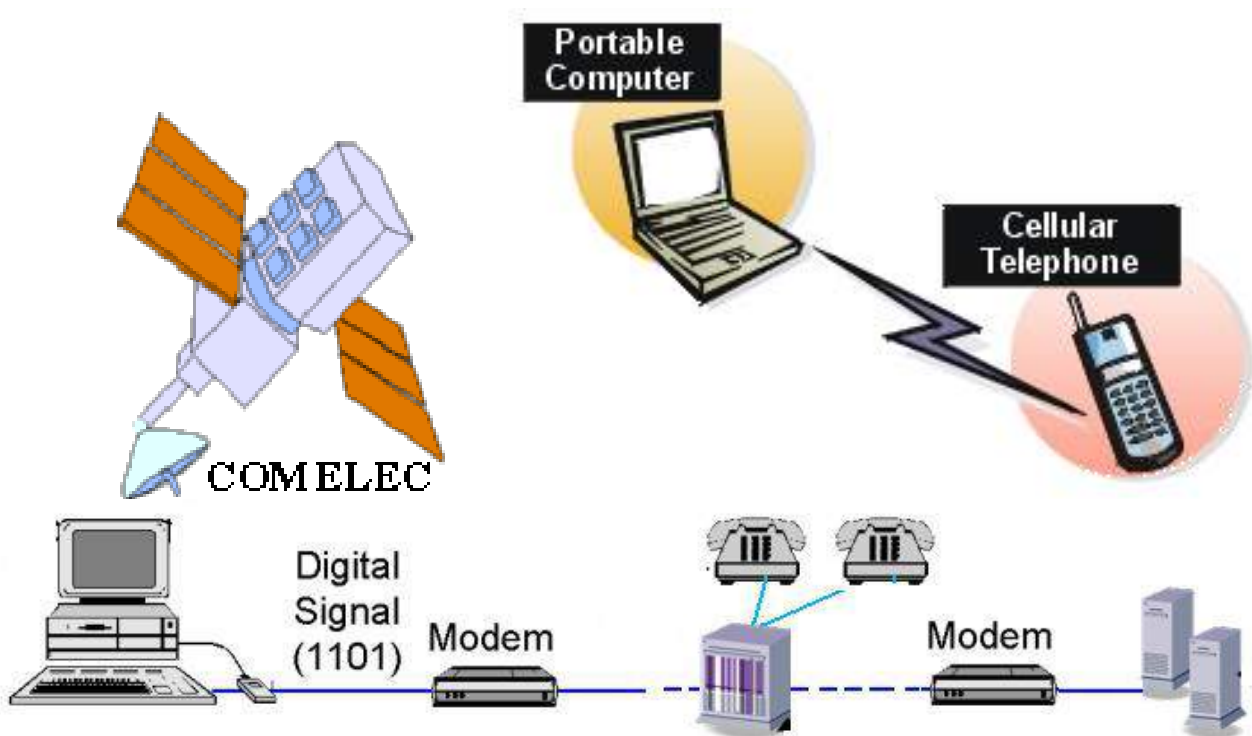


Fundamentals of Communication Systems & Networks



Prof. Dr. Muhammad EL-SABA

2015

ABOUT THE AUTHOR



Muhammad. El-SABA is currently professor at the Department. of Electronics & Communication Engineering in the Faculty of Engineering, Ain-Shams University in Cairo. He obtained his Ph.D. on Integrated Electronics & Electronic Design Automation (EDA) from INSA-Lyon, France in 1993. During his Ph.D. work, He designed and implemented the generic object-oriented device simulator GOOD-SIM™, which was adopted in the electronic industry. Dr. EL-SABA has been a Deputy Director and Head of department in several Universities and Engineering Institutes, all over the world. For instance, in Unisa (SA), Unitech (Australia), King Faisal (West Africa), Qassim (Saudi Arabia), and Hadramout(Yemen) and Ain-Shams (Cairo). He authored 28 books and more than 50 articles on the simulation of electronic devices and solid-state circuits. He also prepared and animated several courses in different areas of industrial electronics, microcontrollers, mobile communications, SystemC and VHDL-AMS. Dr. EL-SABA has many patents and breakthrough papers in IEEE Transactions in the field of high-power THz generation from solid-state silicon devices. He's currently interested in automatic synthesis of mixed signal communication IC's, with emphasis on THz communication systems.



Telecommunication Systems & Data Networks

Prof. Dr. Muhammad EL-SABA

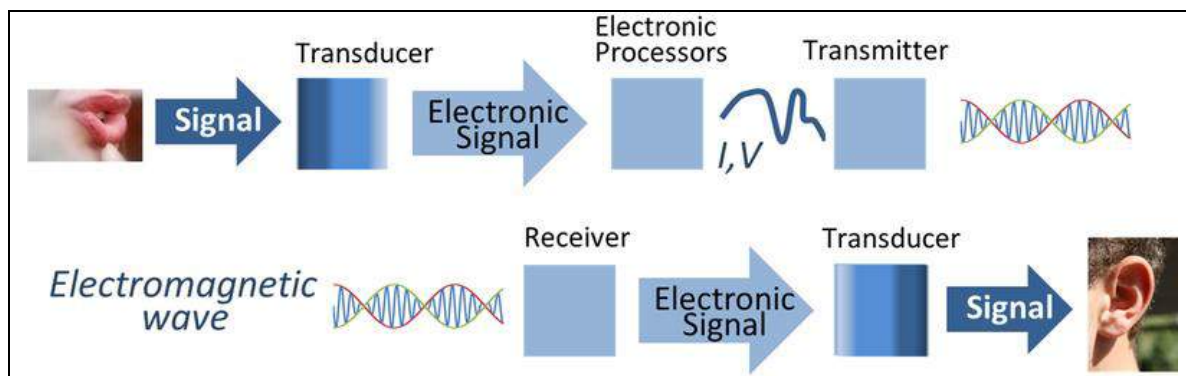
3rd Edition

2015

Preface

Telecommunication is communication at a distance by technological means, particularly through electrical signals or electromagnetic waves. This book presents the principles of telecommunication systems & networks, which are vital topics in communication and computer engineering. The course will help students to get acquaintance with the fundamentals of digital communication systems and data networks.

A communication **system** is a piece of hardware (like FM radio, TV, or cellphoner) or algorithm that reads input information, processes them and transmits outputs over a certain **channel**. Communication systems support people who are working together, by enabling the exchange of data and information electronically. The following figure depicts an audio signal communication system, via electromagnetic waves.



The field of communication systems includes two major traditional research areas: **telecommunication** and **computer communication**.

Modern communication systems most probably refer to the transmission and reception of digital data. This is in contrast with analog communications. While analog communications use a continuously varying signal, a digital transmission can be broken down into discrete messages. Transmitting data in digital form allows for greater signal processing capability and higher noise immunity.

This book is an extended (2-terms) communication course. The course is organized to provide the students with a logical and pedagogic flow of information with a description of various communication systems and technologies that are commonly used, nowadays. A special emphasis is put on the hardware components (in its simplest form), which lie behind the building blocks of communication systems. This is important, for

electronics & communication engineers, who are supposed to be able to design or maintain – or even will *see* and *deal with* - the **real** communication equipment, in their daily work.

This book is organized in 13 Chapters and 11 Appendices, as follows. **First chapter** of this book will review the basics of communication systems and analog modulation systems (AM & FM),. This may be covered in previous courses about the principles of analog communication systems.. However, we review these subjects for the matter of completeness. In fact, the basic analog modulation techniques lay behind all variations of pulse and digital modulation techniques. In addition, the different types of noise sources and their mathematical models, as well as the types of transmission channels are presented. This Chapter will also offer a brief description on the **information theory** and coding theorems.

Chapter 2 presents the **pulse modulation**, multiplexing and detection techniques. The pulse code modulation (**PCM**) as well as the so0called Sigma-Delta (Σ - Δ) modulation techniques are covered in this Chapter.

Chapter 3 is a brief discussion about the **line coding** methods. **Chapter 4** presents the topic of **digital modulation** techniques. We cover most of the coherent and differential digital modulation techniques. For instance, we present the transmission and reception techniques of quadrature phase shift keying (**QPSK**), digital quadrature amplitude modulation (**QAM**) and spread spectrum techniques (e.g., direct sequence **DSSS** and frequency hopping **FHSS**) among many other modulation methods.

Chapter 5 delves into the subject of errors and error bit rate (**BER**) in both analog, pulse and digital communication systems. The error detection and correction (**EDC**) methods are covered in Chapter 6. The famous error-correcting codes (**ECC**), such as **block** codes, **convolutional** codes, forward error correction (**FEC**), **Turbo Codes** and **Viterbi** decoders, are presented and briefly discussed, with illustrating examples in this chapter. **Chapters 7** and **8** cover data compression and data encryption, respectively. We present the different lossless and lossy compression algorithms, which may be used for digital files and streams, to reduce storage size, secure data and increase the speed of data transfer across digital communication links.

Chapter 9 covers the **computer communication links** and **protocols**. We present the fundamentals of **parallel** and **serial communications** and

their protocols. The open system interconnect (**OSI**) model as well as the **SNA** standards are presented at **Chapter 10**. **Chapters 11** depicts the **data networks** and their protocols. Both local area networks (**LAN's**) and wide-area networks (**WAN's**), as well as the world-wide web (**WWW**) and Internet are covered. The network interface devices, such as **gateways, bridges** and **routers** are presented. The **circuit switching** and **packet switching** networks and protocols are also described in this chapter. In addition, we also present the essentials of the **Ethernet** and the **wireless Ethernet** and their protocols, in this chapter.

In **Chapter 12** we present different examples of **digital communication systems** and demonstrate how to calculate the link power budget for such systems. We handle different communication links, such as telephone lines, digital subscriber lines (**xDSL**), wireless, links, digital **satellite** links, **VSAT** links, and optical communication links.

This book contains some computer programs, written in C/C++/Java languages (and sometimes Matlab scripts), that implement the different building blocks of digital communication systems.. The prerequisite topics, to be studied before this course, are: Probability theory, Random Variables, Signals and Systems.

I'm greatly indebted to all my professors (all of them) in Egypt, France and United States, from whom I've learnt so much. In particular, I feel I'm truly grateful to Prof. Dr. **Safwat Mahrous** (deceased), who had taught me the principles of communications and RF engineering.

Prof. Dr. Muhammad EL-SABA,

Windhoek in May, 2015.

Contents

Subject	Page
PREFACE	iii
CHAPTER 1: Basic Elements of Communication Systems	1
1-1. Signals and Systems	3
1-2. Communication Systems	7
1-3. Modulation & Coding Techniques	10
1-4. Baseband and Passband Signals	11
1-5. Communication Channels	13
1-5.1. AWGN Channels	13
1-5.2. Fading Channels	13
1-5.3. Multipath & Fading Problems	14
1-6. Noise and Electromagnetic Interference	15
1-6.1. Noise Types	16
1-6.2. Modeling of Noise	16
1-6.3. Signal to Noise Ratio	18
1-6.4. Noise Figure and Noise Temperature	19
1-7. Basics of Information Theory	20
1-7.1. Quantity of Information	21
1-7.2. Information Entropy	22
1-7.3. Joint Entropy	23
1-7.4. Conditional Entropy (equivocation)	24
1-7.5. Mutual information (trans-information)	24
1-7.6. Channel Capacity	25
1-7.7. Shannon's Theorem	26
1-7.8. Coding Theorem	27
1-8. Spectral Efficiency	28
1-9. Principles of Queuing Theory	29
1-10. Basics of Teletraffic Engineering	31
1-11. Communication Standardization Organizations	34
1-12. Spectra of Electromagnetic Waves & RF Bands	35
1-13. Summary	38
1-14. Problems	41
1-15. Bibliography	42
CHAPTER 2: Analog Modulation & Radio Systems	43
2-1. Radio Communication Systems	45
2-1.1. Superheterodyne Receiver	46
2-1.2. Image-Reject Receivers (IRR) Receiver	49
2-1.3. Direct Conversion (Zero IF) Receivers	49

	Subject	Page
2-2.	Analog Modulation Techniques	50
2-3.	Amplitude Modulation (AM)	53
	2-3.1. AM Signals	53
	2-3.2. AM Schemes	55
	2-3.3. AM System Architecture	56
	i. Product Detector	58
	ii. Envelop Detector	59
	2-3.4. AM Over-Modulation & its Effects	59
	2-3.5. Advantages & Disadvantages of AM	60
2-4.	Frequency Modulation (FM)	61
	2-4. 1. FM Signals	61
	2-4. 2. FM Spectrum & Bandwidth	62
	2-4. 3. FM System Architecture	65
	i. Fooster-Seely FM Discriminator	65
	ii. Ratio Detectorr	67
	iii. Quadrature Detectorr	67
	iv. PLL	68
	2-4. 4. Pre-emphasis and De-emphasis	69
	2-4. 5. Stereo FM Broadcast	70
2-5.	Noise in Radio Systems	72
	2-5.1. Friis Formula for Noise factor	72
	2-5.2. Minimum Detectable Signal (MDS) of Receiver	75
	2-5.3. Sensitivity of a Receiver	76
2-6.	Testing of a Radio Receiver	77
	2-6.1. Sensitivity Test	77
	2-6.2. Selectivity Test	79
	2-6.3. Fidelity Test	80
2-7.	Summary	81
2-8.	Problems	84
2-9.	Bibliography	85
CHAPTER 3: Pulse Modulation, Multiplexing & Detection		87
3-1.	Introduction	89
3-2.	Pulse Amplitude Modulation (PAM)	91
3-3.	Pulse Width Modulation (PWM)	94
3-4.	Pulse Position Modulation (PPM)	95
3-5.	Pulse-Code Modulation (PCM)	96
	3-5.1. Modulation Process	96
	3-5.2. Quantization Errors	99
	3-5.3. Encoding	100
	3-5.4. PCM Demodulation	102
	3-5.5. Compression/Expanding (Companding)	104
	3-5.6. Applications of PCM	104
	3-5.7. Other Forms of PCM	106

	Subject	Page
3-6.	Segma -Delta (Σ-Δ) Modulation	108
	3-6.1. Implementation of Σ - Δ Modulators	109
	3-6.2. Quantization Errors in Δ - Σ Modulation	112
	3-6.3. Signal-to-Noise Ratio & Dynamic range	113
	3-6.4.. Dynamic range in Σ - Δ Modulators	113
	3-6.5. Higher order Δ - Σ Modulators	113
3-7.	Multiplexing & Multiple Access Techniques	115
	3-7.1. Time Division Multiplexing (TDM)	116
	i- TDMA Transmission	117
	ii- T1 Framing	117
	iii- E1 Framing	117
	iv- Higher Order Multiplexing	118
	v- Statistical TDM (STDM)	119
	3-7.2. Frequency Division Multiplexing (FDM)	119
	3-7.3. Code Division Multiplexing (CDM)	120
	i- Code Division Modulation & Demodulation	121
	ii- Properties of Spreading Codes	123
	3-7.4. Wave Division Multiplexing (WDM)	124
3-8.	Baseband Transmission Problems	126
	3-8.1. Channel Limitations	126
	3-8.2. Inter-Symbol Interference (ISI)	126
	3-8.3. Jitter	127
	3-8.4. Eye Pattern	128
3-9.	Pulse Detection & Matched Filters	131
	3-9.1. Pulse Detection	131
	3-9.2. Matched Filters	131
	3-9.3. Practical Pulse Shaping & Detection	137
	3-9.4. Cosine-Raised Filters	138
3-10.	Summary	141
3-11.	Problems	146
3-12.	Bibliography	147
CHAPTER 4: Digital Baseband Modulation (Line Coding)		149
4-1.	Introduction	151
4-2.	RZ Coding	152
4-3.	NRZ Coding	153
4-4.	NRZI Coding	155
4-5.	AMI Coding	156
4-6.	Manchester Coding	157
4-7.	Comparison of Different Line Codes	158
4-8.	Summary	161
4-9.	Problems	164
4-10.	Bibliography	165

Subject	Page
CHAPTER 5: Digital Modulation Methods	167
5-1. Introduction	169
5-2. Amplitude Shift Keying (ASK)	173
5-3. Frequency Shift Keying (FSK)	174
5-4. Phase Shift Keying (PSK)	176
5-5. Binary Phase Shift Keying (BPSK)	177
5-5.1. BPSK Constellation Diagram	177
5-5.2. BPSK Signals	178
5-5.3. Spectral Density of BPSK Signals	178
5-5.4. Implementation of BPSK	179
5-5.5. Costas Loop	181
5-6. Quadrature Phase Shift Keying (QPSK)	183
5-6.1. QPSK Constellation Diagram	183
5-6.2. QPSK in the Time Domain	184
5-6.3. Power Spectral Density of QPSK Signals	186
5-6.4. Alternative Types of QPSK Signals	187
i. Offset QPSK	187
ii. $\pi/4$ QPSK	188
5-6.5. Implementation of QPSK	190
5-7. Differential Encoding (DBPSK, DQPSK)	193
5-8. Quadrature Amplitude Modulation (QAM)	194
5-8.1. QAM Constellation Diagram	195
5-8.2. QAM in the Time Domain	196
5-8.3. Implementation of QAM	196
5-9. Orthogonal FDM (OFDM)	197
5-10 Continuous Phase Modulation (CPM)	202
5-10.1. Minimum-Shift Keying (MSK)	202
5-10.2. Gaussian MSK (GMSK)	204
5-10.3. Very-Minimum Shift Keying (VMSK)	206
5-11. Spread Spectrum Techniques	207
5-11.1. DSSS	207
5-11.2. FHSS	210
5-11.3. Features of Spread Spectrum Techniques	212
5-12. Trellis Code Modulation (TCM)	214
5-13. Summary	217
5-14. Problems	220
5-15. Bibliography	222
CHAPTER 6: Noise Estimation & Bit Error Rate	223
6-1. Introduction	225
6-2. SNR in AM & FM Systems	225

	Subject	Page
6-3.	Error in Digital Systems	226
	6-3.1. Modulation Error Magnitude (MER)	227
	6-3.2. Error Vector Magnitude (EVM)	227
6-4.	Bit Error Rate (BER) in Digital Systems	228
6-5.	Measurement of BER in Digital Modulation Systems	230
6-6.	Computing the BER in Digital Modulation Systems	231
6-7.	BER of BPSK	232
6-8.	BER of QPSK	234
6-9.	Error Probability of Higher-order PSK	235
	6-9.1. Symbol-Error Rate in M-PSK	235
	6-9.2. Bit-Error Rate in M-PSK	236
6-10.	BER of Differential Phase-Shift Keying (DPSK)	237
6-11.	BER Measurement of QAM	238
	Case 1: Rectangular QAM with Even k	239
	Case 2: Rectangular QAM with Odd k	239
	Case 3: Non-Rectangular QAM	240
6-12.	Comparison between Digital Modulation Methods	242
6-13.	Summary	243
6-14.	Problems	246
6-15.	Bibliography	247
CHAPTER 7: Error Detection & Correction		249
7-1.	Introduction	251
	7-1.1. Channel Coding Methods	252
	7-1.2. Coding Gain & Coding Performance	253
7-2.	Error Detection Schemes	254
	7-2.1. Repetition Scheme	255
	7-2.2. Parity Scheme	256
	7-2.3. Hamming Codes	256
	7-2.4. Cyclic Redundancy Check (CRC)	260
	7-2.5. Checksum Scheme	264
7-3.	Error-Correcting Codes (ECC)	265
	7-3.1. Block Codes	266
	7-3.2. Convolutional Codes	269
	7-3.3. Viterbi Algorithm Decoders	264
	7-3.4. Concatenated Codes	265
	7-3.5. List of ECC Codes	266
7-4.	Forward Error Correction (FEC)	268
7-5.	Turbo Codes	269
	7-5.1. Turbo Encoders Structure	280
	7-5.2. Turbo Decoders	282
	7-5.3. Illustration Example of a Turbo coder/	285
	7-5.4. Performance of Turbo Coders	286

Subject		Page
7-6.	Applications of Error-Correcting Codes	287
	7-6.1. Satellite TV and DVB-x	287
	7-6.2. Data Storage	288
	7-6.3. Internet	289
	7-6.4. Deep-Space Telecommunications	289
7-7.	Summary	290
7-8.	Problems	294
7-9.	Bibliography	296
CHAPTER 8: Source Coding & Data Compression		299
8-1.	Source Coding Theorem	301
8-2.	Data Compression	303
8-3.	Lossless (Data File) Compression	305
	8-3.1. Lossless Compression Algorithms	305
	8-3.2. Run-Length Encoding (RLE) Algorithm	305
	8-3.3. Huffman Coding Algorithm	308
	8-3.4. LZ and LZW Algorithms	313
	8-3.5. Delfate Algorithm	318
	8-3.6. ZIP File Format	318
8-4.	Lossy (Image, Audio, Video Files) Compression	320
	8-4.1. Lossy Compression Algorithms	320
	8-4.2. Audio Compression (MP3, AAC,..)	320
	8-4.3. Image Compression (BMP, JPG, GIF, PNG,..)	326
	8-4.4. Video Compression (AVI, MPEG, 3GP,..)	331
8-5.	Summary	336
8-6.	Problems	339
8-7.	Bibliography	340
CHAPTER 9 Data Encryption		341
9-1.	Introduction	343
	9-1.1. Elements of Encryption Systems	343
	9-1.2. Illustration Example	344
	9-1.3. Encryption Types & Algorithms	345
9-2.	Symmetric-Key Encryption Algorithms	347
	9-2.1. DES Algorithm	348
	i. Feistel (F) Function	350
	ii. Key Schedule	351
	9-2.2. AES Algorithm	352
9-3.	Asymmetric-Key Encryption Algorithms	355
	9-3.1. Public Key Infrastructure (PKI)	356
	9-3.2. RSA Algorithm	357
	9-3.3. EL-Gammal Algorithm	358

Subject		Page
9-4.	Hashing Encryption	359
	9-4.1. Difference between Hashing and Encryption	359
	9-4.2. Digital Signature	360
	9-4.3. SHA-1 Algorithm	361
9-5.	Security and Cryptanalysis	362
	9-5.1. Brute Force Attack	362
	9-5.2. Attacks Faster than Brute-Force	363
	9-5.3. Cryptanalytic Properties	364
9-6.	Applications of Encryption	366
	9-6.1. Securing Networks	366
	9-6.2. SSL	369
	9-6.3. Wireless Networks	370
	i. Wired Equivalent Privacy (WEP)	370
	ii. Wi-Fi Protected Access (WPA)	371
	iii. Wi-Fi Protected Access 2 (WPA2)	371
	9-6.4. Access Control	371
	9-6.5. Private Use	372
9-7.	Future Developments	373
9-8.	Summary	374
9-9.	Problems	378
9-10.	Bibliography	379
CHAPTER 10: Computer Communications & Protocols		383
10-1.	Fundamentals of Computer Communications	385
10-2.	Resolution of Communication Conflicts (Protocols)	387
10-3.	Principles of Data Transmission	388
10-4.	Parallel Data Transmission & Parallel Ports	392
	10-4.1. Parallel Data Link	392
	10-4.2. Handshaking in Parallel Communication	393
	10-4.3. Centronics Parallel Port	394
	10-4.4. Data Transmission, with IEEE-488 (GPiB)	400
10-5.	Serial Data Transmission & Serial Ports	405
	10-5.1. Parallel to Serial Conversion	405
	10-5.2. Synchronous Serial Data Communications	407
	10-5.3. Asynchronous Serial Data Communications	411
	10-5.4. Error Detection Techniques	414
	10-5.5. UARTS and USARTS	417
	10-5.6. RS-232C Standard	420
	10-5.7. Universal Serial Bus (USB)	427

Subject	Page
10-5.8. Other Serial Communication Standards	430
i- ACCESS.bus	430
ii- FireWire	430
iii- IrDa Bus	431
iv- I ² C	432
v- SMBus	432
vi- JTAG (IEEE 1149)	432
vii- Serial Peripheral Interface (SPI) Bus	433
viii- Local Interconnection Network (LIN)	434
10-5.9. Selecting a Serial Communication bus	435
10-6. Summary	437
10-7. Problems	439
10-8. Bibliography	440
CHAPTER 11: Communication Network Models	441
11-1. Introduction	443
11-2. ISO / OSI Seven Layer Model	445
11-2.1. The Seven Layers	445
1- Physical Layer	445
2- Data Link Layer	445
3- Network Layer	446
4- Transport Layer	446
5- Session Layer	446
6- Presentation Layer	446
7- Application Layer	446
11-2.2. Sending Data Via the OSI Model	447
11-2.3. OSI Model Protocols	448
11-2.4. Protocol Data Units (PDUs)	449
11-2.5. Network Components	450
11-3. SNA Model	452
11-4. National Transportation Control Interface Protocol (NTCIP)	454
11-5. Telecommunication Management Network (TMN)	455
11-6. Summary	458
11-7. Problems	461
11-8. Bibliography	462
CHAPTER 12: Data Networks	463
12-1. Introduction	465
12.1.1. Network Types	465
12.1.2. Network Traffic Control Mechanisms	468
i. CSMA/CD	468
ii. Token Passing	469
12.1.3. Network Components	470

	Subject	Page
12-2.	LAN's & Their Physical Layer	474
	12.2.1. LAN Topology	474
	i. Star Topology	475
	ii. Bus Topology	476
	iii. Ring Topology	476
	iv. Tree Topology	477
	12.2.2. LAN Cables	479
	12.2.3. Wireless LAN	480
	12.2.4. LAN Standards	481
12-3.	Ethernet & its Physical Layer	482
	12-3.1. Ethernet Cables & Connectors	483
	12-3.2. Fast Ethernet & Gigabit Ethernet	484
	12-3.3. Ethernet Repeaters, Hubs and Switches	485
	12-3.4. Ethernet Adapter Cards	487
	12-3.5. Wireless Ethernet (IEEE 802.11b)	488
12-4.	Wide Area Networks (WAN)	491
	12-4.1. Public Data Networks (PDN)	492
	12-4.2. Switching Networks	492
	12-4.3. Circuit Switched Data Networks (CSDN)	493
	12-4.4. Packet Switched Data Networks (PSDN)	493
12-5.	Data Link Protocols	495
	12-5.1. Binary Synchronous Control (BSC) or BiSync	498
	12-5.2. HDLC / SDLC Protocols	498
	12-5.3. MAC & LLC	500
	12-5.4. Ethernet Protocol for LANs	502
	12-5.5. Point-to-Point Protocol (PPP) for LANs	503
12-6.	Internet Protocol (IP) and TCP/IP Suite	504
	12-6.1. IP Addressing (IPv4)	504
	12-6.2. Transmission Control Protocol (TCP)	508
	12-6.3. User Datagram Protocol (UDP)	508
	12-6.4. TCP/IP Suite Model	510
	12-6.5. IPv6 & IPTV	510
12-7.	Other WAN Protocols	512
	12-7.1. X.25 Standard of Packet Switching	513
	12-7.2. Frame Relays	514
	12-7.3. Integrated Service Digital Network (ISDN)	515
	12-7.4. Cell Relay Networks	516
	12-7.5. Asynchronous Transfer Mode (ATM)	516
	12-7.6. VoIP and VoATM	518
12-8.	Integration of Voice and Data Networks	520

Subject	Page
12-9. Summary	523
12-10. Problems	527
12-11. Bibliography	528
CHAPTER 13: Miscellaneous Communication Systems	529
13-1. Radio Relay Links	531
13-1.1. Line of Sight (LOS) Radio Links	532
13-1.2. Non LOS Radio Links	534
13-1.3. Other Media Links	535
13-2. Satellite Communications & VSAT	536
13-2.1. Types of Satellite Links	537
i. Orbiting Satellites	537
ii. Geostationary Satellites	538
iii. VSAT	540
13-2.2. Satellite Link Budget	542
13-2.3. Satellite DownLink Budget Analysis	545
13-3. Cellphone Communications	546
13-3.1. Cellphone Network Components	548
13-3.2. Cell Phone-Satellite Link Budget	549
13-4. Computer and Internet Links	551
13-4.1. Wi-Fi and WiMax	552
13-4.2. Satellite Internet Access	553
13-5. Public Telephone Switching Network (PSTN)	554
13-6. Digital Subscriber Lines (DSL)	559
13-6.1. Asymmetric Digital Subscriber Lines (ADSL)	560
13-6.2. ADSL Technology	561
13-6.3. ADSL Wiring and Filters	562
13-6.4. xDSL Standard	564
13-7. Optical Fiber Communications Links	564
13-7.1. Optical Link Power Budget	564
13-7.2. Practical Optical Link Budget	565
13-7.3. Fiber to the Premises (FTTP)	567
13-7.4. Fiber to the Home (FTTH)	569
13-8. DVB Systems	571
13-9. Summary	579
13-10. Problems	581
13-11. Bibliography	584

Subject		Page
Appendices		585
Appendix A:	Continuous-time Signals and their Properties	589
Appendix B:	Fourier Series of Periodic Signals	591
Appendix C:	Laplace Transform	593
Appendix D:	Fourier Transform (FT) & Inverse FT	593
Appendix E:	Discrete-time Signals and their Properties	595
Appendix F:	Discrete Fourier Transform (DFT) & Inverse DFT	597
Appendix G:	Fast Fourier Transform (FFT)	599
Appendix H:	Z-Transform (ZT) & Inverse ZT	607
Appendix I:	Discrete Cosine Transform (DCT)	609
Appendix J:	Wavelet Transforms	612
Appendix K:	MATLAB Communication Toolbox	617

Chapter 1

Basic Elements of Communication Systems

Contents

- 1-1. Signals & Systems**
- 1-2. Communication Systems**
- 1-3. Modulation & Coding Techniques**
- 1-4. Baseband and Passband Signals**
- 1-5. Communication Channels**
 - 1-5.1. AWGN Channel
 - 1-5.2. Fading Channel
 - 1-5.3. Multipath and Fading Problems
- 1-6. Noise and Electromagnetic Interference**
 - 1-6.1. Types of Noise
 - 1-6.2. Noise Modeling
 - 1-6.3. Signal to Noise Ratio (SNR)
 - 1-6.4. Noise Figure and Noise Temperature
- 1-7. Basics of Information Systems**
 - 1-7.1. Quantity of Information
 - 1-7.2. Information Entropy
 - 1-7.3. Joint Entropy
 - 1-7.4. Conditional Entropy (equivocation)
 - 1-7.5. Mutual information (trans-information)
 - 1-7.6. Channel Capacity
 - 1-7.7. Channel Capacity of Particular Channels
 - 1-7.8. Shannon's Theorem
 - 1-7.9. Coding Theorem
- 1-8. Spectral Efficiency**
- 1-9. Basics of the Queuing Theory**
- 1-10. Basics of the Traffic Theory**
- 1-11. Communication System Standardization Organizations**
- 1-12. Spectra of Electromagnetic Waves & RF Bands**
- 1-13. Summary**
- 1-14. Problems**
- 1-15. Bibliography**

Chapter

1

Basic Elements of Communication Systems

1-1. Review about Signals and Systems

A **signal** is a detectable physical quantity or impulse (as voltage or current) by which messages or information can be transmitted. In communication systems, there exist two broad classes of signals, **continuous-time** (analog) signals and **discrete-time** (digital) signals.

Continuous-time analog signals directly represent physical variables, like sound, pressure, and temperature. Such physical signals can be easily transformed to electrical signals and vice versa, using special transducers.

Discrete-time signals have discrete values. The binary digital signals, which may have one of 2 values (0 or 1), are the most common form of discrete-time signals. Digital signals can be obtained from analog signals using specific devices, called analog-to-digital converters (**ADC's**).

A system is any process that produces an output signal in response to an input signal. This is illustrated by the block diagram in figure 1-1. Continuous systems input and output continuous signals, such as in analog electronics. Discrete systems input and output discrete signals, such as computer programs that manipulate the values stored in arrays.

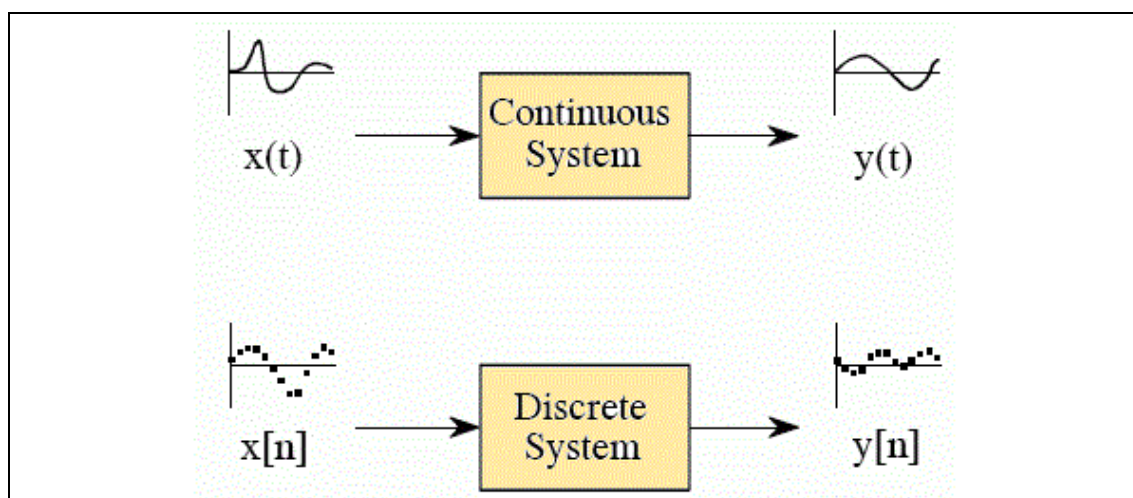


Fig. 1-1. Time domain and frequency domain of some signals.

The easiest form of continuous-time systems to treat with is called linear time-invariant (**LTI**) systems. LTI systems have a number of properties that make them ideal to study. The specific properties of such systems and their transformation methods, e.g. from time-domain to frequency domain, are briefly covered in **Appendix A** of this book. More mathematical details about LTI systems can be found in specialized signal processing books.

On the other hand, the system which handles digital signals and processes them using digital circuits is called digital system. Many functions that were traditionally implemented by analog systems are nowadays implemented by digital circuits. The most common form of discrete-time systems that we treat in this book are called linear shift-invariant (**LSI**) systems. The properties of LSI systems and their transformation methods are covered in **Appendix B**. More details about digital signals and systems can be found in specialized books on digital signal processing. Most of the signals in practice are **time-domain** signals in their raw format. That is, whatever that signal is measuring, is a function of time.

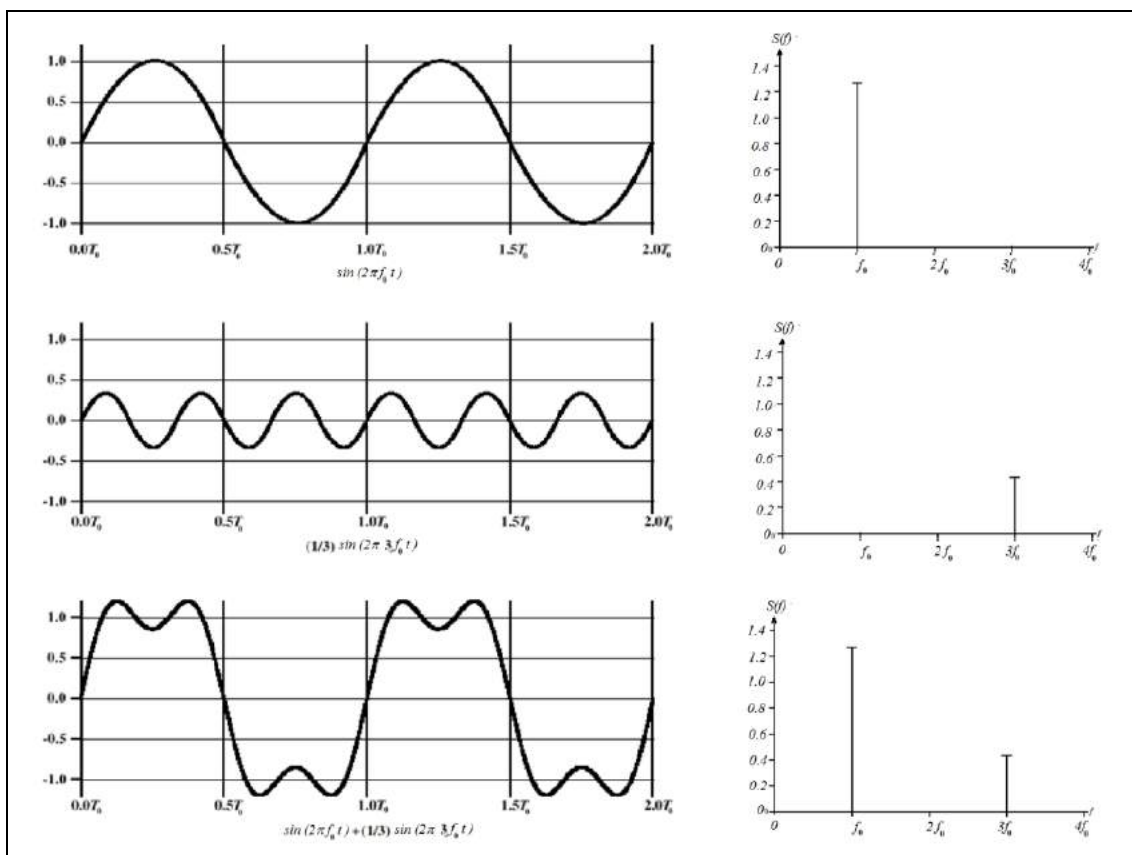


Fig. 1-2. Time domain and frequency domain of some signals.

In other words, when we plot the signal one of the axes is time (independent variable), and the other (dependent variable) is usually the amplitude. When we plot time-domain signals, we obtain a **time-amplitude** representation of the signal. This representation is not always the best representation of the signal for most signal processing related applications. In many cases, the most distinguished information is hidden in the frequency content of the signal. The **frequency spectrum** of a signal is basically the frequency components (spectral components) of that signal. The frequency spectrum of a signal shows what frequencies exist in the signal. In order to obtain the frequency-domain representation of a signal from its time-domain representation, and vice versa, we apply some transforms, such as the Fourier series and Fourier transform.

Note 1-1. Fourier Series

The Fourier series of a periodic signal $v(t)$, with period T , is given by

$$v(t) = \sum_{n=-\infty}^{\infty} C_n \exp(j n \omega_o t) = C_0 + \sum_{n=1}^{\infty} [A_n \cos(n \omega_o t) + B_n \sin(n \omega_o t)]$$

or

$$v(t) = V_0 + V_1 \cos(\omega_o t + \theta_1) + V_2 \cos(2 \omega_o t + \theta_2) + \dots + V_n \cos(n \omega_o t + \theta_n)$$

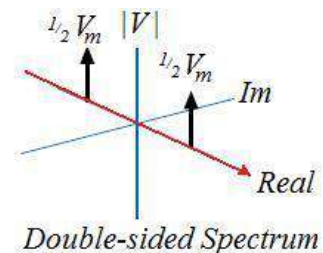
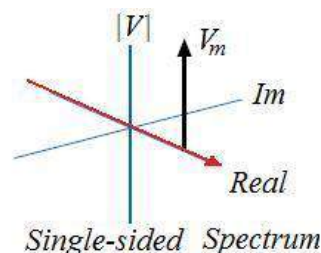
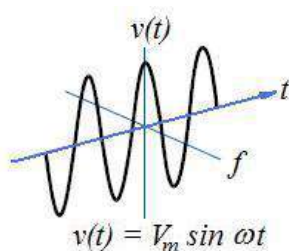
where ω_o is the radial frequency of the periodic signal ($\omega_o = 2\pi f_o = 2\pi/T$) and n is an integer.

Also, C_o is the average (DC) value of the signal and C_n are the amplitudes of the spectral components of the periodic signal.

$$C_o = V_o = (1/T) \int_0^T v(t) dt, \quad C_n = (1/T) \int_0^T v(t) \cdot \exp(-j n \omega_o t) dt$$

$$A_n = (2/T) \int_0^T v(t) \cdot \cos(j n \omega_o t) dt, \quad B_n = (2/T) \int_0^T v(t) \cdot \sin(+j n \omega_o t) dt$$

$$V_n = (2/T) \int_0^T v(t) \cdot \sin(n \omega_o t) dt, \quad \theta_n = \tan^{-1} (A_n/B_n)$$


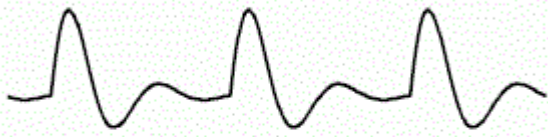




Note 1-2. Fourier Transforms

The **Fourier Transform** converts a time-domain continuous signal $s(t)$, which is not necessarily periodic, into its frequency-domain representation, as a function of the radial frequency, ω

$$S(\omega) = \int_{-\infty}^{+\infty} s(t)e^{-j\omega t} dt$$

This is sometimes called: Continuous-time Fourier transforms (**CTFT**). Actually, there exist 4 types of Fourier transforms, which can be applied to different types of signals, as shown in the following table.

Example Signal	Type of Transform
	Fourier Transform <i>signals that are continuous and aperiodic</i>
	Fourier Series <i>signals that are continuous and periodic</i>
	Discrete Time Fourier Transform <i>signals that are discrete and aperiodic</i>
	Discrete Fourier Transform <i>signals that are discrete and periodic</i>

The discrete Fourier transform (**DFT**) is the Fourier family member used with discrete (sampled) signals, $s(n)$, with a finite number of samples (say N). It is defined as follows:

$$S(k) = \sum_{n=0}^{N-1} s(n)e^{-j2\pi nk/N}; \quad 0 \leq k \leq N-1$$

1-2. Communication Systems

A communication system process communications signals between a transmitting **source** (transmitter) and receiving **destination** (receiver). In fact, the ability to process a communications signal means that errors caused by random processes (noise) can be **detected** and **corrected**. The main feature of a communication system is its ability to convey information, between receiver and transmitter, in the presence of transmission impairments such as noise, distortion and losses.

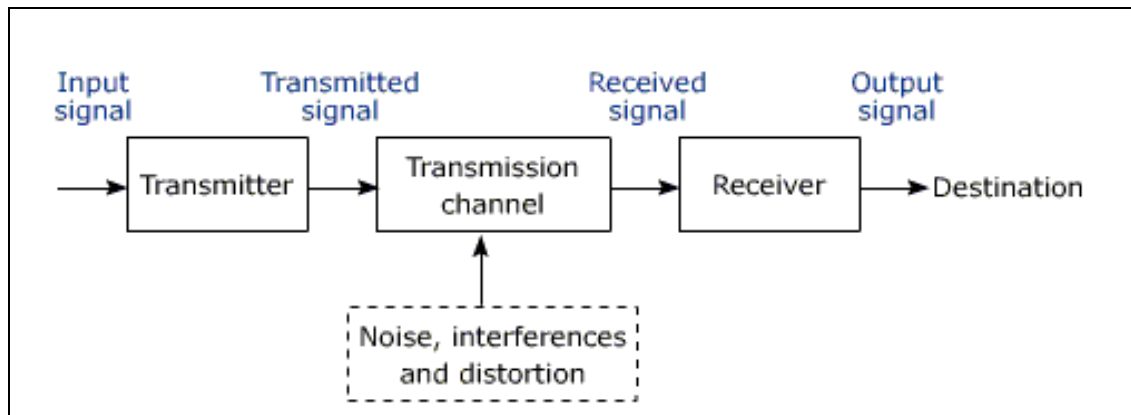


Fig. 1-3. Block diagram of a communication system.

The above figure depicts the elements of a communication system. There are three essential parts of any communication system, the transmitter, transmission channel, and receiver. Each part plays a particular role in signal transmission, as follows: The transmitter processes the input signal to produce a suitable transmitted signal suited to the characteristics of the transmission channel. Signal processing for transmissions almost always involves modulation and may also include coding. The transmission channel is the electrical medium that bridges the distance from source to destination. It may be a pair of wires, a coaxial cable, or a radio wave or laser beam. Every channel introduces some amount of transmission loss or attenuation. So, the signal power progressively decreases with increasing distance.

Radio and TV as well as computer communication represent distinct traditions within the field of telecommunications. The field of telecommunication is no doubt one of the most exciting occupational fields that modern society has to offer. New technology is constantly being developed and finds its applications in the technical systems that make up a telecommunications network. The following list enumerates some typical examples of communication systems and networks.

- Telephone Networks,
- Computer Networks (Internet , Ethernet, VoIP, IPTV),
- Digital Video Broadcasting (DVB),
- Optics Fiber Networks: backbone and Fiber to the Home (FTTH),
- Power Line Communication (PLC),
- Wireless Communications,
- Satellite Communications,
- Cellular Networks (GSM, 3G, 4G)
- Wireless LAN (Wi-Fi or IEEE 802.11), WiMAX
- Bluetooth

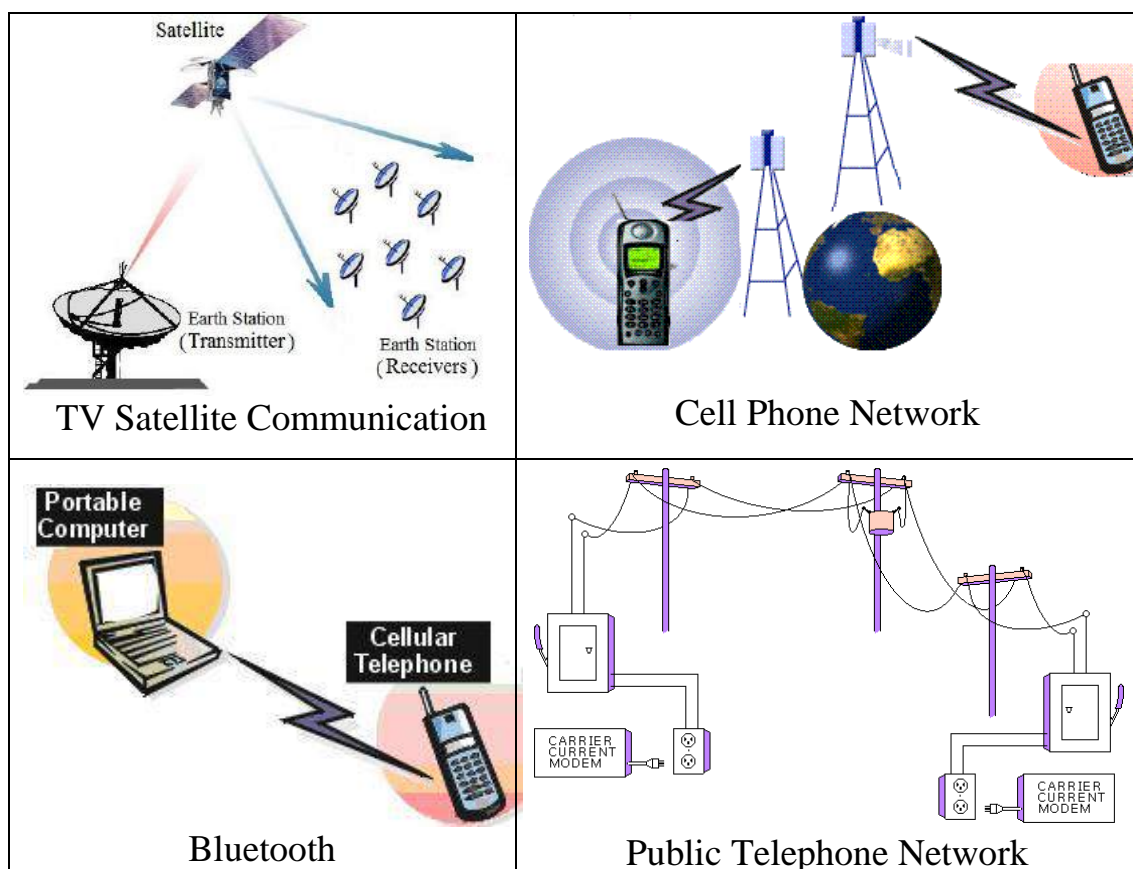


Fig. 1-4 Examples of modern communication systems

Any communication system is composed of a **transmitter**, a **receiver** and a **channel** to convey information between them. Figure 1-5 illustrates the block diagram of a communication system. The **source** may be either an analog signal, such as an audio or video signal, or a digital signal, such as the output of a teletype machine. The source data is converted into a high frequency signal. The process of conversion is called **modulation**.

At the receiving end of a communication system, the **demodulator** processes the transmitted waveform, which is corrupted across the channel, and reduces the waveforms to the original sent information.

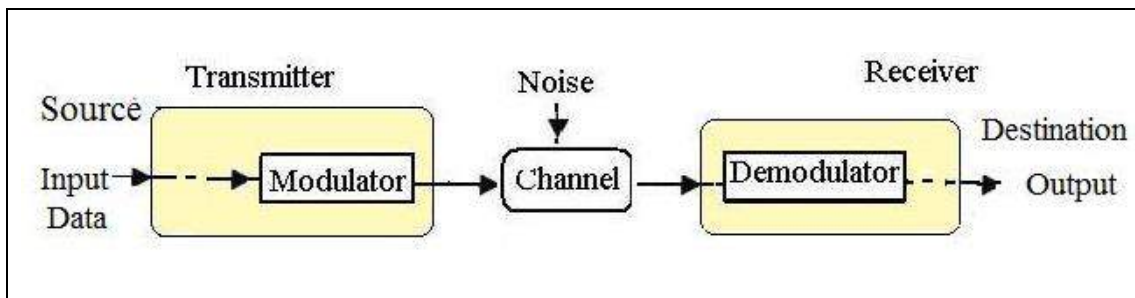


Fig. 1-5. Block diagram of a communication system

1-3. Modulation & Coding Techniques

In telecommunications, **modulation** is the process of varying a periodic waveform, in order to use that signal to convey (transmit) a message. Normally a **high-frequency** sinusoidal wave is used as a carrier signal. In fact, it is easier to transmit a high frequency carrier, via antennas with reasonable size (usually about $\lambda/2$), and moderate bandwidth. The three key parameters of a carrier are its amplitude, its phase and its frequency, all of which can be modified in accordance with the low frequency information signal to be transmitted. The various modulation techniques in analog systems (such as amplitude modulation, **AM** frequency modulation **FM** or phase modulation **PM**), offer different solutions in terms of cost-effectiveness and quality of received signals. We can also communicate information in digital form by digital modulation of a carrier waveform. The digital modulation process involves some form of AM and/or FM or PM. Hence digital modulation may be regarded as alternative ways of AM/FM/PM to communicate information in terms of bits. Figure 1-4 shows the basic modulation and coding techniques, which are used in different communication systems. Coding methods are used for data compression, cryptography, error-correction and more recently also for network coding.

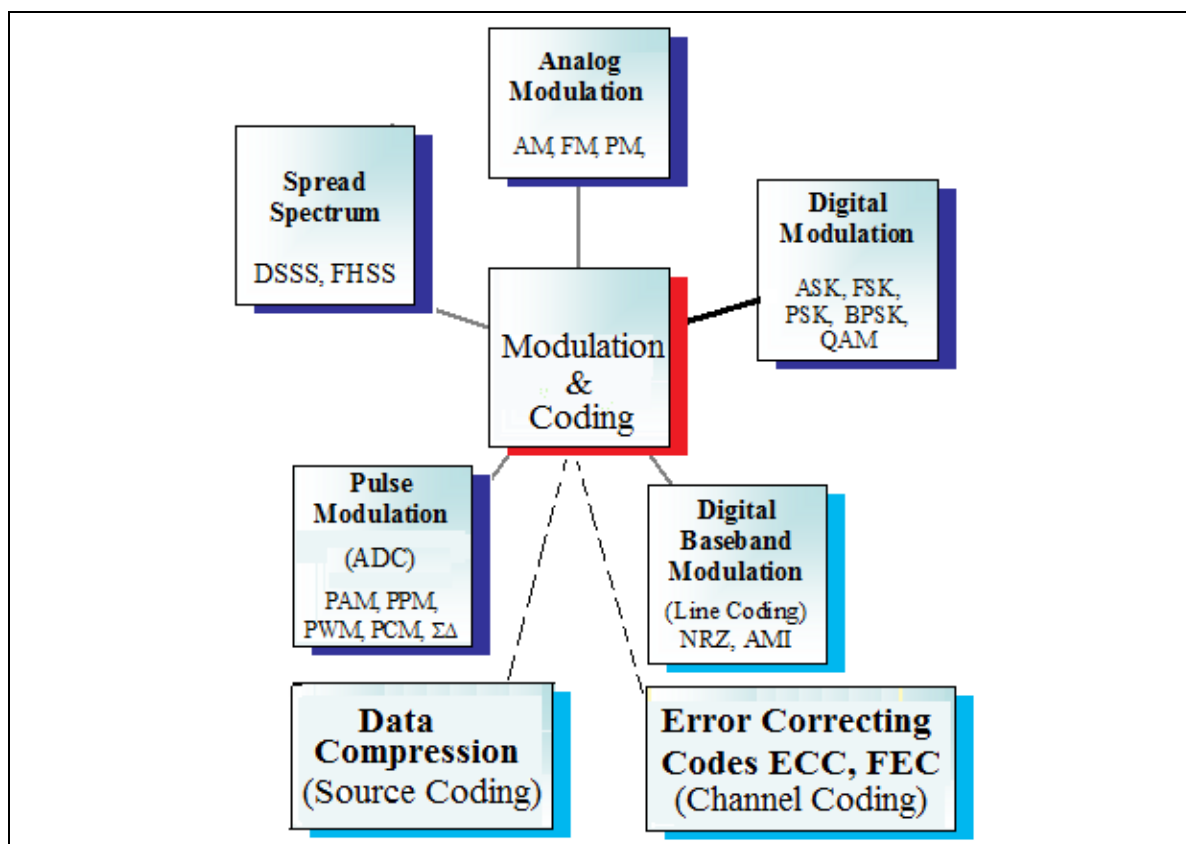


Fig. 1-6. Modulation and coding techniques

1-4. Baseband and Passband Signals.

One of the most important features of information signals is that they are generally low frequency. Sometimes this is due to the nature of data itself such as human voice which has frequency components from 300 Hz to 20 KHz. Other times, this is due to the hardware limitations inside a computer, which limit data transfer rates. Such low frequency information signals are called the **baseband** signals. Due to their low frequency content, the information signals have a lower spectrum, as shown in figure 1-6 below. Note that the frequency response of time-domain signal, $m(t)$, is obtained by taking its Fourier transform, such that $M(f) = \mathcal{F}\{m(t)\}$. There are a lot of low frequency components and the one-sided spectrum is located near the zero frequency. The hypothetical signal in figure has 5 sinusoids, all of which are fairly close to zero. The frequency range of this signal extends from zero to a maximum frequency of f_m . We say that this baseband signal has a **bandwidth** of f_m .

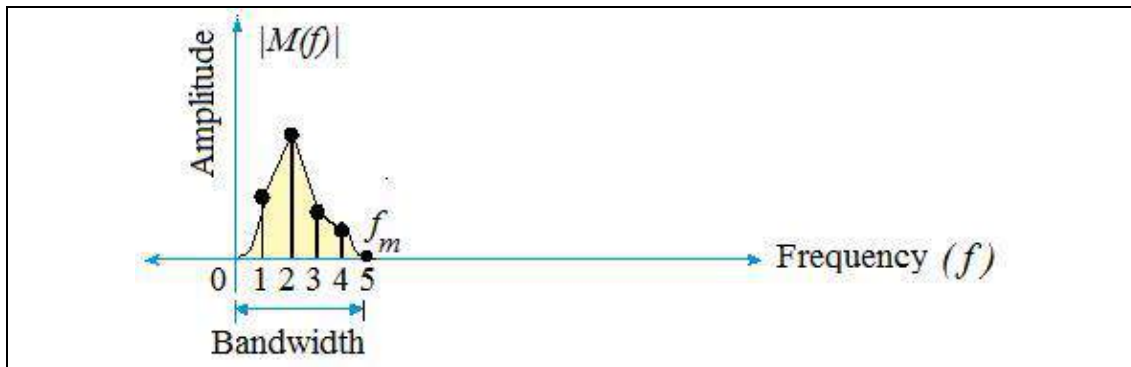


Fig. 1-7. Spectrum of a baseband signal.

Now assume this baseband signal is modulated, which means we are going to transfer it to a higher (usually much higher) frequency. The simplest type of modulator for nearly **all** modulation schemes is called the **product modulator** consisting of a multiplier or a mixer and a band-pass filter. Let's modulate the above signal using a product modulator, where $m(t)$ is the low frequency message signal and $c(t)$ is the high frequency **carrier** signal. The modulator takes these two signals and multiplies them, as follows:

$$s(t) = m(t) \cdot c(t) \quad (1-1)$$

The frequency domain representation of a product modulator or a mixer has a curious quality that it produces sums and differences of the frequencies of the two input signals in both the positive and negative frequency domains. $\pm (f_c \pm f_m)$.

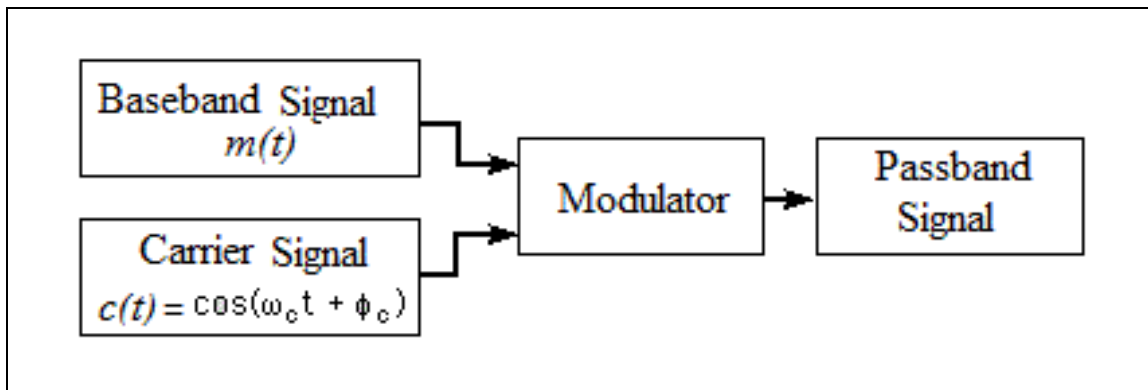


Fig. 1-8. Transformation of a baseband signal to a passband signal, by modulation

As shown in figure 1-9, the two-sided spectrum of the signal is shifted up to the plus and minus carrier frequency. This signal is now called the **passband** signal. Note that in figure 1-7, the passband spectrum has two parts (around f_c) that are identical. The upper part of the passband spectrum above carrier is called *upper sideband* and the one below is called the *lower sideband*.

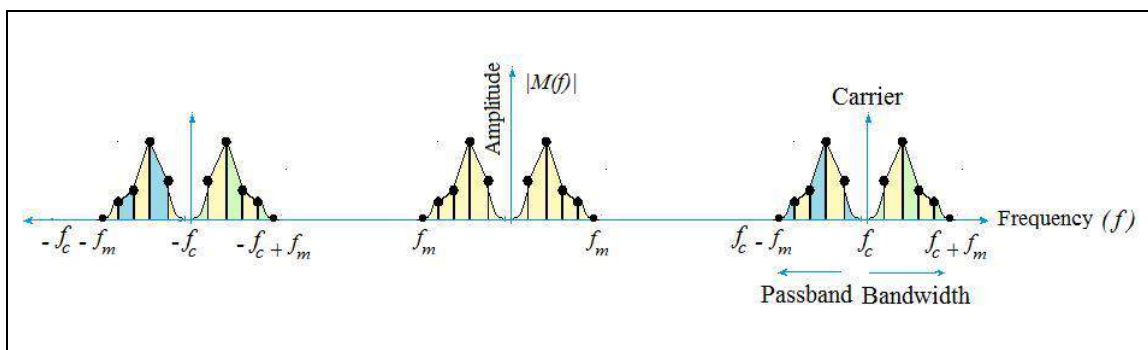


Fig. 1-9. Double-sided spectrum of a passband signal.

1-5. Communication Channels

The communication **channel** is the physical medium that is used to send the signal from the transmitter to the receiver. Thus, all communication systems incorporate a physical channel, which is intrinsically **analog**. The communication channel may be twisted pair copper wires, coaxial cables, fiber optic cables, or free space for radio communication. In digital computers, we consider the computer storage as communication channel.

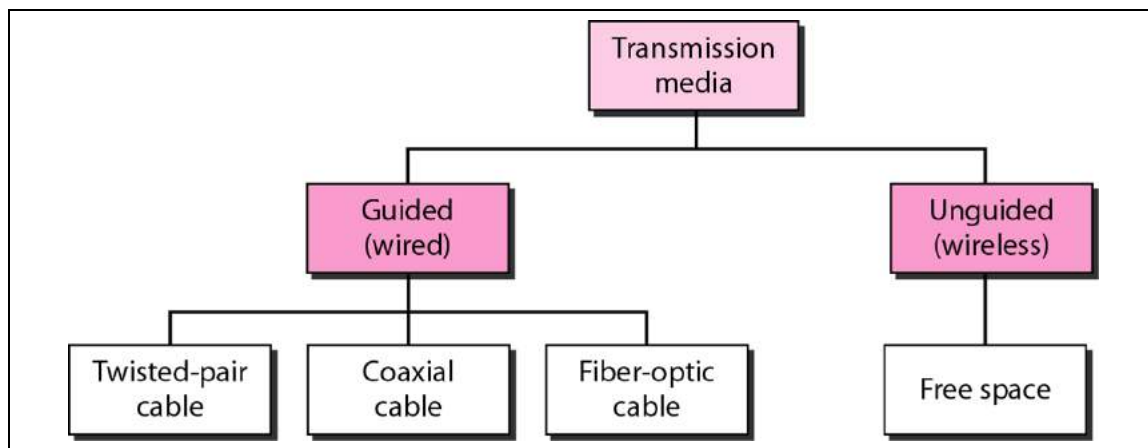


Fig. 1-10. Types of transmission media (communication channels).

An **ideal channel** will pass a transmitted signal to the receiver **without any attenuation or distortion or noise**. However, the real world channels are stochastic in nature and never ideal. Therefore, communication channels are always subject to thermal noise, distortion, amplitude attenuation, echoes, and other interfering impairments.

1-5.1. AWGN Channel

Different channels have different characteristics and the transmitter and receiver must be designed to deal with a particular channel. In this book, we consider the basic channel; which is subject to additive white Gaussian noise (AWGN).

1-5.2. Fading Channel

A fading channel is a communication channel that experiences fading. In wireless systems, fading is due to **multipath** propagation. Fading channel models are often used to model the effects of electromagnetic transmission of information over the air in cellular networks and broadcast communication. Mathematically, fading is usually modeled as a time-varying random change in the amplitude and phase of the transmitted signal. The effects of fading can be combated by using **diversity** techniques to transmit the signal over multiple channels that experience fading and coherently combining them at the receiver.

1-5.3. Multipath & Fading Problems

The transmission and reception of RF waves at high frequencies are confronted with some unwanted effects. Multipath is a common problem in radio communications, which cause errors and **fading** of received signals at the receiver side. As the name indicates, this is the situation where signals may travel from the transmitter to the receiver by more than one path.

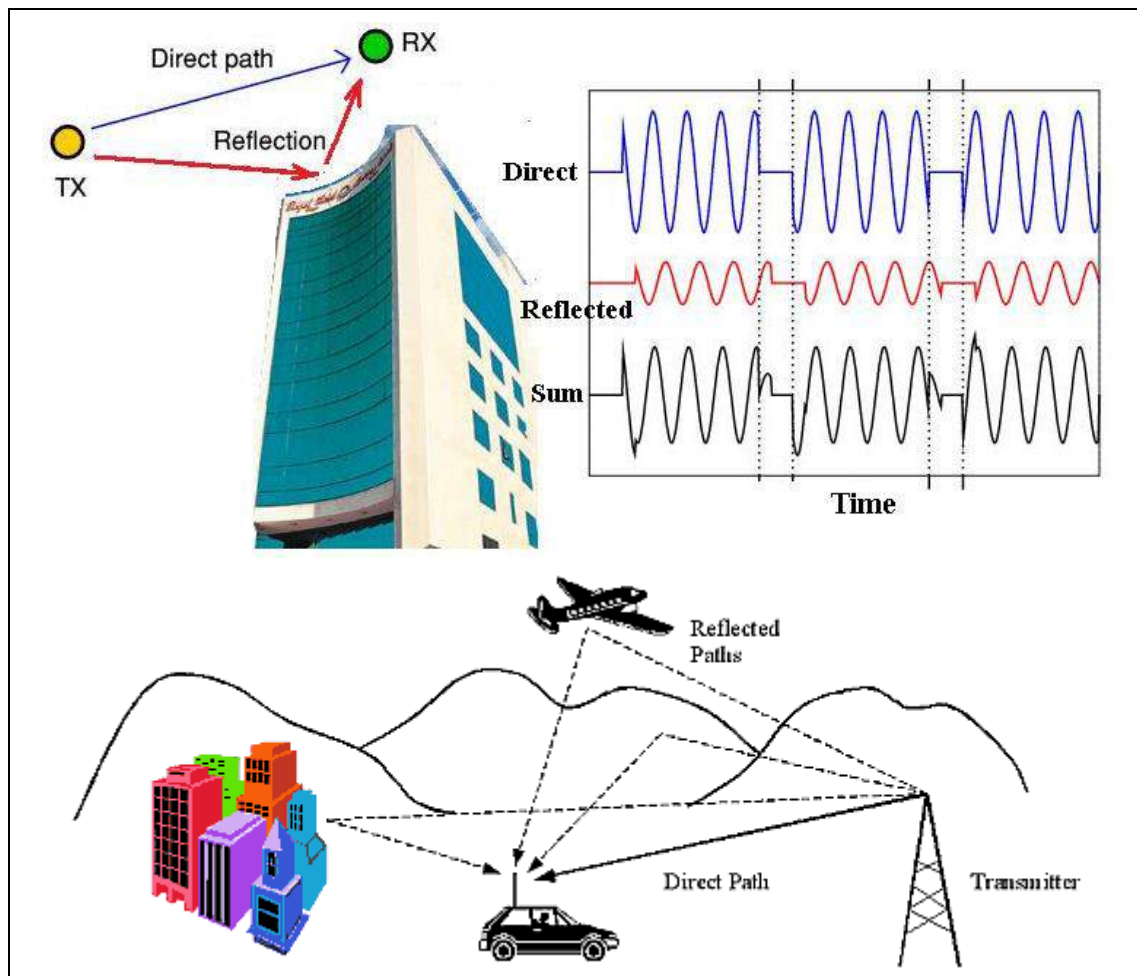


Fig.1-11. Illustration of the effect of multipath

For example, we may get one signal path in a straight line from the transmitter to receiver, and another via a reflection from a hill or tall building. When these paths have different lengths their contributions to the received results arrive with different propagation delays. This situation is illustrated in the figure 1-11. The relative phase of multiple reflected signals can cause constructive or destructive interference at the receiver. This is experienced over very short distances (typically at half wavelength distances), thus is given the term fast fading. These variations can vary from 10-30dB over a short distance. The **Rayleigh** distribution

is commonly used to describe the statistical time varying nature of the received signal power. It describes the probability of the signal level being received due to fading.

1-6. Noise and Electromagnetic Interference

Noise is any electrical signal present in a system other than the desired signal. The significance of the noise analysis of a circuit is the limitation it places on the smallest input signal that can be distinguished and treated. This doesn't apply to internal distortion, which is a by-product due to nonlinearities of electronic components. All electrical systems have noise. However, this noise is not a problem if it did not interfere with system performance. The noise sources are:

- 1- Man-made noise, due to motors, switches, digital electronics, radio transmitters, which produce electromagnetic interfering (**EMI**) signals,
- 2- Natural noise sources, like sunspots and lightning, cosmic rays,
- 3- Intrinsic noise sources, due to unknown fluctuations of the system, like **thermal noise** and **shot noise** or **flicker noise**

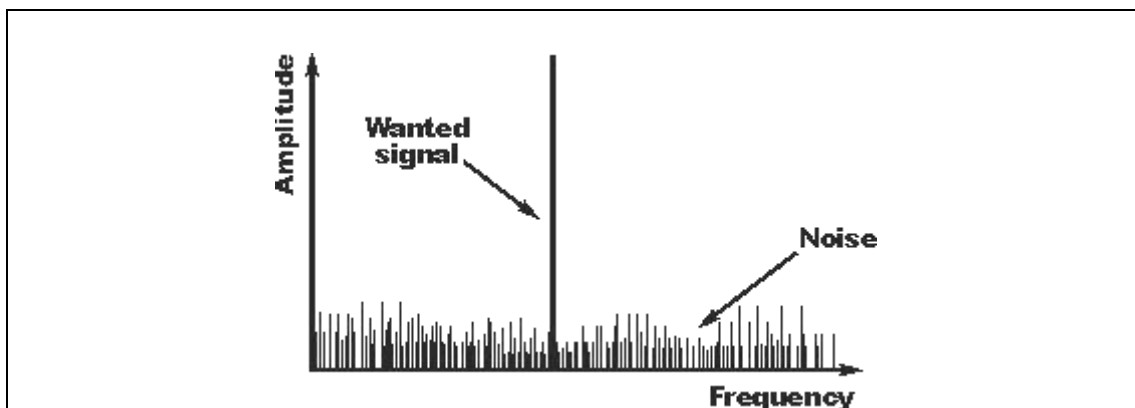


Fig. 1-12. A communication system, subjected to noise attack

Man-made noise is noise originated by human technologies. It is strongly dependent on the distance from the sources (power lines, radio, TV communication installations, etc...) that can be very variable, on frequency and on emitted power. Man-made sources are mainly located in business, industrial and residential areas as well as in rural areas.

The natural noise has many sources, such as lightning. You may notice that when lightning strikes, nearby magnetic compass needles may be seen to jerk in response to the sudden electrical discharge and subsequent emission of electromagnetic fields.

1-6.1. Types of Noise

Thermal noise (**Johnson** noise or **Nyquist** noise) is the electronic noise generated by the thermal agitation of the charge carriers (electrons) inside electrical conductor at equilibrium, which happens regardless of any applied voltage. Thermal noise is approximately **white**, meaning that the power spectral density is equal throughout the frequency spectrum. Additionally, the amplitude of the signal has very nearly a **Gaussian probability** density function

Shot noise, arising from the discreteness of charge quanta, is also a sort of white noise.

Flicker noise is a type of electronic noise with a $1/f$ dependence. It is therefore often referred to as **$1/f$ noise**. It occurs in almost all electronic devices, and results from a variety of effects, such as impurities in a conductive channel, generation and recombination noise in semiconductor devices. It is always related to a direct current. In electronic devices, it is a low-frequency phenomenon, as the higher frequencies are overshadowed by white noise from other sources. In oscillators, however, the low-frequency noise is mixed up to frequencies close to the carrier which results in oscillator **phase noise**. It should be noted that noise cannot be totally eliminated, but its interference may be reduced in digital systems using special techniques, of digital modulation and channel coding.

1-6.2. Modeling of Noise

In a communication system, the white thermal noise source may be represented, using an equivalent input noise voltage v_n and source resistance R , as shown in figure 1-13, the mean square voltage of the **Johnson noise**, produced by a resistor R , is given by:

$$\overline{v_n^2} = 4 k T.R.B \quad (1-2a)$$

where k is Boltzmann's constant (1.38×10^{-23} J/K), T is the resistor temperature in Kelvin, R is its resistance in Ohms and B is the bandwidth (in Hz) over which the noise voltage is observed. Note that the mean square noise voltage is sometimes referred to as the noise power. For a given resistor, R , we can maximize this by matching the noise source resistance and the subsequent system input resistance to get the maximum available noise power (N_{max}),

$$N_{\max} = \frac{\overline{v_n^2}}{4R} = 4kTB \quad (1-2b)$$

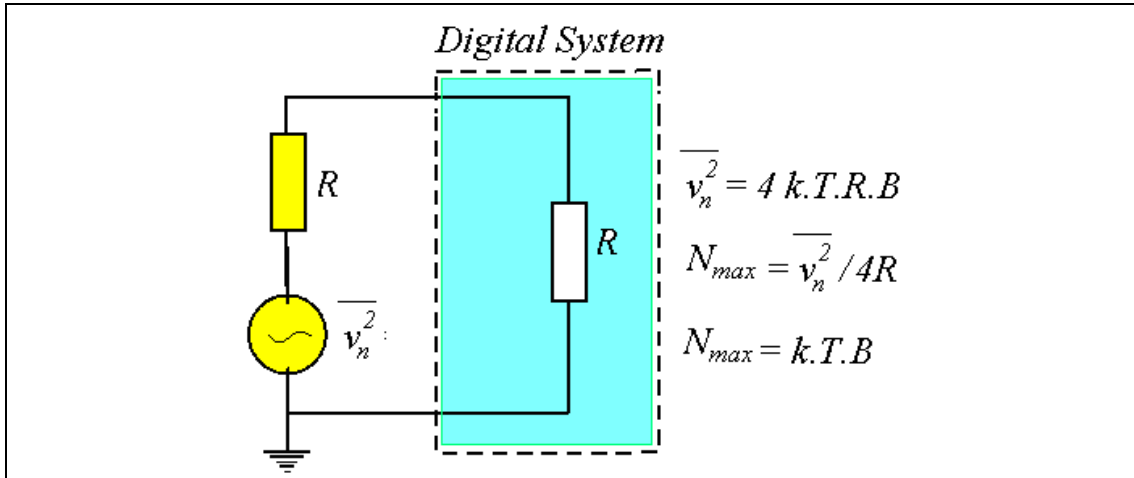


Fig. 1-13. Noise source model

The **Noise Power Spectral Density** (NPSD) at any frequency is defined as the noise power in a 1 Hz bandwidth at that frequency. Putting $B = 1$ into the above equation, we can see that Johnson (thermal) noise has a maximum available NPSD of just $k_B T$.

$$S_v \text{ (Thermal Noise)} = N_{\max} / B = kT \quad [\text{W/Hz}] \quad (1-3a)$$

This means that Johnson noise has an NPSD which is white and doesn't depend upon the fluctuation frequency. Note that some references refer to the NPSD by the symbol N_o , rather than S_v , so that the single sided power spectral density $N_o = kT$. Note also that the double-sided power spectral density of white noise is then given by:

$$\text{Double-sided NPSD} = \frac{1}{2} N_o = \frac{1}{2} kT \quad [\text{W/Hz}] \quad (1-3b)$$

However the NPSD does fall at extremely high frequencies because the total noise power is always finite.

Shot noise is due to discreteness of the electronic charge arriving at any anode giving rise to impulses of current. The current noise power spectrum is given by:

$$S_I = 2eI \quad [\text{A}^2/\text{Hz}] \quad (1-4a)$$

where I is the average flowing current and e is the electronic charge. This

means the shot noise is white, with constant spectral density, over the whole bandwidth of the system. It worth noting that electronic noise levels are often quoted in units of Volts per root Hertz [$V/\sqrt{\text{Hz}}$] or Amps per root Hertz. [$A/\sqrt{\text{Hz}}$]. In practice, because noise levels are low, the actual units may be [$\text{nV}/\sqrt{\text{Hz}}$].

Unlike Johnson or shot noise, which depend upon simple physical parameters (the temperature and current level respectively), **the flicker noise** (or $1/f$ noise) is strongly dependent upon the details of the particular system. In fact the term ' $1/f$ noise' covers a number of noise generating processes, some of which are poorly understood. For this form of noise the NPSD, S_f , varies with frequency approximately as follows:

$$S_f \sim 1/f^n \quad (1-4b)$$

where the value of the *index*, n , is typically around 1 but varies from case to case over the range, $1/2 < n < 2$. The following figure depicts the power spectral density of the thermal (Johnson) noise, the shot noise and the flicker noise. Note that the first two types are white, and have a constant spectral density over the whole bandwidth of any system.

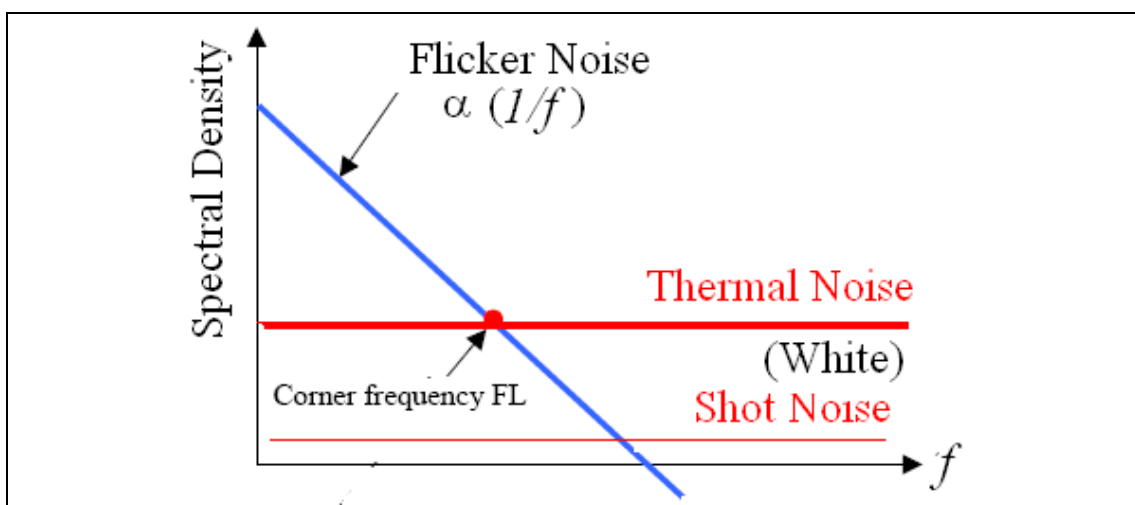


Fig. 1-14. Spectral density of thermal, shot (white) and flicker noise.

1-6.3. Signal to Noise Ratio

In a communication system, signal detection is more difficult in the presence of noise. The noisiness of a signal is specified by the signal-to-noise ratio (**SNR**). Signal-to-noise ratio is defined as the power ratio between a signal (meaningful information) and the background noise (unwanted signal):

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}}, \quad (1-5a)$$

where P is average power. Both signal and noise powers must be measured at the same or equivalent points in a system, and within the same system bandwidth. In decibels, the SNR is defined as

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) = P_{\text{signal,dB}} - P_{\text{noise,dB}}, \quad (1-5b)$$

For example, an average quality stereo tape will usually have a signal-to-noise ratio of about 60 dB to 70 dB. On the other hand, CD players usually produce a signal-to-noise ratio of 100 dB or higher

Usually, the noise has zero mean, which makes its power equal to its variance. Therefore, if both the signal (s) and the noise (N) are considered as random variables, then the SNR may be defined as: $\text{SNR} = s^2/\sigma_N^2$, with σ_N^2 is the variance of the noise.

1-6.4. Noise Figure and Noise Temperature

The noise factor (F) of a given system (device or circuit) is a measure of the degradation of the signal to noise ratio, across the system. The noise factor is defined as the numerical ratio of signal-to-noise ratios (S/N) in output and input of this system:

$$F = (S/N)_{\text{in}} / (S/N)_{\text{out}} \quad (1-6a)$$

The noise figure (**NF**) is defined as follows:

$$\text{NF} = 10 \log F = (S/N)_{\text{in}} - (S/N)_{\text{out}} \quad (1-6b)$$

where (S/N) ratios are in dB here. The noise factor of a device is related to its noise temperature (T_n) via the following relation:

$$F = 1 + T_n/T_o \quad (1-6c)$$

where T_o is the physical temperature of the component (usually 290 K). Devices without gain (e.g., attenuators) have a noise figure equal to their **attenuation** L (in dB) when their physical temperature equals T_o . The so-called first **Friis** formula is used to compute the noise figure or noise temperature of a system composed of a number of cascaded stages in a communication system. This will be explained in Chapter2 of this book.

1-7. Basics of Information Theory

Information theory is a branch of information engineering involving the calculations of **fundamental limits** of data communication and data compression. Applications of information theory include lossless data compression (e.g. ZIP files), lossy data compression (e.g. MP3), and channel coding for data transmission (e.g. over DSL lines). The field is multidisciplinary of computer science, mathematics, statistics, and electrical engineering. Its impact has been crucial to success of the outer space missions, the invention of the CD, the mobile phones, the development of the Internet, the language engineering and human perception, and numerous other fields. Important sub-fields of information theory are **source coding**, **channel coding**, algorithmic information theory, and information measurements. Information theory is considered to have been founded in 1948 by Claude Shannon in his seminal work, "A Mathematical Theory of Communication".

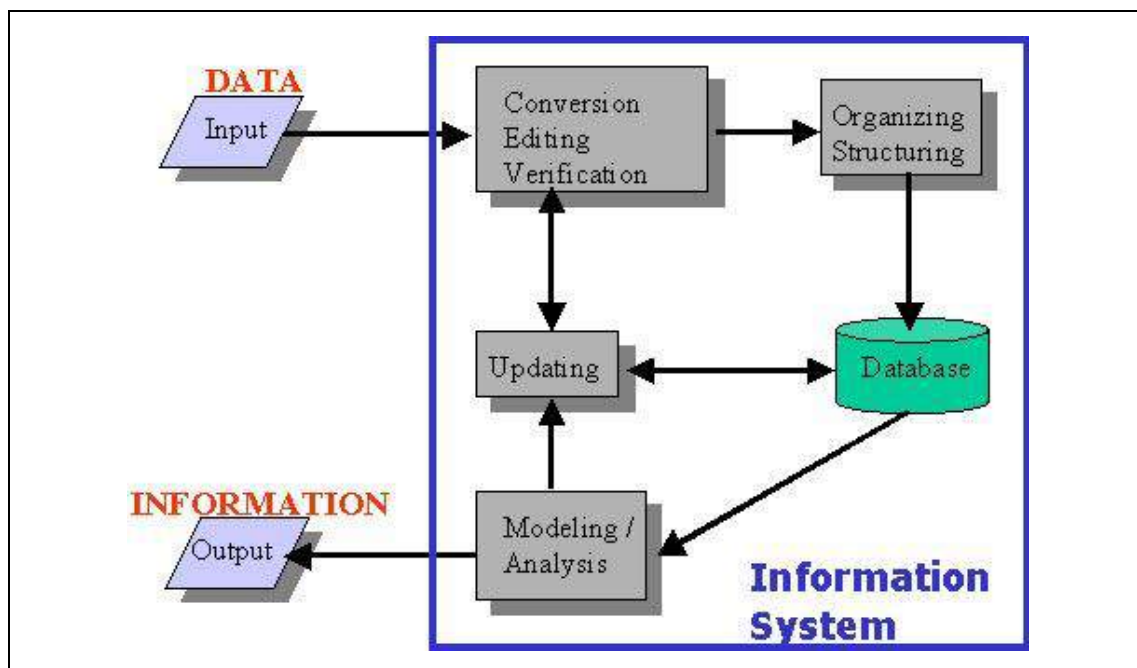


Fig. 1-15. Information versus Data. Data is raw material for information

The central paradigm of classical information theory is the engineering problem of the transmission of information over a noisy channel. The most fundamental results of this theory are the source coding theorem, which establishes that, on average, the number of *bits* needed to represent the result of an uncertain event is given by its entropy; and Shannon's noisy-channel coding theorem, which states that *reliable* communication is possible over *noisy* channels provided that the rate of communication is below a certain threshold called the **channel capacity**.

1-7.1. Quantity of Information

Any system that generates successive messages can be considered a source of information. The most important quantities of information are entropy, the information in a random variable, and mutual information, the amount of common information between two random variables.

The most common **unit of information** is the **bit**, based on the binary logarithm (\log_2). Other units include the **nat**, which is based on the natural logarithm (\log_e), and the **Hartley**, which is based on the common logarithm (\log_{10}). In what follows, an expression of the form $p \log_2(p)$ is considered by convention to be equal to zero whenever p is zero. This is justified because $\lim_{p \rightarrow 0} p \cdot \log_2(p) = 0$ for any logarithmic base.

Note 1-3: Joint and Conditional Probability

Information theory is based on probability theory and statistics. The probability of a certain events is its frequency of occurrence. Let's denote the probability of occurrence of the event A , by $p(A)$. The probability of any event is bounded between 0 and 1, that's:

$$0 \leq p(A) \leq 1.$$

If A and B are mutually exclusive events, with individual probabilities $p(A)$ and $p(B)$, respectively, then the probability of their union $p(A \text{ or } B)$ is given by:

$$p(A+B) = p(A) + p(B) - p(AB)$$

where $P(AB)$ is the joint probability of A and B , i.e., $P(A \text{ and } B)$.

If the event B depends on the event A , then the conditional probability of B , given that A occurred, is given by:

$$p(B/A) = p(AB) / p(A).$$

One can easily prove that

$$p(A/B) = p(B/A) \cdot p(B) / p(A).$$

Note 1-4: Random Variables & Probability Distribution Function

In probability theory, a **random variable** is a quantity whose values are random and to which a probability distribution function (**PDF**) is assigned. The PDF of a random variable, $f(X)$, is often characterized by a small number of parameters, which have practical interpretation. For example, it is often enough to know its average value.

This is captured by the mathematical concept of **expected value** of a random variable (X), denoted $E[X]$ which is given by:

$$E(X) = \int X \cdot f(X) dX.$$

Once the expected value is known, one can then ask how far from this average value the values of X typically are, a question that is answered by the **variance** $\delta^2[X]$ and **standard deviation** $\delta[X]$ of a random variable.

1-7.2. Information Entropy

A key measure of information that comes up in the theory is known as **information entropy**. Intuitively, **entropy quantifies the uncertainty involved in a random variable**.

The **entropy**, H , of a discrete random variable X is a measure of the amount of *uncertainty* associated with the value of X .

Entropy of a Bernoulli trial as a function of success probability, often called the **binary entropy function**, $H_b(p)$. The entropy is maximized at 1 bit per trial when the two possible outcomes are equally probable, as in an unbiased coin toss. Suppose one transmits 1000 bits (0s and 1s). If these bits are known ahead of transmission (certain value with absolute probability), logic dictates that no information has been actually transmitted. If, however, each is equally and independently likely to be 0 or 1, then 1000 bits have been transmitted (in the information theoretical sense). Between these two extremes, information can be quantified as follows. If x is the set of all messages that X could be, and $p(x) = p(X = x)$, then the entropy of X is defined as follows:

$$H(X) = E_X [I(x)] = - \sum_{x \in X} p(x) \log_2 p(x) \quad (1-7a)$$

Here, $I(x)$ is the **self-information**, which is the entropy contribution of an individual message. Note that, like information, the units of entropy are **bits**, when we consider the binary logarithm.

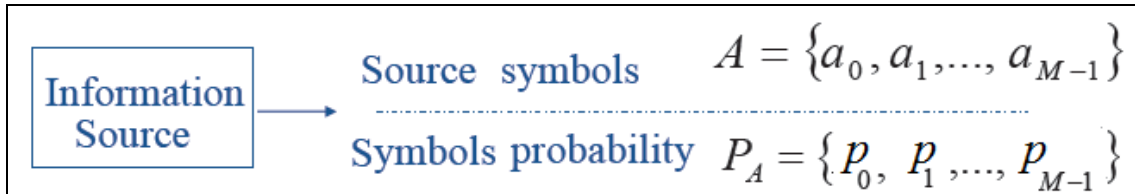
An important property of entropy is that it is maximized when all the messages in the message space are equiprobable—i.e., most unpredictable—in this case: $H(X) = \log_2 |X|$. The special case of entropy for a random variable with two outcomes (e.g., 0 and 1) is the **binary entropy function**:

$$H_b(p) = -p \log_2 p - (1-p) \log_2 (1-p) \quad (1-7b)$$

Entropy is usually expressed by the average number of bits needed for storage or communication. **Entropy** may be also defined in terms of the probabilistic behavior of a source of information. It also represents a limit of the minimum average code length for lossless source coding.

Example 1-1.

Consider a discrete memoryless information source with source alphabet $A = [a_0, a_1, a_2]$ with respective probabilities $p_0 = 1/4, p_1 = 1/4, p_2 = 1/2$. Calculate the entropy of the data source.



Solution:

$$\begin{aligned}
 H(A) &= p_0 \log_2 (1/p_0) + p_1 \log_2 (1/p_1) + p_2 \log_2 (1/p_2) \\
 &= 1/4 \log_2 (4) + 1/4 \log_2 (4) + 1/2 \log_2 (2) = 3/2 \text{ bits}
 \end{aligned}$$

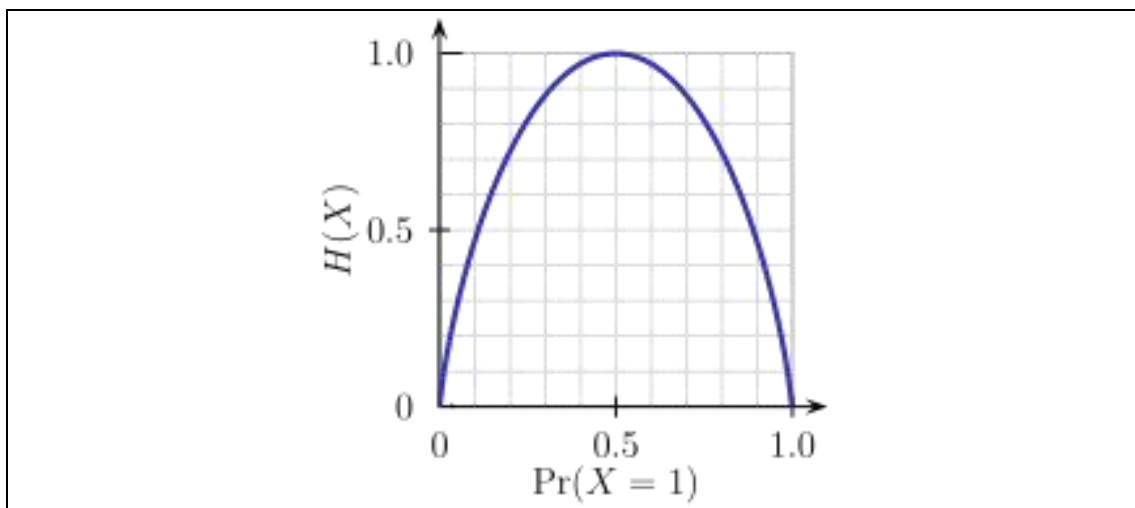


Fig. 1-16. Information entropy versus probability

1-7.3. Joint Entropy

The **joint entropy** of two discrete random variables X and Y is merely the entropy of their pairing: $H(X,Y)$.

$$H(X, Y) = - \sum_{X,Y} p(x, y) \log_2 p(x, y) \quad (1-8a)$$

If X and Y are **independent**, then their joint entropy is the sum of their individual entropies. For example, if (X,Y) represents the position of a

chess piece — X the row and Y the column, then the joint entropy of the row of the piece and the column of the piece:

$$H(X, Y) = H(X) + H(Y) \quad (1-8b)$$

Despite similar notation, joint entropy is not the same as **cross entropy**.

1-7.4. Conditional Entropy (equivocation)

The **conditional entropy** of X given random variable Y (also called the **equivocation** of X about Y) is the average conditional entropy over Y :

$$\begin{aligned} H(X / Y) &= H(X) - H(X/Y) = H(X, Y) - H(Y) \\ &= - \sum_{y \in Y} p(y) \sum_{x \in X} p(x | y) \log_2 p(x | y) = \sum_{x, y} p(x, y) \log_2 \frac{p(y)}{p(x, y)} \end{aligned} \quad (1-9)$$

Because entropy can be conditioned on a random variable or on that random variable being a certain value, care should be taken not to confuse these two definitions, the former of which is more common.

1-7.5. Mutual Information (Trans-information)

Mutual information measures the amount of information that can be obtained about one random variable by observing another. It is important in communication where it can be used to maximize the amount of information shared between sent and received signals. The mutual information of X relative to Y is given by:

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x) p(y)} \right), \quad (1-10a)$$

where SI is the point-wise mutual information. A basic property of the mutual information is that

$$I(X; Y) = H(X, Y) - H(X/Y) \quad (1-10b)$$

That is, knowing Y , we can save an average of $I(X; Y)$ bits in encoding X compared to not knowing Y . **Mutual information** is symmetric such that:

$$I(X; Y) = H(X) + H(Y) - H(X, Y) \quad (1-10c)$$

Mutual information is a measure of how much, on the average, the probability distribution on X will change if we are given the value of Y .

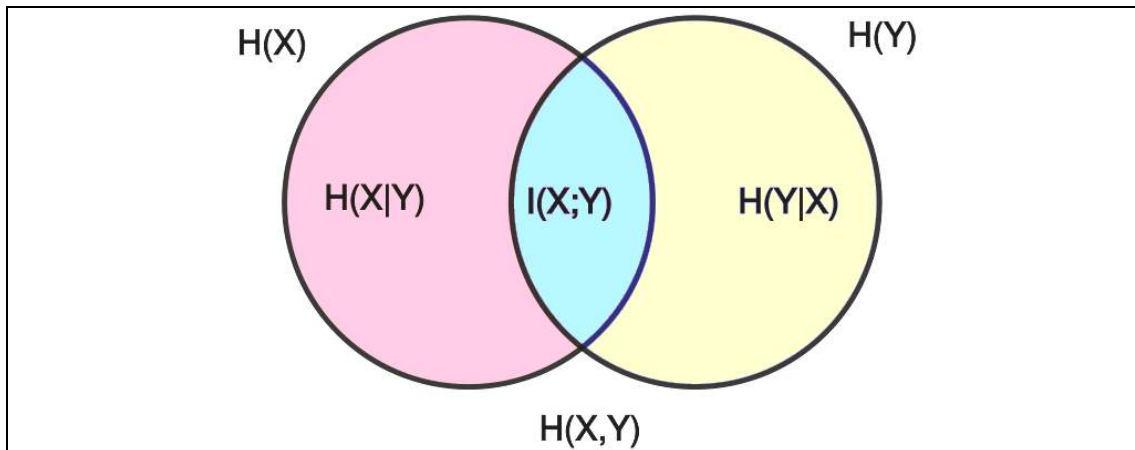


Fig. 1-17. Mutual information and its relation to the entropy of two random variables

1-7.6. Channel Capacity

One of the most important practical questions which arise when we are designing or using an information transmission or processing system is, "What is the *Capacity* of this system? How much information can it transmit or process in a given time?" Technically speaking, we need to know how much information one can hope to communicate over a noisy (or imperfect) channel.

The **channel capacity** is the maximum average information that can be sent per channel use. Therefore, channel capacity represents a fundamental limit of the **maximum bit rate** for reliable (error-free) communication over a noisy channel.

Consider the communications system over a discrete channel. A simple model of the process is shown in figure, below:

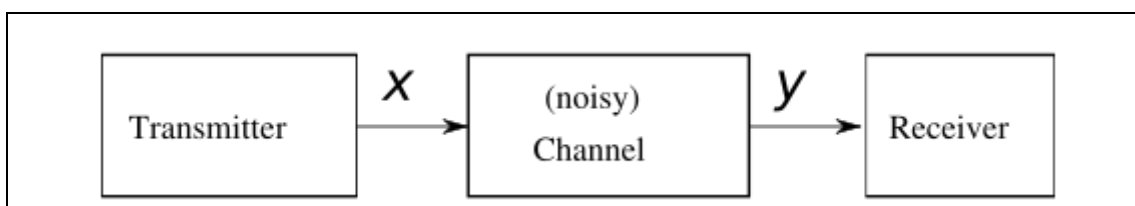


Fig. 1-18. A communication system, with noisy channel

Here **X** represents the space of **transmitted messages**, and **Y** the space of **received messages** during a unit time over our channel. Let $p(y | x)$ be the conditional probability distribution function of Y given X . We will consider $p(y | x)$ to be an inherent fixed property of our communications channel (representing the nature of the **noise** of our channel). Then the joint distribution of X and Y is completely determined by the channel and

by the choice of $f(x)$, the marginal distribution of messages we choose to send over the channel. Under these constraints, we would like to maximize the rate of information, or the signal, we can communicate over the channel. The appropriate measure for this is the mutual information, and this maximum mutual information is the **channel capacity**:

$$C = \max I(X;Y) \quad (1-11a)$$

For a **continuous-time analog** communications channel subject to Gaussian noise, the channel capacity is given by Shannon's equation.

1-7.7. Shannon Equation.

The **Shannon-Hartley equation** allows us to precisely determine the channel capacity, of a signal channel in the presence of noise, as follows:

$$C = B \log_2 (1+SNR) \quad (1-11b)$$

where C is the channel capacity in [bit/sec], SNR is the signal-to-noise ratio and B is the channel bandwidth in [Hz]. Thus, channel capacity is proportional to bandwidth and the logarithm of the signal to noise ratio plus one. What this means is that the more bandwidth and the better the signal to noise ratio, the more bits per second (C) you can push through a channel. This expression represents the maximum possible rate of information transmission through a given channel or system. The maximum rate we can transmit information is set by the bandwidth, the signal level, and the noise level. C is therefore called the channel's information *Capacity*.

Example 1-2.

If the SNR is 20dB, and the bandwidth available is 4 kHz, which is appropriate for telephone communications, then $C = 4 \log_2(1 + 100) = 4 \log_2(101) = 26.63$ kbit/s. Note that the value of $S/N = 100$ is equivalent to the SNR of 20 dB

The last Shannon equation is sometimes called **Shannon-Hartley equation**, because it is based on the old Hartley formula of information rate $R_b = 2B \log_2(M)$, where M was defined as the maximum number of pulse levels that can be transmitted and received reliably over a communications channel. Here, the voltage ratio is replaced by the square root of the power ratio. $M = \sqrt{1+SNR}$. An important result of the information theory is that if the channel capacity is greater than the

source encoder efficiency, then almost error-free communication over a noisy channel may be reached using the proper error correcting codes.

The so-called **channel coding** techniques are concerned with finding such nearly optimal codes that can be used to transmit data over a noisy channel with a small coding error at a rate near the channel capacity.

1-7.8. Coding Theory

Coding theory is concerned with finding explicit methods, called *codes*, for increasing the efficiency and reducing the net error rate of data communication over a noisy channel to near the maximum possible limit that Shannon predicted. These codes can be roughly subdivided into three categories:

- 1- **Source Coding** (Data compression) techniques,
- 2- **Channel Coding** (Error-correction) techniques and
- 3- **Line Codes**.

Source coding is a mapping from a sequence of symbols of an information source to a reduced sequence of bits such that the source symbols can be recovered. There are two formulations for the data compression (**source coding**) problem:

- Lossless data compression: the data must be reconstructed exactly;
- Lossy data compression: allocates bits needed to reconstruct the data, within a specified fidelity level measured by a distortion function. In a common communication system, at the transmitting point there is first source coding, then channel coding.

Channel Coding is concerned with finding such nearly optimal codes that can be used to transmit data over a noisy channel with a small coding error at a rate near the channel capacity. Channel coding should not be confused with **source coding**, which is the elimination of redundancy in order to make efficient use of storage space and/or transmission channels. Channel coding should also not be confused by **line coding**, which is the coding performed in order to adapt the transmitted signal to the (electrical) characteristics of a transmission channel.

Line codes are presented in Chapter 3, while channel codes are illustrated in Chapter 6. Concepts, methods and results from coding theory and information theory are also widely used in **cryptology** and **cryptanalysis**. Source codes are discussed in Chapter 7 and data cryptography is discussed in Chapter 8 of this book.

1-8. Spectral Efficiency

Most communications systems fall into one of three categories: **bandwidth efficient**, **power efficient**, or **cost efficient**. Bandwidth efficiency describes the ability of a modulation scheme to accommodate data within a limited bandwidth. Power efficiency describes the ability of the system to reliably send information at the lowest practical power level. In most systems, there is a high priority on bandwidth efficiency.

Spectral efficiency, or **bandwidth efficiency**, refers to the amount of information that can be transmitted over a given bandwidth in a specific communication system. It is a measure of how efficiently a limited frequency spectrum is utilized by the physical layer protocol, and sometimes by the media access control (the channel access protocol).

The **spectral efficiency** (η_s) of a digital communication system is measured in bit/s/Hz. It is the net bit-rate (R_b) divided by the bandwidth (B) in hertz of a communication channel or a data link.

$$\eta_s = \frac{R_b}{B} \quad (1-12)$$

For example: A transmission technique using 1 kHz of bandwidth to transmit 1000 bits/s has a spectral efficiency of $\eta_s = R_b/B = 1$ (bit/s)/Hz. Examples of numerical spectral efficiency values of some common communication systems can be found in the following table. Alternatively, one can define the spectral efficiency of a transmission media (channel) as follows:

$$\eta_s = C/B = \log_2(1+SNR) \quad (1-13)$$

where the bit rate R_b is replaced with the maximum channel capacity C ., as define by Shannon's theorem.

The Need for a Spectrum Efficient System

In order to illustrate the need for efficient spectrum usage for a radio communications system, take the example where each user is allocated a channel. While more effective systems are now in use, the example will take the case of a classical analogue system. Each channel needs to have a bandwidth of around 25kHz to enable sufficient audio quality to be carried as well as enabling there to be a guard band between adjacent signals to ensure there are no undue levels of interference. Using this

concept it is only possible to accommodate 40 users in a frequency band 1 MHz wide. Even if 100 MHz were allocated to the system this would only enable 4000 users to have access to the system. Today cellular systems have millions of subscribers and therefore a far more efficient method of using the available spectrum is needed.

Table 1-3. Spectral efficiency of common communication systems.

Service	Standard	Bitrate R per channel (Mbit/s)	Channel Bandwidth B (MHz)	Spectral efficiency R/B (bit/s/Hz)
1G	AMPS	0.0096	0.030	0.32
2G	GSM 1993	0.104	0.2	0.52
2.75G	GSM + EDGE	Max 0.384	0.2	Max 1.92 Typical 1.00
3G	CDMA2000	Max 0.0096	1.2288	0.0078
3G	WCDMA	Max 0.384	5	Max 0.077
3.5G	HSDPA 2007	Max 14.4	5	Max 2.88
4G	LTE	Max 326.4	20	Max 16.32
Wi-Fi	IEEE 802.11n	Max 144.4	20	Max 7.22
WiMAX	IEEE 802.16	96	20 (1.75, 3.5, 7...)	4.8
Digital TV	DVB-T	Max 31.67 Typical 22.0	8	Max 4.0 Typical 2.8
Digital TV fiber Cable	256-QAM	38	6	6.33
ADSL2 downlink	OFDM	12	0.962	12.47

1-9. Principles of Queuing Theory

Queuing theory is the mathematical study of waiting lines, or queues. In queuing theory a model is constructed so that queue lengths and waiting times can be predicted. In communication systems, voice or data traffic queue up for transmission. A simple example is the telephone calls. In **data communication**, many jobs share one resource (eg. a printer). In principle only one job or data packet can use a resource at a time; all other jobs are waiting to use the resource in queues. Queuing theory is used to model the time that jobs (or packets) spend in the system queues.

The basic representation widely used in queuing theory is made up of symbols representing three elements: **input/service/number of servers**. For instance, using M for Poisson or exponential distribution, D for deterministic (constant), E_k for the Erlang distribution with scale parameter k , and G for general, we write:

$M/G/1$: Poisson arrivals, general service, single server

$E_k/M/1$: Erlang arrival, exponential service, single server

$M/D/s$: Poisson arrival, constant service, s servers.

The simplest and most widely used queuing model in communication centers is the **M/M/N** system, sometimes referred to as **Erlang-C**. The **M/M/N** model assumes, among other things, a steady-state environment in which **arrivals** conform to a Poisson process, service durations are exponentially distributed, and customers and servers are statistically identical and act independently of each other.

In order to understand the arrival process, assume the following example. The students arrive for a lecture randomly; also packets in data communication system arrive at a node in a random manner; If packets arrive at times t_1, t_2, t_3, \dots , then the variable $\tau_n = t_{n+1} - t_n$ is called **interarrival time** and forms a sequence of Independent and Identically Distributed (**IID**) random variables. The most common arrival process is the Poisson arrivals; this means that the interarrival times are IID and are exponentially distributed

In the **M/M/1** system, customers arrive according to Poisson Process

$$P(n) = (\lambda \tau)^n \exp(-\lambda \tau) / n! \quad (1-14)$$

where λ is the arrival rate (mean number of arrivals per sec) and τ is the mean response time.

Another notation, which is used to describe a queuing system, is expressed as: **a/b/m/K**, where

- **a** specifies the type of arrival process
- If **a** is specified by **M**, then the arrival process is Poisson and the interarrival times are **IID** exponential random variables
- **b** denotes the service time distribution;
- if **b** is given by **M**, then the service times are IID exponential (memory-less) random variables;
- If **b** is given by **G**, then the service times are IID according to some general distribution;
- **m** specifies the number of servers (channels),
- **K** denotes the maximum number of packets (calls) allowed in the system at any time

In general, we deal with the following queues for **data networks**,

- M/M/1
- M/M/1/K
- M/M/m
- M/M/m/m and
- M/G/1

If the interarrival times are exponentially distributed, with mean **1/λ**, the expected time to the next arrival is always **1/λ** regardless of the time since the last arrival.

1-10. Basics of Teletraffic Engineering

Traffic is the flow of information or messages throughout a network. The teletraffic theory use the knowledge of statistics including queuing theory, the nature of traffic, the traffic models, their measurements and simulations to make predictions and to plan telecommunication networks such as telephone networks or the Internet. These tools and knowledge help provide reliable service at lower cost. The performance of a data network depends on whether all origin-destination pairs are receiving a satisfactory service. The measurement of traffic in a network allows us to determine and maintain the quality of service (**QoS**) and in particular the grade of service (**GoS**) that the network operators promise their subscribers. A good application example of the teletraffic theory is in the design and management of a call center. Call centers use teletraffic theory to increase the efficiency of their services and profit through calculating how many operators are really needed at each time of the day. Queuing systems used in call centers have been studied as a science. For example completed calls are put on hold and queued until they can be served by an

operator. If callers are made to wait too long, they may lose patience and default from the queue (hang up), resulting in no service being provided. There exist several traffic models. The simplest **Teletraffic model** consists of a queuing system without loss and infinite number of servers

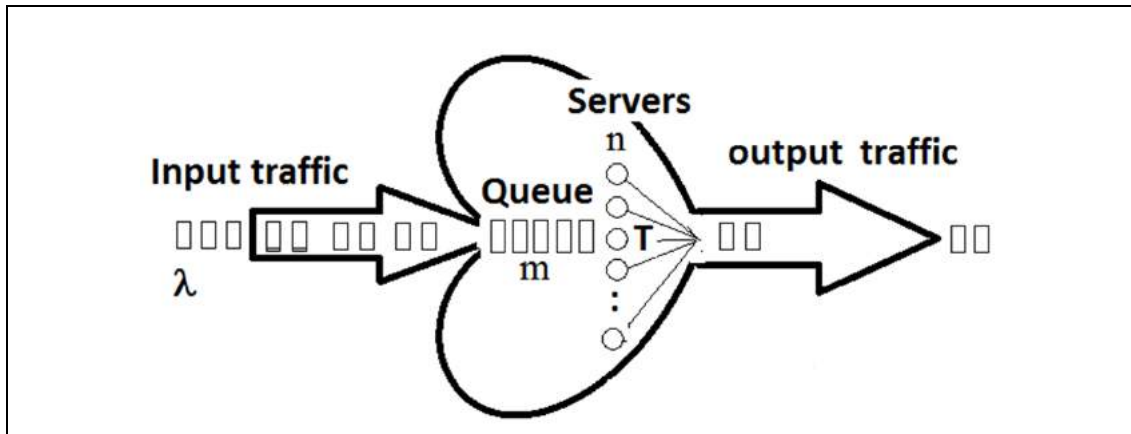


Fig. 1-19. Traffic model

The traffic models provide us with formulae that can be used to estimate the number of lines required in a network, or to a central office (PSTN exchange). A formula also exists to model queuing situations, and lends itself well to estimating the agent staffing requirements of call centers.

1-10.1. Teletraffic Models

The main Erlang traffic models are listed below:

i. Erlang B

This is the most commonly used traffic model, and sometimes called *the Erlang basic formula*. It is used to work out how many lines are required if the traffic figure (in Erlangs) during the busiest hour is known. The model assumes that all blocked calls are immediately cleared.

ii. Extended Erlang B (EEB)

This model is similar to Erlang B, but takes into account that a percentage of calls are immediately represented to the system if they encounter blocking (a busy signal). The retry percentage can be specified.

iii. Erlang C

This model assumes that all blocked calls stay in the system until they can be handled. This model can be applied to the design of call center staffing arrangements where, if calls cannot be immediately answered, they enter a queue.

You may wonder why the teletraffic models share the Erlang name. In fact, the Danish scientist Agner K. Erlang was a pioneer in the study of telecommunications traffic. He gained worldwide recognition for his work in teletraffic, and during the 1940s, the *Erlang* became the accepted unit of telecommunication traffic measurement.

1-10.2. The Erlang Unit

The **Erlang unit** is a measure of the voice call traffic density in a telecommunication system. Strictly speaking, an Erlang represents the continuous use of one voice path. In practice, it is used to describe the total traffic volume of one hour. According to Erlang, it is possible to express the amount of traffic, using the following **Erlang formula**:

$$A = \lambda \times T \quad (1-15)$$

where A is the traffic in Erlangs λ is the mean arrival rate of new calls and T is the mean duration of a connection or holding time. Using this simple Erlang function or Erlang formula, the traffic can easily be calculated. For example, if a group of user made 30 calls in one hour, and each call had average call duration of 5 minutes, then the number of Erlangs this represents is worked out as follows:

Minutes of traffic in the hour	=	number of calls x duration
Minutes of traffic in the hour	=	30 x 5
Minutes of traffic in the hour	=	150
Hours of traffic in the hour	=	150 / 60
Hours of traffic in the hour	=	2.5
Traffic figure	=	2.5 Erlangs

The **Erlang Basic Formula** relates the number of servers or system lines (n), the traffic intensity (A) and quality or grade of service (E)

$$E(n, A) = \frac{\frac{A^n}{n!}}{1 + A + \frac{A^2}{2!} + \dots + \frac{A^n}{n!}} \quad (1-16)$$

Actually, the formula provides the Grade-of-Service (**GoS**) which is the probability that a new arriving call is blocked because resources (servers, lines, circuits) are busy.

1-11. Communication System Standardization Organizations

Standardization is vital in telecommunications. Adopting international standards ensures seamless global communications and interoperability for next generation. Today, there exist so many communication system standardization organizations. For instance, the so-called International Telecommunication Union (**ITU**) is an International Standards Organization (**ISO**) standard that defines the frameworks for implementing telecommunication systems. The following figure depicts the different standardization organizations in the field of telecommunications and data networks.

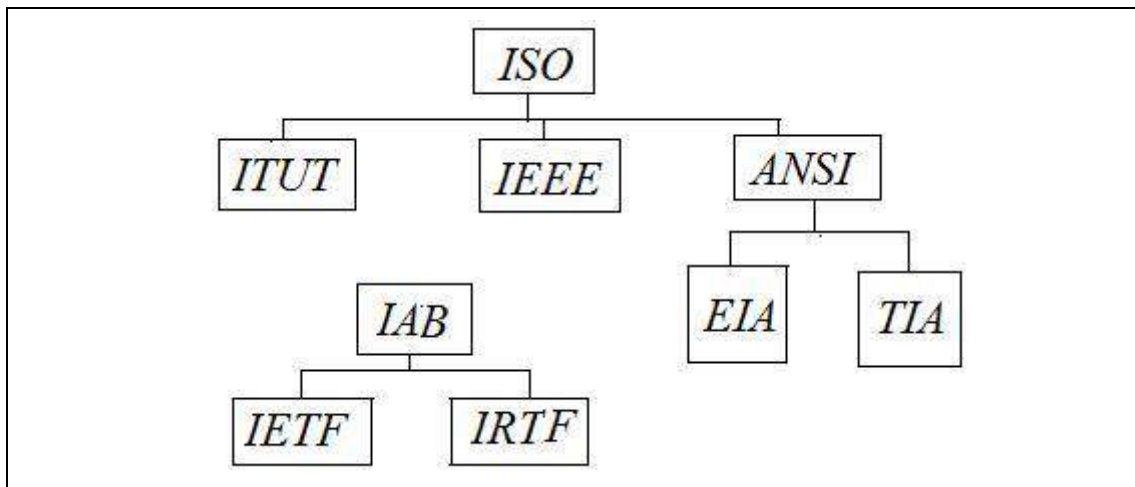


Fig. 1-20. .Communication System Standardization organizations.

ITU's role as creator of the world's most universally-recognized information communications standards dates back as far as the organization itself. Since its inception in 1865, the Union has been brokering industry consensus on the technologies and services that form the backbone of the world's largest, most interconnected man-made system. In 2007 alone, ITU's Telecommunication Standardization Sector (ITU-T) produced over 160 new and revised standards (ITU-T Recommendations), covering everything from core network functionality and broadband to next-generation services. ITU-T Recommendations are defining elements in information and communication technology (**ICT**) infrastructure.

The ITU has some regional groups, such as the pan Arab group (SG2 RG-ARB), which is maintained by the the Egyptian National Telecommunication Regulatory Authority (**NTRA**). The Telecommunication Industry Association (**TIA**) is accredited by the American National Standards Institute (**ANSI**) to develop voluntary industry standards for a wide variety of telecommunications products. TIA's Standards and

Technology Department is comprised of ten technology areas which sponsor more than 70 standards formulating groups. These technology areas are Mobile Private Radio, Steel Antenna Towers, Multi Function Peripheral Devices, Satellites, User Premises Equipment, Premises Cabling (both copper and fiber), Mobile Communications Systems, Terrestrial Mobile Multimedia Multicast, Vehicular Telematics and Healthcare. Each area is represented by engineering committees and subcommittees that formulate standards to serve the industry and users.

The Internet Architecture Board (**IAB**) is chartered both as a committee of the Internet Engineering Task Force (**IETF**) and the Internet Research Task Force (**IRTF**) as an advisory body of the Internet Society (ISOC). Its responsibilities include architectural oversight of IETF activities, Internet Standards Process oversight and appeal. The research groups (**IRTF**) works on topics related to Internet protocols, applications, architecture and technology.

1-12. Spectra of Electromagnetic Waves & RF Bands

The following figure depicts the entire electromagnetic spectrum. The boundaries between far infrared light (FIR), terahertz (THz) radiation, microwaves (μW), ultra-high-frequency (UHF) and very-high frequency (VHF) waves are fairly arbitrary. Both IEC standard 60050 and IEEE standard 100 define microwave frequencies starting at 1 GHz (30 cm wavelength). Electromagnetic waves longer than microwaves are conventionally called radio frequency (RF) waves.

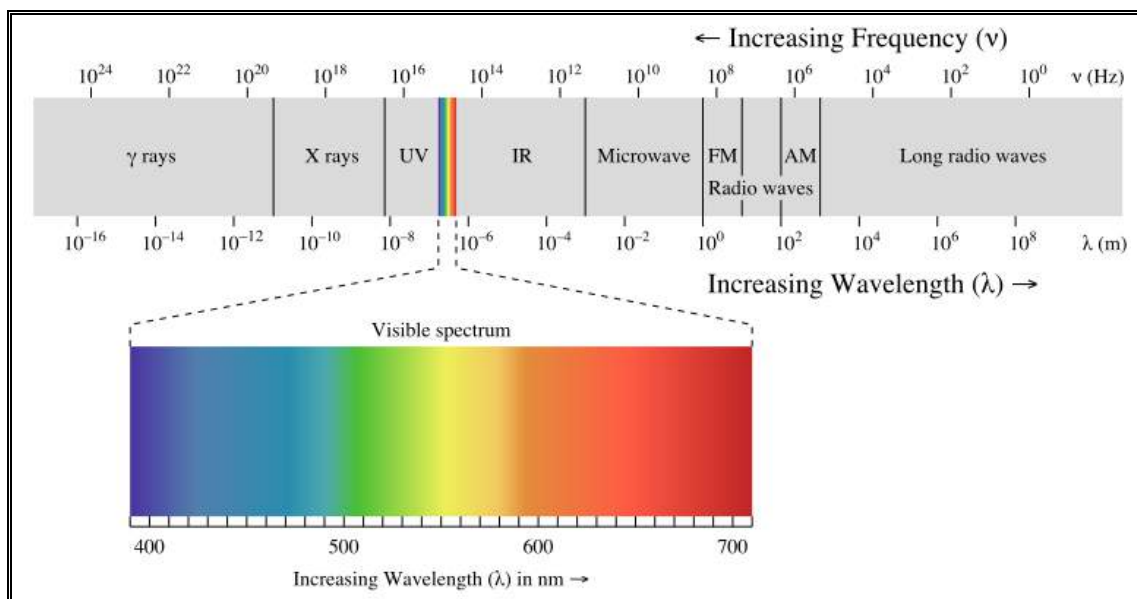


Fig. 1-21.. Schematic representation of electromagnetic spectrum.

1-12.1. RF Bands

The RF spectrum includes Extremely low frequency (ELF), Very low frequency (VLF), Voice frequency (VF), Medium frequency (MF), High frequency (HF), Low frequency (ELF), ultra-high frequency (UHF), super high frequency (SHF), and extremely high frequency (EHF) signals. Above 300 GHz, the absorption of electromagnetic radiation by Earth's atmosphere is so great that it is effectively opaque, until the atmosphere becomes transparent again in the so-called infrared and optical window frequency ranges. The RF spectrum is usually defined as electromagnetic energy ranging from approximately 1 GHz to 1000 GHz in frequency, but older usage includes lower frequencies. Most common frequency bands are shown in the following table.

Table 1-4. Radio spectrum

Frequency Band	Range	Wavelength	Application
Extremely Low Frequency (ELF)	3-30 Hz	10,000-100,000 km	Underwater Communication
Super Low Frequency (SLF)	30-300 Hz	1,000-10,000 km	AC Power (not a transmitted wave)
Ultra Low Frequency (ULF)	300-3000 Hz	100-1,000 km	
Very Low Frequency (VLF)	3-30 kHz	10-100 km	Navigational Beacons
Low Frequency (LF)	30-300 kHz	1-10 km	AM Radio
Medium Frequency (MF)	300-3000 kHz	100-1,000 m	Aviation & AM Radio
High Frequency (HF)	3-30 MHz	10-100 m	Shortwave Radio
Very High Frequency (VHF)	30-300 MHz	1-10 m	FM Radio
Ultra High Frequency (UHF)	300-3000 MHz	10-100 cm	TV, Mobile Phones, GPS
Super High Frequency (SHF)	3-30 GHz	1-10 cm	Satellite Links, Wireless Comm.
Extremely High Frequency (EHF)	30-300 GHz	1-10 mm	Astronomy, Remote Sensing
Visible Light	400-790 THz	380-750 nm	Human Eye

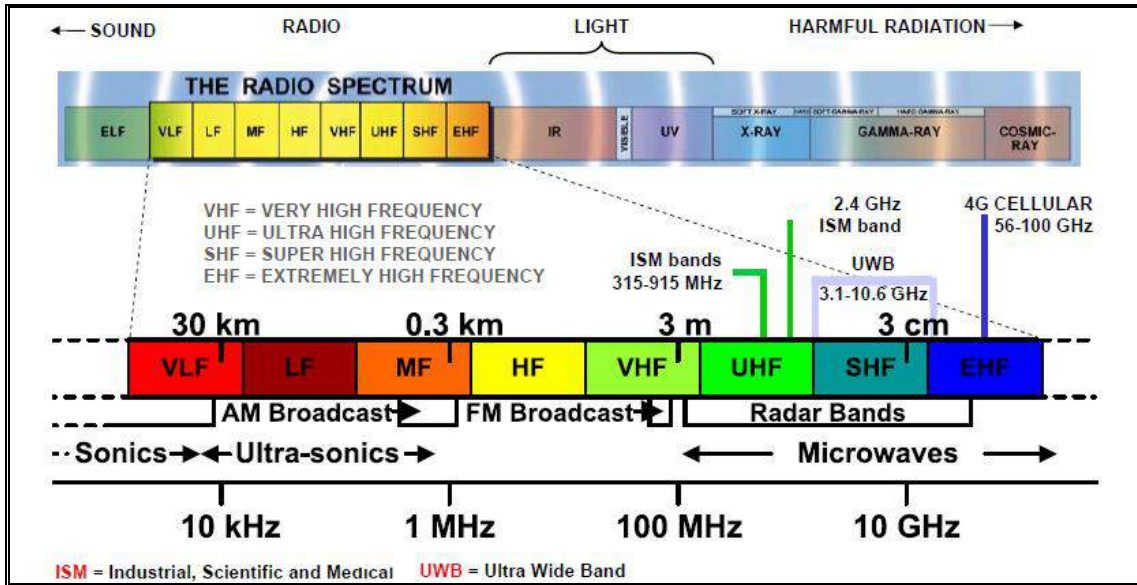


Fig. 1-22. Detailed spectrum of RF waves

1-12.2. ISM Bands

The industrial, scientific and medical (ISM) radio bands are radio bands reserved for the use of industrial, scientific and medical purposes other than telecommunications. Examples of applications in these bands include RF heating, microwave ovens, garage openers, medical diathermy machines and amateur radio. The powerful emissions of these devices can create electromagnetic interference and disrupt radio communication if they are using the same frequency, so these devices were limited to certain bands of frequencies. In general, communications equipment operating in these bands must tolerate any interference generated by ISM equipment. The ISM bands are defined by the ITU-R. In 1997, other bands in the 5GHz range, known as the Unlicensed National Information Infrastructure (U-NII), were added.

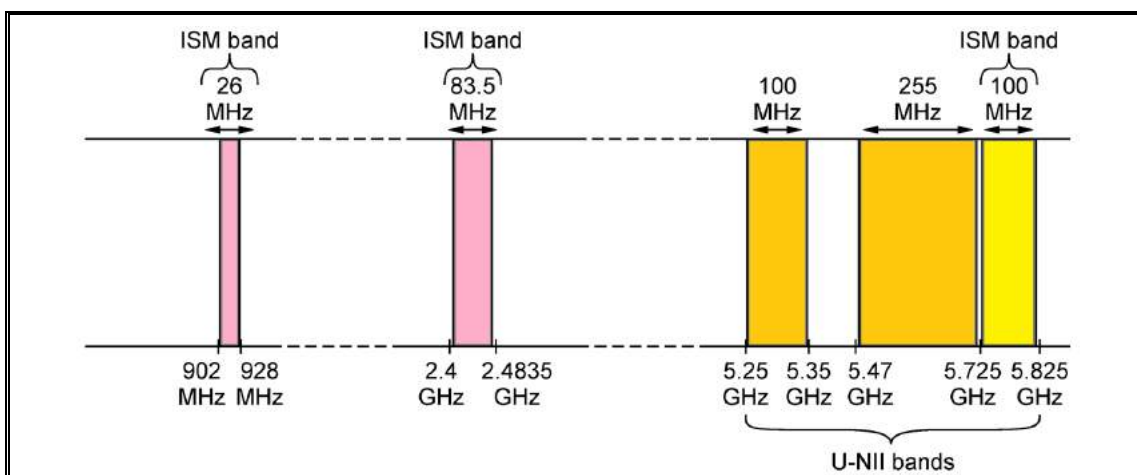


Fig. 1-23. ISM Bands

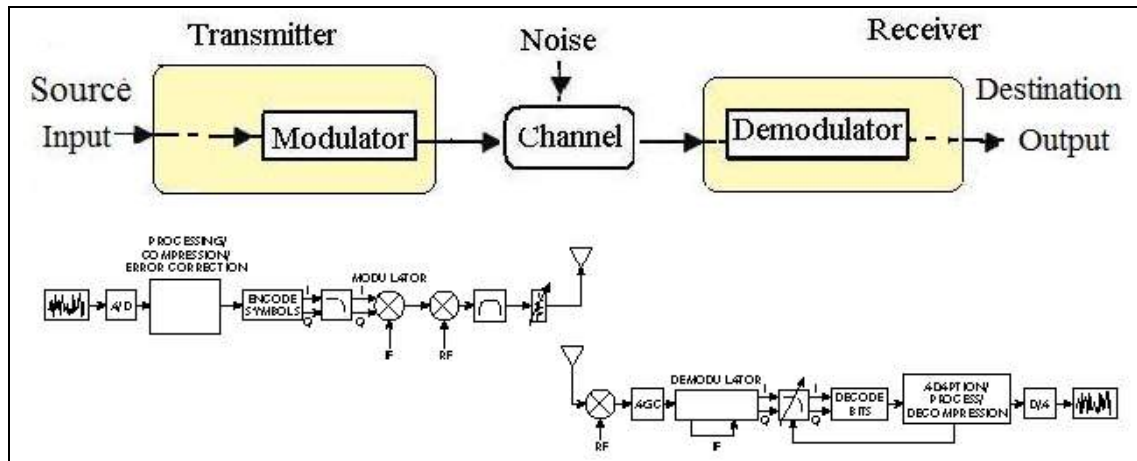
1-13. Summary

The basis of Information Technology and Communications (IT&C) is the processing and transmission of information. A communication system processes communications signals between a source (**transmitter**) and a destination (**receiver**), via a transmission medium (**channel**). The ability to process a communications signal means that errors caused by random processes (noise) can be **detected** and **corrected**. The main feature of a communication system is its ability to convey information, between receiver and transmitter, in the presence of such impairments like **noise** and **losses**. Radio and TV as well as computer communication represent distinct traditions within the field of telecommunications. The following table shows the history of telecommunication development.

YEAR	TYPE OF COMMUNICATION
1831	Samuel Morse invents the first repeater / telegraph
1837	Charles Wheatstone patents "electric telegraph".
1876	Alexander Graham Bell and Elisa Grey independently invent the telephone (although it may have been invented by Antonio Meucci as early as 1857)
1885	Alexander Graham Bell incorporated AT&T
1885	James Maxwell predicts the existence of radio waves
1887	Heinrich Hertz verifies the existence of radio waves
1895	Guglielmo Marconi invents the first radio
1890's	Radio telegraphy or "wireless".
1920's	Radio broadcasting.
1929	The CRT display tube is invented
1935	Edwin Armstrong invents FM
1948	The transistor is invented
1950's	Television broadcasting.
1957	The first artificial satellite, Sputnik goes into orbit
1960's	Geostationary satellite communications.
1970's	Digital computer communications.
1980's	Optical communications.
1986	The first transAtlantic fiber optic cable goes into service
1990's	Internet and mobile communications.

It is remarkable that the earliest form of electrical communication, namely telegraphy developed by Samuel Morse in 1831, was a digital communication system. The beginnings of what we now regard as modern digital communications stem from the work of **Nyquist** in 1924.

He concluded that for binary data transmission (0 or 1) over a noiseless channel of bandwidth B Hertz, the maximum pulse rate is $2B$ pulses per second. In 1948 **Shannon** derived the fundamental limits of bit rate over a noisy channel.



Modulation is the process of varying a periodic waveform, in order to use that signal to convey (transmit) a message. Normally a **high-frequency** sinusoidal wave is used as a carrier signal. The various modulation techniques in analog systems (such as amplitude modulation, **AM** frequency modulation **FM** or phase modulation **PM**), offer different solutions in terms of cost-effectiveness and quality of received signals. We can also communicate information in digital form by digital modulation of a carrier waveform.

Noise and channel **bandwidth** are the fundamental physical limitations of data transmission. Noise consists of any undesired signal in a communication circuit. These fundamental limitations can be overcome by appropriate **modulation** and **coding** techniques.

Most man made electro-magnetic noise occurs at frequencies below 500 MHz. The most significant of these include: Ignition systems, Fluorescent lights and Electric motors. There are also a wide range of natural noise sources which cannot be so easily avoided, namely:

- Atmospheric noise – lightning < 20 MHz
- Solar noise - sun - 11 year sunspot cycle
- Cosmic noise - 8 MHz to 1.5 GHz
- Thermal or Johnson noise. Due to free electrons striking vibrating ions.
- White noise - white noise has a constant spectral density over a big range of frequencies. Johnson noise is an example of white noise.

The following table depicts Some noise sources and their effects

Source of Noise	What Causes It	How to Prevent It
Line Outages	Storms, Accidents	
White Noise	Movement of electrons	Increase signal strength
Impulse Noise	Lightning or sudden increase of electricity	Shield wires
Cross-Talk	Wires too close together	Isolate or shield wires
Echo	Poor connections	Fix/isolate connections
Intermodulation Noise	Signals from several circuits combine	Move or shield wires
Jitter	Signals change phase	Tune equipment
Harmonic distortion	Circuits change phase	Tune equipment

Signal-to-noise ratio (often abbreviated **SNR** or **S/N**) is a measure used in science and engineering to quantify how much a signal has been corrupted by **noise**. It is defined as the ratio of signal power to the noise power corrupting the signal.

Information theory is a branch of information engineering involving the calculations of **fundamental limits** of data communication and data compression. Important sub-fields of information theory are **source coding**, **channel coding**, algorithmic information theory, and information measurements. The central paradigm of classical information theory is the engineering problem of the transmission of information over a noisy channel. The most fundamental results of this theory are **Shannon's source coding theorem**, and **Shannon's noisy-channel coding theorem**. The **Shannon's noisy-channel coding theorem** states that *reliable* communication is possible over *noisy* channels if the rate of communication is below a certain threshold called the **channel capacity**. The **Shannon equation** allows us to precisely determine the information capacity, C , of any signal channel, as follows:

$$C = B \log_2 (1 + SNR)$$

where B is the channel bandwidth and SNR is the signal-to-noise ratio. The **channel capacity** can be approached by using appropriate encoding and decoding systems. The **spectral efficiency** (η_s) of a digital communication system is measured in bit/s/Hz. It is the net bit-rate (R_b) divided by the bandwidth (B) in hertz of a communication channel or a data link. $\eta_s = R_b / B$

1-14. Problems

- 1-1)** Who was first to transmit voice over radio?
- 1-2)** Who was first to send two-way wireless telegraphy messages across the Atlantic Ocean?
- 1-3)** What are the main differences between analog and digital systems?
- 1-4)** What are the main benefits of modulation techniques?
- 1-5)** Mention some applications of both AM and FM modulation techniques in our daily life communication systems.
- 1-6)** What are the main types of modulation & coding methods?
- 1-7)** Mention some applications of both AM and FM modulation techniques in our daily life communication systems.
- 1-8)** Give signal-to-noise ratio guidelines at a receiving device for the following three media: (1) voice, (2) video-TV, and (3) data.
- 1-9)** What are the main impairments, which we face in data communication?
- 1-10)** What are main types of noise that affect the communication equipment?
- 1-11)** What are the main types of transmission channels?
- 1-12)** Explain the point of Shannon's Law, and give an example of a channel capacity and maximum data rate calculation.
- 1-13)** Given a channel with an intended capacity of 20 Mbps. The bandwidth of the channel is 3MHz. What signal-to-noise ratio is required in order to achieve this capacity?
- 1-14)** Find the maximum channel capacity, of a particular channel which has signal-to-noise ratios of 10, 100, and 1000, for three different signals. Consider the signal bandwidth in the three cases as 1 MHz, 2 MHz and 3 MHz. Comment on your answer.
- 1-15)** Calculate the spectral efficiency of the above indicated channels, if the bit rate is equal to the maximum channel capacity.
- 1-16)** Why we need to encoding in digital communication systems?
- 1-17)** In a modem pool there are $n = 4$ modems and the offered traffic intensity is $A = 2$ Erlang.
- What is the probability that a call attempt fails?
 - What is the blocking probability, and grade of service (GoS)?
- 1-18)** Consider a mobile phone network, with a cell site that has 5 FDMA radio channels. The average call rate (I) is 60 calls per hour. The call holding time is distributed exponentially, with an average (T) of 90 sec
- Calculate the offered traffic intensity (A) in Erlangs.
 - What is the blocking probability and Grade of Service in this network,
 - If the bandwidth $W = 1\text{MHz}$ and the cell area $A_c = 1\text{km}^2$, what is the efficiency of such a cellular network

1-15. References

- [1] R. V. L. **Hartley**, "Transmission of Information", *Bell System Technical Journal*, July **1928**.
- [2] C.E. **Shannon**, "A mathematical theory of communication," *Bell. Syst. Tech. J.* **27**, 379–423, **1948**.
- [3] C. E. **Shannon**, "Communication in the presence of noise", *Proc. Institute of Radio Engineers*, Vol. 37, No. 1, pp.10–21, January **1949**.
- [4] J. G. **Proakis**, "Probabilities of error for adaptive reception of M -phase signals," *IEEE Trans. Commun. Tech.* **16**, 71–81, **1968**.
- [5] Simon **Haykin**, *Digital Communications*. Toronto, Canada: John Wiley & Sons, **1988**
- [6] D. **Bertsekas** and R. **Gallager**, "Data Networks", 2nd Ed., Prentice-Hall, **1992**.
- [7] H. P. **Hsu**, *Analog and Digital Communications*, McGraw-Hill, **1993**.
- [8] A. V. **Oppenheim**, et al., *Signals and Systems*, 2nd Ed., Prentice-Hall, **1996**.
- [9] T. S. **Rappaport**, *Wireless communications: principles and practice*. Upper Saddle River, N.J.: Prentice Hall PTR, **1996**.
- [10] M. **Schwartz**, W. R., **Bennett**, and S. **Stein**, *Communication systems and techniques*. New York: IEEE Press, **1996**.
- [11] W. Leon **Couch**, *Digital and Analog Communications*. Upper Saddle River, NJ: Prentice-Hall, **1997**.
- [12] G. L. **Stüber**, "Modulation Methods", in *The Communications Handbook* (J D. Gibson ed.), CRC Press, January **1997**.
- [13] Simon **Haykin**, *Communication Systems*, 4th Ed., Wiley, **2001**.
- [14] M. **Thomas** and A. Thomas Joy, *Elements of information theory*, 1st Edition. New York: Wiley-Interscience, 1991. 2nd Edition, New York: Wiley-Interscience, **2006**.
- [12] K. **Giridhar**, "Wireless Communications – Principles & Practice" 2nd Ed., **2008**.
- [13] Christopher D. **Manning**, Prabhakar **Raghavan**, Hinrich **Schütze**, *An Introduction to Information Retrieval*, Cambridge University Press, **2008**.

Chapter
2

Analog Modulation & Radio Communication Systems

Contents

2-1. Radio Communication Systems	
2-1.1. Superheterodyne Receiver	
2-1.2. Image-Reject Receivers (IRR) Receiver	
2-1.3. Direct Conversion (Zero IF) Receivers	
2-2. Analog Modulation Techniques	
2-3. Amplitude Modulation (AM)	
2-4. Frequency Modulation (FM)	
2-5. Noise in Radio Systems	
2-5.1. Friis' Formula for Noise Figure	
2-5.2. Friis' Transmission Formula	
2-5.3. Minimum Detectable Signal (MDS) of a Radio Receiver	
2-5.4. Sensitivity of a Radio Receiver	
2-6. Testing of a Radio Receiver	
2-7. Summary	
2-8. Problems	
2-9. Bibliography	

46

Analog Modulation & Radio Communication Systems

2-1. Radio Communication Systems

Today, vast amounts of information are communicated using radio communication systems. Both analogue and digital radio communication systems are used. However, any communication system is composed of a **transmitter**, a **receiver** and a **channel** to convey information between them. Communication systems can be considered to be wired (e.g., telephone) or wireless (e.g., cellphone). A wireless system uses radio frequencies to connect users and is capable of operating over a much larger geographical area than a wired system. Figure 2-1 illustrates the block diagram of a wireless communication system. The **source** may be either an analog signal, such as an audio or video signal, or a digital signal, such as the output of a teletype machine. The source data is converted into a high frequency signal. The process of conversion is called **modulation**. At the receiving end of a communication system, the **demodulator** processes the transmitted signal, which is corrupted across the channel, and reduces the waveforms to the original sent information.

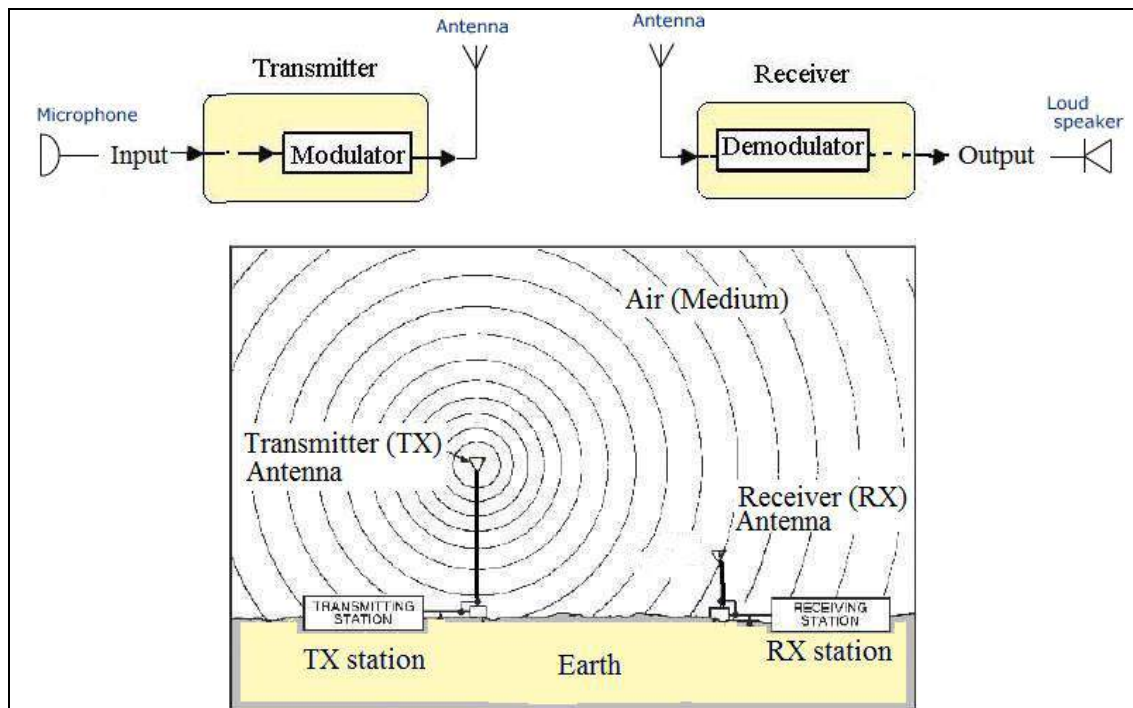


Fig. 2-1. Block diagram of a wireless communication system

The major advantages of radio frequency (**RF**) communication systems over wired communication systems are their ability to provide communications over large distances, through some obstacles, to an unlimited number of users. The range of the system is defined as the distance between the transmitter and the receiver at which the received signal power by the receiver is less than that of the background noise. Radio frequencies occupy the range from 3kHz to 300GHz, although commercial applications of radio use a small part of this spectrum. Other types of electromagnetic radiation, above the RF range, are infrared, visible light, ultraviolet, X-rays and gamma rays. Since the energy of a photon of radio frequency is too low to remove an electron from an atom, radio waves are classified as non-ionizing radiation.

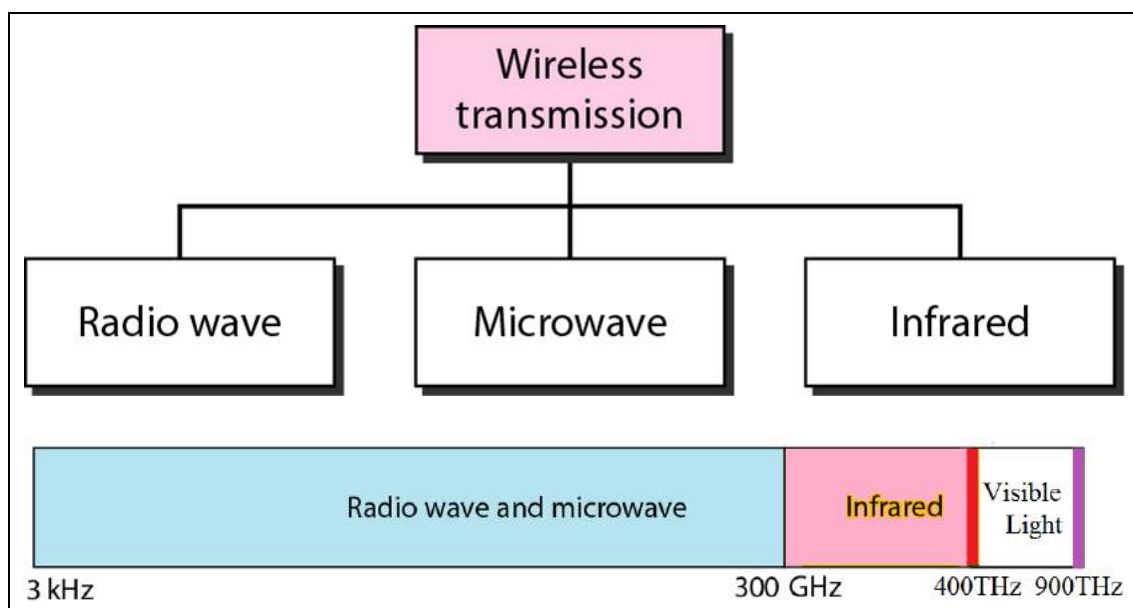


Fig. 2-2. Types of wireless transmission and their frequency range

2-1.1. Superheterodyne Receiver

The superheterodyne or heterodyne receiver is the most widely used reception technique and finds numerous applications from personal communication devices to radio and TV tuners. A typical block diagram of most of the RF blocks that make up a typical superheterodyne receiver is shown in figure 2-3. Many aspects of this receiver are common to all RF circuits. The heterodyne receiver has its RF signal from antenna and its input is filtered through an RF bandpass filter (BPF) to a mixer. Also, a local oscillator (usually VCO) is feeding the mixer that is tuneable and differs from the input RF signal by a fixed amount – known as the Intermediate frequency (IF). Therefore, to tune for a particular RF signal, the local oscillator (LO) is tuned accordingly. As the output of the mixer is always fixed IF frequency then a bandpass IF filter should be used.

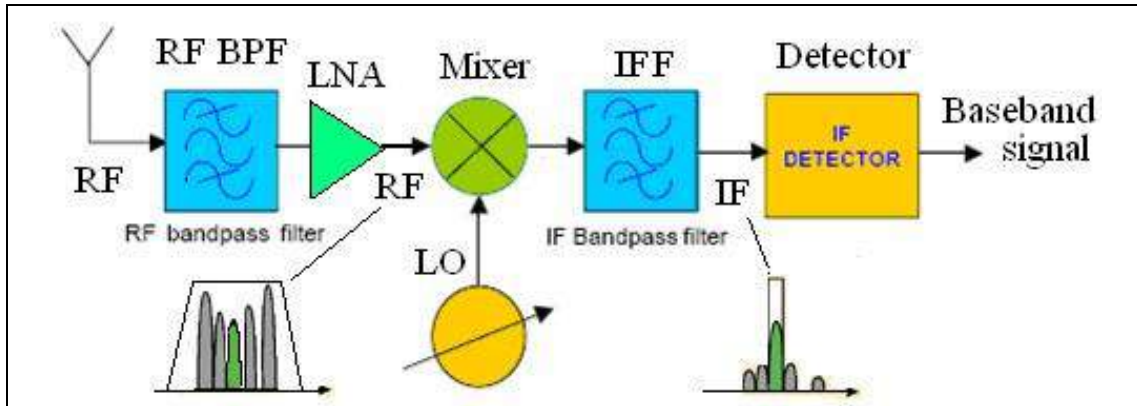


Fig. 2-3. Typical superheterodyne receiver. LNA = Low-noise amplifier, LO = Local oscillator, IF = Intermediate Frequency, IFF =IF bandpass filter

The IF signal is generated by mixing the RF (ω_{RF}) with a single LO (ω_{LO}) carrier as shown by the equation be-low:

$$V_{IF} = V_{LO} \cos \omega_{LO} \cdot t * V_{RF} \cos \omega_{RF} \cdot t \tag{2-1}$$

This multiplication will produce two products the sum and the difference in frequencies (the IF frequency) we want:

$$\frac{V_{LO} \cdot V_{RF}}{2} (\cos[(\omega_{LO} - \omega_{RF}) - \phi] + \cos[(\omega_{LO} + \omega_{RF}) + \phi]) \tag{2-2}$$

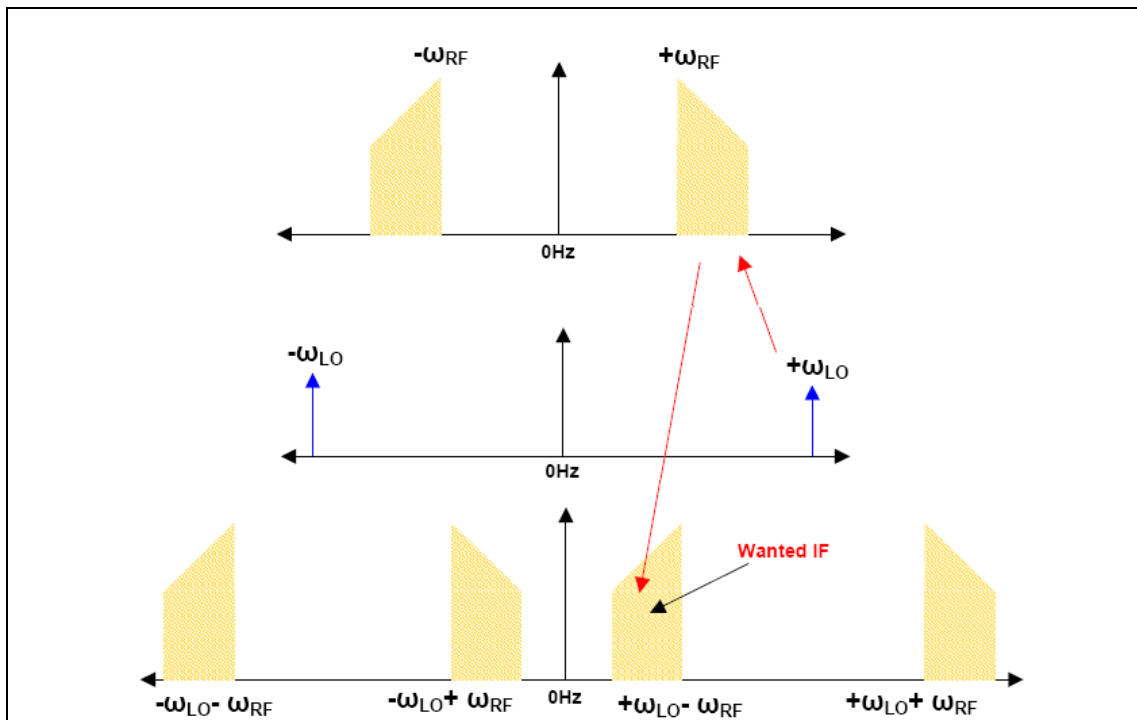


Fig. 2-4. Typical signals of a superheterodyne receiver

The IF filters will only select the wanted difference frequency ($IF = LO - RF$) and reject the much higher sum frequency ($RF + LO$). The wanted IF will be selected and the bandpass filter will reject all other frequencies.

Assuming there is no filtering at the front end of the receiver then not only will the LO mix with the wanted RF to form an IF but also will mix with a RF frequency 2 IF's above the wanted as shown in figure 2-5.

The use of a superheterodyne technique entails several trade-offs. **Image rejection** is a prevailing concern in this architecture. In practice, an RF bandpass filter, usually a surface acoustic wave (SAW) device is utilized to perform band selection ahead of the low noise amplifier (LNA), while a second filter follows the LNA to perform image rejection. If these filters are identical they share the burden of the function. But some amount of image rejection must follow the LNA, for without it, the LNA's noise figure will effectively double due to the mixing of amplified image noise into the IF channel. Instead of the RF SAW filter, other passive filtering technologies such as dielectric or ceramic resonators can also be featured. The higher the IF, the more relaxed the requirements on the cut-off frequency of the image reject filter.

Alternatively, the image can be removed without the need of any post-LNA image-reject filtering. This is the principle of image-reject receivers (IRR). There are two types of IRRs, namely; the **Hartley** receiver and the **Weaver** receiver.

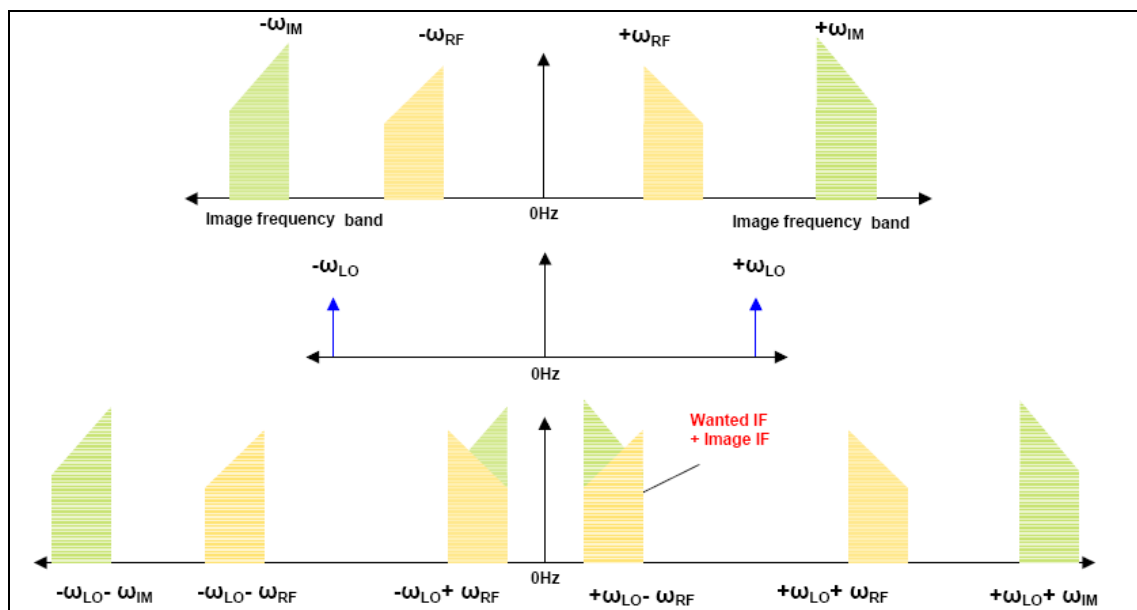


Figure 2-5. Image frequency of a superhet receiver, with no RF bandpass filter

2-1.2. Image-Reject Receivers (IRR)

The Hartley architecture was introduced in 1928. It makes use of two mixers with their local oscillators in a quadrature phase relationship; this separates the IF signal into in-phase (I) and quadrature (Q) components. It then shifts the Q component by 90° before recombining the two paths, where the desired signal is reinforced, while the image is cancelled out.

The dual of the Hartley architecture, known as the Weaver image-reject receiver, achieves the relative phase shift of one path by 90° by the use of a second LO to another IF or to baseband. The same result is achieved. The reliability of these receivers depends on the accuracy of I/Q paths.

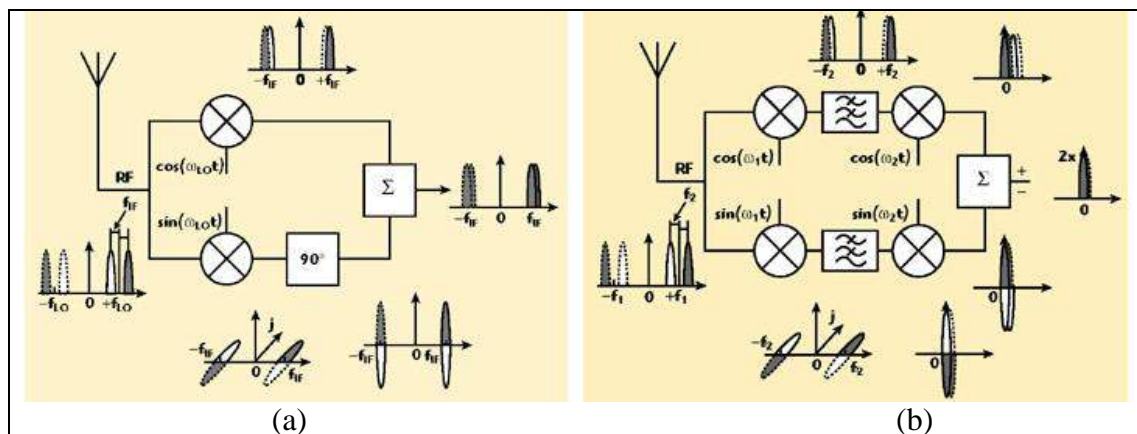


Figure 2-6. Image rejection receivers (IRR). (a) Hartley receiver, (b) Weaver receiver.

2-1.3. Direct Conversion (Zero IF) Receivers

The architecture of direct conversion receivers is shown in figure 2-7. This is also referred to as homodyne, or zero-IF (ZIF), is the most natural solution to receiving information transmitted by a carrier. However, it has only been over the past decade or so that this type of reception has found applications other than pagers. Direct conversion reception has several qualities which make it very suitable for wideband as well as multi-band, multi-standard operation, but there are severe inherent obstacles that have for a long time kept it in the shadow of the superheterodyne technique. First, the problem of the image has been eliminated, since the IF is zero and the image to the desired channel (for all but single-sideband signals) is the channel itself. Then, only one local oscillator is required, which means only one phase noise contribution. The need for the bulky off-chip filters is consequently removed. Filtering now only occurs at low frequencies (baseband) with some amplification, which means less current consumption than at higher frequencies, fewer components and lower cost. Practically, however, strong out-of-band interference or blocking signals may need to be removed prior to down-conversion in

order to avoid desensitizing the receiver by saturating subsequent stages, as well as producing harmonics and intermodulation terms which will then appear in the baseband. Such a filter may be placed after the LNA for example. ZIF receivers, however, have two major problems, namely; **DC offset** and **nonlinearities**.

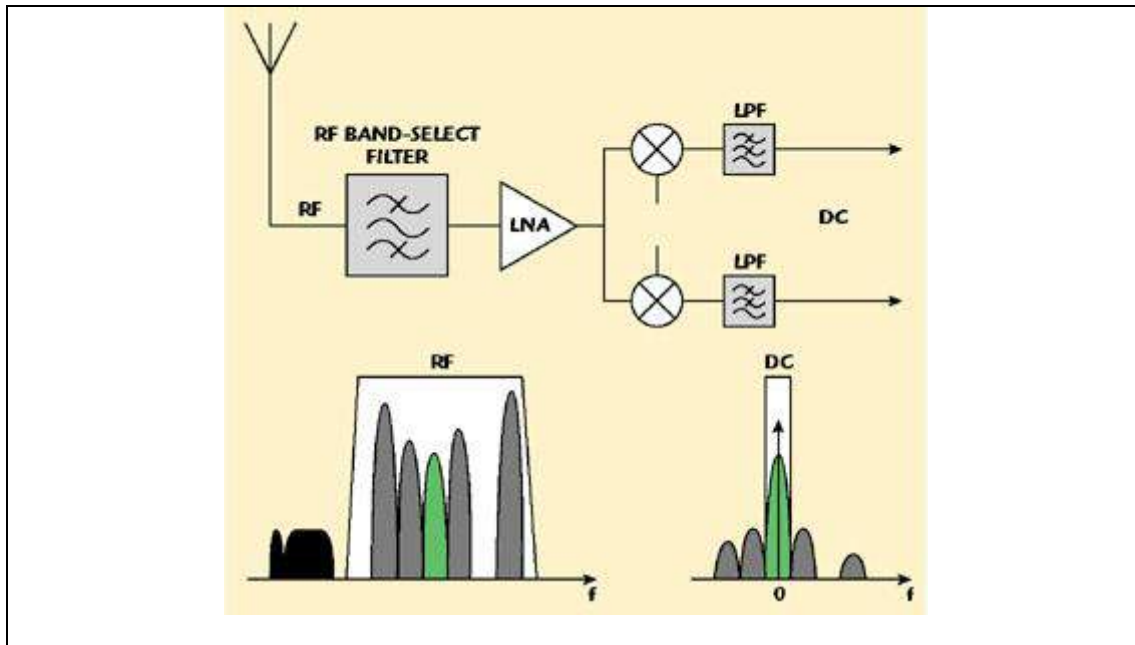


Figure 2-7. Zero-IF receiver (ZIF), for wideband applications

2-2. Analog Modulation Techniques

One of the fundamental aspects of any radio communications system is modulation, or the way in which the information is superimposed on the radio carrier. Therefore, we define **Modulation** as the process of varying a periodic waveform, in order to use that signal to convey (transmit) a signal or a message. Normally a **high-frequency** sinusoidal wave is used as a carrier signal. In fact, it is easier to transmit a high frequency carrier, via antennas with reasonable size (usually about $\lambda/2$), and moderate bandwidth. The three key parameters of a carrier are its amplitude, its phase and its frequency, all of which can be modified in accordance with the low frequency information signal to be transmitted.

The various modulation techniques in analog systems (such as amplitude modulation, **AM** frequency modulation **FM** or phase modulation **PM**), offer different solutions in terms of cost-effectiveness and quality of received signal. We can also communicate digital information by digital modulation of a carrier waveform. The digital modulation process involves some form of AM and/or FM or PM.

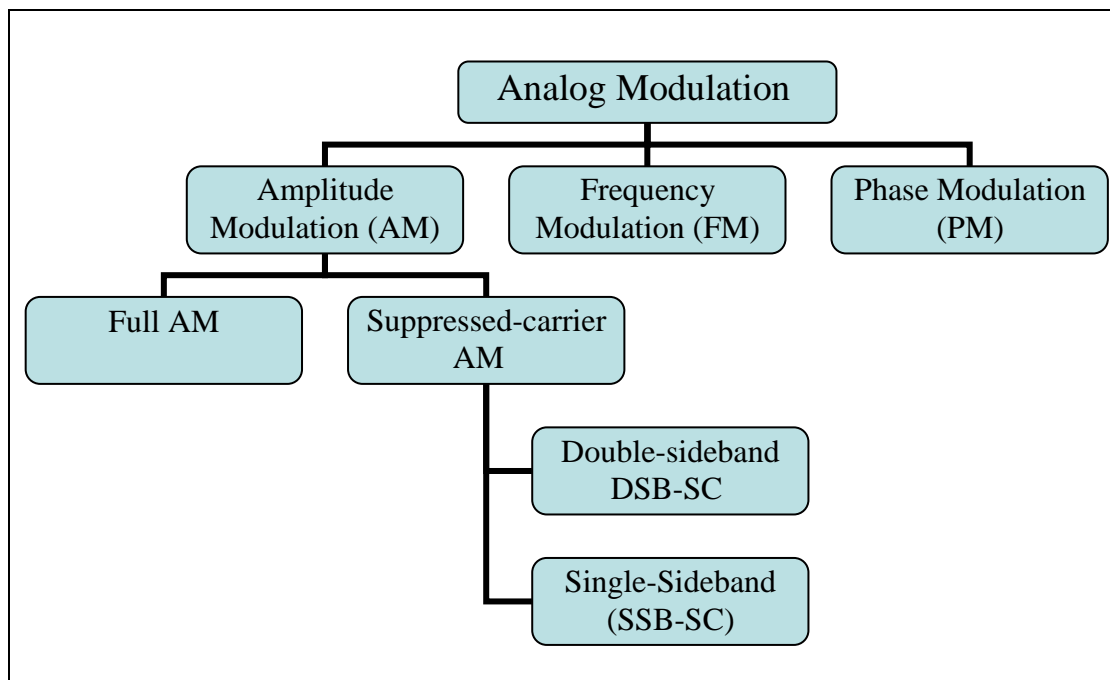


Fig. 2-8. Analog modulation techniques

The performance of any modulation scheme is measured by

- Power efficiency.
- Bandwidth efficiency.
- Power spectral density.
- System complexity.

Power efficiency describes the ability of a modulation technique to preserve the signal quality at low power levels. In digital modulation systems, in order to increase the noise immunity and decrease the bit-error rate (BER), it is necessary to increase the signal power, so there is a trade-off between the signal power and the bit error probability.

Bandwidth efficiency describes the ability of a modulation scheme to accommodate data within a limited bandwidth. In digital systems, as the data rate increases, pulse width of the digital symbols decreases and hence the bandwidth increases. We have already defined the spectral efficiency in chapter 1 as follows: $\eta_B = R_b / B = C / B = \log_2(1 + SNR)$, where C is the channel capacity, R_b is the maximum bit rate and SNR is the signal to noise ratio. The following figure shows a comparison between analog and digital modulation methods

Table 2-1. Advantages and disadvantages of analog modulation techniques

Analog	Digital
Less bandwidth(Advantage)	Large bandwidth(Disadvantage)
More accurate (Advantage)	Less accurate due to the Quantization error that can not be avoided or corrected. (Disadvantage)
Low noise immunity (Disadvantage).	High noise immunity as the amplitude of the digital has two levels only and channel coding (error correcting codes) can be used. (Advantage)
Low level of security. (Disadvantage)	High level of security as you can use Encryption (Ciphering) and Authentication. (Advantage)
No signal conditioning and processing are used (Disadvantage)	Support complex signal conditioning and processing techniques such as source coding, encryption, and equalization(Advantage)
Low QOS. (Disadvantage)	High QOS. (Advantage)
You can use FDM only(Disadvantage)	You can use FDM, TDM, CDM, OFDM multiplexing techniques. (Advantage)
In mobile communications, analog supports voice service only. (Disadvantage)	In mobile communications, digital supports voice, SMS, data (you can access the internet), images and video call. (Advantage)
More difficult to design than Digital. (Disadvantage)	Easily designed using software (Advantage).

2-3. Amplitude Modulation

An **AM signal** encodes the information onto the carrier wave by varying its amplitude in direct sympathy with the analogue signal to be sent. Historically, amplitude modulation has been in use since the very earliest days of radio technology. The first recorded instance of its use was in 1901 when a signal was transmitted by a Canadian engineer named Reginald **Fessenden**. To achieve this, he used a continuous spark transmission and placed a carbon microphone in the antenna lead. The sound waves impacting on the microphone varied its resistance and in turn this varied the intensity of the transmission. Although very crude, signals were audible over a distance of a few hundred metres.

2-3.1. AM Signals

The amplitude modulated signal, can be written as follows:

$$s(t) = A_m(t) \cdot \cos(2\pi f_c t) \quad (2-2)$$

where the instantaneous amplitude $A_m(t)$ is linearly related to the modulating signal $m(t)$.

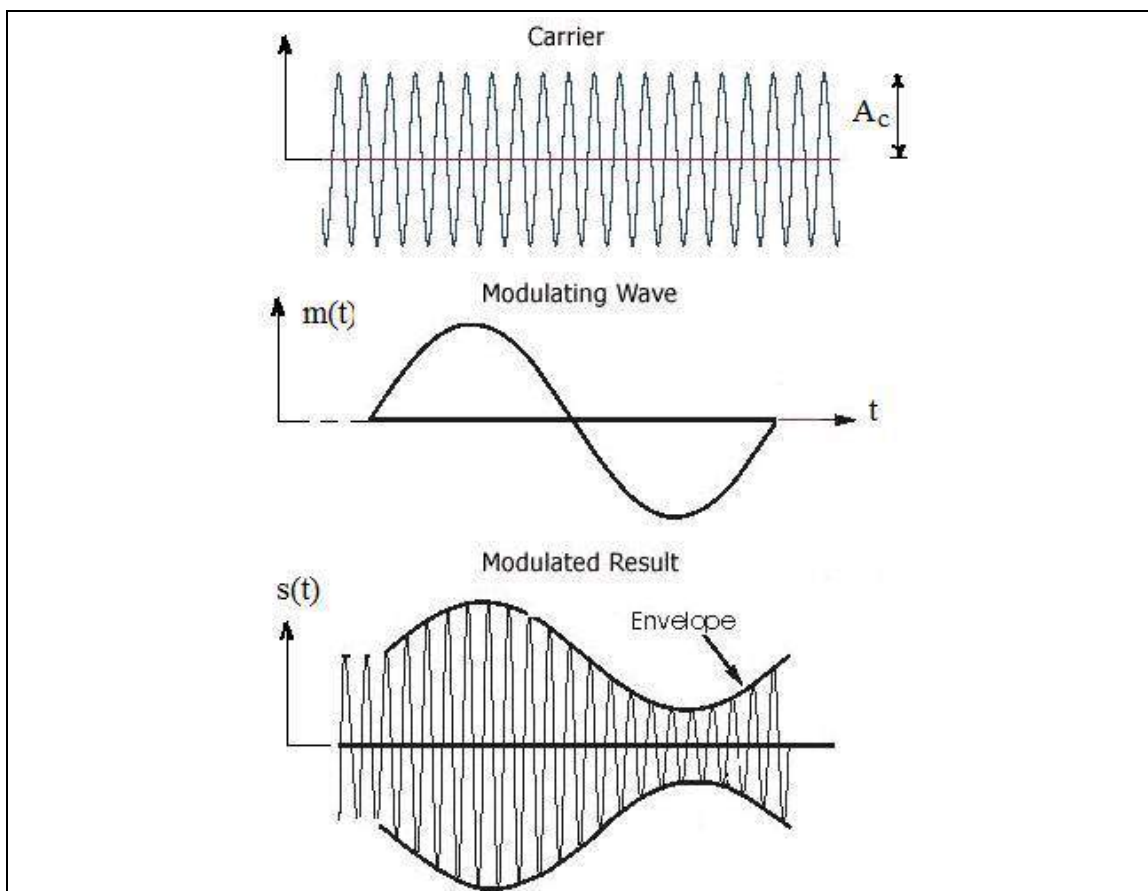


Fig. 2-9. Schematic of the amplitude modulation (AM) waveforms

For normal amplitude-modulated signal we have:

$$s(t) = [A_c + m(t)] \cos(2\pi f_c t) \quad (2-3)$$

where A_c is the carrier amplitude. The *modulation index* m is defined as:

$$m = |m(t)| / A_c \quad (2-4)$$

Figure 2-10 shows AM signals in the time domain and their spectra in the frequency domain. Clearly, the envelope of the modulated signals has the same shape as $m(t)$ when $m < 1$. When $m > 1$, the carrier signal is *over-modulated* and the envelope is distorted.

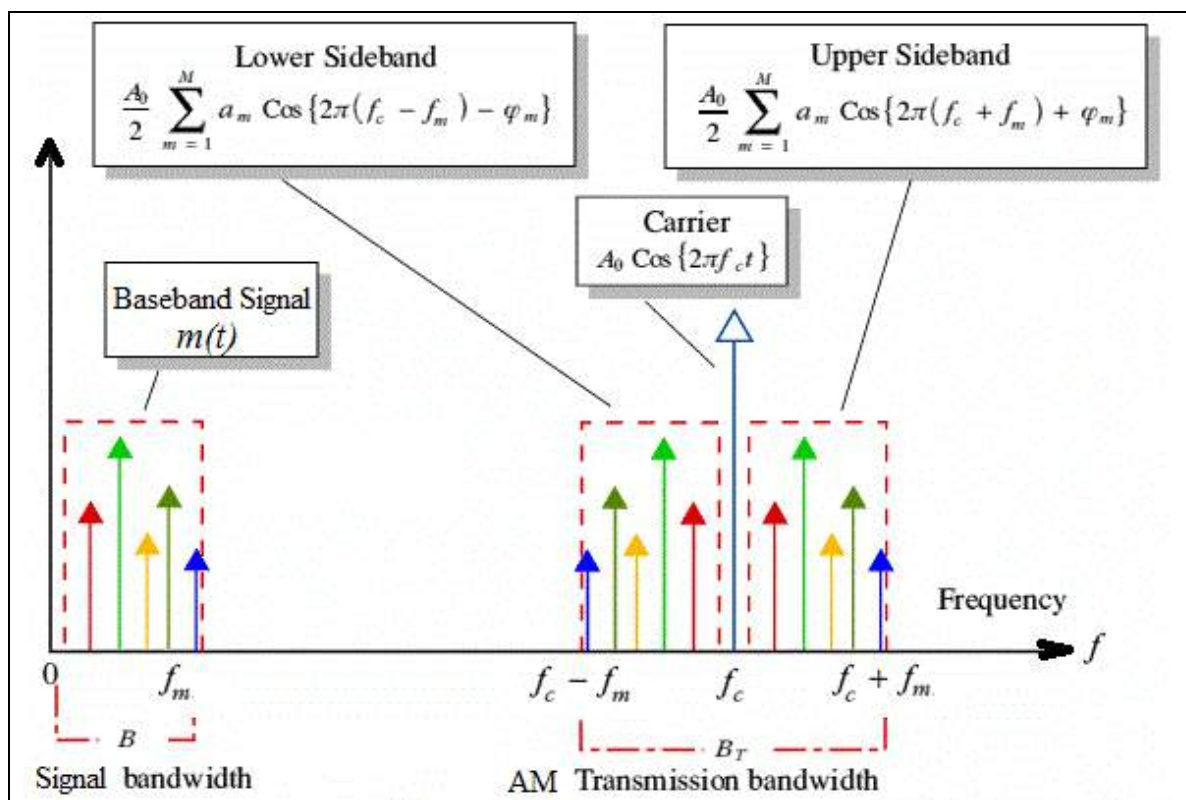


Fig. 2-10. Spectra of an AM signal

The AM transmission results in two additional frequency components (sidebands), at frequencies above and below the carrier frequency. The spacing in frequency between carrier and sidebands is equal to f_m (the maximum modulating frequency). Hence a bandwidth of $2f_m$ is required to transmit the modulated carrier. For a 100% modulated carrier ($m=1$), we have 2/3 of the consumed power in the carrier, the rest in the sidebands

$$P(AM) = P_c + P_s = \frac{1}{2} A_c^2 + \frac{1}{4} m^2 A_c^2 \quad (2-5)$$

2-3.2. AM Modulation Schemes

The Depending on the spectrum of the AM signal, we have one of the following types of amplitude modulation schemes:

- (1) **Full AM**. Here, the modulated signal contains the carrier signal & two side bands of modulating signal.
- (2) **DSB-SC** (Double sideband suppressed carrier). Here, the carrier signal is suppressed.
- (3) **SSB-SC** (Single sideband suppressed carrier) . Here, the carrier signal plus one sideband are suppressed,
- (4) **VSB-AM** (Single sideband suppressed carrier). Here, a major portion of one sideband is suppressed

Obviously, the carrier frequency signal carries no information so that the double sideband, suppressed carrier, amplitude modulation (**DSB-SC-AM**) format has enhanced power efficiency. In single sideband amplitude modulation (**SSB-AM**) format, both the carrier and one complete sideband are suppressed. Thus, SSB-AM is the most efficient in terms of radiated power and frequency space utilization, but involves the greatest complexity in receiver design. A compromise is found in vestigial sideband amplitude modulation (**VSB-AM**) where a major portion of one sideband is suppressed. VSB-AM is used in analog TV broadcasting.

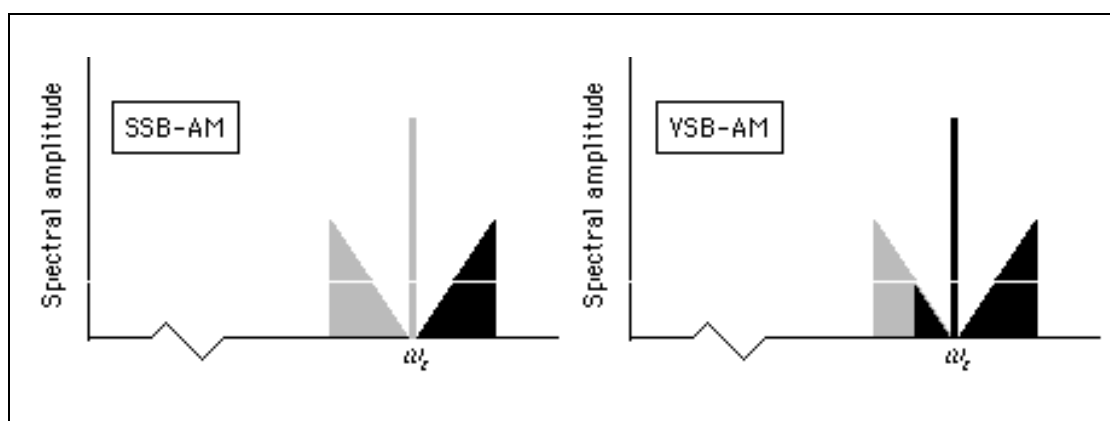


Fig. 2-11. Spectra of an AM signal

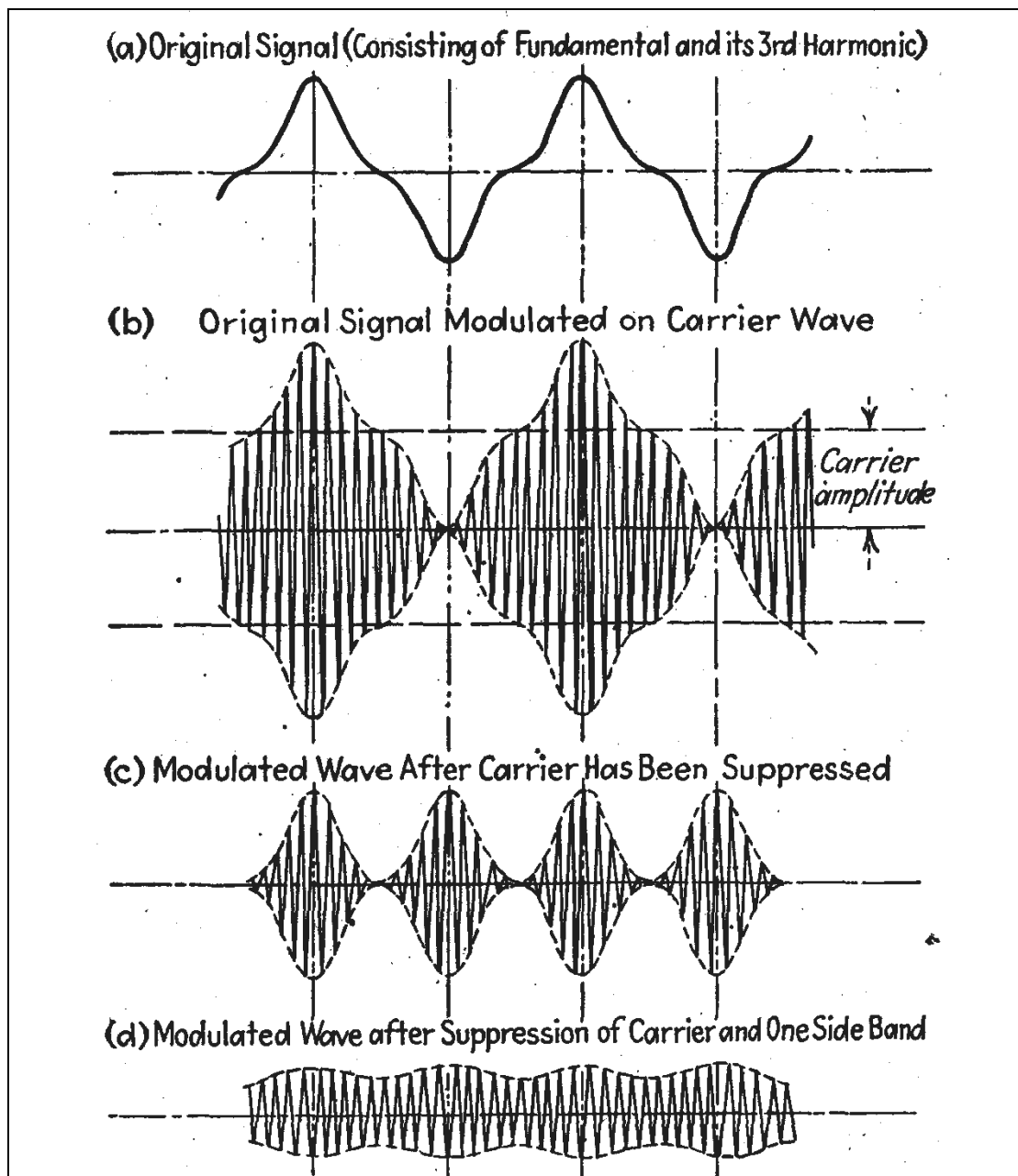


Fig. 2-12. Waveforms AM signals

2-3.3. AM System Architecture

The figure 2-12(a) shows the block diagram of an AM transmitter. The following figure shows the circuit diagram of a balanced modulator, which is used in both DSB-AM and SSB-AM. Figure 2-13, depicts the block diagram of a typical AM **heterodyne** receiver. Heterodyning is a method for transferring a broadcast signal from its carrier to a **fixed** intermediate frequency in the receiver so that receivers work with all channels.

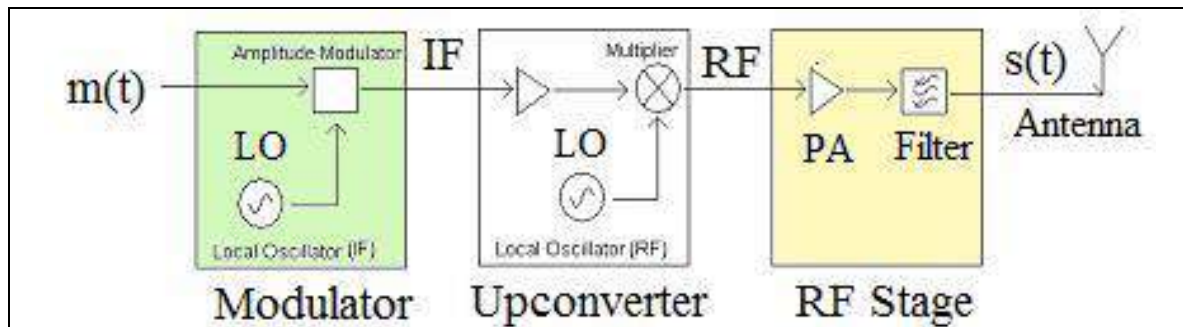


Fig. 2-12(a). Block diagram of an AM transmitter

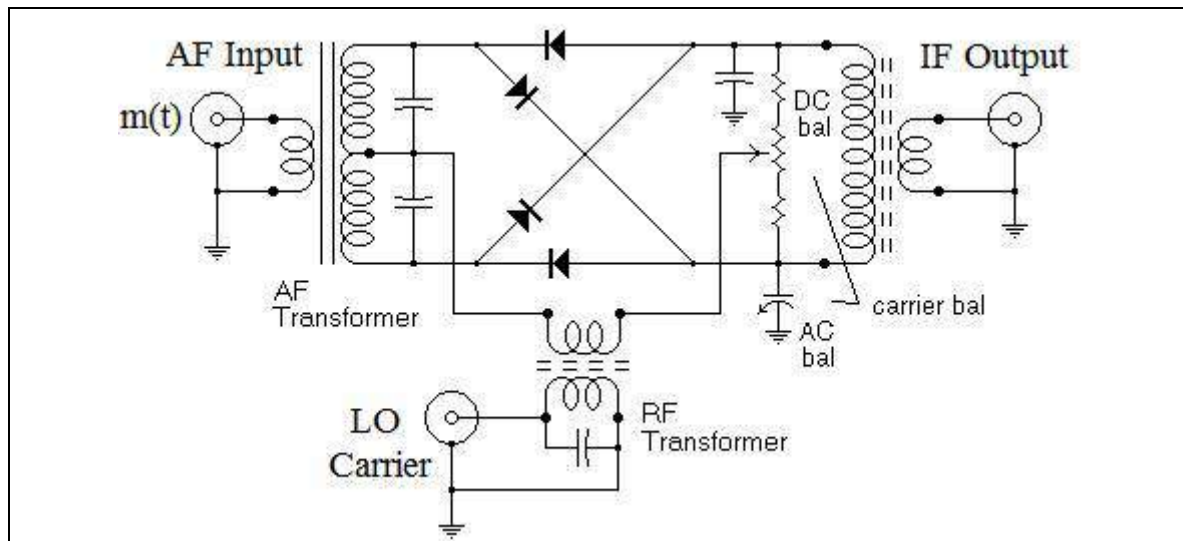


Fig. 2-12(b). circuit diagram of a balanced modulator

The mixing of any channel waves (f_{RF}) with the local oscillator wave (f_o) will produce a fixed intermediate frequency (f_{IF}) or IF. The RF amplifier selects one of the many signals on the antenna (via a tuning circuit), amplifies it, and sends the amplified signal to the mixer. Here, the local oscillator is applied. The two signals are mixed (multiplied) in this stage, producing the sum of the two input frequencies, and the difference between them. The difference frequency is selected as the Intermediate Frequency (**IF**) signal. It carries the information of the received RF signal. The IF amplifier stage amplifies this frequency and sends it to the demodulator (AM detector). Here, the audio frequency (**AF**) is extracted. The carrier is filtered out, leaving only the audio signal, which is amplified by the audio power amplifier and reproduced by the speaker. The Automatic Gain Control (AGC) path gives a control over the IF amplifier gain, according to the received signal strength. There exist so many types of AM demodulators, to extract the original information from an AM signal.

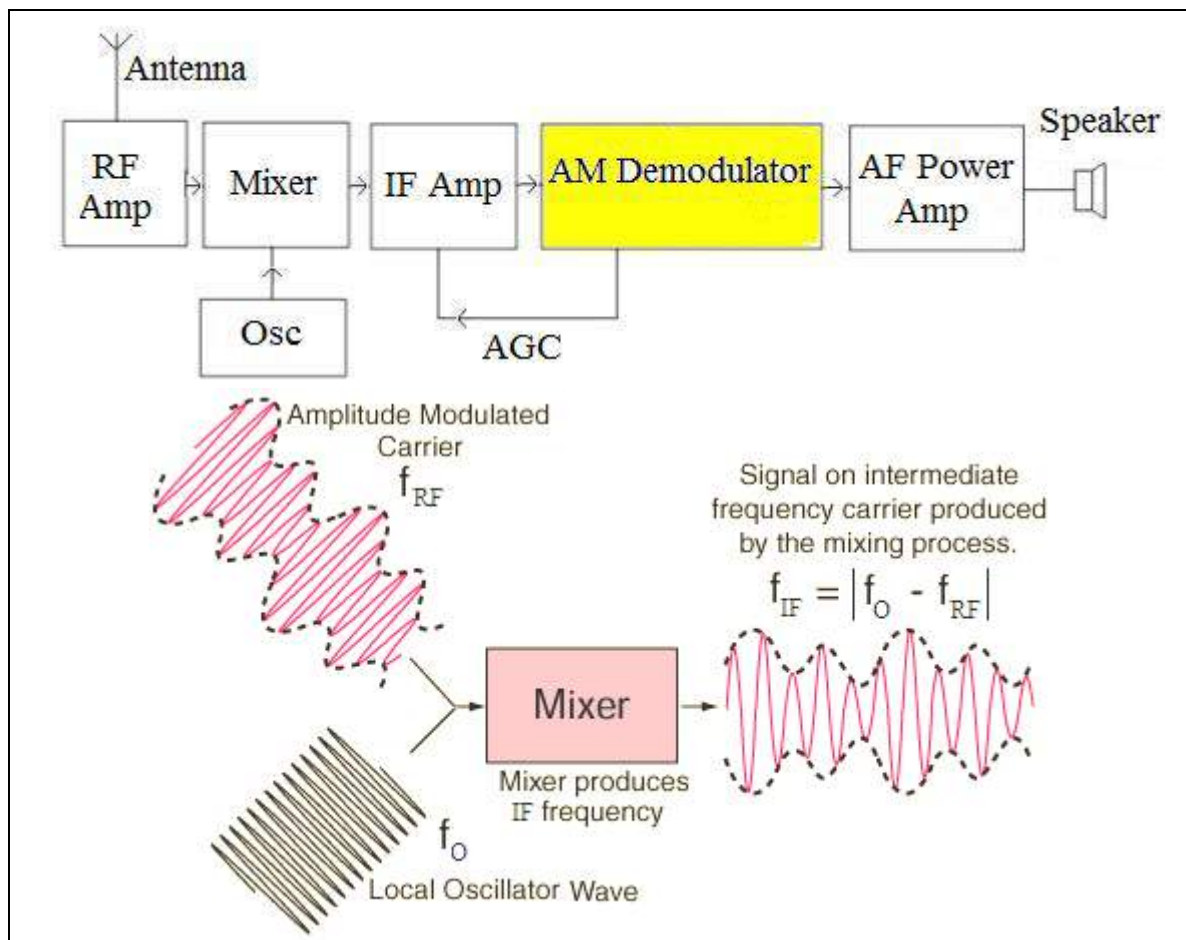


Fig. 2-13. Block diagram of an AM receiver

i. Product detector

The **product detector** is a **coherent** detector that needs a local oscillator, which is a copy of the carrier wave (with the same frequency and phase). Just like a mixer, it multiplies the incoming AM signal by the signal of the local oscillator. Multiplying the AM signal $s(t)$ by an oscillator at the same frequency and phase yields:

$$y(t) = s(t) \cdot \cos(\omega_c t) = [A_c + m(t)] \cos(\omega_c t) \cos(\omega_c t) \quad (2-6a)$$

which can be put in the form:

$$y(t) = [A_c + m(t)] \left[\frac{1}{2} + \frac{1}{2} \cos(2\omega_c t) \right] \quad (2-6b)$$

After filtering the original audio signal $m(t)$ will result. An AM signal can be also rectified without requiring a coherent product demodulator. However, there are some forms of AM (e.g., SC DSB), which require coherent demodulation.

ii. Envelope detector

The **envelope detector** is a very simple method of demodulation. It consists of anything that will pass current in one direction only, that is, a rectifier. This may be in the form of a single diode, or may be more complex. Many natural substances exhibit this rectification behavior, which is why it was the earliest modulation and demodulation technique used in radio. The crystal set exploits the simplicity of the modulation to produce an AM receiver with very few parts.

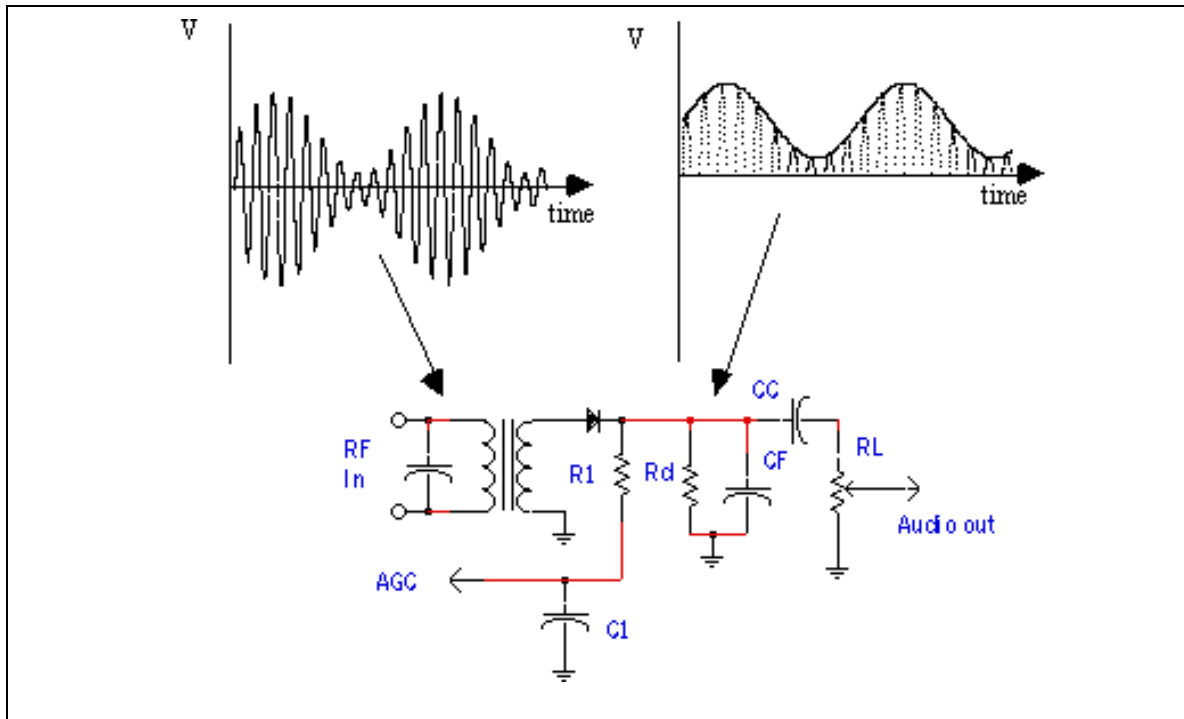


Fig. 2-14 Amplitude demodulation process, by envelop diode detector.

In all communication systems, only a fixed range of frequencies is available for transmission. The AM radio carrier frequencies are in the frequency range 535-1605 kHz. Carrier frequencies of 540-1600kHz are assigned at 10kHz intervals.

2-3.4. AM Over-Modulation and its Effects on Bandwidth

To ensure that an amplitude modulated signal does not create spurious emissions outside the normal bandwidth it is necessary to ensure that the signal does not become over-modulated. This is a conditions that occurs when the modulation index exceeds 100% ($m > 1$). At this point the carrier breaks up and intermodulation distortion occurs leading to large levels of unwanted noise spreading out either side of the carrier and beyond the normal bandwidth. This can cause interference to other users.

If over-modulation occurs, the carrier becomes phase inverted and this leads to sidebands spreading out either side of the carrier.

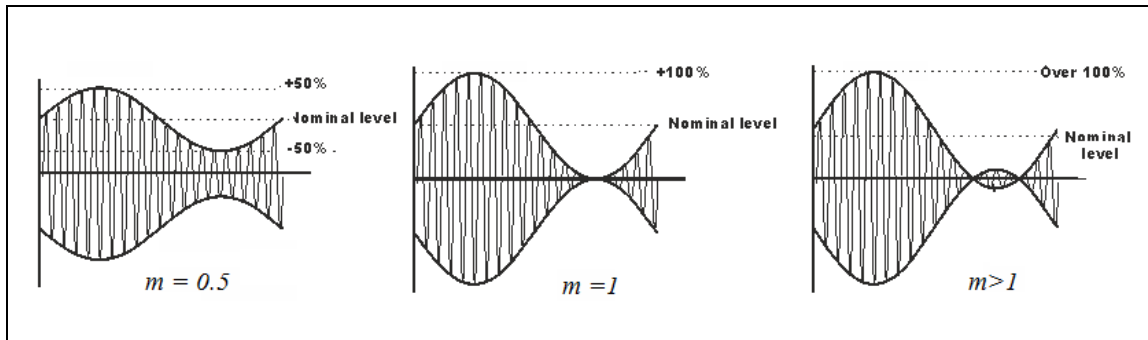


Fig. 2-15 Amplitude demodulation process, by envelop diode detector.

2-3.5. Advantages & Disadvantages of Amplitude Modulation

Like any other system of modulation, amplitude modulation has several advantages and disadvantages. These mean that it is used in particular circumstances where its advantages can be used to good effect. In view of its characteristics advantages and disadvantages, amplitude modulation is being used less frequently. However it is still in widespread use for broadcasting on the long, medium and short wave bands as well as for a number of mobile or portable communications systems including some aircraft communications.

Table 2-2. Advantages and disadvantages of Amplitude Modulation

ADVANTAGES	DISADVANTAGES
It is simple to implement	An AM signal is not efficient in terms of its power usage
It can be demodulated using a circuit consisting of very few components	It is not efficient in terms of its use of bandwidth, requiring a bandwidth equal to twice that of the highest audio frequency
AM receivers are very cheap as no specialized components are needed.	An AM signal is prone to high levels of noise because most noise is amplitude based and obviously AM detectors are sensitive to it.

2-4. Frequency Modulation (FM)

Frequency modulation (FM) is widely used for radio transmissions for a wide variety of applications from radio broadcasting to mobile communications. FM has several advantages over AM, such as its better fidelity, interference reduction and higher noise immunity. FM uses changes in frequency to carry the sound or other information that is required to be sent.

2-4.1. FM Signals

The FM signal, $s(t)$, represents the FM modulated signal as follows:

$$S(t) = A \cos [2\pi f_c t + 2\pi \beta_f \int_0^t m(t') dt'] \quad (2-7)$$

where A is the carrier amplitude, $m(t)$ is the modulating signal and β_f is a modulation parameter. We can also represent the FM signal as follows:

$$S(t) = A \cos [\theta(t)] \quad (2-8a)$$

Assuming a sinusoidal modulating signal, we have:

$$\theta(t) = 2\pi f_c t + (\Delta f / f_m) \cdot \sin(2\pi f_m t) \quad (2-8b)$$

In the above equation, f_m is the modulating signal frequency and Δf is the frequency deviation and it represents the maximum frequency difference between the instantaneous frequency and the carrier frequency. For example, the modulated signal may have a deviation of ± 3 kHz. In this case the carrier is made to move up and down by 3 kHz.

In fact, the ratio $\Delta f / f_m$ is called the modulation index. The FM modulation index, β , is defined as follows:

$$\beta = (\Delta f / f_m) \quad (2-9a)$$

For the traditional FM transmission of voice signals ($f_m=15$ kHz), we take $\Delta f = \pm 75$ kHz. Hence, the message of the FM signal can be represented by the following argument:

$$\theta(t) = 2\pi f_c t + \beta \cdot \sin(2\pi f_m t) \quad (2-9b)$$

Therefore, we can substitute $\theta(t)$ into the original formula to represent the modulated FM signal as follows:

$$s(t) = A \cos [2\pi f_c t + \beta \cdot \sin(2\pi f_m t)] \quad (2-10)$$

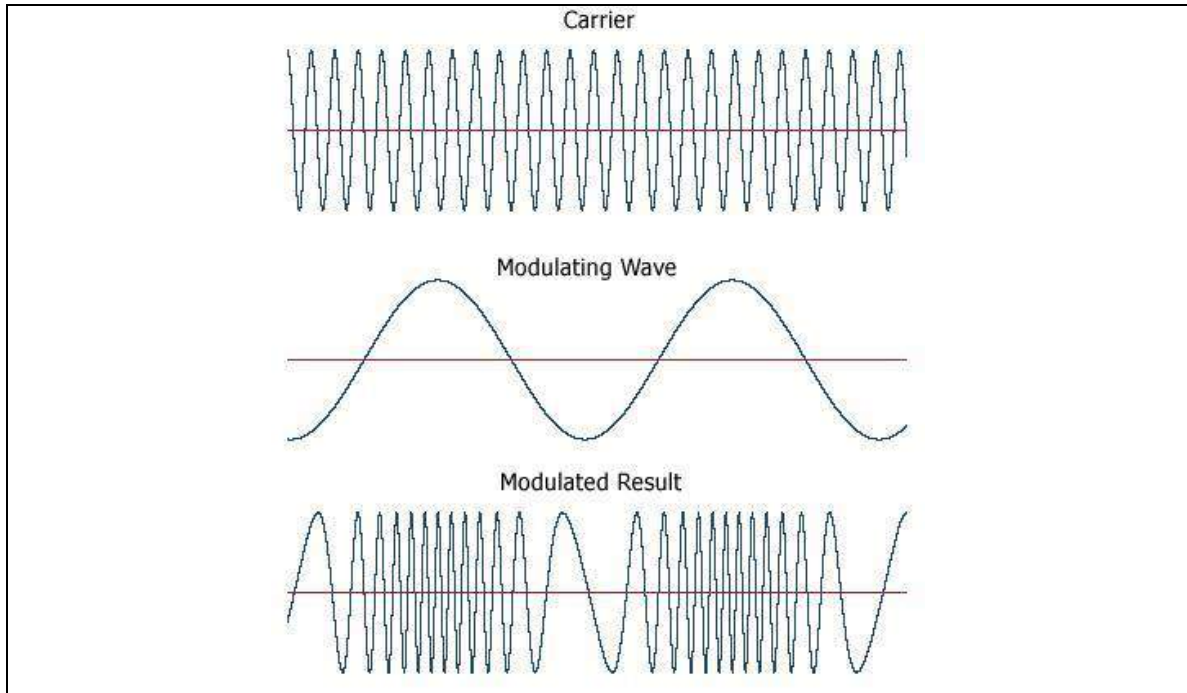


Fig. 2-16. Frequency modulation techniques

2-4.2. FM System Spectrum and Bandwidth

The modulation index β defines the nature of the resultant FM signal and its bandwidth. Mathematically speaking, the bandwidth of the resultant FM signal is theoretically infinite. The total spectrum is an infinite series of discrete spectral components expressed by a complex formula using Bessel functions of the first kind. The total spectrum can be seen to consist of the carrier plus an infinite number of sidebands spreading out on either side of the carrier at integral multiples of the modulating frequency. The relative levels of the sidebands can be obtained by referring to a table of Bessel functions.

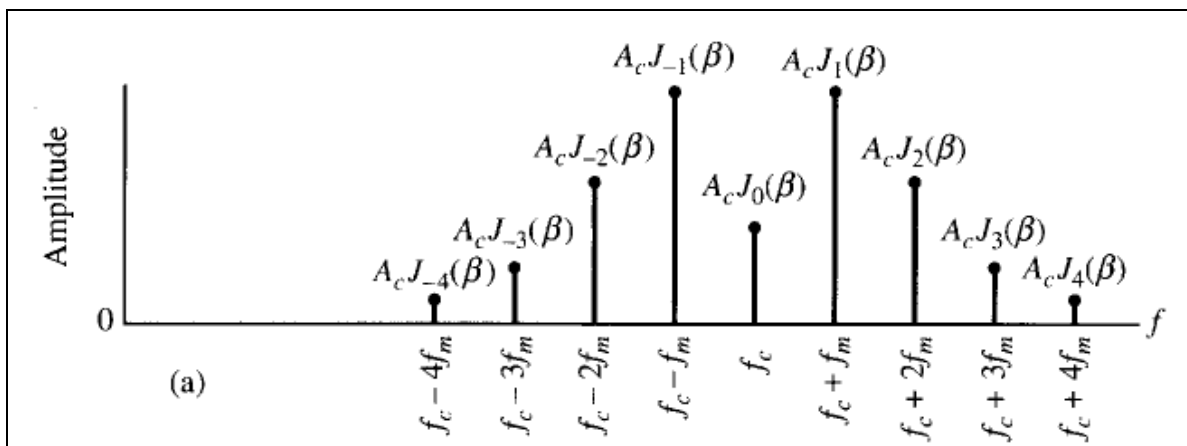


Fig. 2-17(a). FM single-sided spectra

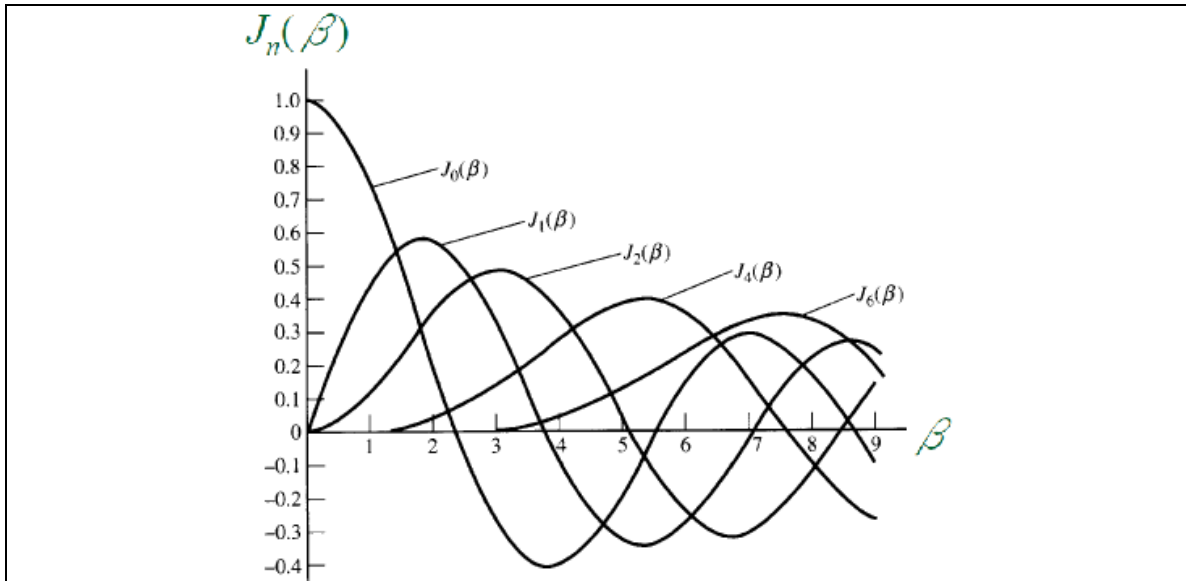


Fig. 2-17(b). Bessel functions $J_n(\beta)$.

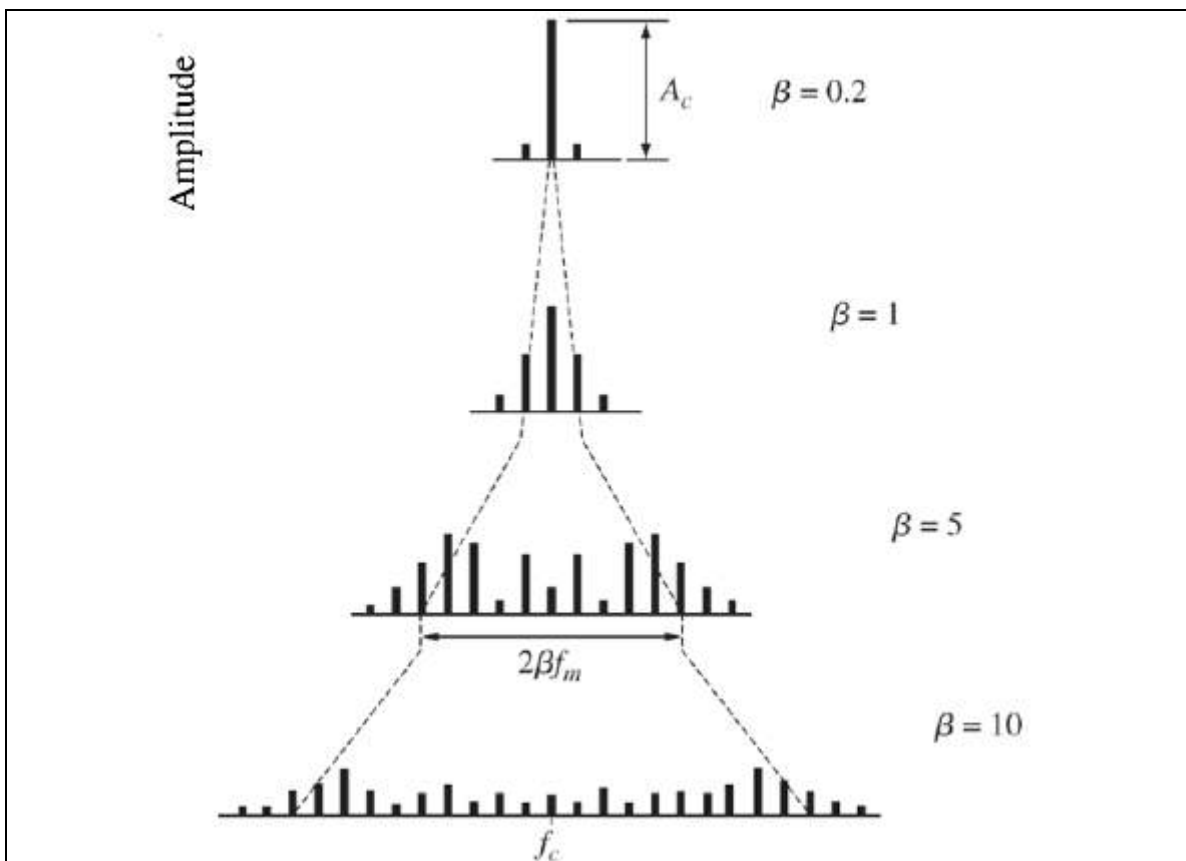


Fig. 2-17(c). FM single-sided spectra, for different values of the modulation index (β)

When the modulation index β is limited, we may have a narrow-band FM (NBFM) signal with limited bandwidth, which may be expressed by Carson's rule.

$$BW (NBFM) \approx 2 (\beta + 1) f_m \quad (2-11)$$

The commercial FM radio band is from 88 to 108 MHz between VHF television Channels 6 and 7. The FM stations are assigned center frequencies at 200 kHz separation starting at 88.1 MHz, for a maximum of 100 stations. These FM stations have a 75 kHz maximum deviation from the center frequency, which leaves 25 kHz upper and lower "guard bands" to minimize interaction with the adjacent frequency band. This is known as wideband FM (WBFM). These signals are capable of supporting high quality transmissions, but occupy a large amount of bandwidth.

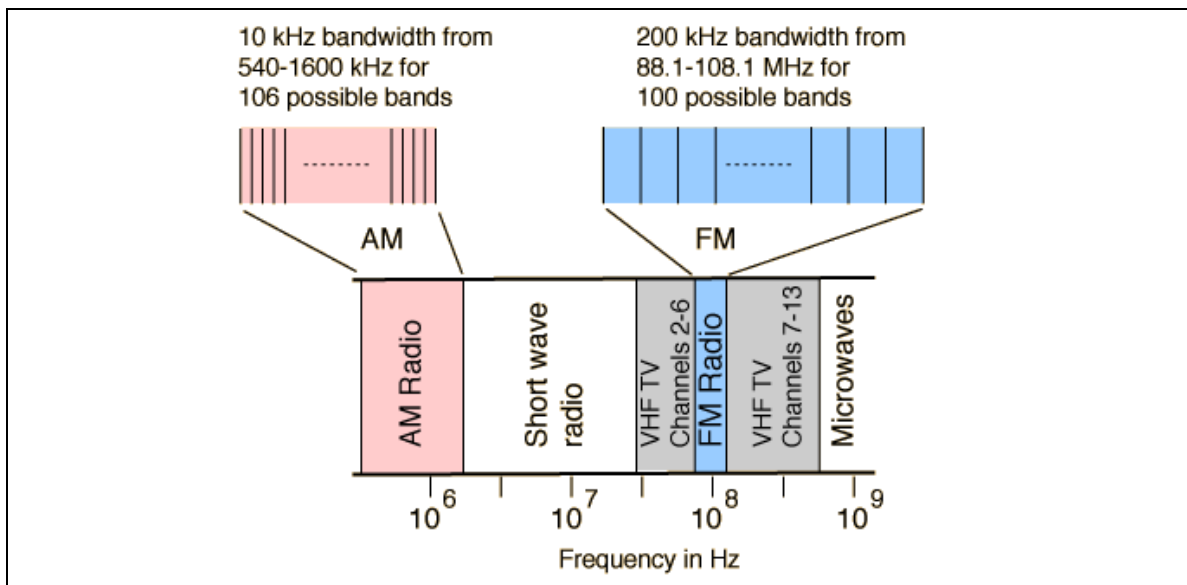


Fig. 2-18(a). AM and FM radio frequencies

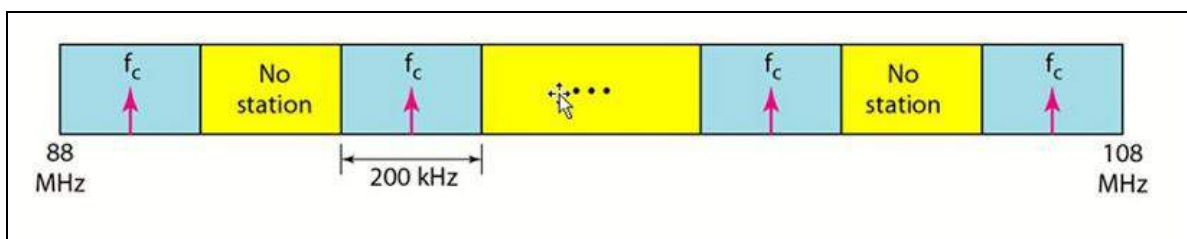


Fig. 2-18(b). FM radio frequencies

2-4.3. FM System Architecture

The following figure depicts the FM receiver architecture. In order to detect FM signals, the FM receiver should be sensitive to the frequency variations of the incoming signals.

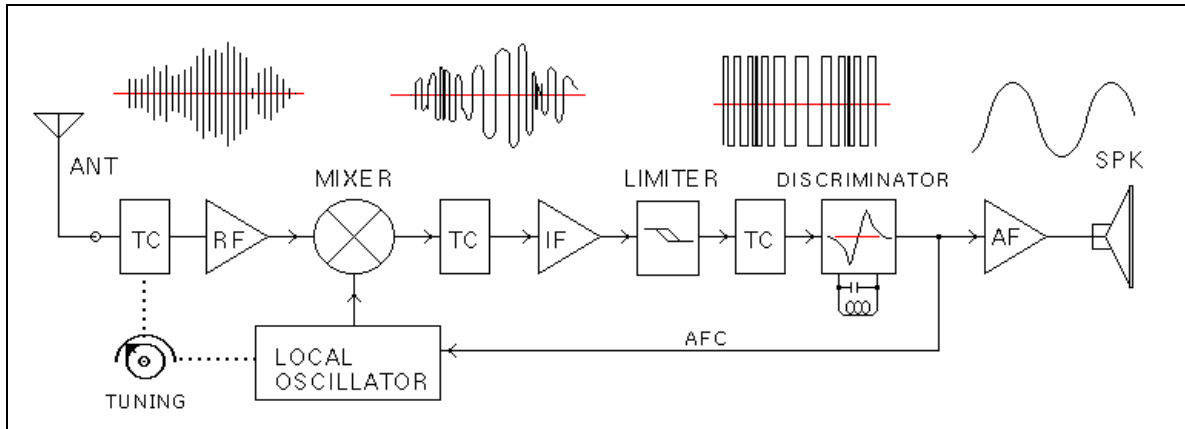


Fig. 2-19. Block diagram of an FM receiver, showing the frequency decimation stage.

There exist several types of FM demodulators, such as:

- **Foster-Seeley** discriminator (or phase-shift discriminator),
- **Slop detector**
- **Ratio detector**,
- **Quadrature detector** (or gated-beam detector),
- **Phase-locked loop (PLL)**.

i. Foster-Seeley discriminator

The Foster-Seeley discriminator (phase-shift discriminator) is composed of an electronic filter (differentiator), followed by an AM demodulator (envelop detector). If the filter response changes linearly with frequency, the final output will be proportional to the input frequency, as desired. The following figure shows the circuit diagram of a Foster-Seeley demodulator and its transfer characteristics (the s-curve).

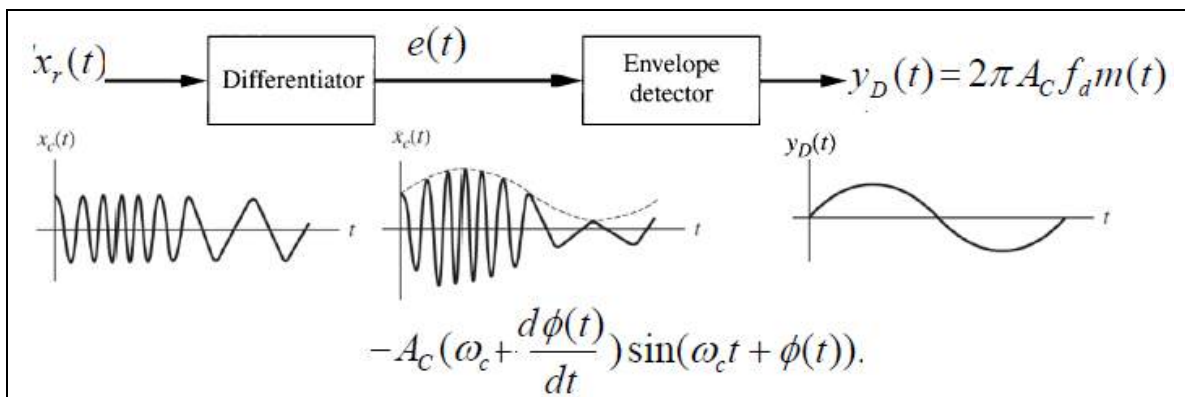


Fig. 2-20. Fooster-Seeley detector

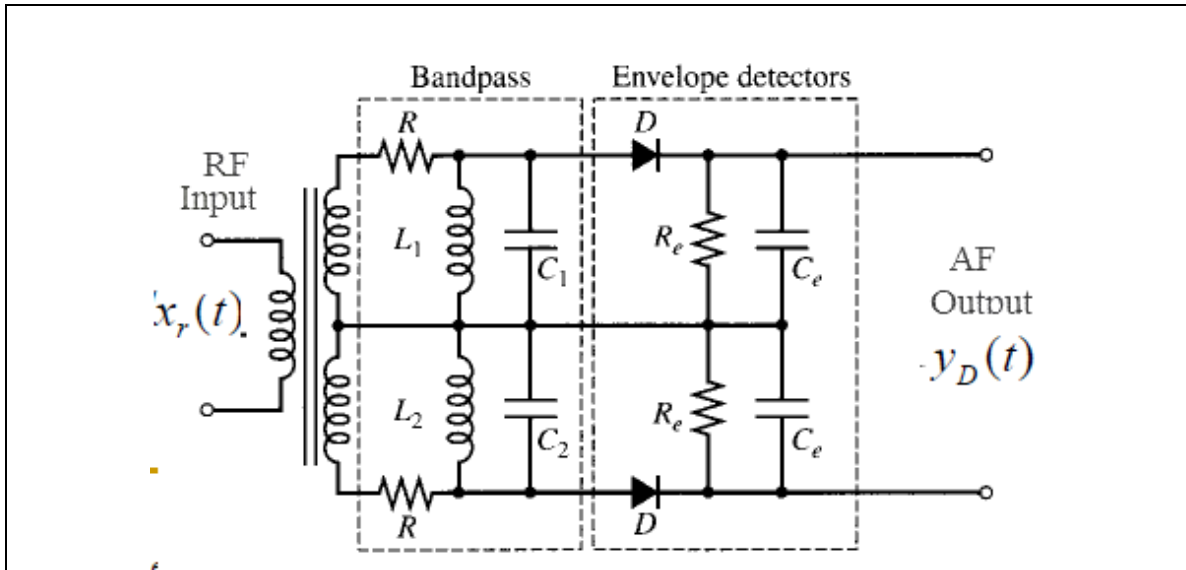


Fig. 2-21. Foster-Seely detector and its circuit implementation

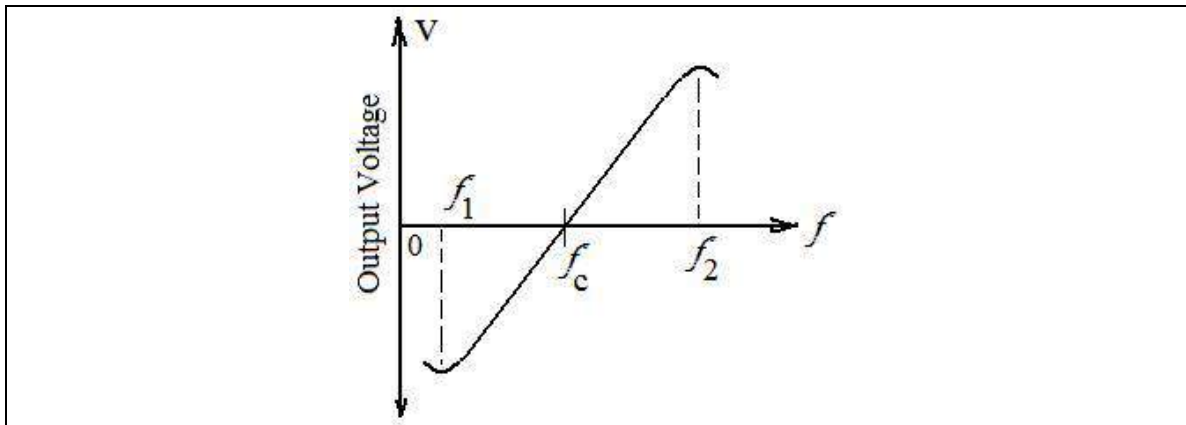


Fig. 2-22. The S-curve characteristics of the Foster-Seely FM detector

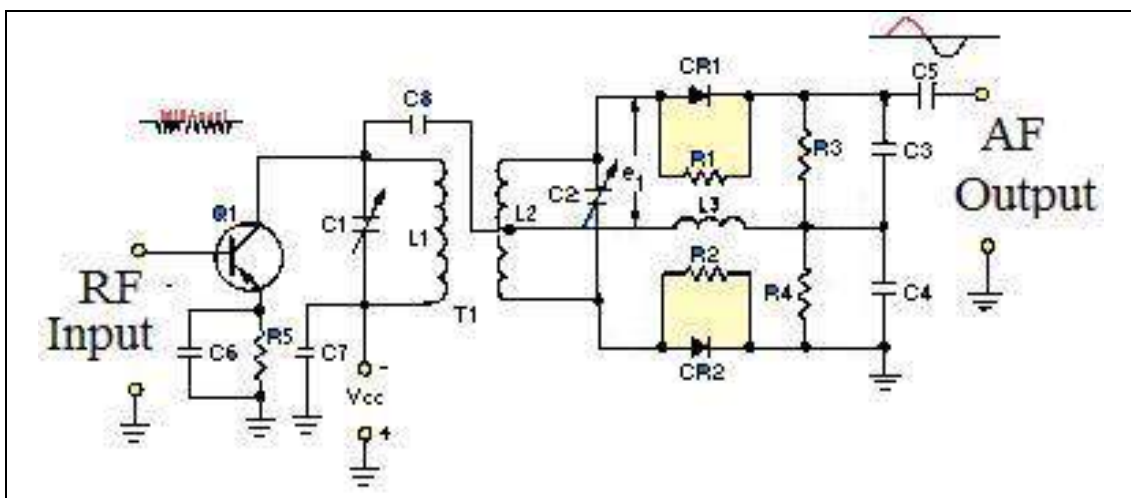


Fig. 2-23. Real circuit of a Foster-Seely frequency discrimination, ,

The output of the Foster-Seeley discriminator is affected not only by the input frequency, but also by the input amplitude. Therefore, using a limiter stages before the detector is necessary.

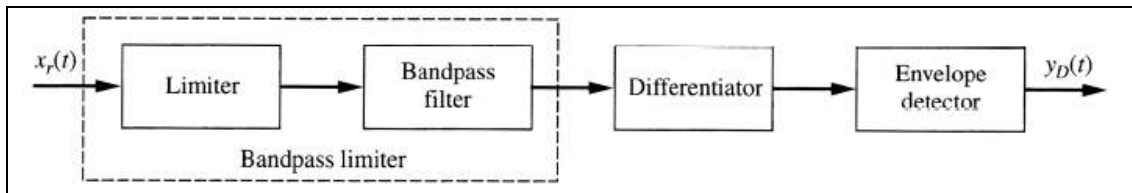


Fig. 2-24. Addition of a limiter stage, before the discrimination, in the FM receiver

ii. Ratio Detector

The ratio detector uses a double-tuned transformer to convert the instantaneous frequency variations of the fm input signal to instantaneous amplitude variations. These amplitude variations are then rectified to provide a DC output voltage which varies in amplitude and polarity with the input signal frequency. Figure 2-24 shows a typical **ratio detector** circuit; where the input tank capacitor (C1) and the primary of transformer T1 (L1) are tuned to the center frequency of the FM signal to be demodulated. This detector demodulates FM signals and suppresses amplitude noise without the need of limiter stages.

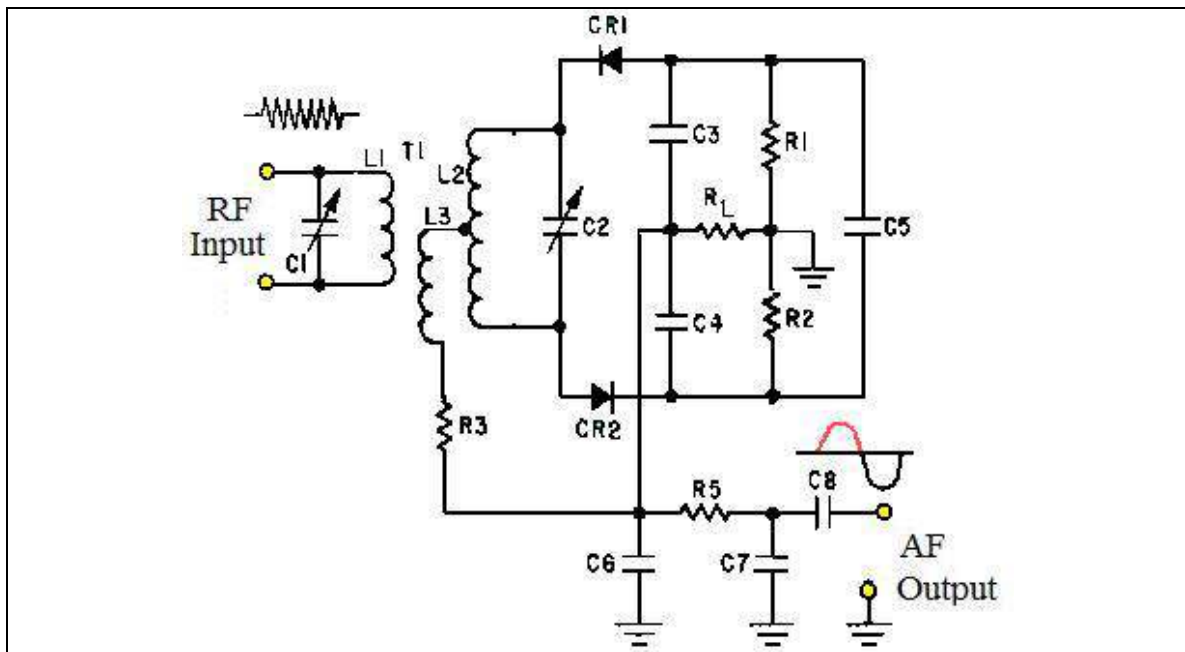


Fig. 2-25. Circuit diagram of a ratio detector

iii. Quadrature Detector

The quadrature detector uses a phase shift network and a phase detector to shift the signal by 90° and multiplies it with the un-shifted signal. The

multiplier may be replaced by a phase detector (PD) to compare the phase of the IF signal (v_1) to v_2 , the signal generated by passing v_1 through a phase shift network. One of the terms that come out of this operation is the original signal, which is selected by a low-pass filter (LPF) and amplified. The primary advantage of the quadrature detector lies in its extreme simplicity. It requires relatively few components to provide linear detection and very easy to be integrated.

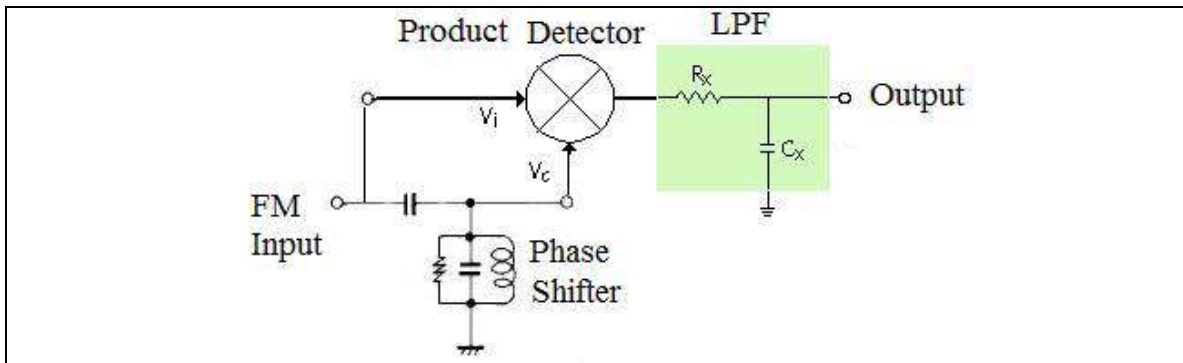


Fig. 2-26. Circuit diagram of a quadrature FM detector

iv. Phase-locked loop (PLL)

In the phase-locked loop (PLL), the difference between the modulated signal and another feedback signal from a voltage-controlled oscillator (VCO) is produced. The error signal is then considered as the demodulated signal. A simplified diagram of an FM detector is shown in figure 2-27, showing the frequency decimation stage.

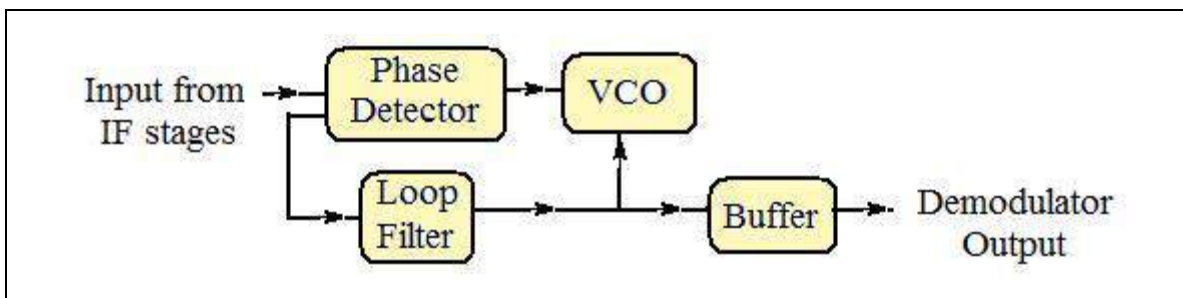


Fig. 2-27. Circuit diagram of a simple PLL FM detector

Recently, the quadrature and PLL demodulators became more widely used than the Foster Seeley discriminator and the Ratio detector. The main reason for this is that the **Foster-Seely** and ratio detectors require the use of wound inductors and these are hard to integrate on chip. Other types of FM demodulator lend themselves more easily to being incorporated into integrated circuits.

2-4.4. Pre-emphasis and De-emphasis

It has already been mentioned that FM can give a better signal to noise ratio than AM when wide bandwidths are used. The amplitude noise can be removed by limiting the signal in the FM receiver. An **additional improvement** in signal to noise ratio can be achieved if the audio signal is pre-emphasized. In order to achieve this, the lower level high frequency sounds are amplified to a greater degree than the lower frequency sounds before they are transmitted. Once the receiver signals are passed through a network with the opposite effect to restore a flat frequency response.

To achieve the pre-emphasis the signal is passed through a capacitor-resistor (CR) network. At frequencies above the cut-off frequency the signal increases in level by 6 dB per octave. Similarly at the receiver the response falls by the same amount. Both the receiver and transmitter networks must match one another. In the UK the CR time constant is chosen to be $50\mu\text{s}$. For this the break frequency f_1 is 3183 Hz. For broadcasting in North America values of $75\mu\text{s}$ with a break frequency of 2.1 kHz is used.

Pre-emphasizing the audio for an FM signal is effective because the noise output from an FM system is proportional to the audio frequency. In order to reduce the level of this effect, the audio amplifier in the receiver must have a response that falls with frequency. In order to prevent the audio signal from losing the higher frequencies, the transmitter must increase the level of the higher frequencies to compensate. This can be achieved because the level of the high frequency sounds is usually less than those lower in frequency.

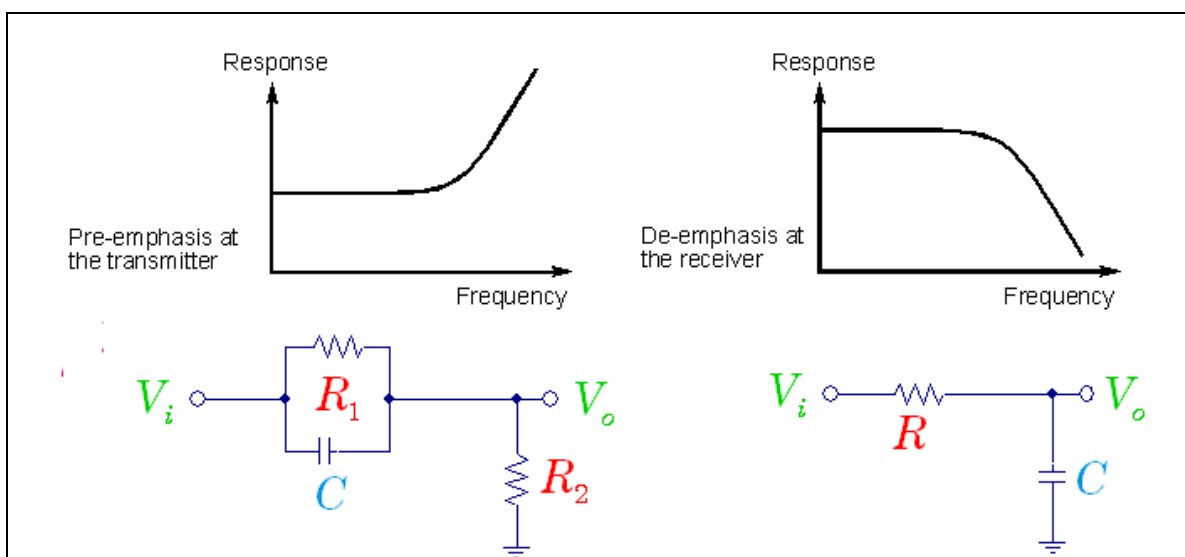


Fig. 2-28. Pre-emphasis and De-emphasis of FM signals and their circuits

2-4.5. Stereo FM Broadcast

In recent years stereo transmission has become an accepted part of VHF FM transmissions. The system that is used maintains compatibility with mono only receivers without any noticeable degradation in performance. A stereo signal consists of two channels that can be labelled L and R, (Left and Right), providing one channel for each of the two speakers that are needed. An ordinary mono signal consists of the summation of the two channels, i.e. $L + R$, and this can be transmitted in the normal way.

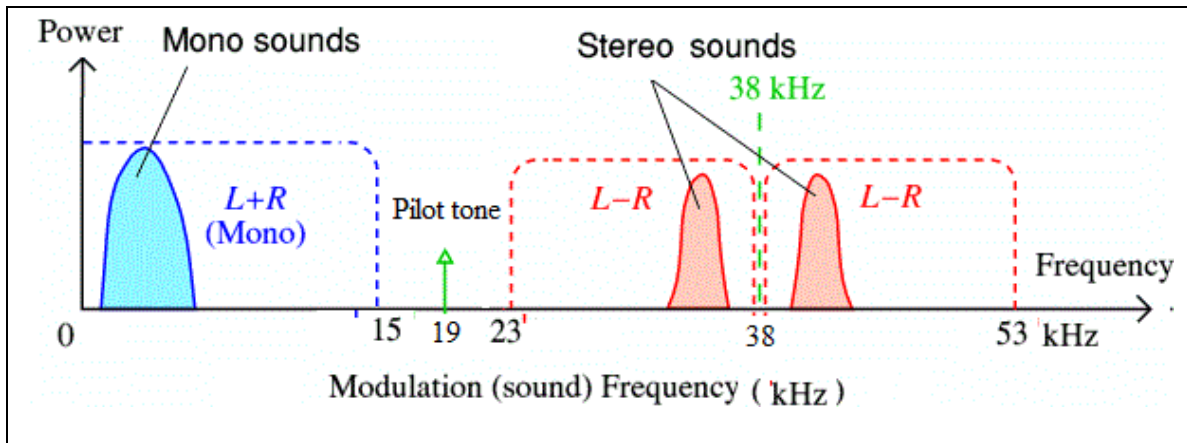


Fig. 2-29. Stereo broadcasting of FM signals (low frequency (modulating) signals).

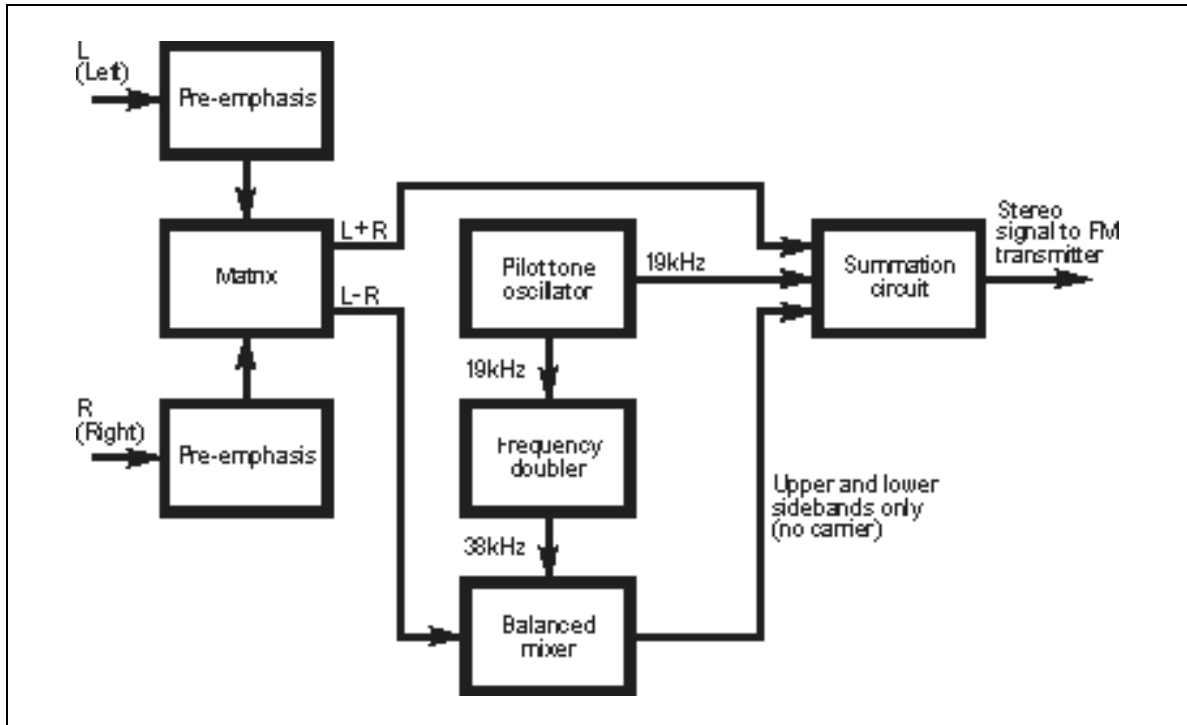


Fig. 2-30(a). Block diagram of a stereo FM system (transmitter side)

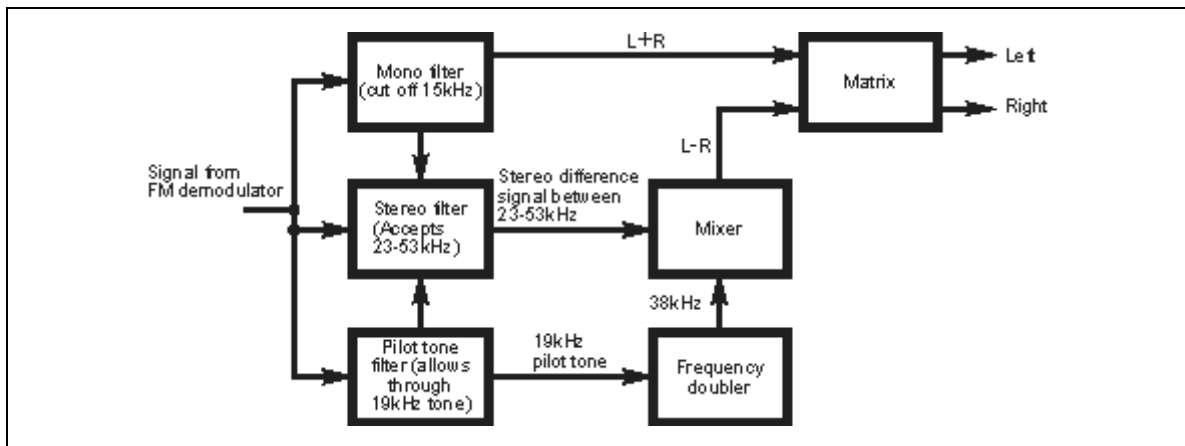


Fig. 2-30(b). Block diagram of a stereo FM system (receiver side)

2-5. Noise in Radio Systems

We have defined noise, in Chapter 1, as any electrical signal present in a system other than the desired signal. We mentioned that all electrical systems have noise. The noisiness of a signal is specified by the signal-to-noise ratio (*SNR*). We have also defined the noise figure (*NF*) and noise factor (*F*) as measures of degradation of the signal-to-noise ratio, caused by components in a radio system. Therefore, the noise factor is a number by which the performance of a radio receiver can be specified.

In this section we present how to model noise in radio systems.

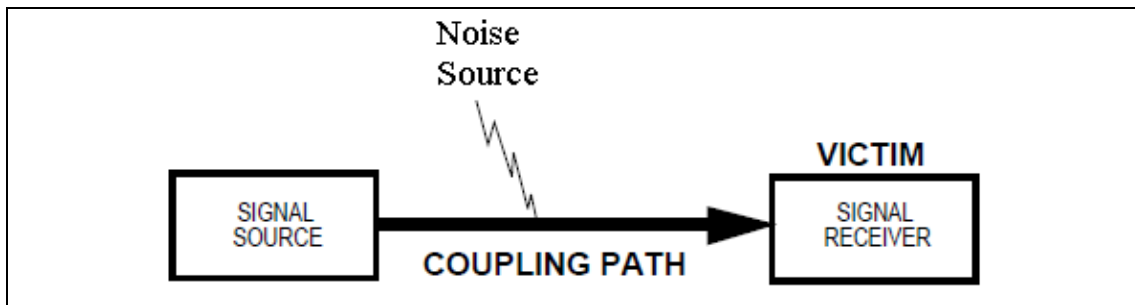


Fig. 2-31. A communication system, subjected to noise attack

2-5.1. Friis Formula for Noise Factor

Friis' first formula is used to calculate the overall noise factor (*F*) of a receiver, which is in turn composed of several stages, each with its own noise figure (*F_i*) and gain (*G_i*). It is given by the following relation:

$$F = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \frac{F_4 - 1}{G_1 G_2 G_3} + \dots + \frac{F_n - 1}{G_1 G_2 G_3 \dots G_{n-1}}, \quad (2-14)$$

where *F_i* and *G_i* are the noise factor and gain, respectively, of the *i*th stage, as illustrated in figure 2-32. Note that both magnitudes are expressed as ratios, instead of decibels. We can express the overall noise figure again in decibels by converting the resulting ratio.

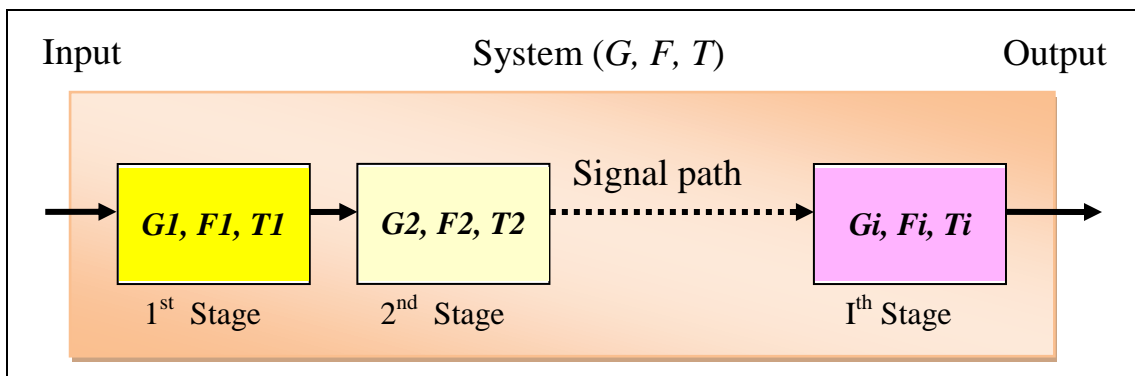


Fig. 2-32. Illustration of the Friis formula for a system of several cascaded stages.

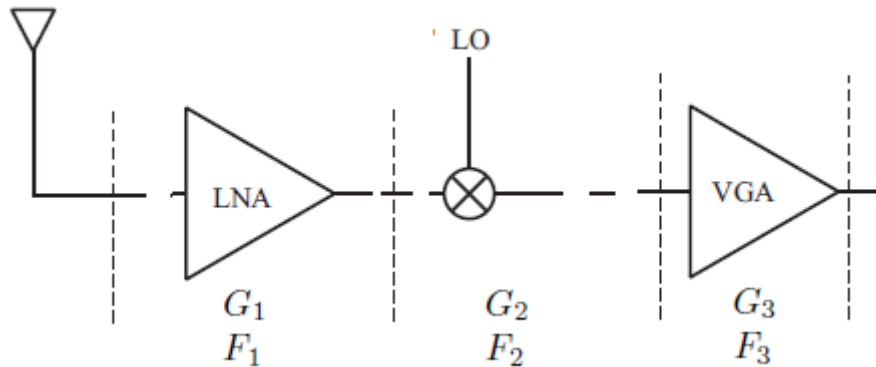
In the case of a radio receiver with the first stage being a low-noise amplifier (LNA), we can write

$$F_{receiver} = F_{LNA} + \frac{F_{rest} - 1}{G_{LNA}} \quad (2-15)$$

where F_{rest} is the overall noise figure of the subsequent stages. According to the equation, the LNA can reduce the overall noise figure of the receiver, but only if the gain is sufficiently high.

Example 2-3

The LNA has $G = 15$ dB and $NF = 1.5$ dB. The mixer has a conversion gain of $G = 10$ dB and $NF = 10$ dB. The IF amplifier has $G = 70$ dB and $NF = 20$ dB.



Solution

Even though the blocks operate at different frequencies, we can still apply the cascade formula if the blocks are impedance matched

$$F = 1.413 + \frac{10 - 1}{60} + \frac{100 - 1}{60 \cdot 10} = 2.4 \text{ dB}$$

Friis' formula can be equivalently expressed in terms of noise temperature, T , as follows:

$$T = T_1 + T_2/G_1 + T_3/G_1G_2 + \dots \quad (2-16)$$

As an example, the following figure depicts the noise temperature of a radio communication system (T_s), including the antenna and cable attenuation losses. Again, the system blocks without gain (e.g., cables and attenuators) have a noise figure equal to their **attenuation** L (in dB) when their physical temperature is equals to T_0 .

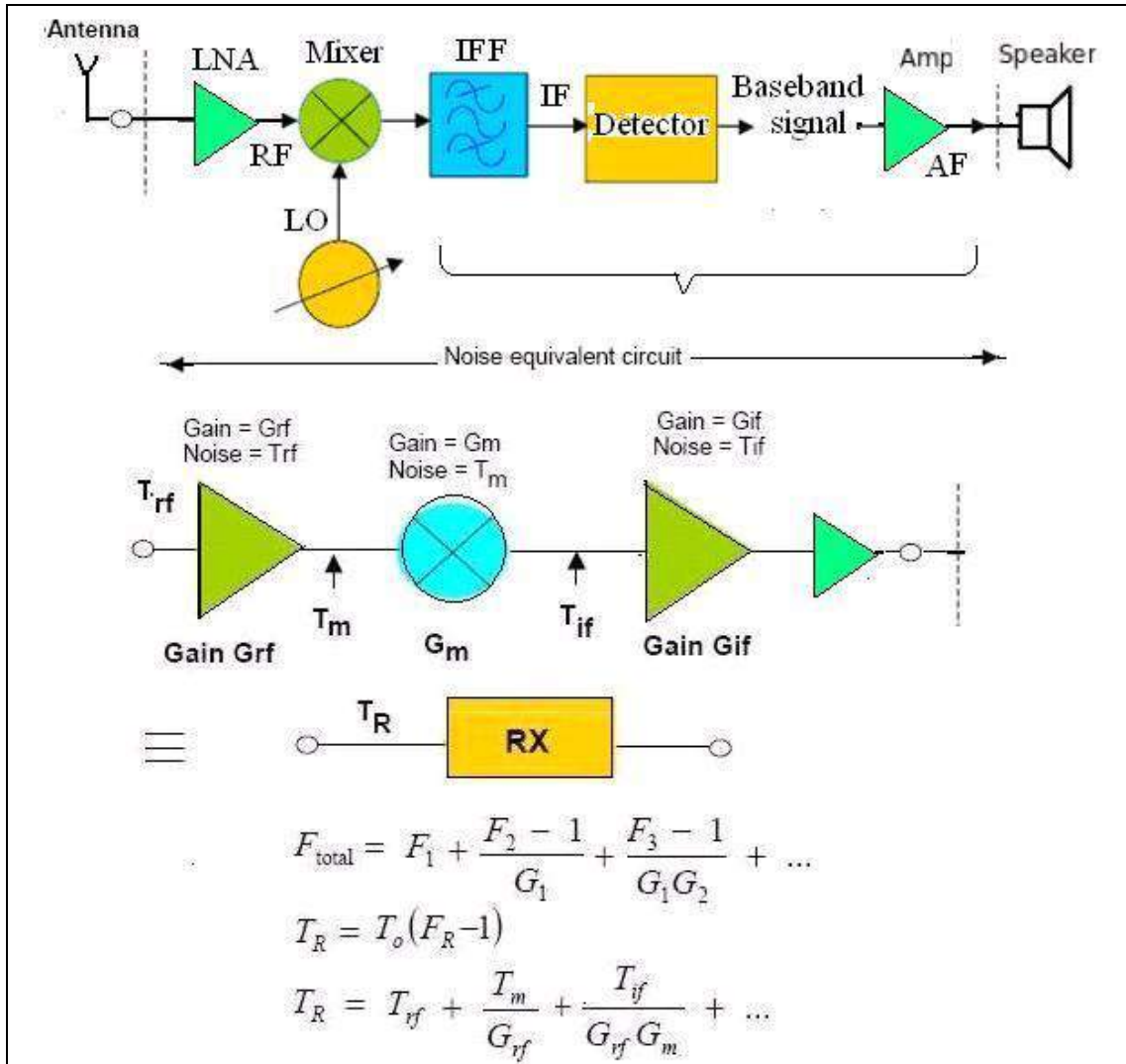


Fig. 2-33. Block diagram of a radio receiver and its equivalent noise diagram

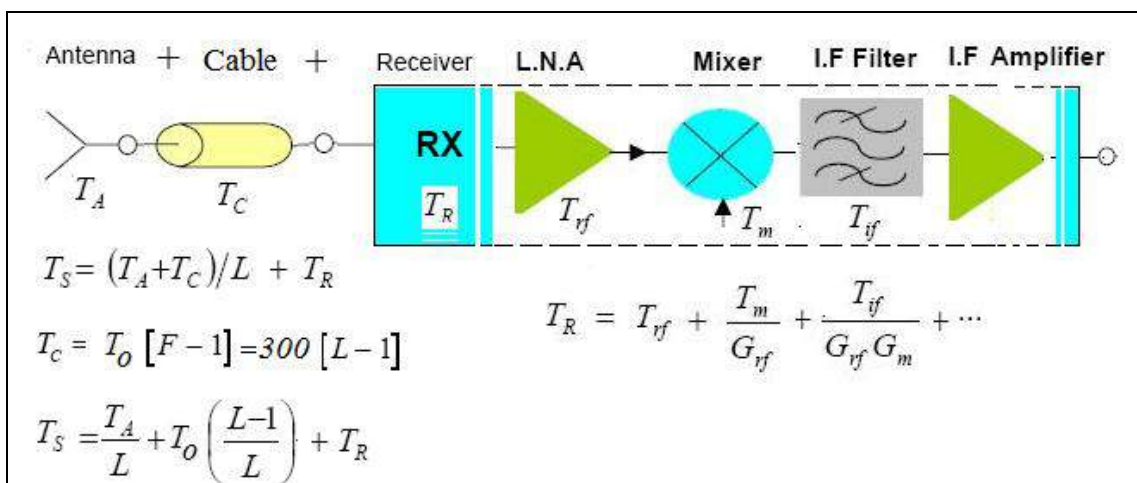


Fig. 2-34. Noise temperature of a radio receiver, with antenna and cable losses.

2-5.2. Minimum Detectable Signal (MDS) of a Radio Receiver

In addition to the desired signal, the receiver also picks up noise from different sources. Minimum detectable signal (**MDS**) in a radio receiver is the smallest signal power that can be received and demodulated by the receiver. This measurement is the minimum detectable (discernable) signal referred to the input and is equivalent to the input channel noise power (noise floor) plus the cascaded noise figure of the receiver:

$$MDS = CNP + CNF \quad (dBm), \quad (2-17)$$

where $CNP = k_B T \cdot B$ is the channel noise floor in a bandwidth B and CNF is the cascaded noise figure of the whole receiver, as given by the **Friis** formula, given by equation (2-3). Therefore, the above relation may be written as

$$MDS = 10 \cdot \log_{10}(kT/1mW) \text{ dBm} + (\text{noise figure, dB}) + 10 \log_{10}(\text{bandwidth, Hz})$$

If the system temperature (T) is 290K and bandwidth $B=1\text{Hz}$, then the effective noise power available in 1Hz bandwidth from a source is -174 dBm. This is the system's noise floor at its input. Any signal of lower power may not be discerned from noise, except for specific situations. 1 Hz noise floor: calculating the noise power available in a one hertz bandwidth at a temperature of $T = 290$ K defines a figure from which all other values can be obtained (different bandwidths, temperatures). 1 Hz noise floor equates to a noise power of -174 dBm so a 1 kHz bandwidth would generate $-174 + 10 \log_{10}(1\text{kHz}) = -144\text{dBm}$ of noise power (the noise is thermal noise, Johnson noise).

Example 2-1

If the thermal noise power input (noise floor) to a receiver at $T=293\text{K}$ is $k_B \cdot T = -174 \text{ dBm/Hz}$, and the channel bandwidth is $B = 1 \text{ MHz}$ (60 dB) what is the input channel power and what is the minimum detectable signal in a 50 Ohm source.

Solution

The input channel power would be $CNP = k_B \cdot T \cdot B = -174 \text{ dBm} + 60 \text{ dB} = -114 \text{ dBm}$. For a cascaded noise figure CNF of 5 dB, the minimum detectable signal is $MDS = CNP + CNF = -114 + 5 = -109 \text{ dBm}$ (0.01259 pW). In a 50 Ohm source this corresponds to $V_{rms} \approx 0.7934 \mu\text{V}$.

2-5.3. Sensitivity of a Radio Receiver

Sensitivity is one of the main specifications of any radio receiver. The receiver sensitivity indicates how well a receiver will capture weak signals. It is a key specification because it directly affects the range of the radio receiver system. Sensitivity of a receiver (S) is based on the thermal noise floor of the system CNP , its total noise figure (CNF), and the minimum required S/N ratio for the receiver system,

$$S = MDS + S/N = CNP + CNF + S/N \quad (2-18)$$

If the minimum detectable signal of a receiver is -109dBm , like the last example, and the required S/N ratio is at least 20dB , then the receiver sensitivity is $S = -109 + 20 = -89\text{ dBm}$. In a 50 Ohm source this corresponds to $V_{rms} = 7.934\text{ }\mu\text{V}$. Therefore the sensitivity of such a receiver is about $8\text{ }\mu\text{V}$, that's it can detect and demodulate low signals as weak as $8\text{ }\mu\text{V}$ at its input (from antenna), with a signal-to-noise ratio S/N as good as 20 dB . The sensitivity of a radio receiver is by no means the whole story. The specification for a set may show it to have an exceedingly good level of sensitivity, but when it is connected to an antenna its performance may be very disappointing because it is easily overloaded (saturated) when strong signals are present, and this may impair its ability to receive weak signals. In today's radio communications environment where there are very many transmitters close by and further away, good levels of sensitivity are needed along with the ability to handle strong signals and the dynamic range of the radio receiver is very important.

2-6. Testing of a Radio Receiver

When you are employed in communication electronics, your work will involve some form of testing and measurement. The work may be installation, operation, servicing, repair, and maintenance or testing to certain standards. Testing of a radio receiver is not only important for maintenance purposes, but also for quality control in production lines. There exist three basic features that characterize the performance of any radio receiver, namely: Sensitivity, Selectivity and Fidelity. The following figure shows the test block diagram of a radio receiver. Note that the antenna is replaced by an artificial antenna (or dummy load, typically of 50Ω or 75Ω). Also, the loud speaker is replaced by a dummy load (typically of 8Ω) with an output indicator (voltmeter).

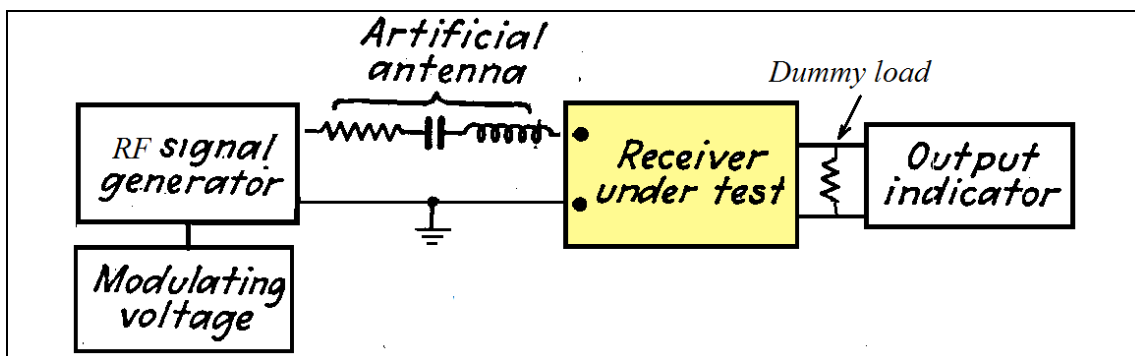


Fig. 2-35. Testing block diagram of a radio receiver.

2-6.1. Sensitivity Test

The primary tests for radio receivers involve sensitivity and noise level. The greater the sensitivity of the receiver, the higher its gain and the better job it does of receiving very small signals. The receiver sensitivity measurements are usually made by measuring the input signal voltage that produce a certain speaker output voltage, at different values of input carrier frequency.

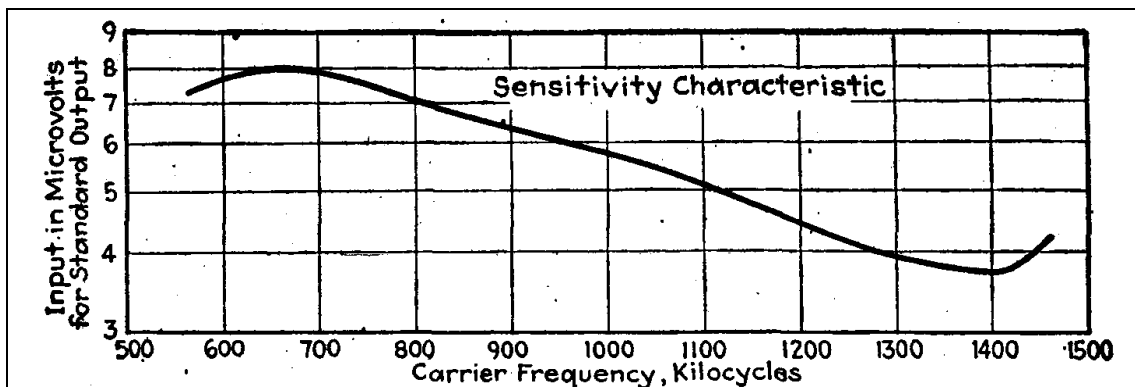


Fig. 2-36. Sensitivity characteristics of a radio receiver.

You need an RF signal generator to provide the input carrier signal. It should have an attenuator so that signals as low as $1\mu\text{V}$ or less can be set. The carrier may be arbitrary modulated at 30% by 400Hz signal. The above figure depicts the sensitivity characteristics of a radio receiver.

Actually, there exist two basic methods, **quieting** sensitivity test and **SINAD** sensitivity test. The quieting method measures the amount of signal needed to reduce the output noise to 20dB. As the signal level increases, the noise level decreases until the limiters in the IF section begin to start their clipping action. When this happens, the receiver output quiets; so that its output becomes silent and blanks out the noise.

The SINAD test is a measure of the input signal voltage that will produce at least a 12- dB signal-to-noise ratio. The noise value includes any harmonics that are produced by the receiver circuits because of distortion.

Follow this procedure to make the 20-dB quieting measurement:

1. Turn on the receiver, and set it to an unused channel.
2. Leave the signal generator off so that no signal is applied.
3. Set the receiver gain to maximum with any RF or IF gain control, if available.
4. Adjust the volume control of the receiver so that you read some convenient value of noise voltage on the meter connected across the speaker. One volt rms is a good value if you can achieve it; but if not, any other convenient value will do.
5. Turn on the signal generator, but set the output level to zero or some very low value. Adjust the generator frequency to the center of the receiver's channel setting. Turn off the modulation so that the generator supplies carrier only.
6. Increase the signal generator output signal level a little at a time, and observe the voltage across the speaker. The noise voltage level will decrease as the carrier signal gets strong enough to overpower the noise. Increase the signal level until the noise voltage drops to one-tenth of its previous value.
7. Measure the generator output voltage on the generator meter or the external RF voltmeter.
8. If an attenuator pad or other impedance-matching network was used, subtract the loss it introduces. The resulting value is the voltage level that produces 20 dB of quieting in the receiver.

2-6.2. Selectivity Test

Selectivity of a radio receiver is its ability to discriminate between radio signals of different carrier frequencies (channels). It is usually expressed by tuning curves, which show the amount by which the input RF signal must be increased in order to maintain the standard output, as the carrier frequency is varied, around a certain frequency (tuned channel). Figure 2-37(a) depicts the selectivity characteristics of a typical radio receiver. Note that the IF tuner, is used to adjust the IF resonance (center) frequency.

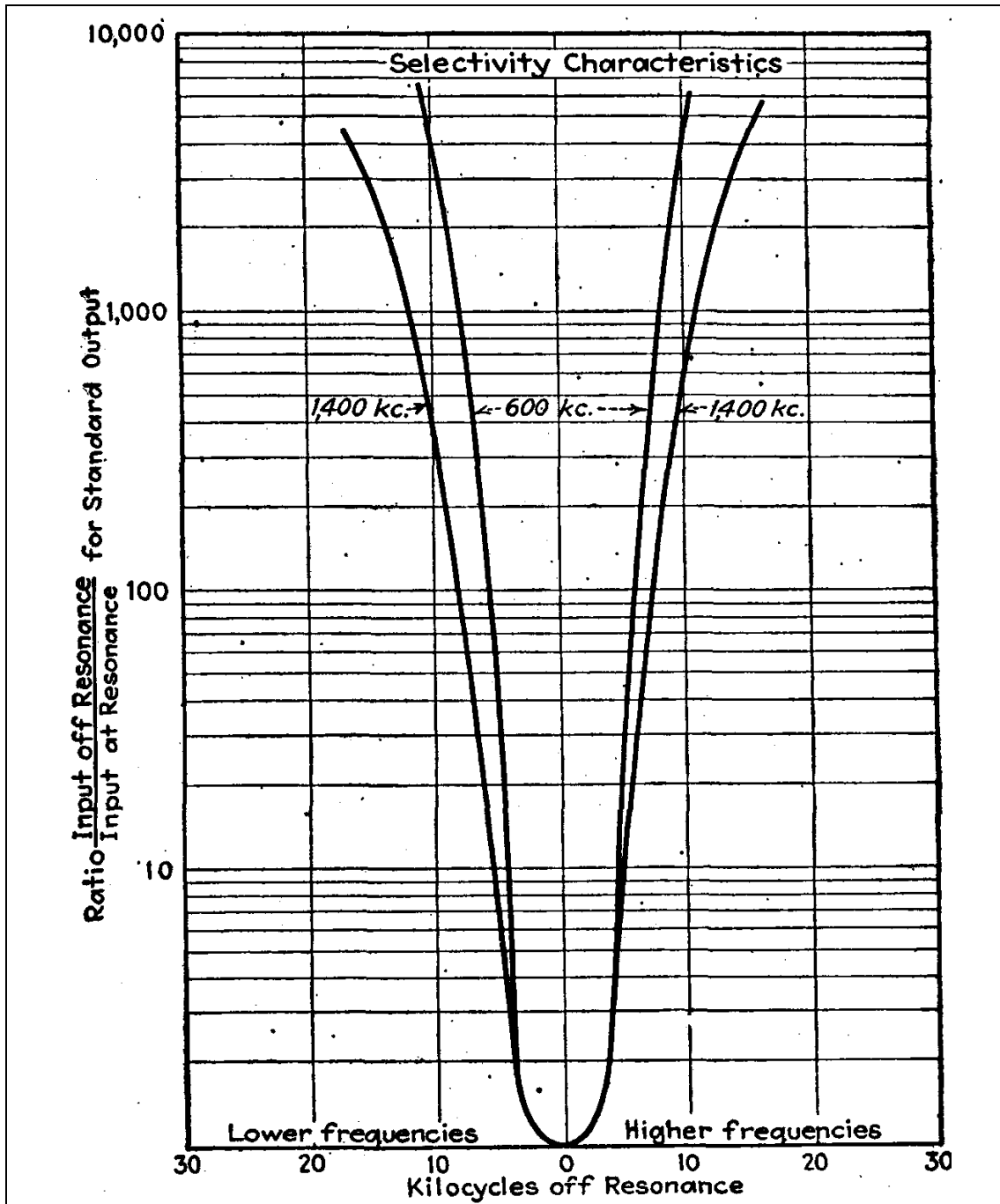


Fig. 2-37(a). Selectivity characteristics of a radio receiver.

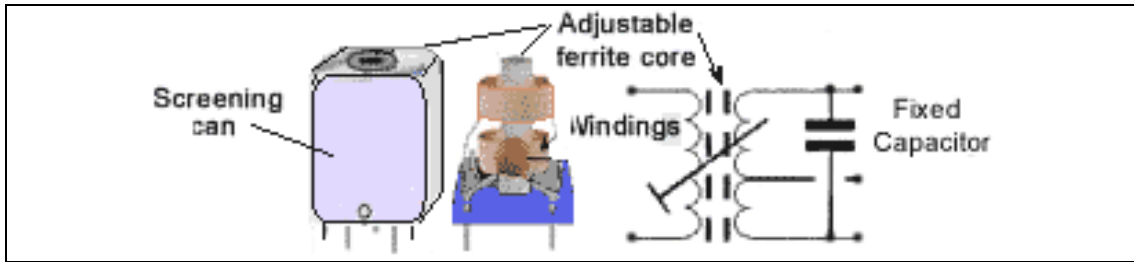


Fig. 2-37(b). Schematic of the IF tuner in a radio receiver.

2-6.3. Fidelity Test

Fidelity of a radio receiver is the range of operation, over which it can detect and reproduce different modulating frequencies, without distortion. The fidelity is expressed in curves, which shows the variation in audio-frequency output voltage, as the modulation frequency of the RF signal is varied. The following figure depicts the fidelity characteristics of a typical radio receiver.

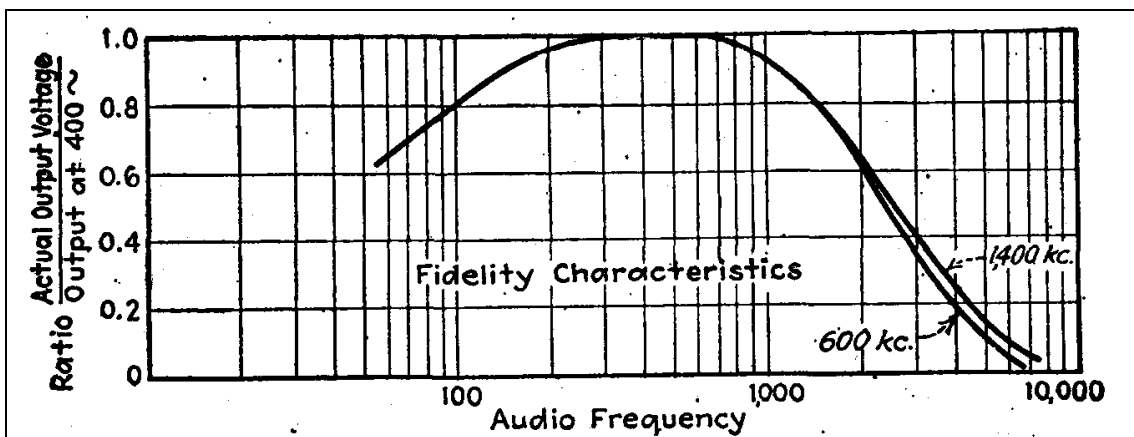


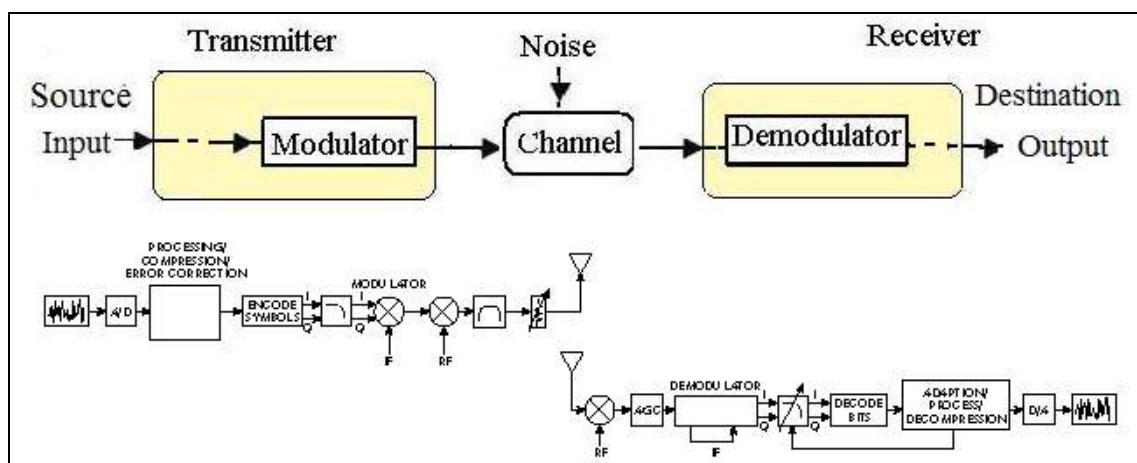
Fig. 2-38. Fidelity characteristics of a radio receiver.

2-7. Summary

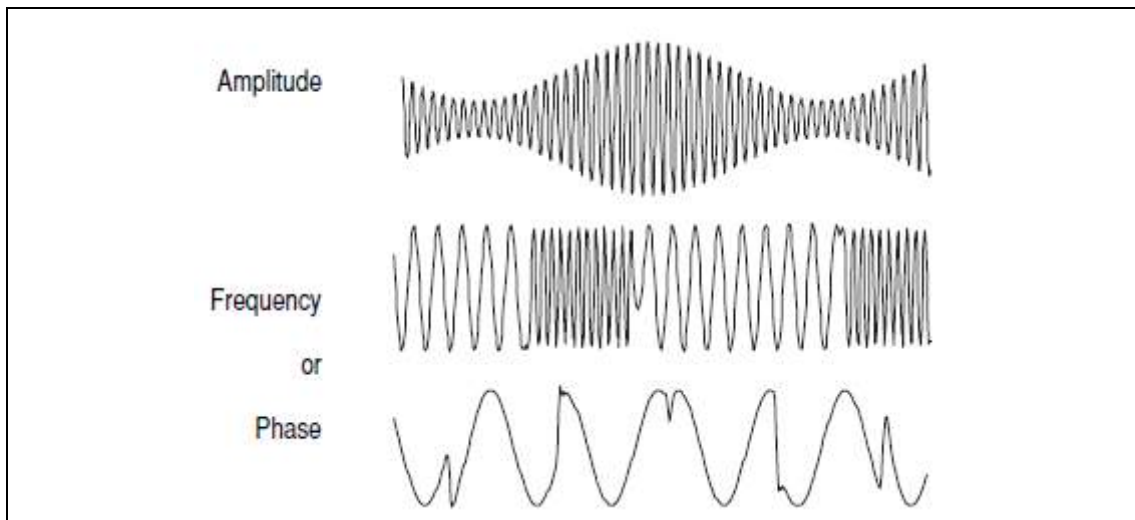
Analog communication is that types of communication in which the message or information signal i.e transmitted is analog in nature. This means that in analog communication the modulating signal (i.e base-band signal) is an analog signal. This analog message signal may be obtained from sources such as speech, video shooting etc.

Radio and TV represent distinct traditions within the field of telecommunications. *Radio communication system* sends signals by radio. Types of radio communication systems deployed depend on technology, standards, regulations, radio spectrum allocation, user requirements, service positioning, and investment.

The radio equipment involved in communication systems includes a transmitter and a receiver, each having an antenna and appropriate terminal equipment such as a microphone at the transmitter and a loudspeaker at the receiver in the case of a voice-communication system



Modulation is the process of varying a periodic waveform, in order to use that signal to convey (transmit) a message. Normally a **high-frequency** sinusoidal wave is used as a carrier signal. The various modulation techniques in analog systems (such as amplitude modulation, **AM** frequency modulation **FM** or phase modulation **PM**), offer different solutions in terms of cost-effectiveness and quality of received signals. Amplitude modulation is one of the simplest and most popular forms of modulation for commercial broadcast. Some homes still have analogue radio receivers and televisions.



Amplitude modulation (**AM**) is the most straightforward way of modulating a signal. Demodulation, or the process where the radio frequency signal is converted into an audio frequency signal is also very simple. An amplitude modulation signal only requires a simple diode detector circuit. The amplitude modulated signal consists of a carrier with two sidebands that extend out from the main carrier. Like any other system of modulation, amplitude modulation has several advantages and disadvantages. While amplitude modulation is one of the simplest and easiest forms of signal modulation to implement, it is not the most efficient in terms of spectrum efficiency and power usage.

The most obvious method of applying modulation to a signal is to superimpose the audio signal onto the amplitude of the carrier. This results in amplitude modulated (**AM**) signals. However this is by no means the only method which can be employed. It is also possible to vary the frequency of the signal to give frequency modulation (**FM**).

The performance of a radio receiver can be characterized by three features, namely: its **sensitivity**, **selectivity** and **fidelity**. The **sensitivity** of a radio receiver reflects its ability to detect weak RF signals. It is measured in terms of the voltage that must be induced in the antenna by an RF signal to produce a standard output (before the speaker terminals). The standard output may be chosen as 0.05W in 8 Ω dummy load.

Selectivity is the property that enables a radio receiver to discriminate between radio signals of different carrier frequencies. Selectivity cannot be defined in a single term as can sensitivity but must be expressed in the form of curves, which show the amount by which the signal input must be increased in order to maintain the standard output as the carrier frequency is varied from the frequency to which the receiver is tuned.

2-8. Problems

- 1-1)** Who was first to transmit voice over radio?
- 1-2)** Who was first to send two-way wireless telegraphy messages across the Atlantic Ocean?
- 1-3)** What are the main differences between analog and digital systems?
- 1-4)** What are the main benefits of modulation techniques?
- 1-5)** Mention some applications of both AM and FM modulation techniques in our daily life communication systems.
- 1-6)** What are the main types of modulation & coding methods?
- 1-7)** Mention some applications of both AM and FM modulation techniques in our daily life communication systems.
- 1-8)** Give signal-to-noise ratio guidelines at a receiving device for the following three media: (1) voice, (2) video-TV, and (3) data.
- 1-9)** What are the main impairments, which we face in data communication?
- 1-10)** What are main types of noise that affect the communication equipment?
- 1-11)** What are the main types of transmission channels?
- 1-12)** Find the total noise figure, of a system of 2 stages, the first has gain and noise figure of 20dB and 2dB and the second stage has a gain and noise figure of 16 dB and 12dB.

2-9. References

- [1] F. E. **Terman**, Radio Engineering, McGraw-Hill, New York & London, **1937**.
- [2] H. P. **Hsu**, Analog and Digital Communications, McGraw-Hill, **1993**.
- [3] A. A. **Abidi**, "Direct-conversion Radio Transceivers for Digital Communications," IEEE Journal of Solid-state Circuits, Vol. 30, No. 12, December **1995**.
- [4] A. V. **Oppenheim**, et al., Signals and Systems, 2nd Ed., Prentice-Hall, **1996**.
- [5] W. Leon **Couch**, Digital and Analog Communications. Upper Saddle River, NJ: Prentice-Hall, **1997**.
- [6] Giovanni **Girlando** and Guiseppa **Palmisano**, Noise Figure and Impedance Matching in RF Cascode Amplifiers, IEEE Transactions on Circuits and Systems-II, vol. 46, no. 11, November **1999**.
- [7] Simon **Haykin**, Communication Systems, 4th Ed., Wiley, **2001**.
- [8] M. **Thomas** and A. Thomas Joy, Elements of information theory, 1st Edition. New York: Wiley-Interscience, 1991. 2nd Edition, New York: Wiley-Interscience, **2006**.
- [9] K. **Giridhar**, "Wireless Communications – Principles & Practice" 2nd Ed., **2008**.
- [10] A. Bruce **Carlson** , Paul B. Crilly, Communication Systems, 5th Edition, UNIV OF TENNESSEE KNOXVILLE, **2010**,
- [11] M. **Farooque** Mesiya, Contemporary Communication Systems, 1st Edition, **2013**.

Chapter
3

Pulse Modulation, Multiplexing & Detection Techniques

Contents

- 3-1. **Introduction**
- 3-2. **Pulse Amplitude Modulation (PAM)**
- 3-3. **Pulse Width Modulation (PWM)**
- 3-4. **Pulse Position Modulation (PPM)**
- 3-5. **Pulse Code Modulation (PCM)**
 - 3-5.1. Modulation Process
 - 3-5.2. Quantization Errors
 - 3-5.3. Encoding
 - 3-5.4. PCM Demodulation
 - 3-5.5. Data Compression / Expanding (**Companding**)
 - 3-5.6. Applications of PCM
 - 3-5.7. Other Forms of PCM
- 3-6. **Sigma-Delta (Σ - Δ) Modulation**
 - 3-6.1. Implementation of Σ - Δ Modulators
 - 3-6.2. Quantization Errors in Σ - Δ Modulation
 - 3-6.3. Signal-to-Noise Ratio in Σ - Δ Modulation
 - 3-6.4. Dynamic range in Σ - Δ Modulators
 - 3-6.5. Higher order Σ - Δ Modulators
- 3-7. **Multiplexing & Multiple Access Schemes**
 - 3-7.1. Time Division Multiplexing (**TDM**)
 - i. TDMA Transmission
 - ii. **T1** Framing
 - iii. **E1** Framing
 - iv. Higher Order Multiplexing & **SDH**
 - v. Statistical TDM (**STDM**)
 - 3-7.2. Frequency Division Multiplexing (**FDM**)
 - 3-7.3. Code Division Multiplexing (**CDM**)
 - i. Code Division Modulation & Demodulation

Contents of Chapter 2 (Cont.)

- ii. Properties of Spreading Codes
- 3-7.4. Wave Division Multiplexing (**WDM**)
 - i. Dense DWM (**DWDM**)
 - ii. Synchronous Optical Network (**SONET**)
- 3-8. **Baseband Transmission Problems**
 - 3-8.1. Channel Limitations
 - 3-8.2. Inter-Symbol Interference (**ISI**)
 - 3-8.3. Jitter
 - 3-8.4. Eye Diagrams
- 3-9. **Pulse Detection & Matched Filters**
 - 3-9.1. Pulse Detection
 - 3-9.2. Matched Filters
 - 3-9.3. Practical Pulse Shaping
 - 3-9.4. Cosine-Raised Filters
- 3-10. **Summary**
- 3-11. **Problems**
- 3-12. **Bibliography**

Chapter

3

Pulse Modulation, Multiplexing & Detection Techniques

3-1. Introduction

Pulse modulation is used to transfer a narrowband analog signal over an **analog lowpass channel** as a binary signal, by modulating a pulse train. Some pulse modulation schemes also allow the narrowband analog signal to be transferred as a digital (discrete-time) signal with a fixed rate. These techniques are not modulation schemes in the conventional sense and they are not channel coding schemes, neither, but may be considered as source coding schemes, and in some cases analog-to-digital conversion (ADC) techniques.

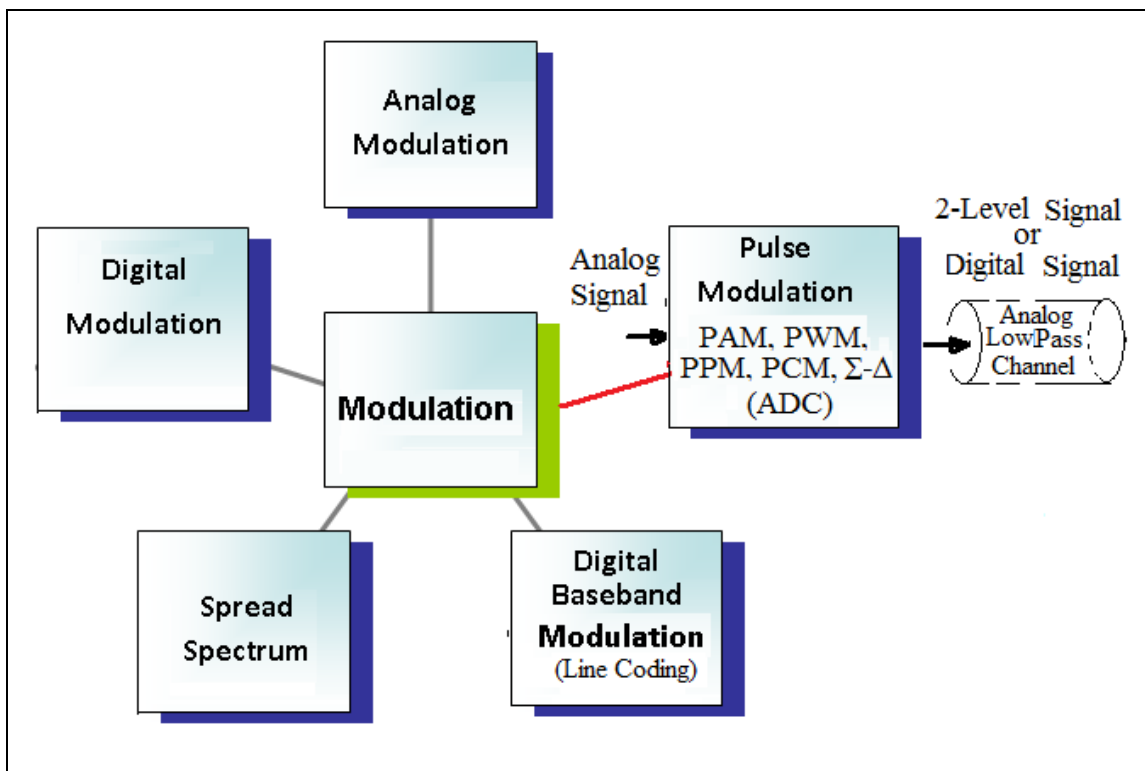


Fig. 3-1. Pulse modulation, among other modulation techniques.

There are three major classes of **pulse modulation** techniques used for source coding:

1- Analog-over-analog methods:

- Pulse-amplitude modulation (PAM)
- Pulse-width (or density) modulation (PWM)
- Pulse-position modulation (PPM)

3- Analog-over-digital methods:

- Pulse-code modulation (PCM)
 - Differential PCM (DPCM)
 - Adaptive DPCM (ADPCM)
- Delta modulation (Δ -modulation)
- Delta-sigma modulation ($\Sigma\Delta$ -modulation)
- Adaptive-delta modulation (ADM), also called Continuously-variable slope delta modulation (CVSDM),

3- **Direct-Sequence Spread Spectrum (DSSS)** is based on pulse-amplitude modulation (PAM).

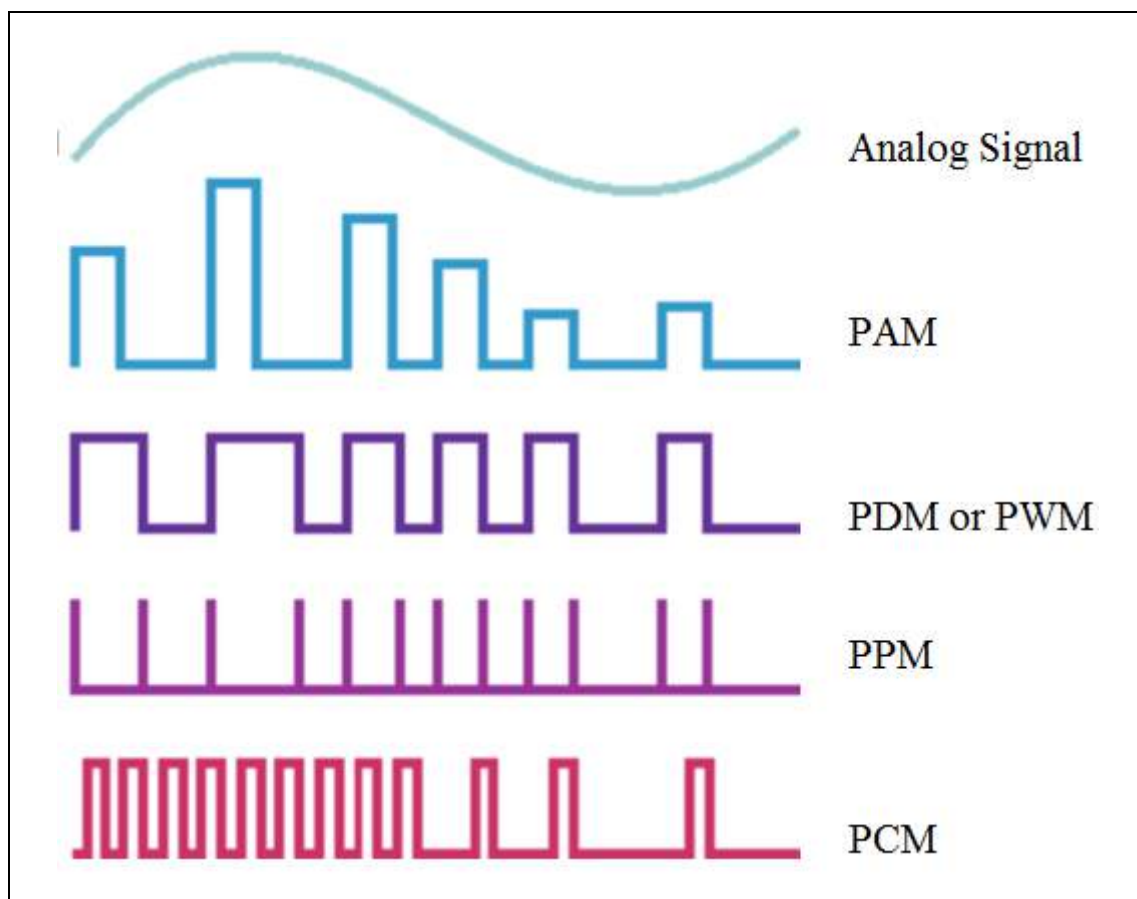


Fig. 3-2. Illustration of pulse amplitude modulation

3-2. Pulse-Amplitude Modulation (PAM),

Pulse-amplitude modulation (**PAM**) is a sort of modulation where the information is sampled in the amplitude by a series of signal pulses.

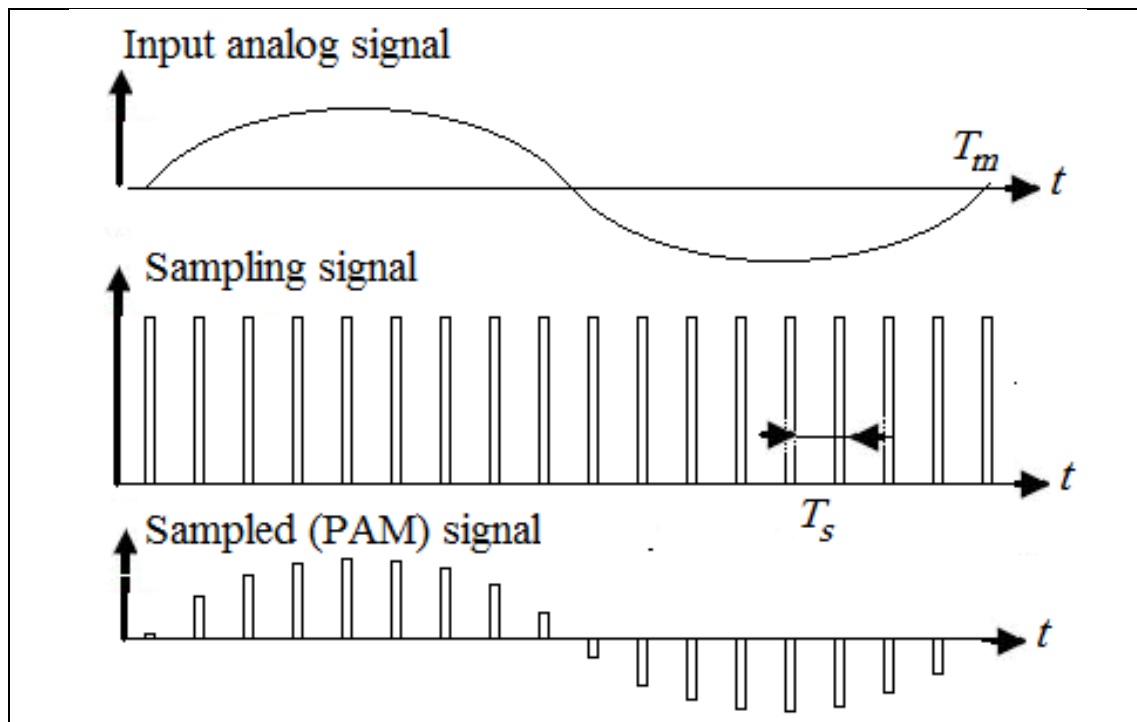


Fig. 3-3. Illustration of pulse amplitude modulation

The signal can be demodulated (recovered) from the sampled waveform by passing it through a low pass filter (LPF). Note that the sampling frequency has equal intervals. Also, the sampling frequency f_s should satisfy the **sampling condition**

$$f_s \geq 2f_m \quad (3-1)$$

where f_m is the maximum frequency component of the analog signal. This condition is sometimes called the **sampling theory**, and sometimes referred to as the Shannon sampling theory. Also, the minimum sampling frequency ($f_s = 2f_m$) is sometimes called the **Nyquist frequency**. If this condition is not respected, the sampled signals overlap in the frequency domain, and it will be difficult to restore the original analog signal by demodulation.

The overlapping of sampled signals in the frequency domain is called the **aliasing error**. The frequency domain of an analog signal $x(t)$ and the output sampled signal (the PAM signal, $x_s(t)$) are shown in figure 3-5..

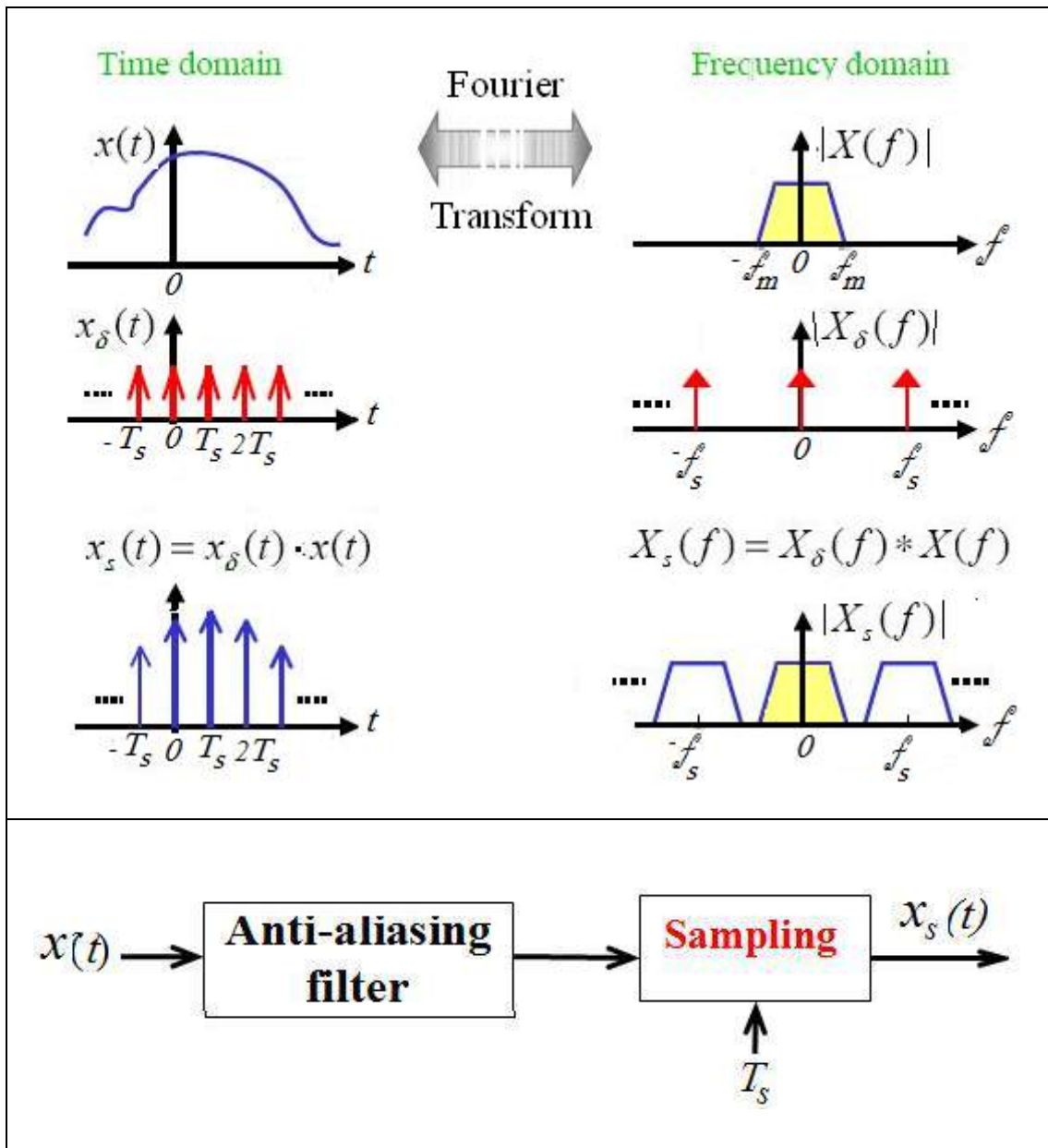


Fig. 3-4. Time- and frequency-domain representation of sampled signals in pulse amplitude modulation

Note that the input analog signal should be band limited (by passing it through a low-pass filter, called **anti-aliasing filter**) before sampling, such that its maximum frequency f_m component meets the sampling condition

In order to avoid aliasing errors, the anti-aliasing filter should be placed before the sampler, as shown in the following figure.

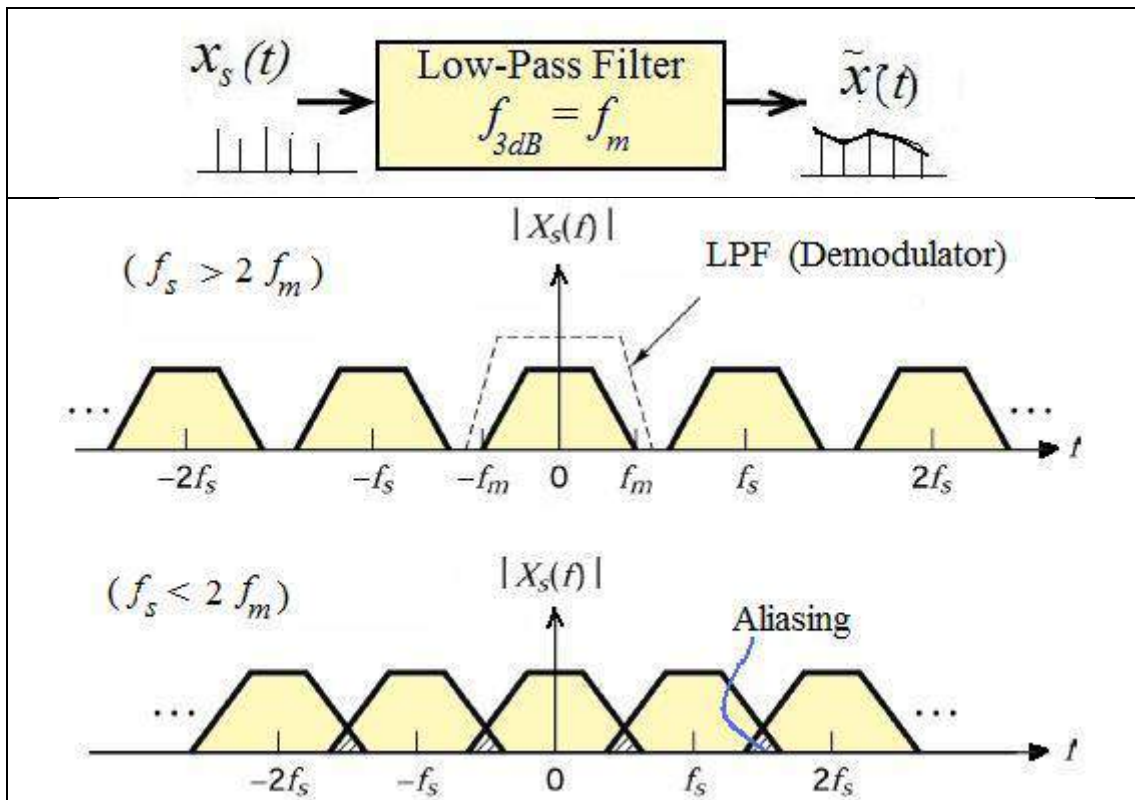


Fig. 3-5. Recovery of sampled data by a low pass filter, when $f_s > 2 f_m$ (no aliasing).

As shown in figure 3-3, the sampling signal, x_δ , can be expressed as an infinite sum of impulses, separated by the sampling period T_s , as follows:

$$x_\delta(t) = \sum \delta(t - nT_s) \quad -\infty < n < \infty \quad (3-2a)$$

Then the output sampled signal $x_s(t)$ is given by:

$$x_s(t) = x(t).x_\delta(t) = \sum \delta(t - n T_s).x(n T_s) \quad -\infty < n < \infty \quad (3-2b)$$

The Fourier transform of this sampled signal is hence given by:

$$X_s(f) = f_s \sum X(f - k f_s) \quad -\infty < k < \infty \quad (3-2c)$$

where k is an integer and $X(f)$ is the Fourier transform of the baseband signal $x(t)$. Alternatively, $X_s(f)$ may be expressed, using the discrete Fourier transform (**DFT**), as follows:

$$X_s(f) = \sum x(n T_s).exp[j2n\pi f T_s] \quad -\infty < n < \infty \quad (3-2d)$$

Finally, we may like to include the effect of the anti-aliasing filter, which band-limits $x(t)$ to a maximum frequency f_m . Therefore, the spectrum of $X(f)$ is zero for $|f| > f_m$. In the special case where the sampling frequency $f_s = 2f_m$, then the Fourier transform of the sampled signal becomes:

$$X_s(f) = \sum x(n T_s) \cdot \exp[jn \pi f / f_m] \quad -\infty < n < \infty \quad (3-2e)$$

Exercise 3-1

What happens to the frequency response of the sampled signal, $X_s(f)$, if the sampling signal is not an impulse, but rather a pulse of duration $T_p \ll T_s$?

3-3. Pulse-Width Modulation (PWM)

The Pulse-width modulation, (**PWM**) of a signal involves the modulation of its duty cycle, to either convey information over a communications channel or control the amount of power sent to a load.

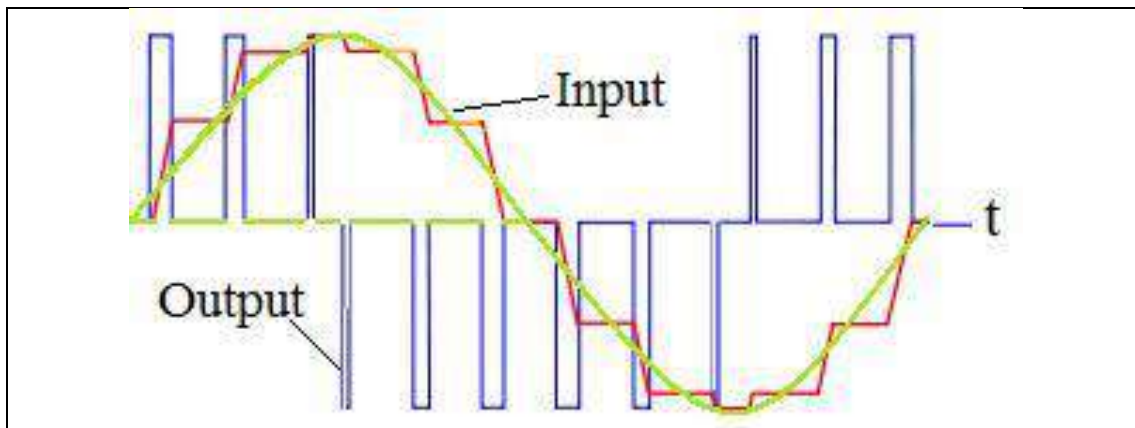


Fig. 3-6. Illustration of the pulse amplitude modulation

In the PWM all the pulses have the same magnitudes but their width varies in proportion to amplitude of the sample. PWM may be obtained from the PAM pulse by charging a small capacitor to the peak voltage of each pulse as it comes along, and then discharging the capacitor through a constant current source before the next pulse arrives. Then the capacitor voltage decreases linearly to zero during a time that is proportional to the pulse amplitude. These triangular pulses may then be passed through a Schmitt trigger circuit to square them up and provide constant amplitude pulses of varying width. A disadvantage of PWM modulation is that we're using a lot of power when the pulses are long. One way to get around this problem is with Pulse Position Modulation (**PPM**) in which just a narrow pulse is transmitted at the end of each PWM pulse.

3-4. Pulse-Position Modulation (PPM)

Pulse-position modulation, acronym PPM, is a form of signal modulation in which M message bits are encoded by transmitting a single pulse in one of 2^M possible time-shifts. This is repeated every T seconds, such that the transmitted bit rate is M/T bits per second. It is primarily useful for optical communications systems, where there tends to be little or no multipath interference.

PPM may then be obtained by differentiating the PWM signals to produce narrow pulses at both the leading and trailing edges of the PWM pulses. The leading-edge pulse is used as the reference pulse, and the time difference between it and the trailing-edge pulse represents the modulation amplitude.

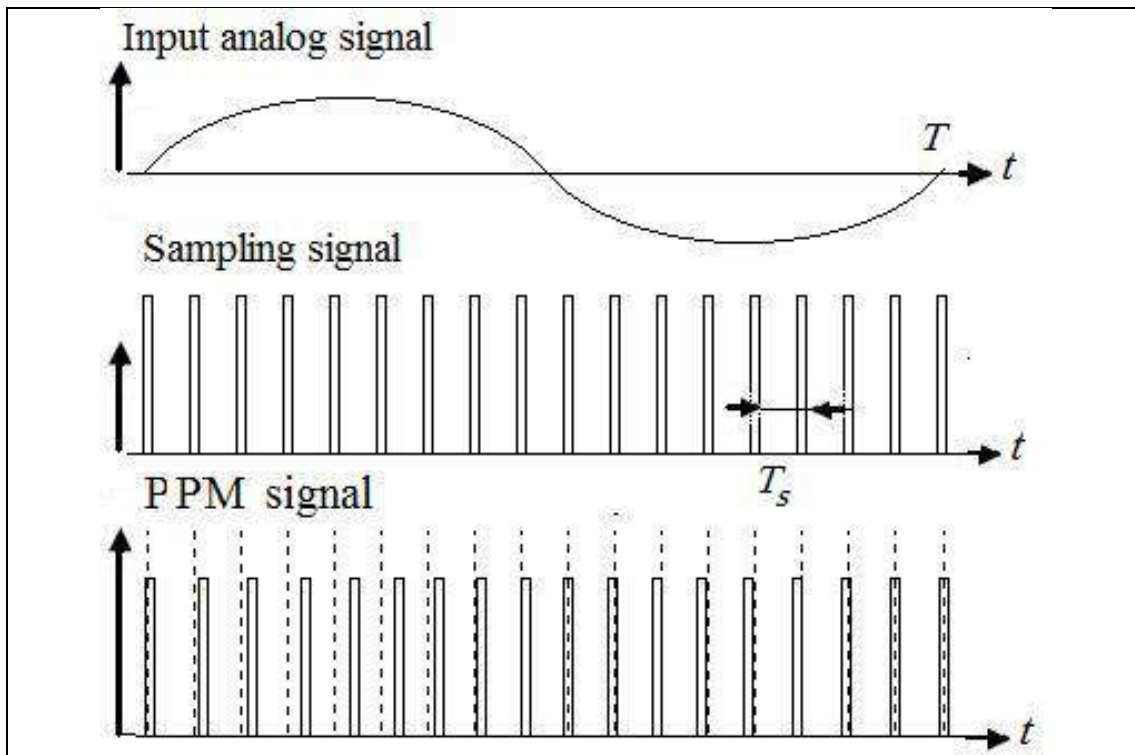


Fig. 3-7. Illustration of pulse position modulation

One of the principal advantages of PPM is that it is an M -ary modulation technique that can be implemented non-coherently, such that the receiver does not need to use a Phase-locked loop (PLL) to track the phase of the carrier. However, both the PWM and PPM signals are also subject to noise so their use may or may not improve the signal-to-noise ratio with respect to the PAM systems.

Note 3-1. History of Pulse Modulation

The earliest reason for sampling a signal was to interlace samples from different telegraphy sources, and convey them over a single telegraph cable. In 1903, the electrical engineer W.M. Miner used an electro-mechanical commutator for time-division multiplex of multiple telegraph signals, and applied this technology to telephony. He obtained intelligible speech from channels sampled at a rate above 3500–4300 Hz. This was actually time-division multiplexing (TDM), but of pulse-amplitude modulation (PAM) rather than PCM.

PAM was a satisfactory method of communication when transmission distances are short to maintain high signal-to-noise ratios (SNR). However, when transmission distances become long, the noise from lightning and other sources that is induced into wire lines or in radio antenna may degrade the SNR and render the message unintelligible.

The British engineer Alec Reeves, unaware of previous work, conceived the use of PCM for voice communication in 1937 while working for International Telephone and Telegraph in France. He described the theory, but no practical use resulted. The first transmission of speech by digital techniques was carried out by the Allied forces during World War II. In 1943, the Bell Labs researchers became aware of the use of PCM coding as already proposed by Alec Reeves. In 1949, Ferranti Canada built a working PCM radio system for the Canadian Navy that was able to transmit digitized radar data over long distance.

3-5. Pulse-Code Modulation (PCM)

Pulse-code modulation (PCM) is a digital representation method of analog signals. In PCM, the magnitude of the input analog signal is sampled regularly at uniform intervals, then quantized to a series of symbols in a numeric (usually binary) code. PCM has been used in digital telephone systems and musical keyboards. It is also the standard form for digital audio in computers and the compact disc (CD) format. It is also standard in digital video. However, uncompressed PCM is not typically used for video in standard definition consumer applications such as DVD because the bit rate required is far too high.

3-5.1. Pulse Code Modulation Process

Figure 3-8 illustrates the steps-by-step procedures, which are followed in the pulse code modulation. As shown in figure, the pulse code modulation

consists in three basic steps:

- 1- **Sampling** the input analog signal. This is done by multiplying the input signal by a periodic pulse train. The sampling frequency should satisfy the sampling condition (Nyquist rate) $f_s \geq 2 f_m$, where f_m is the maximum frequency component of the input analog signal.
- 3- **Quantization** of the sampled signal, which consists in clipping the samples at specific discrete levels, and
- 3- **Encoding** each sample into a specific number of equivalent bits.

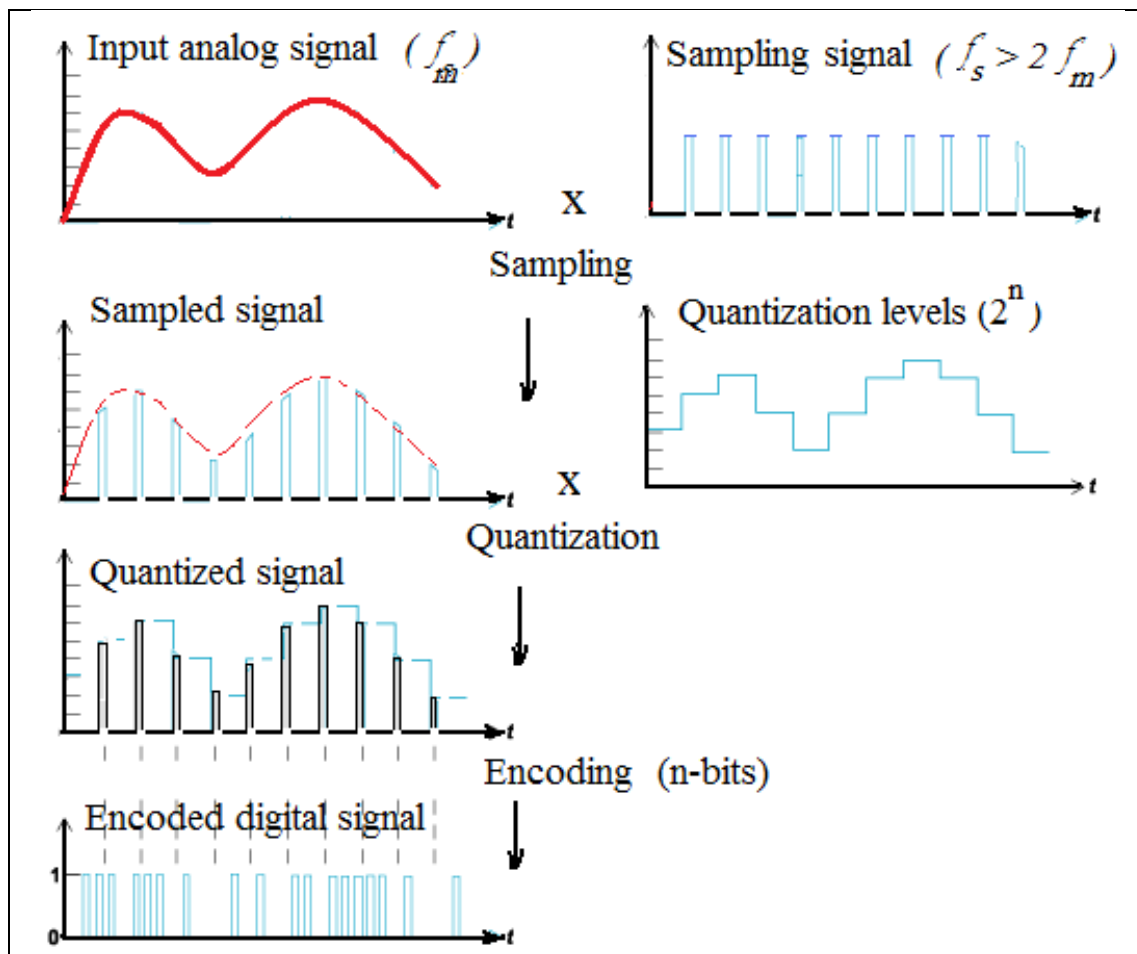


Fig. 3-8. Illustration of pulse code modulation (PCM) technique.

Therefore, the analog signal is sampled at regular intervals, with sampling frequency $f_s \geq 2 f_m$, where f_m is the maximum frequency component of the analog signal. The subsequent quantization process consists in dividing the continuous range of input signal into N discrete levels. The quantization levels are separated by a step q , which is given by the following relation (for positive signals):

$$q = V_m / (N-1) \approx V_m / N \quad (3-3a)$$

where V_m is the peak voltage of the input signal and N is the number of quantization levels. For bipolar waveforms, the quantization step is given by $q = V_{pp}/(N-2)$, where V_{pp} is the peak-to-peak voltage of the input waveform¹. For each sample, one of the nearest quantization values is chosen by a certain rounding algorithm. The following figure illustrates two possible transfer characteristics of a uniform quantizer.

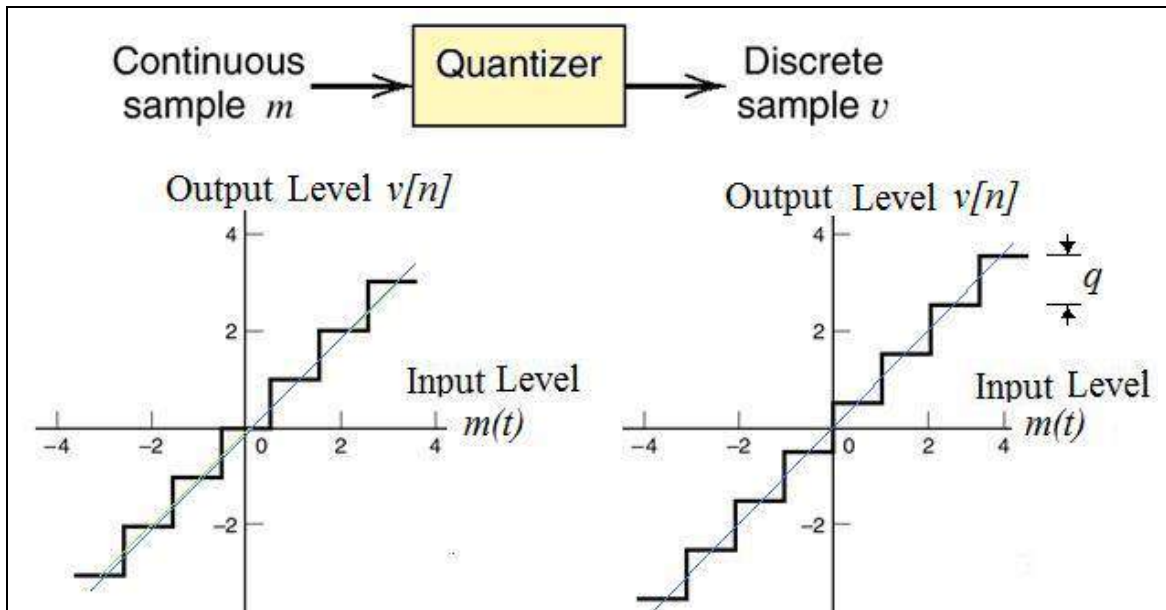


Fig. 3-9(a). Two possible transfer characteristics of a uniform quantizer.

Notice that the first transfer function gives two values for zero-amplitude. In standard PCM, there is a *positive-zero* and *negative-zero*. This produces a discrete (quantized) version of the input signal that can be easily encoded as digital data (bits) for storage or manipulation. If the number of quantization levels is N , then the number of bits/sample $n = \log_2 N$, such that:

$$N = 2^n \quad (3-3b)$$

For instances, if the peak-to-peak voltage is $V_{pp} = 1\text{V}$ and $N = 256$ then the quantization step $q \approx V_{pp}/N \approx 4\text{mV}$ and $n = \log_2 N = 8$. Encoding the quantized samples as binary numbers will result in bit patterns like: 11111111, 10110100, etc. Several PCM streams may be multiplexed into larger data streams, for transmission of multiple streams over a single link. This technique is called time-division multiplexing (**TDM**), and is widely used, notably in the modern public telephone system.

¹ In standard PCM there exist two ground levels, $\frac{1}{2}N-1$ positive levels and $\frac{1}{2}N-1$ negative levels.

3-5.2. Quantization Errors

There are two sources of errors in any PCM system: Choosing a discrete value near the analog signal for each sample results in a quantization error. The quantization error swings between $-1/2q$ to $+1/2q$. In the ideal case, the uniform quantization error is uniformly distributed over this interval such that the error distribution function $\rho(x) = 1/q$ and its mean value is equal to zero:

$$\underline{x} = \int x \cdot \rho(x) dx = 0 \quad (3-4a)$$

The squared variance is given by:

$$\sigma_x^2 = \int (x - \underline{x})^2 \cdot \rho(x) dx = q^2 / 12 \quad (3-4b)$$

Here, the integration is taken on the period between $(-1/2q, +1/2q)$. The root mean square quantization error (ϵ_{rms}) is hence given by:

$$\epsilon_{rms} \text{ (quantization error)} = q / \sqrt{12} \quad (3-5)$$

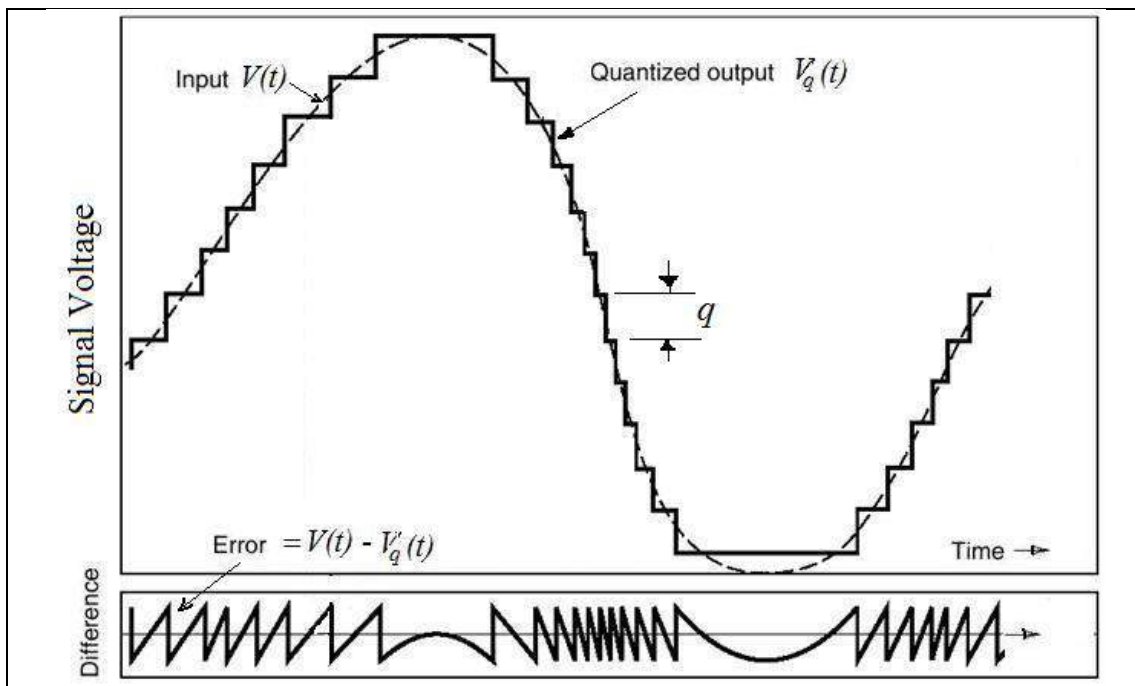


Fig. 3-9(b). Illustration of the quantization noise.

Note that, any frequency above or equal to half the sampling frequency ($1/2 f_s$) will result in distorted signal in the demodulator side. Assuming a signal with peak voltage $V_m = 1/2 V_{pp} = 1/2 qN$, then, the signal-to-quantization-noise ratio (SNR), after uniform quantization, is given by:

$$SNR = 1/2 (1/2 qN)^2 / (q^2/12) = 3/2 N^2 \quad (3-6)$$

Therefore, the SNR ratio increases as the number of quantization levels increases. However, this means more bits per sample. In order to improve the SNR, without excessive increase in the number of quantization levels, we may use a non-uniform quantizer. Alternatively, the input signal can be **compressed** with a nonlinear (logarithmic) function, before quantization. Of course, the signal should be expanded again with inverse function in the demodulator side.

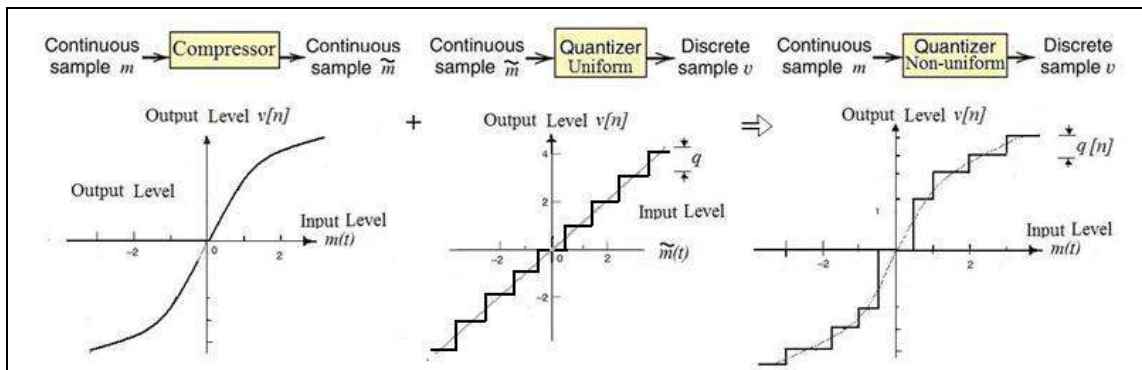


Fig. 3-9(c). Illustration of the non-uniform quantization process, using compressor before a uniform quantizer.

3-5.3. Encoding

The last step of PCM is encoding the quantized samples into equivalent code of binary digits (bits). As we mentioned so far, the number code bits $n = \log_2 N$, where N is the number of quantization levels. For instance, if we have 8 quantization levels, each sample is coded into 3 ($\log_2 8$) bits for each sample, as shown in the figure 3-10.

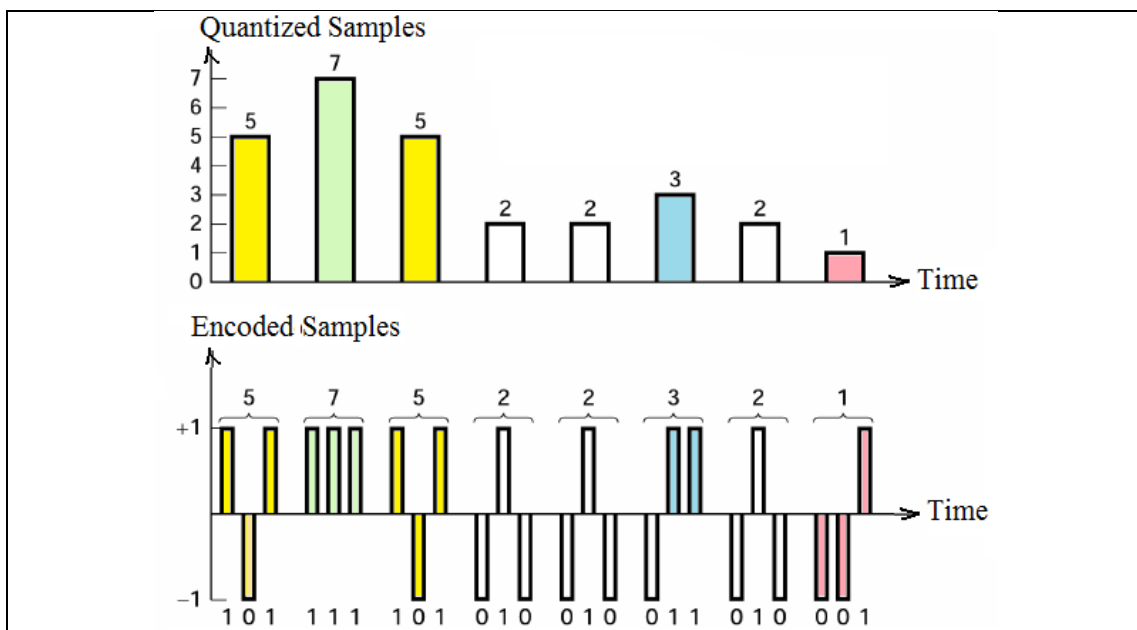


Fig. 3-10. Example of PCM with 8 quantization levels and 3bits per sample.

The Pulse-code modulation can be encoded by various methods such as return-to-zero (**RZ**) or non-return-to-zero (**NRZ**) sequences. The following figure depicts the various PC encoding methods. These codes are called line codes, and they are described in details in Chapter 3. Sometimes, the long term DC value of the PCM signal may build up a DC offset which tends to bias detector circuits out of their operating range. In this case special measures are taken to keep a count of the cumulative DC offset, and to modify the codes if necessary to make the DC offset tend back to zero. Many of these codes are bipolar codes, where pulses can be positive, negative or absent.

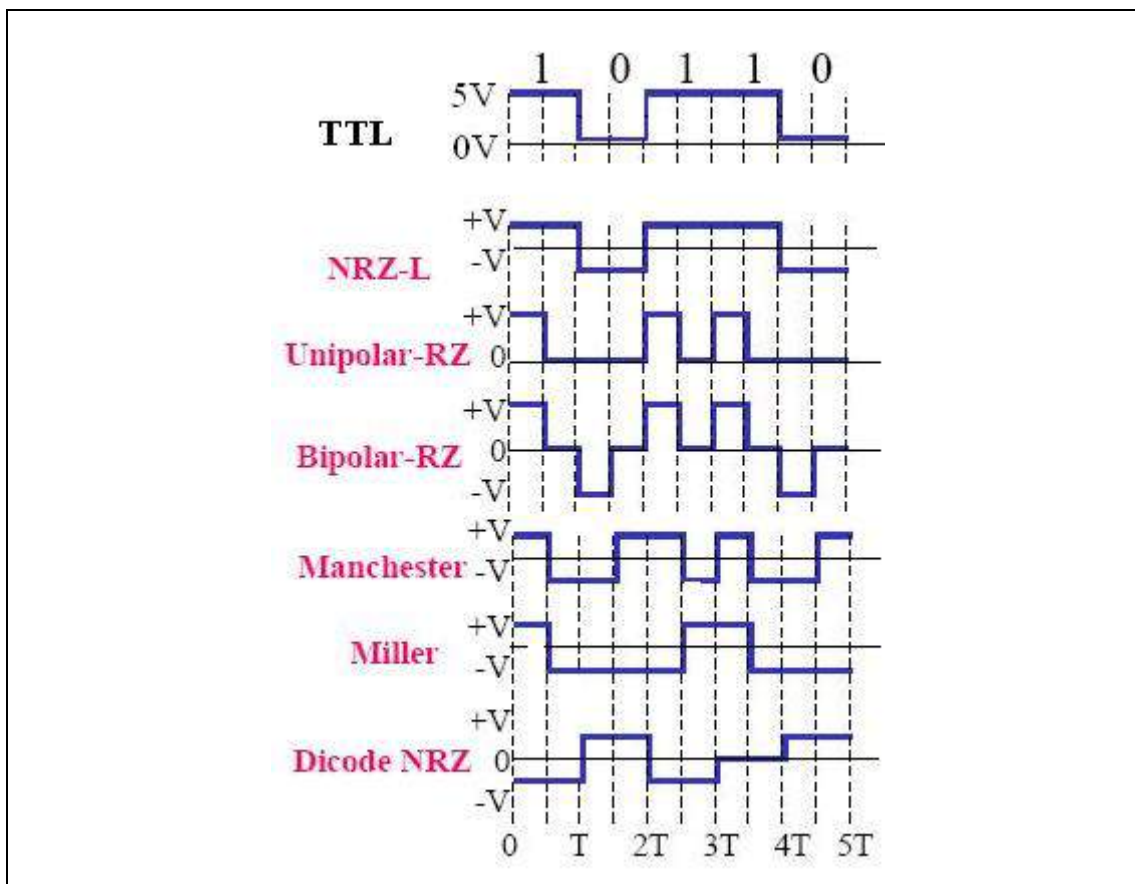


Fig. 3-11. Various encoding methods for PCM binary waveforms.

Example 3-1.

If a PCM system is encoded with 8-bits, we have 256 possible quantization levels. We may assign these 256 possible values as 127 positive and 127 negative encoding levels, plus the zero-amplitude levels. Actually, the standard PCM assigns two samples to the zero level. These levels can be divided into eight bands called **chords**. Within each chord there are sixteen **steps**. The following figure shows the chord/step structure for a linear inverted encoding scheme.

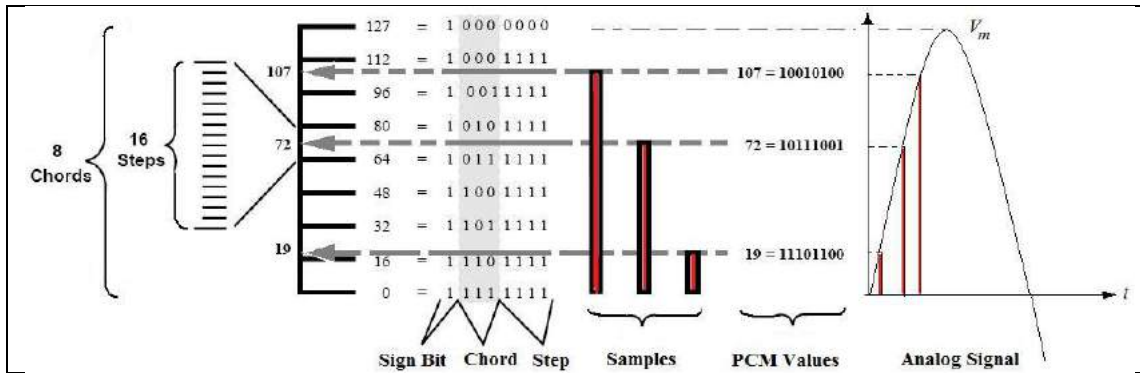


Fig. 3-12. Example of a PCM encoding.

3-5.4. PCM Demodulation

In order to reproduce the original analog signal from the sampled and encoded data, the procedure of modulation is applied in reverse. After each sampling period, the next value is read and the output of the system is shifted to the new value. As a result of these transitions, the discrete signal will have a significant amount of inherent high frequency energy, mostly harmonics of the sampling frequency. In order to smooth out the signal and remove the undesirable harmonics, the signal is passed through analog low-pass filters (LPF) that suppress artifacts outside the expected frequency range (*i.e.*, greater than $\frac{1}{2}f_n$).

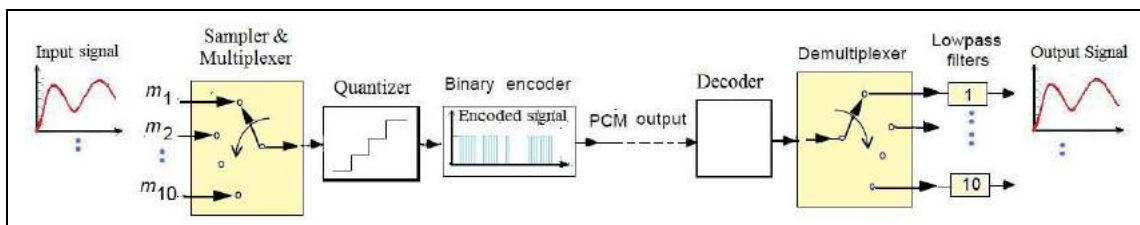


Fig. 3-13. Block diagram of a PCM transmitter/receiver (transceiver) system

The main reason for using PCM for digital baseband transmission is that several signals may be multiplexed and transmitted over a single channel. The simplest technique for multiplexing several signals onto a single channel is to use a rotating multipole switch known as a commutator. The commutator may serve as the sampler, or modulator, as well as the multiplexer. A commutator must be used on the receiving end as well as the transmitting end of the multiplexed channel to unscramble the signals. The commutators must be synchronized so that each message is sent to its proper destination at the receiving end. For example, several telephone conversations might be sampled and multiplexed onto a single telephone line. As shown in figure 3-13, a low-pass filter is used in each line at the

receiving end to pass only the desired message frequencies and remove the higher frequencies of the pulses.

Example 3-2.

Assume an analog signal, which is sampled in 11 points and quantized using 8 quantization levels, as shown in figure 3-14(a). All the values that fall into a specific quantization interval are approximated to a quantization level which lies in the middle of the interval. The samples shown in figure are already quantized - to the nearest quantization level (written to the right of each sample).

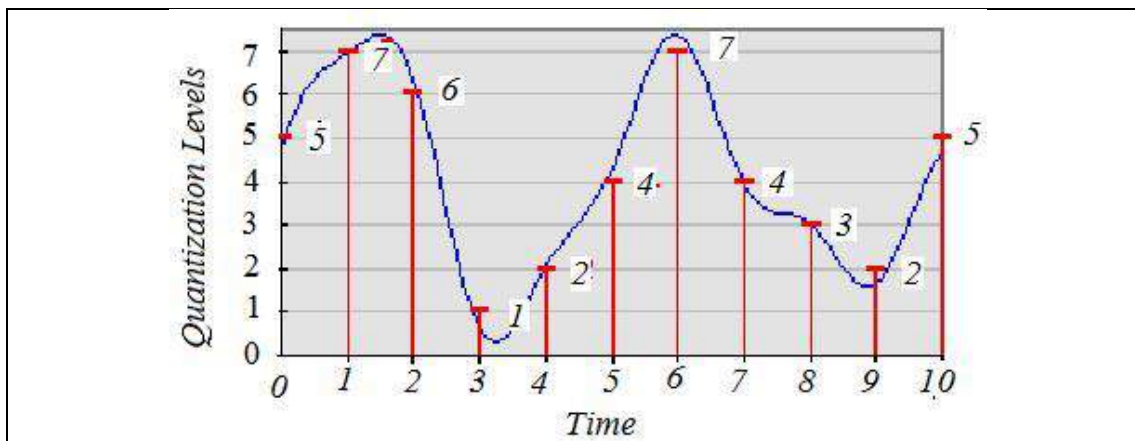


Fig. 3-14(a). Example of PCM. (a) Input analog signal and its digitized samples

The eight quantization levels are encoded into 3-bits as follows:

Level	Codeword	Level	Codeword
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

The PCM encoded signal in binary form is as follows with total of **33** bits

101 111 110 001 010 100 111 100 011 010 101

The following figure shows the process of signal restoration (demodulation), according to the quantized samples. Note that the restored signal is different from the input signal. This difference is a consequence of the quantization noise.

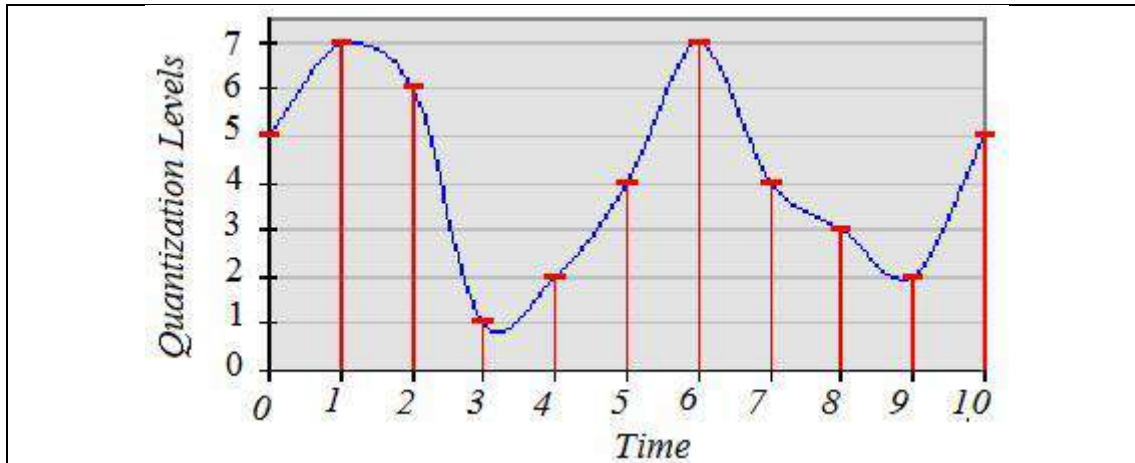


Fig. 3-14(b). Restored samples and output analog signal

2.5.5. Companding

The compression of the input signal before quantization is equivalent to using **non-uniform quantization**, such that smaller steps are used for weak signals and large steps for strong ones. The compression may be done after quantization where a 12 or 13 bit linear PCM sample number is mapped into an 8-bit value. The inverse expanding process is done on the PCM demodulator side.

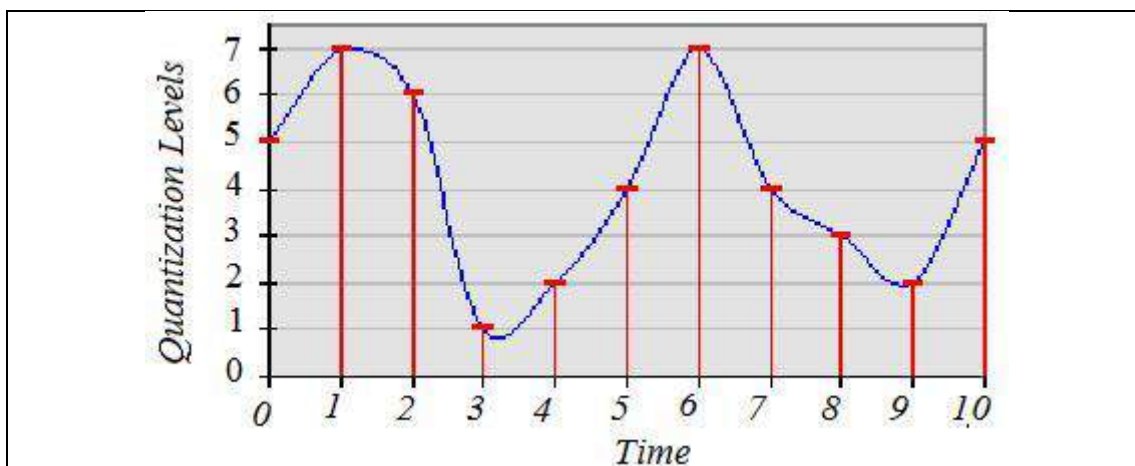


Fig. 3-14(c). Restored samples and output analog signal

The compression-expanding (**companding**) system is described by international standard G.711. In telephony, a standard audio signal for a single phone call is encoded as 8000 analog samples/s, each sample of 8 bits, giving a bit rate of 64 kbit/s. This digital signal is known as DS0. The DS0 signal is usually subjected to compression encoding by either a **μ -law** (North America and Japan) or **A-law** (Europe and most of the rest of the world). These compression laws are logarithmic, as shown in figure 3-15.

The μ -law for compression may be expressed as follows:

$$y(x) = \frac{\ln(1 + \mu x)}{\ln(1 + \mu)}, \quad 0 \leq x \leq 1 \quad (3-7)$$

where μ is a constant. Also, the A-law for compression may be expressed as follows:

$$y(x) = \frac{Ax}{1 + \ln(A)}, \quad 0 \leq x \leq \frac{1}{A}$$

$$y(x) = \frac{1 + \ln(Ax)}{1 + \ln(A)}, \quad \frac{1}{A} \leq x \leq 1 \quad (3-8)$$

where A is a constant (usually taken as 100).

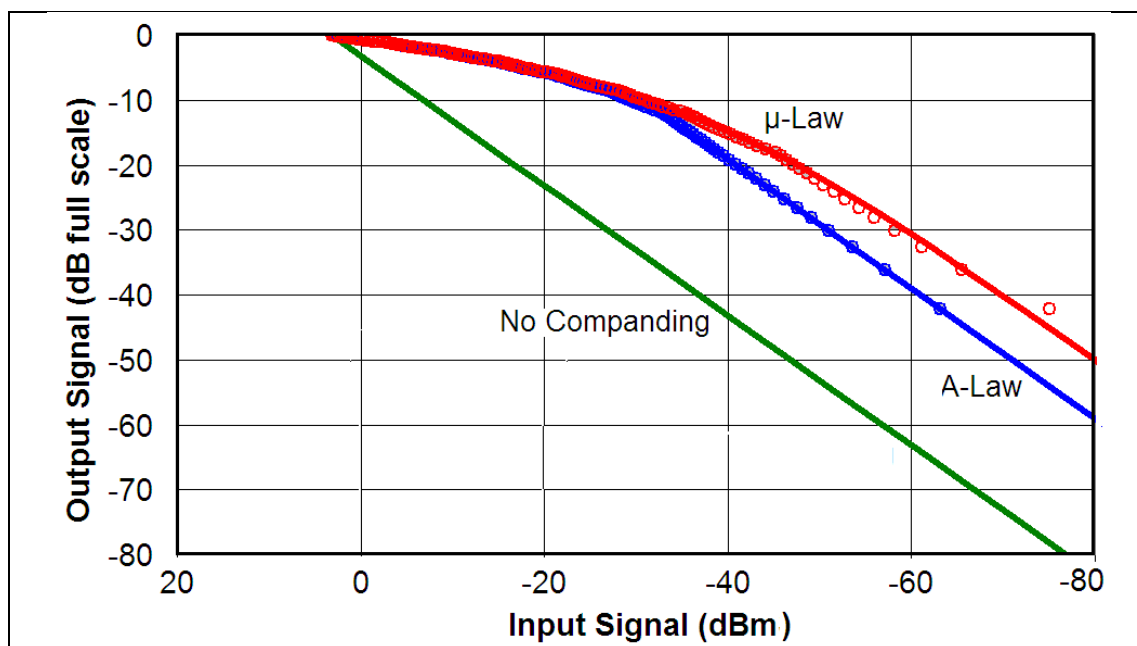


Fig. 3-15. Data compression-expanding (companding) laws, for PCM signals.

Figure 3-15 shows the PCM transceiver block diagram with companding. The following figure shows the output SNR of a PCM quantizer, without and with (μ -law) companding. It can be shown that the companding process improves the output SNR in PCM. The A-law compressor has a greater dynamic range than the μ -law compressor, and has a smaller output SNR than the μ -law compressor.

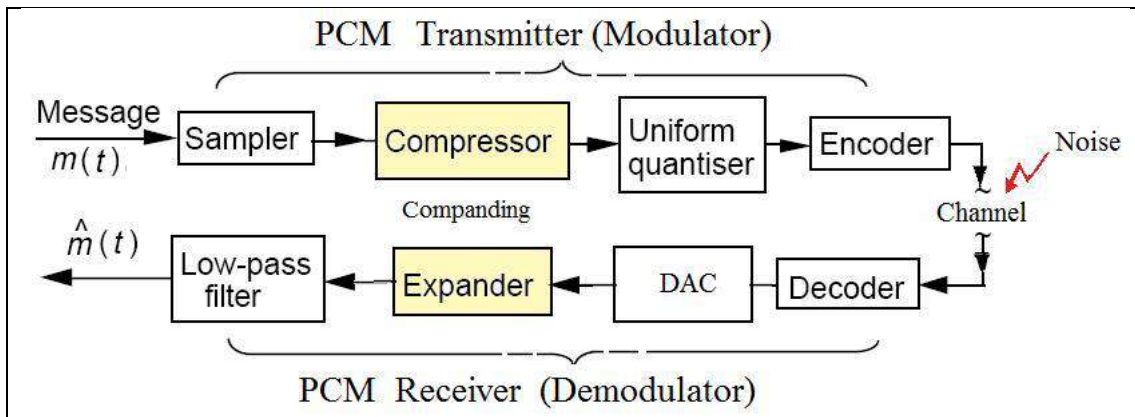


Fig. 3-16. Block diagram of a PCM transceiver system, with companding

3-5.6. Applications of PCM

PCM is the primary way analog waves are converted into digital form for voice conversations as well as music. The PCM was initially introduced when the telephone companies began converting voice to digital for transport over long (noisy) trunk lines. Also, PCM is used in voice reordering and computers sound cards. The microphone circuits, on a PC sound card, generate PCM. Compressed audio formats such as MP3 and AAC are converted to PCM first, and the sound card converts the PCM to analog for the speakers.

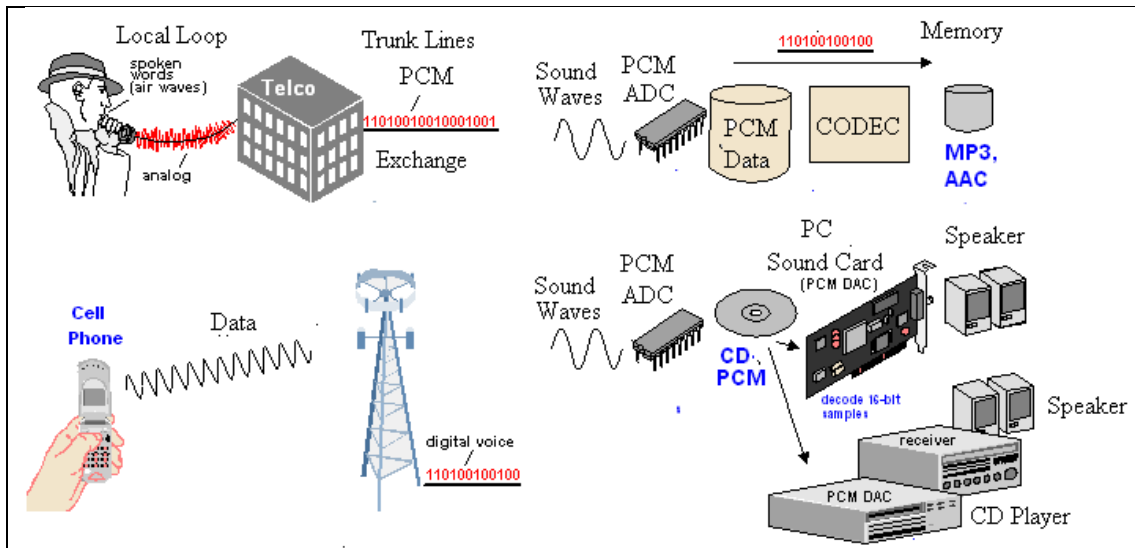


Fig. 3-17. Illustration of some applications of PCM

2.5.7. Other Forms of PCM

In conventional PCM, the analog signal may be processed (e.g. by compression) before being digitized. Once the signal is digitized, the PCM signal is usually subjected to further processing. Some forms of PCM

combine signal processing with coding. These simple techniques have been largely rendered obsolete by modern transform-based audio compression techniques.

i- Differential PCM (**DPCM**) encodes the PCM values as differences between the current and the **predicted** value. An algorithm predicts the next sample based on the previous samples, and the encoder stores only the difference between this prediction and the actual value. If the prediction is reasonable, fewer bits can be used to represent the same information. For audio, this type of encoding reduces the number of bits required per sample by about 25% compared to PCM.

ii- Adaptive DPCM (**ADPCM**) is a variant of DPCM that varies the size of the quantization step to allow further reduction of the required bandwidth for a given signal-to-noise ratio. **ADPCM** techniques are used in Voice over Internet protocol (**VoIP**) communications. The ADPCM algorithm is used in telephone transmission to map the series of 8-bit μ -law or A-law PCM samples into a series of 4-bit ADPCM samples. In this way, the capacity of the telephone line is doubled.

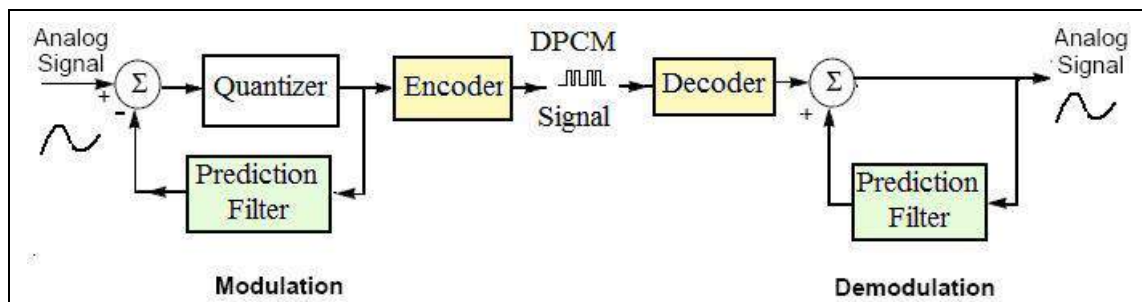


Fig. 3-18. Simple block diagram of a DPCM modulator and demodulator

iii- Delta Modulation is another analog-to-digital conversion variant, which uses one bit per sample. Delta modulation is used for voice transmission when quality is not as important as the transmission itself. In this particular type of transmission the data is converted to a 1-bit data stream. In order to improve the quality of the transmissions, oversampling can be performed, thus making the conversion an economical and practical communication method. The delta modulation and subsequent delta-sigma techniques can be used to transmit narrow-band signals such as audio signals. As shown in figure 3-19, the delta modulator is as simple as a comparator, followed by a 1-bit quantizer. The quantizer output toggles, according to the polarity of the input signal. The delta demodulator is a simple integrator followed by a low-pass filter (LPF), as shown in figure.

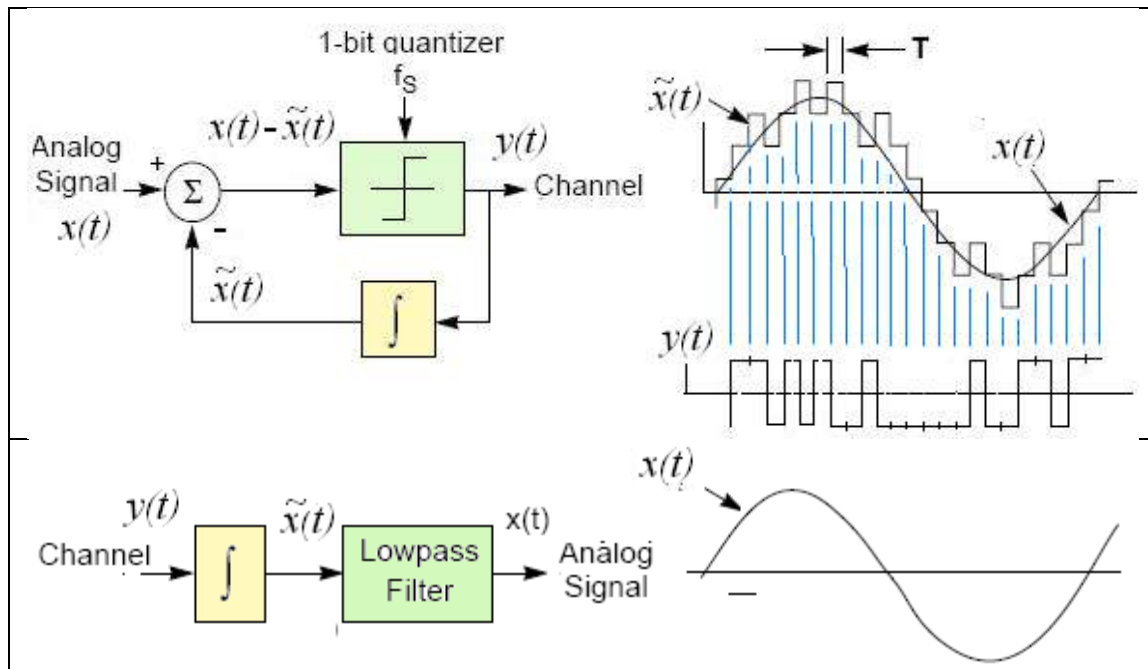


Fig. 3-19. Illustration of a delta modulator and demodulator and their waveforms.

3-6. Sigma-Delta Modulation

The sigma-delta (Σ - Δ) modulator² is a modified version of the delta modulator. The configuration of the Σ - Δ modulator was initially introduced by Inose et al in 1962 and since then it has been a subject to continuous improvements. However, the Σ - Δ modulation technique did not gain importance until recent developments in VLSI technology. The increasing use of digital techniques in communication and audio application has also contributed to the recent interest in cost effective high precision A/D converters, like the Σ - Δ demodulator. The following figure depicts delta-sigma (Σ - Δ) modulator and demodulator. The principle of the Δ - Σ modulator is to make rough evaluations of the signal to measure the error integrate it and then compensate for that error.

The output mean value is then equal to the input mean value if the integral of the error is finite.

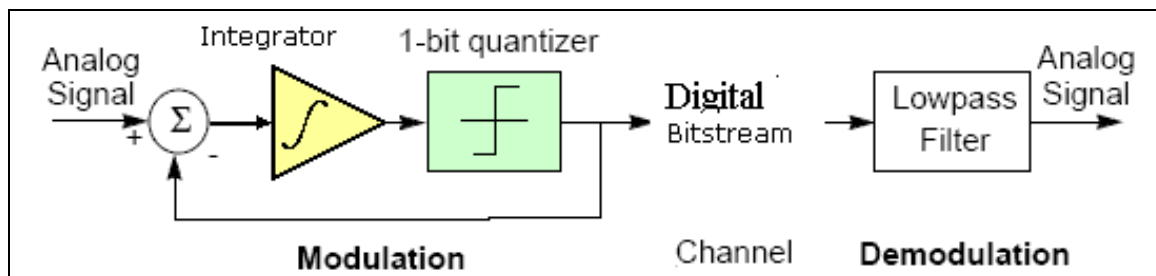


Fig. 3-20. Simple block diagram of a 1st order Δ - Σ modulator and demodulator

² Sometimes called Delta-Sigma (Δ - Σ) modulator.

3-6.1. Implementation of a Delta-Sigma Modulator

For practical reasons, the quantizer of the Σ - Δ modulator, which acts as 1-bit ADC is implemented using a D-type flip-flop (DFF) triggered by a clock, as shown in figure 3-21.

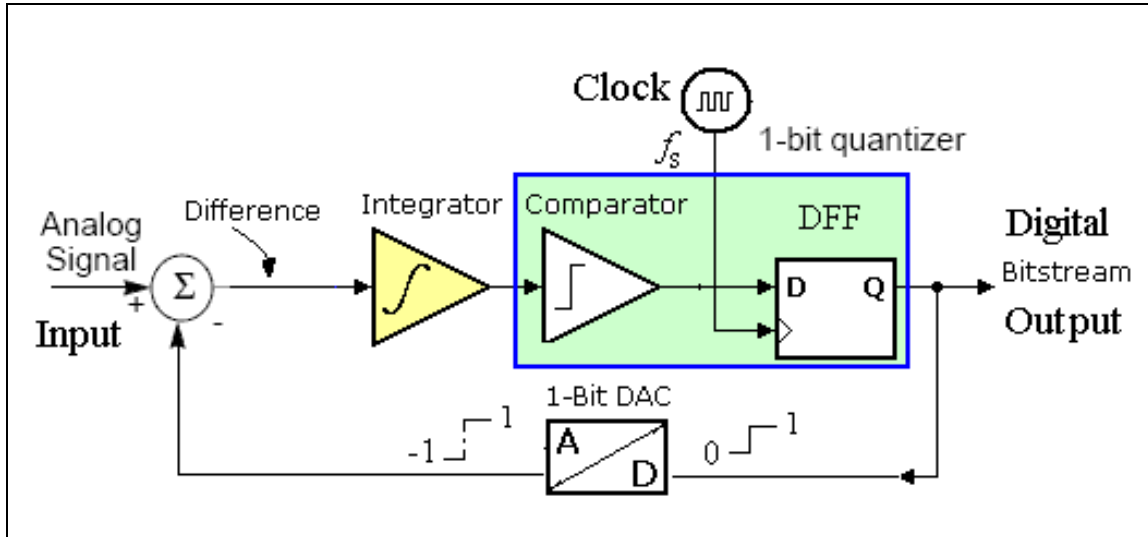


Fig. 3-21(a). Practical block diagram of a 1st order Δ - Σ modulator

The clock frequency of Σ - Δ modulator should satisfy the sampling condition ($f_s \geq 2f_m$). The modulator output (from the DFF) has 1-bit. Also, a 1-bit DAC is put in the feedback path of the modulator. to convert the DFF binary output into bipolar form. i.e.. V_{ref} or $-V_{ref}$.

3-6.2. Quantization Errors in Σ - Δ Modulation

When a signal is quantized, the resulting signal approximately has the second-order statistics of a signal with independent additive white noise. The quantization noise of a modulator has two forms, namely: the granular error and the slope overload error, as shown in figure 3-22. Assuming that the signal value is in the range of one step of the quantization intervals, with an equal distribution, the root mean square value of the quantization noise³ is given by:

$$e_{rms} (\text{Granular error}) = q / \sqrt{12} \quad (3-9)$$

where q is the quantization step size, as shown in figure 3-22.

³ Also called the **granular quantization noise**

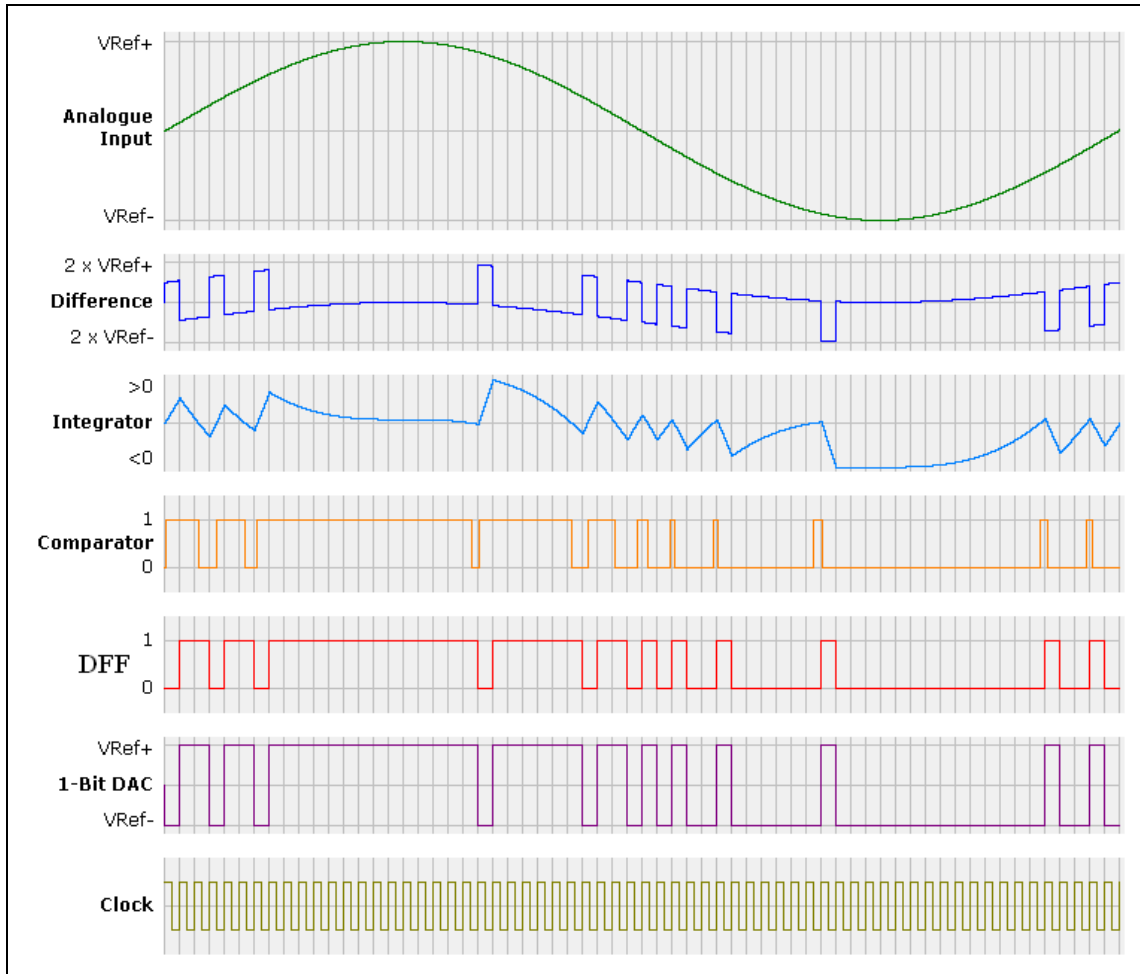


Fig. 3-21(b). Practical waveforms of a 1st order Δ - Σ modulator.

In reality, the quantization noise is not independent of the signal value and waveform. For example, the signal speed maybe too high (when $|dx/dt| > q/T_s$), such that the quantized waveform cannot follow the signal, as shown in figure 3-22. This error is called the **slope overload** distortion.

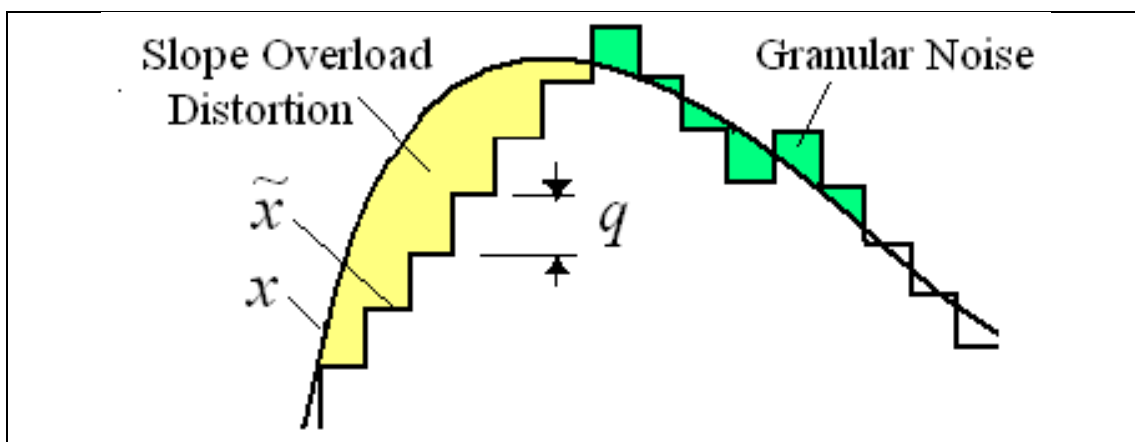


Fig. 3-22. Granular quantization errors of the Δ - Σ modulator

In order to reduce the quantization errors, oversampling is used. The oversampling ratio, OSR, is given by:

$$OSR = f_s / f_o = 1/2 (f_s / f_m) \quad (3-10)$$

where f_s is the sampling frequency and $f_o = 2f_m$ is the Nyquist rate. The RMS value of the quantization noise voltage within the band of interest can be expressed in terms of OSR as follows:

$$v_n = e_{rms} / \sqrt{OSR} \quad (3-11)$$

Note that the noise voltage is inversely proportional to the oversampling ratio OSR. For this reason, the Δ - Σ modulation is usually exploiting the oversampling technique to reduce the noise in the band of interest and to avoid the need to high-cost **anti-aliasing filter** in before the input signal.

Example 3-3.

For the matter of illustration, consider the case of an analog signal with maximum frequency $f_m = 24$ kHz. A sampling frequency $f_s = 48$ kHz allows such a signal to pass without aliasing, but because of practical circuit limitation, the corner frequency is actually about 22 kHz. The anti-aliasing filter in a Nyquist ADC requires a flat response with no phase distortion over the frequency band of interest (about 20 kHz in digital audio). In order to prevent signal distortion due to aliasing, all signals above 24 kHz for a 48 kHz sampling rate should be attenuated by at least 96 dB for a 16 bits of dynamic resolution converter. These requirements are tough to meet with analog LPF. Figure 3-23 depicts the waveforms.

Figure 3-23(a) shows the required analog anti-aliasing filter response, while Fig. 3-23(d) shows the digital domain frequency spectrum of the signal being sampled at 48 kHz. Now consider the same signal sampled at $2f_s$, 96 kHz. The anti-aliasing filter only needs to eliminate signals above 74 kHz, while the filter has flat response up to 22 kHz. This is a much easier filter to build because the transition band can be 52 kHz (22k to 74 kHz) to reach the -96 dB point. However, since the final sampling rate is 48 kHz a sample rate reduction filter commonly called a **decimation filter** is required. This is implemented in the digital domain, as opposed to the anti-aliasing filter which is implemented with analog circuits. Figure 3-23(d) and Figure 3-23(e) illustrate the anti-aliasing filter requirement and its frequency response. The spectrum of the required decimation filter is shown in Figure 3-23(f). Other details about the decimation process are illustrated in a following section (§3-6.4).

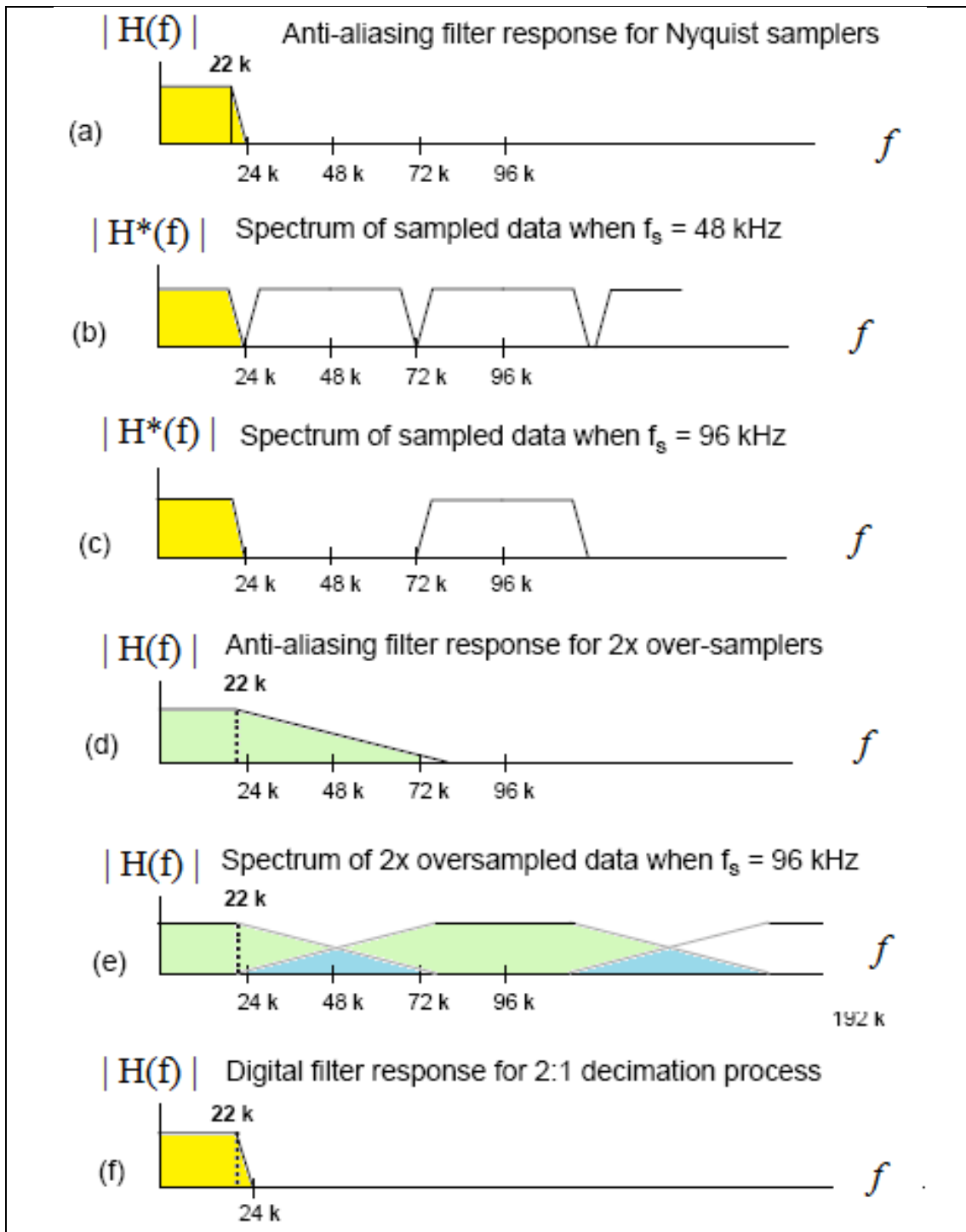


Fig. 3-23. Waveforms of an anti-aliasing and decimation filters.

3-6.3. Signal-to-Noise Ratio of Σ - Δ Modulators

The signal to quantization noise ratio (SQNR or SNR in dB) of an n -bit Σ - Δ converter (without oversampling) is given by the following relation:

$$SNR = 10 \log (\langle V^2 \rangle / \langle e^2 \rangle) = 1.76 + 6.02n \quad (3-12a)$$

However, with oversampling, the error is reduced, according to equation (3-11) and the SNR (in dB) is given by:

$$SNR = 10 \log(\langle V^2 \rangle / \langle v_n^2 \rangle) = 1.76 + 6.02n + 10 \log(f_s/f_o) \quad (3-12b)$$

Therefore, when the sampling frequency is doubled, the signal to quantization noise is improved by $10 \log(2^{2n+1})$ dB for an n^{th} order Δ - Σ modulator. The higher the oversampling ratio (OSR), the higher the signal-to-noise ratio (SNR) and the higher the resolution in bits. As shown in figure 3-23, the noise power is spread between DC and $1/2 f_s$.

3-6.4. Dynamic Range of Σ - Δ Modulators

Another objective measure of quality of the Δ - Σ converter is the dynamic range (DR). Dynamic range pertains to the resolution of a quantization scheme. It is the ratio of the full scale amplitude to the smallest quantized amplitude change (q).

$$DR = 1/2 (V_{pp}/q) = 1/2 (q 2^n/q) = 2^{n-1} \quad (3-13a)$$

or

$$DR \text{ (dB)} = 6.02 (n-1) \quad (3-13b)$$

3-6.5. Higher-order Δ - Σ Modulators

The number of integrators, and consequently, the numbers of feedback loops, indicates the *order* of a Δ - Σ modulator. A 2nd order Δ - Σ modulator is shown in Fig. 3-25. Delta-sigma modulators of orders higher than 2 are possible to construct but they cannot simply be made by adding further stages as shown below. The reason is that the phase turn caused by more than two integrators will make the system unstable. Low pass filters are used instead. Delta sigma ADCs for audio applications typically use 5th order modulators. The architecture may look different, but the basic principle of operation remains the same.

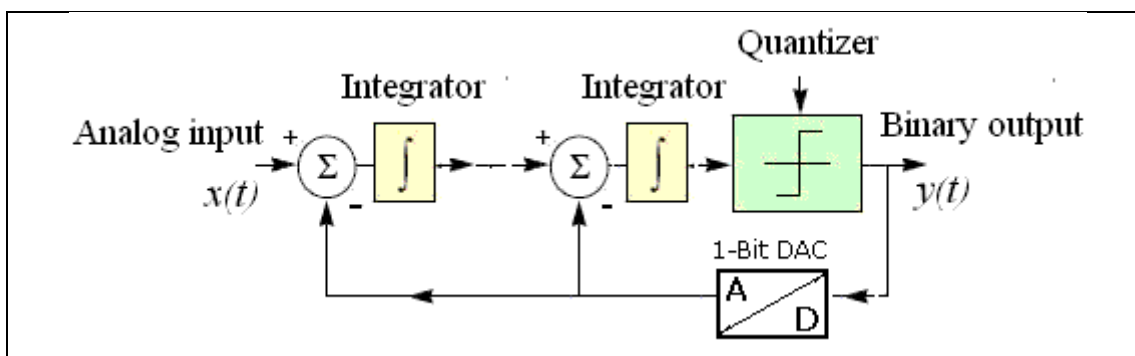


Fig. 3-24. Block diagram of a 2nd order Δ - Σ modulator

The Δ - Σ modulator can also be classified by the number of output bits (n), which depends on the quantizer implementation. If the quantizer is realized with N -level comparator, then the modulator will have $n = \log_2 N$ bit output. For example, a 1-bit modulator has a quantizer realized by a 3-level comparator, whose output is 1 or 0 if the input signal is positive or negative. The noise of a Δ - Σ modulator can be further reduced at low frequencies and filtered at the higher frequencies. This property is known as **noise shaping**

3-7. Multiplexing and Multiple Access Communication Systems

Multiplexing is the combination of several information channels onto a common transmission medium. In electrical communications, the two basic forms of multiplexing are time-division multiplexing (**TDM**) and frequency-division multiplexing (**FDM**). In optical communications, the analog of FDM is referred to as wavelength-division multiplexing (**WDM**). Bit streams can be also transferred over an analog channel by means of code-division multiplexing (**CDM**) techniques such as frequency-hopping spread spectrum (**FHSS**) and direct-sequence spread spectrum (**DSSS**).

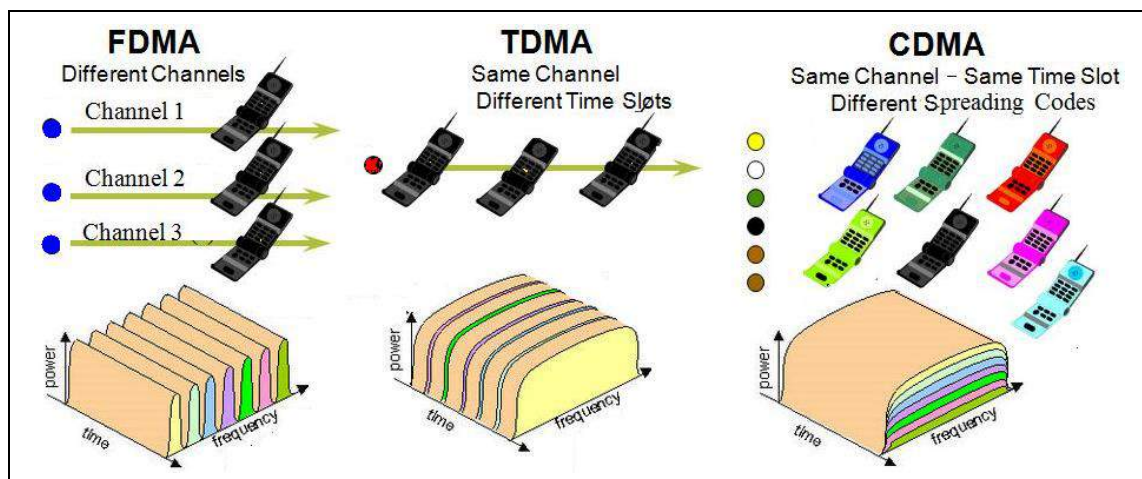


Fig. 3-25. Illustration of the different multiple access schemes

When multiplexing is used as to allow multiple users to share a physical communications link, it is called a multiple access system. For instance, the multi-user time-division multiplexing system is called TDM Multiple access (**TDMA**). Also, the multi-user frequency-division multiplexing system is called FDM Multiple access (**FDMA**). The following figure illustrates the different multiple access schemes.

In a TDMA system, one frequency channel is divided into time slots that are allocated to users, and the users only transmit during their assigned time slots. In a FDMA system, the total system bandwidth is divided into several frequency channels that are allocated to users. In CDMA, each user is assigned a distinct code sequence (spreading code) that is used to encode the user's information-bearing signal. The receiver retrieves the desired signal by using the same code sequence at the reception. Multiple access protocols and control mechanisms are called media access control (**MAC**) for Data links. Each multiple access scheme has its own advantages and disadvantages, which limit its use to specific applications. For instance, the first generation of wireless standards used TDMA and FDMA. In the wireless channels, TDMA proved to be less efficient in handling the high

data rate channels as it requires large guard periods to alleviate the multipath impact. Similarly, FDMA was consuming more bandwidth to avoid inter carrier interference. Therefore, in the second generation (2G) of mobile systems, one set of standard used the combination of FDMA and TDMA and the other set introduced CDMA. Usage of CDMA increased the system capacity and also increased data rate as this access scheme is efficient enough to handle the multipath channel. This enabled the third generation (3G) systems to use CDMA as the access scheme (e.g., CDMA2000 and W-CDMA). The only issue with CDMA is that it suffers from poor spectrum flexibility and scalability. For next generations of mobiles (4G), the orthogonal FDMA (**OFDMA**) is being considered for the downlink. Actually, the **Wi-MAX** technology uses OFDMA in downlink and uplink. The so called multi-carrier code-division multiple-access (**MC-CDMA**) is also gaining more importance for the next generation systems.

3-7.1. Time-Division Multiplexing (TDM)

Time-Division Multiplexing (**TDM**) is a type of digital or (rarely) analog multiplexing in which two or more signals or bit streams are transferred apparently simultaneously as sub-channels in one communication channel, but are physically taking turns on the channel. The time domain is divided into several recurrent **timeslots** of fixed length, one for each sub-channel. A sample byte or data block of sub-channel 1 is transmitted during timeslot 1, sub-channel 2 during timeslot 2, etc. One TDM frame consists of one timeslot per sub-channel. After the last sub-channel the cycle starts all over again with a new frame, starting with the second sample, byte or data block from sub-channel 1, etc.

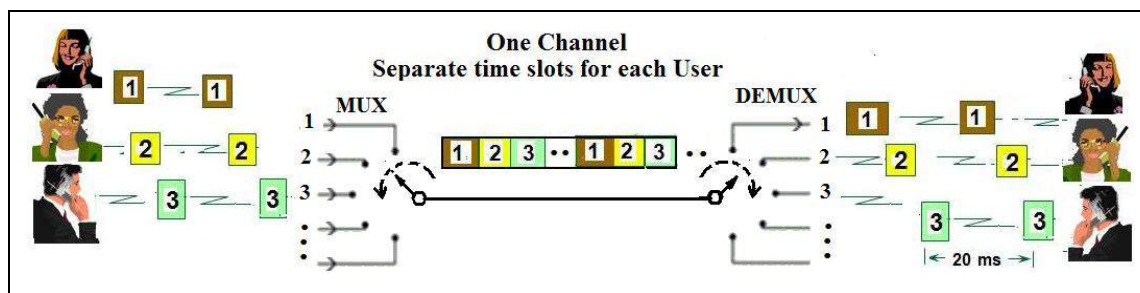


Fig. 3-26. Illustration of the TDMA techniques.

There are a variety of TDM schemes, discussed in the following sections:

- Conventional Time Division Multiplexing
- Statistical Time Division Multiplexing
- ATM Multiplexing / Cell-Relay

i- TDMA Transmission

In circuit switched networks such as the Public Switched Telephone Network (PSTN) there exists a need to transmit multiple subscribers' calls along the same transmission medium. To accomplish this, network designers make use of TDM. TDM takes frames of the voice signals and multiplexes them into a TDM frame which runs at a higher bandwidth. Thus, if the TDM frame consists of n voice frames, and the bandwidth of each channel is 64 kb/s (8k samples/s x 8 bits/sample).

ii- T1 Framing

In American systems, TDM frames contain 24 digital voice frames. and thus operate at 1.544 Mbps (24 x 64 kb/s). The engineers called this data stream DS-1. It has been later known as T1. These framing may be grouped in either Superframe (**D4**) or Extended Superframe (**ESF**). The standard Superframe (**D4**) is 193 bits long (1 Framing bit + 24x8-bit timeslots). Each timeslot is scanned at a rate of 8000 times per second. Therefore, in one second, there are: $8000 \times 8 \text{ bits/TS} \times 24 \text{ TS} = 1.536.000$ Bits of "Payload" data transmitted. There are: $8000 \times 1 = 8.000$ Bits of synchronization bits transmitted within a one second interval. Therefore, the total aggregate rate of the T1 signal is 1.544.000bps (1.544 Mbps). The standard Extended-Superframe Format (**ESF**) frame is 193 bits long (1 Framing bit +24 8-bit timeslots). Each timeslot is scanned at a rate of 8000 times per sec (as in D4/SF). The line rate is 1.544 Mbps wi a data "payload" of 1.536 Mbps.

iii- E1 Framing

In European systems, TDM frames contain 30 digital voice frames and thus operate at 2.048 Mbps (30 x 64 kb/s). The engineers called this data stream as **E1**. The E1 framing is standardized by the CCITT and was adopted by most countries. This framing format is actually defined in CCITT Recommendation G.704. The standard frame has 32 timeslots, with each timeslot consisting of 8-bits. A Multi-frame consists of 16 frames, numbered zero to fifteen. The timeslots are numbered 0 to 31. Timeslot 0 is used for Synchronization, Alarm Transport and International Carrier use. Unfortunately, both T1 and E1 are not really suitable for connection to individual residences.

The so-called digital subscriber lines (**DSL**) has been later invented to enable transmitting *T1/E1* over copper wires, with-out repeaters. However, DSL makes use of other advanced modulation techniques to transmit 1.544 Mb/s over lines up to 4km long.

iv- Higher Order Multiplexing

Multiplexing more than 24 or 30 digital voice frames is called Higher Order Multiplexing. Higher Order Multiplexing is accomplished by multiplexing the standard TDM frames. The Synchronous Digital Hierarchy (SDH) was developed as a standard for multiplexing higher order frames. As shown in figure 3-27, the SDH frames are composite of hierarchical levels of T1(1.544Mb/s), T2(6.312Mb/s), T3(44.736Mb/s) and T4(274.176Mb/s) carriers. The SDH has become the primary transmission protocol in most PSTN networks. It was developed to allow streams of 1.544 Mbit/s and above to be multiplexed. so as to create larger SDH frames known as Synchronous Transport Modules (STM). SDH can also multiplex packet based frames such as Ethernet and asynchronous transfer mode (ATM).

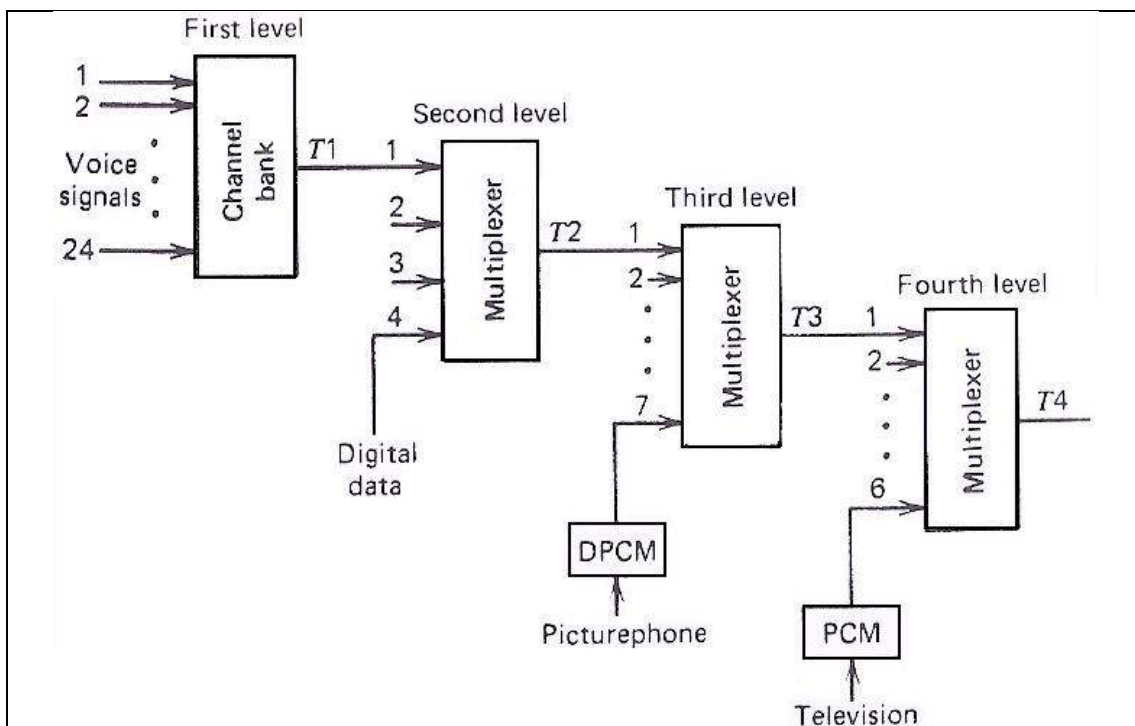


Fig. 3-27. Illustration of the synchronous digital hierarchy (SDH).

Table 3-1. Standard TDM services in American systems

Service	Line	Rate (Mbps)	Voice Channels
DS-1	T-1	1.544	24
DS-2	T-2	6.312	96
DS-3	T-3	44.736	672
DS-4	T-4	274.176	4032

Table 3-2. Standard TDM services in European systems

E Line	Rate (Mbps)	Voice Channels
E-1	2.048	30
E-2	8.448	120
E-3	34.368	480
E-4	139.264	1920

v- Statistical TDM (STDM)

The statistical or asynchronous TDM (STDM) is an advanced version of TDM in which both the address of the terminal and the data itself are transmitted together for better routing. Using STDM allows bandwidth to be split over 1 line. If there is one 10Mb line coming into the building STDM can be used to provide 178 terminals with 56k connection ($178 \times 56k = 9.96\text{Mb}$). A more common use however is to only grant the bandwidth when it is needed. STDM does not reserve a time slot for each terminal. Rather it assigns a slot when the terminal is requiring data to be sent or received.

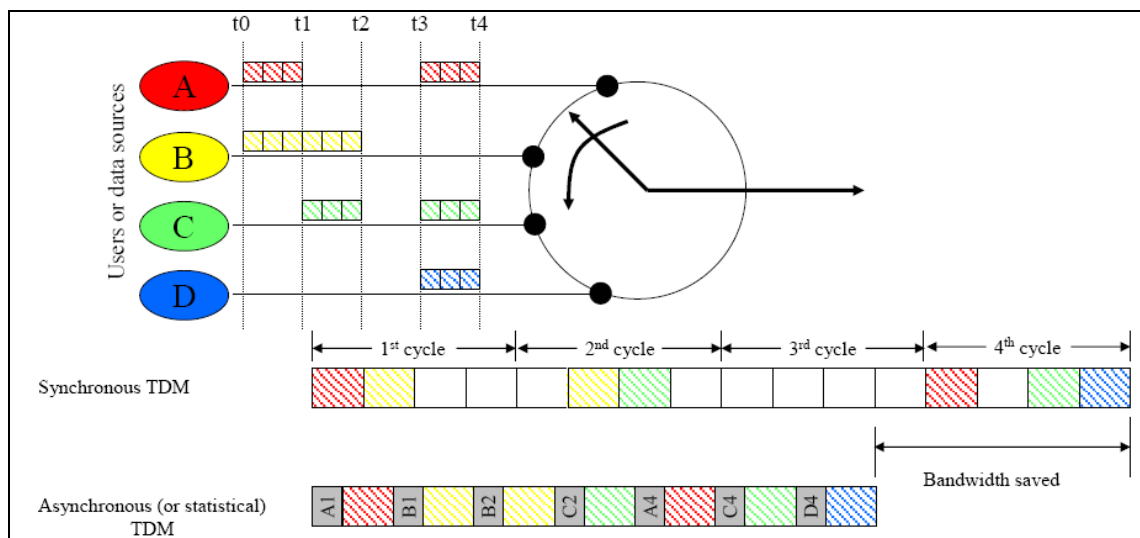


Fig. 3-28. Illustration of the statistical TDM (STDM).

3-7-2. Frequency-Division Multiplexing (FDM)

Frequency-division multiplexing (FDM) is a form of signal multiplexing where multiple baseband signals are modulated on different frequency carriers and added together to create a composite signal. FDM can also be used to combine multiple signals before final modulation onto a carrier

wave. In this case the carrier signals are referred to as subcarriers. For example, a television channel is divided into subcarrier frequencies for video, color, and audio. DSL also uses different frequencies for voice and for upstream and downstream data transmission on the same conductors. By the end of the 20th Century, FDM voice circuits had become rare. Modern telephone systems employ digital transmission, in which time-division multiplexing (TDM) is used instead of FDM.

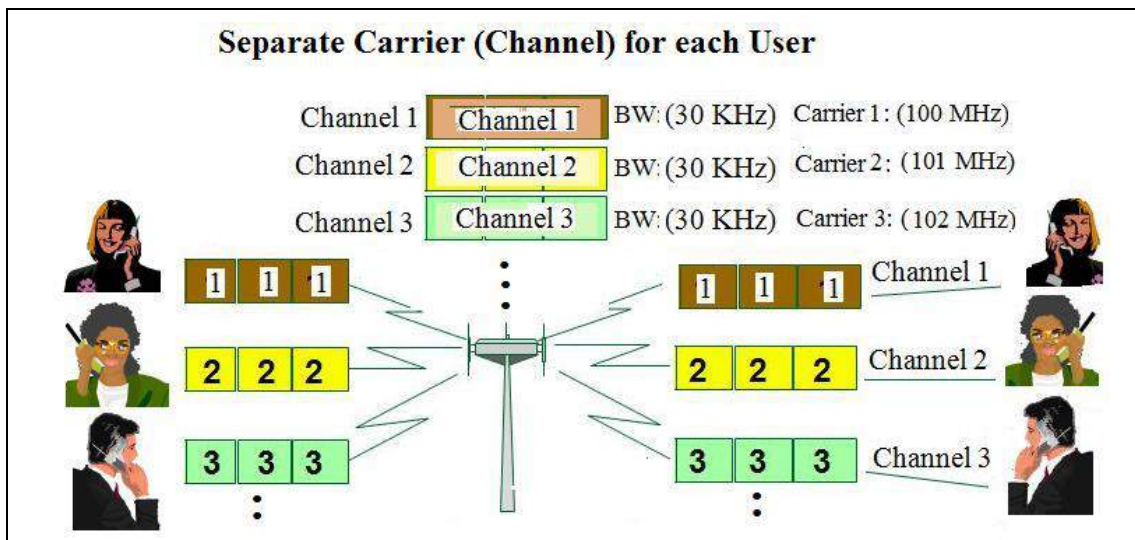


Fig. 3-29. Illustration of the FDM techniques.

3-7.3. Code-Division Multiplexing (CDM)

Code-Division Multiplexing (CDM) is a method of multiple-access that divides a radio channel not by time, nor by frequency, but instead by using different pseudo-random code sequences for each user. CDMA is a form of **spread-spectrum** signalling, since the modulated coded signal has a much higher bandwidth than the data being communicated.

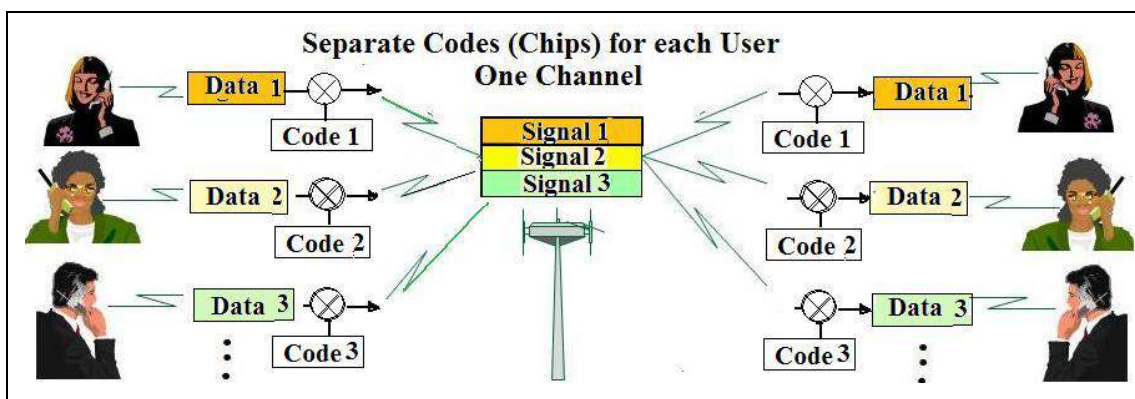


Fig. 3-30. Illustration of the CDMA technique

In CDMA, the data to be transmitted is multiplied (logically XORed) by a faster pseudorandom code. The figure shows how spread spectrum signal is generated. The data signal with pulse duration of T_b is XORed with the code signal with pulse duration of T_c . Therefore, the bandwidth of the data signal is $1/T_b$ and the bandwidth of the spread spectrum signal is $1/T_c$. Since T_c is much smaller than T_b , the bandwidth of the spread spectrum signal is much larger than the bandwidth of the original signal. Multiplication with the code sequence which is of a higher bit rate results in a much wider spectrum. The ratio of the code rate to the information bit rate is called both the **spreading factor** and the **processing gain** of the CDMA system.

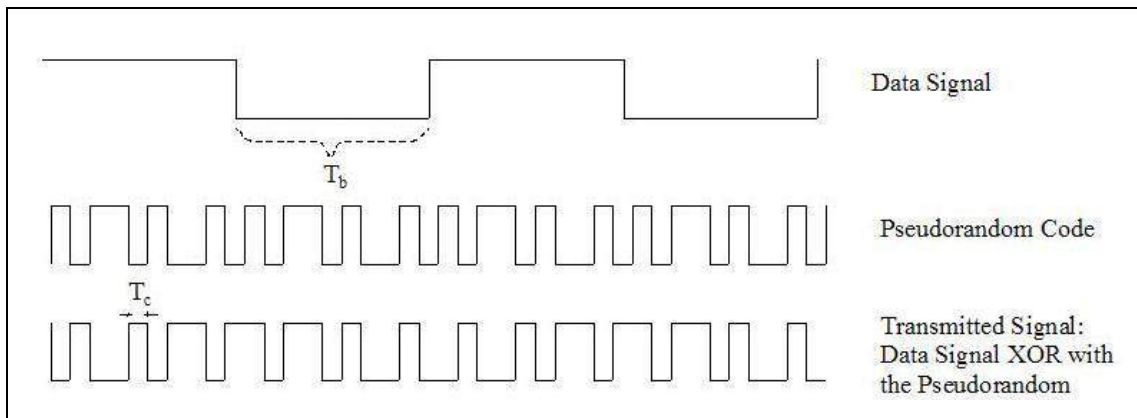


Fig. 3-31. Spreading of data spectrum by pseudorandom code in CDMA technique

In IS-95 cellular phone system, the chipping rate is 1.2288 and the spreading factor is 64. To distinguish the information bit rate from the code rate, we call the code rate **chipping rate**. In effect, we take each data bit and convert it into **k chips**, which is the code sequence. We call it the chipping rate when the code sequence is applied

i. Code Division Modulation & Demodulation

As shown in figure 3-31, both the data and the pseudorandom modulating sequence are binary signals. If original signal is $x(t)$ of power P_s and the code sequence is given by $g(t)$, the resultant modulated signal is

$$s''(t) = \sqrt{2P_s} d(t).g(t) \quad (3-14a)$$

The multiplication of the data sequence with the spreading sequence is the first modulation. Then the signal is multiplied by the carrier which is the second modulation. The carrier here is analog.

$$s(t) = \sqrt{2P_s} d(t).g(t).\sin(2\pi f_c t) \quad (3-14b)$$

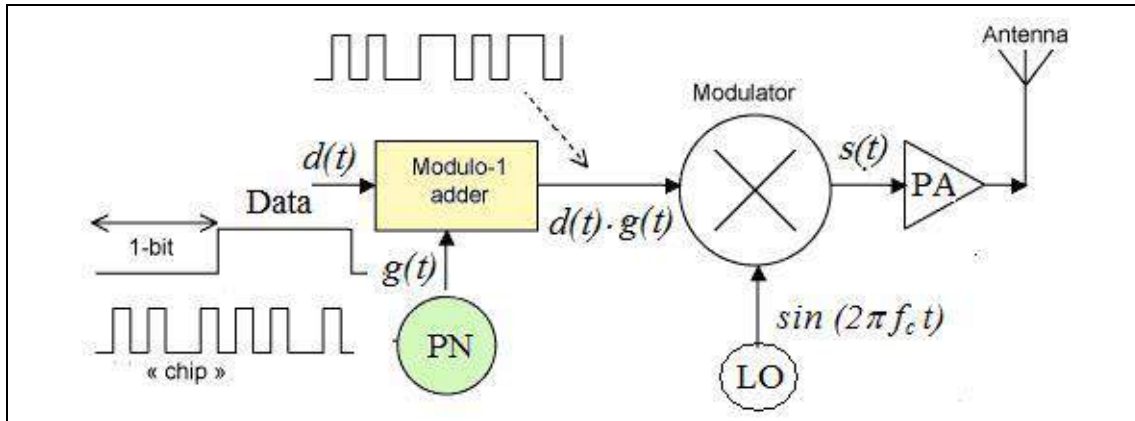


Fig. 3-32(a). CDM Spread spectrum transmitter

On the receive side, we multiply this signal again with the carrier. What we get is this.

$$r(t) = \sqrt{2P_s} d(t).g(t).\sin^2(2\pi f_c t) = \sqrt{2P_s} d(t).g(t).[1 - \cos(2\pi f_c t)] \quad (3-15a)$$

The double frequency term, is filtered out and we are left with the signal.

$$r'(t) = \sqrt{2P_s} d(t).g(t). \quad (3-15b)$$

Now we multiply this remaining signal with $g(t)$, the code sequence and we get:

$$r''(t) = \sqrt{2P_s} d(t).g(t).g(t). \quad (3-15c)$$

Now from having used a very special kind of (orthogonal) sequences, we say that correlation of $g(t)$ with itself is a certain scalar number which can be removed and we get the original signal back.

$$r''(t) = \sqrt{2P_s} d(t). \quad (3-15d)$$

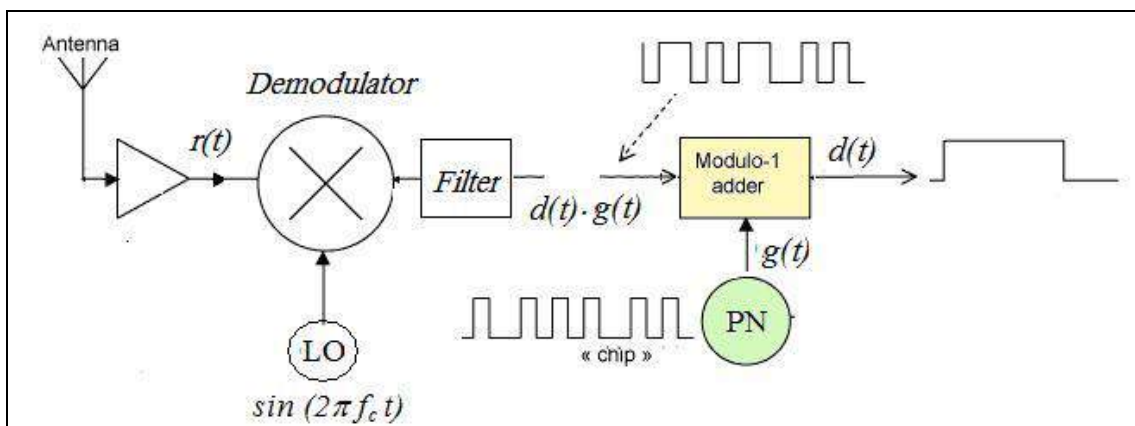


Fig. 3-32(b). CDM Spread spectrum receiver

As shown in figure 3-33, the PN generator can be implemented as a multi-stage feedback shift register (LFSR). Here, some bits of the output are fed back to the input, via exclusive OR gates. Hence, the repeated sequence at the output may be very long. In fact, the maximum sequence is 2^n where n is the number of register stages (bits). If the clock rate is 1GHz and $n = 64$, then the initial sequence needs more than 500 year to repeat. As the possibility of any bit to be '0' or '1' is almost 50%, this circuit is sometimes called a pseudo-random noise generator.

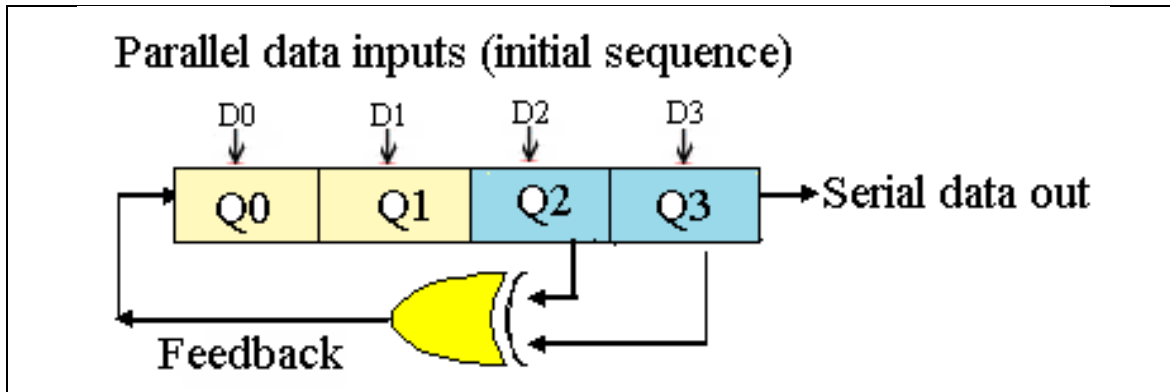


Fig. 3-33. Logic diagram of a simple pseudo noise (PN) generator, in the form of a 4-stage linear feedback shift register (LFSR).

Thus, in CDMA we do modulation twice. First with a binary sequence $g(t)$, the properties of which we will discuss below and then by a carrier. The binary sequence modulation ahead of the carrier modulation accomplishes two functions. First, it spread the signal and second, it introduces a form of encryption because the same sequence is needed at the receiver to demodulate the signal. In IS-95 and CDMA 2000 this is done three times, once with a code called **Walsh Code**, then with a code called **Short Code** and then with one called **Long code**.

ii. Properties of Spreading Codes

Synchronous CDMA, exploits its orthogonally properties. Thus, each user is assigned a different code in the sender side, which may be represented by a **vector**. The dot product of any vector in itself is a scalar number. Also, the dot product of any two different vectors is zero, so that the codes are said to be *orthogonal* to each other.

CDMA has been used in many communications and navigation systems, including the Global Positioning System (**GPS**). CDMA is also used in cellular telephony systems that make use of this multiple access scheme, such as W-CDMA by the International Telecommunication Union (**ITU**).

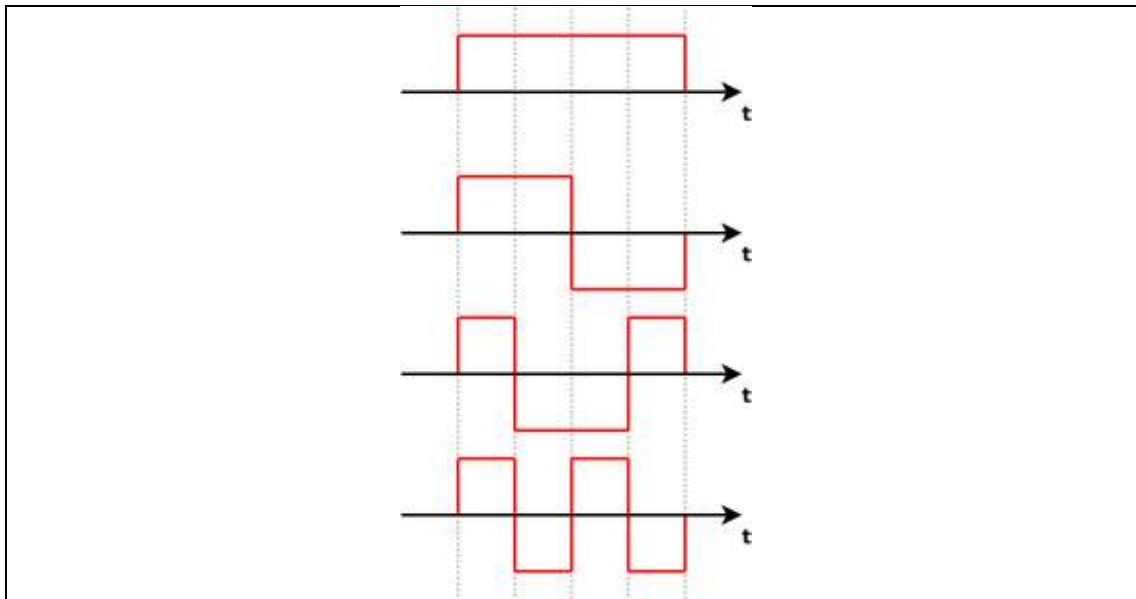


Fig. 3-34. Example of four mutually orthogonal digital codes.

3-7.4. Wave-Division Multiplexing (WDM)

Wave-Division Multiplexing (WDM) is a technology which multiplexes multiple optical carrier signals on a single optical fiber by using different wavelengths (colors) of laser light to carry different signals. This allows for a multiplication in capacity, in addition to enabling bidirectional communications over one strand of fiber. This is a form of frequency division multiplexing (FDM) but is commonly called wavelength division multiplexing. The term *wavelength-division multiplexing* is commonly applied to an optical carrier (which is typically described by its wavelength), whereas frequency-division multiplexing typically applies to a radio carrier (which is more often described by frequency). However, since wavelength and frequency are inversely proportional and since radio and light are both forms of electromagnetic waves, the two terms are equivalent.

The first WDM systems only combined two signals. Modern systems can handle up to 160 signals and can thus expand a basic 10 Gb/s fiber system to a theoretical total capacity of over 1.6 Tb/s over a single fiber pair. WDM systems are popular with telecommunications companies because they allow them to expand the capacity of the network without laying more fiber. By using WDM and optical amplifiers, they can accommodate several generations of technology development in their optical infrastructure without having to overhaul the backbone network. Capacity of a given link can be expanded by simply upgrading the multiplexers and demultiplexers at each end.

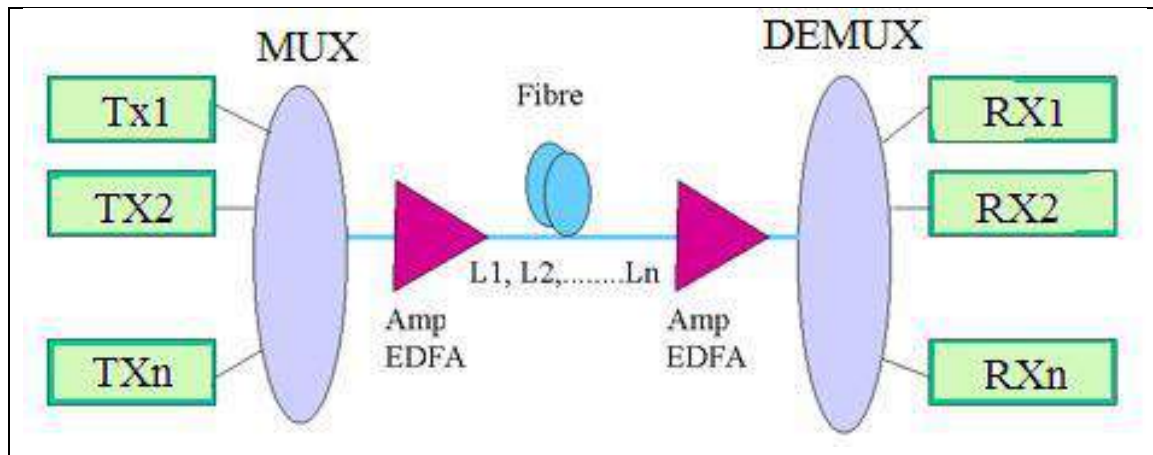


Fig. 3-35. Illustration of the wave-division multiplexing technique.

Dense Wavelength Division Multiplexing (**DWDM**). refers originally to optical signals multiplexed within the 1550-nm band so as to leverage the capabilities (and cost) of erbium doped fiber amplifiers (**EDFAs**), which are effective for wavelengths between approximately 1525 nm - 1565 nm (C band) or 1570 nm - 1610 nm (L band). The EDFAs cost is thus leveraged across as many channels as can be multiplexed into the 1550-nm band. In conclusion, WDM and DWDM are based on the same concept of using multiple wavelengths of light on a single fiber, but differ in the spacing of the wavelengths, number of channels and the ability to amplify multiplexed signals in optical space.

3-8. Baseband Transmission Problems

In this section we study the effects of channel impairments such as **limited bandwidth**, **noise** and **inter-symbol interference (ISI)** on baseband pulse transmission systems. We also introduce the eye pattern concept, and how it can be used to evaluate the performance of the data transmission.

3-8.1. Channel Limitations

The baseband channel may be represented as a low-pass filter. The bandwidth of a channel is thus the cut-off frequency of the filter. Therefore, our goal is to reduce the required transmission bandwidth as much as possible. The reduction of transmitted signal bandwidth is a serious requirement for both baseband and passband communication systems. We mean by **baseband** systems, the communication systems, where we directly transmit the baseband signal (e.g., PCM signals), without any further modulation with RF carriers. On the other hand, in the **passband** communication systems, the baseband signals (analog or digital) are used to modulate an RF carrier, prior to transmission.

We have already talked about the number of pulses per second that can be sent over a channel of bandwidth B . We shall use the term *symbols* rather than pulses from now on. The number of symbols per second is called the *symbol rate*. **It is the symbol rate that determines the transmission bandwidth.** Supposing that a symbol can take on M possible levels and we send $2B$ symbols per second over a B Hz band-limited channel, the transmission bit rate is $2B \cdot \log_2 M$ bits per second. The *bandwidth efficiency* is $2 \log_2 M$ bits per second per Hz. If B is fixed, we can increase the bit rate by simply increasing the value of M .

3-8.2. Inter-Symbol Interference (ISI)

Due to bandwidth limitations of the transmission medium, transmitted pulses may spread in time and overlap, as shown in the following figure. This overlapping of transmitted pulses is called inter-symbol-interference (**ISI**). Also, the **ISI** errors may be caused by multipath interference. If we detect the signals at the sampling instants, overlapping from adjacent signals may result in erroneous detection. This interference can be reduced if we increase the available *channel bandwidth*. Instead, we can **shape** the **pulse** at the transmitter to minimize or eliminate this interference effect rather than expanding the channel bandwidth. One obvious choice is to use a shape that is maximum at the sampling instant, yet goes through zero at all adjacent sampling instants. The pulse goes through zero at multiples of $T = 1/2B$ seconds, where B is the channel bandwidth.

By sampling at multiples of $1/2B$ seconds, pulses of the same shape that are spaced $1/2B$ seconds apart will not interfere with each other. A maximum of $2B$ pulses per second may be transmitted over a channel bandwidth of B Hz without ISI. This is the *Nyquist rate* for no ISI. However, this particular pulse shape is difficult to implement.

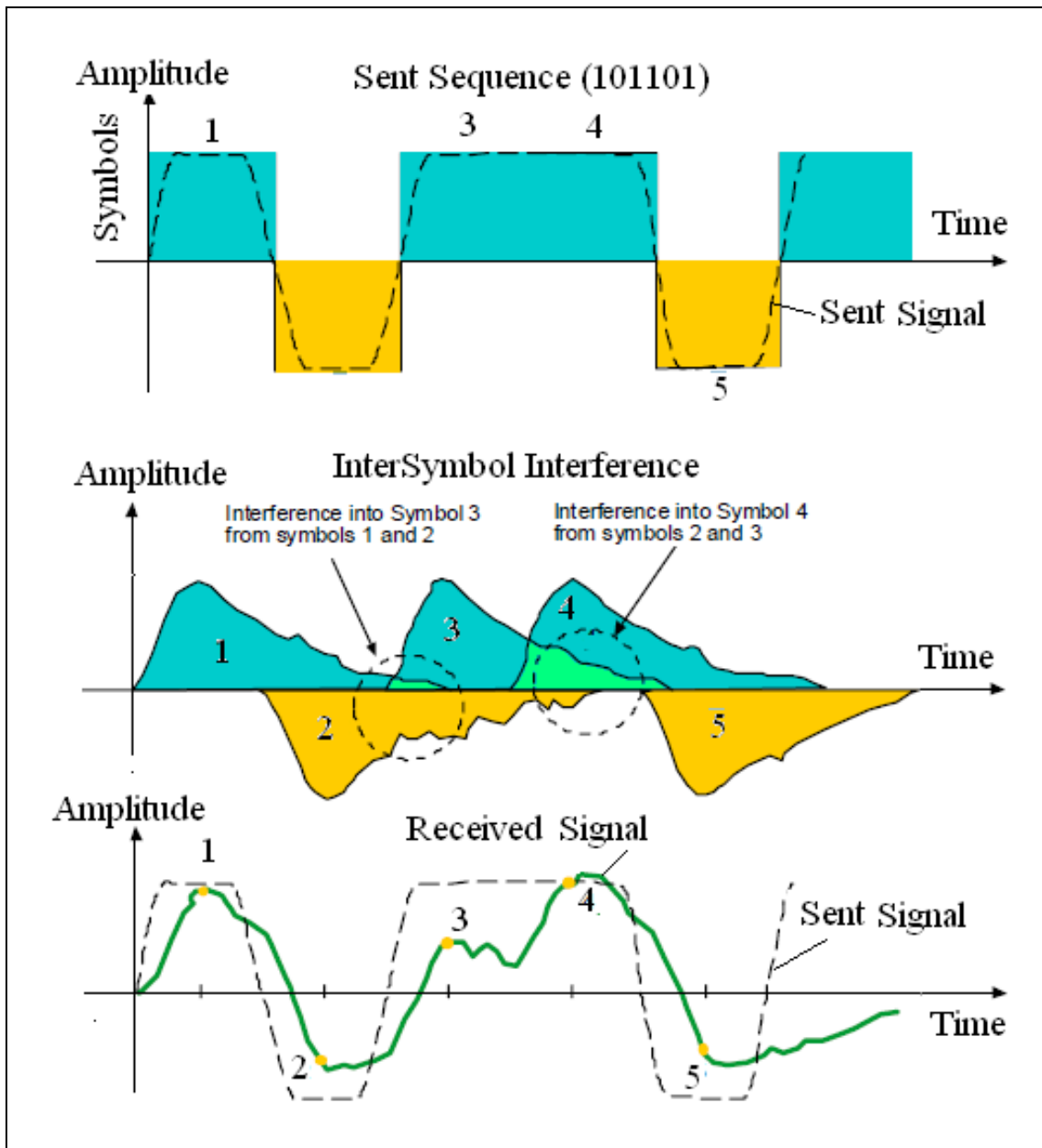


Fig. 3-36. Illustrate the inter-symbol interference (ISI) problem. due channel effects

3-8.3. Jitter

The Jitter is the generic term given to the difference between the (notional) “correct” impossible for this timing to be exact because of the nature of the operation being performed. Some bits will be detected slightly early and

others slightly late. This means that the detected timing will vary more or less randomly by a small amount either side of the correct timing - hence the name *jitter*. It doesn't matter if all bits are detected early (or late) provided it is by the same amount - delay is not jitter. Jitter is a random variation in the timing either side of what is correct.

Jitter is minimized if both the received signal is tackled with high quality phase-locked loop (PLL). But although you can minimize jitter, you can never quite get rid of it altogether. Jitter can have many sources, such as distortion in the transmission **channel** or just the method of operation of a digital PLL. In optical systems the predominant cause of jitter is dispersion.

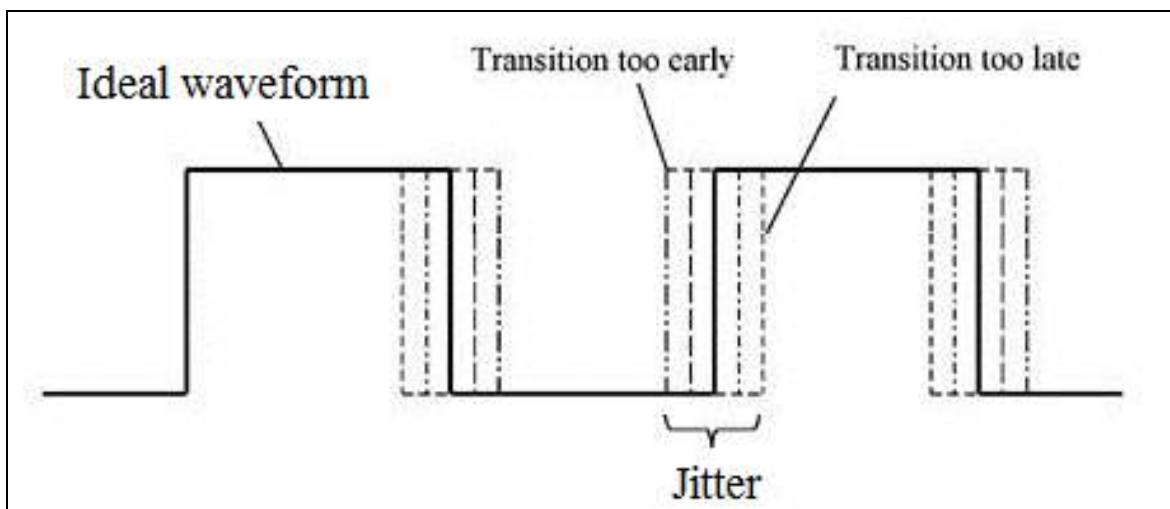


Fig. 3-37. Illustration of the jitter effect

3-8.4. Eye Pattern

The eye pattern or eye diagram is a timing analysis tool which visualizes the timing and error levels of a signal. The eye pattern is an overlay of graphs of the output of a receiver. It consists of many (hundreds to millions) of instances of the signal displayed over the top of one another. In digital communications, the eye diagram provides information such as the suitable interval for sampling of the received signal, the sensitivity of the system to timing errors and its noise margin. The figure 3-38 illustrates the eye pattern diagram.

As we will see in Chapter 5, the digital modulation techniques may involve the transmission of different symbols; called *M-ary* signals. The eye patterns can be used to diagnose transmission problems of such signals. Figure 3-39 shows the setup of an eye pattern experiment.

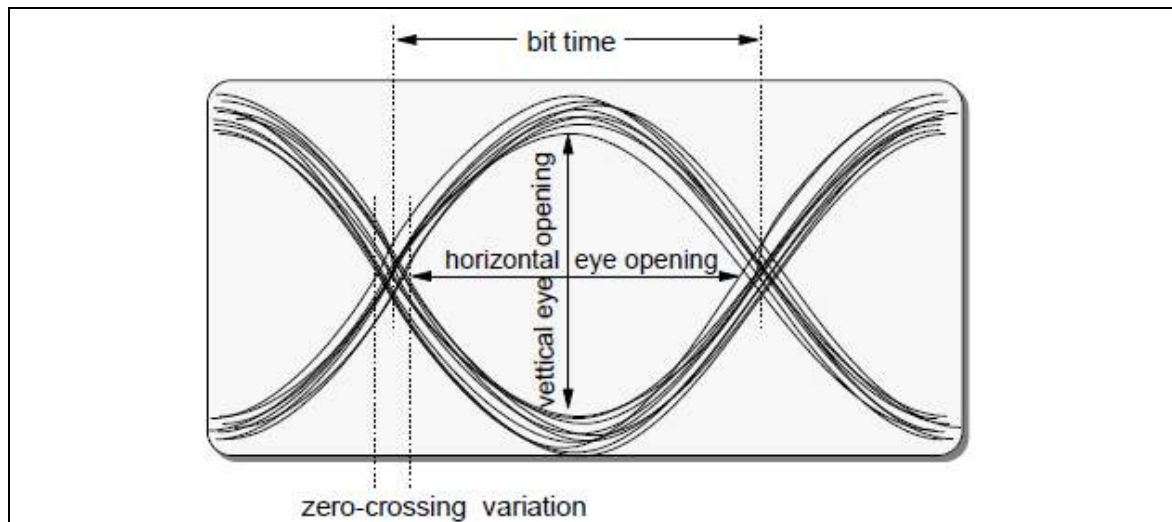


Fig. 3-38. Schematic illustration of the eye diagram

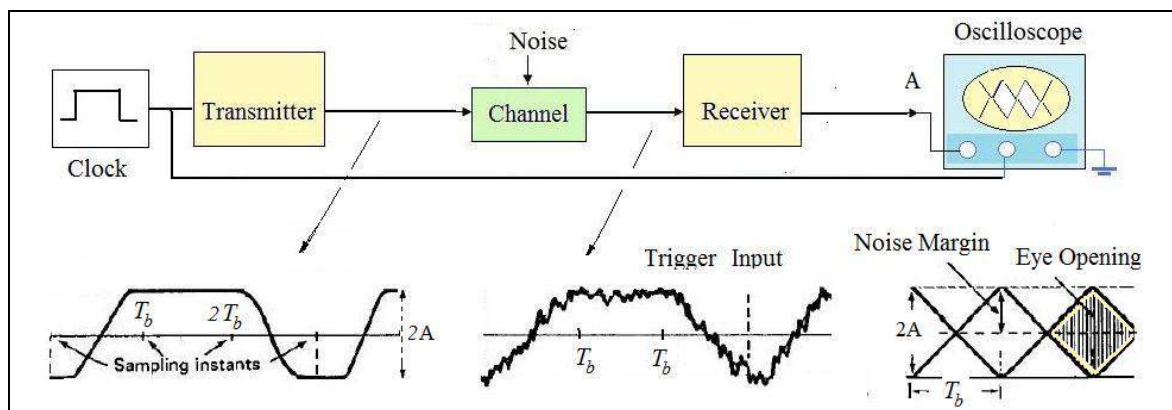


Fig. 3-39. Block diagram to illustrate the eye pattern experiment. The oscilloscope is fed by the digital data signal from a receiver, which is sampled and applied to the vertical input, while the data rate is used to trigger the horizontal sweep

In conclusion we can write the following rules about the eye patterns:

- The **eye opening** is defined as the distance from the decision threshold to the closest trace at the sampling instant.
- The **noise margin** of the system at a certain sampling time. is defined as the height of the eye opening.
- The amount of **timing jitter** is defined by the horizontal width around the zero crossings.
- The amount of amplitude distortion (e.g.. **ISI**) is defined by the vertical trace width at the best sampling instant.
- Non-symmetries in the eye diagram indicate the presence of non-linearity.

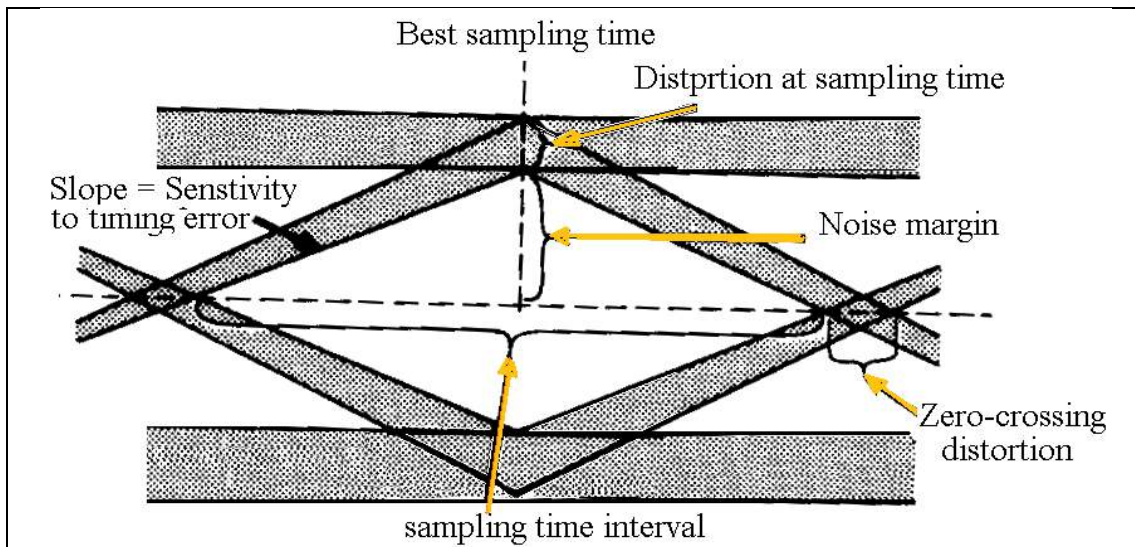


Fig. 3-40. Illustration of the main features in the eye pattern diagram

3-9. Pulse Detection & Matched Filters

The pulse detection and estimation of a transmitted pulse in the presence of noise is a basic problem in digital communications. Figure 3-41 depicts the block diagram of a radio receiver, which is capable of pulse detection and estimation in the presence of noise.

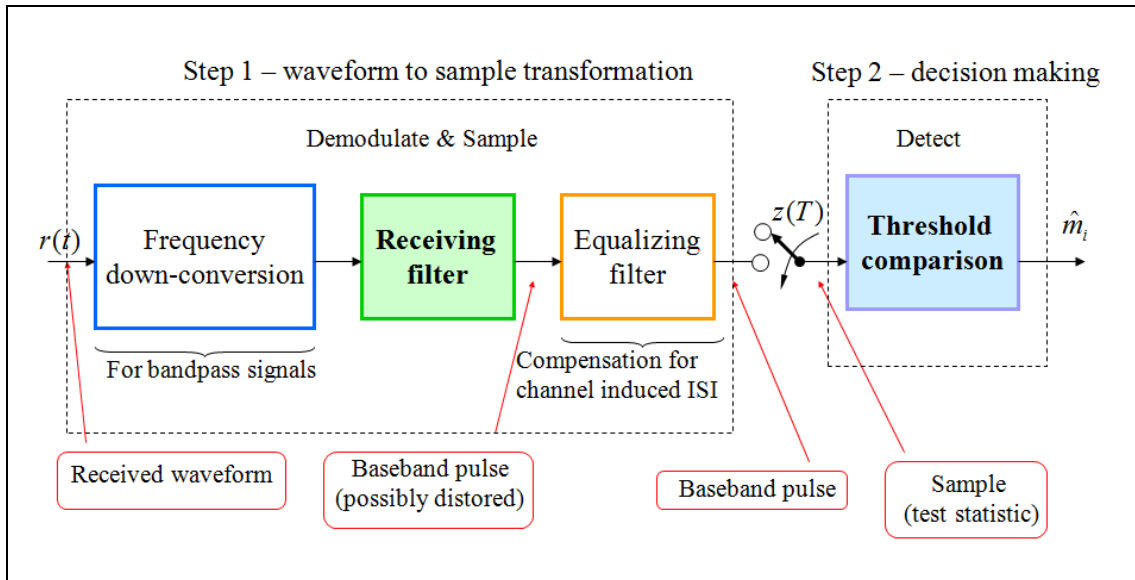


Fig. 3-41. Illustration of the steps of baseband pulse reception.

3-9.1. Pulse Detection

The following figure depicts how a transmitted signal is corrupted through the channel and how it can be restored in a digital receiver. Practical receivers estimate the transmitted signal by using a technique known as matched filtering. A receiver employing such a technique filters the received signal with a filter whose shape is *matched* to the transmitted signal's pulse shape.

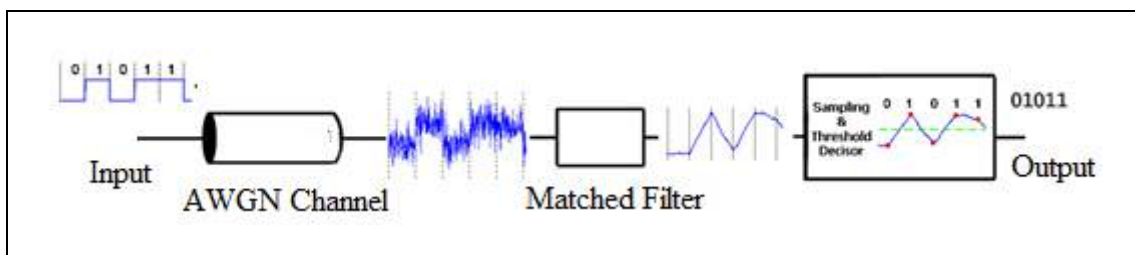


Fig. 3-42. Illustration of the baseband transmission, and reception, by matched filters.

3-9.2. Matched Filters

Matched filters are some sort of ideal filters, and not actually a specific type of filters. They are used to process a received digital signal to minimize the

noise effect and increase the signal-to-noise ratio (SNR). In the context of a communication system that sends binary messages from the transmitter to the receiver across a noisy channel, a matched filter is employed to detect the transmitted pulses in the noisy received signal. We can derive the matched filter response, $h(t)$, that maximizes output SNR by several methods. Mathematically speaking, the matched filter is the linear filter that maximizes the output SNR at the sampling times of the transmitted signal. Thus, we seek a filter such that we maximize the output SNR, where the output, $y(t)$, is the inner product of the filter response and the observed signal $x(t)$.

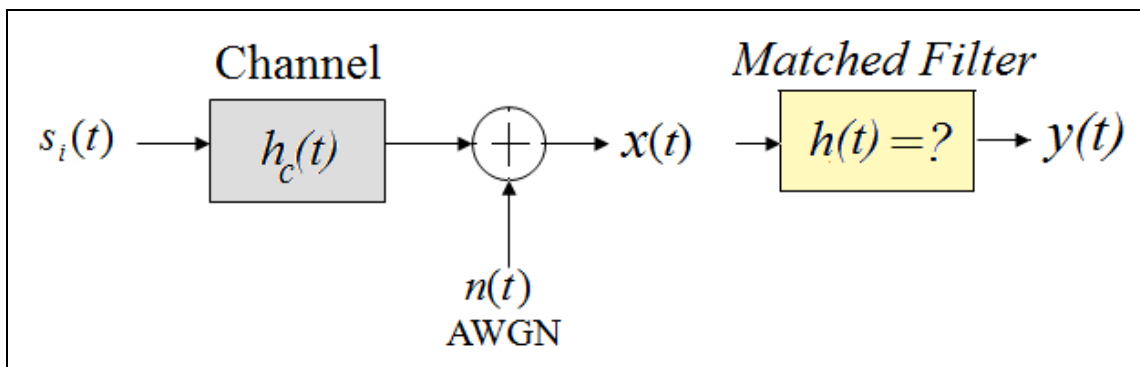


Fig. 3-43. Matched filter receiver.

Our observed signal consists of the desirable signal s (after passing the channel) and additive noise n . For simplicity, we may assume an ideal channel with a unity transfer function, such that $x(t) = s_i(t) + n(t)$. Assuming the matched filter impulse response is $h(t)$, and its output is $y(t)$, it can be proved that the optimum receiver filter is the matched filter, whose impulse response is given by:

$$h(t) = s_i^*(T-t) \quad (3-16a)$$

Here the index i is the sample index and s_i^* is the conjugate of the i^{th} sample of the transmitted pulse signal. The frequency response of the filter is then:

$$H(f) = S_i^*(f) \cdot \exp(-j2\pi f T) \quad (3-16b)$$

This is a time reversed and delayed version of the conjugate of the transmitted signal. The following figure shows some examples of sample shapes and the corresponding matched filters. The Fourier transform of output of the matched filter output is hence:

$$Y(f) = |S(f)|^2 \cdot \exp(-j2\pi f T) \quad (3-16c)$$

which is proportional to energy spectral density (ESD) of the transmitted pulse signal.

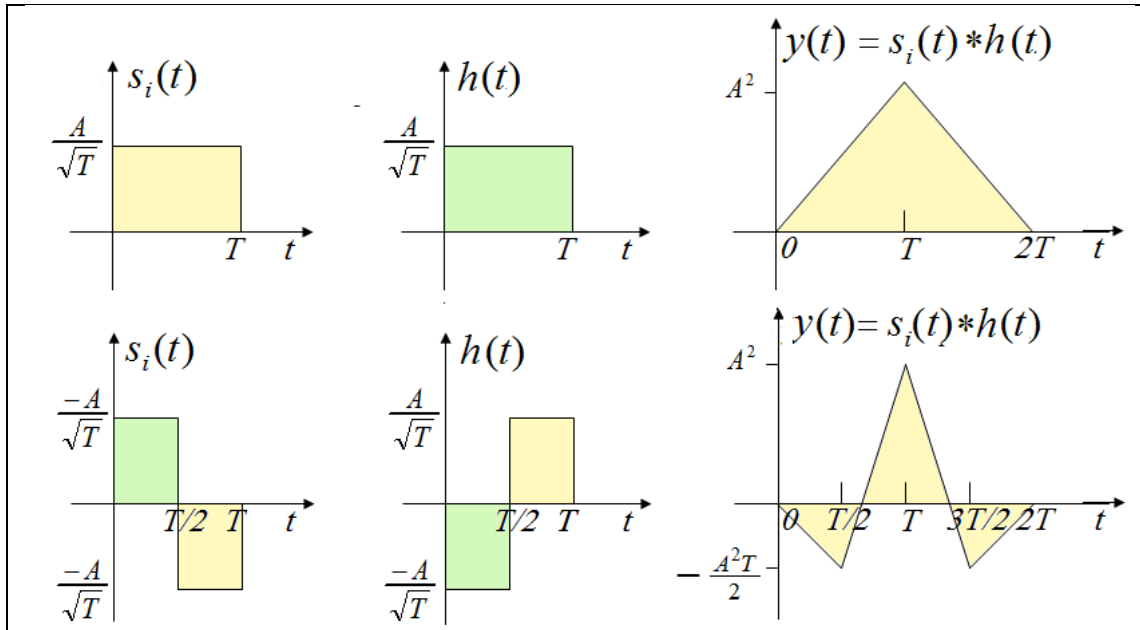


Fig. 3-44. Examples of matched filters response

In other words, the output signal of a matched filter is proportional to a shifted version of the autocorrelation function of the input signal to which the filter is matched.

$$y(t) = R_s(t-T) \cdot y(T) = R_s(0) = E_s \quad (3-17)$$

where E_s is signal energy such that the output SNR of the matched filter depends only on the power spectral density of the white noise at the filter input ($\frac{1}{2} N_o$).

$$SNR = E_s / \frac{1}{2} N_o \quad (3-18)$$

Note 3-2. Signal Autocorrelation & Cross Correlation of 2 Signals

The energy of a periodic (finite power) signal $x(t)$ over a certain duration ($-T/2, T/2$) is

$$E_x^T = \int_{-T/2}^{T/2} |x^2(t)| dt$$

The average power dissipated by the periodic signal over ($-T/2, T/2$) is

$$P_x^T = \frac{1}{T} E_x^T = \frac{1}{T} \int_{-T/2}^{T/2} |x(t)|^2 dt$$

The **autocorrelation** of a periodic signal is given by:

$$R_x(\tau) = \langle x(t)x(t+\tau) \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x^*(t)x(t+\tau) dt$$

The average power is related to the autocorrelation function as follows:

$$P_x^T = R_x(0)$$

Now consider a non-periodic (finite energy) or random signal $x(t)$. Define $X(\omega)$ as the Fourier Transform of $x(t)$

$$X(\omega) := FT\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt$$

The energy spectral density (**ESD**) of $x(t)$ is then

$$ESD = |X(\omega)|^2$$

The energy of the signal $x(t)$ is

$$E_x = \int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega$$

This is called the Parseval Theorem. The autocorrelation of an energy signal is given by:

$$R_x(\tau) := \int_{-\infty}^{\infty} x^*(t)x(t+\tau) dt, \quad -\infty < \tau < \infty$$

The power spectral density **PSD** of the signal is:

$$S_x(\omega) = FT\{R_x(\tau)\}$$

The autocorrelation function has a number of properties:

- $R_x(\tau) = R_x(-\tau)$
- $|R_x(\tau)| \leq R_x(0)$
- $S_x(\omega) = |X(\omega)|^2 = FT\{R_x(\tau)\}$
- $E_x = R_x(0) = \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega$

Example 3-6.

For the matter of illustration, imagine we want to send the sequence "0101100100" coded in the form of non-polar pulses through a certain channel. Mathematically, a sequence in the code may be described as a sequence of unit pulses. each pulse being weighted by +1 if the bit is "1" and by 0 if the bit is "0". In this case, the scaling factor for the k^{th} bit is:

$$a_k = \begin{cases} 1, & \text{if bit } k \text{ is 1,} \\ 0, & \text{if bit } k \text{ is 0.} \end{cases} \quad (3-19)$$

Hence, we can represent the message. $M(t)$, as the sum of shifted unit pulses:

$$M(t) = \sum_{k=-\infty}^{\infty} a_k \times \Pi\left(\frac{t - kT}{T}\right). \quad (3-20)$$

where T is the time length of one bit. Thus, the signal to be sent by the transmitter looks like this

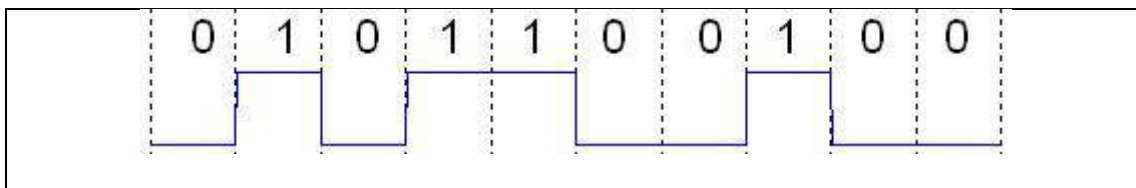


Fig. 3-45(a). Example of a pulse sequence.

We may model our noisy channel as a white Gaussian noise (AWGN) channel, added to the signal. At the receiver end, this may look like the following figure, for a signal-to-noise ratio (SNR) of 3dB.

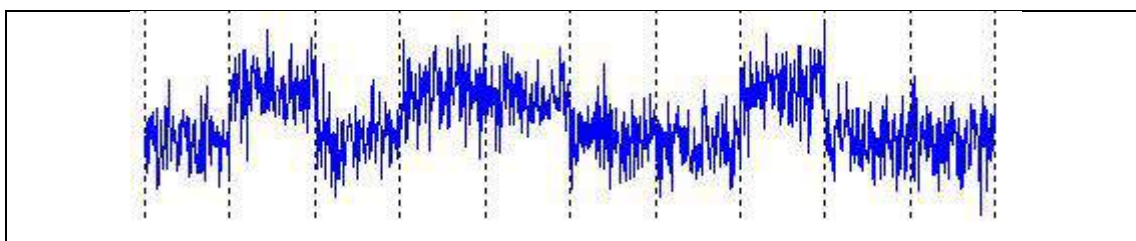


Fig. 3-45(b). Pulse sequence. after adding white noise. such that the SNR = 3dB.

A first glance will not reveal the original transmitted sequence. There is a high noise power relative to the signal power i.e., there is a low SNR. If the receiver were to sample this signal at the correct moments, the resulting binary message would possibly belie the original transmitted one.

In order to increase the SNR, we pass the received signal through a **matched filter**. In this case, the **impulse response** of the ideal matched filter, assuming white (uncorrelated) noise should be a **time-reversed complex-conjugated scaled version of the signal** that we are seeking. We may choose a rectangular pulse, as follows:

$$h(t) = \Pi(t/T) \quad (3-21)$$

In this case, due to symmetry, the time-reversed complex conjugate of $h(t)$ is in fact $h(t)$, allowing us to call $h(t)$ the impulse response of our matched filter convolution system. After convolving with the correct matched filter, the resulting signal, $M_{filtered}(t)$ becomes:

$$M_{filtered}(t) = M(t) * h(t) \quad (3-22)$$

where * denotes the convolution process.

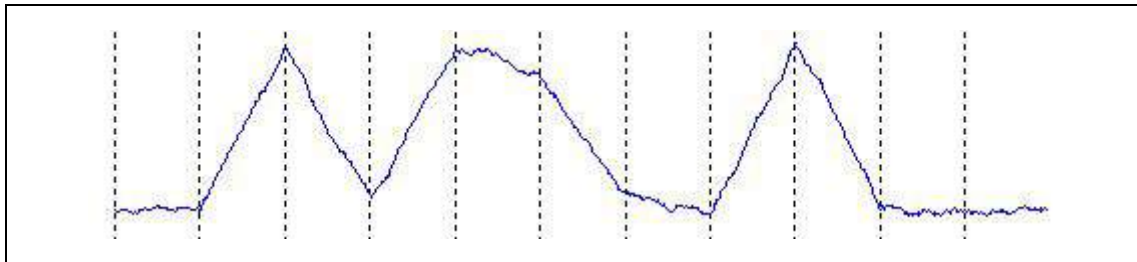


Fig. 3-46(a). Pulse sequence, after the matched filter.

This can now be safely sampled by the receiver at the correct sampling instants, and compared to an appropriate threshold, resulting in a correct interpretation of the binary message.

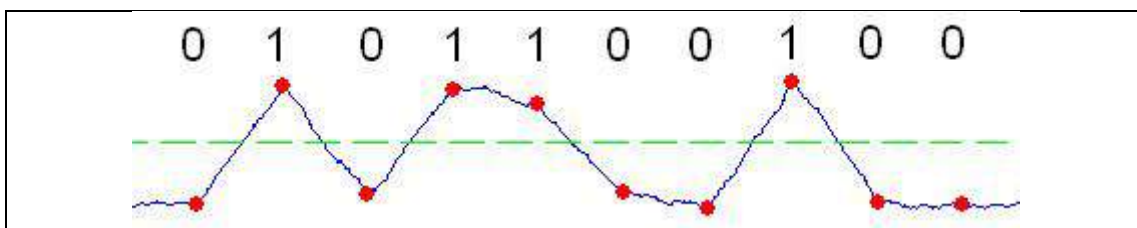


Fig. 3-46(b). Sampling of a received signal, to reproduce the original pulse sequence

Since **the matched filter is the filter that maximizes the SNR** it can be shown that it also minimizes the Bit error ratio (**BER**), which is the ratio of the number of bits that the receiver interprets incorrectly as a fraction of the total number of bits sent.

3-9.3. Practical Pulse Shaping

Although the use of matched filtering gives the optimum performance in the presence of AWGN, there is still a problem with using a rectangular pulse shape. Recall from Fourier theory that a rectangular pulse in the time domain is equivalent to a *sinc* pulse in the frequency domain. Because the tails of the *sinc* pulse extend to infinity, such a pulse shape would require a system with infinite bandwidth. Although the *sinc* pulse represents the ideal pulse shape, it cannot be implemented in practice. However, practical pulse shapes can be formed by sharpening the **roll-off** of the filter spectrum, for instance, using the so-called cosine-raised filters.

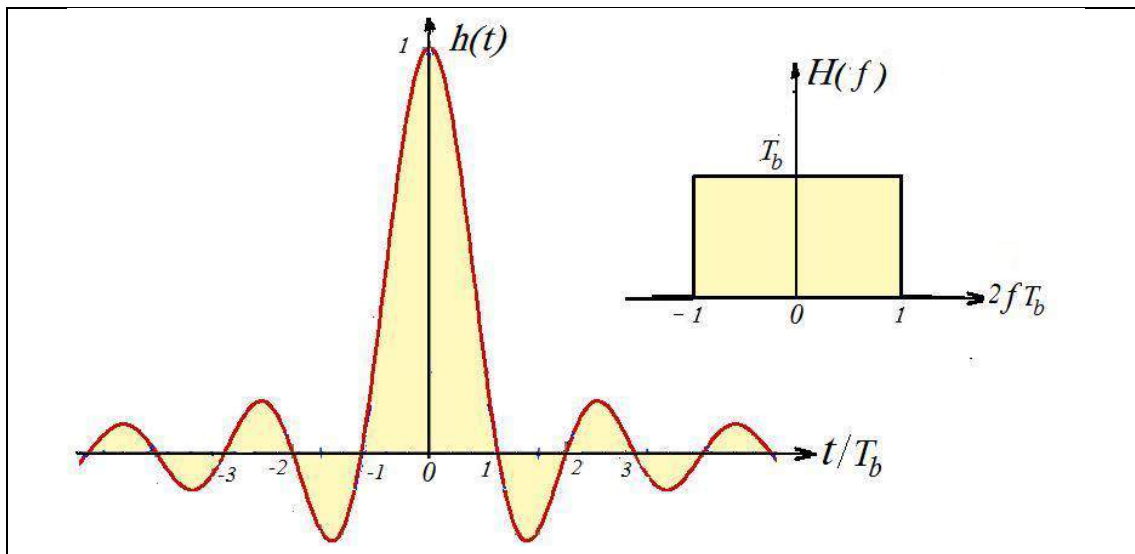


Fig. 3-47. Ideal rectangular *sinc* filter (right) and its impulse response (left)

Note 3-4. The Sinc Function

The *sinc* function has two definitions. In digital signal processing and information theory, the normalized *sinc* function is commonly defined by

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}.$$

It is called normalized because its Fourier transform is the rectangular function and its square integral is 1. In mathematics, the un-normalized *sinc* function is defined as

$$\text{sinc}(x) = \frac{\sin(x)}{x}.$$

3-9.4. Cosine-Raised Filters

In typical transmission schemes, we do not hear of pulse shaping using the $\text{sinc}()$ filters. Rather, pulse shaping using raised cosine filter is frequently used. The cosine raised filter is defined by the following transfer function:

$$h(t) = \text{sinc}(\pi t/T) [\cos(\alpha \pi t/T)] / [1 - (2\alpha \pi t/T)^2] \quad (3-23)$$

where α is the **excess bandwidth parameter** which takes values from 0 to 1. With $\alpha = 0$, the raised cosine filter reduces to the classical Nyquist filter with zero excess bandwidth outside $\pm 1/2T$. With $\alpha = 1$ it is called 100% excess bandwidth and does not occupy frequencies outside $1/T$. The transfer function of a raised-cosine filter is given by:

$$\begin{aligned} H(f) &= [1 + \cos(2\alpha \pi f / 4f_c)] / 4f_c & \text{for } |f| \leq 2f_c \\ &= 0 & \text{elsewhere} \end{aligned} \quad (3-24)$$

where f_c is equal to the channel bandwidth ($f_c = 1/T = B$).

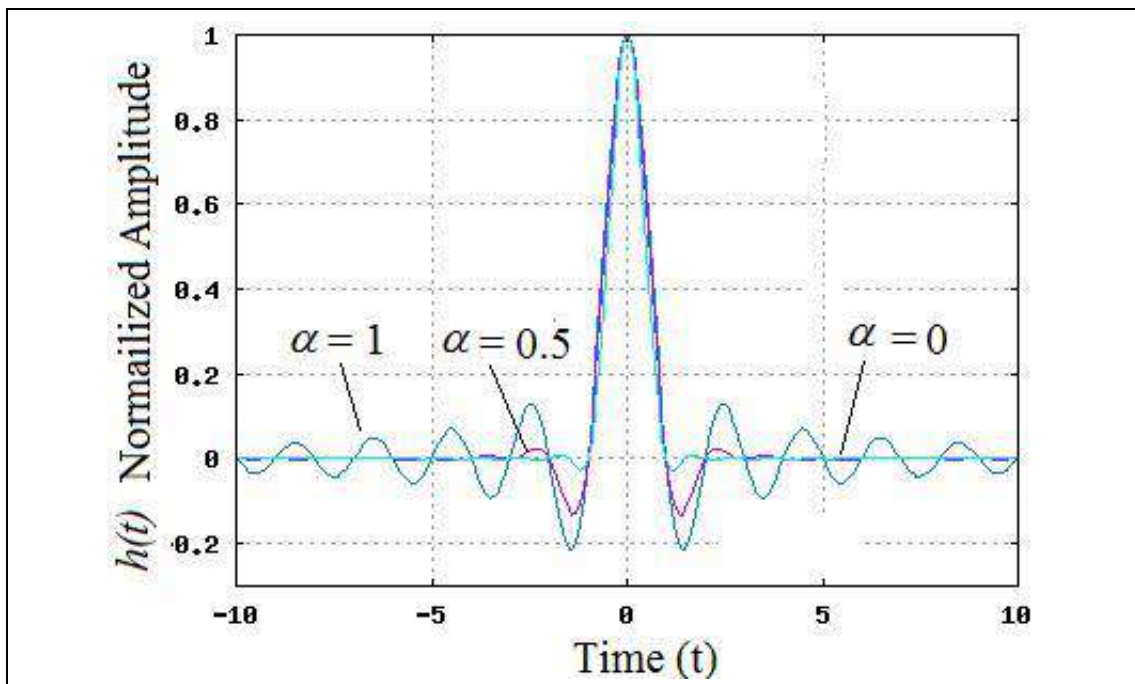


Fig. 3-48. Time-domain response of a cosine-raised filter.

Figure 3-48 shows the impulse or time-domain response of a raised cosine filter, one class of **Nyquist** filter. **Nyquist** filters have the property that their impulse response rings at the symbol rate. It can be seen that, for the same transmission rate of $1/T$ pulses per second, the use of a raised-cosine filter doubles the bandwidth requirement.

Example 3-7.

If the sampling rate = 8000 samples/sec. then calculate the sampling interval and the channel required bandwidth

Solution

Sampling interval = $T = 1/\text{sampling rate} = 1/8000$ sec.

Using a raised-cosine filter, the required bandwidth $B = 1/T = 8$ kHz.

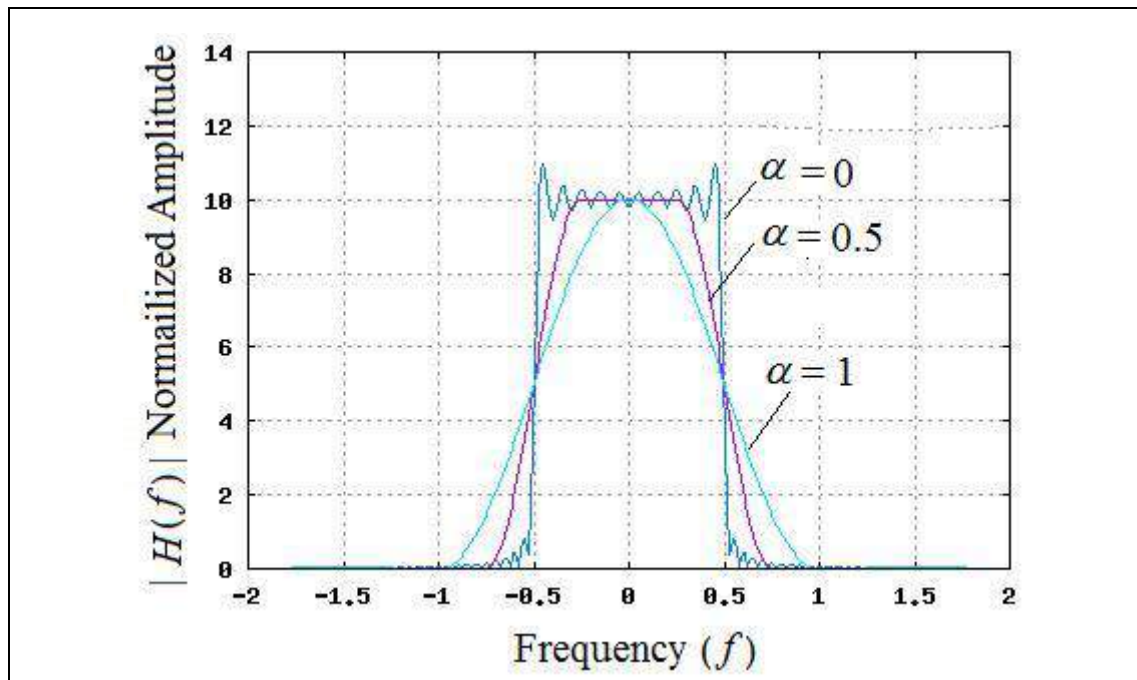


Fig. 3-49.. Frequency-domain response of a *cosine-raised* filter.

Sometimes filtering is desired at both the transmitter and receiver. Filtering in the transmitter reduces the adjacent-channel-power interference. Filtering at the receiver reduces the effects of broadband noise and also interference from other transmitters in nearby channels. This is why root-Nyquist filters are used in receivers and transmitters as $\sqrt{\text{Nyquist}} \times \sqrt{\text{Nyquist}} = \text{Nyquist}$.

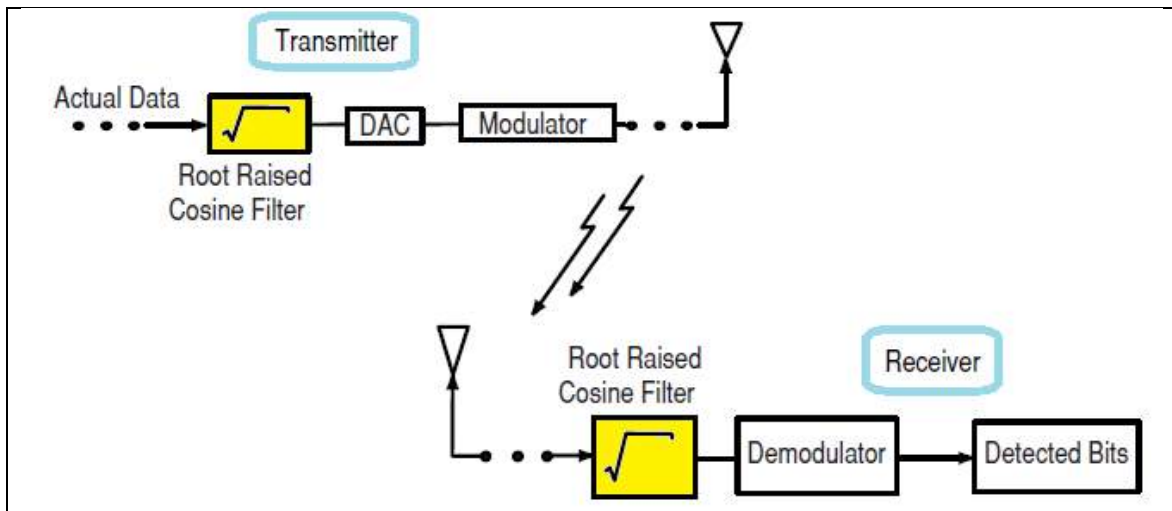


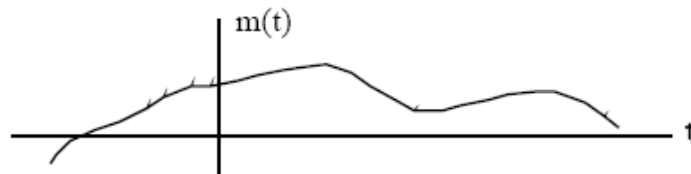
Fig. 3-50.. Frequency-domain response of a *cosine-raised* filter.

3-10. Summary

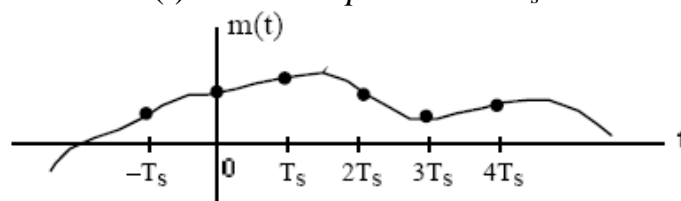
Pulse modulation deals with the time and amplitude discretization of the analog source signal and therefore, is tantamount to the digital communication systems. Typical analog source signals in communication systems are speech and image signals. Pulse modulation schemes aim at transferring a narrowband analog signal over an analog lowpass channel as a two-level quantized signal. Some pulse modulation schemes also allow the narrowband analog signal to be transferred as a digital signal over a digital transmission system. **They are not modulation schemes in the conventional sense** since they are not channel coding schemes but may be considered as source coding schemes and in some cases analog-to-digital conversion techniques.

- | | |
|--|-----------------------|
| 1- Pulse-amplitude modulation (PAM) | (Analog-over-analog) |
| 3- Pulse-width modulation (PWM) | (Analog-over-analog) |
| 3- Pulse-position modulation (PPM) | (Analog-over-analog) |
| 5- Pulse-code modulation (PCM) | (Analog-over-digital) |
| 6- Delta-sigma modulation ($\Sigma\Delta$) | (Analog-over-digital) |

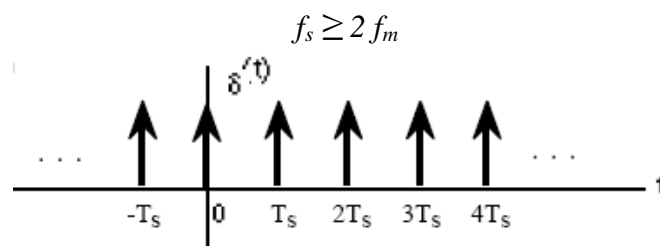
In analog communications, we transmit message signals $m(t)$ like this one:



The analog communication systems like AM and FM we transmit the whole signal. But in digital communications we only transmit some samples, which are the values of $m(t)$ at the *sample times* nT_s as follows:

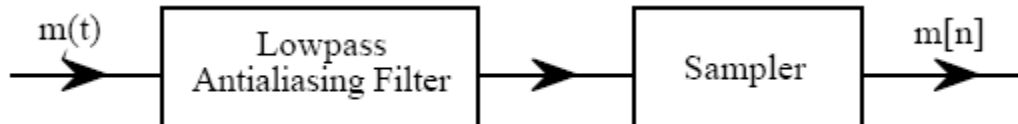


We know that we can reconstruct a band-limited signal $m(t)$ from its samples as long as the sampling rate f_s is fast enough to satisfy the Nyquist criterion:



If we don't sample fast enough we get **aliasing** - the distortion caused by not sampling fast enough. In PAM, we study how fast we can sample $m(t)$ and how to transmit it such that the receiver becomes able to reconstruct $m(t)$ from its samples $m[n] = m(nT_s)$.

In order to avoid aliasing due to high frequency noise or other high frequency interfering signals we typically lowpass filter $m(t)$ before sampling as follows



Another kind of pulse modulation is Pulse Width Modulation (**PWM**) - also referred to as Pulse Duration Modulation (**PDM**). In this scheme all the pulses are the same magnitudes but their length varies in proportion to amplitude of the sample. A disadvantage of PWM modulation is that we're using a lot of power when the pulses are long. One way to get around this problem is with Pulse Position Modulation (**PPM**) in which just a narrow pulse is transmitted at the end of each PWM pulse.

PCM is a method of converting an analog signal into a digital one for baseband transmission. In fact, PCM is a term which was formed during the development of digital audio transmission standards. PCM sequence is a quantized signal encoded into digital words. The quantization levels are separated by a step q which is given by:

$$q \approx V_{pp} / N$$

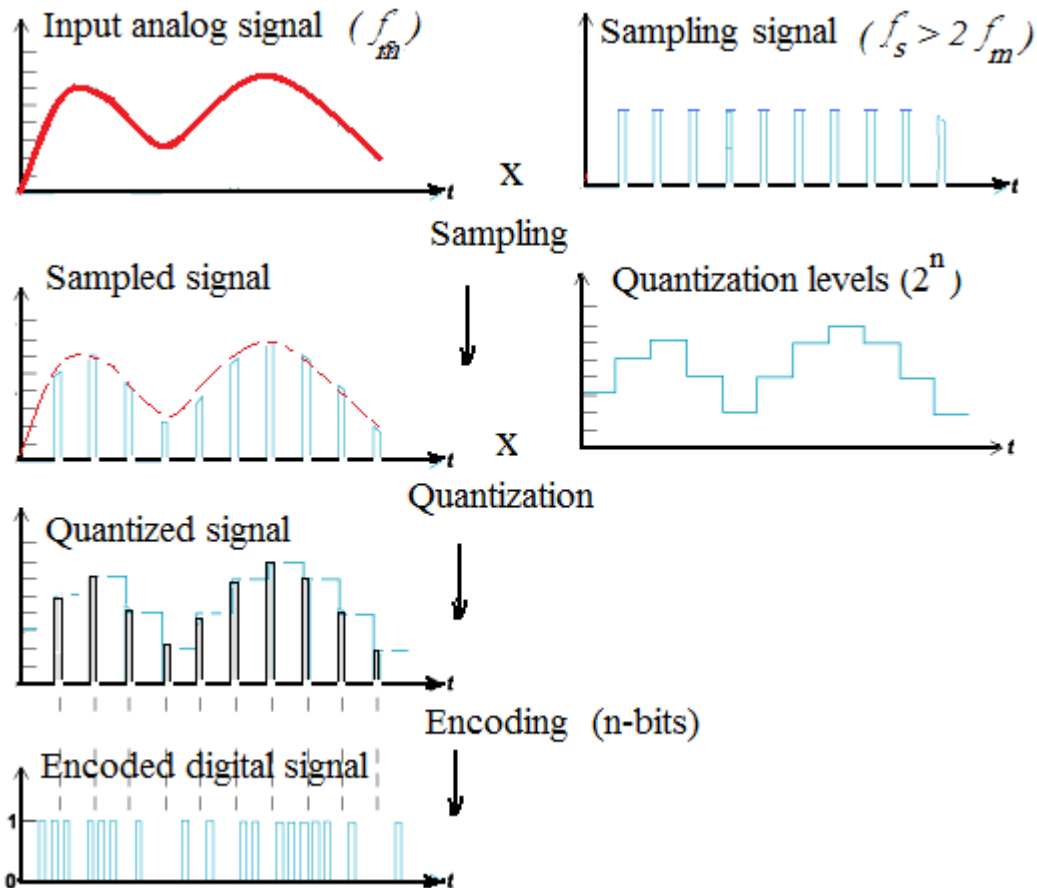
where V_{pp} is the peak-to-peak voltage of the input analog signal. The more quantization levels, N , the less quantization noise, but it requires more binary digits $n = \log_2 N$ to represent each sampled data. The root mean square quantization error (ϵ_{rms}) is hence given by:

$$\epsilon_{rms} (\text{quantization}) = q / \sqrt{12}$$

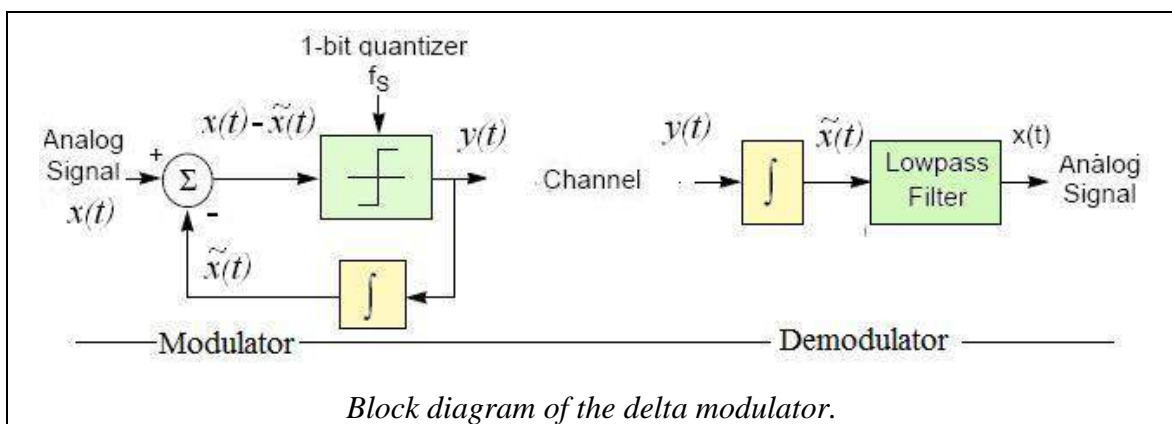
The signal-to-noise ratio (SNR), after uniform quantization, is given by:

$$SNR = (1/4 q^2 N^2) / (q^2 / 12) = 3 N^2$$

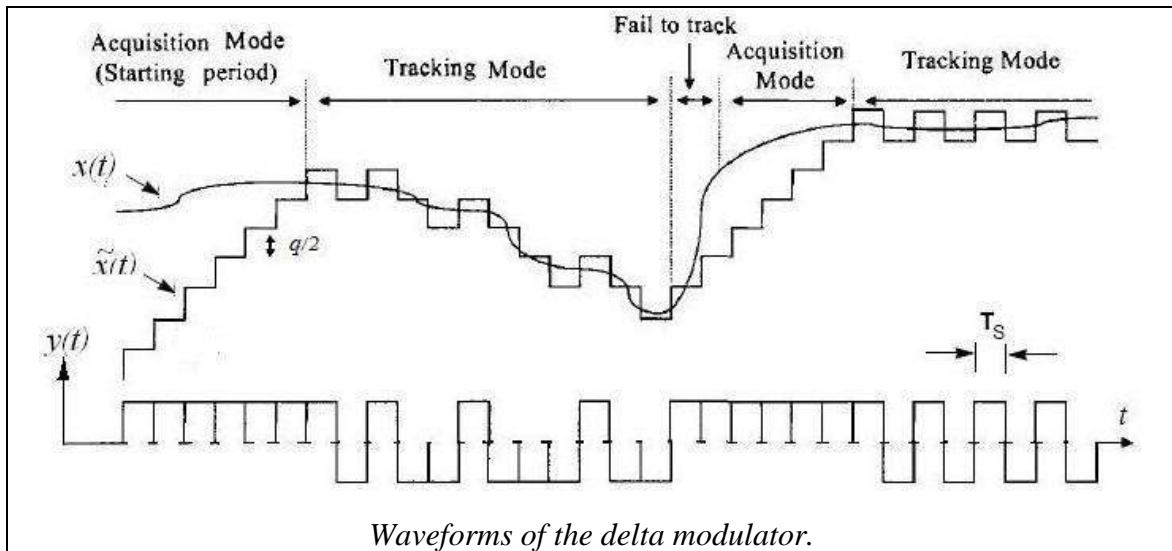
In telephony, PCM waveforms are called **line codes**. There are various PCM waveforms or line codes, such as NRZ, RZ, Manchester codes.



The Delta Modulator is a 1-bit ADC that was initially developed for digitizing analog signals in noisy environments. Its basic advantage was its simple circuitry. The block diagram of the Linear Delta Modulator is and its waveforms are shown in the figure below. The DM consists of a differential loop, where the difference $x(t) - \tilde{x}(t)$ is quantized to give the output sequence $y(t)$ so that the approximate signal $\tilde{x}(t)$ should track the input signal $x(t)$ all the time.

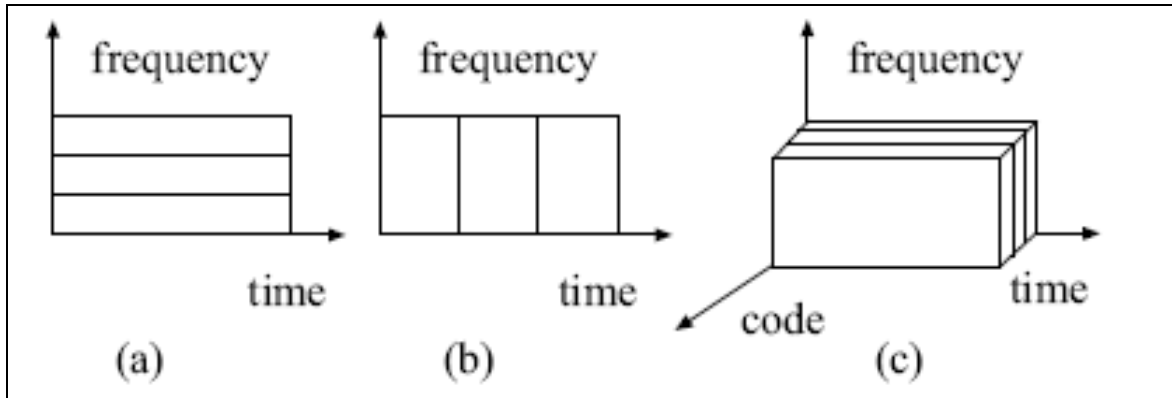


The **Delta-sigma** (Δ - Σ) is a 1-bit ADC that was initially developed for digitizing analog signals in noisy environments. The Δ - Σ modulation is a method for encoding high resolution signals into lower resolution signals using pulse-density modulation. Actually, Delta-sigma modulation technique is popularly used in analog-to-digital conversion (ADC) to provide high-resolution for narrow-band signals such as audio signals.



Even though the Σ - Δ technique was first presented in the early 1960s, it is only in recent years that it has come into widespread use with improvements in VLSI technology. Using sufficiently high over-sampling ratio, a Σ - Δ ADC is able to afford high digitization resolution. The above figure depicts the *delta* modulator from which the Σ - Δ circuit is derived.

Multiple access schemes can be classified into groups according to the nature of the protocol. The basic branches are contention-less (**scheduling**) and **contention** (random access) protocols. The **scheduling** protocols avoid the situation in which two or more users access the channel at the same time by scheduling the transmissions of the users. The fixed-assignment technique is used in frequency division multiple access (FDMA) and time division multiple access (TDMA). In FDMA systems, the total system bandwidth is divided into several frequency channels that are allocated to users. In TDMA systems, one channel is divided into time slots that are allocated to users, and the users only transmit during their assigned time slots. In CDMA, each user is assigned a distinct code sequence (spreading code) that is used to encode the user's signal. The receiver picks the desired signal using the same code sequence. The division of channels into time-frequency plane, FDMA (a), TDMA (b), and CDMA (c) is illustrated in the following figure.



The following list depicts the bit rate of different time-multiplexing techniques, for voice/data transmission.

- POTS 64Kbps
- ISDN 128Kbps
- ADSL 1.5 - 8Mbps/16-640Kbps (VDSL: up to 55Mbps)
- Cable Modem 40Mbps/20Mbps per channel
- DS1/T1 1.5Mbps (24 telephone lines)
- E1 2Mbps (32 lines)
- E_n $4n-1$ E1's
- STS-1 51.840Mbps
- STS- n $n \times$ STS-1 (also called OC- n)

In the context of a communication system that sends binary messages from the transmitter to the receiver across a noisy channel, a **matched filter** is employed to detect the transmitted pulses in the noisy received signal, with maximum output SNR. Therefore, we usually include a *matched filter* in the receiver. The **matched filter** is included before the signal processor. In fact, in some radars, the matched filter is the signal processor itself. The **impulse response** of the ideal matched filter, with white noise should be a **time-reversed complex-conjugated scaled version of the signal** that we are seeking. Although the use of matched filtering gives the optimum performance in the presence of AWGN, there is still a problem with using a rectangular pulse shape. Signals with discontinuities in their spectrum are physically unrealizable. Practical pulse shapes can be formed by sharpening the **roll-off** of the filter spectrum, for instance, by the so-called **cosine-raised filters**. The **cosine-raised filter** is a class of **Nyquist filters**. **Nyquist filters** have the property that their impulse response rings at the symbol rate.

In digital communication, **eye patterns** are widely used as a visual performance indicator of the system. An eye diagram is a plot/trace of consecutive sections of a signal superimposed on a normalized time scale.

3-11. Problems

3-1) An analog signal, which is band limited to f_m , is sampled at its Nyquist rate. and quantized using N quantization levels. The derived digital signal is then transmitted on some channel.

(a) Show that the bit time duration, T_b of one bit of the transmitted binary encoded signal must satisfy:

$$T_b f_m \log_2 N. \leq 1/2$$

(b) When is the equality sign valid?

3-2) How many bits to assign to each sampled and quantized data in PCM with objective to satisfy the following condition $|e| \leq p V_{pp}$ where V_{pp} is the peak-to-peak voltage and p is a fraction. If p should be less than 1%. how many bits per sample are needed in order to satisfy this condition?

3-3) The information in an analog waveform, with maximum frequency $f_m = 3$ kHz. is to be transmitted over an M -ary PAM system, where the number of pulse levels is $M=16$. The quantization distortion is specified not to exceed $\pm 1\%$ of the peak-to-peak analog signal.

(a) What is the minimum number of bits/sample, or bits/PCM word that should be used in digitizing the analog waveform?

(b) What is the minimum required sampling rate, and what is the resulting bit transmission rate?

(c) What is the PAM pulse or symbol transmission rate?

(d) If the transmission bandwidth (including filtering) equals 12 KHz, determine the bandwidth efficiency for this system.

3-4) A signal with the dynamic range of $[-5, 5]$ V is to be quantized. To achieve a peak signal-to-noise ratio of 30 dB for the quantized output, how many quantization levels are required and what is the step size?

3-5) Consider a speech signal of duration 1min, which has been digitalized by the PCM technique. If the sampling frequency was 16 kHz and the quantizer was uniform, with signal-to quantization noise ratio of at least 40dB. Calculate the bit rate and the minimum storage capacity (in kByte), which is needed to record this signal in a digital form.

3-6) Consider an analog signal of amplitude V_m , which is band-limited to f_m , and applied to a delta modulator with step size q . Prove that the oversampling distortion happen in the following condition:

$$V_m > q f_s / 2 \pi f_m.$$

3-7) Briefly describe and explain the operating principles and the advantageous properties of spread spectrum technique.

3-12. References

- [1] H. **Inose**, Y. Yasuda, J. Murakami. A Telemetering System by Code Manipulation Δ - Σ Modulation. IRE Trans on Space Electronics and Telemetry. pp. 204-209. Sep. 1962.
- [2] J. C. **Bellamy**. Digital Telephony. Wiley and Sons. New York. 1982.
- [3] H. **Taub** and D. L. **Schilling**. Principles of Communications Systems. 2nd Edition. McGraw Hill. 1986.
- [4] K. **Feher** *et al.*. Telecommunications Measurements. Analysis. and Instrumentation. Prentice-Hall. 1987.
- [5] W. **Chou**, T.H. Meng, R.M. Gray. "Time domain analysis of sigma delta modulation". *Acoustics, Speech, and Signal Processing*. No..3. pp. 1751-1754. 1990.
- [6] L. W. **Couch**. II. Digital and Analog Communication Systems. 6th Edition. Prentice Hall. 2001.
- [7] S. **Haykin**. Communication Systems. 4th Edition. J. Wiley & Sons. 2001.
- [8] John G. **Proakis**. "Digital Communications". Fourth Edition. McGraw Hill. pp 264-267. 2001.
- [9] **Bell** Telephone Laboratories. Transmission Systems for Communications. 5th Edition. 1982.
- [10] T.S. **Rappaport**. Wireless Communications: *Principles and practice*. Second Edition. Prentice Hall. 2006.
- [11] Thomas M. **Cover**. Data Compression. Chapter 5: *Elements of Information Theory*. John Wiley & Sons. 2006.

Chapter
4

Digital Baseband Modulation (Line Coding)

Contents

- 4-1. Introduction**
- 4-2. RZ Coding**
- 4-3. RZI Coding**
- 4-4. NRZ Coding**
- 4-5. NRZI Coding**
- 4-6. AMI Coding**
- 4-7. Manchester Coding**
- 4-8. Comparison between Different Line Codes**
- 4-9. Summary**
- 4-10. Problems**
- 4-11. Bibliography**

Chapter

4

Digital Baseband Modulation (Line Coding)

4-1. Introduction

Digital baseband **modulation** (or line coding) is the process of transfer of a digital bit stream over a lowpass channel, typically a copper wire such as a serial bus or a wired local area network.

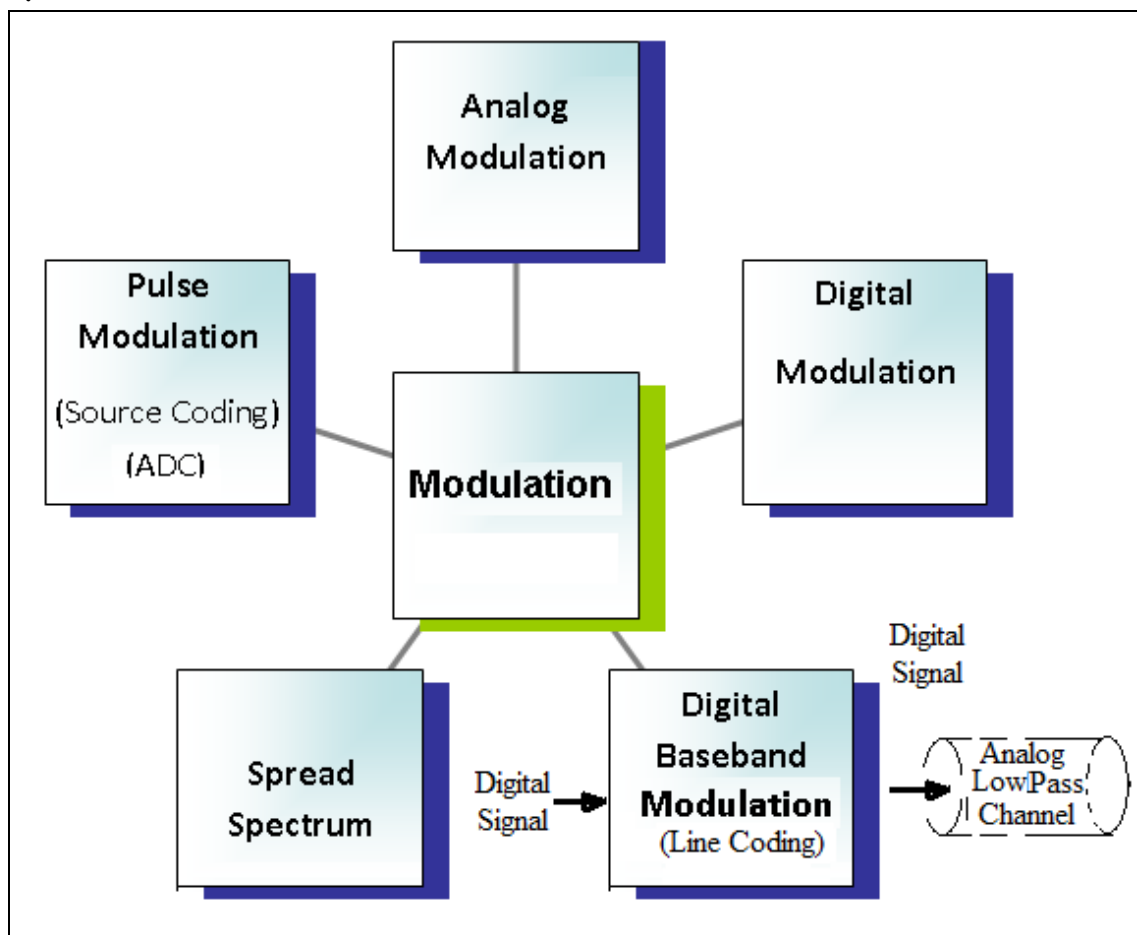


Fig. 4-1. Digital baseband modulation, among other modulation techniques.

After line coding, the signal is put through a physical channel, either a transmission medium or data storage medium. The most common physical channels and their respective applications are:

- line-coded signal can be converted to pits on optical disc (**CD**).
- line-coded signal can be converted to magnetized spots on a **HD** or magnetic tape.
- line-coded signal can be put on a transmission line, such as serial communication cables, in the form of variations of the voltage or current.
- line-coded signal can be used to switch light, **IR** remote control.
- line-coded signal can be printed on paper to create a **barcode**.
- line-coded signal (baseband signal) may undergo further pulse shaping and then used to modulate an RF signal and transmitted in air.

There are numerous ways digital signals can be coded to a transmission line. Some of the common are included in the following graph:

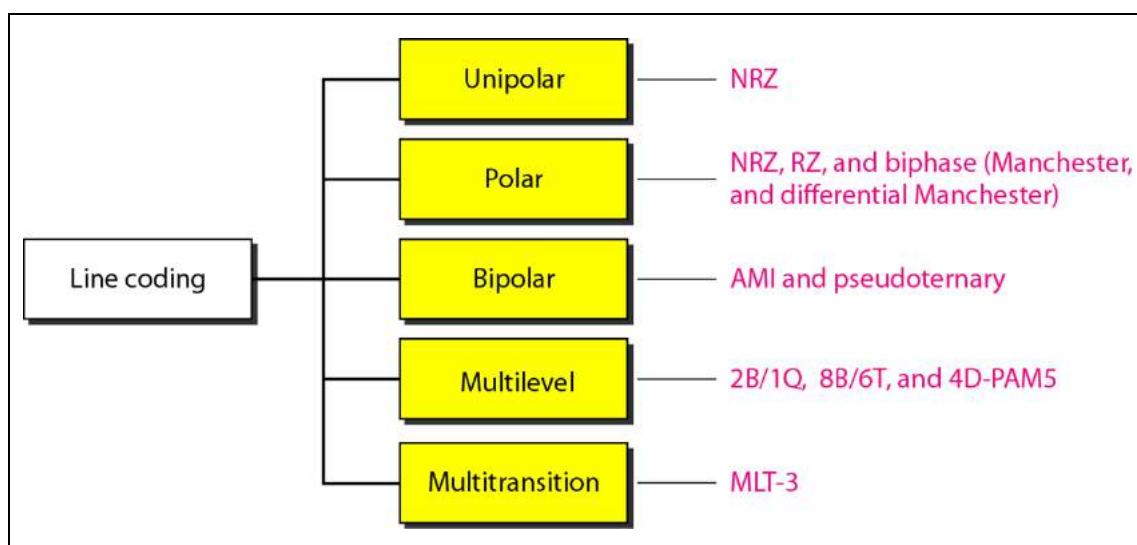


Fig. 4-2. Digital baseband modulation, among other modulation techniques.

The common examples of line codes are unipolar, polar, bipolar, and **Manchester** coding. The simplest possible line code, called **unipolar** because it has an unbounded DC component, gives too many errors on such systems. Unfortunately, most long-distance communication channels cannot transport a DC component, which may result from long sequences of 1's or unipolar pulses. In addition, long sequences of unipolar pulses may hinder the receiver to synchronize itself to the phase of the transmitted signal. Therefore, most line codes eliminate the DC component. In order to get rid of the DC component, there are 2 common techniques:

1- Using a **constant-weight code**. In this case, we design the transmitted code word such that every code word that contains some positive or negative levels also contains enough of the opposite levels. Example of such codes is the **Manchester code**.

2- Using a paired code. Here, we design the receiver such that every code word that averages to a negative level is paired with another code word that averages to a positive level. For example, bipolar encoding or alternate-mark inversion (AMI) codes.

4-2. Return-to-Zero (RZ) Coding

Return-to-zero (RZ) describes a line code used in telecommunications signals in which the signal drops (returns) to zero between each pulse. This takes place even if a number of consecutive 0's or 1's occur in the signal. The signal is self-clocking. This means that a separate clock does not need to be sent alongside the signal, but suffers from using twice the bandwidth to achieve the same data-rate as compared to non-return-to-zero format. The binary signal is encoded using rectangular pulse amplitude modulation with polar return-to-zero code as follows:

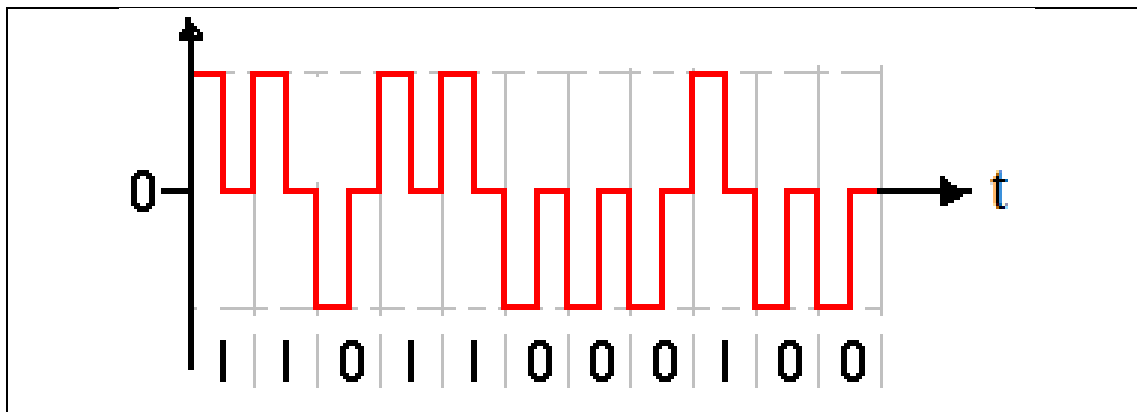


Fig. 4-3. Return-zero (RZ) encoding scheme.

The "zero" between each bit is a neutral or rest condition, such as a zero amplitude in pulse amplitude modulation (PAM), zero phase shift in phase-shift keying (PSK), or mid-frequency in frequency-shift keying (FSK). This "zero" condition is typically halfway between the significant condition representing a 1 bit and the other significant condition representing a 0 bit. Although return-to-zero (RZ) contains a provision for synchronization, it still has a DC average component resulting in "baseline wander" during long strings of 0 or 1 bits. This DC offset may cause errors in the receiver side.

4-3. Return to Zero, Inverted (RZI)

Return-to-zero inverted (RZI) is a method of mapping for transmission. The two-level RZI signal has a pulse (shorter than a clock cycle) if the binary signal is 0, and no pulse if the binary signal is 1. It is used by the IrDA Serial Infrared (SIR) physical layer specification.

4-4. Non-Return-to-Zero (NRZ) Coding

In telecommunication, a **non-return-to-zero (NRZ)** line code is a binary code in which "1s" are represented by one significant condition (usually a positive voltage) and "0s" are represented by some other significant condition (usually a negative voltage), with no other neutral or rest condition. The pulses have more energy than a RZ code. Unlike RZ, NRZ does not have a rest state. NRZ is not inherently a self-synchronizing code, so some additional synchronization technique (like run length limited constraint) must be used to avoid bit slip.

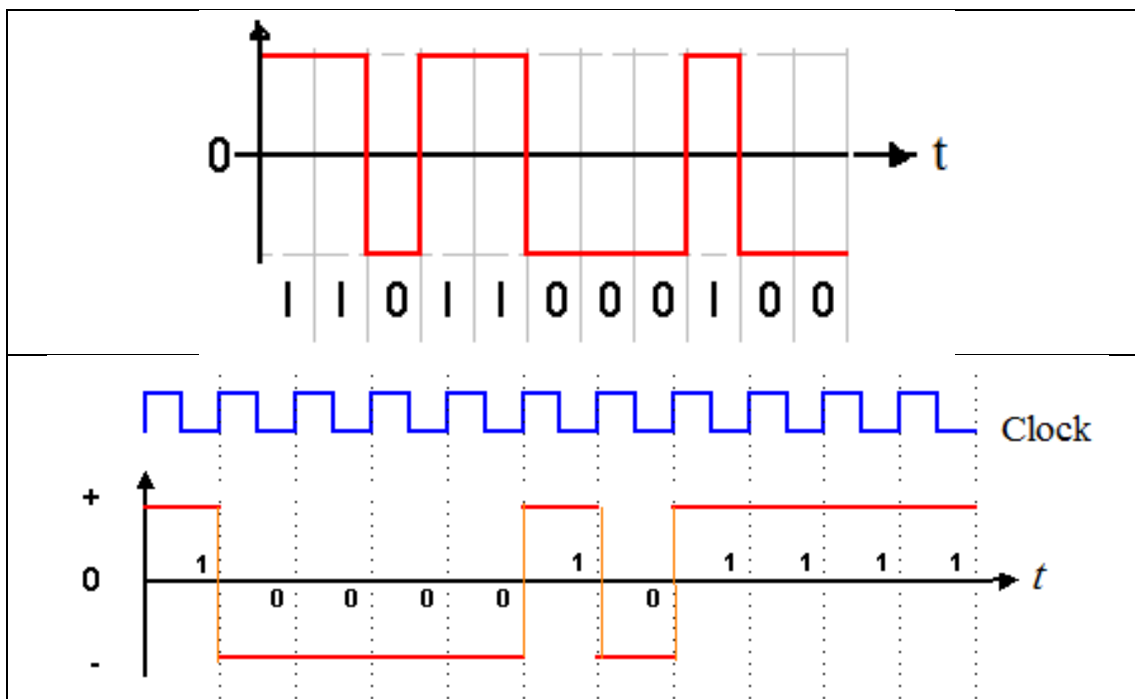


Fig. 4-4. Non-return-to-zero (NRZ) encoding scheme .

For a given data signaling rate, the NRZ code requires only half the bandwidth required by the Manchester code. When used to represent data in an asynchronous communication scheme, the absence of a neutral state requires other mechanisms for data recovery, to replace methods used for error detection when using synchronization information when a separate clock signal is available. NRZ itself is an encoding method that can be used in either a synchronous or asynchronous transmission environment. The real question is that of sampling--the high or low state will be received correctly when the physical line level is sampled at the receiving end. Actually, NRZ transitions as happening on the trailing (falling) clock edge. This shows the difference between NRZ with respect to other encoding methods, such as the Manchester code, which requires clock edge information. One of the most famous communication lines, which

have been adopting the NRZ coding is the serial lines (**RS-232**), which communicate data between computer and peripheral devices via the PC serial port. **The RS232 serial lines make use of** two voltage levels (+15V, -15V). Voltage never returns to 0V (0V means dead!). The bit rate of such protocol was Initially slow (19.2 kbps in RS-232-C) and raised later to about 128 kb/s. The serial line holds asynchronous serial data, one bit at a time, and the clock can be recovered from data. The data is sent asynchronously as characters of 7- or 8-bit, usually using the ASCII code. Framing is performed by start bit (15V for one clock), and stop bit (idle for one clock at -15V), for each character. Thus the total of 9 bits are sent for each character of 7 bits of data (78% efficient).

4-5. Non Return to Zero, Inverted (NRZI)

Non return to zero, inverted (**NRZI**) is a method of mapping a binary signal to a physical signal for transmission over some transmission media. The two level NRZI signal has a transition at a clock boundary if the bit being transmitted is a logical one, and does not have a transition if the bit being transmitted is a logical zero. "One" is represented by a transition of the physical level. "Zero" has no transition. Also, NRZI might take the opposite convention, as in Universal Serial Bus (USB) signaling, when in Mode 1 (transition when signaling zero and steady level when signaling one). The transition occurs on the leading edge of the clock for the given bit. This distinguishes NRZI from NRZ-Mark.

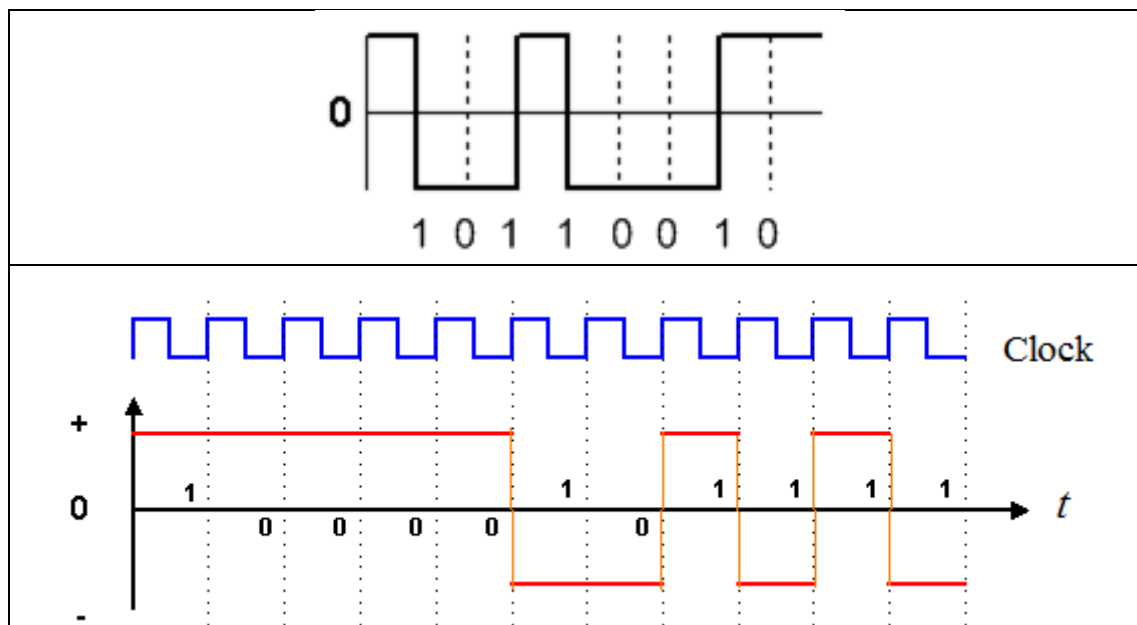


Fig. 4-5. NRZI encoding scheme.

4-6. Alternate Mark Inversion (AMI)

In telecommunication, **bipolar encoding** is a type of line code (a method of encoding digital information to make it resistant to certain forms of signal loss during transmission). Bipolar encoding is preferable to non-return-to-zero where signal transitions are required to maintain synchronization between the transmitter and receiver. When used on a T-carrier, the code is known as Alternate Mark Inversion (**AMI**) because, in this context, a binary 1 is referred to as a "**mark**", while a binary 0 is called a "**space**". The AMI coding was used extensively in first-generation PCM networks, and is still commonly seen on older multiplexing equipment today, but successful transmission relies on no long runs of zeroes being present.

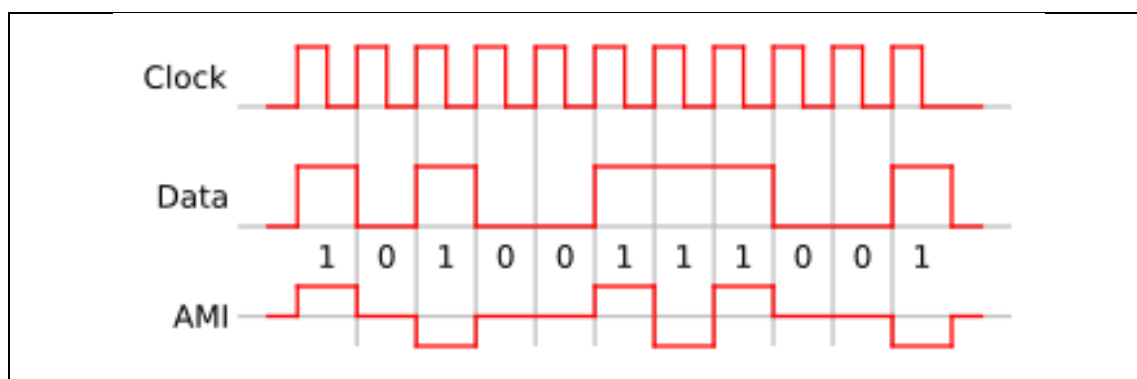


Fig. 4-6. AMI encoding scheme.

No more than 15 consecutive zeros should ever be sent to ensure synchronization. The modification of bit 7 causes a change to voice that is undetectable by the human ear, but it is an unacceptable corruption of a data stream. Data channels are required to use some other form of pulse-stuffing, such as always setting bit 8 to 1, in order to maintain one's density. If the characteristics of the input data do not follow the pattern that every eighth bit is 1, the coder using alternate mark inversion adds a 1 after seven consecutive zeros to maintain synchronization. On the decoder side, this extra 1 added by the coder is removed, resulting that the correct data arrives for the receiver. Due to this, the data sent between the coder and the decoder is longer than the original data by less than 1% in average. Of course, this lowers the effective data throughput to 56 kbit/s per channel

4-7. Manchester Encoding

The **Manchester** code or phase encoding (**PE**) is a line code in which the encoding of each data bit has at least one transition and occupies the same

time. It is, therefore, self-clocking, which means that a clock signal can be recovered from the encoded data. Manchester code is widely-used (e.g. in Ethernet).

As shown in figure 4-7, there are two opposing conventions for the representations of data. The first of these was first published by G. E. Thomas in 1949 and is followed by numerous authors. It specifies that for a 0 bit the signal levels will be Low-High ($0 \rightarrow 1 = 0$) - with a low level in the first half of the bit period, and a high level in the second half. For a 1 bit the signal levels will be High-Low ($0 \rightarrow 1 = 1$).

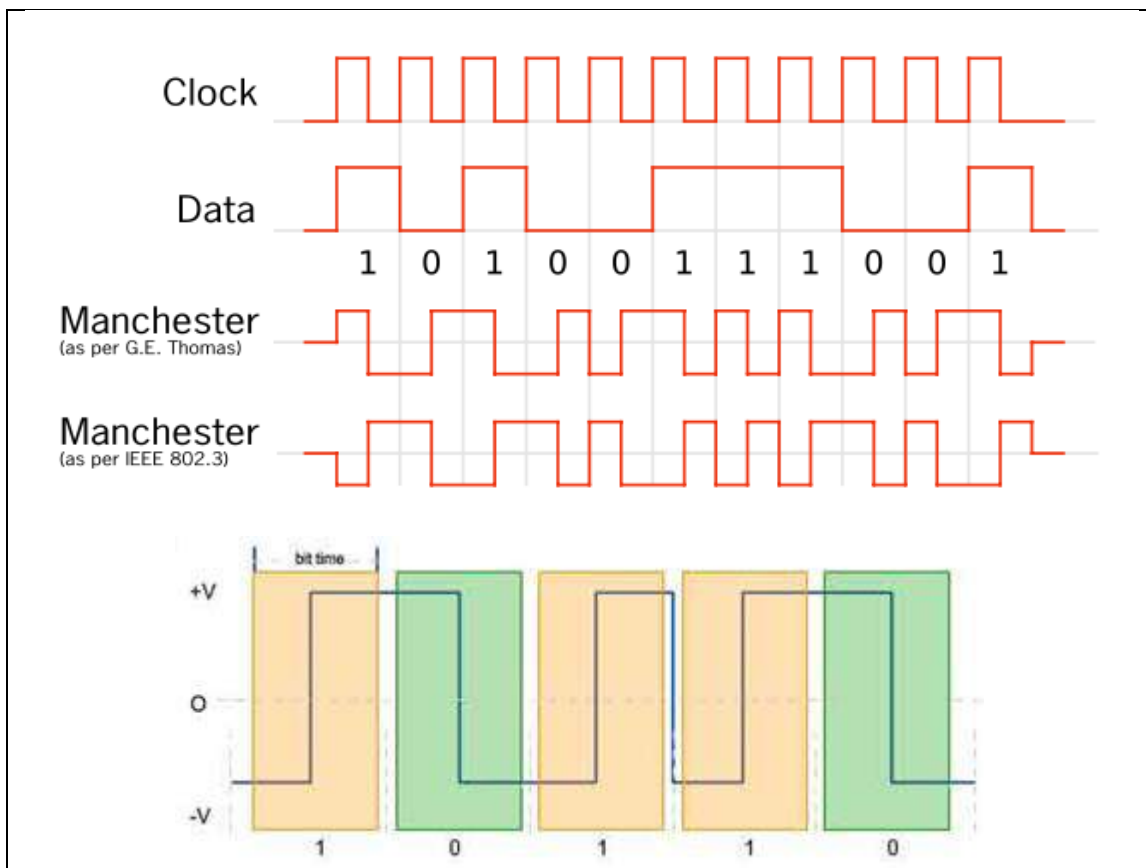


Fig. 4-6. *Manchester coding schemes.*

The second convention is followed by IEEE 802.4 (token bus) and lower speed versions of IEEE 802.3 (Ethernet) standards. It states that a logic 0 is represented by a High-Low transition ($1 \rightarrow 0 = 0$) and a logic 1 is represented by a Low-High transition ($0 \rightarrow 1 = 1$). Manchester code ensures frequent line voltage transitions, directly proportional to the clock rate. This helps clock recovery. The DC component of the encoded signal is not dependent on the data and therefore carries no information, allowing the signal to be conveyed conveniently by media (e.g. Ethernet) which

usually do not convey a DC component. Therefore the Manchester codes have the following features:

- Each bit is transmitted in a fixed time (period).
- A **0** is expressed by a low-to-high transition, a **1** by high-to-low transition (according to G.E. Thomas' convention -- in the IEEE 802.3 convention, the reverse is true).
- The transitions which signify **0** or **1** occur at the midpoint of a period.
- Transitions at the start of a period are overhead and don't signify data.

The Manchester code always has a transition at the middle of each bit period and may (depending on the information to be transmitted) have a transition at the start of the period also. The direction of the mid-bit transition indicates the data. Transitions at the period boundaries do not carry information. They exist only to place the signal in the correct state to allow the mid-bit transition. Although this allows the signal to be self-clocking, it doubles the bandwidth requirement compared to NRZ coding schemes (or see also NRZI). In the Thomas convention, the result is that the first half of a bit period matches the information bit and the second half is its complement. Note that Manchester encoding is a special case of binary phase-shift keying (**BPSK**), where the data controls the phase of a square wave carrier whose frequency is the data rate. Such a signal is easy to generate. In order to control the bandwidth used, a filter can reduce the bandwidth to as low as 1Hz per bit/second without loss of information in transmission. In radio transmission, the encoded signal may also be modulated with a carrier wave; however, the property of 1Hz per bit/second is preserved.

4-8. Comparison between Different Line Codes

The following figure summarizes some line coding methods, as compared to the raw TTL binary data. The word *mark*, and its converse *space*, often appear in a description of a binary waveform. This is a historical reference to the mark and space of the telegraph. In modern terminology these have become Hi and Low, or '1' and '0', as appropriate. Note that unipolar signaling means a '1' is represented with a finite voltage V volts, and a '0' with zero voltage. Also, polar signaling: means a '1' is represented with a finite voltage $+V$ volts, and a '0' with $-V$ volts. In addition, bipolar signaling: means a '1' is represented alternately by $+V$ and $-V$, and a '0' by zero voltage. It should be also noted that Di-code non-return to zero (*4-level*) means that for each transition of the input there is an output pulse, of opposite polarity from the preceding pulse. For no transition between input pulses there is no output.

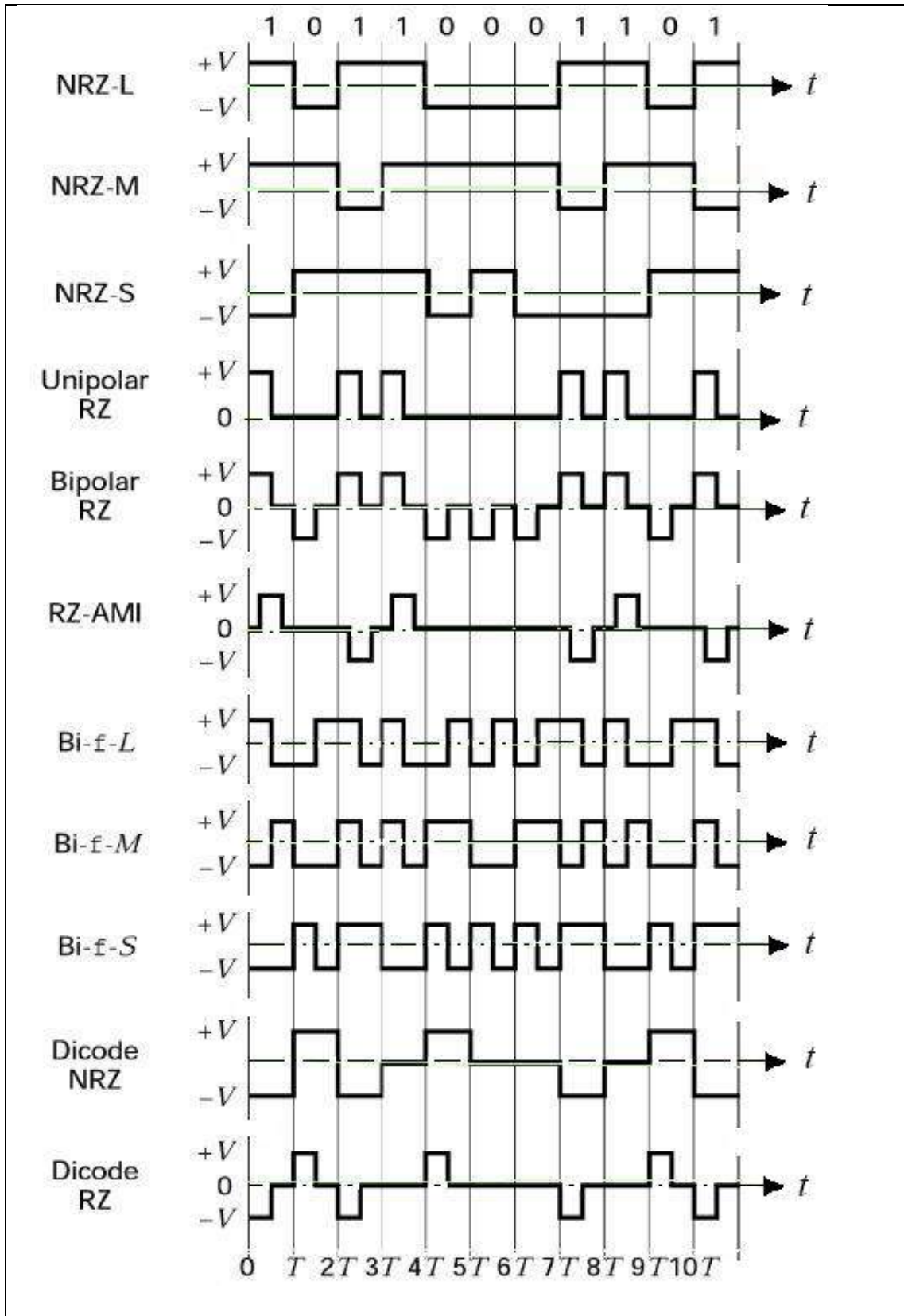


Fig. 4-8. Various encoding methods for binary waveforms.

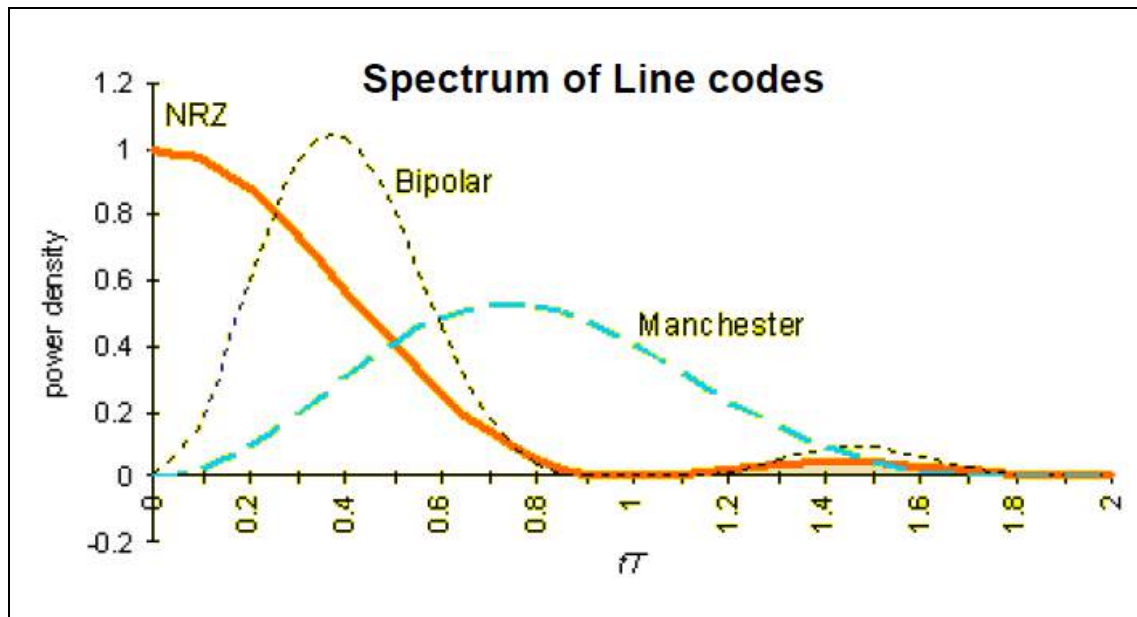


Fig. 4-9. Power spectral density (PSD) of line codes

- The PSD of the transmitted signal should be compatible with the channel frequency response
- Unipolar most of signal power is centered around origin and there is waste of power due to DC component that is present.
- Polar format most of signal power is centered around origin and they are simple to implement.
- Bipolar format does not have DC component and does not demand more bandwidth, but power requirement is double than other formats.
- Manchester format does not have DC component but provides proper clocking.
- Many channels cannot pass dc (zero frequency) owing to ac coupling
- Low pass response limits the ability to carry high frequencies

Table 4-1. Comparison between different line codes.

CODE	BANDWIDTH	TIMING	DC VALUE
Unipolar NRZ	Low bandwidth	No timing information	High DC component
Bipolar NRZ	Lower bandwidth	No timing information	No DC component
Differential NRZ	Lower bandwidth	No timing information	Little or no DC component
Manchester	High bandwidth	Good clock recovery	No DC component
Differential Manchester	Moderate bandwidth	Good clock recovery	No DC Component

4-9. Summary

In addition to pulse shaping, there are a number of useful line codes that we can use to help reduce our errors or to have other positive effects on a digital signal. The term digital baseband modulation is synonymous to line codes. Line codes are methods to transfer a digital bit stream over an analog lowpass channel using a discrete number of signal levels, by modulating a pulse train (a square wave instead of a sinusoidal waveform). Line codes are used commonly in computer communication networks over short distances. There are numerous ways digital signals can be coded to a transmission line.

The simplest form of clock encoding and extraction is achieved through a **bipolar encoding** technique, where binary 1's are represented by a positive voltage and binary 0's are represented by a negative voltage. Between each pair of binary bits, the voltage waveform is at zero volts. This is referred to as Return-to-Zero (**RZ**) waveform.

Another, slightly more complex technique, known as the Non Return to Zero (**NRZ**), produces a waveform with the normal, two voltage levels. It is sometimes referred to as **Manchester** encoding technique. However, the width of pulses in the Manchester scheme varies depending on whether successive bits are the same or not. This provides a mechanism for "extracting" clock information at the receiver.

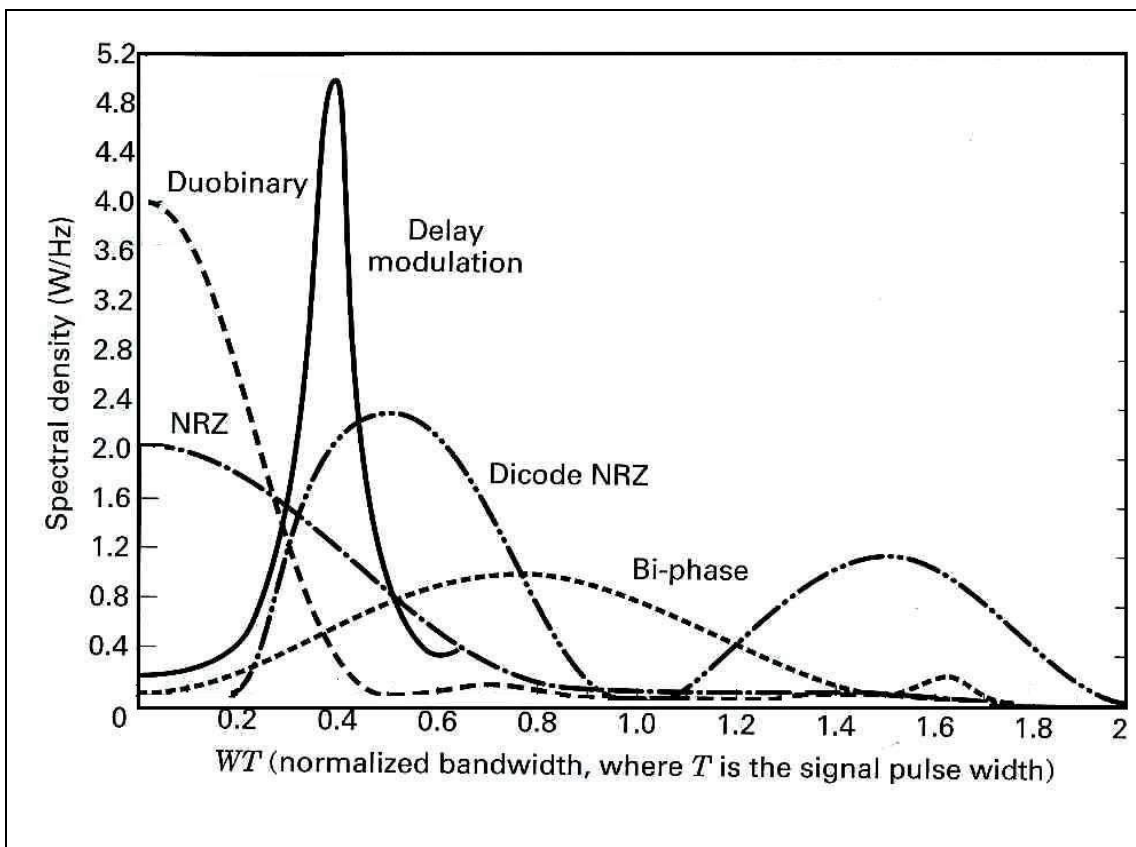
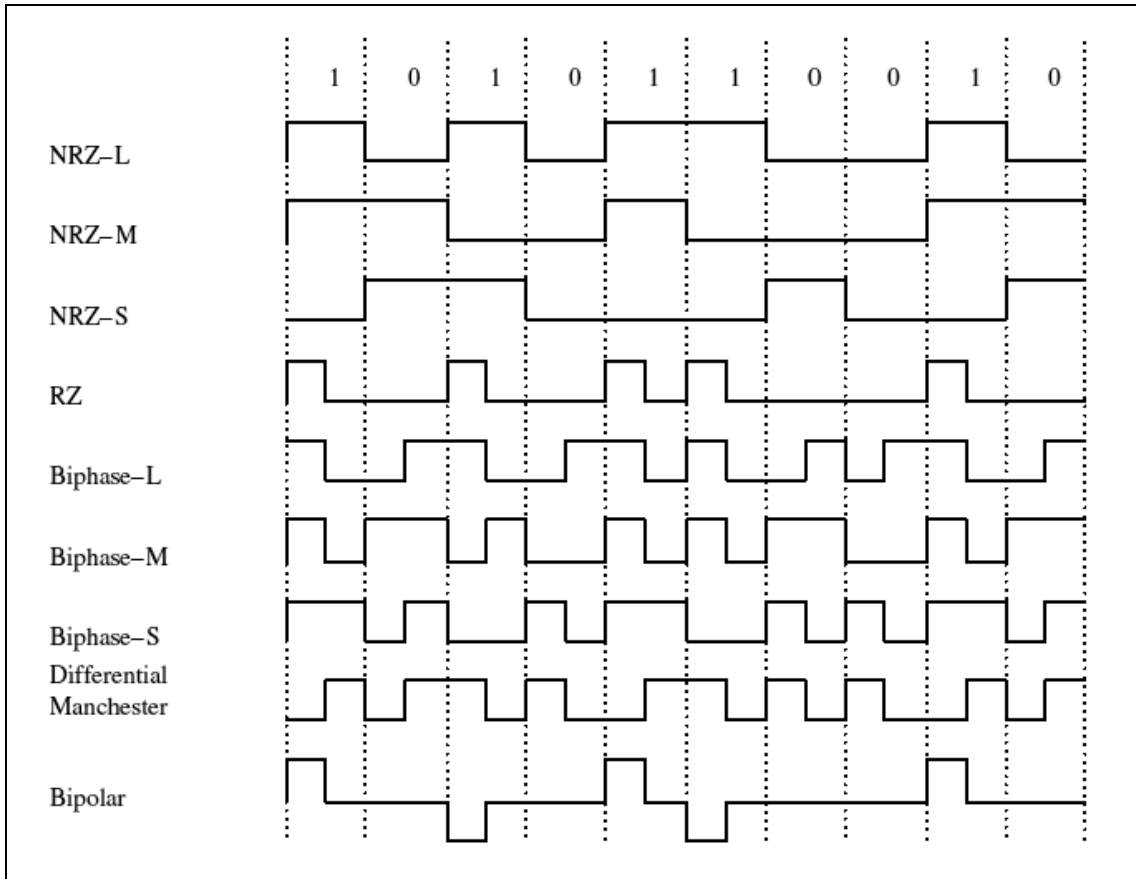
If clocking information is not embedded into the transmitted data signal, there is a possibility that the receiver clock will allow it to drift out of synchronism whenever a long stream of zeros and ones are received. Therefore, the only other means by which a receiving device can synchronize itself to a transmitter is to always ensure that there are enough transitions ($0 \rightarrow 1$ and $1 \rightarrow 0$) in the data signal. This means that the transmitted data must be encoded in a special format. This encoding may be a Non Return to Zero Inverted (**NRZI**) technique or a differential encoding technique. For a NRZ system to be synchronized there must not be long sequences of identical symbols, such as ones or zeroes. The use of 'L' (level) and 'M' (mark) with NRZ would seem to be somewhat illogical or inconsistent with each other.

Signal	Comments
NRZ-L	Non-return to zero level. This is the standard positive logic signal format used in digital circuits. 1 forces a high level 0 forces a low level
NRZ-M	Non return to zero mark 1 forces a transition 0 does nothing
NRZ-S	Non return to zero space 1 does nothing 0 forces a transition
RZ	Return to zero 1 goes high for half the bit period 0 does nothing
Biphase-L	Manchester. Two consecutive bits of the same type force a transition at the beginning of a bit period. 1 forces a negative transition in the middle of the bit 0 forces a positive transition in the middle of the bit
Biphase-M	There is always a transition at the beginning of a bit period. 1 forces a transition in the middle of the bit 0 does nothing
Biphase-S	There is always a transition at the beginning of a bit period. 1 does nothing 0 forces a transition in the middle of the bit
Differential Manchester	There is always a transition in the middle of a bit period. 1 does nothing 0 forces a transition at the beginning of the bit
Bipolar	The positive and negative pulses alternate. 1 forces a positive or negative pulse for half the bit period 0 does nothing

Each of the various line formats has a particular advantage and disadvantage. It is not possible to select one, which will meet all needs. The format may be selected to meet one or more of the following criteria:

- Minimize transmission hardware
- Facilitate synchronization
- Ease error detection and correction
- Minimize spectral content
- Eliminate DC component

The Manchester code is quite popular. It is known as a self-clocking code because there is always a transition during the bit interval. Consequently, long strings of zeros or ones do not cause clocking problems.



4-10. Problems

4-1) Enumerate the unipolar and bipolar coding schemes, which are used in channel coding.

4-2) What are the main differences between alternate mark inversion (AMI) and non-return-to-zero (NRZ) coding schemes?

4-3) An important function of many line encoders is the elimination of the DC component. When is this desirable ?

4-4) What is the category of data transmission if the binary pulse is maintained for the entire bit time?

- a. Return to zero
- b. Bipolar
- c. Unipolar
- d. Non

4-5) Unipolar, bipolar, and polar encoding are types of _____ encoding.

- a. Line
- b. Block
- c. NRZ
- d. Manchester

4-6) _____ encoding has a transition at the middle of each bit.

- a. RZ
- b. Manchester
- c. Differential Manchester
- d. All the above

4-7) Which of the following encoding methods does not provide for synchronization?

- a. NRZ-1
- b. Manchester
- c. RZ
- d. AMI

4-11. References.

- [1] Simon **Haykin**,. *Digital Communications*. Toronto, Canada: John Wiley & Sons, **1988**
- [2] John G. **Proakis**,. *Digital Communications*. Singapore: McGraw Hill., **1995**
- [3] W. Leon **Couch**,. *Digital and Analog Communications*. Upper Saddle River, NJ: Prentice-Hall, **1997**.
- [4] B.P. **Lathi** , Modern digital and analog communication systems, **1998**
- [5] Bernard **Sklar** , Digital communication fundamentals and applications, **2000**
- [6] William **Stallings**, Data and Computer Communications (7th ed.), Prentice Hall, **2004**.
- [7] M. **Thomas** and A. Thomas Joy. *Elements of information theory*, 1st Edition. New York: Wiley-Interscience, 1991. 2nd Edition. New York: Wiley-Interscience, **2006**

A square box with a thick black border containing the word "Chapter" in a serif font above a large, bold, black number "5".

Digital Modulation Methods

Contents

- 5-1. Introduction**
- 5-2. Amplitude Shift Keying (ASK)**
- 5-3. Frequency Shift Keying (FSK)**
- 5-4. Phase Shift Keying (PSK)**
- 5-5. Binary Phase Shift Keying (BPSK)**
 - 5-5.1. BPSK Constellation Diagram
 - 5-5.2. BPSK Signals
 - 5-5.3. Power Spectral Density of BPSK Signals
 - 5-5.4. Implementation of BPSK
 - 5-5.5. Costas Loop
 - 5-5.6. Practical Considerations of BPSK
- 5-6. Quadrature Phase Shift Keying (QPSK)**
 - 5-6.1. QPSK Constellation Diagram
 - 5-6.2. QPSK Signals in Time Domain
 - 5-6.3. Power Spectral density of QPSK Signals
 - 5-6.4. Implementation of QPSK
- 5-7. Differential Encoding Modulation (DBPSK, DQPSK)**
- 5-8. Quadrature Amplitude Modulation (QAM)**
 - 5-8.1. QAM Constellation Diagram
 - 5-8.2. QAM in the Time Domain
 - 5-8.3. Implementation of QAM
- 5-9. Orthogonal FDM (OFDM)**
- 5-10. Continuous Phase Modulation (CPM)**
- 5-11. Spread Spectrum Techniques**
- 5-12. Trellis Code Modulation (TCM)**
- 5-13. Summary**
- 5-14. Problems**
- 5-15. Bibliography**

Chapter 5

Digital Modulation Methods

5-1. Introduction

Communication system design requires the simultaneous conservation of bandwidth, power, and cost. In the past, it was possible to make a low cost radio by sacrificing parameters such as power and bandwidth efficiency. Over the past few years a major transition has occurred from simple analog Amplitude Modulation (AM) and Frequency/Phase Modulation (FM/PM) to new digital modulation techniques. The move to digital modulation provides high noise immunity, more information capacity, better quality communications and compatibility with digital circuits, as well as quicker system availability. When it is required to transmit digital data over a band-pass channel, we use the input data to modulate a carrier wave (sinusoidal) with the frequency limits imposed by the channel. This process is called **digital modulation**. The objective of this chapter is to introduce the key characteristics and salient features of the main digital modulation schemes.

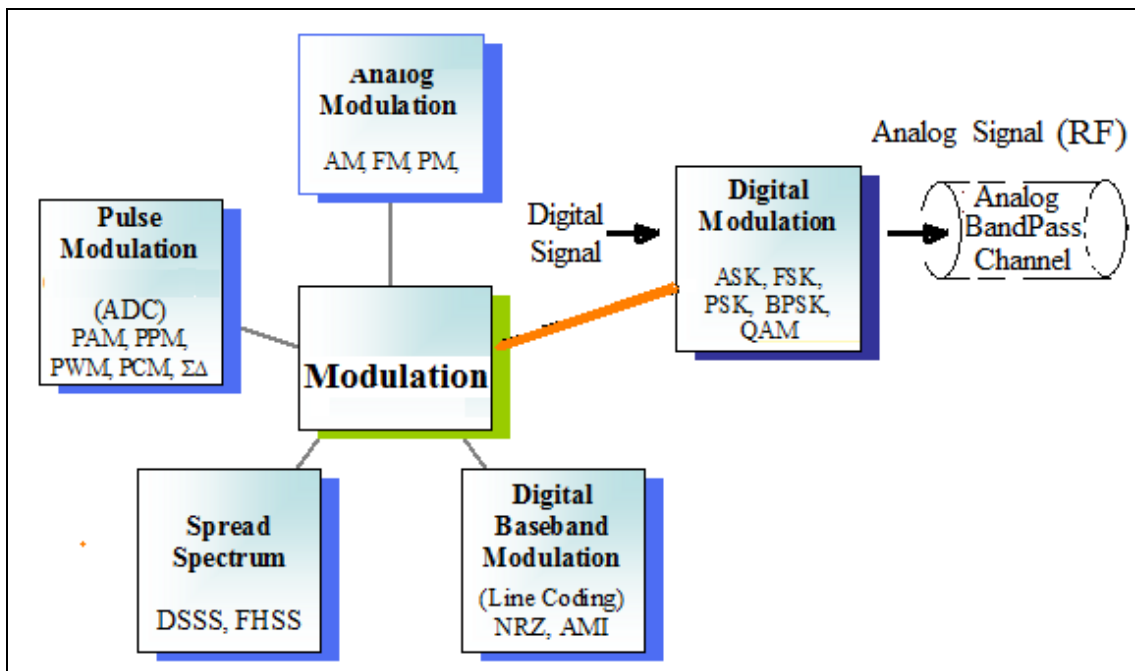


Fig. 5-1. Digital modulation, among other modulation techniques.

Figure 5-2 shows the block diagram of a wireless digital communication system. In digital systems, the input signal is converted into binary digits by an analog-to-digital converter (ADC). Ideally, we like to represent the source by as few binary digits (bits) as possible. The process of converting the source into a compressed sequence of bits is called **source encoding** or **data compression**. The sequence of binary digits from the **source encoder** is passed to the **channel encoder**. The purpose of the channel encoder is to introduce some additional bits (redundancy) in the information sequence that can be used at the receiver to overcome the effect of noise and interference. After channel coding, in the transmitter side, the binary sequence is passed to a **digital modulator** which converts the binary sequence into high frequency signals that can be transmitted on the channel.

The communication channel is the physical medium that is used for transmitting signals from transmitter to receiver. In wireless system, the channel consists of atmosphere. However, in other systems, the channel may be copper wires or coaxial cables or optical fibers. In the receiver side, the **digital demodulator** processes the transmitted waveform, which has been corrupted through the channel, and reduces the waveform to the binary sequence that represents the transmitted data. This sequence of numbers then passed through the **channel decoder** which attempts to reconstruct the original information sequence. At the end, the **source decoder** tries to decode the sequence from the knowledge of the source encoding algorithm.

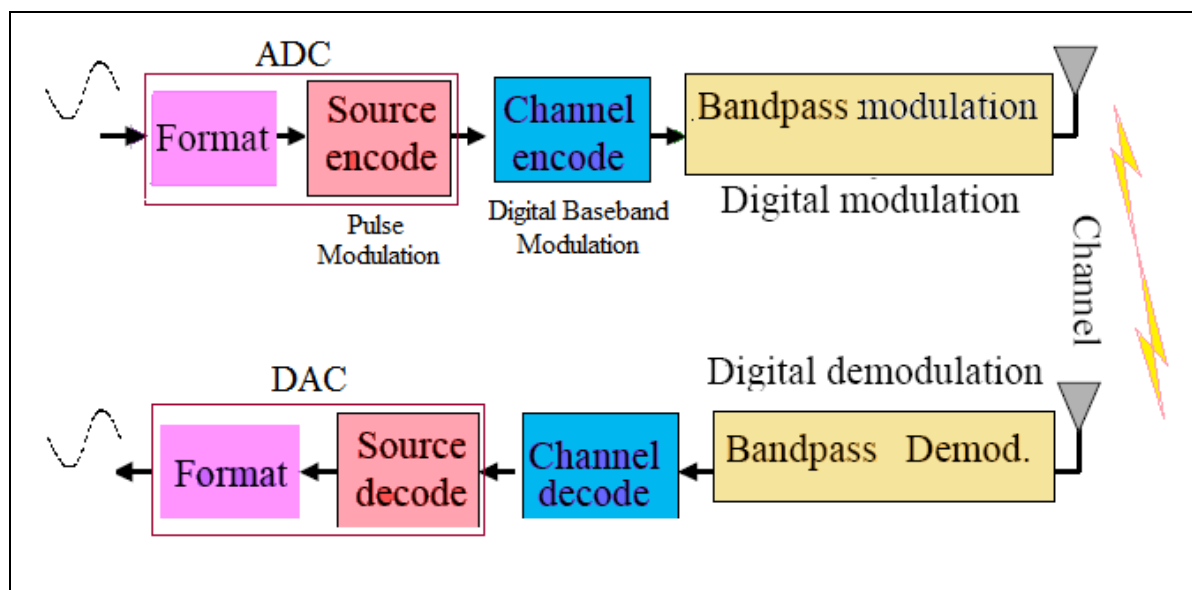


Fig. 5-2. Block diagram of a wireless digital communication system

The simplest form of digital modulation is On-Off keying (OOK). OOK is represented by a series of '1's and '0's by simply switching On and Off a sinusoidal carrier wave. An example of this is shown in Figure 5-3(a). Here a '0' is represented by having the carrier 'Off' (i.e. reducing its amplitude to zero), and a '1' by having the carrier 'On' (i.e. giving it a chosen amplitude, A). Thus the example shown in Fig 5-2(a) shows what we might get if we use this simple On/Off binary method to send a series of bits '010110'.

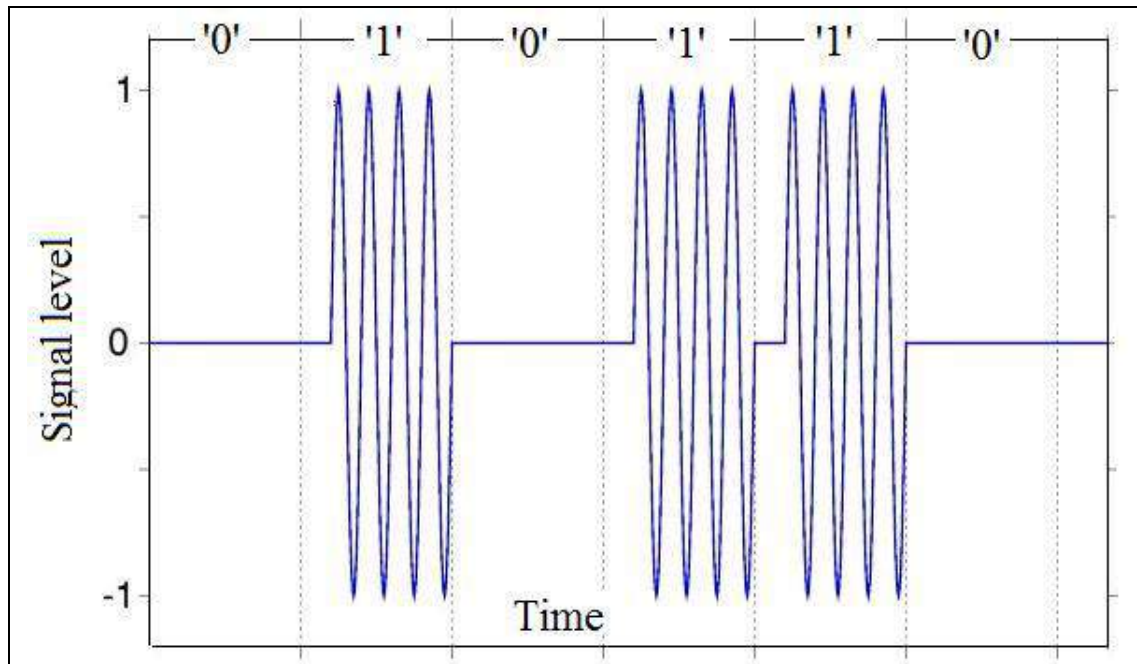


Fig. 5-3(a). Simple binary keying

The signal pattern used during each time chunk is called a **symbol** and represents some information that is being communicated by switching on or off the signal, as appropriate.

Figure 5-3(b) zooms in the above signal pattern so we can see some of the details more clearly. We can see that – when it is non-zero – each symbol consists of an integer number (4, in this example) cycles of the chosen carrier frequency. Hence if we were using a carrier frequency of, say, 10 MHz, then in this example each symbol would have a duration of $4/10^7 = 0.4 \mu\text{s}$. The length of each symbol is called the *Symbol Duration*, T_s . In between successive symbols we may also have a short 'spacing' period called the *Guard Interval*, T_G , which more clearly separates one symbol from the next. We can represent the On/Off modulation in algebraic terms by saying that the waveform consists of a signal

$$S(t) = A_c \sin(2\pi f_c t + \phi) \quad (5-1)$$

where we set $A_c=A$, $\phi=0$ when the sent bit is 1 and $A_c=0$, $\phi=0$ when it is 0. The typical output waveform of OOK is shown below in Figure 5-2(b). For the sake of simplicity, we have chosen $A=1$. Note that T_s is the symbol time and T_G is the guard time, between successive symbols.

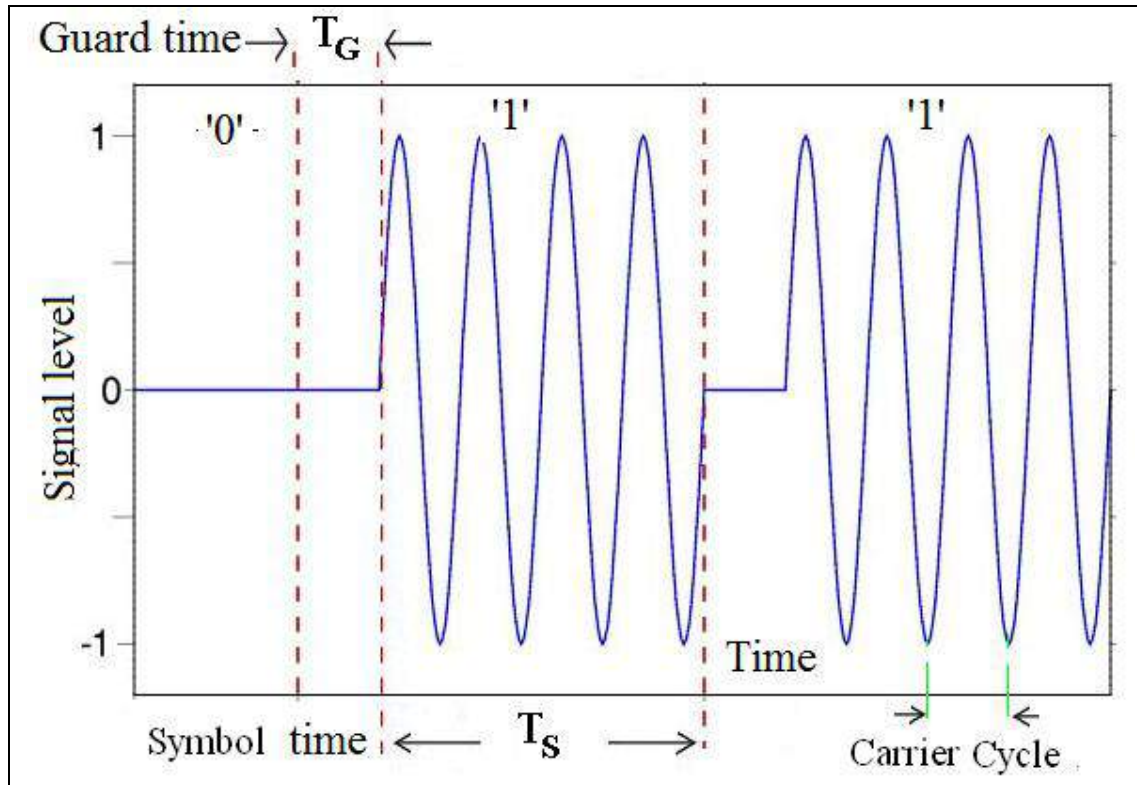


Fig. 5-3(b). Symbol and guard durations

The above mentioned OOK technique is sometimes called amplitude shift keying (ASK). Generally speaking, there are three major classes of **digital modulation** techniques used for transmission of digitally represented data:

- Amplitude-shift keying (**ASK**)
- Frequency-shift keying (**FSK**)
- Phase-shift keying (**PSK**)

The term *keying* goes back to the days of Morse Code when transmissions were switched on and off by hand, using a switch (or a key). All the shift-keying methods convey data by changing some aspect of a base carrier signal, which is usually a sinusoid, in response to a data signal.

Note 5-1: Digital Modulation versus Pulse Modulation Methods

Digital modulation should be distinguished from Pulse modulation (PM) schemes. Pulse modulation techniques aim at transferring a narrowband analog signal over an analog lowpass channel as a two-level quantized signal, by modulating a pulse train.

5-2. Amplitude-Shift Keying (ASK)

In ASK the amplitude of the carrier assumes one of the two amplitudes dependent on the logic states of the input bit stream.

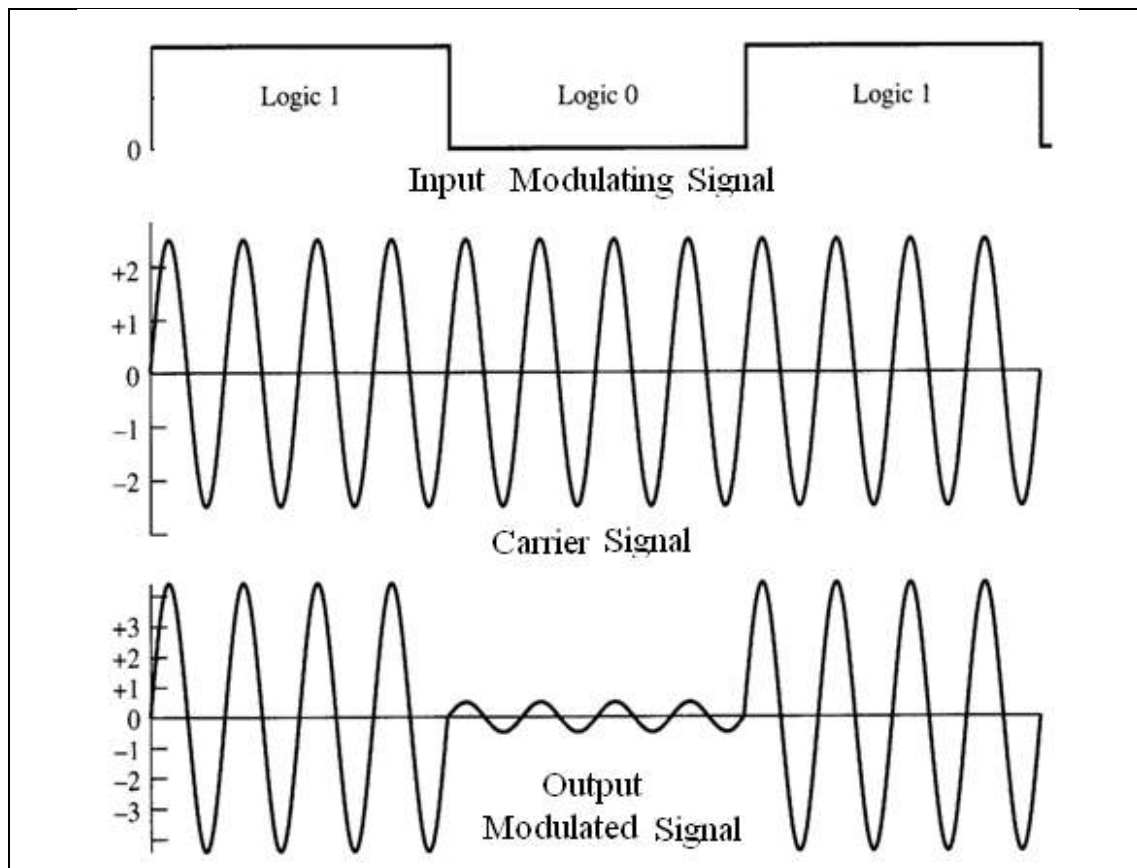


Fig. 5-4. Amplitude shift keying (ASK).

ASK and on-off keying (OOK) receivers are used for intermittent low-data-rate applications like remote controls, home security, and garage-door openers. The ASK technique is also commonly used to transmit digital data over optical fiber. Both ASK modulation and demodulation processes are relatively inexpensive. The following figure depicts a complete ASK transceiver (transmitter and receiver) system. It should be noted that the ASK demodulation process can be simply achieved by an envelope detector, in much the same way as the demodulation of AM signals.

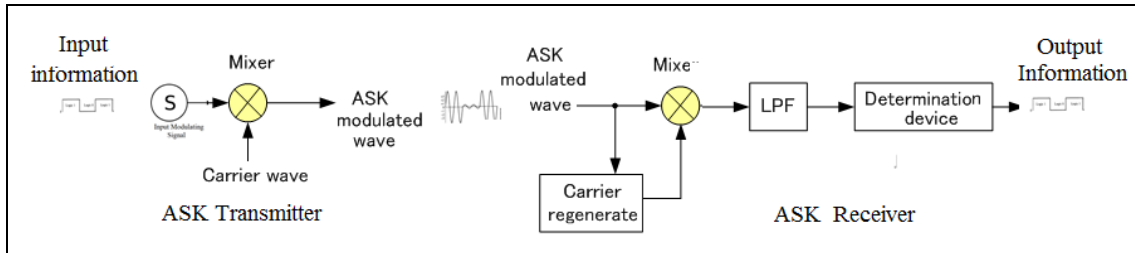


Fig. 5-5. Block diagram of ASK transceiver

5-3. Frequency-Shift Keying (FSK)

The Frequency-Shift keying (FSK) is a digital modulation technique in which digital information is transmitted through discrete frequency changes of a carrier wave. The most common form of frequency shift keying is binary FSK (BFSK or 2-FSK). The BFSK uses two discrete frequencies to transmit binary information (0's and 1's). The following figure depicts the FSK signals in the time domain. The modulated binary FSK signals, which represent the '1' and '0' input symbols, can be represented as follows:

$$s_1(t) = A_c \sin(2\pi f_1 t) \quad (5-2a)$$

$$s_2(t) = A_c \sin(2\pi f_2 t) \quad (5-2b)$$

where A_c is the carrier amplitude and f_1 and f_2 are the carrier frequencies of the two symbols.

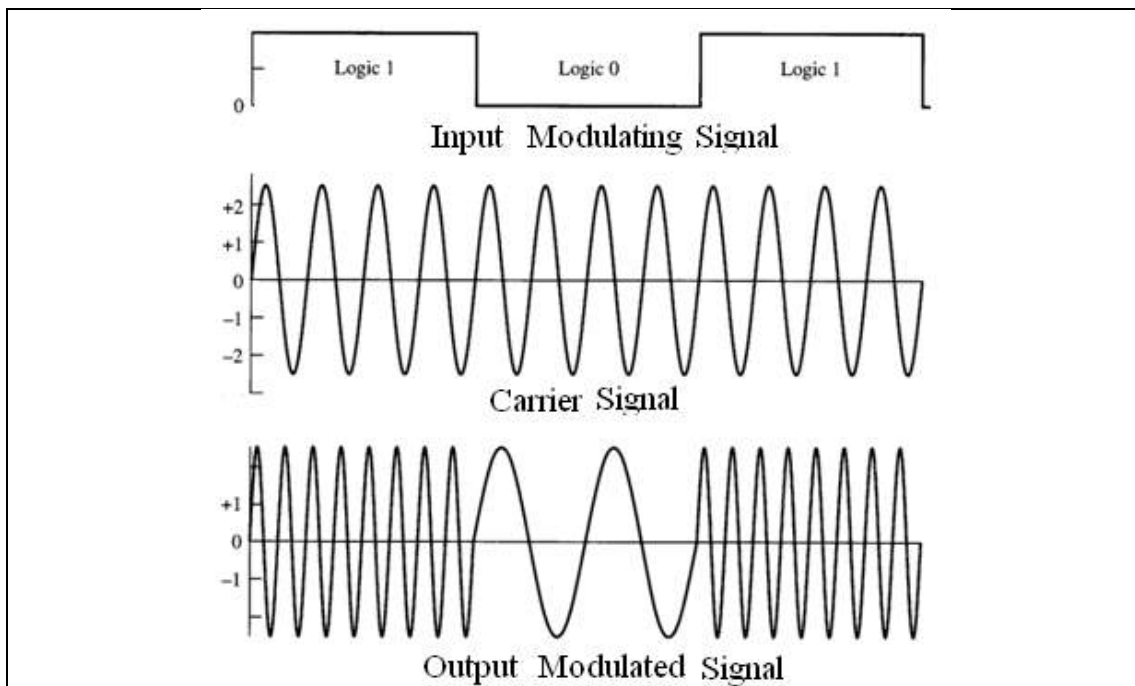


Fig. 5-6. Frequency shift keying (FSK).

The above equations may be also written in the following form:

$$s_1(t) = A_c \sin [2\pi (f_c + \frac{1}{2} \Delta f)t] \quad (5-3a)$$

$$s_2(t) = A_c \sin [2\pi (f_c - \frac{1}{2} \Delta f)t] \quad (5-3b)$$

where $\Delta f = f_1 - f_2$ is called the frequency deviation, or excursion, of the carrier wave. Also, the ratio $m = \Delta f / f_c$ is called the modulation index. In the usual form of FSK, the instantaneous frequency is shifted between two discrete values termed the "mark" and "space" frequencies. This is called non-coherent FSK. However, there exist other coherent FSK modulation methods, in which the carrier waveform is continuous in phase, during transition from '0' to '1' symbols and vice versa. The following figure shows the block diagram of an FSK transceiver system.

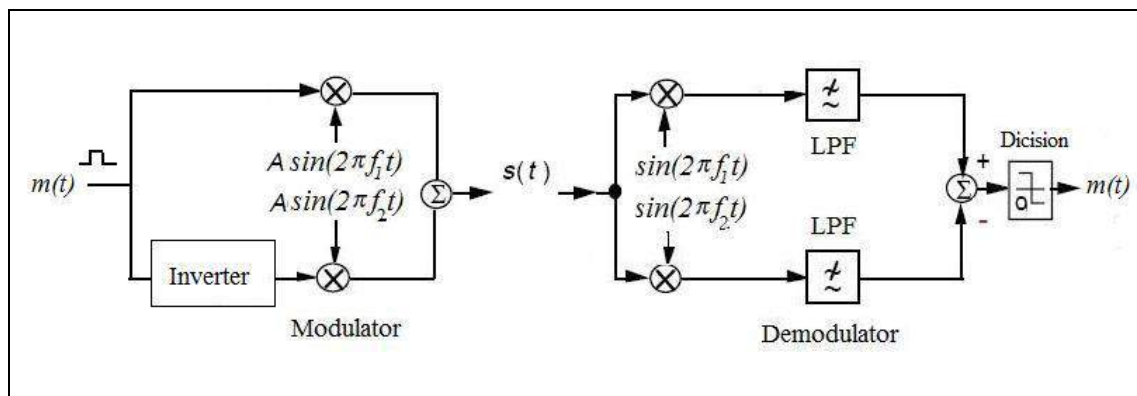


Fig. 5-7. Block diagram of binary frequency shift keying (BFSK) modulator (transmitter)/ demodulator (receiver).

The simplest form of an FSK modulator is composed of a voltage-controlled oscillator (VCO). The FSK demodulator is simply a product detector, which consists of a carrier generator and a multiplier followed by a low-pass filter. Note that a signal is said to be *coherent* if its behavior at both the transmitter and receiver sides is synchronized.

Figure 5-8 depicts the circuit diagram of a complete FSK data modem, which can be used to transmit data over telephone lines

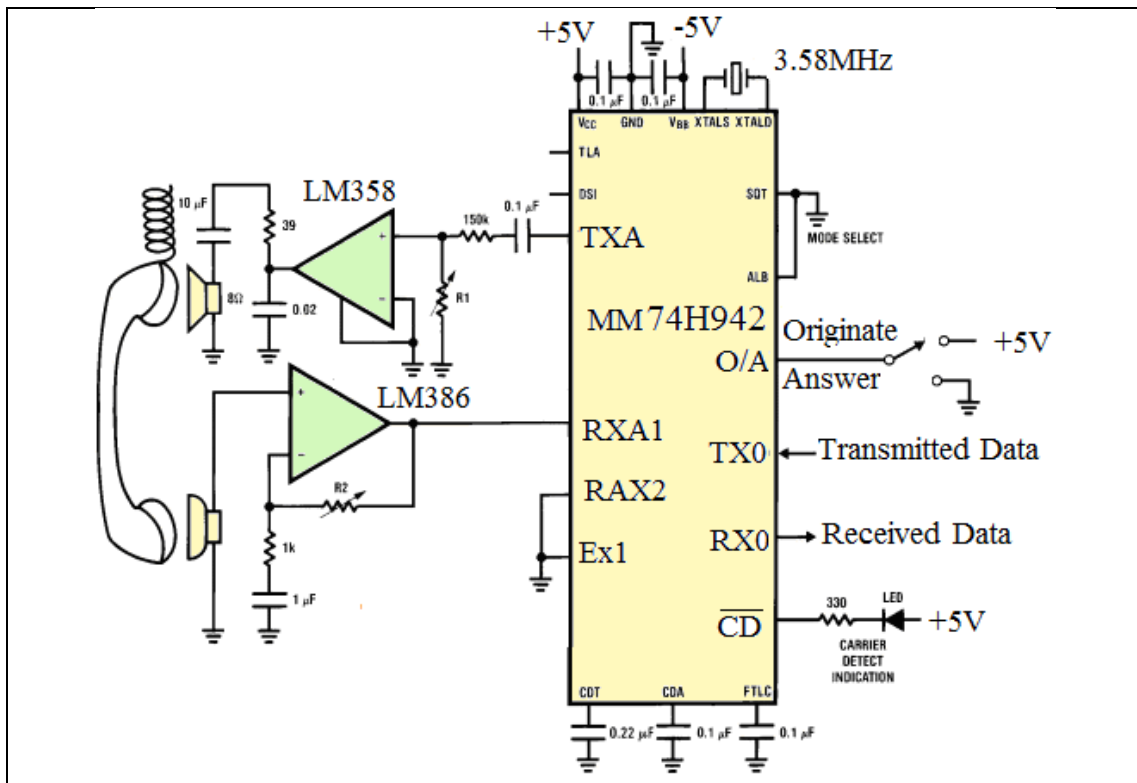


Fig. 5-8. Circuit diagram of a complete FSK modem

5-4. Phase-Shift Keying (PSK)

In PSK, the carrier phase is changed according to the input data signal. For instance, the following figure depicts a PSK signal where logic 1 produces no phase shift and logic 0 produces a 180° phase shift. The figure 5-9 illustrates the timing diagram of the PSK modulation process.

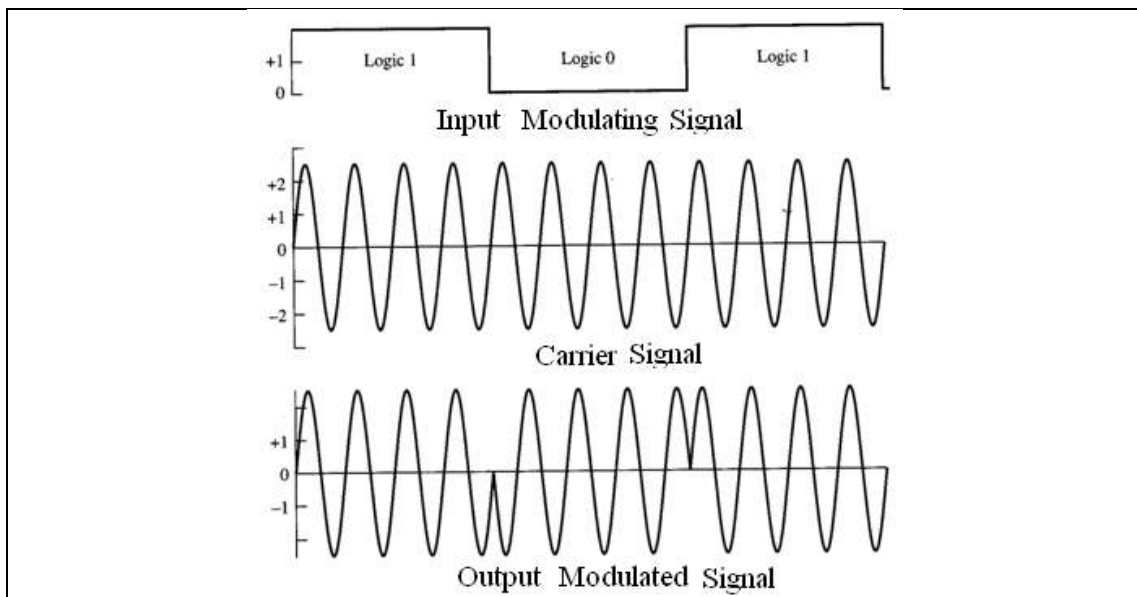


Fig. 5-9. Phase shift keying (PSK).

Another convenient way to represent PSK schemes is on the so-called **constellation diagram**. This diagram shows the points in the Argand plane where the real and imaginary axes are termed the **in-phase** and **quadrature** axes, respectively. As shown in figure 5-10, we can plot the carrier amplitude, a , and phase, ϕ , to represent a symbol as a dot on the graph. The plot on the right of figure 5-10, then represents the effect of noise upon reception. If the typical level of noise is represented by n then we can draw a circle of radius n around the intended transmitted position..

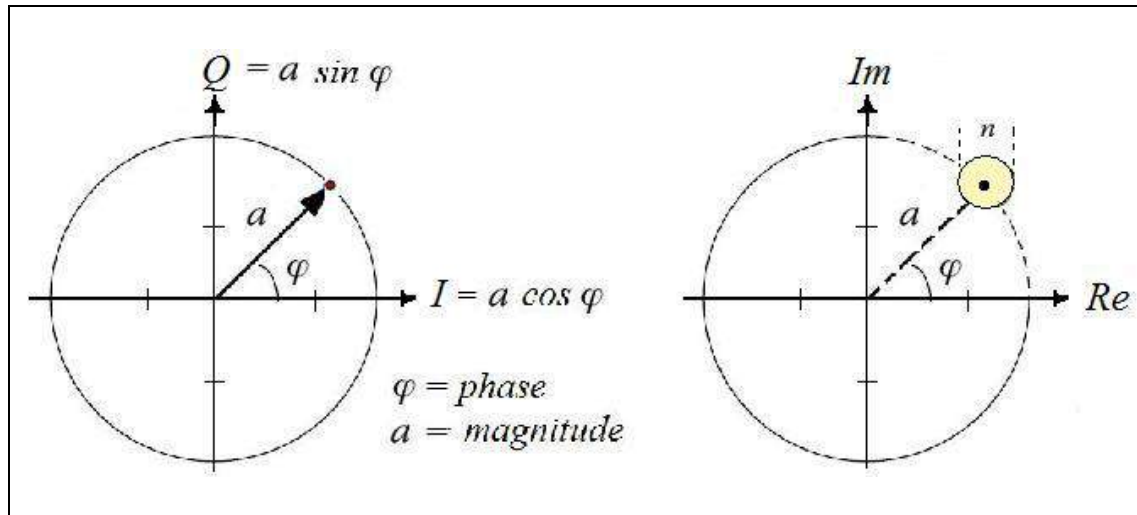


Fig. 5-10. Constellation diagram of symbol amplitude and phase

5-5. Binary Phase Shift Keying (BPSK)

A simple alternative to the On/Off modulation is to change the phase of the carrier to distinguish a '1' from a '0'. An example is shown in Figure 5-11. This method is called Binary Phase Shift Keying (**BPSK**) modulation, and generally preferred to the simple On/Off method. In fact, the BPSK is the simplest form of PSK. We can represent BPSK in different ways. For instance, we can say that for a '0' we have *amplitude* = A , $\phi = 0$ and for a '1' we have *amplitude* = $-A$, $\phi = 0$,

5-5.1. BPSK Constellation Diagram

When the signal is received and presented to a demodulator, the desired signal will always be accompanied by some unknown (random) noise. Hence we need the difference between the signal for a '0' and that for a '1' to be large enough to be recognizable above this noise. In order to consider this we can use the constellation diagram shown in figure 5-11.

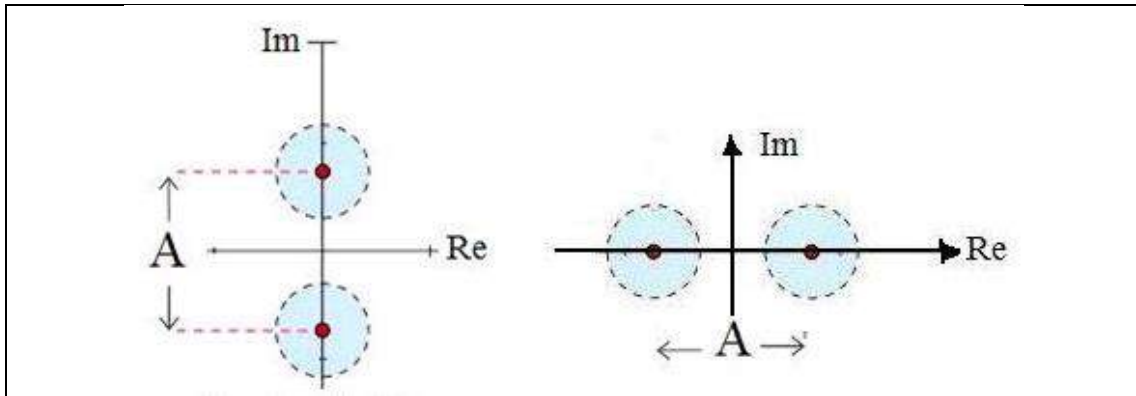


Fig. 5-11. Two possible constellation diagrams of the BPSK

5-5.2. BPSK Signals

The BPSK data is typically conveyed with the following signals:

$$s_0(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t + \pi) = -\sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t) \quad (5-4a)$$

for binary "0" and

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t) \quad (5-4b)$$

for binary "1", where E_b is the signal (bit) energy, T_b is the bit-duration and f_c is the frequency of the carrier-wave. The following figure depicts typical BPSK constellation diagram, as described by (5-7).

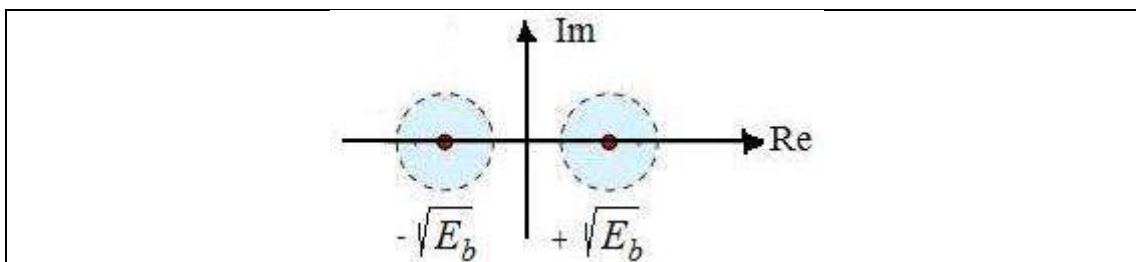


Fig. 5-12. Typical BPSK constellation diagram.

5-5.3. Power Spectral Density of BPSK Signals

The power spectral density (PSD) of a BPSK signal, as described by equation (5-7), can be put in the following form:

$$S_B(f) = 2E_b \text{sinc}^2 [(f - f_c)T_b] \quad (5-5)$$

where T_b is the bit duration, E_b is the bit energy, and f_c is the carrier frequency, and the *sinc* function $\text{sinc}(x) = \sin x/x$.

The following figure depicts the above relation. As shown in figure, the spectrum for the BPSK is *sinc*-shaped with a maximum at f_c and nulls at multiples of $2f_b$:

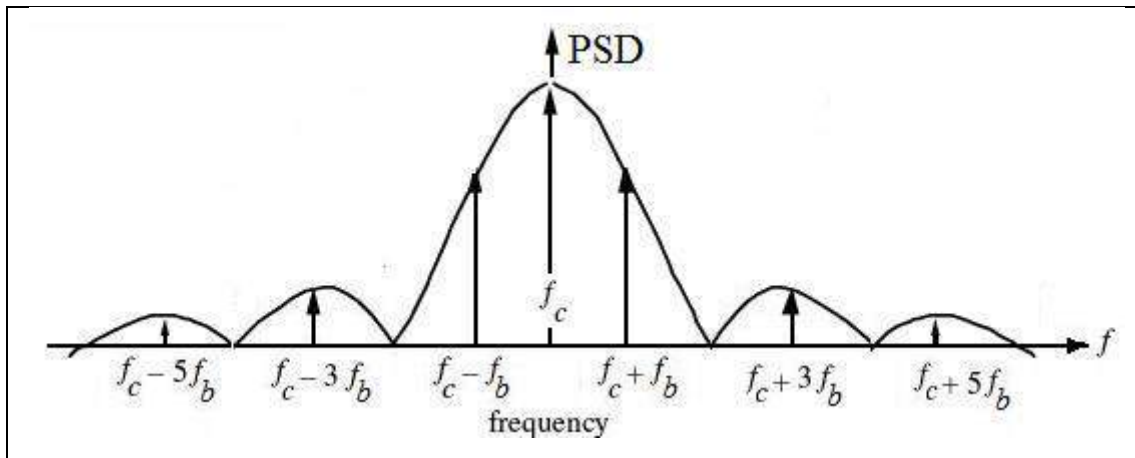


Fig. 5-13. Theoretical spectrum density of a BPSK signal.

5-5.4. Implementation of BPSK

As pointed out above, the BPSK requires a coherent receiver, in order to recognize the phase of the received signals. Figure 5-14 depicts the BPSK modulator block diagram. At the transmitter side of a BPSK system, the steps used by a modulator to transmit data are as follows:

- Group the incoming digital data into code-words;
- Convert the 1 and 0 signals into (DAC) bipolar analog values (+1, -1)
- Modulate the high-frequency carrier waveform, such that the lowpass signal is transformed into a shifted (modulated) RF signal

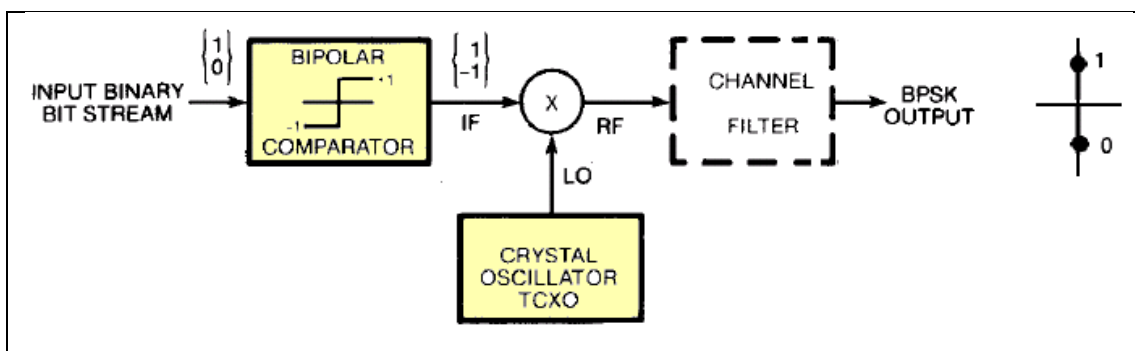


Fig. 5-14. Block diagram of a BPSK transmitter (modulator).

Like any other PSK demodulation process, can be divided into three major subsections, as shown in figure 5-15. First, since the incoming waveform is suppressed carrier in nature, coherent detection is required.

The methods by which a phase-coherent carrier is derived from the incoming signal are termed, **carrier recovery**. Next, the raw data are obtained by coherent multiplication, and used to derive clock-synchronization information. The **raw data** are then passed through the channel filter, which shapes the pulse train so as to minimize inter-symbol-interference (**ISI**) distortion effects. This shaped pulse train is then routed, along with the **derived clock**, to the data sampler which outputs the demodulated data

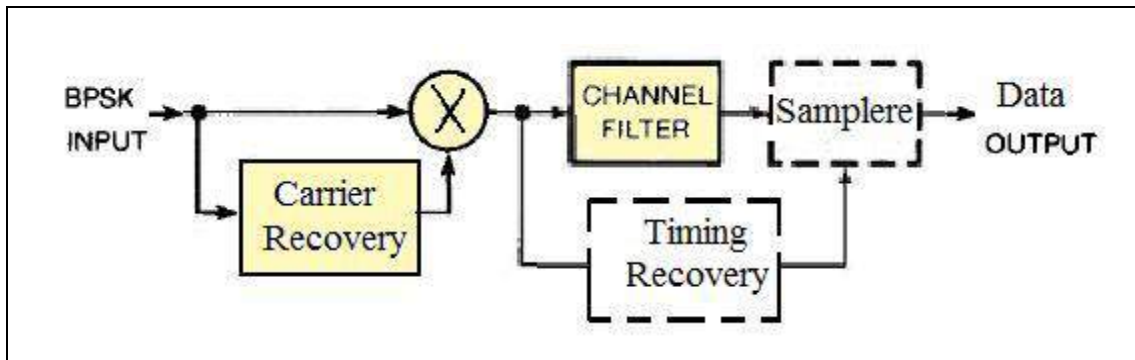


Fig. 5-15. Block diagram of a BPSK receiver (demodulator).

The following figure depicts the details of the time recovery circuit. The nonlinear element is simply a threshold detector, as shown below.

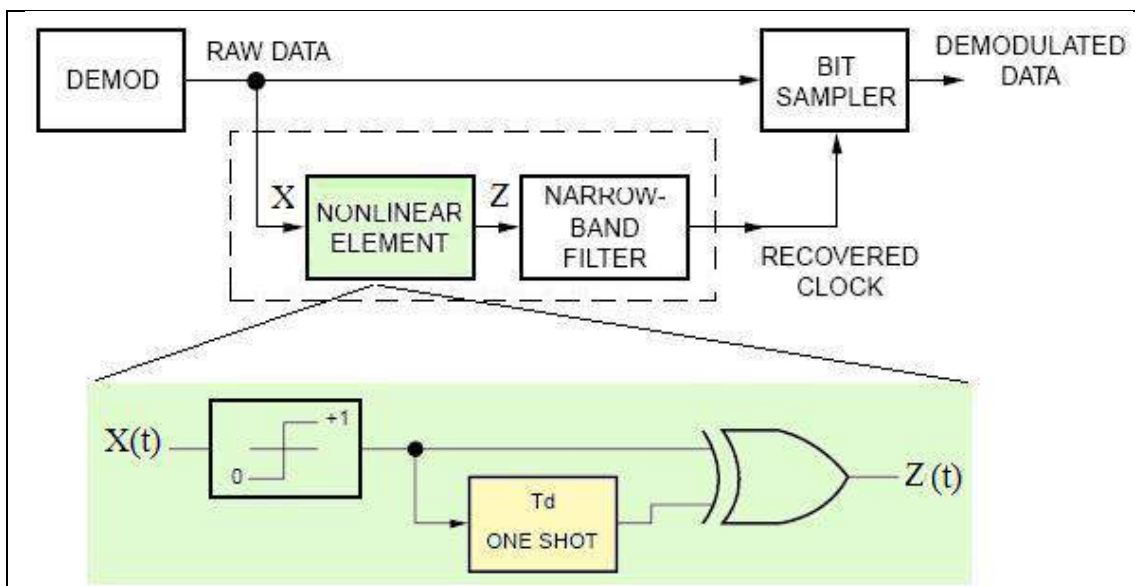


Fig. 5-16. Details of timing recovery in PSK demodulators.

Note that the BPSK receiver (like any coherent receiver) needs to be synchronized with the transmitter. Actually, the synchronization process involves two processes:

- 1- **Carrier recovery** process, which means knowledge of carrier frequency and phase,
- 2- **Time recovery** (or clock recovery) process, which means knowledge of the instances at which the modulation changes its state, i.e., the symbol starting and finishing times.

The carrier recovery and clock recovery processes can be achieved simultaneously, or sequentially, one after another.

5-5.5. Costas Loop

The Costas loop is a famous circuit that can be used for coherent reception of double sideband-suppressed carrier signals. The Costas loop performs both phase-coherent suppressed carrier reconstruction and synchronous data detection within the loop. The conventional Costas loop for BPSK suppressed carrier recovery is shown in figure 5-17. The upper loop is referred to as the quadrature, or **tracking loop**, and functions as a typical phase-locked loop (**PLL**). This loop provides a data-corrupted signal, $Z_c(t)$. Here, the phase-frequency detector (**PFD**) plays the role of a phase discriminator, whose output is filtered, via a loop filter $F(s)$, and the resultant signal is fed to the voltage-controlled oscillator (**VCO**). The lower in-phase, or **decision loop** provides data extraction at the output of the lower mixer, and corrects the data corruption of $Z_c(t)$. The corrected error signal, $Z_o(t)$, is applied through a loop filter $F(s)$ to the **VCO**, which yields a phase estimate in the form $\cos\phi(t)$.

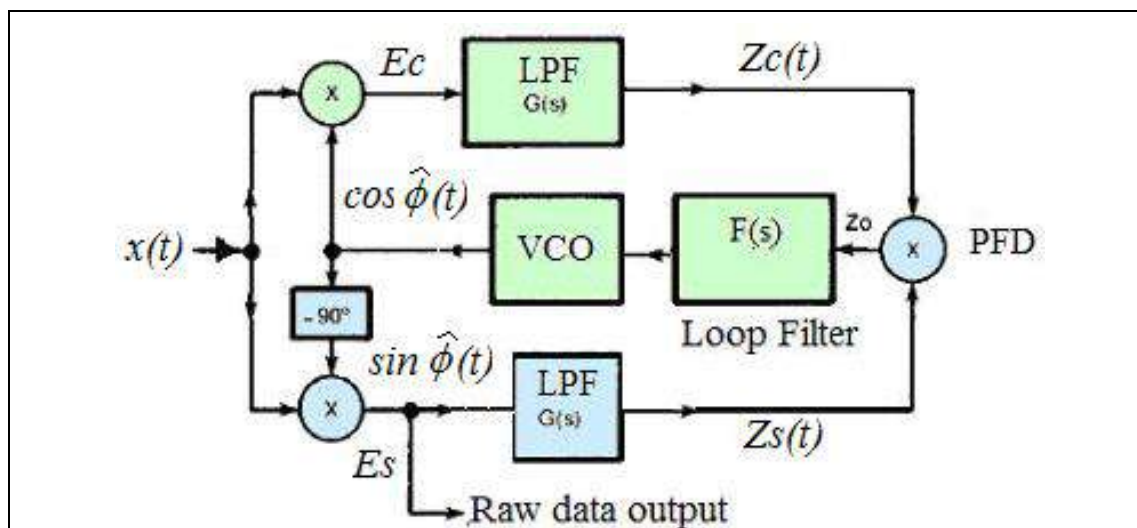


Fig. 5-17. Costas loop for data recovery in BPSK demodulator. The raw data output, represents the demodulated bit stream (before LPF and data sampler).

The following figure depicts the circuit diagram of a real BPSK transceiver (transmitter and receiver). The circuit is based on the AD8346 modulator and AD8347 demodulator IC's from Analog Devices Inc.

Note that MAX2620 is an oscillator IC and Max2640 is a low-noise amplifier (LNA), both from Maxim Inc. Also, the ETC 1-1-13 is a *balun* (balanced-to-unbalanced matching circuit) and Gali-3 is an RF power amplifier (RFPA) IC.

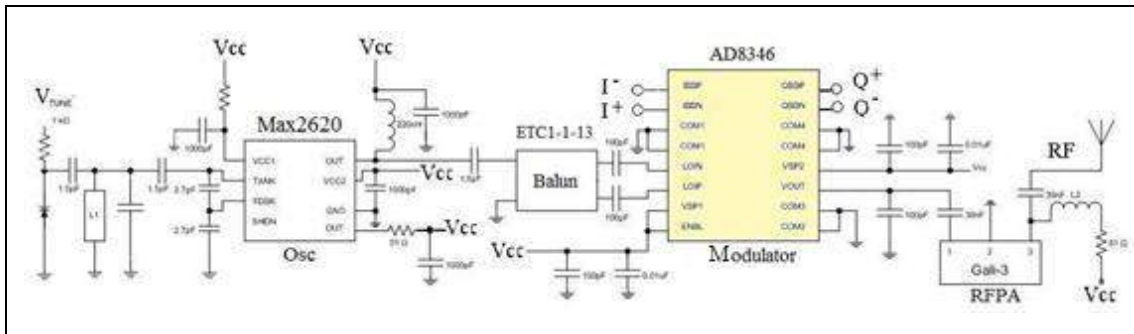


Fig. 5-18(a). Circuit diagram of a BPSK transmitter.

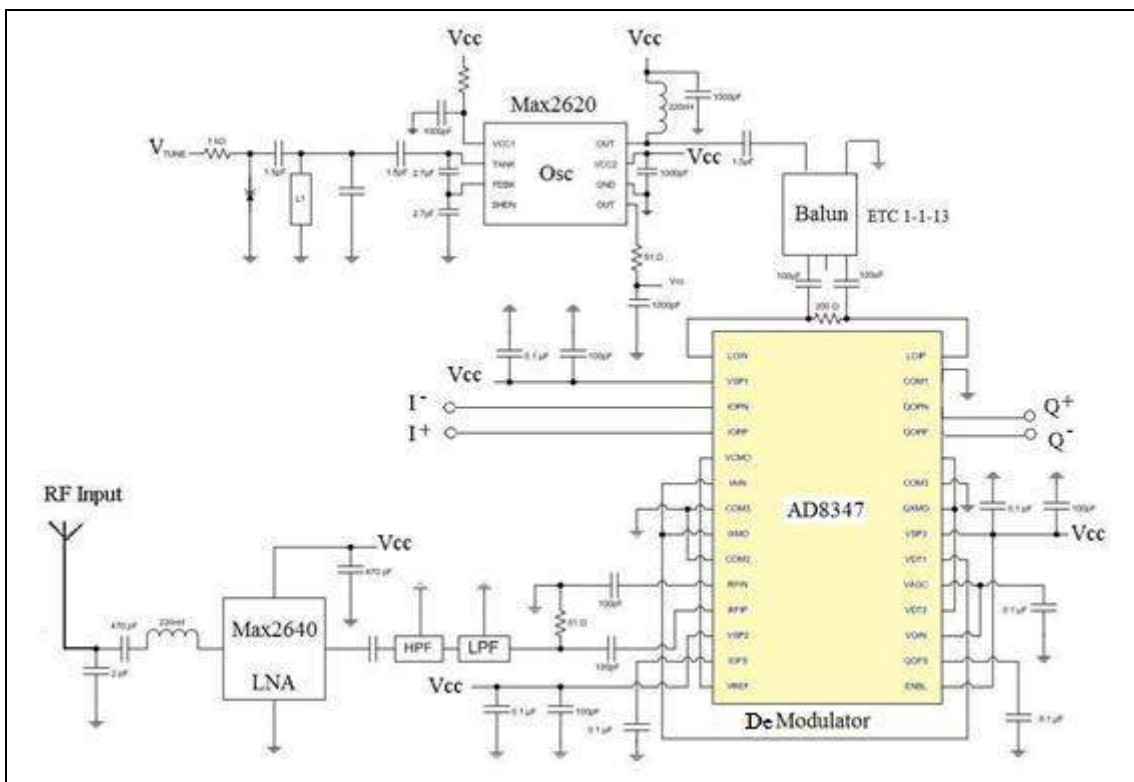


Fig. 5-18(b). Circuit diagram of a BPSK receiver.

5-6. Quadrature Phase Shift Keying (QPSK)

We consider now even more complicated forms of digital modulation by allowing the system to use more than two phases for distinct symbols. Figure 5-19 depicts the time diagram of a typical Quadrature Phase Shift Keying (QPSK) system. This technique is similar to BPSK in that all the symbols have the same amplitude, but here we have four carrier phases to choose between. Since we now have four possible symbols we can use them to represent more than one bit per symbol. In effect, we can communicate more than one bit at a time. Each possible QPSK symbol can be labeled and used to represent two bits, as shown in figure.

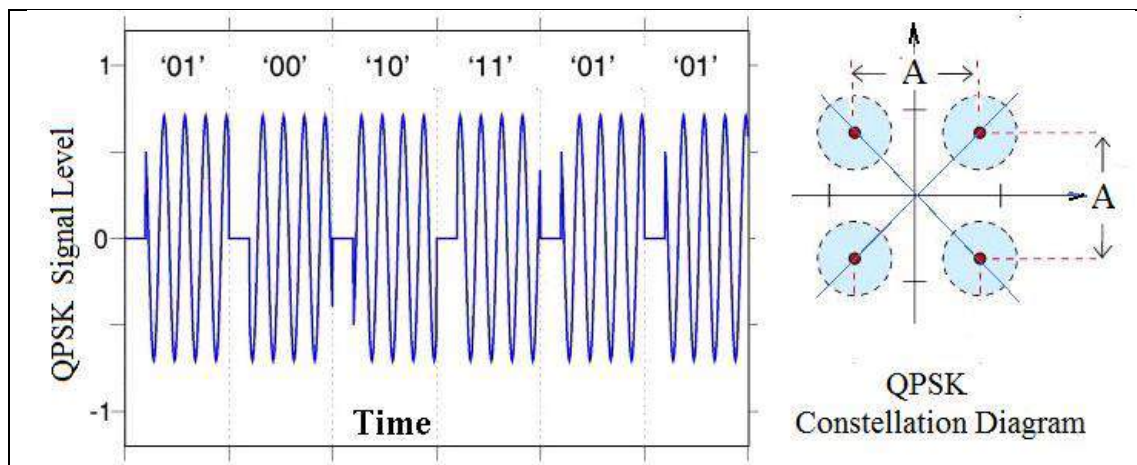


Fig. 5-19. Quadrature Phase Shift Keying (QPSK) and its constellation diagram

5-6.1. Constellation Diagrams of QPSK

Looking at the example of QPSK illustrated in figure 5-19, we can see that the four distinct symbol patterns can be spaced an amplitude, A , apart, but since we are now doing this in two dimensions the actual amplitude of the carrier will be $A/\sqrt{2}$ in each case. The carrier phase can be any one of the following four values:

$$\phi = \pi/4, 3\pi/4, 5\pi/4, 7\pi/4 \quad (5-6)$$

A simpler way to describe this is to define the QPSK signal as follows:

$$s(t) = A \cos(2\pi f_c t - \phi) = a \cos(2\pi f_c t) + b \sin(2\pi f_c t) \quad (5-7)$$

where $a = \frac{1}{2} A \cos(\phi)$ and $b = \frac{1}{2} A \sin(\phi)$. In conventional QPSK, we can represent pairs of bits for each symbol according to the following table and the figure below.

	'11'	'10'	'00'	'01'
a	1	1	-1	-1
b	1	-1	-1	1
ϕ	$\pi/4$	$7\pi/4$	$5\pi/4$	$3\pi/4$

Where we normalized a and b by $A/\sqrt{2}$.

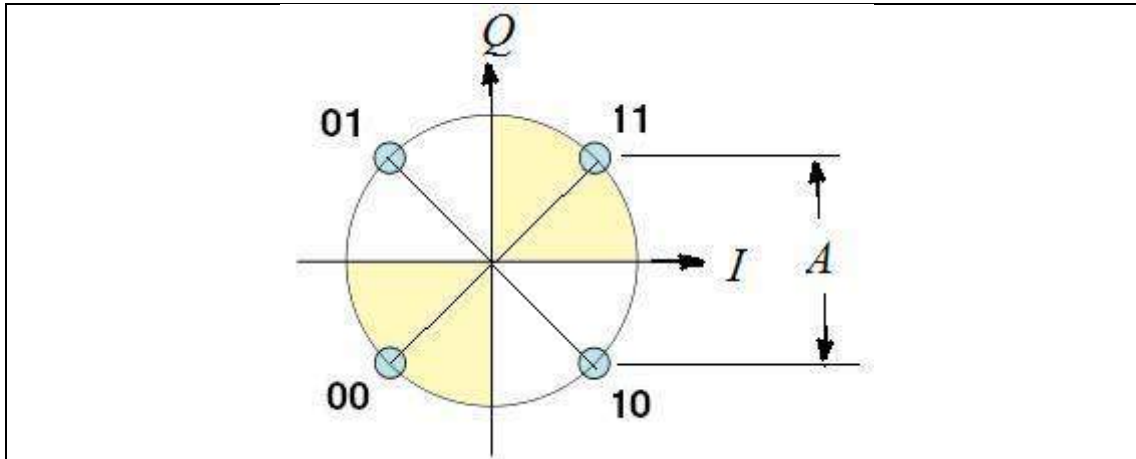


Fig. 5-20. Constellation diagram of a QPSK system, with Gray code.

Note that QPSK provides twice the information per symbol that BPSK provides, but that it requires the signal amplitude, $\sqrt{a^2+b^2}$, to increase from $A/2$ to $A/\sqrt{2}$. This means that the signal carrier power for **QPSK** must be **twice** that of **BPSK** for the same *symbol rate*. Thus we pay for doubling in information capacity by having to provide double the power.

4.6.2. QPSK Signal in the Time Domain

We can write the symbols of the conventional QPSK constellation diagram in terms of the cosine and sine waves as follows:

$$s(t) = A \cos(\omega t - \phi), \quad \phi = \pi/4, 3\pi/4, 5\pi/4, 7\pi/4 \quad (5-8)$$

By comparing equation (5-7) with (5-8) we get $A = \sqrt{2E_s/T_s}$ and $\phi = \pi/4, 3\pi/4, 5\pi/4, 7\pi/4$. This results in a two-dimensional signal space with unit basis functions

$$\phi_1(t) = \sqrt{\frac{2}{T_s}} \cos(2\pi f_c t) \quad (5-9a)$$

$$\phi_2(t) = \sqrt{\frac{2}{T_s}} \sin(2\pi f_c t) \quad (5-9b)$$

Such that we can write the QPSK signal in the following quadrature form:

$$s(t) = \sqrt{E_b} \cdot \cos(\phi) \cdot \phi_1(t) + \sqrt{E_b} \cdot \sin(\phi) \cdot \phi_2(t) \quad (5-10)$$

The first basis function ϕ_1 is used as the in-phase (**I**) component of the signal and the second function ϕ_2 is used as the quadrature (**Q**) component of the signal. Hence, the signal constellation consists of 4 points with 4 polarities, as shown in the following figure.

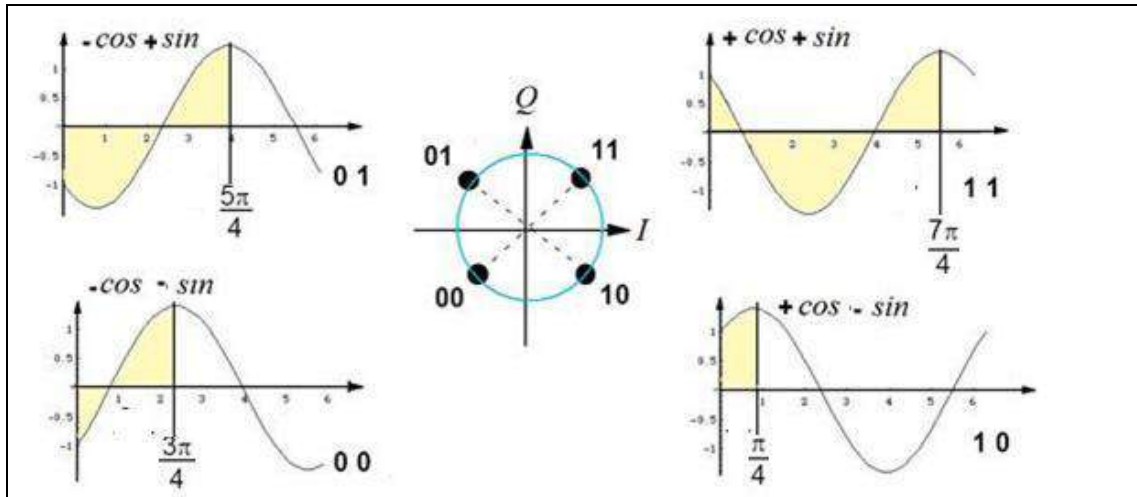


Fig. 5-21. Signals of a QPSK system, with Gray code.

The following table summarizes the four symbols of QPSK

Table 5-1. Summary of QPSK symbols and signals

Symbol	Bits	S(t)	Phase, (Deg.)		I	Q
S1	11	$\sqrt{\frac{2E_s}{T}} \cos(2\pi f_c t + \pi/4)$	45°		1	1
S2	01	$\sqrt{\frac{2E_s}{T}} \cos(2\pi f_c t + 3\pi/4)$	135°		-1	1
S3	00	$\sqrt{\frac{2E_s}{T}} \cos(2\pi f_c t + 5\pi/4)$	225°		-1	-1
S4	10	$\sqrt{\frac{2E_s}{T}} \cos(2\pi f_c t + 7\pi/4)$	315°		1	-1

The modulated signal of a QPSK system is shown in figure 5-22, for a short segment of a binary data-stream. The two carrier waves are a cosine wave and a sine wave, as indicated by the signal analysis above. Here, the odd-numbered bits have been assigned to the in-phase component and the even-numbered bits to the quadrature component (taking the first bit as number 1). The total signal (the sum of the two components) is shown at the bottom. Jumps in phase can be seen as the PSK changes the phase on each component at the start of each bit-period. The topmost waveform alone matches the description given for BPSK. The binary data stream is shown beneath the time axis. The two signal components with their bit assignments the top and the total, combined signal at the bottom. Note the abrupt changes in phase at some of the bit-period boundaries. The binary data that is conveyed by this waveform is: 11000110. The odd bits contribute to the in-phase component: 11000110. The even bits contribute to the quadrature-phase component: 1100 0110

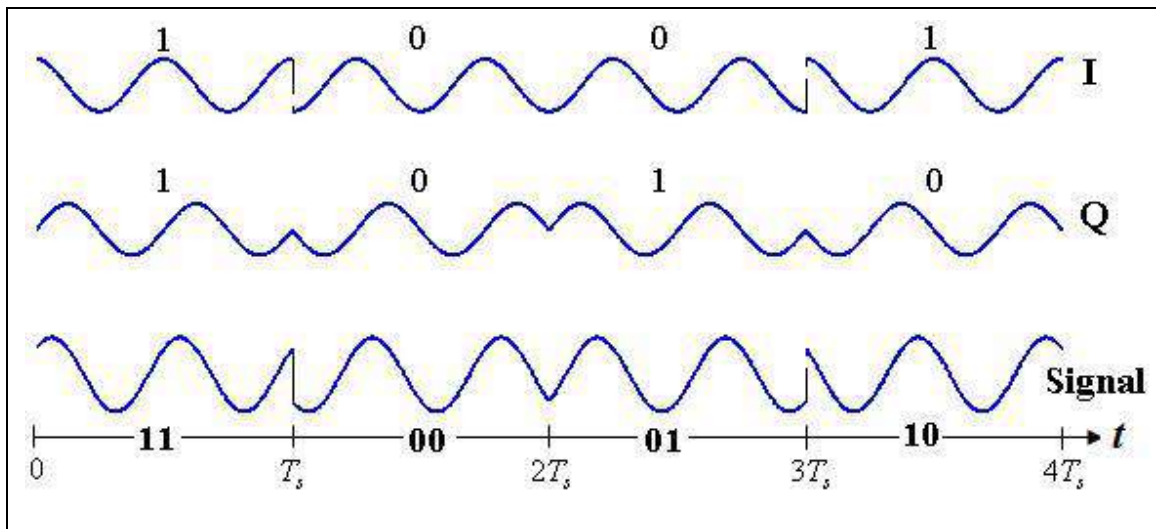


Fig. 5-22. Timing diagram of conventional QPSK signals.

4.6.3. Power Spectral Density of QPSK Signals

The PSD of a QPSK signal is given by:

$$S_B(f) = 4E_b \text{sinc}^2 [2(f-f_c)T_b] \quad (5-11)$$

Generally speaking, the PSD of an M -ary PSK signal can be put in the following form:

$$S_B(f) = 2E_b \log_2(M) \text{sinc}^2 [T_b (f-f_c) \log_2(M)] \quad (5-12)$$

Figure 5-23 depicts the normalized power spectral density (PSD) of a QPSK signal, together with the PSD of BPSK and 8PSK, for the matter of comparison. Note the twice bandwidth requirement of BPSK.

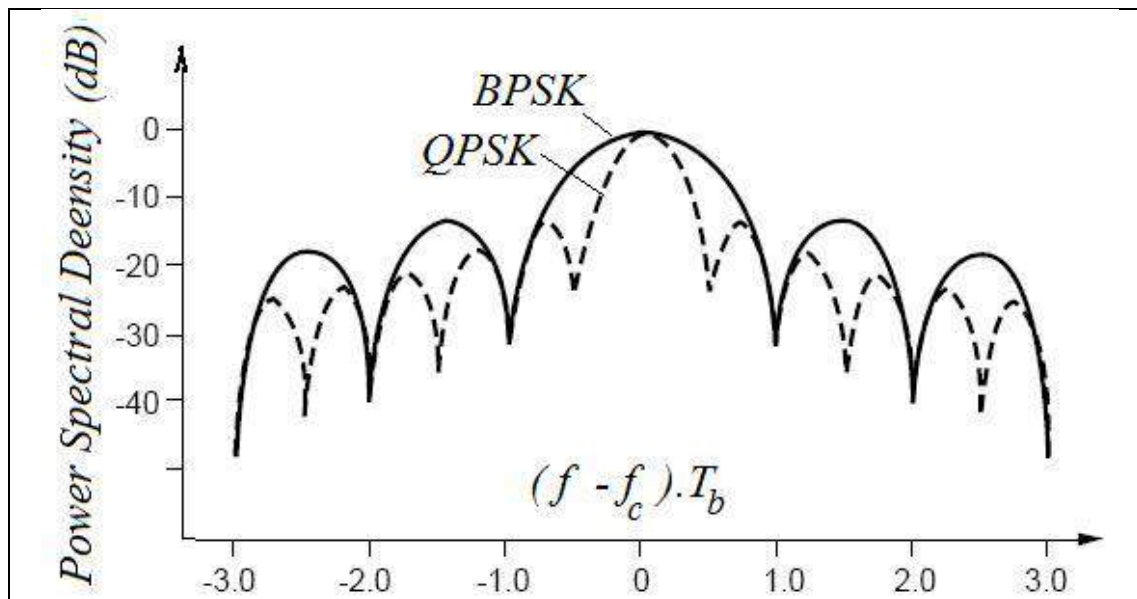


Fig. 5-23. Normalized power spectral density (PSD) of a BPSK and QPSK signals.

5-6.4. Alternative Types of QPSK

In the conventional QPSK, we take four values ($a = \pm 1/2 A$, $b = \pm 1/2 A$), which correspond to four phases to construct the QPSK symbols. This allows the phase of the signal to jump by as much as 180° at a time. When the signal is low-pass filtered (as usual in a digital transmitter), these phase-shifts result in large amplitude fluctuations, which is an undesirable quality in communication systems. The fluctuations can be reduced by using one of the QPSK variants, called **Offset QPSK** or **$\pi/4$ QPSK**. In the following subsections, we describe these techniques.

i- Offset QPSK

Offset quadrature phase-shift keying (OQPSK) is a variant of phase-shift keying modulation using 4 different values of the phase to transmit. It is sometimes called *staggered quadrature phase-shift keying (SQPSK)*. By offsetting the timing of the odd and even bits by **one bit-period**, or half a symbol-period, the in-phase and quadrature components will never change at the same time. In the constellation diagram shown below, it can be seen that this will limit the phase-shift to no more than 90° at a time. This yields much lower amplitude fluctuations than conventional QPSK and is sometimes preferred in practice.

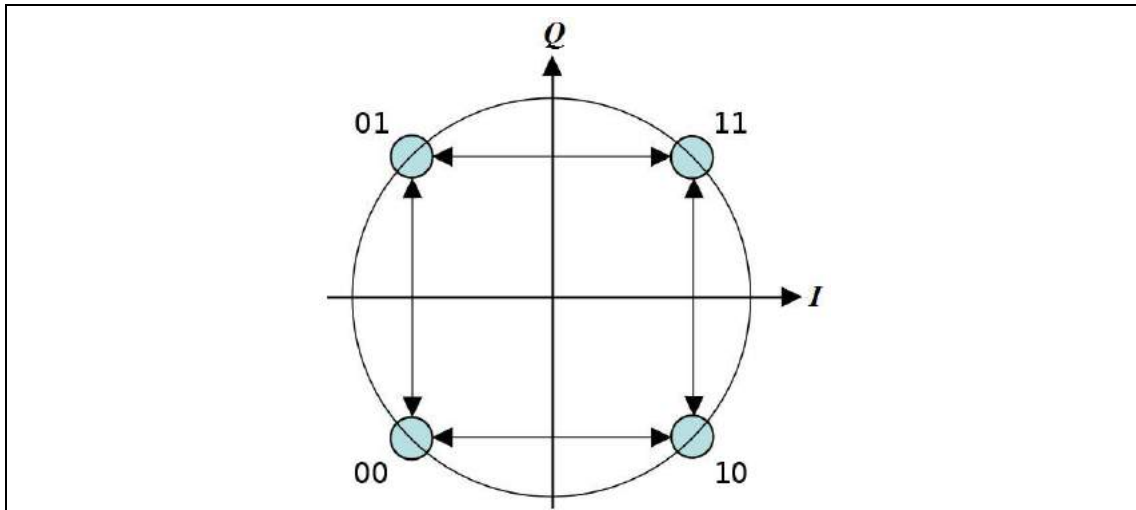


Fig. 5-24. Constellation diagram of an offset QPSK.

The following figure shows the signal behavior of the OQPSK. Obviously, one can see that in OQPSK the changes are never greater than 90° . The modulated signal is shown below for a short segment of a random binary data-stream. Note the half symbol-period offset between the two components. The sudden phase-shifts occur about twice as often as for QPSK (since the signals no longer change together), but they are less severe. In other words, the magnitude of jumps is smaller in OQPSK when compared to conventional QPSK

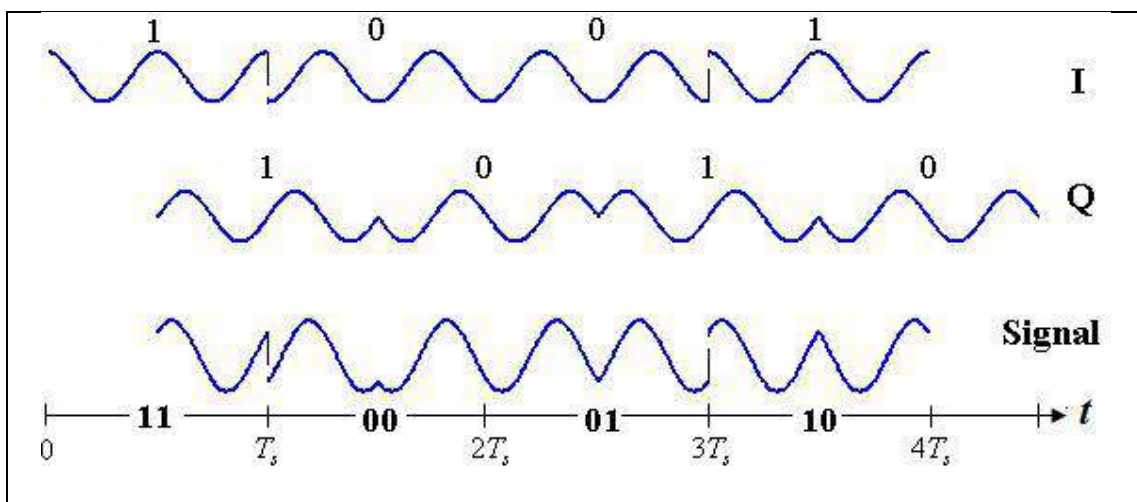


Fig. 5-25. Timing diagram of an offset QPSK.

ii- $\pi/4$ QPSK

This variant of QPSK uses two identical constellations which are rotated by 45° . Usually, either the even or odd symbols are used to select points from one constellation and the other symbols select points from the other

constellation. This reduces the phase-shifts from a maximum of 180° , but only to a maximum of 135° . When the $\pi/4$ QPSK modulated signal is represented in the complex domain, this modulation scheme does not have any paths through the origin. In other words, the signal does not pass through the origin. This lowers the dynamical range of fluctuations of the signal. On the other hand, the $\pi/4$ QPSK is easy in demodulation and has been adopted for use in, many applications like, cellular telephones. The modulated signal is shown below for a short segment of a binary data-stream.

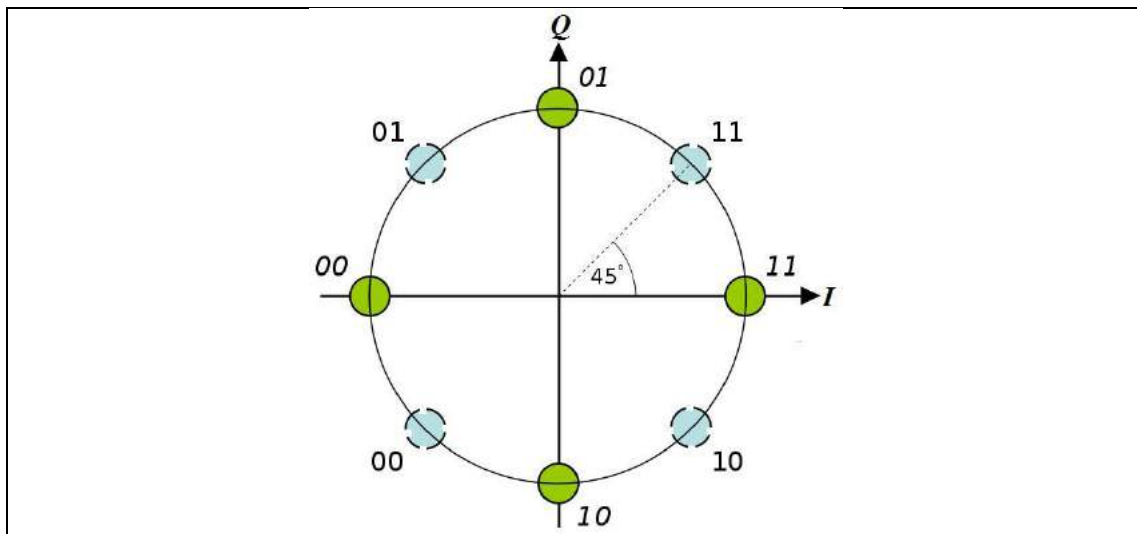


Fig. 5-26. Constellation diagram of the $\pi/4$ -QPSK.

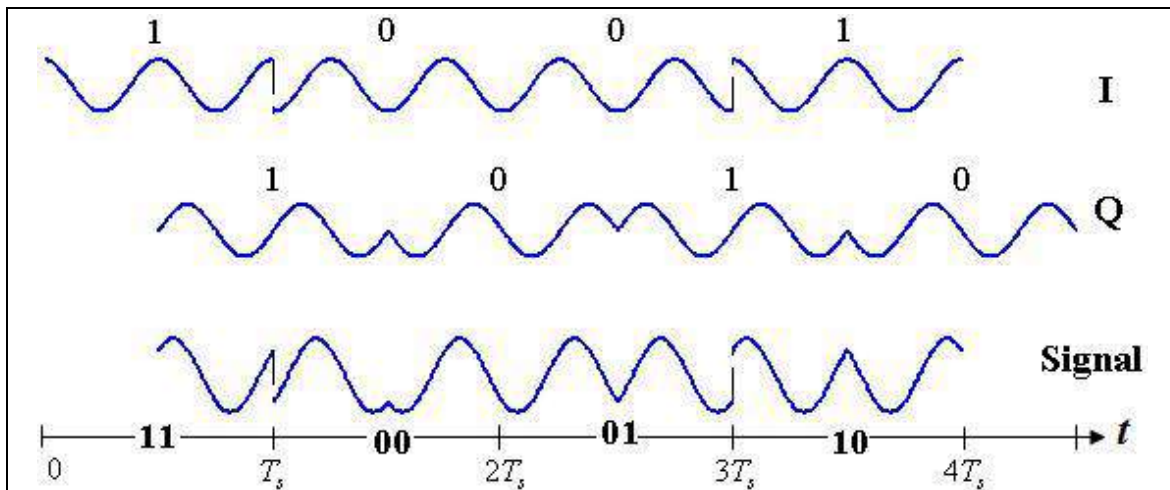


Fig. 5-27. Timing diagram of the $\pi/4$ -QPSK signal.

The signal builds in the same way as for ordinary QPSK. Successive symbols are taken from the two constellations shown in figure 5-28. Thus, the first symbol (1 1) is taken from the 'blue-dotted' constellation

and the second symbol (0 0) is taken from the 'green-sold' constellation. Note that magnitudes of the two components change as they switch between constellations, but the total signal's magnitude remains constant. The phase-shifts are between those of the two previous timing-diagrams. The following figure depicts the different types of QPSK constellations.

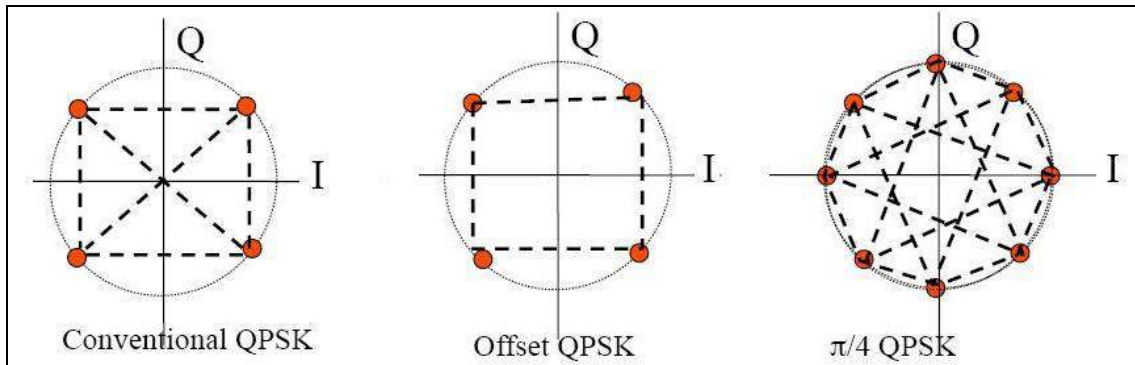


Fig. 5-28. Constellation diagram of different types of QPSK systems.

5-6.5. Implementation of QPSK

The implementation of QPSK is more general than that of BPSK and also indicates the implementation of higher-order PSK. Comparing the basic functions with that for BPSK shows clearly how QPSK can be viewed as two independent BPSK signals. PSK systems can be implemented in a number of ways. A block diagram for a QPSK modulator is shown in Figure 5-29(a). The binary data stream is split into the in-phase (**I**) and quadrature-phase (**Q**) components. These are then separately modulated onto two orthogonal basis functions.

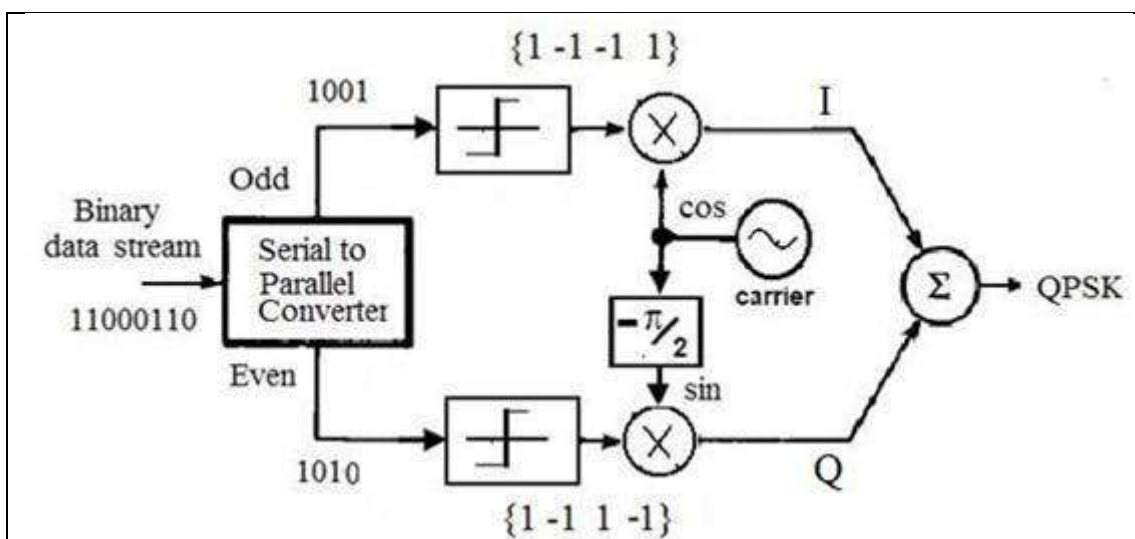


Fig. 5-29(a). Block diagram of a QPSK transmitter (modulator))

In this implementation, two sinusoids are used. The two signals are superimposed, and the resulting signal is the QPSK signal. Note the use of polar non-return-to-zero (NRZ) encoding. These encoders can be placed after the binary data source. The decision device uses a reference threshold value to determine whether a “1” or ‘0’ is detected.

At the transmitter side of a QPSK system, the modulator proceeds as follows:

- Group the input data into code-words;
- Map code-words to phases of the I and Q signals
- Adapt pulse shaping or some other filtering to limit the bandwidth
- Modulate the high-frequency carrier wave, such that the input signal is frequency shifted into a modulated RF signal

At the QPSK receiver, the demodulator performs the following steps:

- Frequency shifting of the RF signal baseband I and Q signals, to IF
- Detection of the phase of the IF signal;
- Quantization of the phases to nearest allowed values, using mapping.
- Map the quantized phases to code-words (bit groups);
- Parallel-to-serial conversion of the code-words into a bit stream

The parallel-to-serial converter block performs the following operations:

- regenerates the bit clock from the incoming data.
- regenerates a digital I and Q signals.
- re-combines the I and Q signals, and outputs a serial data stream.

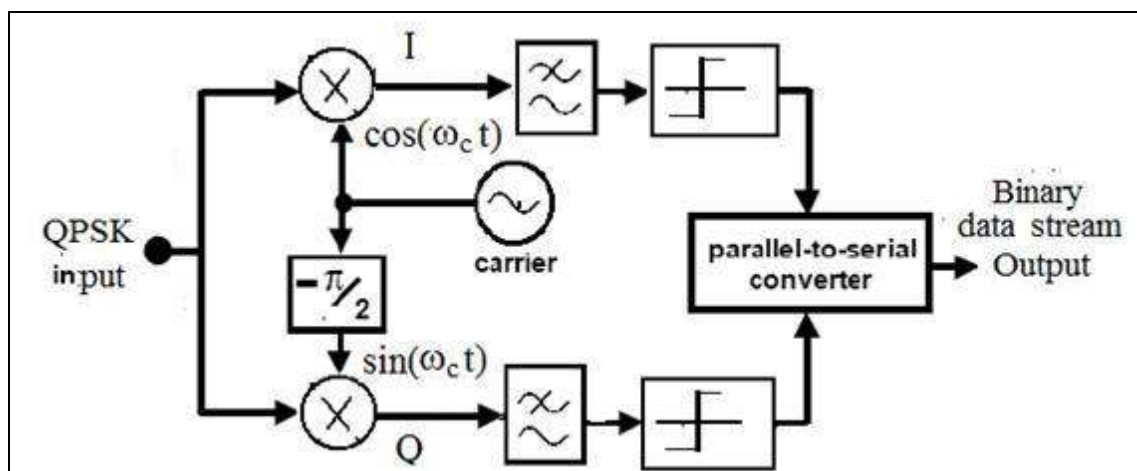


Fig. 5-29(b). Block diagram of a QPSK receiver (demodulator)

Example 5-2.

Consider the modulating signal **11000110**. Draw the output QPSK signal. The In-phase signal ($I = a \cos(2\pi ft)$) is obtained by passing the odd sequence (**1001**) to NRZ encoder to get $\{1 -1 -1 1\}$ and multiplying by $\cos(2\pi ft)$. The Quadratic-phase signal ($Q = b \sin(2\pi ft)$) is obtained by passing the even sequence (**1010**) to NRZ encoder to get $\{1 -1 1 -1\}$ and multiplying by $\sin(2\pi ft)$. The final QPSK signal is the sum of $I+jQ$, as shown in figure 5-30. Note that we consider here the standard QPSK constellation. The output QPSK signal may be written as $s(t) = A \cos(\omega t - \phi)$, where the phase (ϕ) of each symbol is indicated underneath the output signal and the amplitude $A = \sqrt{2E_s/T_s}$.

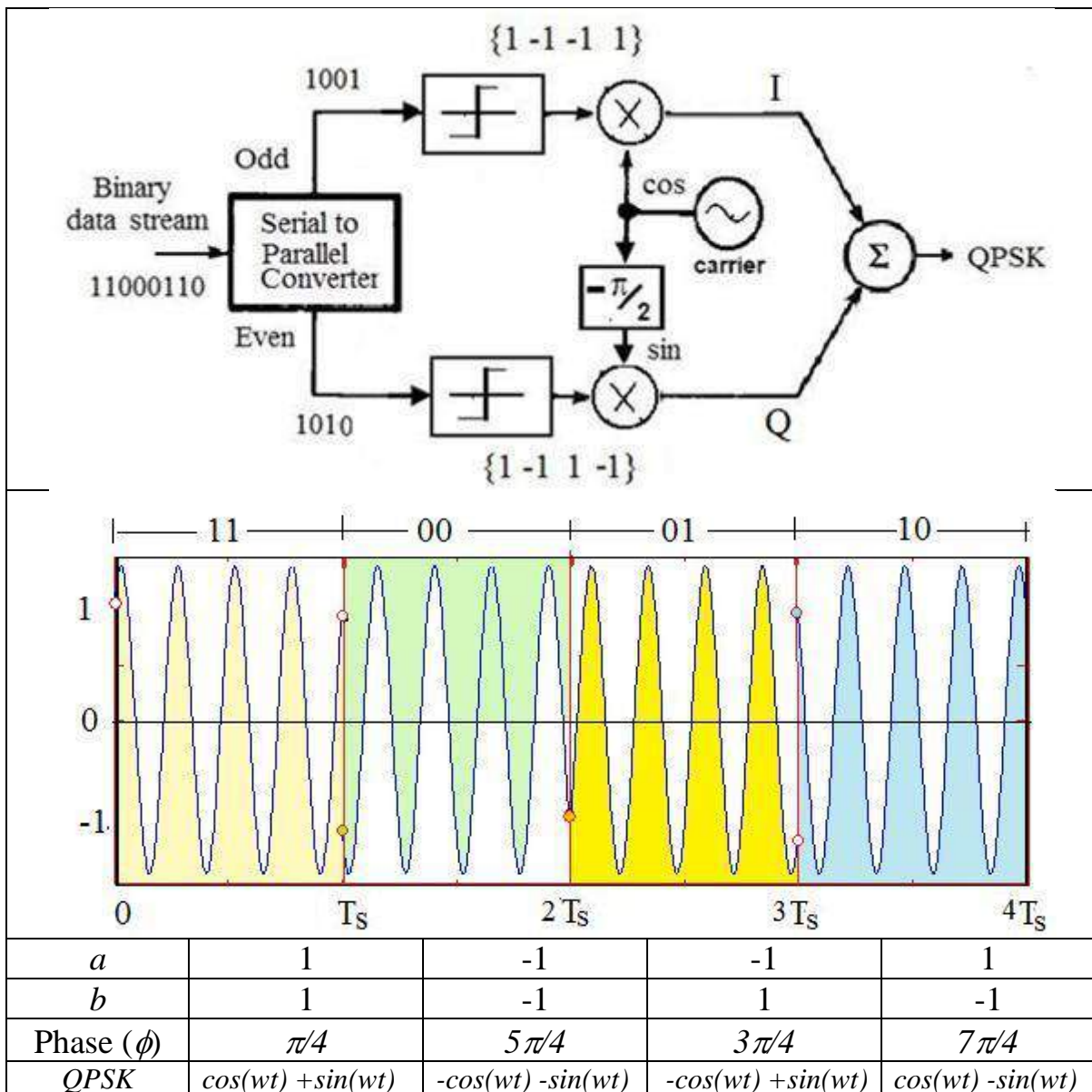


Fig. 5-30. Example of a QPSK modulation with input digital modulating signal [11000110]. The output QPSK signal is shown below.

5-7. Differential Encoding

The conventional PSK modulation (e.g., BPSK and QPSK) are called **coherent** PSK because it needs an exact **reference signal** at the receiver side in order to correctly recover the original signal. However, if the reference signal is not exactly synchronized, due some effect in the communications channel, there will be an ambiguity of phase and the constellation is rotated. This problem can be overcome by using the data to change rather than to set the phase. This is called differential encoding, or differential modulation. Obviously, differential encoding does not need a reference carrier signal at the receiver, and hence it is **incoherent**.

For example, in differentially-encoded BPSK (**DBPSK**), a binary '1' may be transmitted by **adding 180°** to the current phase and a binary '0' by **adding 0°** to the current phase. Therefore, the receiver decides that the received signal is '0' if the received symbol has a 0° shift from its previous symbol. Similarly, the receiver decides that the received signal is '1' if the received symbol has a 180° shift from its previous symbol.

The following figure depicts the waveforms of both DBPSK and DQPSK. It is assumed that the signal starts with zero phase. The binary data stream in the above figure is the DBPSK signal. The individual bits of the DBPSK signal are grouped into pairs for the DQPSK signal, which only changes every $T_s = 2T_b$.

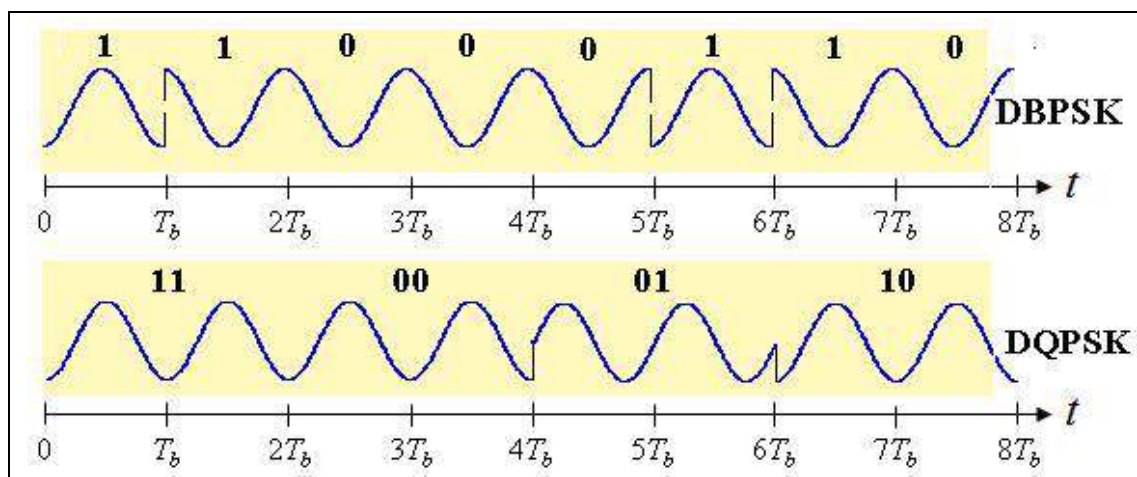


Fig. 5-31. Timing diagram for DBPSK and DQPSK.

Mathematically, DBPSK may be represented by the following difference equation:

$$y[n] = x[n] \oplus y[n-1] \quad (5-13)$$

where $x[n]$ is the input binary sequence and \oplus is the symbol of modulo-2 addition. The input sequence can be then modulated in much the same manner as BPSK. Figure 5-32 shows the block diagram of a DBPSK system. The received sequence $r[n]$ may be incoherently demodulated. However, the DBPSK sequence can be also coherently demodulated (like BPSK). In this case, the resulting binary sequence is delayed and modulo-two subtracted to obtain an estimate of the original bit sequence $x[n]$.

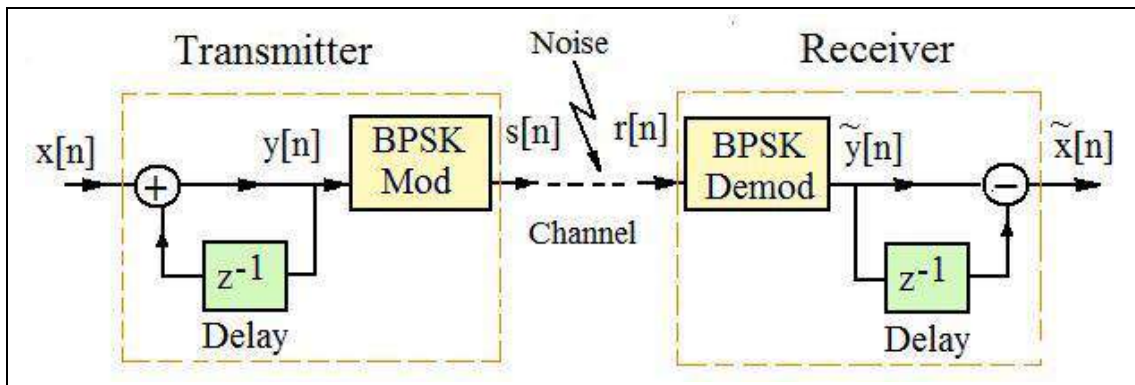


Fig. 5-32. Block diagram of a DBPSK transmitter/receiver system

This kind of encoding may be then demodulated in the same way as for non-differential (coherent) QPSK but the phase ambiguities can be ignored. In time domain, the transmitted DQPSK signal may be represented as follows:

$$x(t) = A \cos[2\pi f_c t + \theta(n)], \quad nT \leq t \leq (n+1)T \quad (5-14)$$

where θ is the phase difference between codes, which is given by:

$$\begin{aligned} \theta(n) - \theta(n-1) &= \pi, & \text{if } (I,Q) &= 00 \\ &= 3/2 \pi, & \text{if } (I,Q) &= 10 \\ &= 0, & \text{if } (I,Q) &= 11 \\ &= 1/2 \pi, & \text{if } (I,Q) &= 01 \end{aligned} \quad (5-15)$$

5-8. Quadrature Amplitude Modulation (QAM)

Having discovered that we can use a symbol to convey more than one bit at a time, we can extend this by allowing a choice of amplitudes as well of phases. A common example of this is *Quadrature Amplitude Modulation (QAM)*. This comes in various forms, and a typical example would be called something like '16QAM' or '64QAM' where the number indicates how many distinct symbols are available. Figure 5-33 shows an example. Note that it is common to describe the array of symbols displayed in this way as a *Constellation* of symbol values.

5-8.1. QAM Constellation Diagram

For simplicity we can consider square arrays of symbol locations, so that we have a number of symbols which is the square of an integer – i.e. have a number of symbols, $M=1, 2, 4, 8, \text{ or } 16$, etc, in the constellation. The example in Figure 5-33 is a 16QAM and hence has $M=16$ symbols. For such an array the amplitude of the locations which are furthest from the center (i.e. the symbols which require the maximum symbol amplitude) will be $A(\sqrt{M}-1)/\sqrt{2}$.

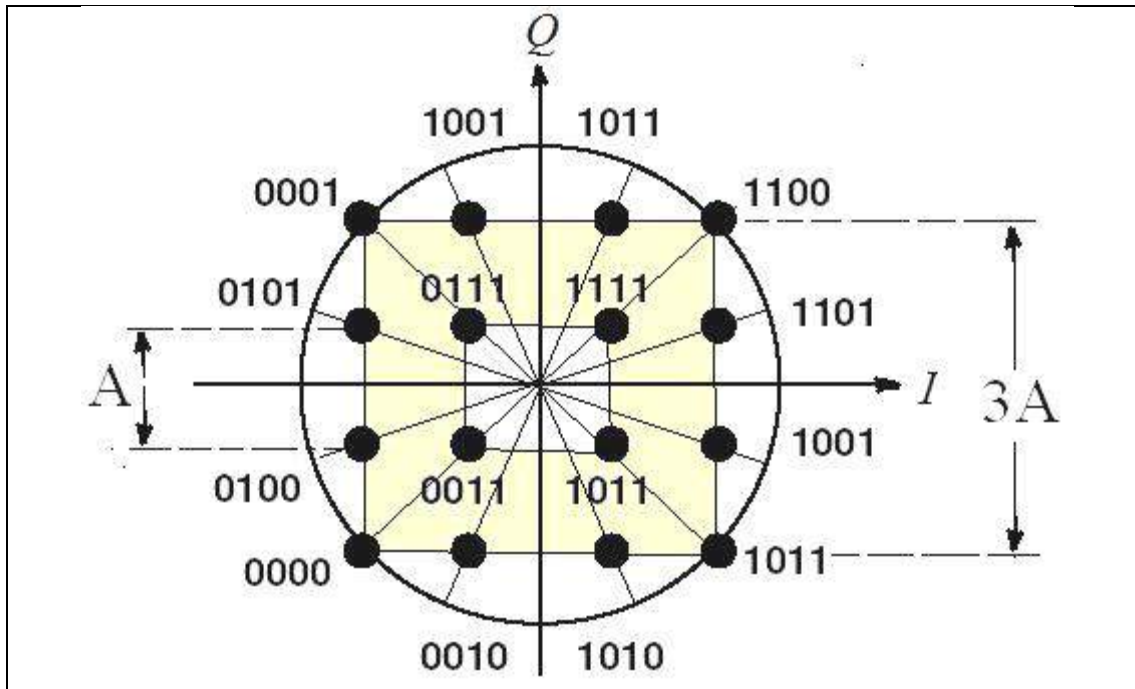


Fig. 5-33. Constellation diagram of 16QAM. Note that there are 16 symbols.

The carrier power required to transmit the highest-amplitude symbols will be:

$$P_{max} \propto A^2(\sqrt{M}-1)^2 \quad (5-16a)$$

When M is reasonably large this will approximate as follows:

$$P_{max} \propto M \quad (5-16b)$$

However the number of bits per symbol, m , varies with the number of symbols, n according to the following relation:

$$M = 2^m \quad (5-17)$$

Hence we can expect that the peak required carrier power will (approximately) tend to vary with the number of bits per symbol. For this

reason the peak powers required for QAM tends to rise steeply as we wish to convey more bits per symbol. Hence in practice we may wish to avoid choosing too high a value for m , and hence for M .

5-8.2. QAM in the Time Domain

The QAM sent signal, $s(t)$, is given by the following equation.

$$s(t) = \sum_{n=-\infty}^{\infty} [v_c[n] \cdot h_t(t - nT_s) \cos(2\pi f_0 t) - v_s[n] \cdot h_t(t - nT_s) \sin(2\pi f_0 t)] \quad (5-18)$$

5-8.3. Implementation of QAM

Figures 5-34 depicts the QAM transmitter and receiver structures. The method of carrier acquisition (recovery) is not shown in the receiver block diagram. This ensures that the oscillator, which supplies the local carrier signal, is synchronized to the transmitter signal in both frequency and phase.

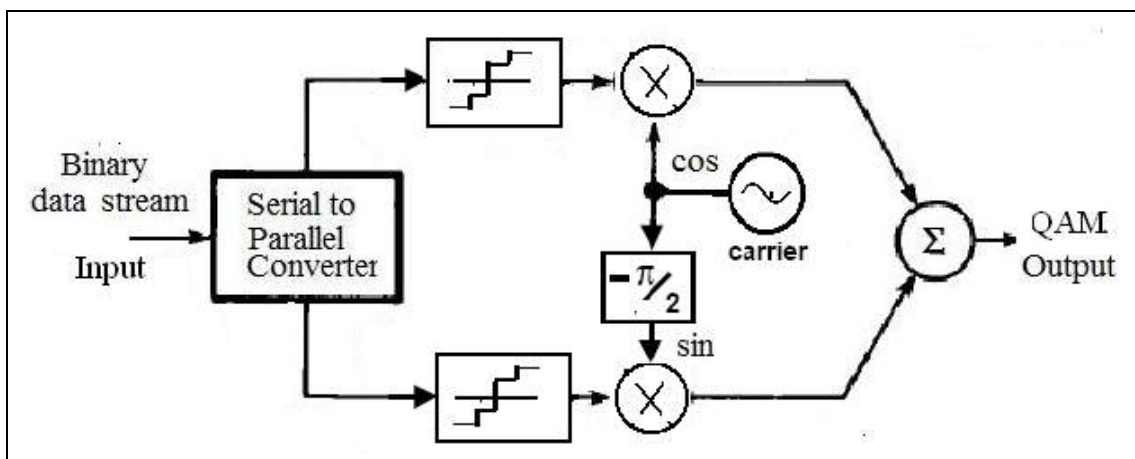


Fig. 5-34(a). Simplified block diagram of a QAM transmitter.

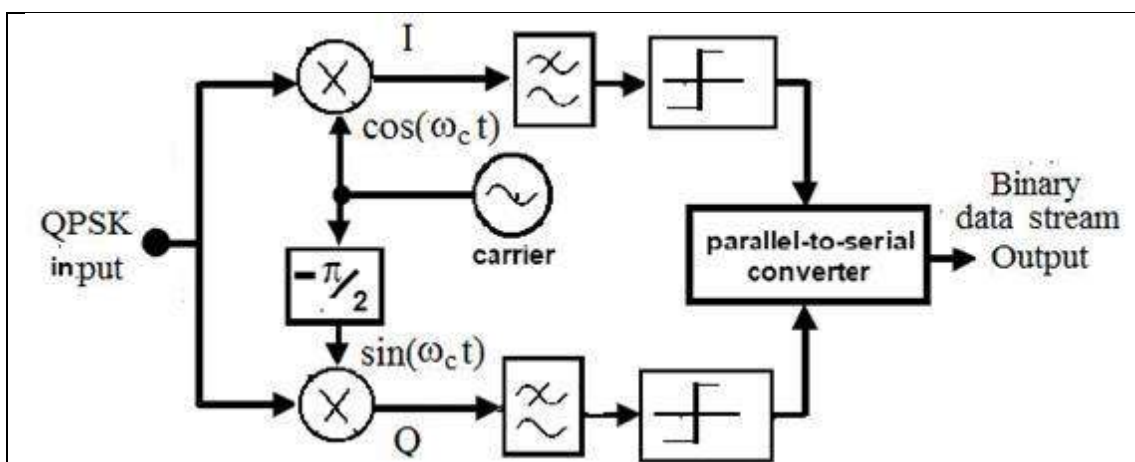


Fig. 5-34(b). Simplified block diagram of a QAM receiver.

5-9. Orthogonal Frequency-Division Multiplexing (OFDM)

OFDM is a frequency-division multiplexing (FDM) scheme utilized as a digital multi-carrier **modulation method**. A large number of closely-spaced orthogonal *sub-carriers* are used to carry data.. The primary advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions, such as attenuation, narrow-band interference and frequency-selective fading due to multipath. OFDM is a subset of frequency division multiplexing (FDM), in which a single channel utilizes multiple sub-carriers on adjacent frequencies. In addition, the sub-carriers in an OFDM system are overlapping to maximize spectral efficiency. Overlapping of adjacent channels can interfere with one another. However, sub-carriers in OFDM systems are precisely orthogonal to one another. Thus, they are able to **overlap without interfering**. As a result, OFDM systems are able to maximize spectral efficiency without adjacent channel interference. The frequency domain of an OFDM system is represented in the diagram below

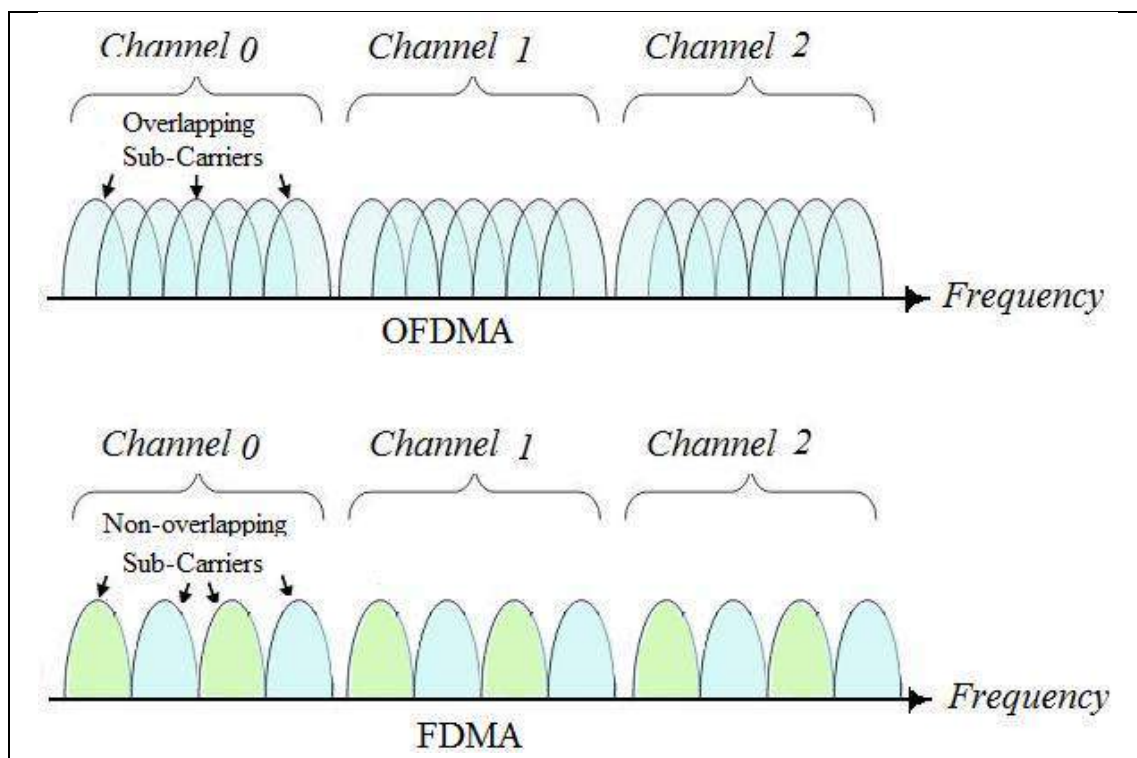


Fig. 5-35. Spectrum of the OFDM techniques.

5-9.1. OFDM Signals in the Time Domain

The orthogonal sinusoidal used in OFDM can be expressed mathematically as follows:

$$g_k(t) = (1/\sqrt{T_s}) \cdot \exp [j 2 \pi k t / T_s] \cdot w(t) \quad (5-19)$$

where k is integer = $0,1,2,\dots$, $w(t) = u(t) - u(t-T_s)$ is a rectangular window function and $u(t)$ is the unit step function.

It can be mathematically shown that if $k \neq q$ then the following integration (over complete period $0 \rightarrow T_s$) is null

$$\int a \sin(2\pi f_k t + \phi_k) + b \cos(2\pi f_q t + \phi_q) = 0 \quad (5-20)$$

This is so regardless of the values we choose for the phases ϕ_k and ϕ_q .

5-9.2. Implementation of OFDMA

The following two figures depict the structure of OFDMA transmitter and receiver, respectively. In the OFDMA transmitter, the OFDM carrier signal is the sum of a number of orthogonal sub-carriers, with baseband data on each sub-carrier being independently modulated commonly using quadrature amplitude modulation (QAM) or phase-shift keying (PSK). This composite baseband signal is typically used to modulate a main RF carrier. By inverse multiplexing, these are first demultiplexed into N parallel streams, and each one mapped to a symbol stream using some modulation constellation (QAM, PSK, etc.). Note that the constellations may be different, so some streams may carry a higher bit-rate than others.

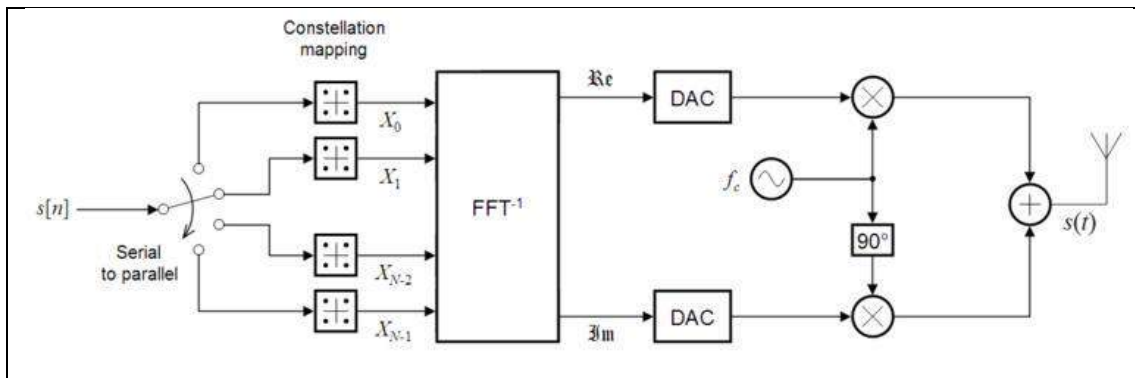


Fig. 5-36(a). Block diagram of an OFDMA transmitter

An inverse FFT is computed on each set of symbols, giving a set of complex time-domain samples. These samples are then quadrature-mixed to passband in the standard way. The real and imaginary components are first converted to the analogue domain using digital-to-analogue converters (DACs); the analogue signals are then used to modulate cosine and sine waves at the carrier frequency, f_c , respectively. These signals are then summed to give the transmission signal, $s(t)$.

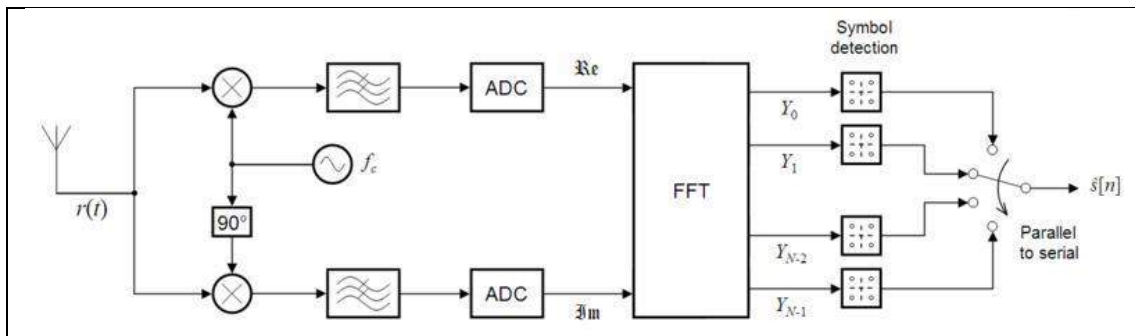


Fig. 5-36(b). Block diagram of an OFDMA receiver

In the OFDMA receiver side, the receiver picks up the signal $r(t)$, which is then quadrature-mixed down to baseband using cosine and sine waves at the carrier frequency. This also creates signals centered on $2f_c$, so low-pass filters are used to reject these. The baseband signals are then sampled and digitized using analogue-to-digital converters (ADCs), and a forward FFT is used to convert back to the frequency domain. This returns N parallel streams, each of which is converted to a binary stream using an appropriate symbol detector. These streams are then recombined into a serial stream, $s[n]$, which is an estimate of the original binary stream at the transmitter. Note that OFDM channels are different from band-limited FDM channels how they apply a pulse-shaping filter. With FDM systems, a *sinc*-shaped pulse (or cosine-raised filter) is applied in the time domain to shape each individual symbol and prevent ISI. With OFDM systems, a *sinc*-shaped pulse (or cosine-raised filter) is applied in the frequency domain of each channel. As a result, each sub-carrier remains orthogonal to one another.

5-9.3. OFDM Variants

There are several other variants of OFDM for which the initials are seen in the technical literature. These follow the basic format for OFDM, but have additional attributes or variations:

- **COFDM:** Coded Orthogonal frequency division multiplex. A form of OFDM where error correction coding is incorporated into the signal.
- **Flash OFDM:** This is a variant of OFDM that was developed by Flarion and it is a fast hopped form of OFDM. It uses multiple tones and fast hopping to spread signals over a given spectrum band.
- **OFDMA:** Orthogonal frequency division multiple access. A scheme used to provide a multiple access capability for applications such as cellular telecommunications when using OFDM technologies.

- **VOFDM:** Vector OFDM. This form of OFDM uses the concept of MIMO technology. It is being developed by CISCO Systems. MIMO stands for Multiple Input Multiple output and it uses multiple antennas to transmit and receive the signals so that multi-path effects can be utilised to enhance the signal reception and improve the transmission speeds that can be supported.
- **WOFDM:** Wideband OFDM. The concept of this form of OFDM is that it uses a degree of spacing between the channels that is large enough that any frequency errors between transmitter and receiver do not affect the performance. It is particularly applicable to Wi-Fi systems.

Each of these forms of OFDM utilise the same basic concept of using close spaced orthogonal carriers each carrying low data rate signals. During the demodulation phase the data is then combined to provide the complete signal.

5-9.4. OFDM Applications

OFDM is more commonly referred to as **COFDM** or Coded Orthogonal Frequency Division Multiplexing, which refers to the way the system is modified and used in practice. The following list is a summary of existing OFDM-based standards.

A. Cable Communication:

- ADSL and VDSL broadband access via telephone copper wires,
- Power-line communication (PLC).

B. Wireless Communication:

- Wireless LAN (**WLAN**) radio interfaces IEEE 802.11a, 11g, 11n.
- Digital radio systems, like DAB, T-DMB and ISDB-TSB.
- Terrestrial digital TV system **DVB-T**.
- Terrestrial mobile TV systems DVB-H, T-DMB, and ISDB-T.
- Cellular communication systems Flash-OFDM and High Speed OFDM Packet Access (**HSOPA**), and 3GPP.
- Fixed broadband wireless access (**BWA**) standard IEEE 802.16 (**WiMAX**) and Wireless MAN.
- Mobile broadband wireless access (**MBWA**) standards IEEE 802.20, and IEEE 802.16e (Mobile **WiMAX**).
- Wireless Personal Area Network (**PAN**), and Ultra wideband (**UWB**) IEEE 802.15.3a systems.

OFDM is considered as a **modulation technique** rather than a multiplex technique, since it transfers one bit stream over one communication channel using one sequence of the so-called OFDM symbols. OFDM is used in high speed mobiles and high data rate communication systems. OFDM is also used in so many applications, such as digital video broadcasting (**DVB**), wireless local area networks (**WLAN**) and asynchronous digital subscriber lines (**ADSL**). OFDM can be extended to multi-user channel access method in the Orthogonal Frequency Division Multiple Access (**OFDMA**) and MC-OFDM schemes, allowing several users to share the same physical medium by giving different sub-carriers to different users.

5-10. Continuous Phase Modulation (CPM)

The continuous phase modulation (CPM) is a digital modulation method, which is commonly used in wireless modems. In contrast to other coherent digital phase modulation techniques where the carrier phase abruptly resets to zero at the start of every symbol (e.g. M-PSK), with CPM the carrier phase is modulated in a continuous manner. For instance, with QPSK the carrier jumps instantaneously from a *sine* to a *cosine* (90° phase shift) whenever one bit of the current symbol differs from the bits of the previous symbol. This discontinuity requires a relatively large percentage of the power to occur outside of the intended band, leading to poor spectral efficiency. Furthermore, CPM is typically implemented as a constant-envelope waveform, i.e. the transmitted carrier power is constant. Therefore, CPM is attractive because the phase continuity yields high spectral efficiency, and the constant-envelope yields good power efficiency. The primary drawback is the complexity of implementation of the CPM receiver system.

5-10-1. Minimum-Shift Keying (MSK)

The minimum-shift keying (MSK) is a particular case of continuous phase modulation (CPM). Actually, MSK is a particular case of a sub-family of CPM known as continuous-phase frequency-shift keying (CPFSK) which is defined by a linearly increasing phase pulse, of one symbol-time duration.

MSK is a continuous phase modulation scheme where the modulated carrier contains no phase discontinuities and frequency changes occur at the carrier zero crossings. MSK is unique due to the relationship between the frequency of a logical zero and one: the difference between the frequency of a logical zero and a logical one is minimum and always equal to half the data rate (R_b). The modulation index for FSK, from which MSK may be derived, is defined as follows:

$$m = \Delta f / 2R_b = T_b \Delta f \quad (5-21a)$$

where T_b is the bit duration ($R_b = 1/2T_b$). For MSK the modulation index $m = 1/2$, such that Δf is minimum and given by:

$$\Delta f = |f_{logic1} - f_{logic0}| = m / T_b = 1/2T_b \quad (5-21b)$$

The resulting MSK signal $s(t)$ is represented by the formula:

$$s(t) = a_I(t) \cos\left(\frac{\pi t}{2T_b}\right) \cos(2\pi f_c t) + a_Q(t) \sin\left(\frac{\pi t}{2T_b}\right) \sin(2\pi f_c t) \quad (5-22)$$

where $a_I(t)$ and $a_Q(t)$ encode the *even* and *odd* information respectively with a sequence of square pulses as of duration $2T_b$. Using the trigonometric identity, this can be rewritten in a form where the phase and frequency modulation is more obvious,

$$s(t) = \cos\left[2\pi f_c t + b_k(t)\left(\frac{\pi t}{2T_b}\right) + \phi_k\right] \quad (5-23)$$

where $b_k(t)$ is +1 when $a_I(t) = a_Q(t)$ and -1 if they are of opposite signs, and ϕ_k is 0 if $a_I(t)$ is 1, and π otherwise. Therefore, the signal is modulated in frequency and phase, and the phase continuously and linearly changes.

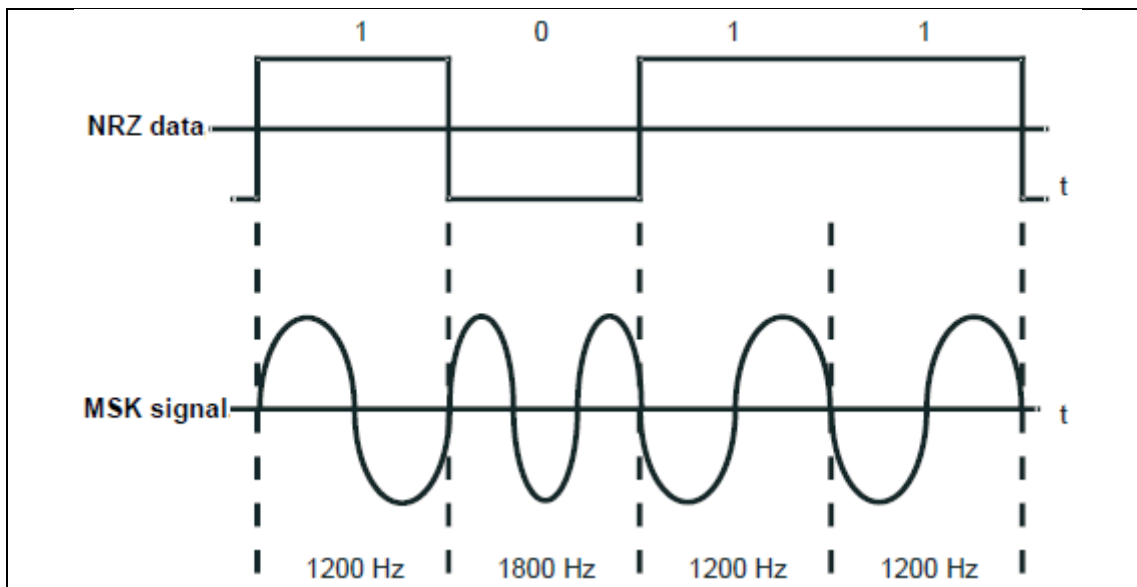


Fig. 5-37. Illustration of the MSK waveforms, showing the input pulses and output MSK signal.

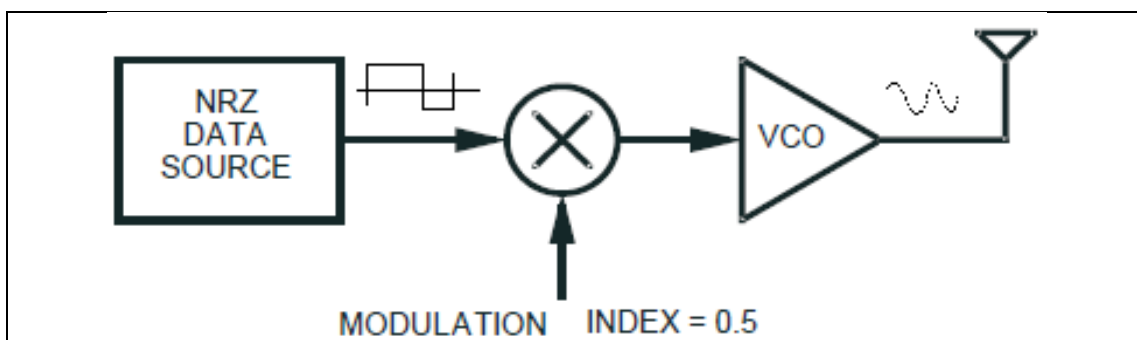


Fig. 5-38. Illustration of the MSK direct modulation.

The most natural choice for MSK demodulation is to use a simple BFSK detector. It has then the same bit error probability as ordinary BPSK.

5-10-2. Gaussian MSK (GMSK)

The so-called Gaussian minimum-shift keying (GMSK) is a form of continuous-phase digital modulation. It has advantages of being able to carry digital modulation while still using the spectrum efficiently. The GMSK is similar to MSK, but uses a Gaussian filter to limit the bandwidth of transmitted sinusoidal pulses. As shown in the following figure, the spectrum of an MSK signal shows sidebands extending well beyond a bandwidth equal to the data rate. This can be reduced by passing the modulating signal through a low pass filter prior to applying it to the carrier. The requirements for the filter are that it should have a sharp cut-off, narrow bandwidth and its impulse response should show no overshoot. The ideal filter is known as a Gaussian filter which has a Gaussian shaped response to an impulse and no ringing. In this way the basic MSK signal is converted to GMSK modulation.

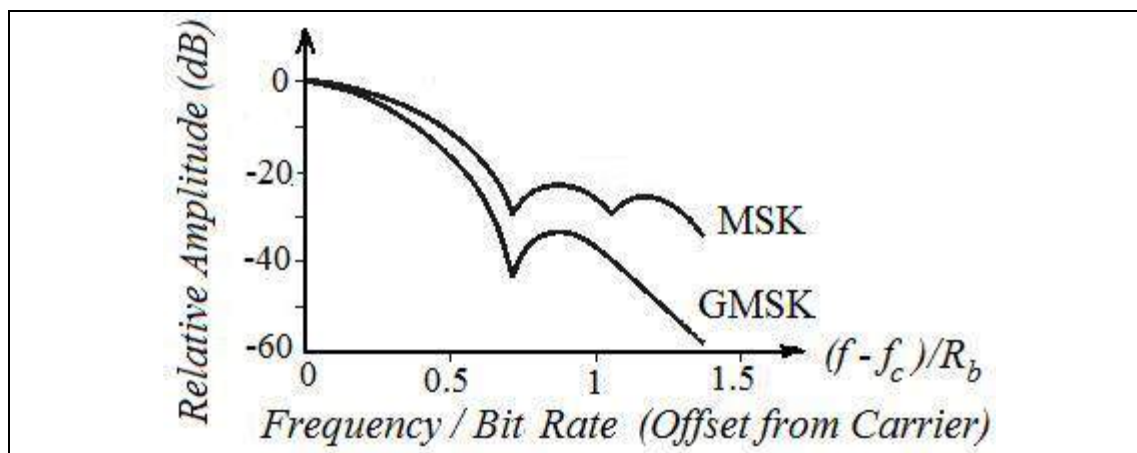


Fig. 5-39. Spectral density of MSK and GMSK systems.

Many blocks of a typical GMSK radio such as the frequency synthesizer (FS), have non-ideal behavior. In particular, the synthesizer presents a problem for GMSK modulation. Most synthesizers will not respond to low frequency signals (a typical synthesizer effectively has a high-pass filter characteristic). However, there exist two common modulation methods, which help considerably where the non-ideal behavior of the synthesizer is concerned, namely: Two-point modulation and Quadrature modulation. As shown in figure 5-40, the two point modulation circumvents the synthesizer problem by splitting the Gaussian filtered signal; one portion is directed to the VCO modulation input, the other is used to modulate a temperature compensated crystal oscillator (TCXO).

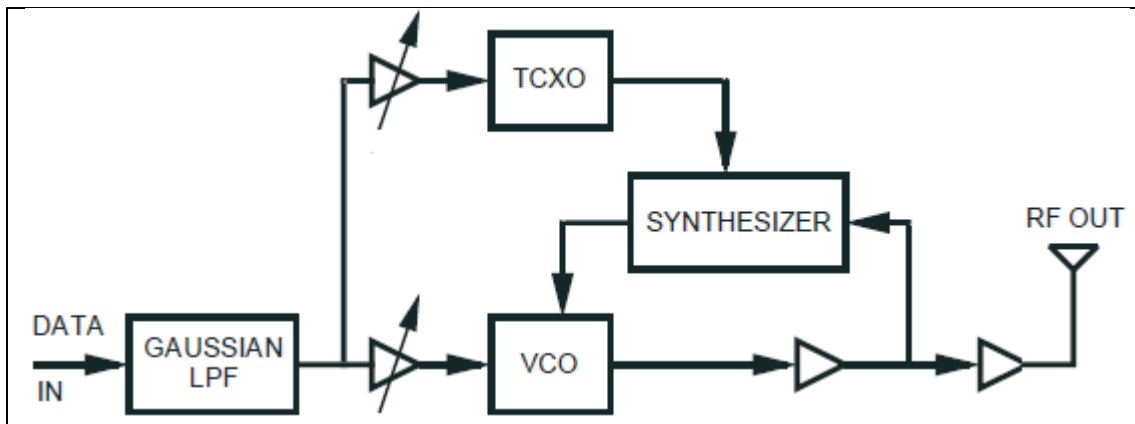


Fig. 5-40. Illustration of the two-point GMSK modulator.

Quadrature (I and Q) modulation can also be effective in eliminating synthesizer problems. In the I and Q modulation, the Gaussian filtered data signal is separated into in-phase (I) and quadrature phase (Q) components, as shown, in figure 5-41. The modulated RF signal is created by mixing the I and Q components up to the frequency of the RF carrier, where they are summed together. The role of the synthesizer has now been reduced to merely changing carrier frequency for channel selection. The key to optimum performance with quadrature modulation is accurate creation of I and Q components. Baseband I and Q signals can be created by using an all-pass phase shifting network. This network must maintain a 90 degree phase relationship between I and Q signals for all frequencies in the band of interest. There are several advantages to the use of GMSK modulation for a radio communications system. One is obviously the improved spectral efficiency when compared to other phase shift keyed modes.

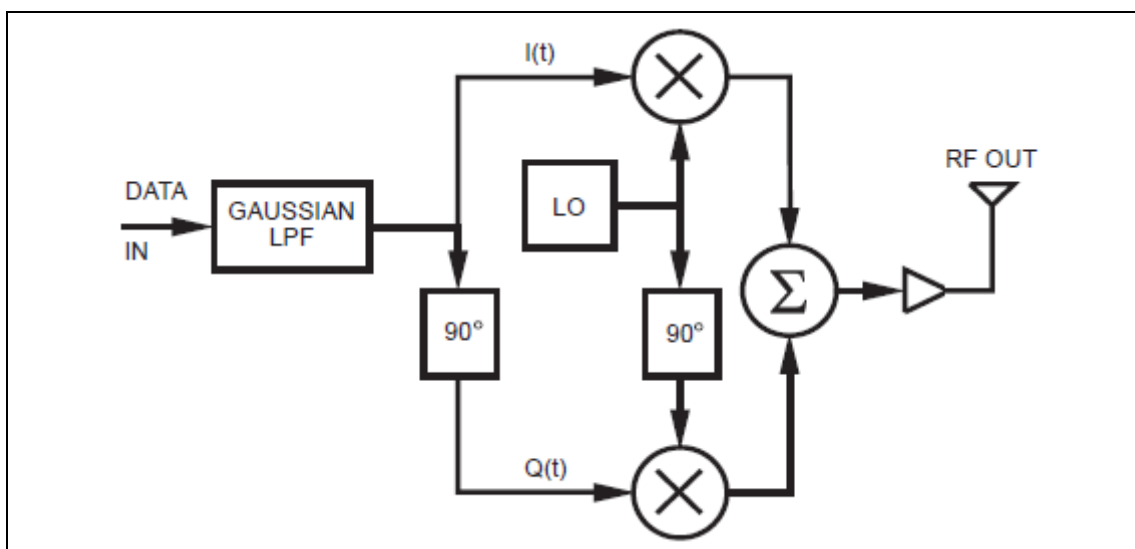


Fig. 5-41. Illustration of the Quadrature GMSK modulator.

5-10-3. Very-Minimum Shift Keying (VMSK)

The Very Minimum Shift Keying (VMSK) is one of several digital modulation methods claimed to send high speed digital data through very low bandwidth (or ultra narrowband UNB) channels.

We can look at VMSK as the sum of two signals: a 50% duty-cycle square wave clock at the data rate, plus one of two narrow pulses depending on the bit being sent. The pulse for a "0" bit advances the negative-going clock transition in the middle of the data bit, and the pulse for a "1" retards it.

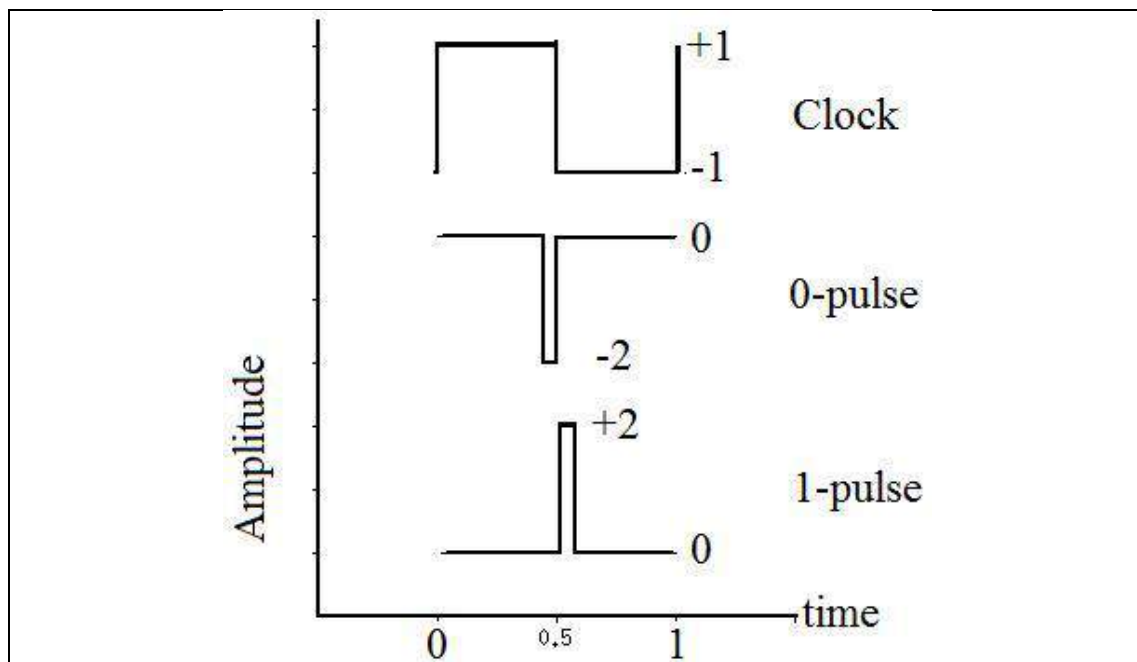


Fig. 5-42. Spectral density of MSK and GMSK systems.

A typical claim is a data rate of 6 Mbit/s in a bandwidth of 1 kHz using the same transmitter power than conventional schemes. VMSK was introduced in 1995 and is still a subject of controversial discussions, because its claim to achieve high speed data transfer over a very narrow bandwidth is in direct violation of the mathematical principles of digital communications of Harry **Nyquist** and Claude Shannon.

5-11. Spread Spectrum Modulation Techniques

The spread spectrum (SS) is a technique in which a telecommunication signal is transmitted on a bandwidth considerably larger than the frequency content of the original information. The spread spectrum modulation techniques are known since long time but were only used in military communication systems. Figure 5-43 represents a narrow band signal in the frequency domain. Such narrowband signal can be easily detected, intercepted and jammed by any other signal in the same band. In fact, the signal can be intercepted since the frequency band is fixed and narrow (i.e. easy to detect).

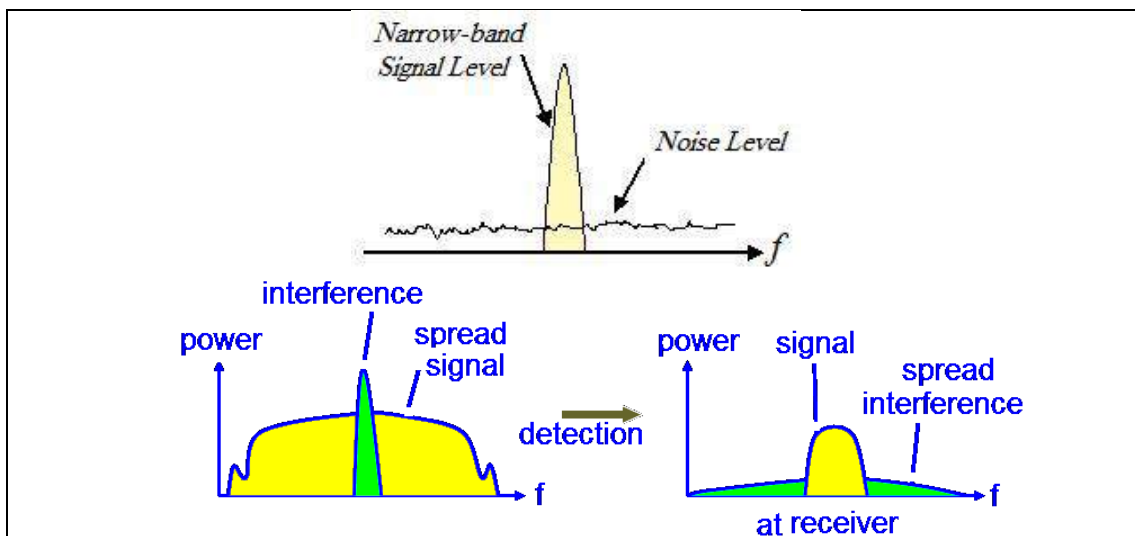


Fig. 5-43. Illustration of narrow-band signals, which are easy to be detected (from noise), intercepted and jammed.

The spread-spectrum techniques *spread* radio signal over a wide frequency range higher than minimum requirement. The core principle of spread spectrum is the use of noise-like carrier waves. The following figure depicts a typical communication system with the difference that the modulator/demodulator has as input the spreading generator.

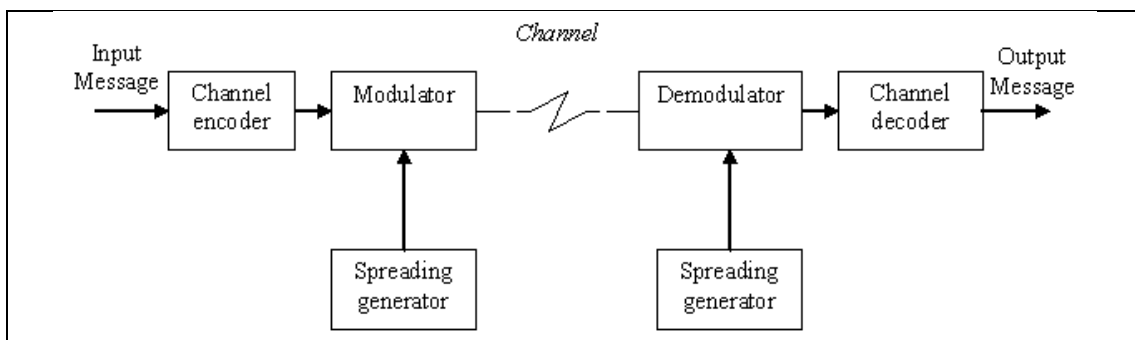


Fig. 5-44. Block diagram of a spread spectrum communication system.

In practice, there exist two main spread spectrum techniques:

- Direct sequence spread spectrum (**DSSS**)
- Frequency hopping spread spectrum (**FHSS**)

In direct sequence spread spectrum (DSSS) technique, rapid phase transition pseudo-noise signals are used to spread data over a larger bandwidth. In frequency hopping spread spectrum (FHSS), the narrow band signal is made to jump in random narrow bands within a larger bandwidth. Both of the two methods have several applications. In particular, DSSS technique is mostly used in the industry (CDMA, UMTS mobiles, GPS, and WLAN).

5-11.1. Direct Sequence Spread Spectrum (DSSS)

The direct-sequence spread spectrum (**DSSS**) is a digital modulation technique; in which rapid phase transition pseudo-noise signals are used to spread data over a larger bandwidth. The name 'spread spectrum' comes from the fact that the carrier signals occur over the full bandwidth (spectrum) of a device's transmitting frequency. Spread spectrum increases BW of message signal by a factor N , called **Processing Gain** such that $N = 10 \log(B_c / B_s)$.

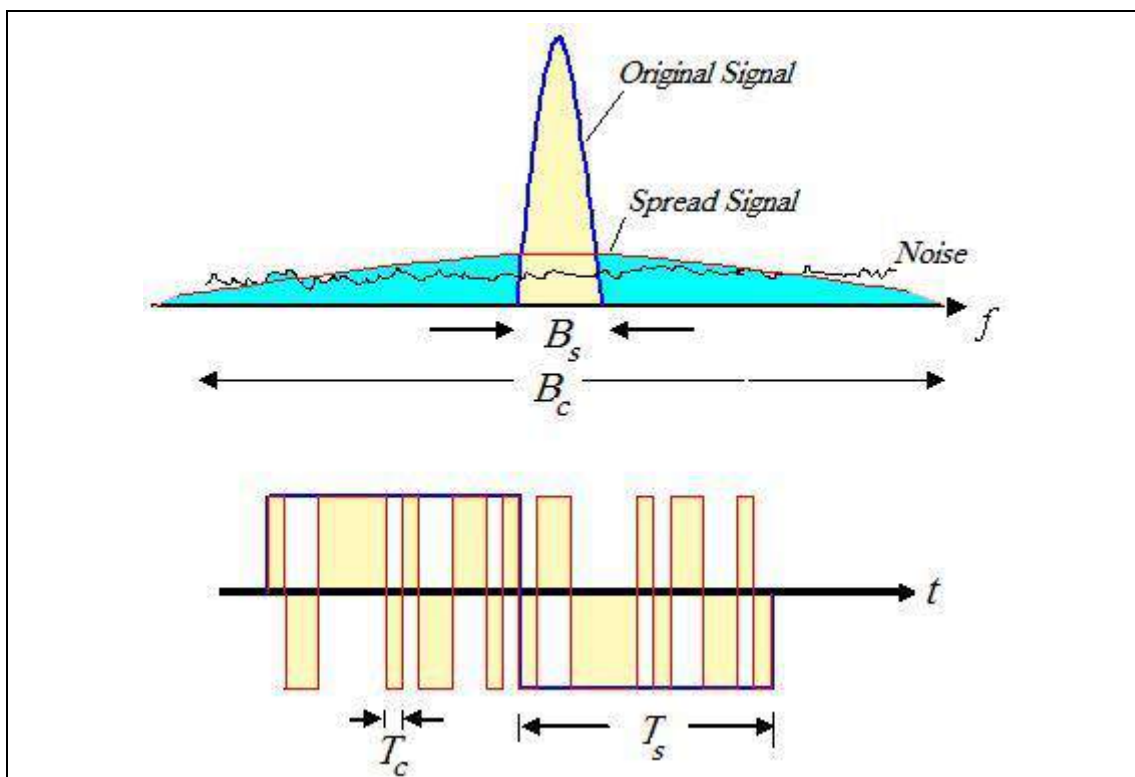


Fig. 5-45. Illustration of how the low frequency data signal (T_s) can be spread over a larger bandwidth, by multiplying it with a fast pseudo-random code.

5-11.1.A. Features of DSSS

In DSSS, each information bit is modulated by a sequence of much faster pseudo-noise (PN) code symbols called "**chips**". Therefore, the chip rate is much higher than the information signal bit rate. As shown in figure 5-41, the rapid phase transition of chips (chip rate $1/T_c$) signal has a larger bandwidth and behaves similar to noise in such a way that their spectrums are similar for bandwidth in scope. In fact, the power density amplitude of the spread spectrum output signal is similar to the noise floor. The resultant signal is *hidden* under the noise. DSSS uses a signal structure in which the sequence of chips produced by the transmitter is known *a priori* by the receiver. The receiver can then use the same *PN sequence* to counteract the effect of the **PN sequence** on the received signal in order to reconstruct the information signal.

5-11.1.B. DSSS Transmission Method

Direct-sequence spread-spectrum transmissions multiply the data being transmitted by a noise-like signal. This noise signal is actually a pseudorandom sequence of binary (1 and -1) values, at a frequency much higher than that of the original data signal, thereby spreading the energy of the original signal into a much wider band. The resulting signal resembles white noise. However, this noise-like signal can be used to exactly reconstruct the original data at the receiving end, by multiplying it by the same pseudorandom sequence. This process, which is known as *de-spreading*, constitutes a correlation of the transmitted PN sequence with the assumed sequence of the receiver.

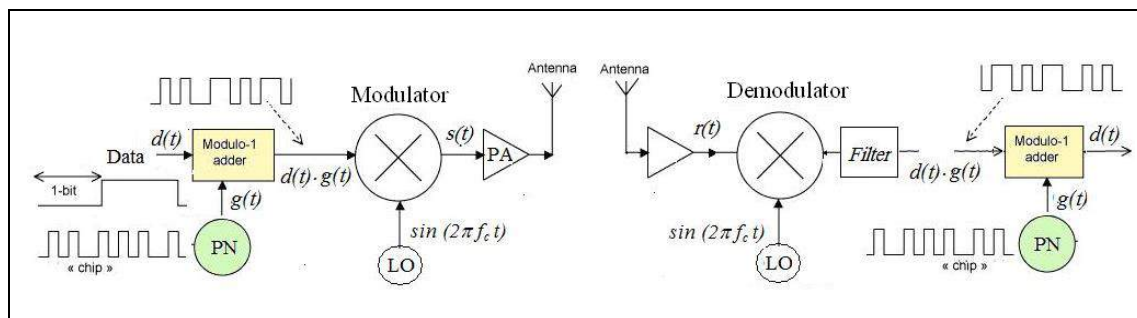


Fig. 5-46. Block diagram of a DSSS communication system.

For de-spreading to work correctly, the transmitter and receiver sequences should be synchronized. This requires the receiver to synchronize its sequence with the transmitter sequence via some sort of timing process. The resulting effect of enhancing signal to noise ratio on the channel is called *processing gain*. This effect can be made larger by employing a longer PN sequence and more chips per bit. If an undesired

transmitter transmits on the same channel but with a different PN sequence, the de-spreading process results in no processing gain. This effect is the basis for the code division multiple access (**CDMA**) property of DSSS, which allows multiple transmitters to share the same channel. As shown in figure 5-47, the DSSS is usually combined with another type of digital modulation, such as the QPSK.

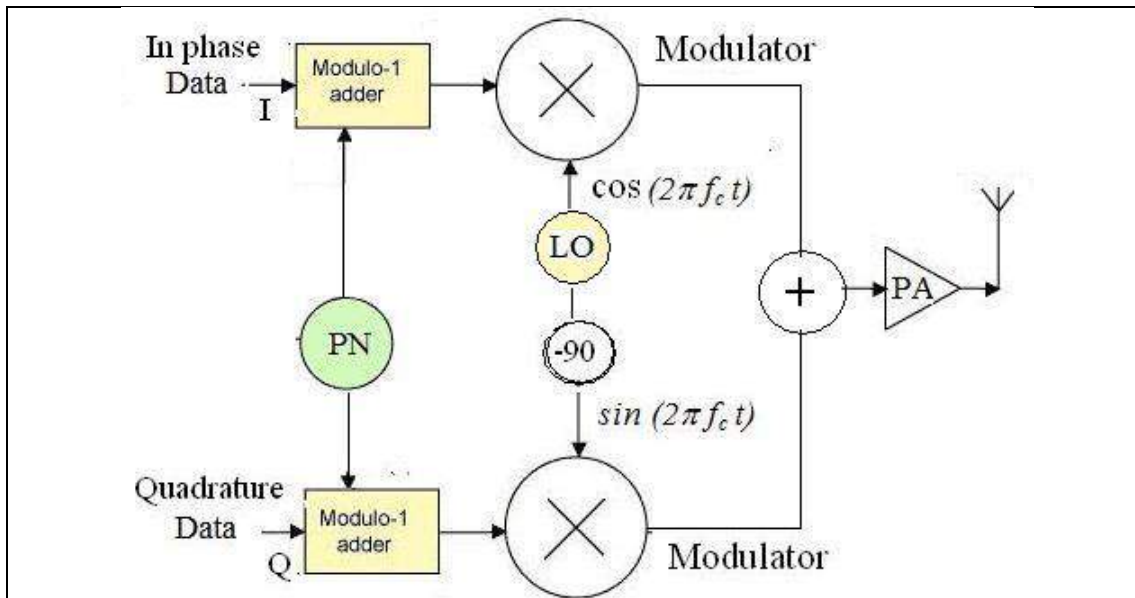


Fig. 5-47. Combining DSSS with QPSK. modulator

5-11.2. Frequency-hopping Spread Spectrum (FHSS)

Frequency-hopping spread spectrum (**FHSS**) is a method of transmitting radio signals by rapidly switching a carrier among many frequencies, using a pseudorandom sequence known to both transmitter and receiver.

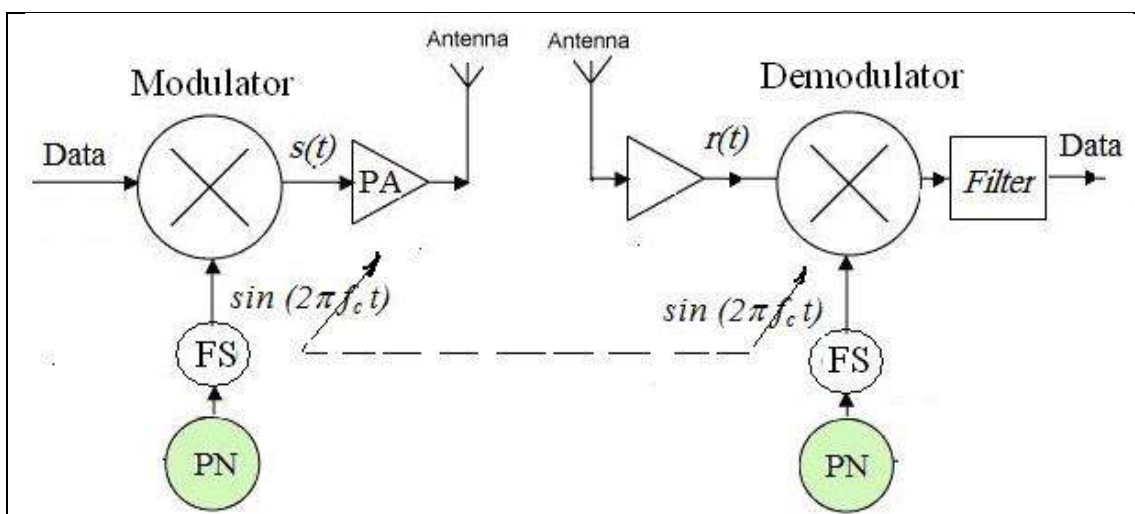


Fig. 5-48. Block diagram of a FHSS communication system

Therefore, the frequency-hopping spreads the spectrum pseudo-randomly, via a frequency synthesizer (**FS**), instead of adding pseudo-random noise directly to the data. This results in a uniform frequency distribution whose width is determined by the output range of the pseudo-random number, which is fed to the frequency synthesizer.

5-11.2.A. Features of FHSS

The FHSS transmission offers three main advantages over a fixed-frequency transmission:

- Spread-spectrum signals are highly resistant to narrowband interference. The process of re-collecting a spread signal spreads out the interfering signal, causing it to recede into the background.
- Spread-spectrum signals are difficult to intercept. A frequency-hop spread-spectrum signal simply sounds like an increase in the background noise to a narrowband receiver.
- The FHSS transmissions can share a frequency band with many types of conventional transmissions with minimal interference. The spread-spectrum signals add minimal noise to the narrow-frequency communications, and vice versa. As a result, bandwidth can be utilized more efficiently.

The overall bandwidth required for frequency hopping is much wider than that required to transmit the same information using only one carrier frequency. However, because transmission occurs only on a small portion of this bandwidth at any given time, the effective interference bandwidth is really the same. Whilst providing no extra protection against wideband thermal noise, the frequency-hopping approach does reduce the degradation caused by narrowband interferers.

One of the challenges of frequency-hopping systems is to synchronize the transmitter and receiver. One approach is to have a guarantee that the transmitter will use all the channels in a fixed period of time. The receiver can then find the transmitter by picking a random channel and listening for valid data on that channel. The transmitter's data is identified by a special sequence of data that is unlikely to occur over the segment of data for this channel and the segment can have a checksum for integrity and further identification. The transmitter and receiver can use fixed tables of channel sequences so that once synchronized they can maintain communication by following the table. On each channel segment, the transmitter can send its current location in the table

5-11.2.B. Adaptive Frequency-hopping Spread Spectrum (AFH)

The so-called Adaptive Frequency-hopping spread spectrum (**AFH**) is used in **Bluetooth** to improve resistance to radio frequency interference by avoiding using crowded frequencies in the hopping sequence. This sort of adaptive transmission is easier to implement with FHSS than with DSSS. The key idea behind AFH is to hop only over the “good” frequencies, by avoiding the frequency channels that are experiencing radio frequency interference from certain transmitter. Therefore, AFH should be associated by a mechanism for detecting good/bad channels.

5-11.3. Features of Spread Spectrum Techniques

The spread spectrum modulation techniques have the following features

5-11.3.A. Resistance of Spread Spectrum to Jamming (interference).

The DSSS technique is better at resisting continuous-time narrowband jamming, while FHSS is better at resisting pulse jamming. In DSSS systems, narrowband jamming affects detection performance about as much as if the amount of jamming power is spread over the whole signal bandwidth, when it will often not be much stronger than background noise. By contrast, in narrowband systems where the signal bandwidth is low, the received signal quality will be severely lowered if the jamming power happens to be concentrated on the signal bandwidth.

5-11.3.B. Resistance of Spread Spectrum to Eavesdropping.

The spreading code (in DSSS systems) or the frequency-hopping pattern (in FHSS systems) is often unknown by anyone for whom the signal is unintended, in which case it "encrypts" the signal and prevents the adversary from making sense of it. What's more, for a given noise power spectral density (PSD), spread-spectrum systems require the same amount of energy per bit before spreading as narrowband systems and therefore the same amount of power if the bitrate before spreading is the same. But since the signal power is spread over a large bandwidth, the signal PSD is much lower, often significantly lower than the noise PSD. Therefore, the adversary may be unable to determine if the signal exists at all. Note that these effects can also be achieved by using encryption and a very low-rate channel code, which can also be viewed as a spread-spectrum method, albeit more complex.

5-11.3.C. Resistance to Fading of Spread Spectrum.

The high bandwidth occupied by spread-spectrum signals offer some frequency diversity, i.e. it is unlikely that the signal would encounter

severe multipath fading over its whole bandwidth, and in other cases the signal can be detected using e.g. a Rake receiver.

5-11.3.D. Multiple Access Capability of Spread Spectrum.

Multiple users can transmit simultaneously on the same frequency (range) as long as they use different spreading codes.

5-12. Trellis Coded Modulation (TCM)

Trellis coded modulation (TCM) is a combination of digital modulation and encoding schemes which allows highly efficient transmission of information over band-limited channels such as telephone lines. TCM has been proposed by **Ungerboeck** in 1982, and now is a well-established technique in digital communications. Figure 5-4, which depicts the phase versus time of a digitally-modulated signal is called a “**trellis**” diagram, because it resembles a garden trellis. The trellis diagram shows time on the X-axis and phase on the Y-axis.

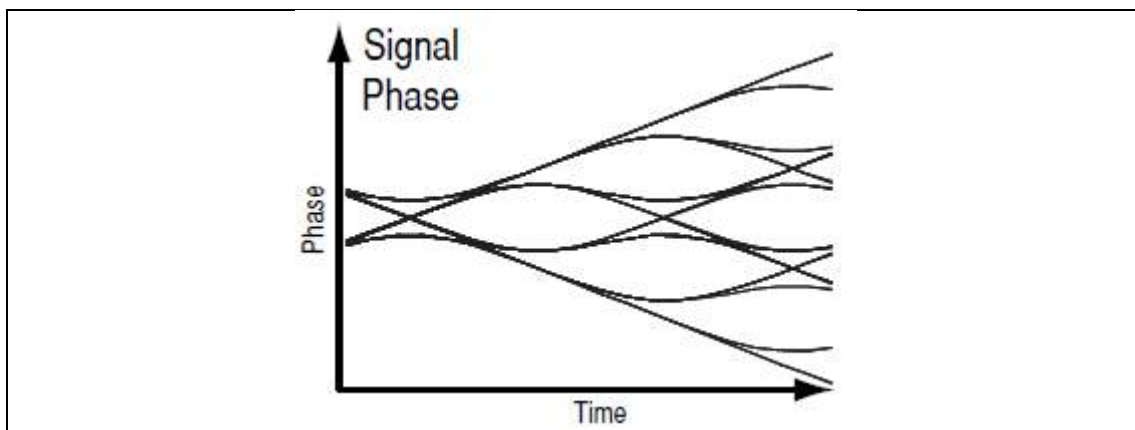


Fig. 5-49. Example of Trellis code modulation (TCM, showing the signal phase versus time of a GMSK signal.

The TCM scheme is basically a **convolutional** code. As shown in figure 5-50, the TCM system is a convolutional encoder of rate $R = k/k+1$ followed by a **mapper** of M -ary signal to map $M=2^k$ input points into a larger constellation of M^{2k+1} constellation points. Convolutional coding will be discussed in Chapter 6, when we talk about error correcting codes.

The key idea of TCM is termed **Mapping by Set Partitions**. The idea is to group the symbols in a tree like fashion then separate them into two limbs of equal size. At each limb of the tree, the symbols are further apart. Although it is hard to visualize in multi-dimensions, a simple one dimension example can illustrate the basic procedure. Suppose the symbols are located at $[1,2,3,4,\dots]$. Then take all odd symbols and place them in one group, and the even symbols in the second group. Take every other one for each group and repeat the procedure for each tree limb. Next, we assign the encoded bit stream onto the symbols in a very systematic procedure. Once the mapping procedure is achieved, the next step is to program the algorithm into a computer and let the computer search for the best codes. Even the most simple code (4 state) produces error rates nearly 1000 times lower than an equivalent uncoded system.

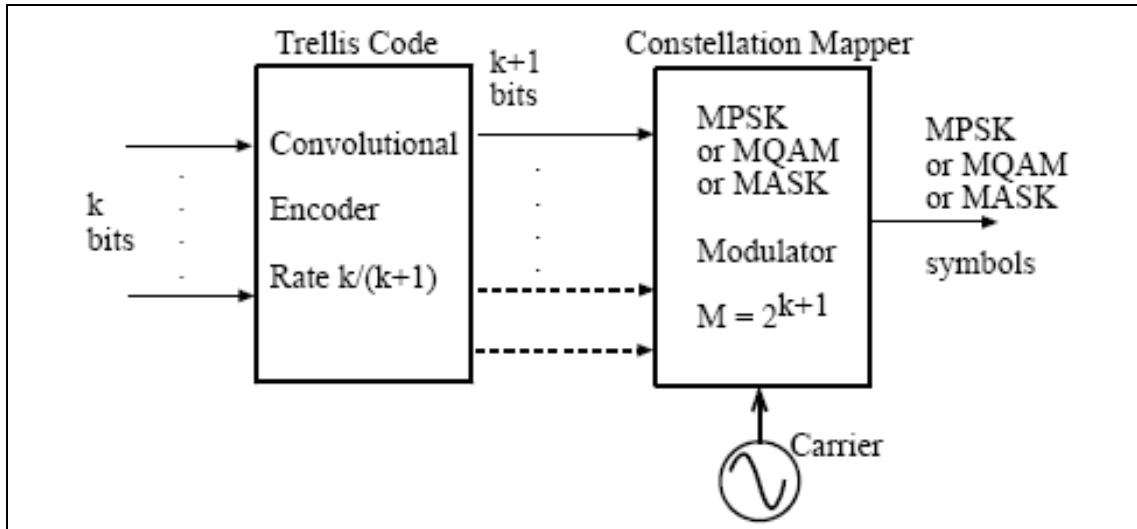


Fig. 5-50. Basic idea of Trellis code modulation (TCM).

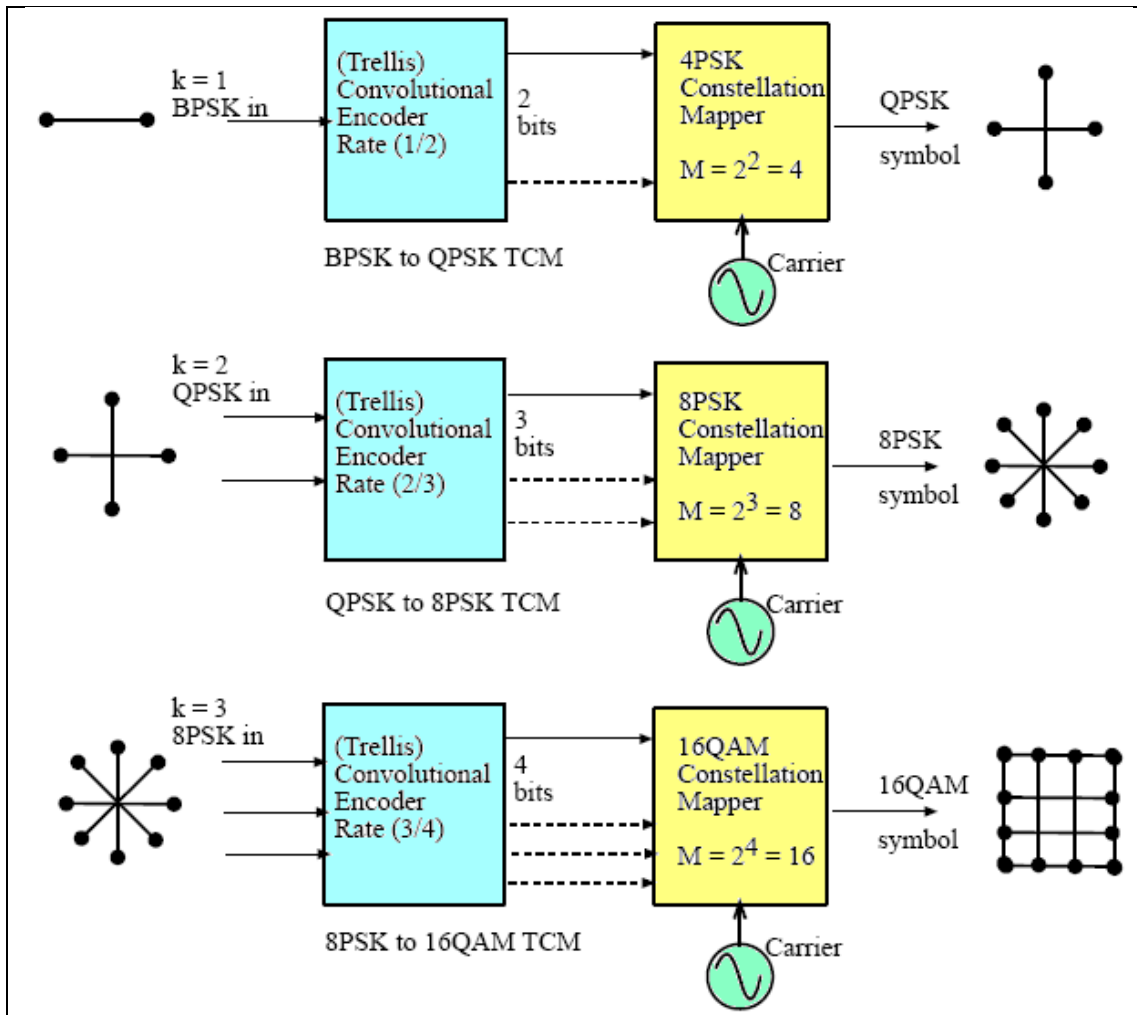


Fig. 5-51. Example showing the constellation doubling in TCM.

TCM has generated a continually growing interest, concerning its numerous applications, spanning high-rate digital transmission over voice circuits, digital microwave radio relay links, and satellite communications. In essence, it is a technique to obtain significant coding gains (3-6 dB) sacrificing neither data rate nor bandwidth. The following figure depicts the block diagram of a TCM modem. By 1990 the International Telecommunication Union (ITU) had adopted modem standards for the trellis-modulated modem at 14.4 kbit/s (2400 baud rate x 6 bits per symbol). Over the next years further advances in encoding, plus a corresponding increase in symbol rate, allowed modems to achieve rates up to 56 kbit/s. The most common trellis-modulated V.32 modems use two 5-dimensional symbols in a single lattice. This set uses 8, 16, or 32 state convolutional codes to squeeze the equivalent of 6 to 10 bits into each symbol sent by the modem.

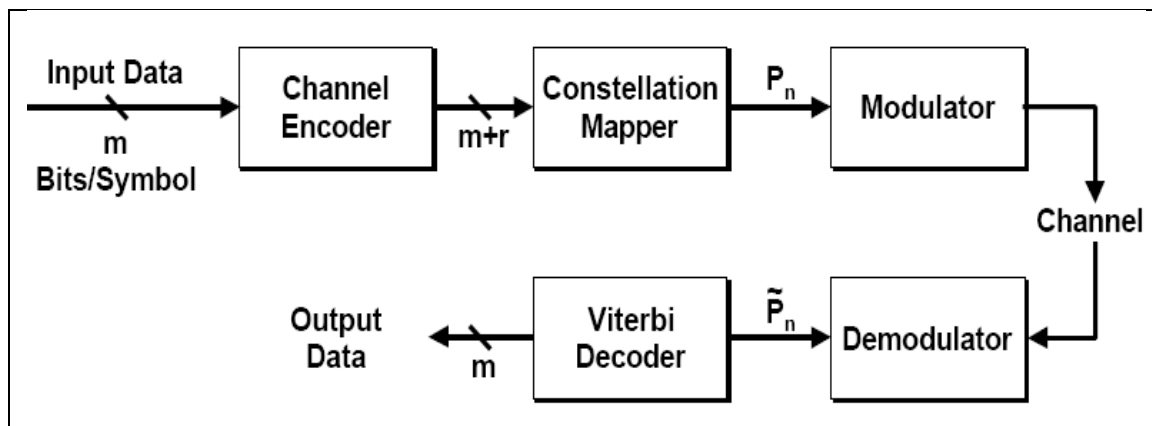


Fig. 5-52. Block diagram of a Trellis communication system.

Once manufacturers introduced modems with trellis modulation, transmission rates increased to the point where interactive transfer of multimedia over the telephone lines became feasible. For example, a 200 kB image and a 5 MB song could be downloaded in less than 1 minute and 30 minutes, respectively, thanks to trellis modulation techniques.

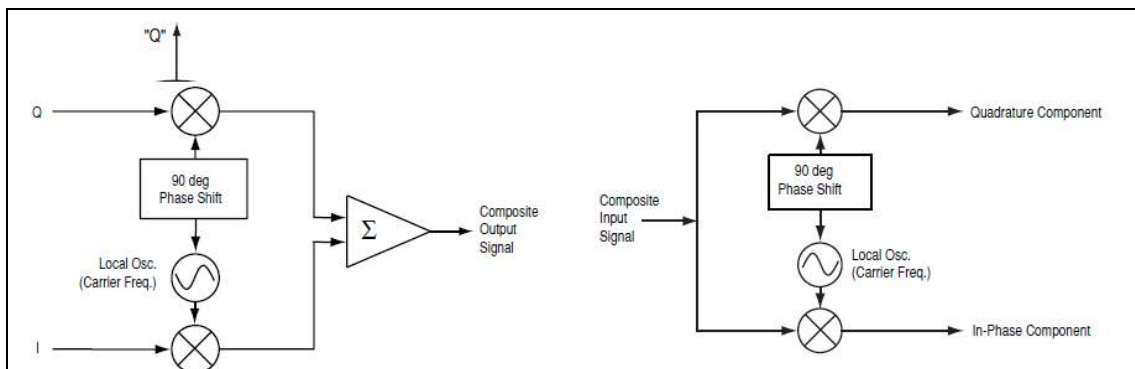
5-13. Summary

The aim of **digital modulation** is to transfer a digital bit stream over an analog bandpass channel, for example over the public switched telephone network or a limited radio frequency band.

The most fundamental digital modulation techniques are:

- **ASK**, where a finite number of amplitudes are used.
- **FSK**, where a finite number of frequencies are used.
- **PSK**, where a finite number of phases are used.
- **QAM**, where an in-phase signal (the **I** signal, cosine wave) and a quadrature phase signal (the **Q** signal, a sine wave) are amplitude modulated with a finite number of amplitudes.

Digital modulation is easy to accomplish with *I/Q* modulators. Most digital modulation maps the data to a number of discrete points on the *I/Q* plane. These are known as constellation points. As the signal moves from one point to another, the amplitude and phase modulation results.



At the **transmitter side** of a digital communication system, the general steps used by a modulator to transmit data are as follows:

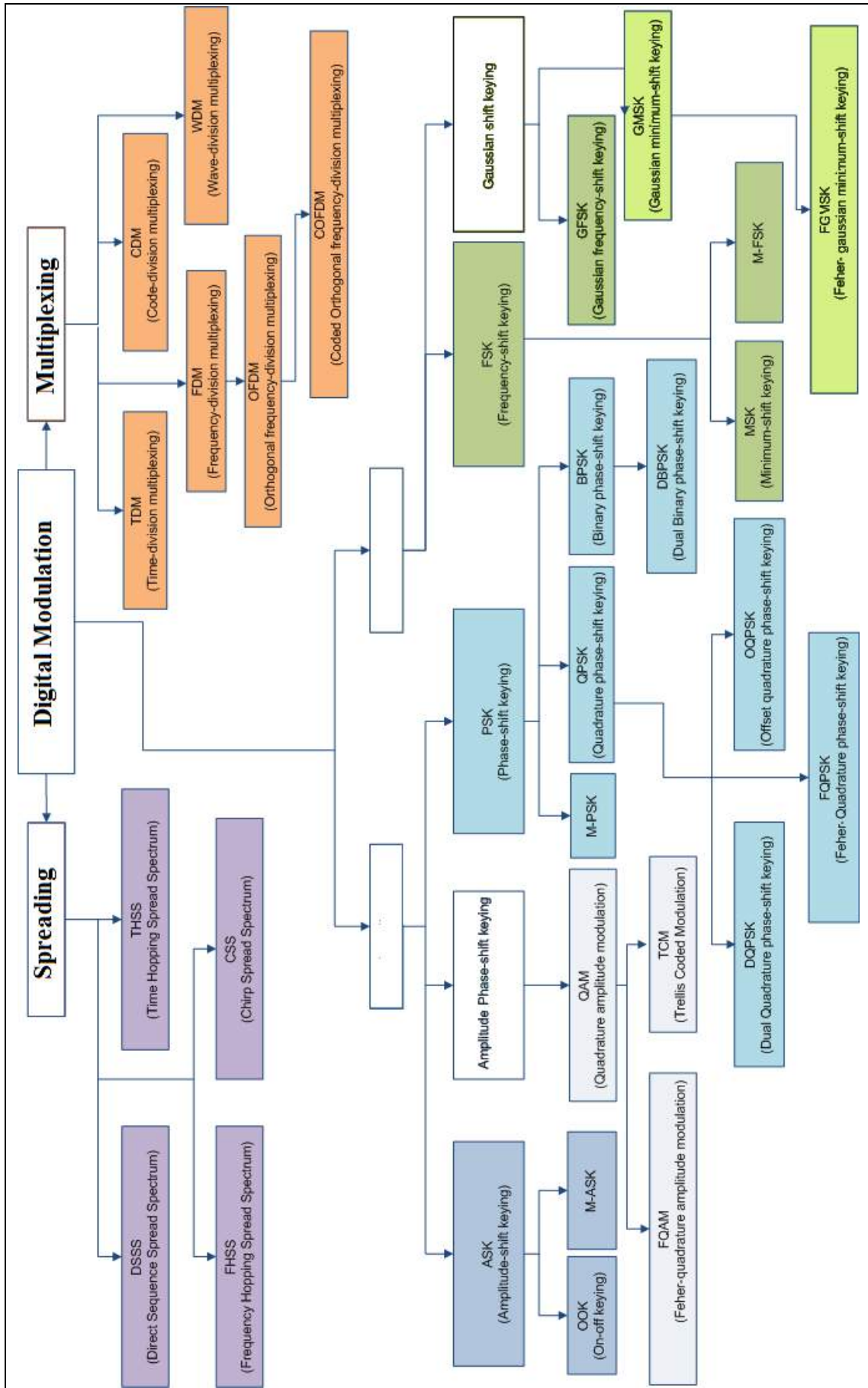
- Group the incoming data into code-words;
- Map the code-words to attributes, for example amplitudes of the **I** and **Q** signals, or frequency or phase values.
- Adapt pulse shaping or some other filtering to limit the bandwidth and form the spectrum, typically using digital signal processing
- Digital-to-analog conversion (DAC) of the **I** and **Q** signals. Sometimes the next step is also achieved using DSP, and then the DAC should be done after that.

- Modulate the high-frequency carrier waveform, resulting in that the equivalent low pass signal is frequency shifted into a modulated passband signal or RF signal
- Amplification and analog bandpass filtering to avoid harmonic distortion and periodic spectrum

At the **receiver side**, the digital demodulator typically performs the following steps:

- Bandpass filtering
- Automatic gain control, AGC (to compensate for attenuation)
- Frequency shifting of the RF signal baseband I and Q signals, or to an intermediate frequency (IF) signal, or
- Sampling and analog-to-digital conversion (ADC), sometimes before the above point;
- Equalization filtering
- Detection of the amplitudes of the I and Q signals, or the frequency or phase of the IF signal;
- Quantization of the amplitudes, frequencies or phases to the nearest allowed values, using mapping.
- Map the quantized amplitudes, frequencies or phases to code-words (bit groups);
- Parallel-to-serial conversion of the code-words into a bit stream
- Pass the resultant bit stream on for further processing such as removal of any error-correcting codes.

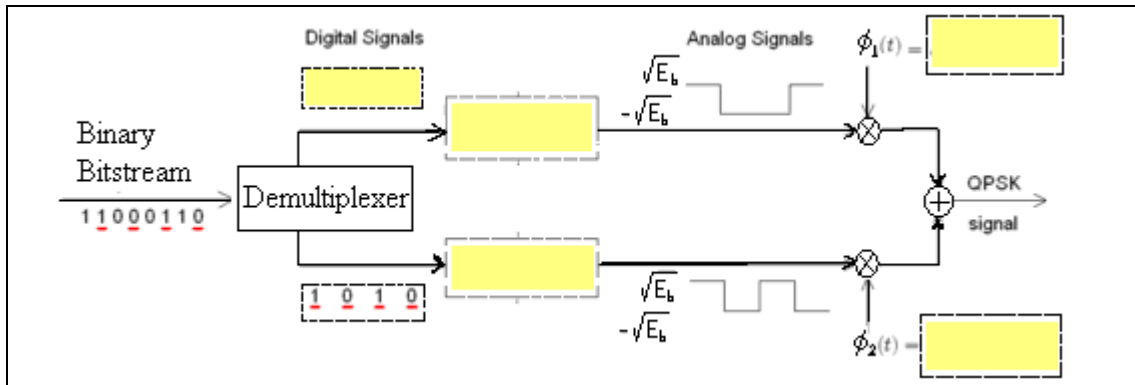
One of the basic concepts in data communication is the idea of allowing several transmitters to send information simultaneously over a single communication channel. This is called multiple access methods. On Code-Division Multiple Access (**CDMA**), each user is assigned a distinct code sequence (spreading code) that is used to encode the user information signal. The receiver retrieves the desired signal by using the same code sequence at the reception. Unlike FDMA and TDMA, CDMA's wide spreading signal makes it difficult to detect and jam. In addition to military applications, the most widely known application of CDMA is for cellular systems.



5-14. Problems

1) Prove that the OOK requires 3 dB more power than the BPSK in a coherent system

5-2) Complete the following block diagram of the QPSK transmitter



5-3) Consider a simple PLL, which consists of a VCO, loop filter $F(s)$ and a multiplier phase detector PFD. Let the applied signal to the multiplier to be a PSK signal in the following form:

$$x(t) = A_c \cos(2\pi f_s t + k_p m(t)),$$

where k_p is the phase sensitivity and $m(t)$ takes the values $+1V$ or $-1V$ for binary symbols 1 and 0. The VCO output is

$$y(t) = A_c \sin(2\pi f_s t + \theta(t)),$$

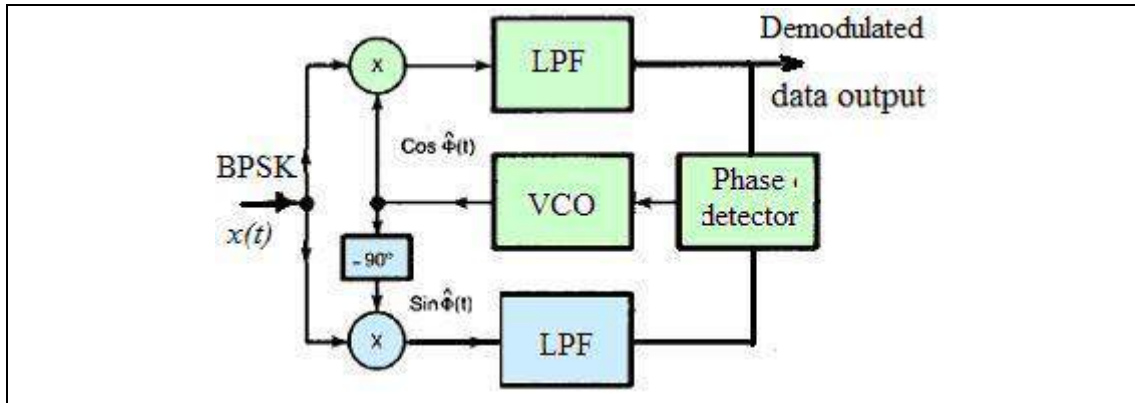
i) Calculate the loop filter output, assuming the filter eliminates only the double frequency $2f_s$.

ii) show that the loop output is proportional to the data signal $m(t)$, when the PLL is locked, i.e., when $\theta(t) = 0$.

5-4) Sketch the in-phase and quadrature components of an MSK signals

5-5) What's the difference between carrier synchronization and clock synchronization in coherent detection?

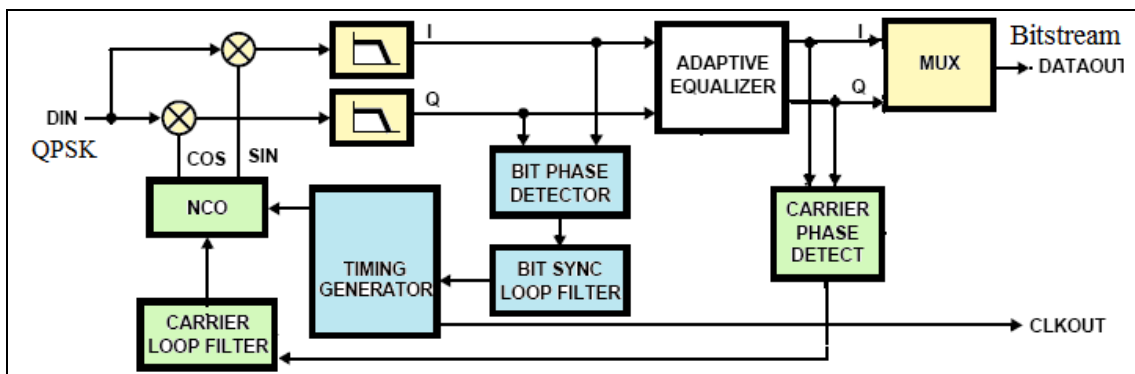
5-6) The following block diagram, depicts a simple Costas loop, which may be used for BPSK demodulators. Describe the different stages and the signals flowing between them. What's the difference between such simple circuit and the one presented in the main text?



5-8) Check the Right phrases with (\checkmark) and the False ones with (\times)

[1]	BPSK is the same as OOK	[]
[2]	QPSK is a sort of PSK	[]
[3]	QPSK has 4 symbols, each symbol is represented by 3 bits	[]
[4]	PCM is a sort of digital modulation	[]
[5]	QAM is a combination of ASK and PSK	[]
[6]	The guard time T_G is usually greater than the symbol time T_S	[]
[7]	The power needed for 16QAM is double the power needed for QPSK	[]
[8]	The error rate of DBPSK is equal to the BPSK	[]
[9]	The carrier phases in QPSK are $\phi = \pi/4, 3\pi/4, 5\pi/4, 7\pi/4$	[]
[10]	The power in QAM is proportional to 2^m , m is the no. of bits/symbol	[]

5-10) The following circuit depicts a practical QPSK receiver, (HSP50306) from Intersil, with carrier and timing synchronization. The part recovers 2.048 MB/s data from samples of a QPSK modulated 10.7MHz.



- i) Determine the carrier recovery and timing recovery loops
- ii) What's the function of the adaptive equalizer in this circuit? What's its relation to multipath distortion?
- iii) Determine the output bitrate of the demodulator

5-15. References

- [1] Subbarayan **Pasupathy**, *Minimal Shift Keying: A Spectrally Efficient Modulation*, IEEE Communications Magazine, **1979**.
- [2] **K. Murota** and **K. Hirade**, "GMSK Modulation for Digital Mobile Radio Telephony," IEEE Transactions on Communications, Vol COM-29, No. 7. pp. 1045-1050, July **1981**.
- [3] **Simon Haykin**, *Digital Communications*. Toronto, Canada: John Wiley & Sons, **1988**.
- [4] Cellular Digital Packet Data System Specification, Release 1.0, **1993**.
- [5] **John G. Proakis**, *Digital Communications*, McGraw Hill, Singapore, **1995**.
- [6] **J. Viterbi**, *CDMA: Principles of Spread Spectrum Communication*, 1st Edition, Prentice Hall, **1995**.
- [7] **J. Kim**, and **G. J. Pottie**, "On Punctured Trellis Coded Modulation", IEEE Trans. on Inf. Theory, March **1996**.
- [8] **Leon W. Couch**, *Digital and Analog Communications*. Upper Saddle River, NJ: Prentice-Hall, **1997**.

Chapter
6

Noise Estimation & Bit Error Rate

Contents

- 6-1. Introduction**
- 6-2. SNR in AM & FM Systems**
- 6-3. Errors in Digital Modulation Systems**
 - 6-3.1. Modulation Error Magnitude (MER)
 - 6-3.2. Error Vector Magnitude (EVM)
- 6-4. Bit Error Rate (BER)**
- 6-5. Measurement of the BER in Digital Systems**
- 6-6. Computing the BER in Digital Modulation Systems**
- 6-7. BER of BPSK**
- 6-8. BER of QPSK**
- 6-9. BER of Higher-order PSK**
 - 6-10.1. Symbol-Error Rate in M-PSK
 - 6-10.2. Bit-Error Rate in M-PSK
- 6-10. BER of Differential Phase-Shift Keying (DPSK)**
- 6-11. BER of QAM**
 - Case 1: Rectangular QAM with Even k
 - Case 2: Rectangular QAM with Odd k
 - Case 3: Non-Rectangular QAM
- 6-12. Bandwidth Efficiency**
- 6-13. Summary**
- 6-14. Problems**
- 6-15. Bibliography**

Chapter

6

Noise Estimation & Bit Error Rate

6-1. Introduction

In statistics, the term *error* arises in different ways. For instance, it arises in the context of statistical modeling where the model predicted value may be in error regarding the observed reality. In communication systems, we refer to the effect of noise and other channel impairments, which hinder the reproduction of the original transmitted signal, in the receiver side. In fact, noise is an ever present part of all systems. Any receiver must contend with noise. In analog systems, noise deteriorates the quality of the received signal, e.g. the appearance of *snow* on the TV screen, or *static hum* sounds during an audio transmission. In digital communication systems, noise degrades the signal quality. In this chapter we examine the noise levels resulting in analog modulation as well as the bit error rate (**BER**) of different digital *modulation* systems.

6-2. SNR in AM and FM Systems

We usually determine and compare the performance of analogue modulation systems on the basis of signal-to-noise ratio (*SNR*) at the receiver input and output. Assuming a sinusoidal input AM signal, it can be proved that the output signal-to-noise ratio of an envelope detector is:

$$\frac{S_o}{N_o} = \frac{m^2}{2+m^2} \frac{S_i}{N_i} \quad (6-1)$$

where m is the modulation index. For large *input SNR* (S_i/N_i) and a fixed modulation index (m), the output *SNR* is directly proportional to the input *SNR*. When the (S_i/N_i) $\ll 1$, the envelope signal is primarily dominated by the envelope of the noise signal and the modulating signal is badly recovered. Under this circumstance, it is meaningless to talk about output *SNR*. The loss of the modulating signal at low input *SNR* is called the **threshold effect**. The threshold occurs when the input *SNR* < 10 dB. As for FM system, it is difficult to analyze their noise immunity performance in the general case. So, we consider here a sinusoidal modulating signal, with

single tone $m(t) = A_m \cos \omega_m t$. One can prove that the output signal-to-noise ratio of an FM demodulator is given by the following relation:

$$\frac{S_o}{N_o} = \frac{3}{(k_f)^2} \beta_f^2 \left(\frac{S_i}{N_i} \right) \quad (6-2)$$

Here f_m is the bandwidth of the modulating signal and the frequency modulation index $\beta_f = k_f A_m / (2\pi f_m)$. For instance, the demodulator of analog mobile phone systems (AMPS) was characterized by a carrier-to-noise ratio of 18 dB, resulting in an output SNR of 40dB. Figure 6-1 compares the output (demodulated) signal quality of AM, FM and PCM systems in terms of their input SNR.

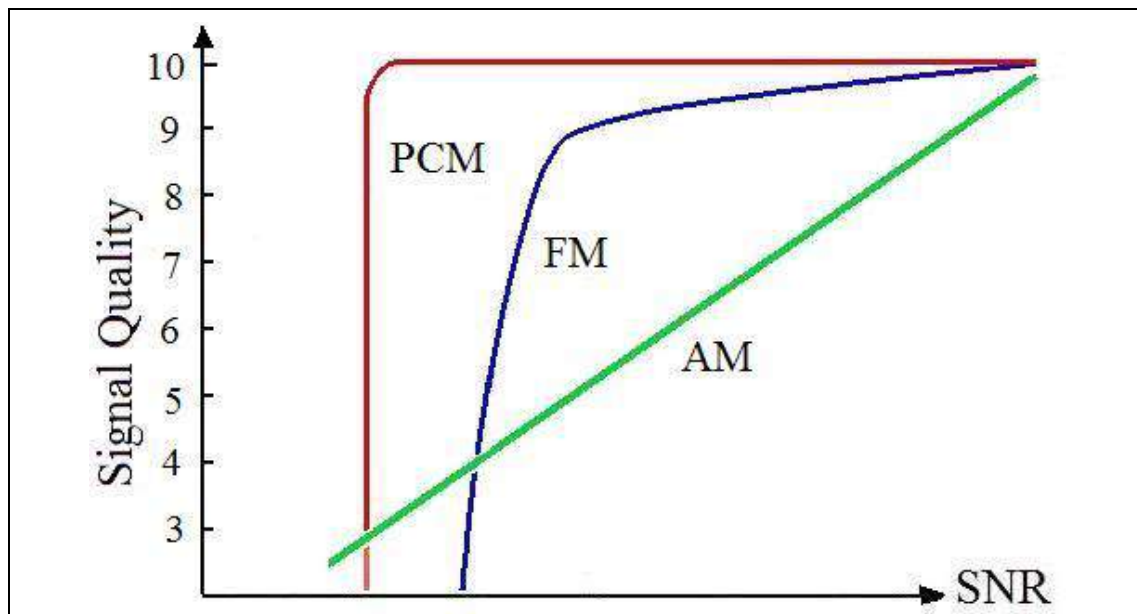


Fig. 6-1. Signal-to-noise ratio (SNR) of different modulation techniques.

6-3. Errors in Digital Modulation Systems

In digital modulation systems, we are able to determine the system performance quite uniquely by calculating the probability of error. For determining the error-rates in digital modulation systems, it is convenient to present the following terms and definitions:

- E_b = Energy-per-bit,
- E_s = Energy-per-symbol = $m E_b$ with m bits per symbol,
- N_0 = Noise power spectral density (W/Hz),
- P_b = Probability of bit-error,
- P_s = Probability of symbol-error,
- P_{bc} = Probability of bit-error per carrier,

- R_s = Symbol rate, R_b = Bit rate, $R_b = m R_s$
- T_b = Bit duration, T_s = Symbol duration.

6-3.1 Modulation Error Ratio (MER)

The modulation error ratio (MER) is a measure of the signal-to-noise ratio in a digital modulated signal. MER is usually measured, by specialized equipment, to quantify the performance of a digital radio transmitter or receiver in a communications system. A signal sent by an ideal transmitter or received by a receiver would have all constellation points precisely at the ideal locations; however various imperfections in the implementation (such as noise, low image rejection ratio, phase noise, distortion, etc.) or signal path cause the actual constellation points to deviate from the ideal locations. If the number of symbols in a digital signal is N , MER is defined as follows:

$$MER = \frac{\sum_{j=1}^N (\tilde{I}_j^2 + \tilde{Q}_j^2)}{\sum_{j=1}^N [(I_j - \tilde{I}_j)^2 + (Q_j - \tilde{Q}_j)^2]} \quad (6-3)$$

where I_j is the I component of the received j^{th} symbol, Q_j is the Q component of the j^{th} received symbol, \tilde{I}_j is the ideal I component of the j^{th} received symbol and \tilde{Q}_j is the ideal Q component of the j^{th} received symbol. Like SNR, MER is usually expressed in decibels (dB).

6-3.2. Error Vector Magnitude (EVM)

The error vector magnitude (EVM) is a measurement of the demodulator performance in the presence of noise and impairments. The measured symbol location obtained after decimating the recovered waveform at the demodulator output are compared against the ideal symbol locations. The root-mean-square (RMS) value of EVM and phase error are then used in determining the EVM over a window of N demodulated symbols.

As shown in figure 6-2, the measured symbol location by the demodulator is given by \underline{w} . However, the ideal symbol location (using the symbol map) is given by \underline{v} . Therefore, the resulting error vector $\underline{e} = \underline{w} - \underline{v}$ is the difference between the actual measured and ideal symbol vectors. In the following figure, we have: \underline{v} is the ideal symbol vector, \underline{w} is the measured symbol vector, θ is the phase error and e/v is the EVM. These parameters do not necessarily reveal the nature of the impairment. In order to remove the dependence on system gain, EVM is normalized by $|\underline{v}|$, which is expressed as a percentage.

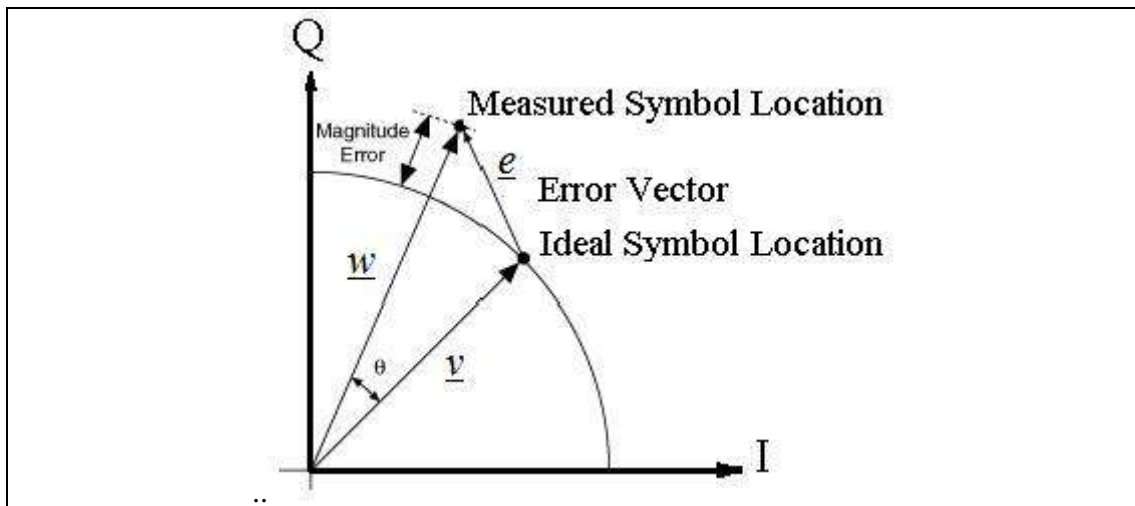


Fig. 6-2. Representation of error vector magnitude (EVM).

Analytically, the RMS EVM over a measurement window of N symbols is defined as follows:

$$EVM = \frac{\sqrt{\frac{1}{N} \sum_{j=1}^N [(I_j - \tilde{I}_j)^2 + (Q_j - \tilde{Q}_j)^2]}}{|v_{max}|} \quad (6-4)$$

where I_j is the I component of the j^{th} symbol received, Q_j is the Q component of the j^{th} symbol received, \tilde{I}_j is the ideal I component of the j^{th} symbol received, \tilde{Q}_j is the ideal Q component of the j^{th} symbol received.

MER and EVM measure essentially the same quantity and easy conversion is possible between the two measures if the constellation is known. When expressed as simple voltage ratios MER_V is equal to the reciprocal of the product of EVM_V and the peak-to-mean voltage ratio for the constellation. Thus, EVM and MER are proportional.

6-4. Bit Error Rate (BER) in Digital Communication Systems

In digital communication systems, the bit error ratio (**BER**) is the ratio of the number of received error bits, to the total number of transmitted bits during a specified time interval. On good connections the BER should be below 10^{-10} . In a noisy channel, the BER is often expressed as a function of the normalized signal-to-noise ratio measure denoted E_b/N_0 (energy per bit to noise power spectral density ratio), The BER can be also plotted versus Carrier-to-noise (**C/N**) ratio. In fact, the C/N ratio measurements can be converted to E_b/N_0 as follows:

$$\frac{E_b}{N_0} = \frac{C}{N} + 10 \log_{10} \frac{BW_{noise}}{f_s \times m} \quad [\text{in dB}] \quad (6-5)$$

where m is the number of bits per symbol ($m = 6$ for 64-QAM) and N is measured in the Nyquist bandwidth (symbol rate). Note that E_b/N_0 is actually equal to the SNR divided by the link spectral efficiency in (bit/s)/Hz, where the bits are transmitted data bits, including error correction coding information. In fact, we have to distinguish between the *coded* and *uncoded* signals in the following aspects:

1- **Coded signal.** A coded signal is a signal that has coding applied to it at the source level so that only some of the bits in the stream are important. We can tolerate a higher gross error rate in this channel as long as we can decode the information bits with much better integrity.

2- **Uncoded signal.** An uncoded signal has no coding so all bits are equally important and the started BER is the BER of the information bits.

Therefore, we have two types of bit error ratios:

(a) transmission BER, *i.e.*, the number of erroneous bits received divided by the total number of bits transmitted; and

(b) information BER, *i.e.*, the number of erroneous decoded (corrected) bits divided by the total number of decoded (corrected) bits.

The BER of the information bits is much less than that of the coded signal.

$$BER (\text{info bits}) \ll BER (\text{coded bits}) \quad (6-6a)$$

If E_b/N_o is defined as the bit energy of signal to noise ratio, then E_b/N_o of the information bit is approximately given by:

$$E_b/N_o (\text{info}) = E_b/N_o (\text{coded}) - \text{Code rate (in dB's)} \quad (6-6)$$

6-5. Measurement of the BER in Digital Modulation Systems

Engineers usually plot the BER curves to describe the functionality of a digital communication system. Measuring the BER helps designers to choose the appropriate error correction codes.

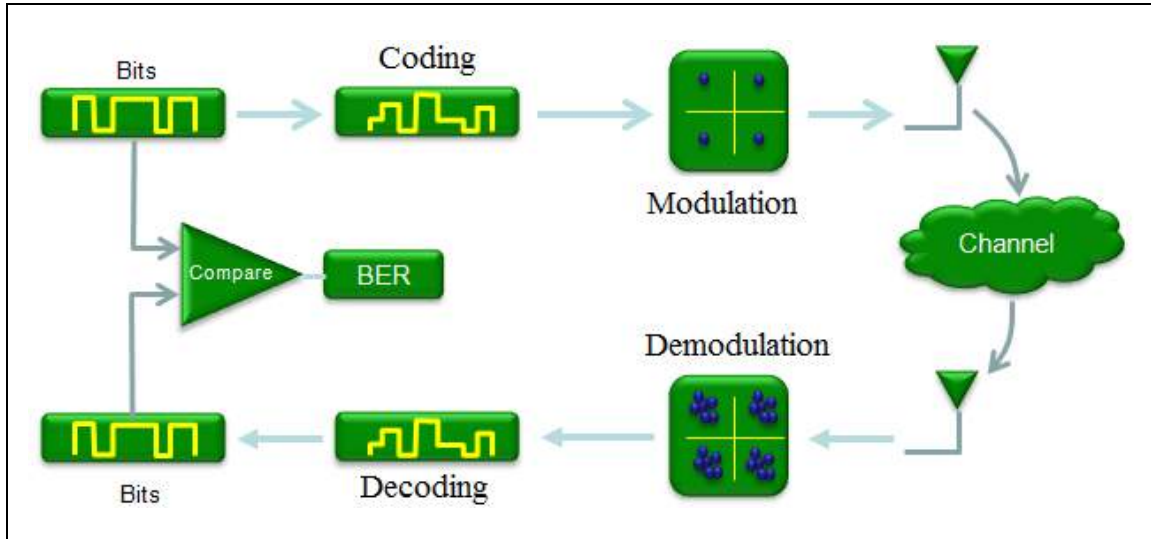


Fig. 6-3(a). Bit error rate (BER) calculation for a specific modulation technique

The BER versus E_b/N_o curve is measured using the RF and noise power measurement techniques. The following figure depicts the test setup for BER measurement. Signal power can be measured directly with RF voltmeter. The spectral density of the noise (per Hz) can be also measured with a spectrum analyzer.

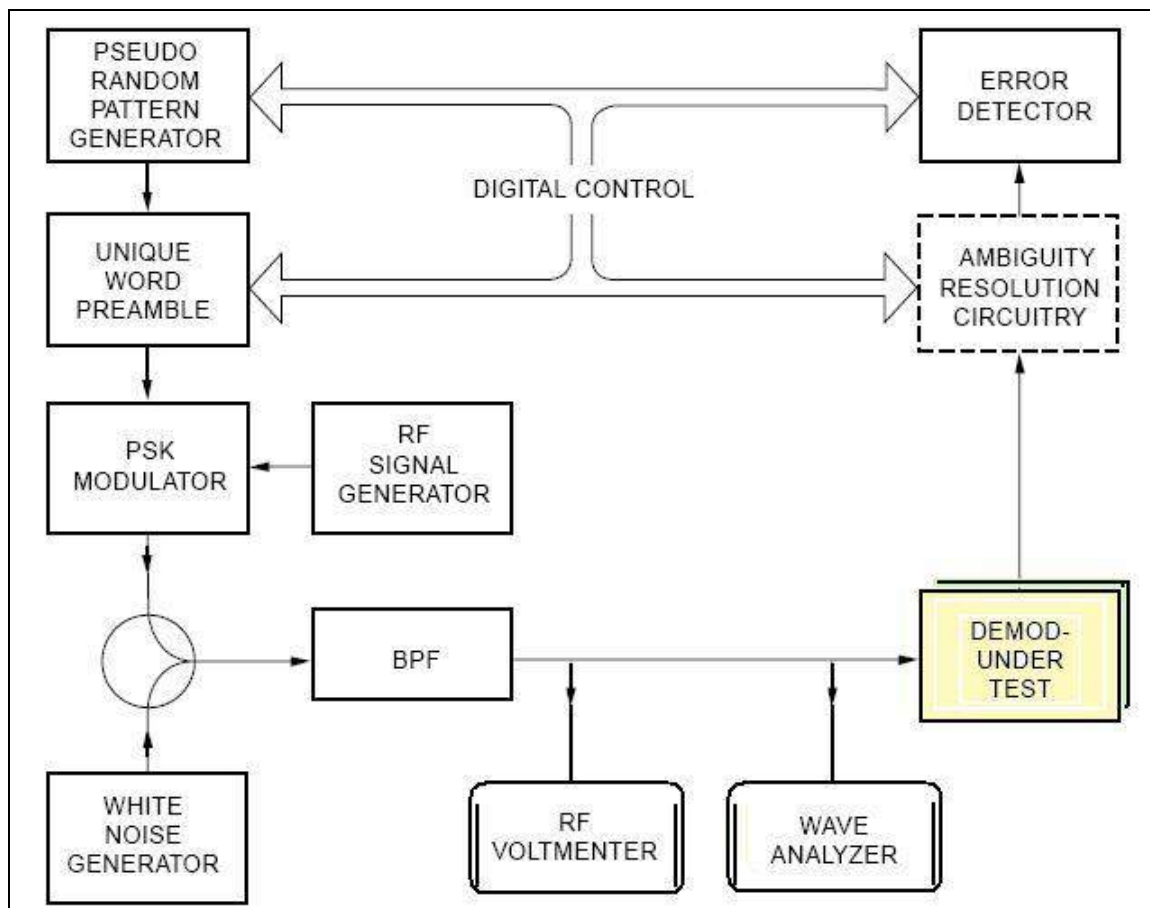


Fig. 6-3(b). Bit error rate (BER) measurement setup

6-6. Computing the BER in Digital Modulation Systems

The *BER*, is approximately equal to the probability of error, occurring to a stream of transmitted bits, over a noisy channel. The transmitted bits are corrupted by noise, typically referred to as Additive White Gaussian Noise (**AWGN**). **Additive** means that the noise gets added (and not multiplied) to the received signal. **White** means that the spectrum of the noise is flat for all frequencies. **Gaussian** means that the values of the noise follow the Gaussian probability distribution function. In order to quantify the error probability and BER, let's define $Q(X)$ as the probability that a single sample, taken from a random process, with zero-mean and unit-variance Gaussian probability density function (**PDF**), will be greater or equal to certain value X .

$$Q(x \geq X) = \frac{1}{\sqrt{2\pi}} \int_X^{\infty} e^{-\frac{1}{2}t^2} dt, \quad X \geq 0 \quad (6-7)$$

Note that $Q(X)$ is related to the complementary error function by:

$$Q(X) = \frac{1}{2} \operatorname{erfc} \left(\frac{X}{\sqrt{2}} \right), \quad X \geq 0 \quad (6-8)$$

which is the probability that X will be under the tail of the Gaussian PDF towards positive infinity.

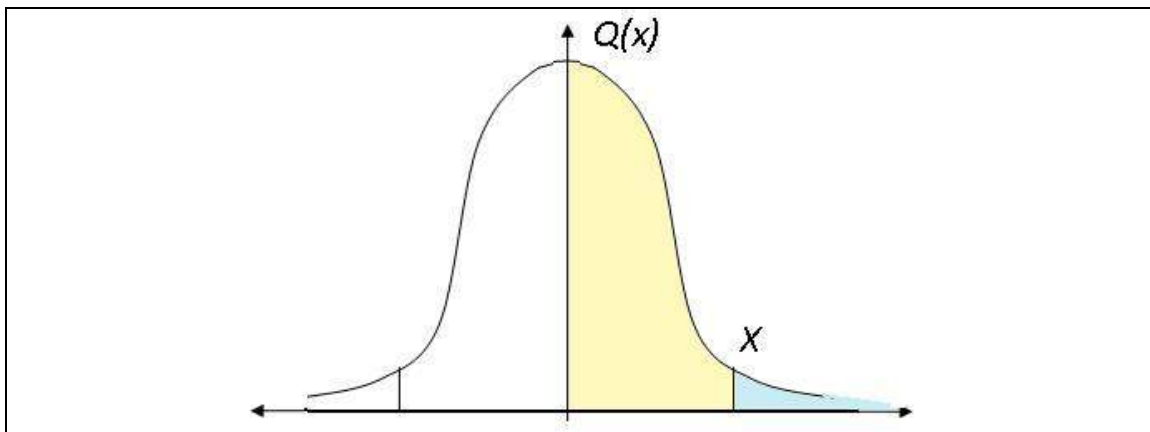


Fig. 6-4. Probability of error as a function of signal strength

Note 6-1: Error Function Complementary (ERFC)

The complementary error function is defined as:

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-x^2} dx$$

For large values of x , the $\operatorname{erfc}(x)$ may be approximated as follows:

$$\operatorname{erfc}(x) \approx (1/\sqrt{\pi x}) \cdot \exp(-x^2) \quad x \gg 0$$

The error-rates calculated using this distribution are those in additive white Gaussian noise (AWGN). Such error rates are lower than those computed in **fading channels**, which are good benchmark to compare with

6-7. BER of Binary Phase-Shift Keying (BPSK)

As we have mentioned so far, the BPSK modulation is the simplest form of PSK but also the most robust PSK. It uses two phases which are separated by 180° . It does not particularly matter where exactly the constellation points are positioned. It is only able to modulate at 1 bit/symbol and so is unsuitable for high data-rate applications. The bit error rate (BER) of BPSK in AWGN can be calculated as follows:

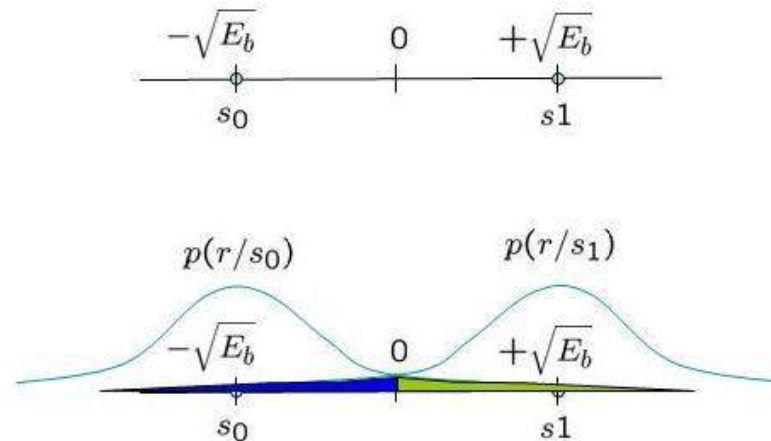
$$P_b(\text{BPSK}) = \frac{1}{2} Q[\sqrt{2E_b/N_o}] = \frac{1}{2} \operatorname{erfc}[\sqrt{(E_b/N_o)}] \quad (6-9)$$

Since there is only one bit per symbol, this is also the symbol error rate.

Note 6-2. Proof of the Error Probability Formula

Assumed that the received signal, y , is given by: $y = s_1 + n$ when bit 1 is transmitted and $y = s_0 + n$ when bit 0 is transmitted. The **conditional probability** distribution function (PDF) of y for the two cases are:

$$p(y|s_0) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y+\sqrt{E_b})^2}{N_0}} \quad \text{and} \quad p(y|s_1) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y-\sqrt{E_b})^2}{N_0}}$$



Assuming that s_0 and s_1 are equally probable i.e. $p(s_1) = p(s_0) = 1/2$, the **threshold 0** forms the optimal decision boundary. Therefore,

- if the received signal is y is greater than 0, then the receiver assumes s_1 was transmitted.
- if the received signal is y is less than or equal to 0, then the receiver assumes s_0 was transmitted. Therefore, $y > 0 \Rightarrow s_1$ and $y \leq 0 \Rightarrow s_0$.

Probability of error given s_1 was transmitted

With this threshold, the probability of error given s_1 is transmitted is (the area in blue region):

$$p(e|s_1) = \frac{1}{\sqrt{\pi N_0}} \int_{-\infty}^0 e^{-\frac{(y-\sqrt{E_b})^2}{N_0}} dy = \frac{1}{\sqrt{\pi}} \int_{\frac{\sqrt{E_b}}{N_0}}^{\infty} e^{-z^2} dz = \frac{1}{2} \text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$

where *erfc* is the complementary error function.

Probability of error given s_0 was transmitted

Similarly the probability of error given s_0 is transmitted is (the area in green region):

$$p(e|s_0) = \frac{1}{\sqrt{\pi N_0}} \int_0^{\infty} e^{-\frac{(y+\sqrt{E_b})^2}{N_0}} dy = \frac{1}{\sqrt{\pi}} \int_{\frac{\sqrt{E_b}}{N_0}}^{\infty} e^{-z^2} dz = \frac{1}{2} \text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$

Total probability of bit error

$$P_b = p(s_1)p(e|s_1) + p(s_0)p(e|s_0).$$

Given that we assumed that s_0 and s_1 are equally probable i.e. $p(s_1) = p(s_0) = 1/2$, the **bit error probability** is,

$$P_b = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right).$$

6-8. BER of Quadrature Phase-Shift Keying (QPSK)

In constellation diagram of QPSK with Gray coding, we have each adjacent symbol only differs by one bit. Although QPSK can be viewed as a quaternary modulation, it is easier to see it as two independently modulated quadrature carriers. With this interpretation, the even (or odd) bits are used to modulate the in-phase component of the carrier, while the odd (or even) bits are used to modulate the quadrature-phase component of the carrier. BPSK is used on both carriers and they can be independently demodulated. As a result, the probability of bit-error P_b for **QPSK is the same as for BPSK**:

$$P_b (QPSK) = \frac{1}{2} \operatorname{erfc} [\sqrt{(E_b/N_o)}] \quad (6-10)$$

However, with two bits per symbol, the symbol error rate in QPSK is increased:

$$P_s = 1 - \text{Probability of correct symbol} \quad (6-11)$$

$$\text{Probability of correct symbol (2-bits)} = (1-P_b).(1-P_b) \quad (6-12)$$

Therefore:

$$\begin{aligned} P_s (QPSK) &= 1 - (1 - P_b)^2 \\ &= 2Q \left(\sqrt{\frac{E_s}{N_0}} \right) - Q^2 \left(\sqrt{\frac{E_s}{N_0}} \right) \end{aligned} \quad (6-13)$$

If the signal-to-noise ratio is high, which is necessary for practical QPSK systems, the probability of symbol error may be approximated as follows:

$$P_s (QPSK) \approx 2 Q[\sqrt{(E_b/N_o)}] = \operatorname{erfc}[\sqrt{(E_b/N_o)}] \quad (6-14)$$

As with BPSK, there are phase ambiguity problems in QPSK systems at the receiver, This phase ambiguity may be eliminated by adopting the differentially encoded QPSK (DQPSK) technique in practice.

6-9. Error Probability of Higher-order PSK

Any number of phases may be used to construct a PSK constellation but 8-PSK is usually the highest order PSK constellation deployed. With more than 8 phases, the error-rate becomes too high and it is better to use other complex modulation methods, such as quadrature amplitude modulation (QAM). Although any number of phases may be used, the fact that the constellation must usually deal with binary data means that the number of symbols is usually a power of 2 — this allows an equal number of bits-per-symbol.

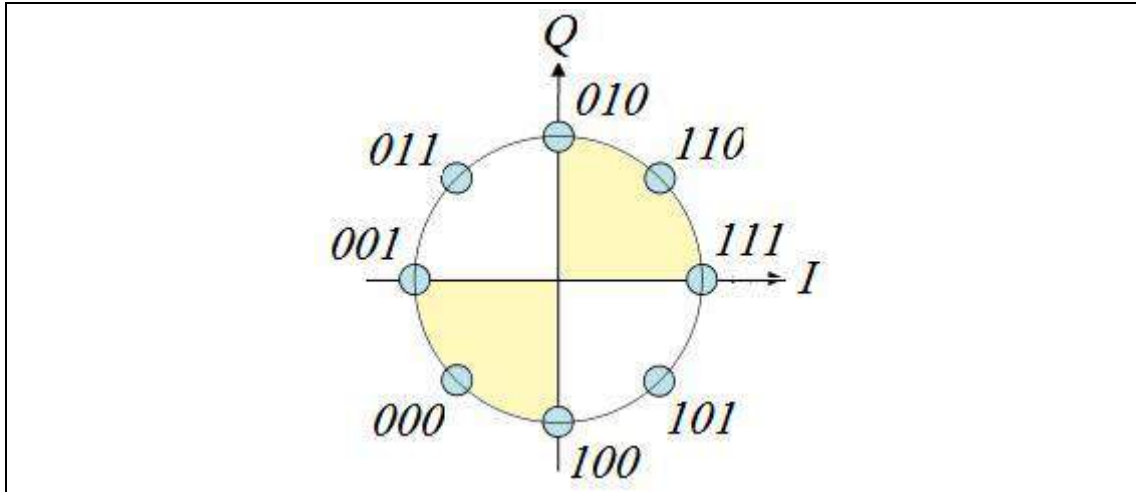


Fig. 6-5. Possible constellation diagram of an 8-PSK with Gray coding

6-9.1. Symbol Error Rate in M-PSK

For the general M -PSK there is no simple expression for the **symbol-error probability** if $M > 4$. Unfortunately, it can only be obtained from the following relation:

$$P_s = 1 - \int_{-\alpha}^{\alpha} p(\Theta_r) d\Theta_r \quad (6-15)$$

where $\alpha = \pi/M$,

$$\Theta_r = \tan^{-1}(r_2/r_1), \quad (6-16a)$$

$$V = \sqrt{(r_1^2 + r_2^2)}, \quad (6-16b)$$

and

$$\gamma_s = E_s/N_o \quad (6-16c)$$

Here, r_1 and r_2 are jointly-Gaussian random variables such that:

$$r_1 \sim N(\sqrt{E_s}, 1/2N_o), \quad r_2 \sim N(0, 1/2N_o) \quad (6-17)$$

This may be approximated for high M and high values of (E_b/N_o) by:

$$P_s (M\text{-PSK}) \approx \text{erfc}[\sqrt{(E_b/N_o)} \cdot \sin(\pi/M)] \quad (6-18)$$

6-9.2. Bit-Error Rate in M-PSK

The bit-error probability, P_b , for M -PSK can only be determined exactly once the bit-mapping is known. However, when Gray coding is used, the most probable error from one symbol to the next produces only a single bit-error and

$$P_b \text{ (M-PSK, with Gray Coding)} \approx P_s \text{ (M-PSK)} / k \quad (6-19)$$

Figure 6-6 compares the bit-error rates of BPSK, QPSK (which are the same, as noted above), 8-PSK and 16-PSK. As shown in figure, the higher-order modulations exhibit higher error-rates; in exchange however they deliver a higher raw data-rate.

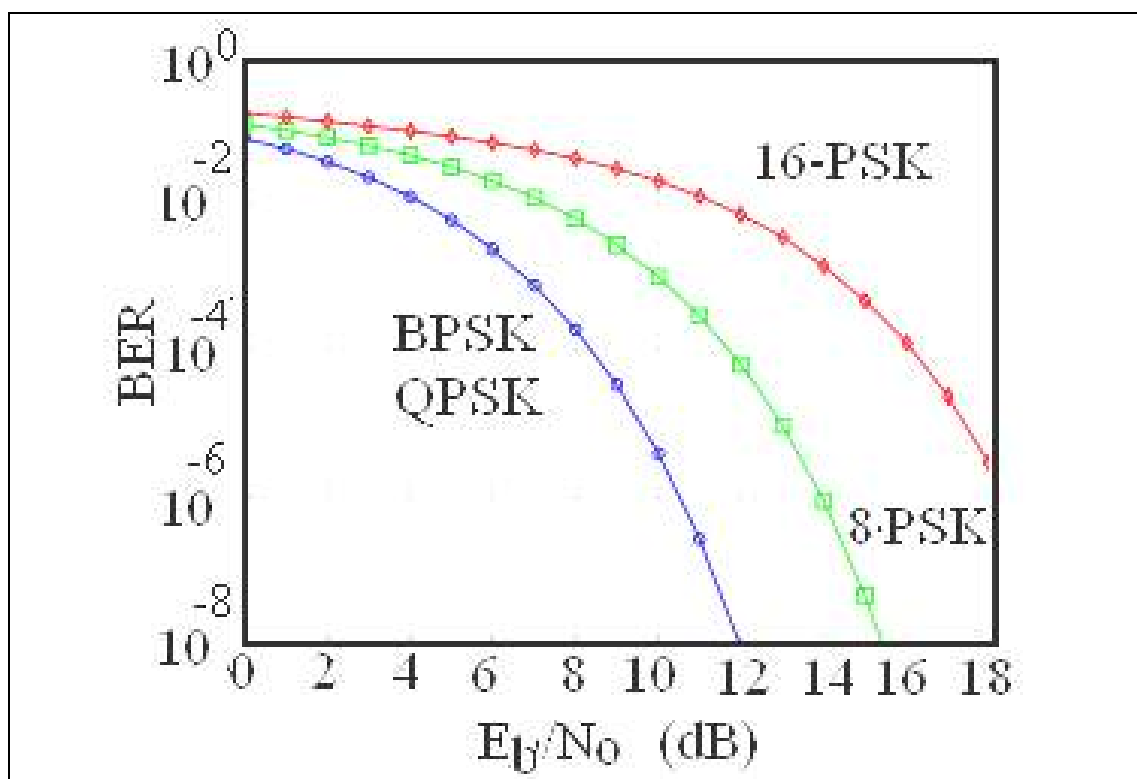


Fig. 6-6. Bit-error rate curves for BPSK, QPSK, 8-PSK and 16-PSK

6-10. BER of Differential Phase-Shift Keying (DPSK)

For a signal that has been differentially encoded, there is an obvious alternative method of demodulation. Ignoring carrier-phase ambiguity, the phase between two successive received symbols is compared and used to determine what the data must have been. When differential encoding is used in this manner, the scheme is known as differential phase-shift keying (DPSK). Note that this is subtly different to just differentially-encoded PSK since, upon reception; the received symbols are *not* decoded one-by-one to constellation points but are instead compared directly to one another. Assume the received symbol in the k^{th} timeslot r_k and let it have phase ϕ_k . Assume also without loss of generality that the phase of the carrier wave is zero. Denoting the noise (AWGN) term as n_k , we can write:

$$r_k = \sqrt{E_s} e^{j\phi_k} + n_k \quad (6-20)$$

The decision variable for the $(k-1)^{\text{th}}$ symbol and the k^{th} symbol is the phase difference between r_k and r_{k-1} . That is, if r_k is projected onto r_{k-1} , the decision is taken on the phase of the following resultant complex number:

$$r_k r_{k-1}^* = E_s e^{j(\theta_k - \theta_{k-1})} + \sqrt{E_s} e^{j\theta_k} n_{k-1}^* + \sqrt{E_s} e^{-j\theta_{k-1}} n_k + n_k n_{k-1} \quad (6-21)$$

where the superscript * denotes complex conjugation. In the absence of noise, the phase of this is $\theta_k - \theta_{k-1}$, the phase-shift between the two received signals which can be used to determine the data transmitted. Obviously, the probability of error for DPSK is difficult to calculate in general, but, in the case of DBPSK it is given by:

$$P_b (\text{DBPSK}) = 1/2 \exp(-E_b/N_o) \quad (6-22)$$

This probability, when numerically evaluated, is only slightly **worse** than ordinary BPSK, particularly at higher E_b / N_0 values. However, using DPSK avoids the need for possibly complex carrier-recovery schemes to provide an accurate phase estimate and can be an attractive alternative to ordinary PSK. The bit-error rates of DBPSK and DQPSK are compared to their non-differential counterparts in figure 6-7. The loss for using DBPSK is small enough compared to the complexity reduction that it is often used in communications systems that would otherwise use BPSK. For DQPSK though, the loss in performance compared to ordinary QPSK is larger and the system designer must balance this against the reduction in complexity.

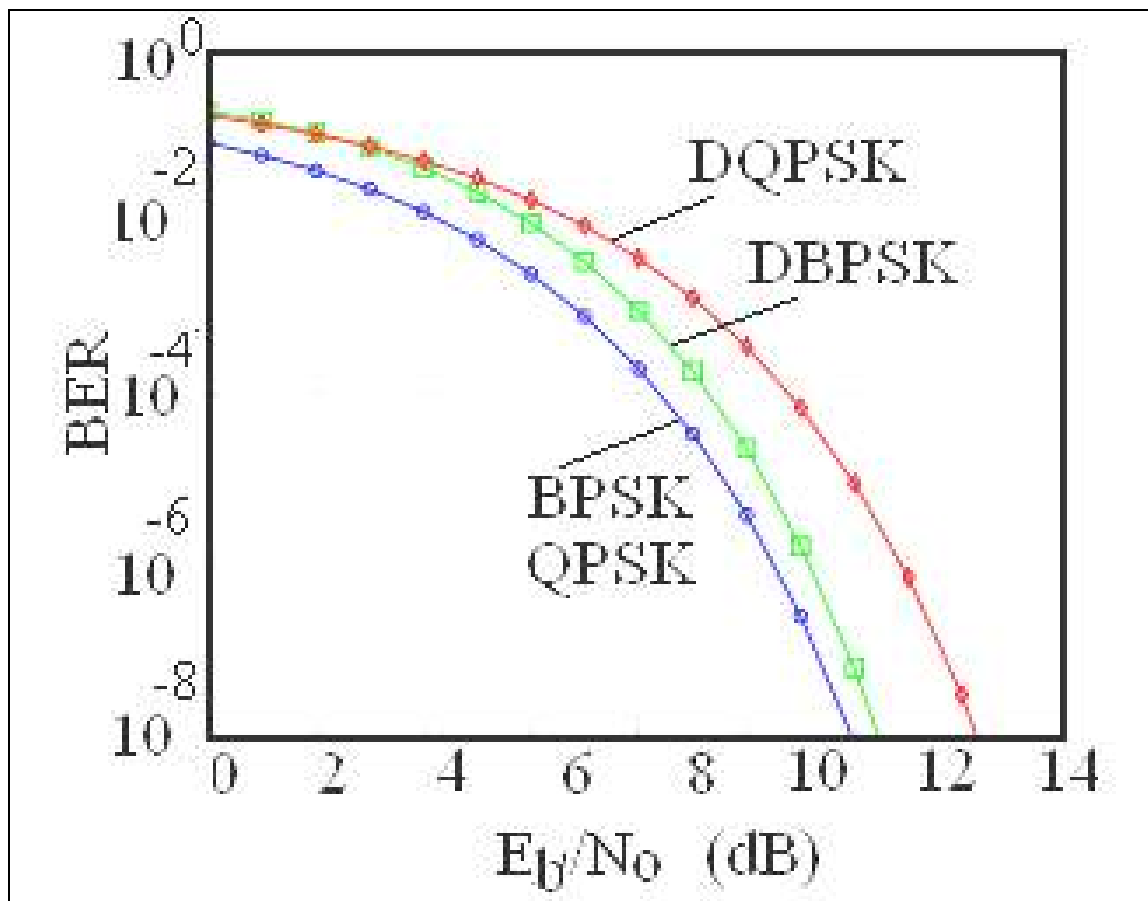


Fig. 6-7. BER comparison between DBPSK, DQPSK and their non-differential forms using gray-coding and operating in white noise

6-11. BER of QAM Systems

In this section we calculate the BER rate for QAM systems, with M symbols (M-QAM). We concentrate our attention on rectangular constellation QAM, particularly the case of rectangular 16-QAM. In the latter case, we have 16 symbols to transmit, with each symbol standing for 4 bits. We also assume a Gray coded bit-assignment. The reason that 16-QAM is usually the first rank of M-QAM, is that 2-QAM is equivalent to BPSK and 4-QAM is equivalent to QPSK. Also, the error-rate performance of 8-QAM is close to that of 16-QAM (about 0.5dB better), but its data rate is only three-quarters that of 16-QAM.

When the received signal is perturbed by Additive White Gaussian Noise (AWGN) there is a probability that any particular symbol will be wrongly decoded into one of the adjacent symbols. Expressions for the symbol error-rate of rectangular QAM are not hard to derive but yield rather unpleasant expressions.

Case 1: Rectangular QAM with Even k (Bits per Symbol)

For an even number of bits per symbol, k , exact expressions of rectangular QAM error rate are available. They are most easily expressed *per carrier* sense:

$$P_{sc} = 2 \left(1 - \frac{1}{\sqrt{M}} \right) Q \left(\sqrt{\frac{3}{M-1} \frac{E_s}{N_0}} \right) \quad (6-23)$$

Therefore, the symbol error rate is given by:

$$P_s = 1 - (1 - P_{sc})^2 \quad (6-24)$$

The bit-error rate will depend on the exact assignment of bits to symbols, but for a Gray-coded assignment with equal bits per carrier:

$$P_{bc} = \frac{4}{k} \left(1 - \frac{1}{\sqrt{M}} \right) Q \left(\sqrt{\frac{3k}{M-1} \frac{E_b}{N_0}} \right) \quad (6-25)$$

Case 2: Rectangular QAM with Odd k (Bits per Symbol)

Now we consider the case of odd k (like 8-QAM, with $k=3$ bits/symbol). In this case, it is hard to obtain symbol-error rates, but an upper bound is:

$$P_s \leq 4Q \left(\sqrt{\frac{3kE_b}{(M-1)N_0}} \right) \quad (6-26)$$

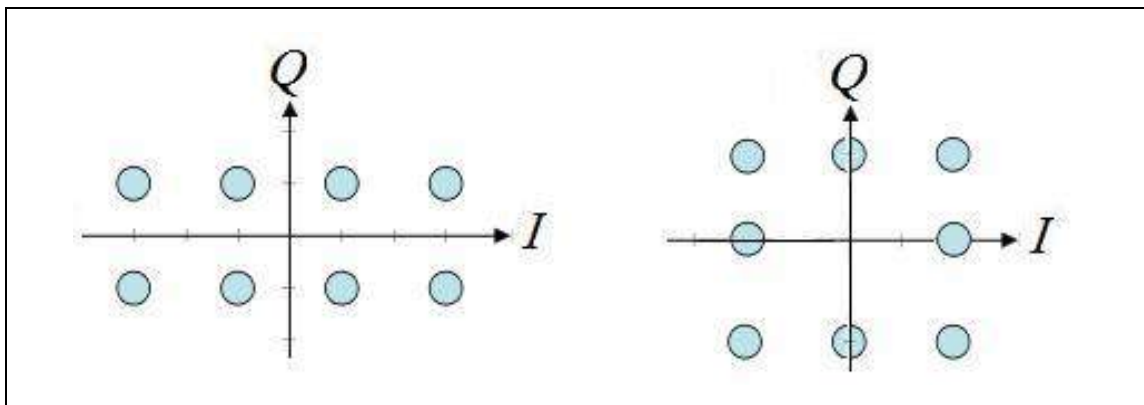


Fig. 6-8 Possible constellations of a rectangular 8-QAM

The above figure depicts two rectangular 8-QAM constellations without bit assignments. They have the same minimum distance between symbol points, and thus the same symbol-error rate (to a first approximation). The

exact bit-error rate, P_b will depend on the bit-assignment. Note that neither of these constellations are used in practice, as the rectangular version of 8-QAM is not optimal.

Case 3: Non-Rectangular QAM

It is the nature of QAM that most orders of constellations can be constructed in many different ways. Two diagrams of circular QAM constellation are shown, for 8-QAM and 16-QAM. The circular 8-QAM constellation is known to be the optimal 8-QAM constellation in the sense of requiring the least mean power for a given minimum Euclidean distance.

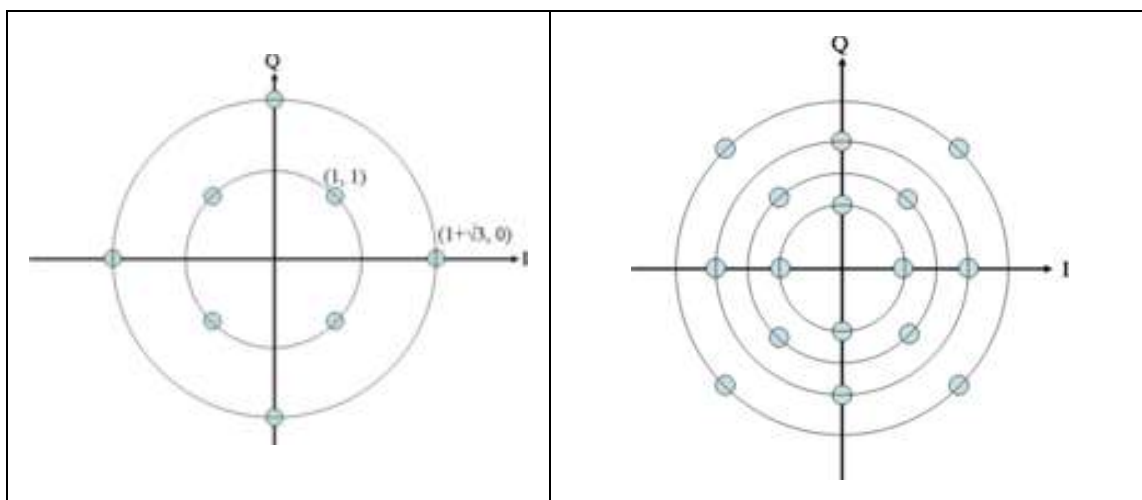


Fig. 6-9. Possible constellations of non-rectangular (Circular) 8-QAM and 16-QAM

The 16-QAM constellation is suboptimal although the optimal one may be constructed along the same lines as the 8-QAM constellation. The circular constellation highlights the relationship between QAM and PSK.

Other orders of constellation may be constructed along similar or different lines. It is consequently hard to establish expressions for the error-rates of non-rectangular QAM since it necessarily depends on the constellation. Nevertheless, an obvious upper bound to the rate is related to the minimum Euclidean distance of the constellation (the shortest straight-line distance between two points):

$$P_s < (M-1) \cdot Q \left[\sqrt{(d_{min}^2 / 2N_o)} \right] \quad (6-27)$$

Again, the bit-error rate will depend on the assignment of bits to symbols. Although, the optimal constellation for a particular M is non-rectangular in general, the non-rectangular are not often used since the rectangular

QAM's are much easier to modulate and demodulate.

The following figure depicts the bit error rates versus E_b/N_o for various modulation techniques. Note that in optical communication, BER (dB) versus Received Power (dBm) is usually used; while in wireless communication, BER (dB) versus SNR (dB) is used.

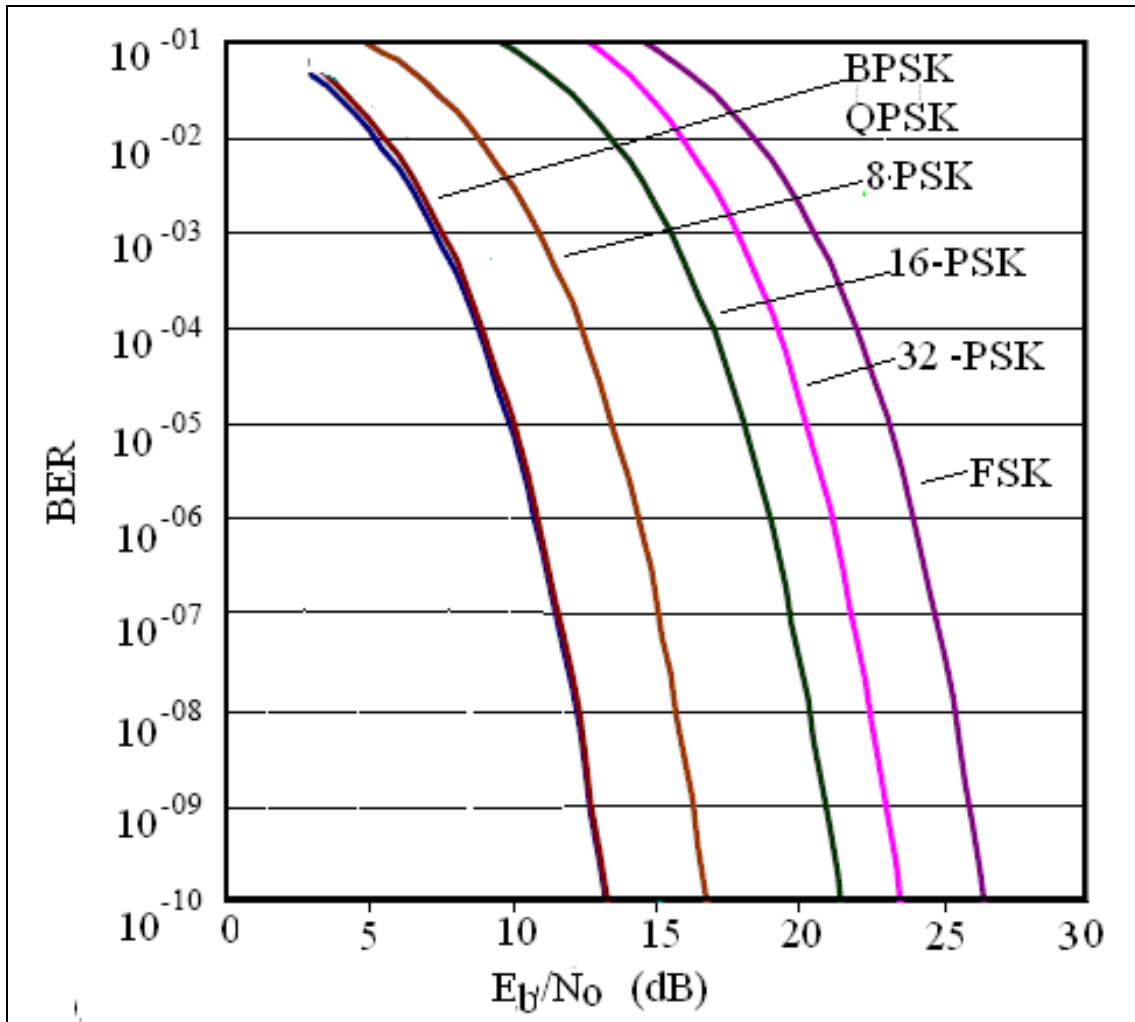


Fig. 6-10. Maximum bound of BER versus E_b/N_o for various modulation techniques

6-12. Comparison between Digital Modulation Methods

The error probability is not the only key issue when we compare between different digital modulations techniques. Another important key issue is the bit rate and the required bandwidth to transmit data. The bandwidth efficiency was already defined (in Chapter 1) as follows:

$$\rho = R_b / B \quad [\text{Bit/s/Hz}] \quad (6-28)$$

where R_b is the bit rate [Bit/s] and B is the bandwidth [Hz]. The following table summarizes the probability of error (P_e) and required bandwidth for various modulation techniques.

Table 6-1. BER and required bandwidth of different digital modulation schemes.

Modulation	$M = 2^N, N$	P_e	Bandwidth
BPSK	2, 1	$\frac{1}{2} \text{erfc} \left[\frac{E_b}{N_0} \right]^{1/2}$	R_b
QPSK	4, 2	$\frac{1}{2} \text{erfc} \left[\frac{E_b}{N_0} \right]^{1/2}$	$R_b/2$
SQPSK	4, 2	$\frac{1}{2} \text{erfc} \left[\frac{E_b}{N_0} \right]^{1/2}$	$R_b/2$
M-PSK	M, N	$\text{erfc} \left[\frac{NE_b \sin^2 \pi / M}{N_0} \right]^{1/2}$	R_b/N
16-QAM	16, 4	$2 \text{erfc} \left[0.4 \frac{E_b}{N_0} \right]^{1/2}$	$R_b/4$
M-QAM	M, N	$2 \left[1 - \frac{1}{N} \right] \text{erfc} \left[\left(\frac{3}{N^2 - 1} \right) \frac{E_b}{N_0} \right]^{1/2}$	R_b/N
BFSK	2, 1	$\frac{1}{2} \text{erfc} \left[\frac{E_b}{2N_0} \right]^{1/2}$	$2 R_b$
M-FSK	M, N	$\frac{M-1}{2} \text{erfc} \left[\frac{NE_b}{2N_0} \right]^{1/2}$	$N/M R_b$
MSK	4, 2	$\frac{1}{2} \text{erfc} \left[\frac{NE_b}{2N_0} \right]^{1/2}$	$R_b/2$

6-13. Summary

We usually determine and compare the performance of analogue modulation systems on the basis of signal-to-noise ratio (*SNR*) at the receiver input and output. In digital communication systems, an **error ratio** is the ratio of the number of received information entities, bits, characters, or blocks incorrectly to the total number of transmitted elements during a specified time interval. The error ratio is usually expressed in scientific notation, either by signal-to-noise ratio (**SNR**) in analog systems or by bit error ratio (**BER**) in digital systems. For example, 2.5 erroneous bits out of 100,000 bits transmitted in a digital communication system would be 2.5 out of 10^5 or $\text{BER} = 2.5 \times 10^{-5}$. This is equivalent to an SNR of about 26dB.

The Bit Error Ratio (**BER**) is defined as the ratio of the bits wrongly received to all data bits sent. The **BER** is sometimes referred to as **bit error rate**. For a given communication system, the BER is affected by both the data transmission rate and the signal power margin. People usually plot the BER curves to describe the functionality of a digital communication system. In optical communication, BER(dB) versus Received Power (dBm) is usually used; while in wireless communication, BER(dB) versus SNR(dB) is used.

The probability of bit error for ASK may be put in the following form:

$$P_b \text{ (ASK)} = Q \left(\sqrt{\frac{E_b}{N_o}} \right)$$

where $Q[\sqrt{(E_b/N_o)}] = \frac{1}{2} \text{erfc}[\sqrt{(E_b/2N_o)}]$, E_b is energy per bit, $N_o = k_B T$ is the noise spectral density, T is the equivalent noise temperature of the receiver and $\text{erfc}(x)$ is complementary error function of x .

Similarly, the probability of bit error for FSK may be put in the following form:

$$P_b \text{ (FSK)} = Q \left(\sqrt{\frac{1.217 E_b}{N_o}} \right)$$

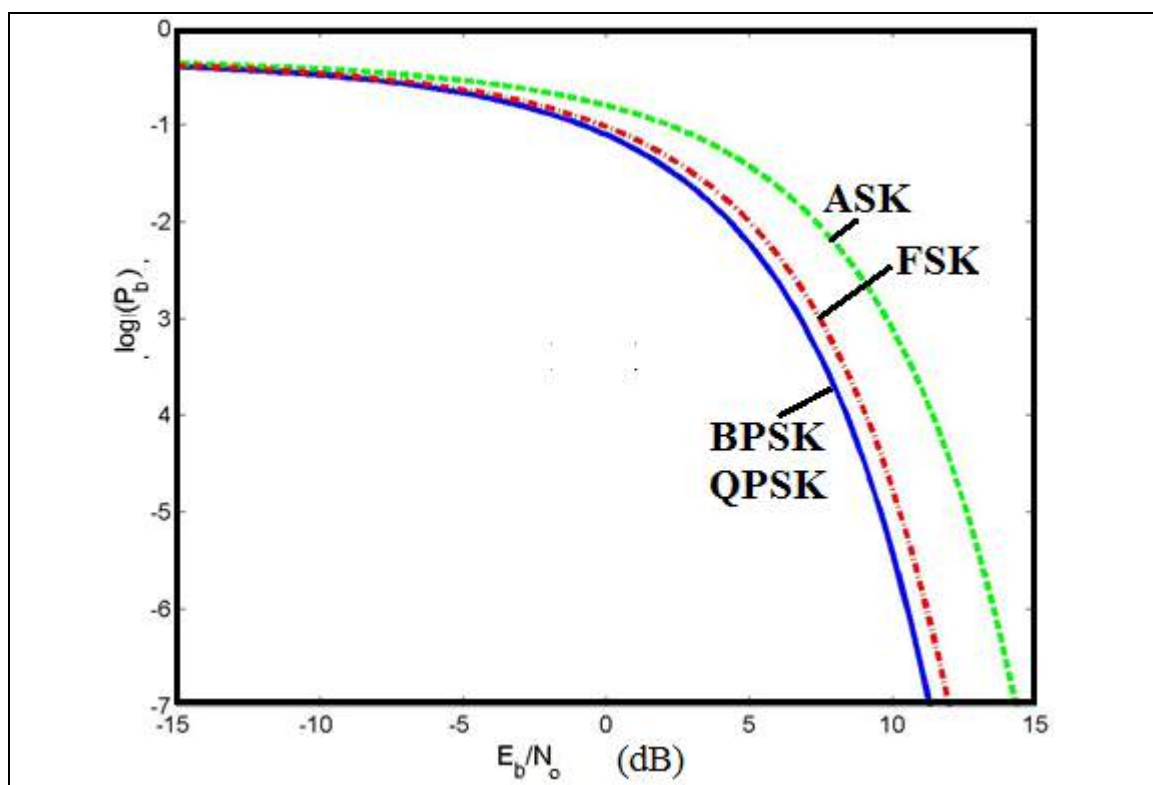
The probability of bit error for BPSK (and QPSK) may be put in the following form:

$$P_b \text{ (BPSK)} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

The average probability of symbol error for coherent M -ary PSK may be put in the following form:

$$P_e = \text{erfc}\left(\sqrt{\frac{E}{N_0}} \sin\left(\frac{\pi}{M}\right)\right)$$

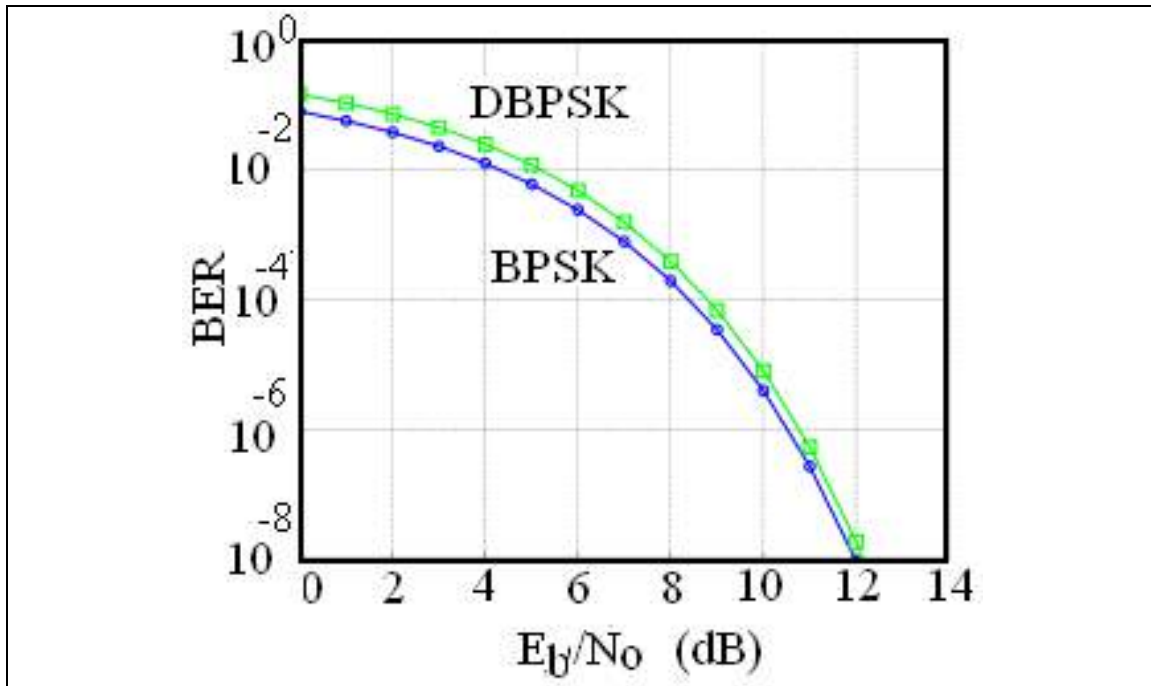
Here, it is assumed that $M \geq 4$, E is energy per symbol. The following figure depicts the BER (or probability of error) versus signal-to-noise ratio



Differential schemes for other PSK modulations may be devised along similar lines. The BER curve for this example is compared to ordinary BPSK. As mentioned above, whilst the error-rate is approximately doubled, the increase needed in E_b/N_0 to overcome this is small. The performance degradation is a result of non-coherent transmission - in this case it refers to the fact that tracking of the phase is completely ignored. Obviously, the probability of error for DPSK is difficult to calculate in general, but, in the case of DBPSK it is given by:

$$P_b (\text{DBPSK}) = \frac{1}{2} \exp (-E_b/N_o)$$

This probability, when numerically evaluated, is only slightly worse than ordinary BPSK, particularly at higher E_b / N_0 values. This probability, when numerically evaluated, is only slightly worse than ordinary BPSK, particularly at higher E_b / N_0 values.



6-14. Problems

6-1) Calculate the BER and required bandwidth of a BPSK with a 1MHz sine wave, and signal to noise ratio (SNR) of 30 dB.

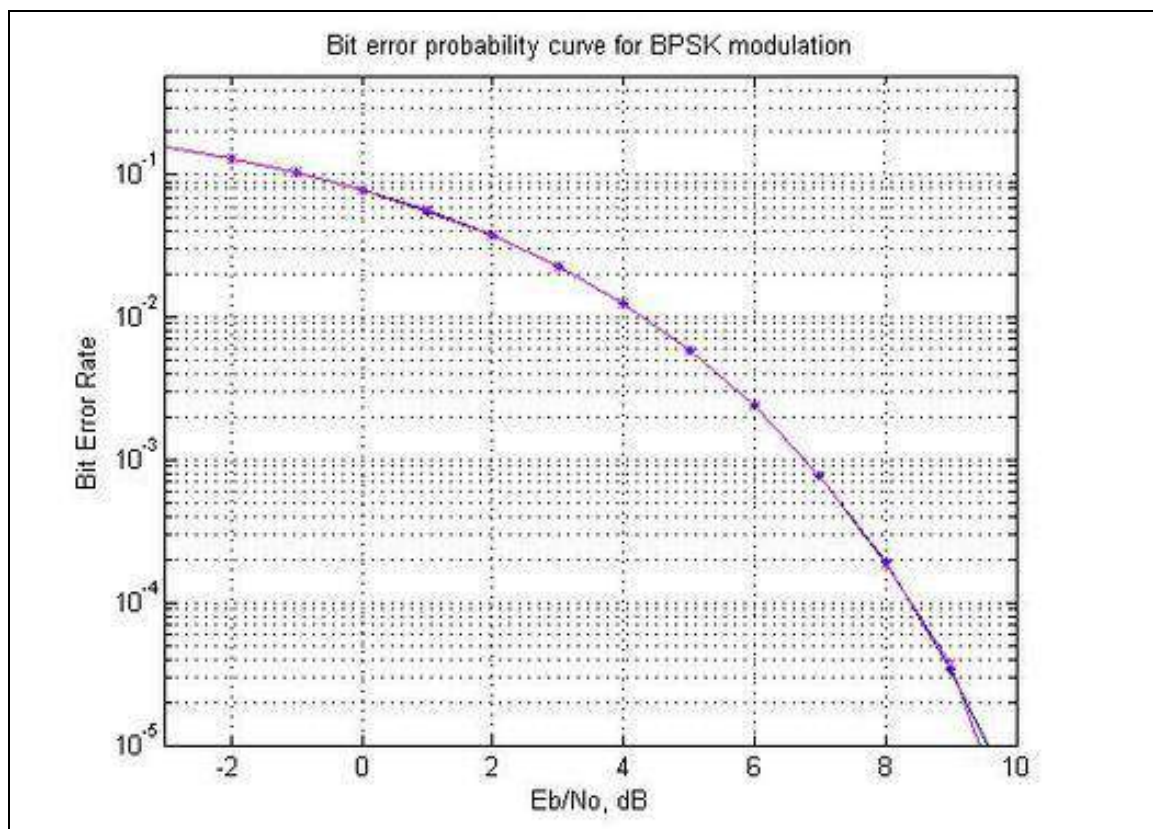
6-2) What's meant by bandwidth efficiency? Compare the bandwidth efficiency of the different types of M-ary PSK (in a table form).

6-3) Consider the case of digital link, over which binary data are transmitted with $R_b=10^9$ bits/s. If the power spectral density of noise at the receiver was 10^{-10} W/Hz, find the average carrier power, which is required to maintain an average error probability $P_e < 10^{-12}$ for coherent BPSK and QPSK? What's the required bandwidth in both cases?

Hint: The average carrier power = Bit energy * Bit rate = $E_b * R_b$.

6-4) Calculate the BER of a QPSK of a 1MHz sine wave and SNR of 20dB. Show, using the phasor diagram, the impact of poor synchronization with the local oscillator. Show also how digital signal processing (DSP) may be used to remove phase and frequency errors.

Hint: Make use of the following chart, for the BER of the BPSK



6-5) Using the "waterfall" curves for M-ary PSK compare the Energy per bit for $M = 8$ and $M = 4$ for a P_b of 10^{-5} . Which modulation scheme gives the best performance?

6-15. References

- [1] H. P. **Hsu**, Analog and Digital Communications, McGraw-Hill, **1993**.
- [2] Couch W. **Leon**, Digital and Analog Communications. Upper Saddle River, NJ: Prentice-Hall, 3rd Edition, **1997**
- [3] B. P. **Lathi**, "**Modern Digital and Analog Communication Systems**", Third edition, Oxford University Press, **1998**.
- [4] Sergio **Benedetto** and Ezio **Biglieri**, Principles of Digital Transmission: With Wireless Applications. Springer. **1999**.
- [5] Simon **Haykin**, "Communication Systems", Fourth Edition, John Wiley & Sons, **2001**.
- [6] Simon **Haykin**, Digital Communications. Toronto, Canada: John Wiley & Sons, 4th Edition, **2001**.
- [7] John G. **Proakis**, and M. Masoud **Saleh**, Digital Communications. Singapore: McGraw Hill, **2007**.

Chapter
7

Error Detection & Correction

Contents

- 7.1. Introduction (Error Detection and Correction)**
- 7-2. Error Detection Schemes**
 - 7-2.1. Repetition Schemes
 - 7-2.2. Parity Scheme
 - 7-2.3. Check Polarity Scheme
 - 7-2.4. Hamming Codes
 - 7-2.5. CRC Scheme
 - 7-2.6. Checksum Scheme
- 7-3. Error Correction Codes (ECC)**
 - 7-3.1. Block Codes
 - 7-3.2. Convolutional Codes
 - 7-3.3. Viterbi Algorithm
 - 7-3.4. Concatenation Codes
 - 7-3.5. List of ECC Codes
- 7-4. Forward Error Correction (FEC) Codes**
- 7-5. Turbo Codes**
 - 7-5.1. Maximum a-posteriori Probability (MAP)
 - 7-5.2. Log Likelihood Ratio (LLR)
 - 7-5.3. Turbo Encoders
 - 7-5.4. Turbo Decoders
 - 7-5.5. Performance of Turbo Coders
- 7-6. List of Error Detection and Correction Codes**
- 7-7 Applications of Error Correcting Codes**
 - 7-7.1. Satellite Communications
 - 7-7.2. Internet
 - 7-7.3. Deep Space Communications

7-7.4. Data Storage

7-7.5. Computer Memory

7-7.6. Mobile Communications

7-8. Information Theory and Error Detection and Correction

7-9, Summary

7-10. Problems

7-11, Bibliography

Chapter
7

Error Detection and Correction

7-1. Introduction to Error Detection and Correction

Error detection is the ability to detect errors caused by noise or other impairments during transmission of data from a transmitter to a receiver. Error correction has an additional feature that enables identification and correction of the errors. Error coding is used in many digital applications like computer memory, magnetic and optical data storage media, satellite and deep space communications, network communications, and cellular telephone networks. Error-detecting and correcting codes operate in general by introducing redundancy to combat errors introduced by the noise in the channel. Rather than transmitting digital data in a raw bit for bit form, the data is encoded with extra bits at the source. Therefore, to detect or correct errors, we need to send extra (redundant) bits with data. The following figure depicts the error detection and correction process throughout a digital link.

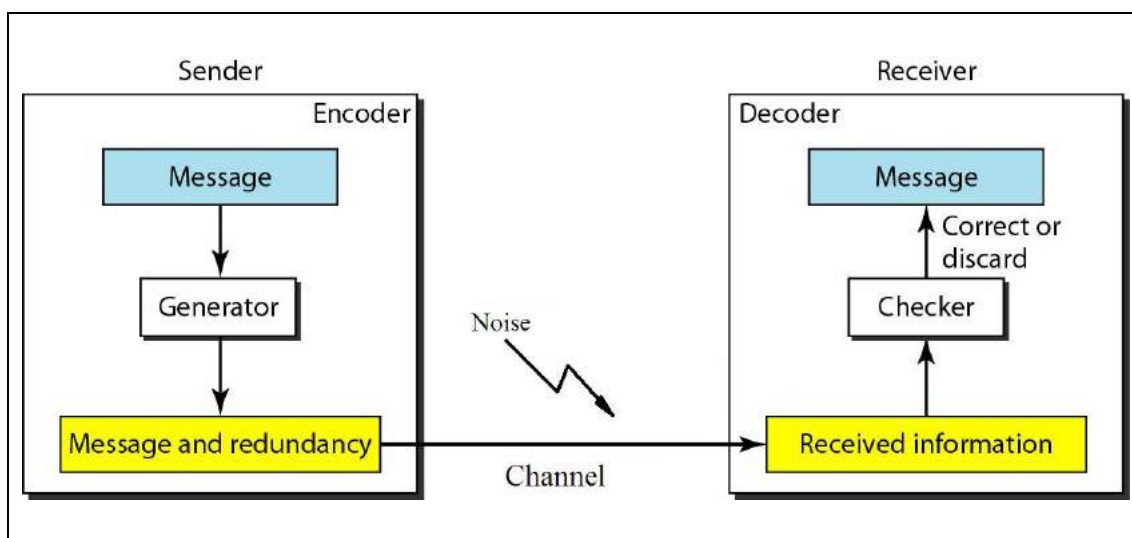


Figure 7-1. Error detection and correction throughout a digital link

7-1.1. Channel Coding Methods

There are *two ways* to design the channel code and protocol for error correcting systems:.

- **Automatic repeat request (ARQ):** The transmitter sends the data and also an error detection code, which the receiver uses to check for errors. If it does not find any errors, it sends a message (an ACK, or acknowledgment) back to the transmitter. The transmitter re-transmits any data that was not ACKed.
- **Forward error correction (FEC):** The transmitter encodes the data with an *error-correcting code* and sends the coded message. The receiver never sends any messages back to the transmitter. The receiver decodes what it receives into the most likely data. The codes are designed so that it would handle an unreasonable amount of noise to trick the receiver into misinterpreting the data.

In this chapter we start by demonstrating the various error detection schemes, which may be used in automatic repeat request (ARQ). Then we demonstrate the different error correcting codes (ECC), which are employed along with the forward error correction (FEC) codes. Figure 7-2 summarizes various error detection and correction codes which we cover in this chapter n codes

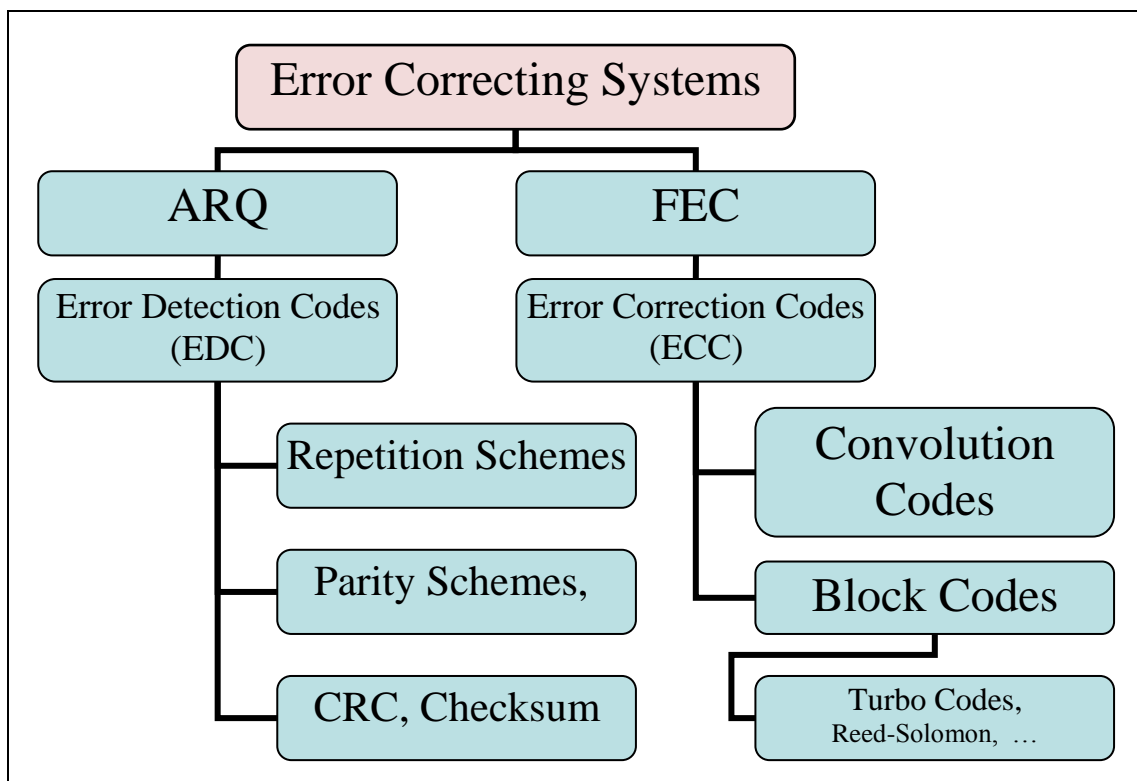


Fig. 7-2. Different error detection and correction methods.

The ARQ technique is used in a number of applications, such as the Global System for Mobile (**GSM**). On the other hand, the more efficient FEC has been recently employed in so many applications, such as the third generation of mobile phones (**3G**).

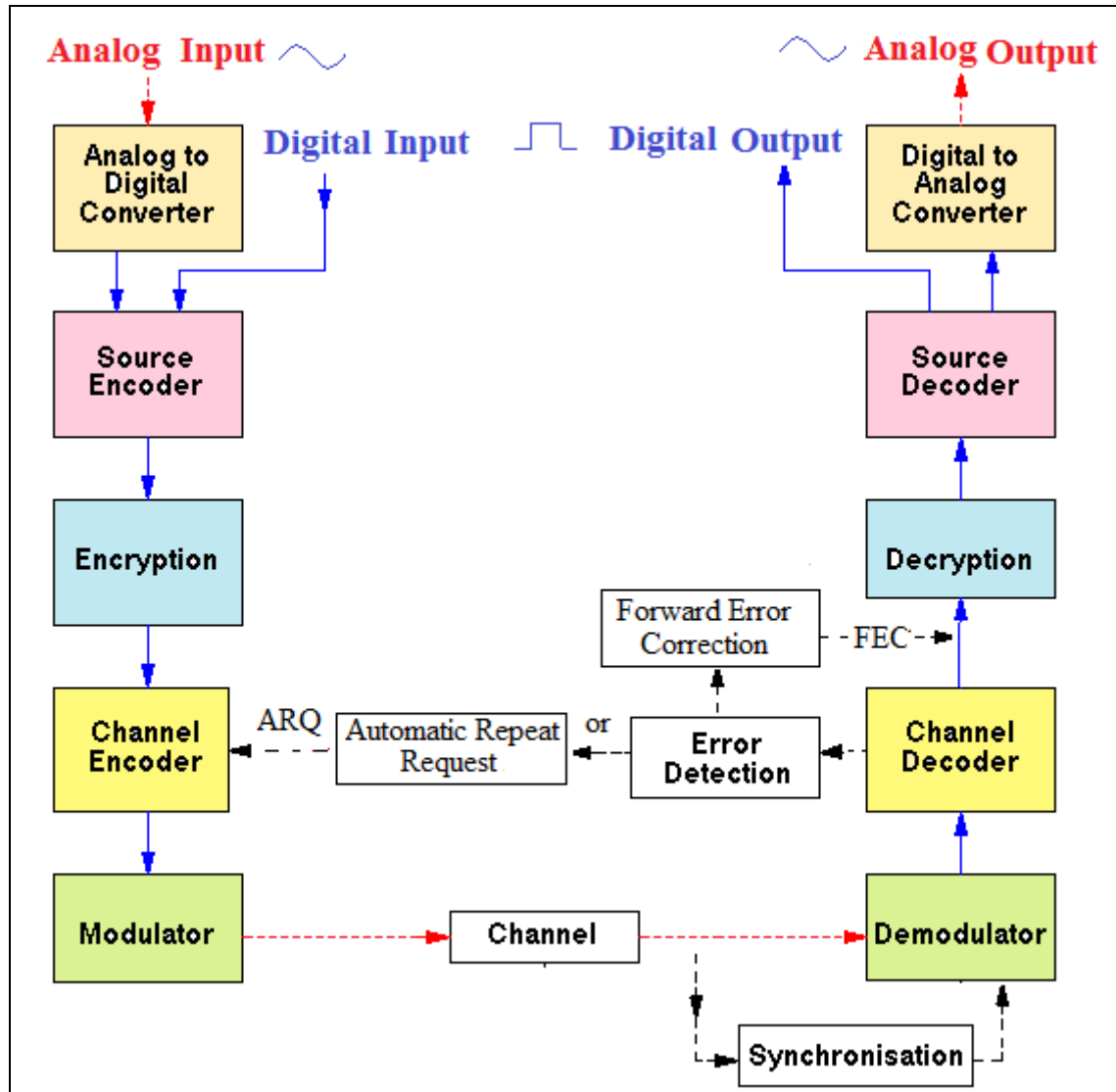


Fig. 7-3. Diagram of a communication system with error detection & correction

7-1.2. Coding Gain & Coding Performance

In coding theory and related engineering problems, **coding gain** is the measure in the difference between the signal to noise ratio (SNR) levels between the uncoded system and coded system required to reach the same bit error rate (BER) levels when used with the error correcting code (ECC).

By introducing channel coding, we reduce the rate of information transfer from 1 bit per channel use, to a fraction $R = (k/n)$ bits per channel use, where k is number of information bits, before adding redundancy (usually grouped

in frames called datawords), and n is number of bits after encoding (usually grouped in frames called codewords). The ratio $R(k/n)$ is called the **code rate**.

Bit Error Rate (BER) is the probability of bit error within data transmission. The signal to noise ratio (SNR), which is usually expressed as (E_b/N_0) is the ratio of the channel power to the noise power. Bit Error Rate (BER) and Signal to Noise Ratio (SNR) of the transmission determine channel performance.

As shown below in figure 7-4, low bit error rates are reduced with channel coding, coded or uncoded. The difference, however, is how much power (SNR) is necessary to achat a fixed SNR. As shown in figure, the difference in SNR between uncoded channel and coded channel (by a certain channel coding method), at a fixed BER, is the code gain.

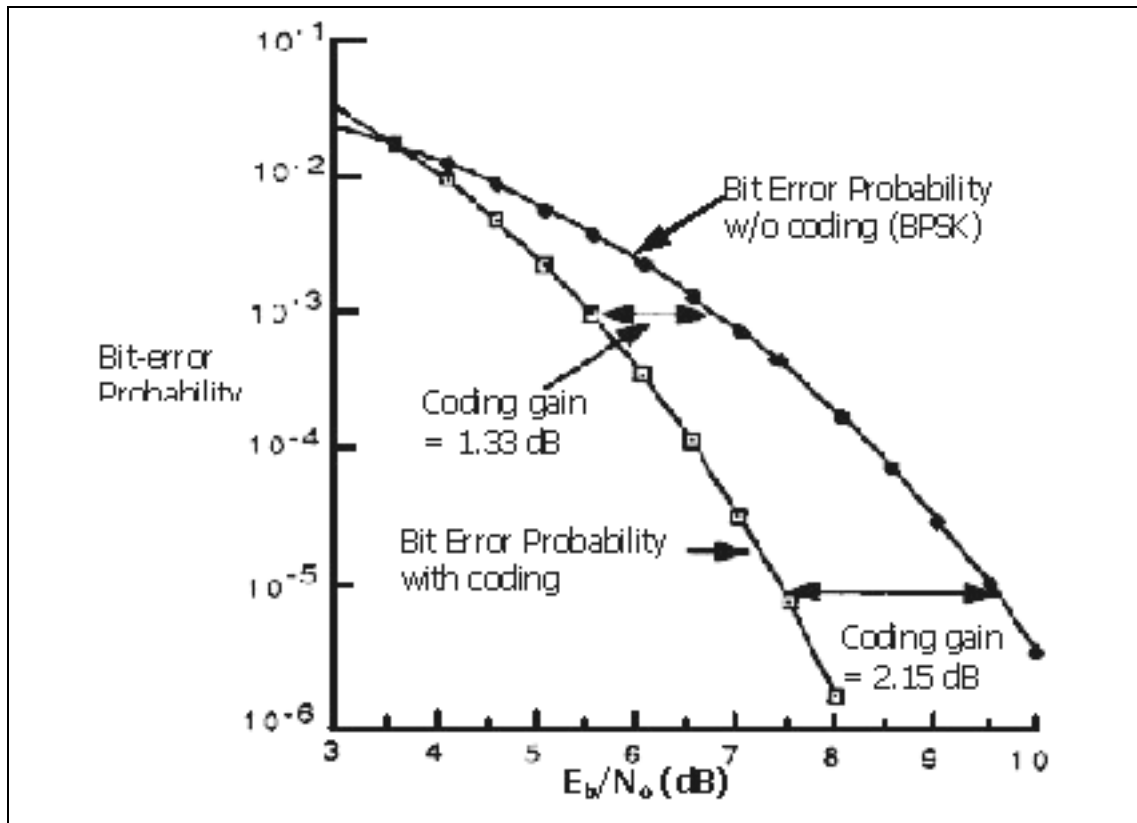


Fig. 7-4. Illustration of the concept of **code gain**, due to channel coding

7-2. Error Detection Schemes

There exist several schemes to achieve error detection, and are generally quite simple. All error detection codes transmit more bits than were in the original data. Most codes are **systematic** so that the transmitter sends the original data bits, followed by **check bits**. Thus, extra bits, referred to as redundancy, accompany data bits for the purpose of error detection. In non-systematic codes, data bits are transformed into at least as many code bits and the transmitter sends only the code bits.

7-2.1. Repetition Schemes

Given a stream of data that is to be sent, the data is broken up into blocks of bits, and in sending, each block is sent some predetermined number of times. For example, if we want to send "1011", we may repeat this block three times each. Suppose we send "1011 1011 1011", and this is received as "1010 1011 1011". As one group is not the same as the other two, we can determine that an error has occurred. This scheme is not very efficient, and can be susceptible to problems if the error occurs in exactly the same place for each group (e.g. "1010 1010 1010" in the example will be considered correct). The scheme however is extremely simple, and is actually used in some transmissions.

Example 7-1.

In the following table, each sent bit is repeated 3 times, so the code rate is (1/3). You may think of the first bit as the message, and bits 2 and 3 as the error correction bits, then the rate turns out to be $1/(1+2) = 1/3$

Received codeword	Decoded as
000	0 (no errors)
001	0
010	0
100	0
111	1 (no errors)
110	1
101	1
011	1

7-2.2. Parity Schemes

The stream of data is broken up into blocks of bits, and the **number of 1 bits is counted**. Then, a "**parity bit**" is set (or cleared) if the number of one bits is odd (or even). This scheme is called odd (or even) parity check.. There is a limitation to parity schemes. A parity bit is only guaranteed to detect an **odd number of bit errors** (1, 3, 5, and so on). If an even number of bits (1, 4, 6 and so on) are flipped, the parity check appears to be correct, even though the data are corrupt.

Table 7-1. Illustration of the single-parity bit checking

Parity bit (odd)	Transmitted Number					
1	1	0	0	1	1	1
	Received Number					
1	1	0	0	1	0	1

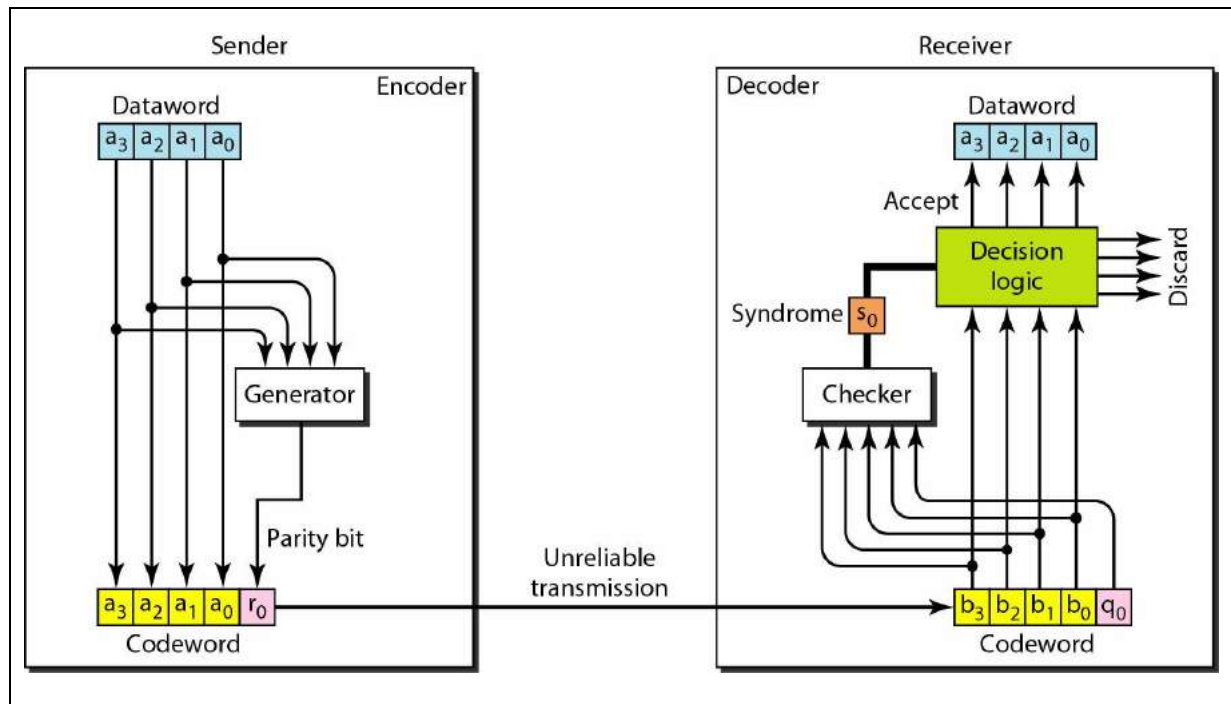


Figure 7-5. Error detection by parity check

7-2-3. Hamming Codes

The so-called *Hamming codes* can help us to detect and correct errors. In the Hamming code, k parity bits are added to an n -bit data word, forming a new word of $n + k$ bits. The bit positions are numbered in sequence from 1 to $n + k$. The positions numbered as power of 2 are reserved for parity bits. The remaining bits are data bits. The code can be used with words of any length.

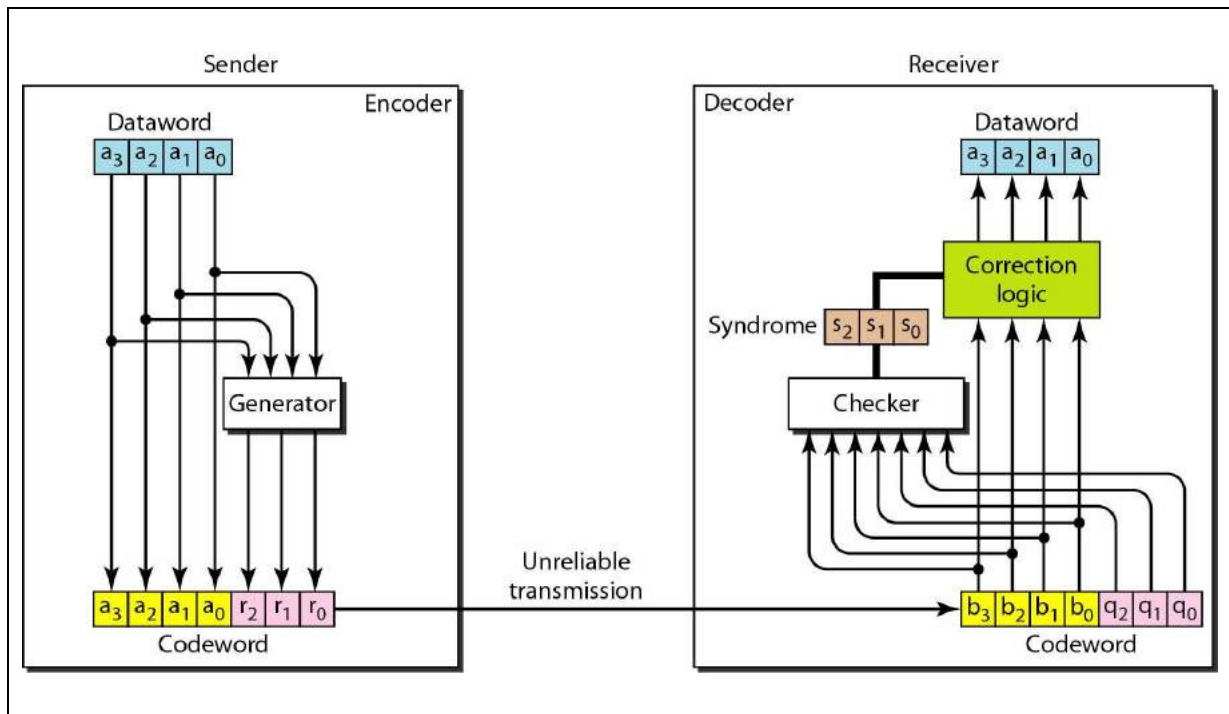


Figure 7-6. Illustration of error detection (and correction) using Hamming codes.

Before giving the general characteristics of the code, we illustrate its operation with a **data word of eight bits**.

1	2	3	4	5	6	7	8	9	10	11	12
0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100
P₁	P₂	1	P₄	1	0	0	P₈	0	1	0	0

The 4 parity bits, *P₁*, *P₂*, *P₄*, and *P₈*, are stuffed in positions 1, 2, 4, and 8, respectively. The 8 bits of the data word are in the remaining positions of the 12-bit codeword. Each parity bit is calculated as follows:

$$P_1 = \text{XOR of bits } (3, 5, 7, 9, 11) = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$$

$$P_2 = \text{XOR of bits } (3, 6, 7, 10, 11) = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$P_4 = \text{XOR of bits } (5, 6, 7, 12) = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$P_8 = \text{XOR of bits } (9, 10, 11, 12) = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

Note that **P₁** covers all bit positions which have the **least significant bit set: 3, 5, 7, 9, 11**. **P₂** covers all bit positions which have the **second least significant bit set: bits 3, 6, 7, 10, 11**. **P₄** covers all bit positions which have the **third least significant bit set: bits 5, 6, 7, 12**. **P₈** covers all bit positions which have the **fourth least significant bit set: bit 9, 10, 11, 12**.

Remember that the exclusive-OR operation performs the odd function: It is equal to 1 for an odd number of 1s in the variables and to 0 for an even number of 1s. Thus, each parity bit is set so that the total number of 1s in the checked positions, including the parity bit, is always even. The 8-bit data word is transmitted or stored in memory together with the 4 parity bits as a **12-bit codeword**. Substituting the 4 P bits in their proper positions, we get:

1	2	3	4	5	6	7	8	9	10	11	12
P_1	P_2	1	P_4	1	0	0	P_8	0	1	0	0
0	0	1	1	1	0	0	1	0	1	0	0

When the 12 bits are received or read from memory, they are checked for errors. The parity is checked over the same combination of bits, including the parity bit. The 4 check bits (*syndrome* bits) are evaluated as follows:

$C_1 = \text{XOR of bits (1, 3, 5, 7, 9, 11)}$
 $C_2 = \text{XOR of bits (2, 3, 6, 7, 10, 11)}$
 $C_4 = \text{XOR of bits (4, 5, 6, 7, 12)}$
 $C_8 = \text{XOR of bits (8, 9, 10, 11, 12)}$

A 0 check bit designates even parity over the checked bits and a 1 designates odd parity. Since the bits were stored with **even parity**, the result, $C = C_8C_4C_2C_1 = 0000$, indicates that no error has occurred. However, if $C \neq 0$, then the 4-bit binary number formed by the check bits (syndrome) gives the position of the erroneous bit. For example, consider the following three cases:

1	2	3	4	5	6	7	8	9	10	11	12	
0	0	1	1	1	0	0	1	0	1	0	0	No error
1	0	1	1	1	0	0	1	0	1	0	0	Error in bit 1
0	0	1	1	0	0	0	1	0	1	0	0	Error in bit 5

In the first case, there is no error in the 12-bit word. In the second, there is an error in bit position number 1. The third case shows an error in bit position 5, with a change from 1 to 0. Evaluating the XOR of the corresponding bits, we determine the 4 check bits to be as follows:

	C_8	C_4	C_2	C_1
For no error:	0	0	0	0
With error in bit 1:	0	0	0	1
With error in bit 5:	0	1	0	1

Thus, for no error, we have $C = 0000$; with an error in bit 1, we obtain $C = 0001$; and with an error in bit 5, we get $C = 0101$. When the binary number C is not equal to 0000, it gives the position of the bit in error. The error can be corrected by complementing (flipping) the corresponding bit. Note that an error can occur in the data word or in one of the parity bits.

The Hamming code can be used for data words of any length. In general, the Hamming code consists of k check bits and n data bits, for a total of $n + k$ bits. The **syndrome** value C consists of k bits, and from which we can detect $k-1$ bit errors. The so-called *Hamming(7,4)* encodes 4 bits of data into 7 bits by adding 3 parity bits. It can correct a single-bit error, or detect single-bit and two-bit errors.

Note 7-1. Code Generating Matrix

Since Hamming code is a linear code, we can write the encoding operation as a matrix multiply (using mod 2 arithmetic): $c = Gd$ where the d is the data-word, c is codeword, and G is called the generating matrix. Similarly, the syndrome bits can be calculated from a syndrome matrix H as follows: $s = H.r$, where r is the received codeword. For instance, the generating and syndrome matrices of the *Hamming(7,4)* are as follows:

$$G := \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad H := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Example 7-3.

Suppose we want to devise a single-bit error-correcting Hamming code for a 17bit data string. How many parity bits are needed? How about a 32bit string.

Solution:

The number of bits actually transmitted is the number of data bits plus the number of parity bits. If we have 16 data bits and only use 4 parity bits, then we would have to transmit 20 bits. Since there are only 16 combinations that can be constructed using 4 parity bits, this is not enough. With 5 parity bits, we must transmit 21 bits. Since there are 32 combinations that can be constructed using 5 parity bits, there will be enough combinations to represent all single-bit errors.

7-2-4. Cyclic Redundancy Check (CRC)

The cyclic redundancy check (CRC) is a sort of checksum algorithms in modern communication systems. The generic name for these check codes, which are appended to blocks of data is a "*Polynomial Code*". Note that the simplest error-detection system, the parity bit, is in fact a simple CRC-1, which uses a 2-bit-long divisor (11 or $1+x$).

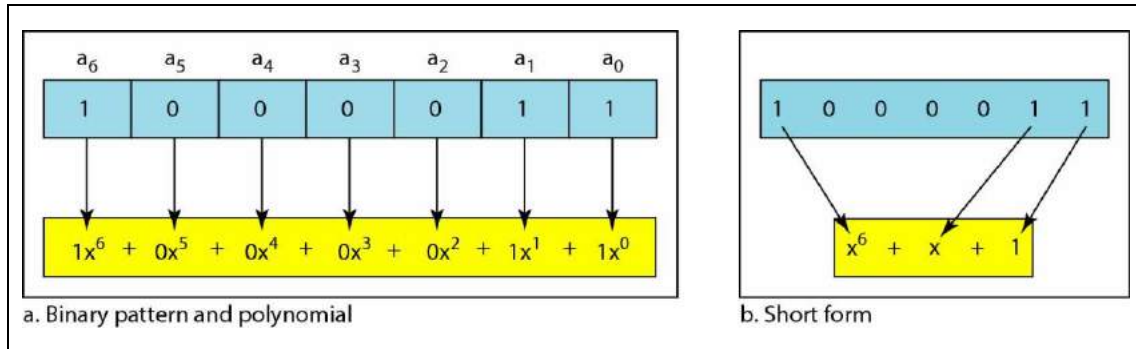


Figure 7-7. CRC polynomials

The table below lists some polynomials of the various algorithms in use. The selection of generator polynomial is the most important part of implementing the CRC algorithm. The polynomial must be chosen to maximize the error detecting capabilities. The most important attribute of the polynomial is its length, because of its influence on the length of the computed checksum.

Table 7-2. CRC polynomials in different applications.

Name	Polynomial	Application
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs

i. CRC Procedure

The theory can be summarized as follows. At first, a CRC polynomial of n -bits is chosen. This sequence is used with the message to calculate a check sequence. The check sequence is appended to the original message. At the receiver, the same division is performed on the message and check sequence combined. If the result is 0, no transmission error is assumed.

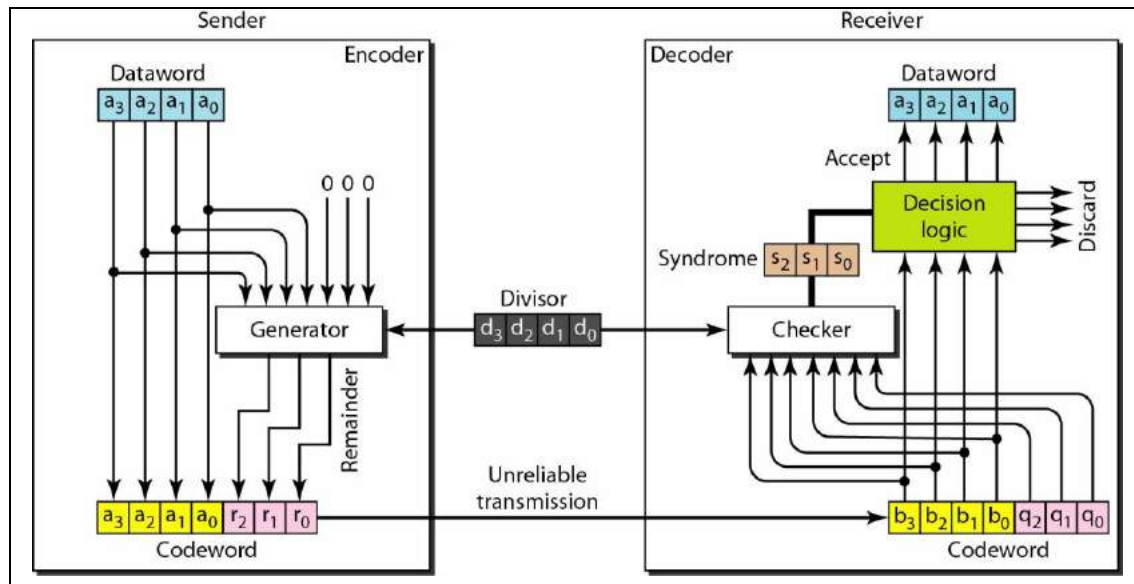


Figure 7-8. Illustration of error detection using CRC technique.

A. CRC Calculation (Transmitter Side)

Take a binary message and convert it to a polynomial then divide it by another predefined polynomial called the **key** or the **CRC polynomial**. The remainder from this division is the CRC. Now append the CRC to the original payload message and transmit the result.

B. CRC Checking (Receiver Side)

The recipient does the same operation (divides the message by the key) and compares this CRC with yours. If they differ, the message must have been mangled. If they are equal, the message went uncorrupted. Most localized corruptions (burst of errors) will be caught using this scheme.

ii. CRC Calculation Example

The CRC algorithm requires the division of the data message polynomial (a) by the key polynomial (d). The straightforward implementation follows the idea of long division. The coefficients of our polynomials are ones and zeros. We start with the leftmost coefficient of data. If it is zero, we move to the next coefficient. If it is one, we subtract the divisor. Except that subtraction modulo 2 is equivalent to XOR. This is shown in the following example, dividing a message $a=110101$ by the key $d=1001$. Remember that the corresponding polynomials are $x^5 + x^4 + x^2 + 1$ and $x^3 + 1$. Since the degree of the key is 3, we append two zeros to our message. We don't bother calculating the quotient, all we need is the remainder (the CRC), which is **011** in this case. The original message with the CRC attached reads **110101011**. You can easily discover that it is divisible by the key, $d=1001$, with no remainder. In practice we don't write the top bit of the key--it is always one. In this example, we only store 001 as key.

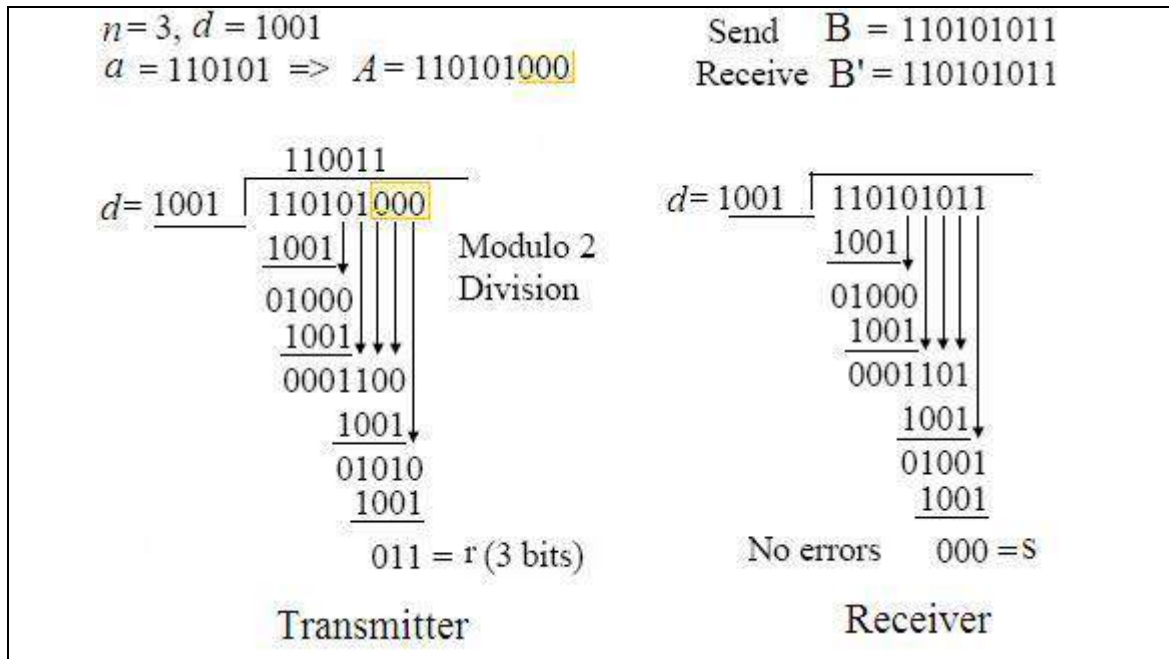


Figure 7-9. Example of error detection using CRC technique.

iii. CRC Implementation

The CRC algorithm can be implemented by software or hardware. The calculation in the above example could be **implemented** using a 3-bit register for storing intermediate results (again, the top bit is always one, so we don't store it). Note that we subtract (or XOR, since this is arithmetic modulo 2) the key from the register every time a 1 is shifted out of it. The following figure depicts the hardware implementation.

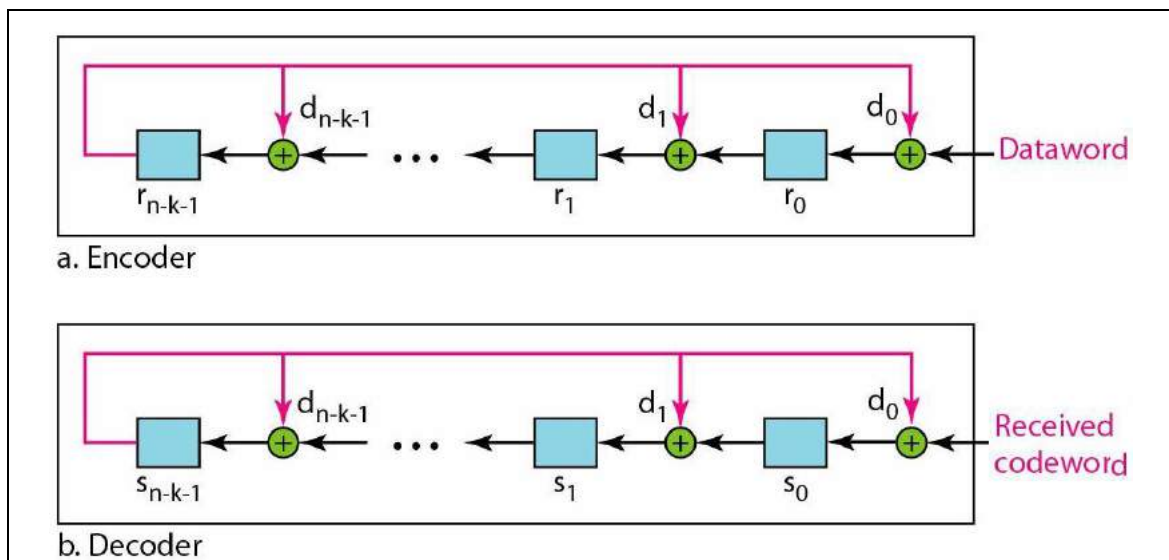


Figure 7-10. Hardware implementation of the CRC encoder and decoder

The following C-list depicts how to implement CRC-32 in software

```

/* CRC-32b version 1.03 by Craig Bruce, 27-Jan-94
#include <stdio.h>
int main();
unsigned long getcrc();
void crcgen();
unsigned long crcTable[256];
/*****
int main(int argc, char *argv[])
{
    int        i;
    FILE       *fp;
    unsigned long  crc;
    crcgen();
    if (argc < 2) {
        crc = getcrc( stdin );
        printf("crc32 = %08lx for <stdin>\n", crc);
    } else {
        for (i=1; i<argc; i++) {
            if ( (fp=fopen(argv[i],"rb")) == NULL ) {
                printf("error opening file \"%s\"!\n",argv[i]);
            } else {
                crc = getcrc( fp );
                printf("crc32 = %08lx for \"%s\"\n",
                    crc, argv[i]);
                fclose( fp );
            }
        }
    }
    return( 0 );
}*****/
unsigned long getcrc(FILE *fp)
{
    register unsigned long  crc;
    int c;
    crc = 0xFFFFFFFF;
    while( (c=getc(fp)) != EOF ) {
        crc = ((crc>>8) & 0x00FFFFFF) ^ crcTable[ (crc^c) & 0xFF ];
    }
    return( crc^0xFFFFFFFF );
}*****/
void crcgen()
{
    unsigned long  crc, poly;
    int i, j;
    poly = 0xEDB88320L;
    for (i=0; i<256; i++) {
        crc = i;
        for (j=8; j>0; j--) {
            if (crc&1) {
                crc = (crc >> 1) ^ poly;
            } else {
                crc >>= 1;
            }
        }
        crcTable[i] = crc;
    }
}

```

7-2-5. Checksum

A checksum of a message is an arithmetic sum of message code words of a certain word length, for example byte values, and their carry value. The sum is negated by means of one's-complement, and stored or transferred as an extra code word extending the message. On the receiver side, a new checksum may be calculated from the extended message. If the new checksum is not 0, an error is detected. Checksum has several schemes include parity bits, check digits, and longitudinal redundancy check.

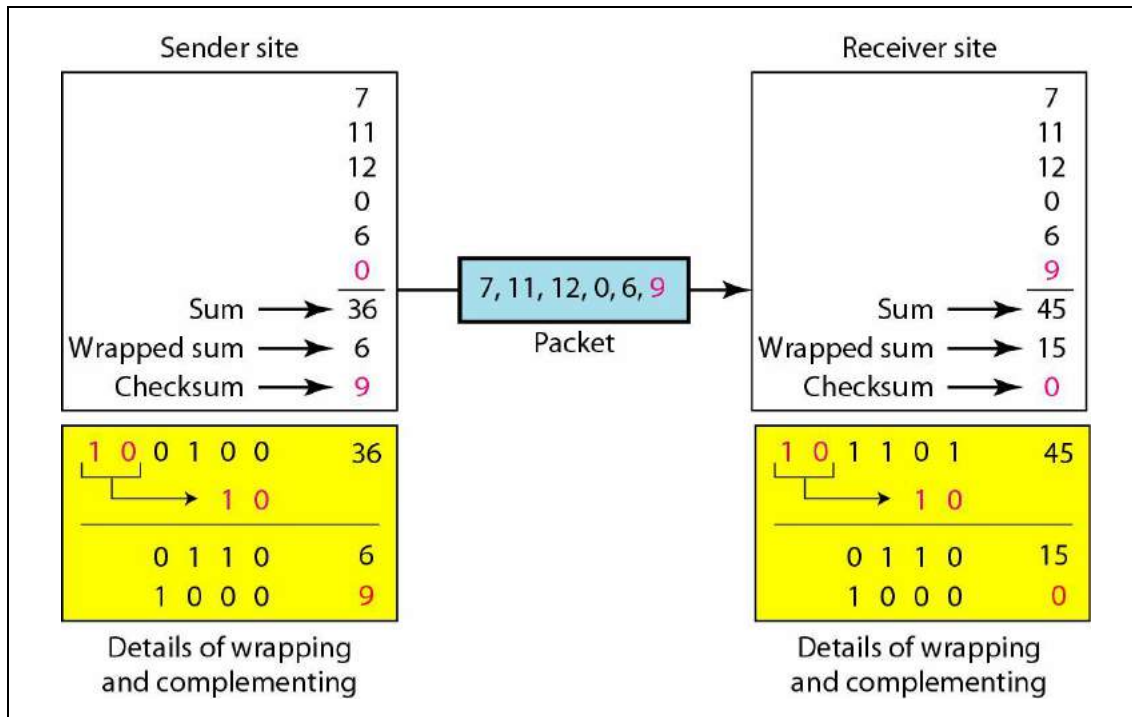


Figure 7-11 Illustration of the checksum error detection technique.

Checksum Generator

- In the sender,
- The Checksum generator subdivides data unit into equal segments of n bits (usually 16).
 - These segments are added together using one's complement arithmetic in such a way that the total is also n bits long.
 - The total (sum) is then complemented and appended to the end of the original data unit as redundancy bits, called the checksum field.
 - The extended data unit is transmitted across the network.
 - If the sum of the data segment is T , the checksum will be $-T$.

Checksum Checker

- In the recipient,
- The receiver subdivides the data unit
 - The receiver then adds all segments together and complements the result.
 - If the extended data unit is intact, the total value found by adding the data segments and the checksum field should be zero.
 - If the result is not zero, the packet contains an error and the receiver rejects it.

7-3. Error-Correcting Codes (ECC)

The error detection methods are sufficient to determine whether some data has been received in error. But often, this is not enough. An error-correcting code (ECC) is a code in which each data signal conforms to specific rules of construction so that departures from this construction in the received signal can generally be automatically detected and corrected. ECC methods are used in digital communication systems as well as computer memory and data storage.

In general, the error correcting methods put redundant information into the data stream following certain algebraic or geometric relations so that the decoded stream, if damaged in transmission, can be corrected. The effectiveness of the coding scheme is measured in terms of **Code Rate**, which is the code length divided by the useful information, and the **Coding Gain**, which is the difference of the SNR levels of the uncoded and coded systems required to reach the same BER levels. Coding only makes sense if the error rate using coding is less than without coding, such that the coding gain is positive. An error-correcting code which corrects all errors of up to n bits correctly is also an error-detecting code which can detect at least all errors of up to $2n$ bits.

There are two main categories of ECC codes, namely:

- 1- **Block (Algebraic) Codes.**
- 2- **Convolution Codes and**

Convolution codes process the incoming bits in **streams** rather than in blocks. As shown in figure 7-14, the Block codes such as Hamming codes and Reed-Solomon codes break a message stream up into fixed size blocks and add redundancy symbols to offer error correction capability. Binary convolutional codes take a different approach to error control coding. The message stream is not broken up into blocks, and appended with redundancy. Instead, redundancy is added continuously and is dependent on past bits of the input message. This converts the entire message stream into one long codeword (or code frame).

Some types of **block** coding are most effective in combating burst errors. However, convolutional coding is generally more robust when faced with random errors or white noise. In addition to the above two basic categories of error correcting codes, we may add the so-called **concatenated codes**, which are composed of some parallel and series or hybrid configurations of the basic codes.

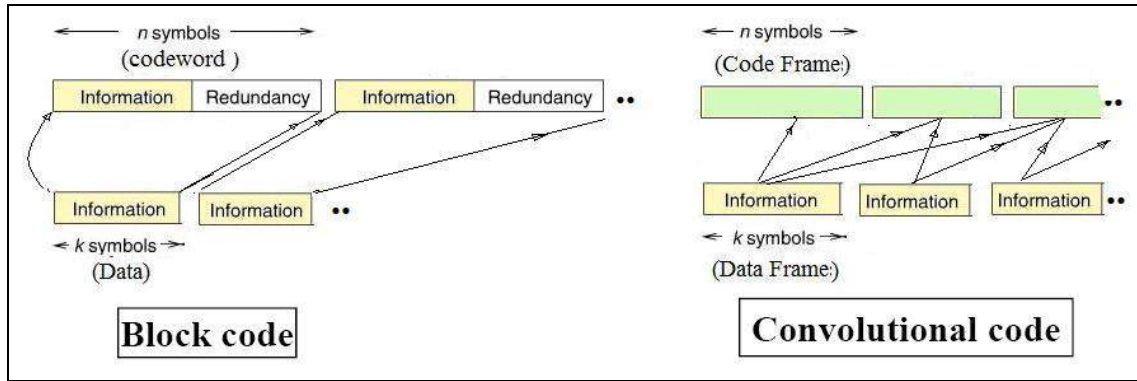


Figure 7-12. . Input and output data configuration in block and convolutional codes.

7-3.1. Block (Algebraic) Codes

In **Block** (or **Algebraic**) coding, the encoder intersperses parity bits into the data sequence using a particular algebraic algorithm. On the receiving end, the decoder applies an inverse algorithm to identify and correct errors due to channel noise.

In block coding, we divide our message into blocks, each of k bits, called **data-words**. We add r redundant bits to each block to make the length $n = k + r$. The resulting n -bit blocks are called **codewords**.

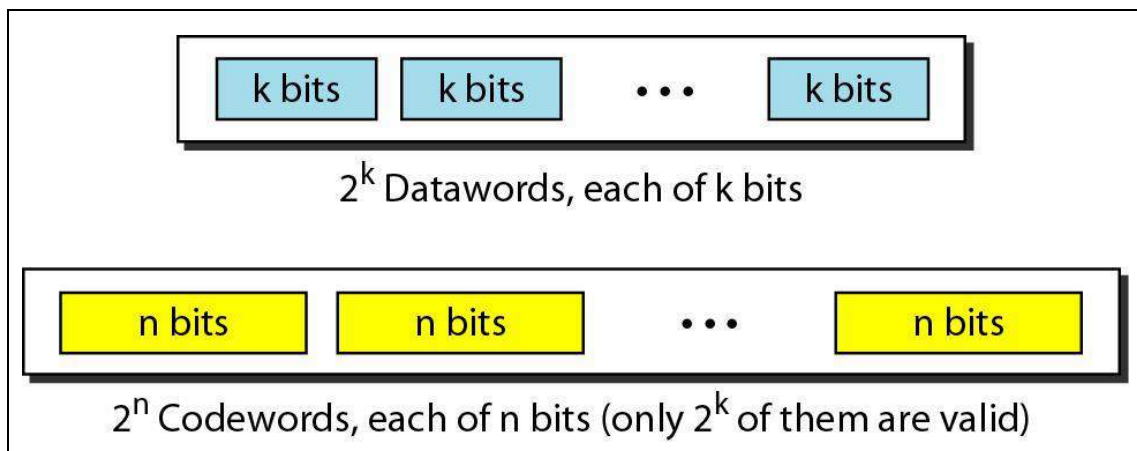


Figure 7-13. Illustration of the block coding input and output data streams.

Almost all block codes used today belong to a subset called linear block codes. Linear block codes are so named because each codeword is a linear combination of a set of generator code words. If the messages are k bits long, and the codewords are n bits long (where $n > k$), there are k linearly independent codewords of length n that form a generator matrix. To encode a message of k bits, you simply multiply the message vector u by the generator matrix to produce a codeword vector v that is n bits long.

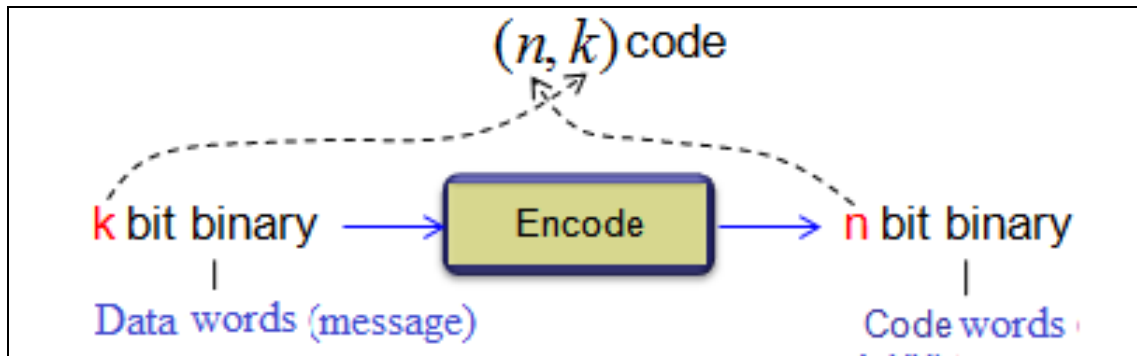


Fig. 7-14. Operation of algebraic (block) encoder.

For the matter of illustration, consider the following table, which shows datawords and codewords of certain block code. We add 3 redundant bits to detect and correct errors and see if the receiver can correct an error without knowing what was actually sent. We add 3 redundant bits to the 2-bit dataword to make 5-bit codewords. The following table shows the datawords and codewords. Assume the dataword is 01. The sender creates the codeword 01011. Assume the codeword is corrupted during transmission, and 01001 is received. First, the receiver finds that the received codeword is not in the table. This means an error has occurred. Assuming that there is only 1 bit corrupted, the receiver uses the following strategy to guess the correct dataword.

Table 7-3. Example of datawords and codewords in a linear block code

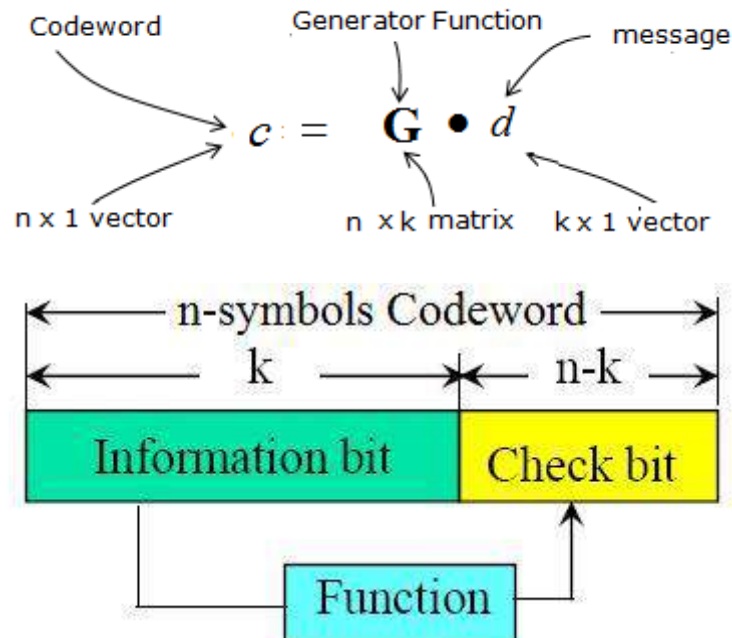
Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

1. Comparing the received codeword with the first codeword in the table (01001 versus 00000), the receiver decides that the first codeword is not the one that was sent because there are two different bits.
2. By the same reasoning, the original codeword cannot be the third or fourth one in the table.
3. The original codeword must be the second one in the table because this is the only one that differs from the received codeword by 1 bit. The receiver replaces 01001 with 01011 and consults the table to find the dataword 01.

Linear block codes are very easy to implement in hardware, and since they are algebraically determined, they can be decoded in constant time. They have very high code rates and low coding overhead, but they have limited error correction capabilities.

Note 7-2. Generating Function & Code Generating Matrix

One of the ways to perform coding is to use what is called 'Generator Function'. It looks similar to what we learned in linear algebra but the way they are used is different



Let's suppose you have a message $d = (1 \ 1 \ 0 \ 1)$

Now, let's figure out a codeword for this when this message goes through the encoder block. This is the point where we use the generator function. Assume we have the following four generating functions

$$\begin{aligned}
 g_0(x) &= 1 + 1x + 0x^2 + 1x^3 + 0x^4 + 0x^6 + 0x^7 = 1 + x + x^3 \\
 g_1(x) &= 0 + 1x + 1x^2 + 0x^3 + 1x^4 + 0x^6 + 0x^7 = 0 + x + x^2 + x^4 \\
 g_2(x) &= 1 + 1x + 1x^2 + 0x^3 + 0x^4 + 1x^6 + 0x^7 = 1 + x + x^2 + x^6 \\
 g_3(x) &= 1 + 0x + 1x^2 + 0x^3 + 0x^4 + 0x^6 + 1x^7 = 1 + x^2 + x^7
 \end{aligned}$$

You can pack these equations into a matrix as follows. $G = [g_0, g_1, g_2, g_3]$

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Such a generator generating/matrix is usually given to you from the specification of the communication system. The process to calculate the **codeword** using the generator function is as simple as follows:

:

$$c = G \cdot d = [1 \cdot g_0 + 1 \cdot g_1 + 0 \cdot g_2 + 1 \cdot g_3]$$

$$\begin{array}{r}
 (1\ 1\ 0\ 1\ 0\ 0\ 0) \\
 (0\ 1\ 1\ 0\ 1\ 0\ 0) \\
 (0\ 0\ 0\ 0\ 0\ 0\ 0) \\
 +) (1\ 0\ 1\ 0\ 0\ 0\ 1) \\
 \hline
 \text{mod } 2) (2\ 2\ 2\ 1\ 1\ 0\ 1) \\
 \hline
 (0\ 0\ 0\ 1\ 1\ 0\ 1) \leftarrow \text{-Codeword for (1,1,0,1)}
 \end{array}$$

7-3.2. Convolutional Codes

The other alternative forward error-correcting technique, known as convolutional coding, was first introduced in 1955. Convolution codes process the incoming bits in **streams** rather than in blocks. As we have stated earlier, the binary convolutional codes (BCC) take a different approach to error control coding. The message stream is not broken up into blocks, but rather, redundancy is added continuously depending on current input and past bits of the input message. This converts the entire message stream (data frame) into one long codeword (code frame).

The basic feature of binary convolutional codes (BCC) is that the encoding of any bit is strongly influenced by the bits that preceded it (the memory of past bits). Convolutional codes are commonly specified by 3 parameters, namely:

n = number of input bits (per symbol)
 k = number of output bits,
 m = number of memory bits (flip-flops)

The quantity $r = k/n$ is called the code rate, and it is a measure of the efficiency of the code. Sometimes, the manufacturers of convolutional decoders specify the code by (n, k, L) , where $L = k(m-1)$ is called the constraint length. The constraint length, L , represents the number of bits in the encoder memory that affect the generation of n output bits. Other manufacturers refer to convolutional codes by (r, K) , where $K = L-1$.

i. Implementation of a Convolutional Encoder

Conventionally encoding the data can be **implemented** using a **shift register** (chain of flip flops) and some combinatorial logic that performs modulo-2 addition. The modulo-2 adder is a combinatorial logic, which is often in the form of cascaded **XOR** gates. In order to encode data, we start with a register of k flip flops, each holding 1 input bit. Generally, all flip flops start with a 0 value. The encoder has n modulo-2 adders, and n generator polynomials; one for each adder, as shown in figure 7-15. An input bit m_1 is fed into the leftmost register. Using the generator polynomials and the existing values in the remaining registers, the encoder outputs n bits. Now bit shift all register values to the right (m_1 moves to m_0 , m_0 moves to m_{-1}) and wait for the next input bit. If there are no remaining input bits, the encoder continues output until all registers have returned to the zero state. The binary convolutional encoder (BCC) shown below is a rate $1/3$ ($k=1, n=3$) with constraint length $K=2$.

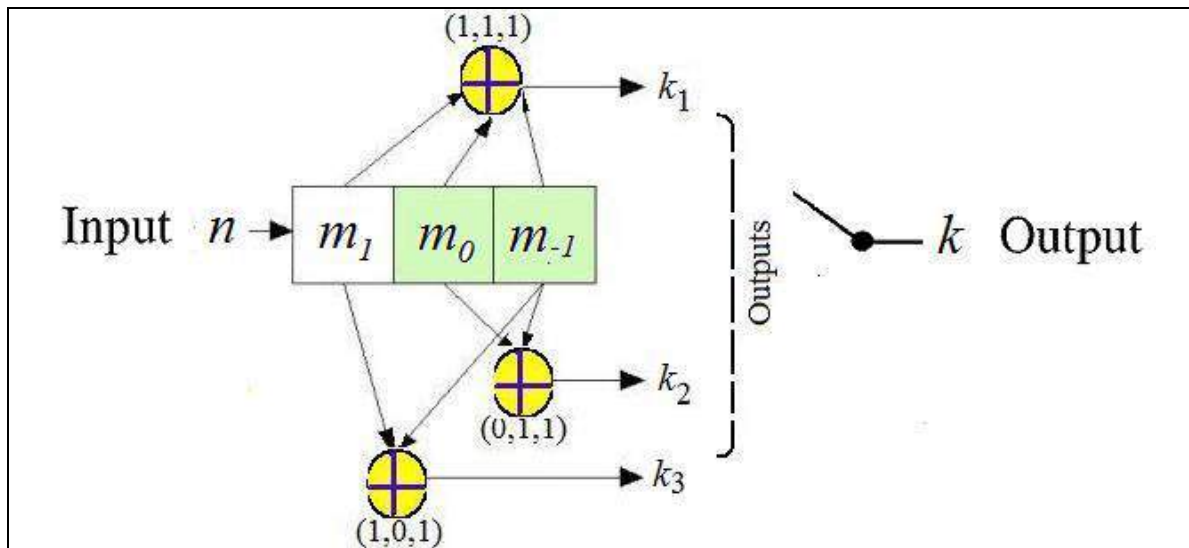


Fig. 7-15. Example of a rate $1/3$ convolutional encoder

The above encoder example may be considered as a **state machine**. The next state of this machine is determined by the next input bit combined with the previous two input bits which were stored in the shift register. Each input bit is coded into 3 output bits. The selection of which bits are to be added to produce the output bit is called the generator polynomial G . In this example, the code generator polynomials, which may be considered as the adding weights of the modulo-2 adders (XOR), are given by: $G_1 = (1,1,1)$, $G_2 = (0,1,1)$, and $G_3 = (1,0,1)$. Therefore, output bits are calculated (XOR) as:

$$k_1 = m_1 + m_0 + m_{-1}, \quad k_2 = m_0 + m_{-1}, \quad k_3 = m_1 + m_{-1}. \quad (7-1)$$

The output bits just the sum of these bits. Sometimes, the binary convolutional code is represented by the so-called **transfer function matrix**, usually termed as G . For instance, the above example of BCC encoder, with 1/3 rate, has the following transfer function matrix:

$$G = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad (7-2)$$

This coder polynomials may be also written as: $G_1 = 1 + D + D^2$, $G_2 = D + D^2$ and $G_3 = 1 + D^2$, where D is the unit delay. The output codeword is generated by shifting and multiplication: $k(D) = m(D).G(D)$.

ii. Presentation Methods of a Convolutional Encoder

Generally speaking, convolution coders can be represented by one of the following methods:

- Shift registers,
- State diagrams, (of a state machine),
- Tree diagrams or
- Trellis diagrams.

Example 7-3.

Consider a rate 1/2 convolutional encoder, and illustrate how it works. The encoder shift register starts with zeros at all three stored locations (i.e. 0, 0, 0). The input data sequence to be encoded is 1, 1, 0, 1 in this example. The shift register contents thus become, after each data bit arrives and propagates into the shift register: 100, 110, 011, 101.

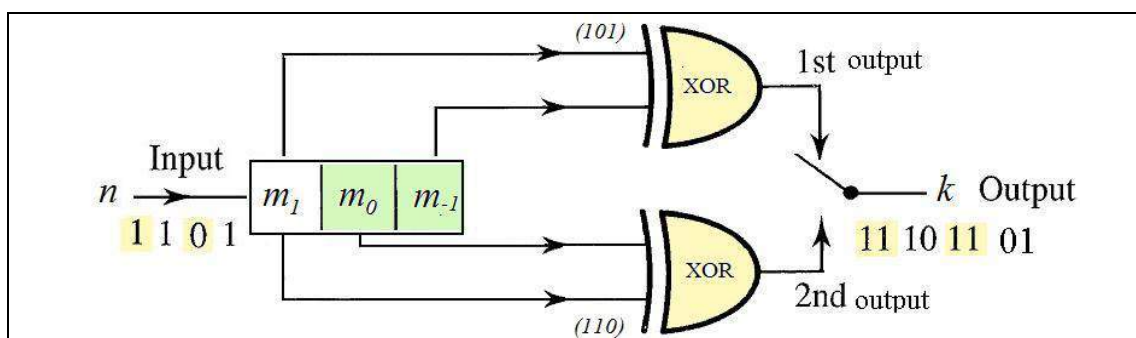


Fig. 7-16. Another example of a rate 1/2 Convolutional encoder.

As there are two outputs for every input bit the above encoder is rate 1/2. The first output is obtained after arrival of a new data bit into the shift register when the switch is in the upper position, the second with the switch in the lower position. Thus, in this example, the switch will generate, through the

XOR gates, from the four input data bits: 1, 1, 0, 1, the corresponding four output digit pairs: 11, 10, 11, 01. This particular encoder has 3 stages shift register, and therefore we say that the constraint length $K=3$. We can consider the coder outputs from the exclusive OR gates as being generated by two polynomials: $G_1 = (1,0,1)$, and $G_2 = (1,1,0)$,

A. State Diagram Representation

We can regard this as a **Mealy state machine** with 4 states corresponding to all the possible combinations of the first two stages in the shift register.

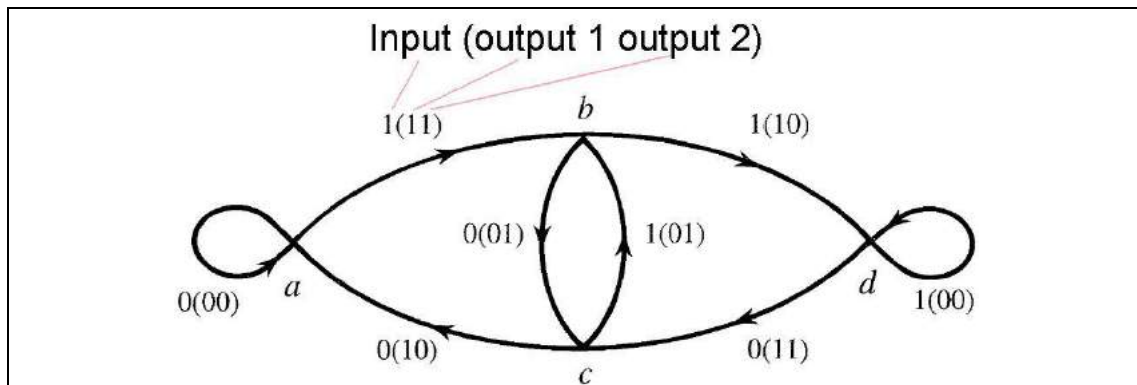


Fig. 7-17. Transition-state diagram of a rate 1/2 Convolutional encoder example

B. Tree State Diagram

The **tree diagram** for this state machine (starting from the zero states or conditions) is shown in figure 7-18.

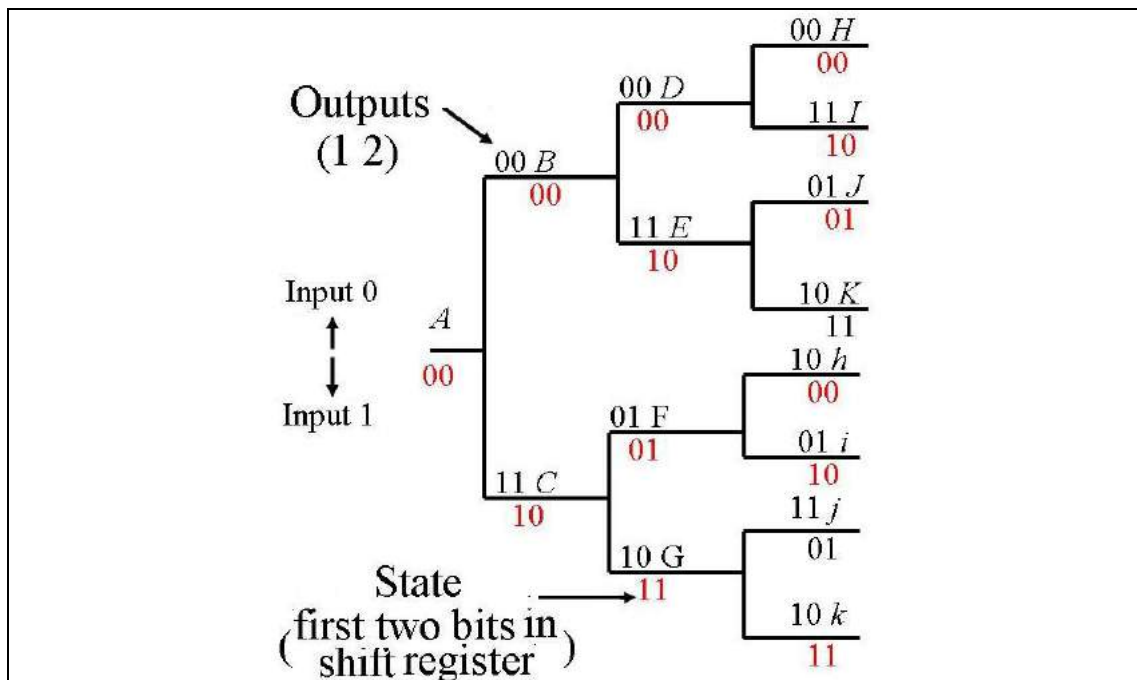


Fig. 7-18. Tree-state diagram of a rate 1/2 Convolutional encoder example

The encoder starts in state A holding two zeros (00) within the first two stages of the shift register. If the next input bit is a zero (0) we follow the upper path to state B where the stored data is updated to 00. If the next input bit is a one (1) we follow the lower path to progress to the corresponding state C where the stored data is now 10. The convention is to enter the updated new stored state values below the state letter. One can derive the output data bits generated within the encoder as follows. For state B these are 00 and for state C these are 11. States B/C correspond to the arrival of the first new data bit to be encoded, while D/E/F/G correspond to the second data bit and H/I/J/K/h/i/j/k the third data bit, and so on

C. Trellis State Diagram

The tree diagram can be folded into a trellis, as shown in figure 7-19. As the constraint length is $K=3$ we have $2^{K-1} = 4$ unique states: 00, 01, 10, 11. As shown in figure, the states are shown as 00x to denote the third bit, x, which is lost or discarded following the arrival of a new data bit. Note that the upper path means input 0 and the lower path means input 1. Also the output is written above arrows. Note also the horizontal arrangement of states A, B, D, H and L. The same applies to states C, E, I and M. The horizontal direction corresponds to time, such that the whole trellis diagram in figure corresponds to encoding 4 input data bits. The vertical direction here corresponds to the stored state values a, b, c, d in the shift register.

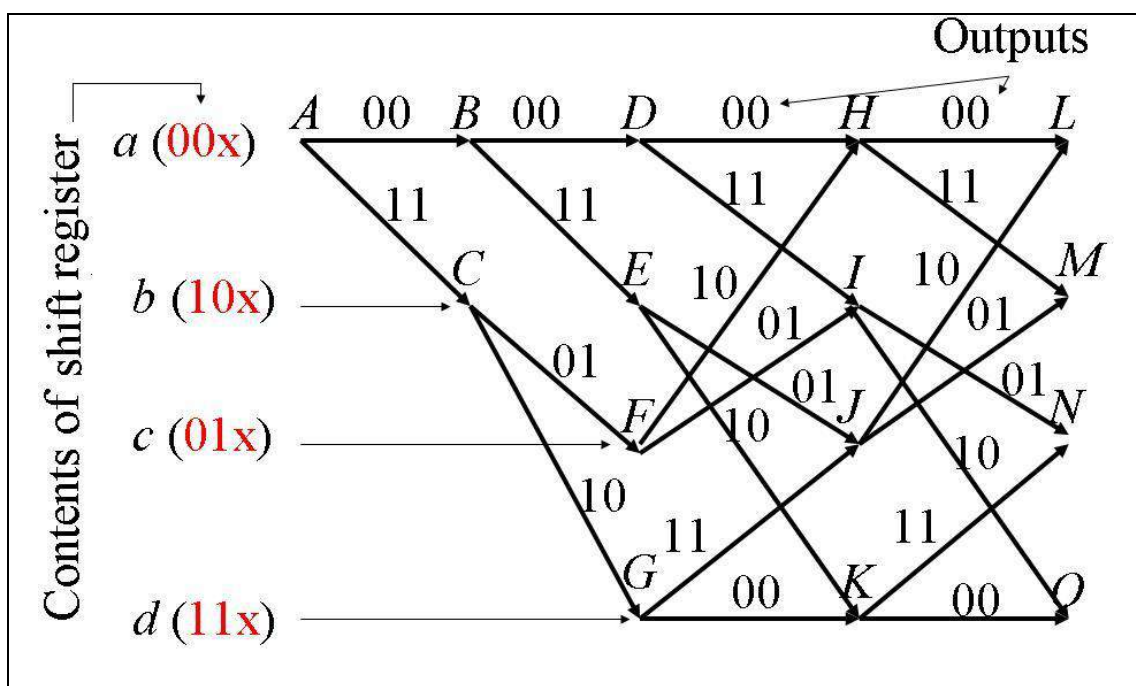


Fig. 7-19.. Trellis-state diagram of a rate $\frac{1}{2}$ Convolutional encoder example

iii. Implementation of a Convolutional Decoder

There are several different approaches to decoding of convolutional codes. These are grouped in two basic categories

- **Sequential** decoding (e.g., **Fano** algorithm)
- **Maximum Likelihood** decoding (e.g., **Viterbi** algorithm)

7-3.3. Vitterbi Algorithm Decoders

Viterbi algorithm (named after Andrew **Viterbi**) is one of several algorithms for decoding convolutional codes over noisy digital links. The algorithm has found universal application in digital cellular phones, modems, satellite transceivers, deep-space communications, speech recognition and wireless networks. The Viterbi algorithm is used to find the most probable sequence of hidden states given a sequence of observed states. At each bit-interval, the Viterbi algorithm compares the actual received code bits with the code bits that might have been generated for each possible memory-state transition. For instance, in speech-to-text (speech recognition), the acoustic signal is treated as the observed sequence of events, and a string of text is considered to be the *hidden cause* of the acoustic signal. Then, the Viterbi algorithm finds the most probable string of text corresponding to the input acoustic signal. The Viterbi decoders are state-machine systems. That is, at any time the system being modeled is in some state. There are a finite number of states. While multiple sequences of states can lead to a given state, at least one of them is a most likely path to that state, called the survivor path. This is a fundamental assumption of the algorithm because the algorithm will examine all possible paths leading to a state and only keep the most likely one. The most important concept to aid in understanding the Viterbi algorithm is the trellis diagram.

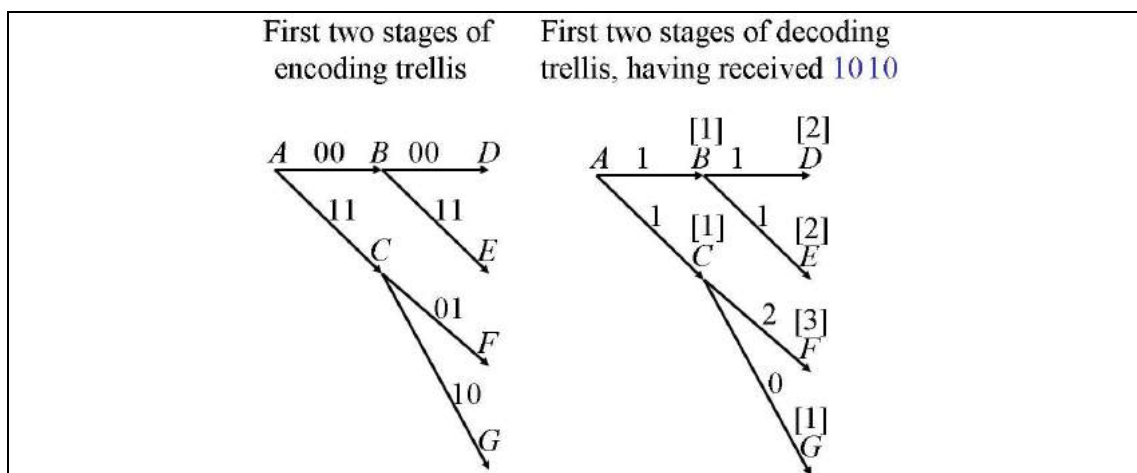


Fig. 7-20(a). Part of the trellis diagram (first and second stage) of a Viterbi decoder of the convolutional coder of example 7-4 (rate 1/2).

Any section of the trellis shows the possible state transitions and output codewords for one period of the decoder. The figure below shows a part of the trellis diagram of a convolutional codec (with rate $r = k/n = 1/2$ with constraint length $K = 3$). Only two paths (first and second stage) of the decoding trellis after receiving second pair of data bits are shown in figure. We continue building our decoding trellis until it is complete after receipt of all ten data bits, as shown in figure 7-20(b). If we have two paths to a state, as in the later states: H, I, J, K, L, M, N, P, we write the smaller (more likely) Hamming distance in square brackets above the state and discard the larger distance (as this is less likely to represent the correct path). In our example, we assumed the last two bits were 0, so we must expect to finish back in state P, which is the same as the starting state A.

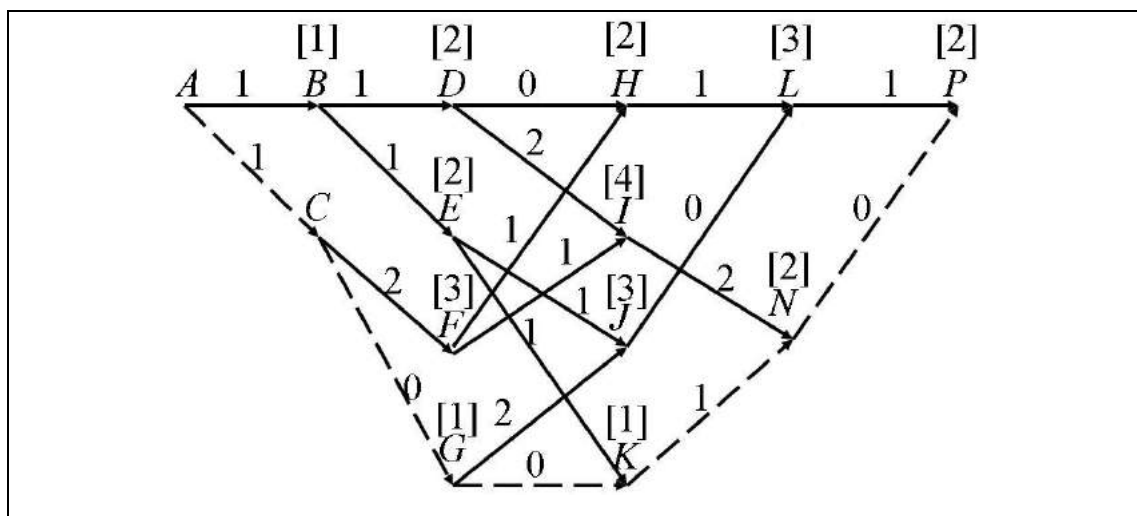


Fig. 7-20(b). Trellis of a Viterbi decoder for 1010 001010 received sequence.

We finally need to find the path from state A to P which gives the lowest overall Hamming distance. We then retrace the path and remember that the upper path from a state represented a 0 transmitted and the lower path represented a 1 transmitted.

7-3.4. Concatenated Codes

In 1974, Joseph **Odenwalder** combined the two coding techniques (**Block** and **Convolutional**) to form a **Concatenated Code**. In this arrangement, the encoder links an algebraic code followed by a convolutional code. The decoder, which is a mirror image of the encoder, consisted of a convolutional decoder followed by an algebraic decoder. Thus, any burst errors resulting from the convolutional decoder could be effectively corrected by the algebraic decoder. Performance was further enhanced by using an **interleaver** between the two encoding stages to mitigate any bursts noise that might be too long for the decoder to handle. This particular structure was introduced in 1993, by the

French scientist Claude **Berrou** and associates and was called the **turbo code**. Turbo coding is an iterative scheme that combines two or more **convolutional codes** and an interleaver to produce a **block code** that can perform to within a fraction of a decibel of the Shannon limit. More details about Turbo codes are covered in section (7-5) of this Chapter.

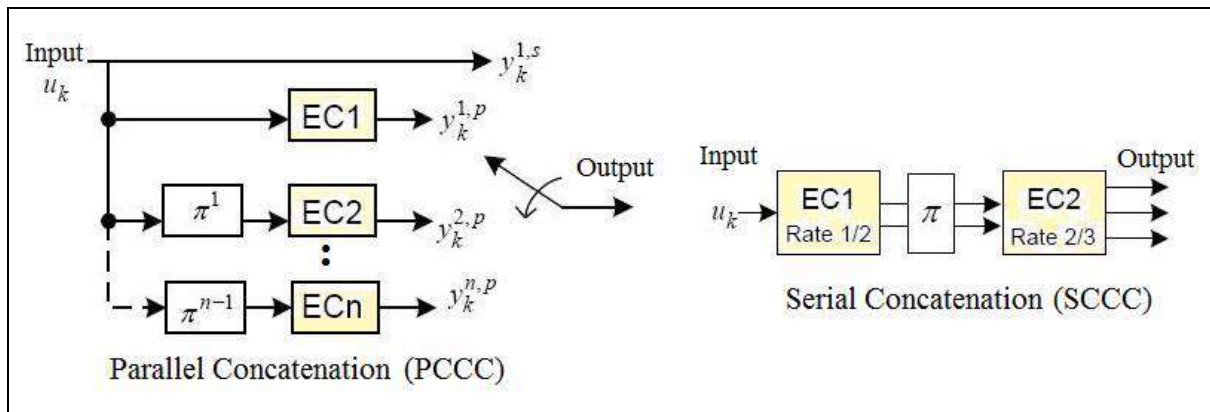


Figure 7-21. Illustration of the concatenated (parallel and serial) convolutional coders

7-3-5. List of Error-Detection & Correction Methods

The following figure and list contain the most famous practical methods and techniques of error detection and correction, which are currently employed in modern communication systems.

List of most important error detecting and correcting codes

- **Berger** code
- **Chipkill**, an application of ECC techniques to system memory.
- **Convolution** codes
- **Forward** error correction
- **Golay** code,
- **Goppa** code
- **Hadamard** code
- **Hamming** code
- **Raptor** codes are high speed (near real time) fountain codes.
- **Reed-Solomon** error correction
- **Reed-Muller** code
- **Tornado** codes (optimal Fountain codes)
- **Turbo** code
- **Viterbi** algorithm
- **Walsh** code (used in cellular telephony for high noise immunity)

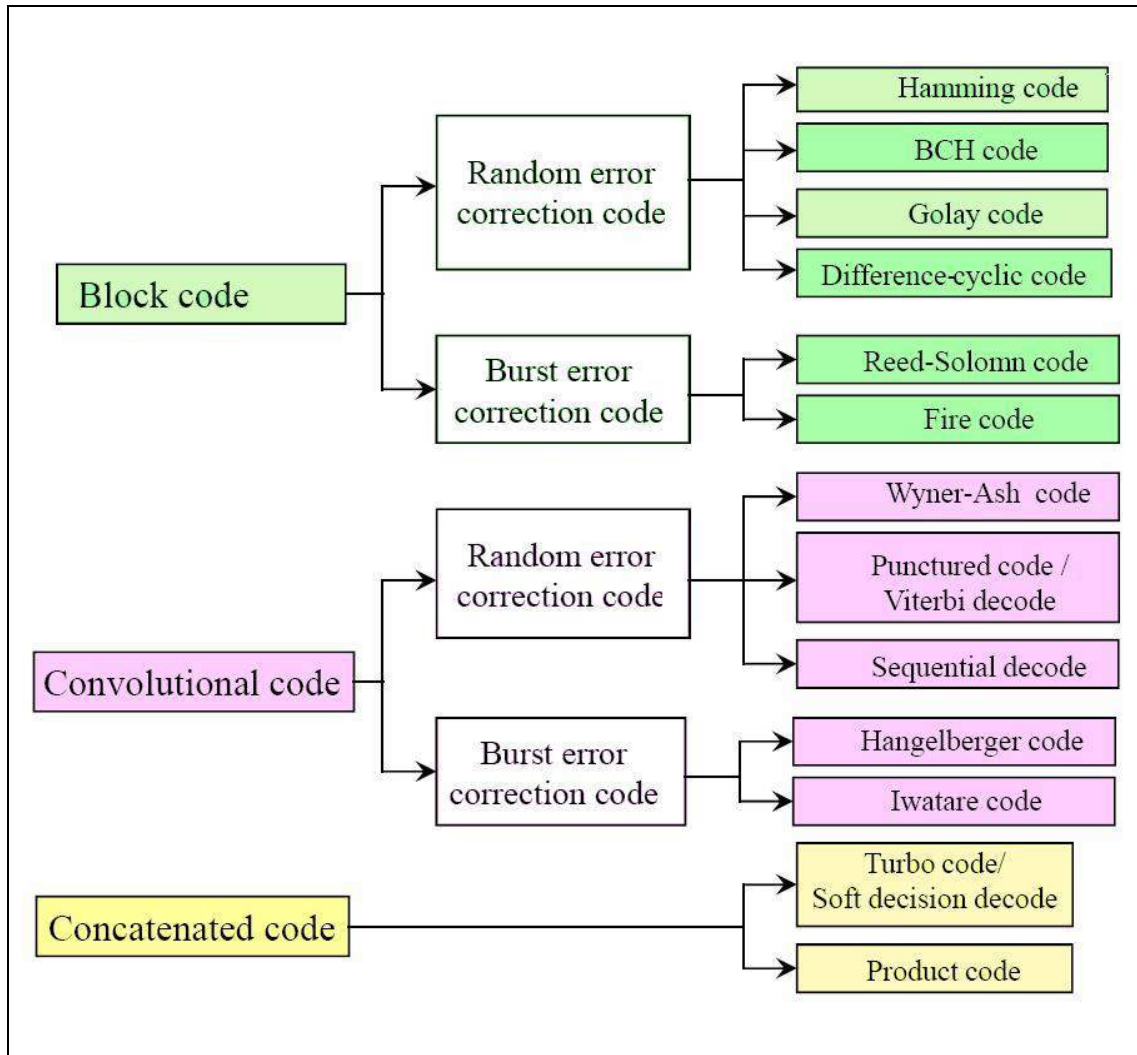
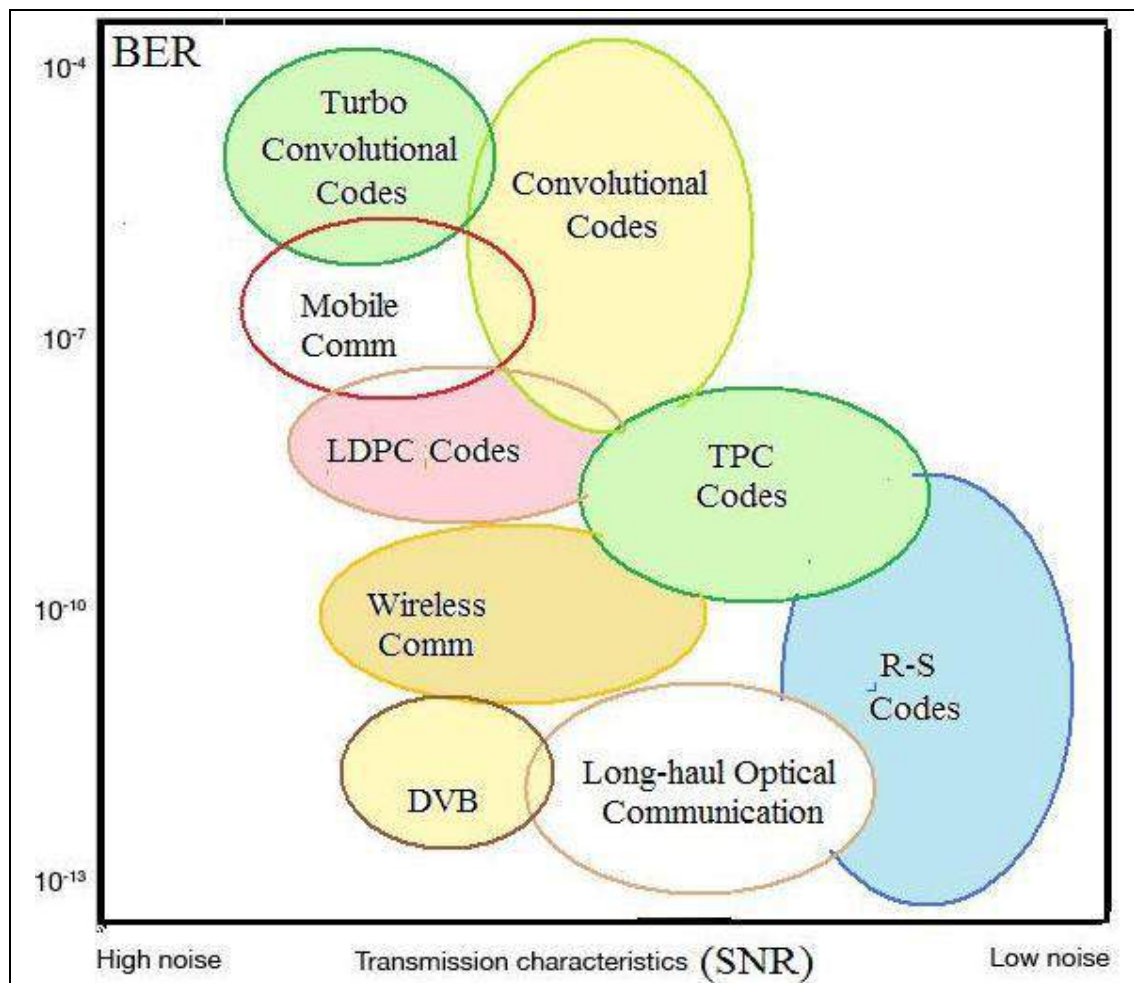


Figure 7-22. Types of error correcting codes

7-4. Forward Error-Correction (FEC)

Forward error-correction coding (also called **channel coding**) is a class of error correcting codes (ECC). FEC is actually a type of encoding that improves data reliability by introducing a known structure into a data sequence **prior** to transmission or storage. This coding structure enables a receiving system to detect and possibly correct errors caused by corruption from the channel and/or the receiver. As the name implies, this coding technique enables the decoder to correct errors without requesting retransmission of the original information.



or Figure 7-23. Forward error correction (FEC) techniques.

Like other ECC methods, there are two major types of FEC coding schemes: linear block codes and convolutional codes. In addition, some combinations of these two coding techniques (**Block** and **Convolutional**) can be made to form a **Concatenated Code**. Performance may be further enhanced by using an **interleaver** between the two encoding stages. This particular structure was introduced in 1993, by the French scientist Claude **Berrou** and associates and was called the **turbo code**.

There is another form of FEC codes, called **Turbo Product Code** or **TPC**. This form has a very different structure from the parallel and serial concatenated coders. TPC use block codes instead of convolutional codes. Two different block codes (usually Hamming codes) are concatenated serially without an interleaver in between. Since the two codes are independent and operate in rows and columns, this alone offers enough randomization that no interleaver is required.

Another alternative scheme is called the low-density parity code (**LDPC**). LDPC codes are actually a class of linear block codes. The name comes from the characteristic of their parity-check matrix which contains only a few 1's in comparison to the amount of 0's. Their main advantage is that they provide a performance which is very close to the capacity for a lot of different channels and linear time complex algorithms for decoding.

7-5. Turbo Codes

Turbo codes are a class of forward error correction (FEC) codes used in digital communications systems to recover lost or corrupted data in an information-bearing signal. Turbo codes are the most powerful forward error-correction codes yet. Using the turbo code, communication systems can approach the theoretical limit of channel capacity, as characterized by the so-called Shannon Limit, which had been considered unreachable for more than four decades. Turbo codes **DO NOT** work at the bit level. Rather, turbo codes typically work at the character or symbol level.

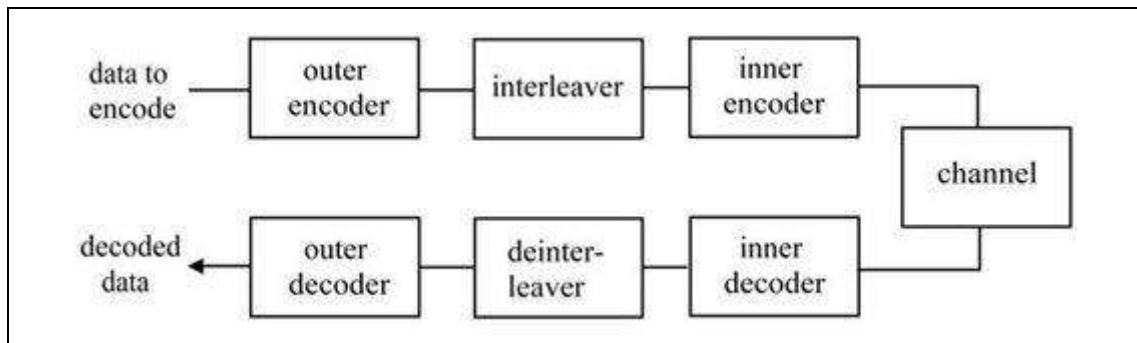


Figure 7-24. Turbo codec in a digital communication system.

The turbo encoder may be formed by the parallel or serial or even hybrid concatenation of two convolutional codes separated by an interleaver. An iterative process through two corresponding decoders is used to decode the data received from the channel. Each elementary decoder passes to the other soft (probabilistic) information about each bit of the sequence to decode. This soft information, called extrinsic information, is updated at each iteration. When we have two such codes, the signal produced is rate **1/3**. If there are

three encoders, then the rate is $\frac{1}{4}$ and so on. Usually two encoders are enough as increasing the number of encoders reduces bandwidth efficiency and does not buy proportionate increase in performance.

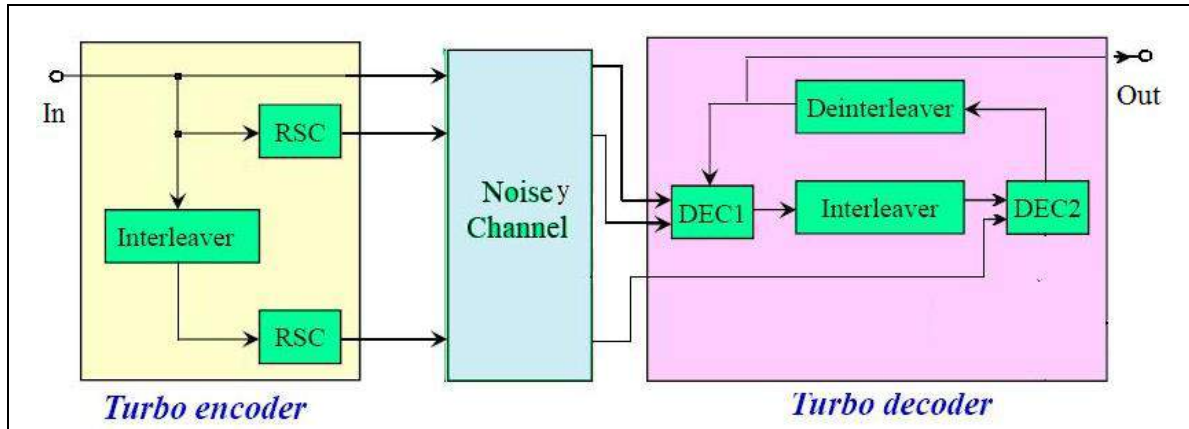


Figure 7-25. Block diagram of a Turbo Codec (Encoder/Decoder). Note that DEC means soft-in/soft-out decoder and RSC=Recursive Systematic Convolutional encoder

7-5.1. Turbo Encoders Structure

One of the requirements of a turbo coder-decoder (*codec*) configuration is that the encoder must include at least two encoders. Although each encoder may employ algebraic (block) coding or convolution coding, the overall encoder can be considered a block encoder because data are processed in blocks. The size of these blocks is dictated by the length of the interleaver that separates each component encoder. The following figure depicts a turbo codec. Typical block sizes are on the order of hundreds or thousands of bits. Bit times are indicated by subscripts. The figure represents a rate- $\frac{1}{3}$ turbo encoder (that is, one data bit produces a 3-bit codeword), but other rates are possible. The interleaver in a turbo encoder serves a different purpose than classic interleavers, in other parts of a communication system. Conventional interleavers **scramble** code bits among multiple blocks so that they are not contiguous when transmitted; as a result, any burst **errors** caused by channel corruption are **spread out**, into more-random errors after de-interleaving. The interleaver in a turbo encoder, is designed so that the second encoder gets an interleaved (scrambled) version of the same data block that went into the first encoder; thus, the second encoder generates an independent set of code bits. As shown in figure 7-26, the turbo encoder consists of two *recursive systematic convolutional* (**RSC**) encoders, C_1 and C_2 , which are connected in a parallel concatenation scheme. The RSC output can be obtained by the convolution of the input data sequence and the generator sequence of the encoder. For instance, in UMTS mobile phone systems, the generator sequences are: $G1(n) = (1011)$ and $G2(n) = (1101)$.

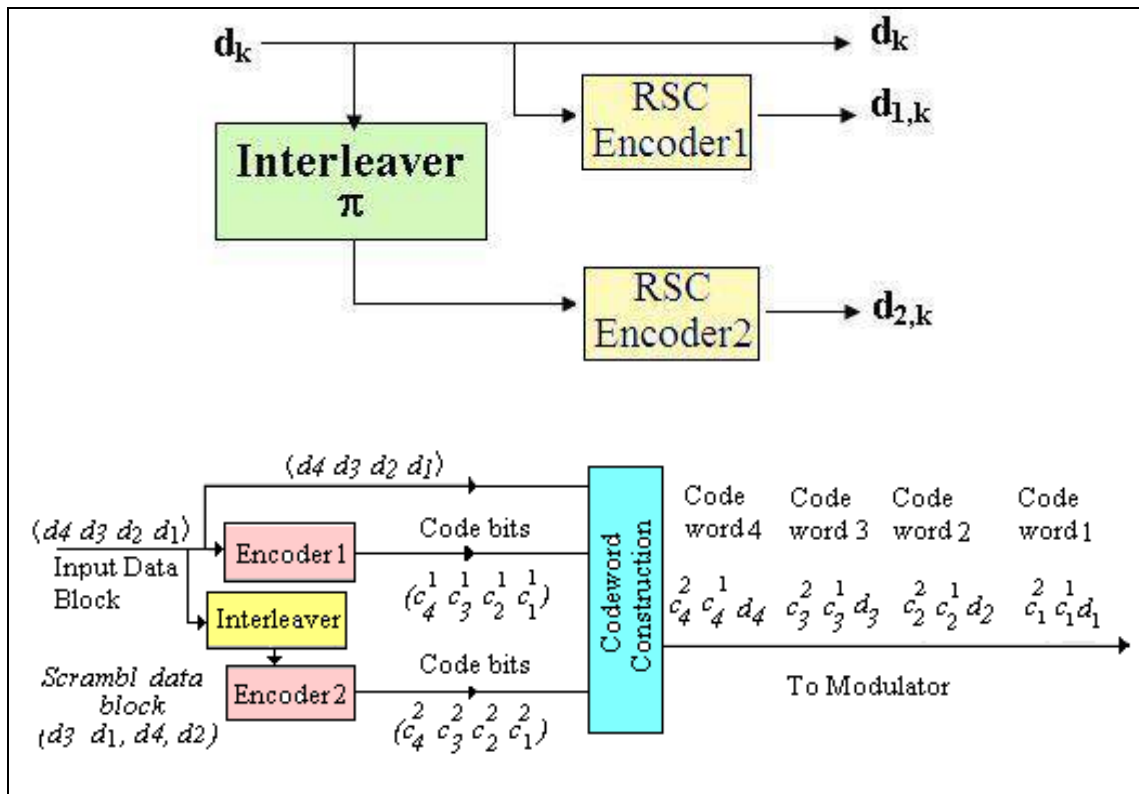


Fig. 7-26. Block diagram of a turbo encoder, with an example

7-5.2. Turbo Decoder Structure

The turbo codec must have as many decoders on the receiving end as encoders on the transmitting end. These decoders are concatenated in serial fashion and are joined by a series of interleavers and de-interleavers in a feedback loop, as shown in figure 7-27.

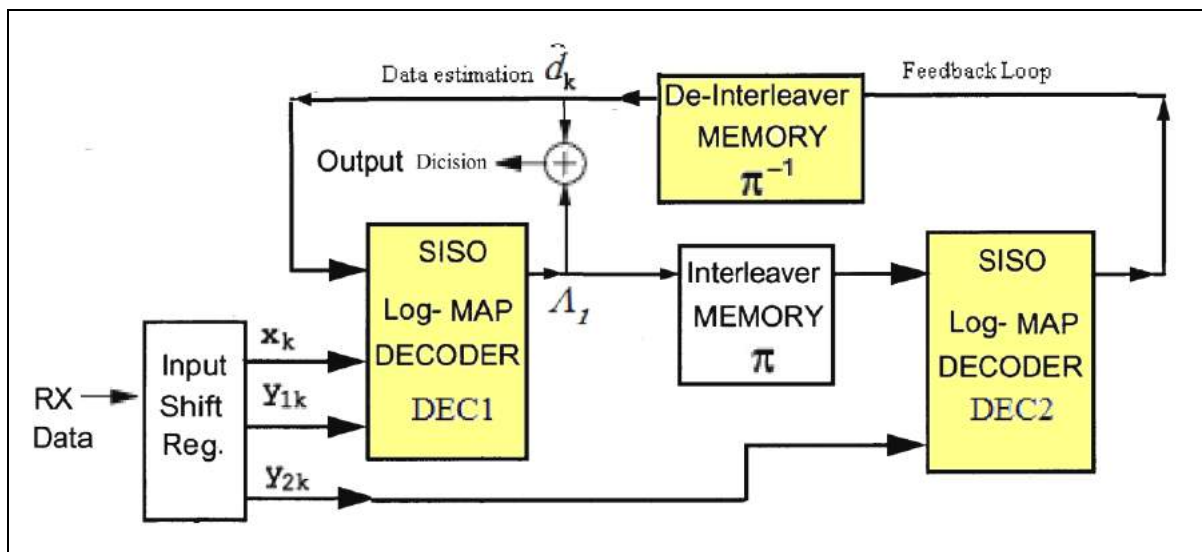


Figure 7-27 Block diagram of a turbo decoder

In a typical decoding operation, the first decoder generates statistical information based on the data received from the first component encoder. This information is then fed to the second decoder, which processes it along with the data received from the second component encoder. After decoding, the improved and updated statistical information is fed back to the first decoder, which starts the process again. This process typically continues for six to ten iterations (**loops**) for each block of data, at which point a switch is triggered and the actual data estimates are produced.

As shown in figure, the decoder is built in the similar way as the encoder, but in serial way, not parallel. The first decoder DEC_1 operates on lower speed, thus, it is intended for the first encoder RSC_1 , and second decoder DEC_2 is for RSC_2 correspondingly. DEC_1 yields a **soft decision** (Λ_i). The de-interleaver installed between the two decoders is used to **scatter errors** coming from DEC_1 output. The input *shift register* block acts as a demultiplexing and insertion module. It works as a switch, redirecting input bits to DEC_1 at one moment and to DEC_2 at another.

In order to understand, the operation of the turbo decoder, in the presence of noise, consider a memoryless AWGN channel and assume that at k^{th} iteration, the decoder receives a couple of random variables:

$$x_k = (2d_k - 1) + a_k \quad (7-3a)$$

$$y_k = 2(Y_k - 1) + b_k \quad (7-3b)$$

where a_k and b_k are independent noise components having the same variance. Here, y_k is a k^{th} bit from y_k encoder output. Redundant information is demultiplexed and sent to DEC_1 (y_{1k}) and to DEC_2 (y_{2k}). The DEC_1 yields a **soft decision**, (Λ_i) like this:

$$\Lambda(d_k) = \log \frac{p(d_k = 1)}{p(d_k = 0)} \quad (7-4)$$

and delivers it to DEC_2 . The result $\Lambda(d_k)$ is called the **log-likelihood ratio (LLR)**. Also, $p(d_k=i)$, $i = 0,1$ is **a posteriori probability (APP)** of the d_k data bit which shows the probability of interpreting a received d_k bit as i . After taking the **LLR** into account, it is fed to DEC_2 to get the decoded bit,

Note 7-3. Log Likelihood Ratio (LLR) and Maximum Likelihood (ML)

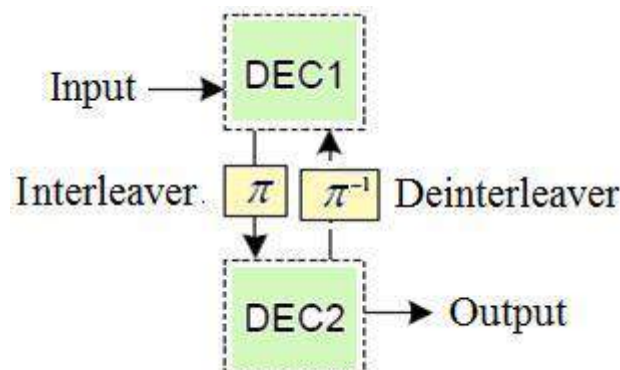
The Log-likelihood Ratio (LLR) of a binary variable (u_k) is defined as the natural log of its base probabilities.

$$\Lambda(d_k) = \log \frac{p(d_k = 1)}{p(d_k = 0)}$$

where u is the value of the binary variable at time k . Also, $p(d_k=i)$, $i = 0,1$ is **a posteriori probability (APP)** of the d_k data bit

Note 7-4. Maximum A-posteriori Probability (MAP) Algorithm

Turbo codes are usually decoded using a method called the **Maximum Likelihood Detection (MLD)**. Filtered signal is fed to the decoders, and the decoders work on the signal amplitude to output a soft “decision” The a priori probabilities of the input symbols is used, and a *soft* output indicating the *reliability* of the decision (amounting to a suggestion by decoder 1 to decoder 2) is calculated which is then iterated between the two decoders. There are two main types of soft-decoding algorithms, which are used in MLD. The first algorithm is a modified Viterbi algorithm but it produces soft outputs and hence called, **Soft Output Viterbi Algorithm (SOVA)**. The second algorithm is the **Maximum a-posteriori Probability (MAP)**. The direct implementation of MAP is computationally intensive and hence not feasible for real-time applications. In order to minimize the decoding complexity, the logarithms of the state metrics are taken (**Log-MAP** algorithm). In turbo codes, the MAP algorithm is used iteratively to improve performance. It is like a multiple questions game, where each previous guess helps to improve your knowledge of the hidden information. The number of iteration is often preset as in 20 questions. More iteration are done when the SNR is low, when SNR is high, lesser number of iterations are required since the results converge quickly. The algorithm is often pre-set with number of iterations. On the average, seven iterations give adequate results and no more 20 are ever required. The following figure depicts the Iterative decoding in MAP algorithm.



Note 7-5. Soft Output Viterbi Algorithm (SOVA)

Some recent turbo decoders are based on the so called Soft Output Viterbi Algorithm (SOVA). SOVA uses Viterbi decoding method but with **soft outputs** instead of hard output. SOVA maximizes the probability of the sequence, whereas the conventional **MAP** algorithm maximizes the bit probabilities at each time, even if that makes the sequence not-legal.

As we mentioned, so far in the last Note 7-3, the direct implementation of MAP is computationally intensive and hence not feasible for real-time applications. In order to minimize the decoding complexity, the logarithms of the state metrics are taken. This converts the multiplication operation to additions (Log-MAP algorithm). The problem with the Log-MAP algorithm is that now we have logarithms of sum of exponentials. This can be simplified using the **Jacobian** logarithm,

$$\log(e^{L_1} + e^{L_2}) = \max(L_1; L_2) + \log(1 + e^{L_1/L_2})$$

Most implementations compute the maximum term **$\max(L_1, L_2)$** and ignore the correction factor. Such methods are called **Max-Log-MAP** algorithms.

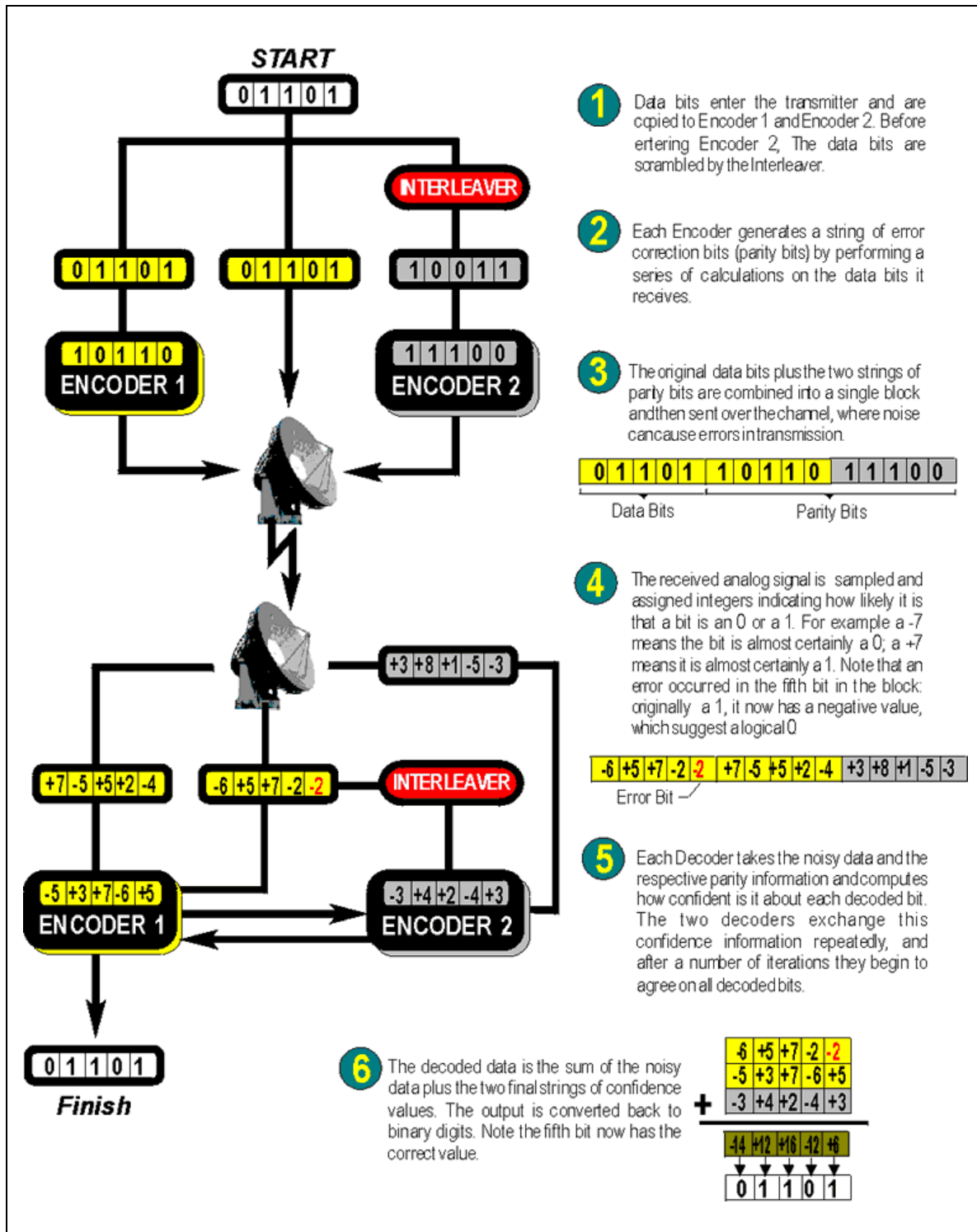
As shown in the following table, the Log-MAP is approximately 3 times more complex than SOVA while the Max-Log-MAP is about twice as complex as SOVA. However, the performance of MAP is about 0.5 dB better than SOVA at lower SNR and high BERs. This is very important for turbo codes since the output BERs from the first stage of iterative decoding is quite high and any improvement at this stage, leads to significant overall performance improvements. Another limitation of the MAP algorithm is the large memory requirement for the state metrics. Due to the limited on-chip memory in an embedded processor.

DECODER algorithms COMPLEXITY COMPARISON

Operation	MAP	Log-MAP	Max-Log-MAP	SOVA
add.	$2 \times 2^k \times 2^\nu + 6$	$6 \times 2^k \times 2^\nu + 6$	$4 \times 2^k \times 2^\nu + 8$	$2 \times 2^k \times 2^\nu + 9$
multipl.	$5 \times 2^k \times 2^\nu + 8$	$2^k \times 2^\nu + 6$	$2 \times 2^k \times 2^\nu$	$2^k \times 2^\nu$
max ops		$4 \times 2^\nu - 2$	$4 \times 2^\nu - 2$	$2 \times 2^\nu - 2$
table look-ups		$4 \times 2^\nu - 2$		
exp.	$2 \times 2^k \times 2^\nu$			

7-5.3. Illustration Example of a Turbo code

The following figure depicts the operation of a Turbo codec, with numerical example.



1 Data bits enter the transmitter and are copied to Encoder 1 and Encoder 2. Before entering Encoder 2, the data bits are scrambled by the Interleaver.

2 Each Encoder generates a string of error correction bits (parity bits) by performing a series of calculations on the data bits it receives.

3 The original data bits plus the two strings of parity bits are combined into a single block and then sent over the channel, where noise can cause errors in transmission.

4 The received analog signal is sampled and assigned integers indicating how likely it is that a bit is a 0 or a 1. For example a -7 means the bit is almost certainly a 0; a +7 means it is almost certainly a 1. Note that an error occurred in the fifth bit in the block: originally a 1, it now has a negative value, which suggests a logical 0.

5 Each Decoder takes the noisy data and the respective parity information and computes how confident it is about each decoded bit. The two decoders exchange this confidence information repeatedly, and after a number of iterations they begin to agree on all decoded bits.

6 The decoded data is the sum of the noisy data plus the two final strings of confidence values. The output is converted back to binary digits. Note the fifth bit now has the correct value.

Figure 7-28. Illustration example of a Turbo codec (encoder / decoder)/

7-5.4. Error Performance of Turbo Codes

Thanks to the iterative decoding, turbo codes can achieve a bit-error rate that approaches the Shannon limit. The following figure shows a comparison between different ECC codes and the Turbo code performance. These techniques are FEC techniques that are particularly suited to a channel in which the transmitted signal is corrupted mainly by additive white Gaussian noise (AWGN). You may think of AWGN as noise whose voltage distribution over time has characteristics that is described by a Gaussian, or normal, distribution (a bell curve). This noise voltage distribution has zero mean and a standard deviation that is a function of the signal-to-noise ratio (SNR) of the received signal. If we assume that the received signal level is fixed, then if the SNR is high, the standard deviation of the noise is small, and vice-versa.

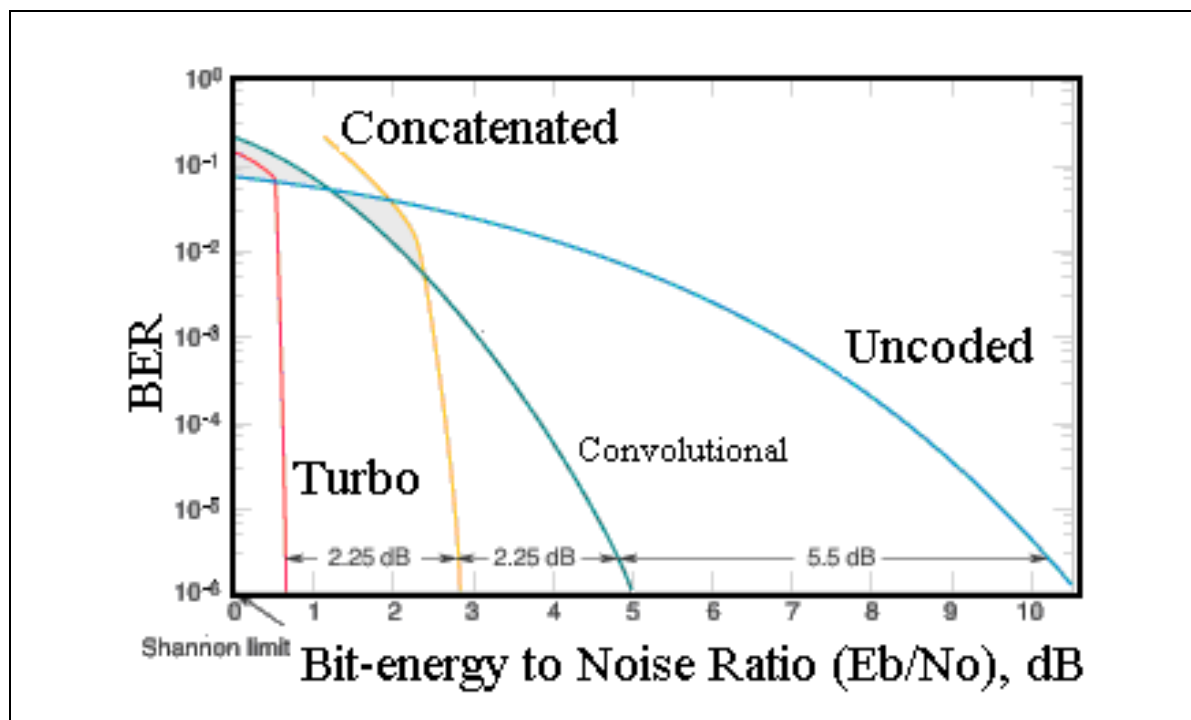


Fig. 7-29. The bit-error rate (BER) of a rate 1/2 turbo coded system after 10 iterations as compared with systems that use no coding, convolution, and concatenated coding.

For example, at a desired bit-error rate of 10^{-6} , convolutional codes can typically provide a 5.5 dB improvement (72% power savings) and concatenated codes a 7.75 dB improvement (83% power savings) over an uncoded system. Using a turbo code, an additional 2.25 dB improvement over the concatenated code can be attained, resulting in a total coding gain of 10 dB (90% power savings) compared with the uncoded system. For a rate $\frac{1}{2}$ turbo-coding system, the error performance comes within about 1 dB of the Shannon limit.

7-6-Applications of Turbo Codes

There exist so many applications of turbo codes in various digital communication systems. In fact, the tests conducted by the latest chipsets demonstrate that the performance achieved by using Turbo Codes may be even lower than the 0.8 dB figure assumed in early designs.

7-6-1. Digital Satellite TV & Video Broadcasting (DVB)

The demand for satellite transponder bandwidth continues to grow, fueled by the desire to deliver more TV channels and High Definition TV (HDTV) as well as IP data. Transponder availability and bandwidth constraints have limited this growth, because transponder capacity is determined by the selected modulation scheme and Forward error correction (FEC) rate. In satellite communications, **block 2D & 3D** bit allocation models are used by ECC coding systems. It worth notice that QPSK coupled with traditional Reed Solomon and **Viterbi codes** have been used for nearly 20 years for the delivery of digital satellite TV. Higher order modulation schemes such as 8PSK, 16QAM and 32QAM have enabled the satellite industry to increase transponder efficiency by several orders of magnitude.

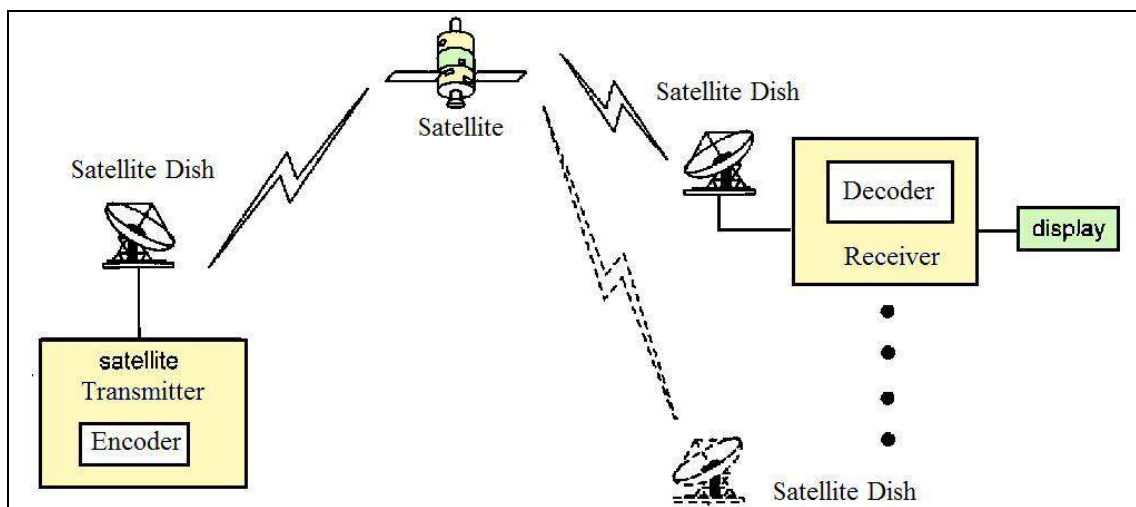


Figure 7-30. Application of Turbo code. In satellite communications

The satellite transmission is made up of a sequence of RF carrier pulses, called symbols. According to the modulation method each symbol represents 1, 2 or 3 or more bits of transmission rate data. For instance, in QPSK, each symbol contains two bits of data. For higher order PSK schemes more carrier to noise (C/N) ratio is required. In 17-QAM modulation the amplitude and the phase are changed from symbol to symbol, making a matrix pattern with the dots even closer together, and thus requiring even higher C/N ratio.

Forward error correction is applied to the customer's information data at the transmit end. Therefore,

$$\text{transmission data rate} = \text{customer information rate} \times 1/(\text{FEC rate}). \quad (7-5)$$

FEC rate is typically in the range 1/2 to 7/8, so the transmission data rate is always significantly more than the customer information rate. Ideally you want to use all of both the available bandwidth and power to obtain the highest user information rate. The following table depicts some important parameters for digital video broadcasting (DVB) by satellite communications

Table 7-5. Typical coding parameters of digital video broadcasting (DVB)

Modulation, FEC rate & FEC coding method	Minimum threshold Eb/No (BER=10E-8).	Bit rate (Rb) bit/s	Symbol rate. (Rs)	Bandwidth at -10 dB = 1.19 Rs	Bandwidth (carrier spacing) = 1.35 Rs
QPSK 1/2 rate FEC Viterbi	7.2 dB	1	1	1.19	1.35
QPSK 21/44 FEC Turbo	3.1 dB	1	1.048	1.246	1.414
QPSK 3/4 rate FEC Turbo	4.3 dB	1	0.667	0.793	0.9
QPSK 7/8 FEC Turbo	4.4 dB	1	0.571	0.68	0.77
8-PSK 3/4 rate FEC Turbo	6.7 dB	1	0.444	0.53	0.6
17-QAM 3/4 rate FEC Turbo	8.1 dB	1	0.333	0.397	0.536
17-QAM 7/8 rate FEC Turbo	8.2 dB	1	0.286	0.340	0.386

7-6-2. Data Storage

Error detection and correction codes are often used to improve the reliability of data storage media. A parity track was present on the first magnetic tape data storage. The Optimal Rectangular Code (**ORC**) used in group code recording tapes not only detects but also corrects single-bit errors. Some file formats, such as the **ZIP** (file format) include a checksum (most often **CRC32**) to detect corruption and truncation. Reed-Solomon codes are used in compact discs (**CD**) to correct errors caused by scratches. Modern hard drives (**HDD**) use **CRC** codes to detect and **Reed-Solomon** codes to correct minor

errors in sector reads, and to recover data from sectors that have gone bad and store that data in the spare sectors. Nowadays, the **RAID** storage systems use a variety of error correction methods, to correct errors when a hard drive completely fails.

7-6-3. Internet

In a typical TCP/IP stack, error detection is performed at multiple levels. Each Ethernet frame carries a CRC-32 checksum. The receiver discards frames if their checksums do not match. The IPv4 header contains a header checksum of the contents of the header (excluding the checksum field). Packets with checksums that don't match are discarded. The checksum was omitted from the IPv6 header, because most current link layer protocols have error detection. UDP has an optional checksum. Packets with wrong checksums are discarded. TCP has a checksum of the payload, TCP header (excluding the checksum field) and source- and destination addresses of the IP header. Packets found to have incorrect checksums are discarded and eventually get retransmitted when the sender receives a triple-ack or a timeout occurs.

7-6-4. Deep-space Telecommunications

The aerospace industry has been always using many different error correcting codes, for satellite and airspace shuttle communications. For instance, the NASA missions between 1969 and 1977, the Mariner spacecraft used a Reed-Muller code. The noise these spacecraft were subject to was well approximated by a bell-shape curve (normal distribution), so the **Reed-Muller** codes were well suited to the situation. Color image transmission required 3 times the amount of data, so the **Golay** (24, 12,8) code was used. The Golay code is only 3-error correcting, but it could be transmitted at a much higher data rate. Voyager-2 space shuttle went on to Uranus and Neptune and the code was switched to a concatenated **Reed-Solomon** convolution code for its error correcting capabilities.

7-7. Summary

The general idea for achieving error detection and correction is to add some redundancy (some extra data) to a message, which receivers can use to check consistency of the delivered message, and eventually correct corrupted data. So, error detection and correction techniques are based on the addition of a code to the signal at the transmitter side. A decoder in the receiver side can detect and eventually corrects errors, by making use of the properties of this code.

There are *two ways* to design the channel code and protocol for an error correcting system:

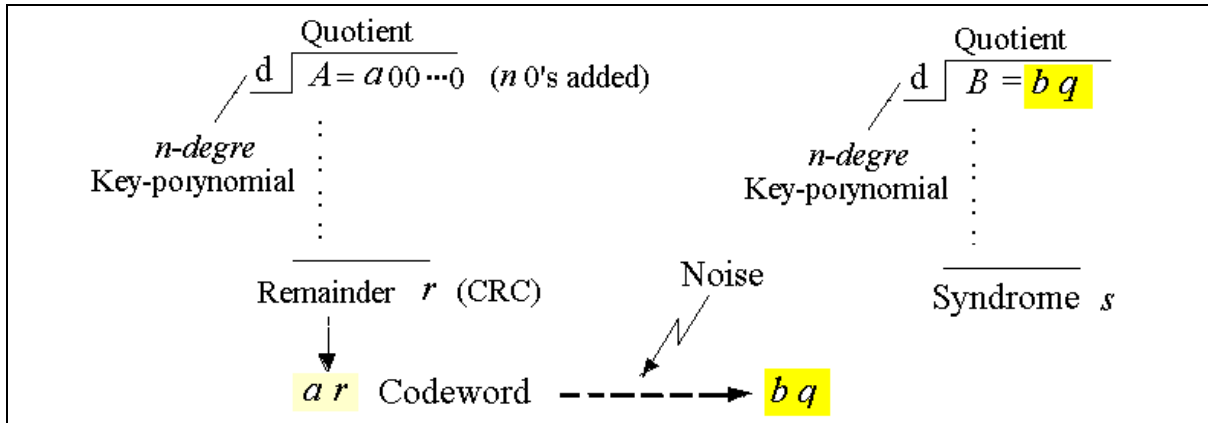
- **Automatic repeat request (ARQ)**
- **Forward error correction (FEC)**

Automatic Repeat-reQuest (**ARQ**) is an error control method for data transmission which makes use of error detection codes, acknowledgment and/or negative acknowledgement messages and timeouts to achieve reliable data transmission. An acknowledgment is a message sent by the receiver to the transmitter to indicate that it has correctly received a data frame. Usually, when the transmitter does not receive the acknowledgment before the timeout occurs (i.e. within a reasonable amount of time after sending the data frame), it retransmits the frame until it is either correctly received or the error persists beyond a predetermined number of retransmissions.

Cyclic redundancy check (CRC) is an error-detecting method. The cyclic redundancy check considers a **block of data** and divides it by a fixed, predetermined polynomial, called the CRC polynomial. The following five principles are used in calculating a CRC:

- (i) An entire packet of data is treated as one long binary number " b "
- (ii) The number is multiplied by 2^n by adding n zeros to the end of the number (resulting in the number " B "). The value of n depends upon the specification for the CRC calculation and is equal to the number of CRC digits to be transmitted. That's $n = 16$ for CRC16, and $n = 32$ for CRC32.
- (iii) The new number " B " is divided by another selected binary number " g ", which is referred to as a "**generator polynomial**". The value of " g " depends upon the specification for the CRC calculation, but it always contains one bit more than the number of CRC bits to be transmitted

The following figure shows the division process in CRC technique.



Hamming codes are error detecting and correcting codes, that can detect up to 2 errors and correct one error. The following general algorithm generates a single-error correcting Hamming code for any number of bits.

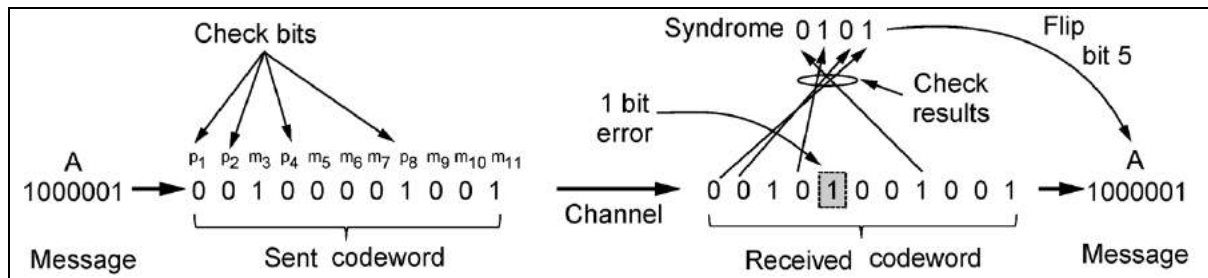
1. Number the bits starting from 1: bit 1, 2, 3, 4, 5, etc.
2. Write the bit numbers in binary: 1, 10, 11, 100, 101, etc.
3. All bit positions that are powers of two are parity bits: 1, 2, 4, 8, etc.
4. All other bit positions, with two or more 1 bits, are data bits.
5. Parity bit 1 covers all bit positions which have the **least significant bit set**: bit 1 (the parity bit itself), 3, 5, 7, 9, etc. So, **P1** = 3 XOR 5 XOR 7 XOR 9 XOR ...
6. Parity bit 2 covers all bit positions which have the **second least significant bit set**: bit 2 (the parity bit itself), 3, 6, 7, 10, 11, etc. So, **P2** = 3 XOR 6 XOR 7 XOR 10 XOR ...
7. Parity bit 4 covers all bit positions which have the **third least significant bit set**: bits 4–7, 12–15, 20–23, etc. So, **P4** = 4 XOR 5 XOR 6 XOR 7 XOR 12 XOR 13 XOR 14 XOR 15 XOR 20 XOR 21 XOR 22 XOR 23 XOR ...
8. Parity bit 8 covers all bit positions which have the **fourth least significant bit set**: bits 8–15, 24–31, 40–47, etc. So, **P8** = 8 XOR 9 XOR 10 XOR 11 XOR 12 XOR 13 XOR 14 XOR 15 XOR 24 XOR 25 XOR 26 XOR 27 XOR 28 XOR 29 XOR 30 XOR 31 XOR 40 XOR 41 XOR 42 XOR 43 XOR 44 XOR 45 XOR 46 XOR 47 XOR ...

The form of the parity is irrelevant. Even parity is simpler from the perspective of theoretical mathematics, but there is no difference in practice.

This general rule can be shown visually:

Bit position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Encoded data bits	p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11	p16	d12	d13	d14	d15
Parity bit coverage	p1	X		X		X		X		X		X		X		X		X		X
	p2		X	X		X	X		X	X		X	X		X		X	X		X
	p4				X	X	X	X				X	X	X	X					X
	p8								X	X	X	X	X	X	X					
	p16																X	X	X	X

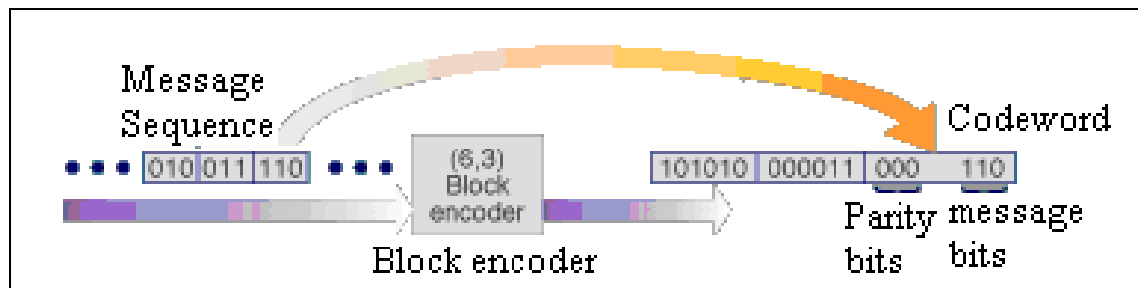
The following figure depicts the Hamming(11,7) code, with one bit correction



The **forward error correction (FEC)** is a technique used for controlling errors in data transmission over unreliable or noisy communication channels. There are two main categories of error correcting codes (ECC),:

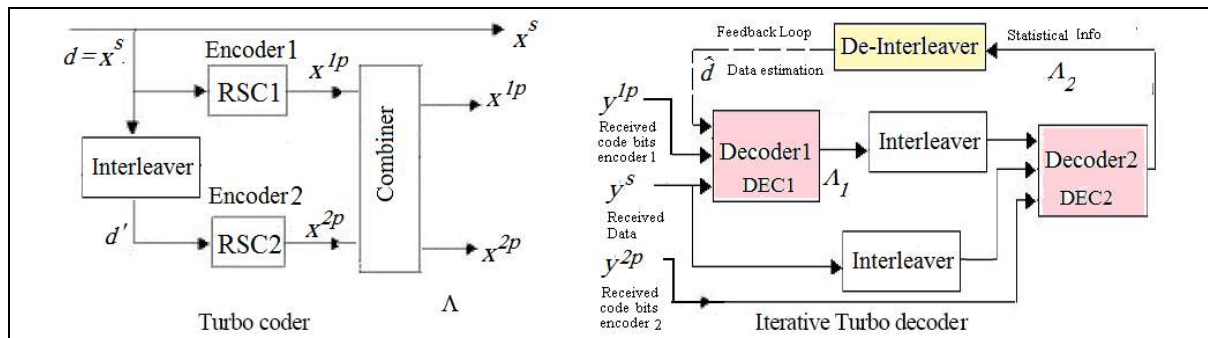
- 3- **Convolution Codes** and
- 4- **Algebraic (Block) Codes.**

Algebraic (or **Block**) codes, transform a chunk of bits into a (longer) chunk of bits in such a way that errors up to some threshold in each block can be detected and corrected. The following figure depicts an *encoding example* of a (6, 3) **algebraic encoder** that produces an output of 7-bit codeword for every 3-bit input message sequence. In this example, each 7-bit output codeword is composed of the original 3-bit message and a 3-bit parity sequence. This codeword format is known as *systematic*.



The idea of **convolutional** codes is to make every codeword symbol be the weighted sum of the various input message symbols. Convolutional codes are used in voiceband modems and in GSM mobile phones, as well as satellite and military communication devices.

Turbo codes are a class of error correction codes that were introduced in 1993 along with a practical decoding algorithm. The turbo codes enable reliable communications with efficiencies close to the theoretical limit predicted by Shannon's theorem. Since their introduction, turbo codes have been proposed for many applications such as deep-space and satellite communications, as well as cellular phones.



In conclusion, channel coding is a method to replace original data bits' with **codewords**, normally longer than the original bits. When we implement a digital communication or storage system, with error detection and correction, we have to ask ourselves the following questions. How many bits of redundancy bits I have to add to minimize the number of redundancy bits and maximize the error correction? Which Encoding/Decoding algorithms would be the best for the communication system I am implementing? In general answer, the more redundancy bits you add, you would have higher error detection / correction capability. However, the more redundancy bit you add, the less throughput you get and the larger bandwidth you need for transmission

7-8. Problems

7-1) Calculate the **CRC8** for original data bits, $b = 10001111$, if the generator polynomial g is:

$$x^8 + x^6 + x^5 + 1$$

If the received bits are 10010001, what will happen in the receiver checker side?

7-2) Describe the different types of ECC codes. Obtain the 15-bit Hamming code word for the 11-bit data word 11001001010.

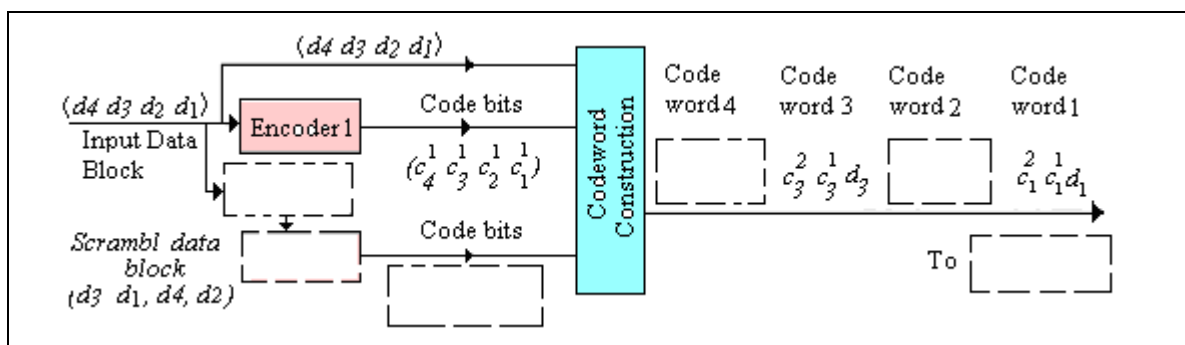
7-3) A 12-bit Hamming code word containing 8 bits of data and 4 parity bits is read from memory. What was the original 8-bit data word that was written into memory if the 12-bit word read out is as follows:

- (a) 000011101010
- (b) 101110000110
- (c) 101111110100

7-4) How many parity check bits must be included with the data word to achieve single-error correction and double-error detection when the data word contains

- (a) 16 bits.
- (b) 32 bits.
- (c) 48 bits.

7-5) Complete the following block diagram of a Turbo Code encoder



7-6) What's the Shannon limit of a channel information capacity? Can Turbo codes achieve a bit-error rate that approaches the Shannon limit? Explain your answer with graphs and supporting equations

7-9. References

[1] R. W. **Hamming**, Error Detecting and Error Correcting Codes. *Bell Syst. Tech. Journal*, Vol.29, pp. 147–160, **1950**.

[2] Vera **Pless**, Introduction to the Theory of Error-Correcting Codes, John Wiley & Sons, Inc., **1982**.

[3] Shu **Lin** and J. **Costello**, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, **1983**.

This is a comprehensive book on the basic theory of error coding. All common coding schemes from linear block codes to cyclic codes and convolutional codes. It is good book for reference and learning.

[4] F.J. **MacWilliams** and N.J.A. **Sloane**, The Theory of Error-Correcting Codes, North-Holland: New York, NY, **1977**.

[5] C. **Berrou**, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Code: Turbo Code," *Proceedings of the IEEE International Conference on Communications*, pp.1064–1070, Geneva, Switzerland, **1993**.

[6] C. C. **Wang**, "Mitigating the Error Floor for Turbo Codes," *Proceedings of Globecom*, November **1998**.

[7] Richard B. **Wells**, *Applied Coding and Information Theory for Engineers*. Upper Saddle River, NJ: Prentice-Hall, **1999**

[8] M. R. **Shane** and R. Wesel, "Parallel Concatenated Turbo Codes for Continuous Phase Modulation," *Proceedings of 2000 IEEE Wireless Communications and Networking Conference*, September **2000**.

[9] B. **Vucetic** and J. Yuan, *Turbo Codes*, Kluwer Academic Publishers, Boston, **2000**.

[10] C. C. **Wang** and D. Sklar, "A Novel Metric Transformation to Improve Performance of the Turbo Coded System with DPSK Modulation in Fading Environments," *Proceedings of Military Communications Conference*, October **2001**.

- [11] G. **Lui** and K. Tsai, "A Soft-Output Viterbi Algorithm Demodulation for Pre-coded Binary CPM Signal," *Proceedings of 20th AIAA International Satellite System Conference*, May **2002**.
- [12] D. **Sklar** and C. C. **Wang**, "On the Performance of High Rate Turbo Codes," *Proceedings of IEEE Aerospace Conference*, March **2002**.
- [13] S. **Lin** and D. J. Costello, *Error Control Coding*, 2nd Ed. Englewood Cliffs, NJ: Prentice-Hall, **2004**.
- [14] . Robert Morelos-**Zaragoza**, *The Art of Error Correcting Codes*, Wiley-Sons,. **2008**
- [15] I A **Glover** and P M **Grant**, "Digital Communications", Pearson Education, **2009**.

Chapter
8

Source Coding & Data Compression

Contents

- 8-1. Source Coding Theorem**
- 8-2. Data Compression**
- 8-3. Lossless (Data File) Compression**
 - 8-3.1. Lossless Compression Algorithms
 - 8-3.2. Run-Length Encoding (RLE) Algorithm
 - 8-3.3. Huffman Coding Algorithm
 - 8-3.4. LZ Algorithm
 - 8-3.5. ZIP File Format
- 8-4. Lossy (Image, Audio, Video Files) Compression**
 - 8-4.1. Lossy Compression Algorithms
 - 8-4.2. Audio Compression
 - MP3
 - AAC
 - 8-4.3. Image Compression
 - BMP & DIB Formats
 - JPG Format
 - GIF Format
 - PNG Format
 - 8-4.4. Video Compression
 - AVI Format
 - MPEG Format
 - 3GP Format
- 8-5. Summary**
- 8-6. Problems**
- 8-7. Bibliography**

Chapter

8

Source Coding & Data Compression

8-1. Source Coding Theorem

The aim of source coding is generally to take the source data and make it smaller. For this reason, source coding is sometimes referred to as **data compression**. In data networking, the reduction of transmitted file size corresponds directly to a reduction in the network bandwidth required to transmit the file. In this section we present the source coding theorem, from the information theory point of view.

As we have pointed out earlier, in Chapter 1, the **entropy** of a source is a measure of information content. Basically source codes try to reduce the **redundancy** present in the source, and represent the source with fewer bits that carry more information. Thus, the data compression techniques which explicitly try to minimize the average length of data according to a particular assumed probability model is called **entropy encoding**.

Various techniques used by source coding schemes try to achieve the limit of entropy of a source of alphabet.

$$C(x) \geq H(x), \quad (8-1)$$

where $H(x)$ is the entropy of the source (bit-rate), as defined in equation (1-17) in Chapter 1, and $C(x)$ is the bit-rate after compression. In particular, no source coding scheme can be better than the entropy of the source. The above equation may be also expressed as follows, in the general case where the information source is encoded into symbols of different lengths:

$$\underline{L} \geq H(x), \quad (8-2)$$

where \underline{L} is the average length of code words, i.e., the average number of bits per source symbol used in source encoding.

$$\underline{L} = \sum p_s l_s \quad (8-3)$$

with l_s are the individual lengths of all symbols (in bits) and p_s are the probabilities of these symbols. Therefore, the entropy represents here the fundamental limit (minimum) on the average number of bits per source symbol. The **efficiency of a source encoder**, η , is defined as follows:

$$\eta = L_{min}/\underline{L} = H(x)/\underline{L} \quad (8-4)$$

Note 8-1. What's Information Entropy?

Entropy is the measure of information content in a message. Messages with higher entropy carry more information than messages with lower entropy.

How to determine the entropy in an information message?. Find the probability $p(x)$ of symbol x in the message. The entropy $H(x)$ of the symbol x is:

$$H(x) = - p(x) \cdot \log_2 p(x)$$

The average entropy over the entire message is the sum of the entropy of all symbols in the message

8-2. Data Compression

Data compression is required for efficient data **transmission** or **storage**. Data compression is a sort of source coding which involves encoding information using fewer bits through use of specific encoding schemes. In fact, we sometimes consider the computer storage as channel in computer data communication. Therefore, data compression, which may be used to minimize storage size and increase the data transfer speed, are a sort of coding techniques. Examples include the *ZIP* file format (which acts as an archiver, storing many source files in a single file) and the *gzip* utility.

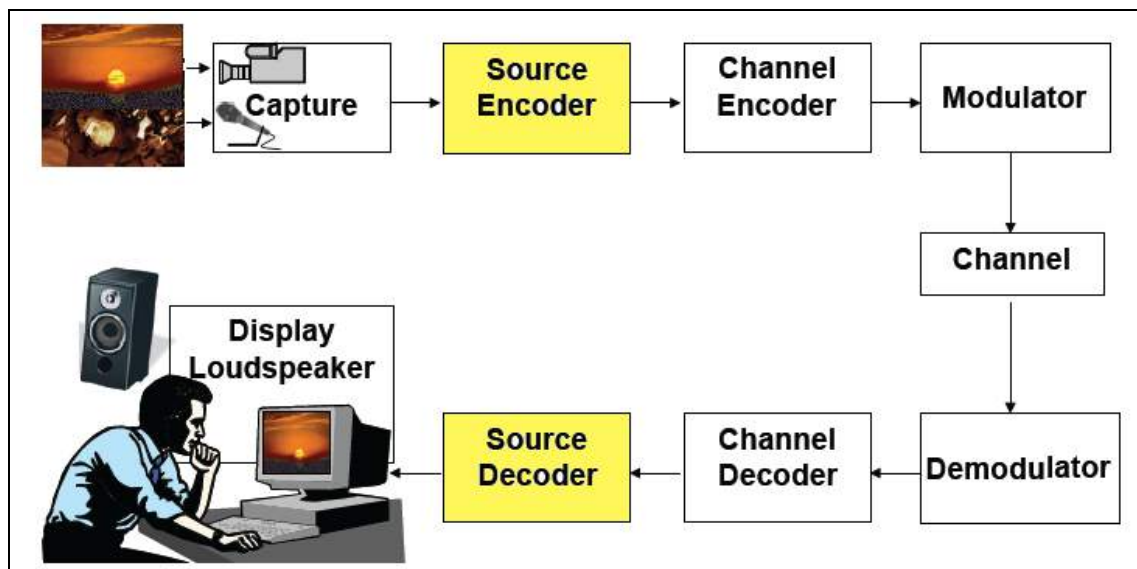


Figure 8-1. Source coding in a communication system

Compression is useful because it helps reduce the consumption of expensive resources, such as hard disk space or transmission bandwidth. The compression of data may be subdivided into two broad categories:

- 1- **Lossless data compression,**
- 2- **Lossy data compression.**

Lossless compression schemes are reversible so that the original data can be reconstructed, while lossy schemes accept some loss of data in order to achieve higher compression. For visual and audio data, some loss of quality can be tolerated without losing the essential nature of the data. By taking advantage of the limitations of the human sensory system, a great deal of space can be saved while producing an output which is nearly indistinguishable from the original. These lossy data compression

methods typically offer a three-way tradeoff between compression speed, compressed data size and quality loss.

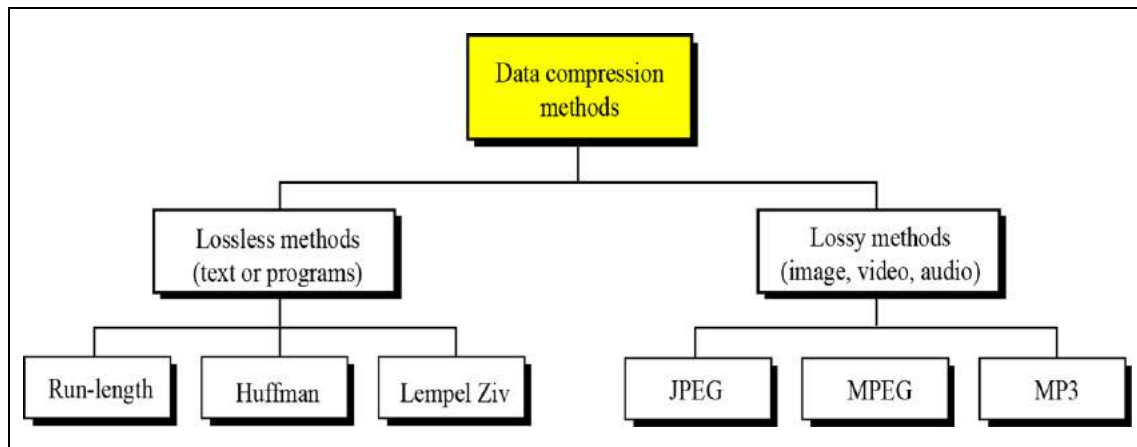


Figure 8-2. Source coding and data compression methods

Lossy image compression is used in digital cameras, to increase storage capacities with minimal degradation of picture quality. Similarly, DVDs use the lossy MPEG-2 codec for video compression.

The design of data compression schemes therefore involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (if using a lossy compression scheme), and the computational resources required to compress and uncompress the data.

8-3. Lossless Data Compression

Lossless compression algorithms usually exploit statistical redundancy in such a way as to represent the sender's data more concisely without error. Lossless compression is possible because most real-world data has statistical redundancy. For example, in English text, the letter 'e' is much more common than the letter 'z', and the probability that the letter 'q' will be followed by the letter 'z' is very small. However, lossless data compression algorithms will always fail to compress some files; indeed, any compression algorithm will necessarily fail to compress any data containing no discernible patterns. Attempts to compress data that has been compressed already will therefore usually result in an expansion, as will attempts to compress all but the most trivially encrypted data.

8-3.1. Lossless Compression Algorithms

There are several lossless compression algorithms, among them, one can cite:

- Run-Length Encoding (**RLE**),
- Entropy Encoding (e.g., **Huffman**, Shannon-Fano, Fibonacci codes)
- Dictionary Codes (e.g., LZ, LZW),
- Dynamic Markov Compression (**DMC**),

8-3.2. Run-Length Encoding

Run-length encoding (**RLE**) is a very simple form of data compression in which *runs* of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs: for example, simple graphic images such as icons, line drawings, and animations.

For instance, the string 25.88888888 can be compressed as: 25.[9]8. Interpreted as, "twenty five point 9 eights", the original string is perfectly recreated, just written in a smaller form. Another example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. Let us take a hypothetical single scan line, with B representing a black pixel and W representing white:

```

WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWW
WWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWW
WWWW

```

If we apply the run-length encoding (RLE) data compression algorithm to the above hypothetical scan line, we get the following:

12W1B12W3B24W1B14W

This is to be interpreted as twelve Ws, one B, twelve Ws, three Bs, etc. The run-length code represents the original 67 characters in only 18. Of course, the actual format used for the storage of images is binary rather than ASCII characters like this, but the principle remains the same

RLE is often used to optimize disk space on office computers, or better use the connection bandwidth in a computer network. For symbolic data such as spreadsheets, text, executable programs. RLE is also used in some formats like **PCX** and possibly in **BMP** and **TIFF**.

Run-Length Encoder in MATLAB

```
function []=runenc
name=uigetfile('*.*');
file_open=fopen(name,'r');
file_read=fread(file_open,'uint8');
fclose(file_open);
a=file_read;
a=reshape(a,[],1);
code=a;
count=1;k=1;i=1; % Run Length Encoder
while i~=length(code)
for j=i+1:length(code)
if code(i)==code(j)
count=count+1;
end
if code(i)~=code(j)
Run_Len(k)=code(i);
Run_Len(k+1)=count;
k=k+2;
count=1;
i=j;
break
end
end
if j==length(code);
Run_Len(k)=code(i);
Run_Len(k+1)=count;
k=k+2;
count=0;
i=j;
break
end
```



```

end
file=fopen('File.RUN','w');
fwrite(file,Run_Len,'ubit8');
fclose(file);

```

Calling code: **runenc**

Inputs: Data File , Output: Run-Length Encoded File (*.RUN)

The function reshapes the data file to a vector of symbols then it holds the symbol and counts the repetitions of that symbol after its position it takes another symbol when the different symbol is found.

Run-Length Decoder in MATLAB

```

function []=rundec
file_open=fopen('File.run','r');
file_read=fread(file_open,'ubit8');
fclose(file_open);
run_code=file_read;
data=[];
pos=1;k=1;j=1;
for i=1:length(run_code)
if mod(i,2)~=0
values(k)=run_code(i);
k=k+1;
end
if mod(i,2)==0
runs(j)=run_code(i);
j=j+1;
end
end
i=1;k=1;r=1;
while i~=sum(runs)+1
data(i)=values(k);
if runs(r)~=1
for j=0:runs(r)
data(i+j)=values(k);
end
i=i+runs(r); k=k+1;r=r+1;
else i=i+1;k=k+1;r=r+1;
end
end
file=fopen('file.txt','w');
fwrite(file,char(data),'ubit8');
fclose(file);

```

Calling code: **rundec**

Inputs: File.run , Output: the original data file (file.txt)

8-3.3. Huffman Coding Algorithm

The Huffman coding is an entropy encoding algorithm used for lossless data compression. The term refers to the use of a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol.

The algorithm basically *encodes* a string of **symbols** (e.g., strings of characters or bytes in a data file) into a prefix code (optimal Huffman code) of **codewords** (made up of symbols from an alphabet) with the minimum expected codeword length using a tree constructed from the **weights** of the symbols, to encode and to decode.

i. Creating a Huffman tree

The first step to the algorithm is constructing a Huffman tree. Symbols that appear more often will come "higher", that is, closer to the root node, in the tree, and will have shorter codewords; those that appear less often will come "lower" in the tree, and will have longer codewords.

Steps in constructing a binary Huffman tree:

1. Determine the probability (or frequency) of each symbol that occurs in the *source code*, the code to be encoded. One may start by counting or tallying them and placing the results in a table.
2. Sort the symbols by their probability (or frequency). (Sorting will finish the algorithm in linear time.)
3. If there are only two symbols (or root nodes of previously constructed trees, if any), construct a tree with a root node (you can assign the probability 1 if you like) with the symbols as nodes, the value of one path to one symbol being 0 and the other being 1.
4. Take the two symbols (or root nodes of previously constructed trees, if any) with the smallest frequencies or probabilities (weights) and construct a tree with the root node having a weight equal to the sum of the the weights of the symbols and the nodes as the symbols, the value of one path to one symbol being 0 and the other being 1.
5. Repeat step 3 until only one tree remains.
For example, in constructing the Huffman tree for the string "SEASHELLS":

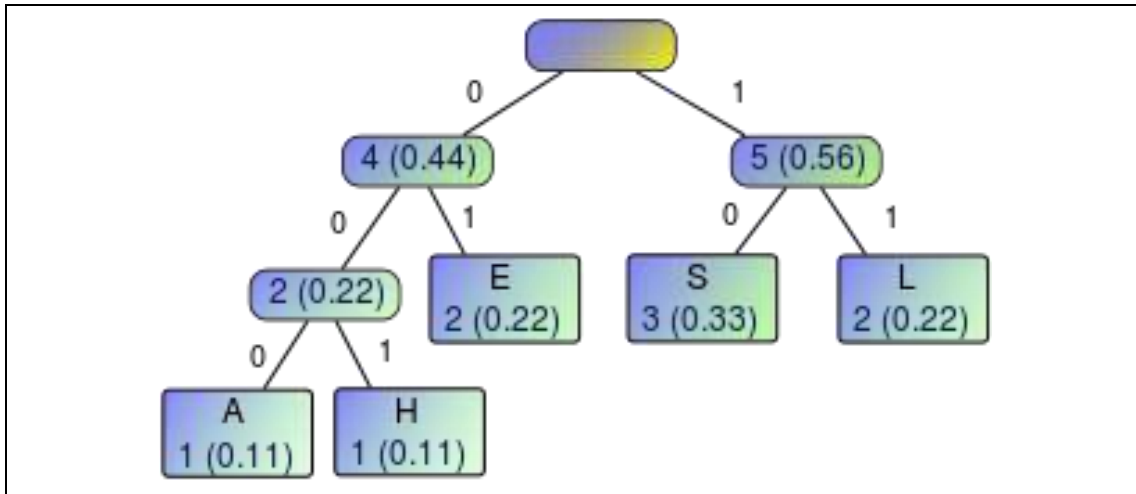


Figure 8-3(a). Example on Huffman coding

An optimal Huffman tree constructed from the string "SEASHELLS". The corresponding Huffman code encoded from this string with this tree is 100100010001011111.

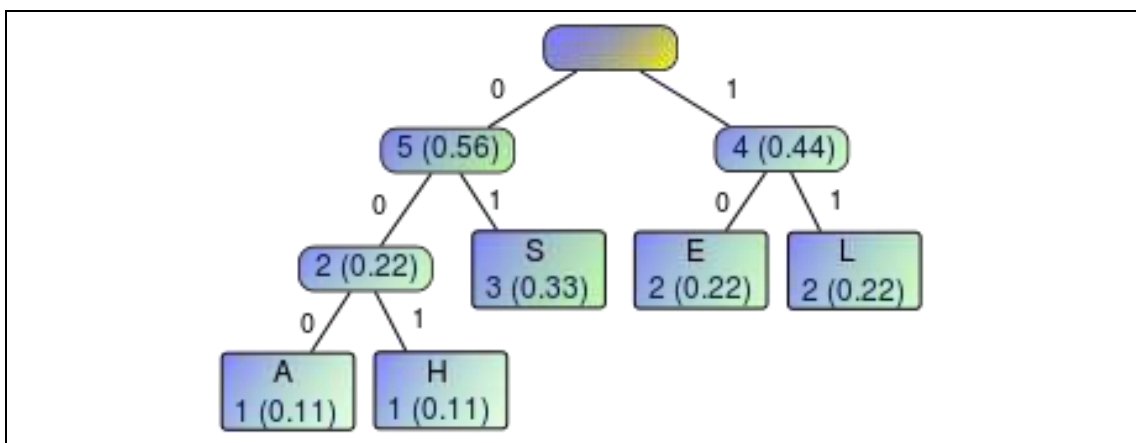


Figure 8-3(b) Example on Huffman coding

Another optimal Huffman tree constructed from the string "SEASHELLS". The corresponding Huffman code encoded from this string with this tree is 011000001001101111, which has the same length as the previous tree, 17 characters. The following table depicts the Frequency of symbols (each character is a symbol):

Symbols (sorted by appearance in string)	Frequency
S	3
E	2
A	1
H	1
L	2

- Constructing the tree, starting from A(1) and H(1). We will consider the left node as 0 and the right node as 1.
 - [A(1) , H(1)](2) - 2 being the sum of the weights of A and H.
- Taking E(2) and the previous tree. Note that we can also take L(2), but any tree constructed with this algorithm is optimal anyway (as optimal as any other tree with any chosen combination).
 - [A(1) , H(1)](2) , E(2)](4)
- Taking S(3) and L(2), since the previous tree already has a weight of 4, and these nodes have smaller weights than the tree.
 - [S(3) , L(2)](5)
- Finally, we have two nodes left.
 - [[A(1) , H(1)](2) , E(2)](4) , [S(3) , L(2)](5)]

ii. Encoding and Decoding

Using a Huffman tree (optimal or not), one may determine the codeword for a symbol (encoding) by looking for the symbol in the tree then following through the the tree from the root node to the node of the symbol, taking note of the path values taken. The resulting string of path values is the codeword. For example, to determine the code for "S" using the example tree in the previous section, starting from the root node, we go right (1), then left (0); the code is "10". We can just concatenate these codes since the tree is a prefix code. For example, the code for "SHE" is "10" + "001" + "01" = "1000101".

Likewise, one may determine the symbol for a string of codewords (decoding) by following through the tree, following the path determined by the character in the string of codewords until a symbol is encountered; and the act of following goes back to the root node in case more codewords are to be decoded. For example, to decode the string "001011111", we start with the root node, then left (0), left (0), then right (1). We encounter a symbol ("H"), so we go back to the root node. The string decodes to "HELL".

Huffman Encoder in MATLAB

```
function []=huffenc(type)
name=uigetfile('*.*');
if strcmp(type,'txt')==1
file_open=fopen(name,'r');
file_read=fread(file_open,'uint8');
fclose(file_open);
a=file_read;
end
```

```

if strcmp(type,'wav')==1;
a=wavread(name);
end
symbols=unique(a);
freq=histc(a,unique(a));
p=freq./sum(freq);
dict = huffmandict(symbols,p); % Create the dictionary.
hcode = huffmanenco(a,dict); % Encode the data.
file_Huff=fopen('File.HUFF','w');
fwrite(file_Huff,hcode,'ubit1');
fclose(file_Huff);
if strcmp(type,'wav')==1;
file_Huff=fopen('DictS.HUFF','w');
fwrite(file_Huff,symbols,'double');
fclose(file_Huff);
end
if strcmp(type,'txt')==1;
file_Huff=fopen('DictS.HUFF','w');
fwrite(file_Huff,symbols,'ubit8');
fclose(file_Huff);
end
power=1+ceil(log2(max(freq)));
file_Huff=fopen('DictF.HUFF','w');
command=['ubit' int2str(power)];
fwrite(file_Huff,freq,command);
fclose(file_Huff);
powerL=1+ceil(log2(length(hcode)));
commandL=['ubit' int2str(powerL)];
Len=length(hcode);
file_Huff=fopen('Len.HUFF','w');
fwrite(file_Huff,Len,commandL);
fclose(file_Huff);
sym=length(symbols);
key=[power,sym,powerL];
file_power=fopen('Key.HUFF','w');
fwrite(file_power,key,'ubit8');
fclose(file_power);
AVL=0;
for i=1:length(p)
AVL=AVL+p(i)*length(dict{i,2});
end
message=['AVL : ' num2str(AVL) 'bits/symbol'];
disp(message)

```

Calling code: **huffenc()**

Inputs: data file, type of file

Output: Huffman compressed file with header file contains the source symbols frequencies and length of data,

Huffman Decoder in MATLAB

```

function []=huffdec(type)
file_open=fopen('Key.HUFF','r');
file_read=fread(file_open,'ubit8');
fclose(file_open);
[key]=file_read;
power=key(1);
sym=key(2);
powerL=key(3);
if strcmp(type,'wav')==1;
file_open=fopen('DictS.HUFF','r');
file_read=fread(file_open,'ubit8');
fclose(file_open);
symbols(1:sym)=file_read(1:sym);
end
if strcmp(type,'wav')==1;
file_open=fopen('DictS.HUFF','r');
file_read=fread(file_open,'double');
fclose(file_open);
symbols(1:sym)=file_read(1:sym);
end
file_open=fopen('DictF.HUFF','r');
command=['ubit' int2str(power)];
file_read=fread(file_open,command);
fclose(file_open);
freq(1:sym)=file_read(1:sym);
file_open=fopen('Len.HUFF','r');
commandL=['ubit' int2str(powerL)];
file_read=fread(file_open,commandL);
fclose(file_open);
Len=file_read;
file_open=fopen('File.HUFF','r');
file_read=fread(file_open,'ubit1');
fclose(file_open);
hcode(1:Len)=file_read(1:Len);
p(1:sym)=freq(1:sym)/sum(freq);
dict = huffmandict(symbols,p); % Create the dictionary.
dhsig = huffmandeco(hcode,dict); % Decode the code.
if strcmp(type,'wav')==1;
wavwrite(dhsig,44100,'a.wav');
end
if strcmp(type,'txt')==1
file=fopen('file.txt','w');
fwrite(file,char(dhsig),'ubit8');
fclose(file);
end

```

Calling code: **huffdnc()** Inputs: huffman encoded file (*.huff),header files Output: data File

iii. Disadvantages of Huffman Code

If the **ensemble changes** the frequencies and probabilities change the optimal coding. For example in text compression symbol frequencies vary with context. Re-computing the Huffman code by running through the entire file in advance? Saving or transmitting the code too?

The second problem of Huffman code is that it **does not consider blocks of symbols**. For example, in strings_of_ch' the next nine symbols are predictable 'aracters_', but bits are used without conveying any new information.

8-3.4. LZ, and LZW Algorithms

The Lempel-Ziv (**LZ**) compression method is one of the popular **dictionary coding** algorithms. LZ and variant methods utilize a table-based compression model where table entries are substituted for repeated strings of data. For most LZ methods, this table is generated dynamically from earlier data in the input. The table itself is often Huffman encoded. Abraham Lempel and Jacob Ziv published two compression algorithms: LZ77 in 1977 and LZ78 in 1978.

In 1984, Terry Welch published the **LZW** algorithm as an improvement of LZ78. **LZW** (Lempel-Ziv-Welch) is used in GIF images. LZW is just one of the original LZ algorithms' many derivatives, the more famous ones being LZSS (used in RAR), and **Deflate** (used in ZIP). Also noteworthy is the **LZX** coding scheme that is used in Microsoft's **CAB** format.

i, LZ Algorithm

LZ77 and LZ78 are the two lossless data compression algorithms. These two algorithms are basis of LZ and variant algorithms. The basic idea is that repetitions are common in meaningful information and that space can be saved by replacing occurrences of information that was already seen by a reference to the first occurrence. The references are coded as (offset, length) pairs (tuples). For example, the text 'abracadabra' can be compressed as 'abracad(7,4)'. The 7 refers to the substring 7 position behind, and the 4 defines the repetition length.

Encoding

The basic idea of LZ encoding can be summarized as follows.

1. Initialize the dictionary to contain all blocks of length one ($D=\{a,b\}$).
2. Search for the longest block W which has appeared in the dictionary.
3. Encode W by its index in the dictionary.
4. Add W followed by the first symbol of the next block to the dictionary.
5. Go to Step 2.

The following figure shows a flowchart of the LZ compression and decompression algorithm. The subsequent two figures explain the encoding and decoding processes by an example.

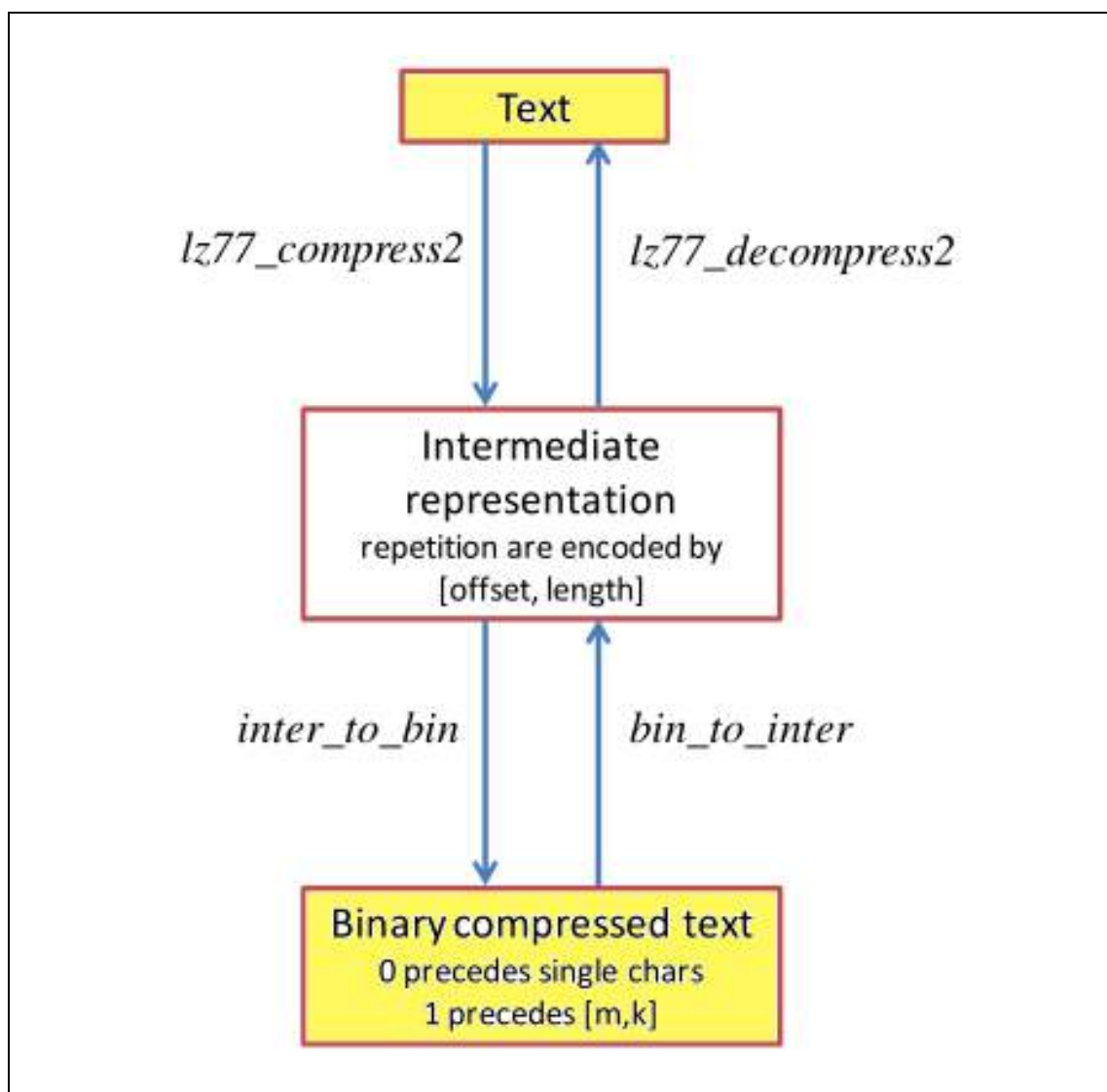


Figure 8-4 Flowchart of the LZ compression and decompression algorithm

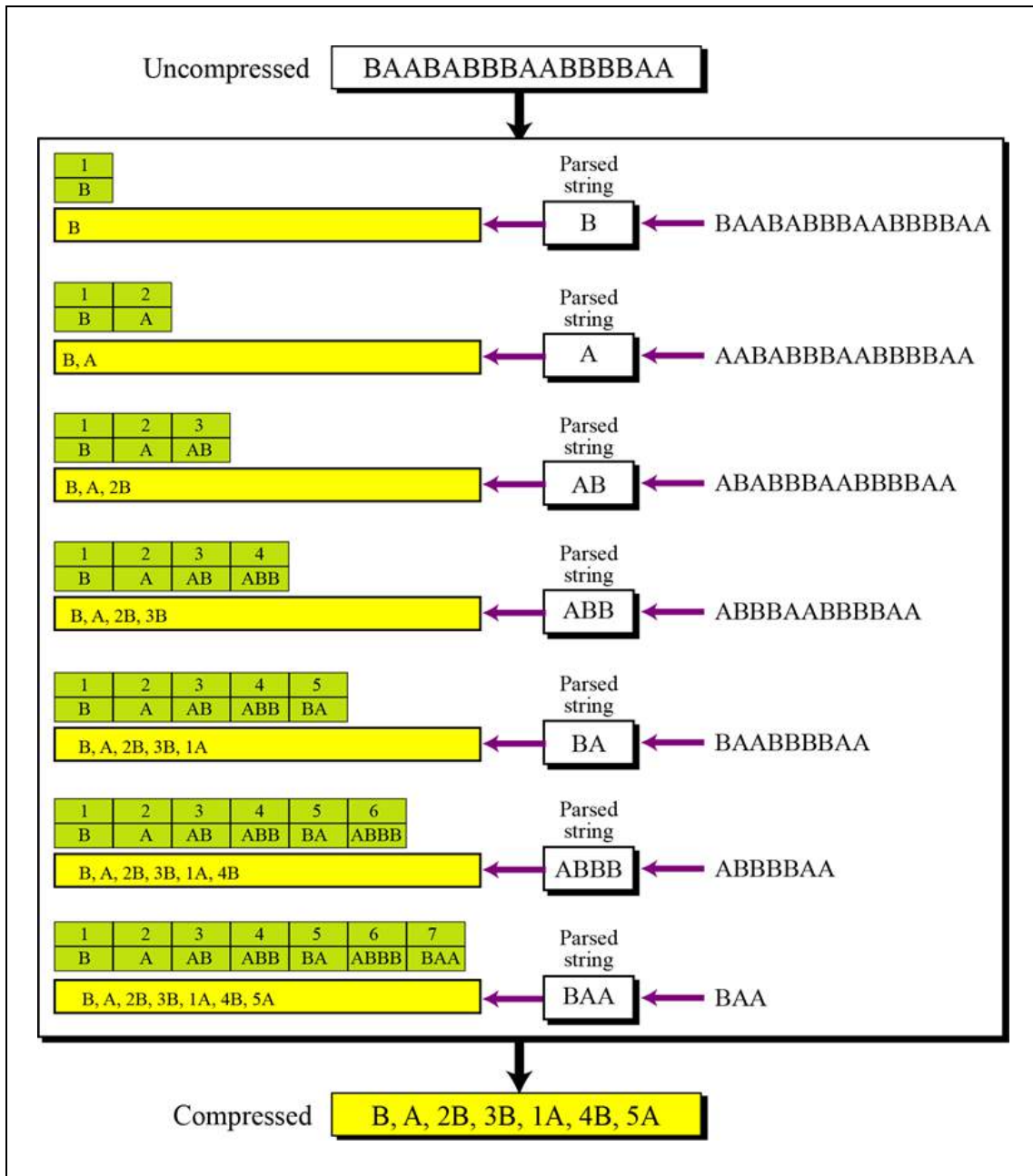


Figure 8-5(a) Example of LZ compression algorithm

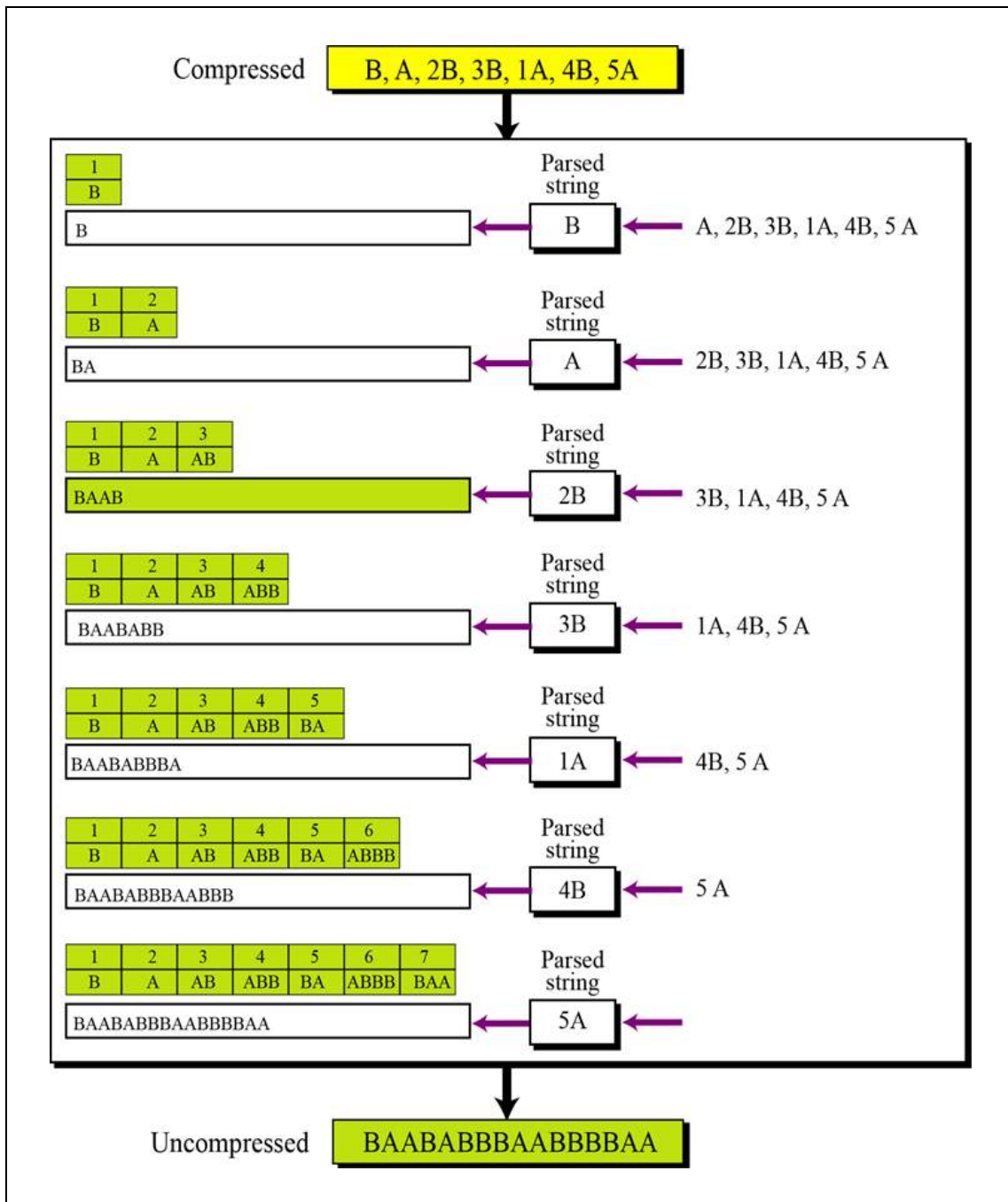


Figure 8-5(b) Example LZ decompression algorithm

ii, LZW Algorithm

The **LZW** is a universal lossless data compression algorithm created by Abraham Lempel, Jacob Ziv, and Terry Welch. It was published by Welch in 1984 as an improved implementation of the LZ algorithm.

Encoding

The LZW encoding algorithm may be summarized as follows:

1. Initialize the dictionary to contain all strings of length one.
2. Find the longest string *W* in the dictionary that matches the current input.
3. Emit the dictionary index for *W* to output and remove *W* from the input.
4. Add *W* followed by the next symbol in the input to the dictionary.
5. Go to Step 2

In the beginning, a dictionary is initialized to contain the single-character strings corresponding to all the possible input characters (except the clear and stop codes). The algorithm proceeds by scanning through the input string for successively longer substrings until it finds one that is not in the dictionary. When such a string is found, the index for the string without the last character (i.e., the longest substring that *is* in the dictionary) is retrieved from the dictionary and sent to output, and the new string (including the last character) is added to the dictionary with the next available code. The last input character is then used as the next starting point to scan for substrings.

Decoding

The LZW decoding algorithm may be summarized as follows. The decoding algorithm starts by reading a value from the encoded input and outputting the corresponding string from the initialized dictionary. In order to rebuild the dictionary in the same way as it was built during encoding, it also obtains the next value from the input and adds to the dictionary the concatenation of the current string and the first character of the string obtained by decoding the next input value, or the first character of the string just output if the next value cannot be decoded (If the next value is unknown to the decoder, then it must be the value that will be added to the dictionary this iteration, and so its first character must be the same as the first character of the current string being sent to decoded output). The decoder then proceeds to the next input value (which was already read in as the "next value" in the previous pass) and repeats the process until there is no more input, at which point the final input value is decoded without any more additions to the dictionary. In this way the decoder builds up a dictionary which is identical to that used by the encoder, and uses it to decode subsequent input values. Thus the full dictionary does not need be sent with the encoded data; just the initial dictionary containing the single-character strings is sufficient.

8-3.5. Deflate Algorithm

The so-called **Deflate** algorithm is a variation on LZ which is optimized for decompression speed and compression ratio. Deflate is used in **ZIP**, **PKZIP**, **GZIP**, **PNG** (portable network graphics).

The following figure depicts a high level overview of the major blocks inside the Deflate compression and decompression. Raw data enters the LZ encoder where string matching is performed using hash algorithms, hash tables, and a history buffer. The output of the LZ encoder is a queue, called the LLD (literals and length/distance) queue, which stores the symbols generated during the LZ encoding stage. The output of the LLD queue is processed by the Huffman Encoder which results in a standards compliant compressed bit stream.

The process is somewhat less complex in the decompression path. Compressed data is decoded in the Huffman Decoder to construct a stream of symbols required by the LZ decoder. The LZ decoder operates directly off of this symbol stream to reconstruct the original data, and does not require hashing or hash tables.

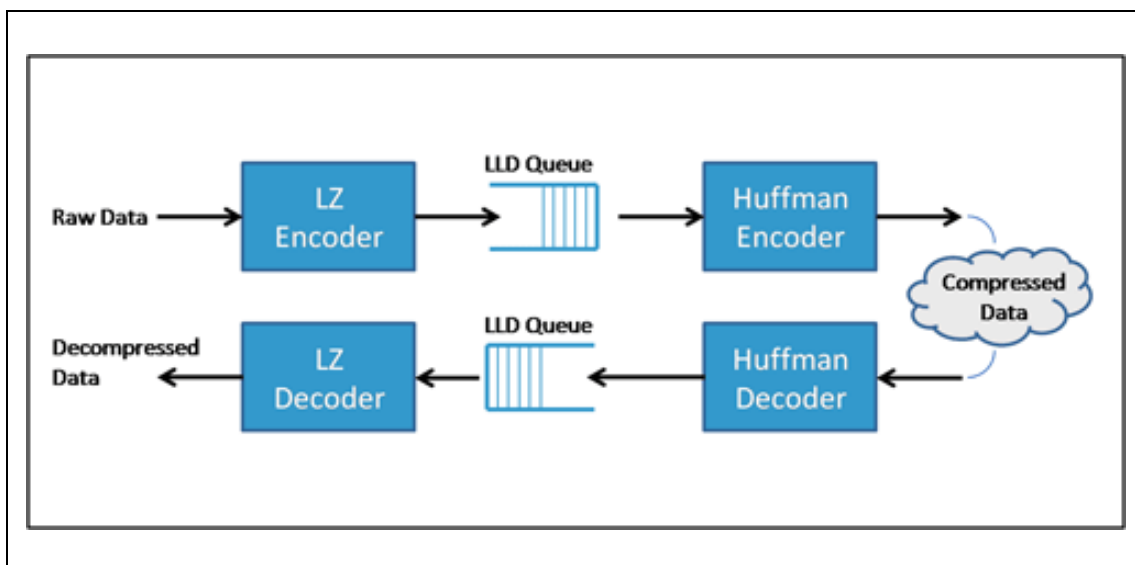


Figure 8-6 Deflate compression algorithm

8-3.6. ZIP File Format

A ZIP file contains one or more files that have been compressed to reduce file size, or stored as-is. The ZIP format was originally developed by Phil Katz for PKZIP from the previous ARC compression format. Katz publicly released technical documentation on the ZIP file format making it an open format, in January 1989.

The ZIP file format permits a number of compression algorithms, but as of 2008, only Deflate is widely used and supported. Many software utilities other than PKZIP are now available to create, modify, or open ZIP files, notably WinZip, Zip Genius (GZIP) and WinRAR. Microsoft has included built-in ZIP support (under the name "compressed folders") in later versions of its Windows operating system. Apple also has included built-in ZIP support in **Mac OS 10.X** operating system. Similar ZIP programs are also supported on UNIX operation systems.

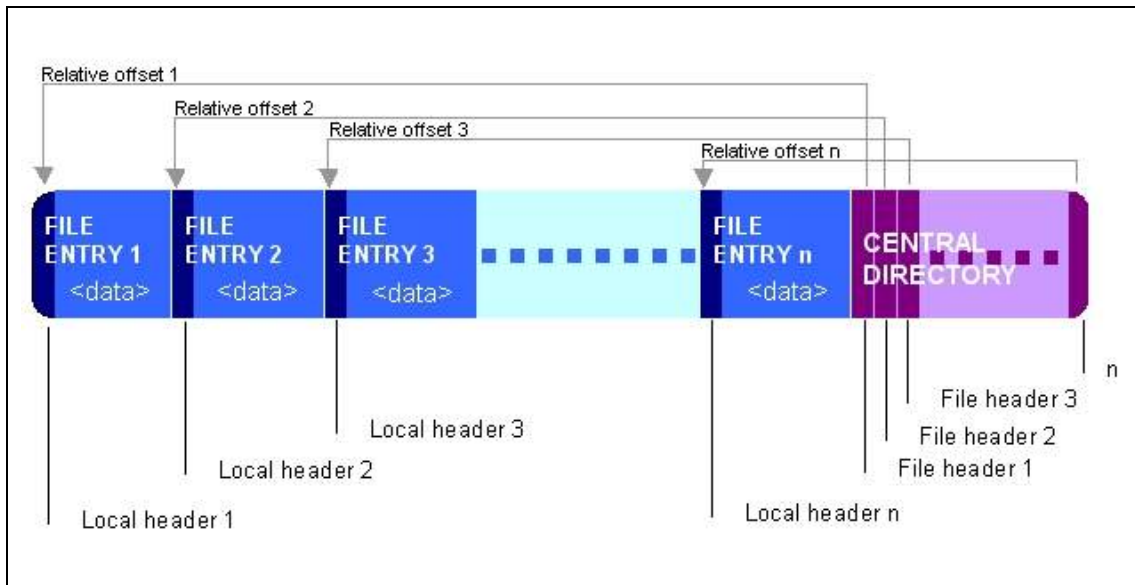









Figure 8-7 Zip file format

8-4. Lossy Data Compression

The other kind of compression, called **lossy data compression** or perceptual coding, is possible if some loss of fidelity is acceptable. Lossy compression techniques are often used in audio and video decoders where human perception of potential data loss is leveraged to improve the compression efficiency. Popular compression algorithms such as MPEG, JPEG, and MP3 are examples of lossy compression. For instance, JPEG image compression works in part by "rounding off" some of the less-important information. Lossy data compression provides a way to obtain the best fidelity for a given amount of compression. In some cases, transparent (unnoticeable) compression is desired; in other cases, fidelity is sacrificed to reduce the amount of data as much as possible.

8-4.1. Lossy Compression Algorithms

Many methods of lossy compression have been developed; however, a family of techniques called *transform compression* has proven the most valuable. The following methods are the most famous lossy compression algorithms:

-  Discrete Cosine Transform (**DCT**),
-  Wavelet Transform Compression,
-  Fractal Compression
-  Vector Quantization
-  Linear Predictive Coding (**LPC**)
-  A-Law Compander
-  Mu-Law Compander

For instance, the best example of transform compression is embodied in the popular JPEG standard of image encoding. The JPEG format for image compression is actually making use of the discrete cosine transform (DCT), then quantization, then Huffman coding. Also, MPEG which is a widely-used standards for audio and video compression, makes use of the DCT and motion-compensated prediction techniques. Also, the linear predictive coding (LPC) is usually employed for speech processing and speech recognition.

8-4.2. Audio Compression

Audio data compression techniques such as **MP3**, Advanced Audio Coding (**AAC**), Vorbis (**OGG**), Real Media (**RM**) or **AIFF** are commonly employed to reduce audio and speech file size.

An **audio codec** is a hardware device or a computer program that compresses/decompresses digital audio data according to a given audio file format or streaming audio format. The term **codec** is a combination of 'coder-decoder'. The object of a codec algorithm is to represent the high-fidelity audio signal with minimum number of bits while retaining the quality. This can effectively reduce the storage space and the bandwidth required for transmission of the stored audio file. Most codecs are implemented as libraries which interface to one or more multimedia players, such as **Winamp** or Windows Media Player and Real Player. In some contexts, the term audio codec can refer to a hardware implementation or sound card. When used in this manner, the phrase *audio codec* refers to the device encoding an analog audio signal.

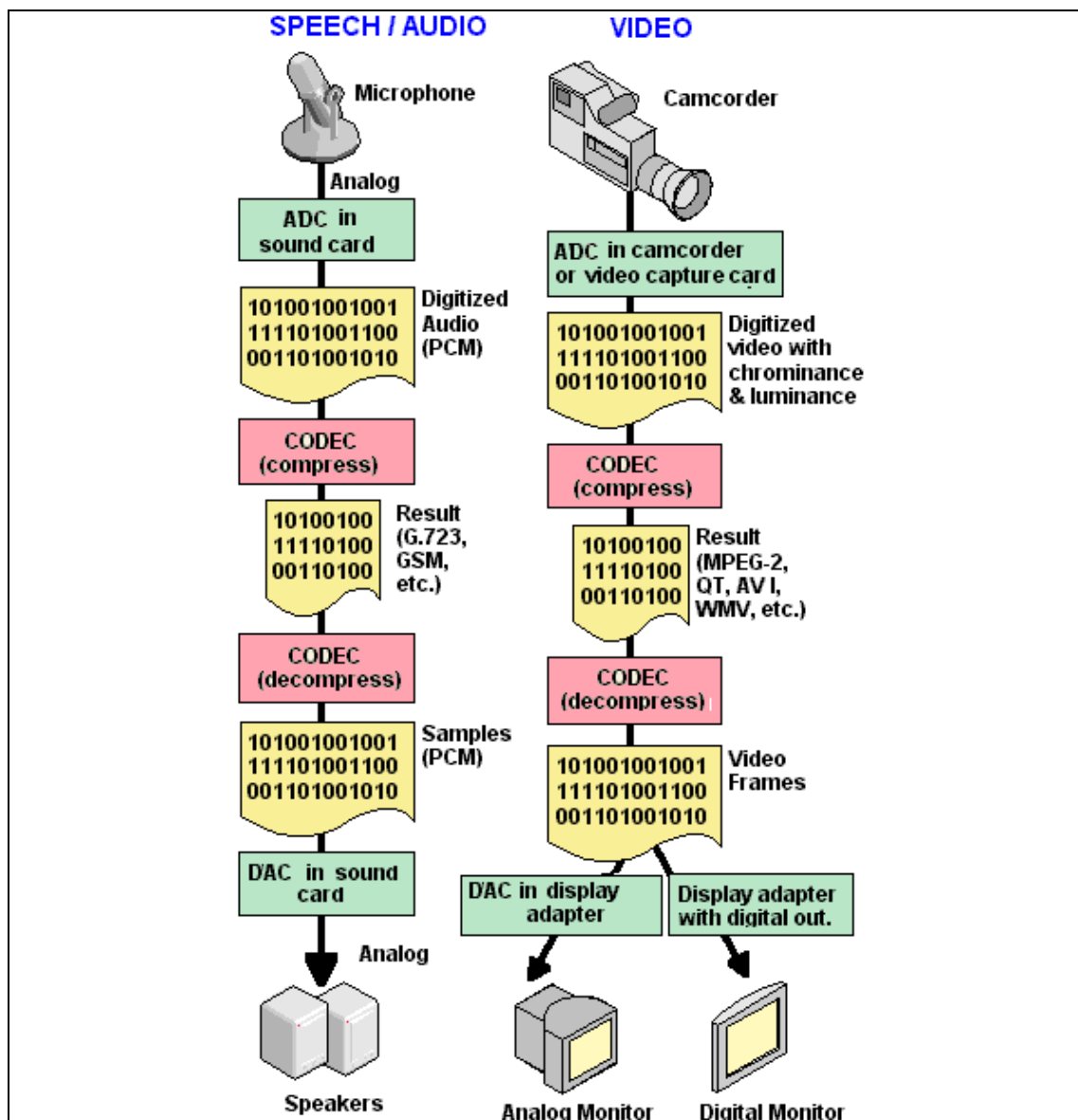


Figure 8-8. Illustration of audio and video codecs

A digital audio signal starts with an analog-to-digital converter (ADC) that converts an analog signal to a digital signal. The ADC runs at a sampling rate and converts at a known bit resolution. For example, CD audio has a sampling rate of 44.1 kHz (44,100 samples per second) and 16-bit resolution for each channel (stereo). If the analog signal is not already band-limited then an **anti-aliasing filter** is necessary before conversion, to prevent aliasing in the digital signal. Note that aliasing occurs when frequencies above the Nyquist frequency have not been band limited, and instead appear as audible artifacts in the lower frequencies.

Some audio signals such as those created by digital synthesis originate entirely in the digital domain, in which case analog to digital conversion does not take place. After being sampled with the ADC, the digital signal may then be altered in a process which is called digital signal processing where it may be filtered or have effects applied. The digital audio signal may then be stored or transmitted. Digital audio storage can be on a CD, an MP3 player, a hard drive, or USB flash drive.

i. WAV Format

Waveform audio format (**WAV**) is a Microsoft and IBM audio file format standard for storing an audio bitstream on PCs. It is an application of the RIFF bitstream format method for storing data in “chunks”, and thus also close to the IFF and the AIFF format used on Amiga and Macintosh computers, respectively. It is the main format used on Windows systems for raw and typically uncompressed audio. The usual bitstream encoding is the Pulse Code Modulation (PCM) format.

Both WAVs and **AIFFs** are compatible with Windows and Macintosh operating systems. The format takes into account some differences of the Intel CPU such as little-endian byte order. The RIFF format acts as a wrapper for various audio compression codecs. Though a WAV file can hold compressed audio, the most common WAV format contains uncompressed audio in the linear pulse code modulation (LPCM) format. The standard audio file format for CDs, for example, is LPCM-encoded, containing two channels of 44,100 samples per second, 16 bits per sample. Since LPCM uses an uncompressed, lossless storage method, which keeps all the samples of an audio track, professional users or audio experts may use the WAV format for maximum audio quality. WAV audio can also be edited and manipulated with relative ease using software. The WAV format supports compressed audio, using, on Windows, the Audio Compression Manager. Any ACM codec can be used to compress a WAV file.

The UI for Audio Compression Manager is accessible by default through Sound Recorder. Beginning, a WAVE_FORMAT_EXTENSIBLE header was defined (with Windows 2000) which specifies multiple audio channel data along with speaker positions, eliminates ambiguity regarding sample types and container sizes in the standard WAV format and supports defining custom extensions to the format chunk.

More frequently, the smaller file sizes of compressed but lossy formats such as MP3, AAC and WMA are used to store and transfer audio. Their small file sizes allow faster Internet transmission, as well as lower consumption of space on memory media. However, lossy formats trade off smaller file size against loss of audio quality.

ii. MP3 Format

MP3, (or MPEG-1 Audio Layer 3) is a standard digital audio encoding format using lossy data compression. MP3 is a common audio format for consumer audio storage, as well as a digital audio players. MP3 was designed by the Moving Picture Experts Group (MPEG). The group was formed by several teams of engineers at Fraunhofer in Germany, AT&T-Bell Labs in USA, Thomson, and others. It was approved as an ISO/IEC standard in 1991.

The use in MP3 of a lossy compression algorithm is designed to greatly reduce the amount of data required to represent the audio recording and still sound like a faithful reproduction of the original uncompressed audio for most listeners, but is not considered high fidelity audio by audiophiles. An MP3 file that is created using the mid-range bit rate setting of 128 kbit/s will result in a file that is typically about 1/10th the size of the CD file created from the original audio source.

An MP3 file can also be constructed at higher or lower bit rates, with higher or lower resulting quality. The compression works by reducing accuracy of certain parts of sound that are deemed beyond the auditory resolution ability of most people. This method is commonly referred to as perceptual coding. It internally provides a representation of sound within a short term time/frequency analysis window, by using psychoacoustic models to discard or reduce precision of components less audible to human hearing, and recording the remaining information in an efficient manner. This is relatively similar to the principles used by JPEG, an image compression format. Compression efficiency of encoders is typically defined by the bit rate, because compression ratio depends on the bit depth and sampling rate of the input signal.

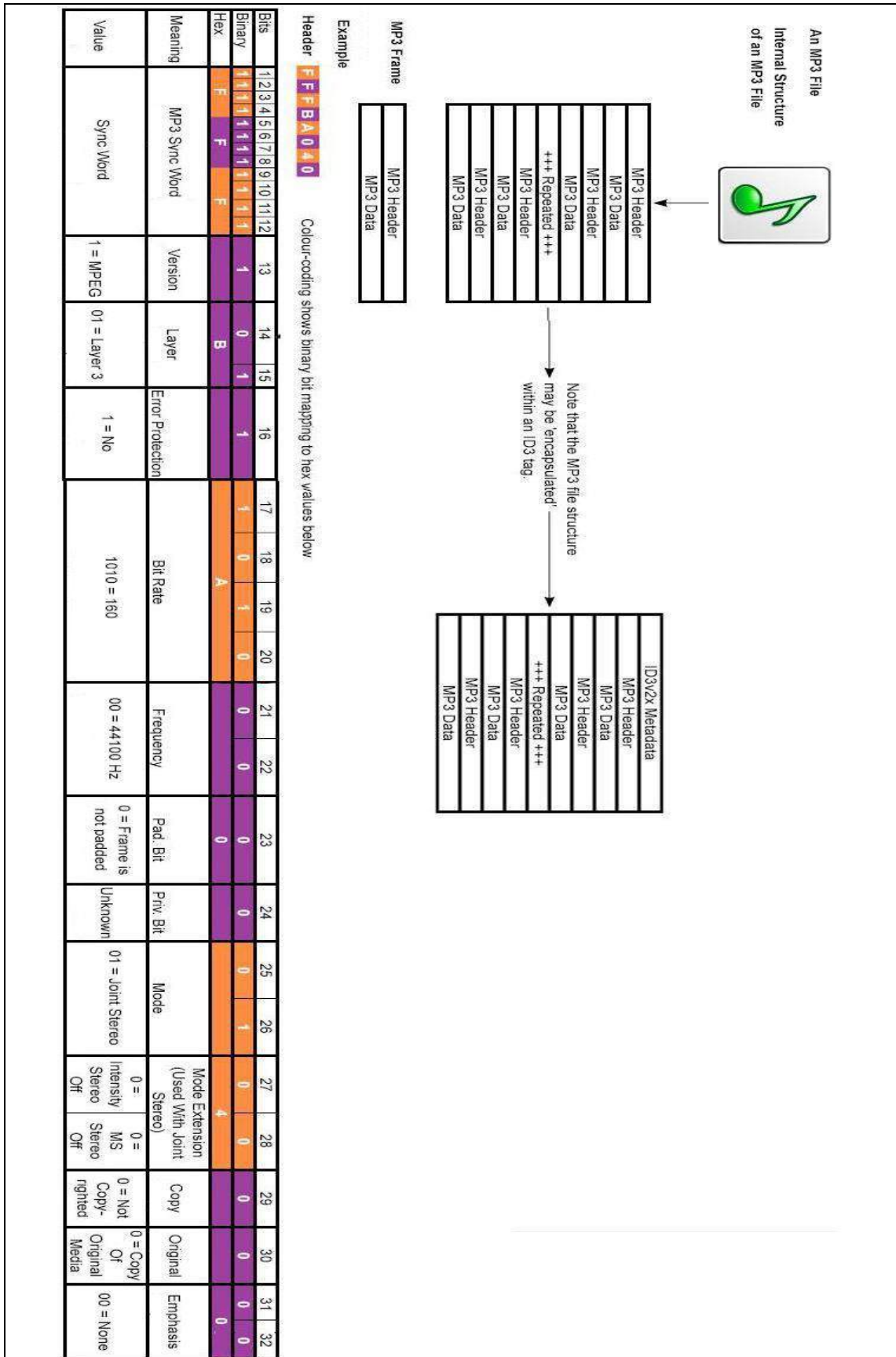


Figure 8-9. MP3 file format

Nevertheless, compression ratios are often published. They may use the CD parameters as references (44.1 kHz, 2 channels at 16 bits per channel or 2×16 bit), or sometimes the Digital Audio Tape (DAT) SP parameters (48 kHz, 2×16 bit). Compression ratios with this latter reference are higher, which demonstrates the problem with use of the term compression ratio for lossy encoders.

An MP3 file is made up of multiple MP3 frames, which consist of a header and a data block. This sequence of frames is called an elementary stream. Frames are not independent items ("byte reservoir") and therefore cannot be extracted on arbitrary frame boundaries. The MP3 Data blocks contain the (compressed) audio information in terms of frequencies and amplitudes. The diagram shows that the MP3 Header consists of a sync word, which is used to identify the beginning of a valid frame. This is followed by a bit indicating that this is the MPEG standard and two bits that indicate that layer 3 is used; hence MPEG-1 Audio Layer 3 or MP3. After this, the values will differ, depending on the MP3 file. ISO/IEC 11172-3 defines the range of values for each section of the header along with the specification of the header. Most MP3 files today contain ID3 metadata, which precedes or follows the MP3 frames.

iii. AAC Format

Advanced Audio Coding (AAC) is a standardized, lossy compression and encoding scheme for digital audio. Designed to be the successor of the MP3 format, AAC generally achieves better sound quality than MP3 at many bit rates.

The AAC format has been standardized by ISO and IEC, as part of the MPEG-2 & MPEG-4 specifications. The MPEG-2 standard contains several audio coding methods, including the MP3 coding scheme. AAC is able to include 48 full-bandwidth (up to 96 kHz) audio channels in one stream plus 15 low frequency enhancement (LFE, limited to 120 Hz) channels and up to 15 data streams. AAC is able to achieve good audio quality at data rates of 320 kbit/s for five channels. The quality for stereo is satisfactory to modest requirements at 96 kbit/s in joint stereo mode, however hi-fi transparency demands data rates of at least 192 kbit/s (VBR), as with MP3. AAC's best known use is as the default audio format of Apple's iPhone, iPod, iTunes, and the format used for all iTunes Store audio. AAC is also the standard audio format for Sony's PlayStation 3, Sony Walkman, Walkman Phones from Sony-Ericsson, Nokia N-Series Phones, the Nintendo's Wii, Nintendo DSi, and the MPEG-4 video standard.

8-4.3. Image Compression

The objective of image compression is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. Image compression can be lossy or lossless. Lossless compression is sometimes preferred for artificial images such as technical drawings, icons or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossless compression methods may also be preferred for high value content, such as medical imagery or image scans made for archival purposes. Lossy methods are especially suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate.

Methods for lossless image compression are:

- **Run-length encoding (RLE)** used as default method in PCX and as one of possible in BMP, TGA, TIFF
- **Entropy encoding**
- **Adaptive dictionary** algorithms such as **LZW** – used in GIF and TIFF
- **Deflation** – used in PNG, MNG and TIFF

Methods for lossy image compression:

- Reducing the color space to the most common colors in the image. The selected colors are specified in the color palette in the header of the compressed image. Each pixel just references the index of a color in the color palette. This method can be combined with dithering to avoid posterization.
- Chroma sub-sampling. This takes advantage of the fact that the eye perceives brightness more sharply than color, by dropping half or more of the chrominance information in the image.
- Transform coding. This is the most commonly used method. A Fourier-related transform such as DCT or the wavelet transform are applied, followed by quantization and entropy coding.
- Fractal compression.

i, BMP and DIB Format

The bitmap **BMP** file format, sometimes called Device Independent Bitmap (**DIB**) file format, is an image file format used to store bitmap digital images, on Microsoft Windows. Many graphical user interfaces use bitmaps in their built-in graphics subsystems; for example, the GDI subsystem of Microsoft Windows platforms.

Some bitmap images are compressed with an RLE-type compression. Uncompressed formats are generally unsuitable for transferring images on the Internet or other slow or capacity-limited media. However, in uncompressed BMP files, and many other bitmap file formats, image pixels are stored with a color depth of 1, 4, 8, 16, 24, or 32 bits per pixel. The bits representing the bitmap pixels are packed within rows. Depending on the color depth, a pixel in the picture will occupy at least $n/8$ bytes (n is the bit depth, since 1 byte equals 8 bits). Images of 8 bits and fewer can be either grayscale or indexed color. Uncompressed bitmap files (such as BMP) are typically much larger than compressed image file formats for the same image. For example, a 1058×1058 picture, occupies about 287.65 kB in the PNG format, takes about 3358 kB as a 24-bit BMP file. A typical BMP file usually contains the following blocks of data:

Bmp File Header	Stores general information about the Bmp file.
Bitmap Information	Stores information about the bitmap image.
Color Palette	Stores the definition of the colors being used for indexed color bitmaps.
Bitmap Data	Stores the actual image, pixel by pixel.

ii. TIFF Format

TIFF is a tag-based file format designed to promote universal interchanges of digital image data. Because TIFF files do not have a single way to store image data, there are many versions of TIFF. LEADTOOLS supports the most common TIFF formats. The usual file extension is TIFF for single-image files, and MPT for multipage files.

iii. JPEG Format

The name "JPEG" stands for Joint Photographic Experts Group, the name of the committee that created this standard. The group was organized in 1986, issuing a standard in 1992, which was then approved in 1994 as ISO 10918-1. As we've mentioned so far, the JPEG standard makes use of lossy compression techniques, on the basis of transforms. Transform compression is based on a simple premise: when the signal is passed through the Fourier (or other) transform, the resulting data values will no longer be equal in their information carrying roles. In particular, the low frequency components of a signal are more important than the high frequency components. Removing 50% of the bits from the high frequency components might remove, say, only 5% of the encoded information.

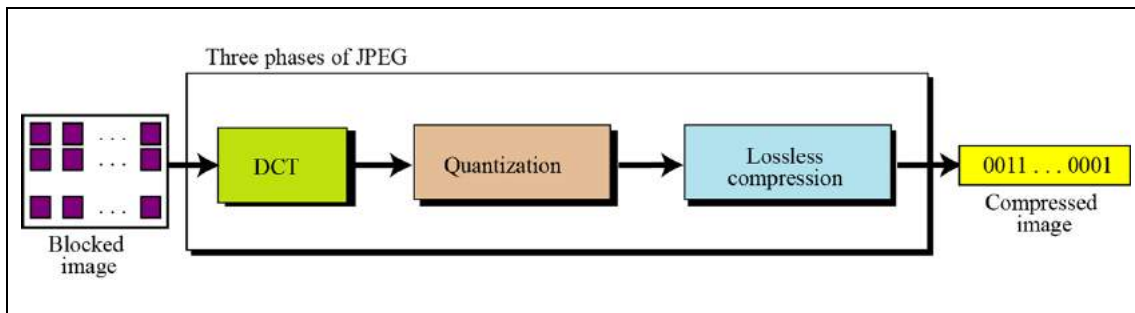


Figure 8-10(a). Image compression in JPEG file format

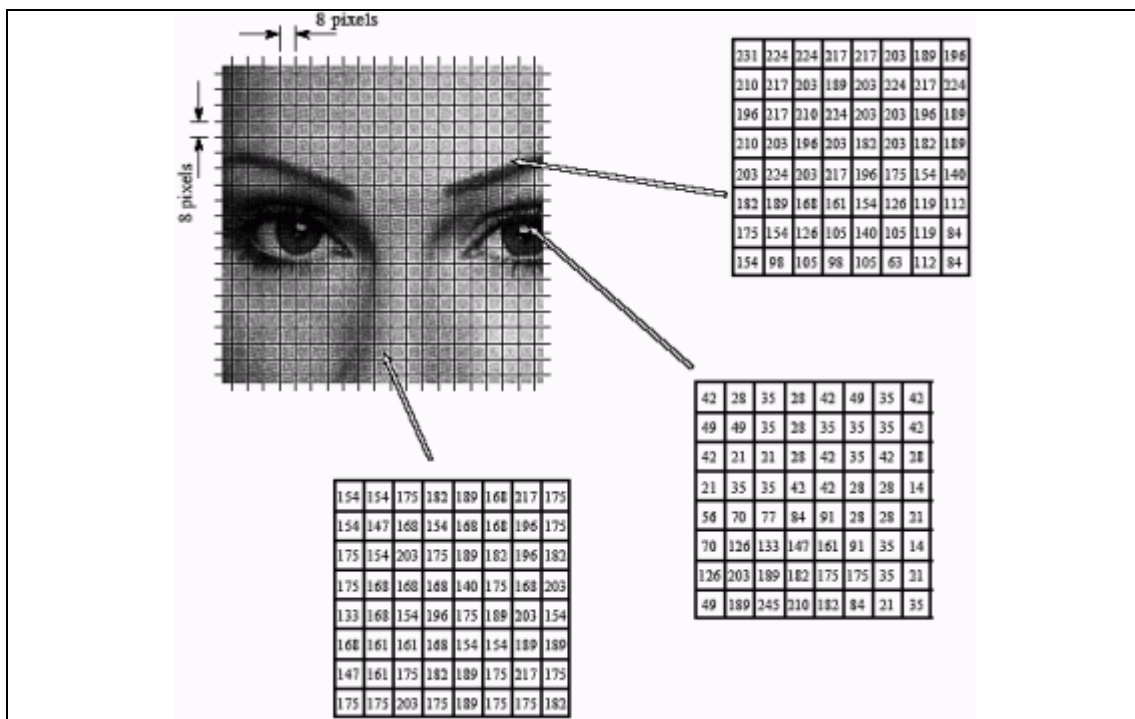


Figure 8-10(b). Example of image compression in JPEG file format

As shown in figure 8-10, the JPEG compression starts by breaking the image into 8×8 pixel groups. The full JPEG algorithm can accept a wide range of bits per pixel, including the use of color information. In this example, each pixel is a single byte, a grayscale value between 0 and 255. These 8×8 pixel groups are treated independently during compression. That is, each group is initially represented by 64 bytes. After transforming and removing data, each group is represented by, say, 2 to 20 bytes. During decompression, the inverse transform is taken of the 2 to 20 bytes to create an approximation of the original 8×8 group. These approximated groups are then fitted together to form the uncompressed image. Why use 8×8 pixel groups instead of, for instance, 16×16 ? The

8×8 grouping was based on the maximum size that integrated circuit technology could handle at the time the standard was developed. In any event, the 8×8 size works well, and it may or may not be changed in the future.

Many different transforms have been investigated for data compression, some of them invented specifically for image compression purpose. For instance, the *Karhunen-Loeve* transform provides the best possible compression ratio, but is difficult to implement. The *Fourier transform* is easy to use, but does not provide adequate compression. After many discussions and comparisons, the winner was the Discrete Cosine Transform (**DCT**).

A JPEG image file contains a sequence of *markers*, each of which begins with a 0xFF byte followed by a byte indicating what kind of marker it is. Some markers consist of just those two bytes; others are followed by two bytes indicating the length of marker-specific payload data that follows. (The length includes the two bytes for the length, but not the two bytes for the marker.) Some markers are followed by entropy-coded data; the length of such a marker does not include the entropy-coded data. Within the entropy-coded data, after any 0xFF byte, a 0x00 byte is inserted by the encoder before the next byte, so that there does not appear to be a marker where none is intended. Decoders must skip this 0x00 byte. This technique, called *byte stuffing*, is only applied to the entropy-coded data, not to marker payload data.

Although a JPEG file can be encoded in various ways, most commonly it is done with **JFIF** encoding. The JFIF encoding process consists of several steps:

1. The representation of the colors in the image is converted from RGB to YCbCr, consisting of one *luma* component (Y), representing brightness, and two *chroma* components, (Cb and Cr), representing color. This step is sometimes skipped.
2. The resolution of chroma data is reduced, by a factor of 2. This reflects the fact that the eye is less sensitive to fine color details than to fine brightness details.
3. The image is split into blocks of 8×8 pixels, and for each block, each of the Y, Cb, and Cr data undergoes a discrete cosine transform (DCT). A

DCT is similar to a Fourier transform in the sense that it produces a kind of spatial frequency spectrum.

4. The amplitudes of the frequency components are quantized. Human vision is much more sensitive to small variations in color or brightness over large areas than to the strength of high-frequency brightness variations. Therefore, the magnitudes of the high-frequency components are stored with a lower accuracy than the low-frequency components. The quality setting of the encoder (for example 50 or 95 on a scale of 0–100 in the Independent JPEG Group's library affects to what extent the resolution of each frequency component is reduced. If an excessively low quality setting is used, the high-frequency components are discarded altogether.
5. The resulting data for all 8×8 blocks is further compressed with a loss-less algorithm, a variant of Huffman encoding.

The decoding process reverses these steps. In the remainder of this section, the encoding and decoding processes are described in more detail.

Note 8-2. DCT & DFT in Image Compression

Why is the DCT better than the discrete Fourier transform (DFT) for image compression? The main reason is that the DCT has one-half cycle basis functions, as shown in Appendix H. This makes a gentle slope from one side of the array to the other. In comparison, the lowest frequencies in the Fourier transform form one complete cycle. Images nearly always contain regions where the brightness is gradually changing over a region. Using a basis function that matches this basic pattern allows for better compression.

8-4.4. Video Compression

Video compression is a combination of image compression and motion compensation. Compressed video can effectively reduce the bandwidth required to transmit video via terrestrial broadcast, via cable TV, or via satellite TV services.

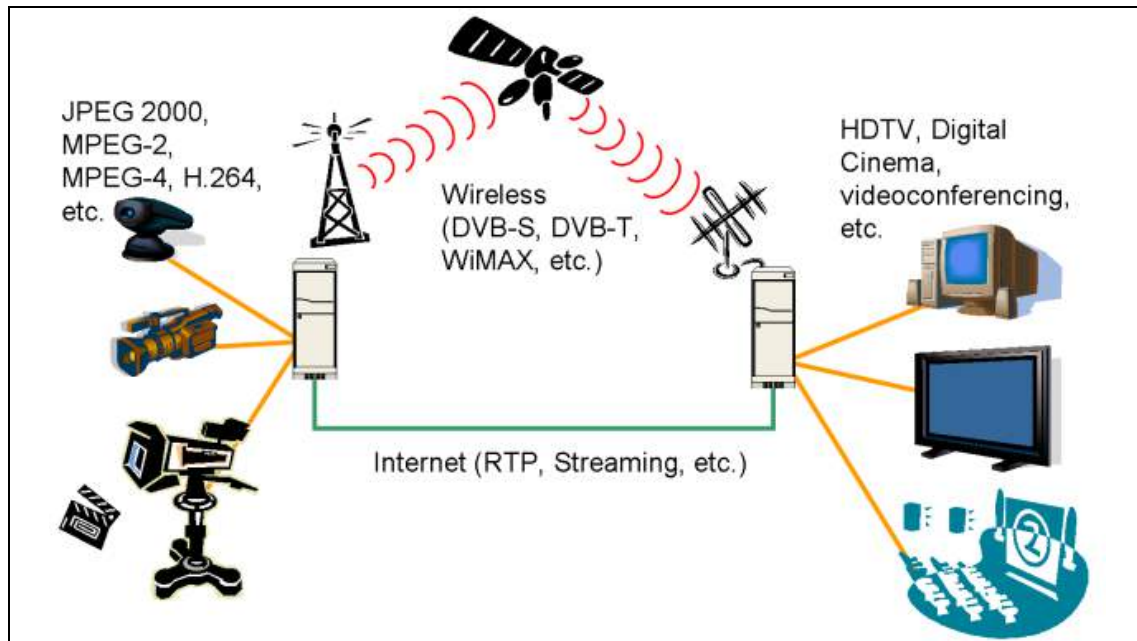


Figure 8-11. Some applications of video compression

Most video compression algorithms are lossy. While lossless compression of video is possible, it is rarely used. In fact video compression operates on the premise that much of the data present before compression is not necessary for achieving good perceptual quality. For example, DVDs use a video coding standard called MPEG-2 that can compress video data about 15 to 30 times. Video compression typically operates on square-shaped groups of neighboring pixels, often called macroblocks. These macroblocks of pixels are compared from one frame to the next and the video compression codec (encoder/decoder) sends only the *differences* within those blocks. This works extremely well if the video has fixed big portion, like text or background images. A data frame is a set of all pixels that correspond to a single time moment. Basically, a frame is the same as a still picture.

One of the most powerful techniques for compressing video is **interframe** compression. Interframe compression uses one or more earlier or later frames in a sequence to compress the current frame, while **intraframe** compression uses only the current frame, which is effectively image compression.

Nowadays, nearly all video compression methods apply a discrete cosine transform (**DCT**) for spatial redundancy reduction. Other methods, such as fractal compression, matching pursuit and the discrete wavelet transform (**DWT**) are still research phase.

i. AVI Format

AVI stands for Audio Video Interleave. AVI was defined by Microsoft and became the most common format for audio/video data on the PC. AVI files can contain both audio and video data in a file container that allows synchronous audio-with-video playback.

AVI is a derivative of the Resource Interchange File Format (**RIFF**), which divides a file's data into blocks, or chunks. Each chunk is identified by a *4CC* tag. An AVI file takes the form of a single chunk in a RIFF formatted file, which is then subdivided into two mandatory "chunks" and one optional chunk. The first sub-chunk is identified by the *hdrl* tag.

This sub-chunk is the file header and contains metadata about the video, such as its width, height and frame rate. The second sub-chunk is identified by the *movi* tag. This chunk contains the actual audio/visual data that make up the AVI movie. The third optional sub-chunk is identified by the *idx1* tag which indexes the offsets of the data chunks within the file. By way of the RIFF format, the audio/visual data contained in the *movi* chunk can be encoded or decoded by a codec (Coder/Decoder) software. Upon creation of the file, the codec translates between raw data and the (compressed) data format used inside the chunk. An AVI file may carry audio/visual data inside the chunks in virtually any compression scheme, including Full Frame (Uncompressed), Intel Real Time (Indeo), Cinepak, Motion JPEG, Editable MPEG, RealVideo, and MPEG-4 Video.

ii. MPEG Format

MPEG is a compression standard for digital video sequences, such as used in computer video and digital television networks. Moving Picture Experts Group (MPEG) is a working group of ISO/IEC in charge of the development of standards for coded representation of digital audio and video. Established in 1988, the group has produced MPEG-1, the standard on which such products as Video CD and MP3 are based, MPEG-2, the standard on which such products as Digital Television set top boxes and DVD are based, MPEG-4, the standard for multimedia for the fixed and mobile web, MPEG-7, the standard for description and search of audio and visual content and MPEG-21, the Multimedia Framework.

In addition to reducing the data rate, MPEG has several important features. The movie can be played forward or in reverse, and at either normal or fast speed. The encoded information is random access, that is, any individual frame in the sequence can be easily displayed as a still picture. This goes along with making the movie editable, meaning that short segments from the movie can be encoded only with reference to themselves, not the entire sequence. MPEG is designed to be robust to errors. The last thing you want is for a single bit error to cause a disruption of the movie.

The approach used by MPEG can be divided into two types of compression: **within-the-frame** and **between-frame**. Within-the-frame compression means that individual frames making up the video sequence are encoded as if they were ordinary still images. This compression is performed using the JPEG standard, with just a few variations. In MPEG terminology, a frame that has been encoded in this way is called an intra-coded or I-picture.

Most of the pixels in a video sequence change very little from one frame to the next. Unless the camera is moving, most of the image is composed of a background that remains constant over dozens of frames. MPEG takes advantage of this with a sophisticated form of delta encoding to compress the redundant information between frames. After compressing one of the frames as an I-picture, MPEG encodes successive frames as predictive-coded or P-pictures. That is, only the pixels that have changed since the I-picture are included in the P-picture.

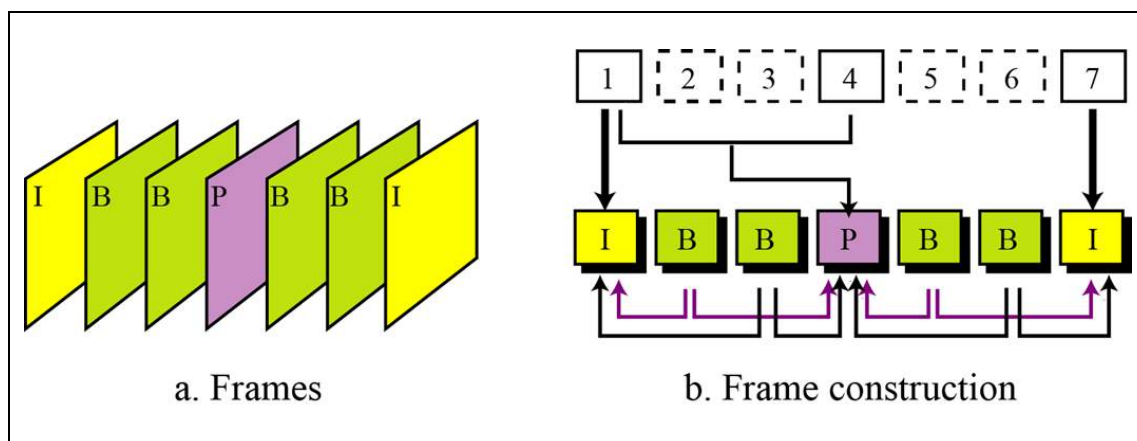


Figure 8-12. Illustration of the video compression process

While these two compression schemes form the backbone of MPEG, the actual implementation is immensely more sophisticated than described

here. For example, a P-picture can be referenced to an I-picture that has been shifted, accounting for motion of objects in the image sequence. There are also bidirectional predictive-coded or B-pictures. These are referenced to both a previous and a future I-picture. This handles regions in the image that gradually change over many of frames. The individual frames can also be stored out-of-order in the compressed data to facilitate the proper sequencing of the I, P, and B-pictures. The addition of color and sound makes this all the more complicated.

The main distortion associated with MPEG occurs when large sections of the image change quickly. In effect, a burst of information is needed to keep up with the rapidly changing scenes. If the data rate is fixed, the viewer notices "blocky" patterns when changing from one scene to the next. This can be minimized in networks that transmit multiple video channels simultaneously, such as cable television. The sudden burst of information needed to support a rapidly changing scene in one video channel, is averaged with the modest requirements of the relatively static scenes in the other channels.

iii. 3GP Format

3GP is a multimedia container format defined by the Third Generation Partnership Project (3GPP) for use on 3G mobile phones but can also be played on some 2G and 4G phones.

3GP is a simplified version of the MPEG-4 Part 14 (MP4) container format, designed to decrease storage and bandwidth requirements in order to accommodate mobile phones. It stores video streams as MPEG-4 Part 2 or H.263 or MPEG-4 Part 10 (AVC/H.264), and audio streams as AMR-NB, AMR-WB, AMR-WB+, AAC-LC or HE-AAC. A 3GP file is always big-endian, storing and transferring the most significant bytes first. It also contains descriptions of image sizes and bit rate. There are two different standards for this format:

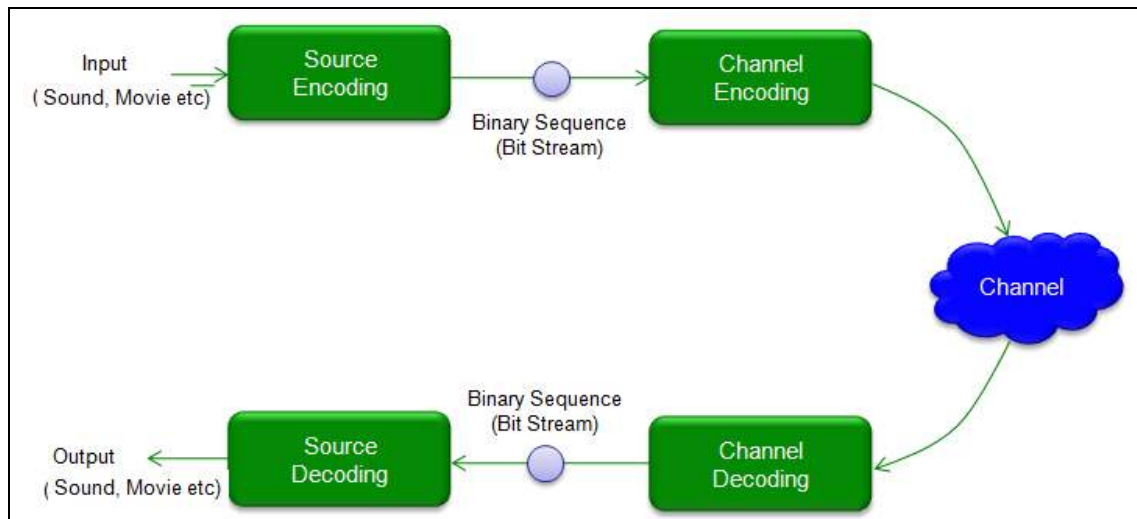
- **3GPP** (for GSM-based Phones, with file extension .3GP)
- **3GPP2** (for CDMA-based Phones, with file extension .3G2)

Both are based on MPEG-4 and H.263 video, and AAC or AMR audio. When transferred to a computer, 3GP movies can be viewed on Linux, Mac, and Windows platforms with MPlayer and VLC media player. Programs such as Media Player Classic, Totem, RealPlayer, QuickTime, and GOM Player can also be used. Some cell phones use the .mp4 extension for 3GP video.

Most 3G mobile phones support the playback and recording of video in 3GP format. Memory, maximum file-size for playback and recording, and resolution limits exist and vary, for different manufacturers. Some 2G/2.5G (Edge) cellphones may also playback and record in 3GPP format, with certain limitations. In *iMovie*, a movie exported using the Tiny setting is saved as a .3GP file and can be played on a Mac, an *iPhone*, an *iPod* or using Apple's .Mac Web Gallery service. Audio imported from CD onto a PlayStation 3 will also copy onto USB devices in the 3GP format.

8-5. Summary

Source coding is a mapping from (a sequence of) symbols from an information source to another compressed sequence of alphabet symbols (usually bits) such that the source symbols can be exactly recovered from the binary bits (lossless source coding) or recovered within some distortion (lossy source coding). This is the concept behind data compression.



Reducing data size and transmission time are the key benefits of lossless data compression. The compression of data may be subdivided into two broad categories:

- 1- Lossless data compression,
- 2- Lossy data compression.

Lossless compression schemes are reversible so that the original data can be reconstructed, while lossy schemes accept some loss of data in order to achieve higher compression. The design of data compression schemes therefore involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (if using a lossy compression scheme), and the computational resources required to compress and uncompress the data.

There's no unique Huffman code and every Huffman code has the same average code length. The Huffman code algorithm can be summarized as follows:

1. Make a leaf node for each code symbol
Add the generation probability of each symbol to the leaf node
2. Take the two leaf nodes with the smallest probability and connect them into a new node
Add 1 or 0 to each of the two branches
The probability of the new node is the sum of the probabilities of the two connecting nodes.
3. If there is only one node left, the code construction is completed. If not, go back to (2)

Shannon-Fano Coding:

Shannon-Fano coding is prefix codes which produces variable size codes for the symbols occurring with different probabilities. The coding depends on the probability of occurrence of the symbol and the general idea is to assign shorter codes for symbols that occur more frequently and long codes for the symbols occurring less frequently. so the probabilities must be known This makes the algorithm inefficient.

The algorithm used for generating Shannon-Fano codes is as follows:

- 1) For a given list of symbols, develop a corresponding list of probabilities so that each symbol's relative probability is known.
- 2) List the symbols in the order of decreasing probability.
- 3) Divide the symbols into two groups so that each group has equal probability.
- 4) Assign a value 0 to first group and a value 1 to second group.
- 5) Repeat steps 3 and 4, each time partitioning the sets with nearly equal probabilities as possible until further partitioning is not possible.

8-7. Problems

8-1) What's difference between lossy and lossless compression techniques?

8-2) Describe the RLE coding operation in 3 points

8-3) Mention the different techniques which are typically used in speech compression and explain the main differences between them.

8-4) What do you know about image compensation, and how this technique is use in video compression?

8-5) Calculate the efficiency of a source encoder, whose symbols are (11, 10, 00, 01) and their probabilities are ($3/8, 1/4, 1/2, 1/4$).

Hint: Calculate the Entropy of the source H and its average word length \underline{L} and then find the efficiency $\eta = L_{min} / \underline{L} = H(x) / \underline{L}$
The entropy is given by:

$$H(x) = \sum p_i \log_2 (1/p_i) = p_0 \log_2 (1/p_0) + p_1 \log_2 (1/p_1) + \dots$$

8-7. References

- [1] D. A. **Huffman**, "A Method for the Construction of Minimum Redundancy Codes," *Proceedings of the IRE*, Vol. 40, pp. 1098--1101, **1952**
- [2] J. **Ziv** and A. **Lempel**, "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, Vol. 23, pp. 338--342, **1977**.
- [3] J. **Ziv** and A. **Lempel**, "Compression of Individual Sequences Via Variable-Rate Coding," *IEEE Transactions on Information Theory*, Vol. 24, pp. 530--536, **1978**.
- [4] T. A. **Welch**, "A Technique for High-Performance Data Compression," *Computer*, pp. 8--18, **1984**
- [5] M. **Thomas** and A. Thomas Joy. *Elements of information theory*, 1st Edition. New York: Wiley-Interscience, 1991. 2nd Edition. New York: Wiley-Interscience, **2006**

Chapter
9

Data Encryption

Contents

- 9-1. Introduction**
- 9-2. Encryption Algorithms**
 - 9-2.1. DES Algorithm
 - 9-2.2. Feistel (F) Function
 - 9-2.3. Key Schedule
 - 9-3.4. AES Algorithm
- 9-3. Security and Cryptanalysis**
 - 9-3.1. Brute Force Attack
 - 9-3.2. Attacks Faster than Brute-Force
- 9-4. Cryptanalytic Properties**
- 9-5. Two-Level Key**
- 9-6. Public Key Infrastructure (PKI)**
- 9-7. Functional Encryption**
- 9-8. Applications of Encryption**
- 9-9. Securing Database Networks**
- 9-10. Future Developments**
- 9-11. Summary**
- 9-12. Problems**
- 9-13. Bibliography**

Chapter

9

Data Encryption

9-1. Introduction

Data encryption is basically the process of hiding information. Data Communications usually require some form of encryption in order to keep the data safe from interception. Encryption is increasingly used to protect digital information, from personal details held on a computer to financial details transmitted over the Internet. Data encryption is the process of scrambling of stored or transmitted information so that it is unintelligible until it is unscrambled by the intended recipient.

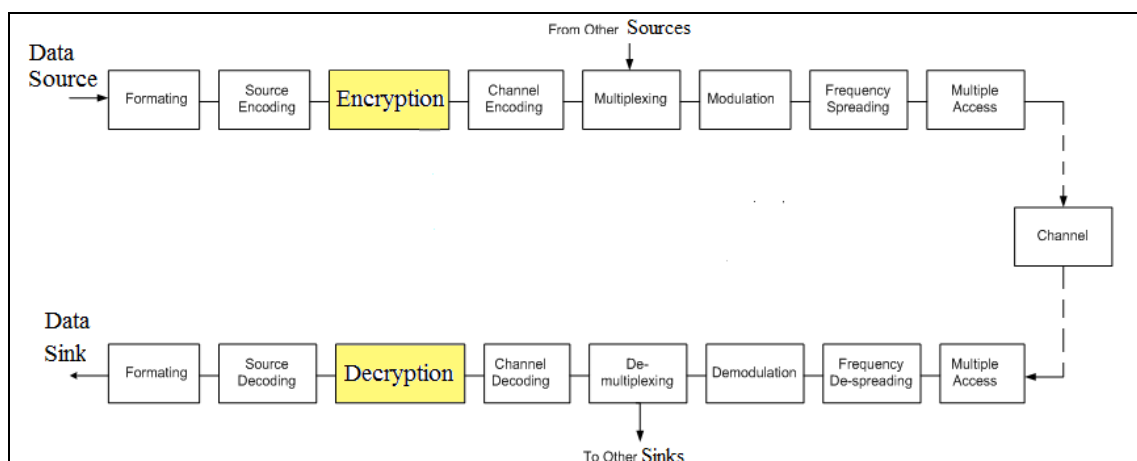


Figure 9-1. Overall digital communication system, with encryption/decryption

Encrypting data provides additional benefits besides protecting the confidentiality of a message. These advantages include ensuring that messages have not been altered during transit and verifying the identity of the sender.

9-1.1. Elements of Encryption Systems

The modern cryptography is based on the use of algorithms to scramble (encrypt) a message, called **plaintext**, into unintelligible babble, called **ciphertext**. This operation requires the use of a **key**. Therefore, the encrypting and decrypting data is nothing more than passing the data through an **algorithm** to make something readable only to the intended

recipients. The process for encryption is essentially identical to the process for decryption. At the document level, *encryption* takes an easily read *plaintext* file and turns it into *ciphertext* using a *key* in conjunction with a specific algorithm. .

1. **Plaintext:** A message before encryption, i.e. in its usual form which anyone can read. This is sometimes called *cleartext*.
2. **Ciphertext :** Text which is encrypted by some encryption system.
3. **Key:** A method of opening an encryption. A key can be as simple as a string of text characters, or a series of hexadecimal digits.
4. **Algorithm:** A set of computing steps to achieve a desired result.

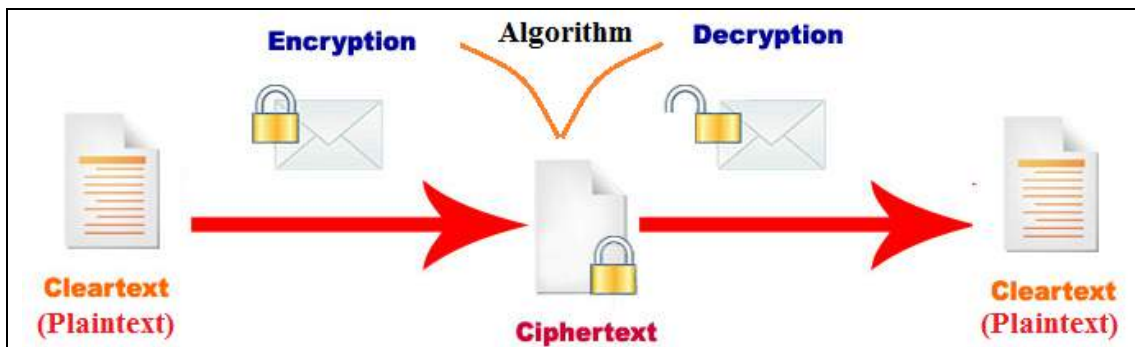


Figure 9-2 Elements of the encryption/decryption system

9-1.2. Illustration Example

As we pointed out so far, encryption involves taking an original message or **plaintext** and converting it into **ciphertext** using an **encryption algorithm** and an **encryption key**. Historically, encryption acted on letters of the alphabet. The **Caesar Cipher**, one of the oldest techniques, is a very simple encryption example:

- Take the plaintext: *Parliament is in session*;
- Encrypt according to the encryption algorithm “replace each letter with that k places to the right of it in the alphabet”, where k , the encryption key, is 3;
- The ciphertext is *sduoldphqw lv lq vhwvrlq* can be converted back to plaintext with a decryption **algorithm** and decryption **key**, in this case “replace each letter with that 3 places to the left of it in the alphabet”.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Note that the letters “wrap around” at the end of the alphabet, which can be mathematically be expressed as reduction modulo 26, e.g., $23 + 3 =$

$26 \equiv 0 \pmod{26}$.

This encryption may be described mathematically as follows:

Let k = key, x = plaintext, y = ciphertext, such that $k, x, y \in \{0, 1, \dots, 25\}$

- Encryption equation: $y = e_k(x) \equiv x + k \pmod{26}$
- Decryption equation: $x = d_k(y) \equiv y - k \pmod{26}$

9-1.3. Encryption Types & Algorithms

To protect data transmissions against interception, various encryption methods are used. The old classical methods were manual or mechanical, and were based on the substitution and transposition of characters, in the original message to be encrypted.

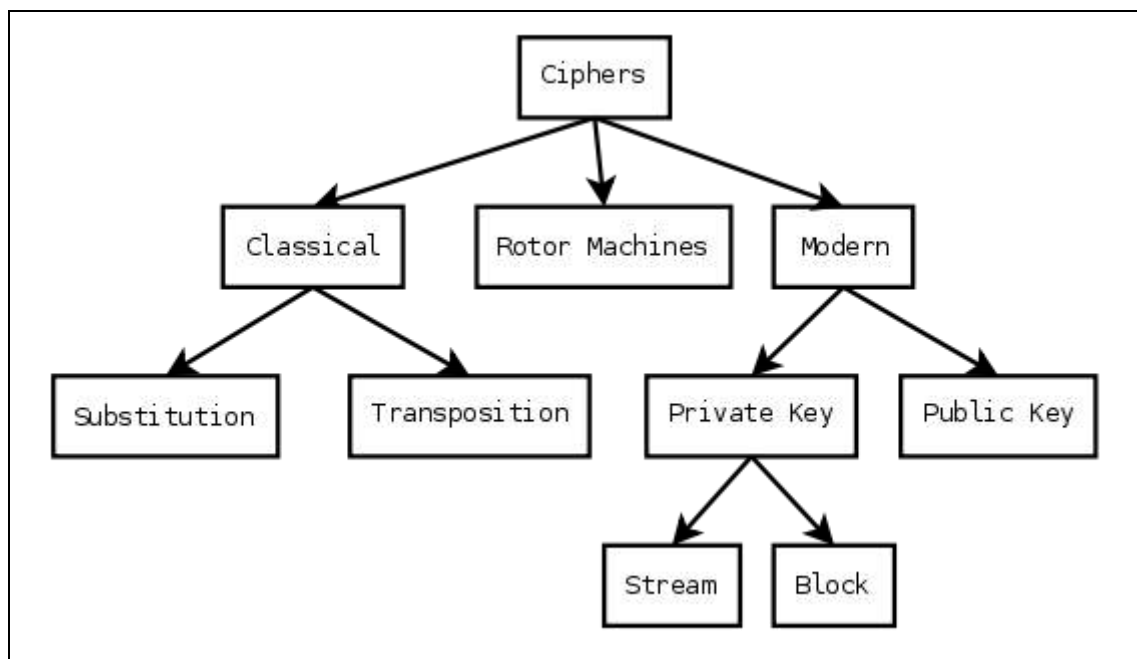


Figure 9-3. Types of encryption

The modern encryption methods can be divided into three categories: **symmetric** cryptography, **asymmetric** and **hashing** cryptography. Each of these encryption methods has its own uses, advantages, and disadvantages.

Symmetric key algorithms use the same key to both encrypt and decrypt data. They are generally faster than asymmetric key algorithms and are often used to encrypt large blocks of data. Algorithms you'll hear mentioned include DES, RC5, and AES (the standard for the U.S. government). Either DES or AES is required when encrypting stored credit card numbers.

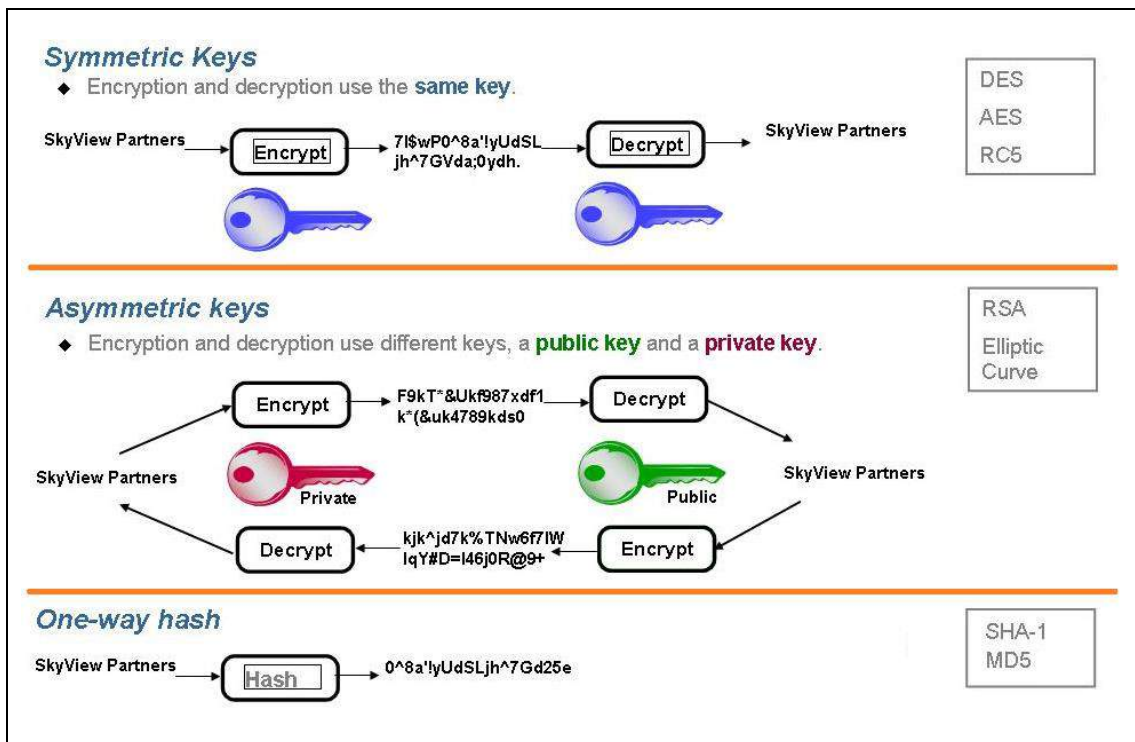


Figure 9-4. Modern encryption/decryption algorithms

Asymmetric key algorithms require two encryption keys: one to encrypt the data and the other to decrypt the data. It doesn't matter which one does which function; it's just that you can't use only one key to both encrypt and decrypt data. These algorithms are typically used for authentication; for example, they're used during an SSL or VPN "handshake" to verify that the client knows the server to which it's trying to connect. Once the verification is complete, a "session key" is exchanged by the client and server, and symmetric key encryption is used for the actual data transmission. Asymmetric key algorithm types include RSA and Elliptic Curve.

Hash algorithms produce a result that is not decryptable. They are typically used for comparing two values to ensure they're the same. For example, digital signature verification uses a hash function to ensure that the signed document has not been altered. Because a hashed value cannot return the original value, this method provides the most secure form of storing data and is the most appropriate method when the original (cleartext) form is not required. MD5 and SHA-1 are two examples of hash algorithms.

9-2. Symmetric-key Encryption Algorithms

The Symmetric-key cryptography¹ is one of the old and secure encryption methods. The name "private key" comes from the fact that the key used to encrypt and decrypt data must remain secret because anyone with access to it can read the coded messages. A sender encodes a message into ciphertext using a key, and the receiver uses the same key to decode it. People can use this encryption method as either a "stream" cipher or a "block" cipher, depending on the amount of data being encrypted or decrypted at a time. A stream cipher encrypts data one character at a time as it is sent or received, while a block cipher processes fixed chunks of data. Common symmetric encryption algorithms include Data Encryption Standard (DES), Advanced Encryption Standard (AES), and International Data Encryption Algorithm (IDEA).

The following table shows some symmetric-key encryption algorithms and their features.

Table 9-1. Famous Encryption algorithms (Ciphers).

Cipher	Author	Key length	Comments
DES	IBM	56 bits	Weak & old
RC4	Ronald Rivest	1-2048 bits	some keys weak
RC5	Ronald Rivest	129-256 bits	Good, patented
AES (Rijndael)	Daemen , Rijmen	129-256 bits	Best choice
Serpent	Anderson, Biham, Knudsen	129-256 bits	Very strong
Triple DES	IBM	168 bits	Good, old
Twofish	Bruce Schneier	129-256 bits	Very strong;

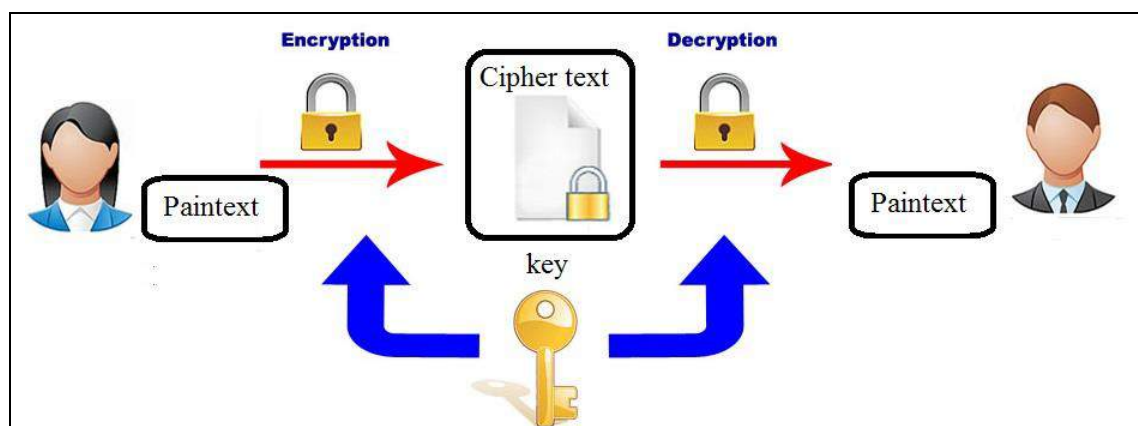


Figure 9-5. Schematic representation of the symmetric-key encryption.

¹ Also called: **private-key**, **single-key** or **secret-key** cryptography

The Data Encryption Standard (**DES**) is a method for encrypting information (cipher) that was selected as an official Federal Information Processing Standard (FIPS) in 1976 and has then enjoyed an international widespread use. It is based on **Symmetric-key** algorithm that uses a 56-bit **key**. The algorithm was initially controversial with some design elements. DES was considered to be insecure for many applications. This is due to the 56-bit key size being too small. In January, 1999, the Electronic Frontier Foundation (EFF) and distributed.net collaborated to break a DES key in 22 hours and 15 min. There are also some analytical results which demonstrate theoretical weaknesses in the cipher. The algorithm is believed to be secure in the form of Triple DES, although there are theoretical attacks. DES consequently came under intense academic scrutiny which motivated the modern understanding of block ciphers and their cryptanalysis. **DES** takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another ciphertext bitstring of the same length. In the case of DES, the block size is 64 bits. DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56 bits, and it is usually quoted as such. Like other block ciphers, DES by itself is not a secure means of encryption but must instead be used in a mode of operation. FIPS-81 specifies several modes for use with DES. Further comments on the usage of DES are contained in FIPS-74. In some documentation, a distinction is made between DES as a standard and DES the algorithm which is referred to as the Data Encryption Algorithm (**DEA**). Later on, the DES cipher has been superseded by the **Advanced Encryption Standard (AES²)**.

9-2.1. DES Algorithm

The structure of the DES cryptographic algorithm is shown in figure 9-3. There are 16 identical stages of processing, termed *rounds* or *iterations*. There is also an initial and final permutations (transpose), termed **IP** and **FP**, which are inverses (IP undoes the action of FP, and vice versa). IP and FP have almost no cryptographic significance, but were included to facilitate loading blocks, as well as to make DES run slower in software.

² The AEC algorithm is sometimes called *Rijndael*. The Rijndael algorithm is distributed for free.

Before the main rounds, the block is divided into two 32-bit halves and processed alternately; this algorithm is known as the **Feistel** scheme.

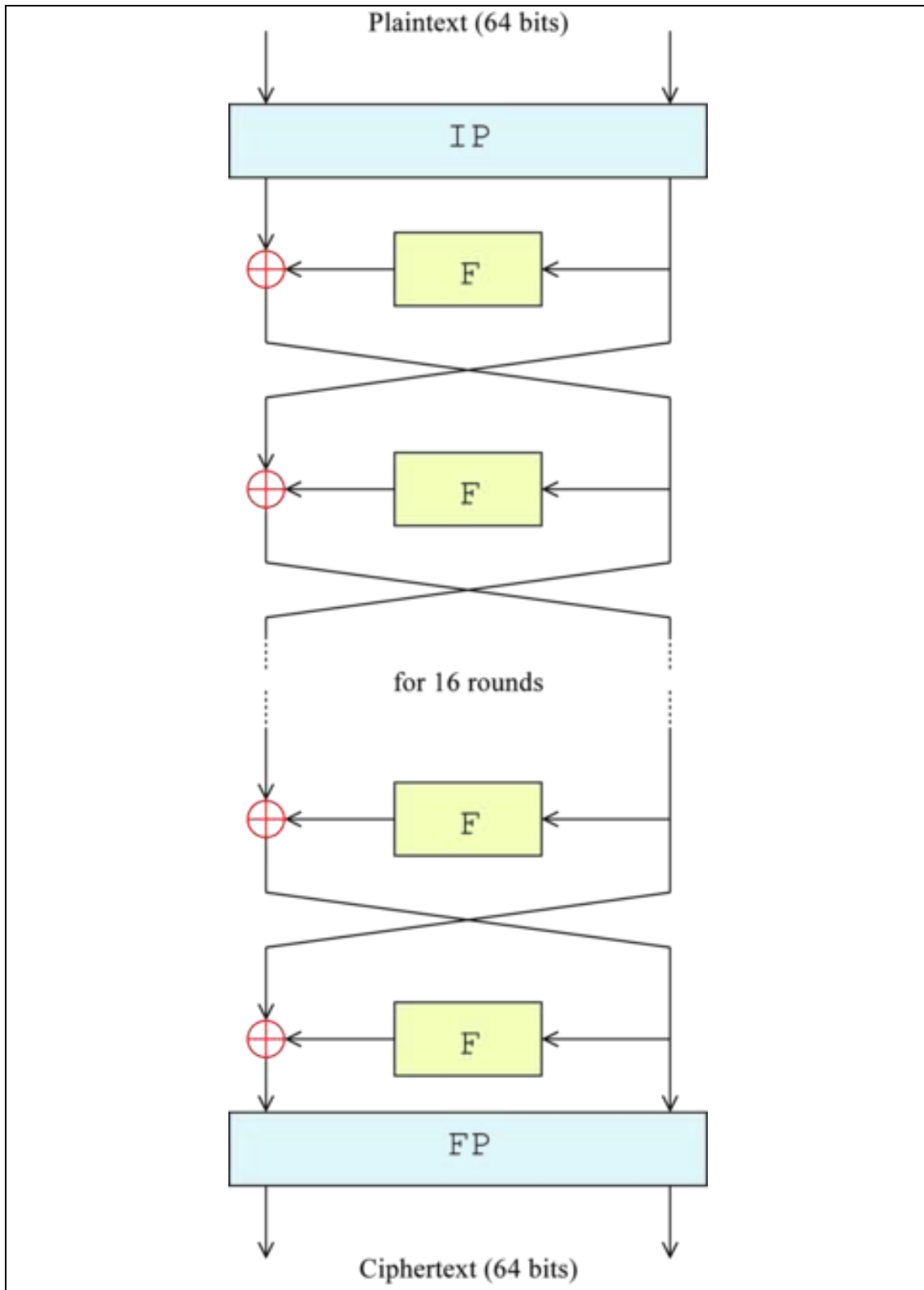


Figure 9-6. Overall Feistel structure of **DES** For brevity, the following description omits the exact transformations and permutations which specify the algorithm

The **Feistel structure** ensures that decryption and encryption are very similar processes — the only difference is that the subkeys are applied in the reverse order when decrypting. The rest of the algorithm is identical. This greatly simplifies implementation, particularly in hardware, as there is no need for separate encryption and decryption algorithms. The \oplus symbol denotes the exclusive-OR (XOR) operation. The *F-function* scrambles half a block together with some of the key. The output from the F-function is then combined with the other half of the block, and the halves are swapped before the next round. After the final round, the halves are not swapped; this is a feature of the Feistel structure which makes encryption and decryption similar processes.

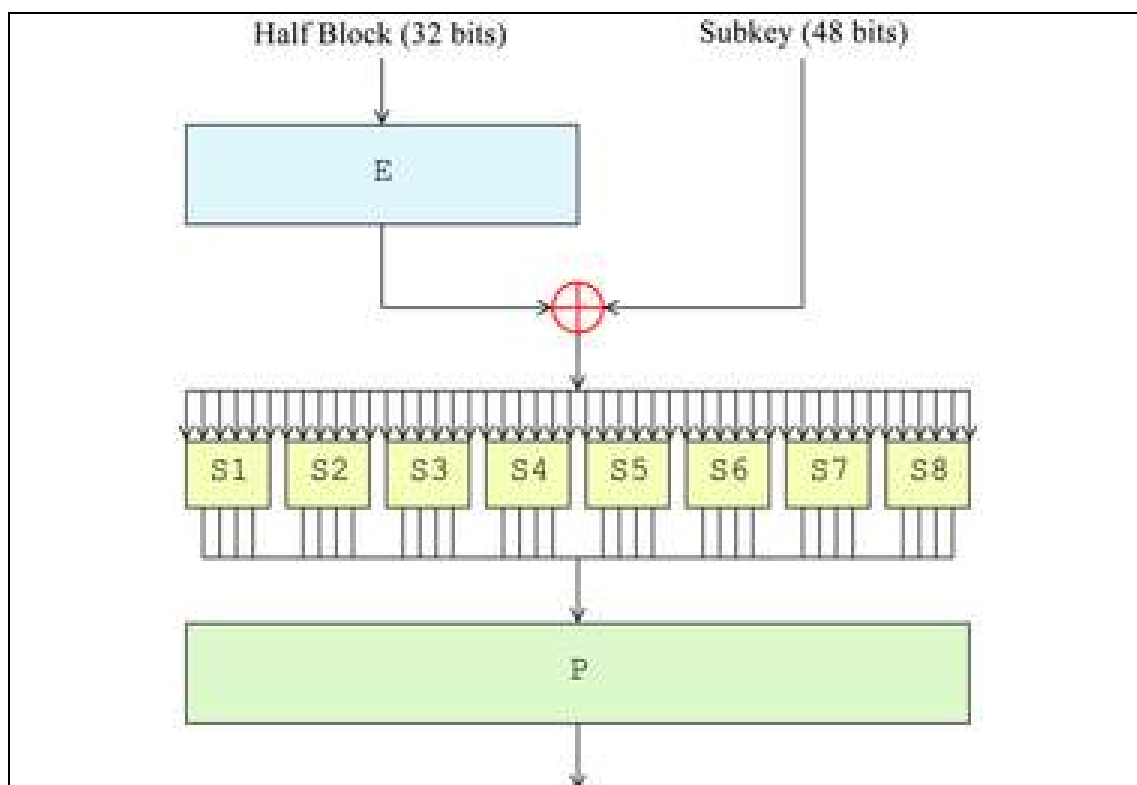


Figure 9-7. Illustration of the Feistel function (*F-function*) of DES

i. Feistel (F) Function

The F-function, depicted in figure 9-7, operates on half a block (32 bits) at a time and consists of four stages:

1. **Expansion**: the 32-bit half-block is expanded to 49-bits using the *expansion permutation*, denoted *E* in the diagram, by duplicating some of the bits.
2. **Key mixing**: the result is combined with a *subkey* using an XOR operation. Sixteen 49-bit subkeys — one for each round — are derived from the main key using the *key schedule* (described below).

3. **Substitution**: after mixing in the subkey, the block is divided into eight 6-bit pieces before processing by the *S-boxes*, or *substitution boxes*. Each of the eight S-boxes replaces its six input bits with four output bits according to a non-linear transformation, provided in the form of a lookup table. The S-boxes provide the core of the security of DES, without them, the cipher would be linear, and trivially breakable.
4. **Permutation**: finally, the 32 outputs from the S-boxes are rearranged according to a fixed permutation, the *P-box*. The alternation of substitution from the S-boxes, and permutation of bits from the P-box and E-expansion provides so-called "confusion and diffusion" respectively, a concept identified by Claude Shannon in the 1940s as a necessary condition for a secure yet practical cipher.

ii. Key Schedule

Figure 9-8 illustrates the *key schedule* for encryption — the algorithm which generates the subkeys. Initially, 56 bits of the key are selected from the initial 64 by *Permuted Choice 1 (PC-1)* — the remaining eight bits are either discarded or used as parity check bits. The 56 bits are then divided into two 29-bit halves; each half is thereafter treated separately.

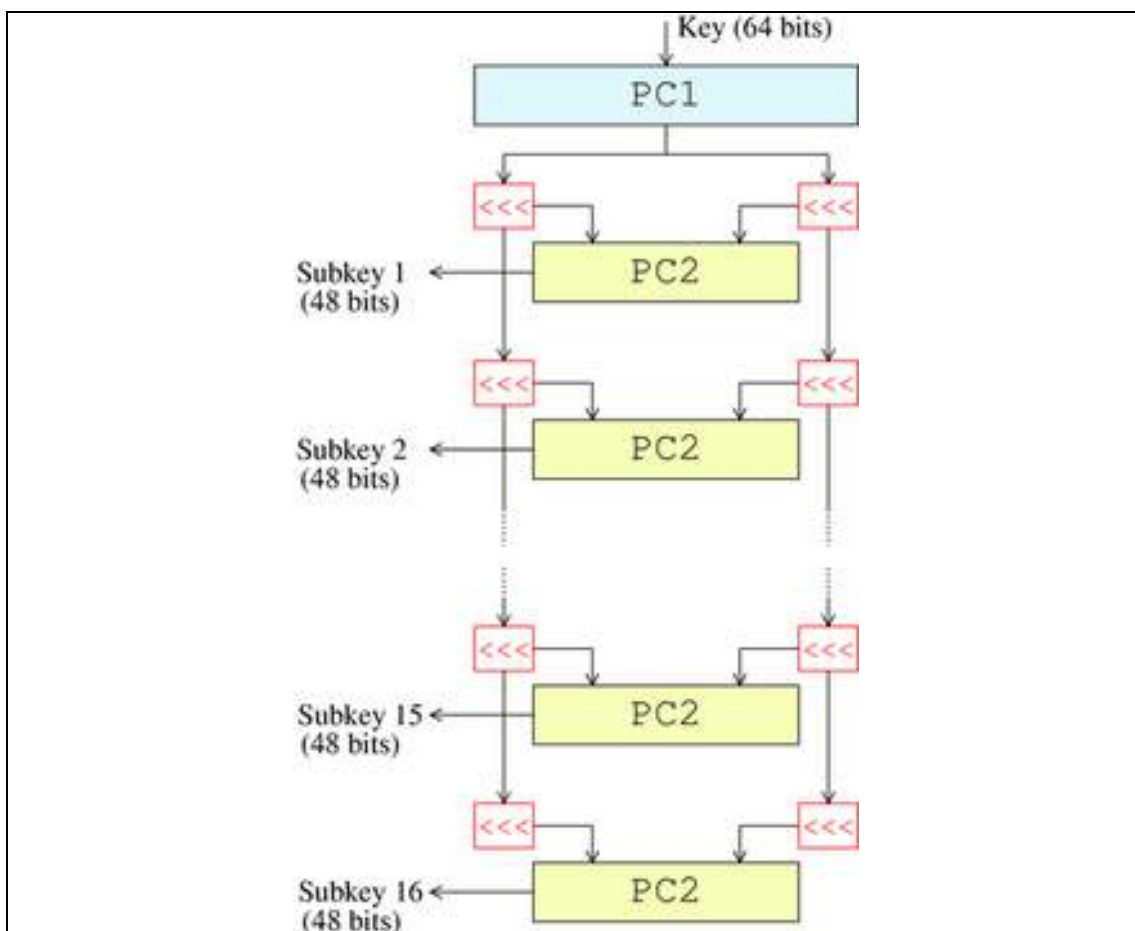


Figure 9-8. Illustration of the key-schedule of DES

In successive rounds, both halves are rotated left by one or two bits (specified for each round), and then 48 subkey bits are selected by *Permuted Choice 2 (PC-2)* — 24 bits from the left half, and 24 from the right. The rotations (denoted by "<<<" in the diagram) mean that a different set of bits is used in each subkey; each bit is used in approximately 14 out of the 16 subkeys. The key schedule for decryption is similar — the subkeys are in reverse order compared to encryption. Apart from that change, the process is the same as for encryption.

9-2.2. AES Algorithm

The Advanced Encryption Standard (**AES**) is a specification for the encryption of electronic data. AES is based on the **Rijndael** cipher developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, who submitted a proposal to NIST during the AES selection process. Rijndael is a family of ciphers with different key and block sizes. AES is based on a design principle known as a **substitution-permutation network**, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. However, AES is a variant of the original Rijndael algorithm which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input (plaintext), into the final output (ciphertext). The following listing describes a simple software implementation of the AES cryptographic algorithm in C-language

```
#define LENGTH 16 /* # bytes in data block or key */
#define NROWS 4 /* number of rows in state */
#define NCOLS 4 /* number of columns in state */
#define ROUNDS 10 /* number of iterations */
typedef unsigned char byte; /* unsigned 9-bit integer */
rijndael(byte plaintext[LENGTH], byte ciphertext[LENGTH], byte key[LENGTH])
{
    int r; /* loop index */
    byte state[NROWS][NCOLS]; /* current state */
    struct {byte k[NROWS][NCOLS];} rk[ROUNDS + 1]; /* round keys */
    expand_key(key, rk); /* construct the round keys */
    copy_plaintext_to_state(state, plaintext); /* init current state */
    xor_roundkey_into_state(state, rk[0]); /* XOR key into state */
    for (r = 1; r <= ROUNDS; r++) {
        substitute(state); /* apply S-box to each byte */
        rotate_rows(state); /* rotate row i by i bytes */
        if (r < ROUNDS) mix_columns(state); /* mix function */
        xor_roundkey_into_state(state, rk[r]); /* XOR key into state */
    }
    copy_state_to_ciphertext(ciphertext, state); /* return result */
}
```

As shown in the following figure, each regular round in the algorithm involves **four** steps.

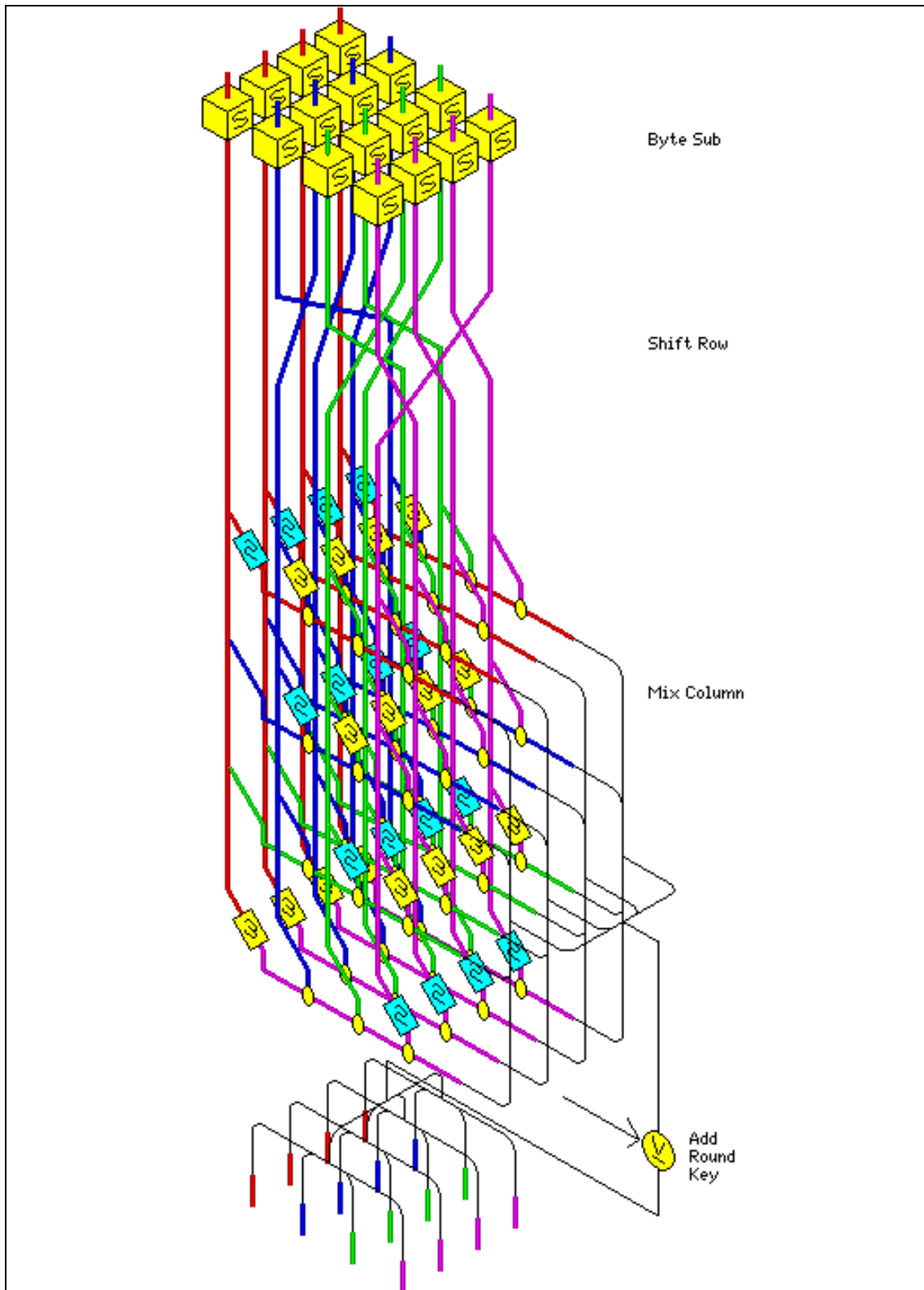


Figure 9-9(a). Illustration of the AES algorithm

First step is the **Byte Sub**, where each byte of the block is replaced by its substitute in an S-box. Next is the **Shift Row** step. Considering the block to be made up of bytes 1 to 16, these bytes are arranged in a rectangle, and shifted as follows:

from	to
1 5 9 13	1 5 9 13
2 6 10 14	6 10 14 2
3 7 11 15	11 15 3 7
4 8 12 16	16 4 8 12

Next step is the **Mix Column**. Matrix multiplication is performed: each column, in the arrangement we have seen above, is multiplied by the matrix. The final step is **Add Round Key**. This is simply XOR the current round in the *subkey*.

Although the three-dimensional color diagram of the round has already appeared at the beginning of the page, the **Rijndael** round can also be illustrated in the following 2-dimensional representation:

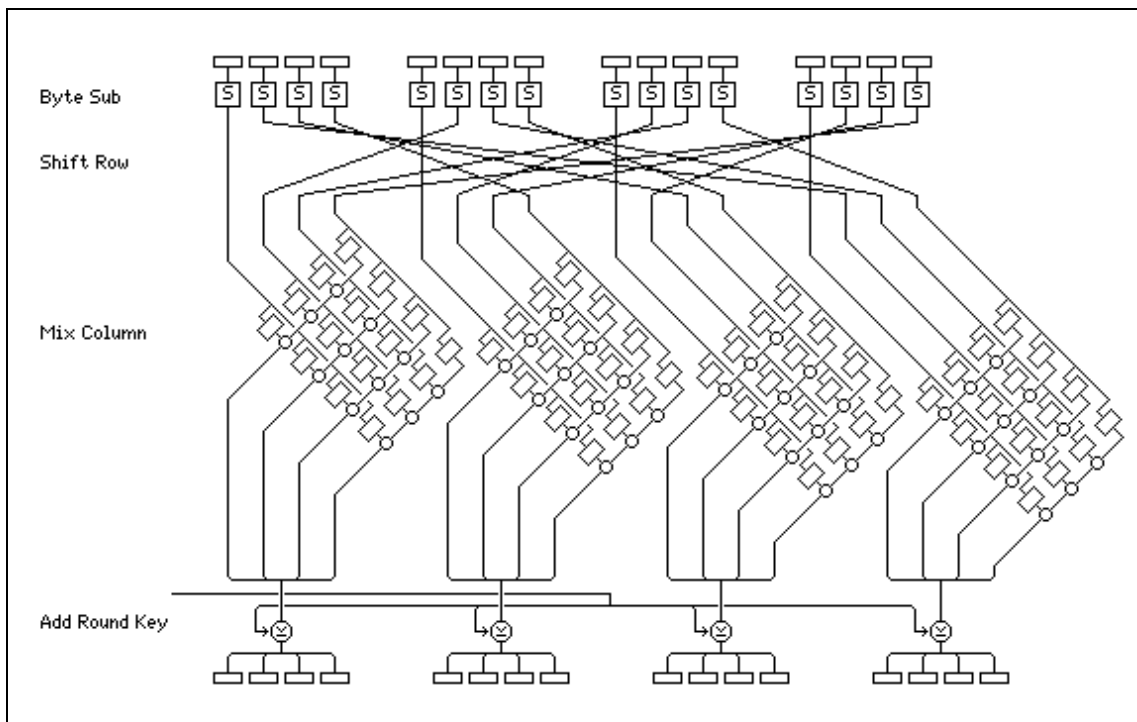


Figure 9-9(b). Illustration of the AES algorithm, in 2-dimensions

9-3. Asymmetric-Key (Two-Level Public Key) Encryption

The encryption methods, that we described so far, are not perfect—a fact highlighted by the numerous data security breaches that have occurred over years. Technological limitations in the "trusted server" model for encryption have hindered the robust protection of data.

Until the 1970s, all encryption was **symmetric**, so that anyone who knew how to encrypt a message could work out how to decrypt it. This was adequate for communication between a small number of trusted people sharing a secret encryption key. However, in a situation where large numbers of people want to communicate securely (like Internet Commerce) it is impossible for everyone to share such a 'secret' key.

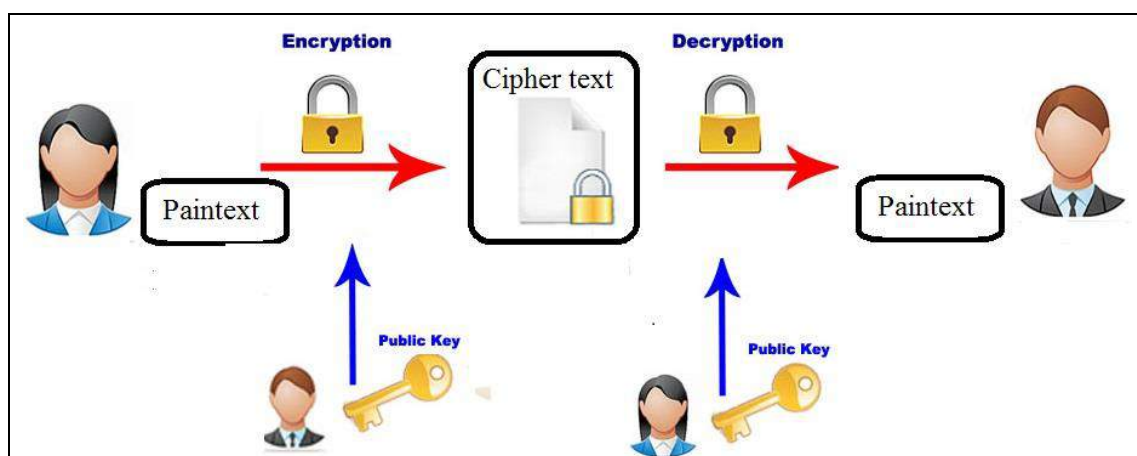


Figure 9-10. Schematic representation of the asymmetric-key encryption.

In 1977 the asymmetric or **public key algorithm** was introduced by the American mathematicians W. Diffie and M. E. Hellman. The so-called public key cryptography (**PKC**) is a very novel cryptography that involves pairs of keys: a 'public' key which can be made openly available, and a 'private' key. Once information has been encrypted with the public key, nobody but the holder of the private key can decrypt it. In reverse, if the private key is used for encryption, anyone with the public key can decrypt it. It is very hard to derive the private key from the public key. Because the private key does not need to be exchanged, PKC is much more secure than earlier techniques, so it can be used for applications such as internet commerce. Asymmetric encryption is slower than symmetric encryption, even on fast computers, so most modern encryption uses a combination of both methods.

The obvious use of public key cryptography is to allow secret communications in the absence of prior secure contact. Publish your public key. People can send you messages that no one but you can read. That's simple enough. But if other people don't really know who you are, why would they send you messages that need to be encrypted?

A big corporation could put a one-way hash or checksum of its public key on its billboards or its magazine ads, to make you feel safe about sending your credit card number to it over the Web with that public key. However, things are usually done a bit differently.

Another use of some public key algorithms, particularly RSA, is digital signatures. Suppose you encrypt a message using your private key. Then, anyone can decrypt it who knows your public key; but only you could have encrypted it in the first place. Thus, you've proved that you saw that message, and chose to encrypt it. So, one of the commonest uses of digital signatures is in key certificates. The company that wrote your web browser includes, built right into the program, the public key of a company that certifies the ownership of other public keys. When you visit a site where you want to make a credit card order, your browser can then check that the public key you will use to encrypt your credit card number really belongs to the company you think you're ordering from.

9-3.1. Public Key Infrastructure (PKI)

Public key cryptography enables communication without the necessity of sharing secret encryption keys. However there remains a significant problem: establishing whether the person publishing a public key is genuine. Certification and Registration Authorities (CAs and RAs) are an established centralized way of managing keys. CAs and RAs validate the identity of people (or companies and their websites) and issue them with Certificates which they **digitally sign** to show their endorsement of that identification. The resulting digital certificates associate a given public key with an identity. When a browser connects to a website, the digital certificate can be checked. Provided that the CA is trusted, the user can be assured that the website is genuine. **VeriSign** is an example of a large CA that provides a digital certificate service to the financial and retail sectors.

The **digital signature** of a message is based on the **hash algorithm**³ to condense an entire message into a short fingerprint. This fingerprint is encrypted with the sender's private key to produce the digital signature,

³ The hash algorithm and digital signature are explained in section 8-4 Of this Chapter

which is attached to the original message and sent to the recipient. The recipient uses the sender's public key to decrypt the digital signature. In addition, he uses the hash algorithm to calculate the fingerprint of the message itself. If the two hashes are identical, this proves that the message is not tampered with and comes from the sender.

9-3.2. RSA Algorithm

In 1978, the so-called **RSA** algorithm was introduced by Ron **Rivest**, Adi **Shamir**, and Leonard **Adleman**. RSA is a public key (asymmetric key) cryptography algorithm. It is used in Internet encryption and authentication systems. The RSA algorithm is actually included as part of the Web browsers from Netscape and Microsoft

The mathematical details of the algorithm used in obtaining the public and private keys are as follows. Briefly, the algorithm involves multiplying two large prime numbers⁴ and through additional operations deriving a set of two numbers that constitutes the public key and another set that is the private key. Once the keys have been developed, the original prime numbers are no longer important and can be discarded. Both the public and the private keys are needed for encryption /decryption but only the owner of a private key ever needs to know it

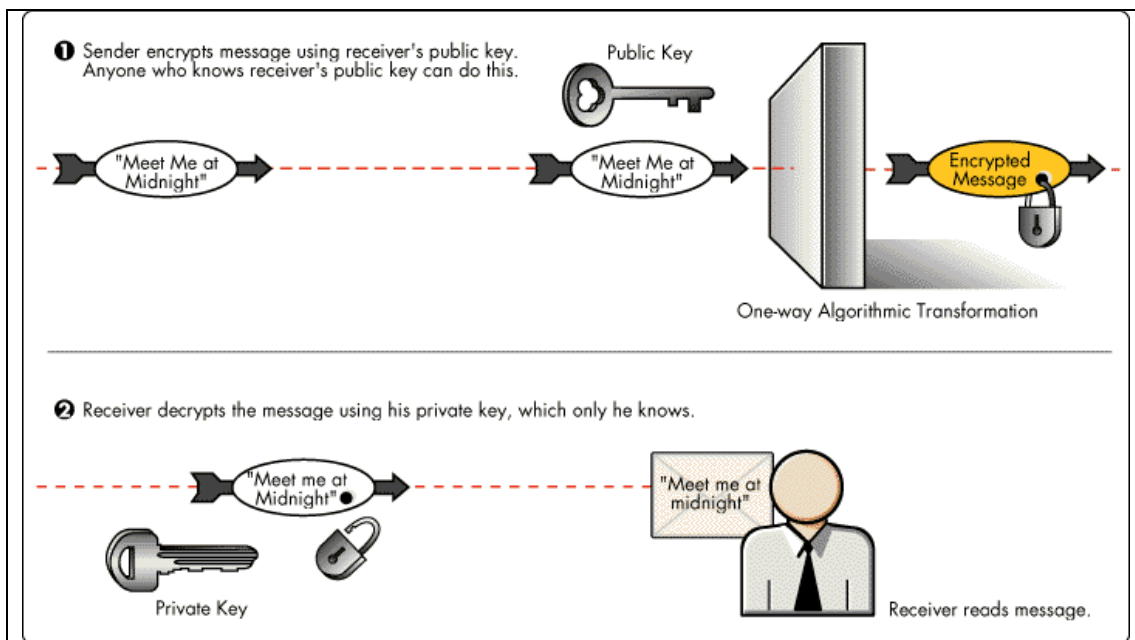


Figure 9-11. Principle of the RSA algorithm

⁴ A prime number is a number divisible only by that number and 1

Using the RSA system, the private key never needs to be sent across the Internet. The private key is used to decrypt text that has been encrypted with the public key. Thus, if I send you a message, I can find out your public key (but not your private key) from a central administrator and encrypt a message to you using your public key.

Here is an example of RSA encryption and decryption. The parameters used here are small for the matter of simplicity.

1. Choose two distinct prime numbers, such as $p = 61$ and $q = 53$
2. Compute $n = pq$ giving $n = 61 \times 53 = 3233$
3. Compute the *totient* of the product as $\varphi(n) = (p - 1)(q - 1)$ giving $\varphi(3233) = (61 - 1)(53 - 1) = 3120$
4. Choose any number $1 < e < 3120$ that is *coprime* to 3120. So, that e is not a divisor of 3120. Let $e = 17$
5. Compute d , the modular multiplicative inverse of $e \pmod{\varphi(n)}$ yielding $d = 2753$

The **public key** is $(n = 3233, e = 17)$. For a padded plaintext message m , the encryption function is $c(m) = m^{17} \pmod{3233}$

The **private key** is $(n = 3233, d = 2753)$. For an encrypted ciphertext c , the decryption function is $m(c) = c^{2753} \pmod{3233}$. For instance, in order to encrypt $m = 65$, we calculate $c = 65^{17} \pmod{3233} = 2790$
To decrypt $c = 2790$, we calculate $m = 2790^{2753} \pmod{3233} = 65$

Both of these calculations can be computed efficiently using the square-and-multiply algorithm for modular exponentiation. In real-life situations the primes selected would be much larger; in our example

9-3.3. EL-Gammal Algorithm

El-Gammal encryption system is an asymmetric key encryption algorithm for public-key cryptography which is based on the **Diffie-Hellman** key exchange. It was described by Taher Elgamal in 1985. The El-Gammal cryptosystem is usually used in a hybrid cryptosystem. I.e., the message itself is encrypted using a symmetric cryptosystem and ElGamal is then used to encrypt the key used for the symmetric cryptosystem. This is because asymmetric cryptosystems like Elgamal are usually slower than symmetric ones for the same level of security. The Digital Signature Algorithm is a variant of the El-Gammal signature scheme, which should not be confused with El-Gammal encryption.

9-4. Hashing Encryption

The hashing encryption creates a unique, fixed-length signature for a message or data set. Hashes are created with an algorithm, or hash function, and people commonly use them to compare sets of data. Since a hash is unique to a specific message, even minor changes to that message result in a dramatically different hash, thereby alerting a user to potential tampering.

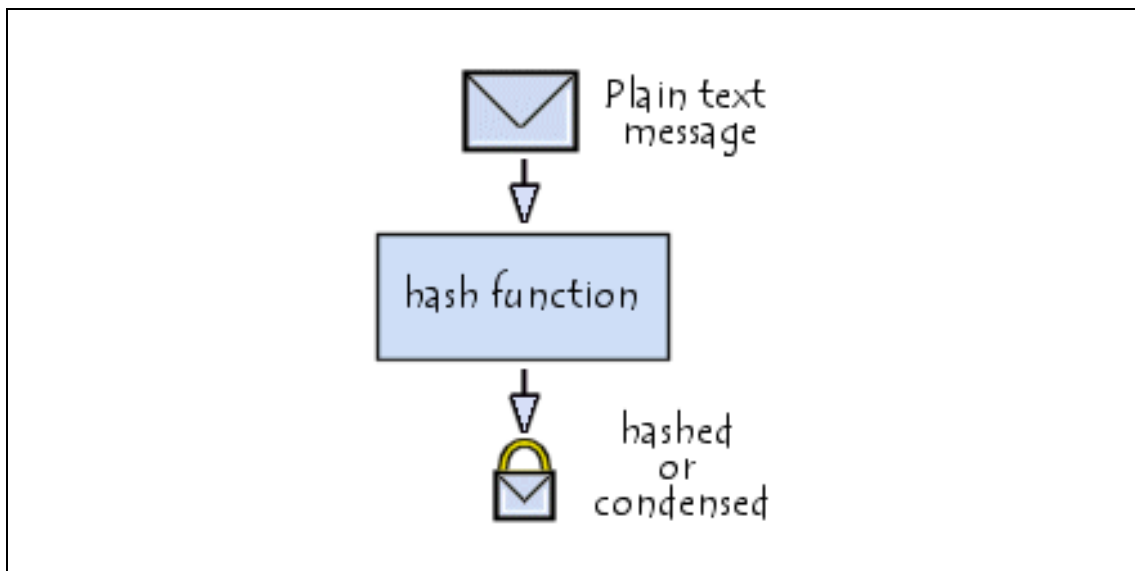


Figure 9-12. Hash algorithms

9-4.1. Difference between Hashing and Encryption

A key difference between **hashing** and the other **encryption** methods is that once the data is encrypted, the process cannot be reversed or deciphered. This means that even if a potential attacker were able to obtain a hash, he or she would not be able to use a decryption method to discover the contents of the original message. Some common hashing algorithms are Message Digest5 (**MD5**) and Secure Hashing Algorithm (**SHA**).

There are quite a few uses for hashes, and we'll mention them here so we have something to build on when we try to subvert them for evil purposes, and to see the ramifications of the recent developments. '

- Verifying file integrity
- Digitally Signed Documents

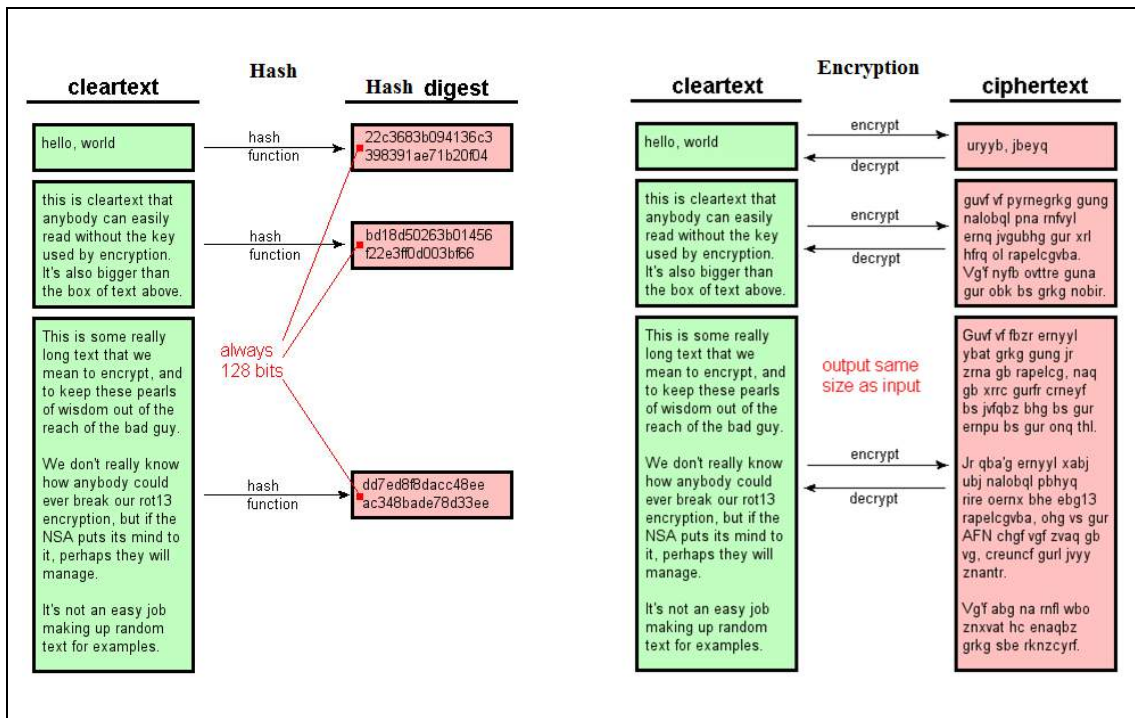


Figure 9-13. Difference between Hash digest and Encryption algorithms. Encryption is a two-way operation, Hashing is a one-way operation

9-4.2. Digital Signature

A **digital signature** is a mathematical scheme for demonstrating the authenticity of a digital message or document. Signing a document electronically is the digital equivalent of placing an autograph on paper. The question that arises here is how the signature is represented. How does one know that *this* digital signature applies to *this* document? The answer is as simple as follows: One signs (encrypts with a private key) then makes **hash** of the document, the result of which is a **digital signature**. At any later time, we can prove that he signed the document by decrypting the signature with his public key, which yields the hash, and showing that the document's hash matches the signed one. The process is shown in the following figure:

Note that an ink signature could be replicated from one document to another by copying the image manually or digitally, but to have credible signature copies that can resist some scrutiny is a significant manual or technical skill, and to produce ink signature copies that resist professional scrutiny is very difficult. On the other hand, the digital signatures bind an electronic identity to an electronic document and the digital signature cannot be copied to another document.

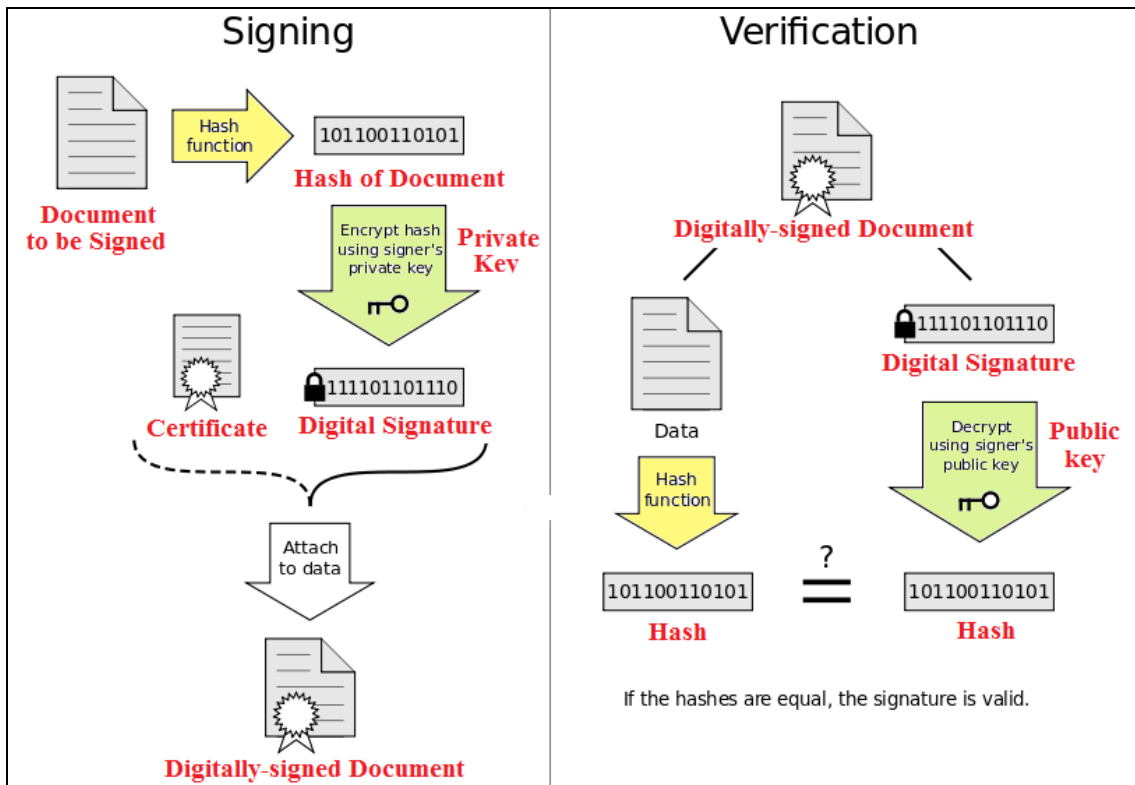


Figure 9-14. Digital signature of a document

9-4.3. SHA-1 Algorithm

SHA-1 is a cryptographic hash function designed by the United States National Security Agency and is a U.S. Federal Information Processing Standard published by the United States NIST.

SHA-1 produces a 160-bit (20-byte) hash value. A SHA-1 hash value is typically rendered as a hexadecimal number, 40 digits long. SHA stands for "secure hash algorithm". The four SHA algorithms are structured differently and are named *SHA-0*, *SHA-1*, *SHA-2*, and *SHA-3*. *SHA-0* is the original version of the 160-bit hash function published in 1993 under the name "SHA": it was not adopted by many applications. Published in 1995, *SHA-1* is very similar to *SHA-0*, but alters the original SHA hash specification to correct alleged weaknesses. *SHA-2*, published in 2001, is significantly different from the *SHA-1* hash function. *SHA-1* is the most widely used of the existing SHA hash functions, and is employed in several widely used applications and protocols. In 2005, cryptanalysts found attacks on *SHA-1* suggesting that the algorithm might not be secure enough for ongoing use NIST required many applications in federal agencies to move to *SHA-2* after 2010 because of the weakness. Although no successful attacks have yet been reported on *SHA-2*, it is algorithmically similar to *SHA-1*. In 2012, following a long-running

competition, NIST selected an additional algorithm, Keccak, for standardization under SHA-3. In 2013 Microsoft announced their deprecation policy on SHA-1 according to which Windows will stop accepting SHA-1 certificates in SSL by 2017

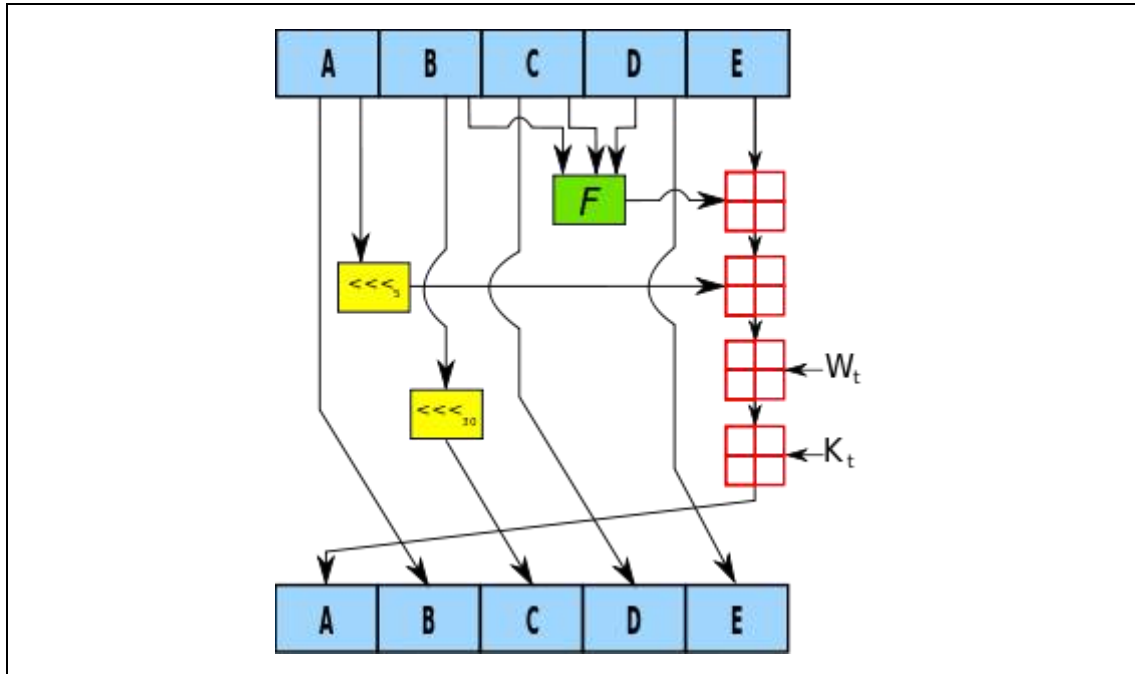


Figure 9-12. One iteration within the SHA-1 function:

A, B, C, D and E are 32-bit words of the state; F is a nonlinear function that varies;

\lll_n denotes a left bit rotation by n places; n varies for each operation;

W_t is the expanded message word of round t ; K_t is the round constant of round t ;

\boxplus denotes addition modulo 2

9-5. Security and Cryptanalysis

Although more information has been published on the cryptanalysis of DES than any other block cipher, the most practical attack to date is still a brute force approach. Various minor cryptanalytic properties are known, and three theoretical attacks are possible which, while having a theoretical complexity less than a brute force attack, require an unrealistic amount of known or chosen plaintext to carry out, and are not a concern in practice.

9-5.1. Brute Force Attack

For any cipher, the most basic method of attack is brute force — trying every possible key in turn. The length of the key determines the number of possible keys, and hence the feasibility of this approach. The vulnerability of DES was practically demonstrated in the late 1990s. The feasibility of cracking DES quickly was demonstrated in 1998 when a

custom DES-cracker was built by the Electronic Frontier Foundation (EFF). The EFF's DES cracking machine contained 1,856 custom chips and could brute force a DES key in a matter of days.

The only other confirmed DES cracker was the COPACOBANA machine built more recently by teams of the Universities of Bochum and Kiel, both in Germany. Unlike the EFF machine, COPACOBANA consist of commercially available, reconfigurable integrated circuits. 120 of these Field-programmable gate arrays (FPGAs) of type XILINX Spartan3-1000 run in parallel. They are grouped in 20 DIMM modules, each containing 6 FPGAs. The use of reconfigurable hardware makes the machine applicable to other code breaking tasks as well. The cost decrease by roughly a factor of 25 over the EFF machine is an impressive example for the continuous improvement of digital hardware.

9-5.2. Attacks Faster Than Brute-Force

There are three attacks known that can break the full sixteen rounds of DES with less complexity than a brute-force search: differential cryptanalysis (DC), linear cryptanalysis (LC), and Davies' attack. However, the attacks are theoretical and are unfeasible to mount in practice; these types of attack are sometimes termed certification weaknesses.

- **Differential cryptanalysis** was rediscovered in the late 1980s by Eli Biham and Adi Shamir; it was known earlier to both IBM and the NSA and kept secret. To break the full 16 rounds, differential cryptanalysis requires 2^{47} chosen plaintexts. DES was designed to be resistant to DC
- **Linear cryptanalysis** was discovered by Mitsuru Matsui, and needs 2^{43} known plaintexts. The method was implemented and was the first experimental cryptanalysis of DES to be reported. There is no evidence that DES was tailored to be resistant to this type of attack. A generalization of *linear cryptanalysis* (LC) was suggested in 1994, and was further refined by Biryukov et al in 2004. Their analysis suggests that multiple linear approximations could be used to reduce the data requirements of the attack by at least a factor of 4 (i.e. 2^{41} instead of 2^{43}). A similar reduction in data complexity can be obtained in a chosen-plaintext variant of linear cryptanalysis. In 2001, Junod performed several experiments to determine the actual time complexity of linear cryptanalysis, and reported that it was somewhat faster than predicted, requiring time equivalent to 2^{39} – 2^{41} DES evaluations.

- **Improved Davies' attack:** while linear and differential cryptanalysis are general techniques and can be applied to a number of schemes, Davies' attack is a specialised technique for DES, first suggested by Donald Davies in the eighties, and improved by Biham and Biryukov (1997). The most powerful form of the attack requires 2^{50} known plaintexts, has a computational complexity of 2^{50} , and has a 51% success rate. There have also been attacks proposed against reduced-round versions of the cipher, i.e. versions of DES with fewer than sixteen rounds. Such analysis gives an insight into how many rounds are needed for safety, and how much of a "security margin" the full version retains. Differential-linear cryptanalysis was proposed by Langford and Hellman in 1994, and combines differential and linear cryptanalysis into a single attack. An enhanced version of the attack can break 9-round DES with $2^{15.8}$ known plaintexts and has a $2^{29.2}$ time complexity, as demonstrated by Biham et al, in 2002.

9-5.3. Cryptanalytic Properties

DES exhibits the complementation property, namely that

$$E_K(P) = C \Leftrightarrow E_{\bar{K}}(\bar{P}) = \bar{C} \quad (9-1)$$

where \bar{x} is the bitwise complement of x . E_K denotes encryption with key K . P and C denote plaintext and ciphertext blocks respectively. The complementation property means that the work for a brute force attack could be reduced by a factor of 2 (or a single bit) under a chosen-plaintext assumption.

The DES also has four so-called *weak keys*. Encryption (E) and decryption (D) under a weak key have the same effect (see involution):

$$E_K(E_K(P)) = P \text{ or equivalently, } E_K = D_K \quad (9-2)$$

There are also six pairs of *semi-weak keys*. Encryption with one of the pair of semiweak keys, K_1 , operates identically to decryption with the other, K_2 :

$$E_{K_1}(E_{K_2}(P)) = P \text{ or equivalently, } E_{K_2} = D_{K_1}. \quad (9-3)$$

It is easy enough to avoid the weak and semi-weak keys in an implementation, either by testing for them explicitly, or simply by choosing keys randomly; the odds of picking a weak or semi-weak key by chance are negligible. The keys are not really any weaker than any other keys anyway, as they do not give an attack any advantage.

DES has also been proved not to be a group, or more precisely, the set $\{E_K\}$ (for all possible keys K) under functional composition is not a group, nor "close" to being a group (Campbell and Wiener, 1992). This was an open question for some time, and if it had been the case, it would have been possible to break DES, and multiple encryption modes such as Triple DES would not increase the security. It is known that the maximum cryptographic security of DES is limited to about 64 bits, even when independently choosing all round **subkeys** instead of deriving them from a key, which would otherwise permit a security of 768 bits.

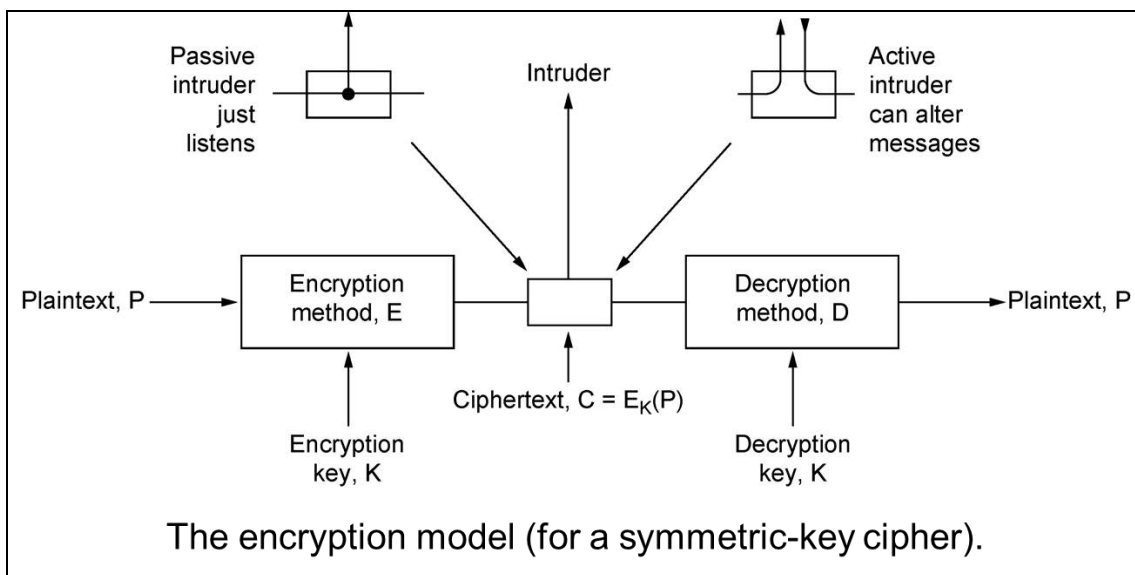


Figure 9-14. Encryption model (for a symmetric-key cipher).

9-6. Applications of Encryption

Data encryption has many applications in so many diversified fields, among them one cite the following uses.

- **Securing networks.**
- **Wireless networks**
- **Access control**
- **Private use.**

In the following subsections, we discuss some details of each application

9-6.1. Securing Networks

With the rapid growth of the Internet, network security has become a major concern to companies all over the world. The fact that the information and tools needed to penetrate the security of corporate networks are widely available has increased that concern. In fact, network administrators often spend more effort protecting their networks than on actual network setup and administration. These statements foreshadow the key questions behind most security issues on securing networks in general and the the Internet-enabled networks, in particular:

- How do you protect confidential information from those who do not explicitly need to access it?
- How do you protect your network and its resources from malicious users and accidents that originate outside your network?

Confidential information can reside in two states on a network. It can reside on physical storage media, such as a hard drive or memory, or it can reside in transit across the physical network wire in the form of packets. These two information states present multiple opportunities for attacks from users on your internal network, as well as those users on the Internet. We are primarily concerned with the second state, which involves network security issues. The following are five common methods of attack that present opportunities to compromise the information on your network:

- Network packet sniffers
- IP spoofing
- Password attacks
- Distribution of sensitive internal information to external sources
- Man-in-the-middle attacks

When protecting your information from these attacks, your concern is to prevent the theft, destruction, corruption, and introduction of information that can cause irreparable damage to sensitive and confidential data. This section describes these common methods of attack and provides examples of how your information can be compromised.

Password attacks can be implemented using several different methods, including brute-force attacks, Trojan horse programs (discussed later in the chapter), IP spoofing, and packet sniffers. Although packet sniffers and IP spoofing can yield user accounts and passwords, password attacks usually refer to repeated attempts to identify a user account and/or password; these repeated attempts are called brute-force attacks.

Often, a **brute-force attack** is performed using a dictionary program that runs across the network and attempts to log in to a shared resource, such as a server. When an attacker successfully gains access to a resource, that person has the same rights as the user whose account has been compromised to gain access to that resource. If this account has sufficient privileges, the attacker can create a back door for future access, without concern for any status and password changes to the compromised user account.

Application layer attacks can be implemented using several different methods. One of the most common methods is exploiting well-known weaknesses in software commonly found on servers, such as sendmail, PostScript, and FTP. By exploiting these weaknesses, attackers can gain access to a computer with the permissions of the account running the application, which is usually a privileged system-level account.

Trojan horse attacks are implemented using bogus programs that an attacker substitutes for common programs. These programs may provide all the functionality that the normal application or service provides, but they also include other features that are known to the attacker, such as monitoring login attempts to capture user account and password information. These programs can capture sensitive information and distribute it back to the attacker. They can also modify application functionality, such as applying a blind carbon copy to all e-mail messages so that the attacker can read all of your organization's e-mail.

One of the oldest forms of application layer attacks is a Trojan horse program that displays a screen, banner, or prompt that the user believes is the valid login sequence. The program then captures the information that the user types in and stores or e-mails it to the attacker. Next, the program

either forwards the information to the normal login process (normally impossible on modern systems) or simply sends an expected error to the user (for example, Bad Username/Password Combination), exits, and starts the normal login sequence. The user, believing that he or she has incorrectly entered the password, retypes the information and allows access for the intruder.

One of the newest forms of application layer attacks exploits the openness of several new technologies: the HyperText Markup Language (HTML) specification, web browser functionality, and HTTP. These attacks, which include Java applets and ActiveX controls, involve passing harmful programs across the network and loading them through a user's browser. Users of Active X controls may be lulled into a false sense of security by the Authenticode technology promoted by Microsoft. However, attackers have already discovered how to utilize properly signed and bug-free Active X controls to make them act as Trojan horses. This technique uses VBScript to direct the controls to perform their dirty work, such as overwriting files and executing other programs. These new forms of attack are different in two respects

- They are initiated not by the attacker, but by the user, who selects the HTML page that contains the harmful applet or script stored using the <OBJECT>, <APPLET>, or <SCRIPT> tags.
- Their attacks are no longer restricted to certain hardware platforms and operating systems because of the portability of the programming languages involved.

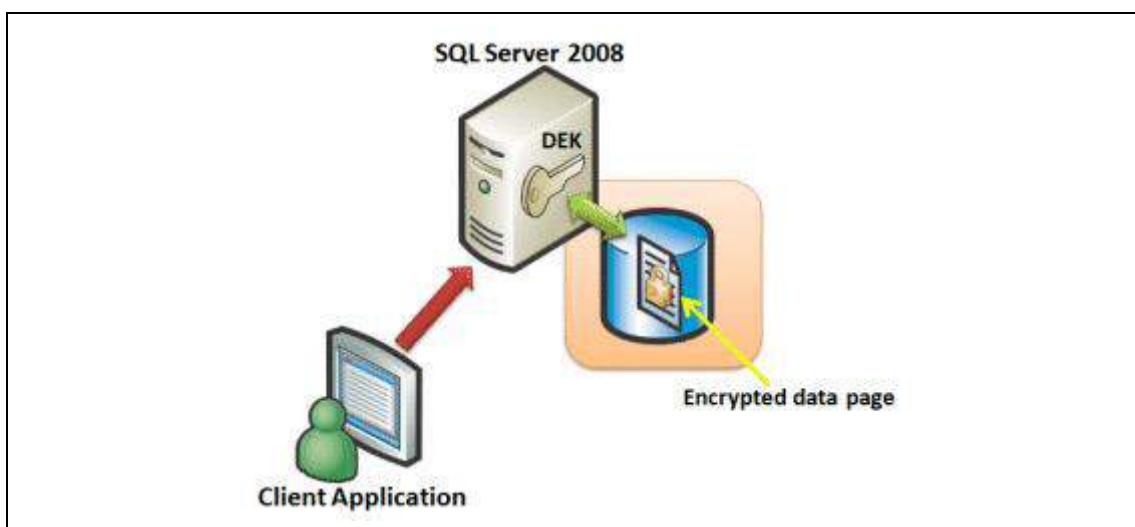


Figure 9-16. Transparent encryption in a databases.

When you define a network security policy, you must define procedures to safeguard your network and its contents and users against loss and damage. From this perspective, a network security policy plays a role in enforcing the overall security policy defined by an organization. A critical part of an overall security solution is a **network firewall**, which monitors traffic crossing network perimeters and imposes restrictions according to security policy. Perimeter routers are found at any network boundary, such as between private networks, intranets, extranets, or the Internet. Firewalls most commonly separate internal (private) and external (public) networks.

9-6.2. Secure Sockets Layer (SSL)

The Secure Sockets Layer (SSL) is an encryption protocol that enables secure communications and user authentication over open, unsecured networks like the **Internet**. Its use is usually indicated in web browsers by a small padlock icon, seen for example when a user submits credit card details online. Besides protecting data, this system also checks that a given website is authentic and sometimes verifies the identity of the user. Similar protocols are used to secure private networks

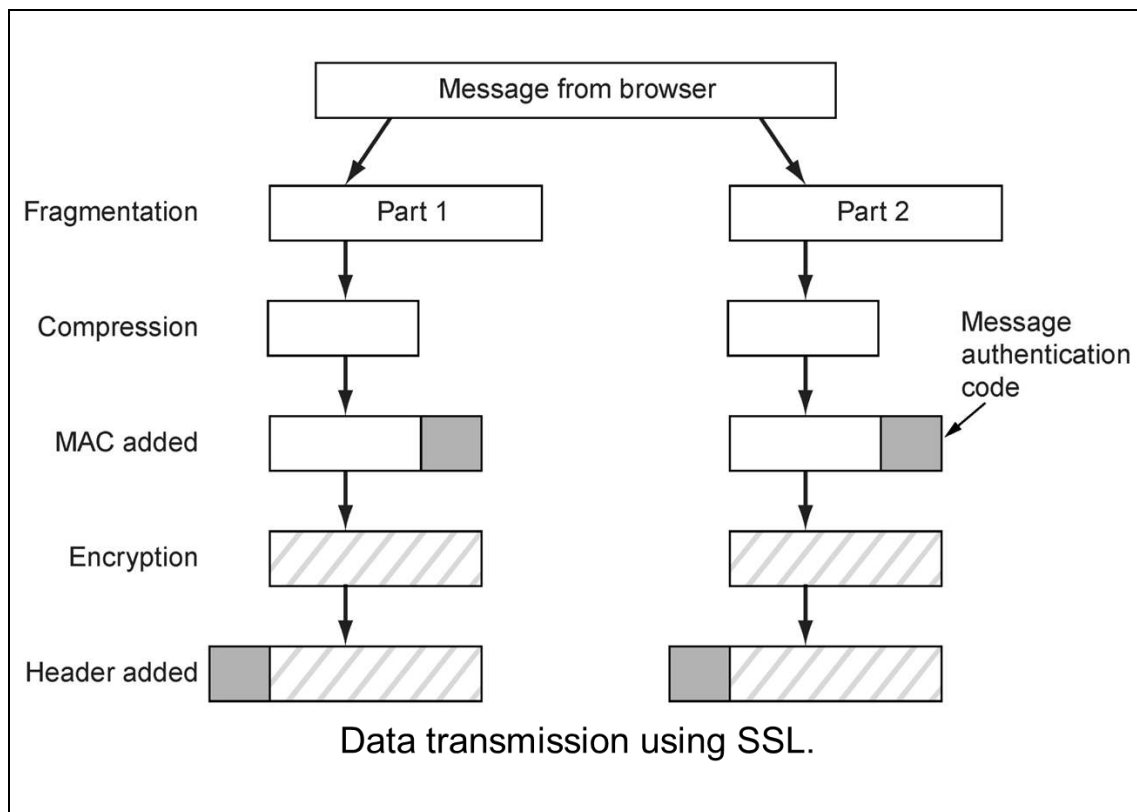


Figure 9-17. Illustration of the Secure Sockets Layer (SSL) encryption protocol

9-6.3. Wireless Networks

When you connect your wireless network to the Internet, you are actually connecting your network to thousands of unknown networks and all their users. Wireless networks, such as WLAN 802.11 standards (Wi-Fi) are vulnerable to interception. Several protocols, with different encryption mechanisms, have been developed to secure wireless networks.

i. Wired Equivalent Privacy (WEP)

The first wireless security protocol is called Wired Equivalent Privacy (WEP). WEP provides security by **encrypting** data sent over radio waves from end point to end point. Wired Equivalent Privacy (WEP) is intended to stop the interference of radio frequency that is signaled by unauthorized users and this security measure is most suitable for the small networks. There is not key management protocol and each key is entered manually into the clients that's why this is very time consuming administrative task. The WEP security method is based on the RC4 encryption algorithm. In the WEP all the client computers and Access points are configured with the same encryption and decryption keys.

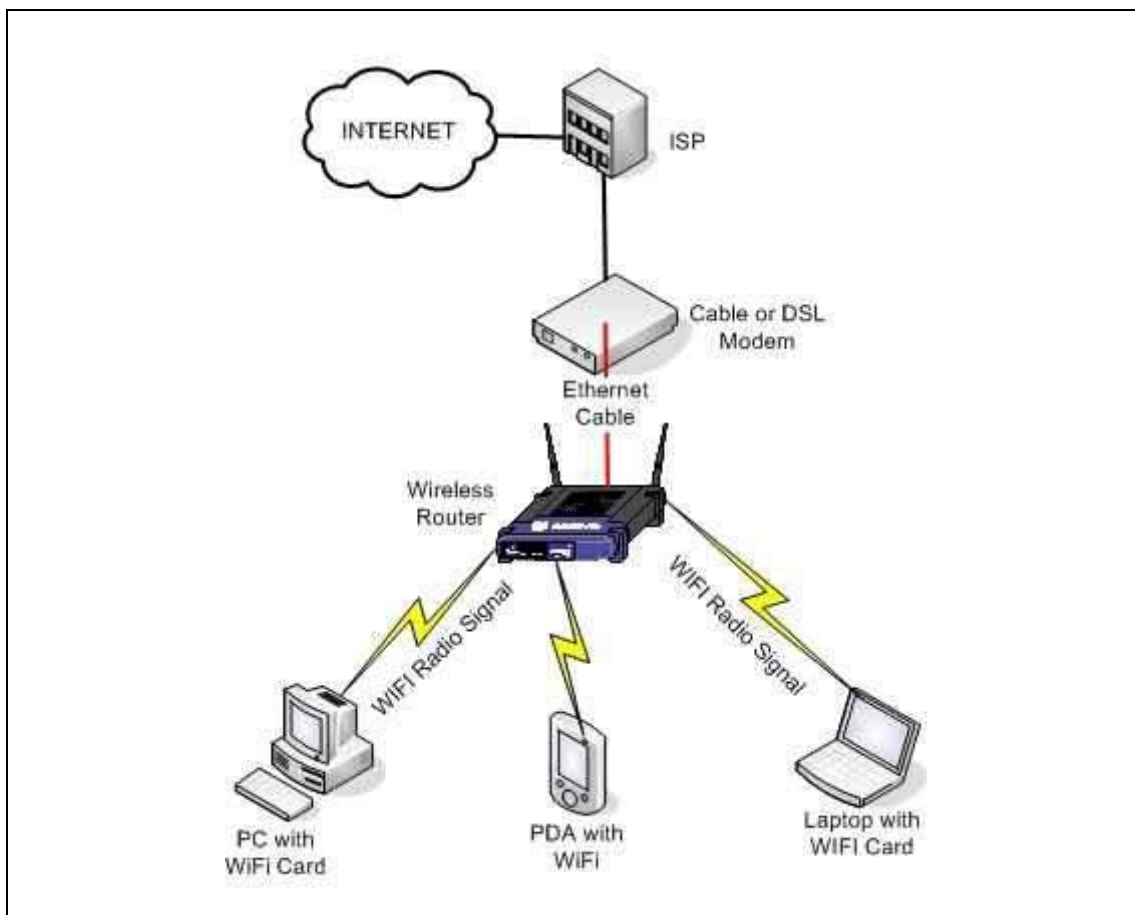


Figure 9-18. Example of a private wireless network, which is connected to the Internet

ii. Wi-Fi Protected Access (WPA)

The second WLAN security protocol is Wi-Fi Protected Access (**WPA**). WPA was developed as an upgrade to WEP. It works with existing WEP-enabled products but provides two key improvements: improved data encryption through the temporal key integrity protocol (TKIP) which scrambles the keys using a **hash algorithm**. It has means for integrity-checking to ensure that keys have not been tampered with. WPA also provides user authentication with the extensible authentication protocol (**EAP**).

iii. Wi-Fi Protected Access 2 (WPA2)

The Wi-Fi Protected Access 2 (WPA2) is the second generation of WPA encryption security. It provides Wi-Fi users with a high level of assurance that only admits authorized users to access their wireless networks. WPA2 is based on the IEEE 802.11i amendment to the 802.11 standard and is eligible for the encryption standard FIPS 140-2 compliance.

9-6.4. Access Control

With crime rates on the rise, it is more important than ever to protect your sites with a comprehensive building access system. In July, 2200 homes in Lexia, Australia were without power when West Power Company's mission-critical equipment was damaged by vandals. Later that month, 100,000 customers in Wisconsin were without cable when Charter Communications had network equipment maliciously damaged. While physical keys can be copied very easily, duplicating electronic keys requires a much higher degree of sophistication. This makes your access system much more secure than it could ever be with physical keys. The following figure shows the block diagram of an access control system, with electronic keys. An electronic user database means that you never have to change locks at your sites. If a keycard is ever lost, it can be immediately removed from the database and a new one can be issued. Electronic access control gives you the ability to set user-level access rights all the way down to individual doors and times. This minimizes your exposure to risk by granting no more site access than is necessary.

Digital television providers also control subscriber access by encrypting audio and video signals. Subscribers are equipped with a descrambling device comprising the decryption algorithm and decryption key, which together decrypt pictures and sound

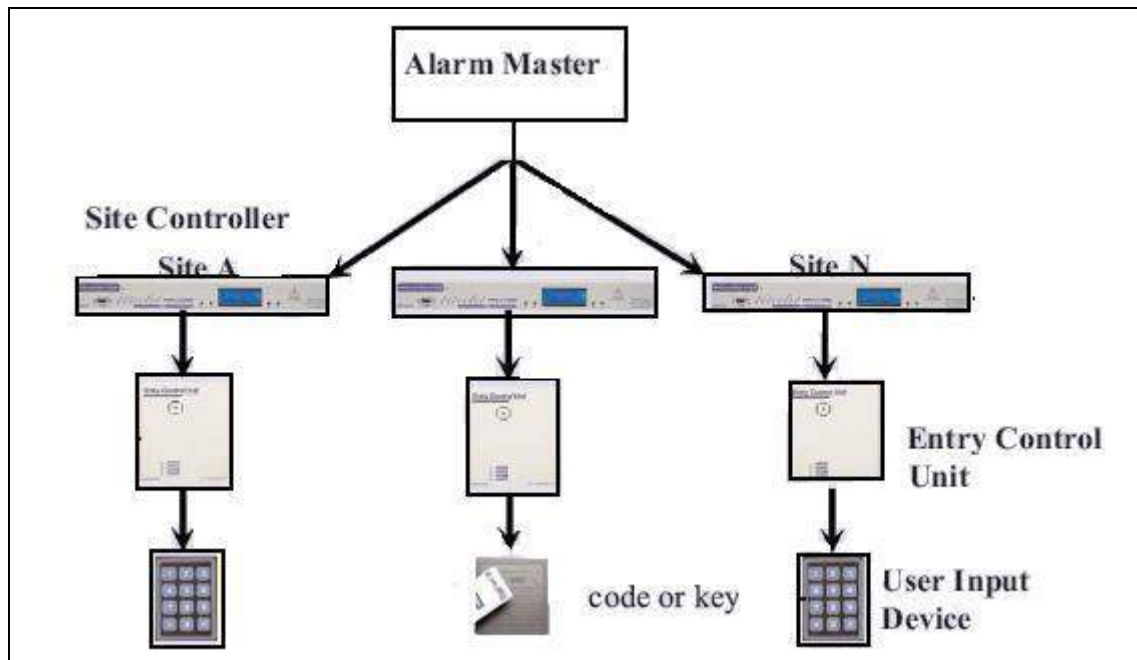


Figure 9-11. Example of an access control system

9-6.5. Private Use

Encryption software packages are readily available commercially and by free download from the Internet. One free e-mail encryption package is Pretty Good Privacy (PGP), available since 1992. An alternative is S/MIME, supported by most e-mail vendors.

9-7. Future Developments

Commercially available encryption tools are becoming more sophisticated. Microsoft launched its computer operating system, Windows Vista, in 2007. Two versions incorporated 'BitLocker Drive Encryption' which enables the entire contents of a hard drive to be encrypted. This makes data inaccessible to unauthorized users who do not have the decryption key. It can also help to identify whether a computer has been tampered with. The aim is to limit disclosure of sensitive data if computer devices are lost or stolen. However, some say that widespread availability of this and other encryption products will frustrate criminal investigations.

9-8. Summary

Cryptography is the study of **secret** (crypto-) **writing** (-graphy). The data encryption is the process of scrambling stored or transmitted information so that it is unintelligible until it is unscrambled by the intended recipient. The following table shows some of the potential security threats that may happen by some people, and why they do it. In this chapter we presented the basic methods of sending secret messages ranging from simple substitution systems performed by hand to today's advanced block ciphers, such as Rijndael, the cipher chosen as the Advanced Encryption System (AES) and public-key methods.

Adversary	Goal
Student	To have fun snooping on people's email
Cracker	To test out someone's security system; steal data
Sales rep	To claim to represent all of Europe, not just Andorra
Corporation	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by email
Identity thief	To steal credit card numbers for sale
Government	To learn an enemy's military or industrial secrets
Terrorist	To steal biological warfare secrets

Encrypting and decrypting data is nothing more than passing the data through an *algorithm* to make something readable only to the intended recipients. The process for encryption is essentially identical to the process for decryption. At the document level, *encryption* takes an easily read *plaintext* file and turns it into *ciphertext* using a *key* in conjunction with a specific algorithm. Encryption has become important not only in regard to e-mail, but also for network communications.

1. **Key:** A method of opening an encryption. A key can be as simple as a string of text characters, or a series of hexadecimal digits.
2. **Ciphertext :** Text which has been encrypted by some encryption system.
3. **Algorithm:** A computable set of steps to achieve a desired result.
4. **Plaintext:** A message before encryption or after decryption, i.e. in its usual form which anyone can read, as opposed to its encrypted form, ciphertext.

The following table the types of encryption algorithms

Encryption type	Description	Common Uses
Symmetric	Uses a single key to encrypt and decrypt data	Used for encrypting large amounts of data
Asymmetric	Uses a mathematically related public/private key pair. Also known as public-key encryption	Email
One-way	Uses a hash function to create encrypted data that cannot be decrypted	Credit card encoding, message digests
Applied	Uses a combination of encryption types for the most secure data	S/MIME protocol, SSL protocol, Secure HTTP, payment transactions

The following table shows some symmetric-key encryption algorithms and their features.

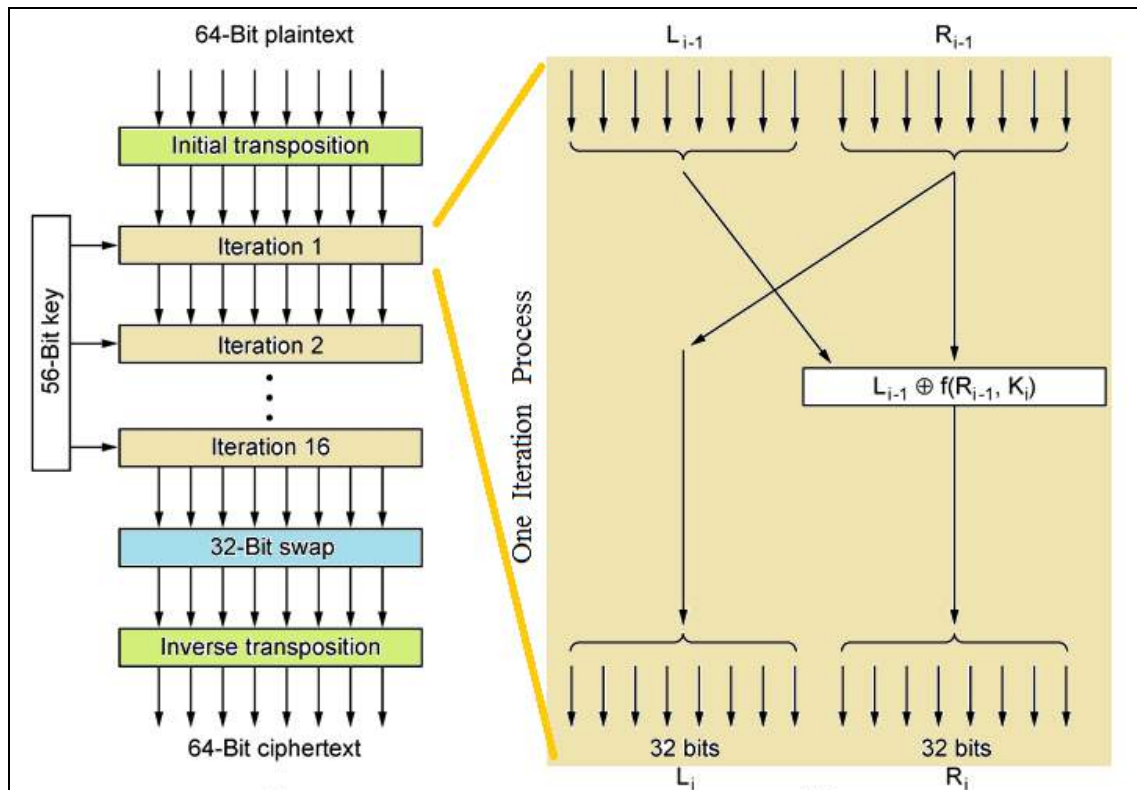
Cipher	Author	Key length	Comments
DES	IBM	56 bits	Weak & old
RC4	Ronald Rivest	1-2048 bits	some keys weak
RC5	Ronald Rivest	129-256 bits	Good, patented
AES (Rijndael)	Daemen , Rijmen	129-256 bits	Best choice
Triple DES	IBM	168 bits	Good, old
Twofish	Bruce Schneier	129-256 bits	Very strong;

Symmetric, or single-key, encryption is a simple process that is fast and strong. However, all parties must know and trust each other completely, and have confidential copies of the key. *Hackers* can compromise symmetric keys either with a *dictionary program* , *password sniffing* , or by simply snooping through a desk, purse, or briefcase. One countermeasure is to change your *key* regularly. This can reduce the

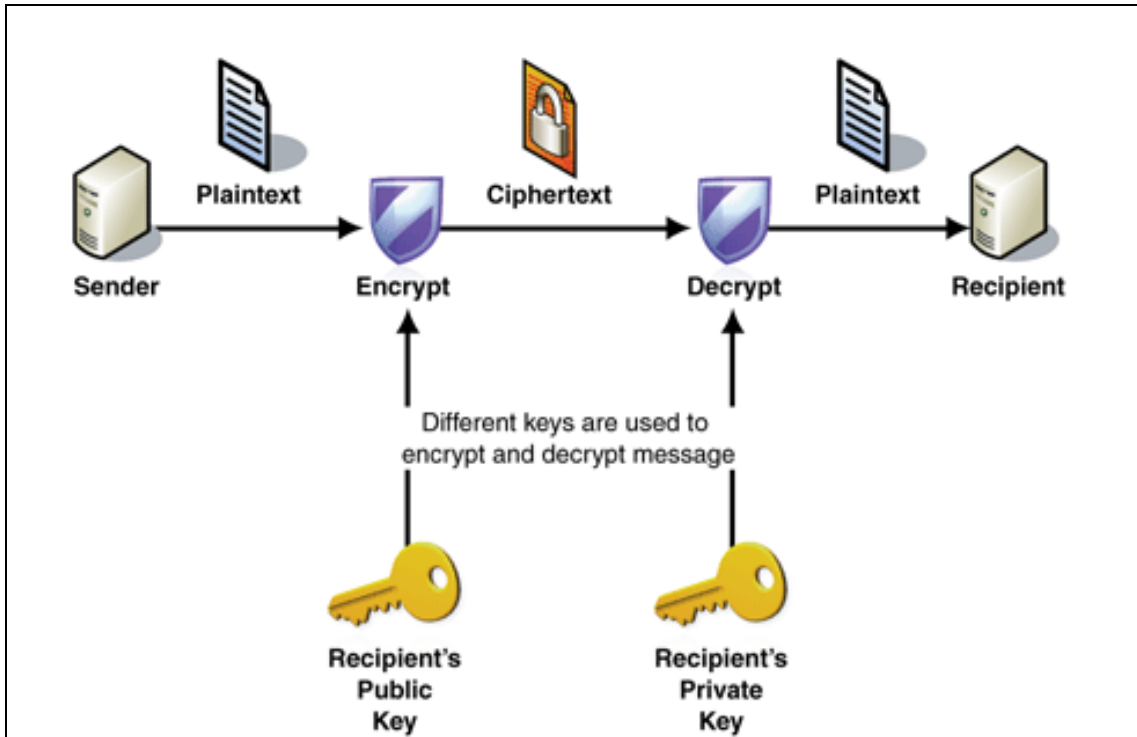
danger of having a symmetric key compromised. Encryption strength is based on three primary factors:

1. The first is strength of the algorithm
2. The second factor is the secrecy of the key
3. Length of the key

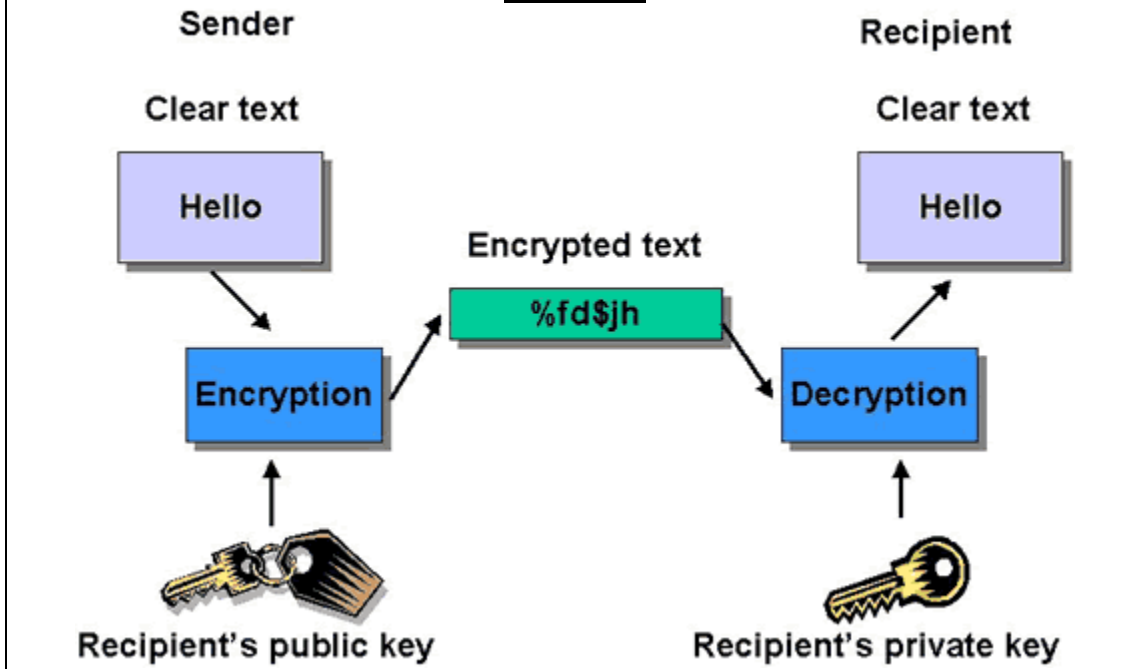
The next figure describes the **DES** cryptographic algorithm and the details of its internal iterations.



The asymmetric or public key algorithm was introduced in 1977 by the American mathematicians W. Diffie and M. E. Hellman. This algorithm requires two keys, an unguarded public key used to encrypt the plaintext and a guarded private key used for decryption of the ciphertext; the two keys are mathematically related but cannot be deduced from one another. The advantages of asymmetric algorithms are that compromising one of the keys is not sufficient for breaking the cipher and fewer unique keys must be generated. In 1978, the so-called **RSA** algorithm was introduced by Ron Rivest, Adi Shamir, and Leonard Adleman. RSA is an Internet encryption and authentication system. The RSA algorithm is the most commonly used encryption and authentication algorithm and is included as part of the Web browsers from Microsoft and Netscape



Example



9-9. Problems

9-1) Name three common network attacks used to undermine network security.

9-2) Describe the DES operation in 10 points.

9-3) What's meant by the 2-key encryption? And how it can be use to encrypt data.

9-4) What's the main difference between functional encryption and the 2-key encryption

9-5) What tools and applications are available to help monitor and test for system and network vulnerabilities?

9-10. References

- [1] R. **Rivest**, A. **Shamir**, and L. **Adleman**, "A method for obtaining digital signatures and public key cryptosystems," Commun. ACM, vol. 21, no. 2, pp. 120-126, Feb. **1978**.
- [2] W **Diffie**, M E **Hellman**, "Privacy and Authentication: An Introduction to Cryptography", in Proc. IEEE, Vol. 67, No.3, pp.397-427, Mar **1979**.
- [3] Taher **EL-Gamal**, Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Trans. Information Theory, Vol. it31, No.4, **1985**.
- [4] Coggins **Cooper**, et al. Implementing Internet Security. Indianapolis: New Riders, **1997**.
- [5] M. **Thomas** and A. Thomas Joy. Elements of information theory, 1st Edition. New York: Wiley-Interscience, 1991. 2nd Edition. New York: Wiley-Interscience, **2006**
- [6] The Home Affairs Committee, Fourth Report of Session 2005-06, *Terrorist Detention Powers* HC 910-II, **2006**

Chapter
10

Computer Communications & Protocols

Contents

- 10-1. Fundamentals of Computer Communications**
- 10-2. Resolution of Communication Conflicts (Protocols)**
- 10-3. Principles of Data Transmission**
- 10-4. Parallel Data Transmission and Parallel Ports**
 - 10-4.1. Parallel Data Link
 - 10-4.2. Handshaking in Parallel Communication
 - 10-4.3. Centronics Parallel Port
 - 10-4.4. Networked Parallel Data Transmission & IEEE-488 (GPIB)
- 10-5. Serial Data Transmission and Serial Ports**
 - 10-5.1. Parallel to Serial Conversion
 - 10-5.2. Synchronous Serial Data Communications
 - 10-5.3. Asynchronous Serial Data Communications
 - 10-5.4. Error Detection Techniques
 - 10-5.5. UARTS and USARTS
 - 10-5.6. RS-232C Standard
 - 10-5.7. USB
 - 10-5.8. Other Serial Bus Standards (Fire Wire, IrDA and)
- 10-6. Serial Digital Networks**
- 10-7. Summary**
- 10-8. Problems**
- 10-9. Bibliography**

Chapter 10

Computer Communications & Protocols

10-1. Fundamentals of Computer Communications

Computer and Communications networks are key infrastructures of the information society with high socio-economic value. Such networks contribute to the correct operations of many critical services, from healthcare to finance, scientific research, transportation, video broadcasting and entertainment.

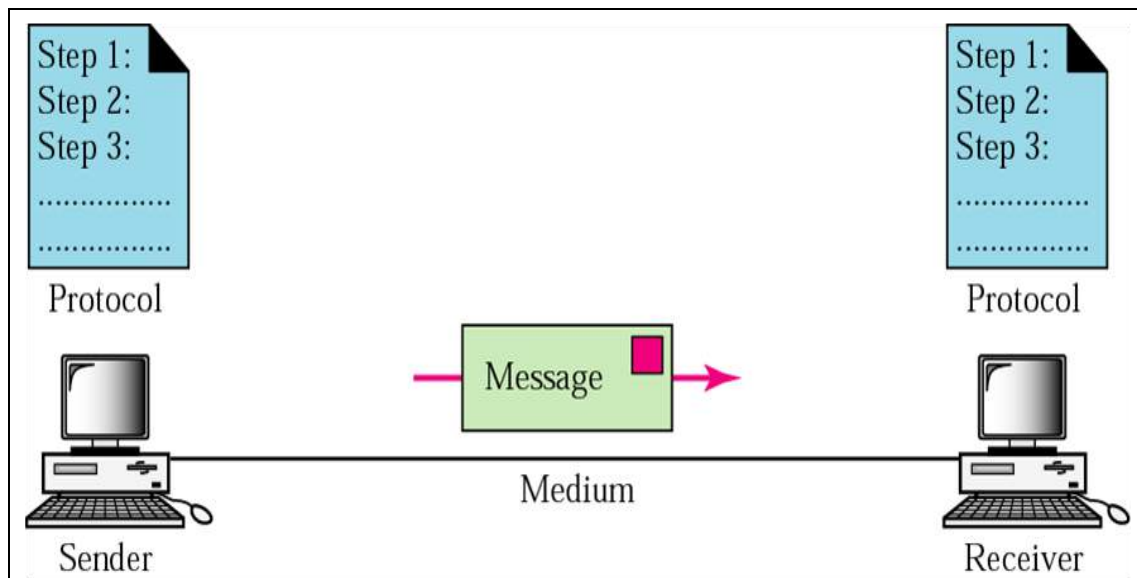


Fig. 10-1(a). Basic elements for data communications.

In order for two computers (or computer-based devices) to communicate with one another, a number of fundamental criteria must be fulfilled:

- (i) A **channel** or a physical link between the two computers or devices must be established. This is sometimes referred to as the transmission medium and may be realized through conducting cable, or through electromagnetic wave propagation at radio frequencies.
- (ii) Both devices must use the same representation for binary data on the communications link. For example, both devices may need to accept that a binary "1" is represented by voltage level "5V" and binary "0" is represented by voltage level "0V" on the link.

(iii) Both devices must use the same physical reference level with which to interpret data on the communication link.

(iv) If transmitted bit streams represent alpha-numeric characters, then both devices must interpret the characters in the same way. For example, if one device transmits bits representing ASCII characters, then the receiver must decode the bits as ASCII characters.

(v) The transmitting device must run application software that sends data through its communications port and the receiving device must actively run software that reads in data through its communications port. The software must be coordinated so that transmission does not occur until the receiving device is ready to handle incoming data.

(vi) The transmitting device must not send out data at a rate which is too high for the receiving device to handle.

(vii) There must either be some pre-defined programming of the receiving device, so that it always handles incoming data in the same way, or the instructions for the handling of data must be sent by the transmitter, with the data. For example, a printer directly outputs all characters that enter its port, unless they are identified as special strings, which command the printer to perform a special function.

(viii) If there is any possibility of the receiver being unable to handle the rate of incoming data, because of the time it has to take to process that data, then there must be some form of "hand-shaking" implemented. This should enable the receiver to signal the transmitter to stop and re-start transmission at any time, thereby preventing the loss of data.

These fundamental criteria may appear to be self-evident, but they are frequently misunderstood or not understood at all. Unless all of the above conditions are met, then there is no scope whatsoever for sensible data transfer to occur. You should therefore look upon these criteria as the 8 prerequisites of data communications. The next question to be examined is, whether or not error-free data communications can be achieved by meeting only these 8 criteria. The answer to this is no. Satisfying the above criteria will allow data communications to take place, but much more work needs to be done in order to provide links that are both bidirectional and reliable.

10-2. Resolution of Communication Conflicts (Protocols)

Internal data communication within a single computer system is very well coordinated and synchronized. Devices are specially selected and integrated into a system to perform in a unified and ordered manner. Regardless of the autonomy of any one device or chip, within a computer system, a Central Processing Unit (microprocessor or processor board) is responsible for supervising the activities of all other devices. Hence a master-slave relationship exists and provides a stable platform for co-ordination of data transfer.

Once we decide to provide data communications outside the shell of a single computer then many of the luxuries of internal communications are lost. We face an environment of multiple vendors, differing electrical interfaces and differing forms of data representation. Moreover, we often have two equally intelligent devices that cannot automatically be configured into a master to slave relationship. Within a single computer system, we have a situation where many of the devices (chips) are passive and need to be activated by the CPU in order to transmit data. Conflicts for use of the transmission media do not arise. However, when two or more computers wish to communicate, then it is the human user (system designer) who must ensure that conflicts for use of transmission media are resolved through appropriate hardware and software.

Within a single computer we generally have a sealed system, where the likelihood of fatal transmission errors is minimized by the absence of human intervention. If a device, such as a memory chip, fails within a computer then a fatal error occurs and the computer may completely lock-up. In an engineering control environment, this is probably more stable than a situation where a computer misinterprets data from an incorrectly functioning remote device and acts inappropriately upon that data. It is therefore essential that when two, or more, computerized devices communicate with one another externally, they can react to errors caused by external (remote) equipment. Typically, the sorts of errors that can arise are:

- Transmission medium can be physically broken,
- Remote computer may be switched off or inoperative,
- Transmission medium is to electromagnetic interference (EMI) and
- Two devices may attempt to transmit data at the same time.

The errors, listed above, cannot be corrected by only following the basic communications criteria outlined in section 10-1. In order for the normal

process of error detection and correction to take place between communicating devices, the software and hardware on all computer devices must be made to adhere to a **common set of rules**. These rules outline the methods for data transmission, error detection, resolution of conflicts for use of the transmission media and so on. When combined with a specification for the fundamental communications criteria of section 10-1, the total set of rules is referred to as a **communications protocol**. Needless to say, there is no universal communications protocol for computer communication. There are countless numbers of specifications and protocols, some proprietary to computer manufacturers, others evolved from historical teletype transmission techniques and still more that have been generated by various professional computer bodies around the world.

10-3. Principles of Data Transmission

As we have seen above, a communications **protocol** is the set of standard rules for data representation, signaling, authentication and error detection required to send information from one device (e.g. a Computer) to another (i.e. a Modem), over a communications channel.

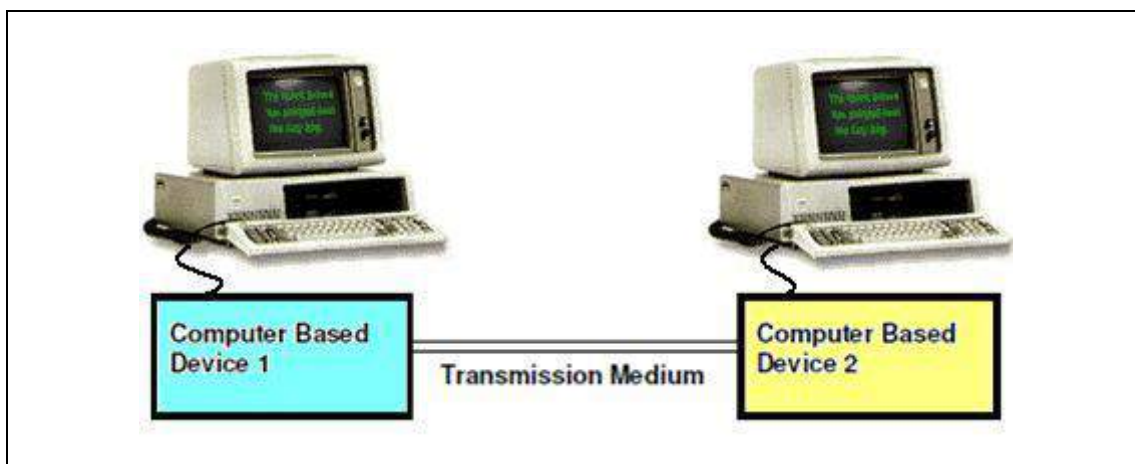


Fig. 10-1(b) Point-to-point communication link, between 2 computer-based devices.

10-3.2. Transmission Protocols

There is no particular protocol being universally better than others. Individual protocols have strengths and weaknesses, primarily because they are designed for specific applications and specific areas. For instance, the **Hypertext Transfer Protocol (HTTP)** is a software (Application Layer) protocol for distributed hypermedia information systems. We now look, qualitatively, at the way in which a protocol works in regard to a simple, one-to-one, computer to computer link (a

point to point protocol). Figure 10-2(a) shows a point-to-point communication link. Let us assume that we wish to transfer an ASCII file from device 1 to device 2. Let us also assume that the physical aspects of the link protocol have already been matched. That is, both devices satisfy common mechanical interfaces, voltages, binary data representation, etc. How does a protocol enable us to reliably transfer data between from one device to another? There are a number of phases that both devices must pass through in order to perform the common communications function of file transfer (from device 1 to 2). These phases ensure that the software on each device is structured to correct for errors or inconsistencies from the corresponding, remote device.

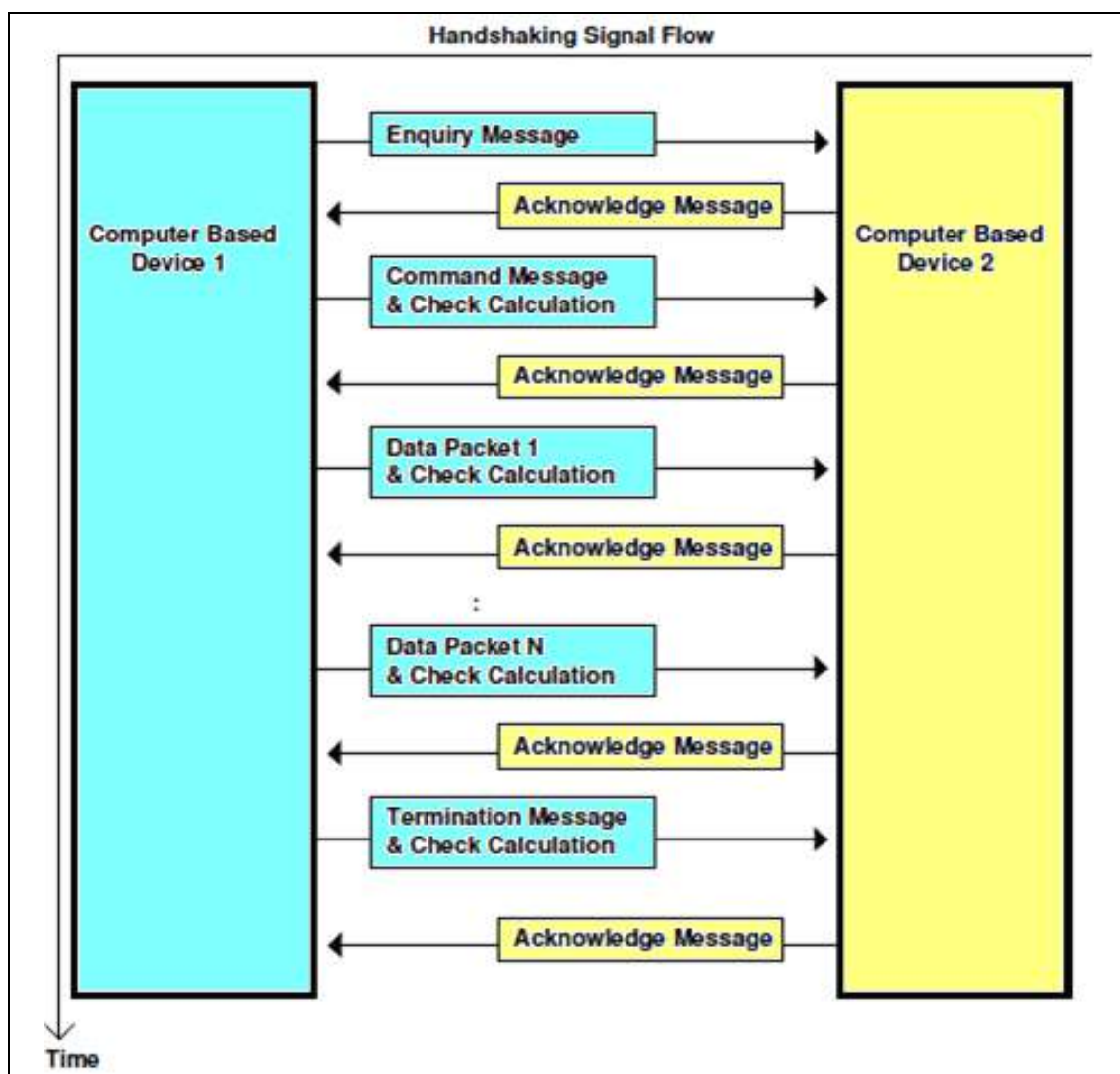


Fig. 10.2(a). File transfer sequence under typical protocol

The rules for each of the above phases are clearly defined by a protocol and typical phases are as follows:

(i) Establishment of Link

Device 1 checks to see if Device 2 **is present** on the link by sending a specific *enquiry* message. If the link is active and device 2 is active then it should respond by sending back an "acknowledgement" message. Device 1 must track the time that device 2 takes to respond. If device 2 does not respond within a specific time interval then device 1 assumes that the link is not active. This is called a "**time-out**" error.

(ii) Issue of a Command and Command Qualifier

Device 1 sends device 2 a message, in a predefined format, which tells device 2 that a file is to be transferred. As a qualifier within the message, device 1 tells device 2 what to do with the file. For example, device 1 may tell device 2 to place the incoming file onto disk storage, with the file-name *newname*.

(iii) Acknowledgement of a Command

If device 2 has correctly received the command and qualifier from device 1, and is capable of carrying out the command, then it sends device 1 an acknowledgement message. The acknowledgement message tells device 1 that it can now proceed with further action needed to fulfill the command. If device 2 is unable to act upon the command from device 1, then it must respond with an error message. An error could occur on the receiver if, for example, the disk on which the incoming file is to be stored, is already full. The error response message would tell device 1 that it should not proceed with its proposed course of actions.

(iv) Dissection of Messages

All messages, commands and otherwise, must be broken down into packets of manageable size for transmission. Thus if an error should occur in a packet, then only that packet needs to be re-transmitted. Therefore, when device 1 wishes to transfer a large file to device 2, the file is broken up into packets and transmitted packet by packet.

(v) Error Detection and Correction

When device 1 sends a message to device 2, it performs a mathematical check on every unit of the transmitted data. This check is transmitted to device 2 immediately after the message. Device 2 performs the same check on its incoming data as device 1. Device 2 also reads in the check sent by device 1 and compares it with the local check. If the two checks are identical, then the incoming message is assumed corrupted on the link. Device 2 can then issue a positive acknowledgement to device 1 to indicate that it is ready for the next message. If the two checks are

inconsistent, then the sent data is assumed corrupted, and device 2 issues a "negative acknowledgement" message to device 1, which indicates that the previous data message must be re-transmitted.

(vi) Termination of Transmission

Device 1 transmits a file, piece-wise, ensuring that each packet is correctly received by device 2. After the last piece of the file is transmitted to device 2, then device 1 must terminate the transmission. Device 1 sends an "end of transmission" message to device 2. This allows device 2 to close the stored file and return to other duties. The various phases of communication for a file transfer, under this typical protocol, assuming no error conditions, are illustrated in figure 10-2. If you can understand the inherent uncertainties with each phase, then you should begin to realize the number of things that can go wrong in the above process. For example, what should device 1 do if device 2 is switched off in the middle of file transfer? What happens if both devices wish to transfer files simultaneously - which device should have preference?

10-3.2. Data Transmission Modes

Data transmission, between computers and other connected devices, may be carried out in series or in parallel.

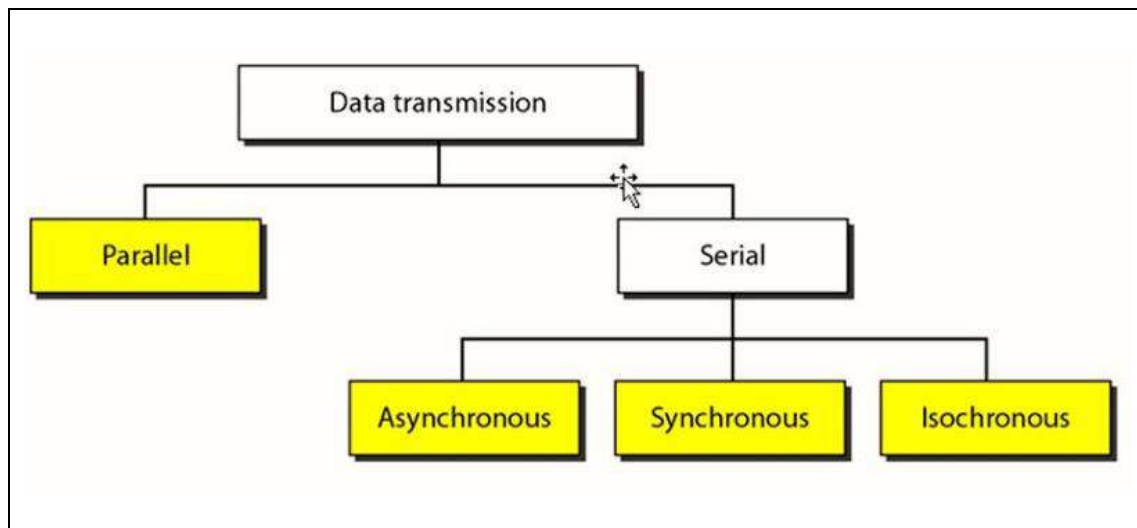


Fig. 10.2(b). Data transmission modes

10-4. Parallel Data Transmission and Parallel Ports

Data within a computer system is usually transferred in a parallel manner. This means that each group of binary digits (bits) are essentially transmitted at the same time and received at the same time. If the basic internal unit of computer architecture has 10-bit data bus, then at least 8 conductors are required to link any two devices for parallel operation.

The objective of data-communication is to transfer binary information from the data bus of one computer to the data bus of another computer, from which it can then be directed to any other internal location (such as memory, disk, etc.). At a first glance, it may appear sensible to extend the concept of parallel communications beyond the boundaries of a single system, to enable computer-to-computer communications to occur. The concept is shown schematically in figure 10-3.

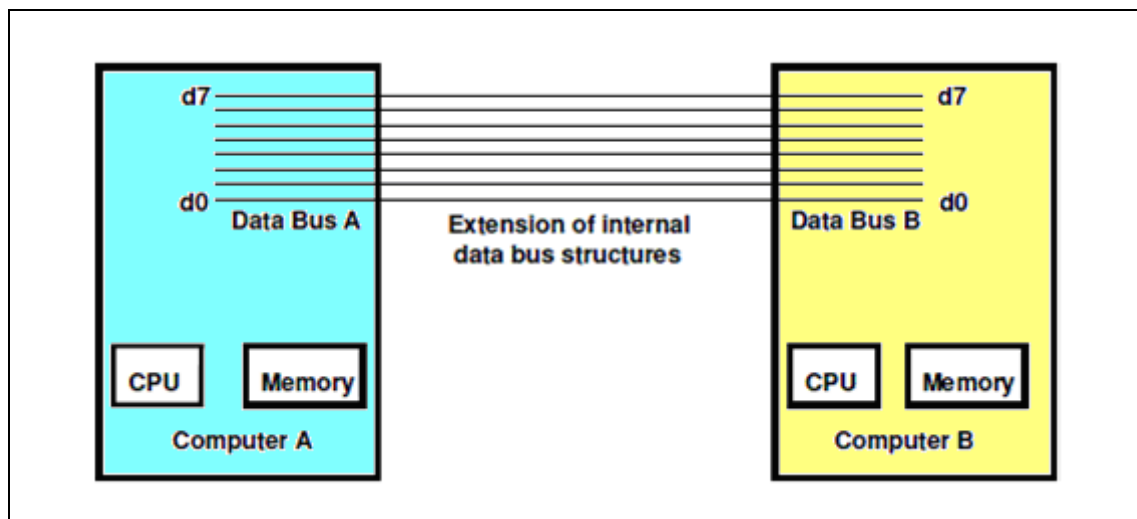


Fig. 10-3 - Expanding "Bit-Parallel" Communications

10-4.1. Parallel Data Link

The problems with implementing a simplistic parallel link, such as that shown in Figure 10-3, are numerous. Firstly there is the problem of physical incompatibility. Computer "A" may use one set of voltages to represent binary digits, whilst computer "B" may use a completely different set of voltages. Secondly, the data bus of computer "B" may be of a different size to the data bus of computer "A". Finally, even if we could link the two devices in this way, we have a situation where two intelligent devices (CPUs) may both attempt to act as masters over the use of the data bus. For example, a contention situation could arise where device "A" sets a line low, while device "B" tries to set the same line high, thereby causing a temporary short-circuit.

Some of these problems are overcome by providing an electronic interface or "buffer" between each device and the external communication link. In order for the two devices to then communicate, the interface on each device must perform a two-way conversion between the internal data bus signals and the common external representation. The circuitry in such a buffer needs to be designed with a view to withstanding possible short-circuits because the interfaces are normally not capable of resolving contentions for use of the external communication medium.

10-4.2. Handshaking in Parallel Communication

When we connect two devices for parallel communication, they are generally not of the same intelligence level. For example, device "A" may be a Personal Computer, whilst device "B" is a Printer (a dedicated, low-level computer). It may therefore be necessary to resolve contentions on such a link by providing additional lines on the interface circuits for external "hand-shaking" purposes. This is shown in Figure 10-4.

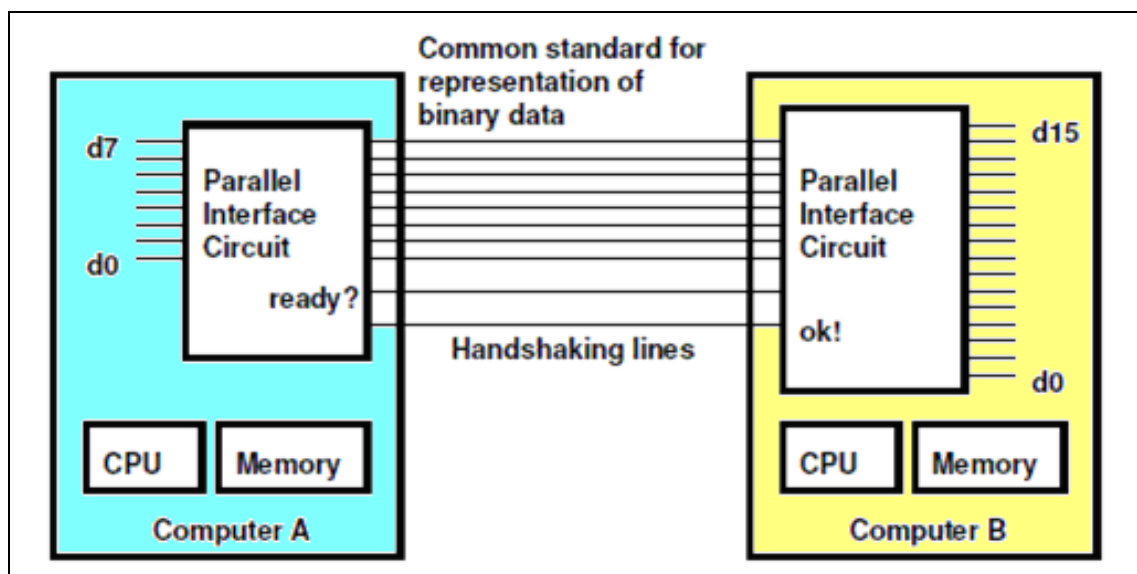


Fig. 10-4. Hand shaking Lines in Parallel Communications.

Handshaking lines on communication links are used in order to synchronize and/or co-ordinate the flow of data between two intelligent devices. They are sometimes referred to as *hardware protocols* because they are designed to achieve the same ends as software protocols. Hardware protocols are not unique to external communications. Within any individual computer system, the address bus structure is the *defacto* hardware hand-shaking system that effectively controls the flow of data from the CPU to the system memory and vice-versa.

Figure 10-4 shows the most common form of hardware hand-shaking between smart devices. If we continue with the scenario where device "A" is a Personal Computer and device "B" is a printer, then the "ready?" and "ok" lines serve the same function as the "Enquiry / Acknowledge" sequence in a software protocol. Device "A" may assert the "ready?" line to a "true" (enable) state, and if device "B" is ready (on-line), then it should respond by asserting the "ok" line to a "true" (enable) state.

This would allow people to develop software on device "A" that takes into account the possibility of device "B" being inoperative or disconnected. Additional control lines are often used in communication links for the coordination of data flow and device to device signaling. A good example is a parallel link between a computer and a printer. The parallel link contains hand-shaking and control lines that pertain to the status of the printer. An "out of paper" line is a common hand-shaking line in such links. Having established how simple parallel links can be made, we are again confronted with the problem of standards for the links between devices. A number of conformance issues are immediately apparent with respect to the parallel link:

- Number of data lines to be used
- Voltage representation of binary data
- Representation of characters
- Number and role of hand-shaking lines.

These issues are addressed by a number of common specifications and standards for parallel links.

10-4.3. Centronics Parallel Port

A number of specifications and so-called "standards" exist to define the parallel communication process between intelligent devices. A parallel communications specification that has become very widely accepted for computer to printer links is called the **Centronics** Parallel Interface.

The **Centronics** specification has been widely accepted by printer manufacturers, primarily because it is the common parallel interface on IBM (and compatible) personal computers. The Centronics connector plug and pin allocation are shown in figure 10-5. The Centronics parallel interface is a 36 line link. Eight of these lines (pins 2 - 9) are used to carry data. Pins 19 to 30, labeled with an "(R)", are signal ground return lines for the corresponding data lines. Binary data in the Centronics parallel link is represented with standard Transistor to Transistor Logic (TTL)

it is ready to receive more – otherwise the printer is not ready to accept data. Table 10-2 lists the various Centronics line functions and their direction of information flow. Note however that the functions of lines 12, 13, 14, 15, 31, 32, 34, 35 and 36 vary according to individual applications, and they are often used to convey additional printer status information (such as out of paper, etc.).

The Centronics Parallel link is designed for fast, one way data flow, from a master device (computer) to what is generally a slave device (printer). It does not readily lend itself to an environment, where two, equally intelligent computer devices can both talk and listen to each other at the same time.

Table 10-1. TTL Voltage Levels for Binary Representation

<i>Quantity</i>	<i>Item</i>	<i>Voltage Range</i>
Binary 0	Inputs to Circuits	0.0 → 0.8 V
	Outputs from Circuits	0.0 → 0.4 V
Binary 1	Inputs to Circuits	2.0 → 5.0 V
	Outputs from Circuits	2.4 → 5.0 V

The Centronics link is primarily intended as a point to point system, with only one device at either end. However, its capabilities can be extended through commonly available "Parallel Exchange Network Units". These enable a single computer to feed a number of Centronics compatible devices as shown in Figure 10-6. Alternatively, a number of computers can use the exchange to share high cost printers and other peripherals. This is referred to as "resource sharing".

The parallel exchange network unit is simply a switching (multiplexing) device, which can make a direct, point to point link between any two devices that are connected to it. The operation, sophistication and number of ports on these exchange units vary markedly between vendors but in relative terms the devices are all inexpensive. This is therefore a common and simple technique for sharing equipment through a relatively simple Centronics "Star" Network. A common resource-sharing implementation might be as shown in Figure 10-6.

Table 10-2. Centronics Pin Configuration and Common Data Flow

<i>Line or Pin</i>	<i>Corresponding Return Line</i>	<i>Signal</i>	<i>Direction</i>
1	19	Strobe	From Computer
2	20	Data Bit 1	From Computer
3	21	Data Bit 2	From Computer
4	22	Data Bit 3	From Computer
5	23	Data Bit 4	From Computer
6	24	Data Bit 5	From Computer
7	25	Data Bit 6	From Computer
8	26	Data Bit 7	From Computer
9	27	Data Bit 8	From Computer
10	28	Acknowledge	From Printer
11	29	Busy	From Printer
12		Paper End	Application Dependent
13		Select	Application Dependent
14		Supply Ground	Application Dependent
15		Oscxt	Application Dependent
16		Logic Ground	
17		Chassis Ground	
18		+5 Volt Rail	
31	30	Input Prime	Application Dependent
32		Fault	Application Dependent
33		Undefined	
34		Undefined	
35		Undefined	
36		Undefined	

This arrangement would enable either of the two computers to use either of the two printers. The majority of exchange units use "software switching" to facilitate this. This means that in order for one device to access another, through the exchange, the originator first sends a simple command string to the exchange. The command string tells the exchange which device is to receive information from the originator. The exchange then makes the physical "point to point" connection and thereafter becomes transparent. The transmitter (e.g., PC) can then talk to the receiver (e.g., printer) as if the exchange was not present. The receiver (printer) is therefore never aware that it is connected to the exchange and functions as normal.

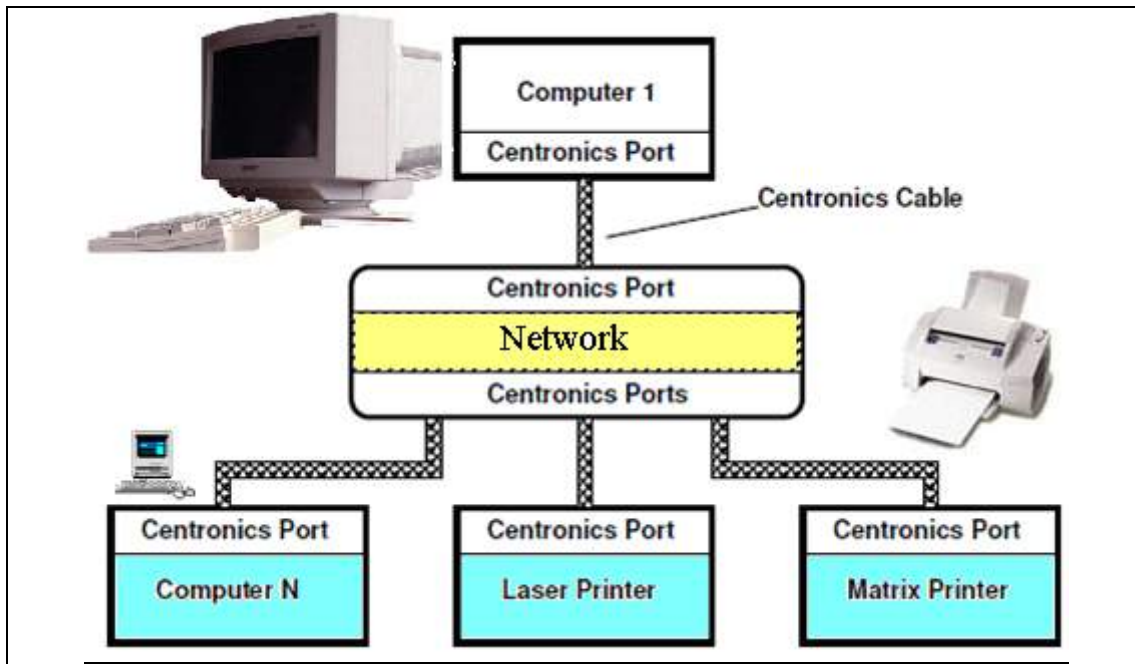


Figure 10-6. Connecting multiple devices through Centronics

It is possible in such a system that a "contention" will arise when both computers wish to use the same printer at the same time. The more sophisticated exchange units can resolve these contentions and "queue" requests for connection in their own built-in memory buffers.

The IBM-PC printer cable has a DB25P connector on the "computer end" and a 36-pin Centronics connector on the "printer end". To limit the Radio Frequency Interference (RFI) generated the cable should be a shielded cable with shielded connectors on both ends. The original official limit on cable length was 3m but this depends on the type of Parallel Port hardware, some can drive far longer cables. The parallel port, as implemented in the original IBM PC, consisted of a DB25S connector with 17 signal lines and 8 ground lines. The signal lines are can be divided into three groups:

- Control (4 lines, C0-C3)
- Status (5 lines, S3-S7)
- Data (8 lines, D0-D7).

All these signals are connected to a 210-pin connector (DB25), as shown in figure 10-17 above. All the bits have TTL logic levels. The control lines (C0-C3) were originally designed as Control and Flow control (handshaking) signals from the PC to the printer.

The status lines (S3-S7) were used for Flow Control signals and as Status Indicators for such things as paper empty, busy indication and interface or peripheral errors. The data lines (D0-D7) were used to provide data from the PC to the printer, in that direction only. As we have already said, later implementations of the parallel port allowed for data to be driven from the peripheral to the PC.

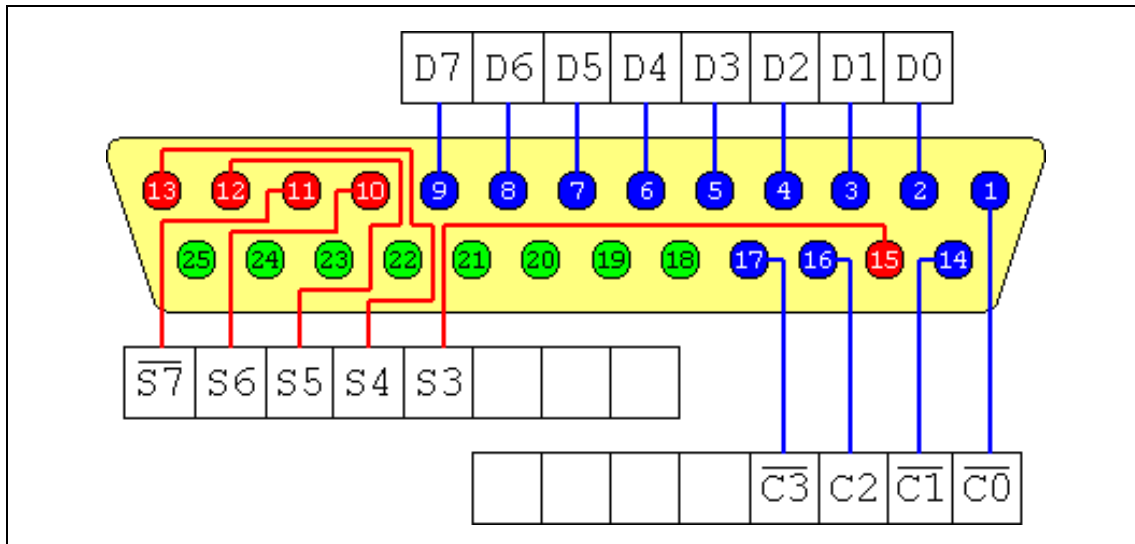


Fig. 10-7. The PC parallel port DB25 connector.

The original Parallel Interface Port used open collector TTL devices and can be damaged by ESD (electrostatic discharge). The Parallel Ports in modern PCs use CMOS devices which are also sensitive to ESD. These outputs often do not conform to the TTL standards, and may have trouble driving long cables, and external signal-powered devices.

Table 10-3. The Pinouts for the Parallel Interface cable.

PIN NAME	PIN NUMBER		DESCRIPTION
	DB25	36 pin Centronics	
Strobe	1	1	a 1 u sec pulse used to clock data into the printer
Data 0	2	2	
Data 1	3	3	
Data 2	4	4	
Data 3	5	5	
Data 4	6	6	
Data 5	7	7	
Data 6	8	8	
Data 7	9	9	

PIN NAME	PIN NUMBER		DESCRIPTION
	DB25	36 pin Centronics	
Acknowledge	10	10	acknowledge signal from printer to computer
Busy	11	11	used by the printer to stop the flow of data
Paper Empty	12	12	indicates the printer has run out of paper
Select Out	13	13	indicates the printer is "on line"
Auto Feed	14	14	not often implemented - wired to ground
Error	15	32	indicates a fault in the printer (motor or paper jammed)
Initialization	16	31	clears the printers buffers and resets defaults
Select input	17	36	a signal on this line is same as select button
Ground	20 to 25	18 to 25, 16, 19 to 30, 33	18 to 25 are paired with the Data wires pins 2 to 9 as shields

10-4.4. Data Transmission with GPIB (IEEE-488)

In a scientific or engineering environment it is often necessary for a number of "intelligent" devices to be linked to one another so that data can be transferred and shared. For example, in a metrology laboratory, a Personal Computer may need to be linked to a number of microprocessor controlled data-acquisition devices so that it can process data and issue control signals. A simple point to point link is clearly inadequate for this purpose. There needs to be some form of "network" by which data interchange can occur.

Within an individual computer system, a parallel network exists in the form of the data and address bus. A Central Processing Unit (CPU) can readily communicate with other chips by selectively setting address bus lines high and low. Each and every memory location in such a system is activated by a unique pattern of high and low bits on the address bus. This is referred to as "addressing". Each chip device in the system has a unique range of addresses allocated to it. Each memory location or register in each chip has a unique address within the range allocated to the chip itself. All the chips share the same conductors (bus) for data transfer and contentions for use of the conductors inevitably arise. These are resolved either through the CPUs "masterly" use of the address bus or through special "bus controller" chips. The internal parallel data transfer network can be expanded for external use, provided that the following parameters are standardized:

- Physical representation of data
- Size of the external data bus
- Contention resolution (for media usage)
- Device addressing.

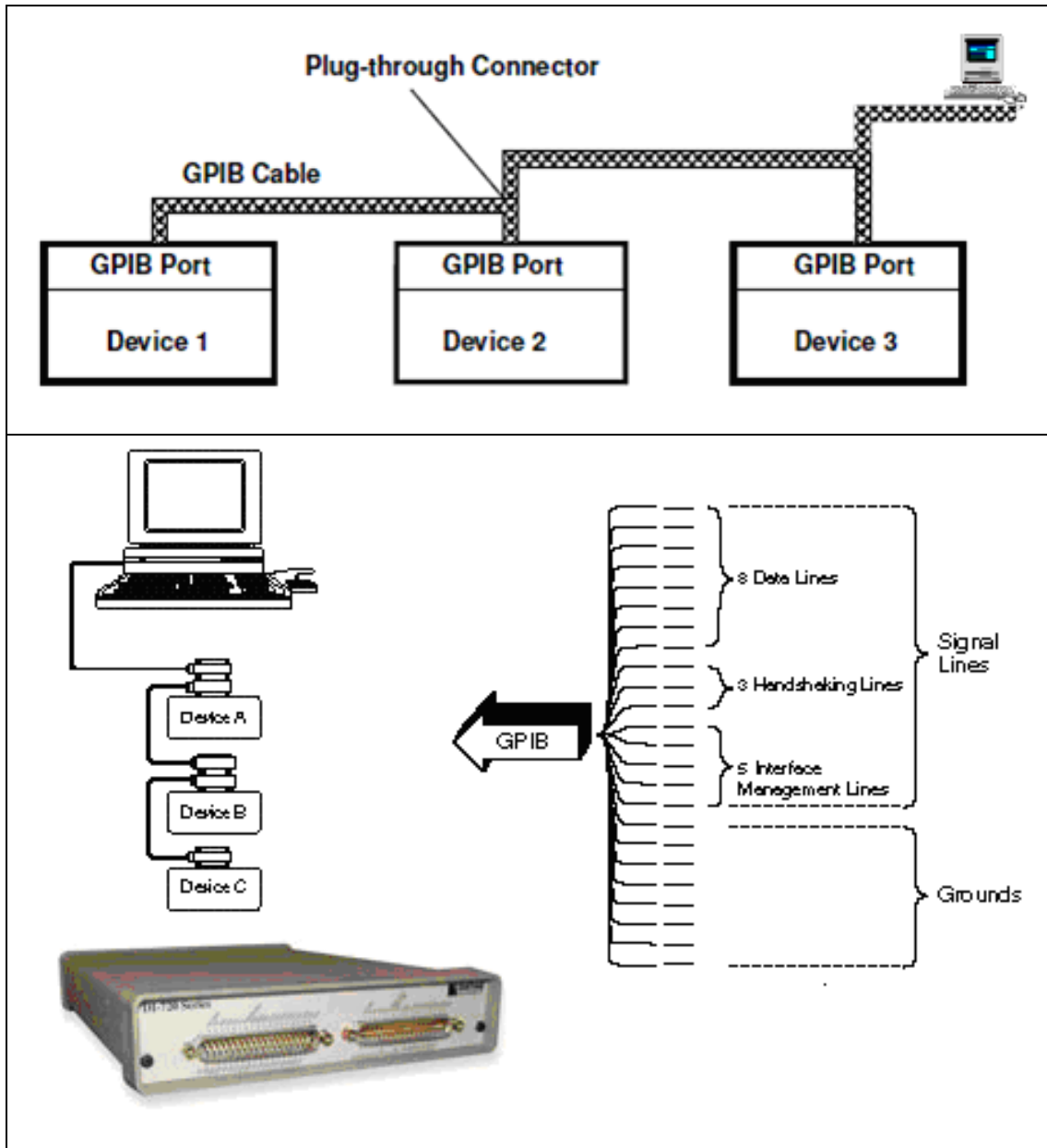


Fig. 10-8. Daisy-chaining to form a Parallel Network

The most common specifications for parallel data communication, in network form, are those referred to as the IEEE-488 Instrumentation Bus or Hewlett Packard Instrumentation Bus (**HPIB**) or General Purpose Instrumentation Bus (**GPIB**). These are all generic or proprietary terms for essentially the same specification. The instrumentation bus is very widely used on scientific, metrological and general laboratory equipment for short distance communications.

The **GPIB** specifications help us to define:

- Number of data lines to be used
- Shapes and pin configurations of connecting plugs
- Values of voltage required to represent binary digits
- Device addressing
- Hand-shaking lines
- Control lines.

The concept of the GPIB is not unlike that of the Centronics parallel system, except that it is far more flexible (in terms of the devices that can be connected to it) and incorporates a number of features that make it amenable for use as a well as a point to point link. Connecting plugs in the GPIB system have a "plug through" facility so that by plugging one connector into the back of another, a daisy-chained, parallel network can be generated. This concept is illustrated in Figure 10-8. The GPIB system is intended as general purpose, network in which data-flow can be bi-directional, unlike the uni-directional, computer to slave (printer) relationship in the Centronics link. The IEEE-488 (GPIB) connector plug and pin allocation are shown in figure 10-9.

The GPIB connector is a 24 pin device with 16 active lines. Each device in the GPIB system must be identifiable through a unique address to avoid bus conflicts. Each device in the system must be aware of its own address in order for the GPIB to function correctly. Pins 1-4 and 13-16 in the GPIB are used for data lines. The data on these lines may either represent unrelated binary information or ASCII character types. The data bus lines (DIO1- DIO8) allow the transfer of data, control words and device addresses. Note however, that in contrast to the Centronics representation of binary information, a binary "0" is represented by +5V DC and a binary "1" is represented by a voltage of 0 Volts.

Within the IEEE parallel network system, there are essentially three types of devices that can be connected:

(i) Controllers

The controller is the device that is capable of getting other devices to accept commands. It does this by asserting the Attention (**ATN**) line (pin 11). Only one controller is permitted to exist on the parallel data bus at any one time.

(ii) Talkers

A talker is a device that is configured to transmit data on the data bus to other devices. Normally, only one talker is permitted to transmit on the data bus at any one time.

(iii) Listeners

Listeners are devices that read in data from the data bus, utilizing a pre-defined hand-shaking sequence. More than one listener can exist and be active on the bus at any one time.

Hand shaking lines work on a similar principle to those in the Centronics system except that a number of devices may assert or monitor the lines. In the GPIB there are three hand-shaking lines, two of which can be asserted by listeners and the third by a controller.

The Data Valid (**DAV**) line (6) is one that is asserted by the controller to indicate that it has placed a control byte on the data bus. The No Data Accepted (**NDAC**) line (8) is one that is asserted by a listener on the GPIB to indicate that it has not yet accepted the last byte that was placed on the bus. The Not Ready for Data (**NRFD**) line (7) is used by an active listener to prevent new data or control bytes to be placed on the bus. Talkers all monitor the NRFD line and wait until it is de-asserted before sending the next 8 bits of information.

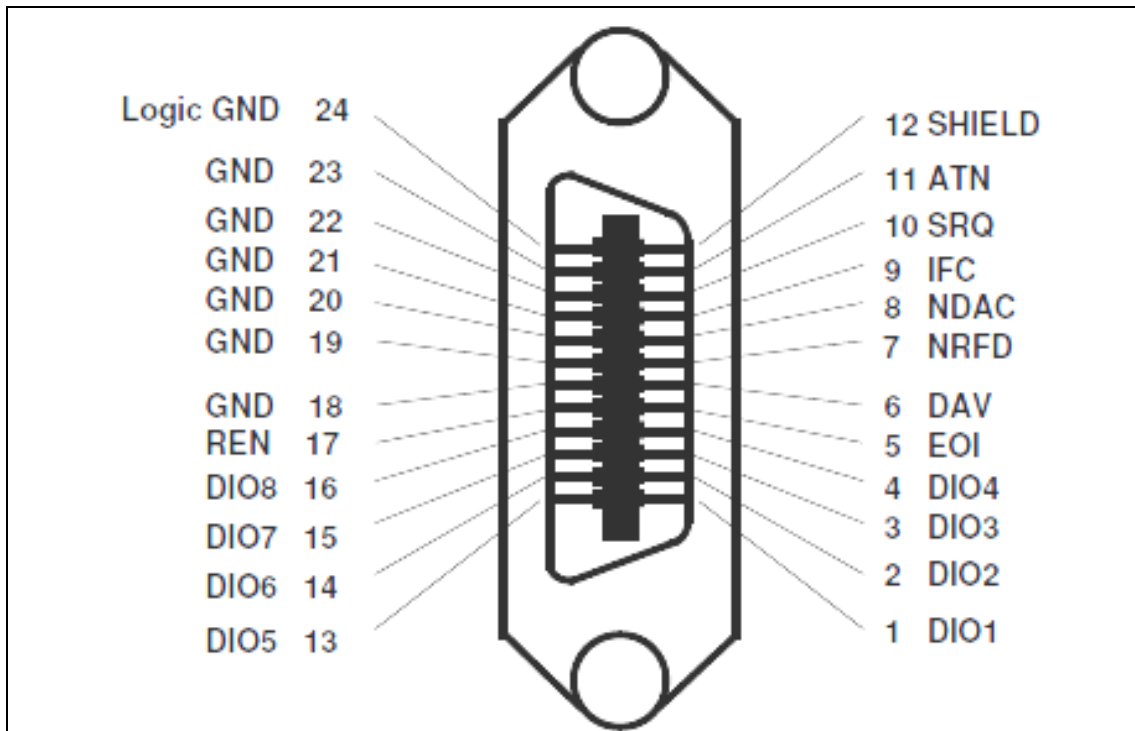


Figure 10-9. IEEE-488 Connector and Pin Configuration

The control lines are similar to hand-shaking lines except that they are used to define the way in which binary information on the data lines is to be interpreted. The Interface Clear (**IFC**) line is used by a controller on the GPIB to place the entire interface system, and all its connected devices, into a defined "quiescent" state.

The **EOI** line is used by talkers to indicate the end of a multiple byte, data transfer message. In this mode it acts as an "end of data" indicator flag. A GPIB controller also asserts the End or Identify (**EOI**) line in conjunction with the Attention (**ATN**) line, when conducting a "poll" on the status of other devices in the system.

The Service Request (**SRQ**) line is asserted by a GPIB device to indicate to other devices the need for some specific service program. The Remote Enable (**REN**) is asserted by a controller to force a listening device to ignore its "front panel" controls.

The IEEE-488 network is a relatively fast and efficient means of transferring data between devices over short distances. This makes it particularly suitable for the laboratory environment, which is electromagnetically clean. As with all parallel data networks and links, the ability to transfer 8 bits of data simultaneously from source to destination is a major factor in performance. A number of computer and scientific equipment vendors, including Hewlett Packard, have used the IEEE-488 system as the back-bone system for interconnecting devices over limited distances.

10-5. Serial Data Transmission and Serial Ports

The only mechanisms for communication between many peripheral devices and smart equipment, with host computers are "**point to point**" serial links. Although serial links are conceptually simple, they sometimes tend to be a major cause of problems. Each serial link between a control system and a host computer represents a unique problem. Each link requires a unique hardware and software solution. A sound understanding of the principles of point to point serial communication is therefore vital in order to reduce development times and costs.

10-5.1. Parallel to Serial Conversion

There exist some common digital circuits, such as shift-registers, that allow us to convert parallel information, as found on a computer data bus, into serial information. This is illustrated in Figure 10-10.

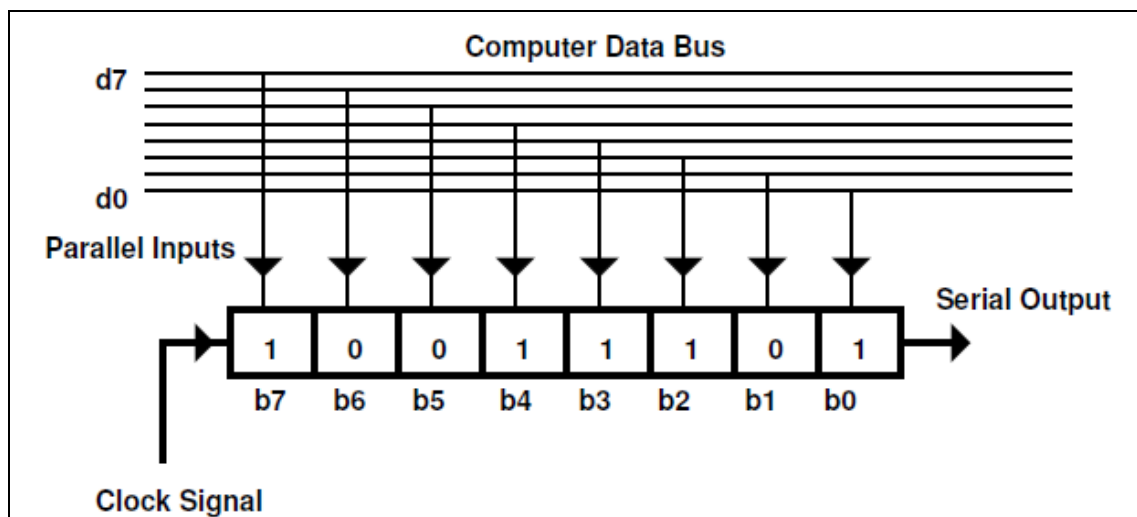


Figure 10-10. Shift-registers for parallel to serial data conversion

The data bus bits, b0-b7, enter the shift-register simultaneously and are fed out of the register, serially in sequence, each time the clock-signal reaches a certain point (usually the negative edge of the clock). Once data is converted from parallel form into a serial bit stream, then it can be transmitted over long distances using only a single conductor. This provides significant savings over parallel transmission, in terms of cabling cost.

A transmission system in which data communications only occurs in one direction is referred to as a **simplex** link. A system in which data communications can occur in two directions, but not simultaneously is referred to as a **half-duplex** link. When interconnected devices can simultaneously transmit and receive then the link is referred to as a "**full-**

duplex" link. When we transfer data in parallel systems, the minimum unit of information (byte or word) is defined by the number of data conductors. For an 8 bit, parallel system, an entire byte of information is transferred simultaneously from one device to another. If that byte represents alpha-numeric information then the minimum amount of information transferred is one character.

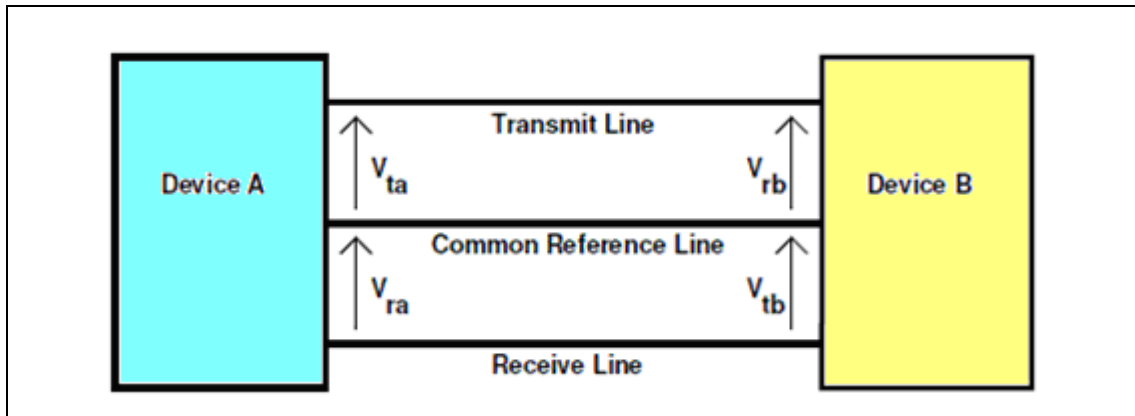


Figure 10-11. Simultaneous two-way serial data transmission

In serial systems the minimum unit of data that can be transferred is one bit. However, sometimes it is necessary to treat a group of bits as a single character, such as in the ASCII system. Therefore in a serial link, we define systems that use one bit as a discrete unit to be "Bit-Oriented". Serial systems in which bits are only interpreted in clusters of 8bits, in order to represent alpha-numeric, are called "Character-Oriented" systems.

In summary, the concepts of serial data communications between computer systems, consists of 4 conversion phases:

- (i) Conversion from parallel to serial representation
- (ii) Conversion from internal form to external line transmission form
- (iii) Conversion from line transmission form to internal form
- (iv) Conversion from serial to parallel representation

The four phases are shown schematically in Fig. 10-12 for a simplex link.

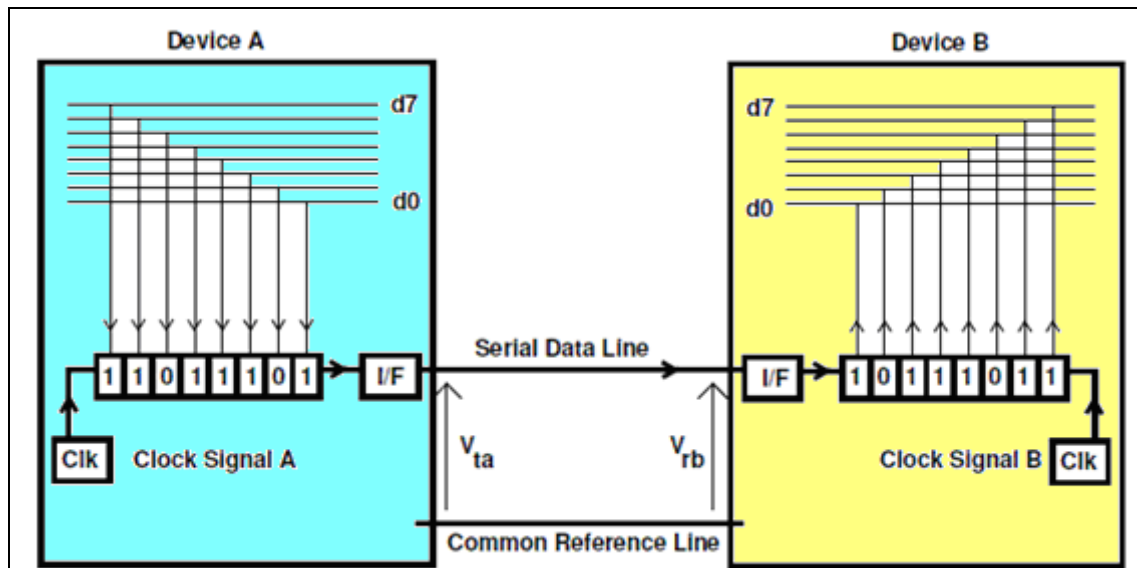


Figure 10-12. Schematic of a simplex serial data transfer between two devices

10-5.2. Synchronous Serial Data Communications

The schematic of figure 10-13 shows two devices that are linked by a common pair of conductors. Device "A" clocks out voltage pulses (the data signal), whose widths are determined by the frequency of clock signal "A". It is assumed that the transmission medium is ideal and that the signal appearing at the receiver is exactly the same as that emanating from the transmitter. The receiving device, "B", reads incoming voltage pulses at time intervals determined by the frequency of its clock signal "B". That is, at a predefined point in its clock cycle (the negative going edge say), device B examines the input signal and accordingly places a high or low value into the register (and shifts other values along).

If the frequency of clock signal A and the frequency of clock signal B are not precisely the same, then it is possible that device B will not be able to interpret the incoming signal correctly. The mismatch between the transmitter clock and the receiver clock eventually causes the receiver to misinterpret the incoming signal - even though it has been correctly received.

The phenomenon of mismatched clocks, between the clock can shift information out of the transmitter register and this can shift data into the receiver register, and can ultimately lead to erroneous interpretation of data. Since it is not feasible to simply expect two independent free-running clocks to be in synchronism, a mechanism must be found to coordinate the clocking of data between a receiver and transmitter.

The first technique enables a receiver to correctly decode incoming serial data. This is referred to as "synchronous" serial transmission. This involves "synchronizing" the receiver's clocking signal to that of the transmitter.

The most obvious way to synchronize the receiver's clock to the transmitter is to provide an additional line, containing the transmitter's clock signal, in parallel with the signal line. This is shown in Figure 10-13.

Although this appears to be a good idea, in practice, this technique is not generally used. This is partly because the use of an additional line diminishes the value of serial transmission. An alternative is to somehow embed the transmitter's "clocking" information into the data signal itself. The receiver needs to "extract" or "recover" the clock signal in order to correctly decode the incoming signal. Whenever the receiver extracts clocking information from the transmitter, the scheme is referred to as "*synchronous transmission*".

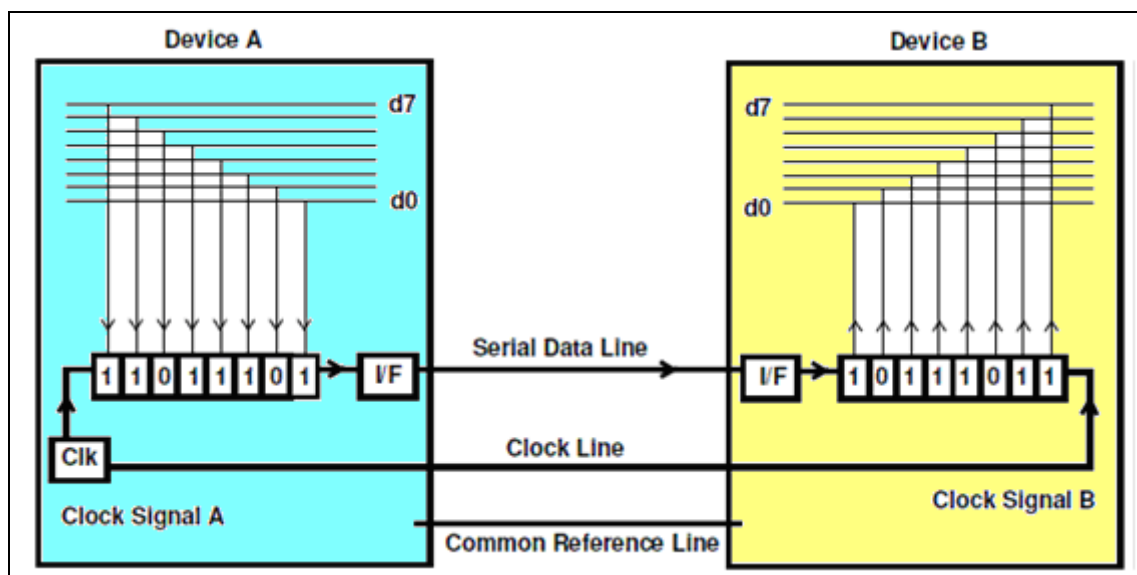


Fig. 10-13. Synchronizing a receiver to a transmitter

There are a number of means by which synchronizing information can be transmitted within a data signal. The first method is referred to as "clock encoding and extraction". Under this scheme, clocking information is placed into the data stream by use of special transmission techniques. The receiving device extracts this clock information, by special clock recovery circuits, and uses it as the reference clock for data decoding.

The different encoding schemes were presented so far in Chapter 3 of this book. However, we recapitulate them here in the framework of computer

data communication. The simplest form of clock encoding and extraction is achieved through a **bipolar encoding** technique, where binary 1's are represented by a positive voltage and binary 0's are represented by a negative voltage. Between each pair of binary bits, the voltage waveform is at zero volts. This is referred to as Return-to-Zero (**RZ**) waveform. A bipolar, **RZ** waveform is shown in figure 10-14.

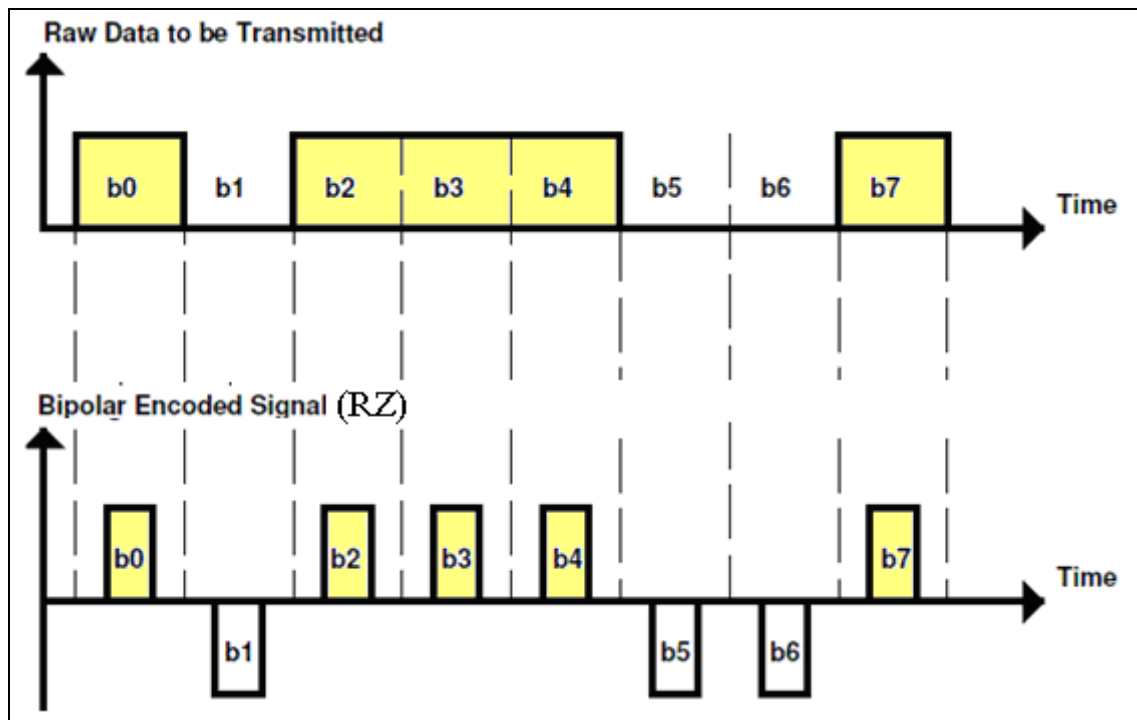


Fig. 10-14. Bipolar encoded or return-to-zero (**RZ**) waveform

The fact that we now have a zero voltage state in between successive bits means that the receiver can detect the time duration between bits and therefore extract "clock" information. Note how this is not possible with the raw waveform of figure 10-14, because successive, contiguous streams of 0's or 1's make it impossible for the receiver to differentiate between bits.

Another, slightly more complex technique, known as the Non Return to Zero (**NRZ**) or Phase Encoding (**PE**) technique, produces a waveform with the normal, two voltage levels. It is sometimes referred to as **Manchester** encoding technique. The width of pulses in the Manchester scheme varies depending on whether successive bits are the same or not. This provides a mechanism for "extracting" clock information at the receiver.

The other means by which a receiving device can synchronize itself to a transmitter is to always ensure that there are enough transitions ($0 \rightarrow 1$ and $1 \rightarrow 0$) in the data signal. This means that the transmitted data must be encoded in a special format. This encoding is referred to as a **Non Return to Zero Inverted (NRZI)** technique or a **differential encoding** technique. Under this scheme, the original waveform is transmitted raw, until a binary "0" occurs. From this point onwards, the waveform is inverted after every binary "0". This is shown in figure 10-15.

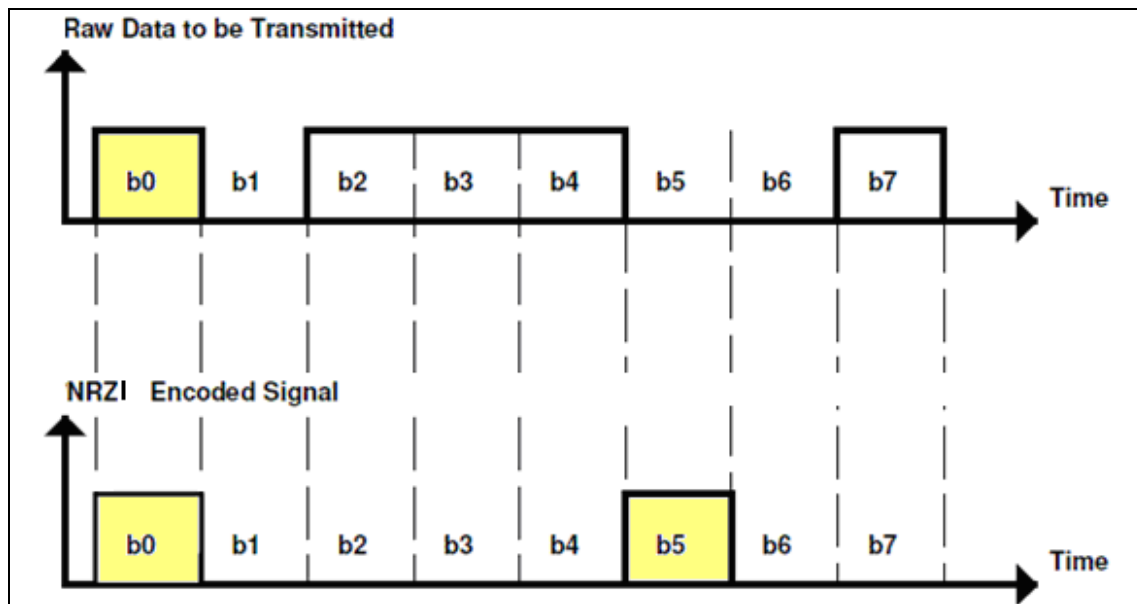


Figure 10-15. Differential (NRZI) encoding of waveforms

Differential encoding (NRZI) ensures that there will always be sufficient bit transitions in the transmitted waveform, to which the receiver's clock can synchronize, provided that there are no continuous streams of binary "1s". In practice, this does not occur, purely because of the way in which binary data is transmitted in synchronous links. Data in bit oriented schemes is broken up into groups of bits, referred to as "frames". In order to distinguish between frames, binary 0's are always strategically inserted into the data to generate special bit patterns.

10-5-3. Asynchronous Serial Data Communications

Asynchronous serial communication is perhaps the most common form of data communications that must be handled within a manufacturing environment. It is widely used for communication between:

- Computers and terminals
- Computers and printers
- Computers and plotters
- Computers and computers.
- Production Controllers (PLCs) and computers

Historically, asynchronous serial communication was introduced for links in which data transmission occurred only at spasmodic intervals - say between a terminal (used by a human) and a computer. In these situations, the transmission line spends a large proportion of its time in an "idle" state, with no transitions between high and low. The receiver in these links therefore needs to be able to synchronize itself to the transmitter only at the start of an incoming data unit (and these are intermittent).

In as much as asynchronous links are designed for low volume data flow, the speeds at which transmission occurs are much lower than those associated with synchronous links. This means that it is not necessary to use complex encoding techniques to supply clock synchronizing information to the receiver. The inclusion of a simple transition (1→0 or 0→1) between basic data units is enough to allow a receiver to decode the incoming waveform..

Another feature that has been incorporated into asynchronous transmission, both for historical and practical reasons, is that the system is usually **character-oriented**. The basic unit of data transmission is normally an 10-bit quantity. This permits the standard **ASCII** (10-bit), extended ASCII (10-bit) and **EBCDIC** (10-bit) character sets. There is no reason why asynchronous systems must be restricted to character-oriented transmission, but in view of their common, it has been a standard. Assuming an asynchronous ASCII system is used, figure 10-16 shows how the character 'u', which has an ASCII value of 117 decimal (01110101 binary), is transmitted. This shows what is referred to as a **data frame**. A frame is a complete basic unit of information that incorporates data, plus the essential encapsulating bits that are used in the transmission.

Figure 10-16 illustrates a number of important points in regard to asynchronous serial data transmission. These are:

- Binary 0 is represented by positive voltage
- Binary 1 is represented by negative voltage
- Line is kept at binary 1 when idle
- Each character is surrounded by start and stop bits.

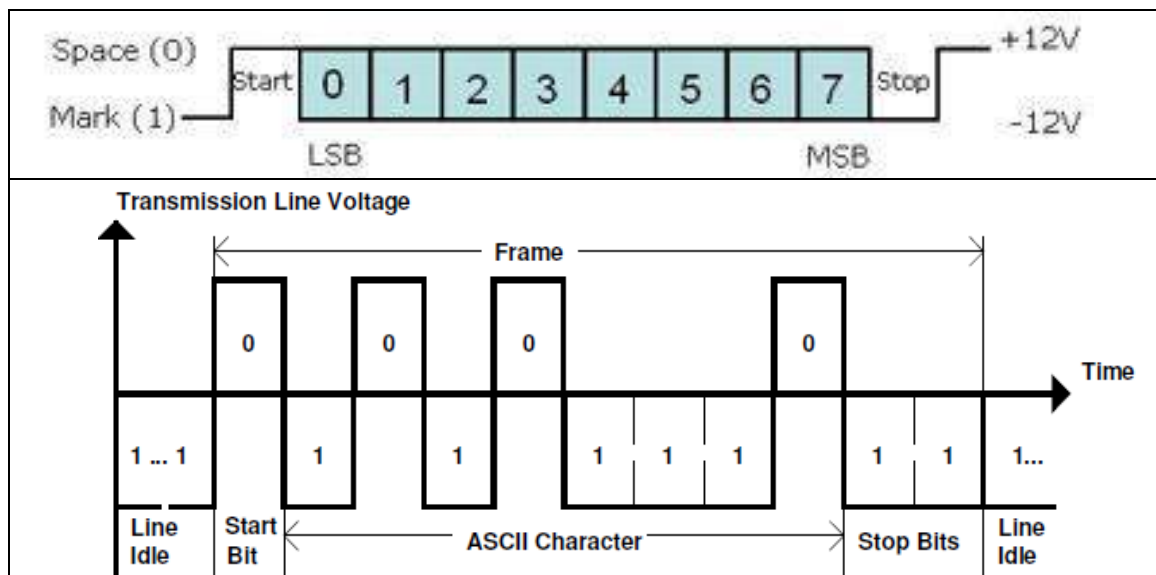


Fig. 10-16. Asynchronous transmission format for the ASCII character 'u'

It is a source of confusion that the voltage logic for transmission of data on an asynchronous link is the exact opposite to what one would naturally expect. This representation is **not a universal standard** but it is a very **common usage** on asynchronous links. In fact, in early teletype-machines, data was transmitted on a link by the flow and interruption of current on the line. It was discovered that the reliability of such links could be greatly improved by maintaining a fixed current during idle periods. Data was then transmitted by interrupting the quiescent state and stopping current flow. The idle state is when the system is "marking time" and hence is referred to as the "**MARK**" condition or binary 1 (current ON). The condition where current is absent is referred to as the "**SPACE**" condition or binary 0 (current OFF). Once we come to terms with the inverted voltage logic, we can see how the start and stop bits allow a receiver to synchronize itself with incoming data. Figure 10-17 shows how the transition appears between characters. The transition region from stop bits to a start bit provides enough information for the receiver to synchronize itself to the incoming data. Note that the number of stop bits

may be 1 or 2 stop bit. As long as both transmitting and receiving devices use the same convention it doesn't make any difference.

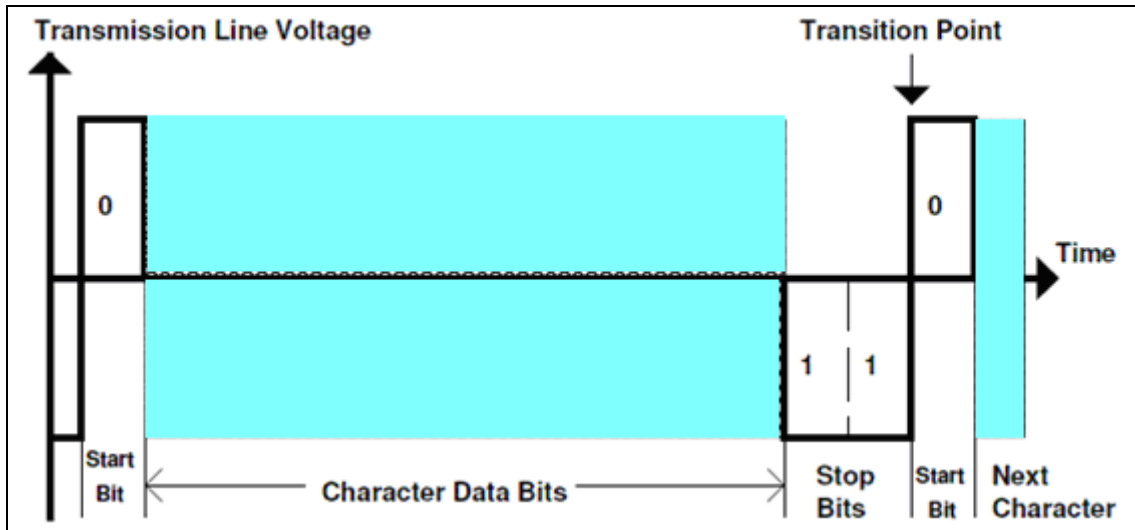


Fig. 10-17. Transitions between frames in asynchronous communications

At this point it is imperative to note another convention in asynchronous data communications. That is, **START** is signified by a binary **0** (positive voltage or SPACE) and **STOP** is signified by a binary **1** (negative voltage or MARK). This makes sense if we adhere to the convention of an idling line being in a MARK state. A change from MARK to SPACE indicates that data is being transmitted. This is again the exact opposite to what we have come to expect in digital circuits, where we normally set a value to binary 1 in order to enable and binary 0 to disable. You need to be extremely careful when reading literature related to this topic. You should try to deal with binary values when analyzing asynchronous serial circuits and you can use Table 10-4 for equivalent terms. As long as you remember that in order to start or enable any function in serial communications you must supply a binary 0 and to stop or disable any function you must supply a binary 1 then there is no confusion.

Table 10-4 - Equivalent Terms in Serial Communications

Binary 0	Binary 1
Space	Mark
Positive	Negative
Start	Stop
Enable	Disable
Low	High
False	True

The fundamental difference between the synchronous and asynchronous communication schemes is **speed**. Asynchronous links generally transmit data at rates of either 1200, 2400, 4800, 9600, 19200, .. , 114200 bits/s. These are relatively slow speeds in comparison to synchronous links that run at speeds in the order of Mb/s. In fact, the clock of an asynchronous receiver should be as many times (x16) as the transmission frequency, in order to be able to extract the timing information from the stop to start bit transition.

The second, basic difference between synchronous and asynchronous schemes is in the amount of data transmitted, and the overheads imposed by adding start and stop bits to a data frame. Since asynchronous schemes relate to low data rates, the insertion of start and stop bits, this increases overheads by a minimum of 2 bits or 25%. This is not acceptable in high speed transfer of large quantities of data.

10-5.4. Error Detection Techniques

Regardless of whether we use synchronous or asynchronous communications techniques, there is always the possibility that an error will occur when we transmit data from one device to another.

The gravity of an error in communications depends entirely upon the application. For example, if an industrial communication link is established to transfer a program from a computer to a PLC and an error occurs, then the corrupted program may ultimately result in damage to either a machine cutting-tool or a work-piece. On the other hand, an error in communications between a computer and line-printer may simply cause a spurious symbol to appear on a print-out. For each application, risk factors must be considered and appropriate error detection and correction measures taken to ensure that link reliability is achieved.

There are a number of error detection techniques in use. The scheme which one might intuitively expect to be simple and effective is called "echoing". The system relies upon the receiving device sending its incoming message back to the transmitter. The transmitter checks the echo message with the original and if they are the same, then it is assumed that the message at the receiver is correct. If the transmitter notes a difference between the original message and the echo, then the original message is re-transmitted. The most obvious problem with the echoing technique is the amount of time required to achieve error detection and correction.

An additional checking information can be transmitted with each message unit, so that the receiver can judge for itself whether or not it has correctly received the unit. These two points are the basis for all error detection techniques used in communication.

Each unit of information that is transmitted on a link is followed by a piece of **checking data** that mathematically relates all the elements in the message unit. The checking data therefore summarizes all the contents of the message unit in a few number of bits. A receiving device takes in both the message and the checking information. It then performs the same checking calculation on the incoming message (as the transmitter performed on the outgoing data) and compares this with the incoming check calculation. If they are the same then the receiver assumes the message unit is correct - otherwise it assumes the message unit is incorrect (corrupt). This is shown schematically in figure 10-18. There are two problems with these systems:

- 1- A transmission error in both the check calculation and the message unit could cause the receiver to mistake an incorrect message unit for a correct one (and vice-versa).
- 2- Since the checking calculation is physically smaller than the transmitted data unit it should be evident that it cannot uniquely describe that data.

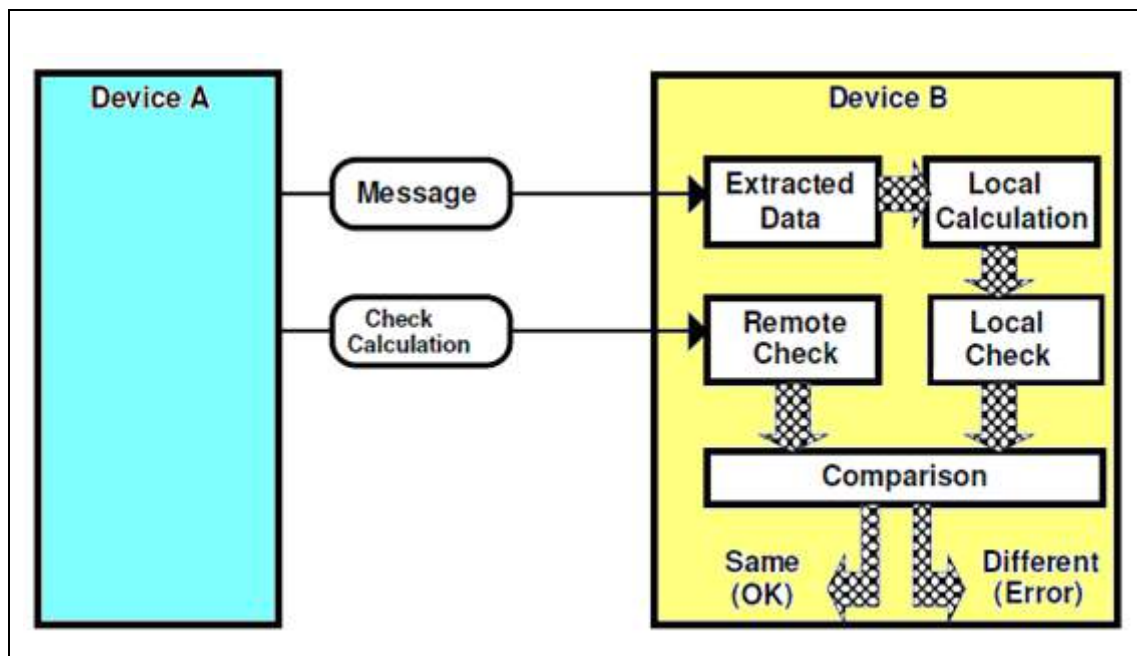


Fig. 10-18. Error detection in the receiver side, through check calculations

The Cyclic Redundancy Checks (**CRC**) is a technique for detecting errors in digital communications. This code is also applied for data storage. CRC is more complex than bit-oriented techniques (such as parity bits and Hamming codes) but also suitable for character-oriented systems.

The CRC belongs to the error checking techniques which are referred to as Frame Check Sequences (**FCS**). The generic name for these check codes, which are appended to blocks of data is **Polynomial Codes**. The number of binary digits in a CRC is chosen to reflect the worst case error burst conditions, but 16 and 32 bit CRCs are predominant in standards for calculations. It is possible to prove that the CRCs are far more effective than other codes in minimizing the probability of errors.

The principles of the CRC error detection techniques have been previously introduced in Chapter 6 of this book. They may be summarized as follows:

- (i) An entire packet or block of data is treated as one binary number " b "
- (ii) The number is multiplied by 2^n by adding n zeros to the end of the number (resulting in the number " B "). The value of n depends upon the specification for the CRC calculation and is equal to the number of CRC digits to be transmitted. That's $n = 16$ for CRC16, and $n = 32$ for CRC32.
- (iii) The new number " B " is divided by another selected binary number " g ", which is referred to as a **generator polynomial**. The value of " g " depends upon the specification for the CRC calculation, but it always contains one bit more than the number of CRC bits to be transmitted
- (iv) The remainder of the binary division is the CRC which is transmitted following the data.
- (v) The receiver takes the incoming bit stream (b'), multiplies it by adding " n " zeros (giving B'), adds the CRC digits (modulo-2) and divides each of them by the **generator polynomial** (g). If the transmission is successful, then the result of the division (r) is always zero - otherwise it is non-zero

10-5.5. UARTS and USARTS

Now that we have examined the basic concepts of serial communications, we can return to two important circuits that are the basis of parallel to serial conversion within a computer. The Universal Asynchronous Receiver Transmitter (**UART**) and the Universal Synchronous Receiver Transmitter (**USART**) are the basic hardware units of asynchronous and synchronous computer communication. These circuits are generally used for character-oriented transmission schemes, since they divide bit streams into blocks of 7 or 8 data bits, encapsulated in stop and start bits.

The schematic diagram for the UART is shown in figure 10-19, together with its links to the computer data bus. The computer address bus has been omitted to preserve some clarity. The control logic block within the UART is responsible for adding parity bits, stop and start bits to outgoing characters. The control block also strips incoming frames of stop and start bits, checks incoming parity bits and the frames, for validity. It is capable of detecting errors in:

- Parity
- Overrun
- Framing.

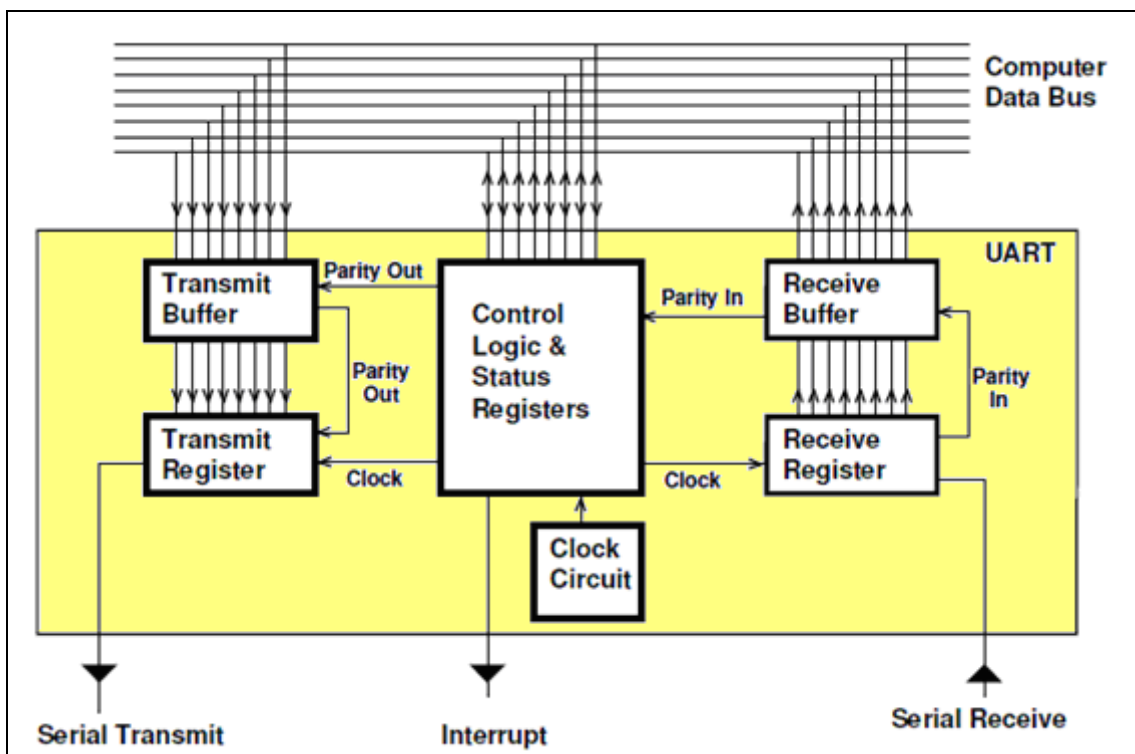


Figure 10-19. Schematic diagram of a UART

Overflow errors occur when incoming characters enter the buffer faster than they can be cleared. This often happens when the bit rate (clock speed) on the transmitting device does not match that of the receiver. Framing errors occur when a stop bit is not received after the control logic has counted the correct number of data bits within a frame.

The control logic circuitry of the UART has a number of internal registers that contain all the relevant status information for the device. These registers are generally mapped onto the computer's memory and can be accessed from the main CPU. UART overflow, parity and framing errors do not halt the operation of the device. These errors are instead flagged within the various status registers of the UART and it is the responsibility of the CPU to act upon them. Most UARTs are designed to assert an interrupt line when data has been received. This enables the CPU to run an Interrupt Service Routine (**ISR**) that removes data from the volatile registers and buffers in the UART and places it into a more stable area of general purpose computer memory.

Another important feature of the control circuitry within a UART is the mode control register. By transferring an appropriate bit pattern from the computer's CPU into this register, the UART can be programmed for a number of different parity schemes, including:

- Odd parity
- Even parity
- Mark parity (always 1)
- Space parity (always 0).

The number of stop bits (1 or 2) and the number of data bits within a frame (5,6,7 or 8) can also be programmed through the status register. Bit rates in the order of 110 bps to 115200 bps can be provided by the control logic, which scales the clock signal supplied by the external clock circuit.

For synchronous transmission, the USRT is used for conversion between serial and parallel data formats. Its schematic is shown in Fig. 10-20. In principle, the USRT is similar to the UART, except that the incoming data is clocked into the receive register with a clock signal that is derived (extracted) from the data itself. From a user point of view, the key point to note about both the UART and the USRT devices is that they perform hardware checking of incoming data.

Some of the hardware checking cannot normally be by-passed. This is of importance because it means that if these devices are not programmed (set-up) correctly by the CPU (i.e, user-program), through use of the UART/USRT mode control registers, then they will continually flag and report hardware errors back to the CPU. These hardware errors and mismatches in framing, cause incoming data to be interpreted incorrectly by user software.

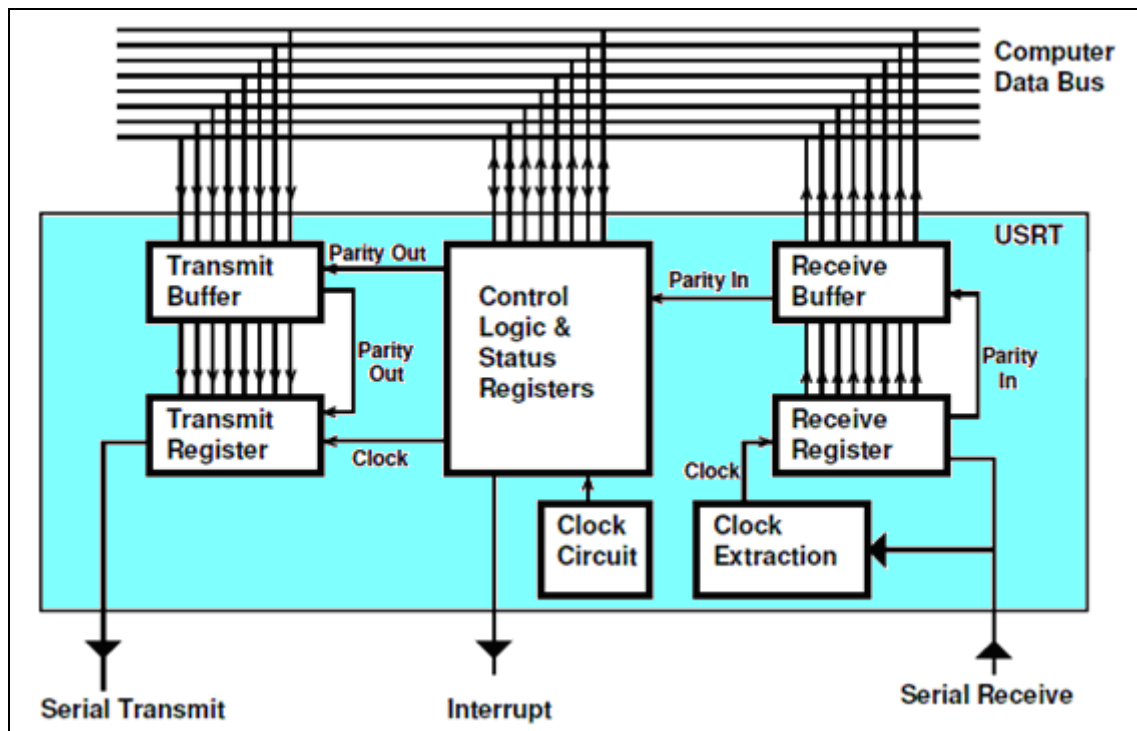


Figure 10-20. Schematic diagram of a USRT

The programming of a USRT or UART involves matching its parameters to those of the device at the other end of the transmission line. These parameters are entered as data in special control registers in each device. This fundamental step of setting and matching:

- Data bits within a frame
- Bit rate
- Parity
- Stop bits within a frame (only UART) must have first priority.

Without a matching set of these key parameters on transmitting and receiving devices, there can be no sensible interpretation of the data transferred on a serial link.

10-5.6. RS-232C Standard

The RS-232C (commonly abbreviated RS-232) standard is a specification for asynchronous serial communications links from the Electronic Industries Association (EIA) and the *Comité Consultatif Internationale de Telegraphie et Telephonie* (CCITT). RS-232 is sometimes referred to as the V.24 standard. The system was originally introduced as a specification for the connection of Data Terminal Equipment (DTE) to a Post Telephone and Telecommunications (PTT) modem. This is shown schematically in figure 10-21.

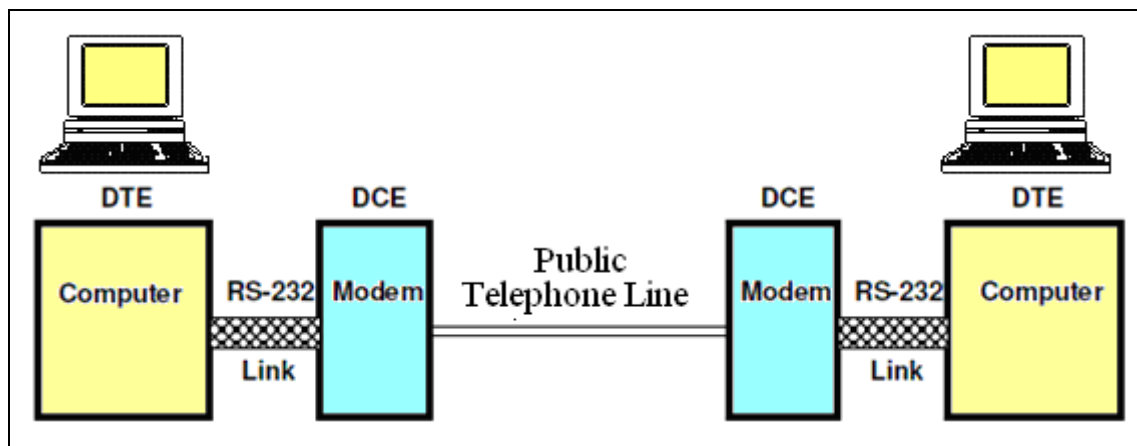


Figure 10-21. Block diagram of the original application of RS-232 C

The RS-232 specification was designed for short-distance, low-bit-rate links between computers devices (computers and modems). The RS-232 port on each of the devices shown in Figure 10-21 is driven by a Universal Asynchronous Receiver Transmitter (UART). The UART performs conversion of outgoing data from parallel form to serial form and incoming data from serial to parallel form. In addition, it is also equipped with special inputs and outputs, which are used to co-ordinate the flow of data with an external device. These special purpose inputs and outputs are referred to as the hardware handshaking lines.

Figure 10-22 shows a schematic of the UARTs in DCE and DTE devices and the way in which data transfer and hand-shaking occurs in the RS-232 system. Hand shaking inputs and outputs in Figure 10-22 are signified by a circle containing the input or output number (on the UART). An inequality sign (< or >) inside the circle denotes the normal direction of information flow with respect to the local device. In figure 10-22, the "Request to Send", "Clear to Send" and "Carrier Detect" lines are directly responsible for switching data transmission on and off. The "Data Set Ready" and "Data Terminal Ready" lines are used so that DCE

and DTE devices, respectively, can indicate whether they are powered up and ready for communication.

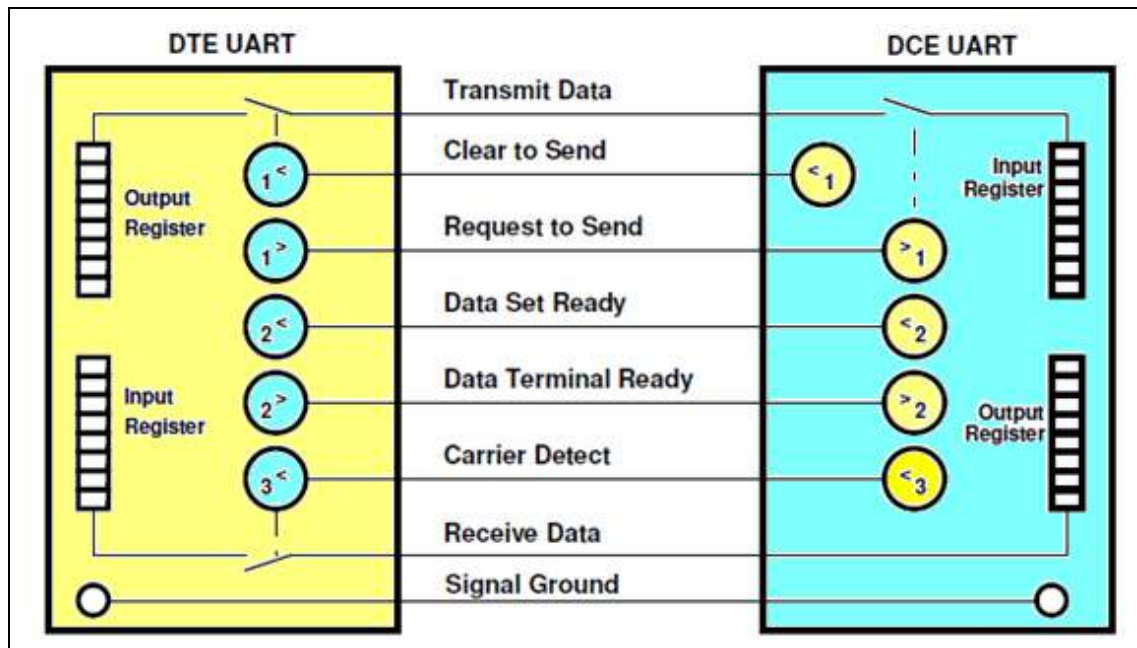


Figure 10-22. Schematic showing DCE and DTE UARTs and Hardware Hand-Shaking on an RS-232C Link

We will look at these lines in more detail in this chapter, but for the moment we only need observe the structure of the hand-shaking in relation to the UARTs. We also need to recognize that RS-232 is one of the most important of all the communications specifications associated with manufacturing. For long time, most computer controlled devices have had, what is referred to, as RS-232 communications port.

i. RS232 Connectors

There are two, very common connectors (plugs) used for RS-232 communications. These are "D" shaped connectors that come in either a 25 pin (called **DB25**) or 9 pin (called **DB9**), male or female form. These are shown in Figure 10-23. As with most elements of the specification, "D" plug configurations are not universally and some equipment manufacturers choose to use a totally different connector. For example, round "DIN" plugs are sometimes used in an industrial environment because they are more rugged.

Table 10-6 is provided for reference and shows the side-by-side pin functionality for both the DB9 and DB25 RS-232 connectors. Again, these pin configurations are not universally adopted, so we need to check manufacturers' specifications for each application. We also note that the 9

pin connectors obviously cannot accommodate the same number of signal lines as the 25 pin connectors. Therefore how can the 9 pin connectors be used? As it happens, modern RS-232 applications require only the 9 pins of the DB9 connector in order to function. Nowadays, the extra pins on the DB25 connector are rarely used in equipment connections.

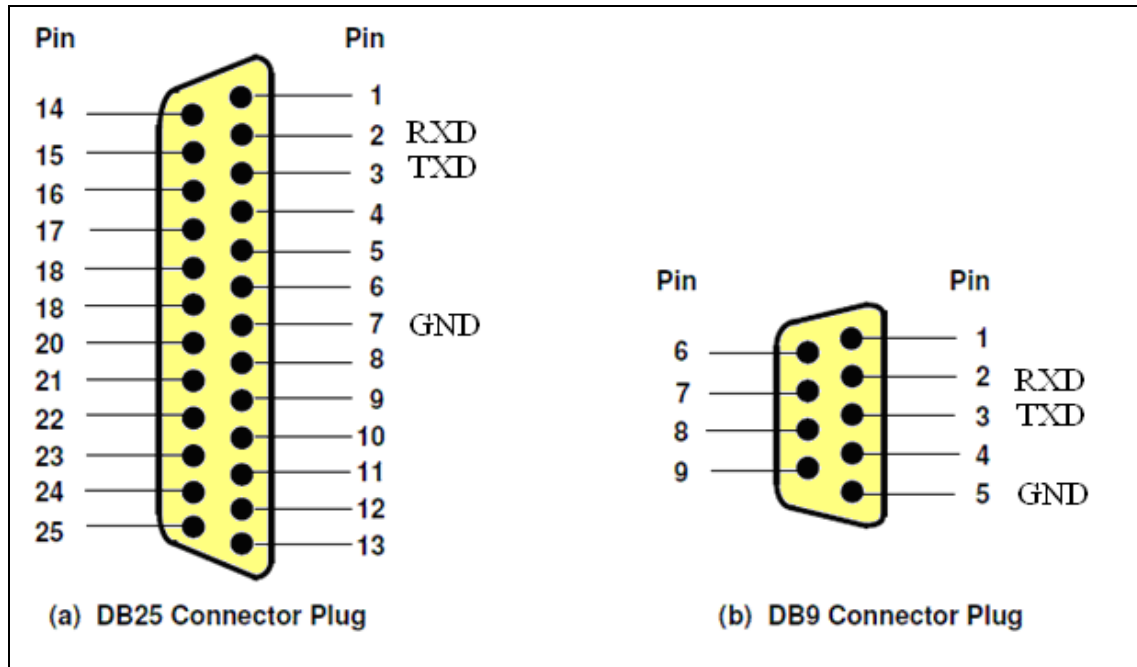


Figure 10-23. Common RS-232 connector plugs

When we talk of data communications, we sometimes refer to the gender of a device in terms of whether that device is Data Terminal Equipment (**DTE**) or Data Communications Equipment (**DCE**). In terms of the original intention of RS-232, these terms had a good significance. Unfortunately, the practical applications of the specification have muddled the original meanings of the gender. For this reason, some modern computerized devices turn out to be DCE whilst others are DTE, and still others can be switched from DCE to DTE. The only way to be certain is by electrical testing of the connectors, and we will look at techniques for this in subsequent sections.

When we talk of RS-232 communications, we also need to talk about the gender of hardware connectors and whether they are male or female. If the original RS-232C definitions were strictly adhered to, then all devices with male connectors could be identified as DTE and all devices with female connectors could be identified as DCE. However, this is yet another specification. It is therefore possible for DTE devices to have female connectors and DCE devices to have male connectors.

Table 10-6. The pin assignment of the DB25 serial connector

<i>Function</i>	<i>Abbreviation</i>	<i>Normal Source of Signal</i>	<i>DB25 Pin</i>	<i>DB9 Pin</i>
Protective Ground	GND		1	
Transmit Data	TXD	DTE	2	3
Receive Data	RXD	DCE	3	2
Request to Send	RTS	DTE	4	7
Clear to Send	CTS	DCE	5	8
Data Set Ready	DSR	DCE	6	6
Common Signal Ground	COM		7	5
Carrier Detect	CD	DCE	8	1
Negative Voltage Rail	-V		9	
Positive Voltage Rail	+V		10	
			11	
Secondary Received Line Signal Detector		DCE	12	
Secondary Clear to Send		DCE	13	
Secondary Transmitted Data		DTE	14	
DCE Transmitter Signal Element Timing		DCE	15	
Secondary Received Data		DCE	16	
Receiver Signal Element Timing		DCE	17	
			18	
Secondary Request to Send		DTE	19	
Data Terminal Ready	DTR	DTE	20	4
Signal Quality Detector	SQD	DCE	21	
Ring Indicator	RI	DCE	22	9
Data Signal Rate Selector	DSRS		23	
DTE Transmitter Signal Element Timing		DTE	24	
			25	

ii. Handshaking on RS232

Hardware handshaking on RS-232 links is extremely undesirable, purely because of the variations that exist in non-standard links (that is, when we use RS-232 for connecting devices other than computers and modems). Unfortunately, understanding the hardware handshaking is a necessary because a large number of devices (particularly printers) already use it extensively and must be accommodated accordingly. Before we proceed any further, we need to examine how binary data is represented on the RS-232 link. The voltage levels for RS-232 logic levels are shown in Figure 10-23. Note the inverted logic and also the error margin between outputs and inputs. This margin is designed to cater for line attenuation.

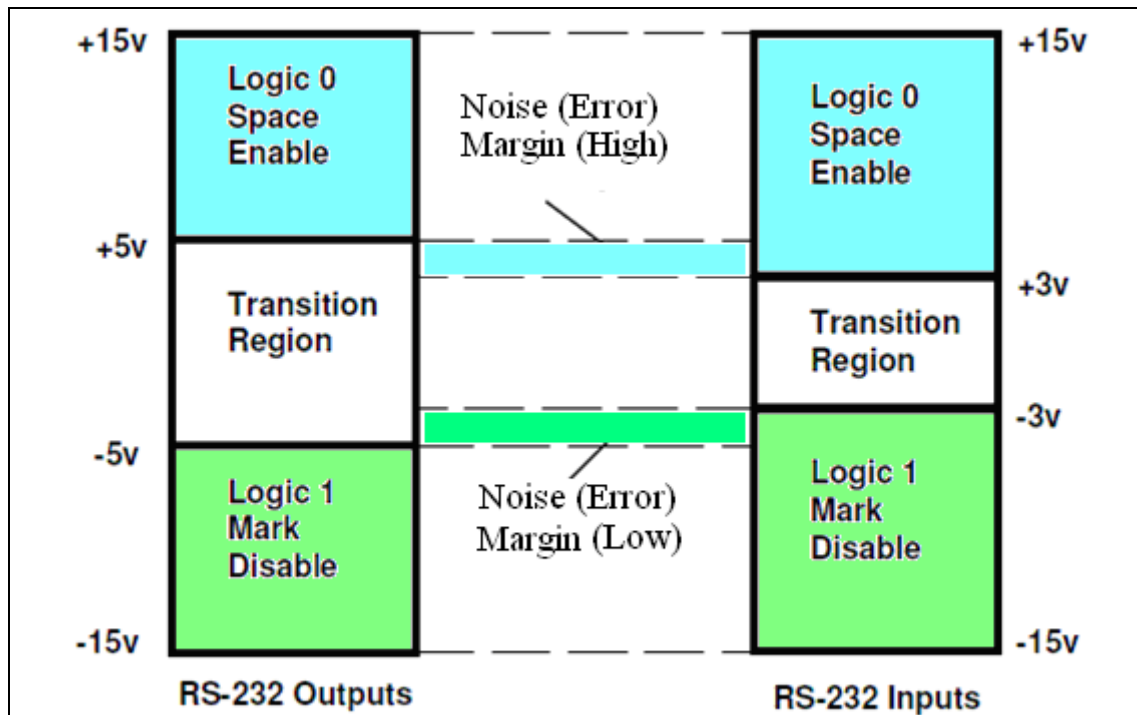


Fig. 10-24. RS 232 voltage levels.

Fortunately, the RS-232 voltage ranges are generally adhered to most implementations. Another feature of RS-232 is that all the outputs and inputs are "buffered". In other words, it is possible to short-circuit any two pins on an RS-232 port without causing damage to the circuitry. In light of the amount of experimentation that is often required to establish an RS-232 hardware link, this is an absolutely vital feature of the system.

Now that we have established the correlation between voltage levels and binary logic levels, we can examine what should theoretically happen, in terms of hardware hand-shaking, on an operational link. We use the model shown in Figure 10-25 as the basis of our examination and assume that the entire 9 wire connection is in place for the RS-232 link. As fate would have it, the link of Figure 10-25 can be used in either half-duplex or full-duplex mode and the hardware hand-shaking differs depending upon which mode is active. The explanation which follows attempts to define both the half and full-duplex hand-shaking arrangements. Let us assume that Computer A wishes to access Computer B by dialing it on the telephone network. In the normal course of events, once Computer A dials, Computer B should respond with an answering tone, then a brief log-on message (prompt) to Computer A. The normal communications exchange then begins.

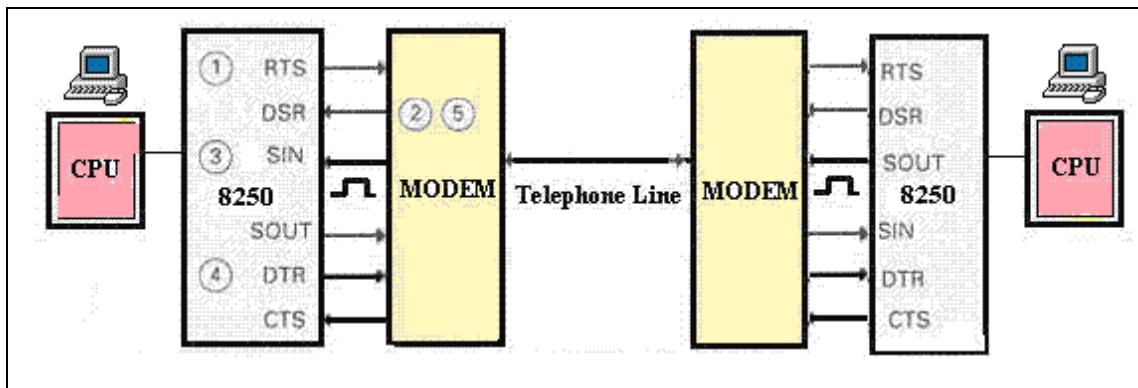


Figure 10-25 Basic model for RS-232 Handshaking lines.

The following handshaking sequence occurs:

- (a) If Computers A and B are both ready for communications then they each enable their Data Terminal Ready (**DTR**) lines.
- (b) If Modems A and B are both ready for communications then they each enable their Data Set Ready (**DSR**) lines.
- (c) After Computer A has dialed the number, Modem B enables its Ring Indicator (**RI**) to tell Computer B that it has been called.
- (d) Computer B responds to the Ring Indicator by enabling its Request to Send (**RTS**) line.
- (e) When Modem B receives an RTS from Computer B it transmits a carrier tone across the phone line to Modem A. When Modem A receives the carrier signal from Modem B, it enables its Carrier Detect (**CD**) line. In a half-duplex link, this tells Computer A that it should not send data because computer B wishes to transmit. In a full-duplex link, an enabled CD tells Computer A that the link is active.
- (g) After a short delay (which gives the A side time to get ready to receive information) Modem B gives Computer B a Clear to Send (**CTS**) signal indicating that it should now proceed with data transmission.
- (h) Computer B transmits its message to Modem B, which then modulates the binary information into data tones. In a half-duplex link, when Computer B has finished transmission it disables its RTS line. In a full duplex link both DTEs keep their RTS lines enabled.
- (i) In a half-duplex link, when Modem B notes that the RTS line is disabled, it stops transmitting the carrier to Modem A. Modem A

responds by disabling its Carrier Detect. In the full-duplex link the carrier always remains on.

(j) In the half-duplex arrangement, when Computer A notes that the Carrier Detect is disabled, it enables its Request to Send line in order to send a response message.

(k) After a short delay (which gives the B side time to get ready to receive information) Modem A responds to the **Request to Send** signal from Computer A with a **Clear to Send** signal. Computer A can then transmit its data. This two-way process continues until neither of the devices has any more data to transmit. The entire process is shown schematically in Figure 10-25. Note the convention where enabled signals are qualified by a "↑" and disabled signals are qualified with a "↓".

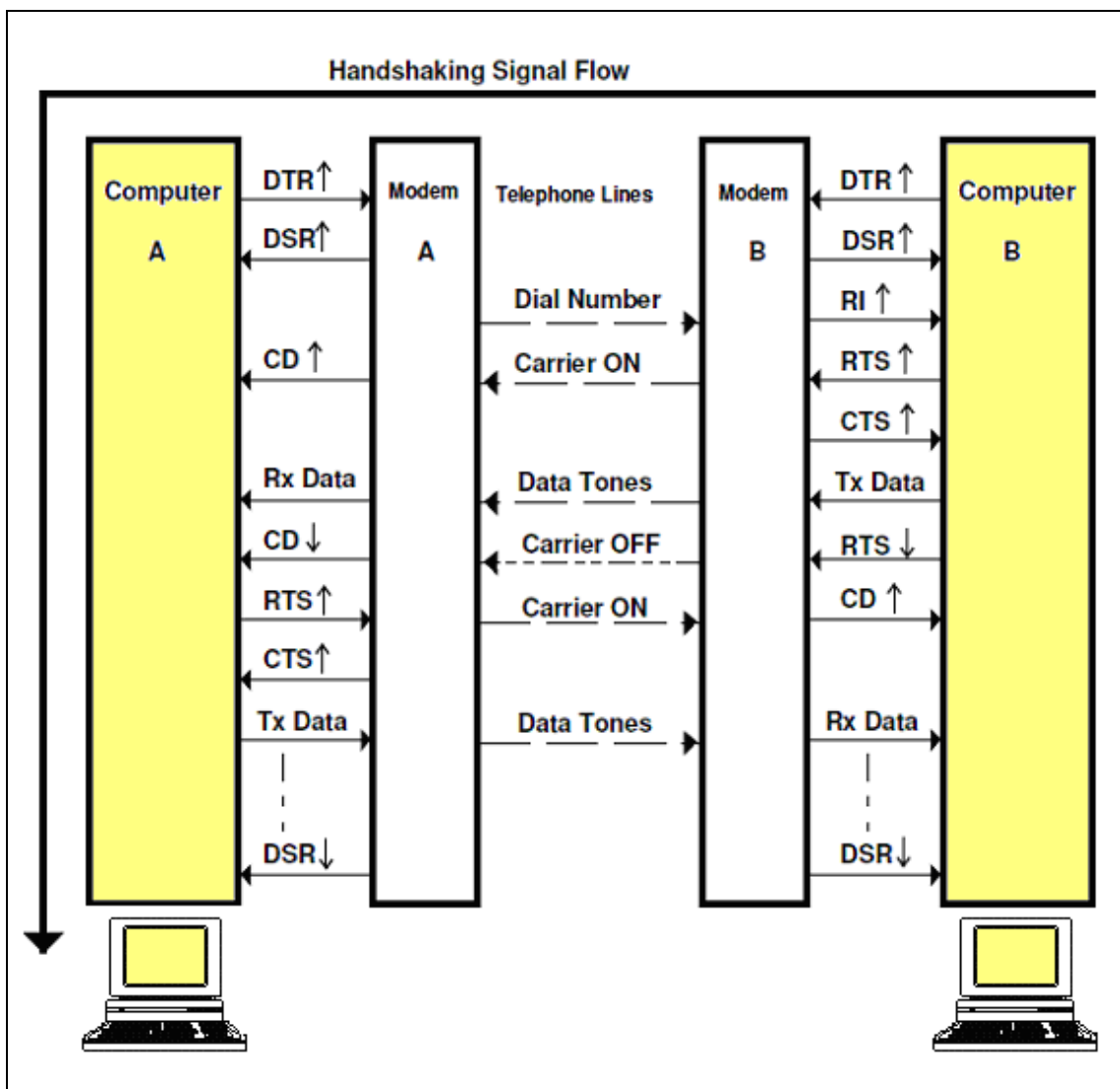


Figure 10-26. Basic RS-232 Hardware Hand shaking Model for a Half-Duplex Link

10-5.7. Universal Serial Bus (USB)

The Universal Serial Bus (**USB**), introduced in 1997 took some time to gain acceptance but by the middle of 1998 manufacturers were starting to make peripheral devices interfaced to the PC via USB. Nowadays, the Universal Serial Bus (USB) is the basic serial bus for notebook and desktop computers. With a simple 4-wire interface, and speeds of up to 12 Mb/s (and even higher! in recent versions), USB has soon become the standard interface for most medium speed peripherals.

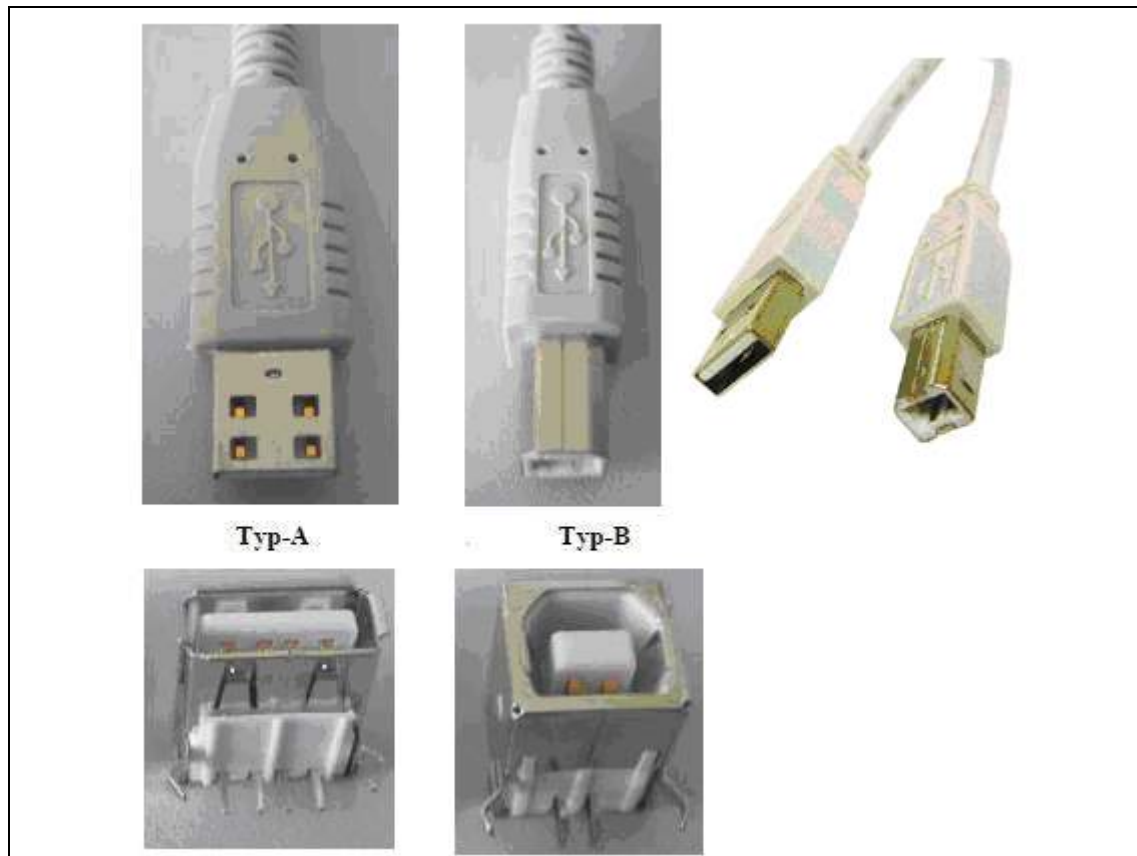


Fig. 10-27. USB connectors and cables. The USB connectors for the USB devices side (A-plug) and host computer side (B-plug) are shown in the right side.

The biggest factor in USB wide acceptance by computer industry is its ease of use. Some key features of the USB that provide ease of use include:

- Completely **Plug and Play** - Peripherals are correctly detected and configured automatically as soon as they are attached to computer.
- Hot attach and detach - Allowing adding and removing devices at any time, without powering down or rebooting computer system.
- USB provides a direct connection to devices without add-in cards.

A single connector can be used for chaining many devices that have to be interfaced via Serial, Parallel or Games Ports in the past. All kinds of devices can be hooked to the PC through the same connector simultaneously. These devices include:

- The keyboard, mice and other pointing devices
- Modems
- Printers, scanners, bar code scanners and Digitizer Tablets
- Digital cameras and web cams
- Memory sticks (flash memory devices)

USB requires less real estate (less space on the back plane) than existing I/O ports and this is particularly important for laptop and hand-held PDA systems. It reduces the number of BUS slots required on the system board, allowing a footprint reduction for desktop systems.

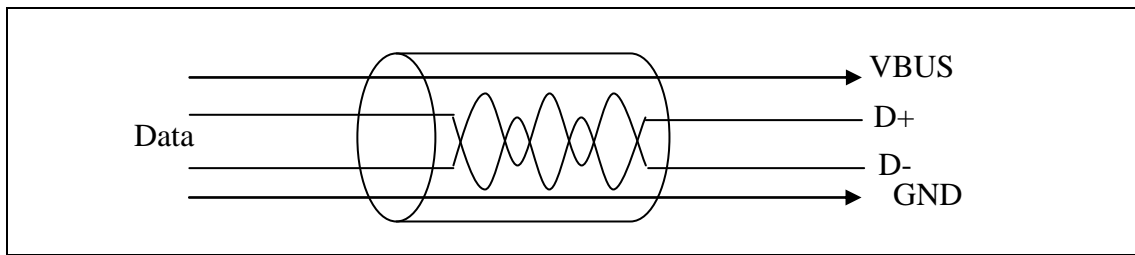


Fig. 10-28. USB cables (4 lines). Data is transferred on differential pair (D-, D+).

The USB system is generally composed of 3 parts:

- **USB host**, which is usually the computer (only one host is allowed)
- **USB devices** (serial devices, like mice and KB, or hubs)
- **USB interconnects.**

i. USB 1.1 & USB 2.0 Specifications

USB 1.1 has ample bandwidth for digital gaming peripherals and video applications, and provides cost effective connection for peripheral devices. The main features of USB 1.1 are as follows:

- 12Mbps design with low cost peripherals
- Supports up to 127 devices
- Both synchronous and asynchronous data transfers
- Up to 5 m cable length
- Built in power distribution (VBUS) for low power devices
- Supports chaining through a tiered star multi-drop topology, via hubs

USB 2.0, is 40X faster than USB 1.1. It is the current generation of USB connectivity, providing additional bandwidth. USB 2.0 moves data at 480 Mb/s, and is fully backwards compatible with the USB 1.1 devices.

ii. USB to RS232 Interface.

The FT232R is a USB to serial UART interface with optional clock generator output. The internally generated clock can be brought out of the device and used to drive a microcontroller or microprocessor. Software drivers, which allow FTDI devices to work with several operating systems like Windows XP/Vista, are available..

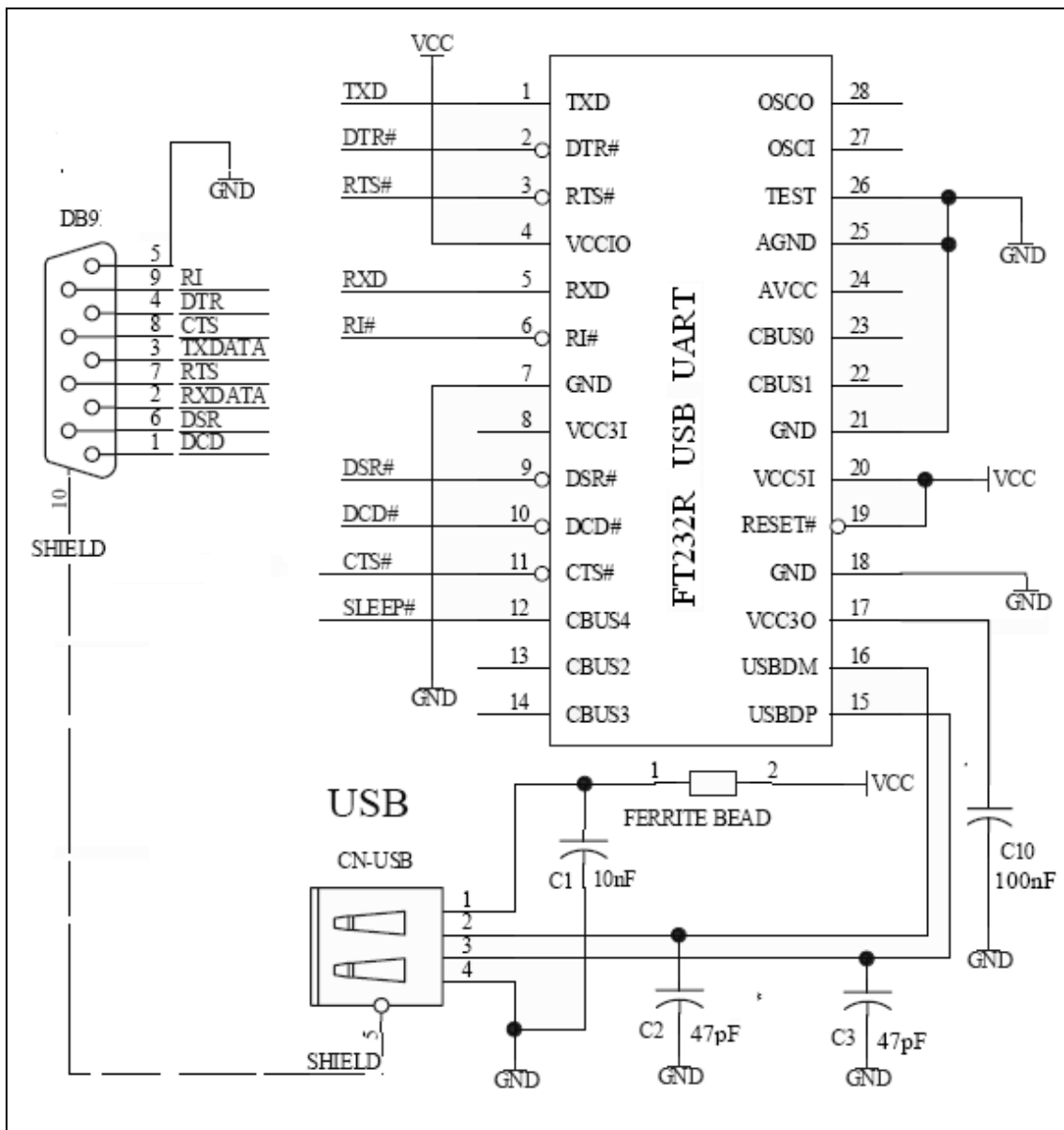


Fig. 10-29. RS232 –to-USB interface via the FT232R chip

10-5.8. Other Serial Communication Standards

The conventional RS232 standards based serial interfaces (serial ports) provided on PC's are limited to a data rate of about 115 kb/s and this is only suitable for slow peripheral devices like mice, and keyboards. The PC industry has been always demanding a faster external serial I/O port or bus that could be used to simplify the interfacing of a wide range of existing and new I/O devices.

i. ACCESS.bus

The concept of ACCESS.bus and its technology, was originally developed by Digital Equipment Corp. (DEC) and Philips Semiconductors, and was taken over by the Independent ACCESS.bus Industry Group (ABIG). At first the ACCESS.bus showed the potential to become an industry standard. Unfortunately it did not attract much support from PC hardware manufacturers.

ii. FireWire

The FireWire was an initiative by Apple in the area of a fast universal serial bus. FireWire may be considered an alternative to USB. FireWire is a registered trademark of Apple and this caused problems with the name used for this technology, when it was widely accepted by the PC industry. FireWire has been endorsed by the IEEE who has formulated a specification for the technology, outlined in document IEEE 1394.

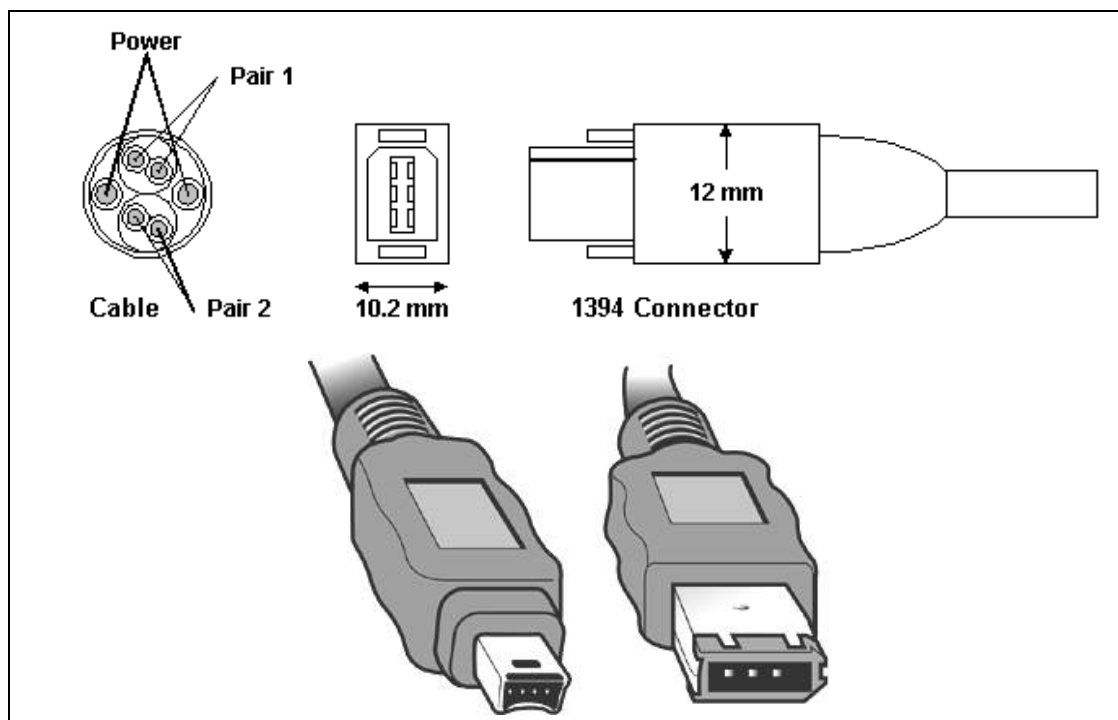


Fig. 10-28. FireWire cable cross section and connectors (type A and type B).

The specifications for FireWire are impressive with a transfer rate of **400 Mb/s**. There are also plans for an even faster version with a 1 GB transfer rate. FireWire allows for **63 devices** on a single bus without pre-assigning addresses and without the need for terminating devices.. In the computer industry some people see USB and FireWire as complementary technologies. They see USB being used for applications that do not need the high-speed transfer rate. Devices such as keyboards, mice and modems, and FireWire used for video processors, video capture, virtual reality gaming devices and high speed data transfer to and from external storage devices.

iii. IrDA B

The IrDA is a standard that was started by the **Infrared Data Association** in June 1993 and has been through several revisions since then, with proposed revisions increasing the maximum data rate from the original 115 kb/s to a rate of 4 Mb/s. It was intended as an alternative to Wireless (RF) networks for connecting Laptop computers to local area networks (**LAN's**), or for replacing parallel or serial cables to printers.

The first IrDA standard was published in June 1994, and it was revised by August 1994 with the high speed extensions to the specification. The original basic specifications for IrDA were:

- Serial, a synchronous communications
- Range of at least 1 m
- Data rate, 2400 to 115 kb/s (kilo bits per sec)

Support for IrDA must be provided at both the hardware and software level. Windows 9x/XP provides support for IrDA devices via add-on drivers available from Microsoft. Older hardware requires an IrDA device plugged into a serial or parallel port and suitable device drivers to provide support. The implementation of IrDA on system boards usually takes the place of one of the conventional serial ports. This means when IrDA is used only one serial port is available. This may not be a problem with system boards which provide a PS/2 mouse port. This also means the data rate is limited to the fastest speed available from the UART, 115 kb/s. However, some attempts have been made to run at higher speeds, included in the IrDA specification. The high-speed extensions to the IrDA standard allow data rates of 1.152 Mb/s and 4 Mb/s. High speed devices are backward compatible with the first IrDA standard and devices built to this specification.

iv. I²C (Inter-Integrated Circuits) Bus

The I²C bus is a simple 2 wire serial interface which has been developed by Philips. It is widely used in consumer electronics and industrial applications. In addition to microcontrollers, several peripherals also exist that support the I²C bus. The I²C bus physically consists of 2 active wires and a ground wire. The active wires, SDA (Serial Data) and SCL (Serial Clock), are both bidirectional. Up to 128 devices can exist on the network and they can be spread out over 10 meters. I²C devices can act as receiver and/or transmitter. As shown in Fig. 10-29, the I²C bus is a multi-master, multi-slave network interface. The master is usually the microcontroller and the clock is always generated by the master. Each node (microcontroller or peripheral device) may initiate a message, and then transmit or receive data. Each node on the network has a unique address which accompanies any message passed between nodes. The I²C bus interface typically has data transfer speeds up to 3.4 Mb/s.

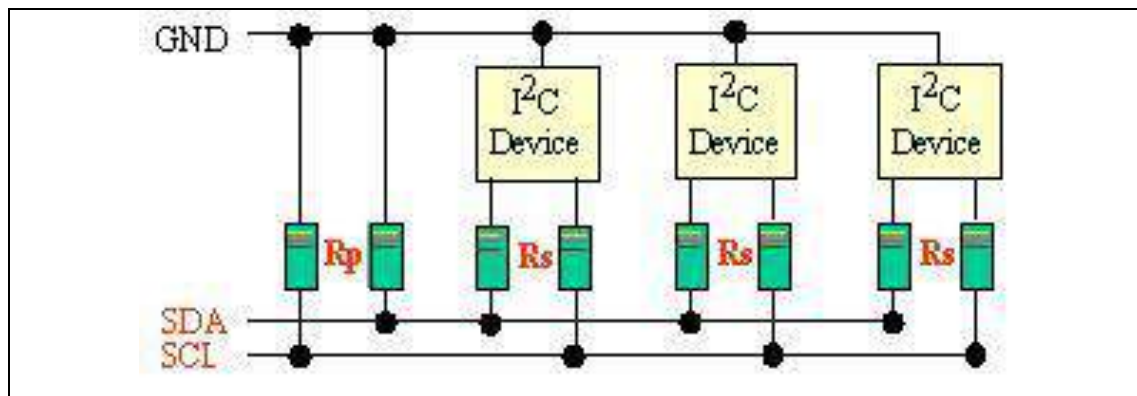


Fig. 10-29. Simplified model of I²C bus.

v. SMBus (System Management Bus)

The System Management Bus (SMBus) is more or less a derivative of the 100kbps I²C bus, which has been developed by Intel. The main application of the SMBus is to monitor critical parameters (e.g., supply voltage and processor temperature) on computers and embedded systems.

vi. JTAG (IEEE 1149.1) Bus.

The JTAG is a standard for providing external test access to integrated circuits serially, via a 5-pin external interface. JTAG is an acronym for the *Joint Test Action Group*, which developed the standard. The JTAG standard has been adopted as an IEEE standard (IEEE 1149.1 or DOT1) in 1990 under the name “Standard Test Access Port and Boundary-Scan”. In 1994, the standard has been supplemented with the boundary scan, description language (BSDL).

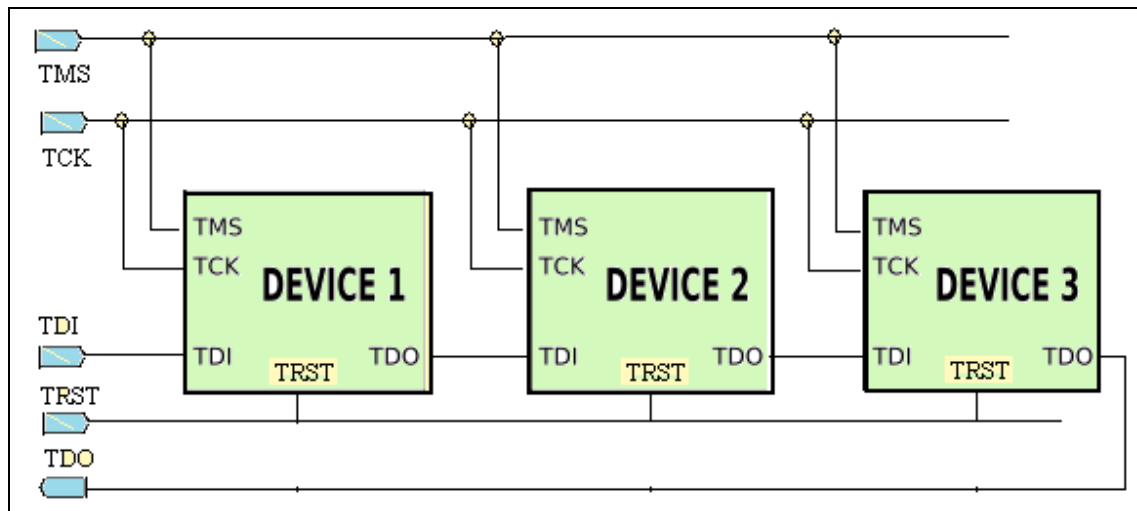


Fig. 10-30. Simplified model of the **JTAG** bus. The connector pins are: **TDI** (Test Data In), **TDO** (Test Data Out), **TCK** (Test Clock), **TMS** (Test Mode Select) and an optional **TRST** (Test Reset).

vii. Serial Peripheral Interface (SPI) Bus.

The Serial Peripheral Interface (**SPI**) is a synchronous serial data bus, which was introduced by Motorola. It is used for short distance, single master communication, for example in embedded systems, sensors, and SD cards. Devices communicate in master/slave mode where the master device initiates the data frame. Multiple slave devices are allowed with individual slave select lines. Sometimes SPI is called a *four-wire* serial bus, contrasting with three-, two-, and one-wire serial buses. SPI is often referred to as SSI (Synchronous Serial Interface).

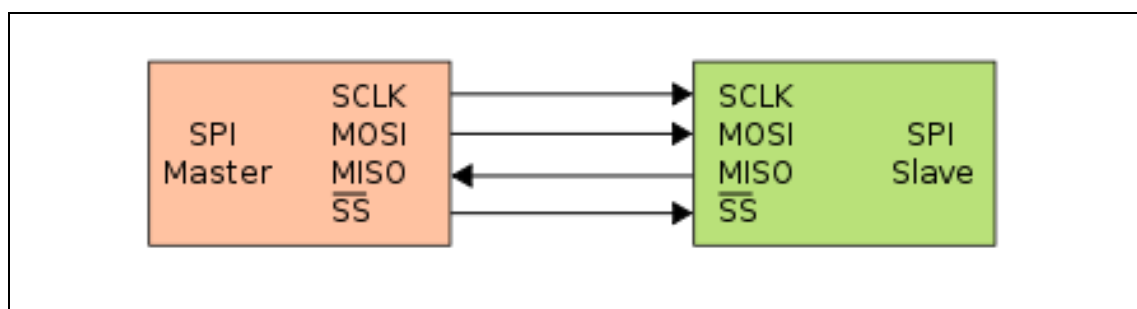


Fig. 10-31. Serial Peripheral Interface (**SPI**)

The SPI bus specifies four logic signals:

- SCLK : Serial Clock (output from master).
- MOSI : Master Output, Slave Input (output from master).
- MISO : Master Input, Slave Output (output from slave).
- SS : Slave Select (active low, output from master).

viii. Local Interconnection Network (LIN)

The *Local Interconnection Network* (LIN) standard defines a low cost, serial communication network for automotive distributed electronic systems. LIN is a complement to the other automotive multiplex networks, including the *Controller Area Network* (CAN), but it targets applications that require networks that do not need excessive bandwidth, performance, or extreme fault tolerance. LIN enables a cost-effective communication network for switches, smart sensors and actuator applications inside a vehicle. The communication protocol is based on the UART data format, a single-master/multiple-slave concept, a single-wire (plus ground) 12V bus, and a clock synchronization for nodes without a precise time base (i.e., without a crystal or resonator).

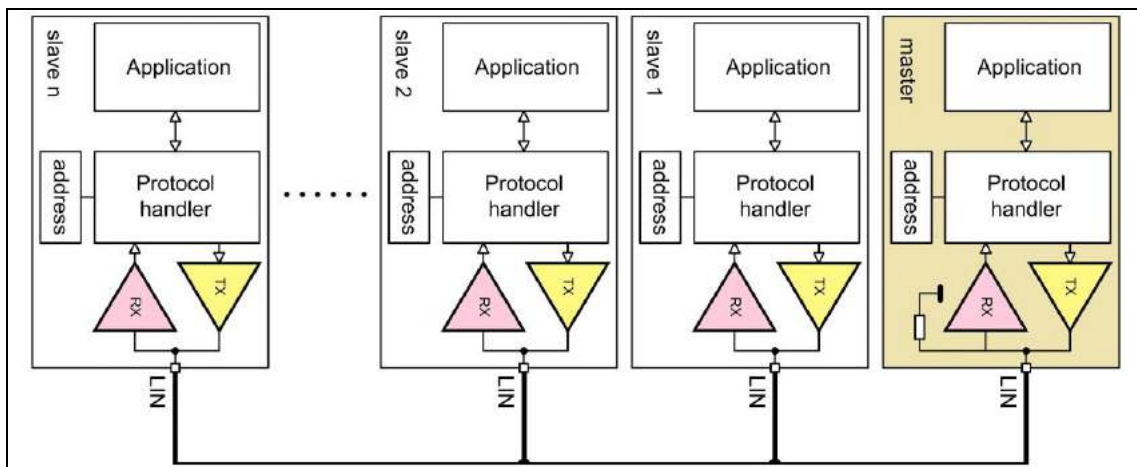


Fig. 10-32. Local interconnection network (LIN) master-slave architecture.

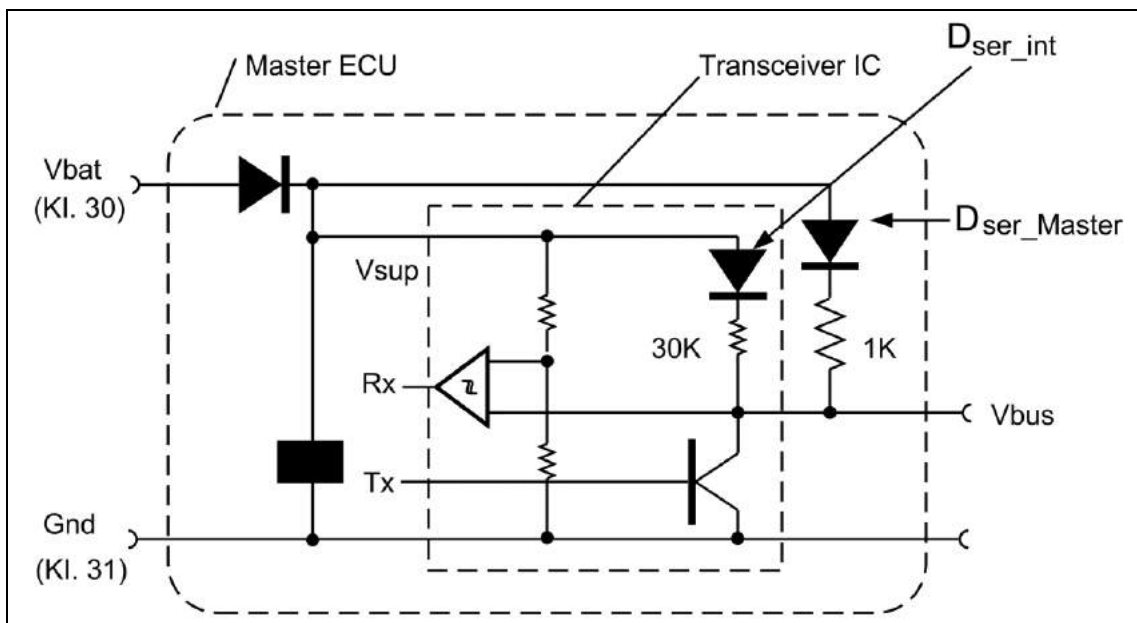


Fig. 10-33. Serial Peripheral Interface (LIN) transceiver structure.

As shown in the figure above, LIN bus is a single-wire bus connected via a termination resistor to the positive battery node V_{bat} . The bus is terminated with a pull-up resistance of $1k\Omega$ in the master node, and typically $30k\Omega$ in a slave node. LIN versus CAN.

10-5.9. Selecting a Serial Communication Bus

Defining a serial bus is not a trivial task. Besides data rate, bit sequence, and voltage, one needs to consider the following:

- How the peripheral device gets selected (by hardware through the chip-select input or by a software protocol).
- How the peripheral device stays synchronized with the computer or microcontroller (μC) through a hardware clock line or through clocking information embedded in the data stream.
- Whether data is transmitted on a single line (switching between "high" and "low") or a two-line differential connection (both lines changing their voltage simultaneously, but in the opposite direction).
- Whether both ends of the communication line are electrically terminated with a matching impedance (with differential signaling), or whether they are left un-terminated or terminated at one end only (with single-ended buses).

The following table provides top-level characteristics for serial interface standards by which two or more digital devices can be connected for communication. Design engineers can use the table to compare interface options for their application based on the design constraints like number of signal lines, network size, speed, distance, noise immunity, fault tolerance and reliability.

Table 10-7. Specifications of some serial communication network standards

	I²C	SMBUS	SPI	CAN
Network Concept	multiple masters, multiple slaves	Multiple masters, multiple slaves	single master, multiple slaves	multiple masters, multiple slaves
Number of Signal Lines	2, (SCL, SDA)	2, (SMBCLK, SMBDAT)	4, (CS, SI, SO, SCK)	2 (CAN_H, CAN_L)
Network Interface	open drain, or master pull-up	open drain, or master pull-up	Push-pull with tristate	Differential open drain/source
Voltage	1.8 to 5.5V,	2.7V to 5.5V	1.8V to 5.5V	$V_{DD} - 4.5V$.
Logic level	1.5V-3V	0.8V, 2.1V	V_{DD}	Differential >1.5V
Transmission	half-duplex	half-duplex	full-duplex	half-duplex
Address	7 bits,	7 bits,	N/A	identifier 11 bits

Data Rate	up to 3.4M bps	10k to 100k bps	0 to 10 M b/s	0 to 1M bps
Access Time	95 μ s ~23 μ s	95 μ s	N/A	19 μ s
EDC	N/A	Packet Code	N/A	15-bit CRC
Collision Detection	Yes	Yes	N/A	CSMA/CD

The following table shows the matrix of variations using popular bus systems. Only 4 of the possible 16 combinations are currently known commercially.

Table 10-8. Serial Bus Systems Overview

		ADDRESSING (SELECTION)			
SYNCHRONIZATION		PROTOCOL	CHIP-SELECT LINE		IMPEDANCE
SELF-CLOCKING		1-Wire [®] , LIN bus, SensorPath [™]			NOT MATCHED
		RS-485, LVDS, CAN, USB 2.0, FireWire [®]			MATCHED
CLOCK LINE					
		I ² C, SMBus [™]		SPI [™] , MICROWIRE [™]	NOT MATCHED
		SINGLE-ENDED	DIFFERENTIAL	SINGLE-ENDED	
TRANSMISSION MODE					

Beyond these parameters, the application can add further requirements such as the method of power delivery, isolation, noise immunity, the maximum distance between the μ C (master) and the peripheral device (slave), or the structure of the cabling (linear, star, insensitive to reversal of wires). These requirements lead to applications such as building automation, industrial control, and reading of utility meters for which special standards have been developed.

Requirements for Applications from Circuit Board to Backplane

A serial bus system for peripheral functions must not add any significant burden on the system. In particular,

- The connection must be easy to route (the fewer signals the better).
- The protocol must be easy to implement in software (or natively supported by the chosen μ C/ μ P).
- There should be an adequate selection of device functions.
- The bus must be easy to expand.

10-6. Summary

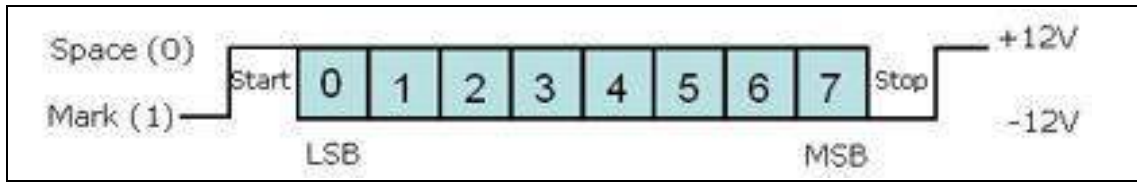
Data Transmission means movement of the bits over a transmission medium connecting two devices. In this chapter, we presented the general architecture of serial and parallel communications and their protocols. Parallel transmission sends each bit using a separate wire. In addition, extra wires are needed to transfer the data between the sender and receiver. These handshake signals allow the data to be transferred in the correct sequence. Computers often send data in parallel form because it is fast. An example of a parallel format is the Centronics interface. Serial data is slower than parallel, but suited to long distances. There is no need for extra wires to convey handshake signals, as the data is packaged with additional checking information (prefixed and suffixed bits). An example of a serial format is RS232.

Data within a computer system is usually transferred in a **parallel** manner. Communication between computers and with other devices is usually carried out in a serial manner. Asynchronous serial transmission requires an overhead of two or three bits per character, and is, therefore, significantly less efficient than synchronous serial transmission. However, the synchronous **serial** communication is perhaps the most common form of data communications that must be handled with external devices. Regardless of whether we use **synchronous** or **asynchronous** communications techniques and regardless of how well we engineer a communication link, there is always the possibility that an error will occur when we transmit data from one device to another. There are a number of error detection techniques in use.

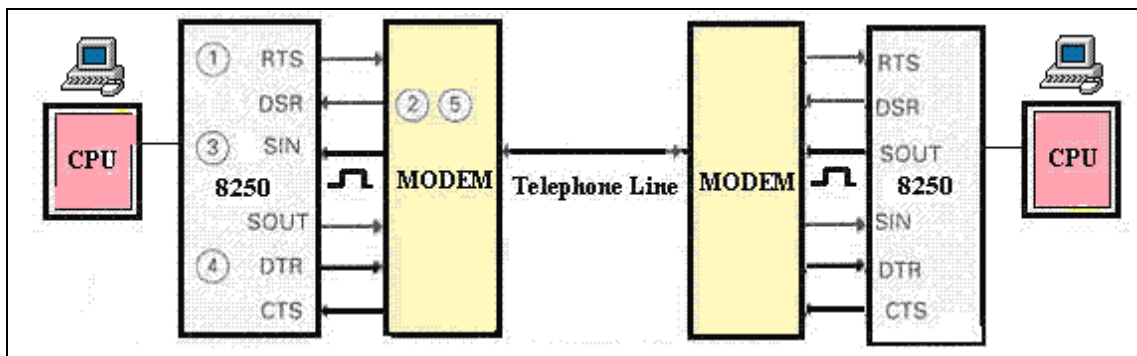
A **protocol** is a set of rules which governs how data is sent from one point to another. In data communications, there are widely accepted protocols for sending data. Both the sender and receiver must use the same protocol when communicating.

RS232 is an **asynchronous serial** communications interface, widely used on computers. Asynchronous means it doesn't have any separate synchronizing clock signal, so it has to synchronize itself to the transmitter data. It does this using the 'START' and 'STOP' pulses. The signal itself is slightly unusual for computers, as rather than the normal 0V to 5V range, it uses +12V to -12V - this is done to improve noise immunity. You can use the MAX232 chip, to interface between 5V logic levels and the +12V/-12V of RS232. There are various data types and speeds used for RS232. The most common type in use, known as

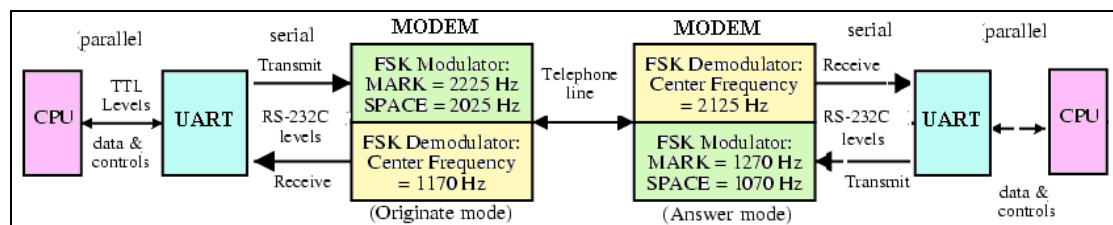
8N1 - the 8 signifies '8 Data Bits', the N signifies 'No Parity' (may be also 'Even or Odd Parity') and the final 1 signifies '1 Stop Bit'.



Modems are devices that allow digital data signals to be transmitted across an analogue link. As has already been discussed, the connections provided by telephone companies for the use of speech via dial up telephones is analogue based. However, with the development of the Internet, it is now standard practice to use the dial-up telephone connection to access these services.



As the computer uses digital information, and the telephone line uses analogue signals, a device is needed which converts the digital data from the computer and converts it to analogue tones (within the voice channel range of 300Hz to 3400Hz) so that the signals can travel across the dial up speech connection. At the other end, the signals are converted back to digital. The device that converts digital signals to analogue for transmission across a dial-up telephone connection, and then converts them back again, is a modem.



10-7. Problems

10-1) define the term "protocol"

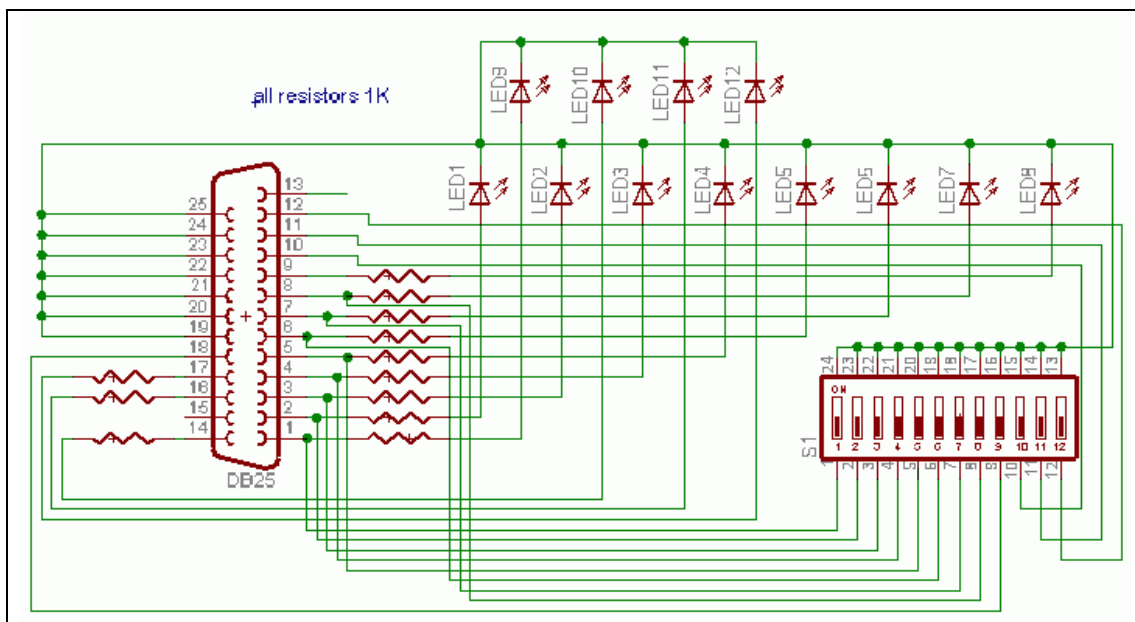
10-2) describe the transfer of data using asynchronous transmission

10-3) explain how data transfer differs in synchronous transmission

10-4) Show how to connect and initialize an external serial device, to COM1, with given baud rate and parity. Write the software driver that you'll use, in C-language.

Hint: Browse the hyper terminal program, which is supplied with your Windows operating system

10-5) Consider the parallel port test circuit shown below. Show how to build the project (*pptest*), and run the resultant *pptest.exe* under DOS such that when you set the dip switches to "11111111", the LED1 to LED8 in the hardware will glow.



10-6) Design a parallel printer ISA interface card on the basis of the 8255. Use a 3-to-8 decoder (74137) for assigning the appropriate port address.

10-7) Calculate the CRC of the sequence 1111151161111 using the CCITT standard polynomial generator.

10-8. References

- [1] C. C. **Wang** and D. **Sklar**, "Performance of the Turbo Coded System with DPSK Modulation Using Enhanced Decoding Metrics and Matched Channel Side Information," *Proceedings of International Communications Conference*, April 2002.
- [2] M. **EL-SABA**, Introduction to Microprocessors and interface circuits, Hakim Press, 2008.
- [3] S. **Haykin**, Communication Systems, 4th Edition, 2002.
- [4] A. V. **Oppenheim**, et. al., Signals and Systems, 2nd Ed., Prentice-Hall, 1996.

Chapter
11

Communication Networks & Their Models

Contents

- 11-1. Introduction**
- 11-2. ISO / OSI Seven Layer Model**
 - 11-2.1. The Seven Layers
 - 1- Physical Layer
 - 2- Data Link Layer
 - 3- Network Layer
 - 4- Transport Layer
 - 5- Session Layer
 - 6- Presentation Layer
 - 7- Application Layer
 - 11-2.2. Sending Data Via the OSI Model
 - 11-2.3. Network Protocols
 - 11-2.4. Protocol Data Units (PDUs)
 - 11-2.5. Network Components
- 11-3. Systems Network Architecture (SNA)**
- 11-4. National Transportation Control Interface Protocol (NTCIP)**
- 11-5. Telecommunication Management Network (TMN)**
- 11-6. Summary**
- 11-7. Problems**
- 11-8. Bibliography**

Chapter

11

Communication Networks & Their Models

11-1. Introduction

Communications networks are key infrastructures of the information society with high socio-economic value. Such networks contribute to the correct operations of many critical services, from healthcare to finance, scientific research, transportation, video broadcasting and entertainment.

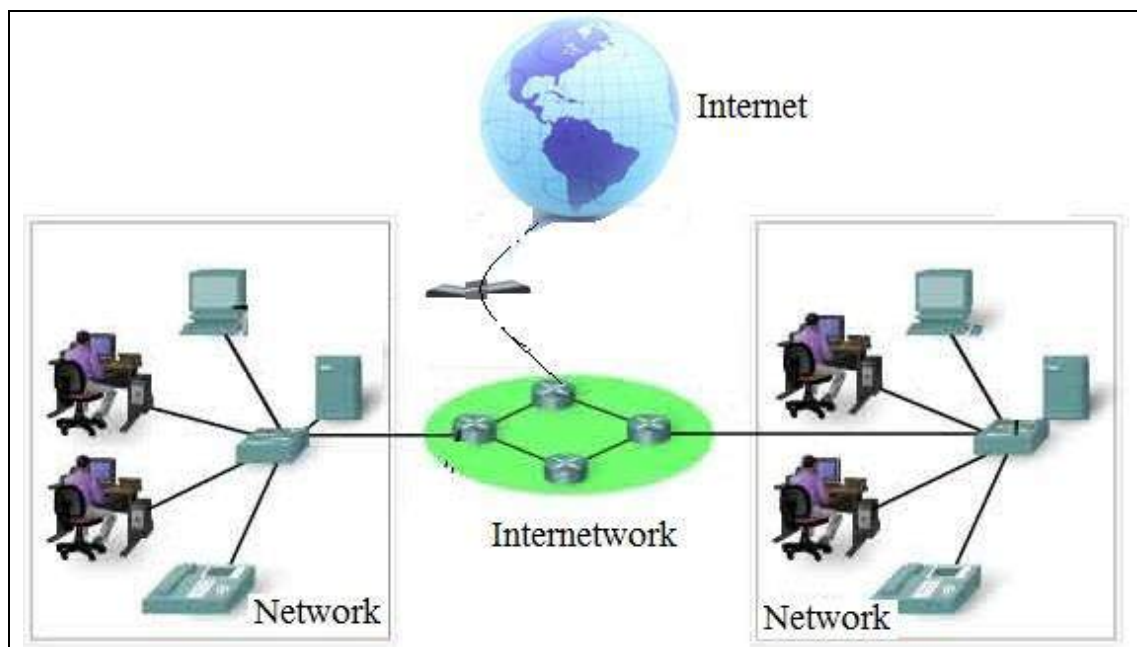


Figure 11-1 – Communication networks

Networking is the practice of linking two or more computers or devices with each other. In order to make a number of computers communicate with one another, on a meaningful basis, there are many definitions that need to be established, including:

- Communication Protocols and contention schemes
- Synchronous/asynchronous transmission mode
- Bit rates
- Error detection and correction techniques
- Electrical signaling methods

- Cable types and connector types
- Modulation techniques
- Software structure to support applications programs.

The common language between networking communication devices is known as **protocols**. The communication networks are usually described by a stack of protocols that defines a framework for implementing the communication process. In the past, we experienced an explosion of different networks, many of which were proprietary in nature and not compatible with each other. Since different computer manufacturers tended to specialize in different aspects of computing, users seldom had the luxury (nor the risk) of the single vendor environment. It also became apparent that no, single networking solution was universally applicable to all environments and hence a number of different standards would always be in existence. The obvious course of action was to find a way to rationalize the development of networking standards so that there may be some hope of creating links between different networks. Unless there is a systematic approach to network development, it is extremely difficult to link different networks together and to create low cost networking hardware. The problem with trying to rationalize networking standards is that it is a very complex task to decide for any individual networking requirement. That is, should a standard define just cables or connectors or cables, connectors, signaling techniques and contention schemes? Where does one draw the boundaries for standards? The International Standards Organization (**ISO**) tackled this problem by developing a framework referred to as Open Systems Interconnect or "**OSI**" model.

11-2. Open Systems Interconnect (OSI) Model

The OSI model is organized into seven hierarchical layers. Control is passed from one layer to the next starting at the application layer and proceeding down to each successive layer and back as required for any given process. Most of the functionality of the OSI model exists in all communications systems - however, two or three layers may be combined into one. The most significant role of the OSI model is to serve as a reference for the development of other protocol stacks. The following figure depicts the OSI Model Protocol Stack.

Table. 1-1. The OSI model seven layers and their data units

LAYER	FUNCTION	DATA
7. Application	Software application programs	Data stream
6. Presentation	Data representation and encryption	Data stream
5. Session	Inter-host communication	Data stream
4. Transport	End-to-end connections and reliability	Segment
3. Network	Path determination and logical addressing	Packet
2. Data Link	Physical addressing	Frame
1. Physical	Equipment, Media, and signal transmission	Bit

11-2.1. The Seven Layers

The modern computer communication links are based on the open-system interconnect (**OSI**) model. The objective of this framework was to place all the requirements, for making a number of computers communicate with one another, into seven functional groups called *layers*. The end result of the OSI 7-layer Communications model is shown in Figure 11-1.

The **Physical** Layer defines the electrical and physical specifications for network devices. This includes the specifications of network cards, hubs, cable, and voltage levels. The main functions of this layer is the signal **modulation** and **transmission** over a physical channel (cables, or air).

The **Data Link** Layer provides the procedural means to transfer data between network nodes and to detect and correct errors. Examples of data link protocols are **Ethernet** and Point-to-Point (**PPP**) protocol.

The **Network Layer** performs network **addressing** and **routing** functions, and might also perform assembly (**encapsulation**) and fragmentation of data. Routers, which are sending data throughout the network, operate at this layer. The best-known example of a Layer 3 routing protocols is the Internet Protocol (**IP**).

The **Transport Layer** controls the reliability of a given link through flow control, data segmentation/assembly, and error control. Although not developed under the OSI Model, typical examples of Layer 4 are the Transmission Control Protocol (**TCP**) and User Datagram Protocol (**UDP**). The Session Layer controls the dialogues between computers. It establishes, manages and terminates the connections between the local and remote application, and commonly implemented in application environments that use remote procedure calls.(**RPC**).

The **Presentation Layer** works to transform (format and encrypt) data into the form that the application layer can accept.

The **Application Layer** is the closest OSI layer to the end user, so that he can interact with the software application. Some examples of application layer include Hypertext Transfer Protocol (**HTTP**), and Simple Mail Transfer Protocol (**SMTP**).

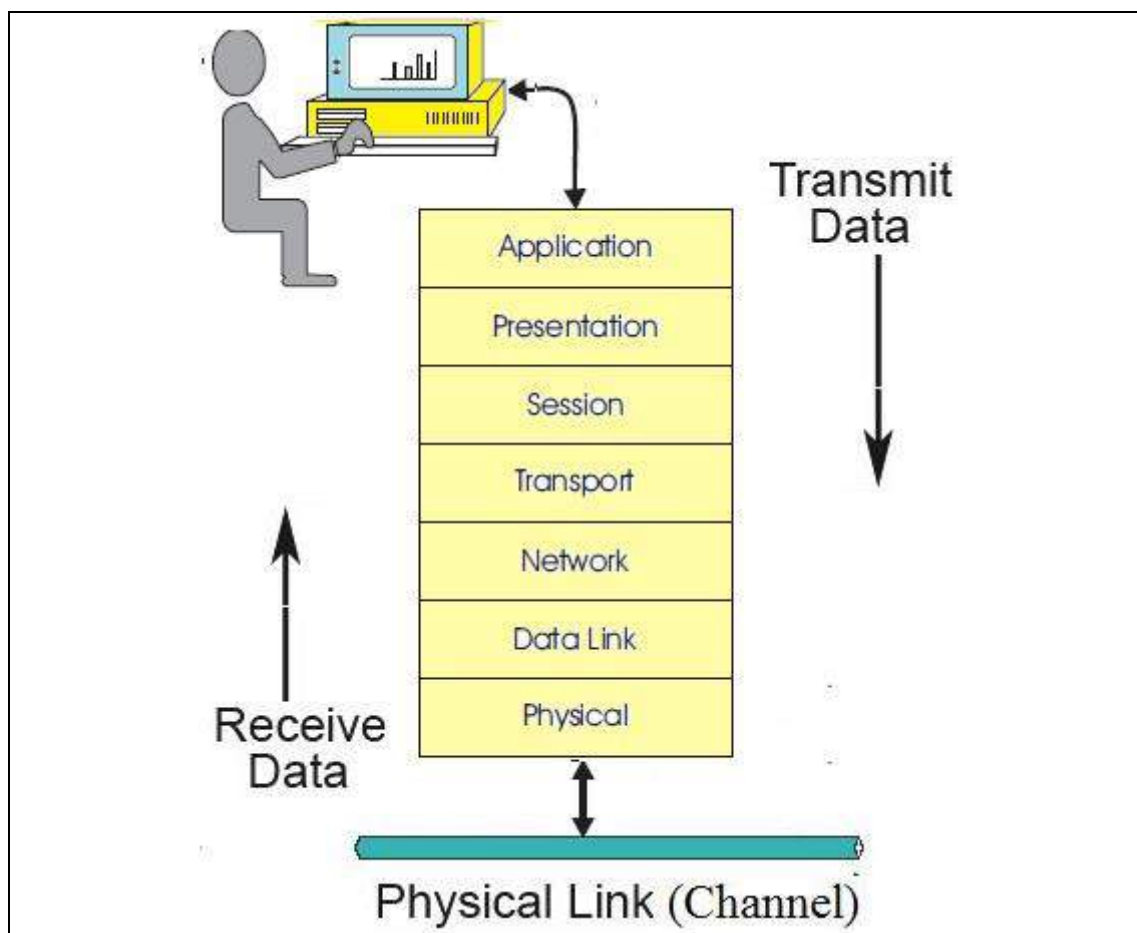


Figure 11-2 - Seven Layer ISO/OSI Model for Communications

It is important to note that the OSI model is not a networking standard in its own right. It is the framework, upon which, the development of communications standards is based. The choice of seven layers is somewhat arbitrary but is based upon many practical considerations. Each layer of the OSI model defines a number of related functions that must be enacted in order to take information from the layer below/above, process it and feed it to the layer above/below.

11-2.2. Sending Data Via the OSI Model

Each layer acts as though it is communicating with its corresponding layer on the other end. In reality, data is passed from one layer down to the next lower layer at the sending computer, till its finally transmitted onto the network cable by the Physical Layer. As the data is passed down to a lower layer, it is encapsulated into a larger unit (in effect, each layer adds its own layer information to that which it receives from a higher layer). At the receiving end, the message is passed upwards to the desired layer, and as it passes upwards through each layer, the encapsulation information is stripped off.

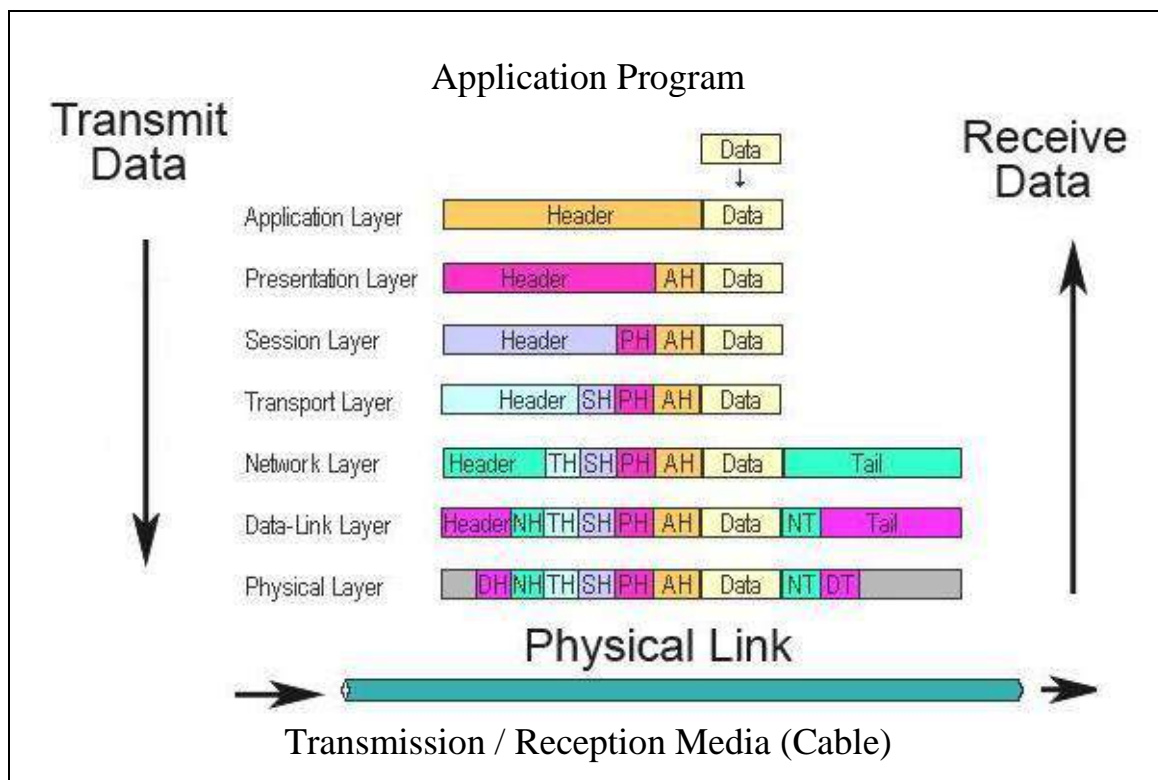


Figure 11-3 – Details of the packets of the OSI model seven layers

11-2.3. OSI Model Protocols

The OSI model provides a conceptual framework for communication between computers, but the model itself is not a method of communication. A common communication language between networking computers and communication devices are known as protocols. Actual communication is made possible by using communication **protocols**.

In the context of data networking, a **protocol** is a formal set of rules and conventions that governs how computers exchange information over a network medium. A protocol implements the functions of one or more of the OSI layers. A wide variety of communication protocols exist, but all tend to fall into one of the following groups: *LAN protocols*, *WAN protocols*, *network protocols*, and *routing protocols*. *LAN protocols* operate at the network and data link layers of the OSI model and define communication over the various LAN media. *WAN protocols* operate at the lowest three layers of the OSI model and define communication over the various wide-area media. *Routing protocols* are network-layer protocols that are responsible for path determination and traffic switching. Finally, *network protocols* are the various upper-layer protocols that exist in a given protocol suite

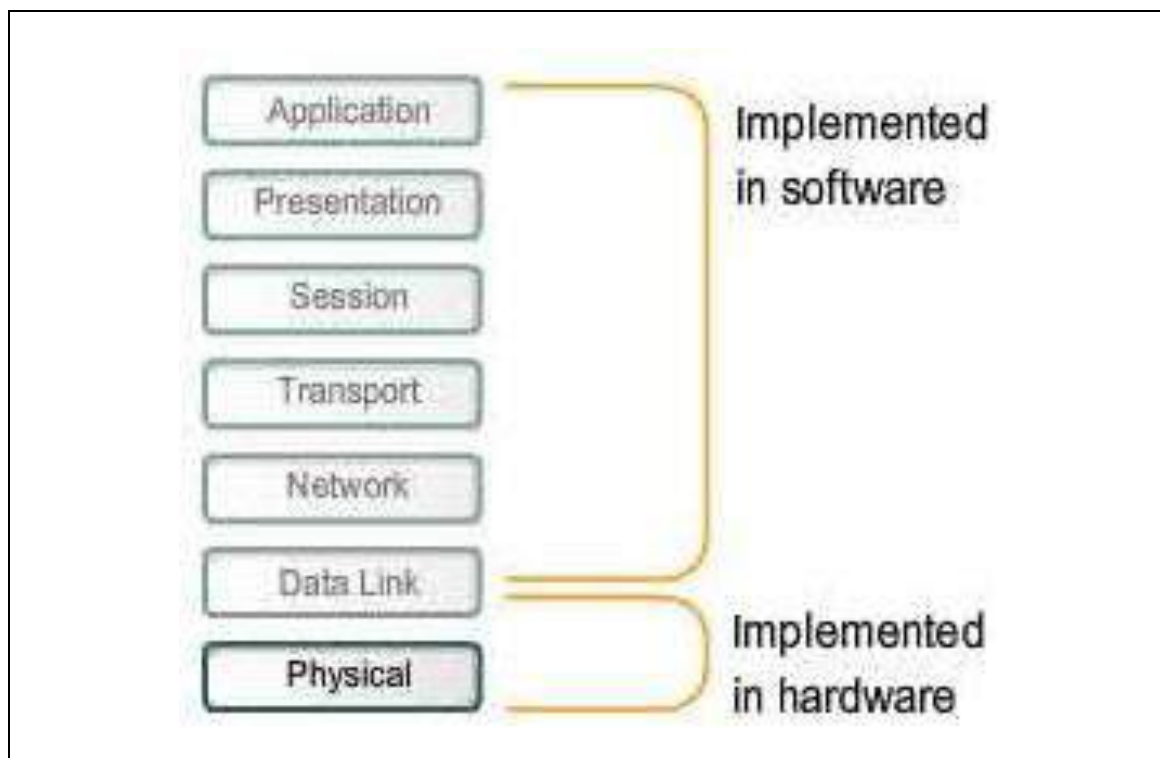


Figure 11-4/ Details of the packets of the OSI model seven layers

Table 11-2. *The OSI model protocol stack*

OSI Protocols
7. Application Layer - HTTP, POP3, SMTP, DHCP, DNS, FTP, IMAP4, NNTP, SNMP, SSH, TELNET, ...
6. Presentation layer – SSL, WEP, WPA, ...
5. Session layer – Logical Ports 21, 22, 23, 80 etc...
4. Transport - TCP, UDP and SPX,....
3. Network - IPv4, IPV6, IPX, OSPF, ICMP, IGMP and ARPMP
2. Data Link - 802.11 (WLAN), Ethernet , Wi-Fi, WiMAX, ATM, Token Ring, Frame Relay, PPTP, L2TP and ISDN, ...
1. Physical - Hubs, Cables, Optical Fiber, SONET, Coaxial Cable, Twisted Pair Cable and Connectors

11-2.4. Protocol Data Units (PDUs)

Protocol Data Units (PDUs) are relevant in each of the first 4 layers of the OSI model as follows:

- Layer 1 (Physical Layer) PDU is the **bit** or, more generally, **symbol**
- Layer 2 (Data Link Layer) PDU is the **frame**
- Layer 3 (Network Layer) PDU is the **packet**
- Layer 4 (Transport Layer) PDU is the **segment** for TCP, or the **datagram** for UDP
- Layer 5-6-7 (Application Layer) PDU is the **message**

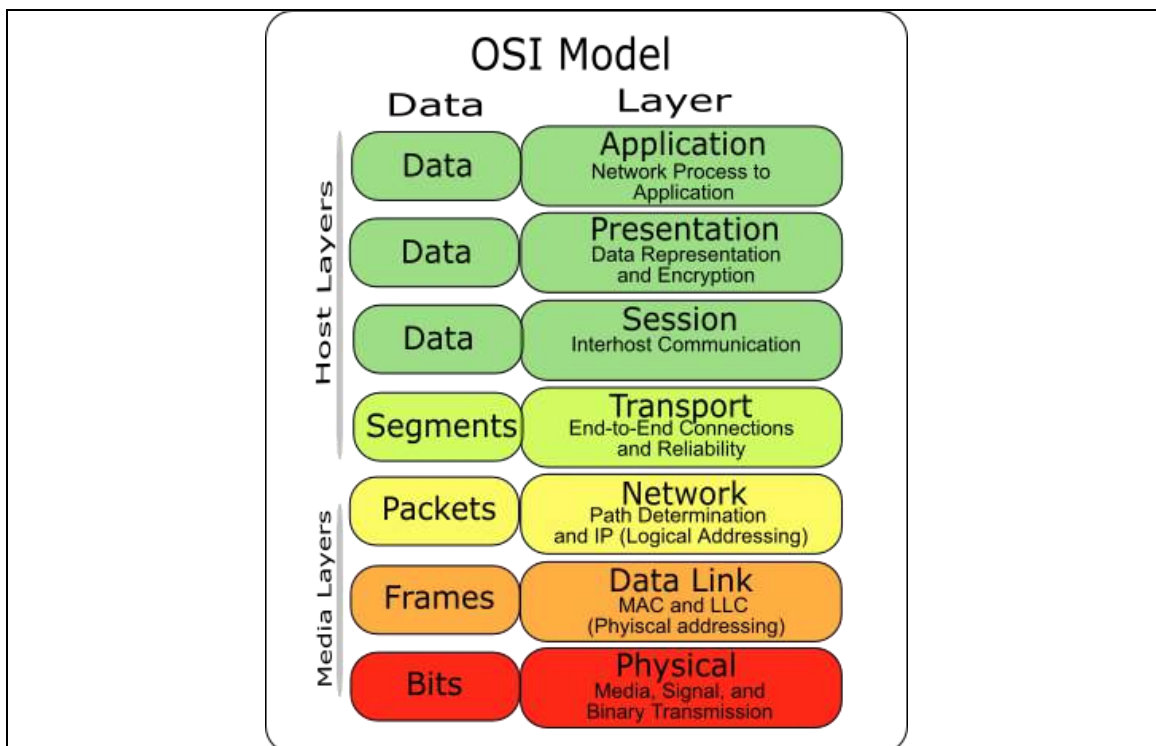


Figure 11-5 – Protocol data units (PDUs) in the OSI model

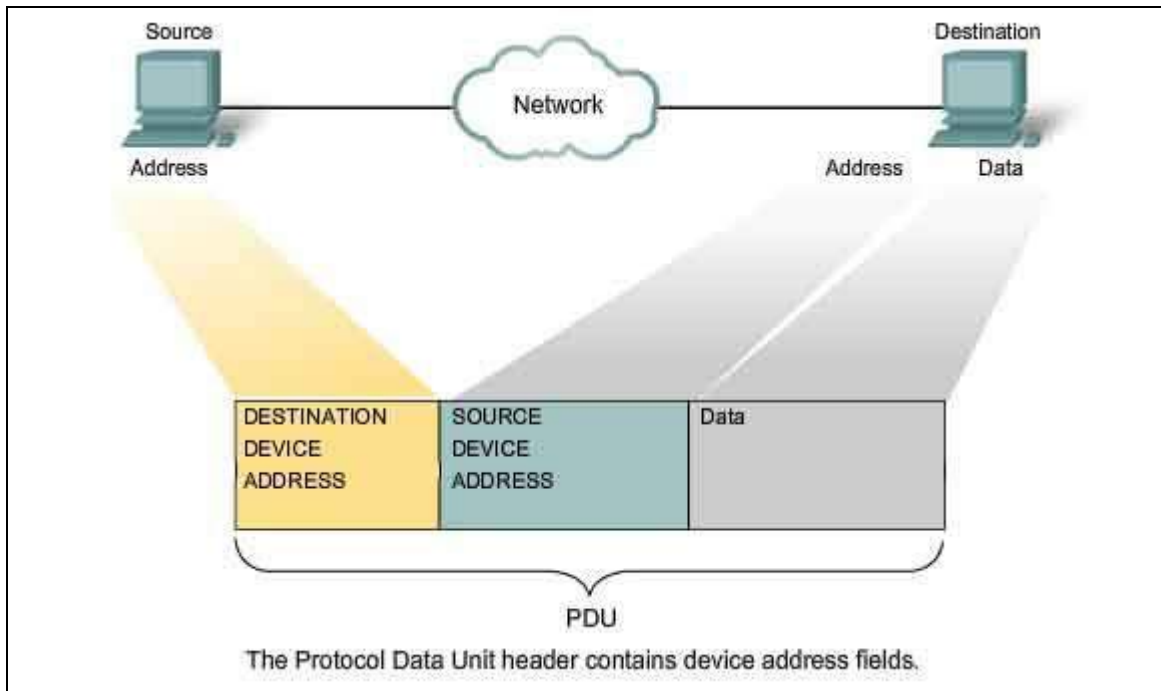


Figure 11-6 – Internal structure of a PDU in the OSI model

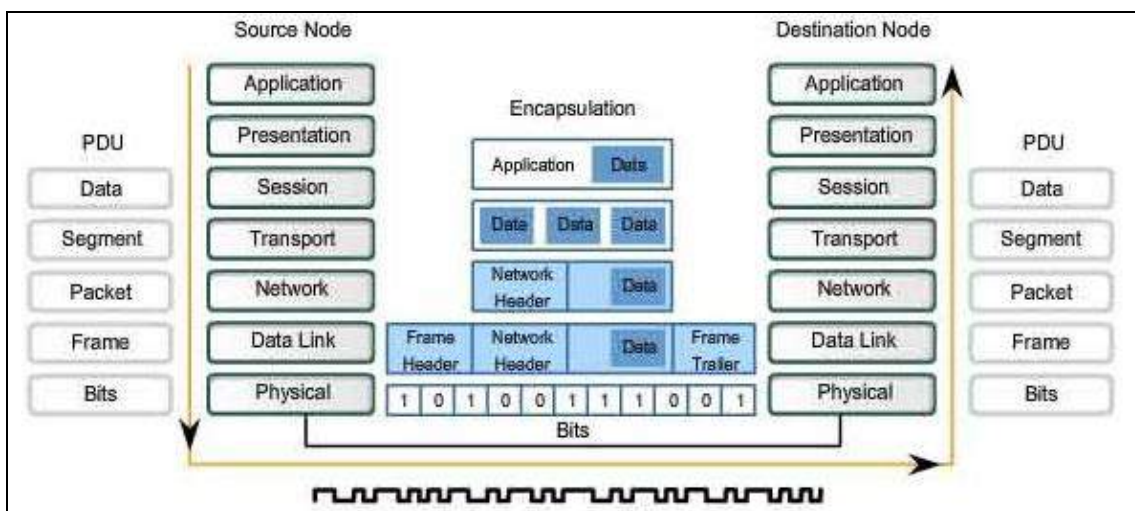


Figure 11-7 – Details of data encapsulation units in the OSI model

11-2.5. Network Components

Any communication network consists of computer nodes and some switching and routing devices. Some of these devices can be implemented as software and others are hardware components. Each of these components will work in a certain layer of the OSI model. Repeaters, Hubs and Switches are typically hardware devices. Hubs and repeaters are found in the Physical Layer. Switches /Bridges and Wireless Access Point are found in the Data Link Layer. Routers are found in the Network Layer. Gateways are found in All 7 of the OSI Layers

Hubs & Repeater (Physical layer 1)

- Copies bits from one network to another
- Does not look at any bits
- Allows the extension of a network beyond physical length limitations

Bridge & Switches (Layer 1 and 2)

- Copies frames from one network to another
- Can operate selectively - does not copy all frames
- Extends the network beyond physical length limitations.

Router (Layer 1 and 2) can be implemented in hardware or software.

- Copies packets from one network to another.
- Makes decisions about what *route* a packet should take

Gateway (All 7 layers) implemented in software

- handle data conversions above the network layer.
- Conversions include: encapsulation, and protocols encryption

OSI Layer	Application/Example	Central Device/ Protocols	
Application (7) Serves as the window for users and application processes to access the network services.	End User layer Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management	User Applications SMTP	G A T E W A Y Can be used on all layers
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	Syntax layer encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • Character Set Translation	JPEG/ASCII EBDIC/TIFF/GIF PICT	
Session (5) Allows session establishment between processes running on different stations.	Synch & send to ports (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.	Logical Ports RPC/SQL/NFS NetBIOS names	
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	TCP Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	TCP/SPX/UDP	
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	Packets ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting	Routers IP/IPX/ICMP	
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	Frames ("envelopes", contains MAC address) [NIC card — Switch — NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control	Switch Bridge WAP PPP/SLIP	
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	Physical structure Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts	Hub Land Based Layers	

Figure 11-8. Details of data encapsulation units in the OSI model

11-3. Systems Network Architecture (SNA)

The IBM Systems Network Architecture, commonly known as **SNA**, is IBM's proprietary networking architecture. **SNA** was created in 1974. An IBM customer could acquire hardware and software from IBM and lease private lines from a common carrier to construct a private network. SNA is IBM's proprietary protocol architecture. Its structure precedes the OSI 7-layer framework, but it does contain a number of similarities. The OSI model and SNA model are shown side by side in Figure 11-4.

The SNA protocol is based upon synchronous serial transmission, with framing and error control carried out by the IBM implementation of HDLC, known as **SDLC** (Synchronous Data Link Control).

The basic entities within SNA are called Logical Units (LUs) and these can represent either end users or applications programs. Communications within the SNA system is between these Logical Units. It is the Logical Units which are assigned addresses and not the end users as such. Logical Units are therefore referred to as Network Addressable Units or NAUs.

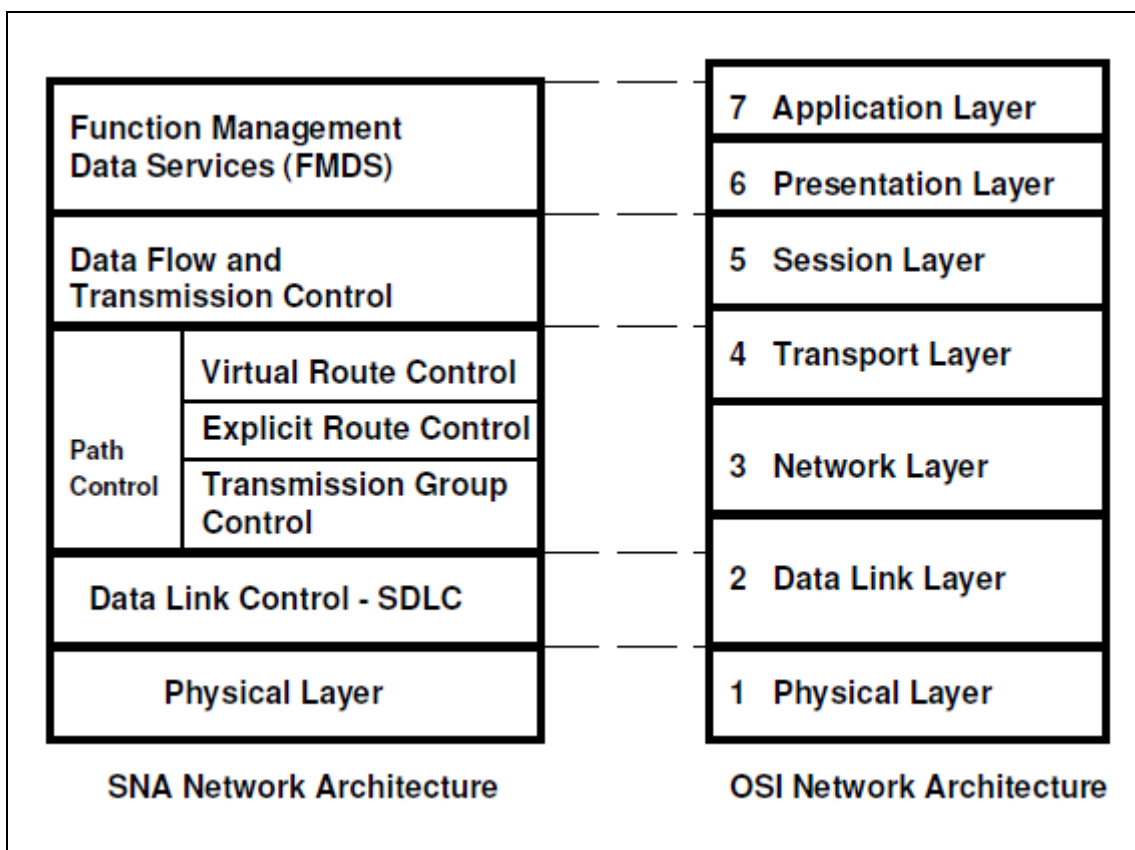


Figure 11-9 - IBM Systems Network Architecture versus OSI

Although the levels of functionality differ between OSI and SNA, the end results are not dissimilar and consist of a large range of services that are available to application programs and end users. SNA not only preceded the OSI framework, but also the push for manufacturing networking, capable of handling industrial control systems. The FMDS layer therefore does not provide services for communications with the majority of manufacturing devices.

Although SNA is a proprietary networking system, there are a range of networking gateways that will enable other computer manufacturers to interface their systems to SNA. The SNA strategy was vendor-driven rather than end-user-driven. The fact that SNA was not universally adopted is another illustration of how difficult it is, even for very large organizations, to develop networking standards that gain widespread acceptance.

11-4. National Transportation Control Interface Protocol (NTCIP)

Telecommunications hardware generally utilizes layers **one** and **two** of the OSI protocol stack. Modems, multiplexers, and network components (bridges, routers, switches), as well as coders-decoders (codecs) are also examples of devices that exist at the physical and data link layers of the OSI protocol stack. However, some communication hardware devices are designed to operate at higher layers. The so-called National Transportation Control Interface Protocol (NTCIP) is one of the most famous protocol stacks, that have been based on OSI model. The NTCIP is specially developed with embedded telecommunication standards, such that communication system designers can simply use the pre-defined telecommunication standards.

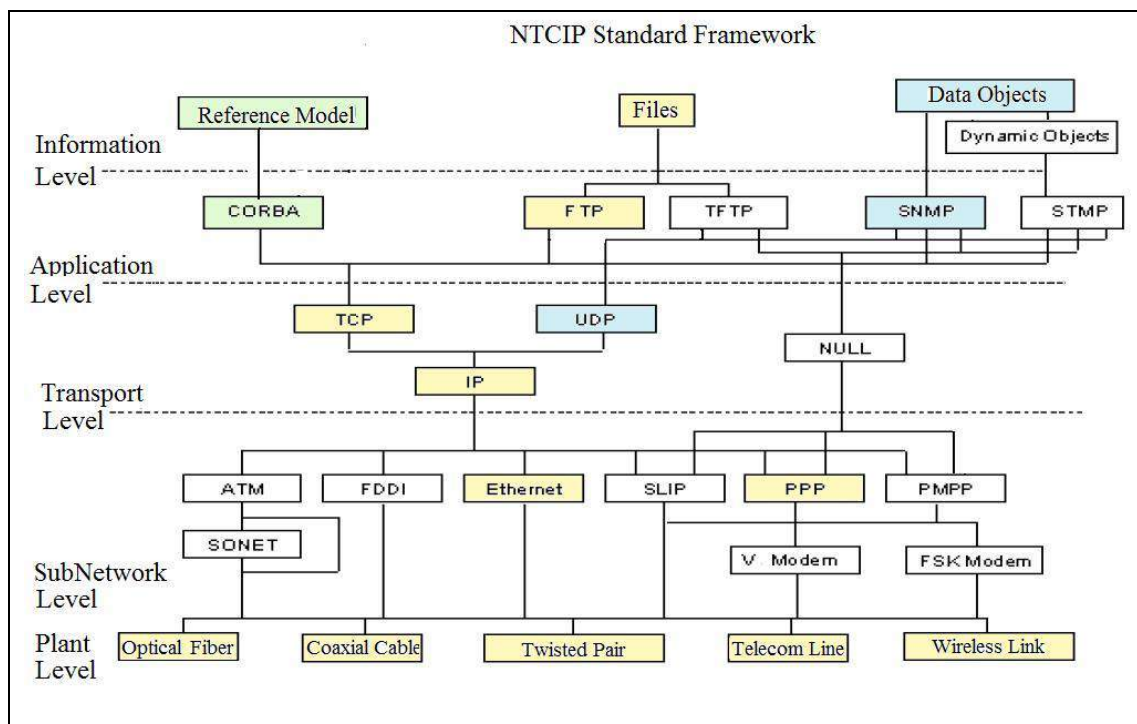


Fig. 11-9..The NTCIP protocol stack.

11-5. Telecommunication Management Network (TMN)

The Telecommunications Management Network (TMN) is a protocol model defined by ITU-T for managing open systems in a communications network. TMN is based on the OSI management specifications in ITU-T Recommendation series. The TMN model is used as the guiding model for achieving interconnectivity and communication across heterogeneous systems and telecommunication networks. It is widely recognized among industry analysts and equipment and software vendors. Figure 1 shows the TMN model.

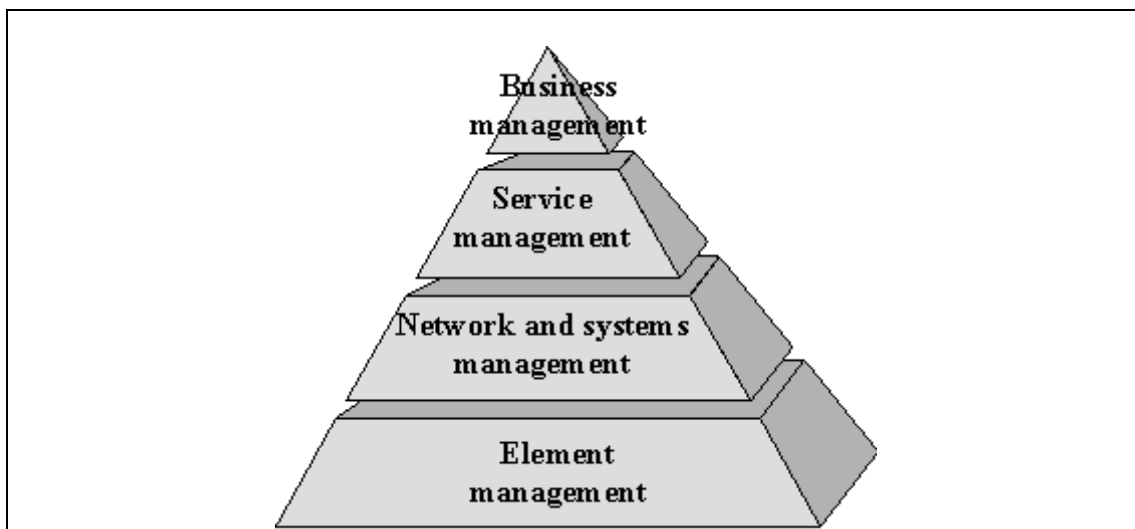


Fig. 11-10.. Telecommunications Management Network (TMN) model

The framework identifies four logical layers of network management:

Business management

Includes the functions related to business aspects, analyzes trends and quality issues, for example, or to provide a basis for billing and other financial reports.

Service management

Handles services in the network: definition, administration and charging of services.

Network management

Distributes network resources, performs tasks of: configuration, control and supervision of the network.

Element management

Handles network elements including alarm management, handling of information, backup, logging, and maintenance of hardware and software.

A network element provides agent services, mapping the physical aspects of the equipment into the TMN framework. Network and systems management resembles network management done in all organizations with computer networks. Also the tools and processes are fairly similar. Here the tasks fall into the categories performance management, configuration management, accounting management, fault management, and security management. It involves issues like can the network take the load of the services, are all the servers up and running all the time, how bandwidth is allocated to different services, are all the computer systems safe and backed up, etc.

The rapid advancement of technology is enabling new kinds of methods for end users to access information networks. The old analog modem is challenged by technologies running on copper, fiber, and coaxial networks or transmitted from base stations and satellites over the air. Table 3 summarizes a few of these and some of their main characteristics.

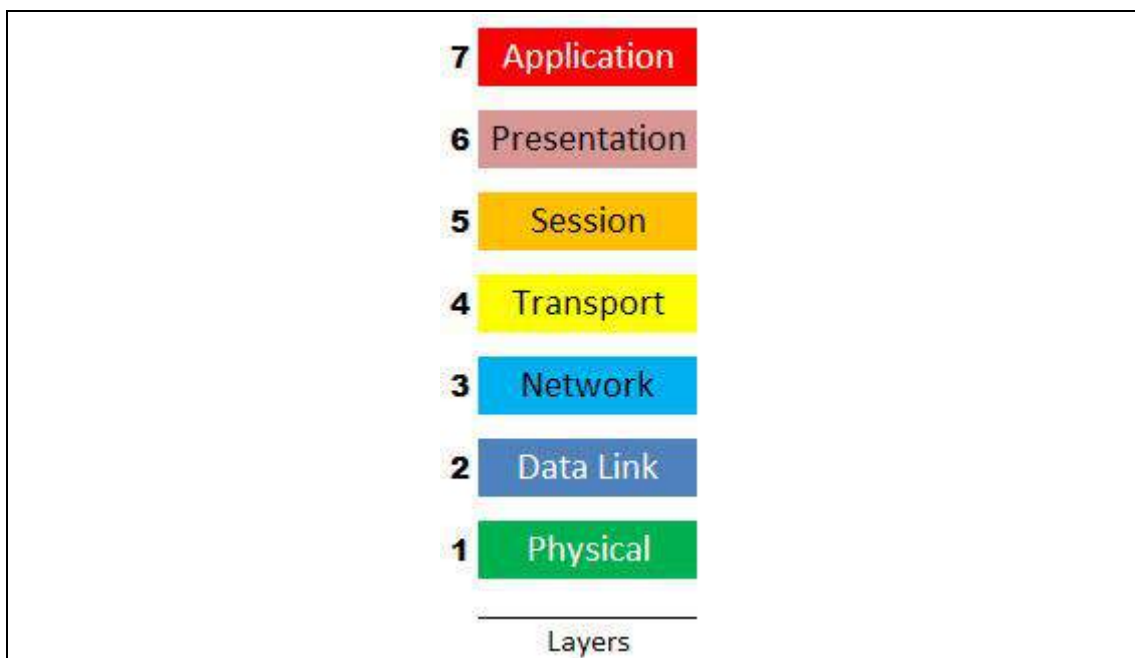
Table 11-3 Communication Access Technologies

Type	Data rate	Transmission medium	Terminals	End user equipment Price
Analog modems	14.4 Kbps to 56 Kbps	copper lines	External boxes, PC cards.	Less than \$30 - \$500
ISDN	2 x 64 Kbps, 30 x 64 Kbps	copper lines	External modems, PC cards, routers	\$150 - \$500
xDSL	Upstream 64 kbps to 1.5 Mbps Downstream 1 Mbps to 9 Mbps	copper lines	internal and external modems	\$250 - \$1,000
Cable modems	Upstream 128 Kbps to 10 Mbps Downstream 10 Mbps to 27 Mbps	Hybrid fiber/coax	External box	\$300 - \$900
1st and 2nd generation mobile	9.6 Kbps to 100 Kbps	Air, radio frequencies	Data modems access by mobile	\$300 - \$1000

3rd generation mobile	2 Mbps	Air, radio frequencies	Data modems access by mobile	\$600 - \$1000
Satellite	Upstream 56 Kbps Downstream 400 Kbps (2 Mbps/16 Mbps by 2003)	Air, several frequencies (Ka-band, Ku-band)	Mobile receivers, stationary dishes	Under \$1000
Microwave links, radio links	10 Mbps - 100 Mbps	Air, e.g. 400 - 900 MHz, 15 GHz	antennas, network cards	\$3000-\$15000
Optical transmission	155 Mbps	Air, laser	Stationary receivers	\$5000 - \$20000

11-6. Summary

We had an overview of the OSI Reference Model. It is a very broad subject, because this model, as the name suggests, serves as a reference for various communication and networking applications. The OSI model provides a conceptual framework for communication between computers, but the model itself is not a method of communication. Actual communication is made possible by using communication protocols. In the context of data networking, a *protocol* is a formal set of rules and conventions that governs how computers exchange information over a network medium. A protocol implements the functions of one or more of the OSI layers.



It is appropriate to provide a summarized list of fundamental points related to the Open Systems Interconnection (**OSI**) model, namely:

Physical Layer

The physical layer is responsible for passing bits onto and receiving them from the communication channel. This layer has no understanding of the meaning of the bits, but deals with the electrical and mechanical characteristics of the signals and signalling methods.

Data Link Layer

Data link layer is responsible for both Point-to-Point Network and Broadcast Network data transmission. It hides characteristics of the physical layer (e.g. transmission hardware from the upper layers.

It is also responsible to convert transmitted bits into frames it transmits the frames into an error free transmission line by adding error control and flow control.

Network Layer

Network layer is responsible for the controls of routers and subnets operation. It also handles the formation and routing of packets from source to destination with congestion control.

Transport Layer

Transport layer is a kind of software protocol to control packets delivery, crash recovery and transmission reliability between sender and receiver. Multiplexing between transport and network connections is possible.

Session Layer

Session layer provides dialogue control and token management.

Presentation Layer

When data is transmitted between different types of computer systems, the presentation layer negotiates and manages the way data is represented and encoded. Essentially a 'null' layer in case where such transformations are unnecessary.

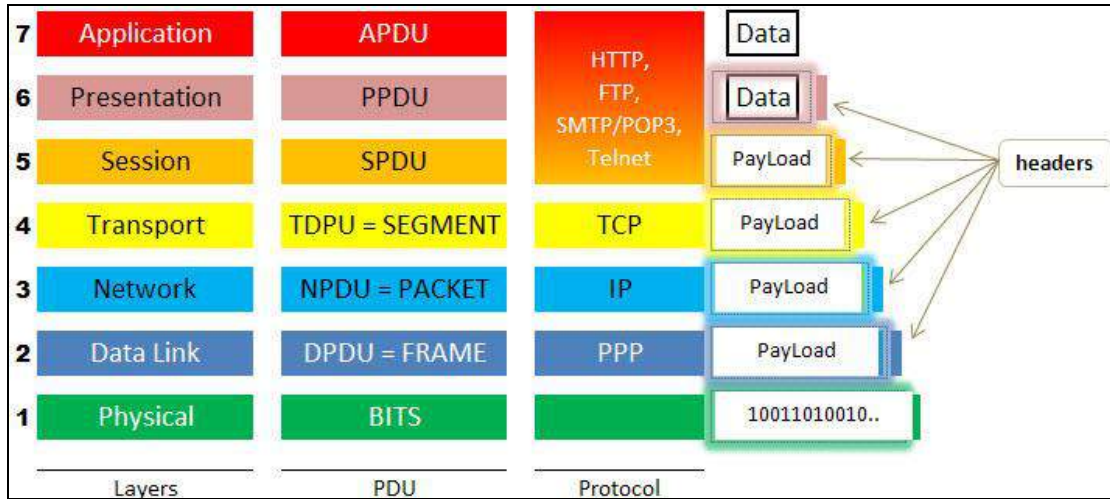
Application Layer

This top layer defines the language and syntax that programs use to communicate with other programs. For example, a program in a client workstation uses commands to request data from a program in the server. Common functions at this layer are opening, closing, reading and writing files, transferring files and e-mail messages, executing remote jobs and obtaining directory information about network resources.

Each of the OSI seven layers has its own **protocols** and always prepares (or format) data to understand the next layer. This occurs at all stages of a communication in both directions. The following figure shows the different protocols of different layers and the protocol data units (**PDU**'s) of each layer of the OSI reference model. PDUs are relevant in relation to each of the first 4 layers of the OSI model as follows:

- The Layer 1 (Physical Layer) PDU is the bit or, more generally, symbol (can also be seen as "stream")
- The Layer 2 (Data Link Layer) PDU is the frame
- The Layer 3 (Network Layer) PDU is the packet

- The Layer 4 (Transport Layer) PDU is the segment for TCP, or the datagram for UDP
- The Layer 5-6-7 (Application Layer) PDU is the message



11-7. Problems

11-1) What are the main differences between the OSI model and the SNA?

11-2) In which layer does the computer modem lie? And where does the TCP/IP lie?

11-3) Choose a communication system, such as the mobile phone or any other system, and describe it in terms of the OSI model?

11-4) What are differences between the IBM system network architecture (SNA) and the standard OSI model?

11-5) Is there any other OSI-based model, which is more suitable for describing the architecture of telecommunication systems?

Hint: Look at the short description of the **NTCIP** model at the end of Chapter 1, and compare it with the OSI model.

11-8 References

[1] M. **Castells**, Rise of the Network Society. 3 Vols. Cambridge, MA: Blackwell Publishers, **1996**.

[2] CISCO Network Handbook, CISCO Inc., **2006**.

Chapter
12

Data Networks & Their Protocols

Contents

- 12-1. Introduction**
 - 12-1.1. Network Types
 - i. Local Area Networks (**LANs**)
 - ii. Metropolitan Area Networks (**MANs**)
 - iii. Wide Area Networks (**WANs**)
 - 12-1.2. Network Traffic Control Mechanisms
 - i. CSMA/ CD
 - ii. Token Passing
 - 12-1.3. Network Components (Gateways, Routers, Switches ,...)
- 12-2. LANs & Their Physical Layer**
 - 12-2.1. LAN Topologies
 - i. Star Topology
 - ii. Bus Topology
 - iii. Ring Topology
 - iv. Tree Topology
 - 12.2.2. LAN Cables
 - 12.2.3. Wireless LAN (WLAN)
 - 12.2.4. LAN Standards
- 12-3. Ethernet Physical layer**
 - 12-3.1. Ethernet Cables
 - 12-3.2. Fast Ethernet & Gigabit Ethernet
 - 12-3.3. Ethernet Repeaters, Hubs and Switches
 - 12-3.4. Ethernet Network Adapter Cards
 - 12-3.5. Wireless Ethernet

- 12-4. WANs & Their Physical Layer**
 - 12-4.1. Public Data Networks (PDN)
 - 12-4.3. Switching Networks
 - 12-4.4. **Circuit Switched** Data Networks (CSDN)
 - 12-4.5. **Packet Switched** Data Networks (PSDN)
- 12-5. Data Link Protocols**
 - 12-5.1. Binary Synchronous Control (BSC) or BiSync
 - 12-5.2. HDLC / SDLC Protocols
 - 12-5.3. Ethernet Protocol for LANs
 - 12-5.4. Point-to-Point Protocol (PPP) for WANs
- 12-6. Other WAN Protocols**
 - 12-6.1. X.25 Standard of Packet Switching
 - 12-6.2. Internet Protocol and TCP / IP
 - 12-6.3. IP6 and IPTV
 - 12-6.4. Frame Relay Network
 - 12-6.5. ISDN
 - 12-6.6. Cell Relay Network
 - 12-6.7. Asynchronous Transfer Mode (ATM)
 - 12-6.8. VoIP, VoATM and VoFR
- 12-7. Integration of Voice and Data Networks**
- 12-8. Summary**
- 12-9. Problems**
- 12-10. Bibliography**

**Chapter
12**

Data Networks & Their Protocols

12-1. Introduction

Data networks are key infrastructures of the information society with high socio-economic value. Such networks contribute to the correct operations of many critical services, from healthcare to finance, scientific research, transportation, video broadcasting and entertainment.

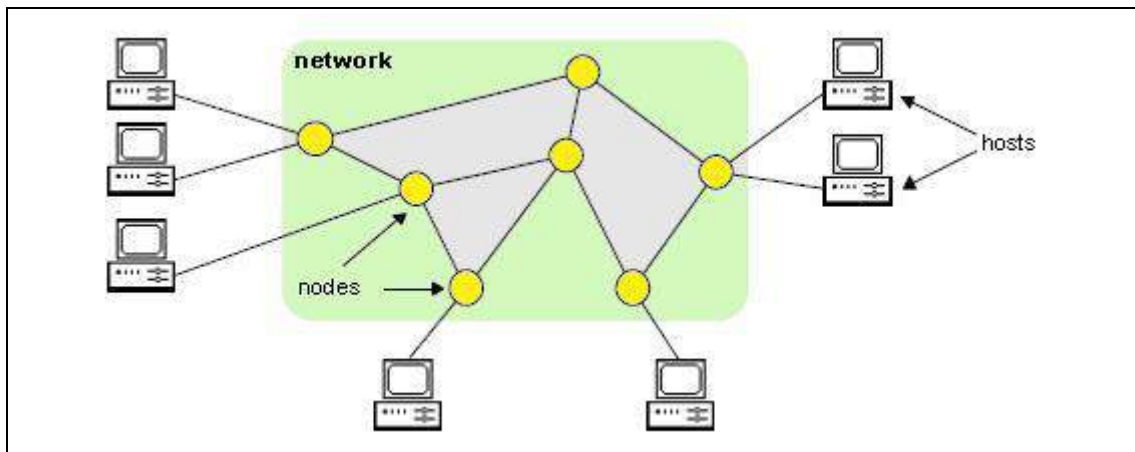


Figure 12-1. Illustration of a data networks

The common language between networking communication devices is known as **protocols**. The communication networks are usually described by a stack of protocols that defines their implementation. The International Standards Organization (**ISO**) tackled this problem by developing the Open Systems Interconnect (**OSI**) model

12-1.1. Network Types

There are three basic types of data networks, according to their graphical extension, namely: the local-area network (**LAN's**), metropolitan area networks (**MAN's**), and wide-area networks (**WAN's**). In addition, one can add other emerging technologies, such as Personal area network (**PAN**), Bluetooth, Wireless LAN (**WLAN**), Wi-Fi and WiMAX. In this chapter, we discuss the different types of data networks, with emphasis on LAN's, Ethernet and WAN's as well as their components and protocols of communication.

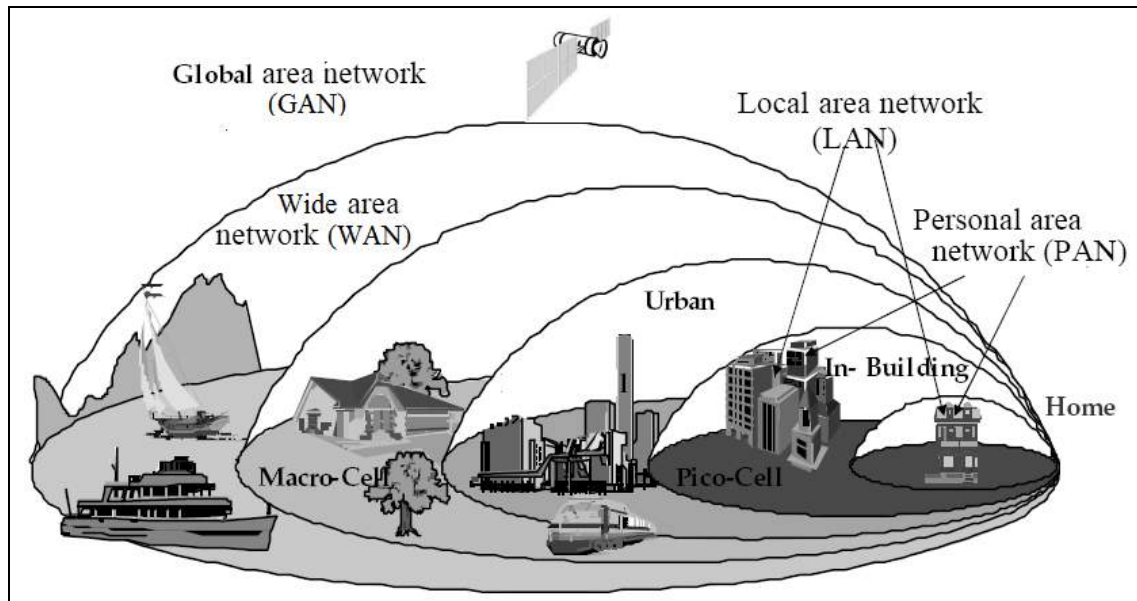


Figure 12-2. Different networks and their relative geographic extensions

i. Local Area Networks (LAN)

Local area networks (**LANs**) are private networks within a single building or campus of up to one kilometer in size. They are widely used to connect personal computers and workstations in company offices and factories to share resources and exchange information. LANs are distinguished from other kinds of networks by three characteristics: (1) size, (2) transmission technology, and (3) topology. LANs are restricted in size, which means that the worst-case transmission time is bounded and known in advance. Knowing this bound makes it possible to use certain kinds of designs that would not otherwise be possible. It also simplifies network management. LANs may use a transmission technology consisting of a cable to which all the machines are attached, like the telephone company party lines once used in rural areas. Traditional LANs run at speeds of 10 Mb/s to 100 Mb/s, have low delay, and make few errors. Newer LANs operate at up to 10 Gb/s. There are various possible topologies for broadcast LANs. Figure 12-3 shows three of them, namely: **star** (a), **bus** (b) and **ring** (c) topologies. In a ring network, each bit propagates around on its own, not waiting for the rest of the packet to which it belong. In a bus (linear cable) network, at any instant only one machine is the master and is allowed to transmit. All other computers are then required to refrain from sending. **Therefore, an arbitration mechanism is needed in such networks to resolve conflicts when two or more computer want to transmit simultaneously.** The IEEE 802.3 standard, which is popularly called **Ethernet**, is a bus broadcast network with decentralized control, usually operating at 10Mb/s to 10Gb/s.

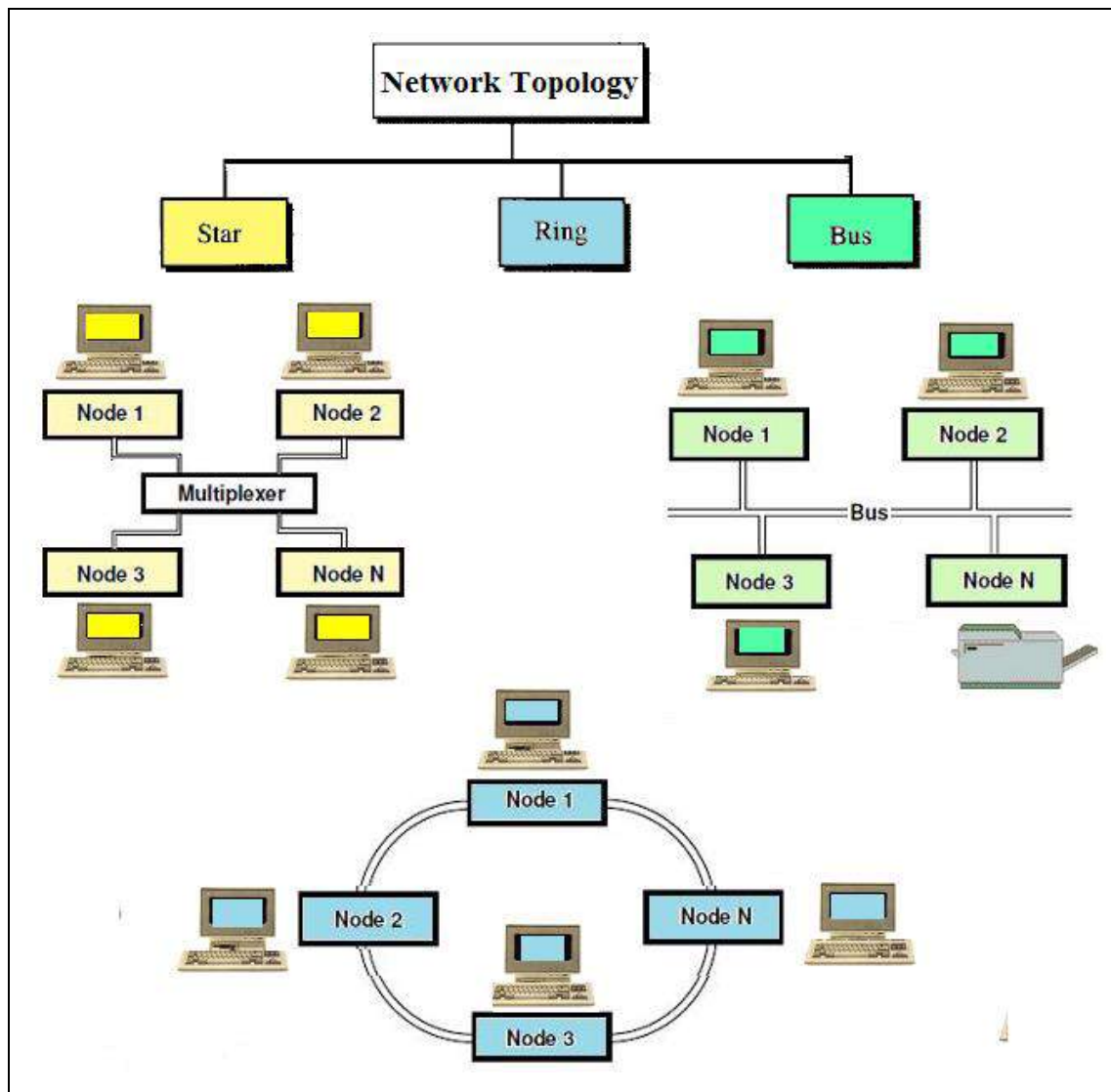


Figure 12-3. Interconnection techniques for creating a local area data networks

ii. Metropolitan Area Networks (MAN)

A **metropolitan area network**, or **MAN**, is basically a bigger version of a LAN and normally uses similar technology. It might cover a group of nearby corporate offices or a city and might be other private. A MAN can support both data and voice, and might even be related to the local cable TV network. A MAN just has one or two cables and dose not contain switching element, which shunt packets over one of several potential output lines. Not having to switch simplifies the design.

iii. Wide Area Networks (WAN)

A wide area network (**WAN**), spans a large geographical area, often a country or continent. It contains a collection of machines intended for

running user programs. We call these machines hosts. The hosts are connected by a communication subnet. The job of the subnet is to carry messages from host to host, just as the telephone system carries words from speaker to listener. Separation of the pure communication aspects of the network (the subnet) from the application aspects (the hosts), greatly simplifies the complete network design. In most wide area networks, the subnet consists of two distinct components: transmission lines and switching elements. Transmission lines can be made of **copper wire**, **optical fiber**, or **radio links**. Switching elements, such as **bridges**, **switches** and **routers**, are special devices that connect three or more transmission lines. When data arrive on an incoming line, the switching element chooses an outgoing line on which to forward them.

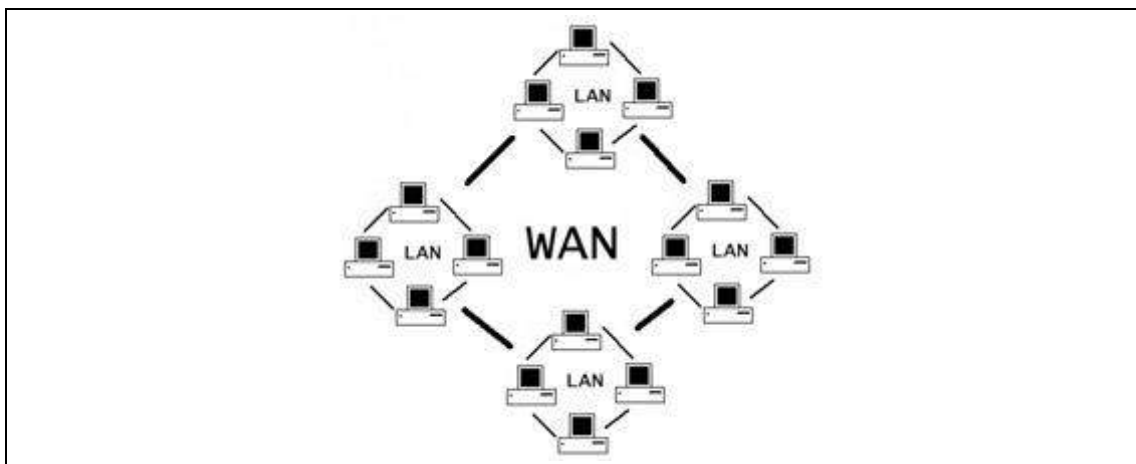


Figure 12-4. WAN as a network of LANs.

12-1.2. Network Traffic Control Mechanisms

There are two generic techniques of traffic control and contention resolution that are widely spread in data networks, namely:

- 1- **CSMA/CD** and
- 2- **Token passing.**

i. CSMA/CD

CSMA/CD is an abbreviation for Carrier Sense, Multiple Access with Collision Detection. The CSMA/CD scheme sounds complex but is straightforward to implement in practice. It is used within a number of different bus networks. Each device in a CSMA/CD system is allowed to attempt to transmit on the network bus at any time. In other words, multiple access. However, prior to attempting a transmission, each device

must monitor the bus for the presence of a carrier signal, emanating from another node. This is called *carrier sensing*. If a carrier is already present on the bus (another node is already transmitting), then the device must wait until that transmission has ceased before attempting to place a message packet (frame) on the bus. Even when a device has the right to transmit, it must still monitor the bus to ensure that the signal that is being sent is the same as that on the bus.

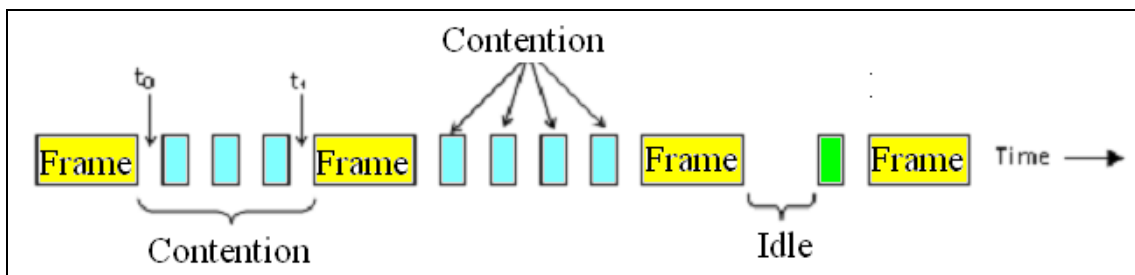


Figure 12-5. CSMA/CD can be in one of 3 states: contention, transmission, or idle

It should be apparent, that two devices may simultaneously detect that the bus is clear and simultaneously attempt to transmit a message frame. In this case a collision will occur. Since all devices are monitoring the state of the bus, both of the devices will detect the collision. The first device to detect a collision must transmit a random bit pattern, referred to as a *jam sequence* for a short period of time. The jam sequence is designed to last long enough for the other colliding devices to realize that a collision has occurred. Since data is corrupted by a collision, both devices back-off for a certain time interval and attempt to complete re-transmission of message frames later. The CSMA/CD system is probabilistic in nature. In other words, there is no way of knowing how long it will take for a message to get from source to destination. A CSMA/CD network is therefore said to be non-deterministic. The sCSMA/CD system is not ideally suited to the industrial environment. The irony of CSMA/CD is that the time delay for messages is longest when the network is busiest and the network is generally busiest when abnormal or emergency conditions arise. Since we demand the fastest response time under emergency and abnormal conditions the network is often classified as unacceptable in the factory. CSMA/CD is however an excellent system for the office environment, where data transfer between nodes is sparse, and occasional time delays are of little consequence.

ii. Token Passing

In principle, the so-called **token passing** scheme sounds much simpler than the CSMA/CD system. In practice it is more difficult to implement.

The scheme is based upon a binary bit pattern that is referred to as a "token". Before any node is permitted to place message frames onto a network, it must be in possession of the token. Once a node has the token, it is permitted to transmit a message frame and must then pass the token on to another node. The movement of the token from node to node forms a logical **ring** between devices. The token is itself a message frame (packet) with a special control section that defines its characteristics. A node wishing to use the token modifies these characteristics so that it can become a message frame. The node can then place data into the message frame. The token passing scheme is deterministic, because it is possible to precisely define the maximum delay that will arise in transmitting a data frame. It is for this reason that the scheme is often promoted as a basis for industrial networks. There are a number of problems with the token passing scheme. Since it is possible for a device to fail while it is in possession of the token, steps must be taken to ensure that there is a mechanism for regenerating a lost token. This makes the system much more complex than the CSMA/CD system. The token passing scheme also introduces delays into the network even under light traffic conditions. In other words, a token is still passed from device to device, whether or not that device is to broadcast on the network. In the CSMA/CD scheme, a transmitting device can export information immediately if the bus is clear - with token passing the transmitter must always wait for the token. Having introduced two of the more prolific contention resolution schemes, and the expressions deterministic and non-deterministic, it is important to qualify their definitions. A network is the sum total of the communication medium, the contention resolution scheme and all the other software functions that go together to form application programs.

12-1.3. Network Components (Gateways, Routers, Bridges and Switches)

It is a difficult enough task to make a number of devices meaningfully communicate with one another on a single network. It is even more difficult to make devices which are attached to totally different networks communicate with one another. Schematically, the dilemma is shown in figure 12-6. As shown, if we assume that networks 1 and 2 of are totally different in terms of framing, contention, modulation, etc., then we can see the complexity of the problems to be solved by the inter-network interface device. A typical example might be where Device B (network 1) wishes to communicate with Device Y (network 2). The inter-network device must simultaneously satisfy the physical requirements and access requirements of both networks, whilst performing a translation of data frames from one form to another. If the two networks are OSI based, then

at least there is some scope for breaking this problem down into layers of compatibility and functionality.

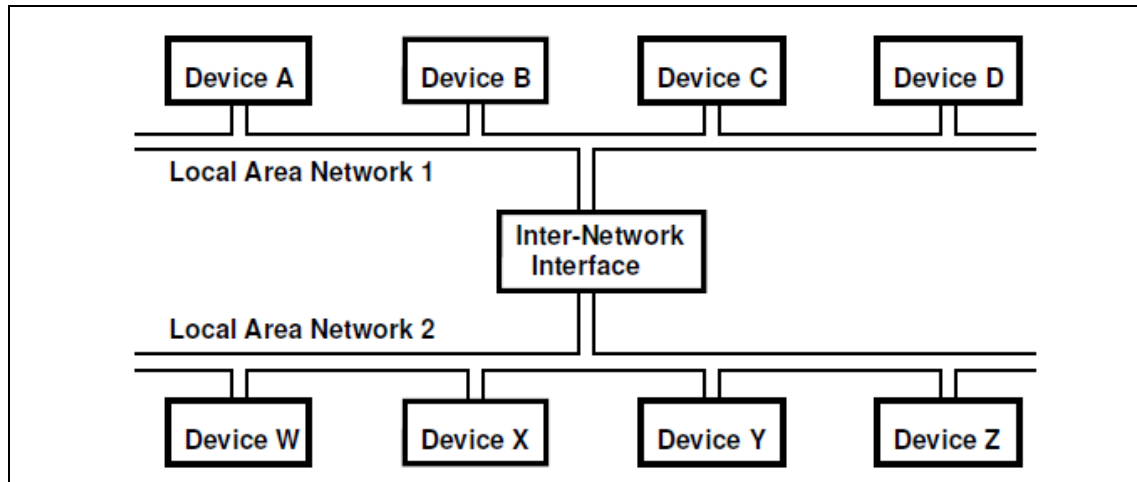


Figure 12-6. Interconnecting different networks

There are three, commonly used devices which perform the role of the **inter-network** interface between OSI systems. These are referred to as:

- Gateways
- Routers
- Bridges.

In physical terms all these devices should be considered as computers with varying degrees of sophistication. The most complex of these systems is the **Gateway**, which is designed to translate all seven layers from the protocol of one network to another, as shown in figure 12-6.

The upper layers of an OSI network are relatively complex and therefore the amount of time required for a Gateway computer to perform a protocol translation may be significant. In addition to the inter-network time delays involved with Gateways, their cost is substantial and clearly they are a last alternative solution for communication between totally dissimilar networks. If the upper layers (4, 5, 6 and 7) of two, OSI networks are the same, then it is possible to use a Router to perform protocol translation for the lower 3 layers. Routers can be used to connect a number of such networks together at a common point as shown in figure 12-7. Packets are routed from one network to another based upon the destination address specified within the network layer (3) of the packet.

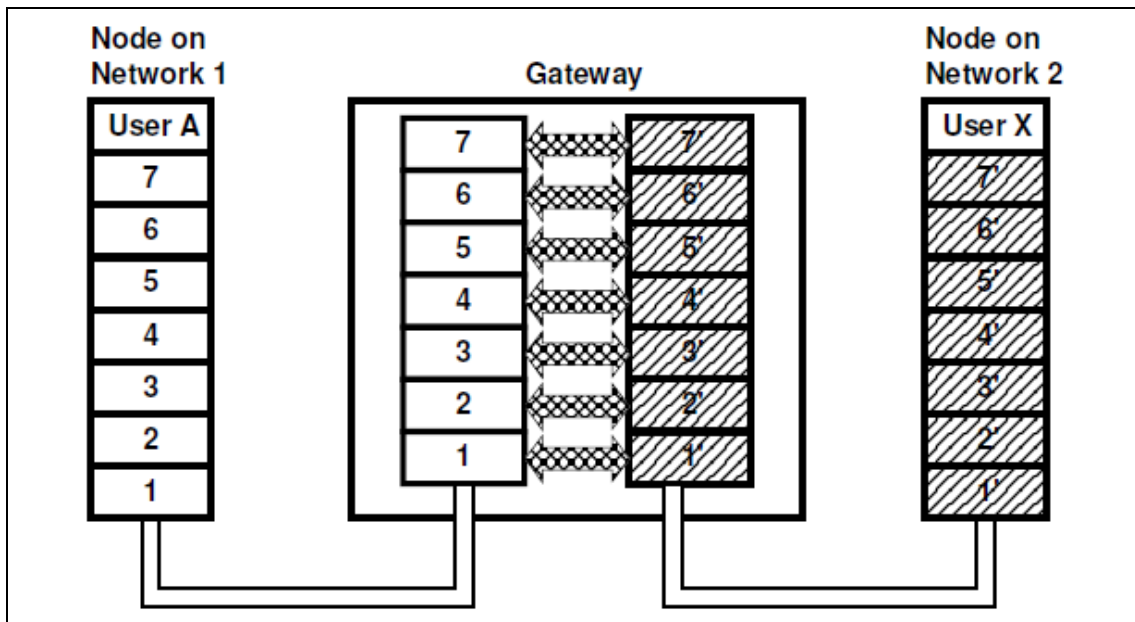


Figure 12-7. Gateway between two completely different OSI networks

Routers are used to interconnect such networks, which are only different in the lower 3 layers. The **OSI** networks which are easiest to interconnect are those which are completely identical or those which differ only in the lower one or two layers.

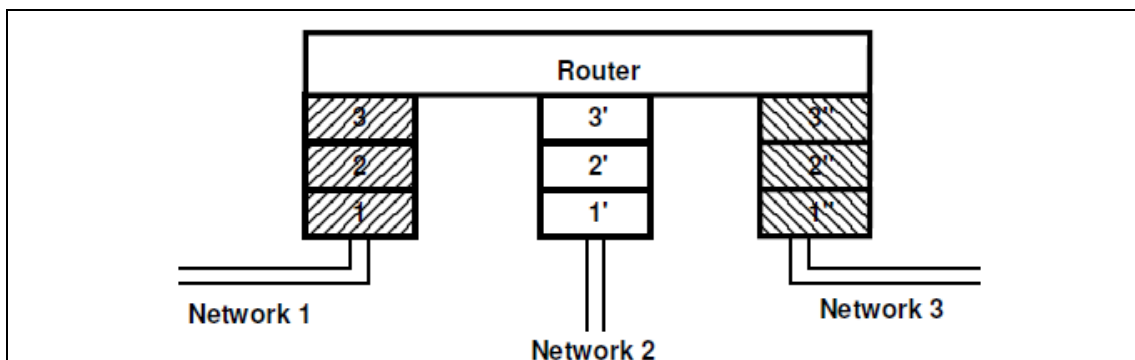


Fig. 12-8. Router between OSI network, which are different in the lowers 3 layers.

The **Bridge** are used to interconnect the similar systems, which are only different in lower layers, as shown schematically in Figure 12-9. In addition to Gateways, Routers, and Bridges, we may need to Switches, and sometimes Hubs, in order to connect all computers in one network. We now appreciate that whilst in theory it may be desirable to have all devices connected to a common network, in reality this is unlikely to occur. We realize that environmental and performance features are the major reasons for using different networks. The physical and data link layers of the OSI model are the ones which are most closely related to

environmental factors. Figure 12-10 depicts the utilization of Gateways, bridges and routers in a heterogeneous network.

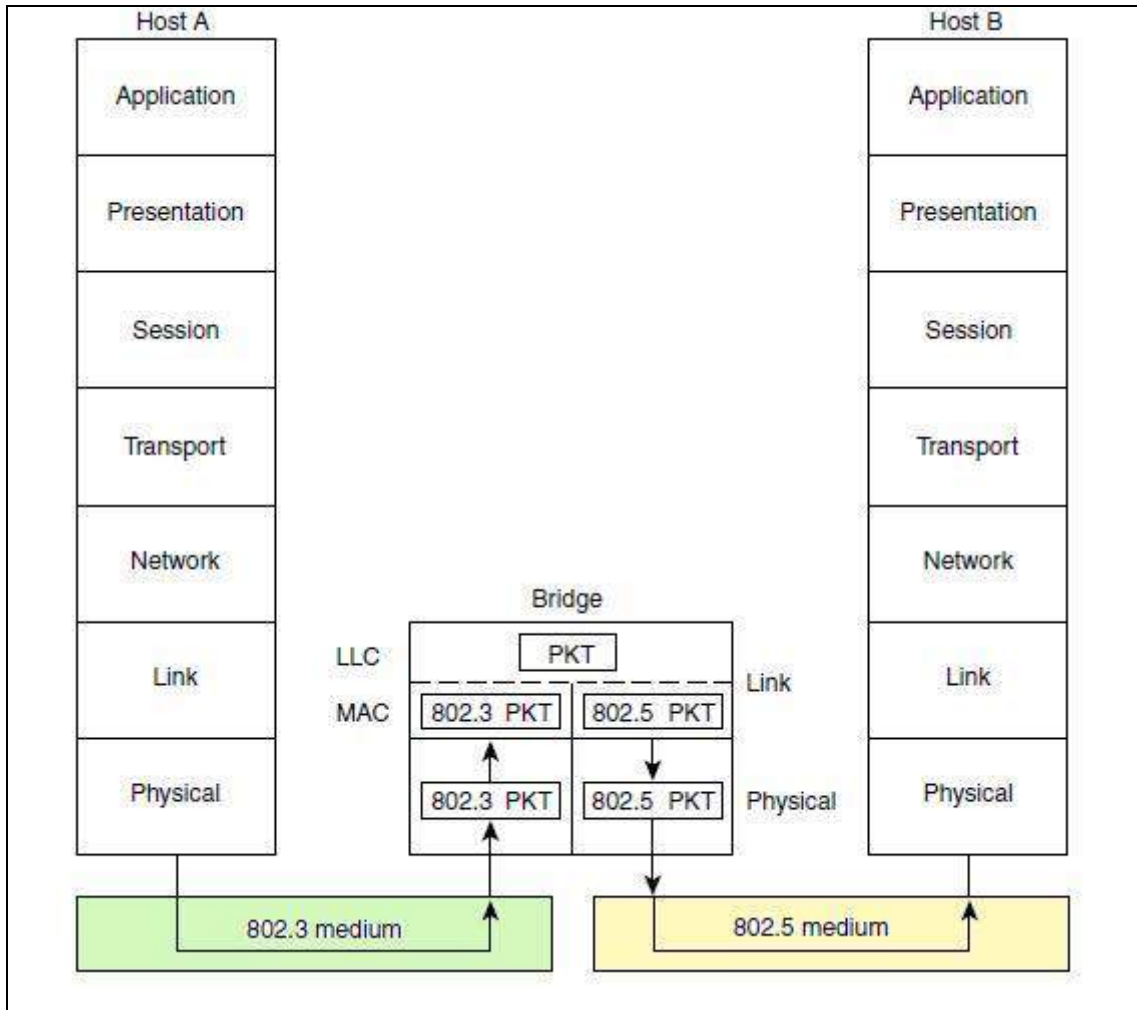


Figure 12-9. Bridging of two similar OSI networks, which are only different in the lower two layers.

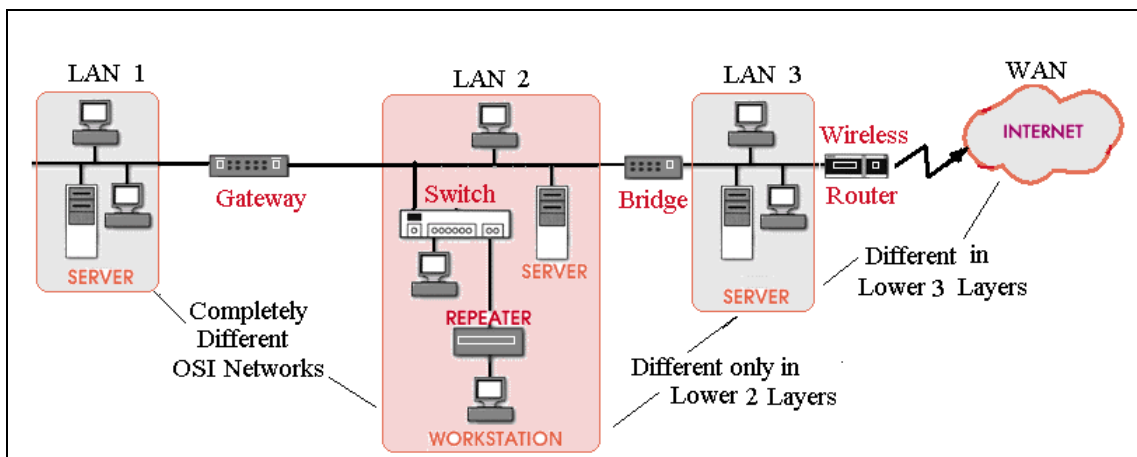


Figure 12-10. Networking of different OSI networks with different layer structures.

12-2. LAN's & Their Physical Layer

A Local Area Network (LAN) is so named because its nodes are located in proximity (less than a kilometer) of one another. A point to point link between two nodes can therefore be considered as a network with two nodes and it shares many of the characteristics of larger networks. We already know that even a simple, point to point serial communication link between two nodes needs to have many rules of protocol resolved before it can function correctly. We need to use the common signaling techniques, common character representations, complementary communications hardware and so on. We need the communications interchanges between nodes to be strictly governed by these rules of protocol so that conflicts can be resolved. All these issues are also true of networks - except that now we need to coordinate the communications between many nodes simultaneously. The majority of networks use serial communication between nodes.

12-2.1. LAN Topologies

We have seen so far in the last section that local area networks may have several ways for connecting several nodes. Figure 12-11 summarizes the basic network topologies together with their common names. The three basic topologies, are the **star** topology, the **ring** topology the **bus** topology and the tree topology. These physical arrangements are more commonly referred to as network **topologies**. In the following subsections, we describe these topologies with ample details.

i. Star Topology

The **star network** is composed of a number of point to point links emanating from the star node, as shown in figure 12-11(a). A star network has an intelligent central node, referred to as the **star node**, or multiplexer or concentrator. In a simple star network, the star node may be a microprocessor-controlled private automatic branch exchange (**PABX**) or a **switch**. At the other extreme however, a large star network could have a **workstation** as the star node, with many terminals communicating to one another through it. Once the star node (switch) has made a physical connection, then two nodes can communicate with one another as though the node was not present. The star node therefore needs to have some intelligence, particularly if a **conflict (contention)** situation arises, where several nodes request connection to one target node at the same time. Then, the star node makes the decision of connecting any pair of nodes. It therefore needs to be able to resolve any contentions that may arise. The star node is also responsible for tasks such as queuing requests for link establishment between nodes.

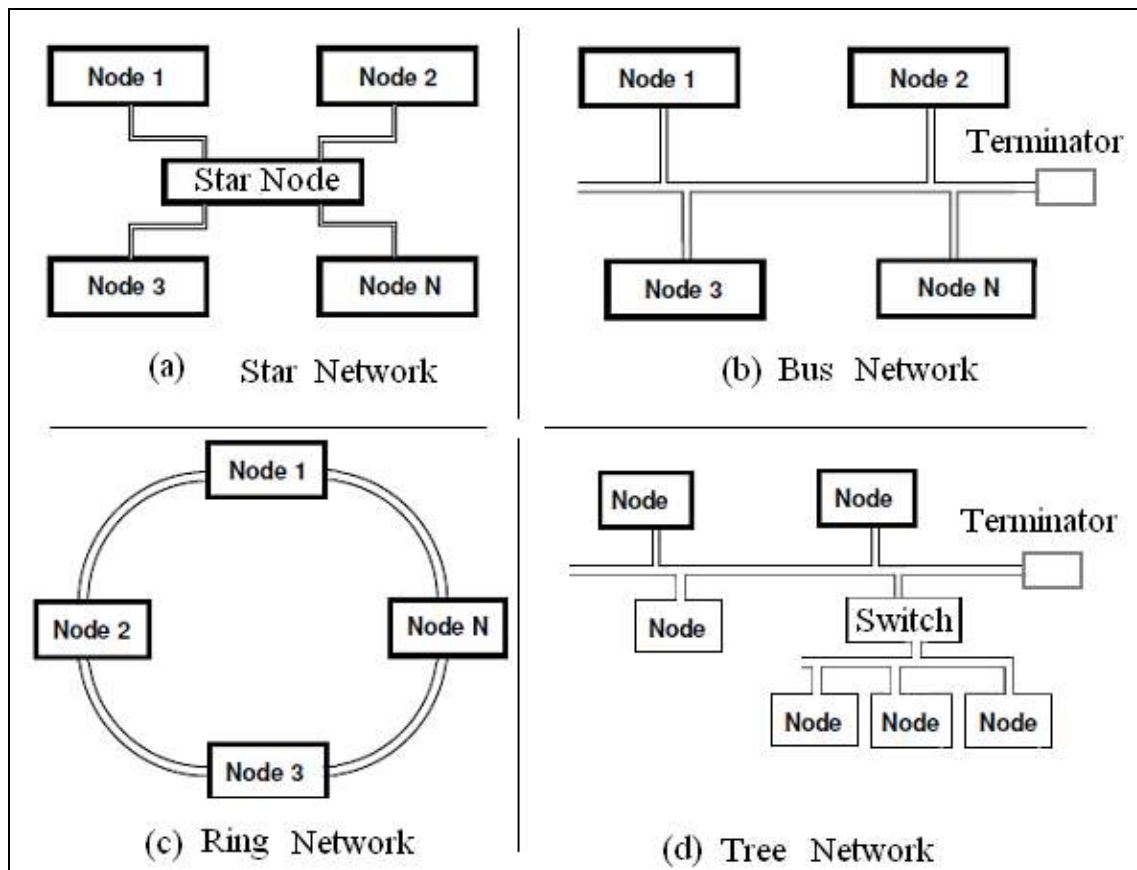


Figure 12-11. Common network topologies

Several advantages stem from this structure. Firstly, the star node is transparent to communicating nodes once a connection has been made. Provided that the connected nodes agree on a software protocol, the nature of that protocol is of no consequence to the star node and the remainder of the network. In other words, devices 1 and 2 can talk to each other through "protocol A" and devices 3 and 4 can talk to each other through "protocol B". It is then also possible that device 3 can talk to device 1 through "protocol A". This has merits in several situations where it is not practical to have all nodes using a single protocol and yet it may still be necessary to have all nodes capable of talking to one another. A good example of this would be where devices 1, 2 and 3 are computers and device 4 is a robot or programmable logic controller (PLC). Another advantage of the star network is that the physical medium used between any node and the star node can be varied to suit the environment. For example, if device 1 is in the factory then it can be linked to the star node through an optic fiber cable. If device 2 is in the office, close to the star node, then it can be linked via a twisted-pair cable and so on.

There are also a number of disadvantages to the star network topology. All communication is dependent upon the star node - if it fails then all communications ceases. Another shortcoming of the star network system is the cost of cabling. It is more cost effective to lay a single trunk cable through a networking area, and to use short tap cables from each node to the trunk, rather than to have long cables all meandering their way towards a central node. This is best visualized in terms of the power supply within a home. Another significant factor that arises in the cost equation for star networks is that of cable maintenance. In large star networks, many cables need to converge on the central node. Hence the concentration of cables, within ducts near the star node, is always very high. This makes trouble-shooting more difficult and time-consuming than it would otherwise be with other network configurations. In a large area environment, the problem of cable maintenance in star networks is severe. Often, defective old links are simply replaced with new links, and the old links left in the ducts, because the cost of removing them is so high. The dead links then further add to the cable concentration near the star node. The problems of high cable concentrations are automatically eliminated in bus networks, where a central trunk is laid down throughout a networking zone. Each node is connected into the network through a short tapping cable in a manner that is completely analogous to the domestic power supply scenario cited earlier.

ii. Bus Topology

The **bus network** is perhaps the most common form of the networking topologies - particularly in the industrial environment. As shown in figure 12-12, the bus network is similar to the internal bus structure used for communications within a computer. The major differences are that in bus networks, data transfer is serial and secondly that there is no simple master-slave relationship between devices and therefore many contention situations (conflicts) may arise. Bus networks offer flexibility in terms of cable utilization, which is not the case with other topologies. The fact that a bus network is based upon a trunk cable, which is laid throughout an entire area, means that video and voice channels can share one cable, through the use of modulation techniques. This greatly increases the cost of the bus network.

iii. Ring Topology

In a **ring network**, neighboring nodes are interconnected with point to point serial links to form a complete ring, as shown in Fig. 12-11(c). There is no intelligent coordinating node and hence all nodes must have the intelligence to cope with contentions and recognize appropriately data

Each device receives a message and then retransmits it. Nodes in between the source and the destination do not alter the message. However, when the destination node receives the message, it modifies the control portion of the message packet and places it back onto the loop. The originator of the message packet determines whether or not the message has reached its target correctly by the modifications on the returning packet. Ring networks are relatively commonplace in the office environment, where the area they cover is relatively small.

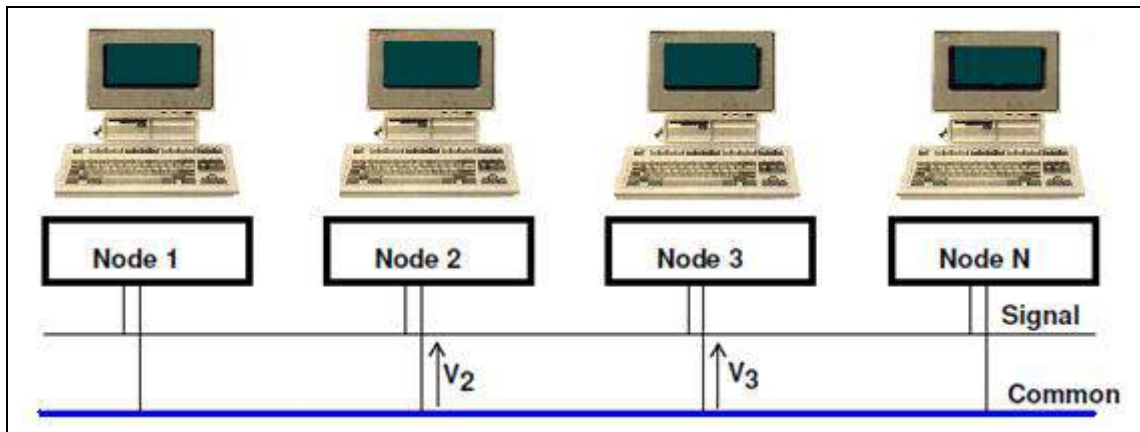


Figure 12-12. Network with devices connected via a conducting bus

iv. Tree Topology

A tree topology combines characteristics of linear bus and star topologies. As shown in figure 12-5(d), the tree network consists of groups of star-configured workstations connected to a linear bus backbone cable. Tree topologies allow for the expansion of an existing network, and enable schools to configure a network to meet their needs. This topology is supported by several hardware vendors. In bus topology, all the nodes are physically connected together on a central cable trunk (bus). Whenever one node places data on the trunk cable, all the other nodes receive that data. Since there is no intelligent switching device in this network, it is necessary to establish a system **protocol** where each node can recognize relevant data and ignore irrelevant data. Although it is possible to have multiple channels existing on a cable (by modulation), a contention still arises if two devices attempt to transmit at the same time on the same channel. Each device on such a network must have the intelligence to independently react to and resolve such contention situations. In a point to point serial link there is only one possible destination for all transmitted data - that is, the node at the other end of the serial link. However in a network, each transmitting node must theoretically be capable of sending information to any other node on that network. All the

network forms therefore need to cope with the fundamental issue of addressing. All nodes need to be given a unique address in order for data to be targeted correctly. The problem of network addressing is not unlike the problem of addressing devices (sharing the same data bus) within a computer system. The need for addressing means that regardless of the physical network arrangement, data must be placed into suitable packets for transfer. Each packet of data moving through a network needs to contain some source and target addressing information. This enables a receiving device know which device is transmitting to it and where to send response messages. The concept of packet addressing is shown schematically in figure 12-13.

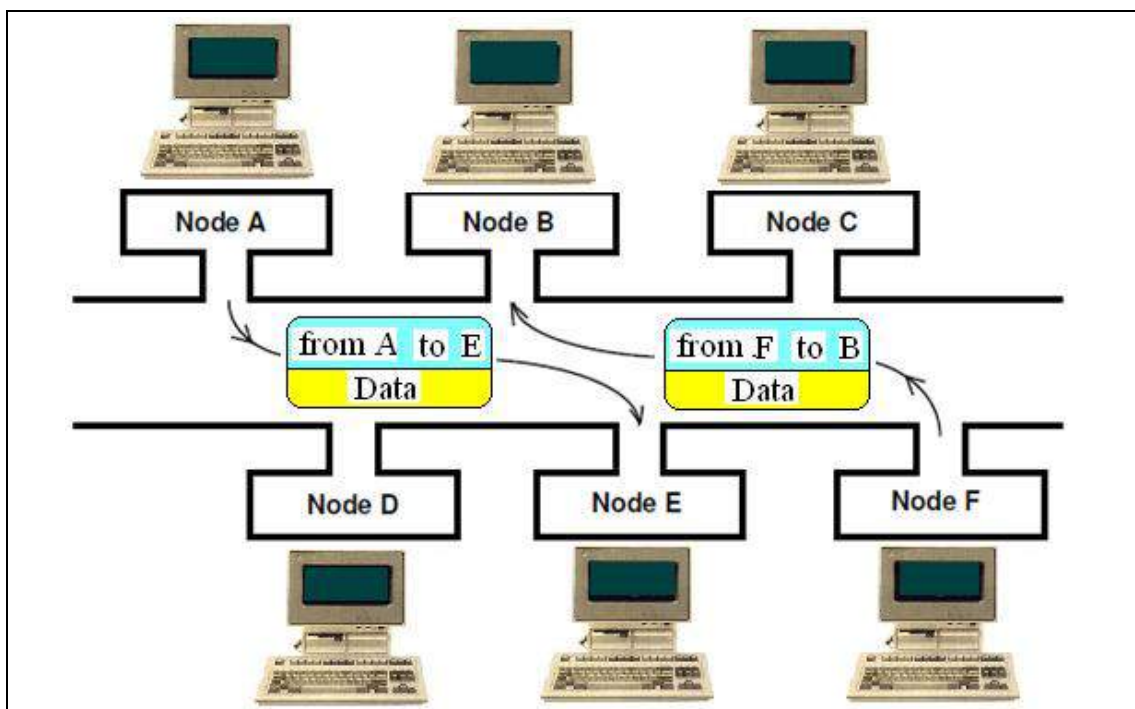


Figure 12-13. Addressing packets of data on a network

With the exception of traffic control (**contention**) and addressing functions, the issues related to networks are essentially the same as those in point to point links. We still need error checking mechanisms in the form of block **Check Sums** or more commonly, Cyclic Redundancy Checks (**CRC**). We still need hardware to perform parallel to serial (and vice-versa) conversion on each node. We still need layers of data handling software that can provide our applications programs with powerful communications sub-programs. The major difference between network media and point to point media is in the connectors. If we choose to have a bus type network, then we need to have special connectors that can tap into the transmission medium at selected points. On the other

hand, if we have a switched ring network, then we can often use the same connectors as in point to point communication - for example RS-232.

12-2.2. LAN Cables

One of the most common LAN media, in the bus network arrangement, is the co-axial cable. One of the major reasons for this is because of the proliferation of low cost connectors and adaptors that resulted from the introduction of the American cable television system. Coaxial cables lend themselves to tapping at any point so that many devices can be connected to a central bus. When coaxial cables are used in LANs, only one conductor is available for signal transmission (plus common return line). Therefore if we wish to achieve full-duplex communications in LANs, we need to separate transmitted and received data channels. Twisted-pair (**TP**) cables are also extensively used in networking, but are more prone to electromagnetic interference (EMI) than coaxial cables.

Twisted pair cabling comes in two varieties: shielded and unshielded. Unshielded twisted pair (**UTP**) is the most popular and is generally the best option for simple networks. The quality of UTP may vary from telephone-grade wire to high-speed types. The cable has four pairs of wires inside a jacket. Each pair is twisted with a different number of twists to help eliminate interference from adjacent pairs and other electrical devices. The Electronic Industry Association and Telecommunication Industry Association (EIA/TIA) have established standards of UTP and rated 6 categories of wire, as shown in the following table.

Table 12-1. Standard categories of UTP cables.

Type	Max Frequency	Max Rate	Use, Network
Cat1	10 MHz	Voice Only	Telephone Wire
Cat2	10 MHz	4 Mb/s	LocalTalk
Cat3	10 MHz	10 Mb/s	Ethernet
Cat4	10 MHz	20 Mb/s	16 Mb/s Token Ring
Cat5	100 MHz	to 1Gb/s	Fast Ethernet, 10/100Base-T
Cat5e	100 MHz	to 1Gb/s	Fast Ethernet, 10/1000Base-T
Cat6	250 MHz	1Gb/s	Gigabit Ethernet, 10/1000Base-T

Fiber optic cabling consists of a center glass core surrounded by several protective layers. It transmits light rather than electrical signals,

eliminating the problem of interference. Optical fiber cables are currently not as prolific in LANs. The major reasons for this relate to the relative difficulty of tapping into optic fibers, and in terminating them without the use of specialized equipment, called *splizers*. However, as optic fiber technology develops and the cost of connectors and adaptors decreases, it is evident that the system will become the dominant networking medium because of its superior noise immunity and higher bandwidth.

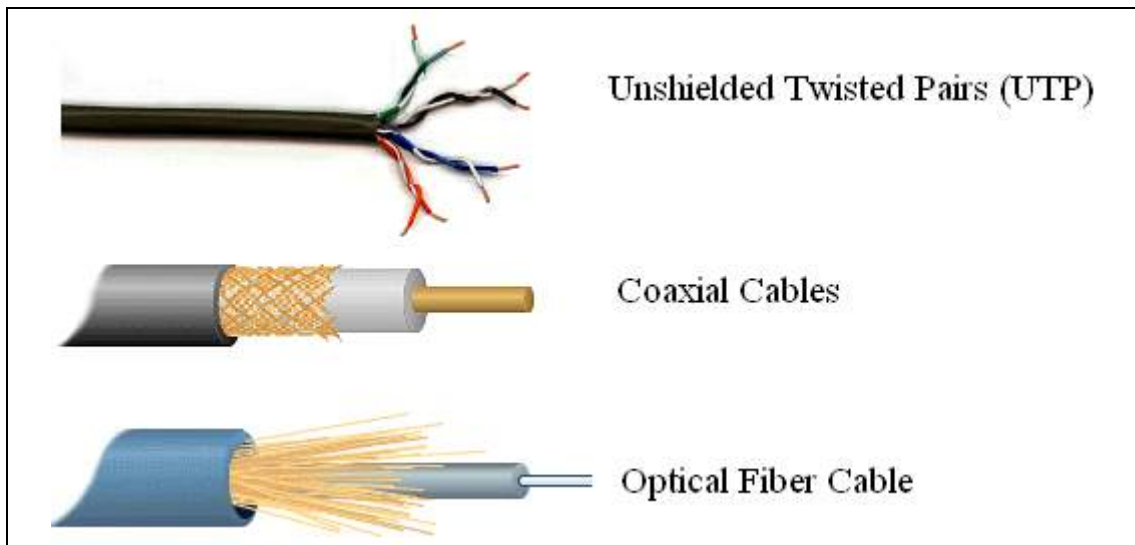


Figure 12-14. Different types of LAN cables

Cable types and connectors are not really the major issue in networking as far as end-users are concerned. In general, we have to accept a transmission medium from a limited range of commercial solutions that conform to an overall protocol. There are however, many other issues to be resolved when we attempt to physically interconnect a number of intelligent devices through a network. We shall look at these as we progress through the chapter.

12-2.3. Wireless LANs

Not all networks are connected with cabling; some networks are wireless. Wireless LANs use RF signals or infrared light beams to transmit and receive data over the air. WLANs give users mobility as they allow connection to a local area network without having to be physically connected by a cable. This freedom means users can access shared resources without looking for a place to plug in cables, provided that their terminals are mobile and within the designated network coverage area. With mobility, WLANs give flexibility and increased productivity, appealing to both entrepreneurs and to home users.

WLANs may also enable network administrators to connect devices that may be physically difficult to reach with a cable. For longer distance, wireless communications can also take place through cellular telephone technology or by satellite. Wireless networks are great for allowing laptop computers or remote computers to connect to the LAN. Wireless networks are also beneficial in older buildings where it may be difficult or impossible to install cables. Wireless LANs also have some disadvantages. They are relatively expensive, with poor security and are susceptible to electrical interference from other radio signals. They are also slower than LANs using cabling.

12-2.4. LAN Standards

The Institute for Electrical and Electronic Engineers (IEEE) developed the **802.1** and **802.2** standards specification for LAN technology. The following figure depicts the IEEE 802.1 and 802.2 standards, and how they are related to the OSI reference model. The IEEE 802.3 is dedicated for the second version of Ethernet LAN's. The IEEE also developed the **802.11** specification for wireless LAN technology. The 802.11 specifies over-the-air interface between a wireless client and a base station, or between two wireless clients.

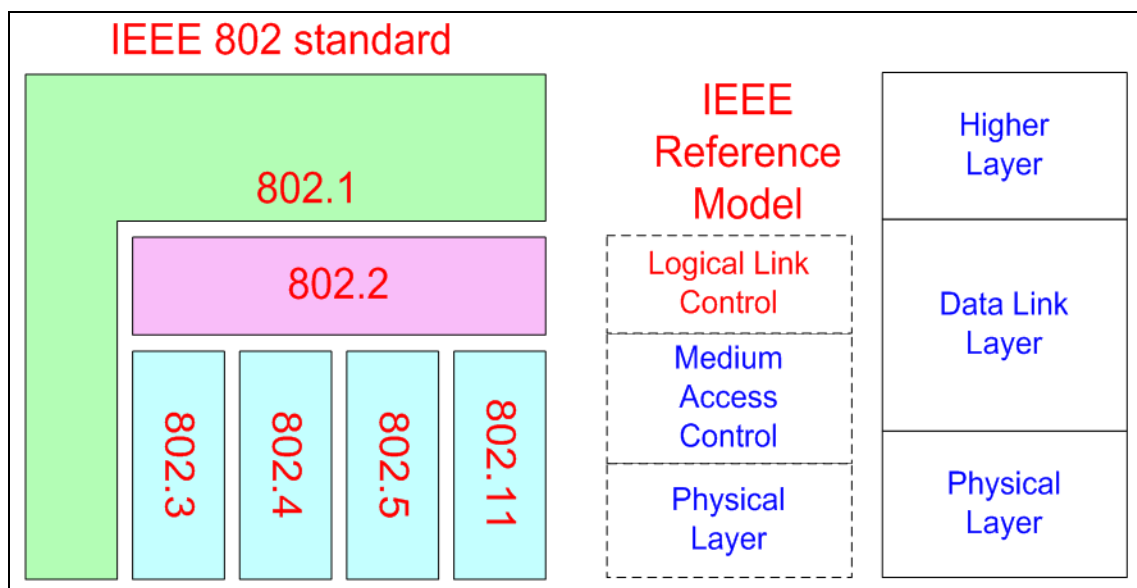


Figure 12-15. IEEE 802 and 801 LAN standards

WLAN 802.11 standards also have security protocols that were developed to provide the same level of security as that of a wired LAN. The first security protocol is called Wired Equivalent Privacy (**WEP**). WEP provides security by encrypting data sent over radio waves from end point to end point. The second WLAN security protocol is Wi-Fi

Protected Access (**WPA**). WPA was developed as an upgrade to WEP. It works with existing WEP-enabled products but provides two key improvements for data encryption and user authentication with the extensible authentication protocol (**EAP**).

Table 12-2. WiFi Wireless standards and their characteristics

SPECIFICATION	DATA RATE	MODULATION SCHEME	FREQUENCY BAND
802.11	1 or 2 Mbps	FHSS, DSSS	2.4 GHz
802.11a	54 Mbps	OFDM	5 GHz
802.11b / Wi-Fi	11 Mbps	DSSS	2.4 GHz
802.11g / Wi-Fi	54 Mbps	OFDM, DSSS	2.4 GHz

12-3. Ethernet & its Physical Layer

Ethernet is a family of networking technologies that are defined in the IEEE 802.2 and 802.3 standards. **Ethernet** is the most popular networking topology standard for computer networking. Ethernet was invented in the 1970's at the Xerox Research Center and formalized as a universal standard (IEEE 802.3) in 1985. Ethernet is very frequently and sometimes incorrectly referred to as a Local Area Network. The Ethernet system defines only the lowest two layers (data link and physical layers) of the OSI model and hence it does not represent a network in itself. In other words, one can't simply buy an "Ethernet Network" and expect to have applications support routines available. The Ethernet specification is however used as the foundation (backbone) for a range of commercial networks that provide the additional, upper five layers of OSI model functionality needed to support communications. Ethernet was designed to provide a bus network of 2500 meters maximum length, established from cable segments of 500m maximum length. The cable segments themselves are joined together with repeaters. Classical Ethernet allows for data transmission rates of 10 Mb/s, with as many as 1024 network nodes. The system is shown schematically in Figure 12-16. As shown in figure, traditional Ethernet employs a **bus** topology, meaning that all devices or **hosts** on the network use the same communication line and each device has an Ethernet address.

The Ethernet makes use of the **CSMA/CD** (Carrier Sense Multiple Access / Collision Detection) protocol for broadcasting, listening, and detecting data collisions. Some new forms of Ethernet do not use CSMA/CD. Instead, they use the so-called **full duplex** Ethernet protocol, which supports point-to-point 2-way transmission. However, the

Ethernet specifications for the physical and data link layers of the OSI model are based on the CSMA/CD bus. Although the Ethernet system was originally designed for baseband transmission over coaxial cable, the CSMA/CD contention scheme will function over any multi-access broadcasting medium. **Radio links** (air), **twisted-pair** and **optical fiber** systems have all been successfully used in Ethernet systems.

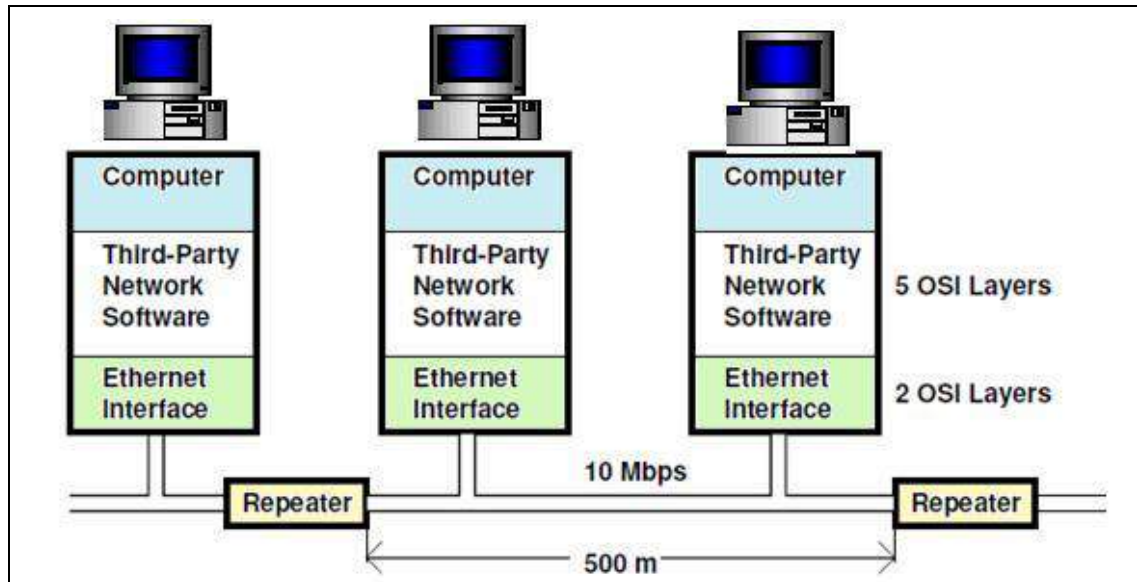


Figure 12-16 - Ethernet as the basis for networking

12-3.1. Ethernet Cables & Connectors

There have been many kinds of Ethernet, but the classic is 10 Mb/s running over copper twisted pair wires. Older Ethernet standards ran on coaxial cable and were referred to as 12-Base2 thin Ethernet and 12-Base5 thick Ethernet. The subsequent generation of Ethernet cables had 8 wires, of which 4 are used for data. The other wires are twisted around the data lines for electrical isolation from electrical interference. The cables end with **RJ-45** connectors that resemble large telephone line connectors, as shown in figure 12-17. Only two pairs of wires in the eight-pin RJ-45 connector are used to carry Ethernet signals. Both **10BASE-T** and **100BASE-T** use the same pins, a cable made for one will also work with the other. The newer Gigabit Ethernet or 1000 Base-T can run over copper wires. In modern Ethernet networks, we can distinguish two kinds of wiring schemes which are available for Ethernet cables.

- 1- **Patch (Straight-Through) Cables** and
- 2- **Crossover Cables.**

When you connect only two computers without a hub, or when you connect two hubs together, you need a **crossover cable**. Crossover cables have special arrangement, as shown in the following figure¹.

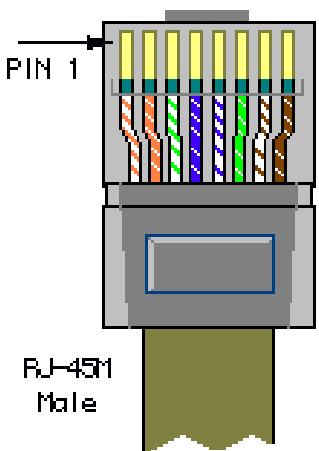
Crossover Cable			Straight Thru Cable	
RJ-45 PIN (PC)	RJ-45 PIN (PC)		RJ-45 PIN (PC)	RJ-45 PIN (Hub)
1 Rx+	3 Tx+	1 Tx+	1 Rx+	
2 Rx-	6 Tx-	2 Tx-	2 Rx-	
3 Tx+	1 Rx+	3 Rx+	3 Tx+	
4 NA	4 NA	4 NA	4 NA	
5 NA	5 NA	5 NA	5 NA	
6 Tx-	2 Rx-	6 Rx-	6 Tx-	
7 NA	7 NA	7 NA	7 NA	
8 NA	8 NA	8 NA	8 NA	

Fig. 12-17. RJ-45 standard connector and corresponding UTP cable connections.

If you are connecting many computers via a hub or a switch, you need straight-through **patch cables**. As we mentioned so far, there are different grades of cable quality. The most common are CAT5, CAT5e and CAT6. CAT5 is good for most purposes and can transfer data at 100Mbps. **CAT5e** is rated for 200 Mb/s and **CAT6** is rated for gigabit Ethernet. Ethernet cables are limited in their reach, and these distances (about 100m) are insufficient to cover medium and large network installations. A **repeater** in Ethernet networking is a device that allows multiple cables to be joined and greater distances to be spanned.

12-3.2. Fast Ethernet Links & Gigabit Ethernet

The Fast Ethernet standard (IEEE 802.3u) has been established for Ethernet networks that need high transmission speeds. This standard raises the Ethernet speed limit from 10Mb/s to 100Mb/s with minimal change of cable structure.

There are three types of Fast Ethernet:

- 100BASE-TX for use with category 5 (Cat5) UTP cable;
- 100BASE-FX for use with fiber-optical cable; and
- 100BASE-T4 which utilizes extra 2 wires for use with category 3 (Cat3) UTP cable.

¹ The shown crossover connections are valid for 10/100 Mbps. For 1Gbps, crossover connections are the same, except for the NA pins (4, 5, 7, 8). In 1Gbps crossover connection, connect 4 \leftrightarrow 7 and 5 \leftrightarrow 8 on both sides.

Gigabit Ethernet is a new technology that promises a migration path beyond Fast Ethernet so the next generation of networks will support even higher data transfer speeds. Gigabit Ethernet was developed to meet the need for fast networks with multimedia applications. Gigabit Ethernet is also known as gigabit-Ethernet-over-copper or 1000Base-T.

The **10 Gigabit** Ethernet is the fastest and most recent of the Ethernet standards. **IEEE 802.3ae** defines a version of Ethernet with a nominal rate of 10Gb/s that makes it 10 times faster than Gigabit Ethernet. Unlike other systems, 10 Gigabit Ethernet is based on the use of optical fibers.

Table 12-3. Different categories of Ethernet cables.

Category	Specification	Data Rate (Mbps)	Use
1	Unshielded twisted-pair used in telephone	< 0.1	Telephone
2	Unshielded twisted-pair used in T-lines	2	T-1 lines
3	Improved CAT 2 used in LANs	10	LANs
4	Improved CAT 3 used in Token Ring networks	20	LANs
5	Cable wire is normally 24 AWG with a jacket and outside sheath	100	LANs
5E	An extension to category 5 that includes extra features to minimize crosstalk and interference	125	LANs
6	A new category with matched components coming from the same manufacturer.	200	LANs
7	Sometimes called SSTP (shielded screen twisted-pair). The shield decreases the effect of crosstalk and increases the data rate.	600	LANs

12-3.3. Ethernet Repeaters, Hubs and Switches

Repeaters, Hubs and switches are used to connect together two or more Ethernet segments of any type. **Hubs** or repeaters are simple devices that interconnect groups of users. Hubs provide the signal amplification required to allow a segment to be extended a greater distance. A hub repeats any incoming signal to all ports. A multi-port twisted pair hub allows several point-to-point segments to be joined into one network. One end of the point-to-point link is attached to the hub and the other is attached to the computer. If the hub is attached to a backbone, then all computers at the end of the twisted pair segments can communicate with all the hosts on the backbone. **Switches** act as full-duplex traffic cops making your network more efficient. As shown in figure 12-18, a group of PC's can be connected via hubs and switches.

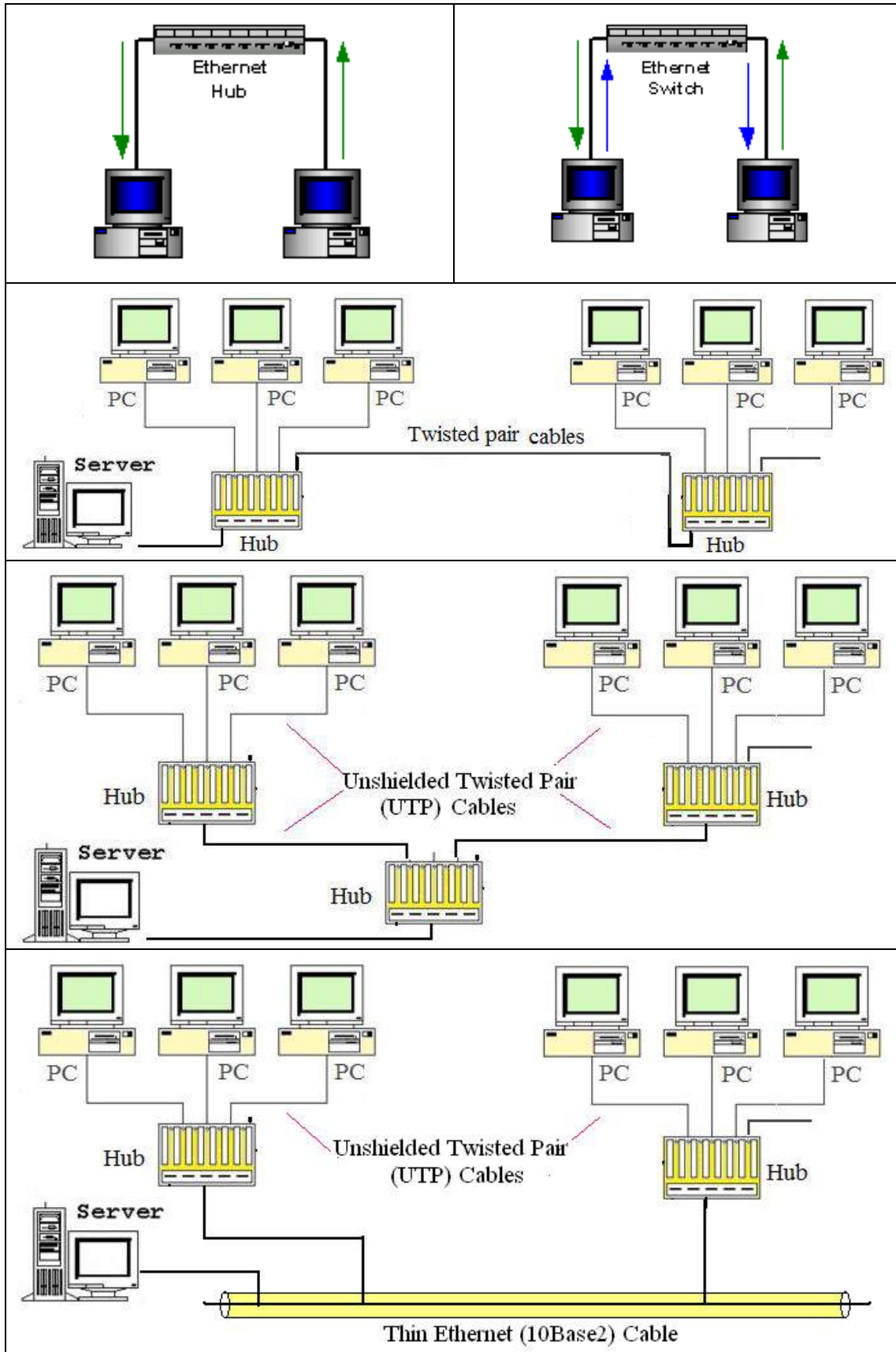


Figure 12-18. Networking of PC's via hubs or switches, with different configurations.

Figure 12-19 shows a typical Ethernet switch, which looks like a data concentrator (hub), but actually it is not. Switches are a lot smarter than data concentrators (hubs) and operate on the second layer of the OSI model. This means that a switch won't simply receive data and transmit it throughout every port of the network, but it will read the data and find out the packet destination by checking the Ethernet address.

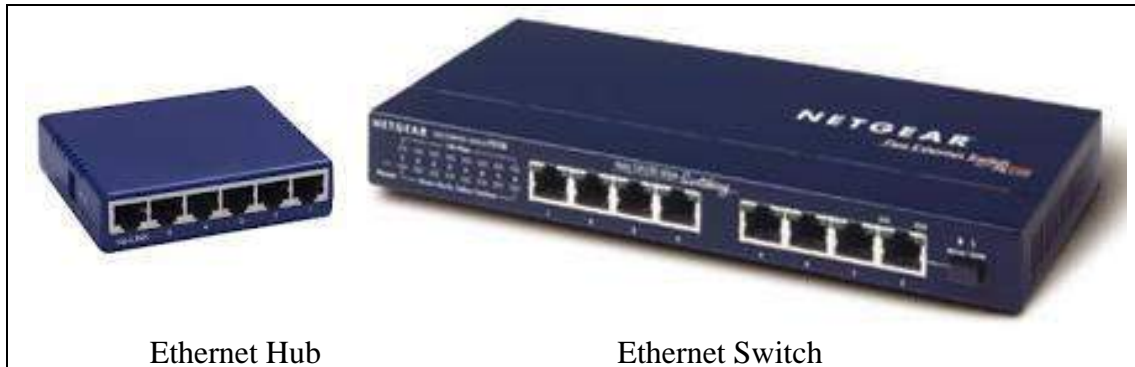


Figure 12-19. Photograph of an Ethernet hub and an Ethernet switch.

Generally speaking, switches come in 3 different types: Store & Forward, Cut-Through and Fragment Free.

i- Store & Forward Mode

This is one of the most popular switching methods. In this mode, when the switch receives a frame from one of its ports, it will store it in memory, check it for errors and corruption, and if it passes the test, it will forward the frame out the designated port, otherwise, if it discovers that the frame has errors or is corrupt, it will discard it. This method is the safest, but also has the highest latency.

ii- Cut-Through (Real Time)

Cut-through switching is the second most popular method. In this mode, the switch reads the frame until it learns the destination MAC address of the frame it's receiving. Once it learns it, it will forward the frame straight out the designated port without delay. This is why we say it's -Real Time, there is no delay or error checking done to the frame.

iii- Fragment Free

The fragment free switching method is mainly used to check for *frames* which have been subject to a collision. The frame's first 64 bytes are only checked before forwarding the frame out the designated port. Reason for this is because almost all collisions will happen within the first 64 bytes of a frame. If there is a corruption in the first 64 bytes, it's most likely that that frame was a victim of a collision.

12-3.4. Ethernet Adapter Cards

Network Interface Cards (NICs) are used to connect a computer to a network. Most NICs are internal, with the card fitting into an expansion slot inside the computer. The NIC provides a physical connection between the networking cable and the computer's internal bus. Different computers have different bus architectures. Most NICs are designed for a particular type of network, protocol, and medium, though some can serve multiple networks. The three most common network interface cards are Ethernet cards, LocalTalk cards, and Token Ring cards. Many NIC adapters comply with plug-and-play specifications. On these systems, NICs are automatically configured without user intervention, while on non-plug-and-play systems; configuration is done manually through a set-up program. Cards are available to support almost all networking standards. Fast Ethernet NICs are often 10/100 Base-T capable, and will automatically set to the appropriate speed. Gigabit Ethernet NICs are 10/100/1000 Base-T capable with auto negotiation depending on the Ethernet speed.

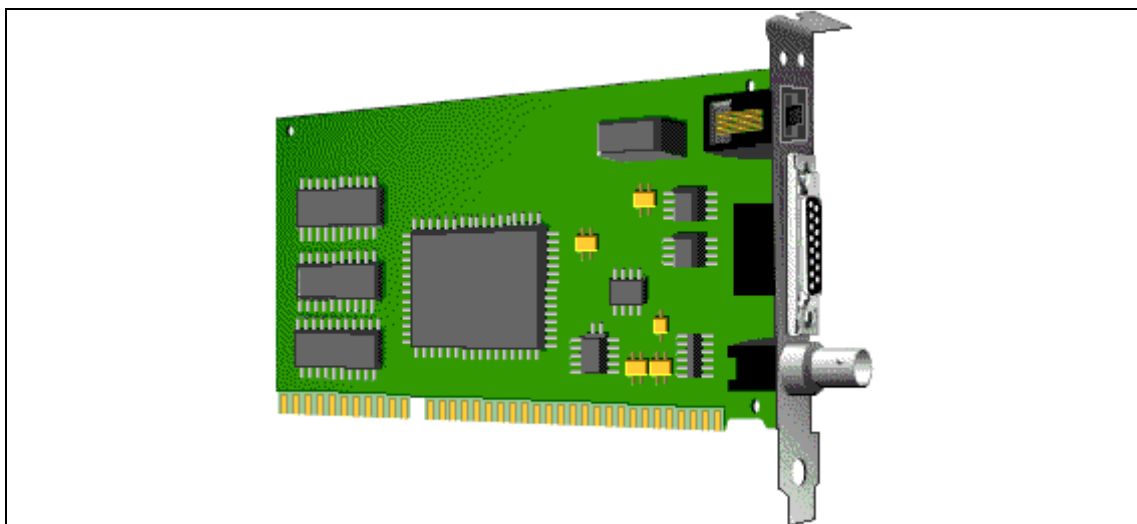


Fig. 12-20. Schematic of a network interface card (NIC).

12-3.5. Wireless Ethernet (IEEE 802.11b)

The wireless Ethernet technology extends the geographic coverage of Ethernet networks by enabling them to access even far places or the locations which are hard to reach by cables. In 2000, the IEEE **802.11b** standard became the standard wireless Ethernet networking technology for wireless LAN's (WLAN's). The 802.11b is a half duplex protocol – it can send or receive, but not both at the same time.

An 802.11b wireless network adapter can operate in two modes, Ad-Hoc and Infrastructure. In infrastructure mode, all your traffic passes through

a wireless **access point**, as shown in figure 12-21. In Ad-hoc mode your computer talks directly to other computers and does not need any access point. The so-called *Wi-Fi* technology refers to a set of wireless networking technologies more specifically referred to as **802.11a** **802.11b** and **802.11g**. The **Wi-Fi** organization was created to ensure interoperability between 802.11b products. These standards are universal, and allow users that have a Wi-Fi devices, like a laptop or PDA, to connect anywhere (where a Wi-Fi access point is available). The three Wi-Fi standards signify the speed of the connection they are capable of delivering.

The **802.11b** (which transmits at 11 Mb/s) is the most common, although it is quickly getting replaced by the faster Wi-Fi standards. Both **802.11a** and **802.11g** are capable of 54 Mb/s. Generally speaking, all of these Wi-Fi standards are fast enough to generally allow a broadband connection. Table 12-3 shows a comparison between different communication networking technologies.

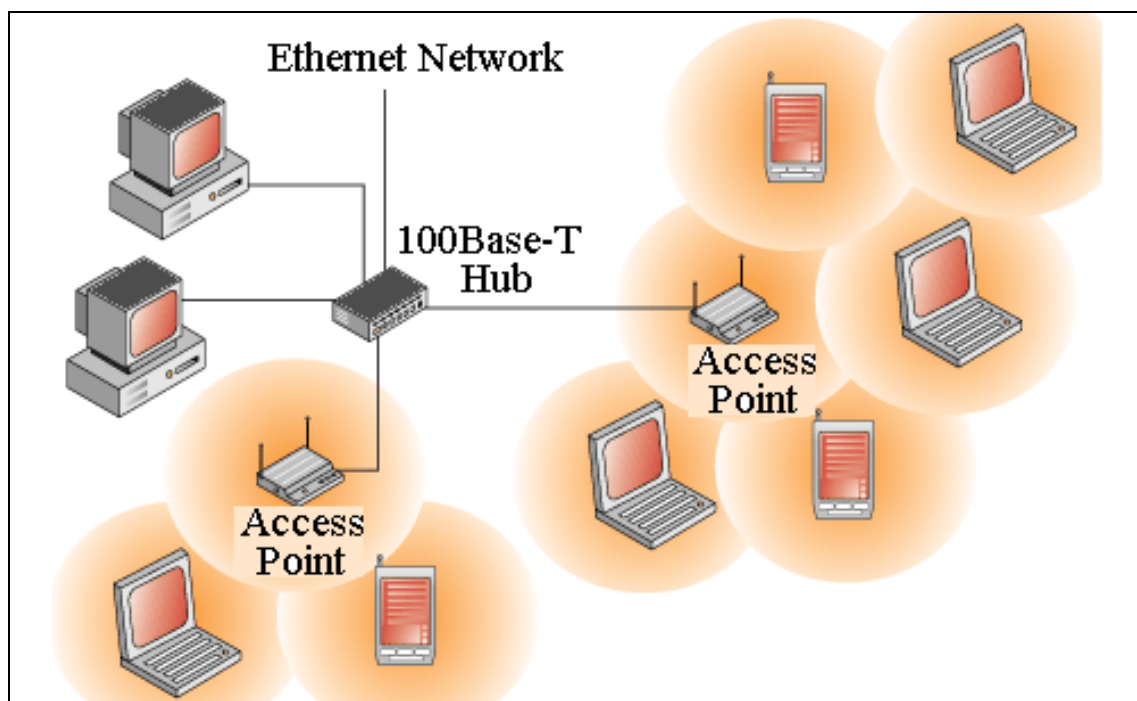


Fig. 12-21. Illustration of a wireless Ethernet network

The 802.11b uses the same 2.4GHz carrier as many cordless and cell phones so plenty of opportunity exists for interference. In addition, any network adapter coming within range of another 802.11b network adapter or access point can instantly connect and join the network unless **WEP** – wireless encryption protocol – is enabled. WEP is secure enough for most

homes and small business. Encryption comes in 64bit or 128bit key varieties. All your nodes must be at the same encryption level with the same key to operate. However, there are still some flaws making WEP unusable for high security applications, such as e-business and mobile commerce (**m-commerce**). The following table depicts the different communication networking technologies, with their respective speed and cost. Note that the Bluetooth technology is different from the Wi-Fi technology in both range and speed.

Table 12-4. Comparison between different communication networking technologies. The A letter designate the highest range (or cost) while D designates the lowest one.

Technology	Speed	Type	Range	Cost
FireWire	400 Mb/s	Cables	D ↓	A ↑
Ethernet 10/100	10/100 Mb/s	Cables	A	A
Gigabit Ethernet	1000 Mb/s	Cables	A	A
10Gbit Ethernet	10 Gb/s	Optic fibers	A	D
IrDA	1.15 Mb/s	Wireless, Infrared	D	C
Wi-Fi 802.11b	11 Mb/s	Wireless, RF	B	B
Wi-Fi 802.11a	52/72 Mb/s	Wireless, RF	C	C
Wi-Fi 802.11g	22/54 Mb/s	Wireless, RF	C	NA
Bluetooth	1.5 Mb/s	Wireless, RF	D	C
PowerLine (PLC)	15 Mb/s	Cables	D ↓	A ↑

12-4. Wide Area Networks (WAN's) & their Physical Layer

Until now we have only examined networking within a local area, which may be in a single building or a single factory. However, it is important from a communications point of view to be aware of the networks that link computer-based devices across the world. These Wide Area Networks (WAN's) clearly have important consequences for large organizations where international data transfers need to occur on a regular basis. However, even individuals may need to access large area networks, such as the internet, for instance to download a piece of information. Initially data communication between computer systems, separated by large distances, was carried out through the normal lines on the Public Switched Telephone Network (PSTN). Computers transmitted data to one another on these public telephone lines via **modems**. Unfortunately because of the channel bandwidths on public lines and the switching delays, data transfer rates were generally low and the timed charges imposed by the telephone companies were high.

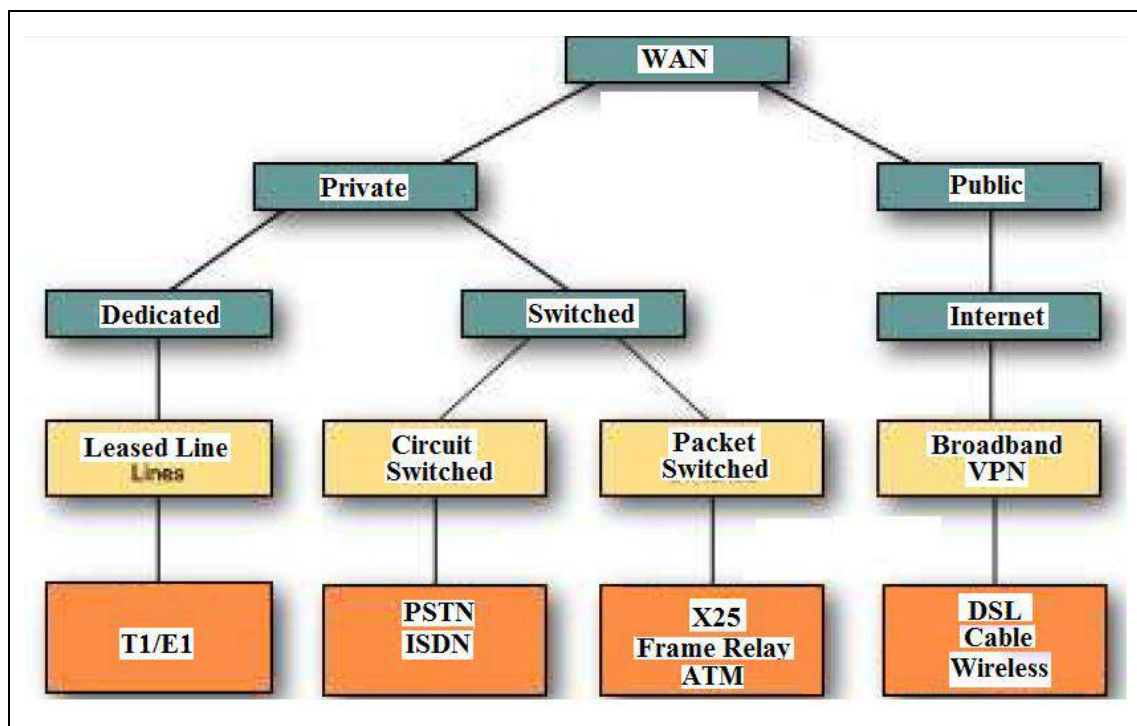


Fig. 12-22. WAN link connections and services

Many large computer organizations and financial institutions therefore chose to introduce their own private networks by leasing dedicated public lines from telephone companies and providing their own exchanges (switching nodes). These allowed companies to decrease the switching delays, maximize transmission bandwidth and speed.

Private data networks functioned well within a single organization or within a single network, but problems arose when one organization wanted to transfer its computer data to another organization that was on a different, private network. It became evident that a public, wide area data transfer system, analogous to the switched telephone network, had to be introduced. This system is referred to as a Public Data Network or **PDN**.

12-4.1. Public Data Networks (PDN)

Public Data Networks are generally established and operated by a national administrative body in order to maintain standards. The traditional, switched telephone network is still widely used for data communications, and therefore, it is also a PDN. There are essentially two different forms of Public Data Networks. These are the **Packet Switched** Data Networks (**PSDN**) and the **Circuit Switched** Data Networks (**CSDN**). The standards that are used in conjunction with both these types of Public Data Networks are those which occupy the "Network Layer" of the OSI 12-Layer model.

12-4.2. Switching Networks.

A switched network goes through a switch instead of a router. This actually is the way most networks are headed, toward flat switches instead of routers. Data switching networks, that is, networks which are explicitly designed for data transmission via switches, can be broadly divided into two major categories:

- 1- **Circuit switching** networks
- 2- **Packet switching** networks

In **circuit-switched networks**, network resources are static, set over copper wires, from the sender to receiver, thus creating a circuit. The resources remain dedicated to the circuit during the entire transfer and the entire message follows the same path. Examples of circuit switching networks, are the public switching telephone networks (**PSTN**) and the GSM cell-phone network. In such networks, each circuit cannot be used by other callers until the circuit is released and a new connection is set up. In **packet-switched networks**, the message is broken into small entities called frames or packets. Each packet is labeled with its destination. These packets can take different routes to the destination where the packets are assembled into the original message.

12-4.3. Circuit Switched Data Networks (CSDN)

A circuit switched network is one in which a group of intermediate exchanges set up a direct physical (electric circuit) connection between a transmitting device and a receiving device by short-circuiting appropriate incoming lines to outgoing lines. The circuit remains connected for the duration of a transaction (call), just like a normal telephone line works for voice communication. In other words, the transmitter and receiver are linked by a cable. This is shown schematically in figure 12-23. The public switching telephone network (**PSTN**) is a good example of a circuit switched network, where exchanges perform electrical switching of transmission lines. However, recent PSTN's are quickly moving towards packet switching schemes. Integrated Services Digital Network (**ISDN**) is another example of circuit switching network

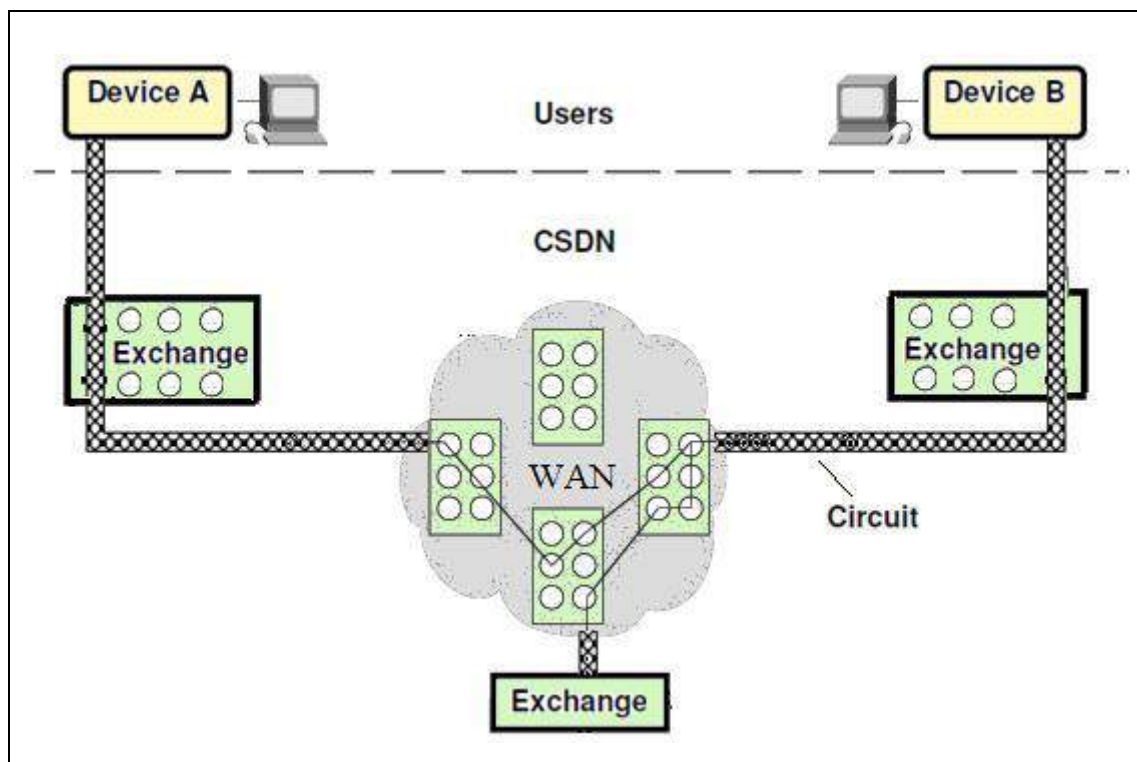


Figure 12-23. Circuit switched data network (CSDN) operation

12-4.4. Packet Switched Data Networks (PSDN)

The so-called packet switching techniques are used for fast communication of large amount of data. In a packet switched data network (**PSDN**), all messages (regardless of length) are divided into discrete units (packets), which are transmitted from a source station to a destination through intermediate exchanges. Each packet contains a **source** and **target address** that intermediate exchanges use for routing the packet. A transmitter sends a packet to its local exchange. The local

exchange reads the destination address and uses its "routing directory" to determine the next exchange to which the packet must be sent. Each exchange is said to perform a "packet store and forward" operation. Unlike the CSDN there is no direct, physical connection or communication between a transmitter and receiver - only between consecutive exchanges. This system is shown schematically in figure 12-24. Many transmitters can access the same exchange and hence the exchange will not necessarily forward packets in the consecutive order in which they are received from any one transmitter. The data traveling from one exchange to another is a mixture of packets from different transmitters. There is a maximum length of packets, defined by the network protocol, and hence no transmitter can block the network with long messages. When packets are transmitted as individual entities, as described above, the packet switched network is said to be operating in a "datagram" mode. However, it is also possible for the exchanges to set up packet transfer so that two communicating devices believe they are talking to one another through a physical circuit. This is referred to as a "virtual call" or "virtual circuit" mode of operation. Although the two user devices are never directly communicating, the exchanges make it appear as though this was the case. Some examples of packet-switching networks include X.25, Asynchronous Transfer Mode (ATM), Frame Relay, and Switched Multimegabit Data Services (SMDS).

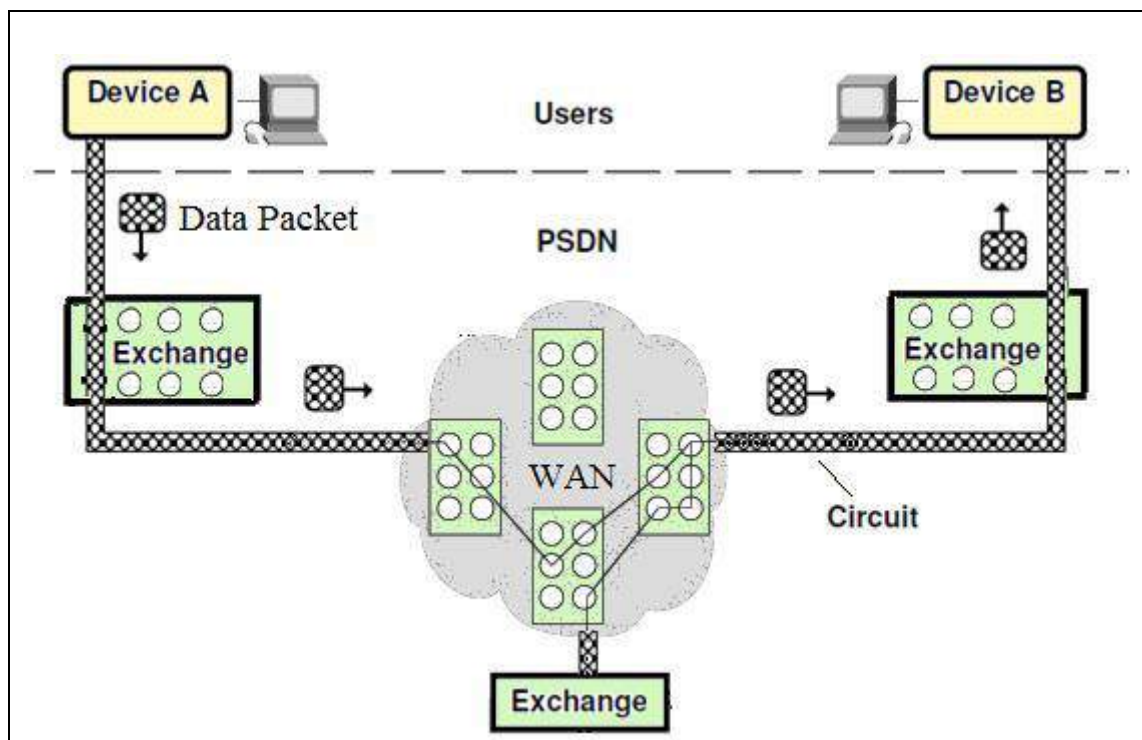


Figure 12-24 - Packet switched data network operation

12-5. Data Link Protocols

In the previous sections, we examined many aspects of the OSI model physical layer (OSI layer 1). The issues that are covered by the physical layer include network topology, communications media, modulation techniques, and traffic control (contention schemes). In this section, we examine some of the **data link layer** (OSI layer 2) protocols that are in common use with various types of networks. The Data Link layer exists as a connecting layer between the software processes of the layers above it and the Physical layer below it. As such, it prepares the Network layer packets for transmission across some form of media, be it copper, fibre, or the atmosphere (air). In many cases, the Data Link layer is embodied as a physical entity, such as an Ethernet network interface card (NIC),

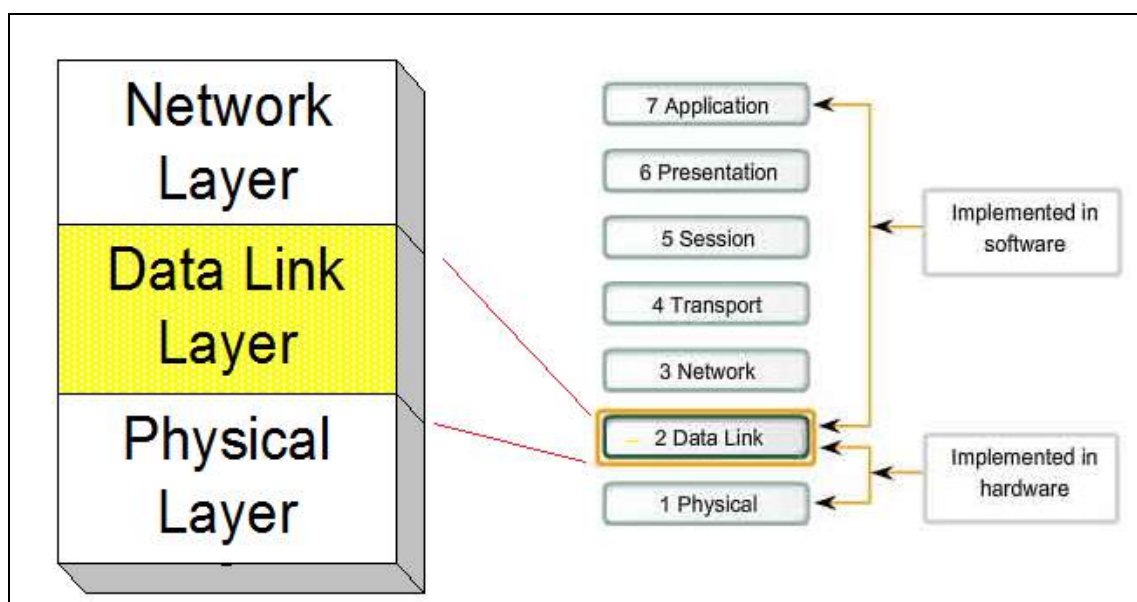


Figure 12-25. The data link layer in the OSI model.

The main tasks of the data link layer are:

- Allows the upper layers to access the media using framing techniques
- Controls how data is placed onto the media and is received from it using techniques such as media access control and error detection

As we'll see in the next sections, the description of a **frame** is a key element of each Data Link layer protocol. Therefore, the protocol data unit (**PDU**) of the Data Link layer is the **frame**. The Data Link layer prepares a packet for transport across the local media by encapsulating it with a header and a trailer to create a frame. Unlike other, the Data Link layer **frame** includes:

- Data - The packet from the Network layer
- Header - Contains control information, such as addressing, and is located at the beginning of the PDU
- Trailer - Contains control information added to the end of the PDU

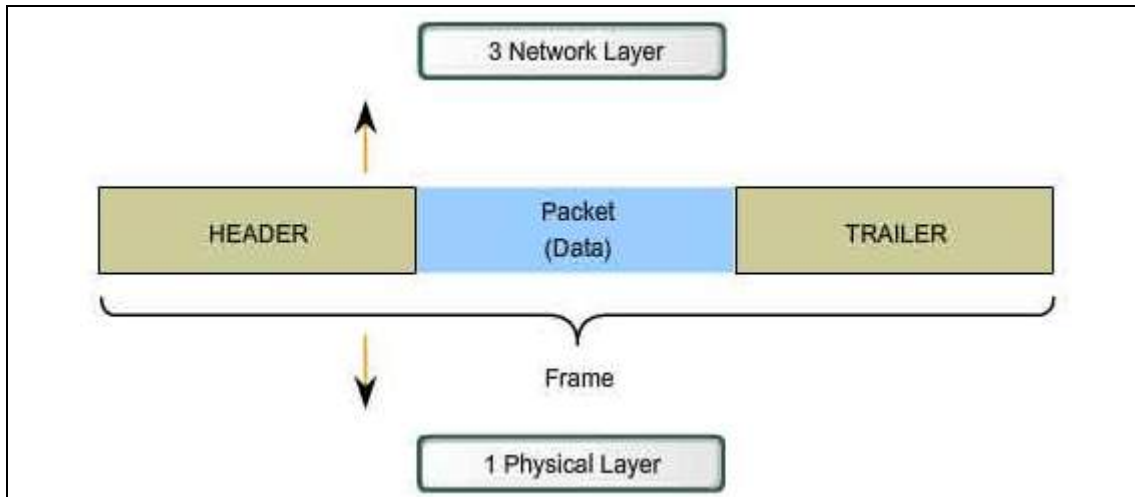


Figure 12-26. Framing data for transmission in the data link layer.

When data travels on the media, it is converted into a stream of bits, or 1s and 0s. If a node is receiving long streams of bits, how does it determine where a frame starts and stops or which bits represent the address? **Framing** breaks the stream into decipherable groupings, with control information inserted in the header and trailer as values in different fields. This format gives the physical signals a structure that can be received by nodes and decoded into packets at the destination. Typical field types include:

- Start and stop fields - The beginning and end limits of the frame
- Addressing or Naming fields
- Type field - The type of PDU contained in the frame
- Quality - control fields
- A data field -The frame payload (information to be sent)

Fields at the end of the frame form the trailer. These fields are used for error detection and mark the end of the frame. Not all protocols include all of these fields. The standards for a specific Data Link protocol define the actual frame format. Examples of frame formats will be discussed at the subsequent sections of this chapter.

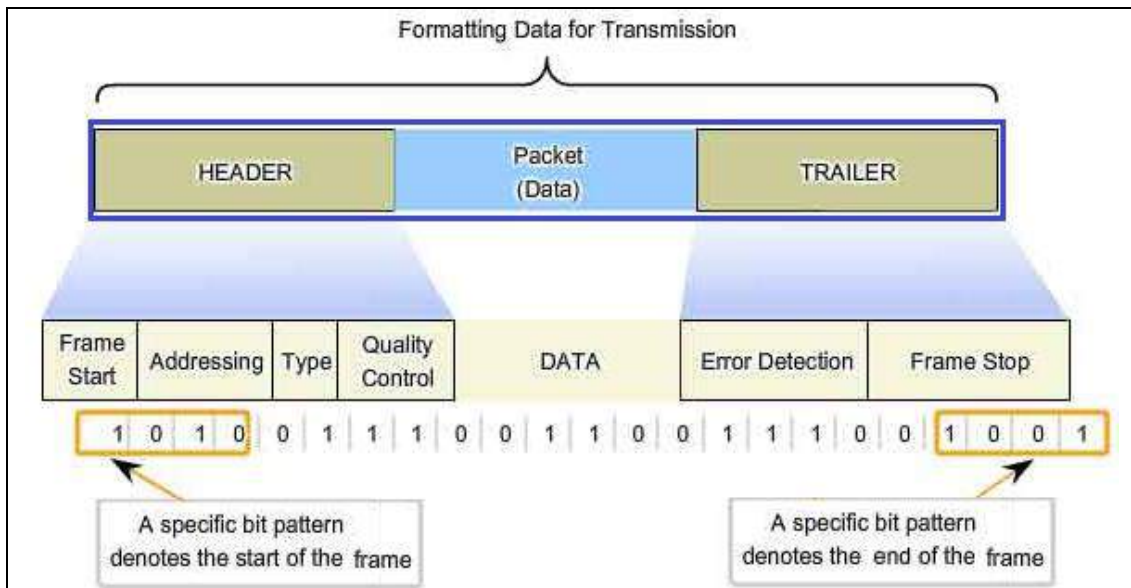


Figure 12-27. Data Link Layer - Connecting Upper Layer Services to the Media

There are a number of protocols that are commonly used to implement data link control functions. Generally speaking, there exist two main network types at the data link layer; namely:

- **Broadcast Networks:** All stations share a single channel
- **Point-to-Point Networks:** Pairs of hosts (or routers) are directly connected

Typically, local-area networks (**LANs**) are broadcast and wide-area networks (**WANs**) are point-to-point. We start by the point-to-point protocols, and then move to the broadcast ones, with emphasis on **LAN** and **Ethernet** protocols.

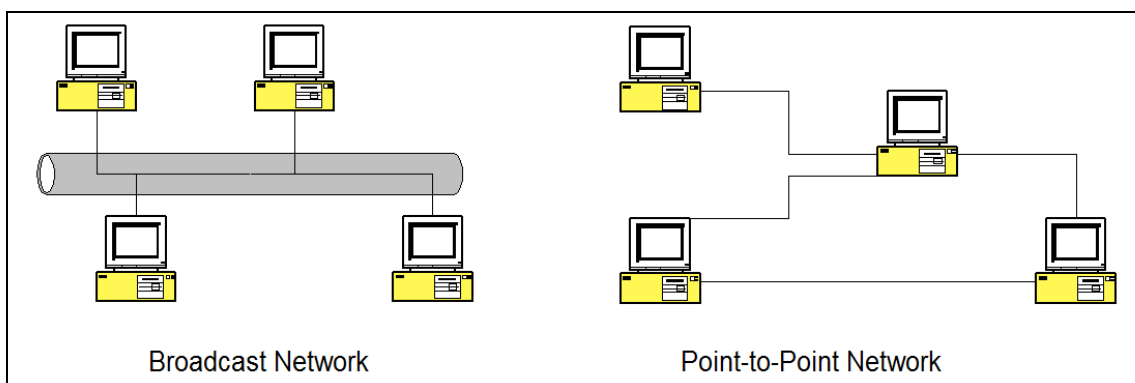


Figure 12-28. Network types at the data link layer.

12-5.1. Binary Synchronous Control (BSC) Protocol

The Binary Synchronous Control (BSC) or BiSync is an old character-oriented protocol for synchronous serial transmission. This protocol was primarily used for point-to-point communications. It was developed by IBM for data transfer between IBM System360 and its terminals. Under Bisync protocol, each block (packet) of information is preceded by sending two, or more, synchronizing characters, known as "SYN". After a prolonged idle period, a receiver synchronizes itself to a transmitter when it detects the bit patterns of these synchronizing characters. Since the BiSync protocol is commonly used with IBM computers, characters are usually represented in EBCDIC form. However the protocol also allows for ASCII and Six Bit Transcode representation.

The BiSync protocol allows for single-block or multiple-block messages, with or without headers (containing addressing information). This is because the protocol can be used for both **point to point** and **multidrop** (network) applications. A typical single-block message is shown in figure 12-29. One of the shortcomings of the BiSync protocol is that it is half-duplex in its operation. Since many network installations now provide cables with sufficient bandwidth for full-duplex transmission, BiSync effectively wastes the bandwidth that is provided for the return channel. Bisync began to be displaced in the 1970s by Systems Network Architecture (SNA) which allowed construction of a network with multiple hosts and multiple programs. The WAN protocols, such as X.25 and the Internet Protocol (IP) are later used.

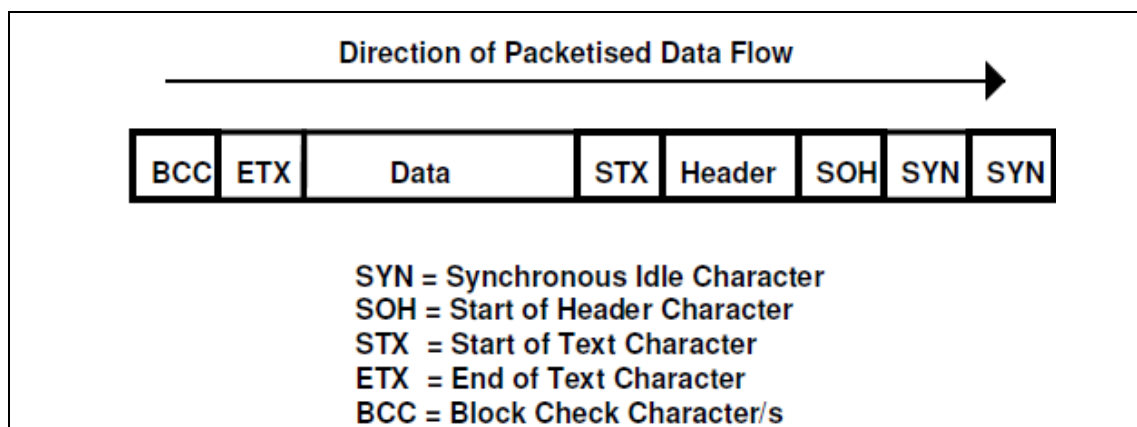


Figure 12-29. Typical block (frame) in BiSync data link protocol

12-5.2. HDLC / SDLC Protocols

The High Level Data Link Control (HDLC), is a bit-oriented, synchronous protocol. It is almost identical to IBM's Synchronous Data Link Control (SDLC). Like BiSync, HDLC is a protocol for the **data**

link layer of the ISO model. The HDLC protocol allows for **full-duplex** communications on either a simple **point to point** link or a **multidrop** network arrangement. Under the HDLC protocol, data can only be transmitted within a packet defined by a standard format. In HDLC parlance, a packet is more commonly referred to as a "frame". The structure of the bit-oriented HDLC frame is shown in figure 12-30.

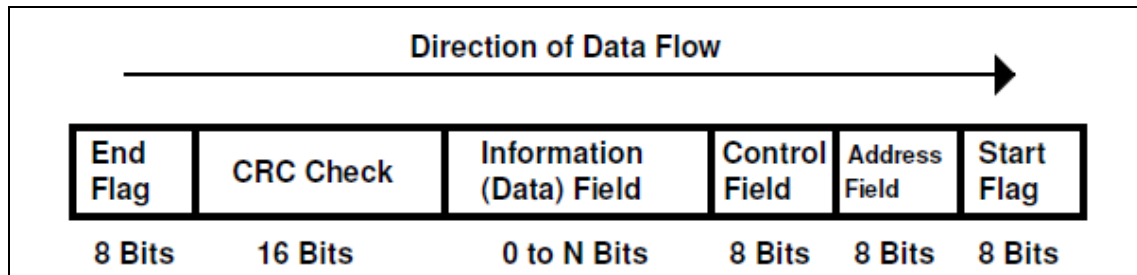


Figure 12-30. High Level Data Link Control (HDLC) frame format

The start and end flags for a HDLC frame are composed of unique bit patterns that are used to synchronize the receiving device. The integrity of each frame is checked through a Frame Check Sequence (**FCS**) which is a Cyclic Redundancy Check (CRC) polynomial. A key feature of the HDLC protocol is the addressing of frames for a networked environment. Each node on a network can be given a unique address. When a primary node transmits a frame to a secondary node, then the address of the target node is placed into the address field of the frame. It is also possible to provide a common address for a number of secondary nodes, so that all the nodes within the common address group receive a message from a primary node. This is referred to as **group addressing** and individual groups of secondary nodes can be specified. When the address field of an HDLC frame contains all ones, then the link is said to be in a **broadcast mode** and all secondary devices receive the broadcast frame from the primary node. An address field with all ones is referred to as a *broadcast address*.

The **HDLC** link protocol, like any other protocol, contains the following transaction phases:

- Link establishment
- Packet transmission
- Error check transmission
- Acknowledgment or negative acknowledgment of packets
- Termination of transmission.

HDLC was a widely used and influential standard since 1979. It was the default protocol for serial links on Cisco routers. Actually, the more recent Point-to-Point Protocol (**PPP**) is based on a variant of HDLC. The HDLC protocol is however, relatively sophisticated because the control field not only contains link establishment and supervisory functions, but also frame sequencing information.

12-5.3. Medium Access Control (MAC) & Logical Link Control (LLC)

In any broadcast network, the stations must ensure that only one station transmits at a time on the shared communication channel. The protocol that determines who can transmit on a broadcast channel are called Medium Access Control (**MAC**) protocol. The MAC protocol are implemented in the MAC sublayer which is the lower sublayer of the data link layer. The higher portion of the data link layer is often called Logical Link Control (**LLC**).

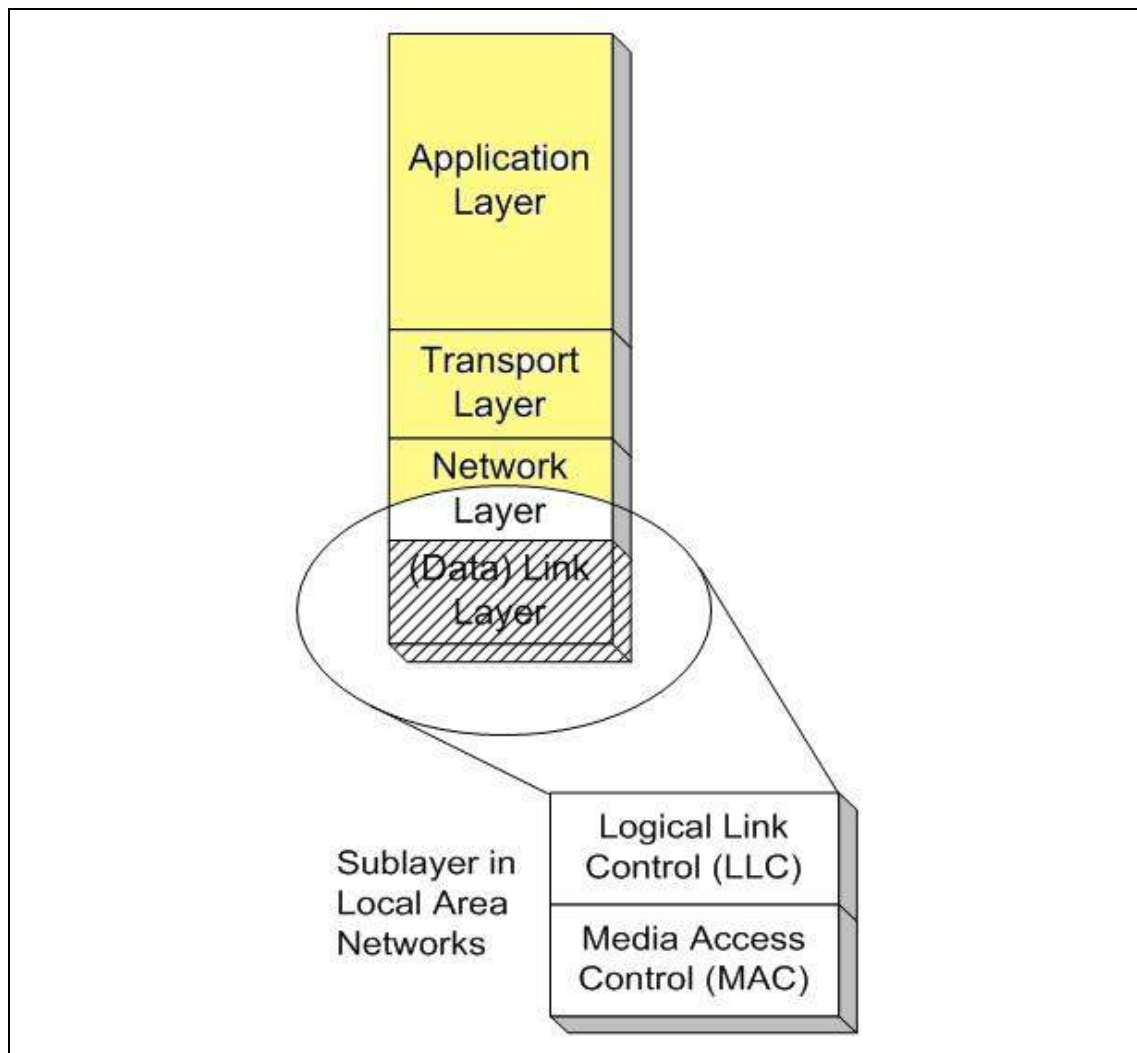


Figure 12-31. Structure of the data link layer in LAN/s.

The MAC sublayer provides addressing and full duplex channel access control mechanisms that make it possible for several terminals or network nodes to communicate within a multiple access network. Examples of common packet multiple access protocols for **multi-drop** networks are:

- Token bus
- Token ring
- CSMA/CD (used in Ethernet)
- Token passing (used in FDDI)

12-5.4. ETHERNET Protocol for LAN's

Referring to the IEEE standards for LAN's, we can say that Ethernet defines the physical layer and the data link layer of the OSI model. It should also be noted that the development of the **IEEE 802.3** specification, for baseband communications on a bus network, was based upon the Ethernet specification and not vice-versa.

The actual Ethernet data frame is shown in figure 12-32. It is similar to the HDLC frame in its form. It consists of a 7 byte preamble, followed by a single byte Starting Frame Delimiter (**SFD**), two or six byte Destination and Source Addresses (**DA** and **SA**), data length specification field, data and padding bits and finally a Frame Check Sequence (**FCS**). The length of the preamble is designed to allow for receiver synchronization and consists of alternating 1s and 0s. The padding bits are added if there are insufficient bytes in the data (provided by the LLC) for the protocol to operate. The Ethernet system is relatively simple and it functions extremely well in the office environment. This is why it is in such widespread use. However, it has already been noted that the CSMA/CD system is non-deterministic in nature and as such may be undesirable within an industrial environment because message delays cannot be predicted with any certainty. Another serious shortcoming of the **IEEE 802.3** standard is the length limitation. This restricts the length of the CSMA/CD system. Whilst this is an acceptable restriction in a typical office environment, it can present problems in large industrial sites.

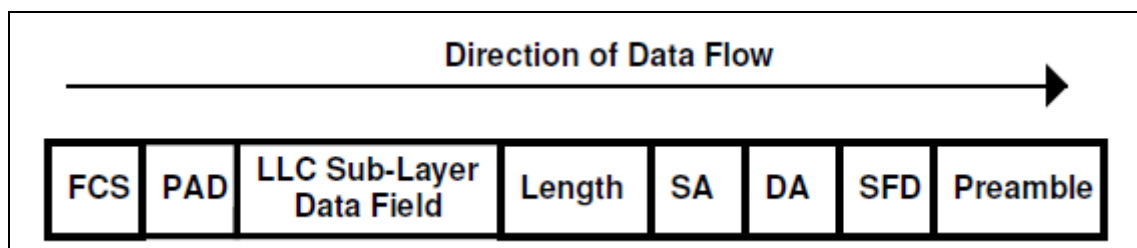


Figure 12-32, Ethernet Frame

An **Ethernet frame** contains a header, a data section, and a footer having a combined length of no more than 1518 bytes. The Ethernet header contains the addresses of both the intended recipient and the sender. Data sent over the Ethernet is automatically **broadcast** to all devices on the network. By comparing their Ethernet address against the address in the frame header, each Ethernet device tests each frame to determine if it was intended for them and reads or discards the frame as appropriate. Devices wanting to transmit on the Ethernet first perform a preliminary check to determine whether the medium is available or whether a transmission is currently in progress. If the Ethernet is available, the sending device transmits onto the wire. It is possible, however, that two devices transmit at the same time. When this **collision** occurs, it causes both transmissions to fail and both devices should re-transmit again. As a performance tradeoff, the Ethernet standard does not prevent such simultaneous transmissions. Ethernet uses an algorithm based on random delay times to determine the proper waiting period between re-transmissions.

12-5.5. PPP (Point-to-Point Protocol) for WANs

PPP is the successor of the Serial Line IP (SLIP) protocol. PPP is based on HDLC and is very similar. It was introduced in 1992, with added functionality for dial-up and high-speed routers. Today, PPP is used for most dial-up connections to the Internet. The frame format of PPP is similar to HDLC and the 802.2 LLC frame format:

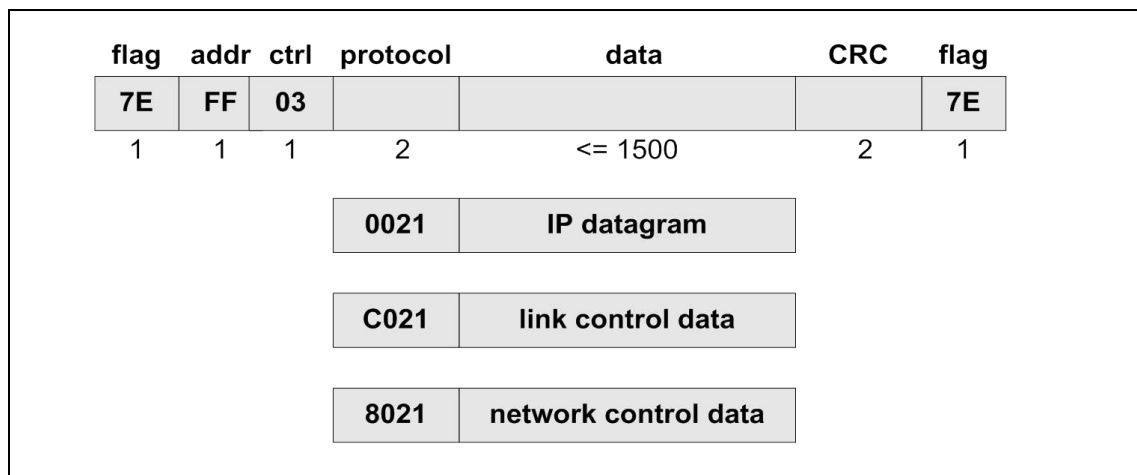


Figure 12-33 . Point-to-point protocol (PPP) frame

Note that PPP does not use addresses. In addition to encapsulation, PPP supports the following services:

- Error detection
- Address notification

- Authentication
- multiple network layer protocols (protocol multiplexing)
- Link configuration
- Link quality testing

12-6.. Internet Protocol (IP) and TCP/IP Suite

The **Internet** is a global system of interconnected computer networks that interchange data by **packet switching** using the standard Internet Protocol (IP) and Transfer Control protocol (TCP/IP). This protocol suite can be used to communicate across any set of interconnected networks and is suitable for LAN and WAN communications. The most prominent component of the Internet model is the Internet Protocol (**IP**) which provides addressing systems for computers on the Internet and facilitates the internetworking of networks.

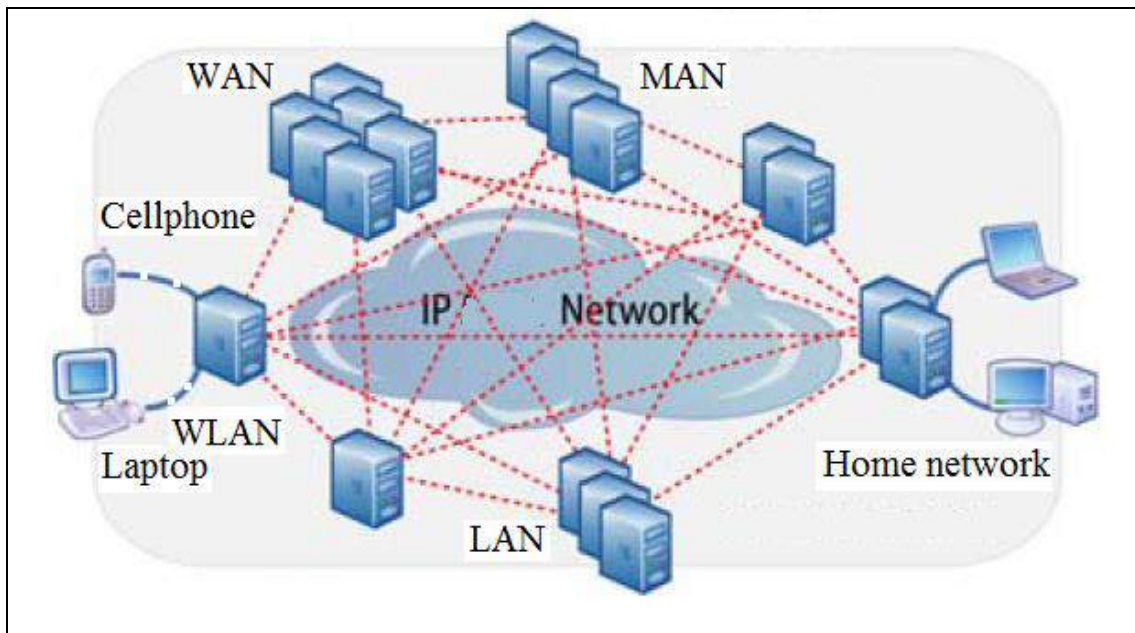


Figure 12-34. Illustration of the frame relay

12-6.1. IP Addressing (IPv4)

The Internet Protocol (**IP**) is a connectionless datagram protocol. IP Version 4 (**IPv4**) is the initial version used on the first generation of Internet and is still in dominant use. It was designed to address up to ~4.3 billion (10^9) Internet hosts. Figure 12-35 depicts the internet protocol suite, and its inter-relation to the OSI model layers. As shown in figure, the IP suite have many services, such as the file transfer protocol (**FTP**), messaging transfer protocol (**SMTP**). IP datagrams are referred to as Internet Protocol Data Units (**IPDUs**).

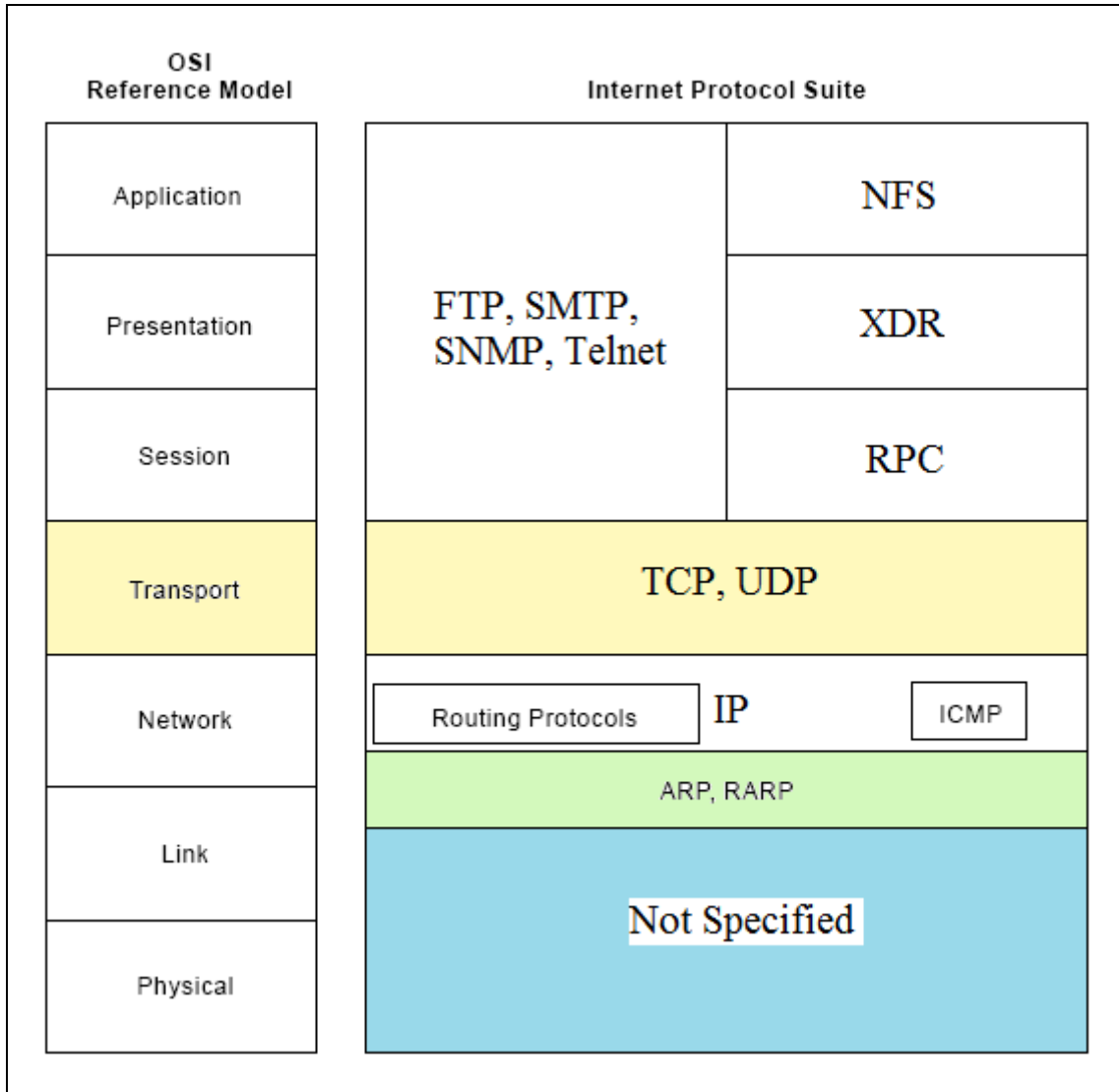


Figure 12-35. The internet protocol suite, and its inter-relation to the OSI model.

Figure 12-36 depicts the IP packet format. Both of the IP source and destination address is a 32-bit field.

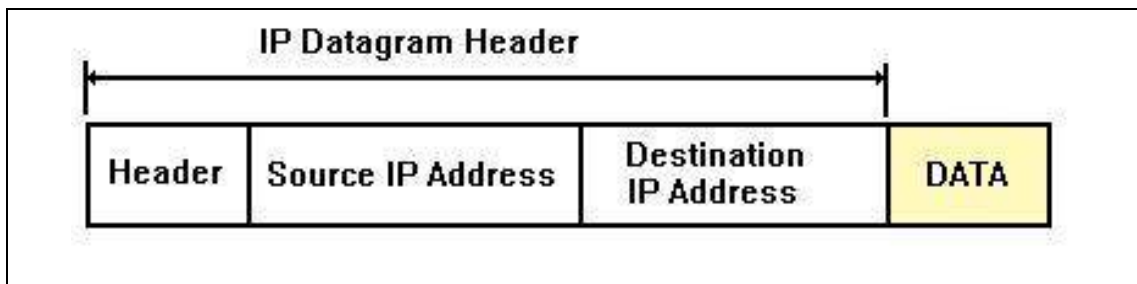


Figure 12-36(a). Internet protocol datagram (frame)

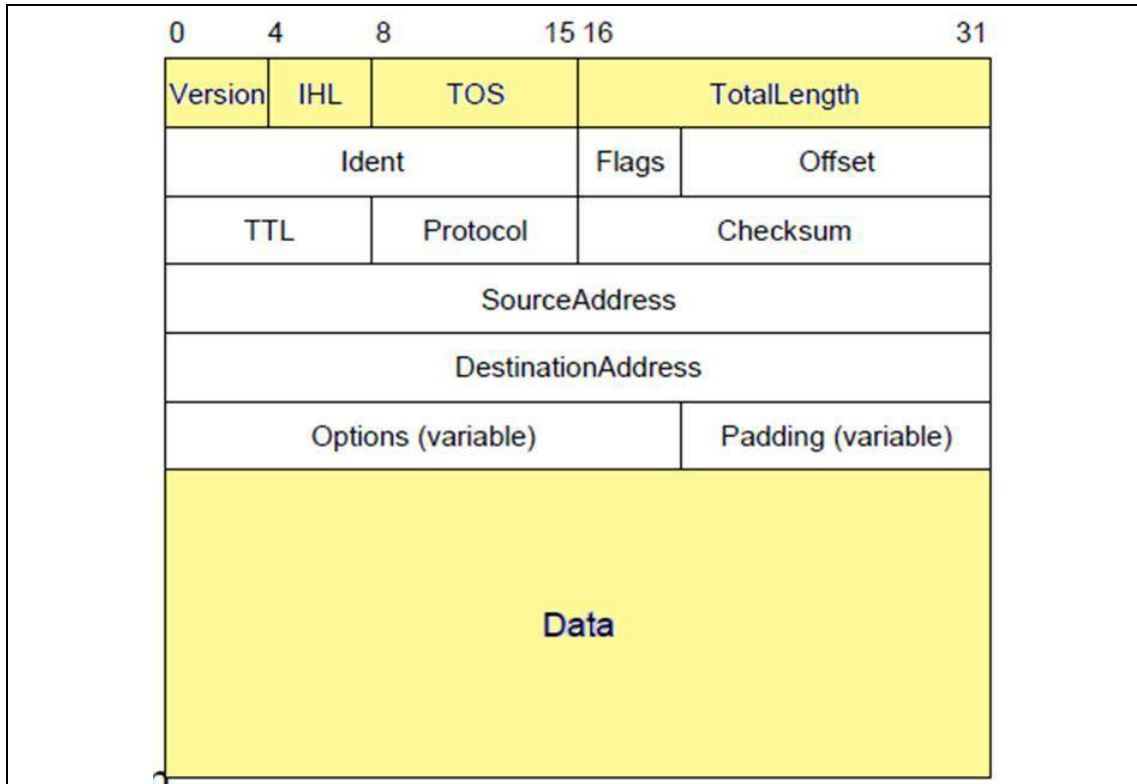


Figure 12-36(b). Details of the IP Datagram (frame).

In IPv4, each unique address consists of 4 integers between 0 and 255, usually separated by dots when written down, e.g. 172.16.254.1

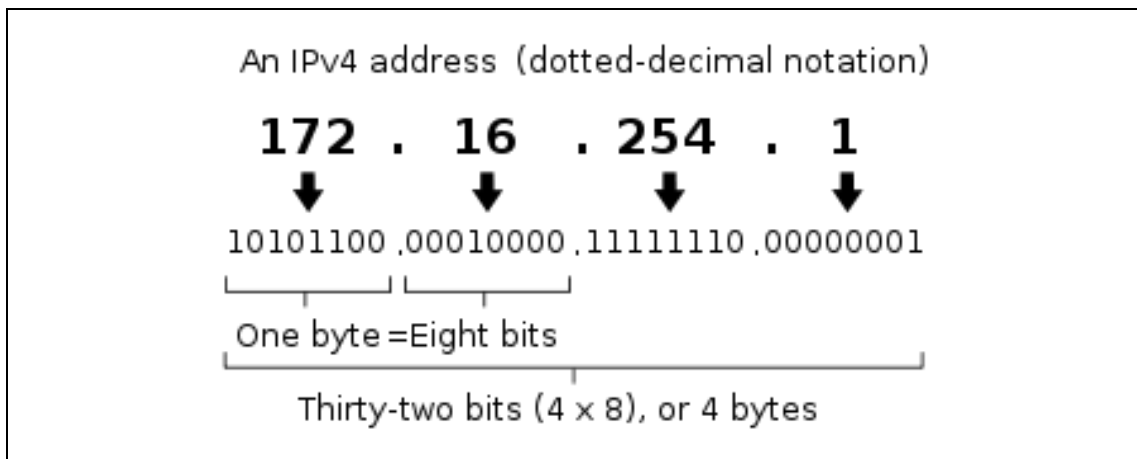


Figure 12-37. IP address notation

The IPv4 address space can be subdivided into 5 **classes** - Class A, B, C, D and E. Each class consists of a contiguous subset of the overall IPv4 address range. With a few special exceptions explained further below, the values of the leftmost four bits of an IPv4 address determine its class as shown in the following table:

Table 12-5. Internet address classes

Class	Leftmost bits	Start address	Finish address
A	0xxx	0.0.0.0	127.255.255.255
B	10xx	128.0.0.0	191.255.255.255
C	110x	192.0.0.0	223.255.255.255
D	1110	224.0.0.0	239.255.255.255
E	1111	240.0.0.0	255.255.255.255

	1 st Byte	2 nd Byte	3 rd Byte	4 th Byte
Class A	Network ID	Host ID		
Class B	Network ID		Host ID	
Class C	Network ID			Host ID

IP Address Class E is reserved for Broadcast. This is a special type of IP address, which has the address **255.255.255.255**. A broadcast involves delivering a message from one sender to many recipients. On local area networks (LANs), senders may direct an IP broadcast to 255.255.255.255 to indicate that all other nodes on the network should pick up that message. The IP address **127.0.0.1** is the **loopback** address in IP. Loopback is a test mechanism of network adapters. Messages sent to 127.0.0.1 do not get delivered to the network. Instead, the adapter intercepts all loopback messages and returns them to the sender. IP applications often use this feature to test the network interface.

When setting up each node with its IP address, a **Netmask** must be specified. This mask is used to specify which part of the address is the network number, and which is the host part. This is accomplished by a logical bitwise-AND between the Netmask and the IP address. The result specifies the network number. For Class C, the Netmask will always be 255.255.255.0; for Class B, the Netmask is always 255.255.0.0; and so on. When A sent a packet to E in the last example, A knew that E wasn't on its network segment by comparing A's network number 200.1.2 to the value resulting from the bitwise-AND between the Netmask 255.255.255.0 and the IP address of E, 200.1.3.2, which is 200.1.3.

12-6.2. The **Transmission Control Protocol (TCP)**

The Transmission Control Protocol (TCP) sits directly on top of IP. TCP accepts arbitrary size messages from a service user and, if necessary, **segments** them into smaller blocks. The TCP provides reliable transmission of data in an IP environment. TCP corresponds to the transport layer (Layer 4) of the OSI reference model. Among the services TCP provides are stream data transfer, reliability, efficient flow control, full-duplex operation, and multiplexing. As shown in figure 12-38, the TCP packet has 12 fields. In particular, the *Source Port* and *Destination Port* identify points at which upper-layer source and destination processes receive TCP services and *Data* contains the upper layer information.

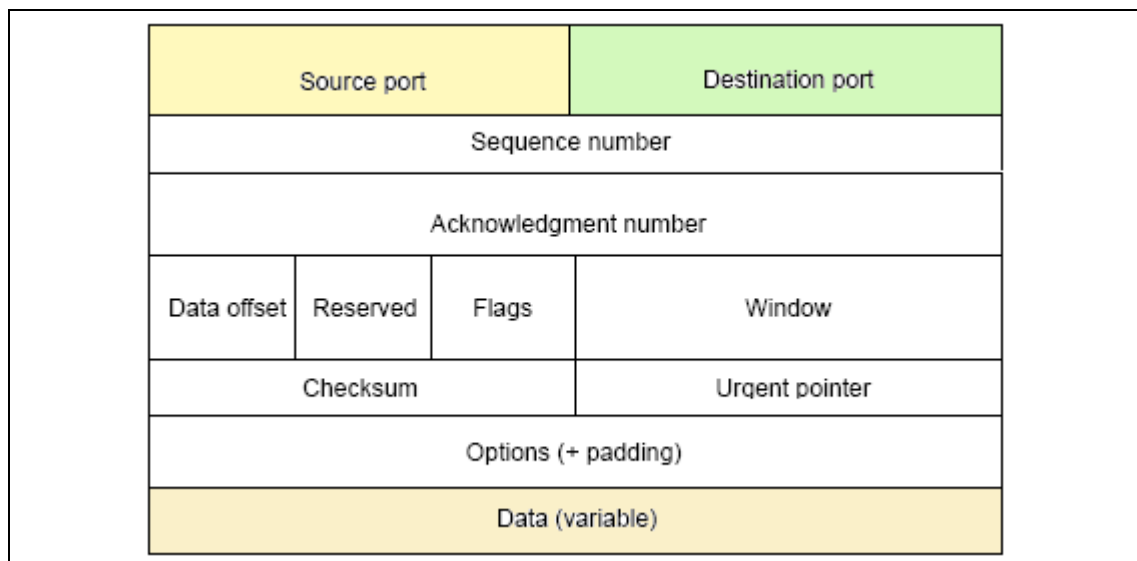


Figure 12-38. Details of the TCP packet frame (Segment)

12-6.3. **User Datagram Protocol (UDP)**

The User Datagram Protocol (UDP) is a connectionless transport-layer protocol (Layer 4) that belongs to the Internet protocol family. UDP is basically an interface between IP and upper-layer processes. UDP protocol ports distinguish multiple applications running on a single device from one another. Unlike the TCP, UDP adds no reliability, flow-control, or error-recovery functions to IP. Because of UDP's simplicity, UDP headers contain fewer bytes and consume less network overhead than TCP. UDP is the transport protocol for several well-known application-layer protocols, including Network File System (NFS), Simple Network Management Protocol (SNMP), and Domain Name System (DNS). The UDP packet format contains four fields, as shown in figure 12-40. These include source and destination ports, length, and checksum fields.

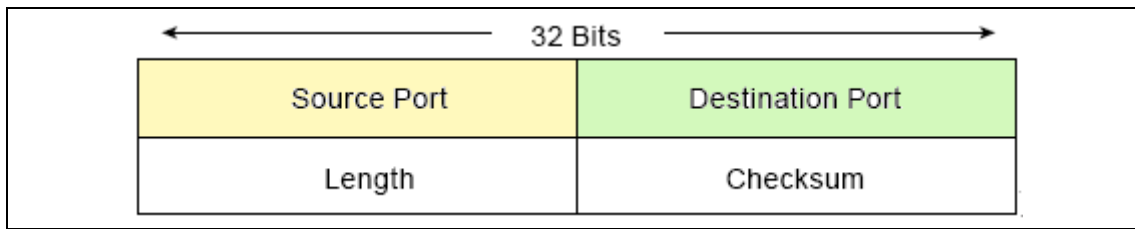


Figure 12-39. Illustration of the UDP packet frame

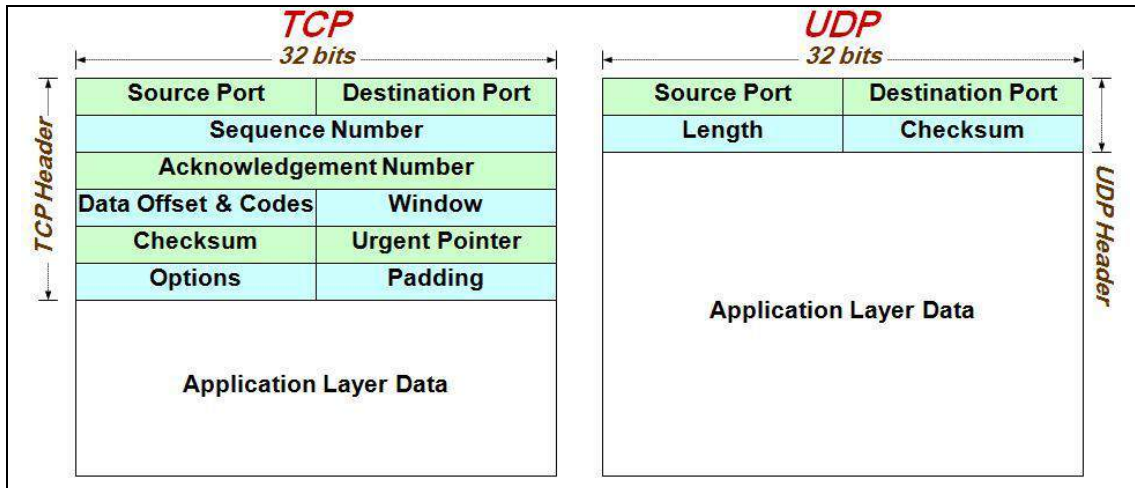


Figure 12-40. comparison between TCP and UDP frames

Example 12-1.

An end system sends 50 packets per second using the User Datagram Protocol (UDP) over a full duplex 100 Mbps Ethernet LAN connection. Each packet consists 1500B of Ethernet frame payload data. What is the throughput, when measured at the UDP layer?

Solution:

Frame Size = 1500B

Packet has the following headers:

IP header (20B)

UDP header (8B)

Total header in each packet = 28B

Total UDP payload data is therefore $1500 - 28 = 1472$ B.

Total bits sent per second = $1472 \times 8 \times 50 = 588800$ bps or 588 kbps.

Example 12-2.

TCP session sends 10 packets per second over an Ethernet Local Area Network (LAN). Each packet has a total size of 1480 B (excluding the preamble and cyclic redundancy check (CRC)). Calculate the size of the headers, and hence the TCP payload data. What therefore is the TCP throughput of the session?

Solution:

First we determine the protocol headers which contribute to the PDU size:

MAC Header (14 bytes) + IP Header (20 bytes) + TCP(20 bytes) + TCP Payload (? bytes)

Next determine the size of the payload:

Payload = $1480 - (14+20+20) = 1426$ B

Throughput = number of useful (data) bits transferred by a layer using the services of the layer below. = $1426 \times 8 \times 10 = 114$ kbps

12-6.4. The TCP/IP Suite Model

The TCP/IP suite of protocols is illustrated in the following figure. The Internet Protocol (IP), defined by IETF RFC791, is the routing layer datagram service of the TCP/IP suite. All other protocols within the TCP/IP suite, except ARP and RARP, use IP to route frames from host to host.

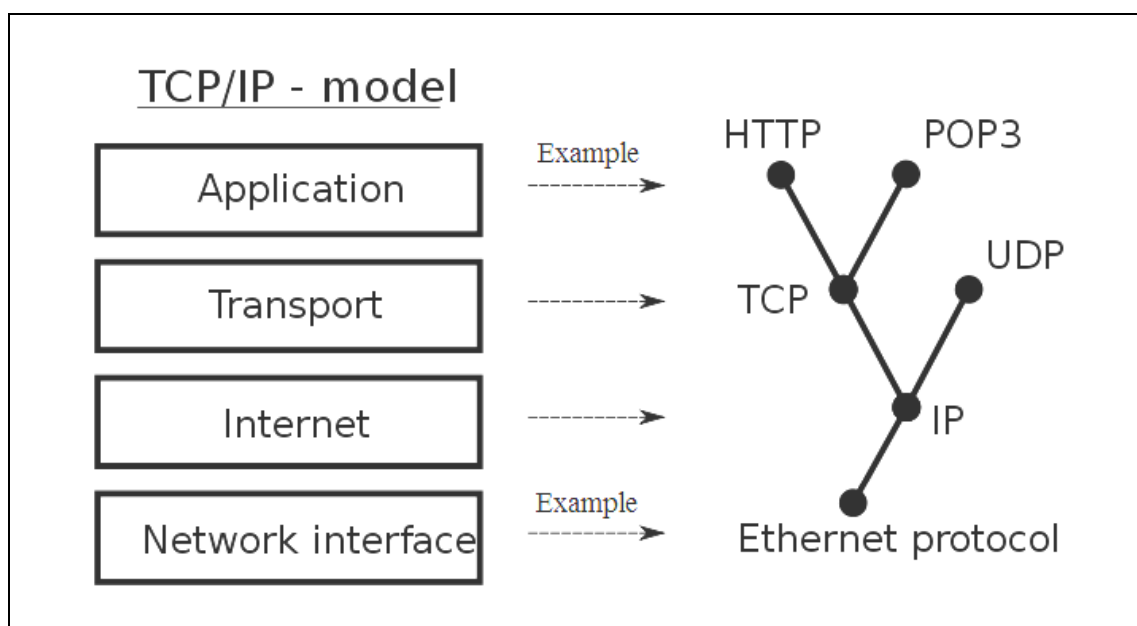


Figure 12-41. Illustration of the TCP/IP model

12-6.5. IPv6 and IPTV

The explosive growth of the Internet has led to the exhaustion of IPv4 addressing scheme. A new protocol version, **IPv6**, has been recently developed to provide vastly larger addressing capabilities (128 bit) and more efficient routing of data traffic. IPv6 is currently in commercial deployment phase around the world.

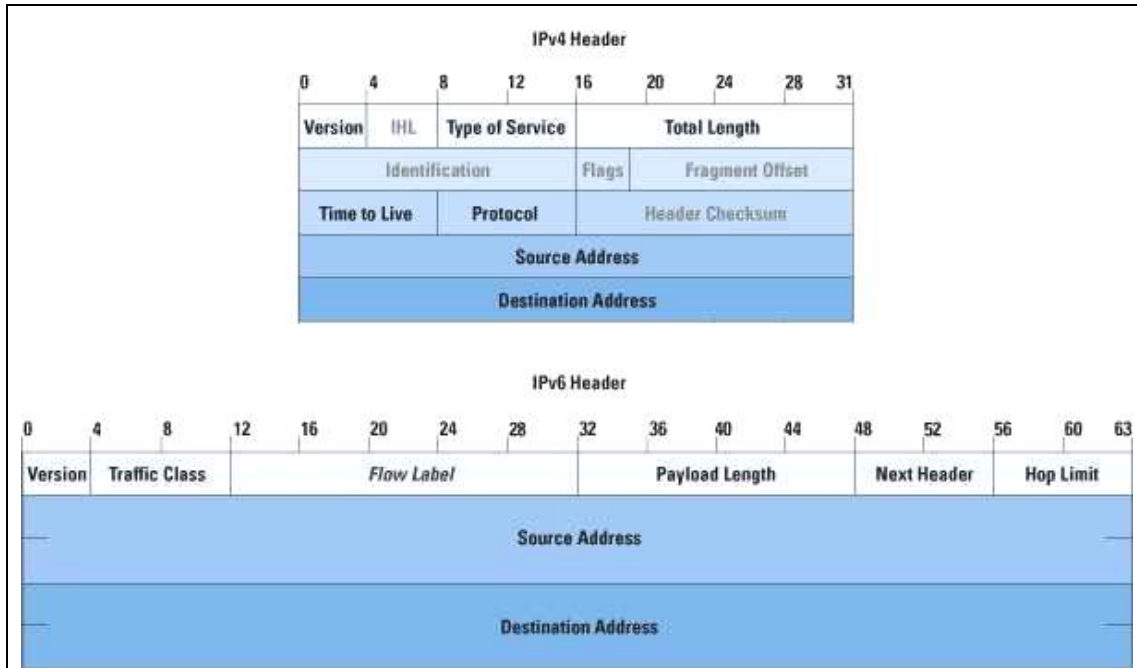


Figure 12-42. IPv4 versus IPv6

IPTV is a planned low-cost broadband replacement of the current IP with massive implications for business and home applications. The new transport and service infrastructure can deliver digital television, data, and voice signals, along with connectivity services to consumers, while enabling on-demand exchanges between content creators and consumers. Virtually any broadband transport media, such as cable, DSL, fiber and wireless transport can convey the modern IP-based rich media signals, to any consumer equipped with the proprietary media gateway and IP. As shown in figure 12-43, Web TV can also be delivered to any IP device with sufficiently managed bandwidth terminating on the media center.

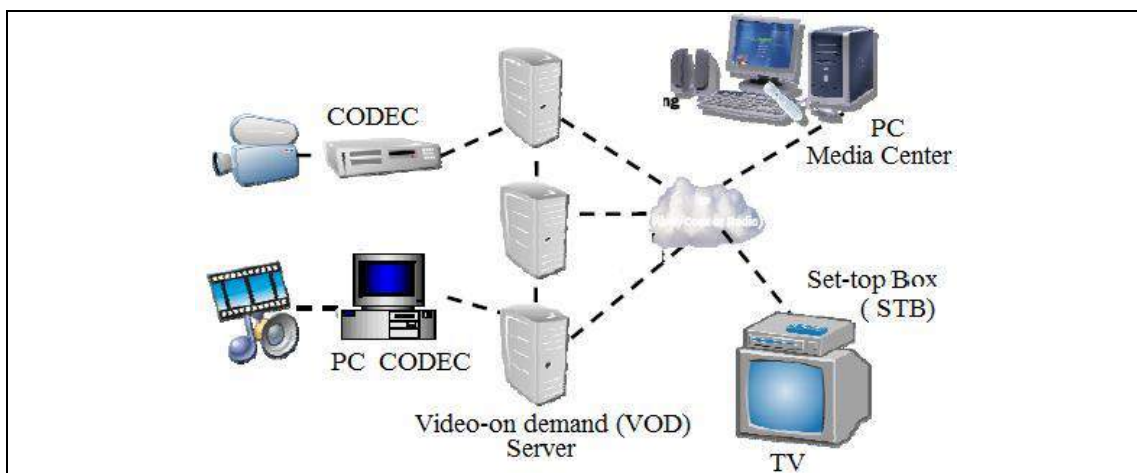


Figure 12-43 IPTV architecture

12-7. Other WAN Protocols

The WAN access encompasses both the Physical (Layer 1) and Data Link (Layer 2) layers of the OSI reference model. The physical layer describes things like mechanical, electrical, functional and operation connections to the service provider. The data link layer defines how the data is **encapsulated**. Different WAN technologies utilizes at this layer are HDLC, PPP, Frame Relay, ISDN and ATM.

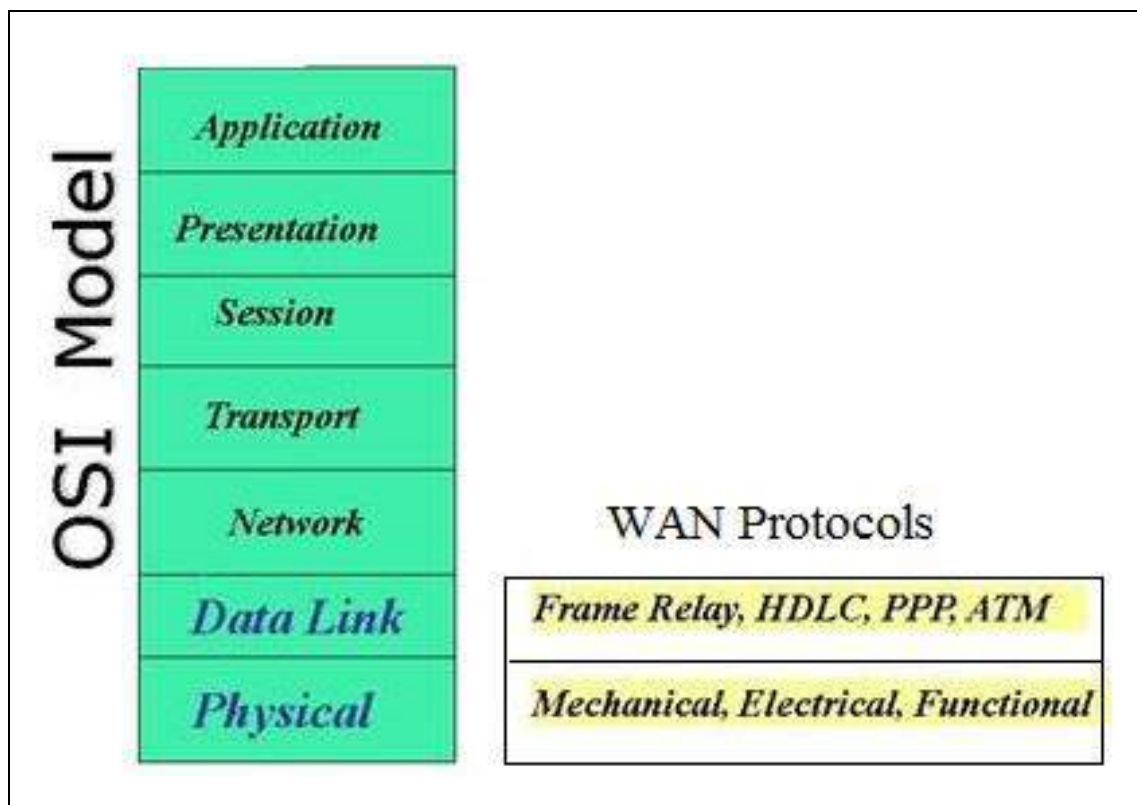


Fig. 12-44. WAN layered structure and protocols

Data from the Network layer (Layer 3) is passed to the Data Link layer for delivery on a physical link, which is normally point-to-point on a WAN connection. The Data Link layer builds a frame around the Network layer data so that the necessary checks and controls can be applied. Each WAN connection type uses a Layer2 protocol to encapsulate a packet while it is crossing the WAN link.

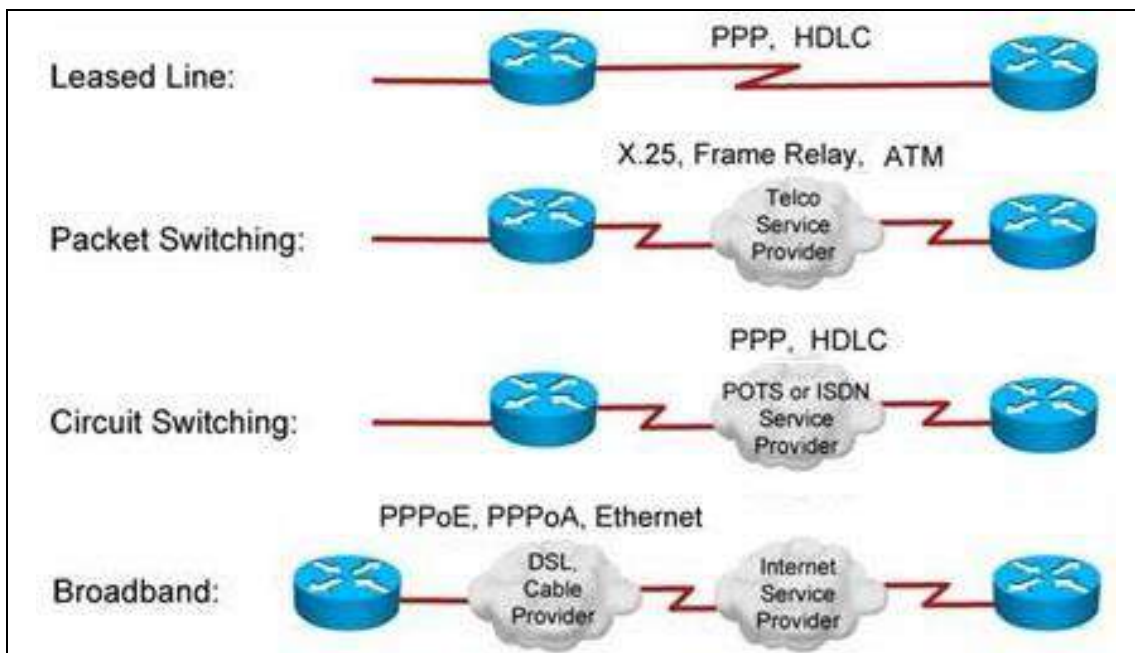


Figure 12-45. WAN encapsulation protocols.

12-7.1. X.25 Standard of Packet Switching

The oldest standard for the format of a packet on a packet switched network is the CCITT **X.25** standard. You will come across this standard regularly when reading material related to Packet Switched Wide Area Networks. X.25 defines the rules for connecting terminals to a packet switched exchange system, the format of the network packet header (containing addressing information) and the **data section** of the packet. The data-link layer of the OSI model treats the total X.25 packet as the data segment for its frame format. For example, the X.25 packet can form the data segment for the HDLC data link frame. This is shown in figure 12-46.

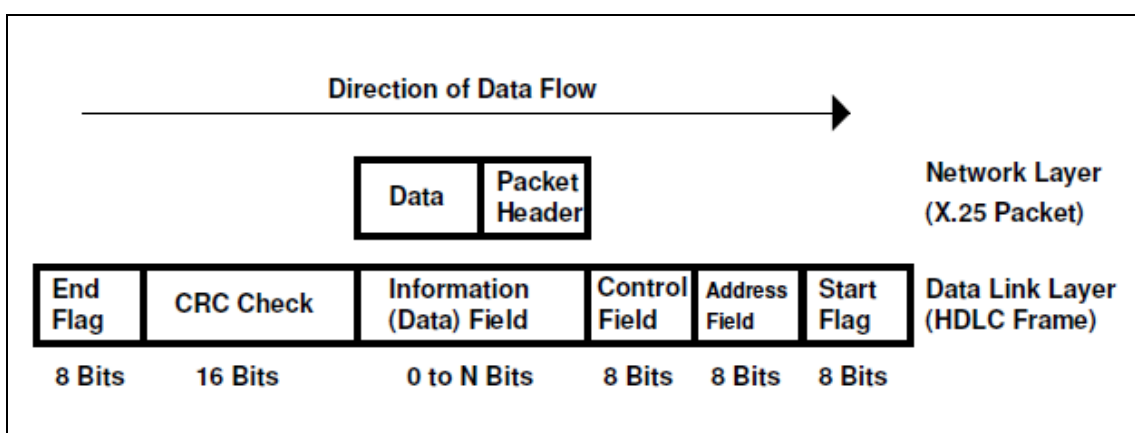


Figure 12-46. Interaction of X25 packets and Data Link Layers frames

12-7.2. Frame Relay Networks

The Frame Relay is a Layer2 WAN protocol. Like X.25, it uses a **packet-switching** technology, but it is more efficient than X.25, in terms of speed and cost. You can obtain a frame-relay circuit by ordering a T1 from the service carrier (PSTN). There are significant differences that make Frame Relay a faster, more efficient form of networking. A Frame Relay network doesn't perform error detection, which results in a considerably smaller amount of overhead and faster processing than X.25. Frame Relay is also protocol independent—it accepts data from many different protocols (including IP, SNA, and all LAN protocols). This data is encapsulated by the Frame Relay equipment, not the network.

The intelligent network devices connected to a Frame Relay network are responsible for the error correction and frame formatting. Processing time is minimized, so the transmission of data is much faster and more efficient. In order to make use of this technology, you need to special bridges, routers, or FRADs (Frame Relay Access Devices). These devices aggregate and convert data into Frame Relay packets.

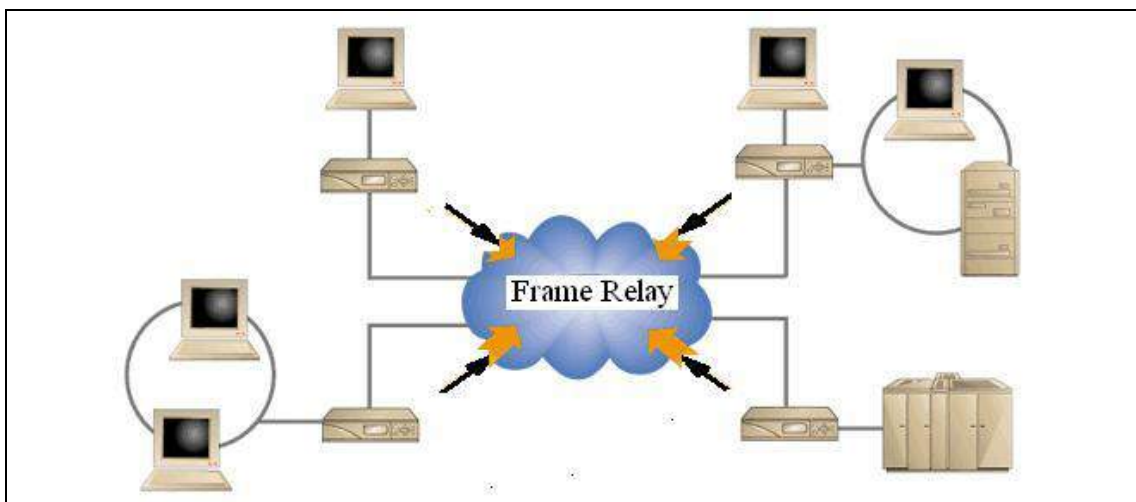


Figure 12-47. Illustration of the frame relay network

As shown in the following figure, the frame relay frame has a variable length data fields. Variable-length packets are used for more efficient and flexible data transfers

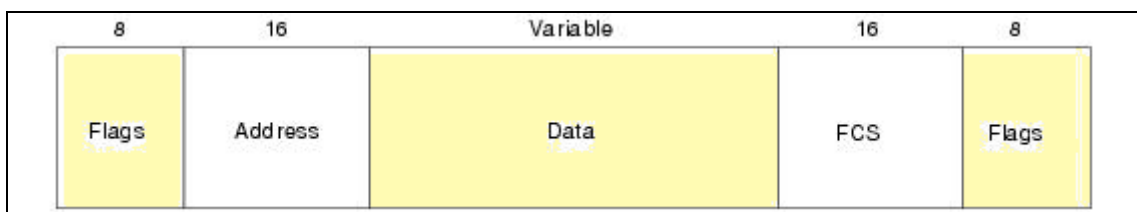


Figure 12-48. Frames of the frame relay network

12-7.3. Integrated Services Digital Network (ISDN)

ISDN is a Circuit Switched technology that was designed to run over existing telephone networks. It is a fully digital end-to-end technology. It consists of a number of protocols for transferring data, voice and video over the traditional telephone system. ISDN has the following features:

- Faster data transmission compared with analog modem connection.
- Perfect for establishing a backup connection to leased line connections.

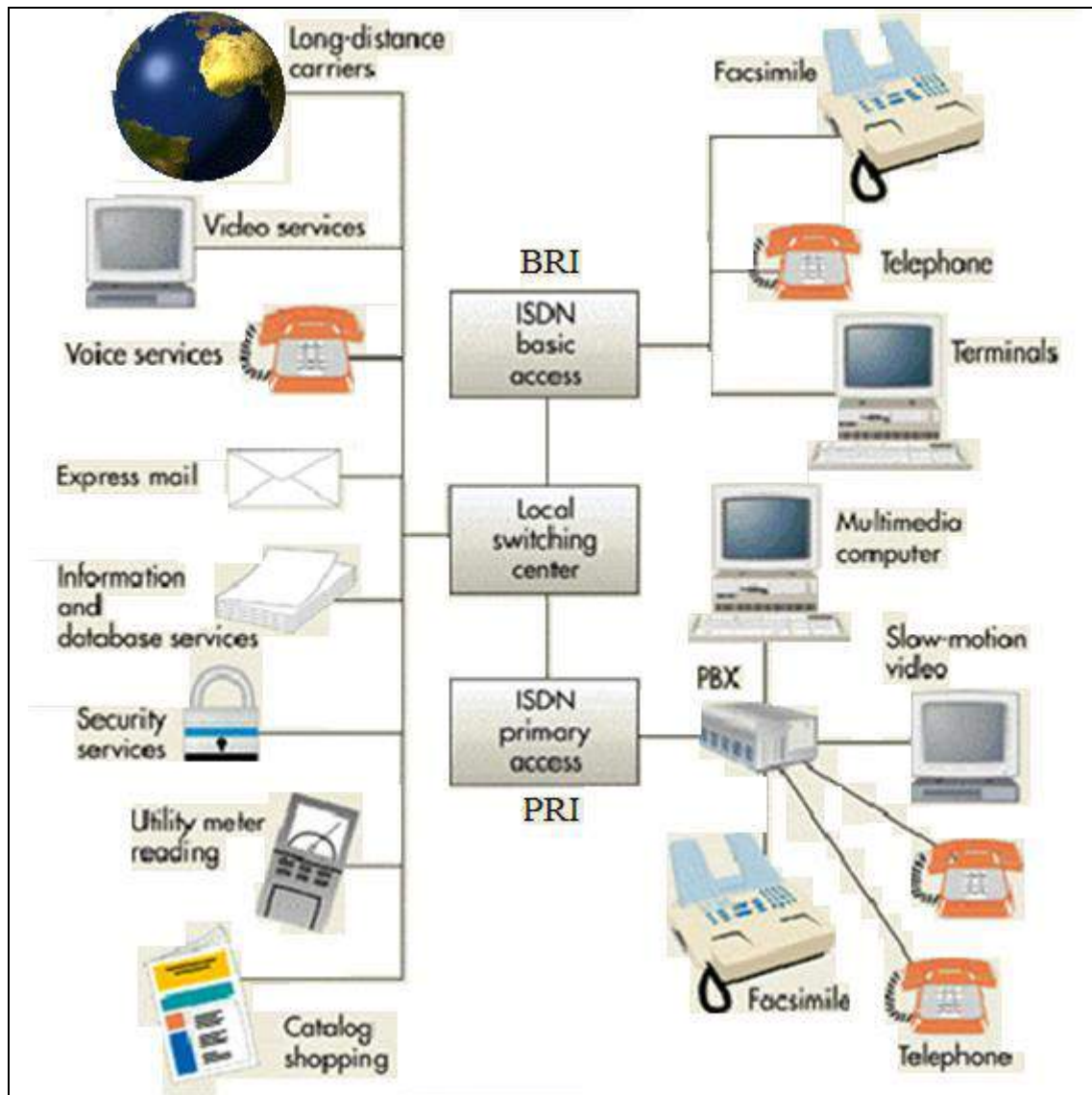


Figure 12-49. Illustration of the frame relay network

As shown in the above figure, the ISDN comes with two flavors:

- 1- ISDN Basic Rate Interface (**BRI**) service also known as 2B+D consists of two data channels (B channels) that operate at 64 Kbps and a single signaling channel (D channel) that operates at 16kbps.

- 2- ISDN Primary Rate Interface (**PRI**) also known as 23B+D in North America and Japan and 30B+D in Europe. In the case of 23B+D, it consists of 23 data channels operating at 64kbps each and one signaling channel operating at 64kbps as well.

12-7.4. Cell Relay Networks

The Cell Relay is an approach for data networks, where the cell is the data unit. In order to support voice, the data units or cells must be small to be processed quickly and sent with minimal delay. This is not true for large data units. Voice requires small data units, but multimedia favors large data units. In a cell relay network, such as **Frame Relay** and **ATM**, the facility is used when needed. Whenever there is information to be transmitted, the switch simply sends the data units. There is no need to negotiate for a connection (like circuit switching), there is no need for a channel to be allocated (no channels in ATM), and as long as there is enough bandwidth, there can be unlimited transmissions.

12-7.5. Asynchronous Transfer Mode (ATM)

The Asynchronous Transfer Mode (**ATM**) is a high performance cell-oriented packet switching and multiplexing technology. ATM combines voice and data communication using short packets (called cells). An ATM cell consists of a 5-byte header and a 48-byte payload., as shown in figure 12-50.

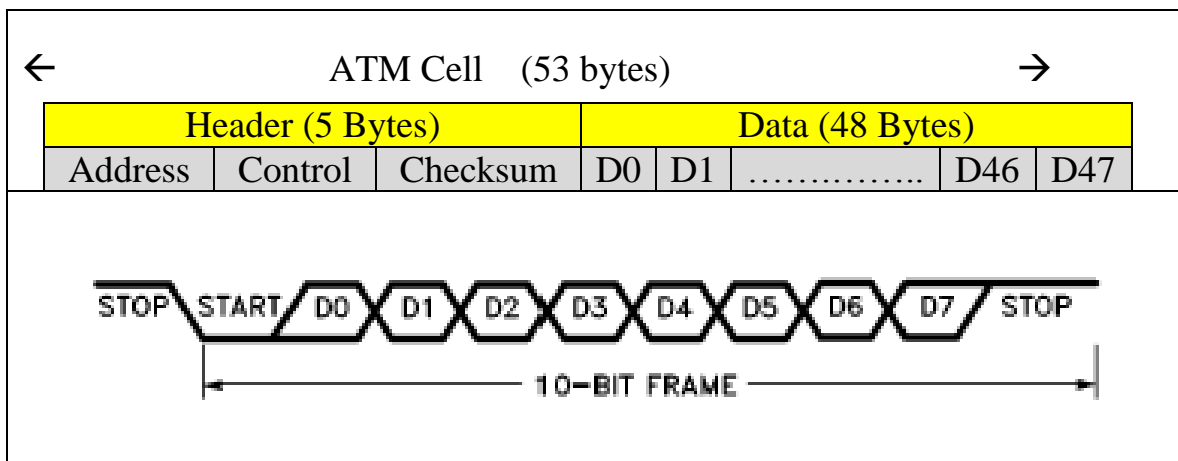


Fig. 12-50. ATM data packets (cells) as compared to conventional RS 232 frames.

In fact, ATM defines two different cell formats: UNI (User-Network Interface) and NNI (Network-Network Interface). Most ATM links use UNI cell format. The detailed structure of these type is indicated in the following figure

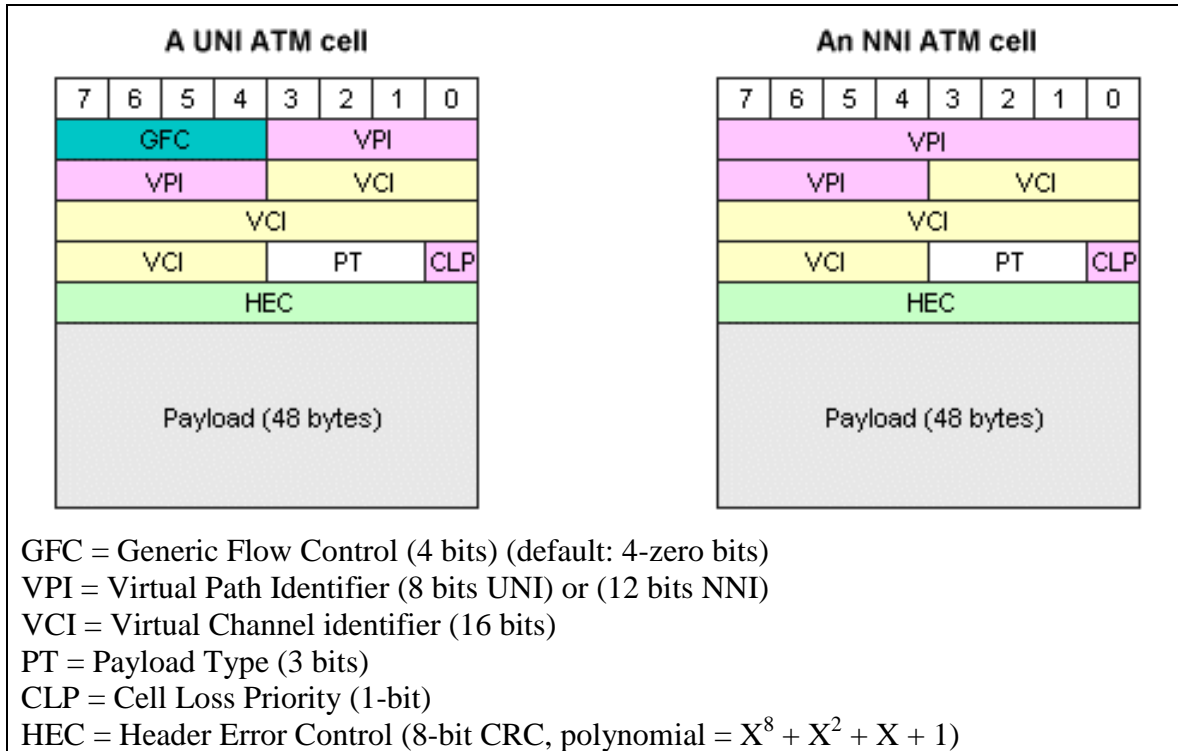


Fig. 12-51. Details of ATM unit cells.

The reference model for ATM almost maps to the three lowest layers of the OSI reference model: network layer, data link layer, and physical layer. ATM defines three layers:

- ATM adaptation layer (AAL)
- ATM layer, roughly corresponding to the OSI data link layer
- Physical layer, equivalent to the OSI physical layer

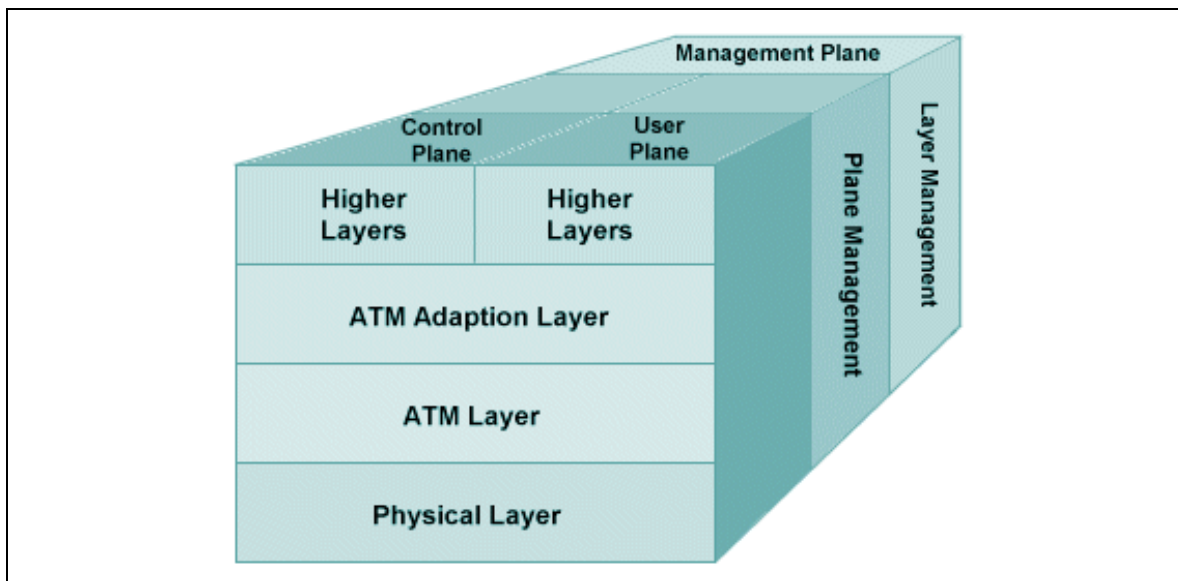


Fig. 12-52 ATM OSI model

In summary, ATM is a connection-oriented, unreliable, virtual circuit packet switching technology. ATM is normally utilized by Internet service providers on their private long-distance networks. ATM became popular with telephone companies and many computer makers in the 1990s. However, by the end of decade, the better price/ performance IP-based products were competing with ATM technology. Due to its complexity, and the advances made in competing technologies, ATM has not lived up to the original expectations, and has not gained widespread acceptance, in particular for LAN applications. Real-time voice and video applications are now being successfully carried over IP networks,

12-7.6. Voice-on-Internet Protocol (VoIP). VoATM and VoFR

Voice over Internet Protocol (VoIP) is a methodology for the delivery of voice /data communications over Internet Protocol (IP) networks, such as the Internet. Also, **VoATM** stands for voice over ATM and **VoFR** stands for voice on Frame relay. The benefit is that, VoIP can turn a standard Internet connection into a way to place **free phone calls**. The practical upshot of this is that by using some of the free VoIP software that is available to make Internet phone calls, you're bypassing the phone company (and its charges) entirely.

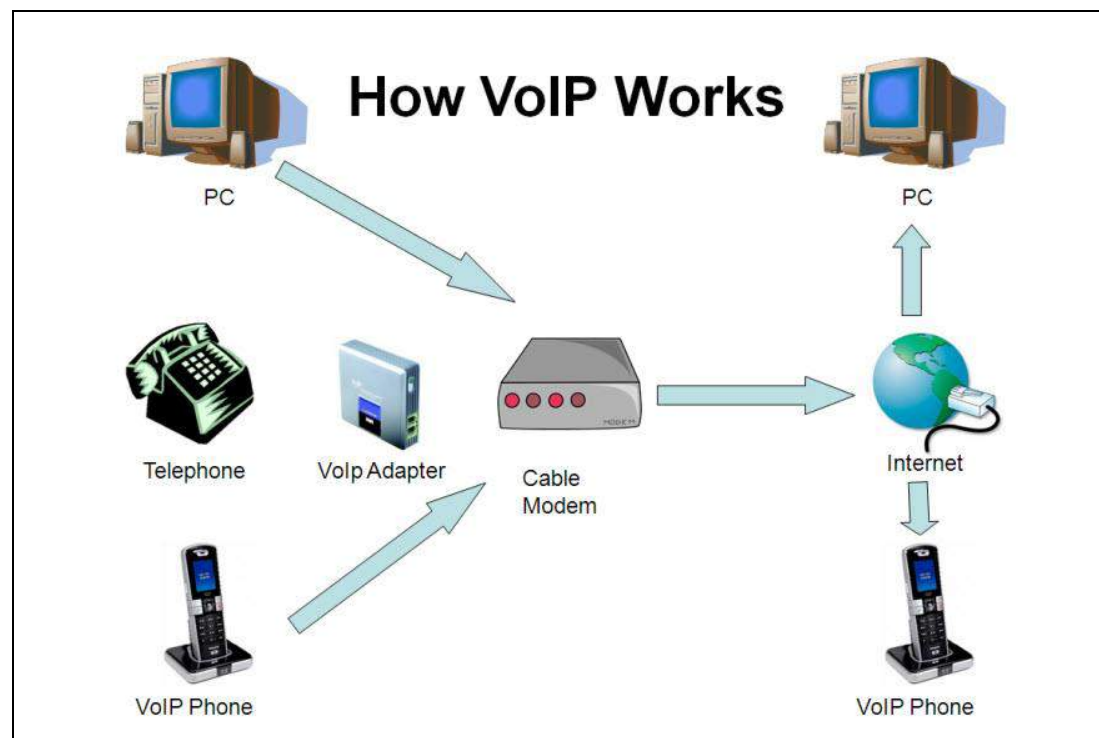


Figure 12-53. Implementation methods of VoIP

VoIP technology uses the Internet packet-switching capabilities to provide phone service. A **packet-switched** phone network is the alternative to circuit switched PSTN. While circuit switching keeps the connection open and constant, packet switching opens a brief connection -- just long enough to send a packet. VoIP has several advantages over circuit switching. For example, packet switching allows several telephone calls to occupy the amount of space occupied by only one in a circuit-switched network. Using PSTN, a 10-min phone call consumes 10 full minutes of transmission time at a cost of 128 Kbps. With VoIP, that same call may have occupied only 3.5 minutes of transmission time at a cost of 64 Kbps.

There are several approaches to implement VoIP. Each makes use of a variety of protocols to handle signaling and data transfer

ATA -- The simplest and most common way is through the use of a device called an ATA (**analog telephone adaptor**). The ATA allows you to connect a standard phone to your computer or your Internet connection for use with VoIP. The ATA is an analog-to-digital converter. It takes the analog signal from your traditional phone and converts it into digital data for transmission over the Internet. Providers like AT&T are bundling ATAs free with their service. You simply plug the phone cable into the ATA, and you're ready to make VoIP calls. Some ATAs may ship with additional software.

Computer-to-computer -- This is certainly the easiest way to use VoIP. You don't even have to pay for long-distance calls. There are several companies offering free or very low-cost software (like **Skype**) that you can use for this type of VoIP. All you need is the software, a microphone, speakers, a sound card and an Internet connection, preferably a fast one like you would get through a cable or DSL modem. Except for your normal monthly ISP fee, there is usually no charge for computer-to-computer calls, no matter the distance.

IP Phones -- These specialized phones look just like normal phones with a handset, cradle and buttons. But instead of having the standard RJ-11 phone connectors, IP phones have an RJ-45 Ethernet connector. IP phones connect directly to your router and have all the hardware and software necessary right onboard to handle the IP call. Wi-Fi phones allow subscribing callers to make VoIP calls from any Wi-Fi hot spot.

In recent years many VoIP systems have become as easy to use and as convenient as usual telephones. Simple, inexpensive VoIP modems are

now available that eliminate the need for a PC. Thus, VoIP is maturing into a viable alternative to traditional telephones. Interoperability between different providers has improved and the ability to call or receive a call from a traditional telephone is available. VoIP has also become increasingly popular within the gaming world, as a form of communication between players. Popular gaming VoIP clients, which offer VoIP chat, include Play station3 and Xbox.

In addition to the VoIP terminal devices, we mentioned so far, VoIP networks usually incorporate gateways, gatekeepers and multipoint control units (MCU). The VoIP is supported by a variety of protocols. The VoIP protocols include the ITU- **H.323** and Session Initiation Protocol (**SIP**), from 3Com

12-8. Integration of Voice and Data Networks

In the last few years, data networks have been growing at a much faster rate than voice networks, mainly due to the growth of the Internet. Soon the amount of data traffic will exceed that of voice traffic. As a result of this trend, more and more voice is being sent over data networks (Voice over Frame Relay, Voice over IP and Voice over ATM) than data is being sent over voice networks (via V.34 and V.90 modems).

When Frame Relay was introduced in the early 1990s, the data technology was not originally designed to carry voice. Despite valid reservations about the reliability of voice over frames, the promise of "free voice" eventually proved too alluring. Soon users were experimenting with transporting voice over their Frame Relay devices while equipment vendors worked overtime to make the promise of quality voice over Frame Relay (VoFR) a reality.

As the public Internet exploded in the mid-1990s and users began implementing IP-based networks, the call for voice over IP (VoIP) grew louder. Here, too, equipment manufacturers are developing products to enable inexpensive, universal voice over data networks. Carriers, however, were caught in a dilemma. Could they afford to cannibalize their highly profitable public switched telephone network? Could they not afford to capitalize on the demand for digital voice? The drama is just unfolding. Although significant progress has been made in engineering packet networks (Frame Relay, IP and ATM) to carry voice as well as data, today's market is demanding a true convergence of these technologies into a single and ubiquitous communications service without being limited by the underlying technology. The next challenge, then, is to develop interconnection and interworking standards in order to deliver voice services ubiquitously over Frame Relay, IP and ATM.

12-8.1. Nature of the Data Network and its Implications for Voice

Frame Relay, IP and ATM are known as packet or cell switching technologies. This is in contrast to the public telephone network, which is a circuit switching technology, designed to carry voice transmissions. Frame Relay and IP insert data into variable-sized frames or packets. ATM chops data into small cells, which facilitates fast switching of data through the network. The packet switching and cell switching networks perform statistical multiplexing. That is, they dynamically allocate bandwidth to various links based on their transmission activity. Since bandwidth is not reserved for any specific path, the available bandwidth is allotted according to network needs at any particular time.

Compare this to the traditional voice (or circuit switching) network, in which a path is dedicated to the transmission for the duration of the call, which is sent in a continuous bit stream. The line is monopolized by a call until it is terminated, even when the caller is put on hold and during periods of silence. Although this guarantees reliable and immediate transmission of voice, it results in very inefficient use of bandwidth. A line that is dedicated to the telephone cannot be utilized by other data even when there are no voice transmissions. Originally designed to handle bursty data traffic, packet switching networks (except for ATM) are inherently less efficient than the circuit switching network in dealing with voice. To achieve good voice quality, the delay of voice packets across the network must be minimal and fixed. Due to the shared nature of the packet/cell switching network, it might take time for transmissions to travel across the network. A transmission can be delayed because of network congestion. For example, it might "get stuck" behind a long data transmission that delays other packets. Network congestion can also result in dropped packets, which also detrimentally affects the integrity of voice transmissions.

12-8.2. Voice-Enabling the Data Network

Unlike most data applications, voice is very sensitive to delay. Good voice quality provides a faithful recreation of the conversation, with the same tone, inflection, pauses and intonation used by the speakers. Long and variable delays between packets result in unnatural speech and interfere with the conversation. Dropped packets result in clipped speech and poor voice quality. Fax transmissions are even more sensitive to the quality of the transmission and are less tolerant of dropped packets than voice.

One way to deal with the problem of delay and congestion is to add bandwidth to the network at critical junctures. Although this is feasible in the backbone, it is a costly and ineffective solution in the access arena, defeating the "bandwidth sharing" benefits of packet networks. The best solution is to implement mechanisms at the customer premises, access node and backbone which manage congestion and delay - without increasing bandwidth - such as setting priorities for different types of traffic. Therefore, smart access equipment was developed, that could implement procedures to reduce network congestion and the delay of voice packets without adding bandwidth.

12-9. Summary

A data network is a mechanism by which many computer or devices (referred to as network nodes) can communicate with one another on *any node to any node* basis. In this chapter, we presented the general architecture of Local Area networks (LAN's) and Wide Area Networks (WAN's) and their communication protocols.

Property	LAN	WAN
Protocols	Ethernet, Token Ring, FDDI, etc.	X.25, Frame Relay, ISDN, Leased line etc.
Communication	Shared Media	Point-to-point
Main Features	Offer high speeds over short distances.	Low speeds over longer distances. Attenuation and noise are significant Equipment is expensive.
Usage Range	Within a building, campus, or city (distances ~km),	1. Between cities or large distances 2. (several hundred kms)
Speed	Up to 1 Gbps typical.	Up to several Gbps shared.
Cost	Very low cost per Mbps	High cost per Mbps.

The following lines provide a summarized list of fundamental points related to Local Area Networks:

- i) A Local Area Network is a collection of intelligent devices (nodes) that are all interconnected via a star, bus or ring topology. In theory, any node should be able to communicate with any other node.
- (ii) The majority of commercially available networks are based upon the use of serial communications techniques. Each node must have a conversion circuit to transform data from the parallel form, used on the internal data bus, to the serial form used on the network.
- (iii) Many available networks operate at high bit rates (5–10 Mb/s) and therefore use synchronous serial communications, because of the difficulties involved in using asynchronous communications at these speeds.
- (iv) In some networks all nodes have access to all information. It is therefore important that each node be given an address, that each node is aware of its address and that all messages contain a source and target address.
- (v) Data transfer in networks can either be bit-oriented or character-oriented, depending upon the communications protocol in use.

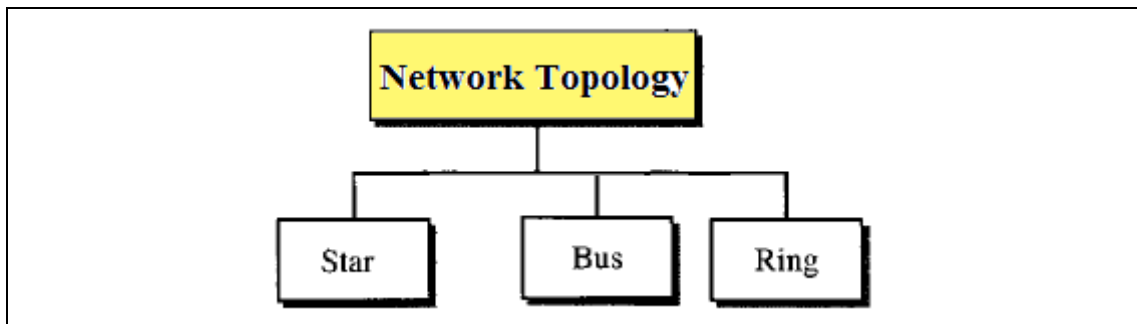
(vi) Networks generally use cyclic redundancy check polynomials for error detection, but may also use Check Sums where data transfer rates are lower and noise immunity is high.

(vii) All data that is transferred on a network must be in packets so that addressing and error detection information can be included with the data.

The fundamental features of a data packet are shown below:

(viii) Within any one communication channel on the network, whenever two or more nodes attempt to use the transmission medium simultaneously, a contention occurs. Contentions can lead to data corruption and therefore a number of contention schemes have been put into place. The two most common are the CSMA/CD and the Token Passing schemes.

(ix) The three basic network topologies are the star, bus and ring.



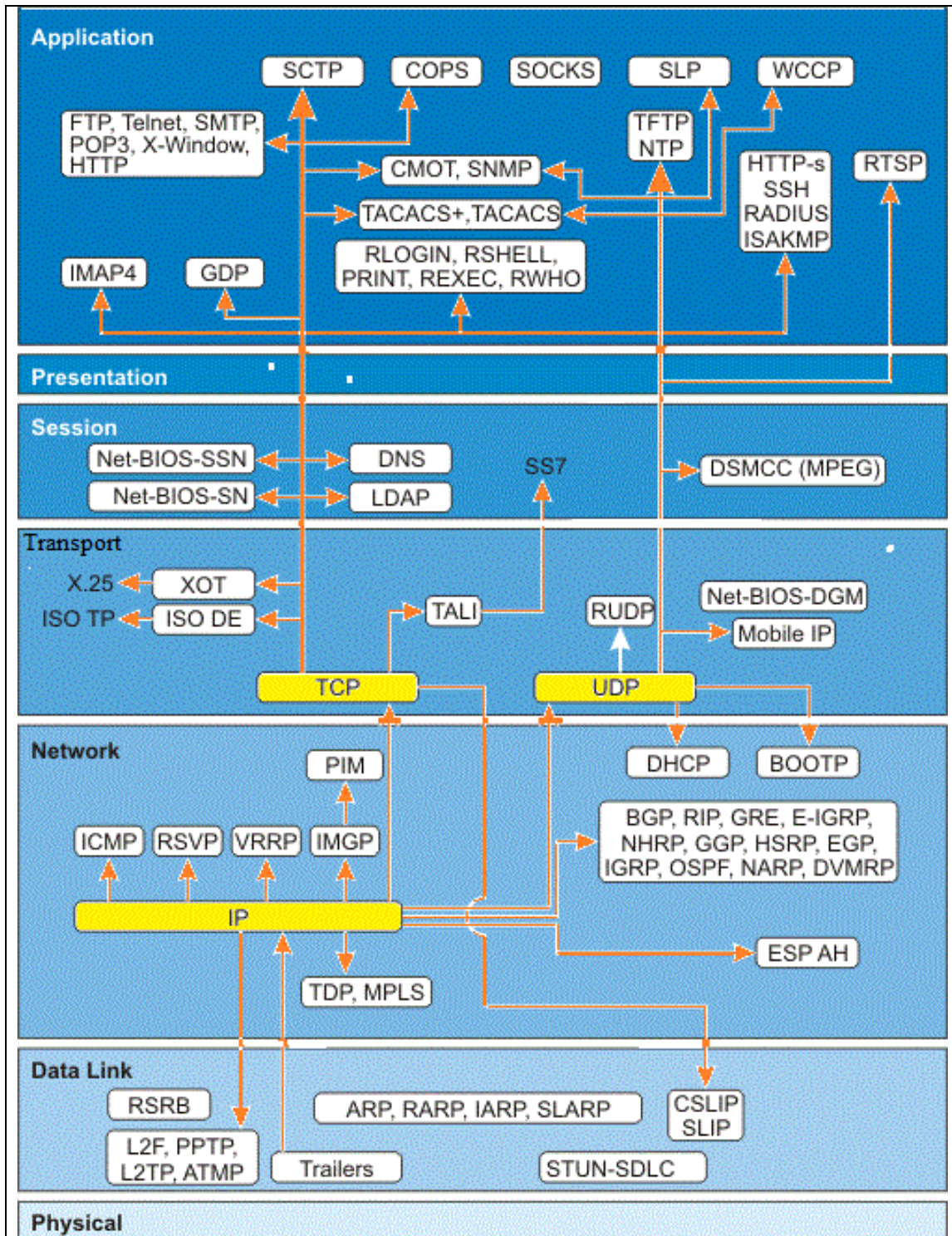
(x) The star network topology is composed of a number of point to point links between peripheral nodes and a central, star node. The intelligent "star node" is used to connect any one peripheral node to any other peripheral node. The star node also resolves contentions for resources.

(xi) The Bus Network consists of a central trunk cable, which is generally a two conductor (signal + return) or two optic fibre media. Nodes on the network all have access to the same data and must selectively ignore or act upon data packets, depending upon the addressing. Since bus networks do not have an intelligent node to resolve contentions, each node must be capable of handling contention situations.

(xii) The ring network topology is made up of a number of devices that are connected via "point to point links" to form a physical ring. Messages are generally sent around a ring from a source node to a destination node. Messages usually only travel in one direction around the ring. Nodes on the network all have access to the same data and must selectively ignore or act upon data packets, depending upon the addressing. Ring networks do not have an intelligent node to resolve contentions.

(xiii) The International Standards Organization (ISO) has defined a 7 layer model for Open Systems Interconnection (OSI) in data communications.

The following figure depicts the details of the internet IP/TCP protocol suite and their relation to the OSI model



Routers and **bridges** link two or more individual **LAN**'s to create an extended-network **LAN** or **WAN**. Technically, a **router** is a Layer 3 gateway, meaning that the router connects different networks, and that the router operates at the Network Layer of the OSI model. Home networks often use an Internet Protocol (**IP**) wired or wireless router, where **IP** is the most common OSI network layer protocol. An **IP** router such as a **DSL MODEM** router joins the home **LAN** to the Internet **WAN**.

A **gateway** is an internetworking system that joins two networks together. A network gateway can be implemented completely in software, completely in hardware, or as a combination of the two. Depending on their implementation, network gateways can operate at any level of the OSI model from application protocols to low-level signaling.

Circuit switching is a method which sets up a limited number of dedicated connections of constant bit rate and constant delay between nodes for exclusive use during the communication session. **Packet switching** is a digital networking method that groups the data to be transmitted into suitably sized blocks, called packets.

In cases where traffic fees are charged, for example in cellular communications, circuit switching is characterized by a fee per unit of connection time, even when no data is transferred, while packet switching is characterized by a fee per unit of information transmitted (packets).

The **X.25 protocol** is a data link protocol, which is part of the OSI protocol suite. It was widely used in switching networks during the 1980s and early 1990s. **Frame relay** is a further development of X.25. The simplicity of Frame Relay made it considerably faster and more cost effective than X.25 packet switching.

Both X.25 and Frame Relay provide connection-oriented packet switching. A major difference between X.25 and Frame Relay packet switching is that X.25 is a reliable protocol, based on node-to-node automatic repeat request, while Frame Relay is a non-reliable protocol, with maximum packet length is 1000 bytes.

The Asynchronous Transfer Mode (**ATM**) is a high performance cell-oriented packet switching and multiplexing technology. **ATM** combines voice and data using short packets (called cells) of 53 bytes

12-10. Problems

12-1) Describe the operation of local area networks (LAN) in 10 points.

12-2) What are the types of switching networks? Give examples of practical switching networks, and show, by neat sketches how devices are interconnected in such networks.

12-3) What are the fundamental features of data packets?

12-4) Describe the HDLC protocol and show its advantages.

12-5) Show, by neat sketches, how data packets are encapsulated at the different layers of an Ethernet network.

12-6) What do you know about the wireless Ethernet protocols? What's the difference between wireless Ethernet and WiFi and WiMAX?.

12-7) Describe TCP/IP Internet protocols and their data frame structures.

12-8) Describe how both voice and video streams can be sent over the Internet protocol (e.g., VoIP).

12-9) Choose the right answer for the following statements:

- WAN circuits may be:
 - i) Packet-switched,
 - ii) Circuit-switching,
 - iii) Point-to-point,
 - iv) All of the above.

- A modem converts :
 - i) Digital signals into analog for transmission over a telephone line.
 - ii) Analog signals into digital for reception over a telephone line
 - iii) Both of the above

- What is the correct name for the standard for sending voice, audio and video using IP on the public Internet and within an intranet?
 - i) ITU-T H.323
 - ii) ITU-T H.562
 - iii) ITU-T H.232
 - iv) ITU-T H.453

12-10) True or false: VoIP saves money and it is less expensive in terms of network infrastructure?

12-11. References

- [1] M. **Castells**, Rise of the Network Society. 3 Vols. Cambridge, MA: Blackwell Publishers, 1996.
- [2] Janet **Abbate**, *Inventing the Internet*, MIT Press, 2000.
- [3] S. **Haykin**, Communication Systems, 4th Ed., Wiley, 2001.
- [4] William **Stallings**, Data and Computer Communications, 7th ed., Prentice Hall, 2004.
- [5] Marguerite **Reardon**, "Optical networking: The next generation", *CNET News*, October 11, 2004
- [6] **CISCO** Network Handbook, CISCO Inc., 2006.
- [7] **Bertsekas** and **Gallager**, Data Networks, 2nd Ed., 2006

Chapter
13

Miscellaneous Communication Systems

Contents

- 13-1. Radio Relay Systems**
 - 13-1.1. Line of Sight (**LOS**) of a Radio Link
 - 13-1.2. Non-Line of Sight Radio Links
 - 13-1.3. Other Media Link
- 13-2. Satellite Communications & VSAT**
 - 13-2.1. Types of Satellites
 - 13-2.2. Satellite Link Budget
 - 13-2.3. Satellite Downlink Budget Analysis
- 13-3. Cellophane Communications**
 - 13-3.1. Cell Phone Network Components
 - 13-3.2. Cell Phone-Satellite Link Budget
- 13-4. Wireless Computer & Internet Links**
 - 13-4.1. Wi-Fi and WiMax
 - 13-4.2. Satellite Internet Access
- 13-5. Public Switching Telephone Networks (PSTN)**
- 13-6. Digital Subscriber Lines (DSL)**
 - 13-6.1. Asymmetric Digital Subscriber Line (ADSL)
 - 13-6.2. ADSL Technology
 - 13-6.3. ADSL Wiring and Filters
 - 13-6.4. xDSL Standards
- 13-7. Digital Video Broadcasting (DVB)**
 - 13-7.1.. Digital TV Standards
 - 13-7.2.. Digital Video Broadcast-Terrestrial (DVB-T)
 - 13-7.3.. DVB-T Single Frequency Network (SFN)
 - 13-7.4.. DVB-Handheld (DVB-H)
 - 13-7.5.. DVB-Satellite to Handheld (DVB-SH)

Contents of Chapter 13 (Cont.)**13-8. Optical Communications Links**

13-8.1. Optical Link Budget

13-8.2. Practical Optical Links

13-9. Optical Networks

13-9.1. Optical Network Architecture

13-9.2. Fiber Distributed Data Interface (FDDI)

13-9.3. Fiber to the Premises (FTTP)

13-9.4. Fiber to the Home (FTTH)

13-10. Summary**13-11. Problems****13-12. Bibliography**

Chapter 13

Miscellaneous Communication Systems

13-1. Radio Relay Links

In this chapter we recapitulate the operation of some communication links, their design principles and link budgets. The process of communication system design is iterative and goes through many phases, as shown in figure 13-1. The figure depicts the design cycle of communication systems, with emphasis on radio links.

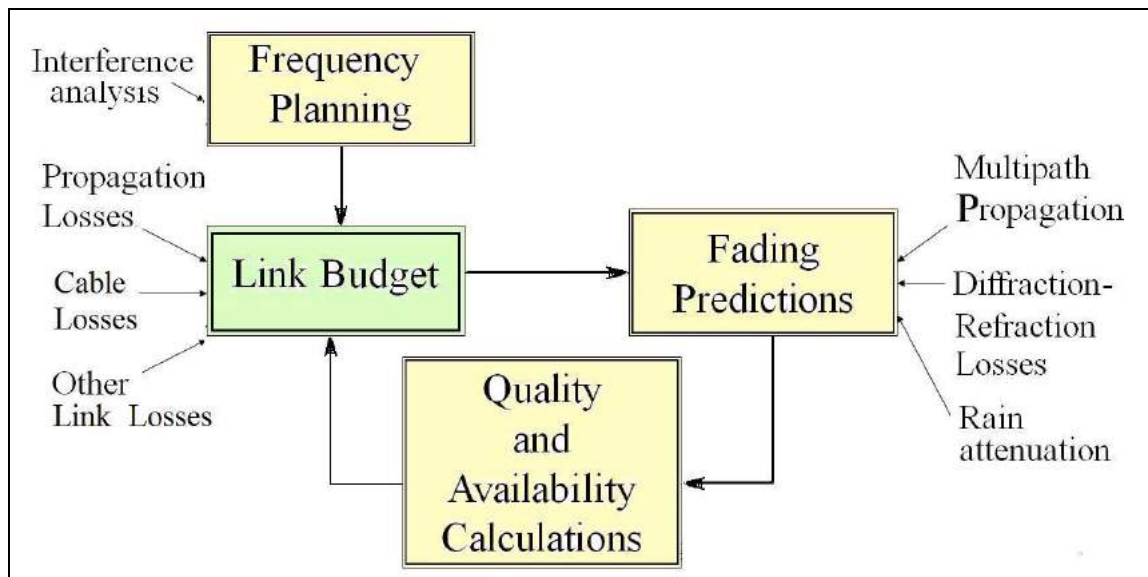


Fig. 13-1. Basic components of a communication link design cycle,

Radio relay is a technology for transmitting digital and analog signals, such as long-distance telephone calls and the relay of television programs to transmitters, between two locations on a line of sight radio path. In microwave radio relay, radio waves are transmitted between the two locations with directional antennas, forming a fixed radio connection between the two points. Because of the high frequencies used, a quasi-optical line of sight (**LOS**) between the stations is generally required. Additionally, in order to form the line of sight connection between the two stations, the first Fresnel zone must be free from obstacles so the radio waves can propagate across a nearly uninterrupted path.

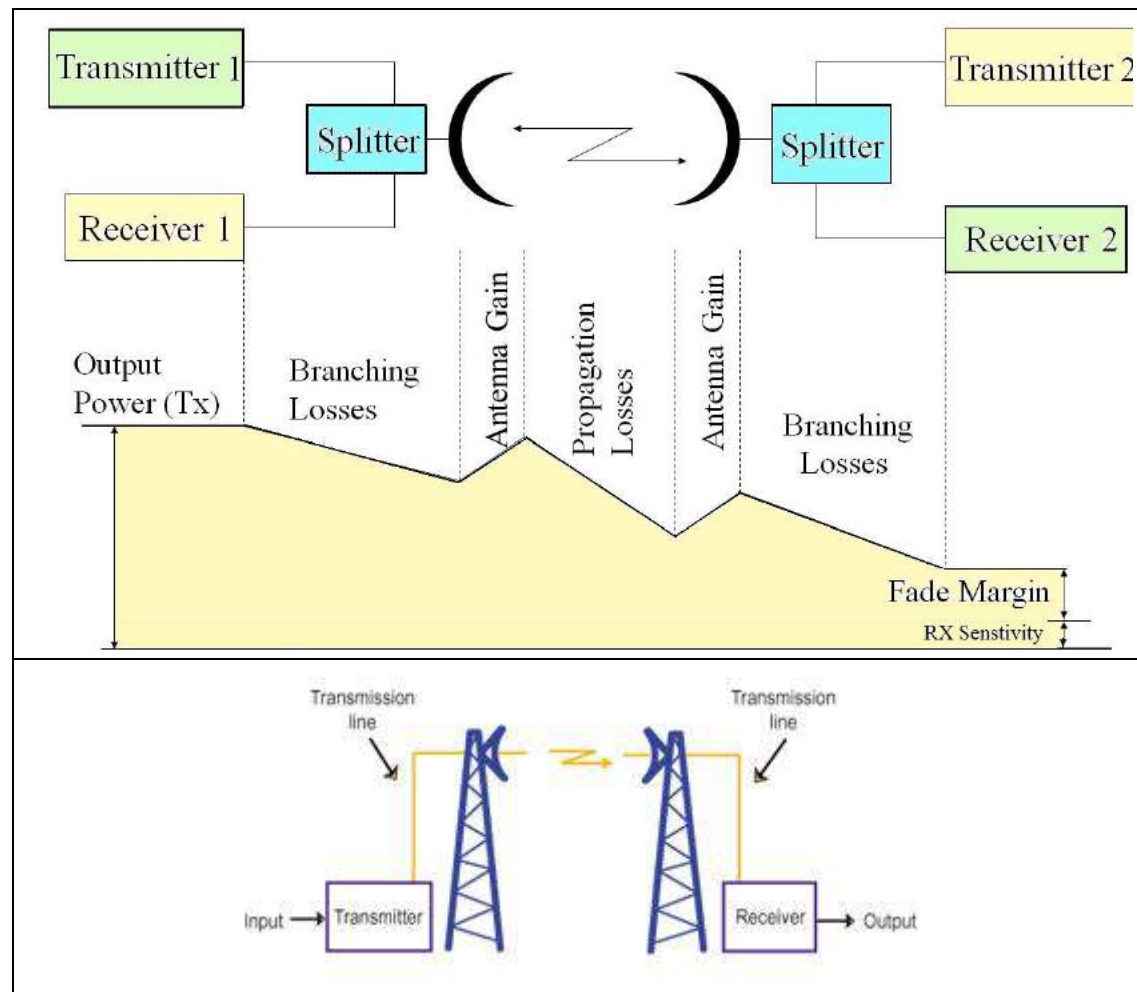


Fig. 13-2. Basic components of a relay radio link. Both full duplex (up) and simplex (down) links are shown.

13-1.1. Line of Sight (LOS) of a Radio Link

Line of sight (LOS) is commonly used to refer to telecommunication links that rely on a line of sight between the transmitting antenna and the receiving antenna. Such capability is necessary for high frequency microwave links that offer relatively high bandwidth communication circuits. Typical operating frequencies are in the gigahertz frequency range where the radio path is not reflected or refracted to any great extent. Typical transmission path lengths are of the order of 50 km but the height of the antennae and intervening terrain have significant influence.

Many links are now being upgraded to fiber optic cable. However, line of sight may be also necessary for optical transmission systems for short distances, between two high buildings, where a cable link might be very long. Figure 13-3 depicts the LOS of microwave link and the lower half of

the so-called **first Fresnel zone**. Fresnel Zones are areas of space around an axis (LOS) where eventual constructive and destructive interference may be created when electromagnetic waves are reflected (multipath) or diffracted as the wave intersects obstacles. Fresnel zones are specified by ordinal numbers that correspond to the number of half wavelength multiples that represent the difference in radio wave propagation path from the direct path. The Fresnel Zone must be clear of all obstructions. Based on this, we can find out what the minimum distance of an obstacle (e.g. a building or a hill) from our LOS should be.

For a line of sight radio system, a link budget equation may look like this

$$RxP = TxP + TxG - TxL - FSL - ML + RxG - RxL \quad (dB)$$

where:

RxP = received power (dBm),

TxP = transmitter output power (dBm) ,

TxG = transmitter antenna gain (dBi),

TxL = transmitter losses (coax, connectors...) (dB),

FSL = free space loss or path loss (dB) ,

ML = miscellaneous losses (fading margin, polarization mismatch,..) (dB)

RxG = receiver antenna gain (dBi), and

RxL = receiver losses (coax, connectors...) (dB).

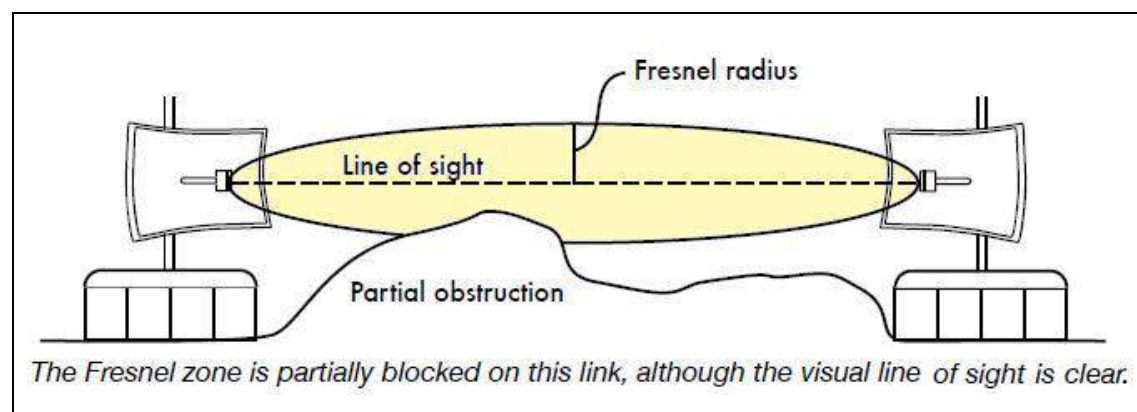


Fig. 13-3. Line of sight (LOS) and Fresnel zone of a radio link

Antenna gain is the measurement of an antennas ability to amplify the incoming microwave signals in a particular direction, compared with the sensitivity of an isotropic antenna in any direction, or a dipole antenna in the equatorial direction. **Free space loss** is the loss in power of an electromagnetic wave that is associated with the phenomenon of beam

divergence and the inverse square law of electromagnetic radiation. Path loss, in communication systems, is the attenuation undergone by an electromagnetic wave in transit between a transmitter and a receiver. **Fading** (or fading channels) are models for the distortion that a carrier-modulated telecommunication signal experiences over certain propagation media.

Line of sight (**LOS**) links have path losses that are the **inverse square** of the **distance**. The free space loss (FSL) can be written, in terms of distance between receiver and transmitter (d) and wavelength (λ) as follows:

$$FSL (dB) = 20 \cdot \log [4\pi d/\lambda] = 32.45 \text{ dB} + 20 \cdot \log[f(\text{MHz})] + 20 \cdot \log[d(\text{km})]$$

When a receive antenna of constant physical area receives a transmission from an isotropic antenna, the receive antenna gain increases 6 dB per octave so the overall loss becomes independent of frequency. When antennas of constant physical area are used on **both** ends, the increase in total antenna gain is 12 dB per octave, so the net transmitter-to-receiver loss actually decreases 6 dB per octave. This comes from the transmitting antenna being able to focus more of its power on the receive antenna. Isotropic radiation has the same intensity regardless of the direction of measurement.

Part of losses in the radio link takes place in the cables that connect the transmitter and the receiver to the antennas. The losses depend on the type of cable and frequency of operation and are normally measured in dB/m. Nevertheless, no matter how good a cable is, it always has a loss. Because of that, remember to always keep the antenna cable as short as possible. Typical loss in cables is 0.1 dB/m to 1dB/m. Also, we must allow at least 0.25 dB (loss) for each connector across cables. Reception is reliable when RxP is greater than the receiver sensitivity. The sensitivity of a receiver is the minimum magnitude of input signal required to produce a specified output signal with a specified signal-to-noise ratio.

13-1.2. Non-Line of Sight Radio Links

The link budget for an over-the-horizon radio path may include other path losses such as refraction, reflection, multipath. For instance, the following figure depicts a non LOS line, which involves surface and sky wave propagation, via the ionosphere layers. Such deployments may have path losses that are related to the inverse cube of the distance.

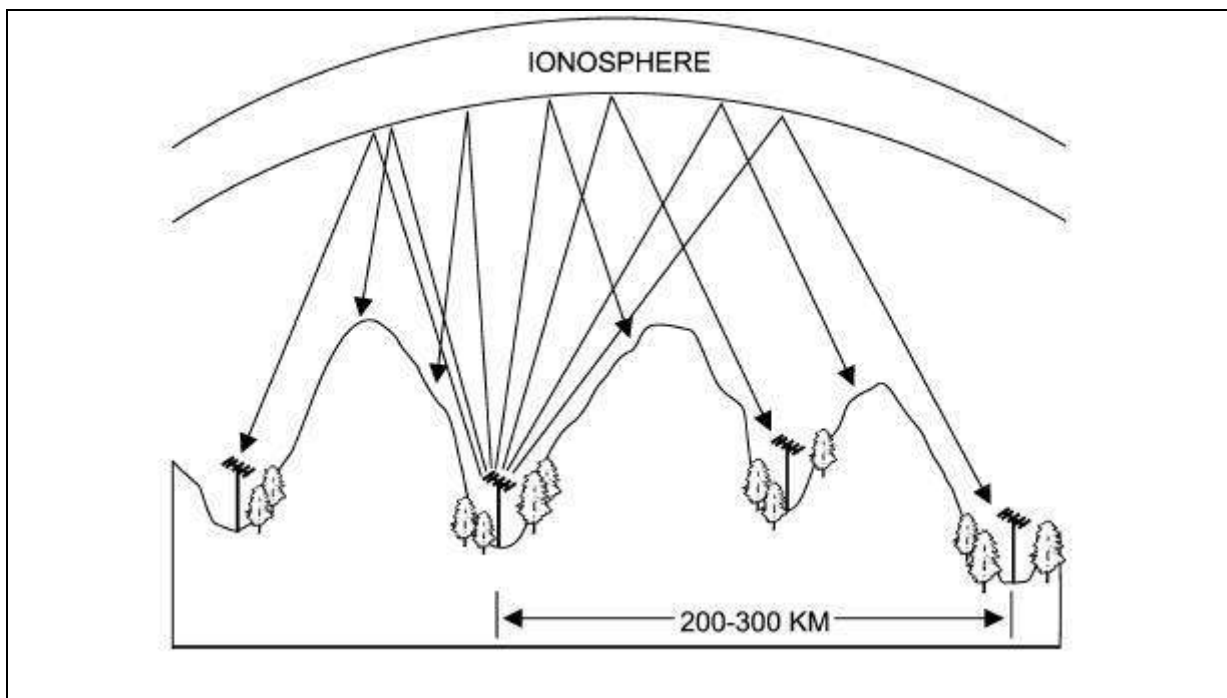


Fig. 13-4(a). Non-LOS links, which involve earth surface and sky waves

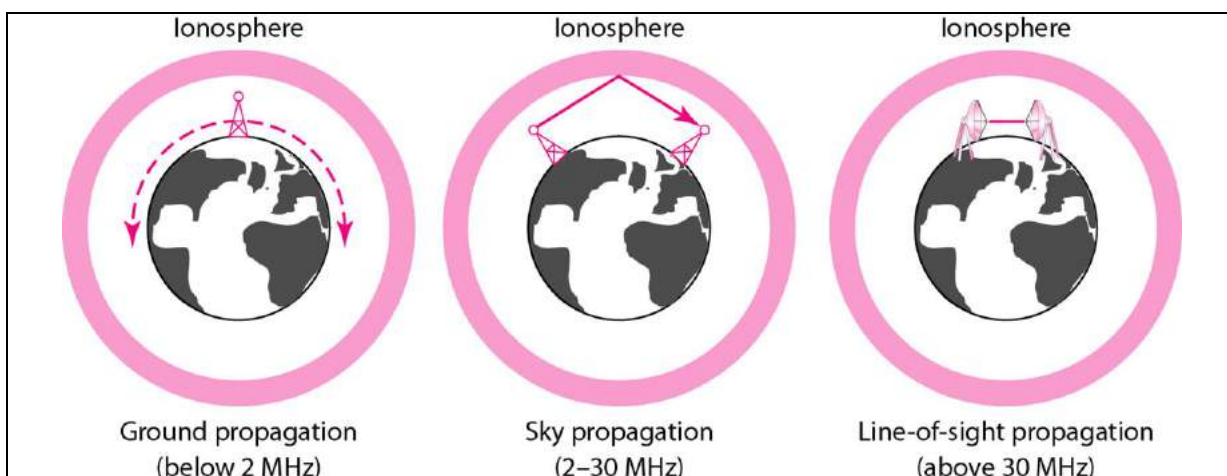


Fig. 13-4. Illustration of RF propagation modes and radio links,

13-1.3. Other Media Links

Guided media such as **coaxial** cables, twisted pair (**TP**) lines, waveguides and optical **fibers** have losses that have exponential decay with distance. This means that there is always a crossover distance beyond which the loss in a guided medium will exceed that of a line-of-sight path of the same length. Long distance fiber-optic communication became practical only with the development of ultra-transparent glass fibers. A typical path loss for single mode fiber is 0.2 dB/km, far lower than any other guided medium.

13-2. Satellite Communication & VSAT

Today, the satellite communication services are used for many different purposes, such as voice and data transmission, radio and television broadcast, maritime and aeronautical communications. It had been theoretically demonstrated during the fifties that an object at about 36,000 km above the earth would rotate at the same speed, and therefore appear stationary. The first satellite equipped with on-board radio-transmitter was the Soviet Sputnik 1, launched in 1957. NASA launched its first satellite in 1960; which contained a passive reflector for radio communications. Nowadays, communications satellites provide a microwave radio relay technology complementary to that of submarine communication cables. They are used for mobile applications and for TV & radio broadcasting, for which cables are impractical or impossible.

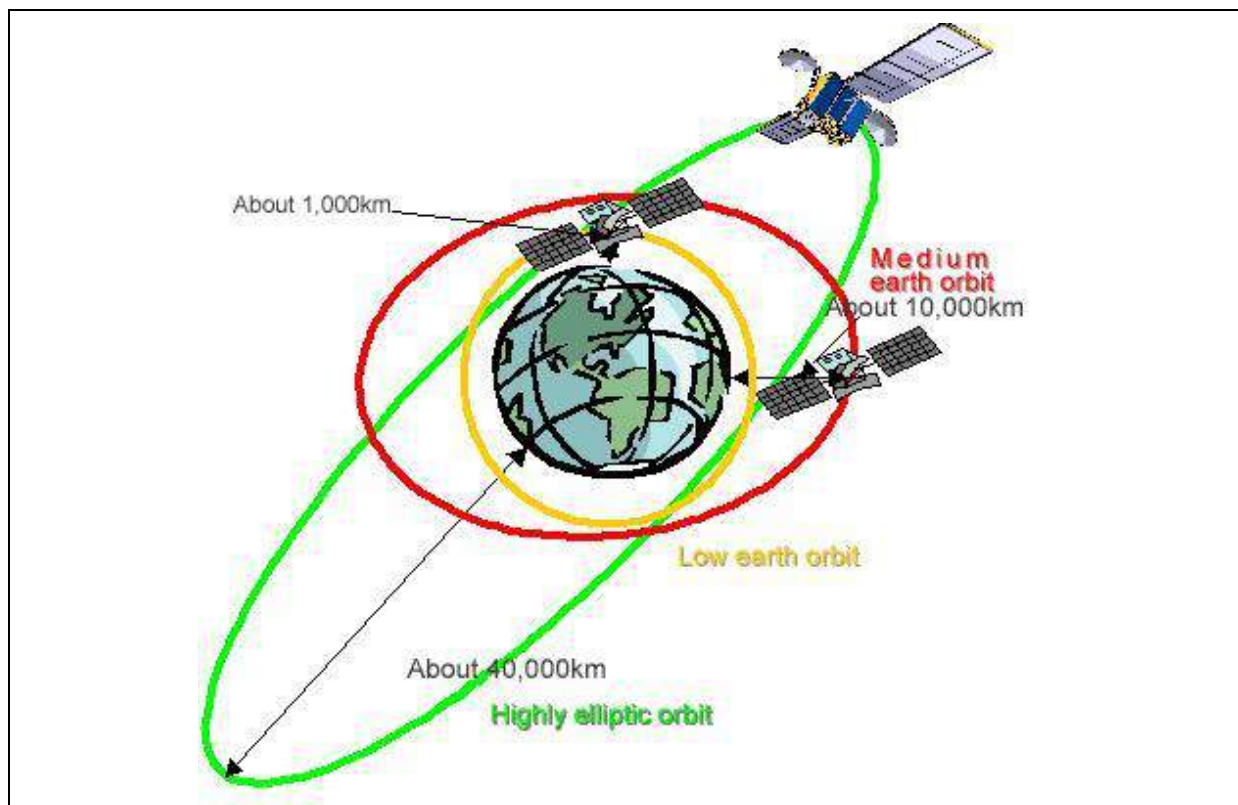


Fig. 13-5. Examples of communication satellites, with different orbits

Television became the main market, its demand for simultaneous delivery of large bandwidth signals to many receivers being a more precise match for the capabilities of geosynchronous comsats. The direct broadcast satellites (**DBS**) transmit to small dishes (45 to 60 cm in diameter) and generally operate in the upper portion of the microwave K_u band. At the late 1990s, satellite communication technology has been used as a means to connect to the Internet via broadband data connections. By 2004

popular mobile direct broadcast applications started to appear, with the geographic positioning system (**GPS**) technology as a reference.

13-2.1. Types of Satellites

Modern communications satellites use a variety of orbits including geostationary orbits, elliptical orbits and low Earth orbits (**LEO**). Low earth orbiting satellites are less expensive to position in space than geostationary satellites and, because of their closer proximity to the ground, require lower signal strength. The Very-Small Aperture Terminal (**VSAT**) is a sort of geostationary satellite communication systems.

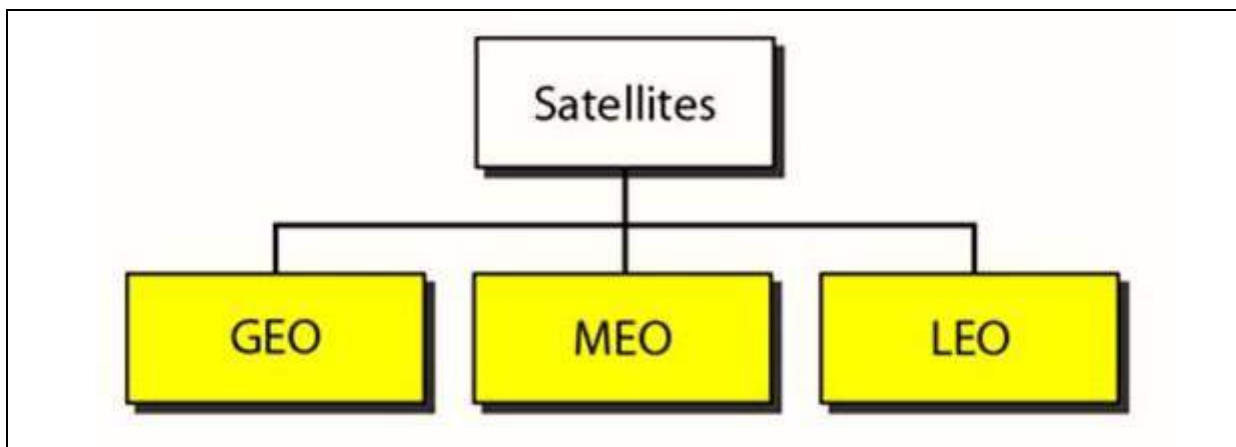


Fig. 13-6. Types of Satellites

i. Orbiting Satellites

Orbiting satellites, such as remote sensing and spying satellites, usually have lower orbits and cheaper to launch. Lower orbit satellites are usually located at about 800km altitude (about 1/8 earth radius). Orbiting satellites also have the following features:

- Not available all the time for communication links
- Earth coverage obtained by rotation of earth beneath satellite.
- Receive antennas must track satellite
- Lower coverage than geostationary

The so-called low Earth orbit (LEO) satellite technology can provide worldwide wireless coverage with no gaps. LEO satellites orbit the earth at high speed, low altitude orbits with an orbital time of 70–100 min, an altitude of 640 to 1120 km. Since the satellites are not geosynchronous, they must fly complete orbits in order to guarantee complete coverage over global areas by at least one satellite at all times.

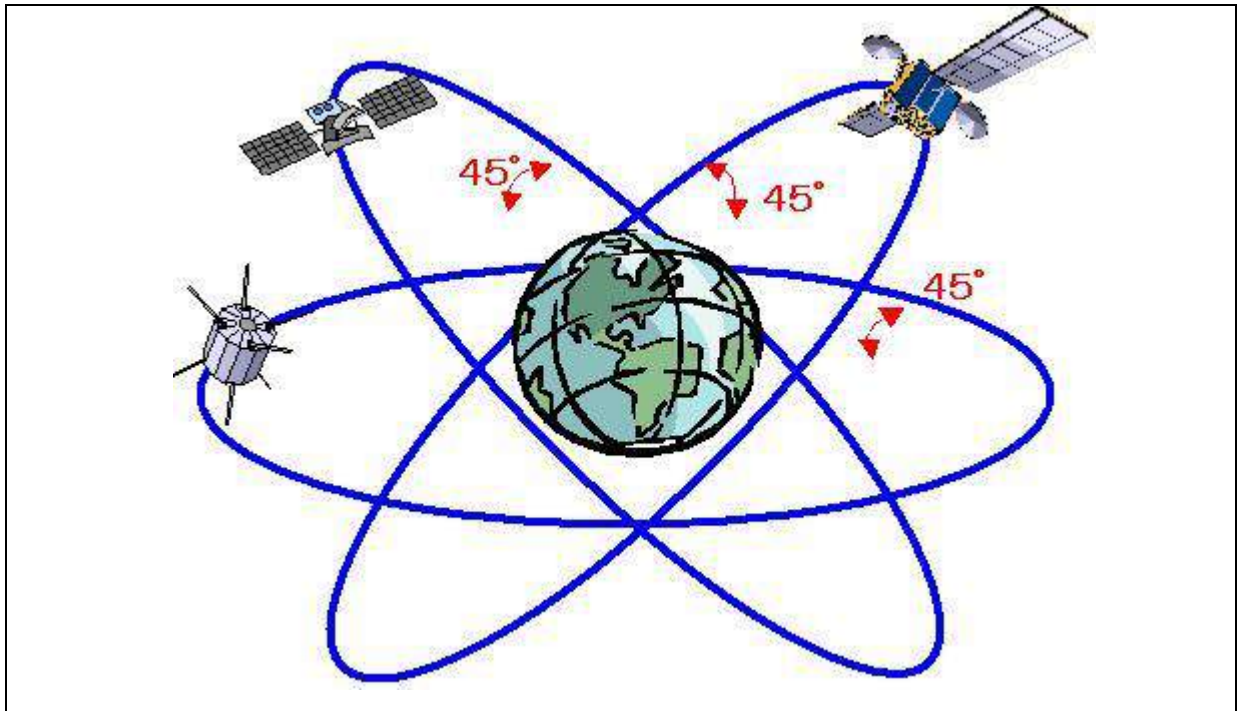


Fig. 13-7. Constellation of orbiting satellites around Earth.

ii. Geostationary Satellites

Geostationary_satellites, such as TV and mobile phone satellites, occupy fixed position with respect to earth above the equator. The satellite antenna beamwidth must correspond to the area of the earth to be illuminated (footprint). The earth station antenna must be able to select a particular geostationary satellite. Figure 13-7 illustrates the fixed area coverage (footprint) of a typical geostationary satellite. Geostationary satellites also have the following features:

- No tracking is required
- Radius of orbit = 42000 km, altitude = 36000 km and orbital period 24h.

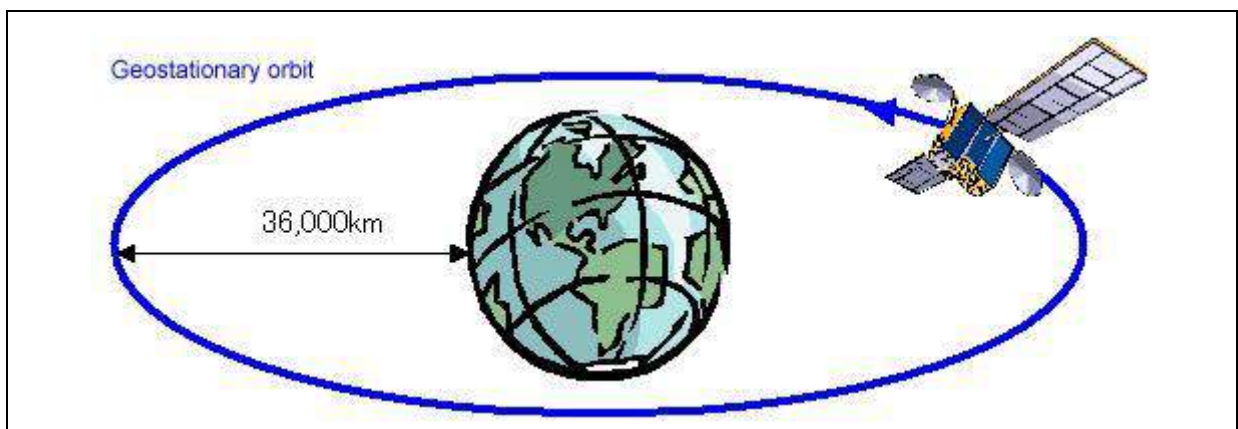


Fig. 13-8(a). Illustration of the geostationary satellite and its orbit around Earth

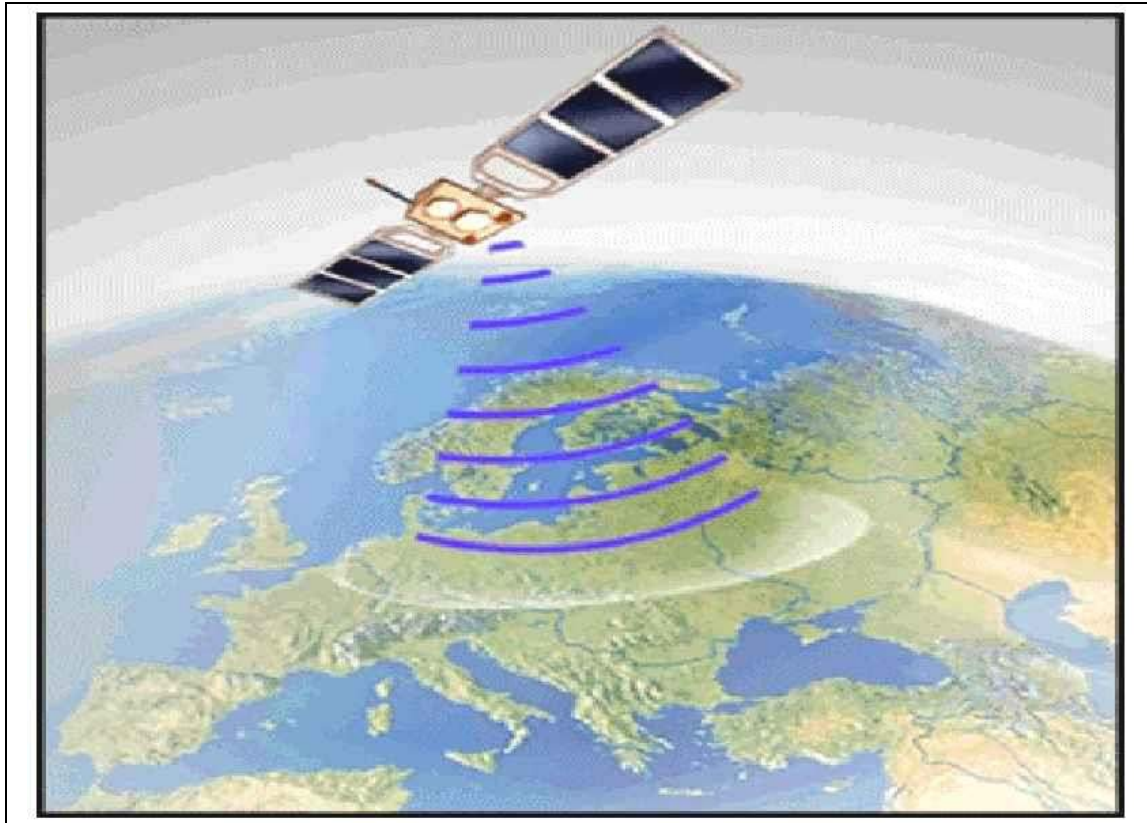


Fig. 13-8(b). Coverage (footprint) of a geostationary satellite.

One of the most important applications of geostationary satellites is for TV direct broadcasting. A typical broadcasting satellite has up to 32 transponders for Ku-band and up to 24 for a C-band only satellite, or more for hybrid satellites. Transponders have a bandwidth between 27 and 50 MHz. Each geostationary C-band satellite needs to be spaced 2° from the next satellite to avoid interference; for K_u , the spacing can be 1° . This means that there is an upper limit of $360/2 = 180$ geostationary C-band satellites and $360/1 = 360$ geostationary K_u -band satellites. C-band transmission is susceptible to terrestrial interference while K_u -band transmission is affected by rain.

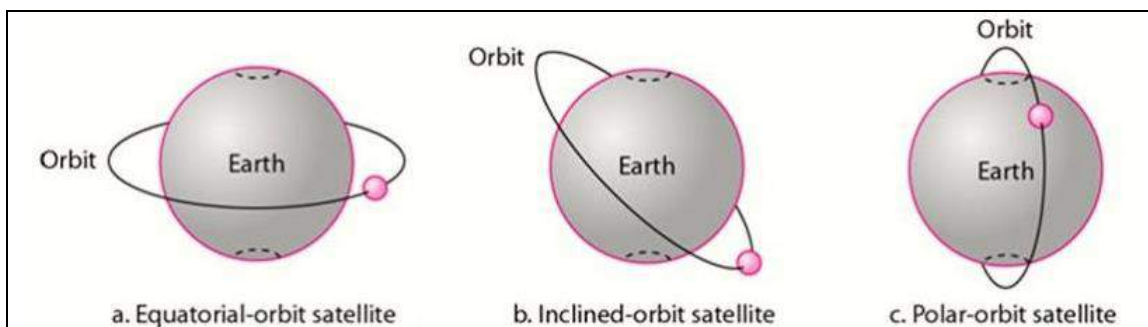


Fig. 13-9.. Block diagram of a satellite link

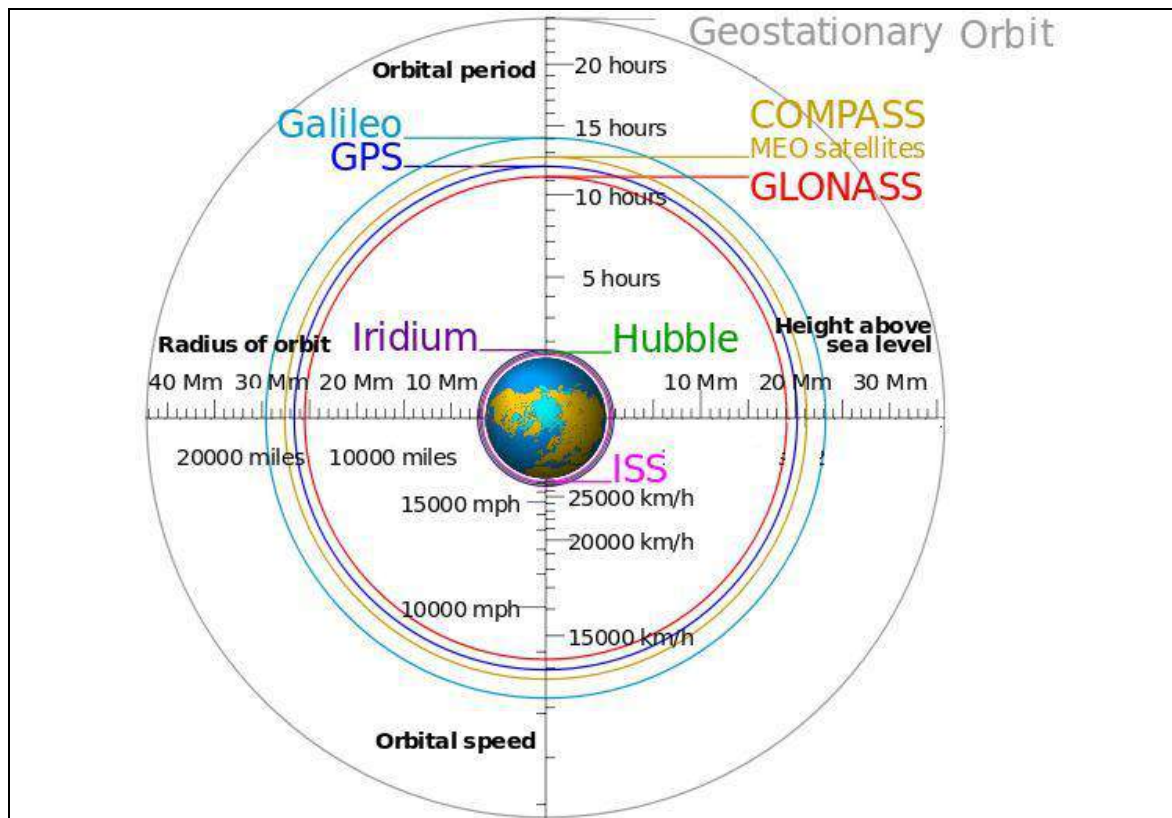


Fig. 13-10. Comparison between geostationary and orbiting satellite systems.

iii. Very-Small Aperture Terminal (VSAT)

VSAT is the acronym for Very Small Aperture Terminal. When you require a number of remote users to have data access to a central hub then a VSAT satellite communications network is worth considering. VSATs are suitable for both point to point networks and mesh networks. Point to point networks may be used to link small ISPs to the internet backbone. VSATs are at the lower end of the product line. At the higher end are large stations that support large capacity satellite links.

VSAT systems are used mainly for international switching networks to support trunk style telephony services between continents. These links are typically in the range of 100Mb/s and are owned and operated by national telecom operators. At the lower end are the VSATs. These are small stations with antenna diameters from 2.4m down to 45cm, hence '*small aperture*' which is in reference to the area of the antenna. These stations are cheap and are easy to install. However, their capacity is in the range of a few tens of kb/s.

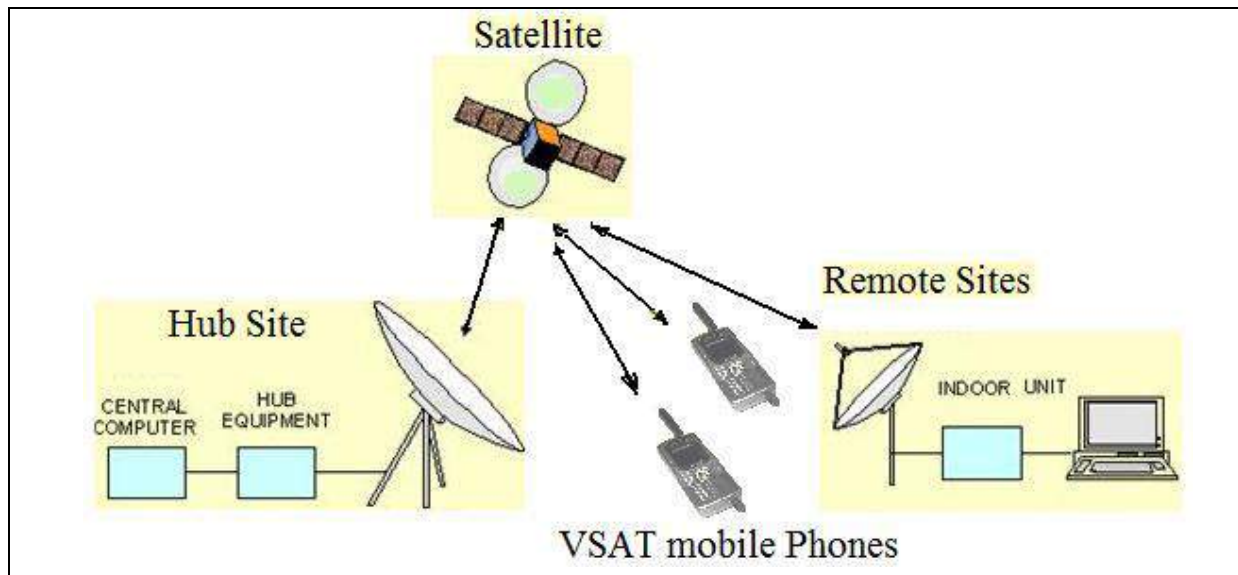


Fig. 13-11. Components of a VSAT system.

VSATs are available at ones premises and this avoids the need to use public network links to access the earth station. The user can directly plug the VSAT equipment into his own terminal, such as telephone or personal computer. As such VSATs are a means to bypass public network operators by directly accessing satellite capacity. They are tools for establishing private and public networks communications solutions that can support the internet, data and voice. Direct satellite to the end user is too expensive for many people and in such cases the VSAT terminal needs to be considered as a local community node, such as internet cybercafé, or town-ISP, linked either with several PCs locally or via wireless network to the local community. A typical VSAT network comprise small remote VSAT antennas (1.2m diameter) located on customer offices to provide data communications via a large central 11m hub antenna. A landline optic fibre data link connects the hub antenna to the head office. The hub antenna and the VSATs are owned, operated and maintained by a VSAT service provider who provides a communication service to the customer company. The hub station is a large satellite dish, typically 6m to 32m diameter. The large size means high receive gain and sensitivity and this minimizes the transmit power and dish size required by the remote customer terminals. In order to work well for small dish transmit services the satellite uplink needs to have high sensitivity (gain to noise temperature ratio, G/T). This is most readily achieved if the uplink beam coverage area is small. Also the satellite transponder has to have a sufficiently high gain. This is not technically difficult but does need to be specified before satellite construction starts. The gain of a satellite may be adjusted in orbit by remote control of a gain step attenuator. The frequency

bands can be any of C band 4/6 GHz, Ku band 10-12/14 GHz or higher (Ka band). The higher Ka bands are rather unused.

Possible VSAT applications include: Internet access, telephony, environmental monitoring, police, and point of sales (POS). Capacity is not a problem with both very small to very large networks all viable. Growth is possible in small steps thus keeping costs related to actual use. VSAT systems provide a both data and, if required, voice services. Voice or data services require slightly larger antennas and/or transmitter power amplifiers. Mesh networks may be attractive for telephony applications, exploration, disaster, emergency relief etc.

Some satellite phones use satellites in geosynchronous orbit. These systems can maintain global coverage with only three or four satellites, reducing the launch costs. However the satellites used for these systems are very heavy (about 5000kg) and therefore very expensive to build and launch. The satellites sit at an altitude of about 35,000 km and therefore a noticeable delay is present while making a phone call or using data services. The amount of bandwidth available on these systems is substantially higher than that of the Low Earth Orbit (**LEO**) systems, all three active systems provide portable satellite internet using laptop-sized terminals with speeds ranging from 60 kbits to 512 kbits. Another disadvantage of geostationary satellite systems is that in many areas, the line of sight between the phone and the satellite is broken by obstacles such as steep hills and forest and the user will need to find higher ground before being able to use the phone. This is not the case with LEO services - even if the signal is blocked by an obstacle one can wait a few minutes until another satellite passes overhead. For instance, **Thuraya** is a VSAT system based in the UAE which operates three satellites. Thuraya provides coverage to the most of Europe, Asia, Africa and Australia.

13-2.2. Satellite Link Budget

Figure 13-12 shows a typical earth-terminal receiver, consisting of a low-noise amplifier (LNA), mixer (down-converter), and intermediate frequency amplifier (IFA). Assume the noise figure of these components, including the receiving antenna, are as follows:

$$F_{ant} = 1.2, F_{LNA} = 1.2, F_{mixer} = 2.6, F_{IF} = 5.0$$

Assume also that the available power gains of the two amplifiers (LNA and IFA) are given by:

$$G_{LNA} = 200, G_{IF} = 1000,$$

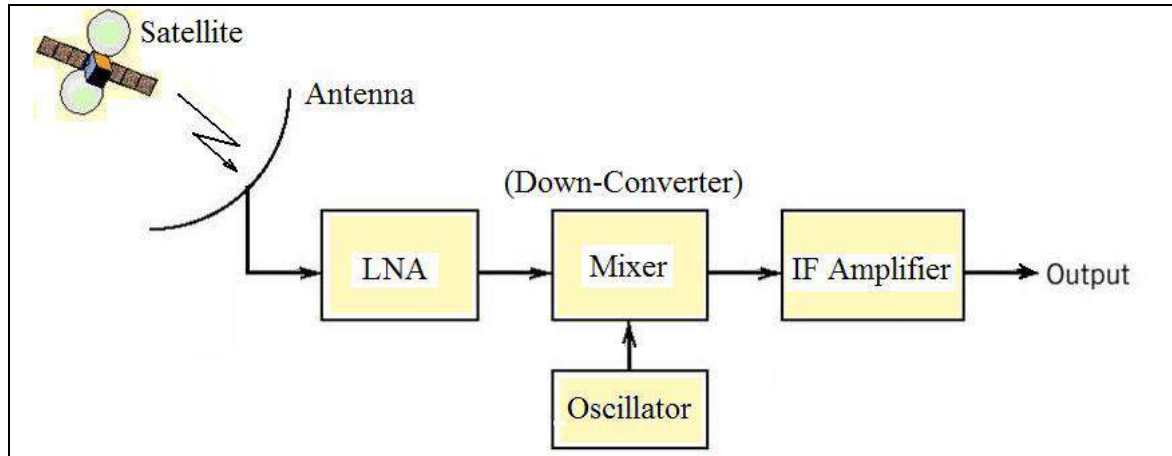


Fig. 13-12. Block diagram of a satellite link

The receiving earth terminal has a 2m-dish antenna with a power gain G of 45 dB. Calculate the ratio of received carrier power-to noise spectral density, denoted by C/N_o .

Step i. Equivalent noise temperature of the satellite receiver

The noise temperatures of the above receiver is calculated as follows:

$$\begin{aligned}
 T_{ant} &= T_o(F_{ant} - 1) = 300(1.2 - 1) = 60K, \\
 T_{LNA} &= T_o(F_{LNA} - 1) = 300(1.2 - 1) = 60K, \\
 T_{mixer} &= T_o(F_{mixer} - 1) = 300(2.6 - 1) = 480K, \\
 T_{IF} &= T_o(F_{IF} - 1) = 300(5 - 1) = 1200K
 \end{aligned}$$

To calculate the equivalent noise temperature of the satellite receiver, we use the **first Friis formula**, which expresses the overall noise temperature of the cascade connection of any number of 2-port networks:

$$T_e = T_1 + T_2/G_1 + T_3/G_1G_2 + \dots$$

or

$$T_e = T_{ant} + T_{LNA} + (T_{mixer} + T_{IF})/G_{LNA} = 60 + 60 + (480 + 1200)/200 = 128.4 K$$

Step ii. Satellite Download Link Budget Analysis

In a digital satellite communication system, one of the key elements in the overall design and analysis of the system is the downlink power budget, which is usually more critical than the uplink power budget because of the practical constraints imposed on the downlink power and satellite antenna size. The critical parameter to be calculated is the ratio of received carrier power-to noise spectral density, denoted by C/N_o . According to the Friis transmission equation (1-5), the relation of average power received (P_r) at the earth terminal to the average power (P_t) transmitted by the satellite is

given by:

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi d} \right)^2$$

Here, G_r is the power gain of the receiving antenna, G_t is the power gain of the satellite antenna, λ is the downlink carrier wavelength, and d is the distance between the satellite and the earth terminal. Given that the equivalent noise temperature of the system is T_e , then the noise spectral density $N_o = kT_e$.

From the first Friis equation we note that $P_t G_t$ is equal to the effective isotropic received power (**EIRP**) of the satellite.

$$(EIRP)_{satellite} = P_t G_t$$

Therefore, dividing P_r by $N_o = kT_e$, we can calculate the carrier to noise ratio (C/N_o) for the downlink as follows:

$$(C/N_o)_{downlink} = P_r / N_o = (EIRP)_{satellite} \cdot (G_r / T_e)_{earth\ terminal} \cdot (1/4\pi d)^2 \cdot (1/k)$$

On the earth terminal side, we see from the above equation that the (C/N_o) ratio is proportional to (G_r / T_e) and may therefore be used to assess the quality of an earth terminal. It is usually abbreviated as (G/T) ratio, which is actually the figure of merit of the receiving earth terminal. Therefore, we can rewrite the above equation for the (C/N_o) ratio measured in dB or in dBW, which denotes power referenced to 1W, that's $10 \cdot \log(P/1W)$.

- Calculate $10 \log k$, to represent the gain in dBW / kHz.
- Calculate $(EIRP)_{satellite}$
- Calculate $(G/T)_{earth\ terminal}$, measured in dB / K.
- Calculate the free-space loss $L_{free\ space} = 10 \cdot \log(4\pi d \lambda)^2$, in dB.

Table 13-1. Satellite downlink budget calculations

Parameter	Value
Boltzmann constant (k)	228.6 dBW / kHz
EIRP	46.5 dBW
$(G/T)_{earth\ terminal}$	23.91 dB/K
Free space loss ($L_{free\ space}$)	-206 dB

(C/N_o)	93.0 dB.Hz

The above Table presents the values of the four terms for the downlink of a typical domestic satellite system, based on the following results:

- The thermal noise contribution is $-10 \log k = 228.6 \text{ dBW/kHz}$
- The transponder is operated at its maximum output power yielding an EIRP of 46.5 dBW.
- The free space loss is then:

$$L_{free\ space} = 92.4 + 20 \log(f) + 20 \log(d) \text{ dB}$$

- The receiving earth terminal has a power gain $G = 45 \text{ dB}$ and the receiver is configured with equivalent temperature, $T = 107.5 \text{ K}$. Therefore,

$$(G/T)_{\text{earth terminal}} = 45 - 10 \log(128.4) = 45 - 20.3 = 23.91 \text{ dB/K}$$

Here, the downlink carrier frequency f is in GHz and the distance d between the satellite and the earth terminal in km. For a geostationary satellite, the distance between the satellite and an earth terminal is about 40000 km. Choosing $d = 40000 \text{ km}$ and assuming $f = 12 \text{ GHz}$, yields:

$$L_{free\ space} = 92.4 + 20 \log(12) + 20 \log(40000) = 92.4 + 21.6 + 92 = 206 \text{ dB}$$

Summing all gains and losses, we can get $(C/N_o)_{\text{download}} = 93.0 \text{ dB/Hz}$

The received downlink value of the (C/N_o) ratio may also be expressed in terms of the **required** value of the bit energy-to-noise spectral density ratio, $(E_b/N_0)_{\text{req}}$ at the receiving terminal as follows:

$$(C/N_o)_{\text{download}} = (E_b/N_0)_{\text{req}} + 10 \log(M) + 10 \log(R) \text{ dB}$$

where $10 \log(M)$ is the link margin in dB, and R is the data rate in bit/s.

The link margin allows for excess rain losses in propagation and other power degradations. Typically, the link margin is selected as 4dB for C-band, 6dB for Ku-band, and higher for the higher K-band frequencies. For operation at the Ku-band frequency of 12GHz, we have chosen a link margin of 6dB. Thus, using the value of $(C/N_o) = 93.8 \text{ dB/Hz}$ as calculated from the link budget, taking the link margin $10 \log(M) = 6 \text{ dB}$, and assuming $(E_b/N_0)_{\text{req}} = 13.3 \text{ dB}$, we get:

$$10 \log(R) = 93.0 - 13.3 - 6 = 73.3 \quad \text{or} \quad R = 21.38 \text{ Mb/s.}$$

Assuming the use of coherent QPSK for the transmission of digital data via the satellite, and substituting $(E_b/N_0)_{\text{req}} = 13.3$ dB in equation , we find that the probability of symbol error $P_e = 1.2 \times 10^{-3}$. Thus, the digital satellite communication system in this example permits the data transmission on the downlink at a rate $R = 21.38$ Mb/s with a probability of symbol error $P_s = 1.2 \times 10^{-3}$, when a QPSK digital modulation technique is employed.

Table 13-2. Properties of the uplink and downlink bands

<i>Band</i>	<i>Downlink, GHz</i>	<i>Uplink, GHz</i>	<i>Bandwidth, MHz</i>
L	1.5	1.6	15
S	1.9	2.2	70
C	4.0	6.0	500
Ku	11.0	14.0	500
Ka	20.0	30.0	3500

13-3. Cellphone Communication Links

The cellular mobile phone systems make use of a large number of handsets to cover several geographic areas, called cells. Each cell has its own low-powered transmitter and receiver. A central computer is used to tracks the call, switching or *handing off* the call to the nearest cell site. As mobile users travel from cell to cell, their conversations are handed off between cells to maintain seamless service. Channels (frequencies) used in one cell can be reused in another cell some distance away. This process is called Frequency Reuse.

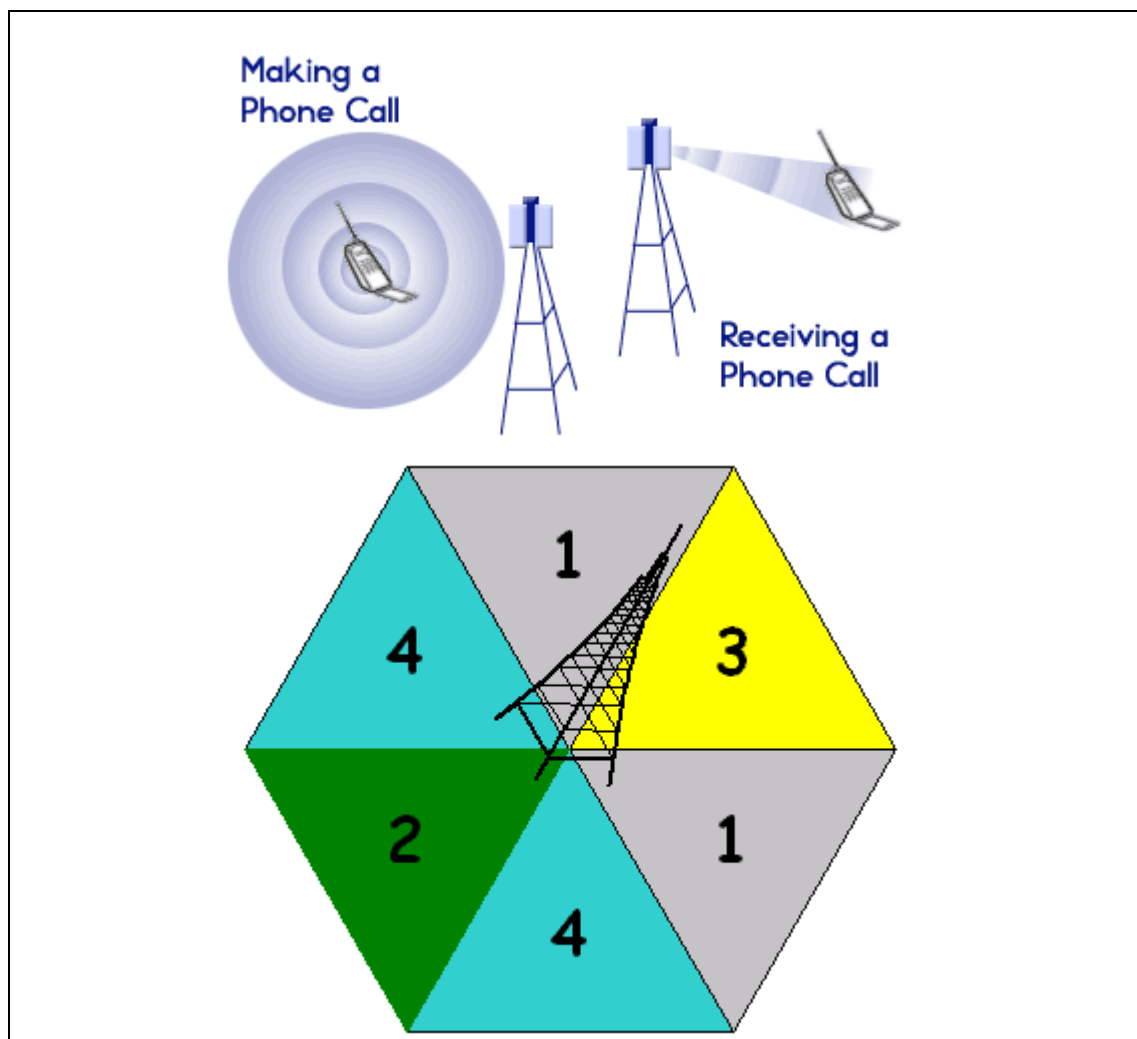


Fig. 13-13. Operation of a cell phone network

The first mobile phones were based on analog technologies, such as **AMPS** in the USA. There was a lack of suitable radio frequencies when the number of users increased. Second generation mobile phones 2G were based on digital modulation schemes. Different industrial and political interests have led to several 2G cellular systems. For the second

generation of mobile phones, the **GSM** standard was adopted in Europe while the 2G TDMA standards (IS-54, IS-136) were adopted, in the USA. There has been also a great effort to develop a 3G standard which made possible a real universal mobile phone. The recent solutions included **CDMA2000** and **WCDMA** technologies.

13-3.1. Cellphone Network

In a cellphone network, each cell can range from one to 20 miles in diameter, depending on the terrain and the capacity of the system. Each cell is equipped with a radio transceiver which is connected through the cellular company's switching center to the local phone network.

There are three main entities in cellular communication:

- 1- Mobile Station (**MS**): A mobile station consists of 2 entities - equipment and SIM card
- 2- Base Transceiver station (**BTS**) or B-node, which consists of a base transceiver (transmitter and receiver) and a base station controller (**BSC**). The BTS is the antenna tower site.
- 3- Main Switching centre (**MSC**): The MSC is the heart of the central switching office which controls all the base stations and provides connection with landline phones. The MSC performs the following tasks:

- Connects calls from sender to receiver,
- Collects details of the calls made and received, and
- Supervises operation of the rest of the network components.

Because cell phone network is a multi-user environment, it makes use of multiple access techniques, that we described in Chapter 2, namely: CDMA, TDMA, FDMA, OFDMA or combinations of them. Actually, the third generation (3G or UTMS) of mobile phones makes use of the CDMA technique. In this method, the direct sequence (input data) is multiplied with a pseudorandom code sequence (PN sequence) which makes it spread over the entire bandwidth of the communication channel. In fact, The signal is spread at two levels first using a Walsh Code and then using a PN Code. The number of bits in either of the two codes is known as the **chip rate**, and each bit in the spreading signal is called a **chip**. Base station is the one that assigns spreading code to each call when a mobile requests for a call. The **WCDMA** have a channel bit rate of 3.84 M Chip/s, which is combined with QPSK modulation. The system operates carrier radio frequencies around 2 GHz (downlink 2110-2170 MHz, uplink 1920-1980 MHz) with 5MHz channel spacing. The 3G made possible new types of services, such as real video and Internet access.

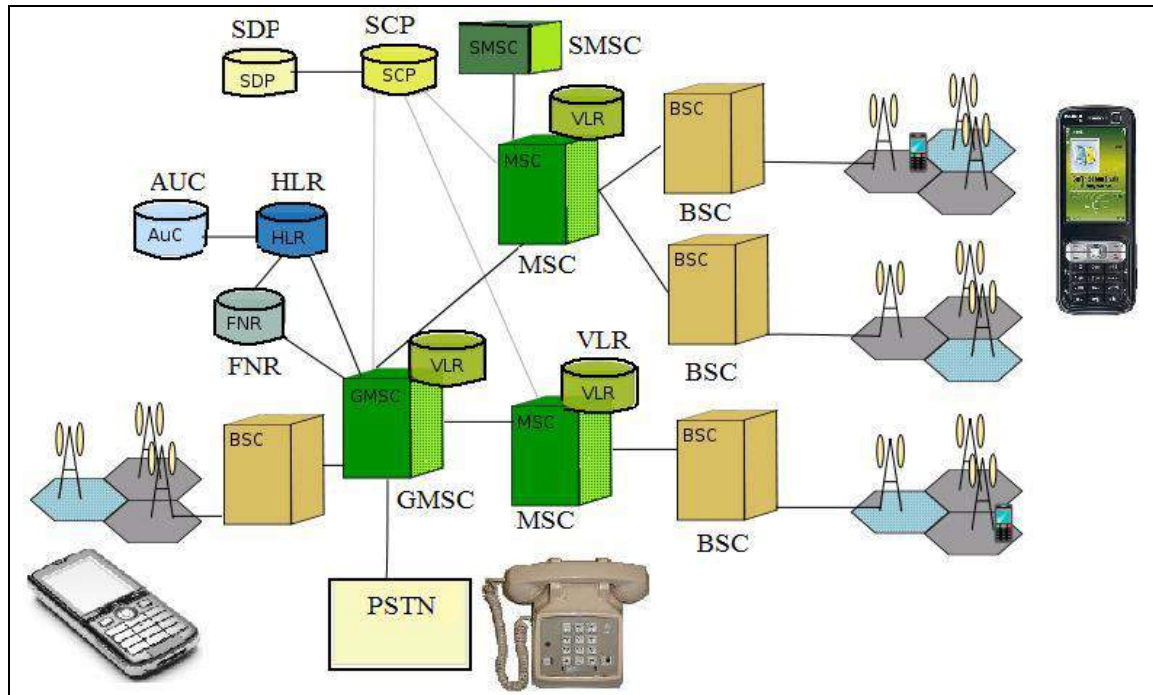


Fig. 13-14. Cell phone network, and its components.

As the propagation losses between BS and MS's are different according to individual communication distances, the received levels at the base station are different from each other when all mobile stations transmit their signals at the same power. Moreover, the received level fluctuates quickly due to fading. In order to maintain the strength of received signal level at BS, power control technique must be employed in cell phone systems.

13-3.2. Cell Phone-Satellite Link Power Budget

We demonstrate here the power calculations of a cell phone satellite - return link. We consider the path from cell phone to hub only. The uplink antenna diameter of the cell phone is usually not known. You may adjust this value to get whatever transmission gain you think realistic. We may estimate 0dBi which means an omnidirectional antenna. You need also to choose a suitable mobile uplink frequency, say 2110-2170 MHz (3G). We may guess the cell phone power output to be 200 mW and carrier bandwidth 170 kHz. For the spacecraft uplink antenna we may guess it as 30m diameter. You need to run the uplink (G/T) calculation for this separately. For instance, 30m diameter at 1.85 GHz, 65% efficiency and 300K noise temp give a (G/T) of about +28.5 dB/K. The noise temp is high because it looks down at the warm earth. Table 13-3 depicts the satellite uplink budget calculation. Also, Table 13-4 depicts the satellite downlink budget calculation.

Table 13-3. Satellite uplink budget calculations

Uplink mobile frequency GHz	1.85
Uplink mobile antenna diameter m	0.066
Uplink mobile antenna aperture efficiency e.g. 0.65	0.65
Uplink mobile antenna transmit gain dBi	?
Uplink mobile antenna, power at the feed W	0.2
Uplink mobile EIRP dBW	?
Range (35778 - 41679) km	38500
Uplink path loss dB	?
Uplink pfd at satellite dBW/m ²	?
Bandwidth Hz	170000
Satellite uplink G/T dB/K	28.5
Uplink C/N dB	?

Table 13-4. Downlink budget calculation

Downlink frequency GHz	34
Downlink hub receive antenna diameter m	0
Downlink hub receive antenna aperture efficiency e.g. 0.65	0.65
Downlink hub system noise temperature(antenna+LNA) K	250
Downlink hub receive antenna gain dBi	?
Downlink hub receive antenna G/T dB/K	?
Downlink satellite EIRP dBW	16
Downlink path loss dB	?
Downlink (C/N) dB	?

On the downlink we may assume some suitable frequency (e.g., 34 GHz). For the satellite downlink EIRP you need to adjust the satellite EIRP per 200 kHz bandwidth till you get an acceptable downlink (C/N_o).

In the default example above a downlink satellite EIRP of +16 dBW per 200kHz is about enough. If the downlink satellite antenna is 43 dBi (e.g. 0.5m diameter at 34 GHz) then you need a spacecraft TX power of -27 dBW per 200kHz which is not difficult to achieve.

13-4. Computer and Internet Links

The **Internet** is a global system of interconnected computer networks that interchange data by packet switching using the standardized Internet Protocol Suite (TCP/IP). It is a network of networks that consists of millions of private and public, academic, business, and government networks that are linked by copper wires, fiber-optic cables, wireless connections, and other technologies. The Internet carries various information resources and services, such as electronic mail, online chat, file transfer and file sharing, online gaming, and the inter-linked hypertext documents and other resources of the World Wide Web (WWW).

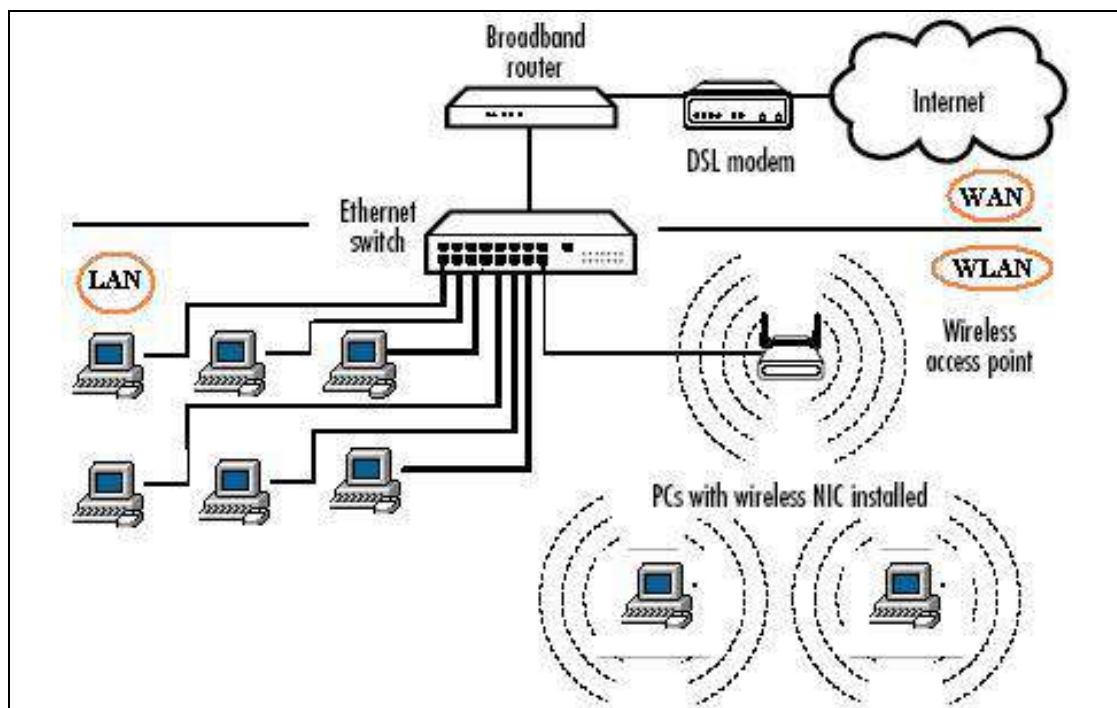


Fig. 13-15. Example of WLAN and its connection to Internet.

The most prominent component of the Internet model is the Internet Protocol (IP) which provides addressing systems for computers on the Internet and facilitates the internetworking of networks. IP Version 4 (IPv4) is the initial version used on the first generation of the Internet and is still in dominant use. A new protocol version, IPv6, was developed which provides larger addressing capabilities and more efficient routing of data traffic. IPv6 is currently in deployment phase around the world.

The Internet can now be accessed virtually anywhere by numerous means. Mobile phones, handheld game consoles and cellular routers allow users to connect to the Internet from anywhere there is a cellular network supporting that device's technology. Common methods of home access

include dial-up, landline broadband (over coaxial cable, fiber optic or copper wires), Wi-Fi, WiMax, Satellite and 3G cell phones.

13-4.1. Wi-Fi and WiMax

The purpose of Wi-Fi is to provide inter-operable wireless access between devices. Wi-Fi generally makes access to information between devices from different manufacturers easier. In some countries (USA, France, etc) the term Wi-Fi is often used by the public as a synonym for wireless Internet (W-LAN), although not every wireless Internet product has a Wi-Fi certification. Wi-Fi technologies are supported by most personal computer operating systems and covers IEEE 802.11 standards including 802.11a, 802.11b, 802.11g, and 802.11n.

The Worldwide Interoperability for Microwave Access (WiMax) is a wireless technology which has recently widespread for computer and Internet connections. WiMAX is based on the IEEE 802.16 standard. The 802.16d-2004 standard covers fixed applications and is rated for point-to-multipoint network coverage with a maximum range of up to km at speeds of 70 Mb/s with 20 MHz channelization.

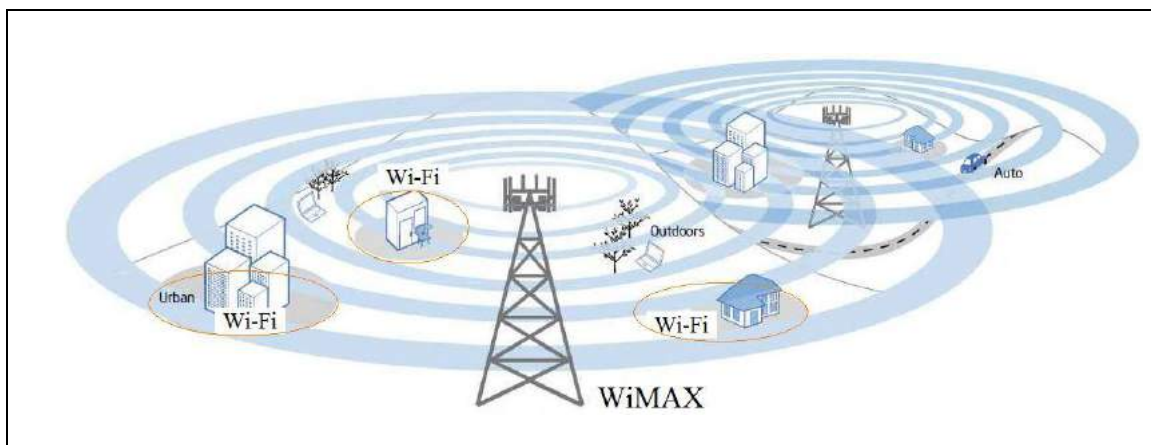


Fig. 13-16. Range of Wi-Fi and WiMAX technologies.

As a metro area network (MAN) its main objective is to be last-mile alternative to DSL and cable modems. But because of the large bandwidth, it is a backhaul alternative for businesses and an alternative to T1 lines.

The IEEE 802.16e-2005 standard, commonly known as mobile WiMax, enables full-featured multimedia mobile applications including voice-over-IP (VoIP), video, data and audio. The 802.16d standard allows for a multitude of modulation schemes (OFDM with BPSK, QPSK, 16QAM, or 64QAM) with channelization of either 1.25 MHz or 20 MHz implemented at any frequency range allowed in the regional deployment below 11 GHz.

13-4.2. Satellite Internet Access

In this configuration many customer VSAT sites, often independent of one another are provided with access to the world internet backbone system, enabling access to web sites, email and other services worldwide. There is a large satellite dish at a Teleport where the interconnection to the terrestrial internet backbone occurs. At the teleport is router equipment and processing equipment to manage all the remote VSAT terminals.

The satellite network typically has a high bit rate (2 to 40 Mb/s) outlink carrier from a large teleport hub dish, used to distribute download data to all sites. The total bit rate is shared amongst all sites. The uplinks, or return link transmissions, from each remote VSAT go to the hub teleport. These messages are mainly mouse click and sent emails. The customer terminals are typically called "satellite internet" dishes (90 cm - 1.2m diameter). The larger sizes are need at the edge of the coverage beam areas. The teleport hub is typically a large 10 m diameter dish.

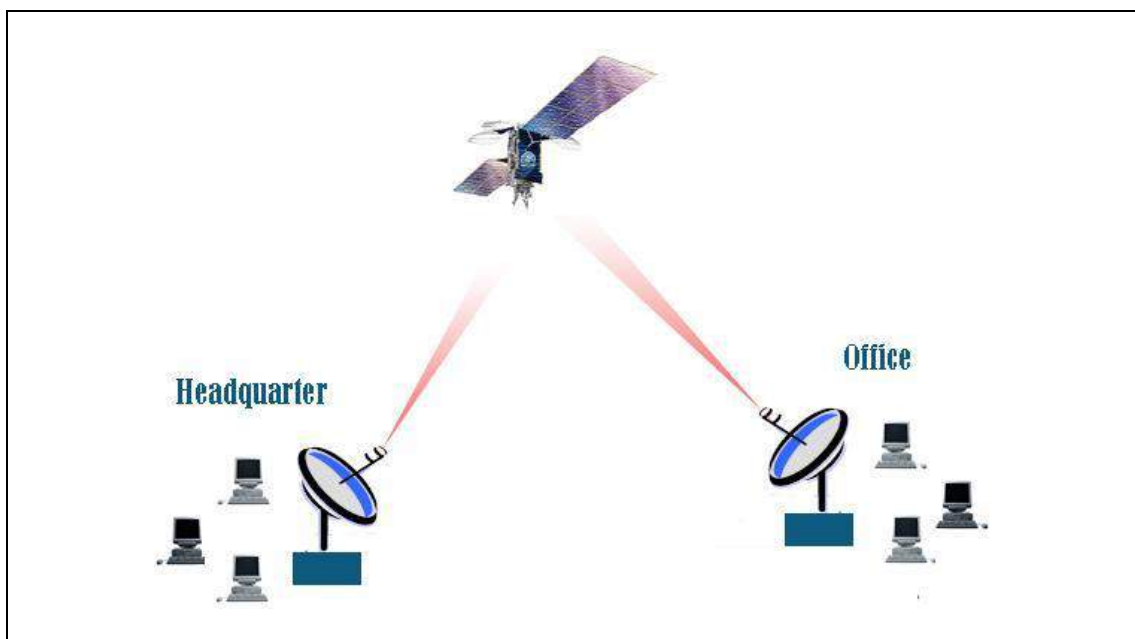


Fig. 13-17. Examples of Internet access via satellite links.

13-5. Public Telephone Switching Network (PSTN)

Now let's look at the basic voice and analog telephone links, including dial-up and special service voice and data networks. The primary reason for existence of traditional telephone (including cellular) carriers is to provide person-to-person voice communications. This fact will continue to remain true as long as our present system of telecommunications endures. Certainly, the methods and processes used to provide such communication have gone through tremendous change; however, we can count on the basic process to continue to be used into the foreseeable future.

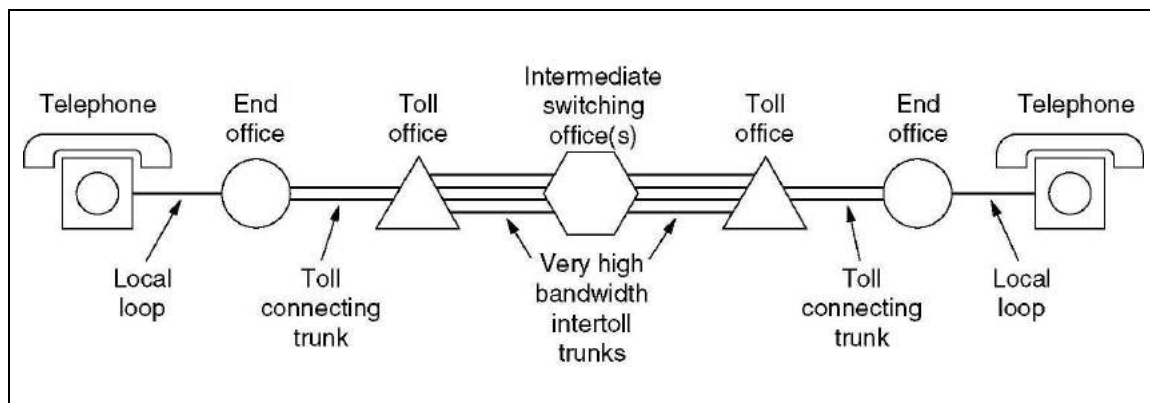


Fig. 13-18. Main components of a telephone network

The dial-up is the term for the primary telecommunication service (POTS) that we all use today. The service is always analog (to the end user), is always switched, and is always 2-wire. The call process involves a protocol that keeps telephone sets in an "idle" status until a user wants to make a call. It is important to understand this protocol, because the dial-up modems must follow the same process. The process involves a telephone number to identify the other end of the communication circuit. Note that POTS is a shared system and that connections between any two points are not guaranteed. Also, a dial-up connection presents the possibility of a security breach that can be used to corrupt the system. Whenever a digital signal is to be transported via the public telephone network it must be conditioned for travel. This is because the basic wiring infrastructure was designed to transport an analog communication signal. Analog communication is accomplished by providing a variable electrical signal which varies as the frequency of a human speaking. Changes in volume and pitch are represented by a smooth flowing electrical current with positive and negative values. A dial-up modem converts your computer output to a series of analog tones which can be transported via the network in the same manner as a voice telephone call.

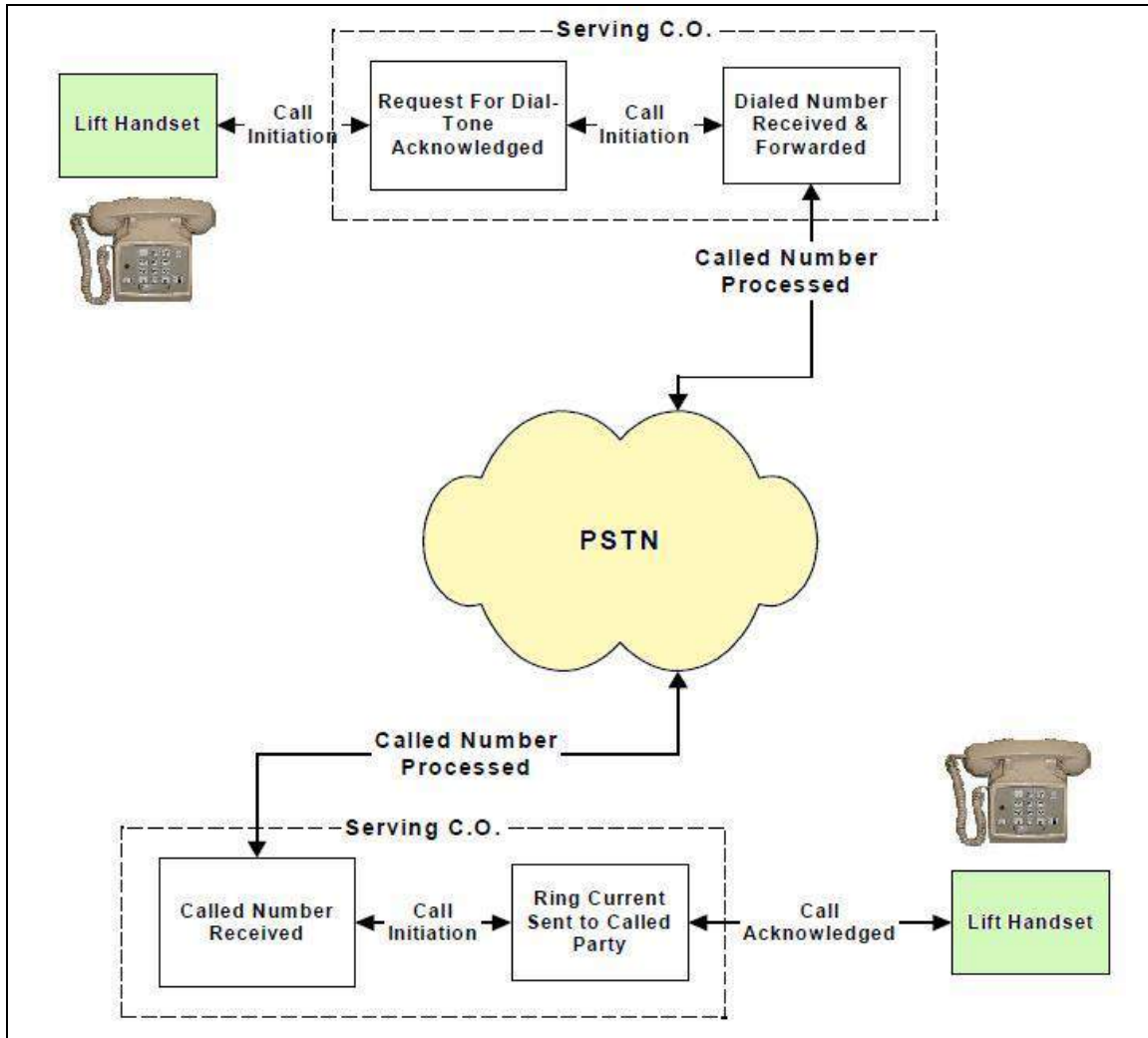


Fig. 13-19. Telephone data network

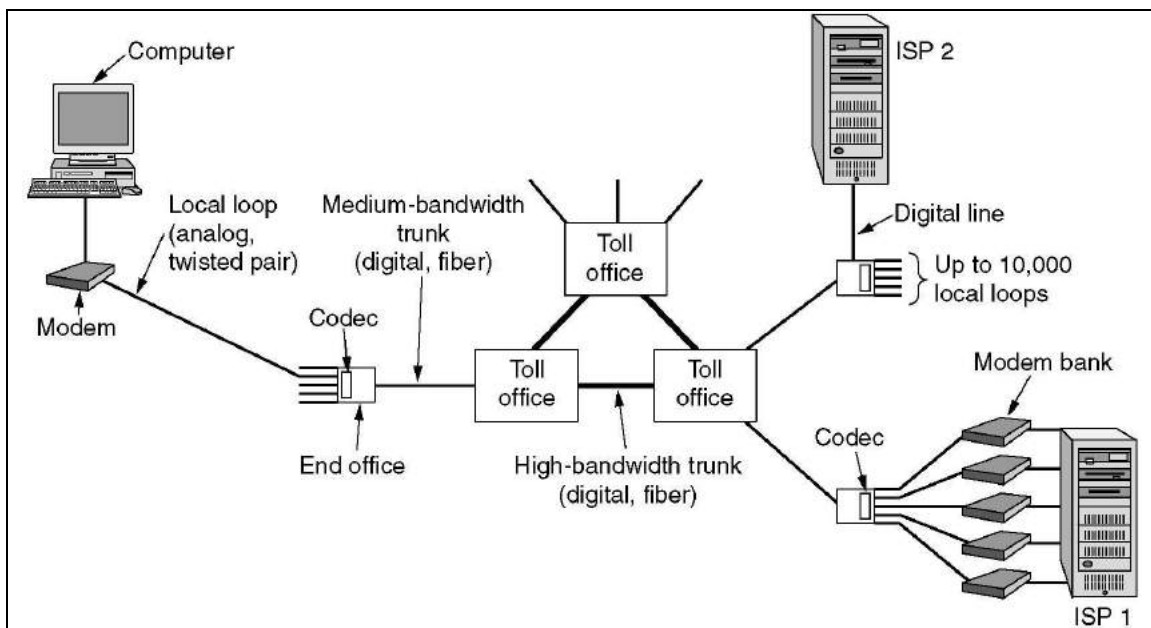
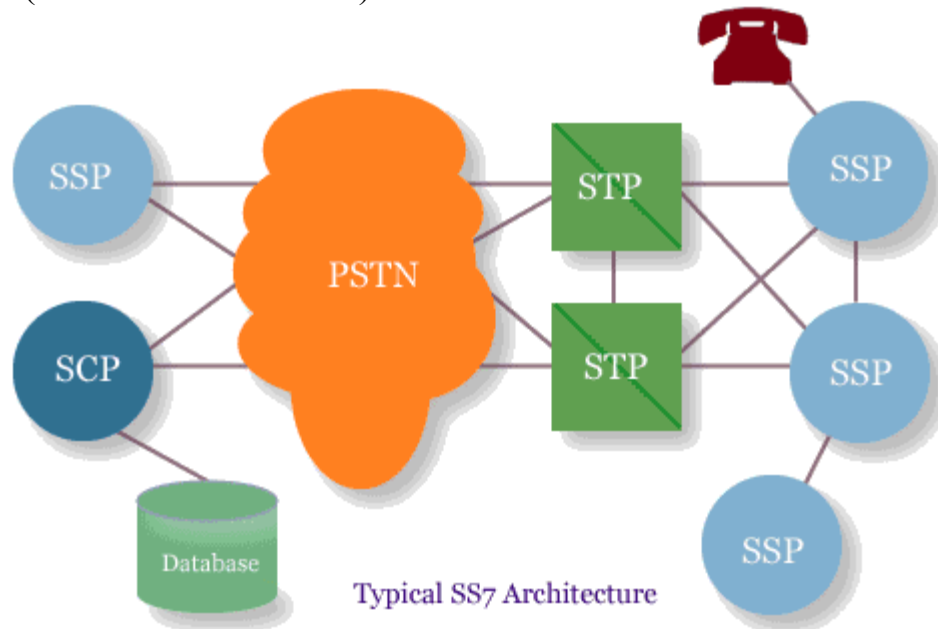


Fig. 13-20. Illustration of telephone data network

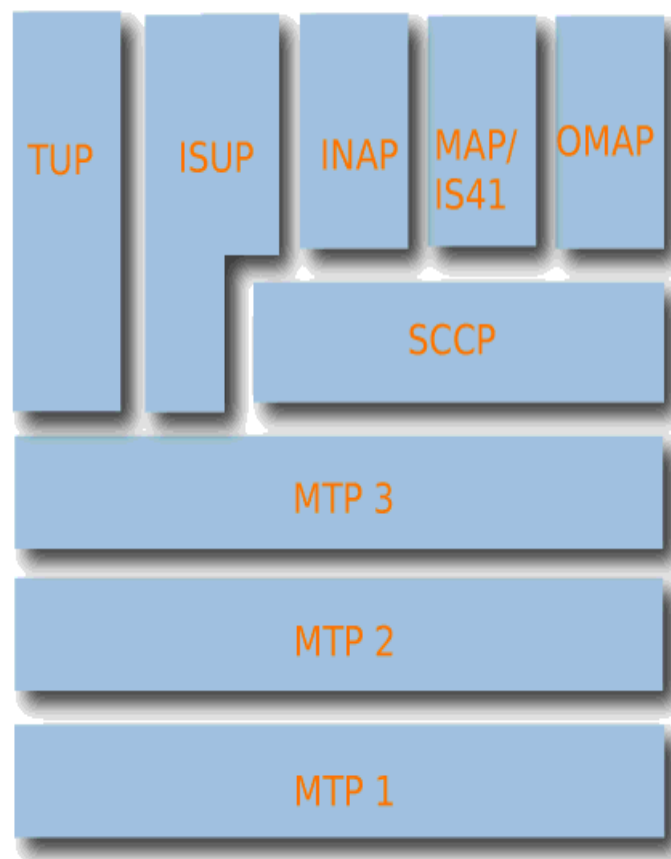
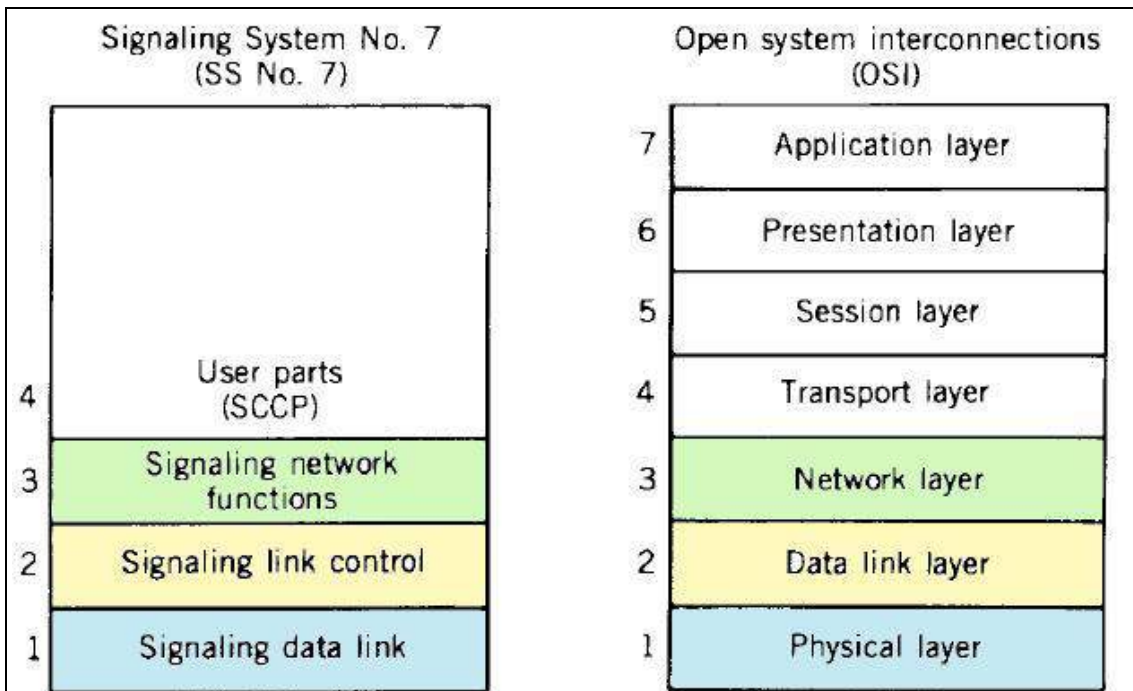
Note 13-2: Signaling and SS7.

The IEEE defines *signaling* as the “exchange of information specifically concerned with the establishment and control of connections and transfer of user-to-user and management information in a telecommunication network”. Conventional signaling has evolved with the telephone network. In fact, there are two essential components to phone calls. The first is the voice and data content. The second is the control signals that instruct telephone exchanges to establish connections and route the content to an appropriate destination. Telephony signaling is concerned with the creation of standards for the latter to achieve the former. These standards are known as protocols. SS7. Thus the Signaling System Number 7 is simply a set of protocols that describe a means of communication between telephone switches in public telephone networks (**PSTN**). They have been created and controlled by various bodies around the world, such as ITU-T. The primary function of SS#7 is to provide call control, remote network management, and maintenance capabilities for the telephone network. There are three kinds of signaling points in the SS7 network:

- SSP (Service Switching Point)
- STP (Signal Transfer Point)
- SCP (Service Control Point)



Like most modern protocols, the SS7 protocol is layered. The following figure depicts the layered structure of the SS7 protocol and its relation to the OSI model/



Physical Layer (MTP-1)

This defines the physical and electrical characteristics of the signaling links of the SS7 network. Signaling links utilize DS-0 channels and carry raw signaling data at a rate of 56 kbps.

Message Transfer Part—Level 2 (MTP-2)

The level 2 portion of the message transfer part (MTP Level 2) provides link-layer functionality. It ensures that the two end points of a signaling link can reliably exchange signaling messages. It incorporates such capabilities as error checking, flow control, and sequence checking.

Message Transfer Part—Level 3 (MTP-3)

The level 3 portion of the message transfer part (MTP Level 3) extends the functionality provided by MTP level 2 to provide network layer functionality. It ensures that messages can be delivered between signaling points across the SS7 network. It includes node addressing, routing, alternate routing, and congestion control.

Signaling Connection Control Part (SCCP)

The signaling connection control part (SCCP) provides two major functions that are lacking in the MTP. The **first** is the capability to address applications within a signaling point. The SCCP allows subsystems to be addressed explicitly.

ISDN User Part (ISUP)

ISUP user part defines the messages and protocol used in the establishment and tear down of voice and data calls over the public switched network (PSN), and to manage the trunk network on which they rely. Despite its name, ISUP is used for both ISDN and non-ISDN calls.

Transaction Capabilities Application Part (TCAP)

TCAP defines the messages and protocol used to communicate between applications (subsystems) in nodes. It is used for database services.

Operations, Maintenance, and Administration Part (OMAP)

OMAP defines messages and protocol to assist the administrators of SS7 network. OMAP includes messages that use the MTP and SCCP for routing.

The SS7 protocol has been the most long-lived, signaling method in telecommunications history. However from the mid 1980s and 1990s, this was not the case. During this time, the reliability of the SS7 network came under scrutiny numerous times due to outages that affected millions of subscribers around the world.

13-6. Digital Subscriber Lines (DSL)

Digital Subscriber Line (DSL) technology is a modem technology that uses existing twisted-pair telephone lines to transport high-bandwidth data, such as multimedia and video, to subscribers. Actually, DSL uses the non-voice frequencies of the telephone line to transmit data. Despite its name, **DSL does not refer to a physical line but to a modem**—or rather a pair of modems. Thus, a DSL modem pair creates a digital subscriber line, but the network does not purchase the lines when it buys a DSL—it already owns those—it purchases modems. A DSL modem transmits duplex data from Exchange or central office (CO) over copper lines of up to 6 km. The other end modem is sometimes called customer premise equipment (CPE). The term xDSL covers a number of similar yet competing forms of DSL, including asymmetric DSL (ADSL), high-speed DSL (HDSL) and very-high speed DSL (VDSL),

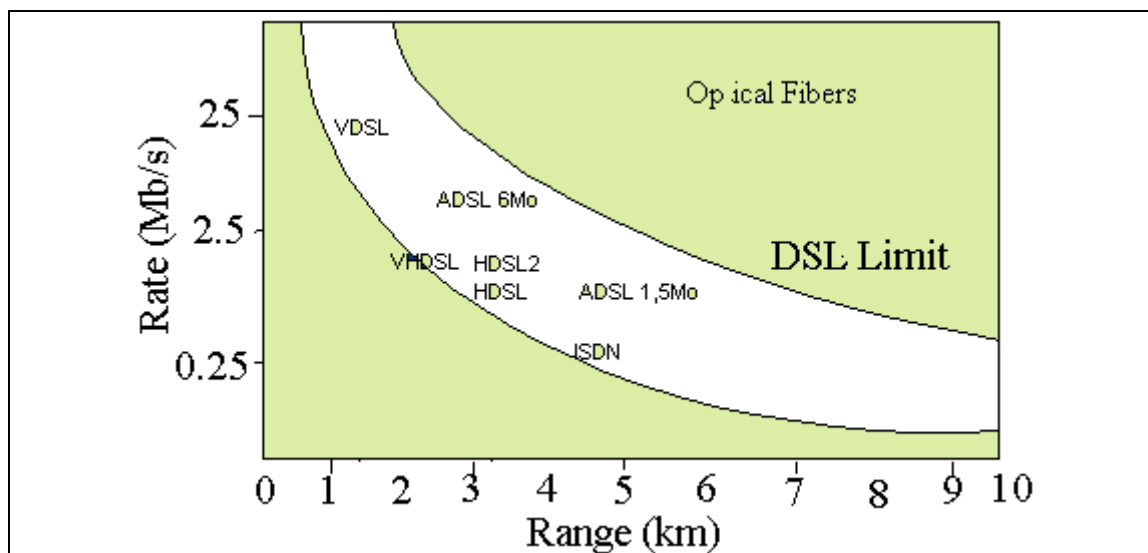


Fig. 13-21. Limits of the DSL technology.

Asymmetric DSL is based on the premise that while surfing the internet, you get a lot more data coming into your browser than what you send to it. Unlike asymmetrical DSL, a high-speed digital subscriber line (HDSL) provides symmetrical pathways for bandwidth—matching upstream and downstream circuits—at speeds ranging from 128 kb/s to 1.544 Mb/s over distances from 1.5km to 6 km. The so-called very-high speed DSL (VDSL) is characterized by asymmetrical transmissions that offer very high data rates over very short distances. The VDSL top speeds are dependent on line length. Maximum downstream rates are projected to reach 55 Mb/s over lines up to 300m long or up to 13 Mb/s over segments as long as 1.5km. Upstream rates will range between 1.6 and 19.2 Mb/s.

13-6.1. Asymmetric Digital Subscriber Line (ADSL)

ADSL can transmit more than 6 Mb/s to a subscriber—enough to provide Internet access, video-on-demand, and LAN access. This increases the existing access capacity by more than 50-fold enabling the transformation of the existing public network. ADSL modems provide data rates consistent with T1 1.544 Mb/s and E1 2.048 Mb/s digital hierarchies. Downstream data rates depend on a number of factors, including the length of the copper line, its wire gauge, presence of bridged taps, and cross-coupled interference. Line attenuation increases with line length and frequency and decreases as wire diameter increases. Thus, ADSL can perform the data rates indicated in Table 13-5.

Table 13-5. ADSL Data Rates as a function of wire and Distance

Data Rate (Mbps)	Wire Gauge (AWG)	Distance (km)
1.5–2.0	24	6
1.5–2.0	26	4.6
6.1	24	3.7
6.1	26	2.7

In order to create multiple channels, ADSL modems divide the available bandwidth of a telephone line in one of two ways -- Frequency Division Multiplexing (FDM) or Echo Cancellation. FDM assigns one band for upstream data and another band for downstream data. The downstream path is then divided by time division multiplexing into one or more high speed channels and one or more low speed channels. The upstream path is also multiplexed into corresponding low speed channels.

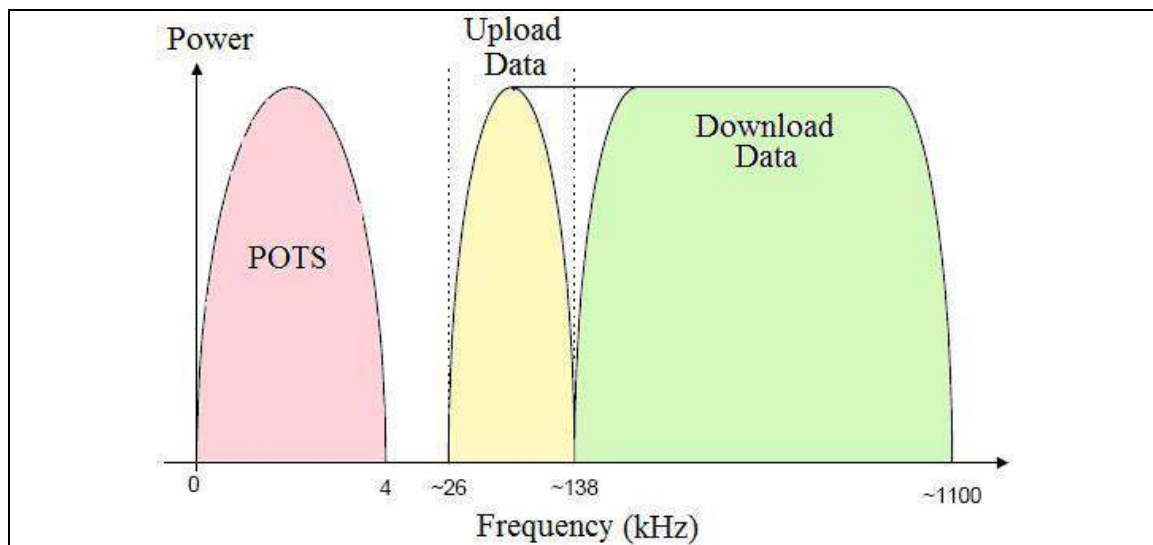


Fig. 13-22. Spectrum of the ADSL.

Echo Cancellation assigns the upstream band to over-lap the downstream, and separates the two by means of local echo cancellation, a technique well known in V.32 and V.34 modems. With either technique, ADSL splits off a 4 kHz region for Plain Old Telephone Service (POTS) at the DC end of the band. As shown in figure 13-14, ADSL transmits two separate data streams with much more bandwidth devoted to the downstream leg to the customer than returning. It is effective because symmetric signals in many pairs within a cable (as occurs in cables coming out of the central office) significantly limit the data rate and possible line length.

13-6.2. ADSL Technology

ADSL depends upon digital signal processing and creative algorithms to squeeze so much information through copper telephone lines. In addition, many advances have been required in transformers, analog filters, and A/D converters. The distance between the subscriber modem and the CO is called loop length. Long telephone lines may attenuate signals at 1MHz (the edge of the band used by ADSL) by as much as 90 dB, forcing analog sections of ADSL modems to work very hard to realize large dynamic ranges, separate channels, and maintain low noise figures.

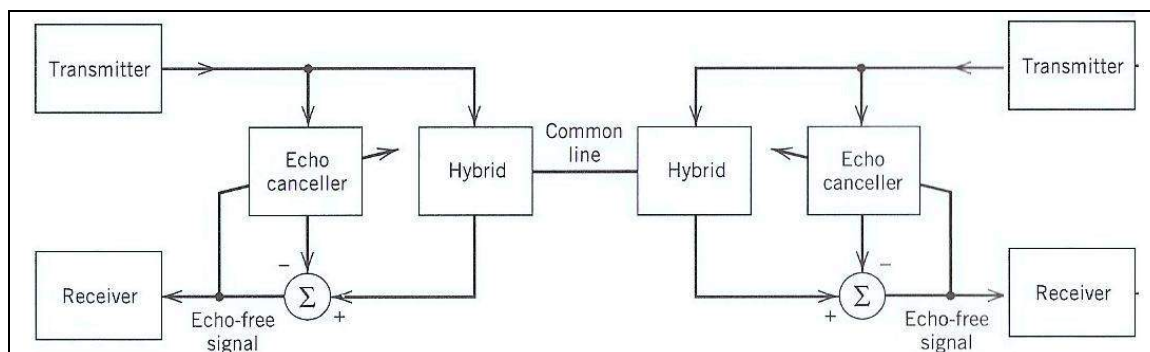


Fig. 13-23. Block diagram of the DSL transceiver.

In order to create multiple channels, ADSL modems divide the available bandwidth of a telephone line in one of two ways -- Frequency Division Multiplexing (FDM) or Echo Cancellation. FDM assigns one band for upstream data and another band for downstream data. The downstream path is then divided by time division multiplexing into one or more high speed channels and one or more low speed channels. The upstream path is also multiplexed into corresponding low speed channels. Echo Cancellation assigns the upstream band to over-lap the downstream, and separates the two by means of local echo cancellation, a technique well known in V.32 and V.34 modems. With either technique, ADSL splits off a 4 kHz region for POTS at the DC end of the band.

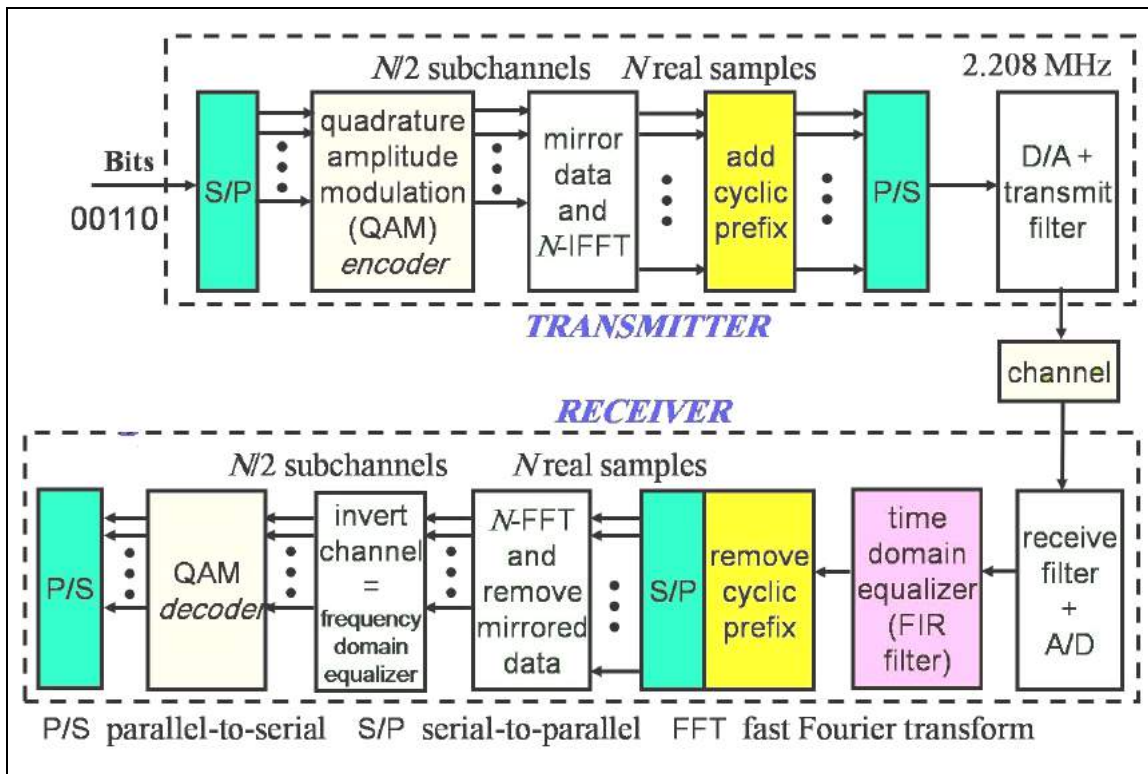


Fig. 13-24. Details of the DSL transceiver.

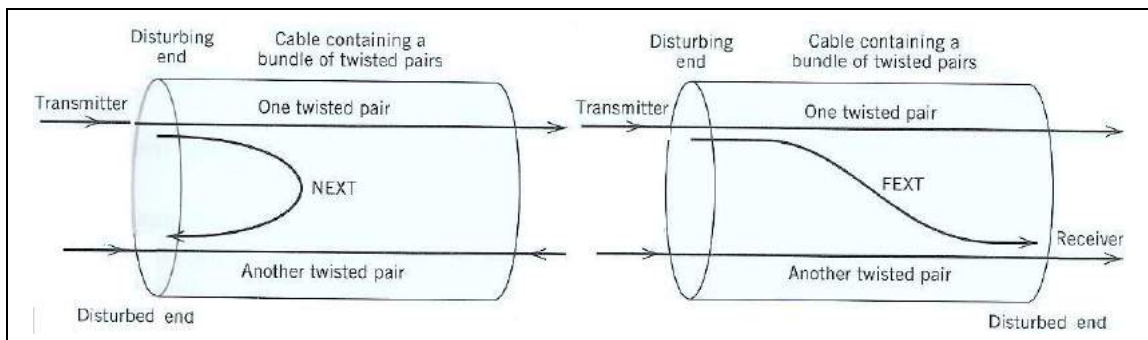


Fig. 13-25. Near and far cross talk.(NEXT & FEXT) in twisted copper wires.

13-6.3. ADSL Wiring and Filters

An ADSL circuit connects an ADSL modem on each end of a twisted-pair telephone line, creating three information channels---a high-speed downstream channel, a medium-speed duplex channel, and a basic telephone service channel. The basic telephone service channel is split off from the digital modem by filters, thus guaranteeing uninterrupted basic telephone service, even if ADSL fails. ADSL uses frequencies very much higher than the speech band (300 Hz to 3.4 kHz) so you finish up with two different systems on the same line. In order to keep these systems apart it is necessary to separate the two components from the telephone line in your home. This is where the Filter/Splitter comes in. It is normally a small plastic box with a short lead that plugs into your phone socket and

two outputs, one for your ADSL Modem and another for a telephone (or multiple telephones on this output, but more of that later). Inside this box are the filters that select the band of frequencies for each of the outputs, phone or ADSL, and send just the correct band to the appropriate socket. Nothing complicated here: just plug the filter / splitter into the phone socket then the modem and phone into the outputs, as shown in the following figure.

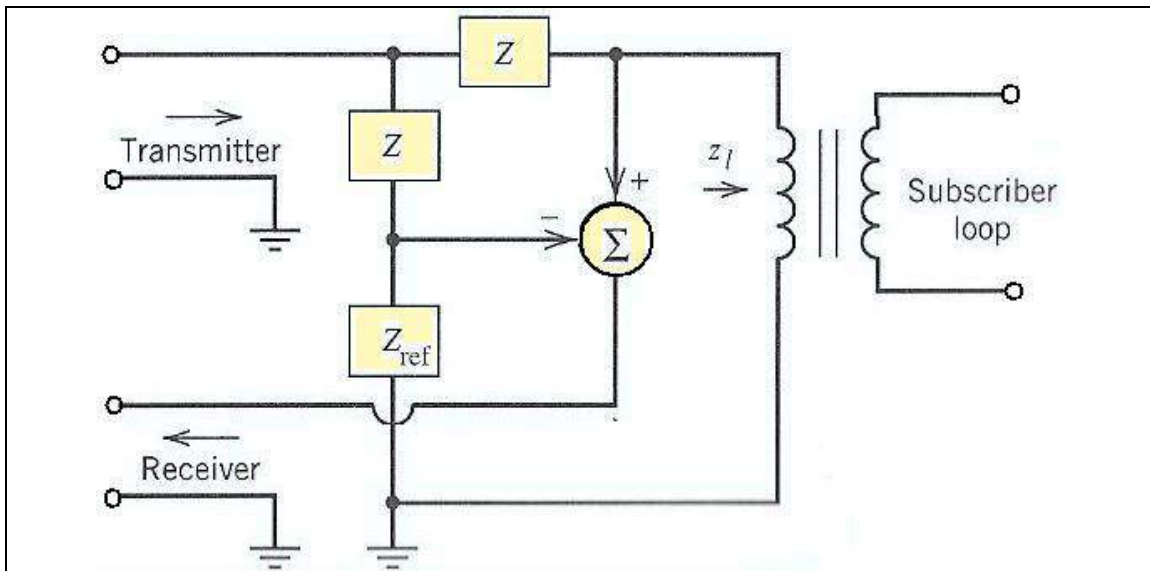


Fig. 13-26. Echo cancellation circuit in DSL.

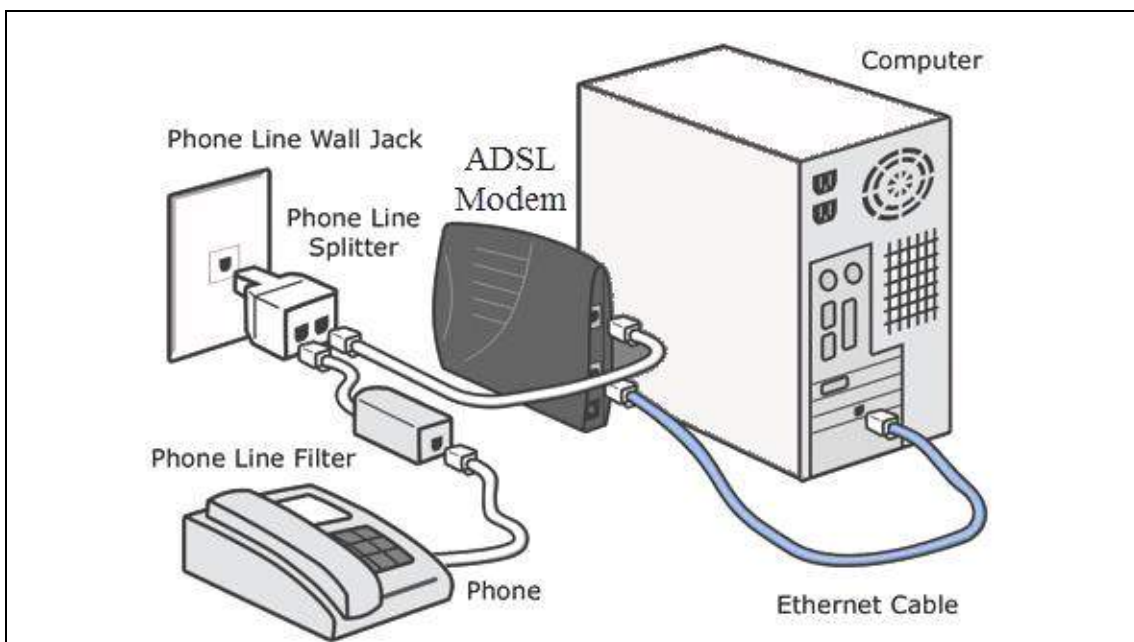


Fig. 13-27. ADSL modem connection to telephone cable and splitter/filter.

13-6.4. xDSL Standards

There are the two basic xDSL standards, namely carrierless amplitude /phase modulation (CAP) and discrete multi-tone (DMT). CAP (carrierless amplitude modulation/phase modulation) is a version of suppressed carrier quadrature amplitude modulation (QAM). For passive NT configurations, CAP would use quadrature phase shift keying (QPSK) upstream and a type of TDMA for multiplexing (although CAP does not preclude an FDM approach to upstream multiplexing). DMT (discrete multitone) is a multicarrier system using discrete fourier transforms to create and demodulate individual carriers. For passive configurations, DMT would use FDM for upstream multiplexing. CAP demonstrated a clear lead in getting product to market. Chips were available in quantity, and they worked. Numerous products that incorporate such chips are installed in a number of locations by service providers. Standards and interoperability issues between vendors and implementations are now being addressed. DMT, on the other hand, has been in the standards arena for some time and continues to evolve. It is now considered a standard by a number of service providers.

13-7. Digital Video Broadcast (DVB) Systems

We mean by video **Broadcasting** the distribution of audio/video programs by digital transmission to an audience (customers). Historically, there have been several different types of electronic broadcasting mediums, such as Radio and TV broadcasting on the air as RF waves from a transmitter antenna to a receiving device. In addition to the classical Radio/TV broadcasting, there have been other forms such as Cable TV, satellite TV and Webcasting of Video/audio programs, over Internet Protocol. Digital Video Broadcasting (**DVB**) is the method used for all-digital TV and Radio broadcasting.

13-7.1. Digital TV Standards

There are several digital TV standards developed and deployed in different countries: ATSC in the US, North America and South Korea, **ISDB-T** in Japan and Brazil, **DVB-T** in Europe and most countries in Africa, Asia and in Australia. You can also receive DVB from satellite (**DVB-S**). Both ISDB-T and DVB-T are based on COFDM modulation that has several variants: QPSK, 16QAM and 64QAM. ATSC is based on 8VSB modulation, which is similar to the vestigial sideband modulation used in analog TV.

All standards use MPEG-2 compression technology for transport of video and audio information in available bandwidth. Demand for more channels and high-definition content forced the development of better compression techniques like MPEG-4 AVC or H.264.

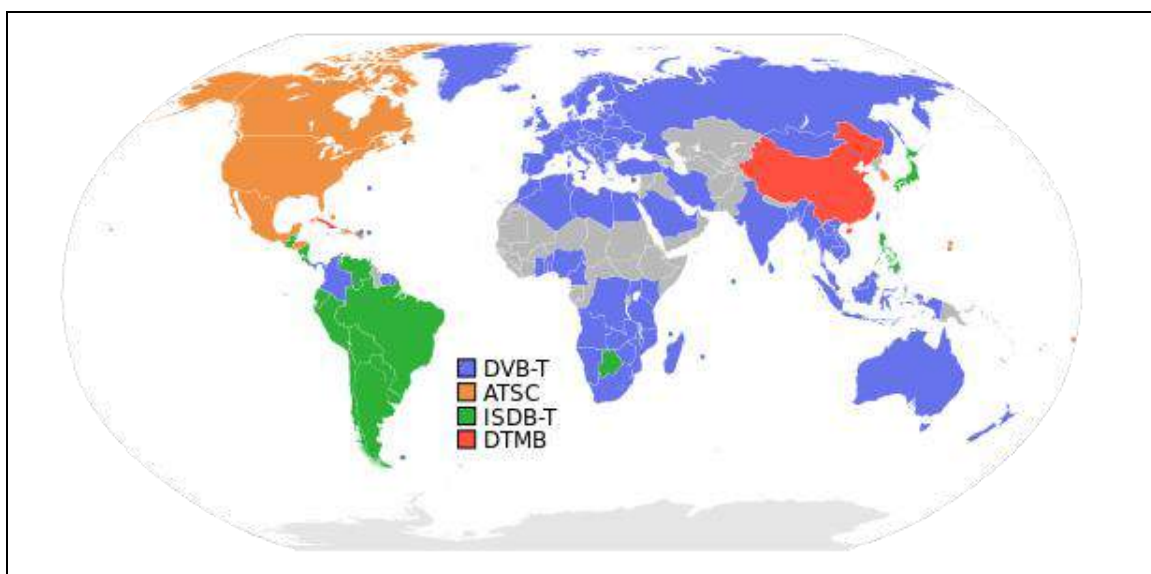


Fig. 13-28. Digital Broadcast-system in the world

13-7.2. Digital Video Broadcast-Terrestrial (DVB-T)

DVB-T or Digital Video Broadcast - Terrestrial is the most widely used digital television standard in use around the globe for terrestrial television transmissions. It provides many facilities and enables a far more efficient use of the available radio frequency spectrum than the previous analogue transmissions. The DVB-T standard was first published in 1997 and since then it has become the most widely used format for broadcast digital in the world. By 2008, it was the standard that was adopted in more than 35 countries and over 60 million receivers deployed and in use. The modulation used in DVB-T is Convolutional Orthogonal Frequency Division Multiplex (COFDM). This provides data payload up to 19 Mbps (depending on type of modulation used: QPSK, 16QAM or 64QAM). With this, it is possible to transmit one HDTV program or up to four standard definition programs in a 6, 7 or 8MHz channel. The used transport streams are MPEG-2 or MPEG-4 AVC. DVB-T can also provide mobile handheld services, data casting and interactive services.

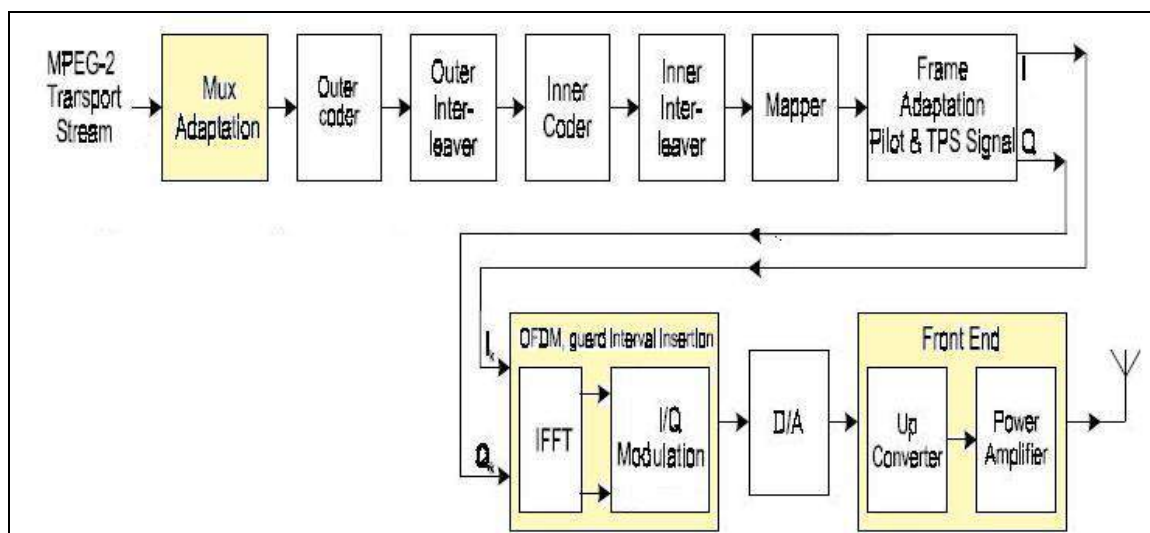


Fig. 13-29. Block diagram of the DVB-T transmitter

COFDM works in this manner: it splits the information stream into several data streams, which then modulates a number of adjacent carrier frequencies within a TV channel (8, 7 or 6 MHz). For DVB-T, there are two modulation schemes in terms of the number of carrier frequencies: 2K (actually 1,705 carriers that are 4 kHz apart) and 8K (actually 6,817 carriers that are 1 kHz apart). DVB-T also employs a "Guard Interval" technique that allows receivers to ignore received data for a short period of time to minimize the effects of multi-path reception. It is this characteristic that gives DVB-T the advantage over ATSC when it comes to multi-path performance making the former more suited for mobile applications.

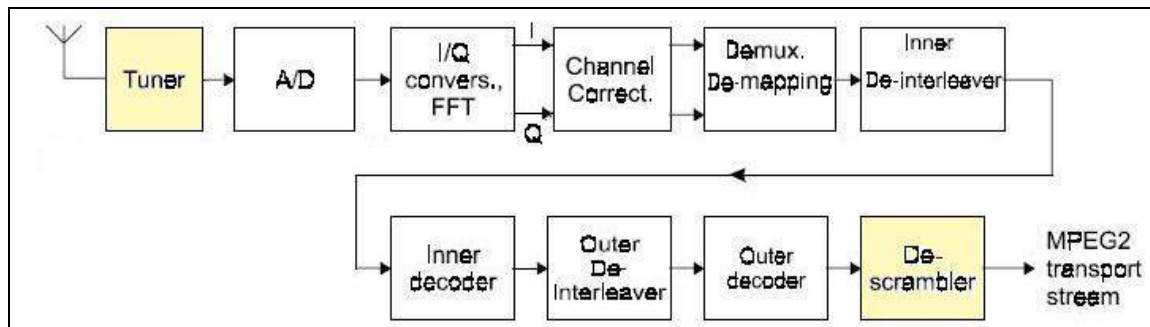


Fig. 13-30. Block diagram of the DVB-T receiver

In brief, the DVB-T network is able to meet the requirements of the operator, by varying the following options:

3 modulation options (QPSK, 16QAM, 64QAM): There is a balance between the amount rate at which data can be transmitted and the signal to noise ratio that can be tolerated. The lower order modulation formats like QPSK do not transmit data as fast as the higher modulation formats such as 64QAM, but they can be received when signal strengths are lower.

5 different FEC (forward error correction) rates: Any radio system transmitting data will suffer errors. In order to correct these errors various forms of error correction are used. The rate at which this is done affects the rate at which the data can be transmitted. The higher the level of error correction that is applied, the greater the level of supporting error correction data that needs to be transmitted. In turn this reduces the data rate of the transmission. Accordingly it is necessary to match the forward error correction level to the requirements of the broadcast network. The error correction uses convolutional coding and Reed Solomon with rates of 1/2, 2/3, 3/4, 5/6, and 7/8 dependent upon the requirements.

4 Guard Interval options:

2k or 8k carriers: According to the transmission requirements the number of carriers within the OFDM signal can be varied. When fewer carriers are used, each carrier must carry high bandwidth for the same multiplex data rate. This has an impact on the resilience to reflections and the spacing between transmitters in single frequency networks. Although the systems are labeled 2k and 8k the actual numbers of carriers used are 1705 carriers for the 2k service and 6817 carriers for the 8k service.

6, 7 or 8MHz channel bandwidths: It is possible to tailor the bandwidth of the transmission to the bandwidth available and the channel separations. Three figures of bandwidth are available.

Video at 50Hz or 60Hz: The refresh rate for the a screen can be varied. Traditionally for analogue televisions this was linked to the frequency used for the local mains supplies. By altering the various parameters of the transmission it is possible for network operators to find the right balance between the robustness of the DVB-T transmission and its capacity..

13-7.3. DVB-T Single Frequency Network (SFN)

One of the advantages of using OFDM as the form of modulation is that it allows the network to implement what is termed a single frequency network. A single frequency network (SFN) is one where a number of transmitters operate on the same frequency without causing interference. Many forms of transmission, including the old analogue television broadcasts would interfere with one another. Therefore when planning a network, adjacent areas could not use the same channels and this greatly increased the amount of spectrum required to cover a country. By using OFDM an SFN can be implemented and this provides a significant degree of spectrum efficiency improvement. A further advantage of using a system such as DVB-T that uses OFDM and allows the implementation of an SFN is that very small transmitters can be used to enhance local coverage. DVB-T is now well established all over the world. Many countries, including Egypt are moving towards a complete switch-over from analogue to digital TV, using the DVB-T. The following table summarizes the main features of the DVB-T

Table 13-6. Main Parameters of the DVB-T.

PARAMETER	DVB-T
Number of carriers in signal	2k, 8k
Modulation formats	QPSK, 16QAM, 64 QAM
Scattered pilots	8% of total
Continual pilots	2.6% of total
Error correction	Convolutional Coding + Reed Solomon 1/2, 2/3, 3/4, 5/6, 7/8
Guard interval	1/4, 1/8, 1/16, 1/32

The so-called **DVB-T2** is the next development of the Digital Video Broadcasting - Terrestrial standards. It builds on the technology and on the success of DVB-T to provide additional facilities and features in line with the developing DTT or Digital Terrestrial television market.

13-7.4. Digital Video Broadcast-Handheld (DVB-H)

DVB-H or Digital Video Broadcast - Handheld, is one of the major systems to be used for mobile video and television for cellular phones and handsets. DVB-H has been developed from the DVB-T television standard that is used in many countries around the globe including much of Europe and Egypt, and also other countries including the USA. Therefore, extensive research has been recently dedicated for the DVB-H for mobile phones and hand-held devices.

The DVB-M committee has been established on 2002 for this purpose. DVB-H aims at providing digital TV reception to mobile devices. It combines traditional TV broadcast standards with elements specific to handheld devices. It also has the following features, smaller screen, mobility, small antennas, indoor coverage and reliance on battery power. The transmission of data mainly done as IP frames. The stream rate is approximately 390 kb/s, with no video compression. Main difference between DVB-T and DVB-H

- DVB-H does not utilize all the service information table defined by DVB
- DVB-H uses IP based Information system, service

There are a variety of modes in which the DVB-H signal can be configured. These are conform to the same concepts as those used by DVB-T. These are 2K, 4K, and 8K modes, each having a different number of carriers as defined in the table below. The following table depicts the main parameters of the DVB-H

Table 13-7. Main Parameters of the DVB-H.

PARAMETER	2K MODE	4K MODE	8K MODE
Number of active carriers	1705	3409	6817
Number of data carriers	1512	3024	6048
Individual carrier spacing	4464 Hz	2232 Hz	1116 Hz
Channel width	7.61 MHz	7.61 MHz	7.61 MHz

One of the key requirements for any mobile TV system is that it should not give rise to undue battery drain. Mobile handset users are used to battery lifetimes extending over several days, and although battery technology is improving, the basic mobile TV technology should ensure that battery drain is minimized.

There is a module within the standard and hence the software that enables the receiver to decode only the required service and shut off during the other service bits. It operates in such a way that it enables the receiver power consumption to be reduced while also offering an uninterrupted service for the required functions. The time slicing elements of DVB-H enable the power consumption of the mobile TV receiver to be reduced by 90% when compared to a system not using this technique. Although the receiver will add some additional power drain on the battery, this will not be nearly as much as it would have been had the TV reception scheme not employed the time slicing techniques.

DVB-H is making use of the so-called Interleaving technique. Interleaving is a technique where sequential data words or packets are spread across several transmitted data bursts. In this way, if one transmitted burst or proportion of the data in each original word or packet is lost and it can be reconstructed using the error detection and correction techniques employed.

Further levels of interleaving have been introduced into DVB-H beyond those used for DVB-T. The basic mode of interleaving used on DVB-T and which is also available for DVB-H is a native interleaver that interleaves bits over one OFDM symbol. However DVB-H provides a more in-depth interleaver that interleaves bits over two OFDM symbols (for 4K mode) and four bits (for 2K mode).

Using the in-depth interleaver enables the noise resilience performance of the 2K and 4K modes to be brought up to the performance of the 8K mode and it also improves the robustness of the reception of the transmissions in a mobile environment.

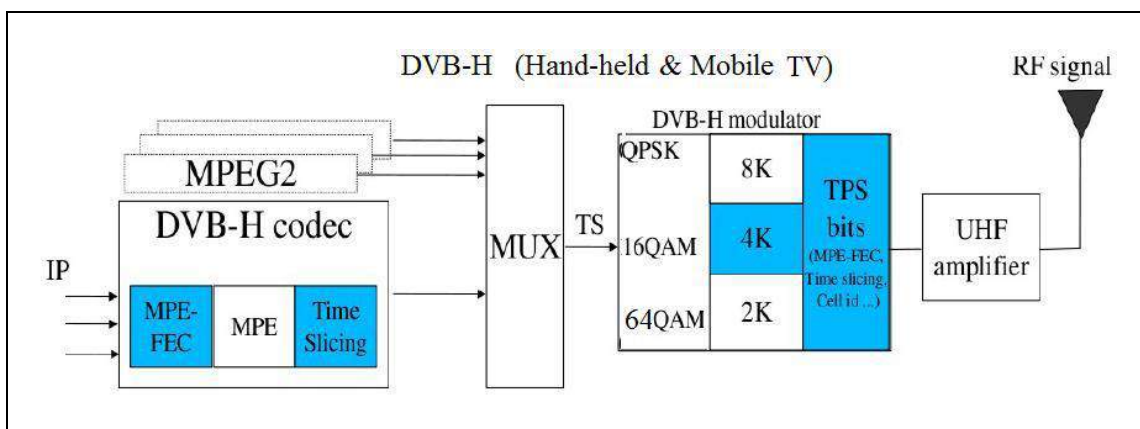


Fig. 13-31. Block diagram of a DVB-H Receiver

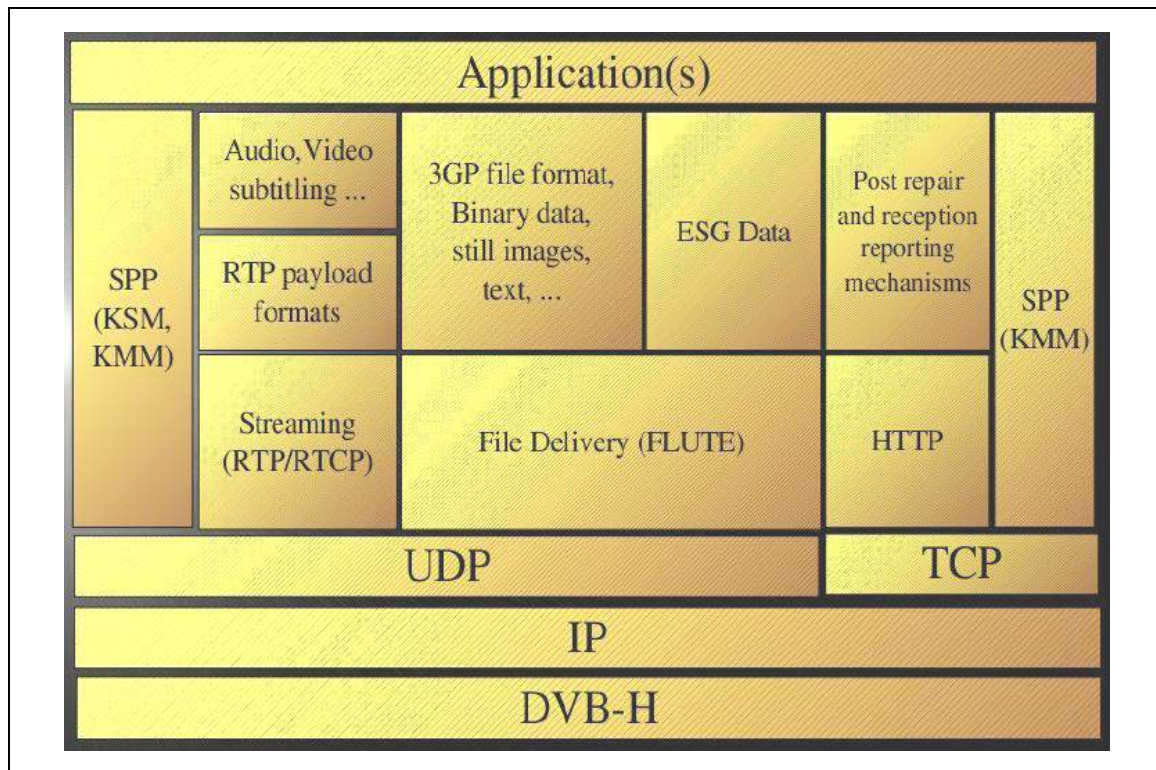


Fig. 13-32. DVB-H protocol stack

13-7.5. Digital Video Broadcast-Satellite to Handheld (DVB-SH)

DVB-SH, Digital Video Broadcast - Satellite services to Hand-held devices is a standard or specification that is likely to be widely used for Mobile TV services. The DVB-SH standard has been developed to deliver video, audio and data services to small handheld devices including mobile phones and PDAs and using frequencies typically within S band but in any case below 3 GHz from either satellite or terrestrial networks. DVB-SH has also been designed to complement DVB-H which is focused on delivering mobile video from terrestrial networks at frequencies within the UHF TV bands. One of the key features of DVB-SH is that it is aimed for use for both satellite and terrestrial delivery. This is a significant advantage because it allows satellite delivery to achieve coverage of large areas of a country and then terrestrial coverage can be used for gap fillers for example in built up areas in cities where tall buildings may shield the satellite signal. In view of its specifications, DVB-SH is will be used alongside other forms of cellular technology as it is estimated that most applications will be in small mobile devices such as cell phones, PDAs, etc. The fact that DVB-SH is focussed on frequencies up to 3 GHz, and expected to operate on frequencies around 2.2 GHz means that the performance requirements for the system are more exacting than those for DVB-H where frequencies up to 900MHz are typically used. Typically it

is found that the signal to noise ratios are inferior, and this could result in high levels of bit error rate and poor performance unless power levels were raised and dense terrestrial networks employed. As these options are not desirable, enhancements have been included in the signal processing areas of the DVB-SH standard.

A state of the art forward error correction system has been included in the form of the 3GPP2 turbo-code. In addition to this, the standard includes a highly effective channel interleaver. This offers time diversity of between 100 ms up to several seconds dependent upon the targeted service levels and also the amount of memory available in the target receivers. By interleaving, the effects of interference can be minimized, and the longer the period of interleaving, the greater the interference duration that can be tolerated. In addition to this, pilot symbols are used to provide a robust form of signal estimation and fast re-acquisition. This provides considerable performance improvements when there are long shadowing or signal blockages. This scheme is used for both TDM and OFDM modes.

13-8. Optical Communication Links

Optical communication is a method of transmitting information from one place to another by sending pulses of light through an optical fiber or free space. Optical fiber can be used as a medium for telecommunication and networking because it is flexible and can be bundled as cables. It is especially advantageous for long-distance communications, because light propagates through the fiber with little attenuation compared to electrical cables or free space. This allows long distances to be spanned with few repeaters. The wave division multiplexing (**WDM**) is usually employed in optical communication networks to multiplex several optical signals on a single fiber by using different wavelengths (colors) of laser light. This allows for a multiplication in capacity and enables bidirectional communications over the same fiber. The following figure depicts a complete optical communication system

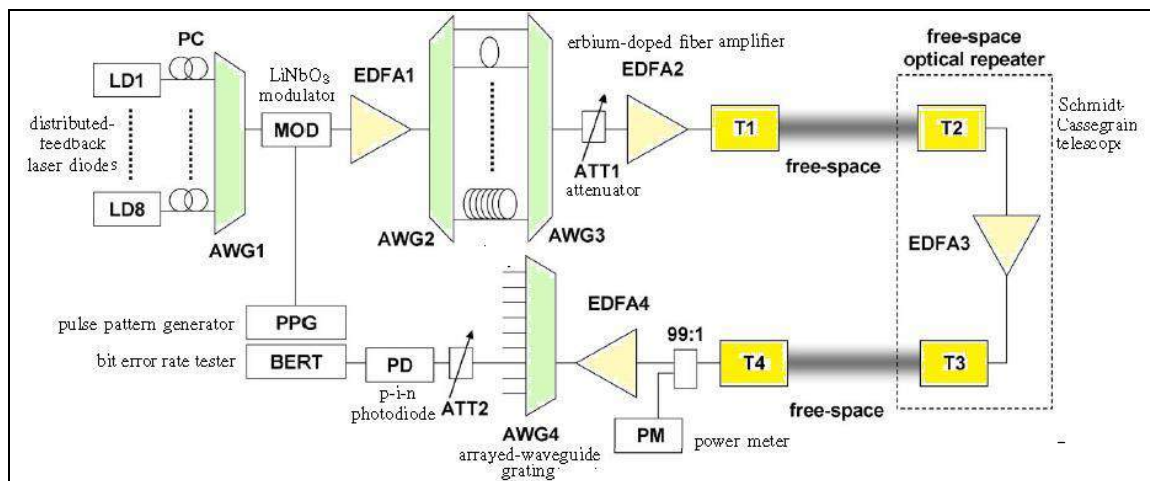


Fig. 13-33. Experimental setup of an optical communication system.

Note: 13-3: SDH, SONET, WDM & DWDM

The synchronous digital hierarchy (**SDH**) was developed as a standard for multiplexing of high order data frames and has become the primary transmission protocol in most PSTN networks. SDH Network functions are often connected using high-speed Optic Fiber networks, such as **SONET** (synchronous optical network). However, modern optic fiber transmission makes use of Wavelength Division Multiplexing (**WDM**) where signals transmitted across the fiber are transmitted at different wavelengths, creating additional channels for transmission. Actually, WDM is a form of frequency division multiplexing (**FDM**) but it is commonly called wavelength division multiplexing.

Nowadays, the so called dense wavelength division multiplexing (**DWDM**), is becoming more popular in optical fiber networks and started to replace the SONET/SDH technology. DWDM refers originally to optical signals multiplexed within the 1550-nm band so as to leverage the capabilities (and cost) of erbium doped fiber amplifiers (**EDFAs**), which are effective for wavelengths between approximately 1525-1565 nm (C band), or 1570-1610 nm (L band).

13-8.1. Optical Link Power Budget

The **optical power budget** in a fiber-optical link is the allocation of available optical power (launched into a given fiber by a given source) among various loss-producing mechanisms, in order to ensure that adequate signal strength (optical power) is available at the receiver. The loss producing devices may be launch coupling loss, fiber attenuation, splice losses, and connector losses.

The amount of optical power launched into a given fiber by a given transmitter depends on the nature of its active optical source (LED or laser diode) and the type of fiber, including such parameters as core diameter and numerical aperture. Manufacturers sometimes specify an optical power budget only for a fiber that is optimum for their equipment—or specify only that their equipment will operate over a given distance, without mentioning the fiber characteristics. Figure 13-28 depicts a simple laboratory optical link, with wave-length division multiplex (**WDM**) of two channels, and Table 13-34 depicts its power budget, via the two channels. The first channel has a light emitting diode (**LED**) source and the second has an injection laser diode (**ILD**).

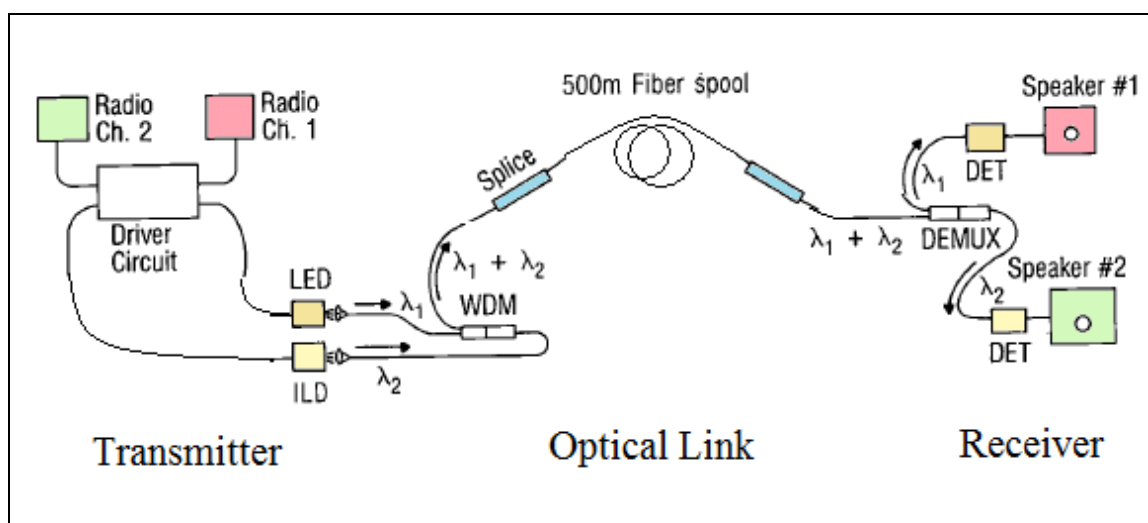


Fig. 13-34. Optical Link example

The user must first ascertain, from the manufacturer or by testing, the transmission losses for the type of fiber to be used, and the required signal strength for a given level of performance. In addition to transmission loss, including those of any splices and connectors, allowance should be made for at least several dB of optical power margin. This margin is necessary to compensate for component aging and to allow for future splices in the event of damaged cable.

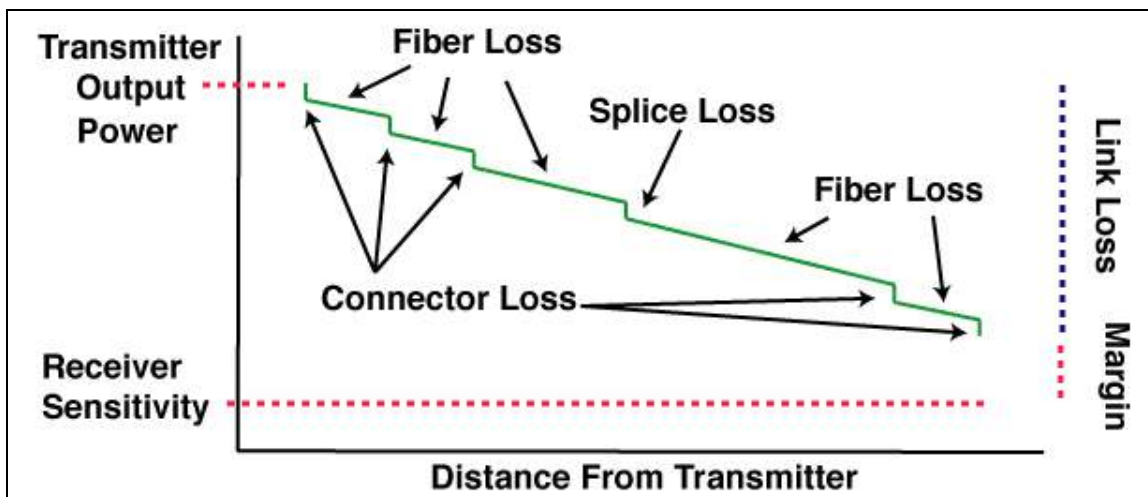


Fig. 13-35. Optical Link example

13-8.2. Practical Optical Link Budget

The following general rules apply to the optical link loss budget for practical data channels. The power loss between the laser source and receiver must not exceed 38 (6 - (-32)) dB or the signal will not be detected accurately. There must be at least 15 (7 - (-8)) dB of attenuation between neighboring nodes to avoid saturating the receiver. In order to validate a network design, the optical link loss must be calculated for each band of channels. At a minimum, any data channel path calculation must include line card transmit loss, channel add loss, fiber loss, channel drop loss, and line card receive loss. In ring topologies, pass-through add losses and pass-through drop losses must be considered. Losses due to external devices also need to be included.

Table 13-8. Optical link budget

	Source	
	LED	ILD
Power from source	7.0 dBm	0.0 dBm

Table 13-8. (Cont.) Optical link budget

Losses:	Source	
	LED	ILD
Source-fiber connection	18 dB	4.0 dB
WDM filter	1.5 dB	1.5 dB
WDM fiber connector	0.5 dB	0.5 dB
Fiber transmission cable (500m)	1.8 dB	1.8 dB
Fiber DEMUX connection	0.5 dB	0.5 dB
DEMUX filter	0.5 dB	0.5 dB
Fiber detector	2.0 dB	2.0 dB
Total loss	24.8 dB	9.8 dB
	-----	-----
Detected Power	-17.8 dBm	-9.8 dBm

13-9. Optical Networks

The rapid advancement and evolution of optical technologies makes it possible to move beyond point-to-point WDM transmission systems to an all-optical backbone network that can take full advantage of the available bandwidth. Such a network consists of a number of optical cross-connects (**OXC**s) arranged in some arbitrary topology. Each OXC can switch the optical signal coming in on a wavelength of an input fiber link to the same wavelength in an output fiber link. The OXC may also be equipped with converters that permit it to switch the optical signal from one wavelength to another. Currently, there is tremendous interest in optical networks in which OXC's provide the switching functionality.

The Internet Engineering Task Force (**IETF**) is investigating the use of Generalized MPLS (GMPLS) and related signaling protocols to set up and divide light-paths. GMPLS is an extension of MPLS that supports multiple types of switching, including switching based on wavelengths usually referred to as Multi-Protocol Lambda Switching (MP λ S).

With GMPLS, the OXC backbone and the IP/MPLS sub-networks will share common functionality in the control plane, making it possible to seamlessly integrate all-optical networks within the overall Internet infrastructure. GMPLS extends the label switching architecture proposed in MPLS to other types of non-packet based networks, such as SONET/SDH based networks and wavelength-routed networks. Specifically, the GMPLS architecture supports the following types of switching: packet switching (IP, ATM, and frame relay), wavelength switching in a wavelength-routed network, port or fiber switching in a wavelength-routed network, and time slot switching for a SONET/SDH cross-connect.

13-9.1. Optical Network Architecture

The architecture for optical networks that is widely expected to form the basis for a future all-optical infrastructure is built on the concept of *wavelength routing*. A wavelength routing network, shown in Figure 1, consists of *optical cross-connects (OXC*s) connected by a set of fiber links to form an arbitrary mesh topology. As shown in figure, each OXC in the mesh optical network has an associated electronic control unit attached to one of its input/output ports. The control unit is responsible of the control and management functions related to setting up and dividing light-paths. Client sub-networks attach to the optical network via edge nodes which provide the interface between non-optical devices and the optical core. This interface is called user-to-network interface (**UNI**).

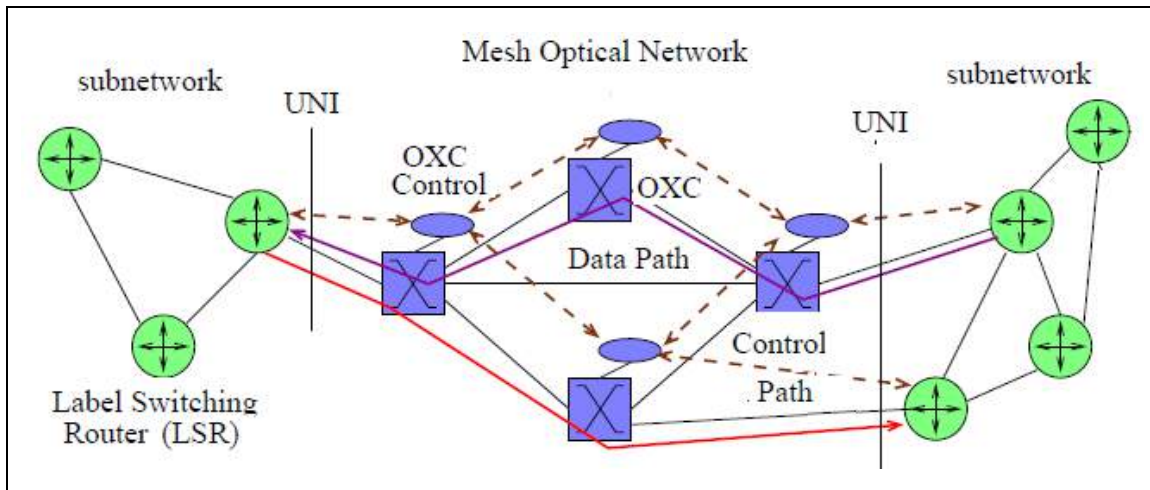


Fig. 13-36. Optical network architecture

The OXCs provide the switching and routing functions for supporting the logical data connections between client sub-networks. An OXC takes in an optical signal at each of the wavelengths at an input port, and can switch it to a particular output port, independent of the other wavelengths. An OXC with N input and N output ports capable of handling W wavelengths per port can be thought of as W independent $N \times N$ optical switches. These switches have to be preceded by a wavelength de-multiplexer and followed by a wavelength multiplexer to implement an OXC, as shown in the following figure.

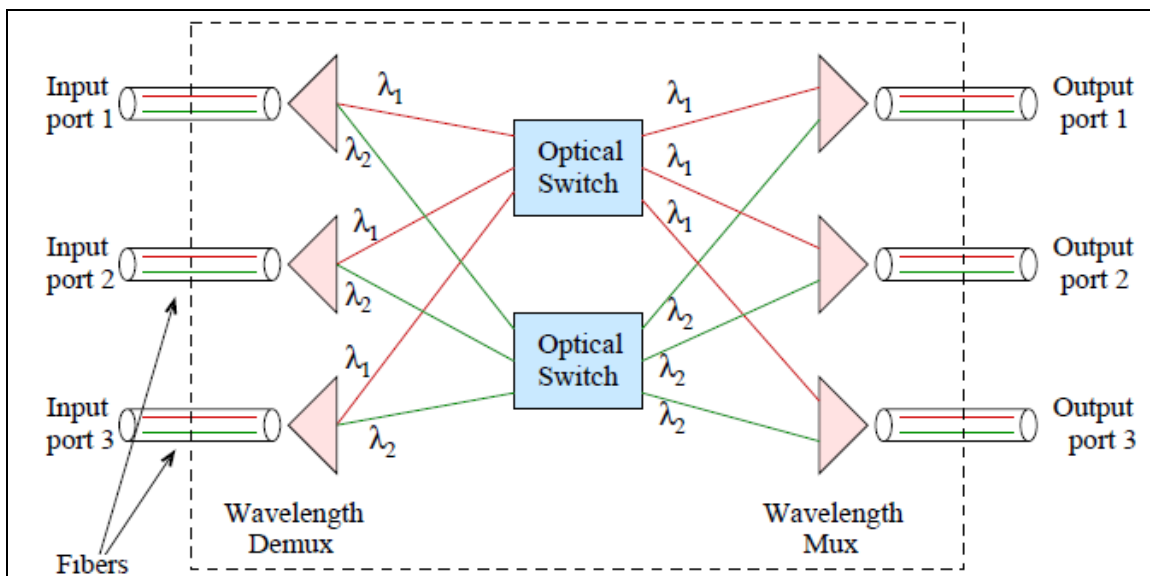


Fig. 13-37. Optical network architecture

An attractive feature of light-trees is the inherent capability for performing multicasting in the optical domain. Such wavelength routed light-trees are

useful for transporting high-bandwidth, real-time applications such as on-chip signaling and high-definition TV (HDTV).

13-9.2. Fiber Distributed Data Interface (FDDI)

Fiber Distributed Data Interface (FDDI) is a standard for data transmission in a local area network (LAN). It uses optical fiber as its standard underlying physical medium. FDDI provides a 100 Mbit/s optical standard for data transmission in local area network that can extend in range up to 200 kilometers. A FDDI network contains two rings, one as a secondary backup in case the primary ring fails.

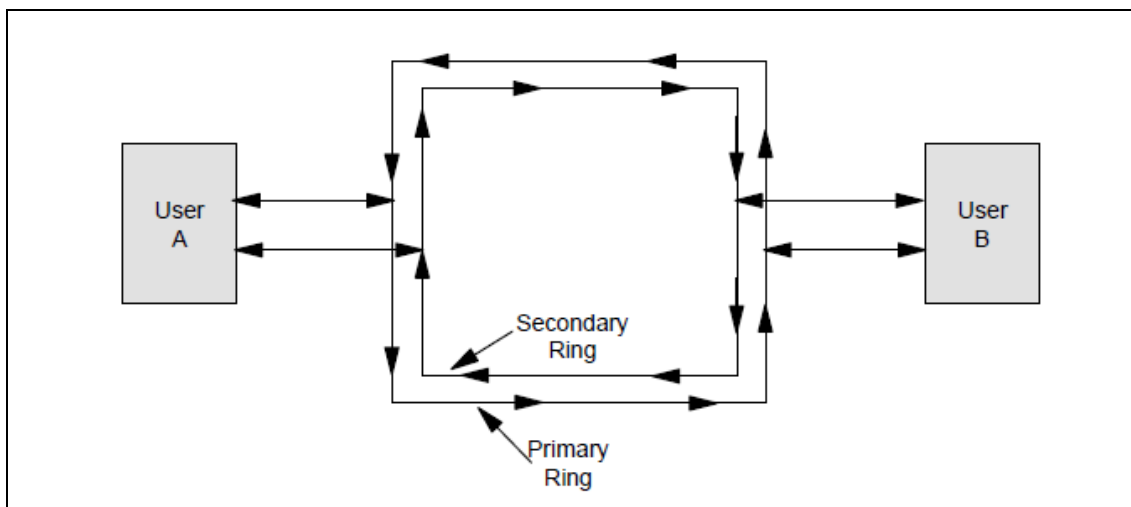


Fig. 13-38. FDDI

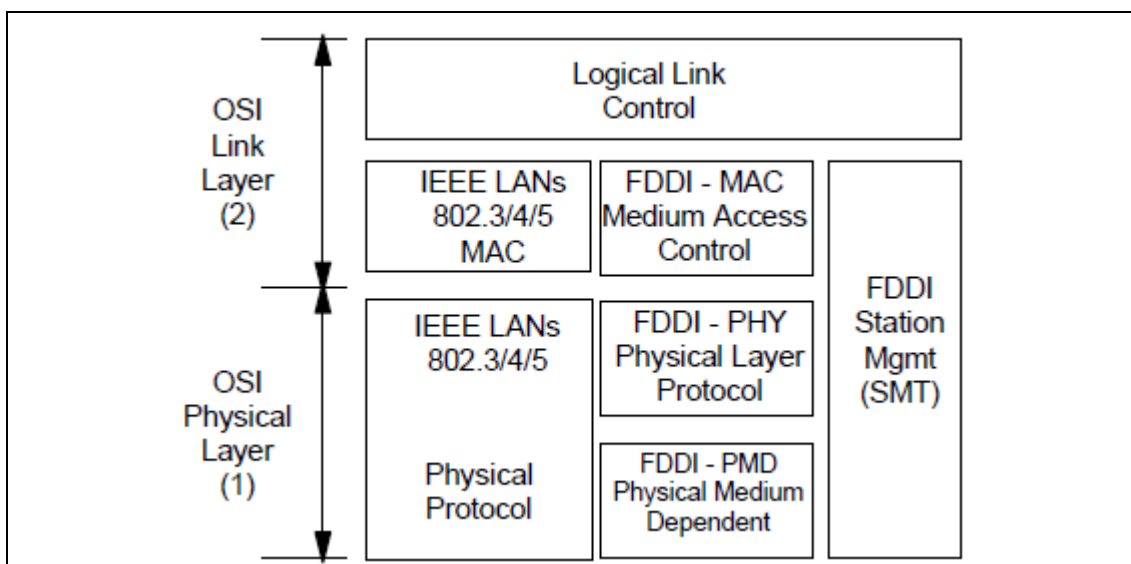


Fig. 13-39. FDDI reference model

FDDI frame structure.	
Field	Description
Preamble	Special bit pattern for synchronization purposes.
Start Delimiter	Marks the beginning of frame.
Frame Control	Denotes the frame type.
Addresses	Source and destination addresses.
Data	Actual user data (MAC or LLC).
FCS	Frame Check Sequence.
End Delimiter	Marks the end of frame.
Frame Status	Used by receiving station to record receipt status.

Fig. 13-40. FDDI frame structure.

13-9.3. Fiber to the Premises (FTTP) Technology

The telecommunications landscape has matured to a point that service providers seek to offer network convergence and enable the revolution of consumer media device interaction. These demands are being met by a deeper penetration of optical fiber access networks. Fiber to the premises (FTTP) is a new optical network access standard. FTTP is a form of fiber-optic communication delivery in which an optical fiber is run directly onto the customers' premises. This contrasts with other fiber-optic communication delivery strategies (termed **FTTx**) such as fiber to the node (FTTN), fiber to the curb (FTTC), or hybrid fiber-coaxial (HFC), all of which depend upon more traditional methods such as copper wires or coaxial cable for last mile delivery.

The optical network terminal (**ONT**) is sometimes called optical network unit (**ONU**). Actually, ONT is an ITU-T term, whereas ONU is an IEEE term, but the two terms mean exactly the same thing. Fiber to the premises can be further categorized according to where the optical fiber ends, namely:

FTTH and FTTB. Fiber to the home (**FTTH**) is a form of fiber optic communication delivery in which the optical signal reaches the end user's living or office space. Fiber to the building (**FTTB**) is a form of fiber optic communication delivery in which the optical signal reaches the private property enclosing the home or business of the subscriber or set of subscribers, but where the optical fiber terminates before reaching the home living space or business office space, with the path extended from that point up to the user's space over a physical medium other than optical fiber (e.g., copper line loops). The following figure depicts the different FTTx distribution strategies.

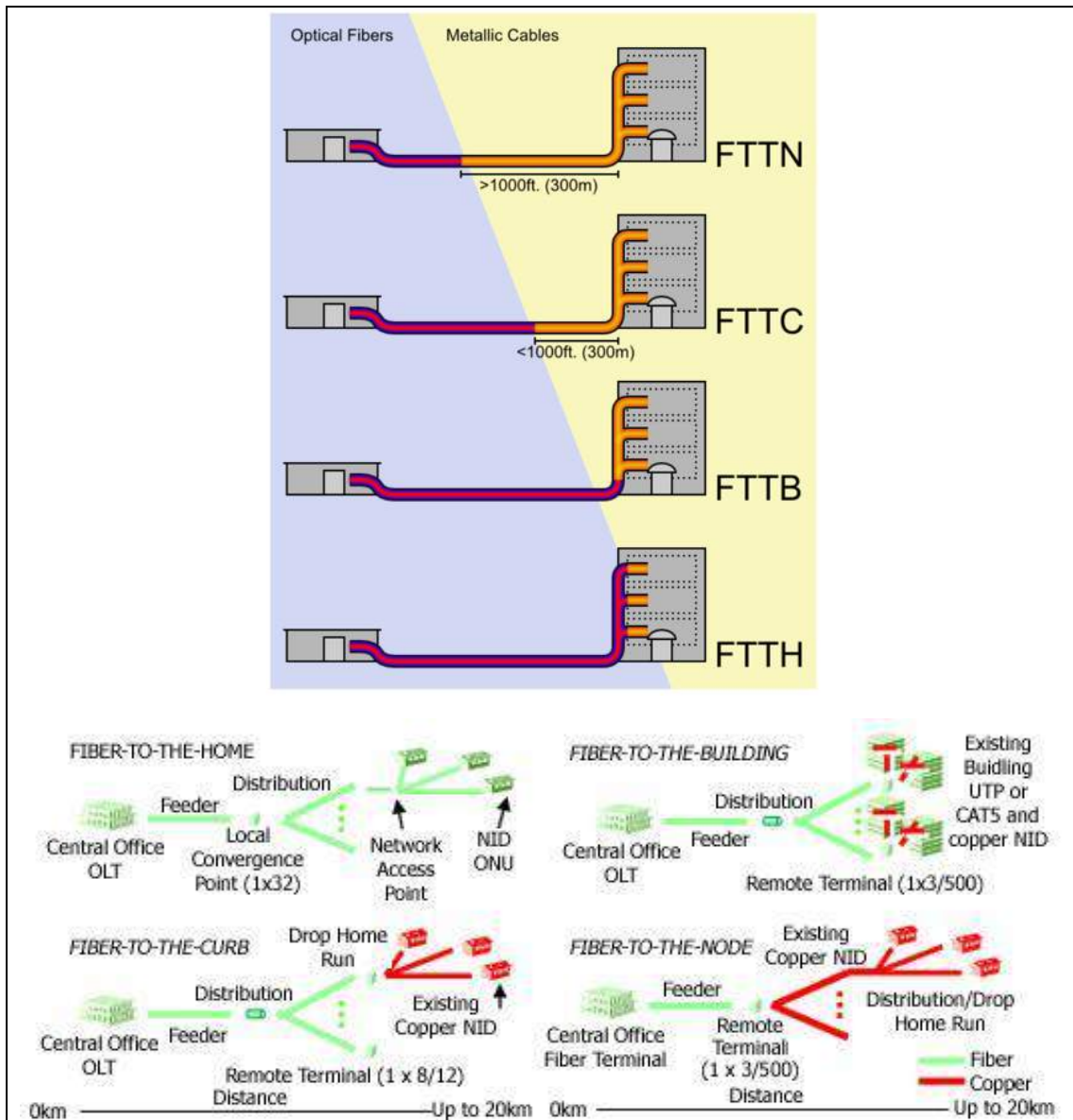


Fig. 13-41. Different forms of the fiber to the x (FTTx) optical communication links.

The simplest optical distribution network can be called direct fiber. In this architecture, each fiber leaving the central office goes to exactly one customer. Such networks can provide high bandwidth since each customer has his own dedicated fiber extending all the way from the central office. However, this approach is more costly due to the amount of fiber and central office machinery required. More commonly, each fiber leaving the central office is actually shared by many customers. It is not until such a fiber gets relatively close to the customers that it is split into individual customer-specific fibers.

There are two competing optical distribution network architectures which achieve this split: active optical networks (**AON**) and passive optical networks (**PON**). Active optical networks rely on some equipment to distribute the signal, such as a switch, router, or multiplexer. Each signal leaving the central office is directed only to the customer for whom it is intended. Incoming signals from the customers avoid colliding at the intersection because the powered equipment there provides buffering. Incoming signals from the customers avoid colliding at the intersection because the powered equipment there provides buffering.

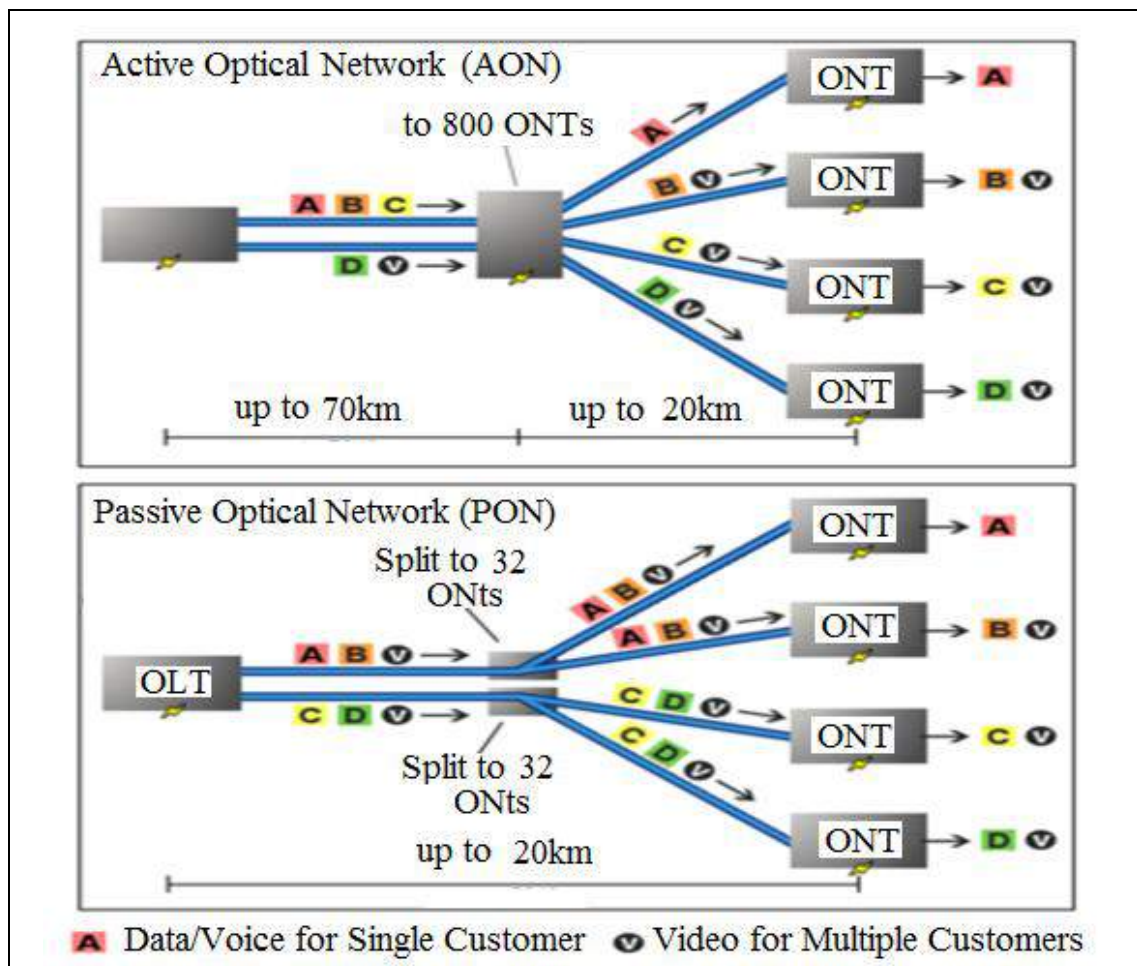


Fig. 13-42. Active optical network (AON) and passive optical network (PON).

The most common type of active optical networks is called **Active Ethernet**. Active Ethernet uses optical Ethernet switches to distribute the signal, thus incorporating the customers' premises and the central office into one giant Ethernet network. Such networks are identical to the Ethernet computer networks used in businesses and academic institutions, except that their purpose is to connect homes and buildings to a central office. Each switching cabinet can handle up to 1000 customers, or more. The IEEE 802.3ah standard enables service providers to deliver up to 100 Mb/s full-duplex over one single-mode optical fiber to the premises.

Passive optical network (PON) is a point-to-multipoint, fiber to the premises network architecture in which unpowered optical splitters are used to enable a single optical fiber to serve multiple premises, typically 32-128. A PON configuration reduces the amount of fiber and central office equipment required compared with point to point architectures.

13-9.4. Fiber to the Home (FTTH)

Communication service carriers try to maximize the number of services they offer to customers via a bundled offering. Technologies such as voice over IP (VoIP), IP television (IPTV), and broadband are becoming commonplace across the society. As bundled services and technologies are deployed, carriers are realizing that their original copper cable networks, which were designed to deliver a single service, are stressed and in many cases incapable of offering the desired services. In addition, after about three years, service carriers will need the capability of more than 40 Mb/s as multiple services are used in the home, faster internet access and high-definition TV (HDTV) becomes more prevalent. Leading this wave is the deployment of single-mode optical fiber deeper into the access networks to curb the high bandwidth requirements of customers. Nowadays, the fiber to the home (FTTH) is the fastest growing global broadband technology. FTTH is commonly deployed in two specific configurations, as shown in figure 13-42. In the first one, fiber is dedicated to each user in the access network. This is called a point-to-point (**PTP**) network. In the second, one fiber is shared (via a power splitter) among a set number of users, typically between 16 and 32. This is called a passive optical network (**PON**). PTP networks are characterized by the use of one fiber and laser per user. They are the simplest FTTH networks to design. PTP networks are sometimes referred to as all-optical Ethernet networks (**AOEN**). Typically a PON is capable of reaching subscribers 20km from the transmitter. Figure 13-43 depicts the different components of the PON. PON is supported by a set of standards, such as broadband PON (**BPON**), Ethernet PON (**EPON**) and Gigabit PON (**GPON**).

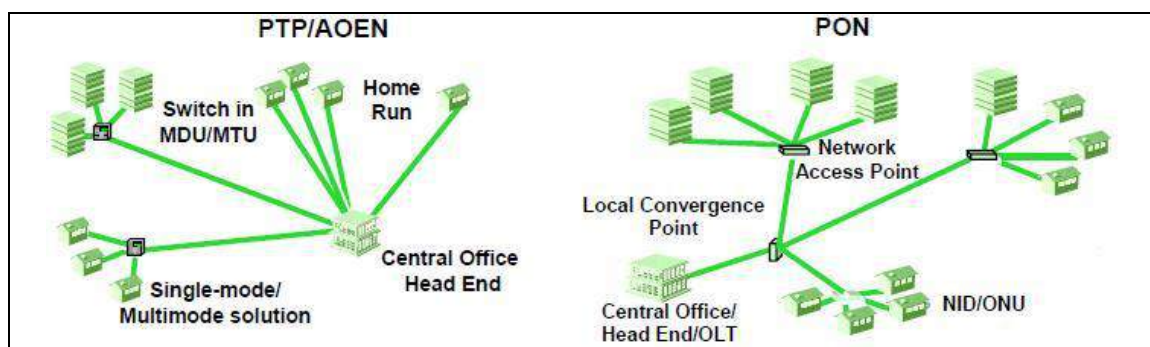


Fig. 13-43. Illustration of the PTP and PON forms of FTTH.

Transmission standards utilized in FTTH networks are based on ATM and Ethernet technologies. Carriers are extremely familiar with both technologies, which support a variety of services. Today, most PTP networks use Ethernet technology and are governed under Institute of Electrical and Electronics Engineers (IEEE) 803.2ah standards. PTP networks are simply an extension of legacy Ethernet used in metropolitan and enterprise networks. Bandwidth rates are only limited to the transmitter type at the CO and the home. Till now, the majority of municipally owned FTTH networks in Japan utilized PTP networks.

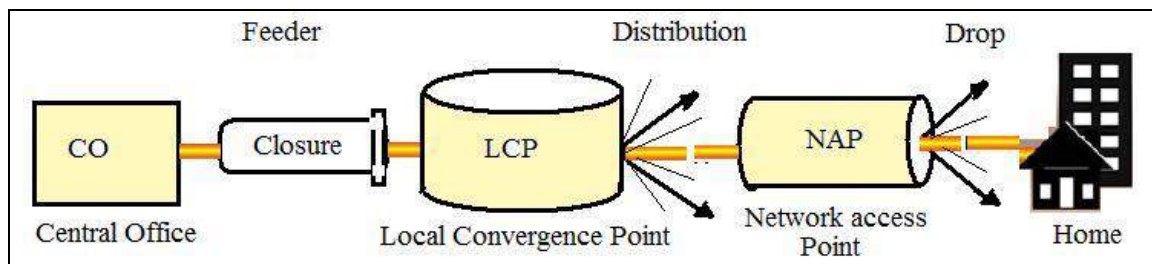


Fig. 13-44. Illustration of the PON components

Table. 13-9. Different types of xPON and their characteristics.

	BPON	EPON	GPON
Standard	ITU-T G983	IEEE 803.2ah	ITU-T G984
Bandwidth	Downstream ~ 622Mb/s Upstream 155Mb/s	Up to symmetric 1.25 Gb/s	Downstream ~ 2.5Gb/s Upstream 155Mb/s
Downlink λ (nm)	1490 and 1550	1550	1490 and 1550
Uplink λ (nm)	1310	1310	1310
Transmission	ATM	Ethernet	ATM, Ethernet

13-10. Summary

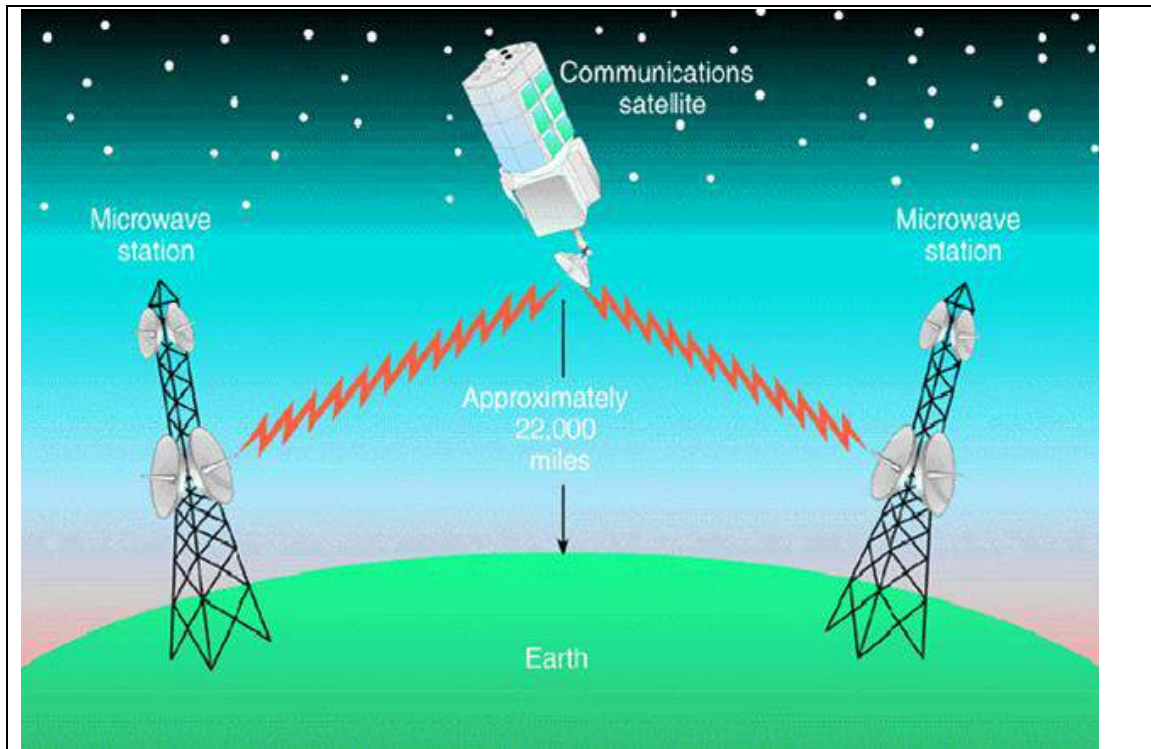
In this chapter we demonstrate the main features of modern communication systems, with emphasis on their link budgets. The communication link budget is a calculation involving the gain and loss factors associated with transmission and reception equipment, the antennas, transmission lines and propagation environment, to determine the maximum distance at which a transmitter and receiver can successfully operate. Alternatively, if the link distance is known, one can calculate the required transmitter power or the receiver sensitivity. A complete radio link budget is simply the sum of all contributions (in dB) across the three main parts of a transmission path. All positive values are gain and all negative values are losses. The radio receiver Sensitivity is then given by:

Margin Receiver Sensitivity	[dBm]	=
Transmitter power	[dBm]	
+Antenna TX gain	[dBi]	
+Antenna RX gain	[dBi]	
– Cable TX Loss	[dB]	
– Cable RX Loss	[dB]	
– Free-space Path Loss	[dB]	

Old telephones were connected directly together in pairs. In modern times, most telephones are connected to public switching telephone networks (**PSTN**).

Digital Subscriber Line (**DSL**) technology is a modem technology that uses existing twisted-pair telephone lines to transport high-bandwidth data, such as multimedia and video, to subscribers. DSL service can support high-speed data transmission over a local loop ranging up to 6 km from the central office (CO). As with most xDSL services, data rates will increase near the CO and taper off with distance. One other important thing to consider when discussing xDSL is that top speeds and distances are almost always expressed as theoretical maximums, assuming ideal line conditions. ADSL depends on advanced digital signal processing and creative algorithms to squeeze so much information through twisted-pair telephone lines. xDSL, a term that encompasses the broad range of digital subscriber line (DSL) service options, has the potential to revolutionize the areas of Internet access and telecommuting by offering a low-cost, high-speed data transport via telephone copper wires. Nowadays, Voice over IP (VoIP) is beginning to emerge as a replacement for traditional switched analog voice services.

The satellite communication services are used for many different purposes, such as voice and data transmission, radio and television broadcast, maritime and aeronautical communications. It had been theoretically demonstrated during the fifties that an object at about 36,000 km above the earth would rotate at the same speed, and therefore appear stationary.



The telecommunications landscape has matured to a point that carriers seek to offer network convergence and enable the revolution of consumer media device interaction. These demands are being met by the deeper penetration of optical fiber in access networks and increasing deployment of fiber-to-the-home (**FTTH**).

13-11. Problems

13-1) Show how to estimate the feasibility of a 5km radio communication link, with one access point (*AP*) and one client radio. The access point is connected to an omni-directional antenna with 10dBi gain, while the client is connected to an antenna with 14dBi gain. The transmitting power of the *AP* is 100mW and its sensitivity is -89dBm. The client transmitting power is 30m and its sensitivity is -82dBm. The cables loss is 2dB at each side.

Hint: Calculate the total path loss and subtract it from the total gain, and show that the difference is greater or equal to the receiver sensitivity.

13-2) The downlink C/N_o ratio in a direct broadcast satellite (DBS) system is estimated to be 85 dB-Hz. The specifications of the link are as follows: Satellite EIRP = 57 dBW, Downlink carrier frequency = 12.5 GHz, Data rate = 10 Mb/s, Required E_b/N_o at the receiving earth terminal = 10 dB, Distance of the satellite from the receiving earth terminal = 41000 km. Calculate the minimum diameter of the dish antenna to provide a TV receiver, assuming the dish efficiency is 55% and it is located where the temperature is 310K. Assume downlink-limited operation of the DBS.

13-3) Consider the uplink power budget of the digital satellite system in Example 13-2. The parameters of the link are as follows: Carrier frequency = 14 GHz, Power density at the amplifier in saturation = -81 dBW/m², Satellite figure of merit, $G/T = 1.9$ dB/K, Distance of the satellite from the transmitting earth terminal = 40000 km.

(a) Assuming no power backoff, calculate the C/N ratio at the satellite.

(b) Given that the data rate in the uplink is the same as that of the downlink, calculate the probability of symbol error in the uplink for a link margin of 6 dB.

13-4) Choose the most suitable answer, to complete the following phrase:

i) Digital subscriber line (DSL) refers to

- a specific gauge of wire used in modem communications
- a modem enabling high-speed communications
- a connection by a modem pair enabling high-speed communications
- a specific length of wire

13-5) Consider a cellular system with channel spacing of 25kHz and $SIR_{\min} = 20$ dB for each voice channel. A total bandwidth of 6 MHz is provided to forward and reverse links and is divided into 240 channels. Assume $K_1 = -40$ in SIR_{\min} formula. Determine the number of cells in the frequency reuse plan, the maximum number of voice channels in a cell and

the spectrum efficiency in terms of number of voice circuits per cell per MHz. Repeat if SIR_{\min} requirement is dropped to 14 dB. Describe how can the frequency reuse factor N be reduced to improve spectral efficiency.

13-12. References

- [1] Timothy **Pratt**, and Charles W.**Bostian**: "Satellite Communications", John Wiley & Sons, **1986**.
- [2] A. V. **Oppenheim**, et. al., Signals and Systems, 2nd Ed., Prentice-Hall, **1996**.
- [3] W. Leon **Couch**, Digital and Analog Communications. Upper Saddle River, NJ: Prentice-Hall, 3rd Edition, **1997**.
- [4] L. **Li** and A. K. Somani. A newanalytical model for multifiber WDM networks. IEEE Journal Selected Areas in Communications, Vol. 18, No. 10, pp.2138–2145, **2000**.
- [5] E. **Rosen**, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. RFC 3031, January **2001**.
- [6] S. **Haykin**, Communication Systems, 4th Edition, Chapter 8, **2003**.
- [7] Problems from “Wireless Communications 2nd Ed.”, by **Rappaport**, Chap. 3, **2005**.
- [8] Gilbert, **Sean**; Nelson, **John** and Jacobs, **George**, World Radio TV Handbook, Watson-Guptill, **2007**.
- [9] M. EL-**SABA**, Microwave Electronics, Qassim University Press, **2008**

Appendices

In this book we refer to a number of signal properties and transforms and the reader is assumed to have at least a small prior knowledge of them. We start with the introduction of the properties of continuous-time signals and systems and the transforms that apply on them and then we continue with discrete-time signals and their transforms. For instance, there exist several forms of the Fourier Transform, namely:

- Fourier Series (**FS**) – which transforms an infinite periodic time signal into an infinite discrete frequency spectrum.
- Fourier Transform (**FT**) – which transforms an infinite continuous time signal into an infinite continuous frequency spectrum
- Discrete Fourier Transform (**DFT**) – which transforms a discrete periodic time signal into a discrete periodic frequency spectrum
- Discrete Time Fourier Transform (**DTFT**) – which transforms a discrete time signal into a discrete periodic frequency spectrum
- Fast Fourier Transform (**FFT**) – which is a computer algorithm for calculating the DFT

	Continuous Time	Discrete Time
Fourier Series	$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t}$ continuous and periodic in time $a_k = \frac{1}{T} \int_T x(t) e^{-jk\omega_0 t} dt$ discrete and aperiodic in frequency	$x[n] = \sum_{k=\langle N \rangle} a_k e^{jk \frac{2\pi}{N} n}$ discrete and periodic in time $a_k = \frac{1}{N} \sum_{n=\langle N \rangle} x[n] e^{-jk \frac{2\pi}{N} n}$ discrete and periodic in frequency
Fourier Transform	$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega) e^{j\omega t} d\omega$ continuous and aperiodic in time $X(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$ continuous and aperiodic in frequency	$x[n] = \frac{1}{2\pi} \int_{2\pi} X(e^{j\omega}) e^{j\omega n} d\omega$ discrete and aperiodic in time $X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$ continuous and periodic in frequency

It is not the intention of this book to teach the topic of signals and systems or the topic of transforms. However, we present here a brief description to remind people with these topics. If you do not know what the Laplace Transform or the Fourier Transform are, it is highly recommended that you use this appendix as a quick guide, and look for detailed information in other dedicated books.

Appendix A:	Continuous-time Signals and their Properties
Appendix B:	Fourier Series (FS) of Periodic Signals
Appendix C:	Laplace Transform (LT)
Appendix D:	Fourier Transform (FT) & Inverse FT
Appendix E:	Discrete-time Signals and their Properties
Appendix F:	Discrete Fourier Transform (DFT) & Inverse DFT
Appendix G:	Fast Fourier Transform (FFT)
Appendix H:	Z-Transform (ZT) & Inverse ZT
Appendix I:	Discrete Cosine Transform (DCT) & Inverse DCT
Appendix J:	Wavelet Trnsform
Appendix K:	MATLAB Communication ToolBox

Appendix

A

*Continuous-Time
Signals & Systems*

A system is considered **continuous** if the signal exists for all time. Frequently, the terms "analog" and "continuous" are usually employed interchangeably, although they are not strictly the same. Assume a continuous-time system with impulse response $h(t)$, as shown in the following figure. The system processes the continuous-time input signal $x(t)$ in some way, and produces an output $y(t)$. At any value of time, t , we input a value of $x(t)$ into the system and the system produces an output value for $y(t)$. For instance, if we have a multiplier system whose input/output relation is given by $y(t) = 7x(t)$, and the input value $x(t) = \sin(\omega t)$ then the output is $y(t) = 7\sin(\omega t)$.

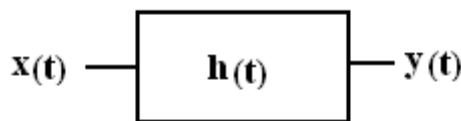


Illustration of a continuous-time system.

There exist a specific category of discrete-time systems called linear time-invariant (**LTI**) which have interesting properties.

Properties of Linear Time-Invariant (LTI) Systems

A system is considered to be an LTI system if it satisfies the requirements of time-invariance and linearity. This book will only consider linear systems. The LTI systems have the following properties:

1- Additivity

A system satisfies the property of **additivity**, if a sum of inputs results in a sum of outputs. By definition: an input of $x_3(t) = x_1(t) + x_2(t)$ results in an output of $y_3(t) = y_1(t) + y_2(t)$.

2- Homogeneity

A system satisfies the condition of **homogeneity** if an output scaled by a certain factor produces an output scaled by that same factor. By definition: an input of ax_1 results in an output of ay_1

3- Linearity

A system is considered **linear** if it satisfies the conditions of Additivity and Homogeneity.

4- Causality

A simple definition of a causal system is a system in which the output does not change before the input. A more formal definition of a causal system is the system whose output is only dependant on past or current inputs. A system is called **non-causal** if the output of the system is dependant on future inputs. This book will only consider causal systems, because they are easier to work with and understand, and since most practical systems are causal in nature.

5- Memory

A system is said to have memory if the output from the system is dependant on past inputs (or future inputs!) to the system. A system is called **memoryless** if the output is only dependant on the current input. Memoryless systems are easier to work with, but systems with memory are more common in digital signal processing applications.

6- Time-Invariance

A system is called **time-invariant** if the system relationship between the input and output signals is not dependant on the passage of time. If the input signal $x(t)$ produces an output $y(t)$ then any time shifted input, $x(t + \delta)$, results in a time-shifted output $y(t + \delta)$. This property can be satisfied if the transfer function of the system is not a function of time except expressed by the input and output.

Appendix B

Fourier Series

The Fourier series of a periodic signal $v(t)$, with period T , is given by

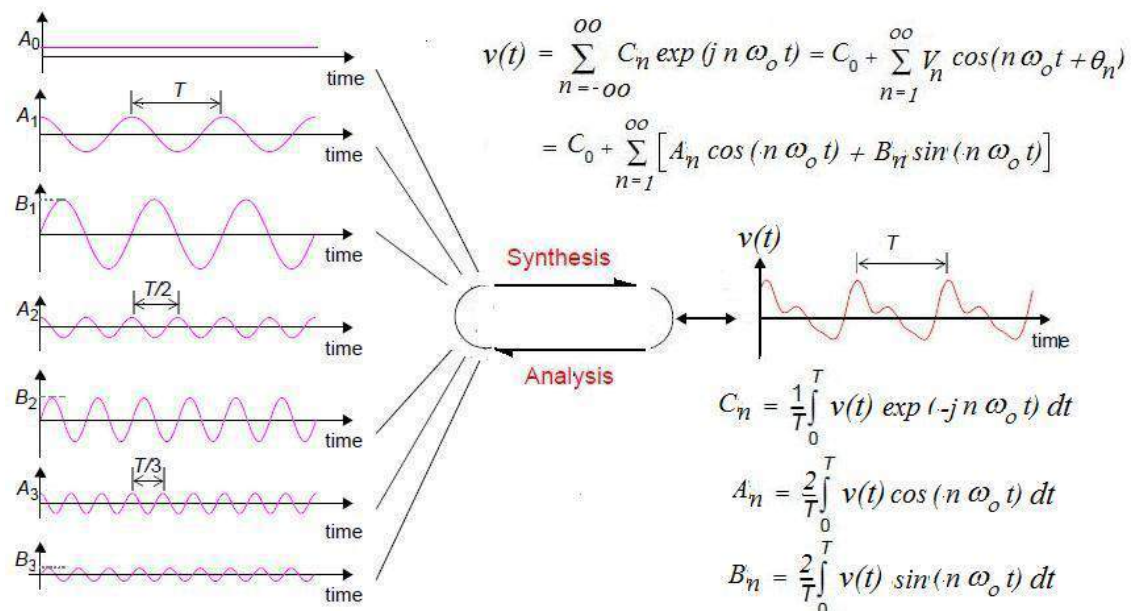
$$v(t) = \sum_{n=-\infty}^{\infty} C_n \exp(j n \omega_o t) = C_0 + \sum_{n=1}^{\infty} [A_n \cos(n \omega_o t) + B_n \sin(n \omega_o t)]$$

where ω_o is the radial frequency of the periodic signal ($\omega_o = 2\pi f_o = 2\pi/T$) and n is an integer. Also, C_0 is the average (DC) value of the signal and C_n are the amplitudes of the spectral components of the periodic signal.

$$C_0 = (1/T) \int_0^T v(t) dt$$

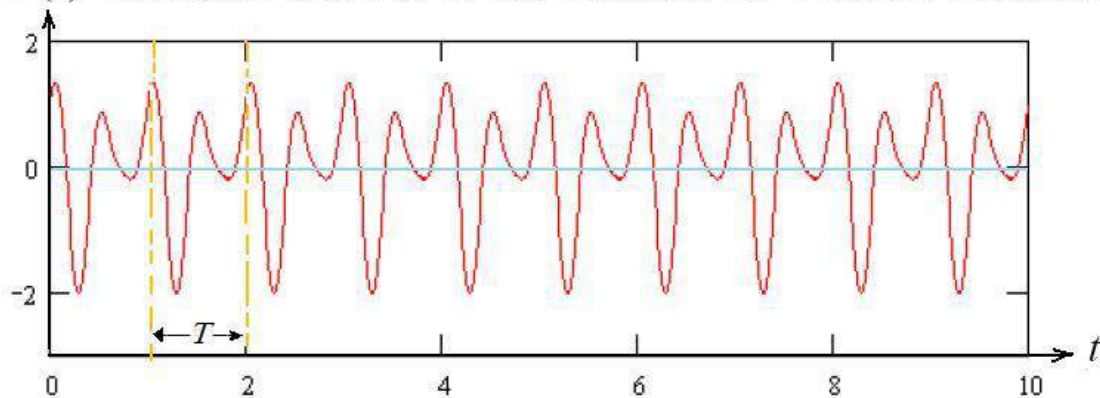
$$C_n = (1/T) \int_0^T v(t) \exp(-j n \omega_o t) dt$$

The following figure summarizes the Fourier series analysis. It shows how to analyze a periodic signal to its fundamental and harmonic components. The same figure shows how to synthesize (re-combine) a composite signal from such components.



Example: consider the following periodic signal, with its Fourier analysis components. Note that the period $T=1$ and the average DC value is zero.

$$v(t) = 3 \cos 2\pi t - 5 \sin 2\pi t + 9 \cos 4\pi t + 6 \sin 4\pi t - 2 \cos 6\pi t + 3 \sin 6\pi t$$



There exist alternative forms for the Fourier series. For instance, one can write:

$$v(t) = V_0 + \sum_{n=1}^{\infty} V_n \cos(n\omega_0 t + \theta_n)$$

where

$$\theta_n = \tan^{-1} (A_n/B_n)$$

Here, V_0 is the average (DC) value of the signal ($V_0 = C_0$) and V_n are the amplitudes of the spectral components of the periodic signal.

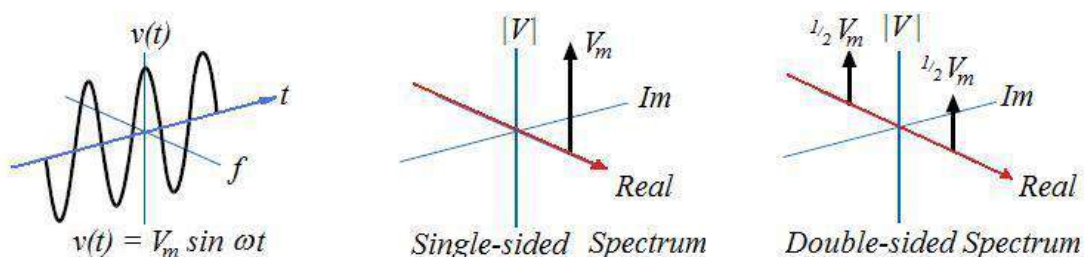
$$V_0 = (1/T) \int_0^T v(t) dt$$

$$V_n = (2/T) \int_0^T v(t) \cdot \sin(n\omega_0 t) dt$$

Note that the coefficients V_n can be calculated from double-side band spectrum, such that $V_n = C_n + C_{-n}$. If $|C_n| = |C_{-n}|$, then $|C_n| = 1/2 |V_n|$. Also the exponential coefficients C_n can be obtained from the co-sinusoidal coefficients A_n and B_n , by the relation.

$$C_n = \sqrt{(A_n^2 + B_n^2)}$$

The following figure shows the single-sided and double sided spectrum of a single tone (single frequency) cosine waveform:

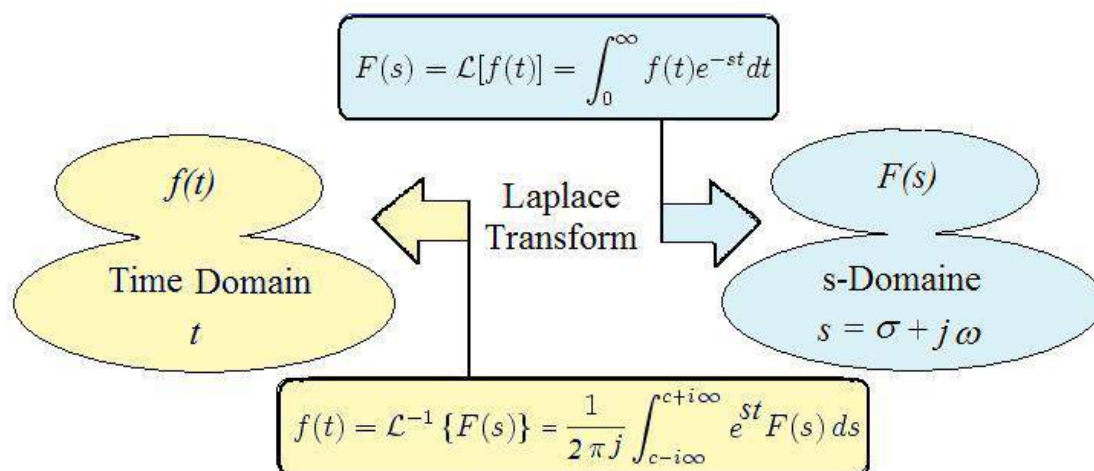


Appendix

C

Laplace Transform

The Laplace Transform converts an equation from the time-domain into the so-called **s-domain**, or the **Laplace domain**, or even the **complex frequency domain**. These are all different names for the same mathematical space, and they all may be used in this book.



The Laplace transform is defined as follows:

$$F(s) = \mathcal{L}[f(t)] = \int_0^{\infty} f(t)e^{-st} dt \quad (\text{B-1})$$

Laplace transform results have been tabulated extensively. The following table includes the Laplace transform of famous functions.

Table B1. Laplace Transforms:

Time Domain Function	Laplace Transform
$x(t) = \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{st} ds$	$X(s) = \int_{-\infty}^{\infty} x(t)e^{-st} dt$
$\delta(t)$	1
$\delta(t - a)$	e^{-as}

Time Domain Function	Laplace Transform
$u(t)$	$\frac{1}{s}$
$u(t - a)$	$\frac{e^{-as}}{s}$
$t \cdot u(t)$	$\frac{1}{s^2}$
$t^n u(t)$	$\frac{n!}{s^{n+1}}$
$\frac{1}{\sqrt{\pi t}} u(t)$	$\frac{1}{\sqrt{s}}$

Time Domain	Laplace Domain
$e^{at} u(t)$	$\frac{1}{s - a}$
$t^n e^{at} u(t)$	$\frac{n!}{(s - a)^{n+1}}$
$\cos(\omega t) \cdot u(t)$	$\frac{s}{s^2 + \omega^2}$
$\sin(\omega t) u(t)$	$\frac{\omega}{s^2 + \omega^2}$
$\cosh(\omega t) \cdot u(t)$	$\frac{s}{s^2 - \omega^2}$
$\sinh(\omega t) \cdot u(t)$	$\frac{\omega}{s^2 - \omega^2}$
$e^{at} \cos(\omega t) u(t)$	$\frac{s - a}{(s - a)^2 + \omega^2}$
$e^{at} \sin(\omega t) \cdot u(t)$	$\frac{\omega}{(s - a)^2 + \omega^2}$
$\frac{1}{2\omega^3} (\sin \omega t - \omega t \cos \omega t)$	$\frac{1}{(s^2 + \omega^2)^2}$
$\frac{t}{2\omega} \sin \omega t$	$\frac{s}{(s^2 + \omega^2)^2}$
$\frac{1}{2\omega} (\sin \omega t + \omega t \cos \omega t)$	$\frac{s^2}{(s^2 + \omega^2)^2}$

Properties of the Laplace Transform

Property	Definition
Linearity	$\mathcal{L}\{af(t) + bg(t)\} = aF(s) + bG(s)$
Differentiation	$\mathcal{L}\{f'\} = s\mathcal{L}\{f\} - f(0^-)$ $\mathcal{L}\{f''\} = s^2\mathcal{L}\{f\} - sf(0^-) - f'(0^-)$ $\mathcal{L}\{f^{(n)}\} = s^n\mathcal{L}\{f\} - s^{n-1}f(0^-) - \dots - f^{(n-1)}(0^-)$
Frequency Division	$\mathcal{L}\{tf(t)\} = -F'(s)$ $\mathcal{L}\{t^n f(t)\} = (-1)^n F^{(n)}(s)$

Properties of the Laplace Transform (Cont.)

Property	Definition
Frequency Integration	$\mathcal{L}\left\{\frac{f(t)}{t}\right\} = \int_s^\infty F(\sigma) d\sigma$
Time Integration	$\mathcal{L}\left\{\int_0^t f(\tau) d\tau\right\} = \mathcal{L}\{u(t) * f(t)\} = \frac{1}{s}F(s)$
Scaling	$\mathcal{L}\{f(at)\} = \frac{1}{a}F\left(\frac{s}{a}\right)$
Initial value theorem	$f(0^+) = \lim_{s \rightarrow \infty} sF(s)$
Final value theorem	$f(\infty) = \lim_{s \rightarrow 0} sF(s)$
Frequency Shifts	$\mathcal{L}\{e^{at}f(t)\} = F(s - a)$ $\mathcal{L}^{-1}\{F(s - a)\} = e^{at}f(t)$
Time Shifts	$\mathcal{L}\{f(t - a)u(t - a)\} = e^{-as}F(s)$ $\mathcal{L}^{-1}\{e^{-as}F(s)\} = f(t - a)u(t - a)$
Convolution Theorem	$\mathcal{L}\{f(t) * g(t)\} = F(s)G(s)$

where: $f(t) = \mathcal{L}^{-1}\{F(s)\}$, $g(t) = \mathcal{L}^{-1}\{G(s)\}$ and $s = \sigma + j\omega$

Inverse Laplace Transform

The inverse Laplace Transform is defined as such:

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} e^{st} F(s) ds \quad (\text{B-2})$$

The inverse transform converts a function from the Laplace domain back into the time domain.

Note that $F(s)$ may not fall into any standard form, for which the inverse Laplace transform is well-known. In this case we use the so-called **Partial Fraction Expansion** (also called **Heaviside expansion**) **Method** to expand $F(s)$ into a series of simple terms, whose inverse Laplace transform is known. Assume

$$F(s) = \frac{N(s)}{D(s)} = \frac{K(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)} \quad (\text{B-3})$$

Then, we can expand $F(s)$ in the following form:

$$F(s) = \frac{N(s)}{D(s)} = \sum_{i=1}^n \frac{a_i}{s+p_i} = \frac{a_1}{s+p_1} + \frac{a_2}{s+p_2} + \dots + \frac{a_n}{s+p_n} \quad (\text{B-4})$$

where the coefficients a_i ($i=1, 2, \dots, n$) are called the **residues** of $F(s)$ and p_i are called the poles of $F(s)$. The residues a_i can be obtained from $F(s)$ as follows:

$$a_i = \lim_{s \rightarrow -p_i} \left[(s+p_i) \frac{N(s)}{D(s)} \right] \quad (\text{B-5})$$

Therefore, the inverse Laplace transform of $F(s)$ is simply equal to:

$$f(t) = \mathcal{L}^{-1}[F(s)] = \mathcal{L}^{-1} \left[\sum_{i=1}^n \frac{a_i}{s+p_i} \right] = \sum_{i=1}^n a_i \exp(-p_i t) \quad (\text{B-6})$$

Example:

Assume:

$$F(s) = \frac{s+5}{s^2+3s+2} = \frac{(s+5)}{(s+1)(s+2)}$$

So, we have two poles: $p_1 = -1$ and $p_2 = -2$. Using the partial expansion method, equation (B-5), we get $a_1 = 4$ and $a_2 = -3$, so that

$$F(s) = \frac{4}{(s+1)} - \frac{3}{(s+2)}$$

Therefore: $f(t) = 4 \cdot \exp(-t) - 3 \cdot \exp(-2t)$

N.B. When a pole, p_i , is repeated k times, then it is expanded in k -terms and the coefficients ($a_{i0}, a_{i1}, a_{i(k-1)}$) are obtained as follows:

$$a_{ij} = \lim_{s \rightarrow -p_i} \left[\frac{d^{k-j}}{ds^{k-j}} (s - p_i)^k \frac{N(s)}{D(s)} \right], \quad \text{for } j = 1, 2, \dots, k \quad (\text{B-7})$$

Example:

Assume the following transfer function:

$$F(s) = \frac{s^2 + 2s + 3}{s^3 + 3s^2 + 3s + 1} = \frac{s^2 + 2s + 3}{(s+1)^3}$$

Here, we have 3 identical poles ($k=3$) at: $p_1 = -1$. So, we can write $F(s)$ as follows:

$$F(s) = \frac{a_{11}}{(s+1)} + \frac{a_{12}}{(s+1)^2} + \frac{a_{13}}{(s+1)^3}$$

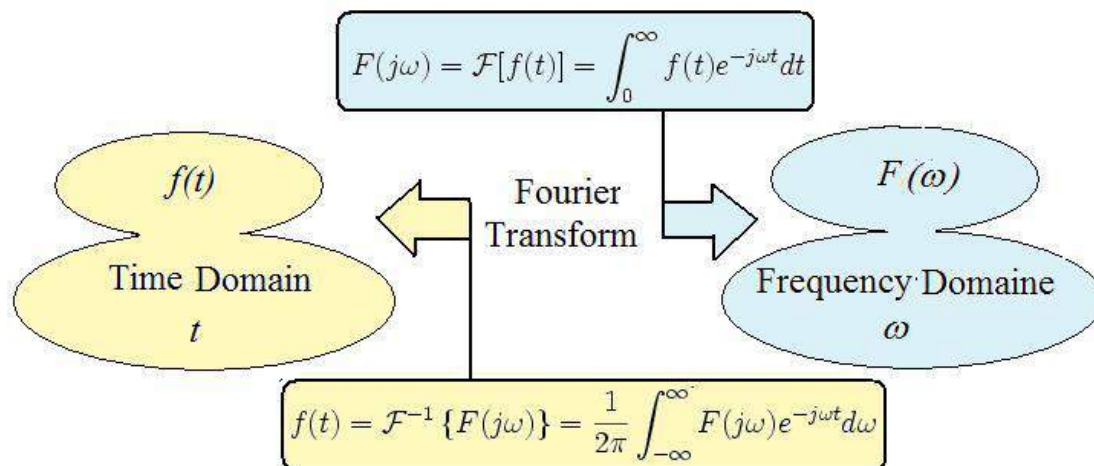
Using the partial expansion method, equation (B-7), we get $a_{11} = 1$ and $a_{12} = 0$, and $a_{13} = 2$, so that:

$$f(t) = 1 * \exp(-t) + 0 * t \exp(-t) + t^2 \exp(-t) = (1 + t^2) \exp(-t)$$

Appendix D

Fourier Transform (FT)

The **Fourier Transform** (or integral Fourier transform) is very similar to the Laplace transform. The Fourier transform uses the assumption that any finite time-domain can be broken into an infinite sum of sinusoidal (sine and cosine waves) signals. Under this assumption, the Fourier Transform converts a time-domain signal into its frequency-domain representation, as a function of the radial frequency, ω .



The Fourier Transform is defined as such:

$$F(j\omega) = \mathcal{F}[f(t)] = \int_0^{\infty} f(t)e^{-j\omega t} dt \quad (\text{C-1})$$

We can show that the Fourier Transform is equivalent to the Laplace transform, when the condition $s = j\omega$ is true. Because the Laplace and Fourier Transforms are so closely related, it does not make much sense to use both transforms for all problems. For frequency problems, it makes life much easier to use the Fourier Transform representation. Like the Laplace Transform, the Fourier Transform has been extensively tabulated. The following table illustrates the Fourier Transform of famous functions as well as the properties of the Fourier transform.

Table C-1. Fourier Transforms

Time Domain	Frequency (Fourier) Domain
$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega) e^{j\omega t} d\omega$	$X(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$
1	$2\pi\delta(\omega)$
$\delta(t)$	1
$\delta(t - c)$	$e^{-j\omega c}$
$u(t)$	$\pi\delta(\omega) + \frac{1}{j\omega}$
$e^{-bt} \cdot u(t)$	$1/(j\omega + b)$
$\cos \omega_0 t$	$\pi [\delta(\omega + \omega_0) + \delta(\omega - \omega_0)]$
$\cos(\omega_0 t + \theta)$	$\pi [e^{-j\theta} \delta(\omega + \omega_0) + e^{j\theta} \delta(\omega - \omega_0)]$
$\sin \omega_0 t$	$j\pi [\delta(\omega + \omega_0) - \delta(\omega - \omega_0)]$
$\sin(\omega_0 t + \theta)$	$j\pi [e^{-j\theta} \delta(\omega + \omega_0) - e^{j\theta} \delta(\omega - \omega_0)]$
$p_\tau(t)$	$\tau \text{sinc}(\tau\omega / 2\pi)$
$\tau \text{sinc}(t\tau / 2\pi)$	$2\pi p_\tau(\omega)$
$(1 - 2 t /\tau) p_\tau(t)$	$1/2 \tau \text{sinc}^2(\tau\omega / 4\pi)$
$1/2 \tau \text{sinc}^2(t\tau/4\pi)$	$2\pi (1 - 2 \omega /\tau) p_\tau(\omega)$

Note: $u(t)$ is the unit step function, $p_\tau(t) = \text{rect}(t/\tau)$ is the rectangular pulse function of width τ and $\text{sinc}(x) = \sin(x)/x$

Inverse Fourier Transform (IFT)

The **inverse Fourier Transform** is defined as follows:

$$f(t) = \mathcal{F}^{-1}\{F(j\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega) e^{-j\omega t} d\omega \quad (\text{C-1})$$

This transform is nearly identical to the Fourier Transform. Using the above equivalence, we can show that the Laplace transform is always equal to the Fourier Transform, if the variable s is an imaginary number ($s = j\omega$). However, the Laplace transform is different if s is a real or a complex. Generally, the variable s can be broken down into 2 independent components ($s = \sigma + j\omega$). Therefore, it is frequently to plot the variable s on its own S -plane. The S -plane graphs the variable σ on the horizontal axis, and the value of $j\omega$ on the vertical axis.

Appendix E

Discrete-Time Signals & Systems

Digital data is represented by discrete number values. Assume a discrete-time system with impulse response $h[n]$, as shown in the following figure. The system processes the input $x[n]$ in some way, and produces an output $y[n]$. At each integer value for discrete time, n , we input a value of $x[n]$ into the system and the system produces an output value for $y[n]$. For instance, if we have a multiplier system whose input/output relation is given by $y[n] = 7 x[n]$, and the input sequence $x[n] = [1 \ 0 \ 1 \ 3]$ then the out output sequence is $y[n] = [7 \ 0 \ 7 \ 21]$.

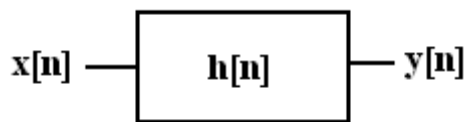


Illustration of a discrete-time system.

There exist a specific category of discrete-time systems called linear shift-invariant (**LSI**) which have interesting properties.

Properties of Linear Shifty-Invariant (LSI) Systems

A system is considered to be an LSI system if it satisfies the requirements of shift-invariance and linearity. LSI systems have the following properties:

1- Additivity

A system satisfies the property of **additivity**, if a sum of inputs results in a sum of outputs. By definition: an input of $x_3[n] = x_1[n] + x_2[n]$ results in an output of $y_3[n] = y_1[n] + y_2[n]$.

2- Homogeneity

A system satisfies the condition of **homogeneity** if an output scaled by a certain factor produces an output scaled by that same factor. By definition: an input of ax_1 results in an output of ay_1

3- Linearity

A system is considered **linear** if it satisfies the conditions of Additivity and Homogeneity.

4- Causality

A simple definition of a causal system is a system in which the output does not change before the input. A more formal definition of a causal system is the system whose output is only dependant on past or current inputs. A system is called **non-causal** if the output of the system is dependant on future inputs. This book will only consider causal systems, because they are easier to work with and understand, and since most practical systems are causal in nature.

5- Memory

A system is said to have memory if the output from the system is dependant on past inputs (or future inputs!) to the system. A system is called **memoryless** if the output is only dependant on the current input. Memoryless systems are easier to work with, but systems with memory are more common in digital signal processing applications.

6- Shift-Invariance

A system is called **shift-invariant** if the system relationship between the input and output signals is not dependant on the passage of time.

If the input signal $x[n]$ produces an output $y[n]$ then any time shifted input, $x[n + N]$, results in a time-shifted output $y[n + N]$ This property can be satisfied if the transfer function of the system is not a function of time.

Appendix F

Discrete Fourier Transform (DFT)

The Discrete Fourier Transform (**DFT**) of a signal x may be defined by:

$$X(\omega_k) \triangleq \sum_{n=0}^{N-1} x(t_n) e^{-j\omega_k t_n}, \quad k = 0, 1, 2, \dots, N-1,$$

where

“ \triangleq ” means “is defined as” or “equals by definition”

$$\sum_{n=0}^{N-1} f(n) \triangleq f(0) + f(1) + \dots + f(N-1)$$

$x(t_n)$ \triangleq input signal *amplitude* (real or complex) at time t_n (sec)

t_n \triangleq $nT = n$ th sampling instant (sec), n an integer ≥ 0

T \triangleq sampling interval (sec)

$X(\omega_k)$ \triangleq *spectrum* of x (complex valued), at frequency ω_k

ω_k \triangleq $k\Omega = k$ th frequency sample (radians per second)

Ω \triangleq $\frac{2\pi}{NT} =$ radian-frequency sampling interval (rad/sec)

f_s \triangleq $1/T =$ *sampling rate* (samples/sec, or Hertz (Hz))

$N =$ number of time samples = no. frequency samples (integer).

The sampling interval T is also called the sampling *period*.

When all N signal samples $x(t_n)$ are real, we say $x \in R^N$. If they may be complex, we write $x \in C^N$. Finally, $n \in N$ means n is any integer.

Inverse DFT

The *inverse* DFT (**IDFT**) is given by :

$$x(t_n) = \frac{1}{N} \sum_{k=0}^{N-1} X(\omega_k) e^{j\omega_k t_n}, \quad n = 0, 1, 2, \dots, N-1.$$

In the signal processing literature, it is common to write the DFT and its inverse in the more pure form below, obtained by setting $T=1$ in the previous definition:

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}, \quad k = 0, 1, 2, \dots, N-1$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi nk/N}, \quad n = 0, 1, 2, \dots, N-1$$

where $x(n)$ denotes the input signal at time (sample) n , and $X(k)$ denotes the k^{th} spectral sample. This form is the simplest mathematically, while the previous form is easier to interpret physically.

Having completely understood the DFT and its inverse mathematically, we go on to proving various Fourier Theorems, such as the **shift theorem**, the **convolution theorem**, and **Parseval's theorem**. The Fourier theorems provide a basic thinking vocabulary for working with signals in the time and frequency domains.

Rayleigh Energy Theorem (Parseval's Theorem)

For any $x \in C^N$,

$$\|x\|^2 = \frac{1}{N} \|X\|^2.$$

i.e.,

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2.$$

This is a special case of the power theorem. It, too, is often referred to as **Parseval's theorem** (being a special case).

Appendix G

Fast Fourier Transform (FFT)

The fast Fourier transform (**FFT**) is an efficient implementation of the discrete Fourier transform (**DFT**) for highly composite transform lengths N . When N is a power of 2, the computational complexity drops from $O(N^2)$ for the DFT down to $O(N \ln N)$ for the FFT, where $\ln N$ denotes the logarithm-base-2 of N . The FFT was first described by Gauss in 1805 and rediscovered in the 1960s by Cooley and Tukey. The first stage of a radix-2 FFT algorithm is derived below; the remaining stages are obtained by applying the same decomposition recursively. As we have seen so far, the DFT is defined as follows:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, \quad k = 0, 1, 2, \dots, N-1,$$

where $x(n)$ is the input signal amplitude at time n , and

$$W_N \triangleq e^{-j\frac{2\pi}{N}}. \quad (\text{primitive } N\text{th root of unity})$$

Note also that $W_N^N = 1$. There are two basic varieties of FFT, one based on *decimation in time* (**DIT**), and the other on *decimation in frequency* (**DIF**). Here we will derive decimation in time. When N is even, the DFT summation can be split into sums over the odd and even indexes of the input signal:

$$\begin{aligned} X(\omega_k) &\triangleq \text{DFT}_{N,k}\{x\} \triangleq \sum_{n=0}^{N-1} x(n)e^{-j\omega_k nT}, \quad \omega_k \triangleq \frac{2\pi k}{NT} \\ &= \sum_{\substack{n=0 \\ n \text{ even}}}^{N-2} x(n)e^{-j\omega_k nT} + \sum_{\substack{n=0 \\ n \text{ odd}}}^{N-1} x(n)e^{-j\omega_k nT} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(2n)e^{-j2\pi \frac{k}{N/2} n} + e^{-j2\pi \frac{k}{N}} \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)e^{-j2\pi \frac{k}{N/2} n}, \end{aligned} \quad (\text{F.1})$$

$$\begin{aligned}
X(\omega_k) &= \sum_{n=0}^{\frac{N}{2}-1} x_e(n)W_{N/2}^{kn} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_o(n)W_{N/2}^{kn} \\
&\triangleq \text{DFT}_{\frac{N}{2},k}\{\text{DOWNSAMPLE}_2(x)\} \\
&\quad + W_N^k \cdot \text{DFT}_{\frac{N}{2},k}\{\text{DOWNSAMPLE}_2[\text{SHIFT}_1(x)]\}, \tag{F.2}
\end{aligned}$$

where $x_e(n) \equiv x(2n)$ and $x_o(n) \equiv x(2n+1)$ denote the even- and odd-indexed samples from x . Thus, the length N DFT is computable using two length $1/2 N$ DFTs. The complex factors:

$$W_N^k = e^{-j\omega_k} = \exp(-j2\pi k/N)$$

are called *twiddle factors*. The splitting into sums over even and odd time indexes is called *decimation in time*. (For *decimation in frequency*, the inverse DFT of the spectrum $X(\omega_k)$ is split into sums over even and odd *bin numbers* k .)

Appendix H

Z- Transform

The Z-transform is used primarily to convert discrete data sets into a continuous representation. The Z-transform is very similar to the star transform, except that the Z-transform does not take explicit account for the sampling period. The Z-transform has a number of uses in the field of digital signal processing and the study of discrete signals in general, and is useful because Z-transform results are extensively tabulated, whereas star-transform results are not. The Z-Transform is defined as:

$$X(z) = \mathcal{Z}[x[n]] = \sum_{i=-\infty}^{\infty} x[n]z^{-n} \quad (\text{G-1})$$

The following table depicts the z-transform of main signals and their region of convergence (**ROC**).

Z-Transform Tables:

	Signal, $x[n]$	Z-transform, $X(z)$	ROC
1	$\delta[n]$	1	all z
2	$\delta[n - n_0]$	$\frac{1}{z^{n_0}}$	all z
3	$u[n]$	$\frac{z}{z - 1}$	$ z > 1$
4	$a^n u[n]$	$\frac{1}{1 - az^{-1}}$	$ z > a $
5	$na^n u[n]$	$\frac{az^{-1}}{(1 - az^{-1})^2}$	$ z > a $
6	$-a^n u[-n - 1]$	$\frac{1}{1 - az^{-1}}$	$ z < a $

Z-Transform Tables (Cont.)

	Signal, $x[n]$	Z-transform, $X(z)$	ROC
7	$-na^n u[-n-1]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	
8	$\cos(\omega_0 n)u[n]$	$\frac{1-z^{-1}\cos(\omega_0)}{1-2z^{-1}\cos(\omega_0)+z^{-2}}$	$ z > 1$
9	$\sin(\omega_0 n)u[n]$	$\frac{z^{-1}\sin(\omega_0)}{1-2z^{-1}\cos(\omega_0)+z^{-2}}$	$ z > 1$
10	$a^n \cos(\omega_0 n)u[n]$	$\frac{1-az^{-1}\cos(\omega_0)}{1-2az^{-1}\cos(\omega_0)+a^2z^{-2}}$	$ z > a $
11	$a^n \sin(\omega_0 n)u[n]$	$\frac{az^{-1}\sin(\omega_0)}{1-2az^{-1}\cos(\omega_0)+a^2z^{-2}}$	$ z > a $

The Inverse Z-Transform is sufficiently complex that we will not consider it here.

Appendix

I

Discrete Cosine Transform

The discrete cosine transform (DCT), like any Fourier-related transform, expresses a function or a signal in terms of a sum of sinusoids with different frequencies and amplitudes. The DCT is often used in signal and image processing, especially for lossy data compression. For instance, the DCT is used in JPEG image compression, as well as MPEG, and DV video compression. A related transform, called the modified discrete cosine transform (MDCT), is used in AAC, Vorbis, WMA, and MP3 audio compression.

The DCT is similar to the discrete Fourier transform (DFT), in the sense that it operates on a function at a finite number of discrete data points. The obvious distinction between a DCT and a DFT is that the former uses only cosine functions, while the latter uses both cosines and sines (or complex exponentials).

There are several variants of the DCT with slightly modified definitions. The formal DCT is defined as follows. The N real numbers x_0, \dots, x_{N-1} are transformed into the N real numbers X_0, \dots, X_{N-1} according to one of the formulas:

DCT-I

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos \left[\frac{\pi}{N-1} nk \right] \quad k = 0, \dots, N-1. \quad (\text{H-1})$$

DCT-II

$$X_k = \sum_{n=0}^{N-1} x_n \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right] \quad k = 0, \dots, N-1. \quad (\text{H-2})$$

The DCT-II is probably the most commonly used form of DCT. Some authors multiply the X_0 term by $1/\sqrt{2}$ to make the DCT matrix orthogonal.

Multidimensional variants of the various DCT types follow straightforwardly from the one-dimensional definitions. For instance, For example, a two-dimensional DCT-II of an image or a matrix is simply the one-dimensional DCT-II, performed along the rows and then along the columns (or vice versa).

$$X_{k_1, k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1, n_2} \cos \left[\frac{\pi}{N_1} \left(n_1 + \frac{1}{2} \right) k_1 \right] \cos \left[\frac{\pi}{N_2} \left(n_2 + \frac{1}{2} \right) k_2 \right].$$

The following table depicts the DCT of main signals and their region of convergence.

Appendix

J

Wavelet Transforms

The wavelet transform was developed as an alternative approach to the short time Fourier transform to overcome the resolution problem. One of the most popular applications of wavelet transform is image compression. The advantage of using wavelet-based coding in image compression is that it provides significant improvements in picture quality at higher compression ratios over conventional techniques. Since wavelet transform has the ability to decompose complex information and patterns into elementary forms, it is commonly used in acoustics processing and pattern recognition. Moreover, wavelet transforms can be applied to the following scientific research areas: edge and corner detection, partial differential equation solving, transient detection, filter design, Electrocardiogram (ECG) analysis, texture analysis and business information analysis. Continuous Wavelet Transform (CWT) is very efficient in determining the damping ratio of oscillating signals.

I-1. Wavelet Series

In mathematics, a **wavelet series** is a representation of a square-integrable (real- or complex-valued) function by a certain orthonormal series generated by a wavelet. This appendix provides a formal, mathematical definition of an **orthonormal wavelet** and of the **integral wavelet transform**. A function $\psi \in L^2(\mathcal{R})$ is called an **orthonormal wavelet** if it can be used to define a Hilbert basis, that is a complete orthonormal system, for the Hilbert space $L^2(\mathcal{R})$ of square integrable functions. The Hilbert basis is constructed as the family of functions $\{\psi_{jk} : j, k \in \mathcal{Z}\}$ by means of dyadic translations and dilations of ψ ,

$$\psi_{jk}(x) = 2^{j/2} \psi(2^j x - k)$$

for integers $j, k \in \mathcal{Z}$. This family is an orthonormal system if it is orthonormal under the inner product

$$\langle \psi_{jk}, \psi_{lm} \rangle = \delta_{jl} \cdot \delta_{km}$$

where δ_{jl} is the Kronecker delta and $\langle f, g \rangle$ is the standard inner product

$$\langle f, g \rangle = \int_{-\infty}^{\infty} \overline{f(x)} g(x) dx \quad \text{on } L^2(\mathbb{R})$$

The requirement of completeness is that every function $f \in L^2(\mathbb{R})$ may be expanded in the basis as

$$f(x) = \sum_{j,k=-\infty}^{\infty} c_{jk} \psi_{jk}(x)$$

with convergence of the series understood to be convergence in norm. Such a representation of a function f is known as a **wavelet series**.

I-2. Continuous Wavelet Transform (CWT)

The continuous wavelet transform (CWT) was developed as an alternative approach to the short time Fourier transform (STFT) to overcome the resolution problem. The CWT is used to divide a continuous-time function into wavelets. Unlike Fourier transform, the continuous wavelet transform possesses the ability to construct a time-frequency representation of a signal that offers very good time and frequency localization. In mathematics, the continuous wavelet transform of a continuous, square-integrable function $x(t)$ at a scale $a > 0$ and translational value $b \in \mathbb{R}$ is expressed by the following integral

$$X_w(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t-b}{a} \right) dt$$

where $\psi(t)$ is a continuous function in both the time domain and the frequency domain called the **mother wavelet** and represents operation of complex conjugate. The main purpose of the mother wavelet is to provide a source function to generate the **daughter wavelets** which are simply the translated and scaled versions of the mother wavelet. To recover the original signal $x(t)$, inverse continuous wavelet transform can be expressed as follows:

$$x(t) = \int_0^{\infty} \int_{-\infty}^{\infty} \frac{1}{a^2} X_w(a, b) \frac{1}{\sqrt{|(a)|}} \tilde{\psi} \left(\frac{t-b}{a} \right) db da$$

$\psi(t)$ is the dual function of $\psi(t)$. And the dual function should satisfy

$$\int_0^{\infty} \int_{-\infty}^{\infty} \frac{1}{|a^3|} \psi\left(\frac{t_1 - b}{a}\right) \tilde{\psi}\left(\frac{t - b}{a}\right) db da = \delta(t - t_1)$$

Sometimes,

$$\tilde{\psi}(t) = C_{\psi}^{-1} \psi(t),$$

where

$$C_{\psi} = \frac{1}{2} \int_{-\infty}^{+\infty} \frac{|\hat{\psi}(\zeta)|^2}{|\zeta|} d\zeta$$

is called the admissibility constant and $\hat{\psi}$ is the Fourier transform of ψ . For a successful inverse transform, the admissibility constant has to satisfy the admissibility condition:

$$0 < C_{\psi} < +\infty .$$

It is possible to show that the admissibility condition implies that $\hat{\psi}(0) = 0$ so that a wavelet must integrate to zero.

In general, it is preferable to choose a mother wavelet that is continuously differentiable with compactly supported scaling function and high vanishing moments.

I-3. Properties of Continuous Wavelet Transform

In definition, the continuous wavelet transform is a convolution of the input data sequence with a set of functions generated by the mother wavelet. The convolution can be computed by using the Fast Fourier Transform (FFT). Normally, the output $X_w(a,b)$ is a real valued function except when the mother wavelet is complex. A complex mother wavelet will convert the continuous wavelet transform to a complex valued function. The power spectrum of the continuous wavelet transform can be represented by $|X_w(a,b)|^2$

I-4. Discrete Wavelet Transform (DWT)

The foundations of the DWT go back to 1976 when Croiser, Esteban, and Galand devised a technique to decompose discrete time signals. The DWT is considerably easier to implement when compared to the CWT. The main idea is the same as it is in the CWT. A time-scale representation of a digital signal is obtained using digital filtering techniques. The continuous wavelet transform was computed by changing the scale of the analysis window, shifting the window in time, multiplying by the signal (mother wavelet), and integrating over all times. In the discrete case, filters of different cutoff frequencies are used to analyze the signal at different scales. The signal is passed through a series of high pass filters to analyze the high frequencies, and it is passed through a series of low pass filters to analyze the low frequencies. The procedure starts with passing the discrete signal (sequence) through a half band digital lowpass filter with impulse response $h[n]$. Filtering a signal corresponds to the mathematical operation of convolution of the signal with the impulse response of the filter. The convolution operation in discrete time is defined as follows:

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n - k]$$

A half band lowpass filter removes all frequencies that are above half of the highest frequency in the signal. For example, if a signal has a maximum of 1000 Hz component, then half band lowpass filtering removes all the frequencies above 500 Hz. After passing the signal through a half band lowpass filter, half of the samples can be eliminated according to the Nyquist's rule. The signal is then sub-sampled by 2 since half of the number of samples are redundant. This doubles the scale. This procedure can mathematically be expressed as

$$y[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[2n - k]$$

The signal can therefore be sub-sampled by 2, simply by discarding every other sample. This constitutes one level of decomposition and can mathematically be expressed as follows:

$$y_{high}[k] = \sum_n x[n] \cdot g[2k - n]$$

$$y_{low}[k] = \sum_n x[n] \cdot h[2k - n]$$

where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the highpass and lowpass filters, respectively, after subsampling by 2. Note that this decomposition has halved the time resolution since only half of each filter output character- the signal. However, each output has half the frequency band of the input so the frequency resolution has been doubled.

I-5. Inverse DWT (Reconstruction)

The interesting point in DWT is that the analysis and synthesis of digital filters are identical to each other, except for a time reversal. Therefore, the reconstruction formula becomes (for each layer)

$$x[n] = \sum_{k=-\infty}^{\infty} (y_{high}[k] \cdot g[-n + 2k]) + (y_{low}[k] \cdot h[-n + 2k])$$

I-6. Other Forms of DWT

Other forms of discrete wavelet transform include the non-decimated wavelet transform (where down-sampling is omitted), the Newland transform (where an orthonormal basis of wavelets is formed from appropriately constructed top-hat filters in frequency space). Wavelet packet transforms are also related to the discrete wavelet transform. Complex wavelet transform is another form.

**Appendix
K****Overview of the
MATLAB
Communication Toolbox**

Communications Toolbox extends the MATLAB® technical computing environment with functions, plots, and a graphical user interface (GUI) for exploring, designing, analyzing, and simulating algorithms for the physical layer of communication systems.

You can execute Communications Toolbox functions from the MATLAB command line and with your custom MATLAB scripts and functions. The MATLAB editor and graphical user interface development environment (GUIDE) accelerate the development of your system simulations. The Communications Toolbox helps you create algorithms for commercial systems, such as mobile handsets and base stations, wired and wireless local area networks, and digital subscriber lines. You can also use it in research and education for communication systems engineering.

The MATLAB communication toolbox includes, but not limited to, the following features:

- Functions for designing the physical layer of communications links, including source coding, channel coding, interleaving, modulation, channel models, and equalization
- Graphical plots for visualizing communications signals, such as eye diagrams, constellations, and channel scattering functions
- Graphical user interface for comparing the bit error rate of your system with a wide variety of proven analytical results
- Standard channel models such as GSM/UTMS, and ionosphere propagation, for evaluating system performance under a wide range of propagation conditions
- Channel visualization tool for visualizing and exploring time-varying communications channels

You can download free demos of the communication toolbox, from the following website: www.mathworks.com/demos

