the exclusive use.

Service features of the circuit switched data network are summarized below:

- There are connection establishment, data transfer, and connection release phases.
- There is connection set-up delay that increases with traffic.
- Destination address is specified only during connection establishment phase.
- Data delivery delay constant irrespective of traffic. Delivery delay is minimal.
- Data rates at the source and destination are same.
- Time relationships and order of data units are maintained.
- There is no error control within the network.

15.3 STORE-AND-FORWARD DATA NETWORKS

In store-and-forward switching, a data unit is accepted by the network node, stored, put in a queue, and when its turn comes, it is forwarded to the next node. Data networks employing this basic technique can be of two types:

- Message switched network
- Packet switched network.

In message switched network, the data unit is full message and it is switched at the network nodes. In packet switched network, the data unit is a small chunk of data bytes, called data packet or simply packet. The message to be transmitted is divided into packets by the end systems and these packets are switched through the network. Although both the approaches employ store-and-forward switching, their service features are quite different.

15.3.1 Message Switching

A message switched network consists of store-and-forward nodes interconnected by trunks. Each node is equipped with a storage device wherein all incoming messages are temporarily stored for onward transmission. A message along with the destination address is sent from node to node till it reaches the destination.

Figure 15.5 shows a message switched network consisting of four nodes. When end system A wants to send a message to end system B, it sends the message along with the source and destination addresses to node 1.

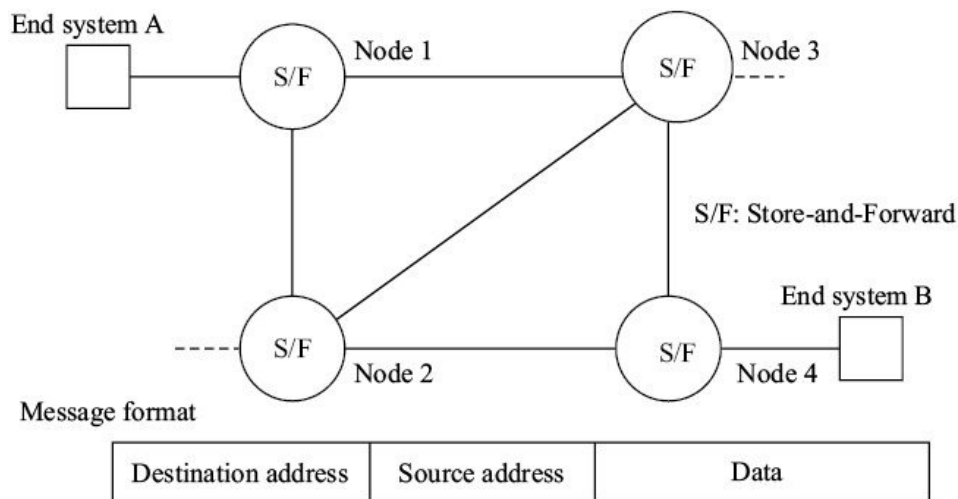


Figure 15.5 Message switched network.

A forwarding table is maintained at each node. It contains entries indicating destinations and the corresponding outgoing ports from the node. Node 1 consults the forwarding table for the outgoing port to the destination B and puts the message in the queue for node 2. When the message is received at node 2, it is again put in a queue of messages awaiting transmission to node 4. When node 4 receives the message, it delivers the message to the destination.

The basic features of store-and-forward message switching are as follows:

- The store-and-forward service is unidirectional. If an end system is required to send an acknowledgement for the received message, the acknowledgement is treated as a separate message.
- Since the message is stored in a bulk buffer (secondary storage) at each stage of transmission, each node-to-node transfer is an independent operation. The trunks can operate at different data rates. Even the source and destination end systems can operate at different data rates.
- Each message carries destination and source addresses and is treated as independent entity by the network.

Delivery delay. Figure 15.6 shows the delays associated with delivery of a message through a message switched network. The message passes through the entry node, two intermediate nodes, and finally through the exit node to arrive at the destination. Message delivery time is the sum of the following components:

- Propagation time of each link (A)
- Node delay (B)
- Transmission time of the message (C).

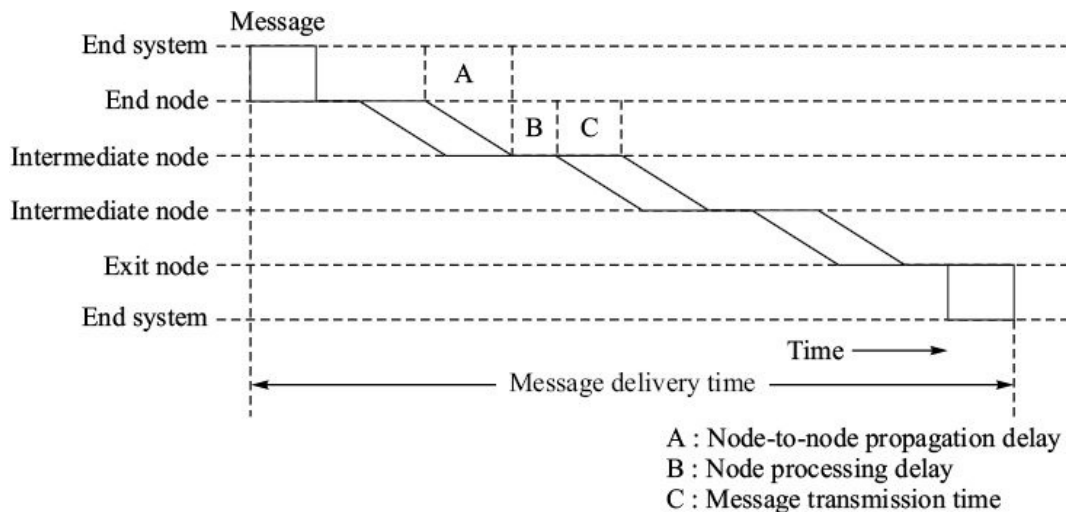


Figure 15.6 Delivery delay in message switched data network.

Propagation time of the message from the end system to the entry node, and from the exit node to the destination end system are assumed negligible. Node delay is composed of processing delay and waiting time in the queue. Transmission time is determined by the message size and the data rate of the

link. Total time required to deliver the message is linear sum of all these components as they occur in a sequential manner. As traffic increases, there is increase in the message delivery time because queues get longer.

The main service features of the message switched network are summarized as follows:

- Connection establishment and release phases are missing.
- Destination address is specified on each message.
- Delivery delay is significant and random.
- Delivery delay increases with traffic.
- Data rates at the source and destination need not be same.
- It provides unidirectional message transmission service.

15.3.2 Packet Switching

In message switching, there is significant delivery delay because a message is transmitted by a node to the next node only after it has been completely received. This delay can be reduced by dividing the message into smaller data packets and then transmitting each packet as an independent entity (Figure 15.7).

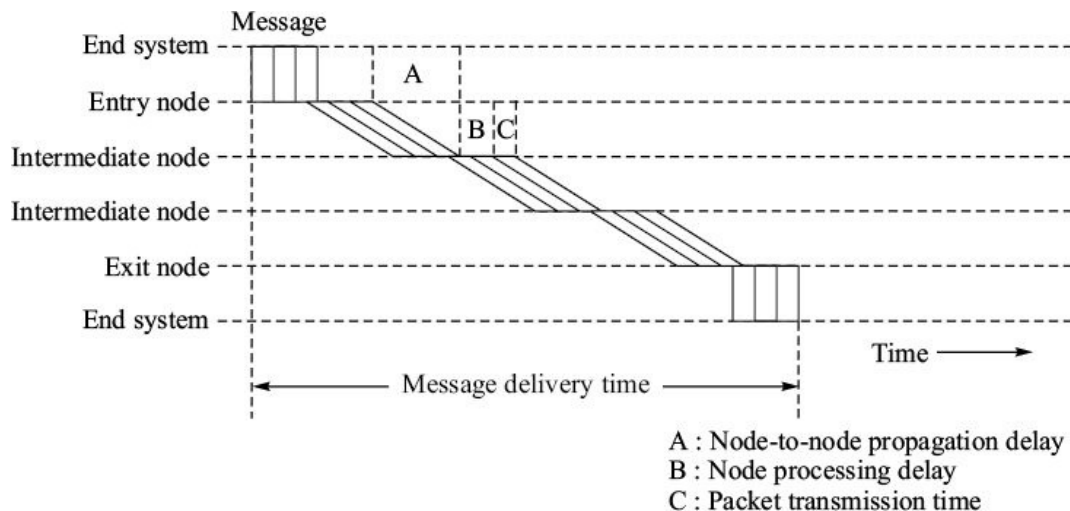


Figure 15.7 Data transfer delays in packet switched data network.

The reduction in delivery delay is on two accounts:

- Reduced processing time in the node
- Reduced message transmission time due to pipelining.

The processing time of the packet switching node is less than that of a

message switching node because the packets are stored in the primary memory of the nodes instead of secondary memory. Access time of the primary memory is much less than the access time of the secondary memory.

Packets are transmitted as soon as they become available for transmission. It is not necessary to wait for all the packets that comprise a message. Conversion of messages into packets and vice versa is usually done by the end systems themselves. All the data networks today are based on packet switching technology.

Basic operation. A packet switched network consists of interconnected packet switching nodes (Figure 15.8). The packet switching nodes are based on store-and-forward technology.

The end system sends data packets containing addressing information in their header to the entry node. The packet switch uses the addressing information to determine the outgoing port and puts the packet in the queue for further transmission. The addressing information in the header depends on the type of packet switching approach deployed, datagram, or virtual circuit. Addressing information may not be the physical addresses of the end systems (e.g. 1A, 2B, 3C, and 4D). We discuss these approaches in the next section.

Note that:

- Statistical multiplexing of data packets is carried out on all the links, *i.e.* packets belonging to different sources and meant for different destinations share the same link.
- An end station can have several simultaneous data communication sessions with different end systems.

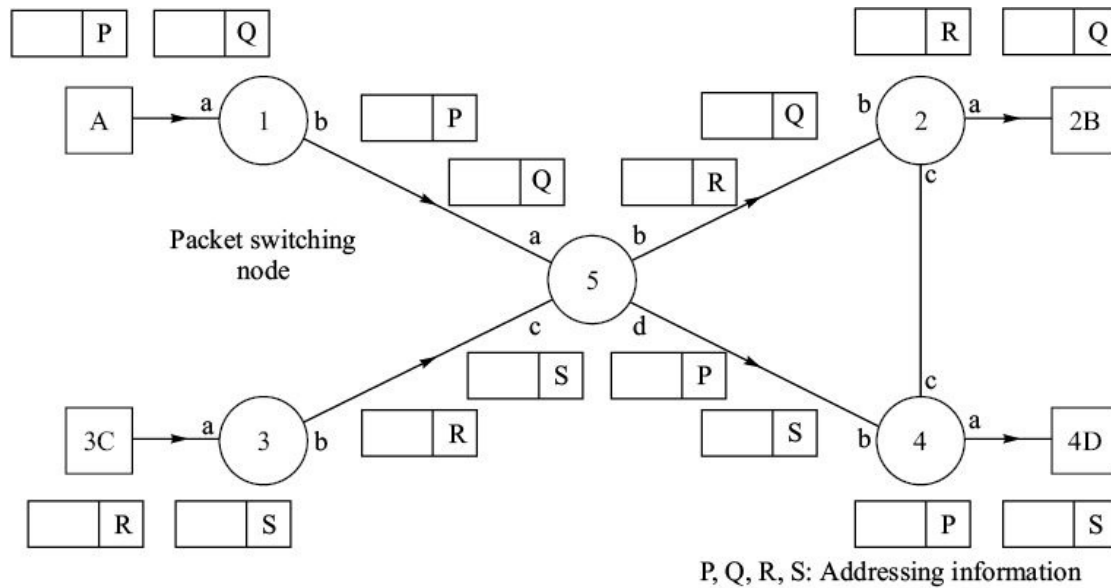


Figure 15.8 Packet switched network.

15.4 TYPES OF PACKET SWITCHED DATA NETWORKS

Packet switched data networks can be of two types:

- Datagram switching
- Virtual circuit packet switching.

Datagram switching network is similar to message switching network. Each data packet is an independent entity. It carries source and destination addresses in the header. The packet switching nodes are store-and-forward nodes and they route the data packets to the destination based on the address on the packets. A packet with source and destination address is called a *datagram*.

Virtual circuit packet switching network is similar to circuit switching network. A virtual connection is set up between the communicating end systems. All the data packets carry identity of the virtual connection they belong to and follow the path of the virtual connection across the packet switching network.

15.4.1 Datagram Switching Network

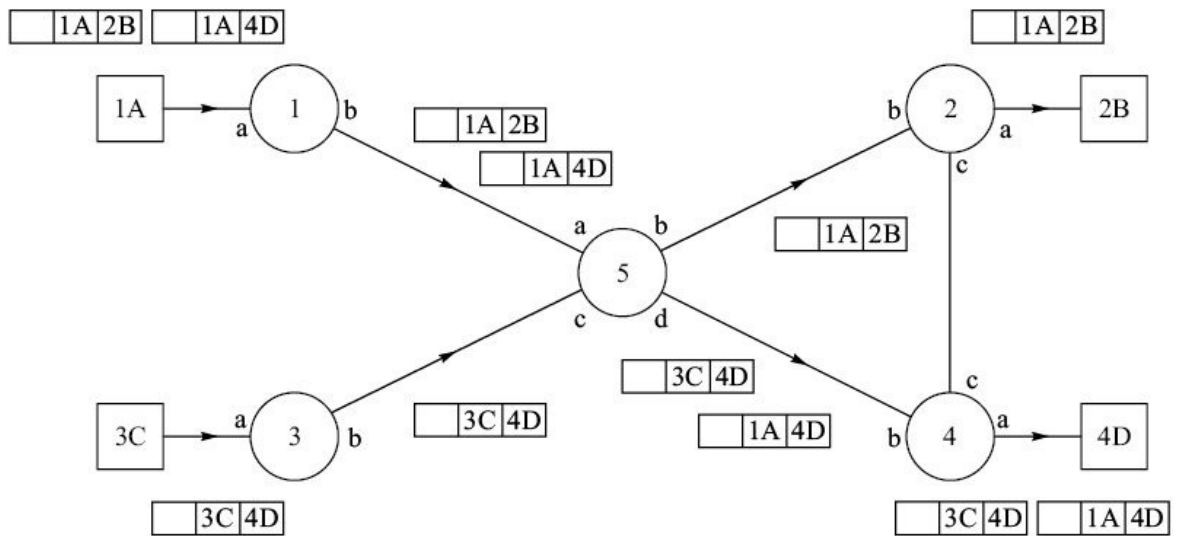
In the datagram approach to packet switching, each packet switching node maintains a forwarding table that indicates the port through which an end system

can be reached. When a data packet, carrying the destination address, is received by a packet switching node, it consults the forwarding table to determine the outgoing port towards destination address and puts the packet in the queue at that port. The process is repeated at each node till the packet reaches the destination. All the datagrams are individually routed across the network by the nodes to their destinations. The nodes are, therefore, called routers.

Let us examine the operation in some detail. Consider that end system 1A has two packets to send to end systems 2B and 4D (Figure 15.9).

- End system 1A sends the two packets addressed to end systems 2B and 4D to node 1.
- Node 1 forwards the packets through its port b after consulting its forwarding table.
- The packets are received by node 5. It consults its forwarding table and forwards the packet addressed to 4D through its port d. The second packet addressed to 2B is forwarded through its port b.
- Node 2 receives the packet through its port b and forwards it through its port a to the destination 2B.
- Node 4 receives the packet through its port b and forwards it through its port a to the destination 4D.

Since a network may consist of a very large number of nodes, the forwarding tables tend to be big. It is quite possible to divide the network in several domains so that the nodes belonging to one domain are required to maintain forwarding table for the nodes within their domain. Inter-domain traffic is routed through specified nodes. If a new node is added in a domain, the forwarding tables in that domain only will need to be updated.



Forwarding table at node 1		Forwarding table at node 2		Forwarding table at node 3		Forwarding table at node 4		Forwarding table at node 5	
To	O/G port	To	O/G port	To	O/G port	To	O/G port	To	O/G port
1A	a	1A	b	1A	b	1A	b	1A	a
2B	b	2B	a	2B	b	2B	c	2B	b
3C	b	3C	b	3C	a	3C	b	3C	c
4D	b	4D	c	4D	b	4D	a	4D	d

O/G port: Outgoing port

Figure 15.9 Routing of datagrams.

Routing of datagrams. Creating and maintaining forwarding tables is a very important task in datagram switching.

- The forwarding tables must be consistent.
- The forwarding must reflect the current topology of the subnetwork.
- There should not be endless routing loops.
- If a link or node is down, alternate routes need to be defined immediately.

Before we proceed further, we need to distinguish the two interpretations of the term ‘routing’. Routing is commonly interpreted as the process of deciding and forwarding a packet along a route. We have been using the term ‘routing’ so far as per its commonly understood meaning. In data networks, routing refers to the process of creation and maintenance of the forwarding tables. The act of switching packets from one port to another is simply switching or forwarding but not routing. Routing protocol is the protocol for learning network topology and updating network topology changes in the forwarding tables. Examples of

routing protocols are RIP, OSPF and BGP. This section on routing, therefore, forms base for Chapter 18, Routing Protocols.

The network layer protocols of layer 3, on the other hand, define format of user data packets and the process of their exchange. Examples of network layer protocols are X.25 and Internet Protocol (IP). We will learn about these network layer protocols in the next two chapters.

There are two approaches for creation and maintenance forwarding tables—static routing and dynamic routing.

Static routing. In static routing, the forwarding tables are preconfigured at the time of creation of the network or when a new node is added to the network. The network administrator updates the forwarding tables periodically or as and when required, *e.g.* when a link or node is down, or when congestion arises in some part of the network. A simplified static routing table is shown in Table 15.1. It corresponds to node 5 of Figure 15.9. Note that on some of the ports, the arriving packets should not carry certain destination addresses. For example, port b of node 5 should not receive a packet with destination address 2B. Therefore, such packets are not forwarded by node 5.

Dynamic routing. In static routing, the forwarding tables become outdated very soon and need frequent updates. As the size of network grows, their maintenance becomes a very cumbersome process. Dynamic routing addresses these issues by automatically creating and updating the forwarding tables.

In dynamic routing, the nodes exchange routing information using a protocol, called routing protocol. Some examples of common routing protocols are RIP, OSPF, and BGP. A routing protocol enables determination of network topology and creation of forwarding tables that indicate the best paths to the destinations.

The routing protocols used in packet switched data networks are based on two fundamental algorithms:

- Distance vector algorithm
- Link state algorithm.

TABLE 15.1 Routing Table of Node 5 (Figure 15.9)

Incoming port	Destination	Outgoing port
		b

For all such situations when a packet is lost or discarded by the network, the transport layer in the end systems is responsible for recovery of the lost data.

Congestion in a network can result in deadlocks. To understand this, consider a network consisting of only three nodes A, B, and C (Figure 15.10). Suppose each node has finite buffer and each buffer is full with packets to be sent to the other nodes. In order for node A to be able to send a packet to node B, the latter must have a free buffer to store the packet. To have a free

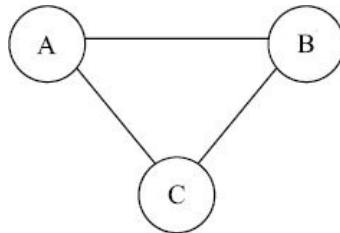


Figure 15.10 Congestion and deadlock.

buffer, B needs to send a packet to node C. Since node C is also full and also cannot send a packet to node A to create a free buffer, movement of packets is not possible in the network. In other words, there is a deadlock. There can be several ways to avoid and overcome deadlocks. One simple way is to discard the packets. Another way is to stop inflow of packets when the buffers reach a threshold.

Out of sequence packets. Dynamic routing allows for automatic network reconfiguration when there is any link or node failure. If there is congestion in any part of the network, alternative routes can be provided. Therefore, it is quite possible that packets pertaining to the same communication session take different routes and arrive at the common exit node out of sequence. In datagram switching, the data packets do not bear any sequence number. Therefore, the exit node has no means of knowing that the packets are out of sequence and delivers them to the destination as they are received.

Before we take up virtual circuit packet switching, let us review quickly the features of datagram switching as follows:

- There is only data transfer phase. No end-to-end connection is set up.
- All packets carry source and destination addresses.
- There is finite and fluctuating delay in delivery of packets.
- There can be packet loss due to errors and discarded packets.
- The packets may be received out of sequence.

- Datagram service is not reliable in the sense that there is no acknowledgement for the received packets.
- Datagram service is best effort service. The end systems generating packets have to make their own efforts to recover from non-delivery or disordering of packets.
- Forwarding tables are required for switching the packets. Forwarding tables need to be regularly updated to account for failures, changes and congestion in the network.

15.4.2 Virtual Circuit Packet Switching

In the virtual circuit approach, a virtual connection is established through the network and all the data packets are transported on the virtual connection. Unlike the datagram approach, the nodes do not make the routing decision for each packet. It is made once for all the packets at the time of establishing the virtual connection.

To understand how the virtual connections are set up and released between two end systems, consider a five node packet switching network with end systems 1A, 2B, 3C, and 4D as shown in Figure 15.11. Suppose end system 1A has some data packets to send to end system 4D. Like circuit switching, virtual circuit packet switching also has three phases, connection establishment phase, data transfer phase, and connection release phase. End system 1A must go through these phases.

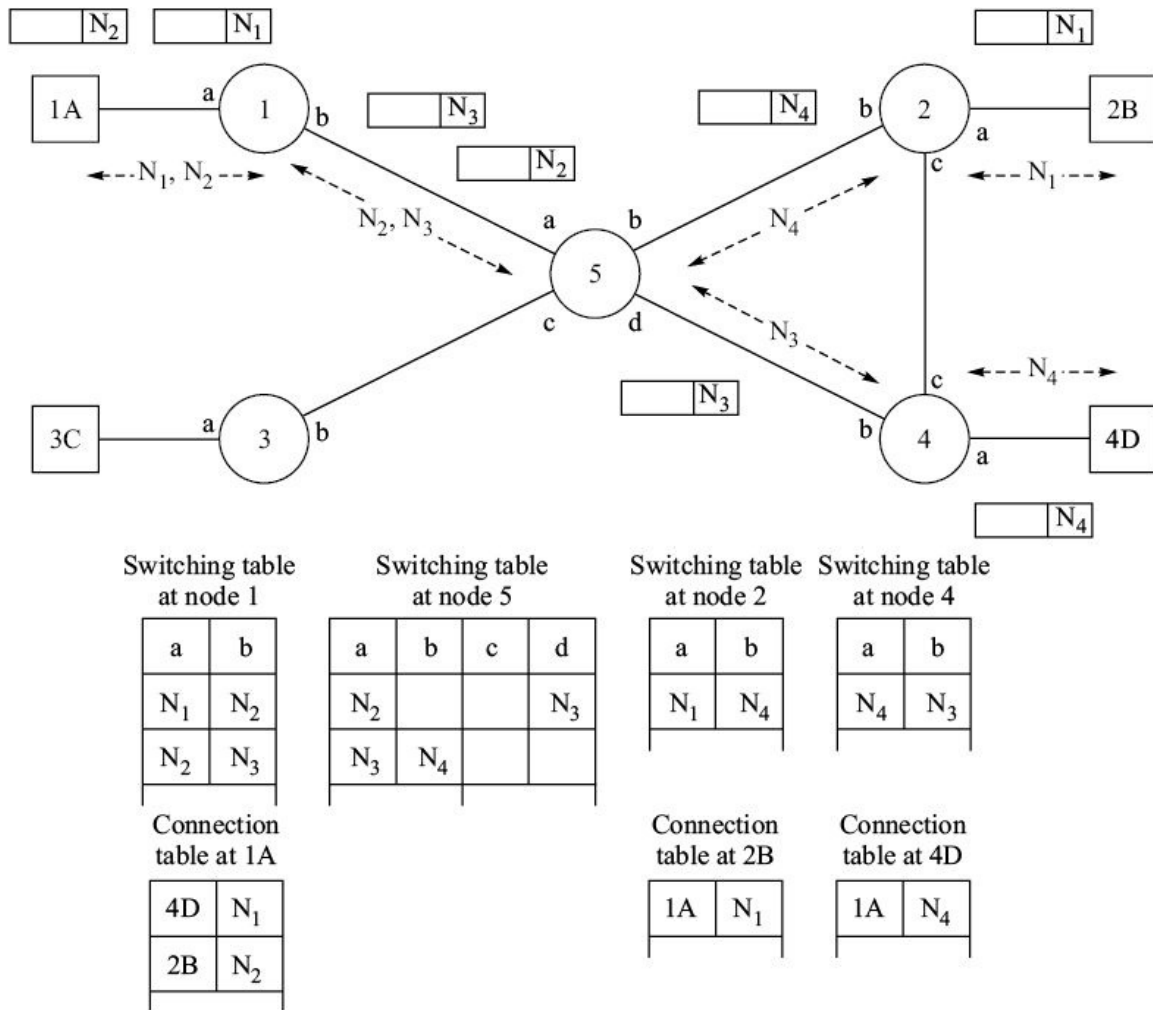


Figure 15.11 Virtual circuit packet switching.

Connection establishment phase. It has taken place as follows:

- End system 1A sends a CONNECT REQUEST packet to node 1 specifying the destination address 4D and the source address 1A. The CONNECT REQUEST packet also specifies a connection identifier (N₁) that is later used for identifying the packets meant for the particular destination. Thus the CONNECT REQUEST is essentially:
 “Connect 1A to 4D. Logical channel identifier is N₁.”
 End system 1A keeps a record of this connection (destination address and identifier N₁) in its connection table, where it maintains records of all the active connections.
- On receipt of the CONNECT REQUEST packet, node 1 examines the destination address specified in the packet and works out the outgoing link

towards the destination using its routing table.¹ Node 1 selects a 'free' identifier N_2 for this link to node 5. Node 1, then, sends a modified CONNECT REQUEST packet to node 5:

“Connect 1A to 4D. Logical channel identifier is N_2 .”

All the future data packets of the connection being established shall bear identifier N_2 while traversing the link between node 1 and node 5. Node 1 maintains a switching table in which it maintains records relating the logical channel identifiers and respective ports (Figure 15.11). It makes a new entry in the switching table for the new identifiers N_1 and N_2 .

- On receipt of this packet, node 5 works out the further route and forwards the packet to node 4 using another identifier N_3 . Node 5 also updates its switching table.
- On receipt of the packet, node 4 forwards this packet to destination 4D. It does so using another free identifier N_4 .
- If 4D decides to accept the call, it returns an ACCEPTANCE packet to node 4. It uses the identifier N_4 already being used for this link of the connection. There is no need to specify the addresses now. The logical identifier N_4 suffices.
- Node 4 sends this ACCEPTANCE packet to node 5 using identifier N_3 .
- Node 5 forwards this packet to node 1 using identifier N_2 .
- On receipt of the ACCEPTANCE packet, node 1 forwards the packet to 1A confirming establishment of the connection to the destination. The confirmation bears the identifier N_1 .

Thus, in the connection establishment phase, a virtual path to the destination is finalized and a confirmation is received from the destination. The connection is virtual in the sense that a record relating the identifiers and ports has been created in the switching table of each intervening node (Figure 15.11). The end systems maintain connection tables which contain the addresses of end systems at the other end of the connections and the respective connection identifiers.

Data transfer phase. After the connection is established, 1A can send its data packets to 4D. Each data packet bears the identifier N_1 . 4D uses identifier N_4 on its packets for 1A. Destination address is not needed in the data packets. When a network node receives a data packet, it looks up the switching table and sends

the packet to the port as indicated therein. It also gives to the packet a new identifier as indicated in the switching table. The data packets bear sequence numbers. Flow control and acknowledgement mechanisms can be readily implemented.

Connection release phase. The virtual connection is released when 1A sends a CLEAR REQUEST packet. This packet also bears the connection identifier N_1 . End system 4D confirms connection release by sending CLEAR CONFIRMATION packet. With the exchange of these packets, the respective entries in the connection and switching tables are erased.

It may be noted that an end system can operate several connections simultaneously by using different connection identifiers. This is illustrated in Figure 15.11. 1A is operating two connections, one to end station 2B and the other to end station 4D.

EXAMPLE 15.1 Figure 15.12 shows six nodes of a virtual circuit packet switching network. The switching tables maintained at nodes Q and R are shown. Determine the various connections with their paths working on the network.

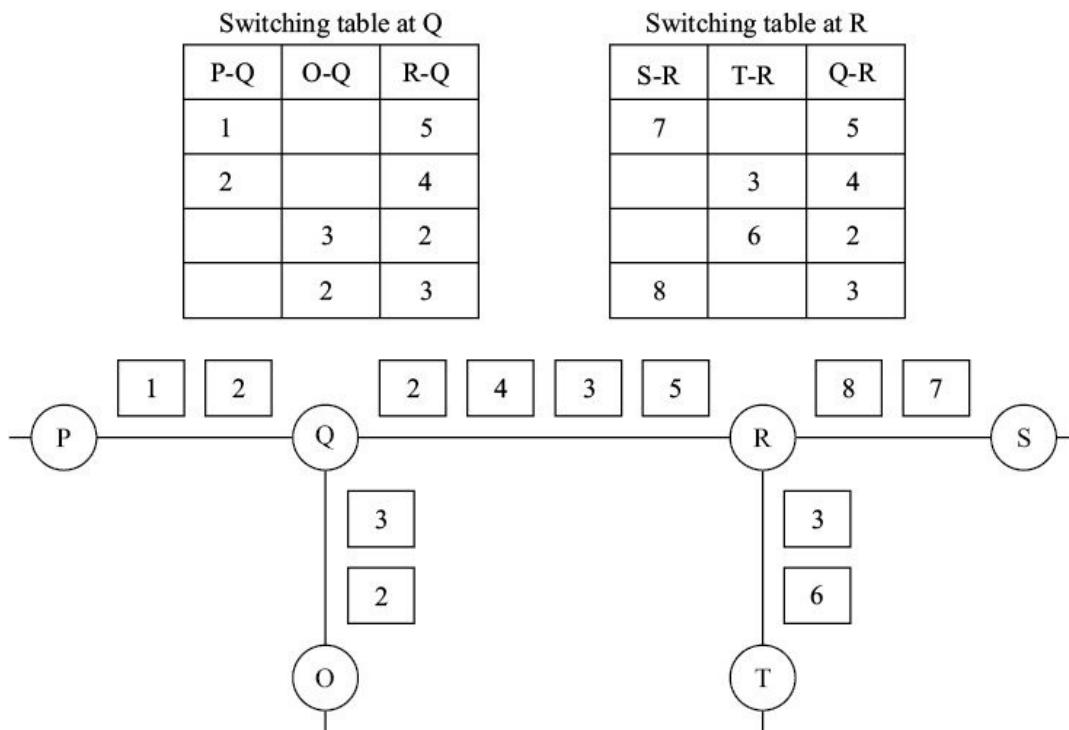


Figure 15.12 Example 15.1.

Solution Node P has two connections, one to node S and the other to node T.

The connection paths are indicated below. The numbers within brackets indicate the logical channel identifiers associated with the connections.

Node P : P -- (1) -- Q -- (5) -- R -- (7) -- S
 : P -- (2) -- Q -- (4) -- R -- (3) -- T

Node O also has two connections, one to node S and the other to node T.

Node O : O -- (3) -- Q -- (2) -- R -- (6) -- T
 : O -- (2) -- Q -- (3) -- R -- (8) -- S

Let us now review quickly the features of virtual circuit packet switching:

- There are three phases—connection establishment, data transfer, and connection release.
- Source and destination addresses are indicated only during connection establishment phase. Routing tables are also required during connection establishment phase.
- All packets carry logical channel identifiers.
- Switching tables are used forwarding the packets during data transfer phase. Switching is based on logical channel identifiers.
- All data packets take the same path. Therefore the delivery delay is almost constant. Minor variations can be there due to retransmission of packets between two nodes if there is an error.
- There is no packet loss due to errors. Packets are never discarded.
- There is delivery confirmation for each packet in form of acknowledgement. Virtual circuit service is, therefore, reliable.
- As the packets follow the same route, their sequence is retained across the network.
- At the time of connection set-up, the nodes allocate some buffer resources for temporary storage of the packets. By specifying a maximum flow control window size based on the buffer size, it is possible to avoid deadlocks.

15.5 PURPOSE OF THE NETWORK LAYER

Having understood the operation of various types of packet switching data networks, let us now examine the layered architecture of these networks. The layered architecture helps in identifying the functions carried out by each layer.

Consider that there are two end systems A and B connected to a switched data network (Figure 15.1). Let us examine the layered architecture of the end to end path taken by a data packet.

15.5.1 The End System to Access Node Link

As we have seen earlier, the physical layer provides the capability to exchange bits on physical transmission medium which interconnect the two devices. The end system can be connected to the access node on a point-to-point dedicated physical link or through a local area network (Figure 15.13).

Point-to-point link is usually through a pair of modems (Figure 15.13a). The data link layer protocols are HDLC or PPP. The data link layer takes care of the errors introduced during transmission of the bits over the physical connection. Thus, together with the physical layer, the data link layer provides an error free data link from end system A to the access node of the network. Note that this layer-2 link does not extend beyond the access node.

It is possible that the end system is connected through a local area network (Figure 15.13b). The interface of the access node that connects to the LAN has LAN interface with MAC and LLC sublayers. The LLC sublayer provides connectionless service to the network layer mentioned above.

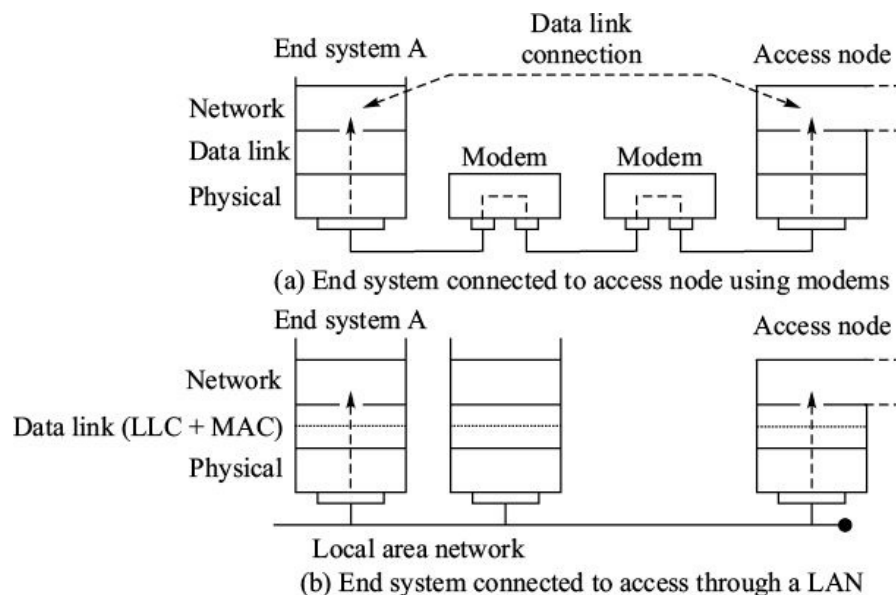


Figure 15.13 End system to access node link.

15.5.2 Node to Node Link

15.5.2 NODE-TO-NODE LINK

The nodes of a packet switched data network have multiple ports. All the ports have associated physical and data link layers (Figure 15.14). The data link layers of various ports in a node provide service to a common network layer. HDLC or PPP data link protocols are used between adjacent nodes. The data links between adjacent nodes take care of errors introduced in the physical connection between the two nodes.

There can be point-to-point full duplex LAN connectivity between two nodes if they happen to be in the same building. If point-to-point LAN connectivity is used, the data link protocols are MAC and LLC.

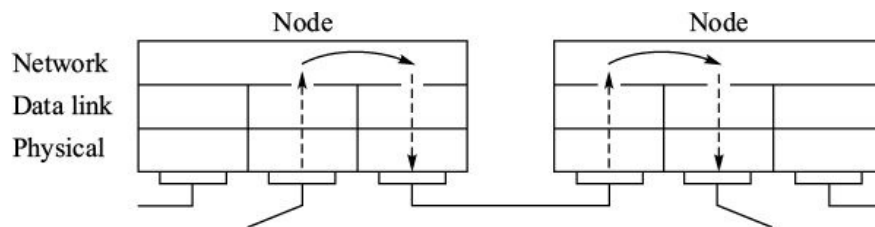


Figure 15.14 Node-to-node data link connection.

15.5.3 End System-to-End System Layered Network Architecture

The end-to-end transport of data packets is achieved by routing the data packets through series of data links across the network (Figure 15.15). As each node is connected to several other nodes, there is need to decide the data link to be chosen at each node for further transport of the data units. This routing decision is taken by the network layer of the nodes. The forwarding tables (and switching tables in virtual circuit routing) maintained at each node indicate the next data link through which a data packet is to be forwarded to the next node.

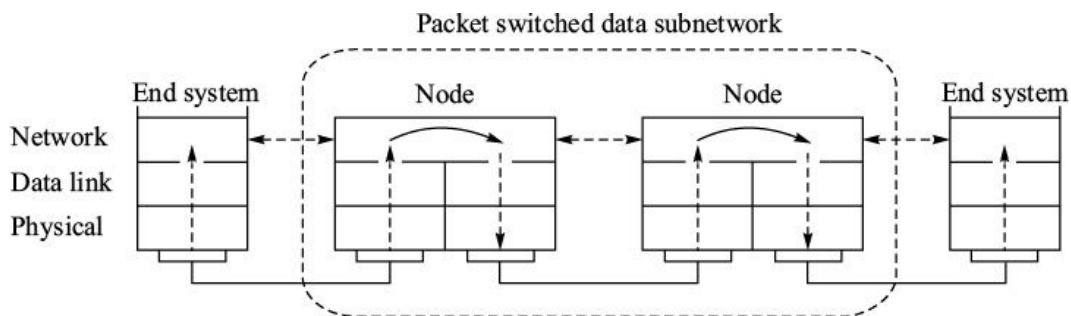


Figure 15.15 End system-to-end system layered architecture.

In virtual circuit packet switching, the virtual path through the network is established on per call basis. The network layer of the end system interacts with the network layer of the access node to establish the end-to-end virtual

connection, exchange data packets, and for releasing the connection. In datagram routing, the network layer of an end system generates data packets that carry the destination and source addresses. The network layer of the nodes route the data packets to their destinations.

Thus, the overall purpose of the network layer is to route the data packets from one end system to the destination by transporting these data packets over a series of data links. The mechanisms deployed for their transport can be connection-oriented or connectionless.

15.6 NETWORK SERVICE

‘Services’ are the visible capabilities provided to the next higher layer. The network layer provides service to the transport layer (Figure 15.16). Note that the transport layer resides in the end systems only. The highest layer in the network nodes is the network layer.

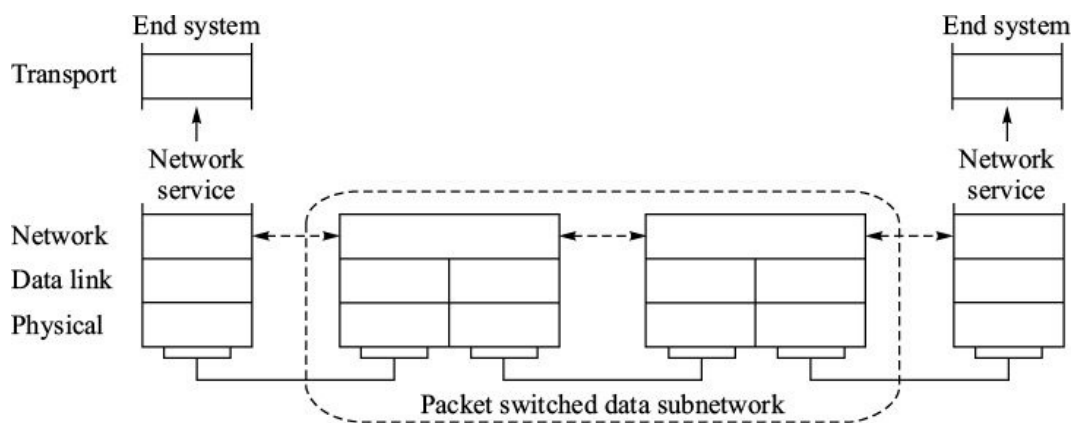


Figure 15.16 Network service.

The network service provides for transparent exchange of Network-Service-Data-Units (N-SDU) between two transport entities that reside in the end systems at the Network-Service-Access-Point (N-SAP). Network service can be of two types:

- Connection-mode Network Service (CONS)
- Connectionless-mode Network Service (CLNS).

15.6.1 Connection-oriented Network Service (CONS)

Connection-oriented network service is specified in ISO 8348 document. The

corresponding ITU recommendation is ITU-T X.213. In CONS, a network connection is first established and then N-SDUs are transported over the connection. N-SDUs are always delivered in the same sequence and an attempt is made to ensure that they are not lost or duplicated. Therefore, CONS is a 'reliable' service. In case of network connection failure, the transport layer is informed.

15.6.2 Connectionless-Mode Network Service (CLNS)

Connectionless-mode network service is specified in ISO 8348 Addendum-1 and ISO 8473 documents. The IP protocol of TCP/IP suite also provides connectionless-mode service to the TCP layer. Datagram service that we just discussed, is the connectionless-mode network service. In CLNS, each N-SDU carries the destination and source addresses and is delivered independent of other N-SDUs. As already mentioned, N-SDU may be lost, duplicated or delivered out of sequence. The network layer cannot report these failures to the transport layer because N-SDUs do not bear sequence numbers. In other words, there is no guarantee that the N-SDUs will be correctly delivered or delivered at all. Therefore, CLNS cannot be considered as reliable, rather it is best 'effort service' service. The transport layer has to make its own efforts to correct delivery of T-PDUs, transport protocol data units.

15.6.3 Basic Features of Network Service

The basic features of the network service are described below. Some of the features are specific to one of the two types of network service—CONS and CLNS.

- The network layer masks the dissimilarities of various networks. Therefore, transport layer is relieved from all concerns regarding how various networks are to be used.
- Network service is transparent, *i.e.* it does not restrict the content, format or coding of the user data.
- When requested, the network layer establishes, maintains, and releases network connection between transport entities.
- Unrecoverable errors are notified to the transport entities.
- The transport layer may reset a network connection. Reset causes all the N-SDUs still in transit to be discarded.
- N-SDUs can be expedited when requested. Expedited N-SDUs are

delivered before the N-SDUs issued subsequently.

15.7 FUNCTIONS OF THE NETWORK LAYER

Functions are the activities that are carried out by a layer to provide the required service. The network layer carries out its functions by adding a header, in the form of Protocol Control Information (PCI) to the N-SDUs. The N-PDUs so formed are transported over the data links. The header contains all the required information necessary to perform the functions. Some of the functions carried out by the network layer are listed below. We will learn all the functions in detail when we describe the network layer protocols in the following chapters.

Network connection. On receipt of connect request from the transport layer, the network layer establishes end-to-end network connection. It makes use of point-to-point data link connections between the nodes of the network.

Routing and forwarding. The network layer of the nodes forwards the N-SDUs to the next node. Forwarding decision is made at each node based on routing function which generates a forwarding table at each node.

Multiplexing. In order to optimize the use of data link connections, the network layer may multiplex several network connections on one data link connection (Figure 15.17a).

Segmenting and blocking. Segmenting and blocking of NSDUs is done by the network layer to get the N-PDU of the required size (Figures 15.17b and c). The delimiters of the N-SDUs are preserved during these operations.

Error detection and recovery. Error detection functions are used to check that the quality of network service is maintained. Error recovery and reporting mechanisms are incorporated for the errors encountered.

Other functions. The network layer also carries out flow control to ensure that the network nodes are not congested. The network layer also provides for expedited delivery of N-SDUs, reset of network connections.

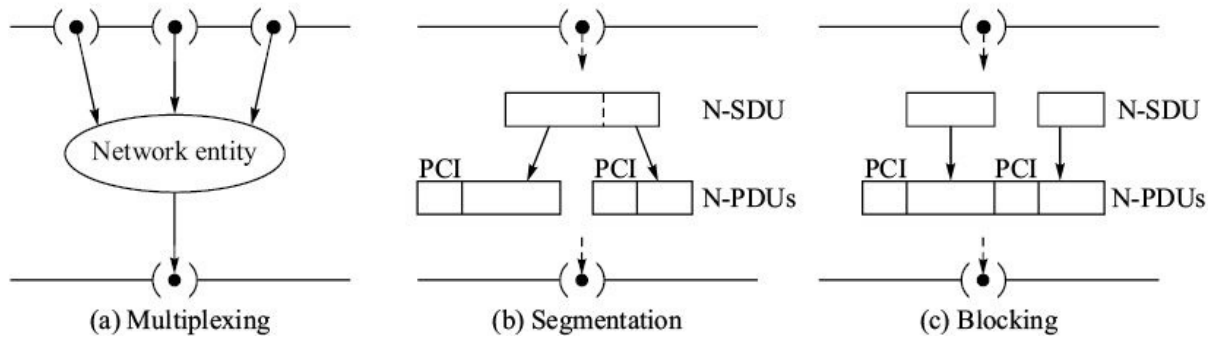


Figure 15.17 Multiplexing, segmentation, and blocking in the network layer.

15.8 INTERNETWORKING

We have studied so far two broad categories of the data networks—LANs and WANs. Communication requirements are usually not restricted one network and there is always need to interconnect networks. We saw how the local area networks were interconnected using bridges in Chapter 13. When two or more networks are interconnected, we refer to such extended network as internetwork (Figure 15.18). The devices used for interconnecting two networks are called Intermediate Systems (IS) or Internetworking Unit (IWU).

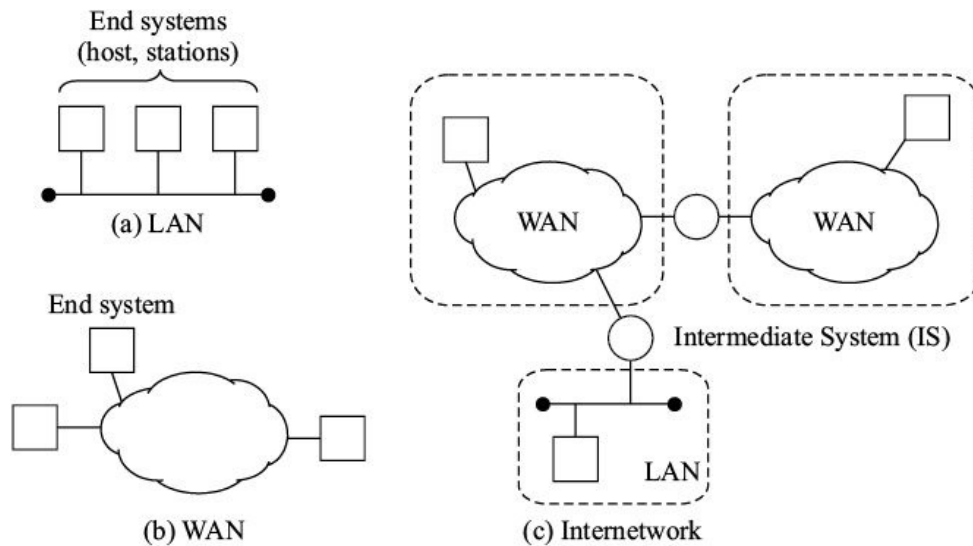


Figure 15.18 Internetworking.

Majority of the data networks is based on datagram switching and routers are used as internetworking devices. Figure 15.19 shows an example of a simple internetwork consisting of four routers. The routers are interconnected by point-to-point WAN links. The routers also connect to the local area networks. The

internetwork appears to be a seamless network from the user's perspective and it is not surprising that the terms 'internetwork' and 'network' are used interchangeably.

The public Internet that we are all familiar with as users, is a global internetwork that connects millions of computers. It is based on datagram approach and uses TCP/IP protocol. Wide spread deployment of TCP/IP has forced other packet switching technologies into background. Virtual circuit switching technologies (X.25, ATM, frame relay) that gained prominence in 1980s and early 1990s are today used for supporting Internet. The point-to-point WAN links in Figure 15.19 may not necessarily be TDM links. They can be virtual point-to-point WAN links,

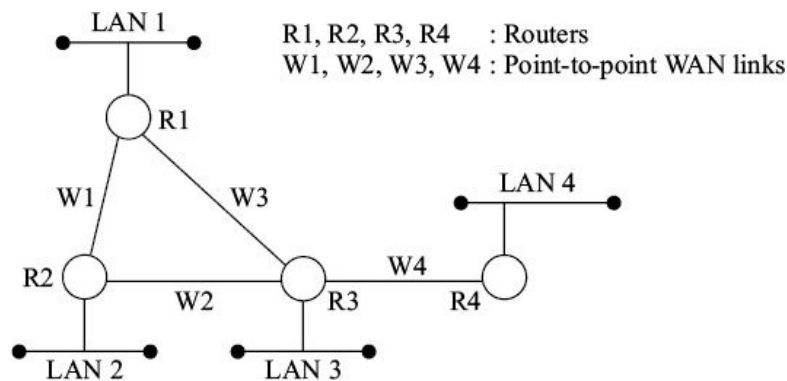


FIGURE 15.19 An internetwork.

provisioned on X.25, ATM or frame relay networks. We will describe all these networking and internetworking technologies and related protocols in the coming chapters.

SUMMARY

Switched data networks provide flexible interconnection topology and enable sharing of the network resources. Switching techniques are primarily of two kinds—circuit switching and store-and-forward switching. In the former case a dedicated path is established between the end systems for transport of data units. In store-and-forward switching, data units are individually switched based on addressing information carried by them.

Store-and-forward switch can be of two types, message switching and packet switching. Message switching has inherently high delivery delay. To reduce the delivery delay, the messages are partitioned into packets. There are two approaches for routing of the packets through the network—datagram switching

and virtual circuit switching. In datagram switching, a network node switches the received packet to one of its outgoing ports based on a forwarding table maintained by it. Each packet is switched independent of the other packets of the same flow. In virtual circuit switching, a virtual path for the packets is established between the source and the destination. All the packets are delivered in sequence on this path.

Forwarding tables for datagram switching can be static or dynamic. Static tables are created and updated manually. There are routing protocols (RIP, OSPF, BGP) for creating and updating forwarding tables dynamically.

The network layer provides the means to route data units from one end system to another end system through a cascade of data links across a network. It provides Connectionless-mode Network Service (CLNS) or Connection-oriented Network Service (CONS) to the transport layer. There is always need to interconnect two networks. Such interconnected networks are called internetwork.

EXERCISES

1. Match the features given in the first column to the switching technologies in the second column.

I

- Data rates at the source and at the destination must be same.
- Source and destination address are specified once and there is minimal delivery delay.
- No connection and release phases.
- Significant and random delivery delay.
- Sequence delivery and data rates at the source and destination can be different.

II

- Circuit switching
- Message switching
- Virtual circuit switching

2. If the bit rate is r bits/sec., the message length is l bits, and the propagation delay per hop is p seconds, what is the message transmission delay across k nodes of a network, when:

- (a) the network is circuit switched. Assume that the connection is already established
- (b) the network is message switched. Assume overhead of h bits per message for addressing and average processing/queue delay of d seconds at each node.

(c) the network is packet switched and virtual circuit routing is used. Assume that
 packet size is s bits and each packet contains m bits of the message,
 packet processing delay at each node is f seconds,
 packetization and reassembly are carried out at the entry and exit nodes and
 the respective delay is u seconds at these nodes, and
 virtual connection establishment delay is t seconds.

3. Figure E15.20 shows a virtual circuit packet switching network. Determine the routes of the various connections established by the end systems P, Q, and R. The switching tables at the various nodes are given below:

A-P	A-B	A-C
3	6	
8		4

B-Q	B-A	B-C	B-D
	6	2	
2		3	
5			8

C-A	C-B	C-D
	2	5
4	3	

D-R	D-B	D-C
1		5
9	8	

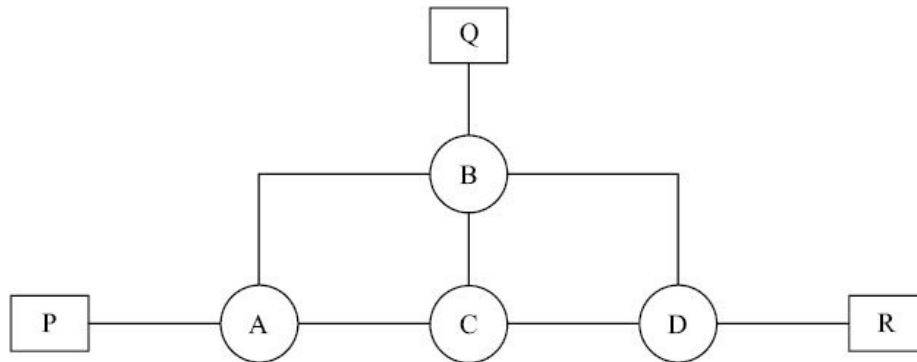


Figure E15.20.

4. Figure E15.21 shows a virtual circuit switched packet network consisting of four packet switching nodes. Write the switching tables at the nodes and the connection tables at the end systems when the following connections are established in the sequence indicated without clearing the previously established connections. Assume that the connection identifiers are chosen starting from 0 on each link.

- (a) End system A connects to end system B.
- (b) End system C connects to end system G.
- (c) End system E connects to end system I.
- (d) End system D connects to end system B.

- (e) End system F connects to end system J.
- (f) End system H connects to end system A.

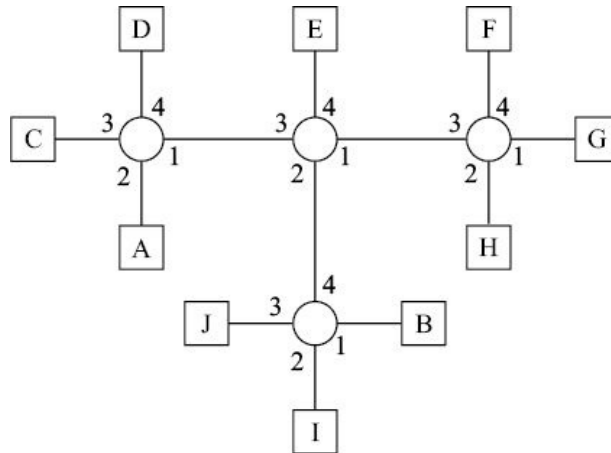


Figure E15.21.

5. Figure E15.22 shows a datagram switching network consisting of four packet switching nodes P, Q, R, and S. Write the forwarding table of each node. The datagrams should take the path with the lowest cost. The cost of sending a datagram along a link is indicated in the figure. The forwarding table should contain the following attributes:

- (a) Destination
- (b) Outgoing port
- (c) Path cost to destination.

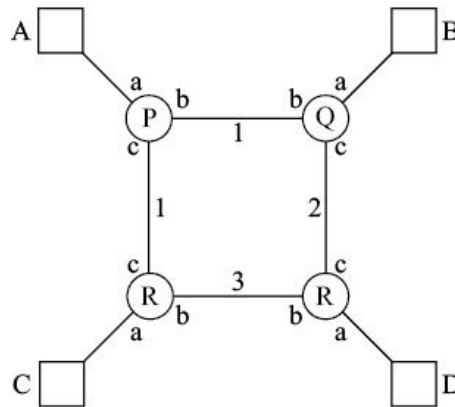


Figure E15.22.

6. Give an example of a virtual link that interconnects end systems A and B and passes through all three nodes P, Q, and R and the packets travel twice across the link PQ.

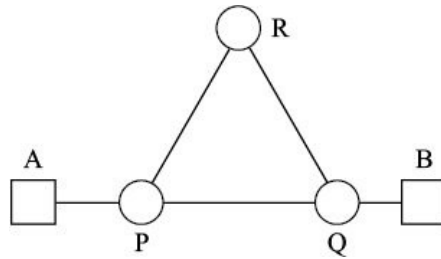


Figure E15.23.

7. Sam argues with Ram that packet switching requires overhead bits (addressing information) with each packet of user data. In circuit switching, on the other hand, user data is transmitted without any overhead bits. Therefore, circuit switching should be more efficient in link utilization than packet switching. What is the flaw in his reasoning?
8. Two end systems A and B are connected through store and forward node N as shown in Figure E15.24. The propagation delay of the interconnecting links is 100 ms. The links operated at 1 Mbps. The store-and-forward node has negligible processing delay. Calculate the delivery time of a 100 k bits long message from node A to B when
 - (a) the node is a message switch.
 - (b) the node is a packet switch. Assume packet size of 1k bits.
 Neglect the overhead of addressing bits.

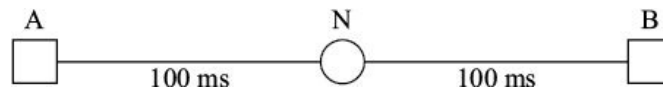


Figure E15.24.

9. Repeat Exercise 8 when an addressing overhead of 1k bits is required.
10. Repeat Exercise 8 when the node introduces processing delay of 1 ms.
11. Figure E15.25 shows a virtual circuit switching network consisting of four nodes A, B, C, and D. During the call set-up, the CONNECT REQUEST packet is sent along the lowest link cost path. The link costs are asymmetrical, *i.e.* the cost of sending a packet from A to B is different from the cost of sending a packet from B to A.
 - (a) Draw the virtual connection X set up by end system P to end system Q. What is the total path cost?
 - (b) Draw the virtual connection Y set up by end system Q to end system P. What is the total path cost?
 - (c) If P sends a data packet to Q on X and Q acknowledges, what route will the

acknowledgement take?

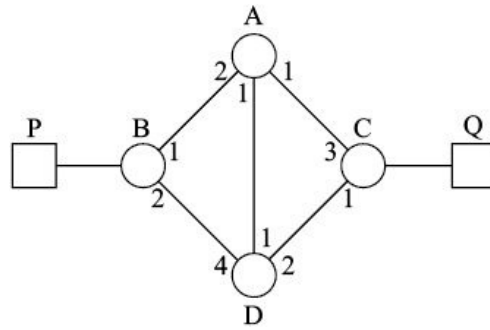


Figure E15.25.

1 This table is constructed in similar manner as in datagram switching using static or dynamic routing methods.

16

Virtual Circuit Packet Switching Network

In the last chapter, we studied principles of virtual circuit packet switching data networks. In this chapter we build on this knowledge and learn three important virtual circuit switching technologies, X.25, frame relay, and ATM. The X.25 technology is based on the first three layers of the OSI reference model and is suitable for data traffic. Frame relay was developed subsequent to X.25 to overcome some of its limitations. It is based on the first two layers of the OSI reference model. Asynchronous transfer mode, or simply ATM, came in the last and supports voice, video, and data applications.

We begin this chapter with X.25 and cover its application, services, and operation in detail. Then we discuss an intermediary device called Packet Assembler and Disassembler (PAD). It enables connecting non-packet mode terminals to X.25 interface of a packet switched data network. Frame relay networks are taken up next. We discuss in detail the congestion control mechanism of the frame relay. Finally we move over to ATM and describe its layered architecture and operation.

16.1 X.25 INTERFACE

ITU-T X.25 recommendation was first developed in 1976. The ISO standard corresponding to X.25 is ISO 8208. It defines the interface for exchange of data packets between a packet mode end system (called DTE—Data Terminal Equipment) operated by the user and the access node of switched packet data network (called DCE—Data Circuit Terminating Equipment) operated by the service provider.

Figure 16.1 shows various possible interconnection configurations between the DTE and the DCE and location of X.25 interface between them. Note that the DTE and DCE are always connected directly or by using modems.

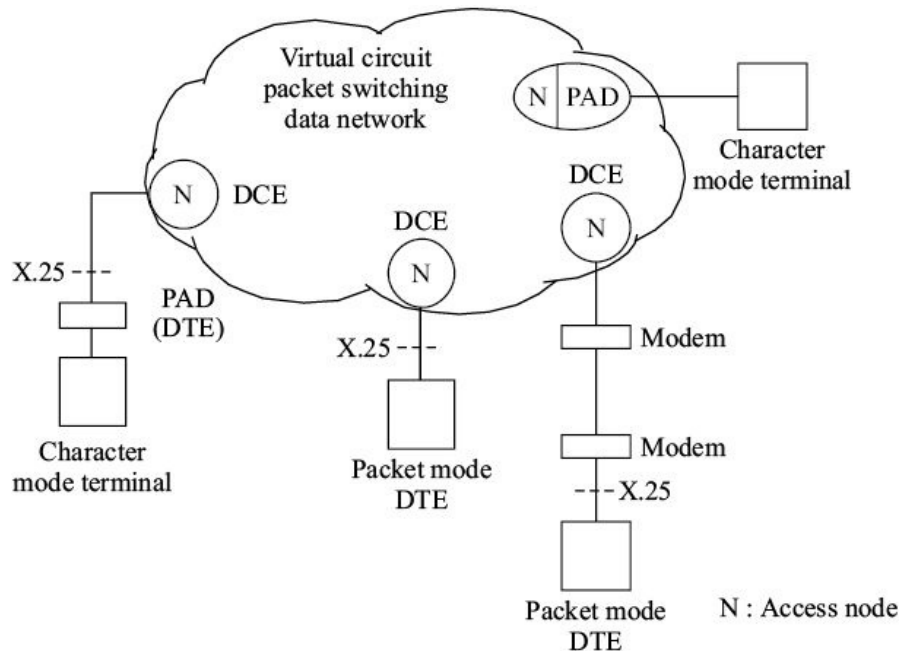


Figure 16.1 Location of X.25 interface.

A non-packet mode terminal requires an intermediary device called PAD (Packet Assembler and Disassembler). The PAD as a packet mode DTE has X.25 interface. It can also be built into the access node, in which case there is no visible X.25 interface.

16.1.1 Scope of X.25 Interface

X.25 interface is defined at three levels, which correspond to the first three layers of the OSI reference model (Figure 16.2).

- At level 1 (Physical layer), X.25 makes use of level 1 of ITU-T recommendation X.21 and X.21bis. X.21 recommendation is for circuit switched data networks.
- At level 2 (Data link layer), X.25 specifies LAP-B protocol. We discussed X.21 and LAP-B protocol in Chapters 7 and 9, respectively.

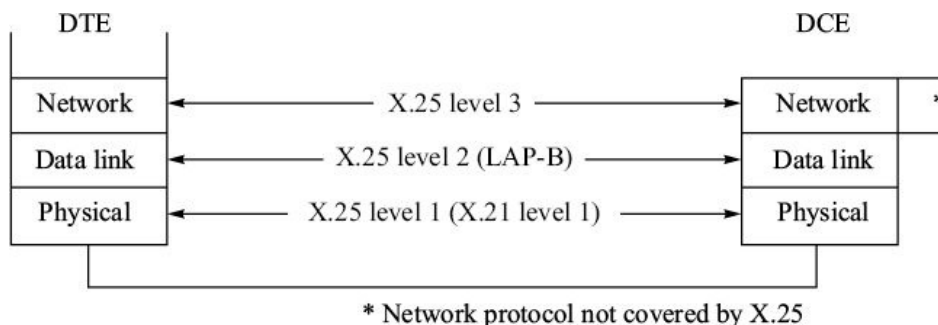


Figure 16.2 The three levels of X.25.

- At level 3 (Network layer), X.25 defines the protocol for accessing and using packet switched data network. It is operated on point-to-point data link connection provided by LAP-B protocol at layer 2.

16.2 X.25 SERVICES

X.25 provides 'connection mode' service by establishing an end-to-end virtual communication path through the network. The virtual path can be of two types:

- Switched Virtual Circuit (SVC)
- Permanent Virtual Circuit (PVC).

Switched virtual circuit (SVC). A switched virtual circuit is a switched connection established at the request of a DTE and is cleared at the end of the call. The network resources are allocated for the duration of the call. Its operation has three phases—call establishment, data transfer, and call clearing phases.

Permanent virtual circuit (PVC). A permanent virtual circuit is a constant virtual connection through the network between two DTEs. It is not to be established and cleared on each instance of communication between the two DTE. It is always in data transfer phase.

16.3 LOGICAL CHANNELS

At level 3, there is exchange of control information and data in the form of packets. Each packet carries a number that identifies the virtual circuit to which it belongs. Multiple virtual circuits can be realized by statistical multiplexing of the packets as shown in Figure 16.3. DTE A is operating three virtual circuits to DTEs B, C, and D. Each of these circuits can be an SVC or a PVC. Packets to and from B, C, and D are assigned numbers N_1 , N_2 , and N_3 , respectively and

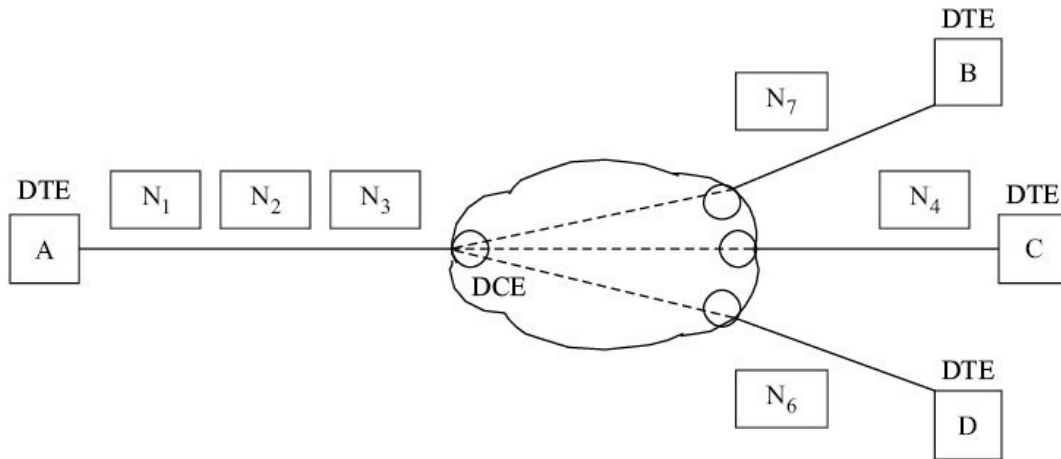


Figure 16.3 Statistical multiplexing of packets for realizing multiple logical channels.

are multiplexed statistically between the DCE and A. The three channels so realized between the DTE and DCE are called logical channels and their numbers are called Logical Channel Identifiers (LCI). Logical channel identifiers have local significance between the DTE and the DCE. An end-to-end virtual circuit is association of LCIs at the two ends, *e.g.* the virtual circuit between A and B is association of LCIs N₁ and N₇ at the two ends (Figure 16.3).

16.3.1 Grouping of Logical Channels

In X.25, the logical channel identifier is 12-bit long and therefore 4096 logical channel identifiers are possible on each port of a DCE. 4096 channel identifiers are divided into several ranges as shown in Table 16.1.

TABLE 16.1 Grouping of Logical Channels		
Channel assignment by	Ranges	Used for
	0	RESTART
	1-HPC	Permanent virtual circuits (PC)
DCE	LIC-HIC	Incoming virtual circuits (IC)
DTE, DCE	LIOC-HIOC	Incoming and outgoing virtual circuits (IOC)
DTE	LOC-HOC	Outgoing virtual circuits (OC)

H: High end of the range L: Low end of the range

- Channel identifier 0 is reserved for RESTART.
- First range of identifiers from 1 to HPC is for permanent virtual circuits.
- Second range of identifiers (LIC to HIC) is for incoming virtual circuits from DCE. These assignments are used by the DCE when there is an incoming call to a DTE.
- Third range of identifiers (LIOC to HIOC) is for incoming/outgoing virtual circuits. These assignments can be used for incoming calls by the DCE and for the outgoing calls by the DTE. This range is used only when the specific ranges for incoming and outgoing circuits have been exhausted.
- Fourth range of identifiers (LOC to HOC) is for the outgoing virtual circuits and is used only by the DTE for its outgoing call.

Whenever there is an incoming call, the DCE uses the lowest free logical channel identifier from the LIC to HIC range. For an outgoing call, the DTE uses the highest free logical channel identifier from the range LOC to HOC. Therefore, the DTE and DCE always search for free logical channel identifier from opposite ends, thereby minimizing the chances of their selecting the same channel identifier for an outgoing and an incoming call.

16.4 GENERAL PACKET FORMAT

All interactions between a DTE and a DCE are in the form of exchange of data packets and control packets. The data packets carry user data and the control packets are used for managing the virtual circuits and for flow control. Figure 16.4 shows the general format of these packets. A packet is usually drawn as a stack of octets. The first octet is on the top and the first bit is on the right hand side. Like a frame at the second layer, a packet consists of several fields.

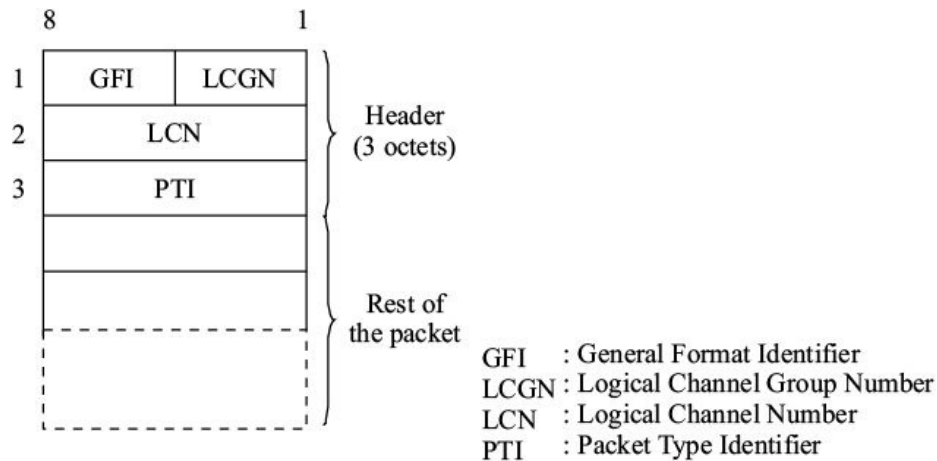
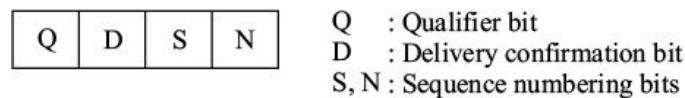


Figure 16.4 General format of a packet.

The first three octets constitute the header. The header is always present in all the types of packets. It contains the following fields: **Logical channel group number (LCGN)**. The logical channels are divided into 16 groups, each group containing 256 logical channels. First four bits of the first octet contain the 4-bit LCGN.

General format identifier (GFI). Bits 5–8 of the first octet comprise the GFI field and are coded as shown in the Figure 16.5. We shall describe the use of these bits later.



Packet type	GFI			
	8	7	6	5
Call set-up packets	0	x	S	N
Data packets	x	x	S	N
Other packets	0	0	S	N

x : 0 or 1
 S N : 0 1 for modulo-8 sequence numbering scheme
 : 1 0 for modulo-128 sequence numbering scheme

Figure 16.5 General format identifier field.

Logical channel number (LCN). Bits 1–8 of the second octet comprise the logical channel number associated with the LCGN. Four bits of LCGN and eight bits of LCN together constitute 12-bit Logical Channel Identifier (LCI) introduced earlier.

Packet type identifier (PTI). The packet type is identified by the third octet which is called the Packet Type Identifier (PTI) field. If the first bit of this field is 0, it is a data packet, otherwise, it is one of the control packets. Table 16.2 gives the PTI field of the various control packets used in the X.25 interface.

8	7	6	5	4	3	2	1	DTE → DCE	DCE → DTE
0	0	0	0	1	0	1	1	CALL REQUEST	INCOMING CALL
0	0	0	0	1	1	1	1	CALL ACCEPTED	CALL CONNECTED
0	0	0	1	0	0	1	1	CLEAR REQUEST	CLEAR INDICATION
0	0	0	1	0	1	1	1	CLEAR CONFIRMATION	CLEAR CONFIRMATION
0	0	1	0	0	0	1	1	INTERRUPT REQUEST	INTERRUPT INDICATION
0	0	1	0	0	1	1	1	INTERRUPT CONFIRMATION	INTERRUPT CONFIRMATION
0	0	0	1	1	0	1	1	RESET REQUEST	RESET INDICATION
0	0	0	1	1	1	1	1	RESET CONFIRMATION	RESET CONFIRMATION
1	1	1	1	1	0	1	1	RESTART REQUEST	RESTART INDICATION
1	1	1	1	1	1	1	1	RESTART CONFIRMATION	RESTART CONFIRMATION

Note that the same PTI is used for a pair of logically related packets, *e.g.* a CALL REQUEST packet from a DTE to a DCE results in an INCOMING CALL packet from the remote DCE to the remote DTE and these two packets have the same PTI field.

Besides the header, a packet has other fields containing addresses, facilities required, and the user data. While a header is always present in all packets, these fields may or may not be present in a packet depending on its type. We will describe these fields as and when we come across them while discussing operation of the X.25 interface.

Before we go into the actual operation of X.25 (level 3) interface, it must be kept in mind that packets, whatever the type, are assembled in the network layer and handed over as data to the data link layer. The data link layer, having already established the data link connection between DTE and DCE by exchange of mode-setting commands and responses, sends one packet at a time in the information field of an I-frame. If the I-frame is received without errors by the data link layer at the other end of the data link connection, the packet in the information field of the frame is handed over to the network layer. The procedures described below apply to the packets successfully transferred across the data link connection.

16.5 PROCEDURES FOR SWITCHED

VIRTUAL CIRCUITS

Operation of switched virtual circuits involves three phases—call establishment, data transfer, and call clearing. In addition, there are special procedures to deal with abnormal situations — call collision, reset, restart and error recovery. The three phases of operation and these procedures are described now.

16.5.1 Call Establishment Phase

In the call establishment phase, the calling DTE issues a CALL REQUEST packet to the local DCE, the access node of the switched packet data network (Figure 16.6). The network establishes a virtual circuit to the remote DCE that serves the called DTE. The remote DCE sends an INCOMING CALL packet to the called DTE. If the called DTE decides to accept the call, it returns a CALL ACCEPTED packet. The CALL ACCEPTED packet is passed to the local DCE on the same virtual circuit. The local DCE notifies the calling DTE with a CALL CONNECTED.

If the called DTE decides not to accept the call, or the network cannot establish a virtual circuit to the remote DCE, the call clearing procedure is initiated.

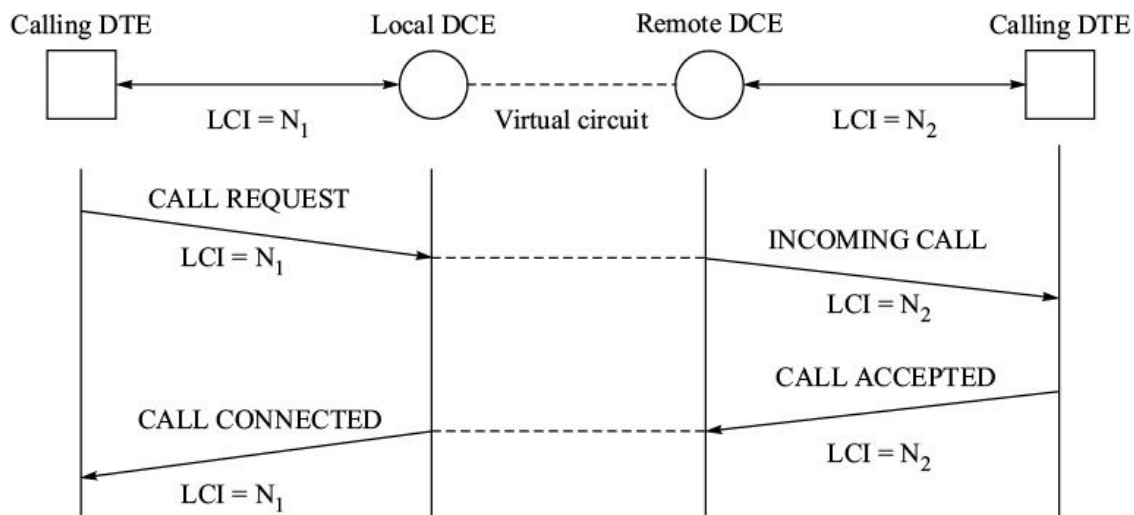


Figure 16.6 Call establishment procedure.

Detailed scenario

1. The calling DTE:

- Selects a free logical channel identifier N₁ from the higher end of LOC–

HOC range (Table 16.1).

- Builds a CALL REQUEST packet specifying logical channel identifier N_1 , the calling and called DTE addresses in BCD form, and special facilities required (Figure 16.7). Length of the address and facilities fields are also specified to facilitate their identification. The PTI field of the packet identifies it as CALL REQUEST packet.
- Sends the CALL REQUEST packet to the local DCE.

2. On receipt of the CALL REQUEST packet, the local DCE:

- Records the logical channel identifier N_1 .
- Checks the validity of required facilities.
- Forwards the CALL REQUEST packet to the next node and thus building the virtual circuit up to the remote DCE that serves the required destination.

Q	D	S	N	LCGN			
LCN							
0	0	0	0	1	0	1	1
Calling DTE address field length				Called DTE address field length			
DTE addresses							
Facility field length							
Facilities							

(a) Format of CALL REQUEST and INCOMING CALL packets

Q	D	S	N	LCGN			
LCN							
0	0	0	0	1	1	1	1
Calling DTE address field length				Called DTE address field length			
DTE addresses							
Facility field length							
Facilities							

(b) Format of CALL ACCEPTED and CALL CONNECTED packets

Figure 16.7 Packets used in call establishment phase.

3. On receipt of the CALL REQUEST packet, the remote DCE:

- Checks the validity of the required facilities.
- Selects a free logical channel N_2 from the lower end of LIC–HIC range (Table 16.1). Note that the two logical channel identifiers, N_1 on the CALL REQUEST packet and N_2 on the INCOMING CALL packet are different.
- Assembles an INCOMING CALL packet and forwards it to the called DTE.

4. On receipt of the INCOMING CALL packet, the called DTE, if it is ready to accept the call, it:
 - Records the logical channel identifier N_2 .
 - Assembles a CALL ACCEPTED packet using logical channel identifier N_2 received in the INCOMING CALL packet and indicates the accepted facilities in the packet.
 - Sends CALL ACCEPTED packet to the DCE.
 5. The remote DCE forwards the acceptance to the call originating DCE using the virtual circuit already established for the CALL REQUEST packet.
 6. On receipt of the CALL ACCEPTED packet, the local DCE sends a CALL CONNECTED packet using the logical identifier N_1 to the calling DTE. It also indicates the accepted facilities in the packet.
- On receipt of CALL CONNECTED packet carrying logical channel identifier N_1 , the calling DTE comes to know that the required connection has been established.

16.5.2 Data Transfer Phase

The data transfer phase involves exchange of DATA packets between the DTE and the DCE. The local and remote DCE exchange DATA packets on the virtual circuit already established between them (Figure 16.8). The connection having already been established between the two DTEs, the

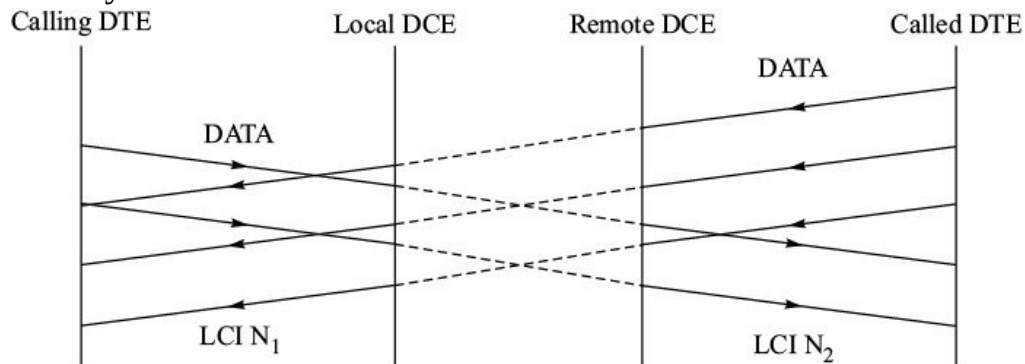


Figure 16.8 Data transfer phase.

source and destination addresses are no longer required. The logical channel identifiers N_1 and N_2 identify the connection and they only are specified in the DATA packets.

Flow control. To avoid congestion in the network, X.25 implements a flow control mechanism at the network layer also. The sliding window flow control mechanism that we studied in Chapter 8 is also used here.

- Each DATA packet and acknowledgement is given a sequence number.
- Windows are maintained by the DTE and the DCE. The window size for each logical channel is usually 2.
- The acknowledgements are Receive Ready (RR), Receive Not Ready (RNR) and Reject (REJ). These acknowledgements pertain to layer 3 and have nothing to do with similar named acknowledgements at layer 2.
- RR can be sent piggybacked on a DATA packet.

Formats of the flow control and DATA packets. Figure 16.9 shows the formats of the flow control and the DATA packets. The flow control packets consist of 3-octet header only. The first

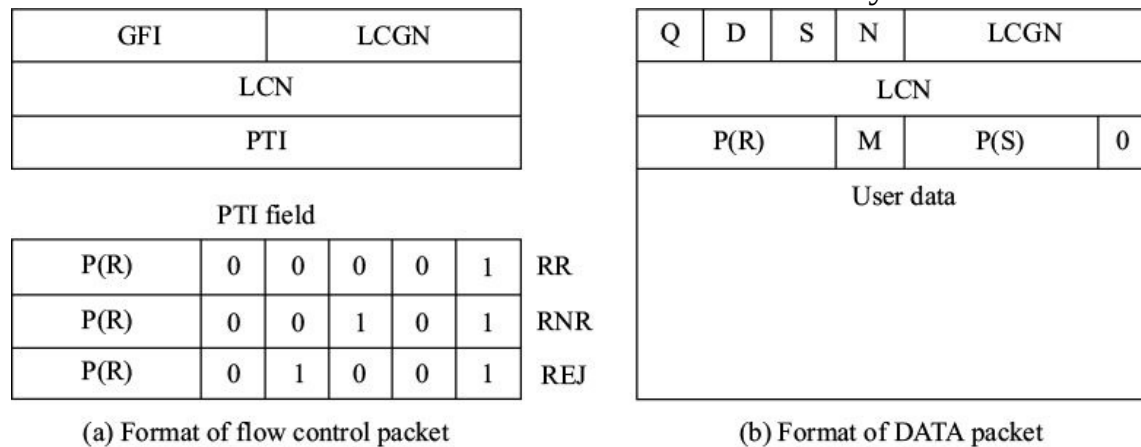


Figure 16.9 Packets used in data transfer phase.

bit of the PTI field (third octet) identifies a packet as a DATA packet. It is always 0 in the DATA packets. P(S) is the sequence number of the DATA packet and P(R) is the sequence number of the piggybacked acknowledgement (RR). These are 3 bits long.

The maximum size of the data field is usually 128 octets but the users have options for restricting it to 16, 32, 64, 256, 512, 1024, 2048, and 4096 octets. The maximum size of the data field is negotiated during the call establishment phase by inserting the desired size in the facility field.

Local and remote acknowledgements. When a DTE sends a DATA packet, there are two alternatives for sending the acknowledgement—local or remote

acknowledgements. Acknowledgement can be either given by the local DCE when it receives a DATA packet from the DTE. In remote acknowledgement option, the local DCE waits for acknowledgement from the called DTE. When the acknowledgement is received, the local DCE forwards it to the DTE (Figure 16.10).

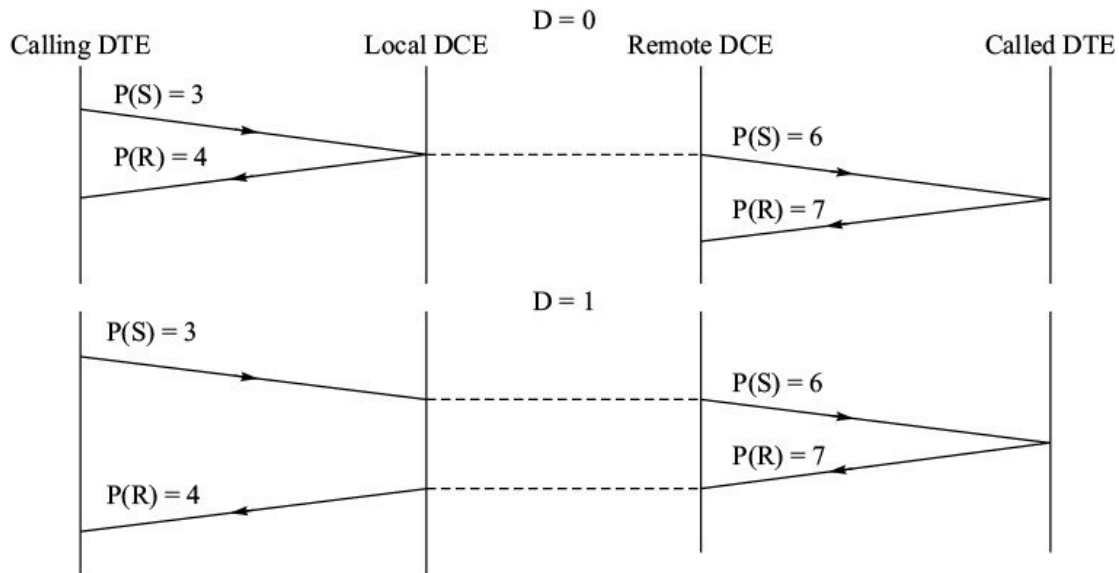


Figure 16.10 Acknowledgement alternatives.

The D (delivery confirmation) bit, which is the seventh bit of octet 1 (Figure 16.9), specifies whether the acknowledgement has local or remote significance. D bit is set to 1 for remote acknowledgements. When it is set to 0, the local DCE sends the acknowledgement.

Remote acknowledgement is more useful because it ensures that the DATA packet has reached the destination although waiting for acknowledgement slows down the data transfer. Figures 16.11 and 16.12 show data transfer situations when $D = 0$ and $D = 1$. For the sake of illustration, the window sizes for the interface and within the subnetwork are kept different. The window size is 2 at the DTE-DCE interfaces, and the window size is 3 within the network.

Figure 16.11 shows an example of data transfer when $D = 0$. The local DCE cannot accept more than three packets from the DTE due to window size restriction of the network. It acknowledges only the first packet having $P(S) = 0$ so that the DTE is permitted to send only two more packets having $P(S) = 1$ and $P(S) = 2$. The local DCE sends the next acknowledgement

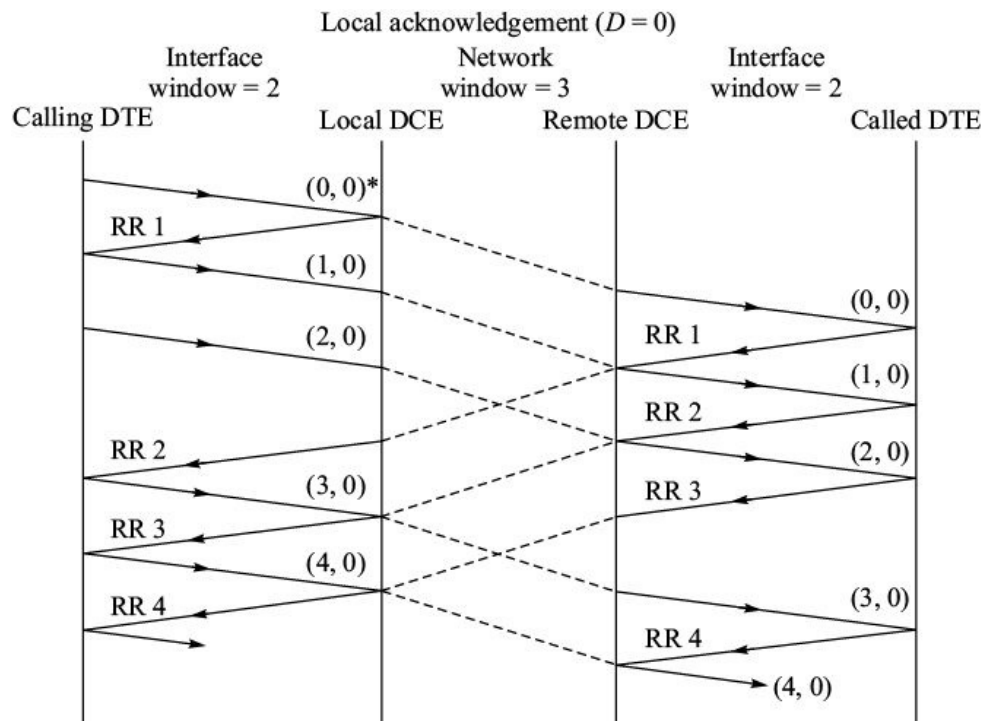


Figure 16.11 Data transfer when $D = 0$.

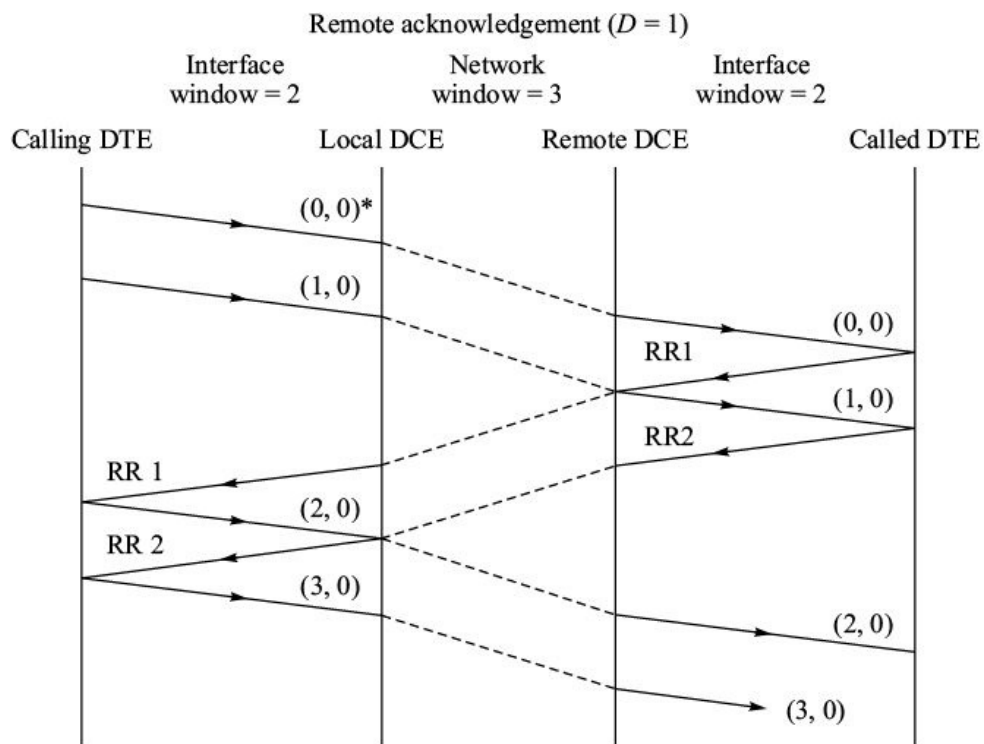


Figure 16.12 Data transfer when $D = 1$.

only when place for another packet in the network window is vacated. Effective throughput, in this case, is determined by the network window size and the round trip delay from DTE to DTE.

Figure 16.12 shows an example of data transfer when $D = 1$. The local DCE can release acknowledgement only after it receives acknowledgement from the remote DCE. In this case, effective throughput is governed by the size of the smaller of the two windows and the round trip delay from DTE to DTE.

INTERRUPT packet. When a DTE does not receive the acknowledgement and has exhausted its window, there is still a way to getting through to the remote DTE. It can send an INTERRUPT packet which is not restricted by the flow control mechanism and can even overtake other DATA packets still in transit. Only one INTERRUPT packet can be sent at a time and it is acknowledged by an INTERRUPT CONFIRMATION packet (Figure 16.13). The INTERRUPT packet can contain 32 octets of user data.

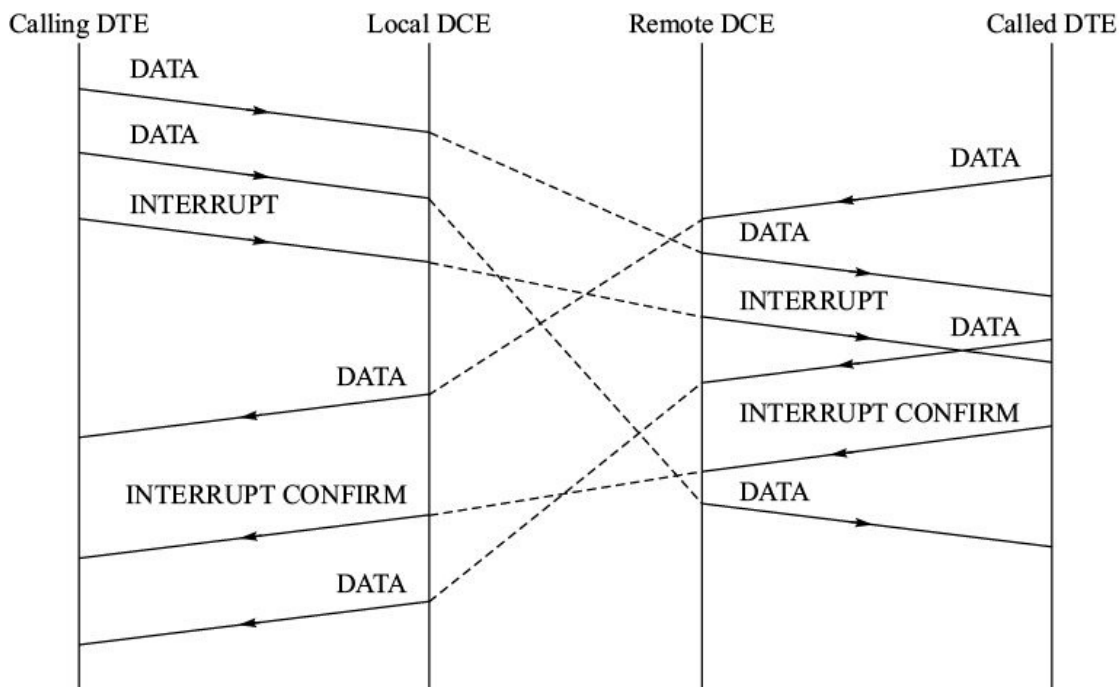


Figure 16.13 INTERRUPT packet and its confirmation.

More data bit (M). The fifth bit of the third octet (Figure 16.9) is called the M bit or more data bit. When it is 1, it indicates that there is a sequence of more than one packet. To illustrate its use, let us assume that the two interfaces at the originating and destination ends have different agreed sizes of the DATA packets (Figure 16.14). The interface at the destination has a smaller packet size. Therefore, the remote DCE splits each received DATA packet into four packets

and sets M bit to 1 in each of the first three packets indicating “more packets of the sequence to follow”. In the last packet it resets the M bit to 0.

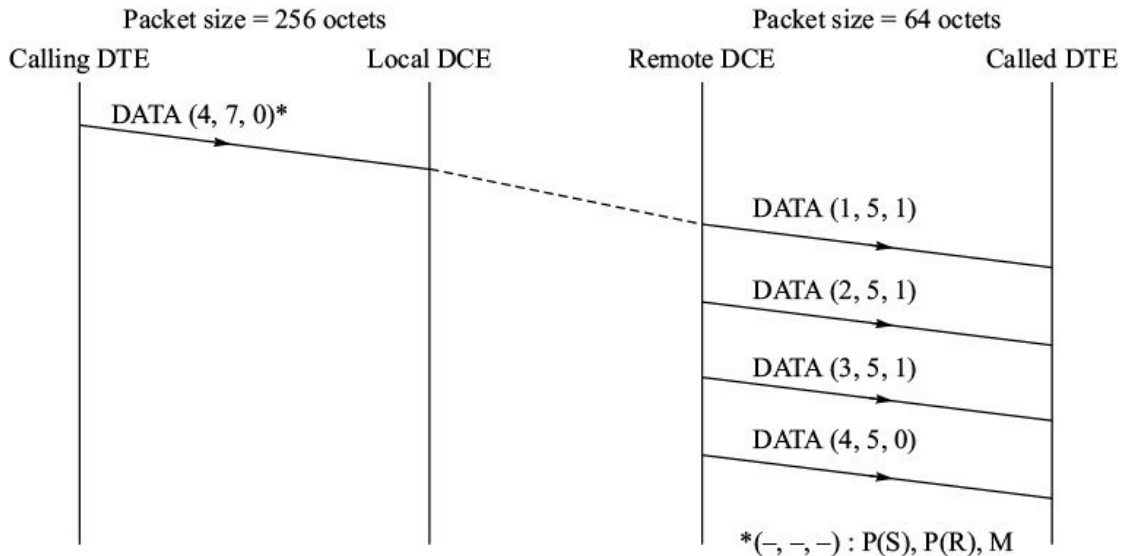


Figure 16.14 M bit for accommodating difference in agreed packet sizes.

Sequence numbering. The three bits sequences numbers provided in X.28 packets can support maximum window size of 7 using modulo-8 numbering scheme. For window sizes greater than 7, X.25 specifies the modulo-128 numbering scheme. The S and N (sequence numbering) bits of GFI indicate the numbering scheme being followed (Figure 16.9). For modulo-8 numbering, the SN bits are 01 and for modulo-128 numbering, the SN bits are 10. The format of DATA and acknowledgement packets undergoes slight change as modulo-128 requires 7 bits for a sequence number. Figure 16.15 shows the changed formats.

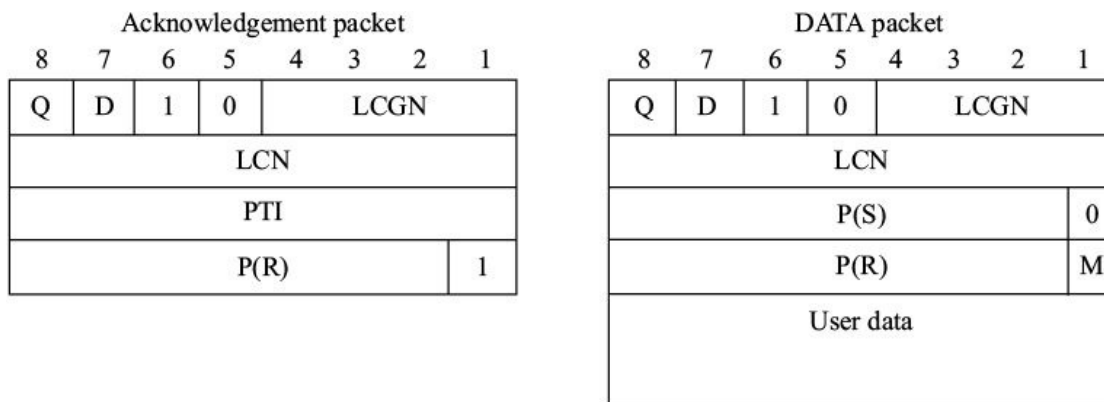


Figure 16.15 Formats of acknowledgement and DATA packets for modulo-128 numbering scheme.

16.5.3 Call Clearing Phase

A virtual circuit can be cleared by either party at any time by sending CLEAR REQUEST packet on the particular logical channel. Figure 16.16a shows the sequence of packets exchanged during the call clearing phase when a DTE requests for call clearing. Under certain conditions, the network can also clear the call. Figure 16.16b shows call clearing by the network. The clearing process is destructive. Any undelivered DATA packets in the network are discarded.

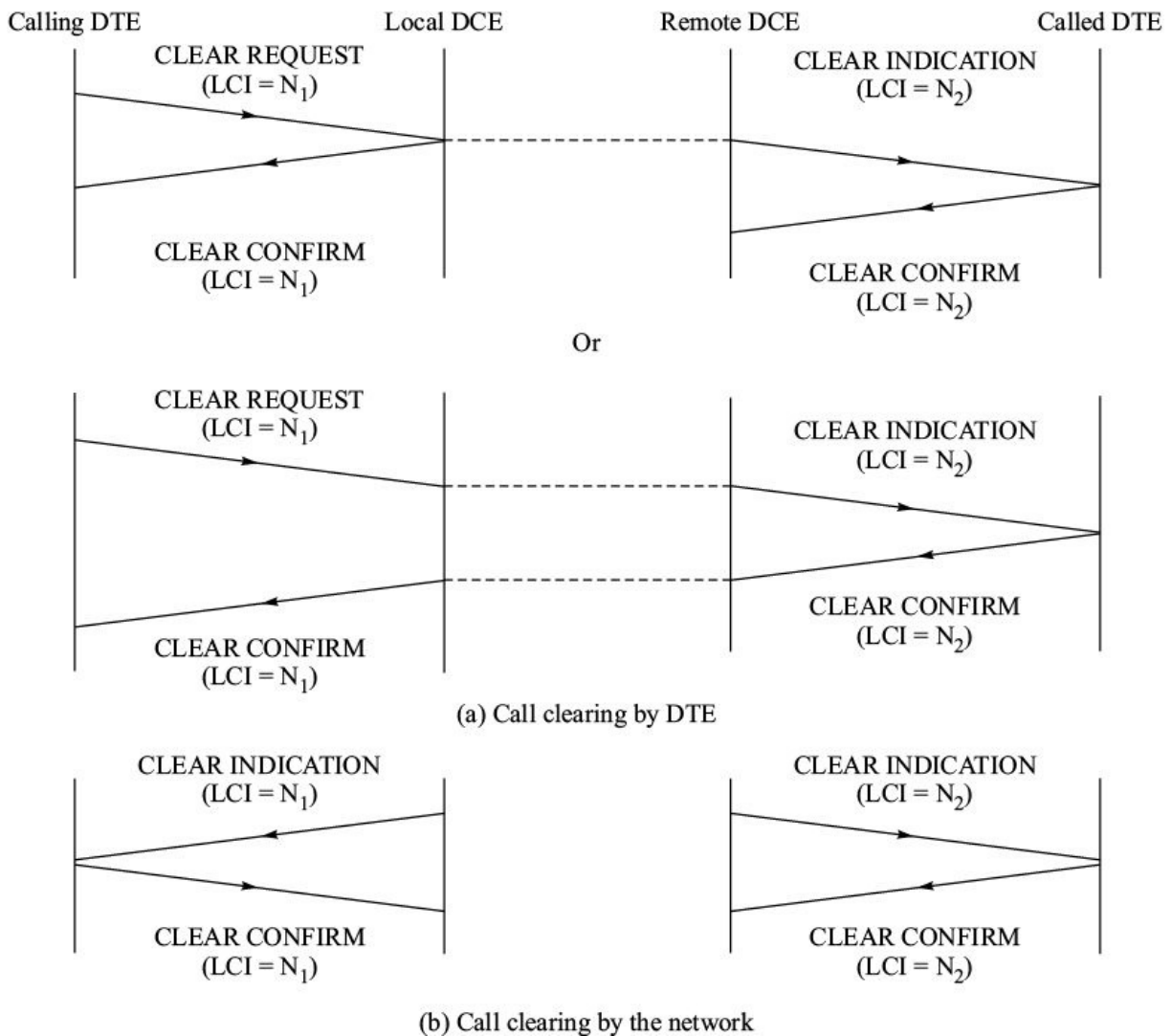


Figure 16.16 Call clearing phase.

16.5.4 Call Collision

Call collision can occur if the same logical channel is simultaneously selected by both the DTE (for an outgoing call) and the DCE (for an incoming call) from the LI/OC–HI/OC range (Table 16.1). If call collision occurs, the outgoing call is given preference over the incoming call. The incoming call is cleared by the

network (Figure 16.17).

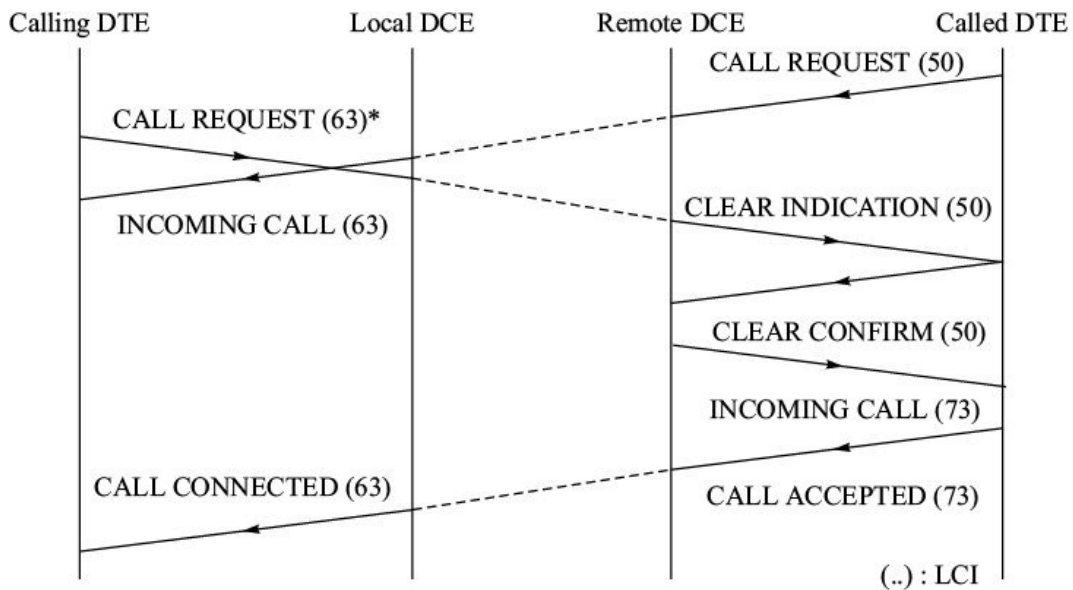
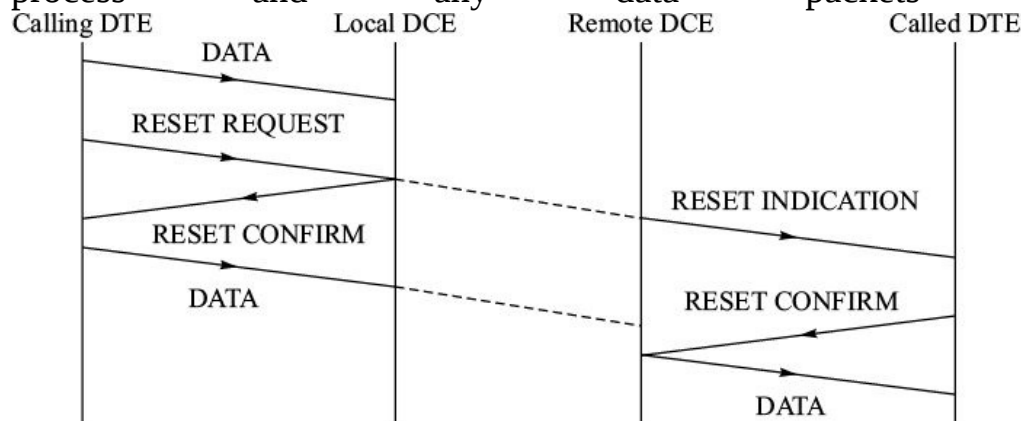


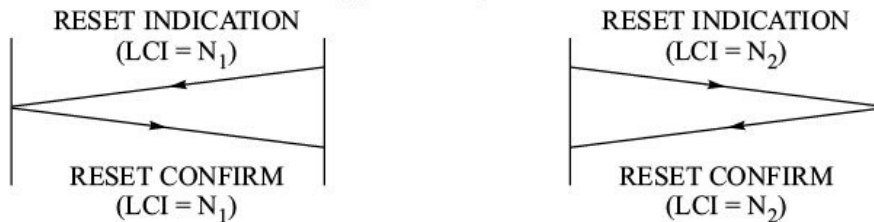
Figure 16.17 Call collision.

16.5.5 Virtual Circuit Reset

The reset procedure is used to re-initialize an SVC or a PVC. The counters P(S) and P(R) are reset. Either user is free to reset the logical channel at any time by transmitting a RESET REQUEST packet (Figure 16.18a). Reset is a destructive process and any data packets in the



(a) RESET by DTE



(b) RESET by the network

Figure 16.18 RESET procedure.

network awaiting delivery are discarded. The network can also reset a connection if it considers that a user has infringed the protocol, *e.g.* transmission of a wrongly formatted packet may result in reset. The subnetwork resets a connection by transmitting RESET INDICATION packet to both the DTEs simultaneously (Figure 16.18b).

The RESET REQUEST, RESET INDICATION, and RESET CONFIRMATION packets consist of the usual 3 octet header plus 2 additional octets which indicate the reason of reset.

16.5.6 Restart

The most destructive action a DTE or DCE can take is to transmit a RESTART packet. This packet is always transmitted on logical channel zero which is reserved for this packet. When a RESTART packet is transmitted or received, all the switched virtual circuits operating between a DTE and a DCE are cleared and the permanent virtual circuits are reset (Figure 16.19).

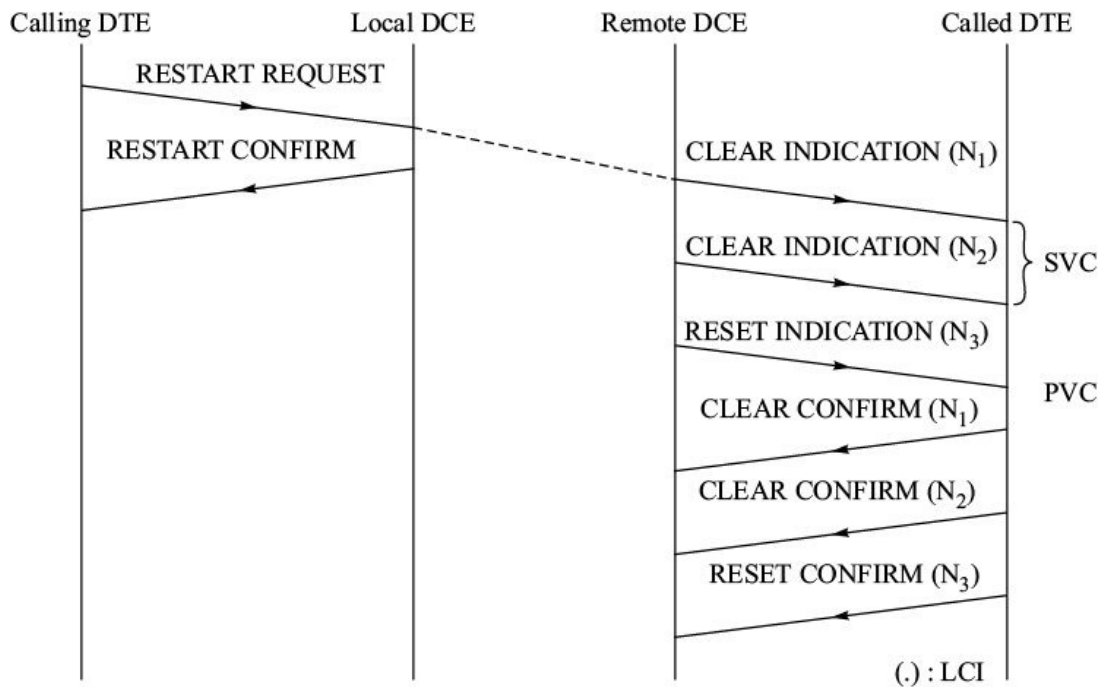


Figure 16.19 Restart procedure.

A DTE transmits the RESTART packet when it is switched on. This ensures that all logical channels start in a known state. A DCE may send the RESTART packet in the unlikely event of severe traffic congestion problem in the network.

16.5.7 Error Recovery by Timers

A number of timers have been defined at level 3 of X.25 to permit recovery when there is no response. The timers can be grouped into two categories:

- DCE timeouts
- DTE timeouts.

DCE timeouts refer to a DCE when it issues a packet and waits for the response. Similarly, DTE timeouts refer to a DTE. Table 16.3 gives a list of the timers and typical values of the timer parameters. The actual values may be different as set by the network operator.

Timer	DTE/ DCE	Timer value	Starting instance (Issue of)	Terminating instance (Receipt of)
T10	DCE	60 s	RESTART INDICATION	RESTART CONFIRMATION
T11	DCE	180 s	INCOMING CALL	CALL ACCEPTED or CLEAR REQUEST
T12	DCE	60 s	RESET INDICATION	RESET CONFIRMATION
T13	DCE	60 s	CLEAR INDICATION	CLEAR CONFIRMATION
T20	DTE	180 s	RESTART REQUEST	RESTART CONFIRMATION
T21	DTE	200 s	CALL REQUEST	CALL CONNECTED
T22	DTE	180 s	RESET REQUEST	RESET CONFIRMATION
T23	DTE	180 s	CLEAR REQUEST	CLEAR CONFIRMATION

16.5.8 Procedures for Permanent Virtual Circuit (PVC)

A PVC is always in data transfer phase. Therefore, it does not have call establishment and call clearing phases. The procedures described for data transfer phase of the SVCs also apply to the PVCs.

16.6 USER FACILITIES IN X.25

There are a number of optional facilities which a user can subscribe to. Some of the important facilities are:

- Fast select
- Reverse charging and reverse charge acceptance
- Closed user group
- Flow control parameter negotiation.

Parameters of some of the facilities are indicated at the time of subscription

while for some they are negotiated for each call during its establishment. The CALL REQUEST, INCOMING CALL, CALL ACCEPTED, and CALL CONNECTED packets have a facility field (Figure 16.7). In the facility field of these packets, the desired parameters of the facilities are indicated and accepted.

16.6.1 Fast Select

To send just one packet of data, a DTE must exchange at least four more packets to establish and later release the virtual connection. There are some applications where establishing and releasing a virtual connection can be a significant overhead. One example of such applications is credit card verification. Subscribers to the fast select facility can avoid this overhead. They can append up to 128 octets of data in the CALL REQUEST/CLEAR REQUEST packet after the facility field. Figure 16.20 shows the packets exchanged when fast select facility is used.

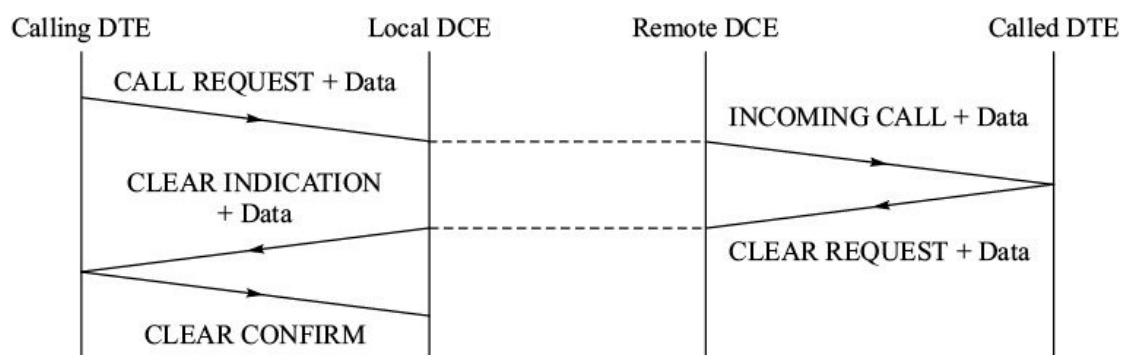


Figure 16.20 Fast select procedure.

Note that when the remote DCE sends the INCOMING CALL packet with the data field to the called DTE, it is obliged to respond with CLEAR REQUEST packet which can also have up to 128 octets of user data.

16.6.2 Reverse Charging and Reverse Charge Acceptance In reverse charging, the called user is billed for the call. By inserting the appropriate code into the facility field of the CALL REQUEST packet, a calling user who has subscribed to the facility may request for reverse charging. The remote DCE forwards the request to the called DTE in the INCOMING CALL packet if the called DTE has subscribed to reverse charge acceptance facility. If the called DTE is ready to accept the call charges, it returns CALL

ACCEPTANCE packet else it returns CLEAR REQUEST packet which results in clearing the virtual circuit. If the called DTE is not a subscriber to the reverse charge acceptance facility, the remote DCE itself clears the call.

16.6.3 Closed User Groups

A group of users may form themselves into a Closed User Group (CUG) creating a pseudo 'private' network utilizing the X.25 subnetwork facilities. As a general rule, members of a CUG can communicate only among themselves. Calls to and from DTEs outside the CUG are turned back by the DCE by sending CLEAR INDICATION as in case of reverse charging. There can be several CUGs in a data network (Figure 16.21). A CUG is formed at the time of initial subscription of individual DTEs. A user may belong to more than one CUG. Thus, the closed user groups may even overlap.

There are three variants of the basic CUG facility (Figure 16.21). Note that it is not necessary that all DTEs within a CUG should have same facilities.

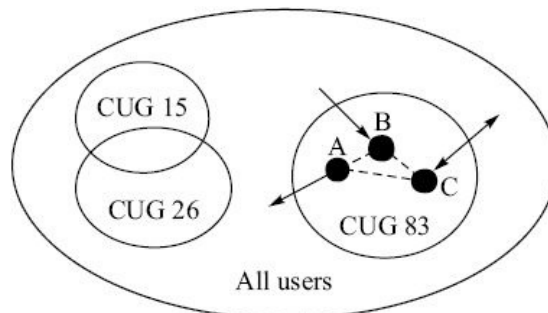


Figure 16.21 Closed user groups.

DTE with outgoing access. DTE with this facility can also originate calls to the DTEs outside the CUG (DTE A).

DTE with incoming access. A DTE with this facility can also receive calls from the DTEs outside the CUG (DTE B).

DTE with outgoing and incoming access. A DTE with this facility can also originate and receive calls from the DTEs outside the CUG (DTE C).

16.6.4 Flow Control Parameter Negotiation

The flow control parameters to be negotiated are:

- Maximum size of the DATA packets

- Window size.

The size of data packet refers to the size of data field which can at the most be 4096 octets long. These parameters need not be the same in both directions of transmission and at the ends of the virtual circuit.

These parameters are negotiated between a DTE and a DCE on a per call basis during the call set-up phase. To negotiate the packet and window sizes, the calling DTE encodes the values of these parameters in the CALL REQUEST packet, and the local DCE indicates the accepted values in the CALL CONNECTED packet. Similarly each INCOMING CALL packet indicates the proposed window and packet sizes. The called DTE accepts the parameter values in the CALL ACCEPTED packet. No relationship need exist between the parameters requested in the CALL REQUEST packet and those indicated in the INCOMING CALL packet. The valid values of parameters in the responses during parameter negotiation are given in Table 16.4.

TABLE 16.4 Valid Ranges of Negotiated Parameter Values

	Requested value	Accepted value
Window size (<i>W</i>)	$W_R \geq 2$	$2 \leq W_A \leq W_R$
	$W_R = 1$	$W_A = 2$
Packet size (<i>P</i>)	$P_R \geq 128$	$128 \leq P_A \leq P_R$
	$P_R = 128$	$P_A = 128$

Here subscript R: 0 stands for requested, and A: stands for accepted.

16.7 ADDRESSING IN X.25

The address field is present in CALL REQUEST, INCOMING CALL, CALL ACCEPTED, and CALL CONNECTED packets only. It contains addresses of the calling and called DTEs. In X.25, addressing is based on ITU-T Recommendation X.121 which provides for international addressing of DTEs. X.121 numbering scheme has the following features (Figure 16.22):

- The maximum number of digits in an international number can be 14 or less.
- The first four digits are called Data Network Identification Code (DNIC).
- The first three digits of DNIC identify the country.

- The fourth digit identifies the network within the country.
- Subsequent digits are called the National Terminal Number (NTN) and are used to identify the DTE. The maximum allowable length of NTN is 10.

The lengths of the calling and the called addresses are specified in octet 4 of the call establishment packets. For example, if the calling and called address lengths are 9 and 12 digits respectively, the octet 4 will be 1001 1100.

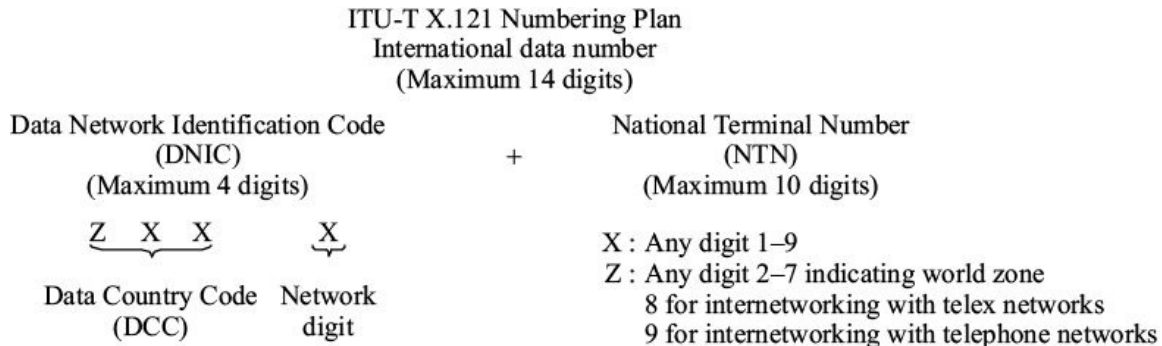


Figure 16.22 X.121 numbering plan.

The calling DTE may indicate only the called DTE address in the CALL REQUEST packet. The calling DTE address is filled by the local DCE in this case. Each digit of the address is binary coded and each octet of the address field contains two digits. If the total number of digits is odd, the last octet is padded with zeros.

EXAMPLE 16.1 Write the address field of the CALL REQUEST packet for the called DTE address 4321987654321.

Solution

International Data Number

DNIC NTN 4321 987654321 (Called DTE address of 13 digits)
--

Address fields of call set-up packets

Octet 4	0000 1101	Called DTE address length (13 digits)
Octet 5	0100 0011	Digits 4 and 3
Octet 6	0010 0001	Digits 2 and 1
Octet 7	1001 1000	Digits 9 and 8
Octet 8	0111 0110	Digits 7 and 6
Octet 9	0101 0100	Digits 5 and 4

Octet 10 0011 0010	Digits 3 and 2
Octet 11 0001 0000	Digits 1 and *

*Octet 11 is padded with zeroes to complete the address field.

16.8 PACKET ASSEMBLER AND DISASSEMBLER (PAD)

X.25 interface requires the DTE to be a packet mode device, *i.e.* a device which transmits data and receives X.25 packets. To extend services of a switched packet data network to the devices that do not have such interface, an additional intermediary device called Packet Assembler and Disassembler (PAD) is required (Figure 16.1). The interface between the terminal device and the PAD is the native-mode protocol of the device. The interface between the PAD and access node is the packet mode protocol, *i.e.* the X.25. Some basic features of the PAD are as follows:

- X.25 packet mode access allows simultaneous operation of multiple logical connections on one physical connection. Therefore, the PADs are designed to handle multiple non-packet mode devices simultaneously (Figure 16.23).

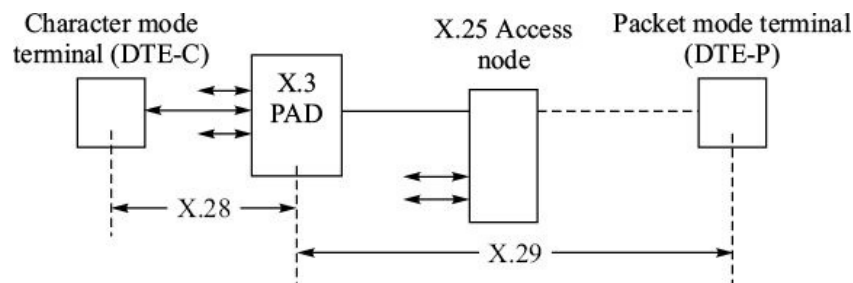


Figure 16.23 PAD as intermediary device.

- A PAD can be located either at the subscriber's premises or at the access node of the network. When the PAD is located at the access node, the PAD can be a 'stand-alone' device or integrated into the access node.
- The remote DTE for which the call is meant can be a packet mode terminal, DTE-P, or a non-packet mode terminal. In the second case, the remote DTE is connected to the access node through a PAD. Thus, there can be PADs at both the ends of a virtual circuit.
- ITU-T has produced three recommendations of the PAD for character mode terminals (asynchronous terminals). These are X.3, X.28, and X.29 (Figure 16.23).
 - ITU-T Recommendation X.3 specifies the PAD parameters and their

values.

- ITU-T Recommendation X.28 defines the procedures for exchange of commands and service signals between the DTE-C and the PAD.
- ITU-T Recommendation X.29 defines the procedures for exchange of PAD messages between the PAD and the remote DTE-P.

16.8.1 PAD Operation

An asynchronous character mode terminal (DTE-C) generates and receives data in the form of octets with start and stop timing elements. The PAD assembles the octets received from the DTE-C into a packet after removing their start and stop timing elements and adding a header to them. The incoming octets are stored in a buffer in the PAD until a decision is made to forward them as a packet. The DTE-C may ask the PAD to transmit the packet by sending a special character code. Else the PAD forwards the packet on its own after timeout or when its buffer is full.

The incoming packets received from the access node by the PAD are disassembled, start and stop timing elements are added to the octets of their data field. The octets with start/stop elements are transmitted to DTE-C. The exception to this simple situation is that the PAD needs to insert additional special characters such as Line Feed (LF), Carriage Return (CR) in the incoming data stream to enable formatting of the message on the screen of the DTE-C or the on the printer attached to it. Therefore, the PAD is configured for the DTE-C in advance by setting PAD parameters.

PAD commands. In addition to user data octets, the DTE-C sends commands to the PAD for sending certain control packets to the access node or for setting/reading PAD parameters. To distinguish the PAD commands from the user data, the DTE-C prefixes a DLE character before the command. The PAD commands are always delimited using a carriage return character. The important pad commands are:

- Commands for setting or querying PAD parameters
- Request for setting up a virtual connection
- Request for clearing or resetting a connection.

PAD service signals. Other than the disassembled user data octets, the PAD sends service signals to the DTE-C. These service signals could be the responses

to the commands or indications of control packets received by the PAD from the access node. Some important service signals are as follows:

- Indication of incoming call
- Indication of having completed call set-up or call clearing
- Parameters values as requested by the DTE-C.

PAD messages. The remote DTE-P also communicates with the local PAD through the network. This communication takes place on the virtual connection between the PAD and DTE-P using the DATA packets. PAD messages from the DTE-P are for setting and reading the PAD parameters and for inviting the PAD to clear a call after it has sent the user data octets to DTE-C.

The data packets containing PAD messages are distinguished from those containing user data meant for DTE-C by the Q bit in the GFI field of the DATA packet. The DATA packets containing PAD messages have Q bit set to 1.

PAD parameters. The PAD is required to interface with a variety of character mode asynchronous devices—dumb terminal, paper tape reader, printer or a PC. In order to serve these devices, the PAD functions need to be tailored for each specific device. This is achieved by assigning appropriate values to the parameters associated with the various PAD functions. There are twenty two PAD parameters. Some of the important PAD parameters and their typical values are given below:

- DTE-C speed : 2 (300 bps), 12 (2400 bps)
- Flow control (X-ON, X-OFF) : 1 (X-ON, X-OFF)
- Parity bit treatment : 1 (Check parity bit)
- Packet forwarding character : 2 (For CR as forwarding signal)

16.9 FRAME RELAY

When X.25 was defined, the error rates of telecommunication network were much higher than what we have today. The end systems were not as powerful as the present systems. X.25 compensated for this by using store and forward switching with hop-to-hop error and flow control at layer 2 and layer 3. Error control and flow control functions in X.25 added considerable overhead and limited the throughput that it could support.

As fibre optic network was deployed in late eighties, the quality of telecom circuits improved and X.25's error control capabilities were no longer acutely needed. At the same time, the processing power and speed of end systems vastly improved. End systems could control errors end-to-end without support from the interconnecting network. With the advent of local area networks, X.25 services proved inadequate for the high throughput LAN-to-LAN inter-connectivity. Therefore, a more efficient protocol for transporting data was required in place of X.25.

Frame relay was the first protocol that met the requirements of the changed scenario. It was originally designed for use across ISDN as a switched data service, but it evolved later as WAN technology. Frame relay is based on virtual circuit switching at layer 2 and provides connection-oriented service at layer 2. Thus one entire layer of processing is eliminated. Error and flow control functions are not part of frame relay networks. These functions are performed by the higher layer of the end systems. Studies indicate that frame relay improves the throughput by an order of magnitude when compared to X.25.

16.9.1 Frame Relay Network Topology

The frame relay provides switched virtual connection service to the end systems called DTEs (Data Terminal Equipment) (Figure 16.24). The service provider's node offering frame relay service is called DCE (Data Circuit Terminating Equipment). DTE-to-DCE communication is standardized. DCE-to-DCE communication is not standardized and may have vendor specific implementation. ITU-T, ANSI, and Frame Relay Forum have been driving the standardization of frame relay interface. Various ITU-T standards or frame relay are:

- Q.922 Link access procedure for frame mode services (LAP-F)
- Q.933 ISDN signaling specifications for frame mode bearer services
- Q.921 (LAP-D).

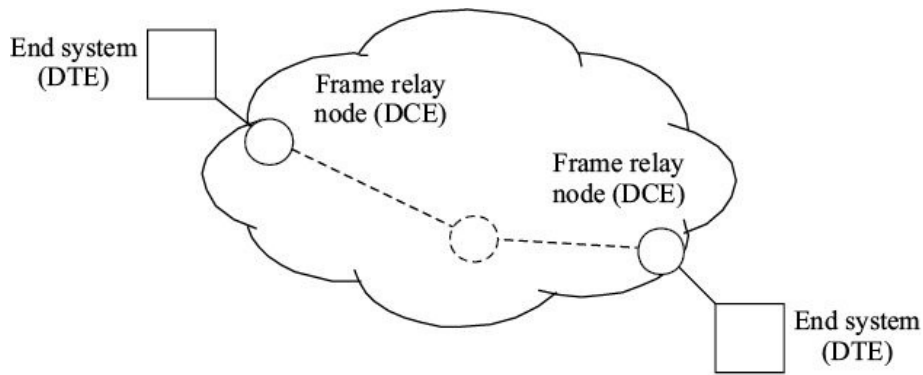


Figure 16.24 Frame relay network.

16.9.2 Frame Relay Connection

Frame relay network provides switched virtual connection services similar to those provided by X.25 network. The frame relay virtual connection is identified by a connection identifier called Data Link Connection Identifier (DLCI). This identifier is associated with frames at layer 2, unlike logical channel number of X.25, which is associated with the packets at layer 3.

Virtual connection in frame relay is established as association of DLCIs (Figure 16.25). Connection between DTEs A and B is established through frame relay nodes P, Q, and R. Association of DLCIs 21, 55 defines this connection. The network maps these two DLCIs to one another. Since internal architecture of frame relay network is vendor driven, there can be different ways for implementing this mapping.

There can be multiple simultaneous connections on each link. Frames belonging to different connections are distinguished by the DLCI value and are statistically multiplexed on a link. For example, 32, 47 is the second virtual connection being operated by end station A in Figure 16.25. The second connection is with end station C.

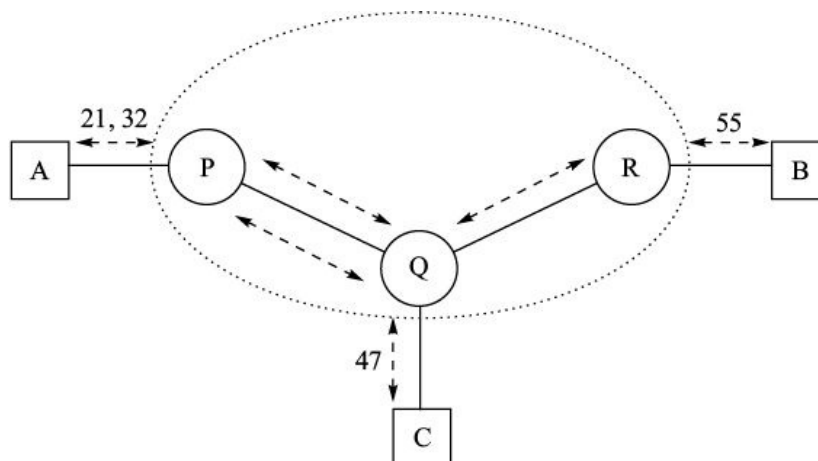


Figure 16.25 Frame relay virtual connection.

16.9.3 Frame Relay Services

There are two types of frame relay services:

- Switched Virtual Circuit (SVC) service, and
- Permanent Virtual Circuit (PVC) service.

SVC involves three phases of operation—connection establishment, data transfer, and connection release phases. PVC is established at the time of subscription and is always in data transfer phase. Many frame relay implementations are in form of access link, *i.e.* frame relay is used for connecting the Customer Premises Equipment (CPE) to the nearest IP node (Figure 16.26). The access link is provided on frame relay as a PVC.

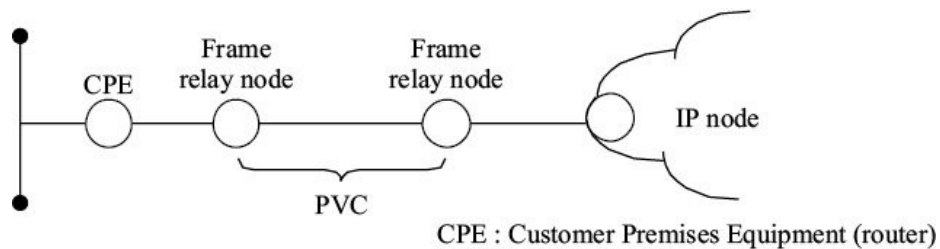


Figure 16.26 Frame relay in the access link.

16.9.4 Layered Architecture of Frame Relay Network

Figure 16.27 shows the layered architecture of frame relay network. The architecture depicts two separate planes of operation—user plane for supporting bearer services for carrying user data, and control plane which is used for signaling purposes, *i.e.* for establishing and terminating virtual connections. In X.25, these planes were not separate. Recall that call establishment, call clearing, and data packets in X.25 were sent by the same network layer.

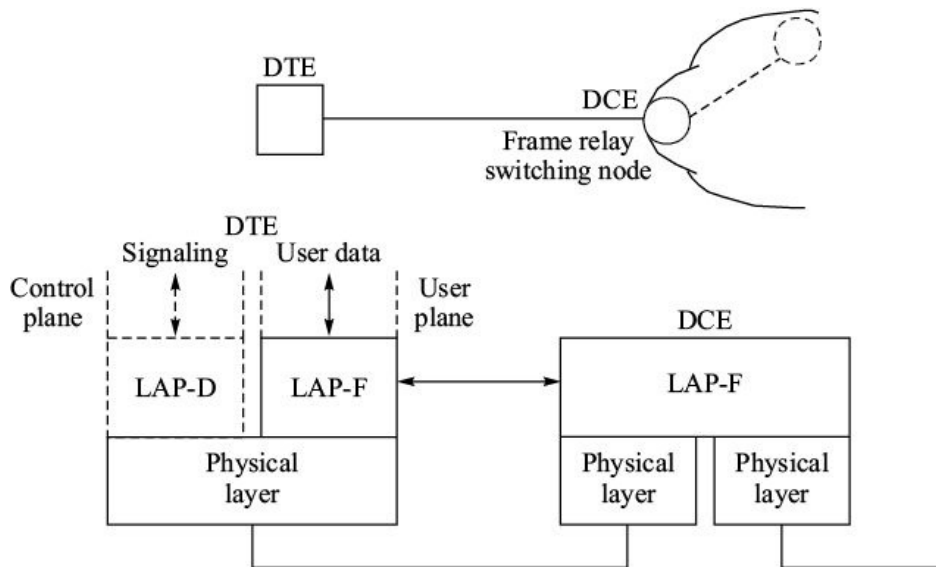


Figure 16.27 Layered architecture of frame relay.

Control plane uses LAP-D protocol for signaling at layer 2. LAP-D provides a reliable data link with error and flow control functions. LAP-D protocol was discussed in Chapter 9. Discussion on the signaling aspects of frame relay are beyond the scope of this book and therefore we will limit the discussion to the user plane architecture of frame relay.

User plane is responsible for transport of user data using frame relay service. The frame relay network uses LAP-F (link access procedure for frame mode bearer services) at the data link layer. LAP-F is documented in ITU-T Q.922, Annexure A. It specifies the communication between DTE and DCE.

Frame relay service is based on layer 2 and therefore frame relay switching nodes have only two layers for the bearer traffic—the physical layer and the data link layer (LAP-F). There is no error and flow control in the layer 2 of frame relay. Errors detected using FCS bits result in mere dropping of the frame. Instead of flow control, there is congestion control function. We will describe layer 2 protocol of frame relay later in this chapter.

Frame relay supports a variety of physical layer interfaces. The options for the physical layer defined by the Frame Relay Forum in FRF.1 are as follows:

- ITU-T V.35, V.36, V.37
- ITU-T G.703 (2.048 Mbps, 34.368 Mbps), G.704
- ITU-T X.21
- ANSI T1.403 (1.544 Mbps)
- ANSI T1.107a (44.736 Mbps).

16.10 CONGESTION CONTROL IN FRAME RELAY

Congestion in a network may occur if the data pumped into a network is more than what the network resources can handle. Congestion in a network is avoided by building flow control mechanisms. X.25 network has flow control both at data link level and network level and flow control mechanisms have inherent property of reducing the throughput and increasing the delay.

Frame relay networks were designed with a goal to have high throughput and low delay. Therefore, flow control is not built into the frame relay networks. Because there is no flow control, frame relay networks are prone to congestion. Frame relay, therefore, requires some form of congestion control.

16.10.1 Parameters for Congestion Control

When congestion occurs in a frame relay node, it simply drops the frames it cannot handle. A node operates several connections simultaneously and therefore it needs to ensure that service to a customer is not affected due to excess traffic pumped into the network by another customer. Therefore, the frame relay service for service level agreements (SLAs) is defined in terms of the following parameters:

- Access rate
- Measurement interval (T)
- Committed burst size (B_C)
- Committed Information Rate (CIR)
- Excess burst size (B_e).

Access rate. Access rate is determined by the transport link that connects the customer's DTE to the service provider's DCE. An E1 link will thus give an access rate of 2.048 Mbps.

Measurement interval (T). It is time interval over which the other parameters (B_C , B_e , CIR) are computed. Timer for measurement of T is started when the first bit of a frame is received.

Committed burst size (B_C). Committed burst rate (B_C) is the maximum number

of bits that the frame relay network is committed to transfer in time T under normal conditions. For example, if B_C 500 kilobits and T is 5 seconds, a customer can transmit 500 kilobits during an interval of 5 seconds without any worry about frame discard. The 500 kilobits need not be uniformly distributed over the 5 seconds period. If the customer has 2.048 Mbps access rate, he can send 500 kilobits in less than quarter of a second.

Committed information rate (CIR). Committed information rate is average bit rate committed to the customer. It is defined over the measurement period T and is calculated as $CIR = B_C/T$

Thus, CIR in the example given above is $500/5 = 100$ kbps.

Excess burst size (B_e). Excess burst size defines the maximum additional bits that a frame relay network will try to deliver in excess of B_C over time interval T in normal network conditions. The network will transfer these many excess bits if there is no congestion in the network.

Figure 16.28 illustrates the relationship of these parameters. Note that the frames sent by the DTE is in the form of bursts. The access data rate is always

same. After the number of bits

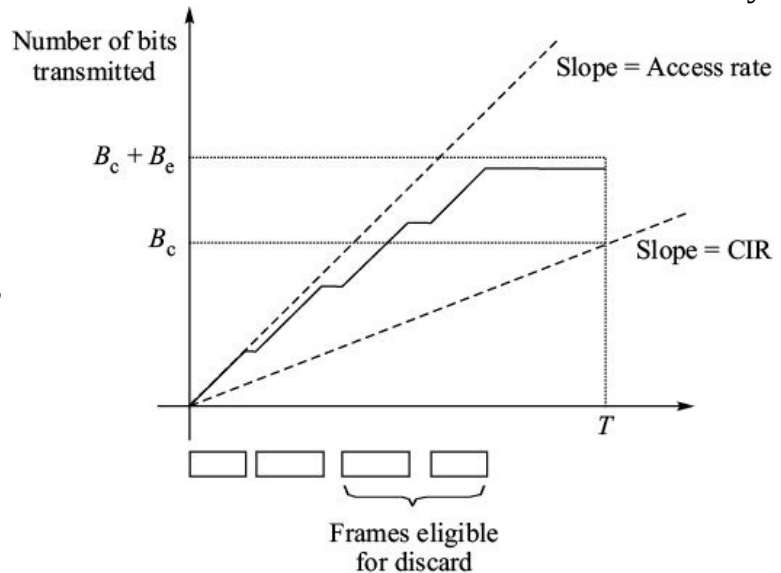


Figure 16.28 Parameters for congestion control.

sent by the customer exceeds B_C , there is no commitment the subsequent data frames will be delivered. There is Discard Eligibility (DE) bit in the frame for this purpose. This bit is set to 1 when B_C is exceeded. Thus, the last two frames in Figure 16.28, that result in B_C to be exceeded, the DE bit is set to 1 by the DCE. By setting DE bit, a node is indicating to the following node that with this

particular frame, the frame relay connection has already exceeded its CIR, and if there is congestion on the link, the frame can be dropped.

After the $(B_C + B_e)$ limit is reached, no more frames are transferred till expiry of T .

16.10.2 Congestion Control Using FECN, BECN, and DE Bits For congestion control, frame relay has three bits built into its frame:

- Forward Explicit Congestion Notification (FECN) bit
- Backward Explicit Congestion Notification (BECN) bit
- Discard Eligibility (DE) bit.

When a frame relay node notices congestion in a direction, it does three things:

- It sets the FECN bits in the frames going in that direction to 1. For example, in Figure 16.29, node B notices congestion of traffic going in the direction towards C. It sets the FECN bits in the frames going towards C to 1. By setting FECN bit, the frame relay network is sending a message to the following nodes and to the receiving end system that this frame suffered congestion on the route. The destination end system needs to take some action to reduce the data rate. The destination end system can enforce flow control at higher layer. For example, the transport layer of the destination end system can reduce the window size of the transport layer of the sending end system.
- It sets the BECN bits in the frames having the same DLCI and going in opposite direction to 1. In Figure 16.29, node B sets the BECN bits in the frames going towards A to 1 when it notices congestion of frame traffic going in the direction towards C. By setting the BECN bits to 1, the node is sending an indications to the source end system that there is congestion in the direction towards the destination end system. The source end system is required to take necessary steps for reducing the data rate. The action is taken at a higher layer as the data link layer has no flow control mechanism built into it.
- It discards those frames going in the direction towards C (Figure 16.29):
— that have DE bit set to 1, or

— whose Committed Information Rate (CIR) has already been exceeded. Discard Eligibility (DE) bit is set to 1 in those frames which can be discarded if there is congestion. DE bit can be set to 1 by an end user in those less important frames that can be discarded when congestion occurs or it is set to 1 by a frame relay node in those frames that exceed CIR.

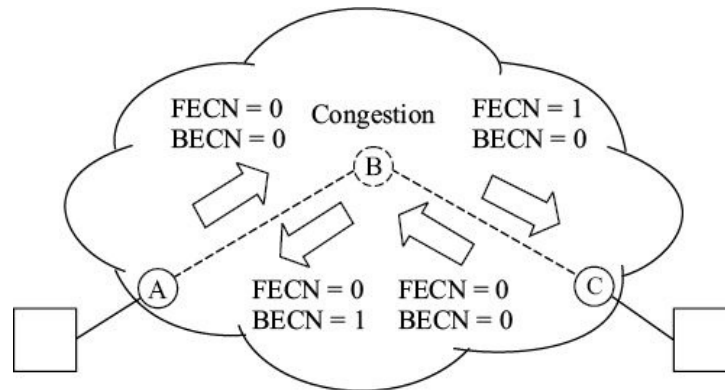


Figure 16.29 Congestion control.

16.11 FRAME FORMAT IN FRAME RELAY

The format of LAP-F frame is based on HDLC frame and is shown in Figure 16.30. The control field of HDLC frame is not there as error and control functions are not provided. The address field contains the following DLCI and congestion control bits: **Flag (1 octet)**. The flag identifies start and end of the frame. It is 01111110.

Data link connection identifier (DLCI, 10–23 bits). 10-bit DLCI is the virtual connection identifier. The first octet contains six most significant bits of DLCI. The rest four bits of DLCI are in the second octet. DLCI can be up to 23 bits and is accommodated in up to four octets (Figure 16.30).

Discard eligibility bit (DE, 1 bit). This bit marks a frame eligible for discard should congestion occur in the network. It can be set by the DTE or DCE.

Forward explicit congestion notification (FECN, 1 bit). When a frame relay switch encounters congestion towards a direction, it sets the FECN bit of a frame going in that direction to 1.

Backward explicit congestion notification (BECN, 1 bit). When a frame relay switch encounters congestion in the onward direction, it sets the BECN bit of a frame going in opposite direction to 1.

Command/response (C/R, 1 bit). C/R bit is not used.

Extended address (EA, 1 bit). This is used for extending the header field. When EA bit is zero, it indicates that the next octet is also part of the header. In the last octet of the header, the EA bit is set to 1.

Information (variable). It contains the data packet received from the network layer. The information field is of variable size and can have maximum length of 8192 octets. Frame Relay Forum has defined maximum default size of 1600 octets.

FCS. Frame Check Sequence (FCS) is used for error detection and contains CRC bits.

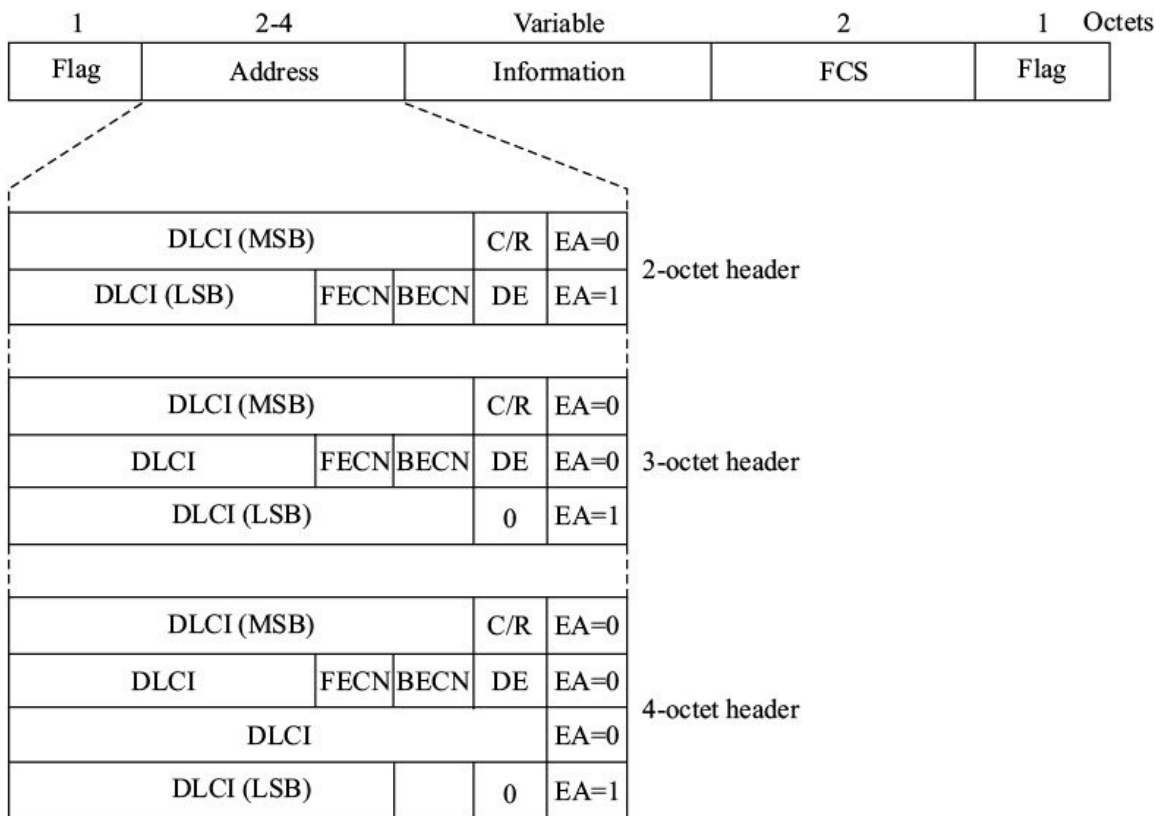


Figure 16.30 Frame format.

16.11.1 Reserved Data Link Connection Identifiers

10-bit DLCI field can accommodate 1024 identifiers ranging from 0 to 1023. 32

of these are reserved and the balance can be used for PVC and SVCs (Table 16.5).

Range	Number	Use
0	1	Call control signaling
1–15	15	Reserved
16–1007	992	SVC, PVC
1008–1022	15	Reserved
1023	1	Reserved

16.11.2 Basic Operation of LAP-F

The user data (e.g. IP packet of the network layer) is encapsulated in LAP-F frame and is sent to the frame relay node. The LAP-F layer of the frame relay node checks for errors using FCS and if there is any error, the frame is discarded. If there is no error, it consults the switching table and forwards the frame to the next node after making the following changes:

1. The DLCI field is changed to the value as per the switching table. The switching table is populated with DLCI values at the time of establishing the virtual connection.
2. The congestion control bits FECN and BECN bits are modified if there is congestion. DE bit is set if the CIR is exceeded.
3. If there is congestion and DE bit already set in the received frame, the frame is discarded.

16.11.3 IP Encapsulation

Frame-relay can encapsulate various types of network layer PDUs in its information field. Internet Protocol (IP) which we discuss in the next chapter, is the most widely deployed network layer protocol. Encapsulation of IP packet in a LAP-F frame is shown in Figure 16.31. Note that there is control field also in this frame. It corresponds to UI (Unnumbered Information) frame and carries value 0x03. The content of information field is identified by Network-Layer Protocol-Identifier (NLPID) field. It is 0xCC for IP packet.

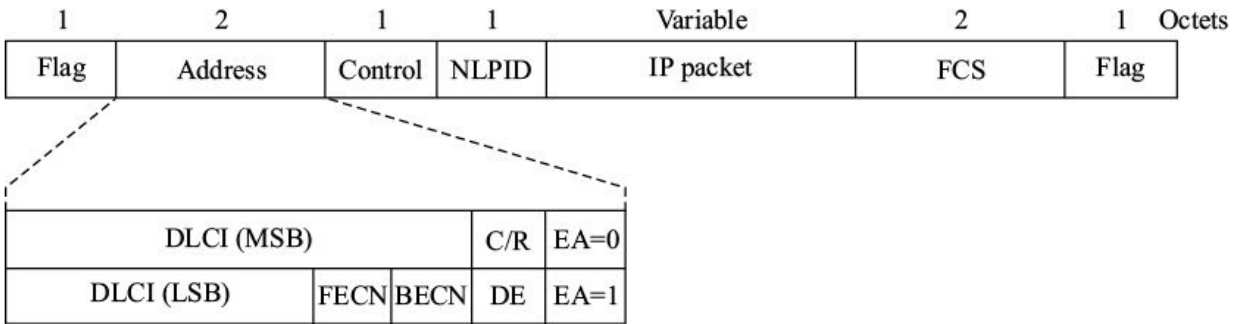


Figure 16.31 IP encapsulation in LAP-F frame.

16.12 ASYNCHRONOUS TRANSFER MODE (ATM)

There can be two modes of information transfer:

- Synchronous transfer mode
- Asynchronous transfer mode.

Time Division Multiplexer (TDM), which we studied in Chapter 4, is synchronous transfer mode. In TDM each user is assigned a time slot. Whenever the assigned time slot comes up, his data is transported. If he has nothing to send, the time slot goes empty. If he has more data to send than a time slot can accommodate, he cannot use any other empty time slot. In a switched connection, a fixed assignment of time slots is made for each connection by the TDM switch (Figure 16.32). The switch transfers the contents of one time slot to another as per the assignment during the connection set-up.

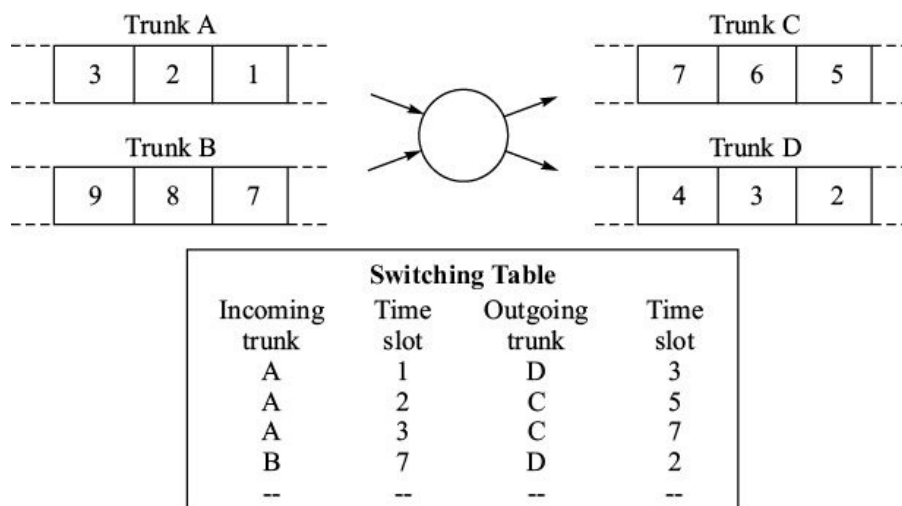


Figure 16.32 Synchronous transfer mode.

The main features of synchronous transfer mode are:

- There is constant transport delay, *i.e.* the bits are delivered after a fixed delay. Therefore, there is no jitter. Jitter is time variations in data delivery.
- There is fixed assignment of time slots. Therefore, time slots cannot be shared. If there is no data to be sent, time slot is wasted.

In asynchronous transfer mode, the information (voice, video or data) is sent in a small fixed size packet called cell (Figure 16.33). Each cell has a header of 5 octets and payload of 48 octets. The header contains an identifier that identifies the connection to which a cell belongs. ATM switch switches the cells based on the identifier. The identifier has local significance as in X.25 and frame relay networks. The ATM switch maintains a switching table that associates the identifiers on the incoming and outgoing links.

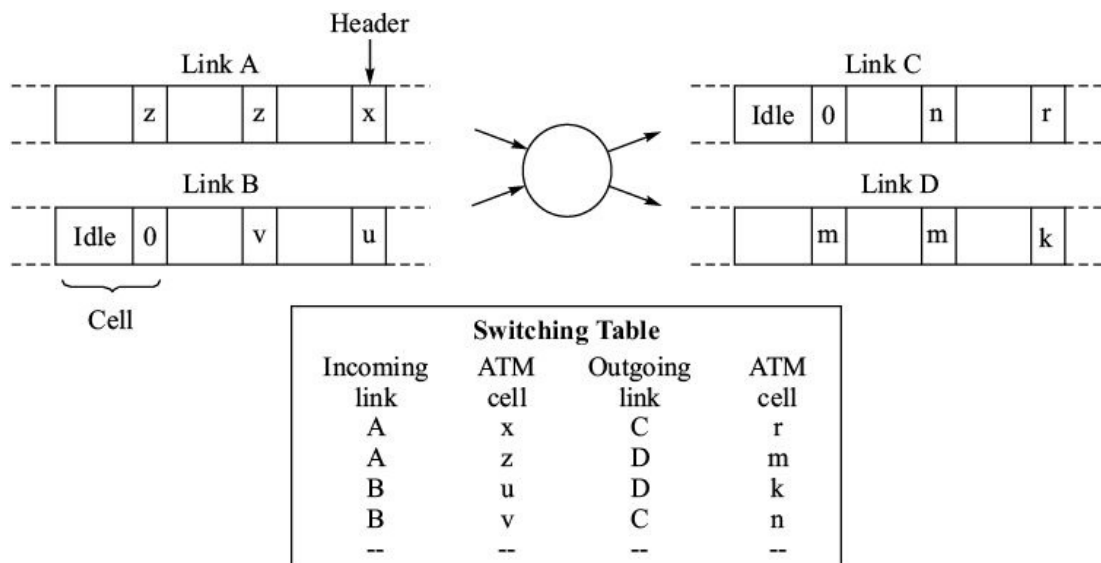


Figure 16.33 Asynchronous transfer mode.

Note that a connection may dynamically have additional cells if there is a burst of data, *e.g.* cells z in Figure 16.33. The cells are statistically multiplexed. It is possible that there are instants when none of connections generates any cell. Such inter-cell gaps are filled with idle cells of same size (53 octets). Idle cells have identifier equal to 0.

The main features of ATM are:

- Cell transport is based on switched virtual circuit concept. Therefore, statistical multiplexing is possible.
- Cells are of fixed size and therefore hardware implementation of the switch is possible. Hardware switch implementation can have very high speed.

- The transport is connection oriented and thereby the cells are delivered in sequence and delay is predictable and bounded.
- ATM is suitable for voice, video, and data communications and can integrate these services on one network.

16.12.1 UNI and NNI

The interface between the ATM end system and the ATM switched is called User-Network Interface (UNI). The interface between two ATM switches is referred to as Network-Network Interface (NNI) (Figure 16.34). The formats of ATM cells at UNI and at NNI are somewhat different as we will see later.

16.12.2 ATM Virtual Channel Connection (VCC)

As mentioned above, ATM is a connection-oriented switched virtual circuit technology. Virtual circuits are identified using an identifier which consists of the following two parts:

- Virtual Path Identifier (VPI)
- Virtual Channel Identifier (VCI).

VPI is 8 or 12 bits long and VCI is 16 bits long. The identifier has local significance on a link. The ATM switch translates the identifier to a new value as indicated in the switching table before forwarding an ATM cell to the next node (Figure 16.34). We have already seen this mechanism in X.25 and frame relay networks. This function is carried out in the ATM layer of the switch.

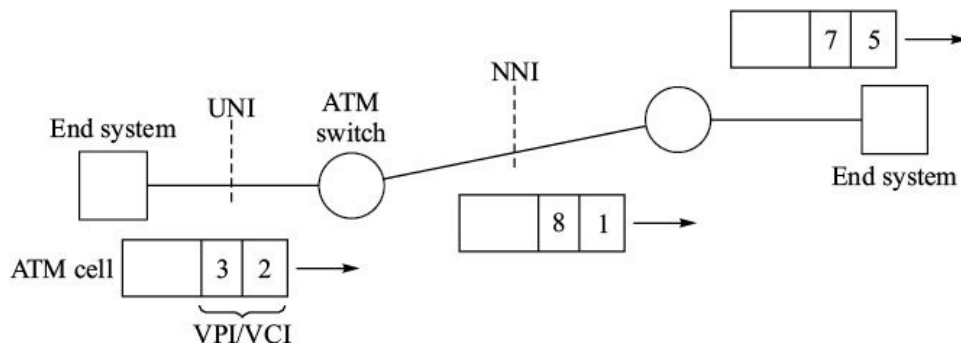


Figure 16.34 Virtual circuit identifiers in ATM.

A connection between two end points is called Virtual Channel Connection (VCC). It is made up of a series of virtual channel links that extend between the ATM switches and end systems to ATM switches. For example, the VCC

between the two end systems in Figure 16.34 consists of virtual channel links 3, 2 8, 1 7, 5. Virtual channel connections may be:

- between end systems for user applications,
- between an end system to ATM node for user-network applications, and
- between two ATM nodes for traffic management and routing.
- A VCC has the following properties:
- A VCC user is provided with a Quality of Service (QoS) specified in terms of parameters such as Cell-Loss Ratio (CLR) and Cell-Delay Variation (CDV).
- There can be two types of virtual channel connections, Permanent Virtual Channel Connection (PVC) and Switched Virtual Channel Connection (SVC). SVC has usual three phases of operation—call establishment, data transfer, and call termination phase. PVC is always in data transfer mode.
- A VCC maintains cell-sequence integrity.
- A VCC allows negotiation of service parameters.

16.12.3 Virtual Path Connection (VPC)

A group of virtual channels going along on the same route can be given same VPI. The ATM node switches this group of virtual channels based on the common VPI. A virtual path is bundle of virtual channels, all of which are switched together based on the common VPI. Figure 16.35

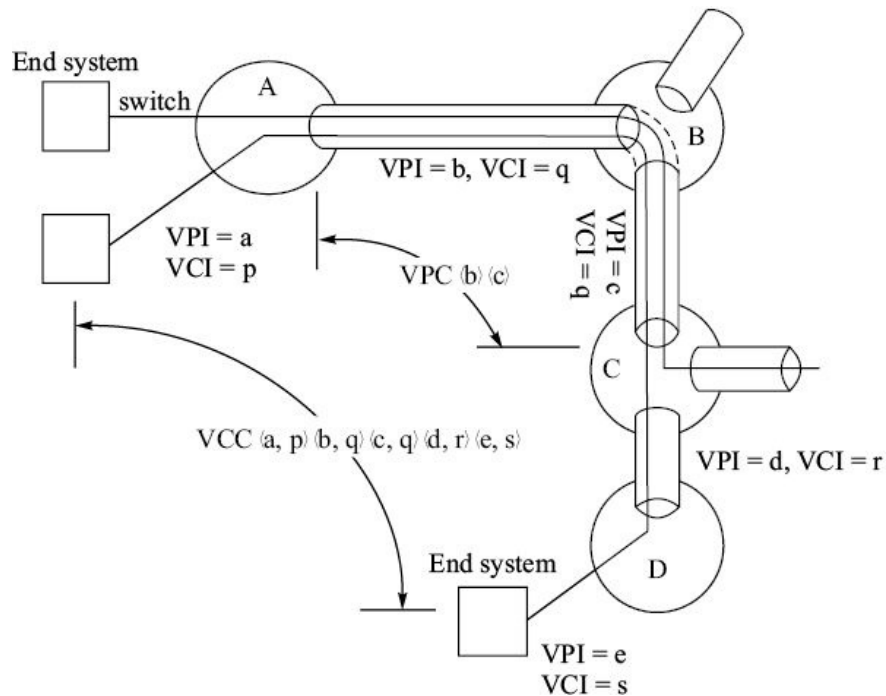


Figure 16.35 Virtual channel and path connections.

shows a virtual path from ATM switch A to ATM switch C via switch B. Just like VCC, we call such path as Virtual Path Connection (VPC). Note that VCI (= q), when within the VPC, is not changed at the intermediate switch B.

16.13 LAYERED ARCHITECTURE IN ATM

The ATM standard defines three layer architecture (Figure 16.36):

- Physical layer
- ATM layer
- Adaptation layer.

The end systems have all the three layers and the ATM nodes have only two layers—the ATM layer and the physical layer (Figure 16.36).

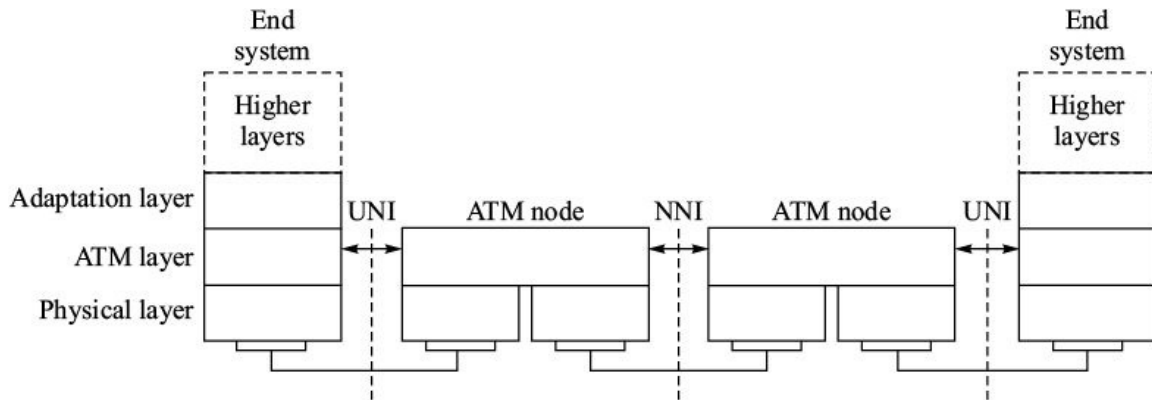


FIGURE 16.36 Layered architecture of ATM network.

The layered architecture shown in Figure 16.36 is only for the user plane, *i.e.* for carrying user information. For call control (establishing and terminating connections) there is control plane architecture just like we saw in frame relay. The higher layers of control plane rest over the common adaptation layer shown in Figure 16.36. We will skip the description of the control plane as it is beyond scope of this book.

16.13.1 Physical Layer

The ATM physical layer is analogous to the physical layer of the OSI reference model. The basic purpose of the layer is to map the ATM cells received from the ATM layer onto the transmission frame (e.g. SDH frame) and transmit the frame as electrical/optical signal. At the receiving end the electrical/optical signals are received, ATM cells are retrieved from the transmission frame and handed over to the ATM layer.

The physical layer is divided into two sublayers (Figure 16.37):

- Physical Media Dependent (PMD) sublayer
- Transmission Convergence (TC) sublayer.

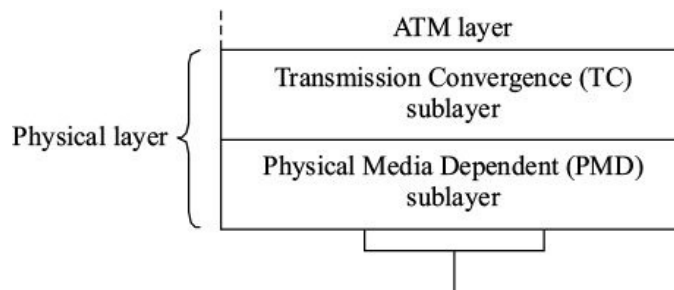


Figure 16.37 The physical layer in ATM networks.

PMD sublayer. Its basic function is transmission and reception of bits over the physical transmission medium. It involves:

- conversion of bits into electrical/optical signals,
- line coding,
- bit synchronization, and
- specifying connector types and signal levels.

The number of physical interfaces—PDH (E1, E2, E3, E4) and SDH (STM-1, STM-4, STM-16), has been specified for the ATM networks.

TC sublayer. Its basic function is to map ATM cells into the transmission frame of the transmission system being used. At the receiving end the ATM cells are retrieved from the transmission frame. This basic function involves several other associated functions described below as follows: *Cell rate decoupling.* This function decouples ATM cell generation rate and the payload capacity of transmission frame. This is achieved by inserting idle ATM cells to fill the gaps (Figure 16.33). At the receiving end the idle ATM cells are discarded.

HEC generation and verification. This function generates HEC at the one end of the physical link and checks for errors in the header of the ATM cell at the other end. Cells with more than one error in the header are discarded. Figure 16.38 illustrates the formation of complete ATM cell. The ATM layer sends the first four octets of the header and the 48-octet ATM payload to the TC sublayer. The fifth octet of the ATM header, HEC is generated and inserted by the TC sublayer. The physical layer at the other end hands over the first four octets of the header that have been verified by TC sublayer and the 48-octet payload the ATM layer.

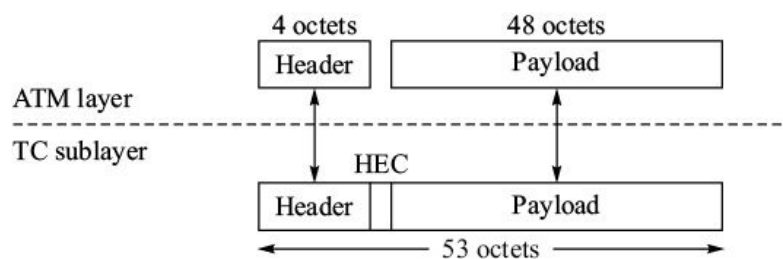


Figure 16.38 HEC generation and verification.

Cell delineation. Cell delineation function enables locating cell boundaries within the payload of a transmission frame. Header Error Check (HEC) octet in the ATM header is used for this purpose. As bits arrive, HEC check is

continuously performed on the last 40 bits. When a valid HEC check is successful, it is assumed that the last octet received was HEC octet. After that cell by cell HEC check is made for the next five cells. If HEC check is successful on these cells, it is assumed that HEC position has been correctly detected. Once HEC is known, cell boundaries are readily determined since HEC is always the fifth octet of the 53-octet cell.

Transmission frame adaptation. This function maps ATM cells onto the payload of the transmission frame (Figure 16.39).

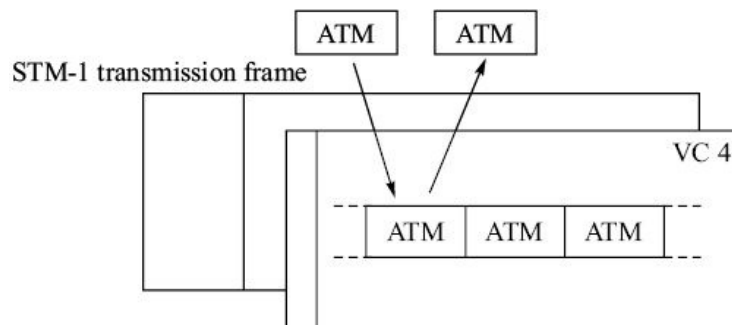


Figure 16.39 Mapping of ATM cell on a transmission frame.

16.13.2 ATM Layer

Combined with the adaptation layer, the ATM layer is roughly analogous to the data link layer of the OSI reference model. It constructs the ATM cell and forwards the cell to the next node. The main functions carried out by the ATM layer include the following: *Cell switching.* Cell switching is carried out at ATM layer based on VCI/VPI information by the node.

Policing. Policing function monitors the behaviour of traffic stream of each virtual connection. If a user is sending more cells in a given time than contracted, policing function will enable discard of excess cells. We saw similar function being performed in frame relay where frames above EIR were discarded.

Congestion control. If there is congestion at any node, the ATM layer forwards this information to the next node just like in frame relay. A cell can be marked for discard by the ATM layer should congestion occur.

ATM Cell Format. The size of ATM cell is an important factor that determines the quality of service that can be provided by the network. A large cell size gives better payload to header overhead ratio. But the delays are more variable, and loss of a cell impacts quality of service more. Smaller cells overcome these

problems, however, the ratio of payload to the header overhead is reduced. As compromise between these conflicting requirements, size of the ATM cell has been standardized at 53 octets.

Format of ATM cell is shown in Figure 16.40. It consists of 5-octet header and 48-octet payload. The header consists of 6 or 7 fields. Generic flow control is the additional field in the header of ATM cells that are exchanged at the User-Network Interface (UNI). This field is replaced with additional VPI bits at the Network-Network Interface (NNI).

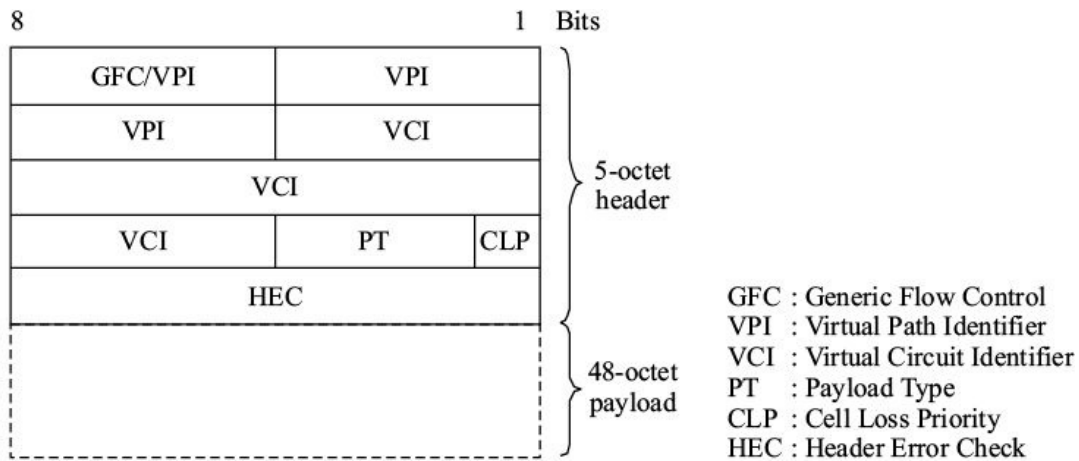


Figure 16.40 Format of ATM cell.

Various fields of the ATM cell are as follows.

Generic flow control (GFC, 4 bits). This field was provided for flow control function between the ATM end system and the ATM switch. It is typically not used and is set to its default value 0. For flow control, special flow-control cells are used instead.

Virtual path identifier (VPI, 8/12 bits). VPI is 8 bits or 12 bits depending on the interface type. UNI interface has 8-bit VPI field and NNI interface has 12-bit VPI field. It identifies a particular Virtual Path Connection (VPC).

Virtual channel identifier (VCI, 16 bits). VCI along with VPI identifies a virtual channel connection. It is similar to logical channel identifier in X.25 and DLCI in frame relay.

Reserved values of VPI and VCI are given in Table 16.6.

VPI VCI		Function
0-16		
0	18-	ITU-T

0	31	ATM forum
0	0	Idle cell
0	3	Link OAM (Operation, Administration, Maintenance) End-to-end OAM (Operation, Administration, Maintenance) Signaling
0	4	ILMI (Interim Local Management Interface)
0	16	LANE (LAN Emulation)
0	17	PNNI (Private Network-Network Interface)
0	18	

Payload type (PT, 3 bits). First bit indicates the type of information in the payload field (Figure 16.41). The cells containing user data have this bit equal to 0. The cells containing management data have this bit equal to 1.

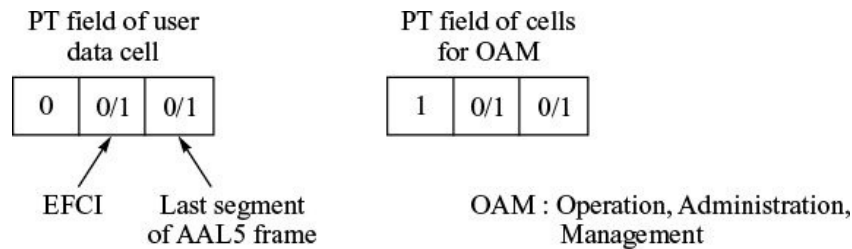


Figure 16.41 PT field of the ATM cell.

In the cells containing user data,

- the second bit of the PT field is called Explicit Forward Congestion Indication (EFCI) bit. It indicates the status of congestion. If it is 0, there is no congestion. It is set to 1 if there is congestion.
- the third bit indicates whether the cell is the last cell in a series of cells that represent a single AAL5 frame. It is set to 1 in the last cell of the frame.

PT field in the cells containing management data is coded as follows:

100	Cell containing OAM data for management of a link
101	Cell containing OAM data for end-to-end management
110	Resource management cell
111	Reserved

Cell loss priority (CLP, 1 bit). This bit is similar to DE bit of frame relay. A user may set CLP bit to 1 in the low priority cells so that if there is congestion in the network, the ATM switch may drop these cells. The ATM switch may also set this bit to 1 when a user sends more than the contracted information rate.

Header error check (HEC, 8 bits). Header checksum is generated using CRC polynomial $x^8 + x^2 + x + 1$ over the first four octets of the header. It can correct

single bit errors and detects multi-bit errors in the header. HEC field is introduced and processed by the physical layer.

Payload (48 octets). Payload contains 48-octet data unit received from the next higher layer which is Segmentation and Reassembly (SAR) sublayer. The idle ATM cells introduced by the physical layer contain 48-octet long sequence of 0x55 pattern (01010101).

16.14 ATM ADAPTATION LAYER (AAL)

Combined with the ATM layer, the adaptation layer is roughly analogous to the data link layer of the OSI reference model. The ATM layer transports 48 octet payload to the destination. The payload may contain user data from real time applications or simply a file transfer data. ATM layer does not differentiate between various types of user data in its treatment. This independence of ATM layer functionality is achieved by building rest of the functions required for transporting different types of traffic in the adaptation layer. The basic functions carried out by the adaptation layer are:

- It segments user data into chunks of 48 octets that are loaded as payload in the ATM cell. Reassembly of the segments is also carried out in the adaptation layer.
- It detects errors in the user data and takes suitable action which may be discarding the data, regeneration of user data or other actions depending on the type of traffic. Note that ATM layer does not bother about the errors in payload.
- It filters time jitter (explained in the next section) and carries out synchronization functions for the traffic that has time related issues (e.g. voice traffic).

Before we examine these functions in detail, we must first understand the traffic classification.

16.14.1 Traffic Classification

Various user applications that generate traffic puts different set of transport service requirements. For example, transporting voice traffic is different from transporting file transfer traffic. The adaptation layer takes care of these differences and makes ATM layer independent of type of traffic. The traffic is

broadly classified as:

- Constant Bit Rate (CBR) traffic
- Variable Bit Rate (VBR) Traffic.

Constant bit rate (CBR) traffic. A CBR application generates a traffic stream that requires ATM cells to be transmitted at almost uniform time spacing and expects that the receiver will receive the stream with a small delay jitter (Figure 16.42).

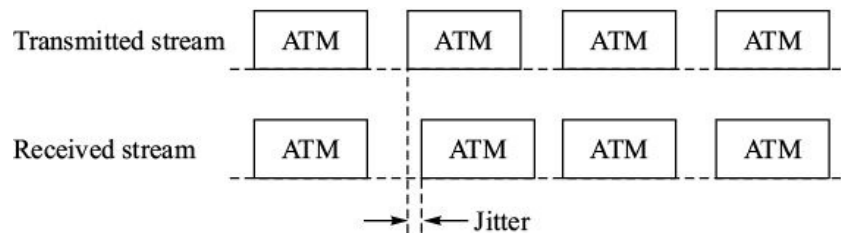


Figure 16.42 Jitter in ATM.

Examples of CBR applications are:

- Voice telephony
- Emulation of transmission link, *e.g.* E1.

CBR applications are connection-oriented.

Variable bit rate (VBR) traffic. A VBR application generates traffic in bursts. It has constant long term average bit rate but has occasional bursts of high traffic. VBR applications expect that the network will carry its burst of high traffic rate but they can tolerate some delay jitter. VBR applications can be connection-oriented or connectionless and, based on service requirements, are categorized as: (a) Connection-oriented with low timing jitter

(b) Connection-oriented without timing relationship

(c) Connectionless without timing relationship.

One typical application of category (a) is compressed video (*e.g.* MPEG). Examples of categories (b) and (c) are X.25 service and IP service respectively, when ATM is used as transport mechanism for these networks.

In ATM, the CBR and VBR traffic with their subcategories is mapped to four traffic classes—Class A, Class B, Class C, and Class D as shown in Table 16.7. The adaptation layer in ATM adapts the various traffic classes to a common

ATM layer for transport. Four different types of adaptation layers are required, one AAL for each type of traffic class.

AAL	Class	Features
AAL 1	Class A	Constant Bit Rate (CBR), connection-oriented, with timing relation.
AAL 2	Class B	Variable Bit Rate (VBR), connection-oriented with timing relation.
AAL 3	Class C	Variable Bit Rate (VBR), connection-oriented without timing relation.
AAL 4	Class D	Variable Bit Rate (VBR), connectionless without timing relation.
AAL 3/4	Classes C and D	Variable Bit Rate (VBR) without timing relation.
AAL 5	Classes C and D	Simplified version of AAL 3/4 for data communications.

While standardizing the AAL layers, it was felt that there were few differences in terms of needed functions of AAL 3 and AAL 4 and these adaptation layers were combined as common AAL 3/4. But AAL 3/4 was too complex and introduced a lot of overhead not really needed for data communication. Therefore, a simplified version AAL 5 was introduced primarily for data communications.

16.14.2 Structure of the Adaptation Layer The adaptation is divided into two sublayers (Figure 16.43):

- Convergence Sublayer (CS)
- Segmentation and Reassembly (SAR) Sublayer.

The Convergence Sublayer (CS) carries out functions relating to error detection, cell delay variations, and synchronization. It adds a header and trailer to the user data forming a CS-PDU (Figure 16.43).

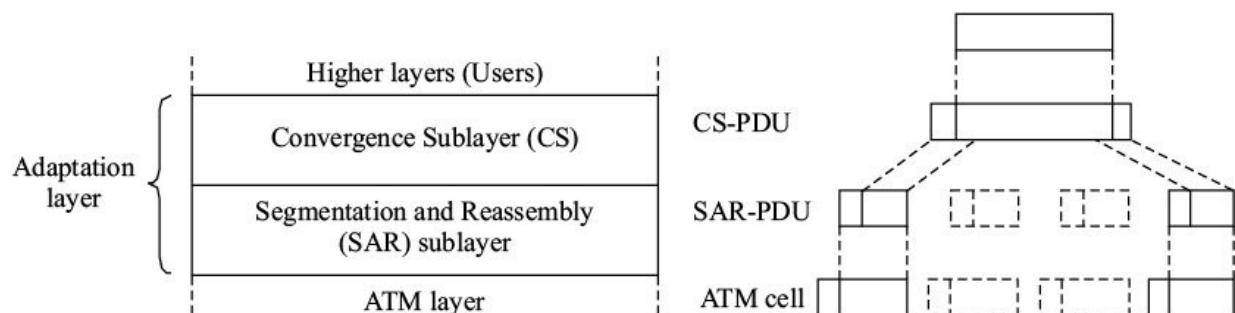


Figure 16.43 Sublayering of the ATM adaptation layer.

Segmentation and Reassembly (SAR) sublayer segments the CS-PDU and produces SAR-PDUs which are 48 octets long and fit into the ATM cell. SAR-PDUs may or may not have headers or trailers depending on type of the

adaptation layer.

16.14.3 ATM Adaptation Layer 1 (AAL 1) AAL 1 offers constant bit rate connection-oriented service to the users. CBR service is required for digital voice, video signals or for leased circuit emulation (e.g. E1 emulation). CBR service requires that timing relation between the source and receiver is maintained. Therefore, timing jitter control and synchronization of clocks at the source and receiver are the two very important functions performed by this layer.

Convergence sublayer (CS). Basic function of the convergence sublayer of AAL 1 is time and clock synchronization and as such no CS-PDU has been defined.

Segmentation and reassembly (SAR) sublayer. Format of SAR-PDU is shown in Figure 16.44. It consists of a header of one octet and payload of 47 octets. Various fields of the header are as follows:

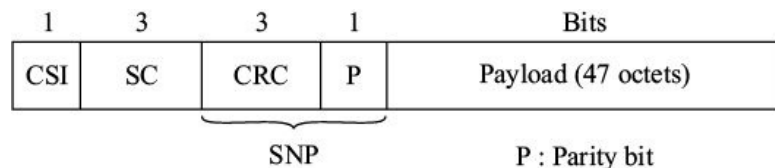


Figure 16.44 Format SAR-PDU of AAL 1.

Convergence sublayer indication (CSI, 1 bit). At the sending end the CS sublayer provides a 4-bit timing stamp value. This value is the time difference with respect to a reference clock. Four consecutive odd numbered (1, 3, 5, and 7) SAR-PDUs carry this 4-bit information in the 1-bit CSI field. At the receiving end the time stamp is recovered and made over to the convergence sublayer for synchronization.

Sequence count (SC, 3 bits). This field carries modulo-8 sequence number that is used for detecting lost SAR-PDUs and for giving an 8-frame structure to SAR-PDU. The 8-frame structure is required for retrieving the information present in the CSI fields of consecutive SAR-PDU.

Sequence number protection (SNP, 4 bits). This field consists of 3-bit CRC code calculated over the four bits of CSI and SC fields, and an even parity bit for the SAR header. Together these bits provide error detection and single-bit error

correction capability.

16.14.4 ATM Adaptation Layer 2 (AAL 2)

AAL 2 is intended for VBR applications that require timing relation to be maintained between the source and destination. AAL 2 has not been defined so far.

16.14.5 ATM Adaptation Layer 3/4 (AAL 3/4)

AAL 3/4 supports both connection-oriented and connectionless data service. It consists of convergence and SAR sublayers (Figure 16.45a).

Convergence sublayer (CS). Functions of the convergence sublayer are as follows:

- Detection of missing user data segments
- Indicating buffer resources required at the receiver for user data message.

The convergence sublayer of AAL 3/4 adds a header and trailer to the user data frame. The header and the trailer consist of the following fields (Figure 16.45b).

Common part indicator (CPI, 1 octet): It is set to 0 in the current version of AAL 3/4.

Begin tag (BT, 1 octet): The value in BT field matches with the value in ET field. These two fields identify the first and the last segments of CS-PDU.

Buffer allocation (BA, 2 octets): This field tells the receiver what size of buffer is needed for the incoming CS-PDU.

Pad: Padding bytes are added to make the length of CS-PDU a multiple of 44 octets. The number padding bytes can be from 0 to 43.

Alignment (AL, 1 octet): This is a filler field to make the trailer 4 octets long.

Ending tag (ET, 1 octet): This field matches with BT field and marks the last segment of the CS-PDU.

Length (L, 2 octets): This field indicates length of user message (excluding the padding bytes). It enables separating the padding bytes and the user data.

Segmentation and reassembly (SAR) sublayer. Figure 16.45c shows the format of SAR-PDU of AAL 3/4 layer. This sublayer segments the CS-PDU into

44-octet segments and adds header (2 octets) and trailer (2 octets) to each segment, making required 48-octet ATM layer payload. At the receiving end the segments are reassembled to form CS-PDU. In addition to segmentation, this sublayer carries out error detection using CRC and detects missing segments.

Various fields that constitute SAR header and trailer are described below.

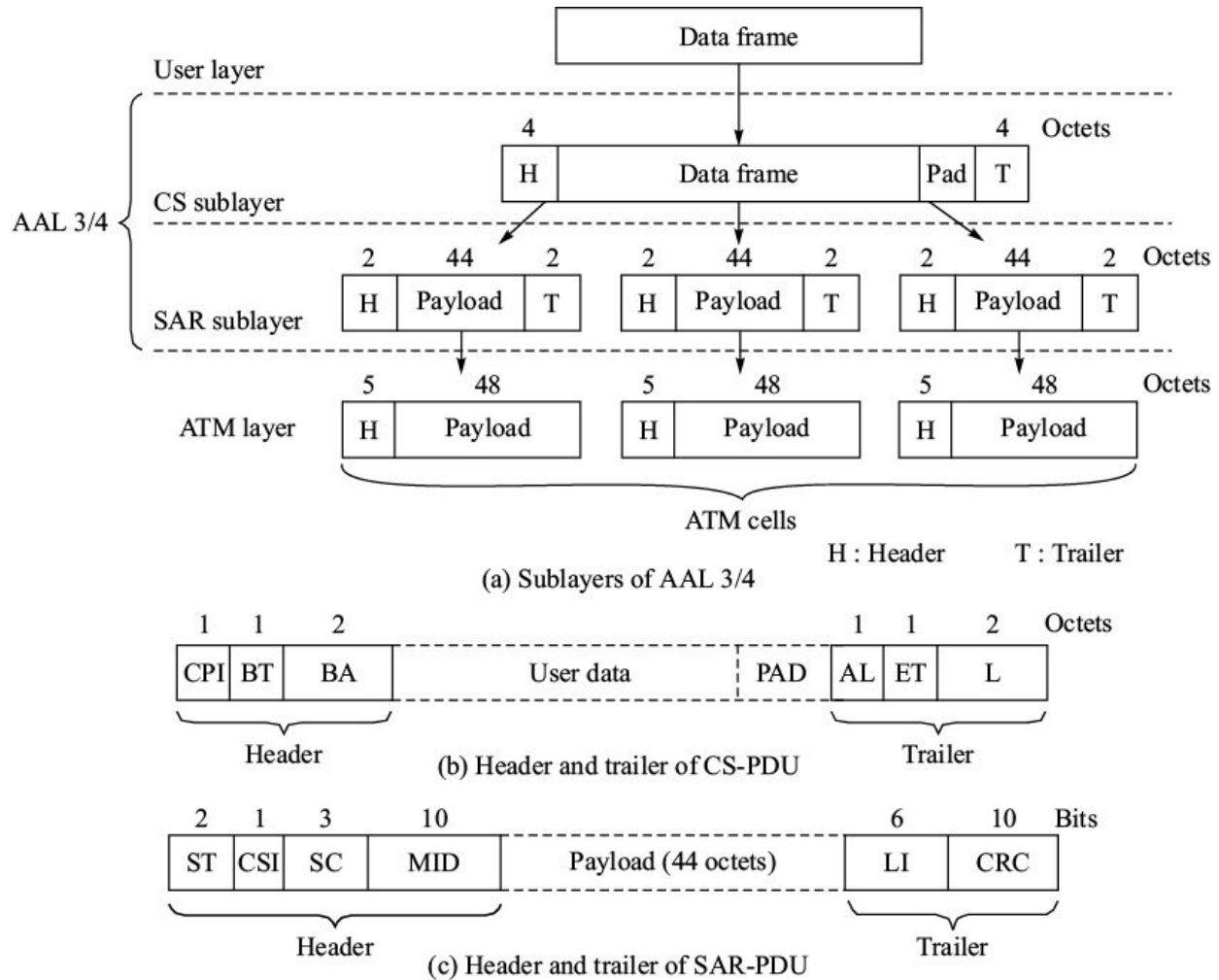


Figure 16.45 Structure of AAL 3/4.

Segment type (ST, 2 bits): The 2-bit ST identifier indicates whether the segment is the first, continuation or last segment of the message, or whether there is only one segment in the message.

- BOM (10): Beginning of Message
- COM (00): Continuation of Message
- EOM (01): End of Message
- SSM (11): Single Segment Message.

Convergence sublayer identifier (CSI, 1 bit): Use of this bit is yet to be defined.

Sequence count (SC, 3 bits). 3-bit SC field is a modulo-8 sequence number used for ordering the SAR-PDUs and locate the missing PDUs.

Multiplexing indication (MID, 10 bits): The 10-bit MID field identifies data segments belonging to different data flows but multiplexed on the same virtual channel (VCI/VPI). For example, a point-to-multipoint virtual connection can be set up and MID field is used to identify data flows of various users on the point-to-multipoint connection. There can be 2^{10} multipoint users.

Length indicator (LI, 6 bits): This field gives the number of useful octets in the payload of the SAR-PDU. This is useful only in EOM and SSM SAR-PDUs because other SAR-PDUs contain full payload of 44-octet useful data.

CRC (10 bits): This field is used for error detection. It contains CRC based on polynomial $1 + x + x^4 + x^5 + x^9 + x^{10}$. This is applied to the entire payload including header.

It may be noted that there is some degree of functional redundancy between SAR and convergence sublayers. For example, padding is done at both at convergence and SAR sublayers. ST, BT, and ET bits are used determining first and last segments.

16.14.6 ATM Adaptation Layer 5 (AAL 5) AAL 3/4 provides error detection, segmentation, and reassembly but at the cost of considerable overhead. Part of this inefficiency is due to strict layering principles. Four octets of header/trailer at the SAR sublayer can be removed if advantage of the services of ATM layer is taken.

- One bit of ATM layer header can be for marking the last segment of the message.
- It is also assumed that ATM layer provides sequenced delivery of message segments.

AAL 5 makes use of these services of ATM layer and provides the following basic functions:

- Error detection
- Segmentation and reassembly of messages
- Padding function to make the length of convert the entire PDU a multiple of 48 octets.

Figure 16.46 shows the convergence and SAR sublayers of AAL 5.

Convergence sublayer (CS). The convergence sublayer accepts the user data from the upper layer and constructs convergence sublayer PDU (CS-PDU) having length that is multiple of 48 octets. It adds an 8-octet trailer and sufficient padding bytes to make the length multiple of 48 octets. The number of padding bytes can be from 0 to 47.

The trailer consists of the following fields:

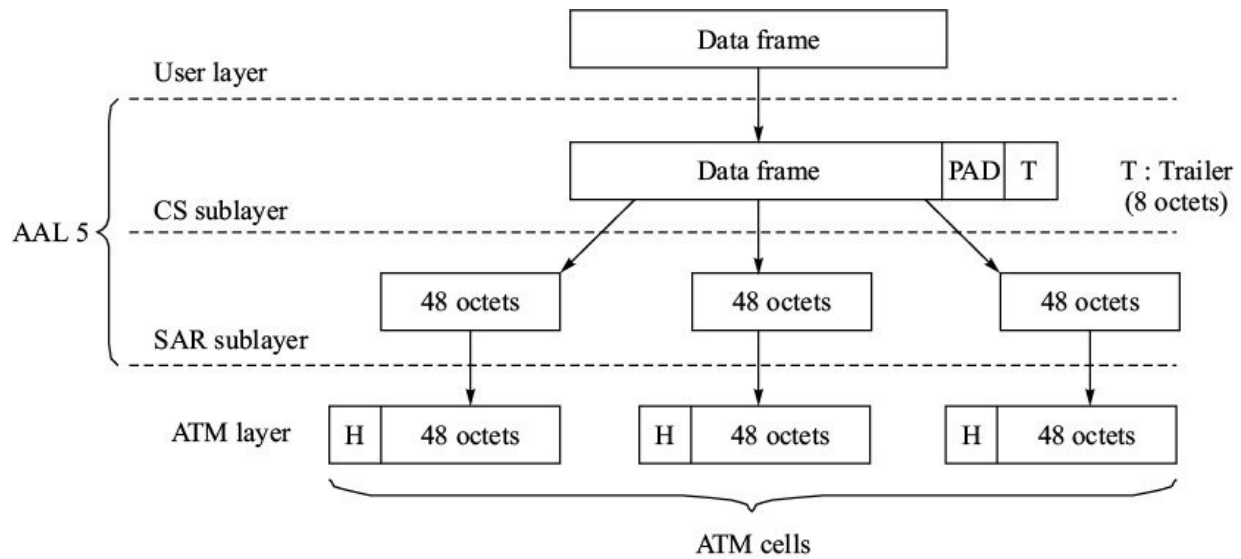
Use-to-user indication (UU, 1 octet): This byte is used by the higher layer to transfer one byte of information.

Common part indicator (CPI, 1 octet): This field is reserved for future use. It is set to 0.

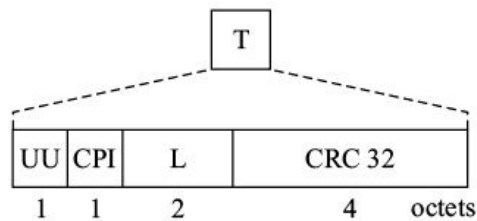
Length (L, 2 octets): This field indicates the length of message field (excluding the padding bytes). It enables separation of user data and the padding bytes.

CRC-32 (4 octets): This field is detection of errors in the entire CS-PDU. CRC-32 polynomial as described in Chapter 5 is used.

Segmentation and reassembly (SAR) sublayer: SAR sublayer simply segments the CS-PDU into 48-octet segments. Since the CS-PDU has length that is multiple of 48 octets, there is no need for padding bytes at SAR sublayer. No header or trailer is added at the SAR sublayer.



(a) Convergence and SAR sublayers in AAL 5.



(b) Convergence sublayers trailer

Figure 16.46 Convergence and SAR sublayers in AAL 5.

The segments of a CS-PDU are identified as belonging to one CS-PDU by the third bit of the PT field of the ATM header (Figure 16.41). This bit is 0 in all the ATM cells containing SAR-PDU except in the ATM cell that contains the last SAR-PDU. This bit is set to 1 in this ATM cell.

SUMMARY

In this chapter we examined three virtual circuit switching network technologies—X.25, frame relay, and ATM. X.25 interface defines the protocol for virtual circuit connection between a packet mode DTE and the DCE which is access node of the subnetwork. It defines protocols at the physical, data link, and packet levels. The packet level corresponds to the OSI network layer.

X.25 provides Switched Virtual Circuits (SVC) and Permanent Virtual Circuits (PVC). SVCs have call establishment, data transfer, and call clearing phases. PVCs, on the other hand, are always in data transfer phase. X.25 (level 3) uses the services of LAP-B at the data link layer. It implements sliding window flow control with local and remote acknowledgements at layer 3. If a

non-packet mode terminal is to be connected to the X.25 interface, an intermediary device PAD is required. ITU-T recommendations for the PAD are X.3, X.28, and X.29.

X.25 provides reliable and sequenced delivery of the data packets. It bundles a reasonably wide spectrum of subscribed services. Hop-by-hop error control and flow control functions in X.25, however, limit the data rates that it can support.

Frame relay is based on virtual circuit switching at layer 2 and provides connection-oriented SVC and PVC services. Compared to X.25, it provides higher throughput and lower end-to-end delay. Frame relay supports a congestion control mechanism which results in discarding of the frames that exceed defined customer specific limits of traffic parameters (CIR, B_c , B_e). Frame relay also informs the end systems when a frame encounters congestion in the network.

Asynchronous Transfer Mode (ATM) is the third virtual circuit switching technology that we discussed in this chapter. It uses fixed size, 53 octets, cells for transport of real time traffic (voice and video) and data traffic. The ATM nodes have two layers, physical and ATM layer. ATM layer switches the cells based logical connection identifier which consists of virtual path identifier (VPI) part and virtual channel identifier (VCI) part. It also performs the policing function which results in discard of cells when committed rate is exceeded. The end systems have additional layer called adaptation layer to support constant bit rate traffic and variable bit rate traffic.

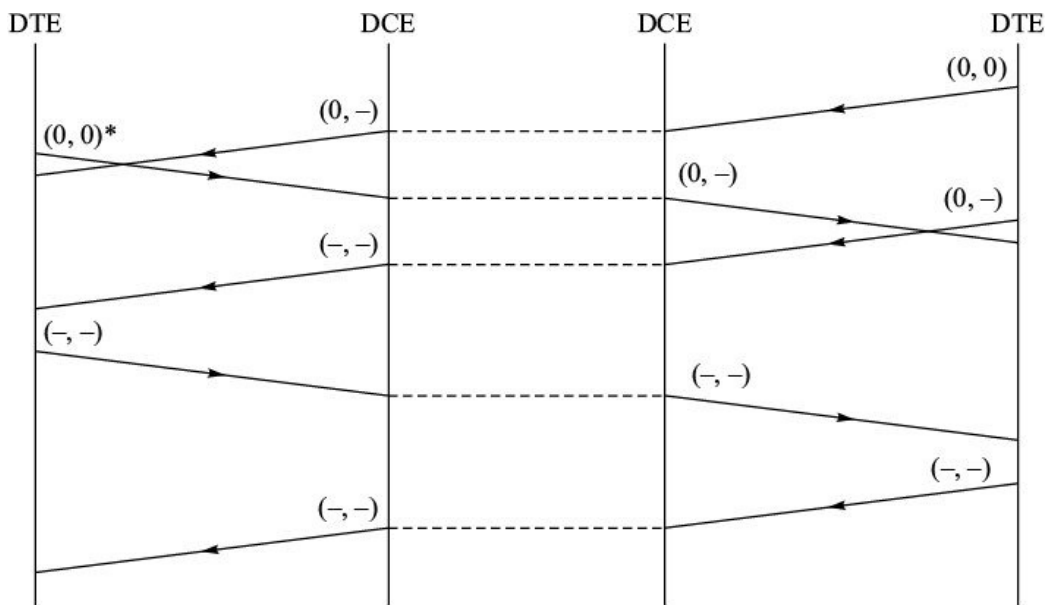
EXERCISES

1. Draw a DATA packet containing one user data octet 10010011. The other parameters of the packet are:

Logical channel group number	1010
Logical channel number	101111001
Acknowledgement option	Local
Sequence numbering scheme	Modulo 8
Sequence number of the packet	101
Sequence number of the acknowledgement	001
More data bit	0

2. The DATA packet of Exercise 1 is transmitted from DTE-P to the access node at 2400 bps. What is the transmission time? Assume single octet control field at level 2.

3. A DTE-P is operating three virtual connections simultaneously. It sends one DATA packet on each connection in sequence.
 - (a) Will the data link layer distinguish between the packets belonging to different connections?
 - (b) If the first frame is given sequence number 0, what will be sequence numbers of the frames containing DATA packets of the other connections?
4. Fill in the packet sequence numbers P(S), and acknowledgement numbers P(R) in the following exchange of DATA packets. Assume $D = 0$.



* Numbers within brackets indicate P(S), P(R)

Figure E16.47.

5. Do the Exercise 4 with $D = 1$.
6. Draw address fields of a CALL REQUEST packet if the calling address is 23456789234 and the called address is 3456678901234.
7. A DTE needs to send a 1500 octets of user data. The maximum size of the data field in the X.25 packet is restricted to 128 octets. The distant end DTE has negotiated the data field size of 256 octets with the local DCE. If $D = 0$, and the window size is 3, show the transfer of data and acknowledgement packets assuming that the acknowledgements are given after the receipt of every third packet. The first packet bears sequence number 0.
8. Flow control is used at level 2 and level 3 in X.25. Explain why flow control at both the levels is necessary.
9. Explain why error control is not needed at level 3 in X.25.

10. A user is connected to a frame relay network on an E1 (2048 kbps). The CIR is 1 Mbps with $B_C = 5$ M bits per 5 seconds and $B_e = 1$ M bit per 5 seconds.
- Can the user send data at 3 Mbps?
 - Can the user send data at 1 Mbps at all the times?
 - Can the user send data at 1.5 Mbps at all the times? Can frames be discarded at this data rate? If yes, when such situation can occur? Is it guaranteed that the frames will be discarded only if there is congestion in the network?
 - What is the maximum rate at which the user can send data if there is no congestion in the network?
11. A user has CIR of 64 kbps with $B_C = 128$ k bits in 2 seconds for frame relay service. He sends the following frames with their sizes indicated in brackets.

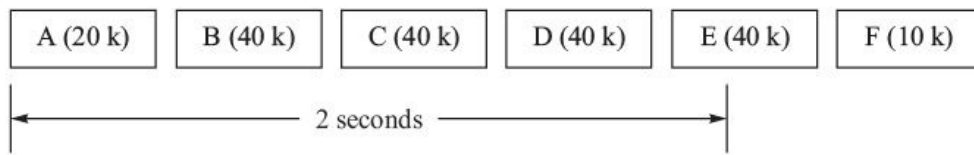


Figure E16.48.

- Which frames will have DE bit set to 1?
 - If EIR = 150 Mbps, which frame will be discarded?
12. Suppose that AAL 5 is being used and the receiver is in idle state. Then a block of user data is sent as a sequence of SAR-PDUs contained in ATM cells.
- Suppose that a one-bit error occurs in one of the SAR-PDU. What happens at the receiving end?
 - Suppose the SAR-PDU with third bit of PT field equal to 0 is lost. What happens at the receiving end?
 - Suppose the SAR-PDU with third bit of PT field equal to 1 is lost. What happens at the receiving end?
13. If the user data is 512 octets, how many ATM cells are required to transmit this payload assuming that (a) the adaptation layer is AAL 3/4?
- the adaptation layer is AAL 5?
- How many additional octets of headers and trailers are required for the payload of 512 octets in each case?

14. Explain why AAL 3/4 will not detect loss of sixteen consecutive cells of a single CS-PDU.

17

Internet Protocol (IP)

X.25, frame relay and ATM are the connection-oriented networking technologies based on virtual circuit switching approach. In this chapter, we study the most widely deployed connectionless-mode networking technology, the Internet Protocol (IP). It is based on datagram approach to packet switching. We also study some associated protocols.

We begin this chapter with examination of Internet Protocol, version 4 (IPv4). We study its packet format, operation, and addressing structure in detail. Classful subnetting and VLSM are explained with numerous examples. Then we move over to the protocols ARP, RARP, ICMP, and PPP, which are associated with IP. We also look at the main features of two more connectionless-mode packet switching protocols—the next generation IP protocol IPv6 and ISO 8473. IPv6 is yet to be deployed and ISO 8473 which does not enjoy as much acceptability as IP by the industry. We close the chapter with an introduction to concept of differentiated service as applied to IP networks. The routing protocols are left for the next chapter.

17.1 CONNECTIONLESS-MODE SWITCHED DATA NETWORKS

ITU-T X.25, frame relay, and ATM are connection-oriented packet switching protocols that establish virtual connections between communicating entities. Packet switched data networks based on datagram approach do not establish such connections and are therefore called connectionless-mode switched data networks. They transport each datagram from a source to the destination as an independent entity. Each datagram, therefore, carries the source address and the destination address, and is routed to the destination by the network nodes.

The network nodes that switch datagrams are called routers. The routers have forwarding tables that enable them to decide the next hop of each datagram towards its destination. There have been two protocols for packet switched networks based on datagram approach:

- The current and most popular protocol is Internet Protocol (IP), version 4 as documented in RFC 791. It works with Transmission Control Protocol (TCP) which roughly corresponds to the transport, session, and presentation layers of the OSI reference model. TCP and IP protocols together are popularly known as TCP/IP protocol suite.
- The other protocol is ISO 8473, “Protocol for providing connectionless-mode network service”.

We will discuss both these protocols in this chapter. IP is discussed in detail because it is widely deployed.

17.2 INTERNET PROTOCOL (IP)

The Internet Protocol (IP) is a network layer protocol that provides best effort and connectionless delivery of datagrams. The delivery is best effort in the sense that it is non-guaranteed and there is no acknowledgement of delivery having been made to the destination. The datagrams may be lost, duplicated, delayed or delivered out of sequence.

Figure 17.1 shows placement of the Internet Protocol in the layered architecture. IP is a layer-3 protocol and supports variety of layer 2 protocols (IEEE 802.2, Ethernet-DIX, PPP, HDLC, ATM, frame relay). It provides service to layer 4 protocols which include TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

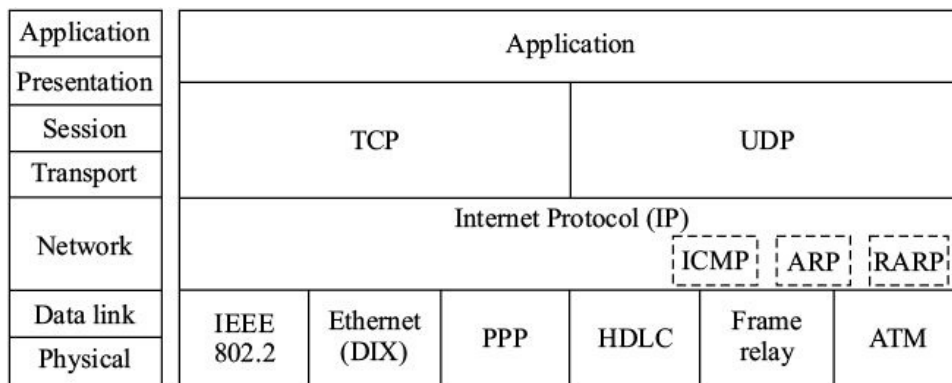


Figure 17.1 Layered architecture of TCP/IP suite.

Like any other protocol, IP specifies the format of datagrams (or simply IP packets) and the procedures for transport of IP packets at layer 3. The protocol is implemented in end systems and in packet switching nodes (called routers).

The IP packets are encapsulated in a layer 2 frame before being sent over the physical transmission medium (Figure 17.2). Since layer 2 may impose limitation on the size of IP packets, the Internet Protocol provides fragmentation and reassembly of datagrams. Fragmentation can be carried out at any stage (by the source end system or by any router). Reassembly of the fragments is carried out only by the destination end system.

There are two versions of Internet Protocol:

- IP version 4 (IPv4)
- IP version 6 (IPv6).

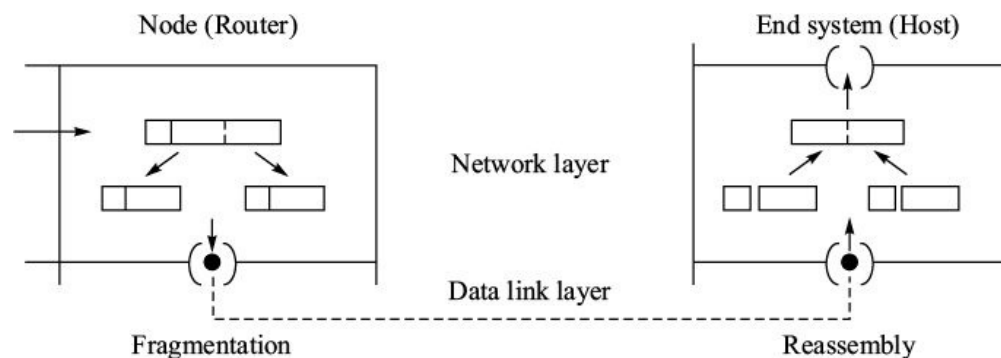


Figure 17.2 Fragmentation and reassembly.

Current deployed version is IPv4. Therefore, we will study IPv4 in detail and cover only the salient features of IPv6 in this chapter.

17.2.1 Associated Protocols

Along with Internet Protocol, there are three other important layer-3 protocols that support its functionality:

- Address Resolution Protocol (ARP)
- Reverse Address Resolution Protocol (RARP)
- Internet Control Message Protocol (ICMP).

ARP is used for determining layer-2 address for a given layer-3 address. RARP is used for determining layer-3 address for a given layer-2 address. ICMP is used for reporting errors and other messages. We discuss these protocols after

covering the Internet Protocol.

It is to be noted that the routing protocols (RIP, OSPF, BGP) are a different category of protocols that enable creation and maintenance of forwarding tables in the routers. The routers make use of these forwarding tables to determine the next hop of IP packets across the packet switching network.

17.3 IPV4 PACKET FORMAT

Figure 17.3 shows the format of IPv4 packet. It consists of a header and a data field. The data field contains user data (the TCP/UDP packet) and is variable in length. The header consists of several fields as described below.

Version (4 bits). It indicates the version of Internet Protocol being used. The current version is 4 (IPv4).

IP header length (4 bits). It indicates the length of IP header in terms of number of 32-bit words. The minimum header length is 5 (20 octets) and the maximum length is 15 (60 octets).

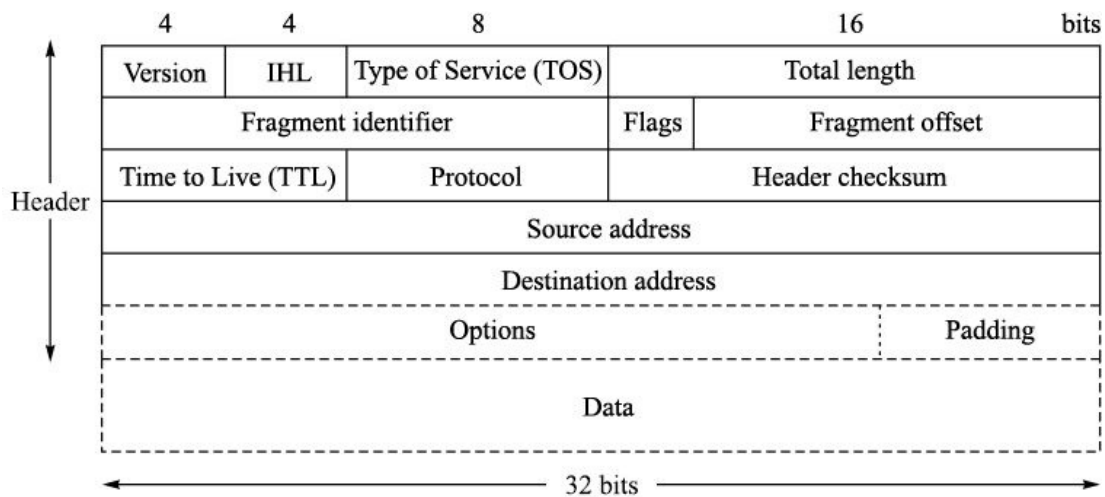


Figure 17.3 IPv4 packet format.

Type of service (8 bits). It specifies priority, throughput, and reliability parameters (RFC 1349). It consists of five subfields as shown in Figure 17.4.

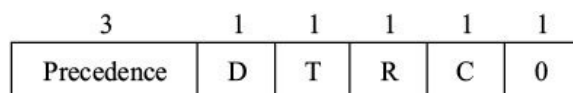


Figure 17.4 Type of service field.

Precedence value assigns the priority level. For example, 000 is no priority. D, T, R, and C bits are used as under:

D	T	R	C	
0	0	0	0	Normal service
1	0	0	0	Minimum delay
0	1	0	0	Maximum throughput
0	0	1	0	Maximum reliability
0	0	0	1	Minimum cost

This field was later changed in RFC 2474 and the first six bits were renamed as Differentiated Service Code Point (DSCP) and the last two bits were reserved for future use (Figure 17.5). Using DSCP, an IP packet is given a defined class of handling (limited delay, guaranteed throughput) within a Quality of Service (QoS) domain. We will discuss this field later when we introduce the concept of differentiated services.

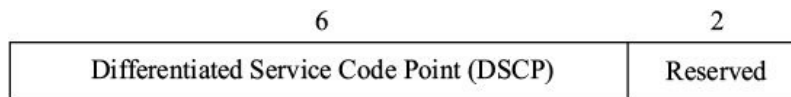


Figure 17.5 Type of service field (RFC 2474).

Total length (16 bits). It indicates the total length of this IP packet, including header and data fields. It is specified as number of octets. Maximum size of the packet is 65,535 octets. Every router and end system must be capable of handling at least 576-octet packet.

Fragment identifier (16 bits). It contains an integer that identifies the current IP packet. It is used to put together the IP packet if its data field is fragmented and put into smaller IP packets.

Flags (3 bits). The first bit is not used. The middle bit of this field is Don't Fragment (DF) flag. If it is 1, the IP packet cannot be fragmented. If MTU (maximum transmission unit) size of the data link is less than the packet size, and DF bit is 1, the packet is discarded.

The last bit is More Fragment (MF) bit that indicates if there are more fragments of a packet. If it is 1, there are more fragments of the IP packet. This bit is 0 in the last fragment.

Fragment offset (13 bits). It indicates the position of the data fragment relative to the beginning of unfragmented data. Offset is measured in multiple of eight octets.

Time to live (TTL, 8 bits). It maintains a counter that gradually decrements

down to zero, at which point, the IP packet is discarded. This keeps the packet from looping endlessly. The counter can be set in the range of 0–255 seconds. The recommended value is 64 seconds. Every router decrements the TTL by an amount equal to its processing/waiting time. If the processing/waiting time is less than one second, TTL is decremented by 1.

Protocol (8 bits). This field indicates the protocol which is to receive the contents of the data field of the IP packets. The examples of the protocols are ICMP (1), TCP(6), UDP (17), EGP(8), OSPF(89).

Header checksum (16 bits). It is 16-bit 1's complement of the checksum of all 16-bit words in the header. It is used for detection of errors in the header. Since a router changes contents of TTL and fragment offset fields, header checksum field is recomputed by every router when it forwards a packet.

Source and destination addresses (32 bits each). These fields contain IP addresses of the source and of the destination.

Options (Variable size). This field contains the options chosen by the source. Options field includes provisions for time stamps, recording routes, *etc.* If option is chosen for time stamps, every router puts time stamp when it forwards the packet. It will put its IP address if the route of an IP packet is to be traced. There are also other option such as security, source route, *etc.*

Padding (Variable size). Padding field ensures that the header ends on 32-bit boundary. This field is filled with 0s.

17.3.1 Fragmentation

An IP packet may be fragmented into multiple smaller IP packets if a data link towards the destination has packet size limitations. Each fragment contains the IP header and a part of data field of the original IP packet (Figure 17.6).

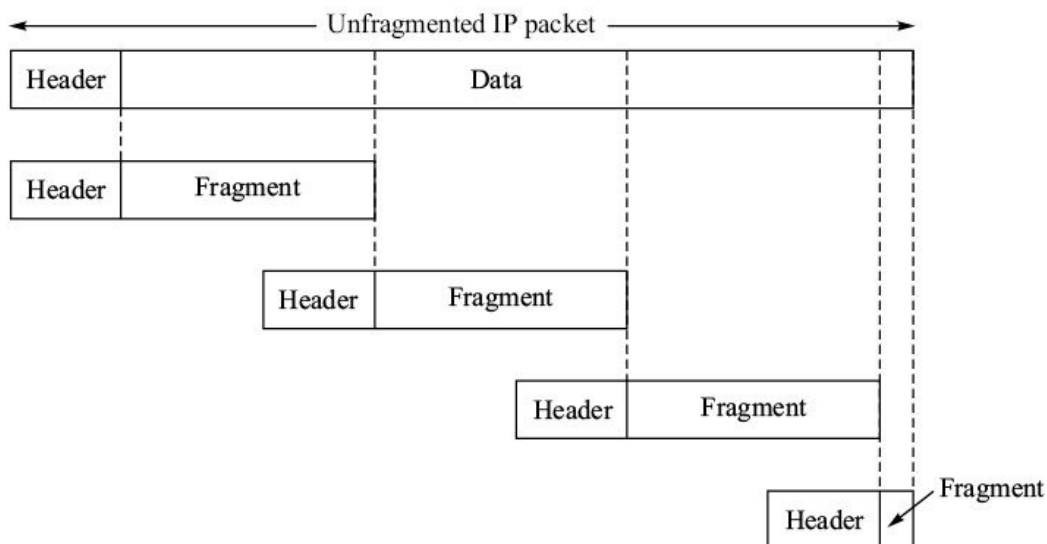


Figure 17.6 Fragmentation of IP packet.

Fragments of an IP packet are identified as belonging to the same IP packet by the fragment identification field. Location of a fragment within the original data field is indicated by the fragment offset field (Figure 17.7).

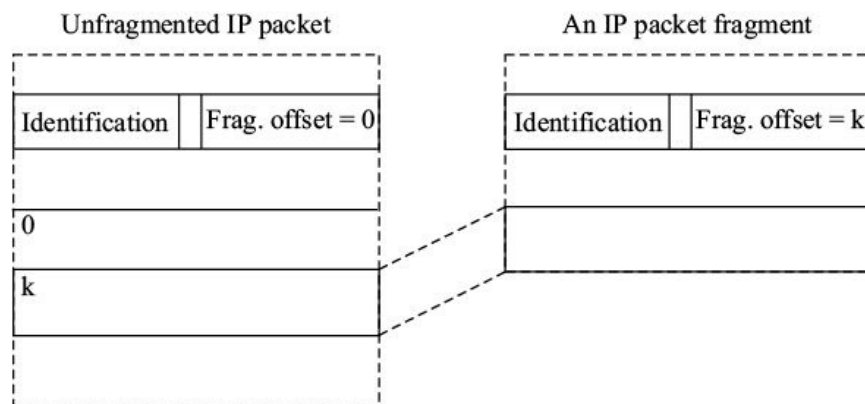


Figure 17.7 Fragment offset.

Figure 17.8 shows an example of fragmentation. Here, header size is 20 octets and unfragmented IP packet has size of 1040 octets. Fragmentation is carried in two different routers. The original IP packet is first divided into two fragments by a router, and each fragment is further divided into two fragments in a subsequent router. Only relevant fields of the IP packets are indicated in the figure. Note that:

- The unfragmented IP packet and the first fragment have an offset of zero.
- All fragments except the last fragment have sizes in multiple of eight octets. This excludes the header, whose size is in multiple of four.

- Offset is measured in multiple of eight octets. Thus offset of 64 is equivalent to 512 octets.

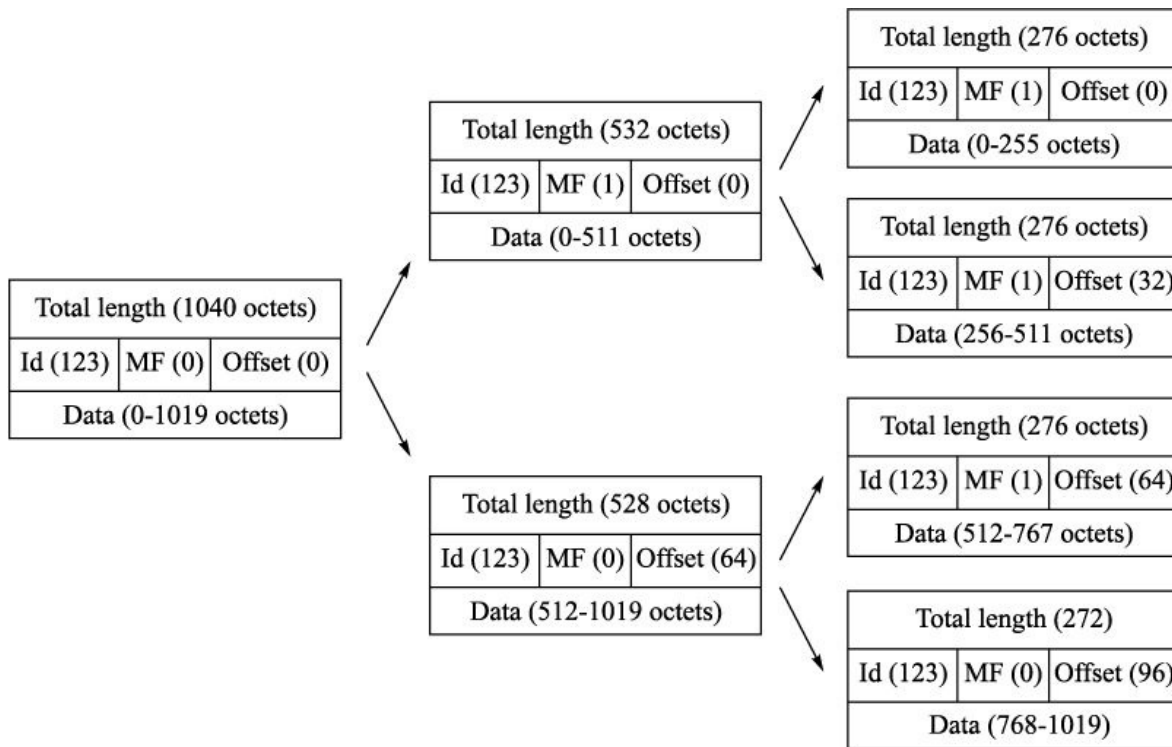


Figure 17.8 Example of fragmentation.

Reassembly. Reassembly is carried out only at the destination since the fragmented IP packets may take different routes to reach the destination. Fragments with same combination of the source address, the destination address, and the identification field are reassembled.

It is possible that some fragments do not arrive in time because IP service is best effort. When the first fragmented IP packet with MF bit equal to 1 and offset field equal to 0 is received at the destination, its remaining TTL time is transferred to the reassembly timer. If the timer expires before the IP packet is reassembled, all the received IP fragments are discarded.

17.3.2 Options

The options field in an IP header includes provisions for testing, debugging, and for specifying security measures. If an IP packet is fragmented, some or all options are copied on all the fragments. Important options are described below. We will not go into the specific formats of each option. In general, option field consists of option code (one octet), option length (one octet), and data pertaining

to that option. The first bit of the option code indicates whether the option is to be copied onto all the fragments. If it is 1, the option is copied onto the fragments. Option field includes the following four provisions:

- Record route
- Source route (loose and strict)
- Time stamp
- Security.

Record route. The record route option enables the source end system to create an empty list of addresses in the header. The router that handles the packet records its IP address in the list. When the packet arrives, the destination end system can extract the route taken by the IP packet.

Source route. Source route option enables the sender to dictate a path through the IP network. Source routing can be of two types, strict and loose. In strict source route option, a sequence of IP addresses of the routers is specified. The IP packet strictly follows the specified path. If it cannot, it is discarded. The loose source route option specifies a list of address that the packet must follow. But it allows the packet to traverse additional intermediate routers not included in the list.

Time stamp. The time stamp option works like the record route option. Every router that process the IP packets records time stamp in milliseconds since midnight in addition to its IP address in the empty list.

Security. It allows a security label to be attached to an IP packet.

17.4 HIERARCHICAL ADDRESSING

Addressing scheme can be flat or hierarchical. In flat addressing scheme, the whole address of an end system is used for routing the data units. The local area networks use flat addressing scheme. The bridges maintain forwarding tables that contain whole addresses of all the stations. Because of limited number of end stations, it is possible to have forwarding tables with flat addressing scheme.

For nationwide or global networks, flat addressing scheme is not scalable. Therefore, hierarchical addressing scheme is used. In hierarchical addressing, part of the address indicates a partition of a network. Typical example of

hierarchical addressing is the telephone network. A telephone number, *e.g.* 912237310106 consists of country code (91), area code (22), exchange code (3731), and the customer number (0106).

Exchange service area is one partition of the network. Several exchange service areas constitute a bigger domain that is identified by an area code and several such domains make the telephone network of a country, which is identified by a country code.

Hierarchical addressing reduces the size of forwarding tables by aggregation of routes. Figure 17.9 shows a network having 12 nodes. In Figure 17.9a, the network is not partitioned. With flat addressing, each node will need a forwarding table having eleven route records. If the network is partitioned into three networks as shown in Figure 17.9b, each node will have a routing

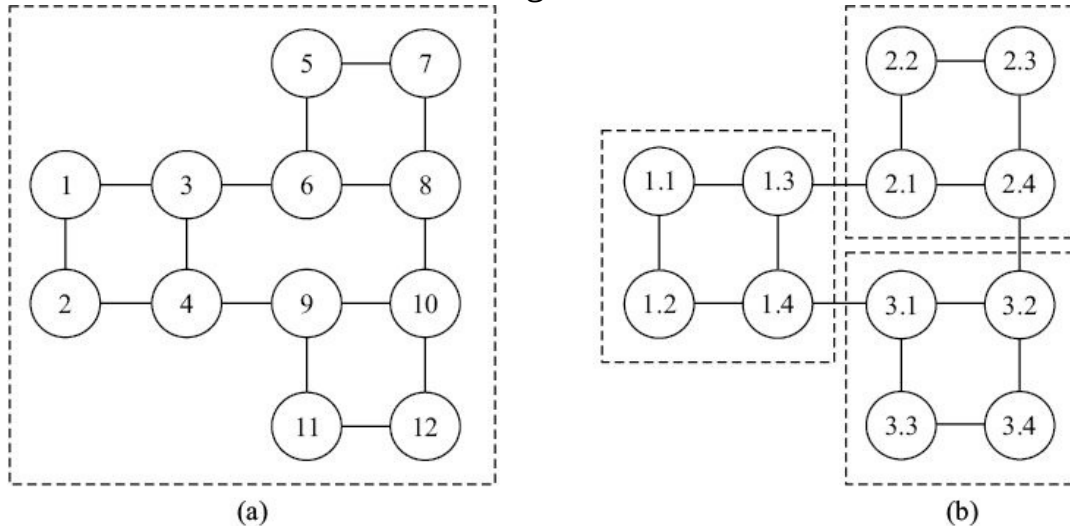


Figure 17.9 Partitioning of a network.

table containing only five records. For example, forwarding table of node 1.1 will contain routes for 1.2, 1.3, 1.4, network 2.x, and network 3.x, *i.e.* for each partition of the network, only one route is required. Note that each node address has a common network part, which identifies the partition of the network to which the node belongs.

17.4.1 Addressing Scheme of IPv4

IPv4 specifies 32-bit IP address for the end systems (called hosts in Internet terminology) and the router interfaces. Each host and router interface has a unique IP address. It uses two-level hierarchical addressing scheme. The address consists of two parts, network number and host number (Figure 17.10). Network number identifies the network on which the host, identified by host number,

resides. Hosts connected onto the same network will have same network number.

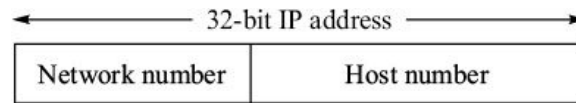


Figure 17.10 32-bit IP address.

Network number is unique and is assigned by Internet Network Information Centre (InterNIC) if the network is to be part of the public Internet. Host number assignments are made by the network manager locally.

For ease of reading and writing, 32-bit address is expressed using ‘dotted-decimal notation’. 32-bit address is divided into four octets separated by a dot and each octet is written as its decimal equivalent. For example, 32-bit IP address 1000000011110000000000001101101 is written as 10000000 . 11110000 . 00000001 . 01101101 = 128.240.1.109

17.4.2 Classful IP Addressing

The IP address space consists of 2^{32} addresses. In order to support networks of various sizes, it is divided into several classes. The classification and associated address structure adopted in the public Internet is shown in Figure 17.11. It must be borne in mind that this classification is applicable to the Internet only. A private IP network can have its own address structure.

- Class A is identified by the leading bit of the address, which is always 0. Class A addresses have 8 bits of network number and 24 bits of host number.
- Class B addresses have 16 bits of network number and 16 bits of host number. Class B is identified by the first two bits of network number, which are 10.
- Class C addresses have 24 bits of network number and 8 bits of host number. Class C is identified by the first three bits of network number, which are 110.
- Class D addresses are used for multicasting.¹ Class D is identified by 1110 as the leading four bits of the addresses. There is no host number. The rest 28 bits identify the multicast group.
- Class E addresses are reserved for experimental use and are identified by 1111 as the first four bits of the address.

In classful IP addressing, the dividing point between network part and host part is fixed depending on the class of an address. For example, a class B address as identified by leading two bits (10), always has network part 16 bits long. For more explicit description of an address, size of network part of the address is appended with a slash after the address. For example, 180.15.0.1/16 represents a class B address having 16-bit network part.

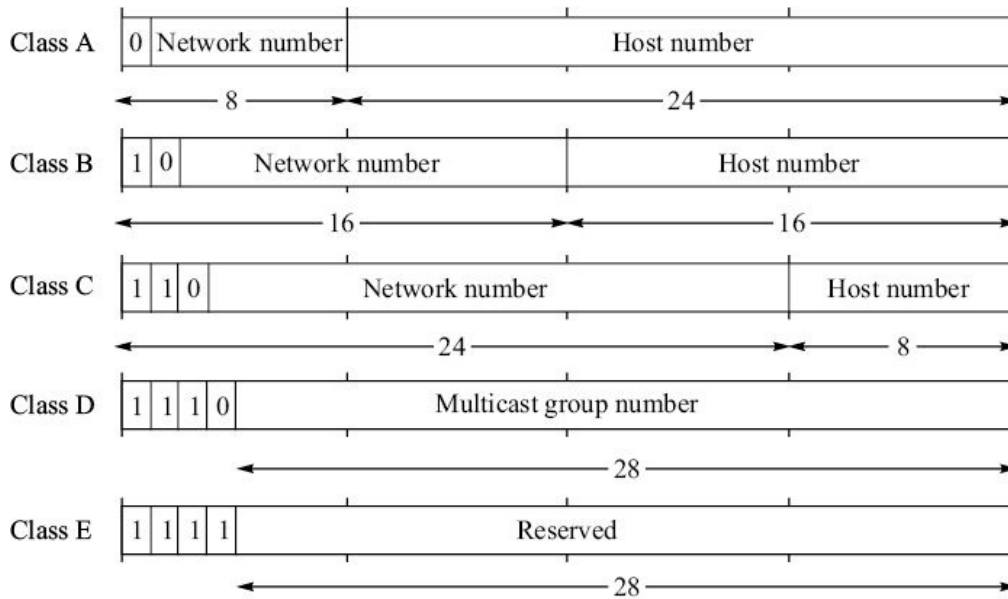


Figure 17.11 Classification of IP address space.

When an address is expressed in dotted decimal notation, its class can be determined readily from the decimal equivalent of first octet. Table 17.1 gives the range of values of the first octet for each class. Network numbers having all 0s and all 1s, apart from the bits that identify the class, have special meanings described later. We have not considered these network numbers while specifying the range of values of the first octet. Note that for classes B and C, the network part extends to second and third octets. Therefore, apart from the class identifying bits, the rest of first octet bits can be all 0s or all 1s.

TABLE 17.1 First Octet Rule for Class Determination			
Class	Class identifier	Range of first octet	Decimal equivalent
A	0	00000001–01111110	1–126
B	10	10000000–10111111	128–191
C	110	11000000–11011111	192–223
D	1110	11100000–11101111	224–239

EXAMPLE 17.1 Determine the range of decimal values of first octet for class B addresses.

Solution The network part of class B address can take the range values from 10000000 . 00000000 to 10111111 . 11111111. Since the underlined part cannot have all 0s and all 1s, the range is limited to 10000000 . 00000001 to 10111111 . 11111110. The first octet can take decimal values from 128 (binary 1000000) to 191 (binary 1011111).

Figure 17.12 shows an example of addresses used in an IP internetwork. Let us examine Router 2 in detail. Router 2 is connected to three networks—10.0.0.0/8, 200.168.18.0/24, and 200.168.19.0/24.

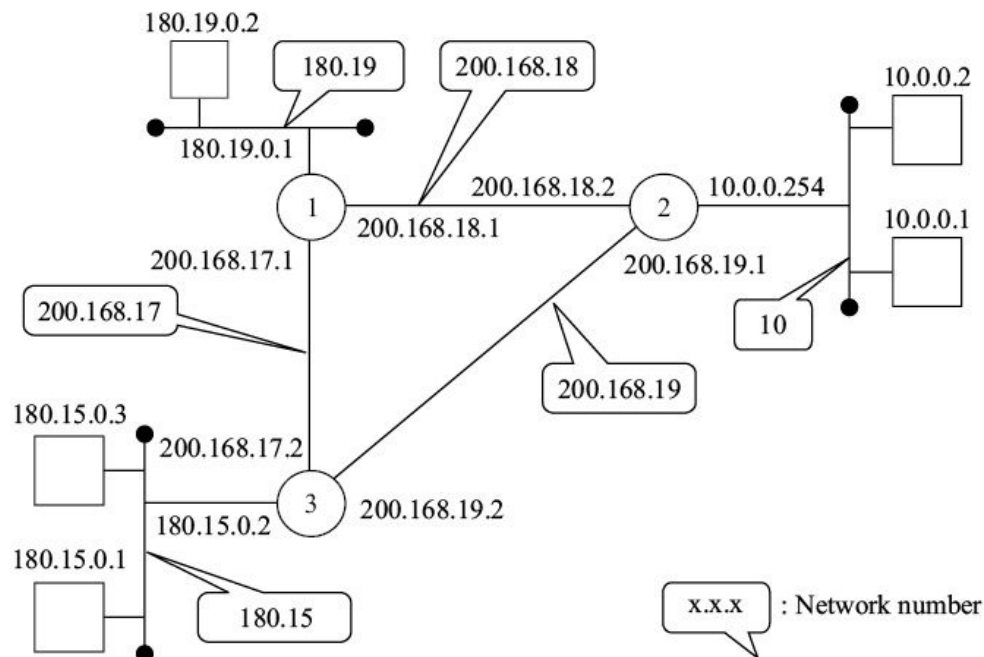


Figure 17.12 Example of IP addressing.

1. There are three IP addresses on 10.0.0.0/8 network—10.0.0.254, 10.0.0.1, and 10.0.0.2. Note that 10.0.0.254 is the IP address of the interface of the node 2 that connects to that local area network.
2. 200.168.18.0/24 is a point-to-point network connecting node 1 and node 2. The respective interfaces of these nodes have IP addresses 200.168.18.1 and 200.168.18.2.
3. 200.168.19.0/24 is a point-to-point network connecting node 2 and node 3. The respective interfaces of these nodes have IP addresses 200.168.19.1 and 200.168.19.2.

17.4.3 Number of Networks and Hosts

Each class can have a maximum number of network addresses, and each

network of a class can support a maximum number of host addresses. We can calculate these numbers based on the range of values the network part and host part can take. Network part without the class identifier bits, cannot be all 0s or all 1s as these addresses have special meaning. The same holds for host number also.

As we saw in Example 17.1, network part of class B address can range from 10000000. 00000001 (128.1) to 10111111 . 11111110 (191.254). The total network address space has 16,382 ($= 2^{14} - 2$) network addresses. The host part has 16 bits, and therefore, each network address can support 65,534 ($= 2^{16} - 2$) hosts. We can calculate these numbers for other classes also. Table 17.2 summarizes the results of these calculations.

Class	Class identifier	Network part (bits)	Size of host part (bits)	Number of networks	Number of hosts
A	0	8	24	126	16,777,214
B	10	16	16	16,382	65,534
C	110	24	8	2,097,150	254

17.4.4 Special Addresses

Some combination of bits have been reserved for special functions, such as broadcast, loopback testing, and other functions. Table 17.3 summarizes such special addresses.

IP address [net-id, host-id]	Implication
[net-id, all 1s]	Directed broadcast to the network net-id
[all 1s, all 1s]	Local broadcast within the network
[all 0s, all 0s]	'This network', 'This host'
[all 0s, host-id]	A host on this network
[127. any]	Loopback address

Directed broadcast. As we mentioned earlier, all 0s and all 1s combinations in network and host part are used for special purposes. A field consisting of 1s is interpreted as 'all'. Thus, if a host wants to send a directed broadcast to a network having address net-id, it puts IP destination address on the packet as [net-id, all 1s]. The packet is routed to the network net-id and is received by all the hosts on the network.

Local broadcast. A host may want to broadcast an IP packet to the other local hosts (hosts connected on the same network). This is called local broadcast.

Local broadcast is done by sending the IP packet with destination IP address as all 1s.

All 0s field. A field consisting of all 0s is interpreted as ‘this’. Thus a host host-id not knowing its own network address, can send an IP packet with source address as [all 0s, host-id]. When a packet is received with source address having net-id = all 0s, the receiver immediately realizes that the packet has been sent by a host on the same network.

Loopback address. [127. any] is used as loopback address for diagnostic purposes.

EXAMPLE 17.2 In Figure 17.12, host 10.0.0.1 sends a packet with destination address as (a) 180.15.0.3

(b) 180.15.255.255

(c) 255.255.255.255.

Which are the hosts that receive this packet?

Solution

(a) The packet is received by the host having IP address 180.15.0.3.

(b) This is a directed broadcast to network 180.15.0.0. Therefore, the packet is received by the hosts 180.15.0.1 and 180.15.0.3.

(c) This is a local broadcast. Therefore, the packet is received by the host 10.0.0.2.

17.5 SUBNETTING

The main advantage of dividing IP address into two parts is optimization of the size of forwarding tables. Instead of keeping one route-entry per destination host, a router is required to keep one route-entry per network and examine only the network part of the IP address for making routing decision. But this scheme has one major limitation. The number of network addresses that can be allotted is inadequate. The total network address space is only about 2 million network addresses (Table 17.2), which is not sufficient. Every local area network that needs to be connected to the Internet, requires unique network address from the network address space.

This problem is caused by the division of 32-bit address into network part and

host part at octet boundaries. Note that class A and class B addresses are about 3.2 billion ($= 2^{31} + 2^{30}$) and constitute 75% of the total address space of 4.29 billion ($= 2^{32}$). But the number of network addresses these classes generate is merely 16,508 which is just 0.78% of 2.1 million total network addresses (Table 17.2).

There can be several ways to work around this situation. But we need to keep in mind that the Internet will require a major overhaul if a totally new addressing concept is introduced. Therefore, the solution needs to be within the original framework of the addressing scheme. One possible way is to allow sharing of a network number by multiple networks. The networks that share a common network number are called subnetworks (or subnets, in short) and the process is called subnetting. The original network number is called major network number and is referred to as major net.

17.5.1 Classical Subnetting

Consider an organization has single class B IP network address 130.10.0.0, but it has four networks to connect to the Internet. Figure 17.13 shows how a common network address is shared among four subnets.

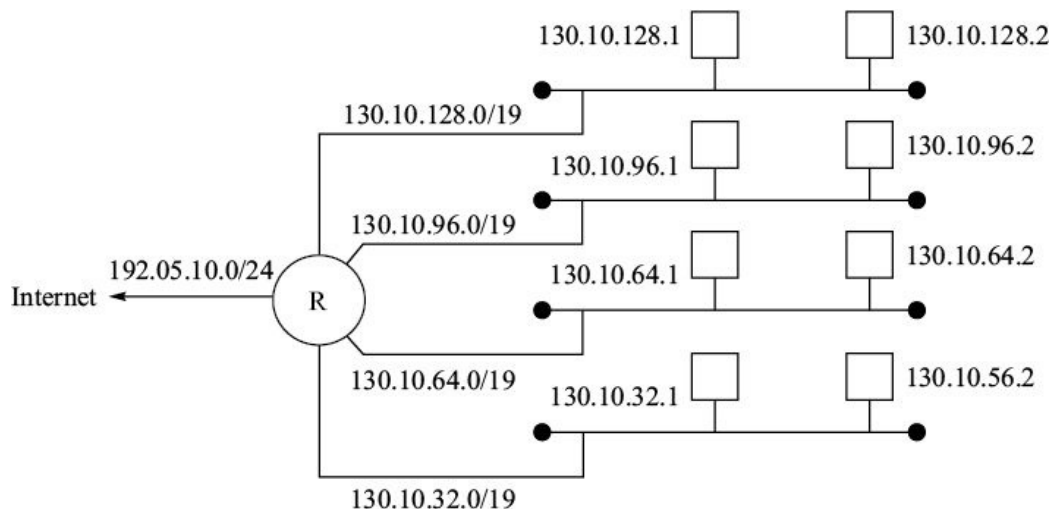


Figure 17.13 Classical subnetting.

- Class B address has 16 bits in the host part. Three leading bits of the host part are used for identifying the subnet and the balance 12 bits are used for individual end systems on each subnet. Thus, an IP address has three parts now (Figure 17.14). The major network number with subnet-id is called subnet number or subnet address.

- With 3-bit subnet-id, we have provision for at the most 8 ($=2^3$) subnets. All 0s and all 1s subnet-ids are not used as they have special meanings. Thus we have six subnet-ids. (Table 17.4). The organization uses four subnets out of these six for its current requirements.

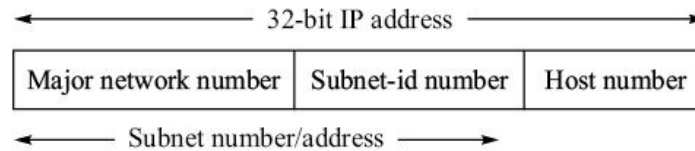


Figure 17.14 Subnet number.

TABLE 17.4 Available Subnets on 130.10.32.0/19		
Subnet	Binary representation	Available host addresses
130.10.32.0/19	10000010 . 00001010 . 00100000 . 00000000	130.10.32.1 to 130.10.63.254
130.10.64.0/19	10000010 . 00001010 . 01000000 . 00000000	130.10.64.1 to 130.10.95.254
130.10.96.0/19	10000010 . 00001010 . 01100000 . 00000000	130.10.96.1 to 130.10.127.254
130.10.128.0/19	10000010 . 00001010 . 10000000 . 00000000	130.10.128.1 to 130.10.159.254
130.10.160.0/19	10000010 . 00001010 . 10100000 . 00000000	130.10.160.1 to 130.191.63.254
130.10.192.0/19	10000010 . 00001010 . 11000000 . 00000000	130.10.192.1 to 130.10.223.254

- Subnet address is 19 bits long. When a packet arrives from the Internet for a particular host, the router must be able to demarcate the subnet address from the 32-bit IP address to determine the subnet to which the packet is to be sent. This is done using a subnet mask. The subnet mask is 32 bits long and has binary 1 in those bit positions that pertain to subnet address. A router retrieves the subnet address by performing AND operation on the IP address and subnet mask. For example,

IP address : 130.10.32.1 10000010 . 00001010 . 00100000 . 00000001
Subnet mask : 255.255.224.0 11111111 . 11111111 . 11100000 . 00000000
Subnet address : 130.10.32.0 10000010 . 00001010 . 00100000 . 00000000

Let us take a less obvious example:

IP address : 130.10.56.1 10000010 . 00001010 . 00111000 . 00000001
Subnet mask : 255.255.224.0 11111111 . 11111111 . 11100000 . 00000000
Subnet address : 130.10.32.0 10000010 . 00001010 . 00100000 . 00000000

Thus, the subnet mask must be available for these router interfaces. It can be preconfigured.

The subnetting described above is referred to as ‘classical’ subnetting. In classical subnetting, only one subnet mask is allowed for each major net. This results in wastage of address space as we shall see later.

EXAMPLE 17.3 An organization has been assigned the network address 194.1.1.0/24. It needs eight subnets. The maximum number of hosts a subnet can be required to support is 10. Define a subnetwork addressing plan.

Solution

- (a) For eight subnets, four bits long subnet-id is required. Four bits will give 16 ($= 2^4$) subnets, which will meet the current requirements and have provision for the future.
- (b) Being a class C address, these four bits will be borrowed from the last octet. That leaves four bits for the host number. With four bits, 14 ($= 2^4 - 2$) host numbers can be defined. Maximum number of hosts on a subnet is 10. Therefore, the requirement is met.
- (c) Any eight subnet addresses can be chosen out of 16 available addresses:

```

194.1.1.0/28    : 11100010 . 00000001 . 00000001 . 00000000
194.1.1.16/28   : 11100010 . 00000001 . 00000001 . 00010000
194.1.1.32/28   : 11100010 . 00000001 . 00000001 . 00100000
194.1.1.48/28   : 11100010 . 00000001 . 00000001 . 00110000
194.1.1.64/28   : 11100010 . 00000001 . 00000001 . 01000000
194.1.1.80/28   : 11100010 . 00000001 . 00000001 . 01010000
194.1.1.96/28   : 11100010 . 00000001 . 00000001 . 01100000
194.1.1.112/28  : 11100010 . 00000001 . 00000001 . 01110000
194.1.1.128/28  : 11100010 . 00000001 . 00000001 . 10000000
194.1.1.144/28  : 11100010 . 00000001 . 00000001 . 10010000
194.1.1.160/28  : 11100010 . 00000001 . 00000001 . 10100000
194.1.1.176/28  : 11100010 . 00000001 . 00000001 . 10110000
194.1.1.192/28  : 11100010 . 00000001 . 00000001 . 11000000
194.1.1.208/28  : 11100010 . 00000001 . 00000001 . 11010000
194.1.1.224/28  : 11100010 . 00000001 . 00000001 . 11100000
194.1.1.240/28  : 11100010 . 00000001 . 00000001 . 11110000

```

(d) The subnet mask will be 255.255.255.240.

17.5.2 Route Advertisement

In Figure 17.13, the subnetting is not visible to rest of the Internet outside the router R. When router R sends a routing update, it advertises 130.10.0.0 as its network address. Even if it were to advertise the subnet addresses, the other routers will use class B mask of /16 and ignore the subnet part of the addresses. So the forwarding tables in other routers of the Internet continue to have only one entry for the class B address of the organization. Thus, the router R is gateway of all the subnets of 130.10.0.0. It can be put in another way that all the subnets of this network 130.10.0.0 must be contiguous.

Assuming contiguous subnets is a simplistic view of the real situation. For example, suppose one (130.10.32.0/19) of the four subnets shown in Figure 17.13 is located in a different building or city. Figure 17.15 shows one way of setting up the network. R1 and R2 advertise their subnet addresses to R3 for updating its forwarding table. Router R3 takes into account the classful boundary (first 16 bits for class B) for populating its forwarding table. Thus, as far as

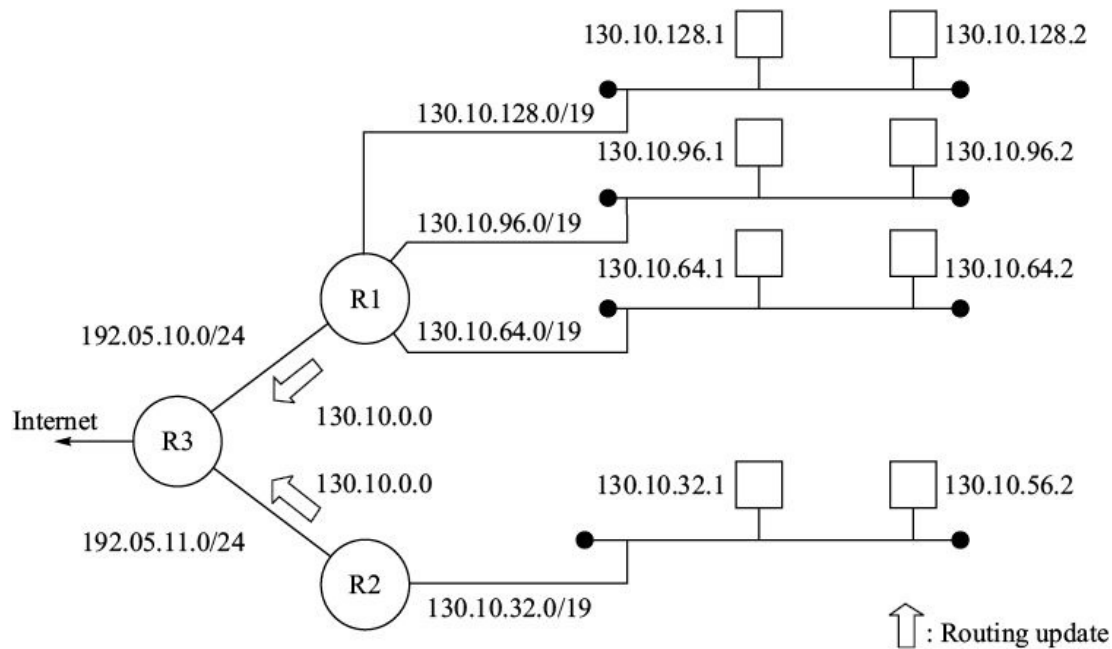


Figure 17.15 Classical subnetting.

R3 is concerned, R1 and R2 advertise the same network 130.10.0.0. It has two alternatives for sending an incoming IP packet with network address 130.10.0.0. It may send it to R1 even if the packet is actually meant for subnet 130.10.32.0/19 via R2.

In real life, need for subnetting arises because the subnets are not contiguous. Therefore, classical subnetting has limited applications. If routers R1 and R2 advertise their subnet masks and R3 populates its forwarding table with the subnet masks also, the problem can be resolved. In Figure 17.16, R1 and R2 advertise their subnet masks which are taken into account by R3

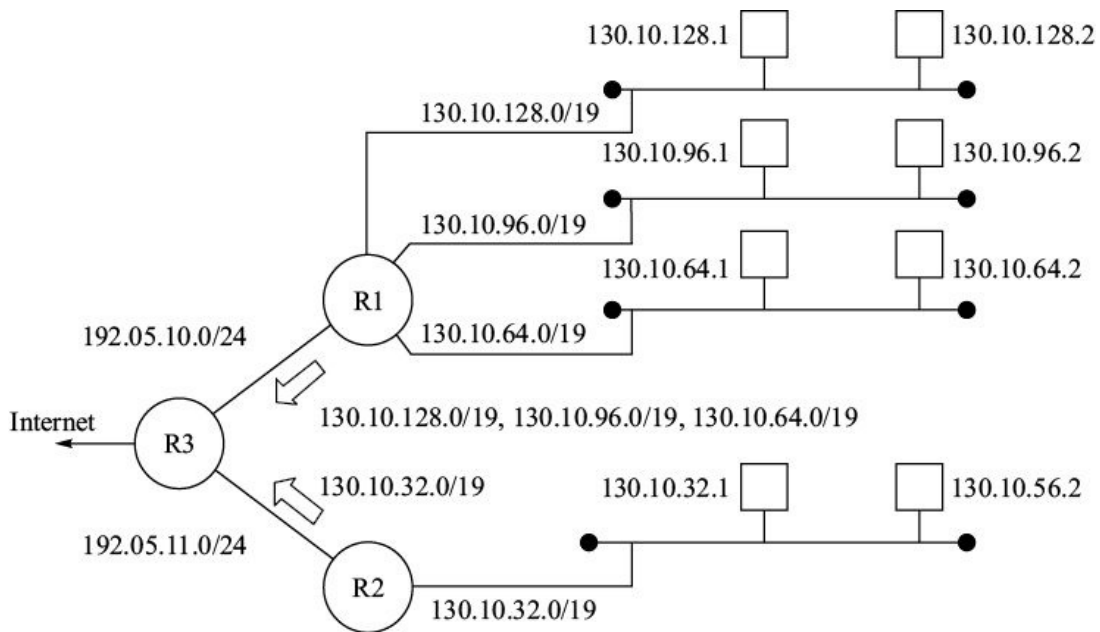


FIGURE 17.16 Classless subnetting.

when it updates its forwarding table. The forwarding table now contains subnet address of 19 bits. IP packets meant for a subnet are forwarded by R3 to the respective router R1 or R2 wherever the subnet exists. As we shall see in the next chapter, routing protocols RIP2 and OSPF advertise subnet masks, but RIP1 does not do so and has a major limitation. Once we deploy routing protocols that support advertisement of subnet masks, it is no longer necessary for the subnets to be contiguous.

17.5.3 Variable Length Subnet Mask (VLSM)

Classical subnetting allows division of a network into equal pieces. For example, in Figure 17.13, with the 3 bits for subnetting a class B address, we could create up to six subnets each having 8190 ($= 2^{13} - 2$) hosts. It is quite possible that some of these subnets may not have requirement of such large number of hosts and 13-bit host part may be an over provision. We would like that the subnets should have the appropriate size so that the address space is not wasted.

Variable length subnet mask (VLSM), as the name suggests, enables creation of subnets of unequal sizes. Each subnet can have a different length of subnet address depending on the number of host addresses required by it. Because the subnet masks are of different length in VLSM, it is essential that routing protocol supports advertisement of subnet mask as well. The contiguity requirement of subnets is also removed because the forwarding tables have subnet masks. As mentioned in the last section RIP2 and OSPF are two such

routing protocols.

Figure 17.17 shows an example of internetwork that uses variable length subnetwork mask. The class B network address allotted for the internetwork is 172.16.0.0. It is subnetted using 6 bits giving 62 subnets as under.

172.16.4.0/22, 172.16.8.0/22,, 172.16.248.0/22

Out of these, one subnets-id is used for WAN links, one for small ethernets and one for the corporate ethernet as under. The rest are reserved for future use.

- The WAN links need only two IP addresses, one for each end. Therefore, 172.16.4.0/22 is further subnetted using next 8 bits. The subnet address becomes 30 bits long with subnet mask 255.255.255.252. There can be 254 such subnets of 172.16.4.0/22. All 0s and all 1s subnet-ids are not used as before.

172.16.4.4/30, 172.16.4.8/30, 172.16.4.12/30, ..., 172.16.4.252/30

172.16.5.0/30, 172.16.5.4/30, 172.16.5.8/30, ..., 172.16.5.252/30

172.16.6.0/30, 172.16.6.4/30, 172.16.6.8/30, ..., 172.16.6.252/30

172.16.7.0/30, 172.16.7.4/30, 172.16.7.8/30, ..., 172.16.7.248/30

Three subnets out of these are used for the internetwork (Figure 17.17). The rest are reserved for future.

The last two bits of IP address are the host part of IP address and there can be two combinations 10 and 01. These are used as the port addresses of the adjacent routers interconnected on WAN links. All zeroes (00) and all ones (11) are not used as host part as explained earlier.

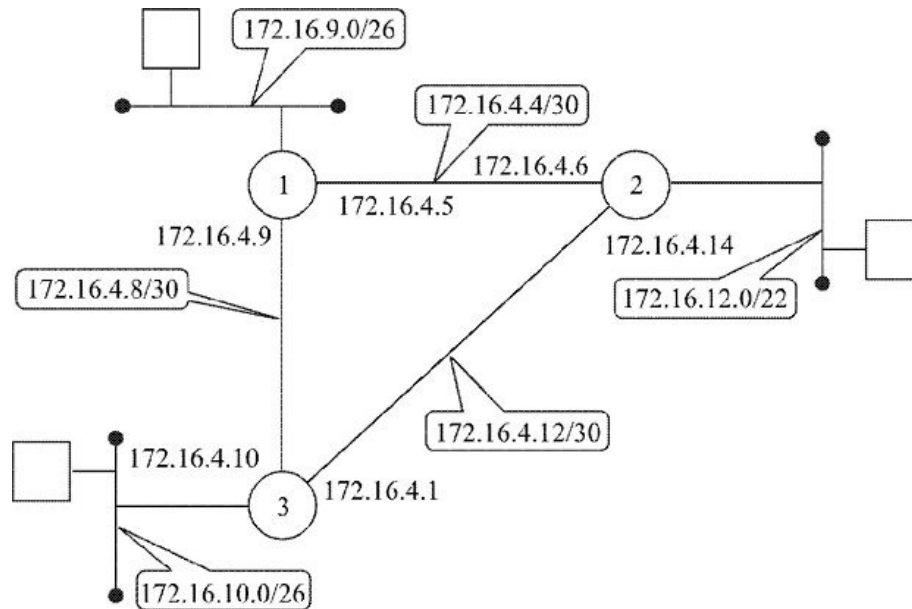


Figure 17.17 Example of VLSM.

- For small ethernet, subnet 172.16.8.0/22 is further subnetted using four more bits. The subnet address becomes 24 bits long with the subnet mask 255.255.252.0. There can be 14 such subnets of 172.16.8.0/22. Out of these, 172.16.8.0/24 and 172.16.9.0/24 are used for small ethernet.

The host part is 6 bits long and therefore each ethernet can support 62 ($2^6 - 2$) hosts.

- There is one local network which is required to support larger number of hosts. 172.16.12.0/22 is allotted for its use without further subnetting.

We notice that large part of class B address space is still unused but it can be used as the internetwork grows.

17.5.4 Classless Addressing and Supernetting The public Internet abolished classful addressing described above in favour of classless addressing to make efficient use of the available network address space. Classless addressing is described in RFC 1519. The network part of an IP address (Figure 17.10) is called *network prefix* in classless addressing. The boundary between the network prefix part and the host part is no longer the octet boundary. The network prefix can be of any length in the 32-bit IP address. The network prefix is

demarcated by network prefix mask which is again contiguous series of 1s up to the length network prefix. The rest of the bits are 0s in the mask. When a network is subnetted, the original network prefix gets extended by the number of bits borrowed from the host part.

An advantage of classless addressing is its capability to aggregate the contiguous multiple addresses into a single block of address called *Supernet* or *Classless Interdomain Routing (CIDR)* block. Supernetting reduces the size of forwarding tables by aggregating several routes into one route as illustrated by the example as follows: **EXAMPLE 17.4** A router receives four routes 57.6.96.0/21, 57.6.104.0/21, 57.6.112.0/21, and 57.6.120.0/21 having the same outgoing router port. Can these be aggregated to a common IP prefix?

Solution

57.6.96.0/21 = 00111001.00000110.01100000.00000000/21

57.6.104.0/21 = 00111001.00000110.01101000.00000000/21

57.6.112.0/21 = 00111001.00000110.01110000.00000000/21

57.6.120.0/21 = 00111001.00000110.01111000.00000000/21

The first 19 bits are same. Therefore, these routes can be aggregated to a single route 57.6.96.0/19 (00111001.00000110.01100000.00000000/19).

17.6 ADDRESS RESOLUTION PROTOCOL (ARP)

IP packets are forwarded by the routers based on the destination IP address on the IP packet. The forwarding tables have information about the next hop through which the destination can be reached. IP address on a packet is a layer-3 address and the IP packet is encapsulated in an HDLC, PPP or MAC frame, depending on the type of the link. Physical transmission of the frame on the link is determined by the layer-2 address. For example, in Figure 17.18, if the router R3, receives an IP packet with destination address 180.15.0.1, it must encapsulate the IP packet in a MAC frame with MAC address of the destination. Similarly to send an IP packet with destination address 10.0.0.2, the end system 180.15.0.1 must know the MAC address of the router R3.

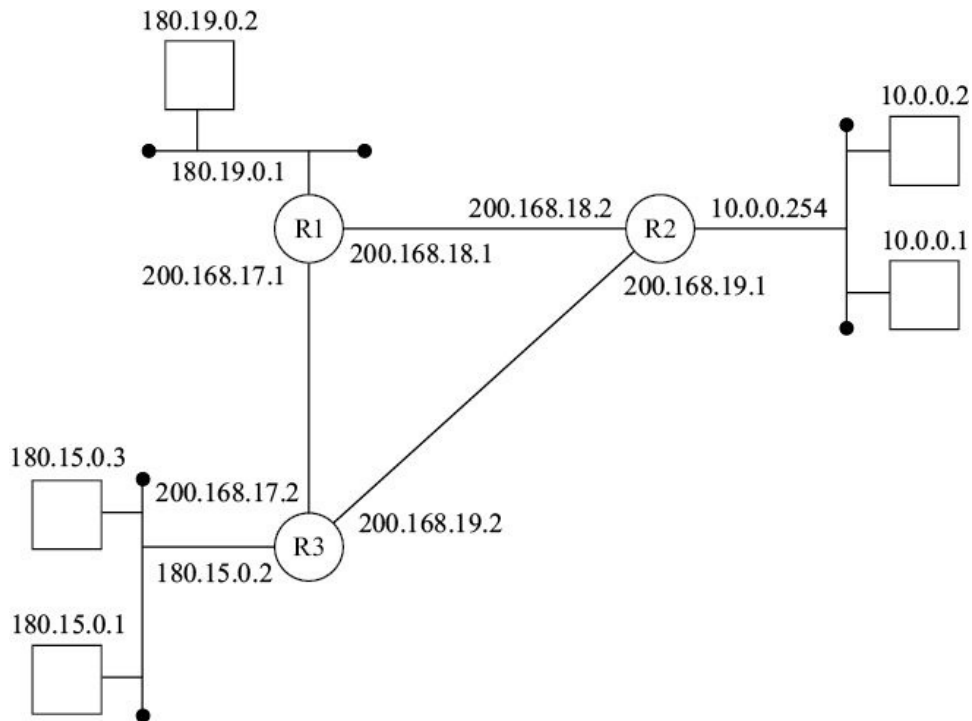


Figure 17.18 Need for MAC addresses for forwarding IP packets.

One way of doing this is to send the frame with broadcast address on the LAN. Each station on the LAN will accept the frame analyze it and hand over the contents of the information field of the frame to the IP layer. The IP layer will accept the IP packet if the packet is addressed to it. Since every IP packet, sent or received, will need to be broadcast in this manner, huge unnecessary processing load is there on all the LAN devices.

Another way to handle the situation is to map every IP address to specific MAC address. This mapping is carried out dynamically using Address Resolution Protocol (ARP), which we discuss in this section. ARP protocol is documented in RFC 826.

17.6.1 Layered Architecture for ARP

Address resolution protocol is designed to work for various networking technologies, IP, Ethernet, token rings, ATM, *etc.* We will limit our discussion to address resolution between local area networks and IP networks. As shown in Figure 17.19, ARP is a layer-3 protocol. It interfaces directly with Ethernet (DIX) at layer 2. In case of IEEE LANs, it requires IEEE 802.2 (SNAP) sublayer because the ‘type’ field available in Ethernet (DIX) is not available in IEEE 802.2 (LLC).

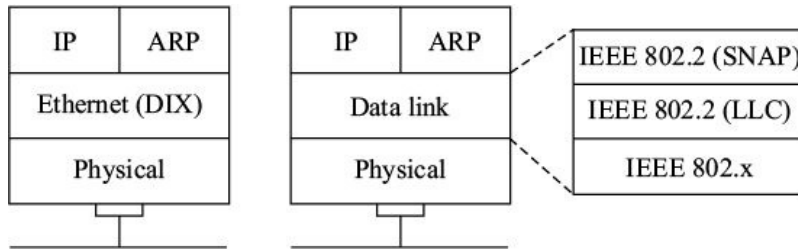


Figure 17.19 Layered architecture for ARP.

17.6.2 ARP Operation

ARP in the context of IP networks and local area networks is used in the following three situations (Figure 17.20):

- Host A wants to send an IP packet to host B on the same local area network but does not know its MAC address (Figure 17.20a).
- Host A wants to send an IP packet to host C on a different LAN. The two LANs are interconnected through the IP network. Host A knows the default gateway IP address of the router R1 but does not know the MAC address of the router R1 (Figure 17.20b).



Figure 17.20 Need for ARP for determining MAC addresses.

- Router R2 receives an IP packet (say from host A) for delivery to host C on the local area network but the router does not have the MAC address of host C (Figure 17.20c).

When a host wants to send an IP packet to another host, it first determines whether the destination is on the same network or on a different network. If the network prefixes of the source and destination IP addresses are different, the destination station is on a different network. If the network prefixes are same, the destination is on the same LAN.

ARP is used within a LAN to determine the MAC address of the station (or router) to whom an IP packet is to be delivered. The basic operation of ARP involves the following five steps. (We refer to hosts and routers as devices below. Thus two devices A and B mentioned below can be any combination of routers and hosts):

1. The device A that wants to locate the MAC address of device B having IP address (called target address), sends ARP-request packet with the IP and MAC addresses of the source device and the target IP address. The ARP-request packet is encapsulated in a MAC frame with broadcast address so that all the devices on the LAN receive it.
2. The frame is received by all the stations/ routers on the LAN. All the stations take note of the MAC and IP address of the source in their ARP cache for future use.
3. Device B having IP address same as the target IP address in the ARP-request packet, responds with ARP-response packet. The ARP-response packet contains its MAC address as one of the fields. The ARP-response packet is encapsulated in a MAC frame addressed to device A.
4. On receipt of ARP-response, device A takes note of the MAC address and IP address of B in its ARP cache. Having obtained the MAC address of B, A can now send the frames containing the IP packets directly to B.
5. Entries in ARP cache are deleted if they are not used for a defined period, which is usually 5 minutes.

If the destination is on the same LAN, the process described above is followed. If the destination is on a different network, the IP packet is to be sent to the router connected on the LAN. Therefore, the station sends ARP-request packet to the router on router's default gateway IP address. Default gateway IP address is preconfigured in each station on the LAN. Having obtained the MAC address of the router, it sends the IP packet addressed to destination host and frame encapsulated in MAC frame with the router's MAC address.

Figure 17.21a illustrates the process when router R3 in Figure 17.18 receives an IP packet addressed to 180.15.0.1. Figure 17.21b illustrates the case when an end station wants to determine the MAC address of the default gateway router.

It is possible for a station not to have IP address. A server on the LAN allots IP address when requested. Reverse ARP (RARP) protocol is used for this purpose. The station that needs an IP address, sends RARP-request packet to RARP server, which replies with RARP-response containing allotted IP address to the originator of RARP-request.

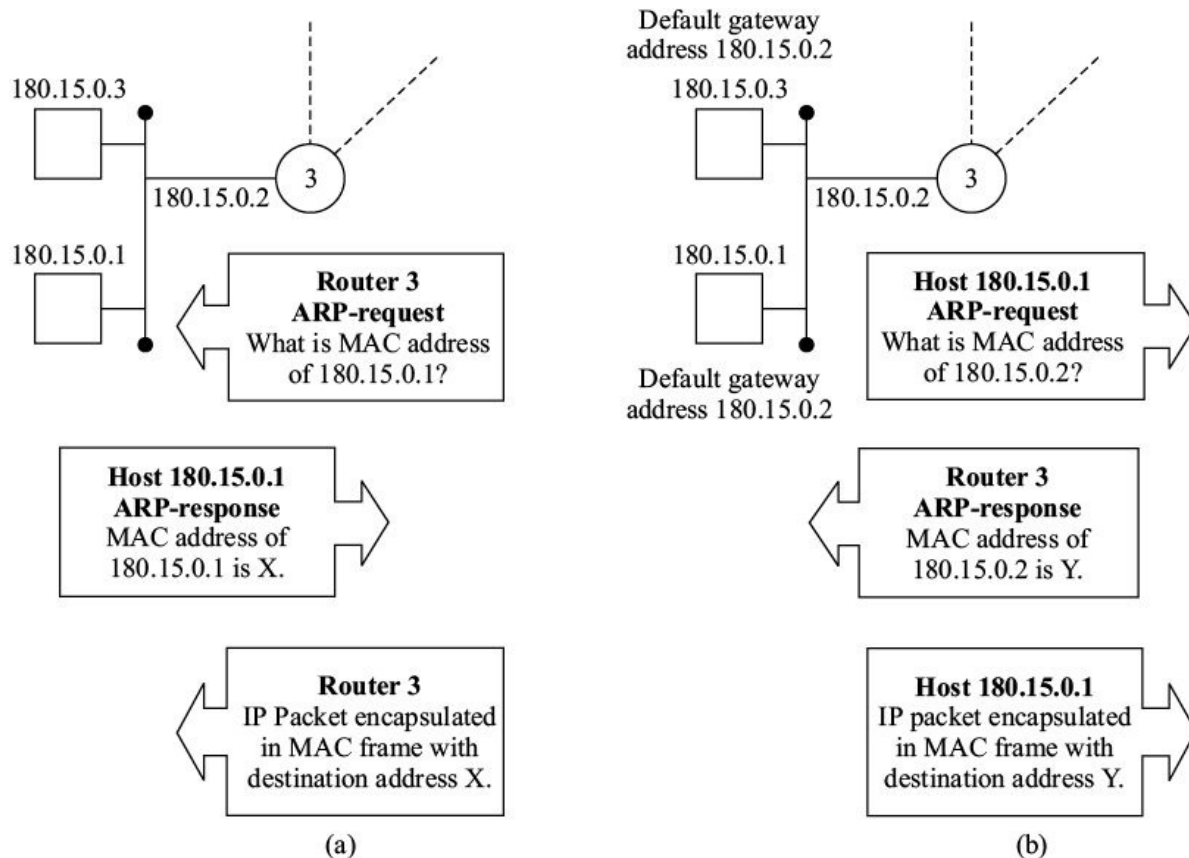


FIGURE 17.21 Address resolution protocol.

17.6.3 Format of ARP Packet

Figure 17.22a shows the format of ARP packet. The terminologies used in ARP is somewhat different. Use of terms ‘Hardware’ and ‘Protocol’ is in connection with layers 2 and 3 respectively. For example, ‘Protocol address’ refers to IP address. The target device is the device to which the ARP packet is sent.

Hardware type (HT). It indicates type of layer-2 network. For Ethernet (DIX), its value is 1. For IEEE 802 LANs, its value is 6.

Protocol type (PT). It indicates the protocol ARP is working for. For IP, as in this case, the value is 0x0800.

Hardware address length (HL). This field indicates the length of hardware address fields in octets. Its value is 6 octets for IEEE 802.x and Ethernet (DIX).

Protocol address length (PL). This field indicates the length of protocol address fields in octets. For IP addresses, its value is 4 octets.

Operation code (OC). This field indicates type of ARP/RARP packet.

- ARP-request 1
- ARP-response 2
- RARP-request 3
- RARP-response 4

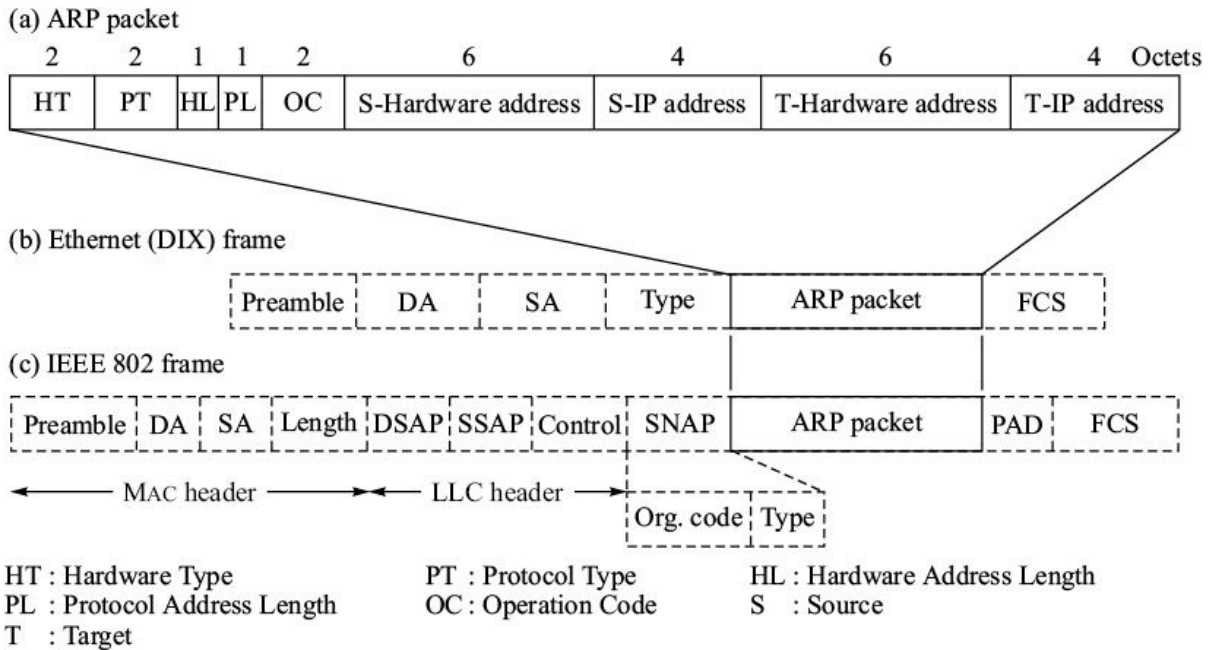


Figure 17.22 Format of ARP packet.

Source-hardware address. It contains the layer-2 address of the source. For Ethernet (DIX) and IEEE 802.x LANs, it is the source MAC address.

Source-protocol (IP) address. It contains the layer-3 address of the source. The IP address of the source is indicated here.

Target-hardware address. It contains the layer-2 address of the target device. For Ethernet and IEEE 802.x LANs, it is the MAC address.

Target-protocol (IP) address. It contains the layer-3 address of the target device. The IP address of the target is indicated here.

The ARP-request contains source-hardware address, source-protocol address, and target—protocol address. The target hardware address field is kept empty (all 1s). ARP-response contains all the four addresses, but source and target addresses get swapped. The responding device is now source.

The ARP packet is encapsulated in layer-2 frame. In case of Ethernet (DIX), it is directly inserted in the information field of the frame (Figure 17.22b). The protocol type field of the frame is 0x0806 for ARP packet.

For the IEEE 802.x LANs, ARP packet is prefixed with SNAP field which contains the organization code (3 octets) and type (2 octets) fields (Figure 17.22c). The type field is set to 0x0806 for ARP packet. The organization code is set to 0x000000. The DSAP and SSAP addresses of LLC header are set to 0xAA indicating presence of SNAP field. The control field is 0x03.

Reverse ARP (RARP, RFC 903) uses the same packet format as for ARP. Operation Code (OC) determines whether it is an ARP packet or RARP packet. The OC values are 3 for RARP-request and 4 for RARP-response.

17.6.4 Complete Picture of IP Packet Delivery It is now time we went through the entire process of IP packet delivery from a host to another host interconnected through an IP network. Let us assume that host with IP address 180.15.0.1 wants to send an IP packet to host having IP address 10.0.0.1 (Figure 17.23). We assume that routers R1, R2, and R3 are interconnected using a data link protocol (e.g. HDLC) which is already in data transfer mode. The local area network connecting the hosts and routers is ethernet. The process is as follows:

- Host 180.15.0.1 sends ARP-request packet using the target IP address (180.15.0.2) of its default gateway router R3. The packet is encapsulated in ethernet frame with destination MAC address as broadcast address.

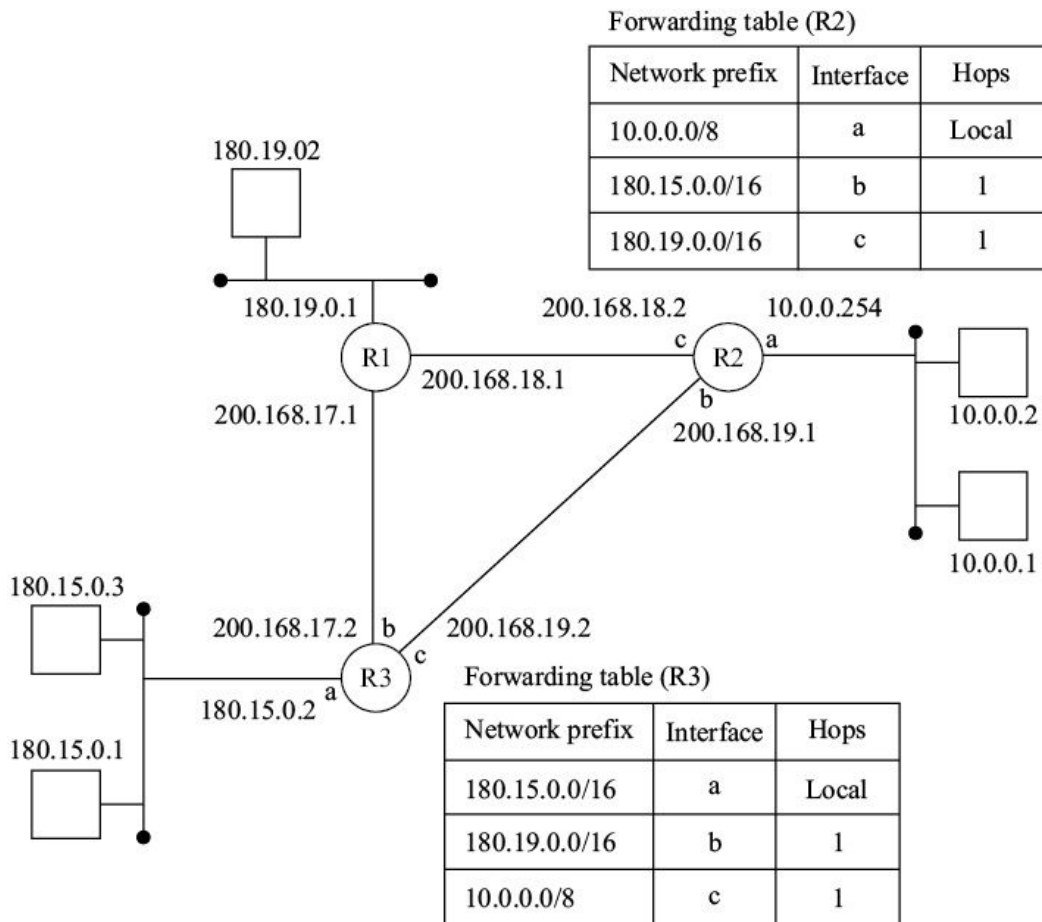


Figure 17.23 Forwarding of IP packets.

- R3 replies with ARP-response indicating its MAC address. ARP-response is encapsulated in ethernet frame with MAC addresses of R3 and the host.
- Host (180.15.0.1) sends the IP packet addressed to the host 10.0.0.1. The IP packet is encapsulated in the ethernet frame with the MAC address of R3.
- R3 receives the IP packet, consults its forwarding table, which indicates that the packet should be forwarded through its interface c. It encapsulates the packet in the data link frame (e.g. HDLC) and sends the frame to R2.
- R2 receives the frame and the encapsulated IP packet. It consults its forwarding table and determines that the destination IP address is on the local network. It sends an ARP-request packet with target IP address 10.0.0.1 to find the MAC address of the destination host. The ARP-request is sent using ethernet encapsulation and broadcast MAC address.
- Host 10.0.0.1 sends its ARP-response to R2 containing its MAC address. It uses ethernet encapsulation with MAC address of R2.
- R2 sends the IP packet received from host 180.15.0.1 to the host 10.0.0.1.

The packet is sent in an ethernet frame with MAC address of the destination host.

The subsequent IP packets from the host 180.15.0.1 to the host 10.0.0.1 do not invoke ARP since the ARP cache contains the required MAC addresses.

17.7 INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

Datagram service is the best-effort service. To enhance the reliability, Internet Control Message Protocol (ICMP) is used, which provides information about errors, loss of packets, unavailable destinations, *etc.* It is documented as RFC 792. It is mandatory that every device that implements IP, must also implement ICMP.

In ICMP, any destination or router that detects any problem in handling a received IP packet, generates an ICMP message addressed to the originating station of the IP packet. ICMP messages can be analyzed by network management systems to generate network reports for the network administrators. ICMP messages are sent as IP packets. The protocol field of IP header is set to 0x01 to indicate that this packet contains ICMP message (Figure 17.24).

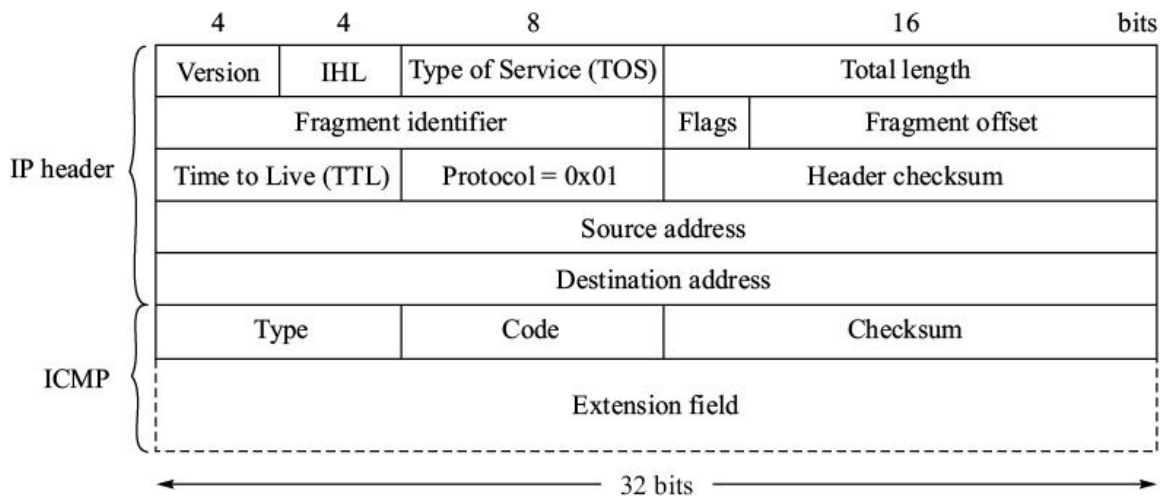


Figure 17.24 IP packet containing ICMP.

17.7.1 ICMP Message

Basic ICMP message consists of three fields—type, code, and checksum. Extension field is used with some of the messages. The type field specifies the message type (e.g. destination unreachable) and code describes the type (e.g. reason why the destination was unreachable). Important type-fields are described

in the following table (Table 17.5):

Type field	Message
0	Echo reply (PING)
3	Destination unreachable
5	Redirect
8	Echo request (PING)
11	TTL exceeded
12	Parameter problem (IP header)
13/14	Time stamp request/reply
15/16	Information (e.g. net-id) request/reply
17/18	Address mask request/reply

Echo (Types 0 and 8). When an IP packet containing ICMP echo request (Type 8) is sent to a host (or router interface), the host (or router interface) returns the IP packet with ICMP echo reply (Type 0) as shown in Figure 17.25a. The primary purpose of sending an echo request is to ascertain if a given IP address is reachable.

Receipt of echo reply indicates that

- there is a functioning path from the source to the destination,
- the interface with the destination IP address is up, and
- there is a functioning path from the destination interface to the source.

Remember that the paths to the destination and from the destination may be different. Non-receipt of echo reply indicates that any one or more of the above conditions are not met. One needs to look at other responses to determine true nature of the problem. For example, if the destination IP interface is down, then we should receive ICMP message ‘Destination unreachable (Type 3, code 1)’ as the IP packet could not be delivered.

It is possible that several echo requests are sent one after the other. To distinguish several echo requests/replies, ICMP messages have extension field that identifies the request and corresponding reply.

PING (Packet Internet Groper) is an application of ICMP echo. It is used for estimation of round trip delay, packet loss, and other parameters. Delay is measured by starting a timer at the time of sending the echo request and noting the time when echo reply is received. Several PINGs are sent one after the other and round-trip delay is expressed as minimum, maximum, and average values. Packet loss is estimated based on number of echo replies not received. If out of

1000 echo requests, only 990 are replied to, the packet loss is 1%.

TTL exceeded (Type 11). If an IP packet is discarded because the TTL specified in the packet header expired, this ICMP message is generated by the device that encounters TTL expiry. In Figure 17.25b, router R2 notices that TTL becomes zero if it sends the IP packet to router R3. It generates TTL exceeded ICMP packet and sends it to the source for taking appropriate action.

Note that the IP packet containing TTL exceeded message will bear the IP address of the intermediate node. This knowledge can be used to trace the route to a destination. Traceroute is an application based on TTL exceeded ICMP message, where a series of IP packets with TTL values starting from 1 are sent to a destination. The first router that receives the first IP packet finds that TTL has exceeded and therefore sends TTL exceeded ICMP packet to the source. On receipt of the ICMP packet, the source comes to know the IP address of the interface of the router. The second IP packet with TTL = 2 is able to reach up to the next router before its TTL expires. The second router also sends TTL exceeded ICMP message to the source. In this way, the source receives a series of TTL exceeded ICMP messages from the routers enroute to the destination. To ensure that the destination also sends an ICMP message, the UDP port in the IP packets sent by the source is kept at value which is not used, say above 33434. The destination returns ICMP packet of Type 3 (Destination unreachable) with code 3 (port unreachable). With the receipt of this ICMP packet, the route up to the destination has been traced.

Traceroute combines the round trip delay measurement also up to each router on the route. A timer is started when the IP packet is sent by the source and time is measured when TTL exceeded/destination unreachable ICMP packet is received.

Destination unreachable (Type 3). If an IP packet cannot be delivered to the destination due to any of the following reasons, 'Destination unreachable' ICMP message is sent to the source by the router/host that encounters the problem (Figure 17.25c).

<i>Code</i>	<i>Description</i>
0	Network is unreachable. Network is down or no further path is known. ICMP packet is generated by an intermediate or the far end router.
1	Host is unreachable or not responding. ICMP packet is generated by the far end router.

- 2 Protocol is unreachable. The protocol specified in the IP packet header is not available in the destination host. ICMP packet is generated by the destination host.
- 3 Port is unreachable. Service port specified in layer 4 is not available. ICMP packet is generated by the destination host.
- 4 Fragmentation is needed but DF (do not fragment) bit is set to 1. ICMP packet is generated by the intermediate router.
- 5 Source route has failed. Path specified in the IP options part cannot be followed. ICMP packet is generated by the routers enroute.

Redirect (Type 5). If a router knows a better path to send an IP packet to the destination, it forwards the packet towards the destination but it sends 'Redirect' ICMP message packet to the source. Figure 17.25d shows a case where host A sends IP packet (DA 3.0.0.1) to the default gateway router R1. Router R1 forwards the packet to R4 but also sends 'Redirect' ICMP message to the source so that the source may send the IP packets meant for 3.0.0.0 network directly to router R4.

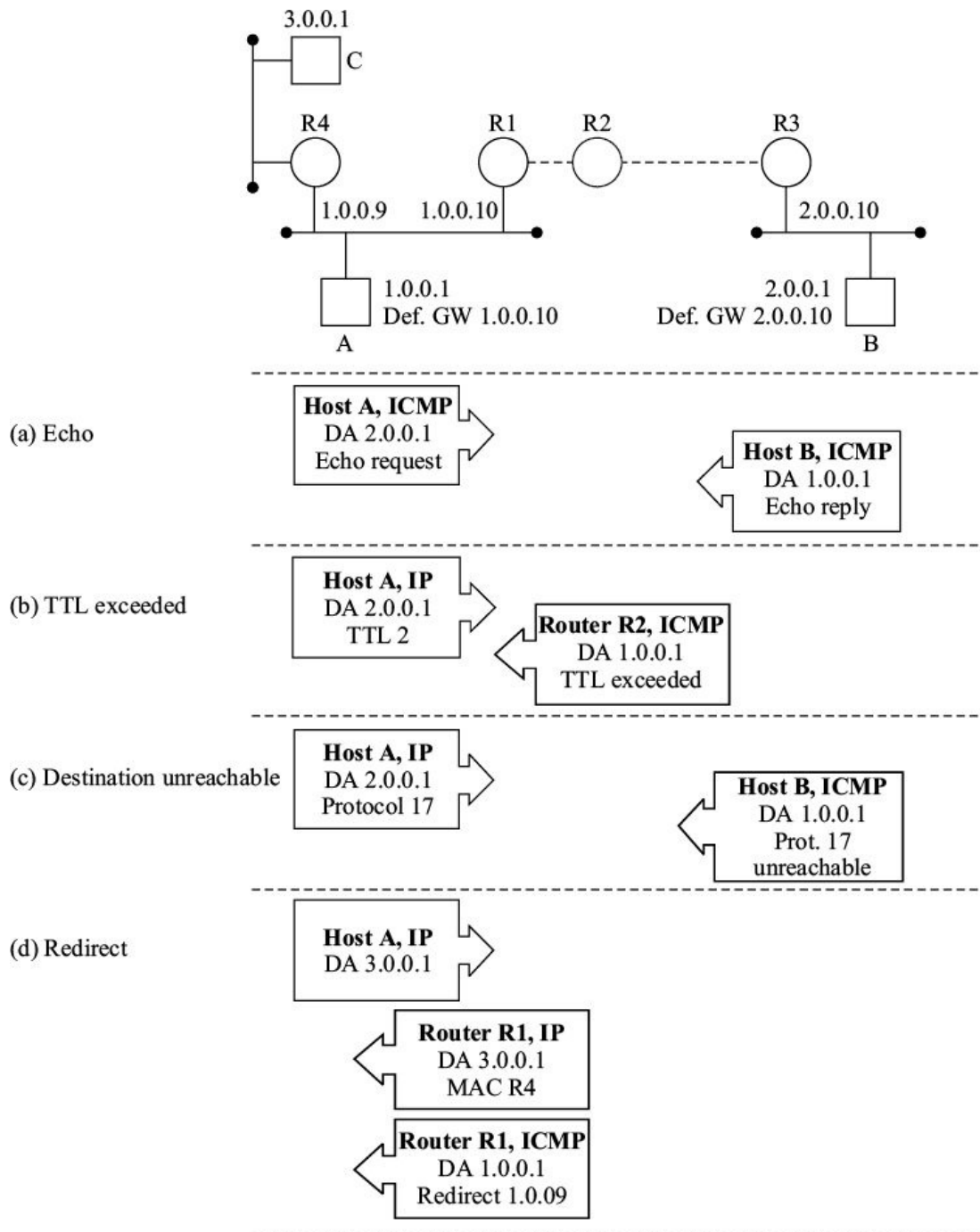


Figure 17.25 ICMP messages.

17.8 IPV6 INTERNET PROTOCOL

The phenomenal growth of the Internet and growth of its scope of applications during last decade brought out the deficiencies in IPv4 design. Some of the important issues that need to be addressed are as follows:

1. Two-level address structure of IPv4 and 32-bit addresses is inadequate to meet the requirements of network prefixes. The address structure with five classes is a waste in utilizing the available address space.
2. The new real time applications (video/audio/speech) require defined maximum delay, jitter and packet loss. Best effort service of IPv4 is not good enough. Strategies for resource reservation and special handling of certain class of packets need be built into the network protocol.
3. Encryption and authentication data must be supported end to end.

IPv6 is the next generation of Internet Protocol that will replace IPv4. It has evolved over a period of one decade but it is yet to be rolled out. In this section we get introduced to IPv6. The overall picture of IPv6 is beyond the scope of this text.

17.8.1 Format of IPv6 Packet

Figure 17.26 shows the format of IPv6 packet. It consists of a fixed base header followed by payload (Figure 17.26a). The payload consists of optional extension headers and data octets from the upper layer.

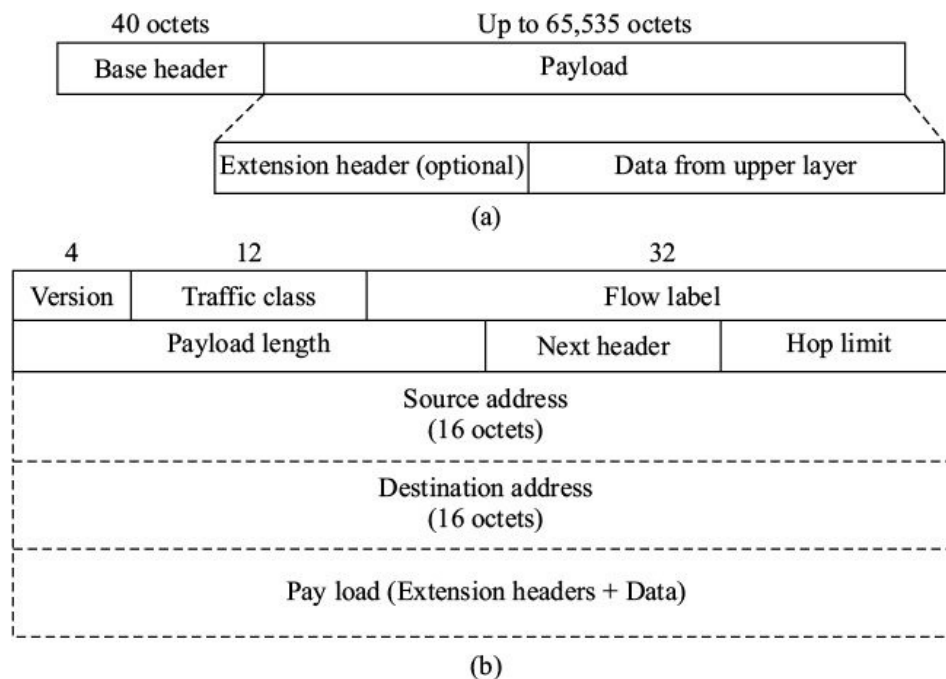


Figure 17.26 IPv6 packet format.

The base header has fixed length of 40 octets. It consists of the following fields: **Version (4 bits)**. It indicates the version of Internet Protocol which is 6

for IPv6.

Traffic class (8 bits). It is used for specifying the class of traffic to which the IP packet belongs. The class of traffic determines the priority level to be accorded to the IP packet by a router.

Flow label (20 bits). It is used to identify all the packets in an individual flow. A flow is uniquely identified by a combination of source address, destination address, and a non-zero flow label. Thus, all the packets that are part of the same flow are assigned the same label by the source.

Payload length (16 bits). It indicates number of octets present in the payload. Maximum payload length can be 65,535 octets.

Next header (8 bits). It describes the next header after the base header. The next header can be an extension header (described later) or header of the upper layer. In the later case, the field indicates the protocol, *e.g.* 6 for TCP, 17 for UDP, 58 for ICMP of IPv6, 89 for OSPF, 4 for IP packet of IPv4. An IP packet of IPv6 can contain in its payload an IP packet of IPv4.

Hop limit (8 bits). It has the same function as TTL in IPv4. In IPv6, it is decremented by one on each hop.

Source and destination addresses. The source and destination addresses are 128-bit long.

17.8.2 Extension Headers

The base header of IPv6 packet can be followed by one or more extension headers. Extension headers correspond to the options field of IPv4. There are six types of extension headers as listed below. Their placement in the IP packet follows the listed order. An IP packet may not contain any extension header. The extension header is identified by its type code which is indicated in the next-header field of the preceding header.

	Type code
(a) Hop-by-hop options header	0
(b) Routing header	43
(c) Fragment header	44
(d) Encapsulating-security-payload header	50
(e) Authentication header	51
(f) Destination options header	60

Hop-by-hop options header. It specifies the options to be processed by the intermediate nodes.

Routing header. It is used when source routing is used. It contains the list of intermediate nodes to be visited by an IP packet. The destination address field of the base header contains the address of the next hop copied from the routing extension header.

Fragment header. Fragmentation in IPv6 is carried out by the end systems only. The routers do not fragment the packets. Fragment extension header is inserted and used by the end systems to fragment and reassemble the data packets. It has fragment-offset field, identification and more fragment fields. ‘Do not fragment’ bit is not required because routers do not fragment the packets.

Encapsulating-security-payload header. It contains encrypted data for its secure transfer across the internet.

Authentication header. It is used for authenticating the source of an IP packet.

Destination options header. Destination options header is defined for use by the destination end system. Its actual application is yet to be defined.

Each extension header has its own set of fields. The first field is always the next-header field in all the extension headers. Formats of fragmentation extension header, routing extension header, and options extension header are shown in Figure 17.27.

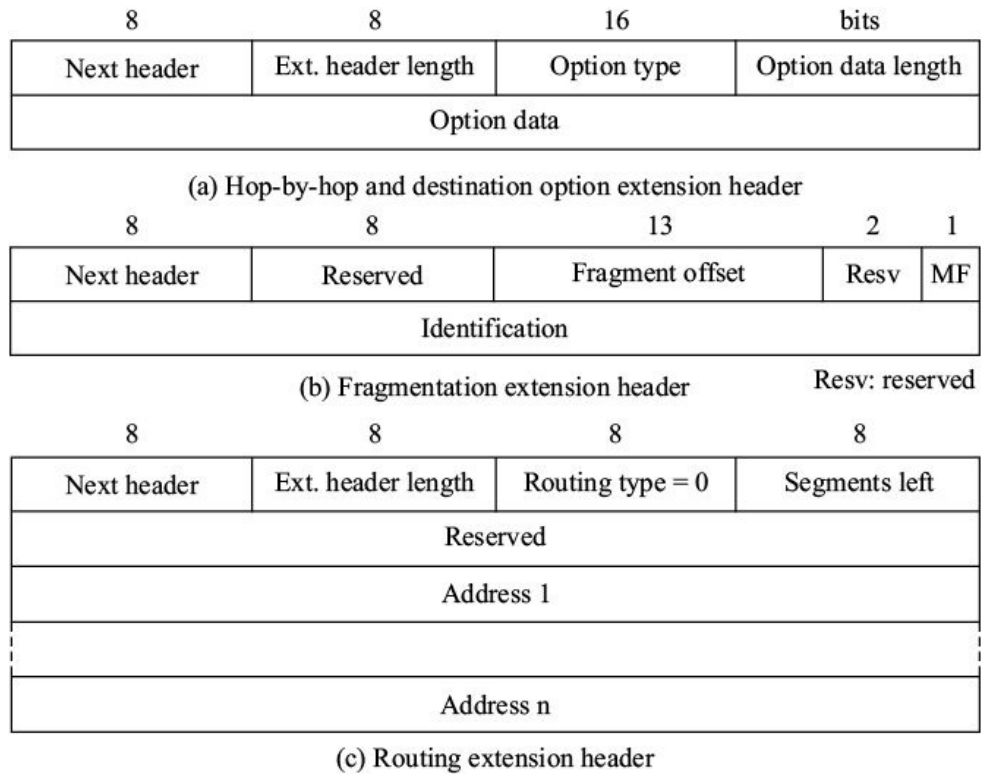


Figure 17.27 Extension headers.

17.8.3 Address Notation in IPv6

IPv6 uses 128-bit addresses against 32-bit addresses used in IPv4. The following notation is used for specifying an address in IPv6:

- The 128-bit address is separated in eight 16-bit parts. These parts are separated by a colon.
- Each 16-bit part is represented using four hexadecimal numbers. For example,
FEDC : 00C3 : 0000 : 0000 : 0000 : 34GE : 7354 : 3510
- Leading zeroes of each part can be suppressed.
FEDC : C3 : 0 : 0 : 0 : 34GE : 7354 : 3510
- Set of all consecutive zeroes of 16-bit parts can be put between two colons. A double colon can be used only once in an address.
FEDC : C3 :: 34GE : 7354 : 3510

17.8.4 Comparison of IPv6 and IPv4 Headers

The major differences in IPv4 and IPv6 packet formats are as follows:

1. IPv6 base header has fixed length of 40 octets. IPv4 header is variable in length and requires header length field (IHL).
2. Header checksum field is absent in IPv6. Thus error detection is not carried out on the header. It reduces the processing time of an IP packet.
3. There is no fragmentation field in the base header in IPv6.
4. In IPv4, the total size of IP packet including header is specified. In IPv6, the size of payload (excluding the header) is specified.
5. TTL field of IPv4 specified time to live in seconds. In IPv6, maximum hop limit is specified.
6. Traffic class field of IPv6 are equivalent to the DSCP field of IPv4.
7. The source and destination address sizes in IPv6 are 128 bits as against 32 bits in IPv4.
8. Options field is moved under extension headers in IPv6.

17.9 ISO CONNECTIONLESS-MODE NETWORK PROTOCOL (CLNP)

The ISO Internet Protocol is described in ISO 8473, *Protocol for Providing the Connectionless-mode Network Service*. ISO 8473 provides Connectionless-mode Network Service (CLNS) to the transport layer in the end systems. It uses the connectionless-mode service provided by the next lower layer. It works in conjunction with the following routing protocols, which we will describe in Chapter 18.

- ISO 9574, *End system to Intermediate System Routing Exchange Protocol for Providing the Connectionless-mode Network Service*.
- ISO 10589, *Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for Use in conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)*.

IP has much wider industry acceptance than ISO 8473. Most of the networks today are based on IP. Therefore, we will restrict description of ISO 8473 to the format of its datagram. Most of the functional aspects of the protocol are similar to IP.

17.9.1 Types of IPDU

Layer 3 data unit in ISO 8473 is called Internet Protocol Data Unit (IPDU). It is

formed by adding a header to the data unit received from the transport layer. IPDU is equivalent to IP packet of Internet. ISO 8473 defines two types of IPDUs:

- Data IPDU
- Error report IPDU.

Data IPDU. It carries user data between two end systems.

Error report IPDU. Error report IPDU is returned to the originating end system when a data IPDU is discarded at any router or end system. There can be several reasons for the discarding of an IPDU. The error report IPDU is generated only if the originating end system has requested for the error report in its data IPDU.

17.10 FORMAT OF ISO 8473 IPDU

The format of IPDU can be divided into five parts. The first four parts constitute the header and the fifth part contains user data (Figure 17.28). The four parts of the header are:

1. Fixed part
2. Address part
3. Segmentation part
4. Options part.

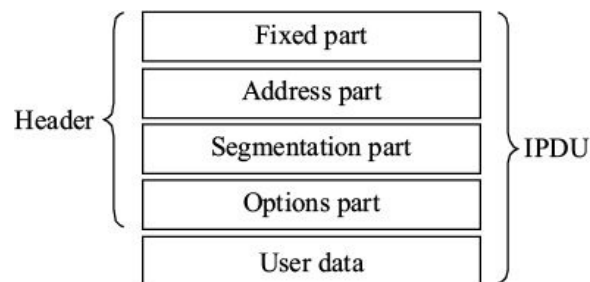


Figure 17.28 Structure of an ISO 8473 IPDU.

17.10.1 Fixed Part

The fixed part of the header is always present in all IPDUs and has fixed length. It consists of the following fields (Figure 17.29).

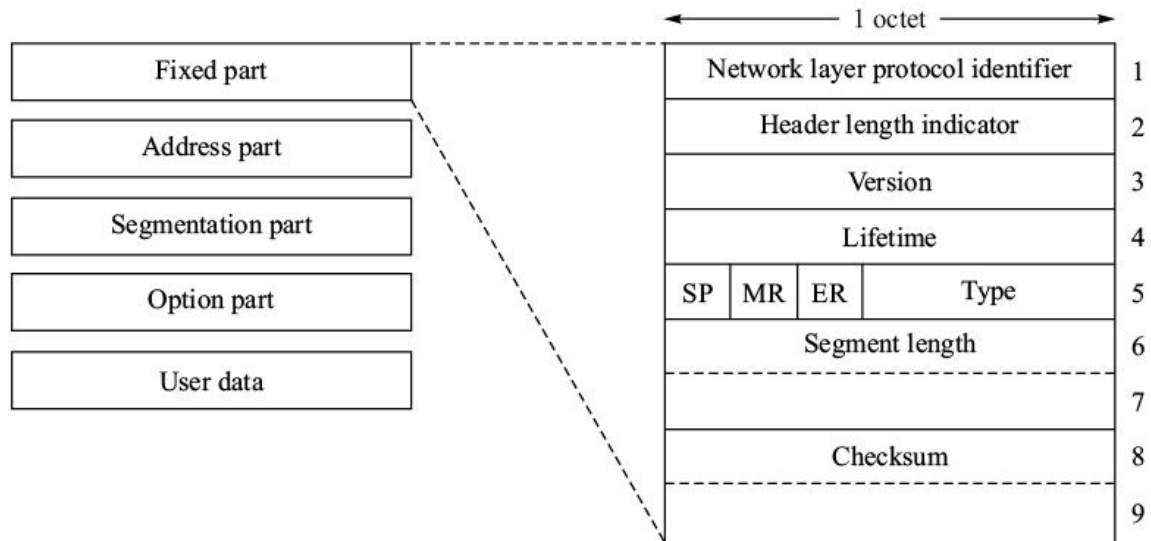


Figure 17.29 Format of the fixed part of ISO 8473 IPDU.

Network layer protocol identifier (1 octet). This field identifies the protocol. For ISO 8473 protocol, the value in this field is 12.

Header length indicator (1 octet). The length of the header of this particular IPDU is specified in this field.

Version (1 octet). The version of the protocol is indicated in this field.

Lifetime (1 octet). The remaining lifetime of the IPDU in units of $\frac{1}{2}$ second is indicated in this field.

Segmentation permitted, SP (1 bit). If segmentation of the IPDU is permitted, this field contains the value 1. Segmentation is equivalent to fragmentation in IP.

More segments, MR (1 bit). This field contains 1 if more segments of a segmented IPDU follow this one. The last segment contains 0 in this field.

Error report, ER (1 bit). This field is used to request the return of the error report IPDU, if this packet is discarded. Error report is sent if ER = 1.

Type (5 bits). This field indicates the type of IPDU. It contains value 28 if it is data IPDU and value 1 if it is an error report IPDU.

Segment length (2 octets). The length of the IPDU including the header is indicated in the field.

Checksum (2 octets). It contains a two-octet checksum of the header portion for error detection.

17.10.2 Address Part

The address part contains source and destination addresses and is variable in size (Figure 17.30).

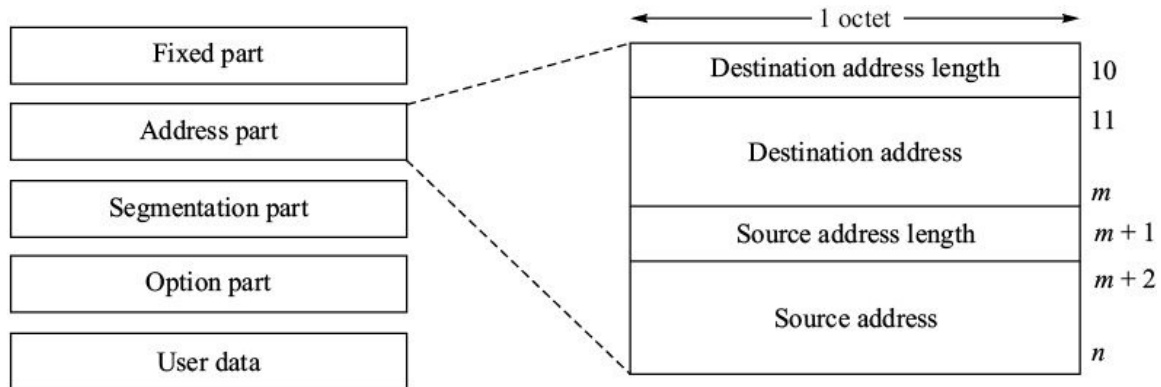


Figure 17.30 Format of the address part of ISO 8473 IPDU.

Destination address length (1 octet). The length of the destination address field is specified in this field.

Destination address (20 octets). This field contains the destination address. The maximum length of the address field is 20 octets.

Source address length (1 octet). The length of the source address field is specified in this field.

Source address (20 octets). This contains the address of the source. The maximum length of the address field is 20 octets.

17.10.3 Segmentation Part

If the SP flag is set to 1, a 6-octet segment part is included in the header. It contains the following fields (Figure 17.31).

Data unit identifier (2 octets). It is a unique identifier generated by the source. It identifies the unsegmented IPDU and all its segments.

Segment offset (2 octets). The number in this field gives the relative position of this segment within the unsegmented IPDU.

Total length (2 octets). This field contains the total length of the entire IPDU before segmentation.

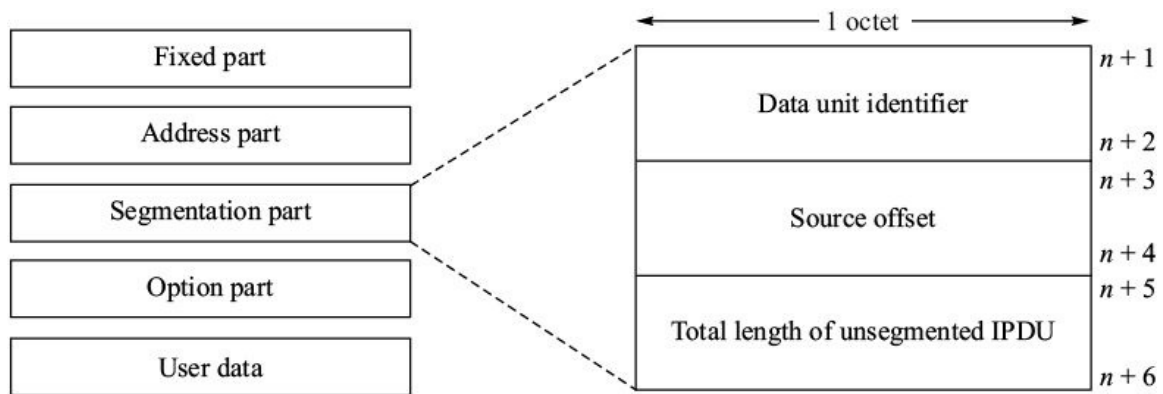


Figure 17.31 Format of the segmentation part of ISO 8473 IPDU.

17.10.4 Options Part

The options part is of variable length. It contains several groups of fields which are used for the following functions (Figure 17.32).

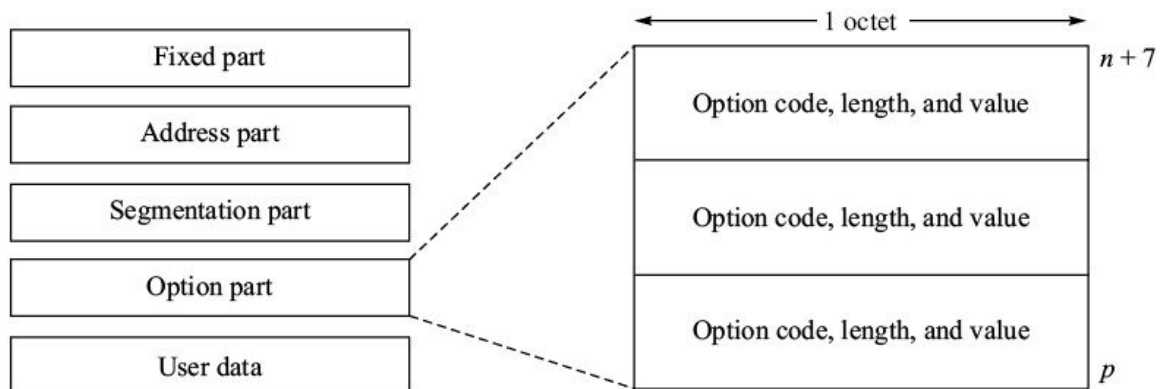


Figure 17.32 Format of the options part of ISO 8473 IPDU.

Padding. It is used to align the IPDU size to the desired boundary.

Security. This field contains security information.

Source routing. When source routing is used, it gives the route of the IPDU.

Route recording. In this field, the route of the IPDU is recorded as it travels through the network.

Quality of service. This field contains values which describe the quality of service parameters.

Priority. This field contains the relative priority of the IPDU.

Reason for discard. It is used in the error report IPDUs to indicate the reason a packet was discarded.

17.11 POINT-TO-POINT PROTOCOL (PPP)

The Point-to-Point Protocol (PPP) originally emerged as layer-2 data link protocol for transporting IP packets over point-to-point serial physical links having interfaces such as V.35, E1, EIA 232D, EIA 449, *etc.* Today it is industry-standard for dial-in Internet service. The most important feature that distinguishes PPP from other data link protocols is that it supports encapsulation of different layer-3 protocol packets over a common layer-2 link. It is based on HDLC protocol and can be used between two routers or between a host and a router. It is an IETF standard protocol specified in RFC-1661.

17.11.1 Layered Architecture of PPP

Figure 17.33 shows the layered architecture of PPP. PPP is a data link layer protocol that provides service to number of network layer protocols (IP, ISO 8473, Novell IPX, DECnet IV, *etc.*). The network layer protocol data units are encapsulated into a frame by PPP and are handed over to the physical layer. In addition, PPP performs certain other functions as follows:

- It manages (establishment, maintenance, and termination) point-to-point data links and negotiates various options (e.g. authentication). For this purpose PPP has Link Control Protocol (LCP).
- PPP has Network Control Protocol (NCP) for each specific type of network layer (e.g. IP, IPX, ISO 8473). NCP negotiates the operational parameters (e.g. size of N-PDU, multilink PPP, compression).

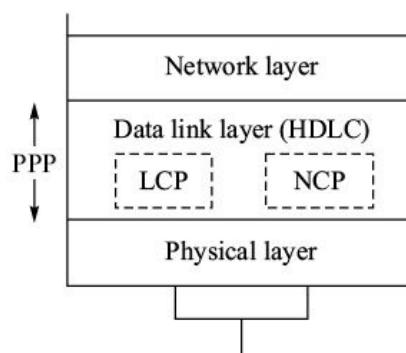


Figure 17.33 Layered architecture of PPP.

17.11.2 Physical Layer for PPP

No physical layer protocol is specified for PPP. It can operate through various physical layer interfaces:

1. EIA-232-D
2. EIA-422/423
3. V.35
4. G 703 (E1 interface).

PPP requires full-duplex circuit, dedicated or switched (POTS or ISDN) circuit that can operate in asynchronous or synchronous transmission mode. There is no limitation on bit rate other than that imposed by the physical layer interface.

17.11.3 PPP Frame Format

Format of PPP frame is based on HDLC protocol (Figure 17.34).

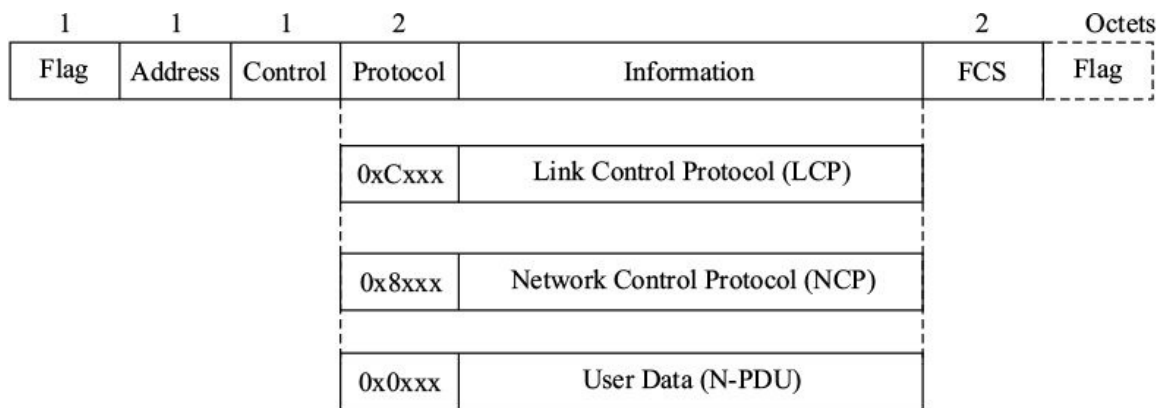


Figure 17.34 PPP frame format.

Flag. Flag (01111110) identifies start and end of the frame.

Control field. The control field (11000000) identifies the frame as unnumbered frame.

Address. Since it is a point-to-point link, the address field is simply all 1s (broadcast address).

Protocol. The protocol and information fields are linked together. Protocol field is 2 octets long and identifies the protocol being carried in the information field, which could be LCP packet or NCP packet, authentication packet or simply data packet received from the network layer. For example, if the information field carries an IP packet, the protocol field is 0x0021.

Some of protocols and their identifiers are given in Table 17.6. Layer-3 network protocol identifiers start with 0, Link Control Protocol (LCP) identifiers start with C, and Network Control Protocol (NCP) identifiers start with 8.

TABLE 17.6 Protocol Identifiers Used in PPP

Identifier	LCP	Identifier	NCP	Identifier	Network layer
0xC021	Link control protocol	0x8021	IP control protocol	0x0021	IP
0xC023	Authentication	0x802b	IPX control protocol	0x002b	Novell IPX
		0x8027	DECnet control protocol	0x0027	DECnet Phase IV
		0x8023	OSI control protocol		

Information. The information field carries data pertaining to the protocol specified in the protocol field. The default maximum size of information field is 1500 octets.

Frame check sequence (FCS). It is normally 2 octets long. It can be 4 octets long by prior arrangement.

17.11.4 PPP Operation

PPP operation is carried out jointly by the LCP and NCP protocols to service the network layer. Typical sequence of events that take place from link establishment, data transfer, and termination is shown in Figure 17.35. Each phase may consist of several steps. The protocol involved in a particular phase is indicated in parentheses in the figure. Although LCP's involvement is shown during link establishment and termination phases, it is active all the time and monitors the link. If there is an error condition, it is reported using LCP.

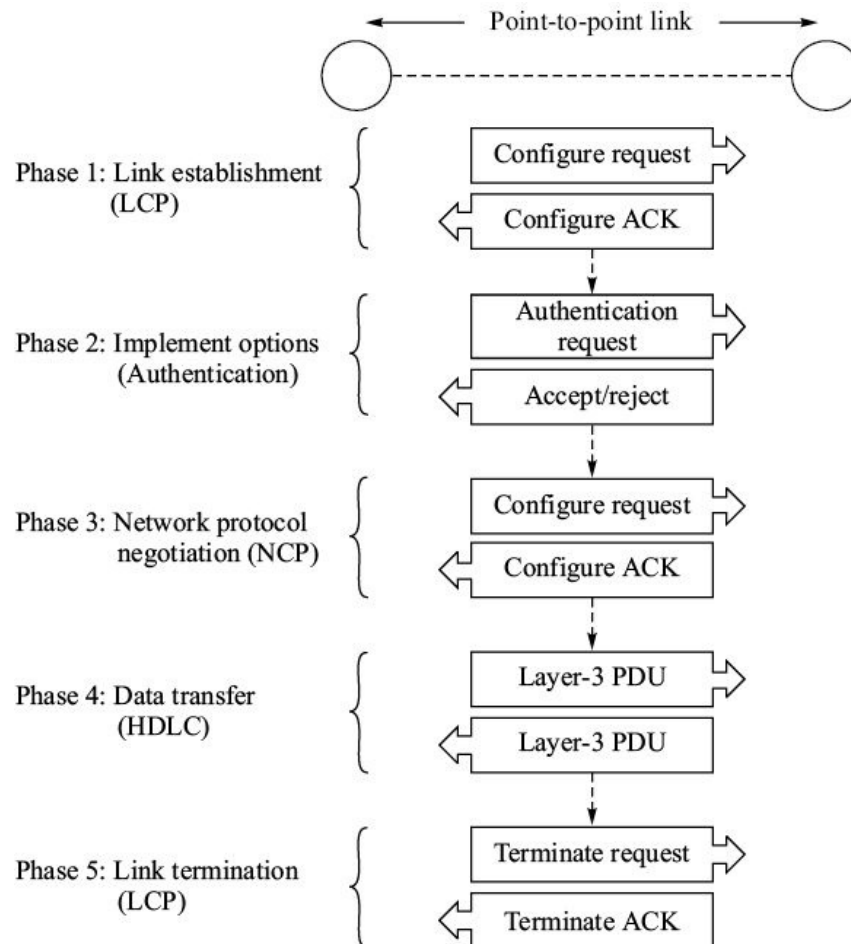


Figure 17.35 PPP operation.

Phase 1. LCP is used to

- agree the encapsulation options (e.g. FCS size),
- agree on the maximum size of information field,
- agree on the authentication protocol (default is OFF), and
- agree on the compression of protocol, control, and address field.

Phase 2. The chosen options are implemented in this phase. For example, if authentication is the chosen option, authentication is carried out using Password Authentication Protocol (PAP) or Challenge Handshake Authentication Protocol (CHAP). If authentication is successful, the next phase using NCP follows. If authentication is not successful, the link is terminated using LCP.

Phase 3. Network Control Protocol (NCP) is used for negotiating and configuring the network layer protocol parameters. For each network layer

protocol, there is a corresponding NCP. The parameters negotiated depend on the type of network layer protocol. In the case of IP, IP address, default gateway, DNS server, header compression, *etc.* are the parameters negotiated.

Phase 4. In this phase, the exchange of data units received from the network layer takes place. Each data unit is packed in the information field of an HDLC frame. The protocol field identifies the type of network layer protocol (IP, IPX, *etc.*) being used.

Phase 5. Termination of the link is carried out by LCP. It can be initiated by any party at the two ends of the point-to-point link.

17.12 LINK CONTROL PROTOCOL (LCP)

The link control protocol is responsible for establishing, configuring, maintaining, and terminating the point-to-point link. It also provides negotiation mechanism for requesting and accepting options.

LCP packets are carried in the information field of the HDLC frame and are identified as being LCP packet by 0xC021 in the protocol field. The format of LCP packet is shown in Figure 17.36.

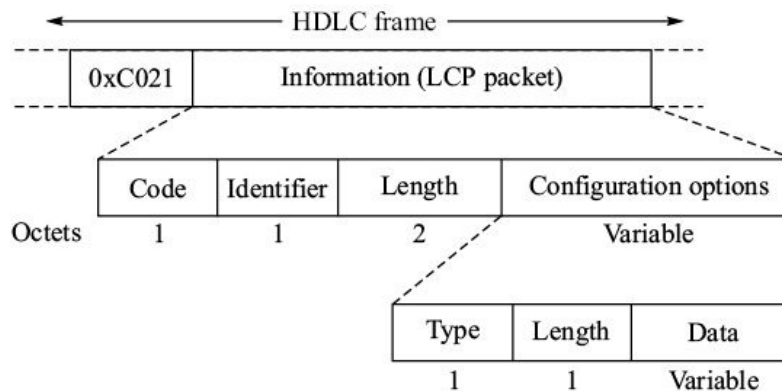


Figure 17.36 LCP packet format.

Code. Code defines the type of LCP packet. It is one octet long.

Identifier. A request is identified by this field. The response to the request carries the same identifier. This field is one octet long.

Length. This field indicates the length of the whole LCP packet. It is two octets long.

Configuration options. This field contains the options associated with the request and response. It contains the following subfields:

- Type (1 octet)
- Length (1 octet)
- Data (variable).

17.12.1 LCP Packets

LCP packets are of several kinds. Configuration packets are used during link establishment. Termination packets are used for terminating the link. Then there are packets used for link monitoring. Link monitoring packets are used for reporting error situations and for testing the link.

Configuration packets. Configuration packets are exchanged for establishing a link and negotiating options.

- Configure-request (0x01) is request to establish a link. It contains the options requested.
- Configure-ack (0x02) conveys acceptance of configure-request and all options specified therein.
- Configure-nak (0x03) is sent in response to configure-request if some or all options are not acceptable.
- Configure-reject (0x04) is sent in response to configure-request if the option(s) are not recognized.

Link termination packets. Link termination packets are used to discontinue the link.

- Terminate-request (0x05) is sent for terminating a link. Either party can terminate a link.
- Terminate-ack (0x06) is sent in response to the terminate request.

Link monitoring packets. These packets are used for link monitoring.

- Code-reject (0x07) is sent if the code in an LCP packet is not recognized by the receiving end.
- Protocol-reject (0x08) is sent if the protocol field in the frame contains unrecognized protocol.
- Echo-request (0x09) is used for link monitoring. The sender expects to receive echo-reply packet in response to this packet.

- Echo-reply (0x0A) is sent in response to echo-request. Echo request and reply use a number for identification called magic number. It is unique to an LCP entity. Magic number enables detection of loopback condition² at the physical layer. If an entity receives an echo request or reply with its own magic number, it implies that there is physical link in loopback condition.
- Discard-request (0x0B) is used for debugging if there is problem in sending the frames on the physical medium at one end of the link. The other end is not involved in this test. If the other end receives this packet, it just discards it.

17.13 AUTHENTICATION PROTOCOLS

Authentication protocols include Password Authentication Protocol (PAP, RFC 1334), Challenge Handshake Authentication Protocol (CHAP, RFC 1994), and Extensible Authentication Protocol (EAP, RFC 2284). We will examine briefly PAP and CHAP in this text.

17.13.1 PAP

Password Authentication Protocol (PAP) is the simplest and least secure authentication protocol. It is a two-way handshake protocol, meaning that exchange of only two packets is required between the two ends to complete the authentication (Figure 17.37). It is based on verification of the password attached with a user. Passwords are sent unencrypted and can be easily intercepted and therefore the protocol is not secure.

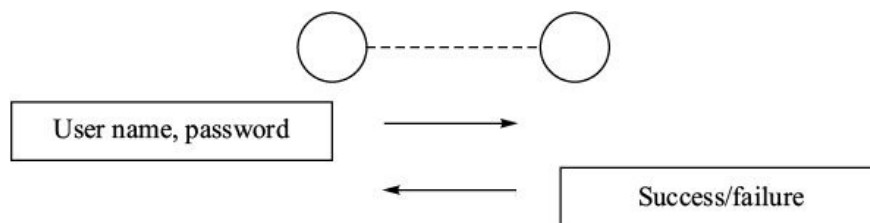


Figure 17.37 Two-way handshake in PAP.

17.13.2 CHAP

Challenge Handshake Authentication Protocol (CHAP) differs from PAP in several ways. It is based on three-way handshake (Figure 17.38). Authentication can be carried out by both the ends independently. The probability of interception and deciphering the password (called *Common secret*) is very low.

Each end knows a ‘common secret’ but the secret is never transmitted on the

link. The authenticating end chooses a random number and sends to the other end. The other end uses an algorithm on the random number and the common secret to generate a one-way hash value, and sends it back. The authenticating end computes the hash value independently and compares the result with the received value. If both are same, the authentication is successful. The hash

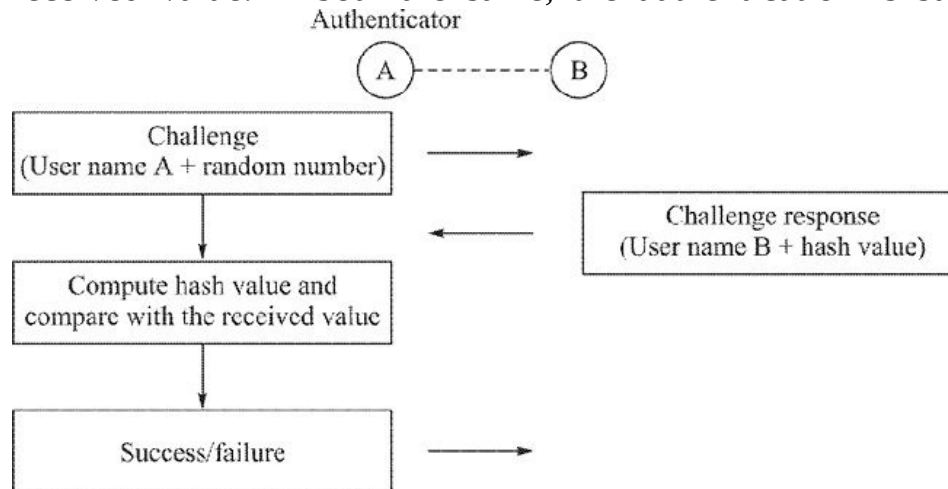


Figure 17.38 Three-way handshake in CHAP.

algorithm is such that the random number and the secret cannot be computed from the hash value if intercepted. Commonly used hash function is Message Digest 5 (MD-5) which yields 16 octets long output.

17.14 NETWORK CONTROL PROTOCOL (NCP)

Network Control Protocol (NCP) is used for configuring the parameters associated with the network layer protocol. The type of NCP being used is indicated in the protocol field of the PPP frame. Some of the important NCPs supported by PPP were listed in Table 17.6. The NCP required for IP is IP Control Protocol (IPCP).

Format of NCP packet contained in the information field is the same as that for LCP (Figure 17.36). NCP uses seven packet types (configure-request, configure-ack, configure-nak, configure-rej, terminate-request, terminate-ack, code-rej). The configuration options field also has the same format. The options depend on the network protocol.

For IP, the examples of options are IP address and compression protocol. IP address option is for allotment of IP address. The sender can send a particular IP address in configure-request packet. The receiving end can accept the proposed address by configure-ack or reject it by sending configure-nak and the allotted IP

address. Alternatively the sender can request for allotment of an IP address by sending IP address equal to 0.0.0.0 in its configure-request. The receiving end replies with configure-nak and allotted IP address. The configuration option type for IP address is 3. IP-compression option allows compression of IP header and/or user data. If this option is not negotiated, there is no compression. For further details on IPCP, the reader can go through RFC1332.

SUMMARY

The Internet Protocol (IP) is a network layer protocol that provides best effort, unreliable, and connectionless delivery of datagrams. It supports variety of layer 2 protocols (IEEE 802.x, Ethernet-DIX, PPP, HDLC, ATM, Frame relay). It provides service to layer 4 protocols, TCP, and UDP.

IPv4, the current version of IP, uses 32-bit addresses, each address consisting of network part and host part. In order to support networks of various sizes, the address space divided into several classes which are defined based on the size of network part. The address classification scheme has constraint of available number of network addresses. The problem is overcome by subnetting. Variable Length Subnet Mask (VLSM) enables creation of subnets of unequal sizes. IPv6 is the new version of the Internet Protocol. It has much bigger address space than that of IPv4. It provides for 128-bit addressing scheme against 32-bit addressing scheme IPv4.

Datagram approach for switching packets as used in IP is robust from the network point of view but it lacks features that can guarantee quality of service. It is possible to engineer an IP network to offer 'differentiated service'. In differentiated service, some traffic is treated better than the rest (separate queues, priority, lower packet discard rate, etc.). The preference accorded is statistical but not guaranteed.

There are several supporting protocol that work with IP. ARP is used for determining layer 2 address for a given layer 3 address. RARP is used for determining layer 3 for a given layer 2 address. ICMP is used for reporting errors and other messages. Point-to-Point Protocol (PPP) is used layer 2 data link protocol for transporting IP packets over point-to-point serial physical links. PPP is based on HDLC protocol and can be used between two routers or between a host and a router.

EXERCISES

1. An IP packet of total length 1500 octets is encountered by a data link that has an MTU³ size of 512 octets. If the IP header size is 20 octets, indicate the following fields of various fragments of the IP packet:
 - ◆ MF bit
 - ◆ Offset field
 - ◆ Total length
 - ◆ Data field octets.Assume LLC overhead of 4 octets.
2. Suppose a TCP message contains 2048 octets of data and 20 octets of TCP header. The message is passed through two IP networks N1 and N2. N1 has an MTU size of 1024 octets and N2 has an MTU size of 512 octets. Give the sizes and offsets of the sequence of fragments delivered to the destination. Assume IP headers to be 20 octets long.
3. Path MTU is the smallest MTU of the path from a source to the destination. If the path MTU in Exercise 2 is 512 octets and is used for both N1 and N2, give the sizes and offsets of the sequence of packets delivered to the destination.
4. Express the following IP addresses in binary form and identify their address classes:
 - (a) 200.42.129.16
 - (b) 145.32.59.22
 - (c) 14.82.19.52
5. Write the six subnets of 172.27.0.0/16. Write the subnet mask in the binary form.
6. What is the maximum number of hosts that can be supported by 193.1.1.64/27? What is the maximum number of hosts if it is further subnetted into six subnets?
7. Assume that an organization has been assigned the 196.35.1.0/24 network address. The organization decides to create subnets that will support at least 20 hosts.
 - (a) Specify the length of subnet address that will allow creation of at least 20 hosts on each subnet.
 - (b) What is the maximum number of hosts that can be supported on one subnet?
 - (c) What is the maximum number of subnets?
 - (d) Write the subnet address in dotted decimal notation.

- (e) What is the broadcast address of subnet 196.35.1.192?
8. ARP cache entries timeout after 5 minutes. Describe the problems that can occur if the timeout value is too large or too small.
 9. Suppose a router has built up the routing table as shown in Table E 17.7. Indicate what the router does when it receives IP packets with the following destination addresses.
 - (a) 128.96.39.10
 - (b) 128.96.40.12
 - (c) 128.96.40.151
 - (d) 192.4.153.17
 - (e) 192.4.153.90

TABLE E 17.7

Subnet	Subnet mask	Interface
128.96.39.0	255.255.255.128	a
128.96.39.128	255.255.255.128	b
128.96.40.0	255.255.255.128	c
192.4.153.0	255.255.255.192	d
Default		e

10. Which of the following IPv6 address notations are correct?
 - (a) :: 0F42:6270:AB00:67DB:BB27:7222
 - (b) 6705:42F3: :: 78F2:B75C:D4BA:12CC
 - (c) 74DC: : 02BA
11. Table E 17.8 shows part of a hypothetical routing table. The router makes the 'longest match' for determining the outgoing port for a received IP packet. Determine the outgoing (O/G)ports, if the IP packets with the following destination addresses are forwarded by the router:
 - (a) 11000100.01001011.00110001.00101110
 - (b) C4.5E.05.09 (Hexadecimal)
 - (c) 11000100.01001101.00110001.00101110
 - (d) C4.5E.03.87 (Hexadecimal)
 - (e) C4.5E.7F.12 (Hexadecimal)
 - (f) C4.5E.D1.02 (Hexadecimal)

TABLE E 17.8

Network Prefix	O/G port
11000100.01011110.00000010.00000000/23	a
11000100.01011110.00000100.00000000/22	b
11000100.01011110.11000000.00000000/19	c
11000100.01011110.00101000.00000000/18	d
11000100.01001100.00000000.00000000/14	e
11000000.00000000.00000000.00000000/2	f
10000000.00000000.00000000.00000000/1	g

12. An IP packet using strict source routing option is to be fragmented. Should the option be copied on each fragment or not?
13. Suppose instead of 16 bits for the network part of class B address, 20 bits are used. How many class B networks can be there?
14. Convert IP address C4.5E.4F.12 into dotted decimal notation.
15. A network has a subnet mask of 255.255.240.0. What is the maximum number of hosts it can support?
16. Three subnets have the following network prefixes:
C42.18.00.00/21 (Hexadecimal)
C4.18.08.00/22 (Hexadecimal)
C4.18.10.00/20 (Hexadecimal)
If these network prefixes are aggregated into a single route, what will be the aggregated network prefix and the mask?

[1](#) Multicasting is sending a datagram to a group of IP addresses. We study multicasting in Chapter 19.

[2](#) Refer to loopback testing at physical layer in Chapter 4.

[3](#) MTU (Maximum transmission unit) is the maximum size of IP packet a data link frame can carry.

18

Routing Protocols

We introduced the concept of routing in Chapter 15. The next two chapters, Chapters 16 and 17, described how the data packets are transferred across a data network using forwarding tables. In this chapter, we build on the knowledge gained so far and describe routing protocols that enable creation and maintenance of the forwarding tables. Routing is the most complex job carried out by a switched data network. It is for this reason that we postponed the detailed discussion on routing to this chapter.

Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) are the two most widely deployed routing protocols. These protocols are described in detail in this chapter. The underlying concepts of distance vectors, link state routing, Dijkstra's algorithm for the shortest path precede the discussion on the specific routing protocol. Before we close the chapter, we discuss briefly ISO's IS-IS routing protocol and Border Gateway Protocol (BGP). IS-IS protocol is presented as a comparative study with OSPF.

18.1 ROUTING

An IP datagram is routed hop-by-hop across a network to the destination using forwarding tables that are stored in the routers in advance. A forwarding table in a router contains destination network prefix, next hop associations (Figure 18.1). When an IP packet is received by a router, its Destination Address (DA) is matched with one of the entries in the forwarding table and the IP packet is forwarded through the interface indicated in the forwarding table.

The entire process described above is the forwarding or packet switching process. Routing that we describe in this chapter is the process of creating and maintaining the forwarding tables.

The routing instance of a forwarding table (or simply a route) can be created

manually. Such routes are called static routes. A forwarding table can have all static routes in it. The network administrator updates the static routes as and when required.

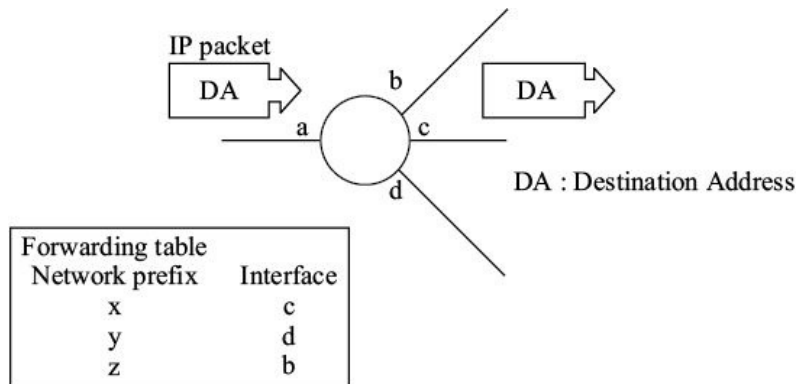


Figure 18.1 Forwarding table.

Creation and maintenance of forwarding tables based on static routes can be labour intensive in a complex network. If link or router goes down, which does happen often, the forwarding tables must be manually reconfigured immediately to define alternative paths. Dynamic routing overcomes this limitation. In dynamic routing, routers exchange their network topology information and then use an algorithm to arrive at the optimal paths to various destinations of a network. The output of the algorithm is depicted as the forwarding table. The protocol used for exchanging the network information is called routing protocol. A routing protocol defines:

- the formats of the topology information packets exchanged by the routers,
- the procedures adopted for their exchange, and
- algorithm for determining the optimal paths to the destinations.

Some of the common routing protocols used in the IP networks are listed below.

RIPv1	Routing Information Protocol, version 1
RIPv2	Routing Information Protocol, version 2
OSPF	Open Shortest Path First
I-IS-IS	Integrated Intermediate System to Intermediate System Border Gateway Protocol
BGP	Exterior Gateway Protocol
EGP	Interior Gateway Routing Protocol (Cisco)
IGRP	

18.1.1 Administrative and Routing Domains

In a global network, portions of the network are under different administrations. An administrative domain is the portion of the network owned by an administration (Figure 18.2). Each administration is free to decide the routing policies within its domain and can negotiate policies with the adjoining administrative domains. Routing policies decide to whom traffic for a destination will be sent and from whom traffic for a destination will be accepted.

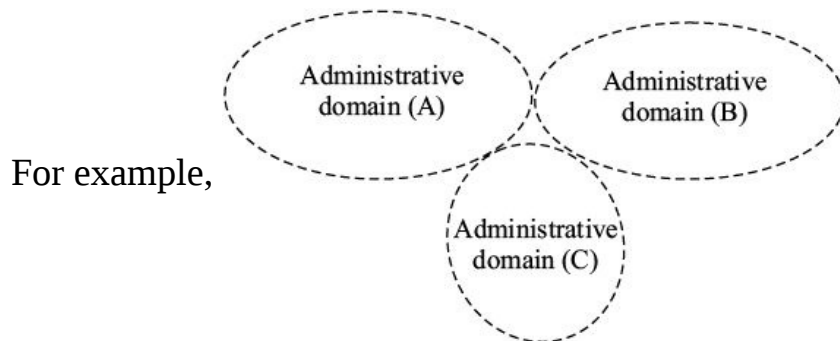


Figure 18.2 Administrative domain.

administrative domain B may allow use of its network by A for routing the traffic to a destination in B only. It may not allow A to route its traffic to C through its domain.

An administration may implement various routing protocols (e.g. RIP, OSPF, I-IS-IS) within its domain. The portion of an administrative domain that implements a particular routing protocol is called routing domain (Figure 18.3).

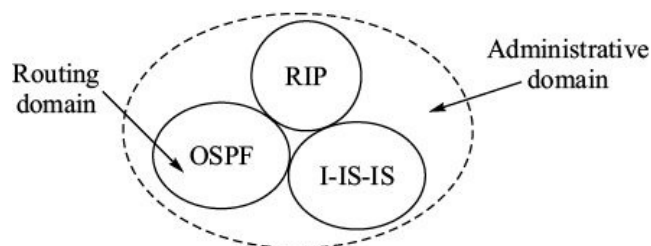


Figure 18.3 Routing domain.

18.2 STATIC ROUTING

Static routing is used in simple networks that lack redundancy, *e.g.* a stub network (Figure 18.4). 172.9.0.0 is a stub network. All the packets to and from it are routed through router R3. Therefore, the forwarding table of R4 contains three static routes, all going through R3.

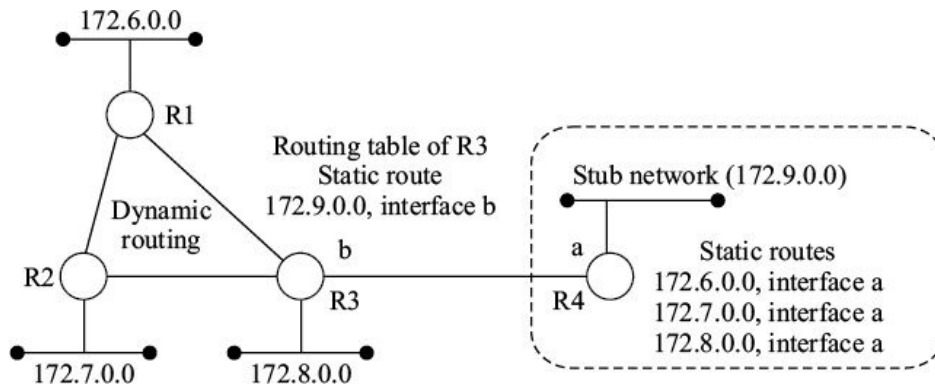


Figure 18.4 Static routes.

Forwarding table of R3 has one static route at its interface ‘b’ that connects to the stub network.

Static routes are at times created on security considerations or as a temporary measure when a special point-to-point route is to be created.

EXAMPLE 18.1 The following figure shows a hub and spoke network topology. The spokes communicate only with hub and not amongst themselves. Write the static routes of each router.

Solution Each spoke communicates only with the hub (172.6.0.0). Therefore, its associated router has only one static route entry.

Network prefix 172.6.0.0 Interface a The hub communicates with the three spokes and therefore router R1 has three static routes.

Network prefix 172.7.0.0 Interface b Network prefix 172.8.0.0 Interface c Network prefix 172.9.0.0 Interface d

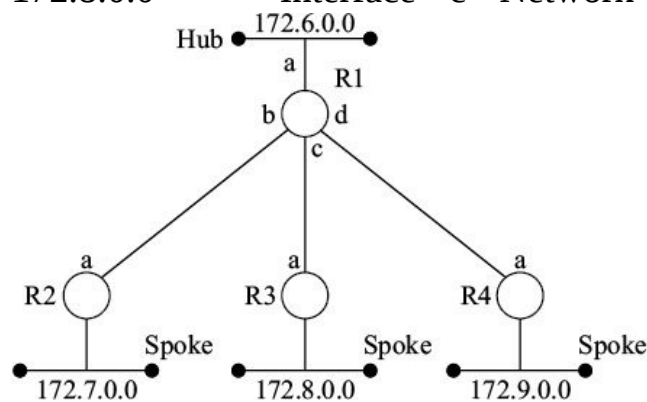


Figure 18.5 Example 18.1.

18.2.1 Default Routes

Routers discard an IP packet with unknown destination network address, *i.e.* address that is not available in the forwarding table. It is possible that some other

router may have route to the specified destination address. Therefore, a default route is defined. All the IP packets with unknown destination addresses are forwarded on the default route. Network prefix 0.0.0.0 is for the default route. Default route generally points towards core of an internetwork.

The default routes permit routers to store less than the full forwarding table as some of the destination networks can be routed through the default route. For example, the forwarding table of R4 in Figure 18.4 can have only one default route entry 0.0.0.0 at interface 'a' instead of three static routes. All the outgoing packets use the default route to R3 which decides the next hop for them depending on the destination address. Another benefit of default route is that if a new network is added, the forwarding tables of all the routers need not be updated. The default route in the routers will enable access to that network also.

18.3 DYNAMIC ROUTING

In dynamic routing, the forwarding tables are continuously updated with the information received from other routers. The routers exchange this information using a routing protocol. A routing protocol enables determination of the internetwork topology and creation of forwarding tables that indicate the best paths to the destinations.

For determination of best path, each link that interconnects two routers is assigned a cost. Path cost is sum of costs of all the links that constitute the path. The metric for cost can be in terms of number of hops, delay, inverse of bandwidth, *etc.* If number of hops is chosen as cost parameter, the minimum cost path would mean minimum number of hops. If delay is used as parameter, the packet will take the shortest path to the destination. If inverse of bandwidth is chosen as cost parameter, the packets will take paths that have larger bandwidths.

The routing protocols are based on one of the following two algorithms:

- Distance vector algorithm
- Link state algorithm.

Both the algorithms enable the routers to find global routing information, *i.e.* next hop to reach every destination by the optimal path. To accomplish this, the routers exchange routing information with other routers. In distance vector

algorithm, a router exchanges this information with its neighbours only. In link state algorithm, a router exchanges this information with every other router in the internetwork.¹

18.4 DISTANCE VECTOR ROUTING ALGORITHM

Distance vector is a subset of the forwarding table of a router. It contains two columns—destination and its distance from the router. All the destinations known to the router are listed in the distance vector. Here ‘distance’ is a routing metric that can have different types of meanings but it is usually the hop count. Distance vector routing algorithm aims to determine minimum-hop paths to various destinations. The algorithm works as follows:

- Each router periodically sends a copy of its distance vector to all its neighbouring routers.
- When a router receives a distance vector from its neighbour, it
 - updates its forwarding table with new destinations declared in the received distance vector,
 - determines whether the distance to reach an existing destination would decrease if it routed the packets to the destination through that neighbour. This is accomplished simply adding distance of its neighbour to the distance vector received from the neighbour, and comparing the result with its current records in the forwarding table.
- The rules for updating the existing record of the forwarding table are as follows:
 - If the distance to an existing destination is reduced, the existing route is replaced in the forwarding table.
 - If the distance to an existing destination is same as in the received vector, the existing route is retained.
 - If the distance to an existing destination has increased as inferred from the received vector, the route is replaced only if the router that sent the distance vector is the next hop router in the forwarding table. In other words, only the router that resulted in creation of a record in the forwarding table can increase the distance subsequently.

Let us illustrate the process with an example. In Figure 18.6, the three routers A, B, and C are directly connected to local networks having prefixes 172.16.0.0 (Net-1), 172.17.0.0 (Net-2), and 172.18.0.0 (Net-3).

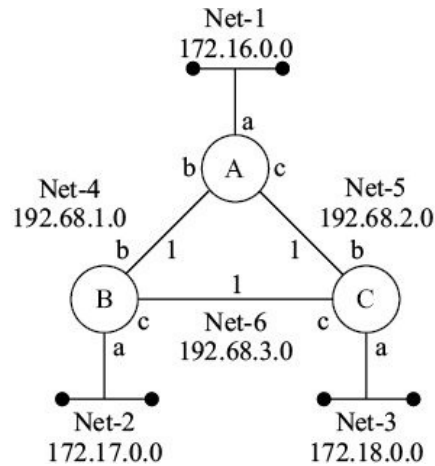


Figure 18.6 Internetwork for illustration of distance vector algorithm.

The routers are interconnected using point-to-point links having network prefixes as shown. We will refer all these networks by their names (Net-1, Net-2, etc.) rather than by their IP addresses to make the reading easy.

We assume that the distance metric is number of hops. The initial forwarding tables of the three routers when the routers are booted up are shown in Figure 18.7. Each table contains only three records that pertain to the three configured networks. The hop distance to the directly connected networks is equal to 0. The distance vector of the router A derived from its forwarding table is [Net-1,0 Net-4,0 Net-5,0]. It contains network IP address and the distance to the network from A.

Let us assume that router A sends its distance vector to its neighbours B and C.

- When B receives this vector, it increments the distances indicated therein by one hop and then compare the result with its existing records. Net-1 and Net-5 are new networks and therefore these networks get added in the forwarding table.
- Net-4 was already available in the forwarding table with distance 0. Therefore, B retains the original entry as it has shorter distance.
- Router C also takes similar steps when it receives the distance vector from A.

The forwarding tables of the three routers at this stage are shown in Figure 18.7b. Let us assume that after updating its forwarding table, router B sends its distance vector [Net-2,0 Net-4,0 Net-6,0 Net-1,1 Net-5,1] to routers A and C.

- When C receives this vector, it first adds one hop to the distances indicated in the vector.

[Net-2,1 Net-4,1 Net-6,1 Net-1,2 Net-5,2]

- It compares the result with the existing records in its forwarding table. Net-2 is a new network and therefore it is added to the forwarding table.
- Existing routes to Net-1, Net-4, Net-5 and Net-6 are shorter or same and therefore are retained.

Forwarding Table (A)			Forwarding Table (B)			Forwarding Table (C)		
Network	Distance	Interface	Network	Distance	Interface	Network	Distance	Interface
Net-1	0	a	Net-2	0	a	Net-3	0	a
Net-4	0	b	Net-4	0	b	Net-5	0	b
Net-5	0	c	Net-6	0	c	Net-6	0	c

(a) Initial Forwarding Tables

Forwarding Table (A)			Forwarding Table (B)			Forwarding Table (C)		
Network	Distance	Interface	Network	Distance	Interface	Network	Distance	Interface
Net-1	0	a	Net-2	0	a	Net-3	0	a
Net-4	0	b	Net-4	0	b	Net-5	0	b
Net-5	0	c	Net-6	0	c	Net-6	0	c
			Net-1	1	b	Net-1	1	b
			Net-5	1	b	Net-4	1	b

(b) Forwarding Tables after Receipt of Distance Vector from A

Forwarding Table (A)			Forwarding Table (B)			Forwarding Table (C)		
Network	Distance	Interface	Network	Distance	Interface	Network	Distance	Interface
Net-1	0	a	Net-2	0	a	Net-3	0	a
Net-4	0	b	Net-4	0	b	Net-5	0	b
Net-5	0	c	Net-6	0	c	Net-6	0	c
Net-2	1	b	Net-1	1	b	Net-1	1	b
Net-6	1	b	Net-5	1	b	Net-4	1	b
						Net-2	1	c

(c) Forwarding Tables after Receipt of Distance Vector from B

Forwarding Table (A)			Forwarding Table (B)			Forwarding Table (C)		
Network	Distance	Interface	Network	Distance	Interface	Network	Distance	Interface
Net-1	0	a	Net-2	0	a	Net-3	0	a
Net-4	0	b	Net-4	0	b	Net-5	0	b
Net-5	0	c	Net-6	0	c	Net-6	0	c
Net-2	1	b	Net-1	1	b	Net-1	1	b
Net-6	1	b	Net-5	1	b	Net-4	1	b
Net-3	1	c	Net-3	1	c	Net-2	1	c

(d) Forwarding Tables after Receipt of Distance Vector from C

Figure 18.7 Forwarding tables creation.

- Router A updates its forwarding tables in similar manner. The resulting forwarding tables at this stage are shown in Figure 18.6c.

The forwarding tables reach steady state after router C sends its distance vector [Net-3,0 Net-5,0 Net-6,0 Net-1,1 Net-4,1 Net-2,1] to routers A and B (Figure 18.7d).

The distance vector algorithm is also called ‘Distributed Bellman-Ford Algorithm’ after its creators. It is used in Routing Information Protocol (RIP) of IP networks.

18.4.1 Slow Convergence to Steady State

It can be shown that even if the routers update their forwarding tables asynchronously, the forwarding tables will eventually converge. To maintain the integrity of the forwarding tables, the routers send their distance vectors to their respective neighbours at regular intervals (30 seconds in RIP). Apart from the scheduled updates, a router sends triggered updates of distance vector as and when a change in the internetwork occurs.

In distance vector algorithm, a router is dependent on the upstream router to perform its distance calculations first. For example, if Net-1 goes down (Figure 18.8), router A will communicate this change to B through its distance vector. B will work out its forwarding table and then send its distance vector to router C. Therefore, the convergence time is inherently large in distance vector algorithm. As we will see later, a change is broadcast to all the routers simultaneously in the link state routing algorithm, and each router works out its forwarding table independent of the others. Thus convergence is faster in link state routing algorithm.

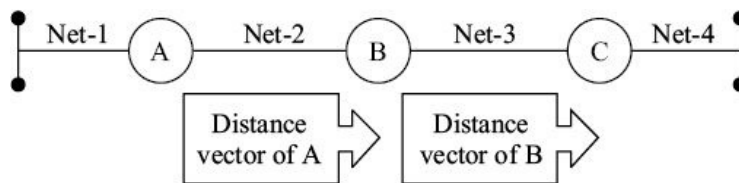


Figure 18.8 Convergence delay in RIP.

Until the forwarding tables reach the steady state since, the internetwork is in a transient state and there can be inconsistencies in the forwarding tables. These inconsistencies may result in ‘routing loops’. An IP packet is in routing loop when it comes back to the same router which forwarded it. IP packets going round in loops are ultimately dropped after expiry of Time-to-Live (TTL) but they generate unproductive traffic.

18.4.2 Count-to-Infinity

Distance vector algorithm does not work well if there are changes in the internetwork. It is primarily due to the fact that the distance vector sent to the neighbours does not contain sufficient information about the topology of the internetwork.

Consider a simple internetwork consisting of two routers, A and B (Figure 18.9). The steady state status of the distance vectors is as shown. Suppose that Net-3 goes down. Before B could dispatch its new distance vector to A with cost

to Net-3 as , it receives a distance vector from A indicating that Net-3 is 1 hop away from router A. A does not tell B in its update that

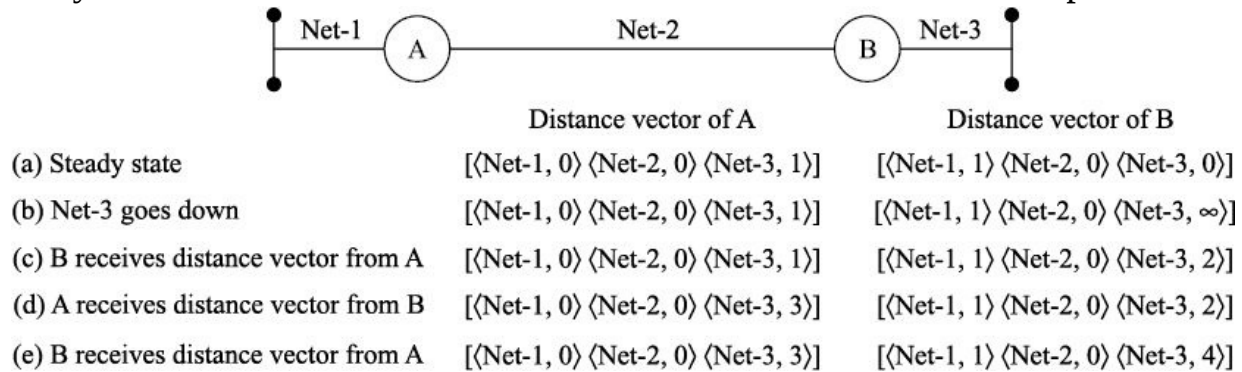


Figure 18.9 Count-to-infinity problem.

the path to Net-3 is via B. Equipped with this incomplete information, B sends its new distance vector to A indicating that Net-3 is 2 hops away from B. It does not indicate distance to Net-3 as , as it has found a new shorter path to Net-3 via A. When A receives this vector, it revises its distance to Net-3 via B to 3 units.

This exchange of vectors goes on and each time distance to Net-3 is incremented by one unit. When the distance reaches infinity, both the routers realize that there is no route to Net-3. During the process of counting-to-infinity, data packets from A or B to the destination Net-3 shuttle between A and B, causing congestion for every one else.

Count-to-infinity situation is handled in several ways. Split horizon, maximum distance limit, hold-down timer, path vector are some of the procedures for counteracting this problem. A routing protocol may implement a combination of these methods.

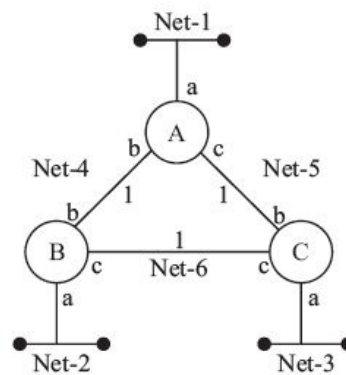
18.4.3 Split Horizon

In split horizon method, a router does not advertise the distance of a destination to the neighbour through which it learnt the route to the destination. In simple terms, it means that a route is not to be advertised to a neighbour if the route passes through the neighbour. For example, in Figure 18.9, router A does not advertise distance of Net-3 to router B because the route to Net-3 passes through B. When Net-3 goes down, B does not receive any information about Net-3 from A, and therefore the count-to-infinity problem is resolved.

It can be shown that split horizon solution for count-to-infinity problem works in case of two adjacent routers only. It is ineffective when more than two routers do so. Figure 18.10 shows an internetwork with three routers connected in a

loop. Suppose Net-1 goes down. Router A indicates infinite distance to Net-1 in its distance vector sent to B and C. Before this information could reach B, let us assume, B had already sent its distance vector to router C indicating 1 hop distance to Net-1. On receipt of these two distance vectors, Router C finds that Net-1 is no longer available through A but B offers a path with 2 hops. It updates its routing table accordingly. It also intimates A that Net-1 is available through it with a distance of 2 hops. Thus A finds a new path to Net-1 at a distance of 3 hops through C. It communicates the same to B and the count-to-infinity starts.

Maximum distance limit. As seen in the above example, split horizon does not resolve the count-to-infinity problem when there are loops in the Internet. One



alternative is to limit infinity

Figure 18.10 Count-to-infinity problem in networks with physical loops.

by defining a maximum distance limit. When the distance to a network reaches the specified limit, the network is declared unreachable. RIP uses maximum distance limit of 16. Note that maximum distance does not resolve the count-to-infinity problem but it merely reduces its impact on the performance.

Split horizon with poison reverse. This is a variant of split horizon and is used in RIP. In this case, a router indicates infinite distance of a destination to its neighbour through which it learnt the route to the destination. In RIP, distance of 16 hops is taken as the infinite distance. For example, in Figure 18.9, router A would indicate infinite distance to reach Net-3 in its distance vector. Thus B would know that Net-3 is not accessible through A. Note that in plain split horizon, A would not tell B of a path to Net-3 at all.

Split horizon with poison reverse also does not prevent count-to-infinity when there are physical loops. But the distance vector algorithm in this case converges faster than split horizon.

18.4.4 Hold-Down Timer

A tool for preventing most instances of count-to-infinity is hold-down timer. When a distance vector received from a neighbour increases the distance to a destination, the receiving router starts a hold-down timer. Until the timer expires, the router does not accept any other update on that destination unless the update is equal to or less than original path distance. The hold-down timer is set at 240 seconds typically.

In Figure 18.10, after router C receives update from A that Net-1 is inaccessible (infinite distance), it does not accept update from router B in respect of Net-1 because the update from B increases the original path distance of 1 hop to 2 hops. By the time the hold down timer expires, C receives next update from B, indicating infinite distance to Net-1. Thus the count to infinity does not take place, but at the cost of some increase in convergence time.

18.4.5 Path Vector

In Figure 18.9, the reason for count-to-infinity was that when B updated its new path to Net-3 via A, it had no knowledge that A's path to Net-3 was through B. Had A sent this information also, B would have immediately realized that no path to Net-3 existed through A.

Path vector is similar to distance vector with the additional information about the path. For example, in Figure 18.9, the path vector of A will be [Net-1, 0, Local Net-2, 0, Local Net-3, 1, B].

Path vector approach is used in Border Gateway Protocol (BGP). The problem with path vector approach is that the vectors are of larger size because an additional parameter has been added. When the number of nodes is large, addition of even one extra parameter to the vector proves expensive in terms of the traffic generated and memory required for storage of the vector.

18.5 ROUTING INFORMATION PROTOCOL (RIP)

Routing Information Protocol (RIP) is a distance vector routing protocol. It was formally defined in two documents RFC 1058 and STD 56. As IP-based networks became numerous and bigger, its limited capabilities became apparent. It was superseded by its second version RIPv2 in 1994 (RFC 1723) and the first version became historic. Nevertheless the first version of RIP (referred to as RIPv1) is still widely deployed routing protocol. The basic features of RIPv1 are as follows:

- RIPv1 is a distance vector routing protocol.

- Distance vector contains all destinations of a forwarding table and their distances from the source router.
- A router sends the distance vector updates to all its neighbours. The distance vector updates are sent at intervals of 30 seconds.
- The entries in the forwarding table live up to 180 seconds after the last update was heard.
- The metric used is number of hops. Maximum number of hops can be 15. Distance equal to 16 is equivalent to infinite distance and indicates that the network is not reachable.
- Split horizon with poison reverse is used to overcome count-to-infinity problem and to reduce the convergence time.
- Triggered updates are used to spread the network changes to all the routers as when they occur.

18.5.1 Format of RIPv1 Packet

Figure 18.11 shows the format of RIP packet. It is encapsulated as UDP message and handed over to the IP layer. The IP layer adds the IP header to the UDP packet. The IP header contains the source IP address. The router then sends the IP packet to its neighbours. Note that:

- Some of the fields of the RIP packet are all zeroes. These have been included to maintain backward compatibility with the pre-standard varieties of RIP.
- RIP header consists of command and version fields.
- Distance vector is included as a series of route objects. Each route object consists of AFI, IP address of destination, and distance metric fields.
- There can be maximum 25 route objects in one RIP packet.

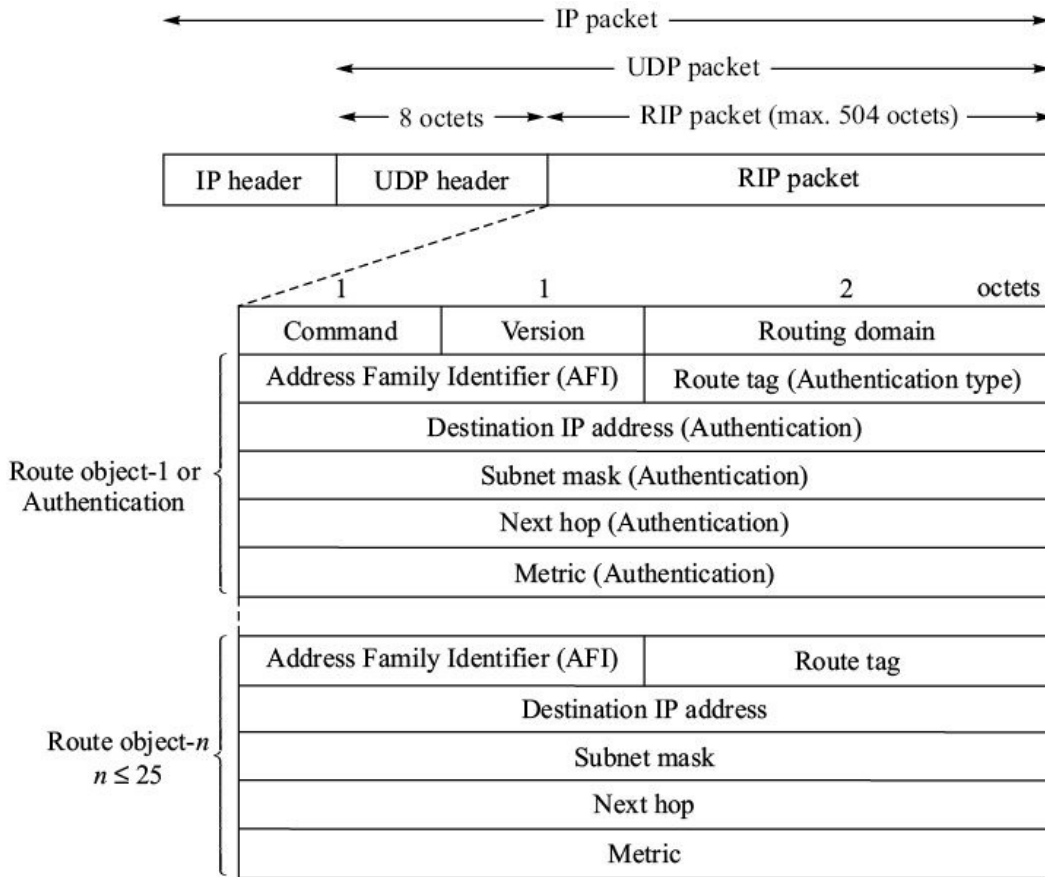


Figure 18.11 Format of RIPv1 packet.

Various fields of RIP packet are as under:

Command (1 octet). This field indicates whether the RIP packet is a request (0x01) or response (0x02).

Version (1 octet). Specifies the RIP version which is 0x01 for RIPv1.

Zero fields. These are not used.

Address family identifier (AFI, 2 octets). RIP is designed to carry destination information for several different network protocols. Each route has an AFI field to indicate the type of address being specified. For an IP address, AFI is equal to 2.

IP address (4 octets). Contains the IP address of the destination network.

Metric (4 octets). Indicates the number of hops to the destination. The metric has value between 1 to 16. The value of 16 indicates an unreachable destination.

An RIP packet can contain at the most 25 route objects. Each route object is 20 octets long. Therefore, the maximum size of RIP packet is 504 octets

including 4-octet header. The UDP header is 8 octets and maximum size of IP header is 60 octets. Thus, the maximum size of IP packet containing RIP packet is 572 octets which is less than the minimum MTU² size of 576 octets. Therefore, it passes through all the links without fragmentation.

18.5.2 Types of RIP Packets

There are two types of RIP packets:

- Request packets
- Response packet.

When a router boots up, it sends its distance vector containing local networks to its neighbours and requests them for sending their distance vectors using the request packet. The neighbours send their distance vectors to the requesting router using response packets.

The response packets can also be sent unsolicited. Unsolicited response packets are sent by the routers for sending scheduled and triggered updates of their distance vectors to their neighbours.

- Scheduled updates are sent at regular intervals of 30 seconds.
- Triggered updates are sent when a network change occurs.

When a router receives an update containing distance vector from a neighbour, it updates its forwarding table and sends an update containing its own distance vector to all its neighbours. The updates serve two purposes:

- Sharing the network information that enables reworking of the forwarding tables
- Resetting the time stamps of records of the forwarding table.

18.5.3 Forwarding Table

RIPv1 forwarding table contains the following fields (Table 18.1): **Net-Id.** Network part of the IP address of the destination network.

Next hop. The IP address of the next router. This field is the source address in the header of IP packet that contains RIP packet.

Interface. The physical interface to the next hop.

Distance. Distance metric in terms of number of hops.

Time stamp. Time stamp of the last update for this entry.

TABLE 18.1 A Typical Forwarding Table of RIPv1

Net-Id	Next hop	Interface	Distance	Time stamp
	193.23.5.20			T1
	148.12.77.5			T2
96.0.0.0		a	5	*
126.0.0.0	—	b	3	
148.12.0.0		b	0	
193.23.5.0		a	0	*
0.0.0.0	—	a	1	
				**
	193.23.5.20			

Local networks * Default route **All these fields are derived from the IP packet and the RIP packet contained therein.**

- Net-Id and distance fields are derived from the RIP packet. Net-Id is the destination IP address of a route object. Distance is obtained by adding one additional hop to the received distance metric.
- IP address of the next hop is obtained from the source address field of the IP header.

Time stamp field is important, because if a record in the forwarding table is not refreshed for 180 seconds since its last update, its distance is increased to 16. Distance of 16 indicates that the Netid is unreachable. If no update for the entry is received for next 120 seconds, it is deleted from the forwarding table.

Updating forwarding table. When a router receives updates from its neighbours, it does the following:

1. If there are multiple updates that contain same Net-Id, it implies that there are multiple paths to the Net-Id. The update with shortest distance to a destination is stored in the forwarding table. Thus there is exactly one path to a destination.
2. If there is tie in path distance, the update received first is kept in the forwarding table.
3. If the update indicates longer distance to a destination than the existing

entry in the forwarding table, the update is taken into account only if it is from the existing next hop router. In other words, a longer path is added in the forwarding table only if the existing next hop router says so.

4. When an updated entry is entered in the forwarding table, the received distance metric is increased by 1.

18.5.4 Limitations of RIPv1

Some of the limitations of RIPv1 are as follows:

1. The redundant paths are not made use of for load sharing. There is only one path in the forwarding table for each target Net-Id.
2. The routing updates contain distance vector pertaining to the entire forwarding table. The updates are, therefore, voluminous. One RIP packet can contain 25 route objects only. Forwarding table of even a medium sized network will have many more records. Therefore, each update will consist of several RIP packets. The updates are sent every 30 seconds. All these factors result in considerable overhead of routing traffic.
3. There is no provision for subnet masks. Therefore, RIPv1 is a classful routing protocol, *i.e.* network part of the IP address is decided based on the class boundaries (8 bits for class A, 16 bits for class B, and 24 bits for class C).

18.5.5 Routing Information Protocol (Version 2)

Routing Information Protocol (Version 2), called RIPv2, is documented in RFC 1723 and it supercedes RIPv1. Basic protocol operation in RIPv2 and RIPv1 is same. RIPv2 supports some additional features as follows:

- RIPv2 is classless routing protocol. The RIPv2 packet contains additional field for subnet masks.
- RIPv2 supports authentication of routing updates.

18.5.6 Format of RIPv2 Packet

Formats of RIPv2 and RIPv1 packets are same in that they have same size and structure. But additional fields are introduced in RIPv2 in place of the zero fields of RIPv1. RIPv2 packet is also encapsulated in UDP and IP headers as in the case of RIPv1. Figure 18.12 shows the format of RIPv2 packet.

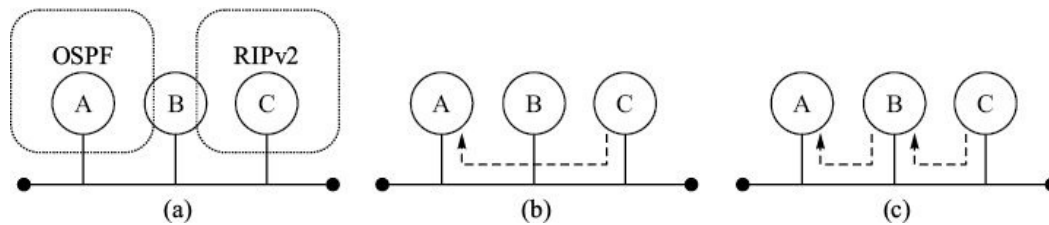


Figure 18.12 Format of RIPv2 packet.

Command (1 octet). This field indicates whether the RIP packet is a request (0x01) or response (0x02).

Version (1 octet). It specifies the RIP version which is 0x02 for RIPv2.

Routing domain (2 octets). A router can be part of several routing domains. This field specifies the routing process for which this update is meant.

Address family identifier (AFI, 2 octets). This field is same as in RIPv1 with one exception. If the first AFI in the RIP packet is 0xFFFF (all ones), then this message contains authentication. For IP addresses, AFI is equal to 2 (0x0002) as before.

Route tag. It provides a method for distinguishing between internal routes (within an autonomous system) and external routes (outside the autonomous system). For EGP and BGP routing protocols, this field contains the Autonomous System (AS) number.

IP address (4 octets). This field is the same as in RIPv1.

Subnet mask (4 octets). It contains the subnet mask to be applied to the IP address.

Next hop (4 octets). This field contains the IP address of the router where the destination network address is directly available. The next-hop field is described below.

Metric (4 octets). This field is the same as in RIPv1.

18.5.7 Next-Hop Field in RIPv2

The next-hop field enables a router to announce which networks can be reached through other routers directly. Consider the network shown in Figure 18.13a. Routers A, B, and C are on the same network. Router A supports OSPF routing protocol. Router B supports both OSPF and RIPv2 protocols and router C supports only RIPv2. Therefore, A and C do not directly communicate for the

purpose of routing, although they are neighbours.

When router B learns a set of destinations from router A, it sends these destinations to router C through RIPv2 packet, indicating in the next-hop field the IP address of router A. Thus router C learns the destinations of OSPF domain reachable via router A. When C has an IP packet to send to any of these destinations, it sends the packet directly to router A (Figure 18.13b). If the next hop attribute had not been used, router C would have considered B to be the next hop for these destinations. Router B would have forwarded them to router A (Figure 18.13c).

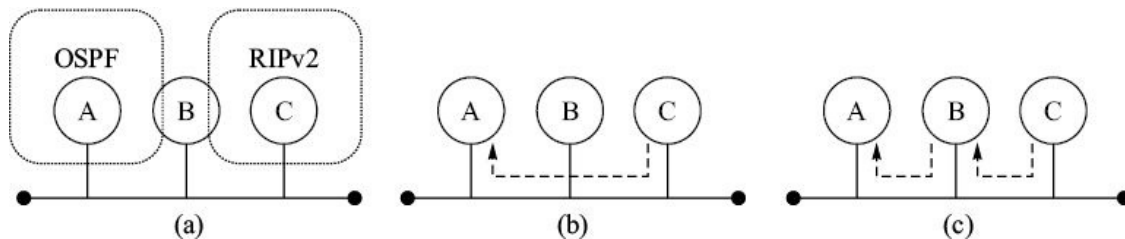


Figure 18.13 Use of the next-hop field of RIPv2 packet.

Not all the IP packets from router C would flow to router A. Those network addresses for which B is the best next hop, have next-hop field set to 0.0.0.0 in the RIPv2 packet. Router C would reach those destinations via router B.

18.5.8 Authentication³

Authentication is a new feature of RIPv2. If a routing update is received without valid authentication, it is ignored by the receiving router. Authentication data occupies what would be the place of the first route object (Figure 18.13). Thus if authentication is used, a RIPv2 packet will be left with space for maximum 24 route objects instead of normal 25. Authentication object is identified by AFI field which is 0xFFFF (all ones). The next two octets are of authentication-type field. The current version of RIPv2 supports only one type of authentication (Type 2), which is simple (clear text) password protection. RFC 2082 describes the method of using MD5 for authentication (Type 3).

18.6 LINK STATE ROUTING

In distance vector routing, a router calculates distances to every other router of the network based on the distance vectors it receives from its neighbours. Many problems of this method are due to the fact that every router tells its neighbours,

its distances to all the networks without knowing the network topology. This results in misleading conclusions, as we illustrated with count-to-infinity problem.

Link state routing overcomes this limitation of distance vector routing. In link state routing, a router tells every other router the information it truthfully knows, its neighbours and distances to them. Every router works out from this information (a) the network topology and (b) the optimal paths. Open Shortest Path First (OSPF) is the most widely deployed link state routing protocol.

18.6.1 Basic Operation

In link state routing, every router maintains a database of network topology. The database contains records of the links of the entire network. Each record consists of source router identifier, its neighbouring router identifier, and the cost associated with the link between them (Figure 18.14). Each record is called link state. The cost can be defined in terms of distance, hop, delay, inverse of bandwidth or any other parameter.

Identical database is available on all the routers. The database is refreshed at fixed intervals (30 minutes in OSPF). For refreshing the database, every router sends updates called Link State Advertisements (LSAs).

If there is a change in neighbourhood (e.g. a link/router goes down or new router is added), LSAs are sent immediately by the routers that detect the change. They do not wait for the regular schedule of advertisement for refreshing the records of the database. LSAs are sent using controlled flooding⁴ across the internet so that every router receives them.

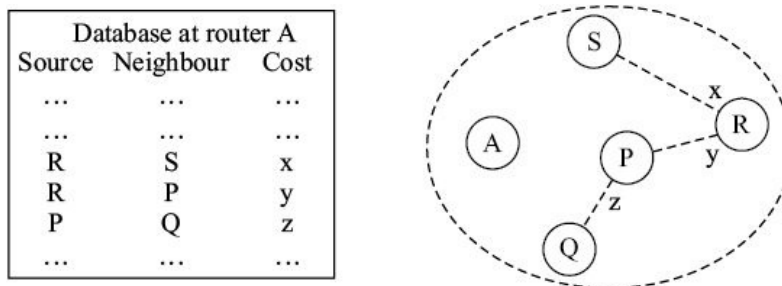


Figure 18.14 Link state routing.

Each router works out the shortest paths to every other router using the database and Dijkstra's algorithm. Once the shortest paths are known, the forwarding table can be constructed readily.

An advantage of link state routing is the availability of alternate paths. If a link

goes down, a router can readily work out alternative path from its topology database. This is not possible in distance vector routing because when a distance vector is received, a router discards all the alternative paths and it does not have network topology information to work out alternative paths on its own.

We will describe next the Dijkstra’s algorithm for computing shortest paths. The link state routing protocol for creating forwarding tables is described as part of OSPF protocol which we discuss after learning Dijkstra’s algorithm.

18.6.2 Dijkstra’s Algorithm

Dijkstra’s algorithm computes the shortest paths from a node (called root) to all other nodes from the link state database. The root node selects one of its neighbours having the least cost. The link costs of the neighbours of these two nodes are next examined. One of the neighbours having least cost to the root is selected again. The process is repeated, and each time a neighbour with the least cost to the root is selected and added to the set of nodes whose least costs have been computed (Figure 18.15).

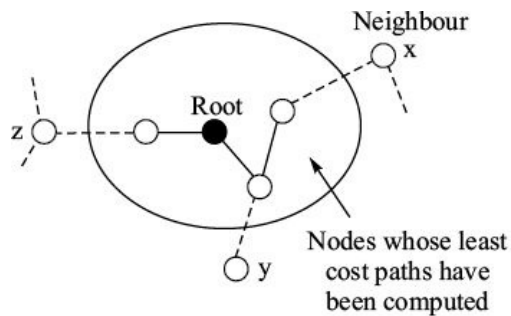


Figure 18.15 Selection of node with least path cost to the root.

To understand the algorithm, let us consider a simple a graph consisting of nodes A, B, C, D, and E (Figure 18.16). We assume that each node is aware of the link costs between any pair of interconnected nodes. Table 18.2 shows the link costs associated with each link of the graph.

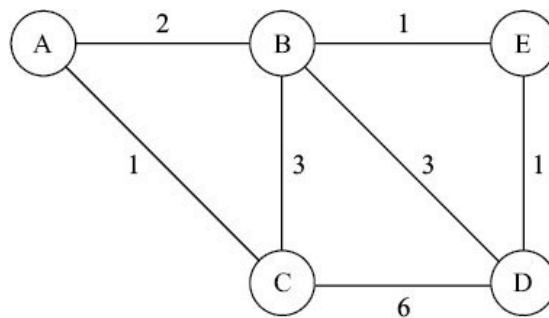


Figure 18.16 Dijkstra’s algorithm.

TABLE 18.2 Link Costs of Neighbouring Nodes

Nodes	A	B	C	D	E
A		2	1		
B	2		3	3	1
C	1	3		6	
D		3	6		1
E		1		1	

We will use Dijkstra's algorithm to determine the least cost paths from A to the rest of the nodes.

We will use the following terminology for describing Dijkstra's algorithm:

Root : The node from which the least cost paths are being determined.

Set (S) : Set of those nodes whose least cost paths to the root have been determined.

Set (N) : Set of neighbours of set S.

I (J, p) : Node I has path cost 'p' to the root via node J.

Since we are to determine the forwarding table of node A, A is the root. The flow chart of Dijkstra's algorithm is shown in Figure 18.17. Table 18.3 shows the steps of the Dijkstra's algorithm applied to A as the root.

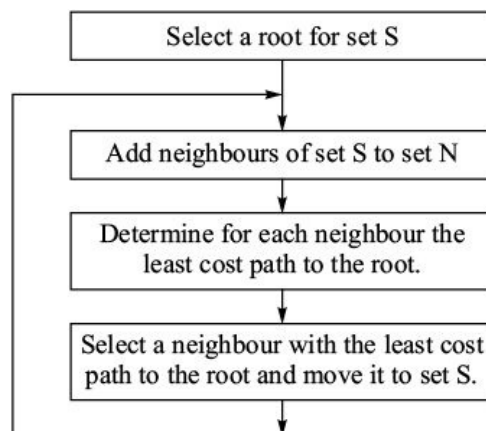


Figure 18.17 Flow chart for Dijkstra's algorithm.

TABLE 18.3 Least Cost Path Determination Using Dijkstra's Algorithm

Steps	Set S	Set N
1	AA,0	BA,2, CA,1
2	AA,0, CA,1	BA,2, DC,7
3	AA,0, CA,1, BA,2	DB,5, EB,3
4	AA,0, CA,1, BA,2, EB,3	DE,4
5	AA,0, CA,1, BA,2, EB,3, DE,4	

Step-1: The set S contains root A and set N contains B and C. C has lower cost to A than B.

Step-2: C moves into set S. Another node D, the neighbour of C, is added to set N. node B of set N has the least cost to the root A.

Step-3: B moves into set S. D now has lower cost path to the root A via node B. Another neighbouring node E is added to set N. node E of set N has the least cost to A.

Step-4: E moves into set S. D now has lower cost path to A via E and D is the only node in set N.

Step-5: D moves into set S. Set N is now empty.

With step-5, all the least cost paths to the root A have been determined. Figure 18.18 shows the resulting tree.

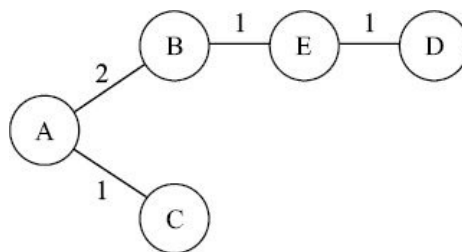


Figure 18.18 Shortest path tree of root node A.

Having understood Dijkstra's algorithm for determining the shortest paths from a node to other nodes of a graph, let us now apply this algorithm to an internetwork consisting of routers and networks as shown in Figure 18.19. Note that we have assigned different link costs to packet flows in opposite directions on the same link. For example, the cost of sending a packet from A to B is 4 and the cost of sending a packet from B to A is 1. This is quite possible in data networks.

We need to first transform this internetwork into a graph consisting of nodes and links. The ground rules for constructing the graph are as follows:

1. All networks and routers are represented as nodes. The networks are represented as square (□) nodes and the routers as circular (○) nodes.
2. Multi-access network (e.g. LAN) is also represented as a network node and it can have multiple router nodes connected to it.
3. Cost is associated with sending a packet from a router node to a network node. No cost is associated with sending a packet from the network node.
4. No cost is associated with receiving a packet.

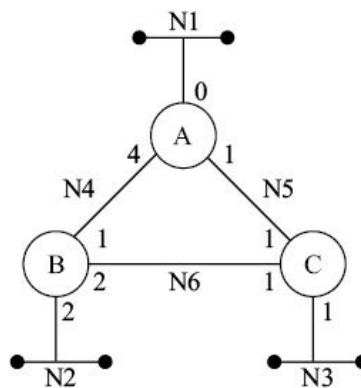


Figure 18.19 Internetwork for determining the shortest paths using Dijkstra's algorithm.

The graphical representation of the internetwork (Figure 18.19) is shown in Figure 18.20. Now we can apply Dijkstra's algorithm without differentiating the two types of nodes. It is to be remembered that (a) the cost of sending a packet from a network node is 0, and (b) it is the root node that sends packets towards various destinations.

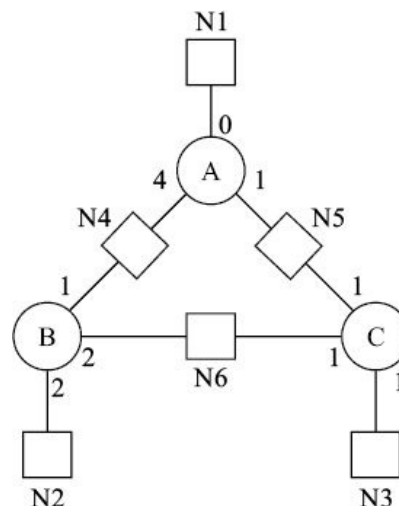


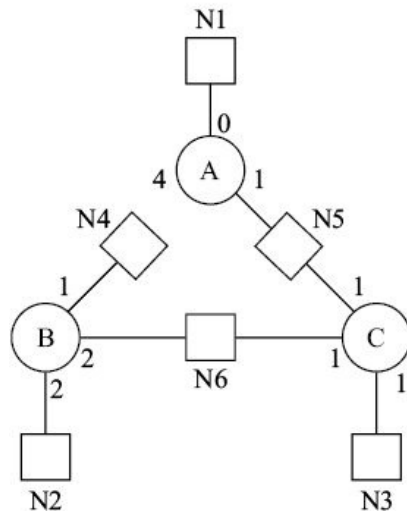
Figure 18.20 Graph of the internetwork shown in Figure 18.19.

Table 18.4 shows the various iterations of Dijkstra’s algorithm. Router node A is chosen as the root node. Figure 18.21 gives the resulting tree and the forwarding table constructed from the tree.

Each router of the network creates its forwarding table using its link state database independently. It is, therefore, essential that the link state database is identical across the internetwork. Otherwise the routers will create forwarding tables that are incompatible. Any average sized network is in state of continuous change. Some links or routers go down or

TABLE 18.4 Dijkstra’s Algorithm Applied to Figure 18.20 with Node A as the Root

Steps	Set S	Set N
1	AA,0	N1A,0, N4A,4, N5A,1
2	AA,0, N1A,0	N4A,4, N5A,1
3	AA,0, N1A,0, N5A,1	N4A,4, CN5,1
4	AA,0, N1A,0, N5A,1, CN5,1	N4A,4,N3C,2, N6C,2,
5	AA,0, N1A,0, N5A,1, CN5,1, N3C,2	N4A,4, N6C,2
6	AA,0, N1A,0, N5A,1, CN5,1, N3C,2, N6C,2	N4A,4, BN6,2
7	AA,0, N1A,0, N5A,1, CN5,1, N3C,2, N6C,2, BN6,2	N4B,3, N2B,4
8	AA,0, N1A,0, N5A,1, CN5,1, N3C,2, N6C,2, BN6,2, AA,0, N1A,0, N5A,1, CN5,1, N2B,4, N4B,3	
9	N3C,2, N6C,2, BN6,2,	N4B,3, N2B,4



Destination	Next hop	Cost
N1	Local	0
N2	N5	4
N3	N5	2
N4	N5	3
N5	Local	1
N6	N5	2
B	N5	2
C	N5	1

Figure 18.21 Forwarding table of router A and the shortest path tree of the graph in Figure 18.20.

new routers or new links are added. If these changes are not quickly communicated across the network, the link state databases become inconsistent soon.

18.7 OPEN SHORTEST PATH FIRST (OSPF) ROUTING

PROTOCOL

Open Shortest Path First (OSPF) routing protocol is the most widely deployed link state routing protocol in IP and ATM networks. We will discuss OSPF in the context of IP networks in this chapter. OSPF routing protocol consists of two basic mechanisms:

- Establishing and maintaining adjacencies with neighbours.
- Advertising the neighbourhood relationships to all the routers in a network.

By establishing adjacency, we mean that two neighbouring routers confirm their presence to each other and exchange their link state databases. By maintaining adjacency, we mean that a router periodically checks that the neighbour is still alive. OSPF carries out this function using 'Hello' protocol.

A router periodically advertises its link states relating to its neighbours to all the routers in the internetwork (OSPF specifies interval of 30 minutes). These advertisements enable maintenance of an identical link state database in all the routers. Each router independently works out the shortest paths to destination networks from the database using Dijkstra's algorithm and creates its forwarding table.

Before we go into details of OSPF routing protocol, we must understand terminology relating to hierarchical routing, which is described next.

18.7.1 Hierarchical Routing

If an internetwork consisting of N routers deploys link state routing, it can be shown that the algorithm for generating shortest paths will make order of $N \log N$ computations and the forwarding table of each router will have order of N records. The number of routers in the Internet is of the order of 1 million. Each router will have a forwarding table of 1 million records and the link state routing algorithm will make 6 million computations.

To resolve this problem, we need to partition a large internetwork into several domains and then carry out inter-domain and intra-domain routing. Each router is required to maintain database pertaining to the domain it belongs to. For example, suppose we create 1000 domains each having 1000 routers in the previous example of 1 million routers. If we allow a few routers in each domain to participate in inter-domain routing, the orders of computations required and size of forwarding tables of a router of domain are of order 3000 and 1000 respectively. In OSPF we call the domains as areas.

Partitioning the Internet into autonomous systems and areas. An autonomous system is an administrative domain (Figure 18.2). It is a portion of global Internet under one administration, and has a common routing policy. Each autonomous system is identified by a unique number (Figure 18.22). For the purpose of routing, an autonomous system is divided into several areas in the following manner:

1. Each area is identified by a unique number. It is written in IP address like format or as a simple number.
2. There is one backbone area that interconnects other areas. The backbone area has area identifier as 0.
3. All the areas connect only to the backbone area. Thus inter-area communication takes place through the backbone area.
4. The routers that interconnect an area to the backbone area are called Area Border Routers (ABRs).
5. The routers that interconnect autonomous systems are called Autonomous System Border Router (ASBR). Communication between two autonomous systems takes place through ASBRs. ASBR can be located in Area 0 or in any other area.

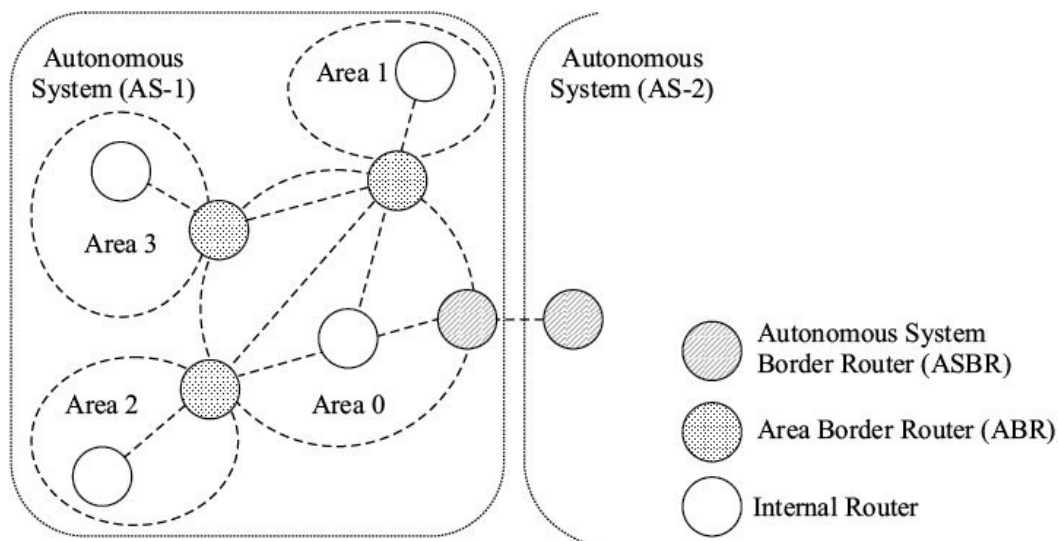


Figure 18.22 Areas and autonomous systems.

Routing in the partitioned Internet. All routers in an area maintain identical link state database. The forwarding table is created based on this database. The IP packets to the destination addresses within the area are routed using this

forwarding table. The IP packets meant for another area are sent through the ABR, which forwards them to the respective ABR that connects to the destination.

Since ABRs are at the border of two areas, a non-backbone and the backbone area (Area 0), they maintain two databases, one for each area. IP packets to the other autonomous systems are routed through ASBRs.

18.7.2 OSPF Packets

Format of OSPF routing packet is shown in Figure 18.23. There are five types of OSPF packets. All the types have a common OSPF header of 24 octets. The type field in the common header indicates the type of OSPF packet. Various fields of the common OSPF header are as follows: **Version (1 octet)**. This field identifies the OSPF version of this packet.

Type (1 octet). It identifies the type of OSPF packet. Various types of OSPF packets are as under.

Type 1: Hello packet. It is used for establishing and maintaining adjacency.

Type 2: Database Description (DD) packet. It describes the contents of the link state database. It contains LSA headers. It is exchanged when an adjacency is initialized.

Type 3: Link state request packet. It is used for requesting link state records from the neighbours.

Type 4: Link state update packet. This packet is sent as a reply to link state request. It contains the database records in form of link state advertisements. Link state update packet is also sent unsolicited at intervals of 30 minutes by a router to all other routers in an area, or when a change in the neighbourhood is detected by a router.

Type 5: Link state acknowledgement packet. It is sent on receipt of a link state update packet.

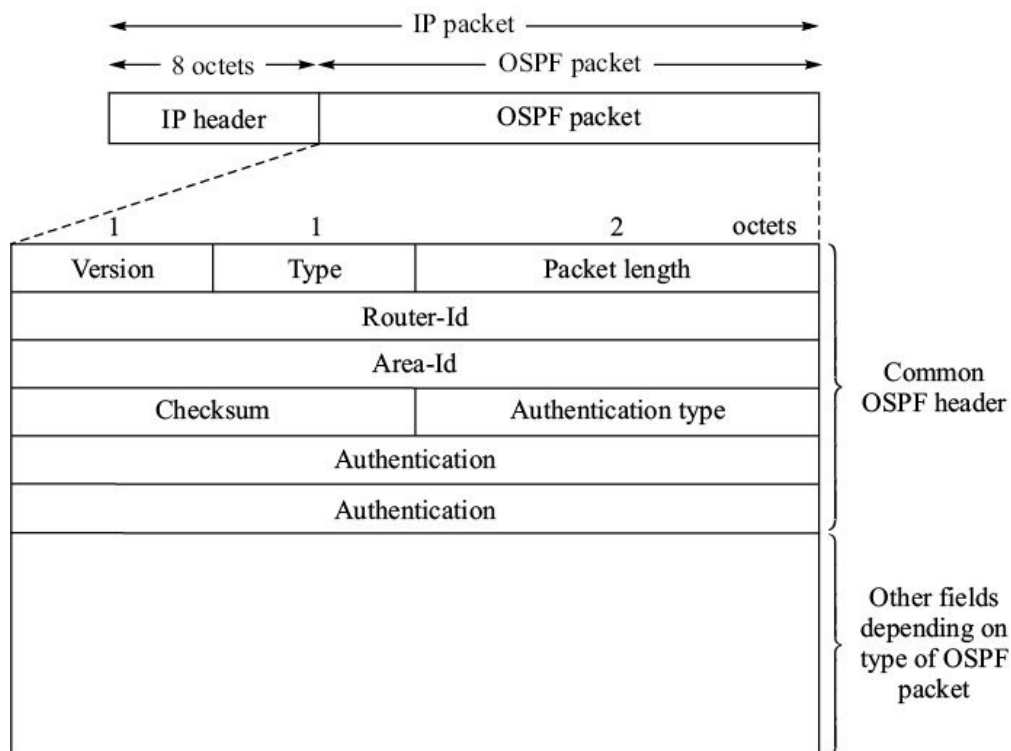


Figure 18.23 Format of OSPF packet.

Packet length (2 octets). It specifies the length of OSPF packet including header in octets.

Router-Id (4 octets). It identifies the source of OSPF packet. Each router is given a unique identifier number, which is inserted in this field.

Area-Id (4 octets). It identifies the area to which the packet belongs. All OSPF packets are associated with an area.

Checksum (2 octets). It is used for error detection. It checks the entire packet except 64-bit authentication field. The checksum is calculated as the 16-bit 1's complement of sum of 16-bit words in the packet.

Authentication type (2 octets). It indicates the type of authentication used. Authentication type is configurable on per area basis.

Type 0: No authentication.

Type 1: Password authentication.

Type 2: Cryptographic authentication.

Authentication (8 octets). It contains the authentication information *e.g.* password.

OSPF routing packet is converted into an IP packet by adding the IP header to it (Figure 18.23). The protocol type for OSPF is 89. Therefore, the protocol field of IP header contains '89' to indicate that the payload of the IP packet is OSPF packet.

18.8 FORMATION OF ADJACENCIES IN OSPF

In link state protocol, each routers tells about its links with its neighbours to all other routers. It is necessary, therefore, for a router to first establish a relationship with its neighbours and then maintain these relationships. The process establishing this relationship is termed as forming adjacencies. Formation of adjacencies with neighbours is a two step process:

- Establishing contact with a neighbour using Hello protocol.
- Link state database synchronization.

The first step enables a router to know all its neighbours and other operational parameters. In the second step, the neighbouring routers exchange the database records to remove inconsistencies if any in their databases.

18.8.1 Hello Protocol

The Hello protocol is used to establish and maintain contact with the neighbours. A router sends a Hello packet at intervals of 10 seconds to its neighbours. It also expects to receive similar packets from its neighbours at the same periodicity. Hello packets contain list of neighbours of a router. If four Hello intervals (i.e. 40 seconds) pass without hearing a Hello from a neighbour, the neighbour is declared to be dead.

The mode of exchange of Hello packets between neighbouring routers depends on the way they are interconnected. They may be connected on

- point-to-point link using HDLC, PPP or similar protocols, or
- multi-access network using LAN protocols.

In the latter case, there can be two or more routers on the same LAN.

Point-to-point links. Figure 18.24 shows typical exchange of Hello packets between two routers R1 and R2. When R2 comes up, it starts receiving Hello packets from R1 but R2 is

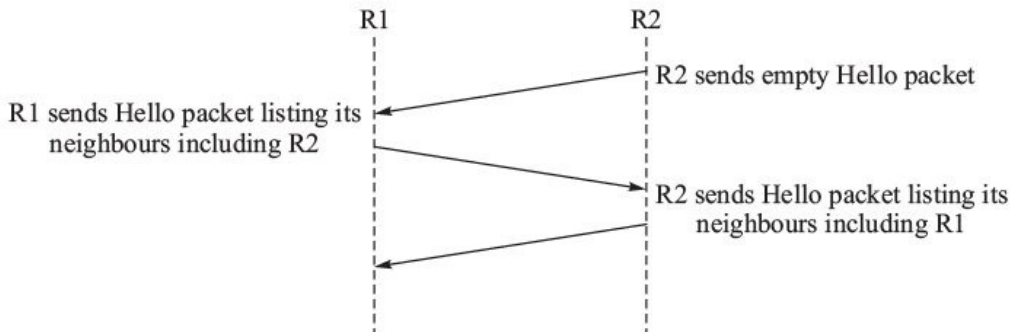


Figure 18.24 Hello protocol.

not listed in these Hello packets. To start with, R2 sends empty Hello packet to R1. When R1 receives the Hello packet from R2, it starts listing R2 in its Hello packets. Similar Hello packets listing R2 are received from other neighbours of R2 also. R2 thus acquires all its neighbours and thereafter sends Hello packets that list all its neighbours.

Multi-access networks. It is possible that several routers may be interconnected on multi-access network like a LAN (Figure 18.25a). If there are N routers on the LAN, number of adjacencies required to be formed is $N(N - 1)/2$. Since there is a common shared link interconnecting these routers, forming $N(N - 1)/2$ adjacencies serves little purpose. Ultimately, all these routers are going to advertise the same link information.

The alternative adopted is to appoint one of the routers on the LAN as Designated Router (DR). All the other routers on the LAN form adjacencies with the DR only (Figure 18.25b). In this case only $N - 1$ adjacencies will be formed. It is possible that the designated router may fail, therefore a Backup Designated Router (BDR) is also appointed. Appointment of DR and BDR is a part of Hello protocol.

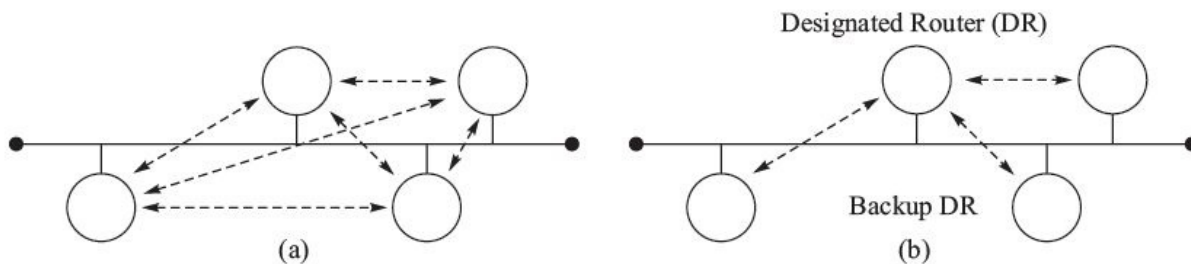


Figure 18.25 Designated router in a LAN.

Appointment of DR is carried out in one of the following ways: Hello packets are used for appointment of the DR.

1. Each router is given a priority. The router with the highest priority becomes DR. If there is a tie in priority, the router with higher router-Id becomes DR.
2. The router which comes up first assumes the role of DR if no other router comes up within a specified interval (called dead interval). If another router comes up before expiry of dead interval, DR is decided based on priority as above.
3. Once a router is elected as DR, it remains appointed as DR even if routers with higher priority or router-Id come up later. New designated router is appointed only when the existing DR fails.

The next router in order of priority or router-Id assumes the role of BDR. The BDR continuously monitors the responses (acknowledgements) released by the DR. If it finds DR is not responding, it assumes the role of DR.

Format of hello packets. Format of a Hello packet is shown in Figure 18.26. The common OSPF header contains type field as 1. Hello packets are multicast to the neighbours using IP address 224.0.0.5 in the IP header. This IP address is reserved for this purpose. Various fields of Hello packet are as follows.

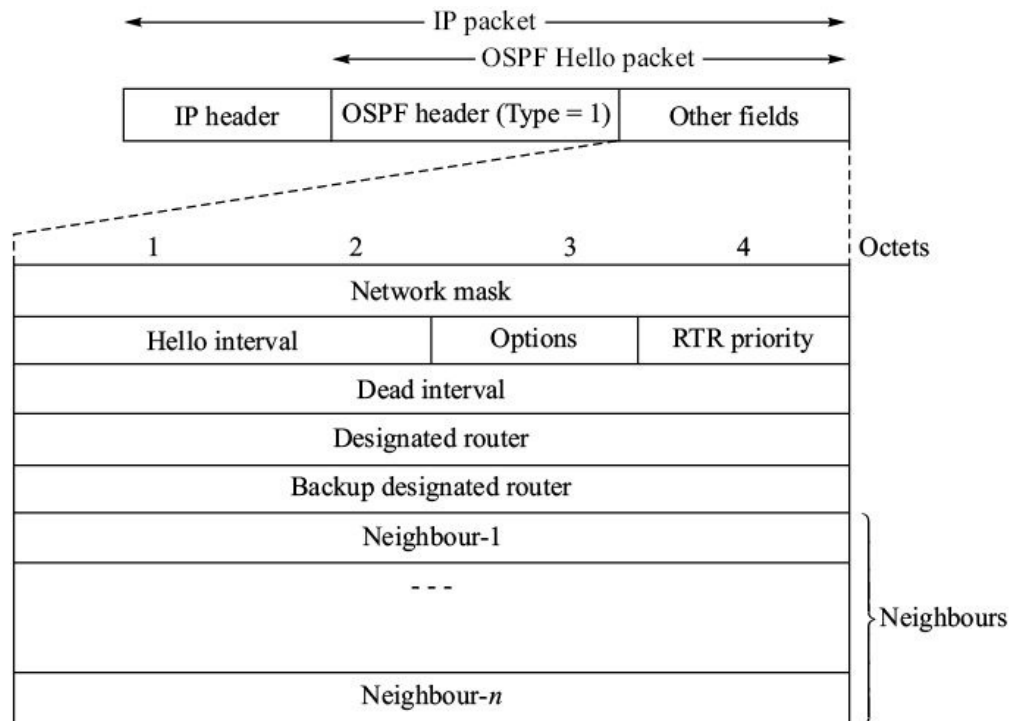


Figure 18.26 Format of Hello packet.

Network mask (4 octets). This is network mask of the interface through which the Hello packet is being sent. The neighbouring router that receives this packet, uses this mask on the source IP address⁵ to determine whether the sending router has the same network prefix as its own. Two neighbours must have same network prefix.

Hello interval (2 octets). This field defines the interval between two Hello packets sent by a router. The default value is 10 seconds.

Dead interval (4 octets). It defines the time on expiry of which the adjacency is severed if the Hello packets are not received. Default value is forty seconds.

Router priority (RTR priority, 1 octet). It is used to determine the designated router on LAN. The router with the highest priority becomes the designated router. Router with priority 0 can never be a designated router.

Designated router and backup designated router (4 octets each). These fields contain the router-Ids of the designated and backup designated routers on a local area network.

Options (1 octet). The options field indicates several features of the router, for example, whether it supports Type of Service (TOS), or whether it handles external routes, etc.

Neighbouri (4 octets). Each of these fields contains router-Id of the neighbours.

18.8.2 Database Synchronization

After exchange of Hello packets, the next step for forming adjacency is to synchronize the link state databases of the two neighbours. The initiative is taken by one of the neighbours, called master⁶ (Figure 18.27). Let us assume that R2 is master.

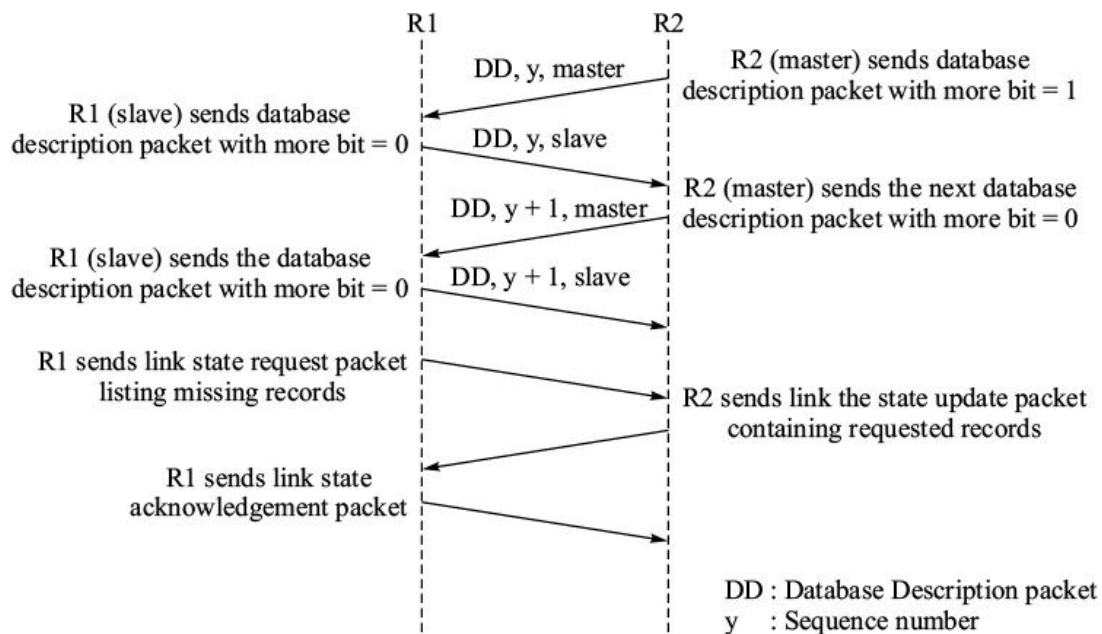


Figure 18.27 Database synchronization.

1. R2 sends Database Description (DD) packet (Type 2 OSPF packet) to the neighbour R1 (called slave). DD packet contains outline of the link state database. The outline consists of list of LSA headers. Several DD packets may be required to send all the LSA headers. Therefore, the DD packet bears a sequence number which is incremented by one in the next DD packet. Each packet carries 'more (M)' bit. M bit equal to one indicates that there are more LSA headers to send.
2. Each time R1 receives a DD packet from R2, R1 sends its DD packet to master using the same sequence number, indicating thereby that it has received the master's DD packet. In Figure 18.26, R1 sends all its LSA headers in the first DD packet and therefore it sets 'more' bit to 0.
3. The exchange of DD packets continues till M-bit is set to zero by R1 and

R2 in their DD packets.

4. R1 checks for missing LSAs by comparing its existing database and the received LSA headers. It requests for the full record of missing link states by sending link state request packet (Type 3 OSPF packet).
5. The requested LSAs are sent by R2 using link state update packet (Type 4 OSPF packet).
6. On receipt of the update packet, R1 responds with link state acknowledgement packet (Type 5 OSPF packet).
7. R2 also updates its LSA database in similar manner by sending link state request packet (Type 3 OSPF packet).

Database description packet and LSA update packets are described in the next two sections. The LSA request and acknowledgement packets are Type 3 and Type 4 OSPF packets respectively.

18.8.3 Format of Database Description Packet

Format of the database description packet is shown in Figure 18.28. IP header is followed by OSPF common header containing the type field, which identifies the OSPF packet as database description packet. The rest of the packet contains several fields, some of which have been shown in the figure and are as under. For the description of balance fields, the reader is urged to refer to RFC 2328.

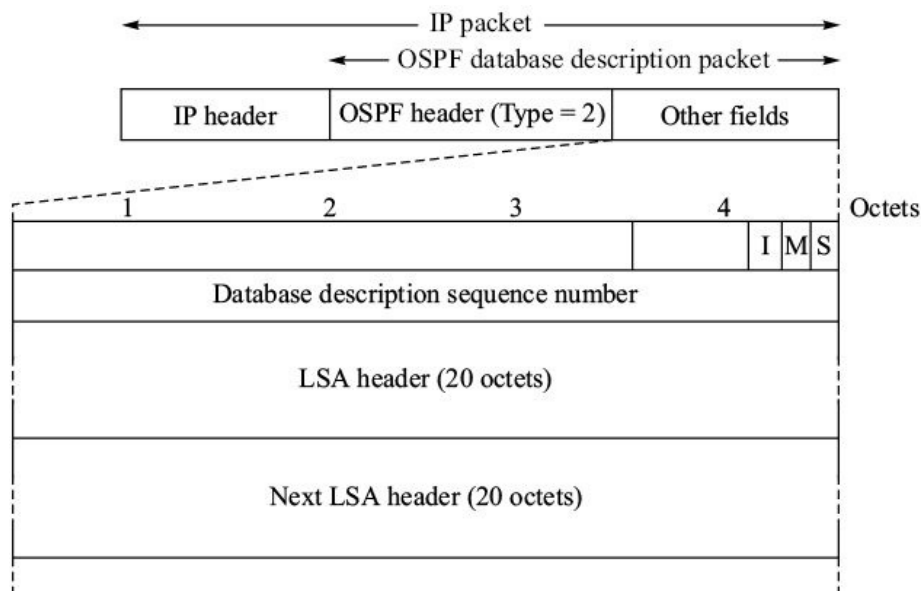


Figure 18.28 Format of database description packet.

I bit. Database description may require several IP packets due to size limitations.

I bit identifies the first database description packet.

M bit. M = 1 indicates that there are more LSA headers to send.

S bit. S = 1 indicates that the sending router is master. S = 0 indicates that the sending router is slave.

Sequence-number. It gives sequence number of the database description packets so that missing packets could be detected.

LSA header. Each link state header has a length of 20 octets and identifies a link state record of the database. Description of LSA header is given in the next section where LSA update packet is described. The number of headers present in one packet is determined by the maximum IP packet size limitation.

18.9 LINK STATE UPDATES IN OSPF

Every router maintains a database of the link states based on which the forwarding table is worked out. The database is identical in all the routers of an area, including the ABRs of the area. The database is created and maintained using OSPF link state update packets (Type 4) that contain Link State Advertisements (LSA). An LSA contains link state data with a header. The LSA header contains sequence number of the LSA and other fields. Several LSAs can be packed in one link state update packet (Figure 18.29).

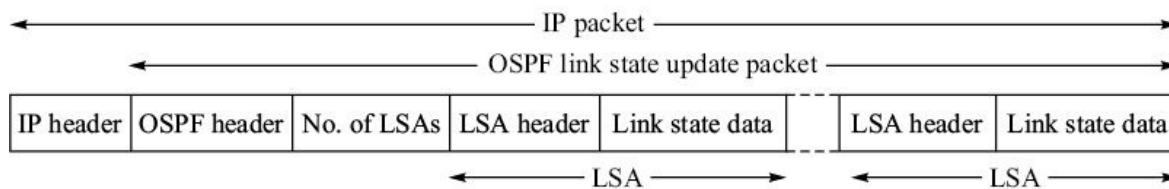


Figure 18.29 OSPF link state update packet.

The link state update packets are sent in three situations:

1. During formation of adjacency between two neighbours, a router sends link state update packet containing the LSAs requested by the other router.
2. Each link state record of the database has life of 3600 seconds (1 hour) on expiry of which it is purged from the database. The link states in the database need to be refreshed before expiry of their life. Every router, therefore, sends link state update packets to all the other routers of an area

at intervals of 30 minutes to refresh the link states of their databases. The update packets sent by a router contain LSAs pertaining to the links with its neighbours only.

3. If during exchange of hello packets, a router notices change in its neighbourhood, it communicates the change to all other routers of the area using the link state update packet.

18.9.1 Controlled Flooding

OSPF link state update packets are sent across the area using controlled flooding. When a router receives an update packet, it keeps a copy of the LSAs and forwards the update packet on all its interfaces other than the one on which it arrived. IP multicast address 224.0.0.5 is reserved for this purpose.

The receiving router sends OSPF acknowledgement packet (Type 5) to the sending neighbouring router when it receives an update packet. If acknowledgement is not received, the sending router retransmits the update packet. If the update is still not acknowledged, the adjacency with the router is severed, *i.e.* it assumes that its neighbour is no longer accessible.

Figure 18.30 shows an example of link state update from router A. A sends the update packet to routers B and C. B and C update their databases and send acknowledgement packet to A. B and C forward the update packet received from A to router D, which returns acknowledgement packet to B and C. Since D receives two update packets bearing same sequence number, it ignores the later update packet.

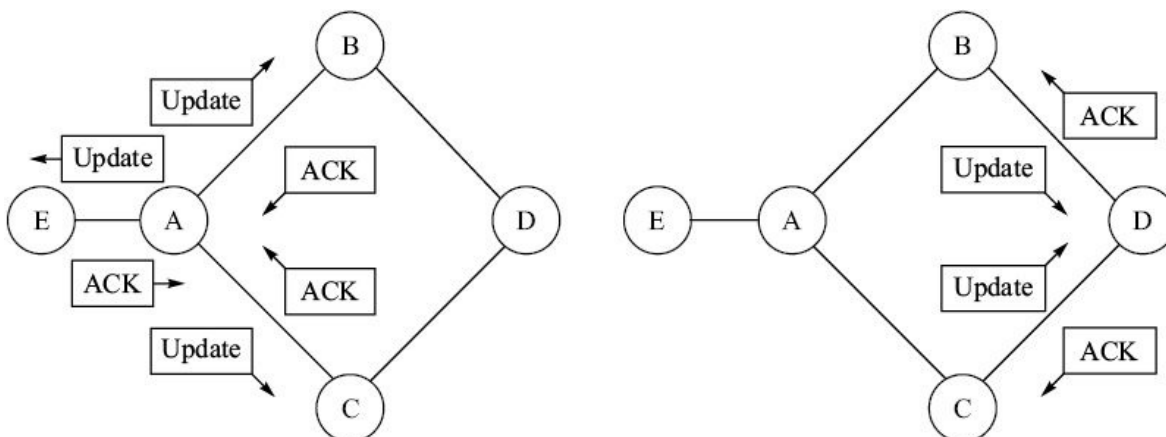


Figure 18.30 Flooding of link state updates.

EXAMPLE 18.2 Figure 18.31 shows pictorial topology of network after link A-P fails. Draw the topology changes that take place at each router when link A-P

- comes up and (a) A and P synchronize their databases;
 (b) A advertises new neighbour acquisition;
 (c) P advertises new neighbour acquisition;
 (d) C sends scheduled link state update packet;
 (e) R sends scheduled link state update packet.

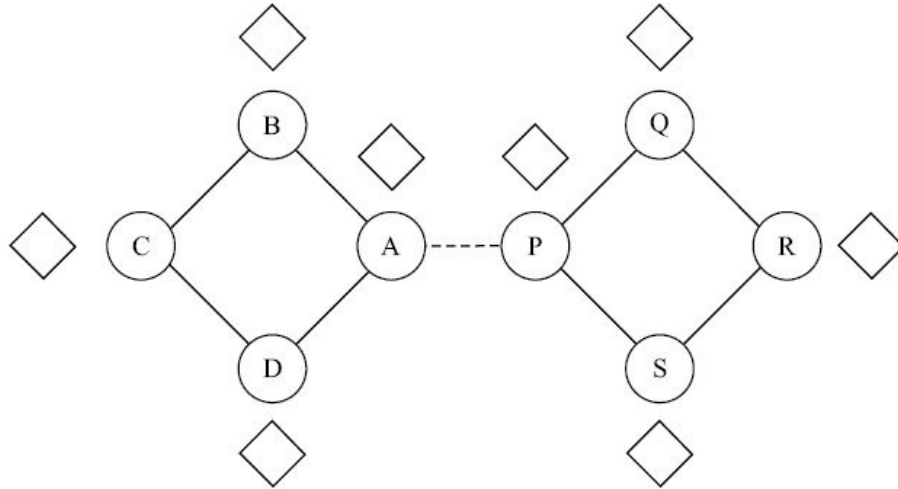


Figure 18.31 Example 18.2.

Solution

- (a) After A and P have synchronized their databases, they have full picture of network topology. Rest of the routers are still in the initial state.
 (b) When A advertises its neighbourhood, B, C, and D come to know about link A-P. Q, R, and S come to know about links A-B and A-D.
 (c) When P advertises its neighbourhood, B, C, and D come to know about link P-Q and P-S.
 (d) When C sends scheduled update, routers Q, R, and S come to know about links C-B and C-D.
 (e) When R sends scheduled update, routers B, C, and D come to know about links R-Q and R-S.

	Routers B, C, D	Router A	Router P	Router Q, R, S
Initial state				
(a) A & P synchronize their database				
(b) A advertises its neighbourhood				
(c) P advertises its neighbourhood				
(d) Scheduled update from C				
(e) Scheduled update from R				

18.9.2 Types of LSAs

There are eleven types of LSAs. Each LSA type describes a portion of OSPF routing domain. The following are the five basic LSA types (Figure 18.30):

1. Router LSA
2. Network LSA
3. Summary LSA
4. Summary LSA (ASBR)
5. AS external LSA.

Router LSA (Type 1). It is sent by a router to describe its links with its neighbours. It primarily indicates (a) number of links connected to the router, (b) type of each link, (c) link-id of each link, and (d) cost associated with each link.

For example, router A in Figure 18.32 will indicate in its router LSA that it has two links with associated costs 10 and 100.

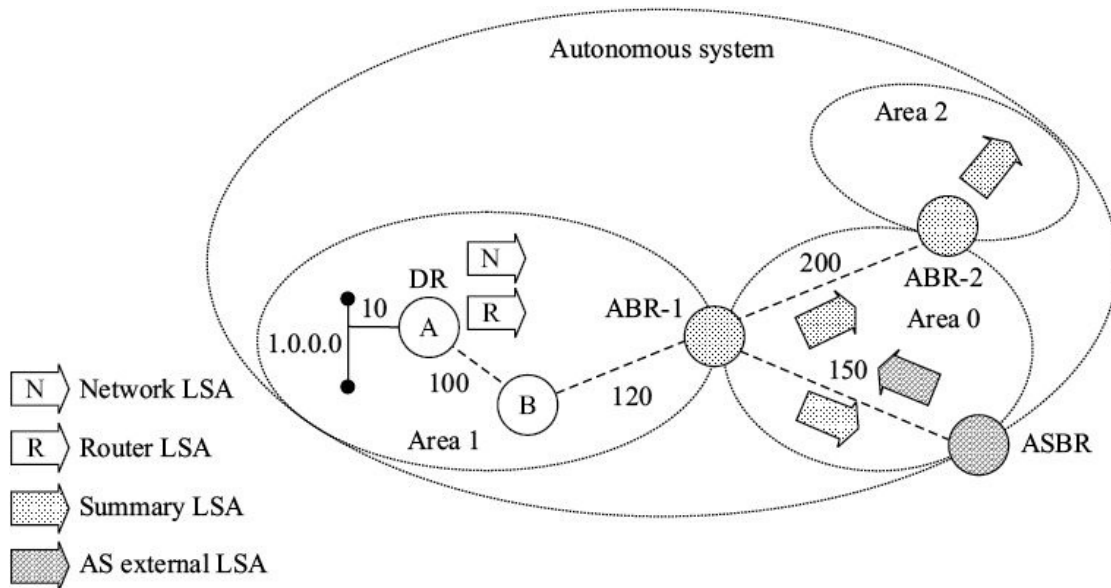


Figure 18.32 Types of LSAs.

Network LSA (Type 2). Network LSA is sent only by the designated routers. It lists the routers attached on the LAN including DR itself. It also contains the IP subnet mask for the interface of DR connected to the LAN. For example, router A which is the designated router, will indicate in its network LSA that it is the only router connected on the LAN (Figure 18.32).

Summary LSA (Type 3). It is sent by an Area Border Router (ABR) to indicate the destination addresses and the cost to reach them through the ABR. For example, ABR-1 in Figure 18.32 will indicate in its summary LSA (Type 3) released in area 0 that destination 1.0.0.0 is accessible through it at a cost of 230. On receipt of this LSA, ABR-2 will release its summary LSA (Type 3) in area 2 indicating that destination 1.0.0.0 is accessible through it at a cost of 430.

Summary LSA (Type 4). It is sent by ABR to indicate ASBR-ID and associated cost to reach the ASBR through the ABR. For example ABR-1 in Figure 18.32 will indicate in its summary LSA (Type 4) released in area 1 that ASBR is reachable through it at the cost of 150.

AS external LSA (Type 5). It is sent by an Autonomous System Boundary Router (ASBR) to describe the routes external to the autonomous system.

18.9.3 Format of LSA

Each type of LSA has a header and associated link state data. The format of the LSA header is shown in Figure 18.33. Formats of link state data portion of each

type of LSA is beyond scope of this book.

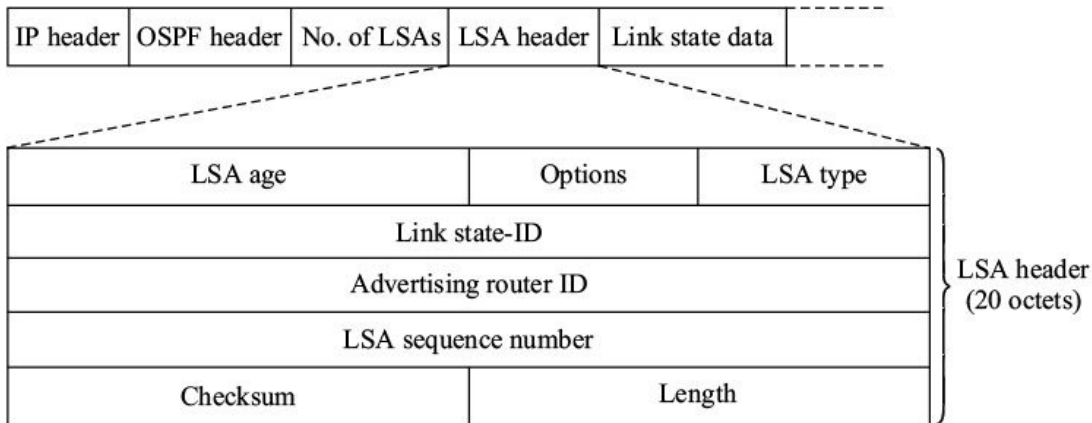


FIGURE 18.33 Format of LSA header.

LSA age (2 octets). It is the time since the LSA was first originated and is measured in seconds. During transmission, every forwarding router increments the age by one second. Unless refreshed, an LSA is purged after 3600 seconds.

LSA type (1 octet). It encodes the specific type of LSA.

Link state-ID (4 octets). Contents of this field depend on the type of LSA.

	:	
	:	
Router LSA	:	Source router-ID
Network LSA	:	IP address of DR
Summary LSA (Type 3)	:	Destination address
Summary LSA (Type 4)	:	Router-ID of ASBR
AS external LSA	:	Network address of external routes.
	:	
	:	

Advertising router (4 octets). It identifies the router that first originated the LSA.

LSA sequence number (4 octets). LSA sequence number identifies LSA version. It enables verification that each router has the most recent version of an LSA. Each time an LSA is originated, its sequence number is incremented by one.

LSA checksum (2 octets). It contains the difference of the checksum of entire

LSA and the LSA age field. It is used to ensure data integrity.

Length (2 octets). It is the length of LSA including LSA header.

18.9.4 Advantages of Link State Routing over Distance Vector Routing There are many advantages of link state routing over distance vector routing. They are:

- The routing traffic generated in link state routing is significantly smaller than that in distance vector routing because
 - the changes and ‘hello’ messages are exchanged between adjacent routers are much smaller than distance vectors. Hello messages contain only the neighbours, while the distance vectors contain all the Net-Ids in the forwarding table of a router. Even the link state update packets carry information that describe the links between neighbours only.
 - distance vectors are exchanged between neighbours at regular interval of 30 seconds.⁷ The LSAs are broadcast at an interval of 30 minutes.
- The convergence of link state algorithm across the network is faster in link state routing than distance vector routing. This is so because each router calculates its optimal paths independently. In distance vector routing, every router is dependent on its neighbours for updating its distance vector.
- In link state routing it is possible to have alternate paths.
- In link state routing, it is possible to have multiple cost metrics. The optimal path trees can be worked out for each metric separately. Forwarding decision can be based on any one of the metric.

18.10 OSI ROUTING PROTOCOLS

RIP and OSPF have been implemented in data networks widely because of the Internet. ISO also developed routing protocols for use in OSI CLNP (Connectionless Network Protocol) networks. There are three protocols:

- ES-IS protocol
- IS-IS routing protocol
- Inter Domain Routing Protocol (IDRP).

ES stands for end system and IS stands for intermediate system, which in our terminology is a router. These are not alternative protocols like RIP and OSPF. All the three protocols form a suite and are required in an internetwork.

- ES-IS protocol is a discovery protocol which enables ES and IS to discover each other.
- IS-IS protocol is routing protocol based on link state algorithm.
- IDRP is based on BGP and enables routing of data packets from one domain to another.

ISO 10589 defines IS-IS, ISO 9542 defines ES-IS, and ISO 10747 defines IDRP. We will not go into details of these protocols because of their limited deployment and because the basic principles are same as already discussed.

18.10.1 OSI Routing Terminology

OSI routing protocol terminology, the distinguishing features and similarities with respect to OSPF are as follows:

- OSI partitions internetwork into areas and domains. Domain is equivalent to autonomous system.
- Routing within an area is referred to as level-1 routing. Routing between areas is referred to as level-2 routing.
- ES-IS protocol is a configuration and discovery protocol. An ES sends ES Hello messages (ESH) to an IS. An IS sends IS Hello messages (ISH) to an ES. These hello messages are intended to convey network layer addresses and physical addresses.
- IS-IS routing protocol is a link state routing protocol. Link state PDUs (LSPs) are equivalent of link state update packets of OSPF.
- Sequence number PDUs in IS-IS protocol are equivalent of DD packets in OSPF. These are used for synchronizing the link state database.
- IS-IS uses a default metric with maximum path value of 1024. Any single link can have maximum value of 64 and path value is sum of metrics of the constituent links. There are three optional metrics—cost, delay, and error rate.
- IS-IS maintains link state databases and uses Dijkstra's algorithm to find the shortest paths.
- The designated routers are chosen in manner similar to OSPF.

- IS-IS uses ISO addressing scheme. An address can have maximum 20 bytes (Figure 18.34). It consists of three parts—Area-Id, System-Id, and *n*-selector.

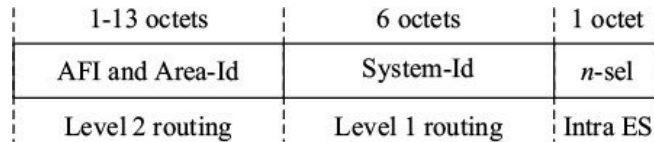


Figure 18.34 ISO addressing scheme.

The following example of ISO address illustrates the format. All the digits are in hexadecimal format.

49.0001.00a0.c96b.c490.00

First octet 49 Authority and Format Indicator (AFI) Next two octets 0001 Area-Id

Next six octets 00a0 c96b c490 System-Id (MAC address) Last octet 00 *n*-selector *n*-selector determines the N-SAP. Thus, up to 256 transport layer entities can be selected.

- An ES can have multiple addresses. These addresses differ only by the last octet, *n*-selector.
- IS-IS supports authentication.
- Dual IS-IS routing protocol (also called integrated IS-IS protocol) supports both IP and CLNP.

18.11 INTERIOR AND EXTERIOR GATEWAY PROTOCOLS

The routing protocols are categorized as:

- Interior gateway protocols
- Exterior gateway protocols.

18.11.1 Interior Gateway Protocols

RIP and OSPF routing protocols are called Interior Gateway Protocols IGP and are implemented within an autonomous system. All the routers within an

autonomous system cooperate and follow a common administrative policy. The routing decisions within an autonomous system are based on technical parameters like bandwidth, delay, hop-count, *etc.* These routing protocols have the following limitations when used in large networks like the Internet:

1. Maximum hop count (RIP).
2. Long time to transmit distance vectors on slow speed links (RIP).
3. High CPU utilization for shortest path calculations (OSPF).
4. Large memory space required for storing forwarding table (RIP, OSPF).
5. Large memory space required for storing link state database (OSPF).
6. Only two level hierarchy in OSPF.
7. Non-technical parameters for route cannot be taken into account for routing decisions.

18.11.2 Exterior Gateway Protocols

For interconnecting autonomous systems, we use a different category of protocols called, Exterior Gateway Protocols (EGP). These are different from IGP because we need to implement routing policies that must take into account non-technical parameters because the routers that interconnect two ASs may not necessarily trust each other.

Figure 18.35 shows three autonomous systems AS1, AS2, and AS3. Suppose link A-B, which is an internal link of AS1 goes down. Network administrator of AS1 may transit its traffic between A and B via router C of AS2. Unless AS1 and AS2 have previously agreed for such use of their network resources, transit of traffic between A and B via C may upset network administrator of AS2. Let us examine another similar situation. Suppose link B-D goes down. AS3 gets isolated from AS1. Network administration of AS1 can use the link C-D between AS2 and AS3 for its traffic to AS3.

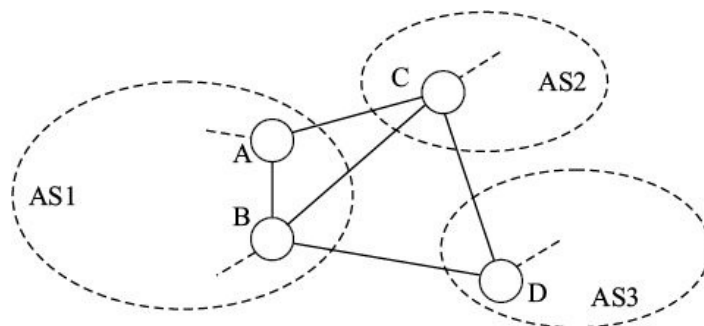


FIGURE 18.35 Exterior gateway protocols.

Exterior gateway protocols implement policies that protect the interests of the individual network administrations and ensure that their network is not illegally intruded. Border Gateway Protocol (BGP) is the most common exterior routing protocol implemented globally.

18.12 BORDER GATEWAY PROTOCOL (BGP)

Border gateway protocol, version 4, commonly referred to as BGP4 is an inter-autonomous routing protocol. It is a very robust and scalable routing protocol as evidenced by its use in the Internet. It is documented in RFC 1771. Its basic features are

- It is an exterior gateway protocol used between ASBRs (Autonomous System Border Routers).
- It supports classless routing.
- It is a path vector protocol where the distance vectors are annotated with path and the policy attributes.

BGP is also used as routing protocol between two ASBRs within an autonomous system (Figure 18.36). It is then referred to as Interior BGP (IBGP). When used between autonomous systems, BGP is referred to as Exterior BGP (EBGP).

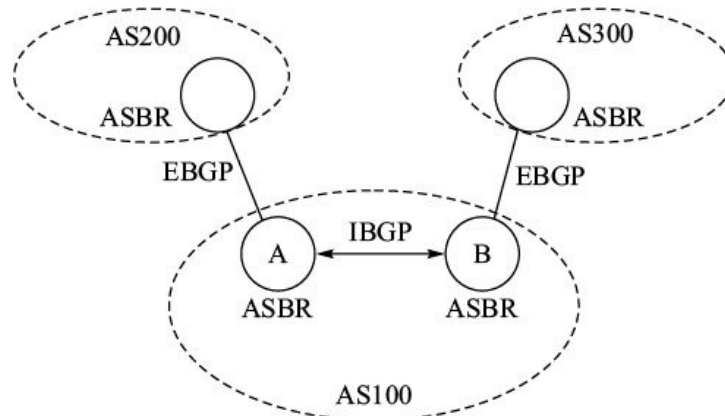


Figure 18.36 IBGP and EBGP.

18.12.1 Basic Operation

BGP session consists of exchange of BGP messages that convey network reachability information. The session can be between two ASBRs of different

autonomous systems or of the same autonomous system. BGP messages are sent over TCP connection between neighbours. TCP port 179 is used for BGP.

Network reachability information exchanged between two BGP routers consists of the following parameters (Figure 18.37):

- The destination network-id
- List of AS numbers of intervening autonomous systems
- The next hop.

Thus BGP is a path vector routing protocol. Being a path vector routing protocol, there are no routing loops when BGP is used. Full path vectors are exchanged when the TCP connection is established for the first time. Thereafter only the changes are sent as and when they occur. There are no periodic routing updates.

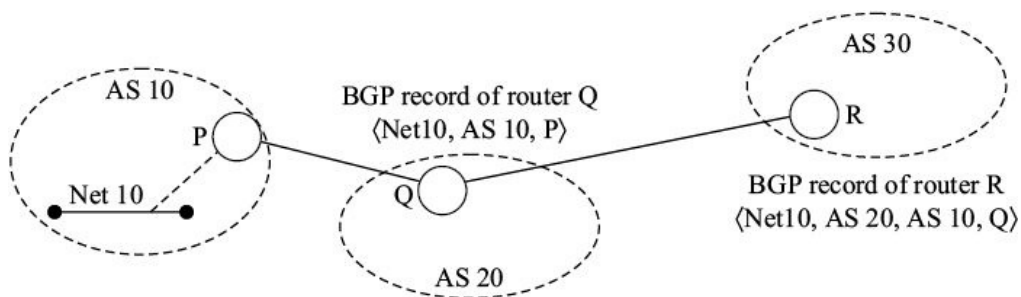


Figure 18.37 Path vector approach in BGP.

Each path is associated with path attributes that determine selection of a path as the accepted route. Path attributes are described later.

18.12.2 BGP Messages

There are four types of BGP messages:

1. Open message
2. Update
3. Notification
4. Keep-alive.

Open message. It is used to open neighbour relationship with another ASBR. It contains AS number, IP address, and other operational parameters.

Update message. It is used for sending routing information which can be

- the routes previously advertised being withdrawn and/or
- a new route being added.
- The new route being advertised contains the following information:
- List of the network prefixes that can be reached through this route. This field is called Network Layer Reachability Information (NLRI).
- Length of path attributes field (described next).
- List of path attributes that apply to this particular route.

Notification message. It is sent when an error condition is detected.

Keep-alive message. It is sent by a router (a) as reply to open-message and (b) to intimate its alive status to the neighbour at regular interval equal to the hold timer.

18.12.3 Path Attributes

BGP uses many path parameters, called attributes that are used to determine the best route to a particular destination. The ‘best route’ is within the framework of defined routing policies. The various path attributes are

- Local preference
- AS path
- Multi-exit discriminator
- Origin
- Next hop
- Community.

Local preference. The local preference attribute determines the preferred path out of an autonomous system. It is a numeric value. Higher value indicates higher preference. In Figure 18.38, autonomous system 100 (AS100) advertises destination 192.168.27.0/27 through paths via AS200 and AS300. Routers A and B of AS400 receive these routes from AS200 and AS300 respectively. Administrator of AS400 based on its routing policy may decide to use the path through AS200 for destination 192.168.27.0/27. Therefore, it attaches higher local preference value to the path through AS200. The decision to use the path through AS200 may be based on several considerations:

- Commercial terms of agreement between AS200 and AS400

- Availability spare bandwidth between AS200 and AS400 to take this traffic
- Others.

When routers A and B exchange the received route using IBGP, they internally exchange the local preference values also. Note that the local preference values remain within AS400 and are not to be transmitted on EBGP links.

AS path. When a route advertisement passes through an autonomous system, the BGP router adds the AS number to the ordered list of AS numbers that the advertisement has traversed. For example, in Figure 18.38, when the route advertisement for network prefix 192.168.27.0/27

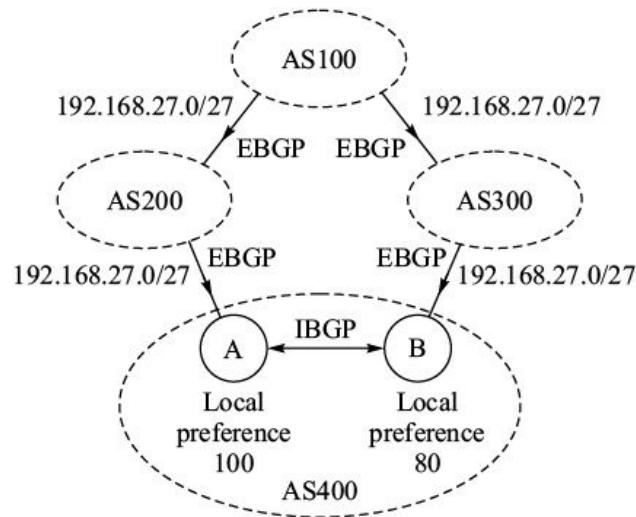


Figure 18.38 Local preference.

reaches router B via AS300, the route will contain AS path numbers AS100 and AS300. AS path attribute is mandatory and it is always present when a route is sent to a BGP peer.

If a router finds that the local AS number is already in the path, it means that the BGP route has been through the AS already. Accepting the route would cause a loop. Therefore, BGP drops the route.

Multi-exit discriminator (MED) attribute. When an autonomous system P has multiple exits to another autonomous system Q, P can suggest its preferred route by this attribute. The route with lower value of the attribute is the preferred route. Q can set local preference parameter based on this attribute if it so wishes. In Figure 18.39, router A advertises route 192.168.27.0/27 with MED equal to 10 and router B advertises the same route with MED 5. Thus autonomous system

P clearly indicates its preference for the route through B.

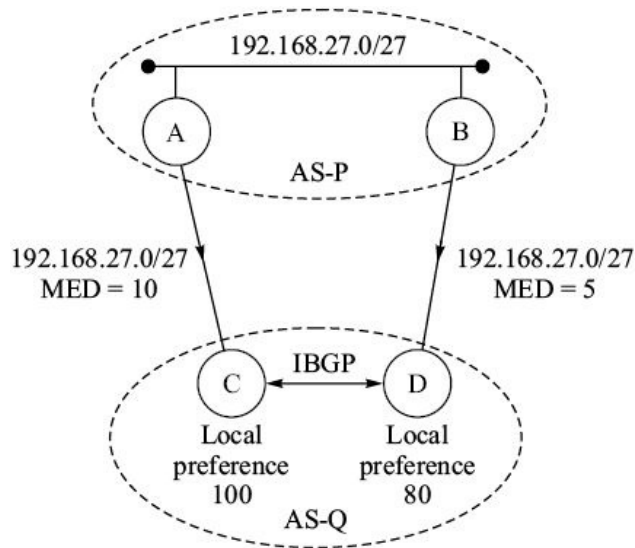


Figure 18.39 Multi-exit discriminator.

Origin. It indicates whether this route was learnt by interior gateway protocol or an exterior gateway protocol (e.g. BGP). IGP route is given code 0. EGP route is given code 1 and unknown route is given code 2. For example, network 192.168.27.0/27 is advertised with origin code 0 by router A (Figure 18.39). In Figure 18.38, AS200 will advertise 192.168.27.0/27 to AS400 as external route and therefore give origin code 1. Thus routes with origin code 0 are direct routes and are given preference.

Next hop. It is the IP address of the ASBR that should be used as next hop for the network prefixes listed in NLRI field. When a BGP update is sent, the BGP router puts its IP address in the path attributes list. For example, in Figure 18.40, the BGP router A puts its IP address in the next hop field. When this update leaves AS200 towards AS400, router B replaces this IP address with its own IP address. Thus, router C has the next hop IP address for advertised destination.

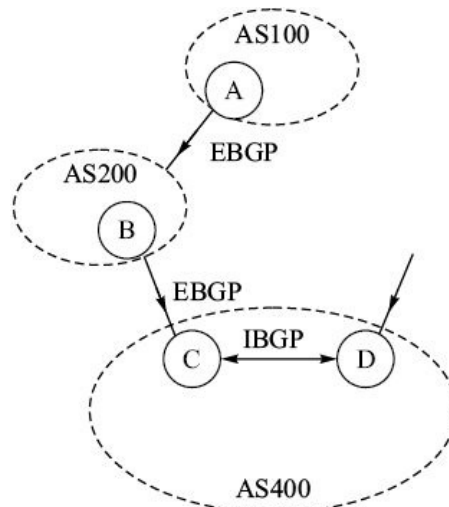


Figure 18.40 Next hop.

Next hop address is replaced on EBGP links only. When C communicates this route to router D on IBGP, it does not change the IP address of B in the next-hop field (Figure 18.40). Thus, as far as router D is concerned, the next hop for the traffic to network 192.168.27.0/27 is router B. If it has a path to B, it will retain the route else it will drop the route received from C.

Community. A BGP community is a group of destination networks that share a common property. Community information is used to perform a certain action on all the routes belonging to a community. For example, we can define routing policy for a community so that the policy becomes applicable to all the destination routes belonging to that community. Predefined community attributes are:

- No-export
- No-advertise
- Internet.

No-Export. In the case of no-export, the route is not further advertised to other EBGP peers (Figure 18.41a). When router A (AS1) advertises a route with no export community attribute, router B advertises the route within AS2. Router C which receives this route from B does not further export it out of AS2.

No-Advertise. In the case of no-advertise, the route is not further advertised to any router (Figure 18.41b). Router B does not advertise the route any further even to internal routers.

Internet. In the case of Internet community, the route is advertised to all the

peer and the Internet. There is no restriction (Figure 18.41c).

Community attribute information is added as a path attribute in the BGP update message.

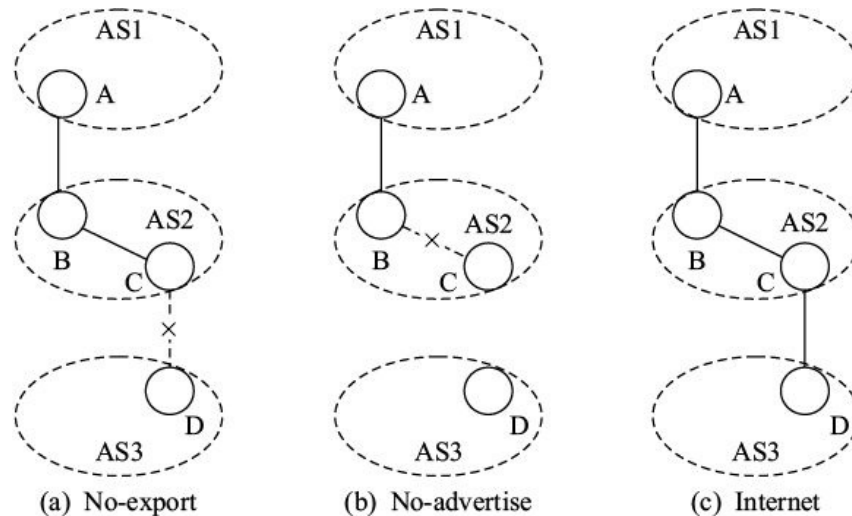


FIGURE 18.41 Communities.

18.12.4 Choosing the Active Route

A BGP router may have multiple routes to same network prefix. It must select one route put in the forwarding table. The various attributes associated with the routes determine the selection. A route is selected in the following order. If there are multiple choices at any step, the next selection criterion is adopted. The process is stopped as soon as single route is left and it is put in the forwarding table.

1. Drop those routes in which the specified next-hop is inaccessible.
2. Select the route with the highest local preference value.
3. Select the route with the shortest AS path.
4. Select the route with lowest origin code.
5. Select the route with the lowest MED value.
6. Select the route learnt from EBGP peer rather than that learnt from IBGP peer.
7. Select the route with the lowest IGP metric.
8. Select the route through the peer BGP router with the lowest router-Id.
9. Select the route through the peer BGP with the lowest IP address.

SUMMARY

In this chapter, we described routing protocols that enable creation and maintenance of the forwarding tables. The routing protocols are based on one of two basic algorithms, distance vector algorithm and link state algorithm. In distance vector algorithm, a router periodically sends its distance vector to all its neighbouring routers. Distance vector lists all the known destinations and their distances from the router. Each router works out its forwarding table based on the distance vectors it receives from its neighbours. In link state algorithm, every router broadcasts list of its neighbours and their link costs to all the routers in the internetwork.

Distance vector algorithm suffers from the problem of count-to-infinity, which is contained by using techniques like split horizon, path vector and hold down timer. Routing Information Protocol (RIP) is a distance vector routing protocol. The first version of RIP had limitations of not making use of the redundant paths and not having provision for subnet masks. It could support only classful routing. RIPv2 is classless routing protocol. The RIPv2 packet contains additional field for subnet masks. It supports authentication of routing updates.

Link state algorithm allows each router to get complete topology of the internetwork. The shortest paths to the various destinations are calculated using Dijkstra's algorithm. Open Shortest Path First (OSPF) routing protocol uses link state algorithm for constructing and updating the forwarding tables.

In a large network, a router cannot store information about every other router. There is large overhead of exchanging this information periodically. Therefore, an Internet is divided into hierarchy of autonomous systems and areas. Each router knows only about the routers of its area. There are area border routers that summarize the routing information of an area and exchange it as the inter-area routing information.

RIP and OSPF are called Interior Gateway Routing Protocols (IGP). Exterior Gateway Routing Protocols (EGP) work across autonomous systems. These protocols are different from IGP because two autonomous systems do not necessarily trust each other. Border Gateway Routing Protocol (BGP) is an example of such protocol. It is a path vector protocol where the distance vectors are annotated with path and the policy attributes. It supports classless routing.

EXERCISES

1. In the given figure, routers A, B, C, and D use split horizon with poisonous

reverse. The initial status of their distance vectors is as shown.

- (a) What are the revised distance vectors of routers C and D after they receive the distance vector from B?
- (b) What are the distance vectors sent by router D to routers B and C after receiving distance vector from router B?

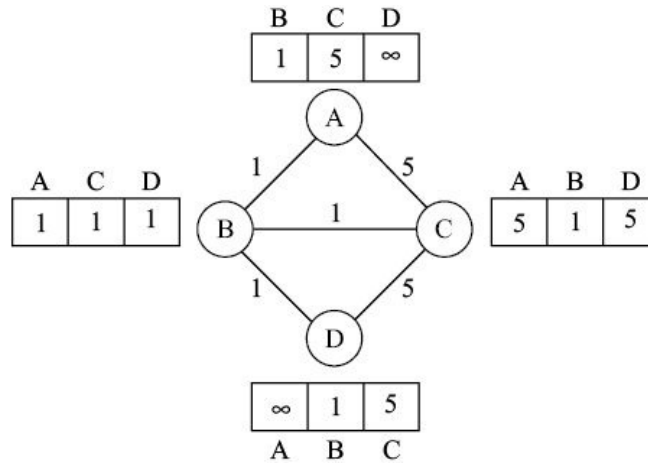


Figure E18.42.

2. In the previous figure, B sends its path vector to A.
 - (a) What is the path vector of A after receipt of the path vector from B?
 - (b) If link B-D goes down, how does B arrive at the optimal path to D?
3. For the internetwork given in the following figure give the distance vector of each node
 - (a) when each router knows the distances to its immediate neighbours only.
 - (b) when A sends its distance vector to its neighbours.
 - (c) when B sends its distance vector to its neighbours after step (b) above.
4. Using Dijkstra's algorithm, determine the shortest paths from node A to the rest of the nodes of the internetwork shown in Exercise 3. Write the routing table of node A.

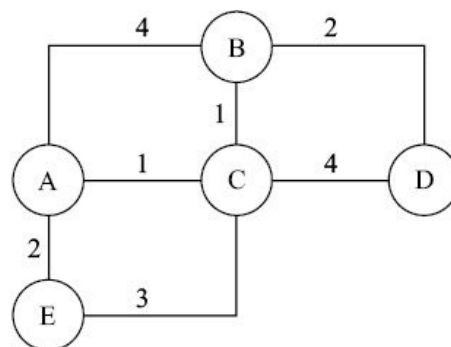


Figure E18.43.

Figure E18.43.

5. Using Dijkstra's algorithm, determine the shortest paths from node A to the rest of the nodes of the internetwork shown below. Write the routing table of node A.

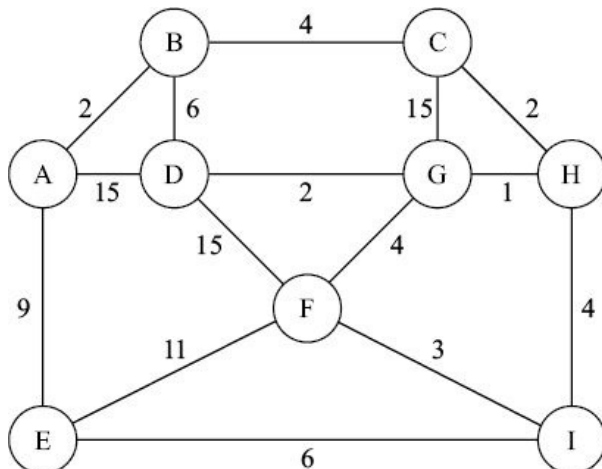


Figure E18.44.

6. Figure E18.45 shows a internetwork that employs link state routing. Links B-D and C-D have been down for a while. The pictorial topology of the LSA databases at each router is as shown. Show how the database at each router builds up when (a) Link B-D comes up and routers B and D synchronize their databases.
 (b) B broadcasts its LSAs.
 (c) D broadcasts its LSAs.
 (d) Link C-D comes up and routers C and D synchronize their databases.
 (e) D broadcasts its LSAs.
 (f) C broadcasts its LSAs.

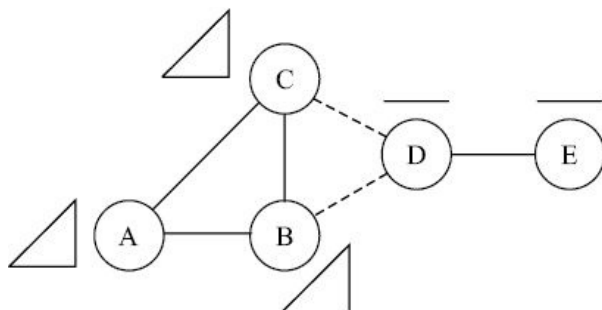


Figure E18.45.

7. Assume that a BGP router is learning the same route from two different EBGP peers. The AS path information from peer 1 is 51, 86, 2345 and the

path information from peer 2 is 51, 2346. What BGP attributes could be adjusted to force the router to prefer the route advertised by peer 1?

8. What are the two capabilities supported by RIPv2 but not by RIPv1?
9. In OSPF, Area 0 contains ABRs P, Q, and R and Area 1 contains routers R, S, and T. Which routers is router T aware of?
10. An internetwork consists of routers A, B, C, D, E, and F. The forwarding tables of routers A and F are shown in the following tables. The cost metric is hops-count. Draw the internetwork consistent with the forwarding tables.

Router A		
Destination router	Cost	Next hop
B	1	B
D	1	D
C	2	B
E	3	D
F	2	D

Router F		
Destination router	Cost	Next hop
A	2	D
B	3	D
D	1	C
C	2	D
E	1	E

11. Determine the forwarding table of router D using Dijkstra's algorithm. Assume the following link costs. The costs of releasing a packet into various networks are: Cost of releasing a packet into N1, N3, N4, N5 : 1
 Cost of releasing a packet into N2 : 3
 Cost of releasing a packet into N6 : 2
 Cost of releasing a packet into N7 : 4

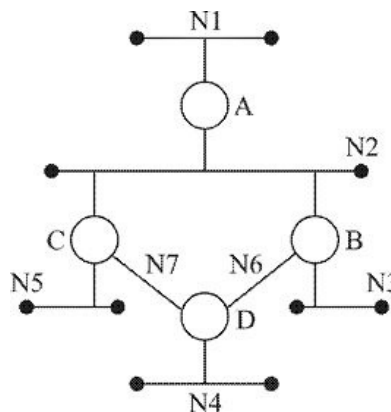


Figure E18.46.

- 1 Strictly speaking, the routing information exchange is limited to the routers within an area in hierarchical internetworks.
- 2 MTU is maximum transmittable unit which the maximum size of an IP packet that can be transmitted on a link.

- 3 We will discuss authentication in detail in Chapter 21 on Network Security.
- 4 Controlled flooding is described later. The objective is to send the packet to each and every router.
- 5 Source IP address is available in the IP header.
- 6 The router with higher router-Id is the master.
- 7 RIP uses 30 seconds refresh interval.

19

Multicasting and Multiprotocol Label Switching (MPLS)

In the last chapter we examined the routing protocols used in the IP networks. In this chapter we continue the discussion on IP networks and examine two very important concepts—multicasting and MPLS (Multiprotocol Label Switching). These are discussed in the context of IP networks. Multicasting refers to sending an IP packet to several destinations simultaneously. It is required for services like videoconferencing in which several destinations simultaneously get the same video signal from a source. MPLS is used in IP networks primarily for traffic engineering and for transporting data units of other protocols, *e.g.* ATM.

We begin this chapter with the concepts of multicasting and study Reverse Path Forwarding (RPF) and Core Based Tree (CBT) mechanisms for multicasting. Then we examine multicasting protocols PIM (Protocol Independent Multicast), DVMRP (Distance Vector Multicasting Routing Protocol), MOSPF (Multicast extensions to OSPF), and IGMP (Internet Group Management Protocol). We study multicast addressing scheme before moving over to MPLS. We study the basic MPLS operation, MPLS label distribution mechanisms, traffic engineering, and MPLS tunneling concepts.

19.1 MULTICASTING

Typically, an IP packet is required to be routed from the source to one destination as specified in the destination address field. This mode of operation is referred to as unicast. There are number of applications, where a packet is required to be sent to a group of destinations. The mode of sending a packet from one source to multiple destinations is called multicast. Multicasting is different from broadcasting where the every destination receives the packets from the source. In multicasting only the defined members of a multicast group

receive the packets from the source.

Multimedia and videoconferencing are the two main applications of multicasting. Multimedia application involves distribution of video and audio signals from one source to several workstations using an IP network. In videoconferencing, a group of workstations communicates with each other so that transmission from one member of the group is received by the other members. Multicasting can be used for locating resources (sending queries to several servers) and for advertising (sending an update to several servers).

In principle, the objective of multicasting can be achieved by sending multiple copies of the same IP packet using normal unicast. The copies are individually addressed to all the members of a multicast group. But it is not very efficient way of doing multicast since same packet will appear more than once on interconnecting links (Figure 19.1a) and therefore excess traffic is generated on the links. The ideal way of multicasting will be to forward only one packet on the next link (Figure 19.1b). We will examine the various ways of achieving this objective in this chapter.

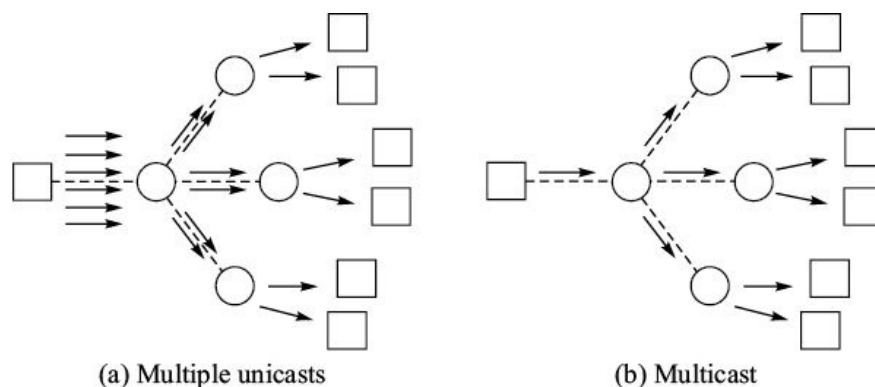


Figure 19.1 Multiple unicasts and multicast.

19.1.1 Multicast Group

A multicast group associates a set of senders and receivers, and it is identified by a unique IP multicast group address. The sender sends its IP packets with multicast group address and the routers forward the packets to the members of the multicast group. There can be one or several simultaneous senders in a group.

Multicast routing addresses the issue of linking the members of a group. The basic scheme for linking members of a group works in the following manner:

- The receivers (members of the group) register themselves with the router

they are attached to, indicating their multicast group address.

- If an IP packet with the multicast group address is received by the router, it forwards the packet to the registered member(s) of the group.
- It is possible that the members of a multicast group are mobile, *i.e.* they may move to a different router. The routing scheme must take into account this factor.

The multicast routing ensures that multicast packets are delivered to all such routers that have registered multicast group members. One simplistic way can be to flood the multicast packets to all the routers and let an edge router discard the packet if it does not have any registered member of the group. This scheme creates large unnecessary traffic. There can be more efficient ways of routing multicast packets. We examine these in the following sections.

19.2 MULTICAST ROUTING PRINCIPLES

Multicast on IP network can be implemented by extending the concepts of routing and forwarding already implemented in the routers of the IP network. Let us start from the basics and see the limitations of the following intuitive solutions for multicasting:

- Controlled flooding
- Spanning tree.

19.2.1 Multicast Using Controlled Flooding

Controlled flooding is the simplest multicast algorithm. It works in the following manner:

- When a node receives a multicast packet, it forwards the packet to all the interfaces except to the interface through which the packet came.
- The above step is carried out if the packet is received for the first time. It is possible that the packet goes in a loop and comes back to the router again. If a multicast packet is received second time, or in other words, if it is a duplicate, it is discarded by the router.

Controlled flooding mechanism is successfully used for propagation of LSAs

in OSPF. The multicast packet is received by all destinations and duplicates are discarded at the first point of detection. Discarding of duplicate packets avoids generation of unnecessary traffic and packets circulating in loops. Figure 19.2 illustrates the mechanism.

1. End system A sends multicast packet to router R1. R1 forwards it to routers R2 and R3.
2. R2 receives the multicast packet from R1. It forwards the packet to end system B and routers R3 and R4.
3. R3 receives the multicast packet from R1. It forwards the packet to R2 and R4. The packet sent by R3 to R2 is discarded by R2 since it is a duplicate packet. R3 also receives the multicast packet from router R2. It discards this packet because it is a duplicate packet.
4. R4 receives a multicast packet from R2. It forwards the packet to end system C and router R3. R3 discards this packet received from R4 as it is a duplicate packet. R4 also receives a duplicate packet from R3, which it discards.

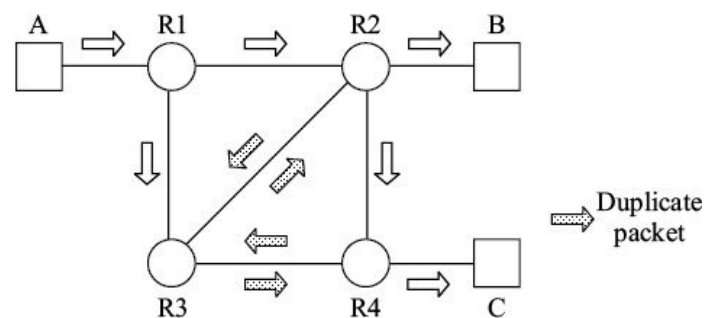


Figure 19.2 Multicast using controlled flooding.

Note that if the duplicate packets are not discarded, they will keep circulating in the loop. Using controlled flooding for IP multicast has the following problems:

Identification of duplicate packets. In OSPF the LSAs have sequence numbers and age. If a received LSA is already in the database, then it must be duplicate. IP packets do not have sequence numbers. Thus the routers will need to maintain a list of ‘recently’ received IP packets based on some identifier. Even if it is done somehow, routers will require large memory, and referring to the list will slow down their forwarding mechanism.

Unnecessary traffic. Note that the shaded packets in Figure 19.2 were not

required to be transmitted but these transmissions could not be avoided.

Group membership. We have not considered the group memberships. Controlled flooding sends multicast packets to all the edge routers even if there are no multicast group members. There was no need for R1 to forward the multicast packet to R3. This could not be avoided because the controlled flooding generates traffic on all the available paths rather than the required paths.

19.2.2 Multicast Using Spanning Tree

Spanning tree is a more efficient solution than flooding as far as generation of extraneous traffic is concerned. From each source, a spanning tree that touches every edge router in the network is computed. If link state routing mechanism is used, computation of spanning tree is straightforward as each router has LSA database of the entire network.

Figure 19.3 shows the spanning tree of the network of the last example. The spanning tree has been worked out when system A is the source of the multicast. Router R1 is, therefore, at the root of the tree. The multicast IP packets from A follow the tree and reach every router, R2, R3, and R4 taking the shortest path. There are no duplicates.

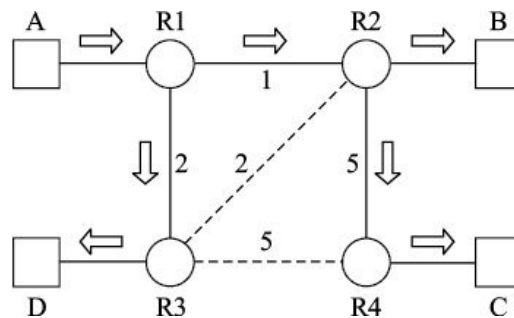


Figure 19.3 Spanning tree for multicasting.

Spanning tree multicasting has the following drawbacks:

Large router memory. Consider that instead of A, end system B is the source of multicast. In this case, the spanning tree will be constructed with R2 as the root. We have not yet considered that a router will be attached to several end systems, belonging to different multicast groups. Thus, each router will be required to store a spanning tree for each source of every multicast group. This is obviously very expensive from the point of view of router memory.

Sub-optimal routing. The number of spanning trees can be reduced, if we

restrict to one spanning tree for each multicast group. For example, we can use the spanning tree with R1 as root for multicast from C as source. Though duplicates are not there, but the routes taken by multicast IP packets are sub-optimal. For example, direct path from C to D via R4 and R3 is shorter than the path that follows the tree (R4-R2-R1-R3).

Non-uniform dispersal of traffic. There is another problem in the above solution. If all the end systems (A, B, C, and D) send their multicasts simultaneously, common link between R1 and R2 will soon get choked as all the multicast transmissions go through it.

Group members. Spanning tree multicast algorithm also does not take into account the group membership as in the case of controlled flooding.

19.3 REVERSE PATH FORWARDING (RPF)

Multicasting mechanisms based on controlled flooding and spanning tree have their limitations. We need some specialized routing approaches for multicast traffic. The Reverse Path Forwarding (RPF) multicasting mechanism is one such approach. RPF is based on the following principle:

- When a multicast packet is received by a router (R), take note of its Source Address (SA) and the incoming port (I). If the shortest path from the router (R) to the SA goes through the interface I, then forward the packet to all other interfaces except I. Otherwise discard the packet.

This rule overcomes the problem of identification of duplicates described earlier. It assumes that a packet arriving at a port that is not in the shortest path must be a duplicate and therefore should be discarded. Figure 19.4 illustrates this mechanism.

The cost metrics for determining the shortest path are indicated in Figure 19.4.

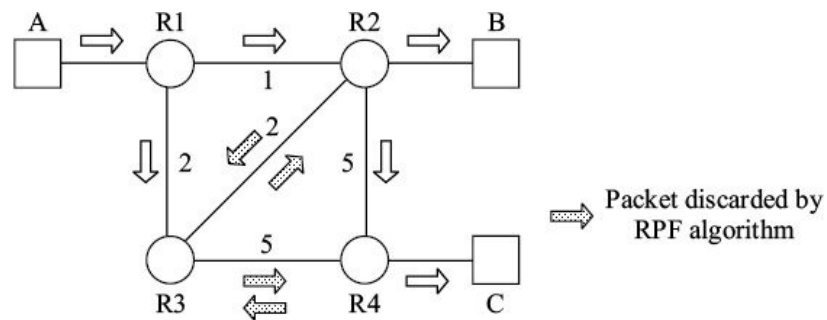


Figure 19.4 Reverse path forwarding (RPF).

1. R1 forwards the multicast packet to R2 and R3.
2. R2 receives the packet on its interface which has the shortest path (cost = 1) to the source A. Therefore, it forwards the packet to R3, R4, and B.
3. R3 receives the packet from R1 on its interface that has the shortest path (cost = 2) to the source A. Therefore, it forwards the packet to R2 and R4.
4. R3 also receives a packet from R2 but this packet is received on the interface which is not in the shortest path to A. The cost of this path to A is 3 compared to the cost = 2 of the shortest path. R3 assumes this is a duplicate packet and discards this packet.
5. R2 also receives a packet from R3 in the similar way. R2 too discards the packet received from R3.
6. R4 receives two packets, one from R2 and the other from R3. The packet from R2 comes via the shortest path to A (cost = 6) and is forwarded to C and R3. The packet from R3 is discarded because it is received on the interface which is not in the shortest path. The packet from R4 to R3 is discarded by R3 on the same grounds as above.

We have assumed that the routers know the shortest path to the source. The forwarding tables, as we know, contain the cost of forwarding a packet to the destination using the shortest path. RPF algorithm, on the other hand, requires the cost of receiving a packet from a given source. If we assume that the path costs are symmetric (that is cost of sending a packet to a given destination is same as cost of receiving a packet from it), the normal forwarding table will be sufficient for this algorithm. It is possible that the link costs are not symmetric, in which case a separate multicast forwarding table is required. If link state routing is used, the LSA database of a router has the needed information to generate this table.

19.3.1 Improved RPF

In Figure 19.4, the packets flowing on links between R1 and R3, R3 and R4, and R2 and R3 constitute extraneous traffic. The routers that received these packets discarded them. RPF algorithm could be further improved to filter part of this extraneous traffic proactively. The routers are required to look a step ahead. The following strategy is adopted for look ahead:

- A multicast packet is forwarded by a router (say R1) to the next router (say

R2) if the router (R1) is on the shortest path between the source (S) and the next router (R2).

In this case the duplicates are not generated at all. Let us examine this with the example given in Figure 19.5.

1. R1 forwards the multicast packet to R2 and R3 because it is in the shortest paths from A to R2 and R3.
2. R2 forwards the received packet to R4 (and B) because the shortest path from A to R4 is via R2. R2 does not forward the packet to R3 because the shortest path between R3 and A is not via R2.
3. R3 does not forward the received packet to R2 and R4, because the shortest paths to R2 and R4 from A are not through R3. R3 merely discards the packet.
4. R4 forwards the packet received from R2 only to C. It does not forward the packet to R3 because the shortest path to R3 from A is not through R4.

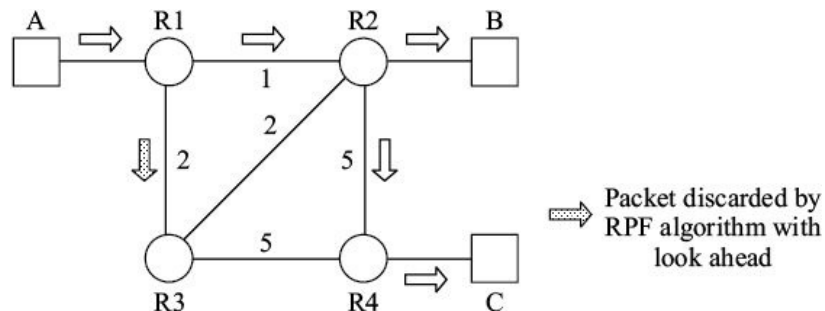


FIGURE 19.5 Improved RPF.

Note that we have filtered a large amount of extraneous multicast traffic. If the routing protocol is based on link state routing (e.g. OSPF), this algorithm can be readily implemented.

EXAMPLE 19.1 Figure 19.6, B is the source and multicast group members are A, B, and C. Show the flow of multicast packets if improved RPF with look ahead is deployed.

Solution The flow of multicast packets is as follows:

- (a) R2 forwards the packet to R1, R3, and R4.
- (b) R3 finds that it is not in the shortest path to R1 (cost metric via R3 is 4) and R4 (cost metric via R3 is 7) and therefore it discards the packet.

- (c) R1 forwards the packet received from R2 to A. It does not forward the packet to R3 because cost metric from B to R3 via R1 is 3, which is greater than the shortest path.
- (d) R4 forwards the packet received from R2 to B. It does not forward the packet to R3 because cost metric from B to R3 via R4 is 10, which is greater than the shortest path.

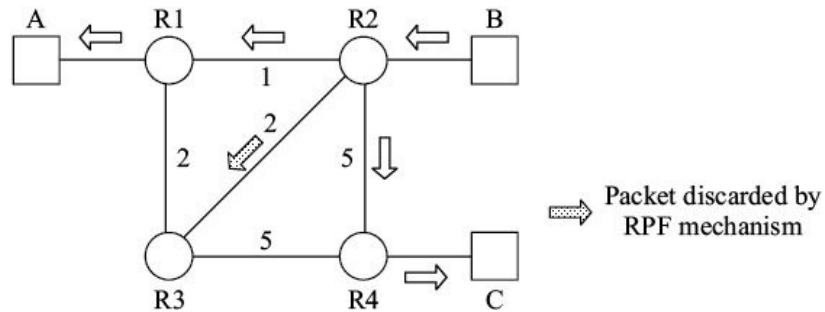


Figure 19.6 Example 19.1.

RPF with look ahead resolves the extraneous traffic problem to a large extent. The remaining problem is group membership. The algorithm discussed so far enables multicast packets to reach all the edge routers of the network. We would like that the multicast traffic of a group touches the only those routers that are connected to the members of the group. In Figure 19.7, there are two multicast groups, $G1 = A, B, C$ and $G2 = D, E, F$. When a source sends multicast IP packets to group $G1$, the IP packets should be delivered only to routers R1, R3, and R4. They should not be delivered to routers R2 and R5. Pruning and grafting are the two methods that achieve this objective.

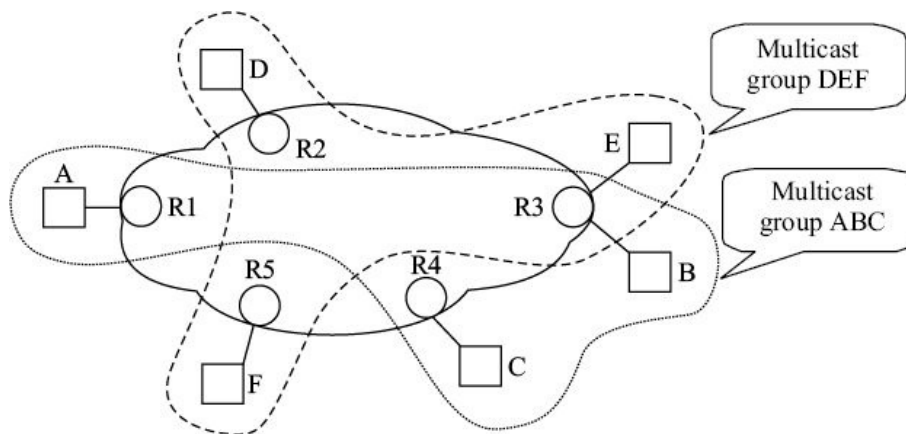


Figure 19.7 Multicast groups.

19.3.2 Pruning

Pruning is an extension of multicast routing mechanisms, which works as follows (Figure 19.8):

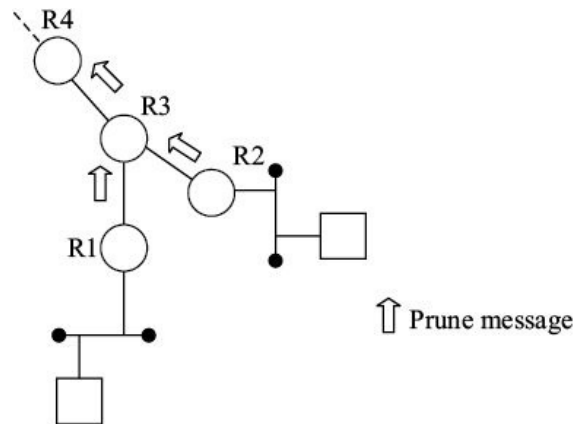


Figure 19.8 Prune messages.

1. The first multicast IP packet bearing multicast group address (say G1) from the source is propagated to all the routers of the network up to the edges.
2. The edge routers (R1 and R2) use IGMP¹ to discover if there are members of the multicast group (G1) attached to them.
3. If there is no group member attached to an edge router, it sends a prune message to the upstream multicast router (R3).
4. The upstream multicast router takes note of the interface through which the prune message is received. It does not send any multicast packets of the group (G1) through that interface after receiving the prune message.
5. If similar prune messages for the group (G1) are obtained at all the downstream interfaces of the upstream router (R3), the router sends a prune message to its upstream router (R4), which thereafter stops sending the multicast packets of group (G1) to the downstream router (R3).

Prune messages can be associated either with a multicast group ($(*, G)$ ² or with a source and a multicast group (S, G). In the first case, the routers need to store only one prune indication per group for each of its downstream interfaces. In the later case, a host is not ready to receive multicast packets for a group from one or more sources. Therefore, the router needs to store prune information with respect to every source and every multicast group for its all downstream interfaces.

There is one more twist. Multicast group members may be mobile or their

interest in receiving multicast may be sporadic. Therefore, the prune status at each interface of the router is to be maintained dynamically. This issue is resolved by giving the prune state of an interface on the router a lifetime. After expiry of the lifetime, the prune state is deleted. Once the prune state is deleted, the next multicast packet is flooded again through the interface. This is called refresh flooding.

If there is change in the network topology (because a link or a router is down), the unicast and multicast algorithms converge dynamically and packets follow new shortest paths. Prune states also age out, and flooding and pruning build new states along new paths.

19.3.3 Grafting

An edge multicast router periodically sends IGMP query on its LAN to its hosts. The replies from the hosts indicate to the router, if there are interested members of a multicast group on the LAN. If there are interested members of the multicast group that was pruned earlier, the router updates the state of its interface connected to the LAN and sends a ‘graft’ message to the upstream router. The graft message deletes the prune state with respect to the particular multicast group at the interface of the upstream router.

19.4 CORE-BASED TREES (CBT)

RPF with flood and prune has two limitations:

- Periodic flooding and pruning
- Prune records are maintained for each source and each multicast group at each interface of the routers.

Core-Based Tree (CBT) approach addresses these limitations. To understand its working, we will consider the example shown in Figure 19.8.

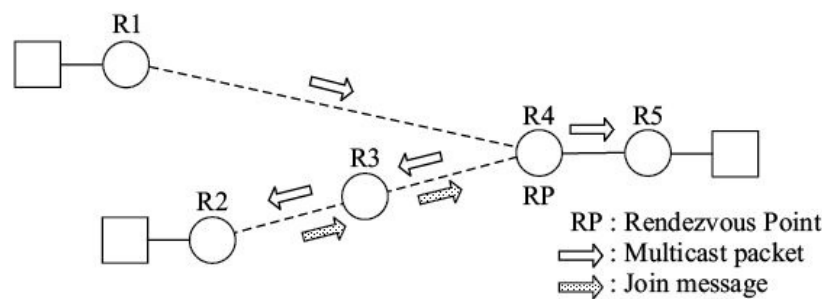


Figure 19.9 Core-based tree.

1. For each multicast group, one of the routers of the network is designated as the core or Rendezvous Point (RP). For example, R4 is designated as RP in Figure 19.9.
2. All the members of a group receive multicast packets from the RP of the group instead of the source.
3. A member may join the group by sending join message towards the RP. The join message is sent over the shortest path to the RP.
4. The intermediate router (R3) that processes the join message, marks the respective interface where join message is received for the multicast group. The intermediate router
 - (a) forwards the join message to the next intermediate router towards the RP;
 - (b) does not forward the join message to upstream router if the multicast path to RP is already active; and
 - (c) sends an acknowledgement to the downstream router.
 Thus with RP as the root, a multicast tree is created for the group.
5. All the multicast sources send their multicast packets to the RP for distribution to the members of the group. The IP packets from the source are encapsulated in another IP header with the destination address of the RP. Usual unicast packet transmission to RP is used.
6. The RP removes the encapsulation and sends the multicast packet to the members of the group along the multicast tree. The same tree is used for distribution of multicast packets irrespective of the source.
7. There can be different RPs for different multicast groups.

CBT has the following advantages:

- The core-based tree remains the same for all the sources. Therefore, state information is only per group (*, G) rather than per source per group.
- The first packet need not be flooded across the network. Refresh flooding is also not required.

CBT on the other hand is not an optimal solution, because same tree is used for all sources of the group. Further, if RP fails, the entire multicast group loses the service.

10.5 MULTICASTING PROTOCOLS

19.5 MULTICASTING PROTOCOLS

Having examined the various approaches to multicasting of IP packets, it is time we look at the multicasting protocols. The three multicasting protocols that have been implemented in the IP networks are:

1. Protocol Independent Multicasting (PIM)
 - (a) PIM dense mode
 - (b) PIM sparse mode
2. Distance Vector Multicast Routing Protocol (DVMRP)
3. Multicast extensions to OSPF (MOSPF).

PIM is the most successful protocol for multicasting. It consists of two protocols—PIM dense mode and PIM sparse mode. We will examine its operation in some detail in this section. A brief overview of DVMRP and MOSPF is given at the end of the section.

19.5.1 Protocol Independent Multicast (PIM)

Protocol Independent Multicast (PIM) gets its name from the fact that it is independent of the IP unicast routing protocol. It uses unicast forwarding table which can be populated using OSPF, RIP, BGP or static routes.

PIM uses different multicast routing strategies depending on whether a multicast group is ‘dense’ or ‘sparse’. A dense group is one where most of the edge multicast routers in the network have the multicast group members attached to them. A sparse group, on the other hand, has its members attached to a few edge multicast routers of the network.

The sparse group is more important from the point of routing strategy. In the dense group, multicast packets are to be sent to almost all the edge routers of the network. In sparse group, large volume of extraneous traffic will be generated if we use controlled flooding. Even if we use prune messages, the prune messages themselves will generate large volume of extraneous traffic.

PIM dense mode (DM). PIM dense mode is specified in draft IETF-IDMR-PIM-DM 05. txt. It is based on RPF with pruning. The first multicast packet is sent using controlled flooding. The receivers not interested in receiving the multicast, send prune messages to the upstream routers. For routing the multicast packets, the unicast forwarding table is used. As explained in the section on

RPF, the path cost metrics cannot be asymmetric.

PIM sparse mode (SM). PIM sparse mode, version 2 is defined in RFC 2362. It is a variant of CBT discussed earlier. The main features of CBT may be summarized as:

- There is common Rendezvous Point (RP) to which all the multicast group members send join messages.
- RP creates a shared tree to the members of the group.
- The first multicast router from the source encapsulates the IP packets in another IP header and sends them to the RP. The RP removes encapsulation and sends the multicast IP packets along the shared tree.

In Figure 19.10, R3 is designated as RP for the multicast group consisting of the members A, B, and C. R1, R2, and R4 send join messages to the RP (Figure 19.10a). These join messages are for receiving multicast from any source through the RP. End system A acting as a source sends its multicast IP packets with multicast group address to R1. R1 encapsulates the packets and sends them to the RP (Figure 19.10b). RP removes the encapsulation and sends the IP packets to B and C along the shared tree created by the flow of join messages.

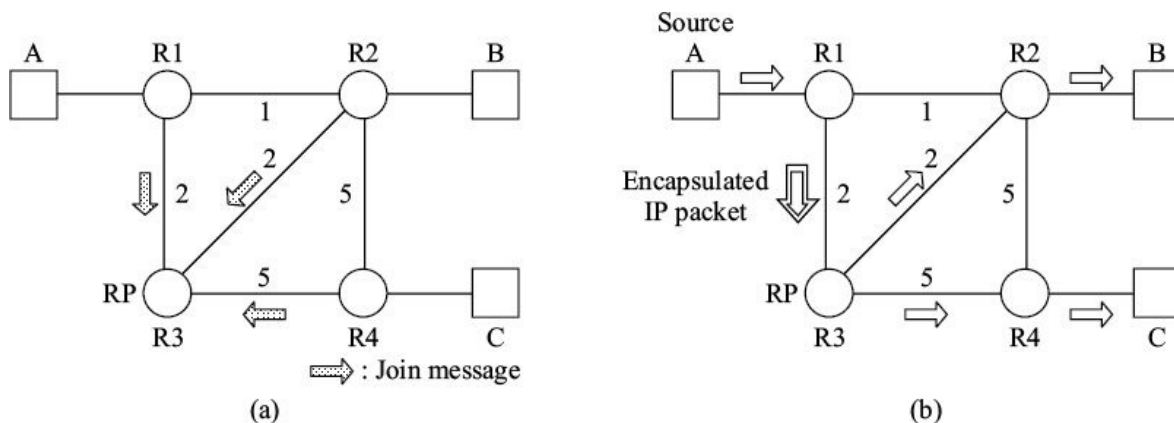


Figure 19.10 PIM sparse mode.

The same tree with RP at the root is used if B were the source instead of A. In that case R2 would have sent the encapsulated multicast packets to the RP (R3). R3 would have send the multicast packets to all the members of the group. In other words the tree from RP is shared tree independent of the source.

PIM sparse mode introduces the following extensions to CBT method:

- If the traffic justifies, the RP can move the point of removal of

encapsulation towards the source. This is done by sending a join message towards source (Figure 19.11). Encapsulation is totally avoided if the point of removal of encapsulation moves up to the point of encapsulation, which is the first router from the source. Note that the multicast packets are still distributed by the RP. The only gain is that the overhead of encapsulation is avoided.

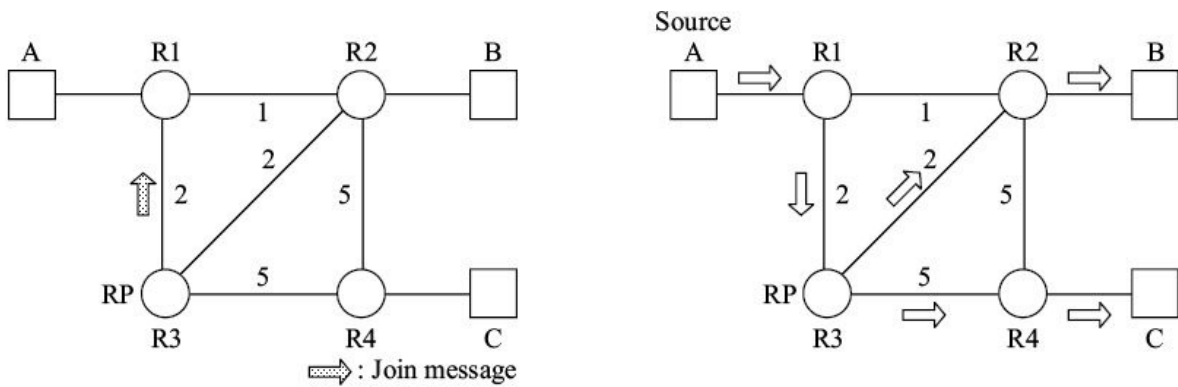


FIGURE 19.11 PIM extensions (1).

- The first router of a multicast receiver can force building a source-based tree if it finds that the source is nearer to it than the RP. In Figure 19.12, R2 finds that its distance to the source via R1 is shorter than that via the RP. Therefore, R2 sends a join message to R1 and a prune message to RP simultaneously. Thereafter, it starts receiving the multicast packets directly from R1. The RP stops forwarding the IP packets to R2 on receipt of the prune message. The actual implementation of the mechanism ensures that no packet is lost during this transition.

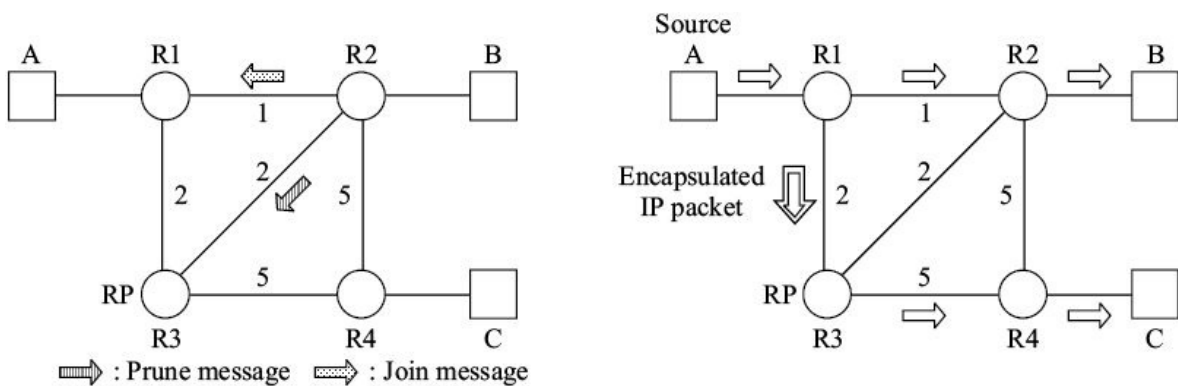


Figure 19.12 PIM extensions (2).

Thus we can have source tree in addition to the shared tree from the RP. The

source tree emanates from the first router of the source and serves the RP and those receivers who are nearer to the source than to the RP. Note that source tree is source and group specific (S, G) and the shared tree of RP is group specific (*, G).

19.5.2 DVMRP and MOSPF

DVMRP is defined in RFC 1075 and IETF-IDMR-DVMRP-V.3-04.txt. It consists of two components:

- Conventional distance vector routing protocol for building a DVMRP forwarding table for all the sources of multicast traffic. The multicast traffic is forwarded along the source specific tree as represented by the DVMRP forwarding tables.
- Truncated RPF with pruning and grafting mechanism for avoiding the traffic on unnecessary branches of the tree.

Thus we have two distance vector routing protocols, one for multicast and the other for unicast traffic. There are two forwarding tables, one for multicast traffic and the other for unicast traffic. DVMRP can handle asymmetric path cost metrics. It has inherent problem of first packet flooding followed by periodic flooding to trigger pruning messages.

MOSPF is defined in RFC 1584. OSPF is a link state routing protocol and extending a link state protocol for multicasting is relatively simpler because each router has topological database of entire network. A new type of link state record, group membership record is added to the database. MOSPF uses RPF with pruning for multicasting. It does not require any additional multicasting routing protocol. However, it can handle asymmetric path cost metrics.

19.6 ADDRESSING IN IP MULTICAST

Unicast and multicast IP packets differ only in one respect, the destination address field. A multicast IP packet has multicast group address in the destination address field. Multicast group addressing scheme is a separate class in IP addressing scheme. Class D IP addresses are reserved for multicast groups.

A class D IP address is identified by the first four bits which are always 1110 (Figure 19.13). These bits are followed by the multicast group address. This

gives IP address range from 224.0.0.0 to 239.255.255.255. Note that this address range is only for the group addresses used as destination addresses in multicast traffic. The source address of multicast IP packet is the unicast IP address of the source.

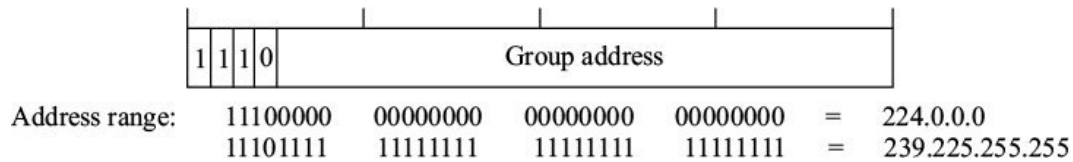


Figure 19.13 IP group addresses for multicast.

Class D multicast address range caters to multicast requirements of various network protocols and of the users. Range of multicast addresses for various applications has been assigned by IANA. RFC 1700 gives full list of multicast addresses. Some important reserved ranges are given below.

Addresses for network protocols on local networks. IANA has reserved addresses in the range 224.0.0.0 to 224.0.0.255 for use by the network protocols in the local networks. Packets with these addresses are never forwarded outside the LAN segment. These packets have TTL of 1. These addresses are used for automatic router discovery and exchange of routing information. For example, OSPF uses 224.0.0.5 to exchange link state information. Some well-known multicast addresses assigned by IANA for specific applications are:

- 224.0.0.1 All systems on this subnet
- 224.0.0.2 All routers on this subnet
- 224.0.0.4 DVMRP routers
- 224.0.0.5 OSPF routers
- 224.0.0.9 RIPv2 routers
- 224.0.0.13 PIM routers
- 224.0.0.15 CBT routers

Globally scoped multicast addresses. The range from 224.0.1.0 through 238.255.255.255 is called globally scoped addresses. These are used for multicast data on the Internet. Some of the addresses from this range are reserved for specific applications. For example, 224.0.1.1 is used for Network Time Protocol (NTP).

Limited scope multicast addresses. The addresses in the range from 239.0.0.0 through 239.255.255.255 are used within a user defined domain. This allows

reuse of the same address in different domains. Multicast packets are restricted within a defined domain by configuring filters in the router.

19.6.1 Multicast on LAN Segments

Multicast of the user traffic on the local area network can be done either using the layer-2 broadcast address or using layer-2 multicast address. The broadcast address is 0xFF-FF-FF-FF-FF-FF for IEEE 802 LAN. In this case all the stations on the LAN receive the multicast packets and process the layer-2 frames. After retrieving the multicast IP packet, they accept the IP packet if they are members of the multicast group.

Alternatively, the I/G (Individual/Group) bit of the MAC destination address (Figure 11.11 of Chapter 11) can be used to indicate that the address is multicast group address. I/G bit is set to 1 for multicast addresses. This is applicable to both Ethernet and token ring LANs.

Recall that in a MAC address

- octets 0, 1, and 2 of the MAC address are assigned to vendors for the vendor code, and
- octets 3, 4, and 5 of the MAC address are used by the vendors for numbering their network components.

IANA owns a block of MAC addresses for multicasting. The first three octets (octets 0, 1, and 2) for the multicast frames containing IP packets are 0x01-00-5E for Ethernet LANs and 0x01-00-7A for token ring LANs (Figure 19.14). Canonical format of byte representation has been used in the figure, *i.e.* the least significant bit of each octet is on the right hand side. The I/G bit of the first octet is set to 1 to indicate that it is a group address. The next two octets contain 0x00 5E indicating that it is the block owned by IANA for Ethernet. Out of the remaining 24 bits, 23 bits are used for multicast group address and one bit is reserved.

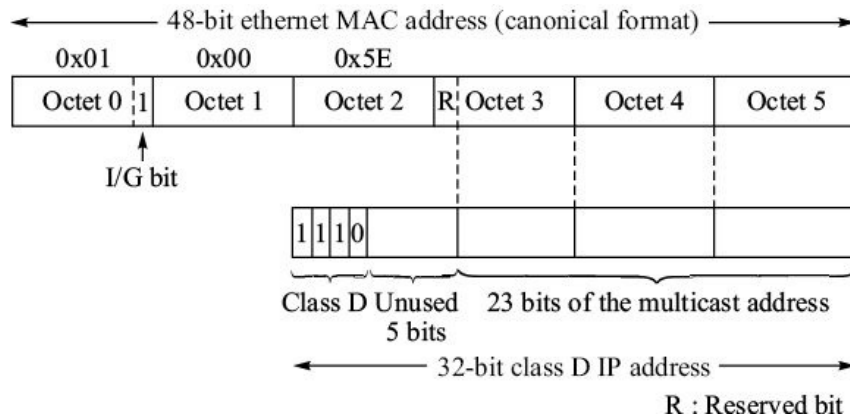


Figure 19.14 Mapping of IP multicast group address to MAC destination address.

In the IP multicast address of 32-bits, only 23 bits are used for the multicast group address. The five bits between 1110 and the 23-bit multicast group address are not used (Figure 19.14). 23-bit group address is mapped to the octets 3, 4, and 5 of the MAC destination address as shown in the figure. One remaining bit of octet 3 is not used.

19.7 INTERNET GROUP MANAGEMENT PROTOCOL (IGMP)

Internet Group Management Protocol (IGMP) is a protocol between the multicast router at the edge of a network and a host connected to it on a local network (Figure 19.15).

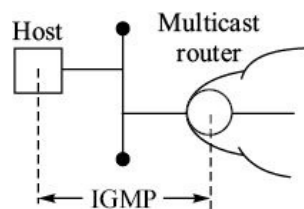


Figure 19.15 Internet Group Management Protocol (IGMP).

IGMP is used by the hosts on a LAN to dynamically report their multicast group membership to the neighbouring multicast router. The multicast router listens to IGMP messages from the hosts and periodically sends out queries on the LAN to discover which group members are active or inactive. IGMP is an integral part of IP multicast as ICMP is integral part of IP unicast.

IGMP messages are sent encapsulated in IP packet. IP protocol type is set to 2 for IGMP messages. The currently used versions of IGMP are Version 1 (RFC 1112) and Version 2 (RFC 2236). Version 3 is in draft state.

19.7.1 IGMP Version 1

19.7.1 IGMP VERSION 1

Format of IGMP message is shown in Figure 19.16. Its various fields are as follows:

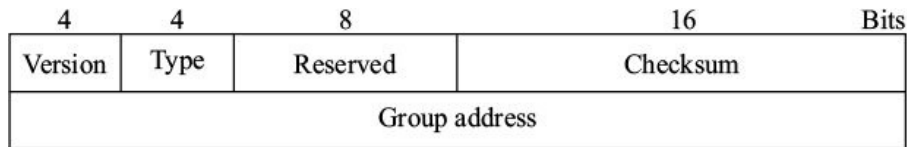


Figure 19.16 IGMP Version 1 message format.

Version (4 bits). It indicates IGMP version number. For version 1, this field is set to value 1.

Type (4 bits). It indicates type of message, query or report.

Type 1: Query from the multicast router.

Type 2: Report from the host.

Checksum (16 bits). It contains the checksum computed in the same manner as in IPv4.

Group address (32 bits). It contains the multicast group address in the report. It is set to all 0s in the query message.

Operation of IGMP Version 1 involves the following steps:

1. The multicast router sends periodically on the LAN segment IGMP query message encapsulated in an IP packet using destination IP address as the multicast address 224.0.0.1 with TTL 1. Recall that this is a reserved multicast address for all systems on this subnet. The query is sent periodically at interval of 60–90 seconds.
2. The hosts send back their report-messages to the router indicating their group memberships. The IP packet containing the report-message bears the IP destination address equal to the group address being reported and TTL of 1. Separate group membership reports are sent generated for each group.
3. To reduce the traffic that would be generated as response to the query on the LAN, each host starts a report delay timer for each of its group membership report. The delay timer value is chosen randomly between 0 to 10 seconds. When the timer for a group (say group G) expires, a host sends its report for the corresponding group. There can be other waiting hosts on the LAN, who are also members of group G. After hearing this report-message, they do not send their reports for group G since even one report

from any member is sufficient for the multicast router to forward the group G multicast traffic on the LAN. Thus, in general, only one report per group will be generated in response to each query.

4. If a member wants to join a new group without waiting for the query message, it can send an unsolicited report to the multicast router.
5. If there is no response to three IGMP queries, the multicast router concludes that there are no multicast group members on this LAN.

19.7.2 IGMP Version 2

IGMP Version 2 differs from Version 1 on the following accounts:

- IGMP Version 1 has high ‘leave latency’. If a group member wants to leave the group, it stops replying to the query messages sent by the multicast router. When no reply is received to three queries, the router stops forwarding the multicast traffic pertaining to that group. The process takes several minutes. IGMP Version 2 includes a new message for leaving a group. Whenever a host wants to quit a group, it sends the leave message to the multicast router.
- IGMP Version 2 introduces group specific query to detect if there is any member of a particular group.

IGMP Version 2 is backward compatible with IGMP Version 1. Its message format is shown in Figure 19.17. The message has the following fields:

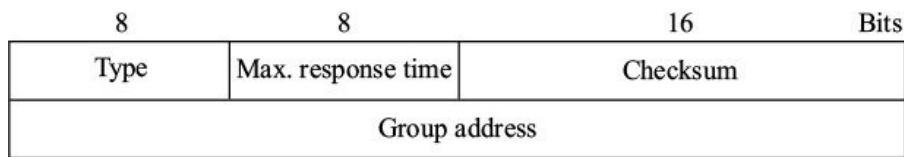


Figure 19.17 Format of IGMP Version 2 message.

Type (8 bits). It indicates the type of message. In order to make Version 2 compatible with Version 1, this field is given the following values:

- 0 x 11 Membership query:
 - General query containing group address field equal to zero.
 - Group specific query containing the address of the group.
- 0 x 16 Version 2 membership report.
- 0 x 17 Leave report.

0 x 12 Version 1 membership report.

Max. response time (8 bits). It is used in the membership query to specify the maximum allowed timeout for the report delay timer. It is specified in units of 1/10 seconds.

19.7.3 IGMP Version 3

IGMP Versions 1 and 2 do not have registration process for the sources. A host can send a multicast to a group without being member of the group. This makes spamming of multicast groups very easy. IGMP Version 3 allows the receiving hosts to specify the sources from which they want to receive the multicast IP packets. The traffic from the other sources is blocked by the multicast routers.

IGMP Version 3 has two types of messages, membership query and report. Membership query has three sub-types:

1. General query
2. Group specific query
3. Group and source specific query.

Queries 1 and 2 are similar to those in Version 2. Query 3 is used to learn if any host is interested in receiving multicast IP packets sent to the specified multicast address and from the sources specified. The receiving host can select the sources and indicate them in its reply.

19.8 MULTIPROTOCOL LABEL SWITCHING

The IP technology uses destination based routing. Routing protocols determine the forwarding table based on various routing algorithm. When an IP packet is received, the router

- checks for errors using the checksum in the header,
- undertakes the longest match of destination address in the forwarding table to determine the outgoing interface, and then
- forwards an IP packet through that interface.

This process is repeated at each router till the IP packet reaches the destination. Hop-by-hop processing for determining the next hop gives an IP network its robustness. If a link goes down, the router forwards the packet through the next best route.

The datagram approach described above, however, has the following limitations:

1. For some applications (e.g. digital voice and video) we would like that the IP packets follow the same path.
2. The routing and forwarding mechanism of datagram is slower compared to other technologies based on virtual circuit approach (e.g. ATM).
3. The datagram service is best effort service. Therefore, quality of service cannot be guaranteed.
4. All the datagrams tend to take the best route and therefore cause congestion on the best route even though alternative paths may be lying unutilized in the network. Recall that the routing protocols that we studied in the last chapter were based on least path cost and destination. Therefore, traffic engineering, which deals with mapping traffic flows along desired paths, is not possible.

The above limitations are readily overcome by the virtual circuit approach adopted in ATM and X.25 networks. Therefore, a new protocol, Multiprotocol Label Switching (MPLS), was developed that makes the IP packets take defined paths which are like a virtual circuits. MPLS does not replace IP. It attaches an additional label on an IP packet, that determines its path through the network.

Multiprotocol Label Switching (MPLS), as the name suggests, can be used for many protocols not just IP. But it is used today exclusively in the context of IP. We will first examine the operation of MPLS and see its applications later.

19.8.1 Basic Approach of MPLS

Router performs two basic functions routing and switching:

- *Routing*. Determination of routes to various destination and creation of forwarding tables based on IP addresses.
- *Switching*. Forwarding the IP packets based on the forwarding table.

MPLS separates these two functions. Routing function is based on IP

addresses as before. The switching function is based on MPLS labels that are attached to IP packets. Labels are just like logical channel identifiers that we came across in ATM and X.25 networks. Figure 19.18 shows a generic schematic of internal processes of an MPLS router. Note that label management and switching functions are add-on functions in a router. We need additional Label Distribution Protocol (LDP) for maintaining label switching tables.

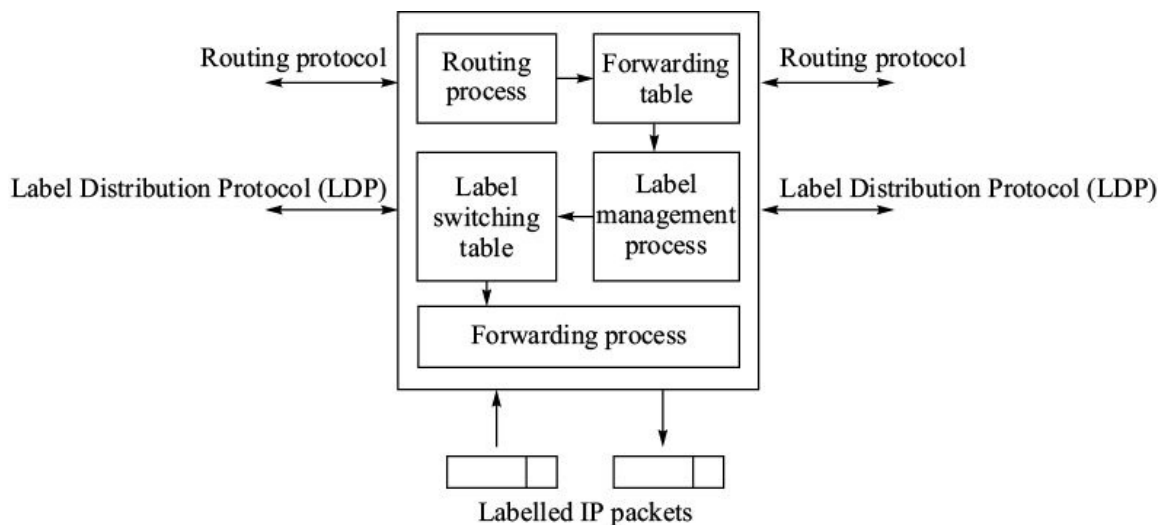


Figure 19.18 Separation of routing and forwarding functions in MPLS.

MPLS enabled IP network consists of Label Edge Routers (LERs) and Label Switching Routers (LSRs) as shown in Figure 19.19. LER adds labels to the incoming IP packets from the customer. Labelled IP packets are sent over virtual path called Label-Switched Path (LSP) in MPLS. The labels have local significance across the link between two routers (just like logical channel number in X.25).

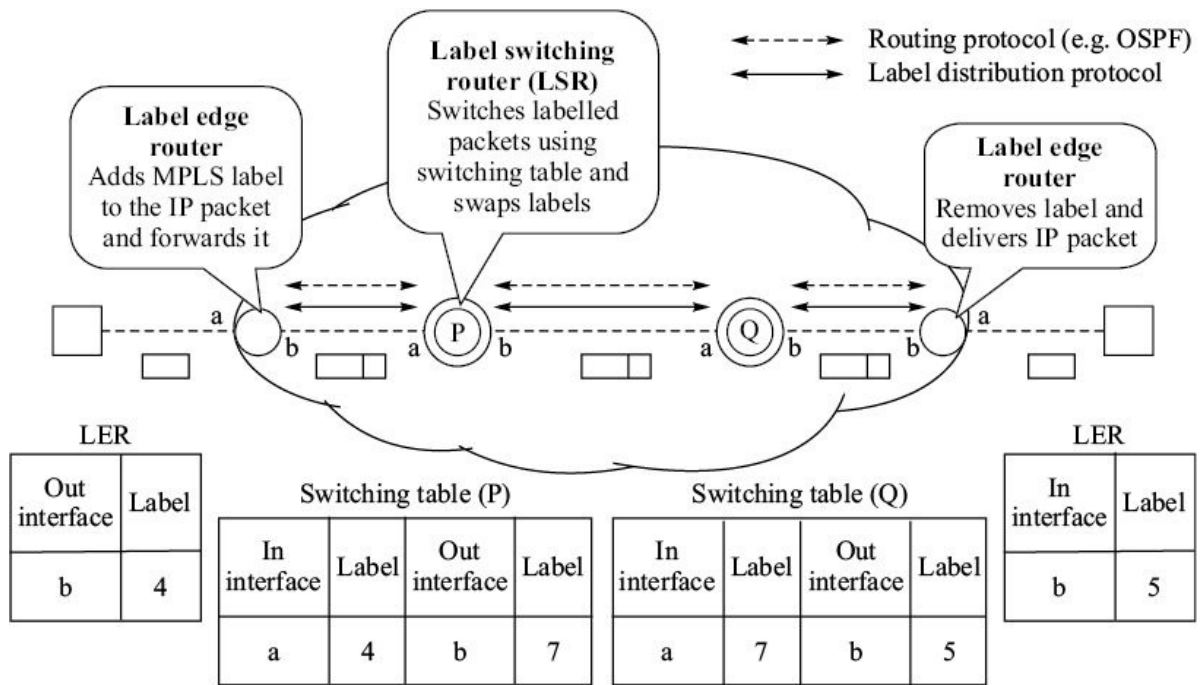


Figure 19.19 Forwarding and label swapping.

The received labelled IP packets are switched to the outgoing port by an LSR. Label switching table is used for this purpose. The IP packet is given a new label on the outgoing link. Note that for forwarding a labelled IP packet only the label is sufficient at an LSR. The IP address is not required. The IP address is required at the LER where a label is attached to the IP packet for the first time.

An LSP is unidirectional. One LSP is required for the packets sent by the source (A) towards a specific destination (B) and another LSP is required for the packets sent by B to A.

Recall that forwarding IP packets required finding ‘longest match’ of the destination address of an IP packet and the entry in the forwarding table. MPLS requires exact match of labels and therefore forwarding action is faster in MPLS.

19.8.2 MPLS Header

The MPLS label is in the form of a header called MPLS header. It is prefixed to the IP packet before adding the layer 2 header. It consists of 32 bits and has four fields (Figure 19.20a).

Label (20 bits). This field carries the actual value of the label.

Experimental (Exp, 3 bits). This field can be used for specifying MPLS class of service.

S bit (1 bit). As shown in Figure 19.20b, there can be a stack of MPLS headers.³ This field points to the oldest label in the stack, which has S bit set to 1. The oldest label is at the bottom of the stack which is just before the IP header. The S bit is zero in the other labels.

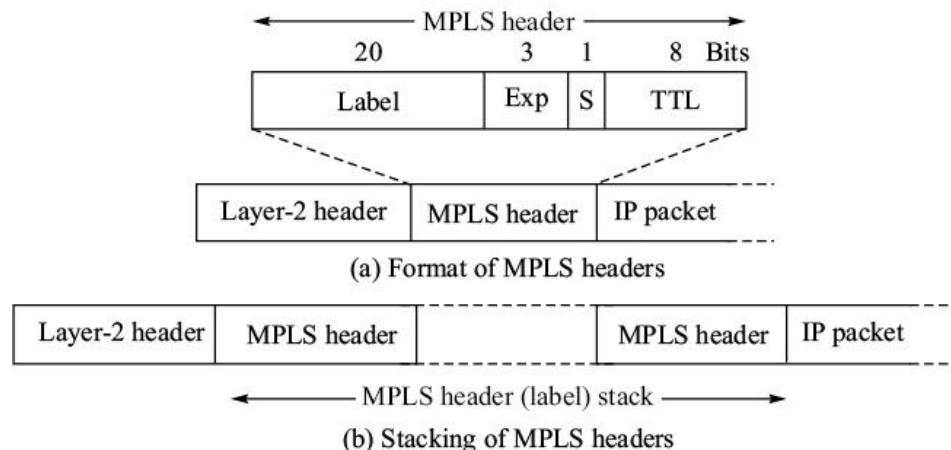


Figure 19.20 MPLS header.

TTL (3 bits). This field indicates Time to Live (TTL). At the ingress LER, this field is copied from the corresponding TTL field of the IP packet. At each LSR, it is decremented by one in the outgoing labelled IP packet. At the egress LER, the remaining TTL value is transferred to the corresponding field of the IP packet. The packet is discarded whenever the TTL value becomes zero.

19.8.3 Forwarding Equivalence Class (FEC)

From the previous discussion, it is clear that labels on IP packets enable their transport on defined paths, called LSPs. The purpose of defining paths for IP packets is to ensure certain operational and quality of service requirements, which cannot be guaranteed otherwise on the connectionless IP network. Thus, the labels have associated significance called Forwarding Equivalence Class (FEC). A label identifies a packet as belonging to the class of packets which take the same LSP and get the same forwarding treatment in the MPLS network as defined for the FEC associated with the label. An FEC can be defined for

- class of packets having a particular destination network prefix.
- class of multicast packets having same source and destination group addresses.
- class of IP packets having a particular destination prefix and a particular Type of Service (TOS).

19.8.4 Label Distribution Protocol (LDP)

Assignment of an IP packet to an FEC is done just once by the MPLS edge router at the ingress to the MPLS network. Within the network, LSRs have label bindings to an FEC. These bindings are created as follows (Figure 19.21):

1. Downstream router R4 selects a label L1 for FEC (F) for the flow of packets to B. It advertises this binding to its neighbours.
2. The neighbour R3 takes note of this binding and selects a label L2 for this FEC (F). It advertises this binding to its neighbours.
3. R2 repeats the above process and conveys its selected label L3 with the FEC (F) to its neighbours that include MPLS edge router R1. Thus an LSP from R1 to R4 for the packets meant for destination B gets created. This LSP is for the transport of packets from R1 to R4.
4. LSPs for the destination B get created in the similar manner from all the other MPLS edge routers that receive these advertisements.

The label distribution is always from the downstream router to the upstream router. There are two modes:

- Downstream-on-demand
- Unsolicited downstream.

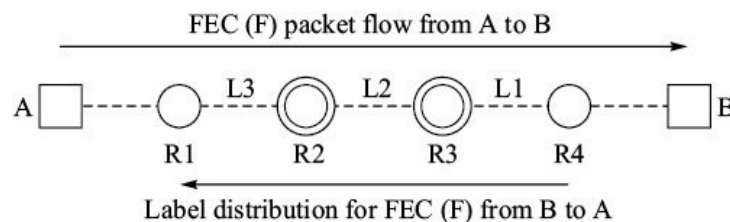


Figure 19.21 Label distribution.

In the first case, the ingress LER (R1 in Figure 19.21) requests for a label from its downstream neighbour LSR (R2) for a specified destination. The request is further passed onto the next downstream neighbour LSR until the egress LER (R4). The egress LER (R4) sends its label binding to the upstream LSR (R3), which in turn sends its binding to the next upstream LSR till the last binding reaches the ingress LER (R1) that originated the request. In the second case, the downstream routers initiates this process on its own. The protocol used for sending requests and label bindings is Label Distribution Protocol (LDP). We

will not go into further details of this protocol.

19.8.5 Other Methods of Creating LSPs

The label distribution process described above makes use of the forwarding table for sending IP packets containing the label and FEC bindings. Therefore, the LSPs that are created are based on the shortest paths as dictated by the routing protocol. But many times we would like to define the route of an LSP. This path may not be the shortest path between the two points. Therefore, LDP is not suited for such applications. The protocols used for setting up an LSP along a specified path are:

- Resource Reservation Protocol (RSVP)
- Constraint-based Routing LDP (CR-LDP).

RSVP predates MPLS and was intended as the protocol for creating bandwidth reservations for traffic flows. It was later extended so that it could be used for setting up an LSP along a specified path, reserve bandwidth resources for that LSP and to distribute MPLS labels.

CR-LDP is an extension of LDP which creates an LSP that must traverse specified sequence of transit nodes of the network. In Figure 19.22, the LSP between R1 and R6 is constrained to pass through R3 and R4. CR-LDP creates the LSP R1-R2-R3-R4-R6. For path sections between the adjacent specified nodes (e.g. R1 and R3, R3 and R4, R4 and R6), CR-LDP makes use of the routing table. CR-LDP implementations are, however, very few in the industry.

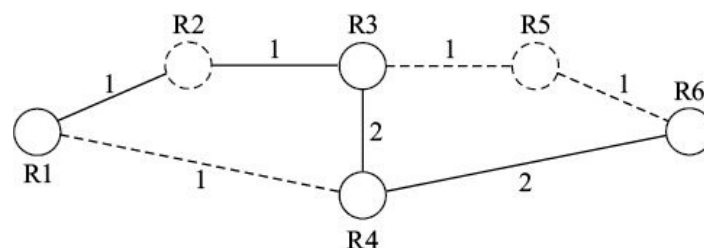


Figure 19.22 Constraint based LDP.

19.8.6 MPLS for Traffic Engineering

The interior gateway routing protocols in an IP network route the packet on the shortest path between two points in the network. This is done irrespective of the traffic flow on the various interconnecting links between the two points. At times, we want to deviate from the shortest path strategy because the shortest

path may not have enough capacity to carry the entire traffic due to its bandwidth limitations. The task of mapping traffic flows on the network topology is termed as traffic engineering.

Figure 19.23, shows a simple IP network working on metric based traffic engineering. R1 and R2 send large volume of data to R6. The path taken by the IP packets to their destination goes through R5 because the overall path cost is lower. We would like to share the traffic between the paths through R4 and R5. For example, IP packets from R1 may take path through R4 and those going from R2 may take path through R5. If we raise the metrics along the path though R5, both the flows to R6 will shift to R4.

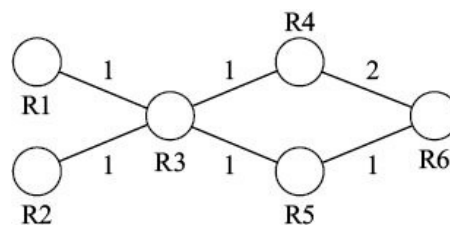


Figure 19.23 Traffic engineering using metrics.

Traffic engineering allows us to move the traffic flows along the desired paths which may not be the shortest. In a complex network, it is almost impossible to manipulate the metrics to achieve the desired goals of traffic engineering, as we saw above.

MPLS builds the capability of traffic engineering in an IP network by identifying the traffic flows by the labels and creating explicit routes (LSPs) for various traffic flows. For example, in Figure 19.23, we can define separate LSPs for the traffic flows between R1 and R6, and between R2 and R6. The LSP between R1 and R6 is created along the path R1-R3-R4-R6. The LSP between R2 and R6 is created along the path R2-R3-R5-R6.

Creation of LSPs along a specified path is done using RSVP as addressed briefly earlier. Explicit routing can be used for building resilience in the network as well. We can define alternate LSPs if an LSP fails.

19.8.7 MPLS Tunnels

So far, we have assumed that the user data is contained in IP packets and MPLS is used transporting these IP packets efficiently across the IP network. It might have been noted that in an MPLS enabled the IP network, an IP packet is transported as a chunk of octets. LSRs do not look at the IP header at all. Labels on the IP packet are sufficient for forwarding the packets to the destination.

Therefore, if we replace an IP packet with another type of data unit (e.g. ATM cell or Ethernet frame) and put the MPLS label on it, the MPLS-enabled IP network should transport the data unit to the egress LER. Thus, an LSP can be conceived as a virtual tunnel across the IP network and we can send any type of data units across the tunnel (Figure 19.24).

There is one more issue we need to address in the scheme described above. The egress LER should know how to treat the payload behind the MPLS label. Therefore, the ingress LER attaches an identifier that specifies the payload. The egress LER removes the MPLS label, examines the identifier, and processes it accordingly. MPLS tunnels are effectively used to build layer-2 Virtual Private Networks (VPN). In Figure 19.24, the two LANs are interconnected using MPLS tunnel and form a virtual layer 2 network.

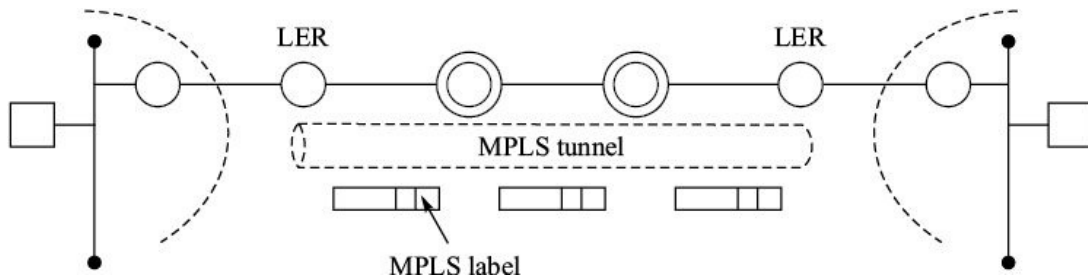


Figure 19.24 MPLS tunnel.

Structure of typical labelled MPLS packet is shown in Figure 19.25. The sequence number is optional. It is used for ensuring sequenced delivery of the packets. The structure of layer-2 frame contained in MPLS packet is slightly modified. The modification depends on the type of frame. In frame relay, the DLCI octets are removed. DLCI is mapped to MPLS label at ingress PE router and the label is mapped back to DLCI at the egress PE router. In case of PPP, HDLC, and Ethernet frames, the flags, preamble, and CRC checksum octets are removed before they are given MPLS labels.

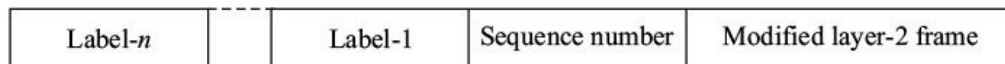


Figure 19.24 Encapsulation of layer 2 frame in MPLS packet.

19.8.8 Label Stacking

The tunneling concept of MPLS described above is one of its most powerful features. Several LSPs can be bundled through another LSP that acts as a tunnel (Figure 19.26). Each LSP has its associated label. Another label pertaining to the tunnel is stacked on already labelled IP packet. The topmost label is processed

within the tunnel. At the exit of the tunnel, this label is removed and thereafter next label is used for the further course of forwarding the IP packet. The label at the bottom of the stack is identified by the S bit of MPLS header (Figure 19.20b). It is set to one in the oldest label.

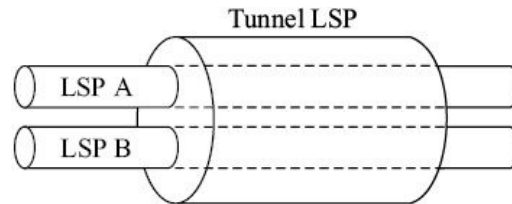


Figure 19.26 Bundling of LSPs using MPLS tunnels.

SUMMARY

In this chapter we explored two very specialized mechanisms for directing the flow of IP packets, multicasting and Multiprotocol Label Switching (MPLS). In multicasting, an IP packet is delivered to a selected group of destinations simultaneously. Typical application of multicasting is videoconferencing. There are two basic multicasting methods—reverse path forwarding and core-based tree. In Reverse Path Forwarding (RPF), a router forwards a multicast packet only if it receives the packet on its port to the shortest path to the source. In Core-Based Tree (CBT), the source sends its multicast packets to a core router called RP (Rendezvous Point) from where the multicast packets are sent on a spanning tree to the destinations. CBT reduces the requirements of having separate spanning tree for each source. Internet Group Management Protocol (IGMP) is used by the edge multicast router to determine whether there are members of a multicast group attached to it.

Distance Vector Multicast Routing Protocol (DVMRP), Multicast extensions to OSPF (MOSPF), and Protocol Independent Multicast (PIM) are the three multicasting protocols. Out of the three PIM is the most important. It consists of two protocols—PIM Dense Mode and PIM Sparse Mode. PIM Dense Mode is based on RPF and PIM Sparse Mode is based on CBT.

Multiprotocol Label Switching (MPLS) tries to combine the properties of virtual circuit switching with robustness of datagrams. In MPLS, labels are attached to IP packets. The routers forward a packet based on its labels and swap the labels while switching the packet. The labels enable defining Label-Switched Path (LSP) for the packets bearing same label. An LSP is set up using the IP routing protocols and Label Distribution Protocol (LDP). Alternatively explicit path can be defined using RSVP or CR-LDP.

MPLS builds the capability of traffic engineering in an IP network by identifying the traffic flows by the labels and creating explicit routes (LSPs) for various traffic flows. Explicit routing can be used for building resilience in the network as well. We can define alternate LSPs if an LSP fails.

EXERCISES

- In Figure E19.27, sources S1 and S2 send multicast packets to the multicast group consisting of end systems A, B, C, D, and E. Show the shortest-path multicast trees for each source. Assume unit cost of each interconnecting link.

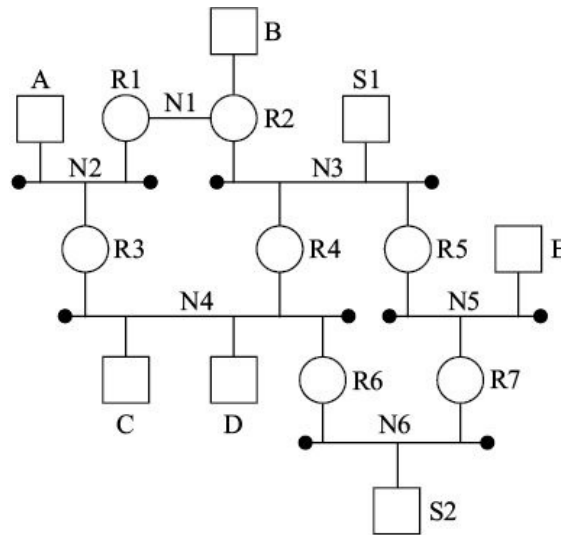


Figure E19.27.

- In the previous figure, S1 is the source and reverse path forwarding is used. How many packets are generated by the routers for each multicast packet sent by the source?
- In Figure E19.28, reverse path forwarding mechanism is used for multicasting the packets sent by A to B, C, and D. Show the flow of packets generated by the routers. How many packets are generated by the routers?

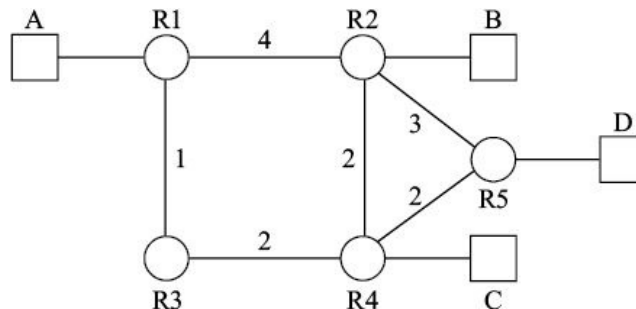


Figure E 19.28.

4. Repeat Exercise 3 if reverse path forwarding with look ahead mechanism was used in place of plain reverse path forwarding.
 5. In Exercise 3, if controlled flooding was used for multicasting, how many packets will be generated by the routers if A is the source?
 6. In Exercise 3, if core-based tree mechanism is used for multicasting with core as R3, show the flow of packets with A as the source.
 7. How would you extend the core-based tree mechanism to allow multiple cores for building resilience?
 8. What will happen if packets are not periodically flooded in DVMRP?
 9. Why IGMP specifies that query messages are sent with TTL = 1?
 10. MPLS labels are 20 bits long. Explain why 20 bits provide enough labels when MPLS is used for destination based forwarding.
 11. MPLS is claimed to improve router performance. Explain the factors that can result in improved router performance.
 12. Internet protocol includes optional field for source routing. What can be the advantages of using explicit routing using MPLS over IP source routing option?
- 1 IGMP is a protocol for discovering members of a particular multicast group on a LAN. We discuss IGMP later in the chapter.
- 2 (*, G) is used for indicating 'any source' and per group. (S, G) is used for indicating per source per group.
- 3 We will discuss the significance of the stack later.

20

Transport Layer

The transport layer is the fourth layer from the bottom in the OSI reference model of the layered network architecture. It sits on the network layer. Unlike the lower layers that exist in all the end systems and network elements (routers, switches), the transport layer and the layers above it are part of end systems only. The basic function of transport layer is to ensure transport of data units from one end system to another. It takes care of flow control, error control, and transport quality of service end-to-end.

There have been three important transport protocols—Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Transport protocol of OSI. TCP and UDP are widely used in the IP-based networks. OSI transport protocol has few implementations. Therefore, we restrict the scope of this chapter to description of TCP and UDP. We begin with the basic tasks performed by the transport layer. Then TCP is described in detail including its congestion avoidance mechanisms. This chapter is concluded with discussion UDP which is a very simple but important protocol.

20.1 TRANSPORT LAYER

The transport layer is situated above the network layer in an end system. While the network layer and the other lower layers reside in all the end systems and intermediate systems (network nodes, routers), the transport layer is implemented only in the end systems (Figure 20.1). The interactions between peer transport layer entities are, therefore, end-to-end and these are made possible by the data transfer service provided by the network layer.

Figure 20.2 shows the two important layered network architectures—OSI and TCP/IP suite. In the OSI architecture, the transport layer is sandwiched between the session and network layers. The network layer of OSI can be X.25 or ISO

CLNP.

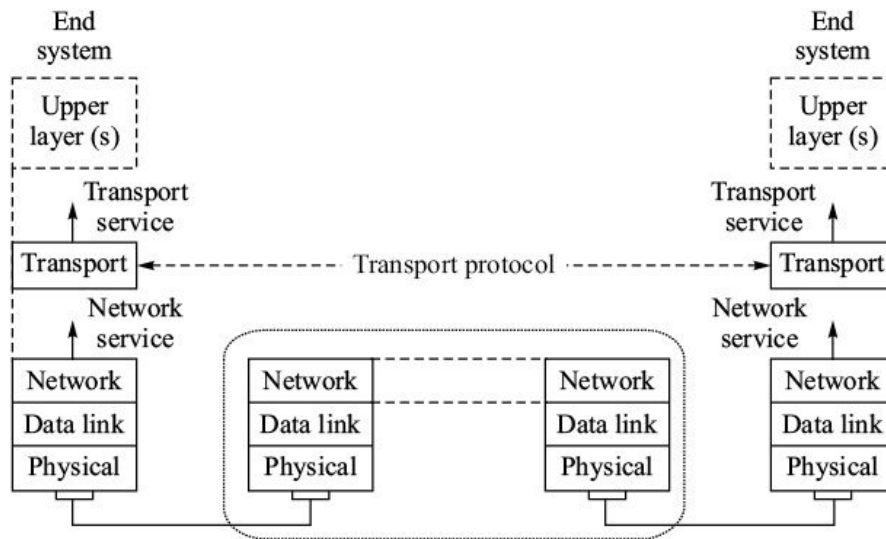


Figure 20.1 Transport layer.

In the TCP/IP suite, the transport layer has two different protocols—TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) as shown in Figure 20.2. The upper layer is application layer and the layer below is IP layer. TCP and UDP are the two most widely used transport layer protocols which are described in detail in this chapter. OSI transport layer protocol is well documented but there are few implementations.

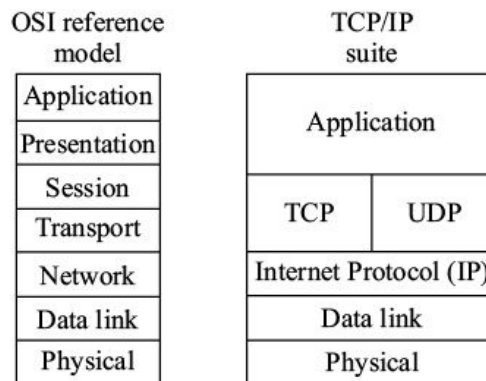


Figure 20.2 Layered architecture of OSI and TCP/IP suite.

20.1.1 Purpose of Transport Layer

The goal of the transport layer is to abstract the structure and function of the network so that the upper layer can communicate without needing to consider the network technology. The transport layer provides transparent, reliable, and cost effective transfer of data units between the upper layer entities in the end

systems. Transparency implies that the transport layer does not place any restriction on the content of the data units received from the user entities. Reliable, in this context, implies that the data units are delivered to the user entities as they are received from them. If an error occurs, the same is either corrected or reported to the user. Cost effectiveness implies that the underlying network service is utilized in the most optimum way to provide the quality of service requested by the users.

20.1.2 Types of Transport Service

Two basic types of transport service are possible:

- Connection-oriented transport service
- Connectionless transport service.

Connection-oriented transport service. In the connection-oriented service, a transport connection is established between the user entities on their request. Transfer of the user data units takes place on this connection. The following capabilities are offered to the connection-oriented transport service users.

Transport connections establishment and release. The transport service user can establish and release a transport connection with another transport service user. The users can request and negotiate a certain quality of transport service parameters. Either user can unconditionally release the connection. The connection may also be released by the transport layer.

Normal data transfer. The users can request for transfer of user data having any integral number of octets. The mode of data transfer can be two-way simultaneous. The connection-oriented transport service maintains integrity of user data in the sense that content errors, disordering of sequence, duplication of user data octets are taken care of.

Urgent data. The user can mark a segment of data as urgent. The transport entity delivers the segment to receiving user alerting him that data being received is urgent.

Forced data delivery. Usually the transport entity does not recognize the boundaries of user data segments and transports user data as a continuous stream of octets. But if required, a user may force the transport entity for delivery of a segment of user data or even a single octet of user data.

Quality of service. The transport layer allows the user entity to specify quality

of service parameters.

Connectionless transport service. Because of the connectionless operation, there is no connection establishment or release phase. The service is unreliable in the sense that user data may be lost, corrupted or disordered. This is consistent with the connectionless-mode of operation.

20.1.3 Functions within Transport Layer

Functions carried out by the transport layer depend on the type of transport service it provides to the users and the type of network service it uses. Primary functions of the transport layer for connection-oriented transport service are listed below in the context of TCP and UDP protocols. Functions for the connectionless transport service are subset of these functions. We assume that the underlying network service is connectionless. These functions are described in detail when we discuss TCP and UDP.

To undertake the transport layer functions listed below, the transport layer attaches a header to the user data received from the upper layer. The header contains several fields as we will shortly see. We will refer to the transport data unit (header plus user data) as Transport Protocol Data Unit (TPDU).

Establishing, maintaining, and releasing transport connections. Transport layer establishes, maintains, and releases end-to-end transport connection on the request of the upper layer. Establishing connection entails allocation of buffers for storing user data, synchronizing the sequence numbers of data units, negotiating the size of TPDU's, *etc.*

A transport connection is released at the request of the upper layer. The connection release can be normal release wherein the connection closed after all the data units in transit are delivered and acknowledged.

Data transfer. The transport layer segments user data and attaches a transport layer header to the user data segment forming a TPDU. The TPDU is handed over to the network layer for its delivery to the destination. The header contains port addresses of the user entities, sequence number, acknowledgement number, checksum and other fields for the various transport layer functions.

Flow control. The transport layer carries out end system-to-end system flow control using sliding window flow control mechanism. Unlike data link layer, the transport layer uses a dynamic window whose size is controlled by the receiver at the other end. In other words, the receiver based on the free buffer

available at its end, can shrink or expand the window at the sending end. This control is exercised by the receiver by prescribing the window size in the header of TPDU's it sends to the other end.

Error control. Since the IP network that interconnects two end systems offers unreliable connectionless service and the data link layer cannot provide end-to-end error control, the responsibility of providing end-to-end error control is given to the transport layer. Errors can be of several types:

1. Content errors due to corruption of bits
2. Content errors due to non-delivery of TPDU's
3. Content errors due to duplicate delivery of TPDU's
4. Sequencing errors due to disordering of the TPDU's
5. Delivery of TPDU's to a wrong destination.

Transport layer has built-in error control checksum mechanism that detects and corrects errors by retransmission of the transport data units. TPDU's carry sequence numbers so that missing TPDU's are readily detected. The missing TPDU's are retransmitted using an acknowledgement mechanism. Sequence numbers of TPDU's also enable putting the data units in the right sequence and detecting duplicate TPDU's, which are discarded.

Forced delivery. The transport layer allows the user to indicate end of user's data segment so that the segment may be sent by the transport layer without waiting for more user data octets. If no such indication is given, the transport layer assumes user data to be a continuous stream of data and takes its own decisions for segmenting the data.

Reset. A connection can be aborted by the users in case the connection is required to be closed immediately. The transport layer uses a reset function in this case. The connection is released without delivery of data units in transit.

Congestion avoidance. The underlying IP layer is susceptible to congestion, in which case the IP packets get dropped. Transport layer incorporates mechanism to gradually build the traffic flow and when it notices congestion, it takes necessary action for reducing the rate at which it hands over TPDU's to the IP layer.

20.2 TRANSMISSION CONTROL PROTOCOL (TCP)

Transmission Control Protocol (TCP) is a connection oriented layer-4 protocol and it provides reliable service between computer processes (applications) that reside in two different end systems. It uses services of underlying IP layer which provides connectionless and unreliable service. TCP, therefore, has inbuilt mechanisms for error control, flow control, and sequenced delivery of the user data octets. TCP is specified in RFC 793. Its basic operational features are as follows:

- It has three phases of operation connection—establishment, data transfer, and disconnection phase.
- A data unit at TCP layer is called TCP segment, which is equivalent to a frame at layer 2 and packet at layer 3. A TCP segment contains a header and user data octets.
- Continuous repeat request with sliding window flow control is used.
- The window contains user data octets, not full TCP segments. Thus flow control is based on volume of user data octets, not on number of TCP segments that can be transmitted.
- Window size is dynamic and is controlled by the receiver at the other end.
- All user data octets are acknowledged but acknowledgements may not be individual. It has inbuilt timeout mechanism for retransmission of unacknowledged TCP segments. Acknowledgements are piggybacked on a TCP data segment.
- Mode of communication is two-way simultaneous.

20.3 TCP PORTS AND CONNECTIONS

TCP provides service to the higher layer through a port. A port is similar to Service Access Point (SAP) in OSI. Each communicating computer process is assigned a port number. By using different port numbers, a TCP layer can simultaneously serve multiple processes, *e.g.* e-mail, FTP, *etc.* as shown in Figure 20.3. Port number range 0–1023 is reserved for common applications. These are assigned by IANA (Internet Assigned Numbers Authority). Some well-known ports are listed below:

- 7 Echo
- 23 Telnet
- 25 Simple Mail Transfer Protocol (SMTP)
- 20 File Transfer Protocol (FTP)
- 69 Trivial File Transfer Protocol (TFTP)
- 80 Hyper Text Transfer Protocol (HTTP)

The combination of IP address and port is called a socket. Each socket pair uniquely identifies a connection. The connection between processes P₁ and P₄ in end systems A and B respectively can be written as socket (33, 10.0.0.1), socket (67, 10.0.0.3).

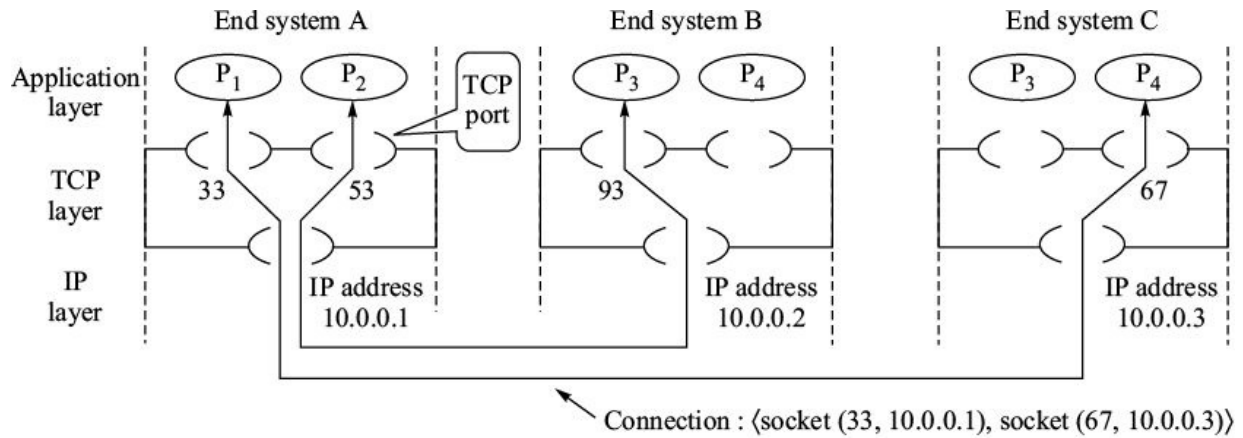


Figure 20.3 TCP ports and connections.

In a client-server environment, a server process may need to maintain several simultaneous sessions with clients on different end systems. Thus a single port supports multiple virtual connections (Figure 20.4).

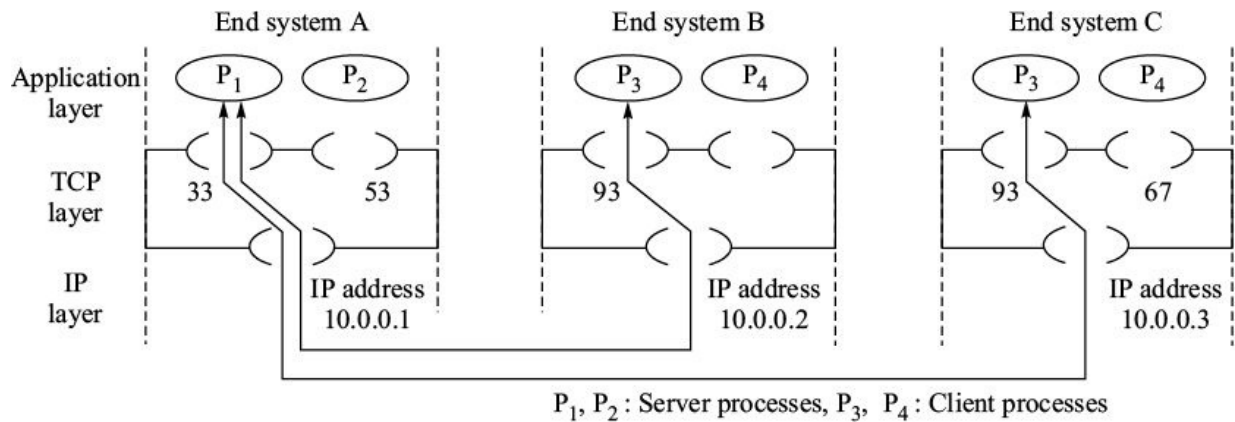


Figure 20.4 Multiple TCP connections through a port.

20.4 FORMAT OF TCP SEGMENT

TCP refers to the Transport Protocol Data Unit (TPDU) as TCP segment, and uses only one type of TCP segment as shown in Figure 20.5. The TCP segment header has minimum size of 20 octets and contains the following fields.

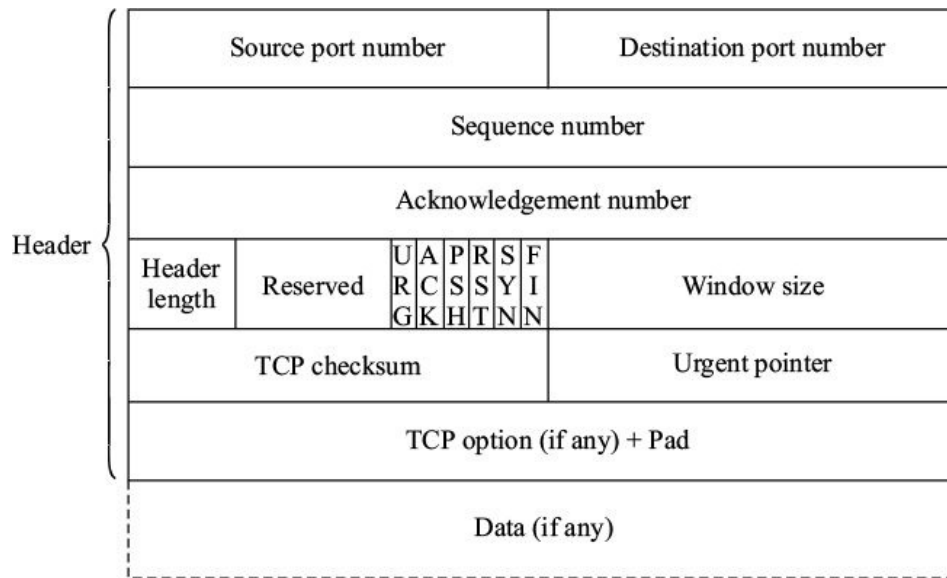


Figure 20.5 Format of TCP segment.

Source port number (2 octets). Port Number of the source process.

Destination port number (2 octets). Port number of the destination process.

Sequence number (4 octets). In TCP, the user data octets in the data field are given a sequence number. The sequence number contained in this field is the sequence number of the first data octet in this TCP segment. If the sequence number is N , and the TCP segment is carrying M data octets, the next TCP segment will have sequence number field equal to $N + M$. The sequence number wraps around to 0 after reaching $2^{32} - 1$. The range has been kept sufficiently large so as to avoid possibility of an original TCP segment after wrap around being confused as duplicate.

Acknowledgement number (4 octets). This field is the piggybacked acknowledgement of correct reception of all previous data octets and indicates the number of next data octet expected.

Header length (4 bits). It indicates length of the header in multiples of 4 octets.

SYN flag (1 bit). This bit is used for synchronizing the sequence numbers. This bit is used only during the connect phase. If it is set to 1,

- it implies that the TCP segment is a request for connection set-up,
- the sequence number field holds the initial value for a new session.

ACK flag (1 bit). If ACK bit is set to 1, it implies that the acknowledgement number field is valid and contains the sequence number of the next octet expected by the receiver.

FIN flag (Final, 1 bit). This bit is set to 1 for closing a TCP connection in one direction. Its use is explained in section 20.5.3.

RST flag (Reset, 1 bit). This bit is used for resetting a connection. When this bit is set to 1, the connection has to be cleared immediately. It is also used for refusing the request for connection set-up.

URG flag (Urgent, 1 bit). This bit is used to indicate that the TCP segment contains urgent user data octets and the urgent pointer field is valid. Use of URG flag and urgent pointer field is explained later.

PSH flag (Push, 1 bit). This bit is used to indicate to the receiver that the application (user) entity at the sending end has requested that this segment of data may not be buffered at the receiving end and handed to the user entity as soon as possible.

Window size (2 octets). The value in this field defines how many additional data octets will be accepted starting from the current acknowledgement number indicated in the acknowledgement number field. This field enables dynamic control of window at the sending end by the receiver.

TCP checksum (2 octets). This field contains 1's complement of the sum of all the 1's complements of 16-bit words in the TCP segment including user data field and pseudo IP header described later. It enables detection of content errors and wrong delivery (delivery at wrong destination) of TCP segments by the IP layer.

Urgent pointer (2 octets). When the value contained in this field is added to the number in the sequence number field, the resulting number is the sequence number of the octet next to the last octet of urgent data. This field is significant when URG field is set to 1.

Options (Variable). Options as defined in RFC 1146, RFC 1323, and RFC 1693 can be specified here. Pad octets are used to make the length of the options field integral multiple of 4 octets.

The commonly used option is to specify the acceptable Maximum Segment Size (MSS). MSS defines maximum number of user data octets that a TCP segment can carry.

It may be noticed that TCP format has no place for quality of service parameters. The parameters are as follows:

- Precedence
- Delay
- Throughput
- Reliability.

These parameters are received along with the service request for establishing connection and are passed onto the IP layer as pass-through parameters and are used in the TOS field of the IP packet.

20.4.1 Maximum Segment Size (MSS)

Ideally, TCP segment size should be such that it fits into IP packet having minimum MTU. If the TCP segment is of larger size, the IP layer fragments it to fit into minimum MTU size. If an IP packet containing a fragment is lost, reassembly of the TCP segment does not take place and the IP layer discards all the other received fragments. If the TCP layer itself restricts the TCP segment size compatible to the minimum MTU size of IP packets, the retransmission overhead is significantly reduced.

Therefore, the two TCP layers in the end systems attempt to discover minimum MTU size at the IP layer before agreeing on Maximum Segment Size (MSS) of the TCP payload. Else, they agree on MSS of 536 octets, which is based on default size of IP packet (default IP packet size = 576 octets, IP and TCP headers = 40 octets).

20.4.2 Pseudo IP Header

In Figure 20.4, suppose a TCP segment is sent by end system A to end system B, but it is delivered to end system C instead. The TCP layer in end system C will not be able to identify the wrong delivery. It will unpack the TCP segment and hand over the user data octets to the addressed TCP port. To avoid such situations, the checksum of the TCP segment is calculated using a pseudo IP header shown in Figure 20.6. The pseudo IP header consists of source IP address, destination IP address, protocol type which is 6 for TCP, and TCP

segment length (including TCP header but excluding pseudo IP header). The pseudo header is not attached to the TCP segment.

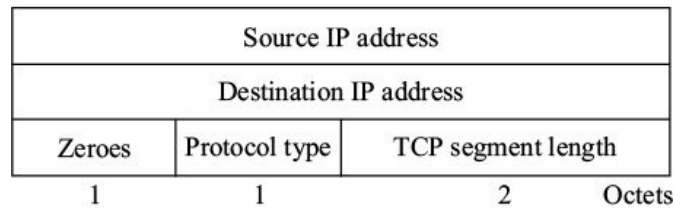


Figure 20.6 Pseudo IP header of TCP.

At the receiving end, the IP layer extracts the TCP segment from the received IP packet. It hands over the TCP segment to the TCP layer. It also hands over source and destination IP addresses that were available on the received IP packet. The TCP layer performs the checksum verification using this information to verify that the TCP segment is intact (i.e. no content errors) and has arrived at the correct destination.

20.4.3 Forced Data Delivery

TCP accumulates data octets received from the user in its buffer and sends a TCP segment containing the received octets when sufficient number of them is available. TCP treats user data to be a continuous stream of data and creates its own segment boundaries without any regard to the boundaries of application messages.

The user application may require the TCP layer to send a message without waiting for the TCP buffer to fill. For example, a user in an interactive session may like to get response when he presses a key. The application uses 'push' attribute of TCP service to force the transmission of TCP segment. But transmission of the TCP segment does not completely address the issue. The receiving TCP entity must also be forced to hand over the data segment to the receiving application entity. This is achieved by setting push flag to 1 in the TCP segment (Figure 20.4).

20.4.4 Urgent Data

A user may specify a block of data as urgent data. The TCP segment(s) that carry these data octets have the urgent flag (URG) set to 1. When the URG flag is set, the urgent pointer field becomes valid. The value in this field when added to the sequence number, points to the position of the data octet following the last urgent data octet. At the receiving end the TCP entity hands over the urgent

segment of the user data with an alert that urgent data is being handed over.

20.5 TCP OPERATION

TCP operates in connection-oriented mode and thus has three phases of operation:

- Connection establishment phase
- Data transfer phase
- Disconnection phase.

20.5.1 TCP Connection Establishment Phase

During connection establishment phase the TCP entities synchronize their TCP segment sequence numbers, acknowledgement numbers, and options. The TCP layer uses a three-way handshake as described below to establish a connection. Three-way handshake between two entities A and B involves transmission of:

- request for connection from A to B,
- acceptance of connection from B to A, and
- acknowledgement of receipt of acceptance from A to B.

To illustrate the connection establishment mechanism, let us assume two end systems A and B want to communicate. We assume B is server and A is client. The sever process in B listens on its well known port. The connection establishment is initiated by the client process in A. It gives a request to the TCP layer of A for setting up the connection to server's well-known destination port. For itself, it chooses a free port number and indicates it to the TCP layer. Since the underlying IP service of the network layer is unreliable, we will consider the following two situations when the connection is being established:

- TCP segments do not get lost or delayed.
- TCP segments get lost or delayed.

No lost or delayed TCP segments

1. The TCP layer of A sends TCP segment with SYN flag set to 1 indicating

its wish to set up connection (Figure 20.7). It selects a random sequence number (270) for the respective field of the TCP segment. Since the ACK flag is not set, the acknowledgement number field is not significant.

2. B replies with TCP segment having SYN flag set to 1. It selects a random sequence number (478) and puts it in the respective field of the TCP segment. It acknowledges receipt of the TCP segment of A by setting ACK flag to 1 and putting the next sequence number (271) of expected data octet in the acknowledgement number field.
3. A replies with TCP segment having ACK flag set to 1 and puts the next sequence number (479) of expected data octet from B in the acknowledgement number field. Note that this TCP segment is from A is just an acknowledgement.

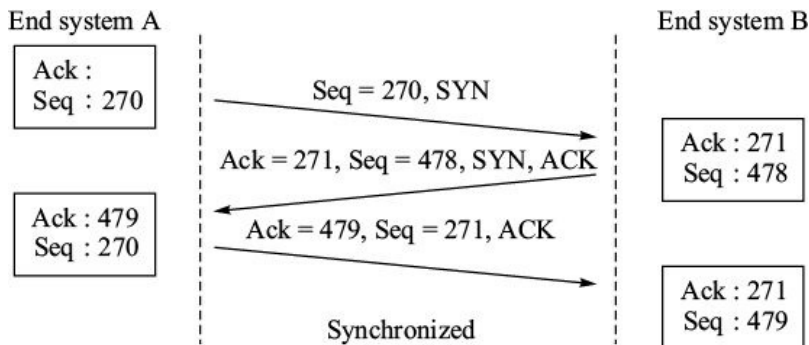


Figure 20.7 Connection establishment phase.

With this three-way handshake, the acknowledgement and sequence numbers in the two end systems are synchronized and data transfer can begin.

Lost or delayed TCP segments. Three-way handshake is necessary and sufficient for synchronization. It is to be remembered that TCP builds on unreliable IP service which may delay, lose or disorder IP packets. Lost or delayed requests and acknowledgements are taken care of by a timeout mechanism whereby the request for connection is retransmitted.

There is possibility of duplicate request being generated when the original request is delayed or when the acknowledgement is delayed or lost. Three-way handshake takes care of all these issues. If a delayed request is received after the connection is established, the duplicate request is ignored.

Initial sequence numbers. IP service being unreliable, it is possible that delayed TCP segments of an already closed TCP connection may arrive after a new connection is established between the same sockets. The old TCP segments

may disturb the working of the new connection. Therefore, sequence numbers must be different for different sessions. This is achieved by picking random numbers as initial sequence numbers that are exchanged during connection establishment phase.

20.5.2 TCP Data Transfer Phase

After the end systems have synchronized their sequence and acknowledgement numbers during the connection establishment phase, data transfer can begin. As before, we will consider two situations, when the TCP segments get lost or delayed and when they do not. Flow control during data transfer requires detailed description and therefore it is described in a separate section.

No lost or delayed TCP segments. Figure 20.8 shows the exchange of TCP segments after the connection is established between end systems A and B. In the example shown, end system A sends TCP segments containing user data and end system B acknowledges. We assume that there are no errors or delayed TCP segments.

1. A sends TCP segment containing ten user data octets and bearing sequence number 271.
2. B was expecting user octet number 271. It counts the number of received user data octets and generates acknowledgement bearing acknowledgement number 281.
3. A sends the next TCP segment containing 14 user data octets, which is acknowledged in the same manner as before by B.

We have considered, for the sake of simplicity, that A sends one TCP segment at a time. The mode of TCP segment exchange is continuous RQ, and A can continuously send TCP segments that contain user data octets in the window without waiting for acknowledgement from the receiver.

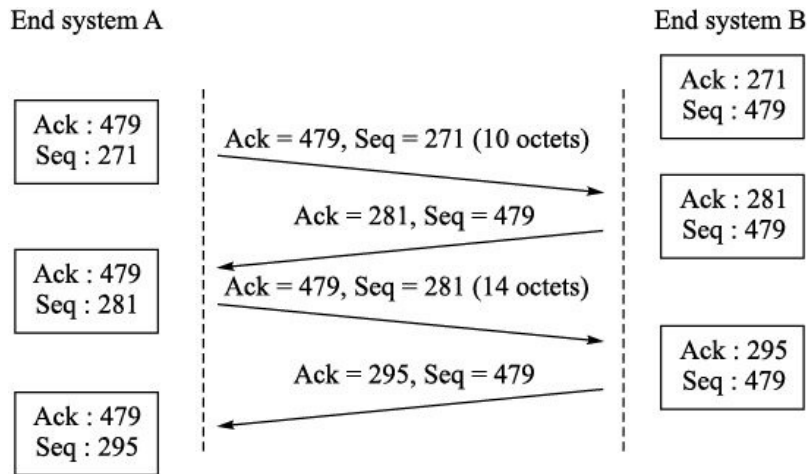


Figure 20.8 Data transfer phase.

Lost or delayed acknowledgements. Figure 20.9 illustrates the mechanism when the acknowledgements are lost or get delayed. When acknowledgement 281 is lost or delayed, A retransmits the unacknowledged TCP segment 271 after timeout (Figure 20.9a). B, therefore, receives a duplicate copy of TCP segment 271. It realizes that its last acknowledgement is lost and it retransmits the acknowledgement.

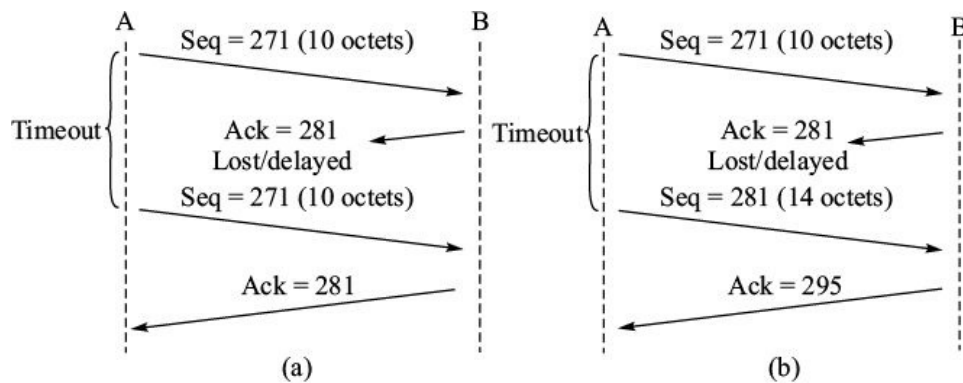


Figure 20.9 Recovery from lost or delayed acknowledgements.

Since continuous RQ is used, it is possible that the next TCP segment may have been sent meanwhile and received before timeout. As shown in Figure 20.9b, A sends the next TCP segment 281 containing 14 octets before timeout. In this case, B having received all the user data octets up to 294, sends Ack 295 which confirms to A that both the TCP segments 271 and 281 have been received by B.

Lost or delayed TCP segments with user data. When a TCP segment containing user data is lost or delayed, the sender retransmits the unacknowledged TCP segment after timeout. Figure 20.10 illustrates the

process. A sends the TCP segment 271 which is lost. After timeout it retransmits the segment. However, A had already sent next TCP segment 281 before timeout. B withholds its acknowledgement until the missing/delayed segment is received. After it receives the retransmitted segment, it sends Ack 295 which acknowledges receipt of all TCP segments up to 294.

If the original TCP segment was delayed, a duplicate TCP segment is received. The receiving end sends acknowledgement for both the TCP segments because it assumes that the first acknowledgement was lost. It is to be remembered that all TCP segments are acknowledged, even if it means repeating the last acknowledgement. For example, in Figure 20.10, if original TCP segment 271 was delayed and received by B after it had already sent acknowledgement 295, B would repeat acknowledgement 295.

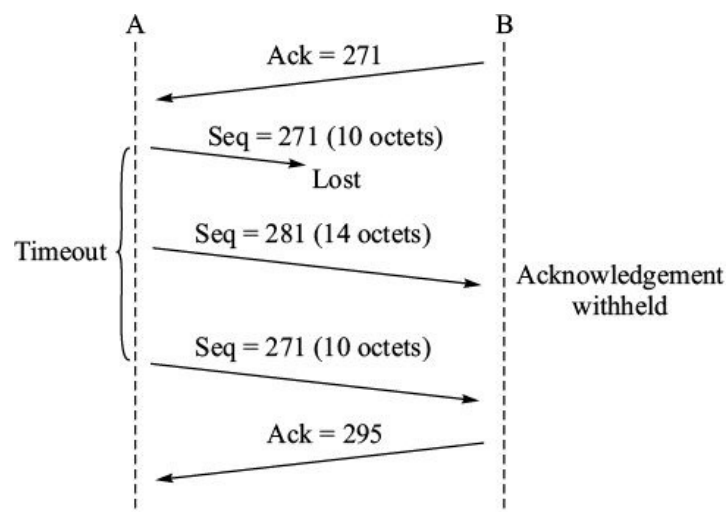


Figure 20.10 Recovery from lost TCP segments containing user data.

Fast retransmission. Typically, when a TCP segment is lost or delayed, the TCP layer withholds transmission of acknowledgements for the subsequent TCP segments until the missing TCP segment is received. At the sending end, the TCP layer waits for timeout before retransmitting the unacknowledged TCP segment. The Retransmission Time Out (RTO) can be as long as two times the round trip delay. This process causes unnecessary delay. Retransmission timeout cannot be kept low because low RTO can result in too many retransmissions.

Fast retransmission requires the receiver to send an acknowledgement as soon as it detects a missing TCP segment. The receiver detects a missing segment when the subsequent TCP segment arrives. Since the receiver cannot selectively acknowledge the received TCP segment,¹ it resends the last acknowledgement. As soon as the sender receives a duplicate acknowledgement, it realizes that the

receiver has detected a missing segment (the acknowledgement number indicates the missing segment) and it retransmits the missing segment without waiting for timeout.

In the example shown in Figure 20.11, B repeats Ack 271 on receipt of TCP segment bearing number 281. Duplicate acknowledgement indicates to A that TCP segment 271 is missing and therefore A retransmits it. This process is faster because timeout is of the order of 2 times Round Trip Time (RTT), while in this case, the sender will receive indication of missing segment within almost one RTT.²

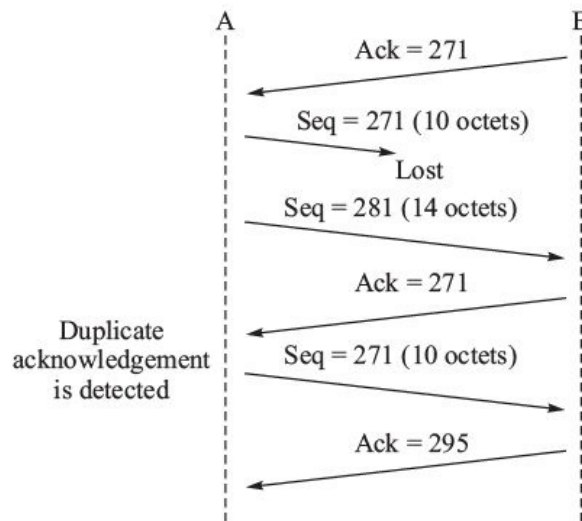


Figure 20.11 Fast retransmission.

Usually the sender retransmits a TCP segment after it receives three duplicate acknowledgements. Three duplicate acknowledgements are generated when the receiver receives three succeeding TCP segments.

Delayed acknowledgement. Normally an acknowledgement for a TCP segment is sent immediately after its receipt. If the window at the receiving end has user data octets to send, this acknowledgement can be piggybacked on next outgoing TCP segment.

If there are no user data octets in the window of the receiver, it delays transmission of the acknowledgement for a specified maximum period. It assumes that the receiving process by this time would process the received data and hand over its response transmission (Figure 20.12). The acknowledgement can be piggybacked on the TCP segment carrying the response. By delaying acknowledgement, TCP has saved transmission of one TCP segment. TCP specifies 500 ms as the maximum value of delay. The typical implementation of

delay is 200 ms.

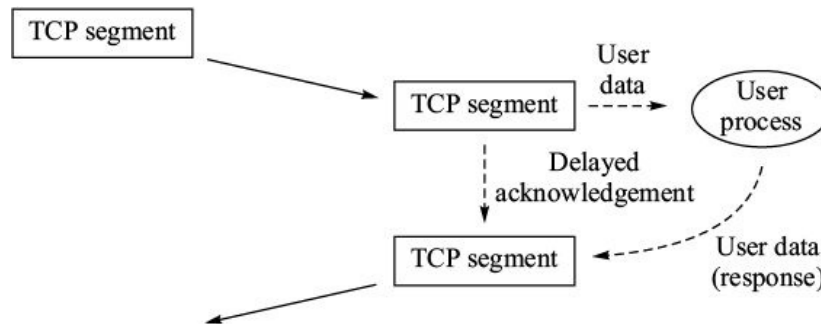


Figure 20.12 Delayed acknowledgement.

Delayed acknowledgements are beneficial in interactive applications in particular. In interactive applications every key punch is echoed back by the distant end. Thus, for every user octet sent, there is an immediate response to send. The acknowledgement for the sent octet can be piggybacked along with the response.

20.5.3 TCP Disconnection Phase

A TCP connection is viewed as two independent one-way connections, one in each direction. When an application entity tells TCP entity to close a connection, the TCP entity closes the connection in the outgoing direction after ensuring that it has sent all the data octets in its buffer and has received acknowledgement for the same. It also indicates its intent to close to connection to the TCP entity at the other end by setting FIN flag to 1. After it receives acknowledgement for this last segment from the other end, the one way connection is closed. The TCP entity, however, continues to accept the TCP segments being received from the other end until this part of the connection is closed by the TCP entity at the other end in a similar manner. Figure 20.13 illustrates the process.

1. After A receives request for terminating the connection from the user, it sends all the user data in its buffer and finally sends the last TCP segment setting FIN flag to 1 in it. When acknowledgement is received from B, connection in the direction from A to B is closed.
2. Having sent the TCP flag with FIN bit set to 1, A continues to receive TCP segments containing user data from B, if any.
3. B indicates request to close the connection to the application entity at its end. After it obtains response from the application entity, it also initiates action for closing the connection from B to A.

4. B sends its last TCP segment 479 setting FIN flag to 1 in it. When acknowledgement is received from A, connection from B to A is closed.

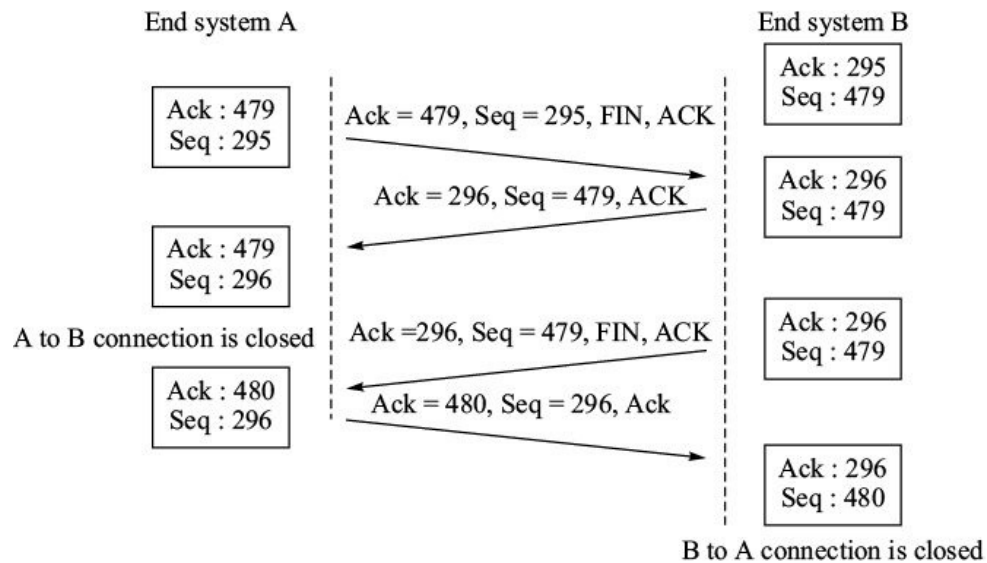


Figure 20.13 Disconnection phase.

20.5.4 TCP Connection Reset

A TCP connection is normally closed in the manner described above. Sometimes there are abnormal conditions that force abrupt termination of the connection. This is carried out using reset (RST flag). The TCP entity sends a TCP segment with RST flag set to 1. The other end responds immediately by aborting the connection. The application entities are informed that a reset has occurred. After reset, the connection ceases immediately and all TCP segments in transit and the user data in the TCP buffer are lost/discarded.

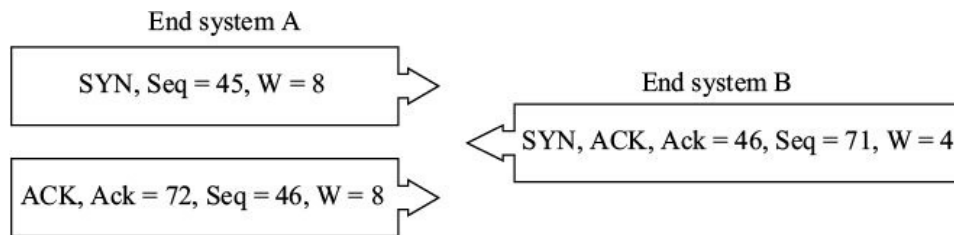
20.6 FLOW CONTROL IN TCP

TCP uses sliding window mechanism for flow control. As we learnt earlier, sliding window mechanism allows a sender to transmit all the data units that are in the window without waiting for their acknowledgement. The TCP sliding window mechanism operates on the volume of user data rather than number of TCP segments that can be sent. In other words, window contains user data octets and window size refers to the maximum number of user data octets that can be sent without waiting for acknowledgement. The user data octets in the window may be sent in one or several TCP segments. The number of TCP segments

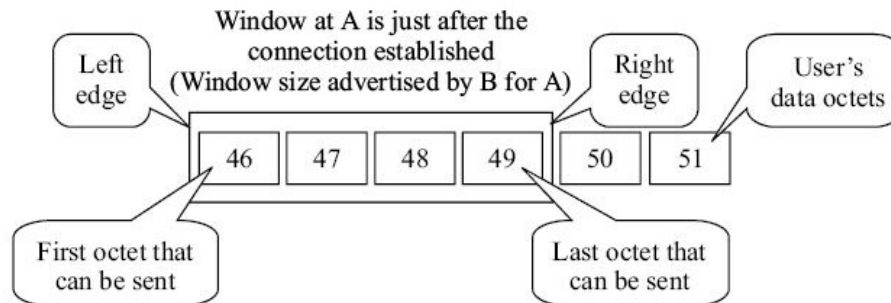
depends only on MSS.

20.6.1 Window Size

The window size can be from 0 to 65535 octets. Another unique feature of TCP sliding window mechanism is that the window size is variable and it is controlled by the receiving end. Thus, if a receiver has full buffers, it can shrink the window at the sending end in order to stop further transmission by the other end. The starting size of the window is negotiated during connection establishment phase as shown in Figure 20.14.



(a) Window size determination (Connection establishment phase)



(b) Status of window at A just after connection establishment

Figure 20.14 Window size determination in connection establishment phase.

1. End system A specifies window size of 8 octets for B in the window size field of the TCP header.
2. Acknowledgement from B implies that it has accepted to keep its window size to 8 octets.
3. B indicates window size of 4 octets for A. Acknowledgement from A confirms to B that A has accepted the specified size of the window.
4. The window at A contains four octets 46 to 49 to start with. Window at B contains 8 octets from 72 to 79.

20.6.2 Sliding Window Mechanism

The sliding window has two edges—left edge and the right edge (Figure 20.14). These edges move independently and are controlled entirely by the receiver at the other. As the acknowledgement is received, the left edge moves towards right. The acknowledged octets move out of the window.

Movement of the right edge is decided by the receiver based on the available buffers for storing the data octets being received from the other end. If sufficient buffer capacity is available, it instructs the sending end to move the right edge of its window to the right. If the buffers are full, the receiver keeps the right edge where it is. It can never instruct the sender to move the right edge to the left.

The control of right edge of the sender's window is exercised by the receiver through window size field available in the TCP segment. Figure 20.15 shows some examples of window status when the receiving end B sends an acknowledgement and prescribes the window size for A. The current window status of A is shown in Figure 20.15a. It contains octets 48 to 51. Octets 46 and 47 have already been acknowledged by B and are outside the window. Octets 52 to 54 are waiting outside to enter the window.

Figure 20.15b shows the case when B sends Ack = 49. The left edge of the window moves to right so that acknowledged octet 48 goes out of window. The right edge is always just ahead of the octet with sequence number equal to acknowledgement number plus advertised window size. In this case, B prescribes window size of 3, therefore the right edge remains just ahead of octet number 52 ($= 49 + 3$). It is to be remembered that the right edge can never move towards left.

Figure 20.15c shows the case when B prescribes window size of 5. In this case, the right edge moves just ahead of octet number 54 ($= 49 + 5$).

Figure 20.15d, shows the case when B sends Ack = 52 and Window = 0. In this case, the left edge moves to extreme right edge of the window as all the octets are acknowledged. The right edge does not move because $52 + 0 = 52$.

When the window is shrunk to zero size, the sender cannot send any TCP segment. The receiver must therefore send another acknowledgement advertising a next window size. To account for possibility of loss of this acknowledgement, in which case deadlock would occur, TCP standard permits the sender to send a TCP segment containing one octet of user data.

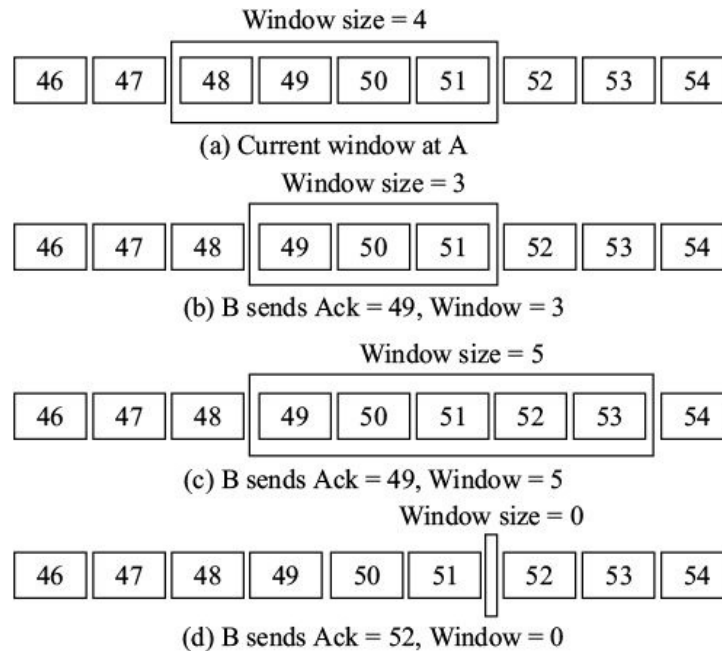


Figure 20.15 Control of left and right edges of the window by the receiver.

20.6.3 Silly Window Syndrome

A serious performance issue is encountered when the sending application process generates data at faster rate than the receiving application process can accept. Eventually the window size will be shrunk to zero and the receive buffers will also be full. If the application process at the receiving end reads one octet at a time from the receive buffer, the receiving end can slide the right edge of the sender's window by one octet by sending an acknowledgement. The sender can send the next TCP segment with one octet of user data. To transport this octet, an overhead of 40 octets (20 octets of TCP header and 20 octets of IP header) is incurred. Thus for every octet read by the process at the receiving end, one TCP segment containing one octet of user data is sent by the other end. This results in very inefficient use of network resources. This state of the TCP connection, when the window size is shrunk to such a large extent that very small amount of user data is transported in each TCP segment, is called silly window syndrome. The silly window syndrome is resolved by implementing the following policies.

Delayed acknowledgement. TCP standard specifies that transmission of the acknowledgements is delayed by the receiver till that time when it has either 50% vacant buffer or sufficient vacant buffer for 1 MSS. The maximum time an acknowledgement can be delayed is 500 ms in TCP because large delay can cause other problems. Estimates of RTT can go wrong. The sender may retransmit the TCP segment if timeout occurs.

Nagle's algorithm. The mechanism for silly window syndrome avoidance at the sending end is known as Nagle's algorithm after its inventor. This is a self-clocking algorithm for pacing generation TCP segments. The first octet handed over by the application process is sent immediately by the TCP layer. Thereafter transmission of TCP segments is clocked by the acknowledgements. The TCP layer accumulates additional octets until an acknowledgement arrives or the accumulated user data becomes equal to MSS. This ensures that the TCP segments will contain reasonable number of user data octets irrespective of how fast or slow the application process is.

20.7 ESTIMATION OF RETRANSMISSION TIMEOUT IN TCP

The performance of TCP is dependent on the unreliable service provided by the underlying IP layer. IP packets may be lost, discarded, delayed or delivered out of sequence. Such situations are caused by the congestion at the IP layer. TCP layer tries to correct the situation by implementing a retransmission timer. If acknowledgement for a TCP segment is not received within a defined Retransmission Time Out (RTO), the TCP segment is retransmitted. RTO is kept greater than the Round Trip Time (RTT) from the source to destination. If RTO is less than RTT, there will be needless retransmissions which will generate more traffic for the IP layer to handle. This has a cascading effect as more traffic results in more congestion causing higher RTT and loss of IP packets.

If RTO is kept much larger than RTT, there will be a long idle time if a TCP segment is lost and there are no data octets in the window to transmit.

Therefore, a good estimate of RTT is required to set the value of RTO. RTT is dependent on

- the distance between the source and destination and the transmission speed. Therefore, RTT is to be estimated for each connection; and
- the congestion in the network layer. Therefore, an adaptive algorithm for estimating RTT for each connection is required.

TCP implements an adaptive retransmission algorithm that monitors RTT on each connection and adjusts RTO to a value somewhat greater than the estimated RTT.

20.7.1 Methods of Estimating RTT and RTO

There are several approaches for estimating RTT and computing RTO. These are:

1. Simple average
2. Exponential average
3. Jacobson's algorithm
4. Karn's algorithm.

Jacobson's algorithm takes into account the variance of observed values of RTT to arrive at an estimate of RTO. Karn's algorithm accounts for RTT of the retransmitted TCP segments. We will not go into details of these two algorithms and restrict discussion on estimation of RTO to simple and exponential averages.

Simple average. In this approach, simple average of observed RTTs of last $(N + 1)$ TCP segments is taken. RTT is the duration between the time of transmission of a TCP segment and time of reception of its acknowledgement. Average RTT (ARTT) is given by

$$\text{ARTT}(N + 1) = \frac{1}{N + 1} \sum_1^{N+1} \text{RTT}(i)$$

where $\text{RTT}(i)$ is the round trip observed for the i^{th} transmitted segment. $\text{ARTT}(N + 1)$ can be expressed in terms of $\text{ARTT}(N)$ as given below. This formula enables iterative computation of the ARTT.

$$\text{ARTT}(N + 1) = \frac{N}{N + 1} \text{ARTT}(N) + \frac{1}{N + 1} \text{RTT}(N + 1)$$

TCP does not use simple average for estimating RTT. It uses the exponential average described next.

Exponential average. In the above formula, the last RTT observation is given a weight of $1/(N + 1)$ and the past average is given a weight of $N/(N + 1)$. RFC 793 specifies the following formula wherein these weights can be specified based on nature of RTT observations. The estimated of RTT is referred to as smoothed RTT (SRTT) estimate.

$$\text{SRTT}(N + 1) = a \text{SRTT}(N) + (1 - a) \text{RTT}(N + 1)$$

where a is smoothing constant and can have any value between 0 and 1. Low value of a means less weight to past estimate and therefore SRTT will be more sensitive to variations in recent RTT observations. Large value of a implies that past estimate is given more weight to arrive at the new estimate. RTO can now be calculated by multiplying SRTT by a constant factor b .

$$RTO(N + 1) = b \text{ SRTT}(N + 1)$$

RFC 793 lists typical values of b between 1.3 to 2. An upper and lower bound on RTO is also defined to ensure that the unrealistic estimated values of RTO may not cause performance degradation.

$$RTO(N + 1) = \min [\text{upper bound of RTO}, \max \{\text{lower bound of RTO}, b \text{ SRTT}(N + 1)\}]$$

20.8 CONGESTION AVOIDANCE IN TCP

If there is congestion in the underlying IP network, it results in increased delay and packet loss. Congestion at the network layer is reflected as loss of TCP segments and their delayed delivery at the TCP layer. As we saw earlier, TCP takes care of the lost or delayed TCP segments using a timeout and retransmission mechanism. Retransmissions at TCP layer due to congestion in the network layer can further aggravate the congestion instead of alleviating it. Therefore, TCP must react to congestion in the network by controlling the rate at which it pumps TCP segments into the IP layer.

To avoid congestion in the network and recover from the effects of congestion, TCP standard recommends the following techniques which are used together:

- Slow start
- Congestion avoidance
- Fast recovery.

All modern implementations use these techniques.

20.8.1 Slow Start

Once a TCP connection is established, the sender is authorized to transmit TCP segments containing the user data octets in the window without waiting for an

acknowledgement. It is possible that the network may not support this traffic between the two ends and there may be congestion in the underlying network.

In slow start mechanism, the sender gradually builds up the rate of sending TCP segments, and if there is no congestion, it peaks at the window size advertised by the receiver. If at any point of building up the rate of sending traffic, it notices congestion, it immediately drops the rate. Congestion is detected by the TCP entities when there are retransmission timeouts and duplicate acknowledgements. Let us examine this in detail.

1. Slow start method defines another window called congestion window (cwnd) as shown in Figure 20.16. This window is controlled by the sender. The sender is allowed to send TCP segments in the cwnd or in the window advertised by the receiver, whichever is less.
2. Just after a TCP connection is established, the congestion window is limited to its minimum size which is equivalent to one MSS.
3. The sender sends a TCP segment containing the octets in the cwnd and waits for the acknowledgement. Remember that the window advertised by the receiver may contain many more data octets, but the sender waits for the acknowledgement of the TCP segment sent because the cwnd is one.
4. If the acknowledgement arrives before retransmission timeout, the sender doubles the size of cwnd, *i.e.* the cwnd size becomes equivalent to 2 MSS.
5. If the acknowledgements for the next two TCP segments also arrive before timeout, cwnd size is again doubled, ³*i.e.* it becomes equivalent to 4 MSS.
6. The process is repeated till the cwnd size becomes equal to or more than the window advertised by the receiver. At this stage, the window is set to the size as advertised by the receiver, and the network is able to sustain the traffic with this size of window.

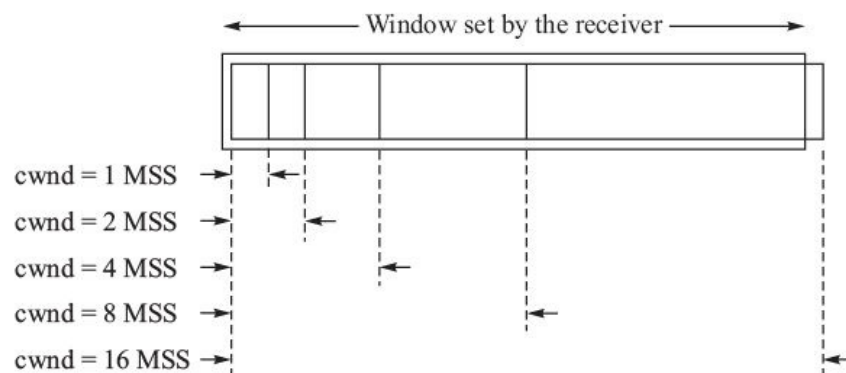


Figure 20.16 Congestion window (cwnd).

Figure 20.17 illustrates the built up of cwnd as described above. The RTT is assumed to be much larger than the transmission time of a TCP segment. Note that with each RTT, cwnd increases exponentially.

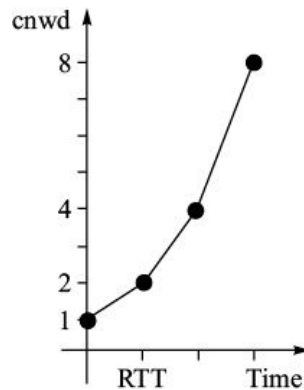


Figure 20.17 Slow start.

Slow start is used just after the TCP connection is established and congestion condition of underlying IP network is unknown. It is possible that congestion in the network becomes severe during the life of the connection. When severe congestion condition is detected, the slow start process beginning with $cwnd = 1$ MSS is restarted. This is explained in detail in the next section.

20.8.2 Congestion Avoidance

Congestion in the network is indicated in two ways:

- Receiving no acknowledgement
- Receiving duplicate acknowledgements.

Receiving no acknowledgement at all implies that no TCP segment is able to reach destination. Otherwise, at least duplicate acknowledgements would have been generated. Therefore, receiving no acknowledgement is indicative of very severe congestion condition. Since no acknowledgement is received, the retransmission timer expires. Expiry of retransmission timer is taken as indication of severe network congestion. At this stage, it becomes necessary for the TCP layer to assess the capacity of the underlying IP network. Therefore, slow start process is started and cwnd is reset to 1 MSS as before.

The receipt of a duplicate acknowledgement indicates that a TCP segment is missing and the TCP segment subsequent to the missing TCP segment has been received. Therefore, the congestion is less severe in this case. Receipt of three

duplicate acknowledgements (four acknowledgements bearing same acknowledge number in all) is taken as indication of congestion. In such situation congestion avoidance scheme with fast recovery is adopted. Let us understand congestion avoidance first.

The congestion avoidance scheme uses one more control parameter called 'slow start threshold' (ssthresh) with cwnd. In this scheme, cwnd is incremented linearly instead of exponentially after it exceeds the ssthresh (Figure 20.18). Linear growth of cwnd is obtained by incrementing cwnd only by 1 MSS after the acknowledgements for all the TCP segments in cwnd have been received. In other words the increment is by a fraction of MSS $1/cwnd$ for every acknowledgement received. Here cwnd is expressed as number of MSS. Linear growth of cwnd is likely to avoid congestion in the network.

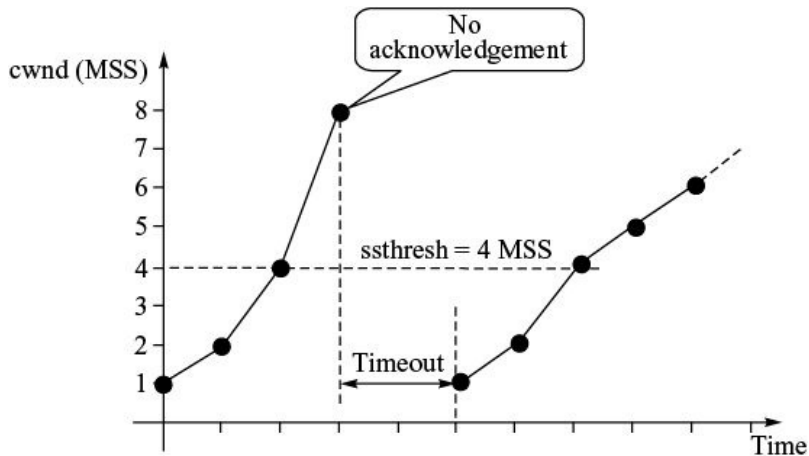


Figure 20.18 Congestion avoidance.

The value of ssthresh is determined by the last cwnd value when the congestion is detected. It is set to half the last cwnd value. Just after the connection set up, the initial value of ssthresh is set equal to the maximum possible value of MSS, *i.e.* 65,535 octets.

Let us examine the process in some detail with an example (Figure 20.18).

- When the connection is established, the slow start procedure for building up the cwnd is started. The ssthresh is at the default value of 65,535 octets to start with.
- At $cwnd = 8$ MSS, timeout occurs. Therefore, cwnd is reset to 1 MSS and ssthresh is set to 4 MSS. Slow start is restarted and cwnd is incremented exponentially up to 4 MSS. At $cwnd = 4$ MSS, congestion avoidance mechanism becomes effective and cwnd increments linearly thereafter.

20.8.3 Fast Recovery

After the sender detects a missing TCP segment as indicated by receipt of 3 duplicate acknowledgements, fast recovery mechanism is initiated. It is linked to the congestion avoidance scheme described above. Fast recovery with congestion avoidance works as follows:

1. Every time a duplicate acknowledgement is received, $cwnd$ is continued to be incremented linearly by one step. It allows the sender to continue sending TCP segments even if the missing TCP segment is not received at the other end.
2. After three duplicate acknowledgements are received, the sender takes it as indication of missing TCP segment. It retransmits the missing segment and sets $ssthresh$ to half the current $cwnd$.
3. The new value of $cwnd$ is set equal to revised $ssthresh$ plus 3 MSS since the receiver has already received three subsequent TCP segments.
4. The sender continues sending TCP segments incrementing $cwnd$ linearly as the duplicate acknowledgements are received.
5. When a next acknowledgement that acknowledges new data is received, $cwnd$ is set to $ssthresh$.

Figure 20.19 illustrates this mechanism. By the time $cwnd$ is equal to 10, three duplicate acknowledgements are received. Taking it as indication of congestion, $ssthresh$ is set to 5 MSS (half of $cwnd$ value). At this stage $cwnd$ is set equal to $ssthresh$ plus 3 MSS, *i.e.* equal to 8. Since $cwnd$ is above $ssthresh$, $cwnd$ increases linearly with every acknowledgement as explained earlier. The next acknowledgement of new data is received when $cwnd$ is equal to 11. At this stage $cwnd$ is set equal to $ssthresh$, *i.e.* 5 MSS.

This process is called fast recovery because after the receiver gets the missing segment, it sends an acknowledgement that acknowledges missing segment and all the received succeeding TCP segments as well. Thus the sender and receiver recover quickly. Note that fast recovery happens before expiry of retransmission timer. If timeout occurs, the sender is forced to retransmit the missing TCP segment and the slow start procedure initiated.

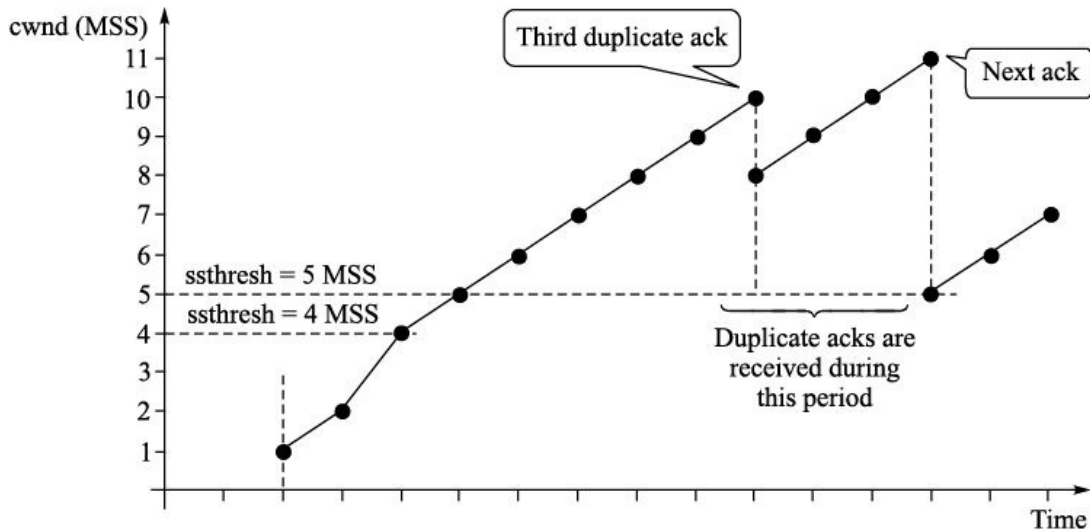


Figure 20.19 Fast recovery.

The steady-state view of effective TCP window is as shown in Figure 20.20.

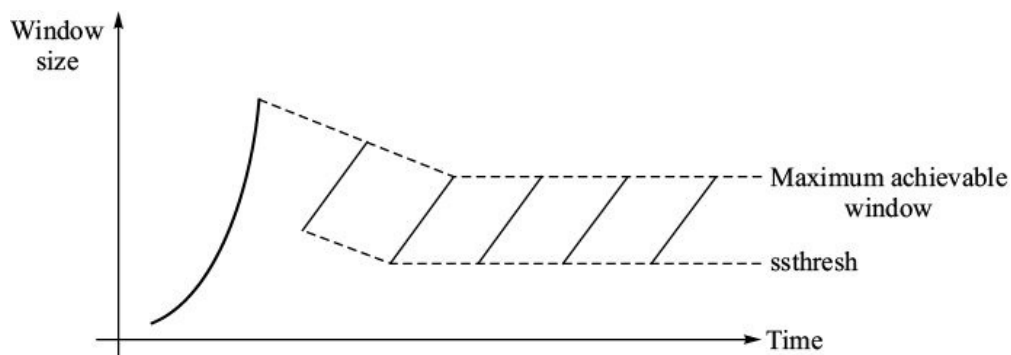


Figure 20.20 Steady-state view of the window size.

20.9 USER DATAGRAM PROTOCOL (UDP)

User Datagram Protocol (UDP) is a connectionless transport protocol that operates on IP layer (Figure 20.2). It is specified in RFC 768. Its main features are:

- UDP provides connectionless service to the application layer. Being connectionless, its service is unreliable.
- Each UDP datagram is transported from source to the destination independent of others. The delivery of user data may not be sequenced.
- There is no flow control or acknowledgement mechanism.
- There is optional checksum field in UDP datagram. It provides end-to-end error detection capability in the user data.

- It is less complex than TCP and easy to implement.

UDP is used where the overhead of a connection-oriented service is undesirable. Examples of such applications are given below:

- Periodic collection of data from sensors, transmission of alarms from security devices.
- Outward transmission of short broadcast messages.

UDP is also used where retransmission of lost segments makes no sense. For example, UDP is used for transmission of digital voice over IP. Retransmission of a lost voice packet does not make sense.

20.9.1 Format of UDP Datagram

There is only one type of UDP datagram. Figure 20.21 shows its format.

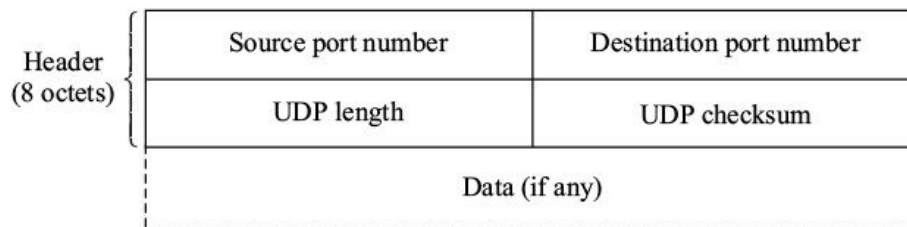


Figure 20.21 Format of UDP datagram.

Source port number (2 octets). Port number of the source process.

Destination port number (2 octets). Port number of the destination process.

UDP length (2 octets). This field indicates length of UDP datagram (header plus data) in octets.

UDP checksum (2 octets). This field contains 1's complement of the sum of all the 1's complements of 16-bit words in the UDP segment including user data field and pseudo IP header. If the computed value UDP checksum comes to all zeroes, all checksum bits are set to 1. UDP checksum is optional. If it is not used, it is set to all zeroes.

As explained earlier, pseudo IP header enables detection of wrong delivery of UDP datagrams by the IP layer. The format of UDP pseudo header is shown in Figure 20.22. The protocol type code is 17 for UDP. The UDP segment length includes number of octets in the UDP datagram (excluding pseudo header).

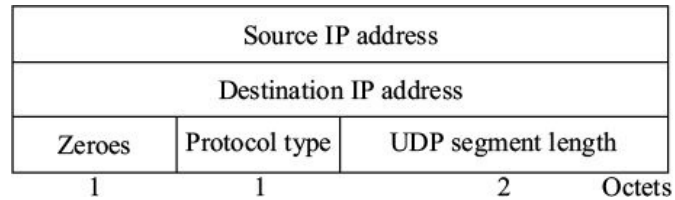


Figure 20.22 Pseudo UDP header.

20.9.2 UDP Operation

UDP operation is very simple. There is no connection establishment phase. UDP is always in data transfer phase. As and when user data is handed over by the application layer, it sends it to the destination as an independent data unit.

As a transport layer protocol, UDP adds little but needed capability to the IP service. It enables port addressing capability. IP addressing scheme enables delivery of a datagram to the end systems. There can be multiple applications in an end system. UDP supports simultaneous communications with these applications and delivers the user data to respective applications in an end system.

There is no flow control in UDP. As regards the error control, recall that IP does not compute checksum over the payload portion of the IP packet. Therefore, content errors in the payload of an IP packet go unchecked. UDP checksum is, therefore, the only way to guarantee that the user data is intact. UDP datagram received with content error are simply discarded.

The checksum also ensures delivery of a datagram to its correct destination. At the destination, the IP layer hands over the UDP datagram along with the source and destination IP addresses. The receiving UDP entity reconstructs pseudo header and then verifies the checksum. Wrong deliveries are discarded.

SUMMARY

In this chapter, we examined two very different transport protocols—TCP and UDP. TCP provides connection-oriented reliable transport service. It carries out end-to-end error and flow control using retransmission mechanism for the lost TCP segments. Flow control is based on sliding window mechanism. TCP allows the receiver to dynamically control the size of window at the sending end.

TCP also implements congestion avoidance mechanisms by controlling the amount of user data that is pumped into the IP network. Congestion avoidance is based on feedback implicitly learnt from the network. This feedback is in form of timeouts, and retransmitted acknowledgements. The feedback determines the

size of congestion window and slow start threshold.

UDP provides connectionless transport service to the user entities. The service is unreliable in the sense that there is no acknowledgement, error control, and the UDP datagrams may be delivered out of sequence. UDP is used where the overhead of connection-oriented service is undesirable or where retransmission of lost segments makes no sense, *e.g.* digital voice.

EXERCISES

1. Consider two-way handshake is used for TCP connection establishment. List all the situations when things go wrong and identify the situation(s) that can lead to wrong establishment of a connection.
2. Plot $SRTT(K)$ for $K = 0$ to 10, when
 - (a) $SRTT(0) = 5$ and $a = 0.4$
 - (b) $SRTT(0) = 5$ and $a = 0.85$The consecutive measured values of RTT are 3, 3, 3, 3, 3, 3, 3, 5, 3, 3. Which of the two values of a do you recommend for estimating SRTT?
3. A TCP connection is opened with slow start. Estimate the number of round trip times required to send n TCP segments.
4. Calculate the window size needed to keep the transport channel between two end systems fully occupied. Assume that
 - (a) Round trip time is 80 ms.
 - (b) IP header is 20 octets.
 - (c) TCP segment size is 480 octets.
 - (d) 100 Mbps bandwidth is available to IP packets.
5. For the parameters given in Exercise 4, estimate the time required to reach this state after the connection is established if slow start is adopted.
6. The $cwnd$ was 20 K octets when three duplicate acknowledgements were detected.
 - (a) At what levels will the $ssthresh$ and $cwnd$ be set thereafter?
 - (b) What will be the value of $cwnd$ after next 5 round trip times if duplicate acknowledgements are received continuously?
7. Would you recommend IP checksum to include the payload also? Give justification for your answer.
8. A host has IP address 11 and process port P1. Another host has IP address 12 and process port P2. Can multiple TCP connections be established

between the two ports simultaneously?

9. In addition to having acknowledgement field in the TCP header, ACK bit is also provided. What would happen if the ACK bit were not provided?
10. If the receiver window is 24 kbytes and MSS is 2 kbytes, calculate the time elapses before the full burst of TCP segments amounting to 24 kbytes is released by the sender if slow start is used as congestion control mechanism. Assume round trip delay of 50 ms.
11. If the MSS is 1 kbytes, what is the window size after
 - (a) occurrence of timeout?
 - (b) successfully sending four full windows of bytes after timeout?
12. What is maximum data rate at which a host can send 1000-octet TCP payloads if the packet life time is 100 seconds without having sequence number wrap-around? Assume TCP header of 20 octets, IP header of 20 octets, and Ethernet overhead of 26 octets.

[1](#) Selective acknowledgement (SACK) is an optional feature in TCP.

[2](#) We assume that propagation time across the network is much larger than transmitting time.

[3](#) In fact, for the acknowledgement received for a transmitted TCP segment, cwnd is incremented by one MSS. The net result is doubling the size of cwnd after all the acknowledgements have been received.

21

Network Security

Network security is a complex subject because security, be it of a bank or a network, needs to be addressed in a wholesome manner. A systematic approach for developing the subject is needed for the purpose of learning the subject. In this chapter, we begin with security concerns and end the chapter with protocols that are built into the network to address the security concerns. We adopt the following learning sequence:

- Security requirements
- Algorithms for encryption and generating integrity checksum
- Mechanisms for authentication and message integrity
- Digital signatures and certificates
- Mechanisms for distributing encryption keys
- Network security protocols for the transport layer and the network layer
- Firewalls.

21.1 SECURITY REQUIREMENTS

In the networked world, people transact business using the connectivity provided by the network. But the network is vulnerable to security breach. There are some basic security requirements as specified in RFC 1825 that must be met:

- Confidentiality
- Integrity
- Authentication
- Non-repudiation.

Confidentiality. The contents of a message when transmitted across a network

must remain confidential, *i.e.* only the intended receiver and no one else should be able to read the message. The users, therefore, want to encrypt the messages they send so that an eavesdropper on the network will not be able to read the contents of the message.

Integrity. Data integrity means that the data must reach the destination unadulterated. If any alteration, malicious or accidental, occurs during transmission, the receiver should be able to conclude that an alteration has happened. For example, if A advises the bank to transfer Rs. 1000 to B and if a malicious attempt is made by a third party to divert the amount to C by changing the message, the bank should be able to recognize that the message has been changed.

Integrity of a message is ensured by attaching a checksum to the message. The algorithm for generating the checksum ensures that an intruder cannot alter the checksum or the message.

Authentication. The receiver should be able to authenticate the source, *i.e.* the actual sender is the same as claimed to be. There can be several ways of doing this:

1. The two parties share a common secret code word. A party is required to show the secret code word to the other for authentication.
2. Authentication can be done by sending digital signature. A party can send its digital signature to the other to prove its identity.
3. A trusted third party verifies the authenticity. There can be several ways. One of them is use of digital certificates issued by a recognized certification authority.

We will learn about digital signatures and digital certificates as we go through this chapter.

Non-repudiation. Non-repudiation means that a sender must not be able to deny sending a message that it actually sent. The burden of proof lies on the receiver. The bank in the last example of money transfer should be able to prove that A in fact sent the message to transfer the money to B, if A denies it later.

Non-repudiation is not only in respect of the ownership of the message, the receiver must prove that the contents of the message are also same as the sender sent. Non-repudiation is achieved by authentication and integrity mechanisms.

The study of network security involves:

- Analysis of algorithms for encrypting messages, decrypting messages, and for generating message integrity checksums.
- Examination of security mechanisms for exchange of messages that ensure integrity, authentication and non-repudiation.
- Analysis of network protocols that implement the above mechanisms.

We used similar approach for error control. We studied the CRC algorithm, then the sliding window mechanism and, finally, HDLC protocol.

21.2 CRYPTOGRAPHY ALGORITHMS

There are two broad categories of encryption algorithms:

- Algorithms for ensuring confidentiality of the message
- Algorithms for ensuring integrity of the message.

Algorithms for ensuring confidentiality encrypt a message in such a way that the encrypted message cannot be understood or made use of unless it is decrypted.

Algorithms for ensuring integrity are designed in such a way that if a message is altered while in transit, the receiver is able to detect that the message has been altered. In this case, we are more concerned with the integrity of the message rather than its confidentiality. An intruder may read and understand the message but he cannot alter it. In cryptography both are important, confidentiality and integrity. Therefore, it is most likely that a security mechanism will make use of both the types of algorithms.

21.3 ALGORITHMS FOR CONFIDENTIALITY

An unencrypted message is called plaintext and the encrypted message is called ciphertext in cryptography. Message encryption and decryption for maintaining confidentiality involves use of its two components (Figure 21.1):

- A common algorithm (F) that carries out encryption and decryption.

- A unique key that algorithm (F) uses with the plaintext for encryption. The decryption algorithm may use the same key or a different decryption key.

The algorithm (F) is well known but the key is kept secret. It is possible to have an encryption algorithm that works without a key, but a different algorithm will be required for each user. Use of key overcomes this problem. Each user or group of users is assigned a unique key.

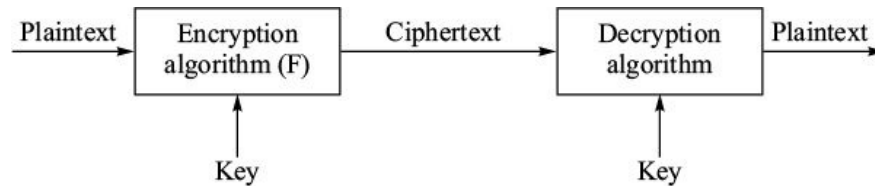


Figure 21.1 Encryption and decryption of messages using keys.

There are of two types of algorithms that encrypt a message:

- Secret key encryption algorithms
- Public key encryption algorithms.

21.3.1 Secret Key Encryption Algorithms Secret key encryption algorithms are symmetric in the sense that the two users share a common secret key for encryption and decryption. Examples of secret key algorithms are as follows:

1. Data Encryption Standard (DES)
2. Advanced Encryption Standard (AES)
3. Triple DES
4. International Data Encryption Algorithm (IDEA)
5. Ron's Code 4 (RC4).

Data encryption standard (DES). DES is a secret key encryption algorithm that encrypts 64-bit block of plaintext using 64-bit key. The key contains 8 parity bits and therefore 56 bits are useable bits. The encryption algorithm is performed in four phases:

1. The 64 bits of the plaintext are shuffled and grouped into two blocks of L_0 and R_0 of 32 bits each.

2. An encryption operation using a key is applied to the shuffled plaintext.
3. The encryption operation is repeated fifteen times more on resulting data after each round.
4. After total sixteen rounds, the inverse of the original shuffling of step (1) is applied to the resulting data to get the ultimate ciphertext.

The key used in each round is derived from the 56-bit key by rotating the bits. High level view of the algorithm is depicted in Figure 21.2. The details of the algorithm are beyond the scope of this book. At the receiving end, the sixteen-stage process is repeated but the keys K_1 to K_{16} are applied in the reverse order.

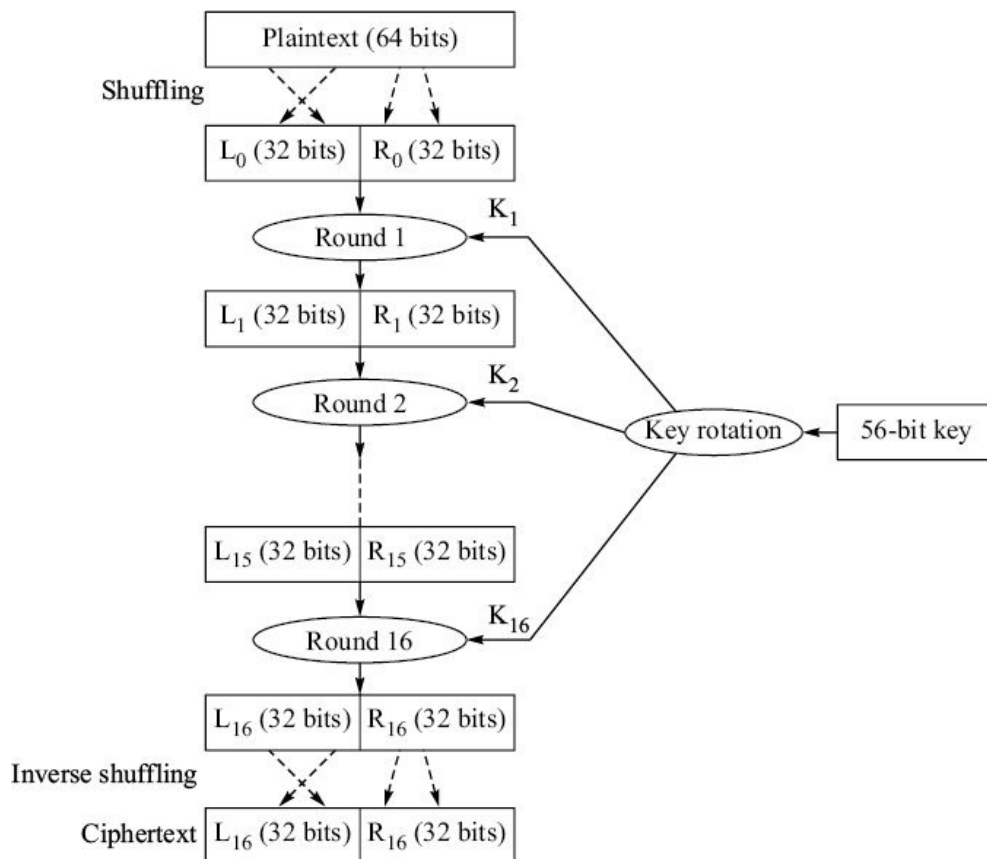


Figure 21.2 DES algorithm for encryption.

For the messages larger than 64 bits, blocks of 64 bit are made and a technique called cipher block chaining is used (Figure 21.3). DES is run in the same manner on 64-bit blocks as shown in Figure 21.2. The initial vector (IV) is a random number to start the algorithm. The receiving side operates the algorithm in reverse fashion.

DES-CBC can be used for integrity check also. The last cipher block, (e.g.

cipher 4 in Figure 21.3) is taken as the checksum of the entire message. We will describe the scheme later when we study integrity check algorithms.

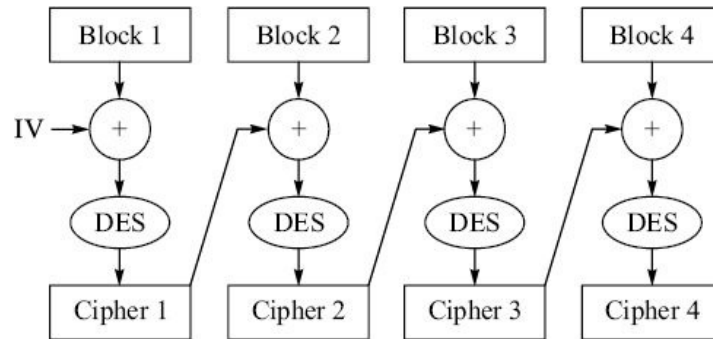


Figure 21.3 DES cipher block chaining (CBC).

21.3.2 Exchange of Secret Key

In secret key encryption, we assumed that the two parties share a common secret key using which they can encrypt and decrypt their messages. Suppose they do not have one. Then sending a secret key on the network or through mail or even on phone is a big security risk. Diffie-Hellman key exchange mechanism establishes a common secret key between two strangers without exchange of the actual secret key between them. We will study another method of secret key exchange using digital signature later.

Diffie-Hellman secret key exchange. The Diffie-Hellman secret key exchange mechanism works as follows:

1. A and B select two large numbers p and g . p is a prime number and $g < p$. These numbers are not secret. A or B can select them and pass onto the other party.
2. A and B pick individually a random number. Let us say A picks x and B picks y . These numbers are secret.
3. A calculates $S_A = g^x \text{ mod } p$ and sends this to B. Similarly B calculates $S_B = g^y \text{ mod } p$ and sends this to A.
4. A and B now can independently calculate the common secret key K which is equal to

$$K = (S_B)^x \text{ mod } p = (g^y \text{ mod } p)^x \text{ mod } p = g^{xy} \text{ mod } p \quad \text{..... at end A.}$$

$$K = (S_A)^y \text{ mod } p = (g^x \text{ mod } p)^y \text{ mod } p = g^{xy} \text{ mod } p \quad \text{..... at end B.}$$

5. Note that secret key K can be calculated only if x and y are known. These random numbers are never sent across by the either party. A and B exchange S_A and S_B and an intruder cannot calculate x and y from S_A and S_B .

EXAMPLE 21.1 If A and B choose $p = 47$, $g = 3$, and A picks a random number $x = 8$ and B picks a random number $y = 10$, show the calculations done by them to get the secret key (K) using the Diffie-Hellman key exchange algorithm.

Solution A calculates S_A and sends it to B. B calculates S_B and sends it to A.

$$S_A = g^x \text{ mod } p = 3^8 \text{ mod } 47 = 28$$

$$S_B = g^y \text{ mod } p = 3^{10} \text{ mod } 47 = 17$$

A calculates the key (K) as

$$K = 17^8 \text{ mod } 47 = 4$$

B calculates the key (K) as

$$K = 28^{10} \text{ mod } 47 = 4.$$

21.3.3 Public Key Encryption Algorithms

Public key encryption algorithms are non-symmetric in the sense that the encryption and decryption keys are different. Each user is assigned a pair of keys —public key and private key. The public key is used for encryption and the private key is used for decryption. Decryption cannot be done using the public key. The two keys are linked but the private key cannot be derived from the public key.

The public key is well known but the private key is secret and known only to the user who owns the key. In other words, everybody can send a message to the user using his (user's) public key. But the user only can decipher the message using his private key. The public key algorithm operates in the following manner

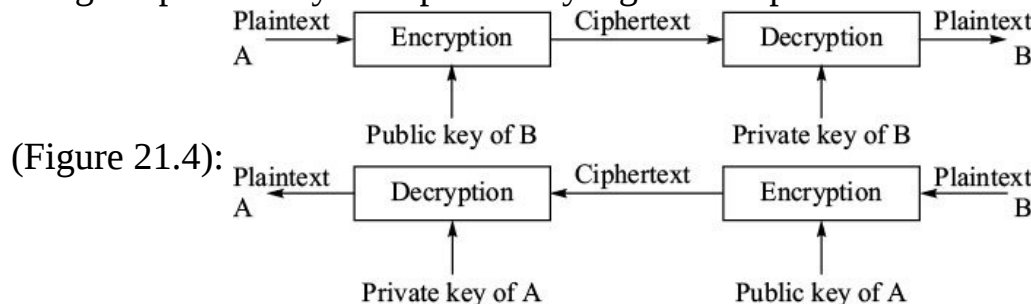


Figure 21.4 Encryption and decryption messages using public/private keys.

1. The data to be sent is encrypted by sender A using the public key of the intended recipient B.
2. B decrypts the received ciphertext using its private key which is known only to him. B replies to A encrypting its message using A's public key.
3. A decrypts the received ciphertext using his private key which is known only him.

Examples of public key encryption algorithms are:

- RSA
- El Gamal
- Digital Signature Standard (DSS).

The first public key encryption algorithm was developed by Rivest, Shamit, and Adleman and is known as RSA after its developers. It is the only widely accepted and implemented public key encryption algorithm. Before we study the RSA algorithm, let us look at some comparative points of the secret key and public key encryption algorithms.

1. Secret key encryption algorithms use key size that can be from 40–168 bits. Public key encryption uses key sizes from 512 to 2048 bits.
2. Secret key encryption algorithms are much faster than public key encryption algorithms.
3. N users require $N(N - 1)/2$ keys for communication among themselves using secret key encryption algorithms as one key is required for every pair of users. They will require $2N$ keys if public key encryption algorithm is used because each user will need a pair of keys.

RSA public key encryption. RSA algorithm for generating public and private keys, and for encryption and decryption is as follows:

1. Select two large prime numbers p and q . p and q are about 256 bits long in practice.
2. Choose a number e such that e is a prime relative to the product $(p - 1)(q - 1)$. Two numbers are relative prime if they do not have a common factor,

e.g. 15 and 8.

3. If $n = p \cdot q$, then the public key is e, n . A plaintext message m is encrypted using public key e, n . The formula given below is used to get ciphertext c .

$$c = m^e \bmod n$$

Here m must be less than n . A larger message ($> n$) is treated as concatenation of messages, each of which is encrypted separately.

4. To determine private key, compute d such that

$$de \bmod \{(p - 1) (q - 1)\} = 1$$

The private key is d, n . The ciphertext message c is decrypted using the formula given below:

$$m = c^d \bmod n.$$

The public key encryption algorithm is based on the premise that number n available in public key cannot be factored and therefore p and q cannot be found out. If p and q are known, decryption key d can be readily determined. It is claimed that a 512-bit product of two prime numbers takes years of computation to determine its factors. The prime numbers p and q are therefore kept more than 256 bits long.

EXAMPLE 21.2 Encrypt plaintext 9 using RSA public key encryption algorithm. Use prime numbers 7 and 11 to generate the public and private keys. Decrypt the ciphertext using the private key.

Solution First we calculate n and $(p - 1) (q - 1)$.

$$n = p \cdot q = 77, (p - 1) (q - 1) = 60$$

Let us choose relative prime e of 60 as 7. Thus the public key is $e, n = 7, 77$. To determine private key, we set $7d \bmod 60 = 1$, which gives $d = 43$

The private key is $d, n = 43, 77$. The ciphertext is given by $c = m^e \bmod n = 9^7 \bmod 77 = 37$

Applying the private key to the ciphertext 37 to get original plaintext, we get $m = c^d \bmod n = 37^{43} \bmod 77 = 9$.

21.4 ALGORITHMS FOR INTEGRITY

The integrity algorithms enable the receiver to check whether the message sent by the sender has been altered in any manner during its transit. In these

algorithms, a cryptographic integrity checksum is calculated and attached to the message by the sender. The receiver recalculates the checksum at its end and compares it with the received checksum. If they are same the message is intact (Figure 21.5).

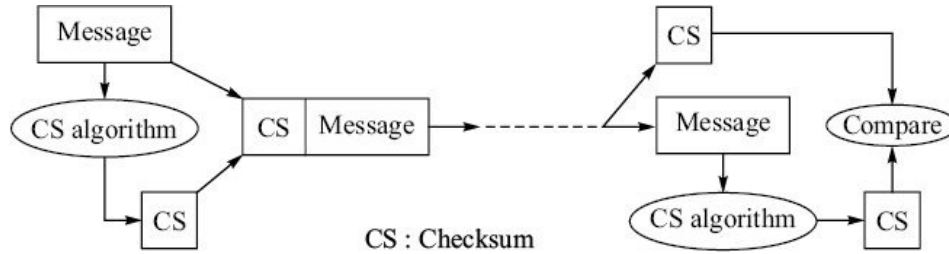


Figure 21.5 Mechanism for integrity check.

Different checksum algorithms use different names for the integrity checksum. For example, Message Authentication Code (MAC), Message Integrity Code (MIC), and Message Digest (MD) refer to the integrity checksum.

The algorithm is designed in such a manner that given a checksum, it is computationally impossible to find an alternate message having the same checksum. Thus an intruder or even the receiver cannot substitute another message having the same checksum. He can, however, certainly substitute the original message with another message and the original checksum with a new checksum. We will see how this is taken care of later.

Note that the confidentiality is not the concern of the integrity checksum algorithm because it does not encrypt the body of the message. Any eavesdropper can read the message sent with the checksum if the message is not encrypted.

Examples of checksum algorithms are Message Digest 5 (MD5), Secure Hash Algorithm (SHA), and DES-CBC. Out of these MD5 is the most important algorithm. MD5 and SHA are similar and use one way hash function. DES-CBC algorithm has already been described. We will shortly see how it is used for message integrity.

21.4.1 MD5 Algorithm

Out of the several versions of message digest algorithms, MD5 is the most popular. The basic operation of MD5 is depicted in Figure 21.6. It operates on 512-bit blocks of data. Messages not multiple of 512 bits are padded with

- a string consisting of 1 followed by zeroes, and
- 64-bit integer that indicates the length of the original message, to make the

length of the composite message multiple of 512 bits.

The cryptographic checksum for integrity test is called message digest in MD5. The Message Digest (MD) calculation begins with an initial value of the message digest, which is combined with the first block of 512 bits using a complex transformation algorithm that generates a new value of the digest (Figure 21.6). The transformation algorithm consists of 4 passes and in each pass a different transformation function and a different set of parameters are used.

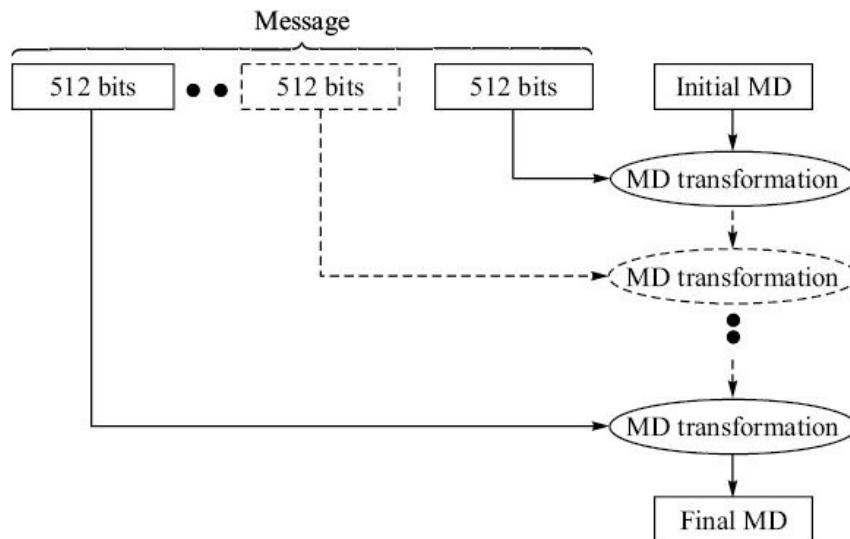


Figure 21.6 Generation of message digest using MD5.

The new value of the digest as obtained above combined with the next 512-bit block in the similar fashion. The process is repeated on each 512-bit block till the final value of digest is obtained from the last block of the message. The digest is 12-bit long for any message length.

21.4.2 DES Cipher Block Chaining (DES-CBC) DES cipher block chaining described in sub-section 21.3.1 can be used for check of message integrity. The message is encrypted using DES algorithm as shown in Figure 21.3. The output of the last message block, called CBC residue, (cipher 4 in Figure 21.3) is used as checksum for integrity test. The checksum, in this case, is called Message Integrity Code (MIC). Only the plaintext and MIC are transmitted to the other party (Figure 21.7). The receiver uses its secret key with the received unencrypted message to generate MIC at its end. The computed MIC and

received MIC are compared and if they are different, either the message has been altered or sender has used a different secret key for generating the MIC.

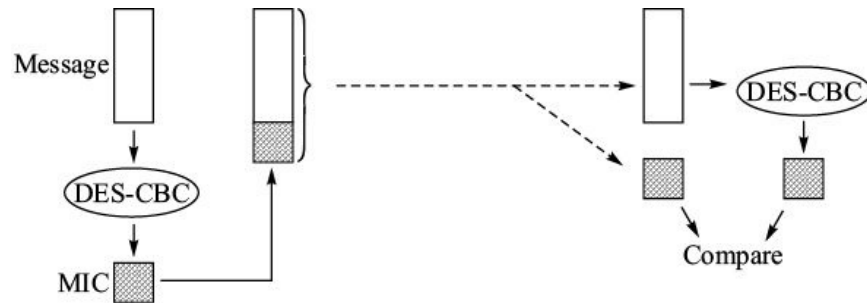


Figure 21.7 DES-CBC message integrity.

In this scheme, the message is sent unencrypted. If the message is also encrypted, the last block of the encrypted message and the attached MIC would be same. Therefore the integrity check loses its purpose.

21.4.3 Keyed MD5

In section 21.4.1, we saw how MD5 can be used for checking integrity of the message. We can add authentication to integrity check by using a secret key. The secret key is actually a secret code used for identification. It is not used for encryption. The method is commonly referred to as keyed MD5, and is shown in Figure 21.8.

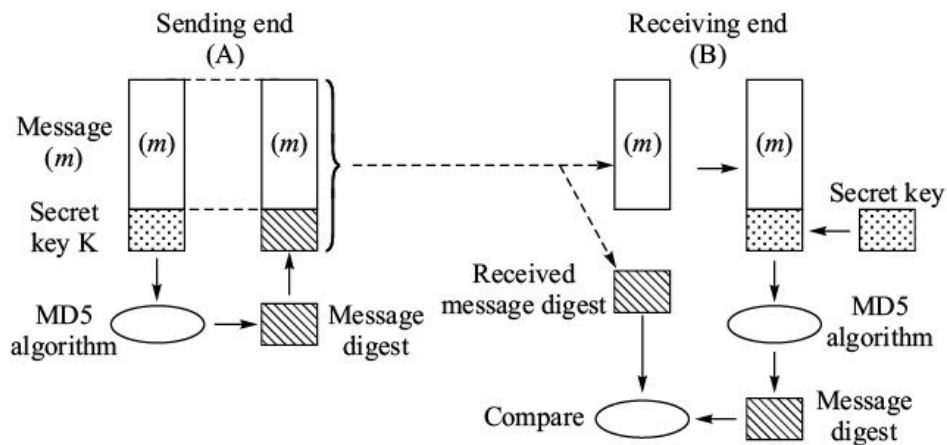


Figure 21.8 Keyed MD5.

Suppose the secret key is K . The key K is attached to the end of the message m . Then the sender (A) runs MD5 algorithm on concatenation of K and the message m to get the digest $MD(m, K)$. The message digest and the original message m are sent to the other party (B). Note that key K is not sent.

B, the receiver, has his own copy of the key K . He appends the key to the received message m and runs MD5 on the concatenation (m, K) . If the result matches the digest $MD(m, K)$ received from the other end, then B concludes that:

- the message has been sent by a party who owns the key K . Therefore, the party must be A,
- the message is unadulterated.

21.5 BASIC AUTHENTICATION MECHANISMS

The cryptography algorithms described above can be used as tools for implementing security mechanisms and policies. In this section, we will examine the ways these tools can be applied for authentication. While going through this section, the readers will realize these tools can be used in numerous ingenious ways for achieving the security objectives. The intruder also uses ingenious ways to defeat them.

In the following sections, we use the expression $K(M)$ to denote a message M that has been encrypted using key K .

21.5.1 Authentication Using Secret Key Figure 21.9 shows a scenario where A and B need to authenticate each other. They share a common secret key K_{AB} between them. A sends request for starting a session indicating its identity (Id) to B. B chooses a random number R_1 and challenges A to encrypt it using the secret key. There is no need of encryption at this stage. When A receives this message, it uses the secret key K_{AB} to encrypt the random number R_1 and sends it back to B. B on its own encrypts the random number R_1 it sent to A and compares it with the received encrypted message. If they are same, it implies A has the same K_{AB} as B has and thus, A is authenticated.

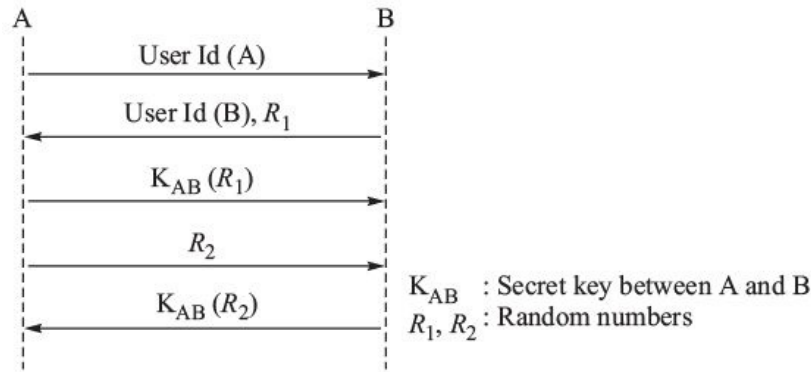


Figure 21.9 Authentication using secret key.

Random number R_1 ensures that the reply from A is in respect of the challenge B sent to A. If a fixed number is used instead of random number, an intruder C can copy the reply from A. C can later pose as A and use the copied reply to establish a session with B. This type of attack is called replay attack. By attaching a different random number every time, B ensures that the old replies become obsolete. An alternative to random number is the time stamp.

The above process authenticates A to B. A also undertakes the same process to authenticate B. A and B can exchange messages using the secret key thereafter.

It may appear that the number of steps of the mechanisms shown in Figure 21.8 can be reduced. Let us do that. Figure 21.10 shows the mechanism that achieves authentication of A and B in three steps. A sends the random number R_1 . B encrypts it using secret key K_{AB} and returns encrypted R_1 along with random number R_2 . R_2 is not encrypted. A verifies authenticity of B by checking encrypted R_1 sent by B. A returns encrypted R_2 to prove its authenticity to B.

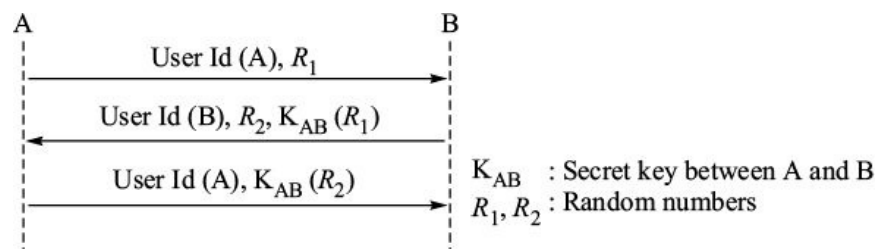


Figure 21.10 Shortened authentication mechanism using secret key.

The mechanism should work in a non-hostile environment. But an imposter C can easily deceive B into believing that he is A. Let us see how it is done (Figure 21.11).

- C poses as A and sends its Id to B with random number R_1 .
- B responds with encrypted R_1 , and challenges A (C) to encrypt R_2 .
- C opens second parallel session again posing as A and challenges B to encrypt R_2 that it received from B in the first session.
- As usual B encrypts R_2 and returns it to A (C) with another random number R_3 for A to encrypt.
- C having obtained the encrypted R_2 from B, resumes the first session and returns the response to B's challenge for encrypting R_2 . C aborts the second session.
- On receipt of this response, B is convinced that C is A and continues with the first session.

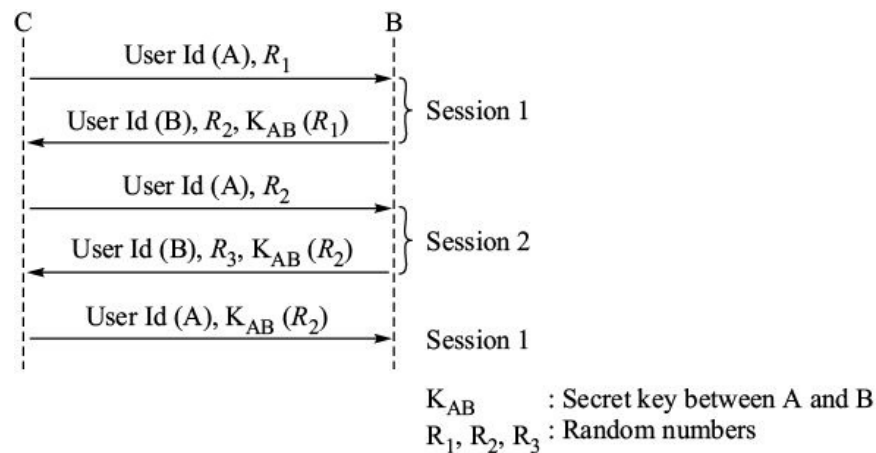


Figure 21.11 Reflection attack by imposter C.

Here C reflects the challenge posed by B, back to B and gets the information using second parallel session. This type of attack is called reflection attack.

Thus designing mechanisms is harder than it appears to be. The general guidelines are as follows:

- The originator of session request should prove his identity first.
- The two ends should use a different key for each direction of communication.
- Use random numbers and time stamps to identify the obsolete responses.
- Parallel sessions are risky. Link a response to the session Id.

21.5.2 Authentication Using Secret Key with Third Party In the

above example, we have assumed that A and B share a secret key, which may not always be the case. In such cases mediation by a trusted third party (C) is required. Every user has a unique secret key that it uses for communication with C. When A wants to communicate with B, it approaches C to allot him another secret key (S) called session key. Session key has a life time and valid only for communication between A and B. The mechanism is illustrated in Figure 21.12.

- A approaches C with its own and B's user Id.
- C sends two messages to A— M_{CA} and M_{CB} .

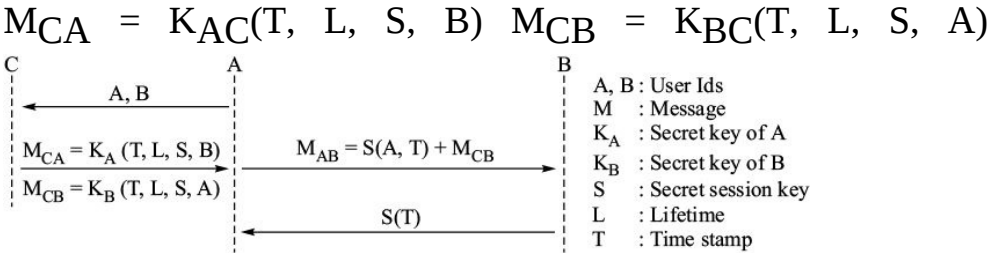


Figure 21.12 Authentication with the help of third party.

Message M_{CA} is encrypted using A's secret key K_{AC} and contains session key S having lifetime L, time stamp T, and user Id of B. The user Id B is included as A may have several ongoing sessions with other users. This key is to be used only for session with B.

- Message M_{CB} is encrypted using the B's secret key K_{BC} . It contains the same time stamp T, life time L, and session key S. It also contains A's user Id. This message is meant for B but is sent to A for handing over to B.
- A decrypts the message M_{CA} and obtains the session key S. It generates a message $M_{AB} = S(A, T)$ for B using secret session key S.
- A cannot decrypt M_{CB} as it is encrypted using B's secret key K_{BC} . A concatenates M_{CB} to $S(A, T)$ and sends the concatenated message to B.
- B first decrypts M_{CB} and obtains the secret session key S, user Id A and time stamp T. It decrypts the other message M_{AB} using session key S. It

- matches the time stamp and A's user Id. If they match, A is authenticated.
- B confirms this to A by returning $M_{BA} = S(T)$ to A.

21.5.3 Authentication Using Message Digest

Figure 21.13 shows an example of authentication using message digest. A and B share a common key K . A and B identify themselves by showing to the other party that they possess this key. Here we use time stamp instead of random number to take care of replay by the intruder.

- A sends the time stamp T_1 and the digest of concatenation the secret K and time stamp T_1 . Note that the secret itself is not sent.
- B is already in possession of the secret K , and runs the message digest algorithm on the received time stamp T_1 and K . If the digest so produced matches with the one received, A is authenticated.
- B follows the same process to authenticate itself to A.

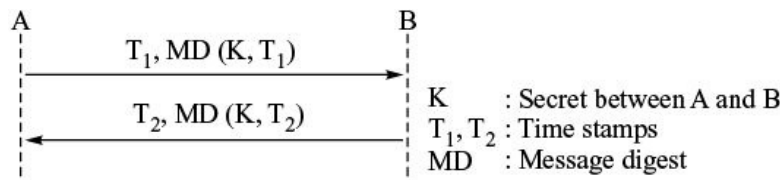


Figure 21.13 Authentication using message digest.

21.5.4 Authentication Using Public Key

In this case, A sends a random number encrypted with B's public key (Figure 21.14). B decrypts the number using its private key and returns the random number to A using A's public key. A decrypts the received message using its private key and verifies that the random number is same. B seeks authentication from A in the similar manner.

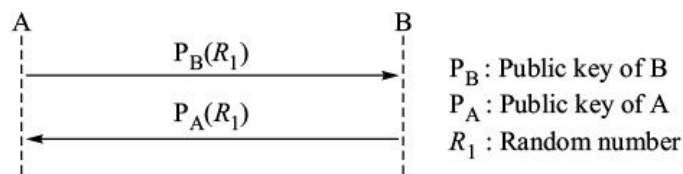


Figure 21.14 Authentication using public key.

21.6 MECHANISMS FOR ENSURING

MESSAGE INTEGRITY

Message integrity is to be ensured in such a way that the

- message must not have been altered;
- message must not have been substituted; and
- receiver must be able to prove that the received message is what the sender has actually sent.

A message can be altered or substituted by an intruder or the receiver. Even the sender at a later date may repudiate sending the message. The message integrity mechanisms should take care of all these issues. The usual way of ensuring these aspects in case of hard copy documents is to take signature of the party who submits the document. As an additional measure of surety, at times a third party verifies the documents and the person who submits them.

In a networked scenario, we use digital signatures and third party verification. There are digital certificates too. We will discuss them later in this chapter. First we concentrate on digital signatures.

21.7 DIGITAL SIGNATURE

We are familiar with written signatures that are used for authenticating documents. The person who signs it takes the responsibility of the content present in the document. In the networked world, there is a similar need for digital signatures. Authenticating messages using digital signatures requires the following conditions to be met:

- The receiver should be able to verify the claimed identity of sender. For example, an imposter may advise a bank to transfer money from the account of another person. The bank should be able to verify the identity of the user before acting on the advice.
- The sender should not be able to repudiate the contents of the message it sent at a later date. For example, a person having sent advice to transfer money from his account should not be able at a later date to repudiate the content of the message sent by him. The bank should be able to prove (a) that he and only he sent the advice and (b) that the contents of the message are

unaltered.

- The receiver should not be able to alter the message or concoct the message himself on behalf of the sender. This requirement is important to protect the interests of the sender.

21.7.1 Digital Signature Using Private Key Public and private keys possess the property that allows mutual reversal of their roles, i.e. d , the decryption key as defined in section 21.3.3 can be used for encryption and e , the encryption key can be used for decryption. This property can be used for generating messages with digital signature. Suppose A encrypts a message using its private key Q_A and sends it to B (Figure 21.15). Since the message has been encrypted using A's private key, it bears A's signature on it. B already knows A's public key P_A and decrypts the message received from A.

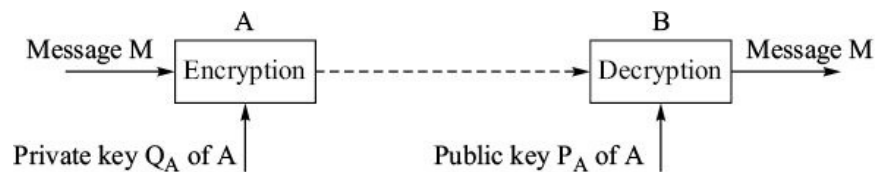


Figure 21.15 Digital signature using private key.

All the requirements (a), (b), and (c) of digital signature are met in this mechanism. The received encrypted message is decrypted using A's public key. So the message is from A. B keeps a copy of the encrypted message so that A cannot later refuse sending it. B cannot alter the encrypted message or on its own concoct the encrypted message without A's private key.

The mechanism still has one weakness. Any intruder knowing A's public key can intercept the message. We need to take care of confidentiality aspect also. This is done simply by second encryption using B's public key (Figure 21.16). A encrypts the message to B using its private key first and then using B's public key. B decrypts the received message using its private key and then using A's public key.

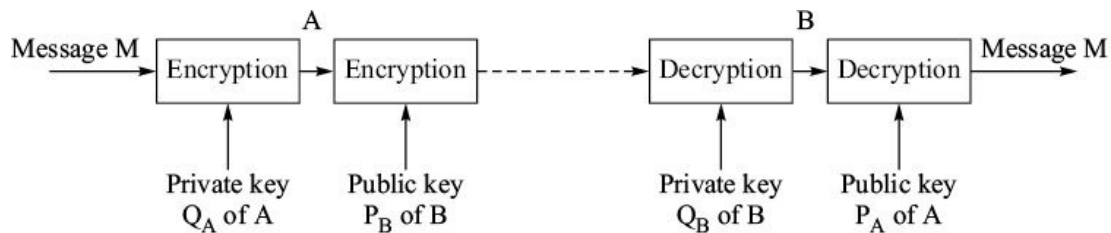


Figure 21.16 Digital signature using public and private keys.

RSA algorithm for encryption and decryption using public and private keys is relatively very slow. The above method, though robust, cannot be used for putting signature on a user data message.

21.7.2 Digital Signature Using Private Key and Message Digest As mentioned above, public key mechanisms for digital signature offer elegant but slow solution. An alternative is to apply private key digital signature to the digest of the message rather than message itself. Figure 21.17 shows the basic mechanism:

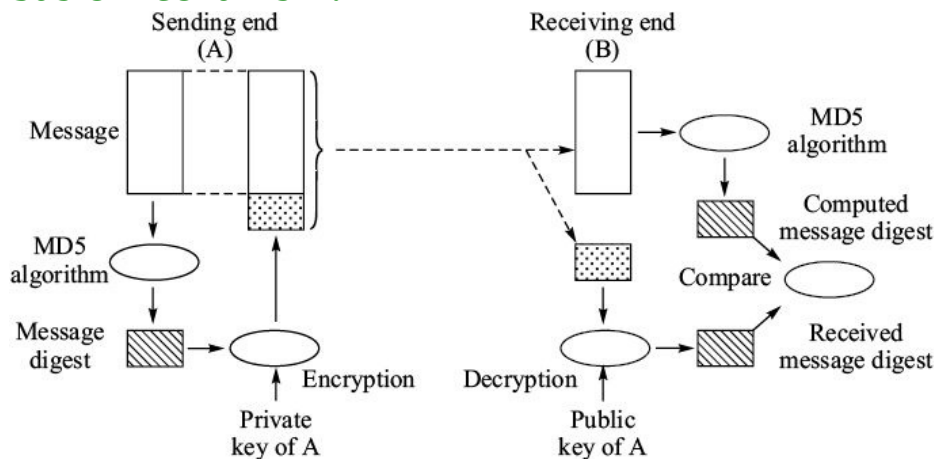


FIGURE 21.17 Digital signature using private key encryption on message digest.

1. The message digest, which is much shorter in length than the message, is encrypted using private key of the sender (A).
2. Message digest and the message are sent to the other end (B). B computes message digest of the received message part. The encrypted message digest part is decrypted using A's public key. The received message digest and the computed message digest are compared to ensure they are same.
3. All the three requirements of digital signature are met.
 - (a) Message digest bears A's signature. Therefore, A cannot repudiate his

signature and the contents of message digest. Message digest is one way hash function and therefore he cannot repudiate the original message also.

(b) B cannot alter the content of the message or concoct a message on its own because it cannot generate the message digest bearing A's signature.

The basic scheme can be supplemented with time stamp, second stage of encryption using secret key to ensure confidentiality.

21.7.3 Digital Signature Using Third Party and Secret Keys Let us assume that there is a central authority (C) who has the trust of every one including the legal system. Everyone in system communicates through C and shares its secret key with C. Figure 21.18 shows an example.

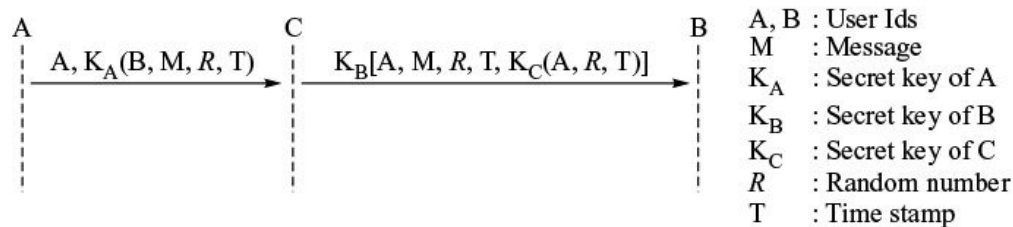


Figure 21.18 Digital signature using third party.

1. A sends its message M encrypted using secret key K_A to C. It concatenates a random number R and time stamp T with the message before encryption. It sends its user Id also to C so that C knows that the message is from A.
2. C decrypts the message using the key K_A . It re-encrypts the message using B's secret key K_B . It concatenates its digital signature to the message from A before encryption. The digital signature of C in this case is secret key encryption of A, T, and R. The secret key K_C belongs to C and no one other than C knows it.
3. B receives the message and decrypts it using its secret key K_B . It retains $K_C(A, T, R)$ for future use. If A repudiates that it sent the message, B can produce $K_C(A, T, R)$ and ask C to verify this.

21.8 MANAGEMENT OF PUBLIC KEYS THROUGH THIRD PARTIES

In the last section, we examined the use of public key mechanisms for authentication and confidentiality. We assumed that the public key of the other party, we wanted to communicate with, was always readily available. It is not so simple. Suppose A wants to know public key of B. B puts its public key on its web page. However, when A attempts to access the web page, an intruder fakes B's web page that advertises the intruder's own public key as B's public key. A uses the key and sends the message which lands straight to the intruder who easily decrypts it using his private key. Therefore, distribution and management of public keys are two important issues that need meticulous planning.

21.8.1 Digital Certificate

One way of managing the task can be to have a public key distribution centre. An organization specially designated by the government for this purpose will have the trust of the users. But the solution may not be scalable if the key distribution is done for every transaction. A possible alternative is to use digital certificate as described below:

1. A signed certificate is issued by a designated authority, called Certification Authority (CA).
2. The certificate contains a user's Id, his public key, CA's Id, and validity period. The certificate thus binds the user Id and his public key.
3. The certificate bears digital signature of the CA in the sense that its digest is encrypted using CA's private key. CA's public key is well known.
4. When a user A wants to advertise its public key, it obtains a signed certificate containing A's public key from the CA. A advertises his public key with this certificate on its website.
5. When B wants to verify A's public key, he decrypts the advertised digest using CA's well-known public key. He also runs the certificate through the message digest algorithm to get the digest and compares the two digests. If they are same, A's public key is authentic.

Further modalities of communication between A and B are mutually decided by them.

- B can possibly send its digital certificate to A and then both can communicate with each other using RSA algorithm for encryption and decryption.

- Since RSA is slow compared to secret key, B can send its secret key to A using A's public key. Then communication can follow using symmetric secret key algorithm. This method is used in SSL handshake protocol described later.

Binding user Id and his public key is one application of digital certificates. Digital certificates can be used in number of ways. It can bind an attribute to a public key. In other words, the certificate certifies that the owner of the public key possesses that attribute. For example, it may certify that the owner is above 18 years of age and is thus eligible to visit an adult site.

21.8.2 X.509

ITU has standardized the format of digital certificate in its recommendation X.509. IETF's version of X.509 is given in RFC 3280. The basic fields of the X.509 certificate are:

- Version : Version of X.509
- Serial number : Serial number of the certificate for its identification
- Signature algorithm : The algorithm used for signing the certificate
- Issuing agency : Name of the CA
- Validity period : Start and ending times of the validity period
- Subject's name : The entity whose key is being certified
- Public key : Subject's public key and algorithm to be used
- Issuer's Id : Issuer's optional Id
- Subject's Id : Subject's optional Id
- Extensions : Several extensions
- Signature : Digest of above encrypted using CA's private key

21.8.3 Certification Authority Hierarchy To meet the requirements of millions of users, large number of CAs spread across the continents will be required. With such large number of practising CAs, users may need authentication of CAs themselves. For example, a user in India may not have heard of a CA in Brazil. Therefore, another CA may be required to prove legitimacy of a CA and so on. Thus, a CA may produce a chain of certificates to prove its legitimacy. The unending

chain of certificates can be restricted by creating a hierarchy of certification authorities (Figure 21.19). The next higher certification authority is called Regional Certification Authority (RCA). At the apex we have the root organization IPRA (Internet Policy Registration Authority). Thus a chain of trusts is required to be built for digital certification. The user A in the last example can attach all the certificates (own, CA's, and RCA's) in the first instance itself to save B the trouble of getting these from the concerned authorities.

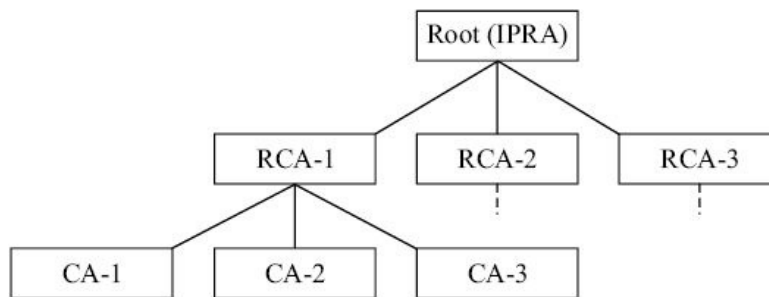


Figure 21.19 Hierarchy of certification authorities.

21.8.4 Revocation of Certificates

A digital certificate may need revocation or cancellation before expiry of its validity. The need for revocation may arise because a user may suspect that someone has discovered his private key or a user may no longer need the certificate. The CA issues a Certification Revocation List (CRL) which is a digitally signed list of certificates that have been revoked. It is made available on public bulletin board and updated regularly. To keep this list from growing indefinitely, the revoked certificates after expiry of their validity period are removed from CRL as they are no longer useable.

21.9 TRANSPORT LAYER SECURITY

The mechanisms for proving security can be implemented in the application layer, the session/presentation layer or the network layer. Building security at the application layer has some negative aspects. Any changes in the security algorithms/mechanisms will entail carrying out changes in all the applications and old versions of the applications may become obsolete. Building security

protocols at the session and presentation layers decouple the applications from security protocols.

One of the most widely used security protocol at the session layer is Secure Socket Layer (SSL). It later formed the basis for the Internet Standard ‘Transport Layer Security’ (TLS) which is described in RFC 2246.

21.9.1 Secure Socket Layer (SSL) Secure Socket Layer (SSL) is sandwiched between the application and the TCP layer (Figure 21.20). SSL provides transport layer services to the applications except that these services are secure, that is the sender can open connection and deliver data octets for transmission. SSL will get these octets to the receiver with required privacy, integrity, and authentication. At times, SSL is found embedded in the applications. For example, Netscape and Internet Explorer browsers come equipped with SSL.

Before we proceed with description of SSL, we need to understand two important concepts—SSL session and SSL connection.

SSL session. An SSL session is an association between a client and a server. A session is established using handshake protocol described later. Security parameters (keys, algorithms) negotiated during session establishment are used for encryption and authentication of messages exchanged during a session.

SSL connection. An application may require multiple TCP connections to be opened. For example, each HTTP operation (e.g. getting a new page from server) requires a new TCP connection to be opened. These multiple TCP connections are mapped to multiple SSL connections that are associated with a common SSL session. Thus, there can be multiple SSL connections in an SSL session. SSL connections associated with a session use the same negotiated security parameters and thus eliminate need for handshake protocol for every SSL connection.

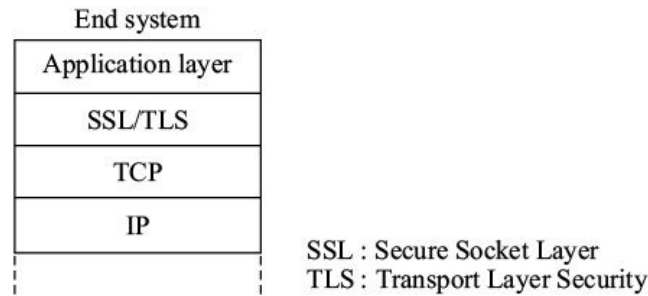


Figure 21.20 Secure Socket layer (SSL).

21.9.2 SSL Architecture

SSL consists of two sublayers (Figure 21.21):

- SSL record protocol sublayer,
- Upper sublayer consisting of
 - handshake protocol,
 - Cipher Change Spec Protocol (CCSP), and
 - alert protocol.

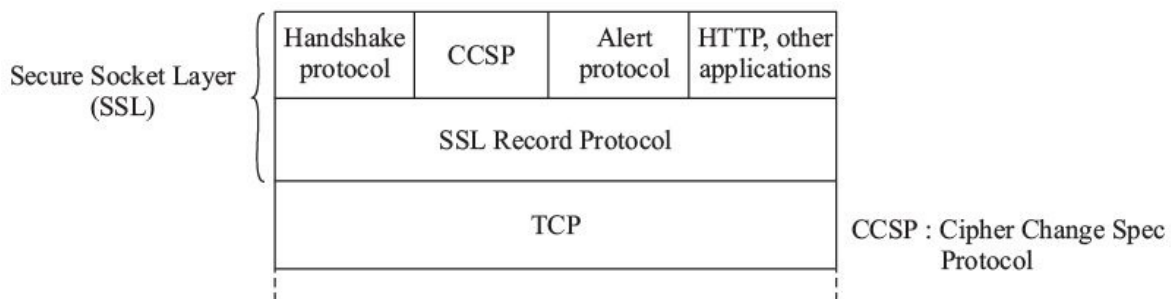


Figure 21.21 SSL architecture.

SSL record protocol provides security services to upper layer protocols which include HTTP and other application protocols. It carries out the fragmentation, compression (if required), authentication, and encryption of the application data. It also attaches a header to the encrypted data. The header defines whether the content is handshake protocol, CCSP, alert protocol or application data. Handshake, alert, and CCSP protocols are for session establishment and session management.

21.9.3 SSL Record Protocol

SSL record layer uses services of the TCP layer below it. It provides the following basic services:

- Confidentiality by carrying out encryption using symmetric keys.
- Message integrity using shared secret key and MD5.
- Compression¹ of user data.

Figure 21.22 summarizes the actions taken by SSL record protocol when it receives user data.

1. The data received from the application or the upper layer of SSL protocol is fragmented into data units having maximum size of 2^{14} bytes.
2. Fragmented data units are compressed using a compression algorithm.
3. The data integrity check generated using checksum based on MD5 or SHA-1 hash algorithms is attached to the compressed data.
4. Bulk encryption of the data using the agreed symmetric keys, DES, triple DES, IDEA, RC4 or other similar algorithms is carried out.
5. A header is attached to the encrypted data. The header contains content type (higher layer protocol, 8 bits), version (16 bits), and length of the compressed fragment (Figure 21.22). The higher layer protocol types are:
 - (a) Handshake protocol
 - (b) Change cipher spec protocol
 - (c) Alert protocol
 - (d) Application data.

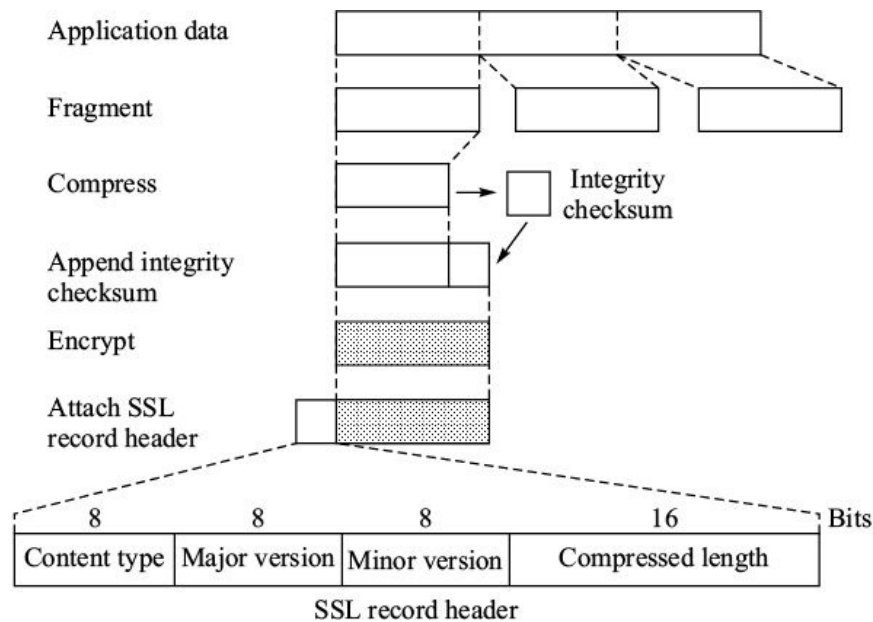


Figure 21.22 Operation of SSL record layer.

Note that no distinction is made among various user applications. The SSL record data unit with header is handed over to the TCP layer.

21.9.4 Handshake Protocol

SSL record protocol needs symmetric keys for integrity checksum and encryption. There is need to decide the algorithm to be used. These are established as a part of SSL handshake protocol. The handshake protocol establishes

- the authenticity of the communicating entities,
- a secret key to derive other keys, and
- cipher suite for encryption and integrity.

The communicating entities are referred to as server and client. The server is always authenticated but, as an option, the server may request client to send its authentication certificates also. Typical exchange of messages between the server and the client during the handshake for establishing a session is given below.

Client to server. The client sends ClientHello that contains

- version of TLS/SSL,
- session Id,
- random number (28 bytes) plus time stamp (4 bytes),
- list of supported cipher suites (encryption algorithms, integrity check algorithms, hash functions, etc.), and
- list of compression methods (optional).

Server to client. The server replies with ServerHello that contains

- session Id, initial values, Diffie-Hellman's part secret key, etc.,
- selected cipher suite and compression method,
- certificate chain for the public key, and
- request for client's certificates (optional).

Then it sends ServerHelloDone.

Client to server. The client sends

- ClientKeyExchange that contains encrypted key using server's public key,
- ChangeCipherSpec that indicates that it is changing over to the agreed cipher,
- Client's certificates (optional).

Then it sends ClientFinished message which carries integrity checksum of all the messages sent so far by both sides.

Server to client. The server sends ChangeCipherSpec to indicate that it is changing over to agreed cipher. This followed by ServerFinished message that contains integrity checksum of all the messages sent so far by both the sides.

Server's ability to decrypt the key and construct the ServerFinished message authenticates the server to the client.

21.9.5 Change Cipher Spec Protocol Change cipher spec protocol is the simplest protocol of the three and consists of only one indication. It indicates to the other entity that the cipher suite is being changed to the cipher suite agreed at the time of handshake. Cipher change indication is given by sending SSL Record Protocol data unit with payload protocol type as cipher change spec and the payload is merely binary 1.

21.9.6 Alert Protocol

The alert protocol is used for management of SSL session and to generate alert messages. The alert messages are compressed and encrypted as the other application messages. Each alert message consists of two bytes—the first byte indicates the severity of the alert and the second byte indicates type of the alert. The severity has two levels, 'Fatal' and 'Warning'. In case of fatal alert, the connection is immediately terminated. Other connections of the session may, however, continue. But new connections cannot be established in the same session. Warning alert is a management message, *e.g.* notify_close is an indication that the sender will not send any more messages on this connection.

21.10 IP SECURITY (IPSEC)

There are people who believe that the security should be built in the network rather than the applications. The Internet in any case needed security mechanisms to be built into its architecture for providing services like Virtual

Private Networks (VPN). In 1994, the Internet Architect Board (IAB) decided that the Internet needed to provide security at network level. As a result, IPSec architecture evolved (RFC 2401). It made IPSec mandatory for IPv6. The design of security capabilities is such that these capabilities can be built into existing IPv4 as well. IPSec allows users to:

- select from variety of encryption algorithms. It does not prescribe one algorithm for all.
- select from menu of security services such as access control, authentication, integrity check, protection against replay, *etc.*
- build security at various levels of network granularity. Security can be applied for a TCP connection, or between two secure routers, or at the end points of a tunnel.

The main point to note is that the security mechanisms are built at the IP layer (network layer) so that secure services are provided to all the applications.

21.10.1 Components of IPSec

Security is built into IP network using two additional IP headers:

- Authentication Header (AH)
- Encapsulating Security Payload (ESP) header.

Authentication header provides access control, message integrity, and protection against replay. It does not carry out encryption. ESP supports all these services plus encryption. There is another part of IPSec that deals with management of encryption keys. It is called ISAKMP (Internet Security Association Key Management Protocol). It is beyond the scope of this book and is therefore not covered here.

21.10.2 Security Association

To build security between two points (hosts, routers, gateways), the two points must agree on the key, authentication mechanism, encryption mechanism, and other operational parameters (e.g. lifetime of a key). A set of such agreements is termed as Security Association (SA). SA database contains the following negotiated parameters and agreed mechanisms of each SA:

1. Sequence number counter for generating 32-bit sequence number for the AH and ESP headers.
2. Sequence counter overflow flag that indicates that the sequence number counter has overflowed and SA must be re-established.
3. Anti-replay window that defines the sequence numbers that are acceptable.
4. AH integrity-check algorithm, keys, lifetime of key, and other parameters.
5. ESP encryption algorithm, integrity-check algorithm, key, Initialization Values (IV), key lifetime, and other operational parameters.
6. Lifetime of an SA that defines the time after which the SA will expire.
7. Other parameters relating to type of transport connectivity.

A security association is identified by Security Parameter Index (SPI), the destination address and security protocol identifier (AH or ESP). SPI is part of AH and ESP headers, the destination address and security protocol type are available in the IP header. When an IP packet is received, the receiver determines its SA and handles the packet in the manner defined therein. The SA defines for the receiver the key and the algorithms to use as mentioned above.

Security association is defined independently for each direction of communication. Thus for a two-way connection, two SAs are required.

21.10.3 Authentication Header (AH) Authentication header is designed to provide payload integrity and authentication of source of origin. It optionally provides protection against replays. It does not support encryption of data and therefore it is useful when confidentiality is not needed. It is defined in RFC 2402.

Authentication header is identified by next header field of the preceding header. The next header field for the authentication header is 51. Format of the authentication header is shown in Figure 21.23.

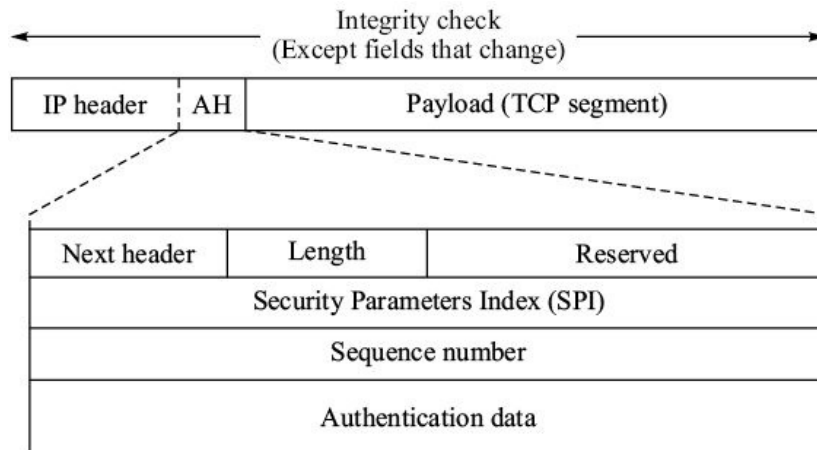


Figure 21.23 Format of authentication header.

Its various fields are as follows.

Length (8 bits). It is used for specifying length of the authentication header as number of 32-bit words. The indicated value is the actual length in 32-bit words minus 2.

Security parameters index (32 bits). It is random value that identifies SA for this packet. It is used in conjunction with the destination address on this packet to identify the SA.

Sequence number (32 bits). This field contains a monotonically increasing counter value for protection against replay. When SA is established, the counters at the sending and receiving ends are initialized to zero. SA is re-established when the counter reaches $2^{32} - 1$.

Authentication data (Variable). It contains the integrity data for this packet. Those fields that change during transit (e.g. TTL) are excluded from the integrity check. The integrity data is in the form of digital signature. The algorithm for digital signature is negotiated during SA establishment. Default integrity check is based on keyed MD5.

21.10.4 Encapsulating Security Payload (ESP)

ESP is designed to provide confidentiality, source authentication, message integrity, and anti-replay services. The set of required services is negotiated at the time of establishment of the SA. ESP is defined in RFC 2406.

ESP consists of ESP header, encrypted IP payload, and authentication data. ESP header is always the last extension header in the daisy chain of the IP extension headers. Then follows rest of the IP packet and finally the

authentication data at the end of the IP packet (Figure 21.24a). ESP header is identified by next header field of the preceding header. The next header field for the authentication header is 50. Format of ESP is shown in Figure 21.24b. Its various fields are as under.

Security parameters index (32 bits). It is a random value that identifies SA for this packet. It is used in conjunction with the destination address on this packet to identify the SA.

Sequence number (32 bits). This field contains a monotonically increasing counter value for protection against replay. When SA is established, the counters sending and receiving ends are initialized to zero. SA is re-established when the counter reaches $2^{32} - 1$.

Encrypted data (Variable). It is the payload data encrypted using the algorithm associated with the SA. Default encryption algorithm is DES-CBC.

Padding/Pad length (Variable). These fields contain padding octets and their number so that multiple of 4-octet boundary is achieved.

Next header (8 bits). It contains the payload type field (e.g. TCP). This field acts as a delimiter and indicates beginning of the authentication data field.

Authentication data (Variable). It contains message integrity checksum value computed over the ESP minus the authentication data field. DES-CBC (cipher block chaining) method is used.

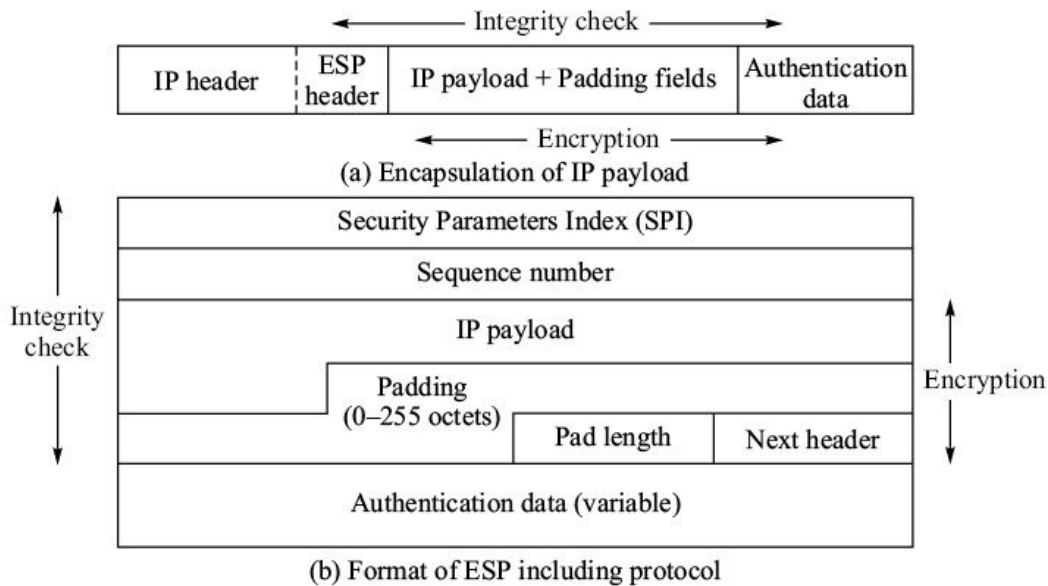


Figure 21.24 Format of ESP.

21.11 FIREWALLS

We will conclude the discussion on network security with an overview of firewalls. As the name suggests, firewall is protection build around network so as to check ingress and egress of data packets across the firewall (Figure 21.25). Firewall is a specially programmed router. It is programmed to filter packets that flow through it. For example, it may discard IP packets addressed to a particular address or TCP port.

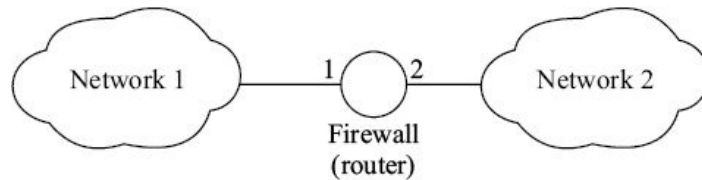


Figure 21.25 Firewall.

If the security algorithms we discussed are implemented, there may not be need of the firewalls. But in reality it never happens. We will always have some legacy networks or applications that are vulnerable and these will require implementation of firewalls.

There are two categories of firewalls:

- Filter-based firewalls
- Proxy-based firewalls.

21.11.1 Filter-Based Firewalls

Filter-based firewalls are simple and are widely deployed. A filter-based firewall router is configured with a table of addresses that decides whether a packet will be forwarded or not. The address may not merely be the destination or source IP address. In general, it may include TCP/UDP ports as well. Table 21.1 shows an example of filter definition. In this case, the firewall blocks all outgoing packets with source address 128.5.0.0 to TCP port 25 (e-mail server). It also blocks all incoming packets with destined for FTP (port 21), Telnet (port 23), and TFTP (port 69). ‘*’ indicates any address/port.

TABLE 21.1 Filtering Table of a Firewall

Interface of arrival	Source IP	Destination IP	Protocol	Source port	Destination port
	*	*		*	

2	128.5.0.0	*	TCP	*	21
1			TCP		25
2	*	*	TCP	*	23
2			UDP	*	69
	*	*		*	

The filter-based approach is simple but has the following limitations:

- The number of well-known ports keeps growing. Therefore, the filter is to be continuously updated.
- Port numbers can be assigned dynamically and users can choose their own port numbers. Thus, tracking port numbers is impossible for filter configuration.
- The firewall can be penetrated using tunneling wherein an IP packet is encapsulated in another IP packet.

Alternative to above is to specify filter that allows certain IP packets destined to specific hosts, networks and application ports and blocks all the rest.

21.11.2 Proxy-Based Firewalls

Proxy is a process that sits between a client and server. To the client, it appears to be the server and to the server it appears to be the client (Figure 21.26). In proxy-based firewall, the security policy is implemented in the proxy firewall. Remote users log into the proxy server and send their HTTP request that contains the URL. The proxy server establishes TCP connection to the local server if the requested page is allowed. When it receives response from the local server, the proxy forwards it to the client. If the requested page is not allowed, the proxy does not establish connection to the local server. It responds on its own to the client giving indication of an error.

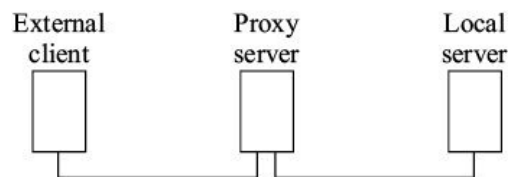


Figure 21.26 Proxy-based firewall.

The proxy server as described above is classical implementation of the

firewall. Here the client knows that it is to log into the proxy server. A transparent proxy server, on the other hand, is not visible to client. The client addresses the message to the local server. The proxy server intercepts it and processes it as before.

Firewall, be it filter based or proxy based, has its limitations. For example:

- It does not provide security against the internal attacks, *i.e.* a client within a network may breach the security.
- With advent of wireless communications, any user can have at least physical access to the network. If access is allowed to a genuine mobile user, there is nothing to prevent an unauthorized user to gain access to the network.

SUMMARY

The basic security requirements are confidentiality, integrity, authentication, and non-repudiation. Confidentiality is achieved by encryption of the message. Integrity is achieved by attaching an unalterable checksum to the message. Authentication of the source is done using shared secret keys, digital signature or through a trusted third party. Non-repudiation means that a sender must not be able to deny sending a message that it actually sent. It is achieved by authentication and integrity mechanisms.

Important encryption algorithms for confidentiality include Data Encryption Standard (DES), Advanced Encryption Standard (AES), and Triple DES. The source and receiver share a common encryption/decryption secret key. The secret key is exchanged using Diffie-Hellman key exchange mechanism or using digital signature based on public key encryption algorithm like RSA. Management and distribution of public keys is done through the recognized Certification Authority.

Integrity checksum is generated in such a manner that given a checksum, it is computationally impossible to find an alternate message having the same checksum. Examples of checksum algorithms are Message Digest 5 (MD5), Secure Hash Algorithm (SHA) and DES-CBC. Out of these MD5 is the most important algorithm.

There are several ingenious ways of authentication based on secret key, public key, and using services of a trusted third party. But one need to be careful as an intruder can find equal ingenious ways to defeat the authentication mechanism.

Security mechanisms can be built at the transport and network layers. Secure

Socket Layer (SSL) sandwiched between the application and the TCP layer provides secure transport layer services to the applications. IPSec protocol builds security at the network layer. It allows users to select encryption algorithms, and avail security services such as access control, authentication, integrity check, protection against replay, *etc.* Firewall is another protection build around network that checks ingress and egress of data packets across the firewall. Firewall is a router that filters packets bearing specific IP address or TCP/UDP port.

EXERCISES

1. Can the authentication mechanism given in Figure 21.9 be misused by an imposter? (*Hint*: Use two-session manipulation.)
2. Figure 21.3 shows DES-CBC encryption. Give the corresponding diagram for decryption.
3. Suppose a message is encrypted using DES-CBC (Figure 21.3). In Cipher-2, one bit gets inverted during transmission. How many plaintext blocks will get garbled? Which are these garbled blocks?
4. In Exercise 3, if in Cipher-2, one extra bit gets added during transmission, which plaintext blocks will get garbled?
5. Suppose you are using RSA encryption with $p = 3$, $q = 5$, and $e = 3$.
 - (a) Find decryption key d
 - (b) Encrypt 8
 - (c) Encrypt the word CAB if $C = 3$, $A = 1$, and $B = 2$.
6. Suppose you are using RSA encryption with $p = 13$, $q = 7$, and $e = 5$.
 - (a) Find decryption key d
 - (b) Encrypt 85
 - (c) Decrypt cipher $c = 2$.
7. Prove that RSA decryption recovers the original message.
8. In Diffie-Hellman algorithm, assume that there is an intruder C who can intercept every message that A and B exchange and substitute them with his own messages. Describe one scheme by which C can decipher all the message without A and B being aware of it. (*Hint*: C acts as middleman posing as B to A and as A to B.)
9. Find the secret key if the parameters for Diffie-Hellman algorithm are:
 $p = 719$, $S_B = 543$, $x = 16$.

1 Compression is not really a security function. Compression is carried out in the presentation layer if present. In TCP/IP suite this function is built in the SSL as there is no separate presentation layer.

22

Application Layer

The data network technology that we learnt in the rest of the book serves the ultimate purpose of supporting applications that make use of the network for transport of information. Each distributed application has a part of it that resides in the application layer of the layered network architecture. For example, Internet Explorer is a familiar application to most of us. It uses HTTP protocol of the application layer for exchanging messages. Like HTTP there are several application layer protocols. This chapter deals with the application layer protocols based on TCP/IP suite. Similar application protocols are available for the OSI architecture as well.

The application protocols we discuss in this chapter include Simple Mail Transfer Protocol (SMTP), Simple Network Management Protocol (SNMP), telnet—the protocol for remote login, File Transfer Protocol (FTP), Bootstrapping Protocol (BOOTP), Dynamic Host Configuration Protocol (DHCP), Hypertext Transfer Protocol (HTTP), DNS protocol and Trivial File Transfer Protocol (TFTP).

World Wide Web (WWW) is the most successful application of the public Internet. It uses the application protocols mentioned above with some additional components. A brief introduction to WWW and its components (web browsers, URL) is included in the chapter to complete the discussion.

22.1 TCP/IP APPLICATION PROTOCOLS

Unlike the OSI reference model, TCP/IP applications run directly on the TCP or UDP layer. The functions of the session and the presentation layers of OSI reference model are integrated in the applications. A selection of the important network applications of the TCP/IP suite consists of (Figure 22.1):

- Bootstrapping Protocol (BOOTP)
- Dynamic Host Configuration Protocol (DHCP)

- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)
- Telnet
- Hyper Text Transfer Protocol (HTTP)
- Trivial File Transfer Protocol (TFTP)
- Simple Network Management Protocol (SNMP)
- Domain Name System (DNS).

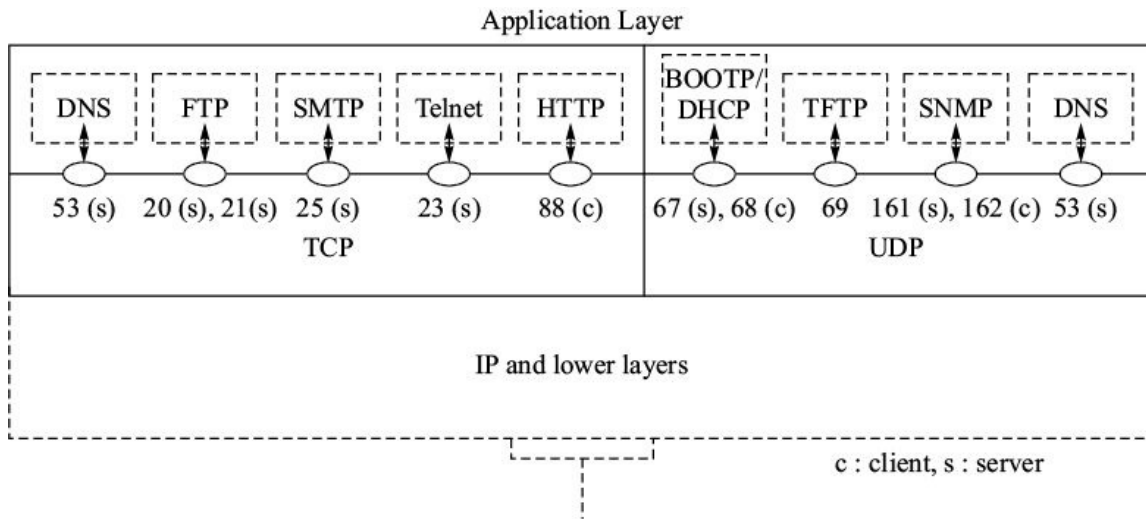


Figure 22.1 Application layer of TCP/IP suite.

These applications usually form part of a larger application or are used by a larger application. For example, e-mail is a very popular distributed application and Simple Mail Transfer Protocol (SMTP) is that part of the e-mail application that resides in the application layer.

These applications use either the TCP port or the UDP port for transfer of information and have been assigned specific port numbers. Note that some of the applications use two ports. The interaction between two communicating application entities (e.g. between two SMTP entities in two end systems) is based on client-server model which we discuss next. The server ports are always specified. The client ports, if not specified, can be any port having port number > 1023.

22.1.1 Client-Server Model

Distributed applications in the end systems are the entities that use the transport service provided by TCP/IP suite for communication. These entities have the following client-server relationship:

- A client process is a program that runs on an end system and generates request for the service from the server. The program is started by a user or by another application and is terminated when the service is complete.
- A server process is a program that runs on an end system and provides the requested service to the client program running on the remote end systems. When the server program is started, it starts accepting requests for service from the clients. When a request arrives, it responds to the request. The server program continues to run infinitely unless a problem arises that ends the program.
- Client-server relationship is many-to-one, *i.e.* several clients are served simultaneously by a server.

22.2 DOMAIN NAME SYSTEM (DNS)

We have been using IP addresses to identify hosts. While these addresses are perfectly suited for processing by routers, these addresses are really not very user friendly. Users are more comfortable with names than with digits. Thus we need a system that maps addresses to names and vice versa. The naming system used by TCP/IP is called Domain Name System (DNS). This system was developed in 1984 by P. Mokapetris of IAB. The RFCs associated with DNS are 1032, 1033, 1034, and 1035.

22.2.1 Hierarchical Naming System

IP address space consists of $2^{32} = 4294967296$ addresses theoretically. This is rather a large number for a naming directory. Assigning mnemonic names for such a large namespace requires a naming system that is easy to use, scalable, and manageable. The Internet used flat namespace¹ initially. Flat namespace is not scalable due to administrative reasons. Therefore, a hierarchical naming system is used.

A hierarchical naming system has vertical and horizontal partitions similar to that of an organization. Consider for example an educational institution UNIV (Figure 22.2). It has computer science (CS), telecommunication (TEL), and administration (ADM) departments. The computer science department has two divisions—software engineering (SW) and hardware engineering (HW). Each engineering division has a laboratory (LAB). If we use a naming scheme that reflects the lineage, a possible name for the software laboratory can be LAB.SW.CS.UNIV. The name LAB.SW.CS.UNIV is readily understood and

unique. The naming scheme is scalable. If computer science department introduces another division, network engineering (NE), the

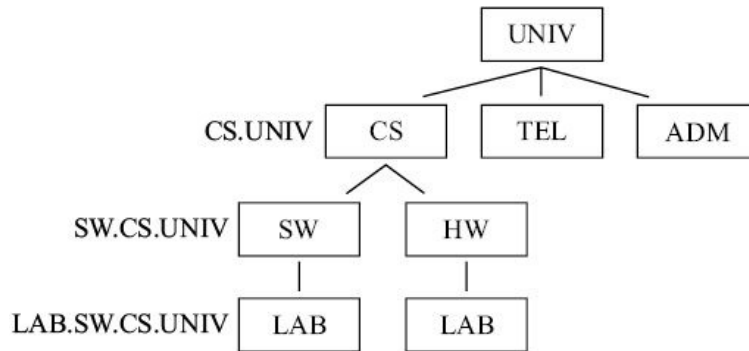


Figure 22.2 Hierarchical naming.

naming scheme readily accommodates the change without having any impact on the rest of the existing name structure.

22.2.2 Internet Naming System

The Internet uses a hierarchical naming system called Domain Name System (DNS). Its tree like structure is similar to the example given above. It maps the IP address of the hosts to a human readable format. It is implemented as distributed database worldwide on DNS ‘name servers’. When an IP host wants to get the IP address of a name, its client process communicates with the name servers to get the IP address. Before going into the structure of DNS, we need to get familiar with its terminology and conventions.

Domain. It is a complete sub-tree under a particular point in the naming tree structure (Figure 22.3). Note the emphasis on naming tree structure. A domain has nothing to do with geographic distribution of hosts.

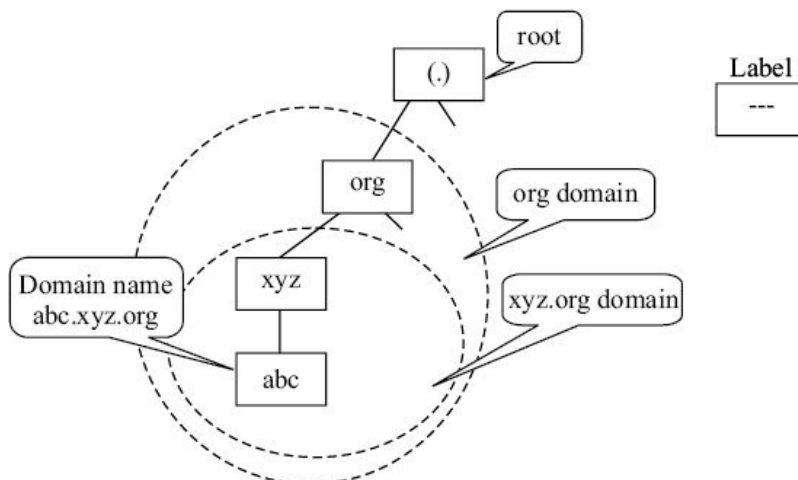


Figure 22.3 DNS terminology.

Label. It is a component of a domain name and identifies a local entity at a particular hierarchy level. For example, SW is the label assigned to the software engineering division of the computer science department. Note that SW in itself is not a fully qualified name.

A label need not be unique across the tree. It needs only to be unique at the particular point in the tree. In the last example, LAB was used as a label both for software and hardware laboratories without any conflict in the names.

Domain name. It is the name given to a node in the naming tree. It consists of all the labels from the root to the node, listed from right to left and separated by dots (.). It has the following characteristics:

- A domain name may consist of maximum 255 characters.
- Domain names are not case sensitive. For example, www.nic.org is same as WWW. NIC.ORG.
- Due to SMTP restrictions, domain names can contain only the characters a to z (lower or upper case), numerals 1 to 9 and symbol (-).
- A Fully Qualified Domain Name (FQDN) is concatenation of all the labels up to the root of the naming tree.
- Hosts with multiple IP addresses can be assigned a single domain name.
- Hosts with single IP address can have multiple domain names depending on applications, *e.g.* the domain names can be www. abc, mail. abc, *etc.* to differentiate several services.

The domain name database consists of master files. A master file is associated with a domain and contains name-address mappings of lower domains and the hosts in the domain.

Figure 22.4 shows naming tree structure of DNS. The root of the DNS tree is represented as a ‘.’. The root is implemented by several name servers at the highest level. The lower hierarchy levels are called top level, second level, and so on. The top level contains seven three-character generic domains as well as two-character country-specific domains:

1. edu (Education)
2. com (Commercial)
3. gov (US government)

4. mil (US military)
5. org (Organizations other than above)
6. int (International organizations)
7. net (Network providers)
8. Country specific, *e.g.* fr — France, in — India.

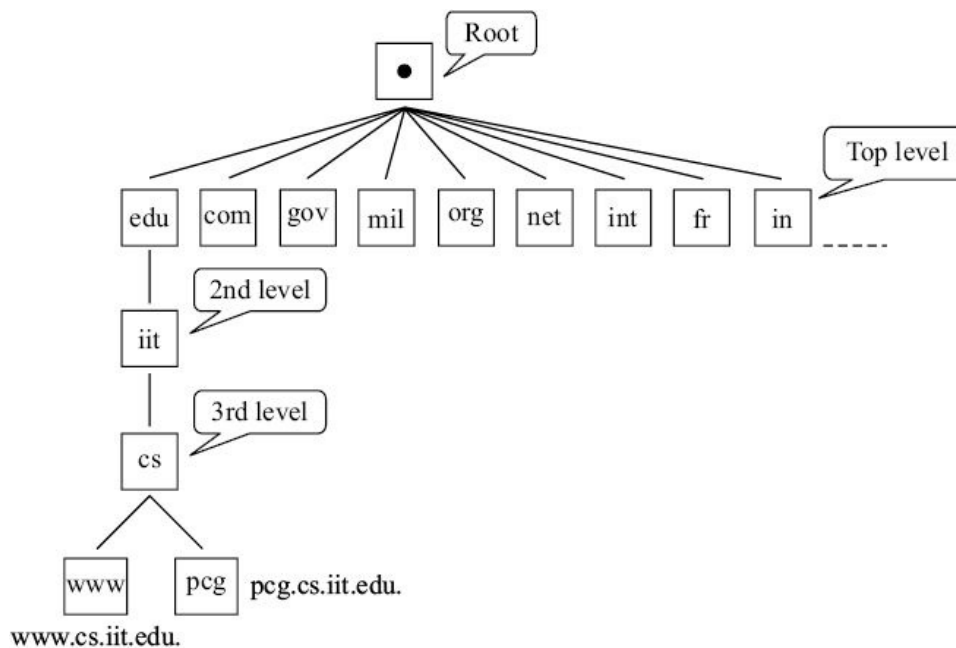


Figure 22.4 DNS naming system in the Internet.

Domain name registration is totally independent of IP address allotment. Domain names are registered with InterNIC (USA), RIPE (Europe), and APNIC (Asia).

22.2.3 DNS Protocol

DNS is based on the client-server model. DNS messages are of two types—queries (and replies) and zone transfers. Queries use UDP transport protocol and zone transfers use TCP transport protocol. DNS queries are limited to 512 bytes. The well-known port for the DNS server is 53. Basic structure of DNS message is depicted in Figure 22.5. It consists of several sections. Header is always present whilst other sections may be empty.

Header
Question
Answer
Authority
Additional

Figure 22.5 Structure of DNS message.

Header. It contains identifiers that associate queries with responses. It indicates the type of message (query or response), and the number of queries and records present in the message. There are several other fields. We will not go into the details.

Question. It contains the queries for the DNS server.

Answer. It contains the Resource Records (RR) that answer the queries.

Authority. It contains the list of authorities (name servers) that serve a particular domain.

Additional. This section contains additional information that does not form part of the sections above.

Figure 22.6 shows a typical exchange of DNS messages to find IP address of the name `www.abc.xyz.org`.

1. The client in the local host has pre-configured address of the default name server (NS). It sends the query to the default server.
2. The default server sends query to the root server which gives list of org name servers.
3. Default NS sends query to a org server which gives the list of xyz.org name servers.
4. Finally, xyz.org server gives the required IP address to the default name server which sends it to the local host.

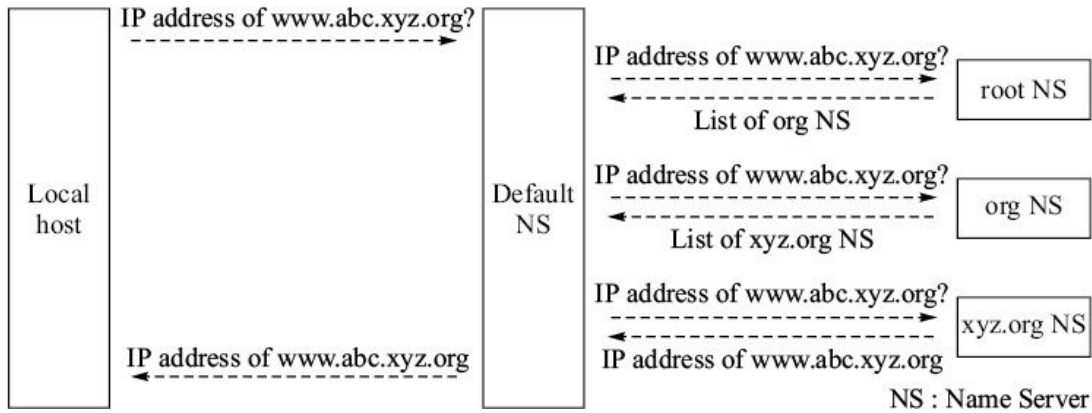


Figure 22.6 Typical exchange of DNS messages.

22.3 BOOTSTRAPPING PROTOCOL (BOOTP)

Bootstrapping allows diskless clients to load the operating system code and configuration parameters from a central server. The code so loaded cannot contain the IP address because the same code is loaded on all the diskless machines and therefore all the clients will get the same IP address. IP address can be obtained using RARP (Reverse Address Resolution Protocol). But RARP has the following limitations:

1. RARP uses layer two frame to obtain the IP address from the server. The server has to be on the local network as the frame cannot be forwarded through a router to another network. BOOTP, on the other hand, sends a UDP message contained in an IP packet. Thus a common server can support hosts on different local networks.
2. RARP provides only the IP address. BOOTP provides the boot-file name, address of the host that has the boot-file and other information in addition to the IP address.
3. RARP cannot be used on the networks that dynamically assign IP addresses because it uses the hardware address (MAC address) to IP address mapping table for assignment of IP addresses. But a machine can move from one local network to another in which case the mapping table in the local server will not have the hardware address. DHCP is another protocol that addresses this issue. We discuss DHCP in the next section.

Bootstrapping Protocol (BOOTP) was developed to replace RARP. It obtains the following information:

1. IP address for the diskless client
2. IP address of the host which will provide the boot-files (OS and configuration files)
3. Names of the boot-files
4. IP address of the router if the host that supplies boot-files is on another network.

BOOTP is based on the client server-model and is described in RFCs 951, 1542 and 2132. It uses well-known UDP port 68 for the client side (diskless machine) and well-known UDP port 67 for the server. Figure 22.7 shows basic interaction between BOOTP client and the server.

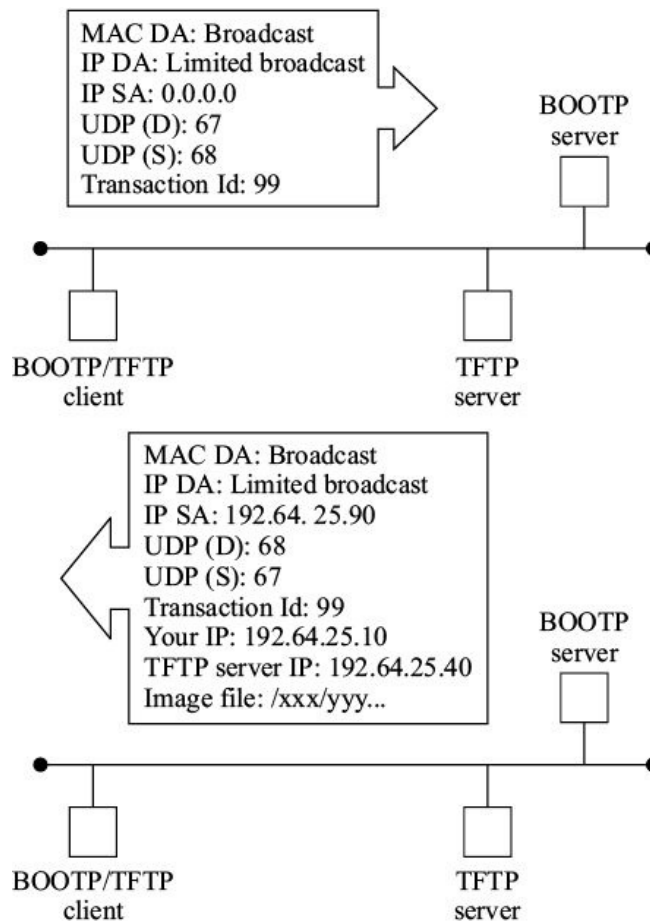


Figure 22.7 Exchange of messages between BOOTP client and server.

- The BOOTP client sends request to the server using IP address 255.255.255.255 (limited broadcast) as destination IP address and 0.0.0.0 as the source address.

- The BOOTP server replies with the desired information using the limited broadcast IP address as destination address.

This ends the BOOTP procedure. The client then uses TFTP (Trivial File Transfer Protocol) to obtain the boot-files from the host as indicated by the BOOTP server. BOOTP message format is shown in Figure 22.8. The client fills as many fields as it can and sets the rest to 0 in its request message. The server fills rest of the fields in its response.

OP. Operation code. Request = 1, reply = 2.

HTYPE. Network (hardware) type. For ethernet, HTYPE = 1.

HLEN. Hardware address length. For ethernet, HLEN = 6.

Hops. The client sets this field to 0. Every BOOTP server that forwards this request to another server increments this field by 1. Thus this field contains number of hops to the ultimate server.

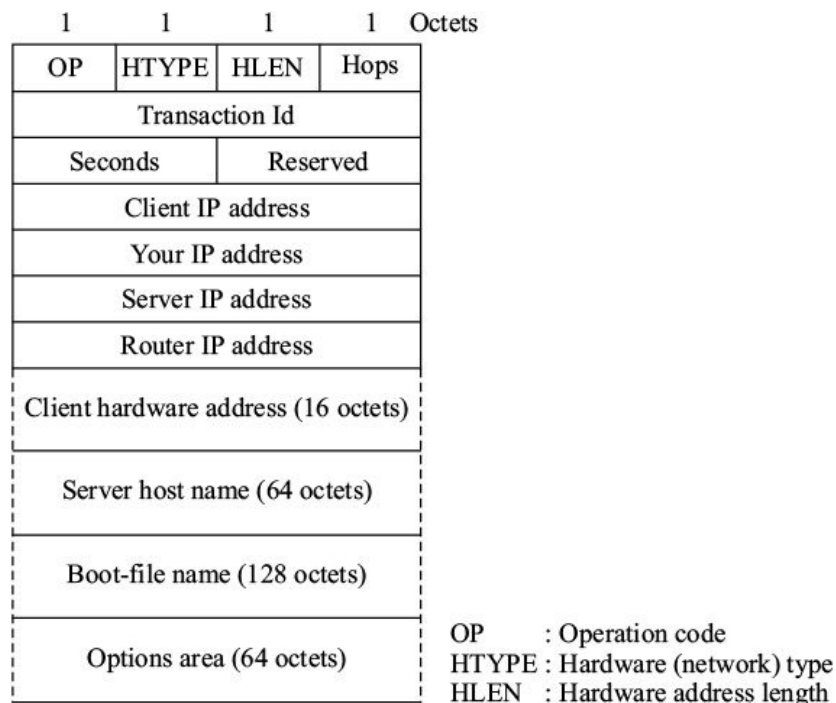


Figure 22.8 Format of BOOTP message.

Transaction Id. This field is used for associating the replies to the requests.

Seconds. This field indicates the seconds elapsed since the client started booting.

Client IP address. It is filled by the client. It contains 0.0.0.0 or the client's IP

address if known to it.

Your IP address. It is filled by the server if the client's address field is 0.0.0.0.

Server IP address. It contains the IP address of the host where boot-file is stored.

Router IP address. It is possible that the BOOTP server is in another network. This field contains the IP address of the BOOTP relay agent on this network that forwards the request.

Client's hardware address. It contains client's MAC address.

Server host name. It contains name of the host where the boot-file is stored.

Boot-file name. It contains directory path and file-name of the boot-file.

Options area. There are several options (RFC 2132). Optional area contains information based on selected option identified by an option code.

22.4 DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP)

BOOTP is suitable for static environment where the client's machine does not move from one network to another. This is so because the network manager creates in the server BOOTP file that contains the configuration parameters for each host. As and when a request comes, the parameters are sent to the requesting client based on the BOOTP file.

With the advent of the portable clients (laptops, wireless networking), the clients can move from one network to another and therefore the configuration file needs dynamic updating. A new protocol called Dynamic Host Configuration Protocol (DHCP) was designed to handle such situation. A DHCP server has an address pool and when a request comes, it allocates an address from the pool. Thus, DHCP extends the functionality of BOOTP to include mechanisms to allocate host (client's) addresses dynamically.

DHCP is based on client-server mode of operation and uses the same UDP ports as BOOTP.

22.4.1 Configurable Parameters

DHCP has three modes of assignment of IP addresses:

1. **Automatic.** DHCP assigns the address on permanent basis.

2. Dynamic. The address is assigned (leased) for a limited period.

3. Manual. The address assignment is manually configured as in BOOTP.

In addition to assignment of IP address, DHCP allows number of parameters to be configured. A client can ask for subnet mask, DNS server, default TTL value, MTU size, source routing option, maximum fragment size, ARP cache timeout, *etc.*

22.4.2 Message Format

The message format of DHCP is same as that of BOOTP (Figure 22.8) with the following changes:

- The options area is at least 312 octets instead of 64 octets. All the DHCP messages (described later) are appended here.
- The reserved field is called flag field in DHCP. The left-most bit of the field is set to 1 by a client if it wants the response from the server in broadcast mode. Other bits of the field are set to zero.

22.4.3 Message Types

The following types of messages are sent by the client and by the server. These messages are contained in the options area and are identified by option code 53 and type field which contains the type value.

DHCPDISCOVER (Type 1). Broadcast from client to find DHCP server(s).

DHCPOFFER (Type 2). Response from the server to DHCPDISCOVER offering IP and other parameters.

DHCPREQUEST (Type 3). Message from the client to indicate selection of the parameters offered by a server. It is also indication to other servers declining their offers. The client also requests extension of lease time using this message.

DHCPDECLINE (Type 4). Message from the client to the server indicating an error.

DHCPACK (Type 5). Acknowledgement from the server to the client.

DHCPNACK (Type 6). Negative acknowledgement from the server to client if the IP address requested by the client is invalid or the lease is expired.

DHCPRELEASE (Type 7). Message from the client to the server canceling

remainder of the lease and releasing the IP address.

DHCPINFORM (Type 8). Message from the client that it already has a configured IP address but needs additional configuration parameters.

Figure 22.9 shows a typical exchange of DHCP messages between a client and a server for IP address lease. The client sends IP lease request using DHCPDISCOVER message. The server sends its offer using DHCPOFFER message. The client indicates its selection of lease to the server by sending DHCPREQUEST message. The server accepts the request by sending DHCPACK message.

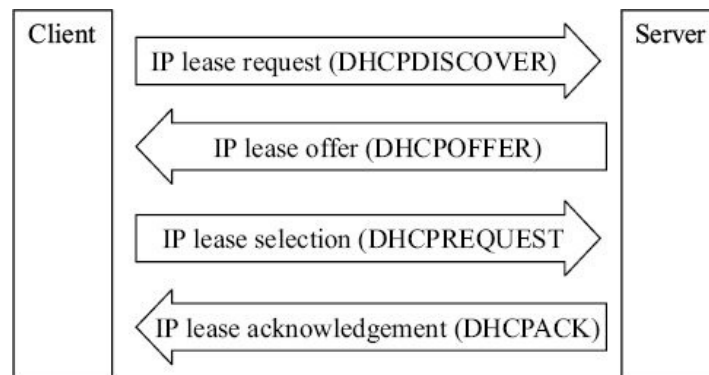


Figure 22.9 Typical exchange of DHCP messages.

22.5 TRIVIAL FILE TRANSFER PROTOCOL (TFTP)

Trivial File Transfer Protocol (TFTP) is suited for applications that do not require rather complex procedures of FTP (File Transfer Protocol) and do not have enough resources (RAM, ROM) for this purpose. For example, FTP requires multiple concurrent session which may not be possible on small diskless machines or network devices such as bridges, routers, *etc.* The code size of TFTP is very small and can be easily fit into bootstrap-ROMs of such machines and devices. Typical applications of TFTP include loading the image on diskless machine and upgrading the operating system in network devices such as routers. The main features of TFTP are:

- TFTP is based on client-server principle and uses well-known UDP port number 69 for the TFTP server.
- TFTP is unsecured protocol and does not support authentication.
- TFTP incorporates idle-RQ (stop and wait) error recovery mechanism.
 - Every TFTP data unit bears a sequence number.

- Each data unit is individually acknowledged. After receiving the acknowledgement the next data unit is sent.
- Error recovery is by retransmission after timeout. TFTP uses adaptive timeout with exponential back-off algorithm.

22.5.1 TFTP Message Formats

There are four types of TFTP messages:

Read request (Type 1). This command is used by the client to get a copy of a file from the server.

Write request (Type 2). This command is used by the client to write a file into the server.

Data (Type 3). This TFTP message contains blocks of data (portions of the file being copied).

Acknowledgement (Type 4). This is used by the client and the server to acknowledge the received data units.

Figure 22.10 shows the format of these messages. The first two octets indicate the type of message. Mode field defines the type of data (ASCII, binary, mail). The file-name and mode

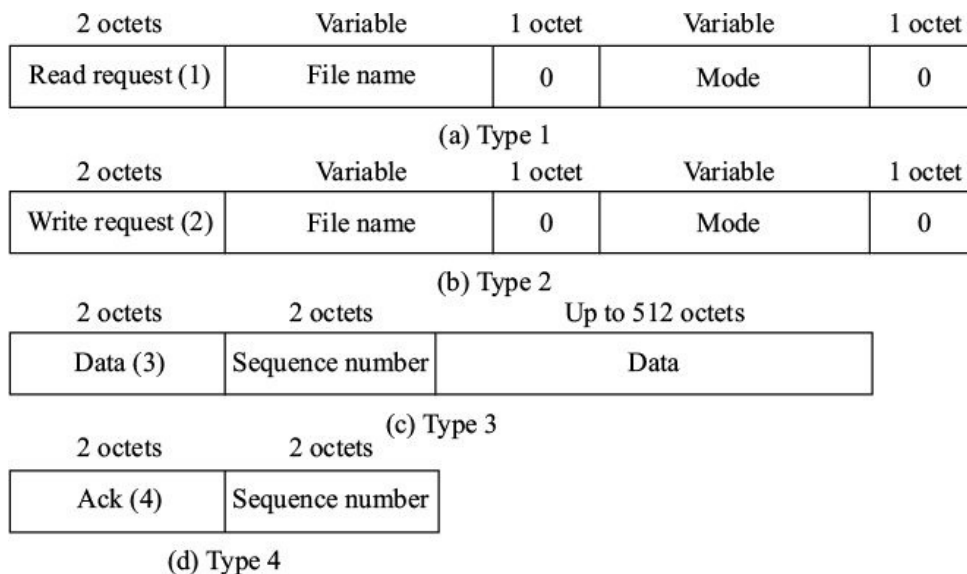


Figure 22.10 Types of TFTP messages.

fields are delimited using an all zeroes octet. Type 3 message contains the data blocks of fixed size of 512 octets. The session is terminated if a data message

arrives with data octets less than 512 octets. The last data message can have data block with EOF having size less than 512 octets. Type 4 message is used for acknowledgement.

22.5.2 TFTP Operation

TFTP operation is very simple. The client sends a read or write request at the server's UDP port 69. The server accepts the request by sending data messages in case of read request, and by sending acknowledgement in case of write request. In either case, the server selects a UDP port to be used for further dialogue and sends its first response to the client through the selected UDP port.

Each data message has fixed size of data block (512 octets) and is individually acknowledged. As mentioned earlier, error recovery is done using retransmission after timeout. The last data block containing EOF (End of File) or a data block containing less than 512 octets terminates the session. Figure 22.11 illustrates the mechanism after a diskless client has received the IP address of the TFTP server and the boot-file name from the BOOTP server.

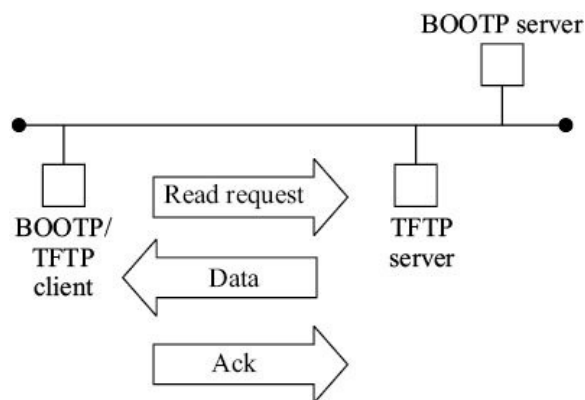


Figure 22.11 TFTP operation.

If a TFTP message is lost and if there is no expected response, the message is repeated by the sender after timeout. Thus, if an acknowledgement is lost, the data message is repeated after timeout. If the next data message is not received after acknowledgement, the last acknowledgement is repeated after timeout.

22.6 TELNET

The main task of the Internet is to let users have access to remote hosts running different applications. One possible way is to write client-server program for

each application. But this is not a practical solution. Alternative is to have a general-purpose client-server program that lets a user access any application running on a remote host. Telnet is a general purpose client-server application that achieves this objective (Figure 22.12). It is one of the most popular Internet applications because of its flexibility, minimal use of resources, and ready availability. It is specified in RFC 854.

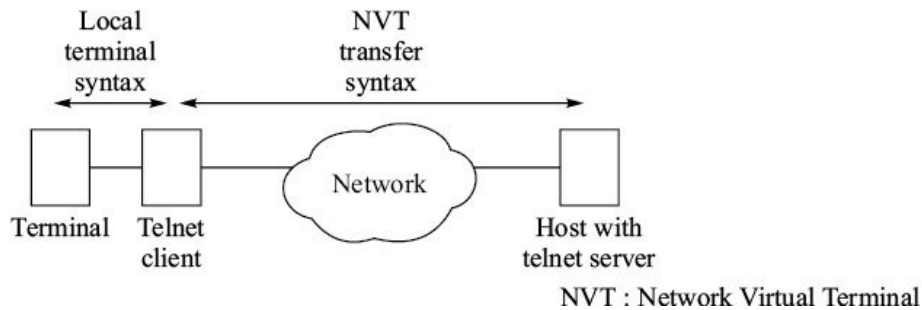


Figure 22.12 Telnet client and server.

Telnet is a connection-oriented protocol and uses TCP at the transport layer. The telnet server is connected to the well-known TCP port 23. The telnet client can be on any TCP port > 1023 (Figure 22.13).

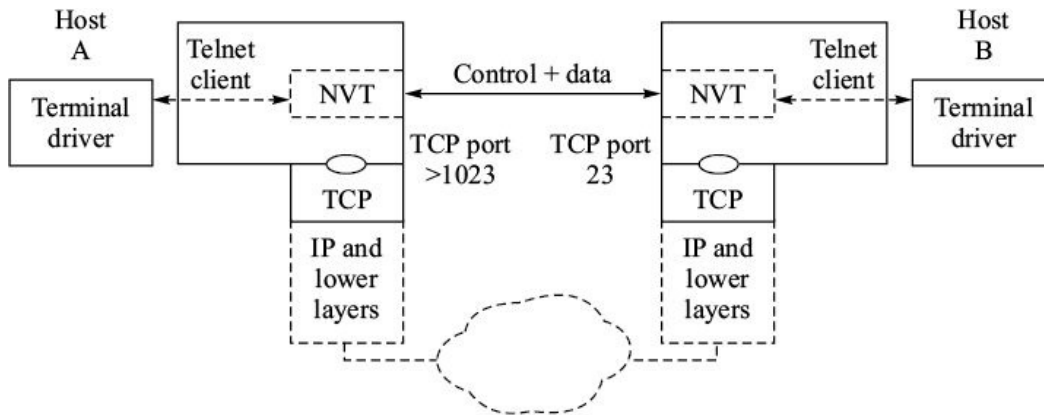


Figure 22.13 Telnet architecture.

A telnet client can emulate the behaviour of a wide range of well-known terminals. Internally the telnet client communicates with the telnet server through a canonical terminal representation called Network Virtual Terminal (NVT) as shown in Figure 22.13. Thus, the local terminal device characteristics are mapped to NVT capabilities. The user's keystrokes received by the local terminal driver are sent to the telnet client which transforms these to a universal character set of the NVT. The process is reversed at the other end.

22.6.1 Character Set of NVT

NVT uses 8-bit data format but the basic character set is same as ASCII 7-bit code. Telnet permits negotiating the options for emulating various terminal characteristics. These options are negotiated using commands based on code words taken from the 8-bit codes from 128 to 255 (up to 127 are used by ASCII). Codes from this set are also assigned for signaling purposes.

22.6.2 Telnet Commands

Telnet commands are identified by a prefix character, IAC (Interpret as Command) having code 255. IAC is followed by the command and option codes. Basic format of a command is shown in Figure 22.14.

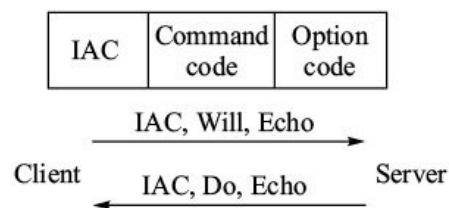


Figure 22.14 Format of telnet commands.

Some important commands are:

- Will Code 251 Sender wants to enable an option.
- Won't Code 252 Sender does not want to enable the option.
- Do Code 253 Sender asks receiver to enable an option.
- Don't Code 254 Sender asks the receiver not to enable the option.

Some important options are:

- Echo (Code 1) The received character is echoed back to the sender and displayed on the screen as he keys in.
- Status (Code 5) Verifies status of remote telnet options.
- Line width (Code 8) Specifies line width.
- Page size (Code 9) Specifies number of lines in a page.

22.6.3 Standard User Commands

The standard user commands for telnet are:

- open < remote IP address> Sets up a connection to the host.
- close Closes the connection to the host.

- quit, Ctrl-D Exits the current telnet session.
- set <telnet variable><value> Sets the telnet variable to the specified value.
- ? Help
- status Gives status of current session.
- type <terminal type> Enables a specific terminal emulation.
- mode It causes toggle between ASCII and binary mode.

22.7 FILE TRANSFER PROTOCOL (FTP)

The way files containing information are stored in a system depends on the architecture of the system which includes the hardware, the operating system, data type, coding styles, file organization, and the access methods. To exchange these files between two different systems requires a common protocol that is acceptable to the two systems. There are two approaches for exchange of the files:

- Virtual files
- Reduction approach.

In virtual file approach, (just like NVT), a virtual file structure is defined. The real files are translated to the virtual files by the one end system. The virtual file is transmitted to the other end where they are translated to the local file system. This approach is quite complex because all the possible representations and file structures must be considered for implementing translators from the real to the virtual file system. An example of virtual file system protocols is FTAM (File Transfer, Access, and Management) protocol of ISO.

The second approach is to reduce all the file systems to a file system having a common minimal set of fundamental properties. Only views and operations as defined by the common set of properties are possible. No translation is carried out as in case of virtual files. File Transfer Protocol (FTP) is based on this concept. FTP enables transfer of a copy of a file from one end system to another. The original remains unchanged and in the original location.

22.7.1 Basic Features of FTP

The basic features of FTP are:

Data representation. FTP handles three types of data representations—ASCII (7 bit), EBCDIC (8-bit), and 8-bit binary data.

File organization. FTP supports unstructured and structured files. An unstructured file contains string of bytes, end-marked by EOF (End of File). A structured file contains a list of records and each record is delimited by EOR (End of Record). EOF and EOR are represented as a sequence of two bytes, 0xFF + 0x01 (EOR), 0xFF + 0x02 (EOF) and 0xFF + 0x03 (EOR + EOF).

Transmission mode. In stream mode, the file is transmitted as continuous bit stream without any modification. EOF and EOR, where applicable, are inserted as the two byte sequence. In block mode, the data is divided into blocks with EOR after every block and EOF at the end of the file. In compressed mode, a sequence of same characters is transmitted only once with a replication counter that enables the receiver to restore the compressed data.

Error control. Since TCP is used for data transfer no additional error recovery mechanism is required.

Access control. File access protection is done using login procedure with login name and password.

22.7.2 FTP Operation

FTP uses client-server model for communications. Two TCP connections are maintained for file transfer (Figure 22.15). On one connection, control signals (commands and responses) are exchanged. This connection is used by the control process, called Protocol Interpreter (PI). The client PI and the server PI communicate using NVT syntax. PIs are responsible for translating local syntax (e.g. DOS or Unix) into NVT syntax and vice versa. The TCP connection for control signals uses well-known FTP server port 21.

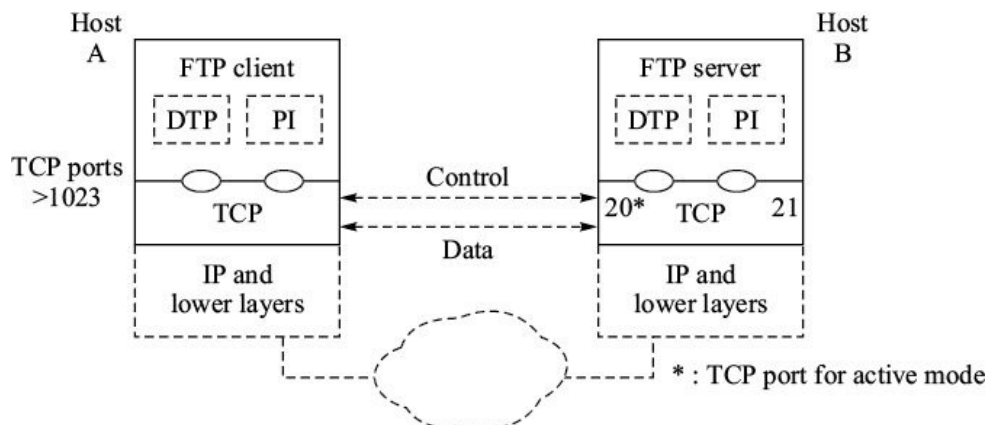


Figure 22.15 FTP architecture.

The other connection is used for file transfer. For this purpose, FTP has Data Transfer Process (DTP). This connection is used by the client DTP and the server DTP. This connection can be established in two ways:

- The default mode is to send request via client PI process to the server PI process for establishing DTP connection. The server PI process sets up the DTP connection from its well-known port 20 (active mode). Port 20 is a well-known port and therefore client may face firewall restrictions.
- Alternatively, the client PI requests for a passive TCP port number of the server. The server indicates the passive port number (>1023) to the client PI, which establishes DTP connection to the indicated TCP port of the server DTP. The passive port indicated by the server does not have firewall restrictions.

The DTP connection is terminated when the data transfer is complete.

22.8 ELECTRONIC MAIL

Electronic mail (e-mail) is one of the oldest network applications and has been the most popular service as well. TCP/IP application protocol that supports e-mail service is Simple Mail Transfer Protocol (SMTP) described in RFC 821. SMTP deals with transfer of messages. The format of the messages is defined in RFC 822. Originally, SMTP was limited to transfer of text messages. But as the usage of e-mail grew, there was demand for transfer of images, voice/video clips, and data. An extension protocol to RFC 822, Multipurpose Internet Mail Extension (MIME) was standardized to meet this requirement (RFCs 2045 to 2049).

22.8.1 Basic Components

Figure 22.16 illustrates the mechanism of flow of e-mails. The basic components that constitute an e-mail system are:

- Mail User Agent (MUA)
- Spool-file
- Mail Transfer Agent (MTA)

- Mail box.

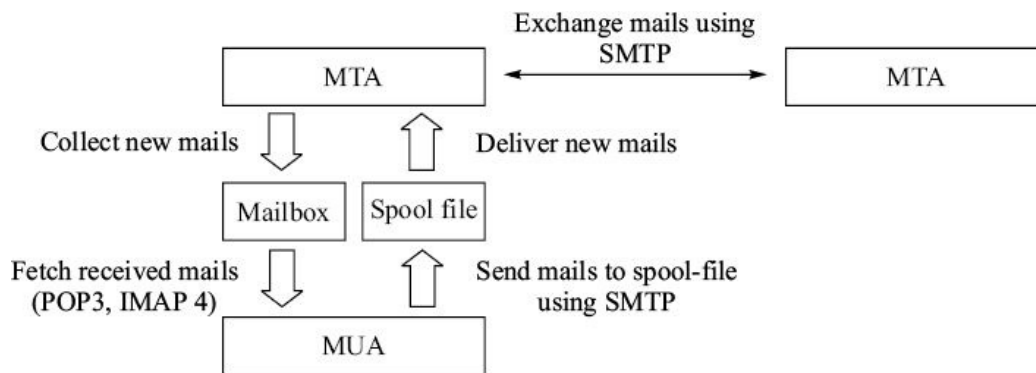


Figure 22.16 Overall schematic of e-mail service.

These components interact using SMTP, POP3, IMAP protocols. The format of content of the messages is as defined in RFC 822 with extension MIME.

Message user agent (MUA). This is a process that reads and writes e-mail. It fetches mail from the mailbox using protocols POP3 or IMAP4. It appends the outgoing mails to the spool-file.

Spool-file. It contains the mails to be sent. MUA appends the outgoing mails in the spool-file using SMTP protocol. MTA extracts the pending mails from the spool-file for their delivery.

Message transfer agent (MTA). It is the process that transfers the mails from the spool-file. It forwards the mails to the mailboxes of the recipients if they are connected in the same machine. It delivers the mails to peer MTA if the destination mailbox is in another machine. The delivery from one MTA to another is done using SMTP. There can be intermediate MTAs also that act as transit for transferring mails from one MTA to another.

Mail box. It is the designated file owned by the receiver. The delivered mails are appended in this file. The user can read and delete the mails from his mailbox file.

22.8.2 Mail Addresses

Every e-mail is identified by a unique e-mail address. It consists of a character string conforming to the format—user@domain. The user part identifies the mailbox of the user in the domain indicated in the domain part of the address. The user part is unique within the domain. Domain part identifies some organization, or a host machine that provides mailbox service and it is unique

globally.

The host machine identified by the domain part is also called Mail Exchange (MX). The domain name assigned to a mail exchange comes from the DNS database. DNS servers enable identification of the mail exchange that serves a domain. DNS server also indicates the IP address of the mail exchange.

It is possible to have several mail exchanges for one domain. If several mail exchanges exist for a domain name, each mail exchange is assigned a preference value. The mail exchange with the lowest preference value is chosen by the MTA to avoid mail loops.

22.8.3 Mail Format

Mail format is defined by RFC 822 with extension MIME. It consists of two parts—an envelope or header and a body. Both parts are represented in ASCII. The body is assumed to be simple text, although it may be encoded image or voice/video clip or data.

Header. The header contains information necessary for transmission and delivery of the mail. It also includes the sender's identification. It consists of a series of lines each terminated with a pair of ASCII characters, CR (carriage return) and LF (line feed). Each line has a type field followed by a colon and the value. The header starts with FROM in the first line. For example, FROM: prakash <prakash@xyz.com> identifies the sender. The header contains information about the sender, receiver, date-time, subject, *etc.*

Body. The body of the mail is separated from the header by a blank line. It contains user's message. RFC 822 restricts the maximum size of the body to 1000 characters. But MIME extends it further in certain cases. The body always contains character codes from ASCII. The body is followed by a signature separated from the body by two dashes '--'. It contains personal information of the sender, keys, *etc.*

22.8.4 Simple Mail Transfer Protocol (SMTP)

SMTP is based on client-server model and it uses well-known TCP port number 25 for the server (Figure 22.17). All the commands, responses, and messages are in ASCII format (printable characters with binary values from 33 to 126, CR and LF).

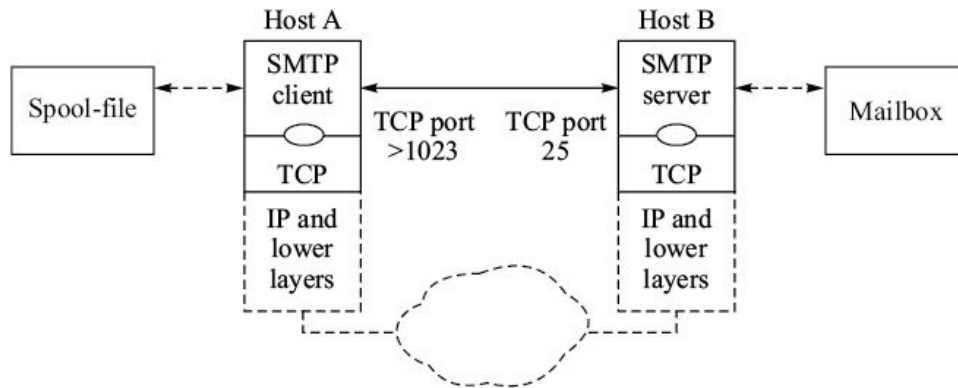


Figure 22.17 SMTP architecture.

The important commands from the client and responses from the server are listed in Tables 22.1 and 22.2 respectively. All the commands and responses are appended with CR LF sequence. A typical example of sending a mail is as follows:

TABLE 22.1 Commands from the Client	
Command	Description
HELO<domain>	It is used for sending identification.
MAIL FROM: <...>	It specifies sender's name.
RCPT TO: <...>	It specifies receiver's name.
DATA	It indicates beginning of mail transmission.
SEND FROM: <...>	It specifies that this mail should be sent directly to user's terminal.
SOML FROM: <...>	It specifies that the mail should be sent directly to the user terminal. If the user's terminal cannot be reached, mail is to be sent to the mailbox (SOML : Send or Mail).
RSET	It aborts current mail transaction. TCP connection remains open though.
QUIT	It closes current session and the TCP connection.

TABLE 22.2 Responses from the Server	
Response code	Description
220 <domain>	It indicates that the server is ready.
250 OK	It indicates that requested mail action has been completed.
354	It is an indication to client to start its mail input and to end it with CR, LF, ., CR, LF.
421 <domain>	It indicates that the service is not available.
451	It indicates that the requested action is aborted due to local error in processing.

500	It indicates that there is syntax error and therefore the command is not recognized.
550	It indicates that the requested action is not taken, <i>e.g.</i> mailbox is not accessible or found.
551 <forward path>	It indicates that the user is not local and to try the <forward path>.
552	It indicates that the requested action is aborted as storage allocation is exceeded.

EXAMPLE 22.1

Client : (Opens TCP connection to port 25 of the server)
 Server : 220 xyz.edu Simple Mail Transfer Service ready
 Client : HELO abc.edu
 Server : 250 OK
 Client : MAIL FROM: <prakash@abc.edu>
 Server : 250 OK
 Client : RCPT TO: <nini@xyz.edu>
 Server : 250 OK
 Client : DATA
 Server : 354 start mail input; end with <CR><LF> . <CR>< LF>
 Client : (sends the message in RFC 822 format)

<pre>Date: Fri 11 January 2005 09:10:23 From: Prakesh Gupta < prakash@abc.edu> Subject: Greetings To: nini@xyz.edu How are you doing? Prakash</pre>
--

Client : <CR><LF> . <CR><LF>
 Server : 250 OK
 Client : QUIT
 Server : 221 xyz.edu closing transmission channel.

22.8.5 Multipurpose Internet Mail Extension (MIME)

RFC 822 allows mail format that uses only ASCII characters in the message body. But it has several limitations:

1. SMTP cannot transfer executable files and binary objects.
2. SMTP cannot transmit text data of other languages, *e.g.* French, Japanese,

Chinese, etc., because these are represented in 8-bit codes.

3. SMTP servers may reject mails having size greater than a certain size.
4. SMTP cannot handle non-textual data (pictures, images, and video/audio content).

Multipurpose Internet Mail Extension (MIME) is a supplementary protocol that resolves these issues and allows non-ASCII data to be sent through SMTP. It is a mechanism for specifying and describing the format and content type in a standardized way. E-mail content can be text, images, audio/video, HTML pages, and application specific data when MIME is implemented. Note that MIME is an extension protocol and cannot replace SMTP.

MIME header. MIME is realized using five header fields.

MIME version. It indicates the MIME version being used.

Content-type. It describes the type and subtype of data in the body of the message. It describes how the object in the body is to be interpreted. The default value is plain text in US ASCII.

Content-transfer-encoding. It describes how the object within the body has been encoded to US ASCII to make it acceptable for mail transfer.

Content-Id. It is used to uniquely identify the MIME entities.

Content-description. It is a plaintext description of the object within the body. This is needed when the object is not displayable (e.g. audio content).

Content-description and content-id fields are optional.

Standard content-types. Seven major types and fourteen subtypes of content are defined in MIME as shown in Table 22.3.

Type	Subtype	Description
Text	Plain	Un formatted text in US ASCII or ISO 8859.
Image	jpeg	JPEG format.
	gif	GIF format.
Audio	Basic	Single channel 8-bit m-law encoding at 8 kbps.
Video	Mpeg	MPEG format.
Application	Postscript	Adobe Postscript.

	Octal-stream	Binary data in 8-bit octets.
Multipart	mixed	Mixed parts to be presented to the receiver sequentially.
	parallel	Same as mixed but the order is not defined.
	Alternative	Different parts are alternative versions of same content and are to be used based on capabilities of the receiver.
	Digest	Similar to mixed but default type/subtype is RFC 822.
Message	rfc822	RFC 822 format.
	partial	Large mail is fragmented.
	external-body	Contains pointer to an object that exists elsewhere and is accessible via FTP, TFTP, local file, and mail server.

Standard content-transfer-encodings. The content-transfer-encoding can be of six values as listed in Table 22.4. In Base64 encoding, three 8-bit octets are combined to form four 6-bit words. 6-bit words are padded with 00 prefix to make them into 8-bit words. 8-bit word, so

Encoding	Description
7-bit	The body contains 7-bit ASCII characters with maximal length of 1000 characters.
8-bit	There can be non-ASCII 8-bit characters but the maximum length of the body is limited to 1000 characters.
Binary	Binary 8-bit characters without limitation of 1000 characters in the body.
Quoted-printable	This is useful when the data consists of largely printable characters. Characters in the range decimal equivalent 33 to 61 in ASCII are represented in ASCII. Others are represented as two-digit hex representation preceded by '=' sign. Non-text characters are replaced with six-digit hex sequence.
Base64	It converts the binary data in a form that is invulnerable to the mail processing.
x-token	A named non-standard encoding.

formed, is converted into character encoding using Base64 encoding table. For example, three octet sequence 00100011-01011100-10010001 is expressed as four 6-bit words 001000-110101-110010-010001. When prefixed with two

zeroes 00001000-00110101-00110010-00010001, Base64 defines these words as IlyR. These characters are mail safe when transmitted as ASCII.

22.8.6 Post Office Protocol (POP3) and Internet Message Access Protocol (IMAP4)

Very often a user writes and reads his e-mails on local machine and has his mailbox on a server that runs SMTP for receiving e-mails (and probably SMTP client for sending e-mails). The server is permanently connected to the Internet. Post Office Protocol, version 3 (POP3) and IMAP are the protocols that let a user fetch his e-mail from its remote mailbox on the server. The user's machine and the mail server run the client and server processes respectively.

POP3 is described in RFC 1939 and it runs on well-known TCP port 110. The communication procedure is similar to SMTP and uses ASCII characters. IMAP is described in RFC 2060. It is more sophisticated than POP3 and allows a client to access and manipulate e-mails and mailboxes. It allows:

- a client to create, delete, and rename mailboxes,
- a client to selectively fetch message attributes such as all, body, envelope or flags,
- searching the mailbox for given match criteria,
- maintaining several flags such as seen, answered, draft, and deleted.

IMAP runs on the well-known TCP port 143.

22.9 SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

Large and complex networks cannot be managed by human effort alone. An automated system for managing such networks is required. Network management function includes several activities:

Provisioning. Provisioning refers to putting network configurations in place and integrating the network. Building a transmission link between two routers, configuring router ports, *etc.* are the provisioning activities.

Faults reporting. Whenever a fault develops or an alarm is generated, the event needs to be reported to a centralized network operations group for taking suitable remedial actions.

Performance reporting. There is need to monitor the network performance in terms of defined performance parameters. For example, network performance of an IP link can be defined in terms of packet loss and latency (round trip delay).

Others. There are other network management activities such as collecting accounting information, security information, and event logs.

Note that these are not one-time activities. These are all ongoing network operations activities. To carry out these activities using an automated tool, requires a comprehensive network management system.

Figure 22.18 shows a typical model of a network and its Network Management System (NMS). A network consists of managed Network Elements (NEs, *e.g.* routers, transmission equipment). These network elements are managed through defined parameters that determine their operation. Examples of the parameters are IP address of a router interface, packet loss at a router interface, alarm of power supply failure, *etc.* Network management system

- polls the NEs to get the values of these parameters,
- gives commands to set the values, or
- receives alarm messages on occurrence of an abnormal situation in the network.

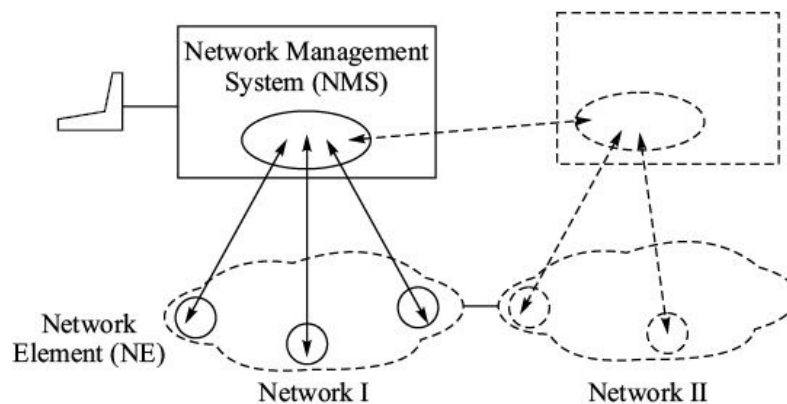


Figure 22.18 Network management system (NMS).

There may be a dedicated separate network for the communication between the NMS and the NEs. Alternatively, this communication can be ‘in-band’, *i.e.* the network being managed also carries the management signals. In the later option, a fault in the network may also disrupt the communication channels for management of the network.

The application layer protocol between the NMS and the NEs is the network management protocol that we will study in this section. Simple Network

Management Protocol (SNMP) was developed for the networks based on TCP/IP and its scope was later expanded to include other network types (OSI, Appletalk, IPX, etc.). The latest version of SNMP is version 3 which also includes security features (authentication, confidentiality, and access control). It is called ‘simple’ protocol because the agent requires minimal software processes and there is minimal set of messages.

22.9.1 Basic Elements

A Network Management System (NMS) consists of four basic elements (Figure 22.19):

- Management process (client)
- Agent process (server)
- Management Information Base (MIB)
- Communication protocol (e.g. SNMP).

These are in addition to the application that receives and processes the information for generating reports and for triggering actions for restoration of affected network performance.

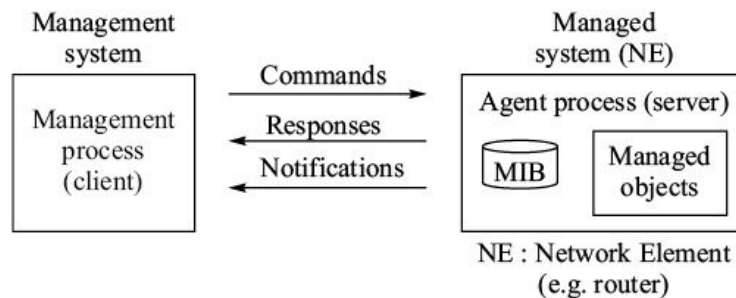


Figure 22.19 Basic elements of NMS.

Management process (Client). The network management system acts as client and periodically polls the agent processes in the network elements to collect the statistics. It also collects unsolicited notifications from the agent processes. The client side must provide for enough resources (CPU, memory) to manage a large number of agents.

Agent process (Server). The agent process (server) resides in the managed network elements and sends responses to the commands from the client. It can also send notification (called trap) in the event of an alarm. Typical Network Elements (NEs) where the agent process (server) resides include managed

servers, routers, bridges, testing equipment, power supplies, transport equipment, *etc.*

Management information base (MIB). Management information base is the collection of objects that the management process can manage. An object is a data variable that represents an aspect of NE's state (e.g. the time stamp when the router rebooted). The management process retrieves the value of an MIB variable, or set its value. The agent process notifies values of MIB variables on being asked or notifies on its own.

Simple network management protocol (SNMP). As mentioned before, SNMP is the communication protocol between management process and the agent process.

Besides describing the SNMP architecture, we now discuss in detail management information base (MIB).

22.9.2 SNMP Architecture

SNMP uses connectionless service provided by the UDP layer. The UDP port numbers are 161 for the agent and 162 for the client (Figure 22.20).

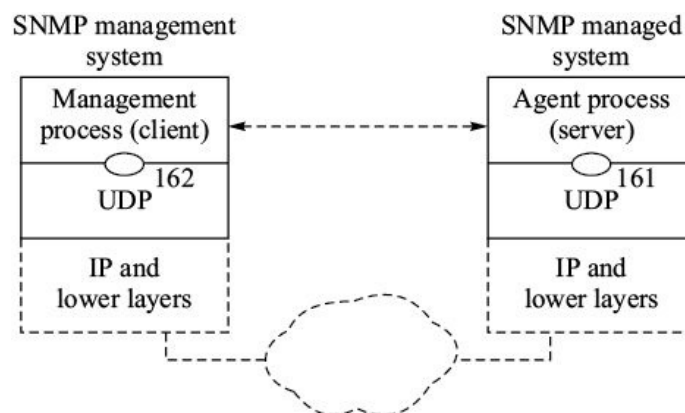


Figure 22.20 SNMP architecture.

22.9.3 Management Information Base (MIB)

The MIB defines the specific pieces of information, the MIB variables, that can be retrieved from the managed system. The current version of MIB is called MIB-II, which organizes the variables into ten groups:

System group. It includes general system parameters like description of equipment, operating system, location, *etc.*

Interface. It includes the information about all the interfaces such as physical

addresses, status (up/down), number of octets received, number of packets received, number of packets sent, number of packets rejected, number of damaged packets, *etc.*

Address translation. It gives information about mapping of the IP-addresses and the physical addresses, *i.e.* address translation table of ARP.

IP. This group contains information on forwarding tables, various IP packet counters, IP addresses, and masks.

TCP. It contains counters for TCP segments, TCP parameters (e.g. retransmission timer value), session information, socket information, *etc.*

UDP. It contains information on UDP counters and socket information.

EGB. It contains table of neighbouring routers and counter of EGP messages.

ICMP. It contains counter for various packet-types and events.

SNMP. This group contains information on SNMP itself.

Other groups. This group contains information of other media.

22.9.4 Structure of Management Information (SMI)

Structure of Management Information (SMI) specifies how the information about the managed objects is to be represented. The relevant RFCs are 1155, 1212, and 1215. Each MIB variable has a defined name, syntax, and encoding. Syntax defines the data type (integer, string of bytes, etc.). Encoding defines how the variable is transmitted. ASN.1 (Abstract Syntax Notation) and ASN.1 BER (Basic Encoding Rules) describe the rules for naming, encoding, and definition of syntax.

22.9.5 SNMP Operation

The SNMP client sends a request indicating ASN.1 identifier of the variable it wants the value for. The server maps the identifier to the local variable and retrieves the current value of the variable from its respective memory location. It uses ASN.1 BER to encode the value and sends the response.

Client's requests. There are five types of requests issued by the client:

- GetRequest
- SetRequest

- GetNextRequest
- InformRequest
- GetBulkRequest.

GetRequest is used for fetching value of the requested variable. SetRequest is used for setting value of a variable. For example, if a router is configured to reboot when the value of a variable is zero, a client can cause a router to reboot by setting the value of the variable to zero.

There are MIB variables that are either tables or data structures. For such variables, GetNextRequest is used. When GetNextRequest with a variable Id is sent, this operation returns the value of the variable plus the Id of the next field of the data structure/next item in the table. This enables the client to get values of all the items in a table or all the fields in a data structure.

InformRequest is used for communication between two management processes of two different NMSs. GetBulkRequest is used to get bulk repetitive data.

Server responses. The server sends two types of messages to the client:

- Response
- Trap.

Response message is the reply sent to the client's request messages. Trap is a notification sent by a server at scheduled intervals or on occurrence of an event, *e.g.* a router reboots. The server sends the notification indicating the event and its time of occurrence.

The formats of the requests, responses, and trap notifications of SNMPv2 are shown in Figure 22.21.

Version. The version field indicates version of SNMP.

Version	Community	Type	Request Id	0	0	Variable bindings
---------	-----------	------	------------	---	---	-------------------

(a) Format of GetRequest, GetNextRequest, SetRequest, InformRequest.

Version	Community	Type	Enterprise	Agent address	Generic trap	Specific trap	Time stamp
---------	-----------	------	------------	---------------	--------------	---------------	------------

Variable bindings

(b) Format of trap.

Version	Community	Type	Request Id	Error-status	Error-index	Variable bindings
---------	-----------	------	------------	--------------	-------------	-------------------

(c) Format of Response.

Version	Community	Type	Request Id	Non-respeaters	Max-repetitions	Variable bindings
---------	-----------	------	------------	----------------	-----------------	-------------------

(d) Format of GetBulkRequest.

Name 1	Value 1	Name 2	Value 2		Name N	Value N
--------	---------	--------	---------	--	--------	---------

(e) Format of variable binding field.

Figure 22.21 Format of SNMP messages.

Community. It defines an access environment for a group of NMSs. Community names serve as a weak form of authentication.

Type. It defines the type of message as indicated below:

GetRequest	0
GetNextRequest	1
GetResponse	2
SetRequest	3
GetBulkRequest	5
InformRequest	6
Trap	7

Request-Id. It associates requests with responses.

Enterprise. In case of trap, this field indicates the type of agent that generated the trap.

Error-status. It indicates type of error encountered while processing a request. This field is set to zero in the request messages.

Error-index. It points to the variable that caused the error. This field is set to zero in the request messages.

Agent address. It indicates address of the agent that generated the trap.

Generic trap. It contains generic types of traps, *e.g.* value 2 indicates that link is down.

Specific trap. This indicates the specific trap types defined by a vendor.

Time stamp. It contains the time at which the event occurred.

Variable bindings. It is the list of object names and their values. In requests the values are set to zero.

Non-repeaters. It specifies the number of objects from the beginning of the request that should be retrieved only once.

Max repetitions. It specifies the maximum number of times other variables (other than non-repeaters) should be retrieved.

22.10 WORLD WIDE WEB (WWW)

World Wide Web (WWW) is so successful today that it has become synonymous with the Internet. WWW consists of a set of components (including applications) that work cooperatively on the Internet to provide Web services to the users. The major components that constitute the Web are as follows:

- Hypertext Markup Language (HTML)
- Uniform Resource Locator (URL)
- Hypertext Transfer Protocol (HTTP).

Each of the above components is a topic on its own that can cover a chapter. We will just have an overview of these to grasp the overall picture. HTTP is covered in some detail as a separate section.

22.10.1 Hypertext Markup Language (HTML)

Hypertext is a way of representing information in a structured way that allows its easy retrieval. The information is presented as text documents in which links are attached to a string of characters. The links enable access to the next set of information pieces that the string of characters represents. The linked information may reside in other machines which may be located any where on the Internet.

Hypertext Markup Language (HTML) is a text description language and is

based on SGML (Standard Generalized Markup Language). SGML is a system of defining structured document types and markup languages to represent instances of those document types.

HTML is a semantic markup language. The ‘commands’ (tags) are included within the text. These tags define the logical structure of the text. For example, `<p>text</p>` indicates the text bound in a paragraph. There are many types of tags. The most important tag is the link that makes the text ‘hyper’. The linked text is written as `text`, where URL (Uniform Resource Locator) is a unique identifier of a resource on the network. For example, statement `prakash` implies that ‘prakash’ when clicked in the document fetches the information from the location pointed by the linked URL `http://www.xyz.com`.

Tags are device independent and are interpreted at the given output system. Web browsers (Internet Explorer, Netscape Navigator) use HTML Interpreter for presentation of the documents. Thus, all the browsers interpret the HTML documents in same way.

22.10.2 Uniform Resource Locator (URL)

WWW uses Uniform Resource Locators (URL) as identifier of a given information resource on the Internet. `http://www.microsoft.com` is an example of a URL. The URL consists of two parts:

`<scheme (access method) >: <scheme specific part (path to the information resource)>`

Scheme part. The first part indicates the mechanism to access the location as described by the second part. ‘http’ in the example above, indicates that Hypertext Transfer Protocol (HTTP) must be used to access the information resource. Internet Assigned Numbers Authority (IANA) has specified several schemes:

- ftp File transfer protocol
- http Hypertext transfer protocol
- gopher Gopher protocol
- mailto Electronic mail address
- news Usenet news
- nntp Usenet news using News Network Transport protocol
- telnet Interactive session using telnet protocol

- wais Wide area information servers
- file Local file access

Scheme specific part. The second part (Scheme Specific) of the URL has the following general syntax:

`//<user>:<password>@<host>:<port>/<URL-path>`

Some of the parts of the above syntax may not be present in a URL. For example `ftp://@host.com/` does not have user name and password. URL syntaxes for some of the schemes are given below as illustrative examples:

`ftp ftp://<user>:<password>@<host>: port>/<cwd1>/.../<name>;type=<typecode>`

‘cwd’ is change working directory command.

`http http://<host>:<port>/<URL-path>?<searchpart>`

If the port is omitted in the URL, the default port number is 80. The ‘searchpart’ is a query string.

`telnet telnet://<user>:<password>@<host>: port>/`

If the port is omitted in the URL, the default port number is 23.

22.11 HYPERTEXT TRANSFER PROTOCOL (HTTP)

HTTP is used for transporting WWW documents between the client (Web browser) and server (Web server). HTTP is described in RFCs (1945, 2616 and 2817), the current version being used is version 1.1.

HTTP runs on the well-known TCP port 80 for the server (Figure 22.22). The basic operation consists of three steps:

1. The client opens a TCP connection and sends request for a document.
2. The server responds with the document.
3. The server closes the connection.

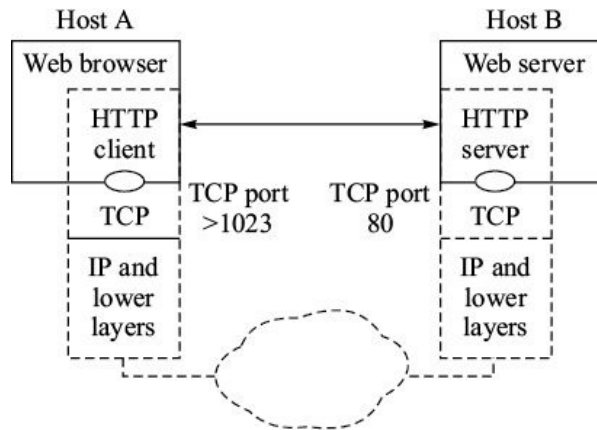


Figure 22.22 HTTP architecture.

HTTP messages from the client to the server are called HTTP request messages. HTTP messages from the server to the client are called response messages. These messages consist of a header and a body as described below. The body contains data described by a MIME header. The data can be HTML document (Web page), graphic, video or sound.

22.11.1 HTTP Request Messages

HTTP request messages from the client consist of three parts (Figure 22.23a). The request line consists of three elements, Method, URL, and HTTP version. The request line terminates

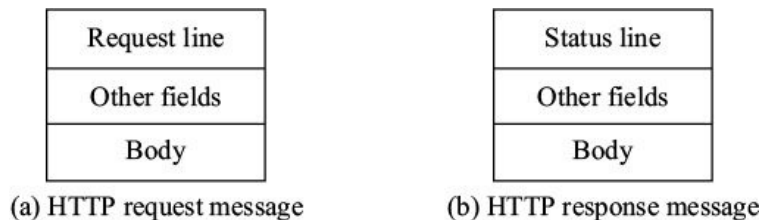


Figure 22.23 Construction of HTTP messages.

with CRLF. The element 'Method' indicates the actual request (e.g. GET, COPY, etc.). The URL element is the requested resource and HTTP version element is the version number of HTTP being used by the client. There are several types HTTP-Methods. The important ones are described below:

GET. GET is a request that allows the client to retrieve the information as identified in the URL. The requested information is returned in the body of the response message of the server. GET request consists of the header only.

HEAD. This request is same as GET except that the server's response consists only of the header as if body were present. This enables the client to get the

resource information without transferring the actual body information. HEAD request consists of the header only.

POST. POST request consists of a header and a body. It is a request to the server to accept the attached entity in the body of the request as a new subordinate to the identified URL.

PUT. It is a request to store the attached entity in the body of the request under the supplied URL. It can be a new URL or an existing URL. In the later case the entity replaces the existing contents of the URL. Note the difference between POST and PUT. In POST request the new contents becomes subordinate, and it does not replace the existing content.

DELETE. DELETE requests the server to delete the resource identified by the URL.

TRACE. TRACE allows the client to see how the message was received and retrieved at the other end. The server returns whatever it receives in the body of the TRACE request from the client. It acts like an application layer loop-back. It is used for diagnostic purposes.

COPY. It is a request to copy the resource identified by the URL in the request line to the locations indicated in the URL-Header field (One of the other fields in Figure 22.23a) that follows the request line of the message.

MOVE. It requests the server to move the resource identified by the URL in the request line to the locations given in the URL-Header field.

22.11.2 HTTP Response Messages

The response of the server to a request from the client consists of three parts—status line, other fields, and body (Figure 22.23b). The status line has three elements, HTTP version, status code, and reason phrase. The status code is three-digit integer that indicates the response to the request. The phrase element is textual brief explanation/reason of status code. For example,

HTTP/1.1 202 Accepted ...indicates that the server was able to fulfill the request.

Many status codes have been defined in HTTP version 1.1 to take care of all possible responses. We can classify them as under. Complete set of status codes can be had from the RFCs mentioned earlier.

- Informational 1xx These include informational responses.
- Success 2xx The request is successfully fulfilled.
- Redirection 3xx Further action by the client is required to complete the request.
- Client error 4xx The request had syntax error or the request cannot be fulfilled.
- Server error 5xx The server failed to fulfill the request.

An example of HTTP exchange is shown in Figure 22.24. Only the request-line of commands and the status-line of responses are shown. An important point to note is that the client opens a TCP connection to send a request and the server terminates the connection after sending the response. For the next request, the client opens a new connection even though the next request is served by the same server from the same TCP port 80.

Cookies. Release of TCP connection immediately after the response is sent by the server creates some problems in applications that require continuity of the session, *e.g.* applications like ‘virtual shopping cart’. This problem is circumvented using ‘cookies’. Cookies introduce session information. Cookie is basically a special header sent by the server. It contains session-Id, lifetime, path, *etc.* The cookie is returned to server by the client when making the next request. The session-Id information enables the server to link the new request to the previous request. Cookies are often used to determine the user behaviour and tastes. Most of the browsers allow turning off the cookies.

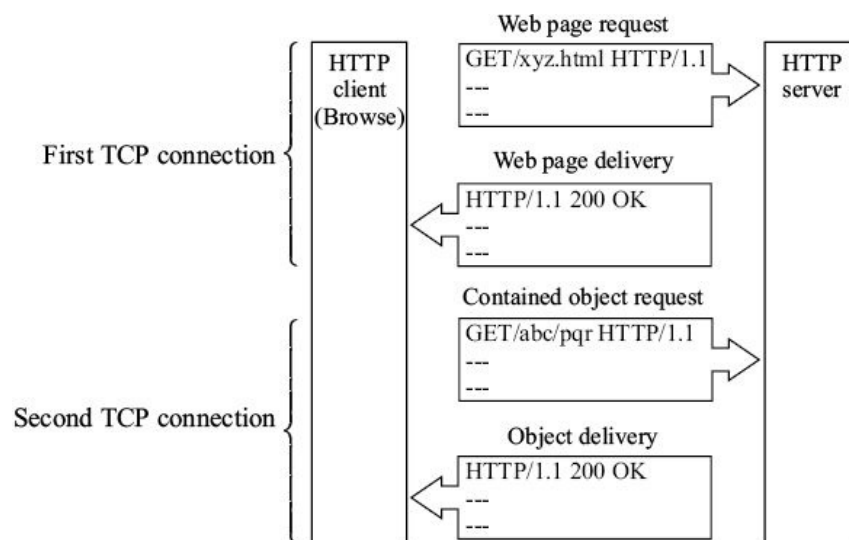


Figure 22.24 Example of typical HTTP exchange.

SUMMARY

All the protocols, from the physical layer to the transport layer are designed to support distributed applications that reside in end-systems. In the TCP/IP suite, the applications rely on TCP and UDP transport protocols for end-to-end exchanges of messages. In addition to the application designed for the specific needs, there are some common distributed applications that users and network managers make use of. The important protocols that support these applications are as follows:

Simple mail transfer protocol (SMTP). It is the most widely used protocol for electronic mail based on simple text. The recent MIME standard expands the scope of SMTP to include multimedia information.

Simple network management protocol (SNMP). This protocol is used for managing Network Elements (NEs) of a network. Network management functions include provisioning, configuring, faults and performance reporting functions. SNMP version 2 is the currently deployed version of the protocol. SNMP version 3 is the latest enhancement that includes security features as well.

Telnet. Telnet is a protocol used for remote login and enables a telnet client to access to an application on a remote system. A telnet client terminal communicates with the telnet server through a canonical terminal representation called Network Virtual Terminal (NVT).

File transfer protocol (FTP). FTP enables transfer of a copy of a file from one end system to another. It reduces all the file systems to a file system having a common minimal set of fundamental properties. FTP handles the following data representations: ASCII (7 bit), EBCDIC (8-bit), and 8-bit binary data.

Trivial file transfer protocol (TFTP). TFTP is used for transfer of files to/from machines that do not have enough resources (RAM, ROM). The code size of TFTP is very small and can be easily fit into bootstrap-ROMs of such machines and devices. Typical applications of TFTP include loading image on diskless machine and upgrading the operating system in network devices such as routers. TFTP is unsecured protocol and does not support authentication.

Bootstrapping protocol (BOOTP) and dynamic host configuration protocol (DHCP). These protocols are used by the clients for getting allotment of IP address, names of boot-files and the IP address of host that serves the boot-files. DHCP extends functionality of BOOTP by allowing the clients to be mobile.

Hypertext transfer protocol (HTTP). HTTP is used for transporting WWW content between the client (Web browser) and server (Web server). The content can be HTML document (Web page), graphic, video or sound. HTTP consists of simple three step process—request from client, response from server, and termination of connection.

Domain name system (DNS) protocol. The Internet uses hierarchical naming system called Domain Name System (DNS). It maps the IP addresses of the hosts to names that are in human readable format. DNS protocol is used for obtaining IP addresses from the name servers. The well-known port for the DNS server is 53.

World Wide Web (WWW) consists of a set of components (including applications) that work cooperatively on the Internet to provide Web services to the users. Hypertext Markup Language (HTML), Uniform Resource Locator (URL), and Hypertext Transfer Protocol (HTTP) are the three major components along with applications that constitute the Web. HTML enables linking URL to a string of characters so as to allow retrieval of information which resides at the resource identified by the URL.

EXERCISES

1. SMTP involves the exchange of several small messages. In most cases, the server responses do not affect the subsequent command. The client can, therefore, implement multiple command-pipelining in a single message. List the commands which may not be pipelined as the server response is needed for the next command.
2. DNS uses UDP instead of TCP which is reliable. If DNS message is lost there is no recovery. Does this affect the usual mode of operation of DNS?
3. UDP packets have maximum length of 576 bytes. Does this affect DNS operation?
4. Write the statement to make the string ‘OPEN’ a hyperlink to <http://www.xyz.com>.
5. The hyperlink of Exercise 4 is clicked. Write the request message generated by the HTTP client.
6. In TFTP, indicate the next message sent by the client/server when
 - (a) the Read Request from the client is lost in transit.
 - (b) a Data message containing less than 512 octets from the client is lost in

transit and the server timesout.

(c) the acknowledgement from the server is lost and the client timesout.

7. A client wants to get a 1 kbyte file from the server using TFTP. Write the messages exchanged between the servers and the client.

8. The given figure shows exchange of UDP messages between a TFTP client and server. Only the source and destination fields of the UDP header are shown attached to the TFTP message.

(a) Indicate which end system is the client and which is the server.

(b) Fill in the blanks from the following choices.

231, 69, 347, Read Request, Acknowledgement, Write Request, Data

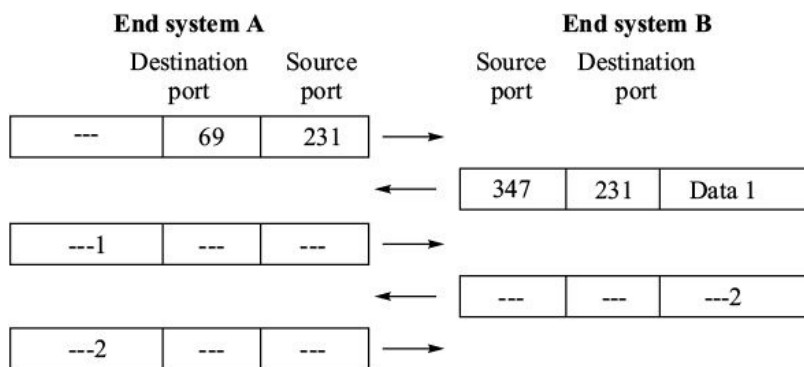


FIGURE E 22.25.

9. When the web pages are sent out, they are prefixed by MIME headers. Why?

¹ Namespace is set of all the names. Flat namespace has no structural hierarchy. If we remove aliases, the namespace of human beings is flat.

23

Quality of Service

Simple forwarding function of an IP node can provide ‘best effort’ datagram service without any assurance of throughput and timely delivery of the packets. Additional functions must be built into the network nodes to assure a minimum level of quality of service. This chapter focuses on these additional functions and the frameworks for implementing quality of service in an IP network. We begin the chapter with the definition of quality of service parameters. Various queuing systems, packet drop policies, traffic shaping and policing functions are described next. There are two major frameworks for implementing quality of service in IP networks, namely, integrated service and differentiated service. We describe these frameworks and the protocols for their implementation. We conclude the chapter with a comment on mapping differentiated service framework in MPLS domain.

23.1 MOTIVATION FOR QUALITY OF SERVICE (QOS)

An IP network serves a wide variety of applications—many more than simple Internet applications discussed in Chapter 22. The network performance required by different applications is different and depends on the nature of an application. We can broadly categorize these applications as:

- Real-time applications
- Non-real-time applications.

Real-time applications. For the real-time applications, timeliness of data is important, *i.e.* the data must be made available within certain time bounds else it loses its worthiness. We have the following two sub-categories of real-time applications:

1. Intolerant real-time applications

2. Tolerant real-time applications.

Applications involving financial transactions, robot commands are typical examples of intolerant real-time applications. These applications are sensitive to timeliness of data and are intolerant to loss of data. For example, loss of data pertaining to a banking transaction may result in financial loss.

There are other real-time applications that are tolerant to data loss. Voice and video signals are typical examples of such applications. These signals are not very sensitive to occasional loss of a voice or video sample because the receiver can readily interpolate the missing samples. But they are sensitive to end-to-end delay and its variation (termed as *jitter*). Transport of encoded samples of these signals requires consistent and bounded end-to-end delay and jitter; otherwise voice quality becomes bad and picture movements become jerky.

Non-real-time applications. These are conventional data applications such as FTP, Telnet, Web browsing and e-mail. These applications can tolerate increased delay and are insensitive to jitter. Note that these applications can benefit from shorter-length delays. The delay requirements are different for different applications.

- Interactive applications require consistent time bounded delay otherwise the sessions may time out.
- End-to-end delay is not as critical for FTP. But it is sensitive to throughput as inadequate throughput can increase the file transfer time significantly.

23.1.1 QOS in IP Networks

The capacity of a network to deliver a defined performance level of network service is called its quality of service (QOS). QOS is defined in terms of certain network performance parameters described later. Assuring a certain level of quality of service in the connection oriented networks like ATM is simple, because the required network resources are reserved at the time of connection set up. An IP network, on the other hand, provides datagram service with the following features that cannot guarantee quality of service:

- IP routers forward the packets towards destination without any regard to the type of data carried by them. In other words packets carrying voice signals will not get any forwarding priority over other data packets.

- Datagram service of an IP network is best effort service, *i.e.* the network makes best effort for delivery of a packet. There is no guarantee that packets will be delivered, or delivered timely, or sequentially.
- IP network does not reserve network resources for individual flow of packets from its source to the destination.

To get an assured minimum level of quality of service (QOS) from an IP network, the routers need the following basic capabilities:

- The routers should have capability to classify IP packets so that packets belonging to a class may be selectively favoured over the packets of another class.
- The routers should be capable of giving differential treatment to various classes of IP packets.

“We need to augment the basic forwarding function of a router with some additional tools to give differential treatment to the classified packets. These tools build the following capabilities in the IP network to make it QOS capable:

- Assigning priorities and forwarding packets based on priority.
- Applying a discard policy selectively when congestion occurs in the network.
- Reserving network resources (e.g. bandwidth) for the selected class of packets.
- Controlling the traffic by ensuring that the users do not pump data packets at higher than contracted data rate and only the authorized users get the contracted service.

It is to be borne in mind that these capabilities improve network performance for some classes of IP packets at the cost of others. By implementing QOS in an IP network we are purposefully choosing to favour one packet over another.

23.1.2 Contractual Agreements for QOS

The contractual agreement for QOS between a network service provider and a customer consists two parts:

- Service level agreement (SLA).
- Traffic conditioning agreement (TCA).

Service level agreement (SLA). Service level agreement consists both technical and non-technical aspects. Non-technical aspects deal with commercial aspects (pricing, penalties for non-performance, etc.) The technical aspects define the QOS parameters and their values which the service provider commits to the customers. These parameters include bandwidth, delay, jitter and packet loss. There can be finer granularity of definition of these parameters, *e.g.* bandwidth parameter may include a committed information rate (CIR) and excess information rate (EIR).

Traffic conditioning agreement (TCA). Traffic conditioning agreement defines the technical aspects of obligations on the part of the customer. QOS commitments are made for certain traffic profile. Traffic profile defines the average data rate and maximum size of data burst. The customer is not expected to exceed the bounds imposed by TCA. If traffic from a customer exceeds these bounds, not only the QOS committed to him will get affected, his excess traffic may disturb service to other customers as well. A customer is required to ‘shape’ his traffic to the contracted traffic profile. The service provider on his part ‘policies’ the traffic so that a customer may not take undue advantage. We will describe shaping and policing functions later in this chapter.

23.2 QOS PARAMETERS

Quality of service is defined in terms of the following parameters: – Bandwidth

- Delay
- Jitter
- Packet loss

23.2.1 Bandwidth

The term *bandwidth* of link refers to the number of bits that can be transported in a second across the link. Its maximum value is restricted to the bit rate of the physical link (Figure 23.1). There are tools that can limit the average number of bits that can be sent in a second on a physical link. For example, a 2048 kbps link can be restricted to bandwidth of say 128 kbps by allowing an average 128 kilobits in a second. The tools used for restricting the bandwidth to a given value are *traffic shaper* and *traffic policer*. We describe these tools later in the chapter.

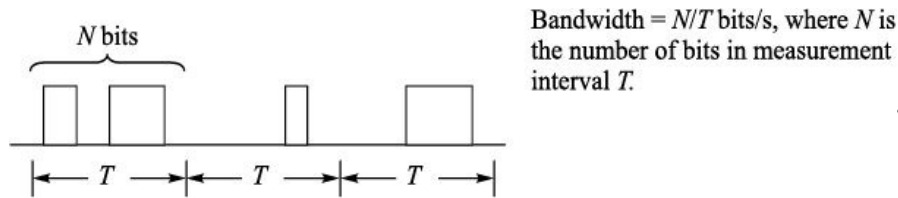


FIGURE 23.1 Bandwidth.

Layer-2 and layer-3 bandwidth. Traffic shaper and traffic policer can be deployed at layer-2 or layer-3. Layer-2 tools restrict the available bandwidth to the frames and layer-3 tools restrict the available bandwidth for IP packets. Thus one should not be taken by surprise by the terms layer-2 bandwidth and layer-3 bandwidth.

23.2.2 Delay

One-way delay (or latency) is the amount of time taken to transport a packet from one end point in the network to another end point. Various components of end-to-end delay are as follows: – Forwarding delay (F)

- Queuing delay (Q)
- Serialization delay (S)
- Propagation delay (P).

Figure 23.2 shows these delay-components for a typical link between two routers R1 and R2. Total end-to-end delay is aggregate of these delays for each intervening link of the path taken by a packet across a network.

Forwarding delay (F). It is the amount of time taken to forward a packet from the input interface of a router to its *uncongested* output interface. If there is congestion at the output interface, it is the time taken to place the packet in the queue at the output interface. In other words, forwarding delay does not include the time a packet spends in the output queue.

Queuing delay (Q). Queuing delay is the amount of time that a packet spends waiting in the queue at the output interface of a router. This delay is variable and depends on several factors:

- The number and size of the packets waiting ahead in the queue.
- The bit rate of output interface. A queue is drained faster when bit rate of the output interface is high.
- The amount of service a queue receives when there are multiple queues at the output interface. The queues can be served in round-robin fashion, or in

order of priority assigned to them, or in a custom-designed manner. The queue which is served more has less delay. We shall study the queuing methods later in the chapter.

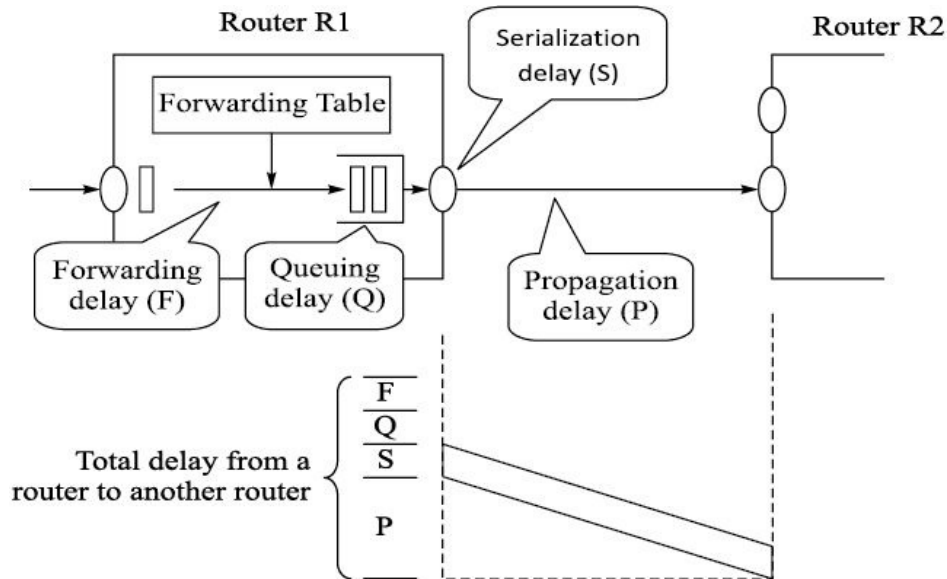


FIGURE 23.2 Components of end-to-end delay.

Queuing delay can be significant and variable. Since other delays are either insignificant or constant, the variability of queuing delay is the major contributing factor to jitter.

Serialization delay (S). It is the time taken to push a packet encapsulated in a frame out of the output interface. It can be calculated by dividing number of bits in a frame by the bit rate. Note that we need to take into account the header and trailer of the layer-2 frame which encapsulate the packet. Serialization delay is of significance for the low bit rate links. It takes 0.000488 ms to send a bit from the output interface if the bit rate is 2048 kbps.

Propagation delay (P). Propagation delay is the time taken for transporting a bit on the physical transmitting medium. It can be estimated readily knowing the length of the physical medium and propagation speed of signals on the transmission medium. Propagation delay is constant between two end points. It changes when packets take alternate path. This can happen when traffic is diverted due to operational constraints, *e.g.* physical link breaks down or there is congestion.

23.2.3 Jitter

Jitter is average variation in end-to-end delay in delivery of packets. Typically, data applications are not sensitive to jitter. But performance of real-time application like packetized voice is seriously affected by jitter. A voice encoder generates uniformly spaced encoded voice samples and the decoder requires the received samples also to be spaced uniformly. The network disturbs the uniform spacing of the samples by introducing varying delay (Figure 23.3a).

Jitter can be removed by introducing additional jitter-offset delay to equalize the end-to-end delay of all the packets (Figure 23.3b). Removal of jitter is possible if the jitter is bounded. If the first received packet is given an offset delay equal to the upper bound of jitter, jitter can be removed from the subsequent packets. Therefore the customers demand an upper bound on jitter.

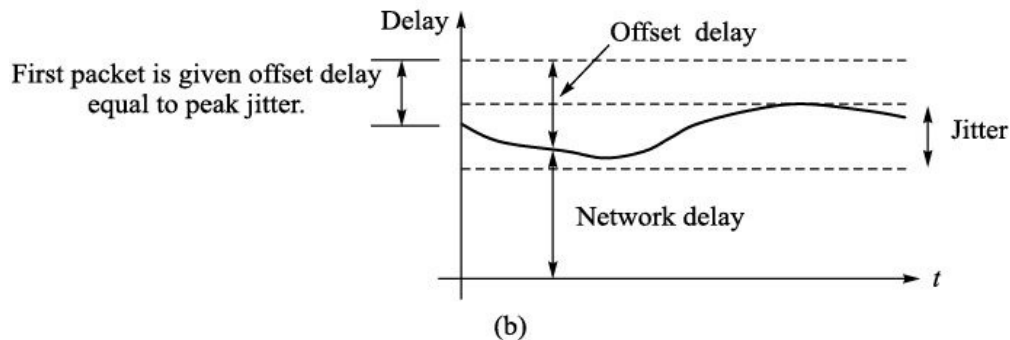
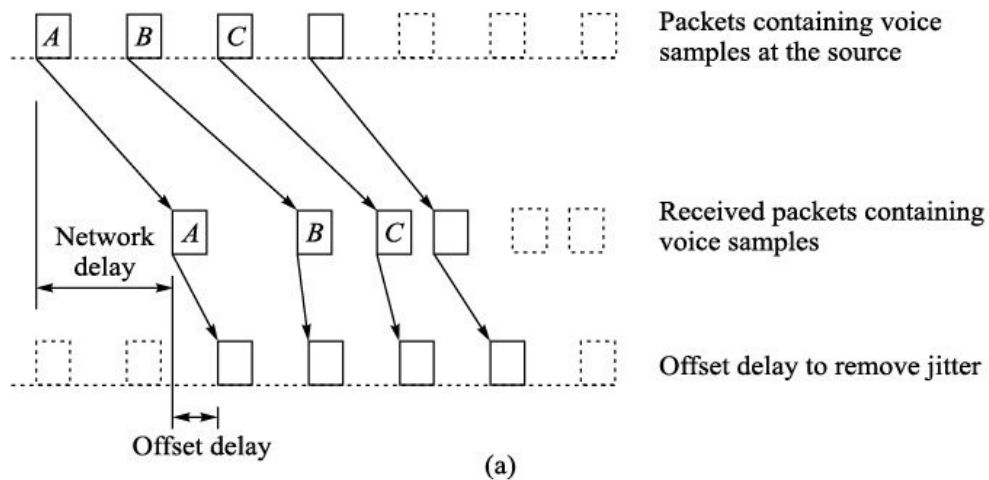


FIGURE 23.3 Removing jitter by introducing offset delay.

23.2.4 Packet Loss

There can be several causes of loss of a packet in a network. Some of them are as follows:

- High error rate of transmission medium can introduce errors that result in

discarding of frames at layer-2. But packet loss due to bit errors is small in most networks today because physical links are on optical links with BER better than $1 \cdot 10^{-9}$. The resulting packet loss is typically less than one in one billion.

- A router may drop a packet due to any of the following reasons:
 - A router drops a packet when it does not have resources to forward the packet. Packets received by a router are queued in a buffer. If the buffer is full when a packet arrives at the router, the router drops it. A router may also drop a packet proactively when the queue depth exceeds a defined threshold.
 - If the traffic received from a customer exceeds TCA, the traffic policer in the router may drop the packet.

Packet loss can have serious implications for the TCP based applications. Recall that when TCP segments are lost, TCP goes into slow start mode. This reduces the throughput and therefore slows down the applications.

23.3 FUNCTIONS REQUIRED FOR SUPPORTING QOS

Routing and forwarding are the two basic functions of a router. These basic functions are not adequate to construct a network having some minimum guaranteed quality of service. A network based on these two functions only would be like having a network of roads with signboards giving directions at the crossings. For smooth flow of the vehicular traffic we need, in addition, traffic signals for management of traffic, reserved lanes for different types of traffic and policemen to maintain traffic discipline. An IP data network also requires similar functions built into it. These additional functions pertain to the following areas:

23.3.1 Traffic Classification and Scheduling

As mentioned earlier, implementing QOS implies that we are choosing to favour one packet over another. Traffic classification refers segregating IP packets in queues based on some criterion, *e.g.* type of traffic they carry, their source and destination addresses, and so on. Scheduling determines how the various queues are served for dispatching packets. A queue that is served more frequently is drained faster.

23.3.2 Traffic Control

From the QOS angle, traffic control includes the following functions: **Traffic shaping**. It refers to limiting the outgoing traffic from an interface to its specified bounds in terms of data rate and maximum data burst size.

Traffic policing. It is the complementary function to traffic shaping. It ensures that the incoming traffic to an interface does not exceed its data rate and the maximum burst size.

Admission control. Admission control function admits new traffic only if the network has resources to handle the additional traffic. It is applicable when the network resources are required to be reserved before sending the traffic.

23.3.3 Congestion Management

When there are several waiting packets to egress through an interface of a router, we say the interface is congested. These packets are put in a queue and are transmitted on first-come-first-served basis. Congestion management refers to managing queue depth. Buffers for the queues in the routers have limited capacity and when they are full, the routers drop the packets. Congestion management is carried out by implementing a packet drop policy and a congestion notification system.

We will describe the tools required for these functions in the following sections of this chapter. Thereafter we will study the frameworks for implementing QOS in IP networks using these tools.

23.4 QUEUING SYSTEM

When an IP packet is received at one of the interfaces of the router, it is buffered in an input queue (Figure 23.4). The router takes one packet at a time from the input queue and determines the output interface for the packet. This decision is based on the forwarding table. The input queues are usually very small and the packets spend very little time waiting in the input queue.

It is possible that the output interface for a packet as determined from the forwarding table is congested, *i.e.* it is busy in sending another packet. A router has two options in such case, it can drop the packet or it can maintain a queue at the output port. Queuing is more desirable because it saves the users from retransmitting the packets. The output queue plays very crucial role in managing

the end-to-end delay and jitter.

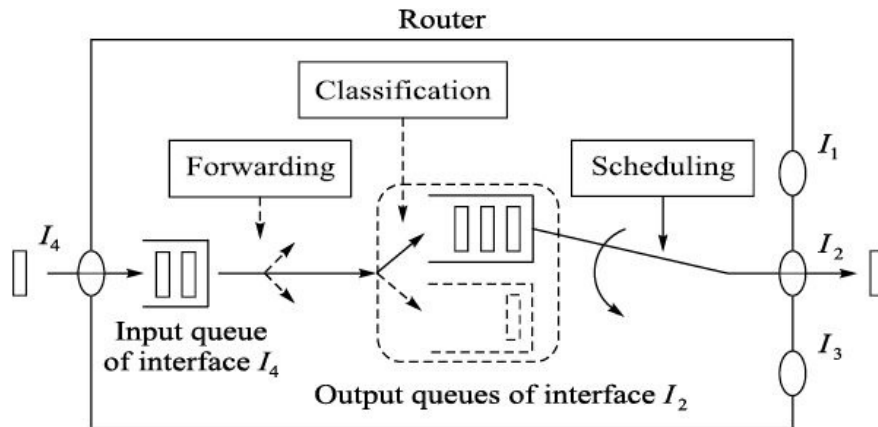


FIGURE 23.4 Input and output queues in a router.

There can be single output queue or multiple output queues (Figure 23.4). Single output queue has several limitations as we will see shortly. Multiple output queues permit us the flexibility of giving different priorities and allocating different bandwidths to the queues. The number of output queues at each interface of a router can vary from one to several hundreds. A multiple queuing scheme consists of the following two basic functions: – Classification
– Scheduling.

Classification. Classification function of a router determines the class of the received packet based on a defined criterion and puts it in appropriate output queue. Each output queue is meant for a class of packets. The following fields of an IP packet are used for classification. Classification can be based on one field or a combination of these fields:

a. Source IP address	(IP header)
b. Destination IP address	(IP header)
c. Protocol	(IP header)
d. Source TCP/UDP port number	(TCP/UDP header)
e. Destination TCP/UDP port number	(TCP/UDP header)
f. TOS field	(IP header)

In the description that follows, we will come across the term *flow* in context of communication between two end users. IP packets flowing from a user to the other user constitute a flow. Packets belonging to a flow are identified by the first five fields listed above. IP packets having the same value in the five fields belong to the same flow from the source to the destination. Between two users,

each direction of flow of packets is a separate flow.

Scheduling. Scheduling function determines how the output queues of an interface are to be served. There can be several alternative strategies for serving queues. For example, the queues can be served in round-robin method or in order of priority. Queues can have weights, so that some of the queues are drained more than others on each instance of service. Note that, the packets in a particular queue are always served on first-in-first-out (FIFO) basis.

The basic queuing schemes used in the routers are as follows:

- Single queue
- Priority queuing
- Weighted round-robin (WRR).
- Fair queuing (FQ), weighted fair queuing (WFQ).

A network may implement mixed queue types in router. For example, a router interface may have one priority queue and the rest WRR queues. Remember that these queuing schemes are used for output queues only. We always have only one input queue at each interface.

23.4.1 Single Queue

This is the simplest and the most obvious way of queuing. There is only one queue and the packet at the head of the queue is sent first. The queue has a maximum defined depth and packets are dropped when the queue size is exceeded. Single queue handles all the packets in same manner irrespective of the type of data, size, priority, or source/destination addresses. This results in some major limitations from QOS angle, which are as follows:

- The aggressive applications like FTP that involve bulk transfer of data, can monopolize the link and can severely degrade the performance of other applications.
 - Interactive applications like Telnet may slow down.
 - Real time and mission critical applications that require low and bounded delay may be affected due to non-availability of data timely.
 - Increased jitter due to large delay variation may affect performance of voice and video applications.

Figure 23.5 shows three users A, B and C attached to router R. The WAN interface of the router operates at 2 Mbps, which is significantly lower than the

10 Mbps ethernet ports of the two LANs. Users A and B are running FTP session and C is running a Telnet session. C's packets are much smaller and fewer but they have to wait behind large packets from A and B. This results in large waiting time in the queue for the Telnet packets from C. C will have a frustrating experience running the Telnet session due to delayed response.

- During periods of severe congestion when the queue buffer becomes full and packets are dropped, UDP flows are benefited over TCP flows. Although packets get dropped irrespective of their type, packet loss causes TCP to go into slow start mode. The UDP flows remain oblivious to the packet loss.

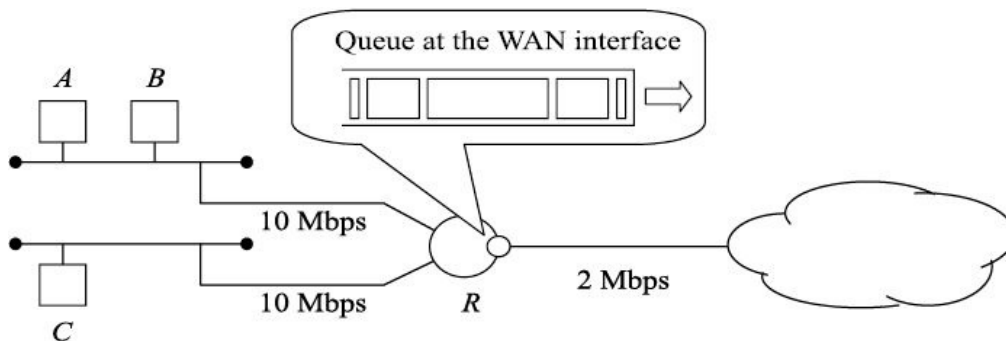


FIGURE 23.5 Single output queue.

Despite of the above limitations, single queue is widely used queuing mechanism in the Internet for the following reasons:

- It is simple, fast and supported by all router platforms.
- It places very small computational load on the router when compared to more elaborate queuing disciplines described later.
- It has predictable behaviour. Maximum depth of the queue and the bit rate of the output interface determine the queuing delay.

23.4.2 Priority Queuing

The basic features of priority queuing are as under:

- There are several queues, each with a different priority level. In practice, four priority levels are defined—high, medium, normal and low. Each queue has defined depth and packets are dropped when a queue size is exceeded.

- A high priority queue is always served before the low priority queue. Or put in another way, a queue is served only when the high priority queues are empty. If a new packet enters in the high priority queue when a low priority queue is being served, the scheduler completes transmission of the current packet in low priority queue and switches over to the high priority queue.
- Potentially there could be denial of service to lower order queues if the high priority queue has uninterrupted flow of packets. It is, therefore, necessary that the high priority traffic is policed at the input interface to restrict it to the committed data rate. The policing function is described later in the chapter.

The criteria for classification of packets into various priority levels can be based on several different parameters. Some typical examples of the parameters are **Source address**. IP packets coming from specified source addresses are given a priority level. For example, IP packets from A are sent to medium priority queue (Figure 23.6).

Protocol type. Recall that an IP packet contains protocol type field that identifies the type of payload. The priority can be specified for specific protocol type. For example, Telnet packets can be sent to high priority queue and FTP packets can be sent to low priority queue (Figure 23.6). This classification is irrespective of the user.

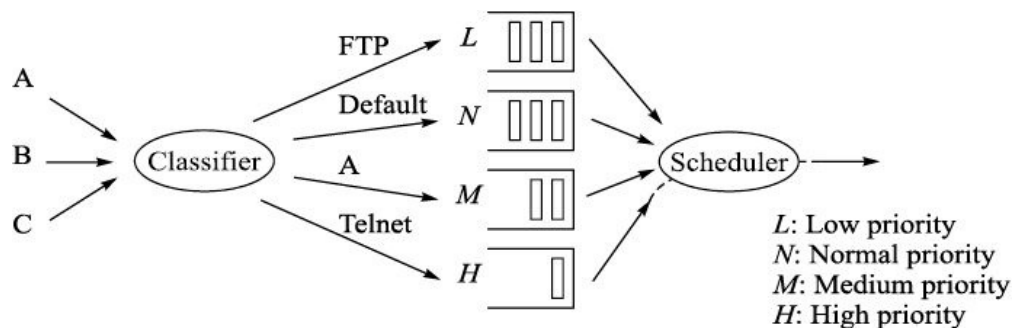


FIGURE 23.6 Priority queuing.

Precedence value. The first three bits of TOS field of an IP packet define its precedence value. This value can be used for classification of IP packets and priority can be assigned to each class.

Mixed set of parameters can also be defined for the various queues. The default queue is normal priority queue. IP packets that do not fall under any classification are sent on this queue.

Priority queuing finds applications where the traffic characteristics are

predictable so that the bandwidth limit and depth of a priority queue can be defined with reasonable accuracy. Unpredictable behaviour of the traffic in the priority queue can severely affect the overall performance as explained earlier. The two typical applications of priority queuing are as given:

- Priority queuing is typically used in VoIP¹ environment where the packets containing encoded voice samples need to be delivered within strict bounds of jitter and delay.
- Priority queuing is used for the routing protocol traffic and network management traffic so that even during periods of congestion, the network routing and management functions are not affected.

23.4.3 Weighted Round-Robin (WRR) Queuing

Weighted round-robin queuing mechanism is known by two other names, *class-based queuing* and *custom queuing*. Its basic features are as follows:

- The packets are classified into various service classes (e.g. real time, interactive, file transfer) and then assigned to a queue dedicated to that service class. The number of queues can be typically 4 to 16 at each interface.
- The scheduler serves the queues in round-robin fashion, draining in each round, configurable number of octets from each queue. After a round is completed, the cycle is repeated.
- The weight of a queue determines the number of octets that can be drawn from the queue in each round of service. For example, if there are four queues having weights 0.4, 0.3, 0.2 and 0.1, the octets drawn from each queue in a round are in the same proportion (Figure 23.7).
- The whole packet is transmitted always. If the defined octet limit is crossed in the middle of a packet, the router drains the whole packet from the queue before switching over to the next queue.
- Each queue, thus, gets allotted a fraction of bandwidth of the output port. For example, if there are four queues having weights 0.4, 0.3, 0.2 and 0.1, the bandwidth is shared in the proportion 4:3:2:1 amongst the four queues.

The number of octets that are drained in a round from a queue is determined keeping in mind the average size of packets in the queue.

- If these octet-limits are less than the size of packets, only one packet will be serviced from a queue in each round. This will result in sharing of the bandwidth in proportion to the average size of packets. Class requiring large bandwidth but having small size packets will be affected most.
- If these octet-limits are too large, the scheduler will spend long time serving a queue each time it visits the queue. This will result in increased jitter and irregular bursts of data.

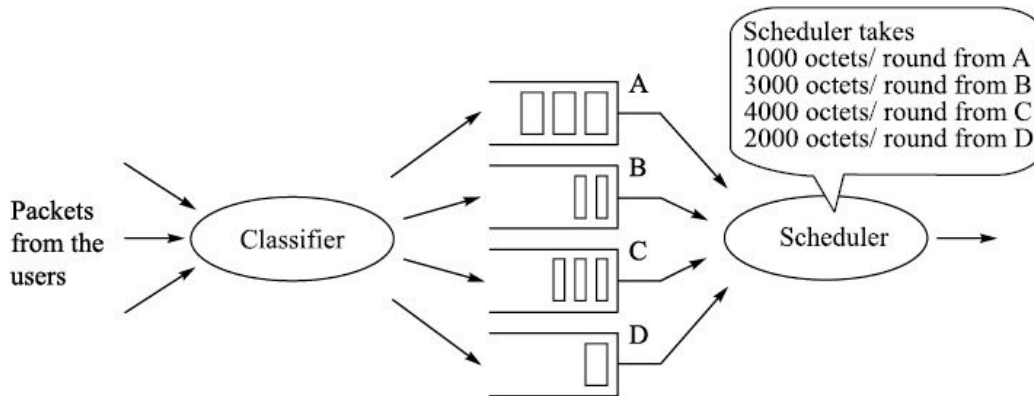


FIGURE 23.7 Weighted round-robin (WRR) queuing.

The advantages of WRR queuing are as follows:

- WRR queuing can be implemented in hardware and therefore it can be applied to high bit rate interfaces.
- WRR queuing provides coarse control of bandwidth to a class of traffic.
- WRR ensures that no queue is ever starved of bandwidth by serving every queue once in a round.
- By aggregating numerous flows into few classes (4 to 16) and enabling coarse bandwidth control, WRR serves as very efficient queuing mechanism for supporting differentiated service QoS framework that we discuss later in the chapter.

The primary limitation of WRR queuing is that it assumes that average packet size of various traffic classes is known in advance. Knowing the bandwidth allocation to the traffic classes and average size of packets, we can compute the number of octets to be drawn from a queue in each round of service. If the size of packets in a queue has high variance, the projected distribution of bandwidth can go haywire.

Deficit weighted round-robin (DWRR) Queuing. This is a variation of WRR

queuing that addresses the issue of variable size of packets in a queue. In DWRR queuing, each queue has a deficit counter which indicates the number of octets that can be drained from a queue. The counter is initialized to a value equal to the allotted quantum of octets to the queue. When the scheduler visits a queue, it drains the queue, decrementing the counter by an amount equal to the number of octets drained. A queue is drained till the packet at the head of the queue has more octets than the remaining value in the counter. The remaining value in the counter is utilized in the next round of service. Before the next round of service, the allotted quantum of octets to queue is added to the remaining value in the counter. If a queue is empty when the scheduler visits it, the deficit counter is set to zero. DWRR ensures that a queue never gets more than the allotted share of bandwidth.

23.4.4 Fair Queuing

Fair queuing, as the name suggests, provides fair allocation of output bandwidth to all the flows. As mentioned in the beginning of this section, a flow is from a source to a destination is identified by the five fields, *i.e.* source IP address, destination IP address, protocol, source port number and destination port number. Basic features of fair queuing are as under:

- Each flow is assigned a separate queue (Figure 23.8). Since each flow has its own queue, no one flow can disturb service to other queues by sending more packets.

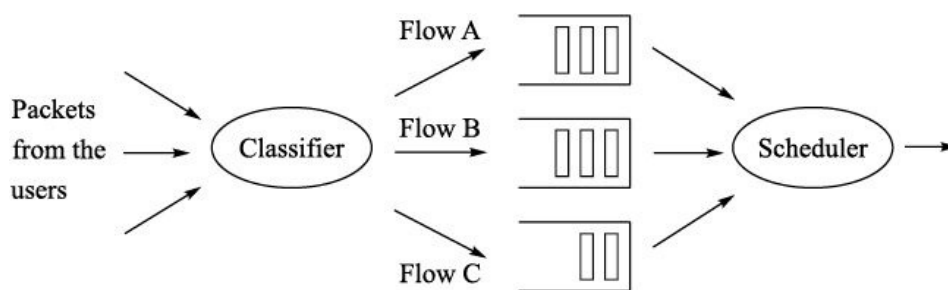


FIGURE 23.8 Fair queuing.

- The router creates a new queue dynamically when it detects a new flow. When the queue of a flow is empty, the router automatically deletes the queue. When another packet of the flow arrives, the router recreates a new queue for the flow.
- The maximum number of queues on an interface of a router depends on its

design. Typically there can be maximum 128 fair queues on an interface.

- Fair allocation of output bandwidth is achieved by serving the queues in round-robin method, draining *notionally* a bit at a time from each active queue. Thus if a queue has a packet containing 10000 bits, it will be emptied notionally in 10000 rounds.

If R bits are transmitted in one second from the output interface and there are N active queues, each queue will get its turn at intervals of N/R seconds. For example, if there are 10 queues and the bit rate at the output interface is 10 Mbps, and if all the queues always have at least a bit to transmit, each queue will be drained at the rate of 1 Mbps. Thus each queue gets fair share of the available bandwidth. Note that the outgoing link is never idle so long as there is at least one bit in any queue. A queuing scheme with this feature is called *work conserving*.

Finish number. In practice, we process a packet as one entity and therefore we cannot send one bit from each queue. The entire packet must be transmitted in one go. If the packet sizes of the queues are same, sending one packet from each queue in a round approximates fair queuing. If the packet sizes are different, and one packet is sent from each queue in a round, the queue with larger size of packets will get higher share of the available bandwidth. For example, if the flow in queue C has packets of size twice those of flows A and B, and if one packet is transmitted at a time from each queue, C will get twice the share of bandwidth as compared to A and B.

This problem is resolved by assigning a *finish number* to every packet. Finish number is the *round number* in which packet would have been completely transmitted, had the bit-by-bit fair queuing mechanism been used. It is assigned when a packet arrives in a queue. Let us see how the finish number is arrived at.

A packet of size $P(i)$ bits will require $P(i)$ rounds of scheduler to transmit it irrespective of the status of other queues. Thus if the queue is empty when the $P(i)$ -bit packet arrives, the finish number $F(i)$ of the packet is given by: $F(i) = P(i) + N$

where, N is ongoing round number when this packet arrives. If there are packets ahead in the queue already, the new packet will be transmitted only after the preceding $(i-1)^{\text{th}}$ packet in the queue. In this case the finish number is given by: $F(i) = P(i) + F(i-1)$ where, $F(i-1)$ is the finish number of the preceding packet in the queue.

To emulate the bit-by-bit performance of fair queuing, the packet with the lowest finish number is transmitted ahead of others. Entire packet is sent in one go. Note that round number that we mentioned above has notional significance and is used for calculating the finish numbers that determine the order of transmission of the packets from different queues. The following examples illustrate the mechanism.

EXAMPLE 23.1 A router uses fair queuing mechanism and has three queues A, B and C. The packet sizes of the three queues are 11, 23 and 67 octets respectively. The queues always have continuous flow of incoming packets. Assuming that the finish number is based on the octets in a packet and the scheduler begins the first round with queue A, determine the sequence of flow of packets at the output interface.

Solution

Since there is continuous flow of packets, the finish number of a packet is equal to the finish number of last packet plus the current packet size. Thus the finish numbers of the packets in the queues A, B and C are as under:

Queue A : 11, 22, 33, ...
 Queue B : 23, 46, 69,...
 Queue C : 67, 134, ...

The packets are transmitted from these queues in the increasing order of their finish numbers as shown in Figure 23.9.

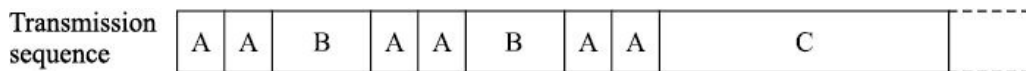


FIGURE 23.9 Example 23.1.

EXAMPLE 23.2 Three queues A, B and C have one packet each of size 1 octet, 2 octets and 4 octets respectively at $t = 0$. The queues are serviced at the rate of one octet per second. Queue A receives another packet of 2 octets at $t = 2.5$ s.

1. Determine the finish numbers of the packets. Assume that the finish numbers are based on octets.
2. When is the transmission of the second packet of the queue A completed?

Solution

1. The finish numbers of the first packet in the queues A, B and C are 1, 2 and

4 respectively. To find the finish number of the second packet that arrives in queue A at $t = 2.5$ s, we need to determine the notional ongoing round number at the time of its arrival. Round numbers are calculated assuming transmission of one octet from each queue in round-robin fashion.

– In the first round all the three queues have octets to send. Therefore the first notional round consists of sending three octets. The first round is notionally completed at $t = 3$ s.

– Thus the second packet of queue A arrives during the first notional round. Its finish number will be $1 + 2 = 3$.

2. Transmission of first packets of queues A and B is completed at $t = 3$ s. Since the second packet of queue A is already in the queue at $t = 3$ and has lower finish number than the packet in queue C, it is transmitted next. Its transmission is completed at $t = 5$ s.

The primary benefit of fair queuing is that a misbehaving flow cannot degrade the quality of service of other flows because each flow is isolated into its own queue and the scheduler serves the queues fairly by distributing the bandwidth uniformly to the active flows. When unequal distribution of bandwidth is required, weighted fair queuing, described next, is used. Fair queuing has two basic limitations: – It is implemented in software and involves considerable computational overhead. This limits its application to lowspeed interfaces at the network edges.

– The number of flows at an interface of a core routers in a network may be very large, several ten thousands. Supporting so many fair queues has design limitations.

23.4.5 Weighted Fair Queuing

Weighted fair queuing extends the concepts of fair queuing for non-uniform bandwidth distribution amongst various queues. This is achieved by assigning a *weight* to each queue. The bandwidth of the outgoing link gets distributed in proportion of the assigned weights. For example, if the weights of 1, 0.5, and 0.5 are assigned to queues A, B and C respectively, 10 Mbps bandwidth gets distributed as 5 Mbps to A, and 2.5 Mbps each to B and C. Intuitively, this can be achieved simply by taking a fraction (w) of bit instead of one bit from queue having weight w in each round. Since we are going to transmit packets as whole, we need to redefine the finish number.

In fair queuing, a P -bit packet require P notional rounds for its transmission.

In weighted fair queuing, a queue having weight of say 0.5, will receive *half-service* in a notional round, and therefore will require $2P$ rounds to transmit a P -bit packet. The equations we derived for calculating finish numbers get modified as given below for weighted fair queuing. If w is the weight assigned to a queue, finish number is given by:

$$F(i) = P(i)/w + N, \quad \text{when the queue is empty, and}$$

$$F(i) = P(i)/w + F(i-1) \quad \text{when the queue contains packets waiting for their turn.}$$

Assignment of weights to different queues requires a criterion to be defined so that weights get dynamically assigned as the queues are dynamically created. The precedence bits of TOS field in the IP header can be used for this purpose. Precedence bits can take values from 0 to 7. A queue is assigned weight roughly in proportion to its precedence value. If there are n active flows, the weight w_i of i th flow is calculated as under: $w_i = (p_i + 1)/S$ ($p_i + 1$) where, p_i is the precedence value of the i th flow. For example, if there are three flows of precedence 5 and one each of the other precedence values, the weight assigned to the flow with precedence value 4 can be calculated as under: $w = (4+1)/[(0+1)+(1+1)+(2+1)+(3+1)+(4+1)+(5+1)+(5+1)+(5+1)+(6+1)+(7+1)] = 5/48$

Weighted fair queuing has the same benefits and limitations as of fair queuing. WFQ is deployed at the edges of a network to provide fair distribution of bandwidth.

Class-based weighted fair Queuing. In class-based weighted fair queuing, there are two stages of classification. The traffic is classified as in WRR queuing, and then, flow-based queues are formed within each class as in weighted fair queuing. Each class is assigned fraction of output bandwidth and the flows within a class share the bandwidth allotted to the class based on their weights.

Table 23.1 gives comparative summary of the features of the basic queuing schemes.

TABLE 23.1 Comparative Features of Queuing Schemes						
	Single queue	Priority queuing	Weighted round-robin queuing	Weighted fair queuing		
Typical number of queues	1	4	16	As many as number of flows		
Classification	No classification.	User defined.	User defined	Flow-based.		

Scheduling	First-in-first out.	Higher priority queue is given preference.	Round-robin	Based on finish numbers
Bandwidth distribution	Not applicable.	Higher priority queue has overriding right of bandwidth usage.	Customized	Fair distribution of bandwidth.

23.5 TRAFFIC CONTROL

A data network is engineered to provide quality of service to a defined volume of data traffic. When the traffic volume exceeds the designed limits, congestion in the network results deterioration in network performance. Congestion can also occur if some of the part of the network fails. For example, if the link interconnecting two routers fails, the packets take alternate path and may cause congestion in the alternate path. Traffic management encompasses all such issues and is of concern for providing a defined level quality of service. We will focus in this section on the control of traffic that enters a network. Other aspects of traffic management are outside the scope of this book.

23.5.1 Traffic Characterization Parameters

When service level agreement (SLA) between a network service provider and a customer is drafted, the commitment to a certain level of quality of service is defined in terms of delay, packet loss and jitter. The QOS commitment is made for specified characteristics of the data traffic from the source. The data traffic is specified in terms of the following parameters: – Peak data rate

- Average data rate
- Maximum burst size.

Peak data rate. Peak data rate is the highest rate at which a source is allowed to send data. Peak rate can be computed in two ways. For the networks with fixed size packets (e.g. ATM cell), peak rate is inverse of the closest spacing between the starting times of two consecutive packets (Figure 23.10). For the variable sized packets (e.g. IP packets), we need to specify a time window (also called *measurement interval*) and the maximum number of bits that can be sent in that interval.

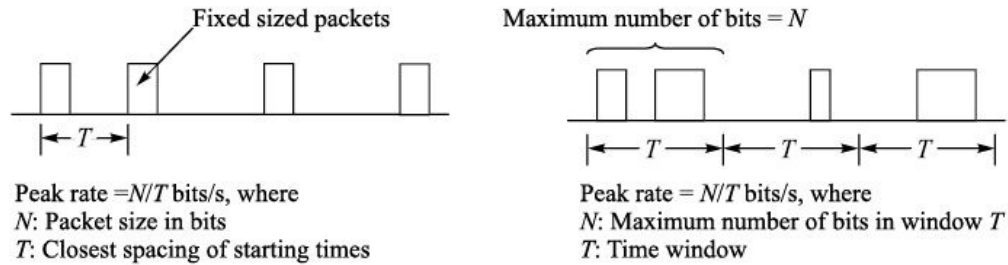


FIGURE 23.10 Peak rate.

Average data rate. Average data rate is defined as the average number of bits that can be sent over the time window. Note that different terminology is used for the same entity in different context. Average rate is same as ‘bandwidth’ introduced in the beginning of this chapter.

Maximum burst size. Maximum burst size specifies maximum number of bits (or fixed size packets) that can be continuously sent at the peak rate.

23.5.2 Basic Traffic Control Functions

For controlling the data traffic that enters a network, the following functions are implemented in the source and the ingress network node: – Admission control.

- Traffic shaping.
- Traffic policing.

Admission control. Admission control was originally developed for virtual circuit packet switching networks. It is called connection admission control (CAC) in ATM. It is used for determining whether the network resources (buffers, bandwidth) required for a new connection are available along the connection path in the network. If resources are available, the new connection is admitted, else the connection is denied.

For connectionless networks such as IP network, admission control function makes little sense as such. There is no connection establishment phase, and the packets of the same flow can take different paths. Admission control makes sense only if the packets of a flow follow the same path. We will learn resource reservation protocol (RSVP) later in this chapter. RSVP is one of the ways of implementing QOS in an IP network. It is used for establishing a path for a flow and for reserving network resources along the path. Admission control function is used by RSVP for determining availability of the required resources in the IP routers along the path from the source to the destination.

Traffic shaping. The source and the receiver of data traffic ensure that the

traffic does not exceed the defined limits by implementing the traffic shaping and policing functions. These functions are implemented in the routers at the edges of the networks (Figure 23.11). *Traffic shaping* refers to the process of altering the traffic characteristics to ensure that the traffic is within the bounds (average data rate, peak data rate, burst size). Traffic shaping function is implemented in the traffic sending router.

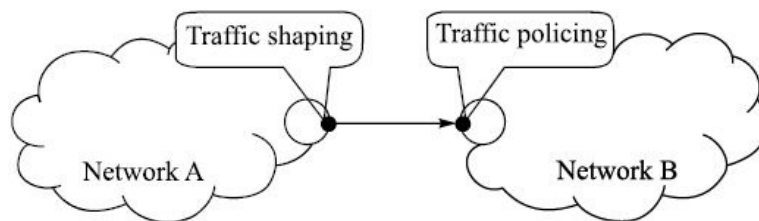


FIGURE 23.11 Traffic shaping and policing functions.

Traffic policing. Policing function is implemented in the traffic receiving router. It monitors the average data rate, and maximum burst size. Policing of the peak rate can also be done if required. The policer either discards or marks the nonconforming packets, the packets that exceed the specified rate and burst limits. Marking essentially raises the drop precedence of these packets. They are carried along the network so long as the network has sufficient available resources. Recall the similar mechanism in frame relay where *DE* bit was set in those frames that exceeded the rate limit and these frames were dropped first whenever congestion occurred.

Traffic policing and shaping functions can be implemented in number of ways. Leaky bucket and token bucket are two algorithms commonly used for implementing these functions.

23.6 LEAKY BUCKET ALGORITHM

As the name suggests, this algorithm uses the analogy of a leaky bucket that has a hole in its bottom. A tap continuously pours water in the bucket. Let us examine basic characteristics of the leaky bucket:

- If the rate at which the water is poured into the bucket is less than the rate at which it gets drained at the bottom, the bucket is always empty.
- If the rate at which the water is poured into the bucket is equal to the rate at which it gets drained at the bottom, the water level in the bucket does not

change.

- If the water is poured into the bucket at higher rate than the drainage rate, then the bucket soon gets filled up to the brim and water starts overflowing.
- If the water is poured into the bucket in spurts, it is still drained at constant rate. The bucket, thus, takes care of momentary spurts of water inflow and smoothens the outflow.
- The size of the bucket and its drainage rate are the two parameters that determine the limits of rate of water inflow and the maximum size of its sudden bursts.

23.6.1 Shaping Using Leaky Bucket Regulator

Leaky bucket shaper is simply a finite queue buffer, drained at fixed rate R (Figure 23.12). When a packet arrives it is appended to the queue. Bursts of data generated by an application are smoothed by the queue to a fixed data rate determined by the queue drainage rate R .

The leaky bucket shaper is very restrictive. It does not allow the bursts to pass through even though the following network may permit some burstiness in the traffic. Token bucket algorithm which we describe later overcomes this limitation.

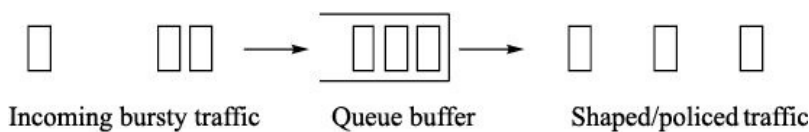


Figure 23.12 Leaky bucket shaper/policer.

23.6.2 Policing Using Leaky Bucket Regulator

When used as policer, the leaky bucket regulator can provide check on the data rate and burst size. It can be implemented using a queue buffer as shown in Figure 23.12. The queue is drained at data rate R . If the size of queue buffer is b bits and the data rate at the input is R_{in} , the maximum size B of data burst at the input is limited to $B = b \times R_{in}/(R_{in} - R)$. The leaky bucket model of the policer checks two parameters, the data rate and the size of maximum burst. If there is need to police the peak data rate also, we can deploy two policers in tandem as shown in Figure 23.13. The first policer checks peak data rate and the second policer is configured for average rate and the maximum burst size.

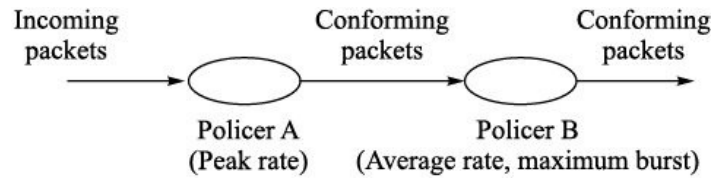


Figure 23.13 Peak rate policer in tandem with average rate limiter.

23.7 TOKEN BUCKET ALGORITHM

Token bucket mechanism is similar to leaky bucket mechanism. The bucket in this case holds tokens instead of data packets (Figure 23.14). The data packets are queued in a separate buffer. A token is an authorization to send one bit of data. We study traffic shaper based on token bucket first. Traffic policer based on token bucket is simple extension of the same. The basic features of token bucket algorithm are as under:

- Tokens are generated at constant rate equal to the average data rate and stored in the token bucket. If the bucket gets full, the arriving tokens are discarded.
- For sending a data packet, of size s bits, s tokens are withdrawn from the bucket. If the bucket is empty, or contains insufficient tokens, the data packet has to wait in the buffer.
- Data packets can be sent at the peak rate so long as there are tokens in the bucket.
- Token bucket has a fixed size. Bucket size roughly determines the maximum data burst size.
- Since a token is always required to send one octet of data packet, the average data rate is same as the average rate of adding tokens in the bucket.

Note that the token bucket algorithm allows sending a burst of data packets when tokens get accumulated in the bucket. Recall that the leaky bucket mechanism when used for traffic shaping function had the limitation of not allowing burst of data packets.

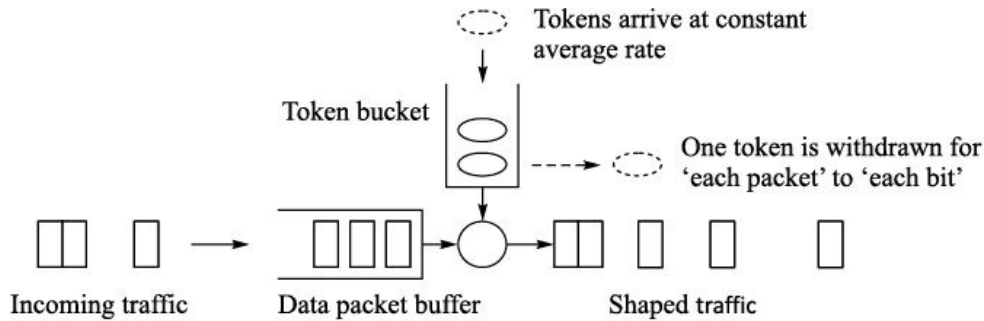


FIGURE 23.14 Token bucket shaper.

23.7.1 Maximum Burst Size

To calculate the maximum burst size for a given token rate and token bucket size, let us assume that the peak rate at which the packets are sent is r_{\max} bits/s size of the token bucket size is b bits and the average arrival rate of tokens is R bits/s. The maximum burst is generated when the token bucket is full initially and sender continues sending bits at the peak rate for the duration t . The tokens added to the bucket in time t need also be taken into account. Thus maximum burst size $r_{\max}t$ is given by: $r_{\max}t = b + Rt$ Token bucket can also be configured as peak rate regulator by keeping the token bucket size equal to zero and generating the tokens at the peak rate. Thus by having two traffic shapers in tandem, one for regulating the average rate and maximum burst size and the second for regulating the peak rate, all the three parameters of the traffic can be regulated.

Token bucket regulator becomes policer by keeping the data buffer size equal to one packet. If there are sufficient tokens in the bucket when a data packet arrives, it is declared as conforming packet. If the token bucket is empty, the arriving data packet is declared as nonconforming.

EXAMPLE 23.3 If the size of token bucket is 100 k octets, the average rate of arrival of the tokens is 1 M/s, and the packets are sent at the peak rate of 11 M octets/s, calculate the maximum burst size. Assume each token has authorization to send one octet of data.

Solution

$b = 100000$ octets, $r = 1000000$ octets/s, $R = 11000000$ octets/s The maximum burst duration t is given by $t = b/(R - r) = 0.01$ s. Therefore the maximum burst size is $0.01 \cdot 11000000 = 110k$ octets.

The leaky and token bucket regulators can be implemented using a counter instead of queue buffer. This implementation is left as an exercise.

23.7.2 Queuing Delay of Shaped Traffic

As we discussed in the beginning of the chapter, the end-to-end delay experienced by a packet across a network consist of three main components—serialization delay, propagation delay and queuing delay. The first two components of the delay are constant, the queuing delay is variable and can be managed by proper allocation of network resources (e.g. bandwidth allocated to a flow). It may not be seemingly obvious, but there is an upper bound on the end-to-end queuing delay that a shaped flow will experience across a network. Knowing the maximum queuing delay margin available for making an end-to-end delay commitment, we can calculate the required network resources.

Let us assume that the token bucket of size b bits and token rate r bits/s for a flow in the edge router A (Figure 23.15). The packets after passing through the token bucket regulator are put in the output queue. The queue has allocation of bandwidth R which is greater than the token rate r . The allocation of bandwidth R can be assured using, for example, WRR queuing. Let us also assume that the queue buffer is greater than bucket size b and the queue is empty at $t = 0$.

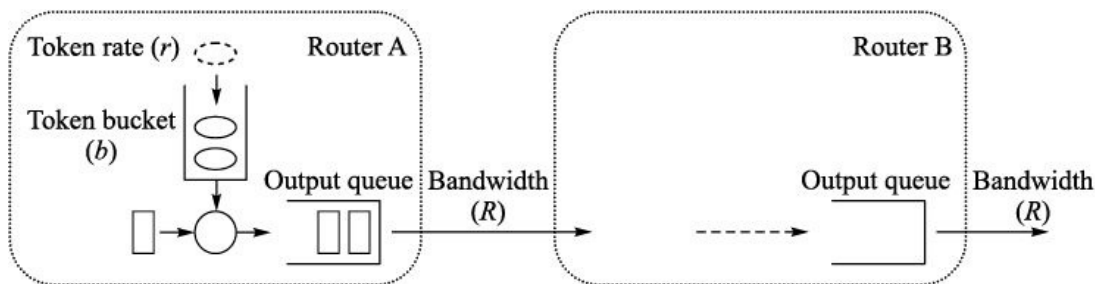


FIGURE 23.15 Queuing delay for shaped traffic.

- Since the queue is drained at the rate R greater than the input rate r , the queue will always be empty unless there is a burst.
- The token bucket regulator can put maximum of b bits in the output queue at a time. Once b bits are put in the queue, the queue depth will become b and it will reduce at the rate $R - r$ thereafter. In other words, the maximum depth of the queue is limited to b .
- The queuing delay of a bit depends on its position in the queue. If it is x bits away from the head of the queue, it will suffer a delay of x/R seconds. Therefore the maximum delay in the queue can be b/R in router A.
- As regards router B, it receives the flow at peak rate of R and dispatches it at the same rate. Therefore its output queue is always empty and there is no queuing delay in router B.

- If all routers along the path taken by the flow have queues with bandwidth at least R , the end-to-end queuing delay is bounded to b/R .

The above model assumes that the flow is handled as stream of bits not packets of varying sizes. It can be shown that when weighted fair queuing is used, the maximum delay d experienced by a flow that is shaped by token bucket of capacity b and token rate r , is given by:
$$d \leq \frac{b}{R} + \frac{(H-1)m}{R} + \sum_1^H \frac{M}{R_j}$$

where, m is the maximum packet size of the flow, M is the maximum packet size in the network, H is the number of hops, R_j is the speed of link j .

Recall that other delays, *viz.* propagation and serialization delays, are constant for given end points and bit rates of physical links. It is the queuing delay that can be controlled by allocating the required network resources. The above result forms the basis for guaranteed upper bound on end-to-end delay for IP networks and can be used for computing the required network resources for a given flow with defined characteristics (average rate, peak rate and burst size).

23.8 QUEUE BUFFER MANAGEMENT

Queue buffer management controls the occupancy of buffer allotted for a queue. If the queue buffer is not properly managed, buffer can get filled to its capacity and thereafter the incoming packets are lost. This can result in fluctuating traffic as we shall shortly see. There can be two approaches to queue buffer management:

- Drop the packets when the depth of a queue exceeds a defined threshold.
- Send notification to the end systems that congestion is building up.

The second approach was used in frame relay using FECN/BECN bits. We will examine this approach in the next section. The first approach involves definition of a packet drop policy that determines when and which packets are to be dropped. There are two basic approaches to packet drop policy:

- Tail drop.
- Random early detection (RED).

23.8.1 Tail Drop

Dropping of packets when the queue is full is called *tail dropping*. It is simple to implement strategy. But it has some very undesirable features for TCP flows. When IP packets containing TCP segments are dropped, the TCP sender does not receive the acknowledgement for the dropped segment. Recall from Section 20.8 of Chapter 20, that the TCP sender goes into slow start mode after expiry of retransmission timer. The incoming packets to the queue may be from several TCP senders and all the packets irrespective of the source get dropped. Therefore all the TCP senders go into slow start mode simultaneously. This global synchronization of slow start mode results in the incoming traffic to the queue to drop suddenly (Figure 23.16). Thereafter all the TCP senders simultaneously build up their traffic exponentially. The exponential build-up of the aggregate traffic again exhausts the queue-buffer and leads to tail dropping. As a result, – the link utilization fluctuates between highs and lows, and the link is not optimally utilized, – the throughput is inconsistent and there is periodic loss of packets. This could cause problems for applications and users.

The UDP traffic, unlike TCP traffic, is not impacted by the tail drop because UDP does not retransmit the lost segments.

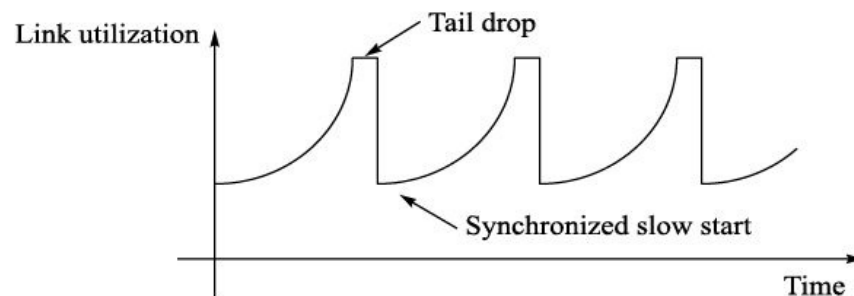


FIGURE 23.16 Synchronized slow start due to tail dropping.

23.8.2 Random Early Detection (RED)

Unlike tail drop, which provides no queue management, RED is an active queue management technique. Its name ‘random early detection’ describes the algorithm: – The decision to drop some packets is made early, before the queue is full.

- After the decision to drop some packets has been made, RED picks the packets to be dropped randomly. Random selection of packets ensures that packets of a specific flow only are not selected.

By randomly dropping a packet, a router sends an implicit indication to the

TCP sender that the dropped segment experienced congestion at some point in the network. Recall that TCP sender detects missing segment on receipt of three duplicate acknowledgements, and retransmits the segment. It also reduces its congestion window (cwnd) and thus the transmission rate.

RED algorithm is based on three parameters:

- Average queue depth.
- Queue depth thresholds.
- Drop profile.

Average queue depth. RED drop policy is based on average queue depth instead of instantaneous value of queue depth. This ensures that sudden spikes in queue depth due to packet bursts do not result in dropping of packets. Average queue depth is computed using the following formula: $A_i = a A_{i-1} + (1 - a) D_i$ where, A_i new average queue depth, A_{i-1} previous average queue depth, D_i is current queue depth and a is a parameter that determines the weight to be given to past average. Its value is less than 1.

Queue depth thresholds. Two thresholds for queue-depth are defined, minimum-threshold T_{\min} and maximum-threshold T_{\max} (Figure 23.17). So long as the average queue depth is below the T_{\min} , no packet is ever dropped. When the average queue depth is above T_{\max} , all the arriving packets are dropped. The minimum threshold is kept large enough so that dropping of the packets is not initiated too early. If $T_{\min} = T_{\max}$, RED degenerates to tail drop. Therefore the gap between the two thresholds is kept large enough to avoid phenomenon of global synchronization of slow start mode.

Drop profile. Drop profile determines the percentage of arriving packets to be dropped when the average depth of a queue is between the two thresholds T_{\min} and T_{\max} . Figure 23.17 shows an example of drop profile function F . It increases linearly as the average queue depth grows. For example, its value is 0.1 when the average queue depth is 50% in Figure 23.17. This implies that on average 1 packet out of 10 packets incoming in the queue will be dropped

The drop profile function F can be written as:
$$F = F_{\max} \frac{A - T_{\min}}{T_{\max} - T_{\min}}$$

where, A is average queue depth. Thus for a given value of F , one packet out of

$1/F$ arriving packets will be dropped.

RED is implemented using counter whose value is used to determine the probability of dropping an incoming packet. The counter counts the number of packets put in the queue since the last packet was dropped. If the counter value is C at any point of time, one packet out of remaining $(1/F) - C$ packets will be dropped. Thus probability (p) of dropping a packet when counter is at C is:

$$p = \frac{1}{\frac{1}{F} - C} = \frac{F}{1 - FC}$$

Note that when the value of C reaches $(1/F - 1)$, the probability p of dropping the next packet becomes 1. Thus out of $1/F$ arriving packets, one packet will certainly be dropped.

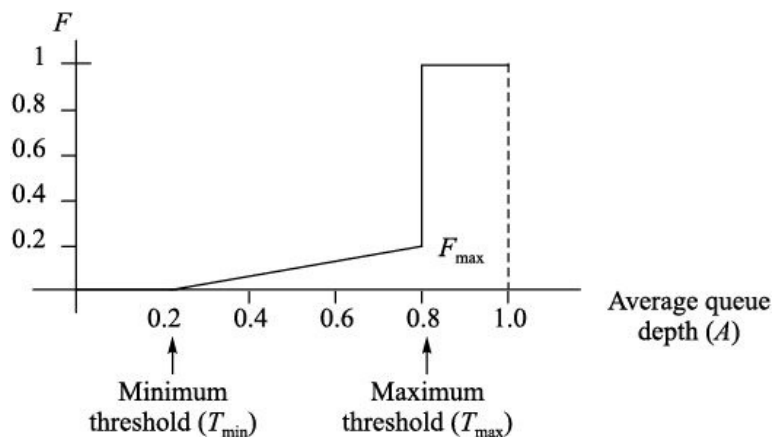


FIGURE 23.17 Drop profile in RED.

Unlike tail drop, RED is complex to configure for achieving the desired performance. Its benefits are as under:

- The global synchronization of TCP sessions to slow-start mode is avoided.
- RED detects early stages of buffer congestion and responds by randomly dropping packets. This action sends implicit indication of congestion proactively to the end stations and thus avoids building up of network congestion.
- Since the average queue depth is maintained at below the maximum threshold, RED allows random bursts of traffic without discarding all the packets.
- By adjusting the parameters of the drop profile, the average queue depth can be maintained at a level that results in the best link utilization of the

output interface.

- RED distributes packet dropping fairly across multiple flows. The flow that shares larger bandwidth is likely to suffer larger number of packet drops.

RED is not used for UDP flows as UDP does not take any action for congestion control when a packet is dropped.

23.8.3 Weighted RED (WRED)

Weighted RED is an extension of RED that allows assignment of different drop profiles to different types of traffic in a queue. Figure 23.18 shows two drop profiles, one for the premium service traffic and the other for the standard service traffic. The drop profile for the premium service traffic is less aggressive than the drop profile for the standard service traffic. Packets belonging to premium and standard services are identified by the DSCP field² in IP header. WRED calculates the average depth of the queue ignoring the traffic type and then applies the respective drop profiles to the premium and standard traffic. For example, when the average depth of the queue is 0.6, the standard traffic has $F = 0.5$, *i.e.* 5 packets out of 10 packets of standard traffic are dropped. The value of F for premium service packets is 0.1 at this queue depth. Therefore only 1 packet out of 10 packets of premium traffic is dropped.

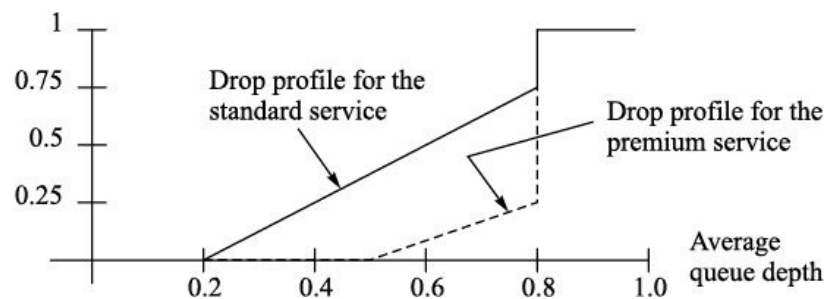


FIGURE 23.18 Drop profiles in weighted RED.

23.9 EXPLICIT CONGESTION NOTIFICATION (ECN)

Congestion occurs when an output interface receives a packet to transmit while the interface is busy in sending another packet. Therefore we introduced queuing system so that the packets can wait for their turn in a queue. To ensure that the average depth of a queue remains within defined thresholds, we introduced RED as the next step. RED merely sends an *implicit* indication to the end systems that congestion is building up in the network. The end stations respond to this indication by reducing the rate of pumping the traffic in the network. As an

alternative, the indication of network congestion can also be given by marking the packets instead of dropping the packets. On receipt of these marked packets, the end system can take “Suitable action for reducing the data rate”. Explicit congestion notification (ECN) scheme for congestion avoidance is based on this philosophy and is implemented in the TCP/IP layers. It is documented in RFC 2481. Its implementation requires some enhancements at the IP and TCP layers.

23.9.1 Enhancements at IP Layer for ECN

ECN requires definition of a two-bit field in the IP header. The two reserved bits of DS field are designated as ECN field (Figure 23.19). These bits are used as under: **ECN capable transport (ECT) bit:** It is set by the source end system to indicate to the routers that its IP/TCP layers are ECN-capable.

Congestion-experienced (CE) bit: It is set by a router when it experiences congestion. It is used only if the ECT bit is set to 1, *i.e.* the end systems are ECN capable.

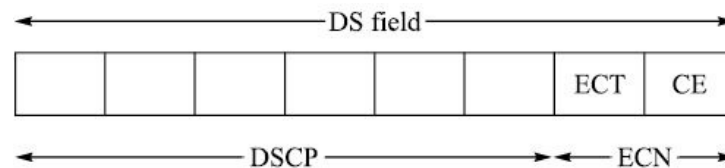


FIGURE 23.19 ECN field.

It is to be noted that routers use RED or similar algorithm for determining whether to set the CE-bit. The CE-bit is set in a packet by a router only if the router would have otherwise dropped the packet using RED algorithm.

23.9.2 Enhancements at TCP Layer for ECN

RFC 2481 addresses the enhancements required in TCP for supporting ECN capability. In general, TCP entity is required to: – determine that if the end-points are ECN-capable,

– react to the receipt of CE-bit at IP layer.

The following three enhancements are required in TCP for supporting ECN:

- During TCP connection set up, a mechanism is required for determining if the communicating TCP entities are ECN capable and want to use their ECN capability.
- A mechanism is required for communicating back to the source TCP entity that CE bit set to 1 has been received at the other end so that the source

TCP entity may take appropriate necessary action for congestion control (e.g. reducing size of the congestion window, cwnd).

- A mechanism for the source TCP entity to inform the other end that it has taken the appropriate necessary action, *i.e.* it has reduced size of the congestion window.

These mechanisms are implemented by introducing two additional flags in the reserved field of TCP header (Figure 23.20).



FIGURE 23.20 TCP enhancements for ECN.

ECN-echo flag. ECN-echo flag in a TCP segment is set by the TCP entity that receives the CE bit from the network through its IP layer. On receipt of this flag, the source TCP entity is required to reduce its congestion window.

Congestion window reduced (CWR) flag. It is set by the source TCP entity after it reduces its congestion window on receipt of ECN-echo flag.

The use of these bits during TCP connection set phase and during data transfer phase is described below.

23.9.3 ECN Operation

We describe ECN operation in two parts:

- ECN capability negotiation during connection set up phase.
- ECN operation during data transfer.

ECN capability negotiation. TCP entities negotiate ECN-capability during TCP connection set up phase (Figure 23.21).

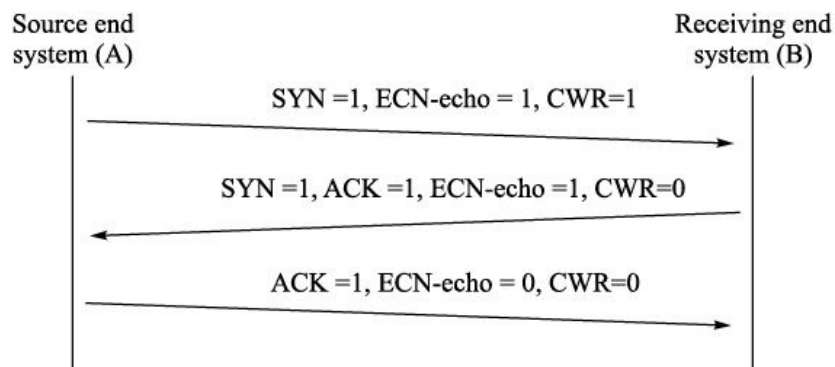


FIGURE 23.21 ECN capability negotiation

FIGURE 23.21 ECN capability negotiation.

- To establish a TCP connection, the source A sends a TCP segment with SYN bit set to 1. It sets CWR and ECN-echo flags to 1 in this segment, indicating to the receiver B at the other end that it wants this connection to be ECN-capable and it will participate both as ECN sender and receiver.
- The receiving TCP entity B responds with usual acknowledgement segment with SYN and ACK bits set to 1. B sets ECN-echo flag to 1 and CWR flag to 0 in the response.
- CWR flag is changed to 0 in the acknowledgement because some of the existing TCP implementations reflect all the reserved bits in the acknowledgement. Such reflection from a non-ECN capable TCP entity will make the source TCP entity to draw wrong conclusion if the CWR flag is kept same in the TCP connection request and acknowledgement.
- As part of three-way handshake for connection establishment, A responds with ACK bit set to 1. The ECN-echo and CWR flags are kept at 0 in this TCP segment.

ECN during data transfer. Figures 23.22 and 23.23 show how the explicit congestion notification works at the IP and TCP layers. A and B are ECN-enabled end systems.

- A sets ECT bit in the IP header to 1 indicating to the routers in the network that it is ECN-enabled.
- There is congestion in router R2 en route when an IP packet travels from A to B. RED algorithm in R2 sets the CE bit of the IP header to 1 (Figure 23.22).

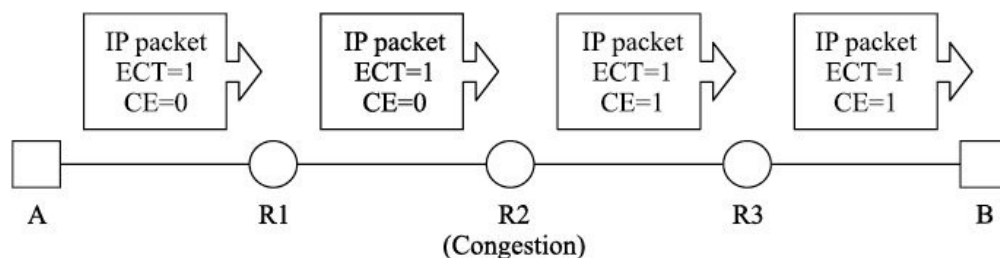


FIGURE 23.22 ECN operation at IP layer during data transfer phase.

- When the IP layer of destination B receives the IP packet, it indicates to the TCP layer that CE bit set to 1 has been received.

- When the TCP layer at B receives indication of CE bit set to 1 from its IP layer, it sets ECN-echo flag in the TCP header of the acknowledgement it sends to A (Figure 23.23).
- ECN-echo flag is repeated in all the subsequent acknowledgements till confirmation from A of having reduced cwnd is received. This is done as protection against loss of the acknowledgement.
- When the TCP layer of A receives a TCP acknowledgement with ECN-echo flag set to 1, it immediately reduces its cwnd and sets CWR flag to 1 in its next data TCP segment sent to B, confirming thereby to B that it has taken the required congestion control action.

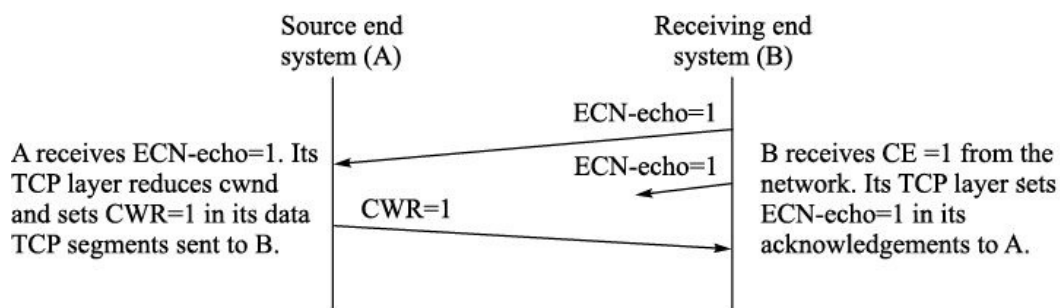


Figure 23.23 ECN operation at TCP layer.

A may receive subsequently multiple ECN-echo flags in following acknowledgements, but it reduces cwnd only once. After sending CWR flag, it waits for the time equal to RTT. It acts on the next ECN-echo flag received after RTT from B assuming the network is again reporting congestion to B.

23.9.4 Benefits and Limitations of ECN

Benefits of ECN when compared to tail drop are same as those of RED. ECN has the following benefits in addition:

- When a packet is dropped due to congestion in the network, the fast recovery mechanism of TCP would wait for three duplicate acknowledgements before reducing the cwnd and retransmitting the lost segment. When ECN is enabled, there is no packet drop and therefore no TCP segment is required to be retransmitted. Further, cwnd is reduced immediately in receipt of ECN-echo flag.
- ECN can be deployed incrementally without having partition the network into islands of ECN-capable and non-ECN-capable nodes.
- ECN does not generate additional traffic for congestion control.

The basic limitation of ECN is that it requires upgrade TCP and IP protocols in the deployed network nodes and hosts.

23.10 FRAMEWORKS FOR IMPLEMENTING QOS

For implementing QOS in IP networks, we need the QOS tools discussed above. Each of these tools builds certain capability in the network. We need a framework that uses these tools to build QOS capability in an IP network. There are two major frameworks for providing QOS on an IP network. These frameworks differ in their approach to quality of service provision.

- Integrated Service.
- Differentiated Service

Integrated service (IntServ). Integrated service framework for QOS is based on the philosophy that the routers need to reserve resources to provide minimum *guaranteed* end-to-end quality of service to a flow. End systems indicate their QOS needs to the network for every flow before its commencement. The routers reserve the required network resources for the flow.

Differentiated service (DiffServ). Differentiated service does not reserve resources for individual flows in advance. It is based on providing several levels of 'best effort' service. For example, a network operator may offer Platinum, Gold, Silver and Bronze services. These services differ in offerings in terms of delay, jitter, packet loss, *etc.* The network is pre-provisioned for offering these services. A customer has the choice of selection of the service he needs for his application.

Integrated and differentiated service frameworks are not mutually exclusive from the point of view of implementation.

23.11 INTEGRATED SERVICE

Integrated service framework for quality service was designed to guarantee predictable IP network behaviour without modifying the basic architecture of the

Internet. It is documented in RFC 1633. Integrated service framework is based on reservation of network resources required for a flow of packets from the sender to the receiver. Resource reservation means that the routers will ensure that a session that has made reservation gets the required bandwidth and queuing priorities, and it is protected from ill effects of congestion caused by other flows.

The request for reservation is sent by the end systems and accepted by the network prior to commencement of the flow. It is possible that the network does not have adequate resources to support the flow, in which case the request for resource reservation is turned down.

23.11.1 Classes of Integrated Service

Integrated service framework has the following two classes of service:

- Guaranteed QOS class.
- Controlled load QOS class.

Guaranteed QOS class. Guaranteed QOS class provides assured bandwidth and a firm upper bound on end-to-end delay for all the conforming packets of a flow. Upper bound on end-to-end delay is achieved by controlling the queuing delay. The end systems specify their QOS requirements and if the network can meet these requirements, it gives its acceptance. Guaranteed QOS class is intended for intolerant real-time applications.

Controlled load QOS class. Unlike guaranteed QOS class, controlled load QOS class does not provide quantitative performance guarantees. The network makes a commitment to offer 'best effort' service equivalent to that of a lightly loaded network, even though the network as a whole may be heavily loaded. In other words, the controlled load class is better than the usual 'best effort' service in the sense that its performance does not deteriorate noticeably when the network load increases. As in guaranteed QOS class, the end systems can specify the service parameters and the network accepts those parameters.

If a flow exceeds the accepted limits of the traffic conditioning agreement, the nonconforming packets are forwarded as best effort datagrams in the above classes of service.

23.11.2 Tspec and Rspec

The QOS requirements are specified as:

- Tspec (Traffic specifications).
- Rspec (Reservation specifications).

Tspec. Tspec includes the following parameters: – Token bucket depth (b).

- Token rate (r).
- Peak rate (p).
- Maximum datagram size (M).
- Minimum policed unit size (m).

Minimum policed unit size (m) specifies the granularity of the flow. All packets smaller than the minimum policed unit size m are counted as having size m .

Rspec. Rspec specifies two parameters, bandwidth R and a slack term S . Slack term S is the upper bound of end-to-end queuing delay. We will study the use of this parameter later in the section.

Guaranteed service is specified using Tspec and Rspec. Controlled load service is specified using only Tspec. There is no Rspec.

23.11.3 Components of Integrated Service

To implement integrated service framework for quality of service, we need to augment the capability of the routers and the end systems for requesting resource reservations and for implementing resource reservations in the network. Once the resource reservations have been made, the flow of IP packets containing user data takes place in the usual manner. For the resource reservation, the following components are required:

- Resource reservation protocol (RSVP).
- Policy control.
- Admission control.
- Packet classifier and scheduler.
- Traffic shaping and policing.

Resource reservation protocol (RSVP). It is used by the end systems to send request for the required network resources, and by the RSVP enabled routers to reserve the network resources.

Policy control. Policy control function authenticates the user who requests for resource reservation and determines whether the user has the rights to make

reservation.

Admission control. Admission control function is required to ensure that the requests for resource reservations are accepted only if sufficient network resources are available.

Packet classifier and scheduler. Packet classifier in an RSVP router segregates the flows for which resource (buffer and bandwidth) reservation has been made. Packets of these flows are buffered in the queues which are served by the scheduler. The scheduler ensures that the reserved bandwidth is made available to these queues.

Traffic shaping and policing. Traffic shaping and policing functions ensure that a flow does not exceed its traffic specifications, based on which the resource reservations were made.

23.12 RESOURCE RESERVATION PROTOCOL (RSVP)

Resource reservation protocol (RSVP) is a signalling protocol that is used to set up resource reservations across an IP network. It is documented in RFC 2205. Basic features of RSVP are as under:

- RSVP is used only for reservation of network resources and for their release. Routing and forwarding functions in the IP network are carried out as before. There is no change in the IP packets carrying user data.
- Resource reservations can be made for a TCP or UDP flow. A flow is called a *session* in RSVP.
- RSVP can make resource reservations for both unicast and multicast applications.
- RSVP reservations are unidirectional. Therefore separate requests and reservations are made for each direction of flow of a bidirectional session.
- The routers that do not support RSVP, forward the RSVP packets transparently.
- RSVP creates 'soft reservation state' in the RSVP enabled routers. This implies that the reservation state needs to be refreshed periodically by repeating the messages for reserving the resources.
- RSVP is supported by IPv4 and IPv6.

23.12.1 RSVP Messages

RSVP uses the following two messages for making resource reservations in the RSVP enabled routers:

- PATH message.
- RESV request message.

There are five additional optional messages:

- PATH error
- RESV error
- PATH tear-down
- RESV tear-down
- RESV confirm

PATH message. PATH message is originated by the sender that wants to reserve resources for a flow in the onward direction towards the receiver (Figure 23.24). The PATH message is addressed to the receiver. It serves the following basic purposes:

- It establishes the reverse routing states in each RSVP router along the path from the sender to the receiver for the intended session. Reverse routing state enables sending the RESV message along the same path but in reversed direction.
- It indicates to the receiver and the intermediate RSVP enabled routers the characteristics of the intended traffic flow from the sender to the receiver.
- The intermediate RSVP routers that forward the PATH message from the sender can insert their resources availability and capability in the PATH message for the information of the receiver.

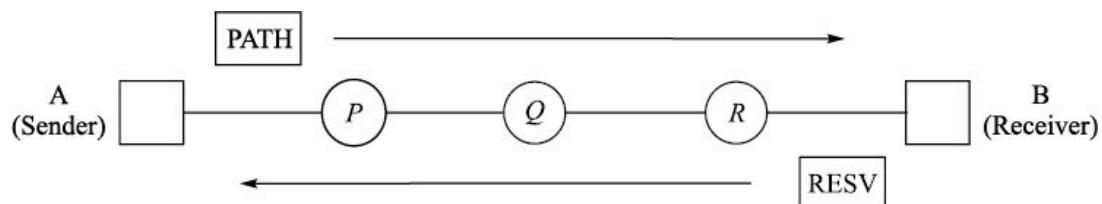


FIGURE 23.24 Flow of PATH and RESV messages of RSVP.

RESV request message. RESV request message is the request to RSVP enabled routers to reserve the network resources indicated in the message. It is generated

by the receiver on receipt of a PATH message (Figure 23.24). The quantum of required resources is decided by the receiver based on the traffic characteristics of the flow and the available network resources/capabilities as specified in the PATH message. The RESV request message retraces the path taken by the PATH message. Path retrace is facilitated by the reverse routing states created by the PATH message. The five optional messages are used as under:

- PATH tear-down message is sent by the sender to remove the reservation from the network.
- RESV tear-down message is sent by the receiver to remove the reservations from the network.
- PATH error message is sent by an RSVP router or the receiver when an error (format error or bit error) is noticed in the PATH message.
- RESV error message is sent by an RSVP router or the sender when an error (format error or bit error) is noticed in the RESV message. RESV error message is also generated when an RSVP router does not have resources to make reservations as requested in RESV message.
- RESV confirm message is sent by the last RSVP router to confirm back to the receiver that the requested reservations have been made.

23.12.2 Format of RSVP Messages

The RSVP messages consist of a common header and body containing several objects. The general format of RSVP messages is shown in Figure 23.25.

Version. This field contains RSVP version. The current version is 2.

Flags. Flags have not been defined yet.

Message type. It indicates the type of RSVP message.

1. PATH
2. RESV request
3. PATH error
4. RESV request error
5. PATH tear-down
6. RESV tear-down
7. RESV confirm

Checksum. 16-bit standard TCP/UDP checksum is used for detecting error in the RSVP message.

Send TTL. It contains the TTL value of the IP packet in which the RSVP message is sent.

RSVP length. It contains the length in octets of RSVP message including its body.

Length. It is length of the object in octets. It must be multiple of four.

Class-number. It identifies the object-class. Some examples of the object classes are Tspec, Flowspec, Adspec, session identifier, time value of refresh interval, RSVP hop (IP address of the last RSVP router).

Class-type. It specifies the type of object within the class-number.

Object contents. It contains the parameter values associated with the class type.

RSVP messages are sent as IP packets by attaching IP header to them. Value of protocol type field in IP packet is 46 for RSVP. Alternatively, an RSVP message is first encapsulated as UDP segment, which is then attached to the IP header. The UDP ports for RSVP messages are 1698 and 1699.

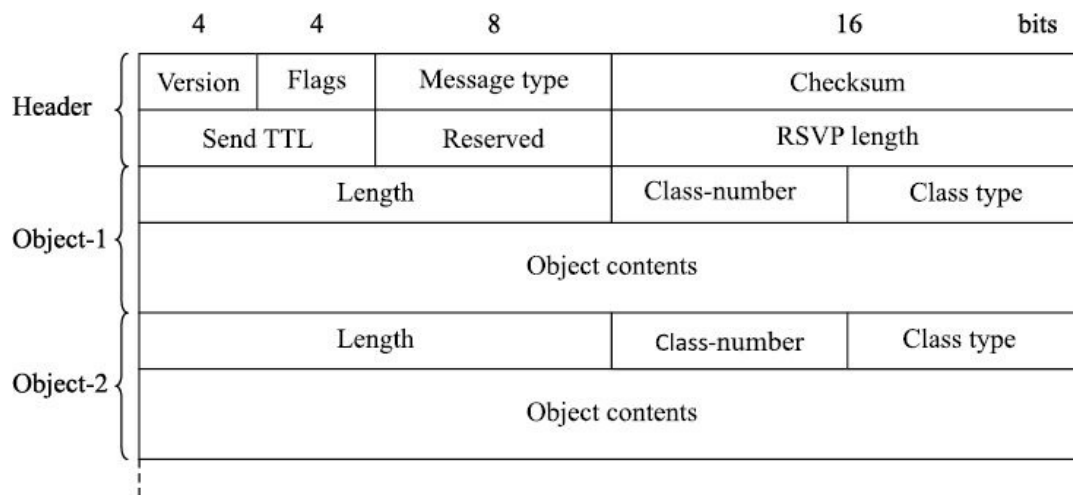


FIGURE 23.25 Format of RSVP messages.

23.12.3 Basic Operation of RSVP

The basic operation of RSVP consists of three phases:

- Establishing resource reservations
- Maintaining resource reservations

- Releasing resource reservations.

Establishing resource reservations. Figure 23.26 illustrates how the resource reservation is made for a flow from A to B.

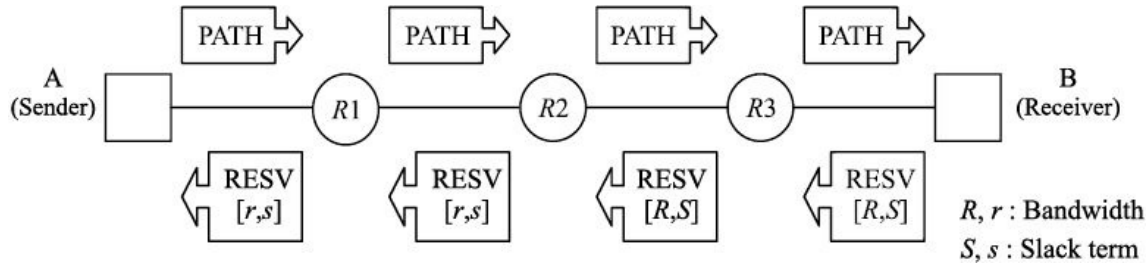


FIGURE 23.26 Establishing resource reservations.

- PATH message is sent as an IP packet by the sender A. The PATH message is routed through the network like normal IP packet using the forwarding tables created by the routing protocols (e.g. OSPF, BGP). The PATH message contains the following key information:
 - *Tspec* specifies the characteristics of the traffic the sender intends to send.
 - The *session identifier* consists of destination IP address, destination TCP/UDP port number and protocol type.
 - It is the *IP address* of the interface through which the PATH message is sent. This IP address is used by the upstream RSVP router later to return the RESV request message from the receiver.
- When an RSVP enabled router receives the PATH message:
 - it takes note of the *Tspec* and the IP address of the last hop through which the path message was sent. It replaces this IP address with the IP address of its interface through which it intends to forward the PATH message.
 - it updates *Adspec* field indicating its capability and resource availability. Important constituents of *Adspec* are class of service supported (controlled-load or guaranteed class), cumulative path delay, minimum bandwidth of links along the path, path MTU, hop count, *etc.*
 - it forwards the PATH message to the next hop towards the receiver. The forwarding decision is based in the forwarding table of the router.
- Each RSVP enabled router on the way to the receiver B carries out the above task. Non-RSVP routers simply forward the IP packet containing the PATH message.
- Ultimately, the receiver B receives the PATH message. It takes note of the *Tspec* and *Adspec*. The *Adspec* gives the receiver path information on class

of service supported, the available bandwidth available, cumulative path delay, number of hops and MTU size. The receiver works out the required quantum of network resources. It generates an RESV request message, and sends it to the last RSVP enabled router from which it received the PATH message. The RESV request message contains the following resource reservation information:

- The required resources are specified as *Flowspec* in the RESV request message. Flowspec includes Tspec, the bandwidth (R) required to be reserved and a slack term (S) for the end-to-end delay. Slack term is the available delay margin if the required bandwidth R is reserved by each router. The routers have flexibility of reserving less than R bandwidth and consuming thereby part of available delay margin.
- *Filterspec* specifies the flow or flows for which reservation is to be made. IP address and TCP/UDP port of the source are specified as Filterspec in the RESV request message for this purpose. This allows aggregation of reservations required for flows of the same application from a source end system.
- The RESV message retraces the path taken by the PATH message. Recall that each RSVP router that forwarded the PATH message, had taken note of the IP address of previous RSVP router from the PATH message. RESV message is forwarded to that IP address.
- When the upstream RSVP router on the way to A receives the RESV request message, it carries out the following tasks:
 - It passes the request to admission control function to find out whether sufficient resources are available to implement the reservation request.
 - It passes the request to policy control function to determine whether the user has rights to make the reservation.
- If the admission control and policy control checks are not successful, the router does not forward the RESV request message any further and generates error message for B.
 - It reserves the required resources if policy control and admission control checks are successful, and it forwards the RESV request message to the next upstream router. It is possible that the available bandwidth is less than the requested bandwidth, in which case, the RSVP router reserves the available bandwidth and works out additional delay on this account. It deducts the additional delay from the slack term and updates the Flowspec to include reduced bandwidth and revised slack term before forwarding the

RESV request message to the next upstream router. In Figure 23.28, the available bandwidth r is less than requested bandwidth R in router Q. It updates Flowspec to $[r,s]$ in the RESV request message before forwarding it to router P.

- The sender A ultimately receives the RESV request message from B and it assumes that the required QOS is in place for the session. It can thereafter start sending the data packets of the session as it would have in a non-RSVP situation. For the flow from B to A, B makes the resource reservation in the same manner by sending a PATH message to A.

Maintaining resource reservations. The resource reservations made in the RSVP routers need to be refreshed periodically. In other words, the sender repeats the PATH messages at defined interval of time and the receiver repeats the corresponding RSVP messages. The RSVP routers refresh the reservation state when they receive the reservation messages. If the reservation message is not received before time out the current resource reservation is revoked. Soft state of reservation is a necessity because of the following reasons: – If the resource reservation is made as hard state, it will necessarily require a tear-down message. If the tear-down message is lost and an RSVP router does not receive the tear-down message of a session, it will continue to reserve the resources for the session indefinitely.

- If the part of network goes down due to any reason, the next PATH message will automatically initiate reservation of resources through alternative path as determined by the routing protocol. Thus the robustness of IP network is maintained.

Releasing resource reservations. Resource reservation release action can be initiated by the sender or by the receiver. The sender uses PATH tear-down message to release the reserved resources. The receiver uses RESV tear-down message to release the reserved resources. Mere stopping of periodic PATH messages or RESV messages also releases the reserved resources of the network.

23.12.4 Resource Reservation for Multicasting

RSVP was designed for reservation of network resources both for unicast and for multicast applications. The PATH messages are sent by the multicast source on the multicast tree (Figure 23.27). The receivers reply with RESV request messages that travel towards the source following the paths up the same tree. As

an RESV request message travels up the tree, it may hit an RSVP router where another receiver has already made reservation for the flow. For example, when the RESV request message from A hits router R2, R2 notices that receiver B has already made resource reservation for the multicast flow. Since the resource reservation for a multicast flow needs to be made only once in a router, R2 does not make any additional reservation and it does not forward the RESV request message from A further up the tree. In general, reservations from the receivers downstream of a node of the multicast tree can be merged.

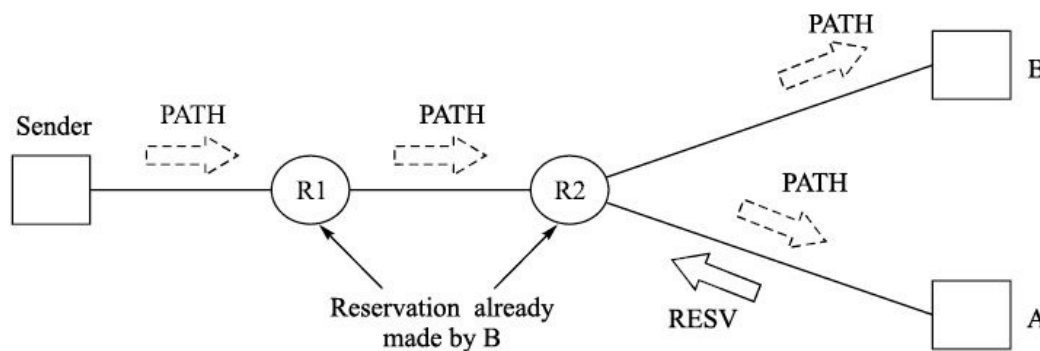


Figure 23.27 Reservation for multicast.

We assumed that the reservation made by B is adequate for the Flowspec given in the RESV request message from A. If it is not so, R2 will revise the reservations accordingly and forward the RESV request message from A up the tree. When a refresh RESV request message is received from B later, R2 need not forward it upwards. Refresh RESV request messages from A will only be forwarded.

23.12.5 Reservation Styles

It is quite possible that a receiver may want to get packets from different senders in one session, if there are be multiple senders. For multiple senders, the required resource reservation may not be aggregate of Tspec indicated by all the senders. For example, in audio conferencing of say five persons, not more than two persons may be allowed to speak. RSVP defines the following ‘reservation styles’ to provide for such situations:

- Fixed-filter (FF) style.
- Shared-explicit (SE) style.
- Wildcard-filter (WF) style.

In fixed-filter style, there is one distinct reservation for each sender. In shared-

explicit reservation style, a single reservation is shared by selected senders. The receiver decides which senders can share the reservation. The RSVP request message from the receiver contains the list of such senders. In wildcard-filter style, there is a single reservation for all the senders participating in the same RSVP session.

23.12.6 Limitations of RSVP

RSVP provides the highest level of IP QOS but it has its limitations. It cannot scale to large networks because of the following reasons:

- The routers need to maintain state information of each session. Number of concurrent sessions can be very high in large networks. CPU and limited memory resources of the routers become constraints for maintaining the state information. Therefore scalability of RSVP to large networks becomes an issue.
- There is continuous flow of RSVP messages to maintain the soft reservation states. This generates additional traffic.
- The end systems need to be RSVP-enabled.

23.13 DIFFERENTIATED SERVICE

Differentiated service framework adopts a different approach to QOS. In this case, the network offers 'best efforts' service with different levels of QOS. For example, a network service provider may grade his service offerings as Platinum, Gold, Silver and Bronze. All these services are best effort, but the IP packets of these different service classes get differential forwarding treatment at the network nodes. Differential forwarding treatment in a router can be implemented using the QOS tools described earlier. For example, the 'Platinum' IP packets can be assigned to the priority queue so that they are always transmitted ahead of packets of other service classes. The drop policies implemented in the routers can be different. The RED drop profile for the 'Bronze' packets can be more aggressive compared to that of 'Silver' packets.

To give differential forwarding treatment, the routers must be enabled to segregate the IP packed based on their class. Differentiated service code point (DSCP) field in the IP header is used for this purpose. DSCP field of an IP packet is set by the network node as the packet enters the network. It can be set

by the user also, if the user is trusted by the network operator.

It is to be noted that there is no flow-based resource reservation in the routers in differentiated service framework. The routers classify the packets based on DSCP field and give them forwarding based on the DSCP field, irrespective of the flow they belong to. All that a network administrator does is to define and implement in each router differential forwarding treatments for the packets belonging to various classes of the service.

23.13.1 Differentiated Service Code Point (DSCP) Field

The type of service (TOS) octet of IPv4 header was intended as some kind of service descriptor so that the routers could extend differential treatment to IP packets based on the various fields in the octet (Figure 23.28). Except for some implementations of the precedence bits, TOS field is not much used in the Internet. In IPv6, this octet appears as ‘Traffic Class’ field. In 1998, RFC 2474 redefined this octet as DS (Differentiated Service) field (Figure 23.28). The last two bits of this DS field were reserved in RFC 2474. Later in 2002, RFC 3260 designated these two bits as explicit congestion notification (ECN) bits. Use of ECN bits for congestion notification was explained in Section 9 of this chapter. The first six bits of DS field are called differentiated service code point (DSCP) and these bits determine the forwarding treatment a packet is entitled to at a network node.

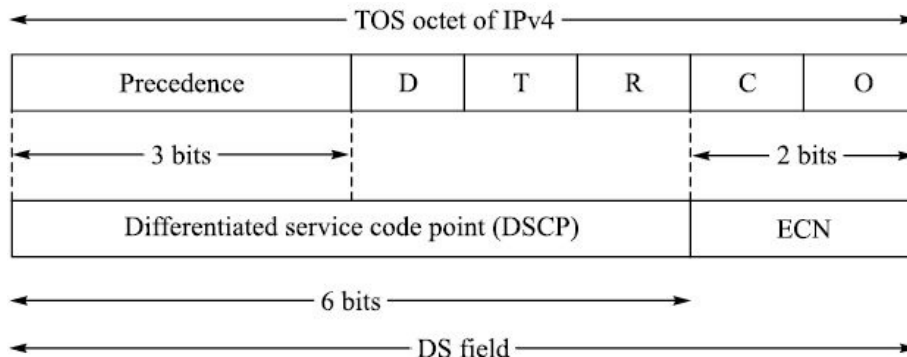


Figure 23.28 DSCP field.

23.13.2 Per-Hop-Behaviour (PHB)

Differentiated service framework introduces the following two new terms:

- Behaviour aggregate (BA)
- Per-hop-behaviour (PHB).

Packets having same value of DSCP field form one *behaviour aggregate*. These packets get same forwarding treatment in a router. *Per-hop-behaviour* (PHB) is externally observable forwarding treatment of a network node for a behaviour aggregate. A PHB can be defined in terms of: – the amount of resources allocated to the PHB (e.g. buffer size, bandwidth).

- the relative priority of the PHB as compared to the other PHBs.
- the traffic characteristics (e.g. delay, jitter and loss).

PHBs are not defined in terms of specific implementation mechanisms used for realizing the above characteristics. Thus there can be different implementations for the same PHB.

Each behaviour aggregate is mapped to a PHB (Figure 23.29). It is possible that same PHB is applied to multiple behaviour aggregates. In other words several DSCP values can be grouped and can be assigned one PHB. Having assigned PHBs to various behaviour aggregates, a network administrator implements in every network node the classification based on DSCP values. For each class, he defines the queue scheduling and drop policy based on the PHB defined for the class.

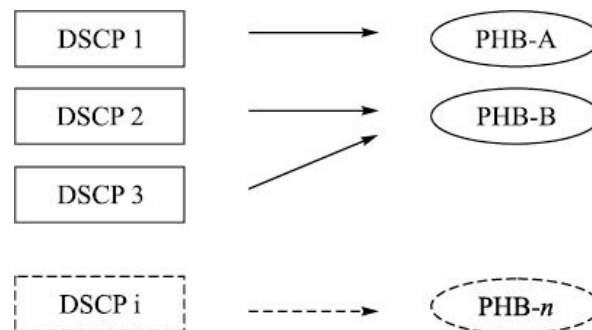


FIGURE 23.29 Mapping of DSCP values to PHBs.

The IETF has defined the following PHBs:

- Expedited forwarding PHB (EF).
- Assured forwarding PHB (AF).
- Class selector PHB.
- Default PHB (DE).

Expedited forwarding PHB (EF). RFC 3246 defines expedited forwarding (EF) PHB as a mechanism for offering premium service having the highest QOS, *i.e.* assured bandwidth with minimal packet loss, jitter and delay. In other words, EF PHB emulates a virtual leased line. EF PHB can be supported in the

nodes by: – assuring that the EF packets have well defined minimum bandwidth at the output interface of a router, – placing packets belonging to EF class in priority queue, or class based weighted fair queuing, and – policing the flows at the edge of the network as per their TCA. This will ensure that aggregate ingress rate of EF packets does not exceed the provisioned bandwidth at various output interfaces in the network.

EF PHB is targeted for applications like VoIP (voice over IP) and video conferencing. These applications are UDP-based. Since UDP does not respond to RED, RED is not used for EF PHB generally. DSCP bits for EF PHB are 101110.

Assured forwarding PHB (AF). Assured forwarding (AF) PHB is defined in RFC 2597. It emulates lightly loaded condition of a network. It ensures that the packets are forwarded with high probability as long as the aggregate traffic of the class does not exceed the TCA.

The AF PHB has four forwarding classes of service. These classes can have different levels of forwarding treatment. The four classes can be, for example, referred to as Platinum, Gold, Silver and Bronze. Platinum class, for example, can be assigned adequate bandwidth to meet the peak bandwidth and burst requirement of the customers. IETF does not prescribe the minimum level of service for any class. It is for the network administrator to define the offered service level of each class. The first three bits of the DSCP field define the AF class (Table 23.2).

Drop Precedence	Class AF1	Class AF2	Class AF3	Class AF4
Low	001 010	010 010	011 010	100 010
Medium	001 100	010 100	011 100	100 100
High	001 110	010 110	011 110	100 110

Each of the four classes has three drop precedence levels—low, medium and high. Packets within a class are marked (either by the customers or by the network) with one of the three drop precedence levels. The last three bits of DSCP field are used for marking the drop precedence (Table 23.2). If there is congestion in the network, the drop precedence of a packet determines the relative importance of the packet within its AF class. The packets with higher drop precedence value are discarded first when congestion occurs. RED is often used for implementing the drop policy. Different drop profiles can be implemented for each drop precedence level. Table 23.3 shows typical RED

drop profiles assignments for the AF classes. The parameters T_{\min} , T_{\max} and F_{\max} are defined in Figure 23.17.

TABLE 23.3 RED drop profiles for the AF classes

DSCP	T_{\min}	T_{\max}	F_{\max}
AF11, AF21, AF31, AF41	32%	40%	10%
AF12, AF22, AF32, AF42	28%	40%	10%
AF13, AF23, AF33, AF43	24%	40%	10%

The AF PHBs can be supported in a router by:

- having four different queues, one for each class,
- assigning adequate resources (buffer and bandwidth) to each queue, – configuring RED to honour the drop precedence bits, and
- by policing AF flows at the edges.

Class selector PHB. Class selector PHB type was defined for backward compatibility of TOS field implementations. The three precedence bits of TOS field are mapped to the first three bits of DSCP field. There can be seven PHBs, one each for a particular precedence level. The service level of DSCP field having higher numerical value (e.g. 111000) is higher than that of a lower DSCP value (e.g. 101000). DSCP values 110000 and 111000 are used for the network control (e.g. management traffic, routing traffic) traffic. The last three bits of the DSCP field are always zero for class selector PHBs as indicated below:

Precedence level	DSCP field
1 (001)	001 000
2 (010)	010 000
3 (011)	011 000
4 (100)	100 000
5 (101)	101 000
6 (110)	110 000
7 (111)	111 000

Default PHB (DE). RFC 1812 specifies the default PHB as the conventional best effort forwarding behaviour. DSCP value for DE PHB is 000000. A node will forward as many of the packets bearing this DSCP value and as soon as possible depending on availability of resources.

23.13.3 Differentiated Service (DS) Domain

It may be noted that IETF has not prescribed quantitatively the externally observable forwarding behaviour for any PHB. It is for the network administrator to select the PHBs he wants to implement in his network and define his service provisioning policies for selected types of PHBs. These policies relate to classification, scheduling, and dropping of packets.

Differentiated service (DS) domain is a contiguous set of IP routers that operate with common set of service provisioning policies and PHB definitions (Figure 23.30). It is typically managed by a single administrative authority. It consists of DS boundary nodes and interior nodes:

- A DS boundary node is basically responsible for classification, marking and possibly traffic conditioning functions. It classifies each incoming IP packet based on its header and writes the DSCP value corresponding to the PHB to be applied to it. It also polices the incoming flows to restrict them to the agreed bandwidth and burst sizes. The outgoing traffic to the next DS domain requires traffic shaping and it may also require rewriting DSCPs in conformance to the definitions adopted in the next DS domain.
- The interior nodes classify the IP packets based on their DSCP field and give the corresponding forwarding treatment of their class.

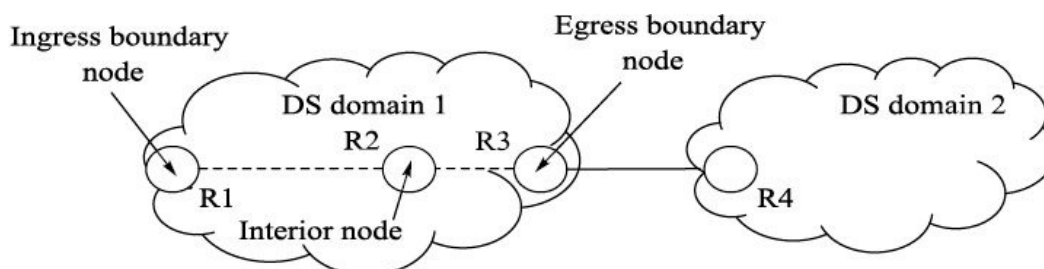


FIGURE 23.30 DS domain.

23.14 DIFFERENTIATED SERVICE SUPPORT IN MPLS

When MPLS is implemented in an IP network, the routers take the forwarding decision based on MPLS labels on the IP packets and the label switching tables. The routers do not look at the IP header. For implementing differentiated service framework in MPLS environment, there are the following two approaches. The names given to these approaches are somewhat drawn-out but the concept

behind them is simple:

- EXP Inferred PHB Scheduling Class LSPs (E-LSP).
- Label Only Inferred PHB Scheduling Class LSPs (L-LSP).

EXP inferred PHB scheduling class LSPs (E-LSP). This approach is based on applying differentiated service framework to each LSP. The three EXP bits of MPLS header are used for this purpose (Figure 23.31). These bits are used for indicating differentiated service class and drop priority of a packet. Thus the MPLS label on packet determines its path and the EXP bits determine the PHB to be applied to it.



FIGURE 23.31 MPLS header.

Three bits can define a maximum of 8 values. There is lack of compatibility between EXP bits and DSCP bits but in the core networks where MPLS is implemented, eight-level service differentiation is considered adequate. There can be number of ways these bits can be used. For example, we can use the first two bits for indicating the class and the third bit for marking a packet in-profile or out-of profile. Or all the three bits can be used for indicating class of traffic.

One possible way is to define classes as given:	111	Network control
	101	EF
	001	AF1
	010	AF2
	011	AF3
	000	Best effort (BE)

Choice of various classes and their codification is left to the network administrator. Different drop policies for in-profile and out-of-profile traffic can be defined each class of traffic.

Label only inferred PHB scheduling class LSPs (L-LSP). This approach is based on having separate LSPs for each <FEC, BA> pair and applying a particular PHB to each LSP. Thus to carry MPLS traffic between two boundary nodes across an MPLS domain, as many LSPs as the number of classes of traffic are required (Figure 23.32). In this case, the MPLS label on a packet determines its path and also the PHB applicable to it. Note that the traffic of the same FEC but different DSCP classes is carried on different LSPs. The EXP field can be

used to indicate drop preference in this case.

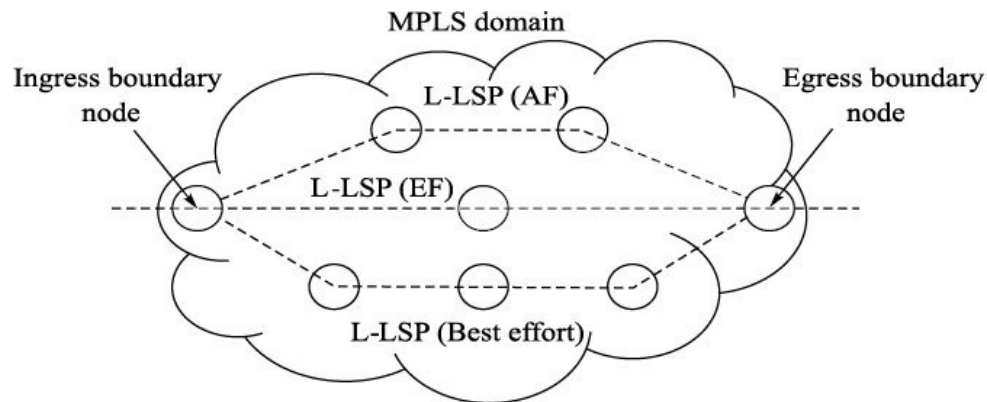


FIGURE 23.32 Separate L-LSPs for supporting three different PHBs.

23.14.1 Limitations of Differentiated Service Framework

Differentiated service framework provides a scalable and coarse-grained framework for implementing QOS in large IP networks. Some of the challenges that it faces today are listed as follows:

- For implementing differentiated service framework, one needs to have knowledge of the applications, their traffic statistics and QOS requirements in advance. Various classes, PHB definitions, marking and remarking policies, drop policies, bandwidth for various classes and so on are implemented in the IP network based on this knowledge.
- There is loss of granularity for QOS assurance. The network resources are provisioned for a class aggregate. A flow may not get the required QOS.

SUMMARY

Traffic classification, traffic scheduling, traffic control and congestion management are the basic requirements for implementing quality of service in an IP network. These requirements are met by implementing QOS tools in the network nodes.

- Traffic classification is based on one or more fields from the IP and TCP headers.
- Traffic scheduling refers to mechanism of serving queues of IP packets at the output ports. Priority queuing, weighted fair queuing and weighted round robin are the three basic mechanism of serving multiple queues.

- Traffic control is carried out by implementing policing and shaping functions.
- For congestion management, admission control, drop policy and ECN (explicit congestion notification) are implemented. Admission control function checks the availability of required network resources before permitting a flow to commence. Tail dropping and RED are two alternative drop policies. ECN is an alternative to packet dropping. It marks the packets with ‘congestion experienced’ notification for the TCP layers to take suitable action.

The two frameworks for implementing quality of service in IP networks are integrated service and differentiated service. Integrated service framework is based on reservation of network resources in advance of commencement of a flow. It uses resource reservation protocol (RSVP) for this purpose. It can offer guaranteed quality of service. Integrated service framework is suitable for small networks only.

Differentiated service framework offers several grades of the ‘best effort’ service. IP packets of these grades of service are differentiated by the value in the DSCP field. Each DSCP value is assigned a ‘per-hop-behaviour (PHB)’ which is simply a defined forwarding treatment given by a network node to an IP packet. We have four types of PHBs, expedited forwarding (EF) PHB, assured forwarding (AF) PHBs, class selector PHBs and default (DE) PHB. EF PHB defines the highest level of the differentiated service. In MPLS environment, these PHBs are mapped to either the three EXP bits in the MPLS header or different LSPs are created for different PHBs.

EXERCISES

1. If average data rate is 1 Mbps, peak data rate is 10 Mbps and the maximum jitter is limited to 200 ms, what is the buffer size needed at the receiver end to remove the jitter? Assume measurement interval of 100 ms for average and peak data rates.
2. Show that if the first received packet is delayed by an amount equal to the upper bound of jitter, the jitter can be eliminated completely.
3. In a queue with FIFO, UDP flows are favoured over TCP flows when congestion occurs. Can priority queuing overcome this limitation by assigning TCP flows to higher priority queue?

4. Queues A and B always have octets to send and are served by weighted fair queue scheduler that is configured to give weights of 1 and 4 to queue A and B respectively. If 5 kb are drained from queue A in interval $[0,1]$, how many bits will be drained from queue B in the same interval?
5. In a fair queuing system, the packet being serviced by the scheduler has finish number F . Is it possible for a new packet of another flow to get a finish number lower than F ? If so, give an example. If no, explain why?
6. Packets of size 100 bits and 200 bits arrive at $t = 0$ in queues A and B almost simultaneously. The queues are empty initially. Fair queuing is used and the rate at which the packets are dispatched on the line is 100 bits/s.
 - (a) What are the finish numbers of the packets? Assume the fair queue scheduler sends notionally one bit from a queue in a round.
 - (b) In how much time both the packets are sent?
 - (c) What is the round number when the scheduler completes transmission of the first packet?
 - (d) If a packet of size 20 bits arrives in queue A at time $t = 1.5$ s, what will be its finish number?
7. Rework exercise 6 if the queues A and B have weights 2 and 5.
8. A router receives three flows A, B and C almost simultaneously. The fair queuing scheduler starts sending the packets after the packets given below are received. Give the order of transmission of these packets.

TABLE 23.4 Order of Transmission				
Flow	Packet 1	Packet 2	Packet 3	Packet 4
A	100	100	100	100
B	180	200		
C	120	50		

9. In Exercise 8, the scheduler uses weighted fair queuing with flows A, B and C sharing the bandwidth in the proportion 1:2:1.5 respectively Give the order of transmission of these packets.
10. User A sends Telnet traffic and user B sends FTP traffic through a router that has separate queues for these flows. The output link of the router is slow

enough to always keep packets in the queues. If A sends packet of size 40 octets and B sends packet of size 560 octets, how is the output bandwidth shared between the two flows if: (a) fair queuing is used?

(b) round-robin queuing with one packet per queue per round is used?

(c) round-robin queuing with 50 octets per queue per round is used?

11. Queue A has higher priority than queue B. Queue A has a leaky bucket policer that limits the arrival rate of equal sized packets to 5 packets/s. The bucket size is 10 packets. The leaky bucket parameters of queue B are 10 packets and 2 packets/s. The scheduler serves the queues at the rate of 10 packets/s. What are the worst case delays in queues A and B?
12. A peak rate token bucket policer limits the peak rate to 1 Mbps. The measurement interval is 10 ms. What is the permitted minimum interval between the arrival times of consecutive packets, if the size of packets is 1250 octets?
13. A link is to be provisioned for L3-bandwidth of 500 kbps. If an L2 policer is used, to what value it should be set if the average IP packet size is 1000 octets and L2 overhead is 29 octets?
14. Draw the flow chart of leaky bucket algorithm using the parameters indicated in Section 6 of the chapter.
15. A leaky bucket shaper receives packet at constant average rate (R packets/s) and dispatches them at the same rate. If there is a burst of n packets which is within the capacity of the queue, how will the performance of shaper be affected by the buffer?
16. The drop policy of a router is based on the cost of retaining a packet in the queue. Packet with the highest cost is dropped first when the queue is full. Cost is defined as the size of a packet in octets multiplied by its position in the queue (number of octets ahead). The status of the queue when it is full is as shown in Figure 23.33.
 - (a) Determine which packet will be dropped.
 - (b) List possible advantages/disadvantages of this policy compared to tail drop.



Figure 23.33 Exercise 16.

17. The destination address in the IP header of the PATH message is the IP address of the receiver. The destination address in the IP header of RESV

- request message is: (a) the IP address of the sender of PATH message.
 (b) the IP address of the next upstream router from which it received the PATH message.
 (c) the IP address of the next upstream RSVP enabled router from which it received the PATH message.
18. Hosts A and B are connected through a source router and a destination router. The probability of dropping a packet by a router is p . When a packet is dropped by a router, the source host retransmits it after time out. If a router is counted as one hop, what is the mean number of: (a) hops a packet makes?
 (b) retransmissions a packet makes?
 (c) hops required per received packet?
19. In an ATM network, the token rate is 10 ms. What is the average data rate?
20. If the token bucket size is 8 Mbits and the token rate is 1 Mbps, what is the maximum size of burst at peak rate of 6 Mbps?
21. The network in Figure 23.34 uses RSVP with multicast trees for transmissions from host A and B. Suppose the C needs 3 Mbps to A and 1 Mbps bandwidth to B. The requirements of D are 2 Mbps to A and 2 Mbps to B for the same transmissions. How much bandwidth is reserved for these requirements in routers R1, R2, R3, R4, R5 and R6?

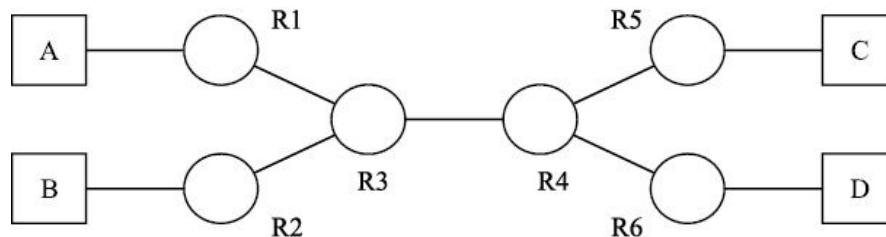


FIGURE 23.34 Exercise 21.

22. When the counter value is at C , the probability of discarding a packet in RED is given by $p = P/(1 - PC)$, where P drop probability demanded by the RED drop profile for a given average queue depth. What is the expected number of packets that are added to the queue between two discards?

¹ VoIP: Voice over IP.

² DSCP field is described later in this chapter.

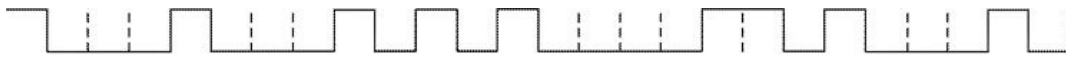
Bibliography


- Black, U., *Data Networks*, Prentice Hall, Englewood Cliffs, New Jersey, 1990.
- , *Computer Networks: Protocols, Standards and Interfaces*, 2nd ed., Prentice-Hall of India, New Delhi, 1993.
- , *Data Link Protocols*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- , *Data Communications and Distributed Networks*, 3rd ed., Prentice-Hall of India, New Delhi, 1995.
- , *Architecture for Digital Signaling Networks*, Prentice Hall, Englewood Cliffs, New Jersey, 1997.
- Buckwater, J.T., *Frame Relay: Principles and Applications.*, Addison-Wesley, Reading, Mass., 2000.
- Chow, M.C., *Understanding Sonet/SDH*, Andan Publishers, Holmdel, New Jersey, 1995.
- Cole, G.D., *Implementing OSI Networks*, John Wiley & Sons, New York, 1990.
- Comer, D.E., *Internetworking with TCP/IP*, 4th ed., Prentice-Hall of India, New Delhi, 1997.
- Davidson, R.P., *Internetworking LANs Operations*, Artech House, Norwood (Mass.), 1992.
- Davie, B.S., and Y. Rekhter, *MPLS: Technology and Applications*, Cisco Press, Indianapolis, 2000.
- Forouzan, B.A., *Data Communications and Networking*, Tata McGraw-Hill, New Delhi, 2003.
- Ginsburg, D., *Implementing ADSL*, Addison-Wesley, Reading (Mass.), 1999.
- Hallsal, F., *Data Communications, Computer Networks and Open Systems*, Pearson Education, 1996.
- Held, G., *Data Communications and Networking Devices*, John Wiley & Sons, 1992.
- Huitema, C., *IPv6: The New Internet Protocol*, Prentice Hall, Englewood Cliffs, New Jersey, 1996.
- , *Routing in the Internet*, Prentice Hall, Englewood Cliffs, New Jersey, 1996.
- Jain, B.N. and A.K. Agrawala, *Open System Interconnection*, Elsevier, New York, 1990.
- Kaufman, G., R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*, 2nd ed., Prentice-Hall of India, New Delhi, 2002.
- Keshav, H., *An Engineering Approach to Computer Networking*, Pearson Education, Delhi, 1997.
- Maufer, T.A., *IP Fundamentals*, Addison-Wesley, Reading (Mass.), 1999.
- McDysan, D.E. and D.L. Spohn, *ATM Theory and Application*, McGraw-Hill, New York 1998.
- Moy, J.T., *OSPF Anatomy of the Internet*, Addison-Wesley, Reading (Mass.), 1998.
- Perlman, R., *Interconnections: Bridges and Routers*, Addison-Wesley, Reading (Mass.), 1999.
- Peterson, L.L. and B.S. Davie, *Computer Networks*, Morgan Kaufman, New York, 2003.
- Seifert, R., *Gigabit Ethernet: Technology and Applications for High Speed LANs*. Addison-Wesley, Reading (Mass.), 1998.
- Sexton, M. and A. Reid, *Broadband Networking: ATM, SDH and Sonet*, Artech House, Norwood (Mass.), 1997.
- Smith, P., *Frame Relay: Principles and Applications*, Addison-Wesley, Reading (Mass.), 1993.
- Stallings, W., *Data and Computer Communications*, 7th ed., Prentice-Hall of India, New Delhi, 2004.
- , *High Speed Networks and Internets*, Pearson Education, Delhi, 2002.
- , *ISDN and Broadband ISDN with Frame Relay and ATM*, 4th ed., Prentice-Hall of India, New Delhi, 1999.
- Stevens R.W., *TCP/IP Illustrated: Protocols*, Addison-Wesley, Reading (Mass.), 1994.
- Tanenbaum, A.S., *Computer Networks*, 4th ed., Prentice-Hall of India, New Delhi, 2002.
- Vesegna, S., *IP Quality of Service*, Cisco Press, Indianapolis, 2001.
- Wright, R., *IP Routing Primer*, Cisco Press, Indianapolis, 1998.

Answers to Selected Exercises

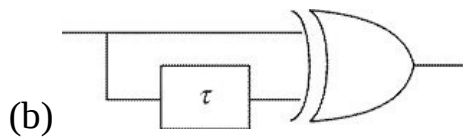
CHAPTER 1

1. (b) 00100010 10000110 00101110 10000110



(c) 

4. (a) NRZ-M



5. (a) 001001110

(b) 11011000

(c) 00111010

6. (a) 11001100

(b) 00101010

7. (a) 00011000

(b) 11100111

8. 11010000010000

9. 0100000001100000000000000000

10. $B = 600$ Hz 11. $S/N = 24.06$ dB

12. (a) Distinct (b) No 13. (a) Uniquely decodable (b) No (c) No 14. 1, 00, 00, 00, 00, 1

15. (a) $R = 20.640$ kbps (b) 10

(c) 18600 bps 16. (a) $H = 2.2925$

(b) $R = 0.7075$

(c) $L = 2.333$

(d) 0.041

17. (a) $H = 2.893$

(b) $a = 11, b = 1000, c = 001, d = 010, e = 101, f = 000, g = 1001, h = 011, L = 2.925, R = 0.032$

(c) $a = 00, b = 1110, c = 011, d = 100, e = 110, f = 010, g = 1111, h = 101, L = 2.925, R = 0.032$

CHAPTER 2

2. $C_{-1} = -5/12$ $C_0 = 1$ $C_1 = -5/12$

3. (i) 50 dBm (ii) 20 dBm (iii) 0 dBm (iv) -60 dBm (v) -90 dBm 4.

0.27 10^{-5} s 5. 55000 ps

CHAPTER 3

2. $3f_1, 3f_2, 3f_3, 2f_1 - f_2, 2f_2 - f_3, 2f_3 - f_1, 2f_2 - f_1; 2f_1 - f_3, 2f_3 - f_2, f_1 + f_2 + f_3, f_1 + f_2 - f_3, f_1 + f_3 - f_2, f_2 + f_3 - f_1.$

Components in frequency range 1 – 2 kHz : $2f_1 - f_3, 2f_2 - f_3, f_1 + f_3 - f_2$

3. -32.5 dBm/kHz 4. $6.7 \cdot 10^{-14}$ W

6. 6.25%

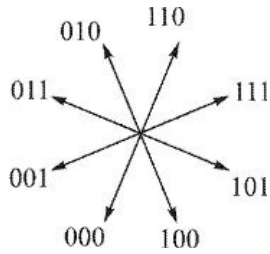
7. $3.9 \cdot 10^{-3}$ ms (E1), $5.14 \cdot 10^{-5}$ ms 8. 1984 kbps 9. 124992 kbps

CHAPTER 4

1. (a) (iii) (b) (ii) (c) (ii) (d) (i) (e) (iii) 2. $f 0 p p p 0 p 0 0$

3. $AB = 00, f = -p/4, AB = 01, f = -3p/4, AB = 11, f = 3p/4, AB = 10, f = p/4$

4. Baud rate = $1/t = 1 \text{ M}$, bit rate = 4 Mbps.



5. 000 100

6. (i) Line efficiency = 0.696, Throughput = 1670.4 cps (ii) (a) No. (b) At the end of 16 seconds, there will be a queue of 3993.6 characters.

(iii) Time required to clear the queue = 2.4s.

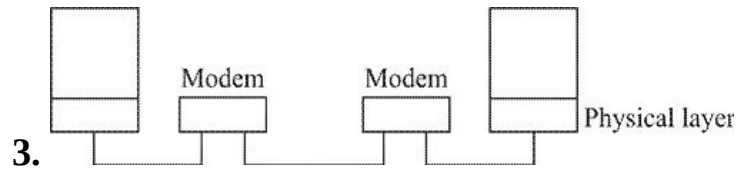
7. Line utilization efficiency = 97.3%

CHAPTER 5

1. Probability of receiving the block with errors = 0.016.
2. (a) 00100011 101000101 1000010 11110010 00100011 10100010 01001010
(b) DOLLAR
(c) Third octet has errors.
3.

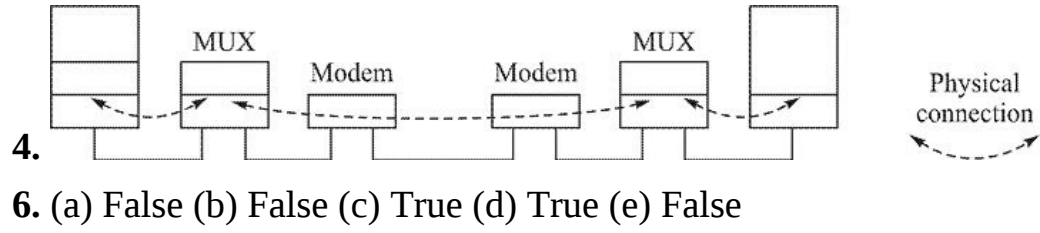
		Bit			sequence
11001011	00101010	10000011	01001010	00101010	00000100 (
10010010	00101010	00000010			
4. Shift 1
5. No.
6. (a) 00000111
(b) There are errors.
7. (a) 10011010, 00001011
8. 10100010111000
10. The sixth bit position has error. The corrected code word is 11110100101.
11. (a) 01100101101
(b) 00011111111
(c) 111010110100
12. (a) > (b) j (c) < 13. 11101000
14. 1110
15. 20
16. 6
17. 0.999999046
18. 7

CHAPTER 6



7. $M/[M + (N - 1)H]$

CHAPTER 7



CHAPTER 8

1. 192 bits 2. (a) 0.689%
(b) 4.8%
(c) 87.6%
(d) 100%

5. (a) 0.194 Mbps (b) 1.1 Mbps 6. Yes 9. 1.0096

CHAPTER 9

6. (b)

A	ENQ		STX- ETB		STX- ETB		STX- ETB		ENQ		EOT
B		ACK0		NAK		ACK1		ACK0		ACK0	

8. (a) Address Control Information FCS
 11001100 01110010 000011100 1010101010011111

(b) Address Control Information FCS
 11011111 01111101 1010101 1111111010101111

9. 0111111000011111000110111110111000011100000010111000101111110

10. (a) 00100101
 (b) 10101101
 (c) 11001110

11. (a)

B U SNRM P		B I 0 0 P		B I 1 1 P
	B U UA F		B I 0 1 F	
	B U DISC P			
B S RR 2 F		B U UA F		

(b)

B U SNRM P		B I 0 0 P	Timeout	B S RR 0 P
	B U UA F			
	B I 0 0 P			B I 1 0 P
B S RR 0 F		B I 0 1*	B I 1 1 F	
		B S RNR 2 P		B U DISC P
B I 0 2	B I 1 2 F		B S RR 2 F	B U UA F

(c)

B U SNRM P		B I 0 0 P	B I 1 0	B I 2 0
	B U UA F		B I 0 1	B I 1 2 F
B S RR 0 P			Timeout	B S RR 1 P
	B I 0 3	B I 1 3 F*		
	B S RNR 2 P		B U DISC P	
B I 1 3 F		B S RR 3 F		B U UA F

12.

B U SNRM P		B S RR 0 P		B I 0 0 P
	B U UA F		B S RR 0 F	
	B U DISC P			
B S RR 1 F		B U UA F		

13.

B I 0 0	B I 1 0 P			B I 2 0	B I 3 0 P		
		B I 0 2	B I 1 2 F			B I 0 2	B I 1 2 F
B I 2 2	B I 3 2 P			B U DISC P			
		B S RR 4 F			B U UA F		

14.

		B S RR 2			B I 0 3	B I 1 3 P	
B I 0 0	B I 1 0		B I 2 0	B I 3 0			B I 3 2 F
B S RNR P		B U DISC P					
	B S RR 2 F			B U UA F			

CHAPTER 10

3. (a) 1.005 ms (b) Yes (c) 200 km 4. 200 m 5. 0.105 ms

CHAPTER 11

4. 600 octets 5. 6000 octets 6. (a) $1/3$
(b) $1/6$

7. B

9. DA: 00 00 66 33 B5 49, SA: 00 00A7 12 36 B7, Length: 00 60, Data: AA AA
03 00 00 00 08 00 48 45 4C 4C ...

It is IEEE 802.3 frame.

10. 1101 1110 1010 1101

CHAPTER 13

5.

Input data sequence	0 1	1 0	0 0	0 0	1 1
Modulator output (MHz)	2	4	1	1	3
Synthesizer binary input	001	110	011	001	001
Synthesizer output (MHz)	20	70	40	20	20
Transmitted frequency (MHz)	22	74	41	21	23

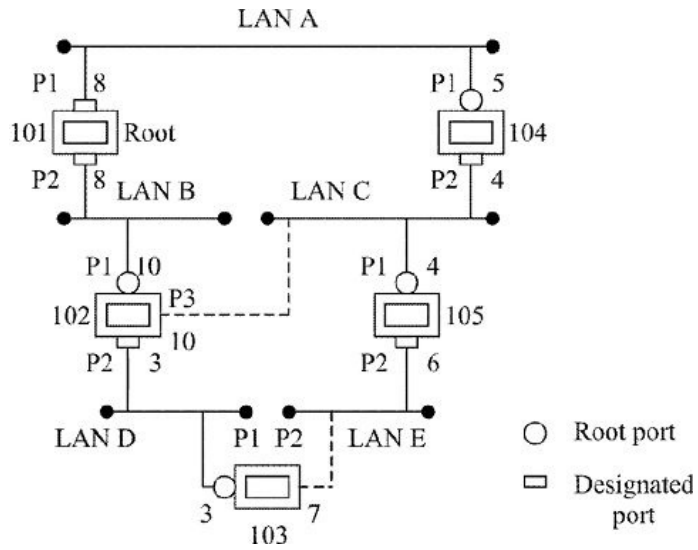
6.

Input data sequence	0 1		1 0		0 0		0 0		1 1	
Modulator output (MHz)	2		4		1		1		3	
Synthesizer binary input	001	110	011	001	001	001	110	011	001	001
Synthesizer output (MHz)	20	70	40	20	20	20	70	40	20	20
Transmitted frequency (MHz)	22	72	44	24	21	21	71	41	23	23

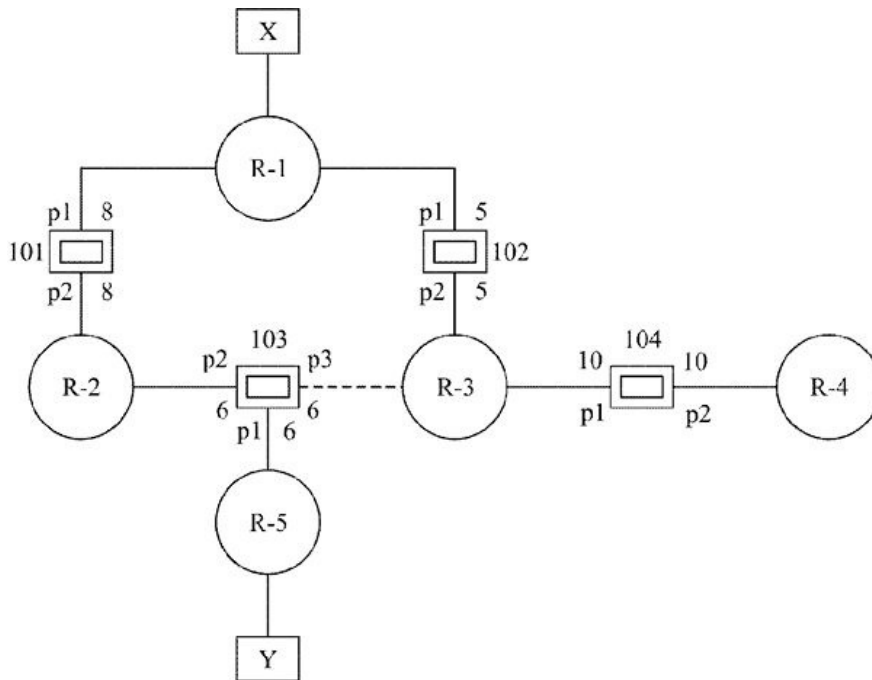
7. 100101100110100110010110

8. 001101

CHAPTER 14



2. (R-1, 101), (R-2, 103), (R-5), (R-1, 102), (R-3, 103), (R-5)
 Both the routes involve same number of hops.



3. (a)

(b) X R-1 101 R-2 103 R-5 Y

X R-1 102 R-3 104 R-4

(c) Y R-5 103 R-2 101 R-1 X

Y R-5 103 R-3 102 R-1 X

Y R-5 103 R-3 104 R-4

4.

B1	
p1	p2
A	

B2	
p1	p2
A	
	C

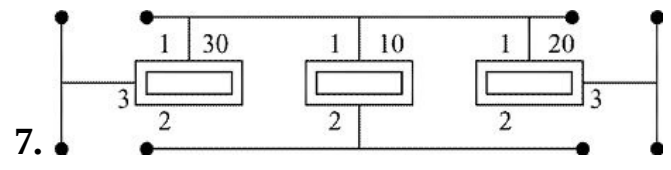
B3		
p1	p2	p3
A		
		C
	D	

B4	
p1	p2
A	
	D

(a)

(b)

(c)



CHAPTER 15

3. P A B C D R, P C B Q, Q B D R

4.

	P				Q				R				S			
	A	b	c	d	A	b	c	d	a	b	c	d	a	b	c	d
A-B	0	0				0	0						0			0
C-G	1		0		0		1		0		0					
E-I						1		0						0		1
D-B	2			0		2	2						1			2
F-J					1	3					1	0			0	3
H-A	3	1			2		3			0	2					

5.

Router P				Router Q				Router R				Router S			
Ic	Ds	Og	Pc	Ic	Ds	Og	Pc	Ic	Ds	Og	Pc	Ic	Ds	Og	Pc
a	B	b	1	a	A	b	1	a	A	c	3	a	A	c	1
a	C	c	1	a	C	b	2	a	B	c	2	a	B	c	2
a	D	b	3	a	D	c	2	a	C	b	3	a	D	b	3
b	A	a	Lc	b	B	a	Lc	b	D	a	Lc	b	C	a	Lc
b	C	c	1	b	D	c	2	c	D	a	Lc	c	C	a	Lc
c	A	a	Lc	c	A	b	1								
c	B	b	1	c	B	a	Lc								

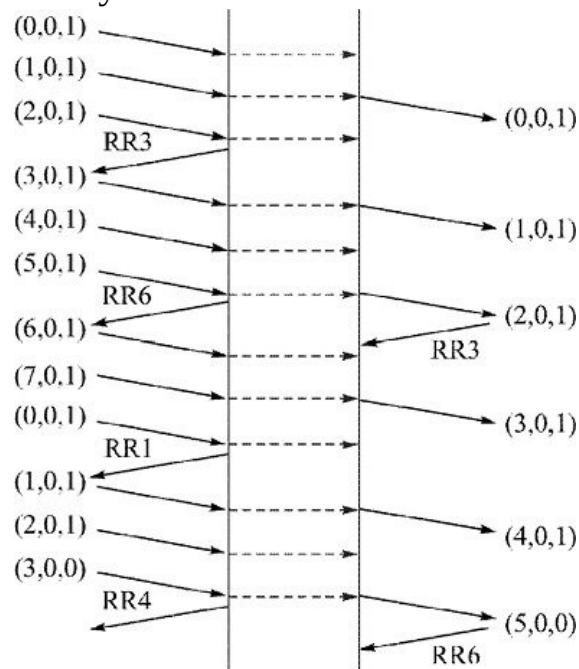
Ic: Incoming port, Ds: Destination, Og: Outgoing port, Pc: Path cost, Lc: Local

11. (a) B-A-C, 2
 (b) C-D-A-B, 3
 (c) C-A-B

CHAPTER 16

2. Total bits (Flag to FCS) = 72, transmission time = 0.03 s.

3. (a) No (b) 1 and 2 respectively 7. Number of data packets required to be sent by the DTE = $1500/128 = 12$ (rounded to next higher integer). At the receiving end only 6 packets will be required. The numbers within the brackets are $N(s)$, $N(r)$ and M bit respectively.



10. (a) No (b) Yes (c) Yes. The frames will be discarded only when the congestion occurs.

(d) 2 Mbps 11. (a) D, E

(b) E

13. (a) ATM cells : 12, additional octets : 124

(b) ATM cells : 11, additional octets : 71

CHAPTER 17

1.

Fragment	Total length	MF	Offset	Data octets
First	508	1	0	0-487
Second	508	1	61	488-975
Third	508	1	122	976-1463
Fourth	36	0	183	1464-1479

2.

N1		N2	
Total length	Offset	Total length	Offset
1020	0	508	0
		508	61
		44	122
1020	125	508	125
		508	186
		44	247
88	250	88	250

5. 172.27.0.0/19, 172.27.32.0/19, 172.27.64.0/19, 172.27.96.0/19,
172.27.128.0/19, 172.27.160.0/19, 172.27.192.0/19, 172.27.224.0/19

Subnet mask 11111111.11111111.11100000.00000000

6. 30 hosts, 16 hosts 7. (a) /27

(b) 30

(c) 8

(d) 196.35.1.0/27, 196.35.1.32/27, 196.35.1.64/27, 196.35.1.96/27,
196.35.1.128/27, 196.35.1.160/27, 196.35.1.192/27, 196.35.1.224/27

(e) 196.35.1.223

9. (a) a (b) c (c) e (d) d (e) e 10. (a) Correct (b) Wrong (c) Correct 13. 262142

14. 196.94.79.18

15. 4094

CHAPTER 18

1. (a) C: A-2, B-1, D-2, D: A-2, B-1, C-2
 (b) To B: A-, B-1, C-, To C : A-2, B-1, C-2
3. (a) A: B-4, C-1, D-, E-2, B: A-4, C-1, D-2, E-, C: A-1, B-1, D-4, E-3, D: A-, B-2, C-4, E-, E: A-2, B-, C-3, D-
 (b) A: B-4, C-1, D-, E-2, B: A-4, C-1, D-2, E-6, C: A-1, B-1, D-4, E-3, D: A-, B-2, C-4, E-, E: A-2, B-6, C-3, D-
 (c) A: B-4, C-1, D-6, E-2, B: A-4, C-1, D-2, E-6, C: A-1, B-1, D-3, E-3, D: A-6, B-2, C-3, E-8, E: A-2, B-6, C-3, D-

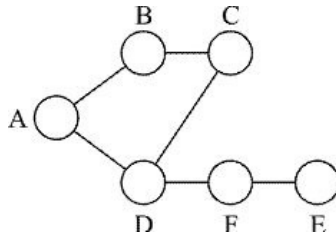
4. Forwarding table of A.

Destination	Distance	Next hop
B	2	C
C	1	C
D	4	C
E	2	E

5 Forwarding table of A.

Destination	Distance	Next hop
B	2	B
C	6	B
D	8	B
E	9	E
F	13	B
G	9	B
H	8	B
I	12	B

7. Local preference 9. R



10.

11. Forwarding table of A.

Destination	Cost	Next hop
N1	1	N1
N2	3	N2
N3	4	N2
N4	6	N2
N5	4	N2
N6	5	N2
N7	7	N2

CHAPTER 19

1. S1: N3 R2 B

N1 R1 N2 A R4 N4 C, D

R5 N5 E

S2: N6 R7 E

R6 N4 C, D

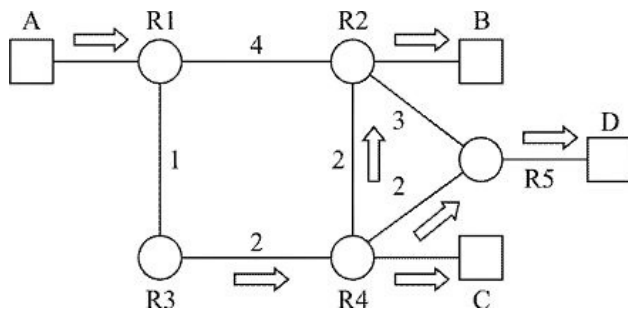
R3 N2 A R4 N3 R2 B

2. $R1 : 1$, $R2 : 2$, $R3 : 1$, $R4 : 1$, $R5 : 1$, $R6 : 1$, $R7 : 1$

3. $R1 = 2$, $R2 = 3$, $R3 = 1$, $R4 = 3$, $R5 = 2$

4. $R1 = 2$, $R2 = 1$, $R3 = 1$, $R4 = 2$, $R5 = 1$

6.



CHAPTER 20

6. (a) ssthresh = 10 k bytes, cwnd = 13 k bytes.

(b) 23 k bytes **8.** No **10.** 200 ms **11.** (a) 1 k bytes (b) 16 k bytes **12.** 366.3 Mbps

CHAPTER 21

3. Two, B2, and B3.

4. From plaintext Block-2 onwards all the blocks will get garbled.

5. (a) $d = 3$

(b) $c = 2$

(c) CAB : 12,1,8

6. (a) $d = 29$

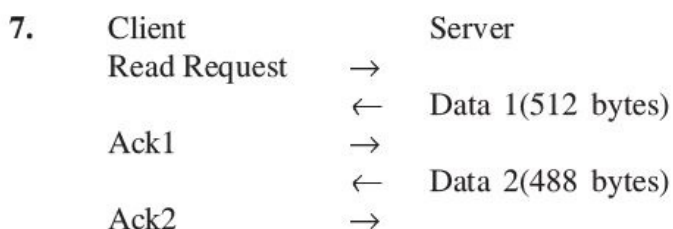
(b) $c = 50$

(c) 32

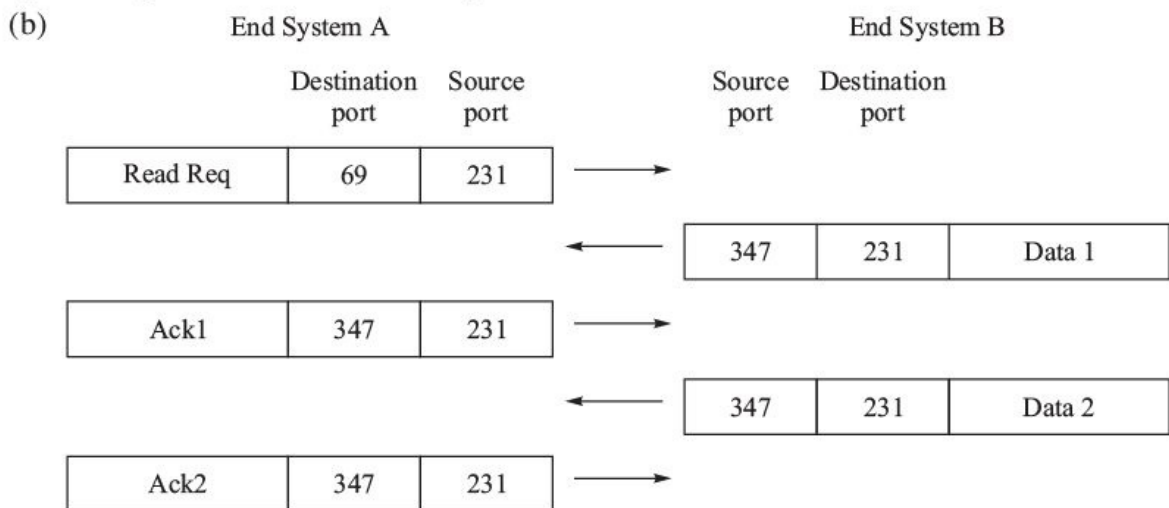
9. 40

CHAPTER 22

2. No. When a process makes a DNS request, it starts a timer. If the timer expires before receiving the response, the request is repeated.
3. No. DNS names are maximum 255 bytes long and they easily fit into 576-byte UDP packet.
4. `OPEN`.
6. (a) There is no response from the server. The client repeats read request after timeout.
 (b) After timeout the server resends the last acknowledgement to get the lost block of data.
 (c) The client resends the last block of data.



8. (a) End system A : Client, End System B : Server



CHAPTER 23

1. 20 Mb 3. No 4. 20 kb 6. (a) 100, 200 (b) 3 s (c) 50 (d) 95
8. A1, C1, C2, B1, A2, A3, B2, A4
9. C1, B1, A1, C2, B2, A2, A3, A4
10. (a) Equally (b) 1:14 (c) 1:7
11. There is no delay in queues A and B.
12. 10 ms 15. After the burst the packets will be delayed by n/R seconds.
21. R1:3 Mbps; R2:2 Mbps; R3 and R4: 3Mbps for flow from A, 2 Mbps for flow from B
R5:3 Mbps from A and 2 Mbps from B; R6: 2 Mbps each for A and B.

List of Acronyms

AAL

ABM

ABR

ACK

ADCCP

ADM

ADSL

AES

AFI

AMI

AMP

ANSI

AP

ARB

ARM

ARP

ARPA

ARQ

AS

ASBR

ASCII

ASK
ASN.1

ATM

AU

BCC

BDR

BECN

BER

BGP

BISDN
BISYNC/BSC

BOOTP

BPDU

BPSK

BRA

ATM Adaptation Layer

Asynchronous Balanced Mode

Area Border Router

Acknowledgement

Advanced Data Communications Control Procedure Add-Drop Multiplexer

Asymmetric Digital Subscriber Line Advanced Encryption Standard Authority and Format Identifier Alternate Mark
Inversion

Active Monitor Present

American National Standards Institute Access Point

All Route Broadcast

Asynchronous Response Mode

Address Resolution Protocol Advanced Research Projects Agency Automatic Repeat Request

Autonomous System

Autonomous System Border Router American Standards Committee for Information Interchange Amplitude Shift Keying
Abstract Syntax Notation.1

Asynchronous Transfer Mode

Administrative Unit

Block Check Character

Backup Designated Router

Backward Explicit Congestion Notification Bit Error Rate

Border Gateway Protocol

Broadband ISDN

Binary Synchronous Control

Bootstrapping Protocol

Bridge PDU

Binary PSK

Basic Rate Access

BRI

BSS

CA

CAP

CBC

CBR

CBT

CCITT

CHAP

CIDR

CIR

CLNP

CLNS

CONS

CPE

CR

CRC

CS

CSMA
CSMA/CA
CSMA/CD

CSPDN

CT

CTS

CUG

DA

DCE

DCF

DE

DES

DHCP

DIFS

DLC

DLCI

DLE

DMT

DNS

DR

DS

DSCP

DSL

DSR

Basic Rate Interface

Basic Service Set

Certification Authority

Carrierless Amplitude and Phase modulation Cipher Block Chaining

Constant Bit Rate

Core-Based Tree

Consultative Committee for International Telegraphy and Telephony Challenge Handshake Authentication Protocol

Classless Interdomain Routing Committed Information Rate

Connectionless-mode Network Protocol Connectionless Network Service Connection-Oriented Network Service

Customer Premises Equipment Carriage Return

Cyclic Redundancy Check

Convergence Sublayer

Carrier Sense Multiple Access CSMA with Collision Avoidance CSMA with Collision Detection Circuit Switched Packet

Data Network Claim Token

Clear-To-Send

Closed User Group

Destination Address

Data Circuit Terminating Equipment Distributed Coordination Function Discard Eligibility (bit)

Data Encryption Standard

Dynamic Host Configuration Protocol DCF-IFS

Data Link Control

Data Link Connection Identifier Data Link Escape (character) Discrete Multitone Modulation Domain Name System

Designated Router

Distribution System

Differentiated Service Code Point Digital Subscriber Line

Data Set Ready

DSSS

DTE

DTR

DVMRP

EBCDIC

ECMA

EDI

EGP

EIA

EIR

ESS

ETSI

FCS

FDDI

FDM

FDX

FE

FEC

FECN

FEXT

FHSS

FLP

FSK

FTP
HDB-3

HDLC

HDSL

HDX

HTML

HTTP
IA5

IAB

IANA

IBSS

ICMP

IDI

IDP

IDRP

IEEE

IETF

IFS

IGMP Direct Sequence Spread Spectrum Data Terminal Equipment

Data Terminal Ready

Distance Vector Multicasting Routing Protocol Extended Binary Coded Decimal Interchange Code European Computer

Manufacturers Association Electronic Data Interchange Exterior Gateway Protocol

Electrical Industries Association Excess Information Rate

Extended BSS

European Telecommunications Standards Institute Frame Check Sequence

Fibre Distributed Data Interface Frequency Division Multiplexing Full Duplex

Fast Ethernet

Forwarding Equivalence Class Forward Explicit Congestion Notification Far End Crosstalk

Frequency Hopping Spread Spectrum Fast Link Pulses

Frequency Shift Keying

File Transfer Protocol

High Density Bipolar-3

High-Level Data Link Control High-data-rate DSL

Half Duplex

Hypertext Markup Language

Hypertext Transfer Protocol International Alphabet Number 5

Internet Architecture Board Internet Assigned Numbers Authority Independent BSS

Internet Control Message Protocol Initial Domain Identifier

Initial Domain Part

Inter Domain Routing Protocol Institute of Electrical and Electronics Engineers Internet Engineering Task Force Inter-Frame Space

Internet Group Management Protocol

IGP
IMAP4

IP

IPDU

IPRA

IS

ISDN

ISI
IS-IS

ISO

ISOC
ITU-T

IWU

LAN

LANE
LAP-B,-
D,-F,-M

LCN

LCP

LDP

LER

LF

LLC

LRC

LSA

LSP

LSR

MAC

MAN

MAP
MD5

MIB

MIME

MLP

MOSPF

MPLS

MSS

MTA

MTU

MUA

MUX

NAK

Interior Gateway Protocol
Internet Message Access Protocol-4

Internet Protocol

Internet Protocol Data Unit Internet Policy Registration Authority Intermediate System

Integrated Services Digital Network Inter Symbol Interference

Intermediate System to Intermediate System (protocol) International Organization for Standardization Internet Society

International Telecommunication Union-Telecommunication Standards Sector Interworking Unit

Local Area Network

LAN Emulation

Link Access Procedure

-Balanced,-D Channel,-Frame mode services,-Modem Logical Channel Number

Link Control Protocol

Label Distribution Protocol Label Edge Router

Line Feed

Logical Link Control

Longitudinal Redundancy Check Link State Advertisement

Label Switched Path

Label Switching Router

Media Access Control

Metropolitan Area Network

Manufacturing Automation Protocols Message Digest-5

Management Information Base Multipurpose Internet Mail Extension Multilink Procedure

Multicast Extensions to OSPF

Multiprotocol Label Switching Maximum Segment Size

Mail Transfer Agent

Maximum Transmission Unit

Mail User Agent

Multiplexer

Negative Acknowledgement

NAV

NCP

NE

NEXT

NMS

NNI

NPDU

NRM

NRZ

NRZI

NSAP

NSDU

NT

NVT

OFDM

OSI

OSPF

PABX

PAD

PAM

PAP

PCI

PCF

PCM

PDH

PDU

PE

PHB

PI

PIFS

PIM

PING

PMD

PNNI
POP3

PPP

PRA

PRI

PSDN

PSE

PSK

PSTN

Network Allocation Vector

Network Control Protocol

Network Element
Near-End Crosstalk

Network Management System
Network-Network Interface

Network Layer PDU

Normal Response Mode

Non Return to Zero

Non Return to Zero Invert on 1s Network Service Access Point Network Service Data Unit

Network Termination

Network Virtual Terminal

Orthogonal Frequency Division Multiplexing Open Systems Interconnection Open Shortest Path First

Private Automatic Branch Exchange Packet Assembler and Disassembler Pulse Amplitude Modulated

Password Authentication Protocol Protocol Control Information Point Coordination Function Pulse Code Modulation

Plesiochronous Digital Hierarchy Protocol Data Unit

Provider Edge (router)

Per Hop Behaviour

Protocol Interpreter

PCF-IFS

Protocol Independent Multicast Packet Internet Groper

Physical Medium Dependent

Private Network-Network Interface Post Office Protocol-3

Point-to-Point Protocol

Primary Rate Access

Primary Rate Interface

Packet Switched Data Network Packet Switching Exchange

Phase Shift Keying

Public Switched Telephone Network

PTT

PVC

QAM

QOS

RARP

REJ

RFC

RI

RIP

RNR

RPF

RR

RSA

RSVP

RTO

RTS

RTT

RZ

SA

SAP

SAR

SDH

SDLC

SDSL

SDU

SHA

SIFS

SMI

SMTP

SNA

SNMP

SNR

SONET

SRB
S-REJ

SRTT

SSL

STM

STP

SVC

TA

Post, Telephone and Telecommunications (Authority) Permanent Virtual Circuit
TC Quadrature Amplitude Modulation Quality of Service
Reverse Address Resolution Protocol Reject

Request For Comments

Ring Indication

Routing Information Protocol Receive Not Ready

Reverse Path Forwarding

Receive Ready

Rivest, Shamit and Adleman (Encryption algorithm) Resource Reservation Protocol Retransmission Time Out

Request To Send

Round Trip Time

Return to Zero

Source Address

Service Access Point

Segmentation and Reassembly (sublayer) Synchronous Digital Hierarchy Synchronous Data Link Control Symmetric DSL

Service Data Unit

Secure Hash Algorithm

Short IFS

Structure of Management Information Simple Mail Transfer Protocol System Network Architecture Simple Network

Management Protocol Signal To Noise Ratio

Synchronous Optical Network Single Route Broadcast

Selective Reject

Smoothed RTT

Secure Socket Layer

Synchronous Transport Module Shielded Twisted Pair

Switched Virtual Circuit

Terminal Adapter
Transmission Convergence (sublayer)

TCP

TDM

TFTP

THT

TOS

TPDU

TRT

TTL

TTRT

TWA

TWS

UDP

UHF

UI

UNI

URL

UTP

VBR

VC

VCC

VCI

VDSL

VHF

VLSM

VOATM

VOIP

VPC

VPI

VRC

WAN Transmission Control Protocol Time Division Multiplexing
Trivial File Transfer Protocol Token Holding Time

Type of Service

Transport Protocol Data Unit Token Rotation Time

Time To Live

Target Token Rotation Time

Two Way Alternate

Two Way Simultaneous

User Datagram Protocol

Ultra High Frequency

Unnumbered Information (frame) User-Network Interface

Uniform Resource Locator

Unshielded Twisted Pair

Variable Bit Rate

Virtual Container

Virtual Channel Connection

Virtual Channel Identifier

Very High Data Rate DSL

Very High Frequency

Variable Length Subnet Mask Voice Over ATM

Voice Over IP

Virtual Path Connection

Virtual Path Identifier

Vertical Redundancy Check

Wide Area Network

Index

4 PSK, 107

- demodulator, 108
- differential, 111
- modulator, 107

8 PSK, 106

16 QAM, 112

- modulator, 112

2B/1Q, 16
4B/3T, 16
4B/5B code, 368, 409
8B/6T, 363

100VG-AnyLAN, 361

- AAL1, 538
- AAL2, 538
- AAL3/4, 538–540
- AAL5, 540
- Abstract Syntax Notation 1 (ASN.1), 752
- Access Point (AP), 413
- Access rate, 522
- Acknowledged connectionless-mode service, 334
- Acknowledgement (ACK), 249
 - piggybacking of, 273
- Ad hoc network, 413
- Adaptive equalizer, 57 (*see also* Equalizer)
- Add-drop multiplexer (ADM), 77
- Address Resolution Protocol (*see* ARP)
- Administrative domain, 593
- Admission control, 767, 778, 792
- ADSL, 137
 - CAP modulation in, 138
 - distance limitations of, 138
 - DMT modulation in, 138
 - ITU-T recommendations for, 139
 - modulation schemes of, 138

- sub-channel allocation, 138
- Alert protocol (*see* Secure Socket Layer) ALOHA, 343
 - pure, 343
 - slotted, 345
 - throughput of, 343, 346
- AMI, 14
- ANSI, 211
- Application layer, 195
- Area Border Router (ABR), 614
- ARP, 547, 564–570
 - layered architecture, 565
 - operation, 565
 - packet format, 567
 - packet types in, 567
 - source hardware address in, 568
 - source protocol address in, 568
 - target hardware address in, 568
 - target protocol address in, 568
- AS external LSA (*see* Link State Advertisement) AS path attribute (*see* Border Gateway Protocol) ASCII, 3–5, 285
 - control characters/symbols, 4, 284
- ASK (Amplitude Shift Keying), 103
 - bandwidth of, 103
- Assured forwarding, 802
- Asynchronous communication, 29
- Asynchronous Balanced Mode (ABM), 292
 - operation, 308
- Asynchronous Disconnected Mode (ADM), 292
- Asynchronous modem, 116
- Asynchronous Response Mode (ARM), 292
 - operation, 305–307
- Asynchronous transfer mode (*see* ATM) Asynchronous transmission, 9–10
- ATM, 526–542
 - adaptation layer (AAL) in, 534–542
 - attenuation constant, 36 (*see also* Transmission line) CBR traffic, 535
 - Cell-Delay Variation (CDV) in, 529
 - cell delineation in, 532
 - cell Format of, 533
 - Cell-Loss Ratio (CLR) in, 529
 - Convergence Sublayer (CS) of, 537, 538, 539, 541
 - HEC generation and verification in, 531
 - layer, 534, 535
 - layered architecture of, 530
 - physical layer of, 530
 - PMD sublayer of, 531
 - Segmentation and Reassembly sublayer (SAR) of, 537, 538, 539, 542
 - TC sublayer of, 531, 532
 - traffic classification in, 536

- VBR traffic, 536
- Virtual Channel Connection (VCC), 528
- Virtual Channel Identifier (VCI), 528, 533
- Virtual Path Connection (VPC), 529
- Virtual Path Identifier (VPI), 528, 533
- Attenuation distortion (*see* Distortion)
- Authentication, 696, 705–708
 - header, 577
 - protocols, 588
 - reflection attack in, 707
- Authentication using:
 - message digest, 708
 - public key, 708
 - secret key, 705
 - third party, 707
- Auto-negotiation, 361, 369, 377
 - priority levels in, 373
- Autonomous System (AS), 614
- Autonomous System Border Router (ASBR), 614
- Average data rate, 778

B8ZS line code, 15

- Backup Designated Router (BDR) (*see* Hello protocol)
- Balanced pair, 43
 - cable, 43
 - characteristic impedance of, 44
 - loading, 44
 - transmission characteristics of, 44
- Bandwidth, 784
- Base, 64, 747
- Baseband transmission, 21
- Basic Service Set (BSS), 413
 - independent, 413
 - infrastructure, 413
- Baud, 20
- Baudot teletype code, 3
- BCC (Block Check Characters), 283
- BCDIC, 3
- BER (*see* Bit Error Rate)
- Binary exponential backoff, 354, 420, 423
- Biphase-M line code, 14
- Biphase-S line code, 14
- Bipolar signal, 7
- BISYNC, 281–290
 - error control in, 285
 - flow control in, 285
 - frame format of, 282
 - limitations of, 290

- point-to-multipoint communication in, 282, 288
- point-to-point communication in, 282, 287
- polling and selecting stations in, 282
- transparency in, 286
- types of stations in, 282

Bit, 2

Bit Error Rate (BER), 159

Bit interleaved data multiplexer, 143

Bit-map statistical multiplexer, 145

- frame format of, 145
- line utilization efficiency of, 147

Bit rate, 7

Block codes, 15–16

Block parity, 172

Blocking, 207, 489

Boltzmann’s constant, 88

Border Gateway Protocol (BGP), 630–636

- AS path attribute in, 632
- community attribute in, 635
- local preference attribute in, 632
- MED attribute in, 633
- message types, 633
- next hop attribute in, 634
- operation, 633
- origin attribute in, 634
- path attributes, 632

BOOTP (Bootstrapping Protocol), 732–734

- message format of, 733

BPDU, 448

- format of, 449

BPSK, 105

- bandwidth of, 105
- differential, 109

Bridge, 441

- layered architecture of, 441
- remote, 462
- types of, 442 (*see also* Transparent bridge, Source Routing bridge)

Broadcast mode, 325

Broadcast storm, 448

BSC (*see* BISYNC) Bus topology, 325

- dual bus in, 326
- head end of, 325

Burst error, 154

- length of, 154

Bypass relay, 327, 383

Byte, 5

- interleaved data multiplexer, 143

CAP (*see* ADSL) Capacitive coupling, 41
CAT3 cable, 45
CAT4 cable, 45
CAT5 cable, 45
CE, 787
Cell Delay Variation (CDV) (*see* ATM) Cell Loss Ratio (CLR) (*see* ATM) Certification authority, 713
 hierarchy, 713
 regional, 713 (*see also* Digital certificates) CHAP, 587
Character Error Rate (CER), 159
Characteristic impedance, 36 (*see also* Transmission line) Checksum, 161–165
Chromatic dispersion, 50 (*see also* Optical fibre) CIDR, 563
Cipher Block Chaining (*see* DES-CBC) Cipher change spec protocol (*see* Secure Socket Layer) Ciphertext, 697
CIR (*see* Frame relay) Circuit switched data network, 471–473
 delays in, 473
 service features of, 473
Claim token frame, 386
Class-based queuing, 771
Class-based weighted fair queuing, 776
Class selector PHB, 803
Client-server model, 727
CLNP (ISO), 577
 types of IPDUs in, 578
CLNS (Connectionless-mode Network Service), 488
 (*see also* Network service) Clock, 10,
Coaxial cable, 46
 attenuation constant of, 46
 attenuation in, 47
 characteristic impedance of, 46
 ITU-T recommendations of, 47
 phase constant of, 46
Code word, 156
 algebraic representation of, 167
 weight of, 156
Collision domain, 356, 441, 463
Collision window, 350
Combined station in HDLC, 291
Communication
 hierarchical, 198
 meaningful, 187
 peer-to-peer, 199
Community attribute (*see* Border Gateway Protocol) Compondors, 89
Compressor, 89
Computer network, 186
 architecture of, 190 (*see also* Layered architecture) components of, 190
Concatenation, 208

- Confirmed service, 203
- Congestion, 767
- Congestion avoidance, 689
- Congestion window (cwnd), 687
- Connection, 200
 - multiplexing of, 206
- Connection mode, 204, 332
- Connectionless mode, 205, 545, 578
- CONS (Connection-Oriented Network Service), 488
 - (*see also* Network service) Constant Bit Rate (CBR) traffic (*see* ATM) Content errors, 153
- Contention access, 342
 - throughput of, 348
- Controlled flooding, 608, 622
- Convergence sublayer (*see* ATM) Convolutional codes, 177
 - Viterbi decoder for, 179, 376
- Convolutional encoder, 177
 - rate 1/2, 177
 - state transition diagram of, 177
 - trellis encoder, 178, 376
- Core-Based Tree (CBT) multicasting, 648–650
 - join message in, 649
 - Rendezvous Point (RP) in, 649
- CRC (*see* Cyclic Redundancy Check) Crosstalk, 41
- Cryptography algorithms, 697
- CSMA (Carrier Sense Multiple Access), 346

1-persistent, 347

- non-persistent, 347
- p*-persistent, 347
- CSMA/CA, 420
- CSMA/CD, 348 (*see also* ethernet) media access control in, 348
- CSPDN, 237
- CTS (Clear To Send), 224, 416
- Custom queuing, 771
- Cut-through switch, 465
- Cyclic Redundancy Check (CRC), 165–172
 - undetected errors in, 169–172

- D-channel, 93, 96, 313
- Data block, 10
- Data communication, 29
- Data compression, 21–29, 125
 - Huffman algorithm for, 26
 - Lempel-Ziv-Welch, 125
 - Shannon-Fano algorithm for, 26

- Data Encryption Standard (DES), 678
- Data link connection, 243
- Data link layer, 197, 243
 - error control in, 257–272
 - flow control in, 248–257 (*see also* Stop-and wait flow control, Sliding window flow control)
 - management, 274
 - protocol data unit (DL-PDU), 244
 - service data unit (DL-SDU), 244
 - service primitives of, 277
 - service provided by, 244
- Data link protocol, 245
 - bit-oriented, 248
 - byte-oriented, 248
- Data multiplexer, 140–150
 - types of, 141 (*see also* Frequency division data multiplexer, Time division data multiplexer, Stat Mux)
 - Data representation, 2
- Data transmission, 5–11
- Database description packet, 615, 621
 - format of, 621 (*see also* OSPF) Datagram, 477
- Datagram switching, 477–481
 - congestion in, 480
 - deadlocks in, 480
 - errors in, 480
 - forwarding tables in, 478
 - routing in (*see also* Routing), 479
 - service features of, 481
- DB-25 connector, 222
- DCE, 219, 496, 519
- DCF with RTS/CTS, 420
- Decibel:
 - dB, 42
 - dBm, 42
 - dBmp, 88
 - dBW, 42
- DECnet, 208
- Default routes (*see* routing)
- Deficit weighted round-robin (DWRR), 773
- DES-CBC, 699, 703
- Descrambler, 122
- Designated Router (DR) (*see* Hello protocol), DHCP, 735–736
 - IP address assignment modes, 735
 - message format, 735
 - message types, 735
- Dibit, 107
- Differential Manchester code, 14
- Differentiated service, 548, 589, 799
 - Code Point (DSCP), 800
 - DiffServ, 791
 - domain, 590, 803
- Diffie-Hellman secret key exchange, 699

- DIFS, 419
- Digital
 - certificates, 712–714
 - certification authority for, 713
 - message digest, 710
 - modulation, 102
 - private key, 709
 - revocation list, 714
 - signature, 709
 - third party, 711
 - X.509, 713
- Digital Subscriber Line (DSL) systems, 136–140
- Dijkstra’s algorithm, 609–613, 628
- Discrete Multitone Modulation (DMT), 138
- Distance vector, 596
- Distance vector algorithm, 598–604
 - convergence in, 599
 - count-to-infinity problem in, 599
 - hold-down timer in, 601
 - split horizon method of, 600
 - split horizon with poisonous reverse method of, 601
- Distinct code, 25
- Distortion
 - attenuation, 39
 - linear, 38
 - non-linear, 86
 - phase, 39
- Distortionless transmission, 38
- Distributed Bellman-Ford algorithm, 599
- Distributed Coordination Function (DCF), 414
- Distribution system, 413
- DLCI (*see* Frame relay) DLE, 285
- DMT (*see* ADSL) DNA, 193, 209
- DNIC, 515
- DNS (*see* Domain Name System) Domain, 729
- Domain name, 729
- Domain Name System (DNS), 728–731
 - FQDN, 730
 - hierarchical naming in, 728
 - label, 729
 - message, 731
 - naming tree structure in, 730
 - protocol, 731
- Downward multiplexing, 207
- Drop precedence, 802
- Dropout, 92
- Drop profile, 785
- DSCP, 548
- DSR, 224

DTE (Data Terminal Equipment), 219, 496, 517
DTR, 224
DVMRP, 652
Dynamic routing, 479

E1, 76

EBCDIC, 3, 285
EBGP, 630
Echo, 84, 571

- cancellation in
 - ADSL, 137
 - ethernet, 376
 - modems, 116
- cancellor, 84, 122
- suppressor, 84

ECN-echo flag, 788
ECT, 787
EIA, 211
EIA-232-D digital interface, 219

- electrical specifications of, 222
- functional specifications of, 223
- interchange circuits of, 223
- limitations of, 233
- mechanical specifications of, 222
- procedural specifications of, 227–230
- standard full duplex configuration, 230
- three wire interconnection, 230, 231

EIA-422-A, 234, 236
EIA-423-A, 234, 236
EIA-449, 234–237

- electrical specifications of, 236
- functional specifications of, 237
- mechanical specifications of, 234

EIA-530, 237
E-LSP, 804

E-mail (Electronic mail) system, 742–748 (*see also* SMTP, POP3, IMAP), components of, 742

- format, 744
- mail addresses in, 743
- mail box, 743
- Mail Transfer Agent (MTA), 743
- Mail User Agent (MUA), 743
- spool-file in, 743

Empty slot media access control, 328
Encapsulating-Security-Payload (ESP) header, 576
End system, 186
Entropy, 23

- Equalizer
 - adaptive, 57, 59
 - attenuation, 39
 - compromise, 121
 - frequency domain, 39
 - group delay, 39
 - transversal filter (*see also* Intersymbol interference), 57
- Error correction, 157
 - Forward (FEC), 157, 172
 - Reverse (REC), 157, 181
- Error detection, 156
 - checksum for, 161
 - CRC for, 165
 - parity checking for, 160
 - transport protocol checksum, 161
- Error syndrome, 174
- ES Hello messages (ESH), 628
- ES-IS protocol, 627, 628
- ESS, 413
- Ethernet, 348 (*see also* IEEE 802.3) 10Base2, 360
 - 10Base5, 359
 - 10BaseF, 361
 - 10BaseFB, 361
 - 10BaseFL, 361
 - 10BaseFP, 361
 - 10BaseT, 360
 - 10Broad36, 360
 - 100BaseT2, 367
 - 100BaseT4, 363
 - 100BaseFX, 369
 - 100BaseTX, 369
 - 100BaseX, 361
 - 1000BaseCX, 374, 377
 - 1000BaseLX, 374, 377
 - 1000BaseSX, 374, 377
 - 1000BaseT, 374, 376

1000BaseX, 374, 377

- auto-negotiation in, 362
- bus topology of, 354
- hub, 355
- link segment in, 356
- maximum cable segment of, 350
- physical layer of, 358, 362
- point-to-point topology of, 355
- repeater, 356

- star topology of, 355
- switches, 463–465
- thick, 359
- thin, 360
- truncated binary exponential backoff mechanism
 - of, 354
- Ethernet (DIX), 348
 - frame format, 352
- Even parity, 160
- Expander, 89
- Expedited forwarding, 801
- Explicit congestion notification (ECN), 786
- Exterior Gateway Protocol (EGP), 630
- Eye pattern, 61

- Fair queuing, 773
- Far-end crosstalk (FEXT), 41
- Fast ethernet, 361
 - MII sublayer, 362
 - PCS sublayer, 362
 - PMA sublayer, 362
 - PMD sublayer, 363
- Fast link pulses, 371
- Fast select (*see* X.25) FDDI (Fibre Distributed Data Interface), 402–409
 - asynchronous service of, 404
 - dual-attached concentrator (DAC), 403
 - dual-attachment station (DAS), 403
 - frame fragmentation in, 405
 - MAC frame format, 407
 - media access control, 405
 - number of stations of, 409
 - physical specifications of, 409
 - primary ring, 402
 - priority management in, 406
 - ring size, 409
 - secondary ring, 402
 - single-attached concentrator (SAC), 403
 - single-attachment station (SAS), 403
 - synchronous service of, 404
 - Target Token Rotation Time (TTRT) in, 406
 - Token Holding Time (THT) in, 406
- FDM (*see* Multiplexing) **Finish number, 774**
- Firewall, 722–723
 - filter based, 722
 - proxy based, 723
- Flag, 10, 294
 - Flow control, 248–257, 370 (*see also* Stop-and-wait flow control, sliding window flow control)
 - Flow integrity errors, 154

- Forward error correction, 172–181
- Forwarding delay, 764
- Forwarding Equivalence Class (FEC) (*see* MPLS) Forwarding table, 478, 593
- Fourier series, 17–19
- Fragment identifier, 548
- Fragment offset, 548
- Fragment reassembly, 547, 551
- Fragmentation, 424, 546, 550
- Frame, 11, 244
 - alignment word, 75
 - data, 282, 283
 - design, 246
 - Error Rate (FER), 159
 - formats, 246
 - header, 244
 - supervisory, 282, 283
 - trailer, 244
- Frame relay, 518, 527
 - Backward Explicit Congestion Notification (BECN) in, 523, 525
 - committed burst size in, 522
 - Committed Information Rate (CIR) in, 522
 - congestion control in, 521
 - Data Link Connection Identifier (DLCI), 519, 524
 - Discard Eligibility (DE) bit in, 522, 525
 - excess burst size in, 522
 - Forward Explicit Congestion Notification (FECN) in, 523, 525
 - frame format of, 524
 - IP encapsulation in, 526
 - layered architecture of, 520
 - physical layer interfaces in, 521
 - PVC in, 520
 - services, 520
 - SVC in, 520
- Frequency division data multiplexer, 142
 - limitations of, 143
 - multidrop operation of, 142
- Frequency shift, 92
- Frequency Shift Keying (FSK), 104
 - bandwidth of, 104
 - frequency spectrum of, 104
- Frequency spectrum, 16–19
- FTAM, 743
- FTP (File Transfer Protocol), 740–742
 - Data Transfer Process (DTP) of, 741
 - operation, 741
 - Protocol Interpreter (PI) of, 741
- Full duplex (FDX), 30

- G.992.1 (G.dmt), 139
- G.992.2 (G-lite), 139
- Gain hit, 91
- Galvanic coupling, 41
- General Format Identifier (GFI), 500
- Generating polynomial, 118, 167
 - CRC-12, 131
 - CRC-16, 132
 - CRC-32, 132
- Gigabit ethernet, 374
 - auto-negotiation in, 378
 - carrier extension in, 375
 - frame bursting in, 375
- Globally scoped multicast addresses (*see* Multicasting) Go-back— N (*see* Reverse error correction) Gray code, 106
- Group delay, 39
 - equalizer, 39

- Half duplex (HDX), 30
- Hamming code, 174
- Hamming distance, 156
- Handshake protocol (SSL), 717
- Harmonics, 18
- HDB3, 14
- HDLC, 290–309 (High Level Data Link Control)
 - BISYNC compared, 309
 - command, 291
 - error control, 294
 - extended addressing in, 308
 - extended control field, 309
 - flow control in, 293
 - frame formats in, 294–296
 - modes of operation, 292
 - mode transition diagram of, 293
 - poll/final bit in, 296, 299
 - response, 291
 - transparency in, 300
 - types of stations in, 291
- HDSL, 140
- Head end, 325
- Header, 199, 246
- Hello interval, 619
- Hello packet, 615
- Hello protocol, 617–619
 - Backup Designated Router (BDR) in, 618
 - Designated Router (DR) in, 618
 - packet format, 618
- HF, 51, 52, 83
- Hierarchical addressing, 552

- Hierarchical naming system, 728
- Hierarchical routing, 614
- HTML, 754
- HTTP, 756–759
 - architecture of, 756
 - cookies in, 758
 - request messages of, 756
 - response messages of, 757
- Hub (LAN), 328, 355, 357
- Huffman code, 26
- Hybrid, 70
- Hypertext Markup Language (*see* HTML) Hypertext Transfer Protocol (*see* HTTP) IA2, 3
- IA5, 3
- IBGP, 630
- ICMP, 549, 571–575
 - message types, 572
- IDRP, 627
- IEEE, 210
- IEEE 802.2, sublayer, 565, 568
- IEEE 802.3, 348
 - frame format of, 351
- IEEE 802.4,
 - frame format of, 396
 - types of frames of, 396
- IEEE 802.5,
 - frame format of, 385
 - functional address in, 388
 - MAC addresses, 388
 - priority bits in, 386
 - routing Information Field of, 388
 - Routing Information Indicator (RII) bit in, 388
- IEEE 802.11, 415
 - address fields in, 429
 - CF-End frame in, 426
 - frame format, 428
 - frame types, 426
- IGMP Version 1, 655
 - group address in, 656
 - message format of, 655
- IGMP Version 2, 656
 - message format of, 656
 - message types, 657
- IGMP Version 3, 657
- Impulse
 - noise, 91
 - response, 40
- Individual/group address, 388
- Inbound carrier, 326
- Inductive coupling, 41

- Information, 22
 - entropy of, 23
 - redundancy in, 23–25
- Information transfer frame of HDLC, 294, 295
- Initialization mode of HDLC, 293
- Integrated service (IntServ), 790, 791
- Interchange circuits (*see* EIA-232-D digital interface)
- Interior Gateway Protocol (IGP), 629
- Interleaved codes, 176
 - depth of, 176
- Intermediate system, 490
- Intermodulation products, 86
- International transit exchange (*see* Telephone network)
- Internet, 193
 - Architecture Board (IAB), 210
 - Assigned Number Authority (IANA), 755
 - format of, 579
 - Group Management Protocol (IGMP), 655–657
 - Message Access Protocol (IMAP4), 748
 - Network Information Centre (InterNIC), 553
 - Policy Registration Authority (IPRA), 713
 - Protocol (IP), 546
 - Protocol (ISO 8473), 578–582
 - Protocol Data Unit (IPDU), 578
 - Research Taskforce (IRTF), 210
 - Society (ISOC), 210
 - types of, 578
- Internetwork, 491
- Internetworking Unit (IWU), 491
- Intersymbol interference, 40, 52–56
 - equalization for, 56
- IP control Protocol (IPCP), 588
- IP Security (IPSec), 718–721
 - authentication header, 719, 720
 - Encapsulating Security Payload (ESP), 719, 720
 - Security Association (SA), 719
 - Security Parameter Index (SPI), 719, 720, 721
- IPv4, 547
 - addressing, 553
 - CIDR, 563
 - Class A address in, 554
 - Class B address in, 554
 - Class C address in, 554
 - Class D address in, 554, 654
 - Class E address in, 554
 - class of service in, 590
 - classful addressing in, 552
 - classless addressing in, 563
 - differentiated service, 589
 - directed broadcast address in, 556
 - host number in, 553

- local broadcast address in, 556
- loopback address in, 557
- network number in, 552
- network prefix in, 564
- packet format, 547
- packet fragmentation, 550
- Per Hop Behaviour (PHB) in, 589
- Quality of Service (QoS), 588
- subnet mask, 559
- subnetting, 557–564
- supernetting, 563
- VLSM, 562–564
- IPv6, 547, 574–578
 - address notation, 577
 - comparison with IPv4, 577
 - extension headers in, 576
 - flow label in, 575
 - packet format, 575
 - traffic class in, 575
- IS 7496, 194
- IS Hello messages (ISH), 628
- ISDN, 93–97, 313
 - B channel in, 93, 96
 - Basic Rate Access (BRA) in, 93, 94
 - BRA frame, 96
 - D channel in, 93, 96, 313
 - H channel in, 94
 - NT1 of, 94, 95
 - NT2 of, 95
 - Packet handler in, 315
 - Primary Rate Access (PRA) in, 93, 94
 - R interface in, 95
 - S interface in, 96
 - T interface in, 96
 - Terminal Adapter (TA) of, 94
 - U interface in, 96
- IS-IS protocol, 627, 628
- ISAKMP (Internet Security Association Key Management Protocol), 719
- ISO, 210
- ISO 2110, 219, 221
- ISO addressing scheme, 628
- ITU-T, 210

- Jacobson's algorithm, 686
 - jam signal, 348
 - join message (*see* Core-based tree) justification bit, 77
 - justification control bit, 77
- Jitter, 765

- Karn's algorithm, 686

- LAP-B (Link Access Procedure-Balanced), 310
 - U-frame commands and responses of, 311
- LAP-D, 313, 519, 520
 - frame format, 313–314
- LAP-F, 518, 520, 526
- LAP-M, 125, 312
- Layer-2, 764
- Layer-3, 764
- Layer entity, 192
- Layered architecture, 192
- LCP, 583–586
 - packet format of, 586
 - packet types in, 586
- LDP (*see* MPLS) Leaky bucket, 779
- Leave latency, 656
- Line codes, 11–16
- Line driver, 136
- Linear distortion (*see* Distortion) Link Code Word (LCW), 372
 - format of, 373, 378
- Link Control Protocol (LCP) (*see* PPP) Link state, 608
- Link State Advertisement (LSA), 608, 624
 - age, 626
 - AS external LSA, 625
 - format of, 626
 - network LSA, 625
 - router LSA, 624
 - summary LSA, 625
 - types, 624
- Link state database synchronization, 617
- Link state request packet, 615
- Link state routing, 608–613
 - Dijkstra's algorithm for, 609–613
- Link state update packet, 615, 622
- Link test pulses, 361
- LLC (Logical Link Control), 329
 - classes of stations in, 332
 - commands and responses, 336
 - PDU format, 330, 334
 - C/R bit of DSAP, 335
 - DSAP in, 334
 - I/G bit of DSAP, 334
 - SSAP in, 334
 - User (U) bit of DSAP, 334
 - procedures, 336
 - service types of, 331
- L-LSP, 805

- Loading coil, 44
- Local Area Network (LAN), 322
 - backbone LAN, 323
 - back-end LAN, 323
 - coaxial cable for, 338
 - IEEE standards of, 330
 - layered architecture of, 329
 - media access control in, 328
 - optical fibre cable for, 339
 - topology of, 324
 - twisted copper pair cable for, 338
 - wireless, 412
- Local preference attribute of BGP (*see* Border Gateway Protocol),
- Logical channel, 498 (*see also* X.25) Logical topology, 328
- Loopback address, 557
- LRC (Longitudinal Redundancy Check), 172
- LSA header, 622
- LSP (*see* MPLS) M.1020 (ITU-T), 99
- M.1025 (ITU-T), 100
- MAC (Media Access Control) sublayer, 329
 - PDU, 330
 - protocol, 337
 - service, 336
- Magic number, 587
- Mail exchange (MX), 745
- Management Information Base (MIB), 750, 751
 - management process, 749, 750
 - Manchester code, 13 (*see also* Differential Manchester code) Mark, 12
- Master station, 282
- Maximum burst size, 778, 781
- Maximum likelihood decoding, 157
- Maximum Segment Size (MSS) (*see* TCP) MD5 algorithm, 703
 - keyed, 704
- Media access control, 328 (*see also* Local area network)
- Message Authentication Code (MAC), 702
- Message Digest (MD), 702, 703
- Message Integrity Code (MIC), 702
- Message switching, 474–476
 - delivery delay in, 475
 - service features of, 475
- MIB (*see* Network Management System) MIME, 744, 746–748
 - content transfer encoding in, 747
 - content types of, 747
 - header, 746
- Minimal prefix code, 25
- MLT, 3
- Modal dispersion, 48 (*see also* Optical fibre) Modem, 21, 98, 113–136

2-wire, 115

4-wire, 115

- asynchronous, 116
- baseband, 136
- block schematic of, 119
- data compression in, 125
- error control in, 125
- full duplex, 115
- group band, 136
- half duplex, 115
- limited distance, 135
- scrambler and descrambler in, 117
- secondary channel in, 123
- synchronous, 116
- test loops in, 123
 - V.21 (ITU-T), 125
 - V.22 (ITU-T), 125
 - V.22*bis* (ITU-T), 126
 - V.23 (ITU-T), 127
 - V.26 (ITU-T), 128
 - V.26*bis* (ITU-T), 128
 - V.26*ter* (ITU-T), 128
 - V.27 (ITU-T), 129
 - V.27*bis* (ITU-T), 130
 - V.27*ter* (ITU-T), 130
 - V.29 (ITU-T), 130
 - V.32 (ITU-T), 132
 - V.33 (ITU-T), 134
 - V.34 (ITU-T), 135
 - V.90 (ITU-T), 135
- Monitoring station, 326
- MOSPF, 652
- MPLS, 657–661, 804
 - Constraint Based Routing LDP (CR-LDP), 662
 - Forwarding Equivalence Class (FEC) in, 661
 - Header, 661
 - Label, 658
 - Label Distribution Protocol (LDP), 658, 661, 662
 - Label Edge Router (LER), 659
 - Label stacking, 659
 - Label Switched Path (LSP), 659
 - Label Switching Router (LSR), 659
 - Traffic engineering using, 661
 - tunnels, 663

- Multi-Exit Discriminator (MED) attribute (*see* Border Gateway Protocol)
- Multistation Attachment Unit (MSAU), 384
- Multicast group, 641
 - address, 653
- Multicasting, 640–642
 - controlled flooding for, 642
 - Core Based Tree (CBT) in, 648–650
 - globally scoped addresses for, 654
 - grafting in, 648
 - IP addressing for, 653
 - limited scope addresses for, 654
 - network addresses for multicast routers on local networks, 653
 - pruning in, 647
 - Reverse Path Forwarding (RPF) in, 644–648
 - spanning tree for, 642, 643
- Multicasting on LAN segments, 654
- Multicasting protocols, 650–653
- Multilevel modulation, 106
- Multilink Procedure (MLP), 311
- Multiple character statistical multiplexer, 145
 - frame format of, 145
 - line utilization efficiency of, 147
- Multiplexer (*see* Data multiplexer)
- Multiplexing, 71–74
 - Frequency Division (FDM), 71
 - hierarchy, 72
 - time division, 73
- Multiprotocol Label Switching (*see* MPLS)
- Multipurpose Internet Mail Extensions (*see* MIME)
- Multistation Attachment Unit (MSAU), 384

- Nagle's algorithm, 685
- NAK (Negative acknowledgement), 285
- NCP, 583, 589
- Network Allocation Vector (NAV), 422
- Network architecture, 190, 193
 - models, 146, 190
 - standardization of, 193
- Network connection, 489
- Network Control Protocol (NCP) (*see* PPP)
- Network layer, 196, 485
 - functions of, 489
 - purpose of, 485
- Network LSA (*see* Link state advertisement)
- Network Management System (NMS), 749–750 (*see also* SNMP) agent process, 750
- Network number, 554
- Network prefix (*see* IPv4)
- Network service, 487
 - access point (N-SAP), 488
 - data units (N-SDU), 488

- features of, 488
- types of, 488 (*see also* CONS, CLNS) NEXT (Near-End Crosstalk), 41

Next-page, 372

- format of, 373

NNI, 528

Noise, 85

- impulse, 91
- intermodulation, 86
- psophometric weighting of, 88
- shot, 85, 88
- thermal, 85, 87

Non-confirmed service, 203

Normal Disconnected Mode (NDM) of HDLC, 292

Normal Link Pulse (NLP), 361

Normal Response Mode (NRM) of HDLC, 292

- operation, 302–305

NRZ (Non Return to Zero) codes NRZ-I, 12

- NRZ-L, 12
- NRZ-M, 12
- NRZ-S, 12
- NTN, 515

Null modem, 231–233

NVT (*see* Telnet) Nyquist criterion, 52, 54

- raised cosine response in, 54
- $\sin(pt/T)/(pt/T)$ response in, 53

Nyquist's theorem, 19

One-Way communication (OW), 30

Optical fibre, 47–51

- advantages of, 51
- attenuation of, 50
- chromatic dispersion in, 50
- graded index, 49
- modal dispersion in, 48
- monomode, 49
- multimode, 47
- Rayleigh scattering in, 50
- step index, 49

OSI (Open System Interconnection), 194

- layered architecture of, 194
- reference model of, 194

OSI routing protocols, 627

- area in, 628
- domain in, 628
- level-1 routing in, 628

OSPF, 613–617 (*see also* Hello protocol) adjacency in, 614, 617

- area, 614
- database synchronization in, 620

- LSA (Link State Advertisement) in,
 - format of, 626
 - types of, 624
- packet format, 615
- packet types, 615
- Outbound carrier, 325

- Packet Assembler and Disassembler (*see* PAD)
- Packet switching, 476
 - delivery delay in, 476 (*see also* Datagram switching, virtual circuit switching)
- Packet Type Identifier (PTI)
- PAD, 497, 516–518
 - commands, 517
 - messages, 517
 - parameters, 516, 518
 - service signals, 516
- PAM 5, 368, 376
- PAP, 587
- Parallel transmission, 5
- Parity bit, 160
 - check (*see also* Block parity) even, 160
 - odd, 160
- PATH error, 794
- PATH message, 793
- PATH tear-down , 794
- Path vector, 596, 601
- Path vector routing protocol, 631
- PCM (Pulse code Modulation), 74

30 channel, 75

- bit rate of, 76
- frame, 75
- time slot, 75
- coding, 75
- quantization, 75
 - m-law, 75
 - A law, 75
 - error, 75
 - non-uniform, 75
- Peak data rate, 779
- Per Hop Behaviour (PHB), 589
 - assured forwarding, 589
 - expedited forwarding, 589
- Permanent Virtual Circuit (PVC), 498 (*see also* X.25)
- Phase coherent carrier, 108
- Phase constant, 37 (*see also* Transmission line)
- Phase delay, 37

- Phase distortion (*see* Distortion) Phase hit, 92
- Phase jitter, 92
- Phase Shift Keying (PSK), 105
 - differential PSK, 109
 - M-ary, 112 (*see also* 4 PSK, 8 PSK, BPSK)
- Phase velocity, 37
- Physical connection, 214
 - activation/deactivation of, 215
 - point-to-multipoint, 214
 - point-to-point, 214
- Physical layer, 197, 213
 - control signals, 218
 - electrical specifications of, 219
 - functional specifications of, 219
 - mechanical specifications of, 219
 - procedural specifications of, 219
 - protocol, 214
 - relaying function in, 216
 - service primitives of, 215
 - service provided by, 215
 - specifications, 218
- Physical Media Dependent (PMD) sublayer, 359, 363
- PIFS, 419
- PIM, 650–652
 - dense mode, 650
 - extensions, 651
 - sparse mode, 650
- PING, 571
- Plaintext, 697
- Plesiochronous digital hierarchy, 76
- Point Coordination Function (PCF), 414
- Point Coordinator (PC) in, 414
- Point-to-multipoint, 31, 281
- Point-to-point, 30, 281
 - protocol (*see* PPP)
- Polar signal, 6
- Policy control, 792
- Polling and selecting, 282
- POP3, 748
- Post Office Protocol (*see* POP3)
- Power spectral density, 18
- PPP, 582–586
 - frame format of, 583
 - layered architecture of, 582
 - operation, 584
 - physical layer for, 583
- Preamble, 351
- Prefix code, 25
- Presentation layer, 195
- Primary parameters (*see* Transmission line)
- Primary station in HDLC, 291
- Priority queuing, 770

- Private key, 699
- Propagation constant, 36 (*see also* Transmission line) Propagation delay, 765
- Protocol, 199
 - control information (PCI), 199
 - data unit (PDU), 199
- Provider initiated service, 203, 204
- Proxy based firewall (*see* Firewall) PSK (*see* Phase Shift Keying) Psophometric weighting, 88
- PSTN (*see also* Telephone network), 65
- Public key, 700
 - encryption algorithm, 700–702
- Pulse Code Modulation (*see* PCM) Pure ALOHA, 343
 - throughput of, 343–345

- Q.921 (ITU-T), 519
- Q.922 (ITU-T), 518
- Q.933 (ITU-T), 519
- QAM (Quadrature amplitude modulation) (*see* 16 QAM)
- Quality of service (QoS), 196, 548, 762
- Queuing, 768
- Queuing delay, 764, 782

- Radio systems, 83
- Raised cosine response, 54
- Random Early Detection (RED), 784
- RARP, 547, 732
- REC (*see* Error) Record route, 552
- Reflection attack, 707
- Regenerative receiver, 60
- Register insertion ring, 328
- REJ— N , 264
- Remodulator, 326
- Remote bridge, 462
 - layered architecture, 463
- Rendezvous-point (*see* Core based tree multicasting) Repeater, 325
 - local, 356
 - remote, 356
- Replay attack, 705
- Reservation styles, 799
- Residual Error Rate (RER), 257
- Resource Reservation Protocol (RSVP), 662, 793
- RESV confirm, 794
- RESV error, 794
- RESV request message, 794
- RESV tear-down, 794
- Reverse charging in X.25,
- Reverse error correction methods

- Go-back— N , 182
 - selective retransmission, 182
 - stop and wait, 181
- Reverse Path Forwarding (RPF), 644–648
 - grafting in, 648
 - pruning in, 647
- Ring indicator circuit, 226
- Ring Interface Unit (RIU), 383 (*see also* Token passing on ring)
- Ring topology, 326,
 - RIU in, 326
- RIPv1, 602–606
 - Address Family Identifier in, 603
 - forwarding table, 605
 - limitations of, 605
 - packet format, 602
 - route objects, 604
 - types of packets in, 604
- RIPv2, 606–608
 - Address Family Identifier in, 607
 - authentication in, 608
 - next hop field in, 607
 - packet format, 606
 - route tag field in, 607
 - subnet mask field in, 607
- RNR— N , 252
- Router LSA (*see* Link State Advertisement) Routing, 593–596
 - default routes in, 595
 - domain, 594
 - dynamic, 596
 - hierarchical, 614
 - source, 442, 553, 575
 - static, 596
- RPF (*see* multicasting) RR— N , 252
- RS-232-C (*see* EIA-232-D) RSA public key encryption algorithm, 701
- Rspec, 792
- RSVP, 793
- RSVP messages, 793
- RTS (Request to send), 224, 416
- RTT, 685
- RZ (Return to Zero) codes biphase-M code, 14
 - biphase-S code, 14
 - differential Manchester code, 14
 - Manchester code, 13

- Sampling theorem, 73
- SAPI, 314
- Scheduling , 769
- Scrambler, 117, 120, 154

- SCU, 216
- SDH, 77
- SDSL, 140
- Secondary channel, 123
- Secondary parameters (*see* Transmission line) Secondary station in HDLC, 291
- Secret key encryption, 697–699
- Secret key exchange, 699
- Secure Socket Layer (SSL), 714–718
 - alert protocol of, 718
 - architecture, 715
 - cipher change spec protocol of, 718
 - connection, 715
 - handshake protocol of, 717
 - record protocol of, 716
 - session, 715
- Security Association (SA), 719
- Security Parameter Index (SPI), 719, 720, 721
- Security requirements, 695
 - authentication in, 696
 - confidentiality in, 696
 - message integrity in, 696, 709
 - non-repudiation in, 696
- Segmentation and Reassembly (SAR) sublayer (*see* ATM)
- Segmenting, 207, 489
- Selective retransmission, (*see* Reverse error correction)
- Serial transmission, 6–8
- Serialization delay, 765
- Service Access Point (SAP), 200
- Service Data Unit (SDU), 201
- Service interface, 201
 - primitives, 202
- Service level agreement (SLA), 763
- Service types of, 203
- Session, 196
- Session layer, 196
- Set successor frame, 397, 399
- Shannon-Fano code, 26
- Shannon's theorem, 19
- Shaping 767, 778
- SHF, 51, 52, 83
- Shielded twisted pair, 45
- SIFS, 419
- Signal encoding, 11–16
 - Non Return to Zero (NRZ) codes (*see also* NRZ), 12
 - Return to Zero (RZ) codes (*see also* RZ), 13
- Signal impairments, 91
- Signal to Noise Ratio (SNR), 88
 - weighted, 88 (*see also* Noise) Silly window syndrome, 685

- Simple Mail Transfer Protocol (*see* SMTP)
- Simple Network Management Protocol (*see* SNMP)
- Simplex transmission, 30
- Single frequency interference, 92
- Single queue, 769
- Single Link Procedure (SLP), 311
 - frame format, 250
- Skin effect, 38
- Slave station, 282
- Sliding window, 253
 - at receiving end, 254
 - at transmitting end, 253
 - error control in, 263–272
- Sliding window flow control, 251
 - link utilization in, 254–257, 267
 - sequence numbering of frames in, 272
- Slotted ALOHA, 345
 - throughput of, 346
- Slow start, 687–689
 - threshold (ssthresh), 689
- Smoothed RTT (SRTT), 686
- SMTP, 744–746
 - architecture, 744
 - commands, 745
 - responses, 745
- SNA (System Network Architecture), 193, 208, 209
- SNMP, 748–754
 - architecture, 750
 - operation, 752
 - message formats of, 753
 - request types, 752
 - responses in, 752
- Solicit successor frame, 398
- SONET, 77
- Source route, 552
- Source Routing bridge, 442, 453–462
 - ARB (All Route Broadcast), 455
 - frame structure of, 453
 - RII bit in, 453
 - route designator field in, 455
 - route discovery in, 455–461
 - routing directives in, 455–458
 - SRB (Single Route Broadcast), 455, 458–461
- Space, 12
- Spanning tree, 447, 643
 - designated bridge in, 448
 - designated port in, 448
 - port cost in, 448
 - root bridge in, 448
 - root path cost in, 448

- root port in, 448
- Spectral width of optical sources, 50
- Split horizon (*see* Distance vector algorithm) Spool file (*see* E-mail) SREJ—N, 263
- SSL record protocol (*see* Secure socket layer) Star topology, 327
- Start bit, 9
- Static routing, 479, 594–595
- Statistical multiplexing, 498
- Statistical time division multiplexer, 144
 - error control in, 146
 - frame format of, 144
 - protocol, 145 (*see also* Bit map stat mux, Multiple-character stat mux) STM-1, 77
 - frame, 78
 - pointer, 80
- STM-4, 82
- STM-16, 82
- STM-64, 82
- Stop-and-wait error control, 258–263
- Stop-and-wait flow control, 249
 - link utilization in, 249, 261
- Stop bit, 9
- Store and forward data network, 473 (*see also* Message switching, Packet switching)
- Structure of Management Information (SMI), 751
- Subnetwork, 186
- Summary LSA (*see* Link State Advertisement) Supervisory frame of HDLC, 294, 295
- Switched virtual circuit (SVC), 498 (*see also* X.25) Synchronous communication, 29
- Synchronous digital hierarchy, 77
- Synchronous modem, 116
- Synchronous transfer mode, 527
- Synchronous transmission, 10–11
 - bit recovery in, 10
- Systematic code, 159

T1, 76

- Tail drop, 783
- Tapped delay line, 57
- Target Token Rotation Time (TTRT), 399, 406
- TCP, 546, 671–691
 - congestion avoidance mechanism of, 687–691
 - connections, 671
 - delayed acknowledgement in, 681, 685
 - fast recovery mechanism, 690
 - fast retransmission in, 680
 - flow control in, 682–685
 - forced data delivery in, 676

- format of, 673
- Maximum Segment Size (MSS) in, 675
- methods for estimation of, 686
- operation, 676
- ports, 671
- pseudo IP header in, 675
- Retransmission Timeout (RTO) in, 685
- segment, 673
- sliding window mechanism of, 683
- slow start mechanism of, 687
- socket, 672
- urgent data in, 686
- window, 683

TCP/IP 208, 209, 668

- application protocols, 726
- layered architecture of, 668

TDM (*see* Multiplexing) TEI, 314

Telephone network, 65–69

2-wire circuit in, 60

- 4-wire circuit in. 70
- access network of, 66
- international transit exchange in, 69
- single exchange area in, 66
- topology of, 66
- transport network of, 66
- trunk automatic exchange in, 68

Telnet, 738–740

- architecture of, 739
- commands, 739
- Network Virtual Terminal (NVT) in, 739
 - character set of, 739
 - user commands for, 740

Test loops in modems, 123

TFTP (*see* Trivial File Transfer Protocol) Thermal noise (*see* Noise) Time division data multiplexer, 143

- frame format of, 143
- synchronization word in, 143 (*see also* Bit interleaved data multiplexer, Byte interleaved data multiplexer)

Timing error, 8

Token, 381

Token bit, 386

Token bucket, 780

Token bus LAN, 395–402 (*see also* IEEE 802.4) addition of new stations in, 398

- bus initialization, 398
- bus management in, 397

- media access control in, 395
- physical specifications of, 401
- priority operation of, 399
- response window in, 397
- Target Token Rotation Time (TTRT) in, 399
- Token Holding Timer (THT) in, 399
- Token Rotation Timer (TRT) in, 399
- TRT-M, 399

Token frame, 385

Token ring LAN, 380–395 (see *also* IEEE 802.5) active monitor station, 392

- AMP frame in, 386, 392
- AMP timer, 392
- Beacon Control (BCN) frame in, 395
- bypass relay in, 383
- claim token frame in, 386, 392
- DAT frame in, 393
- early token release in, 383
- frame format of, 385
- M bit in, 394
- master clock in, 394
- maximum number of stations in, 385
- maximum segment length of, 385
- media access control in, 381
- physical layer in, 385
- priority management in, 389
- Ring Interface Unit (RIU) in, 383
- ring management in, 391
- ring size of, 383
- SMP frame in, 386, 392
- stacking station in, 390
- standards of, 385
- token management in, 393
- upstream neighbour in, 392

Token Holding Time (THT), 383

Token Rotation Timer (TRT), 339

Traceroute, 572

Traffic conditioning agreement (TCA), 763

Traffic control, 777

Traffic engineering (see MPLS) Traffic policing, 767, 779

Trailer, 246

Training sequence, 121

Transfer syntax, 195

Transit system 186

Transmission bridge, 67

Transmission channel, 19

Transmission Convergence (TC) sublayer (see ATM) Transmission line, 35–38

- primary parameters of, 35
- secondary parameters of, 38
- time domain characteristics of, 40

- Transmission media for LAN, 337
- Transparency, 248, 286
- Transparent bridge, 442, 443–452
 - address learning in, 444
 - Bridge Protocol Data Unit (BPDU) in, 448
 - designated bridge in, 448
 - designated port in, 448
 - frame filtering and forwarding in, 443
 - port cost in, 448
 - root bridge in, 448
 - root port in, 448
 - spanning tree algorithm of, 447–452
- Transport layer, 196, 667–671
 - congestion avoidance in, 671
 - data transfer in, 670
 - error control in, 670
 - flow control in, 670
 - functions within, 669
 - purpose of, 668
 - reset in, 671
- Transport Layer Security (TLS), 714
- Transport service, 669
 - connection-oriented, 669
 - connectionless, 669
- Trellis code, 133
- Trellis diagram, 178
- Trivial File Transfer Protocol (TFTP), 736–738
 - message formats, 737
 - message types, 737
 - operation, 738
- Trunk, 470
- Trunk automatic exchange, 68 (*see also* Telephone network)
- Trunk Coupling Unit (TCU), 384
- Tspec, 792
- TTL (Time to Live), 155, 548
- TU12, 80
- TUG2, 80
- TUG3, 80
- Two—Way Alternate (TWA) communication, 30
- Two—Way Simultaneous (TWS) communication, 30
- Type of Service (TOS), 548

- U-frame commands and responses, 298
- UDP, 546, 668, 689–693
 - datagram format of, 692
 - operation, 693
 - pseudo IP header, 692
- UHF, 51, 52, 83
- Unacknowledged connectionless-mode service, 332

- UNI, 528
- Unicast, 640, 641
- Unipolar signal, 6
- Uniquely decodable code, 25
- Unnumbered frame of HDLC, 294, 295
- Upward multiplexing, 206
- URL (Uniform Resource Locator), 755
 - scheme part of, 755
 - scheme specific part of, 755
- UTP cables, 45

- V.10 (ITU-T), 219
- V.11 (ITU-T), 219
- V.24 (ITU-T), 219, 221
- V.28 (ITU-T), 219, 221
- V.35 (ITU-T), 219
- V.36 (ITU-T), 136
- V.37 (ITU-T), 136
- V.42 (ITU-T), 125
- V.42*bis* (ITU-T), 125
- V.54 (ITU-T), 125
- Variable Bit Rate traffic (VBR) (*see* ATM) VC-12, 80
- VC-4, 79
 - B3 byte of, 79
 - C2 byte of, 79

J1 byte of, 79

- Path Overhead (POH) of, 79
- VDSL, 140
- VHF, 51, 52, 83
- Virtual circuit packet switching, 482–485
 - service features of, 485
- Virtual connection, 482, 519
- Virtual Container (VC), 79
- Virtual file, 741
- Viterbi decoder (*see* Convolutional codes) VRC (Vertical Redundancy Check), 172

- WACK, 249, 285
- Weighted fair queuing, 776
- Weighted RED, 786
- Weighted Round-Robin (WRR), 771
- Wide Area Network (WAN), 469
- Wire centre, 327
- Wireless local area network, 412
 - authentication/encryption in, 416

- Contention Free Period (CFP) in, 425
- Contention Period (CP) in, 425
- Distributed Coordination Function (DCF) of, 419–424
 - exposed station in, 417
 - fragmentation in, 424
 - hidden station in, 417
- Inter-frame Spaces (IFS) in, 419
- layered architecture of, 415
- management functions in, 416
- media access control in, 416–419
- Point Coordination Function (PCF) of, 424–426
 - reliable delivery in, 416
 - superframe in, 524
- World Wide Web (WWW), 754–758

- X.121 (ITU-T), 515
- X.200 (ITU-T), 194
- X.21 Physical layer (ITU-T), 237–239
 - specification of,
 - electrical, 238
 - mechanical, 238
 - procedural, 239
- X.21bis (ITU-T), 239
- X.212 (ITU-T), 245
- X.25 (ITU-T) 496–516
 - addressing in, 515
 - call clearing in, 508
 - call collision in, 509
 - call establishment in, 502
 - Closed User Group (CUG) in, 513
 - error recovery timers of, 511
 - fast select in, 512
 - flow control in, 503
 - interrupt packet in, 507
 - local and remote acknowledgements in, 505
 - logical channel, 498
 - Logical Channel Group Number (LGCN), 500
 - logical channel identifier, 499
 - Logical Channel Number (LCN), 500, 501
 - packet format, 500
 - Permanent Virtual Circuit (PVC), 498
 - restart in, 511
 - reverse charging in, 513
 - switched Virtual Circuit (SVC), 498
 - user facilities of, 512
- X.28 (ITU-T), 516
- X.29 (ITU-T), 516
- X.3 (ITU-T), 516
- xDSL, 137, 139