GCA S3 04

Data Communication and Computer Networks

SEMESTER III

BACHELOR OF COMPUTER APPLICATION

Block - 1

KRISHNA KANTA HANDIQUI STATE OPEN UNIVERSITY

Subject Expert

Prof. Anjana Kakati Mahanta, Gauhati University
Prof.(Retd.) Pranhari Talukdar, Gauhati University
Dr. Jyotiprakash Goswami, Assam Engineering College

Course Coordinator

Dr. Sanjib Kr. Kalita, KKHSOU
Dr. Tapashi Kashyap Das, KKHSOU
Sruti Sruba Bharali, KKHSOU

SLM Preparation Team

UNITS	CONTRIBUTORS
1 & 3	Rupam Sarma, Don Bosco University
2	Sruti Sruba Bharali, KKHSOU
4	Dr. Tapashi Kashyap Das, KKHSOU
5 & 6	Dr. Chakradhar Das, Lakhimpur Polytechnic
7	Pritam Medhi, Gauhati University

Editorial Team

Content : Prof. Kandarpa Sarma, Gauhati University
Language : Prof.(Retd) Robin Goswami, Cotton College

Structure, Format & Graphics : Dr. Sanjib Kr. Kalita, KKHSOU

June 2018

This Self Learning Material (SLM) of the Krishna Kanta Handiqui State Open University is made available under a Creative Commons Attribution-Non Commercial-Share Alike 4.0 License (international): http://creativecommons.org/licenses/by-nc-sa/4.0/

Printed and published by Registrar on behalf of the Krishna Kanta Handiqui State Open University.

Headquarters : Patgaon, Rani Gate, Guwahati - 781017
City Office : Housefed Complex, Dispur, Guwahati-781 006; Web: www.kkhsou.in

he University acknowledges with thanks the financial support provided by the **Distance Education Bureau**, **UGC** for the preparation of this study material.

COURSE INTRODUCTION

This course is basically designed for the learners who are beginers in the field of computer networks. This is an age of digital technology. In this age it is very important to have some ideas about digital communication. It is also very essential to have some knowledge of data transmission, speed of transmission, medium of transmission and techniques of transmission. This course aims at introducing to the learners the basic knowledge of communication technology. The different layers of communication technology have been explained in detail.

There are two blocks in this course.

Block 1 deals with the basic concept of networking devices and communication principles. After going through this block the learners will feel comfortable to proceed to Block 2.

Block 2 concentrates on the different layers of the OSI reference network model. At the end of this block, the learners will acquire comprehensive ideas about network protocols and standards.

Each unit of these two blocks includes some along-side boxes to help you know some of the difficult, unseen terms. Some “EXERCISES” have been included to help you apply your own thoughts. You may find some boxes marked with: “LET US KNOW”. These boxes will provide you with some interesting and relevant additional information. Again, you will get “CHECK YOUR PROGRESS” questions. These have been designed for you to self-check your progress of study. It will be helpful for you if you solve the problems put in these boxes immediately after you go through the sections of the units and then match your answers with “ANSWERS TO CHECK YOUR PROGRESS” given at the end of each unit.

BLOCK INTRODUCTION

This course contains fifteen units in two blocks. **Block I** contains 7 units. These units have discussed the basic concept of computer communication and networking. Unit 1 deals with the basic concept of data communication including types of signals, bandwidth, frequency domain transmission etc. Unit 2 introduces data communication models like serial and parallel communication, simplex and duplex communication.The basics of digital transmission like modulation and demodulation, multiplexing, transmission errors, hamming distance are discussed in Unit 3. Introduction to computer networks like goals of computer network, switching techniques, categories of networks, network topologies, transmission type are discussed in unit 4. Unit 5 discusses the network models mainly ISO-OSI and TCP/IP models. The concept of physical layer including transmission media, repeaters are discussed in unit 6. Unit 7 discusses the details of Data Link layers.

BACHELOR OF COMPUTER APPLICATION

Data Communication and Computer Networks

DETAILED SYLLABUS

BLOCK-1

Pages

Unit 1: Introduction to Data Communication 7–24
Introduction, Signals, Types of Signal: Analog, Digital, Block representation of Data Communication System, Definition of Protocols and Standards, Channel, Bandwidth, SNR

Unit 2: Types of Data Communication 25–34
Data Communication, Parallel and Serial Communication, Modes of Communication: Simplex, Half Duplex, Full Duplex; Asynchronous and Synchronous Communication

Unit 3: Digital Transmission Fundamentals 35–56
Digital Signal (PCM, Delta Modulation, Adaptive Delta Modulation, ASK, PSK, FSK), Multiplexing and De-multiplexing, Detection and Correction of Transmission Errors (Parity and Hamming Code Technique), Data Compression and Encryption, Concept of BCH Coding

Unit 4:Introduction to Computer Networks 57–85
Computer Network: Definition, Goals, Structure; Broadcast and Point-To-Point Networks; Types of Networks (LAN, MAN, WAN), Network Topologies, Network Protocols and Standards, Networking Switching Techniques

Unit 5: Network Models 86–99
Network Reference Models, OSI Reference Model, TCP/IP reference Model

Unit 6: Physical Layer 100–116
Physical Layer, Transmission Media, Guided and Unguided Media, Repeaters, Hub

Unit 7: Data Link Layer 117–157
Framing, Error & Flow Control, Data Link Protocols

UNIT 1: INTRODUCTION TO DATA COMMUNICATION

UNIT STRUCTURE

1.1 Learning Objectives
1.2 Introduction
1.3 Data Communication
    1.3.1 Components of Data Communication Systems
1.4 Types of Connection
1.5 Signals
    1.5.1 Types of Signals
    1.5.2 Wavelength
    1.5.3 Time and Frequency Domains
1.6 Composite Signals
1.7 Fundamental Frequency
1.8 Bandwidth
1.9 Digital Signals
1.10 Transmission Of Digital Signal
1.11 Broadband Transmission
1.12 Signal to Noise Ratio(SNR)
1.13 Let Us Sum Up
1.14 Further Reading
1.15 Answers to Check Your Progress
1.16 Model Questions

1.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- define data communication
- describe the different types of communication
- describe the modes of transmission
- describe component of communication
- explain different types of signals

1.2 INTRODUCTION

In this unit, we will learn the basics of data communication, signals and the different types of signals. Topics like wavelength, bandwidth, frequency and

time domain are discussed in this unit. Digital signals and transmission of digital signals along with basic concept of signal to noise ratio is covered in this unit. In the next unit, we will explore the concept of data communication modes.

1.3 DATA COMMUNICATION

Data communication is fundamental in today's day to day communication. It is the backbone of the entire model on which Internet works. The terminology 'data' technically implies information that is stored in digital form. In data communication, the fundamental principles governing how the data is communicated across networks of different devices until it reach the intended receiver is framed and articulated. It ensures that interoperability and interpretation of the data across different devices remains uniform. The different set of rules that guide the above objective is popularly coined by the term protocols. Thus protocols are the set of rules that guide how data is carried across networks of devices until it reaches the intended receiver retaining the integrity of the data. With increasing number of users in internet and more and more devices being connected into the system as an entity, the management of such data in the network demands new protocols. However, how different devices are used to forward the packet from one to another and how these are interconnected is also critical. The structure in which the devices are connected so that maximum efficiency of the same can be achieved is the topology of the network. Different devices sharing the same medium cannot access the medium at the same time. If they do so there will be significant collisions between different systems. It is therefore necessary to frame rules and regulations guiding their access mechanism of the shared network. This is called *Medium Access Protocols* of the network. In the initial stages of network development, there were no standards. Any company could practically manufacture and deploy their own devices following their own standards(protocols). However, the difficulty with such approaches has been the interoperability between different devices. A device "X" from a company "A" could not talk to device "Y" from company "B", because the design principles of operation in the device "X" and "Y" are not same. Hence, the way "X" interprets or understands the data may be drastically different from the way "Y" interprets the data. Therefore there has to be a "standard" to be defined so that the uniformity in the devices manufactured by one comply with the devices of another operating in the

same domain is maintained. This has brought in the concept of layered architecture. The entire principle of operation of a network now can logically be identified into different layers. All these layers being put together in the order of processing becomes a model. The two popular models are OSI and TCP/IPl. Each of the models would be discussed in the later section. Internet today is based upon the TCP/IP model framework. Such a framework provides the basis of data transfer in varied forms between source and destination. Figure 1.1 shows the diagrammatic representation between the throughput and different services.

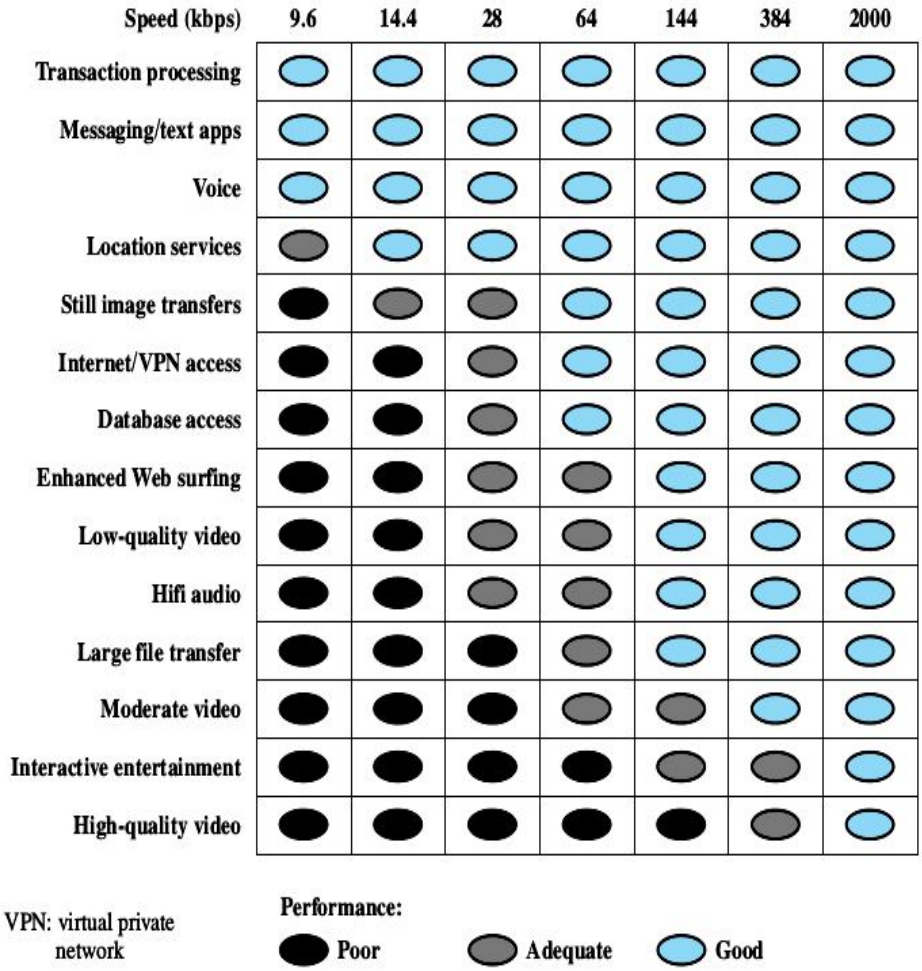


Figure 1.1: Throughput of different data formats

Figure 1.1 makes it clear that the bandwidth demand for different services are not same. Hence, the data of different services are guided by different set of rules and regulations called protocols in each layer of the model.

1.3.1 Components of Data Communication System

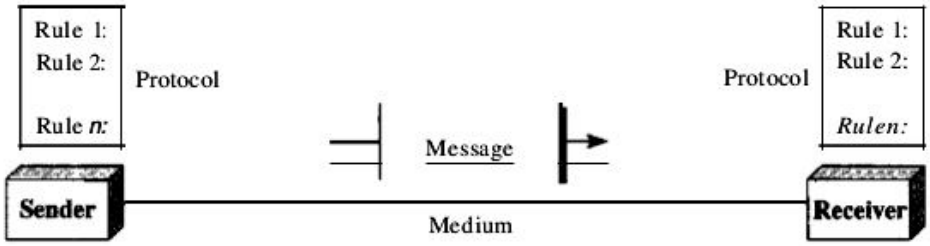


Figure 1.2 : Components of data communication system

There are five components in data communication system. These are:

a. **Message.** The actual data that is generated in the application layer by an user using certain applications such as email etc. are messages. These are generated on the senders and receivers machine. The integrity of the message should be retained throughout the communication process.

b. **Sender.** The origin of the message destined to a particular receiver on the network is the sender. The sender may generate data of any size but corresponding fragmentation has to be taken care of by other layers so that the size of the data becomes compatible for transmission. The sender popularly interacts using the applications falling in the application layer of TCP/IP model.

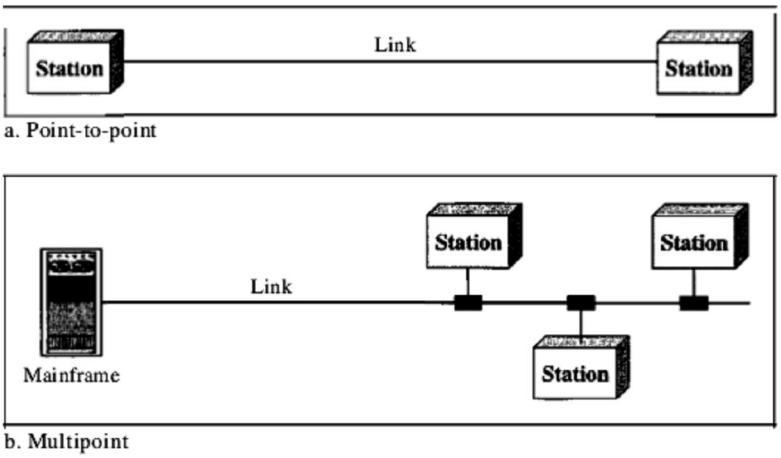
c. **Receiver.** The receiver is the device that accepts the message. It can be a computer, workstation, telephone handset, television, and so on.

d. **Transmission medium.** The network path through which the message propagates is the transmission medium. Examples of transmission medium are the twisted pair cable, coaxial cable, fiber optic cable etc.

e. **Protocol.** These are sets of rules and regulations that guide the communication process. It enables a sender to interpret the data for transmission that provide the format of the data, data coding etc. Any device within a given layer is capable of interpreting the data and processing and facilitating propagation using these rules and regulations. These rules also cover data rate.

1.4 TYPES OF CONNECTION

In computer network, systems are either connected in point to point or multipoint format as shown in Figure 1.3. In point to point connection one device is linked to other on one end. For example the unidirectional transmitter antenna can be mounted on a roof top in line of sight form with the unidirectional receiver antenna. If any one of the antenna is out of alignment, the communication is going to fail.



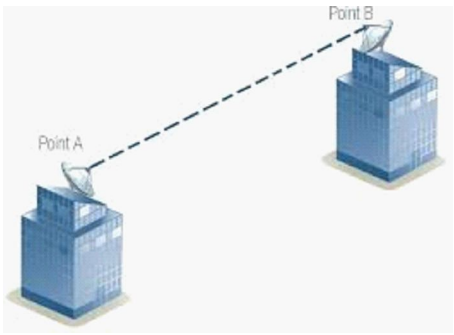


Figure 1.3: Basic connection types

PPP is popularly used by Internet Service Providers(ISP) for customer dial up access to internet.

In multipoint communication, a primary source is connected to a number of secondary sources as shown in Figure 1.4. Here the primary source broadcasts the message in the link which is received by all other systems on the network. However, only that system whose address is carried on the message actually processes the packet while others discard it.

1.5 SIGNALS

Signal is a function of physical quantities that demonstrate variation with respect to time and space. It provides some information about the status of a system or conveys a message from the source to the destination. Examples of signals are such as radio waves, electromagnetic waves, infra red waves etc. In figure 1.4 the basic block diagram of a communication system is shown.

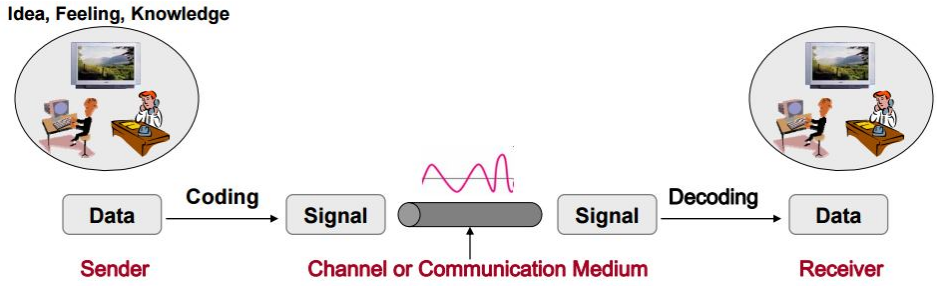


Figure 1.4: Block diagram of communication system

The channel or communication medium carries signal either in analog or digital form. If the signal is analogous to real world signals then it is an analog signal; otherwise, if it is converted to a binary form we call it a binary digital signal.

1.5.1 Types of Signals

Analog signal: In Analog form, the signal is continuous in time. Given a point "A" and another point "B", the value of an analog signal may be infinite in nature. Both analog and digital signals exhibit periodic and aperiodic property. In periodic signals , a pattern exists that repeats over the time. One complete pattern is called a cycle. Using one cycle the entire signal can be regenerated. Periodic analog signals can be again either simple or composite. A simple analog signal cannot be further decomposed into any simpler signals. However, a composite signal can be further decomposed into simpler signals(i.e a composite signal is combination of several simpler analog signal).

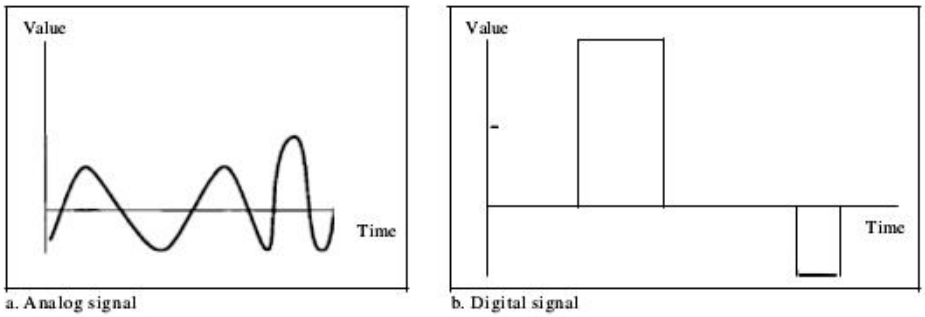
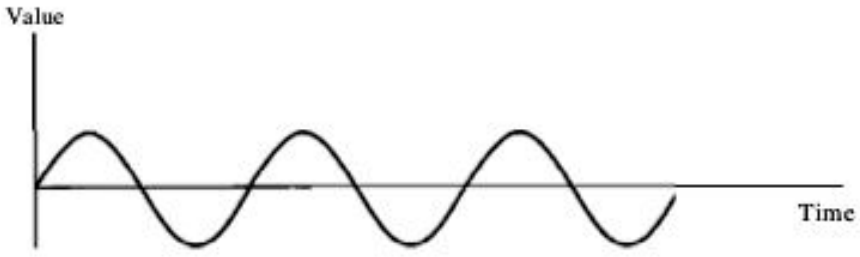


Figure 1.5: Analog and Digital signal representation

Sine and cosine waves are the most common form of periodic signals. A signal f(t) is said to be periodic with period T0 if

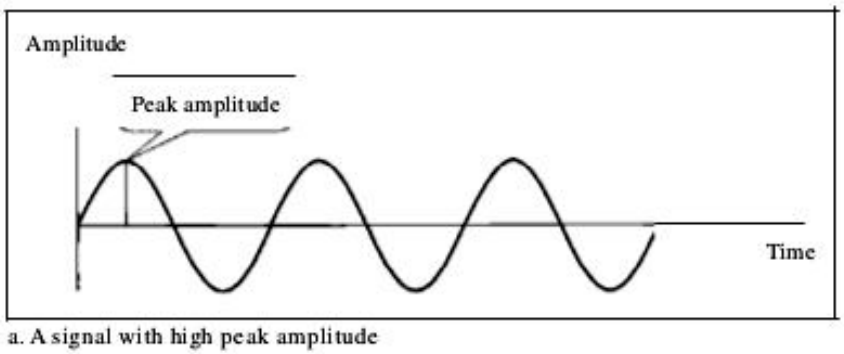


f(t)=f(t + T0) for all t.

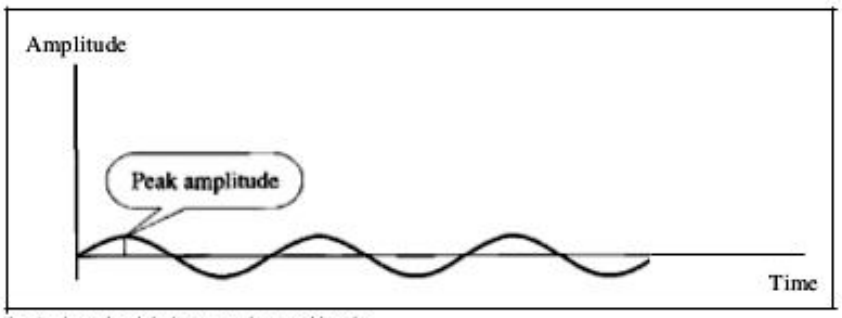
Figure 1.6: A periodic sinusoidal signal

A sine wave is characterized by three parameters: (a) peak amplitude, (b) frequency and (c) phase.

(a) Peak amplitude: Peak amplitude of a signal indicates the maximum strength of the signal. The strength(for electric signals measured in volts) vary with respect to time and at a certain instant of time it attains the maximum value and eventually the amplitude starts decreasing.



a. A signal with high peak amplitude

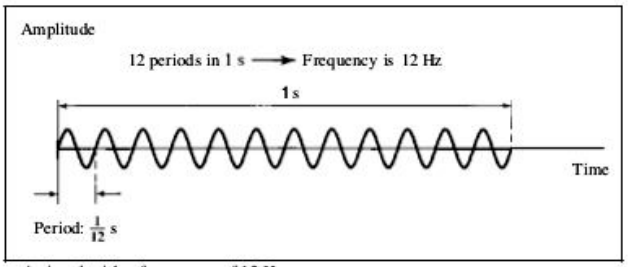


b. A signal with low peak amplitude

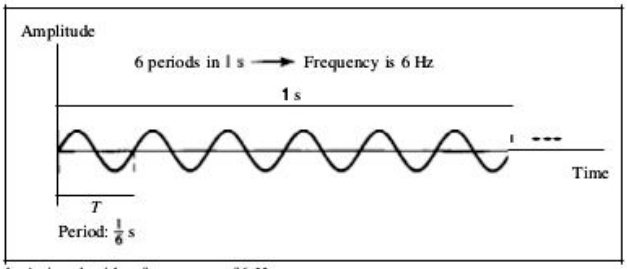
Figure 1.7: Two signals with same phase and frequency but different amplitudes.

(b) Time Period and Frequency: Time period represents the amount of time required to complete one cycle of a periodic signal. Time period is the inverse of frequency i.e

f=1/T ,Frequency is the number of cycles in 1 sec(unit time).



a. A signal with a frequency of 12 Hz



b. A signal with a frequency of 6 Hz

Figure 1.8: Two different periodic sinusoidal signal of different frequencies

Example :-The frequency of power used in our home is 60Hz. Determine the period of this sine wave.

Solution:-T=1/f =1/60=0.0166s

(c) Phase: The position of the waveform in relative to time 0 at different instants of time is phase. Phase is the starting reference of a signal. Phase is either measured in degree or radians. A phase shift of 3600 corresponds to a shift of complete period.

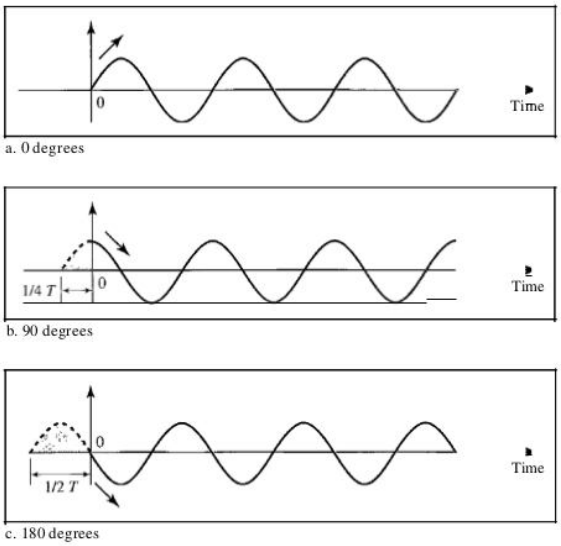


Figure 1.9: Sine waves with same amplitude and frequency, but different phases.

Looking into the figure 1.9 the following conclusions can be drawn :

a. The amplitude of a periodic sine wave at phase 0 is 0 and goes on to increase until it reaches a peak point.
b. The sinusoidal wave starting with a phase of 900 starts with the maximum amplitude and eventually the amplitude starts decreasing.
c. The sinusoidal signal with a phase of 1800 starts with a 0 amplitude and the amplitude starts further decreasing.
d. It can be further observed that with phase of 1800, the signal shifts towards left by a half cycle.
e. If v is equal to the velocity of light c, then for a radio wave (with speed equal to that of light c) P=c/f

1.5.2 Wavelength

The wavelength of a signal is the distance a simple wave can travel in one period. The wavelength can be derived from the following formula:

$$Velocity(v) = \frac{Wavelength(\lambda)}{Time\ period(T)}$$

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

$$\Rightarrow v = f \times \lambda \Rightarrow \lambda = \frac{v}{f}$$

Therefore, it is observed that the wavelength of a signal is dependent on the frequency and the medium. Whatever distance a simple signal can traverse in one period is wavelength of the signal.

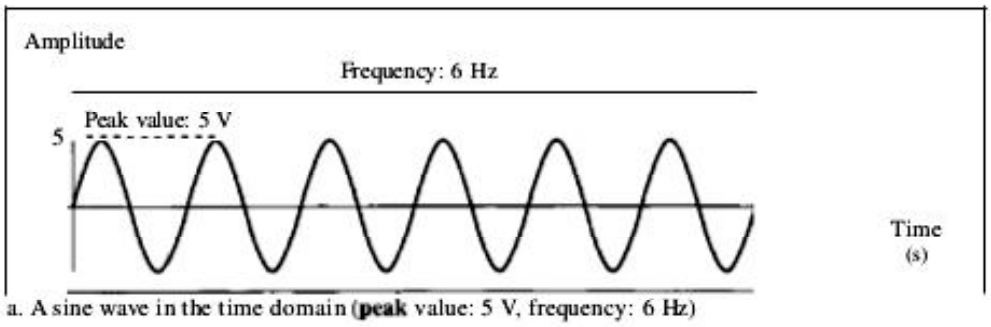


CHECK YOUR PROGRESS

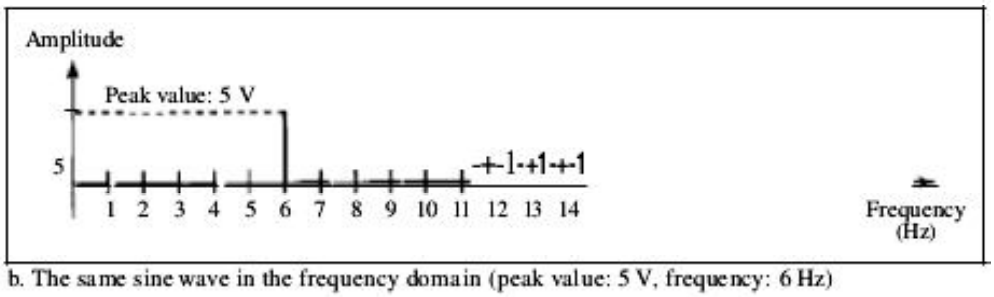
1. What is an analog signal?
2. What are frequency and phases?
3. How are wavelength and time period related to each other?

1.5.3 Time and Frequency Domains

Signals may be represented either in time or frequency domains. A signal in time domain representation shall show instaneous variations of amplitude with respect to time. A frequency domain signal, on the other hand, shall show variations in amplitude with frequency. A time domain signal may be converted to frequency domain by using Fourier transform.

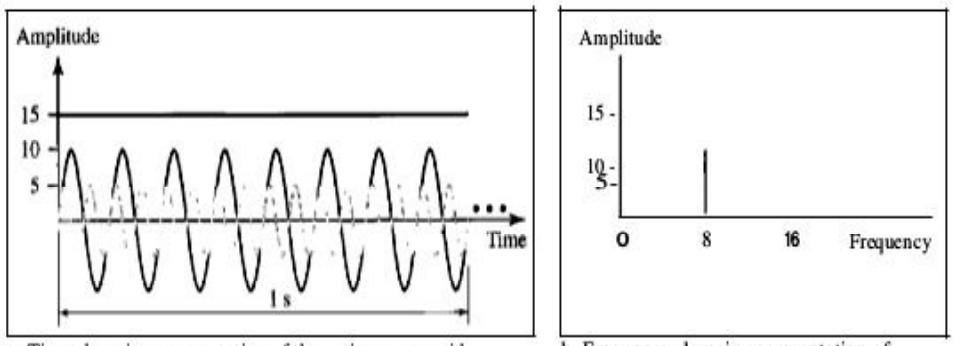


a. A sine wave in the time domain (peak value: 5 V, frequency: 6 Hz)



b. The same sine wave in the frequency domain (peak value: 5 V, frequency: 6 Hz)

Figure 1.10: Time and Frequency Domain representation of a signal



a. Time-domain representation of three sine waves with frequencies 0, 8, and 16

b. Frequency-domain representation of the same three signals

Figure. 1.11 : Time domain representation of three sine waves

1.6 COMPOSITE SIGNALS

Composite signal is a combination of simple signals. A composite signal can be decomposed into set of simple signals using different approaches such as Fourier transform. Whenever a communication is taking place over telephone , the form of signal set is composite. Had it been a simple wave the receiver would have just received a buzzing sound.

Example:

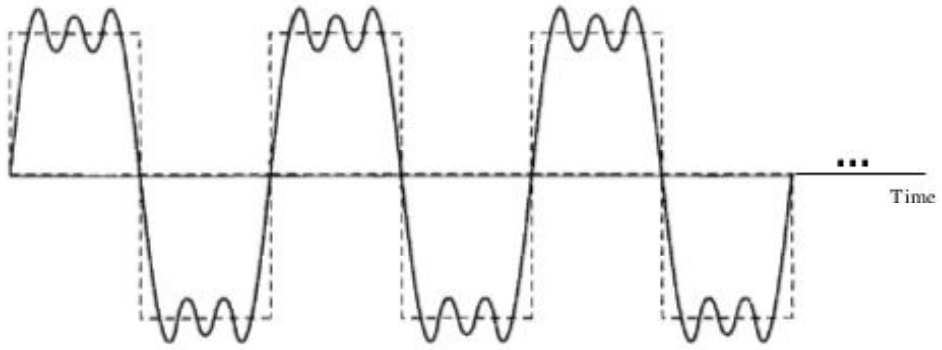


Figure 1.12: Periodic Composite signal

The above composite signal after decomposition into simpler signals looks as follows:-

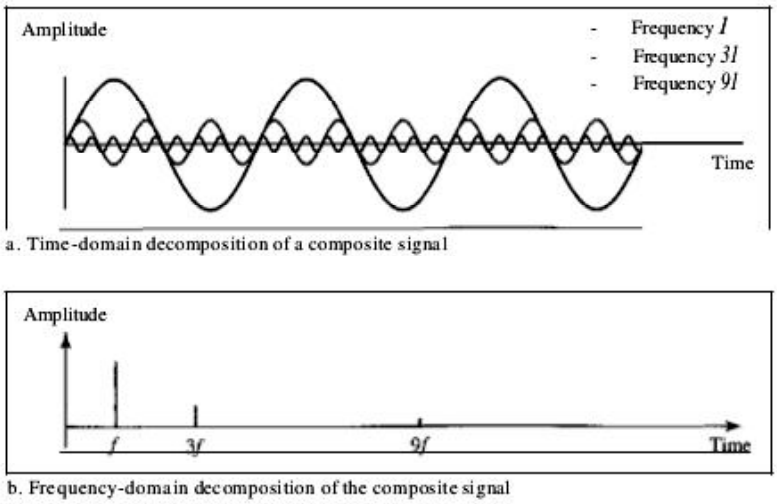


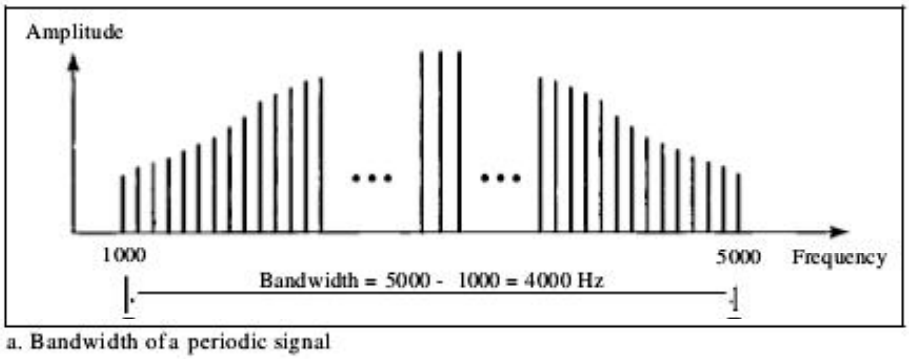
Figure 1.13 : Decomposition of above composite signal into different simpler signals

1.7 FUNDAMENTAL FREQUENCY

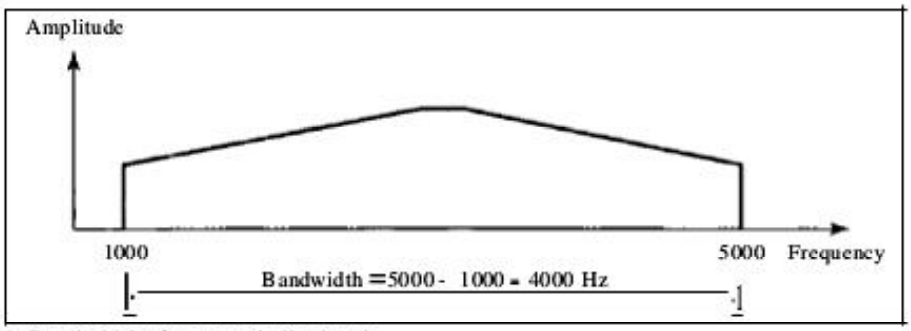
The amplitude of the signal with frequency 'f' is same as the amplitude of the composite signal. However, the amplitude of the signal with frequency 3f is is having 1/3rd of the amplitude of the signal with frequency 'f'. The signal with frequency 'f' is called the fundamental frequency or first harmonic. Likewise, the signal with frequency 3f have frequency 3 times the fundamental frequency and signal with frequency 9f have frequency value with 9 times the fundamental frequency and amplitude value of 1/9th of f.

1.8 BANDWIDTH

Given a composite signal in hand, the range of frequencies available is the bandwidth of the signal. Bandwidth represents a group of frequencies where the amplitude of the signal remain constant. If the frequencies of the composite signal varies from 1000KHz to 5000KHz, the bandwidth of the signal is going to be (5000-1000) KHz.



a. Bandwidth of a periodic signal



b. Bandwidth of a nonperiodic signal

Figure 1.14: Bandwidth of periodic and non periodic composite signal

Example

A composite periodic signal is decomposed into five sine waves with frequencies of 100,300,500,700 and 900Hz, what is its bandwidth?

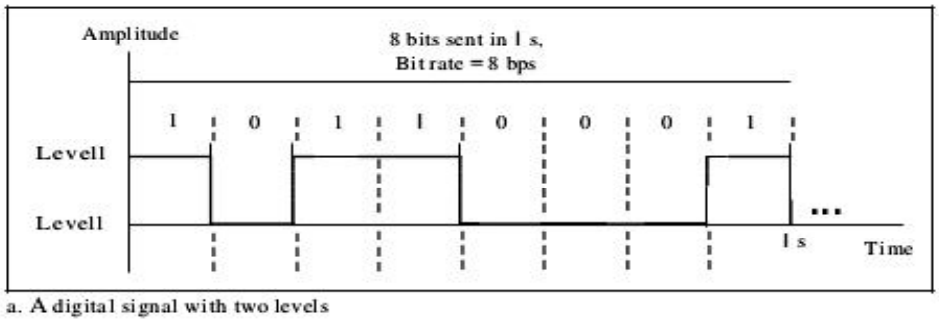
Solution: Let fH be the highest frequency, fL be the lowest frequency and B the bandwidth , Then

B=fH- fL T=900-100=800 Hz.

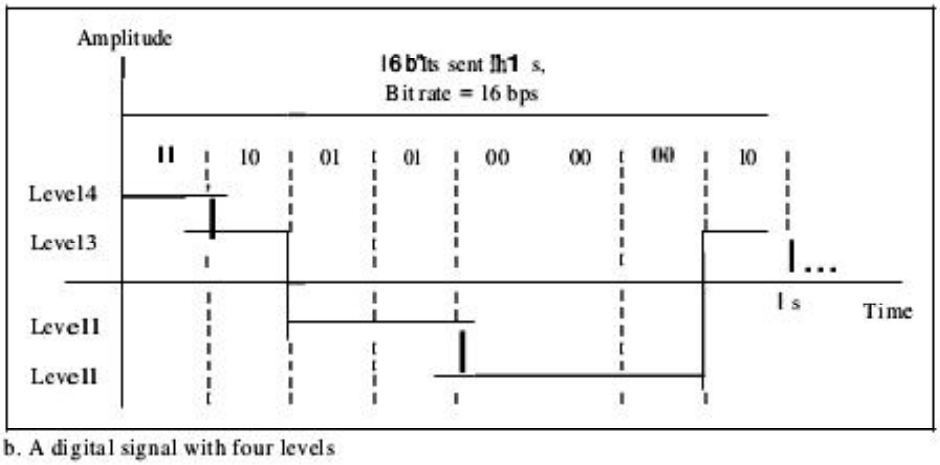
The spikes of the spectrum are at 100,300,500,700 and 900Hz.

1.9 DIGITAL SIGNALS

In this type of signal, information are represented in binary form either as 0 or 1. There can be more than one levels in a digital signal.



a. A digital signal with two levels



b. A digital signal with four levels

Figure 1.15: Two digital signals with 2 levels and 4 levels

As shown in Figure 1.15, a digital signal with one level carries 1 bit in a bit interval, whereas in two level representation each bit interval carries 2 bits each. Thus the number of bits in a given interval is given by , where N is the total number of levels in a signal.

Bit Rate:

The number of bits transmitted per second is the bit rate. Figure 2.16 shows the bit rate for two signals.

a. Bit length: Bit length is the distance one cycle covers in the transmission medium.

Bit length= propagation speed x duration

1.10 TRANSMISSION OF DIGITAL SIGNAL

For transmission of digital signals, there should be a definite medium. Digital signal can be transmitted in either BaseBand or BroadBand.

Baseband Transmission:

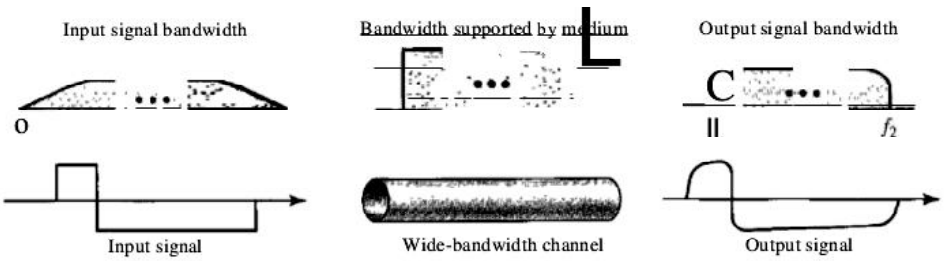


Figure 1.16: Baseband Transmission

This type of transmission have low pass channel,i.e a channel that starts from zero bandwidth. This usually is the case when the systems are connected in peer. For example a system in a LAN to a switch or one system to another.

Low Pass channel with wide Bandwidth

For retaining the structure of a non periodic signal , a wider spectrum is necessary. As such an infinite spectrum is necessary. However, the amplitude at the border of the non periodic digital signal is not steep; hence they can be ignored limiting the spectrum size. Examples of such medium are coaxial cable, fiber cable etc.

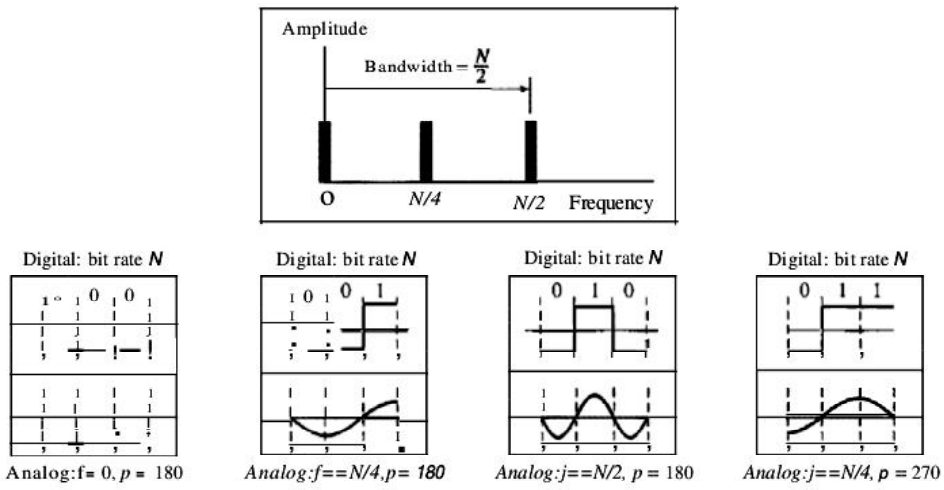
Low pass channel with limited Bandwidth

Here the digital signal is approximated to analog signal. For the maximum changes in transmission of digital signal the sequence has to be of the form 010101... Or 1010101....

To simulate the above , an analog signal of frequency fs is required which is at least greater or equal to the two times of highest frequency in the analog signal. If fm is the highest analog frequency in a signal then fs $\geq$ 2fm where fs is called the sampling frequency. The inverse of sampling frequency fs is called sampling time.

$$T_s = \frac{1}{f_s}$$

Time is needed convert an analog signal into digital sampling . Figure 1.17 shows how a digital stream of 3-bit pattern can be represented in analog form.



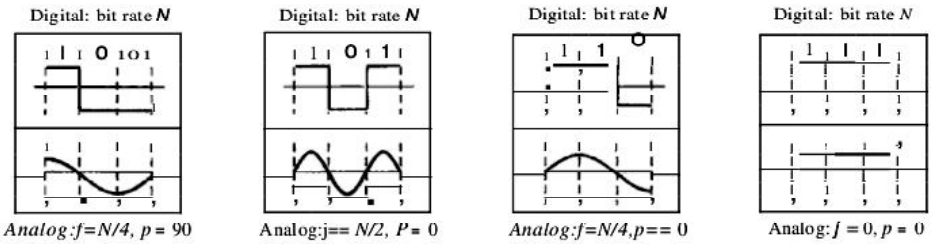


Figure 1.17:Rough approximation of a digital signal using the first harmonic

1.11 BROADBAND TRANSMISSION

In this type of communication the digital signal is converted into analog form and modulated with a high frequency carrier for transmission.

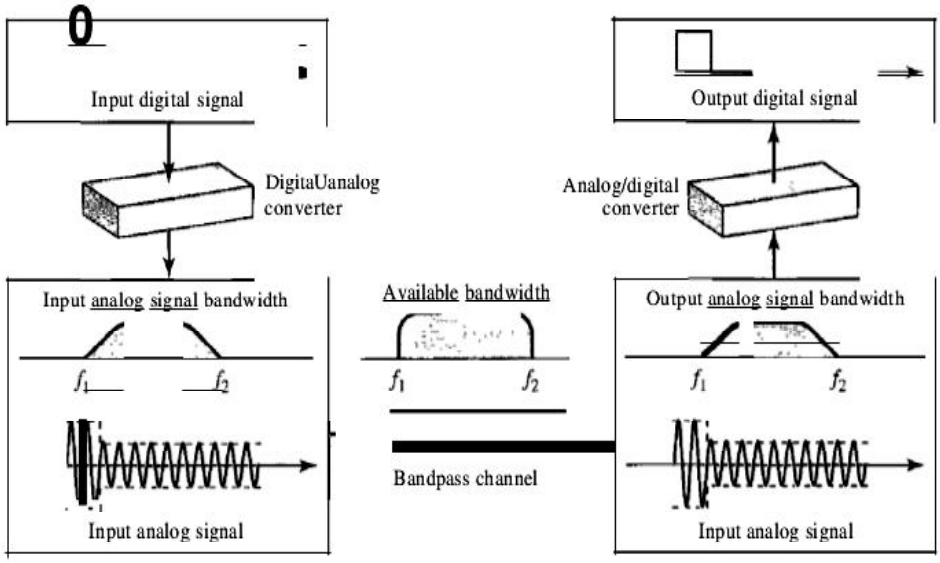


Figure 1.18: Modulation of a digital signal for transmission on a bandpass channel

Noise:

Noise is a cause of impairment of signal used for communication. Noise gets induced due to different factors in the environment such as thunder, induction between two parallel wires during transmission etc.

1.12 SIGNAL TO NOISE RATIO(SNR)

The signal to noise ratio is defined as the average signal power (P) and average noise power (N) i.e. SNR=P/N. SNR gives us the strength of the signal with respect to noise power. SNR is expressed in log scale as

$$SNR\ dB = 10 \log_{10} SNR$$



1.13 LET US SUM UP

- In data communication, the fundamental principles governing how the data is communicated across networks of different devices until it reach the intended receiver is framed and articulated.
- There are five components in data communication system : Message, Sender, Receiver, Transmission medium, Protocol.
- Signal is a function of physical quantities that demonstrate variation with respect to time and space.
- Signals may be represented either in time or frequency domains.



1.14 FURTHER READING

- Kennedy, G. and Davis, B. (1985) *Electronic communication systems*. (Vol. 20, b. 21). Tata McGraw-Hill Publishing Co. Ltd., New Delhi.
- Forouzan, A. Behrouz (2006). *Data communications & networking* (sie). Tata McGraw-Hill Education.



1.15 ANSWERS TO CHECK YOUR PROGRESS

Answer 1: A signal whose characteristics such as amplitude varies with time is an Analog Signal.

Answer 2: When we observe a signal w.r.t amplitude, than the signal is in time domain

Answer 3: $\lambda = C / V$ therefore, λ = cT, where T-time period



1.16 MODEL QUESTIONS

Q 1. What is the relationship between time period and frequency?

Q 2. We send a digital signal from one station to another station through a switch. Is this a LAN or a MAN?

Q 3. What is the phase shift for the following ?

i. A sine wave with the maximum amplitude at time zero.

ii. A sine with maximum amplitude after 1/4 cycle.

iii. A sine wave with zero amplitude after 3/4 cycle and increasing.

Q 4. A periodic composite signal contains frequencies from 10 to 30KHz , each with an amplitude of 10V. Draw the frequency spectrum.

Q 5. If the bandwidth of the channel is 5 kbps, how long does it take to send a frame of 100000 bits out of this device?

Q 6. Assume that a voice channel occupies a bandwidth of 4khz. We need to multiplex 10 voice channels with guard bands of 500Hz using FDM. Calculate the required bandwidth.

Q 7. We need to transmit 100 digitized voice channels using a pass-band channel of 20Khz. What should be the ratio of bits/Hz if we use no guard band?

Q 8. Distinguish between a link and a channel in multiplexing.

****

UNIT 2: DATA COMMUNICATION MODES

UNIT STRUCTURE

2.1 Learning Objectives
2.2 Introduction
2.3 Data Communication
2.4 Types of Communication
2.5 Modes of Communication
2.6 Let Us Sum Up
2.7 Further Reading
2.8 Answers to Check Your Progress
2.9 Model Questions

2.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- define data communication models
- describe the different types of modes of data communication
- describe the modes of transmission

2.2 INTRODUCTION

In the previous unit you have been introduced to the concept of data communication. Signals and different types of signals like analog and digital have been covered in the previous unit besides providing introductory definitions of many important concepts of data communication.

In this unit you will be introduced to different data communication modes. You will also learn about the types of communication - parallel and serial communication including synchronous and asynchronous types of communication. In addition to these types of communication the different communication modes such as simplex, half-duplex and full duplex will also be introduced in this unit.

Once you are familiar with these basic modes of data communication, you will be introduced with digital transmission fundamental concepts like modulation and multiplexing in the next unit. Concepts like data compression and encryption are also introduced in the next unit.

2.3 DATA COMMUNICATION

Data communication means sending data or information from a source to a receiver. This data or information can be anything. This information can be a simple data file, or an audio file or video file. Data communication has become an essential part of our lives. Transferring of data from one computer to another computer is an example of data communication.

Communication usually has three important parts – a *source*, a *destination* and a *communication* channel. Let us look at these three components:

- **Source:** This is where the data originates. Generally, it is a computer but it can be any electronic device like a mobile phone, an audio recorder etc.
- **Destination:** This is where the data is absorbed. It can be a computer system or a television set.
- **Communication Channel:** This is the entity which is used to send data or signals from a source to a destination. This can be an optical fiber or a wireless medium. There are provisions for the data being lost in the communication channel.



Figure 2.1: A simplified communication system

A simplified communication system can be seen in figure 2.1. The data or signal is sent from the source to the destination using the communication channel. Let us now look at what could be the different types of communication.

2.4 TYPES OF COMMINICATION

The sender sends data to the receiver through a communication channel. The data could be sent using one communication unit or many units. Depending on these differences, communication can be broadly divided into two types: serial communication and parallel communication. The

communication can be synchronized using a clock or some other mechanisms. Depending on these, the serial communication is again subdivided into two major categories: synchronous communication and asynchronous communication.

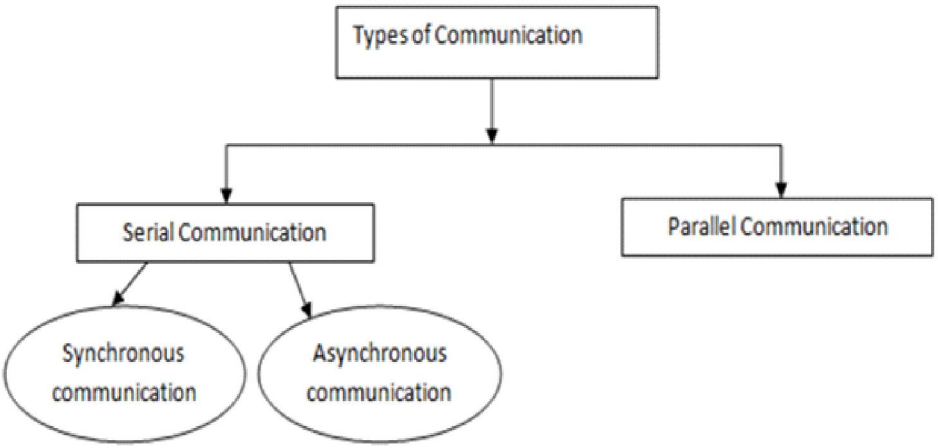


Figure 2.2: Types of Communication

Figure 2.2 shows the different types of communication. Let us now try to understand the definitions and concept of serial and parallel communications.

Serial communication is the sequential transmission of data using a single communication unit like a single copper wire or an optical fiber. In the digital domain, the data is represented as bits and can be sent over using a single electrical wire sequentially.

Parallel communication is the concurrent transmission of data using many communication units for transmission of data. In the digital domain, multiple data bits are sent simultaneously by using multiple electrical wires.

In serial communication, since there is only one communication unit so less data is sent in one transmission, whereas in parallel communication more data can be sent since we can use multiple communication units for one transmission.

To find the relative suitability between serial and parallel transmission, let us take an example. Suppose we have a parallel connection with 4 wires and a serial connection with 1 wire. It would seem to us that the parallel connection could send more data than serial communication and faster

than serial communication. But less data means the error in data transfer will be less as compared to parallel communication. This means that the error rates are higher in parallel communication corresponding to the increase in data.

Serial communication is preferred for sending data over longer distances due to less data error. It also has low cost and is easier to use than parallel communication. Also, this communication can be used when high data rates are not needed. Examples of serial communication are USB (Universal Serial Bus), Ethernet, Bluetooth etc. among many. Parallel communication is used for faster data transfer since they use multiple wires and can transmit data on all the wires simultaneously. Examples of parallel communication are IDE hard disk connectors among many.

Serial communication can be again broadly divided into two categories: **Synchronous and Asynchronous communication.**

- **Synchronous communication:** In this mode of communication, the data bits are sent over a single communication unit, using a clock to synchronize the timing between the sending and the receiving communication devices.

  The clocks of both the sender and receiver are required to be at sync with each other in this communication. The sending and receiving time for data is already known to both parties and using this protocol the data are being transferred. The disadvantage of synchronous communication is that eventually the clocks may get out of sync and may have to be resynchronized from time to time.

- **Asynchronous communication:** In this mode of communication, the data bits are sent over a single communication unit, and start and stop bits are used before and after the data to signify the starting and ending of the data sent.
  The data will be sand witched between the start and stop bit in this communication. Since this communication does not use clocks, which is why it is necessary to use other means to signal the starting and ending of data to the receiver. For this purpose, the start bit is used to signal the beginning of data and the stop bit is used to signal the ending

of data. For example, if the data to be sent is '0010', then it can be sent using a stat bit '1' and an end bit 0' and would look like this : '**1**0010**1'.** The start and stop bit are shown in bold here.



CHECK YOUR PROGRESS

Q1. Define serial and parallel communication.

Q2. Describe the difference between asynchronous and synchronous communication.

2.5 MODES OF COMMUNICATION

The manner in which information flows between two communication devices is known as the communication or transmission mode. The communication modes deal with the way traffic flows from one computer to another. In simple terms, it is the way or direction in which the flow of information takes place. This can be in any one direction or in both the directions between two systems. Depending on these differences the communication modes are basically divided into three groups. These are simplex and duplex. Further duplex communication can be half duplex and full duplex.

Let us discuss these modes in detail now:

- **Simplex:** In this mode of communication mode the data can travel through in only one direction. In other words, we can say that the communication is unidirectional.

In simplex mode of communication, a device can either receive or transmit information but cannot do both. For example, a television set can only receive information from the satellite channels but not transfer any information in return.

Let us take another example to understand this concept better. Suppose we need to write a paper on Indian Freedom Movement. One of the most common ways now-a-days is to collect the information from internet. For that we may use a computer and enter the information we need to search through the keyboard. The keyboard is only used to send the information to the computer but it cannot receive any information in return from the computer. This has been illustrated in the Figure 2.3.

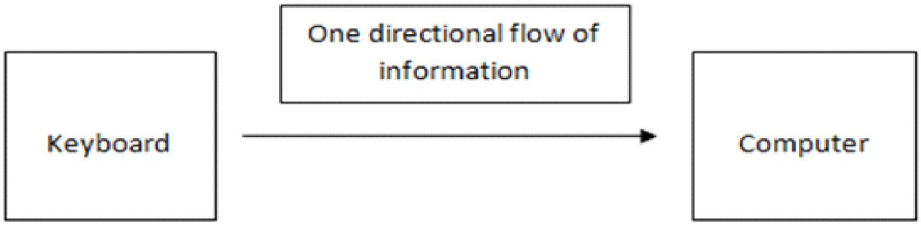


Figure 2.3 Example of Simplex communication

- **Half-Duplex:** In this type of communication mode the data can travel through in both directions but in only one direction at a time. In other words, we can say that this mode supports bidirectional transmission but can perform only one-directional transmission at a time.

In half-duplex mode of communication, a device can both receive and transmit information but cannot do both at the same time. That is, it can either receive or transmit at a particular instant of time. For example, a computer system cannot send information to another computer system while it is receiving information from that computer system. After the information has been received the computer system can now start to send information to the other computer system.

Let us take another example to understand this concept better. We may have seen many police personnel talking in their walkie-talkie. The police officers can listen to information or send some information through their walkie-talkie. But the act of both listening and speaking at the same time using the walkie-talkie cannot be done. This is why we mostly see them either holding it against their ear or in front of their mouth while communicating with their officers and saying the word “over” when they have finished delivering the information. This has been illustrated in the figure below.

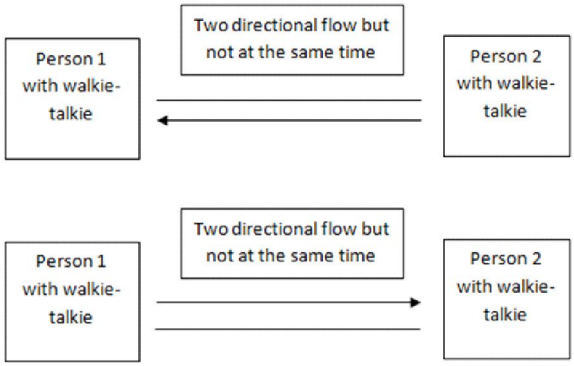


Figure 2.4 Example of half-duplex communication

- **Full-Duplex:** In this type of transmission mode the data can travel through in both directions at a time. In other words, we can say that is mode supports bidirectional communication and can perform communication in both directions at the same time.

In full-duplex mode of communication, a device can both receive and transmit information at the same time. It means that the device can receive information while sending some other information. For example, on a local area network that has full-duplex transmission, one computer system can transmit information while it is receiving data from another computer system which is also receiving information.

Let us take another example to understand this concept better. We have used the telephone for communication many times. We can both talk and listen using the telephone at the same time. Even though it may be difficult for us to hear and speak at the same time, but it is possible to use it in full-duplex transmission mode. This has been illustrated in the Figure 2.5.

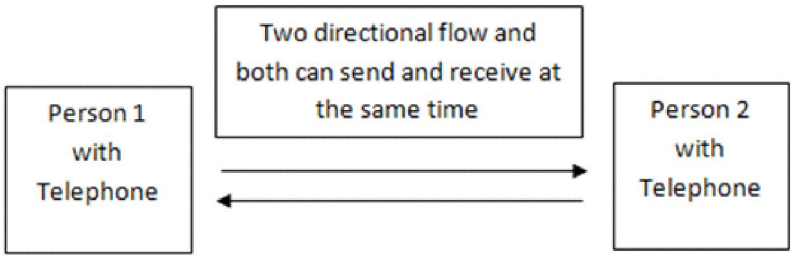


Figure 2.5: Example of full-duplex communication

The simplex communication is a one directional communication. The half-duplex is bi-directional but only one way communication at a time and full-duplex is bi-directional communication at all times. The concept of simplex, half-duplex and full-duplex communication can be illustrated further using the figure 2.6.

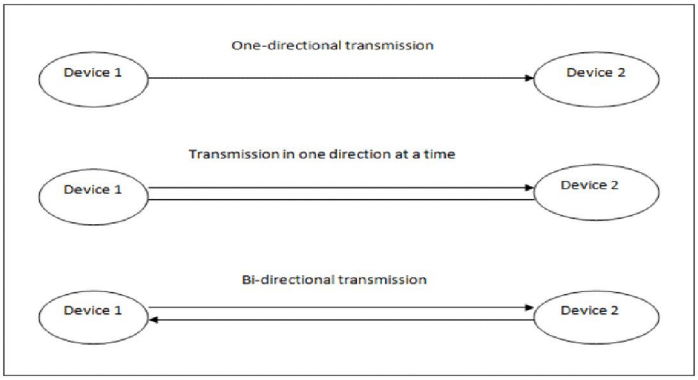


Figure 2.6: Simplex, half-duplex and full-duplex communication



CHECK YOUR PROGRESS

Q3. Define communication modes.

Q4. Describe the different types of communication modes.



2.6 LET US SUM UP

- Data communication means sending data or information from a source to a receiver.
- Communication usually has three important parts – a source, a destination and a communication channel.
- Serial communication is the sequential transmission of data using a single communication unit like a single copper wire or an optical fiber. In the digital domain, the data is represented as bits and can be sent over using a single electrical wire sequentially.
- Parallel communication is the concurrent transmission of data using many communication units for transmission of data. In the digital domain, multiple data bits are sent simultaneously using multiple electrical wires.
- In synchronous communication, the data bits are sent over a single communication unit, using a clock to make it happen at the same time concurrently. The timing between the sending and receiving devices are the controlling factors common to both ends. synchronize the timing between the sending and the receiving communication devices.
- In Asynchronous communication, the data bits are sent over a single communication unit, and start and stop bits are used before and after the data to signify the starting and ending of the data sent.
- There are two broad types of communication modes. They are: simplex and duplex, Duplex modes may be half-duplex and full–duplex.
- In simplex communication mode, the data travel through in only one direction.
- In half-duplex communication mode, the data travel through in both directions but in only one direction at a time.
- In full-duplex communication mode, the data travel through in both directions at a time.



2.7 FURTHER READING

- Tanenbaum, A. S. & Wetherall, D. (2011), *Computer Networks.* Prentice Hall India.
- Stallings, W. (2007), *Data and Computer Communications.* Pearson Prentice Hall.



2.8 ANSWERS TO CHECK YOUR PROGRESS

Answer 1: Data communication means sending data or information from a source to a receiver. This data or information can be anything. It can be a data file, or audio or video files. Communication usually has three important parts – a source, a destination and a communication channel.

Answer 2: Depending on whether data sent could be sent using one communication unit or many units, communication can be broadly divided into two types: serial communication and parallel communication. The communication can be synchronized using a clock or some other mechanisms. Depending on these, the serial communication is again subdivided into two major categories: synchronous communication and asynchronous communication.

Answer 3: The direction of flow of information between two communication devices is known as communication or transmission mode. The communication modes deal with the way traffic flows from one computer to another. In simple terms, it is the way or direction in which the flow of information takes place. This can be in any one direction or in both the directions between two systems. Depending on these differences the communication modes are basically divided into two groups. They are: simplex and duplex. Duplex communication is further subdivided into half-duplex and full–duplex.

Answer 4: There are two types of communication modes. They are: simplex and duplex. Duplex communication takes place in half and full–duplex. In simplex communication mode the data can travel through in only one direction. In other words, the communication is unidirectional. In half-duplex communication mode data can travel through in both directions but in only one direction at a time. In other words, we can say that this mode supports

bidirectional communication but can perform only one-directional communication at a time. In full-duplex communication mode the data can travel through in both directions at all time. In other words, we can say that is mode supports bidirectional communication and can perform communication in both directions at the same time.



2.9 MODEL QUESTIONS

Q1. Define data communication?

Q2. What are the different types of communication?

Q3. Give the different modes of communication.

Q4. What are the basic differences between half-duplex and full-duplex modes of communication?

Q5. Give some applications where half-duplex communication is used.

Q6. Give some applications where full-duplex communication is used.

Q7. Give some applications where simplex communication is used.

****

UNIT 3: DIGITAL TRANSMISSION FUNDAMENTALS

UNIT STRUCTURE

3.1 Learning Objectives
3.2 Introduction
3.3 Modulation and Demodulation
  3.3.1 Pulse Code Modulation
  3.3.2 Delta Modulation
  3.3.3 Adaptive Delta Modulation
  3.3.4 Amplitude Shift Keying(ASK)
  3.3.5 Phase Shift Keying (PSK)
  3.3.6 Frequency Shift Keying (FSK)
3.4 Multiplexing
  3.4.1 Frequency Division Multiplexing (FDM)
  3.4.2 Time Division Multiplexing (TDM)
  3.4.3 Wavelength Division Multiplexing (WDM)
3.5 Errors
  3.5.1 Single Bit Error
  3.5.2 Burst Error
3.6 Block Coding
3.7 Hamming Distance
3.8 Linear Block Codes
3.9 Let Us Sum Up
3.10 Further Reading
3.11 Answers to Check Your Progress
3.12 Model Questions

3.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- define modulation and demodulation
- describe the different types of modulation techniques
- describe multiplexing and demultiplexing
- describe data transmission error.

3.2 INTRODUCTION

In the previous units we have learnt about the concept of data communication. Different types and modes of communication were also discussed in detailed in the previous unit. In this chapter we will discuss about the fundamental concepts of modulation and multiplexing. Further, transmission error and certain coding techniques are also discussed in detail in this unit. In the next unit, we will explore the concepts of computer networks in detail.

3.3 MODULATION and DEMODULATION

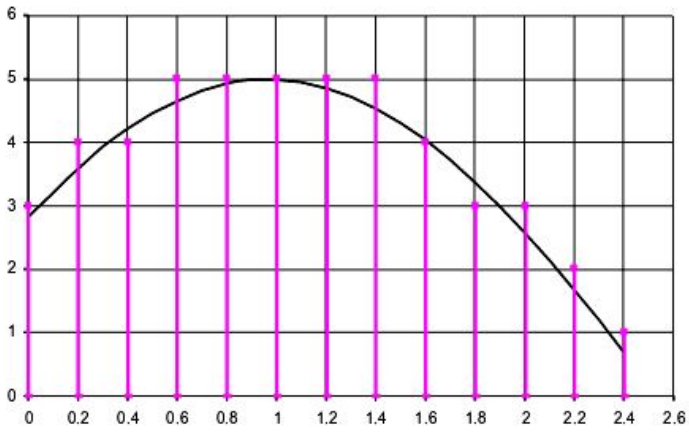
Modulation is a technique of transforming the original signal into a representation such that the transmission of the signal becomes possible in a given medium over a larger distance. The signal transmitted out of our mobile phones need to reach the mobile towers at a large distance. However, the information signal generated corresponding to the voice data is of low frequency. According to the laws of physics, the energy of the signal is quite low and may die off before reaching the tower. This signal is needed to be transformed into a high frequency carrier so that the signal can travel over a longer distance. This process of transforming a low frequency signal into a high frequency wave capable of transmission to a longer distance is modulation. The corresponding process of decoding (extracting from the high frequency wave) the information signal is demodulation. Modulation can be done both for analog and digital signals. In terms of analog signals, modulation may be classified amplitude modulation(AM), frequency modulation(FM) and phase modulation(PM). In case of AM, the amplitude of a high frequency carrier is varied or controlled by a low frequency modulating signal which bears the useful content. This control or variation results in miding of the signals and results in AM. Similarly, if the high frequency of a carrier is varied by a low frequency content having signal it results in FM. If the phase of a high frequency carrier signal is varied by a low frequency signal it results in PM. In terms of digital systems like a computer, the preferred scheme is pulse code modulation.

3.3.1 Pulse Code Modulation

In most of the other analog modulation technique the high frequency

sine signal is used as carrier, However, it is also possible to have a signal of pulse train as a carrier. Such modulation is called pulse modulation.

Pulse code modulation is a pulse modulation technique where the message signal is applied in analog form that is transmitted as sequence of binary bits. The background concept behind this is to represent message signal in digital form rather than analog form. The steps after input are sampling, quantization and coding at the transmitter side and just the reverse operations like decoding, resampling and recovery and low pass filtering at the receiver side. As shown in Figure 3.1 , the signal is quantized i.e instead of sending continuous value of the analog signal certain discrete values of the signal is read and rounded off to next nearest level in the quantization plane. In Figure 3.1 the spikes are different values read at instant 0,0.2,0.4..etc. The final spotted dot are the points mapped to the nearest quantized level. In the table below, for example, the actual amplitude value 2.8232 is mapped to quantized value 3 which in binary is 011.



Actual	2.823212	3.58678	4.207355	4.660195	4.927249
Code	3	4	4	5	5
Binary-Coded	011	100	100	101	101

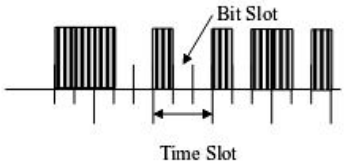
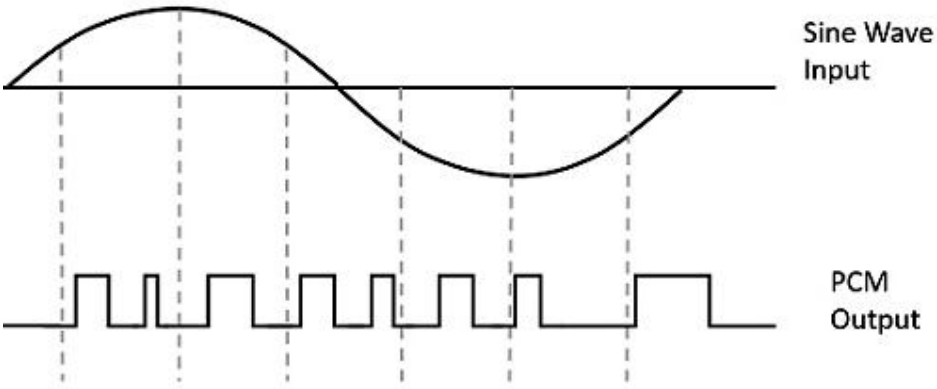


Figure 3.1 Pulse Code Modulation

The output of a PCM is an indirect representation of bit streams. The role of encoder as shown in Figure 3.1 is to encode into a binary code each quantization value. Figure 3.2 shows the input sine wave analog signal and the corresponding output of PCM modulator and the different stages of a PCM system.



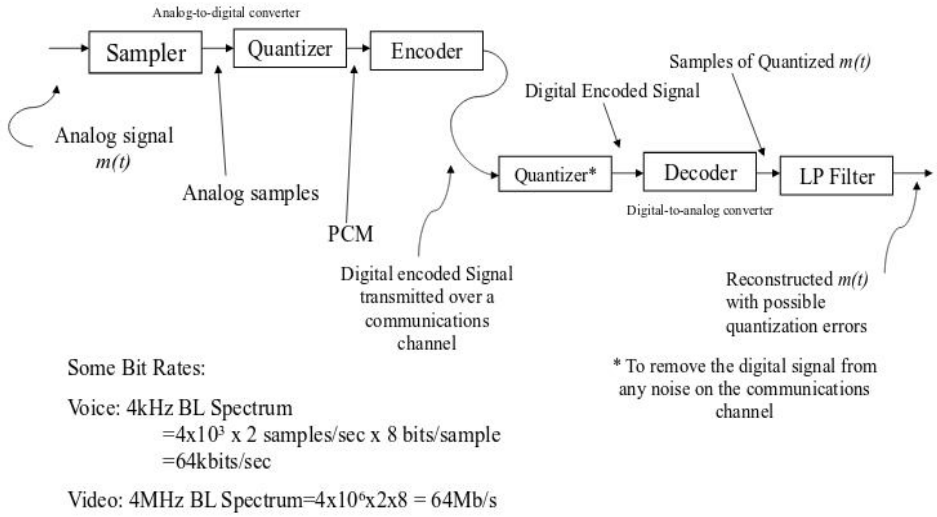


Figure 3.2: PCM System

3.3.2 Delta Modulation

In delta modulation, the oversample signal is approximated by a quantizer which varies by a steps size $\Delta = \frac{2v_{max}}{L}$ where V_{max} is the maximum strength of the input signal and L is the number of levels considered for quantization. The value of $L=2^R$ where R is the number of bits used for coding. Delta modulation is a low cost PCM and its primary objective is to minimize quantization noise. This is done during the quantization process where rf the quantize signal overshoots the sampled signal, it is reduced by $\triangle$. Similarly, if the

quantized signal falls below the message signal, it is lifted by $\triangle$. This complementing – supplementing mechanism involved during the quantization process is the basis of delta modulation. A block diagram of the delta modulation is shown in Figure 3.3. In this type of modulation , the sampling rate is significantly high and at the same time the step size after quantization is a very small value delta. The fundamental concept of delta modulation is to pick up samples that are not of much proximity but are instead at least separated by some value delta.

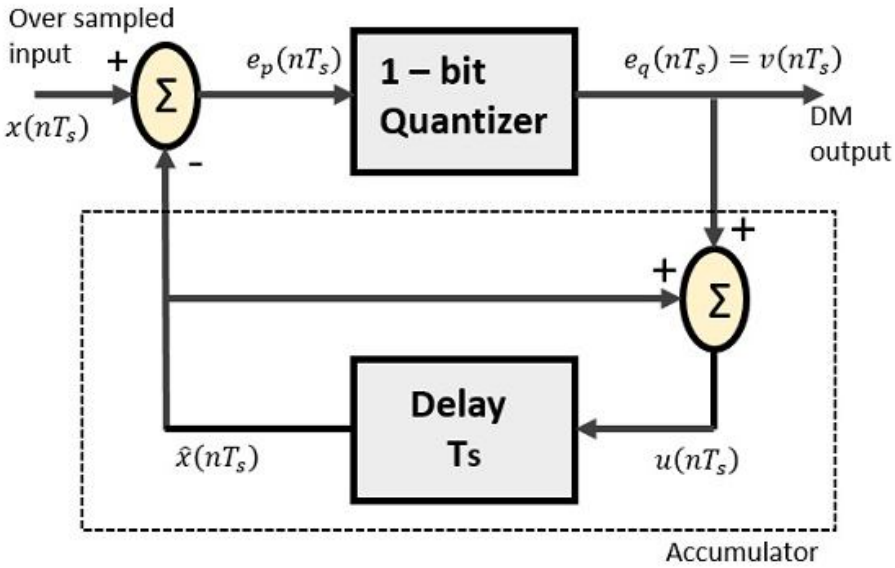


Figure 3.3 :Delta Modulation

3.3.3 Adaptive Delta Modulation

This is a delta modulation technique where the value of $\triangle$ changes adaptively as per the variations in the input signal. In DM, when $\triangle$ value is small, a distortion called slope overload noise is created. This is because of the fact that a quantizer with a small $\triangle$ fails to cover the rapid variations in the signal. On the other hand, if $\triangle$ is a larger granular noise produced. The quantizer with large $\triangle$ tracks over a flat portion of the waveforms and fails to capture the variations in the signal. This is illustrated in Figure 3.4.

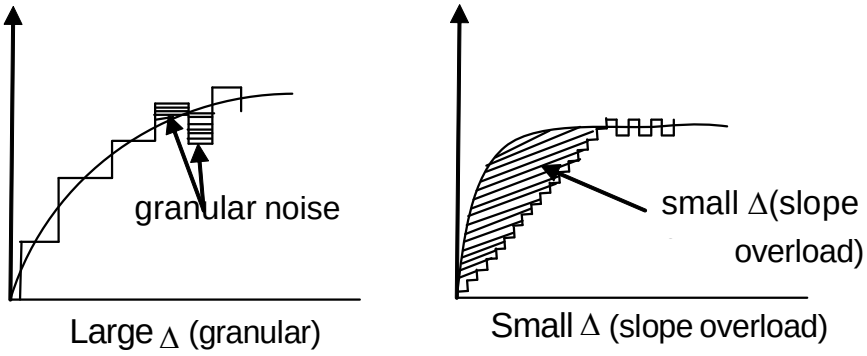


Figure 3.4 : Distortions generated by delta modulation

To overcome both granular noise and slope overload distortion, the size of the $\triangle$ is varied adapting. It improves the performance of the system dramatically. In most cases, the adaptively fixed $\triangle$ size is achieved by placing an integrator in the feedback path as shown in figure 3.5.

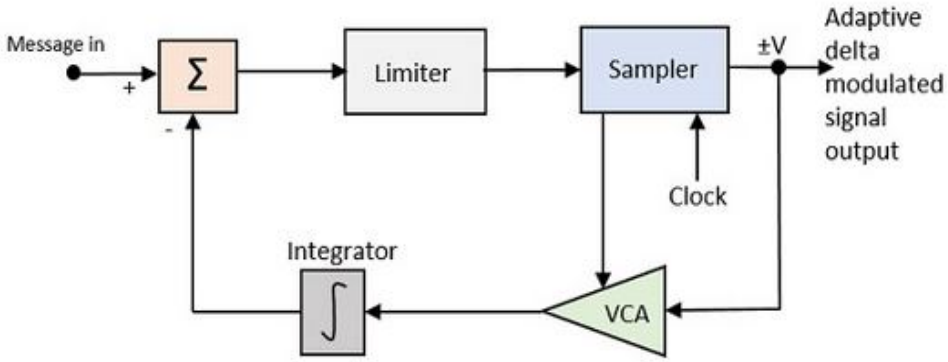


Figure 3.5 :Adaptive Delta Modulation

In the Figure 3.5 is Voltage Controlled Amplifier. The gain of the voltage controlled amplifier (VCA) is seen to be adjusted by the output signal from the amplifier.

3.3.4 Amplitude Shift Keying(ASK)

In ASK , two different values of the carrier wave is considered to represent two different binary values.

For a signal

$$\textbf{ASK} \qquad s(t) = \begin{cases} A\cos(2\pi f_c t) & \text{binary 1} \\ 0 & \text{binary 0} \end{cases}$$

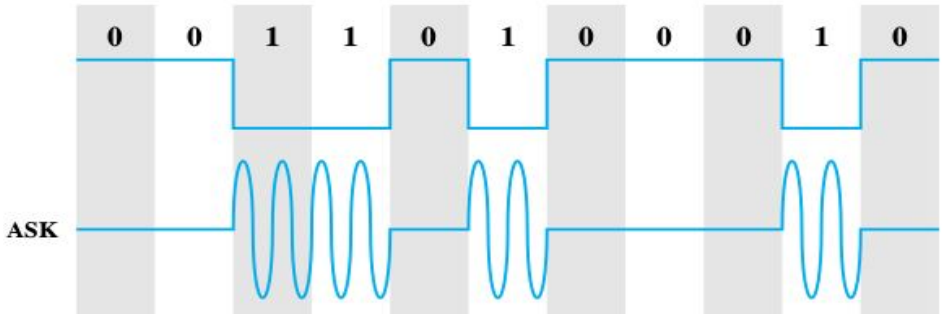


Figure 3.6:Amplitude Modulation

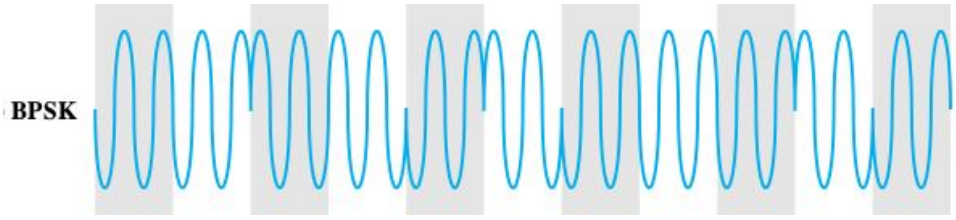
The fundamental concept of ASK is that the binary data can be represented in analog form by the changes in the amplitude of the carrier signal as shown in figure 3.6. Such techniques are used to transmit digital data over fibre cable etc.

3.3.5 Phase Shift Keying (PSK)

In this type of technique , the phase of the carrier is shifted to represent different data.

Two level PSK (BPSK)

To represent two binary digit , the simplest approach could be to use two different phases. For the same binary stream considered in ASK , the corresponding output of BPSK is shown in figure 3.7



$$\text{BPSK} \quad s(t) = \begin{cases} A\cos(2\pi f_c t) \\ A\cos(2\pi f_c t + \pi) \end{cases} = \begin{cases} A\cos(2\pi f_c t) & \text{binary 1} \\ -A\cos(2\pi f_c t) & \text{binary 0} \end{cases}$$

Figure 3.7 : BPSK waveform

It is therefore observed that whenever the value of the stream changes from 0 to 1 or 1 to 0, there is a shift in the phase of the carrier signal. However the amplitude remains the same. In Figure 3.7 it is observed that initially the signal starts from a negative phase for value 0 and the signal continues until the phase changes to positive after encountering 1.

3.3.6 Frequency Shift Keying (FSK)

In this technique the two values of binary are represented by two different frequency value.

$$\textbf{BFSK} \qquad s(t) = \begin{cases} A\cos(2\pi f_1 t) & \text{binary 1} \\ A\cos(2\pi f_2 t) & \text{binary 0} \end{cases}$$

It is therefore , observed that the signal corresponding to frequency f_1 represents the binary value 1 , while the signal corresponding to frequency f_2 represents the binary value 0. The Figure 3.8 shows the output of binary FSK(BFSK) corresponding to the binary stream considered in ASK.

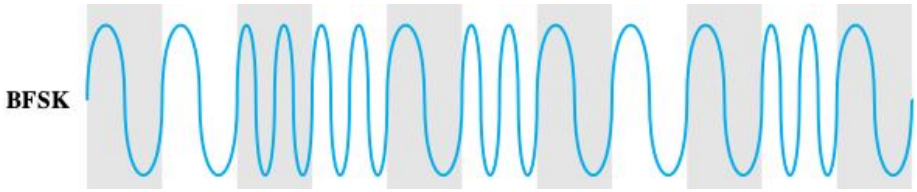


Figure 3.8 : BSFK signal

From the figure 3.8 it is clear that whenever the value is 0 ,the frequency of the signal is different from the value whenever it is 1.

3.4 MULTIPLEXING

Multiplexing is the technique of transmitting data from multiple sources using the same channel.

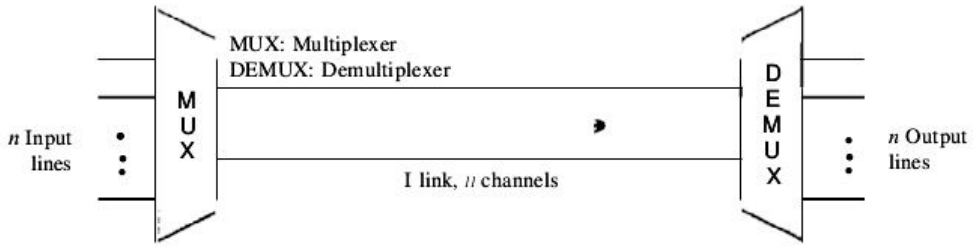


Figure 3.9 : Multiplexing

As shown in the figure 3.9, n-input lines are fed into the multiplexer , which has one outgoing link. The above multiplexer is Nx1 i.e N input line and 1 outgoing link. In each input line, message from a sender arrives which would than be multiplexed into the outgoing channel.

There are three basic multiplexing techniques shown in Figure 3.10

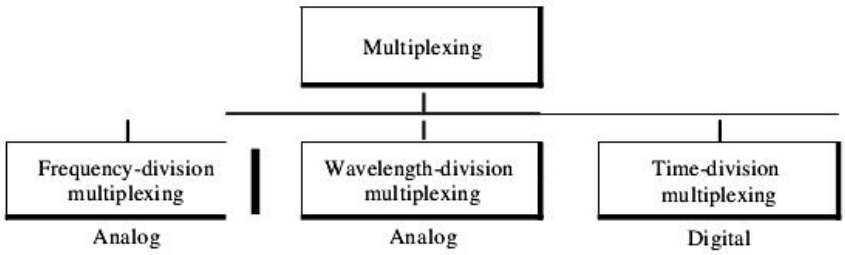


Figure 3.10 : Types of multiplexing

3.4.1 Frequency Division Multiplexing (FDM)

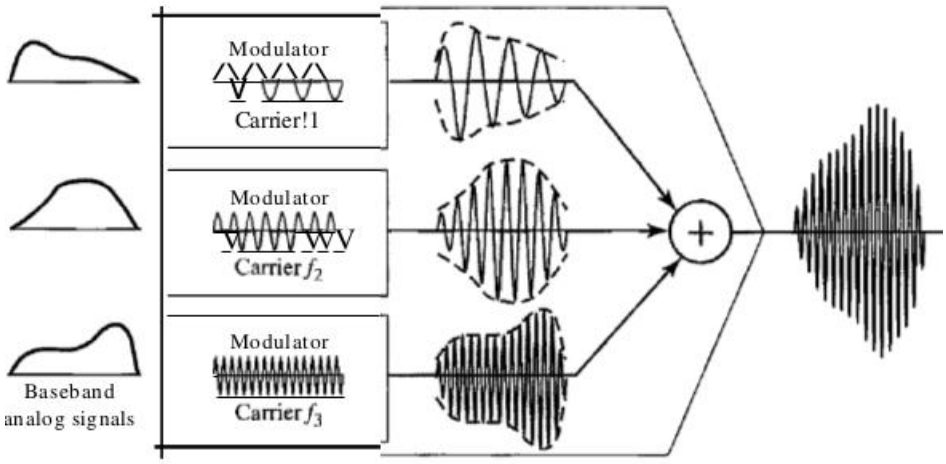


Figure 3.11:Frequency Division Multiplexing

In this process the multiplexed signal in the channel is decomposed into all its constituent signals. Each of the constituent signals is then passed to a demodulator to recover the original signal waves. In FMD, the given frequency spectrum is shared between prospective customers. Slots are pre-assigned to sources. Between the slots there are bandgaps to prevent spillovers of contents from nearby channels. The preassignment of the slots is done using carrier frequencies f_1, f_2 etc. Figure 3.11 shows a FDM scheme with different carrier frequencies. The recovery of the signals is done by the demultiplexer using demodulation. This is shown in figure 3.12

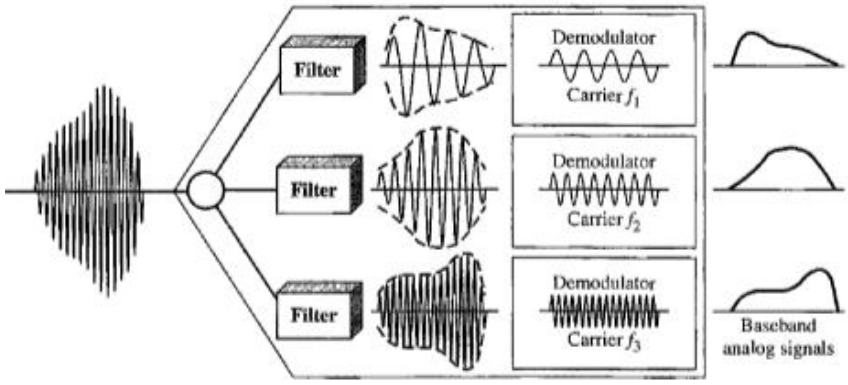


Figure 3.12: Frequency Demultiplexing

3.4.2 Wavelength Division Multiplexing (WDM)

WDM is used in optical fiber communication. In optical fiber, data is carried by light. Different wavelengths of light represent different sources of transmission. Therefore, when a spectrum of light represented by a wavelength is shared by several sources, the multiplexing scheme used is WDM. Figure 3.13 represents a WDM scheme of data transmission. Hence, analog signals are first converted into binary bits and then transmitted using WDM with varying wavelengths. In the receiver side, the individual light of various wavelengths are received from the multiplex signal.

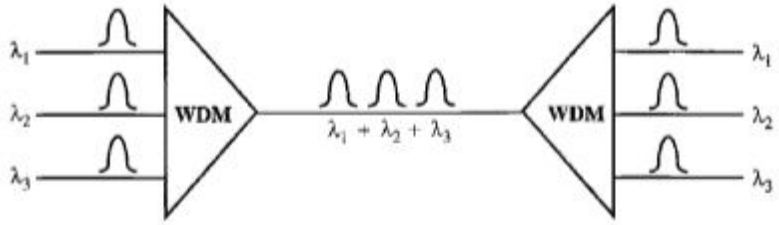


Figure 3.13: WDM for three channels

3.4.3 Time Division Multiplexing (TDM)

In FMD, the entire bandwidth is not available to a single signal at a time, because the available bandwidth is shared between all the signals with the same or different frequency width. However, in TDM the entire bandwidth is available to a signal in this turn of corresponding

slot. In TDM the sharing factor between the transmitters is time. TDM is the result of sampling theorem which requires that discrete segments of a signal are selected at a rate specified by at least two times of the maximum frequency content in the signal. It converts the analog signal into a discrete one. This results in slots between the signal samples where other signals may be accommodated. This is the basis of TDM. Gaps between discrete time slots of a signal may be used by another signal to carry data in this way during certain fixed time slot, several sources can exploit the gaps between inter-time slots and use the entire frequency spectrum. This way bandwidth efficiency increases.

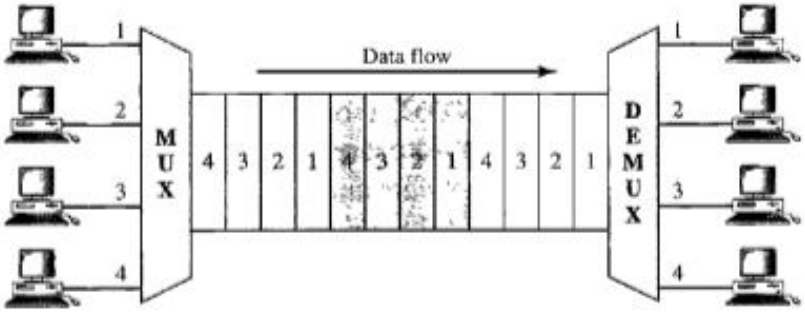


Figure 3.14: TDM for four channel

TDM is divided into synchronous TDM. In synchronous TDM, the output slot is reserved for a system even if it has no data for transmission. Thus it becomes an empty slot with no data.

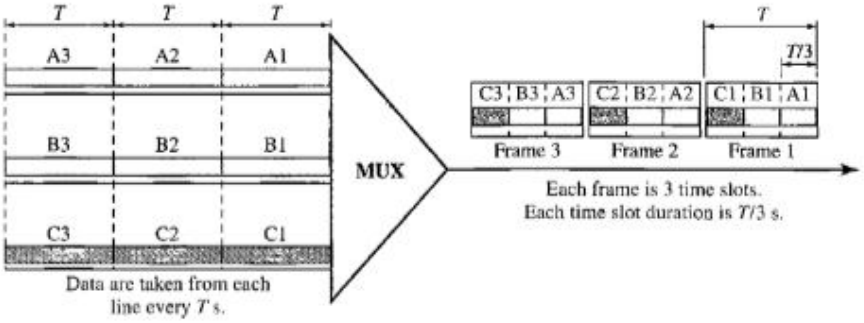


Figure 3.15 : Synchronous Time Division Multiplexing

In statistical time division multiplexing, any system whoever don't have any data to transmit, the corresponding slot is not generated in the multiplexed channel. The communication becomes faster and the bandwidth of the channel is better utilized. TDM is shown in figure 3.14 and synchronous TDM is shown in figure 3.15

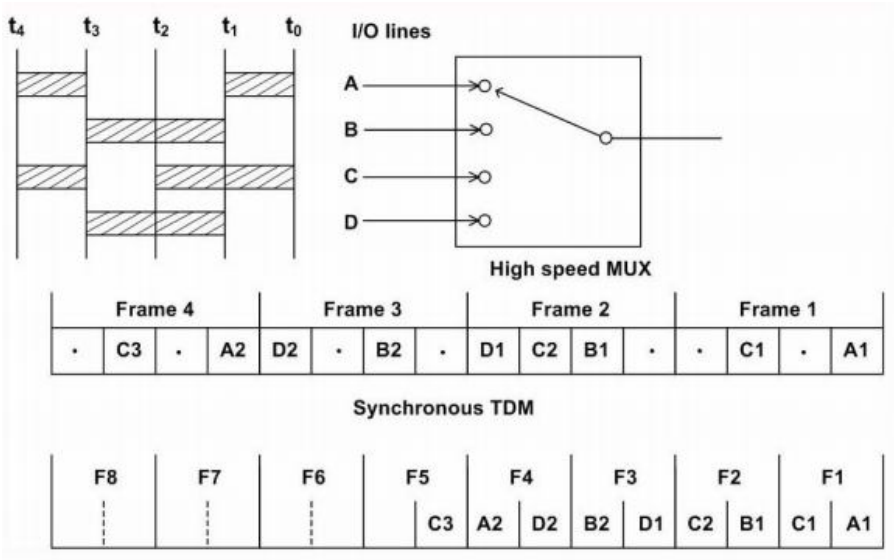


Figure 3.16: Synchronous vs Asynchronous

From the figure 3.16 it is clear that the multiplex channel does not have any empty slots. Slots are created only corresponding to the system that has data to transmit. This creates an overhead on the message. If the message does not carry the sender and receiver address, the demultiplexer would be in a problem to deliver a given message to the appropriate receiver. Thus, every message needs to carry source and destination address.



CHECK YOUR PROGRESS

1. Five channels each with 100Khz bandwidth, are to be multiplexed together, What is the minimum bandwidth of the link if there is a need for a guard band of 10Khz between the channels to prevent interference

We have an available bandwidth of 100Khz which spans from 200 to 300Khz. What should be the carrier frequency and the bit rate if we modulated our data by using FSK with d=1.

3.5 ERRORS

Error is the unwanted noise or additional signal added resulting in changes to the original signal during the communication process , most of the time

by the extrenal factors in the medium and in least cases in the end devices(sender and receiver). Errors can be categorized into different types

3.5.1 Single Bit Error

In this type , the information data is assumed to be of digital form and only one bit changes from 0 to 1 or from 1 to 0 during the communication.

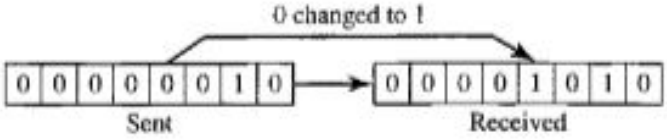


Figure 3.17: Single Bit error

In Figure 3.17 bit 0 to changes to bit 1 during the communication.

3.5.2 Burst Error

In this type of error atleast 2 bits or more have changed during the transmission process. This is shown in Figure 3.18.

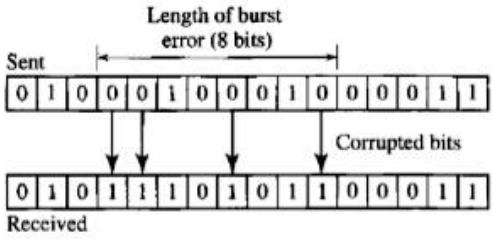


Figure 3.18: Burst error

Detection versus Correction: Detection is always easier than correction. In Detection it is simply important to know if the transmission have undergone any corruption, however in Correction, the total number of bits that have undergone corruption along with the exact location of the bit is also important. Thus correction is expensive in terms of computation power.

3.6 BLOCK CODING

Here a message ‘m’ is divided into blocks and each block is of length ‘k’. These blocks are called datawords. Redundancy is critical for data error detection and recovery. If ‘r’ number of redundant bits are added then the

total size of the bits becomes

$$n=k + r$$

This 'n' bit blocks after adding the redundant bits are called codewords.

Error Detection: For detecting any error by the receiver the following conditions need to be met

1. Valid list of codewords exist with the receiver.
2. The original codeword has changed to some invalid code.

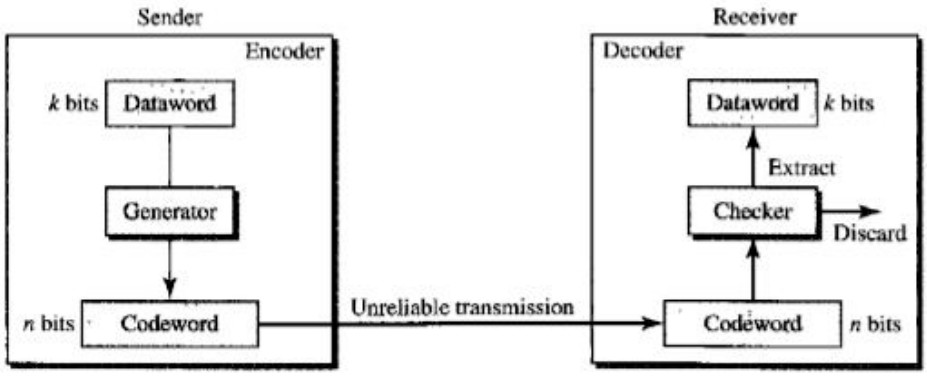


Figure 3.19:Process of Error detection in Block Coding

The generator transforms the k bit dataword into n bit codeword. However, in the receiving end if the codeword matches with existing codewords the data is accepted; otherwise it is discarded. The risk in this type of approach is that if the corrupt codeword matches with any codeword list in the receiver , the error go unnoticed and the message is accepted. If the error takes place only in 1 bit position, this type of approach proves successful, otherwise it fails. This is shown in figure 3.19

Error Correction: One primary requirement is that more redundant bits are needed in error correction than detection.

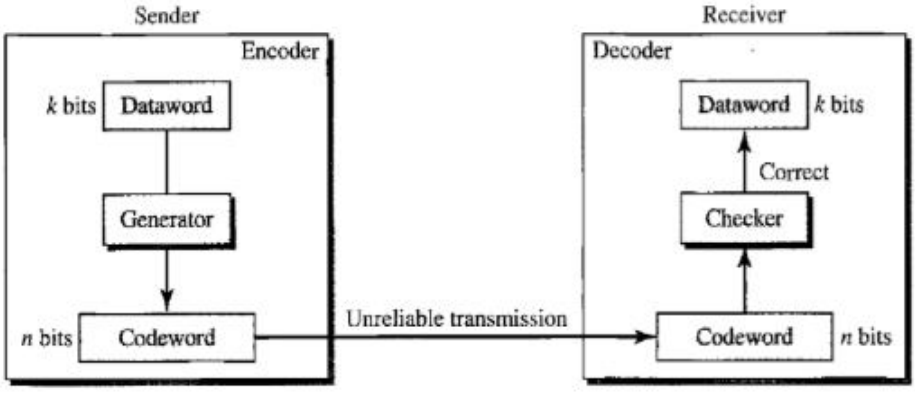


Figure 3.20: Encoder and Decoder in Error Correction

Taken as an example 3-bit redundant bit approach, the following gives an idea of error correction

Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

Figure 3.21: Dataword transformed into Codeword -3 bit redundant

Suppose the sender wants to transmit the dataword 01 whose codeword is 01011; but during the transmission the codeword changes to 01001. The receiver compares it with the first codeword 00000 and he knows the data is not 00 because there are changes in two or more bits(receiver assumes there will be changes only in one bit). The sender than compares with all others like, until the difference is found with 01011 by 1 bit and therefore comes to know that the data word is infact supposed to be 01. This is shown in figure 3.21.

3.7 HAMMING DISTANCE

The above discussion makes it clear that it is always important to control error during transmission process. One such approach is the Hamming Distance. The hamming distance between two codewords is the number of bits by which both the codewords differ from each other.

The easiest way to find hamming distance between two code words is to do an X-OR operation and find the number of 1s in the result.

Let us find the hamming distance between two pairs of words

1. The hamming distance d(000, 011) is 2 because 000 011 is 011 (two 1s)
2. The hamming distance d(10101, 11110) is 3 because 10101 Å 11110 is 01011 (three 1s)

Minimum Hamming Distance: Minimum Hamming Distance is the smallest hamming distance between all the codewords. In error correction or detection it is always this Minimum Hamming Distance that is considered.



CHECK YOUR PROGRESS

2. Find the minimum hamming distance for the following. We need a dataword of atleast 7 bits. Calculate values of k and n that satisfy the requirement.

Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

3.8 LINEAR BLOCK CODES

In simplest terminology, a linear block code is a codeword which when exclusive ORed with another block codeword creates a result with a valid codeword . A few linear block codes are discussed below

Simple parity-check Code

In this type, a k bit is transformed into a n bit with r redundant bits. If the value of r=1, then

n=k + 1. This one bit is the parity bit. This bit is used to check the total number of 1's i,e is even or odd and accordingly even parity and odd parity.

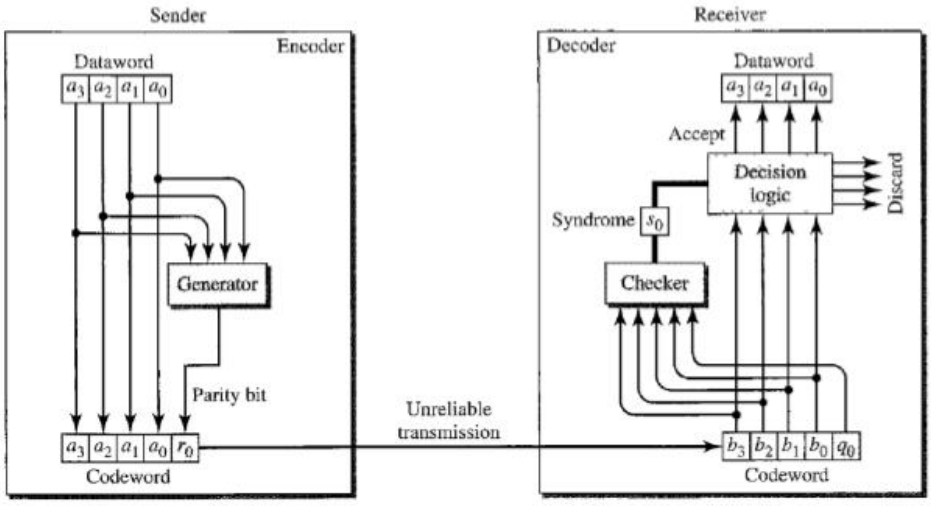


Figure 3.22: Encoder and Decoder for a single parity bit

As shown in Figure 3.22, the 4-bit dataword is appended by the parity bit transforming itself into a 5 bit codeword. The generator produces a parity bit with either value 0 or 1 to make even number of 1's. Even parity is most widely used. This type of approach is not used for error correction and used only for detection.

In the receiver the syndrome(s) is calculated as follows

$$s_0 = b_3 + b_2 + b_1 + b_0 + q_0 \quad \text{(modulo-2)}$$

The value of s becomes 0 when the number of 1's received is even otherwise 0.

Hamming Codes: Parity code is error detection code , but the Hamming Code is an error correcting codes. If the minimum hamming distance considered in the design is 3, then it can detect upto 2 errors and correct one error. Thus formulating

No of corrected errors=Minimum Hamming Distance – 2

An integer m such that m>=3 is chosen , then the relationship between n and k can be defined as follows

n=2m-1 and k=n-m

The number of check bits(redundant bits), r=m.

For example if m=3, then n=7 and k=4.

The Hamming code representation becomes C(7,4) with d_{min}=3 and the table formulation is shown in Figure 3.23

Datawords	*Codewords*	*Datawords*	*Codewords*
0000	0000000	1000	1000110
0001	0001101	1001	1001011
0010	0010111	1010	1010001
0011	0011010	1011	1011100
0100	0100011	1100	1100101
0101	0101110	1101	1101000
0110	0110100	1110	1110010
0111	0111001	1111	1111111

Figure 3.23: Hamming Code C(7,4)

The structure of the encoder and decoder for Hamming code is shown in Figure 3.24

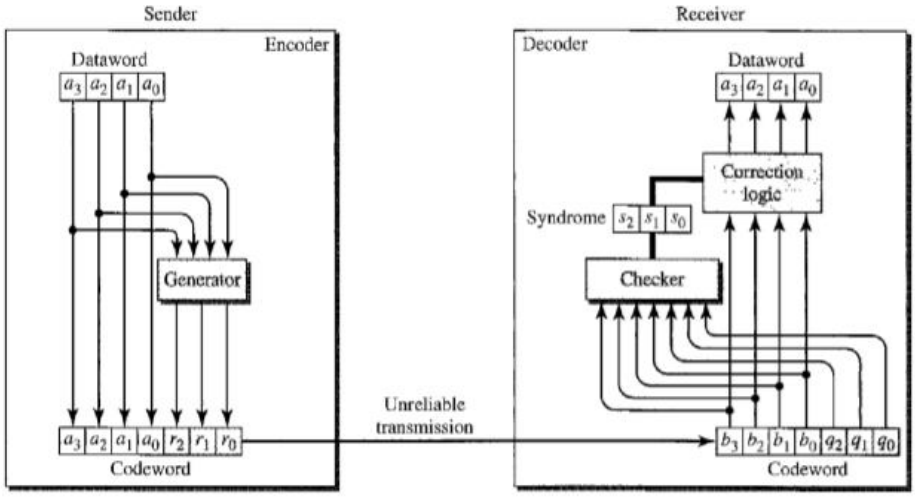


Figure 3.24 :Structure for the Hamming Code

As shown in figure 3.24 a 4-bit data word is transformed as follows:

$$r_0 = a_2 + a_1 + a_0 \quad \text{modulo-2}$$
$$r_1 = a_3 + a_2 + a_1 \quad \text{modulo-2}$$
$$r_2 = a_1 + a_0 + a_3 \quad \text{modulo-2}$$

Thus it is obvious that each of the parity bit is used corresponding to three bits out of a 4-bit data words.(i.e no of parity bit= no of bits in data word –

$$s_0 = b_2 + b_1 + b_0 + q_0 \quad \text{modulo-2}$$
$$s_1 = b_3 + b_2 + b_1 + q_1 \quad \text{modulo-2}$$
$$s_2 = b_1 + b_0 + b_3 + q_2 \quad \text{modulo-2}$$

Corresponding syndromes are computed as follows

The logical decision made by the logical analyzer on the receiver can be tabulated as below

Syndrome	000	001	010	011	100	101	110	111
Error	None	q_0	q_1	b_2	q_2	b_0	b_3	b_1

For example if there is an error only in q_0 , s_0 is only affected and hence the syndrome is 001.

Data Compression and Encryption

Transmission can be improved and bandwidth consumed by reducing the number of characters to be transmitted in a communication. This process is data compression. It removes redundancy in data. Figure 3.24 shows a data compression system.

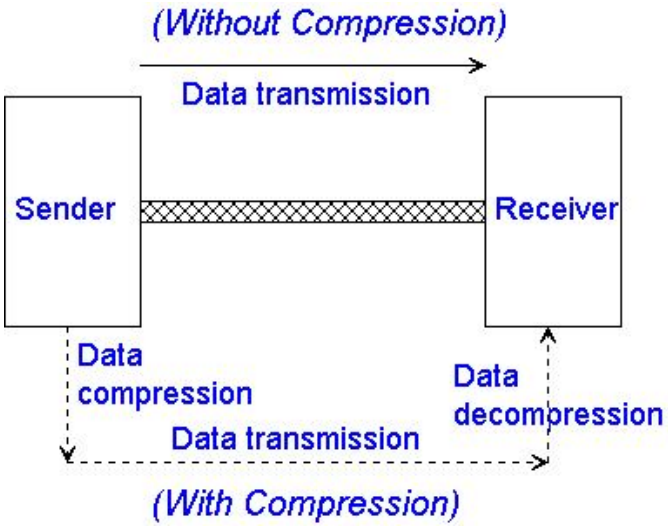


Figure 3.25: Data Compression System

Different data compression techniques exist that can be adopted as mentioned below:

a. Reversible: By the process of backtracking, the data can be recovered by the receiver after compression. (Huffman Code)

b. Nonreversible: Data after compression cannot be recovered. (Voice Compression/Video Compression)

In Huffman Code(not discussed here) the least occurring characters(low frequency) is assigned lesser bits than the total number of bits allocated for higher frequency characters. Data compression may be **lossy** or **lossless**. In lossy data compression some portion of original content is lost. It is useful for cases where is not very important like birth day party photo. But in lossless compression, the original content is nearly preserved. It is suitable in medical application.

Data Encryption

Here the data is transformed from the original format into a format unintelligible to any unauthorized person trying to tamper the data. Different approach of Data Encryption are as follows

a. Symmetric Key Approach

b. Asymmetric key Approach

a. Symmetric Key Approach: In symmetric key approach, both the sender and the receiver uses the same key for encryption and decryption respectively. Modern symmetric key cryptographic approach uses Block cipher techniques(Fiestel Block in DES). One of the popularly used symmetric key cryptographic technique is Advanced Encryption Standard (AES). This is shown in figure 3.26.

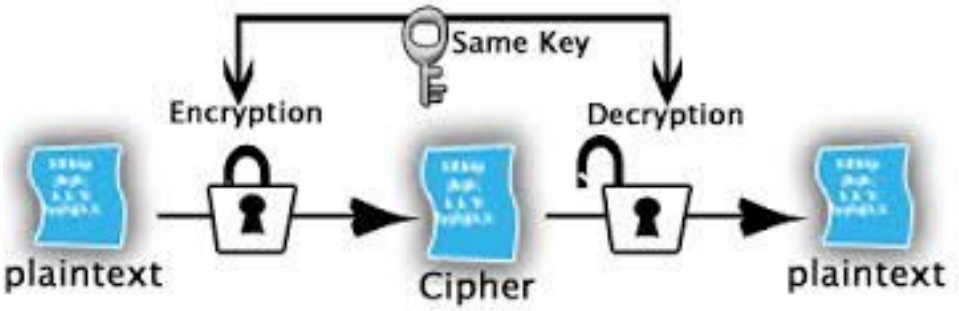


Figure 3.26: Block representation of symmetric key cryptography

b. Asymmetric key cryptography: In this type of approach the same key is not used by the sender and the receiver. The server(any point of interest) for example, maintains a pair of keys, namely, public key and private key. The public key is declared (available in Internet) and anyone can view this key. However, the private key is assigned to authorized system or person by Key Distribution Authority (KDA) and is strictly confidential. The steps of operation can be summarized as follows

a. A user 'X' who wants to transmit data to another user 'Y' , uses the public key of the 'Y' to encrypt the data .Once accomplished he transmits the data to 'Y'.

b. 'Y' on receiving the data decrypts the data by using his own private key.

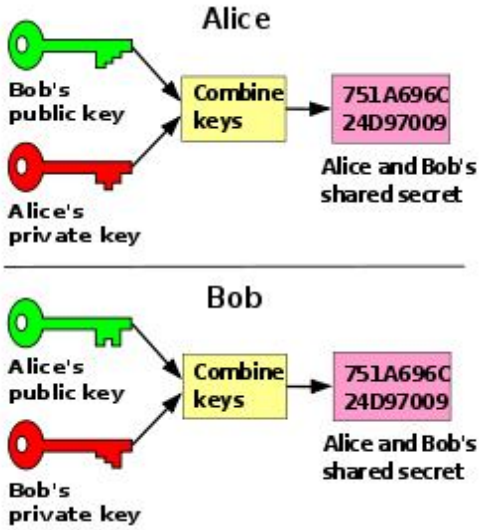


Figure 3.27: Asymmetric cryptography operation

As shown in the figure 3.27 when Alice wants to transmit data to Bob, Alice encrypts the data using the Bob public key. The encrypted data is again encrypted using own (Alice) private key. Thus two levels of encryption takes place here. While receiving the message, Bob initially decrypts the data using public key of Alice. The resultant message is again decrypted using own private key. Thus two level of decryption takes place here.



3.9 LET US SUM UP

- Modulation is a technique of transforming the original signal into a representation such that the transmission of the signal becomes possible in a given medium over a larger distance.
- In terms of analog signals, modulation may be classified amplitude modulation(AM), frequency modulation(FM) and phase modulation(PM).
- Multiplexing is the technique of transmitting data from multiple sources using the same channel.



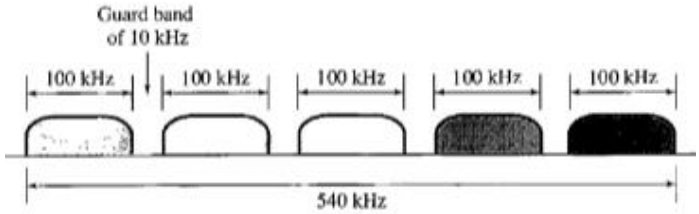
3.10 FURTHER READING

- Kennedy, G., & Davis, B. (1985). *Electronic communication systems*. (Vol. 20, p-21). Tata McGraw-Hill Publishing Co. Ltd., New Delhi.
- Forouzan, A.(2006). *Data communications & networking* (sie). Tata McGraw-Hill Education.
- Hamming, W. (1950) "*Error detecting and error correcting codes." Bell Labs Technical Journal,* 29.(2), 147-160.



3.11 Answers to Check your progress

Answer 1: For five channels we need atleast four guard bands . This means that the required bandwidht is at least 5 X100 + 4 X10=540Khz.



The modulation is done using FSK. The midpoint of the band is 250khz. We choose 2²⁰%f to be 50khz. Therefore, B=(1+d) * S + 2²⁰%f =100 2S=50khz , S=25kbaud.

Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

Answer 2:

We first find all the Hamming distances.

$d(00000, 01011) = 3$ $d(00000, 10101) = 3$ $d(00000, 11110) = 4$
$d(01011, 10101) = 4$ $d(01011, 11110) = 3$ $d(10101, 11110) = 3$

The d_{min} in this case is 3.

Note: To gurantee detection upto 's' errors , the minimum hamming distance in a block code must be $d_{min}=s + 1$.

We need to make k=n-m greater than or equal to 7.

i. If we set m=3, the result is $n=2^3-1$ and k=7-3 or 4 which is not acceptable

ii. If we set m=4 then $n=2^4-1$ =15 and k=15-4=11. Which satisfies the condition



3.12 MODEL QUESTIONS

Q1. How does a single bit error differ from burst error?

Q2. What is a Hamming Distance? What is Minimum Hamming Distance?

Q3. What is the Hamming distance for each of the following code
   i. d(10000,00000)
   ii. d(10101,10000)
   iii. d(11111,11111)

Q4. Assume that a voice channel occupies a bandwidth of 4khz. We need to multiplex 10 voice channels with guard bands of 500Hz using FDM. Calculate the required bandwidth.

Q5. We need to transmit 100 digitized voice channels using a pass-band channel of 20Khz. What should be the ratio of bits/Hz if we use no guard band?

Q6. Distinguish between a link and a channel in multiplexing.

UNIT4 : INTRODUCTION TO COMPUTER NETWORKS

UNIT STRUCTURE

4.1 Learning Objectives
4.2 Introduction
4.3 Computer Network
4.4 Goals of Computer Network
4.5 Switching Techniques
    4.5.1 Circuit Switching
    4.5.2 Message Switching
    4.5.2 Packet Switching
4.6 Connection-Oriented & Connection-Less Services
4.7 Broadcast & Point-to-Point Networks
4.8 Categories of Network
    4.8.1 Local Area Network
        4.8.1.1 LAN Transmission Methods
        4.8.1.2 Peer-to-Peer LAN & Server Based LAN
    4.8.2 Metropolitan Area Network
    4.8.3 Wide Area Network
4.9 Network Topology
    4.9.1 Bus Topology
    4.9.2 Ring Topology
    4.9.3 Star Topology
    4.9.4 Mesh Topology
    4.9.5 Tree Topology
4.10 Transmission Types
    4.10.1 Parallel Transmission
    4.10.2 Serial Transmission
        4.10.2.1 Asynchronous Transmission
        4.10.2.2 Synchronous Transmission
4.11 Protocols and Standards
4.12 Let Us Sum Up
4.13 Further Reading
4.14 Answers to Check Your Progress
4.15 Model Questions

4.1 LEARNING OBJECTIVES

After going through this unit, you will be able to :

- define computer network with their goals
- learn about different switching techniques
- describe connection-less and connection-oriented services
- learn about broadcast and point-to-point networks
- describe the types of computer networks
- describe server and peer-to-peer LANs
- learn about network topology
- describe synchronous and asynchronous types of communi-cations
- learn about different modes of communication
- learn about protocols and standards.

4.2 INTRODUCTION

In the previous units, we have learnt about the different types of data communication modes and different digital transmission techniques. In this unit we are going to discuss computer networks. We will begin the discussion with the definition of computer networks and its goal. While doing so, we will also introduce the various concepts associated with data communication. Different categories of network are also discussed here along with network topology. Towards the end of the unit we will discuss about the different protocols and standards associated with computer networks. In the next unit, we will explore different issues related to network models.

4.3 COMPUTER NETWORK

A computer network is a group of computers that shares information across wireless or wired technology. A computer network is a collection of several computers and terminals interconnected by one or more transmission paths. The main goal of a transmission system is the transfer and exchange of data between the computers and terminals.

The terms DTE and DCE are very common in the data-communication. DTE stands for ***data terminal equipment*** and DCE stands for ***data communication equipment***. DTE is used to describe the end user machine, which is usually a computer or terminal. The function of communication

network is to interconnect DTEs so that they can share resources, exchange data, provide backup for each other, and allow the users to perform their work from any location. The DCE is used to connect the DTEs into the communications line or channel. It also contains a portion of an application process but the primary function of the DCE remains to provide an interface to the DTE with the communication network.

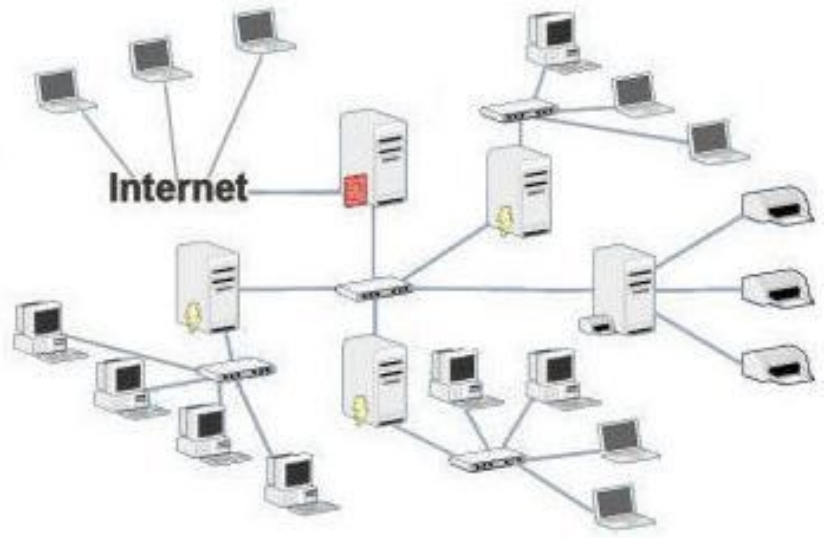


Figure 4.1 : A typical computer network

Before discussing the related issues in detail, it is worth devoting some time to go through the related concepts, importance and application of computer network.

4.4 GOALS OF COMPUTER NETWORKS

Before the advent of computer networks, papers and diskettes were the only means to share information. Now a days, computer networks which comprise a number of computers as well as other devices like printers, scanners etc. are widely used to share resources in a more efficient and faster way.

In a network, whether big or small, all the devices are interconnected to transmit and receive data from each other. These connections are usually made not only by copper wires; but also by the use of optical fiber cable, microwaves, infrared, and communication satellites. The main goals of these networks can be summarized as follows :

- **Resource Sharing :** By using computer networks, it is possible to share programs, data and other resources among several users on the

network. The sharing of resources is independent of the physical location of the user and the resource. Files on a particular user's computer can be shared on the network or files can be placed on a file server, which provides a central location for all files needed by the users on the network. Users can also share devices such as printers, CD-ROM drives, and hard drives etc.

- **High Reliablity :** A second goal is to provide high reliability by having alternative sources of supply. For example, all files could be replicatied on two or three machines; so, if one of them is unavailable, the other copies could be available.
- **Cost Reduction :** Resource sharing automatically reduces cost and hence money can be saved. For example, suppose, there are ten users and each requires a printer. If they could have been working indivually, then ten printers would have to be purchased. If these ten users are allowed to work in a network, then only two or three printers would be sufficient.
- **Communication Medium :** Computer networks provide a powerful communication medium among the users who are in a geographically same or different locations. A file that was updated or modified on a network, can be seen by the other users on the network immediately. Thus, it becomes easy for two or more people living far apart to work on a same project by dividing it by using a network. They can write programs, can discuss or can even change or modify some data using a network while they are far off. It enhances the speed of processing and operation.

4.5 SWITCHING TECHNIQUES

When we have multiple work stations, a problem of connecting them for one to one communication arises. One solution for this problem is to establish point to point connections between each individual pair of stations. To interconnect ***n*** stations, ***n(n-1)/2*** individual connections are needed. For example, to connect 10 components, a total of individual 45 connections are required; for 100 components, 4950 connections are required. The requirements increase drastically as the number of stations to be interconnected increase, and for a large number of stations, it is almost impossible to maintain individual connections. A well known solution for such type of problems is, ***switching***. Let us think, how things would be if we could only use our telephone to talk to just another person! So there are

requirements for switching systems to route our calls around the world. The switches can be placed on the transmission path so that the stations do not need to be interconnected directly. In this section, we will briefly discuss the various switching techniques which are in use currently.
In general, there are three methods of switching have been used: ***circuit switching***, ***message switching*** and ***packet switching***.

4.5.1 Circuit Switching

Circuit switching is the transmission technology that has been used since the first communication networks established during the last century. The basic idea of circuit switching is to set up a dedicated logical path between two users or machines, prior to the commencement of data communication. In circuit switching, a caller must first establish a connection to a callee before any communication. During the connection establishment, resources are allocated between the caller and the destination. The most common application of circuit switching is *telephone* network. Circuit switching mechanism is divided into three phases: circuit establishment, data transfer and circuit disconnect.

Characteristics of Circuit Switching

- An end to end dedicated path is needed.
- Connection path must be established before the beginning of data transmission.
- Channel capacity must be reserved between each pair of nodes whether that capacity is utilized or not.
- The technology is developed to handle voice data because of its key requirement that there should be no delay in the transmission and a constant signal transmission rate must be maintained.

Advantages of Circuit Switching

- It is well suited for real time communication purposes.
- The service provided by circuit switching are guaranteed since the channel remains dedicated to users throughout the communication session.

Disadvantages of Circuit Switching :

- Any unused ***bandwidth*** over the allocated circuit is wasted.
- Line may be idle most of the time.
- Both sender and the receiver must function with the same speed. This limits the utility of the network in interconnecting a variety of host computers and terminals.

4.5.2 Message Switching

Message switching refers to a switching technique involving transmission of messages from node to node through a network. In this technique, the source computer sends data or the message to the switching office first, which stores the data in its buffer. It then looks for a free link to another switching office and then sends the data to this office. This process is continued until the data are delivered to the destination computers. Owing to its working principle, it is also known as *store and forward*. That is, store first (in switching office), forward later, one jump at a time.

Advantages of Message Switching :

- No waiting of the setup of path. As soon as a user has data to send, he/she may transmit it over the channel.
- The channel can be fully utilized.
- Priority of messages can be implemented.
- Broadcasting can be done very easily to all the nodes in a network.

Disadvantages of Message Switching :

- Not suitable for real time transmission
- In message switching, message size is not fixed. For very large message, the nodes are required to have large buffers, which may not be practically implemented due to economical and technical constraints.
- Since there is no size limitation of the message, long messages can keep the channel blocked for a long period of time.

4.5.3 Packet Switching

Packet switching is very similar to message switching. The main difference is that the length of the units of data that may be presented to the network is limited in a packet-switches network.

With message switching, there is no limit on block size; in contrast, packet switching places a tight upper limit on block size. A fixed size of packet which can be transmitted across the network is specified. Another point of its difference from message switching is that data packets are stored on the disk in message switching whereas in packet switching, all the packets of fixed size are stored in main memory. This improves the performance as the access time (time taken to access a data packet) is reduced, and thus, the throughput (measure of performance) of the network is improved. There are two approaches to implement packet switching. These are : the datagram approach and the virtual circuit approach.

4.6 CONNECTION-ORIENTED & CONNECTION-LESS SERVICE

On the basis of *acknowledgement sent by the receiver*, there are two distinct techniques used in data communications to transfer data. These are: *connection-oriented* and *connection-less services*. Each service can be characterized by a quality of service. In general, a reliable service is implemented by having the receiver acknowledge the receipt of each message so that the sender is sure that it arrived.

- ***Connection-oriented service*** is modeled after the telephone system. For making a call to someone, first we have to pick up the phone and dial the number. After the connection is established, we talk, and then hang up.

  Connection-oriented service requires a connection to be established before any data can be sent. It set up virtual links between end systems through a network. After the connection is established, the data is transferred. As soon as the transmission is completed, the service user releases the connection. This procedure requires a specific acknowledge-ment for the information whether the connection is established or not. This method is often called a “reliable” network service. In some cases, when the connection is established, the sender,

receiver, and the subnet conduct a negotiation about parameters to be used such as maximum message size, quality of service required, and other issues. Typically, one side makes a proposal and the other side can accept it, reject it, or make a proposal as an alternative to the earlier proposal.

- The analogy of sending letters and postcards best explains the ***connection-less service.*** Each message (letter) carries a full destination address, and each one is routed through the system independent of all the others. Normally, when two messages are sent to the same destination, the first one sent will be first one to arrive. But it is also possible that the first one sent can be delayed so that the second one arrives first. In a connection-less service, there is no initial end-to-end setup for a session; each packet is independently routed to its destination. The sender simply starts sending packets (called datagrams) to the destination. This service does not have the reliability of the connection-oriented method, but it is useful for periodic burst transfers. Neither system must maintain state information for the systems that they send transmission to or receive transmission from.

4.7 BROADCAST & POINT-TO-POINT NETWORKS

On the basis of *transmission technology*, computer networks can be divided into two categories: *broadcast* and *point-to-point*.

- ***Broadcast networks*** have a single communication channel which is shared by all the machines on the network. Here, messages are broken into *packets* which are then broadcast to all machines in the channel. A packet sent by any machine is received by all the machines but only that machine processes the packet for which it is intended. An address field (destination address), within the packet specifies for whom it is intended. Upon receiving a packet, a machine checks the address field. If the packet is intended for itself, it processes the packet; if it is intended for some other machine, it is just ignored. However, in broadcast networks, we have the problem of deciding who uses the channel, if there is competition for it. For this, *medium access control* (MAC) is used to determine who goes next.

  As an analogy, let us imagine a scenario where a teacher taking the attendance in a particular class. He/she will call every roll number in the

classroom, all will listen but only the respective student will respond. Broadcast implies "*sending a signal where multiple parties may hear a single sender*".

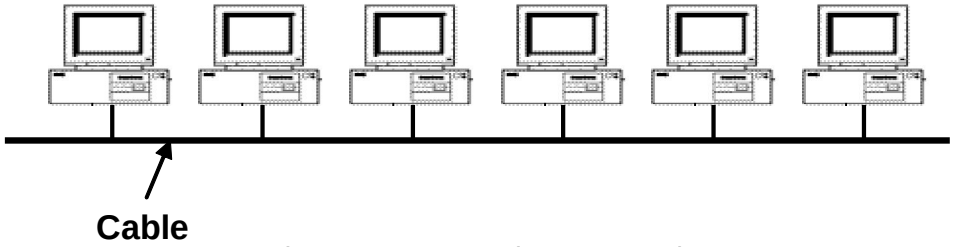


Figure.4.2 : A Broadcast Network

- Again, let us imagine a scenario that a teacher is taking the attendance in an examination hall. He/she will go to each member of the room and will take the attendance individually. Here, only one person listen and he only responds to the call. ***Point-to-point network*** is a method of communication where one "point" (person or device or entity) speaks to another entity. Point-to-point network consists of many connections between individual pairs of machines. Packet, on this type of network, may have to visit one or more intermediate machines to reach from the source to the destination. Often multiple routes of different lengths are possible in point-to-point networks. Therefore, routing algorithms are required to determine the best routes out of multiple available routes.

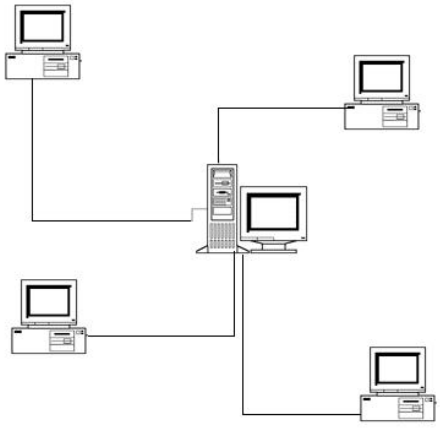


Figure 4.3 : Point-to-Point Network

4.8 CATEGORIES OF NETWORK

Whenever we have a set of computers or networking devices to be connected, we make the connections, depending on the physical layout and our requirements. On the basis of geographical area covered, computer networks can be classified under the following three categories:

- Local Area Network
- Metropolitan Area Network
- Wide Area Network

4.8.1 Local Area Network

A ***local area network*** or **LAN**, is a high-speed data network that covers a relatively small geographic area such as a building, a laboratory, or a school. It typically connects workstations, personal computers, printers, servers, and other devices. LANs offer computer users many advantages, including shared access to devices and applications, file exchange between connected users, and communication between users via electronic mail and other applications. LANs differ in the way the computers are connected (i.e., their topology), in how information moves around the network (i.e., their transmission technology) and their size.

Institute of Electrical and Electronic Engineers(IEEE) is a US publishing and standards organization responsible for many LAN standards such as the 802 series. *IEEE 802.3,* popular called *Ethernet* is the most popular LAN, usually operating at 10 Mbps to 10 Gbps and is present in most large organizations and offices. In a typical LAN configuration, one computer is designated as the *server*. It stores all of the software that controls the network, as well as the software that can be shared by the computers attached to the network. Computers connected to the server are called *workstations*. The various components and standards associated with a LAN will be discussed in *Unit 5* of this course. Two most commonly used LAN implementations are depicted in Figure. 4.4 (a) and (b).

LAN protocols function at the lowest two layers, i.e., the *physical* and the *data link layer* of the OSI reference model. We will discuss the layer concept of ISO-OSI reference model in the next unit which is “Network models”.

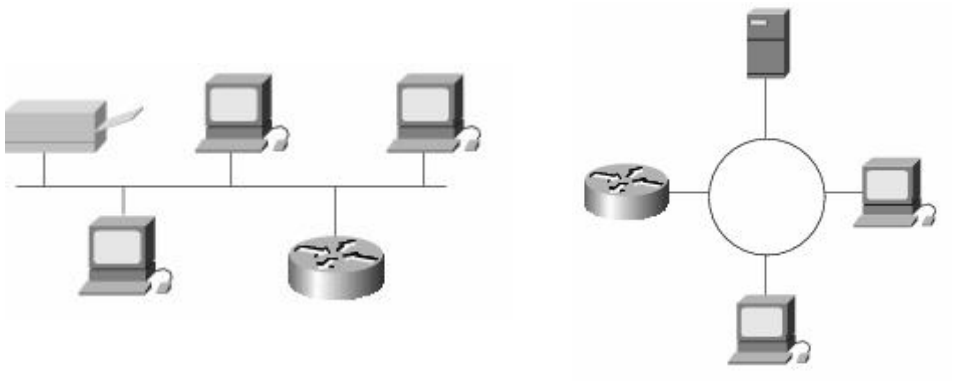


Figure 4.4 : (a) Ethernet or IEEE 802.3 (b) Token Ring or IEEE 802.5

4.8.1.1 LAN Transmission Methods

LAN data transmissions fall into three classifications: *unicast*, *multicast*, and *broadcast*. In each type of transmission, a single packet is sent to one or more nodes.

- ***Unicast :*** Unicast is a one-to-one transmission method in which the network carries a message to one receiver, such as from a server to a LAN workstation. In a unicast environment, even though multiple users might ask for the same information from the same server at the same time, such as a video clip; duplicate data streams are sent. Unicast sends separate data streams to each computer requesting the data, in turn flooding the network with traffic.
- ***Multicast :*** Multicast is a one-to-many transmission method in which the network carries a message to multiple receivers at the same time. Multicast is similar to broadcasting, except that multicasting means sending to a specific group, whereas broadcasting implies sending to everybody, whether they want the traffic or not. When sending large amounts of data, multicast saves considerable network bandwidth because the bulk of the data is sent only once. The data travels from its source through major backbones and is then multiplied, or distributed out, at switching points closer to the end users. This is more efficient than a unicast system, in which the data is copied and forwarded to each recipient.
- ***Broadcast :*** The concept of broadcast is already discussed in previous section. *Broadcast* is a one-to-all transmission method in which the network carries a packet to all devices at the same time, but a particular machice for which the packet is intended accepts it.

4.8.1.2 Peer-to-Peer LAN and Server Based LAN

On a LAN, we expect to share files, programs, or printers, all without being particularly aware of where the physical resources we're using are actually located. LANs providing these types of services are typically set up either as *peer-to-peer* or *client-server* LANs, or perhaps as a combination of the two.

- **Peer-to-Peer LAN :** All the machines on a peer-to-peer LAN are equal. Provided that the file's owners give permission, a file on machine A can be accessed from machine B, and vice versa. Peer-to-peer LANs do not require any one machine to be a dedicated, high-performance server; service by a peer-to-peer LAN is often cheaper for this reason. Peer-to-peer LANs work well when only a small number of machines are connected to it. But as the size of the LAN grows, peer-to-peer services can become quite disorganized. To serve all its peers, each machine on the LAN must be powerful enough and for this reason cost increases. For larger LANs, the dedicated client-server LAN architecture becomes more cost effective.

  Advantages

  Less initial expense – No need for a dedicated server.

  Setup – An existing operating system (such as Windows XP) of the machine may only need to be reconfigured for peer-to-peer operations.

  Disadvantages

  Decentralized – No central repository for files and applications.

  Security – Does not provide the security available on a client/server.

- **Client-Server LAN :** A client-server LAN consists of one or more server machines on which shared files and programs reside and many client machines where people do their task. The LAN server machines are usually have higher configuration and are fast because they must serve many users, while the client machines need only to be fast enough for one person to use at a time. Shared printers are either attached directly to a server, or to a print server (a specialized computer attached to the network), or to a personal computer on network that acts as a print server.

  Advantages

Centralized – In case of client-server architecture, resources and data security are controlled through the server.

Scalability – Any or all elements can be replaced individually as needs increase.

Flexibility – New technology can be easily integrated into system.

Accessibility – Server can be accessed remotely and across multiple platforms.

Disadvantages

Expense – Requires initial investment in dedicated server.

Maintenance – Large networks will require a staff to ensure efficient operation.

Dependence – When server goes down, operations will cease across the network.

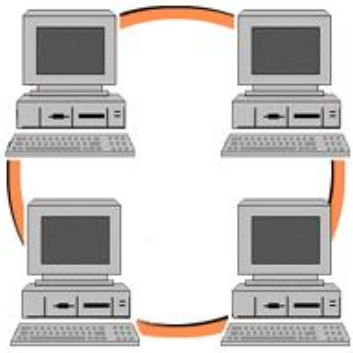


Figure 4.5 : Peer-to-Peer

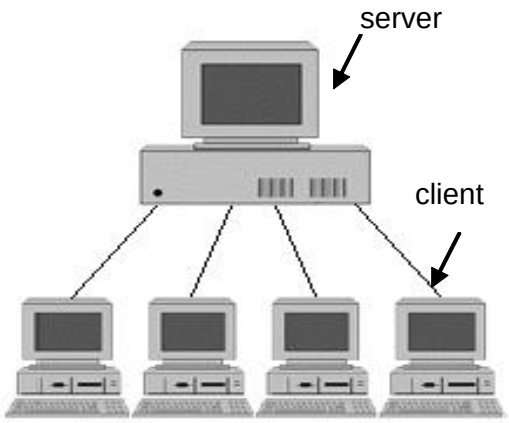


Figure 4.6 : Client-Server

4.8.2 Metropolitan Area Network

A ***metropolitan area network***, or **MAN**, is basically a bigger version of a LAN it and normally uses similar technology. It might cover a group of nearby corporate offices or a city and might be either private or public.

A MAN often provides efficient connections as it has high-speed transmission capabilities which uses some type of telecomminication components to handle long-distance transmission. One very common example of MAN is the cable television network. Another important example is the high-speed wireless Internet access, which has been standardized as *IEEE 802.16.*

4.8.3 Wide Area Network

A ***wide area network*** or **WAN** is a computer network covering multiple distance areas, which may spread across the entire world. WANs often connect multiple smaller networks, such as local area networks (LANs) or metropolitan area networks (MANs).

Typically, a WAN consists of a number of interconnected switching nodes. A transmission from any one device is routed through these internal nodes to the specified destination device. These nodes are not concerned with the content of the data; rather, their purpose is to provide a switching facility that will move the data from node to node until they reach their destination. Traditionally, WANs have been implemented using either *circuit switching* or *packet switching* technologies. More recently, frame relay and *ATM* (*Asynchronous Transfer Mode*) networks have assumed major roles. Frame relay provides higher data rates, lower costs, efficient handling of bursty data transmission in less expenditure. ATM offer more bandwidth to end users at less cost.

The ***Internet*** is the best known example of a WAN. Some segments of the Internet are also WANs in themselves. The Internet is a system of linked networks that are worldwide in scope and facilitate data communication services such as *remote login*, *file transfer*, *e-mail*, the *World Wide Web* etc. With the rise in demand for connectivity, the Internet has become a communications highway for millions of users. The Internet was initially restricted to military and academic institutions, but now it is a full-fledged conduit for any and all forms of information and commerce. Internet websites now provide personal, educational, political and economic resources to every corner of the planet. The last unit (Unit 6) of this course will give us the concept of Internet and various servicices provided by Internet.



CHECK YOUR PROGRESS

Q1. State True or False :

i) Broadcast networks share a single communication channel.

ii) For larger LANs, the client-server LAN architecture is less cost effective as compared to peer-to-peer LANs.

iii) The world's most popular WAN is the Internet.

iv) Cable televishion network is an example of MAN.

4.9 NETWORK TOPOLOGY

In this section we will discuss how computers and others devices are connected in a network. While discussing, we may come across the name of various internetworking devices like hub, switch, router etc.

In computer networking, ***topology*** refers to the layout of connected devices. Network topologies can be *physical* or *logical*. ***Physical topology*** means the physical design of a network including the devices, location and cable installation. It defines how the systems are physically connected. Several physical topologies are in use for networks today. Some of the common ones include the *bus*, *ring*, *star*, *tree* and *mesh*. More complex networks can be built as *hybrids* of two or more of the above basic topologies. The ***logical topology*** defines how the systems communicate across the physical topologies. The two most common types of logical topologies are ***broadcast*** and ***token passing***.

When we decide which topology we should choose for our network, then there are a few basic points that we need to take care of. The factors that decide which topology we should choose are:

- **Cost :** Cost is a factor that plays an important role for the decision of topology. If we want to create a network for 4-5 computer, we should not spend very much in network.
- **Scalability :** The size of the network that we need to make and the kind of topology
- **Bandwidth capacity :** The required speed of the network that can be

taken care of any particular topology.

- **Ease of installation :** Is it easy to install the network using selected topology.
- **Ease of fault finding and maintenance :** If we have a network we will definitely get the problem also. So will it be easy for network administrator to identify the problem and give the solution with ease and least possible time.

4.9.1 Bus Topology

A network of bus topology consists of a long single cable to which all the computers and other devices are connected. Any node attached to the bus can send signals down the cable to all the nodes of the network; that means a bus is a broadcast medium. When more than one node starts sending data through the bus, they mix with each other and the sent data become a garbage. This is called *collision.* To avoid collision there must be some agreement between the nodes so that when one computer starts to send data, others refrain themselves from sending data. To ensure correct data communication, both ends of the cable are terminated by a special device called end terminator.

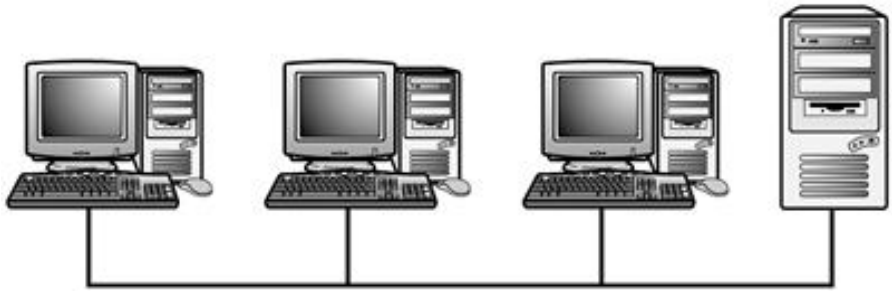


Figure 4.7 : A Bus Topology

Ethernet bus topologies are relatively easy to install and don't require much cabling compared to the alternatives. 10Base-2 (Thin) and 10Base-5 (Thick) both were popular Ethernet cabling options for bus topologies. Some advantages and disadvantages of the bus topology are listed below :

Advantages

- Bus topology is inexpensive in installation
- It requires less cable than other topologies
- Good for smaller networks not requiring higher speed.

- Easy to add systems to network.

Disadvantage :

- Out-of-date technology. Bus topology was used in the early days of networking because it was inexpensive to use and relatively easy to set up.
- If the backbone cable fails, the entire network effectively becomes unusable.
- Unmanageable in a large network. If more than a few dozen computers are added to a network bus, performance problems will likely result.
- Difficult to troubleshoot.

4.9.2 Ring Topology

In Ring topology the computers are connected between each other in such a way as to form a close loop. In practice, a cable connects the first computer to the second computer, another cable connects the second computer to the third and so on until the last computer is connected back to the first to complete the loop. It should be noted that the topology may not physically look like a circle. The ring means the computers are connected with each other in a logical ring. The interconnecting cables may take any shape in practice. All messages travel through a ring in the same direction (either “clockwise” or “counterclockwise”).

In ring topology the communicating computers must follow some agreement between them to avoid collision as it is also a broadcast network. The main difference between the bus and the ring topology is that the ring topology does not require termination. Because the systems are connected all together in a loop, there is no beginning and end point as there is with the bus topology. A failure in any cable or device breaks the loop and can take down the entire network.

Two major network types using the ring topology are :
i)*Fiber Distributed Data Interface* (FDDI) where a large, high speed networks use fiber optic cables in a physical ring topology.
ii)*Token-Ring networks* that use logical ring topology.

Advantages

- No collision of data as data travel in one direction only.
- Easier to fault find. It any point gets broken we can trace that easily.
- No terminator required.

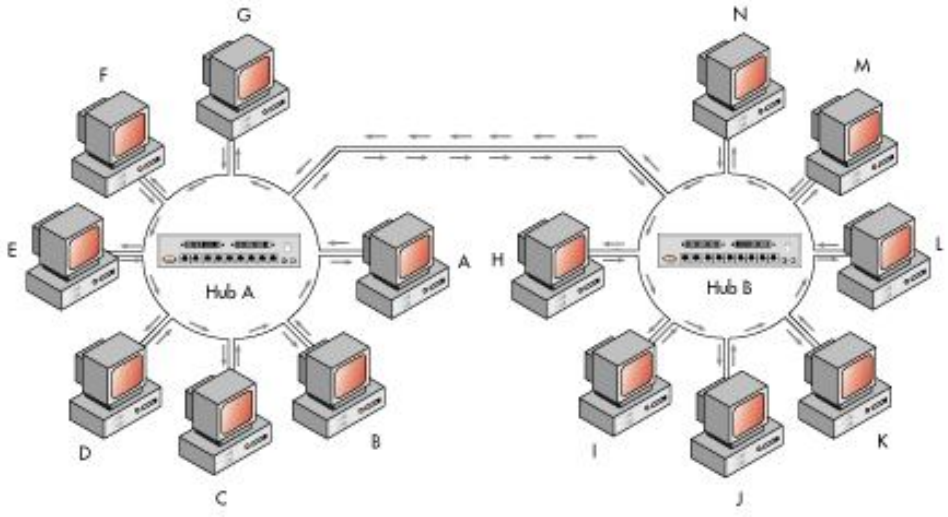


Figure 4.8 : Dual Ring topology

Disadvantages

- Ring topology requires more cable than bus topology
- A break in ring will take the whole network down.
- Addition or removal of any node can affect entire network.

4.9.3 Star Topology

In the *star* topology, all computers and other network devices connect to a central device(controller) called a ***hub*** as depicted in the figure.4.9. Each connected device requires a single cable to be connected to the hub. A hub normally accepts data from a sending computer and delivers it to the computer for which the data is addressed. Hence, a star network is not a broadcast network; rather it is a point-to-point network.

Using a separate cable to connect to the hub allows the network to be expanded without disruption to the network. Because each computer uses a separate cable to connect to the hub, the failure of a network connection affects only the single machine concerned. The other computers can continue to function normally.

Fast Ethernet (100Base-TX & 100Base-FX) in a star topology is the most commonly used LAN today. *10Base-T Ethernet* and *1000Base-TX Gigabit Ethernet* also use star topology.

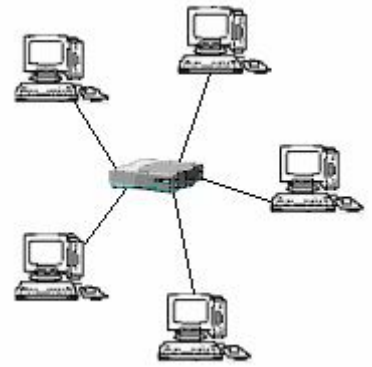


Figure 4.9 : Star Topology

Adavntages

- Star networks are easily expanded without disruption to the network.
- Easy to add/remove devices to/from network
- One break does not bring the whole network down. Cable failure affects only a single user.
- Easy to troubleshoot and isolate problems.
- Widely used centralized management.

Disadvantages

- Costs are usually higher than with bus or ring networks.
- Requires more cable than most of the other topologies.
- If the hub fails, any device connected to it will not be able to access the network.

In Table 4.1, various networks with their cable types, topologies are shown:

Table 4.1 : Network cable types and topologies

Network Type	Standard	Cable Type	Topology
Ethernet	10Base-2	Thin(RG-58)	Bus
Ethernet	10Base-5	Thick(RG-59)coaxial	Bus
Ethernet	10Base-T	CAT3 or CAT5 UTP	Star
Fast Ethernet	100Base-TX	CAT5 UTP	Star
Gibabit Ethernet	1000BaseTX	CAT5, 5e or UTP	Star
Token Ring	all	UTP or STP	Logical Ring

4.9.4 Mesh Topology

The mesh topology incorporates a unique network design in which each computer on the network connects to every other, creating a point-to-point connection between every device on the network. A mesh topology is used when there can be absolutely no break in communications, for example the control systems of a nuclear power plant. The purpose of the mesh topology is to provide a high level of redundancy. If a network cable, computers or other components fail, the data always has an alternative path to get to its destination.

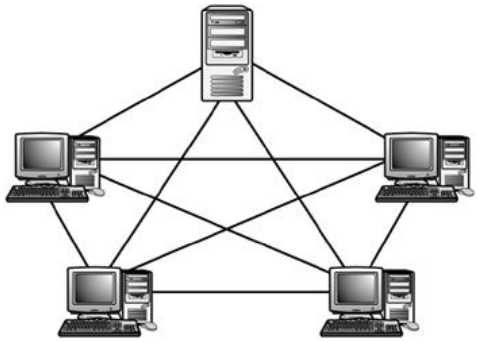


Figure 4.10 : A Mesh Topology

A fully connected mesh network has n(n-1)/2 cables to link 'n' devices. Therefore, every device on the network must have 'n-1' input/output (I/O) ports. For example, in the figure(Fig.1.10), we have five systems that require 10 cables to create a mesh network. This topology is mainly used in environments where high availability outweighs the costs associated with this amount of interconnection. We can see in the figure that the wiring for a mesh network can be very complicated. Further, troubleshooting a failed cable can be tricky. Because of this, the mesh topology is rarely used.

Advantages

- Provides alternative paths between devices in the network.
- The network can be expanded without disruption to the current users.

Disadvantages

- It is expensive as because a large number of cablings are required.
- Routing network traffic can be difficult because of all the

different possible paths between nodes.

- It is very expensive to wire up.

There are also partial-mesh networks where some of the nodes are connected to all the others, but others are only connected to nodes with which they exchange the most data.

4.9.5 Tree Topology

A tree topology combines the characteristics of bus and star topologies. It consists of groups of star-configured workstations connected to a linear bus backbone cable (Figure. 4.11). This bus/ star hybrid approach supports the future expandability of the network much better than a bus (limited in the number of devices due to the broadcast traffic it generates) or a star (limited by the number of hub connection points) alone.

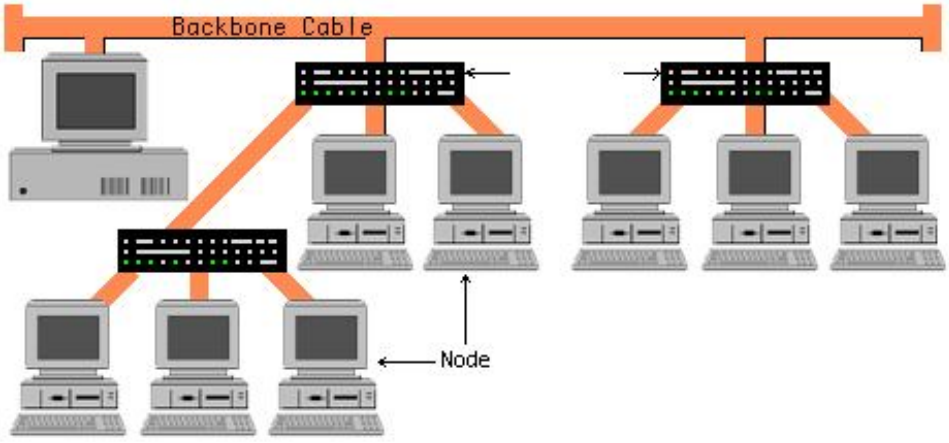


Figure 4.11 : Tree Topology

Advantages of Tree Topology

- The network is easy to extend by just adding another branch.
- Fault isolation is relatively easy.

Disadvantages of Tree Topology

- If the backbone cable breaks, the entire network goes down.
- More difficult to configure and wire than other topologies.
- If any hub goes down, all branches of that hub go down.



CHECK YOUR PROGRESS

Q.2. Fill in the blanks:

i)_____ defines the physical or logical arrangement of links in a network.

ii)In _____ topology, every devices has a dedicated point-to-point link to every other device.

iii)_____ topology is the combination of different types of topologies.

iv)In _____ topology, each device has a dedicated point-to-point link only to a central controller.

v)_____ topology provides alternative paths between devices in the network.

vi)Ring topology requires _____ cables than bus topology.

vii)The topology used in Ethernet is _____ topology.

viii)_____ topology uses central controller or hub.

Q.3.What topology is used by 10Base-5 and 10Base-2 Ethernet?

.....

Q.4.What topology is used by FDDI ?

.....

4.10 TRANSMISSION TYPES

Digital transmission is the sending of information over a physical communications media in the form of digital signals. Analog signals must therefore be digitised first before being transmitted. However, as digital information cannot be sent directly in the form of 0s and 1s, it must be encoded in the form of a signal with two states, for example :

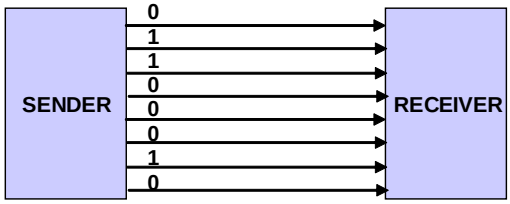
- two voltage levels with respect to earth,
- the difference in voltage between two wires ,
- the presence/absence of current in a wire and
- the presence/absence of light.

Digital data transmission can occur in two basic modes:***parallel*** and ***serial.*** When data is transmitted simultaneously in certain fixed sizes, it is parallel mode of transmission. But when data is sent one after another in a sequence, it is serial mode of transmission. Data within a computer system is

transmitted via parallel mode on buses with the width of the parallel bus matched to the word size of the computer system. Data between computer systems is usually transmitted in bit serial mode. Consequently, it is necessary to make a parallel-to-serial conversion at a computer interface when sending data from a computer system into a network and a serial-to-parallel conversion at a computer interface when receiving information from a network. The type of transmission mode used may also depend upon distance and required data rate.

4.10.1 Parallel Transmission

Parallel transmission communicates bits simultaneously over multiple lines(wires, frequency channels); typically the total consists of one or more bytes at a time. Parallel devices have a wider data bus than serial devices and can therefore transfer data in words of one or more bytes at a time. As a result, there is a speedup in parallel transmission bit rate over serial transmission bit rate. The timing for parallel transmission is provided by a constant clocking signal sent over a separate wire within the parallel cable; thus parallel transmission is considered synchronous. Computers are typically connected to printers and external disk drives via parallel interfaces, ports, and buses.



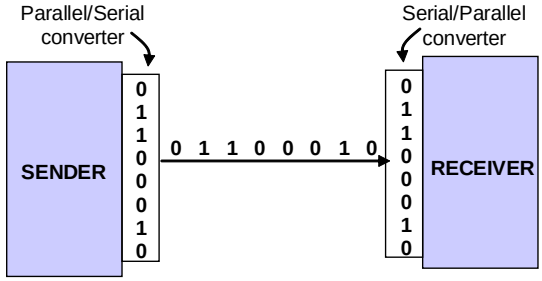
(a) Parallel Transmission;
The eight bits are sent through eight wires at a time

Figure 4.12 : Illustrates a parallel mode of transmission

4.10.2 Serial Transmission

Serial transmission sends one bit at a time over a single transmission line. The cost of communication hardware is considerably reduced since only a single wire or channel is required for the serial bit transmission which also slows the speed of transmission. Telephone lines use serial transmission for digital data, thus modems are connected to the computer via a serial port. A ***serial port*** is a socket on a computer used to connect the serial interface to a serial line or

bus. A ***serial interface*** is a data channel that transfers digital data serially; it is typically implemented as a card that plug into an expansion slot on a computer motherboard. Serial interfaces have multiple lines, but only one is used for data. An external ***serial bus*** carries serial data to any device connected to it, e.g. Ethernet. Serial transmission can be either *synchronous* or *asynchronous*.



(b) Serial Transmission;
The eight bits are sent one after another through a single wire

Figure 4.13 : Serial communication

Serial transmission technology is increasingly used for the transmission of digital data. A large number of up-to-date communications networks apply serial transmission. The numerous applications include computer networks for office communications, building and manufacturing automation, and finally, Internet. The transmission of a stream of bits from one device to another accross a transmission media involves a great deal of cooperation and agreement between the two sides (sender and receiver). One of the most fundamental requirements is ***synchronization***. The receiver must know the rate at which bits are being received so that it can sample the medium at appropriate intervals to determine the value of each received bit. For achieving the desired synchronization, there are two approaches. ***Synchronous*** and ***asynchronous transmissions*** are two different methods of transmission synchronization.

4.10.2.1 Asynchronous Transmission

Communication is called *asynchronous* if the transmitter and receiver do not need to synchronise before each transmission. A sender can wait arbitrarily long between transmissions and the receiver must be ready to receive data when it arrives. Most PC serial devices such as mouse, keyboards and modems are asynchronous.

As the name implies, asynchronous communication is performed between two or more devices which operate on independent clocks. Thus, there is no guarantee that when Point A begins transmitting, Point B will begin receiving, or that Point B will continue to sample at the rate Point A transmits. In asynchronous transmission, data are transmitted one character at a time, where each character is five to eight bits in length. Timing or synchronization must only be maintained within each character; the receiver has the opportunity to resynchronize at the beginning of each new character. Figure 4.14 depicts what happens when transmission clocks differ significantly. In fig.4.14 (a), we see that, the receiver samples at mid point of each bit of the incoming data. In fig.4.14 (b), the receiver clock is too slow; causing bit 4 to be skipped and as a result, the data is corrupted.To combat this type of timing problem, asynchronous communication requires additional bits to be added around actual data in order to maintain signal integrity. The bits of the character are transmitted beginning with the least significant bit.

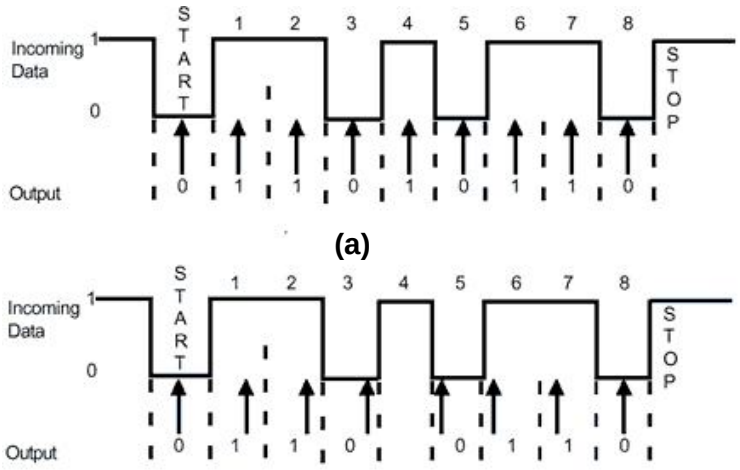


Fig. 4.14 : Asynchronous Data Sampling

Asynchronously transmitted data is preceded with a ***start bit*** which indicates to the receiver that a character is about to begin. The end of a character is followed by a ***stop bit***, which tells the receiver that the character has come to an end, that it should begin looking for the next start bit, and that any bits it receives before getting the start bit should be ignored. To avoid confusion with other bits, the start bit is twice the size of any other bit in the transmission. To ensure data

integrity, a ***parity bit*** is often added between the last bit of data and the stop bit. The parity bit is set by the transmitter such that the total number of ones in the character, including the parity bit, is even (even parity) or odd (odd parity), depending on the convention being used. The receiver uses this bit for error detection. The parity bit makes sure that the data to be received is composed of the same number of bits in the same order in which they were sent.

4.10.2.2 Synchronous Transmission

The term 'synchronous' is used to describe a data transfer method in which a continuous stream of data signals is accompanied by timing signals (generated by an electronic clock) to ensure that the transmitter and the receiver are synchronized with one another. These types of connections are used when large amounts of data must be transferred very quickly from one location to another. The speed of the synchronous connection is attained by transferring data in large blocks instead of individual characters. A block of bits is transmitted in a steady stream without start and stop bits. Data or information is moved from one place to another at instants in time that are measured against the clock signal being used. This signal is usually comprises of one or more high frequency rectangular shaped waveforms, generated by special purpose clock circuitry. These pulsed waveforms are connected to all the devices that operate synchronously, allowing them to start and stop operations with respect to the clock waveform.

Typical examples of synchronous signals include the transfer and retrieval of address information within a computer via the use of an *address bus*. For example, when a processor places an address on the address bus, it will hold it there for a specific period of time. Within this interval, a particular device inside the computer will identify itself as the one being addressed and wide acknowledge the commencement of an operation related to that address.

In synchronous transmission, each block begins with a preamble bit pattern and generally ends with a postamble bit pattern. In addition, there are some other bits that convey control information. The data

with the preamble, postamble, and control information are termed as a ***frame***. A general frame format for synchronous transmission is shown in Figure 4.15. Typically, the frame starts with a preamble called a flag, which is 8 bits long.

8-bit flag	Control fields	Data fields	Control fields	8-bit flag

Figure 4.15 : Synchronous Frame Format

The same flag is used as a postamble. The receiver looks for the occurance of the flag pattern to signal the start of a frame. This is followed by some number of control fields, then a data field, more control fields, and finally the flag is repeated.

4.11 PROTOCOLS AND STANDARDS

Many different types of network protocols and standards are required to ensure that a computer can communicate with another computer located on the next desk or any location around the world.

A network protocol is a set of rules that governs the communications between computers and other devices on a network. These rules include guidelines that regulate the characteristics of a network such as access method, allowed physical topologies, types of cabling, and speed of data transfer. Some protocols also support message acknowledgement and data compression designed for reliable and/or high-performance network communication. Hundreds of different computer network protocols have been developed each designed for specific purposes and environments.
The seven-layer OSI (*Open systems Interconnection*) model, created by the ISO (*International Standards Organization*), defines internetworking environments. It provides a description of how software and hardware interact to permit communication between computers. An *interface* separates each layer from those above and below it; each layer provides services to the layer directly above it. We will learn the role and functionality of these layers in the later units of this material.



CHECK YOUR PROGRESS

5. Choose the correct answer:

i) In simplex mode of transmission
   a) Data transmission is one way
   b) Data can be transmitted to small distances only
   c) Data format is simple
   d) None of the above

ii) In half-duplex data transmission
   a) Data can be transmitted in one direction only
   b) Data can be transmitted in both directions
   c) Data can be transmitted in both directions simultaneously
   d) None of the above

iii) In _____ transmission, a start bit and a stop bit form a character byte.
   a) asynchronous b) synchronous
   c) parallel d) none of these

iv) _____ communicates bits simultaneously over multiple lines.
   a) serial transmission
   b) synchronous communication
   c) asynchronous communication
   d) parallel transmission



4.12 LET US SUM UP

- **Networking** is the linking of computers (not necessarily over large distances so they can communicate, sharing hardware and software, thus uniting processing power.
- Various concepts such as connection-oriented and connection-less services, broadcast and point-point networks etc. are also presented in this unit.
- We are also acquainted with the different types of computer networks with their relative advantages and disadvantages.
- The concept of network topologies refers to the layout of connected devices



4.13 FURTHER READING

- Tanenbaum, A. S., & Wetherall, D. (2011). *Computer Networks,* Prentice Hall India.
- Stallings, W. (2007). *Data and Computer Communications.* Pearson Prentice Hall.



4.14 ANSWERS TO CHECK YOUR PROGRESS

Answer 1 : i) True, ii) False, iii) True, iv) True

Answer 2 : i) Topology, ii) mesh, iii) Hybrid, iv) star, v) Mesh, vi) more, vii) bus, viii) star

Answer 3 : Bus topology

Answer 4 : Ring topology

Answer 5 : i) a) Data transmission is one way

ii) b) Data can be transmitted in both directions

iii) a) asynchronous

iv) d) parallel transmission



4.15 MODEL QUESTIONS

Q1. Define computer network. What are the goals of computer networks?

Q2. What is the difference between circuit switching and packet switching?

Q3. Explain the concept of message switching.

Q4. What are the advantages and disadvantages of circuit switching?

Q5. Compare and contrast the different network topologies.

Q6. List the relative advantages and disadvantages of bus, ring and star topologies.

Q7. Differentiate between connection-less and connection-oriented networks.

Q8. What are the different modes of communication? Distinguish simplex, half-duplex, and full-duplex mode of transmission.

Q9. Distinguish between peer-to-peer LAN and client-server LAN.

Q10. Describe briefly the various features of serial and parallel transmission.

*****

UNIT 5 : NETWORK MODELS

UNIT STRUCTURE

5.1 Learning Objectives
5.2 Introduction
5.3 Design Issues of the Layer
5.4 Protocol Hierarchy
5.5 ISO-OSI Reference Model
5.6 Terminology
5.7 Internet (TCP/IP) Reference Model.
5.8 Comparison of ISO-OSI & TCP /IP Reference Model
5.9 Let Us Sum Up
5.10 Further Reading
5.11 Answer to Check Your Progress
5.12 Model Questions

5.1 LEARNING OBJECTIVES

After going through this unit, you will be able to :

- describe about the design issues of network architecture.
- describe the functions of various network layers.
- define the various network terminologies.
- distinguish between connection-oriented and connection-less services.
- compare between OSI and TCP/IP model.

5.2 INTRODUCTION

In the previous units, we have learnt the basics of computer networks. Topics like different switching twchniques– circuit switching, message switching and packet switching along with connection-oriented and connection less service. Different categories of network and network topology along with different transmission types were disccussed in detail in the previous unit. In this unit, we will discuss the different types of network models like ISO-OSI reference model and TCP/IP model. In the next unit, we will explore the concepts related to physical layer.

5.3 DESIGN ISSUES OF THE LAYER

The subject of computer network and its structure can be best understood by studying its underlying communication software and the related hardware. Historically, the first computer networks were designed to work on hardware which proved to be fatal afterwards. It leads to inflexibility to the computer networks. The later developments of computer networks were based primarily on highly structured software implemented on ever changing hardware. The communication software in a computer network is a much thought out software which is organized as layered software. The internal structure of the layered software is being discussed in this unit.

Before proceeding to learn about the layered structure of network software and the various aspects of computer network, we should be acquainted with the different design issues of the layer.

- For interchanging data between two processes running on two computers there must be some addressing technique for communication correctly.
- The communication may be unidirectional (simplex communication) or in both direction alternately one after another (half-duplex communication) or bidirectional (full-duplex communication).
- Error control is also an important issue because physical communication channels are not perfect; they are error prone. Now both communicating parties must agree on which error detecting and correcting code to use. The receiver should also let the sender know which message is received correctly and which is not.
- During transfer, messages are often divided into pieces and all the pieces may not be transferred in proper order. This loss of sequencing has to be taken care of by the protocol concern.
- There must be provision to refrain a fast sender from swamping a slow receiver.
- The concerned layer may decide to use the same connection for several unrelated conversations by multiplexing and de-multiplexing technique.
- When there are many paths between the sender and the receiver, a decision must be taken as to which route is to be taken. The decision may be static or dynamic.

5.4 PROTOCOL HIERARCHY

From the design issues discussed above it is apparent that the computer network design is a complex job. To reduce the complexity of design, most of the computer networks are organized as a stack of layers, one above another, taking the service offered by the layer below. In this structure, a lower layer gives certain services to the layer just above it, at the same time it shields the details of the implementation of those services from the upper layer.

In the layered organization, a particular layer on one machine talks with a peer layer on another machine. During the conversation, they follow some rules and conventions. These rules and conventions are collectively known as **protocol**. Here we have used the term **peer** which means entities comprising the corresponding layer on different machines. So, it is the peers that talk using their own protocol.

In Fig 5.1 a five layer network is shown. Here we can see dotted arrow between peers and solid arrow between any two adjacent layers. Dotted arrow indicates that layer *n* on one machine communicates with layer *n* on the other machine. During communication they follow their protocol.

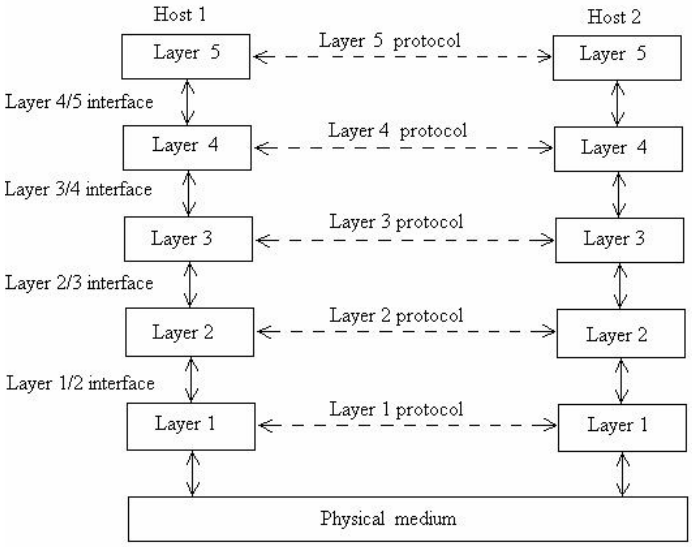


Figure 5.1 : Layers, protocols and interfaces

Every layer communicates with the peer layer on the other machine. But all these communications are virtual since no data really moves through these

dotted lines; rather data moves through solid arrow lines. In the process of communication, each layer hands over their data to the layer immediately below it. In this way data moves downward until the lowest layer, below which lies the physical medium through which actual communication takes place. This is the picture in the sending machine. The picture is just opposite in the receiving machine where data moves upward after it enters the computer from the physical medium.

In the figure 5.1 each layer gets **service** from the layer just below it. A layer performs some primitive operations to give the service required by the immediate upper layer. The transaction between two adjacent layers takes place as per a predefined guideline where each layer performs a specific collection of functions. So, there exists an **interface** between each pair of adjacent layer through which they exchange data and information and in the process, the upper layer gets service from the lower layer. As long as the interface is unchanged, one can change the underlying hardware at any time without affecting the network operations. As for example underground copper cables are gradually replaced by fiber optic cables but it is not hampering the operations of the existing network because the interfaces between the layers remain unchanged.

The layers and their protocols are collectively known as network **architecture**. In network architecture each layer has its own protocol. The protocols used by all the layers of a network system is called **protocol stack**. Stack means something arranged one above another, like stack of bricks or something similar. In a layered network architecture, layers are arranged from bottom to top, one upon another, hence their respective protocols are also lies one upon another. That's why the name protocol stacks. In the same sense, protocol hierarchy indicates that the protocol of any layer is hierarchically above the protocol of the layer immediately below it.

5.5 ISO-OSI REFERENCE MODEL

The OSI reference model is shown in figure 5.2. The International Standard Organization (ISO) developed a proposal for a network model and the resulting model is known as ISO – OSI reference model. The ISO-OSI model is used to describe the complete mechanism of data transfer through different stages of a network and adhere to certain rules for smooth transition. This model has seven layers and the layers are :

i) Application layer
ii) Presentation layer
iii) Session layer
iv) Transport layer
v) Network layer
vi) Data link layer
vii) Physical layer

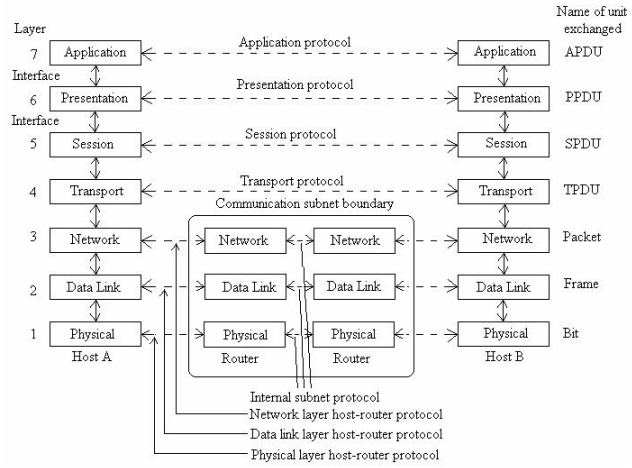


Figure 5.2 : Layers, protocols and interfaces

i. Physical Layer : The physical layer is formed by hardware devices and is responsible for transmitting raw bits over the communication channel. This layer is to ensure how to send a 1 bit from the sending computer as 1 bit to the receiving computer; not as 0 bit. The physical layer also deals with the issues of how many bits per second will be transmitted, what level of voltage will be used to represent 1 and 0, whether transmission will be unidirectional or bidirectional, how the initial connection be established and terminated at the end, how many pins of network the connector has and which pin is for what etc etc. Hence, the design issues of the physical layer is mostly mechanical, electrical and procedure oriented.

ii) Data Link Layer : The data link layer use the raw transmission facility and transforms it to an apparently error free facility to be used by the network layer. This layer breaks the input data into frames by inserting appropriate frame boundary, transmits the frames sequentially and processes the acknowledgment sent back by the receiving computer. If a frame is completely destroyed by noise burst, it is the duty of data link layer to retransmit it from the source machine. Data link layer also ensures that a fast sender be not allowed to swamp a slow receiver by sending data at a higher rate than it can be handled by the receiver. This is called flow control. In a broadcast network it is the duty of Medium Access Control (MAC) sub-layer of

data link layer to decide who will access the transmission medium at a particular time.

iii) Network Layer : The network layer controls the operation of the subnet. The layer is to determine how packets are routed from source to destination. The routing may be static or dynamic depending on traffic load, and availability of channel.

Too many packets may cause congestion (traffic-jam) and control of such congestion is also a duty of network layer.

The subnet operation require cost, hence some accounting function is also there built into the network layer. When a packet crosses national boundary some other aspects of accounting has have to be dealt with by network layer.

Packets have to travel in between heterogeneous network running on different platforms using different network protocol and network layer is also responsible to resolve all the problems arises out of such situations.

iv) Transport Layer : The transport layer is to accept data from the session layer, break it into smaller units if necessary, hand over these to the network layer and ensures that the pieces are all delivered correctly to the receiver. The above duties must be done efficiently and in such a way that it will not affect the upper layer in case there is any change in the hardware.

Normally, the transport layer creates individual connection for each session. If high throughput is required, the transport layer may establish multiple network connections, dividing the data among individual connections thereby improving the through put together.

To reduce cost the transport layer may also multiplex several transport connections on to the network connection. However, multiple connections or multiplexing must not be seen by the session layer.

The transport layer also determines the type of service given to the user of the network. Error free point-to-point connection is the most popular transport layer service where messages or bytes are delivered in order in which they were sent. Another kind of transport service is the transportation of isolated messages with no guarantee of delivery. The transport layer is a true end layer between source to destination. The transport layer also performs flow control.

v) **Session Layer :** This layer offers facility to different users on different computers to establish session between them. A session allows an user to remotely log into a distant machine and to transfer the file between the two machines. Session layer performs token management to provide unidirectional communication. It also provides a service called synchronization.

vi) **Presentation Layer :** This layer performs data presentation job by following the syntax and semantics rules. Before presenting data to the user, it transforms data into their acceptable form. For example, say we write a name as Sri Nilimoy Choudhury, whereas it will be written as Choudhury, Mr. Nilimoy in Europe or USA. We write date as dd/mm/yyyy, currency as Rs. while in a western country it is written as mm/dd/yyyy and $ etc. So, these conversions are done by the presentation layer.

vii) **Application Layer :** This layer is the nearest layer to all the network users. It offers variety of protocols that are commonly needed. It helps to transfer file. Different file systems have different meaning in different machines with different data formats etc. When files are transferred from one machine to another with different file system, the application layer take necessary steps to resolve the abnormalities.



CHECK YOUR PROGRESS

1. What is *service* and interface?
   .....
   .....
   .....
2. What do you mean by network architecture?
   .....
3. How many layers are there in ISO-OSI model?
   .....
4. What are the prime duties of Data Link Layer?
   .....

5.6 TERMINOLOGY

When we study the subject of computer network, we encounter many terminologies associated with the subject. Already we have come across a few terminologies such as **peer, interface, protocol, service** etc. Some others are briefly described below:

- **Entity :** The active components in each layer are called entities. Entity can be a software process or hardware like an intelligent input/ output device etc.
- **Peer Entity :** Entities in the same layer running on different machines are called peer entities.
- **Service provider and Service user :** In layered architecture, layer *n* provides a service which is used by layer *n+1*. Here layer *n* is service provider and layer *n+1* is service user.
- **SAP (Service Access Points) :** A layer *n* offers services to layer *n+1* at a place which is called service access points. Each SAP has a unique address for identification.

5.7 THE INTERNET (TCP/IP) REFERENCE MODEL

The Internet model of network architecture, also known as TCP/IP reference model, was developed much earlier than the OSI reference model. It was evolved from the US Department of Defense's (DoD) research network-ARPANET. Eventually the ARPANET connected many universities and other government organizations. During the process of interconnecting all these through the existing telephone lines, satellite link and radio, the protocol used for the ARPANET had experienced trouble and, therefore, a new architecture was needed to overcome that. The architecture that evolved this way came to be known as the INTERNET or TCP/IP Reference Model. The model has four layers, namely :

1. Host-to-Network Layer
2. Internet Layer
3. Transport Layer
4. Application Layer

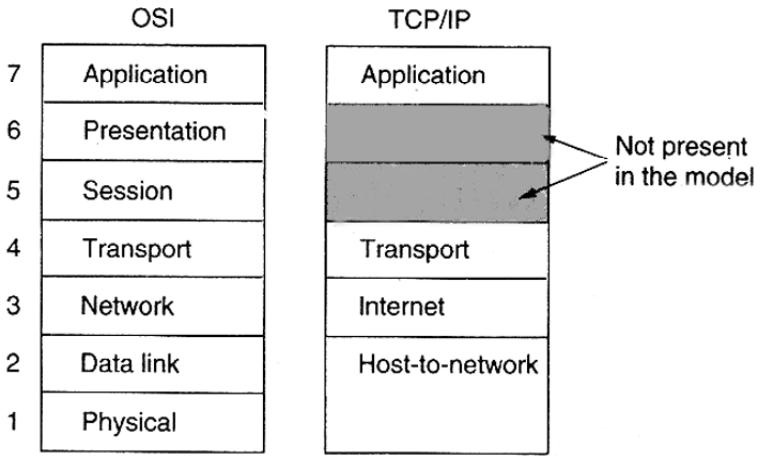


Figure 5.3: The TCP/IP reference model

1. **Host-to-Network Layer :** In TCP/IP reference model the bottom layer is not defined clearly. Yet we can consider that whatever is there below the internet layer is the bottom layer in the TCP/IP model. In the model, the host has to connect to the network using some protocol so that it can send IP packets over it.
2. **Internet Layer :** The DoD planned to set up their inter-network in such a way that it had to survived even if a particular link fail in a probable war. So, this requirement led to connectionless packet-switching network instead of a connection-oriented circuit-switching network. The internet layer is designed to fulfill the goal of the architecture in such a way that it became the linchpin of the whole architecture. The job of this layer is to pump the packets from the host machine into any network and to help the packets to go to the destination independently. In this style of communication, packets may arrive in a different order than they were sent. This layer has its own protocol called **Internet Protocol (IP)** and a specified packet format. Packets routing as well as congestion avoidance are the major issues in this layer. This layer is similar to the network layer in OSI reference model.
3. **Transport Layer :** The layer immediately above the internet layer is called the transport layer. This layer communicates with the peer layer on the other machine. It has two protocols namely **TCP and UDP.**

   TCP is a reliable connection-oriented protocol that ensures delivery of byte stream from source machine to destination machine. This layer breaks the byte stream received from the upper layer into

messages and passes them to the internet layer. At the receiving machine, these messages are reassembled by the TCP and pass them to the upper layer. Flow control is also another job of this layer to restrict a fast sender from swamping a slow receiver.

UDP is another protocol used by this layer which offers an unreliable, connectionless service. Here, unlike TCP, sequencing and flow control is not done. When prompt delivery is more important than accurate delivery, UDP is used. It is also widely used in one time client-server type communication. Email, video or sound transmissions are some of these applications.

4. **Application Layer :** In OSI model, there are session and presentation layer above the transport layer. In TCP/IP model these two layers are absent. So, on top of the transport layer, the application layer is present in TCP/IP reference model. All high level protocols are present here. TELNET, FTP, SMTP are some of the early inclusions. Later, other protocols such as DNS, NNTP, HTTP are added to this layer.

5.8 COMPARISON OF ISO-OSI & TCP /IP REFERENCE MODEL

There is not much difference between the OSI and TCP/IP reference model. In both, the concept of independent stack of protocols is used. The function of the layers are also more or less are same. In both models, starting from bottom up to transport layer, the functions of the layers is to provide end-to-end network independent transport service for communicating processes. The layers above the transport layer are application oriented.

On the other hand, there are some differences between the two models. In OSI reference model, three concepts are distinct. These are–

1. Services,
2. Interfaces and
3. Protocols.

The *Service* tells what a particular layer executes in support of the layer just above it, not how these services are access by the above layer or how these are provided by the layer. The *Interface* tells how the above layer access the services provided by the layer just below it. It also does not say

how these are provided. The *Protocols* used between peer layers are the set of rules agreed upon by both the layers to get the job done. Protocols can be changed without affecting the software in the higher layers.

In TCP/IP model, there is no such distinction between Service*, Interface and Protocol*. Therefore, in TCP/IP reference model the protocols are not so hidden as in the OSI reference model. As the technology advances, protocol between a peer layer can be replaced more easily in OSI model than in TCP/IP model.

In OSI model, the layered structure was thought out before the protocols were invented. the model. So there was no problem of fitting the protocols into the model.

The two models also differ in their number of layers; in OSI model there are seven layers whether TCP/IP model has four layers. Both have (inter)network, transport and application layers in common but the other layers are not same.

In OSI model, network layer supports both *connection-oriented and* connectionless communication and the transport layer supports only *connection-oriented* communication. On the contrary, in TCP/IP model the network layer supports only *connectionless* communication but transport layer supports both the mode i.e. *connectionless* and *connection-oriented* communication.



CHECK YOUR PROGRESS

5. What are the two services that can be offered by a layer to the layer above it?

   .....

6. How many layers are there in TCP/IP reference model?

   .....

7. Which model of network architecture was developed earlier – OSI or TCP/IP?

   .....

8. Which two layers of OSI model are absent in TCP/IP model?

   .....



5.9 LET US SUM UP

- There must be some addressing technique for smooth communication between two computers.
- Simplex communication means one directional communication.
- In half-duplex communication, communication is bidirectional alternatively; not at the same time.
- Full-duplex communication means bidirectional communication at all time.
- Most of the computer networks are organized as a stack of layers, one above another, taking the service offered by the layer below.
- Protocol means a set of rules followed by two peers during communication.
- Entity: The active components in each layer are called entities. Entity can be a software process or hardware like an intelligent input/output device etc.
- Peer Entity: Entities in the same layer running on different machines are called peer entities.
- Service provider and Service user: In layered architecture, layer *n* provides a service which is used by layer *n+1*. Here layer *n* is service provider and layer *n+1* is service user.
- Service Access Points(SAP): A layer *n* offers services to layer *n+1* at a place which is called service access points. Each SAP has a unique address for identification.
- The layers and their protocols are collectively known as network architecture.
- OSI model is a seven layer network architecture.
- A layer can offers two different types of services to the layer immediately above it – connection-oriented and connectionless service.
- In TCP/IP model of network architecture, there are four layers.
- Most of the computer networks are based on TCP/IP model.



5.10 FURTHER READING

- Tanenbaum, A. S., & Wetherall, D. (2011). *Computer Networks,* Prentice Hall India.
- Stallings, W. (2007). *Data and Computer Communications.* Pearson Prentice Hall.



5.11 ANSWERS TO CHECK YOUR PROGRESS

Answer 1: A layer performs some primitive operations to fulfill the requirement of the layer immediately above it. This responsibility of the lower layer is called service.

On the other hand interface is the junction between two adjacent layersacross which the lower layer provides the services to the layer above it.

Answer 2: The layers and their protocols are collectively known as network architecture.

Answer 3: Seven layers.

Answer 4: Framing, flow control , error control and access control.

Answer 5: Connection-oriented service and connection-less service.

Answer 6: Four layers.

Answer 7: TCP/IP reference model was developed earlier than OSI model.

Answer 8: Session layer and presentation layer.



5.12 MODEL QUESTIONS

Q1. What are the design issues of layered architecture of a computer network?

Q2. What is the principal difference between connection-less and connection-oriented communication ?

Q3. Describe ISO-OSI reference model of computer network.

Q4. Define : peer entity, protocol, protocol stack, SAP.

Q5. Which of the OSI layers handles each of the following :

a) Breaking the transmitted bit stream into frames.

b) Determining which route through the subnet to use.

Q6. Describe the functions of various layers of the TCP/IP model.

Q7. What are the responsibilities of the network layer in the TCP/IP model?

Q8. What are the responsibilities of the transport layer in the TCP/IP model ?

Q9. How do the layers of the TCP/IP model correlate to the layers of the OSI model ?

Q10. Match the following to one or more layers of the OSI model:

a) Flow control
b) Route determination
c) Provides access for the end user
d) Interface to the transmission media.

******

UNIT 6 : PHYSICAL LAYER

UNIT STRUCTURE

6.1 Learning Objectives
6.2 Introduction
6.3 Physical Layer
6.4 Transmission Media
    6.4.1 Guided Media
    6.4.2 Unguided Media
6.5 Repeaters
6.6 Let Us Sum Up
6.7 Further Reading
6.8 Answers to Check Your Progress
6.9 Model Questions

6.1 LEARNING OBJECTIVES

After going through this unit you will be able to:

- define role of physical layer
- define transmission media
- describe classifications of transmission media
- compare different transmission media
- describe working of repeater

6.2 INTRODUCTION

In the previous unit, we have learnt about network models. The ISO-OSI reference model along with the TCP/IP model were explained in detail in the previous unit. A comparison between the two models was also provided. In this unit, we will learn the basics about the bottom most layer in both network models i.e. the physical layer. Different types of the transmission media that are used in the physical layer are explained in detail in this unit. In the next unit, we will explore the concepts related to data link layer.

6.3 PHYSICAL LAYER

Physical layer is the bottom most layer in the OSI model of computer network.

Data transmitted by the network in true sense is transmitted by the transmission medium that lies below the physical layer. It is the physical layer which actually controls the activities of the transmission medium. Transmission media connects all the physical components of a computer network.

On the upper side the physical layer provides service to the datalink layer. The datalink layer organizes data to be transmitted in the form of frames which consists of 0s and 1s. These are now handed over to the physical layer to make them ready to be sent through the transmission medium.

The bits of 0s and 1s are first transformed into appropriate signal by the physical layer and then these are transmitted through the real transmission medium for the destination.

Transmission media are the passive things that carry the signals from source to destination. In case of verbal communication, air is the medium of transmission. In computer networks copper cable, fiber optic cable or free space carrying electromagnetic waves play the role of transmission media.

6.4 TRANSMISSION MEDIA

A transmission medium is defined as anything that carries signals from source to destination. For the purpose of networking, transmission media are classified into two categories viz., guided media and unguided media. Classification of transmission media can be shown as block diagram of figure 6.1.

6.4.1 Guided Media

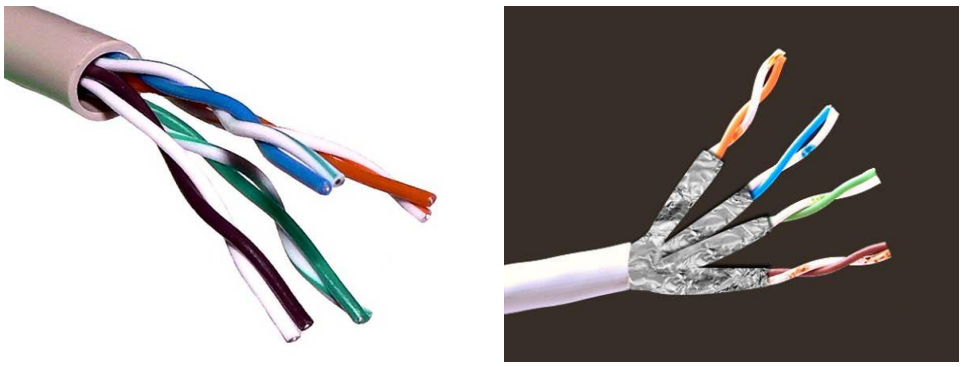
Guided media means some materials that provide a conduit path for the signals to go from one network device to another network device. It provides a guidance to the signals and does not allow the signals to leave the path. Guided media includes-

- Twisted pair cable
- Coaxial cable
- Fiber-optic cable

The first two media are made of copper that carry electric signals and the third one i.e fiber optic cable is made of transparent fiber capable of carrying light signals.

● **Twisted pair**

A twisted pair is a pair of insulated copper wires twisted together. Figure 6.2 shows a twisted pair.



(a) An unshielded twisted-pair cable (b) A shielded twisted-pair cable

Figure 6.1 : Twisted pair

In a twisted-pair cable, there are four pairs of wires twisted together. One of the wires in a pair carries signals to the destination and the other wire is used as ground reference. For any kind of transmission of electrical signal, there is a possibility of external interference. These types of electro-magnetic interferences are known as noise. When two nearby wires carry signals, they create noise for each other. It is called crosstalk. In a twisted pair, twisting is done to reduce noise and crosstalk.

Twisted pair cables are broadly of two types, viz., unshielded twisted pair (UTP) cable and shielded twisted pair (STP) cable. Figure 6.2(a) shows an UTP cable whereas Figure 6.1 (b) depict a STP cable. Shielding is provided for improvement of performance of the cable. STP cables are heavier and costlier than UTP cable.

Depending upon the number of twisting per unit length, twisted pair cables are categorized in another way. The Electronic Industries

Association (EIA) has formulated a standard to classify UTP cable into seven categories, the details of which are tabulated in Table 6.1

Application of twisted pair cable

Twisted pair cables are used in telephone lines that provide voice and data connection to the subscriber from the telephone exchange. DSL lines are provided by twisted pair cables to provide high speed data connection using high bandwidth channel. Twisted pair cables are extensively used in Local Area Network (LAN) in 10Base-T and 100Base-T type of network.

Table 6.1 Category of UTP cable

Category of UTP cable	Description
CAT1	Speed up to 1 Mbps, used in old telephone cable
CAT2	Speed up to 4 Mbps, used in Token ring networks
CAT3	Speed up to 10 Mbps, used in Token ring and 10Base-T Ethernet
CAT4	Speed up to 16 Mbps, used in Token ring Networks
CAT5	Speed up to 100 Mbps, Ethernet, Fast Ethernet, Token Ring
CAT5e	Speed up to 1 Gbps, used in Ethernet, Fast Ethernet, Gigabit Ethernet
CAT6	Speed up to 10 Gbps, used in Gigabit Ethernet
CAT6a	Speed up to 10 Gbps, used in Gigabit Ethernet, 10G Ethernet (55 Meters)
CAT7	Speed up to 10 Gbps, used in Gigabit Ethernet, 10G Ethernet (100 Meters)

- **Coaxial Cable**

Coaxial cables are the guided media that carry the signal of higher frequency range compared to twisted pair cable. Coaxial cables are also called *coax* in short. There are two types of coaxial cables used mostly and these are 50 ohm cable and 75 ohm cable. The 50 ohm cable is used for digital transmission and 75 ohm cable is used for analog transmission. Due to the shield provided, this cable has excellent noise immunity. It has a large bandwidth and low losses. Coaxial cables are easy to install. They are often installed either in a device to device linear bus (Ethernet) or a star (ARC net).

A coaxial cable consists of a small solid core placed at the center covered by an insulating layer. Above the insulation, coper mesh is provided to achieve shielding from external noise. On the outside the cable is covered by a protective plastic jacket that protects it from physical dangers. It has a data rate of 10 Mbps which can be increased with the increase in diameter of the inner conductor. The specified maximum number of nodes on a thin coax segment is 30 nodes and on a thick coax it is 100 nodes. If the thickness of the inner conductor is 0.89 mm, the cable is called thin coax. If the thickness is 2.17 mm, the cable is called thick coax.

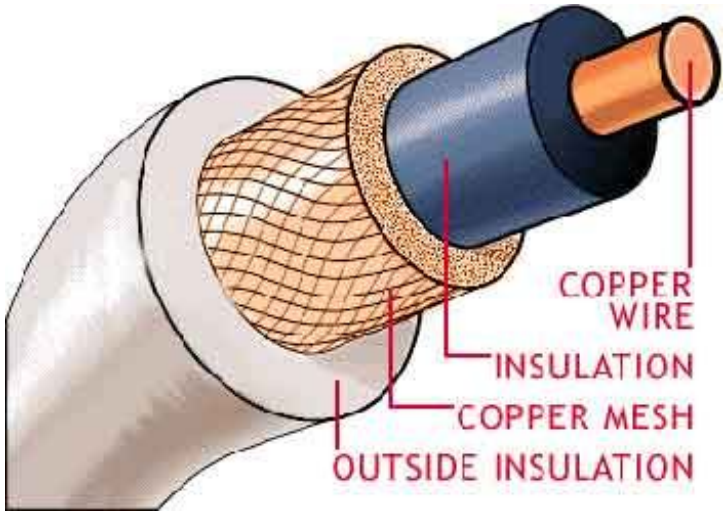
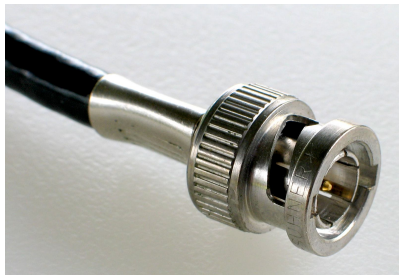


Figure 6.2: Construction of a coaxial cable

A coaxial cable can also be classified into two categories, viz., baseband and broadband. A baseband coaxial cable transmits a single signal at a time at very high speed. On the other hand, a broadband coaxial cable can transmit many simultaneous signals using different frequencies. A baseband cable is mainly used for LANs

Coaxial Cable Connectors



Coaxial cables are connected to different devices by using some connectors. The most common connector is the BNC connector. The abbreviation BNC comes from the name of the three inventors **Bayone, Neill** and **Concelman.**

Applications of Coaxial Cable

Coaxial cables were extensively used in telephone networks to carry analog signals where a single cable could carry about 10,000 voice signals. In later period of time, it was used to carry digital telephone signals with a data rate up to 600 Mbps. Now-a-says coaxial cables are largely replaced by fiber-optic cables.

Cable TV networks also use coaxial cables. However, here also, coaxial cables are gradually being replaced by fiber-optic cables.

Another mostly used application of coaxial cable was in traditional Ethernet LANs. 10Base2 and 10Base5 were the example of Ethernet LAN where coaxial cables were used until recent past.

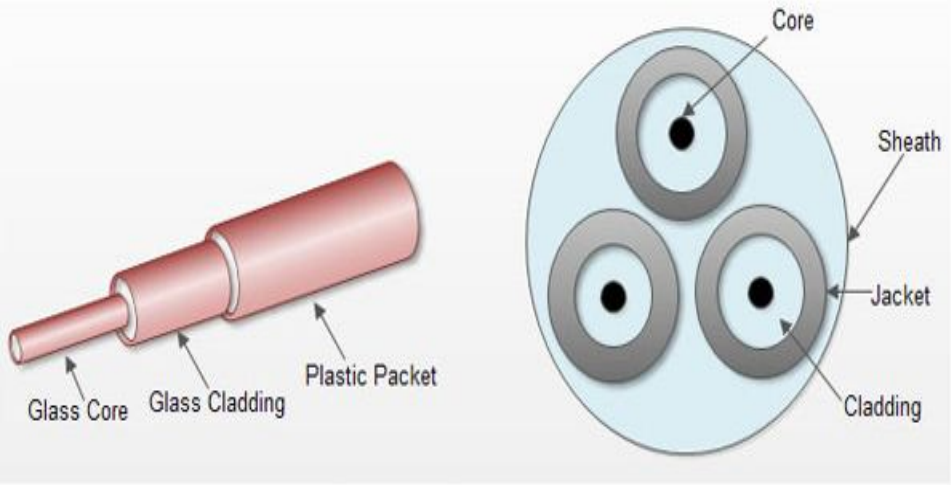


Figure 6.3: (a) Construction of optical fibre (b) Bundle of 3 optical-fibres with outer sheath

- **Fiber-Optic Cable**

Fiber-optic cable consists of a **thin glass fibers** or plastic that can carry information at frequencies in the visible light spectrum and beyond. The typical optical-fiber cable consists of a very **narrow strand of glass** called the **core**. Around the core is a concentric layer of glass called the **cladding**

An optical fiber carries light signal by following the principle of internal reflection of light. In figure 6.4 it is shown how light behaves when it moves from denser to lighter medium. If the angle of incidence is more that critical angle, light gets reflected back to the denser medium. This is called internal reflection.

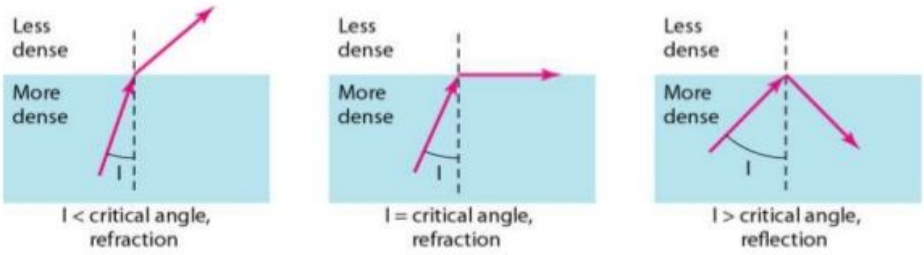


Figure 6.4: Principle of internal reflection of light ray

In a fiber-optic cable, light propagates by repeated internal reflection of light as shown in figure 6.5

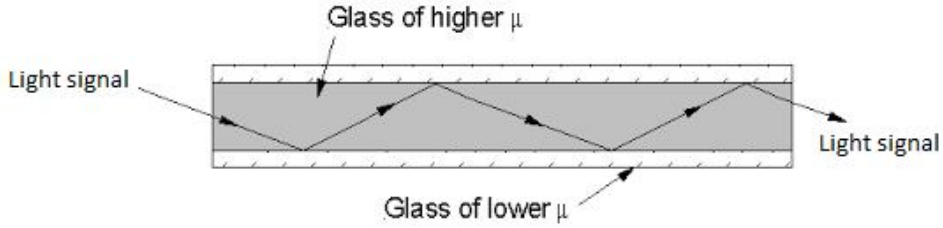


Figure 6.5: Propagation of light through optical-fiber cable

Mode of propagation

There are two mode of light propagation through an optical-fiber cable- mono mode and multimode. In mono mode an optical-fiber can carry only one signal. Such a cable is made with much smaller diameter. Its material density is also much lower. In multimode, the fiber optic cable can carry more than one signal at a time. Multimode cables are of two types- step index and graded index.

In step index fiber, the core is made of glass fiber having same density from center to the edges. The cladding above the core is made of much lower density material. A beam of light travels through this constant density material in a straight line until it reaches the interface of the central core and the cladding above the core. Due to abrupt change in the density from higher to lower, light signal gets reflected according to the principle of internal reflection. In this way light signal propagates through the step index fiber-optic cable.

In graded index type, the density of glass fiber material changes from highest at the center to the lowest at the outside interface gradually.

Components of fiber-optic transmission

An optical transmission system has three basic components

(a) Light source

(b) Transmission medium (fiber optics)

(c) Detector

(a) Light source: In such a system a pulse of light indicates bit 1 and the absence of light indicates bit 0. Light source can be an LED or a laser beam.

(b) Transmission medium: Transmission medium is the ultra-thin fiber of glass.

(c) Detector: A detector generates an electrical pulse when the light falls on it,

Applications of fiber- optic cable

- It is a very good transmission medium for communication. Fiber-optic cables are extensively used in long haul backbone networks such as the Internet subnet.
- Some cable TV networks also starts using fiber-optic cable for their TV signals.
- 100Base-FX network (fast Ethernet) and 1000Base-X LAN are extensively using fiber-optic cable as their transmission media.

Advantages

- High bandwidth- Fiber-optic cable are suitable for much higher bandwidth in comparison to twisted pair or coaxial cable.
- Immunity to electro-magnetic interference- Since fiber-optic cable carries light signals, these are not effected by any stray electro-magnetic noises.
- Less attenuation- Fiber-optic cable can carry signals to a very long distance without significant signal attenuation. In fiber-optic transmission, a signal can be transmitted up to 50 km without regeneration. On the other hand, repeaters are needed after every 5 km in case of a coaxial cable or twisted pair.
- Resistance to corrosion- Glass fiber is highly resistant to corrosive agent than other conducting materials like copper.

- Light weight – Fiber-optic cables are very light in weight. It reduces the cost of transportation, installation etc.
- Immunity to tapping- Light signals passing through a fiber-optic cable is very difficult to trap and therefore it is much immune to data theft risk.

Disadvantages

- Difficulty in installation and maintenance: Due to its new and sophisticated technology, installation of fiber-optic cable needs an expertise which may not be available everywhere. Joining of optical fiber needs splashing machine which is very expensive and needs expertise to operate.
- Unidirectional light propagation: Propagation of light signal is unidirectional in fiber-optic cable. Therefore two cables are needed for bi-directional signal communication.
- Cost : The cost of fiber-optic cable and its interfaces are relatively costly than other guided media.



CHECK YOUR PROGRESS

1. Transmission media are classified as
   (a) guided and unguided media
   (b) guided and bounded media
   (c) unguided and unbounded media
   (d) wireless and free space media
2. Twisting is done
   (a) To increase length of wire
   (b) To decrease data rate
   (c) To decrease cross talk
   (d) To increase interference
3. Coaxial cable is
   (a) A transmission protocol
   (b) A transmission medium
   (c) A wave guide
   (d) An wireless medium
4. Fiber-optic cable
   (a) Transmits electric signal

(b) Transmits light signal
(c) Transmits fibrous signal
(d) Transmits chemical signal

6.4.2 Unguided Media

When transmission is made without using any cables it is called wireless transmission. In wireless transmission, the media used are not bounded by physical conductor and these are called unguided media. Transmission through unguided media wireless communication is becoming popular. This transmission uses Microwave, Radio wave, Infrared etc. Signals are broadcast through free space and therefore anybody can access it by using appropriate device.

Wireless communication uses a part of electro-magnetic spectrum ranging from 3 kHz to 900 THz. The section of electro-magnetic spectrum defined as radio waves and microwaves is divided into eight smaller sections, named as bands. Use of electronic device in these bands are regulated by some government agencies. These bands are technically labeled from very low frequency (VLF) to extremely high frequency (EHF). Table 6.2 describes various aspects of these bands-

Table 6.2 Bands used in wireless communications

Band	Range	Propagation	Application
VLF (very low frequency)	3-30 kHz	Ground	Long range radio navigation
LF(low frequency)	30-300 kHz	Ground	Radio beacons and navigational locators
MF(middle frequency)	300 kHz – 3 MHz	Sky	AM Radio
HF (high frequency)	3-30 MHz	Sky	Citizen Band (CB), ship / aircraft communication
VHF (very high	30-300	Sky and	

frequency)	MHz	Line-of-sight	VHF TV, FM Radio
UHF (ultrahigh frequency)	300 MHz – 3 GHz	Line-of-sight	UHF TV, cellular , phones paging, satellite
SHF (super	3– 30	Line-of-sight	Satellite communication
high frequency)	GHz		
EHF (extremely	30-300	Line-of-sight	Radar, satellite
high frequency)	GHz		

- **Radio Waves**

Electro-magnetic frequencies ranging from 3 KHz to 1 GHz are normally known as radio waves. By nature, radio waves are omnidirectional. It means that radio waves spread in all parts of an area, not confined to a particular zone. When an antenna transmits radio signals, they propagates in all directions. Alignment of sending end antenna and receiving end antenna is not necessary for radio communication. Omnidirectional property of a radio wave has a disadvantage also. Signal transmitted by two antennas using the same frequency will interfere with each other resulting in serious disturbance to the receivers. Radio waves can travel to a very long distance by using the reflection of ionosphere of atmosphere. This property of radio waves is utilized for long distance broadcasting of AM radio.

Radio wave band is comparatively narrow, under 1 GHz. This band is further sub-divided into smaller sub-bands which are allocated to different radio transmitters. Therefore, their data transmission rates are also low.

Application of radio waves

Radio transmission is extensively used for AM, FM radio, television, maritime radio, cordless telephones and paging.

- **Microwaves**

Electro-magnetic waves having frequencies ranging from 1 GHz to

300 GHz are called micro waves. These are directional in nature, means, micro waves propagate in a manner like beam of light coming out from a torch light. Microwaves transmitted from an antenna can be narrowly focused. A pair of transmitting and receiving antennas can be aligned and used for microwave communication without causing interference with other micro wave communication nearby.

Microwave communication is a line-of-sight communication. These waves cannot penetrate thick walls and these are highly obstructed by high rise building, mountains, dust particles, rain etc.

Bandwidth of microwave is 299 GHz. Therefore its sub-bands are also wide and can provide high data rate for communication. Microwave communication requires permission from government authority.

Microwave communication is unidirectional. For bidirectional communication using microwaves two antennas are required – a parabolic dish and a horn antenna. In a parabolic dish antenna signals coming parallel to its principal axis get reflected to its

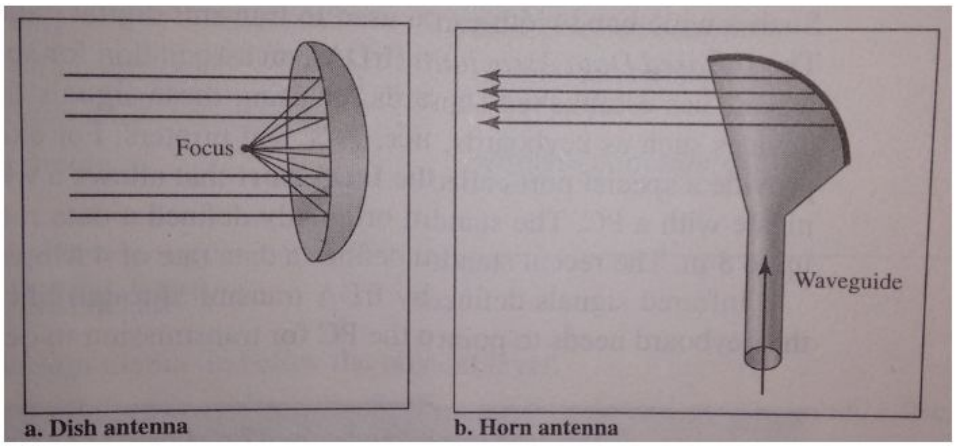


Figure 6.6: Unidirectional antennas of microwaves transmission

focus as shown in Figure 6.6a. Communication device is placed at the focus and receiving is thus accomplished. Outgoing of data is done by horn antenna aligned with the dish antenna. A horn antenna is shown in Figure 6.6b.

Applications of microwaves

Microwaves are very good for one-to-one communication due to their unidirectional properties. For mobile phone communication, satellite networks and wireless LANs microwaves are used.

- **Infrared**

Electro-magnetic spectrum in the range of 300 GHz to 400 THz (wavelength from 1 mm to 770 nm) is known as infrared waves. This range is used for short range communication. Infrared is high frequency waves. Therefore this waves are highly directional and cannot penetrate walls, other obstacles like high rise building, hill etc. These are also greatly absorbed by rain drops, dust particles etc.

Highly directional properties of infrared is advantageous in the sense that one infrared communication does not interfere with other infrared communication taken place nearby. But this same property of infrared is disadvantageous for long distance communication. Moreover infrared communication cannot be used outside a building because of infrared component present in the spectrum of sun light.

Applications of infrared

Infrared waves are excellent for data communication due to its large 400 THz bandwidth. Very high data transmission rate is possible due to its wide bandwidth. An international organization named *Infrared data Association* (IrDA) has established standards for short distance line-of-sight communications by using infrared waves for applications like communication between keyboards, mice, PCs, printers, LCD projectors etc. IrDA ports are now available for performing infrared communications between wireless keyboards, mice and PCs. Such IrDA standard supports data rate up to 4 Mbps.



CHECK YOUR PROGRESS

5. Wireless media means
   (a) guided media
   (b) bounded media
   (c) unguided and unbounded media
   (d) media that transmits optical signal
6. Radio waves are
   (a) unidirectional
   (b) highly directional
   (c) omnidirectional
   (d) unable to penetrate walls
7. Microwaves lies between
   (a) 10 MHz and 500 MHz
   (b) 300 GHz and 400 THz
   (c) 1 GHz and 300 GHz
   (d) 1 GHz and 300 GHz
8. Infrared ranges from
   (a) 3 MHz and 300 MHz
   (b) 30 MHz and 300 MHz
   (c) 1 GHz and 300 GHz
   (d) 300 GHz and 400 THz

6.5 REPEATERS

A repeater is a device used to regenerate a weak or corrupted signal in a computer network. After travelling a certain distance, a signal becomes weak due to attenuation. Repeaters receive signals and regenerate those before attenuation endangers the integrity of the data. After regeneration the signals are forwarded by the repeaters for further distance towards destination. Repeaters are physical layer device. They can practically extend the length of an Ethernet LAN. A repeater cannot connect two LANs; it only connects two segments of the same LAN. Figure 6.7 shows how a repeater connects between two segments of a LAN.

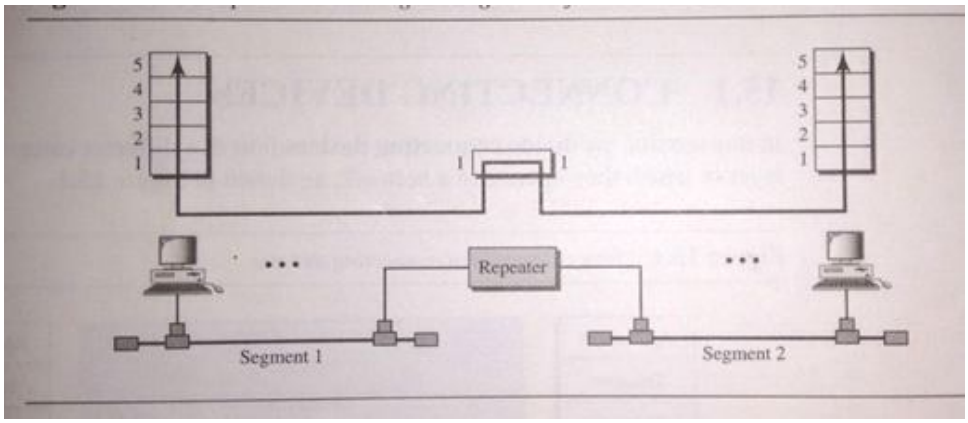


Figure 6.7: A repeater connecting between two segments of a LAN

Sometime a repeater is wrongly termed as an amplifier. But in fact, repeater is not an amplifier. An amplifier amplifies its input signals without discriminating between the intended signal and noise. On the other hand, a repeater does not amplify its input signals. It regenerates the input data bit by bit. Figure 6.8 depicts the function of a repeater.

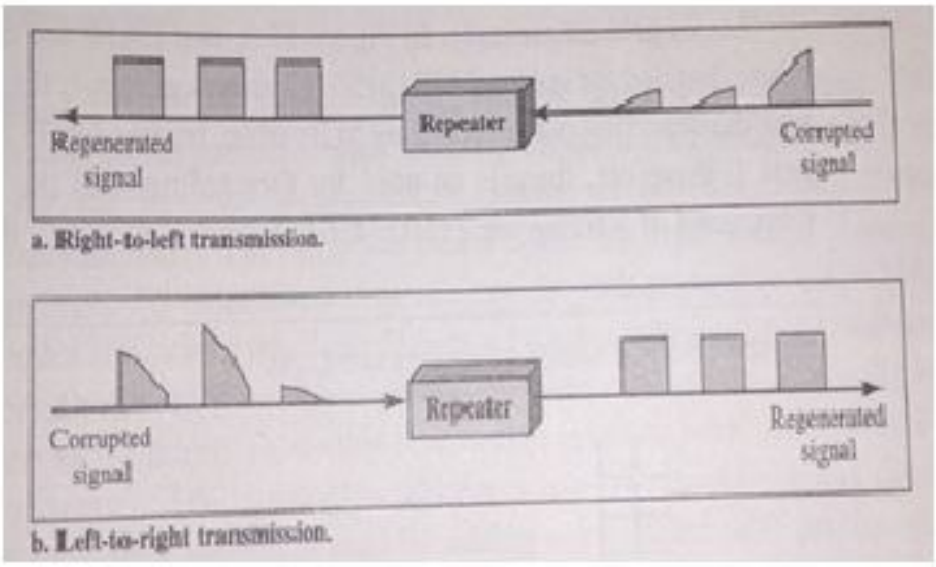


Figure 6.8: Function of a repeater



CHECK YOUR PROGRESS

9. Repeaters are required to
   (a) modify a signal
   (b) transforming a signal
   (c) regenerating a signal
   (d) deform a signal

10. A repeater differs from an amplifier because
    (a) it cannot distinguish between signal and noise
    (b) it regenerates a signal bit for bit.
    (c) it amplifies signals as well as noises also
    (d) it discard signals from its input



6.6 LET US SUM UP

- Transmission media can be anything that can carry data from a source to a destination.
- Transmission media lie below the physical layer.
- Transmission media are broadly classified into guided and wireless media.
- Guided media includes twisted pair, coaxial cable and fiber-optic cable.
- Twisted pair cable consists of two insulated wires twisted together.
- Coaxial cable consists of a central metallic core insulated by a sheath and surrounded by a shield.
- Fiber-optic cables are made from a central glass fiber core surrounded by cladding.
- Unguided media are radio waves, microwaves and infrared.
- Repeaters are signal regenerating device that work in the physical layer.



6.7 FURTHER READING

- Tanenbaum, A. S., & Wetherall, D. (2011). *Computer Networks,* Prentice Hall India.
- Forouzan, A.B. (2006). *Data Communications and Networking*. Tata McGraw-Hill Education



6.8 ANSWER TO CHECK YOUR PROGRESS

Answer 1: (a) **Answer** 2: (c) **Answer** 3: (b) **Answer** 4: (b)
Answer 5: (c) **Answer** 6: (c) **Answer** 7: (c) **Answer** 8: (d)
Answer 9: (c) **Answer** 10: (b)



6.9 MODEL QUESTIONS

Q1. Distinguish between guided and wireless media

Q2. What are the three major classes of guided media?

Q3. Describe the construction of a twisted pair.

Q4. What is the significance of the twisting in a twisted pair cable?

Q5. Name the advantages of fiber-optic cable over twisted pair or coaxial cable.

Q6. Describe the construction of a coaxial cable.

Q7. Write the characteristics of radio waves used for wireless transmission.

Q8. Write the characteristics of microwaves used for wireless transmission.

Q9. Write the characteristics of infrared used for wireless transmission.

Q10. What is a repeater? Distinguish between a repeater and an amplifier.

****

UNIT 7: DATA LINK LAYER

UNIT STRUCTURE

7.1 Learning Objectives
7.2 Introduction
7.3 Data link Layer
7.4 Framing
7.5 Error Control
7.6 Flow Control
7.7 Error Detection and Correction
    7.7.1 Error Correcting Codes
    7.7.2 Error Detecting Codes
7.8 Data Link Protocols
    7.8.1 Stop-and Wait ARQ
    7.8.2 Go-Back-N ARQ
    7.8.3 Selective Repeat ARQ
    7.8.4 HDLC Protocol
    7.8.5 Point-To-Point Protocol
7.9 Let Us Sum Up
7.10 Further Reading
7.11 Answers to Check Your Progress
7.12 Model Questions

7.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- describe error control and flow control
- describe error detection and correction
- describe the error correction and error detection codes
- describe the working of the Stop-and-Wait protocol
- illustrate the Go-Back-N protocol
- illustrate the Selective Repeat protocol
- describe the HDLC and Point-To-Point-Protocol

7.2 INTRODUCTION

In the previous unit, we have learnt the concepts related to physical layer. Different types of transmission media were also covered in detail in the previous unit. In this unit, we will learn about the concepts related to data link layer. Topics like framing, error control, flow control and error detection and correction are discussed in detail. Different data link protocols are also explained in this unit. This is the last unit of the block. In the next block we will discuss the remaining network layers.

7.3 DATA LINK LAYER

In the seven-layer OSI model of computer networking, the data link layer is layer 2. In TCP/IP reference model, it corresponds to, or is part of, the link layer. The data link layer is the protocol layer that transfers data between adjacent network nodes in a wide area network or between nodes on the same local area network segment. The data link layer provides the functional and procedural means to transfer data between network entities and might provide the means to detect and possibly correct errors that may occur in the physical layer. The data link layer is concerned with local delivery of frames between devices on the same LAN. Data-link frames do not cross the boundaries of a local network. Inter-network routing and global addressing are higher layer functions, allowing data-link protocols to focus on local delivery, addressing, and media arbitration. The data link thus provides data transfer across the physical link. That transfer can be reliable or unreliable; many data-link protocols do not have acknowledgments of successful frame reception and acceptance, and some data-link protocols might not even have any form of checksum to check for transmission errors. In those cases, higher-level protocols must provide flow control, error checking, and acknowledgments and retransmission.

In this unit we will study the design principles for the data link layer. which deals with algorithms for achieving reliable, efficient communication of whole units of information called frames between two adjacent machines connected by a communication channel that acts conceptually like a wire. The essential property of a channel that makes it ‘‘wire-like’’ is that the bits are delivered in exactly the same order in which they are sent. Communication channels make errors occasionally. Furthermore, they have only a finite data rate, and there is a nonzero propagation delay between

the time a bit is sent and the time it is received. These limitations have important implications for the efficiency of the data transfer. The protocols used for communications must take all these factors into consideration. These protocols are the subject of this unit.

7.4 FRAMING

To provide service to the network layer, the data link layer must use the service provided to it by the physical layer. What the physical layer does is to accept a raw bit stream and attempt to deliver it to the destination. If the channel is noisy the physical layer will add some redundancy to its signals to reduce the bit error rate to a tolerable level. However, the bit stream received by the data link layer is not guaranteed to be error free. Some bits may have different values and the number of bits received may be less than, equal to, or more than the number of bits transmitted. It is up to the data link layer to detect and, if necessary, correct errors. The usual approach is for the data link layer to break up the bit stream into discrete frames, compute a short token called a checksum for each frame, and include the checksum in the frame when it is transmitted. When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame and possibly also sending back an error report). Breaking up the bit stream into frames is more difficult than it at first appears. A good design must make it easy for a receiver to find the start of new frames while using little of the channel bandwidth. We will look at four methods:

1. Byte count.
2. Flag bytes with byte stuffing.
3. Flag bits with bit stuffing.
4. Physical layer coding violations.

The first framing method uses a field in the header to specify the number of bytes in the frame. When the data link layer at the destination sees the byte count, it knows how many bytes follow and hence where the end of the frame is. This technique is shown in **Figure 7.1(a)** for four small example frames of sizes 5, 5, 8, and 8 bytes, respectively. The trouble with this algorithm is that the count can be garbled by a transmission error. For example, if the byte count of 5 in the second frame of **Figure 7.1(b)** becomes a 7 due to a single bit flip, the destination will get out of synchronization. It

will then be unable to locate the correct start of the next frame. Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts. Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many bytes to skip over to get to the start of the retransmission. For this reason, the byte count method is rarely used by itself.

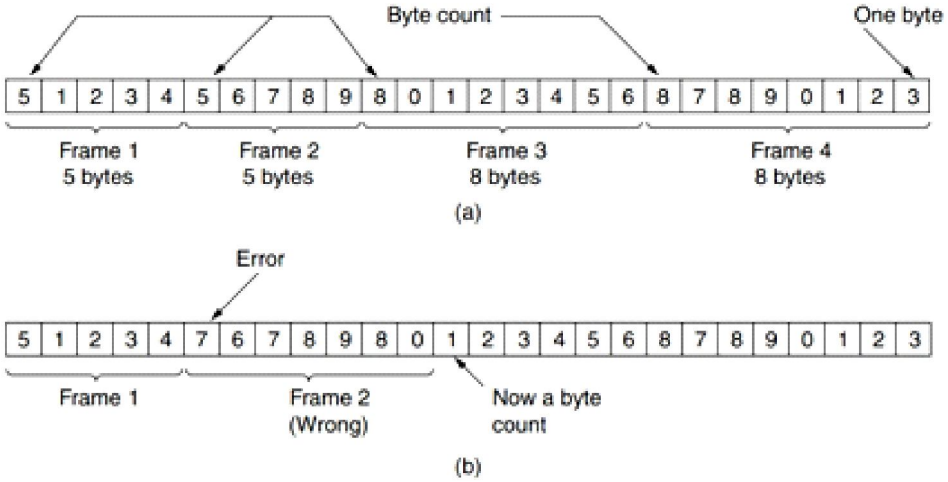


Figure 7.1: A byte stream. (a) Without errors. (b) With one error.

The second framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes. Often the same byte, called a flag byte, is used as both the starting and ending delimiter. This byte is shown in **Figure 7.2(a)** as **FLAG**. Two consecutive *flag bytes* indicate the end of one frame and the start of the next. Thus, if the receiver ever loses synchronization it can just search for two *flag bytes* to find the end of the current frame and the start of the next frame. However, there is a still a problem we have to solve. It may happen that the *flag byte* occurs in the data, especially when binary data such as photographs or songs are being transmitted. This situation would interfere with the framing. One way to solve this problem is to have the sender's data link layer insert a special escape byte (**ESC**) just before each ''**accidental**'' *flag byte* in the data. Thus, a framing *flag byte* can be distinguished from one in the data by the absence or presence of an *escape byte* before it. The data link layer on the receiving end removes the escape bytes before giving the data to the network layer. This technique is called **byte stuffing**. Of course, the next question is: what happens if an *escape byte* occurs in the middle of the data? The answer is that it, too, is stuffed with an *escape byte*. At the receiver, the first *escape byte* is removed, leaving the data byte that follows it. Some

examples are shown in **Figure 7.2(b)**. In all cases, the byte sequence delivered after de-stuffing is exactly the same as the original byte sequence. We can still search for a frame boundary by looking for two *flag bytes* in a row, without bothering to undo escapes. The byte-stuffing scheme depicted in **Fig. 7.2** is a slight simplification of the one used in **PPP** (Point-to-Point Protocol), which is used to carry packets over communications links.

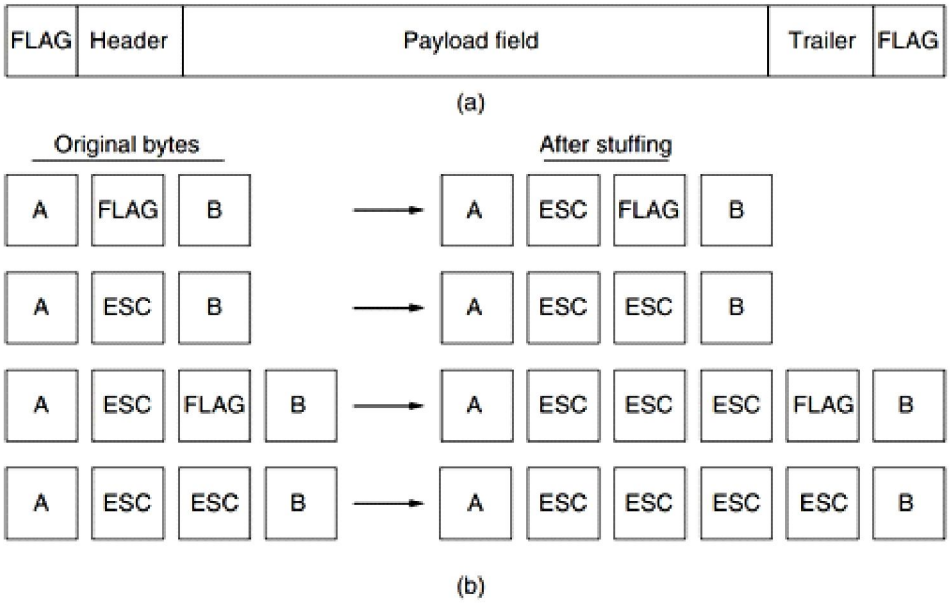


Figure 7.2: (a) A frame delimited by flag bytes. (b) Four examples of byte sequences before and after byte stuffing.

The third method of delimiting the bit stream gets around a disadvantage of byte stuffing, which is that it is tied to the use of 8-bit bytes. Framing can also be done at the bit level, so frames can contain an arbitrary number of bits made up of units of any size. It was developed for the once very popular **HDLC** (High-level Data Link Control) protocol. Each frame begins and ends with a special bit pattern, 01111110 or 0x7E in hexadecimal. This pattern is a *flag byte*. Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This bit stuffing is analogous to byte stuffing, in which an *escape byte* is stuffed into the outgoing character stream before a *flag byte* in the data. It also ensures a minimum density of transitions that help the physical layer maintain synchronization. Universal Serial Bus(USB) uses bit stuffing for this reason. When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically de-stuffs (deletes) the 0 bit. Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing. If the user data contain the flag pattern, 01111110, this flag

is transmitted as 011111010 but stored in the receiver's memory as 01111110. Figure 7.3 gives an example of bit stuffing. With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern. Thus, if the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data.

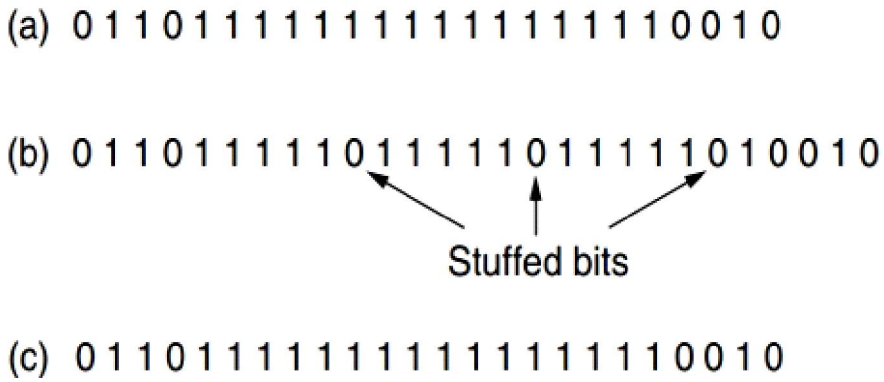


Figure 7.3: Bit stuffing. (a) Original data. (b) Data as appeares on the line. (c) Data as stored in the receiver's memory after de-stuffing

With both bit and byte stuffing, a side effect is that the length of a frame now depends on the contents of the data it carries. For instance, if there are no *flag bytes* in the data, 100 bytes might be carried in a frame of roughly 100 bytes. If, however, the data consists solely of *flag bytes*, each flag byte will be escaped and the frame will become roughly 200 bytes long. With bit stuffing, the increase would be roughly 12.5% as 1 bit is added to every byte. The last method of framing is to use a shortcut from the physical layer. Encoding of bits as signals often includes redundancy to help the receiver which means that some signals will not occur in regular data. For example, in the 4B/5B line code 4 data bits are mapped to 5 signal bits to ensure sufficient bit transitions. This means that 16 out of the 32 signal possibilities are not used. We can use some reserved signals to indicate the start and end of frames. In effect, we are using ''**coding violations**'' to delimit frames. The beauty of this scheme is that, because they are reserved signals, it is easy to find the start and end of frames and there is no need to stuff the data. Many data link protocols use a combination of these methods for safety. A common pattern used for Ethernet and 802.11 is to have a frame begin with a well-defined pattern called a **preamble**. This pattern might be quite long (72 bits is typical for 802.11) to allow the receiver to prepare for an incoming packet. The preamble is then followed by a length (i.e., count) field in the header that is used to locate the end of the frame.

7.5 ERROR CONTROL

Having solved the problem of marking the start and end of each frame, we come to the next problem: how to make sure that all frames are eventually delivered to the network layer at the destination and in the proper order. Assume for the moment that the receiver can tell whether a frame that it receives contains correct or faulty information. For unacknowledged connectionless service it might be fine if the sender just kept outputting frames without regard to whether they were arriving properly. But for reliable, connection-oriented service it would not be fine at all.

The usual way to ensure reliable delivery is to provide the sender with some feedback about what is happening at the other end of the line. Typically, the protocol calls for the receiver to send back special control frames bearing positive or negative acknowledgements about the incoming frames. If the sender receives a positive acknowledgement about a frame, it knows the frame has arrived safely. On the other hand, a negative acknowledgement means that something has gone wrong and the frame must be transmitted again.

An additional complication comes from the possibility that hardware troubles may cause a frame to vanish completely. In this case, the receiver will not react at all, since it has no reason to react. Similarly, if the acknowledgement frame is lost, the sender will not know how to proceed. It should be clear that a protocol in which the sender transmits a frame and then waits for an acknowledgement, positive or negative, will hang forever if a frame is ever lost due to, for example, malfunctioning hardware or a faulty communication channel. This possibility is dealt with by introducing timers into the data link layer. When the sender transmits a frame, it generally also starts a timer. The timer is set to expire after an interval long enough for the frame to reach the destination, be processed there, and have the acknowledgement propagate back to the sender. Normally, the frame will be correctly received and the acknowledgement will get back before the timer runs out, in which case the timer will be canceled. However, if either the frame or the acknowledgement is lost, the timer will go off, alerting the sender to a potential problem. The obvious solution is to just transmit the frame again. However, when frames may be transmitted multiple times there is a danger that the receiver will accept the same frame two or more times and pass it to the network layer more than once. To prevent this from happening, it is

generally necessary to assign sequence numbers to outgoing frames, so that the receiver can distinguish retransmissions from originals.

The whole issue of managing the timers and sequence numbers so as to ensure that each frame is ultimately passed to the network layer at the destination exactly once, no more and no less, is an important part of the duties of the data link layer (and higher layers).

7.6 FLOW CONTROL

Another important design issue that occurs in the data link layer is what to do with a sender that systematically wants to transmit frames faster than the receiver can accept them. This situation can occur when the sender is running on a fast, powerful computer and the receiver is running on a slow, low-end machine. Even if the transmission is error free, the receiver may be unable to handle the frames as fast as they arrive and will lose some. Clearly, something has to be done to prevent this situation. Two approaches are commonly used. In the first one, **feedback-based flow control**, the receiver sends back information to the sender giving it permission to send more data, or at least telling the sender how the receiver is doing. In the second one, **rate-based flow control**, the protocol has a built-in mechanism that limits the rate at which senders may transmit data, without using feedback from the receiver. Feedback-based schemes are seen at both the link layer and higher layers. The latter is more common these days, in which case the link layer hardware is designed to run fast enough that it does not cause loss. For example, hardware implementations of the link layer as NICs (Network Interface Cards) are sometimes said to run at ‘‘wire speed,’’ meaning that they can handle frames as fast as they can arrive on the link. Any overruns are then not a link problem, so they are handled by higher layers. Various feedback-based flow control schemes are known, but most of them use the same basic principle. The protocol contains well-defined rules about when a sender may transmit the next frame. These rules often prohibit frames from being sent until the receiver has granted permission, either implicitly or explicitly

7.7 ERROR DETECTION AND CORRECTION

Communication channels have a range of characteristics. Some channels, like optical fiber in telecommunications networks, have tiny error rates so that transmission errors are a rare occurrence. But other channels, especially wireless links and aging local loops, have error rates that are

orders of magnitude larger. For these links, transmission errors are the norm. They cannot be avoided at a reasonable expense or cost in terms of performance. The conclusion is that transmission errors are here to stay. We have to learn how to deal with them.

Network designers have developed two basic strategies for dealing with errors. Both add redundant information to the data that is sent. One strategy is to include enough redundant information to enable the receiver to deduce what the transmitted data must have been. The other is to include only enough redundancy to allow the receiver to deduce that an error has occurred and have it request a retransmission. The former strategy uses **error-correcting codes** and the latter uses **error-detecting codes**. The use of error-correcting codes is often referred to as **FEC** (Forward Error Correction).

7.7.1 Error Correcting Codes

There are four different error-correcting codes. These are listed below.

1. Hamming codes.
2. Binary convolutional codes.
3. Reed-Solomon codes.
4. Low-Density Parity Check codes.

All of these codes add redundancy to the information that is sent. A frame consists of ***m*** data (i.e., message) bits and ***r*** redundant (i.e. check) bits. In a block code, the ***r*** check bits are computed solely as a function of the ***m*** data bits with which they are associated, as though the ***m*** bits were looked up in a large table to find their corresponding ***r*** check bits. In a systematic code, the ***m*** data bits are sent directly, along with the check bits, rather than being encoded themselves before they are sent. In a linear code, the ***r*** check bits are computed as a linear function of the ***m*** data bits. Exclusive OR (XOR) or modulo 2 addition is a popular choice. This means that encoding can be done with operations such as matrix multiplications or simple logic circuits. The codes we will look at in this section are linear, systematic block codes unless otherwise noted.

Let the total length of a block be ***n*** (i.e., ***n=m+r***). We will describe this as an (***n***, ***m***) code. An ***n***-bit unit containing data and check bits is referred to as an ***n***-bit code word. The code rate, or

simply rate, is the fraction of the code word that carries information that is not redundant, or ***m/n***. The rates used in practice vary widely. They might be 1/2 for a noisy channel, in which case half of the received information is redundant, or close to 1 for a high-quality channel, with only a small number of check bits added to a large message. To understand how errors can be handled, it is necessary to first look closely at what an error really is. Given any two code words that may be transmitted or received—say, 10001001 and 10110001—it is possible to determine how many corresponding bits differ. In this case, 3 bits differ. To determine how many bits differ, just XOR the two code words and count the number of 1 bits in the result.

For example:

10001001
10110001
00111000

The number of bit positions in which two code words differ is called the **Hamming distance** (Hamming, 1950). Its significance is that if two code words are a Hamming distance ***d*** apart, it will require ***d*** single-bit errors to convert one into the other. Given the algorithm for computing the check bits, it is possible to construct a complete list of the legal code words, and from this list to find the two code words with the smallest Hamming distance. This distance is the Hamming distance of the complete code. In most data transmission applications, all 2^m possible data messages are legal, but due to the way the check bits are computed, not all of the 2^n possible code words are used. In fact, when there are ***r*** check bits, only the small fraction of $2^m/2^n$ or $1/2^r$ of the possible messages will be legal code-words. It is the sparseness with which the message is embedded in the space of code-words that allows the receiver to detect and correct errors. The error-detecting and error-correcting properties of a block code depend on its **Hamming distance**. To reliably detect ***d*** errors, you need a distance ***d*+1** code because with such a code there is no way that ***d*** single-bit errors can change a valid code word into another valid code word. When the receiver sees an illegal code-word, it can tell that a transmission error has occurred. Similarly, to correct ***d*** errors, you need a distance **2*d*+1** code because that way

the legal code-words are so far apart that even with ***d*** changes the original code-word is still closer than any other code-word. This means the original code-word can be uniquely determined based on the assumption that a larger number of errors are less likely.

As a simple example of an error-correcting code, consider a code with only four valid code-words:

0000000000, 0000011111, 1111100000, and 1111111111

This code has a distance of 5, which means that it can correct double errors or detect quadruple errors. If the code-word 0000000111 arrives and we expect only single- or double-bit errors, the receiver will know that the original must have been 0000011111. If, however, a triple error changes 0000000000 into 0000000111, the error will not be corrected properly. Alternatively, if we expect all of these errors, we can detect them. None of the received code-words are legal code-words so an error must have occurred. It should be apparent that in this example we cannot both correct double errors and detect quadruple errors because this would require us to interpret a received code-word in two different ways. In our example, the task of decoding by finding the legal code-word that is closest to the received code-word can be done by inspection. Unfortunately, in the most general case where all code-words need to be evaluated as candidates, this task can be a time-consuming search. Instead, practical codes are designed so that they admit shortcuts to find what was likely the original code-word. Imagine that we want to design a code with ***m*** message bits and ***r*** check bits that will allow all single errors to be corrected. Each of the $\mathbf{2^m}$ legal messages has ***n*** illegal code words at a distance of 1 from it. These are formed by systematically inverting each of the ***n*** bits in the ***n***-bit code-word formed from it. Thus, each of the $\mathbf{2^m}$ legal messages requires ***n*+1** bit patterns dedicated to it. Since the total number of bit patterns is $\mathbf{2^n}$, we must have $\mathbf{(n+1)2^m \leq 2^n}$. Using ***n=m+r***, this requirement becomes

$$\mathbf{(m+r+1) \leq 2^r}$$

Given ***m***, this puts a lower limit on the number of check bits needed to correct single errors. This theoretical lower limit can, in fact, be achieved using a method due to Hamming (1950). In Hamming codes the bits of the code-word are numbered consecutively, starting with bit 1 at the left end, bit 2 to its immediate right, and so on. The bits that are powers of 2 (1, 2, 4, 8, 16, etc.) are check bits. The rest (3,

5, 6, 7, 9, etc.) are filled up with the ***m*** data bits. This pattern is shown for an (11,7) Hamming code with 7 data bits and 4 check bits in **Fig. 7.4**. Each check bit forces the modulo 2 sum, or parity, of some collection of bits, including itself, to be even (or odd). A bit may be included in several check bit computations. To see which check bits the data bit in position ***k*** contributes to, rewrite ***k*** as a sum of powers of 2. For example, 11 = 1 + 2 + 8 and 29 = 1 + 4 + 8 + 16. A bit is checked by just those check bits occurring in its expansion (e.g., bit 11 is checked by bits 1, 2, and 8). In the example, the check bits are computed for even parity sums for a message that is the ASCII letter "**A**".

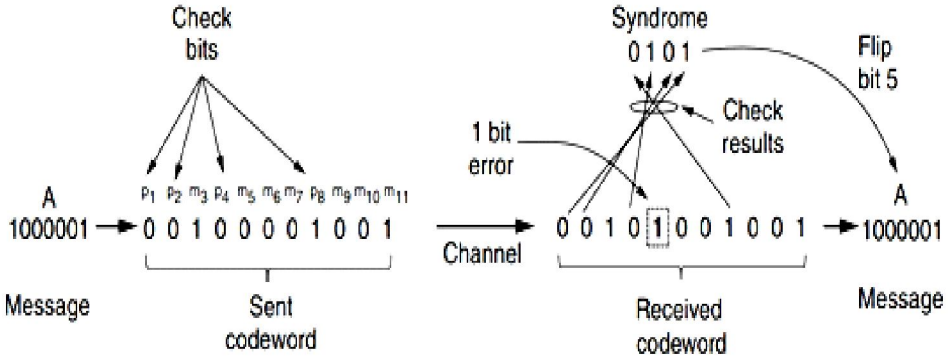


Figure 7.4: Example of an (11, 7) Hamming code correcting a single-bit error.

This construction gives a code with a Hamming distance of 3, which means that it can correct single errors (or detect double errors). The reason for the very careful numbering of message and check bits becomes apparent in the decoding process. When a code-word arrives, the receiver redoes the check bit computations including the values of the received check bits. We call these the check results. If the check bits are correct then, for even parity sums, each check result should be zero. In this case the code-word is accepted as valid. If the check results are not all zero, however, an error has been detected. The set of check results forms the **error syndrome** that is used to pinpoint and correct the error. In **Fig. 7.4**, a single-bit error occurred on the channel so the check results are 0, 1, 0, and 1 fork = 8, 4, 2, and 1, respectively. This gives a syndrome of 0101 or 4+1=5. By the design of the scheme, this means that the fifth bit is in error. Flipping the incorrect bit (which might be a check bit or a

data bit) and discarding the check bits gives the correct message of an ASCII "**A**". Hamming distances are valuable for understanding block codes, and Hamming codes are used in error-correcting memory. However, most networks use stronger codes. The second code we will look at is a **convolutional** code. This code is the only one we will cover that is not a block code. In a convolutional code, an encoder processes a sequence of input bits and generates a sequence of output bits. There is no natural message size or encoding boundary as in a block code. The output depends on the current and previous input bits. That is, the encoder has memory. The number of previous bits on which the output depends is called the constraint length of the code. Convolutional codes are specified in terms of their rate and constraint length. Convolutional codes are widely used in deployed networks, for example, as part of the GSM mobile phone system, in satellite communications, and in 802.11. As an example, a popular convolutional code is shown in **Fig. 7.5**. This code is known as the NASA convolutional code of **r=1/2** and **k=7**, since it was first used for the Voyager space missions starting in 1977. Since then it has been liberally reused, for example, as part of 802.11

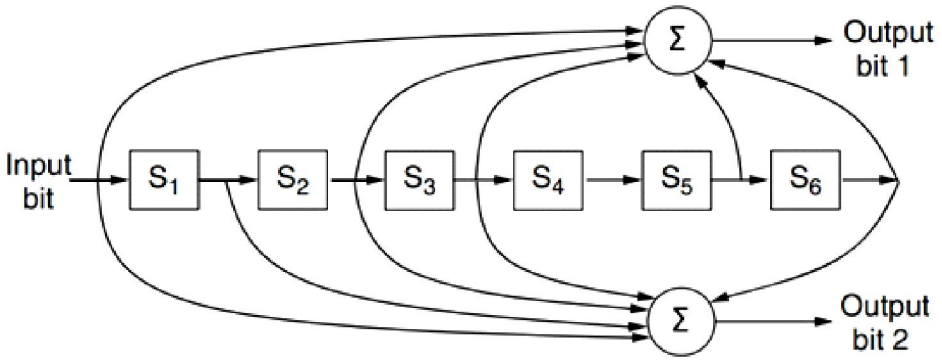


Figure 7.5: The NASA binary convolutional code used in 802.11

In **Fig. 7.5**, each input bit on the left-hand side produces two output bits on the right-hand side that are XOR sums of the input and internal state. Since it deals with bits and performs linear operations, this is a binary, linear convolutional code. Since 1 input bit produces 2 output bits, the code rate is 1/2. It is not systematic since none of the output bits is simply the input bit. The internal state is kept in six memory registers. Each time another bit is input the values in the registers are shifted to the right. For example, if 111 is input and the initial

state is all zeros, the internal state, written left to right, will become 100000, 110000, and 111000 after the first, second, and third bits have been input. The output bits will be 11, followed by 10, and then 01. It takes seven shifts to flush an input completely so that it does not affect the output. The constraint length of this code is thus **k=7**. A convolutional code is decoded by finding the sequence of input bits that is most likely to have produced the observed sequence of output bits (which includes any errors). For small values of ***k***, this is done with a widely used algorithm developed by Viterbi (Forney, 1973). The algorithm walks the observed sequence, keeping for each step and for each possible internal state the input sequence that would have produced the observed sequence with the fewest errors. The input sequence requiring the fewest errors at the end is the most likely message. Convolutional codes have been popular in practice because it is easy to factor the uncertainty of a bit being a 0 or a 1 into the decoding. For example, suppose **"1V** is the logical 0 level and **+1V** is the logical 1 level, we might receive 0.9V and"0.1V for 2 bits. Instead of mapping these signals to 1 and 0 right away, we would like to treat 0.9V as ''very likely a 1'' and "0.1V as ''maybe a 0'' and correct the sequence as a whole. Extensions of the Viterbi algorithm can work with these uncertainties to provide stronger error correction. This approach of working with the uncertainty of a bit is called **soft-decision decoding**. Conversely, deciding whether each bit is a 0 or a 1 before subsequent error correction is called **hard-decision decoding**.

The third kind of error-correcting code we will describe is the **Reed-Solomon code**. Like Hamming codes, Reed-Solomon codes are linear block codes, and they are often systematic too. Unlike Hamming codes, which operate on individual bits, Reed-Solomon codes operate on ***m*** bit symbols. Naturally, the mathematics are more involved, so we will describe their operation by analogy. Reed-Solomon codes are based on the fact that every ***n*** degree polynomial is uniquely determined by **n+1** points. For example, a line having the form **ax+b** is determined by two points. Extra points on the same line are redundant, which is helpful for error correction. Imagine that we have two data points that represent a line and we send those two data points plus two check points chosen to lie on the same line. If one of the points is received in error, we can still recover the

data points by fitting a line to the received points. Three of the points will lie on the line, and one point, the one in error, will not. By finding the line we have corrected the error. Reed-Solomon codes are actually defined as polynomials that operate over finite fields, but they work in a similar manner. For ***m*** bit symbols, the code-words are $\mathbf{2^m-1}$ symbols long. A popular choice is to make **m=8** so that symbols are bytes. A code-word is then 255 bytes long. The (255, 233) code is widely used; it adds 32 redundant symbols to 233 data symbols. Decoding with error correction is done with an algorithm developed by Berlekamp and Massey that can efficiently perform the fitting task for moderate-length codes (Massey, 1969). Reed-Solomon codes are widely used in practice because of their strong error-correction properties, particularly for burst errors. They are used for DSL, data over cable, satellite communications, and perhaps most ubiquitously on CDs, DVDs, and Blu-ray discs. Because they are based on ***m*** bit symbols, a single-bit error and an ***m*-bit** burst error are both treated simply as one symbol error. When ***2t*** redundant symbols are added, a Reed-Solomon code is able to correct up to ***t*** errors in any of the transmitted symbols. This means, for example, that the (255, 233) code, which has 32 redundant symbols, can correct up to 16 symbol errors. Since the symbols may be consecutive and they are each 8 bits, an error burst of up to 128 bits can be corrected. The situation is even better if the error model is one of erasures (e.g., a scratch on a CD that obliterates some symbols). In this case, up to ***2t*** errors can be corrected. Reed-Solomon codes are often used in combination with other codes such as a convolutional code. The thinking is as follows. Convolutional codes are effective at handling isolated bit errors, but they will fail, likely with a burst of errors, if there are too many errors in the received bit stream. By adding a Reed-Solomon

code within the convolutional code, the Reed-Solomon decoding can mop up the error bursts, a task at which it is very good. The overall code then provides good protection against both single and burst errors.

The final error-correcting code we will cover is the **LDPC** (**Low-Density**
Parity Check) code. LDPC codes are linear block codes that were

invented by Robert Gallagher in his doctoral thesis (Gallagher, 1962). Like most theses, they were promptly forgotten, only to be reinvented in 1995 when advances in computing power had made them practical. In an LDPC code, each output bit is formed from only a fraction of the input bits. This leads to a matrix representation of the code that has a low density of 1s, hence the name for the code. The received code-words are decoded with an approximation algorithm that iteratively improves on a best fit of the received data to a legal code-word. This corrects errors. LDPC codes are practical for large block sizes and have excellent error-correction abilities that outperform many other codes in practice. For this reason they are rapidly being included in new protocols. They are part of the standard for digital video broadcasting, 10 Gbps Ethernet, power-line networks, and the latest version of 802.11.

7.7.2 Error Detecting Codes

Error-correcting codes are widely used on wireless links, which are notoriously noisy and error prone when compared to optical fibers. Without error-correcting codes, it would be hard to get anything through. However, over fiber or high-quality copper, the error rate is much lower, so error detection and retransmission is usually more efficient therefor dealing with the occasional error.

We will examine three different error-detecting codes. They are all linear, systematic block codes:

1. Parity.
2. Checksums.
3. Cyclic Redundancy Checks (CRCs).

To see how they can be more efficient than error-correcting codes, consider the first error-detecting code, in which a single parity bit is appended to the data. The parity bit is chosen so that the number of 1 bits in the code-word is even (or odd). Doing this is equivalent to computing the (even) parity bit as the modulo 2 sum or XOR of the data bits. For example, when 1011010 is sent in even parity, a bit is added to the end to make it 10110100. With odd parity 1011010 becomes 10110101. A code with a single parity bit has a distance of 2, since any single-bit error produces a code-word with the wrong parity. This means that it can detect single-bit errors. Consider a

channel on which errors are isolated and the error rate is 10?6 per bit. This may seem a tiny error rate, but it is at best a fair rate for a long wired cable that is challenging for error detection. Typical LAN links provide bit error rates of 10?10. Let the block size be 1000 bits. To provide error correction for 1000-bit blocks, we know from Equation 7.1 that 10 check bits are needed. Thus, a megabit of data would require 10,000 check bits. To merely detect a block with a single 1-bit error, one parity bit per block will suffice. Once every 1000 blocks, a block will be found to be in error and an extra block (1001 bits) will have to be transmitted to repair the error. The total overhead for the error detection and re-transmission method is only 2001 bits per megabit of data, versus 10,000 bits for a Hamming code. One difficulty with this scheme is that a single parity bit can only reliably detect a single-bit error in the block. If the block is badly garbled by a long burst error, the probability that the error will be detected is only 0.5, which is hardly acceptable. The odds can be improved considerably if each block to be sent is regarded as a rectangular matrix n bits wide and k bits high. Now, if we compute and send one parity bit for each row, up to k bit errors will be reliably detected as long as there is at most one error per row. However, there is something else we can do that provides better protection against burst errors: we can compute the parity bits over the data in a different order than the order in which the data bits are transmitted. Doing so is called interleaving. In this case, we will compute a parity bit for each of the n columns and send all the data bits as k rows, sending the rows from top to bottom and the bits in each row from left to right in the usual manner. At the last row, we send the n parity bits. This transmission order is shown in Fig. 7.6 for n=7 and k=7.

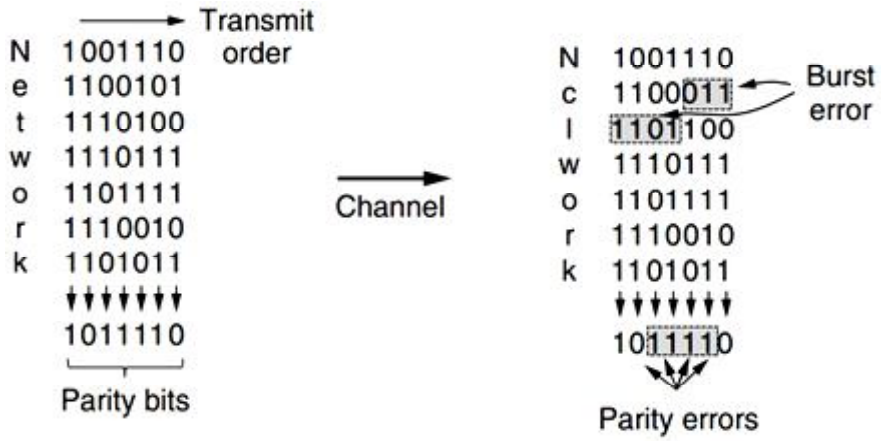


Figure 7.6: Interleaving of parity bits to detect a burst error

Interleaving is a general technique to convert a code that detects (or corrects) isolated errors into a code that detects (or corrects) burst errors. In Fig. 7.6, when a burst error of length n=7 occurs, the bits that are in error are spread across different columns. (A burst error does not imply that all the bits are wrong; it just implies that at least the first and last are wrong. In Fig. 7.6, 4 bits were flipped over a range of 7 bits.) At most 1 bit in each of the n columns will be affected, so the parity bits on those columns will detect the error. This method uses n parity bits on blocks of kn data bits to detect a single burst error of length n or less. A burst of length n+1 will pass undetected, however, if the first bit is inverted, the last bit is inverted, and all the other bits are correct. If the block is badly garbled by a long burst or by multiple shorter bursts, the probability that any of the n columns will have the correct parity by accident is 0.5, so the probability of a bad block being accepted when it should not be is 2?n.

The second kind of error-detecting code, the checksum, is closely related to groups of parity bits. The word ''checksum'' is often used to mean a group of check bits associated with a message, regardless of how are calculated. A group of parity bits is one example of a checksum. However, there are other, stronger checksums based on a running sum of the data bits of the message. The checksum is usually placed at the end of the message, as the complement of the sum function. This way, errors may be detected by summing the entire received code-word, both data bits and checksum. If the result comes out to be zero, no error has been detected. One example of a checksum is the 16-bit Internet checksum used on all Internet packets as part of the IP protocol (Braden et al., 1988). This checksum is a sum of the message bits divided into16-bit words. Because this method operates on words rather than on bits, as in parity, errors that leave the parity unchanged can still alter the sum and be detected. For example, if the lowest order bit in two different words is flipped from a 0 to a 1, a parity check across these bits would fail to detect an error. However, two 1s will be added to the 16-bit checksum to produce a different result. The error can then be detected. The Internet checksum is computed in one's complement arithmetic instead of as the modulo 216 sum. In one's complement arithmetic, a negative number is the bitwise complement of its positive counterpart. Modern computers run two's complement

arithmetic, in which a negative number is the one's complement plus one. On a two's complement computer, the one's complement sum is equivalent to taking the sum modulo 216 and adding any overflow of the high order bits back into the low-order bits. This algorithm gives a more uniform coverage of the data by the checksum bits. Otherwise, two high-order bits can be added, overflow, and be lost without changing the sum. There is another benefit, too. One's complement has two representations of zero, all 0s and all 1s. This allows one value (e.g., all 0s) to indicate that there is no checksum, without the need for another field. For decades, it has always been assumed that frames to be check-summed contain random bits. All analyses of checksum algorithms have been made under this assumption. Inspection of real data by Partridge et al. (1995) has shown this assumption to be quite wrong. As a consequence, undetected errors are in some cases much more common than had been previously thought. The Internet checksum in particular is efficient and simple but provides weak protection in some cases precisely because it is a simple sum. It does not detect the deletion or addition of zero data, nor swapping parts of the message, and it provides weak protection against message splices in which parts of two packets are put together. These errors may seem very unlikely to occur by random processes, but they are just the sort of errors that can occur with buggy hardware. A better choice is Fletcher's checksum (Fletcher, 1982). It includes a positional component, adding the product of the data and its position to the running sum. This provides stronger detection of changes in the position of data.

Although the two preceding schemes may sometimes be adequate at higher layers, in practice, a third and stronger kind of error-detecting code is in wide-spread use at the link layer: the **CRC (Cyclic Redundancy Check)**, also known as a **polynomial code**. Polynomial codes are based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 only. A **k-bit** frame is regarded as the coefficient list for a polynomial with ***k*** terms, ranging from $\mathbf{x^{k-1}}$ to $\mathbf{x^0}$. Such a polynomial is said to be of degree **k-1**. The high-order (leftmost) bit is the coefficient of $\mathbf{x^{k-1}}$, the next bit is the coefficient of $\mathbf{x^{k-2}}$, and so on. For example, 110001 has 6 bits and thus represents a six-term polynomial with coefficients 1,

1, 0, 0, 0, and 1: $\mathbf{1}x^5+\mathbf{1}x^4+\mathbf{0}x^3+\mathbf{0}x^2+\mathbf{0}x^1+\mathbf{1}x^0$. Polynomial arithmetic is done modulo 2, according to the rules of algebraic field theory. It does not have carries for addition or borrows for subtraction. Both addition and subtraction are identical to exclusive OR. For example:

10011011	00110011	11110000	01010101
+ 11001010	+ 11001101	− 10100110	− 10101111
01010001	11111110	01010110	11111010

Long division is carried out in exactly the same way as it is in binary except that the subtraction is again done modulo 2. A divisor is said "to go into" a dividend if the dividend has as many bits as the divisor. When the polynomial code method is employed, the sender and receiver must agree upon a generator polynomial, G(x), in advance. Both the high- and low-order bits of the generator must be 1. To compute the CRC for some frame with m bits corresponding to the polynomial M(x), the frame must be longer than the generator polynomial. The idea is to append a CRC to the end of the frame in such a way that the polynomial represented by the check-summed frame is divisible by G(x). When the receiver gets the check-summed frame, it tries dividing it by G(x). If there is a remainder, there has been a transmission error.

The algorithm for computing the CRC is as follows:

1. Let r be the degree of **G(*x*)**. Append r zero bits to the low-order end of the frame so it now contains ***m+r*** bits and corresponds to the polynomial $x^rM(x)$.

2. Divide the bit string corresponding to **G(x)** into the bit string corresponding to $x^rM(x)$, using modulo 2 division.

3. Subtract the remainder (which is always r or fewer bits) from the bit string corresponding to x^rM**(x)** using modulo 2 subtraction. The result is the check-summed frame to be transmitted. Call its polynomial **T(x)**.

Figure 7.7 illustrates the calculation for a frame 1101011111 using the generator $\mathbf{G(x)=x^4+x+1}$. It should be clear that **T(x)** is divisible (modulo 2) by **G(x)**. In any division problem, if you diminish the dividend by the remainder, what is left over is divisible by the divisor.

For example, in base 10, if you divide 210,278 by 10,941, the remainder is 2399. If you then subtract 2399 from 210,278, what is left over (207,879) is divisible by 10,941. Now let us analyze the power of this method. What kinds of errors will be detected? Imagine that a transmission error occurs, so that instead of the bit string for **T(x)** arriving, **T(x)+E(x)** arrives. Each 1 bit in **E(x)** corresponds to a bit that has been inverted. If there are k 1 bits in **E(x)**, ***k*** single-bit errors have occurred. A single burst error is characterized by an initial 1, a mixture of 0s and 1s, and a final 1, with all other bits being 0. Upon receiving the check-summed frame, the receiver divides it by **G(x)**; that is, it computes **[T(x)+E(x)]/G(x)**. **T(x)/G(x)** is 0, so the result of the computation is simply **E(x)/G(x)**. Those errors that happen to correspond to polynomials containing **G(x)** as a factor will slip by; all other errors will be caught. If there has been a single-bit error, **E(x)=xi**, where **i** determines which bit is in error. If **G(x)** contains two or more terms, it will never divide into **E(x)**, so all single-bit errors will be detected.

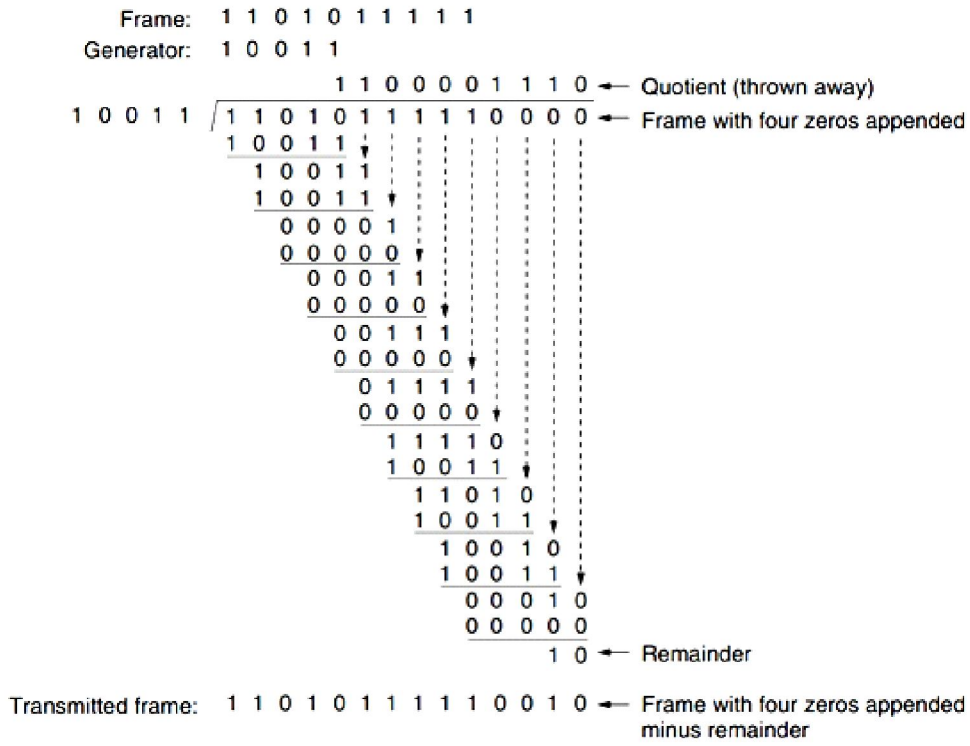


Figure 7.7: Example calculation of the CRC

If there have been two isolated single-bit errors, $E(x)=x^i+x^j$, where

i>j. Alternatively, this can be written as $\mathbf{E(x)=x^j(x^{i-j}+1)}$. If we assume that **G(x)** is not divisible by **x**, a sufficient condition for all double errors to be detected is that **G(x)** does not divide $\mathbf{x^{k+1}}$ for any k up to the maximum value of **i-j** (i.e., up to the maximum frame length). Simple, low-degree polynomials that give protection to long frames are known. For example, $\mathbf{x^{15}+x^{14}+1}$ will not divide $\mathbf{x^{k+1}}$ for any value of **k** below 32,768. If there are an odd number of bits in error, **E(X)** contains an odd number of terms (e.g., x5+x2+1, but not x2+1). Interestingly, no polynomial with an odd number of terms has **x+1** as a factor in the modulo 2 system. By making **x+1** a factor of **G(x)**, we can catch all errors with an odd number of inverted bits.

Finally, and importantly, a polynomial code with r check bits will detect all burst errors of length $\leq$**r**. A burst error of length ***k*** can be represented by $\mathbf{x^i(x^{k-1}+...+1)}$, where **i** determines how far from the right-hand end of the received frame the burst is located. If **G(x)** contains an $\mathbf{x^0}$ term, it will not have xi as a factor, so if the degree of the parenthesized expression is less than the degree of **G(x)**, the remainder can never be zero. If the burst length is **r+1**, the remainder of the division by **G(x)** will be zero if and only if the burst is identical to **G(x)**. By definition of a burst, the first and last bits must be 1, so whether it matches depends on the **r-1** intermediate bits. If all combinations are regarded as equally likely, the probability of such an incorrect frame being accepted as valid is $½^{r-1}$. It can also be shown that when an error burst longer than r+1 bits occurs or when several shorter bursts occur, the probability of a bad frame getting through unnoticed is ½r, assuming that all bit patterns are equally likely.

Certain polynomials have become international standards. The one used in IEEE 802 followed the example of Ethernet and is

$$\mathbf{x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+1}$$

Among other desirable properties, it has the property that it detects all bursts of length 32 or less and all bursts affecting an odd number of bits. It has been used widely since the 1980s. However, this does not mean it is the best choice. Using an exhaustive computational search, Castagnoli et al. (1993) and Koopman (2002) found the best CRCs. These CRCs have a Hamming distance of 6 for typical

message sizes, while the IEEE standard CRC-32 has a Hamming distance of only 4. Although the calculation required to compute the CRC may seem complicated, it is easy to compute and verify CRCs in hardware with simple shift register circuits (Peterson and Brown, 1961). In practice, this hardware is nearly always used. Dozens of networking standards include various CRCs, including virtually all LANs (e.g., Ethernet, 802.11) and point-to-point links (e.g., packets over SONET).



CHECK YOUR PROGRESS

1. Fill in the blanks:

(a) The _____ layer is the protocol layer that transfers data between adjacent network nodes in a wide area network.

(b) The _____ flow control protocol has a built-in mechanism that limits data transmission rate.

(c) _____ codes include enough redundant information to enable the receiver to deduce what the transmitted data must have been.

(d) _____ codes include only enough redundancy to allow the receiver to deduce that an error has occurred and have it request a retransmission.

(e) The error-detecting and error-correcting properties of a block code depend on its _____.

(f) In a _____, an encoder processes a sequence of input bits and generates a sequence of output bits.

(g) _____ are based on the fact that every n degree polynomial is determined by n+1 points.

(h) _____ are practical for large block sizes and have excellent error-correction abilities.

(i) The _____ is usually placed at the end of the message.

(j) _____ are based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 only.

7.8 DATA LINK PROTOCOLS

To introduce the subject of protocols, we will begin by looking at three protocols of increasing complexity. Before we look at the protocols, it is useful to make explicit some of the assumptions underlying the model of communication. To start with, we assume that the physical layer, data link layer, and network layer are independent processes that communicate by passing messages back and forth. A common implementation is shown in Figure 7.8. The physical layer process and some of the data link layer process run on dedicate hardware called a NIC (Network Interface Card). The rest of the link layer process and the network layer process run on the main CPU as part of the operating system, with the software for the link layer process often taking the form of a device driver. However, other implementations are also possible (e.g., three processes offloaded to dedicated hardware called a network accelerator, or three processes running on the main CPU on a software-defined ratio). Actually, the preferred implementation changes from decade to decade with technology trade-offs. In any event, treating the three layers as separate processes makes the discussion conceptually cleaner and also serves to emphasize the independence of the layers.

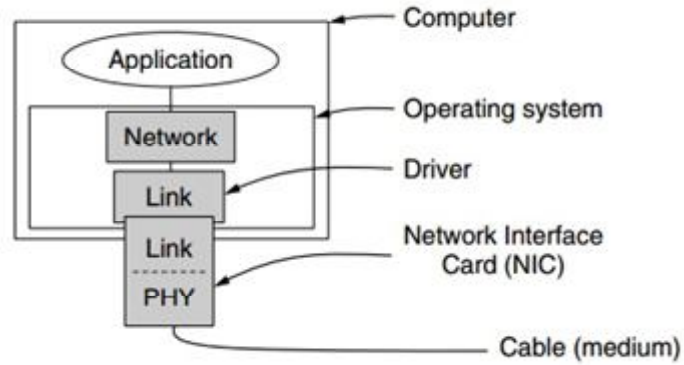


Figure 7.8: Implementation of the physical, data link, and network layers

Another key assumption is that machine A wants to send a long stream of data to machine B, using a reliable, connection-oriented service. A is assumed to have an infinite supply of data ready to send and never has to wait for data to be produced. Instead, when A's data link layer asks for data, the network layer is always able to comply immediately. We also assume that machines do not crash. That is, these protocols deal with communication errors, but not the problems caused by computers crashing and rebooting. As far as the data link layer is concerned, the packet passed

across the interface to it from the network layer is pure data, whose every bit is to be delivered to the destination's network layer. The fact that the destination's network layer may interpret part of the packet as a header is of no concern to the data link layer. When the data link layer accepts a packet, it encapsulates the packet in a frame by adding a data link header and trailer to it. Thus, a frame consists of an embedded packet, some control information (in the header), and a checksum (in the trailer). The frame is then transmitted to the data link layer on the other machine. We will assume that there exist suitable library procedures to physical layer to send a frame and from physical layer to receive a frame. These procedures compute and append or check the checksum (which is usually done in hardware) so that we do not need to worry about it as part of the protocols we develop in this section for example. Initially, the receiver has nothing to do. It just sits around waiting for something to happen. We indicate that the data link layer is waiting for something to happen by the procedure call ***wait_for_event (&event)***. This procedure only returns when something has happened (e.g., a frame has arrived). Upon return, the variable event tells what happened. The set of possible events differs for the various protocols to be described and will be defined separately for each protocol. Note that in a more realistic situation, the data link layer will not sit in a tight loop waiting for an event, as we have suggested, but will receive an interrupt, which will cause it to stop whatever it was doing and go handle the incoming frame. Nevertheless, for simplicity we will ignore all the details of parallel activity within the data link layer and assume that it is dedicated full time to handling just our one channel.

When a frame arrives at the receiver, the checksum is recomputed. If the checksum in the frame is incorrect (i.e., there was a transmission error), the data link layer is so informed ***(event=cksum err)***. If the inbound frame arrived undamaged, the data link layer is also informed ***(event=frame_arrival)*** so that it can acquire the frame for inspection using ***from_physical_layer***. As soon as the receiving data link layer has acquired an undamaged frame, it checks the control information in the header, and, if everything is all right, passes the packet portion to the network layer. Under no circumstances is a frame header ever given to a network layer. There is a good reason why the network layer must never be given any part of the frame header: to keep the network and data link protocols completely separate. As long as the network layer knows nothing at all about the data link protocol or the frame format, these things can be changed without

```
#define MAX_PKT 1024                                        /* determines packet size in bytes */

typedef enum {false, true} boolean;                         /* boolean type */
typedef unsigned int seq_nr;                                /* sequence or ack numbers */
typedef struct {unsigned char data[MAX_PKT];} packet;       /* packet definition */
typedef enum {data, ack, nak} frame_kind;                   /* frame_kind definition */

typedef struct {                                            /* frames are transported in this layer */
  frame_kind kind;                                          /* what kind of frame is it? */
  seq_nr seq;                                               /* sequence number */
  seq_nr ack;                                               /* acknowledgement number */
  packet info;                                              /* the network layer packet */
} frame;


/* Wait for an event to happen; return its type in event. */
void wait_for_event(event_type *event);

/* Fetch a packet from the network layer for transmission on the channel. */
void from_network_layer(packet *p);

/* Deliver information from an inbound frame to the network layer. */
void to_network_layer(packet *p);

/* Go get an inbound frame from the physical layer and copy it to r. */
void from_physical_layer(frame *r);

/* Pass the frame to the physical layer for transmission. */
void to_physical_layer(frame *s);

/* Start the clock running and enable the timeout event. */
void start_timer(seq_nr k);

/* Stop the clock and disable the timeout event. */
void stop_timer(seq_nr k);

/* Start an auxiliary timer and enable the ack_timeout event. */
void start_ack_timer(void);

/* Stop the auxiliary timer and disable the ack_timeout event. */
void stop_ack_timer(void);

/* Allow the network layer to cause a network_layer_ready event. */
void enable_network_layer(void);

/* Forbid the network layer from causing a network_layer_ready event. */
void disable_network_layer(void);

/* Macro inc is expanded in-line: increment k circularly. */
#define inc(k) if (k < MAX_SEQ) k = k + 1; else k = 0
```

Figure 7.9: Some definitions needed in the protocols to follow. These definitions are located in the file protocol.h

requiring changes to the network layer's software. This happens whenever a new NIC is installed in a computer. Providing a rigid interface between the network and data link layers greatly simplifies the design task because communication protocols in different layers can evolve independently. **Figure**

4.9 shows some declarations (in C) common to many of the protocols to be discussed later. Five data structures are defined there: ***boolean***, ***seq_nr,*** ***packet***, ***frame_kind,*** and ***frame***. A boolean is an enumerated type and can take on the values true and false. A seq_nr is a small integer used to number the frames so that we can tell them apart. These sequence numbers run from 0 up to and including *MAX_SEQ*, which is defined in each protocol needing it. A packet is the unit of information exchanged between the network layer and the data link layer on the same machine, or between network layer peers. In our model it always contains *MAX_PKT* bytes, but more realistically it would be of variable length. A frame is composed of four fields: ***kind, seq, ack,*** and ***info***, the first three of which contain control information and the last of which may contain actual data to be transferred. These control fields are collectively called the frame header. The kind field tells whether there are any data in the frame, because some of the protocols distinguish frames containing only control information from those containing data as well. The *seq* and *ack* fields are used for sequence numbers and acknowledgements, respectively. The info field of a data frame contains a single packet; the info field of a control frame is not used. A more realistic implementation would use a variable-length info field, omitting it altogether for control frames.

Again, it is important to understand the relationship between a packet and a frame. The network layer builds a packet by taking a message from the transport layer and adding the network layer header to it. This packet is passed to the data link layer for inclusion in the info field of an outgoing frame. When the frame arrives at the destination, the data link layer extracts the packet from the frame and passes the packet to the network layer. In this manner, the network layer can act as though machines can exchange packets directly. A number of procedures are also listed in **Figure 7.9**. These are library routines whose details are implementation dependent. The procedure *wait_for_event* sits in a tight loop waiting for something to happen. The procedures *to_network_layer* and *from_network_layer* are used by the data link layer to pass packets to the network layer and accept packets from the network layer, respectively. Note that from physical layer and to physical layer pass frames between the data link layer and the physical layer. In other words, *to_network_layer* and *from_network_layer* deal with the interface between layers 2 and 3, whereas from physical layer and to physical layer deal with the interface between layers 1 and 2.

In most of the protocols, we assume that the channel is unreliable and loses entire frames upon occasion. To be able to recover from such calamities, the sending data link layer must start an internal timer or clock whenever it sends a frame. If no reply has been received within a certain predetermined time interval, the clock times out and the datalink layer receives an interrupt signal. In our protocols this is handled by allowing the procedure *wait_for_event* to return *event=timeout*. The procedures *start_timer* and *stop_timer* turn the timer on and off, respectively. Timeout events are possible only when the timer is running and before *stop_timer* is called. It is explicitly permitted to call *start_timer* while the timer is running; such a call simply resets the clock to cause the next timeout after a full timer interval has elapsed (unless it is reset or turned off). The procedures *start_ack_timer* and *stop_ack_timer* control an auxiliary timer used to generate acknowledgements under certain conditions. The procedures *enable_network_layer* and *disable_network_layer* are used in the more sophisticated protocols, where we no longer assume that the network layer always has packets to send. When the data link layer enables the network layer, the network layer is then permitted to interrupt when it has a packet to be sent. We indicate this with *event = network_layer_ready*. When the network layer is disabled, it may not cause such events. By being careful about when it enables and disables its network layer, the data link layer can prevent the network layer from swamping it with packets for which it has no buffer space. Frame sequence numbers are always in the range 0 to *MAX_SEQ*, where *MAX_SEQ* is different for the different protocols. It is frequently necessary to advance a sequence number by 1 circularly (i.e., *MAX_SEQ* is followed by 0).

The macro inc performs this incrementing. It has been defined as a macro because it is used in-line within the critical path. The factor limiting network performance is often protocol processing, so defining simple operations like this as macros does not affect the readability of the code but does improve performance.

7.8.1 Stop and Wait ARQ

Now we will tackle the problem of preventing the sender from flooding the receiver with frames faster than the latter is able to process them. This situation can easily happen in practice so being able to prevent it is of great importance. The communication channel is still

assumed to be error free, however, and the data traffic is still simplex. One solution is to build the receiver to be powerful enough to process a continuous stream of back-to-back frames. It must have sufficient buffering and processing abilities to run at the line rate and must be able to pass the frames that are received to the network layer quickly enough. However, this is a worst-case solution. It requires dedicated hardware and can be wasteful of resources if the utilization of the link is mostly low. Moreover, it just shifts the problem of dealing with a sender that is too fast elsewhere; in this case to the network layer. A more general solution to this problem is to have the receiver provide feedback to the sender. After having passed a packet to its network layer, the receiver sends a little dummy frame back to the sender which, in effect, gives the sender permission to transmit the next frame. After having sent a frame, the sender is required by the protocol to bide its time until the little dummy (acknowledgement) frame arrives. This delay is a simple example of a flow control protocol. Protocols in which the sender sends one frame and then waits for an acknowledgement before proceeding are called stop-and-wait. Figure 7.10 gives an example of a simplex stop-and-wait protocol. Although data traffic in this example is simplex, going only from the sender to the receiver, frames do travel in both directions. Consequently, the communication channel between the two data link layers needs to be capable of bidirectional information transfer. However, this protocol entails a strict alternation of flow: first the sender sends a frame, then the receiver sends a frame, then the sender sends another frame, then the receiver sends another one, and so on. A half-duplex physical channel would suffice here. As in protocol 1, the sender starts out by fetching a packet from the network layer, using it to construct a frame, and sending it on its way. But now, unlike in protocol 1, the sender must wait until an acknowledgement frame arrives before looping back and fetching the next packet from the network layer. The sending data link layer need not even inspect the incoming frame as there is only one possibility. The incoming frame is always an acknowledgement. The only difference between receiver1 and receiver2 is that after delivering a packet to the network layer, receiver2 sends an acknowledgement frame back to the sender before entering the wait loop again. Because only the arrival of the frame back at the sender is important, not its contents, the receiver need not put any particular information in it.

```
/* Protocol 2 (Stop-and-wait) also provides for a one-directional flow of data from
   sender to receiver. The communication channel is once again assumed to be error
   free, as in protocol 1. However, this time the receiver has only a finite buffer
   capacity and a finite processing speed, so the protocol must explicitly prevent
   the sender from flooding the receiver with data faster than it can be handled. */

typedef enum {frame_arrival} event_type;
#include "protocol.h"

void sender2(void)
{
 frame s;                          /* buffer for an outbound frame */
 packet buffer;                    /* buffer for an outbound packet */
 event_type event;                 /* frame_arrival is the only possibility */

 while (true) {
     from_network_layer(&buffer);  /* go get something to send */
     s.info = buffer;              /* copy it into s for transmission */
     to_physical_layer(&s);        /* bye-bye little frame */
     wait_for_event(&event);       /* do not proceed until given the go ahead */
 }
}

void receiver2(void)
{
 frame r, s;                       /* buffers for frames */
 event_type event;                 /* frame_arrival is the only possibility */
 while (true) {
     wait_for_event(&event);       /* only possibility is frame_arrival */
     from_physical_layer(&r);      /* go get the inbound frame */
     to_network_layer(&r.info);    /* pass the data to the network layer */
     to_physical_layer(&s);        /* send a dummy frame to awaken sender */
 }
}
```

Figure 7.10: A simplex stop-and-wait protocol

7.8.2 Go Back N ARQ

Until now we have made the tacit assumption that the transmission time required for a frame to arrive at the receiver plus the transmission time for the acknowledgement to comeback is negligible. Sometimes this assumption is clearly false. In these situations the long round-trip time can have important implications

for the efficiency of the bandwidth utilization. As an example, consider a 50-kbps satellite channel with a 500-msec round-trip propagation delay. Let us imagine trying to use protocol 4 to send 1000-bit frames via the satellite. At t=0 the sender starts sending the first frame. At t=20 msec the frame has been completely sent. Not until t=270 msec has the frame fully arrived at the receiver, and not until t=520 msec has the acknowledgement arrived back at the sender, under the best of circumstances. This means that the sender was blocked 500/ 520 or 96% of the time. In other words, only 4% of the available bandwidth was used. Clearly, the combination of a long transit time, high bandwidth, and short frame length is disastrous in terms of efficiency.

The problem described here can be viewed as a consequence of the rule requiring a sender to wait for an acknowledgement before sending another frame. If we relax that restriction, much better efficiency can be achieved. Basically, the solution lies in allowing the sender to transmit up to ***w*** frames before blocking, instead of just 1. With a large enough choice of ***w*** the sender will be able to continuously transmit frames since the acknowledgements will arrive for previous frames before the window becomes full, preventing the sender from blocking. To find an appropriate value for w we need to know how many frames can fit inside the channel as they propagate from sender to receiver. This capacity is determined by the bandwidth in bits/sec multiplied by the one-way transit time, or the bandwidth-delay product of the link. We can divide this quantity by the number of bits in a frame to express it as a number of frames. Call this quantity BD. Then w should be set to ***2BD+1***. Twice the bandwidth-delay is the number of frames that can be outstanding if the sender continuously sends frames when the round-trip time to receive an acknowledgement is considered. The ''+1'' is because an acknowledgement frame will not be sent until after a complete frame is received.

For the example link with a bandwidth of 50 kbps and a one-way transit time of 250 msec, the bandwidth-delay product is 12.5 kbit or 12.5 frames of 1000 bits each. ***2BD+1*** is then 26 frames. Assume the sender begins sending frame 0 as before and sends a new frame every 20 msec. By the time it has finished sending 26 frames, at

t=520 msec, the acknowledgement for frame 0 will have just arrived. Thereafter, acknowledgements will arrive every 20 msec, so the sender will always get permission to continue just when it needs it. From then onwards, 25 or 26 unacknowledged frames will always be outstanding. Put in other terms, the sender's maximum window size is 26. For smaller window sizes, the utilization of the link will be less than 100% since the sender will be blocked sometimes. We can write the utilization as the fraction of time that the sender is not blocked:

link utilization w / (1+2BD)

This value is an upper-bound because it does not allow for any frame processing time and treats the acknowledgement frame as having zero length, since it is usually short. The equation shows the need for having a large window w whenever the bandwidth-delay product is large. If the delay is high, the sender will rapidly exhaust its window even for a moderate bandwidth, as in the satellite example. If the bandwidth is high, even for a moderate delay the sender will exhaust its window quickly unless it has a large window (e.g., a 1-Gbps link with 1-msec delay holds 1 megabit). With stop-and-wait for which **w=1**, if there is even one frame's worth of propagation delay the efficiency will be less than 50%. This technique of keeping multiple frames in flight is an example of pipelining. Pipelining frames over an unreliable communication channel raises some serious issues. First, what happens if a frame in the middle of a long stream is damaged or lost? Large numbers of succeeding frames will arrive at the receiver before the sender even finds out that anything is wrong. When a damaged frame arrives at the receiver, it obviously should be discarded, but what should the receiver do with all the correct frames following it? Remember that the receiving data link layer is obligated to hand packets to the network layer in sequence.

Two basic approaches are available for dealing with errors in the presence of pipelining, both of which are shown in **Figure 7.11**.

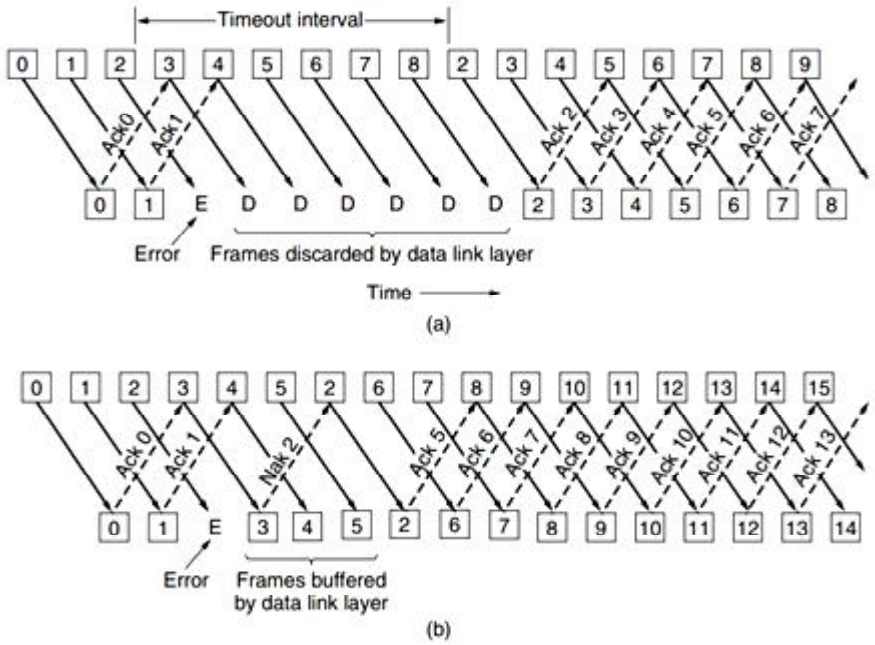


Figure 7.11: Pipelining and error recovery. Effect of an error when (a) receiver's window size is 1 and (b) receiver's window size is large

One option, called **Go-Back-N**, is for the receiver simply to discard all subsequent frames, sending no acknowledgements for the discarded frames. This strategy corresponds to a receive_window of size 1. In other words, the data link layer refuses to accept any frame except the next one it must give to the network layer. If the sender's window fills up before the timer runs out, the pipeline will begin to empty. Eventually, the sender will timeout and retransmit all unacknowledged frames in order, starting with the damaged or lost one. This approach can waste a lot of bandwidth if the error rate is high. In **Figure 7.11** (b) we see go-back-n for the case in which the receiver's window is large. Frames 0 and 1 are correctly received and acknowledged. Frame 2, however, is damaged or lost. The sender, unaware of this problem, continues to send frames until the timer for frame 2 expires. Then it backs up to frame 2 and starts over with it, sending 2, 3, 4, etc. all over again. The next strategy used is the Selective Repeat protocol and shall be discussed in the next section.

7.8.3 Selective Repeat ARQ

The go-back-n protocol works well if errors are rare, but if the line is poor it wastes a lot of bandwidth on retransmitted frames. An alternative strategy, the **selective repeat** protocol, is to allow the receiver to accept and buffer the frames following a damaged or lost one. In this protocol, both sender and receiver maintain a window of outstanding and acceptable sequence numbers, respectively. The sender's window size starts out at 0 and grows to some predefined maximum. The receiver's window, in contrast, is always fixed in size and equal to the predetermined maximum. The receiver has a buffer reserved for each sequence number within its fixed window. Associated with each buffer is a bit telling whether the buffer is full or empty. Whenever a frame arrives, its sequence number is checked by the function between to see if it falls within the window. If so and if it has not already been received, it is accepted and stored. This action is taken without regard to whether or not the frame contains the next packet expected by the network layer. Of course, it must be kept within the data link layer and not passed to the network layer until all the lower-numbered frames have already been delivered to the network layer in the correct order. Non-sequential receive introduces further constraints on frame sequence numbers compared to protocols in which frames are only accepted in order. We can illustrate the trouble most easily with an example. Suppose that we have a 3-bit sequence number, so that the sender is permitted to transmit up to seven frames before being required to wait for an acknowledgement. The sender now transmits frames 0 through 6. The receiver's window allows it to accept any frame with a sequence number between 0 and 6 inclusive. All seven frames arrive correctly, so the receiver acknowledges them and advances its window to allow receipt of 7, 0, 1, 2, 3, 4, or 5. All seven buffers are marked empty. It is at this point that disaster strikes in the form of a lightning bolt hitting the telephone pole and wiping out all the acknowledgements. The protocol should operate correctly despite this disaster. The sender eventually times out and re-transmits frame 0. When this frame arrives at the receiver, a check is made to see if it falls within the receiver's window. Unfortunately frame 0 is within the new window, so it is accepted as a new frame. The receiver

also sends a (piggybacked) acknowledgement for frame 6, since 0 through 6 have been received. The sender is happy to learn that all its transmitted frames did actually arrive correctly, so it advances its window and immediately sends frames 7, 0, 1, 2, 3, 4, and 5. Frame 7 will be accepted by the receiver and its packet will be passed directly to the network layer. Immediately thereafter, the receiving data link layer checks to see if it has a valid frame 0 already, discovers that it does, and passes the old buffered packet to the network layer as if it were a new packet. Consequently, the network layer gets an incorrect packet, and the protocol fails.

The essence of the problem is that after the receiver advanced its window, the new range of valid sequence numbers overlapped the old one. Consequently, the following batch of frames might be either duplicates (if all the acknowledgements were lost) or new ones (if all the acknowledgements were received). The receiver has no way of distinguishing these two cases. The way out of this dilemma lies in making sure that after the receiver has advanced its window there is no overlap with the original window. To ensure that there is no overlap, the maximum window size should be at most half the range of the sequence numbers.

7.8.4 HDLC Protocol

High-Level Data Link Control (HDLC) is a bit-oriented code-transparent synchronous data link layer protocol developed by the International Organization for Standardization (ISO). HDLC provides both connection-oriented and connectionless service. HDLC can be used for point to multipoint connections, but is now used almost exclusively to connect one device to another, using what is known as **Asynchronous Balanced Mode** (ABM). HDLC frames can be transmitted over synchronous or asynchronous links. Those links have no mechanism to mark the beginning or end of a frame, so the beginning and end of each frame has to be identified. This is done by using a frame delimiter, or flag, which is a unique sequence of bits that is guaranteed not to be seen inside a frame. This sequence is '01111110', or, in hexadecimal notation, 0x7E. Each frame begins and ends with a frame delimiter. A frame delimiter at the end of a frame may also mark the start of the next frame. A sequence of 7 or

more consecutive 1-bits within a frame will cause the frame to be aborted.

The contents of an HDLC frame are shown in the following table:

Flag	Address	Control	Information	FCS	Flag
8 bits	8 or more bits	8 or 16 bits	Variable length, 0 or more bits	16 or 32 bits	8 bits

Note that the end flag of one frame may be (but does not have to be) the beginning (start) flag of the next frame. Data is usually sent in multiples of 8 bits, but only some variants require this; others theoretically permit data alignments on other than 8-bit boundaries. The **frame check sequence** (FCS) is a 16-bit CRC-CCITT or a 32-bit CRC-32 computed over the Address, Control, and Information fields. It provides a means by which the receiver can detect errors that may have been induced during the transmission of the frame, such as lost bits, flipped bits, and extraneous bits. However, given that the algorithms used to calculate the FCS are such that the probability of certain types of transmission errors going undetected increases with the length of the data being checked for errors, the FCS can implicitly limit the practical size of the frame. If the receiver's calculation of the FCS does not match that of the sender's, indicating that the frame contains errors, the receiver can either send a negative acknowledge packet to the sender, or send nothing. After either receiving a negative acknowledge packet or timing out waiting for a positive acknowledge packet, the sender can retransmit the failed frame. The FCS was implemented because many early communication links had a relatively high bit error rate, and the FCS could readily be computed by simple, fast circuitry or software. More effective forward error correction schemes are now widely used by other protocols

There are three fundamental types of HDLC frames.

- Information frames, or I-frames, transport user data from the network layer. In addition they can also include flow and error control information piggybacked on data.
- Supervisory Frames, or S-frames, are used for flow and error control whenever piggybacking is impossible or inappropriate, such as when a station does not have data to send. S-frames do not have information fields.
- Unnumbered frames, or U-frames, are used for various

miscellaneous purposes, including link management. Some U-frames contain an information field, depending on the type.

7.8.5 Point-To-Point Protocol

In networking, the **Point-to-Point** Protocol (PPP) is a data link protocol commonly used in establishing a direct connection between two networking nodes. It can provide connection authentication, transmission encryption (using ECP, RFC 1968), and compression. It is used over many types of physical networks including serial cable, phone line, trunk line, cellular telephone, specialized radio links, and fiber optic links such as SONET. PPP is also used over Internet access connections (now marketed as "broadband"). Internet service providers (ISPs) have used PPP for customer dial-up access to the Internet, since IP packets cannot be transmitted over a modem line on their own, without some data link protocol. Two derivatives of PPP, Point-to-Point Protocol over Ethernet (PPPoE) and Point-to-Point Protocol over ATM (PPPoA), are used most commonly by Internet Service Providers (ISPs) to establish a Digital Subscriber Line (DSL) Internet service connection with customers. PPP is commonly used as a data link layer protocol for connection over synchronous and asynchronous circuits, where it has largely superseded the older Serial Line Internet Protocol (SLIP) and telephone company mandated standards (such as Link Access Protocol, Balanced (LAPB) in the X.25 protocol suite). PPP was designed to work with numerous network layer protocols, including Internet Protocol (IP), TRILL, Novell's Internetwork Packet Exchange (IPX), NBF and AppleTalk. PPP permits multiple network layer protocols to operate on the same communication link. For every network layer protocol used, a separate Network Control Protocol (NCP) is provided in order to encapsulate and negotiate options for the multiple network layer protocols. It negotiates network-layer information, e.g. network address or compression options, after the connection has been established.

Structure of a PPP frame

Name	Number of bytes	Description
Protocol	1 or 2	setting of protocol in data field
Information	variable (0 or more)	datagram
Padding	variable (0 or more)	optional padding

The **Protocol** field indicates the type of payload packet (e.g. LCP, NCP, IP, IPX, AppleTalk, etc.).

The **Information** field contains the PPP payload; it has a variable length with a negotiated maximum called the Maximum Transmission Unit. By default, the maximum is 1500 octets. It might be padded on transmission; if the information for a particular protocol can be padded, that protocol must allow information to be distinguished from padding.

PPP frames are encapsulated in a lower-layer protocol that provides framing and may provide other functions such as a checksum to detect transmission errors. PPP on serial links is usually encapsulated in a framing similar to HDLC.

Name	Number of bytes	Description
Flag	1	indicates frame's begin or end
Address	1	broadcast address
Control	1	control byte
Protocol	1 or 2	I in information field
Information	variable (0 or more)	datagram
Padding	variable (0 or more)	optional padding
FCS	2 (or 4)	error check

The **Flag** field is present when PPP with HDLC-like framing is used.

The **Address** and **Control** fields always have the value hex FF (for "all stations") and hex 03 (for "unnumbered information"), and can be omitted whenever PPP LCP Address-and-Control-Field-Compression (ACFC) is negotiated.

The **Frame Check Sequence** (FCS) field is used for determining whether an individual frame has an error. It contains a checksum computed over the frame to provide basic protection against errors

in transmission. This is a CRC code similar to the one used for other layer two protocol error protection schemes such as the one used in Ethernet. According to RFC 1662, it can be either 16 bits (2 bytes) or 32 bits (4 bytes) in size (default is 16 bits - Polynomial x16 + x12 + x5 + 1). The FCS is calculated over the Address, Control, Protocol, Information and Padding fields after the message has been encapsulated.



CHECK YOUR PROGRESS

2. Fill in the blanks:

(a) The physical layer process and some of the data link layer process run on dedicate hardware called a _____.

(b) Protocols in which the sender sends one frame and then waits for an acknowledgement before proceeding are called _____.

(c) In the _____ protocol the receiver simply discards all subsequent frames, sending no acknowledgements for the discarded frames.

(d) In the Selective Repeat protocol both _____ and _____ maintain a window of outstanding and acceptable sequence numbers, respectively

_____ frames can be transmitted over synchronous or asynchronous links.



7.9 LET US SUM UP

- In the seven-layer OSI model of computer networking, the ***Data Link*** layer is layer 2.
- The ***byte count framing*** method uses a field in the header to specify the number of bytes in the frame.
- ***Error control*** makes sure all frames are eventually delivered to the network layer at the destination and in the proper order.

- In ***feedback-based flow control***, the receiver sends back information to the sender giving it permission to send more data.
- The number of bit positions in which two code words differ is called the ***Hamming distance***.
- ***Convolutional codes*** are specified in terms of their rate and constraint length.
- ***Reed-Solomon*** codes are widely used in practice because of their strong error-correction properties, particularly for burst errors.
- ***Selective Repeat*** protocol, is to allow the receiver to accept and uffer the frames following a damaged or lost one.
- ***HDLC*** provides both connection-oriented and connectionless service.
- ***Point-to-Point Protocol*** (PPP) is a data link protocol commonly used in establishing a direct connection between two networking nodes



7.10 FURTHER READING

- Tanenbaum, A. S., & Wetherall, D. (2011). *Computer Networks,* Boston: Pearson Prentice Hall.
- Bonanenture, O. (2011). *Computer Networking: Principles, Protocols, and Prectice.* The Saylor Foundation.



7.11 ANSWERS TO CHECK YOUR PROGRESS

Answer 1:

(a) data link
(b) Rate-based
(c) Error-correcting
(d) Error-detecting
(e) Hamming distance
(f) convolutional code
(g) Reed-Solomon codes
(h) LDPC codes

(i) checksum
(j) Polynomial codes.

Answer 2:

(a) Network Interface Card
(b) stop-and-wait
(c) Go-Back-N
(d) sender, receiver
(e) HDLC.



7.12 MODEL QUESTIONS

Q 1. What is the data link layer of the OSI model? Describe its functions.
Q 2. What is Framing in the data link layer? Describe the different framing methods used.
Q 3. Describe the concept of error control in the data-link layer.
Q 4. What is flow control in the data-link layer? What are the available Flow control techniques?
Q 5. What do you understand by error detection and error correction in the data link layer?
Q 6. Illustrate with examples three error correcting codes present in the data link layer.
Q 7. What are error detecting codes in the data link layer? Describe.
Q 8. Illustrate the Stop-and-Wait protocol of the data link layer.
Q 9. Describe the Go-Back-N protocol of the data link layer.
Q 10. Describe the working of the Selective Repeat protocol of the data link layer.
Q 11. Write a short note on the HDLC protocol of the data link layer.
Q 12. Explain the Point-to-Point protocol of the data link layer.

****