Chapter 1 Introduction to Data and Network Communications

The 1970s and 1980s saw a merger of the fields of computer science and data communications that profoundly changed the technology, products, and companies of the now-combined computer-communications industry. Although the consequences of this revolutionary merger are still being worked out, it is safe to say that the revolution has occurred, and any investigation of the field of data communications must be made within this new context.

OBJECTIVES OF THIS CHAPTER

After completing this chapter, the student should be able to:

- define signal and type signal – analog versus digital signal
- identify characteristics of analog signal - amplitude, frequency and phase
- define and calculate bit rate, baud rate
- identify limits of achievable data rate in digital communication
- define basic data communications terminologies
- have an overview of a data communications system and its basic underlying characteristics

1.1 INTRODUCTION

Now that we are fully absorbed by the Information Age and spending more time communicating and gathering information through the Internet. It has become necessary to have a working knowledge of the technology behind the scenes. We are faced with terms like baud rate, modems, cellular phones, TCP/IP, ATM, ISDN, CDMA, WILL, Broadband etc., and trying to make decisions about our communications needs involving the systems that these terms apply to. In order to develop a useful working understanding of this technology requires you to have a good understanding of the background technology and basics of data communications.

1.2 Signals

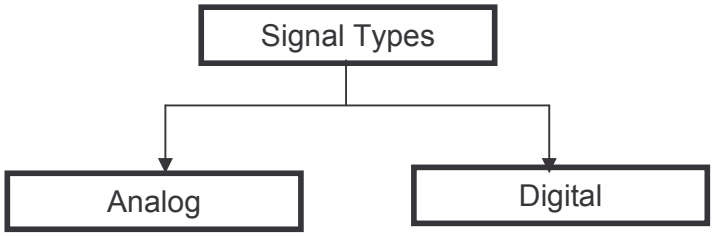


Figure 1.1 Signal Types

Signal is an electrical transmission of alternating current (AC) on network cabling that is generated by a networking component such as a network interface card (NIC). An electromagnetic signal is transmitted through air, vacuum to satellite or antenna to mobile. Signals can be either analog or digital.

An *analog signal* is a continuous wave form that changes smoothly over time. An analog signal can take on any value in a specified range of values. As the wave moves from value A to B, it passes through and includes an infinite number of values along its path. A simple example is alternating current (AC), which continually varies between about +110

volts and -110 volts in a sine wave fashion 50 times per second. A more complex example of an analog signal is the time-varying electrical voltage generated when a person speaks into a dynamic microphone or telephone. Analog signals such as telephone speech contain a wealth of detail, but are not readily accessible to computers unless they are converted to digital form using a device such as an analog-to-digital converter (ADC). Analog signals are usually specified as a continuously varying voltage over time and can be displayed on a device known as an oscilloscope. Amplitude is absolute value of signal at an instance. The maximum voltage displacement of a periodic (repeating) analog signal is called its **amplitude**, and the shortest distance between crests of a periodic analog wave is called its **wavelength**.

An example of *analog data* is the human voice. When somebody speaks, a continuous wave is created in the air. Analog data -- voice, video -- continuously varying patterns of different intensity (amplitude).

Analog signal can be classified as simple or composite. A simple analog signal, a sine wave, cannot be decomposed into simpler signals. A composite analog signal is composed of multiple sine waves. Three characteristics namely – amplitude, frequency and phase fully describes a sine wave.

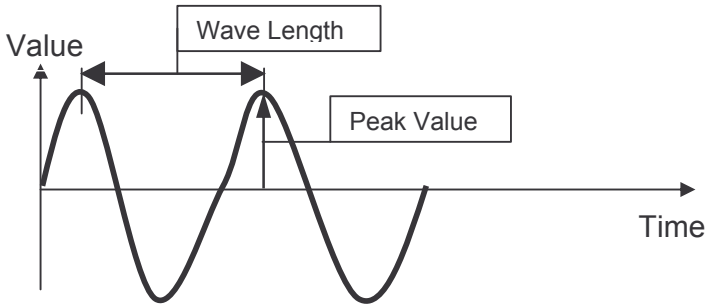


Figure 1.2 Analog Signal

Peak Amplitude

The peak amplitude of a signal represents the absolute value of its highest intensity, proportional to the energy it carries. For electric signal, peak amplitude is normally measured in volts.

Period and Frequency

Period refers to the amount of time, in second, a signal needs to complete one cycle. Frequency refers to the number of periods in one second.

Period is expressed in seconds and frequency is expressed in hertz (Hz).

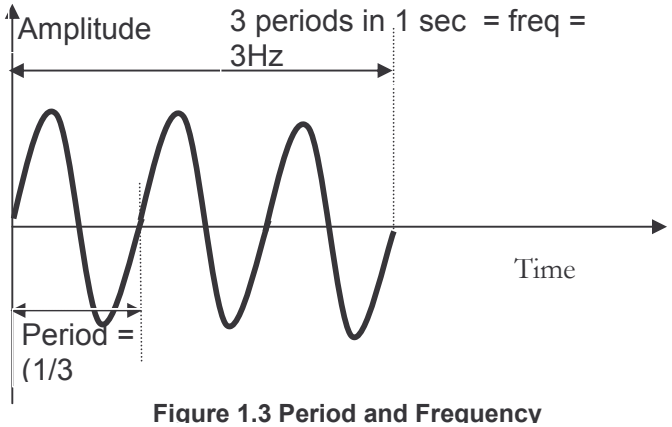


Figure 1.3 Period and Frequency

Unit time	Equivalent	Unit frequency	Equivalent
Seconds (s)	1 s	Hertz (Hz)	1 Hz
Milliseconds (ms)	10^{-3} s	Kilohertz (KHz)	10^{3} Hz = 1 KHz
Microseconds (μs)	10^{-6} s	Megahertz (MHz)	10^{6} Hz = 1 MHz
Nano second	10^{-9} s	Gegahertz (GHz)	10^{9} Hz = 1 GHz

Table 1.1: Units of Period and Frequency

Phase

Phase describes the position of the waveform relative to the time zero. Phase is measured in degrees or radians. The Phase is denoted by symbol [Φ]. A periodic signal is represented by an equation.

$$x(t) = x(t) + t_0 \quad OR \quad x(t) = A \sin (2\pi ft + \Phi]$$

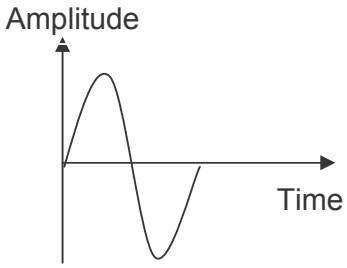


Fig (a) Phase angle 0^0

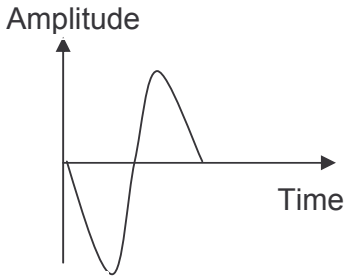


Fig (b) Phase angle 180^0

Figure 1.4 Different Phase state

A *digital signal* is discrete. It has only a limited number of definite discreet values, as 1 and 0.

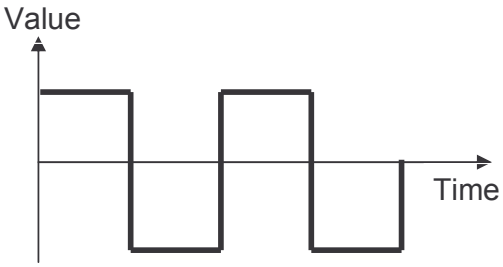


Figure 1.5 Digital Signal

Digital signal : Transmission of signals that vary discretely with time between two values of some physical quantity, one value representing the binary number 0 and the other representing 1. Digital signals use discrete values for the transmission of binary information over a communication medium such as a network cable or a telecommunications link. On a serial transmission line, a digital signal is transmitted 1-bit at a time.

An example of *digital data* is data stored in memory of a computer in the form of 0's and 1's. For example, a 1 can be encoded as a positive voltage and a 0 as zero voltage. Digital data -- text, digitized images -- takes discrete values, usually binary (0,1). Example of digitized text is the ASCII code. 8 - bits so 255 patterns including - upper and lower case characters, integers 0-9, special characters and some "control" characters are used in communication.

Bit interval and **baud rate** are used to describe digital signals.

The *bit interval* is the time required to send one single bit. The *bit rate* is the number of bit intervals per second. This means that the bit rate is number of bits sent in one second, usually expressed in bits per second (bps).

Baud rate refers to the number of signal units per second that are required to represent those bits. Baud rate is less than or equal to the bit rate. The difference between baud rate and bit rate occurs as they define different but related information. Thus Baud rate is effective measure of information transmitted and bit rate is measure of the data transmitted (which might include error correcting codes, frame, frame-packet numbers etc.].

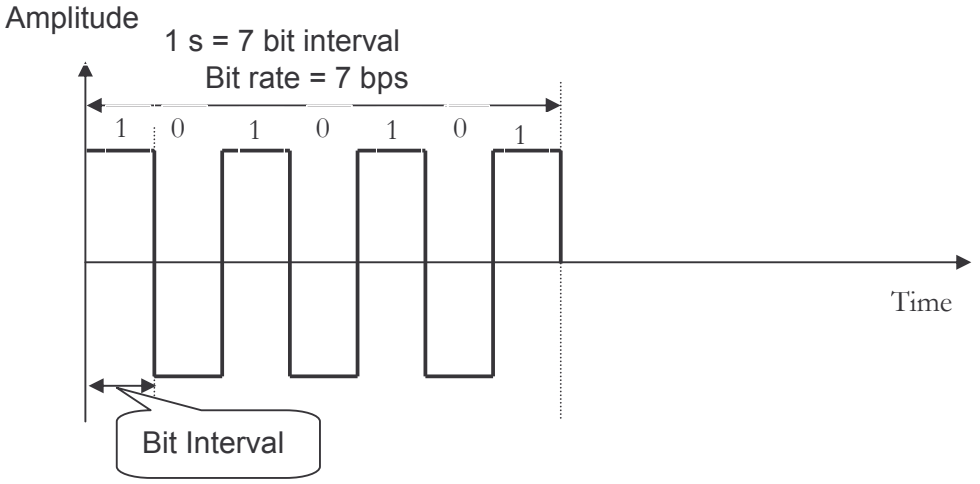


Figure 1.6 Bit rate and bit interval

Example 1

A signal carries three bits in each signal element. If 1200 signal elements are sent per second, find the baud rate and the bit rate.

Solution

Baud rate = Number of signal elements
= 1200 bps

Bit rate = baud rate × Number of bits per signal element
= 1200 × 3
= 3600 bps

Example 2

The bit rate of a signal is 2000. If each signal element carries five bits, what is the baud rate?

Solution

Baud rate = Bit rate / Number of bits per signal element
= 2000 / 5
= 400 bps

Terms used in Data Communications

Data : Data refers to the information or message, which is present in the form that is agreed upon by user and creator of data (mostly Digital data)

Data Communication : is exchange of data between two devices via some form of transmission medium.

Message or Signal : is electrical or electromagnetic wave sent through medium from one point to another, which contains encoded message. A messages can be in the form of sound, text, numbers, pictures, video or combinations of these.

Sender : A sender is device which sends the message, example : computer, workstation, video camera, telephone etc.

Medium : It is physical path over which data travels from a sender to receiver.

Receiver : A receiver is a device which receives the message, example : computer, TV receiver, workstation, telephone receiver, radio receiver etc.

Protocol : A protocol is defined as the set of rules which governs data communication. The connection of two devices takes places via the communication medium but the actual communication between them will take place with take place with the help of a protocol.

1.3 ANALOG versus DIGITAL TRANSMISSION

ANALOG TRANSMISSION -- a means of transmitting ONLY analog signals.

- Data can be analog or digital; signal is always analog.
- Propagation can be over guided [wired, coaxial, optical fiber, cable] or unguided medium (space, atmosphere).
- Analog signal will become weaker in signal strength (attenuate) over distance and will be impaired by noise.
- An AMPLIFIER will boost the energy of the signal but also the noise. Noise is a undesirable random electrical transmission on network cabling that is generated by networking components such as network interface cards (NICs) or induced in cabling by proximity to electrical equipment that generates electromagnetic interference (EMI).
- No coding is possible and thus no self-error correction is possible.

DIGITAL TRANSMISSION -- a means of transmitting both digital and analog signals. Usually assume that the signal is carrying digital (or digitized) data.

- Digital transmission can propagate to a limited distance before attenuation distorts the signal and compromises the data integrity.
- A REPEATER retrieves the (digital) signal; recovers the (digital) data, e.g., a pattern of 1's and 0's; and retransmits a new signal.

Digital transmission is the preferred method for several reasons :

- Equipment used for digital transmission is cheaper as compared to analog transmission.
- Use of repeaters, which recover the data and retransmit, are preferred over amplifiers, which boost both signal and noise.
- Errors are not cumulative and so it is possible to transmit over longer distances, using lower quality guided medium with better data integrity.

- Multiplexing - transmission links have high bandwidth and must propagate multiple signals simultaneously to utilize the bandwidth. In digital transmission, we use time-division multiplexing -- signals share the same medium over different time slots. This is easier than analog transmission where the analog signals occupy different frequency spectrum (frequency-division).
- Encryption of signal is possible for security and privacy.
- Coding is possible and self-error correction is possible.

1.4 Limits on achievable data rate in digital communication

How fast we can send data, in bits per second, through a channel.

Data rate depends upon three factors :

1. The bandwidth available.
2. The levels of signals.
3. The quality of the channel.

Two theoretical formulas were developed to calculate the data rate : one by Nyquist for a noiseless channel, another by Shannon for a noisy channel.

For noiseless channel, the Nyquist bit rate formula defines the theoretical maximum bit rate as:

Bitrate = 2 × Bandwidth × log2 L

Where, Bandwidth is the bandwidth of the channel

L is the number of signal levels used to represent data, and
Bitrate is the bit rate in bits per second.

Example 3

Consider a noiseless channel with a bandwidth of 2000 Hz transmitting a signal with two signal levels. Calculate the bit rate.

Solution Bit rate = $2 \times 2000 \times \log_2 2$

= **4000 bps**

Example 4

Consider the same noiseless channel, transmitting a signal with four signal levels.

Solution Bit rate = $2 \times 2000 \times \log_2 4$

= **8000 bps**

In reality, we cannot have a noiseless channel; the channel is always noisy. Claude Shannon introduced a formula, called the **Shannon capacity**, to determine the theoretical highest data rate for a noisy channel :

Capacity = Bandwidth × $\log_2$ (1 + SNR)

Where, Bandwidth is the bandwidth of the channel

SNR is the signal-to-noise ratio, and
Capacity is the capacity of the channel in bits per second.

The signal-to-noise ratio is the statistical ration of the power of the signal to the power of the noise.

Example 5

Calculate the channel capacity of telephone line using Shannon formula.

Solution

A telephone line has a bandwidth of 3000 Hz (300 Hz to 3300 Hz). The signal-to-nose ration is usually 3162. For this channel capacity is :

$$\begin{aligned} \text{Capacity} &= \text{Bandwidth} \times \log_2 (1 + \text{SNR}) \\ &= 3000 \times \log_2 (1 + 3162) \\ &= 3000 \times \log_2 (3163) \\ &= 3000 \times 11.62 \\ &= \mathbf{34{,}860\ bps} \end{aligned}$$

That is, the highest bit rate for a telephone line is 34.860 Kbps. If we want to send data faster than this, we can either increase the bandwidth of the line or improve the signal-to-noise ratio.

1.5 DATA COMMUNICATION SYSTEM

Data Communications Link

The components of a basic communications link between two endpoints, or **nodes,** are illustrated in Figure 1-8. A node is any connection point to a communications link. For this two-point network, the node points are the **primary station or sender** and the **remote / secondary station or Receiver** at the other end of the communications link. Station refers to any section of hardware whose purpose is to communicate with another piece of communications hardware at a different location. Data link refers to the process of connecting or linking two stations together.

Data sent from one station to another usually originates in parallel binary form from one or more peripheral devices including, but not limited to, computer terminals, printers, keyboards, facsimile (FAX) machines, and data display terminals. Note that the information supplied by the peripherals or networks could be anything from a series of keyboard characters to a stream of digitized video. This information is converted from its natural form into digital form by the peripheral and is presented as groups of parallel binary data to the system. **Parallel data** are a group of digital bits that are available at the same time, often referred to as digital or binary **words.** An individual communications path is required for each bit, allowing data to move quickly, transferring a complete word each time However, the need for multiple data paths is impractical and costly for long distance transfers. Instead, it is preferable to send data along a single data path between two stations. In order to do this, the parallel data need to be converted into **serial** form, with one bit of data sequentially following another. While parallel data allows many bits to be sent at once, serial data requires each bit to be sent separately.

EXAMPLE 1.1

Compare the time it takes to send the following short message from one station to another, first as an 8-bit or byte parallel data and then as serial data. The transfer rate is 1 ms per transfer.

2C3B in hexadecimal, which is 0010110000111011 in binary

Solution

As byte parallel data, there are two groups of 8-bit words for this message, 2C or 00101100 and 3B or 00111011. It takes two transfers, or 2 ms, to send the data in that form. In comparison, sending the 16 bits of data serial at 1 ms per transfer, results in 16 ms of total time required to complete the transmission.

UARTS

Devices that perform the parallel-to-serial conversion (and vice versa at the receiving station) are the **universal asynchronous receiver transmitter (UART)** and the **universal synchronous/asynchronous receiver transmitter (USART).** The conversion process and the rates at which parallel data are sent to the UART or USART and the rate at which serial data are sent and received are controlled by the computer system to which the UART or USART is connected.

DTE and DCE Equipment

The computer system or communication's station, and UART are grouped together and classified as **data terminal equipment (DTE).** The computing system part of the DTE contains software and hardware needed to establish and control the communications link between the primary and secondary stations. An applications program used by the DTE uses **protocol** that defines a set of rules that determine the requirements for the successful establishment of a data link and the transfer of actual information between the stations. Protocols exist at many levels of an overall network or communications system. While they can become quite sophisticated, their basic premise never changes. They are always used to set the requirements for moving information between two or more node points within a network or system.

In the two point system of Figure 1-8, the DCE (**data communications equipment** or **data circuit terminating**) is a **modulator-demodulator (modem).** This device is used to convert the serial data stream into a form that can be used by the connecting **medium (communication channel)** to transfer data over long distances. One common medium is the existing telephone lines that normally carry voice calls. In order to be able to use the telephone system, the serial data needs to be converted to audio range signals. Several types of modems perform this conversion using a number of methods dictated by the rate at which the data needs to be transmitted. One such method changes logic 1 (mark) and logic zero (space) levels to audio signals of two different frequencies. Thus when logic 1 is sent, the mark signal or tone is sent and logic 0 generates the space tone. The receiving modem at the secondary station converts these tones back to binary logic states, which are, in turn, sent to the UART or USART for conversion to parallel data used by the receiver's computer system.

The medium between the primary and secondary stations can be as simple as a coaxial or twisted pair cable as used by local area communications networks. Telephone lines are another example of the use of twisted pairs of wires for local connections between phone users and phone switching stations. The telephone companies, as well as other carriers, have long adopted the use of radio transmission at many levels to complete communication links. These include microwave, cellular, and satellite transmissions.

Hardware Interfaces

In early days, interconnecting data terminal equipment to data communication equipment so they will work harmoniously is complicated because different manufacturers produce

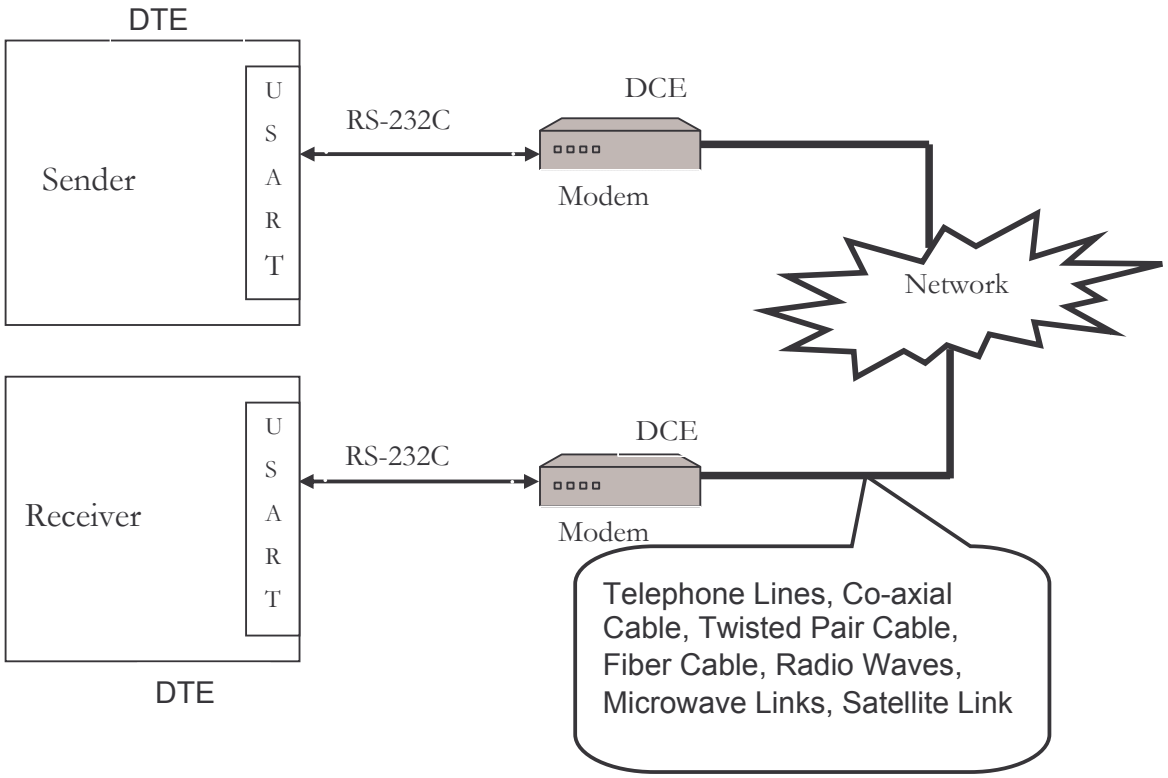


Figure 1.7 Basic Digital Communication Network link

various types of devices. These cannot be easily connected to one another. A need for a standard interface between the DTE and DCE units is crucial. One example of a commonly used interface standard, the RS232C, was written by communication engineers for the *Electronics Industries Association (EIA),* which is one of many organizations responsible for establishing standards for variety of electronics and electrical applications. *RS* stands for *recommended standard*, which means there is no enforcing authority to assure the proper use or complete compliance with the specifications of the standard. Enforcement of its use falls to the consumer, who benefits by not purchasing those systems that do not include the standard and buying those that do. Encompassed in the standard are functional, electrical, and physical specifications for users wishing to connect DTE to DCE equipment.

SUMMARY

Signals and data can be either analog or digital. An *analog signal* is a continuous wave form that changes smoothly over time. A *digital signal* is discrete. It can have only a limited

number of definite values, often as simple as 1 and 0. Bit interval and baud rate are used to describe digital signals.

PRACTICE SET

Review Questions

1. Contrast between an analog signal and a digital signal.
2. A signal has been received that only has values of –1, 0, and 1. Is this an analog or a digital signal?
3. What is a bit interval?
4. What is a baud rate?
5. Describe the characteristics of sine wave.
6. What are the units of period and frequency?
7. What is the unit to measured amplitude?
8. What is the counterpart of bit interval in analog signal?
9. Consider a noiseless channel with a bandwidth of 4000 Hz transmitting a signal with two signal levels. Calculate the bit rate.
10. Consider a noiseless channel with a bandwidth of 2000 Hz transmitting a signal with four signal levels. Calculate the bit rate.
11. A signal carries three bits in each signal element. If 1500 signal elements are sent per second, find the baud rate and the bit rate.
12. A signal carries two bits in each signal element. If 1200 signal elements are sent per second, find the baud rate and the bit rate.
13. The bit rate of a signal is 8000. If each signal element carries five bits, what is the baud rate?
14. The bit rate of a signal is 4000. If each signal element carries four bits, what is the baud rate?

Multiple Choice Questions

1. Unit of frequency is ……
   A) meter B) second C) Hertz D) Degree

2. Unit to measure phase is …….
   A) meter B) second C) Hertz D) Degree

3. Unit to measure amplitude of electric signal is ……..
   A) volts B) seconds C) Hertz D) meters

4. What is the bandwidth of a signal that ranges from 40 KHz to 4 MHz?
   A) 36 MHz B) 360 KHz C) 3.96 MHz D) 396 KHz

5. Bit interval is used to describe …….
   A) Sine wave B) Digital signal C) Analog Signal D) No signal

6. Bit rate is used to describe ……..
   A) Sine wave B) Digital signal C) Analog Signal D) None of above

7. ……..is one of the characteristics of sine wave.
   A) Meter B) Frequency C) Longitude D) None of above

8. Signal can be ……..
   A) Analog B) Digital C) Either A) or B) D) None of the above

9. Which of the following signal can have an infinite number of values in a range?
   A) Analog B) Digital C) Either A) or B) D) None of the above

10. Which of the following signal can have only a limited number of values?
    A) Analog B) Digital C) Either A) or B) D) None of the above

11. For which of type of channel, the Nyquist bit rate formula defines the theoretical maximum bit rate?
    A) noisy B) noiseless C) bandpass D) None of the above

12. For which of the following channel, we need to use the Shannon capacity to find the maximum bit rate?
    A) noisy B) noiseless C) bandpass D) None of the above

* * *

Chapter 2 DATA TRANSMISSION

The successful transmission of data depends principally on two factors: the quality of the signal being transmitted and the characteristics of the transmission medium

OBJECTIVES OF THIS CHAPTER

After completing this chapter, the student should be able to:

- define mode of data transfer, such as simplex, half-duplex and full-duplex
- define the character types represented in a binary character code.
- identify different data types, rates, and binary data formats.
- define serial data formats such as synchronous and asynchronous.
- identify transmission impairments
- define different encoding formats
- understand the devices that performs modulation and demodulation as modem

2.1 Modes of Data Transfer

Data communications links are configured to satisfy particular requirements for a given system. The primary station initiates (or originates) the communication link and maintains control over that link until all data transfers are completed. The answering station is the secondary. Both stations must use the same protocol, data rates, and data codes for data to be correctly sent and received. Data transfer can be one-directional only or bidirectional. The actual method of sending and receiving data is further divided into three types—simplex, half duplex, and full duplex.

A system or a particular data transmission can be configured to send and receive data in one direction only (from primary to secondary, for example). This transmission is referred to as a **simplex** transmission. if, for example, the primary would always transmit data to the secondary and the secondary would not be required to respond or send anything in return. In simplex mode, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive. Simplex means communication runs in one direction.

Simplex transmissions are useful in environments where large quantities of data are sent without acknowledgement of the reception. The data can be sent fast and continuously. For example, keyboard to CPU (Keyboard transfers data to CPU in one direction only) or CPU to Monitor (CPU transfers data or gives output to monitor but does not accept data from monitor). In simplex mode, the communication is unidirectional, as if a one-way street. Only one of the two devices on a link can

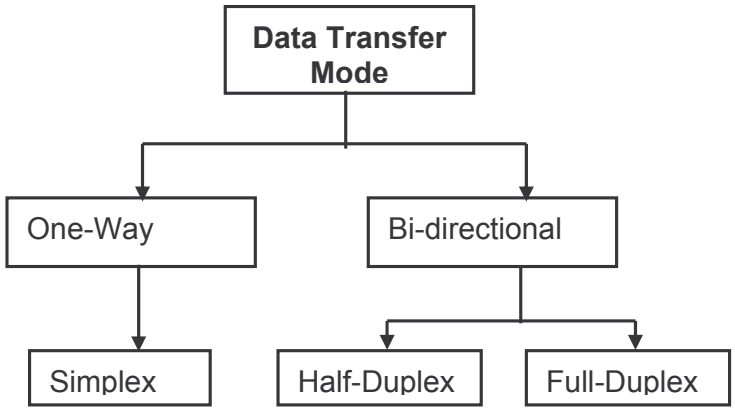


Figure 2.1 Data Transfer

transmit; the other can only receive.

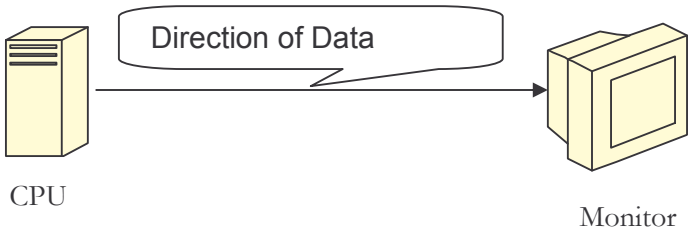


Figure 2.2 (a) Simplex

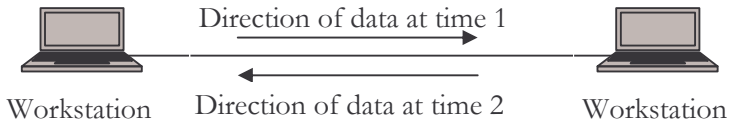


Figure 2.2 (b) Half-Duplex

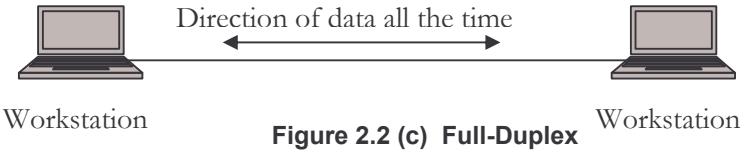


Figure 2.2 (c) Full-Duplex

Figure 2.2 Directions or modes of Communication

There are two basic methods of bidirectional data transfer or communication, **half duplex** and **full duplex.** Half duplex allows transmission of data in both directions between primary and secondary stations, but restricts these transfers to one direction at a time as shown in Figure 2.2 (b). The primary might begin by sending a message from its transmitting circuits to the secondary receiving units, which receives and stores the data. After the message is completely sent, the secondary can reply with a message of its own from its transmitter to the primary's receiver.

The example of simplex communication includes :

- TV and radio broadcasting or pager.
- Keyboards and traditional monitors are both examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output.

Simplex transmission occurs in many common communication applications, the most obvious being broadcast and cable television. It is not used in true network communication because stations on a network generally need to communicate both ways.

Half-duplex

In *half-duplex mode*, each device can both transmit and receive message, but not at the same time. For half-duplex, both end devices can send and receive (they must alternate). In a **half duplex** transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time. When one device is sending, the other can only receive, and vice versa. In a half duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time. The simplest example is a walkie-talkie: You have to press a button to talk and release the button to listen. When two people use walkie-talkies to communicate, at any given moment, only one of them can talk while the other listens. If both try to talk simultaneously, a collision occurs and neither hears what the other says.

Communication through traditional Ethernet networks is another example of half-duplex communication. When one station on an Ethernet transmits, the other stations detect the carrier signal and listen instead of transmitting. If two stations transmit signals simultaneously, a collision occurs and both stations stop transmitting and wait random intervals of time before retransmitting.

The simplest example of half-duplex transmission is a walkie-talkie: You have to press a button to talk and release the button to listen. When two people use walkie-talkies to communicate, at any given moment, only one of them can talk while the other listens. If both try to talk simultaneously, a collision occurs and no one can hears what the other says.

In contrast, full-duplex communication enables stations to transmit and receive signals simultaneously, with the advantage of providing twice the bandwidth of equivalent half-duplex technologies. However, full-duplex requires two communication channels to achieve these results - one to transmit and one to receive signals.

Full-duplex

In *full duplex mode* (also called *duplex*), both stations can transmit and receive message simultaneously. The full-duplex mode is like a two way street with traffic flowing in both direction at the same time. In full-duplex mode, signals going in either direction share the capacity of the link. **Full duplex** is illustrated in Figure 2.2 (c).

Sharing of link can occur in two ways : Either the link must contain two physically separate transmission paths, one for sending and the other for receiving; or the capacity of the channel is divided between signals traveling in both directions.

One way to achieve this is for the primary and secondary to transmit binary data using two different sets of audio tones for logic 1 and 0. For instance, the primary might send data using 1250 Hz and 1750 Hz signals for mark and space while the secondary would use 2050 Hz and 2750 Hz. The receive portions of both stations' modems would be tuned to the appropriate set of tones in order to differentiate between those it sent and those it received. For example, conversation over telephone is full duplex.

Full-duplex requires two communication channels to achieve these results - one to transmit and one to receive signals.

One common example of full duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time.

The full-duplex mode is like a two way street with traffic flowing in both direction at the same time. In full-duplex mode, signals going in either direction share the capacity of the link. Sharing of link can occur in two ways : Either the link must contain two physically separate transmission paths, one for sending and the other for receiving; or the capacity of the channel is divided between signals traveling in both directions. One common example of full duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time.

2.2 CHARACTER CODES

Basic to every system deployed and used today is the transfer of information via a digital code. Text characters—alphabetic, numeric, punctuation, formatting, attribute, etc.—use selected combinations of 1s and 0s in a fixed binary word size, which is tabulated into a set known as a **character code.** Besides data communications applications, binary

character codes are used wherever a digital processor or computer is used to interpret letters, numbers, and directed functions. A common example, most students are familiar with, is the personal computer keyboard and display screen. As the user types on the keyboard, a processor within the keyboard converts the key press into a binary code and signals the computer. When text is to be sent to the screen, the text is converted from the characters we read and are familiar with to the same binary code used by the keyboard.

ASCII

The character code used by this system is the more familiar ASCII code. *ASCII* is an acronym for *American standard code for information interchange.* The original ASCII code used a 7-bit binary word to represent the printable characters, formatting characters, and data-linking, or control code, characters used by data communication systems today.

Somewhere in the 1980s, the code was expanded to an 8-bit word size. Known as **extended ASCII,** this code also includes the Greek symbols used in mathematics, special alphabet characters used by different languages, such as the umlaut on German vowels, and other specialty characters (smiling faces, rudimentary graphics, etc.). *Table 1-1* lists the original ASCII set to allow us to explore some of its components. The table shows the characters (ASCII) and hexadecimal (HEX) representation of the 7-bit binary codes for each ASCII character. By looking at the table, you can find the **alphanumeric** characters you would expect—alphabetic, numeric, and punctuation characters. The remaining characters are classed into two groups—**formatting** and **data-linking** or **control characters.** Formatting characters are responsible for how text appears on the page and includes line feed (LNFD), carriage return (CRET), horizontal and vertical tabs (HTAB, VTAB), form feed (FMFD), etc.

Data-linking or control characters are those used by protocols to establish and maintain a data link. They include characters for indicating the beginning of a transmission (STX—start of text), ending a transmission (EOT—end of transmission), acknowledge (ACK), delimiter (DLE), device control (DC1–DC4), etc. A further extension to the ASCII code set has been formulated under the name of **UNICODE.** Besides the basic extended ASCII set, this 16-bit character code embraces the use of foreign letters and special symbols such as the German umlaut. Since the code includes a total of 65,536 possible characters, a table for it is *not* included with this text.

ASCII Character Code

Binary codes are shown in their hexadecimal (HEX) equivalent.

HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII
00	NULL	20	Space	40	@	60	□
01	SOH	21	!	41	A	61	a
02	STX	22	□	42	B	62	b
03	ETX	23	#	43	C	63	c
04	EOT	24	$	44	D	64	d
05	ENQ	25	%	45	E	65	e
06	ACK	26	&	46	F	66	f
07	BELL	27	□	47	G	67	g

HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII
08	BKSP	28	(	48	H	68	h
09	HTAB	29	)	49	I	69	i
0A	LNFD	2A	*	4A	J	6A	j
0B	VTAB	2B	+	4B	K	6B	k
0C	FMFD	2C	□	4C	L	6C	l
0D	CRET	2D	–	4D	M	6D	m
0E	SHOUT	2E	.	4E	N	6E	n
0F	SHIN	2F	/	4F	O	6F	o
10	DLE	30	0	50	P	70	p
11	DC1	31	1	51	Q	71	q
12	DC2	32	2	52	R	72	r
13	DC3	33	3	53	S	73	s
14	DC4	34	4	54	T	74	t
15	NACK	35	5	55	U	75	u
16	SYNC	36	6	56	V	76	v
17	ETB	37	7	57	W	77	w
18	CAN	38	8	58	X	78	x
19	ENDM	39	9	59	Y	79	y
1A	SUB	3A	:	5A	Z	7A	z
1B	ESC	3B	;	5B	[	7B	{
1C	FLSP	3C	<	5C	\	7C	:
1D	GPSP	3D	=	5D	]	7D	}
1E	RDSP	3E	>	5E	^	7E	~
1F	UNSP	3F	?	5F	—	7F	DEL

Table 2.1: ASCII Character Set

EBCDIC

The chief character code competition for ASCII came from Big Blue—IBM (International Business Machines), which has its own character code, used primarily, for its mainframe computers. This code, known as EBCDIC (pronounced EB—CE—DIC) for **extended binary coded decimal interchange code.** EBCDIC is an 8-bit binary code that represents the same set of characters that the original 7-bit ASCII code did, but in a different sequence.

2.3 Digital Data Transfer Rates

Digital information in serial form is transferred at a distinct data rate. It takes time to send information, one bit at a time, from one place to another. Data are sent serially to reduce the number of transmitting lines or paths to a single pair (electrically active and a return line). The rate at which digital information is sent or received is called the bit rate, whose unit is bits per second (bps).

When other methods are used to send data using groups of bits in each transmission, a different measure of data rate is required. This measure is known as symbols per second (sps), where a symbol represents the bit group. Symbols can be either in binary or any other format. Sometimes they are an analog signal that uses different frequencies or phases to represent groups of data bits. What is most significant for the field of data communications is that a symbol can be created through various processes from a group of binary bits. As an example, the ASCII code, instead of being represented as a 7-bit code, could be set up to use a different voltage level for each character. In other words, 1.2113 volts could indicate the letter C and 1.2114 volts a letter D. We are not concerned with the practicality of such a system, but since there is a unique 7-bit binary code for each of the characters in the ASCII set, there could also be a unique voltage to represent each character. There would then be a correlation between the ASCII binary codes and those voltage levels as well. That is, 1000011, the binary ASCII code for the letter C is equivalent to 1.2113 Volts and 1000100 for a D is equivalent to a 1.2114 Volt level. The bottom line is that 7 bits can be used to generate a single symbol (in this case a voltage level) for each character. The single symbol can be sent as one entity, which is faster than sending the equivalent 7 individual bits.

Given the same rate of transmission, 1,000 bits per second and 1,000 symbols per second, we can compare the time it would take to send a single ASCII character using the voltage example in the preceding paragraph. The time of transmission in both cases is the reciprocal of the rate, or 1 millisecond (ms) per bit or per symbol. Sending the data in digital form requires sending seven consecutive bits for a single character, which means it takes a total of 7 ms to send the letter D in binary. When the seven bits are used to form a single voltage symbol, then only that one symbol is sent, taking only 1 ms time.

2.4 Transmission Mode

Transmission of data across link can be accomplished in either parallel or serial. Data transfer from one device to another device is always either by parallel transmission mode or serial transmission mode. In parallel mode, multiple bits are sent with each clock tick. In serial mode, one bit is sent with each clock tick.

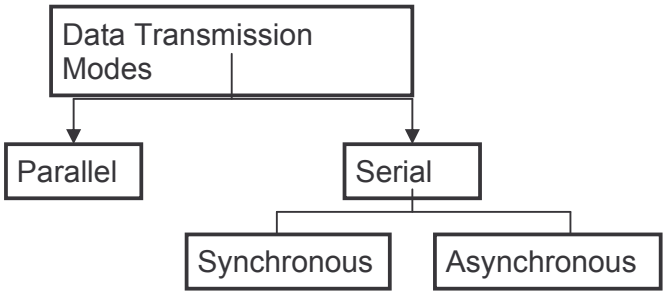


Figure 2.3 Data Transmission modes

Parallel Transmission

A form of signal transmission that sends information 8 or more bits at a time over a cable. Parallel interfaces are used mainly to connect printers, hard drives, and other peripherals to computers.

The mechanism for parallel transmission is a conceptually simple one: use n wires to send n bits at a time. For

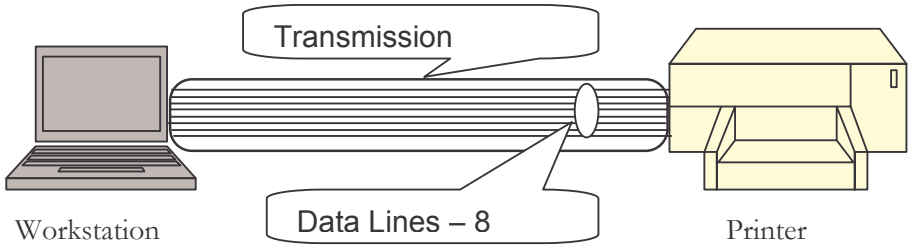


Figure 2.4 Parallel Transmission

example, to send 8 bits at a time use 8 data wires. Typically, the eight wires are bundled in a cable with connector at each end. That way each bit has its own wire, and all n bits of one group can be transmitted with each clock tick from one device to another device.

Serial Transmission

A form of signal transmission that sends information one bit at a time over a single data channel or data link. Serial interfaces are generally used to connect data communications equipment (DCE) such as modems to data terminal equipment (DTE) such as computers and terminals and for connecting a DCE to a DTE. RS-232 is the most commonly used serial interface in ordinary network communication, which supports transmission over a range of 0 to 20 Kbps at distances of up to 50 feet (15.24 meters).

In serial transmission one bit follows another, so we need only one communication channel or wire rather than n to transmit data between two communicating devices. Serial transmission is possible in one of two ways: synchronous and asynchronous.

Serial Data Formats

Whether data are sent as bits or symbols, it is transmitted serially in one of two forms, **synchronous** or **asynchronous.** Synchronous means serial data that requires a synchronizing clock signal between sender and receiver. And, Asynchronous means serial data that does not require a synchronizing clock or signal between sender and receiver.

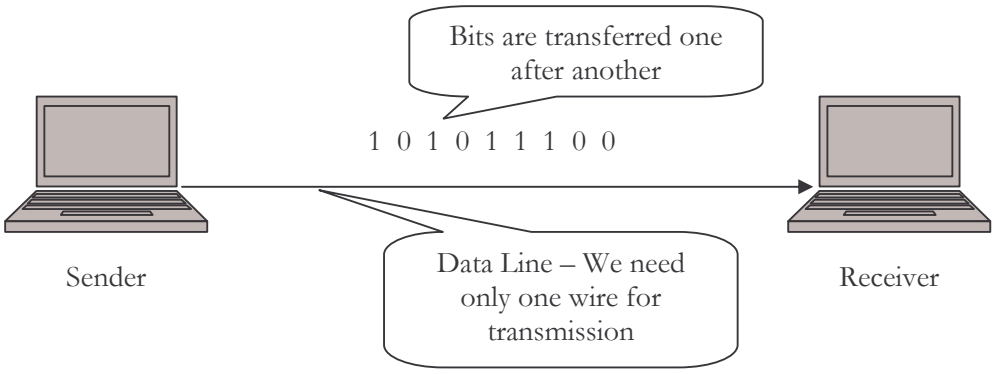


Figure 2.5 Serial transmission

Synchronous Data

Synchronous data require a coherent clocking signal between transmitter and receiver, called a *data clock,* to synchronize the interpretation of the data sent and received. The data clock is extracted from the serial data stream at the receiver by special circuits called *clock recovery circuits.*

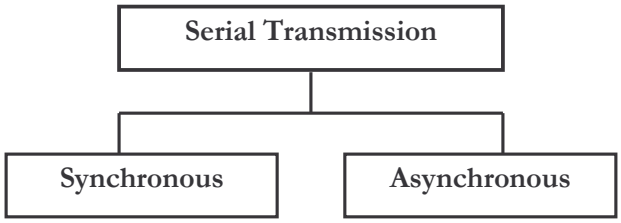


Figure 2.6 Types of serial Transmission

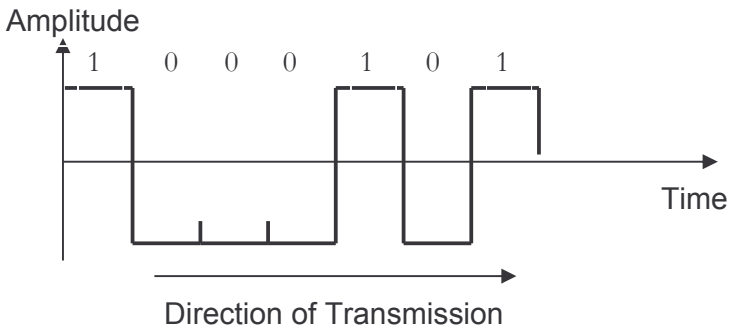


Figure2.7 (a) Synchronous E (45H or 1000101)

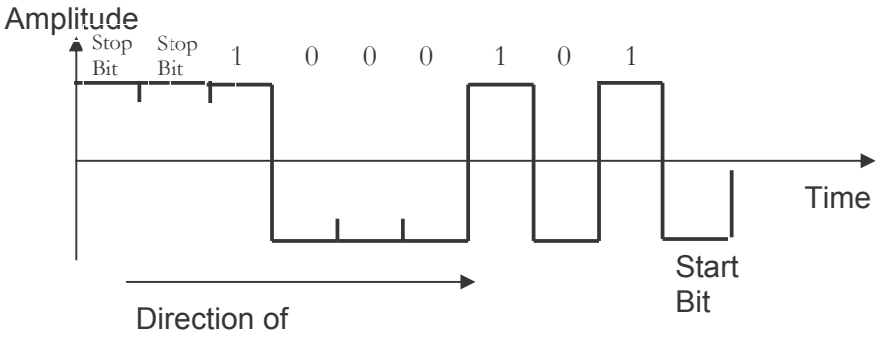


Figure2.7 (b) Asynchronous E (45 H or 100101)

Figure 2.7 Synchronous and Asynchronous Data

Once the clock is recovered at the receiving end, bit and character synchronization can be established. *Bit synchronization* requires that the high and low condition of the binary data sent matches that received and is not in an inverted state. *Character synchronization* implies that the beginning and end of a character word is established so that these characters can be decoded and defined. Overall the clock recovered from the message stream itself maintains synchronization. Figure 2.7(a) shows how a synchronous binary transmission would send the ASCII character E (hex 45 or 1000101). The least significant bit (LSB) is transmitted first, followed by the remaining bits of the character. There are no additional bits added to the transmission.

With synchronous transmission, a block of bits is transmitted in a steady stream without start and stop codes. The block may be many bits in length. To prevent timing drift between transmitter and receiver, their clocks must somehow be synchronized. One possibility is to provide a separate clock line between transmitter and receiver. One side (transmitter or receiver) pulses the line regularly with one short pulse per bit-time. The other side uses these regular pulses as a clock. This technique works well over short distances, but over longer distances the clock pulses are subject to the same impairments as the data signal, and timing errors can occur. The other alternative is to embed the clocking information in the data signal; for digital signals, this can be accomplished with Manchester or Differential Manchester encoding. For analog signals, a number of techniques can be used; for example, the carrier frequency itself can be used to synchronize the receiver based on the phase of the carrier.

With synchronous transmission, there is another level of synchronization required to allow the receiver to determine the beginning and end of a block of data; to achieve this,

each block begins with a preamble bit pattern and generally ends with a postamble bit pattern.

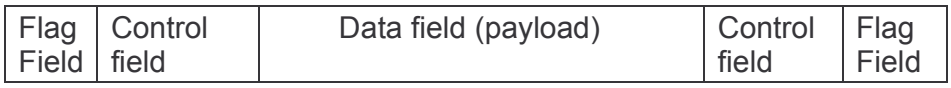


Figure 2.8 Synchronous frame format

Figure shows, in general terms. a typical frame format for synchronous transmission. Typically, the frame starts with a preamble called a flag, which is eight bit-long. The same flag is used as a postamble. The receiver looks for the occurrence of the flag pattern to signal the start of a frame. This is followed by some number of control fields, then a data field (variable length for most protocols), more control fields, and finally the flag is repeated.

For sizable blocks of data, synchronous transmission is far more efficient than asynchronous. Asynchronous transmission requires 20 percent or more overheads. The control information, preamble, and postamble in synchronous transmission are typically less than 100 bits. For example, one of the more common schemes, HDLC, contains 48 bits of control, preamble, and postamble. Thus, for a 1000-character block of data, each frame consists of 48 bits of overhead and 1000 X 8 = 8,000 bits of data, for a percentage overhead of only 0.6%.

Asynchronous Data

Asynchronous data formats incorporate the use of **framing bits** to establish the beginning (start bit) and ending (stop bit) of a data character word as shown in Figure 2.7 (b). A clocking signal is not recovered from the data stream, although the internal clocks of the transmitter and receiver must be the same frequency for data to be correctly received. To understand the format of an asynchronous character, it is first necessary to be aware of the state of the transmission line when it is idle and no data is being sent. The idle condition results from the transmission line being held at a logic 1, high state, or mark condition. The receiver responds to a change in the state of the line as an indication that data has been sent to it. This change of state is indicated by the line going low or logic 0, caused by the transmission of a start bit at the beginning of the character transmission as shown in Figure 2.7 (b). Data bits representing the code of the character being sent follow next ending with one or two stop bits. The stop bits actually specify the minimum time the line must return to logic 1 condition before the receiver can detect the next start bit of the next character.

Asynchronous transmission is simple and cheap but requires an overhead of two to three bits per character. For example, for an 8-bit code, using a 1-bit-long stop bit, two out of every ten bits convey no information but are there merely for synchronization; thus the overhead is 20%. Of course, sending larger blocks of bits between the start and stop bits could reduce the percentage overhead.

Transmission Efficiency

Notice that the synchronous data uses just the seven bits required for the E character's code while the asynchronous stream needs 10 bits (one start, seven data, and two stop bits). The synchronous stream is more efficient than the asynchronous because it does not require the overhead (framing) bits that the asynchronous stream needs. Efficiency is a mark of performance and is calculated as a ratio of data or information bits sent to total bits sent as shown in Equation 2.1.

Efficiency = (data bits / total bits) * 100 ...equation 2.1

A more efficient stream of data takes less time to be transmitted simply because there are less bits to be sent. However, the overall efficiency of a transmission relies on more than the efficiency of individual characters within a message. For asynchronous data, the entire message will retain a 70% efficiency because no additional bits or overhead are required to send the data. Bit and character synchronization are built into the framing bits. Synchronous data, on the other hand, requires a **preamble** message, which is a set pattern of binary ones and zeros used to facilitate clock recovery, so the data to be bit and character synchronized before data can be correctly received. This adds additional bits to be sent and reduces the overall efficiency of the transmission. Despite this added burden, synchronous transmissions remain more efficient than asynchronous ones.

2.5 Transmission Impairment

Transmission is the act of propagation through the medium and receiving and processing of the signal. Transmission media are not perfect. The imperfections cause impairment in the signal sent through the medium. This means that the signal at the beginning and end of the medium are not the same. What is sent is not what is received.

Signal that is received will be impaired or distorted during transmission. For analog signals, signal quality is reduced. For digital signals, errors are introduced. 1 recognized as 0 and vice versa.

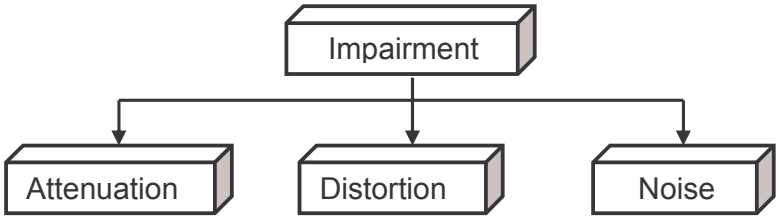


Figure 2.9 Transmission Impairment

Transmission medium is imperfect and these impairments affect the capacity of the channel. Three types of impairments can occur : *attenuation*, *distortion*, and *noise*.

i) ATTENUATION

Attenuation means loss of energy. Signal strength reduces over time. When a signal travels through a medium, it losses some of energy so that it can overcome the resistance of the

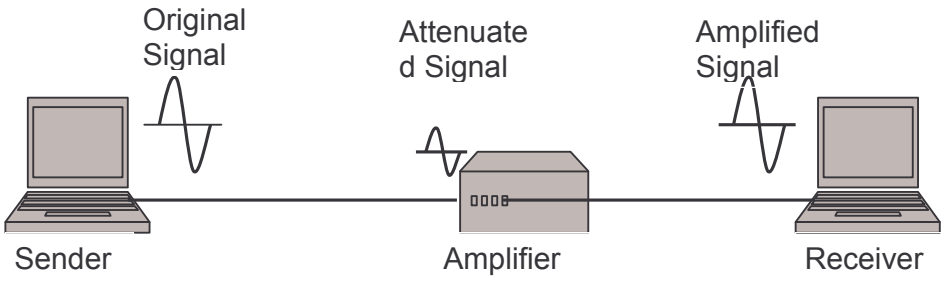


Figure 2.10 Attenuation

medium. Thus a wire carrying electrical signal gets warm, if not hot, after a while. Some of the electrical energy in the signal is converted to heat. To compensate for this loss, amplifiers are used to amplify the signal.

The loss of signal strength with long distances when signals travel along cabling. Attenuation values for actual cables are measured in units of decibels (dB) – a standard measurement value used in communication for expressing the ratio of two values of voltage, power, or some other signal-related quantity. For example, a drop of 3 dB corresponds to a decrease in signal strength of 50 percent or 2:1, while a drop of 6 dB corresponds to a decrease of 75 percent or 4:1. Attenuation values for cabling media are expressed in units of decibels per 1000 feet, which express the amount of attenuation in decibels for a standard 1000 - foot length of cabling composed of that media.

Loss of signal strength is expressed in dB decibel. It is a ratio between final and initial power, using logarithms. Loss will be negative dB and gain will be positive dB.

$$\text{loss in dB} = 10 * \log\{\text{base-10}\} \times (P_{final} / P_{initial}).$$

Attenuation increases at higher frequencies. Copper cabling has much greater attenuation than fiber-optic cabling; therefore, copper is suitable only for relatively short cable runs. Typical attenuation values for copper category 5 cabling vary with frequency and are shown in the table that follows. Attenuation for lower-grade cable is slightly higher.

Signal Frequency	Attenuation
4 MHz	13 dB/1000 feet
10 MHz	20 dB/1000 feet
20 MHz	28 dB/1000 feet
100 MHz	67 dB/1000 feet

Table 2.2 Attenuation Values for Copper Category 5 Cabling

Attenuation is caused by signal absorption, connector loss, and coupling loss. To minimize attenuation, use high-grade cabling such as enhanced category 5 cabling. Also try to minimize the number of connector devices or couplers, ensuring that these are high-grade components as well. When a signal attenuates a large amount, the receiving device might not be able to detect it or might misinterpret it, therefore causing errors.

ii) DISTORTION

Distortion means that the signal changes its form or shape. The distortion of electrical signals occurs as they pass through metallic conductors. Signals that start at the source as clean, rectangular pulses may be received as rounded pulses with ringing at the rising and falling edges. These effects are properties of transmission through metallic conductors, and become more pronounced as the conductor length increases. To compensate for distortion, signal power must be increased or the transmission rate decreased.

Delay Distortion

Delay distortion occurs in guided medium. All frequency components of the signal may not "travel" at the same speed -- can cause distortion -- e.g., for two consecutive bits, the portion of the signal carrying one bit may overlap with the portion of the signal carrying the neighboring bit.

The various frequency components in digital signal arrive at the receiver with varying delays, resulting in delay distortion.

As bit rate increases, some of the frequency components associated with each bit transition are delayed and start to interfere with frequency components associated with a later bit, causing inter-symbol interference, which is a major limitation o maximum bit rate.

iii) NOISE

Noise refers to unintentional signal (voltages) introduced in a line by various phenomenons such as heat or electromagnetic induction created by other sources.

Noise is an undesirable random electrical transmission on network cabling that is generated by networking components such as network interface cards (NICs) or induced in cabling by proximity to electrical equipment that generates electromagnetic interference (EMI). Noise is generated by all electrical and electronic devices, including motors, fluorescent lamps, power lines, and office equipment, and it can interfere with the transmission of signals on a network. The better the signal-to-noise ratio of an electrical transmission system, the greater the throughput of information on the system.

The binary data being transmitted will be altered by noise and result in incorrect data received. Noise and momentary electrical disturbances may cause data to be changed as it passes through a communications channel.

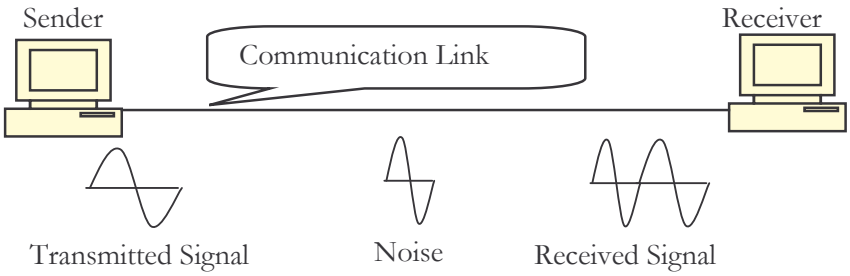


Figure 2.11 Noise

The noisy signals, which cause the data lost or corruption, are classified into different types such as : *white noise* or *thermal noise*, *induced noise*, *interference*, *crosstalk*, *impulse noise and human errors* may corrupt the signal.

White noise is present in all electronic devices and cannot be eliminated by any circuits. It increases with temperature, but it is independent of frequency. That means the white noise covers the whole frequency spectrum and will be picked up by both low and high frequency devices. As bandwidth increases, (thermal) white noise power increases. White noise is also called as thermal noise or additive noise. The amount of noise is directly proportional to the temperature of the medium. White noise usually is not a problem unless it becomes so strong that it obliterates the transmission. *Thermal noise (or additive noise)* is the random motion of electrons in a wire that created an extra signal not originally sent by the transmitter. Thermal noise is also called as additive noise. Additive noise is generated internally by components such as resistors and solid-state devices used to implement the communication system. Thermal noise is the most common impairment in a wireless communication system. There are three general sources, 1) The noise that enters the antenna with the signal, aptly called antenna noise, 2) the noise generated due to ohmic absorption in the various passive hardware components, and 3) noise produced in amplifiers through thermal action within semiconductors.

Induced noise comes from sources such as motors and appliances with coils. These devices act as a sending antenna and the transmission medium acts as a receiving antenna.

What can we do to minimize the white noise?

The medium should be kept as cool as possible

Impulse noise is a spike (a signal with high energy in a very short period of time) that comes from power lines, lightning, and so on. Impulse Noise consisting of random occurrences of energy spikes having random amplitude and spectral content. Impulse noise in a data channel can be a definitive cause of data transmission errors.

Interference is caused by picking up the unwanted electromagnetic signals nearby such as crosstalk due to adjacent cables transmitting electronic signals or lightning causing power surge.

Crosstalk is the undesired effect of one circuit (or channel) on another circuit (or channel). It occurs when one line picks up some of the signal traveling down another line. Crosstalk effect can be experienced during telephone conversations when one can hear other conversations in the background.

Crosstalk is a form of interference in which signals in one cable induce electromagnetic interference (EMI) in an adjacent cable. The twisting in twisted-pair cabling reduces the amount of crosstalk that occurs, and crosstalk can be further reduced by shielding cables or physically separating them. Crosstalk is a feature of copper cables only—fiber-optic cables do not experience crosstalk

The ability of a cable to reject crosstalk in Ethernet networks is usually measured using a scale called near-end crosstalk (NEXT). NEXT is expressed in decibels (dB), and the higher the NEXT rating of a cable, the greater its ability to reject crosstalk. A more complex scale called Power Sum NEXT (PS NEXT) is used to quantify crosstalk in high-speed Asynchronous Transfer Mode (ATM) and Gigabit Ethernet networks.

Human Error

Noise sometimes is caused by human being such as plugging or unplugging the signal cables, or power on/off the related communications equipment.

The effects of noise may be minimized by increasing the power in the transmitted signal. However, equipment and other practical constraints limit the power level in the transmitted signal. Another basic limitation is the available channel bandwidth. A bandwidth constraint is usually due to the physical limitations of the medium and the electronic components used to implement the transmitter and the receiver. These two limitations result in constraining the amount of data that can be transmitted reliably over any communications channel. Shannon's basic results relate the channel capacity to the available transmitted power and channel bandwidth.

Signal to noise ratio to quantify noise

Signal-to-noise ratio (S/N) is a parameter used to quantify how much noise there is in a signal. A high SNR means a high power signal relative to noise level, resulting in a good-quality signal. SNR is represented in decibel (db).

$$S/N = 10 \text{ Log10 } (S/N)$$

Where S = average signal power
N = noise power

Bit Error Rate

The BER (Bit Error Rate) is the probability of a signal bit being corrupted in a define time interval.

BER of 10^{-5} means on average 1 bit in 10^{-5} will be corrupted.

Note that, a BER of 10^{-5} over voice-graded line is typical and BER of less than 10^{-6} over digital communication is common.

A Bit Error Rate (BER) is a significant measure of system performance in terms of noise. A BER of 10^{-6}, for example, means that one bit of every million may be destroyed during transmission.

Several factors affect the BER:

- Bandwidth
- S/N (Signal-to-noise ratio)
- Transmission medium
- Transmission distance
- Environment
- Performance of transmitter and receiver

2.6 COMMUNICATION CHANNEL

A communications channel is a pathway over which information can be conveyed. It may be defined by a physical wire that connects communicating devices, or by a radio, laser, or other radiated energy source that has no obvious physical presence.

The communication channel provides the connection between the transmitter and the receiver. The physical channel may be a pair of wires that carry the electrical signal, or an optical fiber that carries the information on a modulated light beam, or an underwater ocean channel in which the information is transmitted acoustically, or free space over which the information-bearing signal is radiated by use of an antenna. Other media that can be characterized as communication channels are data storage media, such as magnetic tape, magnetic disks, and optical disks.

Information sent through a communications channel has a source from which the information originates, and a destination to which the information is delivered. Although information originates from a single source, there may be more than one destination, depending upon how many receive stations are linked to the channel and how much energy the transmitted signal possesses.

In a digital communications channel, the information is represented by individual data bits, which may be encapsulated into multibit message units. A byte, which consists of eight bits, is an example of a message unit that may be conveyed through a digital communications channel. A collection of bytes may itself be grouped into a frame or other higher-level message unit. Such multiple levels of encapsulation facilitate the handling of messages in a complex data communications network

2.7 Channel Capacity

Channel can be defined as a single path provided by a transmission medium via either (a) physical separation, such as by multipair cable or (b) electrical separation, such as by frequency- or time-division multiplexing.

Channel capacity Definition: The maximum bit rate that can be handled by a channel.

Channel capacity is also defined as maximum number of television channels that a cable system can carry simultaneously.

Signal-to-Noise Ratio (S/N Ratio) is a very important parameter in assessing the channel capacity or throughput of a data channel. From Shannon's Law, the maximum data rate (bit rate), which a channel can possibly support, is given by the product of the line bandwidth and the signal-to-noise ratio of the channel.

Channel capacity, shown often as "C" in communication formulas, is the amount of discrete information bits that a defined area or segment in a communications medium can hold. Thus, a telephone wire may be considered a channel in this sense.

Shannon's Law

The maximum data rate of a noisy channel whose bandwidth W Hz, and whose signal-to-noise ratio is S/N, is given by

$$\mathbf{C = W \, Log2 \, (1 + S/N)}$$

Where W = Bandwidth in Hz
S = Average signal power in watts
N = Random noise power in watts
C = Maximum data rate possible

Example

Calculate maximum data rate for telephone line, which having 30 dB signal-to-noise ratio.

Solution

Bandwidth (W) of telephone line = 3300 – 300 Hz = 3000 Hz.
S/N = 39 dB = 1000
C = 3000 × Log2 (1 + 1000)
C = 3000 × Log2 (1001)
C = 29,897 bps
C ≈ 30 Kbps

2.8 ANALOG AND DIGITAL DATA

Data that needs to be communicated may be in analog or digital form. Analog data is continuous, taking on innumerable values within a range. Voices, images, and temperature readings from a sensor are all examples of analog data. Digital data takes on a limited number of discrete values. In the limiting, and most common case, digital data takes one of two values: zero or one. Logical values such as true or false, integers, and text are commonly encountered examples of digital data.

In order to manipulate or communicate data, it must be encoded as some kind of signal, usually an electrical or electromagnetic signal. Analog data can be encoded as an analog signal. Perhaps the most common example is a plain old telephone in the local loop, though a cassette tape player, the video and audio components of a TV program, and many other household media use analog signals to represent analog data.

Analog data is also commonly encoded with digital signals. If a phone call travels beyond the local exchange carrier's central office into the long distance network, it will be

digitized. If you scan an image or capture a sound on the computer, you're converting analog data to digital signals. This analog-to-digital conversion is usually accomplished with a special device or process referred to as a codec, which is short for coder-decoder.

Digital data is routinely converted to analog signals. The most common example is when you make use of the omnipresent voice infrastructure for computer connectivity and employ a modem to represent your bits in the form of audible tones. (Modem is short for modulator-demodulator, which performs the inverse of what a codec does—though in most cases, of course, both a codec and a modem perform both analog-to-digital and digital-to-analog conversions.) Modulation can be considered to be a special case of encoding, though the terms tend to overlap in ordinary usage. Technically speaking, modulation involves combining two signals, either of which can be analog or digital, to produce a resultant signal, which can be analog or digital. Encoding, then, is the representation of data by a signal using any method.

Finally, digital data is also regularly represented by digital signals. Any time you send e-mail, load a file, or download Web pages, you're encoding digital data with digital signals.

2.9 ENCODING

In data communication system one device sends information or data to another device. In this section we considering devices used in data communication system are computers, printers, scanners, etc. This type of device stores and process information in binary format that is either 0 or 1. Whereas transmission medium used to transfer information from one end to another end may carries digital signal or analog signal. In this section we will study the encoding mechanism to convert binary data to digital signal using line coding technique and binary data to analog signal using digital modulation technique.

LINE CODING

Line coding is also called as digital to digital encoding. Digital to digital encoding or conversion is representation of digital information by a digital signal. Line coding is the process of converting binary data, a sequence of bits, to digital signal. For example, transmission of data from computer to printer. Both original data and the transmitted data are digital.

Types or Categories of Digital to Digital Encoding are:

1. Unipolar
2. Polar
3. Bipolar

Unipolar

Unipolar encoding is simple and very primitive. A digital transmission system sends voltage pulses along a medium link (wire or cable). It uses only one level of value. Polarity is assigned to one of the two binary states, usually the 1. The other state, usually the 0, is represented by zero voltage. Unipolar encoding uses only one voltage level.

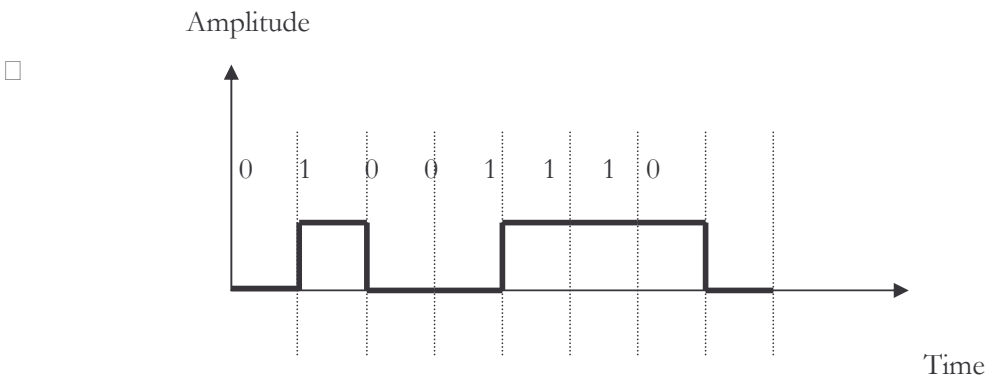


Figure 2.12 Unipolar Encoding

Polar

Polar encoding uses two voltage levels: Positive and Negative of amplitude.

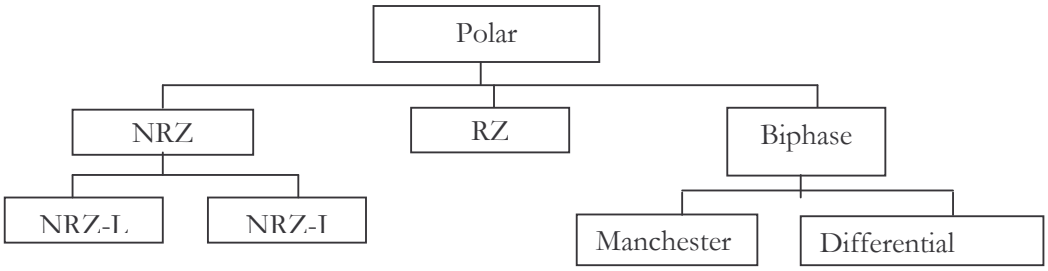


Figure 2.13 Polar techniques

A. Nonreturn to zero (NRZ): In NRZ encoding, the level of the signal is always either positive or negative. NRZ encoding is implemented in two different ways, NRZ-L or NRZ-I

In **Nonreturn to zero-level** (NRZ-L) encoding, the level of the signal is dependent upon the state of the bit. A positive voltage means the bit is a 1, and a negative voltage means the bit is a 0.

A problem can arise when the data contain a long stream of 0s or 1s. The receiver receives a continuous voltage and determines how many bits are sent by relying on its clock, which may or may not be synchronized with the sender clock.

In **Nonreturn to zero-invert** (NRZ-I): An inversion of the voltage level represents a 1 bit. A 0 bit is represented by no change. NRZ-I is also called NRZ-M, Non-return to zero-Mark.

NRZ-I is superior to NRZ-L due to the synchronization provided by the signal changes each time a 1 bit is encountered. The existence of 1s in the data stream allows receiver to synchronize its timer. A string of 0s can still cause problems.

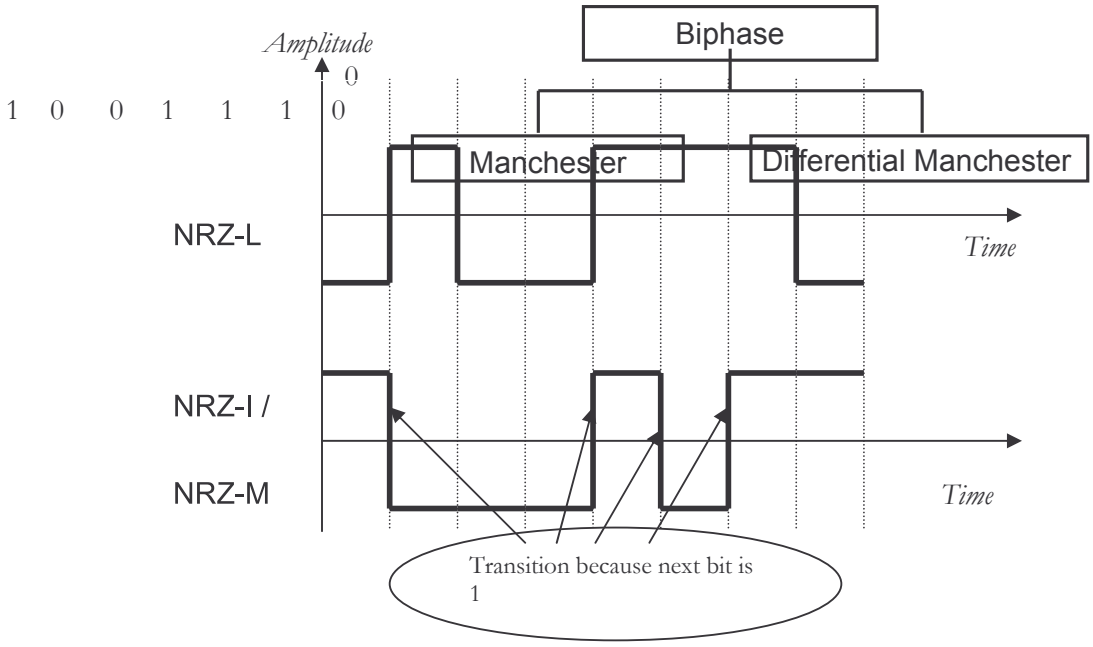


Figure 2.14 NRZ-L and NRZ-I / NRZ-M encoding

B. Return to Zero (RZ): The RZ encoding ensures signal transitions for any bit pattern.

To ensure synchronization, there must be a signal change for each bit. The receiver can use these changes to build up, update, and synchronize its clock. A good encoding digital signal must contain a provision for synchronization.

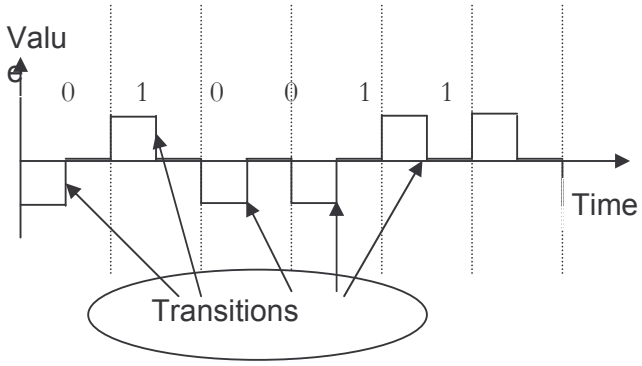


Figure 2.15 Return to Zero

The main disadvantage of RZ encoding is that it requires two signal changes to encode one bit and therefore occupies more bandwidth.

C. Biphase: In this method, the signal changes at the middle of the bit interval but does not return to zero. Instead it continues to the opposite pole. Biphase encoding is implemented in two different ways – Manchester and differential Manchester.

Figure 2.16 Biphase encoding types

1. **Manchester:** Manchester encoding uses the inversion at the middle of each bit interval for synchronization and bit representation. A negative-to-positive transition represents binary 1 and positive-to-negative transition represents binary 0.
2. **Differential Manchester:** In differential Manchester encoding, the inversion at the middle of the bit interval is used for synchronization, but the presence or absence of an additional transition at the beginning of the interval is used to identify the bit. In Differential Manchester, a transition means binary 0 and no transition means binary

1. Differential Manchester requires two signal changes to represent binary 0 but only one to represent binary 1.

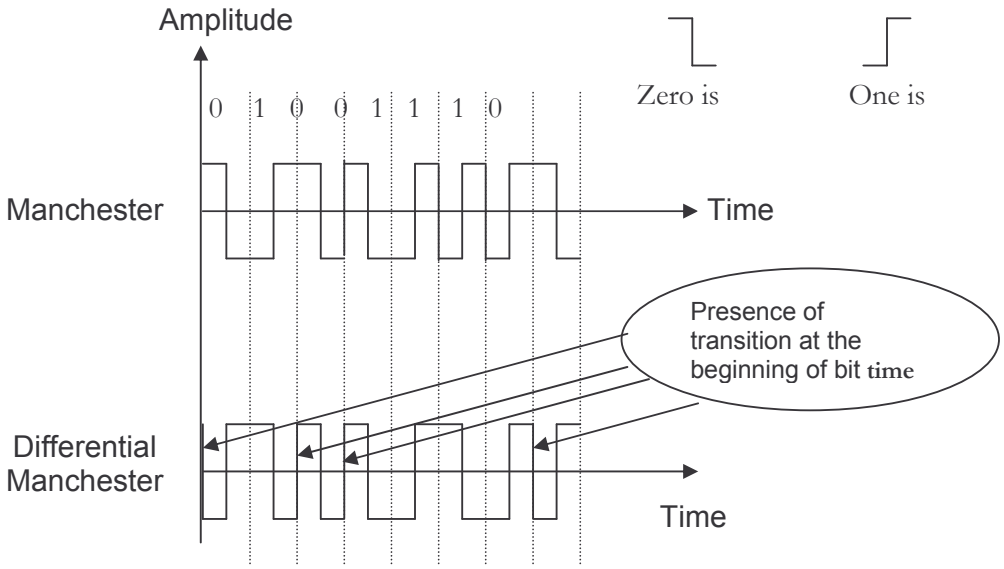


Figure 2.17 Manchester and Differential Manchester encoding

In differential Manchester encoding, the transition at the middle of the bit is used only for synchronization. The bit representation is defined by the inversion or non-inversion at the beginning of the bit.

Bipolar

Bipolar encoding, uses three voltage levels: positive, negative, and zero. The Zero level in bipolar encoding is used to represent binary 0. The 1's are represented by alternating positive and negative voltages. If the first 1 bit is represented by the positive amplitude, the second will be represented by the negative amplitude, the third by the positive amplitude, and so on.

Three types of Bipolar encoding: AMI, B8ZS and HDB3.

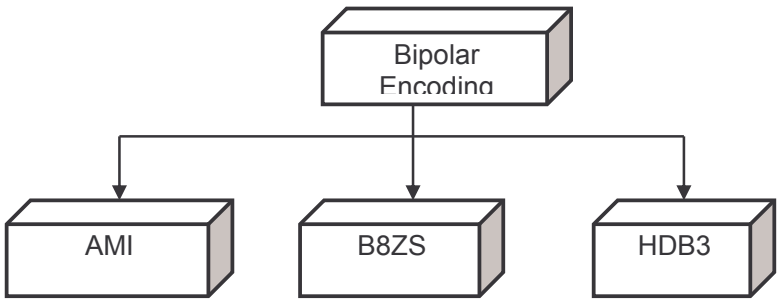


Figure 2.18 Types of Bipolar encoding

Bipolar Alternate Mark Inversion (AMI)

In the term alternate mark inversion, the word mark comes from telegraphy and means 1. So AMI means alternate 1 inversion. A neutral, zero voltage represents binary 0. Binary 1 represented by alternating positive and negative voltages.

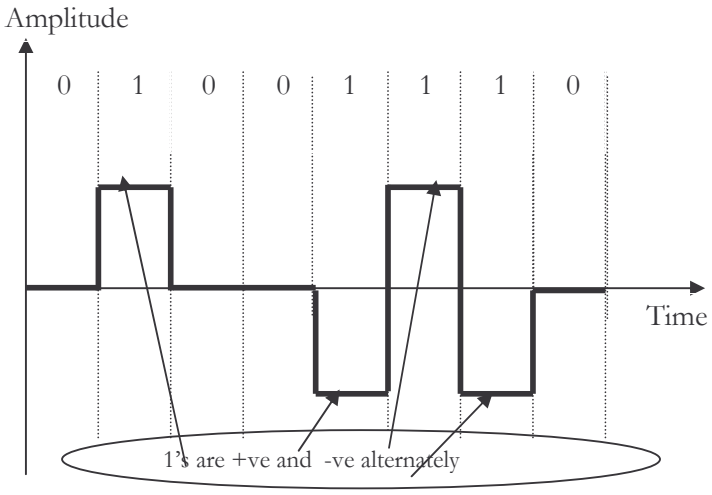


Figure 2.19 AMI

Bipolar 8-Zero Substitution (B8ZS)

Bipolar AMI changes poles with every 1 it encounters. But the signal does not change during a string of 0s. The solution provided by B8ZS is to force artificial signal changes, called violations, within the 0 string. Anytime eight 0s occur in succession, B8ZS introduces changes in the pattern based on the polarity of the previous 1.

B8ZS encoding is used in North America.

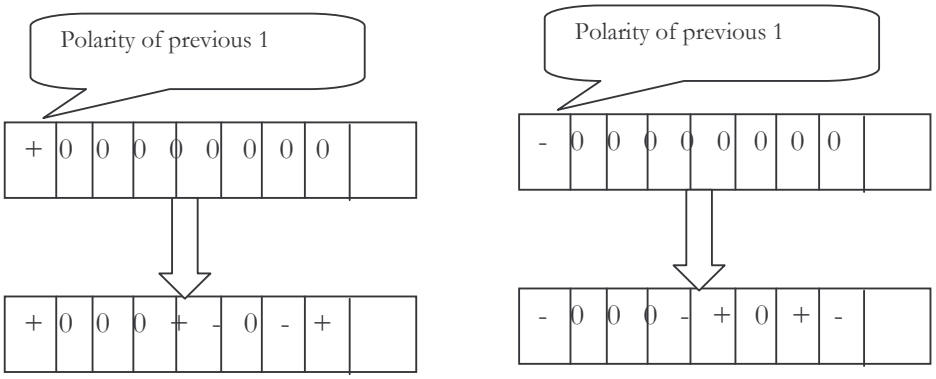


Figure 2.20 B8ZS Encoding

1. High-Density Bipolar 3 (HDB3): Strings of consecutive 0s is solved differently in Europe and Japan than in the United States. This conversion, called HDB3, introduces changes into the bipolar AMI pattern every time four consecutive 0s are encountered instead of waiting for the eight expected by B8ZS in North America.

In HDB3 if four 0s come one after another, we change the pattern in one of four ways based on the polarity of the previous 1 and the number of 1s since the last substitution.

Figure 2.21 (a) If the number of 1s since the last substitution is odd.

Figure 2.21 (b) If the number of 1s since the last substitution is even

Figure 2.21 HDB3

DIGITAL MODULATION METHODS

Modulation of binary data or digital-to-analog modulation is the process of changing one of the characteristics of an analog signal based on the information in a digital signal (0s and 1s). When we transmit data from one computer to another across a public access phone line, for example, the original data are digital, but because telephone wires carry analog signal, the data must be converted. The digital data must be modulated on an analog signal that has been manipulated to look like two distinct values corresponding to binary 1 and binary 0.

Two terms used frequently in data communication are bit rate and baud rate. Bit rate is the number of bits transmitted during 1s. Baud rate refers to the number of signal units per second that are required to represent those bits. A signal unit is composed of one or more bits. Bit rate is the number of bits per second. Baud rate is the number of signal units per second. Baud rate is always less than or equal to the bit rate.

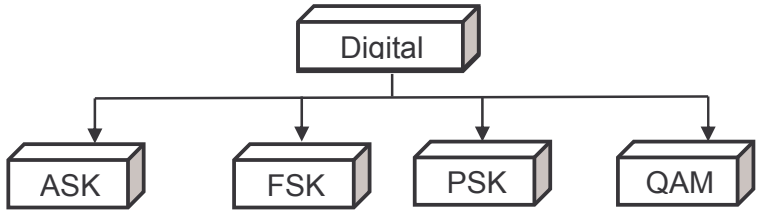


Figure 2.22: Digital Modulation Methods

An analogy can clarify the concept of baud and bits. In transportation, a baud is analogous to a car, and a bit is analogous to passenger. A car can carry one or more passengers. If 2000 cars go from one location to another, carrying only one passenger, then 2000 passengers are transported. However, if each car carries two passengers, then 4000 passengers are transported. Note tat number of cars, not the number of passengers, determines the traffic and therefore, the need for wider highways. Similarly, the number of bauds determines the required bandwidth, not the number of bits.

Amplitude Shift Keying (ASK)

In ASK the strength of the carrier signal is varied to represent binary 1 or 0. In ASK both the frequency and phase remain constant while the amplitude changes.

Which voltage represents 1 and which represents 0 is left to the system designers. Unfortunately, ASK transmission is highly susceptible to noise interference. The term noise refers to unintentional voltage introduced onto a line by various phenomena such as heat or electromagnetic indication created by other sources. These unintentional voltages combine with the signal to change the amplitude. A 0 can be changed to 1, and a 1 to 0.

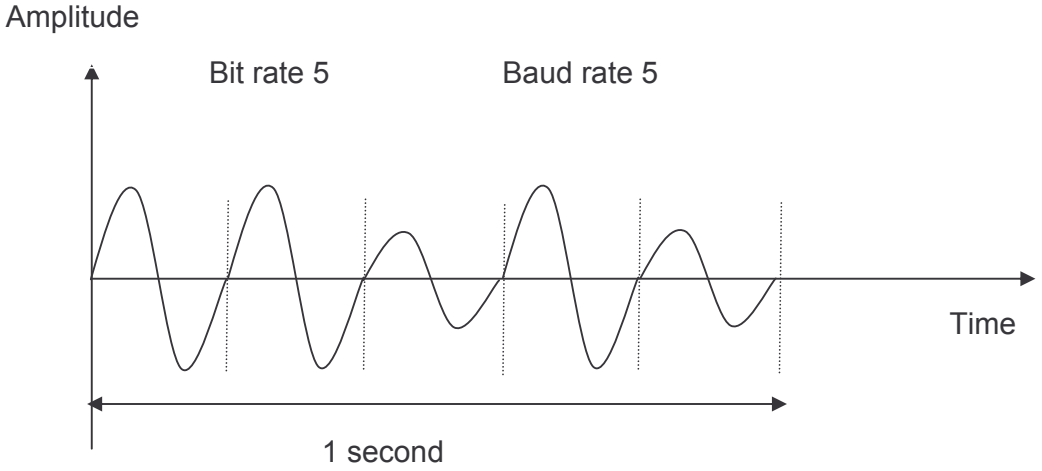


Figure 2.23 Amplitude Shift Key

A popular ASK technique is called on-off keying (OOK). In OOK one of the bit values is represented by no voltage. The advantage is reduction in the amount of energy required to transmit information. Unfortunately, ASK transmission is highly susceptible to noise interference.

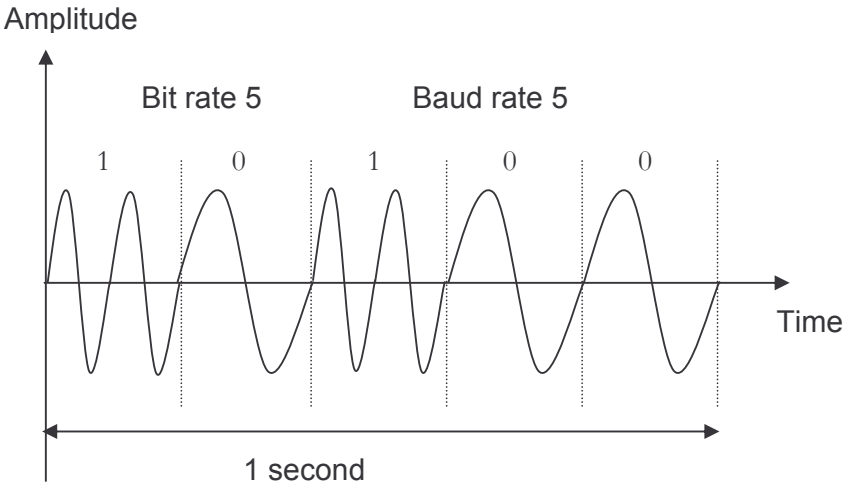


Figure 2.24 Frequency Shift Key

Frequency Shift Keying

In FSK the frequency of the carrier signal is varied to represent binary 1 or 0. The frequency of the signal during each bit duration is constant.

In FSK, value of frequency depends on the bit 1 or 0. In FSK both the amplitude and phase remains constant.

In FSK, two fixed amplitude carrier signal are used, one for a binary 0 and the other for a binary 1. The different between the two carriers is known as the frequency shift.

FSK avoids most of the noise problem of ASK. Because the receiving device is looking for specific frequency changes over a given number of periods, it can ignore voltage spikes. The limiting factors of FSK are the physical capabilities of the carrier.

Phase Shift Keying (PSK)

In PSK the phase of the carrier is varied to represent binary 1 or 0. For example, if we start with phase of 0 degrees to represent binary 0, then we can change the phase to 180 degrees to send binary 1. The above method is often called 2-PSK, or binary PSK, because two different phases (0 and 180 degrees) are used.

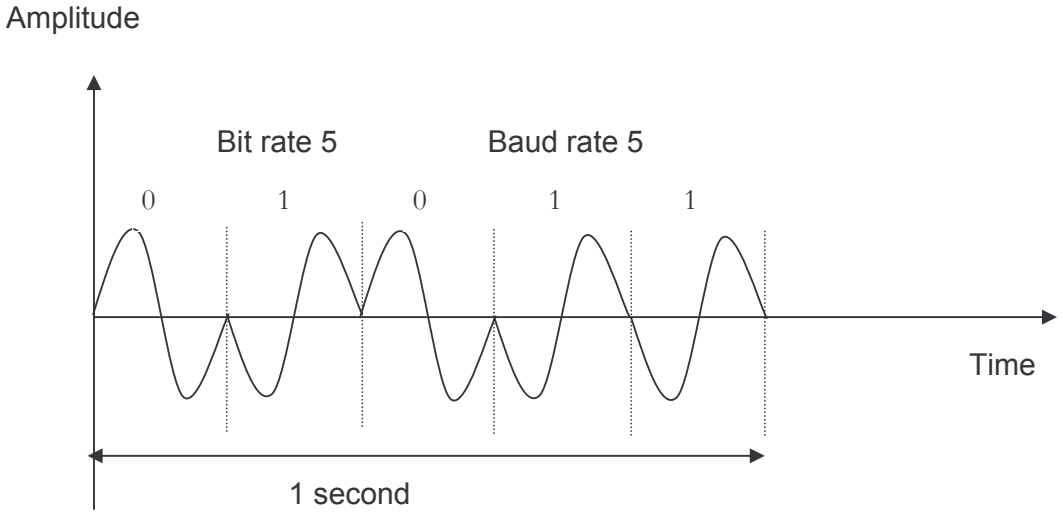


Figure 2.25 Phase Shift Key

Bit	Phase
0	0
1	180

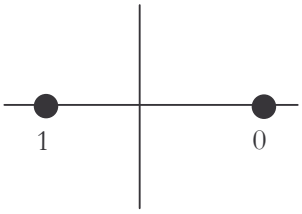


Figure 2.26 Constellation or Phase-state diagram

In PSK, both the amplitude and frequency remains constant as the phase changes. PSK is not susceptible to the noise degradation that mostly affects ASK, nor to the bandwidth limitations of FSK.

QAM (Quadrature Amplitude Modulation)

So far, we have been altering only one of the three characteristics (amplitude, frequency, and phase) of a sine wave at a time, but what if we alter two ?

FSK having bandwidth limitation so combining it with other is practically useless. But why not combine ASK and PSK ? Then we have *x* variation in amplitude and *y* variation in phase, giving us *x* times *y* possible variations.

QAM means combining ASK and PSK in such a way that we have maximum contrast between each bit, dibit, tribit, and so on.

Possible variations of QAM are numerous. Theoretically, any measurable number of changes in amplitude can be combined with any measurable number of changes in phase. For example : 4-QAM or 8-QAM. In 4-QAM, two-amplitude change and 2 phase shift as shown in figure. In 8-QAM, 2-amplitude change and 4 phase shift. In 8-QAM number of amplitude shifts is less than number of phase shifts. Because amplitude changes are susceptible to noise and require greater shift differences than do phase changes, the number of phase shifts used by a QAM system always larger than the number of amplitude shifts.

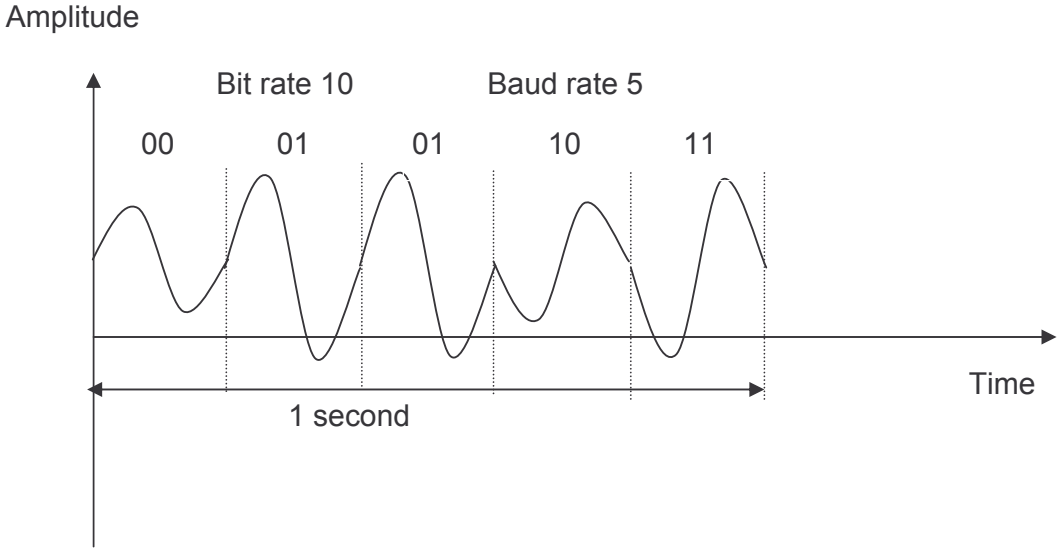


Figure 2.27 4-QAM (2 amplitudes, 2 phases)

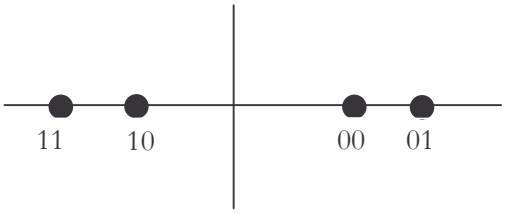


Figure 2.28 Constellation Diagram of 4-QAM

2.10 DTE-DCE DEVICES

Two terms you should be familiar with are DTE and DCE. DTE stands for Data Terminal Equipment, and DCE stands for Data Communications Equipment. These terms are used to indicate the pin-out for the connectors on a device and the direction of the signals on the pins. Your computer is a DTE device, while most other devices are usually DCE devices.

Data Terminal Equipment (DTE)

Any device that is a source of data transmission over a serial telecommunications link. Typically, data terminal equipment (DTE) can be a computer, a terminal, a router, an access server, or some similar device. The earliest form of DTE was the Teletype machine.

The term "DTE" specifically refers to a device that uses serial transmission such as the transmissions involving the serial port of a computer. Most serial interface devices

contain a chip called a universal asynchronous receiver-transmitter (UART) that can translate the synchronous parallel data transmission that occurs within the computer's system bus into an asynchronous serial transmission for communication through the serial port.

To connect a DTE to a telecommunications link, you use data communications equipment (DCE). The DCE provides termination for the telecommunications link and an interface for connecting the DTE to the link. An example of a DCE for connecting a DTE to the local loop Plain Old Telephone Service (POTS) connection is a modem.

Data Communications Equipment (DCE)

Any device that supports data transmission over a serial telecommunications link. Typically, data communications equipment (DCE) refers to modems, Channel Service Unit/Data Service Units (CSU/DSUs), multiplexers, and similar devices. The purpose of a DCE is to provide termination for the telecommunications link and an interface for connecting data terminal equipment (DTE) to the link.

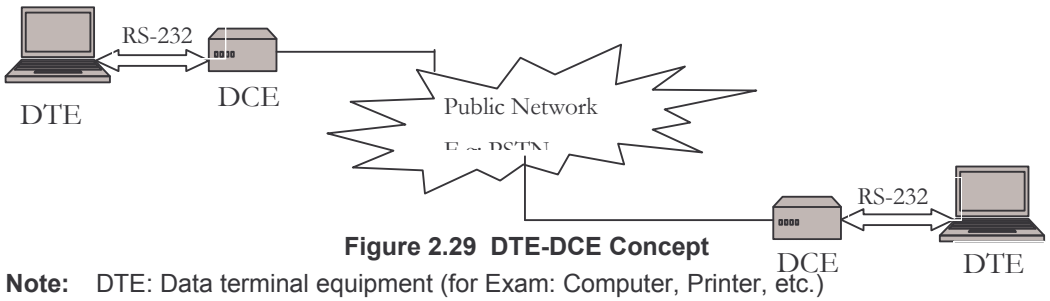


Figure 2.29 DTE-DCE Concept

Note: DTE: Data terminal equipment (for Exam: Computer, Printer, etc.)
DCE: Data circuit-terminating equipment (for Exam: Modem)

The term "DCE" specifically refers to serial transmission, which generally occurs over links such as a local loop Plain Old Telephone Service (POTS) connection, an Integrated Services Digital Network (ISDN) line, or a T1 line. An example of a DCE is an analog modem, which provides a connection between a computer (the DTE) and the local loop POTS phone line (the serial transmission line). A DCE accepts a stream of serial data from a DTE and converts it to a form that is suitable for the particular transmission line medium being used. The DCE also works in reverse, converting data from the transmission line to a form the DTE can use.

The EIA (Electronic Industries Association) and the ITU-T (International Telecommunication Union-Telecommunication Standards Committee) have been involved in developing DTE-DCE interface standards. EIA standards are called EIA-232, EIA-442, and so on. The ITU-T standards are called the V-series and the X series.

Transmission Rate

Every line has an upper limit and a lower limit on the frequencies of the signals it can carry. This limited range is called the bandwidth.

A telephone line has a bandwidth of 3000Hz. The telephone line has bandwidth of 2400 Hz for data transmission.

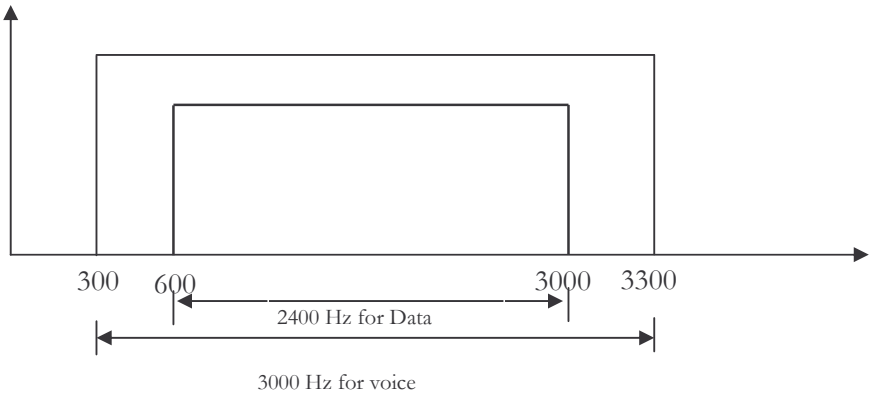


Figure 2.30 Bandwidth of Telephone line

2.11 MODEM

The devices (computers) that generate the digital data (DTE) usually generate a sequence of digital pulses which are not suitable for transmission on a medium. Typically there is another device - a **modem** (DCE) which prepares this signal for the transmission medium.

Modem stands for modulator/demodulator. A *modulator* converts a digital signal into an analog signal using ASK, FSK, PSK, or QAM appropriate for telephone lines. A *demodulator* converts an analog signal into a digital signal.

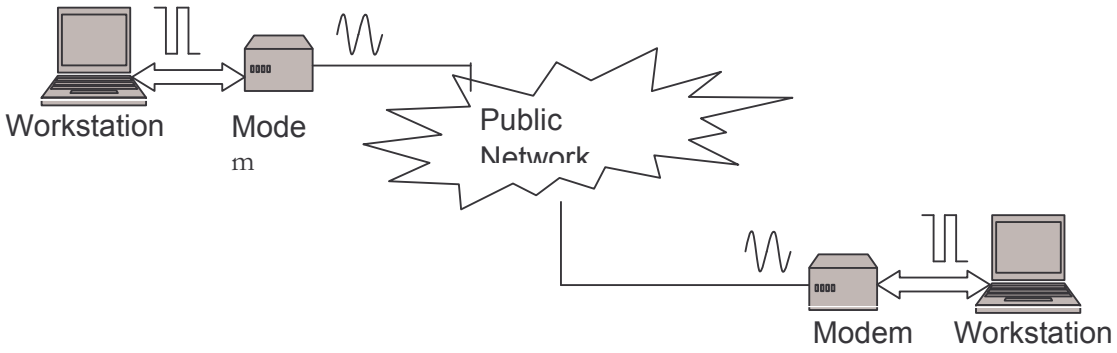


Figure 2.31 Modem

The two PCs at the end are the DTEs; the modems are the DCEs. The DTE creates a digital signal and relay it to the modem via an interface (like the EIA 232). The modulated signal is received by demodulation function of second modem. It decodes it and then relays the resulting digital signal to the receiving computer via an interface.

Generally, modem is any type of data communications equipment (DCE) that enables digital data transmission over the analog Public Switched Telephone Network (PSTN). The term "modem" (which actually stands for "modulator/demodulator") is usually reserved for analog modems, which interface, through a serial transmission connection such as the RS-232 interface, with data terminal equipment (DTE) such as computers. The modem converts the digital signal coming from the computer into an analog signal that can be carried over a Plain Old Telephone Service (POTS) line. The term "digital modem" is sometimes used for ISDN terminal adapters.

Modems were developed in the 1960s by Bell Labs, which developed a series of standards called the Bell Standards. These standards defined modem technologies of up to a 9600-bps transmission speed. But after the breakup of Bell Telephone, the task of developing modem standards was taken over by the International Telegraph and Telephone Consultative Committee (CCITT), which is now called the International Telecommunication Union (ITU). According to ITU specifications, modem standards are classified by a series of specifications known as the V series. The International Telecommunication Union (ITU), which defines standards of up to V.90 (which supports 56-Kbps downloads and 33.6-Kbps uploads).

Modems generally have two interfaces:

- An RS-232 serial transmission interface for connecting to the DTE, usually the computer
- An RJ-11 telephone interface for connecting to the 4-wire PSTN telephone outlet in the local loop connection

Physical types of modem types include the following :

- Internal modems, which are installed as interface cards inside the computer and might use some of the machine's CPU processing power for functions such as encoding and data compression.
- External modems, which are generally more expensive and connect to the serial port on the computer using a DB9 or DB25 connector. External modems are useful when several users need to share a modem.
- PCMCIA modems, which are credit-card-sized modems for laptop computers used by mobile workers.
- Voice/data/fax modems, which can be used for file transfer, sending and receiving faxes, and voice mail using associated software.

Logical types of modem

- Asynchronous and synchronous

  Low speed modems are designed to operate asynchronously. Each data frame conforms an asynchronous transmission mechanism. High-speed modems as well as leased-lines modems use synchronous transmission. The two modems use a common time base and operate continuously at substantially that same frequency and phase relationship by circuit that monitor the connection.

- Half duplex and Full duplex

  A half-duplex modem must alternately sends and receives signals. Half-duplex allows more of the channel bandwidth to be put to use but slows data communications. A full-duplex modem can simultaneously handle two signals using two carriers to transmit and receive data. Each carrier uses half of the bandwidth available to it and its modulation.

Digital Modem

Any type of modem used for synchronous transmission of data over circuit-switched digital lines. One example of a digital modem is an ISDN terminal adapter. Digital modems are not used for changing analog signals into digital signals because they operate on end-to-end digital services. Instead, they use advanced digital modulation techniques for changing data frames from a network into a format suitable for transmission over a digital line such as an Integrated Services Digital Network (ISDN) line. They are basically data framing devices, rather than signal modulators.

Analog Modem

A modem used for asynchronous transmission of data over Plain Old Telephone Service (POTS) lines. Analog modems are still a popular component for remote communication between users and remote networks. The word "modem" stands for "modulator/demodulator," which refers to the fact that modems convert digital transmission signals to analog signals and vice versa. For example, in transmission, an analog modem converts the digital signals it receives from the local computer into audible analog signals that can be carried as electrical impulses over POTS to a destination computer or network. To transmit data over a telephone channel, the modem modulates the incoming digital signal to a frequency within the carrying range of analog phone lines (between 300 Hz and 3.3 kHz). To accomplish this, multiplexing of the digital signal from the computer with a carrier signal is performed. The resulting modulated signal is transmitted into the local loop and transmitted to the remote station where a similar modem demodulates it into a digital signal suitable for the remote computer.

SUMMARY

Simplex is a form of communication in which signals are sent in only one direction. This is different from duplex transmission, in which signals can simultaneously be sent and received by a station, and from half-duplex transmission, in which signals can be sent or received but not both at the same time. Simplex transmission occurs in many common communication applications, the most obvious being broadcast and cable television. It is not used in true network communication because stations on a network generally need to communicate both ways.

Transmission is the act of propagation through the medium and receiving and processing of the signal. Transmission media are not perfect. The imperfections cause impairment in the signal sent through the medium. This means that the signal at the beginning and end of the medium are not the same. What is sent is not what is received.

Modem stands for modulator/demodulator. Generally, modem is any type of data communications equipment (DCE) that enables digital data transmission over the analog Public Switched Telephone Network (PSTN).

PRACTICE SET

Review Questions

1. What is a bit rate?
2. What is a baud rate?
3. Name three types of transmission impairment.
4. Write a short note on Thermal noise and Impulse noise.
5. What is a crosstalk?
6. Write a short note on MODEM.
7. What is the bandwidth of telephone line to carry voice?
8. What are different types of line coding?
9. What are different types of Digital to analog modulation technique?
10. What are different types of polar encoding?
11. What are different types of bipolar technique?
12. What are different types serial transmission
13. What are different type's data transmission techniques?
14. What are different type's data transfer methods of communication link?

Multiple Choice Questions

1. Unipolar, bipolar, and polar encoding are types of …….. encoding.
   A) Line B) Block C) NRZ D) Manchester

2. …… encoding has a transmission at the middle of each bit.
   A) RZ B) Manchester C) Differential Manchester D) All the above

3. ……. encoding has a transmission at the beginning of each bit.
   A) RZ B) Manchester C) Differential Manchester D) All the above

4. Which encoding type always has a nonzero average amplitude?
   A) Unipolar B) Polar C) Bipolar D) All the above

5. Which encoding uses alternating positive and negative values for 1s?
   A) NRZ-L B) NRZ-I C) RZ D) Manchester

6. RZ involves …….. signal levels.
   A] Two B) Three C] Four D] Five

7. In …….transmission, bits are transmitted simultaneously, each across its own wire.
   A] Asynchronous serial B] Synchronous serial
   C] Parallel D] Both option A) and B)

8. In …… transmission, bits are transmitted over a single wire, one at a time.
   A) Asynchronous serial B) Synchronous serial
   C) Parallel D) Both option A) and B)

9. In …… transmission, a serial bit and a stop bit frame a character byte.
   A) Asynchronous serial B) Synchronous serial
   C) Parallel D) Both option A) and B)

10. Synchronous transmission does not have ……..
    A) start bit B) stop bit C) gaps between bytes D) All the above

11. ASK,FSK,PSK and QAM are examples of …….. modulation
    A] Digital-to-digital C] Digital-to analog
    C] Analog-to digital D] Analog –to analog

12. IN QAM, both phase and ---------- of a carrier frequency are varied.
    A] Amplitude B] Frequency C] Bit rate D] Baud Rate

13. Which of the following is most affected by noise?
    A] PSK B] FSK C] ASK D] QAM

14. If a baud rate is 400 for a PSK signal, the bit rate is ……bps
    A] 100 B] 400 C] 800 D] 1600

15. If the bit rate for an ASK signal is 1000 bps, the baud rate is ----------
    A] 100 B] 500 C] 1000 D] 2000

16. If the bit rate for an FSK signal is 1000 bps, the baud rate is ----------
    A] 100 B] 500 C] 1000 D] 2000

17. If the bit rate for a QAM signal is 3000 bps and a signal unit is represented by a tribit (3-bits), what is the baud rate?
    A] 300 B] 400 C] 1000 D] 1200

18. If the baud rate for a QAM signal is 3000 and a signal unit is represented by a tribit (3-bits), what is the bit rate?
    A] 300 B] 400 C] 1000 D] 9000

19. If the baud rate for a QAM signal is 1800 and bit rate is 9000. How many bits are there per signal unit?
    A] 3 B] 4 C] 5 D] 6

20. If the baud rate for a QAM signal is 1000 and bit rate is 3000. How many bits are there per signal unit?
    A] 2 B] 3 C] 4 D] 5

21. Which modulation technique involves tribits, eight different phase shifts, and one amplitude?
A] FSK B] 8-PSK C] ASK D] 4-PSK

22. Modulation of an analog signal can be accomplished through changing the ……. Of the carrier signal.
A] Amplitude B] Frequency C] Phase D] Any of the above.

23. For the telephone line, the bandwidth for the voice is usually ……. The bandwidth for data.
A] equivalent to B] Less than C] Greater than D] Twice

24. For the telephone line, the bandwidth for the data is usually ……. the bandwidth for voice.
A] equivalent to B] Less than C] Greater than D] Twice

25. The bit rate always equals the baud rate in which type of signals?
A] FSK B] 4-QAM C] 4-PSK D] All the above.

26. …… transmission refers to one direction data transfers all the time.
A] Simplex B] Half-Duplex C] Full-Duplex D] None of the above

27. …… transmission refers to one direction data transfers but one at a time.
A] Simplex B] Half-Duplex C] Full-Duplex D] None of the above

28. …… transmission refers to both direction data transfers all the time.
A] Simplex B] Half-Duplex C] Full-Duplex D] All of the above

29. …….. stands for Modulator/Demodulator
A] Bit rate B] Baud Rate C] MODEM D] None of the above

30. Impulse noise is the type of ……
A] Attenuation B] Distortion C] Noise D] None of the above

* * *

Chapter 3 Error Detection and Correction

Errors are all around us. We hate them, but we make them all the time, we expect them, and we try to learn to live with them. We say “to err is human," but it is not only humans who err. Tools, instruments, and machines of all kinds also misbehave or err, bite back, break down from time to time. One area where errors are particularly critical is data processing and communications. One bad bit in a computer program can completely corrupt the program. Similarly, the smallest error in a data file can change the meaning of the data in a crucial way. Fortunately, it is possible to detect, and often correct errors in data and data communication.

OBJECTIVES OF THIS CHAPTER

After reading this chapter, the student will be able to:

- determine how errors in asynchronous digital data transmissions are detected using parity.
- determine how errors in asynchronous digital data transmissions are corrected using two-dimensional parity check.
- determine how errors in synchronous digital data transmissions are detected using checksum and CRC.
- determine how errors in synchronous digital data transmissions are corrected using Hamming Code.
- determine how errors are corrected using automatic request for retransmission.

3.1 INTRODUCTION

Every time information is transmitted, on any channel, it may get corrupted by noise. In fact, even when information is stored in a storage device, errors may suddenly occur, because no piece of hardware is absolutely reliable. This also applies to non-computer information. Printed information may fade with time and may deteriorate from high use. Speech sent through the air may deteriorate due to noise, wind, and variations in temperature. Speech, in fact, is a good starting point for understanding the principles of error-detecting and error-correcting codes. Imagine a noisy cocktail party where everybody talks simultaneously, on top of blaring music. We know that even in such a situation it is possible to carry on a conversation, except that more attention than usual is needed.

Data processing and transmission systems use a variety of techniques to detect and correct errors that occur, usually for any of the following reasons:

- Electrostatic interference from nearby machines or circuits
- Attenuation of the signal caused by a resistance to current in a cable
- Distortion due to inductance and capacitance
- Loss in transmission due to leakages
- Impulses from static in the atmosphere

Most of the LAN technologies and optical cable networks reduce errors considerably. Wireless networks and WAN links can have high error rates. The occurrence of a data bit error in a serial stream of digital data is an infrequent occurrence. Even less frequent is the experience of numerous errors within the transmission of a single message. If a

number of errors occur then it is presumed that either a significant interference occurred affecting the transmission line or that there is a major failure in the communications path. Largely because of the extremely low bit-error rates in data transmissions, most error detection methods and algorithms are designed to address the detection or correction of a single bit error. However, as we shall soon see, many of these methods will also detect multiple errors. Error correction, though, will remain a one-bit error concern.

Bit errors are errors that corrupt bits during transmission, turning a 1 into a 0, or 0 into a 1. These errors are caused by power surges and other interference. *Packet errors* occur when packets are lost or corrupted. Packet loss can occur during times of network congestion when buffers become full and network devices start discarding packets. Errors and packet loss also occur during network link failures. There are two types of errors namely, single bit error and burst errors

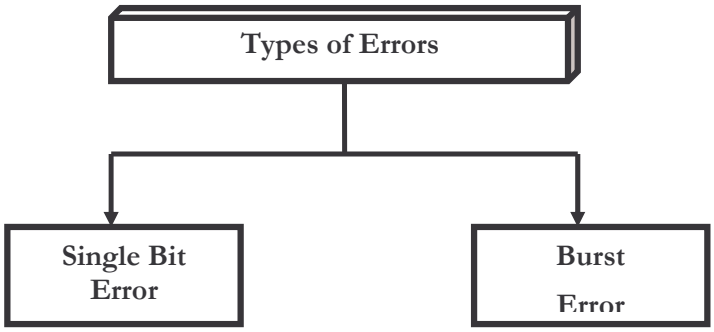


Figure 3.1 Types of errors

Single-Bit Error

The term single-bit error means that 1 bit of a message is changed from 0 to 1 or from 1 to 0 during transmission.

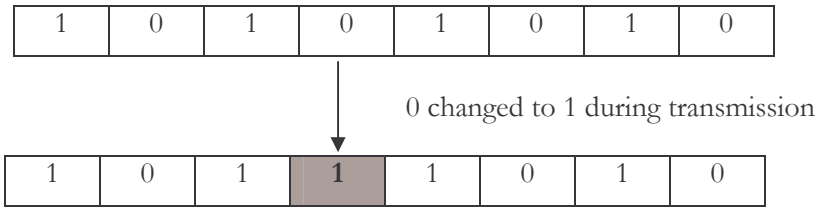


Figure 3.2 Single-bit error

Burst Error

The term burst error means that 2 or more bits of a message changed from 0 to 1 or from 1 to 0 during transmission. Burst error does not mean that the error occurred in consecutive bits. It might be possible some bits in between not been corrupted. The length of burst is measured from the first changed bit to the last changed/corrupted bit.

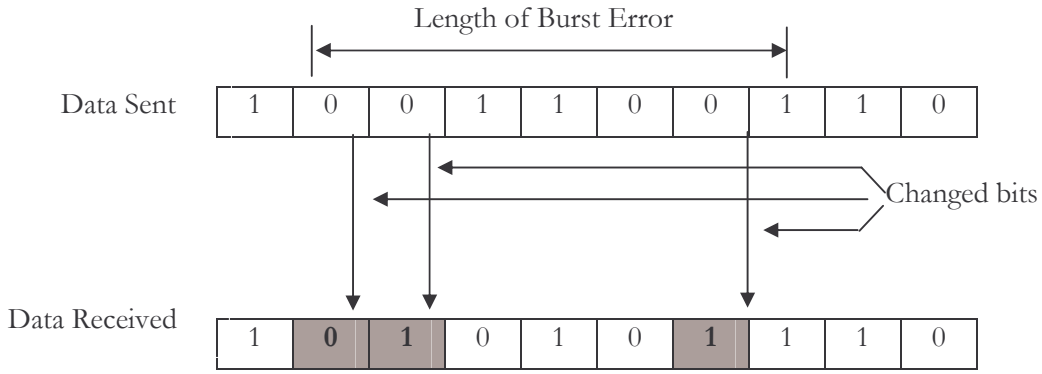


Figure 3.3 Burst error of length 6

In practice, bits are sent on a wire as voltages. A binary 0 may, e.g., be represented by any voltage in the range 3-25 volts. A binary 1 may similarly be represented by the voltage range of — 25v to — 3v. Such voltages tend to drop over long lines and have to be amplified periodically. In the telephone network there is an amplifier (a *repeater)* every 20 miles or so. It looks at every bit received, decides if it is a 0 or a 1 by measuring the voltage, and sends it to the next repeater as a clean, fresh pulse. If the voltage has deteriorated enough in passage, the repeater may make a wrong decision when sensing it, which introduces an error into the transmission. At present, typical transmission lines have error rates of about one in a billion but, under extreme conditions—such as in a lightning storm, or when the electric power suddenly fluctuates—the error rate may suddenly increase, creating a burst of errors.

3.2 ASYNCHRONOUS DATA ERROR DETECTION METHODS

Probably the most common and oldest method of error detection is the use of **parity.** While parity is used in both asynchronous and synchronous data streams, it finds greater use in low-speed asynchronous transmission applications; however, its use is not exclusive to this.

PARITY ERROR DETECTION

Parity works by adding an additional bit to each character (word) transmitted. The state of this bit is determined by a combination of factors, the first of which is the type of parity system employed. The two types are even and odd parity.

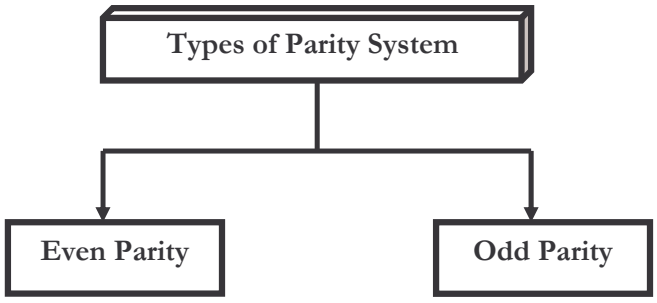
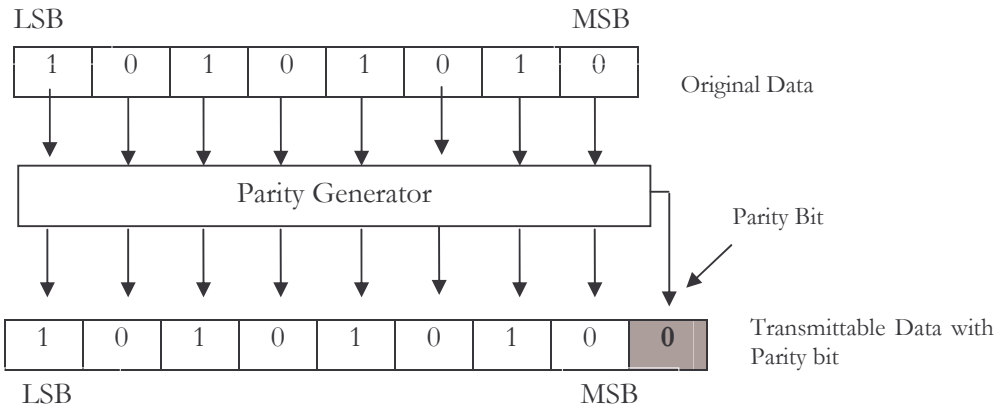


Figure 3.4 Types of Parity system

The second factor is the number of logic 1 bits in the data character. In an even parity system, the parity bit is set to a low state if the number of logic 1s in the data word is even. If the count is odd, then the parity bit is set high. For an odd parity system, the state of the parity bit is reversed. For an odd count, the bit is set low, and for an even count, it is set high.

To detect data errors, each character word that is sent has a parity bit computed for it and appended after the last bit of each character is sent as illustrated in Figure 3.5. At the receiving site, parity bits are recalculated for each received character. The parity bits sent with each character are compared to the parity bits the receiver computes. If their states do not match, then an error has occurred. If the states do match, then the character *may* be error-free.



Note: MSB – Most Significant Bit

LSB – Least Significant Bit

Figure 3.5 Appending Parity Bit

Example 1

What is the state of the parity bit for both an odd and an even parity system for the extended ASCII character A?

Solution

The extended ASCII character 'A' has a bit pattern of 0100 0001 (41 H). The number of logic 1s in that pattern is two, which is an even count. For an even parity system, the parity bit would be set low so that the total number of 1's in transmittable data unit including parity bit will become even and for an odd parity system, it would be set high so that the total number of 1's in transmittable data unit including parity bit will become odd.

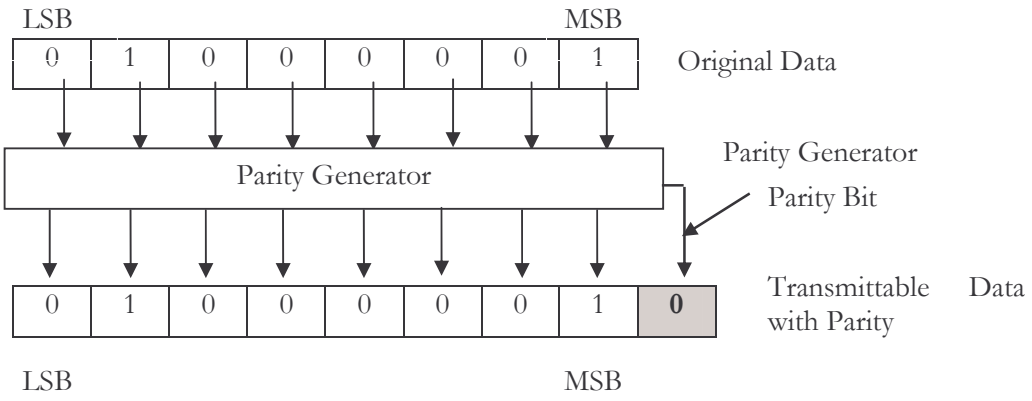


Figure 3.6 (a) Even Parity for ASCII character A

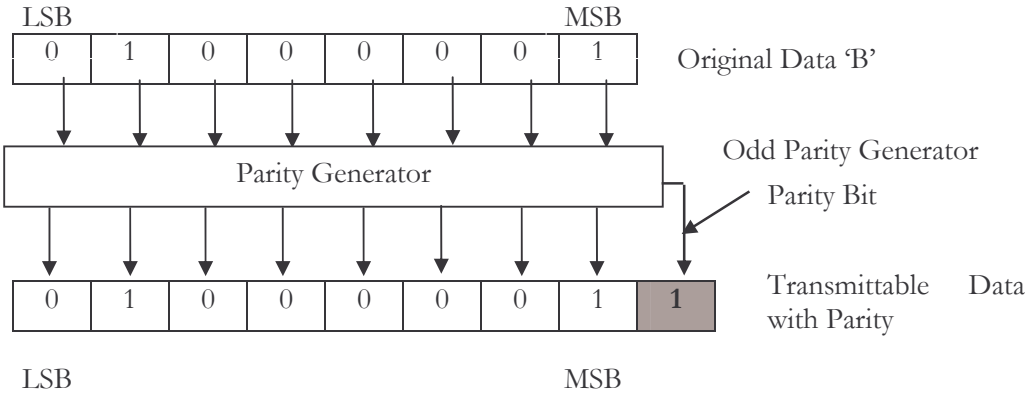


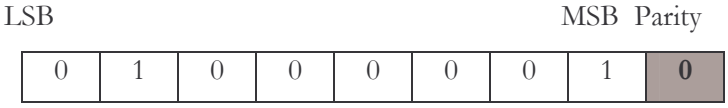
Figure 3.6 (b) Odd Parity for ASCII character A

Example 2

The ASCII character A (0100 0001 = 41h) is transmitted with an even-parity bit appended to it. Illustrate how the receiver would detect an error.

Solution

As shown in Figure 3.6 *(a),* the state of the even-parity bit for the ASCII A is low, so the complete data stream for the character sent, starting with the least significant bit (LSB) is: 010000010. Notice there is now nine bits - eight bits for the extended ASCII character A and one for the parity bit. The breakdown of the data stream is:



Suppose that the LSB becomes corrupted during transmission. The receiver receives the character as: 110000010. When the receiver computes a parity bit for the character data, it results in a high state of the parity. This is compared with the transmitted parity, which is a low state. Since they do no agree, the receiver determines that an error has occurred. Note that the receiver cannot determine which bit is bad, only that one of them is wrong.

A match between transmitted parity and receiver-calculated parity does not guarantee that the data has not been corrupted. Indeed, if an even number of errors occurs in a single character, then the parity for the corrupted data will be the same state as the good data. This does not present a major problem, since the occurrence of two errors in an eight-bit character is excessive and usually indicates a major problem in the system. Such a problem would cause errors to occur in other characters and one of them would eventually be detected.

Parity Generation

The hardware circuit used to generate the state of the parity bit is composed of a number of exclusive OR gates.

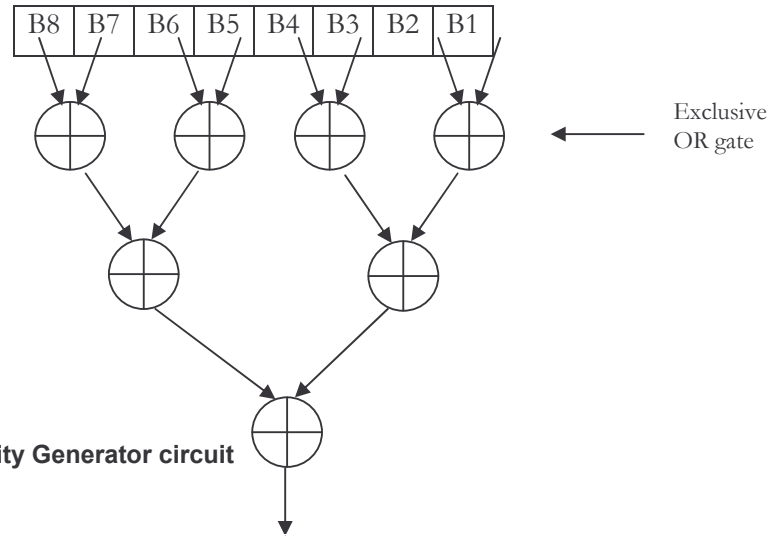


Figure 3.7 Parity Generator circuit

A	B	Result
0	0	0
0	1	1
1	0	1
1	1	0

Table 3.1 Exclusive-OR Truth Table

The figure 3.8 shows the example of even parity generator circuit.

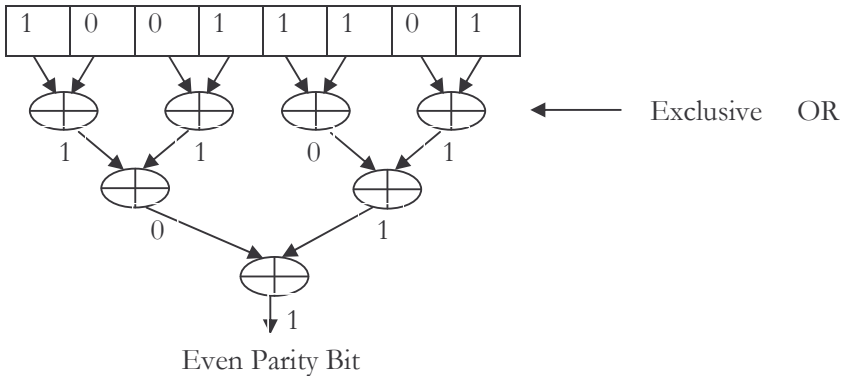


Figure 3.8 Example of even Parity generator

3.3 SYNCHRONOUS DATA ERROR METHODS

Synchronous data are transmitted at higher data rates in as an efficient manner as possible. Start-and-stop framing and parity bits are omitted from the data stream to reduce overhead. **Overhead** is defined as any bits sent that do not contain actual data information. This includes framing bits, preambles, error-detection characters, or bits, etc. It should be noted that in some synchronous data systems, parity is occasionally employed for error detection. Most high-speed synchronous transmissions, however, do not follow this practice. The reason is that most errors in high-speed transmissions occur in bursts, which could render parity-error detection less effective. These error bursts

result from some external interference or other effect on the line that causes several bits to be corrupted at once. Single-bit errors occur less frequently. The duration of noise is normally longer than the duration of one bit, which means that when noise affects data, it affects set of bits. The number of affected bits depends on the data rate and duration of noise. For example, if we are sending data at 1 Mbps, a noise of 1/100 second can affect 10,000 bits. Because of this, and the desire to reduce the overhead in synchronous transmissions, error-detection methods have evolved to detect single and multiple errors within a data stream.

Synchronous error detection works by creating an additional character to be sent with the data stream. At the receive site, the process is duplicated and the two error detection characters are compared similarly to comparing two parity bits. If the characters match then the data received has no errors. If they do not match, an error has occurred and the message has to be retransmitted. Note that one major difference between using error-detection characters versus single-parity bits, is that if the transmitted and received characters match, then the data is good. Using parity, matching parity bits does not guarantee that the character received was good. The computation of error characters is carried on quickly to support the higher data rates of transmission.

CYCLIC REDUNDANCY CHECK (CRC)

One of the most frequently used error-detection methods for synchronous data transmissions is **cyclic redundancy check (CRC)** developed by IBM. This method uses a pseudo-binary-division process to create the error or CRC character, which is appended to the end of the message. The hardware circuitry that generates the CRC character at the transmitter is duplicated at the receiver. This circuitry is incorporated into the transmit-and-receive shift registers that send and receive the actual message.

Unlike the parity check which is based on addition, CRC is based on binary division. In CRC, instead of adding bits to achieve a desired parity, a sequence of redundant bits, called the CRC or the CRC remainder, is appended to the end of the message so that resulting data unit becomes exactly divisible by a second, predetermined binary number. At its destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit or message is assumed to be intact and is therefore accepted. A remainder indicates that the data unit has been damaged during transmission and therefore must be rejected.

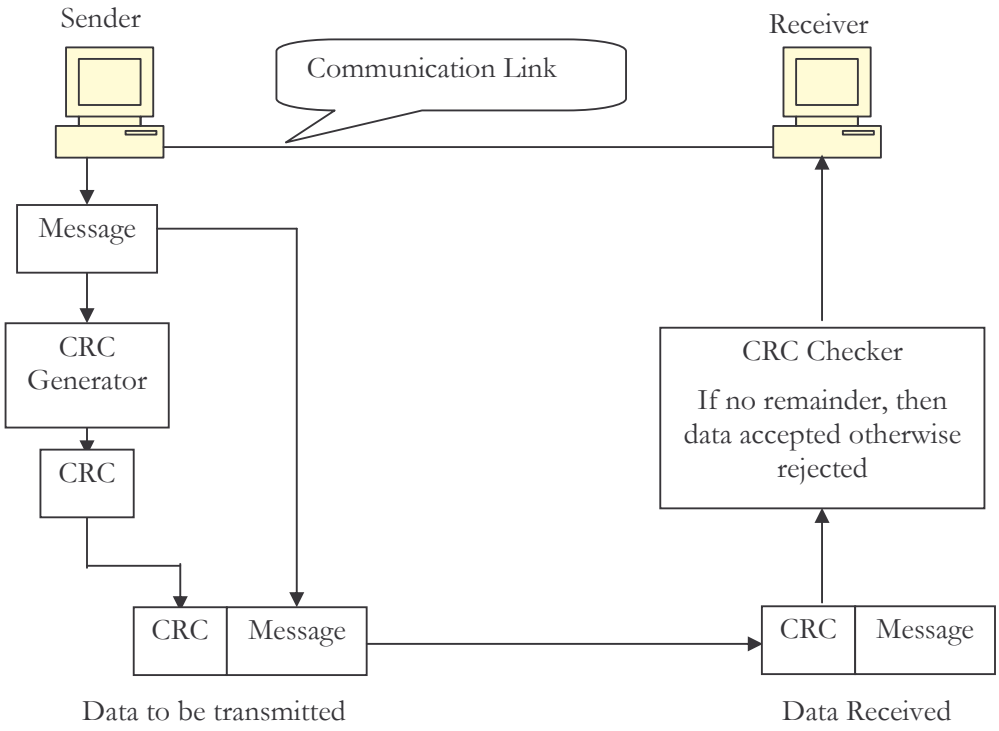


Figure 3.9 CRC generator and checker

Example 3

Compute the CRC-4 character for the following message using a "divisor" constant of 10011 on data unit 1100 0110 1011 01

Solution

Notice that the "divisor" is 5-bits, one more than the number indicated by the CRC type (CRC-4).

We start the process by adding four zeros to the data stream and removing the spaces we have been using for convenience:

110001101011010000

Next, set up the problem to appear as a division problem:

$$10011 \;)\; \overline{110001101011010000}$$

Start the "division" process by exclusive OR the "divisor" with the first five bits of the message:

$$10011 \;)\; \overline{110001101011010000}$$
$$\underline{10011}$$
$$1011$$

Now bring "down" one bit so that the result of the exclusive OR process is filled out to the "divisor" size and repeat the process:

```
10011 )  110001101011010000
         10011
         -----
          10111
          10011
          -----
            100
```

Continue with the process until all of the bits in the message plus the added four zeros are used up:

```
10011 )  110001101011010000
         10011
         -----
          10111
          10011
          -----
            10010
            10011
            -----
              11011
              10011
              -----
                10000
                10011
                -----
                  11100
                  10011
                  -----
                    11110
                    10011
                    -----
                      11010
                      10011   ---> CRC or remainder
                      ----
                      1001
```

The CRC character is appended onto the end of the message and transmitted. At the receiver, the process is repeated, except that there are no zeros added to the message. Instead, the CRC character fills up those positions. If the result of the process at the receiver produces zero then no errors occurred. If any bit or combinations of bits are wrong, then the receiver will yield a non-zero result.

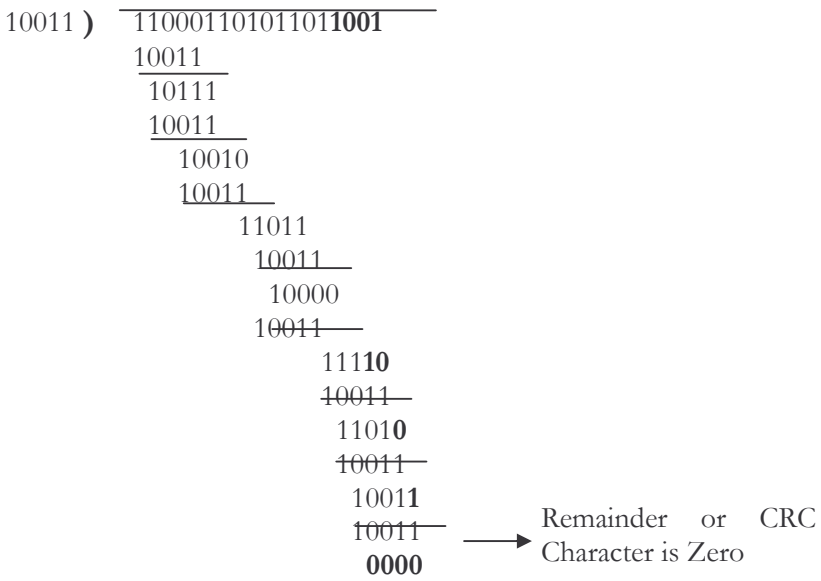
Example 4

Demonstrate how a receiver detects a good message and a message with several errors in it.

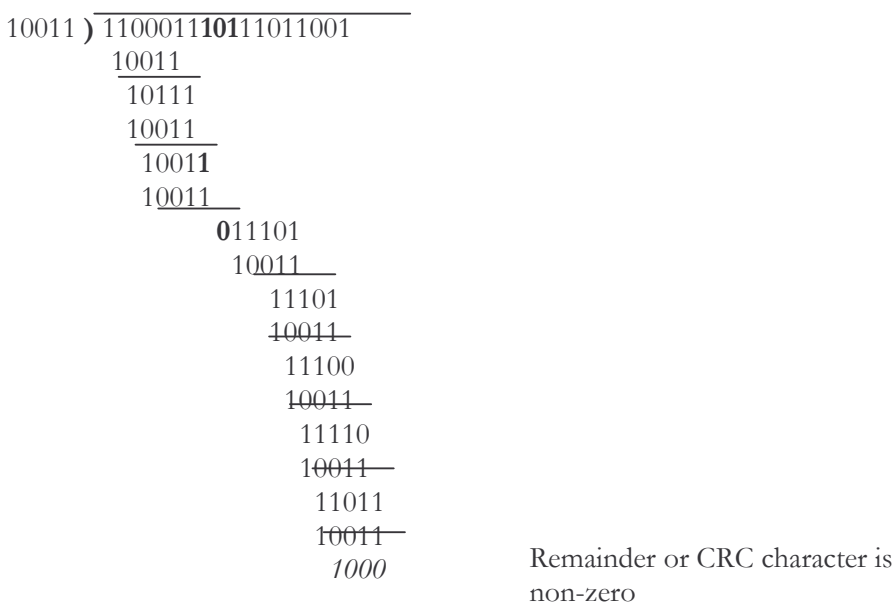
Solution

Repeat steps of EX 3 but this time use CRC character in place of extra zeros:

The changes in the bits brought down are highlighted. Notice how they produce different results from EXAMPLE 3. This eventually results in a CRC of 0000 if everything is correct. It means receiver will accept data unit if CRC checker generates CRC Character Zero at receiver side.



Now let's suppose there are three errors in the message that will also be highlighted. Follow the problem to see how the CRC will be non-zero:



As the CRC character yielded by the CRC ~~checker~~ at receiver side is non-zero; it means there is an error in data unit received. So receiver discards the data unit.

NOTE

Given the small size (CRC-4) of this example, there could easily be error combinations that would produce a zero CRC result. This is the reason that most CRC systems today use either CRC-16, CRC-32 or CRC-64.

CHECKSUM ERROR DETECTION

Another method of error detection uses a process known as **checksum** to generate an error-detection character. The character results from summing all the bytes of a message together, discarding and carry-over for the addition. Again, the process is repeated at the receiver and the two checksums are compared. A match between receiver checksum and transmitted checksum indicates good data. A mismatch indicates an error has occurred.

This method, like CRC, is capable of detecting single or multiple errors in the message. The major advantage of checksum is that it is simple to implement in either hardware or software. The drawback to checksum is that, unless you use a fairly large checksum (16- or 32-bit instead of 8-bit), there are several data-bit patterns that could produce the same checksum result, thereby decreasing its effectiveness. It is possible that if enough errors occur in a message that a checksum could be produced that would be the same as a good message. This is why both checksum and CRC error-detection methods do not catch 100% of the errors that *could* occur, they both come pretty close.

Example 5

What is the checksum value for the extended ASCII message "Help"?

Solution

The checksum value is found by adding up the bytes representing the Help Characters:

01001000	H
01100101	e
01101100	l
01110000	p
00110001	**Checksum**

Example 6

What is the checksum value for the extended ASCII message "Hello"?

Solution

The checksum value is found by adding up the bytes representing the Hello Characters:

0 1 0 0 1 0 0 0	H
0 1 1 0 0 1 0 1	e
0 1 1 0 1 1 0 0	l
0 1 1 0 1 1 0 0	l
0 1 1 0 1 1 1 1	0
0 0 11 0 0 1 0	**Checksum**

3.4 ERROR CORRECTION

It is important to understand the meaning of the word *error* in data storage and transmission. When an n-bit data or message is sent and received, the receiver always receives *n* bits, but some of them may be bad. A bad bit does not disappear, nor does it change into something other than a bit. A bad bit simply changes its value, either from 0 to 1 or from 1 to 0. This makes it relatively easy to correct the bit. The code should tell the receiver which bits are bad, and the receiver can then easily correct the bits by inverting them.

Error correction can be handled by two ways: error correction by retransmission and forward error correction.

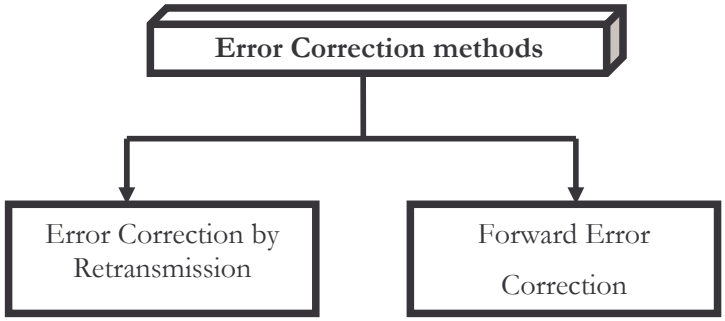


Figure 3.10 Error correction methods

ERROR CORRECTION BY RETRANSMISSION

In error correction by retransmission, when an error is discovered, the receiver can have sender retransmits the entire data unit. In this mechanism, it allows receiver to inform the sender of the damaged or corrupted data units during transmission and coordinates the retransmission of those data units by sender.

However, if propagation delays, due to distance, are large, the technique may become as inefficient as to be useless.

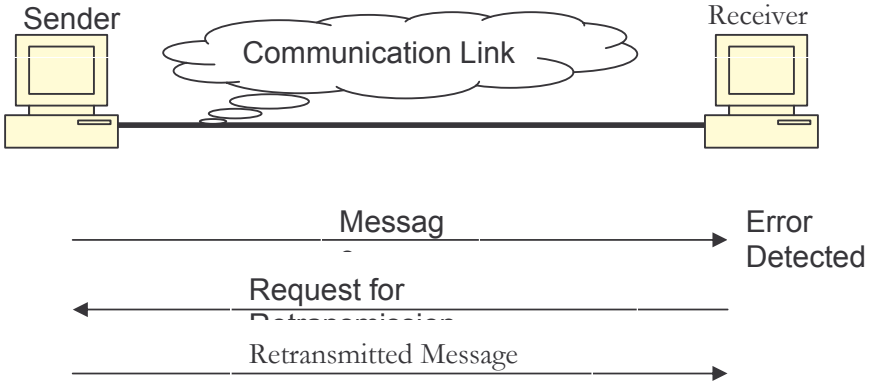


Figure 3.11 Error recovery by detection and retransmission

FORWARD ERROR CORRECTION

Error correction by retransmission is an acceptable method of handling data errors in LAN-based networks because retransmissions of most messages result in a short delay and a little extra use of bandwidth resources. Now, imagine a satellite orbiting around Jupiter or Saturn, transmitting critical visual data as binary stream information. The time it takes for those transmissions to reach Earth is measured in hours. During this time, the satellite has adjusted its orbit and is soaring across new territory and sending additional data. Correcting errors in these messages cannot be done by retransmission. A request for that retransmission takes as long to get to the satellite as the original message took to get to Earth. Then consider the time it would take to retransmit the message. What would the satellite do with new data, reach it while it tries to handle the retransmitting of old data? The memory needed to hold the old data in case it would need to be resent is

astronomical to say the least. Instead, a forward error-correcting method such as the Hamming code is used so that errors can be corrected as they are detected.

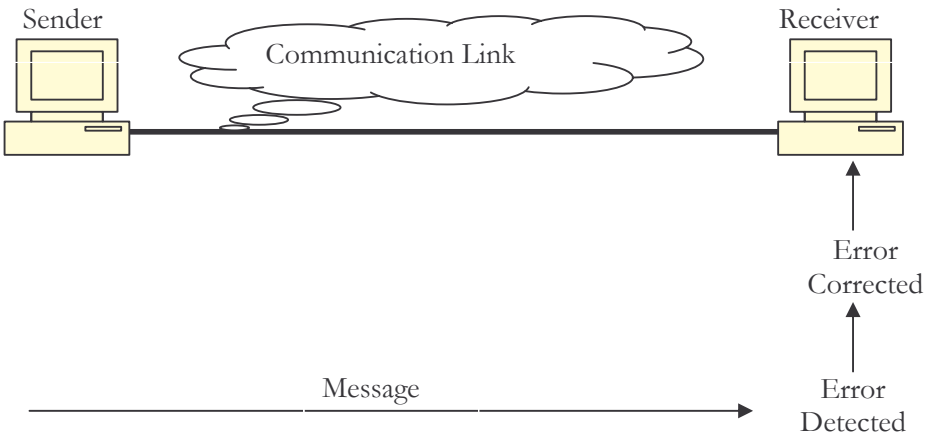


Figure 3.12 Forward Error Correction

ERROR CORRECTION USING VRC/LRC OR TWO-DIMENSIONAL PARITY CHECK

Parity check is primarily used for detecting errors in a serial data character. A bad parity match indicates a logic error has occurred in one of the character's data bits. The use of parity called a **vertical redundancy check (VRC)** can be extended to allow single-bit error correction to take place in a received data stream. By having the ability to correct an error, a receiver would not require a message to be retransmitted, but could do the correction itself. The trade-off in using an error-correction scheme is that an additional character has to be sent with the message and additional software and/or hardware must be used to create and interpret that character. For asynchronous data transmission, that character is known as the **longitudinal redundancy check (LRC)** character.

Using a VRC/LRC or 2-dimensional parity check system, the message is sent with each character containing the regular even-parity bit known as the VRC bit. As with error-detection schemes, any mismatch between transmitted and received VRCs indicates that the character contains a bad data bit. In order to correct the bad bit, what is left to be done is to determine which of the character's bits the bad one is. This is where LRC comes in. It is used to create a cross-matrix type of configuration where the VRC bit denotes the row (character) and the LRC, the column (bit position) of the message's bad bit. At the sending site, each of the data bits of each character is exclusive ORed with the bits of all the other data bits. This is best illustrated in figure 3.13 on next page.

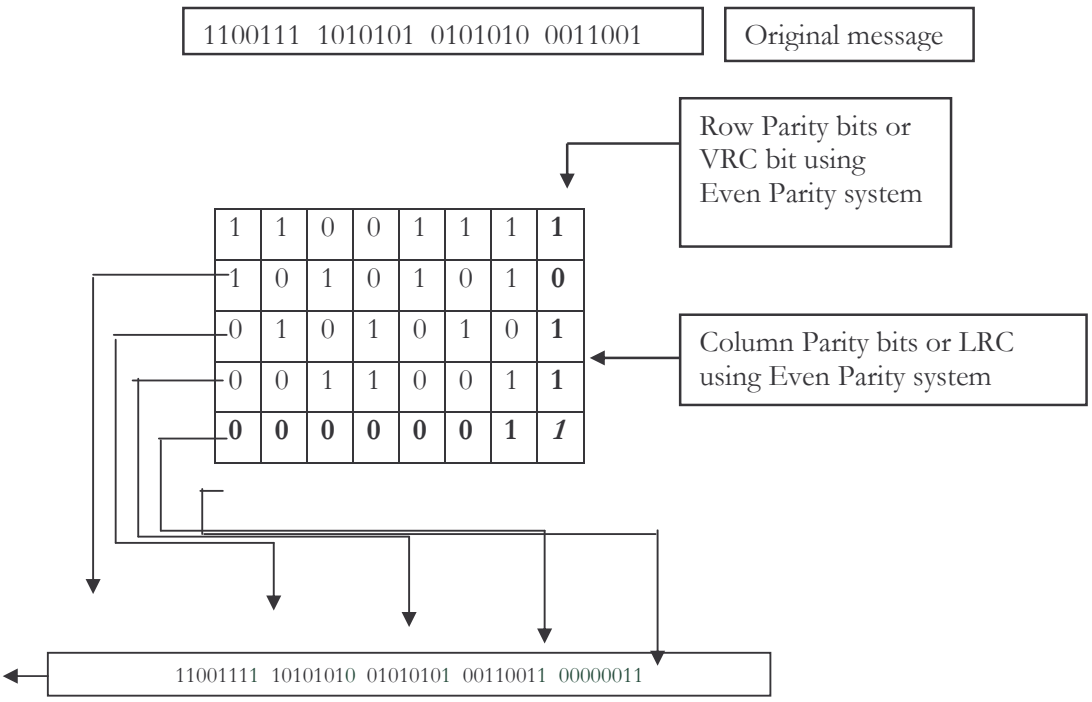


Figure 3.13 Two-dimensional parity check

This error-correction method and others, which are similar, are known as **forward error correction (FEC)** because errors are corrected as the message is received. There is no requirement to retransmit the message as long as the errors remain infrequent. If more than one error occurs in a message, then more than one LRC and one VRC bit will be bad and there is no way to determine which LRC bit goes with which VRC character. In this case, the excessive number of bit errors is indicative of a severe condition. Once the cause of the problem is resolved, the message will have to be retransmitted fully.

HAMMING CODE

For synchronous data streams, a error-correcting process called **Hamming code** is commonly used. This method is fairly complex from the standpoint of creating and interpreting the error bits. It is implemented in software algorithms and relies on a lot of preliminary conditions agreed upon by the sender and receiver. Error bits, called Hamming bits, are inserted into the message at random locations. It is believed that the randomness of their locations reduces the statistical odds that these Hamming bits themselves would be in error. This is based on a mathematical assumption that because there are so many more messages bits compared to Hamming bits, that there is a greater chance for a message bit to be in error than for a Hamming bit to be wrong.

Hamming code differs from other error detection and correction codes such as CRC or Checksum. In the other error detection and correction codes such as checksum or CRC, redundant bit or error control bits are appended at the end of data unit but in case of Hamming code, error control codes are randomly inserted into data unit.

But each and every bit in the message, including the Hamming bits, has the same chance of being corrupted as any other bit. Be that as it may, Hamming bits are inserted into the

data stream randomly. The only crucial point in the selection of their locations is that both the sender and receiver are aware of where they actually are.

The first step in the process is to determine how many Hamming bits (**H**) are to be inserted between the message (**M**) bits. Then their actual placement is selected. The number of bits in the message (M) are counted and used to solve the following equation to determine the number of Hamming (H) bits:

$$2^H \geq M + H - 1$$

Number of Data Bits (M)	Number of Redundancy Bits (H)
1	2
2	3
3	3
4	3
5	4
6	4
7	4
8	4

Table 3.2 relationship between data and redundancy bits.

Once the number of Hamming bits is determined, the actual placement of the bits into the message is performed. It is important to note that despite the random nature of the Hamming bit placements, the exact same placements must be known and used by both the transmitter and the receiver. This is necessary so that the receiver can remove the Hamming bits from the message sent by the transmitter and compare them with a similar set of bits generated at the receiver.

Example 7

How many Hamming bits are required when using the Hamming code with the extended ASCII synchronous message "Help!" ?

Solution

The total number of bits in the message is:

$$M = 8\text{-bits/character} \times 5 \text{ characters} = 40 \text{ bits}$$

This number is used in Equation ($2^H \geq M + H - 1$) to determine the number of Hamming bits:

$$2^H \geq 40 + H + 1$$

The closest value to try is 6 bits for *H,* since $2^6 = 64$, which is greater than $40 + 6 + 1 = 47$. This satisfies the equation. Number of Redundancy bits or Hamming bits required are six.

The Hamming code can be applied to data units of any length, which uses the relationship between data and redundancy bits discussed above. Once the Hamming bits are inserted into their positions within the message, their states (high or low) need to be determined. Starting with the least significant bit (LSB) as bit 1, the binary equivalent of each message-bit position with a high (1) state is exclusive ORed with every other bit position containing a 1. The result of the exclusive-OR process is the states of the

Hamming bits. Once again, as with previous error detection-and-correction processes, it is best to view how the Hamming code works by using an example.

Example 8

Determine the states of the six Hamming bits inserted into the message "Help!" at every other bit position starting with the LSB

Solution

The total number of bits in the message is:

$$M = \text{8-bits/character} \times 5 \text{ characters} = 40 \text{ bits}$$

This number is used in Equation ($2^H \geq M + H - 1$)to determine the number of Hamming bits:

$$2^H \geq 40 + H + 1$$

The closest value to try is 6 bits for *H,* since 26 = 64, which is greater than 40 + 6 + 1 = 47. This satisfies the equation.

Thus, H = 6

For simplicity, we shall insert the Hamming bits a little less randomly:

H e l p !

01001000 01100101 01101100 01110000 001H0H0H0H0H1H

MSB LSB

Starting from the LSB on the right, the first 1 is encountered in bit position 2, the next in position 12, the next in position 19 and so forth:

Bit Position	Equivalent Binary of Bit Position Number using Number of Hamming Bits
2	0 0 0 0 1 0
12	0 0 1 1 0 0
19	0 1 0 0 1 1
20	0 1 0 1 0 0
21	0 1 0 1 0 1
25	0 1 1 0 0 1
26	0 1 1 0 1 0
28	0 1 1 1 0 0
29	0 1 1 1 0 1
31	0 1 1 1 1 1
33	1 0 0 0 0 1
36	1 0 0 1 0 0
37	1 0 0 1 0 1
42	1 0 1 0 1 0
45	1 0 1 1 0 1
Hamming bits	**1 0 0 1 1 0**

All these binary values are exclusive ORed together—an odd number of ones produce a 1, and an even count, a 0—to create the Hamming bits values. These values are

substituted for the H-bits in the message. The entire thing is then transmitted and the process repeated at the receiver:

0100100001100101011011000111000000110000010110

If the message was received without any errors, then the Hamming-bit states produced at the receiver will match the ones sent. If an error in one bit did occur during transmission, then the difference between the transmitted Hamming bits and the receiver results will be the bit position of the bad bit. This bit is then inverted to its correct state.

Example 9

Demonstrate how the Hamming code is used to correct a single-bit error in the data stream.

Solution

Suppose, during the transmission of the message, bit 19 experiences a noise spike that cause it to be received as a 0 instead of 1. The receiver goes through the process of determining the states of the Hamming code, resulting in this calculation:

Bit Position	**Equivalent Binary of Bit Position Number using Number of Hamming Bits**
2	0 0 0 0 1 0
12	0 0 1 1 0 0
20	0 1 0 1 0 0
21	0 1 0 1 0 1
25	0 1 1 0 0 1
26	0 1 1 0 1 0
28	0 1 1 1 0 0
29	0 1 1 1 0 1
31	0 1 1 1 1 1
33	1 0 0 0 0 1
36	1 0 0 1 0 0
37	1 0 0 1 0 1
42	1 0 1 0 1 0
45	1 0 1 1 0 1
Hamming bits	**1 1 0 1 0 1**

Notice that bit 19 is not included in the list since it was received as a low state instead of a high state. Now we compare the Hamming code transmitted to this one the receiver just derived:

Transmitted code: 1 0 0 1 1 0

Receiver code: 1 1 0 1 0 1

0 1 $\overline{0\ 0\ 1\ 1}$ = bit 19 corrupted

To correct the error revert the bit at location 19 in our example in received data unit.

There is no "black magic" mystery to why the Hamming code works. The originally transmitted codes are formulated by adding binary bits together (the exclusive OR

process), ignoring carries. A similar process occurs at the receiver. If a bit has changed, then the two sums will be different and the difference between them will be the bit position number that was not added at either the transmitter or the receiver. By comparing the two Hamming codes using exclusive OR gates, the numbers are effectively being subtracted from one another (another function of the exclusive OR gate) and the difference is the bad bit position.

SUMMARY

Error detection and correction methods are necessary to assure the integrity of the data sent from one location to another. The types of methods used support both asynchronous and synchronous-type data streams. Asynchronous error detection is facilitated by the use of a parity bit with each character of data sent. Error correction for asynchronous data utilizes the LRC/VRC method, which duplicates the parity process (VRC) and examines each character by bit position (LRC). Synchronous data streams apply CRC or checksum for error detection and the Hamming code for error correction. Following Table summarizes the error methods discussed in this chapter and supplies a quick comparison reference for them.

Error Method	**Data Type**	**Detection / correction**	**Overhead**
Parity	Asynchronous	Detection only	One Bit Added per Character
LRC/VRC	Asynchronous	Detection and Correction	One Bit per Character Plus LRC Character
Checksum	Asynchronous/ Synchronous	Detection only	Checksum Character at End of Message
CRC	Synchronous	Detection only	CRC Bytes at End of Message
Hamming Code	Synchronous	Detection and Correction	Hamming Bits Inserted into Data Stream

Table 3.3 Error Methods Summary

PRACTICE SET

Review questions

1. Why are most error detection and correction methods designed for single-bit detection or correction?

2. What is the state of the parity bit using an odd-parity system for the extended ASCII character M?

3. Why doesn't a good match between transmitted and received parity bits guarantee that the character is good?

4. What does an incorrect VRC bit indicate?

5. How many errors in a message can be corrected using LRC?

6. Which error-detection method is used most frequently with asynchronous data streams?

7. How many errors can parity reliably detect in a single character?

8. Which type of logic gate is used to generate parity bits?

9. What is one drawback when using parity for error detection?

10. What is meant by forward error correction?

11. How is the LRC character generated?

12. What does a bad VRC match signify?

13. Why is CRC-16 preferred to parity for error detection in synchronous data systems?

14. Why are larger CRC "divisors" preferred over shorter ones?

15. What is the 16-bit checksum value for the extended ASCII message "Hello World"? Do not forget space character.

16. Why are larger checksums preferred over shorter ones?

Multiple Choice Questions

1. Which error detection method consists of just one redundant bit per data unit?
A) Simple parity check B) Two-dimensional parity check
C) CRC D) Checksum

2. In cyclic redundancy checking, what is the CRC?
A) The divisor B) The quotient C) The dividend D) The remainder

3. In cyclic redundancy checking, the divisor is ……. the CRC.
A) The same size as B) one bit less than
C) one bit more than D) none of the above

4. A burst error means that ……. bits in the data unit have changed.
A) two or more bits B) single-bit C) no bit D) none of the above

5. In ……. error correction, the receiver corrects errors without requesting retransmission.
A) backward B) retransmission C) forward D) none of the above

6. In ……. error correction, the receiver asks the sender to send the data again.
A) backward B) retransmission C) forward D) none of the above

7. We add r redundant bits to each block to make the length n = k + r. The resulting n-bit blocks are called …….
A) datawords B) blockwords
C) codewords D) none of the above

8. A simple parity-check code can detect ……. errors.
A) an even-number of B) two C) no errors D) an odd-number of

9. The ……. of errors is more difficult than the …….
A) correction; detection B) detection; correction
C) creation; correction D) hamming; detection

10. In modulo-2 arithmetic, we use only …….
A) 1 and 2 B) 0 and 2 C) 0 and 1 D) none of the above

11. The checksum of 1111 and 1111 is …….
A) 1111 B) 0000 C) 1110 D) 0111

12. The checksum of 0000 and 0000 is …….
A) 1111 B) 0000 C) 1110 D) 0111

13. How many Hamming bits are required for 7 bit message?
A) 2 B) 3 C) 4 D) 5

14. How many Hamming Bits are required for 4 bit message?
A) 2 B) 3 C) 4 D) 5

* * *

Chapter 4 Transmission Medium

Many different types of medium can be used for the physical layer or physical connection. For example, telephone twisted pair, coax cable, shielded copper cable and optical fibers are the main types used for LANs. Different transmission techniques generally categorized as baseband or broadband transmission may be applied to each of these media types.

OBJECTIVES OF THIS CHAPTER

After completing this chapter, the student should be able to:

- define transmission medium, electromagnetic spectrum
- study guided medium like coaxial, optical fiber or twisted pair cable
- measure performance of transmission medium
- study unguided medium, such as infrared, microwave
- understand the working of satellite

4.1 INTRODUCTION

Transmission medium provides physical path for the transfer of signals. Transmission medium is the physical path between transmitter and receiver in a data transmission system. Transmission media can be classified as guided or unguided based on medium used. In both cases, communication is in the form of electromagnetic waves. In case of guided medium, the waves are guided along a solid medium, such as copper twisted pair, copper coaxial cable and optical fiber. The atmosphere and outer space are examples of unguided medium. In the unguided medium provides a means of transmitting electromagnetic signals but it did not guide them; this form of transmission is usually referred to as *wireless transmission.*

The characteristics and quality of a data transmission are determined both by the characteristics of the medium and the characteristics of the signal. In the case of guided media, the medium itself is more important in determining the limitations of transmission.

For unguided media, the bandwidth of the signal produced by the transmitting antenna is more important than the medium in determining transmission characteristics. One key property of signals transmitted by antenna is directionality. In general, signals at lower frequencies are omnidirectional; that is, the signal propagates in all directions from the antenna. At higher frequencies, it is possible to focus the signal into a directional beam.

In considering the design of data transmission systems, a key concern, generally, is data rate and distance: the greater the data rate and distance, the better result. A number of design factors relating to the transmission medium and to the signal determine the data rate and distance:

Bandwidth : All other factors remaining constant, the greater the bandwidth of a signal, the higher the data rate that can be achieved.

Transmission impairments : Impairments, such as attenuation, limit the distance. For guided media, twisted pair generally suffers more impairment than coaxial cable, which in turn suffers more than optical fiber.

Interference : Interference from competing signals in overlapping frequency bands can distort or wipe out a signal. Interference is of particular concern for unguided media, but it is also a problem with guided media. For guided media, interference can be caused by emission of electromagnetic field from nearby cables. For example, twisted pair cables are often bundled together, and conduits often carry multiple cables. Interference can also be experienced from unguided transmissions.

Proper shielding of a guided medium can minimize this problem.

Number of receivers : A guided medium can be used to construct a point-to-point link or a shared link with multiple attachments. In the latter case, each attachment introduces some attenuation and distortion on the line, limiting distance and/or data rate.

Electromagnetic Spectrum and frequencies : Figure 4.1 depicts the electromagnetic spectrum and indicates the frequencies at which various guided media and unguided transmission techniques operate. In this chapter, we examine these guided and unguided alternatives.

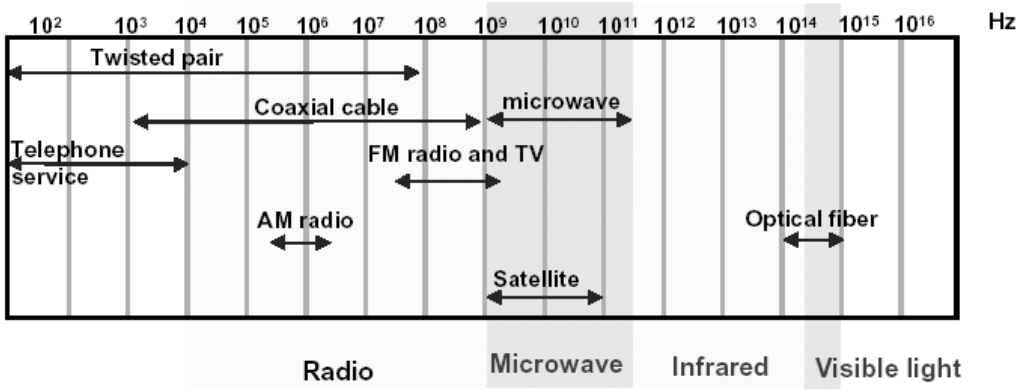


Figure 4.1 Electromagnetic Spectrum with frequency ranges

Transmission media can be divided into two broad categories : Guided and Unguided shown in following diagram.

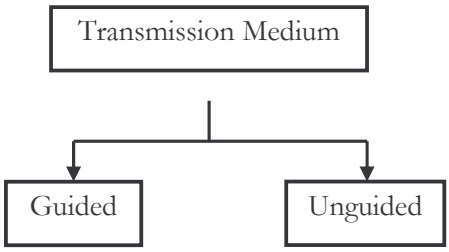


Figure 4.2 Types of transmission media

4.2 GUIDED MEDIA

Guided media are those mediums that provide a conduit from one device to another, include twisted-pair cable, coaxial cable and fiber-optic cable.

Guided Transmission Media uses a "cabling" system that guides the data signals along a specific path. The data signals are bound by the "cabling" system. Guided Media is also known as Bound Media. Cabling is meant in a generic sense in the previous sentences and is not meant to be interpreted as copper wire cabling only. Cable is the medium through which information usually moves from one network device to another.

Twisted pair cable and coaxial cable use metallic (copper) conductors that accept and transport signals in the form of electric current. Optical fiber is a glass or plastic cable that accepts and transports signals in the form of light.

There four basic types of Guided Media :

1. Open Wire
2. Twisted Pair
3. Coaxial Cable
4. Optical Fiber

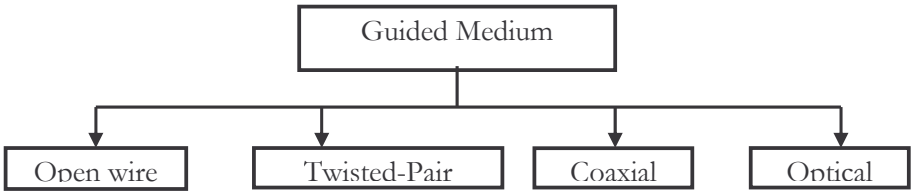


Figure 4.3 Types of guided media

OPEN WIRE

Open Wire is traditionally used to describe the electrical wire strung along power poles. There is a single wire strung between poles. No shielding or protection from noise interference is used. We are going to extend the traditional definition of Open Wire to include any data signal path without shielding or protection from noise interference. This can include multiconductor cables or single wires. This media is susceptible to a large degree of noise and interference and consequently not acceptable for data transmission except for short distances under 20 ft.

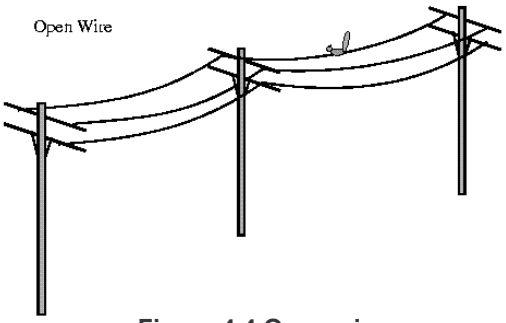


Figure 4.4 Open wire

TWISTED-PAIR (TP) CABLE

Twisted pair cable is least expensive and most widely used. The wires in Twisted Pair cabling are twisted together in pairs. Each pair would consist of a wire used for the +ve data signal and a wire used for the -ve data signal. Any noise that appears on one wire of the pair would occur on the other wire. Because the wires are opposite polarities, they are 180 degrees out of phase When the noise appears on both wires, it cancels itself out at the receiving end. Twisted Pair cables are most effectively used in systems that use a balanced line method of transmission.

Physical description

- Two insulated copper wires arranged in regular spiral pattern.
- Number of pairs are bundled together in a cable.
- Twisting decreases the crosstalk interference between adjacent pairs in the cable, by using different twist length for neighboring pairs.

A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together. One of the wire is used to carry signals to the receiver, and the other is used only a ground reference.

Why the cable is twisted?

In past, two parallel flat wires were used for communication. However, electromagnetic interference from devices such as a motor can create noise over those wires.

If the two wires are parallel, the wire closest to the source of the noise gets more interference and ends up with a higher voltage level than the wire farther away, which results in an uneven load and a damaged signal. If, however, the two wires are twisted around each other at regular intervals, each wire is closer to the noise source for half the time and farther away for the other half. The degree of reduction in noise interference is determined specifically by the number of turns per foot. Increasing the number of turns per foot reduces the noise interference. To further improve noise rejection, a foil or wire braid shield is woven around the twisted pairs.

Twisted pair(TP) cable supports both analog and digital signals. TP cable can be either *unshielded TP (UTP)* cable or *shielded TP (STP)* cable. Cables with a shield are called Shielded Twisted Pair and commonly abbreviated STP. Cables without a shield are called Unshielded Twisted Pair or UTP. Shielding means metallic material added to cabling to reduce susceptibility to noise due to electromagnetic interference (EMI).

IBM produced a version of TP cable for its use called STP. STP cable has a metal foil that encases each pair of insulated conductors. Metal casing used in STP improves the quality of cable by preventing the penetration of noise. It also can eliminate a phenomenon called crosstalk.

Crosstalk is the undesired effect of one circuit (or channel) on another circuit (or channel). It occurs when one line picks up some of the signal traveling down another line. Crosstalk effect can be experienced during telephone conversations when one can hear other conversations in the background.

Twisted-pair cabling with additional shielding is used to reduce crosstalk and other forms of electromagnetic interference (EMI). It has an impedance of 150 ohms, has a maximum length of 90 meters, and is used primarily in networking environments with a high amount of EMI due to motors, air conditioners, power lines, or other noisy electrical components. STP cabling is the default type of cabling for IBM Token Ring networks. STP is more expensive as compared to UTP.

UTP is cheap, flexible, and easy to install. UTP is used in many LAN technologies, including *Ethernet* and *Token Ring*.

In computer networking environments that use twisted-pair cabling, one pair of wires is typically used for transmitting data while another pair receives data. The twists in the cabling reduce the effects of crosstalk and make the cabling more resistant to electromagnetic interference (EMI), which helps maintain a high signal-to-noise ratio for reliable network communication. Twisted-pair cabling used in Ethernet networking is usually unshielded twisted-pair (UTP) cabling, while shielded twisted-pair (STP) cabling is typically used in Token Ring networks. UTP cabling comes in different grades for different purposes.

The Electronic Industries Association (EIA) has developed standards to classify UTP cable into seven categories. Categories are determined by cable quality, with CAT 1 as the lowest and CAT 7 as the highest.

Category	Data Rate	Digital/Analog	Use
CAT 1	< 100 Kbps	Analog	Telephone systems
CAT 2	4 Mbps	Analog/Digital	Voice + Data Transmission
CAT 3	10 Mbps	Digital	Ethernet 10BaseT LANs
CAT 4	20 Mbps	Digital	Token based or 10baseT LANs
CAT 5	100 Mbps	Digital	Ethernet 100BaseT LANs
CAT 6	200 Mbps	Digital	LANs
CAT 7	600 Mbps	Digital	LANs

Table 4.1 Categories of UTP cable

The quality of UTP may vary from telephone-grade wire to extremely high-speed cable. The cable has four pairs of wires inside the jacket. Each pair is twisted with a different number of twists per inch to help eliminate interference from adjacent pairs and other electrical devices. The tighter the twisting, the higher the supported transmission rate and the greater the cost per foot.

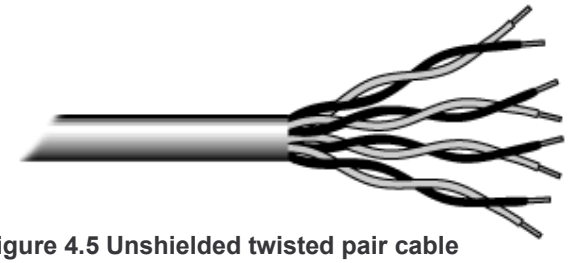


Figure 4.5 Unshielded twisted pair cable

Unshielded Twisted Pair Connector

The standard connector for unshielded twisted pair cabling is an RJ-45 connector. This is a plastic connector that looks like a large telephone-style connector. A slot allows the RJ-45 to be inserted only one way. RJ stands for Registered Jack, implying that the connector follows a standard borrowed from the telephone industry. This standard designates which wire goes with each pin inside the connector.

STP cabling comes in various grades or categories defined by the EIA/TIA wiring standards, as shown in the table 4.2

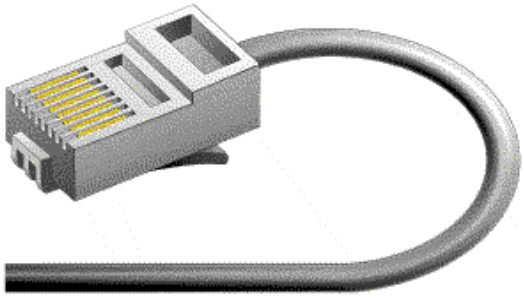


Figure 4.6 RJ-45 connector

STP Cabling Categories :

Category	Description
IBM Type 1	Token Ring transmissions on AWG #22 wire up to 20 Mbps.
IBM Type 1A	Fiber Distributed Data Interface (FDDI), Copper Distributed Data Interface (CDDI), and Asynchronous Transfer Mode (ATM) transmission up to 300 Mbps.
IBM Type 2A	Hybrid combination of STP data cable and CAT3 voice cable in one jacket.
IBM Type 6A	AWG #26 patch cables.

Table 4.2 STP Cabling categories

Transmission characteristics

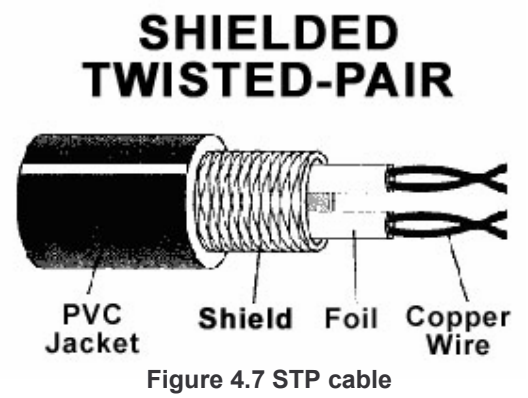


Figure 4.7 STP cable

- Requires amplifiers for analog signals.
- Requires repeaters for digital signals.
- Attenuation is a strong function of frequency.
- Higher frequency implies higher attenuation.
- Susceptible to interference and noise.
- Improvement possibilities.
- Shielding with metallic braids or sheathing reduces interference.
- Twisting reduces low frequency interference.
- Different twist length in adjacent pairs reduces crosstalk.

Comparison of Unshielded and shielded twisted pairs

Unshielded twisted pair (UTP).

- Ordinary telephone wire.
- Subject to external electromagnetic interference.

Shielded twisted pair (STP)

- Shielded with a metallic braid or sheath.
- Reduces interference.
- Better performance at higher data rates.
- More expensive and difficult to work compared to UTP.

Applications of TP cable

- Most common transmission media for both digital and analog signals.
- TP cables are used in telephone lines to provide voice and data channels.
- The line that connects subscribers to the central telephone office is most commonly UTP cable.
- The DSL lines that are used by the telephone companies to provide high data rate connections also use high bandwidth capability UTP cable.
- Local Area Network (LAN) also uses twisted-pair cable.

COAXIAL CABLE

A form of network cabling used primarily in older Ethernet networks and in electrically noisy industrial environments. The name "coax" comes from its two-conductor construction in which the conductors run *co*ncentrically with each other along the *ax*is of the cable. Coaxial cabling has been largely replaced by twisted-pair cabling for local area

network (LAN) installations within buildings, and by fiber-optic cabling for high-speed network backbones.

Coaxial cable (or coax) carries signals of higher frequency ranges than twisted-pair cable. Instead of having two wires, coax has a central core conductor of solid or standard wire (usually copper) enclosed in an insulating sheath, which is, in turn, encased in an outer conductor of metal foil, braid, or a combination of the two (also usually copper).

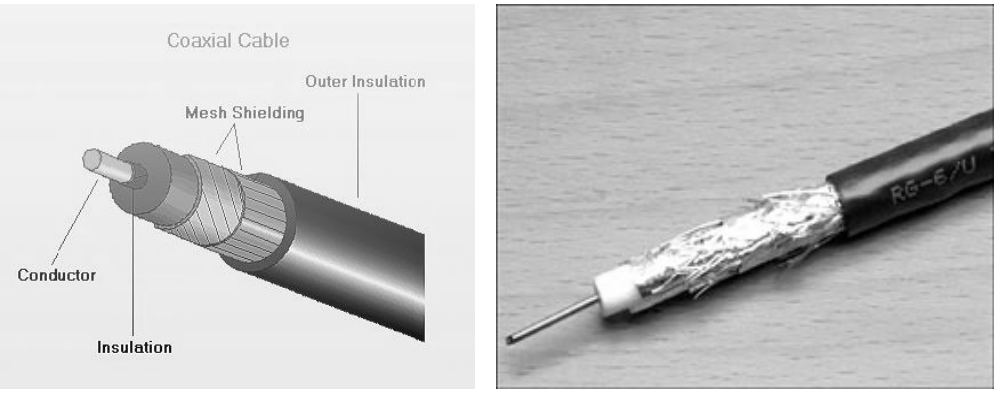


Figure (a) **Figure (b)**

Figure 4.8 Coaxial cable

The outer metallic wrapping serves both as a shield against and as the second conductor, which completes the circuit. This outer conductor is also enclosed in an insulating sheath, and a plastic cover protects the whole cable.

Coaxial cable supports both analog and digital signals.

Physical description

- Consists of two conductors with construction that allows it to operate over a wider range of frequencies compared to twisted pair
- Hollow outer cylindrical conductor surrounding a single inner wire conductor
- Inner conductor held in place by regularly spaced insulating rings or solid dielectrical material
- Outer conductor covered with a jacket or shield
- Diameter from 1 to 2.5 cm
- Shielded concentric construction reduces interference and crosstalk
- Can be used over longer distances and support more stations on a shared line than twisted pair

Coaxial cable Standards

Although Coaxial cabling is difficult to install, it is highly resistant to signal interference. In addition, it can support greater cable lengths between n/w devices than twisted pair cable.

Coaxial cabling comes in various types and grades. The most common are:

Thicknet cabling, which is an older form of cabling, used for legacy 10Base5 Ethernet backbone installations. This cabling is generally yellow and is referred to as RG-8 or N-series cabling. Strictly speaking, only cabling labeled as IEEE 802.3 cabling is true thicknet cabling.

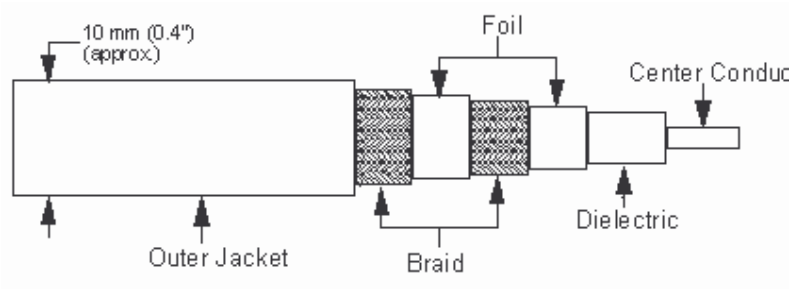


Figure 4.9 Thicknet Coaxial Cable

Thinnet coaxial cabling, *which is used in 10 Base2 networks for small Ethernet installations. This grade of coaxial cabling is generally designated as RG-58A/U cabling, which has a stranded conductor and a 53-ohm impedance. This kind of cabling uses BNC connectors for connecting to other networking components, and must have terminators at free ends to prevent signal bounce.*

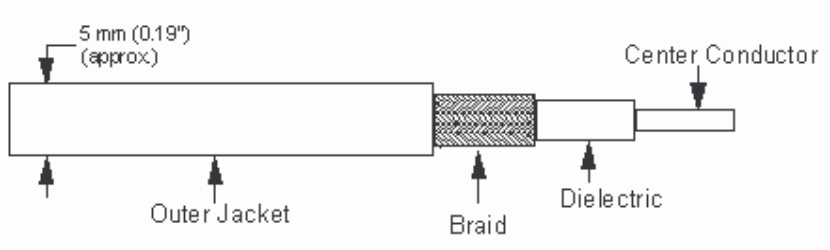


Figure 4.10 Thinnet Coaxial cable

ARCNET cabling, *which uses thin coaxial cabling called RG-62 cabling with an impedance of 93 ohms.*

RG-59 cabling, *which is used for cable television (CATV) connections.*

Coaxial cables are categorized by radio government (RG) rating. Each RG number denotes a unique set of physical specifications, including the wire gauge (gauge is the measure of the thickness of the wire) of the inner conductor, the thickness and type of inner insulator, the construction of the shield, and the size and type of the outer casting.

- *RG-8. Used in thick Ethernet.*
- *RG-9. Used in thick Ethernet.*
- *RG-11. Used in thick Ethernet.*
- *RG-58. Used in thin Ethernet.*
- *RG-59. Used in cable TV.*

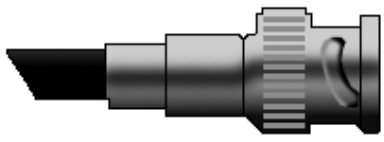


Figure 4.11 BNC connector

To connect coaxial cable to devices, we need coaxial connector. The most common type of connector used today is the Bayone-Neill-Concelman, or BNC connector.

Transmission characteristics

- Used to transmit both analog and digital signals.
- Superior frequency characteristics compared to twisted pair.
- Can support higher frequencies and data rates.
- Shielded concentric construction makes it less susceptible to interference and crosstalk than twisted pair.
- Constraints on performance are attenuation, thermal noise, and intermodulation noise.
- Requires amplifiers every few kilometers for long distance transmission.
- Usable spectrum for analog signaling up to 500 MHz.
- Requires repeaters every few kilometers for digital transmission.
- For both analog and digital transmission, closer spacing is necessary for higher frequencies/data rates.

Application of Coaxial cable

- The use of coaxial cable started in analog telephone networks where a single coaxial network could carry 10,000 voice signals.
- Later it was used in digital telephone networks where a single coaxial cable could carry digital data up to 600 Mbps. (However, coaxial cable in telephone network has largely been replaced today with fiber-optic cable).
- Most common use is in cable TV.
- Coaxial cabling is often used in heavy industrial environments where motors and generators produce a lot of electromagnetic interference (EMI), and where more expensive fiber-optic cabling is unnecessary because of the slow data rates needed.
- Another common application of coaxial cable is in traditional Ethernet LANs. Because of its high bandwidth, and consequently high data rate, coaxial cable are chosen for digital transmission in early Ethernet LANs.
- 10Base-2, or Thin Ethernet, uses RG-58 coaxial cable with BNC connectors to transmit data at 10 Mbps with a range of 185 m.
- 10Base5, or Thick Ethernet, uses RG-11 to transmit 10 Mbps with a range of 5000 m.

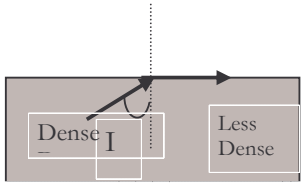
FIBER-OPTIC CABLE

Fiber-optic is a glass cabling media that sends network signals using light. Fiber-optic cabling has higher bandwidth capacity than copper cabling, and is used mainly for high-speed network Asynchronous Transfer Mode (ATM) or Fiber Distributed Data Interface (FDDI) backbones, long cable runs, and connections to high-performance workstations. A fiber-optic cable is made of glass or plastic and transmits signals in the form of light. Light is a form of electromagnetic energy. It travels at its fastest in a vacuum: 3,00,000 kilometers/sec. The speed of light depends on the density of the medium through, which it is traveling (the higher the density, the slower the speed). Light travels in a straight line

as long as it is moving through a single uniform substance. If a ray of light traveling through one substance suddenly enters another (more or less dense), the ray changes direction. This change is called **Refraction** : The direction in which a light ray is refracted depends on the change in density encountered. A beam of light moving from a less dense into a denser medium is bent towards vertical axis.

When light travels into a denser medium, the *angle of incidence* is greater than the *angle of refraction*; and when light travels into a less dense medium, the *angle of incidence* is less than the *angle of refraction*.

Critical Angle : A beam of light moving from a denser into a less dense medium, as the angle of incidence increases the angle of refraction also increases.

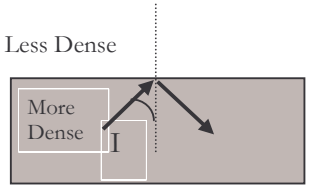


I = Critical Angle, refraction

Figure 4.12 Critical Angle

At some point in this process, the change in the incident angle results in a refracted angle of 90 degrees, with the refracted beam now lying along with horizontal. The incident angle at this point is known as the **critical angle**.

Reflection : When the angle of incidence becomes greater than the critical angle, a new phenomenon occurs called **reflection**. Light no longer passes into the less dense medium at all. Optical fiber use *reflection* to guide light through a channel.



I = Critical Angle, reflection

Figure 4.13 Reflection

A glass or plastic core is surrounded by cladding of less dense glass or plastic. The difference in density of the two materials must be such that a beam of light moving through the core is reflected off the cladding instead of being refracted into it. Information is encoded onto a beam of light as a series of on-off flashes that represents 1 and 0s.

1. **Capacity**

Comparison of optical fiber with twisted pair and coaxial cable

- Much higher bandwidth.
- Can carry hundreds of Gbps (Gigabit per second) over tens of Kilometers (kms).

2. **Smaller size and lightweight**
   - Very thin for similar data capacity.
   - Much lighter and easy to support in terms of weight (structural properties).
3. **Significantly lower attenuation**
4. **EM isolation (Resistance to noise).**
   - Not affected by external EM (Electromagnetic) fields.
   - Not vulnerable to interference, impulse noise, or crosstalk.
   - No energy radiation; little interference with other devices; security from eavesdropping.

5. **Greater repeater spacing**
    - Lower cost and fewer error sources.
6. **Speed**
    - Fiber optic networks operate at high speeds - up into the gigabits.
7. **Distance**
    - Signals can be transmitted further without needing to be "refreshed" or strengthened.
8. **Maintenance**
    - Fiber optic cables costs much less to maintain.

The use of fiber-optics was generally not available until 1970 when Corning Glass Works was able to produce optical fiber with a loss of 20 dB/km. It was recognized that optical fiber would be feasible for telecommunication transmission only if glass could be developed so pure that attenuation would be 20dB/km or less. That is, 1% of the light would remain after traveling 1 km. Today's optical fiber attenuation ranges from 0.5dB/km to 1000dB/km depending on the optical fiber used. Attenuation limits are based on intended application.

In recent years it has become apparent that fiber-optics are steadily replacing copper wire as an appropriate means of communication signal transmission. They span the long distances between local phone systems as well as providing the backbone for many network systems. Other system users include cable television services, university campuses, office buildings, industrial plants, and electric utility companies.

The applications of optical fiber communication have increased at a rapid rate, since the first commercial installation of a fiber-optic system in 1977. Telephone companies began early on, replacing their old copper wire systems with optical fiber lines. Today's telephone companies use optical fiber throughout their system as the backbone architecture and as the long-distance connection between city phone systems. Some 10 billion digital bits can be transmitted per second along an optical fiber link in a commercial network, enough to carry tens of thousands of telephone calls.

A fiber-optic system is similar to the copper wire system. The difference is that fiber-optics use light pulses to transmit information down fiber lines instead of using electronic pulses to transmit information down copper lines. Looking at the components in a fiber-optic chain will give a better understanding of how the system works in conjunction with wire based systems.

At one end of the system is a transmitter. This is the place of origin for information coming on to fiber-optic lines. The transmitter accepts coded electronic pulse information coming from copper wire. It then processes and translates that information into equivalently coded light pulses. A light-emitting diode (LED) or an injection-laser diode (ILD) can be used for generating the light pulses. Using a lens, the light pulses are funneled into the fiber-optic medium where they transmit themselves down the line,

Think of a fiber cable in terms of very long cardboard roll (from the inside roll of paper towel) that is coated with a mirror. If you shine a flashlight in one you can see light at the far end - even if bent the roll around a corner.

Light pulses move easily down the fiber-optic line because of a principle known as total internal reflection. "This principle of total internal reflection states that when the angle of incidence exceeds a critical value, light cannot get out of the glass; instead, the light bounces back in. When this principle is applied to the construction of the fiber-optic strand, it is possible to transmit information down fiber lines in the form of light pulses.

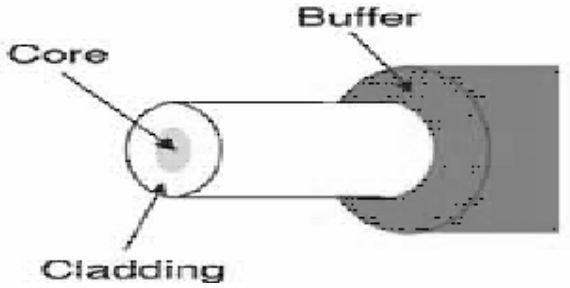


Figure 4.14 Fiber Optic cable

The light is "guided" down the center of the fiber called the "core". The core is surrounded by an optical material called the "cladding" that traps the light in the core using an optical technique called "total internal reflection." The core and cladding are usually made of ultra-pure glass, although some fibers are all plastic or a glass core and plastic cladding. The fiber is coated with a protective plastic covering called the "primary buffer coating" that protects it from moisture and other damage.

Transparent glass or plastic fibers, which allows light to be guided from one end to the other with minimal loss.

Fiber optic cable functions as a "light guide," guiding the light introduced at one end of the cable through to the other end. The light source can either be a light-emitting diode (LED)) or a laser. The light source is pulsed on and off, and a light-sensitive receiver on the other end of the cable converts the pulses back into the digital ones and zeros of the original signals.

While fiber optic cable itself has become cheaper over time - an equivalent length of copper cable cost less per foot but not in capacity. Fiber optic cable connectors and the equipment needed to install them are still more expensive than their copper counterparts.

The bandwidth of a fiber-optic cable depends on the distance as well as the frequency. Bandwidth is usually expressed in frequency distance form, for example in MHz-km. In other words, a 500-MHz-km fiber-optic cable can transmit a signal a distance of 5 kilometers at a frequency of 100 MHz (5 x 100 = 500), or to a distance of 50 kilometers at a frequency of 10 MHz (50 x 10 = 500). In other words, there is an inverse relationship between frequency and distance for transmission over fiber-optic cables.

Propagation Mode

There are two different modes for propagating light along optical channels: *multimode* and *single mode*. There are two basic types of fiber: multimode fiber and single-mode fiber.

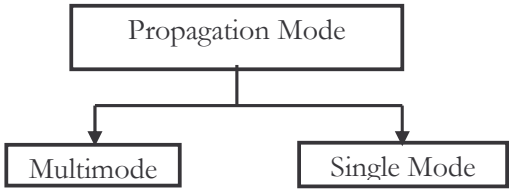


Figure 4.15 Propagation modes

Multimode

Multimode is so named because multiple beams from a light source move through the core in different paths. Multimode cable is made of glass fibers, with a common diameters in the 50-to-100 micron range for the light carry component (the most common size is 62.5).

Multimode fiber gives you high bandwidth at high speeds over medium distances. Light waves are dispersed into numerous paths, or modes, as they travel through the cable's core typically 850 or 1300 nm. Typical multimode fiber core diameters are 50, 62.5, and 100 micrometers. However, in long cable runs (greater than 3000 feet [914.4 meter), multiple paths of light can cause signal distortion at the receiving end, resulting in an unclear and incomplete data transmission.

In **multimode step-index fiber**, the density of the core remains constant from the center to the edges.

A beam of light moves through this constant density in straight line until it reaches the interface of the core and the cladding. At the interface, there is an abrupt change to a lower density that alters the angle of beam's motion. The term *step-index* refers to the suddenness of this change.

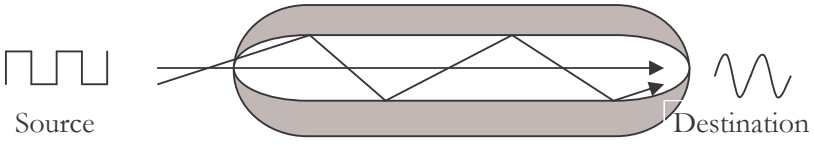


Figure 4.16 Multimode, Step-index fiber

Step-index multimode fiber has a large core, up to 100 microns in diameter. As a result, some of the light rays that make up the digital pulse may travel a direct route, whereas others zigzag as they bounce off the cladding. These alternative pathways cause the different groupings of light rays, referred to as modes, to arrive separately at a receiving point. The pulse, an aggregate of different modes, begins to spread out, losing its well-defined shape. The need to leave spacing between pulses to prevent overlapping limits bandwidth that is, the amount of information that can be sent. Consequently, this type of fiber is best suited for transmission over short distances, in an endoscope, for instance. It is less costly variety of multimode fiber, it uses a wide core with a uniform index of refraction, causing the light beams to reflect in mirror fashion off the inside surface of the core by the process of total internal reflection. Because light can take many different paths down the cable and each path takes a different amount of time, signal distortion can result when step-index fiber is used for long cable runs. Use this type only for short cable runs.

A second type of fiber, called **multimode graded index fiber**, decreases this distortion of the signal through the cable. The word *index* here refers to the index of refraction.

Index of refraction is related to density. A graded-index fiber, therefore, is one with varying density. Density is highest at the center of the core and decreases gradually to its lowest at the edge.

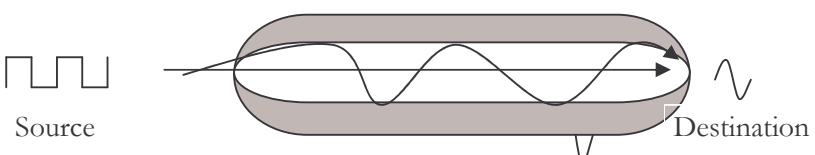


Figure 4.17 Multimode, graded-index fiber

Graded-index multimode fiber contains a core in which the refractive index diminishes gradually from the center axis out toward the cladding. The higher refractive index at the

center makes the light rays moving down the axis advance more slowly than those near the cladding. Also, rather than zigzagging off the cladding, light in the core curves helically because of the graded index, reducing its travel distance. The shortened path and the higher speed allow light at the periphery to arrive at a receiver at about the same time as the slow, but straight rays in the core axis. The result, a digital pulse suffers less dispersion.

Single Mode

Single mode uses step-index fiber and a highly focused source of light that limits beams to small range of angles, all close to the horizontal axis. Single Mode cable is a single stand of glass fiber with a diameter of 8.3 to 10 microns that has one mode of transmission. Single Mode Fiber with a relatively narrow diameter, through which only one mode will propagate typically 1310 or 1550 nm. Carries higher bandwidth than multimode fiber, but requires a light source with a narrow spectral width. Single-mode fiber is also called as mono-mode optical fiber, single-mode optical waveguide, unimode fiber.

The single mode fiber is manufactured with a much smaller diameter than that of multimode fibers, and with substantially lower density (index of refraction).

Figure 4.18 Single-mode fiber

The decrease in density results in a critical angle that is close enough to 90 degrees to make the propagation of beams almost horizontal.

Single-mode fiber gives you a higher transmission rate and up to 50 times more distance than multimode, but it also costs more. Single-mode fiber has a much smaller core than multimode. The small core and single light-wave virtually eliminate any distortion that could result from overlapping light pulses, providing the least signal attenuation and the highest transmission speeds of any fiber cable type.

Fiber Sizes

Optical fibers are defined by the ratio of the diameter of their core to the diameter of their cladding, both expressed in microns (micrometer).

Fiber Type	Core	Cladding
62.5/125	62.5	125
50/125	50.0	125
100/140	100.0	140
8.3/125	8.3	125

Table 4.3 Fiber Types

The last size listed is used only for single mode. Single mode fiber has a very small core causing light to travel in a straight line and typically has a core size of 8 or 10 microns.

Multimode fiber supports multiple paths of light and has a much larger core and has a core size of 50 or 62.5 microns.

Characteristics / Description	Single Mode Fiber	Multimode Fiber
Bandwidth	High	Lower
Signal Quality	High	Lower
Main Source of Attenuation	Chromatic Dispersion	Modal Dispersion
Fiber Designs	Step index, and Dispersion shifted	Step index and Graded index
Application	Long transmission, higher bandwidth	Short transmission, lower bandwidth
care/cladding	8.3/125	62.5/125
Light source	ILD	LED/ILD

Table 4.4 Single Mode and Multimode Characteristics

Light Sources for Optical Fiber

The purpose of fiber-optic cable is to contain and direct a beam of light from source to destination. For transmission to occur, the sending device must be equipped with a light source and the receiving device with a photosensitive cell (called a photodiode) capable of translating the received light into current usable by a computer. The light source can be either a **light-emitting diode (LED)** or an **injection laser diode (ILD)**.

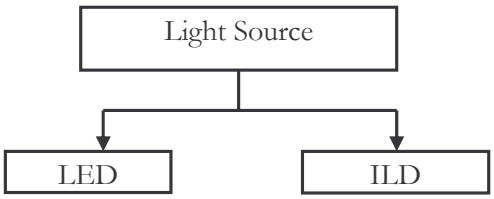


Figure 4.19 Light source in fiber optic cable

LEDs are the cheaper source, but they provide unfocused light that strikes the boundaries of the channel at uncontrollable angles and diffuses over distance. For this reason, LEDs are limited to short-distance use. Modulation bandwidth of LED is up to 100–200 MHz.

Problems with LEDs : Light-emitting volume is large, *p*oor coupling efficiency to fibers, low carrier density.

Lasers, on the other hand, can be focused to a very narrow range, allowing control over the angle of incidence. Laser signals preserve the character of the signal over considerable distances. Laser stands for **Light Amplification by Stimulated Emission of Radiation (LASER)**.

Every laser has a range of optical wavelengths, and the speed of light in fused silica (fiber) varies with the wavelength of the light. Since a pulse of light from the laser usually contains several wavelengths, these wavelengths tend to get spread out in time after traveling some distance in the fiber. The refractive index of fiber decreases as wavelength increases, so longer wavelengths travel faster. The net result is that the received pulse is wider than the transmitted one, or more precisely, is a superposition of the variously delayed pulses at the different wavelengths.

Applications of fiber-optic cable

- Fiber-optic cable is often found in backbone networks because its wide bandwidth is cost-effective. SONET network provides such backbone.
- Some cable TV companies use a combination of optical-fiber and coaxial cable.
- Telephone companies also using optical-fiber cable.
- Local Area Networks (LANs) such as 100BaseFx network (Fast Ethernet) and 1000Base-X also use fiber-optic cable.

Advantages of fiber-optic cable

1. **Higher Bandwidth :** Higher data rate than TP & coaxial cable.

2. **Less signal attenuation:** Fiber-optic transmission distance is significantly greater than that of other guided media. A signal can run for 50 km without requiring regeneration. We need repeaters after every 5km for coaxial or TP cable.

3. **Noise resistance :** Because fiber-optic transmission uses light rather than electricity, noise is not a factor. External light, the only possible interference, is blocked from the channel by the outer jacket.

4. **Light weight :** Fiber-optic cables are much lighter than copper cables.

5. **More immune to tapping (or Security) :** Fiber-optic cables are more immune to tapping than copper cables. Copper cables create antennas that can easily be tapped.

6. *Optical fiber can carry thousands of times more information than copper wire. For example, a single-strand fiber strand could carry all the telephone conversations in the United States at peak hour. Fiber is more lightweight than copper. Copper cable equals approximately 80 lbs/1000 feet while fiber weighs about 9 lbs/1000 feet.*

7. **Reliability** : Fiber optic is more reliable than copper and has a longer life span.

8. Fiber optic cable can carry signals for longer distance without repeater than co-axial cable.

Disadvantages of fiber-optic cable.

1. **Installation/maintenance expertise :** Installation and maintenance need expertise that is not yet available everywhere.
2. **Unidirectional :** Propagation of light is unidirectional.
3. **Cost :** Fiber-optic cable is more expensive.
4. **Fragility :** Glass fiber is more easily broken than wire, making it less useful for applications where hardware portability is required.
5. **Limited physical arc :** Bend the cable too much and it will break.

Trade-offs between electrical and optical cable

Electrical is cheaper, especially for short distances, because silicon circuits can send and receive over wires directly. Other semiconductor materials are required to implement the lasers for optical communication. Thus, optical requires multiple die, and has a higher base cost. Optical provides better performance at high-bandwidths and long distances.

Glass propagates light better than copper propagates electrical currents. Following table shows the comparison of guided media w.r.t the bandwidth.

Cable Type	Bandwidth
Open Cable	0 - 5 MHz
Twisted Pair	0 - 100 MHz
Coaxial Cable	0 - 600 MHz
Optical Fiber	0 - 1 GHz

Table 4.5 Cable types Vs bandwidth used

4.3 PERFORMANCE OF TRANSMISSION MEDIUM

Transmission media are roads on which data travel. To measure the performance of transmission media three concepts are used : *throughput*, *propagation speed*, and *propagation time*.

Throughput

The throughput is the measurement of how fast data can pass through a point. If we consider any point in the transmission medium as a wall through, which bits pass, throughput is the number of bits that can pass this wall in one second.

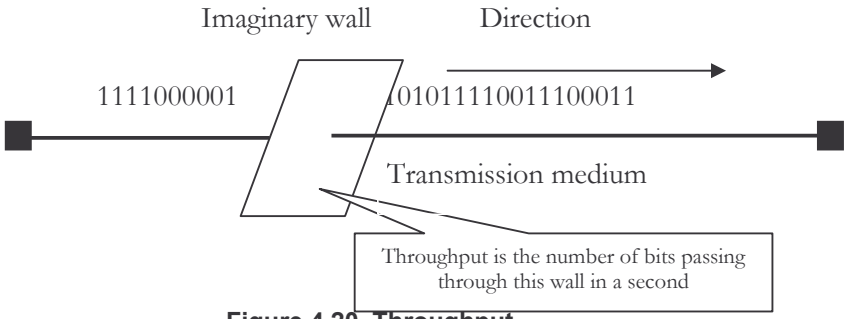


Figure 4.20 Throughput

Propagation speed

Propagation speed measures the distance a signal or a bit can travel through a medium in one second. The propagation speed of electromagnetic signals depends on the medium and on the frequency of the signal. For example, in a vacuum, light is propagated with a speed of 3×10^8 m/s. It is lower in air. It is much lower in a cable. In fiber-optic cable the speed is 2×10^8 m/s.

Propagation Time

Propagation time measures the time required for a signal (or a bit) to travel from one point of the transmission medium to another.

Propagation time is calculated as

Propagation time = Distance / Propagation speed

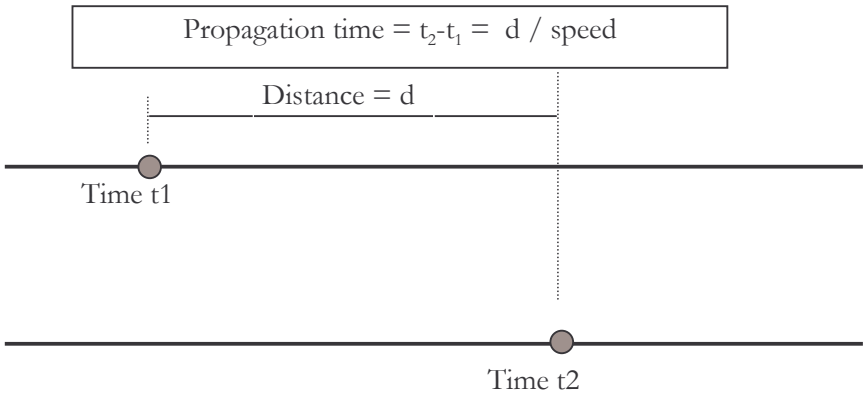


Figure 4.21 Propagation time

4.4 UNGUIDED MEDIA

Unguided media, or *wireless communication*, transport electromagnetic waves without using a physical conductor. Unguided Transmission Media consists of a means for the data signals to travel but nothing to guide them along a specific path. The data signals are not bound to a cabling media and as such are often called Unbound Media.

Signals are broadcast through air and thus are available to anyone who has a device capable of receiving them. In wireless communication, transmission and reception are achieved using an antenna. Transmitter sends out the electromagnetic signal into the medium. Receiver picks up the signal from the surrounding medium.

Wireless transmission can be divided into three groups: *radio waves*, *microwave*, and *infrared waves*. The section of the electromagnetic spectrum defined as radio communication is divided into eight ranges, called bands.

These bands are rated form very low frequency (VLF) to extremely high frequency (EHF).

EHF	Extremely High Frequency
SHF	Super High Frequency
UHF	Ultra High Frequency
VHF	Very High Frequency
HF	High Frequency
MF	Middle Frequency
LF	Low Frequency
VLF	Very Low Frequency

Figure 4.22 Radio Communication band

Satellite communication systems use UHF (Ultra High Frequency) or SHF (Super High Frequency) microwaves.

RADIO WAVES

Radio wave transmission utilizes five different types of propagation: *surface (or ground)*, *tropospheric*, *ionospheric*, *line-of-sight*, and *space*.

Radio technology considers the earth as surrounded by two layers of atmosphere : the troposphere and the ionosphere.

The *troposphere* is the portion of the atmosphere extending outward 30 miles from the earth's surface. Clouds, wind, temperature variation, and whether in general occur in the troposphere. The ionosphere is the layer of atmosphere above troposphere but below space.

Surface (or ground) propagation : Radio waves travel through the lowest portion of the atmosphere, hugging the earth. Distance cover by these signals depends on the amount of power in the signal: the greater the power, the greater the distance. The radio wave travels along the Earth's surface as a result of currents flowing in the ground. This is the dominant mechanism at low frequencies. e.g. Radio 4 $\lambda = 1500m \equiv 200kHz$. Surface propagation uses VLF (Very Low Frequency) & LF (Low Frequency) bands.

Tropospheric Propagation: It can work two ways. Either a signal can be directed in a straight line from antenna to antenna (line-of-sight), or it can be broadcast at an angle into the upper layers of the troposphere where it is reflected back down to the earth's surface. Tropospheric propagation uses MF (Middle Frequency) band.

Ionospheric Propagation : In ionospheric propagation, higher-frequency radio waves radiate upward into the ionosphere where they reflected back to earth. Radio waves can be reflected from the ionosphere. Example of total internal reflection, the refractive index gradually increases with height. The return wave can in turn be reflected back up again. The gap between the ionosphere and the ground acts as a **waveguide**. Inospheric propagation uses HF (High Frequency) band.

Line-of-Sight Propagation: In line-of-sight propagation, very high frequency signal are transmitted in straight line directly from antenna to antenna. Antennas must be directional, facing each other, and either tall enough or close enough together not to be affected by the curvature of the earth. Example of line-of sight system is microwave link using dishes and towers. A 60m-tower gives 60 km line of sight. Satellite communication is an extreme example of line-of-sight radio links. One tower is of height 35600km. Line – of – sight propagation uses VHF (Very High Frequency) & UHF (Ultra High Frequency) bands.

Space Propagation: Space propagation utilizes satellite relays in place of atmosphere refraction. A broadcast signal is received by an orbiting satellite, which rebroadcasts the signal to the intended receiver back on the earth. Radio waves, particularly those waves that propagate in sky mode, can travel long distance. This makes radio waves a good candidate for long-distance broadcasting such, as AM, FM radio.

Applications of Radio waves

Radio waves are used for multicast communication in which there is one sender but many receivers, such as AM and FM radio, television, cordless phone, and paging system.

MICROWAVE

Electromagnetic waves having frequencies between 1 GHz and 300 GHz are called microwaves. Microwaves do not follow the curvature of the earth and therefore require line-of-sight transmission and reception equipment. The distance coverable by a line-of-sight signal depends to a large extent on the height of antenna : the taller the antennas, the longer the sight distance. Height allows the signal to travel farther without being stopped by the curvature of the planet and raises the signal above many surface obstacles, such as low hills and tall buildings. Typically, antennas are mounted on towers that are in turn often mounted on hills or mountains. Microwaves are unidirectional. When an antenna transmits microwave waves, they can be narrowly focused.

To increase the distance a system of repeaters can be installed with each antenna. A signal received by one antenna can be converted back into transmittable form and relayed to the next antenna.

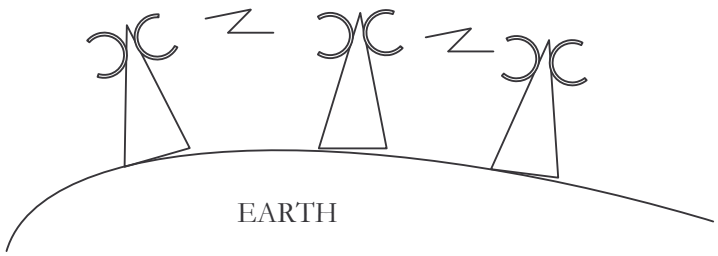


Figure 4.23 Microwave

Antennas used in microwave communications

1) Microwaves need unidirectional antennas that send out signals in one direction.
2) Two types of antennas are used for microwave communications : the parabolic dish and the horn.
3) A parabolic dish antenna is based on the geometry of a parabola: every line parallel to the line of symmetry (line of sight) reflects off the curve at angles such that they intersect in a common point called the focus.
4) The parabolic dish works as a funnel, catching a wide range of waves & directing them to a common point.
5) A horn antenna looks like a gigantic scoop. Outgoing transmission are broadcast up a stem and deflected outward in a series of narrow parallel beams by the curved head.

Applications of Microwaves

1) Microwaves, due to their unidirectional properties, are very useful when unicasting (one-to-one) communication is needed between the sender and the receiver.
2) Microwaves are used in cellular phones, satellite networks, and wireless LANs.

Advantages of Microwave :

1) They require no right of way acquisition between towers.
2) They can carry high quantities of information due to their high operating frequencies.
3) Low cost land purchase: each tower occupies only a small area.
4) High frequency/short wavelength signals require small antennae.

Disadvantages of microwave :

1) Attenuation by solid objects: birds, rain, snow and fog.
2) Reflected from flat surfaces like water and metal.
3) Diffracted (split) around solid objects.
4) Refracted by atmosphere, thus causing beam to be projected away from receiver.

Advantages of microwave over fiber optics :

1) No Need to dedicate complete physical path on land.
2) Putting up simple "tower" cheaper than laying cables.
3) Some frequency band does not need licensing to use.

INFRARED

- Infrared signals, with frequencies from 300 GHz to 400 GHz can be used for short-range communication.
- Infrared signals cannot penetrate walls. This advantageous characteristic prevents interference between one system and another: a short-range communication system in one room cannot be affected by another system in the next room.
- When we use our infrared remote control, we do not interfere with the use of the remote by our neighbors.
- This characteristic makes infrared signals useless for long-distance communication.
- We cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication.
- No licensing is required for infrared signals, that is, no frequency allocation issues with infrared signals
- Infrared (or milimeter) waves characteristics :
    - ✓ Used by remote controls for TV, VCRs, etc.
    - ✓ Cheap and easy to build.
    - ✓ Straight line, no obstacles - even more so than microwaves.
    - ✓ Used for wireless LANs within a room.

Applications of Infrared

- Infrared signals can be used for short-range communication in a closed area using line-of-sight propagation.
- The infrared band, almost 400 THz, has an excellent potential for data transmission.
- The Infrared Data Association (IrDA), an association for sponsoring the use of infrared waves, has established standard for using these signals for communications between devices such as keyboards, mice, PCs, and printers.

- For example, some manufactures provide a special port called the IrDA port that allows a wireless keyboard to communicate with a PC.

4.5 Satellite Communication

Not so long ago, satellites were exotic, top-secret devices. They were used primarily in a military capacity, for activities such as navigation and espionage. Now they are an essential part of our daily lives. We see and recognize their use in weather reports, television transmission by DIRECT TV and the DISH Network, and everyday telephone calls. In many other instances, satellites play a background role that escapes our notice :

- Some newspapers and magazines are more timely because they transmit their text and images to multiple printing sites via satellite to speed local distribution.
- Before sending signals down the wire into our houses, cable television depends on satellites to distribute its transmissions.
- Guided Missiles use the satellite-based Global Positioning System (GPS) to track the proper destination.
- Emergency radio beacons from downed aircraft and distressed ships may reach search-and-rescue teams when satellites relay the signal.

What is a Satellite ?

Satellite is basically any object that revolves around a planet in a circular or elliptical path. The moon is Earth's original, natural satellite, and there are many manmade (**artificial**) satellites, usually closer to Earth. The path a satellite follows is an **orbit**. In the orbit, the farthest point from Earth is the **apogee**, and the nearest point is the **perigee.** Artificial satellites generally are not mass-produced. Most satellites are **custom built** to perform their intended functions. Exceptions include the GPS (Global Positioning System) satellites (with over 20 copies in orbit) and the Iridium satellites (with over 60 copies in orbit).

Although anything that is in orbit around Earth is technically a satellite, the term "satellite" is typically used to describe a useful object placed in orbit purposely to perform some specific mission or task. We commonly hear about weather satellites, communication satellites and scientific satellites. The Soviet *Sputnik* satellite was the first to orbit Earth, launched on October 4, 1957.

Some Examples of Artificial satellites :

- 1957 - launch of SPUTNIK 1, Low Earth orbit (LEO), 200 to 600 km, period 90mins.
- 1958-64 - early developments mainly related to space race !
- TELSTAR I elliptical orbit 960 to 6080 km, period 2 hr 38 minutes.
- 1965 - INTELSAT I (Early Bird). First geosynchronous satellite that provided a routine link between USA and Europe for 4 years

INTELSAT - International Telecommunications Satellite Organization. More than 110 countries are members of this organization. The INTELSAT is responsible for providing communication links between its members, and member's hires out a service.

In satellite transmission signals travel in straight lines, the limitations imposed on distance by the curvature of the earth are reduced. Satellite communication is an extreme

example of line-of-sight radio links. One tower is of height 35600km. Satellite relays allow microwave signals to span continents and oceans with a single bounce. Satellite communication systems use UHF (Ultra High Frequency) or SHF (Super High Frequency) microwaves. This ensures that they penetrate the ionosphere and provides a large bandwidth.

A satellite network is a combination of nodes that provides communication from one point on the earth to another. A node in the network can be satellite, an earth station, or an end-user terminal or telephone.

Although a real satellite, such as the moon, can be used as a relaying node in the network, the use of artificial satellite is preferred because we can install electronic equipment on the satellite to regenerate the signal that has lost its energy during travel. The relay function of the satellite communications system is to receive the up-link signal from the ground, amplify it, change its frequency and retransmit it to the ground. Another restriction on using natural satellites is their distances from the earth, which create a long delay in communication.

Satellite can provide transmission capability to and from any location on earth, no matter how remote. This advantage makes high quality communication available to undeveloped parts of the world without requiring a huge investment in ground-based infrastructure.

Physical description

- Communication satellite is a microwave relay station between two or more ground stations (also called earth stations).
- Satellite uses different frequency bands for incoming (uplink) and outgoing (downlink) data.
- A single satellite can operate on a number of frequency bands, known as transponder channels or transponders.
- Geosynchronous orbit (35,784 km).
- Satellites cannot be too close to each other to avoid interference : This limits the number of available satellites.

Transmission characteristics

- Optimum frequency range in 1 Ghz to 10 GHz.
- Below 1 GHz, significant noise from galactic, solar, and atmospheric noise, and terrestrial electronic devices.
- Most satellites use 5.925 to 6.425 GHz band for uplink and 4.2 to 4.7 GHz band for downlink.
- Propagation delay of about a quarter second due to long distance.
- Problems in error control and flow control.
- Inherently broadcast, leading to security problems.

Orbits

An artificial satellite needs to have an **orbit**, the path in which it travels around the earth.

Geosynchronous orbits (also called **synchronous** or **equatorial-orbit**) are orbits in which the satellite is always positioned over the same spot on Earth. A geosynchronous orbit is one for which the orbital period of the spacecraft is the time taken for the Earth to complete 360^{o} rotation.

Geostationary orbits

This is a special case of the geosynchronous orbit. In such an orbit the satellite remains above the same point on the ground all the time. Geostationary orbits are 36,000 km from the Earth's surface. At this point, the gravitational pull of the Earth and the centrifugal force of Earth's rotation are balanced and cancel each other out. Centrifugal force is the rotational force placed on the satellite that wants to fling it out into space. Many geostationary satellites are above a band along the equator, with an altitude of about 22,223 miles, or about a tenth of the distance to the Moon. The "satellite parking strip" area over the equator is becoming congested with several hundred television, weather and communication satellites ! This congestion means each satellite must be precisely positioned to prevent its signals from interfering with an adjacent satellite's signals. Television, communications and weather satellites all use geostationary orbits. Geostationary orbits are why a DSS satellite TV dish is typically bolted in a fixed position.

The scheduled Space Shuttles use a much lower, **asynchronous (or inclined)** orbit, which means they pass overhead at different times of the day. Other satellites in asynchronous orbits average about 400 miles (644 km) in altitude.

In a **polar** orbit, the satellite generally flies at a low altitude and passes over the planet's poles on each revolution. The polar orbit remains fixed in space as Earth rotates inside the orbit. As a result, much of Earth passes under a satellite in a polar orbit. Because polar orbits achieve excellent coverage of the planet, they are often used for satellites that do mapping and photography

Artificial satellites which orbit the earth follow the same laws that govern the motion of the planets around the sun. Johannes Kepler (1571-1630) was derived law called as Kepler's law, describes planetary motion. The period of a satellite, the time required for a satellite to make a complete trip around the earth, is determined by **Kepler's law**, which defines the period as a function of the distance of the satellite from the center of the earth.

$$\textbf{Period} = C \times \textbf{distance}^{1.5}$$

Where C is a constant approximately equal to 1 /100. The period is in seconds and the distance in kilometers.

Example 1 : What is the period of the moon according to Kepler's law ?

Solution :

The moon is located approximately 3,84,000 km above earth.

The radius of the earth is 6378 km.

$$\text{Period} = C \times \text{distance}^{1.5}$$
$$= (1/100) \times (3,84,000 + 6378)^{1.5}$$
$$= 24,39,090 \text{ sec}$$
$$= 1 \text{ month}$$

Example 2 : According to Kepler's law, what is period of a satellite that is located at an orbit approximately 35,786 km above the earth?

Solution : Period = $C \times \text{distance}^{1.5}$

$= (1/100) \times (35{,}786 + 6378)^{1.5}$

= 86,579 sec

= 24 hrs

This means that a satellite located at 35,786 km has a period of 24 hrs, which is the same as the rotation period of the earth. A satellite like this is said to be geo-*stationary* to the earth.

4.6 Geostationary Satellite

The point 36,000 km from the Earth's surface, the gravitational pull of the Earth and the centrifugal force of Earth's rotation are balanced and cancel each other out. Centrifugal force is the rotational force placed on the satellite that wants to fling it out into space. Many geostationary satellites are above a band along the equator, with an altitude of about 22,223 miles, or about a tenth of the distance to the Moon. The "satellite parking strip" area over the equator is becoming congested with several hundred televisions, weather and communication satellites! This congestion means each satellite must be precisely positioned to prevent its signals from interfering with an adjacent satellite's signals. Television, communications and weather satellites all use geostationary orbits. Geostationary orbits are why a DSS satellite TV dish is typically bolted in a fixed position.

4.7 COMPARISONS

Comparison of Step Index and Graded Index Fiber

No	Step Index Fiber	Graded Index Fiber
1.	Data rate is slow.	Data rate is higher.
2.	Normally plastic or glass is preferred.	Only glass is preferred.
3.	Coupling efficiency with fiber is higher.	Lower coupling efficiency.
4.	Pulse spreading by fiber length is more.	Pulse spreading by fiber length is less.
5.	Typical light source is LED.	LED and Laser.
6.	Attenuation of light source is less, typically 0.34 dB/km at 1.3 μm.	Attenuation of light source is more, typically 0.6 to 1 dB/km at 1.3 μm.
7.	The light rays travel in a straight line due to constant refractive index of the fiber throughout the bulk of the core.	The light rays do not travel in a straight line due to continuous refraction. This is due to continuously changing refractive index throughout the core bulk.
8.	Used in subscribers local area network communication	Local and wide are networks.

Comparison of Satellite Communication and Optical Communication

No	Satellite Communication	Optical Communication
1.	The communication takes place via satellite acting as a relay station.	The communication takes place via an optical fiber.

No	Satellite Communication	Optical Communication
2.	The communication takes place by means of electromagnetic waves.	The communication takes place by means of light rays.
3.	Broadcasting is possible.	Broadcasting is not possible.
4.	Transmission medium is air.	Transmission medium is fiber optic cable.
5.	Satellite communication is useful for very long distance communication.	It is useful for point-to-point short distance communication.
6.	Transponder is an interface device between transmitter and receiver.	Optical amplifier or regenerators are the intermediate device between transmitter and receiver.
7.	Antennas are required for transmission and reception.	Antenna is not required.
8.	Installation and operating cost is very high.	Installation and operating cost is very less as compared to satellite communication.

Comparison of Single mode and Multimode fiber

No	Single Mode Fiber	Multimode Fiber
1.	These fibers support only one mode of propagation.	These fibers support the propagation of many modes.
2.	The traveling signal inside the fiber has only one group velocity.	The different modes have different group velocities and each mode will follow its own path between the transmitter and receiver.
3.	Intermodal dispersion does not present.	Intermodal dispersion exists.
4.	These are high quality fiber for wideband long haul transmission and are fabricated from doped silica for reducing the attenuation.	These are fabricated using the multicomponent glasses.

Comparison of LED and Laser Diode

No	Light Emitting Diode (LED)	Laser Diode (LD)
1.	Spontaneous emission.	Stimulated emission.
2.	Output beam is non-coherent.	Output beam is coherent.
3.	Broad spectrum (20 nm to 100 nm)	Much narrow spectrum (1 to 5 nm)
4.	Data rate is low.	Data rate is very high.
5.	Smaller transmission distance.	Very large transmission distance.
6.	Less temperature sensitivity.	More temperature sensitivity.
7.	Low cost	High cost.
8.	Used in moderate distance low data rate application.	Used in long distance high data rate application.

Comparison of Guided medias

No	Twisted-Pair Cable	Coaxial Cable	Fiber Optic Cable (FOC)
1.	It uses electrical signals for transmission.	It uses electrical signals for transmission.	It uses optical form of signal (i.e. light) for transmission.
2.	It uses metallic conductor to carry the signal.	It uses metallic conductor to carry the signal.	It uses glass or plastic to carry the signal.
3.	Noise immunity is low. Therefore more distortion.	Higher noise immunity than twisted-pair cable due to the presence of shielding conductor.	Highest noise immunity as the light rays are unaffected by the electrical noise.
4.	Affected due to external magnetic filed.	Less affected due to external magnetic filed.	Not affected by the external magnetic filed.
5.	Cheapest	Moderately costly	Costly
6.	Can support low data rates.	Moderately high data rates.	Very high data rates.
7.	Power loss due to conduction and radiation.	Power loss due to conduction.	Power loss due to absorption, scattering, dispersion.
8.	Short circuit between two conductors is possible.	Short circuit between two conductors is possible.	Short circuit is not possible.
9.	Low bandwidth.	Moderately high bandwidth.	Very high bandwidth.

Comparison of Wired and Wireless Media

No	Wired Media	Wireless Media
1.	The signal energy is contained and guided within a solid medium.	The signal energy propagates in the form of unguided electromagnetic waves.
2.	Used for point-to-point communication	Used for broadcasting.
3.	Twisted-pair cable, coaxial cable, fiber optical cables are example of wired media.	Radio and infrared light are the examples of wireless media.
4.	Attenuation depends exponentially on the distance.	Attenuation is proportional to square of distance.

Summary

Transmission medium

- Physical path between transmitter and receiver.
- May be guided (wired) or unguided (wireless).
- Communication achieved by using EM waves.
- Characteristics and quality of data transmission.
- Dependent on characteristics of medium and signal.

Guided medium.

- Medium is more important in setting transmission parameters.

Unguided medium.

- Bandwidth of the signal produced by transmitting antenna is important in setting transmission parameters.
- Signal directionality.
- Lower frequency signals are omnidirectional.
- Higher frequency signals can be focused in a directional beam.

There 4 basic types of Guided Media : Open Wire, Twisted Pair, Coaxial Cable, and Optical Fiber. Coaxial cable (or coax) carries signals of higher frequency ranges than twisted-pair cable. A fiber-optic cable is made of glass or plastic and transmits signals in the form of light.

PRACTICE SET

Review Questions

1. Is the transmission media a part of the physical media? Why or why not?
2. Name two major categories of transmission media.
3. How do guided media differs from unguided media?
4. Write a short note on TP cable
5. Write a short note on coaxial cable
6. Write a short note on Fiber optic Cable.
7. Explain Advantages and Disadvantages of TP, coaxial and Fiber optic cable.
8. Give a use for each class of guided media.
9. What is major advantages of STP over UTP ?
10. What is the significance of the twisting in TP cable?
11. Why is coaxial cable is superior to TP cable ?
12. What is the cladding in an optical cable ?
13. How does the sky propagation differ from line-of-sight propagation ?
14. What is an IrDA port ?
15. Explain various categories of UTP cable and their use.
16. Write a short note on light sources and detectors used in optical cable.
17. Write a short note on Kepler's laws and orbital aspect.
18. Write a short note on Geostationary Satellite.
19. Explain different types of unguided media.

Multiple Choice Questions

1. Transmission media are usually categorized as …….

A] fixed or unfixed B] guided or unguided

C] determinate or indeterminate D] metallic or nonmetallic

2. Transmission media lie below the ……. layer.

A] Physical B] Data Link Layer C] Network D] Transport

3. In fiber optics, the signal is ……. waves.

A] Light B] radio C] infrared D] microwave

4. In copper cable, the signal is ……. waves.

A] Light B] Electric C] Infrared D] microwave

5. Which of the following primarily uses guided media?

A] cellular telephone system B] local telephone system

C] satellite communication D] radio broadcasting

6. Which of the following is not a guided medium?

A] twisted-pair B] coaxial C] fiber-optic D] atmosphere

7. In an optical fiber, the inner core is ……. the cladding.

A] denser than B] less dense than

C] same density D] another name for

8. The inner core of an optical fiber is ……. in composition.

A] glass or plastic B] copper C] liquid D] bimetallic

9. When a beam of light travels through media of two different densities, if the angle of incidence is greater than the critical angle, ……. occurs.

A] refraction B] reflection C] incidence D] criticism

10. ……. medium provides a physical conduit from one device to another.

A] guided B] unguided

C] either option A) or B) D] none of the above

11. ……. cable consists of two insulated copper wires twisted together.

A] coaxial B] fiberC] twisted pair D] none of the above

12. ……. cables are composed of a glass or plastic inner core surrounded by cladding, all encased in an outside jacket.

A] fiber B] coaxial C] twisted-pair D] none of the above

13. ……. cables carry data signals in the form of light.

A] fiber B] coaxial C] twisted-pair D] none of the above

14. In a fiber-optic cable, the signal is propagated along the inner core by …….

A] refraction B] modulation C] reflection D] none of the above

15. ……. media transport electromagnetic waves without the use of a physical conductor.

A] guided B] unguided C] either Option A) OR B) D] none of the above

16. ……. are used for short-range communications such as those between a PC and a peripheral device.

A] radio waves B] microwaves C] infrared waves D] none of the above

* * *

Chapter 5 Introduction to Computer Networking

Although the computer industry is still young compared to other industries (e.g., automobiles), computers have made spectacular progress in a short time. During the first two decades of their existence, computer systems were highly centralized, usually within a single large room. Not infrequently, this room had glass walls, through which visitors could gawk at the great electronic wonder inside. A medium-sized company or university might have had one or two computers, while large institutions had at most a few dozen. The idea that within twenty years equally powerful computers smaller than postage stamps would be mass produced by the millions was pure science fiction.

The merging of computers and communications has had a profound influence on the way computer systems are organized. The concept of the "computer center" as a room with a large computer to which users bring their jobs for processing is now totally obsolete. The old model of a single computer serving all of the organization's computational needs has been replaced by one in which a large number of separate but interconnected computers do the job. These interconnected systems is called computer network. The design and organization of these networks are the subjects of this chapter.

OBJECTIVES OF THIS CHAPTER

After completing this chapter, the student should be able to:

- identify need of the network
- define advantages of the network
- define point-to-point or multipoint configuration of the link
- identify network topologies
- define types of networks

Introduction

Data communication is the exchange of data between two devices via some form of transmission media such as wire cable. Or in other words we can also say, *Data Communication* is the transfer of information from one computer to another over a Communications link. The transfer can be occasional, continuous, or a combination of both. But to transfer the information from one device to another device, first we must have to create network among such devices.

Now we will see different terminologies of concept 'Network'.

- Computers are information tools, and networks are how the computers exchange that information.
- A network is a set of devices (often referred as nodes) connected by communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.
- A network is a set of computers, which are linked together on a permanent basis. This can mean two computers cabled together on the same desk or thousands of computers across the world.
- Networked computers can share data and peripherals (resource sharing), allowing people in an organization to communicate better and more effectively use their hardware resources.

5.1 Computer Network Classification

A group of computers connected in some fashion in order to share resources. Networks with enhanced storage and server-based processing power provide users with greater functionality and security than independent or stand-alone machines.

Networks can be classified based on:

- **Physical size:** They can be classified as local area networks (LANs), metropolitan area networks (MANs), or wide area networks (WANs).
- **Topology:** The physical layout of computers, cables, and other components of a network. Topologies (the manner and complexity of interconnections) include the bus topology, star topology, ring topology, and mesh topology.
- **Security:** Networks can be based on workgroups, in which all machines control their own security, or they can be based on servers (such as domain-based Microsoft Windows 2000/2003 networks).
- **Network architecture:** For example, networks can be classified as Ethernet, Token Ring, or Fiber Distributed Data Interface (FDDI) networks. Network Architecture is an umbrella term describing the topologies, access methods, protocols, and other technologies used for networking.

  The following are examples of network architectures for Local Area Networks (LANs): **Ethernet:** By far the most popular network architecture for LANs. Ethernet supports speeds of 10, 100, and 1000 Mbps and is based on the contention method of media access control. **Token Ring:** It's an architecture developed by IBM. Legacy IBM networks support speeds of 4 and 16 Mbps, although vendors are working on standards for much higher speeds. **AppleTalk:** An architecture developed by Apple for its Macintosh platform that is essentially a protocol suite that can run over Ethernet, Token Ring, or Fiber Distributed Data Interface (FDDI) networks.

- **Network protocol:** A network protocol is a set of rules that governs the communications between computers on a network. These rules include guidelines that regulate the following characteristics of a network : access method, allowed physical topologies, types of cabling, and speed of data transfer. Networks can be classified based on the protocol they use ex : Internet Protocol (IP), Internetwork Packet Exchange (IPX), Systems Network Architecture (SNA), AppleTalk networks, and so on.

A network consists of two or more computers that are linked in order to share resources (such as printers and CD-ROMs), exchange files, or allow electronic communications. The computers on a network may be linked through cables, telephone lines, radio waves, satellites, or infrared light beams.

Network is a group of computers and associated peripheral devices connected by a communications channel capable of sharing files and other resources among several users. A network can range from a peer-to-peer network connecting a small number of users in an office or department, to a LAN connecting many users over permanently installed cables and dial-up lines, to a MAN or WAN connecting users on several

networks spread over a wide geographic area. The purpose of a computer network is to link two or more "clients" together in order to exchange information.

Uses of Computer Network

1. Resource Sharing

Enables users to **share hardware** like scanners and printers on the network. This reduces costs by reducing the number of hardware items bought.

Resources are available to anyone on the network without regard to the physical location of the resource and the user.

2. Information Sharing

Allows users to **access to data** stored on other computers on network. This keeps everyone up-to-date with latest data, since it's all in the same file, rather than having to make copies of files, which are immediately out-of-date

For example – files, database, records, etc.

3. Person-to-person communication

For example – e-mail (electronic mail), Videoconferencing, Teleconferencing, etc.

4. Electronic business

Users can place orders electronically as when needed. 365 days 7 days of week and 24 hours of Day.

5. Interactive entertainment

Real-time simulation games, like flight simulators, Age of empires.

Advantages of Computer Network

1 **Speed**. The network provides path for sharing resources and the speed of data transfer is very good. Without a network, files are shared by copying them to floppy disks, then carrying or sending the disks from one computer to another. This method of transferring files is referred to as sneaker-net. It is very time-consuming and expensive.

2. **Cost :** Network versions of many popular software programs are available at considerable savings when compared to buying individually licensed copies. Besides monetary savings, sharing a program on a network allows easier upgrading of the program. The changes have to be done only once, on the server, instead of on all the individual workstations.

3. **Security :** Files and programs on a network can be designated as "copy inhibit," so that you do not have to worry about illegal copying of programs. Also, passwords can be established for specific directories to restrict access to authorized users. A good encryption together with network packets switching allows security from tapping or listening.

4. **Centralized Software Management :** One of the greatest benefits of a network is the fact that it all of software's to be loaded on one computer (the *file server*). This eliminates that need to spend time and energy installing updates and tracking files on independent computers throughout the building or campus.

5. **Resource Sharing :** Sharing resources is another area in which a network exceeds stand-alone computers. Most organizations cannot afford enough laser

printers, fax machines, modems, scanners, and CD-ROM players for each computer. However, if these or similar peripherals are added to a network, they can be shared by many users.

6. **Electronic Mail :** The presence of a network provides the hardware necessary to install an e-mail system. Due to E-mail systems person to person communication became easy.
7. **Flexible Access :** Same organizations networks allow authorized users to access their files from computers throughout the network of organization.
8. **Workgroup Computing :** Workgroup software (such as *Microsoft BackOffice*) allows many users to work on a document or project concurrently. For example, educators located at various schools within a county could simultaneously contribute their ideas about new curriculum standards to the same document and spreadsheets.

Disadvantages of Network

1. **Expensive to Install :** Although a network will generally save money over time, the initial costs of installation can be prohibitive. Cables, network cards, and software are expensive, and the installation may require the services of a technician.
2. **Requires Administrative Time :** Proper maintenance of a network requires considerable time and expertise.
3. **Server May Fail :** Although a server is no more susceptible to failure than any other computer, when the server (domain controller) "goes down," the entire network may come to a halt.

5.2 LINE CONFIGURATIONS

Line configuration is a way to connect two or more communication devices attached to a link. There are two types of line configurations namely: point-to-point and multipoint line configuration.

POINT-TO-POINT

Point to Point connection provides a dedicated link between two devices. Point-to-point link means a direct connection between two, and only two, locations or nodes. Point-to-point is a form of communication that provides a direct path from one fixed point to another point.

Figure 5.2 Point-to-point communication link

MULTIPOINT

A **multipoint (**also called **multidrop)** in which more than two specific devices share a single link. Multidrop line is a circuit connecting several devices or nodes on a single logical link; also called a multipoint line. A

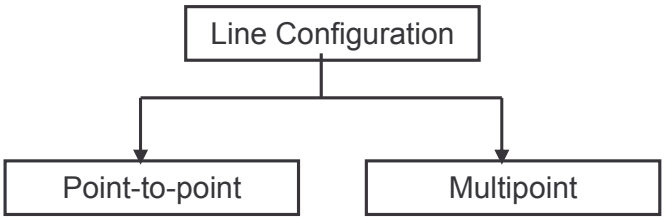


Figure 5.1 Types of Line Configuration

multidrop line is often used in IBM's SNA (Systems Network Architecture). It is controlled by a primary station, and the other nodes are considered secondary.

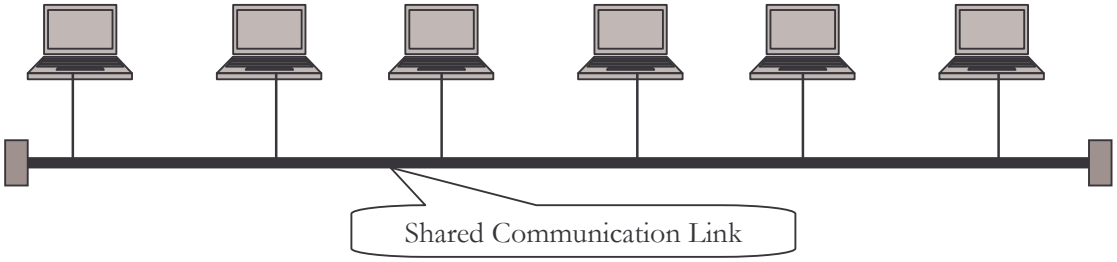


Figure 5.3 Multipoint communication link

5.3 NETWORK TOPOLOGIES

Two or more devices connected to a link; two or more links form a topology. The physical topology of a network refers to the configuration of cables, computers, and other peripherals.

The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called *nodes*) to one another. A node is an active device connected to the network, such as a computer or a printer. A node can also be a piece of networking equipment such as a hub, switch or a router. Topology defines the physical or logical arrangement of links in a network.

A topology defines how nodes/stations are connected. Topology is the map of a network. Physical topology describes where the cables are run and where the workstations, nodes, routers, and gateways are located. Networks are usually configured in bus, ring, star, or mesh topologies. Logical topology refers to the paths that messages take to get from one user on the network to another.

In short, topology is the physical layout of computers, cables, switches, routers, and other components of a network. This term can also refer to the underlying network architecture, such as Ethernet or Token Ring. The word "topology" comes from topos, which is Greek for "place."

Basic network topologies include the following :

- Bus Topology
- Star Topology
- Ring Topology
- Mesh Topology

Extended network topologies include the following:

- Tree Topology
- Hybrid Topology

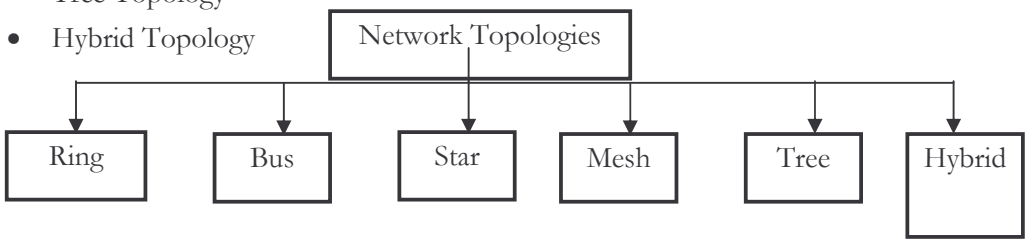


Figure 5.4 Network Topologies

RING TOPOLOGY

Each device has a dedicated point-to-point line configuration only with the two devices on either side of it. (Dedicated means that the link carries traffic only between the two devices is connects.) Nodes form a ring by point-to-point links to adjacency neighbors. A signal is passed along the ring in one direction, from device to device, until it reaches its destination.

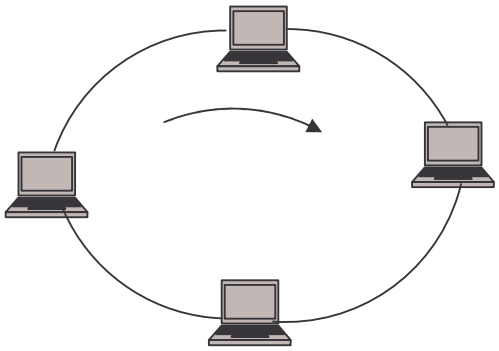


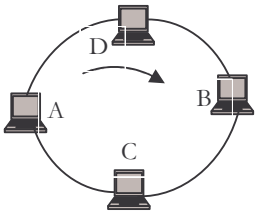
Figure 5.5 Ring Topology

Each device or computer in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along. Each computer acts as a repeater and keeps the signal strong, that is, there is no need for repeaters on a ring topology. Token passing is used in Token Ring networks. The token is passed from one computer to the next, only the computer with the token can transmit. The receiving computer strips the data from the token and sends the token back to the sending computer with an acknowledgment. After verification, the token is regenerated.

Ring topology is a network topology in the form of a closed loop or circle, with each node in the network connected to the next. Messages move in one direction around the system. When a message arrives at a node, the node examines the address information in the message. If the address matches the node's address, the message is accepted; otherwise, the node regenerates the signal and places the message back on the network for the next node in the system. It is this regeneration that allows a ring network to cover greater distances than star networks or bus networks. The failure of a single node can disrupt network operations.

Ring topology usually seen in a Token Ring or FDDI (fiber optic) network.

Example of ring topology : In ring topology each node functions as a repeater. Suppose ring operates in clockwise direction i.e. data is transferred from one node to another in clockwise direction and node "B" wants to transmit data to node "A". Node "B" will first prepare the frame, and then forward it towards node "C". When Node "C" receives the frame, First Node "C" examines the frame and as frame is not intended for Node "C", Node "C" ignores it. Node "C" simply forwards the frame after regenerating it to node "A". Node "A" accepts the frame, because it's intended for it.



Figurer (a) Ring

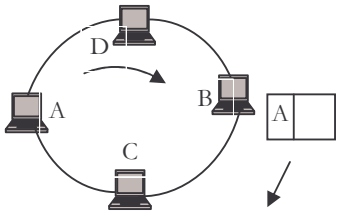


Figure (b) B send frame to A via C

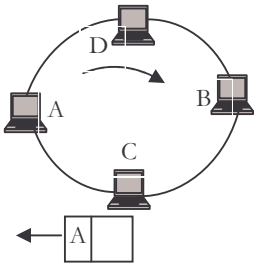


Figure (c) C forward frame to A

Figure 5.6 Example of Ring Topology

Advantages

1. Require less cabling so is less expensive.
2. Fault isolation is simplified. Generally in a ring, a signal is circulating at all times. If one device does not receive a signal within a specified period, it can issue an alarm. The alarm alerts the network operator to the problem and its location.

Disadvantages

1. Traffic is unidirectional.
2. If one node goes down, it takes down the whole network.
3. Slow.
4. Reconfiguration : To add one node, whole network must be down first and after addition the network starts working.

BUS TOPOLOGY (OR LINEAR BUS TOPOLOGY, OR HORIZONTAL TOPOLOGY))

In networking, a topology that allows all network nodes to receive the same message through the network cable at the same time is called as bus topology. The bus pattern connects the computer to the same communications line. In bus topology all nodes/stations are connected to common link/medium. Communications goes both directions along the line. It is a multipoint configuration.

One long cable acts as a backbone to link all the devices in a network.

Nodes are connected to the bus by drop lines and tap. A drop line is a connection running between a device and the main cable. A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core.

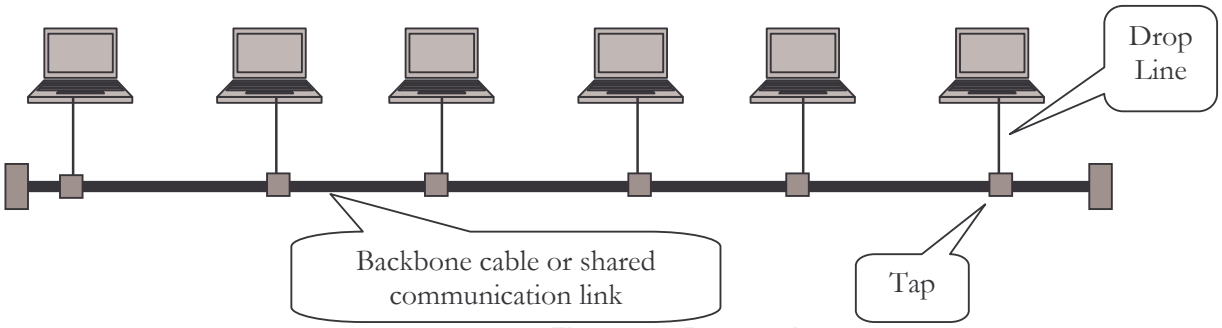


Figure 5.7 Bus topology

A bus topology consists of a main run of cable with a terminator at each end. All nodes (file server, workstations, and peripherals) are connected to the linear cable. Ethernet network use a linear bus topology. A network that uses a bus topology is referred to as a "Bus Network." As a signal travels along the backbone, some of its energy is transformed into heat. Therefore, it becomes weaker and weaker as it has to travel farther and farther. For this reason there is limit on the number of taps a bus can support and on the distance between those taps. Bus topology is the cheapest way of connecting computers to form a workgroup or departmental LAN, but it has the disadvantage that a single loose connection or cable break can bring down the entire LAN.

Bus consists of a single linear cable called a trunk. Data is sent to all computers on the trunk. Each computer examines EVERY packet on the wire to determine whom the packet is for and accepts only messages addressed to them. Bus is a passive topology. Performance degrades as more computers are added to the bus. Signal bounce is eliminated by a terminator at each end of the bus. Barrel connectors can be used to lengthen cable. Repeaters can be used to regenerate signals. Usually bus topology uses *Thinnet* or *Thicknet*. Both of these require 50-ohm terminator. Bus topology is good for a temporary, small (fewer than 10 people) network. But it's difficult to isolate malfunctions and if the backbone goes down, the entire network goes down.

Example of Bus topology : Suppose node "A" wants to transfer data to node "D". In bus topology all nodes will receive the packet sent by node "A" to node "D" because of common/same medium/link used by all nodes. Node "B", node "C", node "E" and node "F" will reject the packet while only node "D" will accept the packet.

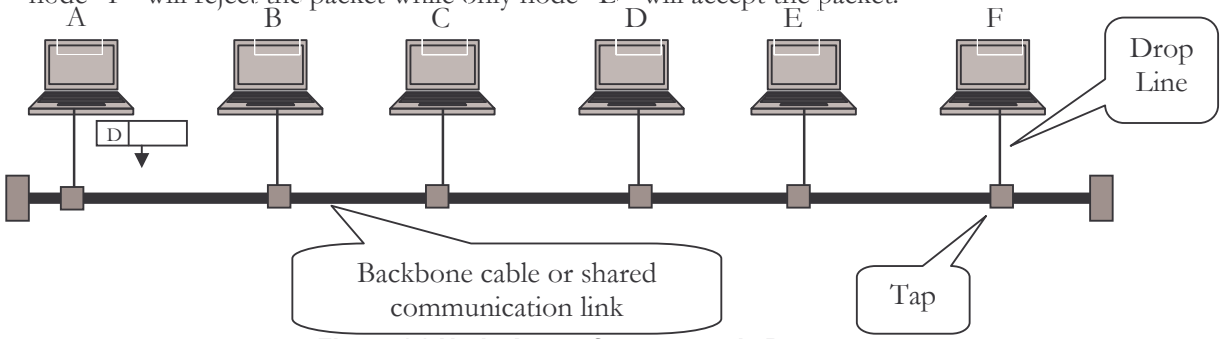


Figure (a) Node A sent frame to node D

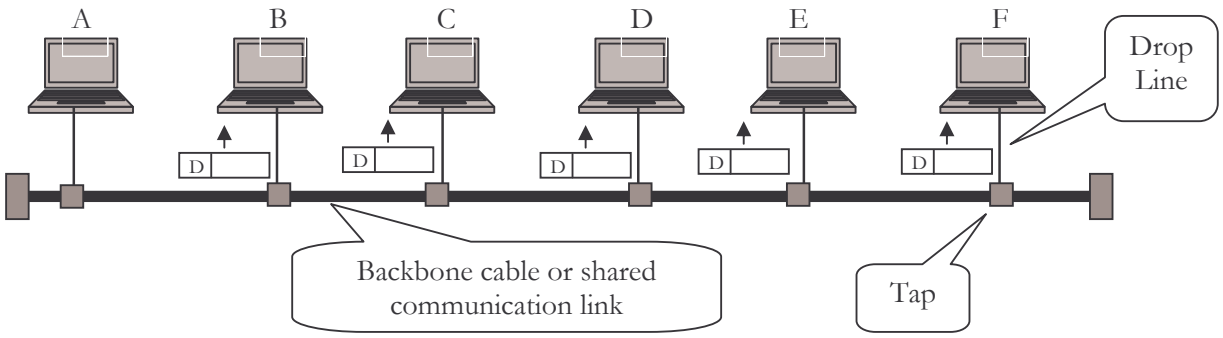


Figure (b) all node rejected frame except D

Figure 5.8 Example of Bus topology

Advantages

1. Easy to install. and connect computers or peripherals to a bus.
2. Requires less cabling length so cheaper.
3. Any one computer or device being down does not affect the others.
4. Fast as compare to ring topology.

Disadvantages

1. Can't connect a large number of computers.
2. Difficult faulty isolation. A fault or break in the bus cable stops all transmission. Difficult to identify the problem if the entire network shuts down.
3. Collision may occur.
4. Signal reflection at the taps can cause degradation in quality.
5. Entire network shuts down if there is a break in the main cable.
6. Terminators are required at both ends of the backbone cable.
7. Not meant to be used as a stand-alone solution in a large building.

STAR TOPOLOGY

The devices are not directly linked to one another. A star topology does not allow direct traffic between devices. Each device has a dedicated point-to-point link to central controller, usually called a **hub** or **switch.** The controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device.

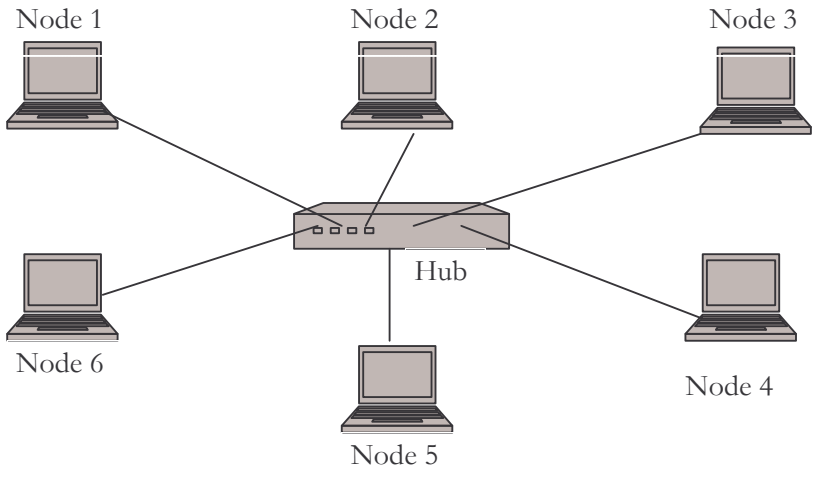


Figure 5.9 Star Topology

At the center of the star is a wiring hub or concentrator, and the nodes or

workstations are arranged around the central point representing the points of the star. Wiring costs tend to be higher for star networks than for other configurations, because each node requires its own individual cable.

A star topology is designed with each node (file server, workstations, and peripherals) connected directly to a central network hub or concentrator. Data on a star network passes through the hub or concentrator before continuing to its destination. The hub or concentrator manages and controls all functions of the network. It also acts as a repeater for the data flow. This configuration is common with twisted pair cable; however, it can also be used with coaxial cable or fiber optic cable. If hub goes down, entire network goes down. If a computer goes down, the network functions normally. The protocols used with star configurations are usually *Ethernet* or *AppelTalk*.

Example of star topology: In star topology each station or node attached to central node (may be hub or switch). Suppose node "C" wants to transfer data to node "A". If hub is use as a central node then it will broadcast packet to each every other node but only station/node "A" copies the packet and all other nodes discard the packet. But if switch is used instead of hub as a central node then it will directly sent the packet only to node "A".

Advantages

1. Easy to install, reconfigure and wire.
2. Robustness : If one link fails, only that link is affected.
3. Fast as compare to ring and bus topology.
4. Multiple devices can transfer data without collision.
5. Eliminates Traffic problem.
6. No disruptions to the network then connecting or removing devices.
7. Easy to detect faults and to remove parts.
8. Supported by several hardware and software venders.

Disadvantages

1. If central node (hub or switch) goes down then entire network goes down.
2. More cabling is required than bus topology, thus expensive than bus topology.
3. More expensive than bus topologies because of cost of the concentrators (hub or switch).

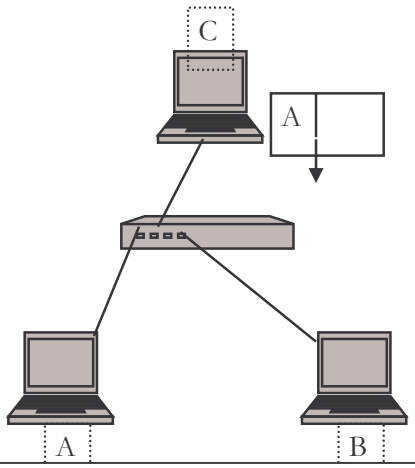


Figure (a) C transmits frame addressed to A

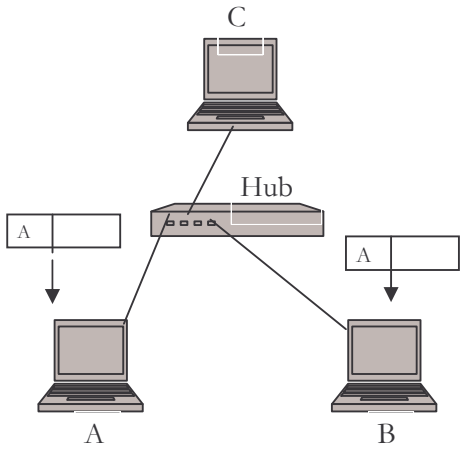


Figure (b) Hub used as a central controller

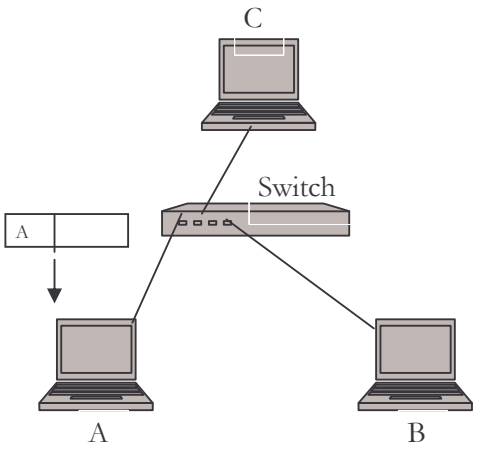
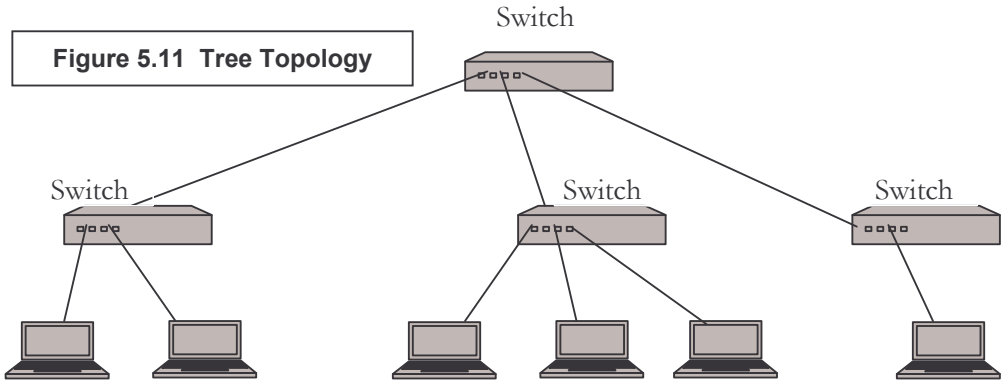


Figure (c) Switch used as a central controller
Figure 5.10 example of star topology

TREE TOPOLOGY (OR HIERARCHICAL TOPOLOGY)

A ***tree topology*** is variation of a star topology. In tree topology not every device plugs to the central hub. The majority of devices connect to a secondary hub that in turn is connected to the central hub.

Figure 5.11 Tree Topology



Advantages

1. Easy to install, reconfigure and wire.
2. Robustness : If one link fails, only that link is affected.
3. Fast as compare to ring topology.
4. Multiple devices can transfer data without collision.
5. Eliminates traffic problem of bus topology.
6. No disruptions to the network then connecting or removing devices.
7. Easy to detect faults and to remove parts.
8. Supported by several hardware and software venders.

Advantages in addition to star topology:

1. More devices can be attached due to secondary hub or switch.
2. Due to secondary devices distance signal can travel are increases.

Disadvantages

1. If central node (hub or switch) goes down then entire network goes down.

2. More cabling is required than bus topology, so expensive than bus topology.
3. More expensive than bus topologies because of the cost of the concentrators (hub or switch).

MESH TOPOLOGY

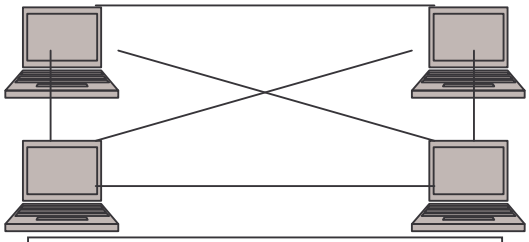


Figure 5.12 Mesh Topology

Each device has a dedicated point-to-point link to every other device. The mesh topology connects each computer on the network to the others. Fully connected mesh network has ***n(n-1)/2*** links for ***n*** devices. To accommodate n(n-1)/2 links, every device on the network must have ***n-1 input/output (I/O) ports.*** Meshes use a significantly larger amount of network cabling than do the other network topologies, which makes it more expensive.

The mesh topology is highly fault tolerant. That is, every computer has multiple possible connection paths to the other computers on the network, so a single cable break will not stop network communications between any two computers. Mesh is a network topology in which every device is connected by a cable to every other device on the network. Multiple links to each device are used to provide network link redundancy.

Advantages

1. Each connection can carry its own data load due to dedicated link.
2. Eliminates Traffic problem.
3. Mesh topology is robust. If one link becomes unusable, it doesn't affect other systems.
4. Privacy or security because of dedicated line.
5. Point-to-point link make fault identification easy.

Disadvantages

1. More cables are required than other topologies
2. **N-1** Input/Output ports are required for **N** devices.
3. Installation and reconfiguration is very difficult because each device must be connected to every other device.
4. Expensive due to hardware requirements such as cables and input/output ports.

HYBRID TOPOLOGY

The hybrid topology is a type of network topology that is composed of one or more interconnections of two or more networks that are based upon different physical topologies or a type of network topology.

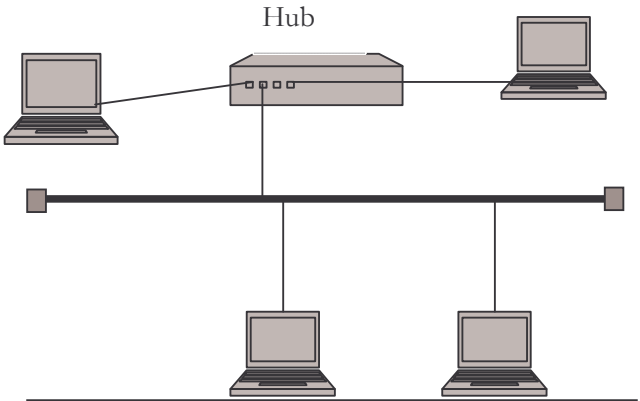


Figure 5.13 Hybrid Topology

5.4 CATEGORIES OF NETWORK

The size of the network, its ownership, distance it covers, and its physical architecture can determine types of network. There are three different categories of the network namely: Local Area Network (LAN), Metropolitan Area Network (MAN), Wide Area Network (WAN).

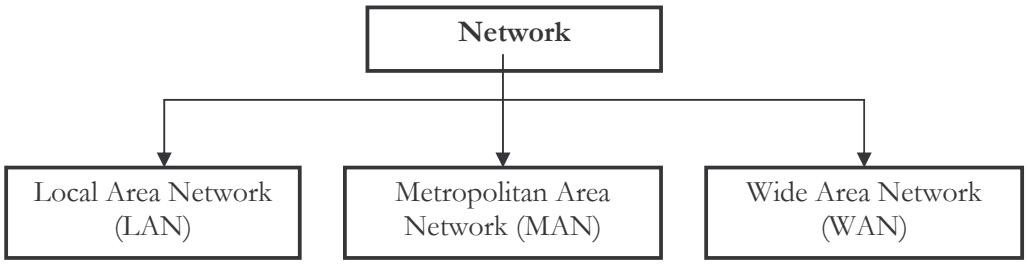


Figure 5.14 Categories of Network

LOCAL AREA NETWORK (LAN)

A LAN is a group of computers and associated peripheral devices connected by a communications channel, capable for sharing files and other resources among several users. **LAN** is usually privately owned and links the devices in a single office, building, or campus. LAN can be as simple as two PCs and printers, scanners, modems, etc. Local area networks or LANs enable computer-based equipment to communicate and share resources. The products that make up a LAN consist of computers, adapters, media and software.

LAN size is limited to few kilometers in distance. LAN have data rate 10 to 1000 Mbps. One of the main capabilities of a local area network is resource sharing, such as data and expensive peripherals. This ability to share resources can mean a decrease in the cost of an individual workstation, since not every workstation may need its own printer or hard disk.

LAN is a group of computers located in the same room, on the same floor, or in the same building that are connected to form a single network. Local area networks (LANs) allow users to share storage devices, printers, applications, data, and other network resources. They are limited to a specific geographical area, usually less than 2 kilometers in diameter. They might use a dedicated backbone to connect multiple sub-networks, but they do not use any telecommunication carrier circuits or leased lines except to connect with other LANs to form a wide area network (WAN).

A network is any collection of independent computers that communicate with one another over a shared network medium. LANs are networks usually confined to a geographic area, such as a single building or a college campus. LANs can be small, linking as few as three computers, but often link hundreds of computers used by thousands of people. The development of standard networking protocols and media has resulted in worldwide proliferation of LANs throughout business and educational organizations.

LANs are distinguished from other types of networks by their transmission media and topology. In general, a given LAN will use one type of transmission medium. The most common LAN topologies are bus, ring and star.

Characteristics of LAN

- Confined within geographical area.
- Relatively high data rate.

- Under single management
- *Topology :* bus, star, ring.
- *Medium :* Twisted pair, coaxial, fiber optic cable, wireless.

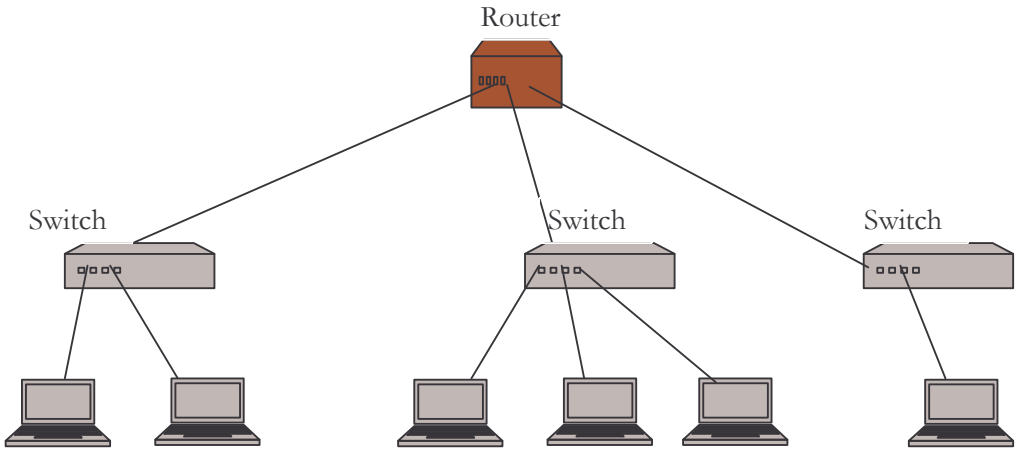


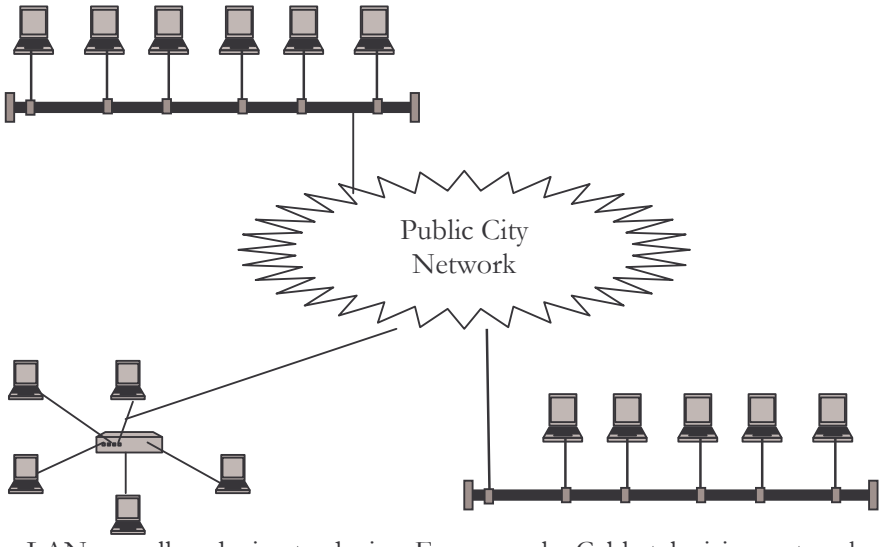
Figure 5.15 Local Area Network

METROPOLITAN AREA NETWORK (MAN)

A public, high-speed network, capable of voice and data transmission over a distance of up to 80 kilometers (50 miles). A MAN is smaller than a wide-area network (WAN) but larger than a local-area network (LAN). **MAN** is designed to extend over an entire city. MAN is a multiple local area networks (LANs) that are connected on a campus or industrial complex using a high-speed backbone. Multiple networks that are connected within the same city to form a citywide network for a specific government or industry is called as MAN. Fiber Distributed Data Interface (FDDI) is a good network technology

Figure 5.16 Metropolitan Area Network (MAN)

for building a metropolitan area network (MAN). A MAN may be wholly owned and operated by a private company. Number of LANs connected so that resources may be



shared LAN-to-LAN as well as device-to-device. For example, Cable television network.

WIDE AREA NETWORK (WAN)

A Wide Area Networks **[WAN]** provides long-distance transmission of data, voice, image, and video information over large geographical areas that may comprise a country, or even whole world. A geographically distributed network composed of local area networks (LANs) or metropolitan are networks (MAN) joined into a single large network using services provided by common carriers is called as Wide Area Networks.. This is accomplished by connecting the different LANs using services such as dedicated leased phone lines, dial-up phone lines (both synchronous and asynchronous), satellite links, and data packet carrier services. Wide area networking can be as simple as a modem and remote access server for employees to dial into, or it can be as complex as hundreds of branch offices globally linked using special routing protocols and filters to minimize the expense of sending data sent over vast distances

WANs utilize public, leased, or private communication devices. A WAN that is wholly owned and used by a single company is often referred to as an enterprise network. WANS are commonly implemented in enterprise networking environments in which company offices are in different cities, states, or countries or on different continents

WAN's span more than one geographical area and are used to connect remote offices to each other. Basically, a WAN is comprised of two or more LAN's joined together by routers. Routers are hardware devices that direct traffic from one LAN to another. A WAN would be found in medium to large sized businesses with more than one office location. For example, a software company might have its headquarters in Delhi, but also have remote office locations in Austin, New York, London and California. The LAN in the Delhi office would be connected to the LAN in the remote offices forming a WAN.

The early WAN technologies were limited to expensive leased lines such as T1 lines, slow packet-switching services such as X.25, cheap but low-bandwidth solutions such as modems, and dial-up Integrated Services Digital Network (ISDN) connections, but this has changed considerably in recent years. Frame relay services provide high-speed packet-switching services that offer more bandwidth than X.25, and virtual private networks (VPNs) created using Internet Protocol (IP) tunneling technologies enable companies to securely connect branch offices by using the Internet as a backbone service. Intranets and extranets provide remote and mobile users with access to company resources and applications and provide connectivity with business partners and resellers. Wireless networking technologies allow roaming users to access network resources by using cell-based technologies. Digital Subscriber Line (DSL) services provide T1 speeds at much lower costs than dedicated T1 circuits. These and other new technologies continue to evolve and proliferate, allowing enterprise network administrators to implement and administer a highly diverse range of WAN solutions.

LAN's and WAN's come in many different flavors. The most popular type of network is Ethernet. Ethernet networks have speeds of 10 Mbps, 100 Mbps, or 1 Gbps. A 10 Mbps Ethernet network transmits data at 10 Mega bits per second. A 100 Mbps Ethernet network transmits data at 100 mega bits per second. A 1 Gbps Ethernet network transmits data at 1000 mega bits per second. The majority of networks today operate at 100Mbps. 1Gbps and 10Gbps networks, however, are becoming more common as technology and bandwidth demand increases.

SUMMARY

Data communication is the transfer of information from one computer to another over a communications link. The transfer can be occasional, continuous, or a combination of both. A network is a set of devices (often referred as nodes) connected by communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network. A group of computers connected in some fashion in order to share resources is called as computer network.

Computer networks can be used for resource sharing, information sharing, person-to-person communication, electronic business and interactive entertainment.

PRACTICE SET

The topology of a network is the geometric representation of the relationship of all the links & linking devices (usually called nodes) to one another. Basic network topologies are bus, ring, star and Mesh.

Networks can be categorized depends on its size, its ownership, distance it covers, and its physical architecture. LAN, MAN, and WAN are the different categories of network.

Review Questions

1. Explain Simplex, Half duplex and full duplex transmission.
2. Write a SN on types of network topologies.
3. Write a SN on types of network.
4. Explain advantages of network.
5. Write the merits and demerits of following topology:
   (i) Bus topology
   (ii) Ring topology
   (iii) Star topology
   (iv) Mesh topology
6. Explain the use of computer network.

Multiple Choice Questions

1. Information to be communicated in a data communications system is the …….
A) Medium B) Message C) Protocol D) Transmission

2. An unauthorized user is a network ……. issue.
A) Performance B) Reliability C) Security D) All of the above

3. Which topology requires a central controller or hub?
A) Mesh B) Bus C) Star D) Ring

4. Which topology requires a multipoint connection?
A) Mesh B) Bus C) Star D) Ring

5. Communication between a computer and a keyboard involves ……. transmission.
A) Simplex B) Half-Duplex C) Full-Duplex D) automatic

6. A television broadcast is an example of ……. transmission.
A) Simplex B) Half-Duplex C) Full-Duplex D) automatic

7. A ……. connection provides a dedicated link between two devices.
A) Point-to-point B) Multipoint C) Primary D) Secondary

8. In a ……. connection, more than two devices can share a single link.
A) Point-to-point B) Multipoint C) Primary D) Secondary

9. This was the first network…….
A) CSNET B) NSFNET C) ANSNET D) ARPANET

10. In a ……. connection, three or more devices share a link.
A) Point-to-point B) Multipoint C) option A) or B) D) option A) and B)

11. Devices may be arranged in a ……. topology.
A) Star B) Ring C) Bus D) all of the above

12. A ……. is a data communication system within a building, plant, or campus, or between nearby buildings.
A) LAN B) MAN C) WAN D) none of the above

13. A ……. is a data communication system spanning states, countries, or the whole world.
A) LAN B) MAN C) WAN D) none of the above

14. A ……. is a set of rules that governs data communication.
A) forum B) protocol C) standard D) none of the above

* * *

Chapter 6 Reference Models

The term *networking model*, or *networking architecture or reference model,* refers to an organized description of all the functions needed for useful communication to occur. Individual protocols and hardware specifications are used to implement the functions described in the networking model. When multiple computers and other networking devices implement these protocols, which, in turn, implement the functions described by the networking model, the computers can successfully communicate.

OBJECTIVES OF THIS CHAPTER

After completing this chapter, the student should be able to:

- define network architecture and reference model
- define different layers of ISO/OSI and TCP/IP reference model
- define benefits of standardization or layered model
- differentiate between ISO/OSI and TCP/IP models
- define function of each layer of ISO/OSI model
- Learn various examples of each layer of ISO/OSI model

6.1 INTRODUCTION

Welcome to the exciting world of internetworking. This chapter will really help you understand the basics of internetworking. First, you need to know exactly what an internetwork is, right? You create an internetwork when you take two or more LANs or WANs and connect them using internetworking device such as router or gateway. We will study these devices in detail in the next chapter.

We are going to dissect the Open Systems Interconnection (OSI) and Transmission Control Protocol / Internet Protocol (TCP/IP) models and describe each part in detail, because you really need a good grasp of it for the solid foundation will build your networking knowledge.

The OSI model has seven hierarchical layers that were developed to enable different networks to communicate reliably between disparate systems.

It is practically impossible to find a computer today that does not support the set of networking protocols called TCP/IP. Every Microsoft, Linux, and UNIX operating system includes support for TCP/IP. Hand-held digital assistants and cell phones also support TCP/IP. Even IBM Mainframe operating systems support TCP/IP. The world has not always been so simple. Once upon a time, there were no networking protocols, including TCP/IP. Vendors created the first networking protocols; these protocols supported only that vendor's computers, and the details were not even published to the public.

Computers could typically communicate only with computers from the same manufacturer. For example, companies ran either a complete DECnet solution or an IBM solution—not both together. In the late 1970s, the *Open Systems Interconnection (OSI) reference model* was developed by the International Organization for Standardization (ISO) to break this barrier.

The OSI model was meant to help vendors create interoperable network devices and software in the form of protocols so that different vendor networks could work with each other. Like world peace, it'll probably never happen completely, but it's still a great goal. The OSI model is the primary architectural model for networks. It describes how data and network information are communicated from an application on one computer, through the network media, to an application on another computer.

The ISO had a noble goal for the OSI: to standardize data networking protocols to allow communication between all computers across the entire planet. The OSI worked toward this ambitious and noble goal, with participants from most of the technologically developed nations on Earth, many companies participated in the process. The OSI reference model breaks this approach into layers.

The Layered Approach

A *reference model* is a conceptual blueprint of how communications should take place. It addresses all the processes required for effective communication and divides these processes into logical groupings called *layers*. When a communication system is designed in this manner, it's known as *layered architecture*.

Think of it like this: You and some friends want to start a company. One of the first things you'll do is sit down and think through what tasks must be done, who will do them, what order they will be done in, and how they relate to each other. Ultimately, you might group these tasks into departments. Let's say you decide to have an order-taking department, an inventory department, and a shipping department. Each of your departments has its own unique tasks, keeping its staff members busy and requiring them to focus on only their own duties. In this scenario, I'm using departments as a metaphor for the layers in a communication system.

Similarly, software developers can use a reference model to understand computer communication processes and see what types of functions need to be accomplished on any one layer. If they are developing a protocol for a certain layer, all they need to concern themselves with is the specific layer's functions, not those of any other layer. Another layer and protocol will handle the other functions. The technical term for this idea is *binding*. The communication processes that are related to each other are bound, or grouped together, at a particular layer.

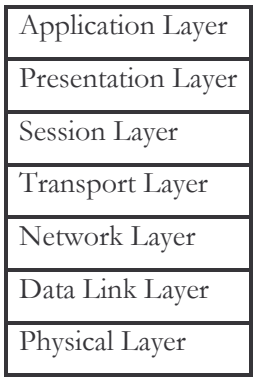


Figure 6.1 ISO/OSI Layers

Advantages of Reference Models

The Reference model is hierarchical, and the same benefits and advantages can apply to any layered model. The primary purpose of all such models, especially the OSI model, is to allow different vendors' networks to interoperate. Advantages of using the layered model include, but are not limited to, the following:

Easier to learn—Humans can more easily discuss and learn about the many details of a protocol specification.

Easier to develop—Reduced complexity allows easier program changes and faster product evolution. Prevents changes in one layer from affecting other layers, so it does not hamper development

Multi-vendor interoperability—creating products to meet the same networking standards means that computers and networking gear from multiple vendors can work in the same network. This allows multiple-vendor development through standardization of network components.

Modular engineering—One vendor can write software that implements higher layers—for example, a web browser—and another can write software that implements the lower layers—for example, Microsoft's built-in TCP/IP software in its operating systems.

The benefits of layering can be seen in the familiar postal service analogy. A person writing a letter does not have to think about how the postal service will deliver a letter across the country. The postal worker in the middle of the country does not have to worry about the contents of the letter. Likewise, layering enables one software package or hardware device to implement functions from one layer, assuming that other software/hardware will perform the functions defined by the other layers. For instance, a web browser does not need to think about what the network topology looks like, the Ethernet card in the PC does not need to think about the contents of the web page, and a internetworking devices such as router in the middle of the network does not need to worry about the contents of the web page or whether the computer that sent the packet was using an Ethernet card or some other networking card. – This example enables us to illustrate and understand how protocols works.

6.2 ISO/OSI MODEL

OSI is the Open System Interconnection reference model for communications developed by International Standardization Organization (ISO). One of the greatest functions of the OSI specifications is to assist in data transfer between disparate hosts—meaning, for example, that they enable us to transfer data between a Unix host and a PC or a Mac.

The OSI reference model consists of seven layers. Each layer defines a set of typical networking functions. When OSI was in active development in the 1980s and 1990s, the OSI committees created new protocols and specifications to implement the functions specified by each layer. In other cases, the OSI committees did not create new protocols or standards, but instead referenced other protocols that were already defined. For instance, the IEEE defines Ethernet standards, so the OSI committees did not waste time specifying a new type of Ethernet; it simply referred to the IEEE Ethernet standards.

The OSI isn't a physical model, rather, it's a set of guidelines that application developers can use to create and implement applications that run on a network. It also provides a framework for creating and implementing networking standards, devices, and internetworking schemes.

Network devices that operate at all seven layers of the OSI model include:

- Network management stations (NMS)
- Web and Application servers
- Network hosts

The upper layers of the OSI reference model (application, presentation, and session—Layers 7, 6, and 5) define functions focused on the application. The lower four layers

(transport, network, data link, and physical—Layers 4, 3, 2, and 1) define functions focused on end-to-end delivery of the data.

The Application Layer

The *Application layer* of the OSI model marks the spot where users actually communicate to the computer. This layer actually only comes into play when it's apparent that access to the network is going to be needed soon. Take the case of Internet Explorer. You could uninstall every trace of networking components from a system, such as TCP/IP, NIC card, etc., and you could still use Internet Explorer (IE) to view a local HTML document—no problem. But things would definitely get messy if you tried to do something like actually view an HTML document that must be retrieved using HTTP, or nab a file with FTP. That's because IE will respond to requests like those by attempting to access the Application layer. And what's actually happening is that the Application layer is acting as an interface between the actual application program—which isn't at all a part of the layered structure— and the next layer down, by providing ways for the application to send information down through the protocol stack. In other words, IE doesn't truly reside within the Application layer—it interfaces with Application-layer protocols when it needs to deal with remote resources.

The Application layer is also responsible for identifying and establishing the availability of the intended communication partner, and determining whether sufficient resources for the intended communication exist.

These tasks are important because computer applications sometimes require more than only desktop resources. Often, they'll unite communicating components from more than one network application. Prime examples are file transfers and e-mail, as well as enabling remote access, network management activities, client/server processes, and information location.

Application layer or Layer 7 defines the interface between the communications software and any applications that need to communicate outside the computer on which the application resides. For example, a web browser is an application on a computer. The browser needs to get the contents of a web page; OSI Layer 7 defines the protocols used on behalf of the application to get the web page.

Examples of Internetworking application are as follows:

World Wide Web (WWW)

Connects countless servers (the number seems to grow with each passing day) presenting diverse formats. Most are multimedia and can include graphics, text, video, and sound. Netscape Navigator and IE simplify both accessing and viewing websites.

E-mail gateways

Can use Simple Mail Transfer Protocol (SMTP) to deliver messages between different e-mail applications.

The Presentation Layer

The *Presentation layer* gets its name from its purpose: It presents data to the Application layer and is responsible for data translation and code formatting.

The presentation layer is concerned with the syntax and semantics of the information exchanged between two system. The presentation layer is concerned with the representation of user or system data. This includes necessary conversions (for example, printer control characters) and code translation (for example, ASCII to or from EBCDIC). By providing

translation services, the Presentation layer ensures that data transferred from the Application layer of one system can be read by the Application layer of another one.

Responsibilities of the Presentation layer includes:

- **Translations:** Different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependant format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependant format.
- **Encryption:** The process of rendering a message (or data) unusable to all but the intended recipients, who have the ability to decrypt it.
- **Compression:** reduces the number of bits to be transmitted. It saves network bandwidth.

Presentation layer's main purpose is to define data formats, such as ASCII text, EBCDIC text, binary, BCD, and JPEG. Encryption also is defined by OSI as a presentation layer service. For example, FTP enables you to choose binary or ASCII transfer. If binary is selected, the sender and receiver do not modify the contents of the file. If ASCII is chosen, the sender translates the text from the sender's character set to a standard ASCII and sends the data. The receiver translates back from the standard ASCII to the character set used on the receiving computer.

Some Presentation layer standards are involved in multimedia operations too. The following serve to direct graphic and visual image presentation:

- **JPEG** Photo standards brought to us by The Joint Photographic Experts Group.
- **TIFF** Tagged Image File Format; a standard graphics format for high-resolution, bitmapped images.
- **PICT** A picture format used by Macintosh programs for transferring QuickDraw graphics.

Other standards defines movies and sound presentation:

- **QuickTime** For use with Macintosh programs; manages audio and video applications.
- **MIDI** Musical Instrument Digital Interface (sometimes called Musical Instrument Device Interface), used for digitized music.
- **MPEG** Increasingly popular Moving Picture Experts Group standard for the compression and coding of motion video for CDs. It provides digital storage and bit rates up to 1.5Mbps.

The Session Layer

The session layer defines how to start, control, and end conversations (called sessions). This includes the control and management of multiple bidirectional messages so that the application can be notified if only some of a series of messages are completed. This allows the presentation layer to have a seamless view of an incoming stream of data. The presentation layer can be presented with data if all flows occur in some cases. For example, an automated teller machine transaction in which you withdraw cash from your checking account should not debit your account and then fail before handing you the cash, recording the transaction even though you did not receive money. The session layer creates ways to imply which flows are parts of the same session and which flows must complete before any are considered complete.

The Session layer is responsible for setting up, managing, and then tearing down sessions between Presentation layer entities. This layer also provides dialogue control between devices, or nodes. It coordinates communication between systems, and serves to organize their communication by offering three different modes: *simplex*, *half duplex*, and *full duplex*. To sum up, the Session layer basically keeps different applications' data separate from other applications' data.

The following are some examples of Session layer protocols and interfaces:

- **Structured Query Language (SQL)** Developed by IBM to provide users with a simpler way to define their information requirements on both local and remote systems.
- **Digital Network Architecture Session Control Protocol (DNA SCP)** A DECnet Session layer protocol.
- **AppleTalk Session Protocol (ASP)** Another client/server mechanism, which both establishes and maintains sessions between AppleTalk client and server machines.
- **Remote Procedure Call (RPC)** A broad client/server redirection tool used for disparate service environments. Its procedures are created on clients and performed on servers.
- **Network File System (NFS)** Developed by Sun Microsystems and used with TCP/IP and Unix workstations to allow transparent access to remote resources.

The Transport Layer

The *Transport layer* segments and reassembles data into a data stream. Services located in the Transport layer both segments and reassembles data from upper-layer applications and unite it onto the same data stream. Thus it provide an end-to-end data transport service and can establish a logical connection between the sending host and destination host on an internetwork.

Responsibilities of the transport layer are as follows

- **Service-point addressing :** Computers often run multiple programs at the same time. Source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on the other. The transport layer header therefore must include a type of address called a service-point address (or **port address**).

- **Segmentation and reassembly :** A message is divided into transmittable *segments*, each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in the transmission.
- **Connection control :** It creates a connection between the two end ports. A connection is a single logical path between the source and destination that is associated with all packets in a message.
- **Flow control :** Flow control at this level is performed end to end rather than across a single link.
- **Error control :** Error control at this level is performed end to end rather than across a single link. The transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication).

The network layer (discuss next) delivers each packet to the correct destination computer; the transport layer delivers the entire message to the correct process on that computer.

The Network Layer

This layer defines end-to-end delivery of *packets*. To accomplish this, the network layer defines logical addressing so that any endpoint can be identified. It also defines how routing works and how routes are learned so that the packets can be delivered.

The *Network layer* (also called layer 3) manages device addressing, tracks the location of devices on the network, and determines the best way to move data, which means that the Network layer must transport traffic between devices that aren't locally attached. Routers (layer-3 devices) are specified at the Network layer and provide the routing services within an internetwork.

It happens like this: First, when a packet is received on a router interface, the destination IP address is checked. If the packet isn't destined for that particular router, it will look up the destination network address in the routing table. Once the router chooses an exit interface, the packet will be sent to that interface to be framed and sent out on the local network. If the router can't find an entry for the packet's destination network in the routing table, the router drops the packet.

Responsibilities of Network layer :

- **Logical addressing :** The physical addressing implemented by the data link layer (discuss next) handles the addressing problem locally. If a packet passes the network boundary, then need of another addressing system to help to distinguish the source and destination systems.
- **Routing :** When independent networks or links are connected together to create an internetwork (a network of networks), the connecting devices (called router or gateway) route the packets to their final destination.

The Data Link Layer

The data link layer (Layer 2) specifications deliver data across one particular link or medium. These protocols are necessarily concerned with the type of media in question; for example, 802.3 and 802.2 define Ethernet for the IEEE, which are referenced by OSI as valid data link layer (Layer 2) protocols. Other protocols, such as High-Level Data

Link Control (HDLC) for a point-to-point WAN link, deal with the different details of a WAN link.

The Data Link layer provides the physical transmission of the data and handles error notification, network topology, and flow control. This means the Data Link layer will ensure that messages are delivered to the proper device on a LAN using hardware addresses, and translates messages from the Network layer into bits for the Physical layer to transmit. The Data Link layer formats the message into pieces; each called a *data frame*, and adds a customized header containing the hardware destination and source address.

For a host to send packets to individual hosts on a local network as well as transmitting packets between intermediate hops such as routers, the Data Link layer uses hardware addressing. Each time a packet is sent between routers, it's framed with control information at the Data Link layer, but that information is stripped off at the receiving router and only the original packet is left completely intact. This framing of the packet continues for each hop until the packet is finally delivered to the correct receiving host. It's really important to understand that the packet itself is never altered along the route; it's only encapsulated with the type of control information required for it to be properly passed on to the different media types.

Responsibilities of the data link layer are as follows :

- **Framing :** The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
- **Physical addressing :** If frames are distributed to different system on the network, the data link layer adds header to the frame to define the **physical address** of the sender (source address) and receiver address (destination address) of the frame. If the frame is intended for the system outside the sender's network, the receiver address is the address of device that connects one network to the next.
- **Flow control :** If the rate at which the data are absorbed by receiver is less than the rate produced in the sender, the data linker layer imposes a flow control mechanism.
- **Error control :** It adds reliability to the physical layer by adding mechanism to detect and retransmit damaged or lost frames. It also prevents duplication of frames.
- **Access control :** When two or more devices are connected to the same link, data link layer protocols determine which device has control over the link at any given time.

The Physical Layer

The *Physical layer* does two things: It sends bits and receives bits. Bits come only in binary format 1 or 0. Physical layer (Layer 1) specifications deal with the physical characteristics of the transmission medium. Connectors, pins, use of pins, electrical currents, encoding, and light modulation are all part of different physical layer specifications. Multiple specifications sometimes are used to complete all details of the physical layer. For example, RJ-45 defines the shape of the connector and the number of wires or pins in the cable. Ethernet and 802.3 define the use of wires or pins 1, 2, 3, and 6. So, to use a Category 5 cable with an RJ-45 connector for an Ethernet connection, Ethernet and RJ-45 physical layer specifications are used.

The physical layer concerned with the following :

- **Physical characteristics of interfaces and media :** Defines type of transmission medium. For example guided (in turn twisted pair cable, fiber optic cable, etc) or unguided (in turn defines infrared, microwave or satellite).
- **Representation of bits :** Data consist of a stream of bits (0's and 1's). To be transmitted, bits must be encoded into signals-electrical or optical. The physical layer defines the type of encoding.
- **Data rate :** The transmission rate- the number of bits sent per second. For example 10 Mbps or 100 Mbps
- **Synchronization of bits :** The sender and receiver clocks must be synchronized.
- **Physical topology :** It defines how devices are connected to make a network. For ex: a *star topology* (devices are connected through a central device), a *ring topology* (every device is connected to the next).
- **Transmission mode :** It defines direction of transmission between two devices : simplex, half-duplex, or full-duplex.

OSI Model Terminology

Remembering the names of the OSI layers is just an exercise in memorization. You might benefit from the following list of mnemonic phrases, with the first letters in each word being the same as the first letters of the OSI layer names, in order:

- All People Seem To Need Data Processing (Layers 7 to 1)
- Please Do Not Take Sausage Pizzas Away (Layers 1 to 7)

Following table summarizes the OSI Layers and there examples.

Application Layer	Telnet, HTTP, FTP, WWW browsers, SMTP gateways, SNMP
Presentation Layer	JPEG, ASCII, EBCDIC, TIFF, GIF, PICT, MPEG, MIDI, encryption, MP3
Session Layer	RPC, SQL, NFS, NetBIOS names, AppleTalk ASP, DECnet-SCP
Transport Layer	TCP, UDP, SPX
Network Layer	IP, IPX, AppleTalk, DDP
Data Link Layer	IEEE 802.3/802.2, HDLC, Frame Relay, PPP, FDDI, ATM
Physical Layer	EIA/TIA-232, V.35, RJ-45, Ethernet

Table 6.1 Examples of OSI Layers

Following table shows devices that function at various OSI Reference Model layers.

Layer	Name of Layer	Device
3	Network	Router
2	Data Link	Switch, Bridge, Network Interface Card (NIC) or LAN card
1	Physical	Hub, Repeater

Table 6.2 List of devices works at OSI Layers

6.3 THE TCP/IP MODEL

TCP/IP an abbreviation for Transmission Control Protocol/Internet Protocol, an industry-standard protocol suite for wide area networks (WANs) developed in the 1970s and 1980s by the U.S. Department of Defense (DoD). TCP/IP is not one protocol, but is a suite of many protocols. The protocols define applications, transport controls, networking, routing, and network management.

Internet communication has become a fundamental part of life. **U.S.** government agencies realized the importance and potential of internet technology many years ago, and have funded research that has made possible a global Internet The ARPA (***Advanced Research Projects Agency***) technology includes a set of network standards that specify the details of how computers communicate, as well as a set of conventions for interconnecting networks and routing traffic. Officially named the **TCP/IP** Internet Protocol Suite and commonly referred to as ***TCP/IP.***

Like other networking architectures, TCP/IP classifies the various protocols into different categories. Following table list the layers of TCP/IP model and there examples.

Layer Name	**Example**
Application Layer	HTTP, POP3, IMAP, SMTP
Transport Layer or Host-to-Host Layer	TCP, UDP
Internetwork Layer	IP
Network Interface	Ethernet, Frame Relay

Table 6.3 Examples of TCP/IP Layers

Application Layer
Transport Layer or Host-to-Host Layer
Internetwork Layer
Network Interface

Figure 6.2 Layers of TCP/IP

Each layer in the stack adds control information to ensure proper delivery. This control information is called a *header* because it is placed in front of the data to be transmitted. Each layer treats all of the information it receives from the layer above as data and places its own header in front of that information. The addition of delivery information at every layer is called *encapsulation.*

Following figure shows OSI, as compared with TCP/IP

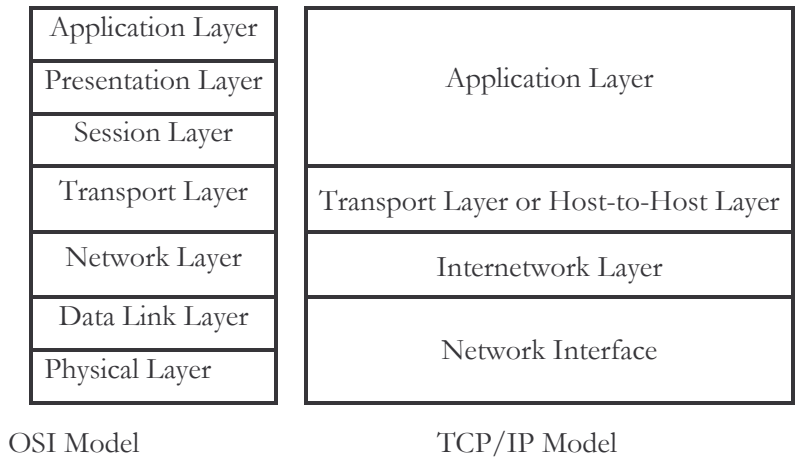


Figure 6.3 OSI and TCP/IP model comparison

Application Layer Protocols

Responsible for application-level access to TCP/IP networking services.

SMTP : Simple Mail Transfer Protocol

Used to transfer e-mail from one computer to another computer over a TCP/IP network such as the Internet.

FTP : File Transfer Protocol

Used to transfer a complete file from one computer to another.

FTP uses Transmission Control Protocol (TCP) for reliable network communication by establishing a session before initiating data transfer.

TELNET : Terminal NETwork

Telnet is a terminal emulation program, which is a command-line interface for issuing commands on a remote computer.

TELNET enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.

DNS : Domain Name Server

Used to translate computer name into equivalent IP Address.

SNMP : Simple Network Management Protocol

Used for collecting *statistical* and *configuration* information about network devices such as computers, hubs, switches, routers, and even network printers. The statistical information includes the number of packets or frames sent or received per second, the number of errors per second, and so on. The configuration information includes the IP address of an interface device, the version of the operating system running on the device, and so on.

TFTP : Trivial File Transfer Protocol

TFTP copies files to and from remote hosts by using the User Datagram Protocol (UDP). TFTP differs from the popular File Transfer Protocol (FTP) in only one aspect that it do not support any form of authentication.

HTTP : Hyper Text Transfer Protocol

Used to transfer a World Wide Web page from one computer to another.

Transport Layer Protocols

Establish communication through connection-oriented sessions and connectionless broadcasts. Protocols at this layer include Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

TCP : Transmission Control Protocol

It is a reliable connection–oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet.

UDP : User Datagram Protocol

It is an unreliable, connectionless protocol.

Internet Layer Protocols

Job of Internet layer is to permit hosts to inject packets into any network and have them travel independently to the destination.

They may arrive in a different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired.

Internet layer protocols are responsible for routing and encapsulation into IP packets. Protocols at this layer include :

IP : Internet Protocol

A TCP/IP network/internet layer protocol for addressing and routing packets of data between hosts on a TCP/IP network.

Internet Protocol (IP) is a connectionless protocol. IP does not guarantee delivery of data. The responsibility for guaranteeing delivery and sending acknowledgments lies with the higher transport-level protocol Transmission Control Protocol (TCP).

ICMP : Internet Control Message Protocol

A TCP/IP network/internet layer protocol used by routers and TCP/IP hosts for building and maintaining routing tables, adjusting data flow rates, and reporting errors and control messages for TCP/IP network communication.

ICMP used by programs like ping.

ARP : Address Resolution Protocol

Its responsible for resolving IP addresses into MAC addresses.

RARP : Reverse Address Resolution Protocol

Used during bootstrap to obtain an IP address.

SUMMARY

You learned about how ISO/OSI protocols at the various layers work with each other. Those same concepts are true of TCP/IP, as well as other networking models.

The OSI model—the seven-layer model used to help application developers design applications that can run on any type of system or network. Each layer has its special jobs and select responsibilities within the model to ensure that solid, effective communications do, in fact, occur.

The basic ideas can be summed up as follows:

- Each layer provides a service to the layer above it in the protocol specification.
- Each layer communicates with the same layer's software or hardware on other computers.
- To accomplish these tasks, the data is encapsulated progressively with new headers when sending the data and is de-encapsulated when receiving the data.

PRACTICE SET

Review Questions

1. Which layer chooses and determines the availability of communicating partners, along with the resources necessary to make the connection; coordinates partnering applications; and forms a consensus on procedures for controlling data integrity and error recovery?

2. Which layer is responsible for converting data packets from the Data Link layer into electrical signals?

3. At which layer is routing implemented, enabling connections and path selection between two end systems?

4. Which layer defines how data is formatted, presented, encoded, and converted for use on the network?

5. Which layer is responsible for creating, managing, and terminating sessions between applications?

6. Which layer ensures the trustworthy transmission of data across a physical link and is primarily concerned with physical addressing, line discipline, network topology, error notification, ordered delivery of frames, and flow control?

7. Which layer is represented by frames?

8. Which layer is represented by segments?

9. Which layer is represented by packets?

10. Which layer is represented by bits?

11. Name the seven layers of the OSI model.

12. What is the main purpose(s) of Layer 7?

13. What is the main purpose(s) of Layer 6?

14. What is the main purpose(s) of Layer 5?

15. What is the main purpose(s) of Layer 4?

16. What is the main purpose(s) of Layer 3?

17. What is the main purpose(s) of Layer 2?

18. What is the main purpose(s) of Layer 1?

19. List the terms behind the acronym TCP/IP.

20. List the terms behind the acronym OSI.

Multiple Choice Questions

1. PDUs at the Network layer of the OSI are called what?
A) Transport B) Frames C) Packets D) Segments

2. PDUs at the Data Link layer are named what?
A) Transport B) Frames C) Packets D) Segments

3. Segmentation of a data stream happens at which layer of the OSI model?
A) Physical B) Data Link C) Network D) Transport

4. Which layer of the OSI provides translation of data?
A) Application B) Presentation C) Session D) Transport E) Data Link

5. When data is encapsulated, which is the correct order?
A) Data, frame, packet, segment, bit B) Segment, data, packet, frame, bit

C) Data, segment, packet, frame, bit D) Data, segment, frame, packet, bit

6. Which of the following is not an advantage of a layered model?
A) Allows multiple-vendor development through standardization of network components

B) Allows various types of network hardware and software to communicate

C) Allows changes to occur in all layers without having to change just one layer

D) Prevents changes in one layer from affecting other layers, so it does not hamper Development

7. Which of the following is Presentation layer protocol?
A) TFTP B) IP C) TCPD) PICT

8. Which of the following protocols are examples of TCP/IP transport layer protocols?
A) Ethernet B) HTTP C) IP D) UDP

9. Which of the following protocols are examples of TCP/IP network interface layer protocols?
A) Ethernet B) HTTP C) IP D) TCP

10. Which OSI layer defines the functions of logical network-wide addressing and routing?
A) Layer 1 B) Layer 2 C) Layer 3 D) Layer 4

11. Which OSI layer defines the standards for cabling and connectors?
A) Layer 1 B) Layer 2 C) Layer 3 D) Layer 4

12. Which OSI layer defines the standards for data formats and encryption?
A) Layer 7 B) Layer 6 C) Layer 5 D) Layer 4

13. Which of the following terms are not valid terms for the names of the seven OSI layers?
A) Application B) Data link C) Transmission D) Internetwork

14. Which of the following terms is used specifically to identify the entity that is created when encapsulating data inside data-link headers and trailers?
A) chunk B) data
C) frame D) None—there is no encapsulation by the data link layer

* * *

Chapter 7 Networking and Internetworking Devices

Networks and networking have grown exponentially over the last 15 years. They have evolved at the speed of light -- just to keep up with huge increases in basic mission critical user needs -- such as sharing data and printers, as well as more advanced demands such as video conferencing. Unless everyone who needs to share network resources is located in the same office area (an increasingly uncommon situation), the challenge is to connect the sometimes many relevant networks together so all users can share the networks' wealth.

OBJECTIVES OF THIS CHAPTER

After completing this chapter, the student should be able to:

- define Internetworking
- define networking and Internetworking devices
- learn the difference between hub and switch
- Understand the working of switch
- learn the difference between switch and bridge
- Understand working of the router

7.1 INTRODUCTION

Industry trends show a dramatic increase in the use of intelligent workstations such as notebooks, portable computers, and desktop computers. Important information is being moved from mainframes to these machines to provide efficient and cost-effective client/server functions. This changing information infrastructure requires faster and more reliable access to data, which in turn drives new applications based on multimedia, incorporating voice, data, and video.

Internetworking involves connecting two or more distinct computer networks or network segments together to form an **internetwork** (often shortened to *internet*), using devices which operate at layer 3 (Network layer) of the OSI Basic Reference Model (such as routers or layer 3 switches) to connect them together to allow traffic to flow back and forth between them.

Internet

The global TCP/IP public internetwork that originated in the ARPANET project of the U.S. Department of Defense in the 1970s. The original purpose of ARPANET was to create a wide area network (WAN) that would allow researchers at various defense and civilian research agencies to communicate with each other and to collaborate on projects. When ARPANET grew larger and an increasing number of civilian agencies such as universities and networking companies wanted access to it. Administration of the network (now called the Internet) was given to the National Science Foundation (NSF) and then to Internet Network Information Center (InterNIC). The Internet is not owned by any one group; it is a collection of networks and gateways that run a common TCP/IP protocol and that all evolved from ARPANET.

Devices which are used for connecting two or more than two networks are classified into two categories namely: networking device and internetworking devices. Networking devices generally used in organization to connect internal networks of the organization, whereas internetworking devices are used to connect organizations network to external world or network such as ISP provider's network or in simple terms to Internet.

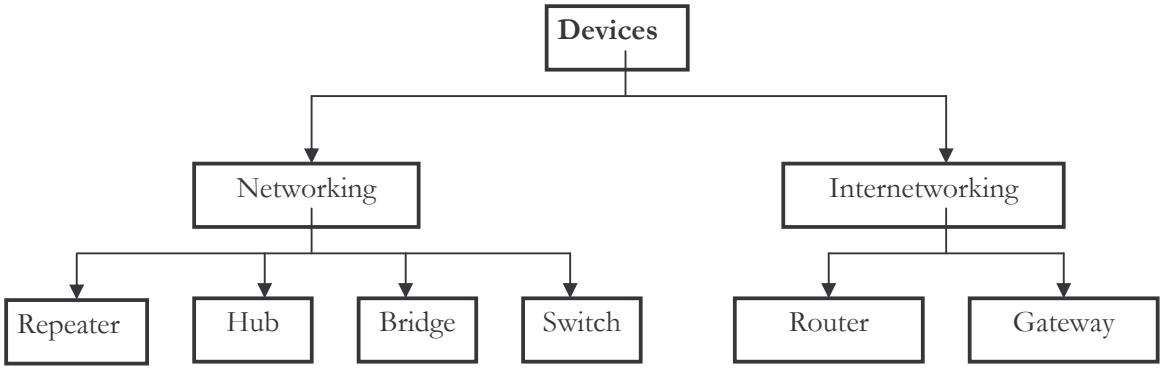


Figure 7.1 Networking and Internetworking Devices

7.2 REPEATER

A ***repeater*** (or regenerator) is an electronic device that operates on physical layer of OSI model.

Since a signal loses strength as it passes along a cable, it is often necessary to boost the signal with a device called a repeater. The repeater electrically amplifies the signal it receives and rebroadcasts it. Repeaters can be separate devices or they can be incorporated into a concentrator. They are used when the total length of your network cable exceeds the standards set for the type of cable being used. A repeater installed on a link receives the signal before it becomes too weak or corrupted, regenerates the original bit pattern, and puts the refreshed copy back onto the link.

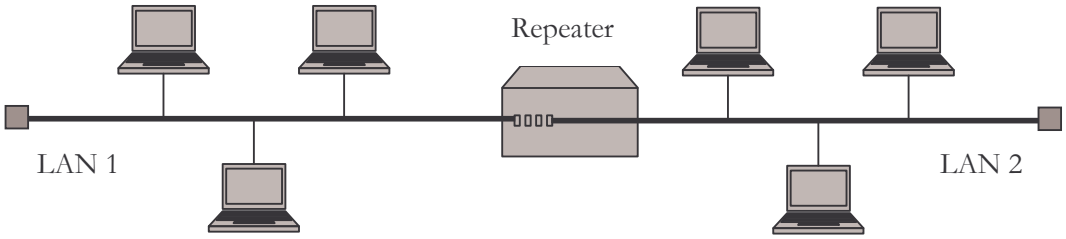


Figure 7.2 Repeater

A repeater allows us to extend the physical length of a network. A repeater forwards every frame; it has no filtering capability. For example, suppose source and destination devices are on same side (one segment) of repeater, repeater regenerates the signal and forwards it to another segment, even though there is no need to do so.

A repeater is a regenerator, not an amplifier. An amplifier cannot discriminate between the intended signal and noise; it amplifies equally everything fed to it. A repeater doesn't amplify the signal; it regenerates it. A repeater receives a weakened or corrupted signal; it creates a copy bit for bit, at the original strength.

Repeater is a networking component that extends a network by boosting the signal so that it can travel farther along the cabling. Digital signals traveling on cables weaken with distance—a phenomenon known as attenuation. A repeater is a form of digital amplifier (here amplifier means regenerator) that works at the physical layer (layer 1) of the Open

Systems Interconnection (OSI) reference model for networking to regenerate (amplify) the signal so that it can travel farther. Repeaters also perform other functions such as filtering out noise caused by electromagnetic interference (EMI), reshaping the signal, and correcting timing to remove signal jitter so that the signal can travel farther. Repeaters can also be used to join dissimilar media such as unshielded twisted-pair (UTP) cabling and thinnet, but they cannot be used to join dissimilar network architectures such as Ethernet and Token Ring. Repeaters are an inexpensive way to extend a network.

A repeater is used to regenerate a signal. However, note that a repeater does not work above the Physical layer so it do not have any knowledge of the protocols and their structure. If a packet is damaged (as with a collision), but the signal is good, the repeater will regenerate the bad packet.

A good example of the use of repeaters would be in a local area network using a star topology with unshielded twisted-pair cabling. The length limit for unshielded twisted-pair cable is 100 meters. The most common configuration is for each workstation to be connected by twisted-pair cable to a multi-port active concentrator. The concentrator amplifies all the signals that pass through it allowing for the total length of cable on the network to exceed the 100 meter limit.

7.3 BRIDGE

Bridge operates in data link layer of the OSI model. Bridges can divide a large network into smaller segments. Bridge is a networking component used either to extend or to segment networks. Bridges work at the OSI data-link layer. They can be used both to join dissimilar media such as unshielded twisted-pair (UTP) cabling and fiber-optic cabling, and to join different network architectures such as Token Ring and Ethernet. Bridges regenerate signals but do not perform any protocol conversion, so the same networking protocol (such as TCP/IP) must be running on both network segments connected to the bridge. Bridges can also support Simple Network Management Protocol (SNMP), and they can have other diagnostic features.

Bridges contain logic that allows them to keep the traffic for each segment separate. Security is provided through this partitioning of traffic. A bridge has a table used in frame filtering decisions. Bridges know the physical addresses of all stations connected to them. When frame enters a bridge, the bridge not only regenerates the signal but checks the address of the destination and forwards the new copy only to the segment to which the address belongs.

A bridge is a device that allows you to segment a large network into two smaller, more efficient networks. A bridge monitors the information traffic on both sides of the network so that it can pass packets of information to the correct location. Most bridges can "listen" to the network and automatically Figure out the address of each computer on both sides of the bridge. The bridge can inspect each message and, if necessary, broadcast it on the other side of the network.

The bridge manages the traffic to maintain optimum performance on both sides of the network. You might say that the bridge is like a traffic cop at a busy intersection during rush hour. It keeps information flowing on both sides of the network, but it does not allow unnecessary traffic through.

Bridges operate by sensing the source MAC addresses (MAC address is also called as physical address, which is 48-bit long and generally represented using hexadecimal notations. For example: 00:1A:58:B3:C8:01) of the transmitting nodes on the network and automatically builds an internal routing (also called as filtering table or look-up table).

This table is used to determine which connected segment will nearest to route packets to, and it provides the filtering capability that bridges are known for. If the bridge knows which segment a packet is intended for, it forwards the packet directly to that segment. If the bridge doesn't recognize the packet's destination address, it forwards the packet to all connected segments except the one it originated on. And if the destination address is in the same segment as the source address, the bridge drops the packet. Bridges also forward broadcast packets to all segments except the originating one.

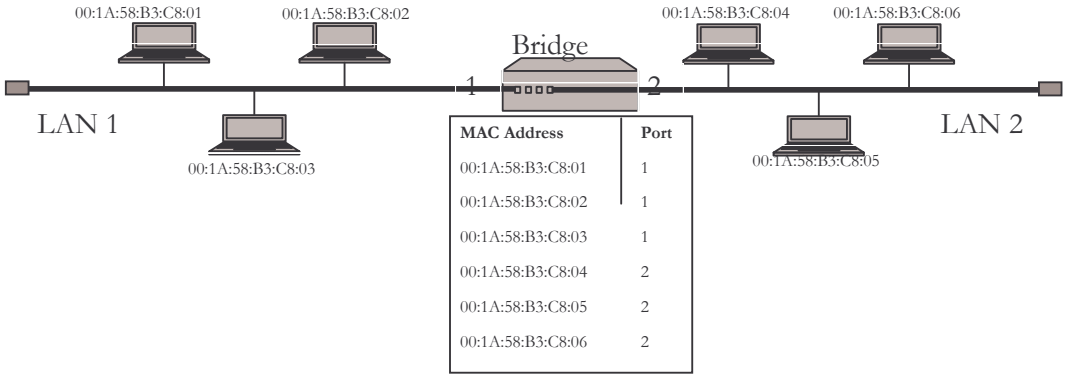


Figure 7.3 Bridge

TYPES OF BRIDGES

To select between segments, a bridge must have look-up table that contains the physical address of every station connected to it. The table indicates to which segment each station belongs.

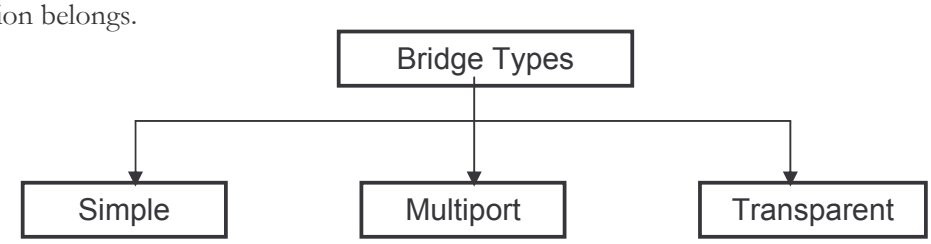


Figure 7.4 Types of Bridge

Simple Bridge

Simple bridges are the most primitive and least expensive type of bridges. A simple bridge links *two segments* and contains a table (generally called as routing table or filtering table or look-up table) that lists the addresses of all the stations included each segment of network. Primitive means address must be entered manually. Whenever new station is added, the table must be modified by adding new stations MAC address. If station is removed, its address must be deleted from the table. MAC address is also called as physical address, which is 48-bit long and generally represented using hexadecimal notations. For example: 00:A9:CA:42:7B:E9.

Installation and maintenance of simple bridge is time consuming, because network administrator must have to add MAC address on each side of the segment manually. Simple bridge links only two segments (LAN).

Multiport Bridge:

A ***Multiport bridge*** can be used to connect more than two LANs or segments.

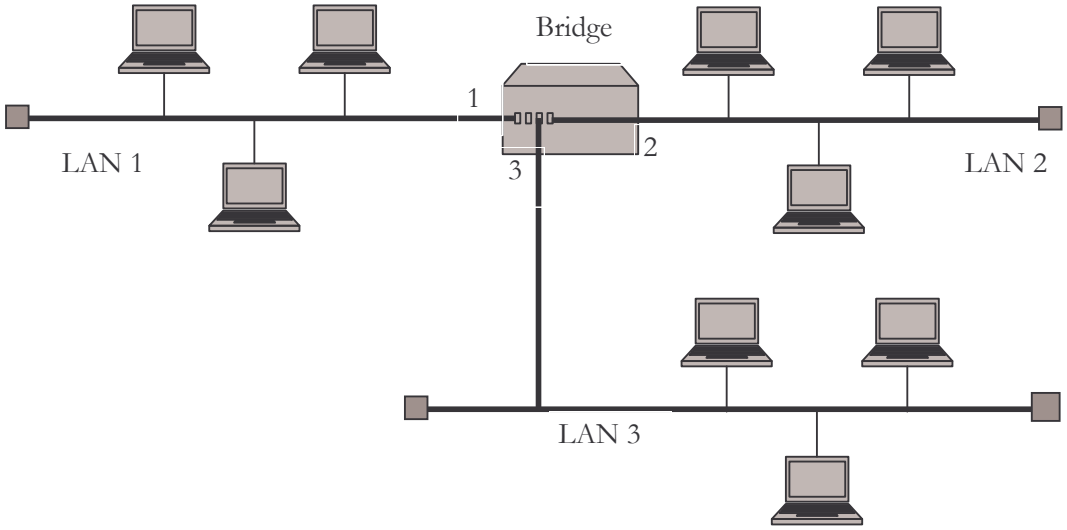


Figure 7.5 Multiport Bridge

Transparent Bridge:

A ***transparent,*** or learning bridge builds its table of station addresses on its own as it performs its bridge functions. Using the information contained in frame it processes. A bridge learns which devices reside on each cable segment. It then forwards frames it has receives according to the physical destination address of the packet. This process is used by transparent or learning bridges.

When the transparent bridge is first installed, its table is empty. It uses the source address in packet to build its table. It checks the destination to decide where to send the packet. If it doesn't yet recognize the destination address, it relays the packet to all of the stations of all segments. When bridges are installed redundantly, which means that two LANs may be connected by more than one bridge. Bridges may create a loop. To avoid this situation bridges use ***spanning tree algorithm*** or ***source routing***. In source routing, the source of the each packet defines the bridge and LANs through which the packet should go before reaching the destination. Source-routing bridges use information provided by the packet's source to determine the path the packet takes.

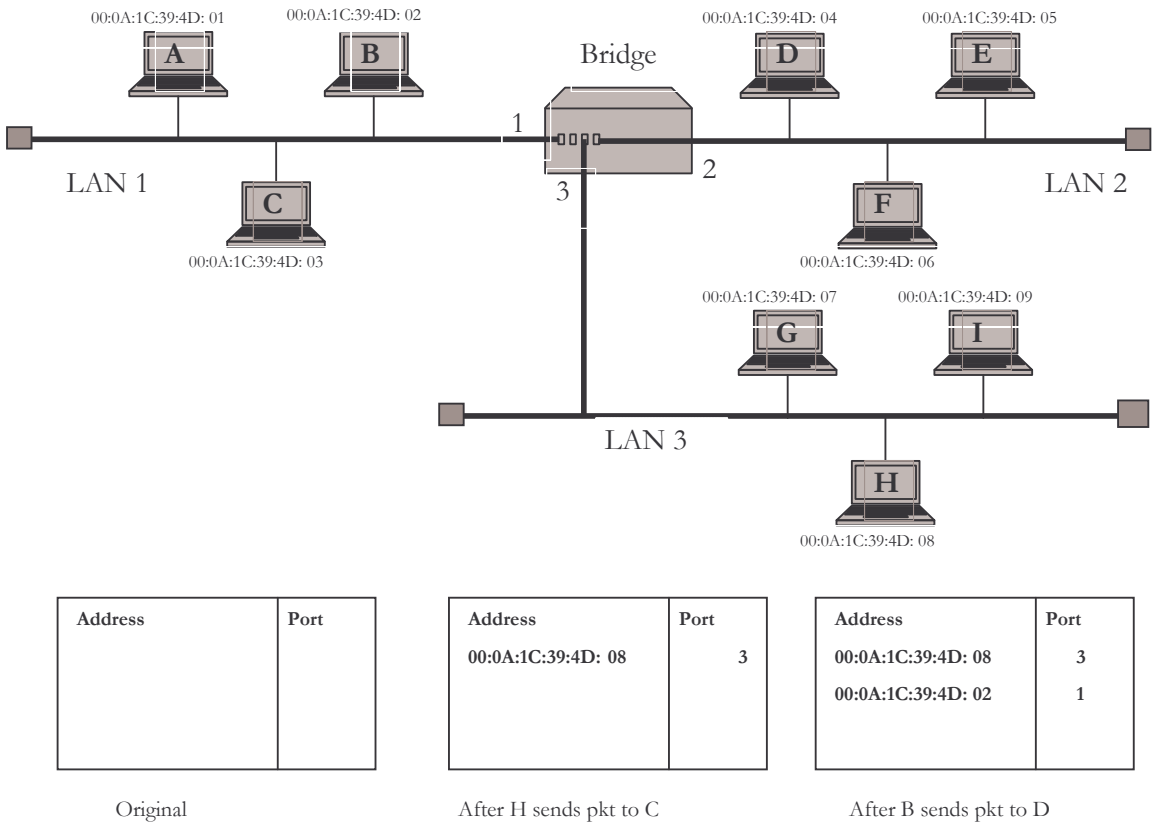


Figure 7.6 Transparent Bridge

7.4 HUB

Hubs are the central elements of LANs where the physical cabling infrastructure is concentrated, hence, the name hub or concentrator. Hubs can support one or several LAN types such as Ethernet, token-ring, FDDI, ATM. Hub operates at layer 1 (physical layer) of ISO/OSI model.

Typically, a hub can act as a multiport repeater with the total bandwidth being limited by the transport protocol: Ethernet at 10 Mbps or 100Mbps, token-ring at 4 and 16 Mbps, FDDI at 100 Mbps, and ATM at 25, 52, 100 and 155 Mbps.

Hubs are the foundation of traditional 10BaseT Ethernet networks. The hub receives signals from each station and repeats the signals to all other stations connected to the hub. In active hubs (which all of today's hubs are), the signal received from one port is regenerated (amplified) and retransmitted to the other ports on the hub. Hubs thus perform the function of a repeater and are sometimes called multiport repeaters. From a logical cabling point of view, stations wired into a hub form a star topology. Hubs generally have RJ-45 ports for unshielded twisted-pair (UTP) cabling, and they range in size from 4 to 24 or more ports for connecting stations to the hub, plus one or more uplink ports for connecting the hub to other hubs in a cascaded star topology. Hubs generally have various light-emitting diode (LED) indicator lights to indicate the status of each port, link status, collisions, and so on. Hubs with several different types of LAN connectors such as RJ-45, BNC, and AUI are commonly called combo hubs.

In a network where HUB is used as a central controller, if one device sends packet to another device, that packet is forwarded from source device to central controller (Hub), hub in turn, regenerates the signal and forwards that frame on all the ports of it except the one where it was arrived. For example, consider a four port hub, Device "A" connected to port 1 of the hub, device "B" to port 2 of the hub, device "C" to port 3 of the hub and device "D" to port 4 of the hub. Suppose, device "A" sends frame to device "B", device "A" forwards it to hub, hub in turn regenerates the frame and puts the copy of frame on port 2, 3, and 4.

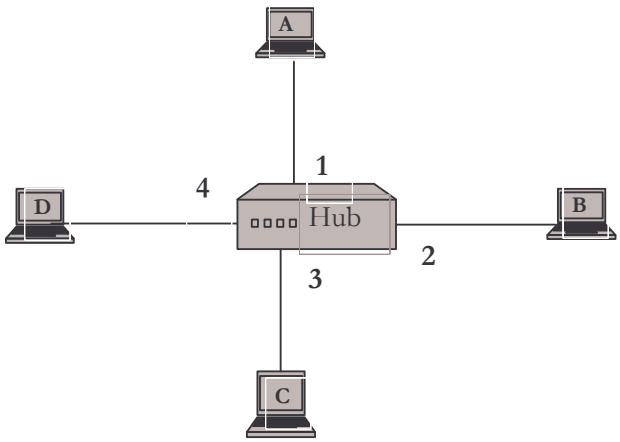


Figure 7.7 Hub

7.5 SWITCH

A switch is a network device with multiple ports in one network whose task is to copy frames from one port to another. Switches operate at Layer 2 of the OSI Model, the Data-Link Layer. This is in contrast to routers, which operate at Layer 3 of the OSI Model, the Network Layer. A switch stores the MAC Address of every device which is connected to it. The switch will then evaluate every frame that passes through it. The switch will examine the destination MAC Address in each frame. Based upon the destination MAC Address, the switch will then decide which port to copy the frame to. If the switch does not recognize the MAC Address, it will not know which port to copy the frame to. When that happens, the switch will broadcast the frame to all of its ports.

Switch is a concentrator, a device that provides a central connection point for cables from workstations, servers, and peripherals. In a star topology, twisted-pair wire is run from each workstation to a central switch/hub. Most switches are active, that is, they electrically regenerates the signal as it moves from one device to another. Switches no longer broadcast network packets as hubs did in the past; they memorize addressing of computers and send the information to the correct location directly.

Switches themselves are hardware devices not entirely different in appearance from routers, hubs, and bridges. However, three important factors separate switches from their networking brethren: overall speed (switches are much faster); forwarding methodology or electronic logic (smarter); and higher port counts. In contrast to the functionality of bridges and routers, which traditionally utilize the less effective and more expensive microprocessor and software methods, switches direct data frames across the various segments in a faster and more efficient manner through an extensive reliance upon on-board logic, through Application-Specific Integrated Circuits (ASICs).

In a network where Switch is used as a central controller, if one device sends packet to another device, that packet is forwarded from source device to central controller (switch), switch in turn, regenerates the signal, checks the destination address in the frame and forwards that frame towards destination. For example, consider a four port switch, Device "A" connected to port 1 of the switch, device "B" to port 2 of the switch, device "C" to port 3 of the switch and device "D" to port 4 of the switch. Suppose, device "A" send frame to device "B", device "A" forwards it to switch, switch in turn regenerates the frame, checks the destination MAC address in frame and puts the copy of frame on port 2 only, where device "B" is connected.

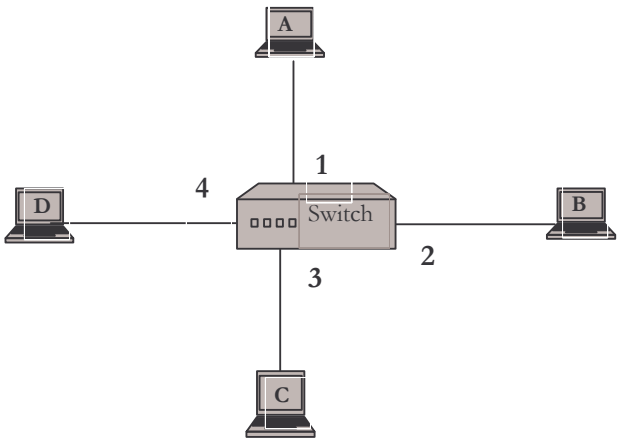


Figure 7.8: Switch

By default, switches break up *collision domains*. This is an Ethernet term used to describe a network scenario wherein one particular device sends a packet on a network segment, forcing every other device on that same segment to pay attention to it. At the same time, a different device tries to transmit, leading to a collision, after which both devices must retransmit, one at a time. Not very efficient! This situation is typically found in a hub environment where each host segment connects to a hub that represents only one collision domain and only one broadcast domain. By contrast, each and every port on a switch represents its own collision domain.

Switch Vs Bridge

A switch is designed with chips which are solely used for fast switching capability called as ASIC chip, on the other hand bridge uses software which runs on top of hardware which hence takes longer to perform the switching function. A switch generally has more ports then a bridge. Bridges had maximum 16 ports, whereas switches can have unlimited number of ports theoretically and also have more configuration possibilities.

Bridges and switches read each frame as it passes through the network. The layer-2 device then puts the source hardware address in a filter table and keeps track of which port the frame was received on. This information (logged in the bridge's or switch's filter table) is what helps the machine determine the location of the specific sending device. After a filter table is built on the layer-2 device, it will only forward frames to the segment where the destination hardware address is located. If the destination device is on the same segment as the frame, the layer-2 device will block the frame from going to any other segments. If the destination is on a different segment, the frame can only be transmitted to that segment. This is called *transparent bridging*.

When a switch interface receives a frame with a destination hardware address that isn't found in the device's filter table, it will forward the frame to all connected segments. If the unknown device has sent the "mystery frame" replies to this forwarding action, the switch updates its filter table regarding that device's location. But in the event the destination address of the transmitting frame is a broadcast address, the switch will forward all broadcasts to every connected segment by default.

All devices that the broadcast is forwarded to are considered to be in the same broadcast domain. This can be a problem; layer-2 devices propagate layer-2 broadcast storms that choke performance, and the only way to stop a broadcast storm from propagating through an internetwork is with a layer-3 device—a router.

The biggest benefit of using switches or bridges instead of hubs in your internetwork is that each switch port is actually its own collision domain. (Conversely, a hub creates one large collision domain.) But even armed with a switch, you still can't break up broadcast domains. Neither switches nor bridges will do that. They'll typically simply forward all broadcasts instead.

BROADCAST STORM

Redundant links between switches are a good idea because they help prevent complete network failures in the event one link stops working. It sounds great, but even though

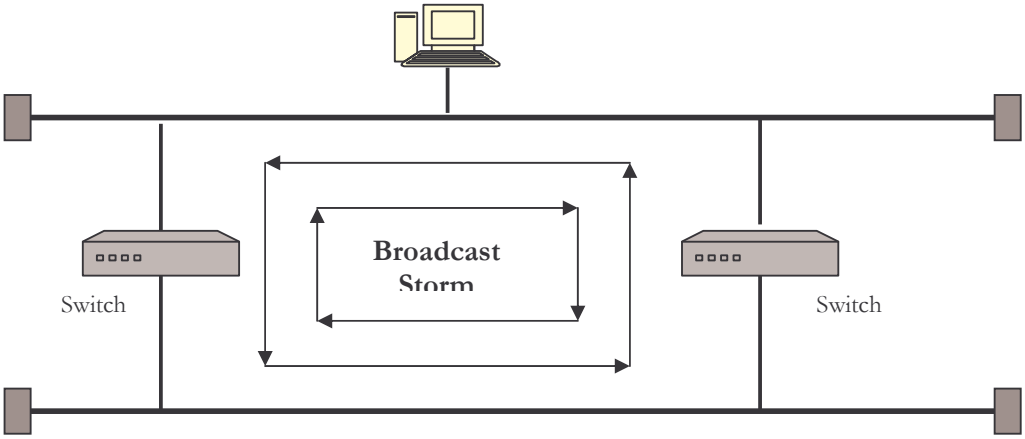


Figure 7.9 Broadcast Storm

redundant links can be extremely helpful, they often cause more problems than they solve. This is because frames can be flooded down all redundant links simultaneously, creating network loops as well as other evils. If no loop avoidance schemes are put in place, the switches will flood broadcasts endlessly throughout the internetwork. This is sometimes referred to as a *broadcast storm*. Following figure illustrates how a broadcast can be propagated throughout the network. Observe how a frame is continually being flooded through the internetwork's physical network media: A device can receive multiple copies of the same frame, since that frame can arrive from different segments at the same time.

SWITCH PROPERTIES

How do you pick a switch that will suit your needs? You'll need to spend a little time getting to know switches, some of their more important features, and how they do that

which they do so well. Once you've got that information under your belt, you should be in a fairly good position to make authoritative choices about switch purchases.

Static Switching Versus Dynamic Switching

If you've gotten to the point where your network is extremely congested and you've called vendors in for demonstrations and quotes, be extraordinarily wary if their solutions depend on *static switches*. Although the devices that you evaluate during the course of re-engineering your network may or may not be explicitly referred to as static switches, take a good look at the functionality of the particular piece or pieces of hardware. If the products perform in a fashion that appears to make them nothing more than glorified hubs, chances are that you really don't want to invest in that type of switch. After all, the point of this whole operation is to segment and intelligently control inter-segment traffic, thus reducing congestion. Static switches just don't hit the mark.

On the other end of the spectrum are the products that you do want to consider seriously: *dynamic switches*. Dynamic switches not only pay special attention to the forwarding of packets to their proper destination, but also maintain a table that associates individual nodes with the specific ports to which they are connected. This information, updated each time a particular machine transmits across the network, or perhaps at operator-defined intervals, keeps the switch's information as to node/port combinations up to date, allowing the switch to quickly direct frames across the proper segments, rather than across all segments on the switch.

Dynamic switches will continue to save you huge amounts of time and energy long after you first integrate them into your network. Because dynamic switches update their forwarding tables every time devices broadcast across the network, you can rearrange your network, switching workstations from port to port to port, until your network is configure in the manner that suits you best, or you're blue in the face, whichever comes first! The tables will be updated automatically and your network won't go down.

Segment Switching Versus Port Switching

There's a great ongoing debate about whether segment switching or port switching provides the optimum solution for resolving network congestion crises. It all boils down to a question of cash on hand: If you've got the cash, go with port switching; if not, then segment switching will be the order of the day. What's great about the segment-versus-port debate is that, for a change, you win either way.

Segment switches are able to handle the traffic from an entire network segment on each port, allowing you to connect a higher number of workstations or segments with fewer switches/physical ports. The great aspect of segment switches is that they are also capable of handling a single workstation on each port (in essence, a segment with one node). This will allow the network engineer to prearrange machines requiring only intermittent network access along the same segment, sharing one (relatively) low-traffic 10Mbps pipe. At the same time, high-end machines, such as network and database servers, optical drives, and other devices can be connected with a one device/one port scheme, allowing these high-bandwidth and critical devices their own dedicated path to the greater network without having to compete with someone's Internet game for network access. Because of the inevitable cost controls that you encounter on a daily basis, segment switching is the preferred and most readily implemented solution because it requires little in the way of additional expenditures for hardware, additional cabling, and so on.

Port switches (also referred to as *switching hubs*) are designed to accommodate a single device on each physical port. This is a network manager's dream--each workstation, server, and random device would have its own dedicated, 10Mbps path to the rest of the network. However, implementing a port-switching solution demands a good deal of capital for additional wiring (cable runs are needed from each device directly to the switch) and enough switches to provide the requisite number of physical ports. Additionally, as your network grows, you'll be faced with significantly increased expansion costs because you'll need new cable runs and possibly entirely new switches every few months. Again, if you've got lots of cash, this is a great option; you'll have quite the impressive network. However, whatever route you choose, you'll certainly end up with a much better network than you had prior to implementing switching.

Cut-Through Switching

Cut-through switching helps speed network communication by forwarding packets much sooner than traditional switching configurations will allow. This is achieved by forwarding packets to their destination machine prior to receiving them in their entirety, sending them on as soon as the switch is able to determine the destination address. Although this generally reduces network latency, cut-through switching can often allow many bad packets to eat up available bandwidth. To prevent this, reconfigure your switch to allow for a marginally longer delay between the receipt and forwarding of packets. Ideally, as soon as the switch receives the packet, it should buffer 64 bytes to ensure that the possibility of packet errors has been eliminated. After the possibility of these errors has passed, the switch can then forward the packets across the appropriate segment to the destination host. This slightly increases network latency, though it will provide for faster speeds than floor-model switching. Unfortunately, if yours is an extraordinarily busy network, the benefits of cut-through switching will be less noticeable, and will reach their limits much sooner than in a less intensive environment.

Store and Forward Switching

Store-and-forward switching devices, as take an entirely different approach than cut-through switching. It's very much like the tortoise and the hare, with store-and-forward devices playing the slower, yet more dependable, role of the two.

Instead of the faster send-it-as-soon-as-you-can rule used by cut-through devices, store-and-forward devices wait until the entire packet is received by the switch, only then sending it on to its destination. This lets the switch verify the packet's CRC and eliminate the possibility of other transmission errors, allowing for highly reliable data transmission across your network. Although this doesn't strictly increase network performance, it *does* eliminate the additional transmissions that must occur as a result of packet errors that otherwise would have occupied network resources, thus providing an associated speed increase.

7.6 ROUTER

Routers are used to create internetworks, or collections of networks. These internetworks function independently, but are able to share information. They are able to provide fault-tolerance in networks by choosing between multiple paths. If a network is designed with more than one path from Office A to Office B, the router will choose the best path for the packets. However, if the best path is no longer available, the router is able to select another path. They can connect multiple network segments and filter traffic like

bridges. It is important to realize that, unlike bridges or switches, routers do not forward broadcasts and can be used to prevent broadcast storms.

Routers are used to connect networks together and route packets of data from one network to another. Routers, by default, break up a *broadcast domain*, which is the set of all devices on a network segment that hear all broadcasts sent on that segment.

Breaking up a broadcast domain is important because when a host or server sends a network broadcast, every device on the network must read and process that broadcast—unless you've got a router. When the router's interface receives this broadcast, it can discard the broadcast without forwarding it on to other networks.

Even though routers are known for breaking up broadcast domains by default, it's important to remember that they break up collision domains as well.

Two advantages of using routers in your network:

1. They don't forward broadcasts by default.

2. They can filter the network based on layer-3 (Network layer) information (i.e., IP address.

Primarily, layer-3 machines (such as routers) need to locate specific networks, whereas layer-2 machines (switches and bridges) need to eventually locate specific devices. So, networks are to routers as individual devices are to switches and bridges. And routing tables that "map" the internetwork are for routers, as filter tables that "map" individual devices are for switches and bridges.

Routers operate at the Network layer of the OSI model. Because of this, not all protocols can be routed, only those with Network information. Because routers operate at the Network layer, they can easily perform the function of translation bridges, i.e., sending information over different network architectures. This is very often the primary task for a router. A good example of this is illustrated in an internetwork which consists of two separate buildings on a campus. The buildings use Ethernet and are connected using FDDI. In this configuration, the router is used to send data between the two buildings and converts all data from Ethernet to FDDI and back to Ethernet. That's why routers are also called as **media converter**. Routers handle unknown destinations and broadcasts differently than bridges. When a router receives a packet whose destination it does not know, it discards the packet. Routers also do not forward broadcasts or corrupted packets. This is very useful in limiting the effect of network failures and broadcast storms.

Routing tables are used by routers to determine the path a packet must take. Routing algorithms or routing is classified into two types: static routing (or non-adaptive) and dynamic routing (or adaptive).

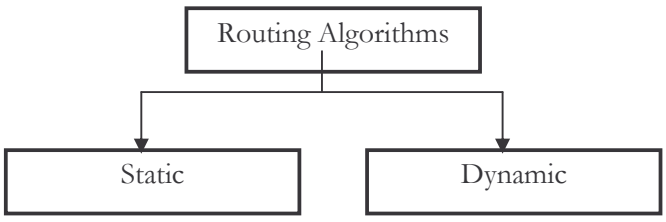


Figure 7.10 Routing Algorithms

Static routing is the process of predefining route paths across data networks and can be used to conserve LAN and WAN bandwidth and optimize processing time. **Dynamic routing** adjusts routing patterns within the network in accordance with varying and uncertain offered traffics, to make better use of spare capacity in the network resulting from dimensioning upgrades or forecasting errors, and to provide extra flexibility and robustness to respond to failures or overloads. With a static table it is the responsibility of the administrator to manually update the table, but with a dynamic table, the routers send changes only to other router in network. This takes up less bandwidth than transmitting the entire table.

Distance-vector is older and takes into consideration how many routers, called hops, the packet has to take before it reaches its destination. The disadvantage of this is that it fails to take into consideration other factors, such as the speed of the line, the congestion of the line and cost of the line. This is considered with the **link-state route** discovery algorithms.

7.7 GATEWAY

Gateways are intricate pieces of equipment, which operate at all layers of the OSI model to convert data from one format to another. They are most often found when connecting PC networks to mainframe computers. Because of the detail involved in making these types of conversions, gateways are slower and more expensive than other networking equipment. Some types of Gateways can be provided for via software. This would be accomplished by installing the module onto the networking operating system software. Gateways are also called as **protocol *converter***.

Gateway is a term for a broad category of network components that allow communication between different networking architectures and different protocols. Gateways are commonly used to provide connectivity between two different protocol stacks that might be running on different systems. Examples include the following:

- E-mail gateways—for example, a gateway that receives Simple Mail Transfer Protocol (SMTP) e-mail, translates it into a standard X.400 format, and forwards it to its destination.
- Gateways between a Systems Network Architecture (SNA) host and computers on a TCP/IP network, such as the one provided by Microsoft SNA Server.
- A packet assembler/disassembler (PAD) that provides connectivity between a local area network (LAN) and an X.25 packet-switching network

A gateway is usually a dedicated device or a set of services running on a dedicated computer. Gateways are essentially devices that direct network traffic in some fashion and translate that information.

SUMMARY

Internetworking involves connecting two or more distinct computer networks or network segments together. Devices, which are used for connecting two or more than two networks are classified into two categories namely: networking device and internetworking devices.

A repeater (or regenerator) is an electronic device that operates on physical layer of OSI model. A switch is a network device with multiple ports in one network whose task is to copy frames from one port to another. Switches operate at Layer 2 of the OSI Model, the Data-Link Layer.

Routers are also called as media converter. Routers are used to create internetworks, or collections of networks. Routers operate at the Network layer of the OSI model. Gateways are also called as protocol *converter*.

PRACTICE SET

Review Questions

1. Write a Short note on Repeaters
2. Write a Short note on Bridge
3. Write a Short note on Switches
4. Explain difference between hub and switch.
5. Explain difference between switch and bridge.
6. Explain difference between switch and router.
7. Explain difference between router and gateway.
8. Why gateways are required?
9. What are the types of routing algorithm?
10. Explain broadcast storm.

Multiple Choice Questions

1. What is purposes for segmentation with a bridge?
A) Add more broadcast domains.
B) Create more collision domains.
C) Decreases broadcast domain.
D) Allow more broadcasts for users.

2. What is the reason to segment a network with a bridge?
A) Increase the amount of collision on a segment.
B) Decrease the amount of broadcast on a segment.
C) Reduce broadcast within a broadcast domain.
D) Increase the number of collision domains.

3. Which of the following is the reason for breaking up a network into two segments with a router?
A) To create fewer broadcast domains
B) To create more bandwidth domains
C) To create one large broadcast domain
D) To stop one segment's broadcasts from being sent to the second segment

4. What is a reason you want to use switches in your network instead of hubs?
A) They are less expensive.
B) Switches are faster than hubs at reading frames.

C) Switches create more collision domains.

D) Switches do not forward broadcasts.

5. MAC address is also called as…
A) Physical Address B) Logical Address

C) IP address D) None of the above

6. Layer 2 Bridge takes it filtering decision based on ……
A) MAC address B) IP Address

C) Logical Address D) Port address

7. Layer 2 switch takes it filtering decision based on ……
A) MAC address B) IP Address C) Logical Address D). Port address

8. Repeater is …….. layer device.
A) Physical B) Data Link C) Network D) Application

9. Router is …….. layer device.
A) Physical B) Data Link C) Network D) Application

10. Bridge is …….. layer device.
A) Physical B) Data Link C) Network D) Application

11. Switch is …….. layer device.
A) Physical B) Data Link C) Network D) Application

12. NIC (Network Interface Card) is …….. layer device.
A) Physical B) Data Link C) Network D) Application

13. Hub is …….. layer device.
A) Physical B) Data Link C) Network D) Application

14. Which is Internetworking device?
A) Hub B) Repeater C) Switch D) Router

15. …… is also called as protocol converter.
A) Bridge B) Switch C) Gateway D) Router

16. ……. use ASIC chip.
A) Hub B) Switch C) Bridge D) None of the above

* * *

Chapter 8 Data Link Layer

In this chapter, we focus on the data link layer, which is responsible for transferring a datagram across an individual link.

OBJECTIVES OF THIS CHAPTER

After completing this chapter, the student should be able to:

- define duties of data link layer (DLL)
- define sub-layers of DLL
- explain elementary protocols of DLL
- explain examples of DLL protocols
- explain HDLC protocol
- explain PPP protocol

8.1 INTRODUCTION

Data Link Layer (DLL) is a layer 2 of the Open Systems Interconnection (OSI) reference model, which converts frames of data into raw bits for the physical layer and is responsible for framing, flow control, error correction, and retransmission of frames. MAC addresses are used at this layer, and bridges and network interface cards (NICs) operate at this layer.

The purpose of the data link layer is to transfer blocks of data without error between two adjacent devices or nodes. Adjacent devices are physically connected by a communication channel such as telephone lines, coaxial cables, optical fibers, or satellites. The implication of such a physical link is that the data bits are delivered in exactly the same order in which they are sent. The physical link has no storage capacity; therefore the delay involved in transmission of data between tow adjacent devices is the propagation delay over the link. Transmission of data over the link would be very simple indeed if no error ever occurred. Unfortunately, this is not so in a real physical link for a number of reasons:

• Natural phenomena such as noises and interference are introduced into the link causing errors in detecting the data.

• There is a propagation delay in the link.

• There is a finite data processing time required by the transmitting and receiving stations.

A data link protocol thus has to be designed to ensure an error-free transmission and also to achieve an efficiency of the data transfer as high as possible.

The data-link layer establishes and maintains the data link for the network layer above it. It ensures that data is transferred reliably between two stations on the network. A number of protocols can be implemented at this layer depending on whether you are establishing local area network (LAN) or wide area network (WAN) connections between stations. Data-link protocols are responsible for functions such as addressing, frame delimiting and sequencing, error detection and recovery, and flow control.

At layer 2, user's messages broken up into chunks. Control information (headers and trailers) is added to each chunk to make up a frame.

For LANs, the Project 802 standards of the Institute of Electrical and Electronics Engineers (IEEE) separate the data-link layer into two sub-layers:

The *Logical Link Control (LLC)* layer, the upper of the two layers, which is responsible for flow control, error correction, and re-sequencing functions for connection-oriented communication, but which also supports connectionless communication

The *Media Access Control (MAC)* layer, the lower of the two layers, which is responsible for providing a method for stations to gain access to the medium

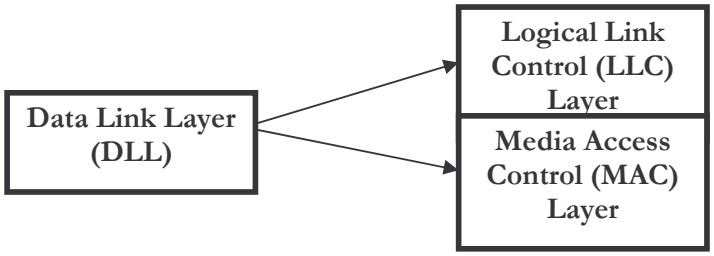


Figure 8.1 Sub-layers of Data Link Layer

Examples of data-link protocols for local area networking (LAN) include the following:

IEEE 802.3, which provides the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method for baseband Ethernet networks.

IEEE 802.5, which provides the token-passing access method for baseband token ring implementations.

For Wide Area Networks (WANs), data-link layer protocols encapsulate LAN traffic into frames suitable for transmission over WAN links. Common data-link encapsulation methods for WAN transmission include the following:

Point-to-point technologies are Point-to-Point Protocol (PPP) and High-level Data Link Control (HDLC) protocol.

Multipoint technologies are frame relay, Asynchronous Transfer Mode (ATM), Switched Multi-megabit Data Services (SMDS), and X.25.

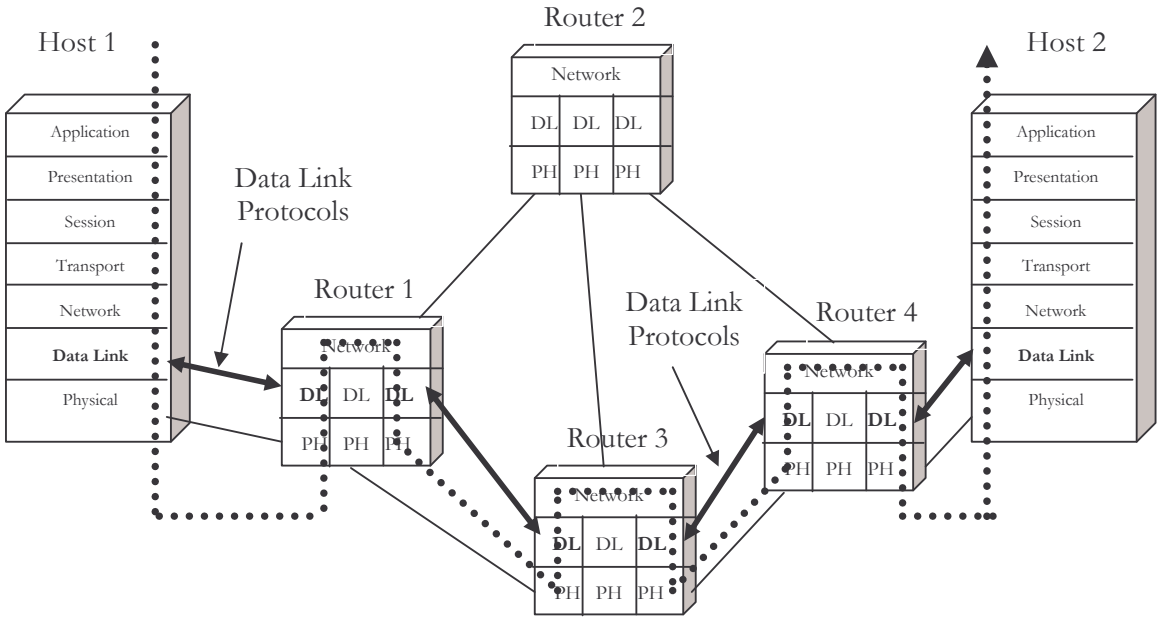
8.2 DUTIES OF DATA LINK LAYER

The basic function or duty of the layer is to transmit frames over a physical communication link. Transmission may be *half duplex* or *full duplex*. To ensure that frames are delivered free of errors to the destination machine a number of requirements are placed on a data link protocol. The protocol should be capable of performing:

1. Identification of frame that is start and end of the frame.
2. The transmission of frames of any length up to a given maximum. Any bit pattern is permitted in a frame.
3. The detection of errors occurred during transmission.
4. The retransmission of frames which were damaged by errors.
5. The assurance that no frames were lost.
6. In a multipoint configuration:
    a. Some addressing mechanism must be used to direct the frames.
    b. Some mechanism must be used for preventing conflicts caused by simultaneous transmission by many stations.

As far as layer 2 is concerned a host message passed across the interface to it from the network layer is pure data, and every single bit of which is to be delivered to the other host.

The communication path starts at the source host, passes through a series of intermediate devices such as routers, and ends at the destination host. We will refer to the hosts and the routers simply as **nodes**, and to the communication channels that connect adjacent nodes along the communication path as **links**. In order to move a datagram from source host to destination host, the datagram must be moved over each of the *individual links* in the path. In this chapter, we focus on the **data link layer,** which is responsible for transferring a datagram across an individual link.



Note: DL means Data Link Layer
PH means Physical Layer

Figure 8.2 Data Link Layer

Router is an intermediate device used to route the packet to destination via best path. The network layer has the end-to-end job of moving transport-layer segments from the source host to the destination host, a link-layer protocol has the node-to-node job of moving a network-layer datagram over a *single link* in the path. An important characteristic of the link layer is that a datagram may be handled by different link-layer protocols on the different links in the path. For example, a datagram may be handled by Ethernet on the first link, PPP (point-to-point) on the last link, and frame relay on all intermediate links. It is important to note that the services provided by the different link-layer protocols may be different. For example, a link-layer protocol may or may not provide reliable delivery. Thus, the network layer must be able to accomplish its end-to-end job in the face of a varying set of individual link-layer services.

In order to gain insight to the link layer and how it relates to the network layer, let's consider a transportation analogy. Consider a travel agent who is planning a tour for a tourist traveling from Pune, India to Lausanne, Switzerland. Suppose the travel agent decides that it is most convenient for the tourist to take a private bus from Pune to Mumbai airport, then a plane from Mumbai airport to Geneva airport, and finally a train from Geneva to Lausanne's train station. (There is a train station at Geneva's airport.)

Once the travel agent makes the three reservations, it is the responsibility of the Pune private bus company to get the tourist from Pune to Mumbai airport; it is the responsibility of the airline company to get the tourist from Mumbai airport to Geneva; and it is responsibility of the Swiss train service to get the tourist from the Geneva to Lausanne. Each of the three segments of the trip is "direct" between two "adjacent" locations. Note that the three transportation segments are managed by different companies and use entirely different transportation modes (bus, plane and train). Although the transportation modes are different, they each provide the basic service of moving passengers from one location to an adjacent location. This *service* is used by the travel agent to plan the tourist's trip. In this transportation analogy, the tourist is analogous to a datagram, each transportation segment is analogous to a communication link, the transportation mode is analogous to the link-layer protocol, and the travel agent who plans the trip is analogous to a Network layer routing protocol (routing protocol discuss in chapter Network Layer).

The basic service of the link layer is to "move" a datagram from one node to an adjacent node over a single communication link. But the details of the link-layer service depend on the specific link-layer protocol that is employed over the link. Possible services that can be offered by a link-layer protocol include:

- **Framing and link access:** Almost all link-layer protocols encapsulate each network-layer datagram within a link-layer frame before transmission onto the link. A frame consists of a data field, in which the network-layer datagram is inserted, and a number of header fields and trailer fields. A data link protocol specifies the structure of the frame, as well as a channel access protocol that specifies the rules by which a frame is transmitted onto the link. For point-to-point links that have a single sender on one end of the link and a single receiver at the other end of the link, the link access protocol is simple (or non-existent) - the sender can send a frame whenever the link is idle. The more interesting case is when multiple nodes share a single broadcast link - the so-called multiple access problem. Here, the channel access protocol serves to coordinate the frame transmissions of the many nodes;
- **Flow control:** The nodes on each side of a link have a limited amount of packet buffering capacity. This is a potential problem, as a receiving node may receive frames at a rate faster than it can process the frames (over some time interval). Without flow control, the receiver's buffer can overflow and frames can get lost.
- **Error detection:** A node's receiver can incorrectly decide that a bit in a frame to be a zero when it was transmitted as a one (and vice versa). These errors are introduced by signal attenuation and electromagnetic noise. Because there is no need to forward a datagram that has an error, many link-layer protocols provide a mechanism for a node to detect the presence of one or more errors. This is done by having the transmitting node set error detection bits in the frame, and having the receiving node perform an error check. Error detection is a very common service among link-layer protocols. Error detection in the link layer is usually more sophisticated and implemented in hardware.
- **Error correction:** Error correction is similar to error detection, except that a receiver can not only detect whether errors have been introduced in the frame but can also determine exactly where in the frame the errors have occurred (and hence correct these errors). Some protocols (such as ATM) provide link-layer error correction for the frame header rather than for the entire frame.

- **Half-Duplex and Full-Dulpex:** With full-duplex transmission, both nodes at the ends of a link may transmit packets at the same time. With half-duplex transmission, a node cannot both transmit and receive at the same time.

8.3 ELEMENTARY DATA LINK LAYER PROTOCOLS

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.

Figure 8.3 Data Link Layer Frame

When a frame arrives at the receiver, the hardware computes the checksum. If the checksum is incorrect (i.e., there was a transmission error), the data link layer discards the frame. If the frame arrived undamaged, the data link layer can acquire the frame for inspection. 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 data link layer header and trailer portion of the frame never be given to the network layer: 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 requiring changes to the network layer's software.

An Unrestricted Simplex Protocol

As an initial example we will consider a protocol that is as simple as it can be. In this protocol:

- Data are transmitted in one direction only
- The transmitting (Tx) and receiving (Rx) devices are always ready
- Processing time can be ignored
- Infinite buffer space is available
- The communication channel between the data link layers never damages or loses frames. That is no errors occur during transmission.

A simplex stop-and-wait protocol

Now we will drop the most unrealistic restriction used in an unrestricted simplex protocol: the ability of the receiving device to process incoming data infinitely quickly.

In this protocol we assume that:

- Data are transmitted in one direction only
- No errors occur
- The receiver can only process the received information at a finite rate

These assumptions imply restriction on transmitter that it cannot send frames at a rate faster than the receiver can process them.

The problem here is how to restrict transmitter? or how to prevent the sender from flooding the receiver?

A general solution to this problem is to have the receiver provide some sort of feedback to the sender. The process could be as follows: The receiver send an acknowledge frame back to the sender telling the sender that the last received frame has been processed and passed to the host; permission to send the next frame is granted. The sender, after having sent a frame, must wait for the acknowledge frame from the receiver before sending another frame. This protocol is known as *stop-and-wait*.

Protocols in which the sender sends one frame and then waits for an acknowledgement before proceeding are called stop-and-wait. 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.

A Simplex Protocol for a Noisy Channel

Now let us consider the normal situation of a communication channel that makes errors. Frames may be either damaged or lost completely. However, we assume that if a frame is damaged in transit, the receiver hardware will detect this when it computes the checksum. One suggestion is that the sender would send a frame, the receiver would send an ACK frame only if the frame is received correctly. If the frame is in error the receiver simply ignores it; the transmitter would time out and would retransmit it.

One fatal flaw with the above scheme is that if the ACK frame is lost or damaged, duplicate frames are accepted at the receiver without the receiver knowing it.

Consider the following scenario:

1. The network layer on machine "A" gives packet 1 to its data link layer. The packet is correctly received at machine "B" and passed to the network layer on machine "B". Machine "B" sends an acknowledgement frame back to machine "A".
2. The acknowledgement frame gets lost completely. It never arrives at machine "A"
3. The data link layer on machine "A" eventually times out. Not having received an acknowledgement, it (incorrectly) assumes that its data frame was lost or damaged during transmission and sends the frame containing packet 1 again.
4. The duplicate frame also arrives at the data link layer on machine "B" perfectly and is unwittingly passed to the network layer there. If machine "A" is sending a file to machine "B", part of the file will be duplicated. The protocol thus fails in this aspect.

To overcome this problem it is required that the receiver be able to distinguish a frame that it is seeing for the first time from a retransmission. One way to achieve this is to

have the sender put a sequence number in the header of each frame it sends. The receiver then can check the sequence number of each arriving frame to see if it is a new frame or a duplicate to be discarded.

The receiver needs to distinguish only 2 possibilities: a new frame or a duplicate; a 1-bit sequence number (0 or 1) is sufficient. At any instant the receiver expects a particular sequence number. Any arriving frame containing the wrong sequence number is rejected as a duplicate. A correctly numbered frame arriving at the receiver is accepted, passed to the host, and the expected sequence number is incremented by 1 (modulo 2).

8.4 SLIDINGWINDOW

In most practical situations there is a need for transmitting data in both directions (i.e. between 2 computers). A full duplex circuit is required for the operation. If stop and wait protocol or simplex protocol for noisy channel is used in these situations the data frames and ACK (control) frames in the reverse direction have to be interleaved. This method is acceptable but not efficient. An efficient method is to absorb the ACK frame into the header of the data frame going in the same direction. This technique is known as *piggybacking*.

When a data frame arrives at a receiver or destination, instead of immediately sending a separate ACK frame, the receiver restrains itself and waits until the host passes it the next message. The acknowledgement is then attached to the outgoing data frame using the ACK field in the frame header. In effect, the acknowledgement gets a free ride in the next outgoing data frame.

This technique makes better use of the channel bandwidth. The ACK field costs only a few bits, whereas a separate frame would need a header, the acknowledgement, and a checksum. An issue arising here is the time period that the receiver waits for a message onto which to piggyback the ACK. Obviously the receiver cannot wait forever and there is no way to tell exactly when the next message is available. For these reasons the waiting period is usually a fixed period. If a new host packet arrives quickly the acknowledgement is piggybacked onto it; otherwise, the receiver just sends a separate ACK frame to the sender.

Ideally, data throughput happens quickly and efficiently. And as you can imagine, it would be slow if the transmitting machine had to wait for an acknowledgment after sending each segment. But because there's time available *after* the sender transmits the data segment and *before* it finishes processing acknowledgments from the receiving machine, the sender uses the break as an opportunity to transmit more data. The quantity of data segments that the transmitting machine is allowed to send without receiving an acknowledgment for them is called a *window*. Windows are used to control the amount of outstanding, unacknowledged data segments. So the size of the window controls how much information is transferred from one end to the other.

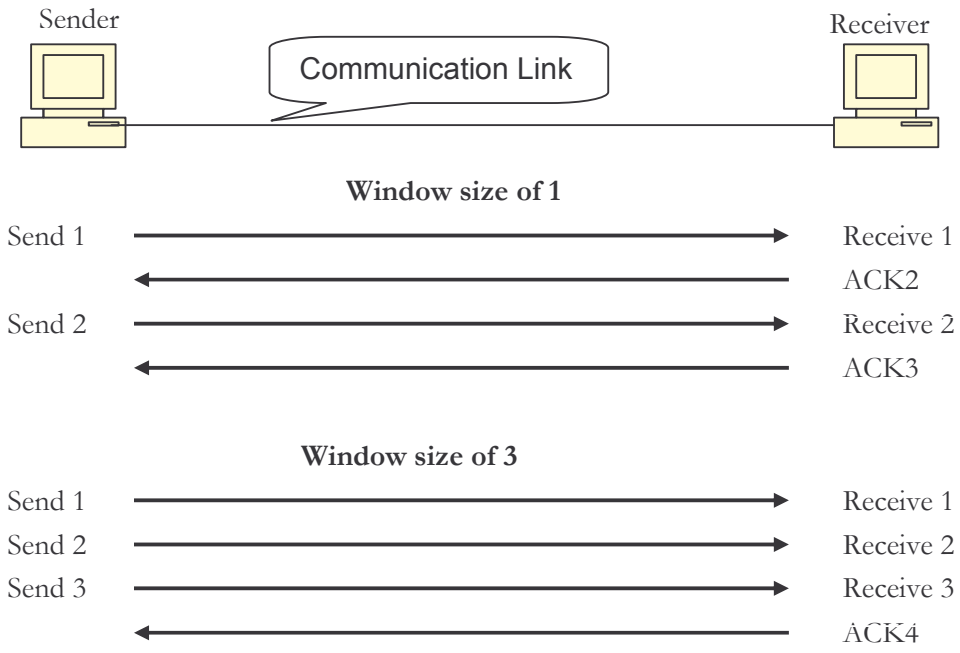


Figure 8.4 Windowing

As you can see in Figure, there are two window sizes—one set to 1, and one set to 3. When you've configured a window size of 1, the sending machine waits for an acknowledgment for each data segment it transmits before transmitting another. If you've configured a window size of 3, it's allowed to transmit three data segments before an acknowledgment is received. In our simplified example, both the sending and receiving machines are workstations.

8.5 EXAMPLE DATA LINK PROTOCOLS

WAN uses *serial transmission,* which takes place one bit at a time over a single channel. Parallel transmission can pass at least 8 bits at a time, but all WANs use serial transmission. Serial links are described in frequency or cycles-per-second (hertz). The amount of data that can be carried within these frequencies is called *bandwidth*. Bandwidth is the amount of data in bits-per-second that the serial channel can carry.

In the following sections we will examine several widely-used data link protocols used over WAN. The first one, HDLC, is a classical bit-oriented protocol. The second one, PPP, byte oriented protocol, is the data link protocol used to connect home or office computers to the Internet.

Byte-oriented protocol is a communications protocol in which data is transmitted as a series of bytes, or characters (for example, point-to-point protocol). Bit-oriented protocols interpret a transmission frame or packet as a succession of individual bits. (for example, SDLC, HDLC). Bit-oriented protocol is a communications protocol in which data is transmitted as a stream of bits rather than as a stream of bytes.

High-Level Data-Link Control (HDLC) Protocol

In this section we will examine a group of closely related protocols that are a bit old but are still heavily used. They are all derived from the data link protocol first used in the IBM mainframe world: SDLC (Synchronous Data Link Control) protocol. After developing SDLC, IBM submitted it to ANSI (American National Standard Institute) and ISO (International Standardization Organization) for acceptance as U.S. and international standards, respectively. ANSI modified it to become ADCCP (Advanced Data Communication Control Procedure), and ISO modified it to become HDLC (High-level Data Link Control). CCITT then adopted and modified HDLC for its LAP (Link Access Procedure) as part of the X.25 network interface standard but later modified it again to LAPB, to make it more compatible with a later version of HDLC.

The High-Level Data-Link Control (HDLC) protocol is a popular ISO-standard, bit-oriented Data Link layer protocol. It specifies an encapsulation method for data on synchronous serial data links using frame characters and checksums. HDLC is a point-to-point protocol used on leased lines. No authentication can be used with HDLC. In byte-oriented protocols, control information is encoded using entire bytes. On the other hand, bit-oriented protocols, may use single bits to represent control information. Bit-oriented protocols include SDLC, LLC, HDLC, TCP, IP, and others.

HDLC wasn't intended to encapsulate multiple Network layer protocols across the same link. The HDLC header carries no identification of the type of protocol being carried inside the HDLC encapsulation. Because of this, each vendor that uses HDLC has their own way of identifying the Network layer protocol, which means that each vendor's HDLC is proprietary for their equipment.

HDLC defines a method for encapsulating or formatting data into frames for synchronous transmission over synchronous serial WAN links to remote sites. HDLC is a bit-stream protocol (bit streams are not broken into individual characters) that uses a 32-bit checksum for error detection and supports full-duplex communication. HDLC frames consist of a flag byte followed by address and control information, data bits, and a CRC byte. A control field at the start of a frame is used for establishing and terminating data link connections.

Note: FCS is Frame Check Sequence uses CRC for error detection

Flag is 8-bit value, contains 01111110

Figure 8.5 HDLC frame format

An HDLC link consists of a primary station and a secondary station, with the primary station issuing the commands and the secondary station issuing the responses. Like another layer 2 WAN protocol called Point-to-Point Protocol (PPP), HDLC is used mainly for point-to-point communication, in contrast to other layer 2 WAN protocols such as Asynchronous Transfer Mode (ATM), frame relay, and X.25, which are used for both point-to-point and point-to-multipoint communication. Because HDLC is used mainly in point-to-point communication, it does not need to have addressing implemented at the data-link layer because the local and remote stations are connected directly. In this configuration, either one station is the primary and the other the secondary (unbalanced point-to-point link) or both stations function in a primary/secondary capacity (balanced point-to-point link).

The Point-to-Point (PPP) Protocol

PPP supports the transmission of network packets over a serial point-to-point link by specifying framing mechanisms for encapsulating network protocols such as Internet Protocol (IP), Internetwork Packet Exchange (IPX), or NetBEUI into PPP frames. PPP encapsulation is based on the High-level Data Link Control (HDLC) derived from the mainframe environment. These PPP frames can be transmitted over serial transmission lines such as Plain Old Telephone Service (POTS), Integrated Services Digital Network (ISDN), and packet-switched networks such as X.25. PPP includes an extensible Link Control Protocol (LCP) for establishing, tearing down, and testing data-link WAN connections, as well as a number of Network Control Protocols (NCPs) for establishing and configuring network communication using each network protocol. PPP also supports a number of authentication schemes, such as Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP).

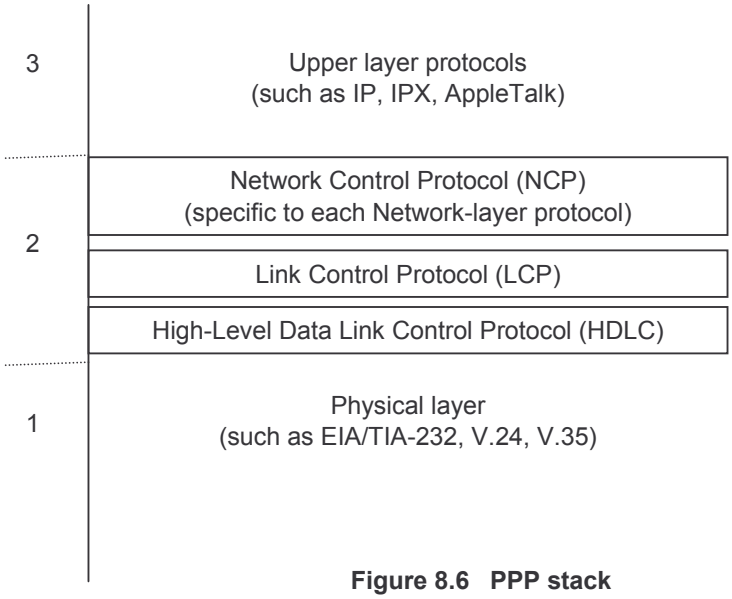


Figure 8.6 PPP stack

The PPP frame format was chosen to closely resemble the HDLC frame format, since there was no reason to reinvent the wheel. The major difference between PPP and HDLC is that PPP is character oriented rather than bit oriented.

Note: FCS is Frame Check Sequence uses CRC for error detection

Flag is 8-bit value, contains 01111110

Figure 8.7 PPP Frame format

The PPP frame contains the following fields:

- *Flag field*. Every PPP frame begins and ends with a 1 byte flag field with a value of 01111110.
- Address field. The only possible value for this field is 11111111.

- *Control Field.* The only possible value of this field is 00000011. Because both the address and control fields can currently take only a fixed value, one wonders why the fields are even defined in the first place. The PPP specification [RFC 1622] states that other values "may be defined at a later time," although none have been defined to date. Because these fields take fixed values, PPP allows the sender to simply not send the address and control bytes, thus saving two bytes of overhead in the PPP frame.
- *Protocol.* The protocol field tells the PPP receiver the upper layer protocol to which the received encapsulated data (i.e., the contents of the PPP frame's info field) belongs. On receipt of a PPP frame, the PPP receiver will check the frame for correctness and then pass the encapsulated data on to the appropriate protocol. [RFC 1700] defines the 16-bit protocol codes used by PPP. Of interest to us are the IP protocol (i.e., the data encapsulated in the PPP frame is an IP datagram) which has a value of 21 hexadecimal.
- *Information.* This field contains the encapsulate packet (data) that is being sent by an upper layer protocol (e.g., IP) over the PPP link. The default maximum length of the information field is 1500 bytes.
- *Checksum.* The checksum field is used to detect bit errors in a transmitted frame. It uses either a two or four byte HDLC-standard cyclic redundancy code.

Byte Stuffing

Let us consider a problem that arises when any protocol uses a specific bit pattern (flag field) to delineate the beginning or end of the frame: what happens if the flag pattern itself occurs elsewhere in the packet? For example, what happens if the flag field value of 01111110 appears in the information field? Will the receiver incorrectly detect the end of the PPP frame?

Solution is to use a technique known as **byte stuffing**.

PPP defines a special control escape byte, 01111101. If the flag sequence, 01111110 appears anywhere in the frame, except in the flag field, PPP precedes that instance of the flag pattern with the control escape byte. That is, it "stuffs" (adds) a control escape byte into the transmitted data stream, before the 01111110, to indicate that the following 011111110 is *not* a flag value but is, in fact, actual data. A receiver that sees a 01111110 preceded by a 01111101 will, of course, remove the stuffed control escape to reconstruct the original data. Similarly, if the control escape byte bit pattern itself appears as actual data, it too must be preceded by a stuffed control escape byte. Thus, when the receiver sees a single control escape byte by itself in the data stream, it knows that the byte was stuffed into the data stream. A pair of control escape bytes occurring back-to-back means that one instance of the control escape byte appears in the original data being sent.

PPP uses LCP to establish and maintain a PPP link over a serial transmission line. LCP frames are sent over the data link to test its integrity and establish the link.

PPP Authentication Methods

There are two methods of authentication that can be used with PPP links:

Password Authentication Protocol (PAP) The *Password Authentication Protocol (PAP)* is the less secure of the two methods. Passwords are sent in clear text, and PAP is only performed upon the initial link establishment. When the PPP link is first established, the remote node sends back to the originating router the username and password until authentication is acknowledged. That's it.

Challenge Handshake Authentication Protocol (CHAP) The *Challenge Handshake Authentication Protocol (CHAP)* is used at the initial startup of a link and at periodic checkups on the link to make sure the router is still communicating with the same host. After PPP finishes its initial link-establishment phase, the local router sends a challenge request to the remote device. The remote device sends a value calculated using a one-way hash function called MD5 (Message Digest version 5). The local router checks this hash value to make sure it matches. If the values don't match, the link is immediately terminated.

8.6 MAC ADDRESS

A unique 6-byte (48-bit) address that is usually permanently burned into a network interface card (NIC) or other physical-layer networking device and that uniquely identifies the device on an Ethernet-based network. A MAC address is also known as an Ethernet address, hardware address, or physical address.

MAC addresses can be hard-coded into circuitry or stored in read-only memory (ROM), and they can be configured using vendor-supplied software. The uniqueness of MAC addresses is ensured by the Institute of Electrical and Electronics Engineers (IEEE), which assigns networking device vendors specific blocks of MAC addresses for the devices they produce. The first 3 bytes (24 bits) represent the manufacturer of the card, and the last 3 bytes (24 bits) identify the particular card from that manufacturer. Each group of 3 bytes can be represented by 6 hexadecimal digits, forming a 12-digit hexadecimal number representing the entire MAC address. Examples of manufacturer 6-digit numbers include the following:

- 00:00:0C (Cisco)
- 00:14:A5 (BroadCom)
- 08:00:07 (Apple)
- 00:20:AF (3Com)

Each MAC address is 6 bytes long, is usually written in hexadecimal, and typically is written with colons separating each set of two hex digits. For example, 00:00:0C:12:34:56 is a valid Ethernet address. Computers use these addresses to identify the sender and receiver of an Ethernet frame. For instance, imagine that Machine "A" and Machine "B" are on the same Ethernet, and Machine "A" sends Machine "B" a frame. Machine "A" puts his own Ethernet MAC address in the Ethernet header as the source address and uses Machine "B"s Ethernet MAC address as the destination. When Machine "B" receives the frame, it notices that the destination address is his own address, so Machine "B" processes the frame. If Machine "B" receives a frame with some other device's address in the destination address field, Machine "B" simply does not process the frame.

The IEEE requires globally unique MAC addresses on all network interface cards. To ensure a unique MAC address, the Ethernet card manufacturers encode the MAC address onto the card, usually in a ROM chip. The first half of the address identifies the manufacturer of the card. This code, which is assigned to each manufacturer by the IEEE, is called the *organizationally unique identifier (OUI)*. Each manufacturer assigns a MAC address with its own OUI as the first half of the address, with the second half of the address being assigned a number that this manufacturer has never used on another card.

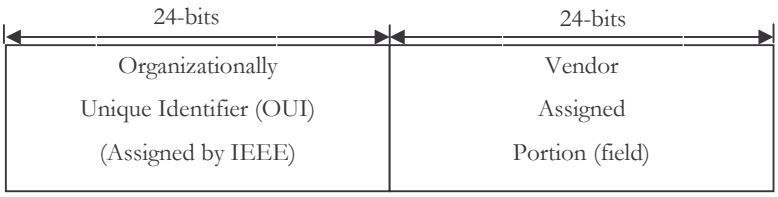


Figure 8.8 MAC address separation

The most often used of IEEE group MAC addresses, the broadcast address, has a value of FF:FF:FF:FF:FF:FF (hexadecimal notation). The broadcast address implies that all devices on the network (same network segment) should process the frame.

SUMMARY

The task of the data link layer is to convert the raw bit stream offered by the physical layer into a stream of frames for use by the network layer. Data link protocols can provide error control to retransmit damaged or lost frames. To prevent a fast sender from overrunning a slow receiver, the data link protocol can also provide flow control. The sliding window mechanism is widely used to integrate error control and flow control in a convenient way.

PRACTICE SET

Review Questions

1. Explain HDLC protocol
2. Explain PPP protocol
3. What do you mean by piggybacking?
4. What is byte stuffing?
5. What are sub-layers of DLL?
6. What are elementary protocols of DLL?
7. What are differences between HDLC and PPP?
8. Why PPP prefer over HDLC?

Multiple Choice Questions

1. Which of the following are valid PPP authentication methods? (Choose two.)
A) LCP B) NCP C) CHAP D) MD5

2. Media Access Control (MAC) is sublayer of …….
A) Physical B) Data Link C) Network D) Application

3. Logical Link Control (LLC) is sublayer of …….
A) Physical B) Data Link C) Network D) Application

4. MAC stands for …….

A) Money Access Control B) More Advanced Control

C) Media Access Control D) None of above

5. LLC stands for ……..

A) Logical Level control B) Low level control

C) Logical Link Control D) None of the above

6 PPP stands for……..

A) Poor password protection B) Point to point protocol

C) Both option A) and B) D) None of the above

7 HDLC stands for…….

A) Heavy Duty Life Cycle B) Heavy Duty Logic Control

C) High-level Data Link Control D) High-level Data Life control

8. PAP stands for……..

A) Password Authentication Protocol B) Password Access Protocol

C) Poor Authentication protocol D) None of the above

9. CHAP stands for…….

A) Challenge Handshake Access Protocol

B) Challenge Heavy Authentication Protocol

C) Challenge Handshake Authentication Protocol

D) None of the above

10. NCP in PPP stack stands for…….

A) Network Control Protocol B) Network Connect protocol

C) New control Protocol D) New connect protocol

11. LCP in PPP stack stands for…….

A) Link Control Protocol B) Link Connect protocol

C). Level control Protocol D) Level connect protocol

12. IP stands for……..

A) Information Protocol B) Internet Protocol

C) Indian Protocol D) None of the above

13. FCS stands for……

A) Full control sequence B) Full connection set

C) Frame check sequence D) Frame control set

14. Which of the following are true about the format of Ethernet addresses?
A) Each manufacturer puts a unique code into the first 2 bytes of the address.

B) The part of the address that holds this manufacturer's code is called the MC.

C) The part of the address that holds this manufacturer's code is called the OUI.

D) The part of the address that holds this manufacturer's code has no specific name.

15. Who assigns OUI code to manufacture of NIC?
A) IBM B) Cisco C) IEEE D) None of the above

16. OUI stands for……..
A) One Unique Identifier B) Organizationally Unique Identifier

C) One Unique Information D) None of the above

17. Which one out of the following address is called as broadcast MAC address?
A) 00:12:34:56:78:90 B) 0A:90:45:6D:2C:3F

C) FF:FF:FF:FF:FF:FF D) 0A:CF:4E:45:CA:08

18. Which one out of the following is valid MAC address?
A) 00:12:34:5H:78:90 B) 0A:90:4K:6D:2C:3F

C) 0A:CH:4E:45:CA:08 D) 0A:C7:4E:45:CA:08

* * *

Chapter 9 NETWORK LAYER

A computer network consists of computers and communications resources and users. The resources are there to be shared, and therefore coordination is required among the users for their efficient and fair use. This coordination is obtained via network protocols, which govern the access to resources, the establishment of connections and the exchange of data through the network.

The OSI model assigns the functions of path selection and logical addressing to the OSI network layer (Layer 3). Path selection includes the process of learning all the paths, or routes, in a network and then forwarding packets based on those paths or routes. In this chapter, you will learn about the core concepts behind OSI Layer 3.

OBJECTIVES OF THIS CHAPTER

After completing this chapter, the student should be able to:

- define duties of network layer
- explain network service model
- explain logical addressing
- define network layer protocols

9.1 INTRODUCTION

A protocol that defines routing and addressing is considered to be a network layer, or Layer 3, protocol. OSI does define a unique Layer 3 protocol called Connectionless Network Services (CLNS), but, as usual with OSI protocols, you rarely see it in networks today. However, you will see many other protocols that perform the OSI Layer 3 functions of routing and addressing, such as the Internet Protocol (IP), Novell Internetwork Packet Exchange (IPX), or AppleTalk Dynamic Data Routing (DDR).

The role of the network layer in a sending host is to begin the packet on its journey to

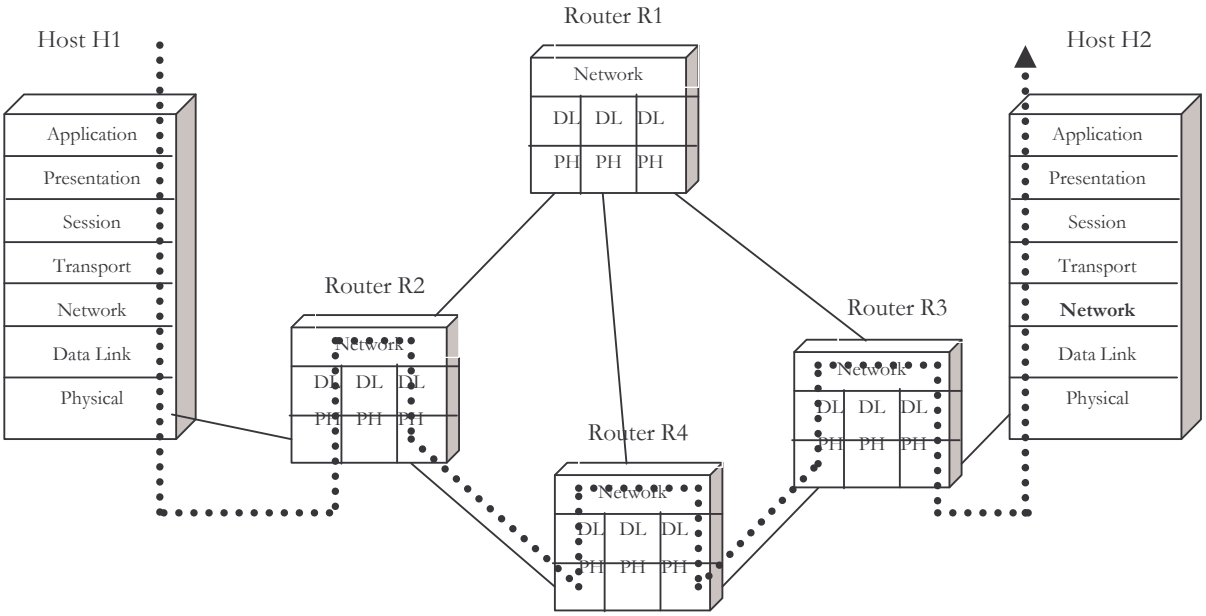


Figure 9.1 Network Layer

the receiving host. For example, if H1 is sending to H2, the network layer in host H1 transfers these packets to it nearby router, R2. At the receiving host (e.g., H2), the network layer receives the packet from its nearby router (in this case, R3) and delivers the packet up to the transport layer at H2. The primary role of the routers is to "switch" packets from input links to output links.

The role of the network layer is simple -- to transport packets from a sending host to a receiving host. To do so, three important network layer functions can be identified:

- **Path Determination.** The network layer must determine the route or path taken by packets as they flow from a sender to a receiver. The algorithms that calculate these paths are referred to as *routing algorithms*. A routing algorithm would determine, for example, whether packets from H1 to H2 flow along the path R2-R1-R3 or path R2-R4-R3 (or any other path between H1 and H2).
- **Switching.** When a packet arrives at the input to a router, the router must move it to the appropriate output link. For example, a packet arriving from host H1 to router R2 must either be forwarded towards H2 either along the link from R2 to R1 or along the link from R2 to R4.
- **Logical Addressing.** The physical addressing is implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, then there is a need of another addressing system to help to distinguish the source and destination systems.

The network layer also supplies connectionless and connection-oriented services to the transport layer above it. The network layer functions closely with the physical layer (layer 1) and data-link layer (layer 2) in most real-world network protocol implementations.

Information data (often in the forms of packets) to be transferred from a source to a destination generally pass through many intermediate nodes, which are interconnected to each other. The goal of the routing protocol is to select the best path from the collection of paths between source and destination, given the traffic requirements and the network configuration. *Best path* usually means path of minimum average delay through the network or path of minimum cost.

9.2 NETWORK SERVICE MODEL

When the transport layer at a sending host transmits a data down to the network layer at the sending host, can the transport layer count on the network layer to deliver the packet to the destination? When multiple packets are sent, will they be delivered to the transport layer in the receiving host in the order in which they were sent? Will the network provide any feedback about congestion in the network? The answers to these questions and others are determined by the *service model* provided by the network layer. The network service model defines the characteristics of end-to-end transport of data between sending and receiving end systems.

Networks may provide connection oriented (e.g., virtual circuit) or connectionless (datagram) service. For example, the Internet offers connectionless network-layer service; ATM networks offer connection-oriented network-layer service. ATM uses virtual circuits.

Datagram or Virtual Circuits?

The most important abstraction provided by the network layer to the upper layers is whether or not the network layer uses virtual circuits (VCs) or datagram. There are three phases in a virtual circuit:

- **Virtual Circuit (VC) setup.** During the setup phase, the sender contacts the network layer, specifies the receiver address, and waits for the network to setup the VC. The network layer determines the path between sender and receiver, i.e., the series of links and switches through which all packets of the VC will travel. This typically involves updating tables in each of the packet switches in the path. During VC setup, the network layer may also reserve resources (e.g., bandwidth) along the path of the VC.
- **Data transfer.** Once the VC has been established, data can begin to flow along the VC.
- **Virtual circuit termination.** This is initiated when the sender (or receiver) informs the network layer of its desire to terminate the VC. The network layer will then typically inform the end system on the other side of the network of the call termination, and update the tables in each of the packet switches on the path to indicate that the VC no longer exists.

The messages that the end systems send to the network to indicate the initiation or termination of a VC, and the messages passed between the switches to set up the VC (i.e. to modify switch tables) are known as *signaling messages* and the protocols used to exchange these messages are often referred to as *signaling protocols*.

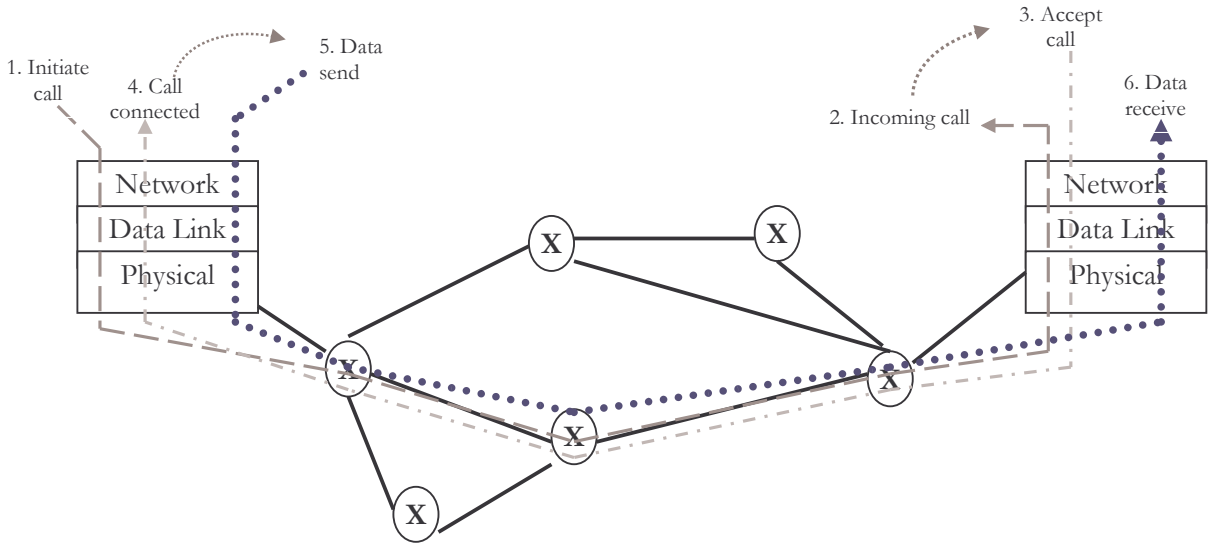


Figure 9.2 Virtual circuit

With a **datagram network layer**, each time an end system wants to send a packet; it stamps the packet with the address of the destination end system, and then injects the packet into the network. This is done without any VC setup. Packet switches (called "routers" in the Internet) do not maintain any state information about VCs because there are no VCs! Instead, packet switches route a packet towards its destination by examining

the packet's destination address, indexing a routing table with the destination address, and forwarding the packet in the direction of the destination. Because routing tables can be modified at any time, a series of packets sent from one end system to another may follow different paths through the network. An alternative terminology for VC service and datagram service is **network-layer connection-oriented service** and **network-layer connectionless service,** respectively. Indeed, the VC service is a sort of connection-oriented service, as it involves setting up and terminating a connection-like entity, and maintaining connection state information in the packet switches. The datagram service is a sort of connectionless service in that it doesn't employ connection-like entities.

Figure 9.3 Datagram service

9.3 LOGICAL ADDRESSING

One of the main features of network layer addresses is that they were designed to allow logical grouping of addresses. That is, something about the numeric value of an address implies a group or set of addresses, all of which are considered to be in the same grouping. In TCP/IP, this group is called a *network* or a *subnet*. In IPX, it is called a *network*. In AppleTalk, the grouping is called a *cable range*.

Routing relies on the fact that Layer 3 addresses are grouped together. The routing tables for each network layer protocol can have one entry for the group, not one entry for each individual address. Imagine an Ethernet with 1000 TCP/IP hosts. A router needing to forward packets to any of those hosts needs only one entry in its IP routing table. This basic fact is one of the key reasons that routers can scale to allow tens and hundreds of thousands of networks. It's very similar to the postal PIN code system—it would be ridiculous to have people in the same PIN code live somewhere far away from each other, or to have next-door neighbors be in different PIN codes. The poor postman would spend all his time driving and flying around the country! Similarly, to make routing more efficient, network layer protocols group addresses together.

Network layer (Layer 3) addressing schemes were created with the following goals:

- The address space should be large enough to accommodate the largest network.
- The addresses should allow for unique assignment. That is each host on network must have unique address.

- The address structure should have some grouping implied so that many addresses are considered to be in the same group.
- Dynamic address assignment for clients is possible.

Protocol	Size of Address in Bits	Name and Size of Grouping (Network) Field in Bits	Name and Size of Local (Host) Address Field in Bits
IP	32	Network or subnet (variable, between 8 and 30 bits)	Host (variable, between 2 and 24 bits)
IPX	80	Network (32)	Node (48)
AppleTalk	24	Network (16)	Node (8)

Table 9.1 Examples of Layer3 Address Structure

There are two different versions of TCP/IP IP addressing: IPv4 and IPv6.

IPv4 (Internet Protocol version 4) addresses are 32 bits in length. However, to make the addresses readable, they are broken into four bytes (called octets), with a period (decimal) between each byte. So that the address is understandable to the human eye, the four sets of binary numbers are then converted to decimal. Let's look at a simple example: 11111111000000001111111100000001, which are 32 1's and 0's. This is broken up into four octets, like this: 11111111.11111111.11111111.00000001. Then each of these octets are converted into decimal, resulting in 255.0.255.1. The format of this address is commonly called *dotted decimal notation*

Computers and networking devices process everything in binary. In a byte (octet), there are eight bits. Each bit, when enabled, represents a specific decimal value.

Bit position	8	7	6	5	4	3	2	1
Decimal value	128	64	32	16	8	4	2	1

Table 9.2 Binary to Decimal

A bit positions are labeled from left-to-right, where the left-most bit is the most significant and the right-most bit is the least-significant. A bit can contain one of two values: 0 or 1. If it is enabled (set to 1), then that equates to a particular decimal value, shown in the second row of Table 9.2. If it is disabled (set to 0), then this equates to a decimal value of zero.

To convert the binary byte value to a decimal value, you look at all the bits that are turned on and add up the equivalent decimal values. For example, assume that you had a byte with a value of 10000011. Bits 8, 2, and 1 are on, so add up the associated decimal values to get the corresponding decimal equivalent of the byte value: 128 + 2 + 1 = 131. If you had a byte value of 10101010, the decimal value would be: 128 + 32 + 8 + 2 = 170. If all the bit positions where set to 0, then the decimal value would be 0. If all the bit positions were set to 1, the equivalent decimal value would be: 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255. Given this, a byte value can range from 0 to 255.

IP, or layer-3, addresses have two components: a network and host number. The network number uniquely identifies a segment or subnet in the network and a host

number uniquely identifies a host device on a segment or subnet. The combination of these two numbers must be unique throughout the entire network.

IP addresses are breakup into five different classes namely: class A, class B, class C, class D, and class E. With a Class A address, the first byte is a network number (8 bits) and the last 3 bytes are for host numbers (24 bits). With a Class B address, the first two bytes are a network number (16 bits) and the last 2 bytes are for host numbers (16 bits). With a Class C address, the first three bytes are a network number (24 bits) and the last 1 byte is for host numbers (8 bits). Class D addresses are used for multicasting and Class E addresses are reserved.

Class	8bit	8bit	8bit	8bit
A	N	H	H	H
B	N	N	H	H
C	N	N	N	H
D	Multicast			
E	Reserved			

Note: N means Network address bits
H means Host address bits

Figure 9.4 Classes of IP addresses

Class A addresses always begin with a "0" in the highest order bit. Class B addresses always begin with "10" in the highest order bits. Class C addresses always begin with "110" in the highest order bits. Class D addresses always begin with "1110" in the highest order bits. Class E addresses always begin with "11110" in the highest order bits.

Class	8bit	8bit	8bit	8bit
A	0xxxxxxx	H	H	H
B	10xxxxxx	N	H	H
C	110xxxxx	N	N	H
D	1110xxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx
E	11110xxx	xxxxxxxx	xxxxxxxx	xxxxxxxx

Figure 9.5 Distinguishing between classes of addresses

Given the above distinctions with the assigned high-order bit values, it is easy to predict, for a given address, what class of network numbers it belongs to: Class A addresses range from 1-126: 0 is reserved and represents all IP addresses; 127 is a reserved address and is used for testing, like a loopback on an interface: 00000001-01111111. Class B addresses range from 128-191: 10000000-10111111. Class C addresses range from 192-223: 11000000-11011111. Class D addresses range from 224-239: 11100000-11101111. Class E addresses range from 240-254: 255 is a reserved address and is used for broadcasting purposes. Now it is easy to predict what address belongs to what class.

A class-A address has the most significant bit 0. The next seven bits contain the network number and the last 24 bits the host number. There are thus 126 possible class-A networks, each having up to about 16,000,000 hosts. (Networks 0 and 127 are reserved.)

A class-B address has the two most significant bits 10. The next fourteen bits contain the network number and the last 16 bits the host number. There are thus 16384 possible class-B networks each containing 65354 hosts.

A class C address has the three most significant bits 110. The next 21 bits contain the network number and the last eight bits the host number. There are thus more than 2000000 possible class C networks each containing 254 hosts

Class	Range
A	0-127
B	128-191
C	192-223
D	224-239
E	240-255

Figure 9.6 IP address class range

Within this range of addresses for Class A, B, and C addresses, there are some reserved addresses, commonly called *Private Addresses*. All the other addresses in these classes are called public addresses. Anyone can use private addresses; however, this creates a problem if you want to access the Internet. Packet with private IP address in its header will never be injected on Internet (line connected to your ISP (Internet Service Provider, such as BSNL or Reliance)) by routers. In order to access the Internet, your source IP addresses must have a unique Internet public address.

Class	Private Address Range	Description
A	10.0.0.0 - 10.255.255.255	1 class A network
B	172.16.0.0-172.31.255.255	16 class B networks
C	192.168.0.0-192.168.255.255	256 class C networks

Figure 9.7 Private IP address range

When you are dealing with IP addresses, there are always two addresses reserved for a given network number: the first address in the network represents the network's address, and the last address in the network represents the broadcast address for this network, commonly called a *directed broadcast*. In IP itself, there are two IP addresses reserved: 0.0.0.0 (the very first address), which represents all IP addresses, and 255.255.255.255 (the very last address), which is the local broadcast address (all devices should process this datagram).

When dealing with a network address, all of the host bits in the host portion of the address are set to zeros. If all of the host bits in a network number are set to ones, making it the very last address, then this is the directed broadcast address. Any combination of bit values between these two numbers in the *host* portion of the address is considered a host address. For example 10.0.0.0 is a class A address and is also a network address. Therefore 10 is a network number and last 3 bytes are use for host address. Setting zero to host address byte gives network address, therefore network address is 10.0.0.0. If you would set all host bits to one it will give direct broadcast address, which is in our example 10.255.255.255.

Any number between the network address and the directed broadcast address is a host address. For example 192.168.1.0 is a class C address, which uses last byte to represent host address. So any number between 0 and 255 is a host address for the network 192.168.1.0: 192.168.1.1 -to- 192.168.1.254.

Class	Size of Network part in bits	Size of Host part in bits	Default Mask
A	8	24	255.0.0.0
B	16	16	255.255.0.0

C	24	8	255.255.255.0

Table 9.3 Default Mask

The net mask is a 32-bit binary number, usually written in dotted-decimal format. The purpose of the net mask is to define the structure of an IP address. In short, the net mask defines the size of the host parts of an IP address, representing the host part of the IP address with binary 0s in the net mask. For example, Class A net mask has its last 24 bits as binary 0, which means that the last three octets of the net mask are 0s.

One restriction of net masks is that all the network bits (1s) must be contiguous and all the host bits (0s) are contiguous. This is true not only in a single octet, but across all the bits in all four octets. A net mask of 11111111.00001111.11111111.00000000 (255.31.255.0) would be invalid since all the 1s are not contiguous. A net mask of 1111111.11111111.11111111.00000000 (255.255.255.0), however, is valid

9.4 NETWORK OR INTERNET LAYER PROTOCOL

The following sections describe the protocols at the Internet layer:

- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- Address Resolution Protocol (ARP)
- Reverse Address Resolution Protocol (RARP)

INTERNET PROTOCOL (IP)

IP is a TCP/IP network layer protocol for addressing and routing packets of data between hosts on a TCP/IP network. Internet Protocol (IP) is a connectionless protocol that provides best-effort delivery using packet-switching services.

Identifying devices on networks requires answering these two questions: Which network is it on? And what is its ID on that network? The first answer is the *software address*, or *logical address*. The second answer is the hardware address or MAC address. All hosts on a network have a logical ID called an IP address.

IP does not guarantee delivery of data. The responsibility for guaranteeing delivery and sending acknowledgments lies with the higher transport-level protocol Transmission Control Protocol (TCP). IP in the network layer adds IP header to data it receives from transport layer. Each router (layer 3 device) that receives a datagram makes routing decisions based on the packet's destination IP address present in packet header attached by source machine's network layer.

The structure of an IP packet is shown in the following diagram.

Version: IP version number, such as IPv4 or IPv6. This 4-bit field contains the version number of IP to which this packet conforms. This field should currently contain the value 4, although IP version 6 is currently being defined.

Header Length (HLEN): Header length in 32-bit words. This 4-bit field contains the length of the header in 32-bit words. If there are no options, the value of this field will be 5 (giving a header length of 20 bytes).

ToS with IP Precedence Bits: Type of Service tells how the datagram should be handled. The first 3 bits are the priority bits. This field gives information about the

quality of service requested for this packet. It contains subfields which indicate the type of packet and its urgency.

Total length: Length of the packet including header and data.

Identifier: Unique IP-packet value.

Flags: Specifies whether fragmentation should occur. This 3-bit field contains three flags, only two of which are currently defined. One flag is used to indicate that this packet cannot be fragmented. This might be used when the destination node is known to be unable to reassemble fragmented packets. The other flag is used to indicate that this is part of a packet which has been fragmented by the system. This bit is clear if this is the last fragment of a larger packet.

Fragment offset: Provides fragmentation and reassembly if the packet is too large to put in a frame. It also allows different maximum transmission units (MTUs) on the Internet. The identification field is used in conjunction with the source and destination address fields to ensure that each packet is uniquely identified. This field can be used to reassemble packets which have been fragmented because they are too long for one of the links. It is measured in multiples of 8 bytes.

TTL: The time to live is set into a packet when it is originally generated. If it doesn't get to where it wants to go before the TTL expires, packet gets discarded from network that is it vanishes from network. This stops IP packets from continuously circling the network looking for a home.

Protocol: Port of upper-layer protocol (TCP is port 6 or UDP is port 17 [hex]). Also supports Network layer protocols.

Header checksum: Cyclic redundancy check (CRC) on header only. This checksum is used to ensure that the header has been transmitted correctly.

Source IP address: 32-bit IP address of sending station or source machine. This field contains the IP address of the originating host for this packet. This does not necessarily correspond to the address of the node which sent this packet within the network but is the address of the host which first put this packet into the network. It thus differs from the data link layer address.

Destination IP address: 32-bit IP address of the station this packet is destined for. This is the address of the host for which this packet is ultimately destined. It is this address which is used to determine the path this packet will take through the network. IP is a datagram oriented (connectionless) protocol.

IP option: Used for network testing, debugging, security, and more.

Data: After the IP option field will be the upper-layer data.

Bit 0 Bit 15 Bit 16 Bit 31

<table>
<tr><td>Version
(4-bit)</td><td>Header
Length
(4-bit)</td><td>Priority and
Type Of service
(8-bit)</td><td colspan="2">Total Length (16-bit)</td></tr>
<tr><td colspan="3">Identification (16-bit)</td><td>Flags
(3-bit)</td><td>Fragment offset (13-bit)</td></tr>
<tr><td>Time to Live
(8-bit)</td><td colspan="2">Protocol
(8-bit)</td><td colspan="2">Header Checksum
(16-bit)</td></tr>
</table>

Source IP address (32-bit)
Destination IP address (32-bit)
Options (0 or 32 if any)
Data or Payload (varies if any)

Figure 9.8 IP Header

IP packets are routed in the following fashion:

- If IP determines that the destination IP address is a local address, it transmits the packet directly to the destination host.
- If IP determines that the destination IP address is a remote address, it examines the local routing table for a route to the destination host. If a route is found, it is used; if no route is found, IP forwards the packet to the default gateway. In either case, the packet destined for a remote address is usually sent to a router.
- At the router, the TTL is decreased by 1 or more (depending on network congestion), and the packet might be fragmented into smaller packets if necessary. The router then determines whether to forward the packet to one of the router's local network interfaces or to another router. This process repeats until the packet arrives at the destination host or has its TTL decremented to 0 (zero) and is discarded by a router.

Figure 9.10 shows how the Network layer sees the protocols at the Transport layer when it needs to hand a packet to the upper-layer protocols.

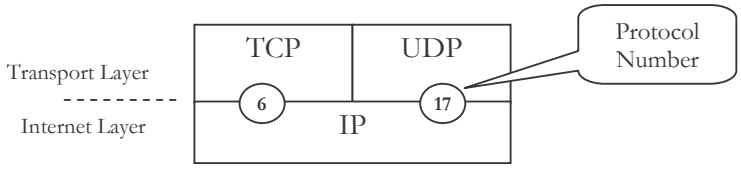


Figure 9.10 Protocol field in IP header

The Protocol field tells IP to send the data to either TCP port 6 or UDP port 17 (both hex addresses). But it will only be UDP or TCP.

PROTOCOL Field

The Internet Protocol version 4 (IPv4) (refer RFC791) there is a field, called "Protocol", to identify the next level protocol. This is an 8 bit field. In Internet Protocol version 6 (IPv6) (refer RFC1883 and RFC2460) this field is called the "Next Header" field. Protocol field in IP header is an eight bit (8-bit) field means maximum protocol number can be 256 ranging from 0 to 255.

Protocol Number	Keyword	Protocol
0	HOPOPT	IPv6 Hop-by-Hop Option

Protocol Number	Keyword	Protocol
1	ICMP	Internet Control Message Protocol
2	IGMP	Internet Group Management Protocol
3	GGP	Gateway-to-Gateway Protocol
4	IP	IP in IP (encapsulation)
5	ST	Internet Stream Protocol
6	TCP	Transmission Control Protocol
7	CBT	CBT
8	EGP	Exterior Gateway Protocol
9	IGP	Interior Gateway Protocol (any private interior gateway (used by Cisco for their IGRP))
10	BBN-RCC-MON	BBN RCC Monitoring
11	NVP-II	Network Voice Protocol
12	PUP	Xerox PUP
13	ARGUS	ARGUS
14	EMCON	EMCON
15	XNET	Cross Net Debugger
16	CHAOS	Chaos
17	UDP	User Datagram Protocol
18	MUX	Multiplexing
19	DCN-MEAS	DCN Measurement Subsystems
20	HMP	Host Monitoring Protocol
21	PRM	Packet Radio Measurement
22	XNS-IDP	XEROX NS IDP
23	TRUNK-1	Trunk-1
24	TRUNK-2	Trunk-2
25	LEAF-1	Leaf-1
26	LEAF-2	Leaf-2
27	RDP	Reliable Data Protocol
28	IRTP	Internet Reliable Transaction Protocol
29	ISO-TP4	ISO Transport Protocol Class 4
30	NETBLT	Bulk Data Transfer Protocol
31	MFE-NSP	MFE Network Services Protocol
32	MERIT-INP	MERIT Internodal Protocol
33	SEP	Sequential Exchange Protocol
34	3PC	Third Party Connect Protocol
35	IDPR	Inter-Domain Policy Routing Protocol
36	XTP	XTP

Protocol Number	Keyword	Protocol
37	DDP	Datagram Delivery Protocol
38	IDPR-CMTP	IDPR Control Message Transport Protocol
39	TP++	TP++ Transport Protocol
40	IL	IL Transport Protocol
41	IPv6	IPv6
42	SDRP	Source Demand Routing Protocol
43	IPv6-Route	Routing Header for IPv6
44	IPv6-Frag	Fragment Header for IPv6
45	IDRP	Inter-Domain Routing Protocol
46	RSVP	Resource Reservation Protocol
47	GRE	Generic Routing Encapsulation
48	MHRP	Mobile Host Routing Protocol
49	BNA	BNA
50	ESP	Encapsulating Security Payload
51	AH	Authentication Header
52	I-NLSP	Integrated Net Layer Security Protocol
53	SWIPE	IP with Encryption
54	NARP	NBMA Address Resolution Protocol
55	MOBILE	IP Mobility
56	TLSP	Transport Layer Security Protocol (using Kryptonet key management)
57	SKIP	SKIP
58	IPv6-ICMP	ICMP for IPv6
59	IPv6-NoNxt	No Next Header for IPv6
60	IPv6-Opts	Destination Options for IPv6
61		Any host internal protocol
62	CFTP	CFTP
63		Any local network
64	SAT-EXPAK	SATNET and Backroom EXPAK
65	KRYPTOLAN	Kryptolan
66	RVD	MIT Remote Virtual Disk Protocol
67	IPPC	Internet Pluribus Packet Core
68		Any distributed file system
69	SAT-MON	SATNET Monitoring
70	VISA	VISA Protocol
71	IPCV	Internet Packet Core Utility
72	CPNX	Computer Protocol Network Executive
73	CPHB	Computer Protocol Heart Beat
74	WSN	Wang Span Network

Protocol Number	Keyword	Protocol
75	PVP	Packet Video Protocol
76	BR-SAT-MON	Backroom SATNET Monitoring
77	SUN-ND	SUN ND PROTOCOL-Temporary
78	WB-MON	WIDEBAND Monitoring
79	WB-EXPAK	WIDEBAND EXPAK
80	ISO-IP	International Organisation for Standardization Internet Protocol
81	VMTP	VMTP
82	SECURE-VMTP	SECURE-VMTP
83	VINES	VINES
84	TTP	TTP
85	NSFNET-IGP	NSFNET-IGP
86	DGP	Dissimilar Gateway Protocol
87	TCF	TCF
88	EIGRP	EIGRP
89	OSPF	Open Shortest Path First
90	Sprite-RPC	Sprite RPC Protocol
91	LARP	Locus Address Resolution Protocol
92	MTP	Multicast Transport Protocol
93	AX.25	AX.25
94	IPIP	IP-within-IP Encapsulation Protocol
95	MICP	Mobile Internetworking Control Protocol
96	SCC-SP	Semaphore Communications Sec. Pro
97	ETHERIP	Ethernet-within-IP Encapsulation
98	ENCAP	Encapsulation Header
99		Any private encryption scheme
100	GMTP	GMTP
101	IFMP	Ipsilon Flow Management Protocol
102	PNNI	PNNI over IP
103	PIM	Protocol Independent Multicast
104	ARIS	ARIS
105	SCPS	SCPS
106	QNX	QNX
107	A/N	Active Networks
108	IPComp	IP Payload Compression Protocol
109	SNP	Sitara Networks Protocol
110	Compaq-Peer	Compaq Peer Protocol
111	IPX-in-IP	IPX in IP

Protocol Number	Keyword	Protocol
112	VRRP	Virtual Router Redundancy Protocol, Common Address Redundancy Protocol (not IANA assigned)
113	PGM	PGM Reliable Transport Protocol
114		Any 0-hop protocol
115	L2TP	Layer Two Tunneling Protocol
116	DDX	D-II Data Exchange (DDX)
117	IATP	Interactive Agent Transfer Protocol
118	STP	Schedule Transfer Protocol
119	SRP	SpectraLink Radio Protocol
120	UTI	UTI
121	SMP	Simple Message Protocol
122	SM	SM
123	PTP	Performance Transparency Protocol
124		IS-IS over IPv4
125	FIRE	
126	CRTP	Combat Radio Transport Protocol
127	CRUDP	Combat Radio User Datagram
128	SSCOPMCE	
129	IPLT	
130	SPS	Secure Packet Shield
131	PIPE	Private IP Encapsulation within IP
132	SCTP	Stream Control Transmission Protocol
133	FC	Fibre Channel
134		RSVP-E2E-IGNORE
135		Mobility Header
136		UDPLite
137		MPLS-in-IP
138-252	Unassigned	
253-254	Use for experimentation and testing	
255	Reserved	

Table 9.4 Protocol field

INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

Internet Control Message Protocol (ICMP) works at the Network layer. ICMP is a management protocol and messaging service provider for IP. ICMP uses connectionless Internet Protocol (IP) datagram of various types for communicating control messages between hosts and routers on a TCP/IP network. The more common ICMP packets include the following:

- Destination Unreachable (ICMP type 3): Indicates that the destination network, host, or port cannot be reached.
- Echo Request (ICMP type 8): The ping command uses this packet type to test TCP/IP connectivity.
- Echo Reply (ICMP type 0): The ping command uses this packet type to test TCP/IP connectivity.

ADDRESS RESOLUTION PROTOCOL (ARP)

Address Resolution Protocol (ARP) finds the hardware address of a host from a known IP address. Address Resolution Protocol (ARP) is defined in Request for Comments (RFC-826).

When a TCP/IP-aware application tries to access another host using its IP address, the destination host's IP address must first be resolved into a MAC address so that the frame can be addressed and placed on the wire and then be recognized by the destination host's network interface card (NIC). Because NIC works on Layer 2, it can only understand Layer 2 address (that is, MAC address), and takes decision whether to accept frame or to discard frame based on destination MAC address in header field of the frame. If the destination MAC address in the header field of the arrived frame matches with its own MAC address then NIC accepts the frame for further processing otherwise rejects the frame.

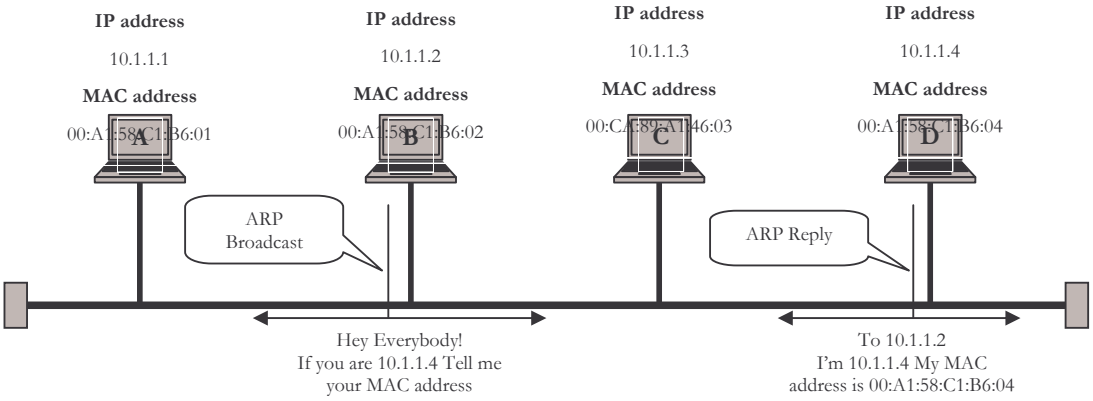


Figure 9.11 ARP

As shown in figure 9.11. Machine with IP address 10.1.1.2 needs to know the Ethernet MAC address used by 10.1.1.4, so 10.1.1.2 issues something called an *ARP broadcast*. An ARP broadcast is sent to a broadcast Ethernet address, so everyone on the LAN receives it. Because 10.1.1.4 is on the LAN, 10.1.1.4 receives the ARP broadcast, and the ARP broadcast is looking for the MAC address associated with 10.1.1.4, so 10.1.1.4 replies with its own MAC address. Remember, when 10.1.1.2 send ARP request, it was ARP broadcast packet (which contained source IP address as 10.1.1.2, destination IP address as 10.1.1.4, source MAC address as 00:A1:58:C1:B6:02, and destination MAC address as FF:FF:FF:FF:FF:FF), but when 10.1.1.4 send ARP reply to 10.1.1.2 (ARP reply packet contained source IP address as 10.1.1.4, destination IP address as 10.1.1.2, source MAC address as 00:A1:58:C1:B6:04, and destination MAC address as 00:A1:58:C1:B6:02).

The requesting host after receiving ARP reply, temporarily stores the IP-to-MAC-address mapping in its local ARP cache in case this is required again within a short interval of

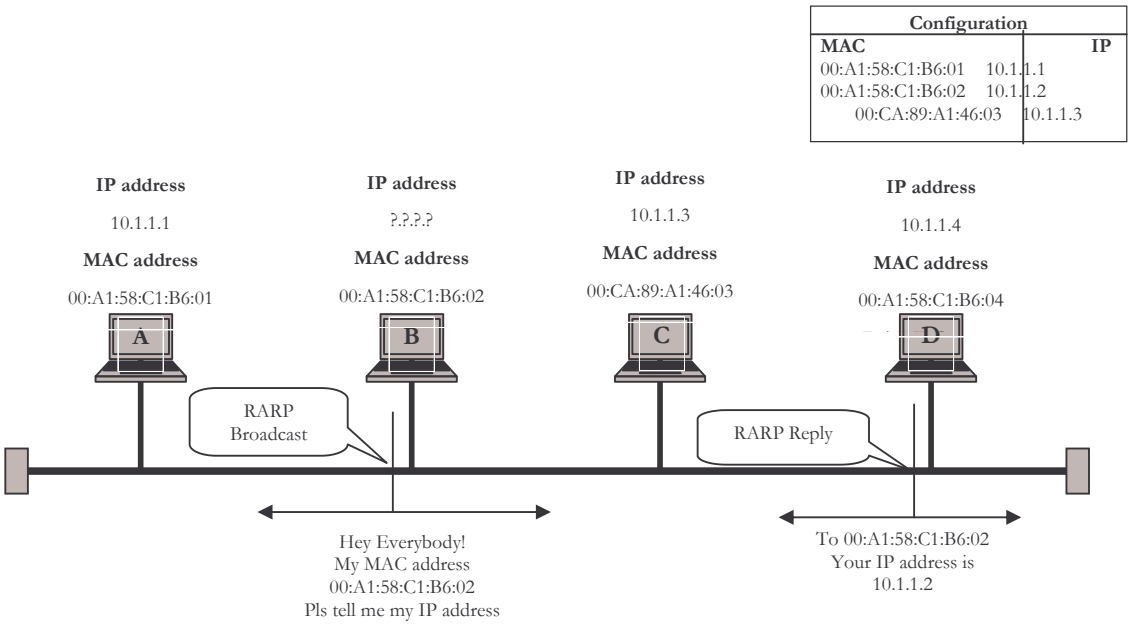


Figure 9.12 RARP

time. If the destination host is on a remote network, ARP obtains the MAC address of the local router interface that connects the local network to the remote network.

REVERSE ADDRESS RESOLUTION PROTOCOL (RARP)

When a TCP/IP machine happens to be a diskless machine, it has no way of initially knowing its IP address. But it does know its MAC address. *Reverse Address Resolution Protocol (RARP)* resolves MAC address into the IP address for diskless machines by sending out a packet that includes its MAC address and a request for the IP address assigned to that MAC address. A designated machine, called a *RARP server*, responds with the answer, and the Machine gets IP address

9.5 ROUTING

Routing takes place at the network layer (layer 3) of the Open Systems Interconnection (OSI) reference model. The term *routing* is used for taking a packet from one device and sending it through the network to another device on a different network. Routing is generally done by routers. Router only care about networks and the best path to each network. The logical network address of the destination host is used to get packets to a network through a routed network, and then the hardware address of the host is used to deliver the packet from a router to the correct destination host.

Router can perform routing by two different ways: static routing or dynamic routing. In *dynamic routing*, a protocol on one router communicates with the same protocol running on neighbor routers. The routers then update each other about all the networks they know about and place this information into the routing table. If a change occurs in the network, the dynamic routing protocols automatically inform all routers about the event. If *static routing* is used, the administrator is responsible for updating all changes by hand into all routers. Typically, in a large network, a combination of both dynamic and static routing is used.

Static Routing

Routers that use static routing are called static routers. Static routers are generally used in smaller networks that contain only a couple of routers or when security is an issue. Each static router must be configured and maintained separately because static routers do not exchange routing information with each other. For a static router to function properly, the routing table must contain a route for every network in the internetwork. Hosts on a network are configured so that their default gateway address matches the IP address of the local router interface. When a host needs to send a packet to another network, it forwards the packet to the local router, which checks its routing table and determines which route to use to forward the packet.

Static routing has the following advantages:

- There is no overhead on the router CPU, which means you could possibly buy a cheaper router than if you were using dynamic routing.
- There is no bandwidth usage between routers, which means you could possibly save money on WAN links.
- It adds security, because the administrator can choose to allow routing access to certain networks only.

Static routing has the following disadvantages:

- The administrator must really understand the internetwork and how each router is connected in order to configure routes correctly.
- If a network is added to the internetwork, the administrator has to add a route to it on all routers—manually.
- It's not feasible in large networks because maintaining it would be a full-time job in itself.

Dynamic Routing

Dynamic routing is a routing mechanism handled by a routing protocol, such as Routing Information Protocol (RIP) or Open Shortest Path First (OSPF) Protocol, which dynamically exchanges routing information among routers on an internetwork. Routers that use this method are called dynamic routers. For a dynamic router to function, a routing protocol must be installed on each router in the internetwork. The routing table of one router is manually seeded with routing information for the first hop, and then the routing protocol takes over and dynamically builds the routing table for each router. Routers periodically exchange their routing information so that if the internetwork is reconfigured or a router goes down, the routing tables of each router are modified accordingly. Hosts on a network need only be configured so that their default gateway address matches the IP address of the local router interface. Dynamic routers are much simpler to administer than static routers, but they are sometimes less secure because routing protocol information can be spoofed. If the network is reconfigured or a router goes down, it takes time for this information to propagate between the various routers on the network. Routing protocols also create additional network traffic, and different routing protocols offer their own advantages and disadvantages.

SUMMARY

This chapter covered network layer and IP routing. A protocol that defines routing and addressing is considered to be a network layer, or Layer 3, protocol. Networks may provide connection oriented (e.g., virtual circuit) or connectionless (datagram) service. The term *routing* is used for taking a packet from one device and sending it through the network to another device on a different network. Routing is generally done by routers.

PRACTICE SET

Review Questions

1. Explain Network layer service model.
2. Explain different types of routing?
3. What are the differences between static routing and dynamic routing?
4. What are the advantages and disadvantages of static routing?
5. Explain ARP protocol?
6. Explain RARP protocol?
7. Explain IP protocol with IP header?
8. Explain virtual circuit.
9. What are the differences between virtual circuit and datagrams?
10. What do you mean by ICMP? Why it is used?

Multiple Choice Questions

1. Which protocol does Ping use?

A) TCP B) ARP C) ICMP D) BootP

2. ARP stands for…….

A) All Reverse Protocol B) Address Reverse Protocol

C) Address Resolution Protocol D) None of the above

3. RARP stand for…….

A) Reverse All Reverse Protocol B) Reverse Address Reverse Protocol

C) Reverse Address Resolution Protocol D) None of the above

4. ICMP stands for ……

A) Internet Connection Managed Protocol

B) Internet Connection message Protocol

C) Internet Control Message Protocol

D) Internet Control Managed Protocol

5. IP stands for …….

A) Internet Policy B) International Policy

C) International Protocol D) Internet Protocol

6. Which of the following describes the functions of OSI Layer 3 protocols?

A) Logical addressing B) Physical addressing

C) Path Recovery D) Arbitration

7. Which of the following does a router normally use when making a decision about routing TCP/IP?

A) Destination MAC address B) Source MAC address

C) Destination IP address D) Source IP address

8. Which of the following are valid Class C IP addresses?

A) 1.1.1.1 B) 200.1.1.1 C) 128.128.128.128 D) 224.1.1.1

9. What is the range for the values of the first octet for Class A IP networks? (excluding reserved addresses)

A) 0 to 127 B) 0 to 126 C) 1 to 127 D) 1 to 126

10. How many valid host IP addresses does each Class B network contain?

A) 2^{16}-2 B) 2^{16} C) 256 D) 254

11. How many valid host IP addresses does each Class C network contain?

A) 256 B) 255 C) 254 D) 32,768

* * *

Chapter 10 Transport Layer

The *Transport layer* segments and reassembles data into a data stream. Services located in the Transport layer both segment and reassemble data from upper-layer applications and unite it onto the same data stream. They provide end-to-end data transport services and can establish a logical connection between the process on sending host and the process on destination host on an internetwork. Transport Layer responds to service requests from the application layer and issues service requests to the Internet layer.

OBJECTIVES OF THIS CHAPTER

After completing this chapter, the student should be able to:

- define role of transport layer
- explain the concept of port address, its use and purpose
- explain the working of TCP protocol
- explain TCP connection establishment and TCP connection termination phases
- explain the working of UDP protocol
- compare between TCP and UDP

10.1 INTRODUCTION

Transport layer resides between the application and network layer is in the core of the layered network architecture. It has the critical role of providing communication services directly to the application processes running on different hosts.

A transport layer protocol provides for **logical communication** between application processes running on different hosts. Logical - communication, mean the communicating application processes are not *physically* connected to each other (indeed, they may be on different machines, connected via numerous internetworking devices (such as routers) and a wide range of link types), from the applications' viewpoint, it is as if they were physically connected. From an application programmer's perspective, the transport layer provides inter-process communication between two processes that most often are running on different hosts. Application processes use the logical communication provided by the transport layer to send messages to each other, without worried of the details of the physical infrastructure used to carry these messages.

The Internet's network-layer protocol called as IP, which abbreviates "Internet Protocol". IP provides logical communication between hosts. The IP service model is a *best-effort delivery service*. This means that IP makes its "best effort" to deliver segments between communicating hosts, *but it makes no guarantees*. In particular, it does not guarantee segment delivery, it does not guarantee orderly delivery of segments, and it does guarantee the integrity of the data in the segments. For these reasons, IP is said to be an **unreliable service**. IP assumes that error detection and recovery, flow control and sequencing of segments should be done by upper layers.

The transport layer provides the end-to-end control of data exchanged between two systems, which include:

- Establishing and releasing connections between two end systems
- Providing end-to-end flow control
- Establishing the optimum data-unit size

- Segmenting single data units into multiple data-units and the reverse
- Providing end-to-end sequencing of segments (transport layer PDUs),
- Providing end-to-end error detection and error recovery

Transport layer protocols are implemented in the end systems but not on network routers. Network routers only act on the network-layer fields of the layer-3 protocol data units (PDU); they do not act on the transport-layer fields. At the sending side, the transport layer converts the messages it receives from a sending application process into transport-layer protocol data units. This is done by breaking the application messages into smaller chunks and adding a transport-layer header to each chunk to create transport layer PDU. The transport layer then passes the transport layer PDU to the network layer, where each transport layer PDU is encapsulated into a network layer PDU. At the receiving side, the transport layer receives the transport layer PDU from the network layer, removes the transport header from the transport layer PDUs, reassembles the messages and passes them to a receiving application process.

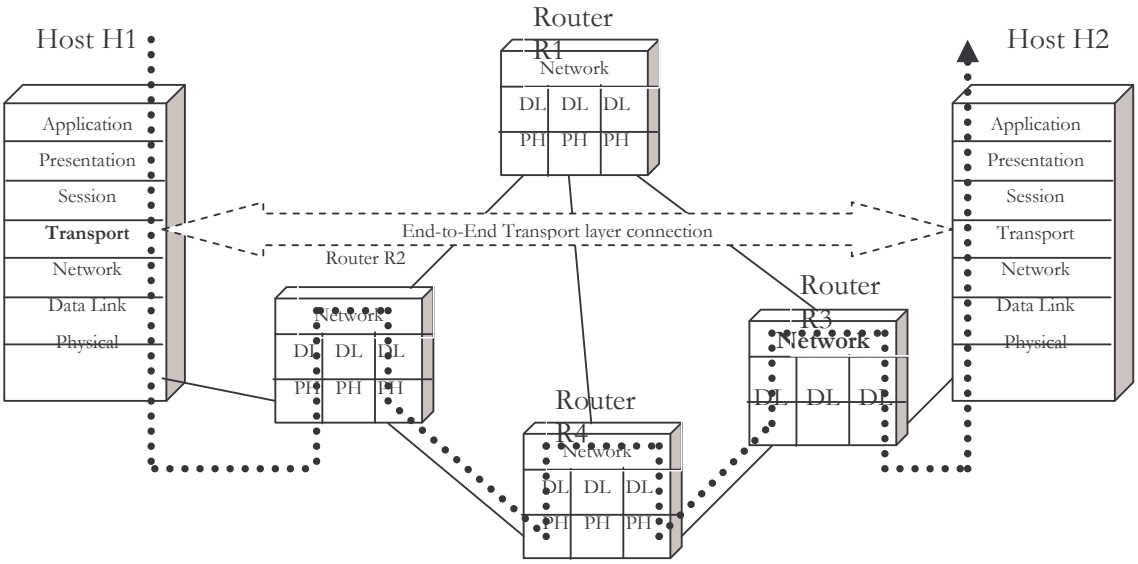


Figure 10.1 Transport Layer

The Transport layer can be connectionless, or connection-oriented. A *transport protocol* is a protocol on the transport layer. The two most widely used transport protocols on the Internet are the connection oriented TCP (Transmission Control Protocol), and UDP (User Datagram Protocol). TCP stands for Transmission Control Protocol and UDP stands for User Datagram Protocol. When designing a network application, the application developer must specify one of these two transport protocols. A transport layer protocol provides *logical communication between processes* running on different hosts; a network layer protocol provides *logical communication between hosts.*

Computers often run multiple programs at the same time. Source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on the other. The transport layer header therefore must include a type of address called a service-point address (or **port address**).

10.2 PORT ADDRESS

The primary duty of the ***transport layer*** is to provide communication from one application program to another. Such communication is often called ***end-to-end*** *or* ***process-to-process.*** The transport layer may regulate flow of information. It may also provide reliable transport, ensuring that data arrives without error and in sequence. To do so, transport protocol software arranges to have the receiving side send back acknowledgements and the sending side retransmit lost packets. The transport software divides the stream of data being transmitted into small pieces (sometimes called ***segments)*** and passes each packet along with a destination address to the next layer for transmission.

A general purpose computer can have multiple application programs accessing an internet at one time. The transport layer must accept data from several user programs and send it to the next lower layer. To do so, it adds additional information to each packet, including codes that identify which application program sent it and which application program should receive it. The code that identifies application program is called as *port* or *port address* or *protocol port number*. It is 16-bit field, that is port number can range from 0 to 65,535.

How should port numbers be assigned? The problem is important because two computers need to agree on port numbers before they can communicate. For example, when computer "A" wants to obtain a file from computer "B", it needs to know what port the file transfer program on computer "B" uses. For this purpose port address are divided into three categories namely: **well-known ports, registered ports** and **dynamic or temporary ports**.

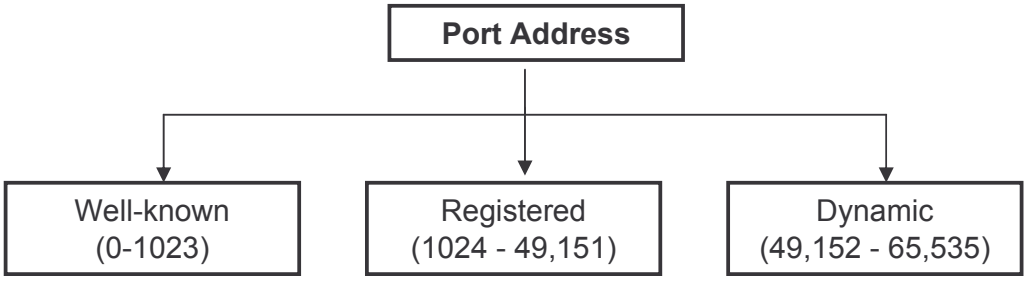


Figure 10.2 Port address

Well-known port assignment is controlled by central authority called as Internet Assigned Numbers Authority (IANA). IANA publish the list of port assignment. These ports are universally accepted. For example, FTP uses port number 21, which is a well-known port that is all over the world, if anyone having ftp server then it's listening on port number21. Well-known port numbers are assigned within the range 0 through 1023. The Registered Ports are those from 1024 through 49151. In the dynamic port numbers, ports are not globally known. Instead, whenever a program needs a port, the network software assigns one. Dynamic port numbers are not controlled by IANA. Dynamic ports range from 49,152 through 65535 and can be used by any process or program requesting it if the operating system has not already allocated it for a specific use.

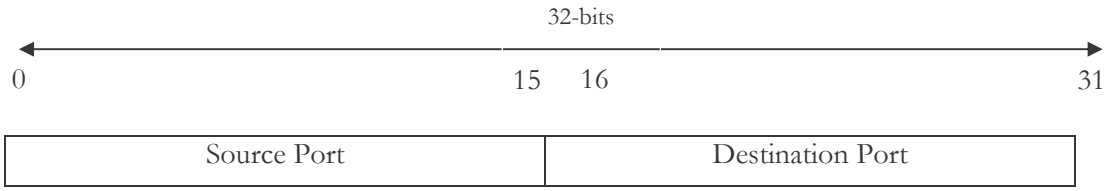
10.3 TRANSMISSION CONTROL PROTOCOL (TCP)

TCP is a transport layer protocol that enables reliable, connection-oriented network communication. The Internet Protocol (IP) works by exchanging groups of information called packets. Packets are short sequences of bytes that contain a header and a body. The header describes the destination that the packet needs to arrive at, and the routers on the internet pass the packets along in generally the right direction until it arrives at the final destination; the body contains application data.

The IP protocol can - in cases of congestion, discard packets, and for efficiency reasons two consecutive packets on the internet can take different routes to the destination. In all such cases, the packets can arrive at the destination in the wrong order. The TCP protocol's software libraries uses the IP protocol and provides to applications simpler interfaces, hides most of the underlying packet structure from applications, rearrange out-of-order packets, acts to minimize network congestion, and retransmits any packets that may have been discarded.

TCP is used extensively by many of the Internet's most popular application protocols and resulting applications, including the World Wide Web, E-mail, File Transfer Protocol (FTP), Secure Shell, and some streaming media applications. However, because TCP is optimized for accurate delivery rather than timely delivery, TCP sometimes incurs long delays while waiting for out-of-order messages or retransmissions of lost messages, and it is not particularly suitable for real-time applications such as Voice over IP. For such applications, protocols like the Real-time Transport Protocol (RTP) running over the User Datagram Protocol (UDP) are usually recommended instead.

The Transmission Control Protocol (TCP) is one of the main transport layer protocols used with IP. It is a connection oriented protocol based on the connectionless IP protocol. Because it is the lowest layer which has end-to-end communication, it needs to handle things such as lost packets.



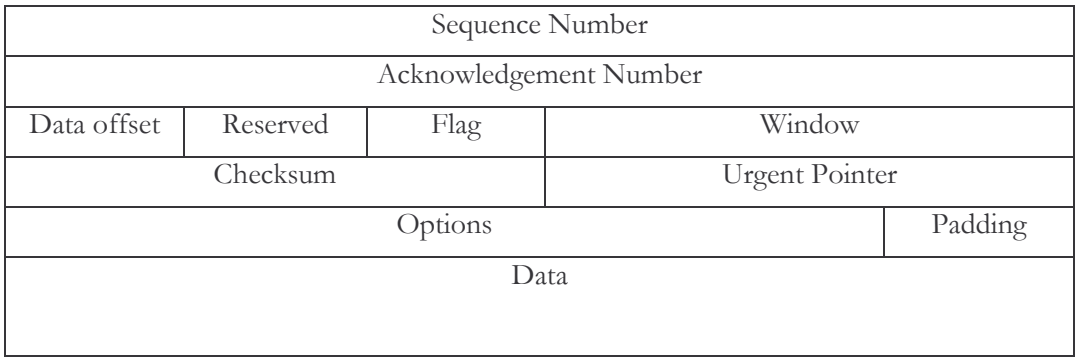


Figure 10.3 TCP Header

Source port

All of the address fields in the lower layer protocols are only concerned with getting the packet to the correct host. Often, however, we want to have multiple connections between two hosts. The source port is simply the number of the outgoing connection from the source host.

Destination port

Similarly, this is the number of the incoming connection on the destination host. There must be a program on the destination host which has some how told the networking system on that host that it will accept packets destined for this port number. Standard system services such as SMTP, HTTP and FTP have well known standard port numbers. Thus to connect to the SMTP (SMTP used to transfer emails) port on a particular host a TCP connection would be set up with the correct destination port number (25). The source port number does not matter except that it should be unique on the sending machine so that replies can be received correctly.

Sequence number

This is the sequence number of this packet. It differs from the usual data-link layer sequence number in that it is in fact the sequence number of the first byte of information and is incremented by the number of bytes in this packet for the next message. In other words, it counts the number of bytes transmitted rather than the number of packets.

Acknowledgement number

This is the sequence number of the last byte being acknowledged. This is a piggy-backed acknowledgement.

Data offset

This field is the offset in the packet of the beginning of the data field.

Flags

This field contains several flags relating to the transfer of the segment. TCP software uses the 6-bit field labeled ***Flag*** to determine the purpose and contents of the segment. The six bits tell how to interpret other fields in the header according to the table 10.1.

Bit	Meaning if set to 1
URG	Urgent pointer field is valid
ACK	Acknowledgement field is valid
PSH	This segment requests a push
RST	Reset the connection
SYN	Synchronize sequence numbers
FIN	Sender has reached end of its byte stream

Table 10.1 Fields in Flag

Window

This field is used in conjunction with the acknowledgement number field. TCP uses a sliding window protocol with a variable window size (often depending on the amount of buffer space available. This field contains the number of bytes which the host is willing to accept from the remote host.

Checksum

This field contains a checksum of the header. It actually uses a modified form of the header which includes some of the information from the IP header to detect some unusual types of errors.

Urgent pointer

There is provision in TCP for some urgent data messages to be sent bypassing the normal sequence number system. This field is used to indicate where such data is stored in the packet.

Options

As with IP, various options may be set.

Padding

Padding is added to make the header a multiple of 32 bits long. This is only necessary when options are used.

Data

The data field is passed intact to the program which is receiving packets addressed to this port.

NEED OF URGENT POINTER

TCP is a stream-oriented protocol; sometimes it is required for the program at one end of a connection to send data *out of band,* without waiting for the program at the other end of the connection to consume octets already in the stream. For example, when TCP is used for a remote login session, the user may decide to send a keyboard sequence that *interrupts* or *aborts* the program at the other end. Such commands are most often needed when a program on the remote machine fails to operate correctly. The command must be sent without waiting for the program to read octets already in the TCP stream.

To accommodate out of band command, TCP allows the sender to specify data as ***urgent,*** meaning that the receiving program should be notified of its arrival as quickly as possible, regardless of its position in the data stream. The protocol specifies that whenever urgent data is found, the receiving TCP should notify whatever application

program is associated with the connection to go into "urgent mode." After all urgent data has been consumed, TCP tells the application program to return to normal operation.

The mechanism used to mark urgent data when transmitting it in a segment consists of the URG flag bit and the U*rgent Pointer* field. When the URG bit is set, the urgent pointer specifies the position in the segment where urgent data ends.

TCP COMMUNICATION

TCP provides connections that need to be established before sending data. TCP communication have three phases. :

1. Connection establishment,
2. Data transfer,
3. Connection termination,

All TCP communication is connection oriented. A TCP session must be established before the hosts in the connection exchange data. Packets that are transferred between hosts are accounted for by assigning a sequence number to each packet. An ACK, or acknowledgment, is sent after every packet is received. If no ACK is received for a packet, the packet is re-sent. A TCP connection is opened by a **three-way handshake**.

Three stages of a TCP three-way handshake are as follows:

1. The initiating host sends a TCP packet requesting a new connection. This packet contains the initiating host's (Sender's) sequence number (for example, J) for the connection. The packet includes information such as a set SYN (synchronization) flag and data about the size of the window buffer on the initiating host.
2. The target host (Receiver) sends a TCP packet with its own sequence number and an ACK of the initiating host's sequence number. ACK (acknowledgement) for sender's sequence number means, sequence number sent by sender in step 1 plus 1 (for example, J+1).
3. The initiating host sends an ACK containing the target sequence number that it received. ACK for receiver's sequence number means, sequence number sent by receiver in step 2 plus 1 (for example, K+1)

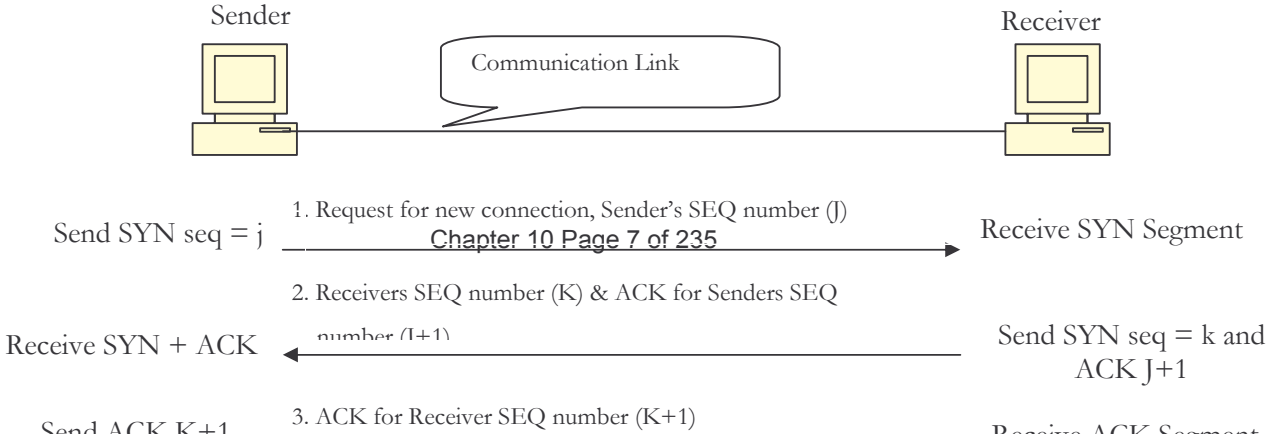


Figure 10.4 TCP Three-way Handshake

TCP 3-way handshake accomplishes two important functions. It guarantees that both sides are ready to transfer data and it allows both sides to agree on initial sequence numbers. Sequence numbers are sent and acknowledged during the handshake. Each machine must choose an initial sequence number at random that it will use to identify bytes in the stream it is sending. Sequence numbers cannot always start at the same value. For example, TCP cannot choose sequence number 1 every time it creates a connection.

Once a connection has been established, the TCP software can release data being held and deliver it to a waiting application program quickly. This phase is called as Data Transfer phase.

A similar three-way process with little bit modification is used to terminate a TCP session between two hosts. Using the same type of handshake to end the connection ensures that the hosts have completed their transactions and that all data is accounted for.

TCP connections are full-duplex. That is during normal operation; both of the devices are sending and receiving data simultaneously. Usually, connection termination begins with the process on just one device indicating to TCP that it wants to close the connection. The matching process on the other device may not be aware that its peer wants to end the connection at all. Several steps are required to ensure that the connection is shut down gracefully by both devices, and that no data is lost in the process. When an application program tells TCP that it has no more data to send, TCP will close the connection *in one direction*. To close its half of a connection, the sending TCP finishes transmitting the remaining data, waits for the receiver to acknowledge it, and then sends a segment with the FIN bit set. The receiving TCP acknowledges the FIN segment and informs the application program on its end that no more data is available. Once a connection has been closed in a given direction, TCP refuses to accept more data for that direction. Meanwhile, data can continue to flow in the opposite direction until the sender closes it. Of course, acknowledgements continue to flow back to the sender even after a connection has been closed. When both directions have been closed, the TCP process at each endpoint deletes its record of the connection.

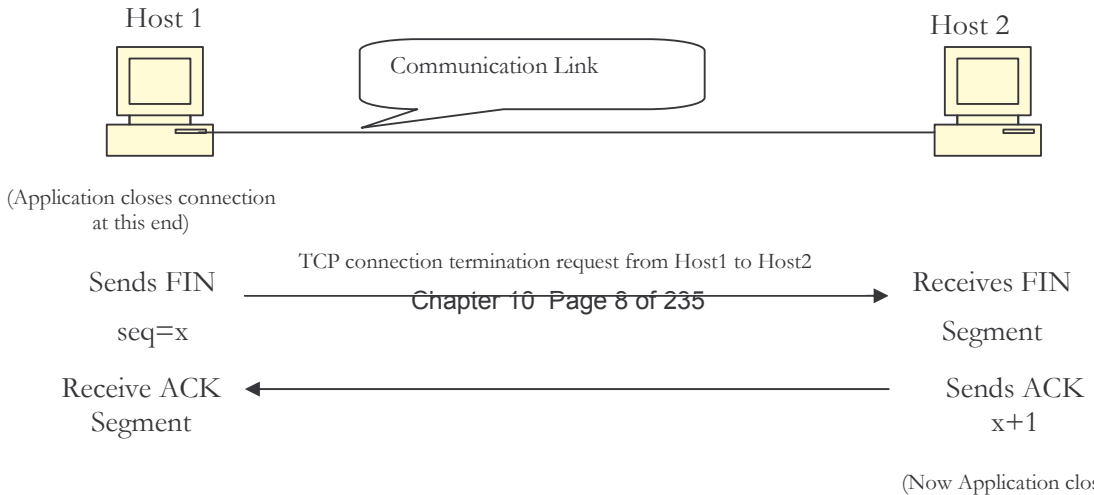


Figure 10.5 4-way Handshake for TCP Connection Termination

While closing the TCP connection, each direction must be shut down independently. The rule is that either end can send a FIN when it is done sending data. When a TCP receives a FIN, it must notify the application that the other end has terminated that direction of data flow. The sending of a FIN is normally the result of the application issuing a close.

The difference between three-way handshakes used to establish and break connections occurs after a machine receives the initial FIN segment. Instead of generating a second FIN segment immediately, TCP sends an acknowledgement and then informs the application of the request to shut down.

TCP PORT NUMBERS

TCP uses the port numbers to identify sending and receiving application end-points on a host. Each side of a TCP connection has an associated 16-bit unsigned port number (1-65535) reserved by the sending or receiving application. Arriving TCP data packets are identified as belonging to a specific TCP connection by its sockets, that is, the combination of source host address, source port, destination host address, and destination port. This means that a server computer can provide several clients with several services simultaneously, as long as a client takes care of initiating any simultaneous connections to one destination port from different source ports.

Following table shows examples of TCP ports:

Port Number	Program or Application
21	FTP
22	SSH
23	Telnet
25	SMTP
80	HTTP
53	DNS

Table 10.2 Example of TCP ports

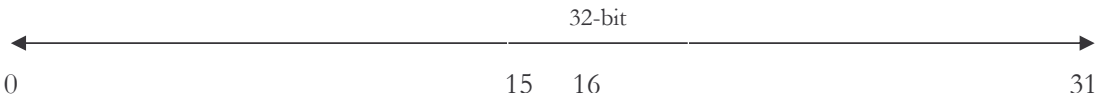
10.4 USER DATAGRAM PROTOCOL (UDP)

The User Datagram Protocol (UDP) is a datagram transport which uses the underlying IP protocol for its network layer. It is used when there is a need to transmit short packets through a network where there is no stream of data to be sent as in TCP. It is consequently a much simpler protocol and therefore much easier to handle. It is also less reliable in that there are no sequence numbers and other error recovery techniques available.

The *User Datagr am Protocol* or *UDP* provides the primary mechanism that application programs use to send datagrams to other application programs. UDP provides protocol ports used to distinguish among multiple programs executing on a single machine. That is, in addition to the data sent, each UDP message contains both a destination port number and a source port number, making it possible for the UDP software at the destination to deliver the message to the correct recipient and for the recipient to send a reply.

UDP does not use acknowledgements mechanism, it does not order incoming data or segments, and it does not provide feedback to control the rate at which information flows between the sender and receiver. Thus, UDP messages can be lost, duplicated, or arrive out of order. Furthermore, packets can arrive faster than the recipient can process them.

Each UDP message is called a ***user datagram***. UDP Header format is shown in the following diagram.



Source Port	Destination Port
Length	Checksum
Data	

Figure 10.6 UDP Header

Source Port

The 16-bit UDP protocol port numbers. It specifies the port to which replies should be sent.

Destination Port

The 16-bit UDP protocol port numbers. It specifies the port to which the message is destined

Length

The ***LENGTH*** field contains a count of octets in the UDP datagram, including the UDP header and the user data. Thus, the minimum value for ***LENGTH*** is eight, the length of the header only.

Checksum

The UDP checksum is optional and need not be used at all; a value of zero in the checksum field means that the checksum has not been computed.

Recall that, IP does not compute a checksum on the data portion of an IP datagram. Thus, the UDP checksum provides the only way to guarantee that data has arrived intact and should be used.

Data

This field is passed to the relevant program. It's an application data or message.

UDP PORT NUMBERS

Port Number	Program or Application
53	DNS
69	TFTP
110	POP3
119	News

Table 10.3 Examples of UDP Port Numbers

Transport Layer Protocol comparison table as shown below:

Parameter Description	UDP	TCP
Packet header size	8 Bytes	20 Bytes
Transport layer packet entity called as	Datagram	Segment
Port numbering used	Yes (16-bit)	Yes (16-bit)
Error detection	Optional	Yes
Reliability: Error recovery by automatic repeat request (ARQ)	No	Yes
Virtual circuits: Sequence numbering and reordering	No	Yes
Flow control	No	Yes

Table 10.4 Transport Layer Protocol Comparison

SUMMARY

Most computer systems permit multiple application process or programs to execute simultaneously. The User Datagram Protocol, UDP, or Transmission Control Protocol, TCP, distinguishes among multiple processes within a given machine by allowing senders and receivers to add two 16-bit integers called protocol port numbers to each UDP or TCP message. The port numbers identify the source and destination.

UDP messages can be lost, duplicated, delayed, or delivered out of order; the application program using UDP must handle these problems.

PRACTICE SET

Review Questions

1. What is the main advantage of using protocol ports instead of process identifiers to specify the destination within a machine?
2. Send UDP datagram across a network and measure the percentage lost and the percentage reordered. Does the result depend on the time of day? The network load?
3. Explain the role and services provided by transport layer?
1. Why is the UDP checksum separate from the IP checksum?
2. Explain logical addressing and its need?
3. Explain TCP 3-way handshake?
4. Explain UDP protocol?
5. Compare TCP and UDP protocol
6. Explain TCP Header format.
7. Explain UDP Header format.

Multiple Choice Questions

1. UDP stands for …….

A) Union Data Packet B) User Data Packet

C) User Datagram Protocol D) None of the above

2. TCP stands for ……..

A) Total Connection Protocol B) Timely Control Protocol

C) Transmission Control Protocol D) All of the above

3. Size of port number is ……..

A) 4 bit B) 8 bit C) 16 bit D) 32 bit

4. Range of Well-known port numbers is …….

A) 0 to 123 B) 0 to 1024 C) 0 to 65,535 D) 0 to 1023

5. Range of Dynamic port is …….

A) 1024 to 65,535 B) 0 to 65,535

C) 49,151 to 65,535 D) 49,152 to 65,535

6. ..... flag specifies that the field in urgent pointer is valid.

A) FIN B) PSH C) RST D) URG

7. TCP connections are …….

A) Half-Duplex B) Simplex C) Full-Duplex D) None of the above

8. TCP connection Establishment is …….. Handshake.

A) One-way B) Two-way C) Three-way D) Four-Way

9. TCP connection Termination is …….. Handshake.

A) One-way B) Two-way C) Three-way D) Four-Way

* * *

Chapter 11 Application Layer

In the application layer network-aware, user-controlled software is implemented - for example, e-mail, file transfer utilities, and terminal access. The application layer represents the interface between the user and the network. Examples of protocols that run at the application layer include File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), Telnet, Simple Mail Transfer Protocol (SMTP) and similar protocols that can be implemented as utilities the user can interface with. The application layer provides protocols for remote access and resource sharing.

OBJECTIVES OF THIS CHAPTER

After completing this chapter, the student should be able to:

- define role of application layer
- differentiate between network application and application-layer protocol
- define World Wide Web (WWW)
- explain HTTP and its working
- explain FTP and its working
- explain TFTP and differentiate between FTP and TFTP
- explain SMTP and its working

11.1 INTRODUCTION

A network application's software is distributed among two or more end systems (i.e., host computers). For example, with the Web there are two pieces of software that communicate with each other: the browser software in the user's host (PC or Workstation), and the Web server software in the Web server. With Telnet, there are again two pieces of software in two hosts: software in the local host and software in the remote host.

A process can be thought of as a program that is running within an end system. When communicating processes are running on the same end system, they communicate with each other using inter-process communication. The rules for inter-process communication are governed by the host system's operating system. We are not interested in how processes on the same host communicate with each other, but instead in how processes running on different systems (with same or different type of operating systems) communicate with each other. Processes on two different end systems communicate with each other by exchanging *messages* across the computer network. A sending process creates and sends messages into the network; a receiving process receives these messages and possibly responds back. Networking applications have **application-layer protocols** that define the format and order of the messages exchanged between processes, as well as the actions taken on the transmission or receipt of a message.

There is a difference between **network applications** and **application-layer protocols**. An application-layer protocol is only one piece (of a network application. Let's look at a couple of examples. The *Web is a network application* that allows users to obtain "documents" from Web servers on demand. The Web application consists of many

components, including a standard for document formats (i.e., HTML: Hyper Text Markup Language), Web browsers (e.g., Netscape Navigator, Mozilla Firefox or Internet Explorer), Web servers (e.g., Apache, IIS), and an application-layer protocol. The *Web's application-layer protocol is HTTP* (HyperText Transfer Protocol). HTTP defines how messages are sent between browser and Web server. Thus, HTTP is only one part of the Web application. As another example, consider the Internet electronic mail (E-mail) application. *E-mail is a network application*. Internet electronic mail (E-mail) also has many components, including mail servers that contains user mailboxes, mail readers that allow users to read and create messages, a standard for defining the structure of an email message (i.e., MIME) and application-layer protocols that define how messages are passed between servers, how messages are passed between servers and mail readers, and how the contents of certain parts of the mail message (e.g., a mail message header) are to be interpreted. *The application-layer protocol for electronic mail (E-mail) is SMTP* (Simple Mail Transfer Protocol). Thus, SMTP is only one part of the email application.

11.2 THE WORLD WIDE WEB (WWW)

The World Wide Web is a global hypertext system that was initially developed in 1989 by Tim Berners Lee at the European Laboratory for Particle Physics, CERN in Switzerland to facilitate an easy way of sharing and editing research documents among a geographically dispersed group of scientists.

The WWW Web consists of a large set of documents, called Web pages, that are accessible over the Internet. Each Web page is classified as a ***hypermedia*** document. The prefix ***hyper*** is used because a document can contain *selectable links* that refer to other, related documents; the suffix ***media*** is used to indicate that a document can contain items other than text (e.g., graphics images).

Two main building blocks are used to implement the Web. They are called as Web Browser and Web Server. A ***Web browser*** consists of an application program that a user invokes to access and display a Web page. The browser becomes a client that contacts the appropriate ***Web server*** to obtain a copy of the specified page. Because a given server can manage more than one Web page, a browser must specify the exact page when making a request. The number of Web servers is increasing rapidly and the traffic over port 80, which is the well-known port for HTTP Web servers

The Web consists of all client and server applications that communicate over the Internet using the client/server protocol Hypertext Transfer Protocol (HTTP).

WEB PAGE

Web page is a file of text information formatted using Hypertext Markup Language (HTML), and possibly including scripts and active content, that is sent by a Web server in response to a Web browser's request. A **Web page** (also called a Web document) consists of objects. An object is a simply file, such as a HTML file, a JPEG (*Joint Picture Encoding/Expert Group*) image, a GIF (*Graphics Interchange Format*) image, a Java applet, an audio clip, etc., that is addressable by a single URL (*Uniform Resource Locator*). Most Web pages consist of a base HTML file and several referenced objects. For example, if a Web page contains HTML text and five JPEG images, then the Web page has six objects: the base HTML file plus the five images.

Web pages are generally of two types:

Static Web page

Static Web pages are stored as files on the server in the same form that they are delivered to the client. These files usually have the extension *.htm* or *.html*.

Dynamic Web page

Pages with included scripts, ActiveX components, Java applets, ActiveX Data Objects (ADO), open database connectivity (ODBC) technologies, Dynamic HTML, or any other type of active content. They can also be Web pages that don't actually exist on the server until the client requests them, whereupon they are generated by the server using Active Server Pages (ASP) or some other server-side technology.

WEB BROWSER

A browser is referred to as an application that provides access to a Web server. You can use a Web browser, also called a "browser," to access (browse) content published on a Web server. This content can be static, meaning it consists of ASCII text files formatted using Hypertext Markup Language (HTML), or the content can be dynamic, meaning it is generated on demand using client-side or server-side scripting in languages such as JavaScript or Microsoft Visual Basic, Scripting Edition (VBScript).

The first graphical Web browser was developed in 1993 by a group of students headed by Marc Andreessen at the National Center for Supercomputing Applications (NCSA). This browser was known as Mosaic and was distributed free.

Web browsers typically offer additional features to make browsing the Web easier and more profitable. These features include the following:

- Toolbar buttons for navigating forward and backward through the tree of previously displayed pages, for stopping the download process, and for manually refreshing a page that loaded incompletely.
- Lists of favorites or bookmarks that store Uniform Resource Locators (URLs) of frequently accessed sites as well as tools for organizing and accessing those URLs.
- Options for specifying a default home page from which to begin browsing.
- Security options for handling such concerns as whether to allow scripts, ActiveX components, or Java applets to run on the browser.
- Facilities for displaying the underlying source code or HTML of a page, and even for editing and publishing Web content.

Examples of Web browsers include Mosaic, Netscape Navigator, Microsoft Internet Explorer, Mozilla Firefox, Opera and many more.

WEB SERVER

Web servers are responsible for servicing requests for information from Web browsers. The information can be a file retrieved from the server's local disk, or it can be generated by a program called by the server to perform a specific application function.

Examples of Web servers include IIS (Internet Information Server), PWS (Personal Web Server), Apache, and many more.

11.3 HYPERTEXT TRANSFER PROTOCOL (HTTP)

HTTP defines how Web clients (i.e., browsers) request Web pages from servers (i.e., Web servers) and how servers transfer Web pages to clients. HTTP is the protocol used for communication between a browser (such as Internet Explorer or Mozilla Firefox) and a Web server (such as IIS or Apache).

Don't confuse HTTP with HTML! HTTP is the protocol through which Web servers communicate with Web browsers. It is a control language for passing commands between clients and servers. HTML is Hypertext Markup Language, the language for constructing Web pages.

HTTP has the following set of characteristics:

Application Layer: H'ITP operates at the application level. It assumes a reliable, connection-oriented transport protocol such as TCP**,** but does not provide reliability or retransmission itself.

Request/Response: Once a transport session has been established, one side (usually a browser) must send an *HTTP* request to which the other side responds.

Stateless: Each H'ITP request is self-contained; the server does not keep a history of previous requests or previous sessions.

Bi-Directional Data Transfer: In most cases, a browser requests a Web page, and the server transfers a copy to the browser. HTTP also allows transfer from a browser to a server (e.g., when a user submits a "form").

Caching: To improve response time, a browser caches a copy of each Web page it retrieves. If a user requests a page again, HTTP allows the browser to interrogate the server to determine whether the content of the page has changed since the copy was cached.

Support For Proxy Server: HTTP allows a machine along the path between a browser and a server to act as a *proxy server* that caches Web pages and answers a browser's request from its cache.

HTTP is based on request-response activity. A client, running an application called a browser (or Web browser), establishes a connection with a server and sends a request to the server in the form of a request method. The server sends responds to the browser.

A browser contacts a Web server to obtain a page. The browser begins with a URL, extracts the hostname portion from URL, uses DNS (Domain Name Server) to map the name into an equivalent IP address, and uses the IP address to form a TCP connection to the server. Once the TCP connection is in place, the browser and Web server use HTTP to communicate; the browser sends a request to retrieve a specific page, and the server responds by sending a copy of the page.

An HTTP transaction is divided into four steps:

1. The browser opens a connection.
2. The browser sends a request to the server.
3. The server sends a response to the browser.
4. The connection is closed.

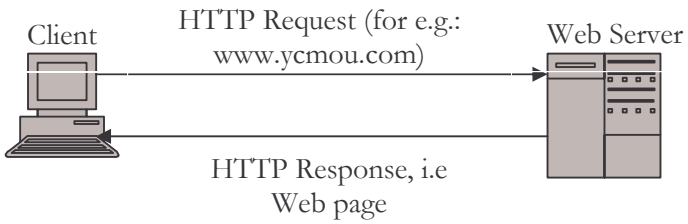


Figure 11.1 HTTP

PROXY SERVER

One of the most important features of HTTP is caching capability. In most cases, client requests and server responses can be stored in a cache within a reasonable amount of time, to handle the corresponding future requests. If the response is in the cache and accurate, there is no need to request another response from the server. This approach not only reduces the network bandwidth requirement but also increases the speed. There is a mechanism that the server estimates a minimum time in which the response message will be valid. That means, an expiration time is determined by the server for that particular response message. Therefore, within this time the message can be used without referring to the server.

A Web cache also called as a proxy server is a network entity that satisfies HTTP requests on the behalf of a client. The Web cache has its own disk storage, and keeps in this storage copies of recently requested web pages.

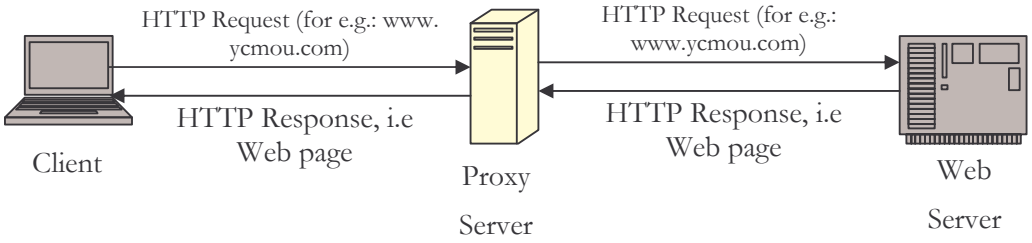


Figure 11.2 Proxy server

As an example, suppose a browser is requesting the web page http://www.ycmou.com

- The browser establishes a TCP connection to the proxy server and sends an HTTP request for the web page to the Web cache or Proxy server.
- The proxy server checks to see if it has a copy of the web page stored locally. If it does, the proxy server forwards the web page within an HTTP response message to the client browser.
- If the proxy server does not have the web page, the Proxy server opens a TCP connection to the origin server, that is, to www. ycmou.com. The Proxy server then sends an HTTP request for the web page into the TCP connection. After receiving this request, the origin server sends the web page within an HTTP response to the Proxy server.
- When the Proxy server receives the web page, it stores a copy in its local storage and forwards a copy, within an HTTP response message, to the client browser (over the existing TCP connection between the client browser and the Proxy server).

11.4 FILE TRANSFER PROTOCOL (FTP)

FTP stands for File Transfer Protocol. It's the concept of moving a file from your machine to FTP server or vice-versa. The file transfer protocol allows you to connect to a remote computer (host) using an FTP program on your machine, browse a list of files available, retrieve files, and navigate the directory structure of the host system.

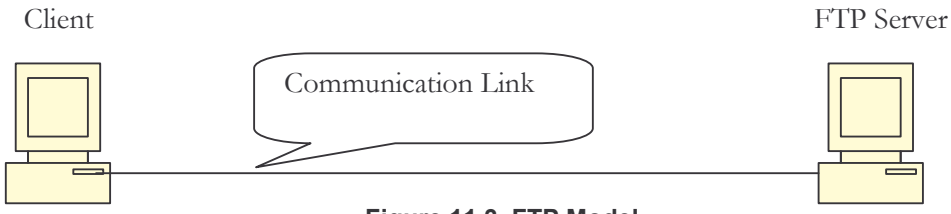


Figure 11.3 FTP Model

In a typical FTP session, the user is sitting in front of one host (the local host, such as FTP Client) and wants to transfer files to or from a remote host (such as FTP Server). In order for the user to access the remote account, the user must provide user identification and a password. After providing this authorization information, the user can transfer files from the local file system to the remote file system and vice versa. Copying files from one machine to another is one of the most frequently used operations. The data transfer between client and server can be in either direction. The client can send a file to the server machine. It can also request a file from this server. To access remote files, the user must identify himself or herself to the server. At this point the server is responsible for authenticating the client before it allows the file transfer.

FTP uses TCP as a transport protocol to provide reliable end-to-end connections. The FTP server listens to connections on port 21. FTP uses two ports; port 21 and port 20. That is, two connections are used: the first is for login and the second is for managing the data transfer. As it is necessary to log into the remote host, the user must have a user name and a password to access files and directories. The user who initiates the connection assumes the client function, while the server function is provided by the remote host.

FTP is an example of a *client-server* system. You use a *client* program on your system to connect to the *server* running on the remote host. The server coordinates activity between the client and the host operating system. FTP differs from other client-server applications in that it establishes two connections between the client and the server. One connection is used for data transfer (established on TCP port 20), the other for control information (commands and response, established on TCP port 21). Whereas, other client-server based applications uses only one connection.

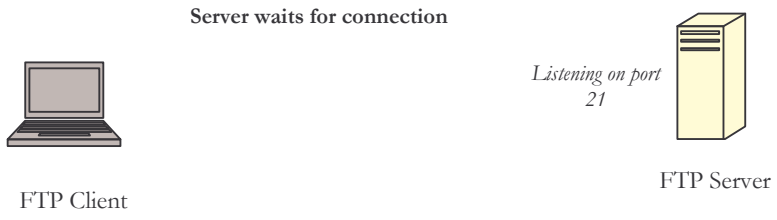


Figure 11.4 (a) Server listening on port 21 and waits for connection

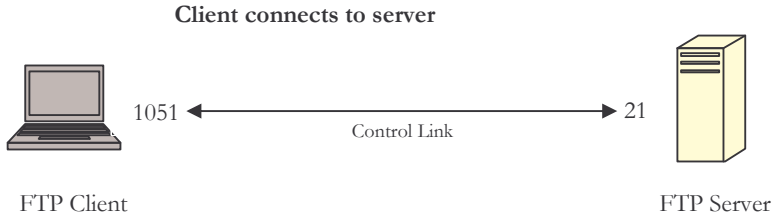


Figure 11.4 (b) Client Connects to server on port 21

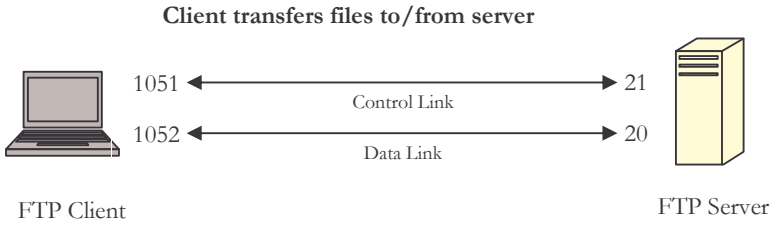


Figure 11.4 (c) Client transfers files to/from server

Figure 11.4 Working of FTP

An FTP client is an application that can issue FTP commands to an FTP server, while an FTP server is a service or daemon running on a server that responds to FTP commands from a client. FTP commands can be used to change directories, change transfer modes between binary and ASCII, upload files, and download files. FTP uses Transmission Control Protocol (TCP) for reliable network communication by establishing a session before initiating data transfer. TCP port number 21 on the FTP server listens for connection attempts from an FTP client and is used as a control port for establishing a connection between the client and server, for allowing the client to send an FTP command to the server, and for returning the server's response to the command. Once a control connection has been established, the server opens port number 20 to form a new connection with the client for transferring the actual data during uploads and downloads.

When using FTP, the user will perform some or all of the following operations:

1. Connect to a FTP Server
2. Select a directory
3. List files available for transfer
4. Define the transfer mode
5. Copy files to or from the FTP Server
6. Disconnect from the FTP Server

1. Connect to FTP Server

To execute a file transfer, the user begins by logging into the remote host. The user must have a user ID and password for the remote host.

Commands that are used to connect FTP Server:

Open	Selects the remote host and initiates the login session
User	Identifies the remote user ID
Pass	Authenticates the user

Table 11.1 FTP connection commands

2. **Directory selection**
   When the control link is established, the user can use the **cd** (change directory) subcommand to select a remote directory to work with.

3. **File Listing**
   This task is performed using the ***dir*** or ***ls*** subcommands.

4. **Define Transfer Mode**
   Transferring data between dissimilar systems often requires transformations of the data as part of the transfer process. The user has to decide on two aspects of the data handling: First, the way the bits will be moved from one place to another; and Second, the different representations of data upon the system's architecture.

   Different transfer mode commands are ASCII or BINARY.

5. **Copying files to or from FTP Server**
   The following commands can be used to copy files between FTP clients and FTP servers:

Get	Copies a file from the remote host (FTP Server) to the local host (FTP Client).
Mget	Copies multiple files from the remote (FTP Server) to the local host (FTP Client).
Put	Copies a file from the local host (FTP Client) to the remote host (FTP Server).
Mput	Copies multiple files from the local host (FTP Client) to the remote host (FTP Server).

Table 11.2 FTP Commands to Transfer Files

6. **Disconnecting from FTP Server**
   The following commands are used to end an FTP session:

Quit	Disconnects from the remote host (FTP Server) and terminates FTP. Some implementations use the BYE subcommand.
Close	Disconnects from the remote host (FTP Server) but leaves the FTP client running. An open command can be issued to work with a new host.

Table 11.3 FTP Commands to Terminate Session

REPLY CODE

In order to manage these operations, the client and server conduct a dialog. The client issues commands, and the server responds with reply codes.

Reply codes are three digits long, with the first digit being the most significant.

Code	Description

Code	Description
100 Codes	**The requested action is being taken. Expect a reply before proceeding with a new command.**
110	Restart marker reply.
120	Service ready in (n) minutes.
125	Data connection already open, transfer starting.
150	File status okay, about to open data connection.
200 Codes	**The requested action has been successfully completed.**
200	Command okay.
202	Command not implemented
211	System status or system help reply.
212	Directory status.
213	File status.
214	Help message.
215	NAME system type. (NAME is an official system name from the list in the Assigned Numbers document.)
220	Service ready for new user.
221	Service closing control connection. (Logged out if appropriate.)
225	Data connection open, no transfer in progress.
226	Closing data connection. Requested file action successful (file transfer, abort, etc.).
227	Entering Passive Mode
230	User logged in, proceed.
250	Requested file action okay, completed.
257	"PATHNAME" created.
300 Codes	**The command has been accepted, but the requested action is being held pending receipt of further information.**
331	User name okay, need password.
332	Need account for login.
350	Requested file action pending further information.
400 Codes	**The command was not accepted and the requested action did not take place.** **The error condition is temporary, however, and the action may be requested again.**
421	Service not available, closing control connection.
425	Can't open data connection.
426	Connection closed, transfer aborted.
450	Requested file action not taken. File unavailable (e.g., file busy).
451	Requested action aborted, local error in processing.
452	Requested action not taken. Insufficient storage space in system.

Code	Description
500 Codes	**The command was not accepted and the requested action did not take place.**
500	Syntax error, command unrecognized. This may include errors such as command line too long.
501	Syntax error in parameters or arguments.
502	Command not implemented.
503	Bad sequence of commands.
504	Command not implemented for that parameter.
530	User not logged in.
532	Need account for storing files.
550	Requested action not taken. File unavailable (e.g., file not found, no access).
552	Requested file action aborted, storage allocation exceeded
553	Requested action not taken. Illegal file name.

Table 11.4 FTP Reply Code Description

FTP Scenario

Suppose a user wants to transfer a file (viit.gif) stored in **sample** directory on a FTP Server (viit.ac.in) to a workstation.

As described above user will perform actions one by one:

1. Connecting to FTP Server
   ```
   ftp ycmou.com
   LOGIN: ycmou
   PASSWORD:xxxxxx
   ```

2. Directory Selection
   ```
   ftp>cd sample
   ```

3. List files available for transfer
   ```
   ftp> ls
                              OR
   ftp>dir
   ```

4. Define the transfer mode
   ```
   ftp>binary
   ```

5. Copy files to or from the FTP Server
   ```
   ftp>get ycmou.gif
   ```

6. Disconnect from the FTP Server
   ```
   ftp>quit
   ```

11.5 Trivial File Transfer Protocol (TFTP)

TFTP stands for Trivial File Transfer Protocol. It's the concept of moving a file from your machine to TFTP server or vice-versa. TFTP differs from the popular File Transfer Protocol (FTP) in that it does not support any form of authentication. TFTP is an extremely simple protocol to transfer files. It is implemented on top of UDP (User Datagram Protocol). Note that, TFTP has no provisions for user authentication; in that respect, it is an unsecured protocol.

TFTP copies files to and from remote hosts by using the User Datagram Protocol (UDP). The remote host must be running the TFTP service or daemon for the TFTP client to be able to communicate with it. In UNIX networks that use diskless workstations and the bootstrap protocol (BOOTP), TFTP is usually used to download the boot disk image from the BOOTP server to the workstation.

Following table shows comparisons between FTP and TFTP:

Parameter Description	**FTP**	**TFTP**
Protocol	TCP	UDP
Port	20 and 21	69
Reliable	Yes	No
Speed	Slow	Fast as compared to FTP
Authentication	Yes	No
Data Transfer	Connection Oriented	Connection-less

Table 11.5 Comparison between FTP and TFTP

11.6 Electronic Mail (E-mail)

The use of a network to transmit text messages, memos, and reports; usually referred to as e-mail. Users can send a message to one or more individuals, to a predefined group, or to all users on the system. When you receive a message, you can read, print, forward, reply, or delete it.

The two most popular messaging formats used today are the Internet Simple Mail Transfer Protocol (SMTP) and X.400 mail systems. X.400 is a popular messaging format that is used throughout much of Europe, but SMTP mail, which was developed in the United States, enjoys worldwide popularity and acceptance. Both systems are based on a client/server architecture, with messaging clients (such as Outlook express) sending e-mail to mail servers that act as message transfer agents by routing messages through a backbone of mail servers to their final destination.

E-mail Address

E-mail address is any of several types of addresses that ensure an e-mail message reaches its intended recipient. An e-mail address must contain sufficient information so that the message can be routed to its specific recipient. There are various kinds of e-mail address formats depending on the e-mail system in use. Address formats typically include at least two parts:

1. A user portion, which indicates the name or alias of the user to whom the mail is directed

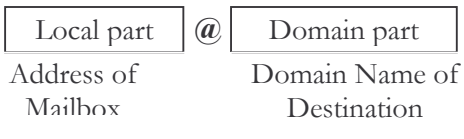
2. A routing portion, which indicates the information needed to route the message to the particular mail system on which the user has his or her mailbox.

The following table shows some examples of e-mail address formats.

Type of Address	Example
SMTP (Internet)	patilkr@viit.ac.in
X.400	C=India; a=SPRINT; p=viit; s=patilkr

Table 11.6 Email addresses

To deliver mail, a mail handling system must use an addressing system with unique address. The addressing system used by SMTP consists of two parts: a local part and a domain name, separated by @ sign.



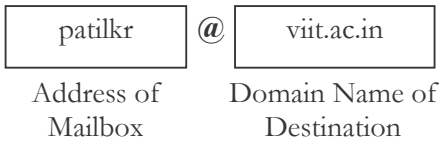


Fig 11.5(a) SMTP E-mail address format **Fig 11.5(b) SMTP Email address example**

Figure 11.5 SMTP E-mail address

The local part defines name of user mailbox, where all the mail received for a user is stored for retrieval by the mail reader (user agent). An organization usually selects one or more hosts to receive and send mail; they are called as *mail exchangers*. The domain name assigned to each mail exchanger either comes from the DNS database or is a logical name (e.g., the name of the organization).

Mailbox

In e-mail systems, an area of hard-disk space used to store e-mail messages until users can access them. An onscreen or audio message often tells users that they have mail. Before email can be sent to an individual, the person must be assigned an *electronic mailbox*. The mailbox consists of passive storage area. An electronic mailbox is private, that is, the permissions are set to allow the mail software to add an incoming message to an arbitrary mailbox, but to deny anyone except owner to examine or remove message. The email system periodically checks the mailboxes. If a user has mail, it informs the user with a notice.

11.7 SIMPLE MAIL TRANSFER PROTOCOL (SMTP)

SMTP stands for Simple Mail Transfer Protocol. SMTP is a standard application-layer protocol for delivery of e-mail over an internetwork such as the Internet. *Simple Mail Transfer Protocol (SMTP)* is the standard mechanism for electronic mail in the Internet. It uses well-known TCP port 25.

SMTP defines the format for messages sent between hosts on the Internet. SMTP uses plain 7-bit ASCII text to send e-mail messages and to issue SMTP commands to receiving hosts. Multipurpose Internet Mail Extensions (MIME) is typically used to encode multipart binary files including attachments into a form that SMTP can handle.

SMTP provides a mechanism for forwarding e-mail from one host to another host over the Internet. SMTP services running on a host first establish a connection to a remote host using Transmission Control Protocol (TCP) port 25. An SMTP session is then initiated by sending a helo command and receiving an OK response. The sending computer then uses the following commands to send messages:

Mail fr:

Identifies the sending host to the receiving host.

Rcpt to:

Identifies the targeted message recipient to the receiving host by using the Domain Name System (DNS) format user@DNSdomain

Data:

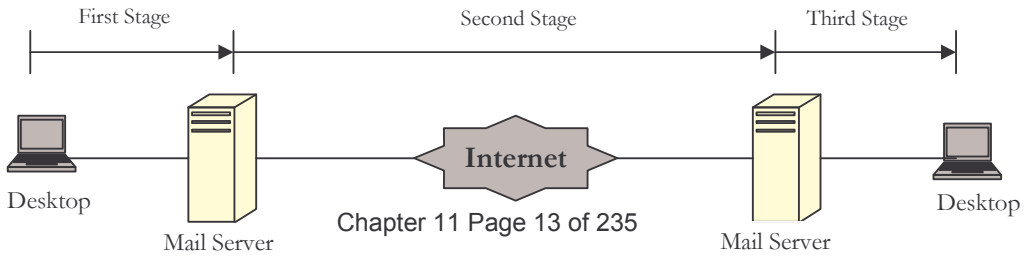
Initiates the sending of the message body as a series of lines of ASCII text, ending with a single period (.) alone on a line

Quit:

Closes the SMTP connection

SMTP is based on end-to-end delivery; an SMTP client will contact the destination host's SMTP server directly to deliver the mail. It will keep the mail item being transmitted until it has been successfully copied to the recipient's SMTP. This is different from the store-and-forward principle that is common in many mailing systems, where the mail item may pass through a number of intermediate hosts in the same network on its way to the destination and where successful transmission from the sender only indicates that the mail item has reached the first intermediate hop

The delivery of email from the sender to the receiver consists of three stages.



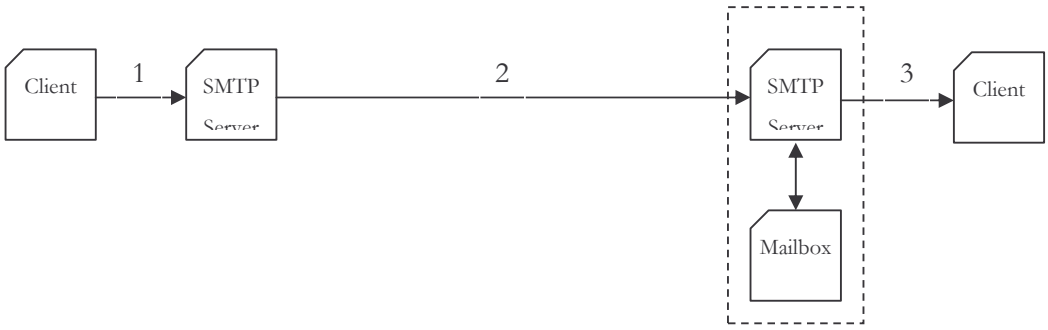


Figure 11.6 E-mail Delivery

In the first stage, the email goes from the user agent to the local server. The mail does not go directly to the remote server because the remote server may not be available at all times. Therefore, the mail is stored in the local server until it can be sent. The user agent uses SMTP client software, and the local server uses SMTP server software.

In the second stage, the email is delivered to remote server, not to the remote user agent. The reason is that a server is always running however, people often turn off their computer when it's not required. The email is received by this mail server and stored in the mailbox of the user for later retrieval.

In the third stage, the remote user agent uses a mail access protocols such as POP3 (Post Office Protocol, version 3) or IMAP4 (Internet Mail Access Protocol, version 4) to access the mailbox and obtain the mail.

WEB-BASED MAIL

Today most of the websites provides email service to anyone how access the site. For example Hotmail, Yahoo, Indiatimes, Rediffmail, GMail, etc.

Mail transfer from Comp_A's browser to its mail server is done thro HTTP. The transfer of message from the sending mail server to the receiving mail server is still thro SMTP. Finally, the message from the receiving server to Comp_B's browser is done thro HTTP. In last phase of Web-based mail access instead of POP3 or IMAP4, HTTP is used.

11.8 DOMAIN NAME SYSTEM (DNS)

We human beings can be identified in many ways. For example, we can be identified by the names that appear on our birth certificates. We can be identified by our driver's license numbers. Just as humans can be identified in many ways, so too can Internet hosts. One identifier for a host is its **hostname**. Hostnames such as www.viit.ac.in, www.gmail.com, www.google.co.in and www.rediffmail.com, are mnemonic and are therefore easily remembered by humans. But, hostnames can consist of variable-length alpha-numeric characters; they would be difficult to process by routers (Internetworking device). For these reasons, hosts are also identified by so-called **IP addresses.**

We have just seen that there are two ways to identify a host; a hostname and an IP address. People prefer the more hostname identifier, while routers prefer fixed-length, hierarchically-structured IP addresses. In order to reconcile these different preferences, we need a directory service that translates hostnames to IP addresses. This is the main task of the Internet's **Domain Name System (DNS)**.

The DNS is a distributed database implemented in a hierarchy of **name servers** and an application-layer protocol that allows hosts and name servers to communicate in order to provide the translation service. The DNS protocol runs over UDP or TCP and uses port 53. DNS stands for two things: Domain Name System and Domain Name Servers. One acronym defines the protocol; the other defines the machines that provide the service.

Every IP address on the Internet is actually a series of four numbers separated by periods (called dots), such as 220.225.40.201. It would be impossible for you to remember these numeric addresses when you wanted to send e-mail or visit a site. Also, because sometimes numeric IP addresses change, you would never be able to know every time those numeric addresses change. The DNS solves these problems.

The Domain Name System (DNS) provides:

- A method for identifying hosts with friendly names instead of IP addresses.
- A distributed mechanism for storing and maintaining lists of names and IP addresses of hosts.
- A method for locating hosts by resolving their names into their associated IP addresses so that network communication can be initiated with the host.

NAME SPACE

Name space is the abstract space or collection of all possible addresses, names, or identifiers of objects on a network, internetwork, or the Internet. A namespace is "the space of all names" for a given type of network name. A DNS name space can be organized in two different ways: Flat and Hierarchical

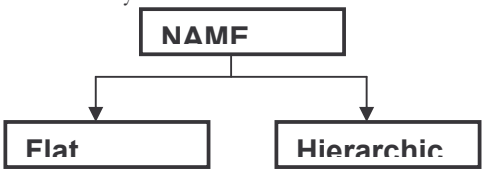


Figure 11.7 Types of name space

Flat Name Space

In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure. The names may or may not have a common section. The main disadvantage of a flat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

Hierarchical Name Space

In a hierarchical name space, each name is made of several parts. The first part can define the nature of organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on. In this case, the authority to assign and control the name space can be decentralized. A central

authority can assign the part of the name that defines the nature of the organization and the name of the organization. The responsibility of the rest of the name can be given to organization itself.

DOMAIN NAME SPACE

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127. Whereas the root gules the whole tree together, each level of the tree defines a hierarchical level. The domain name space (tree) is divided into two different sections: **generic domains**, and **country domains**.

The **generic domains** define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database.

Label	Description
com	Commercial organization, such as Hewlett-Packard (www.*hp.com*), Sun Microsystems (www.*sun.com*), and IBM (www.*ibm.com*).
edu	Educational institute, such as VIT (*www.vit.edu*) and Purdue University (www.*purdue.edu*).
gov	Government institute, such as NASA (*nasa.gov*) and the National Science Foundation (*nsf.gov*).
int	International Organization, such as NATO (*nato.int*).
mil	Military groups, such as the U.S. Army (*army.mil*) and Navy (*navy.mil*).
net	Network support engineers, such as NSFNET (*nsf.net*).
org	Nonprofit organization, such as the Electronic Frontier Foundation (*eff.org*).

Table 11.7 Example of generic domains

The country domain section follows the same format as the generic domains but uses two-character country abbreviations in place of three character organizational abbreviations at the first level.

Label	Description
au	Australia
ca	Canada
in	India
uk	United Kingdom
fr	France
th	Thailand
us	United States

zw	Zimbabwe

Table 11.8 Example of country domains

The tree's hierarchical structure, shown in figure, is similar to the structure of the UNIX filesystem. The tree has a single root at the top. In the UNIX filesystem, this is called the root directory, represented by a slash ("/"). DNS simply calls it the "root." Like a UNIX filesystem, DNS's tree can branch any number of ways at each intersection point, called a node. The depth of the tree is limited to 127 levels.

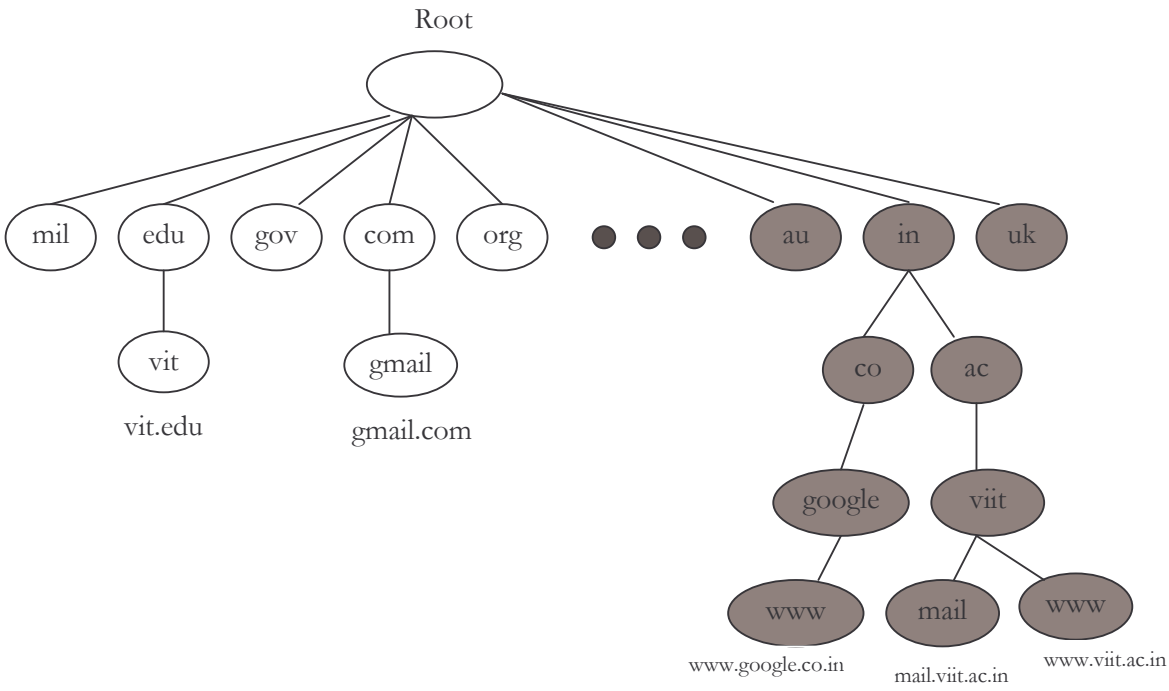


Figure 11.8 Domain Name Space

Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string). .

A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers. The Internet has 13 (thirteen) root name servers spread across different parts of the network.

The DNS creates a hierarchy of domains or groups of computers and it establishes a *domain name* (also known as an Internet address) for each computer on an intranet or the Internet, using easily recognizable letters and words instead of numbers.

DNS domains can be classified as one of the following:

- A parent domain, which contains other domains. An example of a parent domain is vit.edu.
- A child domain, or subdomain, which is contained within a parent domain. Examples of child domains in the vit.edu parent domain are comp.vit.edu, mech.vit.edu, it.vit.edu.

Domain name is a name for a domain within the Domain Name System (DNS), usually registered with the Internet Network Information Center (InterNIC)—for example, the *viit.ac.in* domain owned by "Vishwakarma Institute of Information Technology". Domain names can include only the characters a–z, A–Z, and 0–9, the dash (-), and the period. InterNIC manages root and top level domain names. Top level domain names means country domain names, and generic domain names. Local admin manages third level and more level domain names.

Each node in the tree has a text label (without dots) that can be up to 63 characters long. A null (zero-length) label is reserved for the root. The full *domain name* of any node in the tree is the sequence of labels on the path from that node to the root. Domain names are always read from the node toward the root ("up" the tree), and with dots separating the names in the path. A full domain name is a sequence of labels separated by dots (.). The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

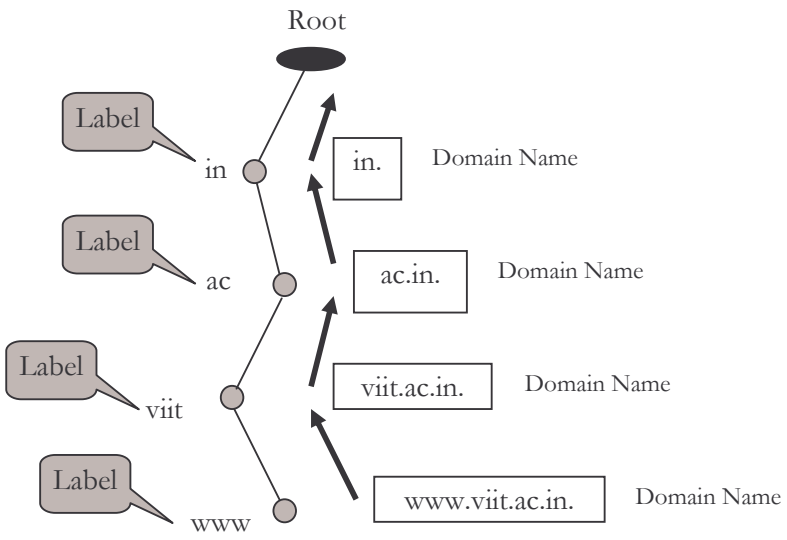


Figure 11.9 Domain Name and Labels

Fully Qualified Domain Name (FQDN) or Absolute Domain Name

In the Domain Name System (DNS), a dotted name that fully identifies a host or machine on the Internet. A fully qualified domain name (FQDN) of a host consists of its host name dotted together with its domain name and any names of subdomains in which the host resides. FQDNs are used in Uniform Resource Locators (URLs) for accessing Web pages on the Internet and provide an absolute path through the DNS namespace to the target host on which the Web page resides. They are also sometimes called absolute domain names.

For example, for the FQDN

www.viit.ac.in.

The host name is *www* and its domain is *viit.ac.in.*.

If a label is terminated by a null string, it is called a fully qualified domain name (FQDN) or absolute domain name. Note that the name must end with a null label, but because null here means nothing, the label ends with a dot (.).

Partially Qualified Domain Name (PQDN) or Relative Domain Name

If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN) or relative domain name. A PQDN starts from a node, but it does not reach the root.

FQDN	PQDN
www.viit.ac.in.	www.viit.ac.in
www.google.co.in.	www
mail.viit.ac.in	mail.viit

Table 11.9 Examples of FQDN and PQDN

A DNS server provides '*name resolution service*' which means that DNS servers resolve names into IP addresses or vice versa. DNS servers are also called *name servers*. Every computer on the Internet has a unique IP addresses (a series of four decimal numbers from 0 to 255 separated by dots). A DNS server is used to 'resolve' a name into an IP address (or vice versa).

A simple design for DNS would have one Internet name server that contains all the mappings. In this centralized design, clients simply direct all queries to the single name server, and the name server responds directly to the querying clients.

The problems with a centralized design include:

- **A single point of failure.** If the name server crashes, so too does the entire Internet.
- **Traffic volumes.** A single name server would have to handle all DNS queries generated from millions of hosts over an Internet.
- **Distant centralized database.** A single name server cannot be "close" to all the querying clients. If we put the single name server in New York City, then all queries from Australia must travel to the other side of the globe, perhaps over slow and congested links. This can lead to significant delays.
- **Maintenance.** The single name server would have to keep records for all Internet hosts. Because of this centralized database will become very huge, as well as it would have to be updated frequently to account for every new host.

In summary, a centralized database in a single name server simply *doesn't scale*. Consequently, the DNS is distributed by design. DNS is a very good example of how distributed database can be implemented over an Internet.

RESOLUTION

Resolution is the process of turning a name into an IP address or an IP address back into a name. The process begins with asking the root servers which name-server to ask about the domain name that is being resolved. The root name-servers will refer you to another

name server, which will refer you to another name server and so on. Finally you will be referred to the name-server, which is authoritative for the domain name.

DNS is designed as a client/server application. A host that needs to map an address to a name or a name to an address calls a DNS client called a **resolver**. Resolvers are the clients that access name servers. Programs running on a host that need information from the domain name space use the resolver.

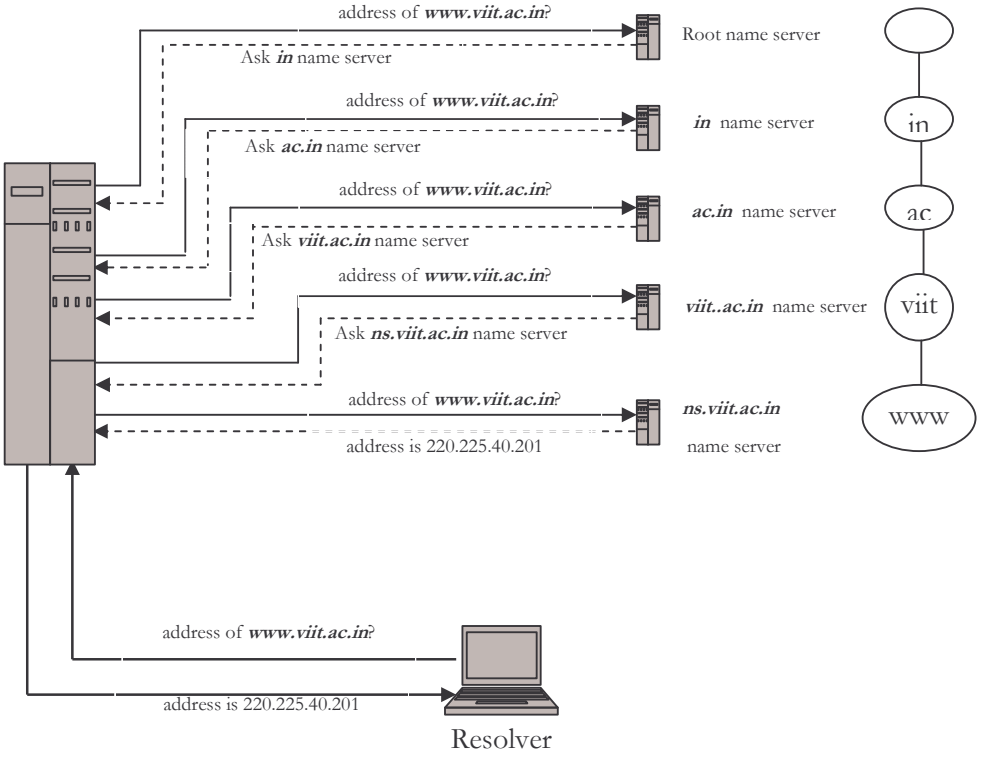


Figure 11.10 Resolution of www.viit.ac.in on the Internet

The local name server queries a root name server for the address of *www.viit.ac.in. Root server refres it to* the *in* name servers. The local name server asks an *in* name server the same question, and is referred to the *ac.in* name servers. The *ac.in* name server refers the local name server to the *viit.ac.in* name servers. The *viit.ac.in* name server refers the local name server to the *ns.viit.ac.in* name servers. Finally, the local name server asks an *ns.viit.ac.in* name server for the address and gets the answer (that is, IP address of www.viit.ac.in).

11.9 COMPARISONS

Comparison of HTTP and FTP

HTTP	FTP
HTTP transfers files (or Web page) from Server to Client	FTP transfers files from Server to Client as well as Client to server machine.
HTTP uses service of TCP	FTP also uses services of TCP

HTTP	FTP
HTTP uses only one port for transmission. (Port 80)	FTP uses two TCP ports for connection. (Port 20 and Port 21)
In HTTP there is no separate control connection; only data are transferred between the client and the server.	In FTP there are two separate connection; one to transfer control information (port 21) and other two transfer data (port 20) between server and client.

Table 11.10 HTTP and FTP Comparisons

Comparison of HTTP and SMTP

HTTP	SMTP
HTTP transfers messages from server to client and client to server.	SMTP transfers messages from client to the server.
HTTP uses service of TCP	SMTP also uses services of TCP
HTTP uses TCP port 80 for transmission.	SMTP uses TCP port 25 for transmission.
HTTP messages are not destined to be read by humans. They are read and interpreted by the HTTP server and HTTP client (browser)	SMTP messages are destined to be read by humans.
HTTP messages are delivered immediately	SMTP messages are stored and forwarded.

Table 11.11 HTTP and SMTP Comparisons

SUMMARY

The heart of intranets and the Internet is the Domain Name System (DNS), the way in which computers can contact each other and do things such as exchange electronic mail, or display Web pages. DNS stands for two things: Domain Name Service and Domain Name Servers. Domain Name System messages are transmitted either as datagrams (UDP) or via stream connection (TCP).

FTP stands for File Transfer Protocol. It's the concept of moving a file from your storage space to your server so others can look at it. The file transfer protocol allows you to connect to a remote computer (host) using an FTP program on your machine, browse a list of files available, retrieve files, and navigate the directory structure of the host system. FTP differs from other client-server applications in that it establishes two connections between the client and the server.

HTTP is a standard Internet protocol that specifies the client/server interaction processes between Web browsers such as Microsoft Internet Explorer and Web servers such as Microsoft Internet Information Services (IIS).

PRACTICE SET

Review Questions

1. Explain the working of DNS?
2. Explain the email delivery process of SMTP?
3. Explain the working of FTP protocol with the help of example?
4. Explain the working of HTTP with the help of example?
5. Compare between HTTP and SMTP.
6. Compare between FTP and HTTP.
7. Compare between FTP and TFTP.
8. Write a short note on Mailbox.
9. Write a short note on Web Browsers.
10. Write a short note on Web Server.
11. Explain Domain Name Space with the help of example.
12. Explain Fully Qualified Domain Name and Partially Qualified Domain Name Space with the help of examples.

Multiple Choice Questions

1. DNS stands for ……..
   A) Double Name Space
   B) Domain Name Switch
   C) Domain Name System
   D) None of the above

2. FTP stands for ……
   A) Film Transport Protocol
   B) First Television Protocol
   C) First Transport Protocol
   D) File Transport Protocol

3. TFTP stands for ……
   A) Television Film Transport Protocol
   B) Trivial First Television Protocol
   C) Television First Transport Protocol
   D) Trivial File Transport Protocol

4. HTTP stands for …….
   A) Hyper Television Transport Protocol
   B) Hollywood Television Transport Protocol
   C) Hyper Text Transport Protocol
   D) None of the above

5. SMTP stands for ……
   A) Simple Money Transport Protocol
   B) Simple Mail Transport Protocol
   C) Both option A) and B)
   D) None of the above

6 WWW stands for ……

A) World Heavyweight Wrestling B) Word Wide Web

C) World Wrestling Web D) World Wide Web

7 Fully Qualified Domain Name also called as ……

A) Partially Qualified Domain Name

B) Partial Domain

C) Absolute Domain Name

D) None of the above

8 There are total ……. Root server all over the world.

A) 11 B) 12 C) 13 D) 14

9 Domain name tree can have ……. Levels

A) 0 B) 127 C) 128 D) Unlimited

10 Label in domain name tree contains maximum …… number of characters.

A) 0 B) 63 C) 127 D) 255

11. …… is a fully qualified domain name.

A) www B) www.viit.ac.in

C) www.viit.ac.in. D) None of the above

* * *

Chapter 12 BOOTP and DHCP

Each computer attached to a TCP/IP network needs to know its IP address before it can send or receive datagram. In addition, a computer needs other information such as the address of a router, the subnet mask to use.

In this chapter, we will also see how a server assigns an IP address to a computer automatically. Such assignment is especially important in environments that permit temporary internet connections or where computers move from one network to another (e.g., an employee with a portable computer moves from one location in a company to another).

OBJECTIVES OF THIS CHAPTER

After completing this chapter, the student should be able to:

- define need of Auto-configuration protocols, such as BOOTP and DHCP.
- explain the working of BOOTP protocol
- explain the working of DHCP Protocol

12.1 INTRODUCTION

Diskless machines usually contain a startup program in nonvolatile storage (e.g., in ROM). To minimize cost and keep parts interchangeable, a vendor uses exactly the same program in all machines. Because computers with different IP addresses run the same boot program, the code cannot contain an IP address. Thus, a diskless machine must obtain its IP address from another source. In fact, a diskless computer needs to know much more than its IP address. Usually, the ROM only contains a small startup program, so the diskless computer must also obtain an initial memory image to execute. In addition, each diskless machine must determine the address of a file server on which it can store and retrieve data, and the address of the nearest IP router.

Three parts of information needed by a system in order to be able to communicate on a TCP/IP network: i) an IP address (to uniquely identify the system on the network), ii) a subnet mask (to determine the network and subnet parts of the address) and iii) the address of at least one router (if the system is to be able to communicate with other devices outside of its network subnet). These three values represent the bare minimum of information that must be programmed into each and every device that is to participate in the TCP/IP world.

There are two ways to assign above mention information to the TCP/IP host: Manually or Automatically.

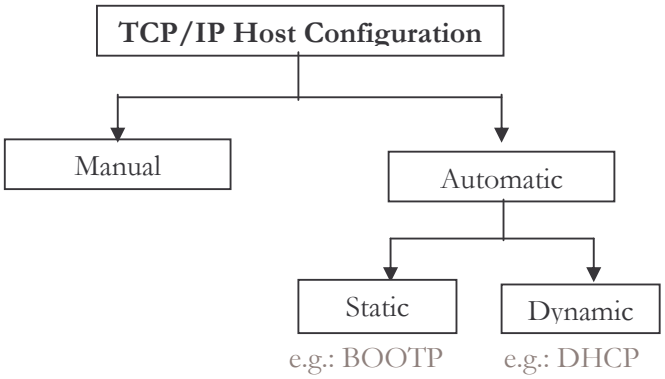


Figure 12.1 TCP/IP Host Configuration Methods

MANUAL TCP/IP CONFIGURATION

When you configure TCP/IP manually on your network, you must enter an IP address, subnet mask, and default gateway on each client computer. This means that users can enter incorrect or invalid values in any of above mentioned three fields (IP address, subnet mask, and default gateway) instead of the valid values from the network administrator. Using an incorrect value can lead to network problems that can be very difficult to trace to the source. Also, manually typing IP address, subnet mask, or default gateway can lead to typographical errors, causing communication problems due to an incorrect default gateway or subnet mask, or problems associated with duplicate IP addresses.

AUTOMATIC TCP/IP CONFIGURATION

Configuring TCP/IP host automatically means that users no longer need to acquire IP address from an administrator. Instead, the server (DHCP, or BOOTP, or RARP) automatically supplies IP address information to the client. It also ensures that network clients use correct configuration information, thereby eliminating common source of network problems. Automatic configuration server updates client configuration information to reflect changes in network structure and the relocation of users to other physical networks, without the need to manually reconfigure client IP addresses.

Manual TCP/IP Configuration	Automatic TCP/IP Configuration
IP address entered manually on each client computer	IP address are supplied automatically to client computers
Possibility of entering incorrect or invalid IP address.	Ensures that clients always use correct configuration information
Incorrect configuration can lead to communication and network problems	Elimination of common source of network problems
Administrative overload if change in network structure or clients frequently moves between different subnets	Client configuration updated automatically to reflect changes in network structure.

Table 12.1 Manual Vs Automatic TCP/IP Configuration

12.2 Why not RARP?

The RARP protocol has three drawbacks. First, because RARP operates at a low level, using it requires direct access to the network hardware. Thus, it may be difficult or impossible for an application programmer to build a server. Second, because RARP uses a computer's hardware address to identify the machine, it cannot be used on networks that dynamically assign hardware addresses. Third, although RARP requires a packet exchange between a client machine and a computer that answers its request, the reply contains only one small piece of information: the client's 4-octet IP address. This drawback is especially annoying on networks like an Ethernet that enforce a minimum packet size because additional information could be sent in the response at no additional cost.

To overcome some of the drawbacks of RARP, researchers developed the Bootstrap Protocol (BOOTP). Later, the Dynamic Host Configuration Protocol (DHCP) was developed as a successor to BOOTP. Because it uses UDP and IP, BOOTP can be implemented with an *application program*. However, BOOTP is more efficient than RARP because a single BOOTP message specifies many items needed at startup, including a computer's IP address, the address of a router, and the address of a server.

12.3 Bootstrap Protocol (BOOTP)

The BOOTP protocol was originally developed as a mechanism to enable diskless hosts to be remotely booted over a network. It allows a minimum IP protocol stack with no configuration information to obtain enough information to begin the process of downloading the necessary boot code. BOOTP does not define how the downloading is done, but this process typically uses TFTP.

BOOTP is commonly used mechanism to deliver configuration information to a client that has not been manually configured. Suppose client machine A wants to use BOOTP to find bootstrap information (including its IP address) and suppose machine B is the server on the same physical network that will answer the request. Because machine A does not know machine B's IP address or the IP address of the network, it must broadcast its initial BOOTP request using the IP broadcast address (that is, 255.255.255.255). What about the reply? Can machine B send a directed reply? No, not usually. Although it may not be obvious, machine B may need to use the broadcast address for its reply, even though it knows machine A's IP address. To see why, consider what happens if an application program on machine B attempts to send a datagram using machine A's IP address. After routing the datagram, IP software on machine B will pass the datagram to the network interface software. The interface software must map the next hop IP address to a corresponding hardware address. However, because machine A has not yet received the BOOTP reply, it does not recognize its IP address, so it cannot answer machine B's ARP request.

The BOOTP process involves the following steps:

1. The client determines its own hardware address; this is normally in a ROM on the hardware.

2. A BOOTP client sends its hardware address in a UDP datagram to the server.

If the client knows its IP address and/or the address of the server, it should use them, but in general BOOTP clients have no IP configuration data at all. If the client does not know its own IP address, it uses 0.0.0.0. If the client does not know the server's IP address, it uses the broadcast address or limited broadcast address (255.255.255.255). The UDP port number is 67.

3. The server receives the datagram and looks up the hardware address of the client in its configuration file, which contains the client's IP address. The server fills in the remaining fields in the UDP datagram and returns it to the client using UDP port 68. (Well-known UDP Port 67 is used by BOOTP Server and Well-know UDP port 68 is used by UDP Clients). One of three methods may be used to do this:

- If the client knows its own IP address (it was included in the BOOTP request), then the server returns the datagram directly to this address. It is likely that the ARP cache in the server's protocol stack will not know the hardware address matching the IP address. ARP will be used to determine it as normal.
- If the client does not know its own IP address (it was 0.0.0.0 in the BOOTP request), then the server must concern itself with its own ARP cache.
- ARP on the server cannot be used to find the hardware address of the client because the client does not know its IP address and so cannot reply to an ARP request.
  There are two possible solutions:
    - If server has a mechanism for directly updating its own ARP cache without using ARP itself, it does so and then sends the datagram directly.
    - If server cannot update its own ARP cache, it must send a broadcast reply.

4. When it receives the reply, the BOOTP client will record its own IP address (allowing it to respond to ARP requests) and begin the bootstrap process.

BOOTP Transmission Policy

BOOTP places all responsibility for reliable communication on the client. We know that because UDP uses IP for delivery, messages can be delayed, lost, delivered out of order, or duplicated. Furthermore, because IP does not provide a checksum for data, the UDP datagram could arrive with some bits corrupted. To guard against corruption, BOOTP requires that UDP use checksums. It also specifies that requests and replies should be sent with the ***do not fragment*** bit set to accommodate clients that have too little memory to reassemble datagrams. BOOTP is also constructed to allow multiple replies; it accepts and processes the first. To handle datagram loss, BOOTP uses the conventional technique of ***timeout and retransmission.*** When the client transmits a request, it starts a timer. If no reply arrives before the timer expires, the client must retransmit the request.

The BOOTP Message Format

To keep an implementation as simple as possible, BOOTP messages have fixed length fields, and replies have the same format as requests. The machine that sends a BOOTP request is referred as the client and any machine that sends a reply is referred as a server.

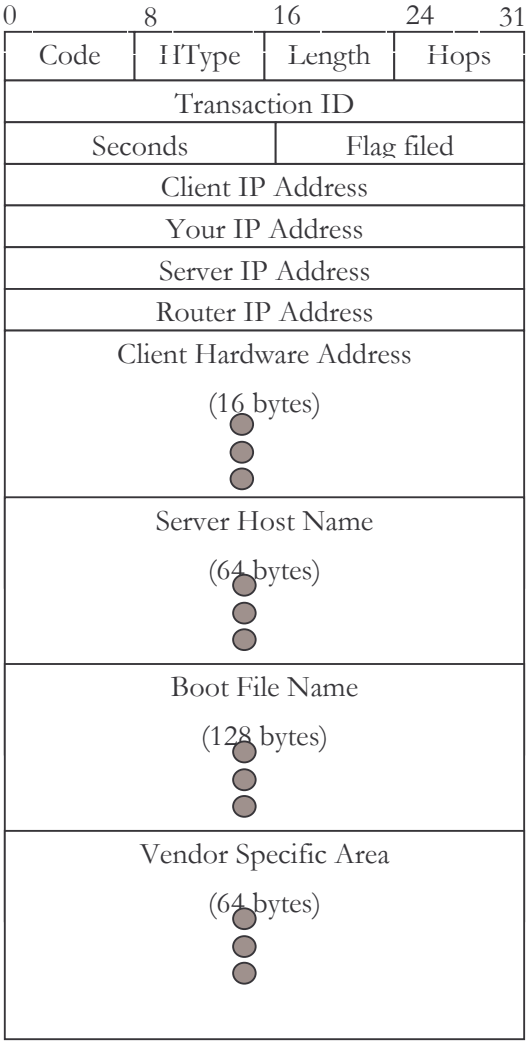


Figure 12.2 BOOTP Message Format

CODE: Indicates a request or a reply.

1 Request
2 Reply

HTYPE: The type of hardware, for example: 1 Ethernet, 6 IEEE 802 Networks.

LENGTH: Hardware address length in bytes. For example, Ethernet and token-ring both use 6 bytes for hardware address.

Hops

The client sets this to 0. It is incremented by a router that relays the request to another server and is used to identify loops. RFC 951 suggests that a value of 3 indicates a loop.

Transaction ID

A random number used to match this boot request with the response it generates.

Seconds

Set by the client. It is the elapsed time in seconds since the client started its boot process.

Flags field

The most significant bit of the flags field is used as a broadcast flag. All other bits must be set to zero; they are reserved for future use.

Normally, BOOTP servers attempt to deliver BOOTREPLY messages directly to a client using unicast delivery. The destination address in the IP header is set to the BOOTP your IP address and the MAC address is set to the BOOTP client hardware address. If a host is unable to receive a unicast IP datagram until it knows its IP address, then this broadcast bit must be set to indicate to the server that the BOOTREPLY must be sent as an IP and MAC broadcast. Otherwise this bit must be set to zero.

Client IP address

Set by the client. Either its known IP address or 0.0.0.0.

Your IP address

Set by the server if the client IP address field was 0.0.0.0.

Server IP address

Set by the server. BOOTP Server IP address.

Router IP address

This is the address of a BOOTP relay agent, not a general IP router to be used by the client. It is set by the forwarding agent when BOOTP forwarding is being used.

Client hardware address

Set by the client and used by the server to identify which registered client is booting.

BOOTP can be used from a client that already knows its IP address (e.g., to obtain boot file information). A client that knows its IP address places it in the ***CLIENT IP ADDRESS*** field; other clients use zero. If the client's IP address is zero in the request, a server returns the client's IP address in the ***YOUR IP ADDRESS*** field.

Server host name

Optional server host name terminated by X'00'.

If a client knows the name or address of a specific server from which it wants information, it can fill in the ***SERVER IP ADDRESS*** or ***SERVER HOST NAME*** fields. If these fields are nonzero, only the server with matching name/address will answer the request; if they are zero, any server that receives the request will reply.

Boot file name

The client either leaves this null or specifies a generic name, such as router indicating the type of boot file to be used. The server returns the fully qualified filename of a boot file suitable for the client. The value is terminated by X'00'.

Vendor-specific area

Optional vendor-specific area. It is recommended that clients always fill the first four bytes with a “magic cookie.” If a vendor-specific magic cookie is not used the client

should use 99.130.83.99 followed by an end tag (255) and set the remaining bytes to zero. The vendor-specific area can also contain BOOTP Vendor extensions. These are options that can be passed to the client at boot time along with its IP address. For example, the client could also receive the address of a default router, the address of a domain name server and a subnet mask.

BOOTP Forwarding

The BOOTP client uses the limited broadcast address (255.255.255.255) for BOOTP requests, which requires the BOOTP server to be on the same subnet as the client. BOOTP forwarding is a mechanism for routers to forward BOOTP requests across subnets. It is a configuration option available on most routers. The router configured to forward BOOTP requests is known as a BOOTP relay agent. A router will normally discard any datagrams containing illegal source addresses, such as 0.0.0.0, which is used by a BOOTP client. A router will also generally discard datagrams with the limited broadcast (255.255.255.255) destination address. However, a BOOTP relay agent will accept such datagrams from BOOTP clients on port 67.

The process carried out by a BOOTP relay agent on receiving a BOOTPREQUEST is as follows:

1. When the BOOTP relay agent receives a BOOTPREQUEST, it first checks the hops field to check the number of hops already completed, in order to decide whether to forward the request. The threshold for allowable number of hops is normally configurable.

2. If the relay agent decides to relay the request, it checks the contents of the router IP address field. If this field is zero, it fills this field with the IP address of the interface on which the BOOTPREQUEST was received. If this field already has an IP address of another relay agent, it is not touched.

3. The value of the hops field is incremented.

4. The relay agent then forwards the BOOTPREQUEST to one or more BOOTP servers. The address of the BOOTP server(s) is preconfigured at the relay agent. The BOOTPREQUEST is normally forwarded as a unicast frame, although some implementations use broadcast forwarding.

5. When the BOOTP server receives the BOOTPREQUEST with the non-zero router IP address field, it sends an IP unicast BOOTREPLY to the BOOTP relay agent at the address in this field on port 67.

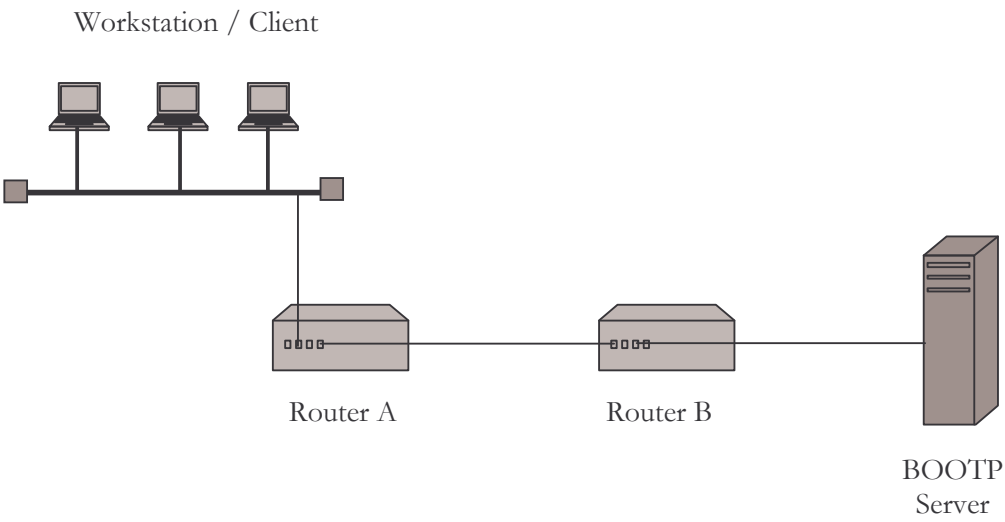


Figure 12.3 Router as a BOOTP Forwarding Agent

6. When the BOOTP relay agent receives the BOOTREPLY, the Htype, length and client hardware address fields in the message supply sufficient link-layer information to return the reply to the client. The relay agent checks the broadcast flag. If this flag is set, the agent forwards the BOOTPREPLY to the client as a broadcast. If the broadcast flag is not set, the relay agent sends a reply as a unicast to the address specified in your IP address.

When a router is configured as a BOOTP relay agent, the BOOTP forwarding task is considerably different to the task of switching datagrams between subnets normally carried out by a router. Forwarding of BOOTP messages can be considered to be receiving BOOTP messages as a final destination, then generating new BOOTP messages to be forwarded to another destination

BOOTP uses a two-step bootstrap procedure. It does not provide clients with a memory image - it only provides the client with information needed to obtain an image. The client then uses a second protocol (e.g., TFTP) to obtain the memory image. While the two-step procedure many seem unnecessary, it allows a separation of configuration and storage procedure. A BOOTP server does not need to run on the same machine that stores memory images. In fact, the BOOTP server operates from a simple database that only knows the names of memory images. Keeping configuration separate from storage is important because it allows administrators to configure sets of machines so they act identically or independently. The ***BOOT FILE NAME*** field of a BOOTP message use for this purpose. Suppose an administrator has several workstations with different hardware architectures, and suppose that when users boot one of the workstations, they either choose to run UNIX or a local operating system. Because the set of workstations includes multiple hardware architectures, no single memory image will operate on all machines. To accommodate such diversity, BOOTP allows the ***BOOT FILE NAME*** field in a request to contain a generic name like "UNIX" which means, "I want to boot the UNIX operating system for this machine." Or "windows," which means, "I want to boot the Windows operating system for this machine." The BOOTP server consults its configuration database to map the generic name into a specific file name that contains the memory image appropriate for the client hardware, and returns the specific (i.e., fully qualified) name in its reply. Of course, the configuration database also allows completely automatic bootstrapping in which the client places zeros in the ***BOOT FILE NAME*** field, and BOOTP selects a memory image for the machine. The advantage of the

automatic approach is that it allows users to specify generic names that work on any machine; they do not need to remember specific file names or hardware architectures.

Need for Dynamic Configuration

BOOTP was designed for a relatively static environment in which each host has a permanent network connection. A manager creates a BOOTP configuration file that specifies a set of BOOTP parameters for each host. The file does not change frequently because the configuration usually remains stable. Typically, a configuration continues unchanged for weeks, months or even years.

With the advent of wireless networking and portable computers such as laptops and notebooks, it has become possible to move a computer from one location to another quickly and easily. BOOTP does not adapt to such situations because configuration information cannot be changed quickly. BOOTP only provides a static mapping from a host identifier to parameters for the host. Administrator must enter a set of parameters for each host, and then store the information in a BOOTP server configuration file; BOOTP does not include a way to dynamically assign values to individual machines.

Static parameter assignment works well if computers remain at fixed locations and a manager has sufficient IP addresses to assign each computer a unique IP address. However, in cases where computers move frequently or the number of physical computers exceeds the number of available IP host addresses, static assignment incurs excessive overhead.

To understand how the number of computers can exceed the number of available IP addresses, consider a LAN in a college that has been assigned a */24* (class C default subnet Mask, that is, 255.255.255.0) address that allows up to 254 hosts. Assume that because the computer laboratory only has seats for 60 students, the college schedules labs at five different times during the week to accommodate up to 300 students. Further assume that each student carries a personal laptop that they use in the computer lab. At any given time, the network has at most 60 active computers. However, because the network address can accommodate at most 254 hosts, an administrator cannot assign a unique address to each computer. Thus, although resources such as physical connections limit the number of simultaneous connections, the number of potential computers that can use the facility is high. A system is inadequate if it requires an administrator to change the server's configuration file before a new computer can be added to the network; an automated mechanism is required.

12.4 DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP)

DHCP is based upon the Bootstrap Protocol (BOOTP), BOOTP is use for automatically delivering configuration information from a BOOTP server to BOOTP clients on boot-up.

In DHCP environment, a computer is designated as the DHCP server. All of the other computers on the network—at least, those that need an IP address—will be DHCP clients (computers that already have a permanently set IP address don't need to participate). The network administrator needs to initially configure the DHCP server. It involves assigning the DHCP server a block of IP address numbers that it can dispense to nodes that need IP addresses.

The DHCP Message Format

DHCP Message format, which is similar to BOOTP Message format, is shown in the following figure:

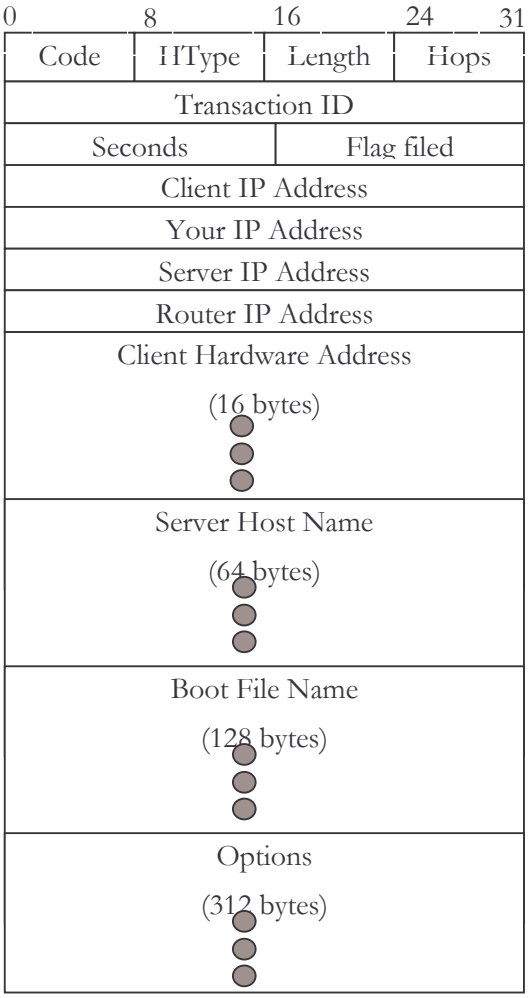


Figure 12.4: DHCP Message Format

CODE: Indicates a request or a reply.

1 Request
2 Reply

HTYPE: The type of hardware, for example: 1 Ethernet, 6 IEEE 802 Networks.

LENGTH: Hardware address length in bytes. For example, Ethernet and token-ring both use 6 bytes for hardware address.

Hops

The client sets this to 0. It is incremented by a router that relays the request to another server and is used to identify loops. RFC 951 suggests that a value of 3 indicates a loop.

Transaction ID

A random number used to match this boot request with the response it generates.

Seconds

Set by the client. It is the elapsed time in seconds since the client started its boot process.

Flags field

The most significant bit of the flags field is used as a broadcast flag. All other bits must be set to zero; they are reserved for future use.

Normally, DHCP servers attempt to deliver DHCP messages directly to a client using unicast delivery. The destination address in the IP header is set to the DHCP your IP address and the MAC address is set to the DHCP client hardware address. If a host is unable to receive a unicast IP datagram until it knows its IP address, then this broadcast bit must be set to indicate to the server that the DHCP reply must be sent as an IP and MAC broadcast. Otherwise this bit must be set to zero.

Client IP address

Set by the client. Either its known IP address or 0.0.0.0.

Your IP address

Set by the server if the client IP address field was 0.0.0.0.

Server IP address

Set by the server. DHCP Server's own IP address.

Router IP address

This is the address of a BOOTP relay agent, not a general IP router to be used by the client. It is set by the forwarding agent when BOOTP forwarding is being used.

Client hardware address

Set by the client and used by the server to identify client.

Server host name

Optional server host name terminated by X'00'.

If a client knows the name or address of a specific server from which it wants information, it can fill in the ***SERVER IP ADDRESS*** or ***SERVER HOST NAME*** fields. If these fields are nonzero, only the server with matching name/address will answer the request; if they are zero, any server that receives the request will reply.

Boot file name

The client either leaves this null or specifies a generic name, such as router indicating the type of boot file to be used. The server returns the fully qualified filename of a boot file suitable for the client. In a DHCPDISCOVER request this is set to null. The server returns a fully qualified directory path name in a DHCPOFFER request. The value is terminated by X'00'.

Options

It is recommended that clients always fill the first four bytes with a "magic cookie." If a vendor-specific magic cookie is not used the client should use 99.130.83.99 followed by an end tag (255) and set the remaining bytes to zero.

DHCP Message Types

DHCP messages fall into one of the following categories:

Message	Description
DHCPDISCOVER	Broadcast by a client to find available DHCP servers.
DHCPOFFER	Response from a server to a DHCPDISCOVER and offering IP address and other parameters.
DHCPREQUEST	Message from a client to servers that does one of the following: • Requests the parameters offered by one of the servers and declines all other offers • Verifies a previously allocated address after a system or network change (a reboot for example). • Requests the extension of a lease on a particular address
DHCPACK	Acknowledgement from server to client with parameters, including IP address.
DHCPNACK	Negative acknowledgement from server to client, indicating that the client's lease has expired or that a requested IP address is incorrect.
DHCPDECLINE	Message from client to server indicating that the offered address is already in use.
DHCPRELEASE	Message from client to server canceling remainder of a lease and relinquishing network address.
DHCPINFORM	Message from a client that already has an IP address (manually configured for example), requesting further configuration parameters from the DHCP server.

Table 12.2 DHCP Message Types

ADDRESS AQUASITION USING DHCP

Suppose the DHCP server has a block of network addresses from which it can satisfy requests for new addresses. Each server also maintains a database of allocated addresses and leases in permanent local storage.

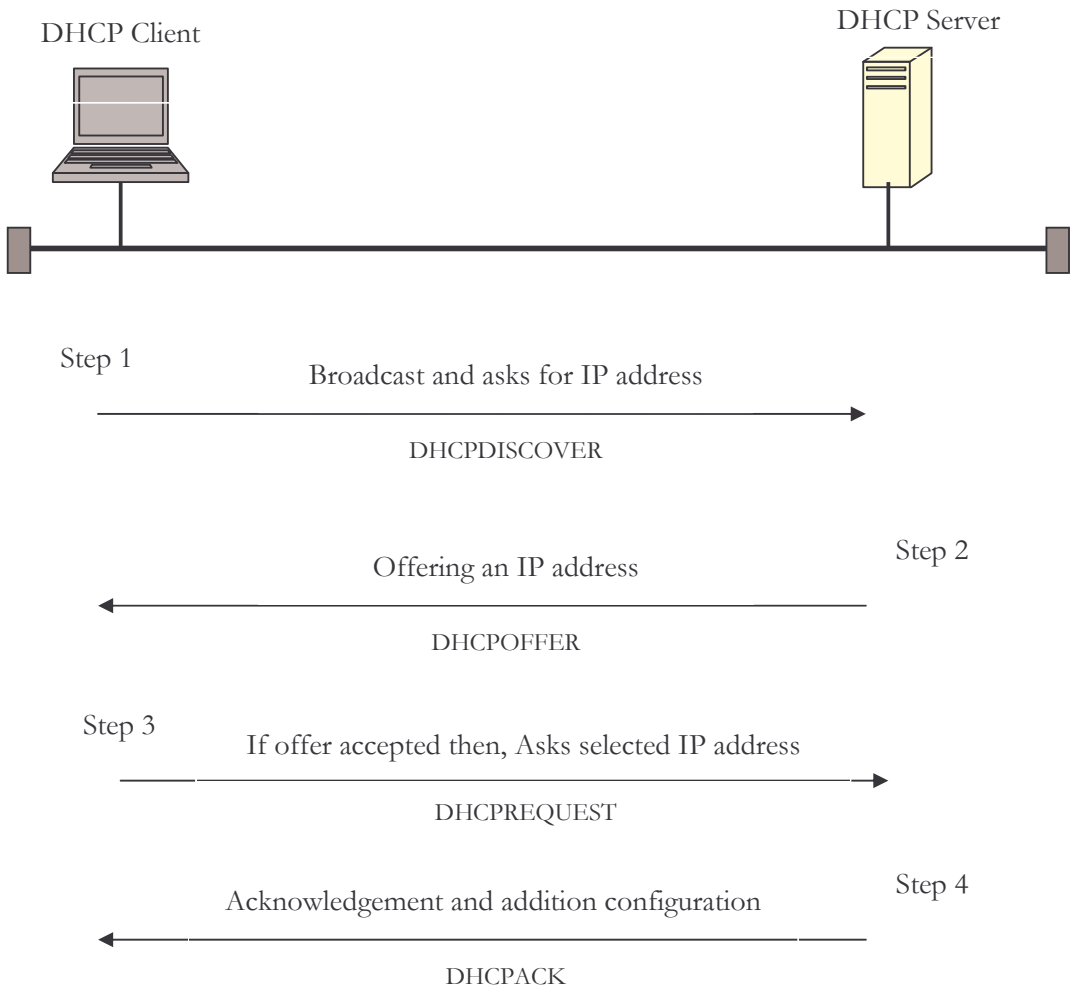


Figure 12.5 DHCP Client Address Acquisition Process

Following steps describes DHCP client address acquisition process from DHCP server:

Step 1:

When a client first boots, it need to acquire an IP address from DHCP server. To start acquiring an IP address, the client first contacts all DHCP servers in the local network. To do so, the client broadcasts a ***DHCPDISCOVER*** message.

Because the protocol is an extension of BOOTP, the client sends the ***DHCPDISCOVER*** message in a UDP datagram with the destination port set to UDP port 67.

Step 2:

All DHCP servers on the local net receive the message, and those servers that have been programmed to respond to the particular client send a ***DHCPOFFER*** message. Thus, a client may receive zero or more responses.

The servers may record the address as offered to the client to prevent the same address being offered to other clients in the event of further DHCPDISCOVER messages being received before the first client has completed its configuration.

Step 3:

The client receives one or more DHCPOFFER messages from one or more servers. The client collects ***DHCPOFFER*** responses from DHCP servers. Each offer contains configuration information for the client along with an IP address that the server is offering to lease to the client. The client must choose one of the responses (for example, the response that arrived first), and negotiate with the server for a lease. To do so, the client sends the server a ***DHCPREQUEST*** message.

Step 4:

The servers receive the DHCPREQUEST broadcast from the client. Those servers not selected by the DHCPREQUEST message use the message as notification that the client has declined that server's offer. The server selected in the DHCPREQUEST message responds with a DHCPACK message containing the configuration parameters for the requesting client.

The client receives the DHCPACK message with configuration parameters. The client performs a final check on the parameters, for example with ARP for allocated network address; at this point client is configured. That is, DHCP client acquired IP address and other configuration information successfully.

If the client detects any problem with the configuration, for example, IP address offered is already used on the network, then client must have to discard those setting as well as notify DHCP server about the IP address conflict. To do so, the client sends a **DHCPDECLINE** message to the server and restarts the configuration process. On receipt of a DHCPDECLINE, the server must mark the offered IP address as unavailable. If the client receives a DHCPNAK message, the client restarts the configuration process. The client may choose to relinquish its lease on a network address by sending a DHCPRELEASE message to the server.

DHCP Lease Renewal Process

DHCP lease is the duration for which a DHCP server loans an IP address to a DHCP client.

Following step describes DHCP lease renewal or expiry process:

Step 1:

When a DHCP server sends the DHCPACK message to a client with IP address and configuration parameters, it also registers the start of the lease time for that IP address. DHCP server send lease time and two other timers called T1 and T2, with DHCPACK message to DHCP client. Now client is authorized to use the given IP address for the duration of the lease time.

When client is configured with received configuration successfully, client also starts the timer T1 and T2. Note that T1 must be less than T2, and T2 must be less than lease time. By default the values of T1 and T2 are; T1 = 0.5 × lease time, and T2 = 0.875 × lease time.

Step 2:

When T1 timer expires, DHCP client sends DHCPREQUEST message in using unicast way to DHCP server for renewal of its lease duration. The server would generally respond with a DHCPACK message indicating the new lease time, and timers T1 and T2 are reset accordingly. The DHCP server also resets its record of the lease time. In normal circumstances, DHCP client would continually renew its lease in this way indefinitely, without the lease ever expiring.

Step 3:

If no DHCPACK is received until timer T2 expires; now this time DHCP client send DHCPREQUEST message in using broadcast way for renewal of its duration to the DHCP server. This request can be confirmed by a DHCPACK message from any DHCP server on the network.

Step 4:

Generally DHCP clients are honest. If the client does not receive a DHCPACK message after its lease has expired, it has to stop using its current network (that is, TCP/IP) configuration. It drops the current network configuration and begins the network address acquisition process.

Summary

The Bootstrap Protocol, BOOTP, provides an alternative to RARP for a computer that needs to acquire its IP address. BOOTP is more general than RARP because it uses UDP, making it possible to extend bootstrapping across a router. BOOTP allows administrators to establish a configuration database that maps a generic name, like "UNIX" into the fully qualified file name that contains a memory image appropriate for the client hardware.

The Dynamic Host Configuration Protocol (DHCP) extends BOOTP in several ways. Most important, DHCP permits a server to allocate IP addresses automatically or dynamically. Dynamic allocation is necessary for environments such as a wireless network where computers can attach and detach quickly. To use DHCP, a computer becomes a client. The computer broadcasts a request for

DHCP servers, selects one of the offers it receives, and exchanges messages with the server to obtain a lease on the advertised IP address.

Practice set

Review Questions

1. In BOOTP message format, there are two fields for IP address and one filed for Boot image. If the client leaves its IP address field empty, the server returns the client's IP address in the second field. If the client leaves the boot file name field empty, the server replaces it with an explicit name. Why?
2. When BOOTP client receives a reply via hardware broadcast, how does it knows that, reply is intended for another client on network?
3. Explain address acquisition process of DHCP.
4. Explain address renewal process in DHCP.
5. Why RARP is not used to assign IP address to host on network?
6. Explain BOOTP process.
7. Explain BOOTP Forwarding.
8. Explain DHCP Message format.
9. compare automatic and manual configurations
10. Why dynamic host configuration is required?

Multiple Choice Questions

1. BOOTP stands for ……..
   A) BOOTStrap Protocol
   B) Booting Privacy
   C) Dynamic Host Configuration Protocol
   D) None of the above

2. DHCP stands for …….
   A) Dynamic Home Configuration Protocol
   B) Dual Host Configuration Protocol
   C) Dynamic Host Configuration Protocol
   D) All of the above

3. T1 used in DHCPACK message contains value equal to ……
   A) 0.2 * lease time
   B) 0.5 * lease time
   C) 0.875 * lease time
   D) None of the above

4. T2 used in DHCPACK message contains value equal to ……
   A) 0.2 * lease time
   B) 0.5 * lease time
   C) 0.875 * lease time
   D) None of the above

5. …… message sent by server in response to a DHCPDISCOVER and offering IP address and other parameters.
   A) DHCPACK
   B) DHCPOFFER
   C) DHCPNACK
   D) DHCPREQUEST

6. …… message means acknowledgement from DHCP server to DHCP client with parameters, including IP address.
   A) DHCPACK
   B) DHCPOFFER
   C) DHCPNACK
   D) DHCPREQUEST

7. If DHCP client does not knows its IP address then in client address field of the message contains …….address.
A) 10.0.0.0 B) 1.1.1.1 C) 0.0.0.0 D) 255.255.255.255

8. If a vendor-specific magic cookie is not used the client uses …….. address.
A) 0.0.0.0 B) 1.1.1.1 C) 255.255.255.255 D) 99.130.83.99

* * *