



S.Y. B.Sc. (IT)
SEMESTER - III (CBCS)

COMPUTER NETWORKS

SUBJECT CODE : USIT303

Prof. Suhas Pednekar

Vice-Chancellor,
University of Mumbai,

Prof. Ravindra D. Kulkarni

Pro Vice-Chancellor,
University of Mumbai,

Prof. Prakash Mahanwar

Director,
IDOL, University of Mumbai,

Programme Co-ordinator : Shri Mandar Bhanushe

Head, Faculty of Science and Technology,
IDOL, University of Mumbai, Mumbai

Course Co-ordinator : Ms. Gouri Sawant

Asst. Professor, B.Sc. I.T.
IDOL, University of Mumbai, Mumbai

Course Writers

: Mr. Ankush Kudale

Assistant Professor,
Sinhgad Institute of Management,
Vadgaon BK., Pune

: Dr. Milind Godase

Professor,
Sinhgad Institute of Management,
Vadgaon BK., Pune

: Ms. Pradnya Patil

Assistant Professor, K.J. Somaiya Institute of
Engineering and Information Technology,
Mumbai

: Ms. Jyotika D. Chheda

Assistant Professor,
Mulund College of Commerce,
Mulund (W), Mumbai

August 2021, Print - I

Published by

: Director
Institute of Distance and Open Learning ,
University of Mumbai,
Vidyanagari, Mumbai - 400 098.

DTP Composed and
Printed by

: Mumbai University Press
Vidyanagari, Santacruz (E), Mumbai - 400098

CONTENTS

Unit No.	Title	Page No.
Unit - I		
1.	Introduction	01
2.	Network Models	17
3.	Introduction to Physical Layer	33
4.	Digital and Analog Transmission	50
Unit - II		
5.	Bandwidth Utilization : Multiplexing and Spectrum Spreading	78
6.	Transmission Media and Switching	100
7.	Introduction to Data Link Layer	131
Unit - III		
8.	Data Link Control	156
9.	Media Access Control	175
10.	Wireless Lan, Connecting Devices and Virtual Lan	195
Unit - IV		
11.	Introduction to Network Layer	212
12.	Network Layer Protocols	232
13.	Unicast Routing	251
14.	Next Generation IP	274
Unit - V		
15.	Introduction to Transport Layer	291
16.	Standard Client - Server Protocols	326



SYLLABUS
B.SC. INFORMATION TECHNOLOGY
SEMESTER - III, (CBCS)
COMPUTER NETWORKS

Unit	Details
I	Introduction: Data communications, networks, network types, Internet history, standards and administration. Network Models: Protocol layering, TCP/IP protocol suite, The OSI model. Introduction to Physical layer: Data and signals, periodic analog signals, digital signals, transmission impairment, data rate limits, performance. Digital and Analog transmission: Digital-to-digital conversion, analog-to-digital conversion, transmission modes, digital-to-analog conversion, analog-to-analog conversion.
II	Bandwidth Utilization: Multiplexing and Spectrum Spreading: Multiplexing, Spread Spectrum Transmission media: Guided Media, Unguided Media Switching: Introduction, circuit switched networks, packet switching, structure of a switch. Introduction to the Data Link Layer: Link layer addressing, Data Link Layer Design Issues, Error detection and correction, block coding, cyclic codes, checksum, forward error correction, error correcting codes, error detecting codes.
III	Data Link Control: DLC services, data link layer protocols, HDLC, Point-to-point protocol. Media Access Control: Random access, controlled access, channelization, Wired LANs – Ethernet Protocol, standard ethernet, fast ethernet, gigabit ethernet, 10 gigabit ethernet, Wireless LANs: Introduction, IEEE 802.11 project, Bluetooth, WiMAX, Cellular telephony, Satellite networks. Connecting devices and Virtual LANs.
IV	Introduction to the Network Layer: Network layer services, packet switching, network layer performance, IPv4 addressing, forwarding of IP packets, Internet Protocol, ICMPv4, Mobile IP Unicast Routing: Introduction, routing algorithms, unicast routing protocols. Next generation IP: IPv6 addressing, IPv6 protocol, ICMPv6 protocol, transition from IPv4 to IPv6.
V	Introduction to the Transport Layer: Introduction, Transport layer protocols (Simple protocol, Stop-and-wait protocol, Go-

	Back-n protocol, Selective repeat protocol, Bidirectional protocols), Transport layer services, User datagram protocol, Transmission control protocol, Standard Client0Server Protocols: World wide-web and HTTP, FTP, Electronic mail, Telnet, Secured Shell, Domain name system.
--	--

Book References :

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Data Communication and Networking	Behrouz A. Forouzan	Tata McGraw Hill	Fifth Edition	2013
2.	TCP/IP Protocol Suite	Behrouz A. Forouzan	Tata McGraw Hill	Fourth Edition	2010
3.	Computer Networks	Andrew Tanenbaum	Pearson	Fifth	2013



INTRODUCTION

Unit Structure

- 1.0 Objectives
- 1.1 Introduction
- 1.2 Data communications
- 1.3 Networks
- 1.4 Types of Network
- 1.5 Internet history
- 1.6 Internet Standards
- 1.7 Internet Administration
- 1.8 Review questions
- 1.9 Summary
- 1.10 References

1.0 OBJECTIVES:

This chapter would make you understand the following concepts

- What is data communication?
- What is network? Types of network.
- Brief history of internet
- Different standards and administration of the internet.

1.1 INTRODUCTION

Communication is sharing information or providing entertainment by speaking, writing or other methods. Probably the most important type of communication is personal communication, which happens when people make their thoughts and wishes known to each other. There are many methods of communication. We have come a long way from the prehistoric times. In those days methods like smoke signals, certain sounds were used to communicate with each other. Then human speech developed and people began to talk and share their thoughts with one another. Not to mention the present days, the world of electronic communication. This world will have to be reinvented if there was no communication. No business would be done. No parent would understand what his child wants from him. There would be no teaching in classes. We would be worst than a rodent.

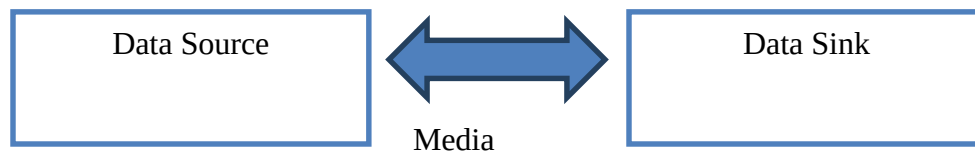
The Concept of Data Communication evolved from sharing the computation power of computer along with various resources available in a computer environment such as printer, hard disk etc. With increasing demand for exchange of information across the globe, the need of data communication has increased in many folds. Data Communication, can be used to transfer or exchange information with in one building, one city, across cities, countries and continents. It is also possible to update and share data at different locations.

By Data Communication we mean the transportation of information from one point to another through a communication media. The word data refers to facts, concepts, and instructions presented in whatever form is agreed upon by parties creating and using the data. In the context of computer information systems data are represented by binary information units (or bits) produced and consume in form of 0s and 1s.

Data communication is exchange of data (in the form of 0s and 1s) between two devices via some form of transmission medium (such as wire cables). Data communication is considered local if the communicating devices are in the same building or a similarly restricted graphical area, and is considered remote if the devices are farther apart.

1.2 DATA COMMUNICATIONS

The main components of data communication are data sources, data sinks and communication media. The source is the originator of information, while sink is the receiver of information. The media is the path through which the information is transported to the sink from the sources. This media could be a telephone wire, a microwave system on a satellite circuit or a fiber optic line. Usually, the media is provided by one or more common communication carriers. The computer equipment is connected to the communication media through a piece of equipment called MODEM. This piece of equipment converted the digital signal to analog and passes it to the communication media through whom they are propagated towards the sink. The sink is similarly connected to the communication media through a modem and receives the propagated signals.



Communication Protocol

All communications between devices require that the devices agree on the format of data. The set of rules defining a format is called a protocol. At the very least, a communications, protocol must define the following

- rate of transmission (in baud or bps)
- whether transmission is to be synchronous or asynchronous
- whether data is to be transmitted in half-duplex or full-duplex mode

In addition, protocols can include sophisticated techniques for detecting and recovering from transmission errors and for encoding and decoding data.

Characteristics of Data Communication

The effectiveness of any data communications system depends upon the following four fundamental characteristics:

- 1. Delivery:** The data should be delivered to the correct destination and correct user.
- 2. Accuracy:** The communication system should deliver the data accurately, without introducing any errors. The data may get corrupted during transmission affecting the accuracy of the delivered data.
- 3. Timeliness:** Audio and Video data has to be delivered in a timely manner without any delay; such a data delivery is called real time transmission of data.
- 4. Jitter:** It is the variation in the packet arrival time. Uneven Jitter may affect the timeliness of data being transmitted

Data Communication Terminology

Data Communication is the process of transferring data from one machine to another machine such that the sender and receiver both interpret the data correctly.

1. Data Channel

In communications the term channel refers to a communications path between two computers or devices. It may refer to the physical medium, such as coaxial cable or to a specific carrier frequency (sub-channel) within a larger channel or wireless medium.

2. Baud

Pronounced bawd, it is the number of signaling elements that occur each second. The term is named after J.M.E. Baudot, the inventor of the Baudot telegraph code.

At slow speeds, only one bit of information(signaling element) is encoded in each electrical change. The baud, therefore, indicates the number of bits per second that are transmitted. For example, 300 baud means that 300 bits are transmitted each second (abbreviated 300 bps). Assuming asynchronous communication, which requires 10 bits per character; this translates to 30 characters per second (cps). For slow rates

you can divide the baud by 10 to see how many characters per second are sent.

At higher speeds, it is possible to encode more than one bit in each electrical change, 4,800 baud may allow 9,600 bits to be sent each second. At high data transfer speeds, therefore, data transmission rates are usually expressed in bits per second (bps) rather than baud. For example, a 9,600 bps modem may operate at only 2,400 baud.

3. Bandwidth

Bandwidth is the amount of data that can be transmitted in a fixed amount of time. For digital devices, the bandwidth is usually expressed in bits per second (bps) or bytes per second. For analog devices, the bandwidth is expressed in cycles per second, or Hertz(Hz).

The bandwidth is particularly important for I/O devices. For example, a fast disk drive can be hampered by a bus with a low bandwidth.

4. Data Transfer Rates

The amount of data transferred per second by a communication channel is known as data transfer rate. It is measured in bits per second (bps)

Components of Data Communication

A Data Communication system has five components as shown in the diagram below:

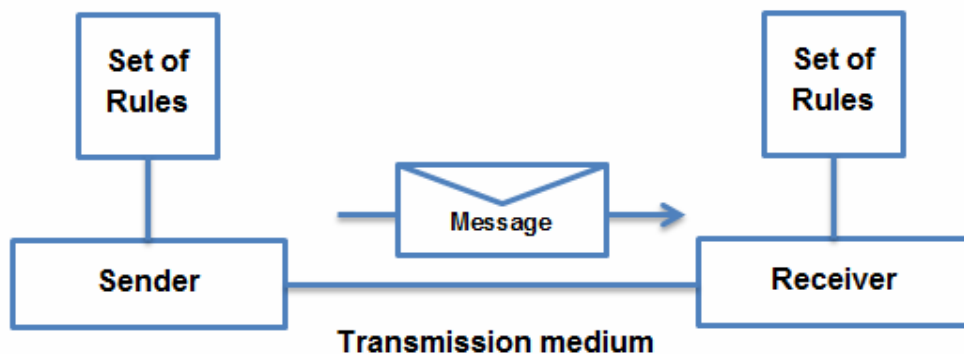


Fig. Components of a Data Communication System

1. Message:

Message is the information to be communicated by the sender to the receiver.

2. Sender

The sender is any device that is capable of sending the data (message).

3. Receiver

The receiver is a device that the sender wants to communicate the data (message).

4. Transmission Medium

It is the path by which the message travels from sender to receiver. It can be wired or wireless and many subtypes in both.

5. Protocol

It is an agreed upon set or rules used by the sender and receiver to communicate data. A protocol is a set of rules that governs data communication. A Protocol is a necessity in data communications without which the communicating entities are like two persons trying to talk to each other in a different language without know the other language

Data Representation

Data is collection of raw facts which is processed to deduce information. There may be different forms in which data may be represented. Some of the forms of data used in communications are as follows:

1. Text

Text includes combination of alphabets in small case as well as upper case. It is stored as a pattern of bits. Prevalent encoding system : ASCII, Unicode

2. Numbers

Numbers include combination of digits from 0 to 9. It is stored as a pattern of bits. Prevalent encoding system : ASCII, Unicode

3. Images

An image is worth a thousand words || is a very famous saying. In computers images are digitally stored. A Pixel is the smallest element of an image. To put it in simple terms, a picture or image is a matrix of pixel elements. The pixels are represented in the form of bits. Depending upon the type of image (black n white or color) each pixel would require different number of bits to represent the value of a pixel. The size of an image depends upon the number of pixels (also called resolution) and the bit pattern used to indicate the value of each pixel. Example: if an image is purely black and white (two color) each pixel can be represented by a value either 0 or 1, so an image made up of 10 x 10 pixel elements would require only 100 bits in memory to be stored. On the other hand an image that includes gray may require 2 bits to represent every pixel value (00 - black, 01 – dark gray, 10 light gray, 11 –white). So the same 10 x 10 pixel image would now require 200 bits of memory to be stored. Commonly used Image formats : jpg, png, bmp, etc

4. Audio

Data can also be in the form of sound which can be recorded and broadcasted. Example: What we hear on the radio is a source of data or information. Audio data is continuous, not discrete.

5. Video

Video refers to broadcasting of data in form of picture or movie.

Data Flow

We devices communicate with each other by sending and receiving data. The data can flow between the two devices in the following ways.

1. Simplex
2. Half Duplex
3. Full Duplex

1. Simplex

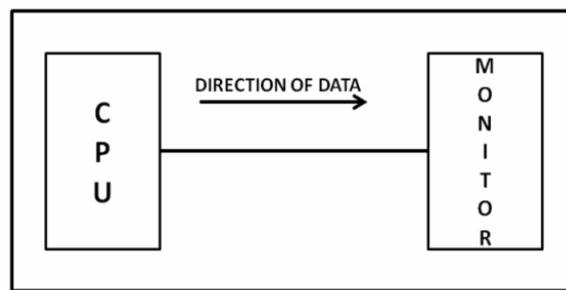


Figure: Simplex mode of communication

In Simplex, communication is unidirectional. Only one of the devices sends the data and the other one only receives the data. Example: in the above diagram: a CPU sends data while a monitor only receives data.

2. Half Simplex

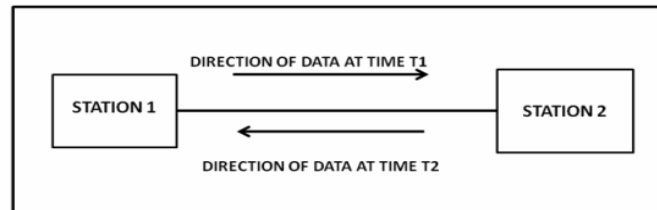


Figure: Half Duplex Mode of Communication

In half duplex both the stations can transmit as well as receive but not at the same time. When one device is sending other can only receive and vice-versa (as shown in figure above.) Example: A walkie-talkie.

3. Full Duplex

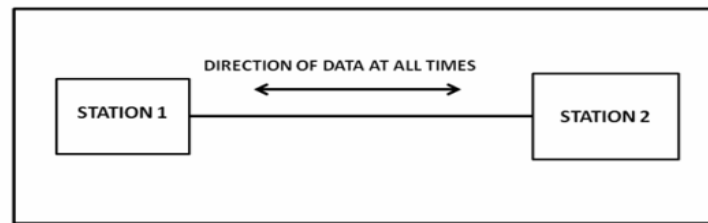


Figure: Full Duplex Mode of Communication

In Full duplex mode, both stations can transmit and receive at the same time. Example: mobile phones

1.3 NETWORKS

A computer network is a group of computers that use a set of common communication protocols over digital interconnections for the purpose of sharing resources located on or provided by the network nodes. The interconnections between nodes are formed from a broad spectrum of telecommunication network technologies, based on physically wired, optical, and wireless radio-frequency methods that may be arranged in a variety of network topologies.

The nodes of a computer network may include personal computers, servers, networking hardware, or other specialized or general-purpose hosts. They are identified by hostnames and network addresses. Hostnames serve as memorable labels for the nodes, rarely changed after initial assignment. Network addresses serve for locating and identifying the nodes by communication protocols such as the Internet Protocol.

Computer networks may be classified by many criteria, including the transmission medium used to carry signals, bandwidth, and communications protocols to organize network traffic, the network size, the topology, traffic control mechanism, and organizational intent.

Computer networks support many applications and services, such as access to the World Wide Web, digital video, digital audio, shared use of application and storage servers, printers, and fax machines, and use of email and instant messaging applications.

1.4 NETWORK TYPES

A computer network is a group of computers linked to each other that enables the computer to communicate with another computer and share their resources, data, and applications.

A computer network can be categorized by their size. A computer network is mainly of four types:

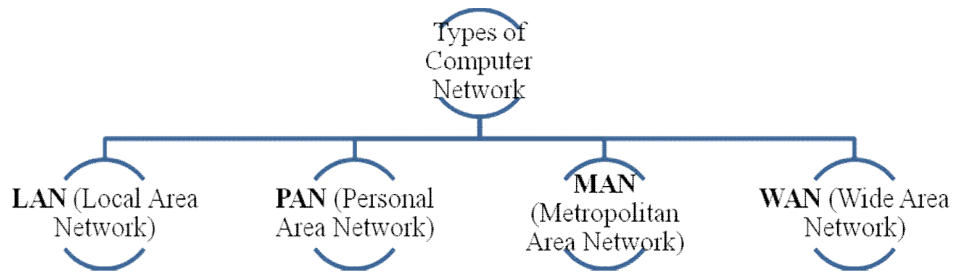


Figure : Types of network

1. LAN (Local Area Network)

- Local Area Network is a group of computers connected to each other in a small area such as building, office.
- LAN is used for connecting two or more personal computers through a communication medium such as twisted pair, coaxial cable, etc.
- It is less costly as it is built with inexpensive hardware such as hubs, network adapters, and Ethernet cables.
- The data is transferred at an extremely faster rate in Local Area Network.
- Local Area Network provides higher security.



Figure LAN (Local Area Network)

2. PAN (Personal Area Network)

- Personal Area Network is a network arranged within an individual person, typically within a range of 10 meters.
- Personal Area Network is used for connecting the computer devices of personal use is known as Personal Area Network.
- **Thomas Zimmerman** was the first research scientist to bring the idea of the Personal Area Network.
- Personal Area Network covers an area of **30 feet**.

- Personal computer devices that are used to develop the personal area network are the laptop, mobile phones, media player and play stations.



Figure PAN (Personal Area Network)

There are two types of Personal Area Network:

➤ **Wired Personal Area Network**

Wireless Personal Area Network: Wireless Personal Area Network is developed by simply using wireless technologies such as WiFi, Bluetooth. It is a low range network.

➤ **Wireless Personal Area Network**

Wired Personal Area Network: Wired Personal Area Network is created by using the USB.

Examples of Personal Area Network:

- **Body Area Network:** Body Area Network is a network that moves with a person. **For example**, a mobile network moves with a person. Suppose a person establishes a network connection and then creates a connection with another device to share the information.
- **Offline Network:** An offline network can be created inside the home, so it is also known as a **home network**. A home network is designed to integrate the devices such as printers, computer, television but they are not connected to the internet.
- **Small Home Office:** It is used to connect a variety of devices to the internet and to a corporate network using a VPN

3. MAN (Metropolitan Area Network)

- A metropolitan area network is a network that covers a larger geographic area by interconnecting a different LAN to form a larger network.
- Government agencies use MAN to connect to the citizens and private industries.

- In MAN, various LANs are connected to each other through a telephone exchange line.
- The most widely used protocols in MAN are RS-232, Frame Relay, ATM, ISDN, OC-3, ADSL, etc.
- It has a higher range than Local Area Network(LAN).

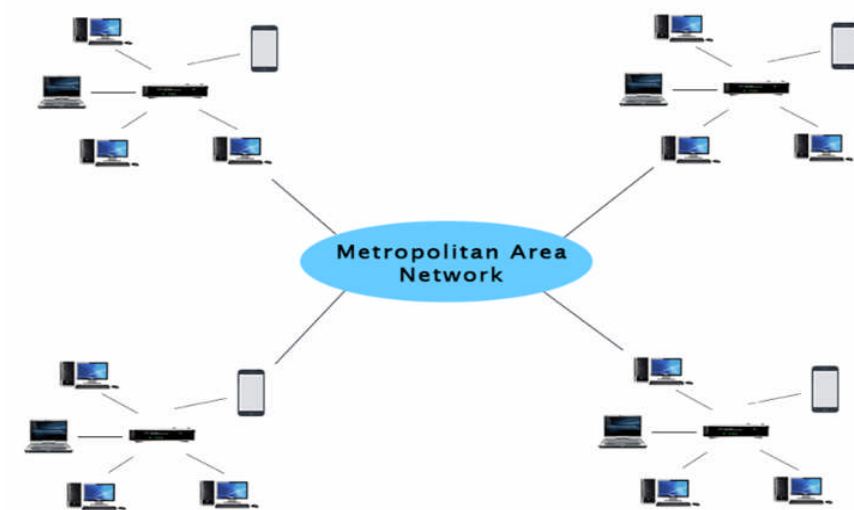


Figure MAN (Metropolitan Area Network)

Uses of Metropolitan Area Network:

- MAN is used in communication between the banks in a city.
- It can be used in an Airline Reservation.
- It can be used in a college within a city.
- It can also be used for communication in the military.

4. WAN (Wide Area Network)

- A Wide Area Network is a network that extends over a large geographical area such as states or countries.
- A Wide Area Network is quite bigger network than the LAN.
- A Wide Area Network is not limited to a single location, but it spans over a large geographical area through a telephone line, fibre optic cable or satellite links.
- The internet is one of the biggest WAN in the world.
- A Wide Area Network is widely used in the field of Business, government, and education.

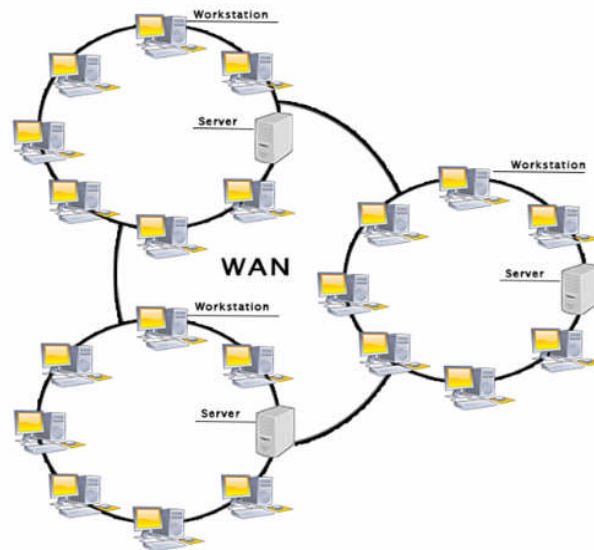


Figure WAN(Wide Area Network)

Examples of Wide Area Network:

Mobile Broadband: A 4G network is widely used across a region or country

Last mile: A telecom company is used to provide the internet services to the customers in hundreds of cities by connecting their home with fiber.

Private network: A bank provides a private network that connects the 44 offices. This network is made by using the telephone leased line provided by the telecom company.

1.5 INTERNET HISTORY

What is Internet?

The Internet (or internet) is the global system of interconnected computer networks that uses the Internet protocol suite (TCP/IP) to communicate between networks and devices. It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web (WWW), electronic mail, telephony, and file sharing.

Brief History of Internet

A network is a group of connected communicating devices such as computers and printers. An internet (note the lowercase letter i) is two or more networks that can communicate with each other. The most notable internet is called the Internet (uppercase letter I), a collaboration of more than hundreds of thousands of interconnected networks. Private individuals as well as various organizations such as government agencies,

schools, research facilities, corporations, and libraries in more than 100 countries use the Internet. Millions of people are users. Yet this extraordinary communication system only came into being in 1969.

In the mid-1960s, mainframe computers in research organizations were standalone devices. Computers from different manufacturers were unable to communicate with one another. The Advanced Research Projects Agency (ARPA) in the Department of Defense (DoD) was interested in finding a way to connect computers so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort.

In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for ARPANET, a small network of connected computers. The idea was that each host computer (not necessarily from the same manufacturer) would be attached to a specialized computer, called an interface message processor (IMP). The IMPs, in turn, would be connected to one another. Each IMP had to be able to communicate with other IMPs as well as with its own attached host.

By 1969, ARPANET was a reality. Four nodes, at the University of California at Los Angeles (UCLA), the University of California at Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah, were connected via the IMPs to form a network. Software called the Network Control Protocol (NCP) provided communication between the hosts.

In 1972, Vint Cerf and Bob Kahn, both of whom were part of the core ARPANET group, collaborated on what they called the Interknitting Project. Cerf and Kahn's landmark 1973 paper outlined the protocols to achieve end-to-end delivery of packets. This paper on Transmission Control Protocol (TCP) included concepts such as encapsulation, the datagram, and the functions of a gateway.

Shortly thereafter, authorities made a decision to split TCP into two protocols: Transmission Control Protocol (TCP) and Internetworking Protocol (IP). IP would handle datagram routing while TCP would be responsible for higher-level functions such as segmentation, reassembly, and error detection. The internetworking protocol became known as TCP/IP.

1.6 INTERNET STANDARDS

An Internet standard is a thoroughly tested specification that is useful to and adhered to by those who work with the Internet. It is a formalized regulation that must be followed. There is a strict procedure by which a specification attains Internet standard status. A specification begins as an Internet draft. An Internet draft is a working document (a

work in progress) with no official status and a six-month lifetime. Upon recommendation from the Internet authorities, a draft may be published as a Request for Comment (RFC). Each RFC is edited, assigned a number, and made available to all interested parties. RFCs go through maturity levels and are categorized according to their requirement level.

Maturity Levels

An RFC, during its lifetime, falls into one of six maturity levels: proposed standard, draft standard, Internet standard, historic, experimental, and informational.

Proposed Standard

A proposed standard is a specification that is stable, well understood, and of sufficient interest to the Internet community. At this level, the specification is usually tested and implemented by several different groups.

Draft Standard

A proposed standard is elevated to draft standard status after at least two successful independent and interoperable implementations. Barring difficulties, a draft standard, with modifications if specific problems are encountered, normally becomes an Internet standard.

Internet Standard

A draft standard reaches Internet standard status after demonstrations of successful implementation

Historic

The historic RFCs are significant from a historical perspective. They either have been superseded by later specifications or have never passed the necessary maturity levels to become an Internet standard.

Experimental

An RFC classified as experimental describes work related to an experimental situation that does not affect the operation of the Internet. Such an RFC should not be implemented in any functional Internet service.

Informational

An RFC classified as informational contains general, historical, or tutorial information related to the Internet. It is usually written by someone in a non-Internet organization, such as a vendor.

Requirement Levels

RFCs are classified into five requirement levels: required, recommended, elective, limited use, and not recommended.

Required

An RFC is labeled required if it must be implemented by all Internets systems to achieve minimum conformance. For example, IF and ICMP are required protocols.

Recommended

An RFC labeled recommended is not required for minimum conformance; it is recommended because of its usefulness. For example, FTP and TELNET are recommended protocols.

Elective

An RFC labeled elective is not required and not recommended. However, a system can use it for its own benefit.

Limited Use

An RFC labeled limited use should be used only in limited situations. Most of the experimental RFCs fall under this category.

Not Recommended

An RFC labeled not recommended is inappropriate for general use. Normally a historic (deprecated) RFC may fall under this category.

1.7 INTERNET ADMINISTRATION

The Internet, with its roots primarily in the research domain, has evolved and gained a broader user base with significant commercial activity. Various groups that coordinate Internet issues have guided this growth and development. Appendix G gives the addresses, e-mail addresses, and telephone numbers for some of these groups. Shows the general organization of Internet administration. E-mail addresses and telephone numbers for some of these groups.

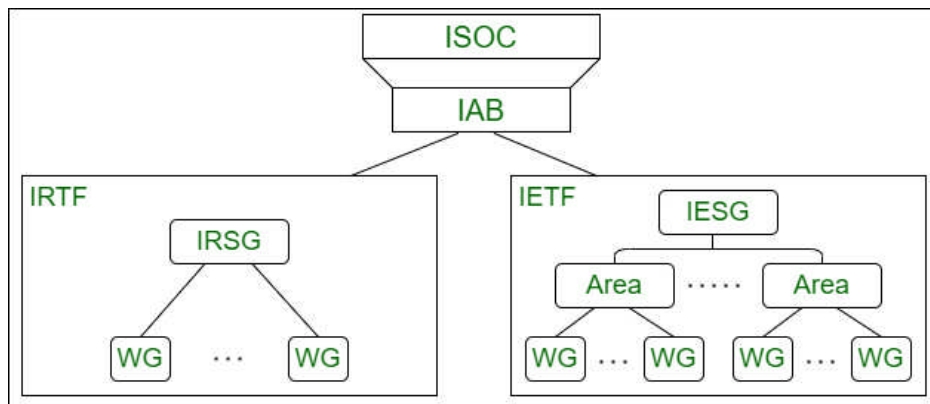


Figure the general organization of Internet administration.

The Internet Society (ISOC) is an international, nonprofit organization formed in 1992 to provide support for the Internet standards process. ISOC accomplishes this through maintaining and supporting other Internet administrative bodies such as IAB, IETF, IRTF, and IANA (see the following sections). ISOC also promotes research and other scholarly activities relating to the Internet.

IAB

The Internet Architecture Board (IAB) is the technical advisor to the ISOC. The main purposes of the IAB are to oversee the continuing development of the TCP/IP Protocol Suite and to serve in a technical advisory capacity to research members of the Internet community. IAB accomplishes this through its two primary components, the Internet Engineering Task Force (IETF) and the Internet Research Task Force (IRTF). Another responsibility of the IAB is the editorial management of the RFCs, described earlier. IAB is also the external liaison between the Internet and other standards organizations and forums.

IETF

The Internet Engineering Task Force (IETF) is a forum of working groups managed by the Internet Engineering Steering Group (IESG). IETF is responsible for identifying operational problems and proposing solutions to these problems. IETF also develops and reviews specifications intended as Internet standards. The working groups are collected into areas, and each area concentrates on a specific topic. Currently nine areas have been defined. The areas include applications, protocols, routing, network management next generation (IPng), and security.

IRTF

The Internet Research Task Force (IRTF) is a forum of working groups managed by the Internet Research Steering Group (IRSG). IRTF focuses on long-term research topics related to Internet protocols, applications, architecture, and technology.

1.8 REVIEW QUESTIONS

1. Explain the concept of Computer network.
2. How data communication is done? Explain in brief.
3. What Computer Network? Explain types of network.
4. What is Internet? Explain brief history of Internet.
5. Why do we require Internet standards? What are they?

1.9 SUMMARY

- Data communications are the transfer of data from one device to another via some form of transmission medium.

- A data communications system must transmit data to the correct destination in an accurate and timely manner.
- The five components that make up a data communications system are the message, sender, receiver, medium, and protocol.
- Text, numbers, images, audio, and video are different forms of information.
- A network can be categorized as a local area network or a wide area network.
- A LAN is a data communication system within a building, plant, or campus, or between nearby buildings.
- A WAN is a data communication system spanning states, countries, or the whole world.
- An internet is a network of networks.
- The Internet is a collection of many separate networks.
- There are local, regional, national, and international Internet service providers.
- A protocol is a set of rules that govern data communication; the key elements of a protocol are syntax, semantics, and timing.
- Standards are necessary to ensure that products from different manufacturers can work together as expected.

1.10 REFERENCES

1. Data Communication & Networking – Behrouz Forouzan
2. TCP/IP Protocol Suite– Behrouz Forouzan
3. Computer Networks–Andrew Tanenbaum



NETWORK MODELS

Unit Structure

- 2.0 Objectives
- 2.1 Protocol layering
- 2.2 TCP/IP protocol suite
- 2.3 The OSI model
- 2.4 Review questions
- 2.5 Summary
- 2.6 References

2.0 OBJECTIVES:

This chapter would make you understand the following concepts

- What is Protocol Layering?
- Principles of Protocol Layering
- What is TCP/IP protocol in brief
- Layers in the TCP/IP Protocol Suite
- The OSI model.
- Comparison of the OSI and TCP/IP Reference Models
- Problems of the TCP/IP Reference Model
- Problems of the OSI Model and Protocols

2.1 PROTOCOL LAYERING

We have discussed the term protocol in the previous chapter. In data communication and networking, a protocol defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively. When communication is simple, we may need only one simple protocol; when the communication is complex, we may need to divide the task between different layers, in which case we need a protocol at each layer, or protocol layering.

To understand the protocol layering let us develop two simple scenarios.

In the first scenario, communication is so simple that it can occur in only one layer. Assume Seeta and Kaveri are neighbors with a lot of common ideas. Communication between Seeta and Kaveri takes place in one layer, face to face, in the same language, as shown in



Figure: A single-layer protocol

Even in this simple scenario, we can see that a set of rules needs to be followed. First, Seeta and Kaveri know that they should greet each other when they meet. Second, they know that they should confine their vocabulary to the level of their friendship. Third, each party knows that she should refrain from speaking when the other party is speaking. Fourth, each party knows that the conversation should be a dialog, not a monolog; both should have the opportunity to talk about the issue. Fifth, they should exchange some nice words when they leave. We can see that the protocol used by Seeta and Kaveri is different from the communication between a professor and the students in a lecture hall. The communication in the second case is mostly monolog; the professor talks most of the time unless a student has a question, a situation in which the protocol dictates that she should raise her hand and wait for permission to speak. In this case, the communication is normally very formal and limited to the subject being taught.

Second Scenario

In the second scenario, we assume that Kaveri is offered a higher-level position in her company, but needs to move to another branch located in a city very far from Seeta. The two friends still want to continue their communication and exchange ideas because they have come up with an innovative project to start a new business when they both retire. They decide to continue their conversation using regular mail through the post office. However, they do not want their ideas to be revealed by other people if the letters are intercepted. They agree on an encryption/decryption technique. The sender of the letter encrypts it to make it unreadable by an intruder; the receiver of the letter decrypts it to get the original letter. We discuss the encryption/decryption methods in but for the moment we assume that Seeta and Kaveri use one technique that makes it hard to decrypt the letter if one does not have the key for doing so. Now we can say that the communication between Seeta and Kaveri takes place in three layers, as shown in Figure. We assume that Seeta and Kaveri each have three machines (or robots) that can perform the task at each layer.

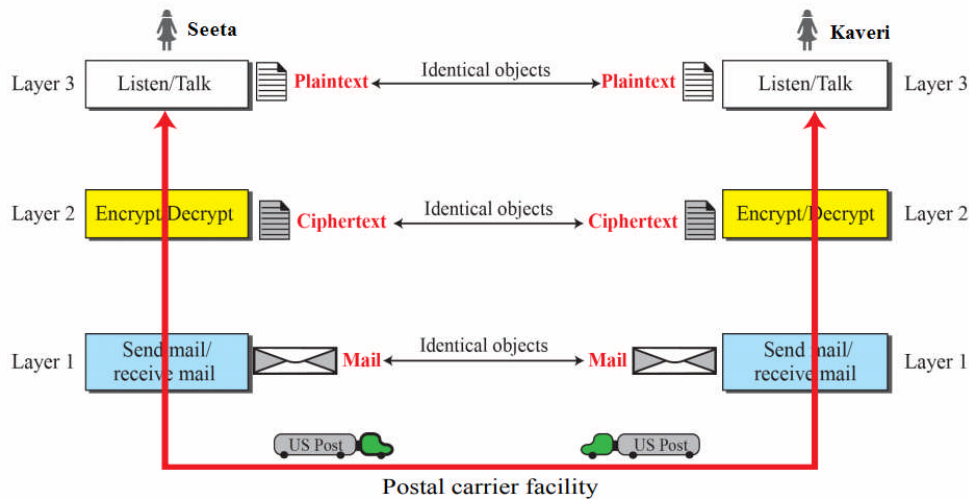


Figure: A three-layer protocol

Principles of Protocol Layering

Let us discuss two principles of protocol layering.

First Principle

The first principle dictates that if we want bidirectional communication, we need to make each layer so that it is able to perform two opposite tasks, one in each direction. For example, the third layer task is to listen (in one direction) and talk (in the other direction). The second layer needs to be able to encrypt and decrypt. The first layer needs to send and receive mail.

Second Principle

The second principle that we need to follow in protocol layering is that the two objects under each layer at both sites should be identical. For example, the object under layer 3 at both sites should be a plaintext letter. Both sites should be a cipher text letter. The object under layer 1 at both sites should be a piece of mail.

Logical Connections

After following the above two principles, we can think about logical connection between each layer as shown in below figure. This means that we have layer-to-layer communication. Seeta and Kaveri can think that there is a logical (imaginary) connection at each layer through which they can send the object created from that layer. We will see that the concept of logical connection will help us better understand the task of layering. We encounter in data communication and networking.

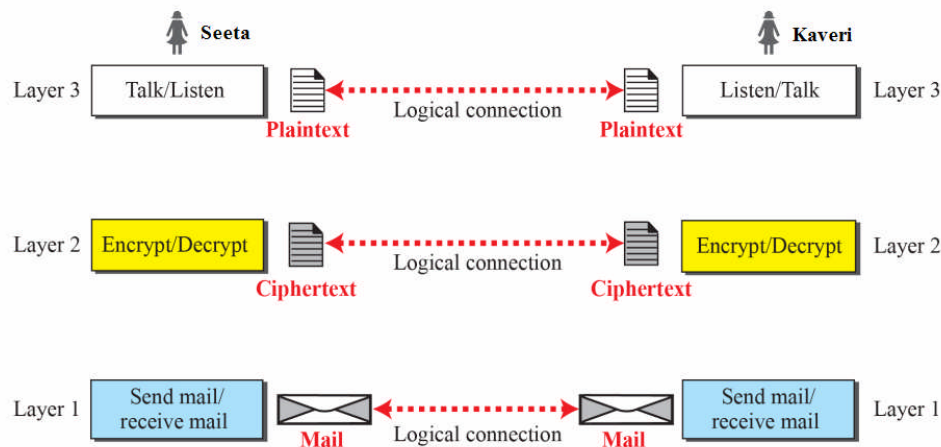


Figure: Logical connection between peer layers

1.2 TCP/IP PROTOCOL SUITE

Now that we know about the concept of protocol layering and the logical communication between layers in our second scenario, we can introduce the TCP/IP (Transmission Control Protocol/Internet Protocol). TCP/IP is a protocol suite (a set of protocols organized in different layers) used in the Internet today. It is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality. The term hierarchical means that each upper level protocol is supported by the services provided by one or more lower level protocols. The original TCP/IP protocol suite was defined as four software layers built upon the hardware. Today, however, TCP/IP is thought of as a five-layer model. Following figure shows both configurations.

Layered Architecture

To show how the layers in the TCP/IP protocol suite are involved in communication between two hosts, we assume that we want to use the suite in a small internet made up of three LANs (links), each with a link-layer switch. We also assume that the links are connected by one router, as shown in below Figure.

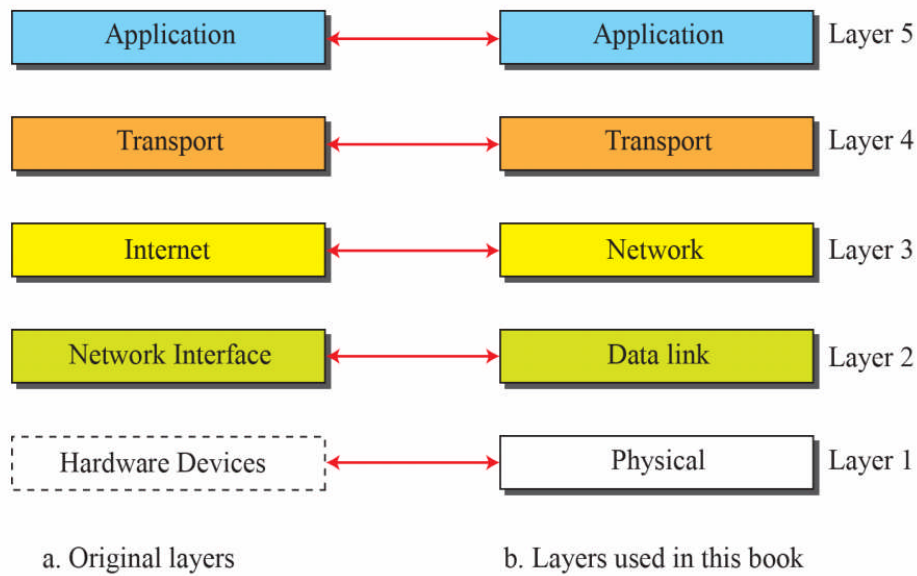


Figure: Layers in the TCP/IP protocol suite

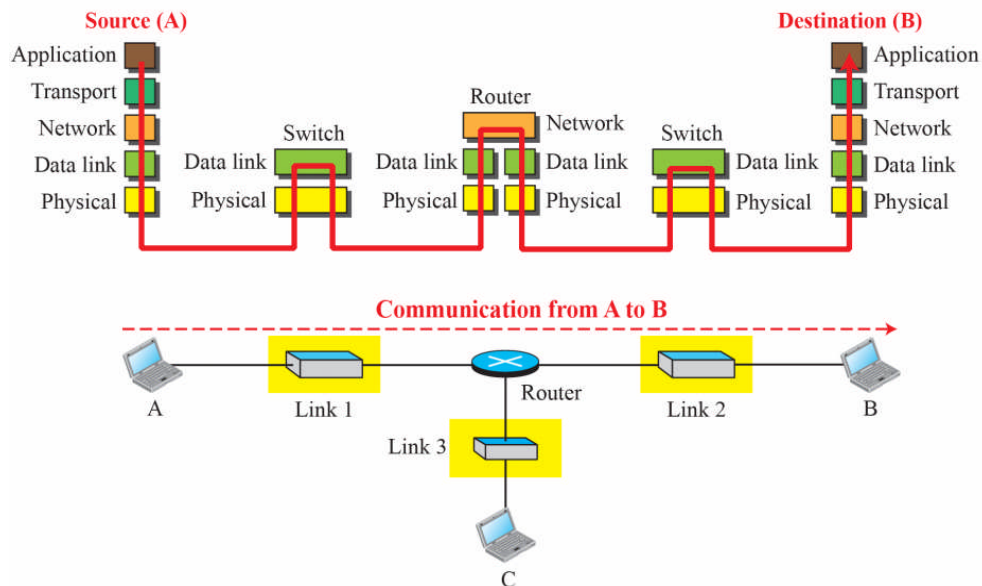


Figure: Communication through an internet

Layers in the TCP/IP Protocol Suite

After the above introduction, we briefly discuss the functions and duties of layers in the TCP/IP protocol suite. Each layer is discussed in detail in the next five parts of the book. To better understand the duties of each layer, we need to think about the logical connections between layers. Below figure shows logical connections in our simple internet.

Using logical connections makes it easier for us to think about the duty of each layer. As the figure shows, the duty of the application, transport, and network layers is end-to-end. However, the duty of the data-link and physical layers is hop-to-hop, in which a hop is a host or router.

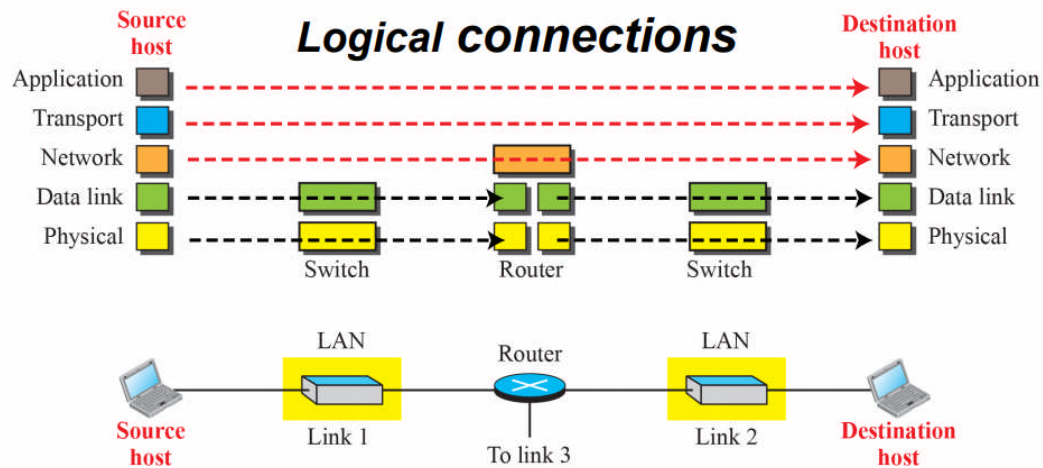


Figure: Logical connections between layers in TCP/IP

In other words, the domain of duty of the top three layers is the internet, and the domain of duty of the two lower layers is the link. Another way of thinking of the logical connections is to think about the data unit created from each layer. In the top three layers, the data unit (packets) should not be changed by any router or link-layer switch. In the bottom two layers, the packet created by the host is changed only by the routers, not by the link-layer switches. Below figure shows the second principle discussed previously for protocol layering. We show the identical objects below each layer related to each device.

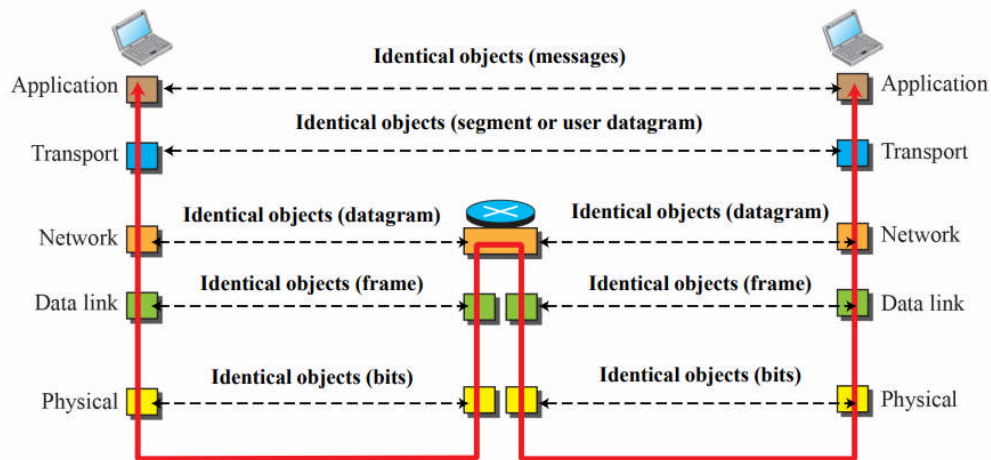


Figure: Identical objects in the TCP/IP protocol suite

Note that, although the logical connection at the network layer is between the two hosts, we can only say that identical objects exist between two hops in this case because a router may fragment the packet at the network layer and send more packets than received. Note that the link between two hops does not change the object.

Description of Each Layer

After understanding the concept of logical communication, we are ready to briefly discuss the duty of each layer

Physical Layer

We can say that the physical layer is responsible for carrying individual bits in a frame across the link. Although the physical layer is the lowest level in the TCP/IP protocol suite, the communication between two devices at the physical layer is still a logical communication because there is another, hidden layer, the transmission media, under the physical layer. Two devices are connected by a transmission medium (cable or air). We need to know that the transmission medium does not carry bits; it carries electrical or optical signals. So the bits received in a frame from the data-link layer are transformed and sent through the transmission media, but we can think that the logical unit between two physical layers in two devices is a bit. There are several protocols that transform a bit to a signal.

Data-link Layer

We have seen that an internet is made up of several links (LANs and WANs) connected by routers. There may be several overlapping sets of links that a datagram can travel from the host to the destination. The routers are responsible for choosing the best links. However, when the next link to travel is determined by the router, the data-link layer is responsible for taking the datagram and moving it across the link. The link can be a wired LAN with a link-layer switch, a wireless LAN, a wired WAN, or a wireless WAN. We can also have different protocols used with any link type. In each case, the data-link layer is responsible for moving the packet through the link. TCP/IP does not define any specific protocol for the data-link layer. It supports all the standard and proprietary protocols. Any protocol that can take the datagram and carry it through the link suffices for the network layer. The data-link layer takes a datagram and encapsulates it in a packet called «frame. Each link-layer protocol may provide a different service. Some link-layer protocols provide complete error detection and correction, some provide only error correction.

Network Layer

The network layer is responsible for creating a connection between the source computer and the destination computer. The communication at the network layer is host-to-host. However, since there can be several routers from the source to the destination, the routers in the path are responsible for choosing the best route for each packet. We can say that the network layer is responsible for host-to-host communication and routing the packet through possible routes. Again, we may ask ourselves why we need the network layer. We could have added the routing duty to the transport layer and dropped this layer. One reason, as we said before, is the separation of different tasks between different layers. The second reason is that the routers do not need the application and transport layers.

Transport Layer

The logical connection at the transport layer is also end-to-end. The transport layer at the source host gets the message from the application layer, encapsulates it in a transport layer packet (called a segment or a user datagram in different protocols) and sends it, through the logical (imaginary) connection, to the transport layer at the destination host. In other words, the transport layer is responsible for giving services to the application layer: to get a message from an application program running on the source host and deliver it to the corresponding application program on the destination host. We may ask why we need an end-to-end transport layer when we already have an end-to-end application layer. The reason is the separation of tasks and duties, which we discussed earlier. The transport layer should be independent of the application layer. In addition, we will see that we have more than one protocol in the transport layer, which means that each application program can use the protocol that best matches its requirement.

Application Layer

The logical connection between the two application layers is end-to-end. The two application layers exchange messages between each other as though there were a bridge between the two layers. However, we should know that the communication is done through all the layers. Communication at the application layer is between two processes (two programs running at this layer). To communicate, a process sends a request to the other process and receives a response. Process-to-process communication is the duty of the application layer. The application layer in the Internet includes many predefined protocols, but a user can also create a pair of processes to be run at the two hosts.

2.3 THE OSI MODEL

Although, when speaking of the Internet, everyone talks about the *TCP/IP* protocol suite, this suite is not the only suite of protocols defined. Established in 1947, the International Organization for Standardization (ISO) is a multinational body dedicated to worldwide agreement on international standards. Almost three-fourths of the countries in the world are represented in the ISO. An ISO standard that covers all aspects of network communications is the Open Systems Interconnection (OSI) model. It was first introduced in the late 1970s.

ISO is the organization; OSI is the model

The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network.

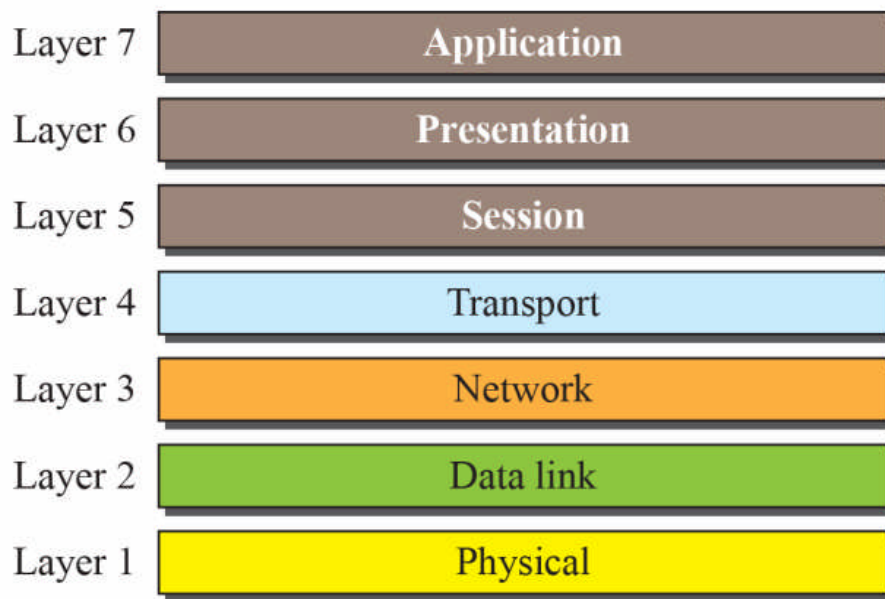


Figure: The OSI model

The TCP/IP Reference Model:

The TCP/IP reference model was developed prior to OSI model. The major design goals of this model were,

1. To connect multiple networks together so that they appear as a single network.
2. To survive after partial subnet hardware failures.
3. To provide a flexible architecture.

Unlike OSI reference model, TCP/IP reference model has only 4 layers.

They are,

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

Application Layer
Transport Layer
Internet Layer
Host-to-Network Layer

Figure: TCP/IP Reference model

Host-to-Network Layer

The TCP/IP reference model does not really say much about what happens here, except to point out that the host has to connect to the

network using some protocol so it can send IP packets to it. This protocol is not defined and varies from host to host and network to network.

Internet Layer

This layer, called the internet layer, is the linchpin that holds the whole architecture together. Its job is to permit hosts to inject packets into any network and have them travel independently to the destination (potentially on a different network). They may even 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. Note that "internet" is used here in a generic sense, even though this layer is present in the Internet.

The internet layer defines an official packet format and protocol called IP (Internet Protocol). The job of the internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly the major issue here, as is avoiding congestion. For these reasons, it is reasonable to say that the TCP/IP internet layer is similar in functionality to the OSI network layer.

The Transport Layer

The layer above the internet layer in the TCP/IP model is now usually called the transport layer. It is designed to allow peer entities on the source and destination hosts to carry on a conversation, just as in the OSI transport layer. Two end-to-end transport protocols have been defined here. The first one, TCP (Transmission Control Protocol), 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. It fragments the incoming byte stream into discrete messages and passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.

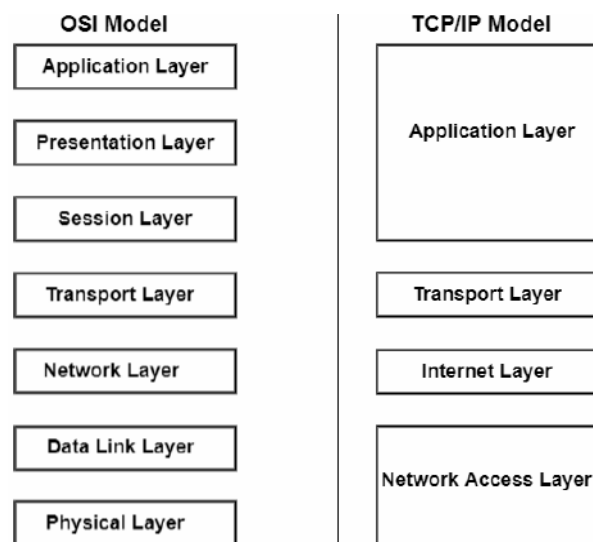


Figure: OSI and TCP/IP Reference Model

The second protocol in this layer, UDP (User Datagram Protocol), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server-type request reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP, and UDP is shown

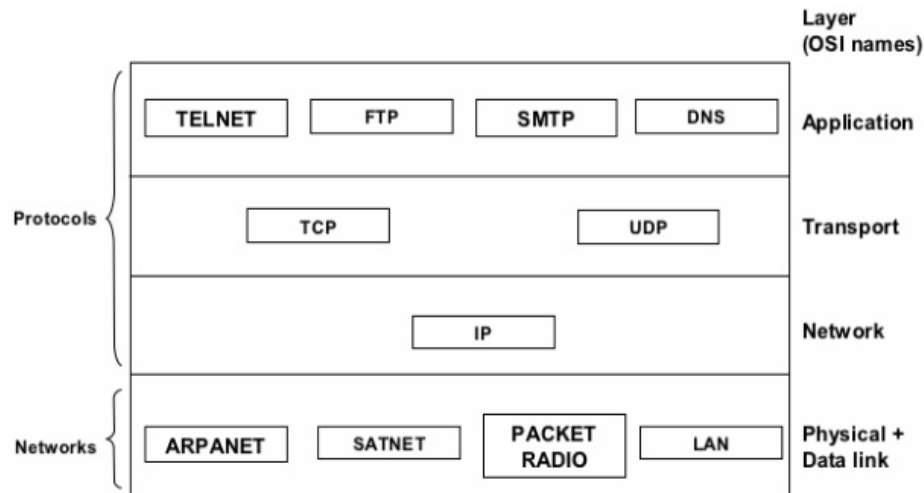


Figure: Protocol and networks in the TCP/IP model initially

Application layer:

The TCP/IP model does not have session or presentation layers. On top of the transport layer is the application layer. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP), as shown in the above figure. The virtual terminal protocol allows a user on one machine to log onto a distant machine and work there. The file transfer protocol provides a way to move data efficiently from one machine to another. Electronic mail was originally just a kind of file transfer, but later a specialized protocol (SMTP) was developed for it.

Many other protocols have been added to these over the years: the Domain Name System (DNS) for mapping host names onto their network addresses, NNTP, the protocol for moving USENET news articles around, and HTTP, the protocol for fetching pages on the World Wide Web, and many others.

Comparison of the OSI and TCP/IP Reference Models

The OSI and TCP/IP reference models have much in common. Both are based on the concept of a stack of independent protocols. Also, the functionality of the layers is roughly similar. For example, in both models the layers up through and including the transport layer are there to provide an end-to-end, network-independent transport service to processes wishing to communicate. These layers form the transport provider. Again in both models, the layers above transport are application-oriented users of the transport service.

Despite these fundamental similarities, the two models also have many differences

Three concepts are central to the OSI model

1. Services.
2. Interfaces.
3. Protocols.

Probably the biggest contribution of the OSI model is to make the distinction between these three concepts explicit. Each layer performs some services for the layer above it. The service definition tells what the layer does, not how entities above it access it or how the layer works. It defines the layer's semantics.

A layer's interface tells the processes above it how to access it. It specifies what the parameters are and what results to expect. It, too, says nothing about how the layer works inside. Finally, the peer protocols used in a layer are the layer's own business. It can use any protocols it wants to, as long as it gets the job done (i.e., provides the offered services). It can also change them at will without affecting software in higher layers.

The TCP/IP model did not originally clearly distinguish between service, interface, and protocol, although people have tried to retrofit it after the fact to make it more OSI-like. For example, the only real services offered by the internet layer are SEND IP PACKET and RECEIVE IP PACKET.

As a consequence, the protocols in the OSI model are better hidden than in the TCP/IP model and can be replaced relatively easily as the technology changes. Being able to make such changes is one of the main purposes of having layered protocols in the first place.

The OSI reference model was devised before the corresponding protocols were invented. This ordering means that the model was not biased toward one particular set of protocols, a fact that made it quite general. The downside of this ordering is that the designers did not have much experience with the subject and did not have a good idea of which functionality to put in which layer.

Another difference is in the area of connectionless versus connection-oriented communication. The OSI model supports both connectionless and connection-oriented communication in the network layer, but only connection-oriented communication in the transport layer, where it counts (because the transport service is visible to the users). The TCP/IP model has only one mode in the network layer (connectionless) but supports both modes in the transport layer, giving the users a choice. This choice is especially important for simple request-response protocols.

Problems of the TCP/IP Reference Model

First, the model does not clearly distinguish the concepts of service, interface, and protocol. Good software engineering practice requires differentiating between the specification and the implementation, something that OSI does very carefully, and TCP/IP does not. Consequently, the TCP/IP model is not much of a guide for designing new networks using new technologies.

Second, the TCP/IP model is not at all general and is poorly suited to describing any protocol stack other than TCP/IP. Trying to use the TCP/IP model to describe Bluetooth, for example, is completely impossible.

Third, the host-to-network layer is not really a layer at all in the normal sense of the term as used in the context of layered protocols. It is an interface (between the network and data link layers). The distinction between an interface and a layer is crucial, and one should not be sloppy about it.

Fourth, the TCP/IP model does not distinguish (or even mention) the physical and data link layers. These are completely different. The physical layer has to do with the transmission characteristics of copper wire, fiber optics, and wireless communication. The data link layer's job is to delimit the start and end of frames and get them from one side to the other with the desired degree of reliability. A proper model should include both as separate layers. The TCP/IP model does not do this. Finally, although the IP and TCP protocols were carefully thought out and well implemented, many of the other protocols were ad hoc, generally produced by a couple of graduate students hacking away until they got tired. The protocol implementations were then distributed free, which resulted in their becoming widely used, deeply entrenched, and thus hard to replace. Some of them are a bit of an embarrassment now. The virtual terminal protocol, TELNET, for example, was designed for a ten-character per second mechanical Teletype terminal. It knows nothing of graphical user interfaces and mice. Nevertheless, 25 years later, it is still in widespread use.

Problems of the OSI Model and Protocols:

- Bad timing
- Bad technology
- Bad implementations
- Bad politics

Bad Timing:

The time at which a standard is established is absolutely critical to its success. David Clark of M.I.T. has a theory of standards that he calls the apocalypse of the two elephants, which is illustrated in Fig. This figure shows the amount of activity surrounding a new subject. When the subject is first discovered, there is a burst of research activity in the form of

discussions, papers, and meetings. After a while this activity subsides, corporations discover the subject, and the billion-dollar wave of investment hits. It is essential that the standards be written in the trough in between the two "elephants." If the standards are written too early, before the research is finished, the subject may still be poorly understood; the result is bad standards. If they are written too late, so many companies may have already made major investments in different ways of doing things that the standards are effectively ignored. If the interval between the two elephants is very short (because everyone is in a hurry to get started), the people developing the standards may get crushed.

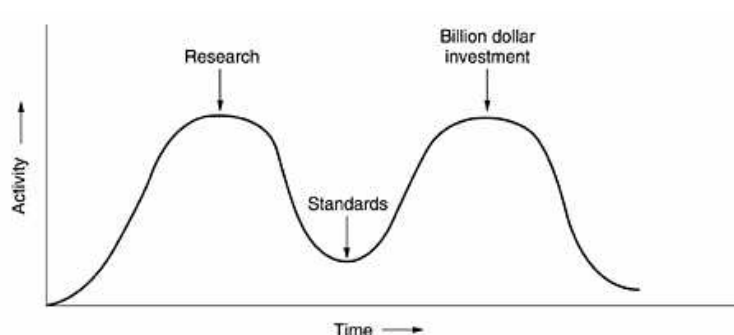


Figure: The apocalypse of the two elephants

Bad Implementations:

Given the enormous complexity of the model and the protocols, it will come as no surprise that the initial implementations were huge, unwieldy, and slow. Everyone who tried them got burned. It did not take long for people to associate "OSI" with "poor quality." Although the products improved in the course of time, the image stuck.

Bad Politics:

On account of the initial implementation, many people, especially in academia, thought of TCP/IP as part of UNIX, and UNIX in the 1980s in academia was not unlike parenthood (then incorrectly called motherhood) and apple pie. OSI, on the other hand, was widely thought to be the creature of the European telecommunication ministries, the European Community, and later the U.S. Government. This belief was only partly true, but the very idea of a bunch of government bureaucrats trying to shove a technically inferior standard down the throats of the poor researchers and programmers down in the trenches actually developing computer networks did not help much. Some people viewed this development in the same light as IBM announcing in the 1960s that PL/I was the language of the future, or DoD correcting this later by announcing that it was actually Ada.

Bad Technology:

The second reason that OSI never caught on is that both the model and the protocols are flawed. The choice of seven layers was more political than technical, and two of the layers (session and presentation)

are nearly empty, whereas two other ones (data link and network) are overfull.

The OSI model, along with the associated service definitions and protocols, is extraordinarily complex. When piled up, the printed standards occupy a significant fraction of a meter of paper. They are also difficult to implement and inefficient in operation. In addition to being incomprehensible, another problem with OSI is that some functions, such as addressing, flow control, and error control, reappear again and again in each layer.

2.4 REVIEW QUESTIONS

1. What is prototype layering?
2. What are the principles of protocol layering?
3. Explain the TCP/IP Protocol in brief.
4. Explain the OSI Model in brief.
5. Differentiate between the OSI and TCP/IP referential model

2.5 SUMMARY

- The International Standards Organization created a model called the Open Systems Interconnection, which allows diverse systems to communicate.
- The seven-layer OSI model provides guidelines for the development of universally compatible networking protocols.
- The physical, data link, and network layers are the network support layers.
- The session, presentation, and application layers are the user support layers.
- The transport layer links the network support layers and the user support layers.
- The physical layer coordinates the functions required to transmit a bit stream over a physical medium.
- The data link layer is responsible for delivering data units from one station to the next without errors.
- The network layer is responsible for the source-to-destination delivery of a packet across multiple network links.
- The transport layer is responsible for the process-to-process delivery of the entire message.
- The session layer establishes, maintains, and synchronizes the interactions between communicating devices.

- The presentation layer ensures interoperability between communicating devices through transformation of data into a mutually agreed upon format.
- The application layer enables the users to access the network.
- TCP/IP is a five-layer hierarchical protocol suite developed before the OSI model.
- The TCP/IP application layer is equivalent to the combined session, presentation, and application layers of the OSI model.
- Four levels of addresses are used in an internet following the TCP/IP protocols: physical(link) addresses, logical (IP) addresses, port addresses, and specific addresses.
- The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN.
- The IP address uniquely defines a host on the Internet.
- The port address identifies a process on a host.
- A specific address is a user-friendly address.

2.6 REFERENCES

1. Data Communication & Networking – Behrouz Forouzan
2. TCP/IP Protocol Suite– Behrouz Forouzan
3. Computer Networks–Andrew Tanenbaum¹



INTRODUCTION TO PHYSICAL LAYER

Unit Structure

- 3.0 Objectives
- 3.1 Data and signals
- 3.2 Periodic analog signals
- 3.3 Digital Signal
- 3.4 Transmission Impairment
- 3.5 Data rate limits
- 3.6 Performance
- 3.7 Review questions
- 3.8 Summary
- 3.9 References

3.0 OBJECTIVES:

This chapter would make you understand the following concepts

- What is data and single?
- What is analog and digital data?
- Concept of periodic and non-periodic single
- Concept of Sine wave, Wavelength, Bandwidth
- Digital Signals, bit rates, bit length
- Concept of transmission impairment
- Different causes of impairment
- Data rate limits
- Measuring the performance, Throughput, Latency and Queuing Time

3.1 DATA AND SIGNALS

One of the major functions of the physical layer is to move data in the form of electromagnetic signals across a transmission medium. Whether you are collecting numerical statistics from another computer, sending animated pictures from a design workstation, or causing a bell to ring at a distant control center, you are working with the transmission of data across network connections.

Generally, the data usable to a person or application are not in a form that can be transmitted over a network. For example, a photograph must first be changed to a form that transmission media can accept. Transmission media work by conducting energy along a physical path.

Analog and Digital Data

Data can be analog or digital. The term analog data refers to information that is continuous; digital data refers to information that has discrete states. For example, an analog clock that has hour, minute, and second hands gives information in a continuous form; the movements of the hands are continuous. On the other hand, a digital clock that reports the hours and the minutes will change suddenly from 9:02 to 9:03.

Analog data, such as the sounds made by a human voice, take on continuous values. When someone speaks, an analog wave is created in the air. This can be captured by a microphone and converted to an analog signal or sampled and converted to a digital signal.

Digital data take on discrete values. For example, data are stored in computer memory in the form of 0s and 1s. They can be converted to a digital signal or modulated into an analog signal for transmission across a medium.

Analog and Digital Signals

Like the data they represent, signals can be either analog or digital. An analog signal has infinitely many levels of intensity over a period of time. As the wave moves from value A to value B, it passes through and includes an infinite number of values along its path. A digital signal, on the other hand, can have only a limited number of defined values. Although each value can be any number, it is often as simple as 1 and 0.

The simplest way to show signals is by plotting them on a pair of perpendicular axes. The vertical axis represents the value or strength of a signal. The horizontal axis represents time. Figure 3.1 illustrates an analog signal and a digital signal. The curve representing the analog signal passes through an infinite number of points. The vertical lines of the digital signal, however, demonstrate the sudden jump that the signal makes from value to value.

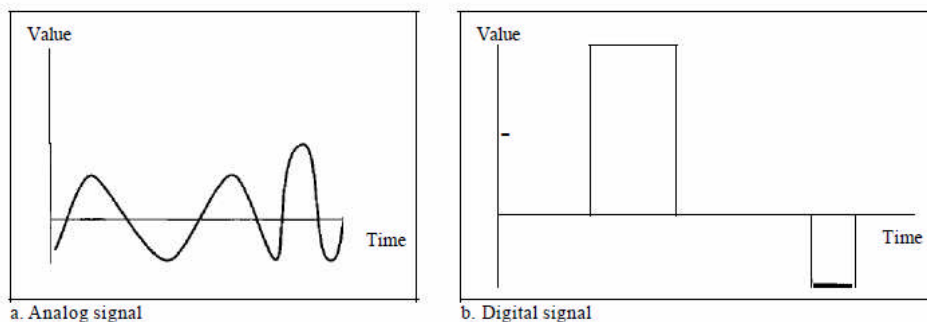


Figure: Comparison of analog and digital signals

Periodic and Non-periodic Signals

Both analog and digital signals can take one of two forms: periodic or non-periodic (sometimes referred to as aperiodic, because the prefix a in

Greek means "non"). A periodic signal completes a pattern within a measurable time frame, called a period, and repeats that pattern over subsequent identical periods. The completion of one full pattern is called a cycle. A non-periodic signal changes without exhibiting a pattern or cycle that repeats over time. Both analog and digital signals can be periodic or non-periodic. In data communications, we commonly use periodic analog signals and non-periodic digital signals

3.2 PERIODIC ANALOG SIGNALS

Periodic analog signals can be classified as simple or composite. A simple periodic analog signal, a sine wave, cannot be decomposed into simpler signals. A composite periodic analog signal is composed of multiple sine waves.

Sine Wave

The sine wave is the most fundamental form of a periodic analog signal. When we visualize it as a simple oscillating curve, its change over the course of a cycle is smooth and consistent, a continuous, rolling flow. Following Figure shows a sine wave. Each cycle consists of a single arc above the time axis followed by a single arc below it

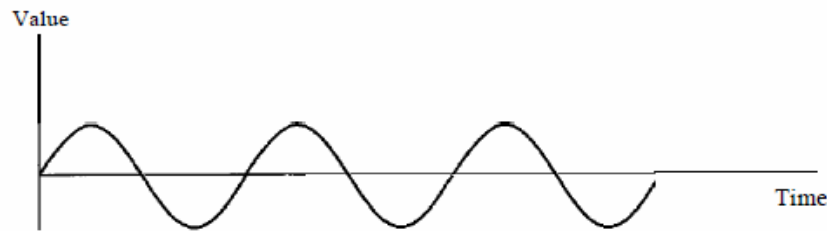


Figure: A sine wave

A sine wave can be represented by three parameters: the *peak amplitude*, the *frequency*, and the *phase*. These three parameters fully describe a sine wave.

Peak Amplitude

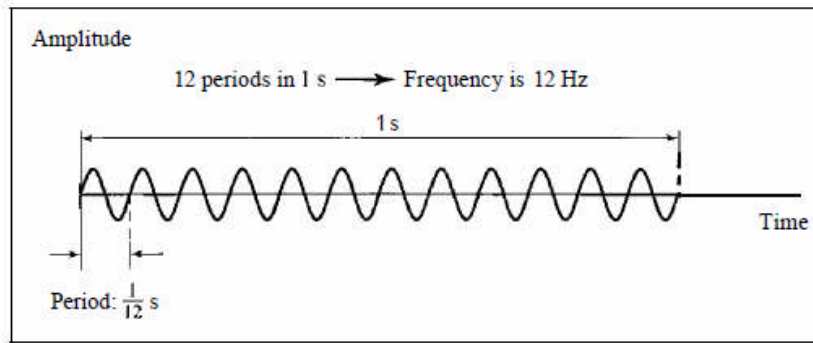
The peak amplitude of a signal is the absolute value of its highest intensity, proportional to the energy it carries. For electric signals, peak amplitude is normally measured in *volts*.

Period and Frequency

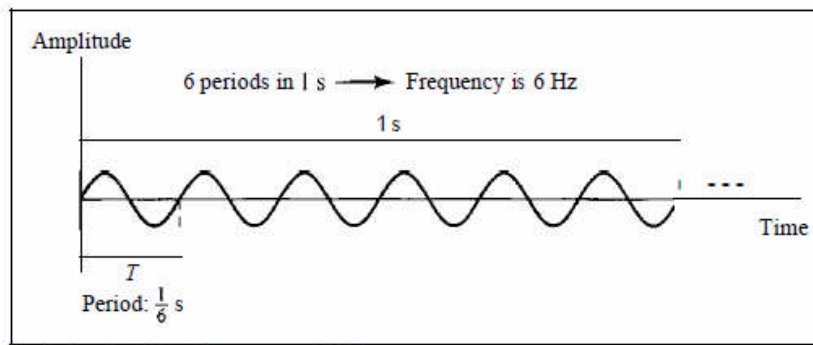
Period refers to the amount of time, in seconds, a signal needs to complete 1 cycle. Frequency refers to the number of periods in 1 s. Note that period and frequency are just one characteristic defined in two ways. Period is the inverse of frequency, and frequency is the inverse of period, as the following formulas show.

$$f = \frac{1}{T} \quad \text{and} \quad T = \frac{1}{f}$$

Frequency and period are the inverse of each other.



a. A signal with a frequency of 12 Hz



b. A signal with a frequency of 6 Hz

Figure: Two signals with the same amplitude and phase, but different frequencies

Period is formally expressed in seconds. Frequency is formally expressed in Hertz (Hz), which is cycle per second. Units of period and frequency are shown in Table

<i>Unit</i>	<i>Equivalent</i>	<i>Unit</i>	<i>Equivalent</i>
Seconds (s)	1 s	Hertz (Hz)	1 Hz
Milliseconds (ms)	10^{-3} s	Kilohertz (kHz)	10^3 Hz
Microseconds (μ s)	10^{-6} s	Megahertz (MHz)	10^6 Hz
Nanoseconds (ns)	10^{-9} s	Gigahertz (GHz)	10^9 Hz
Picoseconds (ps)	10^{-12} s	Terahertz (THz)	10^{12} Hz

Table: Units of period and frequency

Phase

The term phase describes the position of the waveform relative to time 0. If we think of the wave as something that can be shifted backward or forward along the time axis, phase describes the amount of that shift. It indicates the status of the first cycle.

Phase is measured in degrees or radians [360° is 2π rad; 1° is $2\pi/360$ rad, and 1 rad is $360/(2\pi)$]. A phase shift of 360° corresponds to a shift of a complete period; a phase shift of 180° corresponds to a shift of one-half of a period; and a phase shift of 90° corresponds to a shift of one-quarter of a period

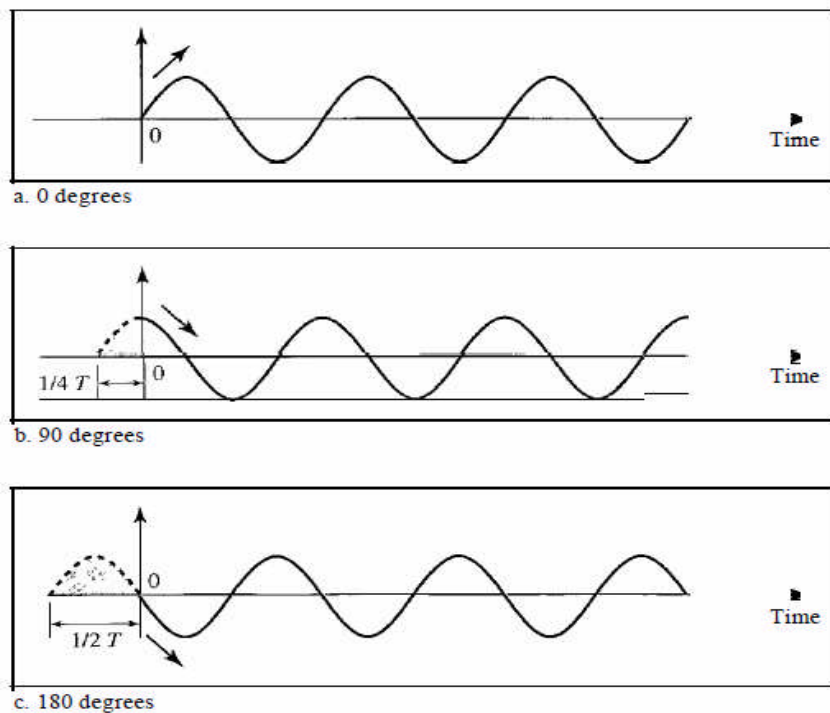


Figure: Three sine waves with the same amplitude and frequency, but different phases

By the looking at above Figure, we can say that

- A sine wave with a phase of 0° starts at time 0 with a zero amplitude. The amplitude is increasing.
- A sine wave with a phase of 90° starts at time 0 with a peak amplitude. The amplitude is decreasing.
- A sine wave with a phase of 180° starts at time 0 with a zero amplitude. The amplitude is decreasing

Another way to look at the phase is in terms of shift or offset. We can say that

- A sine wave with a phase of 0° is not shifted.
- A sine wave with a phase of 90° is shifted to the left by $1/4$ cycle. However, note that the signal does not really exist before time 0.
- A sine wave with a phase of 180° is shifted to the left by $1/2$ cycle. However, note that the signal does not really exist before time 0.

Wavelength

Wavelength is another characteristic of a signal traveling through a transmission medium. Wavelength binds the period or the frequency of a simple sine wave to the propagation speed of the medium.



Figure: Wavelength and period

While the frequency of a signal is independent of the medium, the wavelength depends on both the frequency and the medium. Wavelength is a property of any type of signal. In data communications, we often use wavelength to describe the transmission of light in an optical fiber. The wavelength is the distance a simple signal can travel in one period.

Wavelength can be calculated if one is given the propagation speed (the speed of light) and the period of the signal. However, since period and frequency are related to each other, if we represent wavelength by λ , propagation speed by λ (speed of light), and frequency by f , we get

$$\text{Wavelength} = \text{propagation speed} \times \text{period} = \frac{\text{propagation speed}}{\text{frequency}}$$

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. That speed is lower in air and even lower in cable. The wavelength is normally measured in micrometers (microns) instead of meters.

Time and Frequency Domains

A sine wave is comprehensively defined by its amplitude, frequency, and phase. We have been showing a sine wave by using what is called a time-domain plot. The time-domain plot shows changes in signal amplitude with respect to time (it is an amplitude-versus-time plot). Phase is not explicitly shown on a time-domain plot.

To show the relationship between amplitude and frequency, we can use what is called a frequency-domain plot. A frequency-domain plot is concerned with only the peak value and the frequency. Changes of amplitude during one period are not shown.

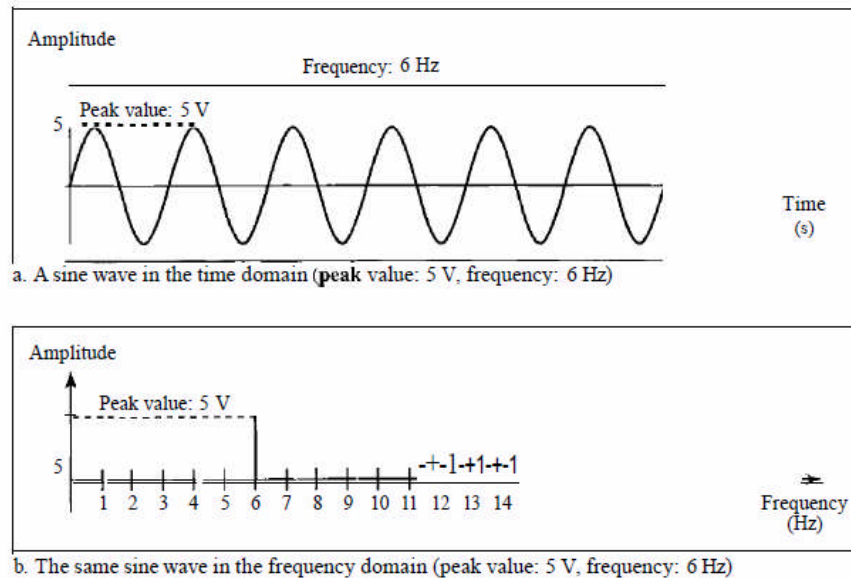


Figure: The time-domain and frequency-domain plots of a sine wave

It is obvious that the frequency domain is easy to plot and conveys the information that one can find in a time domain plot. The advantage of the frequency domain is that we can immediately see the values of the frequency and peak amplitude. A complete sine wave is represented by one spike. The position of the spike shows the frequency; its height shows the peak amplitude.

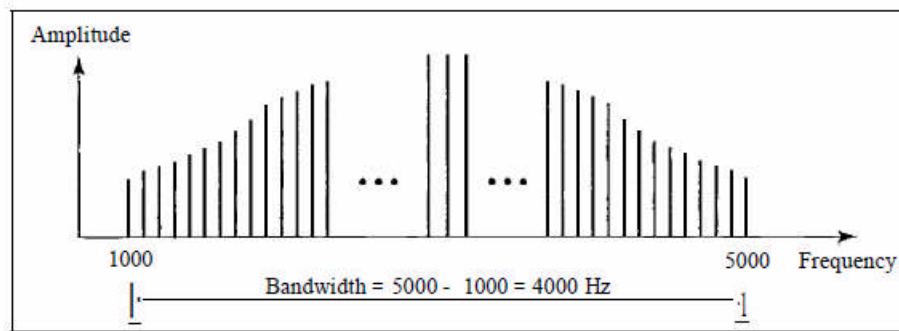
Composite Signals

So far, we have focused on simple sine waves. Simple sine waves have many applications in daily life. We can send a single sine wave to carry electric energy from one place to another. For example, the power company sends a single sine wave with a frequency of 60 Hz to distribute electric energy to houses and businesses. As another example, we can use a single sine wave to send an alarm to a security center when a burglar opens a door or window in the house. In the first case, the sine wave is carrying energy; in the second, the sine wave is a signal of danger.

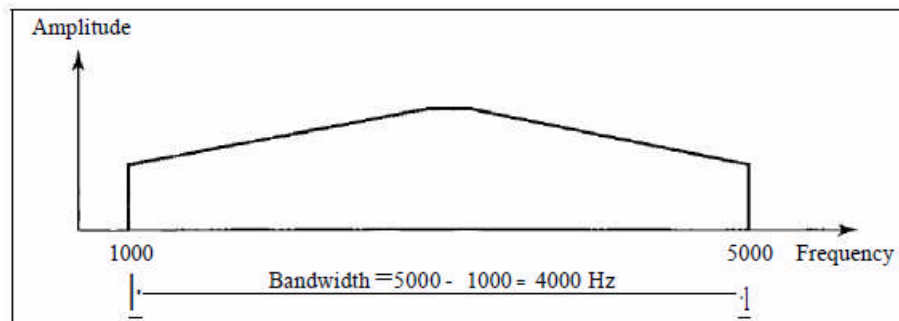
Bandwidth

The range of frequencies contained in a composite signal is its bandwidth. The bandwidth is normally a difference between two numbers. For example, if a composite signal contains frequencies between 1000 and 5000, its bandwidth is $5000 - 1000$, or 4000.

Following figure shows the concept of bandwidth. The figure depicts two composite signals, one periodic and the other non-periodic. The bandwidth of the periodic signal contains all integer frequencies between 1000 and 5000 (1000, 1001, 1002, ...). The bandwidth of the non-periodic signals has the same range, but the frequencies are continuous.



a. Bandwidth of a periodic signal

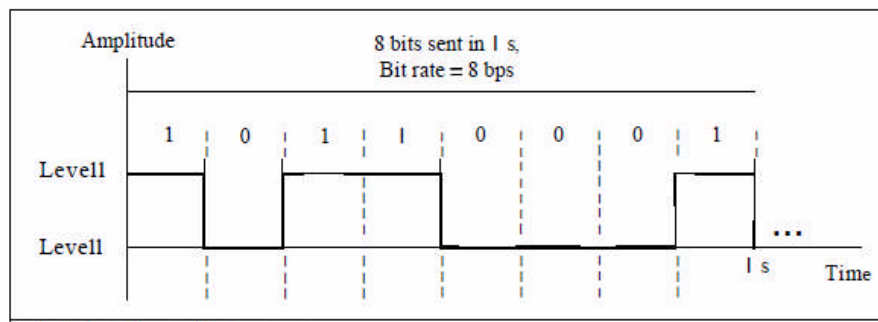


b. Bandwidth of a nonperiodic signal

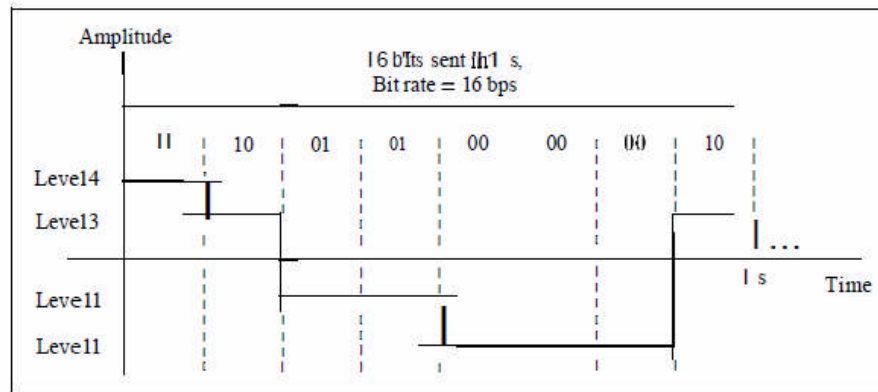
Figure: The bandwidth of periodic and non-periodic composite signals

3.3 DIGITAL SIGNALS

In addition to being represented by an analog signal, information can also be represented by a digital signal. For example, a 1 can be encoded as a positive voltage and a 0 as zero voltage. A digital signal can have more than two levels. In this case, we can send more than 1 bit for each level. Following figure shows two signals, one with two levels and the other with four.



a. A digital signal with two levels



b. A digital signal with four levels

Figure: Two digital signals: one with two signal levels and the other with four signal levels

We send 1 bit per level in part a of the figure and 2 bits per level in part b of the figure. In general, if a signal has L levels, each level needs $\log_2 L$ bits.

Bit Rate

Most digital signals are non-periodic, and thus period and frequency are not appropriate characteristics. Another *term-bit rate* (instead of *frequency*)-is used to describe digital signals. The bit rate is the number of bits sent in 1s, expressed in bits per second (bps). Above figure shows the bit rate for two signals.

Bit Length

We discussed the concept of the wavelength for an analog signal: the distance one cycle occupies on the transmission medium. We can define something similar for a digital signal: the bit length. The bit length is the distance one bit occupies on the transmission medium.

$$\text{Bit length} = \text{propagation speed} \times \text{bit duration}$$

Digital Signal as a Composite Analog Signal

Based on Fourier analysis, a digital signal is a composite analog signal. The bandwidth is infinite, as you may have guessed. We can intuitively come up with this concept when we consider a digital signal. A digital signal, in the time domain, comprises connected vertical and horizontal line segments. A vertical line in the time domain means a

frequency of infinity (sudden change in time); a horizontal line in the time domain means a frequency of zero (no change in time). Going from a frequency of zero to a frequency of infinity (and vice versa) implies all frequencies in between are part of the domain.

Fourier analysis can be used to decompose a digital signal. If the digital signal is periodic, which is rare in data communications, the decomposed signal has a frequency domain representation with an infinite bandwidth and discrete frequencies. If the digital signal is non-periodic, the decomposed signal still has an infinite bandwidth, but the frequencies are continuous. Following figure shows a periodic and a non-periodic digital signal and their bandwidths.

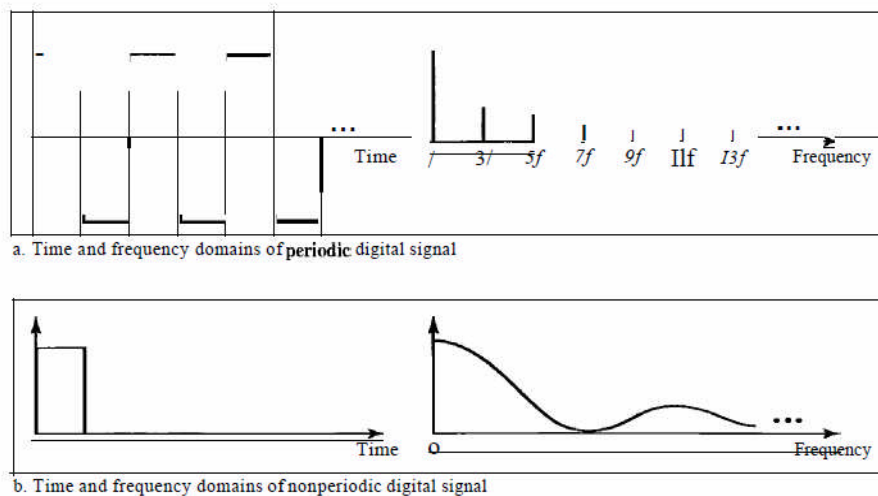


Figure: The time and frequency domains of periodic and non-periodic digital signals

Note that both bandwidths are infinite, but the periodic signal has discrete frequencies while the non-periodic signal has continuous frequencies.

3.4 TRANSMISSION IMPAIRMENT

Signals travel through transmission media, which are not perfect. The imperfection causes signal impairment. This means that the signal at the beginning of the medium is not the same as the signal at the end of the medium. What is sent is not what is received. Three causes of impairment are attenuation, distortion, and noise.

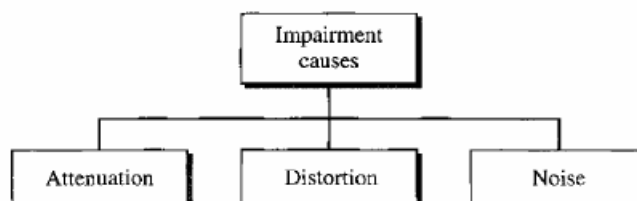


Figure: Causes of impairment

Attenuation

Attenuation means a loss of energy. When a signal, simple or composite, travels through a medium, it loses some of its energy in overcoming the resistance of the medium. That is why a wire carrying electric signals 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. Figure 3.26 shows the effect of attenuation and amplification.

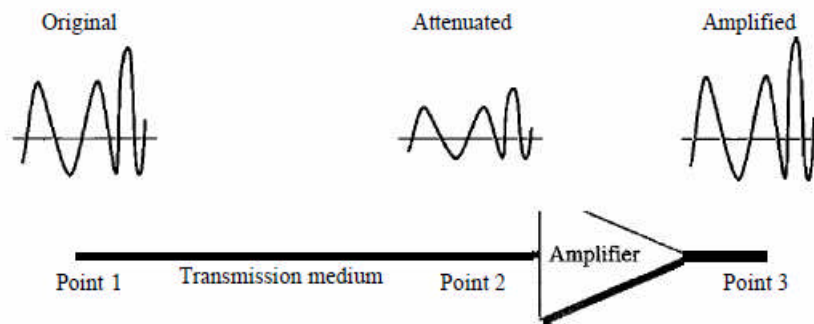


Figure: Attenuation

Decibel

To show that a signal has lost or gained strength, engineers use the unit of the decibel. The decibel (dB) measures the relative strengths of two signals or one signal at two different points. Note that the decibel is negative if a signal is attenuated and positive if a signal is amplified.

$$\text{dB} = 10 \log_{10} \frac{P_2}{P_1}$$

Variables P_1 and P_2 are the powers of a signal at points 1 and 2, respectively. Note that some engineering books define the decibel in terms of voltage instead of power. In this case, because power is proportional to the square of the voltage, the formula is $\text{dB} = 20 \log_{10} (V_2/V_1)$. In this text, we express dB in terms of power.

Distortion

Distortion means that the signal changes its form or shape. Distortion can occur in a composite signal made of different frequencies. Each signal component has its own propagation speed through a medium and, therefore, its own delay in arriving at the final destination. Differences in delay may create a difference in phase if the delay is not exactly the same as the period duration. In other words, signal components at the receiver have phases different from what they had at the sender. The shape of the composite signal is therefore not the same. Following figure shows the effect of distortion on a composite signal.

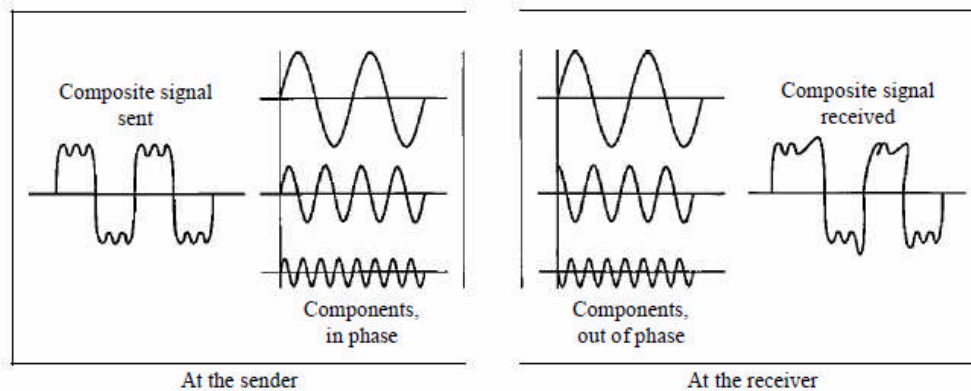


Figure: Distortion

Noise

Noise is another cause of impairment. Several types of noise, such as thermal noise, induced noise, crosstalk, and impulse noise, may corrupt the signal. Thermal noise is the random motion of electrons in a wire which creates an extra signal not originally sent by the transmitter. Induced noise comes from sources such as motors and appliances. These devices act as a sending antenna, and the transmission medium acts as the receiving antenna. Crosstalk is the effect of one wire on the other. One wire acts as a sending antenna and the other as the receiving antenna. Impulse noise is a spike (a signal with high energy in a very short time) that comes from power lines, lightning, and so on. Following figure shows the effect of noise on a signal.

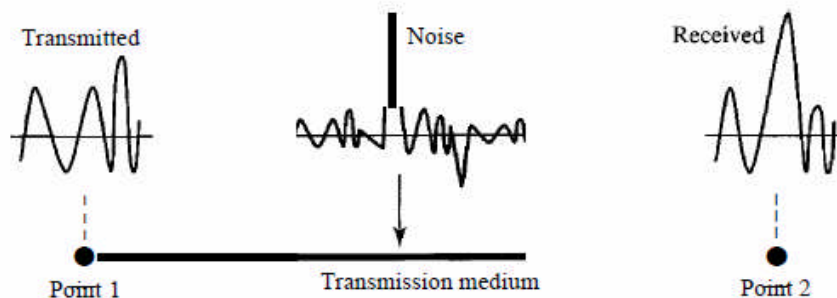


Figure: Noise

Signal-to-Noise Ratio (SNR)

As we will see later, to find the theoretical bit rate limit, we need to know the ratio of the signal power to the noise power. The signal-to-noise ratio is defined as

$$\text{SNR} = \text{average signal power} / \text{average noise power}$$

We need to consider the average signal power and the average noise power because these may change with time. Following figure shows the idea of SNR.

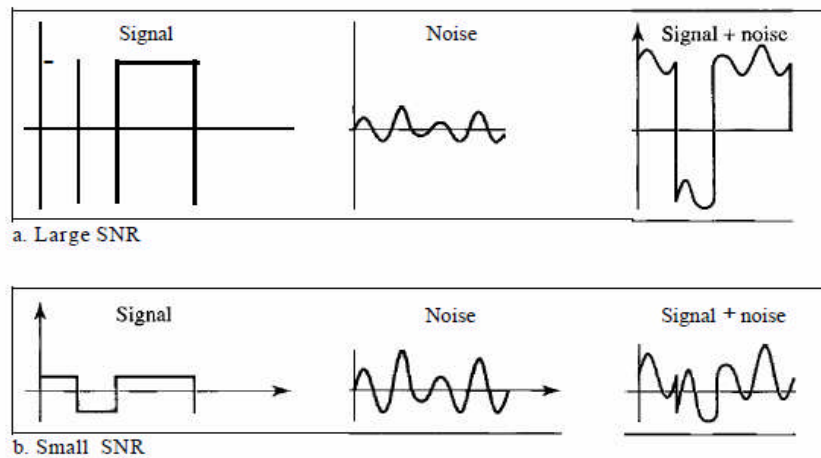


Figure: Two cases of SNR: a high SNR and a low SNR

SNR is actually the ratio of what is wanted (signal) to what is not wanted (noise). A high SNR means the signal is less corrupted by noise; a low SNR means the signal is more corrupted by noise. Because SNR is the ratio of two powers, it is often described in decibel units, SNR_{dB}, defined as

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

3.5 DATA RATE LIMITS

A very important consideration in data communications is how fast we can send data, in bits per second over a channel. Data rate depends on three factors:

1. The bandwidth available
2. The level of the signals we use
3. The quality of the channel (the level of noise)

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

Noiseless Channel: Nyquist Bit Rate

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

$$\text{BitRate} = 2 \times \text{bandwidth} \times \log_2 L$$

In this formula, 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.

According to the formula, we might think that, given a specific bandwidth, we can have any bit rate we want by increasing the number of signal levels. Although the idea is theoretically correct, practically there is a limit. When we increase the number of signal levels, we impose a

burden on the receiver. If the number of levels in a signal is just 2, the receiver can easily distinguish between a 0 and a 1. If the level of a signal is 64, the receiver must be very sophisticated to distinguish between 64 different levels. In other words, increasing the levels of a signal reduces the reliability of the system.

Noisy Channel: Shannon Capacity

In reality, we cannot have a noiseless channel; the channel is always noisy. In 1944, Claude Shannon introduced a formula, called the Shannon capacity, to determine the theoretical highest data rate for a noisy channel:

$$\text{Capacity} = \text{bandwidth} \times \log_2(1 + \text{SNR})$$

In this formula, 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. Note that in the Shannon formula there is no indication of the signal level, which means that no matter how many levels we have, we cannot achieve a data rate higher than the capacity of the channel. In other words, the formula defines a characteristic of the channel, not the method of transmission.

3.6 PERFORMANCE

Up to now, we have discussed the tools of transmitting data (signals) over a network and how the data behave. One important issue in networking is the performance of the network-how good is it? We discuss quality of service, an overall measurement of network performance.

Bandwidth

One characteristic that measures network performance is bandwidth. However, the term can be used in two different contexts with two different measuring values: bandwidth in hertz and bandwidth in bits per second.

Bandwidth in Hertz

We have discussed this concept. Bandwidth in hertz is the range of frequencies contained in a composite signal or the range of frequencies a channel can pass. For example, we can say the bandwidth of a subscriber telephone line is 4 kHz.

Bandwidth in Bits per Seconds

The term *bandwidth* can also refer to the number of bits per second that a channel, a link, or even a network can transmit. For example, one can say the bandwidth of a Fast Ethernet network (or the links in this network) is a maximum of 100 Mbps. This means that this network can send 100 Mbps

Relationship

There is an explicit relationship between the bandwidth in hertz and bandwidth in bits per seconds. Basically, an increase in bandwidth in hertz means an increase in bandwidth in bits per second. The relationship depends on whether we have base band transmission or transmission with modulation.

In networking, we use the term *bandwidth* in two contexts.

The first, *bandwidth in hertz*, refers to the range of frequencies in a composite signal or the range of frequencies that a channel can pass.

The second, *bandwidth in bits per second*, refers to the speed of bit transmission in a channel or link.

Throughput

The throughput is a measure of how fast we can actually send data through a network. Although, at first glance, bandwidth in bits per second and throughput seem the same, they are different. A link may have a bandwidth of B bps, but we can only send T bps through this link with T always less than B . In other words, the bandwidth is a potential measurement of a link; the throughput is an actual measurement of how fast we can send data. For example, we may have a link with a bandwidth of 1 Mbps, but the devices connected to the end of the link may handle only 200 kbps. This means that we cannot send more than 200 kbps through this link.

Imagine a highway designed to transmit 1000 cars per minute from one point to another. However, if there is congestion on the road, this figure may be reduced to 100 cars per minute. The bandwidth is 1000 cars per minute; the throughput is 100 cars per minute.

Latency (Delay)

The latency or delay defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is sent out from the source. We can say that latency is made of four components: propagation time, transmission time, queuing time and processing delay.

$$\text{Latency} = \text{propagation time} + \text{transmission time} + \text{queuing time} + \text{processing delay}$$

Propagation Time

Propagation time measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation speed.

$$\text{Propagation time} = \text{Distance} / \text{Propagation speed}$$

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 mfs. It is lower in air; it is much lower in cable.

Transmission time

In data communications we don't send just 1 bit, we send a message. The first bit may take a time equal to the propagation time to reach its destination; the last bit also may take the same amount of time. However, there is a time between the first bit leaving the sender and the last bit arriving at the receiver. The first bit leaves earlier and arrives earlier; the last bit leaves later and arrives later. The time required for transmission of a message depends on the size of the message and the bandwidth of the channel.

$$\text{Transmission time} = \text{Message size} / \text{Bandwidth}$$

Queuing Time

The third component in latency is the queuing time, the time needed for each intermediate or end device to hold the message before it can be processed. The queuing time is not a fixed factor; it changes with the load imposed on the network. When there is heavy traffic on the network, the queuing time increases. An intermediate device, such as a router, queues the arrived messages and processes them one by one. If there are many messages, each message will have to wait.

3.7 REVIEW QUESTIONS

1. Differentiate between analog and digital signal
2. Explain periodic analog signals in brief
3. Explain the terms
 - a) Wavelength
 - b) Sine Wave
 - c) Bandwidth
4. Explain Digital Signals with suitable
5. What are the causes of impairment transmission?
6. Explain data rate limits with different factors
7. How performance is measured? Explain with suitable example

3.8 SUMMARY

- Data must be transformed to electromagnetic signals to be transmitted.
- Data can be analog or digital. Analog data are continuous and take continuous values. Digital data have discrete states and take discrete values.
- Signals can be analog or digital. Analog signals can have an infinite number of values in a range; digital, signals can have only a limited number of values.
- In data communications, we commonly use periodic analog signals and non-periodic digital signals.
- Frequency and period are the inverse of each other.
- Frequency is the rate of change with respect to time.

- Phase describes the position of the waveform relative to time O.
- A complete sine wave in the time domain can be represented by one single spike in the frequency domain.
- A single-frequency sine wave is not useful in data communications; we need to send a composite signal, a signal made of many simple sine waves.
- According to Fourier analysis, any composite signal is a combination of simple sine waves with different frequencies, amplitudes, and phases.
- The bandwidth of a composite signal is the difference between the highest and the lowest frequencies contained in that signal.
- A digital signal is a composite analog signal with an infinite bandwidth.
- Baseband transmission of a digital signal that preserves the shape of the digital signal is possible only if we have a low-pass channel with an infinite or very wide bandwidth.
- If the available channel is a bandpass channel, we cannot send a digital signal directly to the channel; we need to convert the digital signal to an analog signal before transmission.
- For a noiseless channel, the Nyquist bit rate formula defines the theoretical maximum bit rate. For a noisy channel, we need to use the Shannon capacity to find the maximum bit rate.
- Attenuation, distortion, and noise can impair a signal.
- Attenuation is the loss of a signal's energy due to the resistance of the medium.
- Distortion is the alteration of a signal due to the differing propagation speeds of each of the frequencies that make up a signal.
- Noise is the external energy that corrupts a signal.
- The bandwidth-delay product defines the number of bits that can fill the link.

3.9 REFERENCES

1. Data Communication & Networking – Behrouz Forouzan
2. TCP/IP Protocol Suite – Behrouz Forouzan
3. Computer Networks –Andrew Tanenbaum¹



DIGITAL AND ANALOG TRANSMISSION

Unit Structure

- 4.0 Objectives
- 4.1 Digital-to-digital conversion
- 4.2 Analog-to-digital conversion
- 4.3 Transmission modes
- 4.4 Digital-to-analog conversion
- 4.5 Analog-to-analog conversion
- 4.6 Summary
- 4.7 References

4.0 OBJECTIVES:

This chapter would make you understand the following concepts

- What is the need to conversion of data in different form?
- How the data is converted from one form to another form
- What are the different ways to convert it?
- What are the different transmission modes
- What are the ways of digital to analog conversion?
- And conversion of analog to analog.

4.1 DIGITAL-TO-DIGITAL CONVERSION

As we have discussed in the previous topics data can be either digital or analog. We also saw that signals that represent data can also be digital or analog. In this section, we see how we can represent digital data by using digital signals.

The conversion involves three techniques: **line coding**, **block coding**, and **scrambling**. Line coding is always needed block coding and scrambling may or may not be needed.

4.1.1 Line Coding

Line coding is the process of converting digital data to digital signals. We assume that data, in the form of text, numbers, graphical images, audio, or video, are stored in computer memory as sequences of bits. Line coding converts a sequence of bits to a digital signal. At the sender, digital data are encoded into a digital signal; at the receiver, the digital data are recreated by decoding the digital signal. Following figure shows the process.

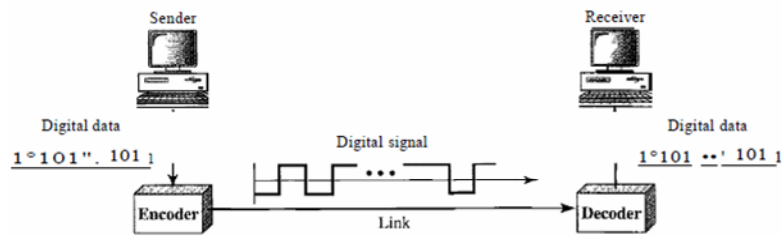


Figure: Line coding and decoding

Characteristics

Before discussing different line coding schemes, we address their common characteristics. **Signal Element Versus Data Element** Let us distinguish between a data element and a signal element. In data communications, our goal is to send data elements. A data element is the smallest entity that can represent a piece of information: this is the bit. In digital data communications, a signal element carries data elements. A signal element is the shortest unit (time wise) of a digital signal. In other words, data elements are what we need to send; signal elements are what we can send. Data elements are being carried; signal elements are the carriers.

We define a ratio r which is the number of data elements carried by each signal element. Following figure shows several situations with different values of r .

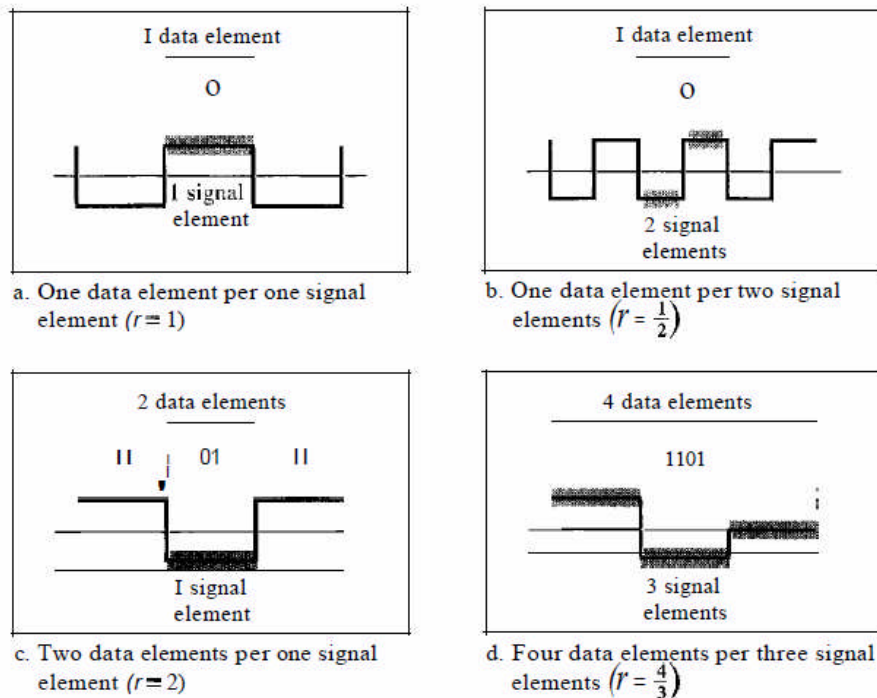


Figure: Signal element versus data element

In part a of the figure, one data element is carried by one signal element ($r = 1$). In part b of the figure, we need two signal elements (two transitions) to carry each data element ($r = \frac{1}{2}$). We will see later that the extra signal element is needed to guarantee synchronization. In part c of

the figure, a signal element carries two data elements ($r = 2$). Finally, in part d, a group of 4 bits is being carried by a group of three signal elements ($r = 4/3$). For every line coding scheme we discuss, we will give the value of r .

An analogy may help here. Suppose each data element IS a person who needs to be carried from one place to another. We can think of a signal element as a vehicle that can carry people. When $r = 1$, it means each person is driving a vehicle. When $r > 1$, it means more than one person is travelling in a vehicle (a carpool, for example). We can also have the case where one person is driving a car and a trailer ($r = 1/2$).

Data Rate Versus Signal Rate The data rate defines the number of data elements (bits) sent in Is. The unit is bits per second (bps). The signal rate is the number of signal elements sent in Is. The unit is the baud. There are several common terminologies used in the literature. The data rate is sometimes called the bit rate; the signal rate is sometimes called the pulse rate, the modulation rate, or the baud rate.

One goal in data communications is to increase the data rate while decreasing the signal rate. Increasing the data rate increases the speed of transmission; decreasing the signal rate decreases the bandwidth requirement. In our vehicle-people analogy, we need to carry more people in fewer vehicles to prevent traffic jams. We have a limited *bandwidth* in our transportation system.

We now need to consider the relationship between data rate and signal rate (bit rate and baud rate). This relationship, of course, depends on the value of r . It also depends on the data pattern. If we have a data pattern of all 1s or all 0s, the signal rate may be different from a data pattern of alternating 0s and 1s. To derive a formula for the relationship, we need to define three cases: the worst, best, and average. The worst case is when we need the maximum signal rate; the best case is when we need the minimum. In data communications, we are usually interested in the average case. We can formulate the relationship between data rate and signal rate as

$$S = c \times N \times \frac{1}{r} \quad \text{baud}$$

where N is the data rate (bps); c is the case factor, which varies for each case; S is the number of signal elements; and r is the previously defined factor.

Bandwidth

The digital signal that carries information is non periodic. We also showed that the bandwidth of a non periodic signal is continuous with an infinite range. However, most digital signals we encounter in real life have a bandwidth with finite values. In other words, the bandwidth is

theoretically infinite, but many of the components have such small amplitude that they can be ignored. The effective bandwidth is finite.

Baseline Wandering

In decoding a digital signal, the receiver calculates a running average of the received signal power. This average is called the *baseline*. The incoming signal power is evaluated against this baseline to determine the value of the data element. A long string of 0s or 1s can cause a drift in the baseline (baseline wandering) and make it difficult for the receiver to decode correctly. A good line coding scheme needs to prevent baseline wandering.

DC Components

When the voltage level in a digital signal is constant for a while, the spectrum creates very low frequencies (results of Fourier analysis). These frequencies around zero, called DC (direct-current) *components*, present problems for a system that cannot pass low frequencies or a system that uses electrical coupling (via a transformer). For example, a telephone line cannot pass frequencies below 200 Hz. Also a long-distance link may use one or more transformers to isolate different parts of the line electrically. For these systems, we need a scheme with no DC component.

Self-synchronization

To correctly interpret the signals received from the sender, the receiver's bit intervals must correspond exactly to the sender's bit intervals. If the receiver clock is faster or slower, the bit intervals are not matched and the receiver might misinterpret the signals. Following figure shows a situation in which the receiver has a shorter bit duration. The sender sends 10110001, while the receiver receives 110111000011.

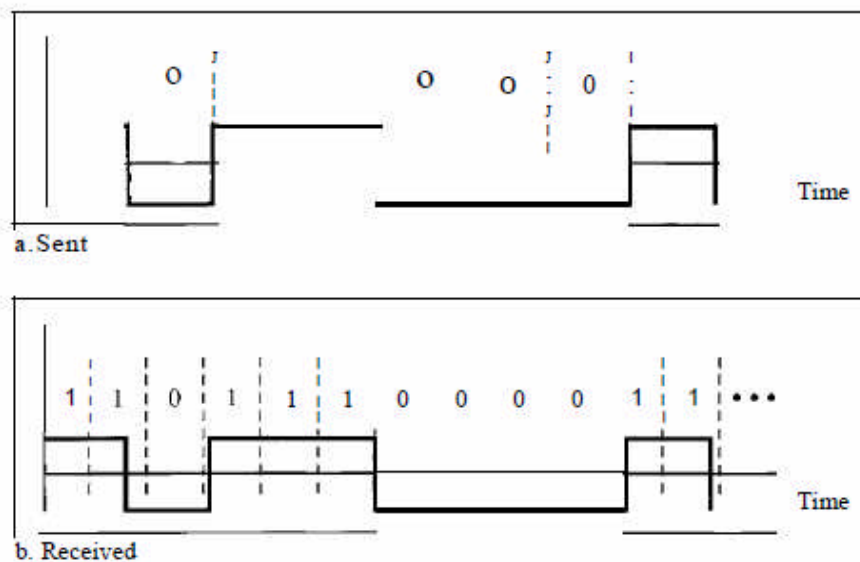


Figure: Effect of lack of synchronization

Built-in Error Detection

It is desirable to have a built-in error-detecting capability in the generated code to detect some of or all the errors that occurred during transmission. Some encoding schemes that we will discuss have this capability to some extent.

Immunity to Noise and Interference

Another desirable code characteristic is a code that is immune to noise and other interferences. Some encoding schemes that we will discuss have this capability.

Complexity

A complex scheme is more costly to implement than a simple one. For example, a scheme that uses four signal levels is more difficult to interpret than one that uses only two levels.

4.1.2 Line Coding Schemes

We can roughly divide line coding schemes into five broad categories, as shown in following figure.

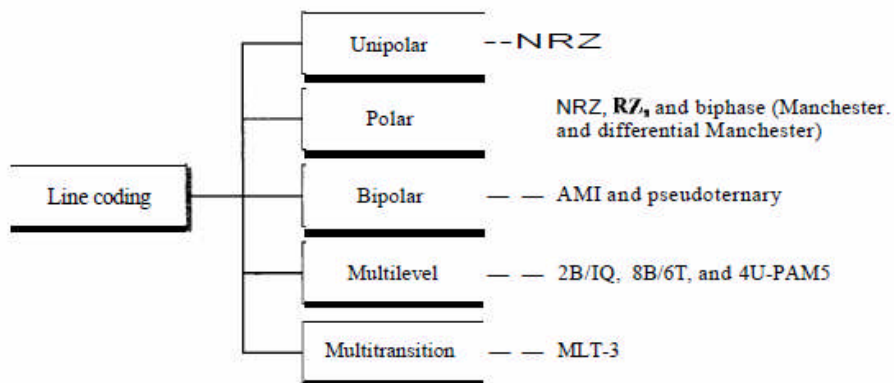


Figure : Line coding schemes

There are several schemes in each category as

In a unipolar scheme, all the signal levels are on one side of the time axis, either above or below.

NRZ (Non-Return-to-Zero) traditionally, a unipolar scheme was designed as a non-return-to-zero (NRZ) scheme in which the positive voltage defines bit 1 and the zero voltage defines bit 0. It is called NRZ because the signal does not return to zero at the middle of the bit. Following figure show a unipolar NRZ scheme.

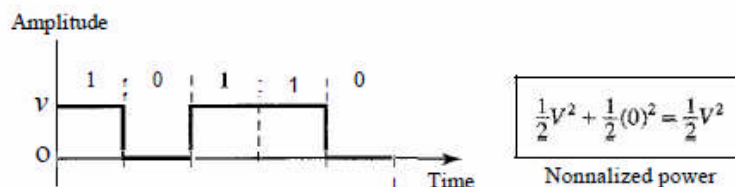


Figure : Unipolar NRZ scheme

Compared with its polar counterpart (see the next section), this scheme is very costly. As we will see shortly, the normalized power (power needed to send 1 bit per unit line resistance) is double that for polar NRZ. For this reason, this scheme is normally not used in data communications today.

Polar Schemes

In polar schemes, the voltages are on the both sides of the time axis. For example, the voltage level for 0 can be positive and the voltage level for 1 can be negative.

Non-Return-to-Zero (NRZ) In polar NRZ encoding, we use two levels of voltage amplitude. We can have two versions of polar NRZ: NRZ-L and NRZ-I, as shown in following figure. The figure also shows the value of r , the average baud rate, and the bandwidth. In the first variation, NRZ-L (NRZ-Level), the level of the voltage determines the value of the bit. In the second variation, NRZ-I (NRZ-Invert), the change or lack of change in the level of the voltage determines the value of the bit. If there is no change, the bit is 0; if there is a change, the bit is 1.

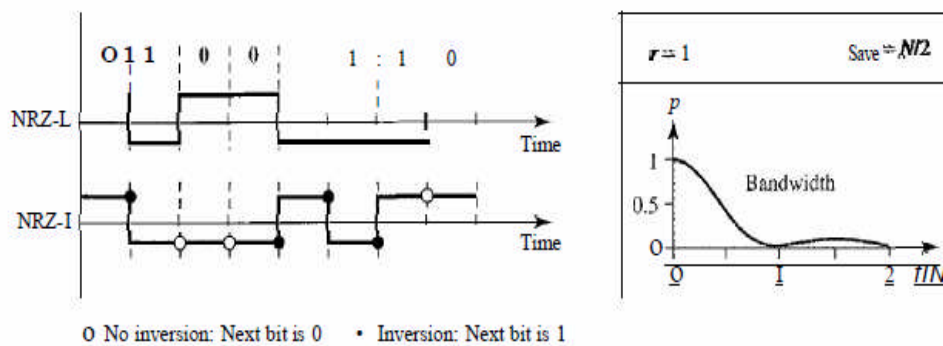


Figure: Polar NRZ-L and NRZ-I schemes

In NRZ-L the level of the voltage determines the value of the bit.
In NRZ-I the inversion or the lack of inversion determines the value of the bit.

When we compare these two schemes based on the criteria

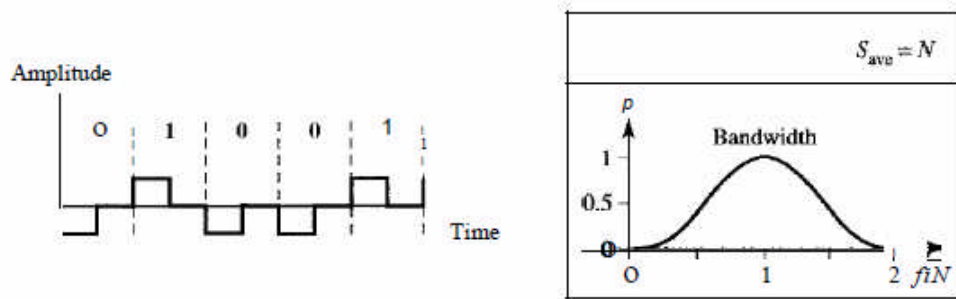
- Although baseline wandering is a problem for both variations, it is twice as severe in NRZ-L. If there is a long sequence of 0s or 1s in NRZ-L, the average signal power becomes skewed. The receiver might have difficulty discerning the bit value. In NRZ-I this problem occurs only for a long sequence of as, If somehow we can eliminate the long sequence of as, we can avoid baseline wandering.
- Another problem with NRZ-L occurs when there is a sudden change of polarity in the system. NRZ-I does not have this problem.

Note Both schemes have an average signal rate of $N/2$ Bd, NRZ-L and NRZ-I both have a DC component problem.

Return to Zero (RZ)

The main problem with NRZ encoding occurs when the sender and receiver clocks are not synchronized. The receiver does not know when one bit has ended and the next bit is starting. One solution is the return-to-zero (RZ) scheme, which uses three values: positive, negative, and zero. In RZ, the signal changes not between bits but during the bit. In following figure we see that the signal goes to 0 in the middle of each bit. It remains there until the beginning of the next bit. The main disadvantage of RZ encoding is that it requires two signal changes to encode a bit and therefore occupies greater bandwidth. The same problem we mentioned, a sudden change of polarity resulting in all as interpreted as 1s and all 1s interpreted as as, still exist here, but there is no DC component problem. Another problem is the complexity: RZ uses three levels of voltage, which is more complex to create and discern. As a result of all these deficiencies, the scheme is not used today. Instead, it has been replaced by the better-performing Manchester and differential Manchester schemes

Figure: Polar RZ scheme



Biphase (Manchester and Differential Manchester)

The idea of RZ (transition at the middle of the bit) and the idea of NRZ-L are combined into the Manchester scheme. In Manchester encoding, the duration of the bit is divided into two halves. The voltage remains at one level during the first half and moves to the other level in the second half. The transition at the middle of the bit provides synchronization. Differential Manchester, on the other hand, combines the ideas of RZ and NRZ-I. There is always a transition at the middle of the bit, but the bit values are determined at the beginning of the bit. If the next bit is 0, there is a transition; if the next bit is 1, there is none. Following figure shows both Manchester and differential Manchester encoding.

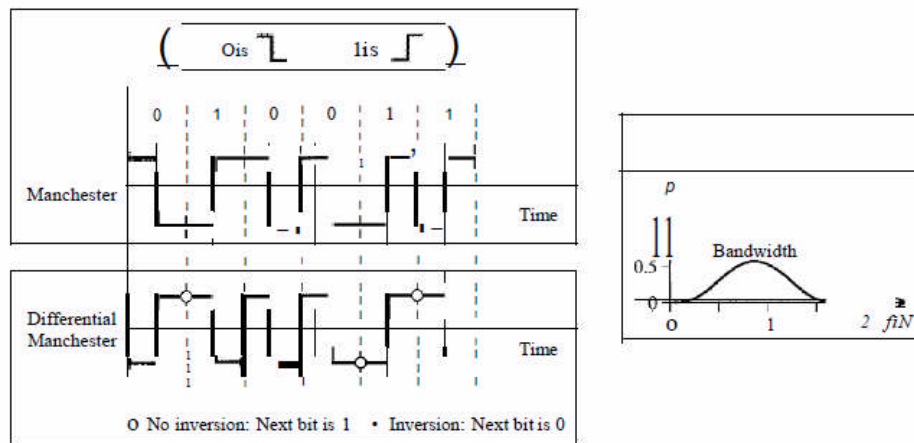


Figure: Polar biphasis: Manchester and differential Manchester schemes

In Manchester and differential Manchester encoding, the transition at the middle of the bit is used for synchronization.

The Manchester scheme overcomes several problems associated with NRZ-L, and differential Manchester overcomes several problems associated with NRZ-I. First, there is no baseline wandering. There is no DC component because each bit has a positive and negative voltage contribution. The only drawback is the signal rate. The signal rate for Manchester and differential Manchester is double that for NRZ. The reason is that there is always one transition at the middle of the bit and maybe one transition at the end of each bit. Above figure shows both Manchester and differential Manchester encoding schemes. Note that Manchester and differential Manchester schemes are also called biphasis schemes.

Bipolar Schemes

In bipolar encoding (sometimes called *multilevel binary*), there are three voltage levels: positive, negative, and zero. The voltage level for one data element is at zero, while the voltage level for the other element alternates between positive and negative.

AMI and Pseudoternary

Following figure shows two variations of bipolar encoding: AMI and pseudoternary. A common bipolar encoding scheme is called bipolar alternate mark inversion (AMI). In the term *alternate mark inversion*, the word *mark* comes from telegraphy and means 1. So AMI means alternate I inversion. A neutral zero voltage represents binary 0. Binary 1s are represented by alternating positive and negative voltages. A variation of AMI encoding is called pseudoternary in which the 1 bit is encoded as a zero voltage and the 0 bit is encoded as alternating positive and negative voltages.

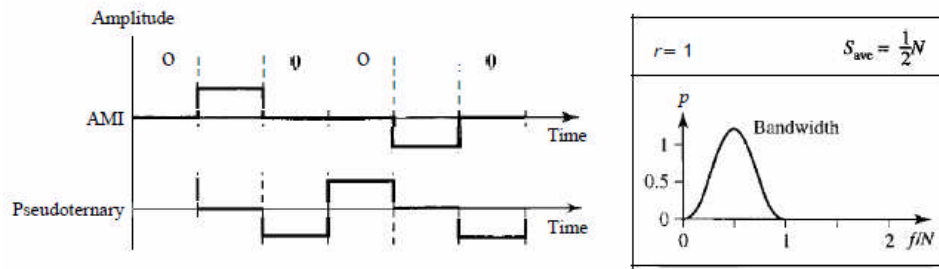


Figure: Bipolar schemes: AMI and pseudoternary

The bipolar scheme was developed as an alternative to NRZ. The bipolar scheme has the same signal rate as NRZ, but there is no DC component. The NRZ scheme has most of its energy concentrated near zero frequency, which makes it unsuitable for transmission over channels with poor performance around this frequency. The concentration of the energy in bipolar encoding is around frequency $N/2$. Above figure shows the typical energy concentration for a bipolar scheme.

One may ask why we do not have DC component in bipolar encoding. We can answer this question by using the Fourier transform, but we can also think about it intuitively. If we have a long sequence of 1s, the voltage level alternates between positive and negative; it is not constant. Therefore, there is no DC component. For a long sequence of 0s, the voltage remains constant, but its amplitude is zero, which is the same as having no DC component. In other words, a sequence that creates a constant zero voltage does not have a DC component.

AMI is commonly used for long-distance communication, but it has a synchronization problem when a long sequence of 0s is present in the data.

Multilevel Schemes

Its goal is to increase the number of bits per baud by encoding a pattern of m data elements into a pattern of n signal elements. Two types of data elements (0s and 1s), which means that a group of m data elements can produce a combination of 2^m data patterns. We can have different types of signal elements by allowing different signal levels. If we have L different levels, then we can produce L^n combinations of signal patterns. If $2^m = L^n$, then each data pattern is encoded into one signal pattern. If $2^m < L^n$, data patterns occupy only a subset of signal patterns. The subset can be carefully designed to prevent baseline wandering, to provide synchronization, and to detect errors that occurred during data transmission.

Data encoding is not possible if $2^m > L^n$ because some of the data patterns cannot be encoded.

Multiline Transmission

NRZ-I and differential Manchester are classified as differential encoding but use two transition rules to encode binary data (no inversion, inversion). If we have a signal with more than two levels, we can design a differential encoding scheme with more than two transition rules. MLT-3 is one of them. The multiline transmission, three level (MLT-3) scheme uses three levels ($+V$, 0 , and $-V$) and three transition rules to move between the levels.

- If the next bit is 0 , there is no transition.
- If the next bit is 1 and the current level is not 0 , the next level is 0 .
- If the next bit is 1 and the current level is 0 , the next level is the opposite of the last non zero level.

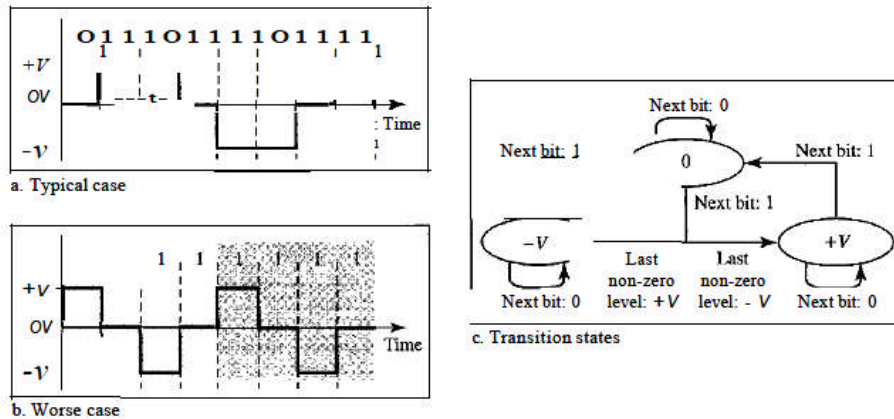


Figure: Multi-transition: MLT-3 scheme

4.1.3 Block Coding

We need redundancy to ensure synchronization and to provide some kind of inherent error detecting. Block coding can give us this redundancy and improve the performance of line coding. In general, block coding changes a block of m bits into a block of n bits, where n is larger than m . Block coding is referred to as an mB/nB encoding technique.

The slash in block encoding (for example, 4B/5B) distinguishes block encoding from multilevel encoding (for example, 8B6T), which is written without a slash. Block coding normally involves three steps: division, substitution, and combination. In the division step, a sequence of bits is divided into groups of m bits. For example, in 4B/5B encoding, the original bit sequence is divided into 4-bit groups. The heart of block coding is the substitution step. In this step, we substitute an m -bit group for an n -bit group. For example, in 4B/5B encoding we substitute a 4-bit code for a 5-bit group. Finally, the n -bit groups are combined together to form a stream. The new stream has more bits than the original bits. Following figure shows the procedure.

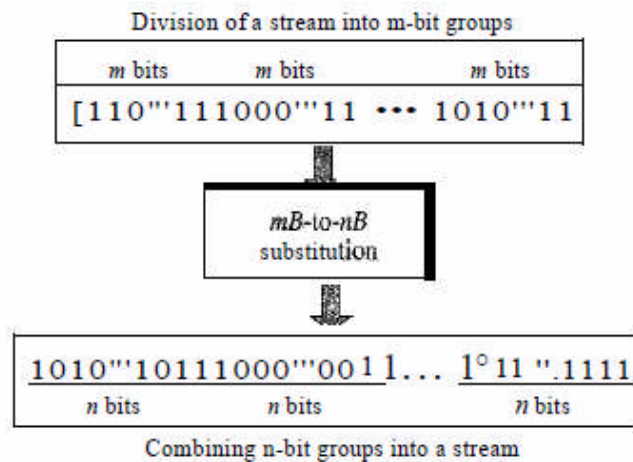


Figure: Block coding concept

4.1.4 Scrambling

Biphase schemes that are suitable for dedicated links between stations in a LAN are not suitable for long-distance communication because of their wide bandwidth requirement. The combination of block coding and NRZ line coding is not suitable for long-distance encoding either, because of the DC component. Bipolar AMI encoding, on the other hand, has a narrow bandwidth and does not create a DC component. However, a long sequence of Os upsets the synchronization. If we can find a way to avoid a long sequence of Os in the original stream, we can use bipolar AMI for long distances. We are looking for a technique that does not increase the number of bits and does provide synchronization. We are looking for a solution that substitutes long zero-level pulses with a combination of other levels to provide synchronization. One solution is called scrambling. We modify part of the AMI rule to include scrambling, as shown in following figure. Note that scrambling, as opposed to block coding, is done at the same time as encoding. The system needs to insert the required pulses based on the defined scrambling rules. Two common scrambling techniques are B8ZS and HDB3.

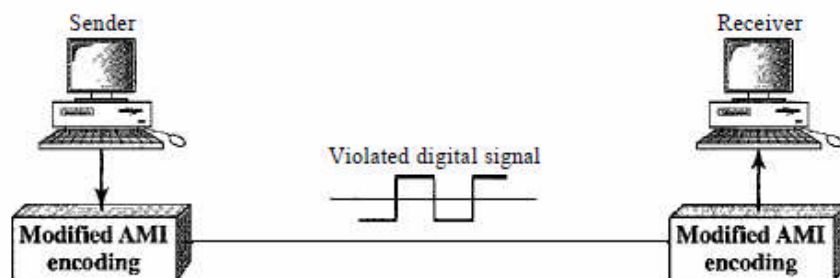


Figure: AMI used with scrambling

R8ZS

Bipolar with S-zero substitution (BSZS) is commonly used in North America. In this technique, eight consecutive zero-level voltages are replaced by the sequence OOOVBOVB.

The V in the sequence denotes *violation*; this is a nonzero voltage that breaks an AMI rule of encoding (opposite polarity from the previous). The B in the sequence denotes *bipolm*; which means a nonzero level voltage in accordance with the AMI rule. There are two cases, as shown in following figure.

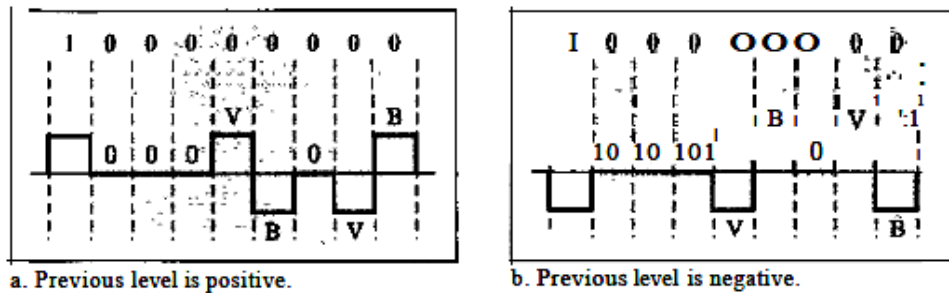


Figure: Two cases of B8ZS scrambling technique

HDB3

High-density bipolar 3-zero (HDB3) is commonly used outside of North America. In this technique, which is more conservative than B8ZS, four consecutive zero-level voltages are replaced with a sequence of OOOV or BOOV. The reason for two different substitutions is to maintain the even number of nonzero pulses after each substitution. The two rules can be stated as follows

- If the number of nonzero pulses after the last substitution is odd, the substitution pattern will be OOOV, which makes the total number of nonzero pulses even.
- If the number of nonzero pulses after the last substitution is even, the substitution pattern will be BOOV, which makes the total number of nonzero pulses even.

Following figure shows the example of HDB3

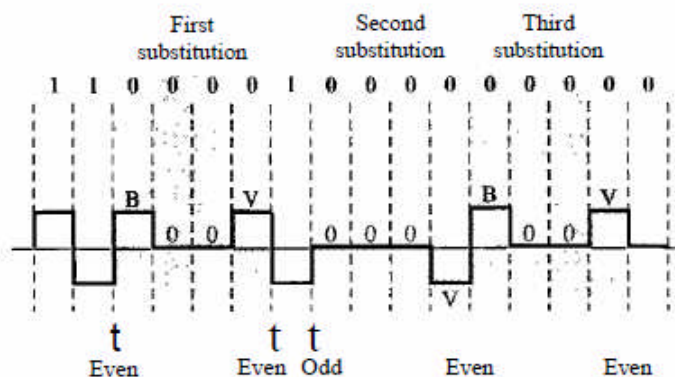


Figure: Different situations in HDB3 scrambling technique

4.2 ANALOG-TO-DIGITAL CONVERSION

Sometimes, we have an analog signal such as one created by a microphone or camera. We have seen in previous chapter that a digital signal is superior to an analog signal. The tendency today is to change an analog signal to digital data. For conversion two techniques are used, pulse code modulation and delta modulation. After the digital data are created (digitization), we can use one of the techniques described of line coding to convert the digital data to a digital signal.

4.2.1 Pulse Code Modulation (PCM)

To convert analog wave into digital data we use Pulse Code Modulation. Pulse Code Modulation is one of the most commonly used method to convert analog data into digital form. It involves three steps: Sampling, Quantization and Encoding.

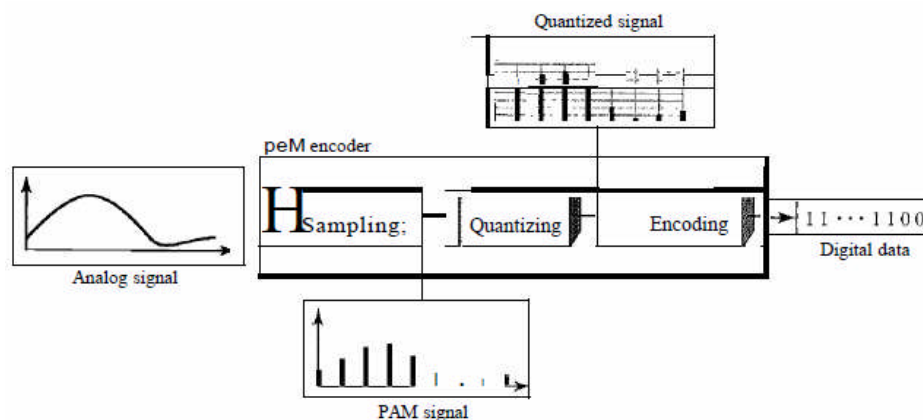


Figure Components of PCM encoder

Steps involved are

1. The analog signal is sampled.
2. The sampled signal is quantized.
3. The quantized values are encoded as streams of bits.

Sampling

The sampling process is sometimes referred to as pulse amplitude modulation (PAM). But, that result is still an analog signal with non integral values.

- The first step in PCM is sampling.
- The analog signal is sampled every T_s s, where T_s is the sample interval or period. The inverse of the sampling interval is called the sampling rate or sampling frequency.
- There are three sampling methods: Ideal, Natural, Flat-top

In **ideal sampling**, pulses from the analog signal are sampled. This is an ideal sampling method and cannot be easily implemented.

In **natural sampling**, a high-speed switch is turned on for only the small period of time when the sampling occurs. The result is a sequence of samples that retains the shape of the analog signal.

The most common sampling method, called **sample and hold**, however, creates **flat-top** samples by using a circuit.

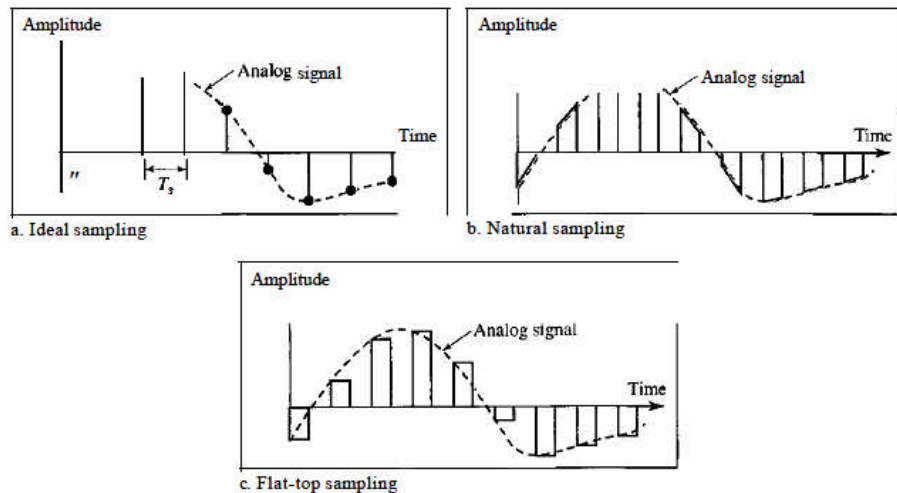


Figure: Three different sampling methods for PCM

Delta Modulation (DM)

PCM is a very complex technique. Other techniques have been developed to reduce the complexity of PCM. The simplest is *delta modulation*. PCM finds the value of the signal amplitude for each sample; DM finds the change from the previous sample. Following figure shows the process. Note that there are no code words here; bits are sent one after another.

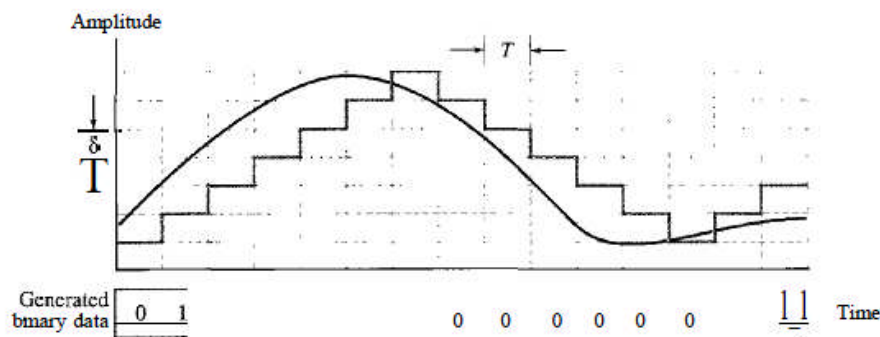


Figure: The process of delta modulation

Modulator

The modulator is used at the sender site to create a stream of bits from an analog signal. The process records the small positive or negative changes, called delta δ . If the delta is positive, the process records a 1; if it is negative, the process records a 0. However, the process needs a base against which the analog signal is compared. The modulator builds a

second signal that resembles a staircase. Finding the change is then reduced to comparing the input signal with the gradually made staircase signal. Following figure shows a diagram of the process.

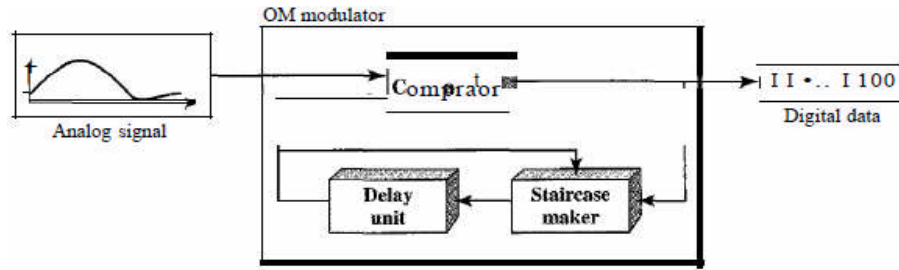


Figure: Delta modulation components

The modulator, at each sampling interval, compares the value of the analog signal with the last value of the staircase signal. If the amplitude of the analog signal is larger, the next bit in the digital data is 1; otherwise, it is 0. The output of the comparator, however, also makes the staircase itself. If the next bit is 1, the staircase maker moves the last point of the staircase signal δ up; if the next bit is 0, it moves it δ down. Note that we need a delay unit to hold the staircase function for a period between two comparisons.

Demodulator

The demodulator takes the digital data and, using the staircase maker and the delay unit, creates the analog signal. The created analog signal, however, needs to pass through a low-pass filter for smoothing. Following figure shows the schematic diagram.

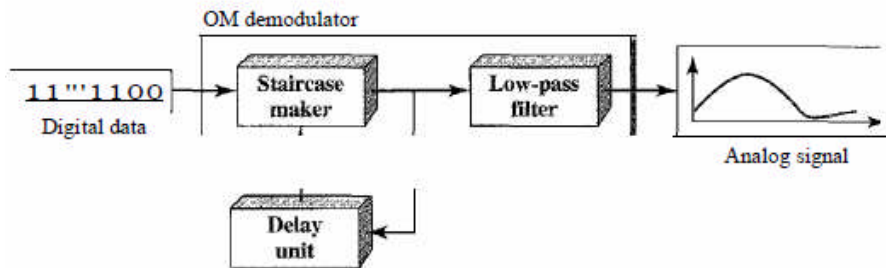


Figure: Delta demodulation components

Adaptive DM

A better performance can be achieved if the value of δ is not fixed. In adaptive delta modulation, the value of δ changes according to the amplitude of the analog signal.

Quantization Error

It is obvious that DM is not perfect. Quantization error is always introduced in the process. The quantization error of DM, however, is much less than that for PCM.

4.3 TRANSMISSION MODES

The transmission of data from one device to another is the wiring, and of primary concern when we are considering the wiring is the data stream. The transmission of binary data (0 and 1) across a link can be accomplished in either **parallel** or **serial** mode.

- In **parallel** mode, multiple bits are sent with each clock tick.
- In **serial** mode, 1 bit is sent with each clock tick. While there is only one way to send parallel data, there are three sub classes of serial transmission: **asynchronous**, **synchronous**, and **isochronous**.

4.3.1 Parallel Transmission

Binary data, consisting of 1s and 0s, may be organized into groups of n bits each. Computers produce and consume data in groups of bits. By grouping, we can send data n bits at a time instead of 1. This is called parallel transmission.

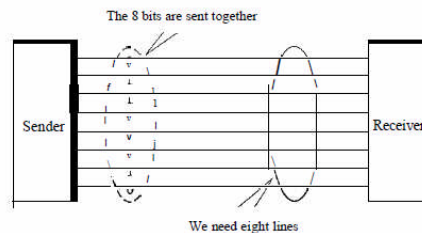


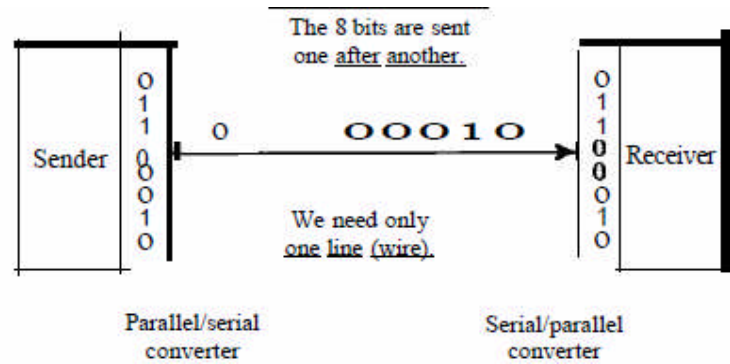
Figure: Parallel transmission

- The mechanism for parallel transmission is a conceptually simple one: Use n wires to send n bits at one time.
- 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.
- The advantage of parallel transmission is **speed**.
- That is parallel transmission can increase the transfer speed by a factor of n over serial transmission.

But there is a significant **disadvantage is cost**: Parallel transmission requires n communication lines (wires in the example) just to transmit the data stream. Because this is expensive, parallel transmission is usually limited to short distances.

4.3.2 Serial Transmission

In serial transmission one bit follows another, so we need only one communication channel rather than n to transmit data between two communicating devices.



In serial transmission one bit follows another, so we need only one communication channel rather than n to transmit data between two communicating devices.

The advantage of serial over parallel transmission is that with only one communication channel, serial transmission reduces the cost of transmission over parallel by roughly a factor of n . Since communication within devices is parallel, conversion devices are required at the interface between the sender and the line (parallel-to-serial) and between the line and the receiver (serial-to parallel).

Serial transmission occurs in one of three ways: **asynchronous, synchronous, and isochronous.**

Asynchronous

Asynchronous transmission is so named because the timing of a signal is unimportant. Instead, information is received and translated by agreed upon patterns. Patterns are based on grouping the bit stream into bytes. Each group, usually 8 bits, is sent along the link as a unit. The sending system handles each group independently, relaying it to the link whenever ready, without regard to a timer. Without synchronization, the receiver cannot use timing to predict when the next group will arrive. To alert the receiver to the arrival of a new group, therefore, an extra bit is added to the beginning of each byte. This bit, usually a 0, is called the start bit. To let the receiver know that the byte is finished, 1 or more additional bits are appended to the end of the byte. These bits, usually 1s, are called stop bits. The start and stop bits and the gap alert the receiver to the beginning and end of each byte allow it to synchronize with the data stream.

This mechanism is called *asynchronous* because, at the byte level, the sender and receiver do not have to be synchronized. But within each byte, the receiver must still be synchronized with the incoming bit stream.

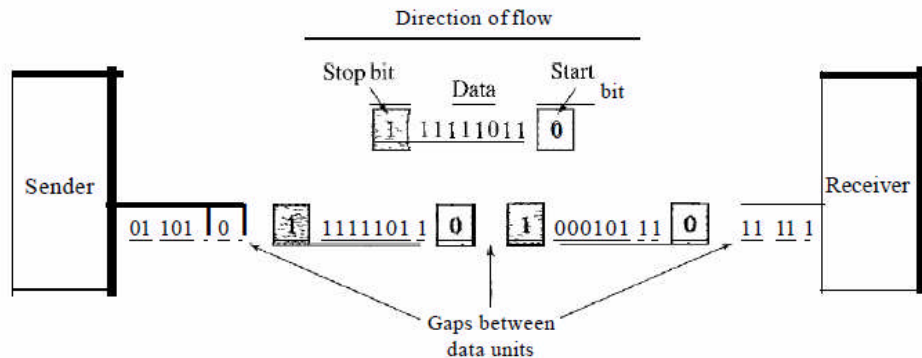


Figure: Asynchronous transmission

Synchronous Transmission

In synchronous transmission, the bit stream is combined into longer "frames," which may contain multiple bytes. Each byte is introduced onto the transmission link without a gap between it and the next one. It is left to the receiver to separate the bit stream into bytes for decoding purposes.

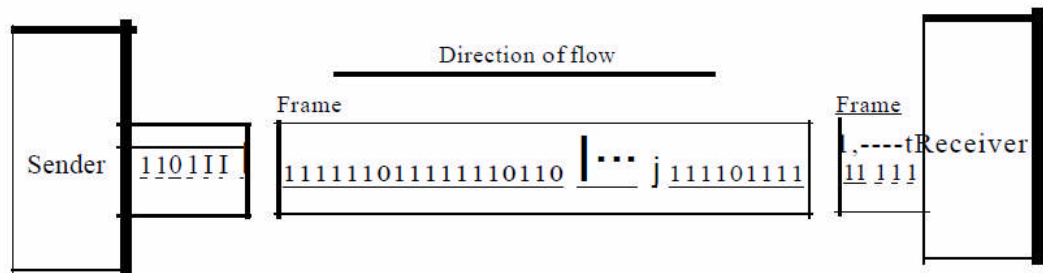


Figure: Synchronous Transmission

In other words, data are transmitted as an unbroken string of 1s and 0s, and the receiver separates that string into the bytes, or characters, it needs to reconstruct the information.

- The advantage of synchronous transmission is speed.
- With no extra bits or gaps to introduce at the sending end and remove at the receiving end, and, by extension, with fewer bits to move across the link.
- Synchronous transmission is faster than asynchronous transmission.
- For this reason, it is more useful for high-speed applications such as the transmission of data from one computer to another.
- Byte synchronization is accomplished in the data link layer.
- Although there is no gap between characters in synchronous serial transmission, there may be uneven gaps between frames.

Isochronous

In real-time audio and video, in which uneven delays between frames are not acceptable synchronous transmission fails. For example, TV images are broadcast at the rate of 30 images per second; they must be

viewed at the same rate. If each image is sent by using one or more frames, there should be no delays between frames. For this type of application, synchronization between characters is not enough; the entire stream of bits must be synchronized. The **isochronous** transmission guarantees that the data arrive at a fixed rate.

4.4 DIGITAL-TO-ANALOG CONVERSION

Digital-to-analog conversion is the process of changing one of the characteristics of an analog signal based on the information in digital data. Following figure shows the relationship between the digital information, the digital-to-analog modulating process, and the resultant analog signal.

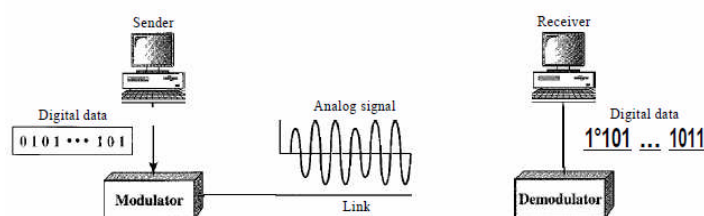


Figure: Digital-to-analog conversion

The above figure shows the relationship between the digital information, the digital-to-analog modulating process, and the resultant analog signal.

We have discussed that sine wave is defined by three characteristics: amplitude, frequency, and phase. When we vary anyone of these characteristics, we create a different version of that wave. So, by changing one characteristic of a simple electric signal, digital data is represented. Any of the three characteristics can be altered in least three mechanisms for modulating digital data into an analog signal: amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK). In addition, there is a fourth (and better) mechanism that combines changing both the amplitude and phase, called quadrature amplitude modulation (QAM). QAM is the most efficient of these options and is the mechanism commonly used today.

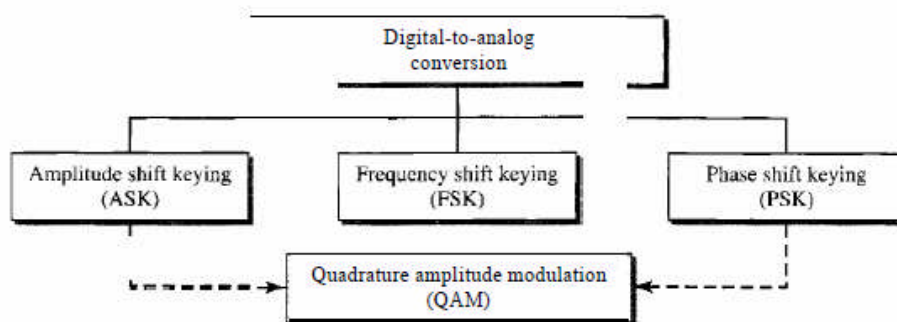


Figure: Types of digital-to-analog conversion

4.4.1 Aspects of Digital-to-Analog Conversion

Before we discuss specific methods of digital-to-analog modulation, two basic issues must be reviewed: bit and baud rates and the carrier signal.

i) Data Element vs Signal Element: we have discussed the data element as the smallest piece of information to be exchanged, the bit and the signal element are also as the smallest unit of a signal that is constant. These terms are only little difference in digital to analog conversion.

ii) Data Rate vs Signal Rate (Bit rate vs Baud rate): Bit rate is the number of bits transmitted during 1 sec. Baud rate refer to the number of signal units per second that are required to represent those bits. Relationship between these two are:

$$\text{Baud Rate} = \text{Bit Rate} / \text{Number of Bits Per Signal Unit}$$

In transportation, a baud is analogous to a vehicle, and a bit is analogous to a passenger.

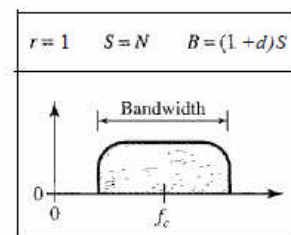
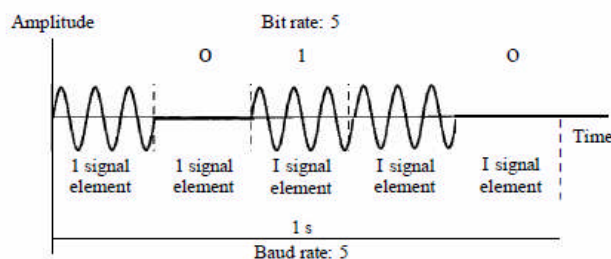
iii) Bandwidth: The required bandwidth for analog transmission of digital data is proportional to the signal rate except for FSK, in which the difference between the carrier signals needs to be added.

iv) Carrier Signal: In analog transmission, the sending device produces a high frequency signal that acts as a **base** for the information signal. This base signal is called the **carrier signal or carrier frequency**. The receiving device is turned to the frequency of the carrier signal that it expects from the sender. Digital information then changes the carrier signal by modifying one or more of its characteristics (amplitude, frequency, or phase). This kind of modification is called **modulation (shift keying)**.

4.4.2 Amplitude shift keying

In amplitude shift keying, the amplitude of the carrier signal is varied to create signal elements. Both frequency and phase remain constant while the amplitude changes.

- ASK is normally implemented using only two levels.
- This is referred to as binary amplitude shift keying or **on-off keying** (OOK). The peak amplitude of one signal level is 0, the other is the same as the amplitude of the carrier frequency.



Bandwidth for ASK: the bandwidth is proportional to the signal rate (baud rate). However, there is normally another factor involved, called d , which depends on the modulation and filtering process. The value of d is between 0 and 1. the relationship can be expressed as

$$B = (1 + d) \times S(N_{\text{baud}})$$

Where, B is the bandwidth, S/N_{baud} is the baud rate and d is the factor related to the modulation process (with minimum value 0).

4.4.3 Frequency Shift Keying

In frequency shift keying, the frequency of the carrier signal is varied to represent data. The frequency of the modulated signal is constant for the duration of one signal element, but changes for the next signal element if the data element changes. Both **peak amplitude** and **phase** remain constant for all signal elements.

Binary FSK (BFSK)

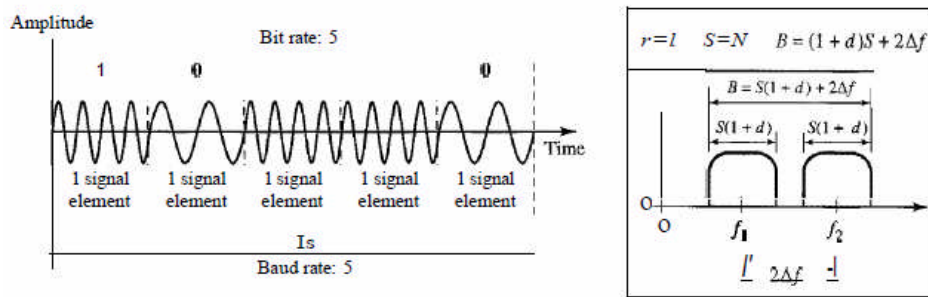


Figure: Binary frequency shift keying

Here binary FSK (or BFSK) is to consider two carrier frequencies. In the above figure we have selected two carrier frequencies, f_1 and f_2 .

- It is assumed for first carrier the data element is 0 for second data element is 1.
- However, note that this is an unrealistic example used only for demonstration purposes.
- Normally the carrier frequencies are **very high**, and the difference between them is very **small**.

Here the middle of one bandwidth is f_1 and the middle of the other is f_2 . Both f_1 and f_2 are Δf apart from the midpoint between the two bands. The difference between the two frequencies is $2\Delta f$.

Bandwidth for BFSK

Carrier signals are only simple sine waves, but the modulation creates a non periodic composite signal with continuous frequencies. For FSK it can think as two ASK signals, each with its own carrier frequency f_2 . If the difference between the two frequencies is $2\Delta f$, then the required bandwidth is

$$B = (1 + d) \times S + 2\Delta f$$

Where, B is the bandwidth, S is the baud rate and d is the factor related to the modulation process (with minimum value 0) and $2\Delta f$ is the frequency difference.

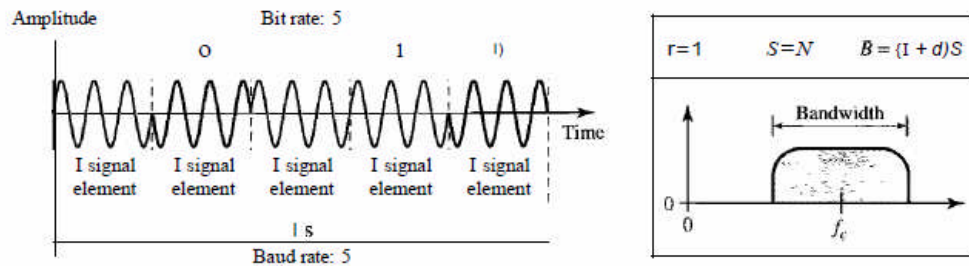
4.4.4 Phase Shift Keying

In phase shift keying, the phase of the carrier is varied to represent two or more different signal elements 0 and 1. Both peak amplitude and frequency remain constant as the phase changes. The phase of the signal during each bit duration is constant and its value depends on the bits (0 and 1). Today, PSK is more common than ASK or FSK. However QAM, which combines ASK and PSK, is the dominant method of digital-to-analog modulation.

Binary PSK (BPSK)

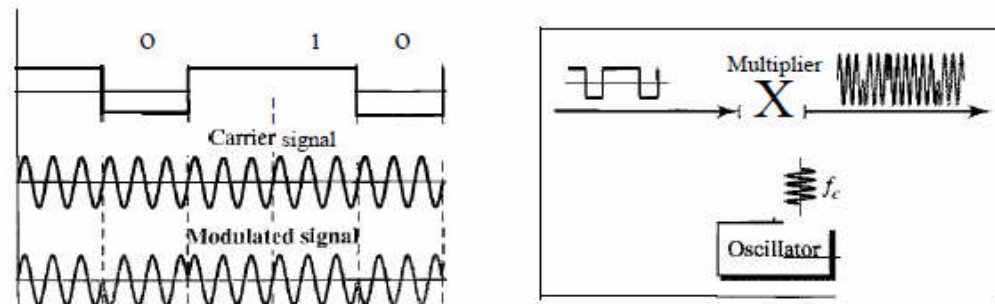
The simplest PSK is binary PSK, in which we have used only two signal elements, one with a phase of 0° , and the other with a phase of 180° . Below figure shows a conceptual view of PSK and relationship of phase to bit value.

Binary PSK is as simple as binary ASK with one big advantage—it is less susceptible to noise.



Bandwidth

The bandwidth for BFSK is the same as that for binary ASK, but less than that for BFSK. Here no bandwidth is **wasted** for separating two carrier signals.



The implementation of BPSK is as simple as that for ASK. The reason is that the signal element with phase 180° can be seen as the complement of the signal element with phase 0° . This gives us a clue on how to implement BPSK. Here it has been used same idea used for ASK but with a polar NRZ signal instead of a uni polar NRZ signal. The polar NRZ signal is multiplied by the carrier frequency, the 1 bit (positive voltage) is represented by a phase starting at 0° , the a bit (negative voltage) is represented by a phase starting at 180° .

4.4.5 Quadrature Amplitude Modulation

Quadrature Amplitude Modulation is the idea of using two carriers, one in-phase and the other quadrature, with different amplitude levels for each carrier.

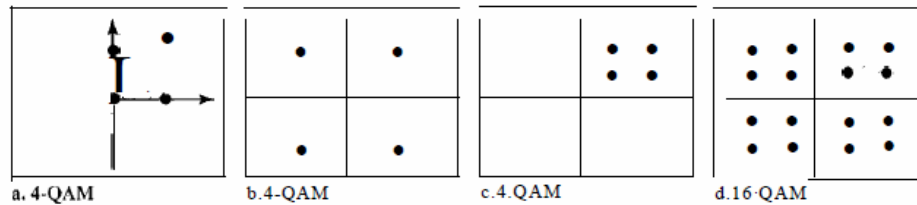


Figure: Constellation diagrams for some QAMs

The possible variations of QAM are numerous. Below shows some of these schemes where 4-QAM scheme (four different signal element types) is a simplest one using a unipolar NRZ signal to modulate each carrier. This is the same mechanism used for ASK (OOK).

Similarly Part b shows another 4-QAM using polar NRZ, but this is exactly the same as QPSK. Part c shows another QAM-4 in which we used a signal with two positive levels to modulate each of the two carriers.

Finally, figure shows a 16-QAM constellation of a signal with eight levels, four positive and four negative.

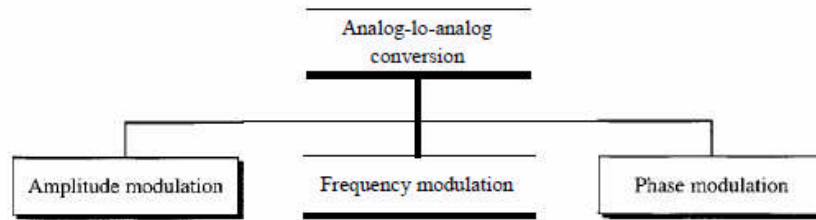
Bandwidth for QAM

The minimum bandwidth required for QAM transmission is the **same** as that required for ASK and PSK transmission. QAM has the same **advantages** as PSK over ASK.

4.5 ANALOG-TO-ANALOG CONVERSION

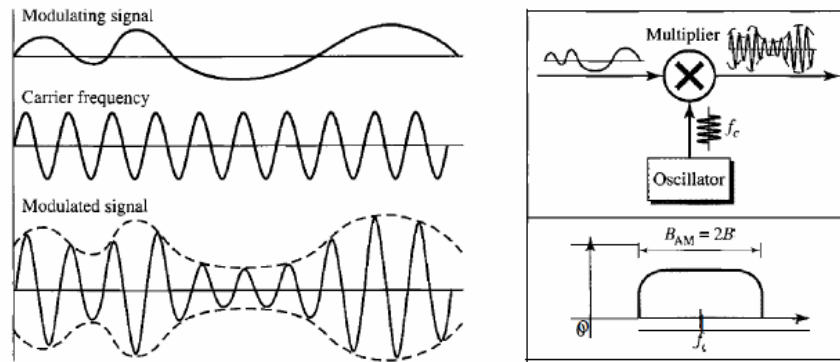
Analog-to-analog conversion, or analog modulation, is the representation of analog information by an analog signal. This modulation is needed if the medium is **band pass** in nature or if only a **band pass channel** is available to us. An example is **radio**. The government assigns a narrow bandwidth to each radio station. The analog signal produced by each station is a **low-pass signal**, all in the same range. To be able to listen to different stations, the low-pass signals need to be **shifted**, each to a different range.

Analog-to-analog conversion can be accomplished in three ways: amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM).



4.5.1 Amplitude Modulation

In AM transmission, the carrier signal is modulated so that its amplitude varies with the changing amplitudes of the modulating (audio) signal. The frequency and phase of the carrier remain the same, only the amplitude changes to follow variations in the information. Where the modulating signal is act as the envelope of the carrier.



AM is normally implemented by using a simple multiplier because the amplitude of the carrier signal needs to be changed according to the amplitude of the modulating signal (audio).

AM Bandwidth

The modulation creates a bandwidth that is twice the bandwidth of the modulating signal and covers a range centered on the carrier frequency. However, the signal components above and below the carrier frequency carry exactly the same information. For this reason, some implementations discard one-half of the signals and cut the bandwidth in half.

The total bandwidth required for **AM** can be determined from the bandwidth of the audio signal:

$$B_{AM} = 2B_m$$

Where, B_{AM} **bandwidth of AM signal** and B_m is bandwidth of modulating signal

4.5.2 Frequency Modulation

In FM transmission, the **frequency of the carrier signal** is modulated to follow the changing voltage level (amplitude) of the modulating signal. The **peak amplitude** and **phase** of the carrier signal remain constant, but as the **amplitude** of the information signal changes, the frequency of the carrier changes correspondingly.

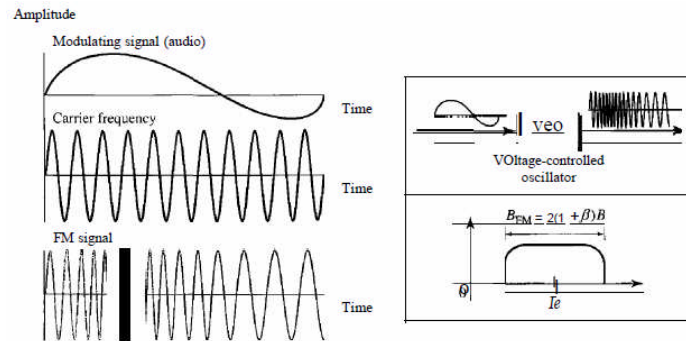


Figure: Frequency Modulation

The actual bandwidth is difficult to determine exactly, but it can be shown empirically that it is several times that of the analog signal or $2(1 + \beta)B$ where β is a factor depends on modulation technique with a common value of 4. In some books it is given that:

FM Bandwidth

The total bandwidth required for FM can be determined from the bandwidth of the **audio** signal as: **$B_{FM} = 10 \times B_m$** Where, B_{FM} is bandwidth of FM signal and B_m is the bandwidth of modulating signal.

4.5.3 Phase Modulation

In **PM** transmission, the phase of the carrier signal is modulated to follow the changing voltage level (**amplitude**) of the modulating signal. The **peak amplitude** and **frequency** of the carrier signal remain constant, but as the amplitude of the information signal changes, the **phase** of the carrier changes correspondingly.

It has been proved that PM is the same as FM with one difference. In FM, the instantaneous change in the **carrier frequency** is proportional to the **amplitude** of the modulating signal where as in PM the instantaneous change in the carrier frequency is proportional to the derivative of the **amplitude** of the modulating signal.

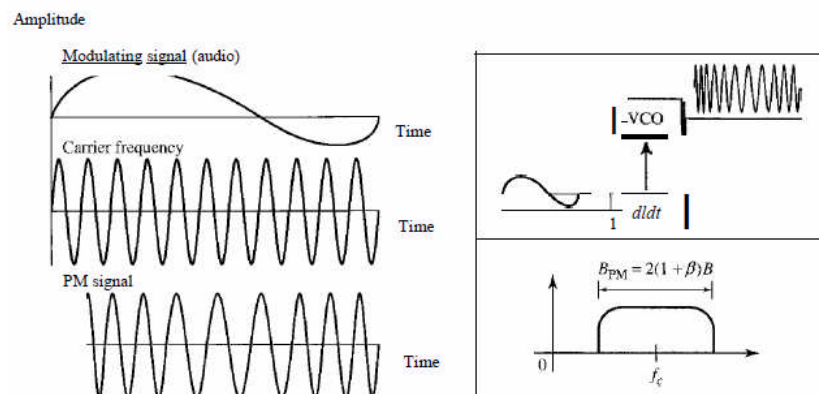


Figure: Phase modulation

As the above figure shows, PM is normally implemented by using a voltage-controlled oscillator along with a derivative. The frequency of

the oscillator changes according to the derivative of the input voltage which is the amplitude of the modulating signal.

PM Bandwidth

The actual bandwidth is difficult to determine exactly, but it can be shown empirically that it is several times that of the analog signal. Although, the formula shows the same bandwidth for FM and PM, the value of β is lower in the case of PM (around 1 for narrowband and 3 for wideband).

The total bandwidth required for PM can be determined from the bandwidth and maximum amplitude of the modulating signal:

$$B_{pm} = 2(1 + \beta)B.$$

4.6 REVIEW QUESTIONS

1. What are the three techniques of digital-to-digital conversion?
2. Enlist different line coding schemes.
3. Explain block coding and give its purpose.
4. Define scrambling and give its purpose.
5. Explain analog transmission.
6. Explain digital-to-analog conversion.
7. Which of the four digital-to-analog conversion techniques (ASK, FSK, PSK or QAM) is the most susceptible to noise? Why explain it?
8. Define constellation diagram and its role in analog transmission.
9. Explain analog-to-analog conversion in brief.
10. Which of the three analog-to-analog conversion techniques (AM, FM, or PM) is the most susceptible to noise? Why explain it?
11. Distinguish between
 - Signal element and a data element.
 - Data rate and signal rate.
 - Parallel and serial transmission

4.7 SUMMARY

- Digital-to-digital conversion involves three techniques: line coding, block coding, and scrambling.
- Line coding is the process of converting digital data to a digital signal.
- We can roughly divide line coding schemes into five broad categories: unipolar, polar, bipolar, multilevel, and multitransition.
- Block coding provides redundancy to ensure synchronization and inherent error detection. Block coding is normally referred to as mB/nB coding; it replaces each m -bit group with an n -bit group.

- Scrambling provides synchronization without increasing the number of bits. Two common scrambling techniques are B8ZS and HDB3.
- The most common technique to change an analog signal to digital data (digitization) is called pulse code modulation (PCM).
- The first step in PCM is sampling. The analog signal is sampled every T_s s, where T_s is the sample interval or period. The inverse of the sampling interval is called the *sampling rate* or *sampling frequency* and denoted by f_s , where $f_s = 1/T_s$. There are three sampling methods-ideal, natural, and flat-top.
- According to the *Nyquist theorem*, to reproduce the original analog signal, one necessary condition is that the *sampling rate* be at least twice the highest frequency in the original signal.
- The simplest is *delta modulation*. PCM finds the value of the signal amplitude for each sample; DM finds the change from the previous sample.
- While there is only one way to send parallel data, there are three subclasses of serial transmission: asynchronous, synchronous, and isochronous.
- In asynchronous transmission, we send 1 start bit (0) at the beginning and 1 or more stop bits (1 s) at the end of each byte.
- In synchronous transmission, we send bits one after another without start or stop bits or gaps. It is the responsibility of the receiver to group the bits.
- The isochronous mode provides synchronized for the entire stream of bits must. In other words, it guarantees that the data arrive at a fixed rate.
- Digital-to-analog conversion is the process of changing one of the characteristics
- of an analog signal based on the information in the digital data.
- Digital-to-analog conversion can be accomplished in several ways: amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK). Quadrature amplitude modulation (QAM) combines ASK and PSK.
- In amplitude shift keying, the amplitude of the carrier signal is varied to create signal elements. Both frequency and phase remain constant while the amplitude changes.
- In frequency shift keying, the frequency of the carrier signal is varied to represent data. The frequency of the modulated signal is constant for the duration of one signal element, but changes for the next signal element if the data element changes. Both peak amplitude and phase remain constant for all signal elements.

- In phase shift keying, the phase of the carrier is varied to represent two or more different signal elements. Both peak amplitude and frequency remain constant as the phase changes.
- A constellation diagram shows us the amplitude and phase of a signal element, particularly when we are using two carriers (one in-phase and one quadrature).
- Quadrature amplitude modulation (QAM) is a combination of ASK and PSK. QAM uses two carriers, one in-phase and the other quadrature, with different amplitude levels for each carrier.
- Analog-to-analog conversion is the representation of analog information by an analog signal. Conversion is needed if the medium is bandpass in nature or if only a bandpass bandwidth is available to us.
- Analog-to-analog conversion can be accomplished in three ways: amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM).
- In AM transmission, the carrier signal is modulated so that its amplitude varies with the changing amplitudes of the modulating signal. The frequency and phase of the carrier remain the same; only the amplitude changes to follow variations in the information.
- In PM transmission, the frequency of the carrier signal is modulated to follow the changing voltage level (amplitude) of the modulating signal. The peak amplitude and phase of the carrier signal remain constant, but as the amplitude of the information signal changes, the frequency of the carrier changes correspondingly.
- In PM transmission, the phase of the carrier signal is modulated to follow the changing voltage level (amplitude) of the modulating signal. The peak amplitude and frequency of the carrier signal remain constant, but as the amplitude of the information signal changes, the phase of the carrier changes correspondingly.

4.8 REFERENCES

1. Data Communication & Networking – Behrouz Forouzan
2. TCP/IP Protocol Suite – Behrouz Forouzan
3. Computer Networks –Andrew Tanenbaum1



BANDWIDTH UTILIZATION: MULTIPLEXING AND SPECTRUM SPREADING

Unit Structure :

- 5.0 Objectives
- 5.1 Introduction
- 5.2 Multiplexing
 - 5.2.1 Types of Multiplexing
 - 5.2.2 Frequency-Division Multiplexing
 - 5.2.3 Wavelength-Division Multiplexing
 - 5.2.4 Time-Division Multiplexing
 - 5.2.4.1 Synchronous Time-Division Multiplexing
 - 5.2.4.2 Statistical Time-Division Multiplexing
- 5.3 Spread Spectrum
 - 5.3.1 Types of Spread Spectrum
 - 5.3.2 Frequency Hopping Spread Spectrum (FHSS)
 - 5.3.3 Direct Sequence Spread Spectrum (DSSS)
- 5.4 Summary
- 5.5 Reference for further reading
- 5.6 Model Questions

5.0 OBJECTIVES:

This chapter would make you to understand the following concepts:

- Concept of Multiplexing.
- Types of Multiplexing: Frequency – division multiplexing, Wavelength – division multiplexing and Time – division multiplexing.
- Concept of Spread Spectrum.
- Types of Spread Spectrum: Frequency Hoping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS).

5.1 INTRODUCTION

In real life, we have links with limited bandwidths. The efficient use of these bandwidths has been, and will be, one of the main challenges of electronic communications. However, the meaning of efficient may depend on the application. Sometimes we need to combine several low-bandwidth channels to make use of one channel with a larger bandwidth. Sometimes we need to expand the bandwidth of a channel to achieve goals such as privacy and anti jamming. In this chapter, we will explore these two broad categories of bandwidth utilization: Multiplexing and Spreading. In multiplexing, our goal is efficiency; we combine several channels into one. In spreading, our goals are privacy and anti-jamming; we expand the bandwidth of a channel to insert redundancy, which is necessary to achieve these goals.

5.2 MULTIPLEXING

Whenever the bandwidth of a medium linking two devices is greater than the bandwidth needs of the devices, the link can be shared. Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link. As data and telecommunications use increases, so does traffic. We can accommodate this increase by continuing to add individual links each time a new channel is needed; or we can install higher-bandwidth links and use each to carry multiple signals. Today's technology includes high-bandwidth media such as optical fiber and terrestrial and satellite microwaves. Each has a bandwidth far in excess of that needed for the average transmission signal. If the bandwidth of a link is greater than the bandwidth needs of the devices connected to it, the bandwidth is wasted. An efficient system maximizes the utilization of all resources; bandwidth is one of the most precious resources we have in data communications.

In a multiplexed system, 'n' lines share the bandwidth of one link. Figure 5.1 shows the basic format of a multiplexed system. The lines on the left direct their transmission streams to a multiplexer (MUX), which combines them into a single stream (many-to-one). At the receiving end, that stream is fed into a de-multiplexer (DEMUX), which separates the stream back into its component transmissions (one-to-many) and directs them to their corresponding lines. In the figure, the word link refers to the physical path. The word channel refers to the portion of a link that carries a transmission between a given pair of lines. One link can have many (n) channels.

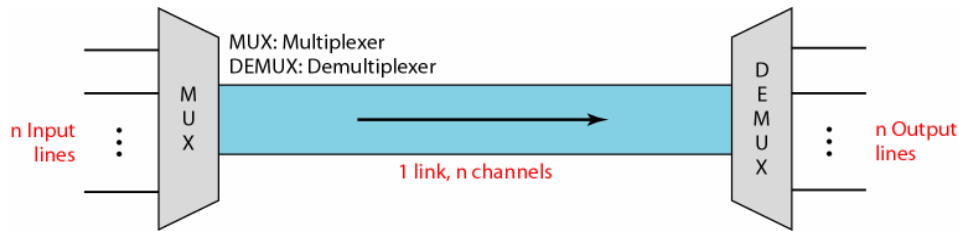


Figure 5.1: Dividing a link into channels

5.2.1 Types of Multiplexing

There are three basic multiplexing techniques: Frequency-division multiplexing, Wavelength-division multiplexing, and Time-division multiplexing. The first two are techniques designed for analog signals, the third, for digital signals (see Figure 5.2).

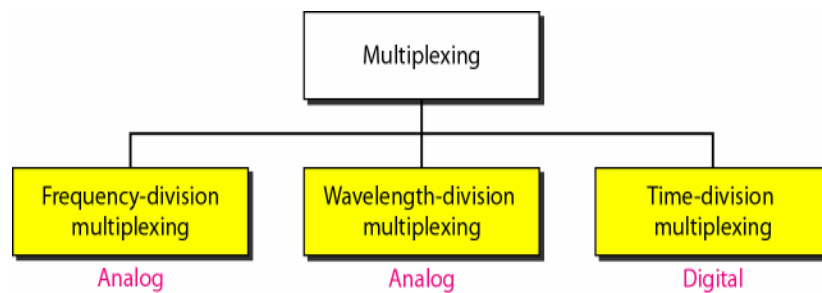


Figure 5.2: Types of Multiplexing

5.2.2 Frequency-Division Multiplexing

Frequency-division multiplexing (FDM) is an analog technique that can be applied when the bandwidth of a link (in hertz) is greater than the combined bandwidths of the signals to be transmitted. In FDM, signals generated by each sending device modulate different carrier frequencies. These modulated signals are then combined into a single composite signal that can be transported by the link. Carrier frequencies are separated by sufficient bandwidth to accommodate the modulated signal. These bandwidth ranges are the channels through which the various signals travel. Channels can be separated by strips of unused bandwidth-guard bands-to prevent signals from overlapping. In addition, carrier frequencies must not interfere with the original data frequencies. Figure 5.3 gives a conceptual view of FDM. In this figure, the transmission path is divided into three parts, each representing a channel that carries one transmission.



Figure 5.3: Frequency Division Multiplexing (FDM)

We consider FDM to be an analog multiplexing technique; however, this does not mean that FDM cannot be used to combine sources sending digital signals. In such case a digital signal can be first converted to an analog signal before FDM is used to multiplex them.

Multiplexing Process

Figure 5.4 is a conceptual illustration of the multiplexing process. Each source generates a signal of a similar frequency range. Inside the multiplexer, these similar signals modulate different carrier frequencies (f_1 , f_2 and f_3). The resulting modulated signals are then combined into a single composite signal that is sent out over a media link that has enough bandwidth to accommodate it.

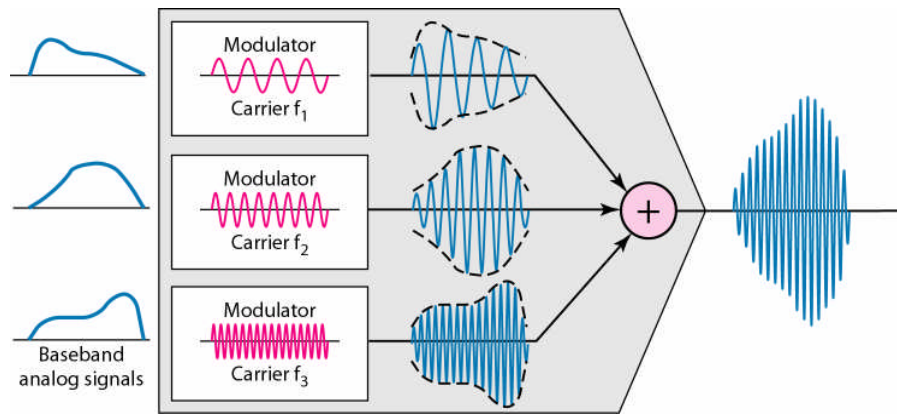


Figure 5.4: FDM – Multiplexing process

De-multiplexing Process

The de-multiplexer uses a series of filters to decompose the multiplexed signal into its constituent component signals. The individual signals are then passed to a demodulator that separates them from their carriers and passes them to the output lines. Figure 5.5 is a conceptual illustration of de-multiplexing process.

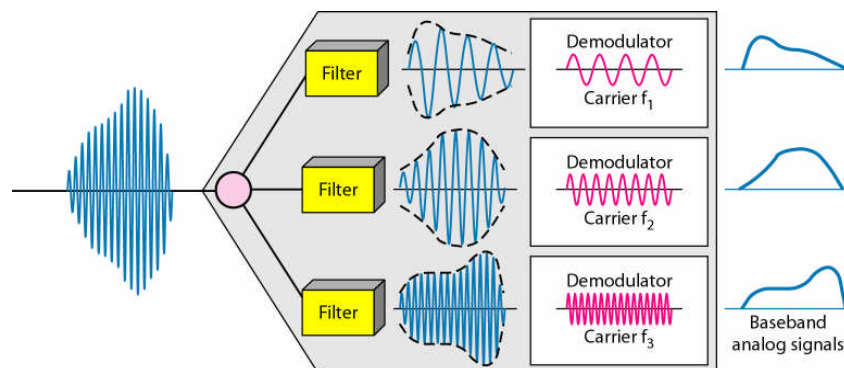


Figure 5.5: FDM – De-multiplexing process

Example 5.1:

Assume that a voice channel occupies a bandwidth of 4 kHz. We need to combine three voice channels into a link with a bandwidth of 12 kHz, from 20 to 32 kHz. Show the configuration, using the frequency domain. Assume there are no guard bands.

Solution:

We shift (modulate) each of the three voice channels to a different bandwidth, as shown in Figure 5.6. We use the 20-kHz to 24-kHz bandwidth for the first channel, the 24-kHz to 28-kHz bandwidth for the second channel, and the 28-kHz to 32-kHz bandwidth for the the third one. Then we combine them as shown in Figure 5.6. At the receiver, each channel receives the entire signal, using a filter to separate out its own signal. The first channel uses a filter that passes frequencies between 20 and 24 kHz and filters out (discards) any other frequencies. The second channel uses a filter that passes frequencies between 24 and 28 kHz, and the third channel uses a filter that passes frequencies between 28 and 32 kHz. Each channel then shifts the frequency to start from zero.

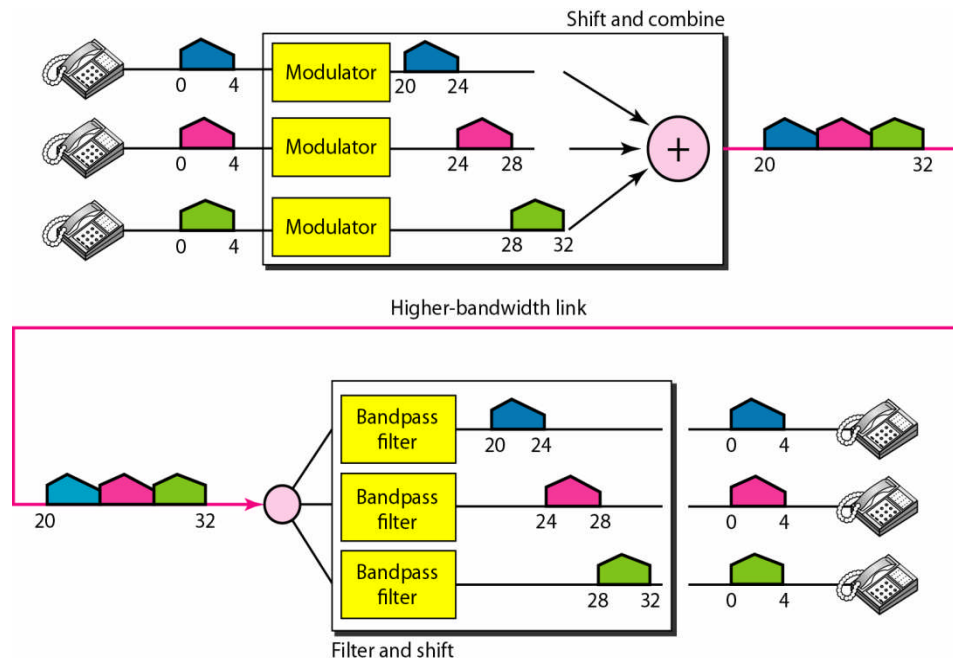


Figure 5.6: Example 5.1

Example 5.2:

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

Solution:

For five channels, we need at least four guard bands. This means that the required bandwidth is at least $5 \times 100 + 4 \times 10 = 540$ kHz.

Applications of FDM

A very common application of FDM is AM and FM radio broadcasting. Radio uses the air as the transmission medium. A special band from 530 to 1700 kHz is assigned to AM radio. All radio stations need to share this band. Each AM station needs 10kHz of bandwidth. Each station uses a different carrier frequency, which means it is shifting its signal and multiplexing. The signal that goes to the air is a combination of signals. A receiver receives all these signals, but filters (by tuning) only the one which is desired. Without multiplexing, only one AM station could broadcast to the common link, the air. However, we need to know that there is physical multiplexer or de-multiplexer here.

The situation is similar in FM broadcasting. However, FM has a wider band of 88 to 108 MHz because each station needs a bandwidth of 200 kHz.

Another common use of FDM is in television broadcasting. Each TV channel has its own bandwidth of 6 MHz.

The first generation of cellular telephones (still in operation) also uses FDM. Each user is assigned two 30-kHz channels, one for sending voice and the other for receiving. The voice signal, which has a bandwidth of 3 kHz (from 300 to 3300 Hz), is modulated by using FM. Remember that an FM signal has a bandwidth 10 times that of the modulating signal, which means each channel has 30 kHz (10×3) of bandwidth. Therefore, each user is given, by the base station, a 60-kHz bandwidth in a range available at the time of the call.

FDM Implementation

FDM can be implemented very easily. In many cases, such as radio and television broadcasting, there is no need for a physical multiplexer or de-multiplexer. As long as the stations agree to send their broadcasts to the air using different carrier frequencies, multiplexing is achieved. In other cases, such as the cellular telephone system, a base station needs to assign a carrier frequency to the telephone user. There is not enough bandwidth in a cell to permanently assign a bandwidth range to every telephone user. When a user hangs up, her or his bandwidth is assigned to another caller.

5.2.3 Wavelength-Division Multiplexing

Wavelength-division multiplexing (WDM) is designed to use the high-data-rate capability of fiber-optic cable. The optical fiber data rate is higher than the data rate of metallic transmission cable. Using a fiber-optic cable for one single line wastes the available bandwidth. Multiplexing allows us to combine several lines into one.

WDM is conceptually the same as FDM, except that the multiplexing and de-multiplexing involve optical signals transmitted through fiber-optic channels. The idea is the same: We are combining different signals of different frequencies. The difference is that the frequencies are very high.

Figure 5.7 gives a conceptual view of a WDM multiplexer and de-multiplexer. Very narrow bands of light from different sources are combined to make a wider band of light. At the receiver, the signals are separated by the de-multiplexer.

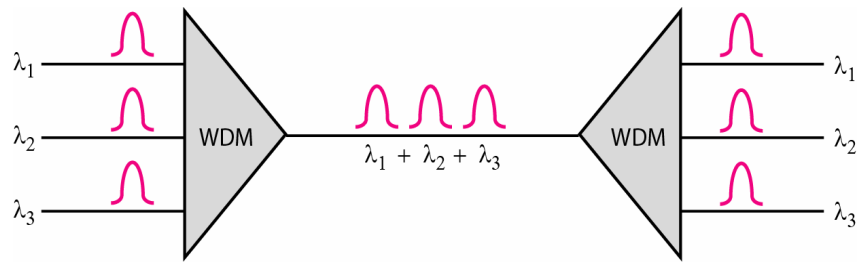


Figure 5.7: Wavelength Division Multiplexing

Although WDM technology is very complex, the basic idea is very simple. We want to combine multiple light sources into one single light at the multiplexer and do the reverse at the de-multiplexer. The combining and splitting of light sources are easily handled by a prism. Recall from basic physics that a prism bends a beam of light based on the angle of incidence and the frequency. Using this technique, a multiplexer can be made to combine several input beams of light, each containing a narrow band of frequencies, into one output beam of a wider band of frequencies. A de-multiplexer can also be made to reverse the process. Figure 5.8 shows the concept.

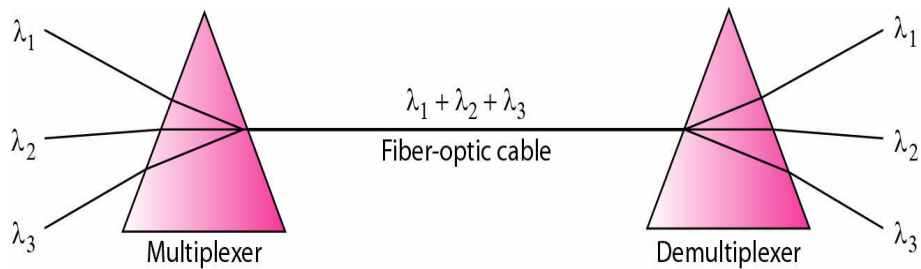


Figure 5.8: Prisms in Wavelength Division Multiplexing and De-multiplexing

Applications of WDM

One application of WDM is the SONET (Synchronous Optical Network) in which multiple optical fiber lines are multiplexed and de-multiplexed. A new method, called Dense WDM (DWDM), can multiplex a very large number of channels by spacing channels very close to one another. It achieves even greater efficiency.

5.2.4 Time-Division Multiplexing

Time-division multiplexing (TDM) is a digital process that allows several connections to share the high bandwidth of a link. Instead of sharing a portion of the bandwidth as in FDM, time is shared. Each connection occupies a portion of time in the link. Figure 5.9 gives a

conceptual view of TDM. Note that the same link is used as in FDM; here, however, the link is shown sectioned by time rather than by frequency. In the figure, portions of signals 1,2,3 and 4 occupy the link sequentially.

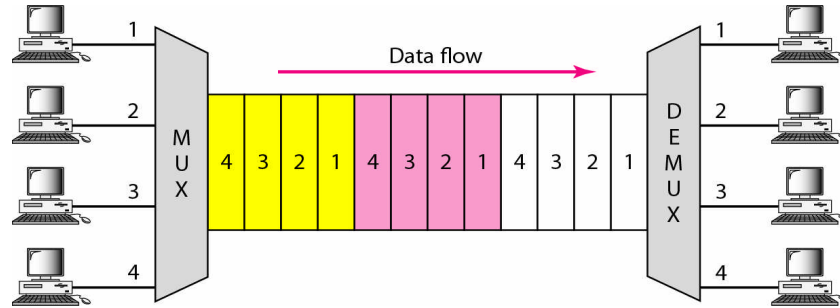


Figure 5.9: Time Division Multiplexing

Note that in Figure 5.9 we are concerned with only multiplexing, not switching. This means that all the data in a message from source 1 always go to one specific destination, be it 1, 2, 3, or 4. The delivery is fixed and unvarying, unlike switching.

We also need to remember that TDM is, in principle, a digital multiplexing technique. Digital data from different sources are combined into one timeshared link. However, this does not mean that the sources cannot produce analog data; analog data can be sampled, changed to digital data, and then multiplexed by using TDM.

We can divide TDM into two different schemes: synchronous and statistical. We first discuss synchronous TDM and then show how statistical TDM differs.

5.2.4.1 Synchronous Time-Division Multiplexing

In synchronous TDM, each input connection has an allotment in the output even if it is not sending data.

Time Slots and Frames

In synchronous TDM, the data flow of each input connection is divided into units, where each input occupies one input time slot. A unit can be 1 bit, one character, or one block of data. Each input unit becomes one output unit and occupies one output time slot. However, the duration of an output time slot is n times shorter than the duration of an input time slot. If an input time slot is T_s , the output time slot is T_{ins} , where n is the number of connections. In other words, a unit in the output connection has a shorter duration; it travels faster. Figure 5.10 shows an example of synchronous TDM where n is 3.

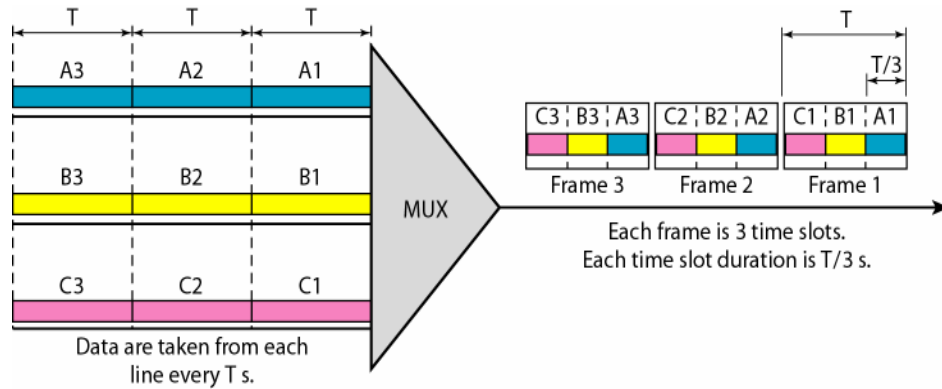


Figure 5.10: Synchronous TDM

In synchronous TDM, a round of data units from each input connection is collected into a frame. If we have n connections, a frame is divided into n time slots and one slot is allocated for each unit, one for each input line. If the duration of the input unit is T , the duration of each slot is T/n and the duration of each frame is T .

The data rate of the output link must be n times the data rate of a connection to guarantee the flow of data. In Figure 5.10, the data rate of the link is 3 times the data rate of a connection; likewise, the duration of a unit on a connection is 3 times that of the time slot (duration of a unit on the link). In the figure we represent the data prior to multiplexing as 3 times the size of the data after multiplexing. This is just to convey the idea that each unit is 3 times longer in duration before multiplexing than after.

Time slots are grouped into frames. A frame consists of one complete cycle of time slots, with one slot dedicated to each sending device. In a system with n input lines, each frame has n slots, with each slot allocated to carrying data from a specific input line.

Example 5.3:

In Figure 5.10, the data rate for each input connection is 3 kbps. If 1 bit at a time is multiplexed (a unit is 1 bit), what is the duration of (a) each input slot, (b) each output slot, and (c) each frame?

Solution:

We can answer the questions as follows:

- a) The data rate of each input connection is 1 kbps. This means that the bit duration is $1/1000$ s or **1 ms**. The duration of the input time slot is **1 ms** (same as bit duration).
- b) The duration of each output time slot is one-third of the input time slot. This means that the duration of the output time slot is **$1/3$ ms**.
- c) Each frame carries three output time slots. So the duration of a frame is **$3 \times 1/3$ ms**, or **1 ms**. The duration of a frame is the same as the duration of an input unit.

Example 5.4:

Figure 5.11 shows synchronous TDM with a data stream for each input and one data stream for the output. The unit of data is 1 bit. Find (a) the input bit duration, (b) the output bit duration, (c) the output bit rate, and (d) the output frame rate.

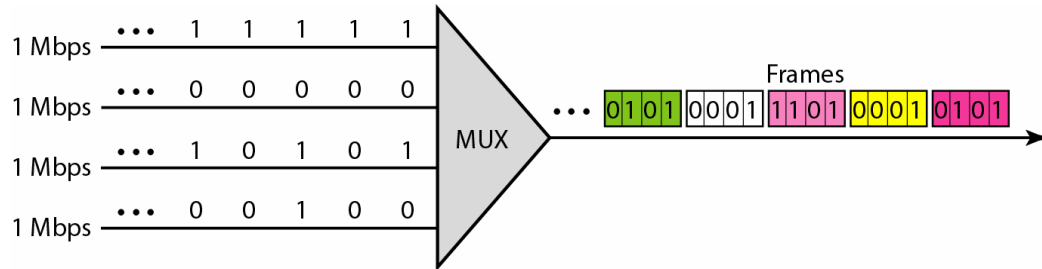


Figure 5.11: Example 5.4

Solution:

We can answer the questions as follows:

- The input bit duration is the inverse of the bit rate: $1/1 \text{ Mbps} = 1 \text{ } \mu\text{s}$.
- The output bit duration is one-fourth of the input bit duration, or $1/4 \mu\text{s}$.
- The output bit rate is the inverse of the output bit duration or $1/(4 \text{ } \mu\text{s})$, or 4 Mbps . This can also be deduced from the fact that the output rate is 4 times as fast as any input rate; so the output rate $= 4 \times 1 \text{ Mbps} = 4 \text{ Mbps}$.
- The frame rate is always the same as any input rate. So the frame rate is $1,000,000$ frames per second. Because we are sending **4 bits** in each frame, we can verify the result of the previous question by multiplying the frame rate by the number of bits per frame.

Interleaving

TDM can be visualized as two fast-rotating switches, one on the multiplexing side and the other on the de-multiplexing side. The switches are synchronized and rotate at the same speed, but in opposite directions. On the multiplexing side, as the switch opens in front of a connection, that connection has the opportunity to send a unit onto the path. This process is called **interleaving**. On the de-multiplexing side, as the switch opens in front of a connection, that connection has the opportunity to receive a unit from the path.

Figure 5.12 shows the interleaving process for the connection shown in Figure 5.10. In this figure, we assume that no switching is involved and that the data from the first connection at the multiplexer site go to the first connection at the de-multiplexer.

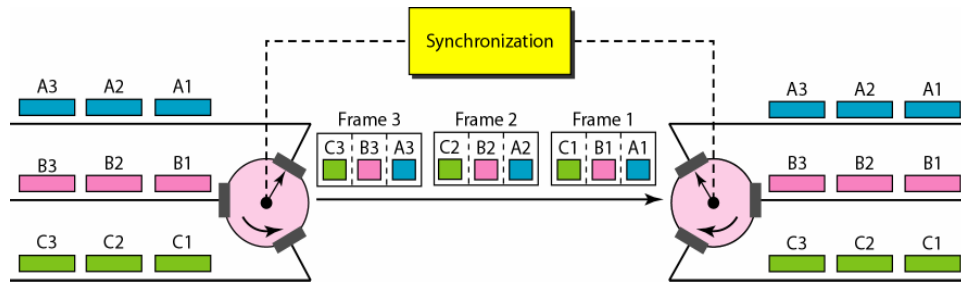


Figure 5.12: Interleaving process

Example 5.5:

Four channels are multiplexed using TDM. If each channel sends **100 bytes/s** and we multiplex

1 byte per channel, show the frame traveling on the link, the size of the frame, the duration of a frame, the frame rate, and the bit rate for the link.

Solution:

The multiplexer is shown in Figure 5.13. Each frame carries **1 byte** from each channel; the size of each frame, therefore, is **4 bytes**, or **32 bits**. Because each channel is sending **100 bytes/s** and a frame carries **1 byte** from each channel, the frame rate must be **100 frames** per second. The duration of a frame is therefore **1/100 s**. The link is carrying **100 frames** per second, and since each frame contains **32 bits**, the bit rate is **100 x 32**, or **3200 bps**. This is actually **4 times** the bit rate of each channel, which is **100 x 8 = 800 bps**.

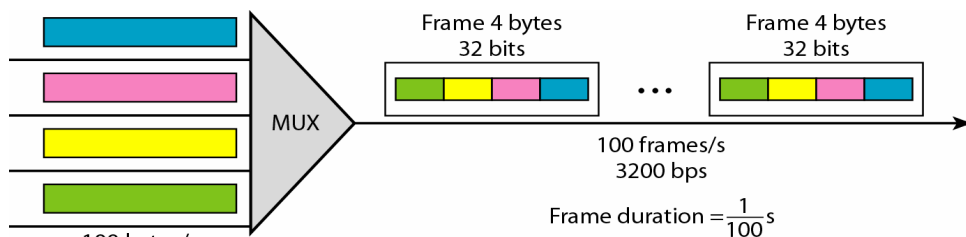


Figure 5.13: Example 5.5

Empty Slots

Synchronous TDM is not as efficient as it could be. If a source does not have data to send, the corresponding slot in the output frame is empty. Figure 5.14 shows a case in which one of the input lines has no data to send and one slot in another input line has discontinuous data.

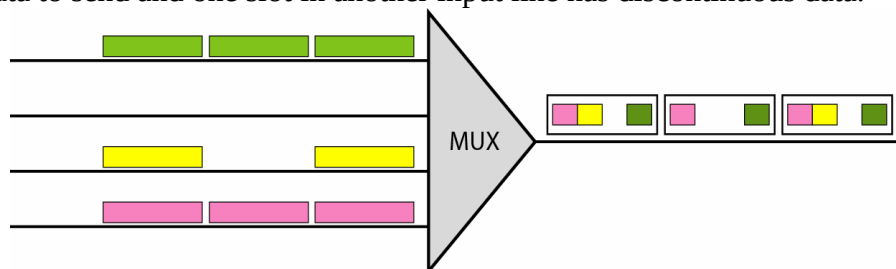


Figure 5.14: Empty Slots

The first output frame has three slots filled, the second frame has two slots filled, and the third frame has three slots filled. No frame is full. We learn in the next section that statistical TDM can improve the efficiency by removing the empty slots from the frame.

Data Rate Management

One problem with TDM is how to handle a disparity in the input data rates. In all our discussion so far, we assumed that the data rates of all input lines were the same. However, if data rates are not the same, three strategies, or a combination of them, can be used. We call these three strategies **Multilevel multiplexing**, **Multiple-slot allocation**, and **Pulse stuffing**.

Multilevel Multiplexing

Multilevel multiplexing is a technique used when the data rate of an input line is a multiple of others. For example, in Figure 5.15, we have two inputs of **20 kbps** and **three** inputs of **40 kbps**. The first two input lines can be multiplexed together to provide a data rate equal to the last three. A second level of multiplexing can create an output of **160 kbps**.

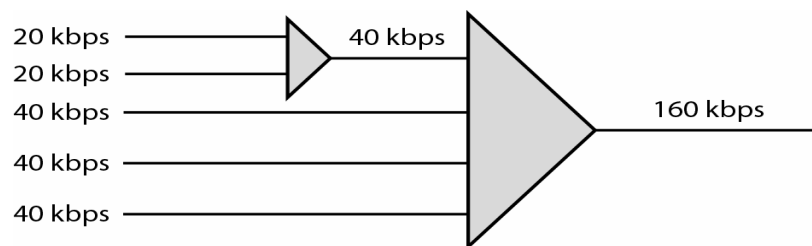
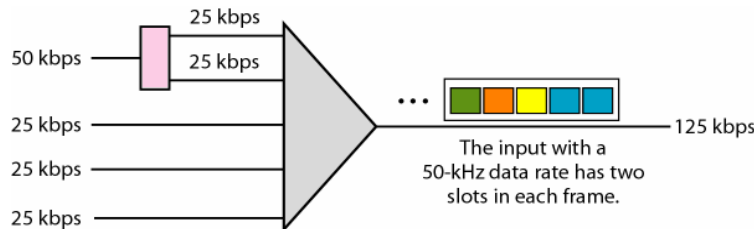


Figure 5.15: Multilevel Multiplexing

Multiple-Slot Allocation

Sometimes it is more efficient to allot more than one slot in a frame to a single input line. For example, we might have an input line that has a data rate that is a multiple of another input. In Figure 5.16, the input line with a 50-kbps data rate can be given two slots in the output. We insert a serial-to-parallel converter in the line to make two inputs out of one.



5.16: Multiple-slot Multiplexing

Pulse Stuffing

Sometimes the bit rates of sources are not multiple integers of each other. Therefore, neither of the above two techniques can be applied. One solution is to make the highest input data rate the dominant data rate and then add dummy bits to the input lines with lower rates. This will increase their rates. This technique is called **Pulse stuffing**, bit padding, or bit

stuffing. The idea is shown in Figure 5.17. The input with a data rate of **46** is pulse-stuffed to increase the rate to **50 kbps**. Now multiplexing can take place.

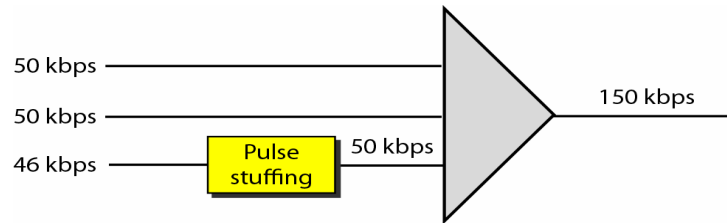


Figure 5.17: Pulse Stuffing

Frame Synchronizing

The implementation of TDM is not as simple as that of FDM. Synchronization between the multiplexer and de-multiplexer is a major issue. If the multiplexer and the de-multiplexer are not synchronized, a bit belonging to one channel may be received by the wrong channel. For this reason, one or more synchronization bits are usually added to the beginning of each frame. These bits, called **framing bits**, follow a pattern, frame to frame, that allows the de-multiplexer to synchronize with the incoming stream so that it can separate the time slots accurately. In most cases, this synchronization information consists of 1 bit per frame, alternating between 0 and 1, as shown in Figure 5.18.

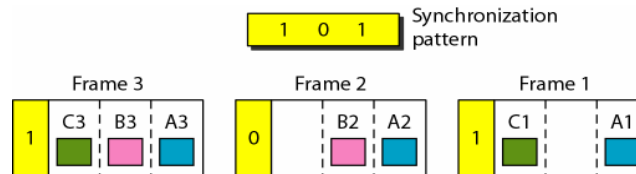


Figure 5.18: Framing bits

Example 5.6:

We have four sources, each creating 250 characters per second. If the interleaved unit is a character and 1 synchronizing bit is added to each frame, find (a) the data rate of each source, (b) the duration of each character in each source, (c) the frame rate, (d) the duration of each frame, (e) the number of bits in each frame, and (f) the data rate of the link.

Solution:

We can answer the questions as follows:

- The data rate of each source is $250 \times 8 = 2000 \text{ bps} = 2 \text{ kbps}$.
- Each source sends 250 characters per second; therefore, the duration of a character is $1/250 \text{ s}$, or **4 ms**.

- c) Each frame has one character from each source, which means the link needs to send **250** frames per second to keep the transmission rate of each source.
- d) The duration of each frame is **1/250s**, or **4 ms**. Note that the duration of each frame is the same as the duration of each character coming from each source.
- e) Each frame carries **4** characters and **1** extra synchronizing bit. This means that each frame is **4 x 8 + 1 = 33 bits**.
- f) The link sends **250** frames per second, and each frame contains **33 bits**. This means that the data rate of the link is **250 x 33**, or **8250 bps**. Note that the bit rate of the link is greater than the combined bit rates of the four channels. If we add the bit rates of four channels, we get **8000 bps**. Because **250** frames are traveling per second and each contains **1** extra bit for synchronizing, we need to add **250** to the sum to get **8250 bps**.

TDM Implementation

Telephone companies implement TDM through a hierarchy of **digital signals**, called **digital signal (DS) service** or **digital hierarchy**. Figure 5.19 shows the data rates supported by each level.

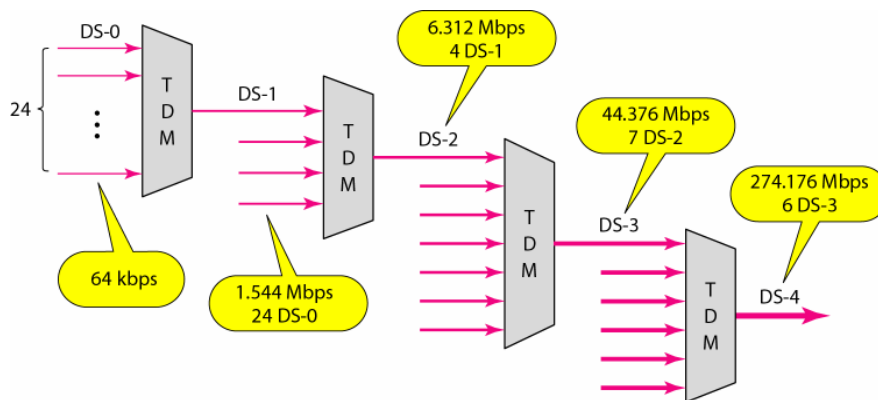


Figure 5.19: Digital Hierarchy

- A **DS-0** service is a single digital channel of 64 kbps.
- **DS-1** is a 1.544-Mbps service; 1.544 Mbps is 24 times 64 kbps plus 8 kbps of overhead. It can be used as a single service for 1.544-Mbps transmissions, or it can be used to multiplex 24 DS-0 channels or to carry any other combination desired by the user that can fit within its 1.544-Mbps capacity.
- **DS-2** is a 6.312-Mbps service; 6.312 Mbps is 96 times 64 kbps plus 168 kbps of overhead. It can be used as a single service for 6.312-Mbps transmissions; or it can be used to multiplex 4 DS-1 channels, 96 DS-0 channels, or a combination of these service types.
- **DS-3** is a 44.376-Mbps service; 44.376 Mbps is 672 times 64 kbps plus 1.368 Mbps of overhead. It can be used as a single service for

44.376-Mbps transmissions; or it can be used to multiplex 7 DS-2 channels, 28 DS-1 channels, 672 DS-0 channels, or a combination of these service types.

- **DS-4** is a 274.176-Mbps service; 274.176 is 4032 times 64 kbps plus 16.128 Mbps of overhead. It can be used to multiplex 6 DS-3 channels, 42 DS-2 channels, 168 DS-1 channels, 4032 DS-0 channels, or a combination of these service types.

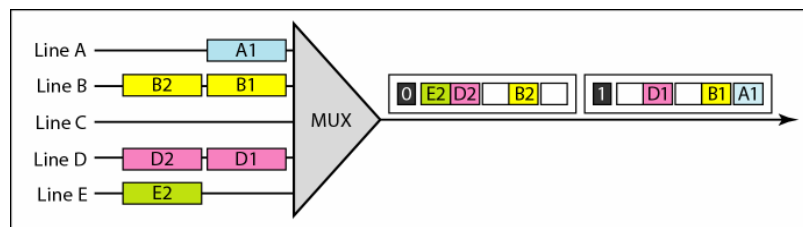
Applications of Synchronous TDM

Some second-generation cellular telephone companies use synchronous TDM. For example, the digital version of cellular telephony divides the available bandwidth into 3D-kHz bands. For each band, TDM is applied so that six users can share the band. This means that each 3D-kHz band is now made of six time slots, and the digitized voice signals of the users are inserted in the slots. Using TDM, the number of telephone users in each area is now 6 times greater.

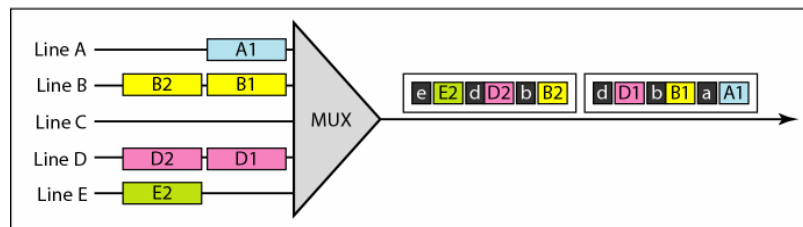
5.2.4.2 Statistical Time-Division Multiplexing

As we saw in synchronous TDM, each input has a reserved slot in the output frame. This can be inefficient if some input lines have no data to send. In statistical time-division multiplexing, slots are dynamically allocated to improve bandwidth efficiency. Only when an input line has a slot's worth of data to send is it given a slot in the output frame. In statistical multiplexing, the number of slots in each frame is less than the number of input lines. The multiplexer checks each input line in round robin fashion; it allocates a slot for an input line if the line has data to send; otherwise, it skips the line and checks the next line.

Figure 5.20 shows a synchronous and a statistical TDM example. In the former, some slots are empty because the corresponding line does not have data to send. In the latter, however, no slot is left empty as long as there are data to be sent by any input line.



a. Synchronous TDM



b. Statistical TDM

Figure 5.20 TDM slot comparison

Addressing

Figure 5.20 also shows a major difference between slots in synchronous TDM and statistical TDM. An output slot in synchronous TDM is totally occupied by data; in statistical TDM, a slot needs to carry data as well as the address of the destination.

In synchronous TDM, there is no need for addressing; synchronization and pre assigned relationships between the inputs and outputs serve as an address. We know, for example, that input 1 always goes to input 2. If the multiplexer and the de-multiplexer are synchronized, this is guaranteed. In statistical multiplexing, there is no fixed relationship between the inputs and outputs because there are no pre assigned or reserved slots. We need to include the address of the receiver inside each slot to show where it is to be delivered. The addressing in its simplest form can be n bits to define N different output lines with $n = \log_2 N$. For example, for eight different output lines, we need a 3-bit address.

Slot Size

Since a slot carries both data and an address in statistical TDM, the ratio of the data size to address size must be reasonable to make transmission efficient. For example, it would be inefficient to send 1 bit per slot as data when the address is 3 bits. This would mean an overhead of 300 percent. In statistical TDM, a block of data is usually many bytes while the address is just a few bytes.

No Synchronization Bit

There is another difference between synchronous and statistical TDM, but this time it is at the frame level. The frames in statistical TDM need not be synchronized, so we do not need synchronization bits.

Bandwidth

In statistical TDM, the capacity of the link is normally less than the sum of the capacities of each channel. The designers of statistical TDM define the capacity of the link based on the statistics of the load for each channel. If on average only x percent of the input slots are filled, the capacity of the link reflects this. Of course, during peak times, some slots need to wait.

5.3 SPREAD SPECTRUM

Multiplexing combines signals from several sources to achieve bandwidth efficiency; the available bandwidth of a link is divided between the sources. In Spread Spectrum, we also combine signals from different sources to fit into a larger bandwidth, but our goals are somewhat different. Spread spectrum is designed to be used in wireless applications (LANs and WANs). In these types of applications, we have some concerns that outweigh bandwidth efficiency. In wireless applications, all stations

use air (or a vacuum) as the medium for communication. Stations must be able to share this medium without interception by an eavesdropper and without being subject to jamming from a malicious intruder.

To achieve these goals, spread spectrum techniques add redundancy; they spread the original spectrum needed for each station. If the required bandwidth for each station is B , spread spectrum expands it to B_{SS} such that $B_{SS} > B$. The expanded bandwidth allows the source to wrap its message in a protective envelope for a more secure transmission.

An analogy is the sending of a delicate, expensive gift. We can insert the gift in a special box to prevent it from being damaged during transportation, and we can use a superior delivery service to guarantee the safety of the package. Figure 5.21 shows the idea of spread spectrum. Spread spectrum achieves its goals through two principles:

1. The bandwidth allocated to each station needs to be, by far, larger than what is needed. This allows redundancy.
2. The expanding of the original bandwidth B to the bandwidth B_{SS} must be done by a process that is independent of the original signal. In other words, the spreading process occurs after the signal is created by the source.

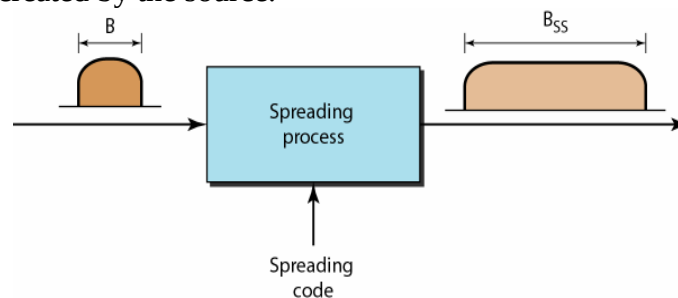


Figure 5.21: Spread Spectrum

After the signal is created by the source, the spreading process uses a spreading code and spreads the bandwidth. The figure shows the original bandwidth B and the spreaded bandwidth B_{SS} . The spreading code is a series of numbers that look random, but are actually a pattern.

5.3.1 Types of Spread Spectrum

There are two techniques to spread the bandwidth: frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS).

5.3.2 Frequency Hopping Spread Spectrum (FHSS)

The frequency hopping spread spectrum (FHSS) technique uses M different carrier frequencies that are modulated by the source signal. At one moment, the signal modulates one carrier frequency; at the next moment, the signal modulates another carrier frequency. Although the modulation is done using one carrier frequency at a time, M frequencies

are used in the long run. The bandwidth occupied by a source after spreading is $B_{pHSS} > B$.

Figure 5.22 shows the general layout for FHSS. A pseudorandom code generator, called Pseudorandom Noise (PN), creates a **k-bit** pattern for every hopping period T_h . The frequency table uses the pattern to find the frequency to be used for this hopping period and passes it to the frequency synthesizer. The frequency synthesizer creates a carrier signal of that frequency, and the source signal modulates the carrier signal.

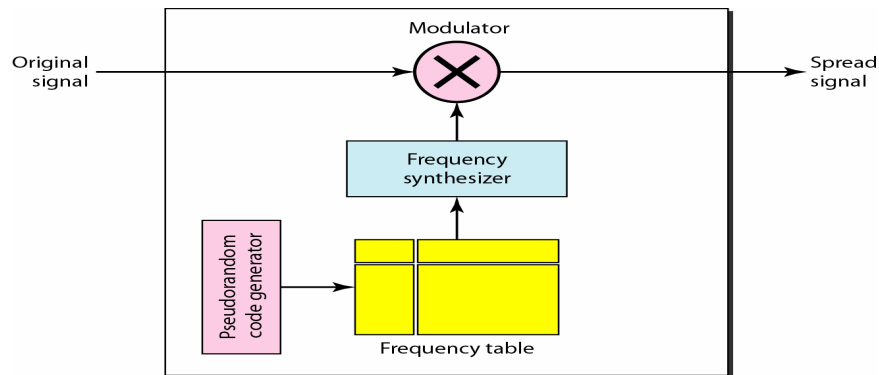


Figure 5.22: Frequency Hopping Spread Spectrum (FHSS)

Suppose we have decided to have eight hopping frequencies. This is extremely low for real applications and is just for illustration. In this case, M is 8 and k is 3. The pseudorandom code generator will create eight different 3-bit patterns. These are mapped to eight different frequencies in the frequency table (see Figure 5.23).

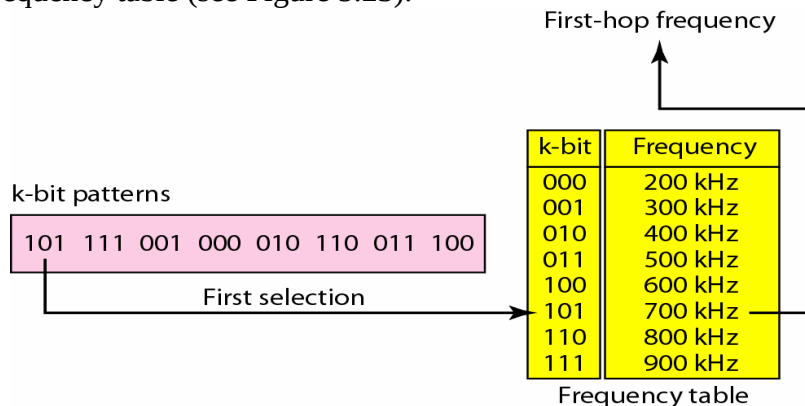


Figure 5.23: Frequency selection in FHSS

The pattern for this station is 101, 111, 001, 000, 010, all, 100. Note that the pattern is pseudorandom it is repeated after eight hoppings. This means that at hopping period 1, the pattern is 101. The frequency selected is 700 kHz; the source signal modulates this carrier frequency. The second **k-bit** pattern selected is 111, which selects the 900-kHz carrier; the eighth pattern is 100, the frequency is 600 kHz. After eight hoppings, the pattern repeats, starting from 101 again. Figure 5.24 shows

how the signal hops around from carrier to carrier. We assume the required bandwidth of the original signal is 100 kHz.

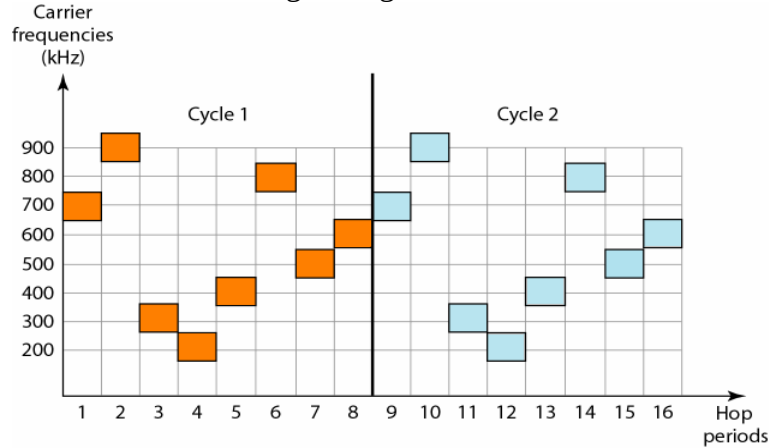


Figure 5.24: FHSS Cycles

It can be shown that this scheme can accomplish the previously mentioned goals. If there are many k-bit patterns and the hopping period is short, a sender and receiver can have privacy. If an intruder tries to intercept the transmitted signal, she can only access a small piece of data because she does not know the spreading sequence to quickly adapt herself to the next hop. The scheme has also an anti jamming effect. A malicious sender may be able to send noise to jam the signal for one hopping period (randomly), but not for the whole period.

Bandwidth Sharing

If the number of hopping frequencies is M , we can multiplex M channels into one by using the same B_{SS} bandwidth. This is possible because a station uses just one frequency in each hopping period; $M - 1$ other frequencies can be used by other $M - 1$ stations. In other words, M different stations can use the same B_{SS} if an appropriate modulation technique such as multiple FSK (MFSK) is used. FHSS is similar to FDM, as shown in Figure 5.25.

Figure 5.25 shows an example of four channels using FDM and four channels using FHSS. In FDM, each station uses $11M$ of the bandwidth, but the allocation is fixed; in FHSS, each station uses $11M$ of the bandwidth, but the allocation changes hop to hop.

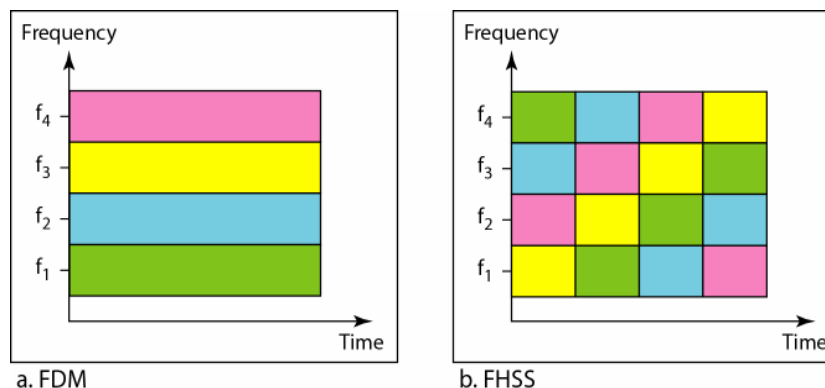


Figure 5.25: Bandwidth sharing

5.3.3 Direct Sequence Spread Spectrum (DSSS)

The direct sequence spread spectrum (DSSS) technique also expands the bandwidth of the original signal, but the process is different. In DSSS, we replace each data bit with 11 bits using a spreading code. In other words, each bit is assigned a code of 11 bits, called chips, where the chip rate is 11 times that of the data bit. Figure 5.26 shows the concept of DSSS.

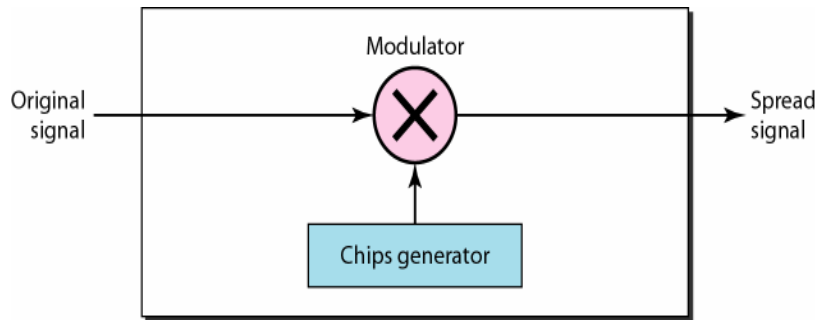


Figure 5.26: DSSS

As an example, let us consider the sequence used in a wireless LAN, the famous Barker sequence where n is 11. We assume that the original signal and the chips in the chip generator use polar NRZ encoding. Figure 5.27 shows the chips and the result of multiplying the original data by the chips to get the spread signal.

In Figure 5.27, the spreading code is 11 chips having the pattern 10110111000 (in this case). If the original signal rate is N , the rate of the spread signal is $11N$. This means that the required bandwidth for the spread signal is 11 times larger than the bandwidth of the original signal. The spread signal can provide privacy if the intruder does not know the code. It can also provide immunity against interference if each station uses a different code.

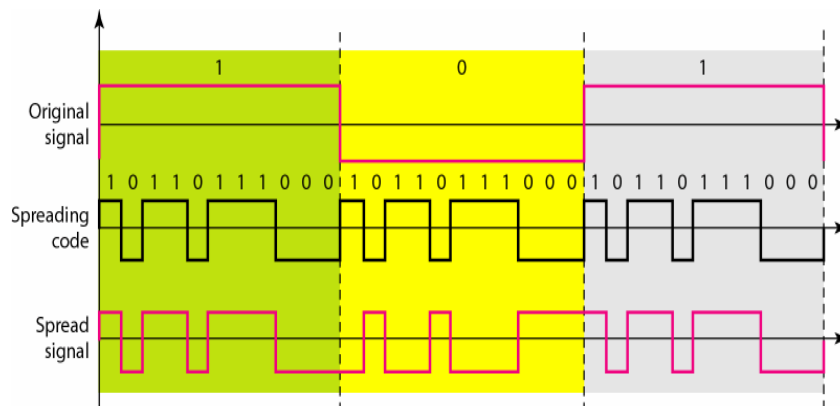


Figure 5.27: DSSS Example

Bandwidth Sharing

Can we share a bandwidth in DSSS as we did in FHSS? The answer is no and yes. If we use a spreading code that spreads signals (from different stations) that cannot be combined and separated, we cannot share a bandwidth. For example, some wireless LANs use DSSS and the spread bandwidth cannot be shared. However, if we use a special type of sequence code that allows the combining and separating of spread signals, we can share the bandwidth. A special spreading code allows us to use DSSS in cellular telephony and share a bandwidth between several users.

5.4 SUMMARY

Bandwidth utilization is the use of available bandwidth to achieve specific goals. Efficiency can be achieved by using multiplexing; privacy and ant jamming can be achieved by using spreading.

- Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link. In a multiplexed system, n lines share the bandwidth of one link. The word link refers to the physical path. The word channel refers to the portion of a link that carries a transmission. There are three basic multiplexing techniques: frequency-division multiplexing, wavelength-division multiplexing, and time-division multiplexing. The first two are techniques designed for analog signals, the third, for digital signals.
- Frequency-division multiplexing (FDM) is an analog technique that can be applied when the bandwidth of a link (in hertz) is greater than the combined bandwidths of the signals to be transmitted.
- Wavelength-division multiplexing (WDM) is designed to use the high bandwidth capability of fiber-optic cable. WDM is an analog multiplexing technique to combine optical signals.
- Time-division multiplexing (TDM) is a digital process that allows several connections to share the high bandwidth of a link. TDM is a digital multiplexing technique for combining several low-rate channels into one high-rate one.
- We can divide TDM into two different schemes: synchronous or statistical. In synchronous, each input connection has an allotment in the output even if it is not sending data. In statistical TDM, slots are dynamically allocated to improve bandwidth efficiency.
- In Spread Spectrum (SS), we combine signals from different sources to fit into a larger bandwidth. Spread spectrum is designed to be used in wireless applications in which stations must be able to share the medium without interception by an eavesdropper and without being subject to jamming from a malicious intruder.
- The Frequency Hopping Spread Spectrum (FHSS) technique uses M different carrier frequencies that are modulated by the source signal. At one moment, the signal modulates one carrier frequency; at the next moment, the signal modulates another carrier frequency.

- The Direct Sequence Spread Spectrum (DSSS) technique expands the bandwidth of a signal by replacing each data bit with n bits using a spreading code. In other words, each bit is assigned a code of n bits, called chips.

5.5 REFERENCE FOR FURTHER READING

For more details about topics discussed in this chapter, we recommend the following books.

1. *Data Communication and Networking* by Behrouz A. Forouzan, McGraw-Hill, 2007.
2. *Basic Communication Theory* by J. E. Pearson, Prentice Hall, 1992.
3. *Digital and Analog Communication Systems* by L.W. Couch, Prentice Hall, 2001.
4. *Digital Baseband and Transmission and Recording* by J. Bergman, Kluwer Academic, 1996.
5. *Data and Computer Communications* by W. Stallings, Prentice Hall, 2004.

5.6 MODEL QUESTIONS

1. Describe the purpose of multiplexing.
2. What are the three main multiplexing techniques?
3. What is difference between a link and a channel in multiplexing?
4. Which of the three multiplexing techniques is common for fiber optic links? Explain the reason.
5. Differentiate between multilevel TDM, multiple slot TDM, and pulse-stuffed TDM.
6. Differentiate between synchronous and statistical TDM.
7. Define spread spectrum and its goal. List the two spread spectrum.
8. Define FHSS and explain how it achieves bandwidth spreading.
9. Define DSSS and explain how it achieves bandwidth spreading.



TRANSMISSION MEDIA AND SWITCHING

Unit Structure

- 6.0 Objectives
- 6.1 Introduction
- 6.2 Guided Media – Wired
 - 6.2.1 Twisted – Pair Cable
 - 6.2.2 Coaxial Cable
 - 6.2.3 Fiber – Optic Cable
- 6.3 Unguided Media – Wireless
 - 6.3.1 Radio Waves
 - 6.3.2 Microwaves
 - 6.3.3 Infrared
- 6.4 Switching
 - 6.4.1 Circuit – Switching (Circuit Switched Networks)
 - 6.4.2 Packet – Switching (Packet Switched Networks)
 - 6.4.2.1 Datagram switching
 - 6.4.2.2 Virtual – Circuit Switching
 - 6.4.3 Message – Switching (Message Switched Networks)
- 6.5 Structure of Switch
 - 6.5.1 Structure of Circuit Switch
 - 6.5.2 Structure of Packet Switch
- 6.6 Summary
- 6.7 Reference for further reading
- 6.8 Model Questions

6.0 OBJECTIVES:

This chapter would make you to understand the following concepts:

- What is Transmission Medium?
- Types of Transmission media – Guided and Unguided.
- Types of Guided transmission media.
- Types of Unguided transmission media.
- Concept of Switching.
- Types of switching – Circuit switching, Packet switching and Message switching.
- Structure of a Switch.

6.1 INTRODUCTION

In this chapter, we are going to discuss transmission media. Transmission media are actually located below the physical layer and are directly controlled by the physical layer. You could say that transmission media belong to layer zero. Figure 6.1 shows the position of transmission media in relation to the physical layer.

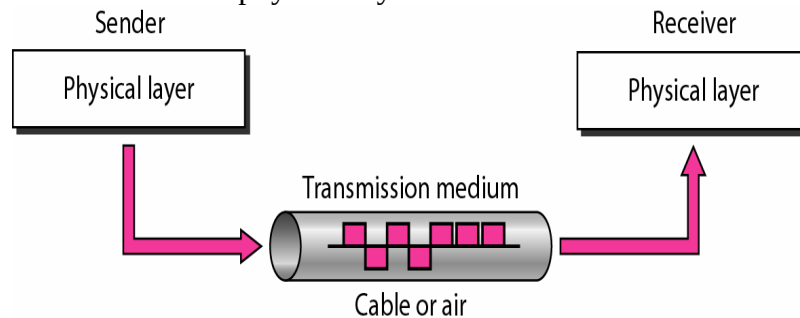


Figure 6.1: Transmission Medium and Physical layer

A transmission **medium** can be broadly defined as anything (either wire or air) that can carry information from a source to a destination. For example, the transmission medium for two people having a dinner conversation is the air. For a written message, the transmission medium might be a mail carrier, a truck, or an airplane. In data communications the definition of the information and the transmission medium is more specific. The transmission medium is usually free space, metallic cable, or fiber-optic cable. The information is usually a signal that is the result of a conversion of data from another form.

In telecommunications, transmission media can be divided into two broad categories: guided and unguided. Guided media include twisted-pair cable, coaxial cable, and fiber-optic cable. Unguided medium is free space. Figure 6.2 shows this classification.

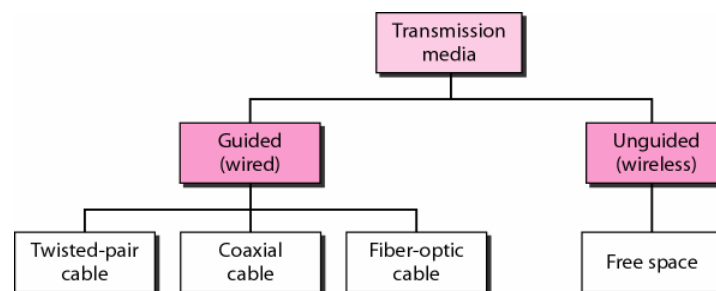


Figure 6.2: Classification of Transmission media

6.2 GUIDED MEDIA – WIRED

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

cable. A signal traveling along any of these media is directed and contained by the physical limits of the medium. Twisted-pair and coaxial cable use metallic (copper) conductors that accept and transport signals in the form of electric current. Optical fiber is a cable that accepts and transports signals in the form of light.

6.2.1 Twisted – Pair Cable

A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together, as shown in Figure 6.3.

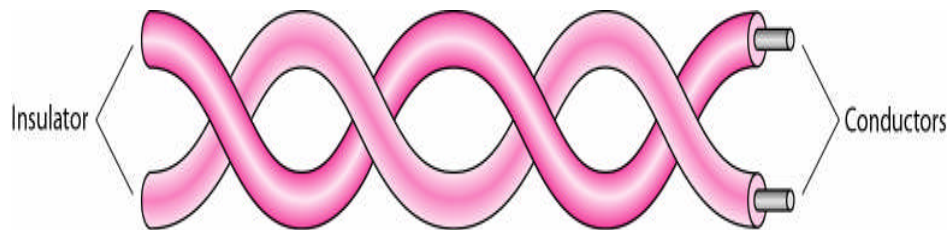


Figure 6.3: Twisted – pair cable

One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference. The receiver uses the difference between the two. In addition to the signal sent by the sender on one of the wires, interference (noise) and crosstalk may affect both wires and create unwanted signals. If the two wires are parallel, the effect of these unwanted signals is not the same in both wires because they are at different locations relative to the noise or crosstalk sources (e.g., one is closer and the other is farther). This results in a difference at the receiver. By twisting the pairs, a balance is maintained. For example, suppose in one twist, one wire is closer to the noise source and the other is farther; in the next twist, the reverse is true. Twisting makes it probable that both wires are equally affected by external influences (noise or crosstalk). This means that the receiver, which calculates the difference between the two, receives no unwanted signals. The unwanted signals are mostly canceled out. Following Figure 6.4 shows how twisting of wires reduces the crosstalk and outside interference. From the above discussion, it is clear that the number of twists per unit of length (e.g., inch) has some effect on the quality of the cable.

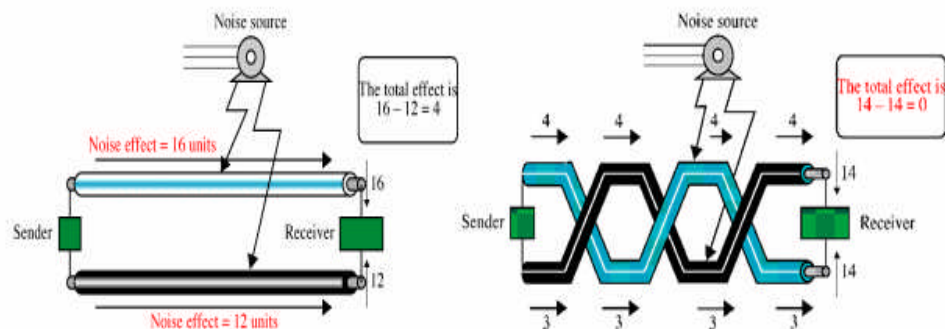


Figure 6.4: Twisting of wires reduces the cross talk and outside interference

Unshielded Vs Shielded Twisted-Pair Cable

The most common twisted-pair cable used in communications is referred to as Unshielded Twisted-pair (UTP). IBM has also produced a version of twisted-pair cable for its use called Shielded Twisted-pair (STP). STP cable has a metal foil or braided mesh covering that encases each pair of insulated conductors. Although metal casing improves the quality of cable by preventing the penetration of noise or crosstalk, it is bulkier and more expensive. Figure 6.5 shows the difference between UTP and STP. Our discussion focuses primarily on UTP because STP is seldom used outside of IBM.

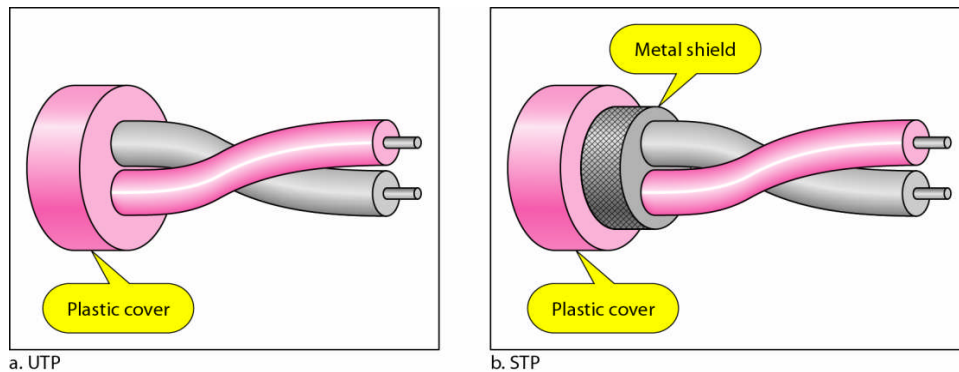


Figure 6.5: UTP and STP cable

Categories

The Electronic Industries Association (EIA) has developed standards to classify unshielded twisted-pair cable into seven categories. Categories are determined by cable quality, with 1 as the lowest and 7 as the highest. Each EIA category is suitable for specific uses. Table 6.1 shows these categories.

Category	Bandwidth (MHz)	Maximum data rate	Application
CAT1	< 1	< 100 Kbps	Telephone Lines
CAT2	4	4 Mbps	IBM Token ring LANs
CAT3	16	16 Mbps (3-4 twists / foot)	10 Base-T LANs Currently used in Telephone Lines
CAT4	20	20 Mbps	16 Mbps Token ring LANs
CAT5	100	100 Mbps 1000 Mbps (4 pairs) (3-4 twists / inch)	100 Base – T (Fast Ethernet) 155 Mbps ATM Network &Gigabit Ethernet
CAT5E	100	100 Mbps 1000 Mbps (4 pairs)	100 Base – T (Fast Ethernet) 155 Mbps ATM Network &Gigabit Ethernet
CAT6	200-250	1 Gbps	Gigabit Ethernet
CAT7	600	1 Gbps	Gigabit Ethernet (over long distance than CAT6)

Table 6.1: Categories of UTP cable

Connectors

The most common UTP connector is RJ45 (RJ stands for registered jack), as shown in Figure 6.6. The RJ45 is a keyed connector, meaning the connector can be inserted in only one way.

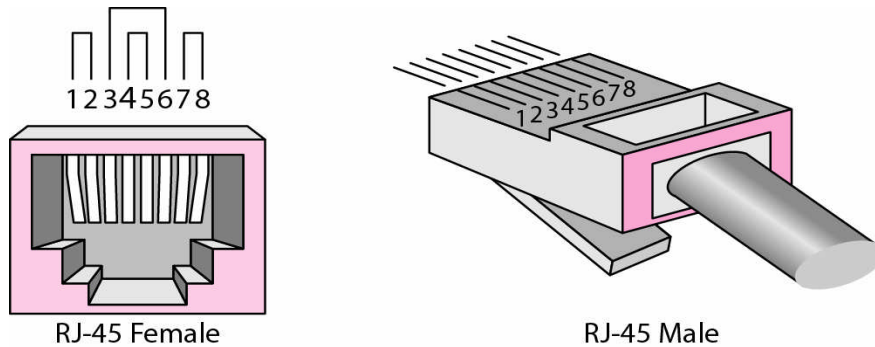


Figure 6.6: UTP – RJ-45 Connector

Applications

Twisted-pair cables are used in telephone lines to provide voice and data channels. Local-area networks, such as 10BaseT and 100Base-T, also use twisted-pair cables.

6.2.2 Coaxial Cable

Consist of two conductors' shares the same axis hence called as coaxial. A solid copper wire runs down the center of the cable, surrounded by insulator (PVC – Poly Vinyl Chloride), surrounded by second conductor (shield), which is further surrounded by insulator and thick plastic jacket forms the cover of the cable as shown in Figure 6.7.

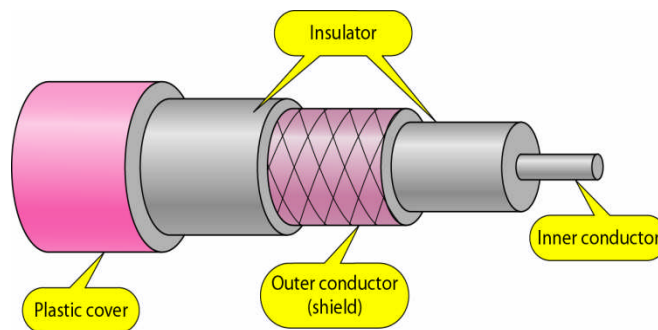


Figure 6.7: Coaxial cable

Coaxial Cable Standards

Coaxial cables are classified by their radio government (RG) ratings and cable's resistance to DC (Direct current) and AC (Alternating current) measured in Ω (ohms). Following Table 6.2 shows the categories of coaxial cable.

Category	Impedance	Application
RG-8 and RG-11	50 Ω	10Base5 – Thick Ethernet
RG-58	50 Ω	10Base2 – Thin Ethernet
RG-59	75 Ω	Cable TV
RG-62	93 Ω	ARCnet (Attached Resource Network)

Table 6.2: Categories of Coaxial cable

Connectors

To connect coaxial cable to devices, we need coaxial connectors. The most common type of connector used today is the Bayone-Neill-Concelman (BNC), connector. Figure 6.8 shows three popular types of these connectors: the BNC connector, the BNC T connector, and the BNC terminator. The BNC connector is used to connect the end of the cable to a device, such as a TV set. The BNC T connector is used in Ethernet networks to branch out to a connection to a computer or other device. The BNC terminator is used at the end of the cable to prevent the reflection of the signal.

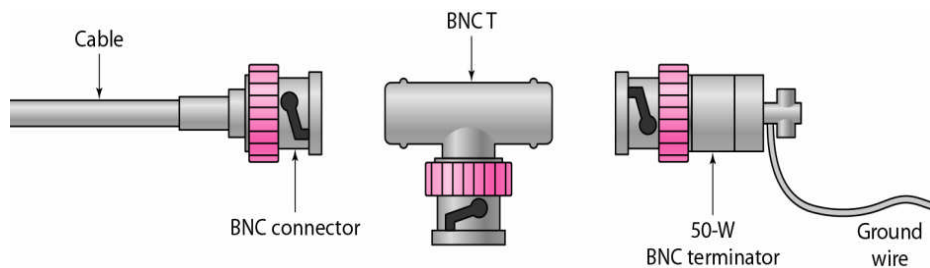


Figure 6.8: BNC connectors

Applications

Coaxial cable was widely used 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 networks has largely been replaced today with fiber-optic cable.

Cable TV networks also use coaxial cables. In the traditional cable TV network, the entire network used coaxial cable. Later, however, cable TV providers replaced most of the media with fiber-optic cable; hybrid networks use coaxial cable only at the network boundaries, near the consumer premises.

6.2.3 Fiber – Optic Cable

A fiber-optic cable is made of glass or plastic and transmits signals in the form of light. To understand optical fiber, we first need to explore several aspects of the nature of light. 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 substance (of a

different density), the ray changes direction. Figure 6.9 shows how a ray of light changes direction when going from a more dense to a less dense substance.

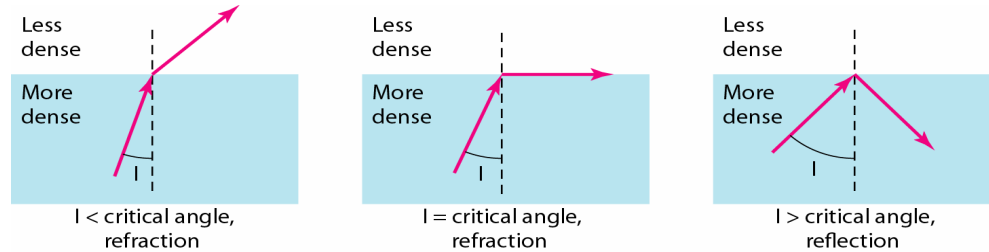


Figure 6.9: Bending of Light rays

As the figure shows, if the angle of incidence I is less than the critical angle, the ray refracts and moves closer to the surface. If the angle of incidence is equal to the critical angle, the light bends along the interface. If the angle is greater than the critical angle, the ray reflects (makes a turn) and travels again in the denser substance. Note that the critical angle is a property of the substance, and its value differs from one substance to another.

Optical fibers use reflection to guide light through a channel. A glass or plastic core is surrounded by a 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 as shown in the following Figure 6.10.

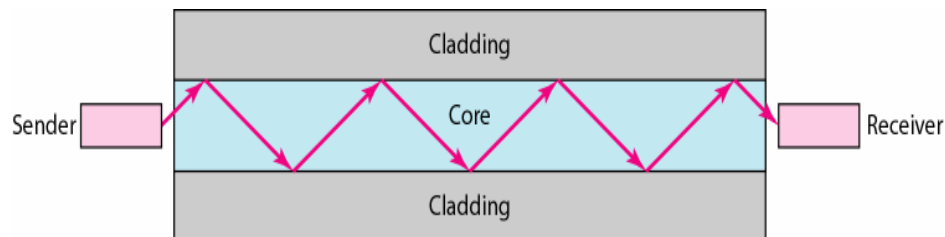


Figure 6.10: Optical – fiber

Propagation Modes

Current technology supports two modes (Multimode and Single mode) for propagating light along optical channels, each requiring fiber with different physical characteristics. Multimode can be implemented in two forms: Step-index or Graded-index (see Figure 6.11).

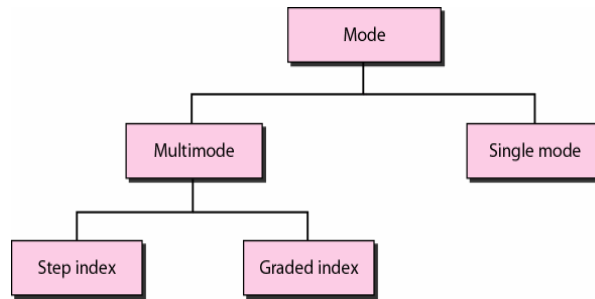


Figure 6.11: Propagation mode – classification

Multimode

It is called Multimode because multiple beams from a light source move through the core in different paths. How these beams move within the cable depends on the structure of the core, as shown in Figure 6.12.

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 a straight-line until it reaches the interface of the core and the cladding. At the interface, there is an abrupt change due to a lower density; this alters the angle of the beam's motion. The term *step index* refers to the suddenness of this change, which contributes to the distortion of the signal as it passes through the fiber.

A second type of fiber, called Multimode Graded-index fiber, decreases this distortion of the signal through the cable. A Graded-index fiber, therefore, is one with varying densities. Density is highest at the center of the core and decreases gradually to its lowest at the edge. Figure 6.12 shows the impact of this variable density on the propagation of light beams.

Single-Mode

Single-mode uses Step-index fiber and a highly focused source of light that limits beams to a small range of angles, all close to the horizontal. The single mode fibers itself is manufactured with a much smaller diameter than that of multimode fiber, and with substantially lower density. The decrease in density results in a critical angle that is close enough to 90° to make the propagation of beams almost horizontal. In this case, propagation of different beams is almost identical, and delays are negligible. All the beams arrive at the destination “together” and can be recombined with little distortion to the signal (see Figure 6.12).

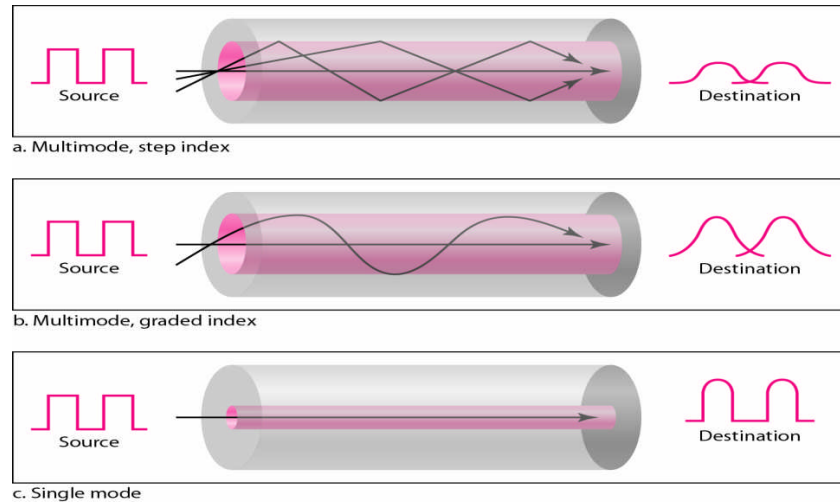


Figure 6.12: Propagation modes

Optical Fiber – Types

Optical fibers are defined by the ratio of the diameter of their core to the diameter of their cladding, both expressed in micrometers. The common sizes are shown in Table 6.3.

Type	Core (μm)	Cladding (μm)	Mode
50/125	50.0	125	Multimode, graded index
62.5/125	62.5	125	Multimode, graded index
100/125	100.0	125	Multimode, graded index
7/125	7.0	125	Single mode

Table 6.3: Optical Fiber – types

Cable Composition

Figure 6.13 shows the composition of a typical fiber-optic cable. The outer jacket is made of either PVC or Teflon. Inside the jacket are Kevlar strands to strengthen the cable. Kevlar is a strong material used in the fabrication of bulletproof vests. Below the Kevlar is another plastic coating to cushion the fiber. The fiber is at the center of the cable, and it consists of cladding and core.

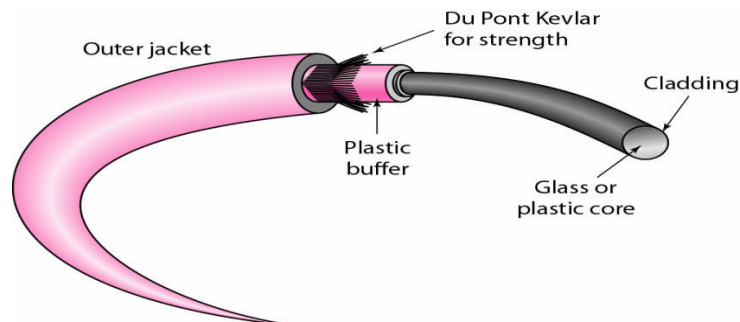


Figure 6.13: Optical Fiber – Composition

Optical Fiber Cable – Connectors

There are three types of connectors for fiber-optic cables, as shown in Figure 6.14. The **Subscriber Channel (SC) connector** is used for cable TV. It uses a push/pull locking system. The **Straight-Tip (ST) connector** is used for connecting cable to networking devices. It uses a bayonet locking system and is more reliable than SC. The **Mechanical Transfer-Registered Jack (MT-RJ)** is a connector that is the same size as **RJ45**.

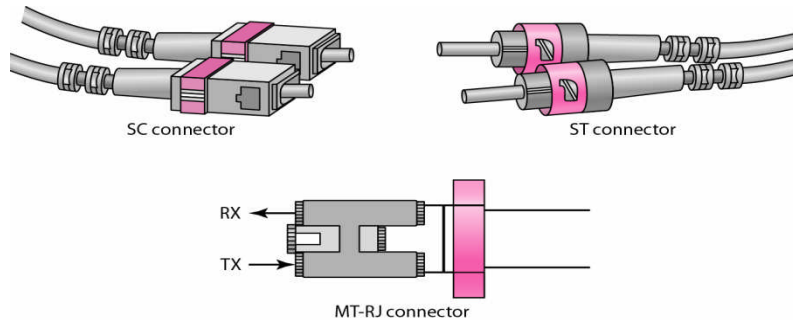


Figure 6.14: Optical Fiber Cable – Connectors

Applications

Fiber-optic cable is often found in backbone networks (The SONET network) because its wider bandwidth is cost-effective. Today, with wavelength-division multiplexing (WDM), we can transfer data at a rate of 1600 Gbps.

Some cable TV companies use a combination of optical fiber and coaxial cable, thus creating a hybrid network. Optical fiber provides the backbone structure while coaxial cable provides the connection to the user premises.

LANs such as 100Base-FX network (Fast Ethernet) and 1000Base-X (Gigabit Ethernet) also use fiber-optic cable.

Advantages of Optical Fiber

Fiber-optic cable has several advantages over metallic cable (twisted pair or coaxial)

- **Higher bandwidth:** Fiber-optic cable can support dramatically higher bandwidths(data rates) than either twisted-pair or coaxial cable. Currently, data rates and bandwidth utilization over fiber-optic cable are limited not by the medium but by the signal generation and reception technology available.
- **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 every 5 km for coaxial or twisted-pair cable.
- **Immunity to electromagnetic interference:** Electromagnetic noise cannot affect fiber-optic cables.
- **Resistance to corrosive materials:** Glass is more resistant to corrosive materials than copper.

- **Light weight:** Fiber-optic cables are much lighter than copper cables.
- **Greater immunity to tapping:** Fiber-optic cables are more immune to tapping than copper cables.

Disadvantages of Optical Fiber

There are some disadvantages in the use of optical fiber.

- **Installation and maintenance:** Fiber-optic cable is a relatively new technology. Its installation and maintenance require expertise that is not yet available everywhere.
- **Unidirectional light propagation:** Propagation of light is unidirectional. If we need bidirectional communication, two fibers are needed.
- **Cost:** The cable and the interfaces are relatively more expensive than those of other guided media

6.3 UNGUIDED MEDIA – WIRELESS

Unguided media transport electromagnetic waves without using a physical conductor. This type of communication is often referred to as wireless communication. Signals are normally broadcast through free space and thus are available to anyone who has a device capable of receiving them. Figure 6.15 shows the part of the electromagnetic spectrum, ranging from 3 kHz to 900 THz, used for wireless communication.

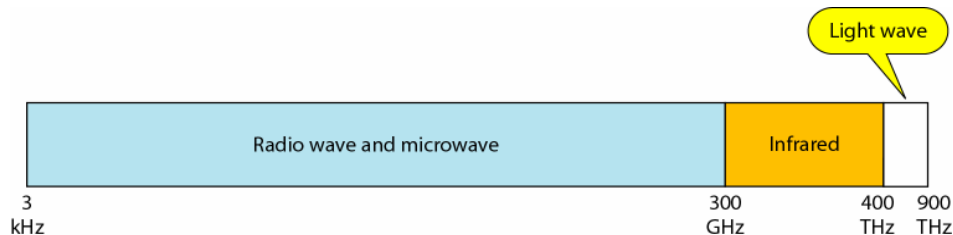


Figure 6.15: Electromagnetic Spectrum for Wireless communication

Propagation Methods

Unguided signals can travel from the source to destination in several ways: ground propagation, sky propagation, and line-of-sight propagation, as shown in Figure 6.16.

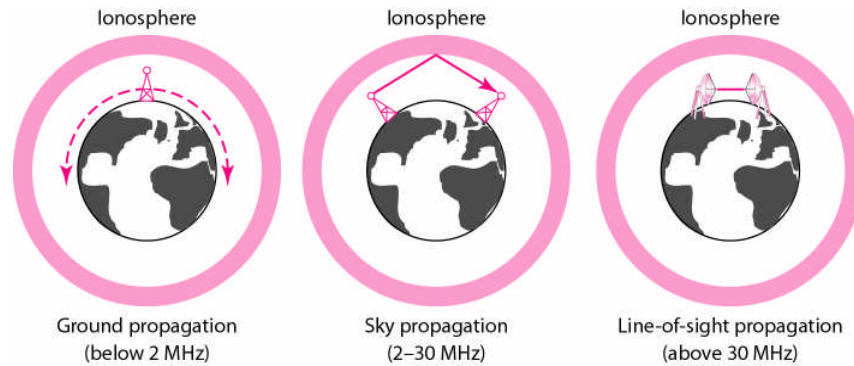


Figure 6.16: Propagation methods

In Ground propagation, radio waves travel through the lowest portion of the atmosphere, hugging the earth. These low-frequency signals emanate in all directions from the transmitting antenna and follow the curvature of the planet. Distance depends on the amount of power in the signal: The greater the power, the greater the distance.

In Sky propagation, higher-frequency radio waves radiate upward into the ionosphere (the layer of atmosphere where particles exist as ions) where they are reflected back to earth. This type of transmission allows for greater distances with lower output power.

In Line-or-sight propagation, very high-frequency signals 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. Line-of-sight propagation is tricky because radio transmissions cannot be completely focused.

The section of the electromagnetic spectrum defined as radio waves and microwaves is divided into eight ranges, called **bands**, each regulated by government authorities. These bands are rated from *Very Low Frequency* (VLF) to *Extremely High Frequency* (EHF). Table 6.4 lists these bands, their ranges, propagation methods, and some applications.

<i>Band</i>	<i>Range</i>	<i>Propagation</i>	<i>Application</i>
VLF (very low frequency)	3–30 kHz	Ground	Long-range radio navigation
LF (low frequency)	30–300 kHz	Ground	Radio beacons and navigational locators
MF (middle frequency)	300 kHz–3 MHz	Sky	AM radio
HF (high frequency)	3–30 MHz	Sky	Citizens band (CB), ship/aircraft communication
VHF (very high frequency)	30–300 MHz	Sky and line-of-sight	VHF TV, FM radio
UHF (ultrahigh frequency)	300 MHz–3 GHz	Line-of-sight	UHF TV, cellular phones, paging, satellite
SHF (superhigh frequency)	3–30 GHz	Line-of-sight	Satellite communication
EHF (extremely high frequency)	30–300 GHz	Line-of-sight	Radar, satellite

Table 6.4: Bands

We can divide wireless transmission into three broad groups: Radio waves, Microwaves, and Infrared waves.

6.3.1 Radio Waves

Although there is no clear-cut demarcation between radio waves and microwaves, electromagnetic waves ranging in frequencies between 3 kHz and 1 GHz are normally called Radio waves; waves ranging in frequencies between 1 and 300 GHz are called microwaves.

However, the behavior of the waves, rather than the frequencies, is a better criterion for classification.

Radio waves, for the most part, are omni directional. When an antenna transmits radio waves, they are propagated in all directions. This means that the sending and receiving antennas do not have to be aligned. The omni directional property has a disadvantage, too. The radio waves transmitted by one antenna are susceptible to interference by another antenna that may send signals using the same frequency or band.

Radio waves, particularly those waves that propagate in the sky mode, can travel long distances. This makes radio waves a good candidate for long-distance broadcasting such as AM radio.

Radio waves, particularly those of low and medium frequencies, can penetrate walls. This characteristic can be both an advantage and a disadvantage. It is an advantage because, for example, an AM radio can receive signals inside a building. It is a disadvantage because we cannot isolate a communication to just inside or outside a building. The radio wave band is relatively narrow, just under 1 GHz, compared to the microwave band.

Omni directional Antenna

Radio waves use omni directional antennas that send out signals in all directions. Based on the wavelength, strength, and the purpose of transmission, we can have several types of antennas. Figure 6.17 shows an omni directional antenna.

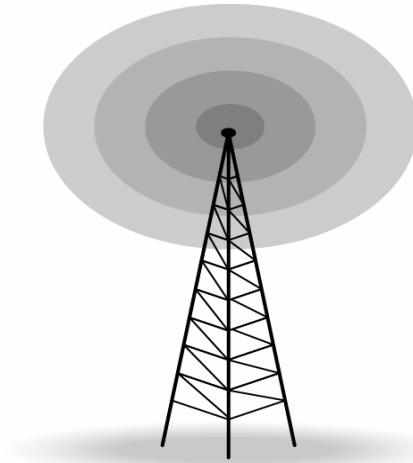


Figure 6.17: Omni directional antenna

Applications

The omni directional characteristics of radio waves make them useful for multicasting, in which there is one sender but many receivers. AM and FM radio, television, maritime radio, cordless phones, and paging are examples of multicasting.

6.3.2 Microwaves

Electromagnetic waves having frequencies between 1 and 300 GHz are called microwaves. Microwaves are unidirectional. When an antenna transmits microwave waves, they can be narrowly focused. This means that the sending and receiving antennas need to be aligned. The unidirectional property has an obvious advantage. A pair of antennas can be aligned without interfering with another pair of aligned antennas. The following describes some characteristics of microwave propagation:

- Microwave propagation is line-of-sight. Since the towers with the mounted antennas need to be in direct sight of each other, towers that are far apart need to be very tall. Repeaters are often needed for long distance communication.
- Very high-frequency microwaves cannot penetrate walls. This characteristic can be a disadvantage if receivers are inside buildings.
- The microwave band is relatively wide, almost 299 GHz. Therefore wider sub-bands can be assigned, and a high data rate is possible.
- Use of certain portions of the band requires permission from authorities.

Unidirectional Antenna

Microwaves need unidirectional antennas that send out signals in one direction. Two types of antennas are used for microwave communications: the parabolic dish and the horn (see Figure 6.18).

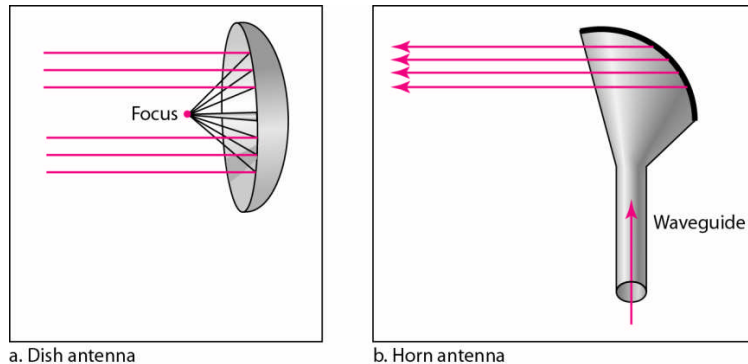


Figure 6.18: Unidirectional antenna

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 all the lines intersect in a common point called the focus. The parabolic dish works as a funnel, catching a wide range of waves and directing them to a common point. In this way, more of the signal is recovered than would be possible with a single-point receiver. Outgoing transmissions are broadcast through a horn aimed at the dish. The microwaves hit the dish and are deflected outward in a reversal of the receipt path.

A horn antenna looks like a gigantic scoop. Outgoing transmissions are broadcast up a stem (resembling a handle) and deflected outward in a series of narrow parallel beams by the curved head. Received transmissions are collected by the scooped shape of the horn, in a manner similar to the parabolic dish, and are deflected down into the stem.

We can categories the Microwave systems into two categories such as terrestrial-microwave and satellite-microwave systems.

Terrestrial Microwaves

Terrestrial microwave uses parabolic dish, signals are highly focused and line of sight is maintained between sender and receiver (see Figure 6.19). It is used in Long haul telecommunications.

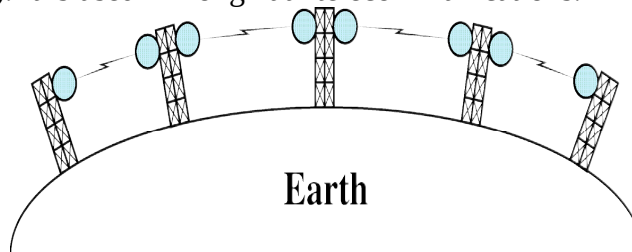


Figure 6.19: Terrestrial Microwave system

Satellite Microwaves

In this, satellite acts as a relay station (repeater). Satellite receives signals on one frequency (uplink) from sending earth station, amplifies or repeats the signals and transmits on another frequency (downlink) to the receiving earth station. Satellite need to be positioned in the geosynchronous orbit (height of 36,000 km from earth) with the help of space shuttle. Geosynchronous means stationary with respect to the earth, hence the position of particular Earth Station with respect to the satellite remains constant at all times(see Figure 6.20). It is used in Television, Long distance telephone and Private business networks.

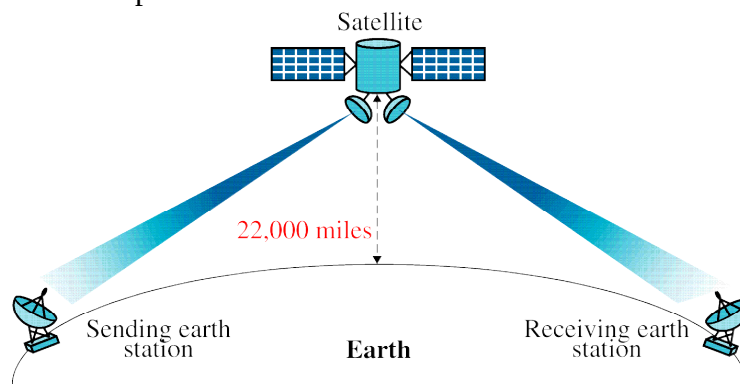


Figure 6.20: Satellite Microwave system

Microwaves Applications

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

6.3.3 Infrared

Infrared waves, with frequencies from 300 GHz to 400 THz (wavelengths from 1 mm to 770 nm), can be used for short-range communication. Infrared waves, having high frequencies, 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. However, this same characteristic makes infrared signals useless for long-range communication. In addition, we cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication.

Applications

The infrared band, almost 400 THz, has an excellent potential for data transmission. Such a wide bandwidth can be used to transmit digital data with a very high data rate. The *Infrared Data Association* (IrDA), an association for sponsoring the use of infrared waves, has established standards for using these signals for communication between devices such as keyboards, mice, PCs, and printers. For example, some manufacturers provide a special port called the IrDA port that allows a wireless keyboard

to communicate with a PC. The standard originally defined a data rate of 75 kbps for a distance up to 8 m. The recent standard defines a data rate of 4 Mbps.

Infrared signals defined by IrDA transmit through line of sight; the IrDA port on the keyboard needs to point to the PC for transmission to occur.

6.4 SWITCHING

A network is a set of connected devices. Whenever we have multiple devices, we have the problem of how to connect them to make one-to-one communication possible. One solution is to make a point-to-point connection between each pair of devices (a mesh topology) or between a central device and every other device (a star topology). These methods, however, are impractical and wasteful when applied to very large networks.

A better solution is switching. A switched network consists of a series of interlinked nodes, called switches. Switches are devices capable of creating temporary connections between two or more devices linked to the switch. In a switched network, some of these nodes are connected to the end systems (computers or telephones). Others are used only for routing. Figure 6.21 shows a switched network.

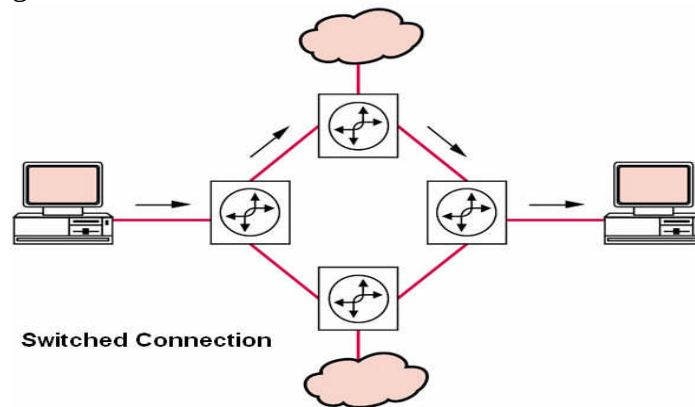


Figure 6.21: Switched network

Traditionally, three methods of switching have been important: circuit switching, packet switching, and message switching. The first two are commonly used today. The third has been phased out in general communications but still has networking applications. We can then divide today's networks into three broad categories: circuit-switched networks, packet-switched networks, and message-switched. Packet-switched networks can further be divided into two subcategories-virtual-circuit networks and datagram networks as shown in Figure 6.22.

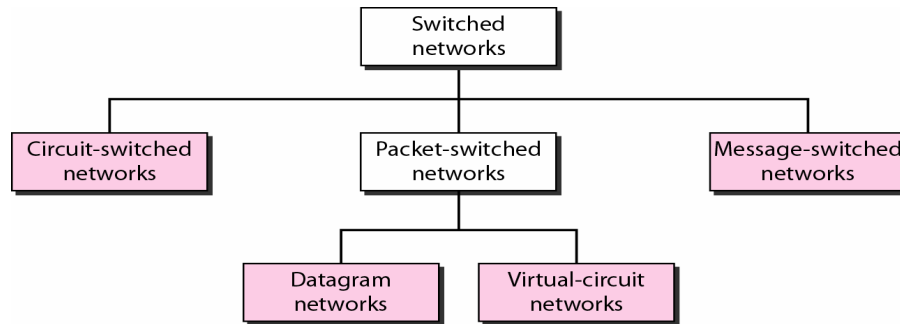


Figure 6.22: Classification of Switched networks

6.4.1 Circuit – Switching (Circuit Switched Networks)

A circuit-switched network consists of a set of switches connected by physical links. A connection between two stations is a dedicated path made of one or more links. However, each connection uses only one dedicated channel on each link. Each link is normally divided into n channels by using FDM or TDM as discussed in Chapter 5. Figure 6.23 shows a trivial circuit-switched network with four switches and four links. Each link is divided into n (n is 3 in the figure) channels by using FDM or TDM.

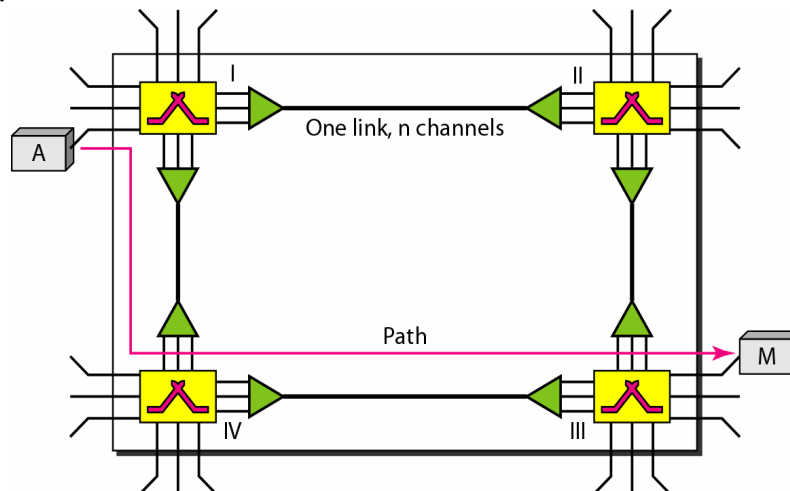


Figure 6.23: Trivial Circuit-switched network

We have explicitly shown the multiplexing symbols to emphasize the division of the link into channels even though multiplexing can be implicitly included in the switch fabric. The end systems, such as computers or telephones, are directly connected to a switch. We have shown only two end systems for simplicity. When end system A needs to communicate with end system M, system A needs to request a connection to M that must be accepted by all switches as well as by M itself. This is called the setup phase; a circuit (channel) is reserved on each link, and the combination of circuits or channels defines the dedicated path. After the dedicated path made of connected circuits (channels) is established, data transfer can take place. After all data have been transferred, the circuits are turned down.

We need to emphasize several points here:

- Circuit switching takes place at the physical layer.
- Before starting communication, the stations must make a reservation for the resources to be used during the communication. These resources, such as channels (bandwidth in FDM and time slots in TDM), switch buffers, switch processing time, and switch input/output ports, must remain dedicated during the entire duration of data transfer until the teardown phase.
- Data transferred between the two stations are not packetized (physical layer transfer of the signal). The data are a continuous flow sent by the source station and received by the destination station, although there may be periods of silence.
- There is no addressing involved during data transfer. The switches route the data based on their occupied band (FDM) or time slot (TDM). Of course, there is end-to-end addressing used during the setup phase.

Example 6.1:

As a trivial example, let us use a circuit-switched network to connect eight telephones in a small area. Communication is through 4-kHz voice channels. We assume that each link uses FDM to connect a maximum of two voice channels. The bandwidth of each link is then 8 kHz. Figure 6.24 shows the situation. Telephone 1 is connected to telephone 7; 2 to 5; 3 to 8; and 4 to 6. Of course the situation may change when new connections are made. The switch controls the connections.

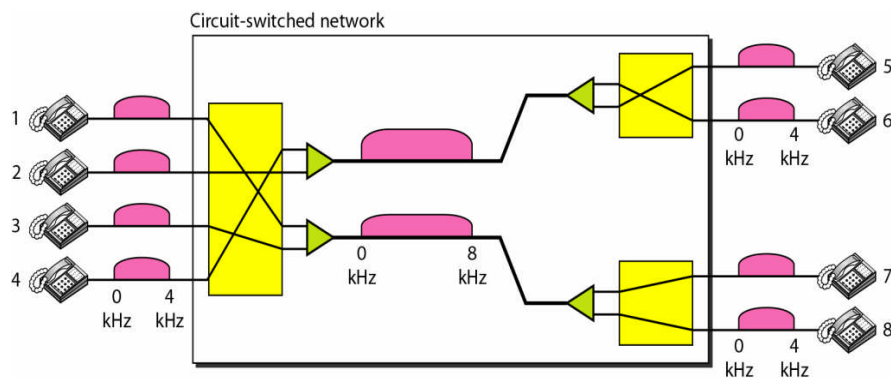


Figure 6.24: Example 6.1(Circuit-switched network)

Example 6.2:

As another example, consider a circuit-switched network that connects computers in two remote offices of a private company. The offices are connected using a T-1 line leased from a service provider. There are two 4 X 8 (4 inputs and 8 outputs) switches in this network. For each switch, four output ports are folded into the input ports to allow communication between computers in the same office. Four other output ports allow communication between the two offices. Figure 6.25 shows the situation.

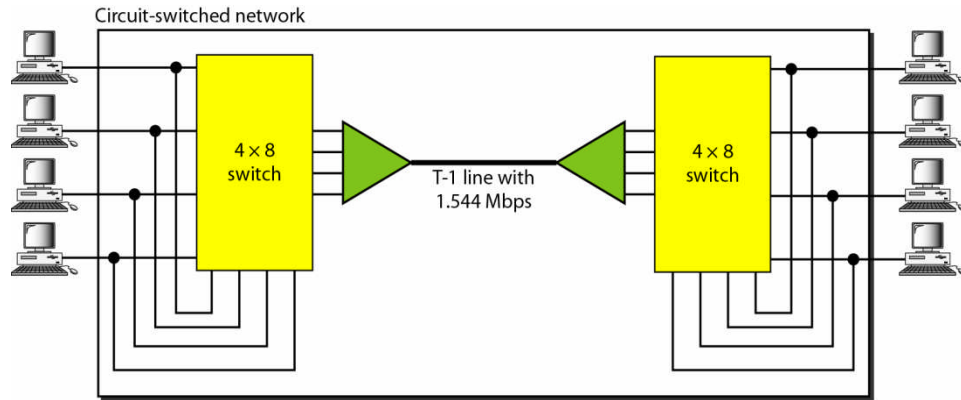


Figure 6.25: Example 6.2 (Circuit-switched network)

Connection Phases

The actual communication in a circuit-switched network requires three phases: connection setup, data transfer, and connection teardown.

Connection setup phase

Before the two parties can communicate, a dedicated circuit (combination of channels in links) needs to be established. The end systems are normally connected through dedicated lines to the switches, so connection setup means creating dedicated channels between the switches. For example, in Figure 6.23, when system A needs to connect to system M, it sends a setup request that includes the address of system M, to switch I. Switch I finds a channel between itself and switch IV that can be dedicated for this purpose. Switch I then sends the request to switch IV, which finds a dedicated channel between itself and switch III. Switch III informs system M of system A's intention at this time.

In the next step to making a connection, an acknowledgment from system M needs to be sent in the opposite direction to system A. Only after system A receives this acknowledgment is the connection established. Note that end-to-end addressing is required for creating a connection between the two end systems. These can be, for example, the addresses of the computers assigned by the administrator in a TDM network, or telephone numbers in an FDM network.

Data Transfer phase

After the establishment of the dedicated circuit (channels), the two parties can transfer data.

Connection teardown phase

When one of the parties needs to disconnect, a signal is sent to each switch to release the resources.

Efficiency

It can be argued that circuit-switched networks are not as efficient as the other two types of networks because resources are allocated during the entire duration of the connection. These resources are unavailable to

other connections. In a telephone network, people normally terminate the communication when they have finished their conversation. However, in computer networks, a computer can be connected to another computer even if there is no activity for a long time. In this case, allowing resources to be dedicated means that other connections are deprived.

Delay

Although a circuit-switched network normally has low efficiency, the delay in this type of network is minimal. During data transfer the data are not delayed at each switch; the resources are allocated for the duration of the connection.

Circuit-switching in Telephone networks

Switching at the physical layer in the traditional telephone network uses the circuit-switching approach.

6.4.2 Packet – Switching (Packet Switched Networks)

In data communications, we need to send messages from one end system to another. If the message is going to pass through a packet-switched network, it needs to be divided into packets of fixed or variable size. The size of the packet is determined by the network and the governing protocol. Each packet contains data and header (priority, source and destination address). In packet switching, there is no resource allocation for a packet. This means that there is no reserved bandwidth on the links, and there is no scheduled processing time for each packet. Resources are allocated on demand. The allocation is done on a first-come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed. Figure 6.26 shows the Packet-switched network.

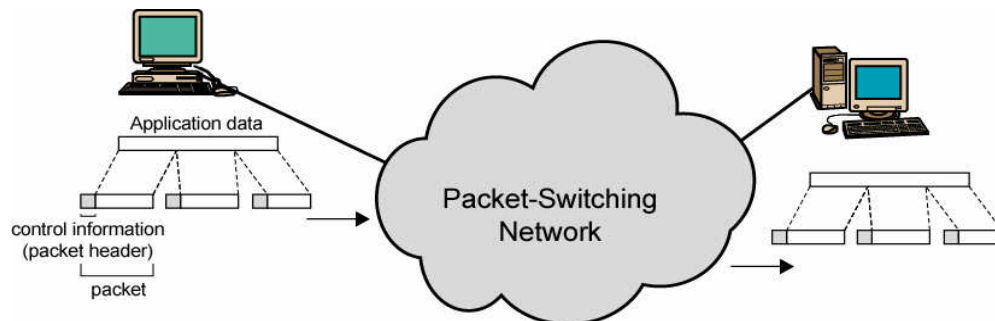


Figure 6.26: Packet-switched network

Packet-switched networks can further be divided into two subcategories Virtual-circuit networks and Datagram networks.

6.4.2.1 Datagram switching (Datagram networks)

In a datagram network, each packet is treated independently from all other packets and is referred as datagram packet. Datagram packets belongs to same message may go by different paths to reach at the destination and they may arrive out of order at the destination. Transport

layer reorders them at the destination. Datagram switching is normally done at the network layer.

Figure 6.27 shows how the datagram approach is used to deliver four packets from station A to station X.

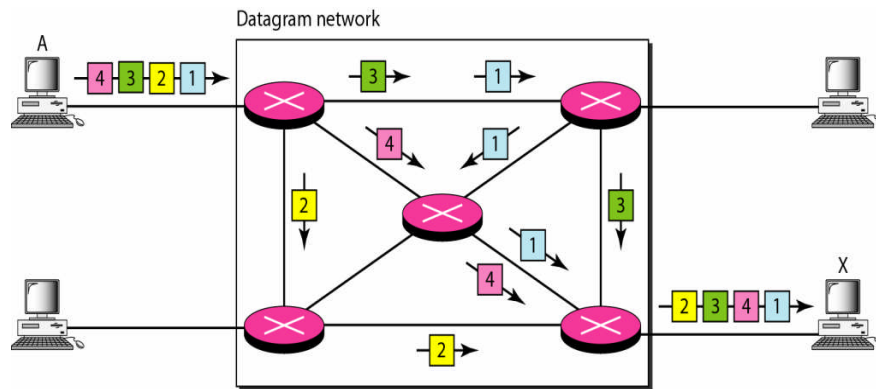


Figure 6.27: Datagram network

The datagram networks are sometimes referred to as connectionless networks. The term *connectionless* here means that the switch (packet switch) does not keep information about the connection state. There are no connection setup or teardown phases. Each packet is treated the same by a switch regardless of its source or destination.

Efficiency

The efficiency of a datagram network is better than that of a circuit-switched network; resources are allocated only when there are packets to be transferred. If a source sends a packet and there is a delay of a few minutes before another packet can be sent, the resources can be reallocated during these minutes for other packets from other sources.

Delay

There may be greater delay in a datagram network than in a virtual-circuit network. Although there are no setup and teardown phases, each packet may experience a wait at a switch before it is forwarded. In addition, since not all packets in a message necessarily travel through the same switches, the delay is not uniform for the packets of a message.

Datagram Networks in the Internet

The Internet has chosen the datagram approach to switching at the network layer. It uses the universal addresses (IP addresses) defined in the network layer to route packets from the source to the destination.

6.4.2.2 Virtual – Circuit Switching (Virtual-circuit networks)

A virtual-circuit network is a cross between a circuit-switched network and a datagram network. It has some characteristics of both.

1. As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.
2. Resources can be allocated during the setup phase, as in a circuit-switched network, or on demand, as in a datagram network.
3. As in a datagram network, data are packetized and each packet carries an address in the header.
4. As in a circuit-switched network, all packets follow the same path established during the connection.
5. A virtual-circuit network is normally implemented in the data link layer, while a circuit-switched network is implemented in the physical layer and a datagram network in the network layer. But this may change in the future.

Figure 6.28 is an example of a virtual-circuit network. The network has switches that allow traffic from sources to destinations. A source or destination can be a computer, packet switch, or a device that connects other networks.

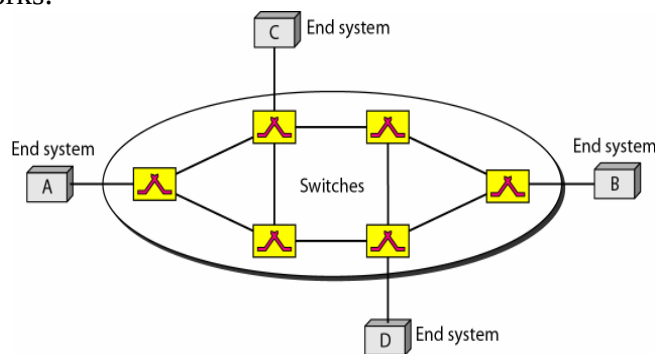


Figure 6.28: Virtual-circuit network

A Virtual-circuit switching further can be implemented in two ways Switched Virtual-circuit (SVC) and Permanent Virtual-circuit (PVC).

Switched Virtual-circuit (SVC)

It works like a dial up line in circuit switching. Circuit is created whenever needed and exist only for the duration of specific exchange. Figure 6.29 shows an example of switched virtual-circuit.

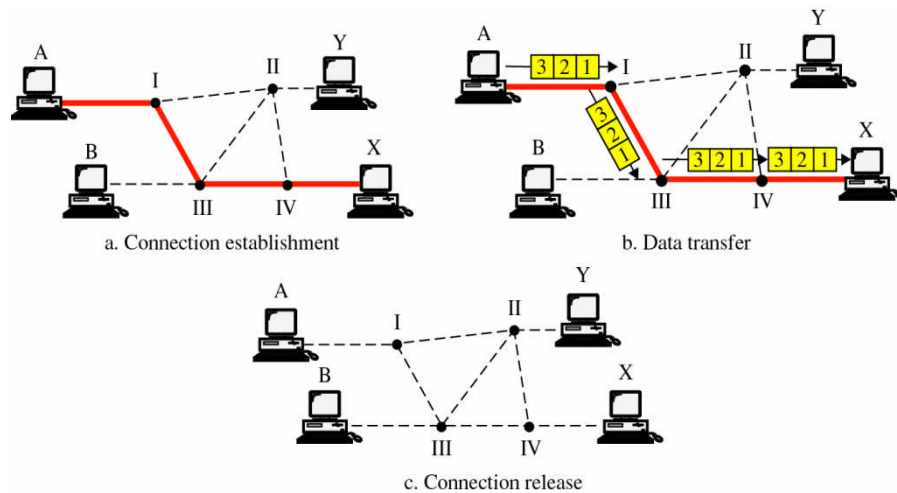
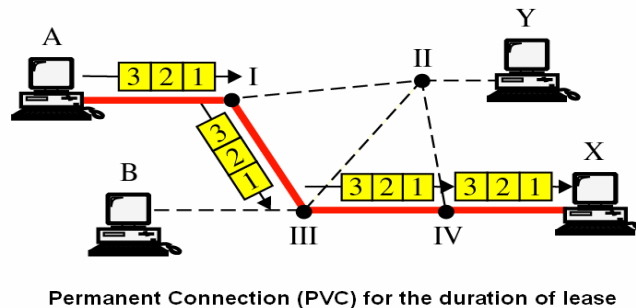


Figure 6.29: Switched virtual-circuit (SVC)

Permanent Virtual-circuit (PVC)

It works like a leased line in circuit switching, where same virtual circuit is provided between two users on a continuous basis. Virtual circuit is dedicated to specific users and no one else can use it and can be used without connection establishment and connection termination. Figure 6.30 shows an example of permanent virtual-circuit.



Permanent Connection (PVC) for the duration of lease

Figure 6.30: Permanent Virtual-circuit (PVC)

Efficiency in Virtual-Circuit Networks

As we said before, resource reservation in a virtual-circuit network can be made during the setup or can be on demand during the data transfer phase. In the first case, the delay for each packet is the same; in the second case, each packet may encounter different delays.

Delay in Virtual-Circuit Networks

In a virtual-circuit network, there is a one-time delay for setup and a one-time delay for teardown. If resources are allocated during the setup phase, there is no wait time for individual packets.

Circuit-Switched Technology in WANs

Virtual-circuit networks are used in switched WANs such as Frame Relay and ATM networks. The data link layer of these technologies is well suited to the virtual-circuit technology.

6.4.3 Message – Switching (Message Switched Networks)

The third method of switching is a message switching; which has been phased out in general communications but still has networking applications. It was best known by the term **store and forward** and most common in the 1960s and 1970s.

In message-switching when a switching node receives a message, stores it until the appropriate route is free, then it sends along and the messages were stored and relayed from its secondary storage (disk). Requirement of large capacity storage media at each switching node was the major disadvantage of this method. Figure 6.31 shows the example of message switching.

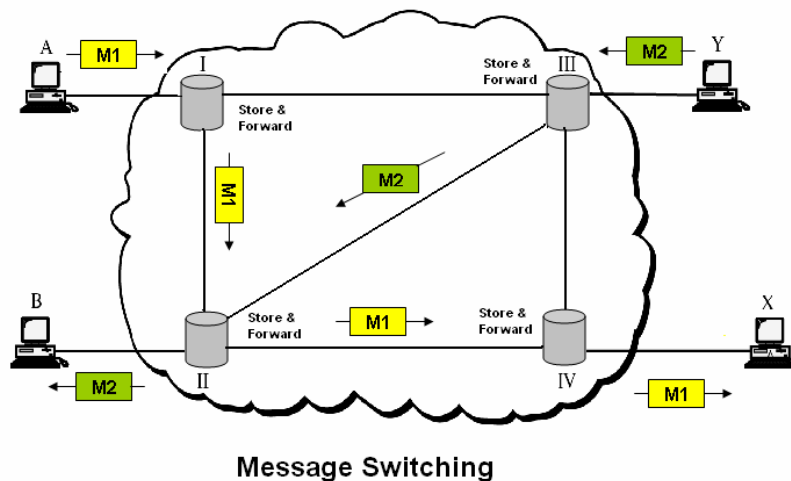


Figure 6.31: Message switching

6.5 STRUCTURE OF SWITCH

We use switches in circuit-switched and packet-switched networks. In this section, we discuss the structures of the switches used in each type of network.

6.5.1 Structure of Circuit Switch

Circuit switching today can use either of two technologies: the space-division switch or the time-division switch.

Space-Division Switch

In space-division switching, the paths in the circuit are separated from one another spatially. This technology was originally designed for use in analog networks but is used currently in both analog and digital networks. It has evolved through a long history of many designs.

Crossbar Switch

A crossbar switch connects n inputs to m outputs in a grid; using electronic micro-switches (transistors) at each cross-point (see Figure

6.32). The major limitation of this design is the number of cross-points required. To connect n inputs to m outputs using a crossbar switch requires $n \times m$ cross-points. For example, to connect 100 inputs to 100 outputs requires a switch with 10000 cross-points. A crossbar switch with this number of cross-points is impractical. Such a switch is also inefficient because statistics show that, in practice, fewer than 25 percent of the cross-points are in use at any given time and the rest are idle.

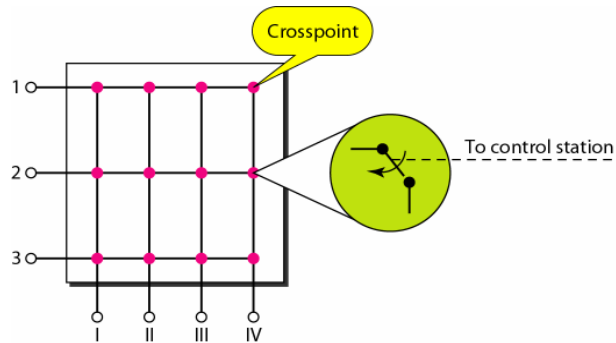


Figure 6.32: Crossbar switch with 3 inputs and 4 outputs

Multistage Switch

The solution to the limitations of the crossbar switch is the multistage switch, which combines crossbar switches in several (normally three) stages, as shown in Figure 6.33. In a single crossbar switch, only one row or column (one path) is active for any connection. So we need $N \times N$ cross-points. If we can allow multiple paths inside the switch, we can decrease the number of cross-points. Each cross-point in the middle stage can be accessed by multiple cross-points in the first or third stage.

In a three-stage switch, the total number of cross-points is $2kN + k(N/n)^2$ which is much smaller than the number of cross-points in a single-stage switch (N^2).

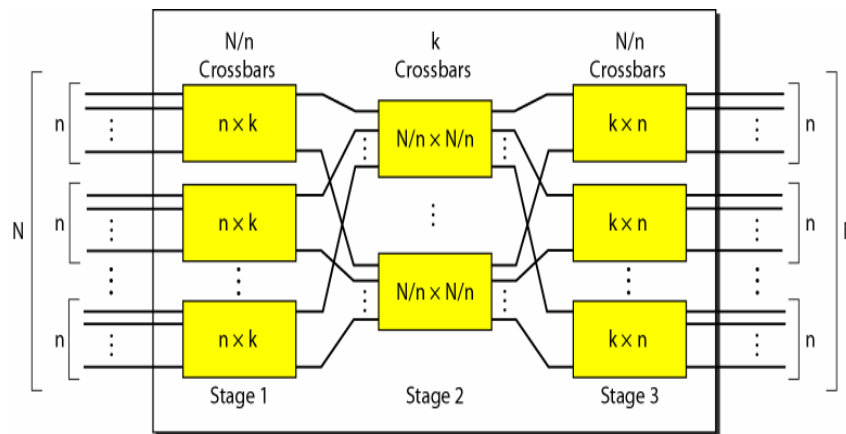


Figure 6.33: Multistage switch

Example 6.3:

Design a three-stage, 200×200 switch ($N = 200$) with $k = 4$ and $n = 20$.

Solution:

In the first stage we have N/n or **10** crossbars switches, each of size **20 x 4**. In the second stage, we have **4** crossbars switches, each of size **10 x 10**. In the third stage, we have **10** crossbars switches, each of size **4 x 20**.

The total number of cross-points is $2kN + k(N/n)^2$ or **2000** cross-points. This is 5 percent of the number of cross-points in a single-stage switch (**200 x 200 = 40,000**).

Time-Division switch

Time-division switching uses time-division multiplexing (TDM) inside a switch. The most popular technology is called the time-slot interchange (TSI). Time-Slot Interchange Figure 6.34 shows a system connecting four input lines to four output lines. Imagine that each input line wants to send data to an output line according to the following pattern for example, **1→3, 2→4, 3→1 and 4→2**.

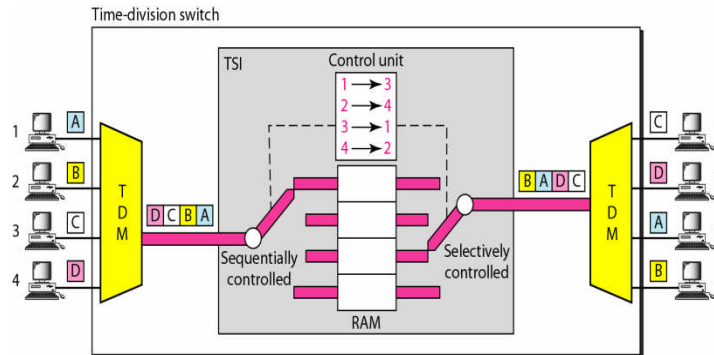


Figure 6.34: Time-division switch

The figure combines a TDM multiplexer, a TDM de-multiplexer, and a TSI consisting of random access memory (RAM) with several memory locations. The size of each location is the same as the size of a single time slot. The number of locations is the same as the number of inputs (in most cases, the numbers of inputs and outputs are equal). The RAM fills up with incoming data from time slots in the order received. Slots are then sent out in an order based on the decisions of a control unit.

6.5.2 Structure of Packet Switch

A switch used in a packet-switched network has a different structure from a switch used in a circuit-switched network. We can say that a packet switch has four components: input ports, output ports, the routing processor, and the switching fabric, as shown in Figure 6.35.

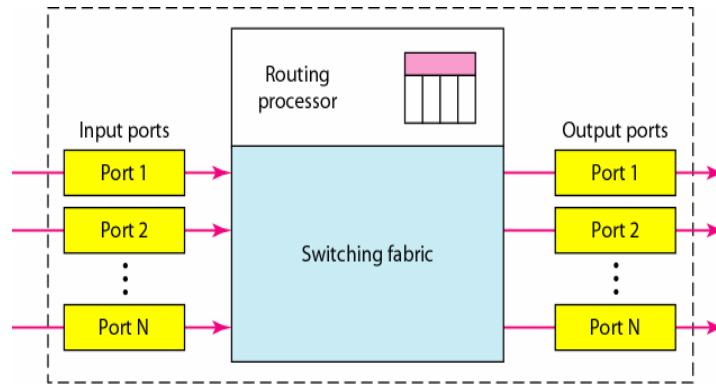


Figure 6.35: Components of Packet-switch

Input Ports

An input port performs the physical and data link functions of the packet switch. The bits are constructed from the received signal. The packet is encapsulated from the frame. Errors are detected and corrected. The packet is now ready to be routed by the network layer. In addition to a physical layer processor and a data link processor, the input port has buffers (queues) to hold the packet before it is directed to the switching fabric. Figure 6.36 shows a schematic diagram of an input port.

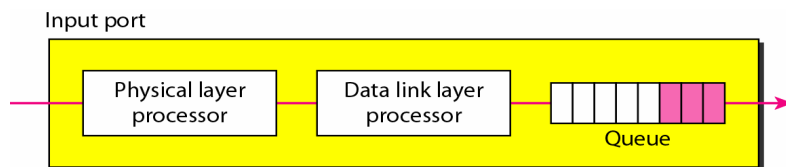


Figure 6.36: Input port

Output Port

The output port performs the same functions as the input port, but in the reverse order. First the outgoing packets are queued, then the packet is encapsulated in a frame, and finally the physical layer functions are applied to the frame to create the signal to be sent on the line. Figure 6.37 shows a schematic diagram of an output port.

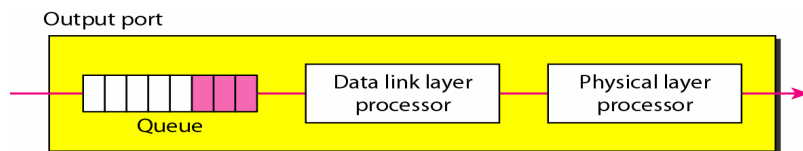


Figure 6.37: Output port

Routing Processor

The routing processor performs the functions of the network layer. The destination address is used to find the address of the next hop and, at the same time, the output port number from which the packet is sent out. This activity is sometimes referred to as **table lookup** because the routing processor searches the routing table. In the newer packet switches, this

function of the routing processor is being moved to the input ports to facilitate and expedite the process.

Switching Fabrics

The most difficult task in a packet switch is to move the packet from the input queue to the output queue. The speed with which this is done affects the size of the input/output queue and the overall delay in packet delivery. In the past, when a packet switch was actually a dedicated computer, the memory of the computer or a bus was used as the switching fabric. The input port stored the packet in memory; the output port retrieved the packet from memory. Today, packet switches are specialized mechanisms that use a variety of switching fabrics.

6.6 SUMMARY

Transmission Media

Transmission media lie below the physical layer.

- A guided medium provides a physical conduit from one device to another. Twisted pair cable, coaxial cable, and optical fiber are the most popular types of guided media.
- Twisted-pair cable consists of two insulated copper wires twisted together. Twisted pair cable is used for voice and data communications.
- Coaxial cable consists of a central conductor and a shield. Coaxial cable can carry signals of higher frequency ranges than twisted-pair cable. Coaxial cable is used in cable TV networks and traditional Ethernet LANs.
- Fiber-optic cables are composed of a glass or plastic inner core surrounded by cladding, all encased in an outside jacket. Fiber-optic cables carry data signals in the form of light. The signal is propagated along the inner core by reflection. Fiber-optic cable is used in backbone networks, cable TV networks, and Fast Ethernet networks.
- Unguided media (free space) transport electromagnetic waves without the use of a physical conductor.
- Wireless data are transmitted through ground propagation, sky propagation, and line-of-sight propagation. Wireless waves can be classified as radio waves, microwaves, or infrared waves. Radio waves are omni directional; microwaves are unidirectional. Microwaves are used for cellular phone, satellite, and wireless LAN communications.
- Infrared waves are used for short-range communications such as those between a PC and a peripheral device. It can also be used for indoor LANs.

Switching

A switched network consists of a series of interlinked nodes, called switches. Traditionally three methods of switching have been important: circuit switching, packet switching, and message switching.

- We can divide today's networks into three broad categories: circuit-switched networks, packet-switched networks, and message-switched. Packet-switched networks can also be divided into two subcategories: virtual-circuit networks and datagram networks.
- A circuit-switched network is made of a set of switches connected by physical links, in which each link is divided into n channels. Circuit switching takes place at the physical layer. In circuit switching, the resources need to be reserved during the setup phase; the resources remain dedicated for the entire duration of data transfer phase until the teardown phase.
- In packet switching, there is no resource allocation for a packet. This means that there is no reserved bandwidth on the links, and there is no scheduled processing time for each packet. Resources are allocated on demand.
- In a datagram network, each packet is treated independently of all others. Packets in this approach are referred to as datagrams. There are no setup or teardown phases.
- A virtual-circuit network is a cross between a circuit-switched network and a datagram network. It has some characteristics of both.
- Circuit switching uses either of two technologies: the space-division switch or the time-division switch.
- A switch in a packet-switched network has a different structure from a switch used in a circuit-switched network. We can say that a packet switch has four types of components: input ports, output ports, a routing processor, and switching fabric.

6.7 REFERENCE FOR FURTHER READING

For more details about topic – Transmission media and Switching discussed in this chapter, we recommend the following books.

1. *Data Communication and Networking* by Behrouz A. Forouzan, McGraw-Hill, 2007.
2. *Data and Computer Communications* by W. Stallings, Prentice Hall, 2004.
3. *Communication Networks* by Alberto Leon-Gracia & Indra Widjaja, McGraw-Hill, 2003.
4. *Computer Networks* by Andrew S. Tanenbaum, Prentice Hall, 2003.
5. *Digital Telephony* by J. Bellamy, Wiley, 2000.

6.8 MODEL QUESTIONS

Transmission Media

1. What is the position of the transmission media in the OSI or the Internet model?
2. Name the two major categories of transmission media.
3. How do guided media differ from unguided media?
4. What are the three major classes of guided media?
5. What is the significance of the twisting in twisted-pair cable?
6. What is refraction? What is reflection?
7. What is the purpose of cladding in an optical fiber?
8. How does sky propagation differ from line-of-sight propagation?

Switching

1. What is the need for switching and define a switch.
2. List the three traditional switching methods. What are the most common today?
3. What are the two approaches to packet-switching?
4. Compare and contrast a circuit-switched network and a packet-switched network.
5. What is the role of the address field in a packet traveling through a datagram network?
6. What is the role of the address field in a packet traveling through a virtual-circuit network?
7. What is TSI and its role in a time-division switching?
8. List four major components of a packet switch and their functions.



INTRODUCTION TO DATA LINK LAYER

Unit Structure

- 7.0 Objectives
- 7.1 Introduction
- 7.2 Data Link Layer addressing
- 7.3 Data Link Layer Design issues
- 7.4 Error Detection and Correction
- 7.5 Block Coding
- 7.6 Linear Block Coding
- 7.7 Cyclic Codes
- 7.8 Checksum
- 7.9 Summary
- 7.10 Reference for further reading
- 7.11 Model Questions

7.0 OBJECTIVES:

This chapter would make you to understand the following concepts:

- Concept of Link layer addressing and Design issues of Data Link layer.
- Concept of Error Detection and Correction.
- Block Coding, Linear Block Coding and its different methods or techniques.
- Cyclic codes, CRC code and CRC Polynomial.
- Checksum and Internet Checksum.

7.1 INTRODUCTION

The data link layer transforms the physical layer, a raw transmission facility, to a link responsible for node-to-node (hop-to-hop) communication. Specific responsibilities of the data link layer include framing, addressing, flow control, error control, and media access control. The data link layer divides the stream of bits received from the network layer into manageable data units called frames. The data link layer adds a header to the frame to define the addresses of the sender and receiver of the frame. If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer

imposes a flow control mechanism to avoid overwhelming the receiver. The data link layer also adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged, duplicate, or lost frames.

When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

7.2 DATA LINK LAYER ADDRESSING

As data link layer adds a header to the frame to define the addresses of the sender and receiver of the frame. The address used by data link layer to define sender as well as receiver's address called as Physical address or Media Access Control (MAC) address. Physical address is uniquely assigned to an individual device by its manufacturer and it is **48 bit (6 bytes)** written as **12 hexadecimal digits**; every byte (**2 hexadecimal digits**) is separated by a colon as shown in Figure 7.1.



Figure 7.1: Physical address or MAC address used by Data Link layer

7.3 DATA LINK LAYER DESIGN ISSUES

Following are the some design issues of the Data Link layer

1. **Services provided to the network layer:** The data link layer act as a service interface to the network layer. The principle service is transferring data from network layer on sending machine to the network layer on destination machine. This transfer also takes place via DLL (Dynamic Link Library).
2. **Frame synchronization:** The source machine sends data in the form of blocks (data units of manageable size) called frames to the destination machine. The starting and ending of each frame should be identified so that the frame can be recognized by the destination machine.
3. **Flow control:** Flow control is done to prevent the flow of data frame at the receiver end. The source machine must not send data frames at a rate faster than the capacity of destination machine to accept them.
4. **Error control:** Error control is done to prevent duplication of frames. The errors introduced during transmission from source to destination machines must be detected and corrected at the destination machine.

Networks must be able to transfer data from one device to another with acceptable accuracy. For most applications, a system must guarantee that the data received are identical to the data transmitted. Any time data are transmitted from one node to the next, they can become corrupted in passage. Many factors can alter one or more bits of a message. Some applications require a mechanism for detecting and correcting errors.

Types of Errors

Single-Bit Error

Diagram illustrating a bit flip error. A 'Sent' byte (0 0 0 0 1 0 1 0) is transformed into a 'Received' byte (0 0 0 0 0 1 0 1). An arrow points from the 5th bit (1) in the sent byte to the 5th bit (0) in the received byte, labeled '0 changed to 1'.

Burst Error

Length of burst error (8 bits)

Sent

0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Corrupted bits

0	1	0	1	1	1	0	1	0	1	1	0	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Received

Redundancy

133

Detection versus Correction

The correction of errors is more difficult than the detection. In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors. A single-bit error is the same for us as a burst error.

In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message. The number of the errors and the size of the message are important factors. If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 possibilities. You can imagine the Receiver's difficulty in finding 10 errors in a data unit of 1000 bits.

Forward Error Correction versus Retransmission

There are two main methods of error correction.

Forward error correction is the process in which the receiver tries to guess the message by using redundant bits. This is possible, as we see later, if the number of errors is small.

Correction by retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Resending is repeated until a message arrives that the receiver believes is error-free (usually, not all errors can be detected).

Coding

Redundancy is achieved through various coding schemes. The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits. The receiver checks the relationships between the two sets of bits to detect or correct the errors. The ratio of redundant bits to the data bits and the robustness of the process are important factors in any coding scheme. Figure 7.4 shows the general idea of coding.

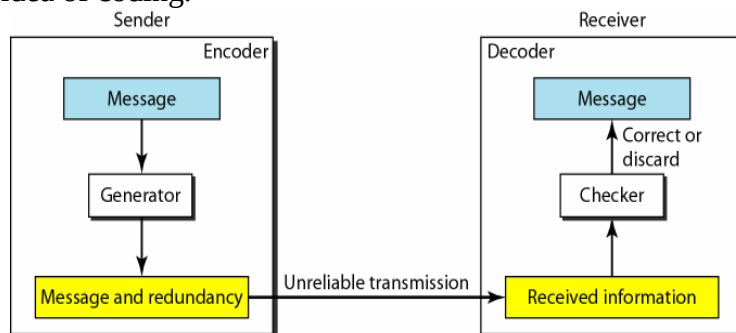


Figure 7.4: Structure of Encoder and Decoder

Modular Arithmetic

In modular arithmetic, we use only a limited range of integers. We define an upper limit, called a modulus N . We then use only the integers 0

to $N - 1$, inclusive. This is called as *modulo-N* arithmetic. For example, if the modulus is 12, we use only the integers 0 to 11, inclusive.

Modulo-2 Arithmetic

Our main interest is in modulo-2 arithmetic. In this arithmetic, the modulus N is 2. We can use only 0 and 1. Operations (addition and subtraction) in this arithmetic are very simple.

In this arithmetic we use the XOR (exclusive OR) operation for both addition and subtraction. The result of an XOR operation is 0 if two bits are the same; the result is 1 if two bits are different. Figure 7.5 shows this operation.

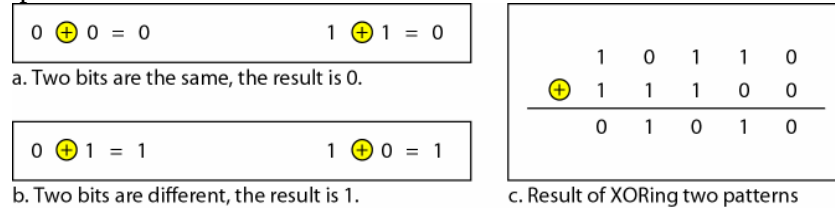


Figure 7.5: XORing of 2 single bits or 2 words

7.5 BLOCK CODING

In block coding, we divide our message into blocks, each of k bits, called data words. We add r redundant bits to each block to make the length $n = k + r$. The resulting n -bit blocks are called code words. For the moment, it is important to know that we have a set of data words, each of size k , and a set of code words, each of size of n . With k bits, we can create a combination of 2^k data words; with n bits, we can create a combination of 2^n codewords.

Since $n > k$, the number of possible code words is larger than the number of possible data words.

The block coding process is one-to-one; the same data word is always encoded as the same codeword. This means that we have $2^n - 2^k$ codewords that are not used. We call these code words invalid or illegal. Figure 7.6 shows the situation.

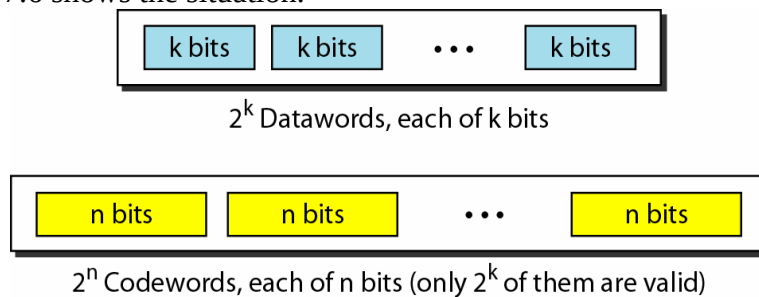


Figure 7.6: Datawords and Codewords in Block Coding

Example 7.1:

The **4B/5B** block coding is a good example of this type of coding. In this coding scheme, $k = 4$ and $n = 5$. As we see, we have $2^k = 16$ data words and $2^n = 32$ code words. We have 16 out of 32 code words are used for message transfer and the rest are either used for other purposes or unused.

Error Detection

How can errors be detected by using block coding? If the following two conditions are met, the receiver can detect a change in the original codeword.

1. The receiver has (or can find) a list of valid code words.
2. The original codeword has changed to an invalid one.

Figure 7.7 shows the role of block coding in error detection.

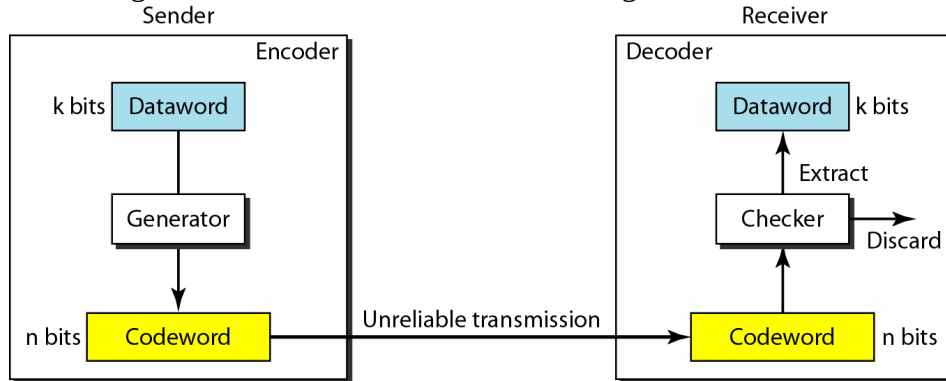


Figure 7.7: Process of Error detection in Block coding

Example 7.2:

Let us assume that $k = 2$ and $n = 3$. Table 7.1 shows the list of data words and code words.

<i>Datawords</i>	<i>Codewords</i>
00	000
01	011
10	101
11	110

Table 7.1: Code for Error detection

Assume the sender encodes the data word 01 as 011 and sends it to the receiver. Consider the following cases:

1. The receiver receives 011. It is a valid codeword. The receiver extracts the data word 01 from it.
2. The codeword is corrupted during transmission, and 111 is received (the leftmost bit is corrupted). This is not a valid codeword and is discarded.
3. The codeword is corrupted during transmission, and 000 is received (the right two bits are corrupted). This is a valid

codeword. The receiver incorrectly extracts the dataword 00. Two corrupted bits have made the error undetectable.

Note: An error-detecting code can detect only the types of errors for which it is designed; other types of errors may remain undetected.

Error Correction

As we said before, error correction is much more difficult than error detection. In error detection, the receiver needs to know only that the received codeword is invalid; in error correction the receiver needs to find (or guess) the original codeword sent. We can say that we need more redundant bits for error correction than for error detection.

Figure 7.8 shows the role of block coding in error correction. We can see that the idea is the same as error detection but the checker functions are much more complex.

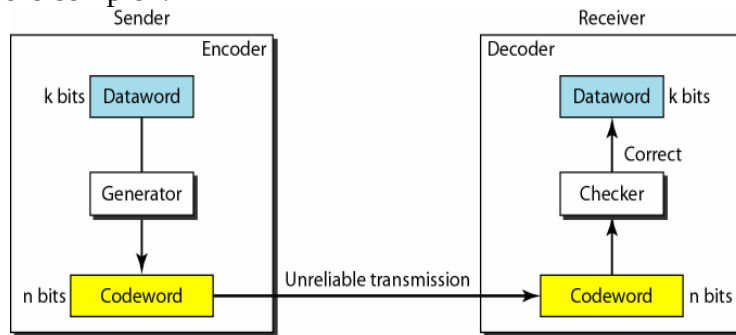


Figure 7.8: Structure of Encoder and Decoder in Error correction

Example 7.3:

Let us add more redundant bits to Example 7.2 to see if the receiver can correct an error without knowing what was actually sent. We add 3 redundant bits to the 2-bit data word to make 5-bit codewords. Table 7.2 shows the data words and code words.

<i>Dataword</i>	<i>Codeword</i>
00	00000
01	01011
10	10101
11	11110

Table 7.2: Code for Error correction

Assume the data word is 01.

The sender consults the table (or uses an algorithm) to create the codeword 01011. The codeword is corrupted during transmission, and 01001 is received (error in the second bit from the right). First, the receiver finds that the received codeword is not in the table.

This means an error has occurred. (Detection must come before correction.) The receiver, assuming that there is only 1 bit corrupted, uses the following strategy to guess the correct data word.

1. Comparing the received codeword with the first codeword in the table (01001 versus 00000), the receiver decides that the first codeword is not the one that was sent because there are two different bits.
2. By the same reasoning, the original codeword cannot be the third or fourth one in the table.
3. The original codeword must be the second one in the table because this is the only one that differs from the received codeword by 1 bit. The receiver replaces 01001 with 01011 and consults the table to find the data word 01.

Hamming Distance

One of the central concepts in coding for error control is the idea of the Hamming distance. The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits. We show the Hamming distance between two words x and y as $d(x, y)$.

The Hamming distance can easily be found if we apply the XOR operation ($\oplus$) on the two words and count the number of 1s in the result. Note that the Hamming distance is a value greater than zero.

Example 7.4:

Let us find the Hamming distance between two pairs of words.

The Hamming distance $d(000, 011)$ is 2 because $000 \oplus 011$ is 011 (two 1s).

The Hamming distance $d(10101, 11110)$ is 3 because $10101 \oplus 11110$ is 01011 (three 1s).

Minimum Hamming Distance

Although the concept of the Hamming distance is the central point in dealing with error detection and correction codes, the measurement that is used for designing a code is the minimum Hamming distance. In a set of words, the minimum Hamming distance is the smallest Hamming distance between all possible pairs. We use $d_{\min}$ to define the minimum Hamming distance in a coding scheme. To find this value, we find the Hamming distances between all words and select the smallest one.

Example 7.5:

Find the minimum Hamming distance of the coding scheme in Table 7.1.

Solution:

We first find all Hamming distances.

$$d(000, 011) = 2 \quad d(000, 101) = 2 \quad d(000, 110) = 2$$

$$d(011, 101) = 2 \quad d(011, 110) = 2 \quad d(101, 110) = 2$$

The $d_{\min}$ in this case is 2.

Example 7.6:

Find the minimum Hamming distance of the coding scheme in Table 7.2.

Solution:

We first find all the Hamming distances.

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

$$d(01011, 10101) = 4 \quad d(01011, 11110) = 3 \quad d(10101, 11110) = 3$$

The $d_{\min}$ in this case is 3.

We need to mention that any coding scheme needs to have at least three parameters: the codeword size n , the data word size k , and the minimum Hamming distance $d_{\min}$. A coding scheme C is written as $C(n, k)$ with a separate expression for $d_{\min}$. For example, we can call our Table 7.1 coding scheme is $C(3, 2)$ with $d_{\min} = 2$ and our Table 7.2 coding scheme is $C(5, 2)$ with $d_{\min} = 3$.

Hamming Distance and Error

Before we explore the criteria for error detection or correction, let us discuss the relationship between the Hamming distance and errors occurring during transmission. When a codeword is corrupted during transmission, the Hamming distance between the sent and received code words is the number of bits affected by the error. In other words, the Hamming distance between the received codeword and the sent codeword is the number of bits that are corrupted during transmission. For example, if the codeword 00000 is sent and 01101 is received, 3 bits are in error and the Hamming distance between the two is $d(00000, 01101) = 3$.

Minimum Distance for Error Detection

Now let us find the minimum Hamming distance in a code if we want to be able to detect up to s errors. If s errors occur during transmission, the Hamming distance between the sent codeword and received codeword is s . If our code is to detect up to s errors, the minimum distance between the valid codes must be $s + 1$, so that the received codeword does not match a valid codeword. In other words, if the minimum distance between all valid code words is $s + 1$, the received codeword cannot be erroneously mistaken for another codeword. The distances are not enough ($s + 1$) for the receiver to accept it as valid. The error will be detected. We need to clarify a point here: Although a code with $d_{\min} = s + 1$ may be able to detect more than s errors in some special cases, only s or fewer errors are guaranteed to be detected.

Example 7.7:

The minimum Hamming distance for Table 7.1 coding scheme is 2. This code guarantees detection of only a single error. For example, if the third codeword (101) is sent and one error occurs, the received codeword does not match any valid codeword. If two errors occur, however, the received codeword may match a valid codeword and the errors are not detected.

Example 7.8:

For Table 7.2 coding scheme has $d_{\min} = 3$. This code can detect up to two errors. Again, we see that when any of the valid code words is sent, two errors create a codeword which is not in the table of valid code words. The receiver cannot be fooled. However, some combinations of three errors change a valid codeword to another valid codeword. The receiver accepts the received codeword and the errors are undetected.

7.6 LINEAR BLOCK CODING

Almost all block codes used today belong to a subset called linear block codes. The use of nonlinear block codes for error detection and correction is not as widespread because their structure makes theoretical analysis and implementation difficult. We therefore concentrate on linear block codes.

A linear block code is a code in which the exclusive OR (addition modulo-2) of two valid code words creates another valid codeword.

Example 7.9:

Let us see if the two codes we defined in Table 7.1 and Table 7.2 belong to the class of linear block codes.

1. The scheme in Table 7.1 is a linear block code because the result of XORing any codeword with any other codeword is a valid codeword. For example, the XORing of the second and third code words creates the fourth one.
2. The scheme in Table 7.2 is also a linear block code. We can create all four code words by XORing two other code words.

Minimum Distance for Linear Block Codes

It is simple to find the minimum Hamming distance for a linear block code. The minimum Hamming distance is the number of 1s in the nonzero valid codeword with the smallest number of 1s.

Example 7.10:

In our first code - Table 7.1, the numbers of 1s in the nonzero code words are 2, 2, and 2. So the minimum Hamming distance is $d_{\min} = 2$. In our second code - Table 7.2, the numbers of 1s in the nonzero code words are 3, 3, and 4. So in this code we have $d_{\min} = 3$.

Let us now we can study some Linear block codes

Simple Parity Check code

The most familiar error-detecting code is the simple parity-check code. In this code, a k -bit data word is changed to an n -bit codeword where $n = k + 1$. The extra bit, called the parity bit, is selected to make the total number of 1s in the codeword even.

The minimum Hamming distance for this category is $d_{\min} = 2$, which means that the code is a single-bit error-detecting code; it cannot correct any error.

Our first code - Table 7.1 is a parity-check code with $k = 2$ and $n = 3$. The code in Table 7.3 is also a parity-check code with $k = 4$ and $n = 5$. Figure 7.9 shows a possible structure of an encoder (at the sender) and a decoder (at the receiver).

<i>Datawords</i>	<i>Codewords</i>	<i>Datawords</i>	<i>Codewords</i>
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110

Table 7.3: Simple Parity-check code $C(5,4)$

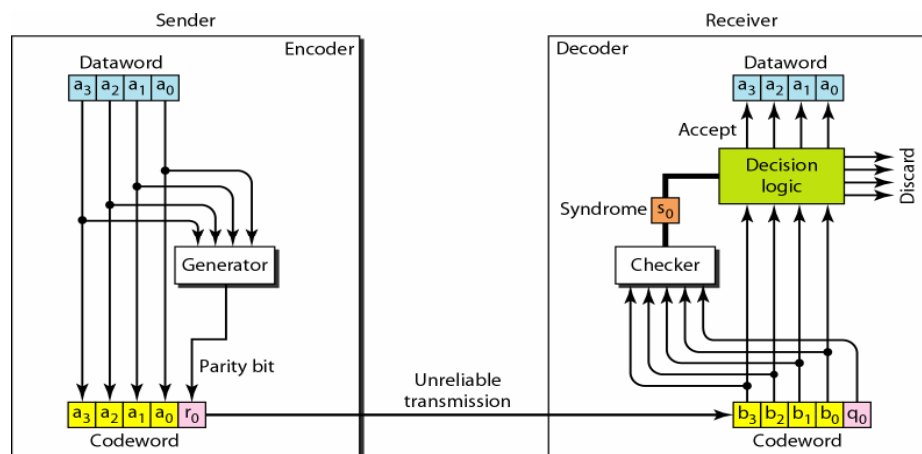


Figure 7.9: Encoder and Decoder for Simple Parity-check code

The encoder uses a generator that takes a copy of a 4-bit data word (a_0, a_1, a_2 and a_3) and generates a parity bit r_0 . The data word bits and the

parity bit create the 5-bit codeword. The parity bit that is added makes the number of 1s in the codeword even.

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

This is normally done by adding the 4 bits of the data word (modulo-2); the result is the parity bit. In other words, if the number of 1s is even, the result is 0; if the number of 1s is odd, the result is 1. In both cases, the total number of 1s in the codeword is even.

The sender sends the codeword which may be corrupted during transmission. The receiver receives a 5-bit word. The checker at the receiver does the same thing as the generator in the sender with one exception: The addition is done over all 5 bits. The result, which is called the syndrome, is just 1 bit. The syndrome is 0 when the number of 1s in the received codeword is even; otherwise, it is 1.

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

The syndrome is passed to the decision logic analyzer. If the syndrome is 0, there is no error in the received codeword; the data portion of the received codeword is accepted as the data word; if the syndrome is 1, the data portion of the received codeword is discarded. The data word is not created.

Example 7.11:

Let us look at some transmission scenarios. Assume the sender sends the data word 1011. The codeword created from this data word is 10111, which is sent to the receiver. We examine five cases:

1. No error occurs; the received codeword is 10111. The syndrome is 0. The data word 1011 is created.
2. One single-bit error changes a_1 . The received codeword is 10011. The syndrome is 1. No data word is created.
3. One single-bit error changes r_0 . The received codeword is 10110. The syndrome is 1. No data word is created. Note that although none of the data word bits are corrupted, no data word is created because the code is not sophisticated enough to show the position of the corrupted bit.
4. An error changes r_0 and a second error changes a_3 . The received codeword is 00110. The syndrome is 0. The data word 0011 is created at the receiver. Note that here the data word is wrongly created due to the syndrome value. The simple parity-check decoder cannot detect an even number of errors. The errors cancel each other out and give the syndrome a value of 0.
5. Three bits- a_3 , a_2 and a_1 are changed by errors. The received codeword is 01011. The syndrome is 1. The data word is not created. This shows that the simple parity check, guaranteed to detect one single error, can also find any odd number of errors.

Note: A simple parity-check code can detect an odd number of errors.

Hamming Codes

Now let us discuss a category of error-correcting codes called Hamming codes. These codes were originally designed with $d_{min} = 3$, which means that they can detect up to two errors or correct one single error. Although there are some Hamming codes that can correct more than one error, our discussion focuses on the single-bit error-correcting code.

First let us find the relationship between n and k in a Hamming code. We need to choose an integer $m \geq 3$. The values of n and k are then calculated from m as $n = 2^m - 1$ and $k = n - m$. The number of check bits $r = m$.

For example, if $m = 3$, then $n = 7$ and $k = 4$. This is a Hamming code $C(7, 4)$ with $d_{min} = 3$.

Table 7.4 shows the datawords and codewords for this code.

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

Table 7.4: Hamming code $C(7, 4)$

Figure 7.10 shows the structure of the encoder and decoder for this example.

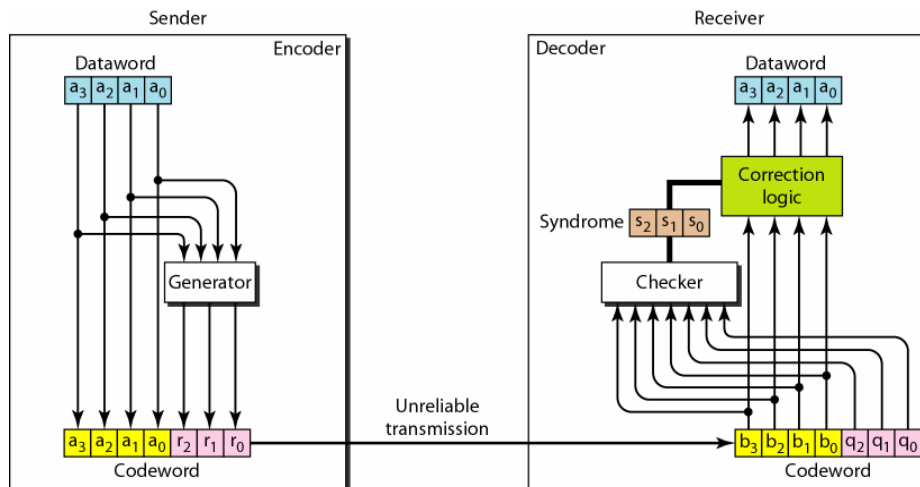


Figure 7.10: Encoder and Decoder for Hamming code

A copy of a 4-bit data word is fed into the generator that creates three parity checks r_0 , r_1 and r_2 as shown below:

$$\begin{aligned} r_0 &= a_2 + a_1 + a_0 \quad (\text{modulo-2}) \\ r_1 &= a_3 + a_2 + a_1 \quad (\text{modulo-2}) \\ r_2 &= a_1 + a_0 + a_3 \quad (\text{modulo-2}) \end{aligned}$$

The checker in the decoder creates a 3-bit syndrome – s_2 , s_1 and s_0 in which each bit is the parity check for 4 out of the 7 bits in the received codeword:

$$\begin{aligned} s_0 &= b_2 + b_1 + b_0 + q_0 \quad (\text{modulo-2}) \\ s_1 &= b_3 + b_2 + b_1 + q_1 \quad (\text{modulo-2}) \\ s_2 &= b_1 + b_0 + b_3 + q_2 \quad (\text{modulo-2}) \end{aligned}$$

The 3-bit syndrome creates eight different bit patterns (000 to 111) that can represent eight different conditions. These conditions define a lack of error or an error in 1 of the 7 bits of the received codeword, as shown in Table 7.5.

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

Table 7.5: Logical decision made by the correction logic analyzer of Decoder

Example 7.12:

Let us trace the path of three data words from the sender to the destination:

1. The data word 0100 becomes the codeword 0100011. The codeword 0100011 is received. The syndrome is 000 (no error), the final data word is 0100.
2. The data word 0111 becomes the codeword 0111001. The codeword 0011001 is received. The syndrome is 011. According to Table 7.5, b_2 is in error. After flipping b_2 (changing the 1 to 0), the final data word is 0111.
3. The data word 1101 becomes the codeword 1101000. The codeword 0001000 is received (two errors). The syndrome is 101, which means that according to Table 7.5, b_0 is in error. After flipping b_0 , we get 0000, the wrong data word. This shows that our code cannot correct two errors.

Example 7.13:

We need a data word of at least 7 bits. Calculate values of k and n that satisfy this requirement.

Solution:

We need to make $k = n - m$ greater than or equal to 7, or $2^m - 1 - m \geq 7$.

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

2. If we set $m=4$, then $n=2^4-1=15$ and $k=15-4=11$, which satisfies the condition. So the code is $C(15, 11)$.

7.7 CYCLIC CODES

Cyclic codes are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword. For example, if 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword.

In this case, if we call the bits in the first word a_0 to a_6 and the bits in the second word b_0 to b_6 , we can shift the bits by using the following:

$$b_1 = a_0, b_2 = a_1, b_3 = a_2, b_4 = a_3, b_5 = a_4, b_6 = a_5, b_0 = a_6$$

In the rightmost equation, the last bit of the first word is wrapped around and becomes the first bit of the second word.

Cyclic Redundancy Check

A category of cyclic codes called the Cyclic Redundancy Check (CRC) that is used in networks such as LANs and WANs.

Table 7.6 shows an example of a CRC code. We can see both the linear and cyclic properties of this code.

Figure 7.11 shows one possible design for the encoder and decoder.

<i>Dataword</i>	<i>Codeword</i>	<i>Dataword</i>	<i>Codeword</i>
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

Table 7.6: CRC code with $C(7, 4)$

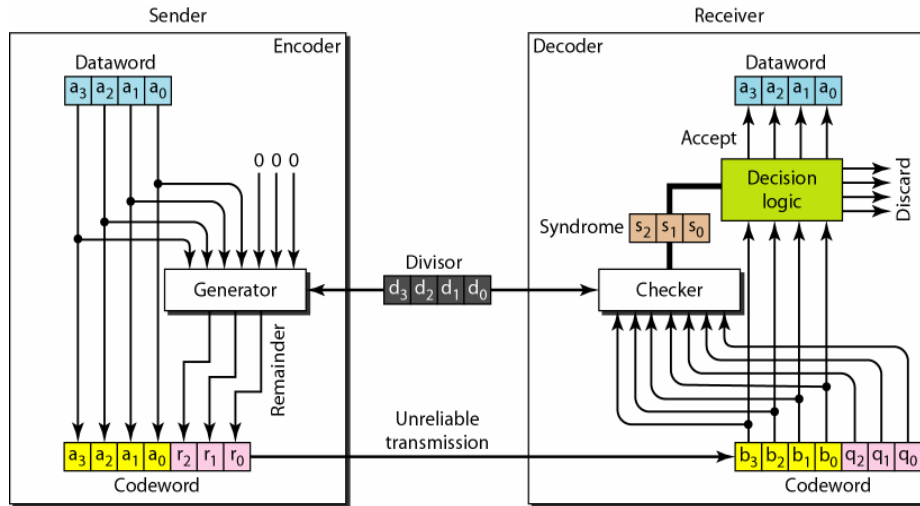


Figure 7.11: Encoder and Decoder for CRC code

In the encoder, the data word has k bits (4 here); the codeword has n bits (7 here). The size of the data word is augmented by adding $n - k$ (3 here) 0s to the right-hand side of the word. The n -bit result is fed into the generator. The generator uses a divisor of size $n - k + 1$ (4 here), predefined and agreed upon. The generator divides the augmented data word by the divisor (modulo-2 division). The quotient of the division is discarded; the remainder (r_2, r_1, r_0) is appended to the data word to create the codeword.

The decoder receives the possibly corrupted codeword. A copy of all n bits is fed to the checker which is a replica of the generator. The remainder produced by the checker is a syndrome of $n - k$ (3 here) bits, which is fed to the decision logic analyzer. The analyzer has a simple function. If the syndrome bits are all as 0s, the 4 leftmost bits of the codeword are accepted as the data word (interpreted as no error); otherwise, the 4 bits are discarded (error).

Encoder

Let us take a closer look at the encoder. The encoder takes the data word and augments it with $n - k$ number of 0s. It then divides the augmented data word by the divisor, as shown in Figure 7.12.

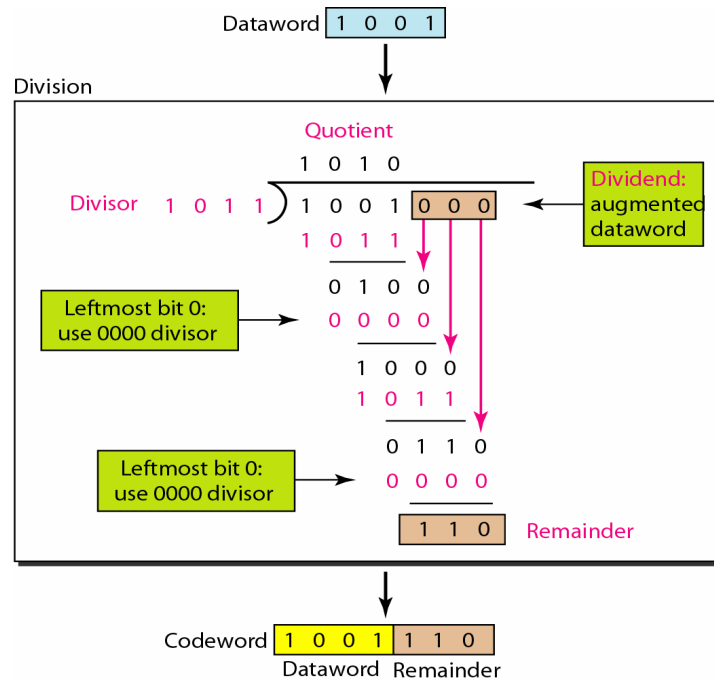


Figure 7.12: Division in the CRC Encoder

Decoder

The codeword can change during transmission. The decoder does the same division process as the encoder. The remainder of the division is the syndrome. If the syndrome is all 0s, there is no error; the data word is separated from the received codeword and accepted. Otherwise, everything is discarded. Figure 7.13 shows two cases: The left-hand figure shows the value of syndrome when no error has occurred; the syndrome is 000. The right-hand part of the figure shows the case in which there is one single error. The syndrome is not all 0s (it is 011).

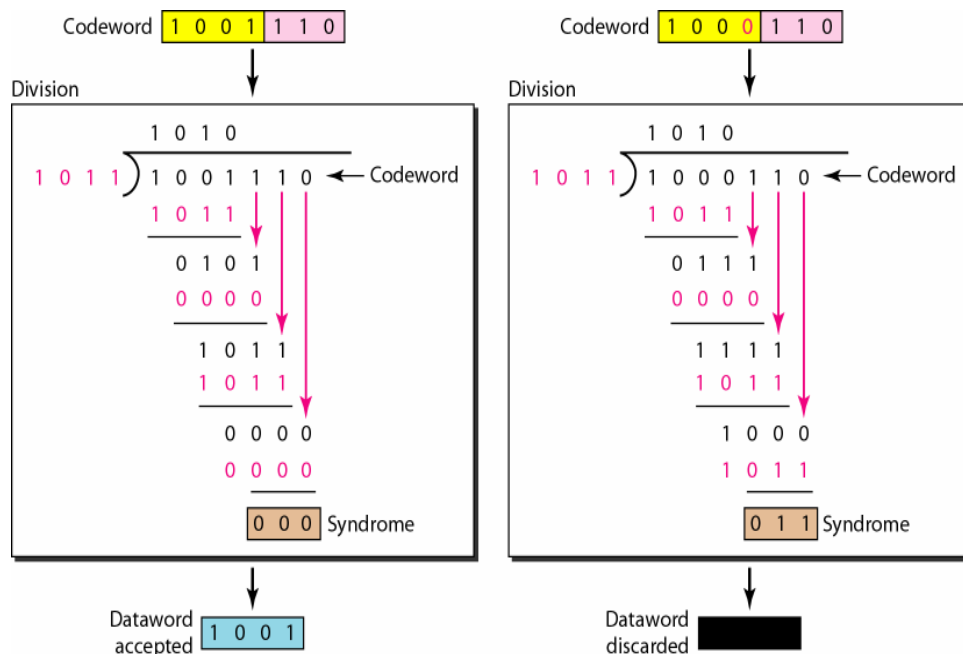


Figure 7.13: Division in the CRC Decoder for two cases

Polynomials

A better way to understand cyclic codes and how they can be analyzed is to represent them as polynomials. A pattern of **0**s and **1**s can be represented as a **polynomial** with coefficients of **0** and **1**. The power of each term shows the position of the bit; the coefficient shows the value of the bit. Figure 7.14 shows a binary pattern and its polynomial representation. In Figure 7.14a we show how to translate a binary pattern to a polynomial; in Figure 7.14b we show how the polynomial can be shortened by removing all terms with zero coefficients.

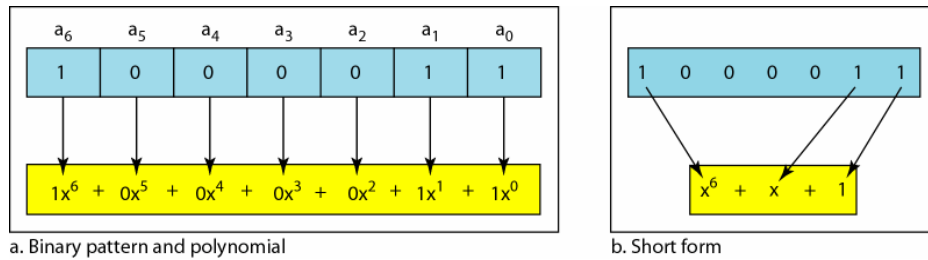


Figure 7.14: A Polynomial to represent a binary word

Degree of a Polynomial

The degree of a polynomial is the highest power in the polynomial. For example, the degree of the polynomial $x^6 + x + 1$ is 6. Note that the degree of a polynomial is 1 less than the number of bits in the pattern. The bit pattern in this case has 7 bits.

Adding and Subtracting Polynomials

Adding and subtracting polynomials in mathematics are done by adding or subtracting the coefficients of terms with the same power. In our case, the coefficients are only **0** and **1**, and adding is in modulo-2. This has two consequences. First, addition and subtraction are the same. Second, adding or subtracting is done by combining terms and deleting pairs of identical terms. For example, adding $x^5 + x^4 + x^2$ and $x^6 + x^4 + x^2$ gives just $x^6 + x^5$. The terms x^4 and x^2 are deleted. However, note that if we add, for example, three polynomials and we get x^2 three times, we delete a pair of them and keep the third.

Multiplying or Dividing Terms

In this arithmetic, multiplying a term by another term is very simple; we just add the powers. For example, $x^3 x^4$ is x^7 . For dividing, we just subtract the power of the second term from the power of the first. For example, x^5/x^2 is x^3 .

Dividing One Polynomial by Another

Division of polynomials is conceptually the same as the binary division we discussed for an encoder. We divide the first term of the dividend by the first term of the divisor to get the first term of the quotient. We multiply the term in the quotient by the divisor and subtract the result

from the dividend. We repeat the process until the dividend degree is less than the divisor degree.

Shifting

A binary pattern is often shifted a number of bits to the right or left. Shifting to the left means adding extra 0s as rightmost bits; shifting to the right means deleting some rightmost bits. Shifting to the left is accomplished by multiplying each term of the polynomial by x^m , where m is the number of shifted bits; shifting to the right is accomplished by dividing each term of the polynomial by x^m . The following shows shifting to the left and to the right. Note that we do not have negative powers in the polynomial representation.

Shifting left 3 bits: **10011** becomes **10011000** $x^4 + x + 1$ becomes $x^7 + x^4 + x^3$

Shifting right 3 bits: **10011** becomes **10** $x^4 + x + 1$ becomes x

Cyclic Code Encoder Using Polynomials

Now we show the creation of a code word from a data word. The data word **1001** is represented as $x^3 + 1$. The divisor **1011** is represented as $x^3 + x + 1$. To find the augmented data word, we have left-shifted the data word 3 bits (multiplying by x^3). The result is $x^6 + x^3$.

Division is straightforward.

We divide the first term of the dividend, x^6 , by the first term of the divisor, x^3 . The first term of the quotient is then x^6/x^3 , or x^3 . Then we multiply x^3 by the divisor and subtract (according to our previous definition of subtraction) the result from the dividend. The result is x^4 , with a degree greater than the divisor's degree; we continue to divide until the degree of the remainder is less than the degree of the divisor. CRC division using Polynomials is shown in the Figure 7.15.

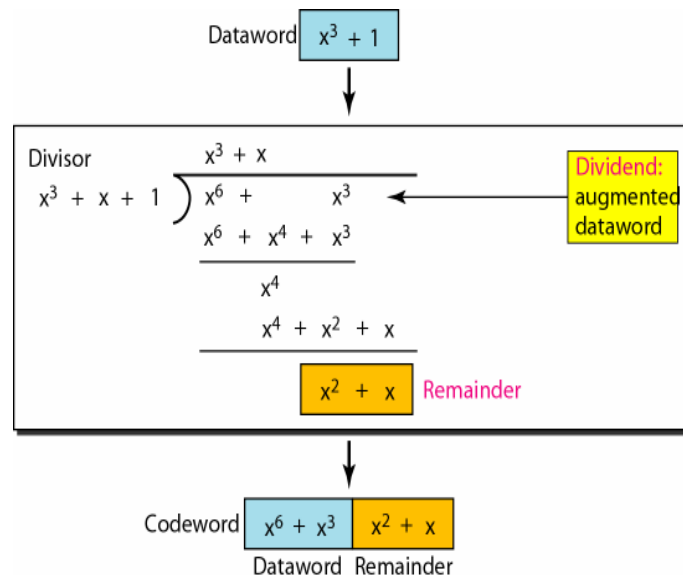


Figure 7.15: CRC division using Polynomials

It can be seen that the polynomial representation can easily simplify the operation of division in this case, because the two steps involving all-0s divisors are not needed here. (Of course, one could argue that the all-0s divisor step can also be eliminated in binary division). In a polynomial representation, the divisor is normally referred to as the generator polynomial $t(x)$.

We can summarize the criteria for a good polynomial generator:

A good polynomial generator needs to have the following characteristics:

1. It should have at least two terms.
2. The coefficient of the term x^0 should be 1.
3. It should not divide $x^t + 1$, for t between 2 and $n - 1$.
4. It should have the factor $x + 1$.

Standard Polynomials

Some standard polynomials used by popular protocols for CRC generation are shown in Table 7.7.

<i>Name</i>	<i>Polynomial</i>	<i>Application</i>
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLCL
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs

Table 7.7: Standard Polynomials

Advantages of Cyclic Codes

Cyclic codes have a very good performance in detecting single-bit errors, double errors, an odd number of errors, and burst errors. They can easily be implemented in hardware and software. They are especially fast when implemented in hardware. This has made cyclic codes a good candidate for many networks.

7.8 CHECKSUM

The last error detection method we discuss here is called the checksum. The checksum is used in the Internet by several protocols although not at the data link layer. However, we briefly discuss it here to complete our discussion on error checking. Like linear and cyclic codes, the checksum is based on the concept of redundancy.

Concept

The concept of the checksum is not difficult. Let us illustrate it with a few examples.

Example 7.14:

Suppose our data is a list of five 4-bit numbers that we want to send to a destination. In addition to sending these numbers, we send the sum of the numbers. For example, if the set of numbers is (7, 11, 12, 0, 6), we send (7, 11, 12, 0, 6, 36), where 36 is the sum of the original numbers. The receiver adds the five numbers and compares the result with the sum. If the two are the same, the receiver assumes no error, accepts the five numbers, and discards the sum. Otherwise, there is an error somewhere and the data are not accepted.

Example 7.15:

We can make the job of the receiver easier if we send the negative (complement) of the sum, called as the **checksum**. In this case, we send (7, 11, 12, 0, 6, -36). The receiver can add all the numbers received (including the checksum). If the result is 0, it assumes no error; otherwise, there is an error.

One's Complement

The previous example has one major drawback. All of our data can be written as a 4-bit word (they are less than 15) except for the checksum. One solution is to use one's complement arithmetic. In this arithmetic, we can represent unsigned numbers between 0 and $2^n - 1$ using only n bits. If the number has more than n bits, the extra leftmost bits need to be added to the n rightmost bits (wrapping). In one's complement arithmetic, a negative number can be represented by inverting all bits (changing a 0 to a 1 and a 1 to a 0). This is the same as subtracting the number from $2^n - 1$.

Example 7.16:

How can we represent the number 21 in one's complement arithmetic using only four bits?

Solution:

The number 21 in binary is **10101** (it needs five bits). We can wrap the leftmost bit and add it to the four rightmost bits. We have **(0101 + 1) = 0110** or **6**.

Example 7.17:

How can we represent the number -6 in one's complement arithmetic using only four bits?

Solution:

In one's complement arithmetic, the negative or complement of a number is found by inverting all bits. Positive 6 is **0110**; negative 6 is **1001**. If we consider only unsigned numbers, this is **9**. In other words, the complement of 6 is 9. Another way to find the complement of a number in one's complement arithmetic is to subtract the number from $2^n - 1$ (**15 - 6 = 9** in this case).

Example 7.18:

Let us redo Exercise 7.15 using one's complement arithmetic. Figure 7.16 shows the process at the sender and at the receiver. The sender initializes the checksum to **0** and adds all data items and the checksum. The result is **36**. However, **36** cannot be expressed in **4** bits. The extra two bits are wrapped and added with the sum to create the wrapped sum value **6**. In the figure, we have shown the details in binary. The sum is then complemented, resulting in the checksum value **9** ($15 - 6 = 9$). The sender now sends six data items to the receiver including the checksum **9**. The receiver follows the same procedure as the sender. It adds all data items (including the checksum); the result is **45**. The sum is wrapped and becomes **15**. The wrapped sum is complemented and becomes **0**. Since the value of the checksum is **0**, this means that the data is not corrupted. The receiver drops the checksum and keeps the other data items. If the checksum is not zero, the entire packet is dropped.

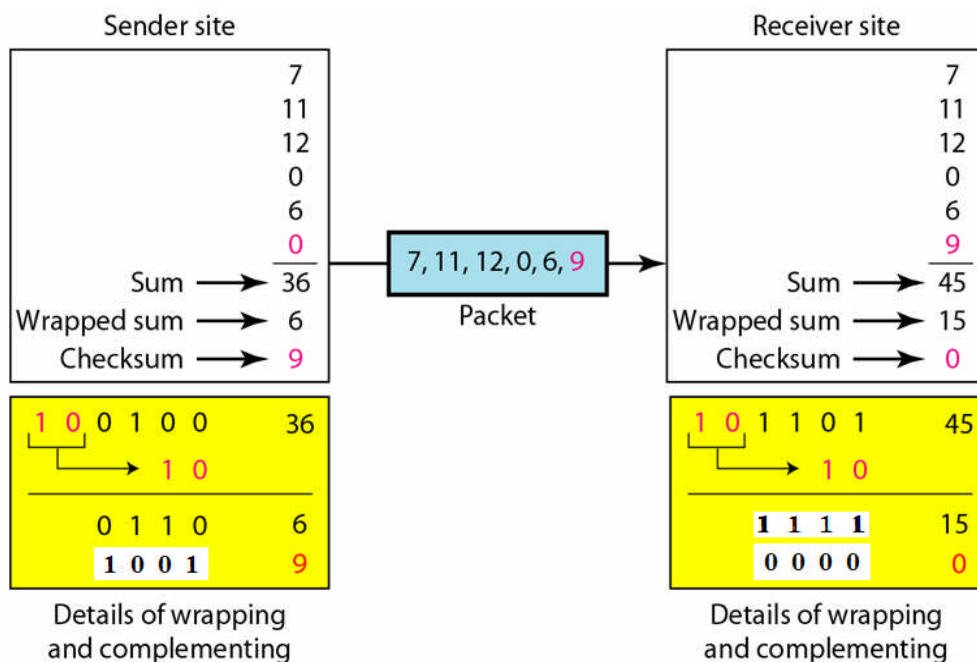


Figure 7.16: Example 7.18

Internet Checksum

Traditionally, the Internet has been using a 16-bit checksum. The sender calculates the checksum by following these steps.

Sender site:

1. The message is divided into 16-bit words.
2. The value of the checksum word is set to 0.
3. All words including the checksum are added using one's complement addition.
4. The sum is complemented and becomes the checksum.
5. The checksum is sent with the data.

The receiver uses the following steps for error detection.

Receiver site:

1. The message (including checksum) is divided into 16-bit words.
2. All words are added using one's complement addition.
3. The sum is complemented and becomes the new checksum.
4. If the value of checksum is 0, the message is accepted; otherwise, it is rejected.

The nature of the checksum (treating words as numbers and adding and complementing them) is well-suited for software implementation. Short programs can be written to calculate the checksum at the receiver site or to check the validity of the message at the receiver site.

7.9 SUMMARY

- Data can be corrupted during transmission. Some applications require that errors be detected and corrected.
- In a single-bit error, only one bit in the data unit has changed. A burst error means that two or more bits in the data unit have changed.
- To detect or correct errors, we need to send extra (redundant) bits with data.
- There are two main methods of error correction: forward error correction and correction by retransmission.
- In coding, we need to use modulo-2 arithmetic. Operations in this arithmetic are very simple; addition and subtraction give the same results. We use the XOR (exclusive OR) operation for both addition and subtraction.
- In block coding, we divide our message into blocks, each of k bits, called data words. We add r redundant bits to each block to make the length $n = k + r$. The resulting n -bit blocks are called code words.
- In block coding, errors be detected by using the following two conditions:
 - The receiver has (or can find) a list of valid code words.
 - The original codeword has changed to an invalid one.
- The Hamming distance between two words is the number of differences between corresponding bits. The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words.
- To guarantee the detection of up to s errors in all cases, the minimum Hamming distance in a block code must be $d_{\min} = s + 1$.
- In a linear block code, the exclusive OR (XOR) of any two valid code words creates another valid codeword.

- A simple parity-check code is a single-bit error-detecting code in which $n = k + 1$ with $d_{\min} = 2$. A simple parity-check code can detect an odd number of errors.
- All Hamming codes discussed in this book have $d_{\min} = 3$. The relationship between m and n in these codes is $n = 2^m - 1$.
- Cyclic codes are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.
- A category of cyclic codes called the cyclic redundancy check (CRC) is used in networks such as LANs and WANs.
- A pattern of 0s and 1s can be represented as a polynomial with coefficients of 0 and 1.
- Traditionally, the Internet has been using a 16-bit checksum, which uses *one's complement* arithmetic. In this arithmetic, we can represent unsigned numbers between 0 and $2^n - 1$ using only n bits.

7.10 REFERENCE FOR FURTHER READING

For more details about topics discussed in this chapter, we recommend the following books.

1. *Data Communication and Networking* by Behrouz A. Forouzan, McGraw-Hill, 2007.
2. *Coding and Information Theory* by R. W. Hamming, Prentice Hall, 1980.
3. *The Art of Error Correcting Coding* by Robert H. Morelos-Zaragoza, Wesley, 2002.
4. *Error Coding Cookbook* by C. Britton Rorabaugh, McGraw-Hill, 1996.

7.11 MODEL QUESTIONS

1. How does a single-bit error differ from a burst error?
2. Discuss the concept of redundancy in error detection and correction.
3. Distinguish between forward error correction versus error correction by retransmission.
4. What is the definition of a linear block code? What is the definition of a cyclic code?
5. What is the Hamming distance? What is the minimum Hamming distance?
6. In CRC, show the relationship between the following entities (size means the number of bits):

- a. The size of the data word and the size of the codeword
 - b. The size of the divisor and the remainder
 - c. The degree of the polynomial generator and the size of the divisor
 - d. The degree of the polynomial generator and the size of the remainder
7. What kind of arithmetic is used to add data items in checksum calculation?
 8. What kind of error is undetectable by the checksum?

Exercises

1. Apply the exclusive-OR operation on the following pair of patterns:
 - a. $(10001) \oplus (10000)$
 - b. $(10001) \oplus (10001)$ (What do you infer from the result?)
 - c. $(11100) \oplus (00000)$ (What do you infer from the result?)
 - d. $(10011) \oplus (11111)$ (What do you infer from the result?)
2. What is the Hamming distance for each of the following code words:
 - a. $d(10000, 00000)$
 - b. $d(10101, 10000)$
 - c. $d(11111, 11111)$
 - d. $d(000, 000)$
3. Find the minimum Hamming distance for the following cases:
 - a. Detection of two errors.
 - b. Correction of two errors.
 - c. Detection of 3 errors or correction of 2 errors.
 - d. Detection of 6 errors or correction of 2 errors.
4. Answer the following questions:
 - a. What is the polynomial representation of 101110?
 - b. What is the result of shifting 101110 three bits to the left?
 - c. Repeat part b using polynomials.
 - d. What is the result of shifting 101110 four bits to the right?



DATA LINK CONTROL

Unit Structure

- 8.0 Objectives
- 8.1 Introduction
- 8.2 Framing
- 8.3 Flow and Error Control
- 8.4 Protocols
- 8.5 Noiseless Channels
- 8.6 Noisy Channels
- 8.7 HDLC
- 8.8 Point to Point Protocol
- 8.9 Summary
- 8.10 Review Your Learning's
- 8.11 Sample Questions:
- 8.12 References for further reading

8.0 OBJECTIVES

1. Define functions of Data Link Layer in data transmission.
2. Describe how Data Link Layer prepares data for transmission on network media.
3. Describe Protocols used in Data Link Layer.
4. Distinguish Stop-and-Wait and Go-Back-N ARQ protocols.
5. Explain the purpose of encapsulating packets into frames to facilitate media access.

8.1 INTRODUCTION

The two main functions of the data link layer are **data link control** and **media access control**. The first, data link control, deals with the design and procedures for communication between two adjacent nodes: node-to-node communication. This we will see in this chapter. The second function of the data link layer is media access control, or how to share the link.

Data link control functions **include framing, flow and error control, and software implemented protocols** that provide smooth and reliable transmission of frames between nodes. Here, we will first discuss framing, or how to organize the bits that are carried by the physical layer, then we will see flow and error control.

To implement data link control, we need protocols. Each protocol is a set of rules that need to be implemented in software and run by the two nodes involved in data exchange at the data link layer. There are five protocols: two for noiseless (ideal) channels and three for noisy (real) channels. Those in the first category are not actually implemented but provide a foundation for understanding the protocols in the second category.

8.2 FRAMING

Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination. The physical layer provides bit synchronization to ensure that the sender and receiver use the same bit durations and timing. The data link layer, on the other hand, needs to pack bits into frames, so that each frame is distinguishable from another. Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter. In addition, each envelope defines the sender and receiver addresses since the postal system is a many-to-many carrier facility.

Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.

Although the whole message could be packed in one frame, that is not normally done. One reason is that a frame can be very large, making flow and error control very inefficient. When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole message. When a message is divided into smaller frames, a single-bit error affects only that small frame.

Fixed-Size Framing

Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells.

Variable-Size Framing

Our main discussion in this chapter concerns variable-size framing, prevalent in local area networks. In variable-size framing, we need a way to define the end of the frame and the beginning of the next. Historically,

two approaches were used for this purpose: a character-oriented approach and a bit-oriented approach.

Character-Oriented Protocols

In a character-oriented protocol, data to be carried are 8-bit characters from a coding system such as ASCII (see Appendix A). The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection or error correction redundant bits, are also multiples of 8 bits. To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame. Figure shows the format of a frame in a character-oriented protocol.

Character-oriented framing was common when only text was exchanged by the data link layers. The flag could be selected to be any character not used for text communication. Now, however, we send other types of information such as graphs, audio, and video. Any pattern used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame. To fix this problem, a byte-stuffing strategy was added to character-oriented framing. In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.

Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it creates another problem. What happens if the text contains one or more escape characters followed by a flag? The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame. To solve this problem, the escape characters that are part of the text must also be marked by another escape character. In other words, if the escape character is part of the text, an extra one is added to show that the second one is part of the text. Figure 8.1 shows the situation. Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.

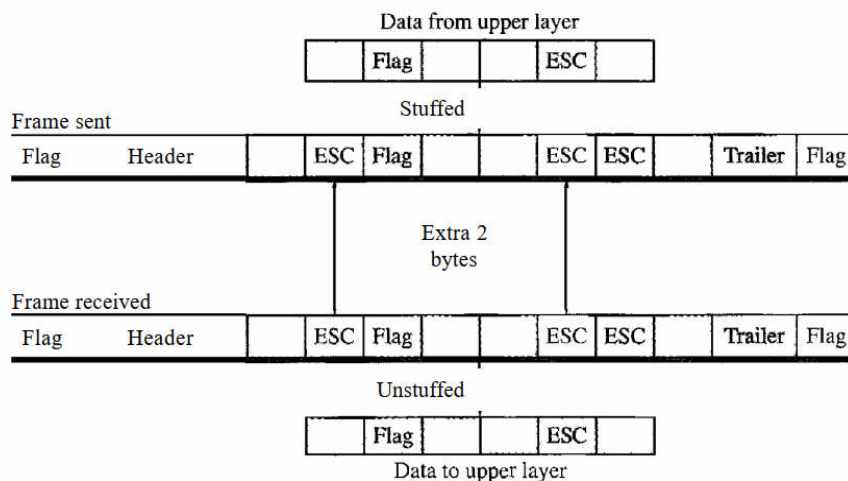


Figure 8.1: Byte Stuffing and Unstuffing

Character-oriented protocols present another problem in data communications.

The universal coding systems in use today, such as Unicode, have 16-bit and 32-bit characters that conflict with 8-bit characters. We can say that in general, the tendency is moving toward the bit-oriented protocols that we discuss next.

Bit-Oriented Protocols

In a bit-oriented protocol, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on. However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame, as shown in Figure 8.2

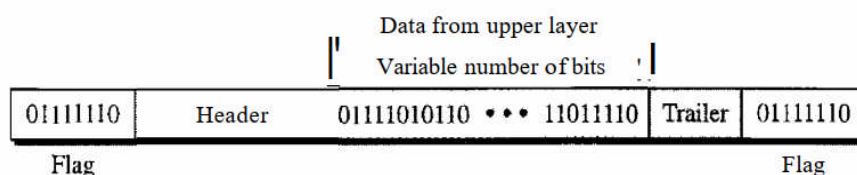


Figure 8.2: A frame in bit-oriented protocol

Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 01111110 for a flag.

This flag can create the same type of problem we saw in the byte-oriented protocols. That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing.

In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver. Note that the extra bit is added after one 0 followed by five 1s regardless of the value of the next bit. This guarantees that the flag field sequence does not inadvertently appear in the frame.

Figure 8.3 shows bit stuffing at the sender and bit removal at the receiver. Note that even if we have a 0 after five 1s, we still stuff a 0. The 0 will be removed by the receiver.

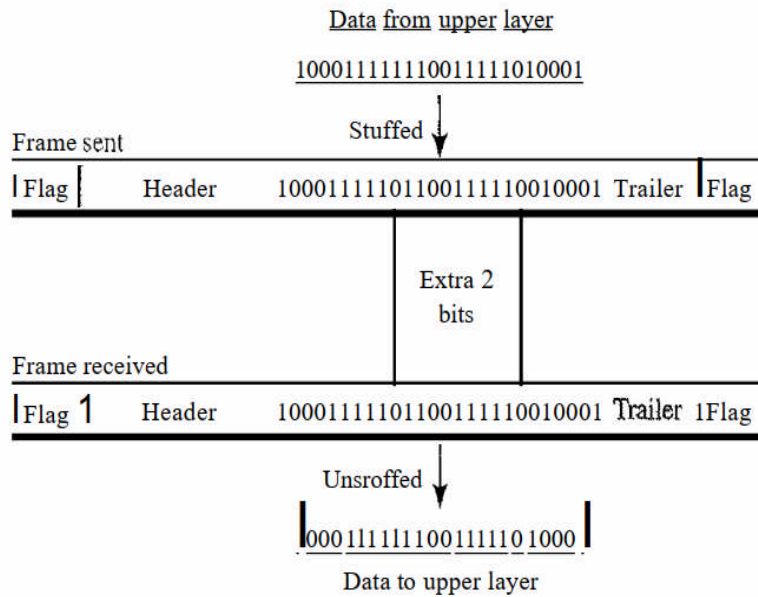


Figure: 8.3: bit Stuffing and Unstuffing

This means that if the flag like pattern 01111110 appears in the data, it will change to 011111010 (stuffed) and is not mistaken as a flag by the receiver. The real flag 01111110 is not stuffed by the sender and is recognized by the receiver

8.3 FLOW AND ERROR CONTROL

Data communication requires at least two devices working together, one to send and thitherto receive. The most important responsibilities of the data link layer are flow control and error control. Together these functions are known as **data link control**.

Flow Control

Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer. In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver. The flow of data must not be

allowed to overwhelm the receiver. Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data. The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily. Incoming data must be checked and processed before they can be used. The rate of such processing is often slower than the rate of transmission. For this reason, each receiving device has a block of memory, called a buffer, reserved for storing incoming data until they are processed. If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.

Error Control

Error control in the data link layer is based on automatic repeat request, which is there transmission of data. Error control is both *error detection and error correction*. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates there transmission of those frames by the sender. In the data link layer, the term *error control* refers primarily to *methods of error detection and retransmission*. Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (ARQ).

8.4 PROTOCOLS

Now we need to combine framing, flow control, and error control to achieve the delivery of data from one node to another. The protocols are normally implemented in software by using one of the common programming languages. Protocols can be used for noiseless (error-free) channels and those that can be used for noisy (error-creating) channels. The protocols in the first category cannot be used in real life, but they serve as a basis for understanding the protocols of noisy channels. Figure 8.4 shows the classifications.

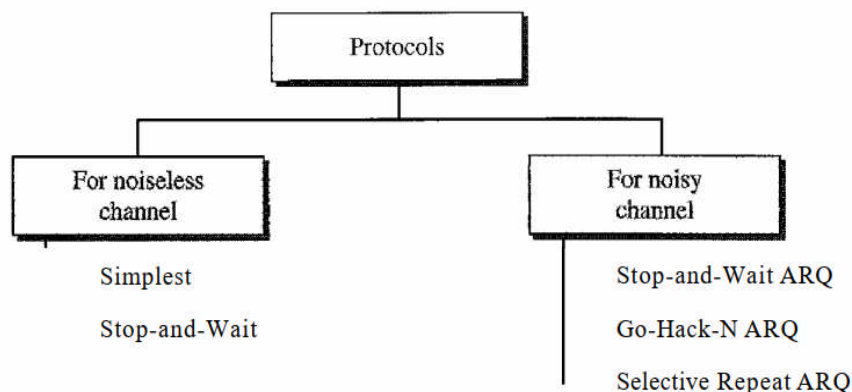


Figure: 8.4: Taxonomy of Data Link Layer Protocols

All these protocols are unidirectional in the sense that the data frames travel from the sender node to receiver node. Special frames, called acknowledgment (ACK) and negative acknowledgment (NAK) can flow

in the opposite direction for flow and error control purposes, but data flow in only one direction. However, in a real-life network, the data link protocols are implemented as bi-directional. In these protocols the flow and error control information such as ACKs and NAKs is included in the data frames in a technique called piggybacking. Because bidirectional protocols are more complex than unidirectional ones.

8.5 NOISELESS CHANNELS

There are ideal channels in which no frames are lost, duplicated, or corrupted. There are two protocols for this type of channel. The first is a protocol which does not use flow control and other is which does.

Stop-and-Wait Protocol

If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use. Normally, the receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either the discarding of frames or denial of service. To prevent the receiver from becoming overwhelmed with frames, we somehow need to tell the sender to slow down. There must be feedback from the receiver to the sender. This protocol is called the Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame. It will have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction.

Design

Following figure 8.10 illustrates the mechanism. At any time, there is either one data frame on the forward channel or one ACK frame on the reverse channel. It uses half-duplex link.

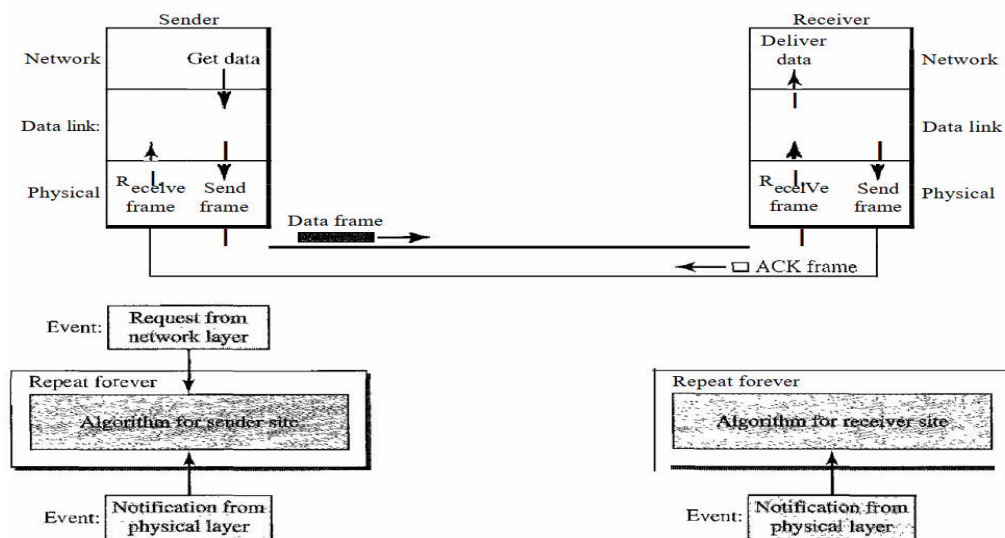


Figure 8.5: Design of Stop-and-Wait Protocol

Example

Following figure 8.13 shows an example of communication using this protocol. It is still very simple. The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame. Note that sending two frames in the protocol involves the sender in four events and the receiver in two events.

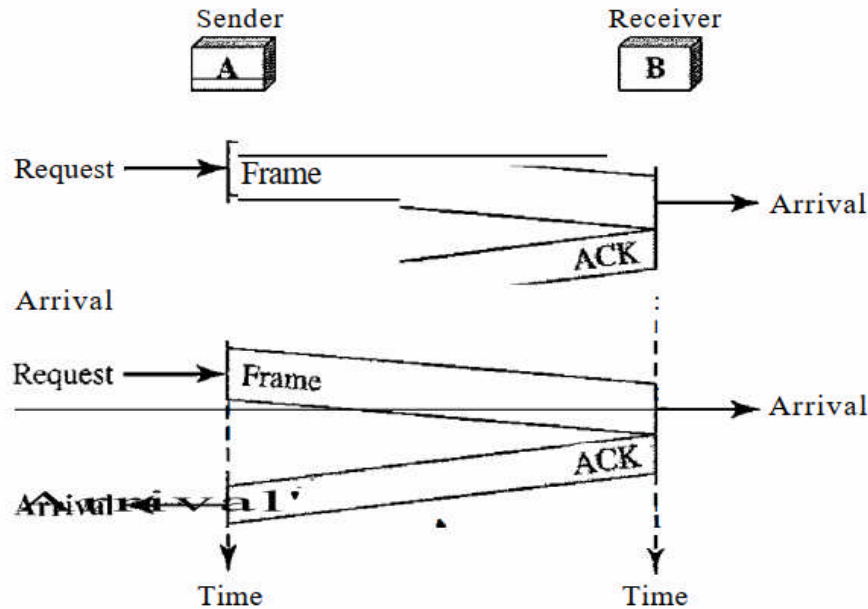


Figure 8.6: Flow Diagram of Above Example

8.6 NOISY CHANNELS

Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are non-existent. We can ignore the error or we need to add error control to our protocols.

Stop-and-Wait Automatic Repeat Request

Our first protocol, called the Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ), adds **a simple error control mechanism** to the Stop-and-Wait Protocol. To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver. Lost frames are more difficult to handle than corrupted ones.

In our previous protocols, there was no way to identify a frame. The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated. Using frame-number we can do sequencing and if any frame is required then we can ask sender to resend it. To remedy this problem, the sender keeps a copy of the sent frame. At the same time, it starts a timer. If

the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted. Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number. The ACK frame for this protocol has a sequence number field. In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one.

1. Sequence Numbers

A field is added to the data frame to hold the sequence number of that frame. For example, if we decide that the field is m bits long, the sequence numbers start from 0, go to $2^m - 1$, and then are repeated.

2. Acknowledgment Numbers

The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver. For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1 (meaning frame 1 is expected next). If frame 1 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 0 (meaning frame 0 is expected)

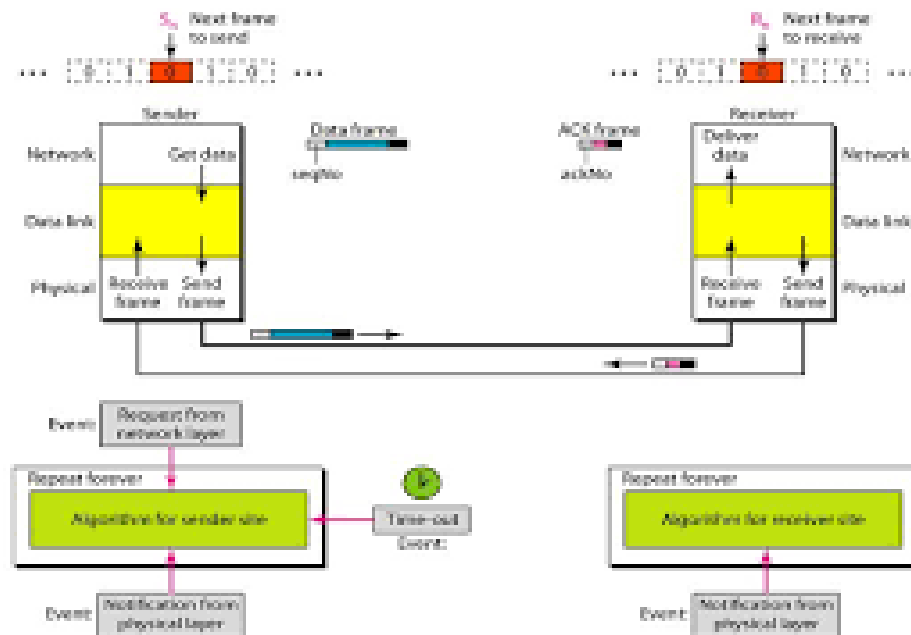


Figure 8.7: Design of Stop-and-Wait ARQ Protocol

Example

Assume that, in a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 ms to make a round trip. What is the bandwidth-delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

Solution

The bandwidth-delay product is
 $(1 \times 10^6) \times (20 \times 10^{-3}) = 20,000$ bits

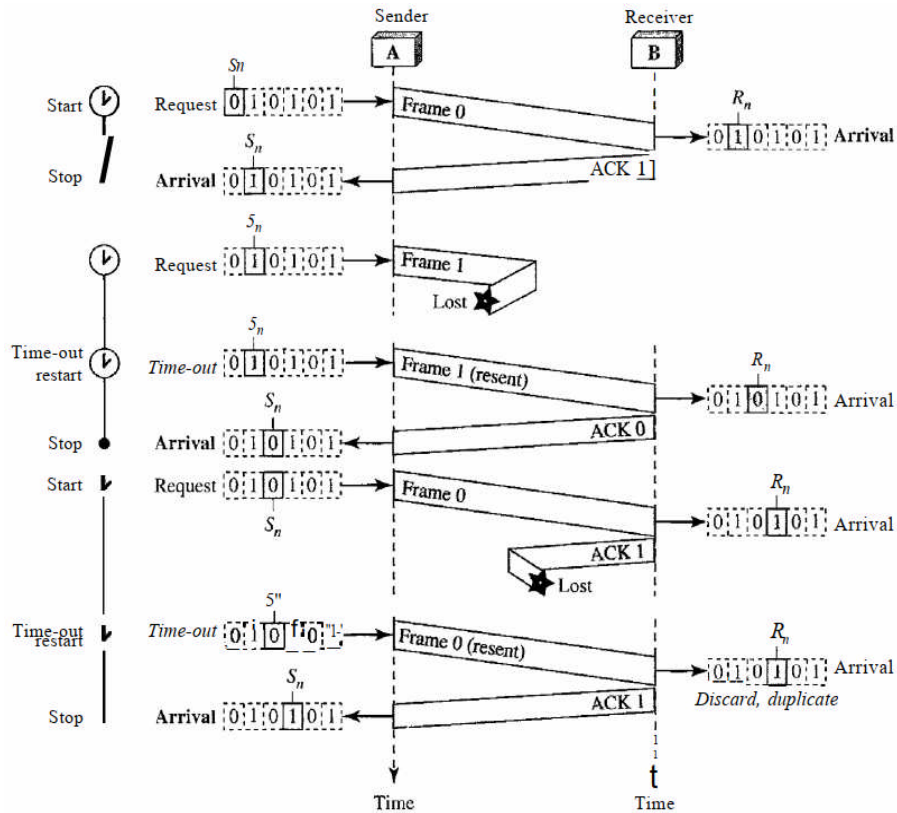


Figure 8.8: Flow Diagram for above example

The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and then back again. However, the system sends only 1000 bits. We can say that the link utilization is only 1000/20,000, or 5 percent. For this reason, for a link with a high bandwidth or long delay, the use of Stop-and-Wait ARQ wastes the capacity of the link.

Example

What is the utilization percentage of the link in above example if we have a protocol that can send up to 15 frames before stopping and worrying about the acknowledgments?

Solution

The bandwidth-delay product is still 20,000 bits. The system can send up to 15 frames or 15,000 bits during a round trip. This means the utilization is 15,000/20,000, or 75 percent. Of course, if there are damaged frames, the utilization percentage is much less because frames have to be resent.

Pipelining

In networking and in other areas, a task is often begun before the previous task has ended. This is known as pipelining. There is no pipelining in Stop-and-Wait ARQ because we need to wait for a frame to reach the destination and be acknowledged before the next frame can be sent. However, pipelining does apply to our next two protocols because several frames can be sent before we receive news about the previous frames.

Pipelining improves the efficiency of the transmission if the number of bits in transition is large with respect to the bandwidth-delay product.

Sliding Window

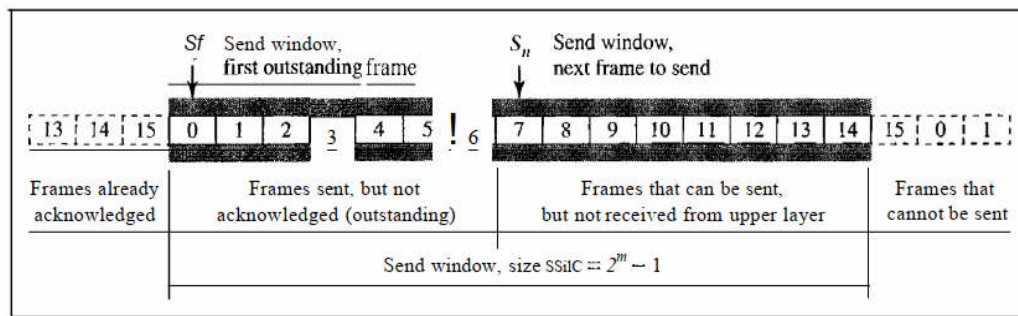
In this protocol the sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver. In other words, the sender and receiver need to deal with only part of the possible sequence numbers. The range which is the concern of the sender is called the send sliding window; the range that is the concern of the receiver is called the receive sliding window. We discuss both here. The send window is an imaginary box covering the sequence numbers of the data frames which can be in transit. In each window position, some of these sequence numbers define the frames that have been sent; others define those that can be sent. The maximum size of the window is $2^m - 1$. Let the size be fixed and set to the maximum value. Figure shows a sliding window of size 15 ($m=4$).

The window at any time divides the possible sequence numbers into four regions.

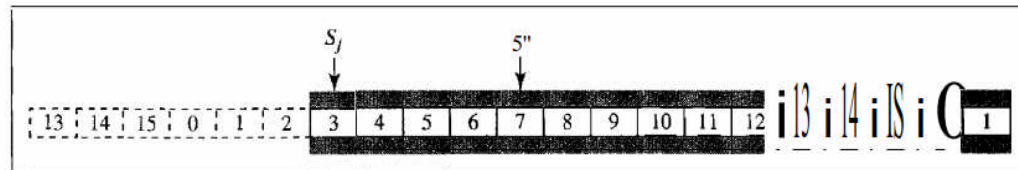
- The first region, from the far left to the left wall of the window, defines the sequence numbers belonging to frames that are already acknowledged. The sender does not worry about these frames and keeps no copies of them.
- The second region, colored in Figure defines the range of sequence numbers belonging to the frames that are sent and have an unknown status. The sender needs to wait to find out if these frames have been received or were lost. We call these outstanding frames.
- The third range, white in the figure, defines the range of sequence numbers for frames that can be sent; however, the corresponding data packets have not yet been received from the network layer.
- The fourth region defines sequence numbers that cannot be used until the window slides.

The send window is an abstract concept defining an imaginary box of size $2^m - 1$ with three variables: S_f , S_n and S_{size} where S_f is send window, i.e., the first outstanding frame, S_n is send window, i.e. the next frame to be sent and S_{size} is Send window having size.

The send window can slide one or more slots when a valid acknowledgment arrives.



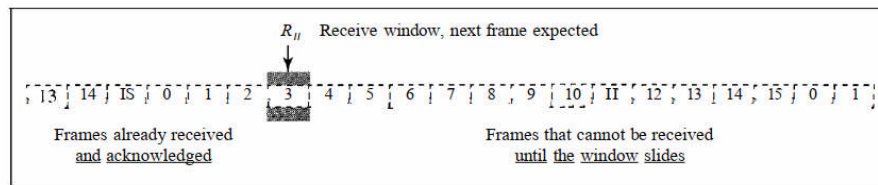
a. Send window before sliding



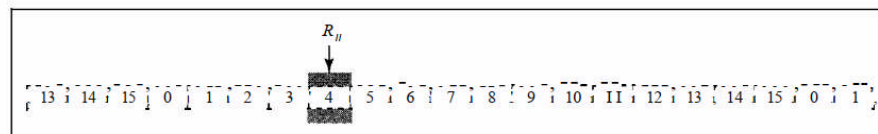
b. Send window after sliding

Figure 8.9: Send Window for Go-Back-N ARQ

The receive window makes sure that the correct data frames are received and that the correct acknowledgments are sent. The size of the receive window is always 1. The receiver is always looking for the arrival of a specific frame. Any frame arriving out of order is discarded and needs to be resent. Following figure shows the receive window.



a. Receive window



b. Window after sliding

Figure 8.10: Receive Window for Go-Back-N ARQ

The receive window is an abstract concept defining an imaginary box of size 1 with one single variable Rn . The window slides when a correct frame has arrived; sliding occurs one slot at a time. Note that we need only one variable Rn (receive window, next frame expected) to define this abstraction. The sequence numbers to the left of the window belong to the frames already received and acknowledged; the sequence numbers to the right of this window define the frames that cannot be received. Any received frame with a sequence number in these two regions is discarded. Only a frame with a sequence number matching the value of Rn is accepted and acknowledged. The receive window also

slides, but only one slot at a time. When a correct frame is received (and a frame is received only one at a time), the window slides.

Acknowledgment

The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order. If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting.

Resending a Frame

When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3, 4, 5, and 6 again. That is why the protocol is called *Go-Back-N* ARQ.

Design

Figure shows the design for this protocol. As we can see, multiple frames can be in transit in the forward direction, and multiple acknowledgments in the reverse direction. The idea is similar to Stop-and-Wait ARQ; the difference is that the send window allows us to have as many frames in transition as there are slots in the send window.

Send Window Size

We can now show why the size of the send window must be less than $2m$. As an example, we choose $m = 2$, which means the size of the window can be $2^m - 1$, or 3.

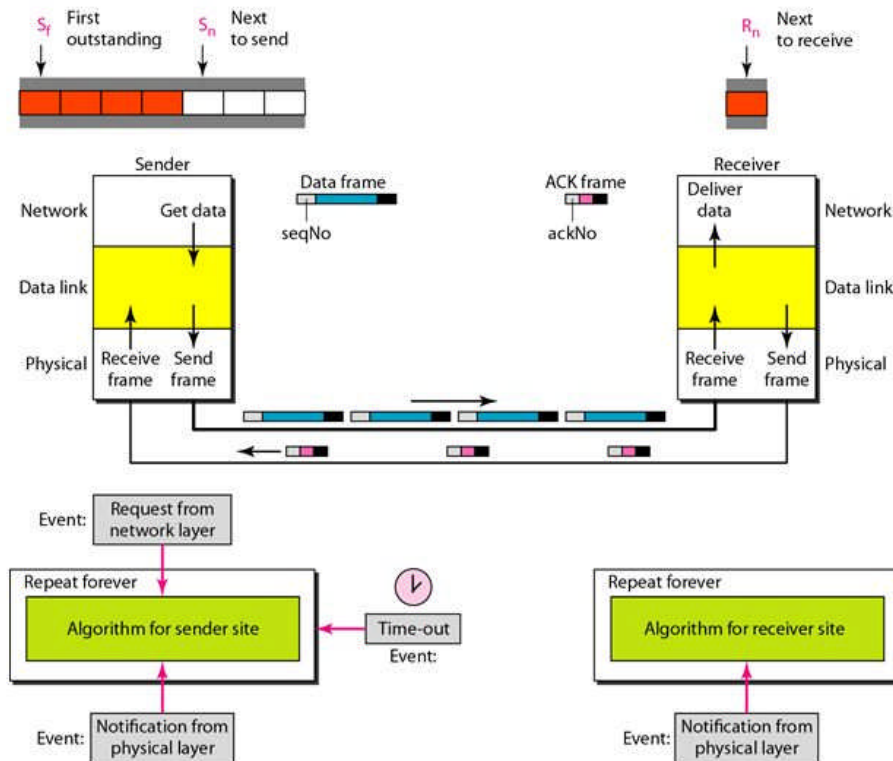


Figure 8.11: Design for Go-Back-N ARQ Protocol

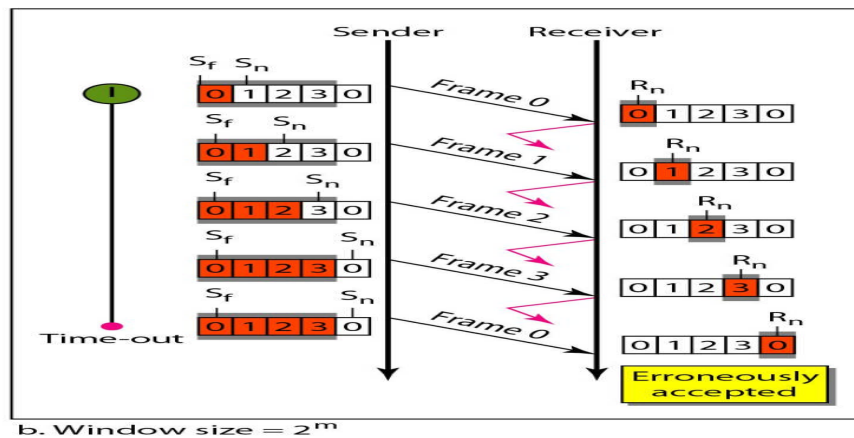


Figure 8.12: Window Size for Go-Back-N ARQ Protocol

8.7 HDLC

High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms we discussed in this chapter above.

Configurations and Transfer Modes

HDLC provides two common transfer modes that can be used in different configurations: normal response mode (NRM) and asynchronous balanced mode (ABM).

Normal Response Mode

In normal response mode (NRM), the station configuration is unbalanced. We have one primary station and multiple secondary stations. A primary station can send commands; a secondary station can only respond. The NRM is used for both point-to-point and multiple-point links, as shown in following figure.

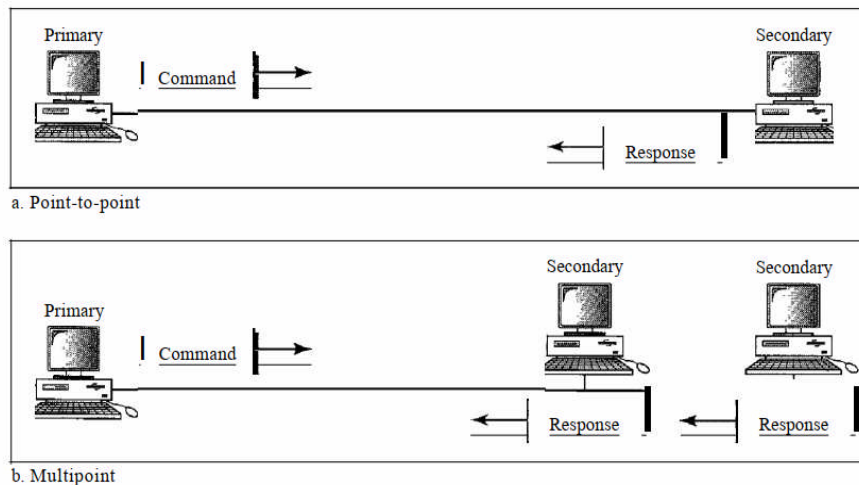


Figure 8.13: Normal Response Mode

Asynchronous Balanced Mode

In asynchronous balanced mode (ABM), the configuration is balanced. The link is point-to-point, and each station can function as a primary and a secondary (acting as peers), as shown in following figure. This is the common mode today.



Figure 8.14: Asynchronous Balanced Mode

Frames

To provide the flexibility necessary to support all the options possible in the modes and configurations just described, HDLC defines three types of frames: information frames (I-frames), supervisory frames (S-frames), and unnumbered frames (V-frames). Each type of frame serves as an envelope for the transmission of a different type of message. I-frames are used to transport user data and control information relating to user data (piggybacking). S-frames are used only to transport control information. V-frames are reserved for system management. Information carried by V-frames is intended for managing the link itself.

Frame Format

Each frame in HDLC may contain up to six fields, as shown in Figure 11.27: a beginning flag field, an address field, a control field, an information field, a frame check sequence (FCS) field, and an ending flag field. In multiple-frame transmissions, the ending flag of one frame can serve as the beginning flag of the next frame.

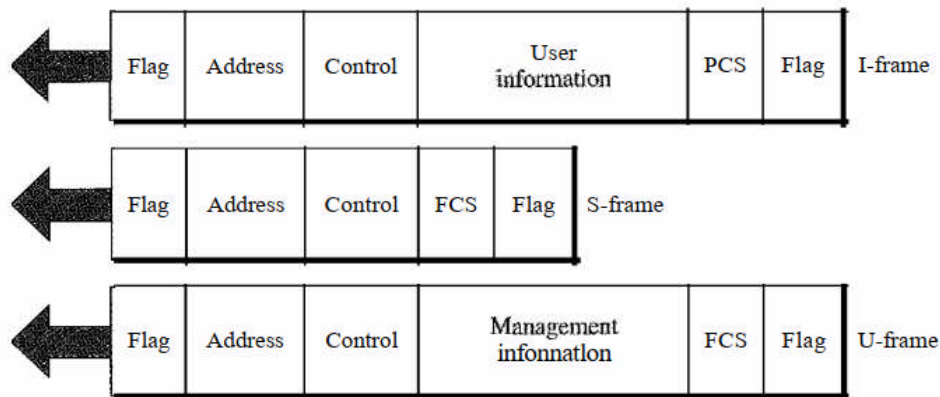


Figure 8.15: HDLC Frames

Fields

Let us now discuss the fields and their use in different frame types.

- **Flag field.** The flag field of an HDLC frame is an 8-bit sequence with the bit pattern 01111110 that identifies both the beginning and the end of a frame and serves as a synchronization pattern for the receiver.

- **Address field.** The second field of an HDLC frame contains the address of the secondary station. If a primary station created the frame, it contains a *to* address. If a secondary creates the frame, it contains a *from* address. An address field can be 1 byte or several bytes long, depending on the needs of the network.
- **Control field:** The control field is a 1- or 2-byte segment of the frame used for flow and error control. The interpretation of bits in this field depends on the frame type.
- **Information field:** The information field contains the user's data from the network layer or management information. Its length can vary from one network to another.
- **FCS field:** The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte ITU-T CRC.
- **Control Field:** The control field determines the type of frame and defines its functionality. The formats are shown in following figure.

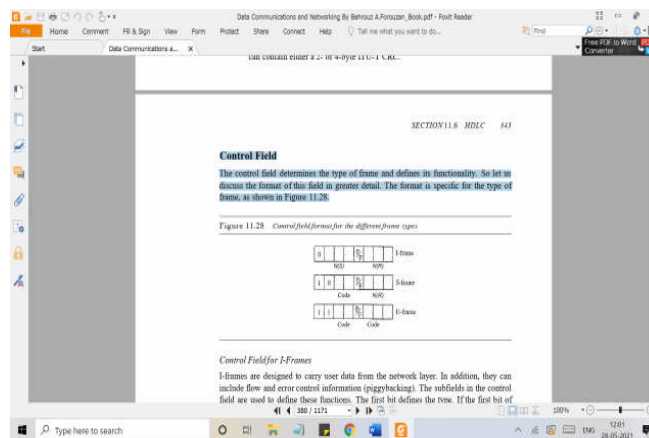


Figure 8.16: Control Field Format

8.8 POINT TO POINT PROTOCOL

Although HDLC is a general protocol that can be used for both point-to-point and multipoint configurations, one of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP). Today, millions of Internet users who need to connect their home computers to the server of an Internet service provider use PPP. The majority of these users have a traditional modem; they are connected to the Internet through a telephone line, which provides the services of the physical layer. But to control and manage the transfer of data, there is a need for a point-to-point protocol at the data link layer. PPP is by far the most common.

PPP provides several services:

1. PPP defines the format of the frame to be exchanged between devices.
2. PPP defines how two devices can negotiate the establishment of the link and the exchange of data.

3. PPP defines how network layer data are encapsulated in the data link frame.
4. PPP defines how two devices can authenticate each other.
5. PPP provides multiple network layer services supporting a variety of network layer protocols.
6. PPP provides connections over multiple links.
7. PPP provides network address configuration. This is particularly useful when a home user needs a temporary network address to connect to the Internet.

On the other hand, to keep PPP simple, several services are missing:

1. PPP does not provide flow control. A sender can send several frames one after another with no concern about overwhelming the receiver.
2. PPP has a very simple mechanism for error control. A CRC field is used to detect errors. If the frame is corrupted, it is silently discarded; the upper-layer protocol needs to take care of the problem. Lack of error control and sequence numbering may cause a packet to be received out of order.
3. PPP does not provide a sophisticated addressing mechanism to handle frames in a multipoint configuration.

8.9 SUMMARY

- Data link control deals with the design and procedures for communication between two adjacent nodes: node-to-node communication.
- Framing in the data link layer separates a message from one source to a destination, or from other messages going from other sources to other destinations,
- Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of frames; in variable-size framing, we need a delimiter (flag) to define the boundary of two frames.
- Variable-size framing uses two categories of protocols: byte-oriented (or character-oriented) and bit-oriented. In a byte-oriented protocol, the data section of a frame is a sequence of bytes; in a bit-oriented protocol, the data section of a frame is a sequence of bits.
- In byte-oriented (or character-oriented) protocols, we use byte stuffing; a special byte added to the data section of the frame when there is a character with the same pattern as the flag.
- In bit-oriented protocols, we use bit stuffing; an extra 0 is added to the data section of the frame when there is a sequence of bits with the same pattern as the flag.

- Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment. Error control refers to methods of error detection and correction.
- For the noiseless channel, we discussed two protocols: the Simplest Protocol and the Stop-and-Wait Protocol. The first protocol has neither flow nor error control; the second has no error control. In the Simplest Protocol, the sender sends its frames one after another with no regards to the receiver. In the Stop-and-Wait Protocol, the sender sends one frame, stops until it receives confirmation from the receiver, and then sends the next frame.
- For the noisy channel, we discussed three protocols: Stop-and-Wait ARQ, GO-Back-N, and Selective Repeat ARQ. The Stop-and-Wait ARQ Protocol, adds a simple error control mechanism to the Stop-and-Wait Protocol. In the Go-Back-NARQ Protocol, we can send several frames before receiving acknowledgments, improving the efficiency of transmission. In the Selective Repeat ARQ protocol we avoid unnecessary transmission by sending only frames that are corrupted.
- High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. However, the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP), which is a byte-oriented protocol.

8.10 REVIEW YOUR LEARNINGS:

1. Are you able to explain Stop-and-Wait Protocol?
2. Are you able to explain framing? Bit oriented and Byte oriented Stuffing?
3. Are you able to explain protocols of Data Link Layer?
4. Are you able to explain the functions of Data Link Layer?
5. Can you explain what is Sliding Window in networks?

8.11 SAMPLE QUESTIONS:

1. Briefly describe the services provided by the data link layer.
2. Define framing and the reason for its need.
3. Compare and contrast byte-oriented and bit-oriented protocols. Which category has been popular in the past (explain the reason)? Which category is popular now(explain the reason)?
4. Compare and contrast byte-stuffing and bit-stuffing. Which technique is used in byte-oriented protocols? Which technique is used in bit-oriented protocols?
5. Compare and contrast flow control and error control.

6. What are the two protocols we discussed for noiseless channels in this chapter?
7. What are the three protocols we discussed for noisy channels in this chapter?
8. Explain the reason for moving from the Stop-and-Wait ARQ Protocol to the GO-Back-N ARQ Protocol.
9. Compare and contrast the Go-Back-N ARQ Protocol with Selective-Repeat ARQ.
10. Compare and contrast HDLC with PPP. Which one is byte-oriented; which one is bit-oriented?
11. Define piggybacking and its usefulness.
12. Which of the protocols described in this chapter utilize pipelining?

8.12 REFERENCES FOR FURTHER READING

- Computer Networks, Andrew S. Tanenbaum,
- Data Communication and Networking, Behrouz A. Forouzan, Tata McGraw Hill Fifth Edition 2013
- <http://eti2506.elimu.net/Introduction/Books/Data%20Communications%20and%20Networking%20By%20Behrouz%20A.Forouzan.pdf>
- <https://nptel.ac.in/courses/106/105/106105082/>
- <http://www.nptelvideos.in/2012/11/data-communication.html>
- <https://www.edx.org/learn/data-communications>



MEDIA ACCESS CONTROL

Unit Structure

9.0 Objectives:

9.1 Introduction

9.2 Random Access

9.3 Controlled Access

9.4 Channelization

9.4.1 Frequency-Division Multiple Access (FDMA)

9.4.2 Time-Division Multiple Access (TDMA)

9.4.3 Code-Division Multiple Access (CDMA)

9.5 Wired LANs

9.5.1 IEEE Standards

9.5.2 Fast Ethernet

9.5.3 Gigabit Ethernet

9.5.4 10 Gigabit Ethernet

9.6 Summary

9.7 Review Your Learnings

9.8 Sample Questions:

9.9 References for further reading

9.0 OBJECTIVES

1. Define the functions of Data Link Layer
2. Describe protocols used in MAC layer
3. Understand IEEE standards used in computer networks
4. Differentiate between various Ethernet connectivity
5. Identify communication network type based on its working like cellular, Bluetooth, Wi-max, etc.
6. Demonstrate use of Virtual LAN

9.1 INTRODUCTION

The medium access control (MAC) is a sub layer of the data link layer of the open system interconnections (OSI) reference model for data transmission. It is responsible for flow control and multiplexing for transmission medium. It controls the transmission of data packets via remotely shared channels. It sends data over the network interface card.

A media access control is a network data transfer policy that determines how data is transmitted between two computer terminals through a network cable. The media access control policy involves sub-layers of the data link layer 2 in the OSI reference model.

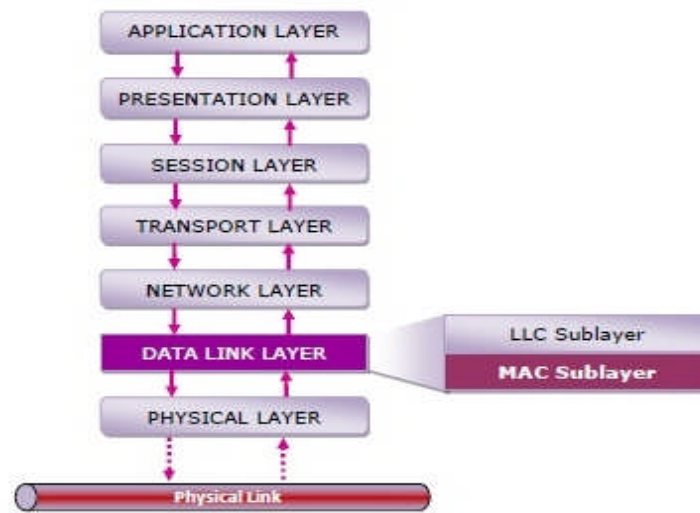


Figure 9.1: Mac Layer in OSI Reference Layer

Functions of MAC Layer

- o It provides an abstraction of the physical layer to the LLC and upper layers of the OSI network.
- o It is responsible for encapsulating frames so that they are suitable for transmission via the physical medium.
- o It resolves the addressing of source node as well as the destination node, or groups of destination nodes.
- o It performs multiple access resolutions when more than one data frame is to be transmitted. It determines the channel access methods for transmission.
- o It also performs collision resolution and initiating retransmission in case of collisions. It generates the frame check sequences and thus contributes to protection against transmission errors.

MAC Addresses

MAC address or media access control address is a unique identifier allotted to a network interface controller (NIC) of a device. It is used as a network address for data transmission within a network segment like Ethernet, Wi-Fi, and Bluetooth.

MAC address is assigned to a network adapter at the time of manufacturing. It is hardwired or hard coded in the network interface card (NIC). A MAC address comprises of six groups of two hexadecimal digits, separated by hyphens, colons, or no separators. An example of a MAC address is 00:0A:89:5B:F0:11.

The essence of the MAC protocol is to ensure non-collision and eases the transfer of data packets between two computer terminals. A collision takes place when two or more terminals transmit data/information simultaneously. This leads to a breakdown of communication, which can prove costly for organizations that lean heavily on data transmission. When nodes or nodes are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link.

9.2 RANDOM ACCESS

In random access or contention methods, no node is superior to another node, and none is assigned the control over another. At each instance, a node that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision depends on the state of the medium (idle or busy). Two features give this method its name. First, there is no scheduled time for a node to transmit. Transmission is random among the nodes. That is why these methods are called random access. Second, no rules specify which node should send next. Nodes compete with one another to access the medium. That is why these methods are also called contention methods.

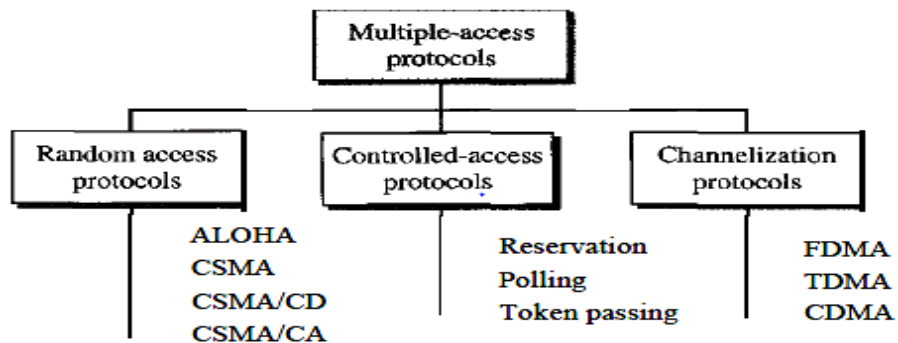


Figure 9.2: Multiple Access Protocols

In this, each node has the right to the medium without being controlled by any other node. However, if more than one node tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified. To avoid access conflict or to resolve it when it happens, each node follows a procedure that answers the following questions:

- When can the node access the medium?
- What can the node do if the medium is busy?
- How can the node determine the success or failure of the transmission?
- What can the node do if there is an access conflict?

The random-access methods are developed from protocol known as ALOHA, which used a very simple procedure called multiple access (MA). The method was improved with the addition of a procedure that

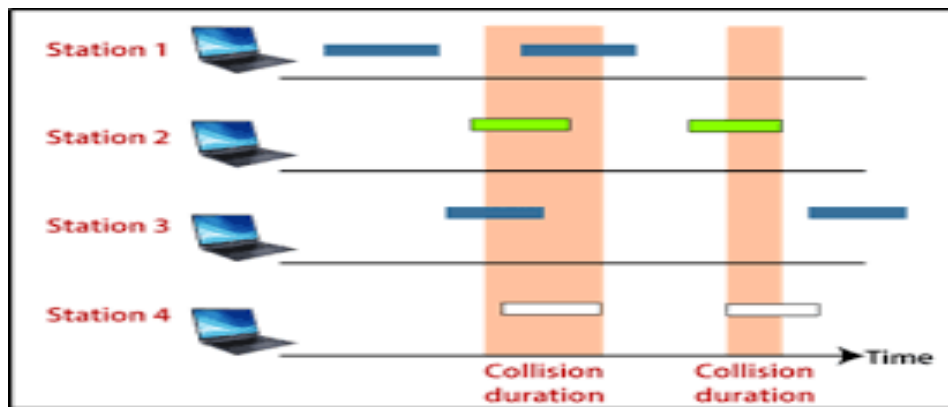
forces the node to sense the medium before transmitting. This was called carrier sense multiple access. This method then developed into two methods namely 1. Carrier Sense Multiple Access with Collision Detection (CSMA/CD) and 2. Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). CSMA/CD tells the node what to do when a collision is detected. CSMA/CA tries to avoid the collision.

ALOHA

ALOHA, the earliest random-access method, was designed for a radio (wireless) LAN, but it can be used on any shared medium. There can be potential collisions in this arrangement as the medium is shared between the nodes.

Pure ALOHA

The original ALOHA protocol is called pure ALOHA and is a simple yet an elegant protocol. Each node sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different nodes. Figure shows an example of frame collisions in pure ALOHA.



There are four nodes (unrealistic assumption) that contend with one another for access to the shared channel. The figure shows that each node sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel.

It is obvious that we need to resend the frames that have been destroyed during transmission. The pure ALOHA protocol relies on acknowledgments from the receiver. When a node sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the node assumes that the frame (or the acknowledgment) has been destroyed and resends the frame. A collision involves two or more nodes. If all these nodes try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each node waits a random amount of time before resending its frame. The randomness will help avoid more collisions. This time is the back-off time T_B .

Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames. After a maximum number of retransmissions attempts K_{max} a node must give up and try later. Figure shows the procedure for pure ALOHA based on the above strategy.

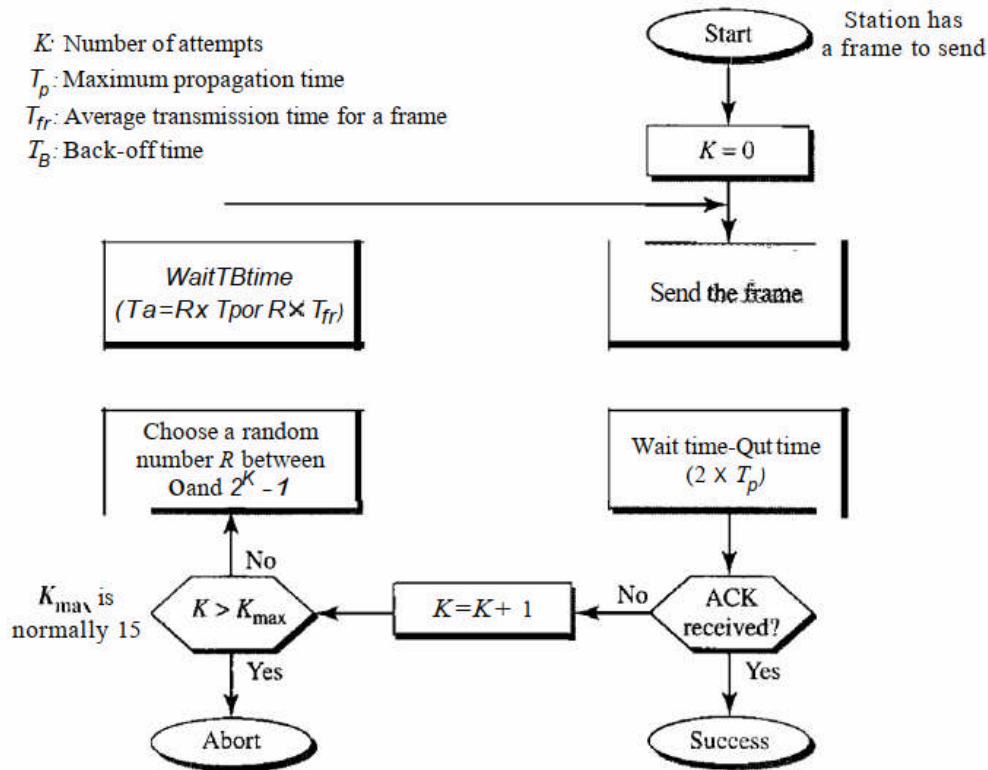


Figure 9.3: Procedure for Pure ALOHA Protocol

The time-out period is equal to the maximum possible round-trip propagation delay, which is twice the amount of time required to send a frame between the two most widely separated nodes ($2 \times T_p$). The back-off time T_B is a random value that normally depends on K (the number of attempted unsuccessful transmissions). The formula for T_B depends on the implementation. One common formula is the **binary exponential back-off**. In this method, for each retransmission, a multiplier in the range 0 to $2K - 1$ is randomly chosen and multiplied by T_p (maximum propagation time) or T_{fr} (the average time required to send out a frame) to find T_B . Note that in this procedure, the range of the random numbers increases after each collision. The value of K_{max} is usually chosen as 15.

Slotted ALOHA

Pure ALOHA has a vulnerable time of $2 \times T_{fr}$. This is so because there is no rule that defines when the node can send. A node may send soon after another node has started or soon before another node has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA. In slotted ALOHA we divide the time into slots of T_{fr} and force the node to send only at the beginning of the time slot. Figure shows an example of frame collisions in slotted ALOHA.

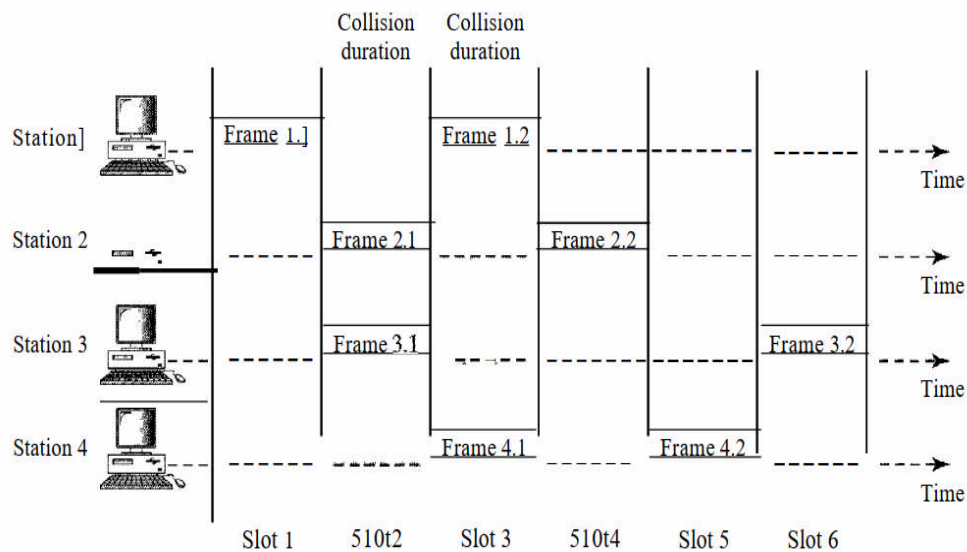


Figure 9.4: Frames in Slotted ALOHA

Because a node is allowed to send only at the beginning of the synchronized timeslot, if a node misses this moment, it must wait until the beginning of the next timeslot. This means that the node which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two nodes try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to T_{fr} . Figure shows the situation. Figure shows that the vulnerable time for slotted ALOHA is one-half that of pure ALOHA.

$$\text{Slotted ALOHA vulnerable time} = T_{fr}$$

Throughput It can be proved that the average number of successful transmissions for slotted ALOHA is $S = G \times e^{-G}$. The maximum throughput S_{max} is 0.368, when $G = 1$. In other words, if a frame is generated during one frame transmission time, then 36.8 percent of these frames reach their destination successfully. This result can be expected because the vulnerable time is equal to the frame transmission time. Therefore, if a node generates only one frame in this vulnerable time (and no other node generates a frame during this time), the frame will reach its destination successfully.

Carrier Sense Multiple Access (CSMA)

To minimize the chance of collision and increase the performance, the CSMA method was formed. The chance of collision can be reduced if a node senses the medium before trying to use it. CSMA is based on the principle "sense before transmit" or "listen before talk." CSMA can reduce the possibility of collision, but it cannot eliminate it. The reason for this is shown in Figure a space and time model of a CSMA network. Nodes are connected to a shared channel. The possibility of collision still exists because of propagation delay; when a node ends a frame, it still takes time for the first bit to reach every node and for every node to sense it.

Persistence Methods

What should a node do if the channel is busy? What should a node do if the channel is idle? Three methods have been devised to answer these questions: the I-persistent method, the no persistent method, and the p-persistent method. Figure shows the behaviour of three persistence methods when a node finds a channel busy.

I-Persistent : The **I-persistent method** is simple and straightforward. In this method, after the node finds the line idle, it sends its frame immediately (with probability 1). This method has the highest chance of collision because two or more nodes may find the line idle and send their frames immediately.

No persistent In the **no persistent method**, a node that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. The no persistent approach reduces the chance of collision because it is unlikely that two or more nodes will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be nodes with frames to send.

p-Persistent: The **p-persistent method** is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency. In this method, after the node finds the line idle it follows these steps:

1. With probability p , the node sends its frame.
2. With probability $q = 1 - p$, the node waits for the beginning of the next time slot and checks the line again.
 - a. If the line is idle, it goes to step 1.
 - b. If the line is busy, it acts as though a collision has occurred and uses the back off procedure.

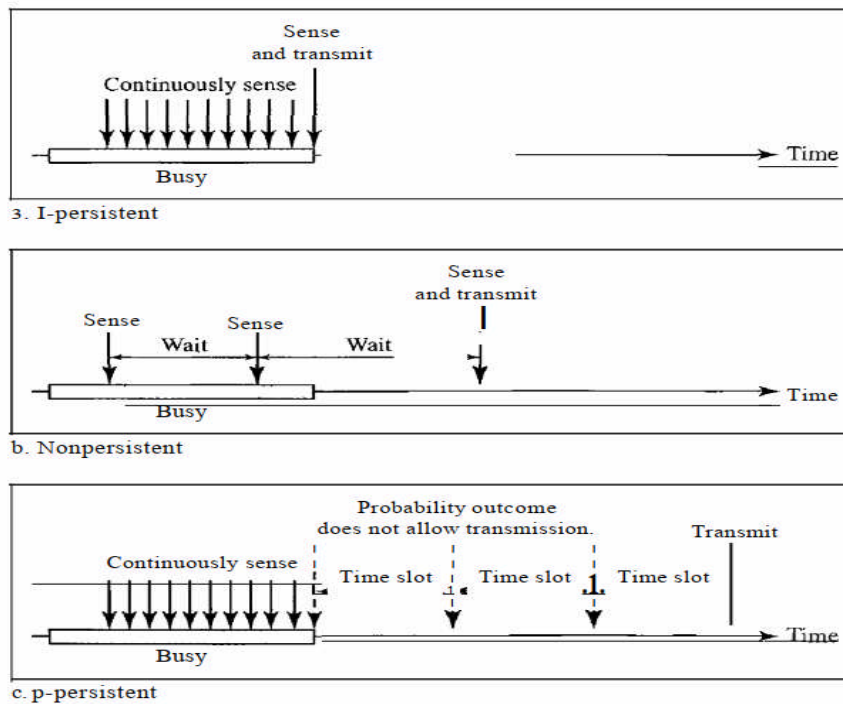


Figure 9.5: Behaviour of Three Persistence Methods

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

The CSMA method does not specify the procedure following a collision. Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision. In this method, a node monitors the medium after it sends a frame to see if the transmission was successful. If so, the node is finished. If, however, there is a collision, the frame is sent again. To better understand CSMA/CD, let us look at the first bits transmitted by the two nodes involved in the collision. Although each node continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide. In Figure nodes A and C are involved in the collision.

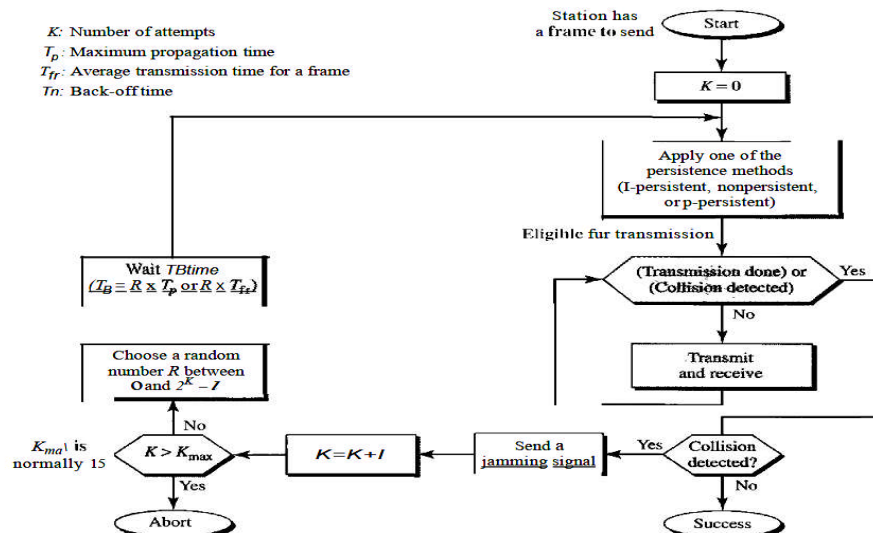


Figure 9.6: Flow Diagram of CSMA/CD

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). The basic idea behind *CSMA/CD* is that a node needs to be able to receive while transmitting to detect a collision. When there is no collision, the node receives one signal: its own signal. When there is a collision, the node receives two signals: its own signal and the signal transmitted by a second node. To distinguish between these two cases, the received signals in these two cases must be significantly different. In other words, the signal from the second node needs to add a significant amount of energy to the one created by the first node.

Collisions are avoided through the use of CSMA ICA's three strategies: the inter frame space, the contention window, and acknowledgments, as shown in Figure.

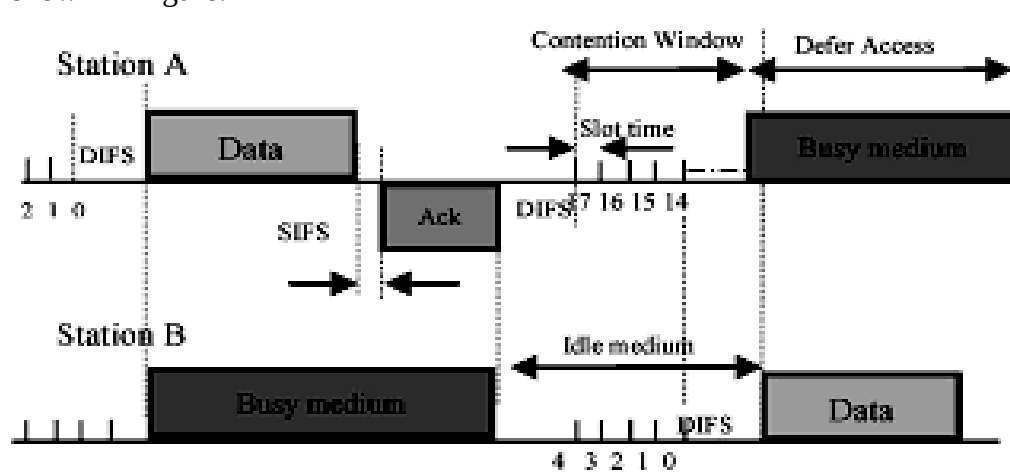


Figure 9.7: Timing in CSMA/CA

Contention Window

The contention window is an amount of time divided into slots. A node that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy. This means that it is set to one slot the first time and then doubles each time the node cannot detect an idle channel after the IFS time. This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting node. One interesting point about the contention window is that the node needs to sense the channel after each time slot. However, if the node finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle. This gives priority to the node with the longest waiting time.

9.3 CONTROLLED ACCESS

In controlled access, the nodes consult one another to find which node has the right to send. A node cannot send unless it has been authorized by other nodes. There are three widespread controlled-access methods.

1. Reservation

In the reservation method, a node needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.

If there are N nodes in the system, there are exactly N reservation minis lots in the reservation frame. Each minis lot belongs to a node. When a node needs to send a data frame, it makes a reservation in its own minis lot. The nodes that have made reservations can send their data frames after the reservation frame. Figure shows a situation with five nodes and a five-minis lot reservation frame. In the first interval, only nodes 1, 3, and 4 have made reservations. In the second interval, only node 1 has made a reservation.

2. Polling

Polling works with topologies in which one device is designated as a primary node and the other devices are secondary nodes. All data exchanges must be made through the primary device even when the ultimate destination is a secondary device. The primary device controls the link; the secondary devices follow its instructions. It is up to the primary device to determine which device is allowed to use the channel at a given time. The primary device, therefore, is always the initiator of a session.

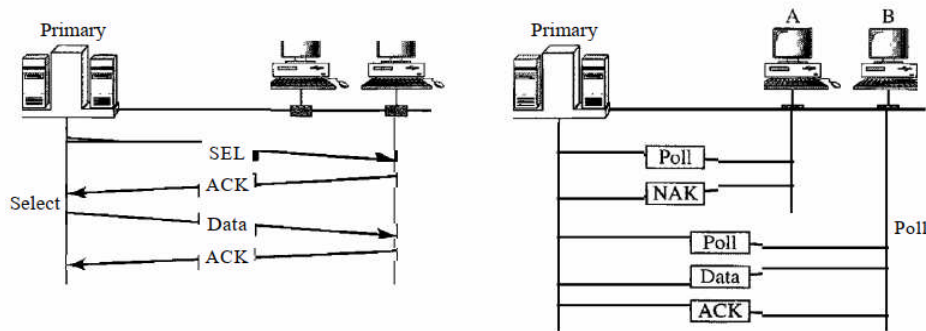


Figure 9.8: Select and Poll function

If the primary wants to receive data, it asks the secondary if they have anything to send; this is called poll function. If the primary wants to send data, it tells the secondary to get ready to receive; this is called select function.

Select

The *select* function is used whenever the primary device has something to send. Remember that the primary controls the link. If the primary is neither sending nor receiving data, it knows the link is available. If it has something to send, the primary device sends it. What it does not know, however, is whether the target device is prepared to receive. So the primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status. Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address of the intended secondary.

Poll

The poll function is used by the primary device to solicit transmissions from the secondary devices. When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send. When the first secondary is approached, it responds either with a NAK frame if it has nothing to send or with data (in the form of a data frame) if it does. If the response is negative (a NAK frame), then the primary polls the next secondary in the same manner until it finds one with data to send. When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (ACK frame), verifying its receipt.

3. Token Passing

In the token-passing method, the nodes in a network are organized in a logical ring. In other words, for each node, there is a *predecessor* and a *successor*. The predecessor is the node which is logically before the node in the ring; the successor is the node which is after the node in the ring. The current node is the one that is accessing the channel now. The right to this access has been passed from the predecessor to the current node. The right will be passed to the successor when the current node has no more data to send.

But how is the right to access the channel passed from one node to another? In this method, a special packet called a token circulates through the ring. The possession of the token gives the node the right to access the channel and send its data. When a node has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data. When the node has no more data to send, it releases the token, passing it to the next logical node in the ring. The node cannot send data until it receives the token again in the next round. In this process, when a node receives the token and has no data to send, it just passes the data to the next node.

Token management is needed for this access method. Nodes must be limited in the time they can have possession of the token. The token must be monitored to ensure it has not been lost or destroyed. For example, if a node that is holding the token fails, the token will disappear from the network. Another function of token management is to assign priorities to the nodes and to the types of data being transmitted. And finally, token management is needed to make low-priority nodes release the token to high priority nodes.

Logical Ring

In a token-passing network, nodes do not have to be physically connected in a ring; the ring can be a logical one. Figure 12.20 show four different physical topologies that can create a logical ring.

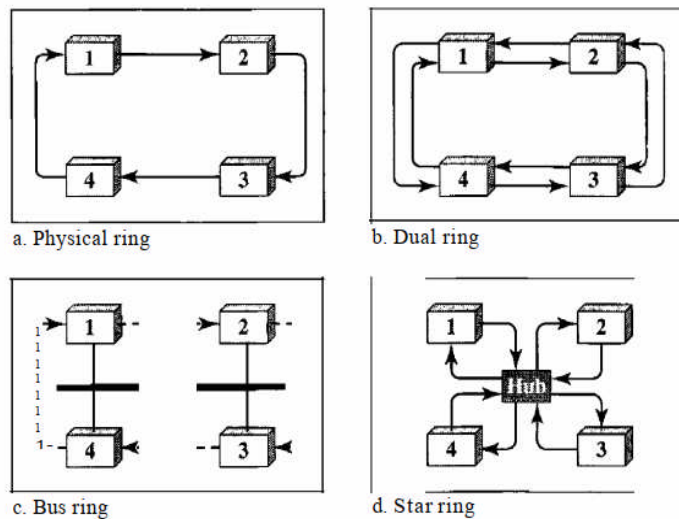


Figure 9.9: Logical Ring and Physical Topology in token-passing access method

In the physical ring topology, when a node sends the token to its successor, the token cannot be seen by other nodes; the successor is the next one in line. This means that the token does not have to have the address of the next successor. The problem with this topology is that if one of the links—the medium between two adjacent nodes fails, the whole system fails.

In the bus ring topology, also called a token bus, the nodes are connected to a single cable called a bus. They, however, make a logical ring, because each node knows the address of its successor (and also predecessor for token management purposes). When a node has finished sending its data, it releases the token and inserts the address of its successor in the token. Only the node with the address matching the destination address of the token gets the token to access the shared media. The Token Bus LAN, standardized by IEEE, uses this topology.

In a star ring topology, the physical topology is a star. There is a hub, however, that acts as the connector. The wiring inside the hub makes the ring; the nodes are connected to this ring through the two wire connections. This topology makes the net workless prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the nodes can operate. Also adding and removing nodes from the ring is easier.

9.4 CHANNELIZATION

Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different nodes. There are three channelization protocols: FDMA, TDMA, and CDMA.

9.4.1 Frequency-Division Multiple Access (FDMA)

In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands. Each node is allocated a band to send its data. In other words, each band is reserved for a specific node, and it belongs to the node all the time. Each node also uses a band pass filter to confine the transmitter frequencies. To prevent node interferences, the allocated bands are separated from one another by small *guard bands*. Figure shows the idea of FDMA.

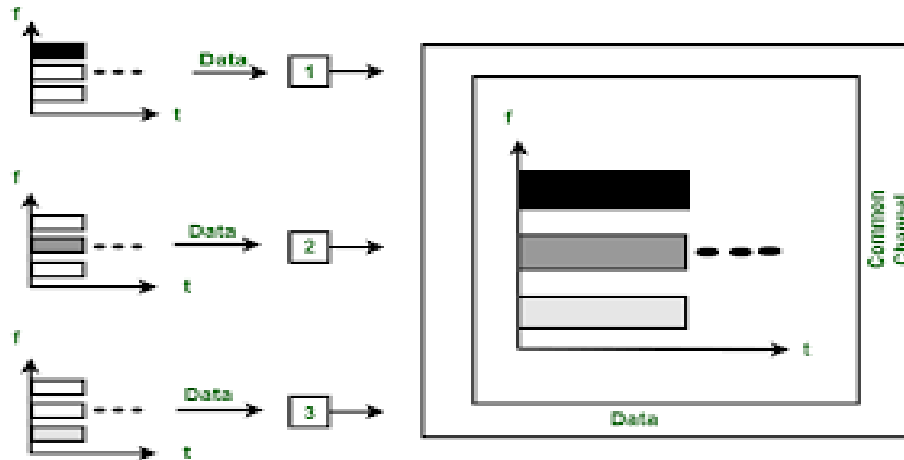


Figure 9.10: FDMA

FDMA specifies a predetermined frequency band for the entire period of communication. This means that stream data (a continuous flow of data that may not be packetized) can easily be used with FDMA. This can be used in cellular telephone systems.

FDM is a physical layer technique that combines the loads from low-bandwidth channels and transmits them by using a high-bandwidth channel. The channels that are combined are low pass. The multiplexer modulates the signals, combines them, and creates a band pass signal. The bandwidth of each channel is shifted by the multiplexer. FDMA is an access method in the data link layer. The data link layer in each node tells its physical layer to make a band pass signal from the data passed to it. The signal must be created in the allocated band. There is no physical multiplexer at the physical layer. The signals created at each node are automatically band pass filtered. They are mixed when they are sent to the common channel.

9.4.2 Time-Division Multiple Access (TDMA)

In time-division multiple access (TDMA), the nodes share the bandwidth of the channel in time. Each node is allocated a time slot during which it can send data. Each node transmits its data in its assigned time slot. Figure 12.22 shows the idea behind TDMA

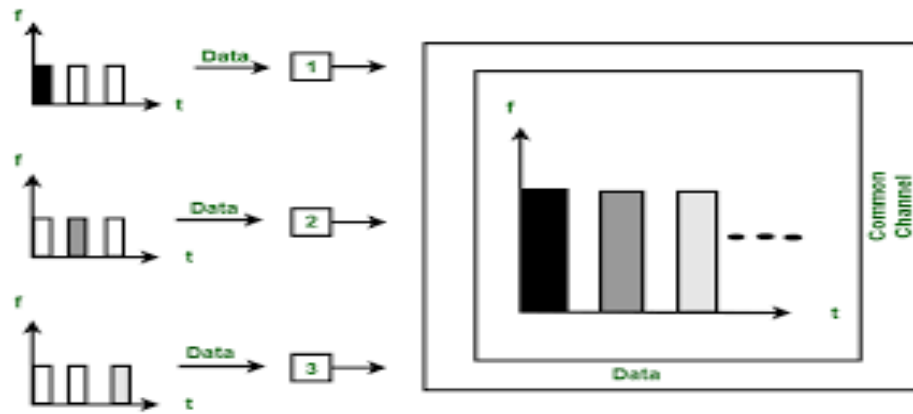


Figure 9.11: TDMA

The main problem with TDMA lies in achieving synchronization between the different nodes. Each node needs to know the beginning of its slot and the location of its slot. This may be difficult because of propagation delays introduced in the system if the nodes are spread over a large area.

9.4.3 Code-Division Multiple Access (CDMA)

Code-division multiple access (CDMA) was conceived several decades ago. Recent advances in electronic technology have finally made its implementation possible. CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link. It differs from TDMA because all nodes can send data simultaneously; there is no timesharing.

Example:

CDMA simply means communication with different codes. For example, in a large room with many people, two people can talk in English if nobody else understands English. Another two people can talk in Chinese if they are the only ones who understand Chinese, and so on. In other words, the common channel, the space of the room in this case, can easily allow communication between several couples, but in different languages (codes).

9.5 WIRED LANS

A local area network (LAN) is a computer network that is designed for a limited geographic area such as a building or a campus. Although a LAN can be used as an isolated network to connect computers in an organization for the sole purpose of sharing resources, most LANs today are also linked to a wide area network(WAN) or the Internet. The LAN market has seen several technologies such as Ethernet, Token Ring, Token Bus, FDDI, and ATM LAN.

9.5.1 IEEE Standards

In 1987, the American National Standards Institute (ANSI) has formed the standards which were approved by the International

Organization for Standardization (ISO) as an international standard under the designation ISO 8802. The relationship of the 802 Standard to the traditional OSI model is shown in following figure. The IEEE has subdivided the data link layer into two sub-layers: logical link control (LLC) and media access control (MAC). IEEE has also created several physical layer standards for different LAN protocols.

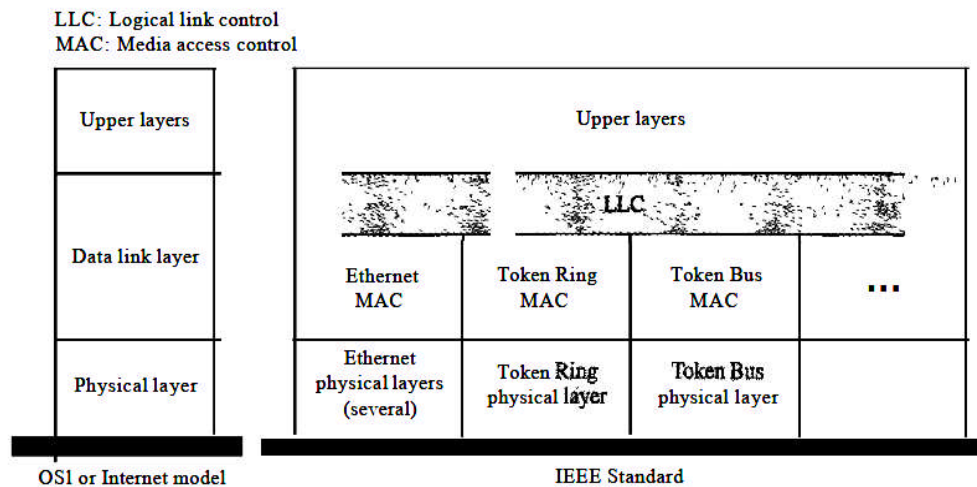


Figure 9.12: IEEE Standards for LAN

Data Link Layer

The data link layer in the IEEE standard is divided into two sub-layers: LLC and MAC.

1. *Logical Link Control (LLC)*

In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sub-layer called the logical link control. Framing is handled in both the LLC sub layer and the MAC sub layer. The LLC provides one single data link control protocol for all IEEE LANs. In this way, the LLC is different from the media access control sub layer, which provides different protocols for different LANs. A single LLC protocol can provide inter-connectivity between different LANs because it makes the MAC sub layer transparent.

Media Access Control (MAC)

Media access control layer defines the specific access method for each LAN. For example, it defines *CSMA/CD* as the media access method for Ethernet LANs and the token passing method for Token Ring and Token Bus LANs. Framing function is also handled by the MAC layer. MAC sub layer contains a number of distinct modules which defines the access method and the framing format specific to the corresponding LAN protocol.

Physical Layer

The physical layer is dependent on the implementation and type of physical media used. IEEE defines detailed specifications for each LAN implementation.

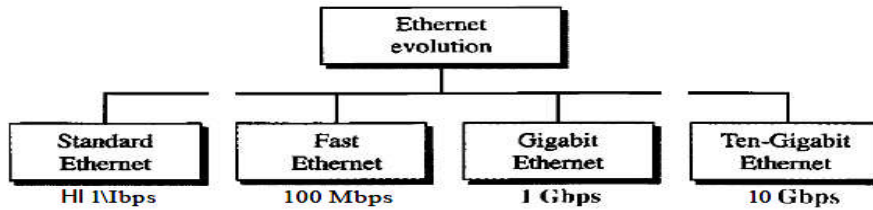


Figure 9.13: Ethernet Evolutions

Addressing

Each node on an Ethernet network (such as a PC, work node, or printer) has its own network interface card (NIC). The NIC fits inside the node and provides the node with a 6-byte physical address. As shown in following figure, the Ethernet address is 6 bytes(48 bits), normally written in hexadecimal notation, with a colon between the bytes.

06:01 :02:01:2C:4B

6 bytes = 12 hex digits = 48 bits

Figure: Example of Ethernet address in hexadecimal notation

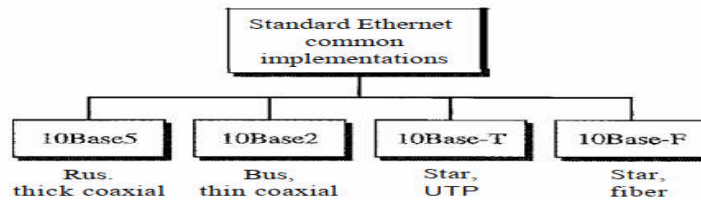


Figure 9.14: Standard Ethernet Categories

9.5.2 Fast Ethernet

Fast Ethernet is a variation of Ethernet standards that carry data traffic at 100 Mbps (Megabits per second) in local area networks (LAN). It was launched as the IEEE 802.3u standard in 1995 and stayed the fastest network till the introduction of Gigabit Ethernet.

Fast Ethernet is popularly named as 100-BASE-X. Here, 100 is the maximum throughput, i.e., 100 Mbps, BASE denoted use of base band transmission, and X is the type of medium used, which is TX or FX.

The common varieties of fast Ethernet are 100-Base-TX, 100-BASE-FX and 100-Base-T4 as shown in figure below.

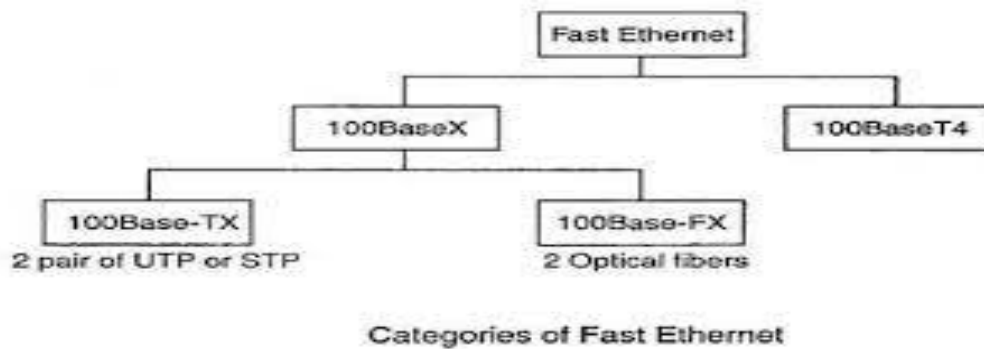


Figure 9.15: Categories of Fast Ethernet

Frame Format of IEEE 802.3

The frame format of IEEE 802.3u is same as IEEE 802.3. The fields in the frame are:

- **Preamble** – It is a 7-bytes starting field that provides alert and timing pulse for transmission.
- **Start of Frame Delimiter (SOF)** – It is a 1-byte field that contains an alternating pattern of ones and zeros ending with two ones.
- **Destination Address** – It is a 6-byte field containing physical address of destination nodes.
- **Source Address** – It is a 6-byte field containing the physical address of the sending node.
- **Length** – It a 2-bytes field that stores the number of bytes in the data field.
- **Data** – This is a variable sized field carries the data from the upper layers. The maximum size of data field is 1500 bytes.
- **Padding** – This is added to the data to bring its length to the minimum requirement of 46 bytes.
- **CRC** – CRC stands for cyclic redundancy check. It contains the error detection information

9.5.3 Gigabit Ethernet

Gigabit Ethernet (GbE) is the family of Ethernet technologies that achieve theoretical data rates of 1 gigabit per second (1 Gbps). It was introduced in 1999 and was defined by the IEEE 802.3ab standard. The popular varieties of fast Ethernet are 1000Base-SX, 1000Base-LX, 1000BASE-T and 1000Base-CX.

The goals of the Gigabit Ethernet design can be summarized as follows:

1. Upgrade the data rate to 1 Gbps.
2. Make it compatible with Standard or Fast Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format.
5. Keep the same minimum and maximum frame lengths.
6. To support auto negotiation as defined in Fast Ethernet.

9.5.4 10 Gigabit Ethernet

10-Gigabit Ethernet is the family of Ethernet technologies that achieve maximum rates up to 10 gigabits per second (10 Gbps). It is also known as 10GE, 10GbE or 10 Gig E. It is defined by the IEEE 802.3ae-2002 standard.

10GE is a thousand times faster than standard Ethernet and supports only full-duplex communication. Multimode fibre having 0.85 μ frequency is used for medium distances and single mode fibre having 1.5 μ frequency is used for long distances.

The popular varieties of fast Ethernet are 1000Base-SX, 1000Base-LX, 1000BASE-T and 1000Base-CX.

The goals of the Ten-Gigabit Ethernet design can be summarized as follows:

1. Upgrade the data rate to 10 Gbps.
2. Make it compatible with Standard, Fast, and Gigabit Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format.
5. Keep the same minimum and maximum frame lengths.
6. Allow the interconnection of existing LANs into a metropolitan area network (MAN) or a wide area network (WAN).
7. Make Ethernet compatible with technologies such as Frame Relay and ATM

9.6 SUMMARY

- We can consider the data link layer as two sub layers. The upper sub layer is responsible for data link control, and the lower sub layer is responsible for resolving access to the shared media.
- Many formal protocols have been devised to handle access to a shared link. We categorize them into three groups: random access protocols, controlled access protocols, and channelization protocols.
- In random access or contention methods, no node is superior to another node and none is assigned the control over another.
- ALOHA allows multiple access (MA) to the shared medium. There are potential collisions in this arrangement. When a node sends data, another node may attempt to do so at the same time. The data from the two nodes collide and become garbled.
- Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different nodes. We discussed three channelization protocols: FDMA, TDMA, and CDMA.

- The common Fast Ethernet implementations are 100Base-TX (two pairs of twisted pair cable), 100 Base-FX (two fiber-optic cables), and 100Base-T4 (four pairs of voice-grade, or higher, twisted-pair cable).
- Gigabit Ethernet has a data rate of 1000 Mbps

9.7 REVIEW YOUR LEARNINGS:

1. List three categories of multiple access protocols discussed in this chapter.
2. Define random access and list three protocols in this category
3. Define controlled access and list three protocols in this category.
4. Define channelization and list three protocols in this category.
5. Explain why collision is an issue in a random-access protocol but not in controlled access or channelizing protocols.
6. Compare and contrast a random-access protocol with a controlled access protocol.
7. Compare and contrast a random-access protocol with a channelizing protocol.
8. Compare and contrast a controlled access protocol with a channelizing protocol.
9. Do we need a multiple access protocol when we use the localloop of the telephone company to access the Internet? Why?

9.8 SAMPLE QUESTIONS:

1. In a *CDMA/CD* network with a data rate of 10 Mbps, the minimum frame size is found to be 512 bits for the correct operation of the collision detection process. What should be the minimum frame size if we increase the data rate to 100 Mbps? To 1 Gbps? To 10 Gbps?
2. One hundred nodes on a pure ALOHA network share a 1-Mbps channel. If frames are 1000 bits long, find the throughput if each node is sending 10 frames per second?
3. Compare the data rates for Standard Ethernet, Fast Ethernet, Gigabit Ethernet, and Ten-Gigabit Ethernet
4. What are the common Ten-Gigabit Ethernet implementations?
5. Explain CSMA/CD.
6. Explain Fast Ethernet and its types.

9.9 REFERENCES FOR FURTHER READING

- Data Communication and Networking, Behrouz A. Forouzan , Tata McGraw Hill Fifth Edition 2013
- Computer Networks, Andrew S. Tanenbaum,
- <https://nptel.ac.in/courses/106/105/106105183/>
- <https://nptel.ac.in/content/storage2/courses/106105080/pdf/M5L2.pdf>
- <https://www.coursera.org/lecture/peer-to-peer-protocols-local-area-networks/medium-access-control-nWWWd>



WIRELESS LAN, CONNECTING DEVICES AND VIRTUAL LAN

Unit Structure :

- 10.0 Objectives
- 10.1 Connecting Devices
 - 10.1.1 Passive Hubs
 - 10.1.2 Repeaters
 - 10.1.3 Active Hubs
 - 10.1.4 Bridges
 - 10.1.5 Two-Layer Switches
 - 10.1.6 Routers
 - 10.1.7 Three-Layer Switches
 - 10.1.8 Gateway
- 10.2 Wireless LANs
 - 10.2.1 IEEE 802.11
 - 10.2.2 Bluetooth
- 10.3 Wi-Max
- 10.4 Cellular Telephony
- 10.5 Satellite Networks
- 10.6 Virtual LAN
 - 10.6.1 Features of VLANs
 - 10.6.2 Types of VLANs
 - 10.6.3 Advantages of VLAN
- 10.7 Summary
- 10.8 Review Your Learnings
- 10.9 Sample Questions
- 10.10 References for further reading

10.0 OBJECTIVES

1. Explain IEEE 802.11 standards
2. Describe Bluetooth, Wi-max technologies.
3. Explain working of Cellular Networks.
4. Analyse difference between satellite networks and cellular networks

5. Analyse connecting devices like routers, gateways and switches which are used in network connections.
6. Explain Virtual LAN

10.1CONNECTING DEVICES

In this section, we divide connecting devices into five different categories based on the layer in which they operate in a network, as shown in figure below:

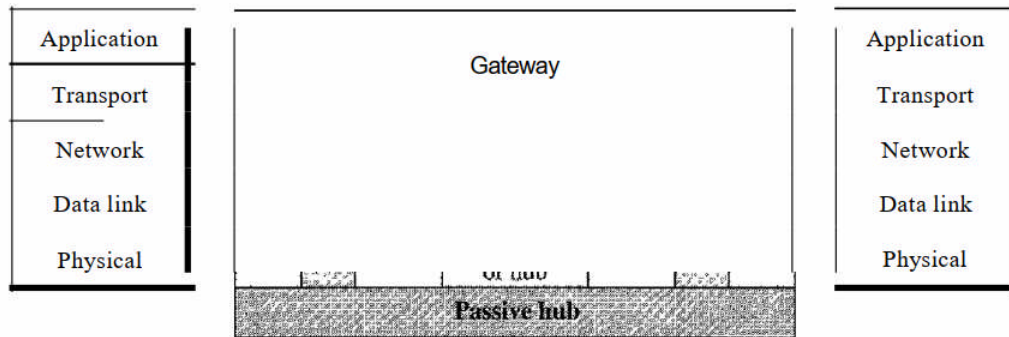


Figure 10.1: Five Categories of Connecting Devices

The five categories contain devices which can be defined as

1. Those which operate below the physical layer such as a passive hub.
2. Those which operate at the physical layer (a repeater or an active hub).
3. Those which operate at the physical and data link layers (a bridge or a two-layer switch).
4. Those which operate at the physical, data link, and network layers (a router or a three-layer switch).
5. Those which can operate at all five layers (a gateway).

10.1.1Passive Hubs

A passive hub is just a connector. It connects the wires coming from different branches. In a star-topology Ethernet LAN, a passive hub is just a point where the signals coming from different stations collide; the hub is the collision point. This type of a hub is part of the media; its location in the Internet model is below the physical layer.

10.1.2Repeaters

A repeater is a device that operates only in the physical layer. Signals that carry information within a network can travel a fixed distance before attenuation endangers the integrity of the data. A repeater receives a signal and, before it becomes too weak or corrupted, regenerates the original bit pattern. The repeater then sends the refreshed signal. A repeater can extend the physical length of a LAN, as shown in figure below.

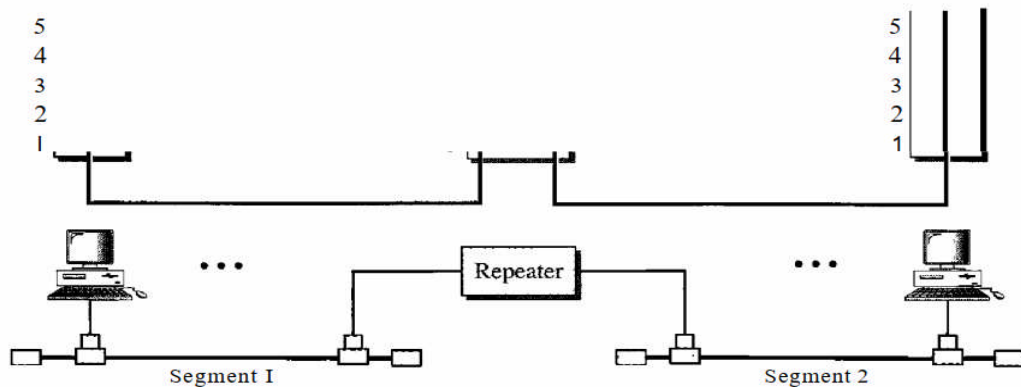


Figure 10.2: Repeater connecting 2 segments of LAN

A repeater does not actually connect two LANs; it connects two segments of the same LAN. The segments connected are still part of one single LAN. A repeater is not a device that can connect two LANs of different protocols.

A repeater can overcome the 10Base5 Ethernet length restriction. In this standard, the length of the cable is limited to 500 m. To extend this length, we divide the cable into segments and install repeaters between segments. Note that the whole network is still considered one LAN, but the portions of the network separated by repeaters are called segments. The repeater acts as a two-port node, but operates only in the physical layer. When it receives a frame from any of the ports, it regenerates and forwards it to the other port.

10.1.3 Active Hubs

An active hub is actually a multipart repeater. It is normally used to create connections between stations in a physical star topology. We have seen examples of hubs in some Ethernet implementations (10 Base-T, for example). However, hubs can also be used to create multiple levels of hierarchy, as shown in figure below. The hierarchical use of hubs removes the length limitation of 10Base-T (100 m).

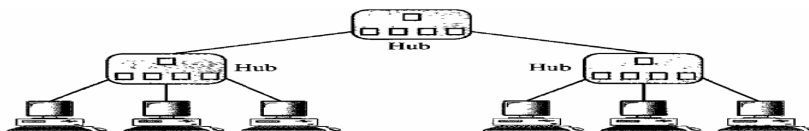


Figure 10.3: Hierarchy of Hubs

10.1.4 Bridges

A bridge operates in both the physical and the data link layer. As a physical layer device, it regenerates the signal it receives. As a data link layer device, the bridge can check the physical (MAC) addresses (source and destination) contained in the frame.

What is the difference in functionality between a bridge and a repeater? A bridge has filtering capability. It can check the destination address of a frame and decide if the frame should be forwarded or dropped. If the frame is to be forwarded, the decision must specify the port. A bridge has a table that maps address to ports.

In following figure, two LANs are connected by a bridge. If a frame destined for station 712B13456142 arrives at port 1, the bridge consults its table to find the departing port. According to its table, frames for 712B 13456142 leave through port 1; therefore, there is no need for forwarding, and the frame is dropped. On the other hand, if a frame for 712B13456141 arrives at port 2, the departing port is port 1 and the frame is forwarded. In the first case, LAN 2 remains free of traffic; in the second case, both LANs have traffic. In this example, we used a two-port bridge; in reality a bridge usually has more ports.

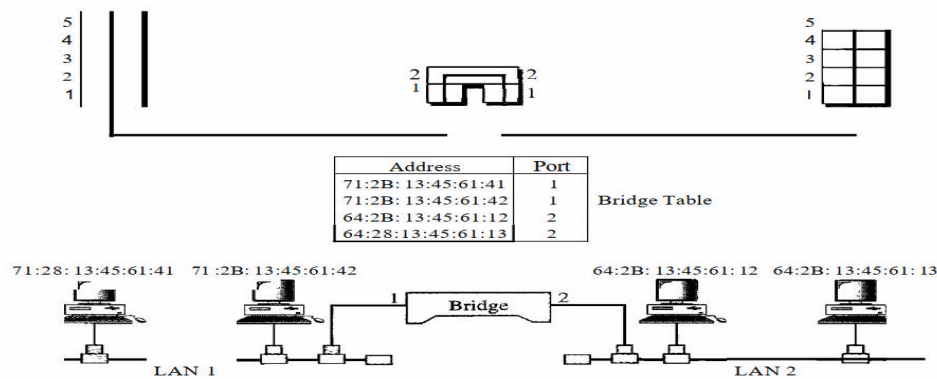


Figure 10.4: A bridge connecting 2 LANs

10.1.5 Two-Layer Switches

When we use the term *switch*, we must be careful because a switch can mean two different things. We must clarify the term by adding the level at which the device operates. We can have a two-layer switch or a three-layer switch. A **three-layer switch** is used at the network layer; it is a kind of router. The **two-layer switch** performs at the physical and data link layers. A two-layer switch is a bridge, a bridge with many ports and a design that allows better (faster) performance

10.1.6 Routers

A router is a three-layer device that routes packets based on their logical addresses (host-to-host addressing). A router normally connects LANs and WANs in the Internet and has a routing table that is used for making decisions about the route. The routing tables are normally dynamic and are updated using routing protocols. Figure shows a part of the Internet that uses routers to connect LANs and WANs.

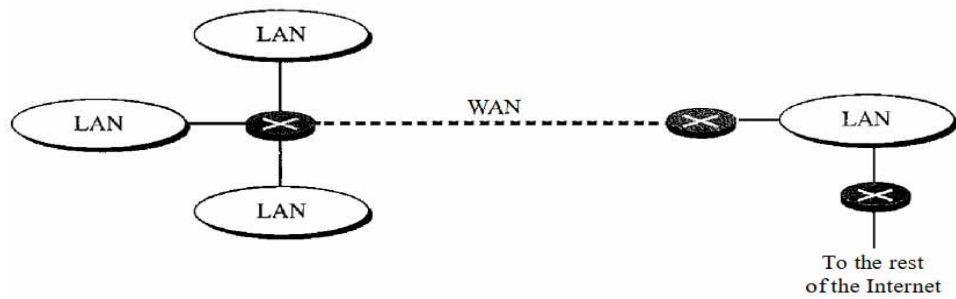


Figure 10.5: Routers Connecting Independent LAN and WAN

10.1.7 Three-Layer Switches

A three-layer switch is a router, but a faster and more sophisticated. The switching fabric in a three-layer switch allows faster table lookup and forwarding. In this book, we use the terms router and three-layer switch interchangeably.

10.1.8 Gateway

Although some textbooks use the terms gateway and router interchangeably, most of the literature distinguishes between the two. A gateway is normally a computer that operates in all five layers of the Internet or seven layers of OSI model. A gateway takes an application message, reads it, and interprets it. This means that it can be used as a connecting device between two internet works that use different models. For example, a network designed to use the OSI model can be connected to another network using the Internet model. The gateway connecting the two systems can take a frame as it arrives from the first system, move it up to the OSI application layer, and remove the message.

10.2 WIRELESS LANS

Wireless communication is one of the fastest-growing technologies. The demand for connecting devices without the use of cables is increasing everywhere. Wireless LANs can be found on college campuses, in office buildings, and in many public areas.

Here we will concentrate on two promising wireless technologies for LANs: IEEE 802.11 wireless LANs, sometimes called wireless Ethernet, and Bluetooth, a technology for small wireless LANs.

10.2.1 IEEE 802.11

IEEE 802.11 standard, popularly known as WiFi, lays down the architecture and specifications of wireless LANs (WLANs). WiFi or WLAN uses high-frequency radio waves instead of cables for connecting the devices in LAN. Users connected by WLANs can move around within the area of network coverage.

IEEE 802.11 Architecture

The components of an IEEE 802.11 architecture are as follows –

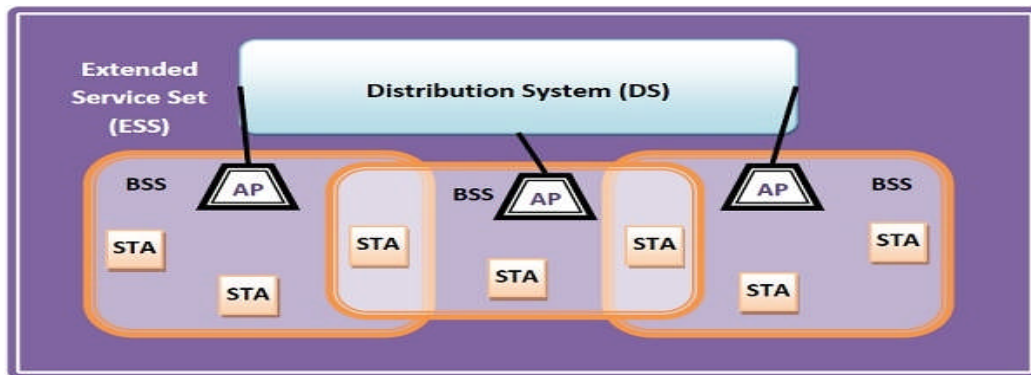
- **Stations (STA)** – Stations comprises of all devices and equipment that are connected to the wireless LAN. A station can be of two types–

Wireless Access Point (WAP) – WAPs or simply access points (AP) are generally wireless routers that form the base stations or access.

Client. Clients are workstations, computers, laptops, printers, smart phones, etc.

Each station has a wireless network interface controller.

- **Basic Service Set (BSS)** – A basic service set is a group of stations communicating at the physical layer level. BSS can be of two categories depending upon the mode of operation–
 - o Infrastructure BSS – Here, the devices communicate with other devices through access points.
 - o Independent BSS – Here, the devices communicate in a peer-to-peer basis in an ad hoc manner.
- 1. **Extended Service Set (ESS)** – It is a set of all connected BSS.
- 2. **Distribution System (DS)** – It connects access points in ESS.

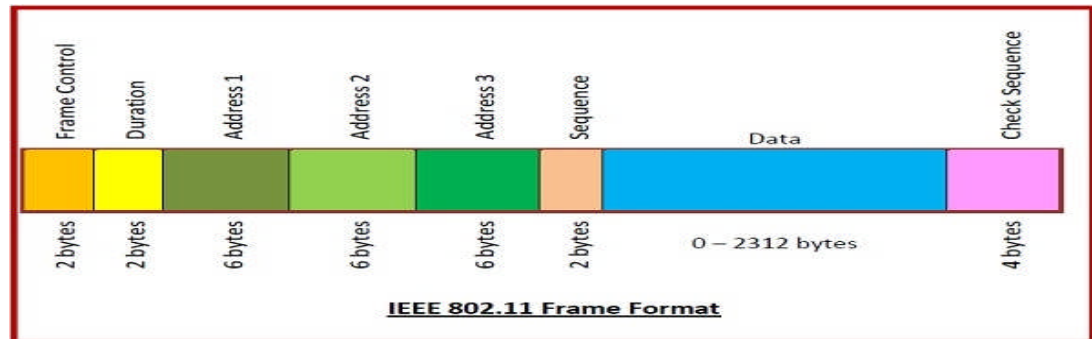


Frame Format of IEEE 802.11

The main fields of a frame of wireless LANs as laid down by IEEE 802.11 are –

- **Frame Control** – It is a 2-bytes starting field composed of 11 subfields. It contains control information of the frame.
- **Duration** – It is a 2-byte field that specifies the time period for which the frame and its acknowledgment occupy the channel.
- **Address fields** – There are three 6-byte address fields containing addresses of source, immediate destination, and final endpoint respectively.
- **Sequence** – It a 2 bytes field that stores the frame numbers.

- **Data** – This is a variable-sized field that carries the data from the upper layers. The maximum size of the data field is 2312 bytes.
- **Check Sequence** – It is a 4-byte field containing error detection information.



10.2.2 Bluetooth

Bluetooth is a network technology that connects mobile devices wirelessly over a short-range to form a personal area network (PAN). They use short-wavelength, ultra-high frequency (UHF) radio waves within the range 2.400 to 2.485 GHz, instead of RS-232 data cables of wired PANs.

Features of Bluetooth

- Bluetooth technology was released in 1999 as Bluetooth 1.0, by Special Interest Group (SIG) who continues to manage it.
- It was initially standardized as IEEE 802.15.1.
- Mobile computing devices and accessories are connected wirelessly by Bluetooth using short-range, low-power, inexpensive radios.
- UHF radio waves within the range of 2.400 to 2.485 GHz are used for data communications.
- A PAN or a piconet can be created by Bluetooth within a 10 m radius.
- Presently, 2 to 8 devices may be connected.
- Bluetooth protocols allow devices within the range to find Bluetooth devices and connect with them. This is called pairing. Once, the devices are paired, they can transfer data securely.
- Bluetooth has lower power consumption and lower implementation costs than Wi-Fi. However, the range and transmission speeds are typically lower than Wi-Fi.
- The lower power requirements make it less susceptible to interference with other wireless devices in the same 2.4GHz bandwidth.
- Bluetooth version 3.0 and higher versions can deliver a data rate of 24 Mbps.

- The Bluetooth version 4.0 came in 2010. It is characterized by low energy consumption, multi vendor interoperability, the economy of implementation, and greater range.

Bluetooth Devices

A Bluetooth device has a built-in short-range radio transmitter. The current data rate is 1 Mbps with a 2.4-GHz bandwidth. This means that there is a possibility of interference between the IEEE 802.11b wireless LANs and Bluetooth LANs.

Bluetooth Layers:

Bluetooth uses several layers that do not exactly match those of the Internet model and these are shown in figure below.

Radio Layer

The radio layer is roughly equivalent to the physical layer of the Internet model. Bluetooth devices are low-power and have a range of 10 m.

Base band Layer

The base band layer is roughly equivalent to the MAC sub layer in LANs. The access method is TDMA (see Chapter 12). The primary and secondary communicate with each other using time slots. The length of a time slot is exactly the same as the dwell time, 625 μ s. This means that during the time that one frequency is used, a sender sends a frame to a secondary, or a secondary sends a frame to the primary. Note that the communication is only between the primary and a secondary; secondaries cannot communicate directly with one another.

L2CAP

The Logical Link Control and Adaptation Protocol, or L2CAP (L2 here means LL), is roughly equivalent to the LLC sub layer in LANs. It is used for data exchange on an ACL link; SCO channels do not use L2CAP.

It uses 16-bit length field which defines the size of the data, in bytes, coming from the upper layers. Data can be up to 65,535 bytes. The channel ID (CID) defines a unique identifier for the virtual channel created at this level. The L2CAP has specific duties: multiplexing, segmentation and reassembly, quality of service (QoS), and group management for multicasting.

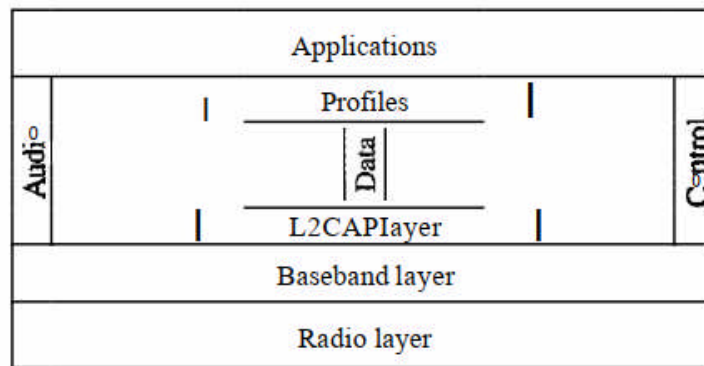


Figure 10.6: Bluetooth Layers

10.3 WI-MAX

WiMAX is one of the hottest broadband wireless technologies around today. It is based on IEEE 802.16 specification, and it is expected to deliver high quality broadband services. This is a brief tutorial that covers the fundamentals of WiMAX.

Wireless means transmitting signals using radio waves as the medium instead of wires. Wireless technologies are used for tasks as simple as switching off the television or as complex as supplying the sales force with information from an automated enterprise application while in the field. Now cordless keyboards and mice, PDAs, pagers and digital and cellular phones have become part of our daily life.



Some of the inherent characteristics of wireless communications systems which make it attractive for users, are given below –

- **Mobility** – A wireless communications system allows users to access information beyond their desk and conduct business from anywhere without having a wire connectivity.
- **Reach ability** – Wireless communication systems enable people to be stay connected and be reachable, regardless of the location they are operating from.

- **Simplicity** – Wireless communication systems are easy and fast to deploy in comparison of cabled network. Initial setup cost could be a bit high but other advantages overcome that high cost.
- **Maintainability** – In a wireless system, you do not have to spend too much cost and time to maintain the network setup.
- **Roaming Services** – Using a wireless network system, you can provide service anywhere any time including train, buses, aeroplanes etc.
- **New Services** – Wireless communication systems provide various smart services like SMS and MMS.

10.4 CELLULAR TELEPHONY

Cellular network is an underlying technology for mobile phones, personal communication systems, wireless networking etc. The technology is developed for mobile radio telephone to replace high power transmitter/receiver systems. Cellular networks use lower power, shorter range and more transmitters for data transmission.

Each cellular service area is divided into small regions called cells. Each cell contains an antenna and is controlled by a solar or AC powered network station, called the base station (BS). Each base station, in turn, is controlled by a switching office, called a mobile switching center (MSC). The MSC coordinates communication between all the base stations and the telephone central office. It is a computerized center that is responsible for connecting calls, recording call information, and billing as shown in following figure.

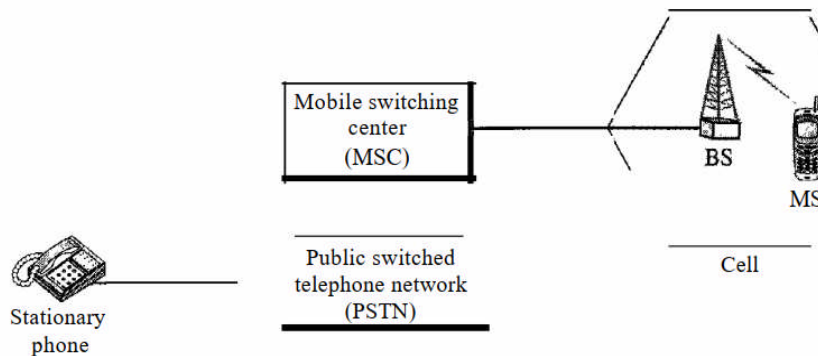


Figure 10.7: Cellular System

Features of Cellular Systems

Wireless Cellular Systems solve the problem of spectral congestion and increase user capacity. The features of cellular systems are as follows –

- Offer very high capacity in a limited spectrum.
- Reuse of radio channel in different cells.

- Enable a fixed number of channels to serve an arbitrarily large number of users by reusing the channel throughout the coverage region.
- Communication is always between mobile and base station (not directly between mobiles).
- Each cellular base station is allocated a group of radio channels within a small geographic area called a cell.
- Neighbouring cells are assigned different channel groups.
- By limiting the coverage area to within the boundary of the cell, the channel groups may be reused to cover different cells.
- Keep interference levels within tolerable limits.
- Frequency reuse or frequency planning.
- Organization of Wireless Cellular Network.
- Cellular network is organized into multiple low power transmitters each 100w or less.

Shape of Cells

The coverage area of cellular networks are divided into cells, each cell having its own antenna for transmitting the signals. Each cell has its own frequencies. Data communication in cellular networks is served by its base station transmitter, receiver and its control unit.

Handoff

It may happen that, during a conversation, the mobile station moves from one cell to another. When it does, the signal may become weak. To solve this problem, the MSC monitors the level of the signal every few seconds. If the strength of the signal diminishes, the MSC seeks a new cell that can better accommodate the communication. The MSC then changes the channel carrying the call (hands the signal off from the old channel to a new one). Hard Handoff Early systems used a hard handoff. In a hard handoff, a mobile station only communicates with one base station. When the MS moves from one cell to another, communication must first be broken with the previous base station before communication can be established with the new one. This may create a rough transition. Soft Handoff New systems use a soft handoff. In this case, a mobile station can communicate with two base stations at the same time. This means that, during handoff, a mobile station may continue with the new base station before breaking off from the old one.

Roaming

One feature of cellular telephony is called roaming. Roaming means, in principle, that a user can have access to communication or can be reached where there is coverage. A service provider usually has limited coverage. Neighbouring service providers can provide extended coverage through a roaming contract. The situation is similar to snail mail between countries. The charge for delivery of a letter between two countries can be divided upon agreement by the two countries.

10.5 SATELLITE NETWORKS

If communication takes place between any two earth stations through a satellite, then it is called as **satellite communication**. In this communication, electromagnetic waves are used as carrier signals. These signals carry the information such as voice, audio, video or any other data between ground and space and vice-versa.

In general terms, a **satellite** is a smaller object that revolves around a larger object in space. For example, moon is a natural satellite of earth.

We know that **Communication** refers to the exchange (sharing) of information between two or more entities, through any medium or channel. In other words, it is nothing but sending, receiving and processing of information.

If the communication takes place between any two earth stations through a satellite, then it is called as **satellite communication**. In this communication, electromagnetic waves are used as carrier signals. These signals carry the information such as voice, audio, video or any other data between ground and space and vice-versa

Need of Satellite Communication

The following two kinds of propagation are used earlier for communication up to some distance.

- **Ground wave propagation** – Ground wave propagation is suitable for frequencies up to 30MHz. This method of communication makes use of the troposphere conditions of the earth.
- **Sky wave propagation** – The suitable bandwidth for this type of communication is broadly between 30–40 MHz and it makes use of the ionosphere properties of the earth.

The maximum hop or the station distance is limited to 1500KM only in both ground wave propagation and sky wave propagation. Satellite communication overcomes this limitation. In this method, satellites provide **communication for long distances**, which is well beyond the line of sight.

Since the satellites locate at certain height above earth, the communication takes place between any two earth stations easily via satellite. So, it overcomes the limitation of communication between two earth stations due to earth's curvature.

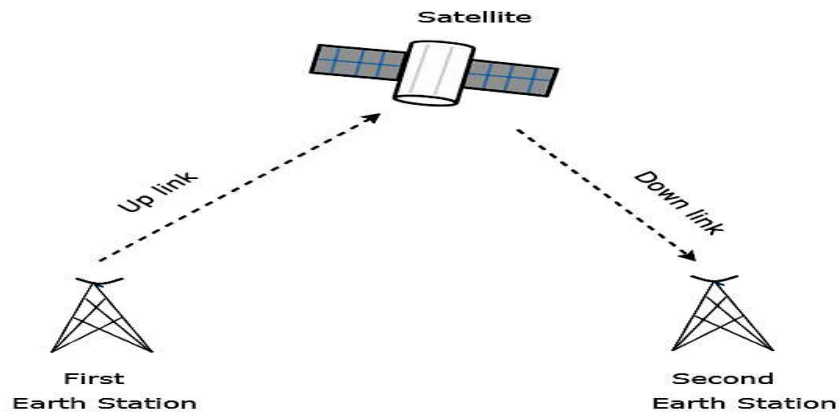


Figure 10.8: Satellite Communication

How a Satellite Works

A **satellite** is a body that moves around another body in a particular path. A communication satellite is nothing but a microwave repeater station in space. It is helpful in telecommunications, radio and television along with internet applications.

A **repeater** is a circuit, which increases the strength of the received signal and then transmits it. But, this repeater works as a **transponder**. That means, it changes the frequency band of the transmitted signal from the received one.

The frequency with which, the signal is sent into the space is called as **Uplink frequency**. Similarly, the frequency with which, the signal is sent by the transponder is called as **Downlink frequency**. The following figure illustrates this concept clearly.

Pros and Cons of Satellite Communication

In this section, let us have a look at the advantages and disadvantages of satellite communication.

Following are the **advantages** of using satellite communication:

- Area of coverage is more than that of terrestrial systems
- Each and every corner of the earth can be covered
- Transmission cost is independent of coverage area
- More bandwidth and broadcasting possibilities

Following are the **disadvantages** of using satellite communication –

- Launching of satellites into orbits is a costly process.
- Propagation delay of satellite systems is more than that of conventional terrestrial systems.
- Difficult to provide repairing activities if any problem occurs in a satellite system.
- Free space loss is more
- There can be congestion of frequencies.

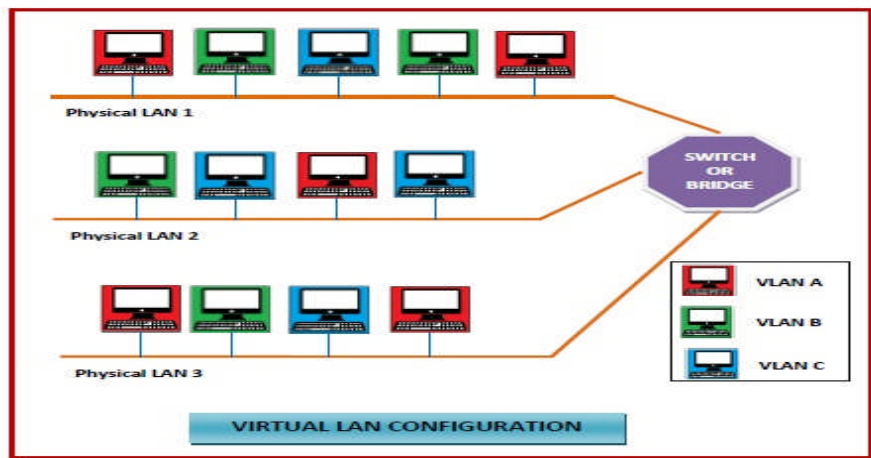
Applications of Satellite Communication

Satellite communication plays a vital role in our daily life. Following are the applications of satellite communication –

- Radio broadcasting and voice communications
- TV broadcasting such as Direct To Home (DTH)
- Internet applications such as providing Internet connection for data transfer, GPS applications, Internet surfing, etc.
- Military applications and navigations
- Remote sensing applications
- Weather condition monitoring & Forecasting

10.6 VIRTUAL LAN

Virtual Local Area Networks or Virtual LANs (VLANs) are a logical group of computers that appear to be on the same LAN irrespective of the configuration of the underlying physical network. Network administrators partition the networks to match the functional requirements of the VLANs so that each VLAN comprise of a subset of ports on a single or multiple switches or bridges. This allows computers and devices in a VLAN to communicate in the simulated environment as if it is a separate LAN.



10.6.1 Features of VLANs

- A VLAN forms sub-network grouping together devices on separate physical LANs.
- VLAN's help the network manager to segment LANs logically into different broadcast domains.
- VLANs function at layer 2, i.e., Data Link Layer of the OSI model.
- There may be one or more network bridges or switches to form multiple, independent VLANs.

- Using VLANs, network administrators can easily partition a single switched network into multiple networks depending upon the functional and security requirements of their systems.
- VLANs eliminate the requirement to run new cables or reconfiguring physical connections in the present network infrastructure.
- VLANs help large organizations to re-partition devices aiming improved traffic management.
- VLANs also provide better security management allowing partitioning of devices according to their security criteria and also by ensuring a higher degree of control connected devices.
- VLANs are more flexible than physical LANs since they are formed by logical connections. This aid is quicker and cheaper reconfiguration of devices when the logical partitioning needs to be changed.

10.6.2 Types of VLANs



Figure 10.9: Types of VLAN

- **Protocol VLAN** – Here, the traffic is handled based on the protocol used. A switch or bridge segregates, forwards or discards frames that come to it based upon the traffic's protocol.
- **Port-based VLAN** – This is also called static VLAN. Here, the network administrator assigns the ports on the switch / bridge to form a virtual network.
- **Dynamic VLAN** – Here, the network administrator simply defines network membership according to device characteristics.

10.6.3 Advantages of VLAN

There are several advantages to using VLANs:

1. **Cost and Time Reduction:** VLANs can reduce the migration cost of stations going from one group to another. Physical reconfiguration takes time and is costly. Instead of physically moving one station to another segment or even to another switch, it is much easier and quicker to move it by using software.
2. **Creating Virtual Work Groups:** VLANs can be used to create virtual work groups. For example, in a campus environment, professors

working on the same project can send broadcast messages to one another without the necessity of belonging to the same department. This can reduce traffic if the multicasting capability of IP was previously used.

3. **Security:** VLANs provide an extra measure of security. People belonging to the same group can send broadcast messages with the guaranteed assurance that users in other groups will not receive these messages.

10.7 SUMMARY

- IEEE 802.11 defines several physical layers, with different data rates and modulating techniques.
- Bluetooth is a wireless LAN technology that connects devices (called gadgets) in a small area.
- A Bluetooth network is called a Pico net. Multiple Pico nets form a network called a scatter net.
- A Bluetooth network consists of one primary device and up to seven secondary devices.
- A backbone LAN allows several LANs to be connected.
- A repeater is a connecting device that operates in the physical layer of the Internet model. A repeater regenerates a signal, connects segments of a LAN, and has no filtering capability.
- A bridge is a connecting device that operates in the physical and data link layers of the Internet model.
- A virtual local area network (VLAN) is configured by software, not by physical wiring.
- Membership in a VLAN can be based on port numbers, MAC addresses, IP addresses, IP multicast addresses, or a combination of these features.
- VLANs are cost- and time-efficient, can reduce network traffic, and provide an extra measure of security

10.8 REVIEW YOUR LEARNINGS:

1. Match the layers in Bluetooth and the Internet model.
2. Can you explain various devices like switches, gateways, bridges and routers used in network connections?
3. What are the two types of links between a Bluetooth primary and a Bluetooth secondary?

4. In multiple-secondary communication, who uses the even-numbered slots and who uses the odd-numbered slots?
5. How much time in a Bluetooth one-slot frame is used for the hopping mechanism?
What about a three-slot frame and a five-slot frame?
6. How does a VLAN save a company time and money?
7. Explain the working of Cellular Network Communication.
8. Which one has more overhead, a router or a gateway? Explain your answer.

10.9 SAMPLE QUESTIONS:

1. How does a VLAN provide extra security for a network?
2. How does a VLAN reduce network traffic?
3. What is the basis for membership in a VLAN?
4. Which one has more overhead, a bridge or a router? Explain your answer.
5. Which one has more overhead, a repeater or a bridge? Explain your answer

10.10 REFERENCES FOR FURTHER READING

- Data Communication and Networking, Behrouz A. Forouzan, Tata McGraw Hill Fifth Edition 2013
- Computer Networks, Andrew S. Tanenbaum
- <https://nptel.ac.in/courses/106/105/106105183/>
- <https://nptel.ac.in/content/storage2/courses/106105080/pdf/M5L2.pdf>



INTRODUCTION TO NETWORK LAYER

Unit Structure

11.0 Objectives

11.1 Introduction

11.2 Network Layer Services

11.2.1 Packetizing

11.2.2 Routing

11.2.3 Forwarding

11.2.4 Error Control

11.2.5 Flow Control

11.2.6 Congestion Control

11.2.7 Quality of Service

11.2.8 Security

11.3 Packet Switching

11.3.1 Datagram Approach

11.3.2 Virtual-Circuit Approach

11.4 Network-Layer Performance

11.4.1 Delay

11.4.2 Throughput

11.4.3 Packet Loss

11.4.4 Congestion Control

11.4.4.1 Open-Loop Congestion Control

11.4.4.2 Closed-Loop Congestion Control

11.5 IPv4 Addresses

11.5.1 Address Space

11.5.2 Classful Addressing

11.5.2.1 Subnetting and Supernetting

11.5.3 Classless Addressing

11.5.3.1 Slash Notation

11.5.4 Block allocation and address aggregation

11.5.5 Dynamic Host Configuration Protocol (DHCP)

11.5.6 Network Address Resolution (NAT)

11.6 Forwarding of IP Packets

11.6.1 Forwarding Based on Destination Address

11.6.2 Forwarding Based on Label

11.7 Summary

11.8 List of References

11.9 Unit End Exercise

11.0 OBJECTIVES

After going through this chapter, you will be able to

- Understand the services provided by the network layer.
- Understand different packet switching techniques that occurs at the network layer.
- Understand the factors that affect network layer performance.
- Understand how congestion can be controlled at the network layer.
- Understand how addressing is managed at the network layer.
- Understand how packets are forwarded at the network layer.

11.1 INTRODUCTION

- The network layer is the third layer in the TCP/IP reference suite model.
- It is responsible for host-to-host delivery of packets.
- It provides service to its higher layer i.e., transport layer and receives services from its lower layer i.e., data link layer.

11.2 NETWORK LAYER SERVICES

- With reference to the figure, we shall discuss the various services provided by the network layer
- The network layer is involved in source, destination and all routers in the path that govern the transmission of packet.

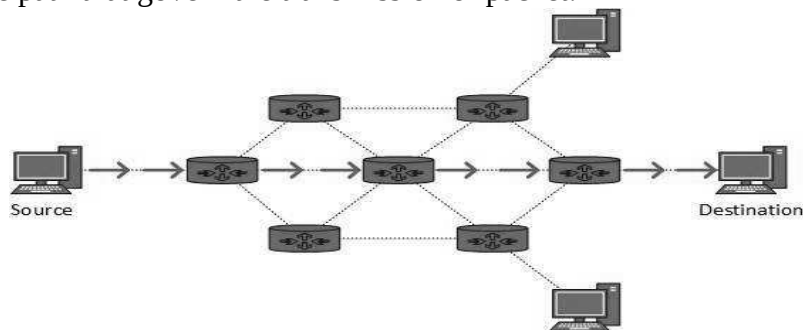


Figure 11.1 – A simple network

11.2.1 PACKETIZING

- The important service is to create packets from datagram received by the transport layer at the source and decapsulate the packet at the destination.
- The source host receives the payload from transport layer and adds a header that contains the source and destination address and extra information that is required at the network layer and forwards the packet to the data link layer.
- The destination host receives the network layer packet from its datalink layer, decapsulates the packet and forwards the payload to the transport layer.
- The routers can only fragment the packet and not allowed to change any information and add fragmentation information to the packet header.

11.2.2 ROUTING

- This service defines that network layer needs to find the best route to deliver the packet from source to destination.
- So, several algorithms and strategies are used by the network layer to find out the best route.

11.2.3 FORWARDING

- Forwarding is an activity individual router takes when the packet arrives on one of its interfaces.
- Routing decides about the entire path from source to destination but forwarding is from one router to another.
- A routing or a forwarding table is maintained at each router for smooth forwarding of the packets.
- The forwarding can be done on basis of destination address available in the packet header or based on labels assigned

11.2.4 ERROR CONTROL

- The network layer does not directly provide error control and it is handled by the higher layers.
- But it includes checksum field that checks for corruption only in the header but not the entire packet.
- However, the ICMP protocol which is an auxiliary network layer protocol provides some kind of error control mechanism.

11.2.5 FLOW CONTROL

- Flow control regulates the amount of data a source can send without overwhelming the receiver.
- The network layer does not directly provide flow control as there is no direct error control and upper layer uses the service of network layer, so double flow control mechanism would increase the complexity and reduce the efficiency of the system.

11.2.6 CONGESTION CONTROL

- Congestion in the network causes the packet to get flooded at one area and as router cannot manage that it discards the packet.
- So, error control mechanism at higher layers cause retransmission of packet and if situation worsens then it ends up no packet reaching the destination.
- The network layer applies several congestion policies to handle such situation.

11.2.7 QUALITY OF SERVICE (QoS)

- Internet allowed new multimedia communication, audio-video communication and so QoS has become vital.
- The basics of this is started in the network layer but better implementations are carried out in the higher layer.

11.2.8 SECURITY

- When Internet was designed, security was not a need of the hour.
- But as data communication grew to a wider scale, the need for security emerged.
- As network layer was already designed, we created another virtual layer to implement security called IPSec.

11.3 PACKET SWITCHING

- The network layer implements packet switching as unit of data in this layer is the packet.
- Packet Switching transmits data across the network by breaking it down into packets for more efficient transfer using various network devices.
- In order for faster transmission, the packets are fragmented and each packet takes a different route to reach the destination and then at the destination, reassembly takes place and packets are arranged in order and handed over to the upper layer.

- A packet switched network uses two different approach to route the packets namely the datagram approach and the virtual circuit approach.

11.3.1 DATAGRAM APPROACH

- In this datagram approach, the network layer is responsible for delivery of packet from source to destination.
- Packets dynamically choose route from source to destination based on the destination address.
- The packets arrive out of order and hence at the destination it is rearranged.
- In case packet size is large, the intermediate router can further fragment the packet but reassembly takes place at the destination only.
- The router maintains a simple routing or forwarding table with destination address and output interface.

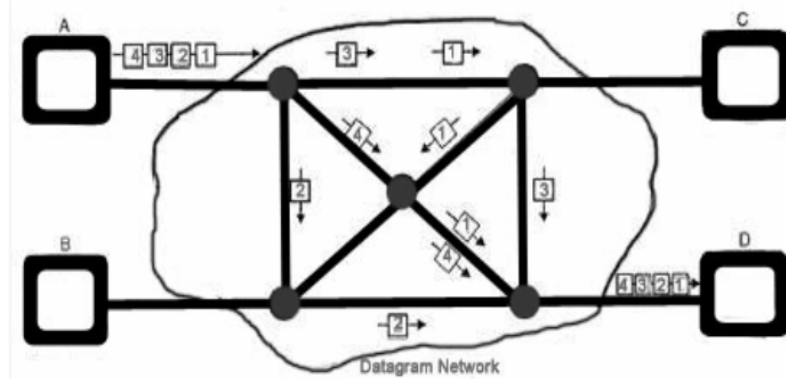


Figure 11.2 – A datagram approach – connectionless service

11.3.2 VIRTUAL-CIRCUIT APPROACH

- In the virtual circuit approach, a connection-oriented service is established between the packets and they take a fixed route to reach from source to destination.
- A logical path is established and each packet is routed based on label or VC identifier.
- The transfer of packets takes place through three phases.

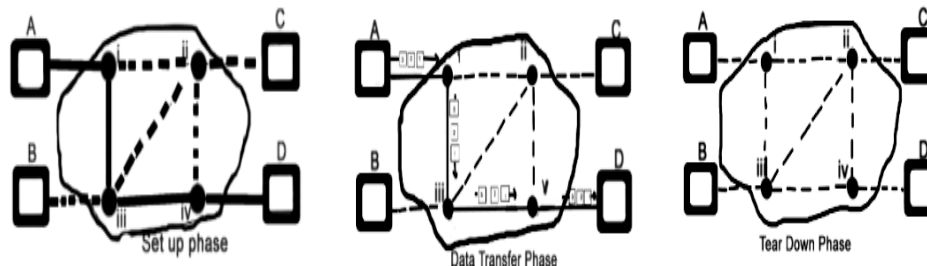


Figure 11.3 – A virtual circuit approach – connection-oriented service

- The first phase is the set-up phase where the router creates an virtual circuit entry in the routing table and tries to establish a logical path.
- It sends a request packet from source to destination informing about its VC ID.
- The packet is moved forward till it reaches the destination and each router updates it with its unique VC.
- The destination now sends the acknowledgment packet and the table is updated with switching entries.
- The second phase is data transfer phase, where all packets belonging to one message are routed in the logical route created based on the label or VC ID.
- The last phase is the tear down phase, where the source sends a special packet called tear down packet to indicate the end of packet transfer.
- The destination acknowledges with confirmation packet and all the routing entries are deleted in the logical path.

11.4 NETWORK-LAYER PERFORMANCE

- The network layer is not perfect.
- The higher layers use the service of network layer and so performance of network layer is vital.
- The performance of a network is measured in terms of delay, throughput, and packet loss.
- We can improve the performance of the network through congestion control.

11.4.1 DELAY

- A packet as it is transmitted from source to destination encounters delay and does not reach instantaneously as expected.
- The delay in the network is dependent on transmission delay, propagation delay, processing delay and queuing delay.
- All packets cannot be sent instantaneously. So, the time taken to put all packets on the link is the **transmission delay**.
- It is calculated as $\text{Delay}_{tr} = (\text{Packet length}) / (\text{Transmission rate})$.
- The time taken for a bit in a packet to travel from one point to another is called the **propagation delay**.
- It is calculated as $\text{Delay}_{pg} = (\text{Distance}) / (\text{Propagation speed})$.
- The **processing delay** is the time required for a router or a destination host to receive a packet from its input port, remove the header, perform an error detection procedure, and deliver the packet to the output port

in the case of a router or deliver the packet to the upper-layer protocol in the case of the destination host.

- It is calculated as $\text{Delay}_{pr} = \text{Time required to process a packet in a router or a destination host.}$
- The **queuing delay** for a packet in a router is measured as the time a packet waits in the input queue and output queue of a router.
- It is calculated as $\text{Delay}_{qu} = \text{The time a packet waits in input and output queues in a router.}$
- So, in a network with n routers there are $n + 1$ links and hence the delay is calculated as $\text{Total delay} = (n + 1) (\text{Delay}_{tr} + \text{Delay}_{pg} + \text{Delay}_{pr}) + n (\text{Delay}_{qu})$

11.4.2 THROUGHPUT

- Throughput is the number of bits successfully transmitted per unit time.
- The transmission time defines the bits transmitted from one point to another but throughput defines the time taken by the entire link.
- For example, consider the simple network

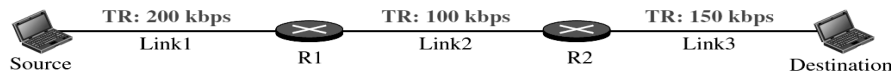


Figure 11.4 – Network with different transmission rate

- The three links have different transmission time i.e link 1 = 200 kbps, link 2 = 100 kbps and link 3 = 150 kbps.
- The throughput is calculated as the minimum transmission time i.e., **Throughput minimum {TR1, TR2, TRn}.**
- So, for the above network the throughput is 100 kbps.

11.4.3 PACKET LOSS

- The routers maintain buffers to store packets as they receive.
- They process the packet and then decide to forward on correct interface.
- The buffer size is limited and if the processing time is longer the buffer is incapable of storing the packets thereby causing the packet loss.
- Loss of packet means the packet has to be retransmitted.
- Over a period of time the network gets congested causing more packet loss and therefore proper queues have to be maintained to prevent packet loss.

11.4.4 CONGESTION CONTROL

- Congestion is defined as a state occurring in network layer when the message traffic is heavy such that it slows down the response time in the network.
- Congestion can be controlled before it happens called as open loop congestion control and remove the congestion after it has happened called closed loop congestion control.

11.4.4.1 OPEN-LOOP CONGESTION CONTROL

- Policies are applied such that congestion does not occur as prevention is a good measure.
- **Retransmission policy** and retransmission timers must be designed to avoid congestion in the network as packet loss causes the packet to be retransmitted.
- So excessive retransmission can cause heavy congestion and this can be avoided by designing optimized and efficient retransmission policy.
- A good **window policy** also helps in preventing congestion.
- A selective repeat strategy would be better choice as only corrupt packet would be retransmitted and in Go-Back N all packets from the corrupted one will be transmitted making several right packets to be retransmitted congesting the network more.
- Even an **acknowledgement policy** is vital in preventing corruption by not congesting the network for sending the acknowledgement for every packet received but sending a cumulative acknowledgment.
- The **discarding policy** adopted by the routers may prevent congestion and same time will not harm integrity of transmission as in case of audio files where we can decide to discard only less sensitive packet and thereby preserve the quality of the sound.
- Finally, **admission policy** is used to prevent congestion in virtual circuit networks where switches can check resource requirement before admitting packets in the link.

11.4.4.2 CLOSED-LOOP CONGESTION CONTROL

- Policies are applied after the congestion has occurred and alleviate the level of effect it has caused.
- In the **Backpressure technique**, the congested router informs its immediate router to stop the sending more packets and this control is carried till the upstream device.
- In **choke packet technique**, the congested router sends a special packet called choke directly to the source to slow down or stop transmission of packets till things normalize.

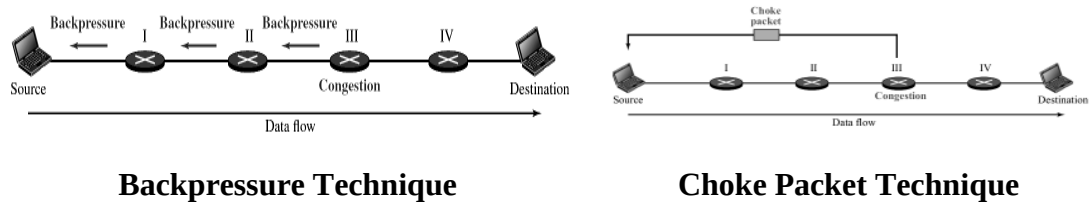


Figure 11.5 – Closed-loop Congestion Control Technique

- In the **implicit signaling technique**, the source makes assumptions that congestion has occurred as it has not received acknowledgement for long and thereby takes decision to slower the rate of packet transmission by itself.
- In the **explicit signaling technique**, the source receives a signal or message that congestion has occurred hence decides to slower the rate of packet transmission.

11.5 IPv4 ADDRESSES

- The addressing scheme used in the network layer for identifying each device is the Internet address or the IP address version 4 (IPv4).
- It is a numerical representation that uniquely identifies a specific interface on the network.

11.5.1 ADDRESS SPACE

- IPv4 uses 32-bit addresses which limits the address space to 4,294,967,296 (2^{32}) addresses.
- IPv4 address is represented 32-bit binary value and most commonly in dotted decimal notation which consist of four octets of address in decimal format separated by dots.
- Each byte (octet) is only 8 bits.
- Each number in the dotted-decimal notation is between 0 and 255.

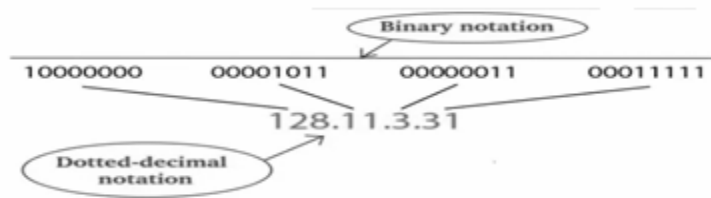


Figure 11.6 – IPv4 address representation

- Each octet represents 8-bit binary; so decimal number has to be converted to eight bit binary number.
- The IPv4 address is hierarchal in nature and is divided into two parts.



Figure 11.7 – Format of an IPv4 address

- The first part is prefix which defines the network and the second part is suffix which defines the host or the node.
- The entire IPv4 address is 32 bits long where n bits define the prefix length and $32-n$ bits define the suffix length.

11.5.2 CLASSFUL ADDRESSING

- When IPv4 was first designed, the prefix length was fixed and this scheme is called the classful addressing.
- The entire address space is divided into five classes.
- Each class has a fixed number of blocks and each block has a fixed number of hosts.

Class	Leading bits	Net ID bits	Host ID bits	No. of Networks	Address per Network	Start Address	End Address
Class A	0	8	24	$2^7 = 128$	$2^{24} = 16,777,216$	0.0.0.0	127.255.255.255
Class B	10	16	16	$2^{14} = 16,384$	$2^{16} = 65,536$	128.0.0.0	191.255.255.255
Class C	110	24	8	$2^{21} = 2,097,152$	$2^8 = 256$	192.0.0.0	223.255.255.255
Class D	1110	Not Defined	Not Defined	Not Defined	Not Defined	224.0.0.0	239.255.255.255
Class E	1111	Not Defined	Not Defined	Not Defined	Not Defined	240.0.0.0	255.255.255.255

Table 11.1 – Classful addressing scheme

- So, we can easily determine the class by identifying the leading bits in binary notation and start address in decimal notation.
- The disadvantages of classful addressing scheme are
 - In class A, the number of addresses in each block is more than enough for almost any organization and this results in wastage of addresses.
 - The same logic applies with class B resulting in wastage of addresses.
 - Whereas a block in class C is too small to fulfil the addresses requirement of an organization.
 - Each address in class D defines a group of hosts that need to multicast the address leading to wastage of address.
 - The addresses of class E are reserved for the future purpose which is also wastage of addresses.

- o Finally, we are not assigning address as per user requirements causing either shortage or excess address being assigned.
- So classful addressing leads to address depletion problem and hence better address classification schemes need to be used.

11.5.2.1 SUBNETTING AND SUPERNETTING

- To alleviate the address depletion problem, two strategies are designed namely subnetting and Super netting.
- Subnetting is a technique to divide the large network into small networks or sub-networks.
- In subnetting the network address bits are increased.
- Supernetting is a technique to combine several small networks into a large network.
- In supernetting the host addresses bits are increased.



Figure 11.8 – Subnetting technique

11.5.3 CLASSLESS ADDRESSING

- Subnetting and supernetting did not solve all of the address depletion problem.
- With growing internet requirements, more addresses were required.
- So immediate short-term solution was to use the classless addressing scheme.
- Here variable-length blocks are used that belong to no classes i.e., whole address space is divided into variable length block with block size being a power of 2 namely 2^0 , 2^1 , 2^2 2^{32} addresses.

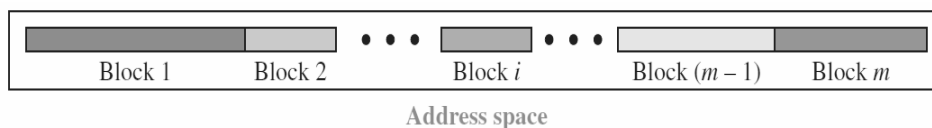


Figure 11.9 – Classless addressing scheme with variable blocks

11.5.3.1 SLASH NOTATION

- It is also referred as classless interdomain routing (CIDR) technique.
- The prefix length is added to the address separated by slash (/) as prefix length is variable and not inherent in the address.



Figure 11.10 – Slash notation representation

- So given an address we can extract three information from it.
 - o The number of addresses in the block can be found as $N = 2^{32-n}$ or $N = \text{NOT}(\text{mask}) + 1$.
 - o The first address is found by keeping the n leftmost bits and set the $(32 - n)$ rightmost bits all to 0s or by (Any address in the block) AND (mask).
 - o The last address is found by keeping the n leftmost bits and set the $(32 - n)$ rightmost bits all to 1s or by (Any address in the block) OR [(NOT (mask))].
- The first address is called the network address and routing of packets in the network is based on this address.
- So given any address, we need to find the network address and then router refers the routing or forwarding table to find the corresponding interface to forward the packet.

11.5.4 BLOCK ALLOCATION AND ADDRESS AGGREGATION

- The responsibility of block allocation is governed by Internet Corporation for Assigned Names and Numbers (ICANN).
- As per ICANN, two rules apply for proper operation of CIDR
 - o The requested address N must be power of two as $N = 2^{32-n}$ or $n = 32 - \log_2 N$ must result in integer value.
 - o The requested block should be from large contiguous location making the first address a vital parameter and this is obtained as a value which is divisible by the number of addresses in the block i.e., first address = (prefix in decimal) $\times 2^{32-n}$ = (prefix in decimal) $\times N$.
- The major advantage of CIDR notation is address aggregation also known as address summarization or route summarization.
- ICANN assigns a large block of addresses to an ISP and each ISP divides its assigned block into smaller subblocks and grants the subblocks to its customers.
- It can also be coined as many blocks of addresses are aggregated into one block and granted to one ISP.
- IPv4 has certain addresses categorized as special address used for specific purpose.

Addresses	CIDR Equivalent	Purpose	RFC	Class	# of addresses
0.0.0.0 – 0.255.255.255	0.0.0.0/8	Zero Address	RFC 1700	A	16,777,216
10.0.0.0 – 10.255.255.255	10.0.0.0/8	Private IP Address	RFC 1918	A	16,777,216
127.0.0.0 – 127.255.255.255	127.0.0.0/8	Local Loop back Address	RFC 1700	A	16,777,216
169.254.0.0 – 169.254.255.255	169.254.0.0/16	Zero Conf	RFC 3330	B	65,636
172.16.0.0 – 172.31.255.255	172.16.0.0/12	Private IP Address	RFC 1918	B	1,048,576
192.0.2.0 – 192.0.2.255	192.0.2.0/24	Documentation	RFC 3330	C	256
192.88.99.0 – 192.88.99.255	192.88.99.0/24	IPv4 to IPv6 Relay Any cast	RFC 3068	C	256
192.168.0.0 – 192.168.255.255	192.168.0.0/16	Private IP Address	RFC 1918	C	65,536
192.18.0.0 – 192.19.255.255	192.18.0.0/15	Network Device Benchmark	RFC 2544	C	131,072
224.0.0.0 – 239.255.255.255	224.0.0.0/4	Multicast	RFC 3171	D	268,435,456
240.0.0.0 – 255.255.255.255	240.0.0.0/14	Reserved	RFC1700	E	268,435,456

Table 11.2 – Special IPv4 Address

11.5.5 DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP)

- The Dynamic Host Configuration Protocol (DHCP) is a network management protocol that is used to automatically assign IP address to an organization after a block of addresses are assigned to it.
- DHCP is application layer protocol using client-server paradigm and performs the function for the network layer.
- A DHCP server dynamically assigns an IP address and other network configuration parameters such as the subnet mask, default gateway address, domain name server (DNS) address and other pertinent configuration parameters to each device on a network so they can communicate.
- The primary reason DHCP is used because it simplifies the management of IP addresses on networks as no two hosts can have the same IP address and configuring them manually leads to chances of errors and wrong allocation.
- DHCP protocol works by DHCP client sending the request message and the DHCP server replying by the response message.
- The DHCP message format is illustrated as follows

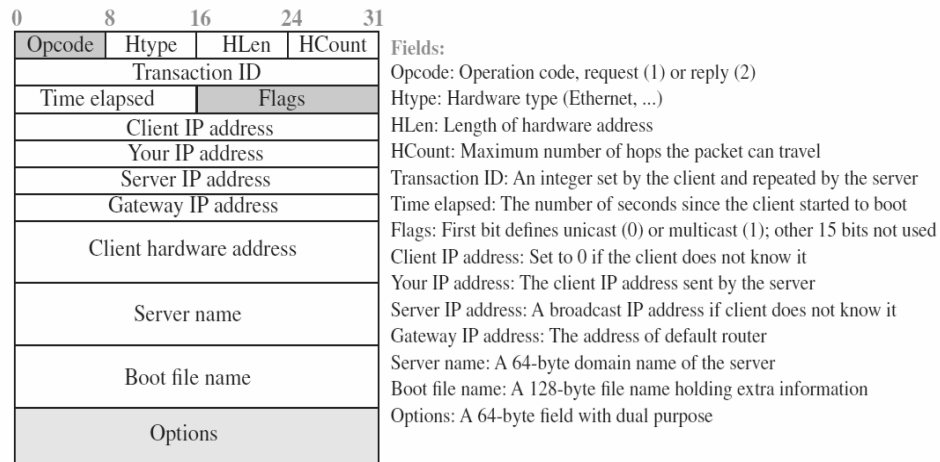


Figure 11.11 – DHCP Message Format

- There are 8 message types used in DHCP namely 1-DHCPDISCOVER, 2-DHCPOFFER, 3-DHCPREQUEST, 4-DHCPDECLINE, 5-DHCPACK, 6-DHCPNACK, 7-DHCPRELEASE, 8-DHCPINFORM.
- DHCP uses two well-known ports 67 and 68 for communication as the message is broadcast.

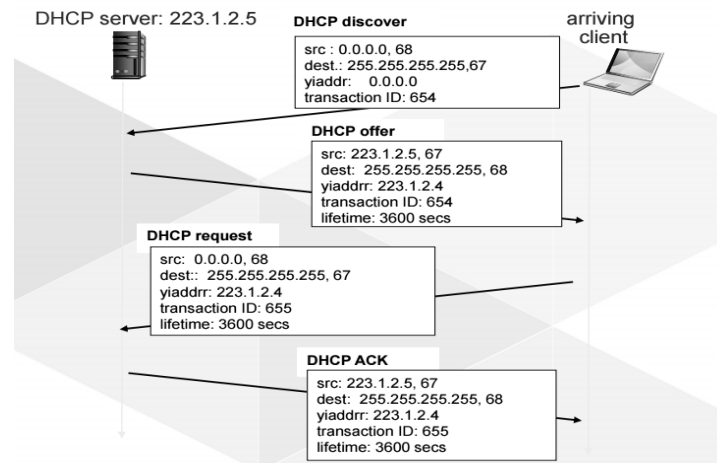


Figure 11.12 – DHCP Operation

- In addition to simplified management, the DHCP server provides benefits such as accurate IP configuration, reduced IP address conflicts, automation of IP address administration and efficient change management.
- DHCP uses services of UDP and provides error control through the implementation of checksum and retransmission policy for missing DHCP response message to the request initiated.

11.5.6 NETWORK ADDRESS RESOLUTION (NAT)

- One public address is needed to access the Internet and private addresses can be used in internal private network.
- The private network addresses can route traffic inside the network well but to access resource outside the Internet and obtain a response a public or global address is needed.
- So, the Network Address Translation (NAT) allows for easy mapping of the private and public IP address.
- This translation is done for both economic and security reasons.
- All of the outgoing packets passing the NAT router will replace the source address in the packet with the global NAT address.
- All incoming packets passing through the NAT router will replace the destination address in the packet (the NAT router global address) with the appropriate private address.

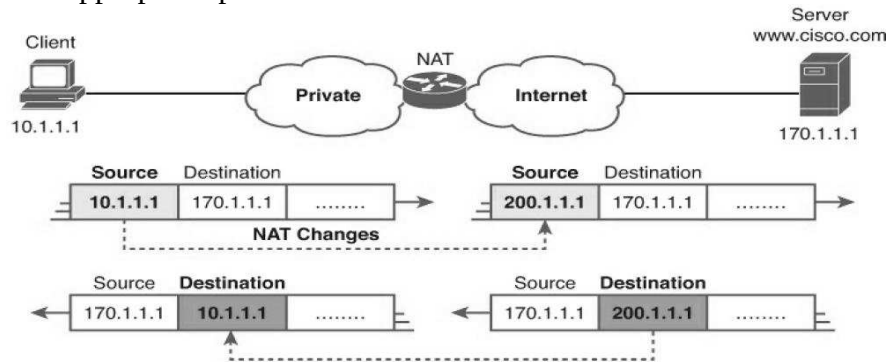


Figure 11.13 – Address Translation Process

- The translation table has two columns namely private address and destination address of the packet.
- The router translates the source address of outgoing packet to routers global address and notes it in the routing table.
- When it receives a response, it checks the routing table entry and changes the corresponding destination address which is NAT router global address with the corresponding private IP address.
- With just one global address, only one internal host can communicate with external host.
- To remove this restriction, a NAT router maintains a pool of IP addresses so that several internal host can communicate simultaneously.
- This allows many to one relationship but to have a good many to many relationships then a modified translation table makes the job easier.
- This is required when two internal host want to communicate with same external host, then additional information would help in proper translation.

Private Address	Private Port	External Address	External Port	Transport Protocol
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
:	:	:	:	:

Table 11.3 – Translation Table

- When the response from HTTP comes back, the combination of source address (25.8.3.2) and destination port address (1401) defines the private network host to which the response should be directed.
- This is feasible because the port address defined are unique.

11.6 FORWARDING OF IP PACKETS

- IP address play a vital role in the forwarding of the packets in the network.
- Forwarding a packet can be delivering a packet to the next hop which could be an intermediate router or even the destination host.
- IP is a connectionless protocol then forwarding is based on destination address and if IP works as connection-oriented protocol then forwarding is based on label.

11.6.1 FORWARDING BASED ON DESTINATION ADDRESS

- In classless addressing the entire address space is one entity as blocks are variable size.
- The /n mask along with the destination address helps in determining in which interface the packet has to be forwarded.
- So, in a classless forwarding table four pieces of information helps in forwarding decision - the mask, the network address, the interface number, and the IP address of the next router.

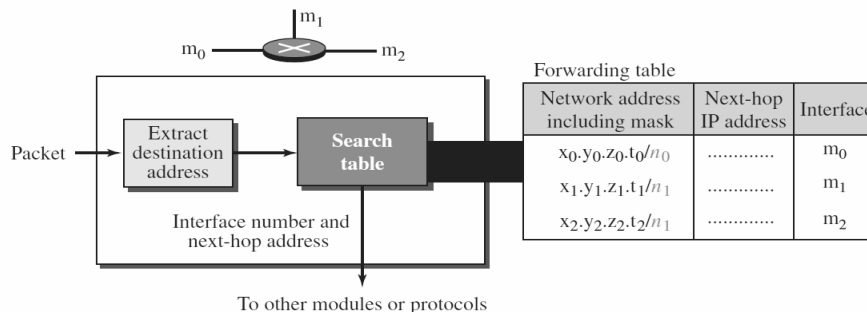


Figure 11.14 – Forwarding module in classless addressing scheme

- The table is searched row wise and each row has n leftmost bits of destination address kept and rest is set to 0s.
- Several algorithms are proposed to find the prefix.
- An easy proposal is to use the address aggregation where larger addresses are aggregated into one larger block thus minimizing the routing table entries as all packets for that network would be forwarded through that interface only.

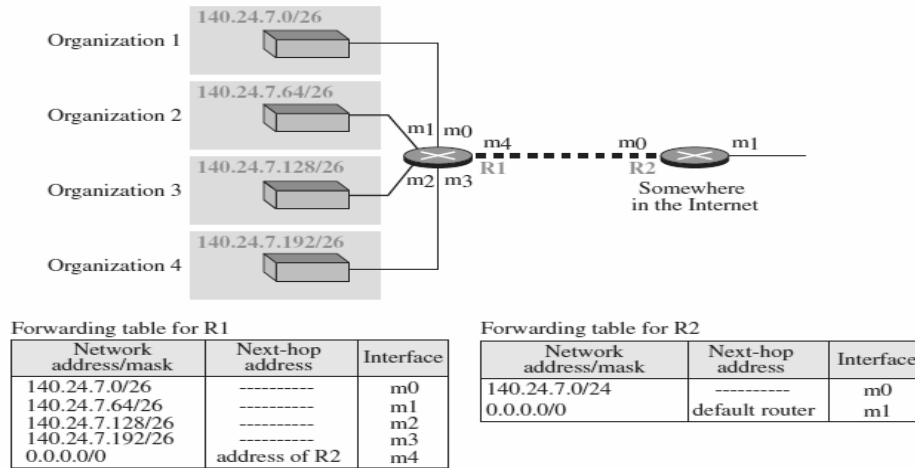


Figure 11.15 – Address aggregation Technique

- An alteration called longest mask matching is applied when an organization connected to one router moves to another organization.
- According to this principle, the forwarding table is stored as sorted from longest to shortest mask.
- In case it is not sorted then wrong mask might be applied resulting in packet being forwarded at wrong interface.

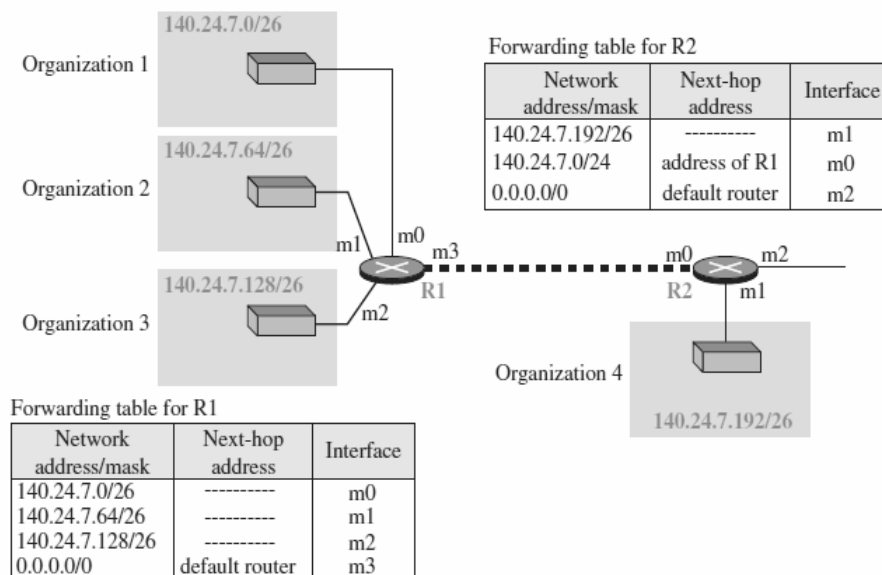


Figure 11.16 – Longest Mask Matching Technique

- Another technique to solve huge routing tables is to create hierarchy which decreases the size of the forwarding table.
- Any number of hierarchies can be created following the basic principle of classless addressing.

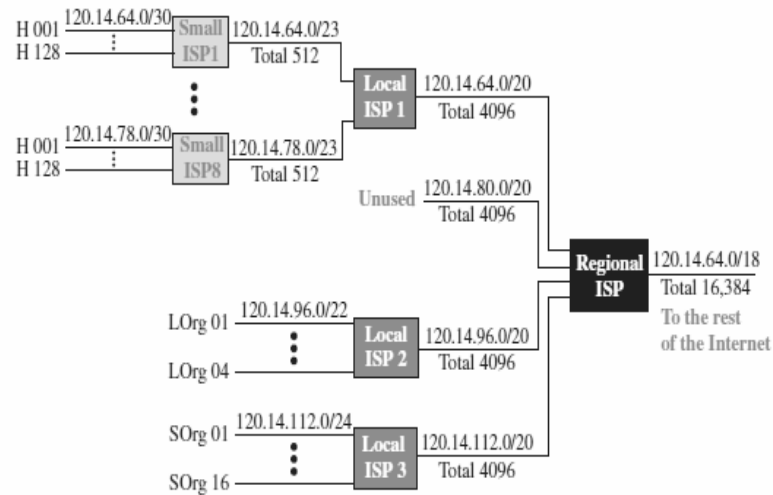


Figure 11.17 – Hierarchical Routing Technique

- Further hierarchical routing is extended to geographical routing to decrease the size of the routing table.
- The entire region is geographically divided into large blocks and each block is assigned an address and mask.

11.6.2 FORWARDING BASED ON LABEL

- When IP is used as a connection-oriented protocol, the forwarding of packets is based on a label attached to a packet.
- Here the forwarding is done by accessing a table using an index and switch helps in this forwarding decision.

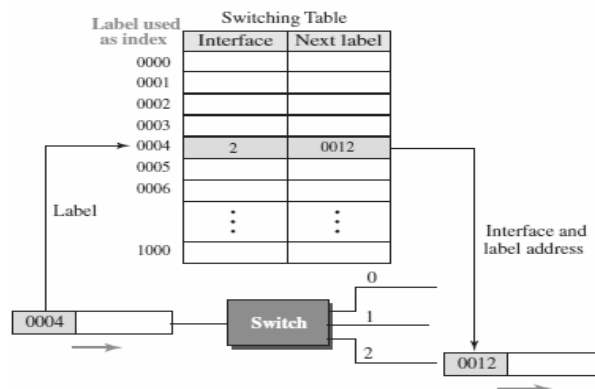


Figure 11.18 – Forwarding based on label

- The latest edition by IETF is to use Multi-Protocol Label Switching (MPLS) where a router can forward the packet based on the destination address; when behaving like a switch, it can forward a packet based on the label.
- The MPLS has a separate header attached to the IP datagram.

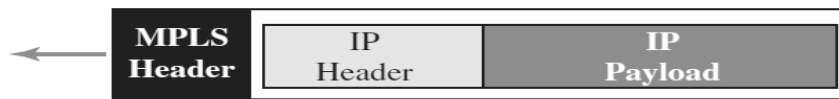


Figure 11.19 –MPLS Header

- The MPLS header is actually a stack of sub headers that is used for multilevel hierarchical switching.

11.7 SUMMARY

- The network layer in the Internet provides services to the transport layer and receives services from the network layer.
- The main services provided by the network layer are packetizing and routing the packet from the source to the destination.
- One of the main duties of the network layer is to provide packet switching.
- Performance of the network layer is measured in terms of delay, throughput, and packet loss.
- Congestion control is a mechanism that can be used to improve the performance.
- IPv4 addressing managed the communication
- Some problems of address shortage in the current version can be temporarily alleviated using DHCP and NAT protocols.
- Forwarding helps to understand how routers forward packets.

11.8 LIST OF REFERENCES

- “Data Communications and Networking” by Behrouz A. Forouzan, 5th Edition, McGraw-Hill Publication
- “Computer Networks” by Andrew Tanenbaum, 5th Edition, Pearson Education
- <https://www.javatpoint.com/network-layer>
- <https://www.paessler.com/it-explained/ip-address>

11.9 UNIT END EXERCISE

1. What are the responsibilities of Network Layer?
2. Explain the three phases in the virtual circuit approach.
3. Explain the delay in the packet switched network.

4. In classless addressing, can two different blocks can have the same prefix length? Explain.
5. Rewrite the IP address in binary notation.
 - a. 125.23.65.123
 - b. 226.36.16.244
6. Rewrite the IP address in dotted-decimal notation.
 - a. 11011111 11000000 01110101 11011101
 - b. 11101111 01011101 01110111 10000111
7. Find the class of the following classful IP address
 - a. 130.34.2.1
 - b. 01010111 1000100 10001110 00001111
8. Find the first address, last address and number of address in the block.
 - a. 200.107.16.17/18
 - b. 14.12.72.8/24
9. Combine the following three blocks of addresses into a single block:
16.27.24.0/26, 16.27.24.64/26 and 16.27.24.128/25
10. An ISP is granted the block 80.70.56.0/21. The ISP needs to allocate addresses for two organizations each with 500 addresses, two organizations each with 250 addresses, and three organizations each with 50 addresses.
 - a. Find the number and range of addresses in the ISP block.
 - b. Find the range of addresses for each organization and the range of unallocated addresses.
 - c. Show the outline of the address distribution and the forwarding table.



NETWORK LAYER PROTOCOLS

Unit Structure

12.0 Objectives

12.1 Introduction

12.2 Internet Protocol (IPv4)

12.2.1 Datagram Format

12.2.2 Fragmentation

12.2.3 Options

12.2.4 Security of IPv4 Datagram

12.3 Internet Control Message Protocol (ICMPv4)

12.3.1 Messages

12.3.1.1 Error-reporting Messages

12.3.1.2 Query Messages

12.3.2 Debugging Tools

12.4 Mobile IP

12.4.1 Addressing

12.4.2 Agents

12.4.3 Three Phases

12.4.4 Inefficiency in Mobile IP

12.5 Summary

12.6 List of References

12.7 Unit End Exercise

12.0 OBJECTIVES

After going through this chapter, you will be able to

- Identify the roles of various protocols in the network layer
- Identify the IPv4 datagram format
- Understand the concept of fragmentation
- Understand how ICMP4 helps in debugging
- Understand the need and use of Mobile IP

12.1 INTRODUCTION

- The main duty of the network layer protocols is to route the packets according to unique network device addresses and render flow and congestion control to prevent network resource depletion.
- The network layer has one main protocol namely Internet Protocol (IP) and three auxiliary protocol namely Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP) and Address Resolution Protocol (ARP).
- The IP and ICMP are available in version 4 and 6.
- The IPv4 protocol is responsible for creation of packet, forwarding, routing and delivering the packet to the destination host.
- The ICMPv4 is responsible for error handling at the network layer.
- The IGMP is responsible for multicasting in IPv4.
- The ARP is responsible for IP address to MAC address mapping and works for the data link layer though it is a network layer protocol.
- We shall learn in this chapter about the IPv4 and ICMPv4 protocol.

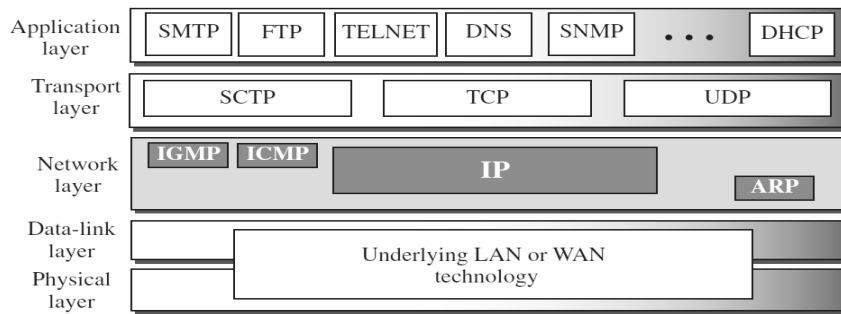


Figure 12.1 – Protocols in the TCP/IP reference suite

12.2 INTERNET PROTOCOL (IPv4)

- IPv4 is a connectionless, unreliable protocol as each datagram is handled independently and takes a different route to reach the destination.
- The network layer cannot handle large datagrams and so source fragments the datagram into smaller ones.
- The datagrams arrive out of order at destination and reassembly takes place at the destination.
- The intermediate routers can further fragment the datagram but cannot perform reassembly.
- So IPv4 handles creation, forwarding and routing of datagrams.

12.2.1 DATAGRAM FORMAT

- The important service of the network layer is to create the datagrams.
- The data obtained from higher layer are created in datagram by adding additional information required by the network layer for forwarding the packet.
- The IPv4 datagram is variable length and consist of two parts: header and data.

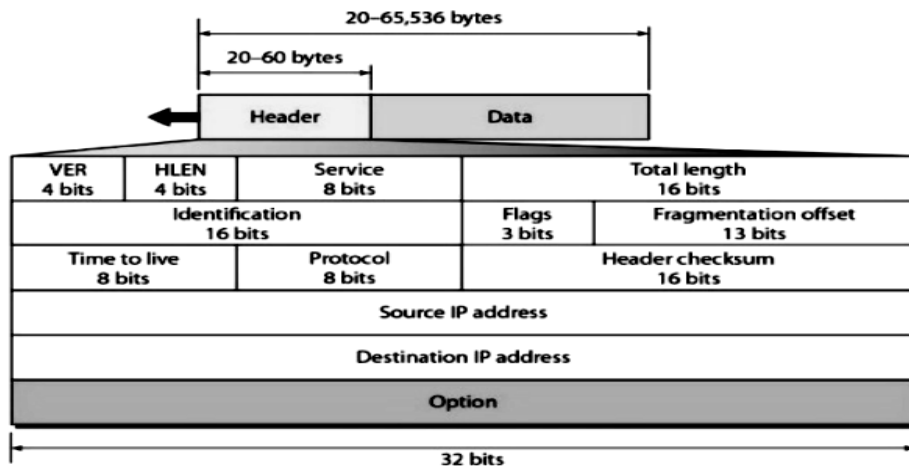


Figure 12.2 – IPv4 datagram

- The header has minimum length of 20 bytes and maximum length of 60 bytes.
- The field VER indicates the version number which is 4 bits and value is 4 i.e., 0100.
- The field HLEN defines the header length which is 4-bit field and calculated by multiplying the value of this field by 4.
- For example, if $HLEN = (0010)_2$ then $2 \times 4 = 8$ and since minimum length is 20 bytes it means this packet is corrupted. If $HLEN = (1000)_2$ then $8 \times 4 = 32$ bytes header and 12 bytes are options and packet is not corrupted.
- The field Service type is now redefined to differential service and following table defines the service available.

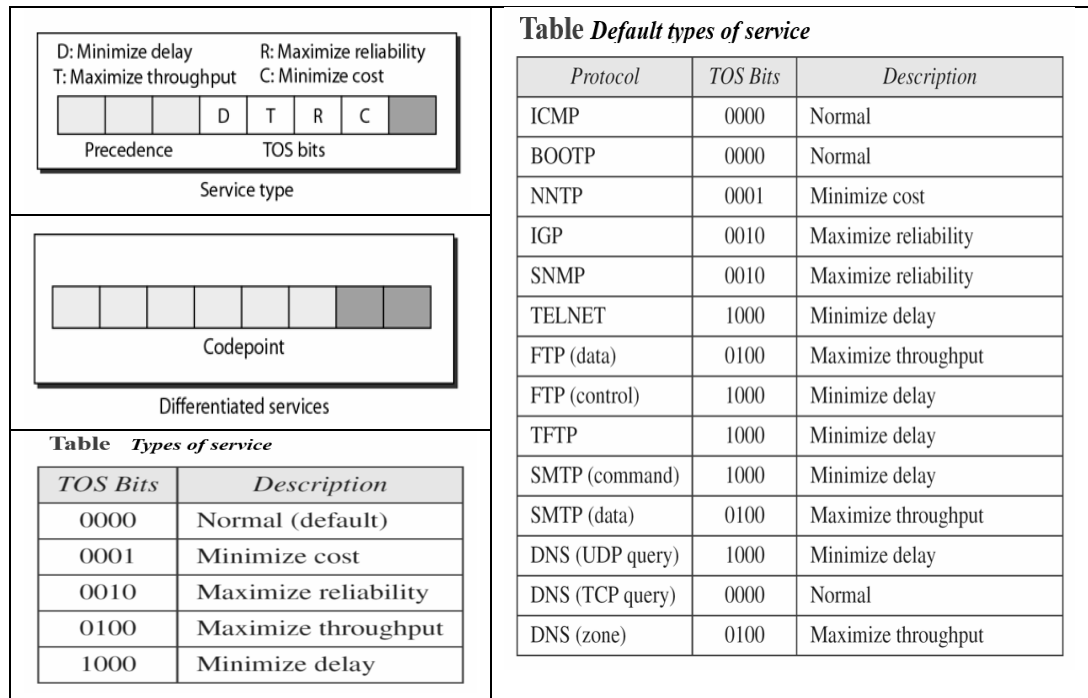


Figure 12.3–Service Type / Differential Service

- The total length field is 16 bit and it defines the total length i.e., header + data in bytes with maximum value 65535 (all 1s).
- With this we can calculate the length of the data as total length – (HLEN) x 4.
- For example, if HLEN = 5 and total length = (0028)₁₆ then length of data = 40 – 5 x 4 = 20 bytes
- The 16-bit identification field identifies uniquely the datagram created by the source. As large message is divided into smaller datagrams, each datagram belongs to the same message needs to be identified.
- The three-bit flag field has leftmost flag unused, middle flag indicating do not fragment (D) and last flag indicating if its first or last fragment.
- The 13-bit field fragmentation offset defines relative position of the fragment with respect to whole datagram.
- The time to live field indicates the number of hops i.e., routers visited by the datagram.
- The protocol field indicates the higher layer payload that is encapsulated in the datagram. In other words, the data is created from which higher layer protocol which is to be transmitted.

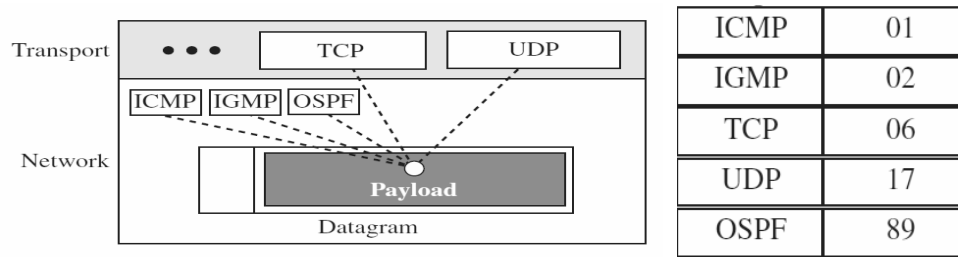


Figure 12.4– Encapsulation of data and protocol values

- The header checksum is a 16-bit field which is 1s complement of the sum of other fields to manage error checking for corrupted datagram not reaching right destination, payload reaching wrong protocol or problems during reassembly.
- The source and destination address are 32 bit and uniquely identify the sender and receiver of the datagram respectively.
- Options can be up to 40 bytes and used for network testing and debugging information.
- Data or payload is the main part of the datagram as it is message obtained for higher layer which is encapsulated and header attached for proper delivery.

12.2.2 FRAGMENTATION

- The network layer cannot handle large data and hence the data is fragmented.
- The fragmentation takes place when the when the maximum size of datagram is greater than maximum size of data that can be held a frame i.e., its Maximum Transmission Unit (MTU).
- The data flow should not be disrupted and data received from the transport layer is fragmented.
- The IP protocol is independent of physical network and the maximum length of IP datagram is 65535 bytes.
- Datagram is fragmented by source and intermediate routers and reassembly is taken care by the destination host.
- Three fields namely flag, fragmentation offset and total length are changed for the fragmentation process.
- The checksum must be calculated each time the datagram passes the router as it may further be fragmented.

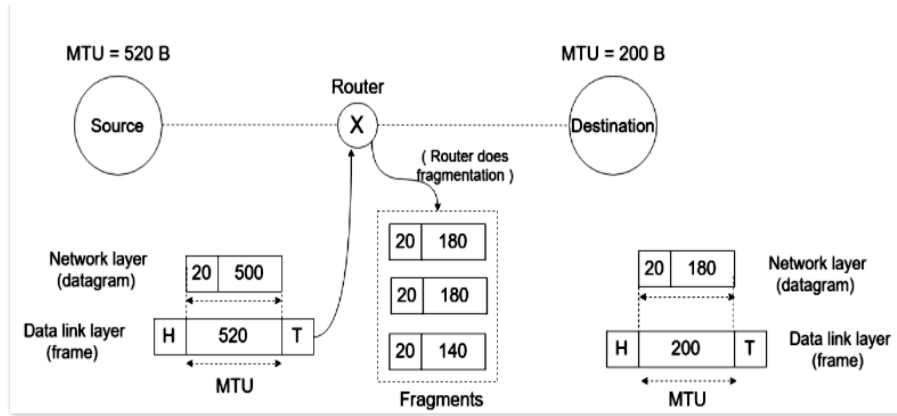


Figure 12.5– Fragmentation

- In the figure, the source creates a datagram with header size as 20 bytes and payload with 500 bytes.
- As the datagram reaches the router the, the router fragments into 3 smaller datagrams with header size remaining same for all three fragments and data divided into three sizes namely 180 bytes, 180 bytes and remaining 140 bytes.
- The destination host identifies the datagram with the identification bits in the IP header as each fragment carries the same identification number.
- The flag values indicate the fragmentation status.

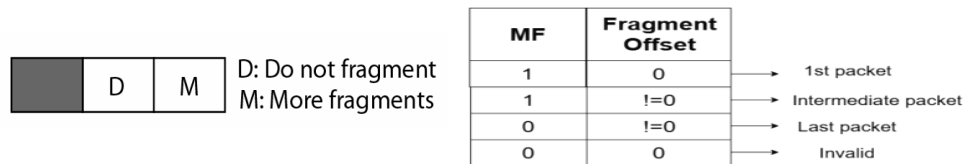


Figure 12.6 – Flags

- D means do not fragment bit. If D = 1 the router should not fragment the datagram and in that case if router cannot handle such large datagram, then it discards it and sends an ICMP error message to source. If D = 0 then datagram can be fragmented if required.
- M means more fragment bit. If M = 1, then it is not the last but first or intermediate fragment. If M = 0 then it indicates it is the last fragment,
- For the purpose of reassembly at the destination host identifies the sequence of datagram from the fragmentation offset.
- For example, the original datagram of 4000 bytes is fragmented into three fragments with bytes numbered 0 to 3999.
- The value of the offset is measured in units of 8 bytes.

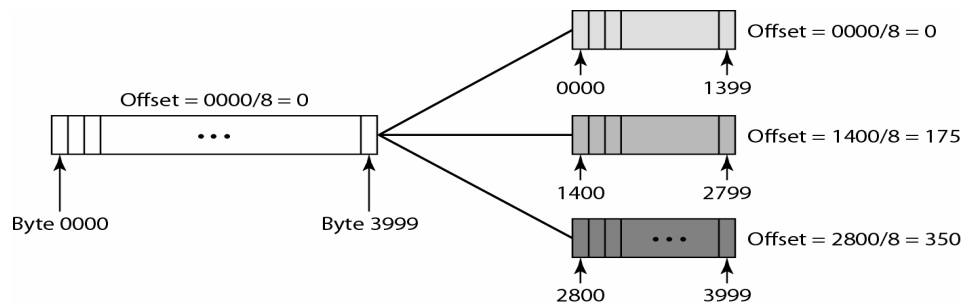


Figure 12.7–Fragmentation Scenario

- The first fragment has offset $0/8 = 0$ and carries bytes 0 to 1399.
- The second fragment has offset $1400/8 = 175$ and carries bytes 1400 to 2799.
- The last fragment has offset $2800/8 = 350$ and carries bytes 2800 to 3999.
- So, the basic strategy involved in fragmentation is
 - o The offset field for first fragment is always zero.
 - o Dividing the length of first fragment by 8 gives offset of second fragment.
 - o Dividing the total length of first and second fragment by 8 gives the offset for the third fragment.
 - o Continue the same calculation for remaining fragments.
 - o The last fragment has M set to 0.
- For example, a packet with $M = 0$ indicates it could be last fragment or the original packet was not fragmented at all. If $M = 1$, then it could be first fragment or intermediate fragment. If $M = 1$ and fragmentation offset = 0 then it is first fragment only.

12.2.3 OPTIONS

- Options provide additional information in the header for network testing and debugging and are not compulsory to be the part of the IPv4 header.
- Options can be maximum 40 bytes.

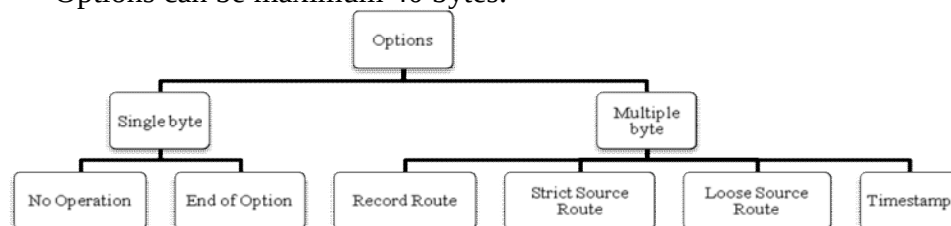


Figure 12.8 – Options in IPv4

- A No-Operation is used as filler between option and End-of-Option is used as padding at end of field.
- A record route option records addresses of up to 9 routers that handled the datagram.
- In strict source route option, the source host predetermines the route of the datagram it has to travel and the datagram has to visit only those routers mentioned in the route else it gets discarded.
- The loose source route option is similar with the variation that datagram has to visit the router listed but can also visit other routers not listed.
- The timestamp option records the time the datagram was processed by the router.

12.2.4 SECURITY OF IPv4 DATAGRAM

- When IPv4 protocol was introduced, security was not a concern.
- But with heavy data transactions and business communication, security is now a major concern.
- Three major issues applicable to security related issues are
 - Packet Sniffing – While packet is travelling from source to destination an attacker may intercept the packet, read the content and create copies of it. We cannot exactly prevent sniffing but encryption makes the content decipherable and understandable for the attacker.
 - Packet Modification – Here the attacker modifies the content of the packet and replays the packet. Using data integrity schemes, we can prevent the data from modification.
 - IP Spoofing – Here the attacker forges as the sender and sends the packet to receiver. Using data authentication schemes, we can prevent this masquerade and forgery attack.
- A new protocol called IPsec is introduced to handle the missing security aspect in IP.
- It creates a connection-oriented service between sender and receiver handling the above three attacks.
- IPSec provides four services namely defining keys and algorithms, encryption of data, data integrity and authentication.

12.3 INTERNET CONTROL MESSAGE PROTOCOL (ICMPv4)

- The IPv4 Protocol does not handle errors or correct errors.
- The router may discard the datagram because it could not find the route to the destination or TTL field has zero value.

- The IP protocol is inefficient to report and handle such problems.
- The ICMP protocol is designed to handle these problems.
- It is also a network layer protocol but the ICMP message is encapsulated into IP datagram before sending it lower layer.
- The protocol field of IP datagram is set to 1 to indicate it is an ICMP message.

12.3.1 MESSAGES

- ICMP messages are of two types: error-reporting message and query message
- The error-reporting message report the errors and problems the router and destination face while processing the datagram.
- The query message occurs in pair to place a request and obtain response of information regarding the router.
- The ICMP message has 8-byte header and variable size data section.
- The first 4 bytes of header are common in both types and next four bytes are different in both.

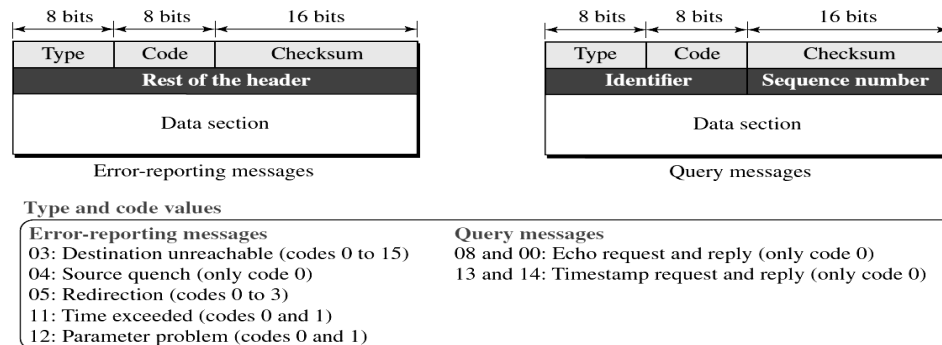


Figure 12.9 – ICMP message structure

- The type indicates the type of message and code indicates reason for message type.
- The checksum is calculated over header and data unlike the IPv4 where checksum is calculated for only header content.
- The data section in error-reporting message carries the error occurred while processing the datagram and in query message it contains information depending on the query.

12.3.1.1 ERROR-REPORTING MESSAGES

- The main role of ICMP is to report error during the processing of datagram as IP is unreliable and incapable of doing so.
- ICMP cannot correct the errors but only informs about the errors.

- The higher layers need to take care of the error correction.
- Errors are reported to source host using the source IP address available in the header.
- ICMP cannot directly float the error message in the network and so it forms an error packet and encapsulates it in the IP datagram.

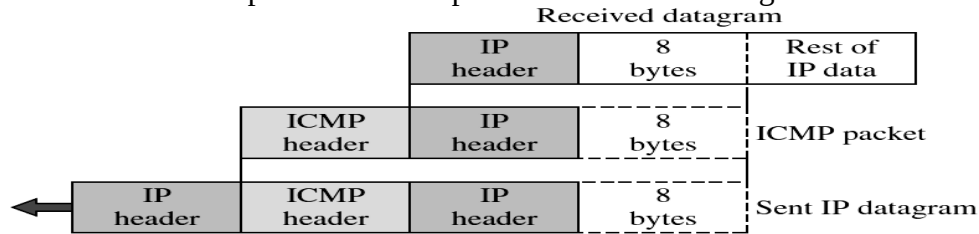


Figure 12.10 – Encapsulation of ICMP packet in IP datagram

- The various error reporting message are
 - **Destination Unreachable** – It is sent by a router when it cannot deliver an IP datagram.
 - **Source Quench** – It is sent by a destination host or router if it is receiving data too quickly and not able to handle the datagram. The message is a request that the source slow down datagram transmission
 - **Redirection**- It is sent by a router to optimize network traffic by redirecting the datagram to another router if it receives a datagram that should have been sent to a different router.
 - **Time Exceeded** – It is sent by a router if the datagram has reached the maximum limit of routers through which it can travel.
 - **Parameter Problem** - It is sent by a router if a problem occurs during the transmission of a datagram such that it cannot complete processing due to invalid header.

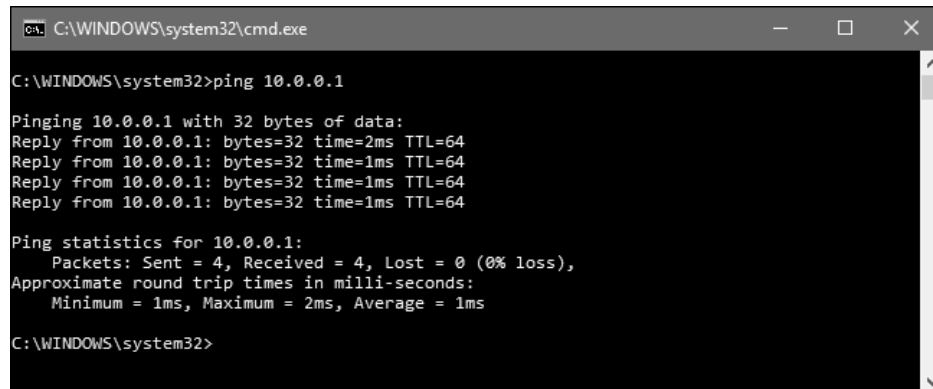
12.3.1.2 QUERY MESSAGES

- Query message are independent of IP datagram but again need to be encapsulated in a datagram as a carrier.
- Query message comes in pair and are used to test the availability and actives ness of router in the network.
- The various query messages are
 - **Echo Request & Echo Reply** – It is used to test destination accessibility and status. A host sends an Echo Request and listens for a corresponding Echo Reply.
 - **Timestamp Request & Timestamp Reply** – It is used to synchronize the clocks between hosts and to estimate transit time.

- The following query messages are now deprecated and replaced by other messages.
 - **Information Request & Information Reply** - These messages were used earlier by hosts to perform the mapping of IP and MAC address. It is now taken care by the Address Resolution Protocol (ARP).
 - **Address Mask Request & Address Mask Reply** – These messages were used to find the mask of the subnet. A host sends an Address Mask Request to a router and receives an Address Mask Reply in return. It is now taken care by the Dynamic Host Configuration Protocol (DHCP).
 - **Router Advertisement and Router Solicitation** – These messages were used to allow hosts to discover the existence of routers. Routers periodically broadcast their IP addresses via Router Advertisement messages. Hosts may also request a router address by broadcasting a Router Solicitation message to which a router replies with a Router Advertisement. It is now taken care by the Dynamic Host Configuration Protocol (DHCP).

12.3.2 DEBUGGING TOOLS

- ICMP makes use of two important tools for debugging purpose – ping and traceroute
- The ping is a computer network administration software utility that is used to test the reachability of a host on an Internet Protocol network.
- It sends an ICMP echo request packet to the destination.
- If the destination is alive, it prompts with the echo reply message.
- The syntax is ping and the IP address of the destination.



```

C:\WINDOWS\system32\cmd.exe

C:\WINDOWS\system32>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=2ms TTL=64
Reply from 10.0.0.1: bytes=32 time=1ms TTL=64
Reply from 10.0.0.1: bytes=32 time=1ms TTL=64
Reply from 10.0.0.1: bytes=32 time=1ms TTL=64

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\WINDOWS\system32>
  
```

Figure 12.11 – ping command output

- The traceroute command in UNIX and tracert command in Windows is used to determine the path taken to a destination by sending ICMP echo request to the destination with incrementally increasing time to live (TTL) field values.

- Each router decrements the TTL count as it forward the packet.
- When TTL = 0, then router throws an ICMP time exceeded message to the source host.
- In case the path is not found then ICMP destination unreachable message is also sent to the source.
- The syntax is tracert and IP address of the destination.

```

C:\Users\Matt>tracert 8.8.8.8

Tracing route to google-public-dns-a.google.com [8.8.8.8]
over a maximum of 30 hops:

  0  <1 ms    <1 ms    <1 ms    192.168.10.254
  1  4 ms      7 ms      1 ms     n41-akl-internet.mdr-bng1.as45177.net.nz [14.1.43.222]
  2  1 ms      1 ms      1 ms     ae3-1303.mdr-cr1.as45177.net.nz [120.136.0.131]
  3  24 ms     24 ms     25 ms     xe-4-0-1-0.sy3-cr1.as45177.net.au [120.136.0.118]
  4  24 ms     24 ms     24 ms     as15169-ip-119.cust.sy3-cr1.as45177.net.au [120.136.0.119]
  5  25 ms     25 ms     25 ms     216.239.40.233
  6  25 ms     25 ms     25 ms     216.239.40.255
  7  25 ms     25 ms     25 ms     google-public-dns-a.google.com [8.8.8.8]

Trace complete.

C:\Users\Matt>

```

Figure 12.12 – tracert command output

12.4 MOBILE IP

- Mobile IP is extension of IP protocol in mobile that are connected to the Internet.
- It is an Internet Engineering Task Force standard communications protocol that is designed to allow mobile device users to move from one network to another.
- In the original network the host are stationary and belong to one specific network.
- So, assigning of IP address is simple defining the prefix and suffix and communication takes place with this address as its valid and host belongs to that network.
- If the host changes the network, then address will become invalid.
- As the name implies, the mobile host moves from network to network and normal addressing scheme will not work as an address valid in one network will not be valid in another network.
- So different scheme is required for mobile host.

12.4.1 ADDRESSING

- The mobile host changes its address as it moves from one network to another.

- The host makes use of DHCP to obtain new address as it moves to a new network.
- But using DHCP leads to four problems:
 - o Need to update the configuration file whenever new address is obtained.
 - o The system needs to be rebooted as it moves to new network.
 - o The DNS tables needs to revised to make the Internet aware of the network changes.
 - o During data transmission, if the host moves from one network to another then data exchange is interrupted.
- So viable solution is to use two addresses – home address and care-of address.
- The original address is called home address and is permanent and host associates with home network.
- The care-of address is temporary and when it associates host with foreign network i.e., the other network the host moves in.

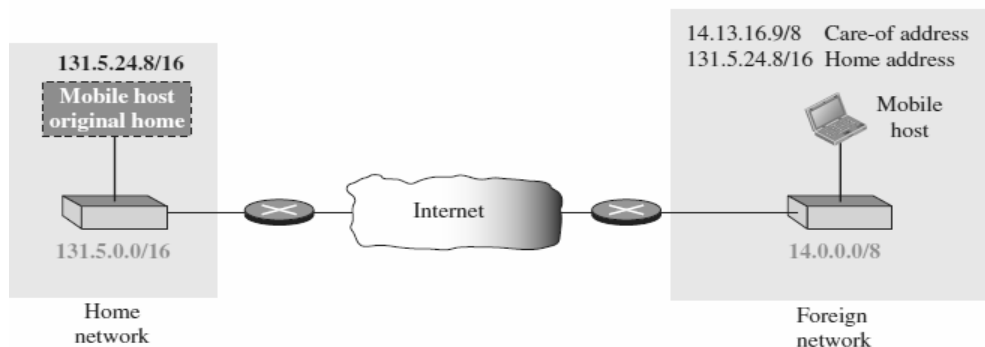


Figure 12.13–Home address and Care-of address of Mobile Host

12.4.2 AGENTS

- To manage the home address and care-of address, we require the home agent and foreign agent.
- The router attached to home network of the mobile host is the home agent.
- The router attached to foreign network of the mobile host is called the foreign agent.

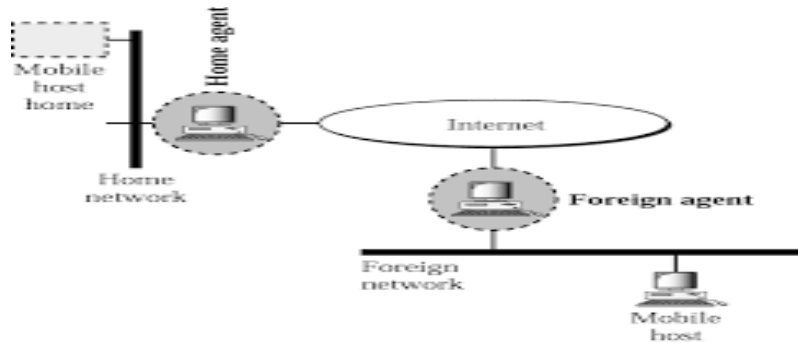


Figure 12.14 – Home agent and foreign agent

- The mobile host and foreign agent can be same and in that case the mobile host obtains the care-of address through DHCP.
- This care-of address is called collocated care-of address.
- The advantage of dual addressing is easy movement from one network to another and disadvantage is extra software required for processing.

12.4.3 THREE PHASES

- The communication of mobile host with remote host goes through three phases.

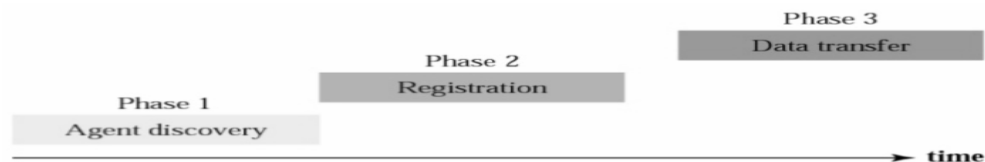


Figure 12.15 – Phases in remote communication

- The first and second phase involves the mobile host and two agents whereas the last phase involves even the remote host.
- The entire communication is a nine-step process.
- Step1 – A mobile host must know its home address and so it requests the home agent by sending the agent solicitation message.
- Step 2 - The home agent responds with the address through the agent advertisement message.
- Step 3 – The same process is to be repeated when the mobile host moves to foreign network and sends an agent solicitation message to the foreign agent.
- Step 4 – The foreign agent responds with the care-of address through the agent advertisement message.
- Step 5 – The mobile host needs to register to foreign agent via the registration request message.

- Step 6 – The mobile host also needs to register with the home agent. This is done by foreign agent on behalf of mobile host via the registration request message.
- Step 7 – The home agent replies to the foreign agent via the registration response message.
- Step 8 – This registration response message is relayed back to the mobile host by the foreign agent.
- Step 9 – The mobile host can now transfer data and communicate with the remote host.

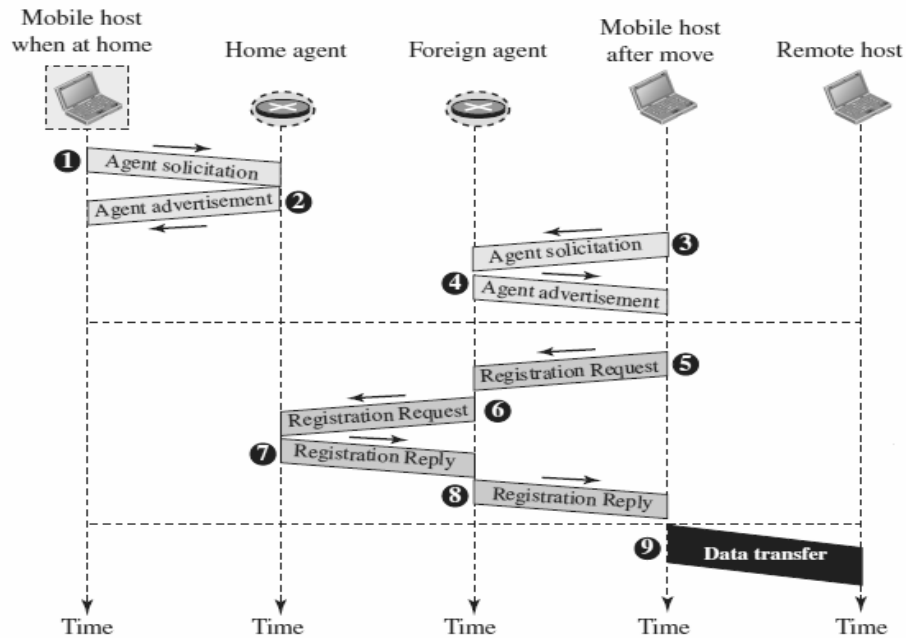


Figure 12.16 – Mobile Host and Remote Host Communication

- The format of the various messages exchanges in the communication are discussed below.
- The router solicitation message in ICMP is used in the place of **agent solicitation message**.
- The **agent advertisement message** is piggy backed with the router advertisement packet.
- Through this message the router advertises its presence on the network.

ICMP Advertisement message				Bit	Meaning
Type	Length	Sequence number		0	Registration required. No collocated care-of address.
Lifetime		Code	Reserved	1	Agent is busy and does not accept registration at this moment.
Care-of addresses (foreign agent only)				2	Agent acts as a home agent.
				3	Agent acts as a foreign agent.
				4	Agent uses minimal encapsulation.
				5	Agent uses generic routing encapsulation (GRE).
				6	Agent supports header compression.
				7	Unused (0).

Figure 12.17 – Agent Advertisement Message

- The type field has value 16.
- The length field indicates the total length of the extension message.
- The sequence number field is used to map in case the message is lost.
- Lifetime field indicates the value in seconds for the agent to accept request. For infinite lifetime the value is all 1s.
- The various value for code field is tabulated.
- The last field is used only by the foreign agent where is a list of care-of address is available and finalized in the registration phase.
- The mobile host sends the **registration request message** to foreign agent and home agent to register the home address, home network and care-of address.

Type	Flag	Lifetime	Bit	Meaning
Home address			0	Mobile host requests that home agent retain its prior care-of address.
Home agent address			1	Mobile host requests that home agent tunnel any broadcast message.
Care-of address			2	Mobile host is using collocated care-of address.
Identification			3	Mobile host requests that home agent use minimal encapsulation.
			4	Mobile host requests generic routing encapsulation (GRE).
			5	Mobile host requests header compression.
Extensions ...			6-7	Reserved bits.

Figure 12.18 – Registration Request Message

- The Type field has the value 1 and the flag field defines forwarding information which is tabulated.
- The lifetime field defines number of seconds the registration is valid and all 0s indicate for deregistration and all 1s indicate infinite lifetime.
- Next set of fields indicate the home address, home agent address and care-of address.
- The identification is a unique value set in request message and repeated in reply message for matching purpose.
- The extensions are used for authentication purpose.

- The **registration reply message** is sent in response to the request which the home agent sends to foreign agent and which is relayed to the mobile host regarding confirming or denying the registration request.

Type	Code	Lifetime
Home address		
Home agent address		
Identification		
Extensions ...		

Figure 12.19 – Registration Response Message

- The Type field has the value 3 and code field replaces the flag field with either acceptance or denial of request. The remaining fields are same as registration request message.
- The registration request and reply message are encapsulated inside a UDP datagram with agent using well known port no 434 and mobile host using ephemeral port number.
- The data transfer process occurs in four steps.

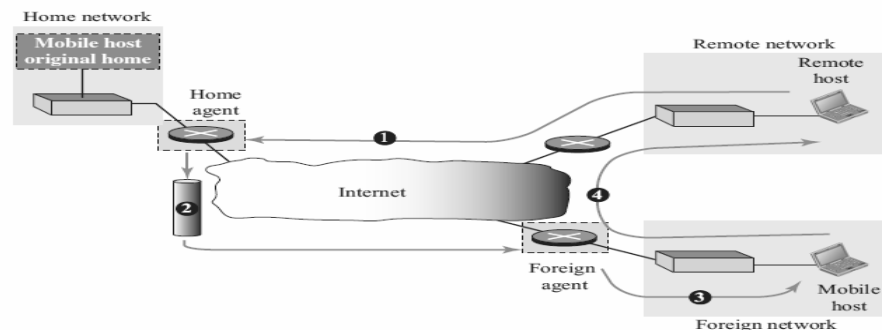


Figure 12.20 – Data transfer between remote host and mobile host

- Step 1 –When the remote host wants to send an IP packet, it creates a packet with source address as its address and home address of mobile host as destination address. The home agent intercepts this packet using proxy ARP technique.
- Step 2 –The home agent creates a special tunnel and forwards the packet to the foreign agent by encapsulating the original IP packet into another packet with its address as source address and foreign agent address as destination.
- Step 3 –When the foreign agent receives the packet it checks the table entry and replaces the home address with care-of address and forwards the packet to the mobile host

- Step 4 – The mobile host responds directly by creating an IP packet with its home address as source address and remote host as destination address which the foreign agent handles.
- The entire data transfer process is transparent.

12.4.4 INEFFICIENCY IN MOBILE IP

- Every communication has flaws and Mobile IP can be inefficient for two reasons.
- The severe case of inefficiency is called the double crossing.
- In situation where remote host and mobile host are associated with the same router, so rather than internal communication taking place, the communication takes place as a long path.

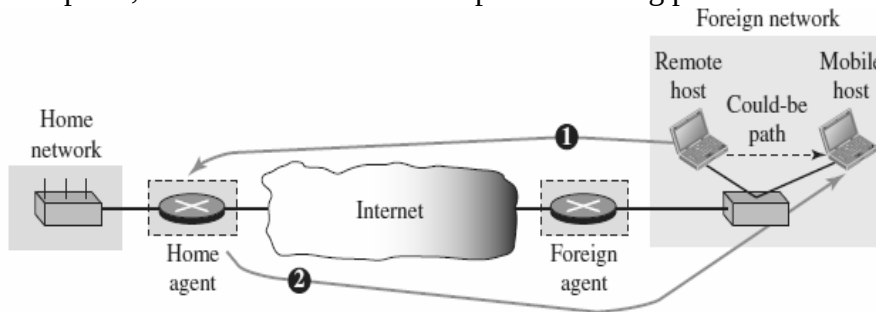


Figure 12.21 - Double Crossing

- So rather than having local communication, the communication is spread across the Internet and same message crosses twice.
- The moderate case of inefficiency is the triangle routing.
- The remote host could be in closer proximity with the mobile host and rather than direct communication, the communication is two-fold as packet goes first to home agent and then to mobile host.

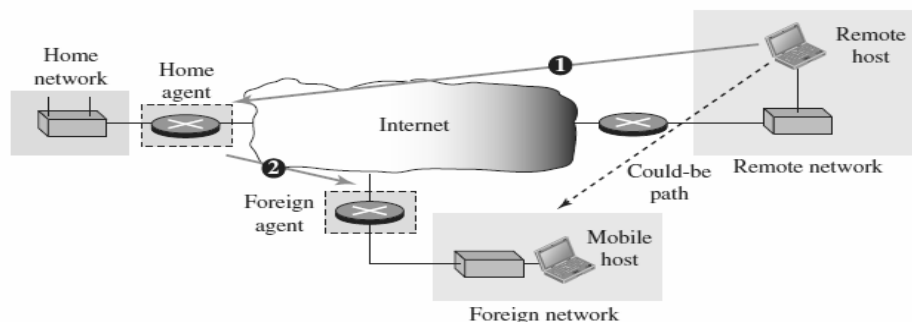


Figure 12.22 – Triangle Routing

- Solution for inefficiency is the remote host needs to bind the care-of and home address of the mobile host.
- Doing so will reduce the traversal of packet depending on the location of the mobile host currently.

12.5 SUMMARY

- The Internet Protocol version 4 (IPv4) is an unreliable connectionless protocol responsible for host-to-host delivery of datagrams
- An IPv4 datagram can be fragmented by source or intermediate router one or more times during its path from the source to the destination but the reassembly of the fragments is done at the destination only.
- The Internet Control Message Protocol version 4 (ICMPv4) supports the unreliable and connectionless Internet Protocol (IP) by handling errors during transmission.
- The error reporting message and query message helps the IPv4 for smooth transition.
- The Mobile IP designed for mobile communication is an enhanced version of the Internet Protocol (IP).

12.6 LIST OF REFERENCES

- “Data Communications and Networking” by Behrouz A. Forouzan, 5th Edition, McGraw-Hill Publication.
- <https://www.javatpoint.com/network-layer-protocols>
- <https://www.geeksforgeeks.org>

12.7 UNIT END EXERCISE

1. A host is sending 100 datagrams to another host. If the identification number of the first datagram is 1024, what is the identification number of the last?
2. In an IPv4 datagram, the value of the header-length (HLEN) field is $(6)_{16}$. How many bytes of options have been added to the packet?
3. What are the source and destination IP addresses in a datagram that carries the ICMPv4 message reported by a router?
4. An IP datagram has arrived with the following partial information in the header (in hexadecimal): 45000054 00030000 2006...
 - a. What is the header size?
 - b. Are there any options in the packet?
 - c. What is the size of the data?
 - d. Is the packet fragmented?
 - e. How many more routers can the packet travel to?
 - f. What is the protocol number of the payload being carried by the packet?
5. Determine if a datagram with the following information is a first fragment, a middle fragment, a last fragment, or the only fragment (no fragmentation):
 - a. M bit is set to 1 and the value of the offset field is zero.
 - b. M bit is set to 1 and the value of the offset field is nonzero.



UNICAST ROUTING

Unit Structure

13.0 Objectives

13.1 Introduction

13.2 Internet as a Graph

13.2.1 Least-cost Routing

13.3 Routing Algorithms

13.3.1 Distance-Vector Routing

13.3.2 Link-State Routing

13.3.3 Path-Vector Routing

13.4 Unicast Routing Protocols

13.4.1 Internet Structure

13.4.2 Routing Information Protocol (RIP)

13.4.3 Open Shortest Path First (OSPF)

13.4.4 Border Gateway Protocol Version 4 (BGP4)

13.5 Summary

13.6 List of References

13.7 Unit End Exercise

13.0 OBJECTIVES

After going through this chapter, you will be able to

- Understand the general concept of unicast routing
- Understand different unicast routing algorithms
- Understand the unicast routing protocols

13.1 INTRODUCTION

- The network layer is responsible for host-to-host delivery of packets.
- The unicast routing governs the transmission of packet to only one destination (one to one) whereas multicast routing governs the transmission of packet to several destinations (one to many).
- Unicast routing can be implemented using hierarchical routing where we route in steps.

13.2 INTERNET AS A GRAPH

- The packet is routed hop by hop from source to destination.
- The forwarding tables are used for the routing process.
- Source host requires no forwarding table reference as it passes onto default router.
- Destination host needs no forwarding table as it receives the packet and it has to reassemble the packets.
- The intermediate router uses the forwarding table and needs to decide the best route to reach the destination.
- So, the internet can be modeled as a graph. i.e., weighted graph.
- A graph consists of nodes and edges, so the source, destination and router represent nodes and links represent edges.
- Consider an internet with 7 routers which has been converted to weighted graph.

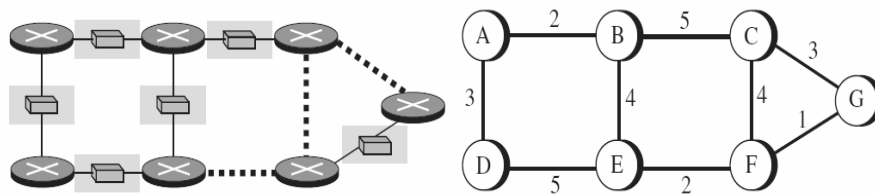


Figure 13.1 - Internet modeled as a Graph

- The weight represents the cost taken to travel from one node to another.
- If there is no edge then cost represent infinity.
- The cost has different interpretation for different routing algorithm.

13.2.1 LEAST-COST ROUTING

- In order to find the best route from source to destination we can create least cost trees from a weighted graph.
- A least cost tree chooses one node as a root node, and finds the best path from that node to all other node.
- The best path is chosen on the basis of least cost to reach from that source to destination.

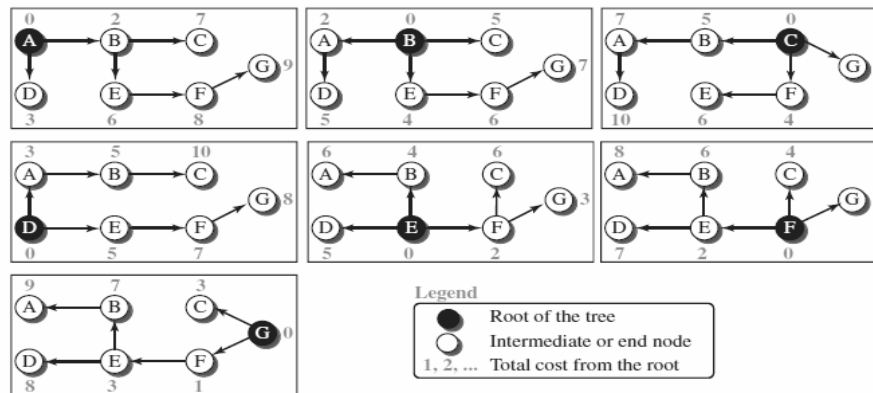


Figure 13.2 - Least Cost Trees for Internet

- The basic principle of least cost trees is that if there are N routers in the Internet, then there are $(N - 1)$ least cost paths from one router to all routers in the network.
- On the whole there are $N \times (N - 1)$ least cost paths for the entire network.
- A better way to visualize is to create a least cost tree.
- We consider one source router and find the shortest path to all the other routers in the network.
- Two important properties to be remembered while constructing a least cost tree are:
 - A least cost tree from router X to router Y in X 's tree is inverse in Y 's tree. For example, route from A to C in A's tree is $A \rightarrow B \rightarrow C = 7$ and in C's tree is $C \rightarrow B \rightarrow A = 7$.
 - The cost involved in travelling from router X to router Z in X 's tree is equal to cost travelling from router X to router Y in X 's tree plus cost travelling from router Y to router Z in Y 's tree. For example, the route from A to F in A's tree is $A \rightarrow B \rightarrow E \rightarrow F = 8$ and route from A to E in A's tree = is $A \rightarrow B \rightarrow E = 6$ and E to F in E's tree $E \rightarrow F = 2$ i.e., $6 + 2 = 8$.

13.3 ROUTING ALGORITHMS

- Routing algorithms are meant for determining the routing of packets in a node.
- Several routing algorithms have been devised.
- The difference in the various algorithms is the interpretation of cost and construction of least cost trees

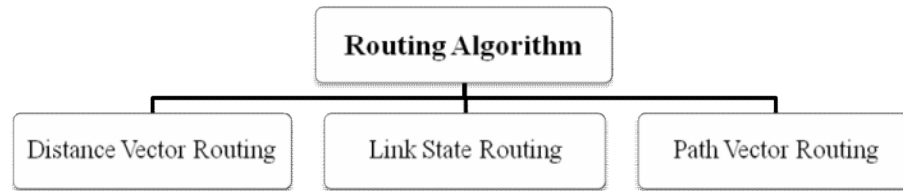


Figure 13.3 – Types of Routing Algorithms

13.3.1 DISTANCE-VECTOR ROUTING

- The Distance Vector algorithm is a dynamic algorithm that determines the best route for data packets based on distance.
- It uses the Bellman–Ford algorithm to calculate the best route.
- This algorithm finds the shortest route between source X and destination Y through intermediary nodes A, B, C.... where least cost (distance) is involved.
- The general case in which D_{ij} is the shortest distance and c_{ij} is the cost between nodes i and j, the equation is $D_{xy} = \min \{(c_{xa} + D_{ay}), (c_{xb} + D_{by}), (c_{xc} + D_{cy}), \dots\}$.
- This can be simplified as let $d_x(y)$ be the cost of the least-cost path from node x to node y then $d_x(y) = \min_v \{c(x,v) + d_v(y)\}$ where the $\min_v$ is the equation taken for all x neighbors. After traveling from x to v, if we consider the least-cost path from v to y, the path cost will be $c(x,v) + d_v(y)$. The least cost from x to y is the minimum of $c(x,v) + d_v(y)$ taken over all neighbors.
- Each router maintains a distance table known as distance vector.
- Consider a network with 4 routers (node) and cost (distance) on its edges.

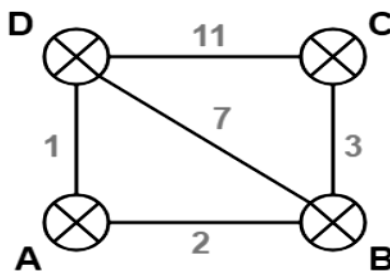


Figure 13.4 – Weighted Graph

- In the first step, each router creates a distance vector of its neighboring routers.
- The cost to self-node is 0 and cost to directly connected node (neighbor) is obtained from the figure and cost to not directly connected node is ∞ .

Distance Vector of A		Distance Vector of B		Distance Vector of C		Distance Vector of D	
A	0	A	2	A	∞	A	1
B	2	B	0	B	3	B	7
C	∞	C	3	C	0	C	11
D	1	D	7	D	11	D	0

Table 13.1 – Distance Vector of all routers

- Next the nodes exchange this information with its neighbor to update the missing information.
- For example, when router A receives distance vectors from its neighbors B and D, then new distance vector is calculated for A.
- Router A can reach the router B via its neighbor B or D and needs to choose the path which gives the minimum cost.
- Cost of reaching router B from A via neighbor B = Cost (A→B) + Cost (B→B) = 2 + 0 = 2
- Cost of reaching router B from A via neighbor D = Cost (A→D) + Cost (D→B) = 1 + 7 = 8
- Since the cost is minimum via neighbor B, so router A chooses the path via B.
- Similarly, we calculate the shortest path distance to each destination router at every router.
- Summarizing the least cost to all nodes now
 - o Cost of reaching destination B from A = $\min \{2+0, 1+7\} = 2$ via B.
 - o Cost of reaching destination C from A = $\min \{2+3, 1+11\} = 5$ via B.
 - o Cost of reaching destination D from A = $\min \{2+7, 1+0\} = 1$ via D.
- The new distance vector of A is

A	0
B	2
C	5
D	1

Table 13.2 – Updated Distance Vector of Router A

- The new distance vector for remaining routers is updated in similar way by calculating the least cost as they receive distance vector from their neighbors.

Distance Vector of B

A	2
B	0
C	3
D	3

Distance Vector of C

A	5
B	3
C	0
D	10

Distance Vector of D

A	1
B	3
C	10
D	0

Table 13.3 – Updated Distance Vector of Router B, C and D

- Now the updated distance vectors are exchanged again and each router updates its distance vector based on the new information in similar manner finding the least cost.

Distance Vector of A

A	0
B	2
C	5
D	1

Distance Vector of B

A	2
B	0
C	3
D	3

Distance Vector of C

A	5
B	3
C	0
D	6

Distance Vector of D

A	1
B	3
C	6
D	0

Table 13.4 – New Updated Distance Vector of all Routers

- The algorithm keeps on repeating periodically and never stops and this is to update the shortest path in case any link goes down or topology changes.
- If there are N routers then routing tables are prepared total (N - 1) times because shortest path between any 2 nodes contains at most N - 1 edges if there are N nodes in the graph.
- The algorithm is as follows

At each node x,

Initialization

for all destinations y in N:

$D_x(y) = c(x,y)$ // If y is not a neighbor then $c(x,y) = \infty$

for each neighbor w

$D_w(y) = ?$ for all destination y in N.

for each neighbor w

send distance vector $D_x = [D_x(y) : y \text{ in } N]$ to w

loop

wait(until I receive any distance vector from some neighbor w)

for each y in N:

$D_x(y) = \min\{c(x,v)+D_v(y)\}$

If $D_x(y)$ is changed for any destination y

Send distance vector $D_x = [D_x(y) : y \text{ in } N]$ to all neighbors

forever

Figure 13.5–Distance Vector Routing Algorithm

- The Distance Vector Routing algorithm suffers from **count to infinity problem** because of routing loops.
- The cost of broken link is infinity and this information is propagated slowly here as it takes several updates before the count is updated to infinity.
- Routing loops usually occur when any interface goes down or two-routers send updates at the same time.
- Consider this scenario

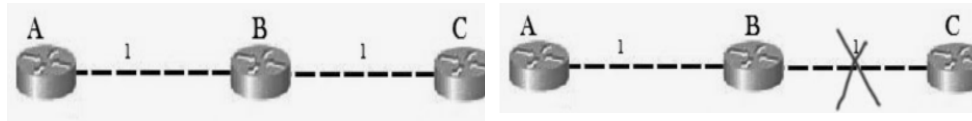


Figure 13.6 – Count to Infinity Scenario

- Router B can reach C at a cost of 1, and A can reach C via B at a cost of 2.
- Now if the link between B and C is disconnected, then B will know that it can no longer reach C via that link and will remove the entry.
- Before it sends the updates, it possibly receives the update from A which advertises that cost to reach C is 2 and so B adds 1 and updates the cost as 3.
- When A receives the updated distance vector from B, it updates the cost to 4 and this cycle repeats towards infinity leading to count to infinity problem.
- This instability can be resolved using **split horizon** technique.
- Here router A does not advertise its route for C to B as direct link from B to C would be more cost effective than route via A and this way updation is stopped and does not reach infinity.
- To achieve efficiency and less increase the size of routing announcements we can combine split horizon with poison reverse where timer is used to record the updation or ignore it.

13.3.2 LINK-STATE ROUTING

- Link state routing is the second family of routing protocols in which each router shares the knowledge of its neighborhood with every other router in the internet work.
- The distance-vector routing algorithm works by having each node share its routing table with its neighbors but in a link-state algorithm the only information passed between nodes is connectivity related.
- Here each router needs a complete map (state) of the network and this collection of state information is called as link-state database (LSDB).
- Each router through the process of flooding creates the link state packet (LSP) which helps in the creation of LSDB.

- The LSP contains identity of the node and the cost of the link.

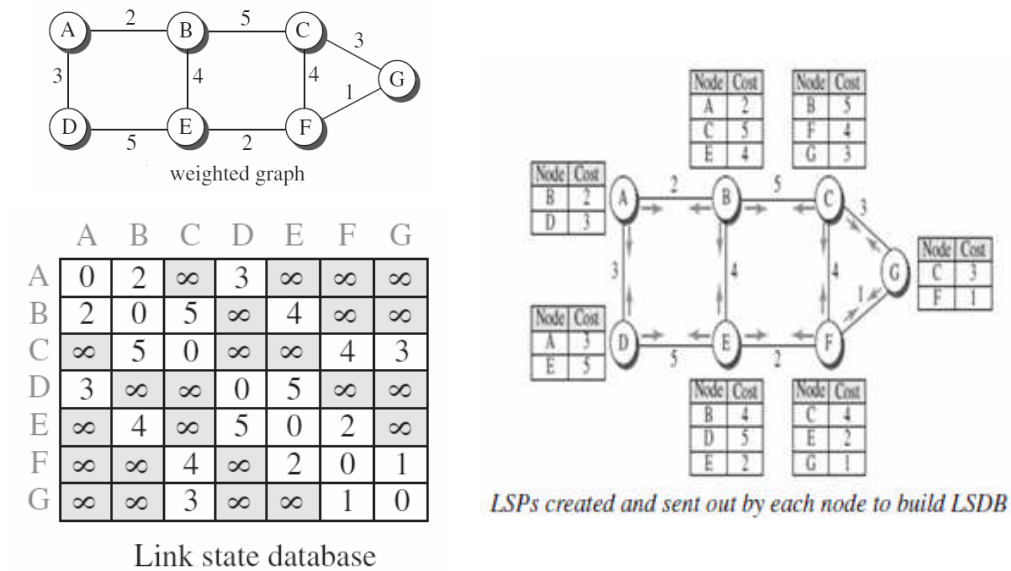


Figure 13.7 – Example of LSDB and LSP

- The LSP is forwarded to the neighbors and it compares its entry with the new LSP and makes the updation accordingly.
- To create least cost tree using the LSDB, the link state routing algorithm uses the **Dijkstra Algorithm**.
- According to Dijkstra's algorithm,
 - A node chooses itself as root and creates a tree with single node and sets cost based on information in LSDB.
 - It then chooses nearby node, adds it to the tree and then rechecks the cost because the paths might have changed.
 - The above process is repeated till all nodes are added to the tree.
- 6 iterations are required for the above example to create the least cost tree.

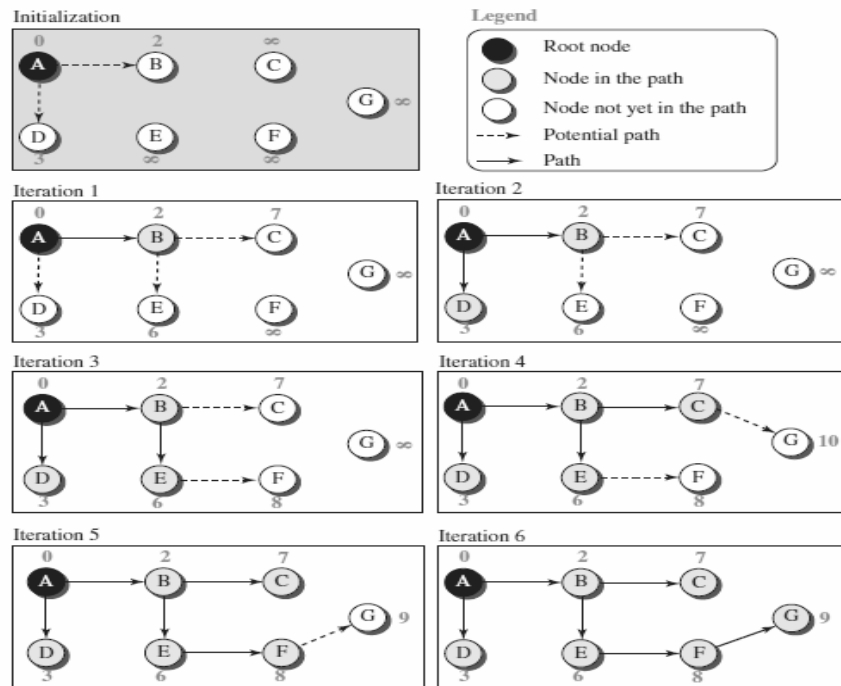


Figure 13.8 – Least cost tree using Dijkstra Algorithm

- The algorithm is as follows

```

Dijkstra's Algorithm ( )
{
    // Initialization
    Tree = {root}
    for (y = 1 to N)
    {
        if (y is the root)
            D[y] = 0
        else if (y is a neighbor)
            D[y] = c[root][y]
        else
            D[y] = ∞
    }
    // Calculation
    repeat
    {
        find a node w, with D[w] minimum among all nodes not in the Tree
        Tree = Tree ∪ {w}
        // Update distances for all neighbors of w
        for (every node x, which is a neighbor of w and not in the Tree)
        {
            D[x] = min{D[x], (D[w] + c[w][x])}
        }
    } until (all nodes included in the Tree)
} // End of Dijkstra

```

Figure 13.9 - Dijkstra Algorithm

13.3.3 PATH-VECTOR ROUTING

- A path vector routing algorithm does not rely on the cost of reaching a given destination to determine whether each path available is loop free or not but instead, it relies on analysis of the path to reach the destination to learn if it is loop free or not.
- Spanning trees are created to learn the path from source to destination.
- They are not based on the least cost trees but can have own policy to create the path.
- In other words, it is essentially a distance vector protocol that does not rely on the distance to destination to guarantee a loop-free path but instead relies on the analysis of the path itself.
- Consider a small network with 5 routers.
- Each router creates its own spanning tree using the policy that it uses minimum number of nodes to reach a destination.

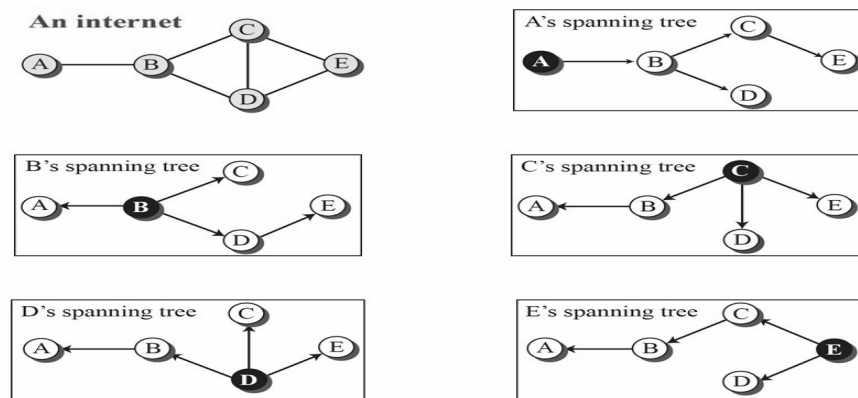


Figure 13.10 - Spanning trees

- The spanning tree are constructed gradually.
- Initially the path vector is created by getting information about its neighbors by sending a greeting message.

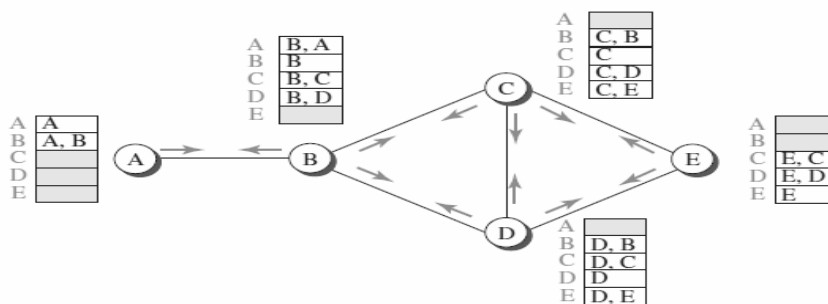


Figure 13.11-Initial Path Vectors

- The routers now send the path vectors to its neighbors and based on policy defined create the spanning tree.
- The equation is as follows $\rightarrow \text{Path}(x, y) = \text{best} \{ \text{Path}(x, y), [(x + \text{Path}(v, y))] \}$ for all v 's in the internet.

- The policy defines choosing the best of multiple paths.
- For example, when node C receives path vector from B, it updates its table as follows

New C		Old C		B	
A	C, B, A	A		A	B, A
B	C, B	B	C, B	B	B
C	C	C	C	C	B, C
D	C, D	D	C, D	D	B, D
E	C, E	E	C, E	E	

$C[] = \text{best}(C[], C + B[])$

Table 13.5 - Updated C vector after obtaining from B

- The algorithm is as follows

```

Path_Vector_Routing ( )
{
    // Initialization
    for (y = 1 to N)
    {
        if (y is myself)
            Path[y] = myself
        else if (y is a neighbor)
            Path[y] = myself + neighbor node
        else
            Path[y] = empty
    }
    Send vector {Path[1], Path[2], ..., Path[y]} to all neighbors
    // Update
    repeat (forever)
    {
        wait (for a vector Pathw from a neighbor w)
        for (y = 1 to N)
        {
            if (Pathw includes myself)
                discard the path // Avoid any loop
            else
                Path[y] = best {Path[y], (myself + Pathw[y])}
        }
        If (there is a change in the vector)
            Send vector {Path[1], Path[2], ..., Path[y]} to all neighbors
    }
} // End of Path Vector

```

Figure 13.12 – Path Vector Routing

13.4 UNICAST ROUTING PROTOCOLS

- Unicast routing is the process of forwarding unicasted traffic from a source to a destination on an internet.

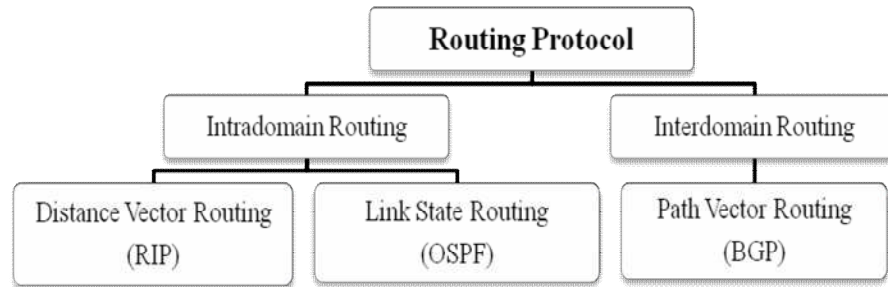


Figure 13.13 – Types of Routing Protocols

13.4.1 INTERNET STRUCTURE

- The Internet is network of network and autonomous system (AS) are huge networks that make up the Internet.
- To manage such huge network, a single protocol cannot be enough for scalability problem and administrative issues.
- Scalability means as size of routing tables increases searching the route to destination becomes time consuming activity and could end up in network traffic.
- The administrative issues deal with controlling and monitoring of the entire Internet.
- So hierarchical routing works well for an AS where a large network or group of networks follow a unified routing policy.
- Every computer or device that connects to the Internet is connected to an AS.
- The AS are not categorized according to their size but based on the way they are connected to other ASs
- There are three types of AS namely
 - Multihomed – Connected to more than one autonomous system.
 - Stub – Only connected to one other autonomous system.
 - Transit – Provides connections through itself. For example, network A can connect to network C directly or by crossing over network B.
- The routing protocol run in each AS is referred to as intra-AS routing protocol or intradomain routing protocol or interior gateway protocol (IGP).
- The global routing protocol is referred to as inter-AS routing protocol or interdomain routing protocol or exterior gateway protocol (EGP).
- The intradomain routing protocols include RIP or OSPF and each AS is free to choose one.
- But it should be clear that we should have only one interdomain protocol such as BGP that handles routing between these entities.

13.4.2 ROUTING INFORMATION PROTOCOL (RIP)

- The Routing Information Protocol (RIP) is an intra domain routing protocol based on distance vector routing algorithm.
- It extends the algorithm where RIP determines the cost of reaching different networks rather than nodes and the cost is calculated based on hop count i.e., number of networks required to reach the destination.
- When the router sends the packet to the network segment, then it is counted as a single hop.

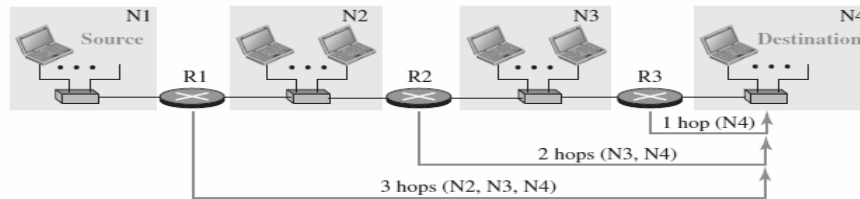


Figure 13.14– Hop Count In RIP

- RIP is suitable for smaller AS where maximum hop count is 15 and 16 represents no connection i.e., infinity.
- The forwarding table contains 3 columns namely address of destination network, address of next router where the packet has to be forwarded and last column is hop count to reach the destination.

Forwarding table of R1			Forwarding table of R2			Forwarding table of R3		
Destination Network	Next Router	Hop Count	Destination Network	Next Router	Hop Count	Destination Network	Next Router	Hop Count
N1	-	1	N1	R1	2	N1	R2	3
N2	-	1	N2	-	1	N2	R2	2
N3	R2	2	N3	-	1	N3	-	1
N4	R2	3	N4	R3	3	N4	-	1

Table 13.6 – Forwarding tables in RIP

- The forwarding table holds address of next router but entire path can be obtained from these tables.
- The forwarding table of R1 defines path to N4 via R2, the table of R2 defines the path via R3 and table of R3 tells no next router which means the route is $R1 \rightarrow R2 \rightarrow R3 \rightarrow N4$.
- RIP creates tables at network layer but is run as service of UDP using well known port 520.
- There are two versions of RIP namely RIP-1 and RIP2.
- The format of RIP-2 message is

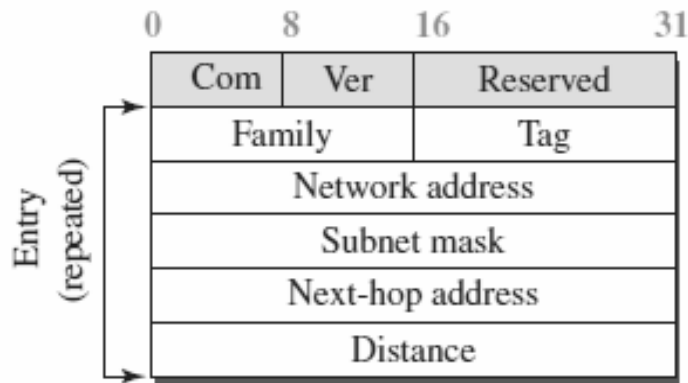


Figure 13.15 – RIP-2 message format

- RIP has two message type i.e., request = 1 and response = 2 indicated in command field.
- Version field holds the value 2 and reserved field is filled with all 0s.
- Family field holds value 2 for TCP/IP and tag field holds details about the AS.
- Next field contains the destination address, prefix length, address length and hop count.
- Request message asks for details and response message provides with the details.
- A solicited response is sent in response to request but an unsolicited response is sent every 30 seconds to update changes in connections.
- The implementation of protocol is same as algorithm that instead of distance vector the entire forwarding table is sent and routes are added and deleted based on the exchange of information.
- RIP updates this information based on three timers.
 - **RIP Update timer** - The routers configured with RIP send their updates to all the neighboring routers every 30 seconds.
 - **RIP Invalid timer**- The RIP invalid timer is 180 seconds, which means that if the router is disconnected from the network or some link goes down, then the neighbor router will wait for 180 seconds to take the update. If it does not receive the update within 180 seconds, then it will mark the particular route as not reachable.
 - **RIP Flush timer** - The RIP flush timer is 120 second means that if the router does not receive the update within 120seconds, then the neighbor route will remove that particular route from the forwarding table.

13.4.3 OPEN SHORTEST PATH FIRST (OSPF)

- The Open Shortest Path First (OSPF) is intra domain routing protocol based on link-state routing protocol.

- It is suitable for large AS cost to reach the destination is calculated based on weight assigned based on throughput, reliability and round-trip time.

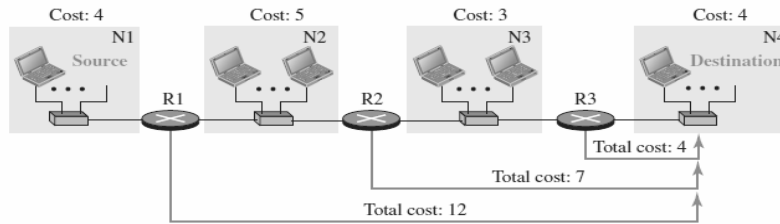


Figure 13.16 – Cost in OSPF

- The OPSF maintain forwarding tables similar as in RIP but calculation of shortest path is done on the basis of Dijkstra's algorithm.

Forwarding table of R1			Forwarding table of R2			Forwarding table of R3		
Destination Network	Next Router	Hop Count	Destination Network	Next Router	Hop Count	Destination Network	Next Router	Hop Count
N1	-	4	N1	R1	9	N1	R2	12
N2	-	5	N2	-	5	N2	R2	8
N3	R2	8	N3	-	3	N3	-	3
N4	R2	11	N4	R3	7	N4	-	4

Table 13.7 – Forwarding tables in OSPF

- As mentioned OSPF are used in large AS, OSPF uses another level of hierarchy in routing i.e., first level is AS and second is area.
- So, each router needs to know link states of its area and other areas as well and to simplify this a backbone area is considered that connects with all other areas.
- The routers in backbone area are responsible for passing information collected by each area to other areas
- The backbone area is identified as zero.

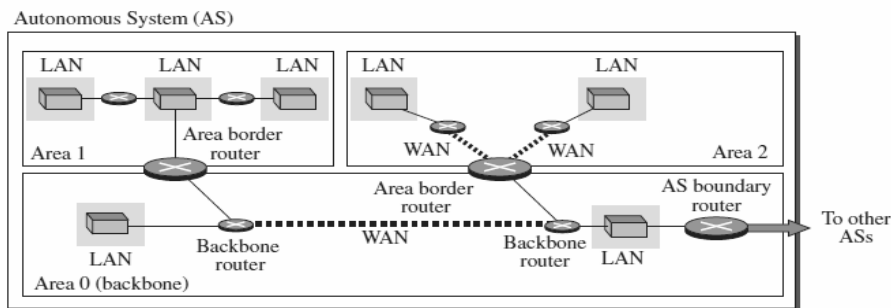


Figure 13.17 – Areas in AS

- OSPF is based on the link-state routing algorithm, which requires that a router advertise the state of each link to all neighbors for the formation of the LSDB.
- Different types of advertisement are available for different situations.

- o **Router link** - It advertises the existence of a router. There are several types of router link such as *transient link* which advertises network connected to several networks via routers, *stub link* that advertises to a stub network and *point-to-point link* that advertises an end point.
- o **Network link** - It advertises the network as a node.
- o **Summary link to network** – It is advertised by an area border router that collects summary of links information from backbone to area and vice-versa.
- o **Summary link to AS** –It is advertised by AS router about summary links from AS which can be used later by the networks in other AS.
- o **External link**– It is also advertised by AS router to inform about single network existence outside AS to backbone area.

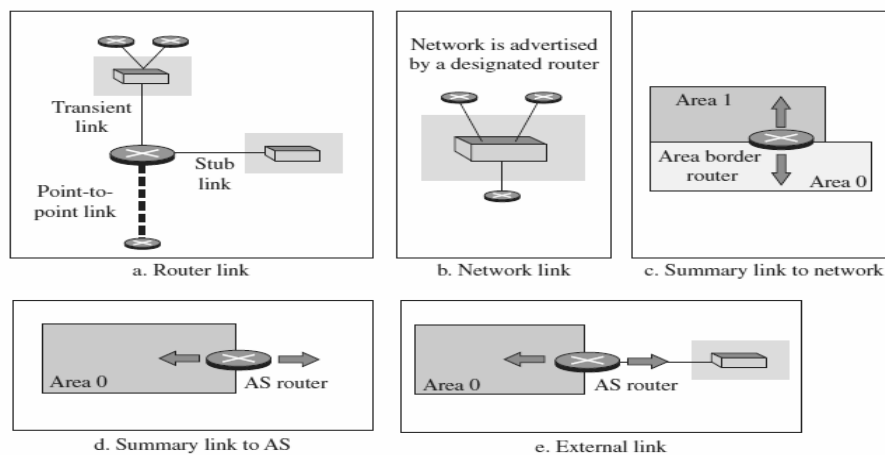


Figure 13.18–OSPF Advertisements

- OSPF uses the services of IP with protocol field = 89 for IP datagram carrying OSPF message.
- Two version of OSPF are available and version 2 is mostly used.
- OSPF is a very complex protocol and defines five different types of messages.
 - o **Hello message** - This is type1 packet and used for neighbor discovery and keep alive with default timer as 10 seconds.

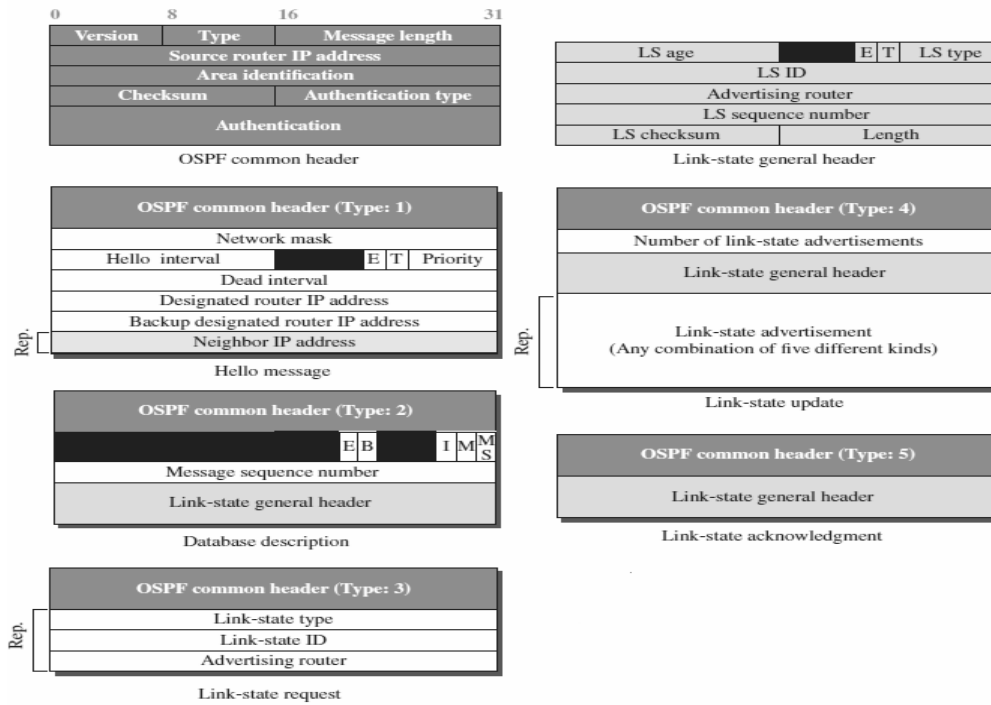


Figure 13.19 – OSPF Message Types

- o **Database Description message** - This is type 2 packet and used to synchronize LSDB between two routers. Along with this, this message helps in neighbor formation and MTU size negotiation.
- o **Link State Request** - This is type 3 packet and used to request specific link-state records from an OSPF neighbor router.
- o **Link State Update** - This is type 4 packet and used as reply to the previous type when slave router contacts the master router for information.
- o **Link State Acknowledgement** - This is type 5 packet and used for acknowledgment purposes.
- The implementation of OSPF is based on link state algorithm where router creates a shortest path tree to reach the destination.

13.4.4 BORDER GATEWAY PROTOCOL VERSION 4 (BGP4)

- It is the only interdomain routing protocol and is based on path vector routing algorithm.
- It is standardized protocol designed to exchange routing and reachability information among AS.
- Consider the following network with AS1 as transient AS and AS2, AS3 and AS4 as stub AS.

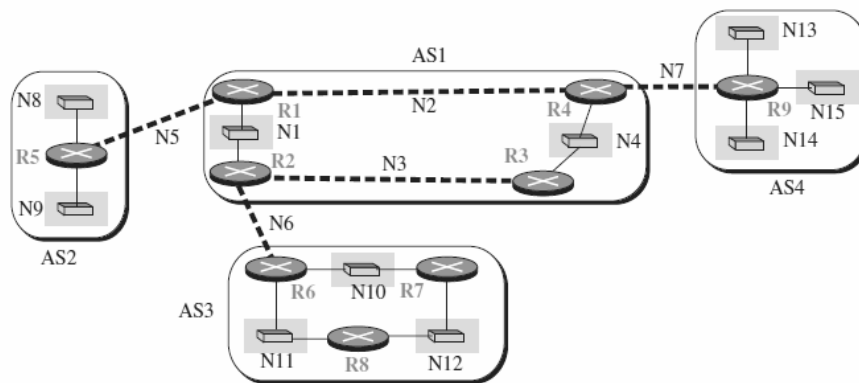


Figure 13.20 – Model network

- Each AS uses intradomain protocol such as RIP or OSPF for routing internally within the AS.
- But to know how to route packets to network in another AS, we require the BGP protocol.
- There are two variants of BGPv4 protocol.
- The external BGP (eBGP) is run on each border router i.e., the one at the edge of each AS which is connected to a router at another AS.
- Another version is internal BGP (iBGP) run on all routers.
- So, border routers run three protocols namely intradomain, iBGP and eBGP and other routers run two protocols namely intradomain and iBGP.
- The border routers that run the eBGP are called as BGP peers or speakers.
- So, in the above network three TCP sessions are created to exchange information between the border routers namely R1-R5, R2-R6 and R4-R9.

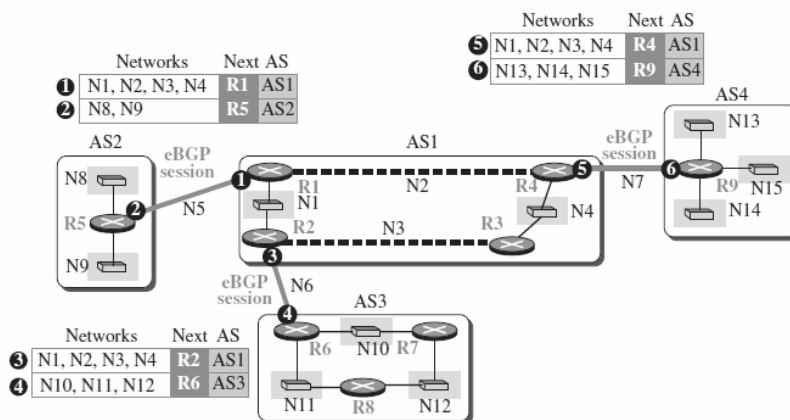


Figure 13.21 - eBGP Session

- The circled numbers in the figure define sending routers that gather the network information through the intradomain routing protocol.
- For example, in message number 1, R1 tells R5 about the network N1, N2, N3 and N4 in AS1 can be reached through it.
- Router R5 now adds these entries to its table and it informs R1 that network N8 and N9 in AS2 can be reached through it.
- This session faces two problems:
 - Often Routers do not know how to route packets to non-neighbor AS i.e., R6 will not know route to AS2 and AS4.
 - None of the non-border routers know how to route a packet destined for any networks in other ASs.
- The above problems are resolved by running the iBGP on all the routers.
- The iBGP uses the service of TCP with well-known port number 179.
- It also creates session but between any pair of routers inside an AS but single router in an AS cannot create session such as AS2 and AS4.
- If there are N routers in AS then $[N \times (N - 1) / 2]$ iBGP sessions are created.

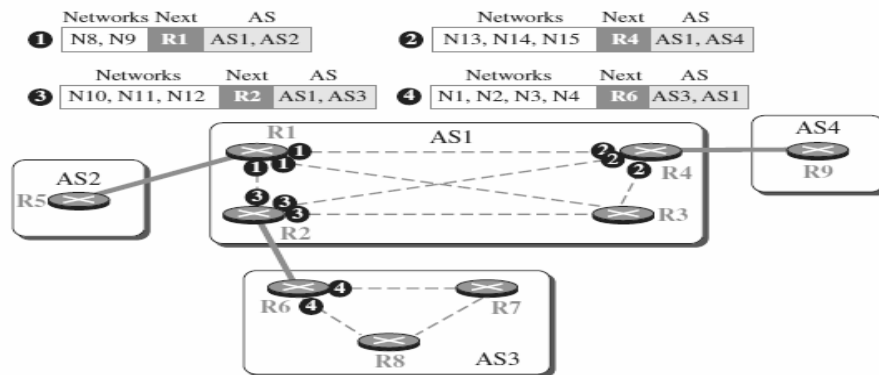


Figure 13.22 - iBGP Session

- The circled numbers in the figure define the router sending separate information to all border routers about the reachability such as R1 announces that network N8 and N9 can be accessed via AS1-AS2 with next router as R1 through separate message to R2, R4 and R6.
- Similarly, information from other routers is sent and each router maintains the path tables.

Path table of R1

Network	Next	Path
N8, N9	R5	AS1, AS2
N10, N11, N12	R2	AS1, AS3
N13, N14, N15	R4	AS1, AS4

Path table of R2

Network	Next	Path
N8, N9	R1	AS1, AS2
N10, N11, N12	R6	AS1, AS3
N13, N14, N15	R1	AS1, AS4

Path table of R3

Network	Next	Path
N8, N9	R2	AS1, AS2
N10, N11, N12	R2	AS1, AS3
N13, N14, N15	R4	AS1, AS4

Path table of R4

Network	Next	Path
N8, N9	R1	AS1, AS2
N10, N11, N12	R1	AS1, AS3
N13, N14, N15	R9	AS1, AS4

Path table of R5

Network	Next	Path
N1, N2, N3, N4	R1	AS2, AS1
N10, N11, N12	R1	AS2, AS1, AS3
N13, N14, N15	R1	AS2, AS1, AS4

Path table of R6

Network	Next	Path
N1, N2, N3, N4	R2	AS3, AS1
N8, N9	R2	AS3, AS1, AS2
N13, N14, N15	R2	AS3, AS1, AS4

Path table of R7

Network	Next	Path
N1, N2, N3, N4	R6	AS3, AS1
N8, N9	R6	AS3, AS1, AS2
N13, N14, N15	R6	AS3, AS1, AS4

Path table of R8

Network	Next	Path
N1, N2, N3, N4	R6	AS3, AS1
N8, N9	R6	AS3, AS1, AS2
N13, N14, N15	R6	AS3, AS1, AS4

Path table of R9

Network	Next	Path
N1, N2, N3, N4	R4	AS4, AS1
N8, N9	R4	AS4, AS1, AS2
N10, N11, N12	R4	AS4, AS1, AS3

Table 13.8 – BGP Path Tables

- The path tables created above are actually not used for routing and they are injected in the intradomain forwarding tables.
- RIP and OSPF use different metrics to calculate cost.

- We use RIP and BGP together to calculate and cost to reach foreign network is same as cost to reach the first AS in path.

Forwarding table of R1			Forwarding table of R2			Forwarding table of R3			Forwarding table of R4		
Network	Next	Cost	Network	Next	Cost	Network	Next	Cost	Network	Next	Cost
N1	-	1	N1	-	1	N1	R2	2	N1	R1	2
N4	R4	2	N4	R3	2	N4	-	1	N4	-	1
N8	R5	1	N8	R1	2	N8	R2	3	N8	R1	2
N9	R5	1	N9	R1	2	N9	R2	3	N9	R1	2
N10	R2	2	N10	R6	1	N10	R2	3	N10	R3	3
N11	R2	2	N11	R6	1	N11	R2	2	N11	R3	3
N12	R2	2	N12	R6	1	N12	R2	2	N12	R3	3
N13	R4	2	N13	R3	3	N13	R4	2	N13	R9	1
N14	R4	2	N14	R3	3	N14	R4	2	N14	R9	1
N15	R4	2	N15	R3	3	N15	R4	2	N15	R9	1
Forwarding table of R5			Forwarding table of R6			Forwarding table of R7			Forwarding table of R8		
Network	Next	Cost	Network	Next	Cost	Network	Next	Cost	Network	Next	Cost
N8	-	1	N10	-	1	N10	-	1	N10	R6	2
N9	-	1	N11	-	1	N11	R6	2	N11	-	1
0	R1	1	N12	R7	2	N12	-	1	N12	-	1
			0	R2	1	0	R6	2	0	R6	2
Forwarding table of R9											
Network	Next	Cost									
N13	-	1									
N14	-	1									
N15	-	1									
0	R4	1									

Table 13.9 – Forwarding Table of RIP after BGP injection

- The tables of intradomain routing protocol may get huge due to the injection and hence address aggregation technique can be used.
- The details in the forwarding table next hop and cost are referred as path attributes in BGP.
- BGP defines 7 path attributes namely type 1 – ORIGIN, type 2 – AS-PATH, type 3 – NEXT-HOP, type 4 – MULT-EXIT-DISC, type 5 – LOCAL-PREF, type 6 – ATOMIC-AGGREGATE and type 7 – AGGREGATOR.

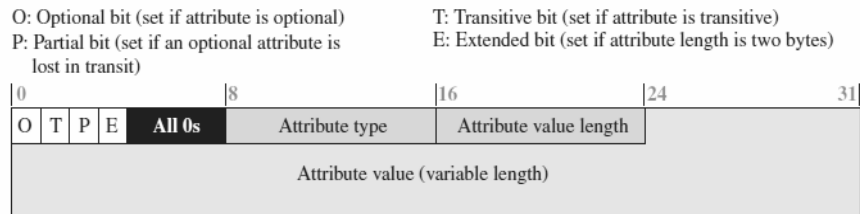


Figure 13.23 – Path Attributes in BGP

- The first four fields define the flags followed by any seven type of attribute and length of attribute value field.
- These attribute type play an important role when there are several routes available from source to destination.

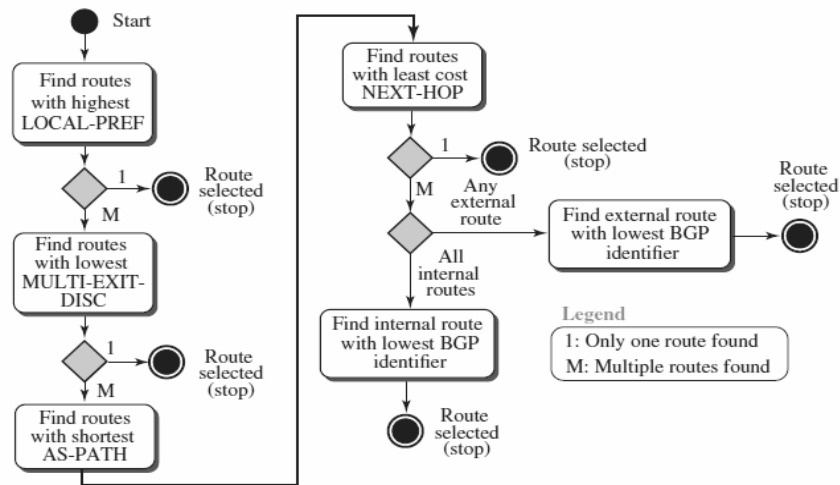


Figure 13.24 – BGP Route Selection

- BGP defines four different messages for communication to establish the routing information.
 - o **Open message**– It is used to establish a BGP adjacency.
 - o **Update message**– It advertises any feasible routes, withdraws previously advertised routes, or can do both.
 - o **Keepalive message** - It is exchanged every one-third of the Hold Timer agreed upon between the two BGP routers as BGP does not rely on the TCP connection state to ensure that the neighbors are still alive.
 - o **Notification message**– It is sent when an error is detected with the BGP session, such as a hold timer expiring, neighbor capabilities change, or a BGP session reset is requested.

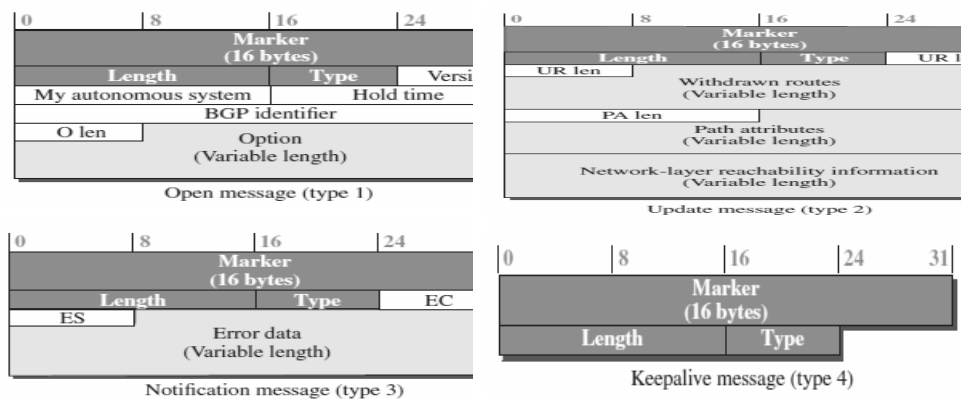


Figure 13.25 – BGP Messages

13.5 SUMMARY

- In Unicast routing a packet is routed hop by hop from its source to its destination by the help of forwarding tables.

- Distance Vector Algorithm updates distances from neighbours distances and uses Bellman-Ford to find the shortest path.
- Link State Algorithm uses flood link state advertisements to all routers that uses Dijkstra's shortest path to find the route the destination.
- Path Vector Algorithm updates paths based on neighbours' paths and uses local policy to rank paths and find route to the destination.
- Routing Information Protocol (RIP) is based on Distance Vector Algorithm and Open Shortest Path First (OSPF) is based on the Link State Algorithm are the commonly used intradomain protocols in the Internet today.
- Border Gateway Protocol (BGP) is based on the Path Vector Algorithm and is the interdomain routing protocol used in the Internet today.

13.6 LIST OF REFERENCES

- "Data Communications and Networking" by Behrouz A. Forouzan, 5th Edition, McGraw-Hill Publication.
- <https://www.gatevidyalay.com>
- <https://www.javatpoint.com>
- <https://www.ciscopress.com>

13.7 UNIT END EXERCISE

1. In a graph, if we know that the shortest path from node A to node G is $A \rightarrow B \rightarrow E \rightarrow F \rightarrow G$. What is the shortest path from node G to node A?
2. Assume the shortest path in a graph from node A to node H is $A \rightarrow B \rightarrow C \rightarrow H$. Also assume that the shortest path from node H to node K is $H \rightarrow J \rightarrow K$. What is the shortest path from node A to node K?
3. List the three types of autonomous system and give the differences between them.
4. Why RIP uses the service of UDP instead of TCP?
5. Why BGP uses the service of TCP and not UDP?



NEXT GENERATION IP

Unit Structure

14.0 Objectives

14.1 Introduction

14.2 IPv6 Addressing

14.2.1 Representation

14.2.2 Address Types

14.2.3 Auto configuration and Renumbering

14.3 IPv6 Protocol

14.3.1 Packet Format

14.3.2 Extension Header

14.4 ICMPv6 Protocol

14.4.1 ICMPv6 Messages

14.4.1.1 Error Reporting Message

14.4.1.2 Informational Message

14.4.1.3 Neighbor Discovery Message

14.4.1.4 Group Membership Message

14.5 Transition from IPv4 to IPv6

14.5.1 Strategies

14.6 Summary

14.7 List of References

14.8 Unit End Exercise

14.0 OBJECTIVES

After going through this chapter, you will be able to

- Understand the need for IPv6 and its addressing and representation of IPv6 addresses
- Understand the new packet and extension header of IPv6
- Understand the different message available in ICMPv6
- Understand how transition should take place from IPv4 to IPv6

14.1 INTRODUCTION

- The world is changing, and everything is connecting to an IP network.
- The IPv4 has been the reigning Internet Protocol version for several decades now even till today.
- But the address depletion problem of IPv4 has caused IPv6 to come into picture.
- The IPv4 is running out of room to accommodate all of the unique IP addresses required for the world's growing number of connected devices.
- The IPv6 is the latest version of the Internet Protocol which identifies devices across the internet so they can be located.

14.2 IPV6 ADDRESSING

- IPv6 provides a larger addressing space.
- An IPv6 address is 128-bit long compared to 32-bit IPv4 address.
- It is four times the address length in IPv4.
- IPv6 uses 128-bit (2^{128}) addresses, allowing 3.4×10^{38} unique IP addresses.

14.2.1 REPRESENTATION

- The 128-bit binary notation is divided into each 16-bit block and each block represented by four hexadecimal digits separated by colon called the colon hexadecimal notation.

Particulars	IPv4	IPv6
Address Size	32-bit	128-bit
No. of Addresses	$2^{32} = 4,294,967,296$	$2^{128} = 340,282,366,920,938,463,374,607,431,768,211,456$
Address Format	Dotted Decimal Notation	Colon Hexadecimal Notation
Example	192.168.101.10	3FFE:F200:0234:AB00:0321:4256:9810:AB12
Prefix Notation	192.168.0.0/24	3FFE:F200:0234::/48

Table 14.1 - IPv4 and IPv6 Addressing Formats

- We can abbreviate IPv6 address as the hexadecimal notation is also very long.
- Abbreviation can be done by omitting the leading zeros as in 0085 is abbreviated as 85 and 000E as E and 0000 as 0.
- Trailing zeros cannot be omitted as 4560 cannot be abbreviated
- Another form of abbreviation called zero compression allows consecutive section of zeros to be combined and replaced with double colon as in E380:0:0:0:0:BBA3:0:FEEE is abbreviated as E380::BBA3:0:FEEE.
- Zero compression can be applied only once in the address.
- IPv6 address can also be represented in mixed notation by combining colon hex and dotted decimal notation
- For example, ::130.24.24.19 is a valid IPv6 address with zero compression.
- IPv6 also uses hierarchical addressing and can be represented via prefix and suffix called the CIDR notation such as FDEC::BBFF:0:FFFE/60

14.2.2 ADDRESS TYPES

- An IPv6 destination address can be unicast, anycast or multicast.
- Unicast address is meant to configure on one interface so that you can send and receive IPv6 packets.
- Anycast address is assigned to a group of interfaces and a packet sent to an anycast address is delivered to only one of the nearest hosts.
- Multicast address is also assigned to a group of interfaces and the packet is sent to all interfaces identified by the address.
- The IPv6 address space is recognized by logically dividing the 128 bits into several blocks of varying size and each block is allocated for a special purpose.

IPv6 Address Type	Subtype	Representation	IPv4 Equivalent
Unspecified		::/128	0.0.0.0
Loop back		::1/128	127.0.0.1
Unicast	Link-Local	FE80::/10	169.254.0.0/16
	Unique Local	FC00::/7	None
	Global	2000::/3	Public IPs
Anycast	Same as unicast	Same as unicast	None
Multicast		FF00::/8	224.0.0.0/4

Table 14.2 – Prefix for IPv6 address

- The block called the **global unicast address** block with address 2000::/3 is used for unicast communication between two hosts in the Internet.
- The three leftmost bits are the same for all addresses in this block i.e., 001 and so 2^{125} bits make the size of the block.
- An address in this block is divided into three parts namely global routing prefix (n bits), subnet identifier (m bits), and interface identifier (q bits).

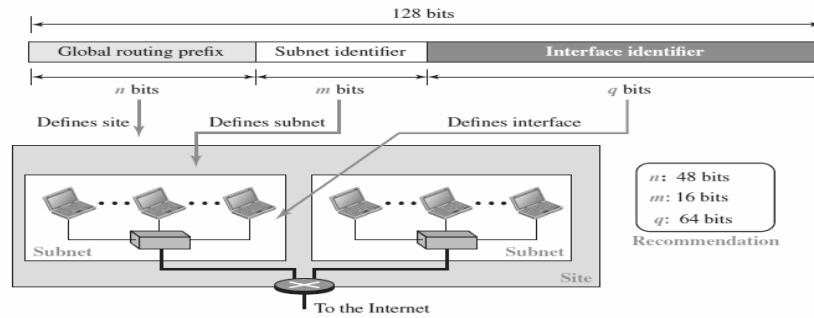


Figure 14.1 – IPv6 Global Unicast Address

- The global routing prefix is used to route packet through the Internet and three bits being fixed so rest 45 bits define 245 sites.
- The m bits define the subnet and so $2^{16} = 65536$ subnets are available.
- For example, an organization is assigned the block 2000:1456:2474/48, then the CIDR notation for the blocks in the first and second subnet in this organization are 2000:1456:2474:0000/64 and 2000:1456:2474:0001/64.
- The last q bits identify the interface similar to host id in IPv4 addressing.
- In IPv4 there is no relation between link layer address and host id of the IP address.
- IPv6 defines a relationship between the two through a mapping process where a link-layer address whose length is less than 64 bits can be embedded as the whole or part of the interface identifier
- Two common mapping process are available.
 - To map a 64-bit physical address, the global/local bit of this format needs to be changed from 0 to 1 (local to global) to define an interface address

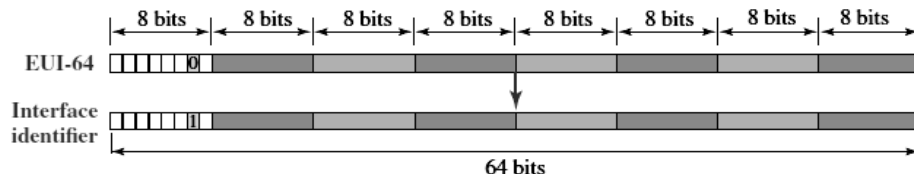


Figure 14.2 – Mapping for EUI-64

- For example, the interface identifier for the physical address in the EUI (F5-A9-23-EF-07-14-7A-D2)₁₆ is obtained by changing the seventh bit of the first octet from 0 to 1 and format to colon hex notation. The result is F7A9:23EF:0714:7AD2.
- To map a 48-bit Ethernet address into a 64-bit interface identifier, we need to change the local/global bit to 1 and insert an additional 16 bits as 15 ones followed by one zero, or FFFE₁₆.

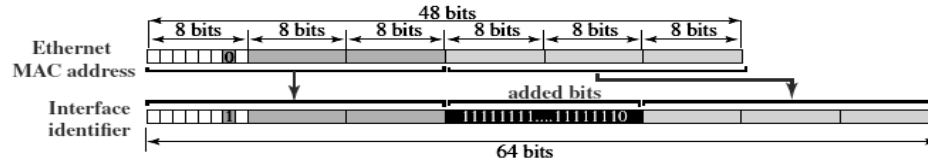


Figure 14.3 – Mapping for Ethernet MAC

- For example, the interface identifier for the Ethernet physical address is (F5-A9-23-14-7A-D2)₁₆ is obtained by changing the seventh bit of the first octet from 0 to 1, insert two octets FFFE₁₆ and change the format to colon hex notation. The result is F7A9:23FF: FE14:7AD2.
- Special addresses with the prefix 0000::

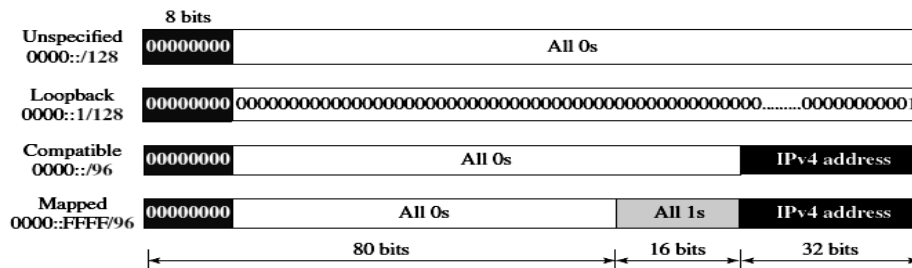


Figure 14.4 - Special addresses

- An **unspecified address** is a single address i.e., 0000::- In IPv6, the **loopback address** block has only a single address in it.i.e., 00001::- A **compatible address** is an address of 96 bits of zero followed by 32 bits of IPv4 address.
- A **mapped address** is used when a host already migrated to version 6 wants to send a packet to a host still using version 4.
- The **unique local unicast block** is privately created and not used for routing but block identifier 1111 110 is fixed, next bit can be 0 or 1 defines how address can be assigned locally followed by 40 bits randomly generated.

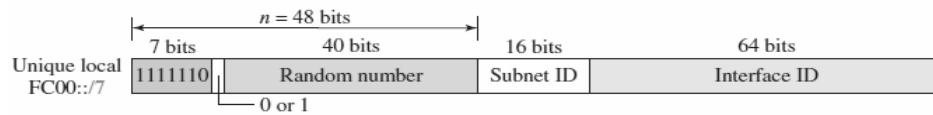


Figure 14.5 – Unique Local Block

- The **link local block** is used as private address has block identifier 111111010 followed by next 54 bits as zero and last 64 bits defining the interface for each computer.

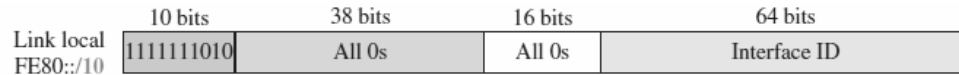


Figure 14.6 – Link Local Block

- The **multicast addresses** define a group of hosts and uses block identifier as 11111111 followed by flag where 0 = permanent means can be used all times and 1 = transient means can be used temporary followed by different definitions of scope.

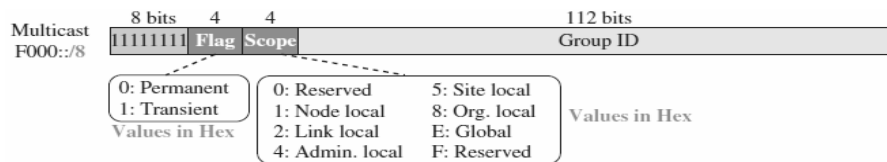


Figure 14.7 – Multicast Address

14.2.3 AUTOCONFIGURATION AND RENUMBERING

- In IPv4 the configuration of host and routers are done manually by the network administrator or through the Dynamic Host Configuration Protocol (DHCP).
- But in IPv6, the host can configure through DHCP and also by itself.
- The auto configuration of host involves the following process.
 - The host creates a 128-bit link local address by setting prefix as 1111 1110 10 followed by 54 zeros and adding 64-bit interface identifier.
 - The host now needs to check if the address created in previous step is unique. This checking is done by sending neighbor solicitation message and waits for neighbor advertisement message. If the address is used, then auto configuration process fails and host can be configured using DHCP only.
 - If address is found to be unique then host stores address as link local and sends router solicitation message to obtain global unicast address. The host receives the router advertisement message that includes the global unicast prefix and subnet prefix that the host needs to add to its interface identifier to create the address. In case the router does not respond then host needs to adopt other configuration techniques.

- For example, in an organization the Ethernet address is (F5-A9-23-11-9B-E2)16, the global unicast prefix is 3A21:1216:2165 and the subnet identifier is A245:1232 then host first creates interface identifier by changing the 7th bit from 0 to 1 and the address is F7A9:23FF:FE11:9BE2. It then creates link local address as FE80::F7A9:23FF:FE11:9BE2. Once the uniqueness is verified, it creates the global unicast address by appending the global unicast prefix and subnet identifier to obtain 3A21:1216:2165:A245:1232:F7A9:23FF:FE11:9BE2.
- Renumbering of devices is a method related to autoconfiguration.
- Like host configuration it can be implemented through DHCP where the use of IP address “leases” that expire after a period of time.
- In IPv6, networks are renumbered by having routers specify an expiration interval for network prefixes when autoconfiguration is done.
- Later, they can send a new prefix to tell devices to regenerate their IP addresses.
- Devices can actually maintain the old “deprecated” address for a while and then move over to the new address.
- DNS support is mandate for renumbering mechanism which needs to propagate the new addressing associated with domain name.
- So new protocol Next Generation DNS is under study to support this mechanism.

14.3 IPv6 PROTOCOL

- The change in IPv6 address size required the change to be bought into the packet format as well.
- IPv6 is way better than IPv4 in terms of complexity and efficiency.
- Several reasons for the changes in the format in addition to address size and format are better header format, new options, new extension to support technologies and application and better support for resource allocation and security.

14.3.1 PACKET FORMAT

- An IPv6 packet has two parts: a header and payload.



Figure 14.8 – IPv6 packet format

- The header consists of a fixed portion with minimal functionality required for all packets and occupies 40 bytes.
- The payload can be up to 65,535 bytes of information.

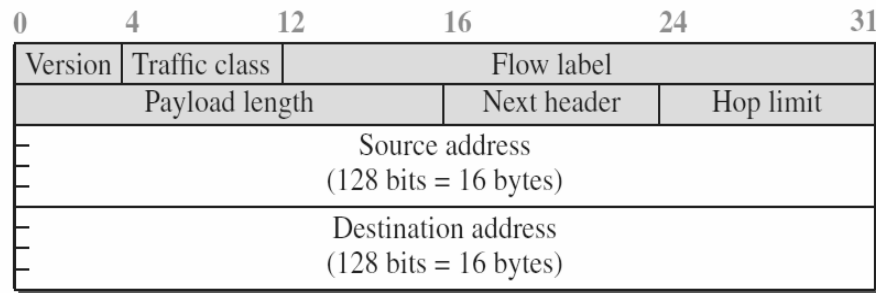


Figure 14.9 – IPv6 packet format

- Version indicates the current version which is 0110.
- The Traffic Class field indicates class or priority of IPv6 packet. It helps routers to handle the traffic based on priority of the packet.
- Flow Label field is used by source to label the packets belonging to the same flow in order to request special handling by intermediate IPv6 routers. This makes IPv6 packet to allow IPv6 to work as a connection-oriented protocol.
- Payload length is 16-bit field that indicates total size of the payload which tells routers about amount of information a particular packet contains in its payload.
- Next Header indicates type of extension header(if present) immediately following the IPv6 header.
- Hop Limit field is same as TTL in IPv4 packets and it indicates the maximum number of intermediate nodes IPv6 packet is allowed to travel.
- The source and destination address are 128-bit field and represents the address of original source and final destination.
- The payload in IPv6 has different format than IPv4 and is combination of zero or more extension header followed by data from other protocols.
- Fragmentation of packets can be performed only by the source and not intermediate routers and reassembly takes place in the destination only.
- The source checks packet size and based on route determines whether to fragment or not and hence packet format does not include fields for fragmentation.
- In case intermediate router cannot forward the packet then it simply drops it and send ICMPv6 error message to the source.

14.3.2 EXTENSION HEADER

- In order to rectify the limitations of IPv4 option field, extension headers are introduced in IPv6 which add extra functionality to the IP datagram.
- There are upto six extension headers.

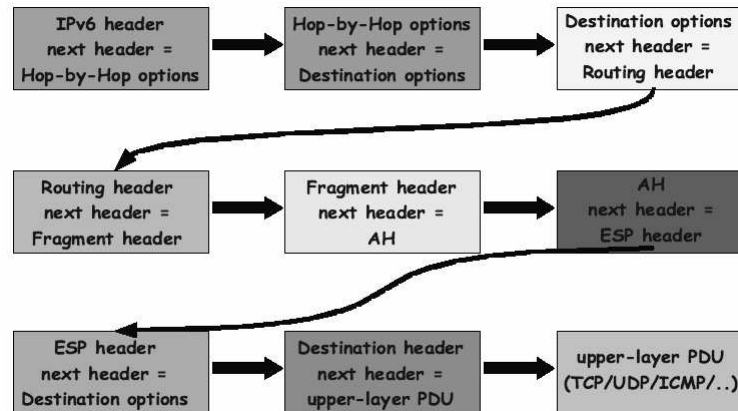


Figure 14.10 – Extension Header

- Hop-by-Hop Option – It specifies the delivery parameters such as length of datagram, management, debugging and control information at each hop on the path to the destination host.
- Destination Option – It specifies packet delivery parameters to the final destination host. Intermediate destination devices are not permitted to access information.
- Source Routing – It defines strict source routing and loose source routing for the packet.
- Fragmentation – As only source host can perform fragmentation, it uses the fragment extension header to tell the destination host the size of the packet that was fragmented so that the destination can reassemble the packet.
- Authentication – It provides authentication, data integrity, and anti-replay protection.
- Encrypted Security Payload –It provides data confidentiality, data authentication, and anti-replay protection
- Destination IP Address – It identifies the host or interface on a node to which the IPv6 packet is to be sent. The destination address may appear twice, the first instance after the hop limit following the source IP address and the second instance after the final extension header.

Order	Header Type	Next Header Code
1	Basic IPv6 Header	-
2	Hop-by-Hop Option	0
3	Destination Option	60
4	Source Routing	43
5	Fragmentation	44
6	Authentication	51
7	Encrypted Security Payload	50
8	Destination IP Address	60
Upper Layer	TCP	6
Upper Layer	UDP	17
Upper Layer	ICMPv6	58

Table 14.3 – IPv6 Next Header Code

14.4 ICMPv6 PROTOCOL

- Just as there is update in version of Internet Protocol (IP) from 4 to 6, the network layer also updated the Internet Control Message Protocol (ICMP) from version 4 to 6.
- The ICMPv6 is an integral part of the IPv6 architecture and must be completely supported by all IPv6 implementations.
- The ICMP, ARP and IGMP protocol in version 4 are combined into single protocol ICMPv6.

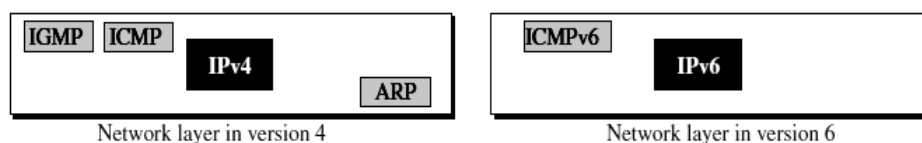


Figure 14.11 – Network layer in Version 4 and 6

- ICMPv6 is much more powerful than ICMPv4 and contains new functionalities such as reporting errors encountered in processing packets, performing diagnostics, performing Neighbor Discovery, and reporting multicast memberships.

14.4.1 ICMPv6 MESSAGES

- ICMPv6 messages may be classified as error messages and information messages.
- The IPv6 packets carry the ICMPv6 message with next header value set to 58.

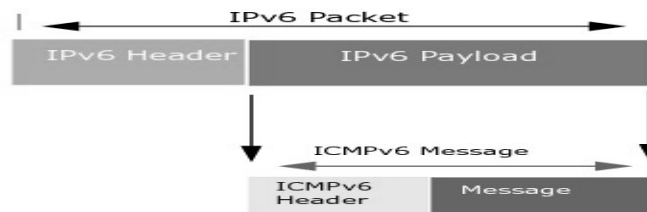


Figure 14.12 – ICMPv6 Embedded in IPv6 Packet

- The ICMPv6 message consists of a header and message.
- Header has three fields namely type, code and checksum.

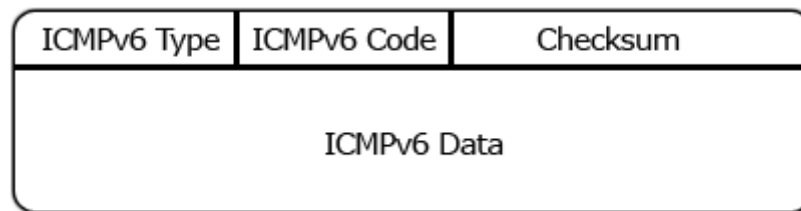


Figure 14.13 – ICMPv6 Packet Format

- Type indicates the type of message with high order value = 0 (0 to 127) = error message and high order value = 1 (128 to 255) = information message.
- Code field is based on message type and checksum is for message content integrity.
- The data contains the ICMPv6 message.
- The ICMPv6 error message are categorized into four groups

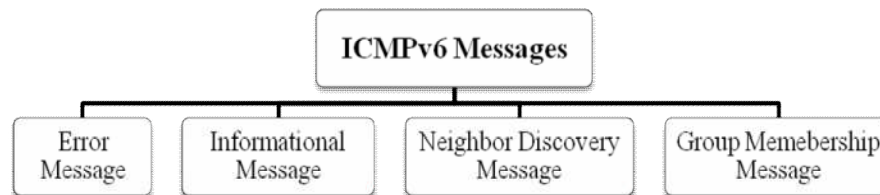


Figure 14.14 – ICMPv6 Message Categories

14.4.1.1 ERROR REPORTING MESSAGE

- ICMPv6 error messages are similar to ICMPv4 error messages.
- There are four ICMPv6 error reporting message.
- The **Destination Unreachable message**(type = 1)is generated when the network fails to deliver an IPv6 packet and hence must discard the packet because the destination is unreachable due to several reasons indicated by different code values.

Code	Meaning
0	No route to destination
1	Communication with destination administratively prohibited
2	Not a neighbour
3	Address unreachable
4	Port unreachable

Table 14.4 – Destination Unreachable: Code Field Values

- The **Packet Too Big message**(type = 2 and code = 0) is generated when the network must discard an IPv6 packet because its size exceeds the MTU of the outgoing link.
- The **Time Exceeded message**(type = 3) is generated when a router must discard an IPv6 packet because its Hop Limit field is zero or decrements to zero and message indicates that either a routing loop or an initial hop limit value is too small.

Code	Meaning
0	Hop limit exceeded in transit
1	Fragment reassembly time exceeded

Table 14.5 – Time Exceeded: Code Field Values

- The **Parameter Problem message** (type = 4) is generated when an IPv6 node must discard a packet because it detects problems in a field of the IPv6 header or of an extension header.

Code	Meaning
0	Erroneous header field
1	Unrecognized Next Header
2	Unrecognized IPv6 option

Table 14.6 – Parameter Problem: Code Field Values

14.4.1.2 INFORMATIONAL MESSAGE

- The **Echo Request message** and **Echo Reply message** are two of the ICMPv6 informational message.
- The Echo Request message and its corresponding Echo Reply message (type = 129 and code = 0) are ICMPv6 diagnostic messages.
- These two messages are used to implement the ping diagnostic application that allows us to test whether a destination is reachable.
- A host or router can send an echo-request message to another host; the receiving computer or router can reply using the echo-reply message.

14.4.1.3 NEIGHBOR DISCOVERY MESSAGE

- Several neighbour discovery messages are redefined in ICMPv6.
- There are seven messages in this category.

- **Router Solicitation message** (type = 133 and code = 0) are sent by host to prompt routers to generate Router Advertisements immediately. It is sent by host to find a router in the network.
- **Router Advertisement message** (type = 134 and code = 0) is sent periodically or in response to Router Solicitation messages.
- **Neighbour Solicitation message** (type = 135 and code = 0) is sent by source host to request link layer addresses of destination host while also providing the destination with its own link layer address. The source knows the IP address of the destination but requires the link layer address to encapsulate packet into the frame.
- **Neighbour Advertisement message** (type = 136 and code = 0) propagates modifications quickly and is sent in response to a Neighbour Solicitation message.
- **Redirect messages** (type = 137 and code = 0) is sent by router to inform other nodes of a better first hop toward a destination. Hosts can be redirected to another router connected to the same link, but more commonly to another neighbour.
- **Inverse Neighbour Solicitation Message** (type = 141 and code = 0) is sent by a host that knows the link-layer address of a neighbour, but not the neighbour's IP address.
- **Inverse Neighbour Advertisement message** (type = 141 and code = 0) is sent in response to the Inverse Neighbour Solicitation message.

14.4.1.4 GROUP MEMBERSHIP MESSAGE

- ICMP Group Membership messages are used to convey information about multicast group membership from nodes to their neighbouring routers.
- There are three types of messages in this category.
- **Group Membership Query message** (type = 130 and code = 0) is sent by the router to find the active group member which can be general, group-specific and group-and-source specific.
- **Group Membership Report message** (type 131 and code = 0) or **Group Membership Reduction message** (type 132 and code = 0) indicates the reporting or termination of the member.

14.5 TRANSITION FROM IPv4 TO IPv6

- If we want to send a request from an IPv4 address to an IPv6 address it is not possible because IPv4 and IPv6 transition is not compatible.
- Complete transition from IPv4 to IPv6 might not be possible because IPv6 is not backward compatible.

- The transition cannot also happen suddenly and it will take a considerable amount of time before every system in the Internet can move from IPv4 to IPv6.
- The transition must be smooth to prevent any problems in both systems.

14.5.1 STRATEGIES

- We have a few strategies that can be used to ensure slow and smooth transition from IPv4 to IPv6.

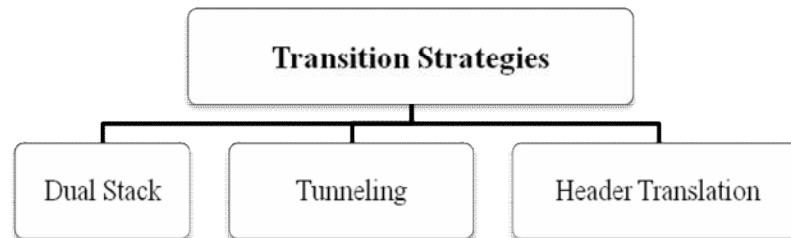


Figure 14.15 –Transition Strategies

- **Dual Stack** - The term dualstack normally refers to a complete duplication of all levels in the protocol stack from applications to the network layer.
- A router can be installed with both IPv4 and IPv6 addresses configured on its interfaces pointing to the network of relevant IP scheme.
- The source host queries the DNS to determine which version to use.
- If the DNS returns an IPv4 address, the source host sends an IPv4 packet and if it returns an IPv6 address, the source host sends an IPv6 packet.

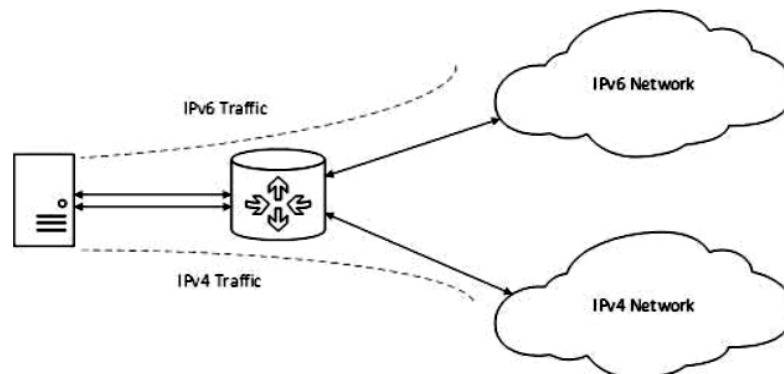


Figure 14.16 – Dual Stack Router

- The Dual Stack Router can now communicate with both the networks and provides medium for hosts to access server without changing the IP versions.
- **Tunneling** –It is used as a strategy to communicate the transit network with the different IP versions.

- When two host using IPv6 format wants to communicate and let us assume the packet has to pass in between a region using IPv4 format, then tunneling is deployed.

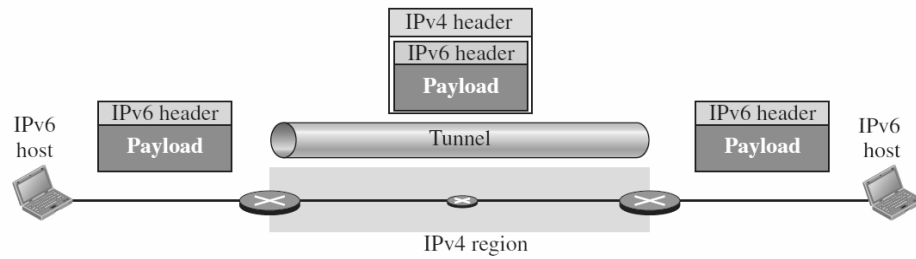


Figure 14.17 – Tunneling Scenario

- In order to reach the destination, the packet must have IPv4 address.
- When the packet enters the region, the IPv6 packet is encapsulated into IPv4 packet and when it leaves the region, decapsulation takes place.
- Vice versa situation may also arise when two IPv4 hosts wants to communicate and intermediate the packet has to pass to IPv6 region.
- In this case the IPv4 packet is encapsulated into IPv6 packet and when it leaves the region, decapsulation takes place.

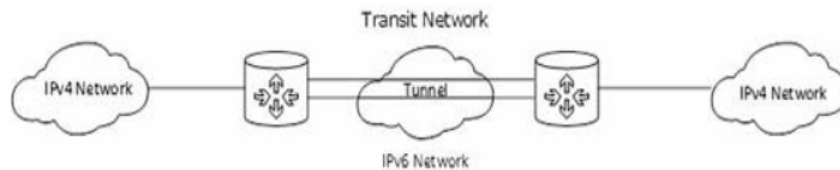


Figure 14.18 – Another Tunneling Scenario

- **Header Translation** - This is another important method of transition to IPv6 by means of a Network Address Translation – Protocol Translation (NAT-PT) enabled device.
- Tunneling is applicable when both source and destination host use the same format but header translation is applicable when source and destination uses different IP formats.
- For example, source host follows the IPv6 format and the destination follows the IPv4 format, then the header format must be changed through header translation.

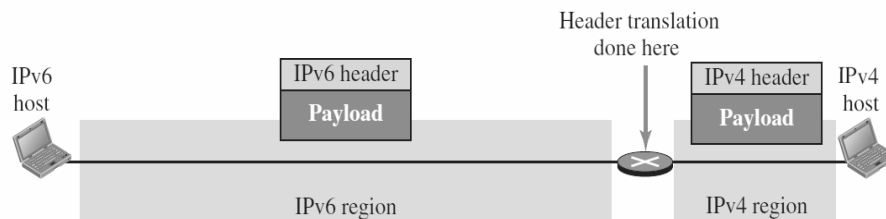


Figure 14.19 – Header Translation

- In the above example the NAT-PT router translates from IPv6 format to IPv4 format.

14.6 SUMMARY

- With Internet Protocol Version 6 (IPv6), everything from appliances to automobiles can be interconnected.
- Address depletion is not the only reason for the migration from IPv4 to IPv6 but reasons such as more efficient routing and packet processing, directed data flows, simplified network configuration, support for new services and lastly security has resulted in the transition.
- The IPv6 address is 128-bits which allows for over 2^{128} or 340 undecillion addresses.
- An IPv6 datagram is composed of a base header and a payload which consists of optional extension headers and data from an upper layer.
- Internet Control Message Protocol (ICMPv6) specifies a set of control messages for IPv6 for feedback, error reporting and network diagnostic functions.
- Three strategies used to handle the transition from IPv4 to IPv6 are dualstack, tunneling, and header translation.

14.7 LIST OF REFERENCES

- “Data Communications and Networking” by Behrouz A. Forouzan, 5th Edition, McGraw-Hill Publication.
- <https://www.juniper.net>
- <https://www.tutorialspoint.com>
- <https://www.geeksforgeeks.org>
- <https://www.networkcomputing.com>

14.8 UNIT END EXERCISE

1. Explain the advantages of IPv6 when compared to IPv4.
2. Compare and contrast the IPv4 header with the IPv6 header. Create a table to compare each field.
3. Show abbreviations for the following addresses:
 - a. An address with 64 0s followed by 32 two-bit (01)s.
 - b. An address with 64 0s followed by 32 two-bit (10)s.
 - c. 0000:FFFF:FFFF:0000:0000:0000:0000:0000
 - d. 1234:2346:3456:0000:0000:0000:0000:FFFF

4. Decompress the following addresses and show the complete unabbreviated IPv6 address:
 - a. ::2222
 - b. 0:23::0
 - c. B:A:CC::1234:A
 - d. 0:A::3
5. An organization is assigned the block 2000:1110:1287/48. What is the IPv6 address of an interface in the third subnet if the IEEE physical address of the computer is (F5-A9-23-14-7A-D2)₁₆.



INTRODUCTION TO TRANSPORT LAYER

Unit Structure :

- 15.0 Objectives
- 15.1 Introduction
- 15.2 Transport Layer services
- 15.3 Transport Layer Protocols
 - 15.3.1 Simple Protocol
 - 15.3.2 Stop-and-wait Protocol
 - 15.3.3 Go-Back-N Protocol
 - 15.3.4 Selective-Repeat Protocol
 - 15.3.5 Bidirectional Protocols
- 15.4 User Datagram Protocol (UDP)
- 15.5 Transmission Control Protocol (TCP)
- 15.6 Summary
- 15.7 Reference for further reading
- 15.8 Model Questions

15.0 OBJECTIVES:

This chapter would make you to understand the following concepts:

- How process to process communication provided at the transport layer.
- Various Services provided at the Transport Layer.
- Flow control and how it can be achieved at the transport layer.
- Error control and how it can be achieved at the transport layer.
- Congestion control and how it can be achieved at the transport layer.
- Transport layer protocols – Simple protocol, Stop-and-wait protocol, Go-Back-N protocol, Selective-Repeat protocol and Bidirectional protocols.
- User Datagram protocol (UDP) and Transmission Control protocol (TCP).

15.1 INTRODUCTION

Transport layer is present between the network layer and application layer. It is responsible for providing services to the application layer; it receives services from the network layer. In this chapter, we can discuss the various services that can be provided by a transport layer and different protocols present in the transport layer.

15.2 TRANSPORT LAYER SERVICES

The transport layer provides the various services such as Process-to-Process Communication, Addressing: Port Numbers, Encapsulation and De-capsulation, Multiplexing and De-multiplexing, Flow Control, Error Control, Congestion Control, Connectionless and Connection-Oriented.

Process-to-Process Communication

First responsibility of a transport layer protocol is to provide process-to-process communication.

A process is an application-layer entity (running program) that uses the services of the transport layer. As we know the network layer is responsible for communication at the computer level (host-to-host communication). A network layer protocol can deliver the message only to the destination computer. However, this is an incomplete delivery. The message still needs to be hand over to the correct process. This is where a transport layer protocol is responsible for delivery of the message to the appropriate process. Figure 15.1 shows the domains of a network layer and a transport layer.

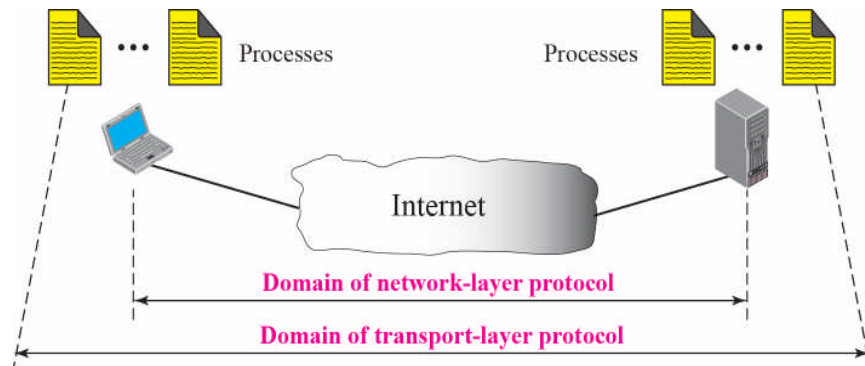


Figure 15.1: Network layer versus Transport layer

Addressing: Port Numbers

Although there are a few ways to achieve process-to-process communication, the most common is through the **client-server paradigm**. A process on the local host, called a *client*, needs services from a process usually on the remote host, called a *server*.

Both processes (client and server) have the same name. For example, to get the day and time from a remote machine, we need a daytime client process running on the local host and a daytime server process running on a remote machine.

For this communication, we must define the Local host, Local process, Remote host and Remote process. The local host and the remote host are defined by using IP addresses. To define the processes, we need second identifiers called **port numbers**. In the TCP/IP protocol suite, the port numbers are integers between 0 and 65,535.

The client program defines itself with a port number, called the **ephemeral port number** (short lived). An ephemeral port number is recommended to be greater than 1,023 for some client/server programs to work properly.

The server process must also define itself with a port number. TCP/IP has decided to use universal port numbers for servers; these are called **well-known port numbers** (always less than 1024). Every client process knows the well-known port number of the corresponding server process. For example, while the daytime client process, discussed above, can use an ephemeral (temporary) port number 52,000 to identify itself, the daytime server process must use the well-known (permanent) port number 13. Figure 15.2 shows this concept.

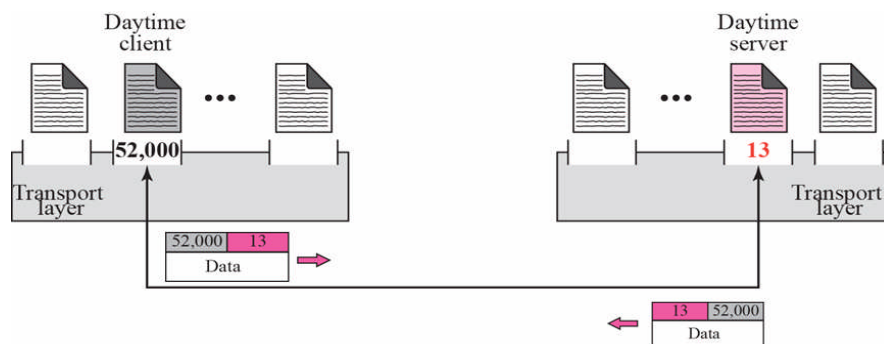


Figure 15.2: Port numbers

It should be clear by now that the IP addresses and port numbers play different roles in selecting the final destination of data. The destination IP address defines the host among the different hosts in the world. After the host has been selected, the port number defines one of the processes on this particular host (see Figure 15.3).

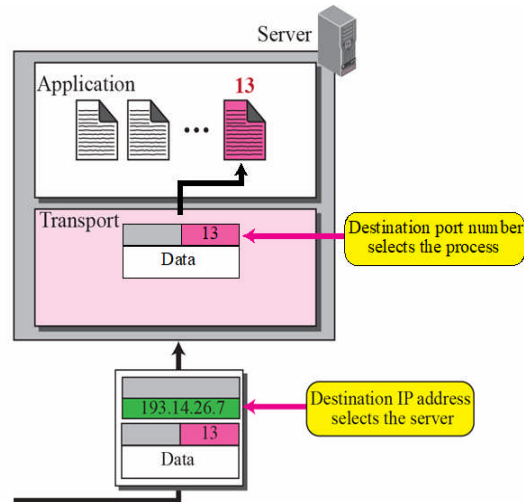


Figure 15.3: IP address versus Port numbers

Socket Addresses

A transport-layer protocol in the TCP suite needs both the IP address and the port number, at each end, to make a connection. The combination of an IP address and a port number is called a **socket address**. The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely (see Figure 15.4).

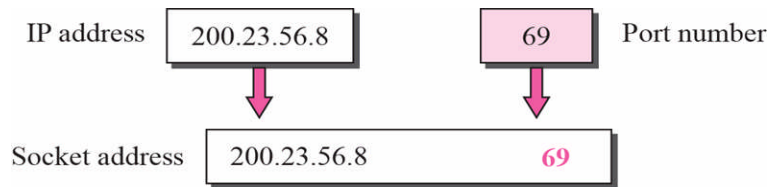


Figure 15.4: Socket address

Encapsulation and De-capsulation

To send a message from one process to another, the transport layer protocol encapsulates and de-capsulates messages (Figure 15.5).

Encapsulation happens at the sender site. When a process has a message to send, it passes the message to the transport layer along with a pair of socket addresses and some other pieces of information that depends on the transport layer protocol. The transport layer receives the data and adds the transport-layer header. The packets at the transport layers in the Internet are called *user datagrams*, *segments*, or *packets*. We call them packets in this chapter.

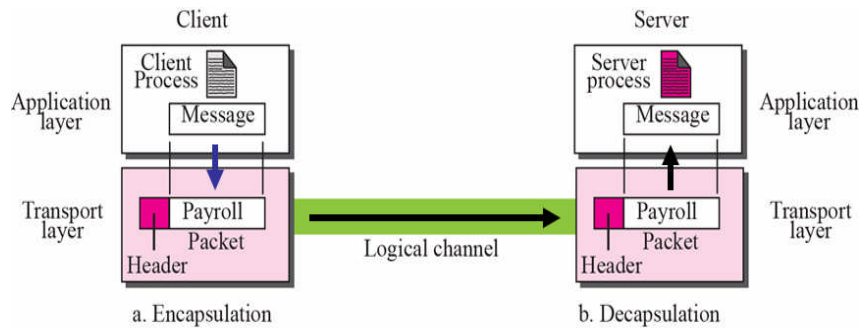


Figure 15.5: Encapsulation and De-capsulation

De-capsulation happens at the receiver site. When the message arrives at the destination transport layer, the header is dropped and the transport layer delivers the message to the process running at the application layer. The sender socket address is passed to the process in case it needs to respond to the message received.

Multiplexing and De-multiplexing

Whenever an entity accepts items from more than one source, it is referred to as **multiplexing** (many to one); whenever an entity delivers items to more than one source, it is referred to as **de-multiplexing** (one to many). The transport layer at the source performs multiplexing; the transport layer at the destination performs de-multiplexing (Figure 15.6).

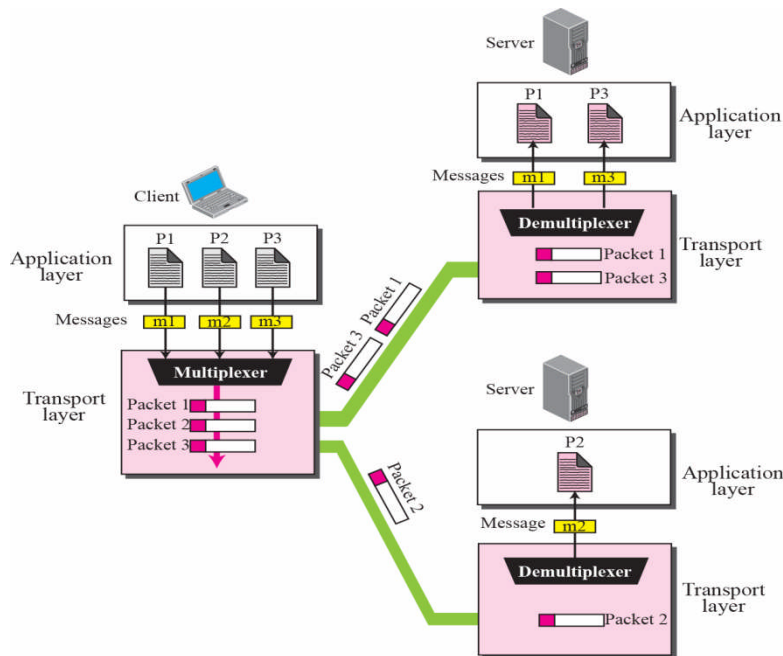


Figure 15.6: Multiplexing and De-multiplexing

Figure 15.6 shows communication between a client and two servers. Three client processes are running at the client site, P1, P2, and P3. The processes P1 and P3 need to send requests to the corresponding server process running in a server. The client process P2 needs to send a request to the corresponding server process running at another server. The

transport layer at the client site accepts three messages from the three processes and creates three packets. It acts as a *multiplexer*. The packets 1 and 3 use the same logical channel to reach the transport layer of the first server. When they arrive at the server, the transport layer does the job of a *de-multiplexer* and distributes the messages to two different processes. The transport layer at the second server receives packet 2 and delivers it to the corresponding process.

Flow Control at Transport Layer

In communication at the transport layer, we are dealing with four entities: sender process, sender transport layer, receiver transport layer, and receiver process. The sending process at the application layer is only a producer. It produces message chunks and pushes them to the transport layer. The sending transport layer has a double role: it is both a consumer and the producer. It consumes the messages pushed by the producer. It encapsulates the messages in packets and pushes them to the receiving transport layer. The receiving transport layer has also a double role: it is the consumer for the packets received from the sender. It is also a producer; it needs to de-capsulate the messages and delivers them to the application layer. The last delivery, however, is normally a pulling delivery; the transport layer waits until the application-layer process asks for messages.

The receiving transport layer has also a double role: it is the consumer for the packets received from the sender. It is also a producer; it needs to de-capsulate the messages and delivers them to the application layer. The last delivery, however, is normally a pulling delivery; the transport layer waits until the application-layer process asks for messages.

Figure 15.7 shows that we need at least two cases of flow control: from the sending transport layer to the sending application layer and from the receiving transport layer to the sending transport layer.

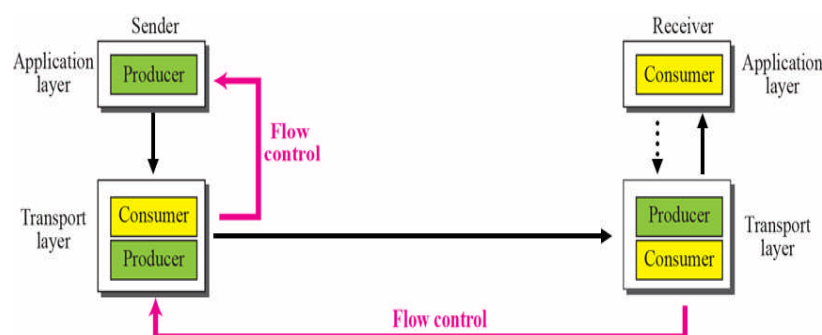


Figure 15.7: Flow control at transport layer

Buffers

Although flow control can be implemented in several ways, one of the solutions is normally to use two *buffers*. One at the sending transport layer and the other at the receiving transport layer. A buffer is a set of memory locations that can hold packets at the sender and receiver. The flow control communication can occur by sending signals from the consumer to producer. When the buffer of the sending transport layer is full, it informs the application layer to stop passing chunks of messages; when there are some vacancies, it informs the application layer that it can pass message chunks again.

When the buffer of the receiving transport layer is full, it informs the sending transport layer to stop sending packets. When there are some vacancies, it informs the sending transport layer that it can send message again.

Error Control

In the Internet, since the underlying network layer (IP), which is responsible to carry the packets from the sending transport layer to the receiving transport layer, is unreliable, we need to make the transport layer reliable if the application requires reliability.

Reliability can be achieved to add error control service to the transport layer. Error control at the transport layer is responsible to

1. Detect and discard corrupted packets.
2. Keep track of lost and discarded packets and resend them.
3. Recognize duplicate packets and discard them.
4. Buffer out-of-order packets until the missing packets arrive.

Error control, unlike the flow control, involves only the sending and receiving transport layers. We are assuming that the message chunks exchanged between the application and transport layers are error free. Figure 15.8 shows the error control between the sending and receiving transport layer. As with the case of flow control, the receiving transport layer manages error control, most of the time, by informing the sending transport layer about the problems.



Figure 15.8: Error control at Transport layer

Sequence Numbers

Error control requires that the sending transport layer knows which packet is to be resent and the receiving transport layer knows which packet is a duplicate, or which packet has arrived out of order. This can be done if the packets are numbered. We can add a field to the transport layer packet to hold the **sequence number** of the packets.

When a packet is corrupted or lost, the receiving transport layer can somehow inform the sending transport layer to resend that packet using the sequence number. The receiving transport layer can also detect duplicate packets if two received packets have the same sequence number. The out-of-order packets can be recognized by observing gaps in the sequence numbers.

Packets are numbered sequentially. However, because we need to include the sequence number of each packet in the header, we need to set a limit. If the header of the packet allows m bits for the sequence number, the sequence numbers range from 0 to $2^m - 1$. For example, if m is 4, the only sequence numbers are 0 through 15, inclusive. However, we can wrap around the sequence. So the sequence numbers in this case are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ... In other words, the sequence numbers are modulo 2^m .

Acknowledgment

We can use both positive and negative signals as error control. The receiver side can send an acknowledgement (ACK) for each or a collection of packets that have arrived safe and sound. The receiver can simply discard the corrupted packets. The sender can detect lost packets if it uses a timer. When a packet is sent, the sender starts a timer; when the timer expires, if an ACK does not arrive before the timer expires, the sender resends the packet. Duplicate packets can be silently discarded by the receiver. Out-of-order packets can be either discarded (to be treated as lost packets by the sender), or stored until the missing ones arrives.

Combination of Flow and Error Control

We have discussed that flow control requires the use of two buffers, one at the sender site and the other at the receiver site. We have also discussed that the error control requires the use of sequence and acknowledgment numbers by both sides. These two requirements can be combined if we use two numbered buffers, one at the sender, and one at the receiver.

At the sender, when a packet is prepared to be sent, we use the number of the next free location, x , in the buffer as the sequence number of the packet. When the packet is sent, a copy is stored at memory location x , awaiting the acknowledgment from the other end. When an acknowledgment related to a sent packet arrives, the packet is purged and the memory location becomes free.

At the receiver, when a packet with sequence number y arrives, it is stored at the memory location y until the application layer is ready to receive it. An acknowledgment can be sent to announce the arrival of packet y .

Sliding Window

Since the sequence numbers used modulo 2^m , a circle can represent the sequence number from 0 to $2^m - 1$ (see Figure 15.9).

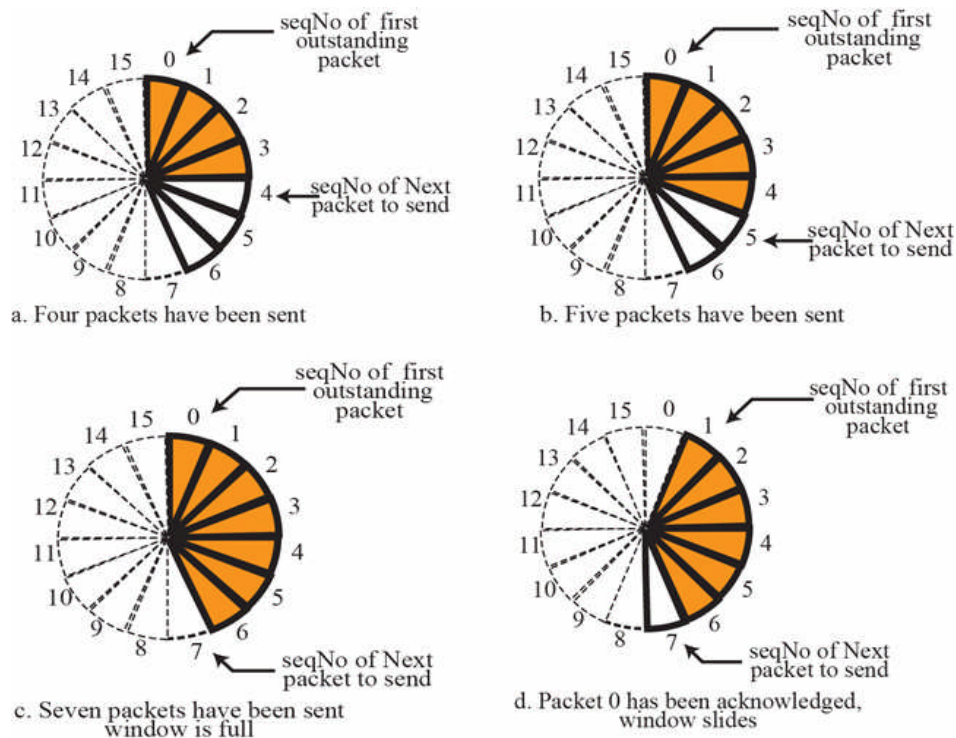


Figure 15.9: Sliding window in circular format

The buffer is represented as a set of slices, called the **sliding window** that occupy part of the circle at any time. At the sender site, when a packet is sent, the corresponding slice is marked. When all the slices are marked, it means that the buffer is full and no further messages can be accepted from the application layer. When an acknowledgment arrives, the corresponding slice is unmarked. If some consecutive slices from the beginning of the window are unmarked, the window slides over the range of the corresponding sequence number to allow more free slices at the end of the window. Figure 15.9 shows the sliding window at the sender. The sequence number are modulo 16 ($m = 4$) and the size of the window is 7. Note that the sliding window is just an abstraction: the actual situation uses computer variables to hold the sequence number of the next packet to be sent and the last packet sent.

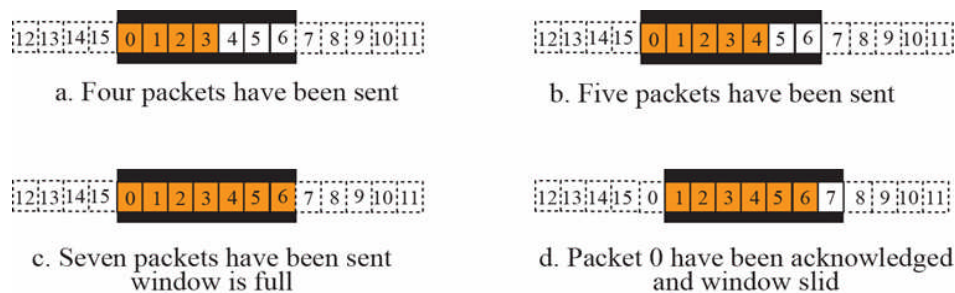


Figure 15.10: Sliding window in linear format

Most protocols show the sliding window using linear representation. The idea is the same, but it normally takes less space on paper. Figure 15.10 shows this representation.

Both representations tell us the same thing. If we take both sides of each part in Figure 15.9 and bend them up, we can make the same part in Figure 15.10.

Congestion Control

An important issue in the Internet is **congestion**. Congestion in a network may occur if the **load** on the network - the number of packets sent to the network - is greater than the *capacity* of the network - the number of packets a network can handle. **Congestion control** refers to the mechanisms and techniques to control the congestion and keep the load below the capacity.

We may ask why there is congestion on a network. Congestion happens in any system that involves waiting. For example, congestion happens on a freeway because any abnormality in the flow, such as an accident during rush hour, creates blockage.

Congestion in a network or internet work occurs because routers and switches have queues - buffers that hold the packets before and after processing. The packet is put in the appropriate output queue and waits its turn to be sent. These queues are finite, so it is possible for more packets to arrive at a router than the router can buffer.

Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened.

Open-Loop Congestion Control

In **open-loop congestion control**, policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination.

Retransmission Policy: Retransmission is sometimes unavoidable. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. Retransmission in general may increase congestion in the network. However, a good retransmission policy can prevent congestion. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion.

Window Policy: The type of window at the sender may also affect congestion. We will see later in the chapter that the *Selective Repeat* window is better than the *Go-Back-N* window for congestion control.

Acknowledgment Policy: The acknowledgment policy imposed by the receiver may also affect congestion. If the receiver does not acknowledge every packet it receives, it may slow down the sender and help to prevent congestion. Several approaches are used in this case. A receiver may send an acknowledgment only if it has a packet to be sent or a special timer expires. A receiver may decide to acknowledge only N packets at a time. We need to know that the acknowledgments are also part of the load in a

network. Sending fewer acknowledgments means imposing less load on the network.

Closed-Loop Congestion Control

Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols. We describe the one used in the transport layer. The size of the window at the sender size can be flexible. One factor that can determine the sender window size is the congestion in the Internet. The sending transport layer can monitor the congestion in the Internet, by watching the lost packets, and use a strategy to decrease the window size if the congestion is increasing and vice versa.

Connectionless and Connection-Oriented Services

A transport-layer protocol, like a network-layer protocol can provide two types of services: connectionless and connection-oriented. The nature of these services at the transport layer, however, is different from the ones at the network layer. At the network layer, a connectionless service may mean different paths for different datagram belonging to the same message.

At the transport layer, we are not concerned about the physical paths of packets (we assume a logical connection between two transport layers), connectionless service at the transport layer means independency between packets; connection-oriented means dependency. Let us elaborate on these two services.

Connectionless Service

In a connectionless service, the source process (application program) needs to divide its message into chunks of data of the size acceptable by the transport layer and deliver them to the transport layer one by one. The transport layer treats each chunk as a single unit without any relation between the chunks. When a chunk arrives from the application layer, the transport layer encapsulates it in a packet and sends it. To show the independency of packets, assume that a client process has three chunks of messages to send a server process. The chunks are handed over to the connectionless transport protocol in order. However, since there is no dependency between the packets at the transport layer, the packets may arrive out of order at the destination and will be delivered out of order to the server process. In Figure 15.11, we have shown the movement of packets using a time line, but we have assumed that the deliveries of the process to the transport layer and vice versa are instant.

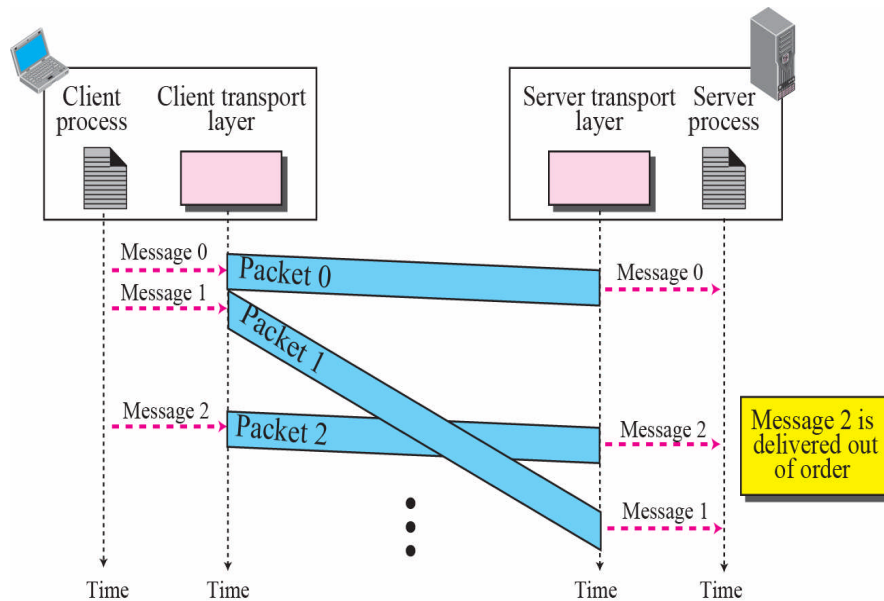


Figure 15.11: Connectionless service

Connection-Oriented Service

In a connection-oriented service, the client and the server first need to establish a connection between them. The data exchange can only happen after the connection establishment. After data exchange, the connection needs to be teared down (Figure 15.12). As we mentioned before, the connection-oriented service at the transport layer is different from the same service at the network layer. In the network layer, connection-oriented service means coordination between the two end hosts and all the routers in between. At the transport layer, connection-oriented service involves only the two hosts; the service is end to end. This means that we should be able to make a connection-oriented protocol over either a connectionless or connection-oriented protocol. Figure 15.12 shows the connection establishment, data transfer, and teardown phases in a connection-oriented service at the transport layer. We can implement flow control, error control, and congestion control in a connection oriented protocol.

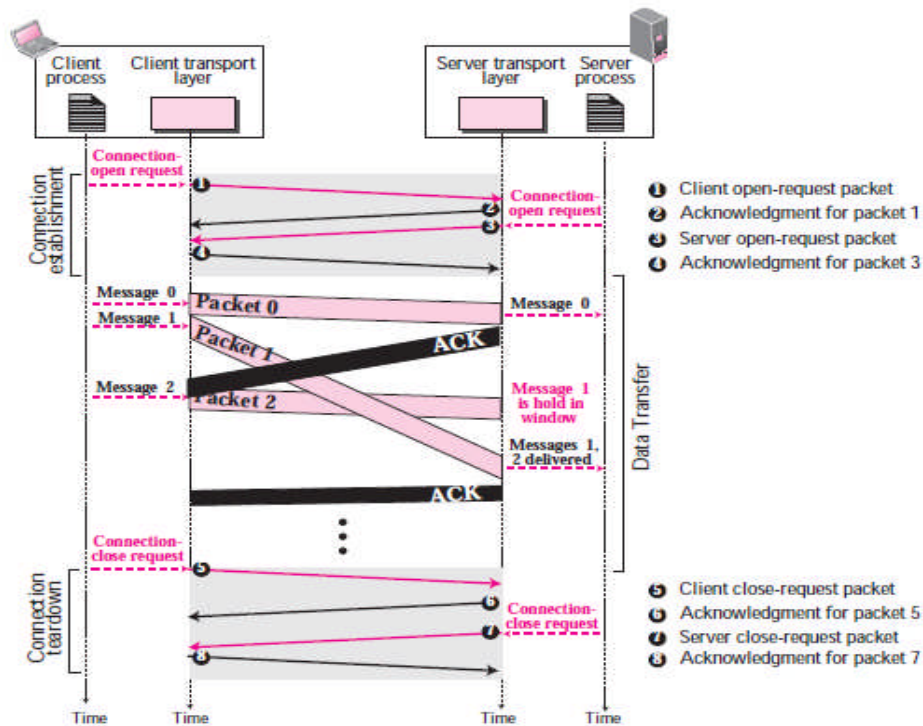


Figure 15.12: Connection-oriented service

15.3 Transport Layer Protocols

We can create a transport-layer protocol by combining a set of services described in the previous sections. To better understand the behavior of these protocols, we start with the simplest one and gradually add more complexity. The TCP/IP protocol uses a transport layer protocol that is either a modification or a combination of some of these protocols.

15.3.1 Simple Protocol

Our first protocol is a simple connectionless protocol which does not provide either flow or error control. We assume that the receiver can immediately handle any packet it receives. In other words, the receiver can never be overwhelmed with incoming packets. Figure 15.13 shows the layout for this protocol.

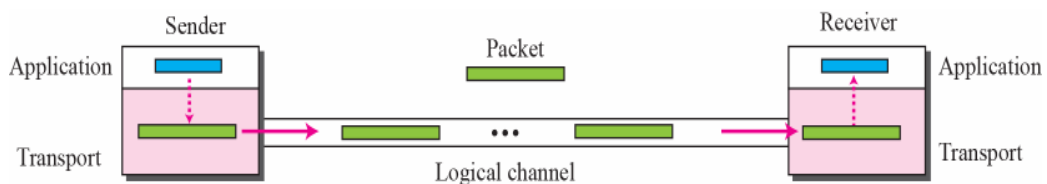


Figure 15.13: Simple protocol

The transport layer at the sender gets a message from its application layer, makes a packet out of it, and sends the packet. The transport layer at the receiver receives a packet from its network layer, extracts the message from the packet, and delivers the message to its

application layer. The transport layers of the sender and receiver provide transmission services for their application layers.

Example 15.1:

Figure 15.14 shows an example of communication using this protocol. It is very simple. The sender sends packets one after another without even thinking about the receiver.

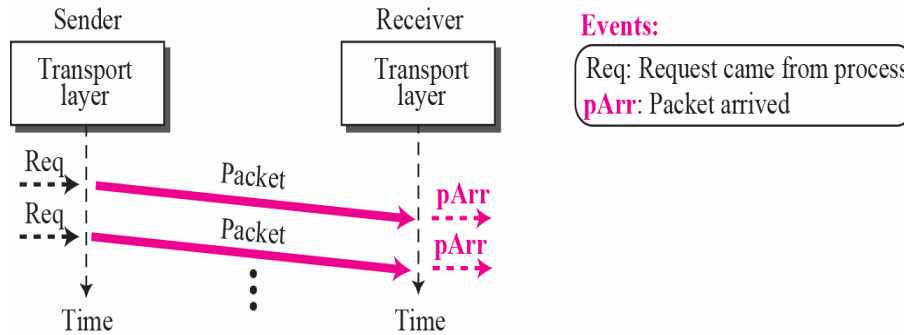


Figure 15.14: Example 15.1

15.3.2 Stop-and-wait Protocol

Our second protocol is a connection-oriented protocol called the **Stop-and-Wait protocol**, which provides both flow and error control. Both the sender and the receiver use a sliding window of size 1. The sender sends one packet at a time and waits for an acknowledgment before sending the next one. To detect corrupted packets, we need to add a checksum to each data packet. When a packet arrives at the receiver site, it is checked. If its checksum is incorrect, the packet is corrupted and silently discarded.

The silence of the receiver is a signal for the sender that a packet was either corrupted or lost. Every time the sender sends a packet, it starts a timer. If an acknowledgment arrives before the timer expires, the timer is stopped and the sender sends the next packet (if it has one to send). If the timer expires, the sender resends the previous packet assuming that either the packet was lost or corrupted. This means that the sender needs to keep a copy of the packet until its acknowledgment arrives. Figure 15.15 shows the outline for the Stop-and-Wait protocol. Note that only one packet and one acknowledgment can be in the channels at any time.

The Stop-and-Wait protocol is a connection-oriented protocol that provides flow and error control.

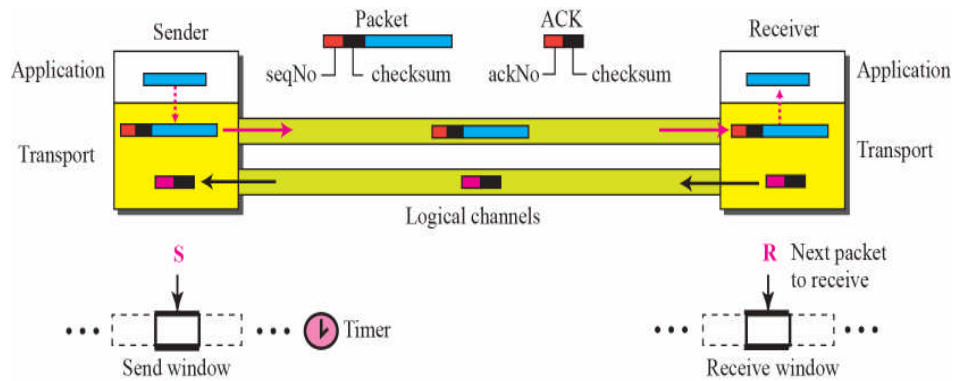


Figure 15.15: Stop-and-wait protocol

Sequence Numbers

To prevent duplicate packets, the protocol uses sequence numbers and acknowledgment numbers. A field is added to the packet header to hold the sequence number of that packet. Assume we have used x as a sequence number; we only need to use $x + 1$ after that. There is no need for $x + 2$. To show this, assume that the sender has sent the packet with sequence number x . Three things can happen.

1. The packet arrives safe and sound at the receiver site; the receiver sends an acknowledgment. The acknowledgment arrives at the sender site, causing the sender to send the next packet numbered $x + 1$.
2. The packet is corrupted or never arrives at the receiver site; the sender resends the packet (numbered x) after the time-out. The receiver returns an acknowledgment.
3. The packet arrives safe and sound at the receiver site; the receiver sends an acknowledgment, but the acknowledgment is corrupted or lost. The sender resends the packet (numbered x) after the time-out. Note that the packet here is a duplicate.

The receiver can recognize this fact because it expects packet $x + 1$ but packet x was received. We can see that there is a need for sequence numbers x and $x + 1$ because the receiver needs to distinguish between case 1 and case 3. But there is no need for a packet to be numbered $x + 2$. In case 1, the packet can be numbered x again because packets x and $x + 1$ are acknowledged and there is no ambiguity at either site. In cases 2 and 3, the new packet is $x + 1$, not $x + 2$. If only x and $x + 1$ are needed, we can let $x = 0$ and $x + 1 = 1$. This means that the sequence is 0, 1, 0, 1, 0, and so on. This is referred to as modulo-2 arithmetic.

Acknowledgment Numbers

Since the sequence numbers must be suitable for both data packets and acknowledgments, we use this convention: The acknowledgment numbers always announce the sequence number of the *next packet expected* by the receiver. For example, if packet 0 has arrived safe and sound, the receiver sends an ACK with acknowledgment 1 (meaning packet 1 is expected next). If packet 1 has arrived safe and sound, the

receiver sends an ACK with acknowledgment 0 (meaning packet 0 is expected).

The sender has a control variable, which we call **S** (sender), that points to the only slot in the send window. The receiver has a control variable, which we call **R** (receiver), that points to the only slot in the receive window.

Example 15.2:

Figure 15.16 shows an example of Stop-and-Wait protocol. Packet 0 is sent and acknowledged. Packet 1 is lost and resent after the time-out. The resent packet 1 is acknowledged and the timer stops. Packet 0 is sent and acknowledged, but the acknowledgment is lost. The sender has no idea if the packet or the acknowledgment is lost, so after the time-out, it resends packet 0, which is acknowledged.

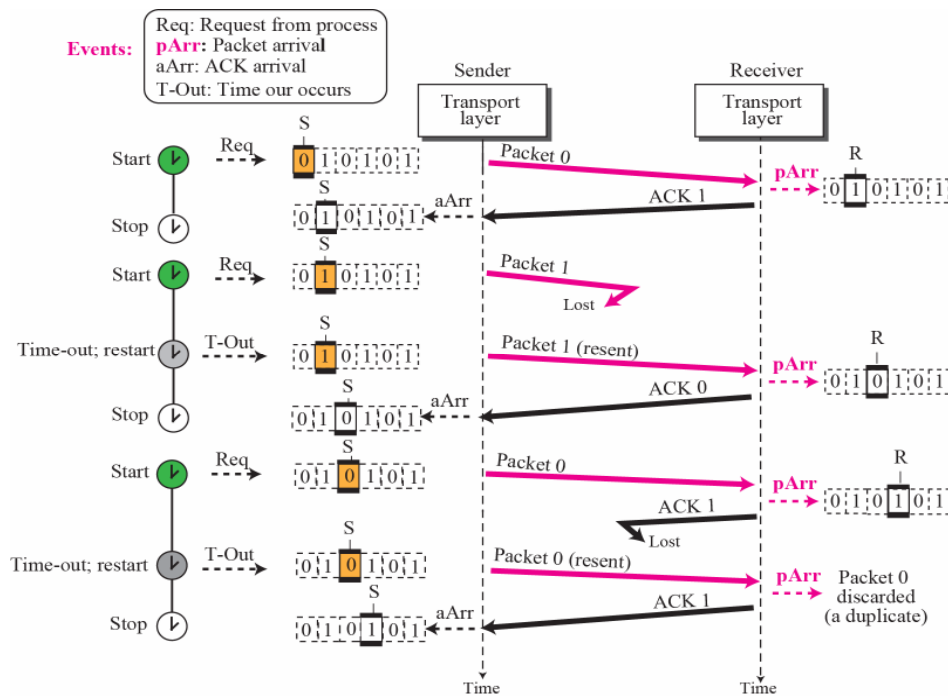


Figure 15.16: Example 15.2

Efficiency

The Stop-and-Wait protocol is very inefficient if our channel is *thick* and *long*. By *thick*, we mean that our channel has a large bandwidth (high data rate); by *long*, we mean the round-trip delay is long. The product of these two is called the **bandwidth-delay product**.

We can think of the channel as a pipe. The bandwidth-delay product then is the volume of the pipe in bits. The pipe is always there. If we do not use it, we are inefficient. The bandwidth-delay product is a measure of the number of bits a sender can transmit through the system while waiting for an acknowledgment from the receiver.

Example 15.3:

Assume that, in a Stop-and-Wait system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 milliseconds to make a round trip. What is the bandwidth-delay product? If the system data packets are 1,000 bits in length, what is the utilization percentage of the link?

Solution:

The bandwidth-delay product is $(1 \times 10^6) \times (20 \times 10^{-3}) = 20,000$ bits. The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and the acknowledgment to come back. However, the system sends only 1,000 bits. We can say that the link utilization is only $1,000/20,000$, or 5 percent. For this reason, for a link with a high bandwidth or long delay, the use of Stop-and-Wait wastes the capacity of the link.

Example 15.4:

What is the utilization percentage of the link in Example 15.3 if we have a protocol that can send up to 15 packets before stopping and worrying about the acknowledgments?

Solution:

The bandwidth-delay product is still 20,000 bits. The system can send up to 15 packets or 15,000 bits during a round trip. This means the utilization is $15,000/20,000$, or 75 percent. Of course, if there are damaged packets, the utilization percentage is much less because packets have to be resent.

Pipelining

In networking and in other areas, a task is often begun before the previous task has ended. This is known as **pipelining**. There is no pipelining in the Stop-and-Wait protocol because a sender must wait for a packet to reach the destination and be acknowledged before the next packet can be sent. However, pipelining does apply to our next two protocols because several packets can be sent before a sender receives feedback about the previous packets. Pipelining improves the efficiency of the transmission if the number of bits in transition is large with respect to the bandwidth-delay product.

15.3.3 Go-Back-N Protocol

To improve the efficiency of transmission, multiple packets must be in transition while the sender is waiting for acknowledgment. In other words, we need to let more than one packet be outstanding to keep the channel busy while the sender is waiting for acknowledgment. In this section, we discuss one protocol that can achieve this goal; in the next section, we discuss a second. The first is called **Go-Back-N**. The key to Go-back- N is that we can send several packets before receiving acknowledgments, but the receiver can only buffer one packet. We keep a copy of the sent packets until the acknowledgments arrive. Figure 15.17 shows the outline of the protocol. Note that several data packets and acknowledgments can be in the channel at the same time.

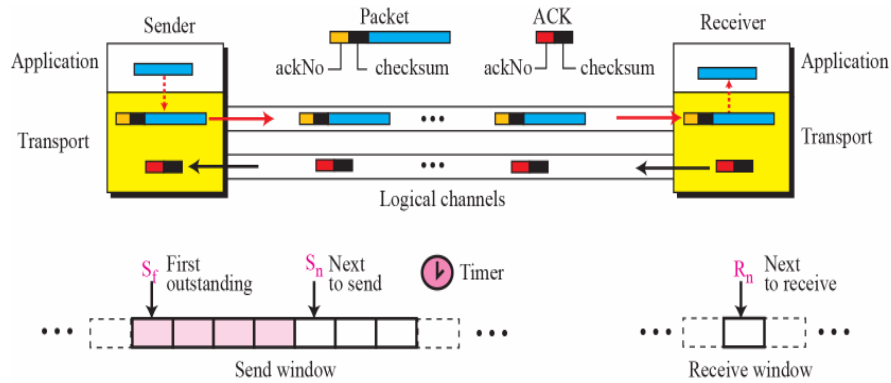


Figure 15.17: Go-Back-N protocol

Sequence Numbers

As we mentioned before, the sequence numbers are used modulo 2^m , where m is the size of the sequence number field in bits.

Acknowledgment Number

Acknowledgment number in this protocol is cumulative and defines the sequence number of the next packet expected. For example, if the acknowledgment number (ack No) is 7, it means all packets with sequence number up to 6 have arrived, safe and sound, and the receiver is expecting the packet with sequence number 7.

Send Window

The send window is an imaginary box covering the sequence numbers of the data packets that can be in transit or can be sent. In each window position, some of these sequence numbers define the packets that have been sent; others define those that can be sent. The maximum size of the window is $2^m - 1$. Figure 15.18 shows a sliding window of size 7 ($m = 3$) for the Go-Back-N protocol.

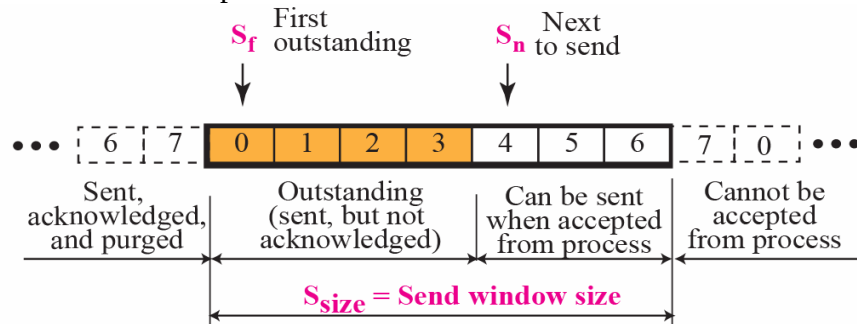


Figure 15.18: Send window for Go-Back-N protocol

The send window at any time divides the possible sequence numbers into four regions. *The first region*, left of the window, defines the sequence numbers belonging to packets that are already acknowledged. The sender does not worry about these packets and keeps no copies of them.

The second region, colored, defines the range of sequence numbers belonging to the packets that are sent, but have an unknown status. The sender needs to wait to find out if these packets have been received or were lost. We call these *outstanding* packets.

The third range, white in the figure, defines the range of sequence numbers for packets that can be sent; however, the corresponding data have not yet been received from the application layer. Finally, the fourth region, right of the window, defines sequence numbers that cannot be used until the window slides.

The window itself is an abstraction; three variables define its size and location at any time. We call these variables S_f (send window, the first outstanding packet), S_n (send window, the next packet to be sent), and S_{size} (send window, size). The variable S_f defines the sequence number of the first (oldest) outstanding packet. The variable S_n holds the sequence number that will be assigned to the next packet to be sent. Finally, the variable S_{size} defines the size of the window, which is fixed in our protocol.

Figure 15.19 shows how a send window can slide one or more slots to the right when an acknowledgment arrives from the other end. In the figure, an acknowledgment with ack No = 6 has arrived. This means that the receiver is waiting for packets with sequence number 6.

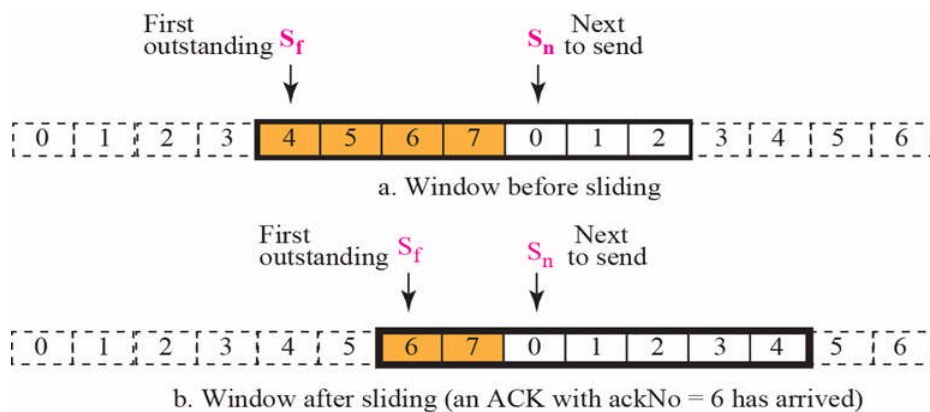


Figure 15.19: Sliding the send window

Receive Window

The receive window makes sure that the correct data packets are received and that the correct acknowledgments are sent. In Go-back- N , the size of the receive window is always 1. The receiver is always looking for the arrival of a specific packet. Any packet arriving out of order is discarded and needs to be resent. Figure 15.20 shows the receive window. Note that we need only one variable R_n (receive window, next packet expected) to define this abstraction.

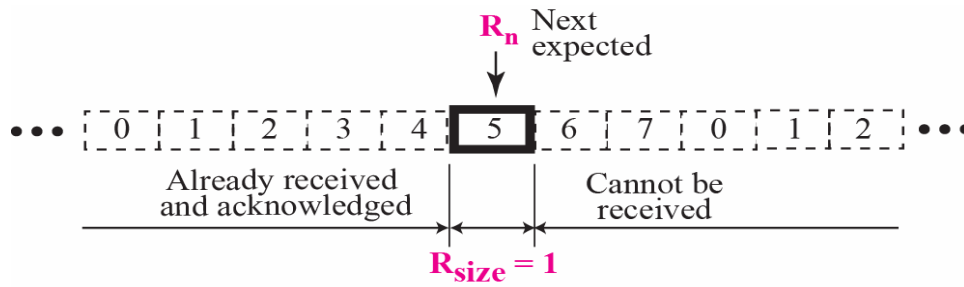


Figure 15.20: Receive window for Go-Back-N protocol

The sequence numbers to the left of the window belong to the packets already received and acknowledged; the sequence numbers to the right of this window define the packets that cannot be received. Any received packet with a sequence number in these two regions is discarded. Only a packet with a sequence number matching the value of R_n is accepted and acknowledged. The receive window also slides, but only one slot at a time. When a correct packet is received, the window slides, $R_n = (R_n + 1) \text{ modulo } 2^m$.

Timers

Although there can be a timer for each packet that is sent, in our protocol we use only one. The reason is that the timer for the first outstanding packet always expires first. We resend all outstanding packets when this timer expires.

Resending packets

When the timer expires, the sender resends all outstanding packets. For example, suppose the sender has already sent packet 6 ($S_n = 7$), but the only timer expires. If $S_f = 3$, this means that packets 3, 4, 5, and 6 have not been acknowledged; the sender goes back and resends packets 3, 4, 5, and 6. That is why the protocol is called Go-Back- N . On a time-out, the machine goes back N locations and resends all packets.

Example 15.5:

Figure 15.21 shows what happens when a packet is lost. Packets 0, 1, 2, and 3 are sent. However, packet 1 is lost. The receiver receives packets 2 and 3, but they are discarded because they are received out of order (packet 1 is expected). When the receiver receives packets 2 and 3, it sends ACK1 to show that it expects to receive packet 1. However, these ACKs are not useful for the sender because the ack No is equal to S_f , not greater than S_f . So the sender discards them. When the time-out occurs, the sender resends packets 1, 2, and 3, which are acknowledged.

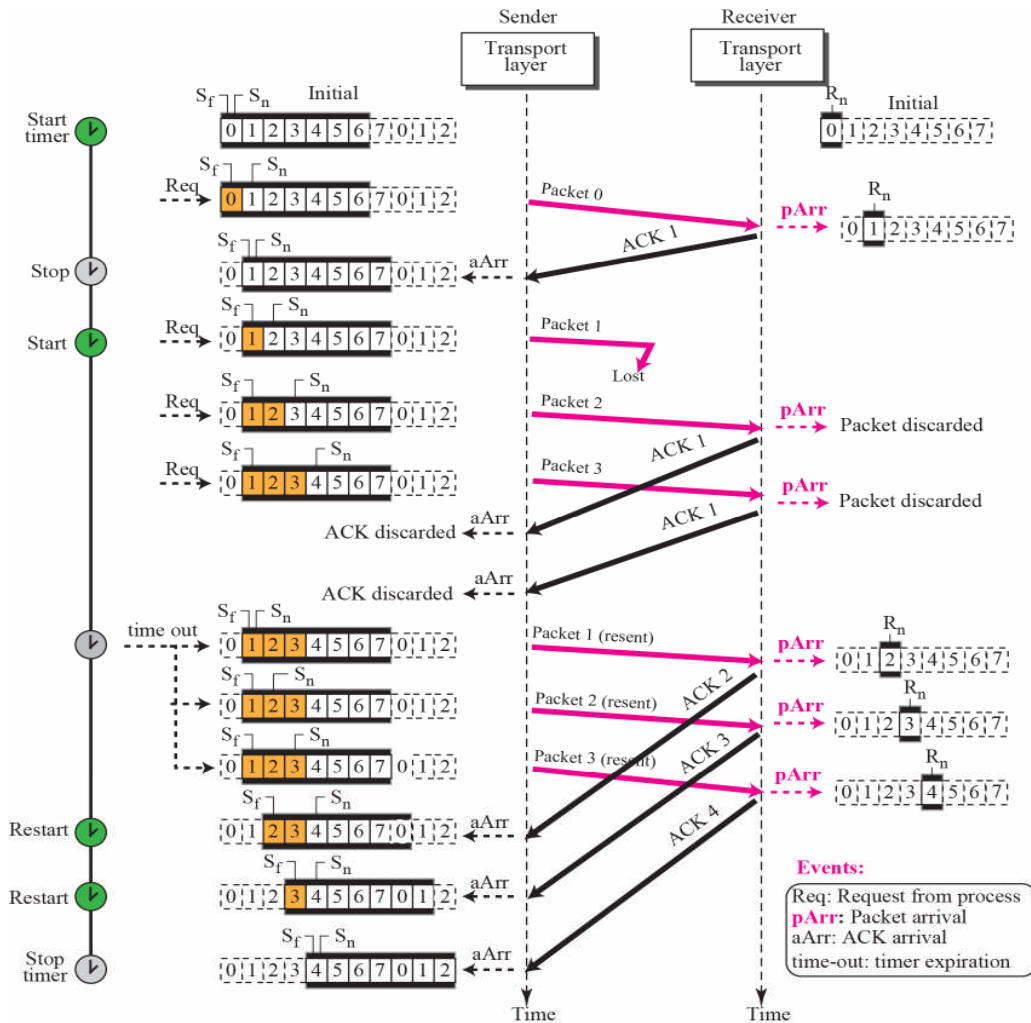


Figure 15.21: Example 15.5

15.3.4 Selective-Repeat Protocol

The Go-Back-N protocol simplifies the process at the receiver. The receiver keeps track of only one variable, and there is no need to buffer out-of-order packets; they are simply discarded. However, this protocol is inefficient if the underlying network protocol loses a lot of packets. Each time a single packet is lost or corrupted, the sender resends all outstanding packets although some of these packets may have been received safe and sound, but out of order. If the network layer is losing many packets because of congestion in the network, the resending of all of these outstanding packets makes the congestion worse, and eventually more packets are lost. This may result in the total collapse of the network. Another protocol, called the **Selective-Repeat protocol**, has been devised that, as the name implies, resends only selective packets, those that are actually lost. The outline of this protocol is shown in Figure 15.22.

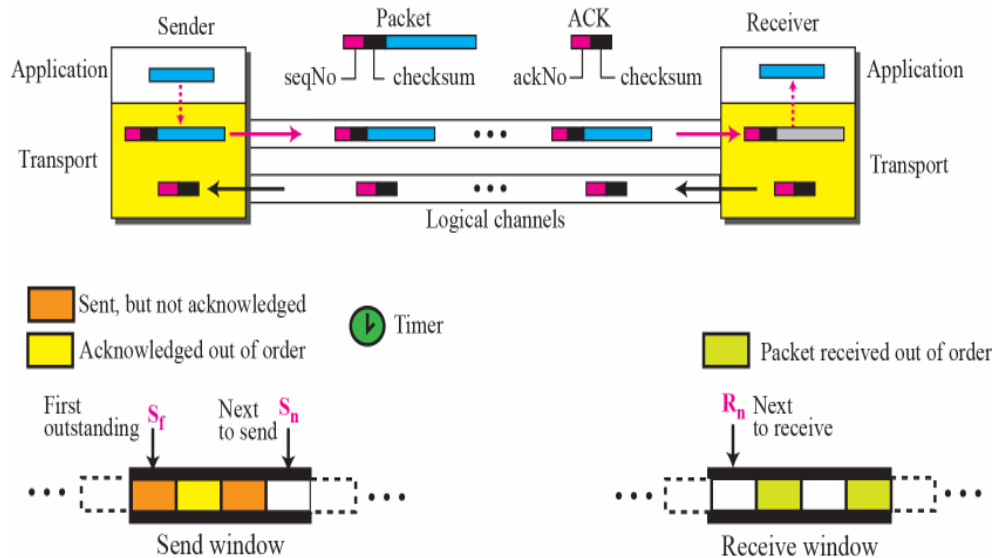


Figure 15.22: Selective-Repeat protocol

Windows

The send window maximum size can be $2^m - 1$. For example, if $m = 4$, the sequence numbers go from 0 to 15, but the maximum size of the window is just 8 (it is 15 in the Go-Back-N Protocol). We show the Selective-Repeat send window in Figure 15.23 to emphasize the size.

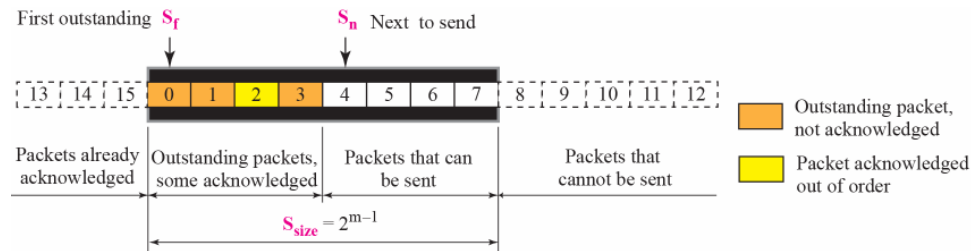


Figure 15.23: Send window for Selective-Repeat protocol

The receive window in Selective-Repeat is totally different from the one in Go-Back-N. The size of the receive window is the same as the size of the send window (maximum $2^m - 1$). The Selective-Repeat protocol allows as many packets as the size of the receive window to arrive out of order and be kept until there is a set of consecutive packets to be delivered to the application layer. Because the sizes of the send window and receive window are the same, all the packets in the send window can arrive out of order and be stored until they can be delivered. We need, however, to emphasize that in a reliable protocol, the receiver never delivers packets out of order to the application layer. Figure 15.24 shows the receive window in the Selective-Repeat. Those slots inside the window that are shaded define packets that have arrived out of order and are waiting for the earlier transmitted packet to arrive before delivery to the application layer.

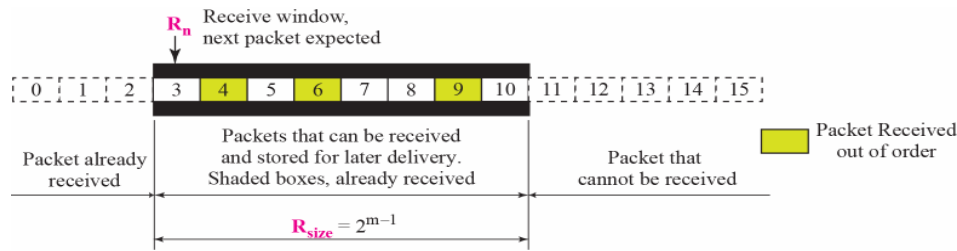


Figure 15.24: Receive window for Selective-Repeat protocol

Timer

Theoretically, Selective-Repeat uses one timer for each outstanding packet. When a timer expires, only the corresponding packet is resent. In other words, GBN treats outstanding packets as a group; Selective-Repeat treats them individually.

Acknowledgments

Still there is another difference between the two protocols. In Go-Back-N an ack No is cumulative; it defines the sequence number of the next packet expected, confirming that all previous packets have been received safe and sound. The semantics of acknowledgment is different in Selective-Repeat. In Selective-Repeat, an ack No defines the sequence number of one single packet that is received safe and sound; there is no feedback for any other.

Example 15.6:

This example is similar to Example 15.5 (Figure 15.21) in which packet 1 is lost. We show how Selective-Repeat behaves in this case. Figure 15.25 shows the situation.

At the sender, packet 0 is transmitted and acknowledged. Packet 1 is lost. Packets 2 and 3 arrive out of order and are acknowledged. When the timer times out, packet 1 (the only unacknowledged packet) is resent and is acknowledged. The send window then slides.

Window Sizes in Selective-Repeat

In Selective-Repeat, the size of the sender and receiver window can be at most one-half of 2^m .

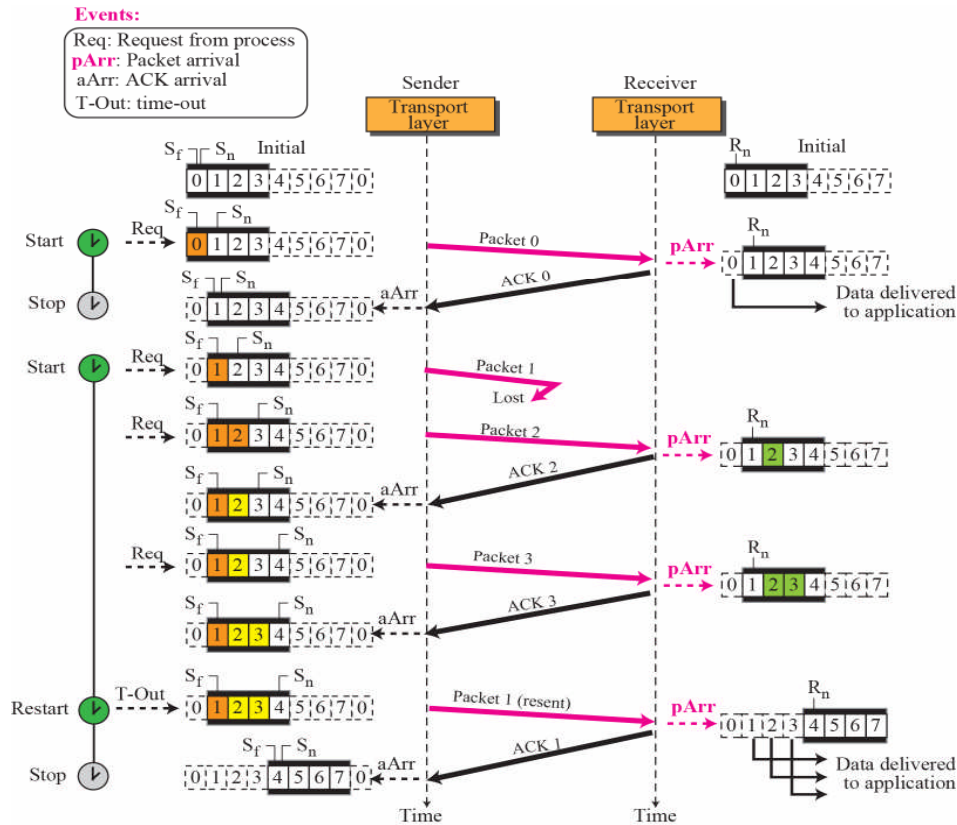


Figure 15.25: Example 15.6

At the receiver site we need to distinguish between the acceptance of a packet and its delivery to the application layer. At the second arrival, packet 2 arrives and is stored and marked (shaded slot), but it cannot be delivered because packet 1 is missing. At the next arrival, packet 3 arrives and is marked and stored, but still none of the packets can be delivered. Only at the last arrival, when finally a copy of packet 1 arrives, can packets 1, 2, and 3 be delivered to the application layer. There are two conditions for the delivery of packets to the application layer: First, a set of consecutive packets must have arrived. Second, the set starts from the beginning of the window.

After the first arrival, there was only one packet and it started from the beginning of the window. After the last arrival, there are three packets and the first one starts from the beginning of the window. The key is that a reliable transport layer promises to deliver packets *in order*.

15.3.5 Bidirectional Protocols – Piggybacking

The four protocols we discussed in this section are all unidirectional: data packets flow in only one direction and acknowledgments travel in the other direction. In real life, data packets are normally flowing in both directions: from client to server and from server to client. This means that acknowledgments also need to flow in both directions.

A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a packet is carrying data from A (client) to

B (server), it can also carry acknowledgment feedback about arrived packets from B (server); when a packet is carrying data from B (server) to A (client), it can also carry acknowledgment feedback about the arrived packets from A (client).

Figure 15.26 shows the layout for the Go-Back-N protocol implemented bi-directionally using piggybacking. The client and server each use two independent windows: send and receive windows.

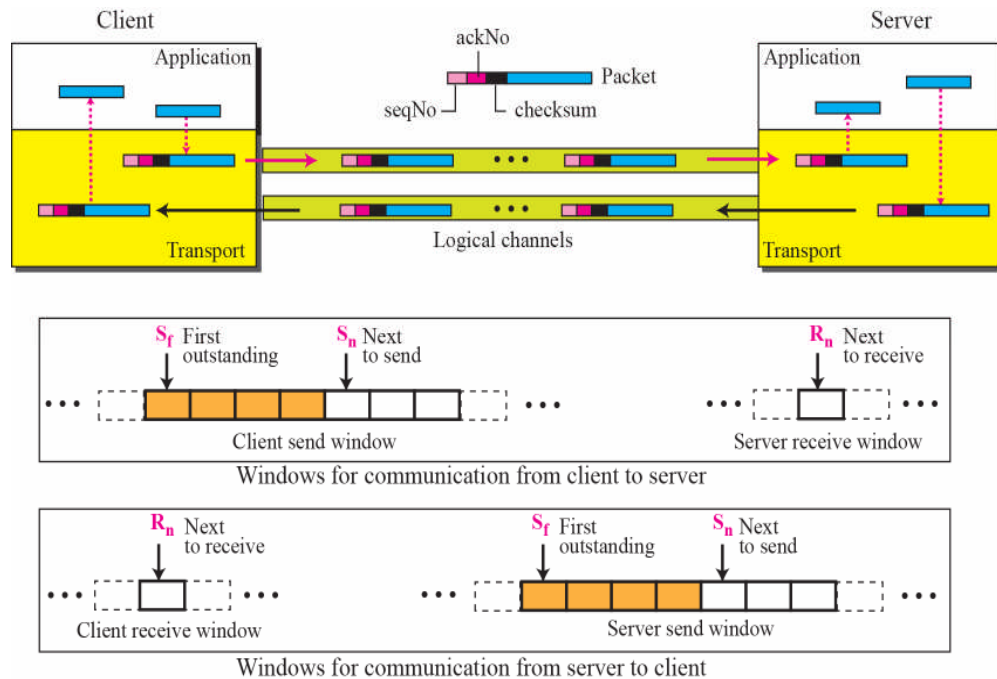


Figure 15.26: Design of Piggybacking for Go-Back-N protocol

15.4 User Datagram Protocol (UDP)

User Datagram Protocol (UDP) is a connectionless and unreliable transport protocol. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication. Also, it performs very limited error checking.

If UDP is so powerless, why would a process want to use it?

Most of the processes use UDP's service because UDP is a very simple protocol using a minimum of overhead. If a process wants to send a small message and does not care much about reliability, it can use UDP. Sending a small message by using UDP takes much less interaction between the sender and receiver than using TCP or SCTP.

User Datagram – format

UDP packets, called as user datagrams, have a fixed-size header of 8 bytes. Figure 15.27 shows the format of a user datagram.

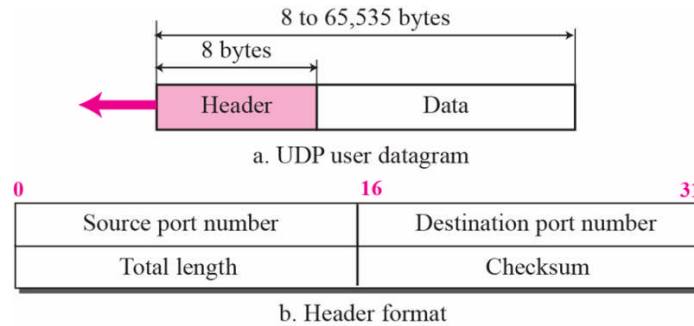


Figure 15.27: User Datagram - format

The fields are as follows:

- **Source port number:** It is the port number used by the process running on the source host. It is 16 bits long, which means that the port number can range from 0 to 65,535.
- **Destination port number:** It is the port number used by the process running on the destination host. It is also 16 bits long.
- **Total Length:** It is a 16-bit field that defines the total length of the user datagram, header plus data. The 16 bits can define a total length of 0 to 65,535 bytes. However, the total length needs to be much less because a UDP user datagram is stored in an IP datagram with the total length of 65,535 bytes.
- **Checksum:** It is used to detect errors over the entire user datagram (header plus data).

UDP Services

UDP provides following services which are already discussed as general services provided at the transport layer in the beginning of this chapter (point - 15.2).

Process-to-Process Communication

UDP provides process-to-process communication by using sockets (combination of IP address and Port number). Table 15.1 shows some well-known ports used with the UDP.

<i>Port</i>	<i>Protocol</i>	<i>Description</i>
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Domain	Domain Name Service (DNS)
67	Bootps	Server port to download bootstrap information
68	Bootpc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

Table 15.1: Well-known ports used with UDP

Connectionless Service

UDP provides a connectionless service, which means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagrams even if they are coming from the same source process. The user datagrams are not numbered and also, there is no connection establishment and no connection termination as is the case for TCP. This means that each user datagram can travel on a different path.

Flow Control

As UDP is a very simple protocol. There is no *flow control*, and hence no window mechanism. The receiver may overflow with incoming user datagram packets. The lack of flow control means that the process using UDP should provide for this service, if needed.

Error Control

There is no error control mechanism in UDP except for the checksum. This means that the sender does not know if user datagram packet has been lost or duplicated. When the receiver detects an error through the checksum, the user datagram is silently discarded. The lack of error control means that the process using UDP should provide for this service, if needed.

Congestion Control

Since UDP is a connectionless protocol, it does not provide congestion control. UDP assumes that the packets sent are small and sporadic, and cannot create congestion in the network. This assumption may or may not be true today when UDP is used for real time transfer of audio and video.

Encapsulation and De-capsulation

To send a message from one process to another, the UDP protocol does encapsulates (at sender process) and de-capsulates (at receiver process) messages (see Figure 15.28).

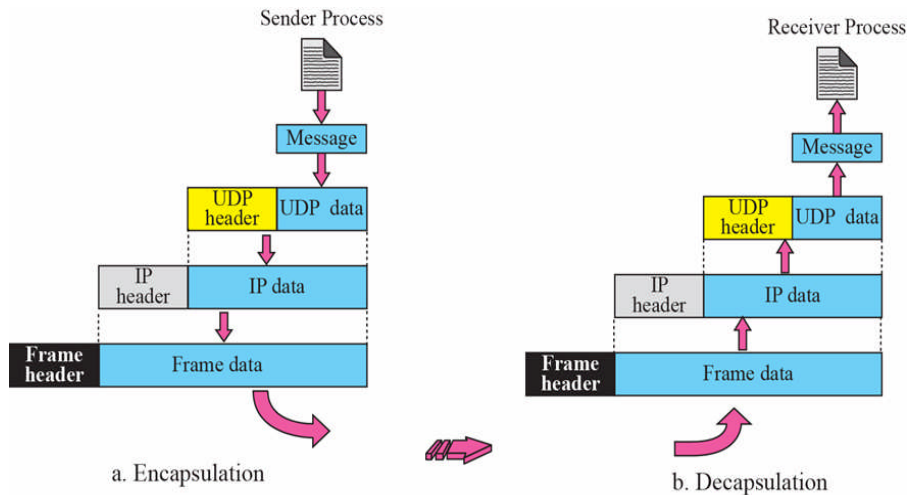


Figure 15.28: UDP – Encapsulation and Decapsulation

Multiplexing and De-multiplexing

In a host running a TCP/IP protocol suite, there is only one UDP but possibly several processes from application layer that may want to use the services of UDP. To handle this situation, UDP multiplexes and de-multiplexes (see Figure 15.29).

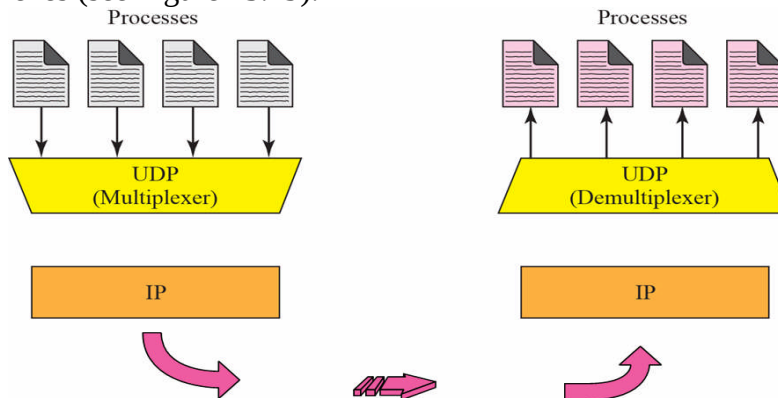


Figure 15.29: UDP – Multiplexing and Demultiplexing

UDP Applications

- UDP is suitable for a process with internal flow and error control mechanisms. For example, the Trivial File Transfer Protocol (TFTP) process includes flow and error control so it can easily use UDP.
- UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.

- UDP is used for management processes such as SNMP (Simple Network Management Protocol).
- UDP is used for some route updating protocols such as RIP (Routing Information Protocol).

15.5 TRANSMISSION CONTROL PROTOCOL (TCP)

The second transport layer protocol is the Transmission Control Protocol (TCP). TCP is a *connection-oriented* and *reliable* transport protocol. It adds connection-oriented and reliability features to the services of IP.

TCP Segment – format

The format of the TCP segment is shown in Figure 15.30. The segment consists of a header of 20 to 60 bytes, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.

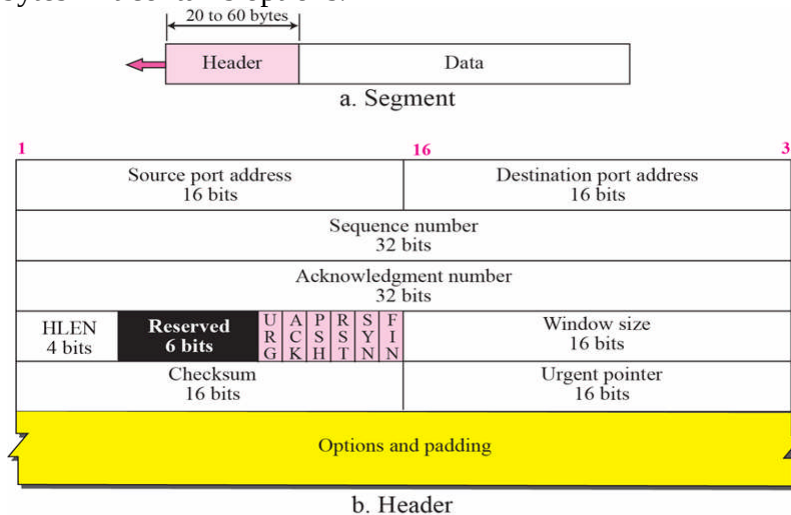


Figure 15.30: TCP Segment - format

Following are some of the fields present in the header of the TCP Segment.

- **Source port address:** It is a 16-bit field that defines the port number of the application program in the host that is sending the segment.
- **Destination port address:** It is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.
- **Sequence number:** This 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered.
- **Acknowledgment number:** This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte

number x from the other party, it returns $x+1$ as the acknowledgment number.

- **Header length:** This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field is always between 5 ($5 \times 4 = 20$) and 15 ($15 \times 4 = 60$).
- **Reserved:** This is a 6-bit field reserved for future use.
- **Control:** This field defines 6 different control bits or flags as shown in Figure. One or more of these bits can be set at a time. These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP.
- **Window size:** This field defines the window size of the sending TCP in bytes. Length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window and is determined by the receiver. The sender must obey the dictation of the receiver in this case.
- **Checksum:** This 16-bit field contains the checksum.
- **Urgent pointer:** This 16-bit field, which is valid, only if the urgent flag is set, is used when the segment contains urgent data. It defines a value that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment.
- **Options:** There can be up to 40 bytes of optional information in the TCP header.

TCP Services

Following are the services offered by the TCP to the process at the application layer

Process-to-process communication

TCP also provides process-to-process communication like UDP by using port numbers. List of well-known port numbers used with TCP as shown in the Table 15.2.

<i>Port</i>	<i>Protocol</i>	<i>Description</i>
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20 and 21	FTP	File Transfer Protocol (Data and Control)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol

Table 15.2: Well-known ports used by TCP

Stream Delivery Service

TCP is a stream-oriented protocol. TCP allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes. TCP creates an environment in which the two processes seem to be connected by an imaginary “tube” that carries their bytes across the Internet. This imaginary environment is shown in Figure 15.31.

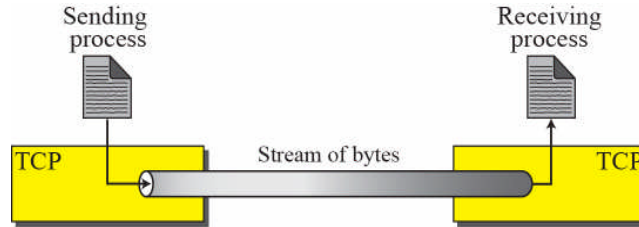


Figure 15.31: Stream delivery

Sending and Receiving Buffers

There are two buffers, the sending buffer and the receiving buffer, one for each direction. One way to implement a buffer is to use a circular array of 1-byte locations as shown in Figure 15.32.

For simplicity, we have shown two buffers of 20 bytes each; normally the buffers are hundreds or thousands of bytes, depending on the implementation. We also show the buffers as the same size, which is not always the case.

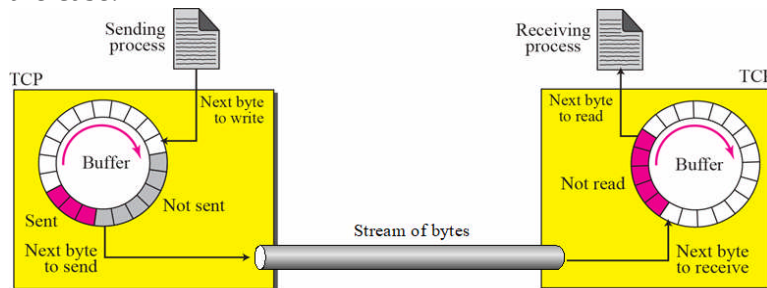


Figure 15.32: Sending and Receiving buffers

Full-Duplex Communication

TCP provides *full-duplex service*, where data can flow in both directions at the same time. Each TCP endpoint then has its own sending and receiving buffer, and segments move in both directions.

Multiplexing and De-multiplexing

TCP performs multiplexing at the sender and de-multiplexing at the receiver. However, since TCP is a connection-oriented protocol, a connection needs to be established for each pair of processes (see Figure 15.33).

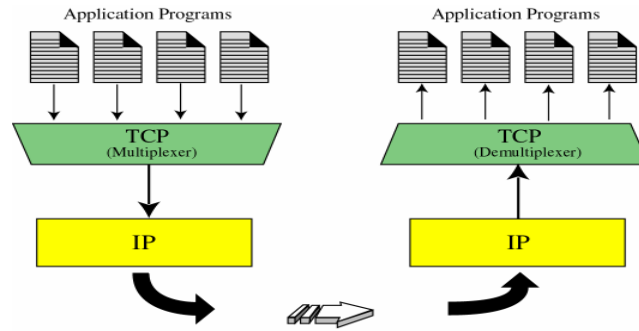


Figure 15.33: TCP – Multiplexing and De-multiplexing

Connection-Oriented Service

TCP is a connection-oriented protocol. As shown in Figure 15.12, when a process at site A wants to send to and receive data from another process at site B, the following three phases occur:

1. The two TCPs establish a virtual connection between them.
2. Data are exchanged in both directions.
3. The connection is terminated.

Note that this is a virtual connection, not a physical connection. The TCP segment is encapsulated in an IP datagram and can be sent out of order, or lost, or corrupted, and then resent. Each may be routed over a different path to reach the destination. There is no physical connection. TCP creates a stream-oriented environment in which it accepts the responsibility of delivering the bytes in order to the other site.

Encapsulation and De-capsulation

Encapsulation happens at the sender site. When a process has a message to send, it passes the message to the transport layer along with a pair of socket addresses and some other pieces of information that depends on the transport layer protocol. The transport layer receives the data and adds the transport-layer header. De-capsulation happens at the receiver site. When the message arrives at the destination transport layer, the header is dropped and the transport layer delivers the message to the process running at the application layer. The sender socket address is passed to the process in case it needs to respond to the message received (see Figure 15.34).

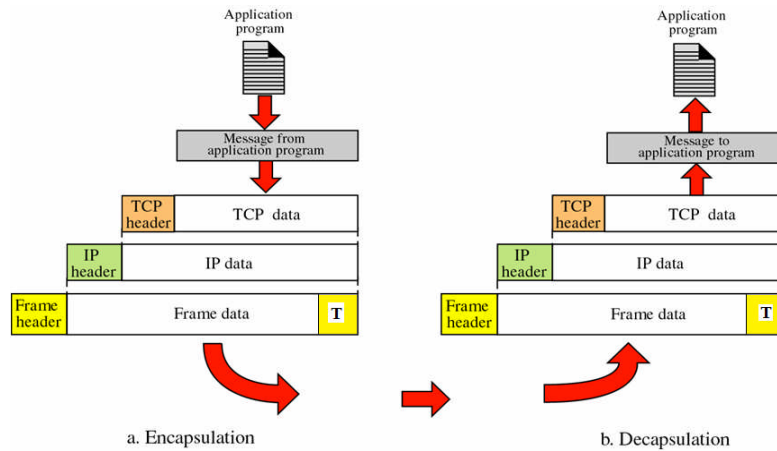


Figure 15.34: TCP – Encapsulation and De-capsulation

Reliable Service

TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data.

Flow Control

TCP provides flow control. The sending TCP controls how much data can be accepted from the sending process; the receiving TCP controls how much data can be sent by the sending TCP. This is done to prevent the receiver from being overwhelmed with data. The numbering system allows TCP to use a byte-oriented flow control.

Error Control

To provide reliable service, TCP implements an error control mechanism. Although error control considers a segment as the unit of data for error detection (loss or corrupted segments), error control is byte-oriented.

Congestion Control

TCP takes into account congestion in the network. The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion, if any, in the network.

15.6 SUMMARY

- We have discussed the main responsibilities or services of a transport-layer in this chapter such as Process-to-Process Communication, Addressing: Port Numbers, Encapsulation and De-capsulation, Multiplexing and De-multiplexing, Flow Control, Error Control, Congestion Control, Connectionless and Connection-Oriented.
- We have also discussed several common transport-layer protocols in this chapter. The simple connectionless protocol provides neither flow control nor error control. The connection-oriented Stop-and-Wait protocol provides both flow and error control, but is inefficient. The Go-back-*N* protocol is the more efficient version of the Stop-and-wait

protocol that takes advantage of pipelining. The Selective-Repeat protocol is a modification of the Go-back- N protocol that is better suited to handle packet loss. All of these protocols can be implemented bidirectionally using piggybacking.

- UDP is a connectionless, unreliable transport layer protocol with no embedded flow or error control mechanism except the checksum for error detection. The UDP packet is called a user datagram. A user datagram is encapsulated in the data field of an IP datagram.
- Transmission Control Protocol (TCP) is one of the transport layer protocols in the TCP/IP protocol suite. TCP provides process-to-process, full-duplex, and connection-oriented service.

15.7 REFERENCE FOR FURTHER READING

For more details about topics discussed in this chapter, we recommend the following books.

1. *Data Communication and Networking* by Behrouz A. Forouzan, McGraw-Hill, 2007.
2. *TCP/IP Protocol Suite* by Behrouz A. Forouzan, McGraw-Hill, 2010.

15.8 MODEL QUESTIONS

1. In cases where reliability is not of primary importance, UDP would make a good transport protocol. Give examples of specific cases.
2. Are both UDP and IP unreliable to the same degree? Why or why not?
3. Do port addresses need to be unique? Why or why not? Why are port addresses shorter than IP addresses?
4. What is the minimum size of a UDP datagram?
5. What is the maximum size of a UDP datagram?
6. What is the minimum size of the process data that can be encapsulated in a UDP datagram?
7. What is the maximum size of the process data that can be encapsulated in a UDP datagram?
8. Compare the TCP header and the UDP header. List the fields in the TCP header that are missing from UDP header. Give the reason for their absence.
9. UDP is a message-oriented protocol. TCP is a byte-oriented protocol. If an application needs to protect the boundaries of its message, which protocol should be used, UDP or TCP?
10. What is the maximum size of the TCP header? What is the minimum size of the TCP header?

Exercises

1. A sender sends a series of packets to the same destination using 5-bit sequence of numbers. If the sequence number starts with 0, what is the sequence number of the 100th packet?
2. Using 5-bit sequence numbers, what is the maximum size of the send and receive windows for each of the following protocols?
 - a. Stop-and-Wait
 - b. Go-Back-*N*
 - c. Selective-Repeat
3. A client has a packet of 68,000 bytes. Show how this packet can be transferred by using only one UDP user datagram.
4. A client uses UDP to send data to a server. The data are 16 bytes. Calculate the efficiency of this transmission at the UDP level (ratio of useful bytes to total bytes).
5. In TCP, if the value of HLEN is 0111, how many bytes of option are included in the segment?



STANDARD CLIENT – SERVER PROTOCOLS

Unit Structure

- 16.0 Objectives
- 16.1 Introduction
- 16.2 World Wide Web and HTTP
- 16.3 FTP
- 16.4 Electronic Mail
- 16.5 Telnet
- 16.6 Secure Shell
- 16.7 Domain Name System.
- 16.8 Summary
- 16.9 Reference for further reading
- 16.10 Model Questions

16.0 OBJECTIVES:

This chapter would make you to understand the following concepts:

- Functionality of an Application layer
- Client – Server architecture
- Client – Server standard protocols
- WWW and HTTP protocol
- FTP and its mechanism for copying file from one host to another
- E-mail protocols such SMTP, POP3, IMAP4 and MIME
- Telnet and Secured shell
- Domain Name System

16.1 INTRODUCTION

An application layer allows users to access the services provided by the network or internet. It provides actual interface to the users and support services such as file access and transfer management, remote login, electronic mail, surfing an internet, network management, directory access service, etc.

For accessing the services provided by the network or internet we need a pair of program, one is running on the local computer and other one is running on remote computer. The program running on the local computer is called as *Client* program whereas the program running on the remote computer is called as *Server* program. Client program is the consumer of the services provided by Server program. Client program always sends the request for the service to Server program whereas the server program provides service to the client program in the form of response.

This type of communication over network or internet is called as *Client – Server architecture*. In this chapter, we briefly discuss some application programs that are designed based on client – server architecture and running in the internet or network.

16.2 WORLD WIDE WEB AND HTTP

It is *repository of information* linked together from various points all over the world is called as World Wide Web (WWW). The WWW project was initiated by CERN (European Laboratory for Particle Physics) to create a system to handle distributed resources necessary for research work. WWW today is a distributed client - server service architecture, in which a client using a browser can access a service from a server (see Figure 16.1).

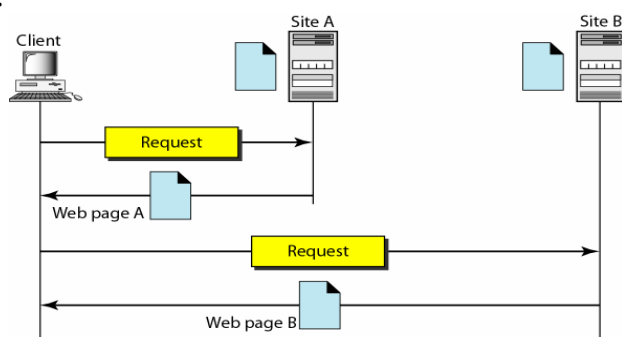


Figure 16.1: WWW architecture

WWW consist of components such as Client browser, Server, URL, cookies and web documents.

Client (Browser): A variety of commercial browsers that interprets and displays a Web document received from the server. Each browser consists of three parts - a controller, client protocol, and interpreter (see Figure 16.2). Most popular web browsers are Google Chrome, Microsoft Edge (formerly Internet Explorer), Mozilla Firefox, and Apple's Safari.

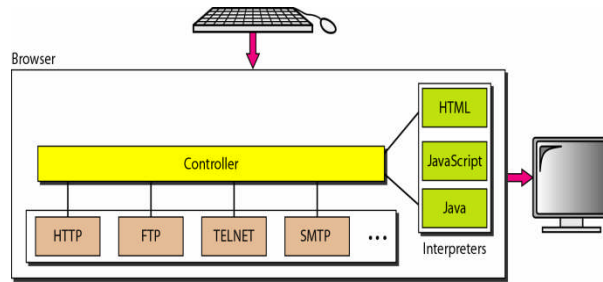


Figure 16.2: Client (browser)

Server: The Web pages are stored at the server. Each time when a client request arrives, the corresponding web document is sent to the client.

Uniform Resource Locator (URL): When a client wants to access a web page, client needs the address of the web page. To facilitate the access of documents distributed throughout the world, HTTP uses locator called as Uniform Resource Locator (URL). It is a standard for specifying any kind of information on the internet. URL defines things such as protocol, host computer; port number and path (see Figure 16.3).



Figure 16.3: URL

Cookies: a string of characters that holds some information about the client and must be returned to the server and vice versa.

Web Documents: web documents in the WWW can be grouped into three broad categories: static (HTML), dynamic (DHTML), and active (Java applets), this categorizing is based on the time at which the contents of the document are determined.

Usually the static web pages are created by using Hyper Text Markup Language (HTML) whereas dynamic and active web pages are created by using Dynamic Hyper Text Markup Language (DHTML) and Java Applets respectively.

The Hypertext Transfer Protocol (HTTP)

Mainly HTTP is used to access the data on the World Wide Web. HTTP functions as a combination of FTP and SMTP.

It is like File Transfer Protocol (FTP) because it transfers files and uses the TCP's service. However, it is much simpler than FTP because it uses only one TCP connection. There is no separate control connection; only data are transferred between the client and the server.

HTTP is similar to SMTP because the data transferred between the client and the server look like SMTP messages. Also, the format of the messages is controlled by Multipurpose Internet Mail Extensions (MIME) -like headers.

Unlike SMTP, the 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 stored and forwarded, but HTTP messages are delivered immediately.

In HTTP, the commands from the client to the server are embedded in a *request message*. The contents of the requested file or other information are embedded in a *response message*. HTTP uses the services of TCP on well-known port 80.

HTTP Transaction

Figure 16.4 shows the HTTP transaction between the client and server. Though HTTP uses the services of TCP, HTTP itself is a stateless protocol. The client initializes the transaction by sending a request message to the server. The server replies by sending a response to the client.

Messages: formats of the request and response messages are similar; both are shown in Figure 16.5. A request message consists of a *request line*, a *header*, and sometimes a *body* whereas a response message consists of a *status line*, a *header*, and sometimes a *body*.

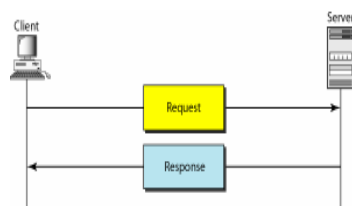


Figure 16.4: HTTP Transaction

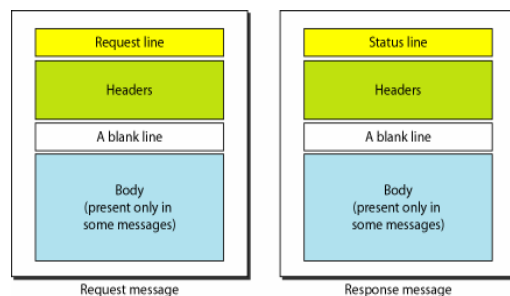


Figure 16.5: Request and response Message

The first line in a request message is called as a request line; the first line in the response message is called as the status line. There is one common field in both is HTTP version, as shown in Figure 16.6.

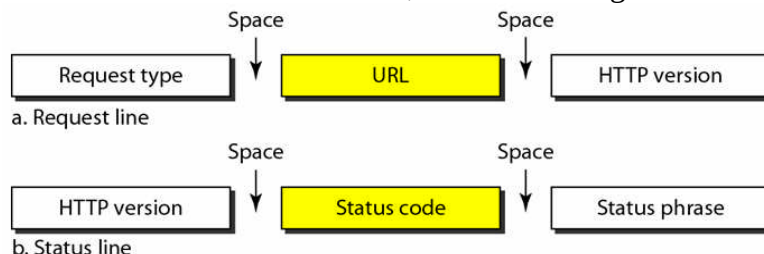


Figure 16.6: Request and Status line

Fields of Request Line and Status Line: There is one common field in both is HTTP version

- Request type: used in request message and are categorized into methods as shown in Table 16.1
- URL: Uniform Resource Locator.
- HTTP version: current version of HTTP is 1.1.
- Status code: used in response message and consists of 3 digits – codes in 100 range are informational, codes in 200 range indicate successful request, codes in 300 range redirect the client to another URL, codes in 400 range indicate an error at client and codes in 500 range indicate error at server site. Most common codes are shown in Table 16.2.
- Status phrase: used in the response message. It explains the status code in text form. Table 16.2 also provides the status phrase for each code.

<i>Method</i>	<i>Action</i>
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires about available options

Table 16.1: Request methods

<i>Code</i>	<i>Phrase</i>	<i>Description</i>
Informational		
100	Continue	The initial part of the request has been received, and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
Success		
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon.
204	No content	There is no content in the body.
<i>Code</i>	<i>Phrase</i>	<i>Description</i>
Redirection		
301	Moved permanently	The requested URL is no longer used by the server.
302	Moved temporarily	The requested URL has moved temporarily.
304	Not modified	The document has not been modified.
Client Error		
400	Bad request	There is a syntax error in the request.
401	Unauthorized	The request lacks proper authorization.
403	Forbidden	Service is denied.
404	Not found	The document is not found.
405	Method not allowed	The method is not supported in this URL.
406	Not acceptable	The format requested is not acceptable.
Server Error		
500	Internal server error	There is an error, such as a crash, at the server site.
501	Not implemented	The action requested cannot be performed.
503	Service unavailable	The service is temporarily unavailable, but may be requested in the future.

Table 16.2: Status codes and Phrases

Header: used to exchange additional information between the client and server. It consists of one or more header lines, belongs to one of four types – general header, request header, response header and entity header.

A request message consists of general header, request header and entity header whereas a response message consists of general header, response header and entity header.

- General Header: provides the general information about the message and present in both request and response message. (see Table 16.3)

- Request Header: provides client's configuration and client's preferred document format; present only in request message. (see Table 16.4)
- Response Header: provides server's configuration; present only in response message. (see Table 16.5)
- Entity Header: provides information about the body of the document; usually present in response message but sometimes present in request message (in which PUT or POST method used). (see Table 16.6)

Body: present in both request and response message and contains the document to be send or received.

<i>Header</i>	<i>Description</i>
Cache-control	Specifies information about caching
Connection	Shows whether the connection should be closed or not
Date	Shows the current date
MIME-version	Shows the MIME version used
Upgrade	Specifies the preferred communication protocol

Table 16.3: General headers

<i>Header</i>	<i>Description</i>
Accept	Shows the medium format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
From	Shows the e-mail address of the user
Host	Shows the host and port number of the server
If-modified-since	Sends the document if newer than specified date
If-match	Sends the document only if it matches given tag
If-non-match	Sends the document only if it does not match given tag
If-range	Sends only the portion of the document that is missing
If-unmodified-since	Sends the document if not changed since specified date
Referrer	Specifies the URL of the linked document
User-agent	Identifies the client program

Table 16.4: Request headers

<i>Header</i>	<i>Description</i>
Accept-range	Shows if server accepts the range requested by client
Age	Shows the age of the document
Public	Shows the supported list of methods
Retry-after	Specifies the date after which the server is available
Server	Shows the server name and version number

Table 16.5: Response headers

Header	Description
Allow	Lists valid methods that can be used with a URL
Content-encoding	Specifies the encoding scheme
Content-language	Specifies the language
Content-length	Shows the length of the document
Content-range	Specifies the range of the document
Content-type	Specifies the medium type
Etag	Gives an entity tag
Expires	Gives the date and time when contents may change
Last-modified	Gives the date and time of the last change
Location	Specifies the location of the created or moved document

Table 16.6: Entity headers

Example 16.1:

This example retrieves a document. We use the GET method to retrieve an image with the path /usr/bin/image1. The request line shows the method (GET), the URL, and the HTTP version (1.1). The header has two lines that show that the client can accept images in the GIF or JPEG format. The request does not have a body. The response message contains the status line and four lines of header. The header lines define the date, server, MIME version, and length of the document. The body of the document follows the header (see Figure 16.7).

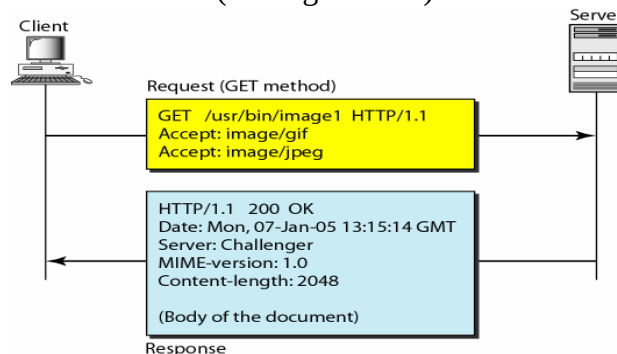


Figure 16.7: Example 16.1

16.3 FTP

It is standard mechanism provided by TCP/IP for copying a file from one host to another.

Problems during File transfer:

- Two systems may use different file name conventions.
 - Two systems may have different ways to represent text and data.
 - Two systems may have different directory structures.
- ☺ All these problems have been solved by FTP in very simple way.

FTP is different than other client-server programs. It establishes two connections between hosts. One is used for data transfer and other is

used for control information (exchange of Commands and Responses). FTP uses the services of TCP, it needs two connections: well-known *port 21* is used for the Control connection and *port 20* is used for the Data connection.

Control connection remains connected during the entire interactive FTP session whereas the Data connection is opened and then closed for each file transferred. (See Figure 16.8)

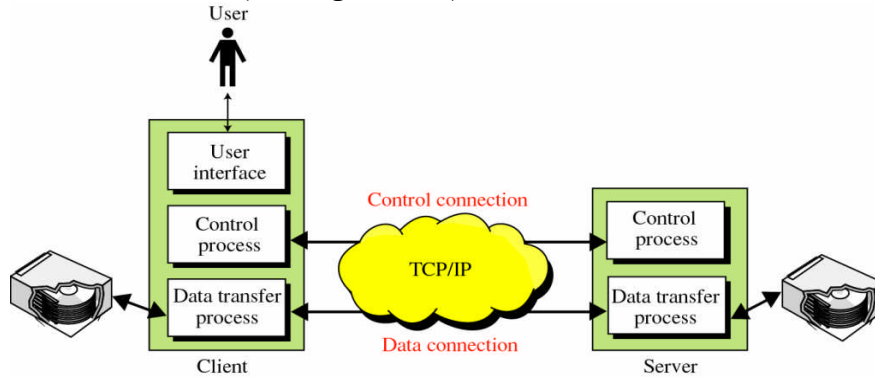


Figure 16.8: FTP

Communication over Control Connection: FTP uses same approach like SMTP to communicate across the control connection. It uses for commands and responses the 7-bit ASCII (NVT ASCII) character set. Each Command or Response is only one short line. Each line is terminated with a two characters (<CRLF> carriage return and line feed) end-of-line token. (See Figure 16.9)

Communication over Data Connection: File transfer occurs over the data connection under the control of the *commands* sent over the **control connection**. A file is to be copied from the server to the client under the supervision of *RETR* command. A file is to be copied from the client to the server under the supervision of *STOR* command. A list of directory or file names is to be sent from the server to the client under the supervision of *LIST* command. (See Figure 16.10)

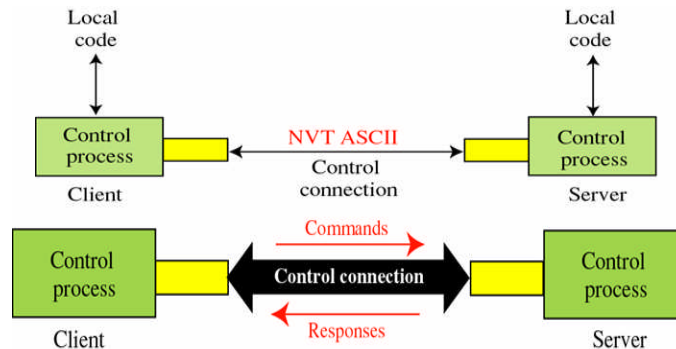


Figure 16.9: FTP – Control connection



Figure 16.10: FTP – Data connection

Before sending a file through the data connection; the client must define the *type of file* to be transferred, *the structure of the data* and *the transmission mode*.

File Type: FTP can transfer one of the following types across the data connection.

- **ASCII FILE** is a default format (7bit ASCII encoding)
- **EBCDIC FILE** is used by IBM (EBCDIC encoding)
- **IMAGE FILE** is the default format for transferring binary file, it is sent as continuous streams of bits without any encoding.

Data Structure: it uses one of the following interpretations about the structure of the data.

- **File structure format:** is a continuous stream of bytes.
- **A record structure:** file is divided into records (text files).
- **Page structure:** file divided into pages, each page consist of page number and page header. Pages can be accessed randomly or sequentially.

Transmission Mode: it uses one of the three transmission modes

- **Stream mode:** default mode, data delivered from FTP to TCP, as continuous streams of bytes (segments of appropriate size).
- **Block mode:** data can be delivered from FTP to TCP in blocks; each block is preceded by a 3 byte header. First byte called block descriptor next 2 bytes defines the size of the block in bytes.
- **Compressed mode:** compression method normally used is *run-length* encoding in which consecutive appearances of data unit are replaced by one occurrence and the number of repetitions (text file spaces-blank).
- **Anonymous FTP:** Some sites have set of files available for public access, to enable anonymous FTP, user does not need to have an account and password to access files, instead, the user can use *anonymous* as the user name and *guest* as the password.

16.4 ELECTRONIC MAIL

An *Electronic - mail* is one of the most popular Internet services. Internet Designers probably never imagined the popularity of this application program. At the beginning of the internet era, the messages sent by email were *short* and consisted of *text* only. Today email is much more complex, it allows a message to include text, audio, and video. It also allows one message to be sent to one or more recipients.

Email Architecture

When both sender and receiver are connected to their mail servers via a LAN or a WAN, we need two UAs and two pair of MTAs (client and server), and pair of MAAs (client and server). This is the most common email architecture used today. (See Figure 16.11)

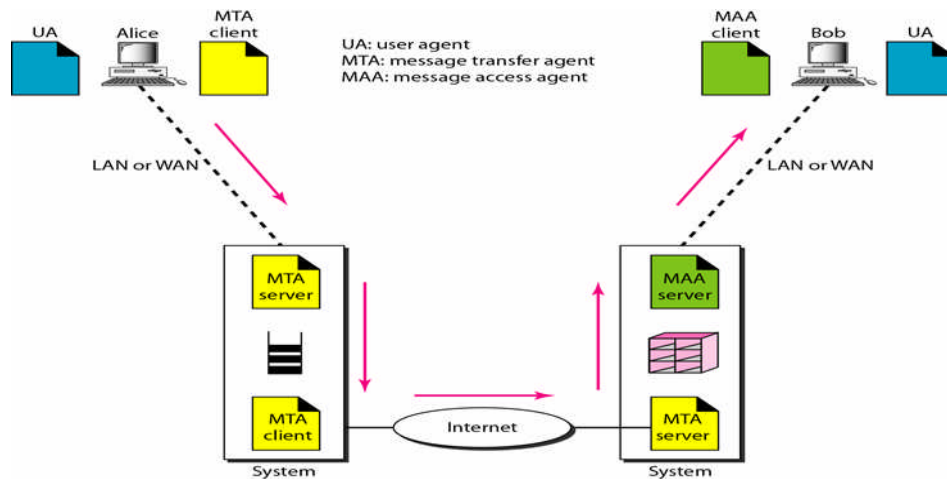


Figure 16.11: Email architecture

- **User Agent (UA):** it provides service to the user to make the process of sending and receiving a message easier.
- **Message Transfer Agents (MTA):** a client-server program used to transfer the message across the internet.
- **Message Access Agent (MAA):** a client-server program that pulls the stored email messages.

User Agent (UA)

First component of email system is a user agent (UA), there are two types of UAs; namely Command driven UA and GUI based UA. Some command driven UA examples are *mail*(Linux), *eml*(UNIX), etc. Graphical User Interface UAs are more sophisticated and easier to use, some GUI based UA examples are *Outlook Express* (Microsoft), *Eudora mail* (open source), *Netscape mail* (Netscape), etc.

Services provided by UAs are composing messages, reading messages, replying messages, forwarding messages and handling mailboxes

Sending Mail: for sending mail, the user, through the UA, creates mail that looks very similar to postal mail. It has an envelope (sender and receiver address) and a message; where message contains header (defines sender, receiver, and subject of the message) and body (actual information to be read by recipient).

Receiving Mail: When a user receives mail, UA informs to the user with a notice and if the user is ready to read the mail. A list is displayed in which each line contains a summary of the information about a particular message in the mailbox.

Email Address: In the Internet, an email address consists of two parts: a local part and a domain name, separated by @ sign (see Figure 16.12).

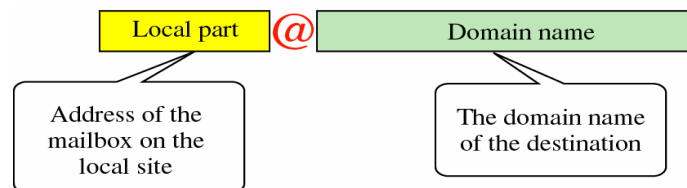


Figure 16.12: Email address

MIME (Multipurpose Internet Mail / Message Extensions)

As we know an electronic mail has a simple structure and it can send messages only in NVT 7-bit ASCII format; hence it cannot be used for languages that are not supported by 7-bit ASCII characters (such as French, German, Hebrew, Russian, Chinese, and Japanese). Also, it cannot be used to send binary files or video or audio data. Solution to this problem is Multipurpose Internet Mail Extensions (MIME), which is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client MTA to be sent through the Internet. The message at the receiving site is transformed back to the original data. MIME as a set of software functions that transforms non-ASCII data (stream of bits) to ASCII data and vice versa, as shown in Figure 16.13.

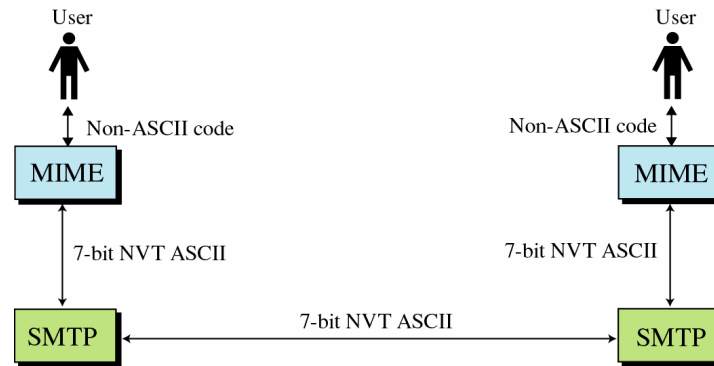


Figure 16.13: MIME

Message Transfer Agent: SMTP (Simple Mail Transfer Protocol)

Actual mail transfer is done through MTA (message transfer agents). To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The protocol that defines the MTA client and server in the Internet is called as Simple Mail Transfer Protocol (SMTP). Figure 16.14 shows the actual place of SMTP in today's email system.

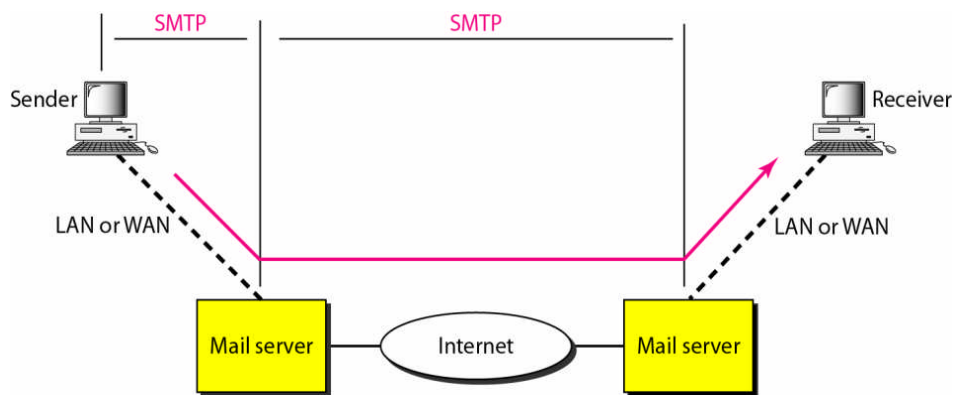


Figure 16.14: MTA – SMTP

SMTP uses Commands and Responses to transfer the mail messages between SMTP client and SMTP server. Every command or response is terminated by a two-character (carriage return and line feed) which is end-of-line token.

SMTP Commands: commands are sent from the client to the server. SMTP defines 14 different commands. Out of these, first five are mandatory; every implementation must support these five commands. Next three are often used and highly recommended. Last six are seldom used. (See Table 16.7)

Keyword	Argument (s)
HELO	Sender's host name.
MAIL FROM	Sender of the message.
RCPT TO	Intended recipient of the message.
DATA	Body of the mail.
QUIT	Specifies that the receiver must send an OK reply, and then close the transmission channel.
RSET	Specifies that the current mail transaction is to be aborted & receiver must send an OK reply.
VRFY	Name of the recipient to be verified.
NOOP	It specifies no action other than that the receiver send an OK reply.
TURN	The receiver must either (1) send an OK reply and then take on the role of the sender-SMTP, or (2) send a refusal reply and retain the role of the receiver-SMTP.
EXPN	Asks the receiver to confirm that the argument identifies a mailing list.
HELP	The receiver to send helpful information to the sender of the HELP command.
SEND FROM	Initiate a mail transaction in which the mail data is delivered to one or more terminals.
SMOL FROM	Initiate a mail transaction in which the mail data is delivered to one or more terminals or mailboxes.
SMAL FROM	Initiate a mail transaction in which the mail data is delivered to one or more terminals and mailboxes.

Table 16.7: SMTP Commands

Code	Description
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input
Transient Negative Completion Reply	
421	Service not available
450	Mailbox not available
451	Command aborted; local error
452	Command aborted; insufficient storage
Permanent Negative Completion Reply	
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

Table 16.8: SMTP Response Codes

Responses: Responses are sent from the SMTP server to the SMTP client. A response is a 3 digit code that may be followed by additional textual information. Table 16.8 shows some of the response codes.

Mail Transfer: Transfer of a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

Message Access Agent: POP3 and IMAP4

In current scenario of the mail system, first and second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because SMTP is a *push* protocol; it pushes the message from the client to the server. Whereas, the third stage needs a *pull* protocol; the client must pull messages from the server. This third stage uses a Message Access Agent.

Presently two MAA protocols are available: POP3 (Post Office Protocol - version 3) and IMAP4 (Internet Mail Access Protocol - version 4). Figure 16.15 shows the place of these two protocols in today's email system.

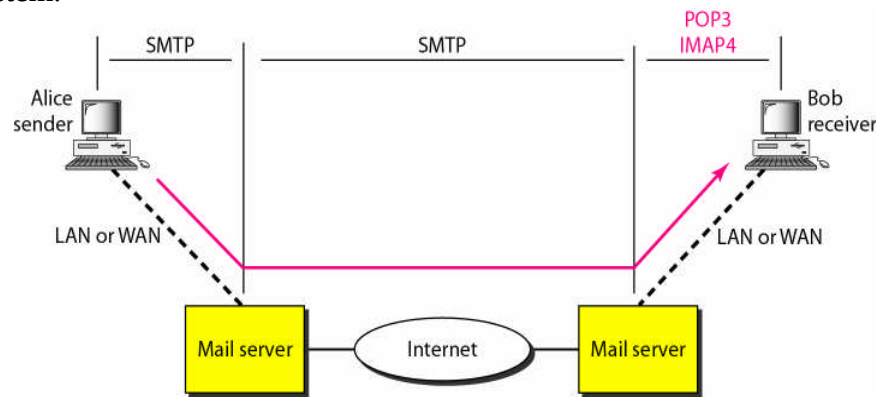


Figure 16.15: POP3 and IMAP4

Post Office Protocol (POP3)

It is a simple and limited in functionality protocol. POP3 client software is installed on the recipient computer; the POP3 server software is installed on the recipient's mail server. Mail access starts with the client when the user needs to download e-mail from the mailbox on the mail server. The client opens a connection with the server on TCP port 110 and then sends its user name and password to access the mailbox. The user can then list and retrieve the mail messages, one by one. Figure 16.16 shows how mails are downloaded using POP3.

POP3 has two modes: the delete mode and the keep mode. In the delete mode, after each retrieval, the mail is deleted from the mailbox. In the keep mode, the mail remains in the mailbox after retrieval. The delete mode is normally used when the user is working at his permanent computer and can save and organize the received mails after reading or replying. The keep mode is normally used when the user accesses his mail away from her primary computer. The mail is read but kept in the system for later retrieval and organizing.

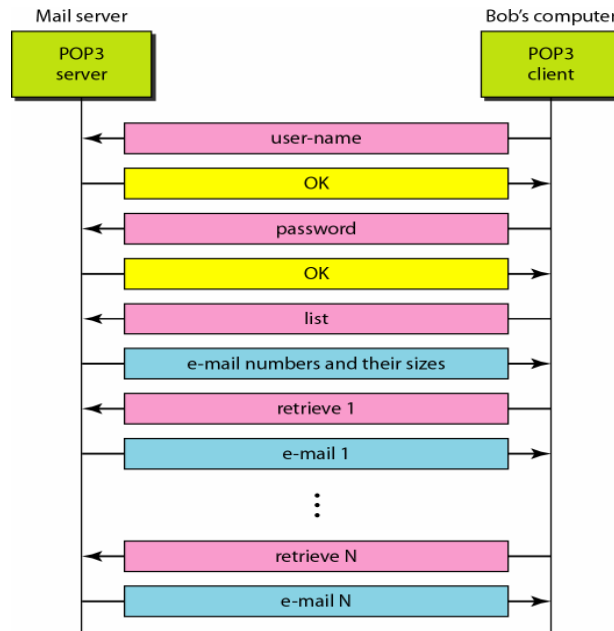


Figure 16.16: POP3 exchange of Commands and Responses

Internet Mail Access Protocol (IMAP4)

Other MAA protocol is IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex. POP3 is deficient in several ways. It does not allow the user to organize his mail on the server; the user cannot have different folders on the server. In addition, POP3 does not allow the user to partially check the contents of the mail before downloading. IMAP4 uses TCP's port number 143.

IMAP4 provides the following some extra functions:

- A user can check the e-mail header prior to downloading.
- A user can search the contents of the e-mail for a specific string of characters prior to downloading.
- A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
- A user can create, delete, or rename mailboxes on the mail server.
- A user can create a hierarchy of mailboxes in a folder for e-mail storage.

16.5 TELNET

Users may want to run different application programs at a remote site and produce results that can be transferred to their local site. For example, to access different application programs required to do their homework assignments and project work; students may want to connect to their university or college intranet server from their home. The best solution for this problem is to use a general purpose client-server program

that allows a user to access any application program running on a remote computer. Means this program allows user to log on to a remote computer; after logging on, a user is allowed to use different services available on the remote computer; produce a result and transfer it back to their local computer.

That client-server application program is called as *TELNET*. The *TELNET* is an abbreviation for *Terminal Network*. It is the standard TCP/IP protocol for virtual terminal service as proposed by the International Organization for Standards (ISO).

TELNET enables the establishment of a connection with a remote computer in such a way that the local terminal appears to be a terminal at the remote computer.

Timesharing Environment

TELNET was actually designed to provide a timesharing environment for operating system, such as UNIX, where the interaction between a user and the computer occurs through a terminal, which is usually a combination of keyboard, monitor, and mouse.

Remote Login

When a user wants to access an application, program located on a remote machine. Both, the TELNET Client and Server programs are used. The user sends a keystroke to a terminal driver, where the local OS accepts the characters but does not interpret them. The characters are sent to TELNET Client, which converts the characters to a universal character set called Network Virtual Terminal (NVT) characters and delivers them to the local TCP/IP stack.

The text in NVT form, travels through the internet and arrives at the TCP/IP stack at the remote machine. Here the characters are delivered to the OS and passed to the TELNET server, which changes characters to the corresponding characters understandable by the remote computer.

However, remote OS is designed to receive characters only from a terminal driver and not from a TELNET server. Hence, the OS uses a pseudo-terminal driver to receive the characters, which in turn emulates the characters coming from a terminal. The OS then passes the received characters to the appropriate application program. (See Figure 16.17)

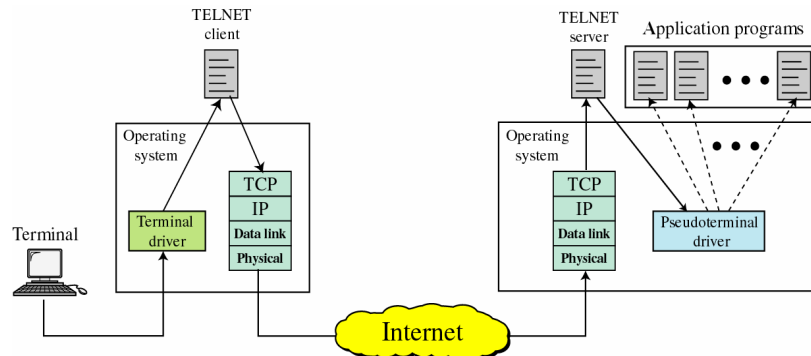


Figure 16.17: TELNET – Remote Login

Network Virtual Terminal (NVT)

Just to deal with heterogeneous systems and want to access any remote computer in the world. TELNET defines universal interface called as NVT character set, through this interface, the TELNET client translates characters (Data/Commands) that come from local terminal into NVT form and delivers them to the network. On the remote computer The TELNET server translates Data and Commands from NVT form into the form acceptable by the remote computer (see Figure 16.18).

NVT uses 2 character sets one for Data and other for Control, both are 8-bit. For Data characters - NVT uses 8-bit character set (7 out of which are same as ASCII) and highest order bit is 0. For Control characters - NVT uses 8-bit character set (7 out of which are same as ASCII) where the highest order bit is set to 1. Table 16.9 shows some Control characters.

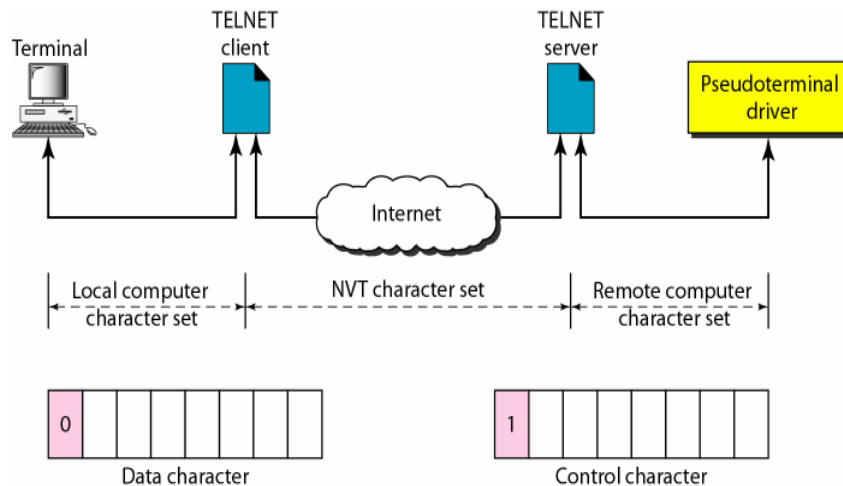


Figure 16.18: NVT concept

<i>Character</i>	<i>Decimal</i>	<i>Binary</i>	<i>Meaning</i>
EOF	236	11101100	End of file
EOR	239	11101111	End of record
SE	240	11110000	Suboption end
NOP	241	11110001	No operation
DM	242	11110010	Data mark
BRK	243	11110011	Break
IP	244	11110100	Interrupt process
AO	245	11110101	Abort output
AYT	246	11110110	Are you there?
EC	247	11110111	Erase character
EL	248	11111000	Erase line
GA	249	11111001	Go ahead
SB	250	11111010	Suboption begin
WILL	251	11111011	Agreement to enable option
WONT	252	11111100	Refusal to enable option
DO	253	11111101	Approval to option request
DONT	254	11111110	Denial of option request
IAC	255	11111111	Interpret (the next character) as control

Table 16.9: NVT characters

Embedding

TELNET uses only one TCP/IP connection, the server uses TCP's well known port 23 and client uses an ephemeral port (short lived). The same connection is used for sending both data and control characters. TELNET embeds control characters in data stream and distinguishes data from control characters, by a special control character called Interpret As Control (IAC).

For example: User wants Server to display a file "file1" on remote server, user can type.

```
cat file1
```

Suppose the filename has been mistyped as "file a" instead of "file1", then the user uses the "Back Space (←)" key to correct this situation.

```
cat file a<backspace>1
```

The Backspace character is translated into two remote characters (IAC, EC) which are embedded into the data and sent to the remote server (see Figure 16.19).

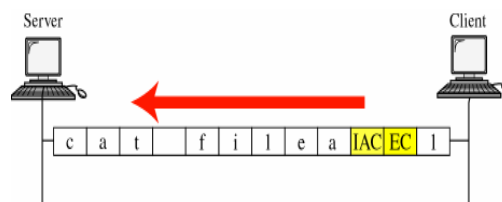


Figure 16.19: Example of Embedding

Options Negotiation

TELNET allows the client and server to negotiate the options before or during the use of service.

Options are extra features available to a user with more sophisticated terminal. Users with simpler terminals can use simpler features. Following Table 16.10 shows some common options.

<i>Code</i>	<i>Option</i>	<i>Meaning</i>
0	Binary	Interpret as 8-bit binary transmission.
1	Echo	Echo the data received on one side to the other.
3	Suppress go ahead	Suppress go-ahead signals after data.
5	Status	Request the status of TELNET.
6	Timing mark	Define the timing marks.
24	Terminal type	Set the terminal type.
32	Terminal speed	Set the terminal speed.
34	Line mode	Change to line mode.

Table 16.10: Options

To use any of the options mentioned in the above table, first requires option negotiation between client and server. Four control characters – WILL, WONT, DO and DONT are needed for this purpose. These characters are shown in Table 16.9.

For example – suppose the client wants the server to ECHO each character sent to the server. The request consists of 3 characters: *IAC*, *DO* and *ECHO*. The server informs the client by sending 3 character approval: *IAC*, *WILL* and *ECHO*. (See Figure 16.20)

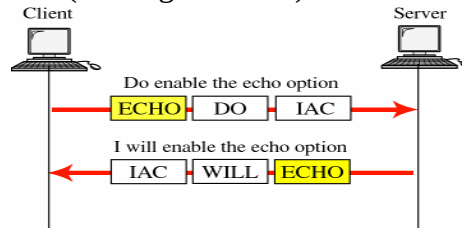


Figure 16.20: Echo option negotiation

Mode of Operations

Most TELNET implementations operate in one of three modes: The Default mode, Character Mode or Line mode.

Default Mode: used if no other modes are invoked through option negotiation. In this, the echoing is done by the client, user types a character and client echoes the character on the screen but does not send it until a whole line is completed.

Character Mode: In this, each character typed is sent by client to the server. The server normally echoes the character back to be displayed on client screen, also creates overhead for the network.

Line Mode: A new mode called Line mode, Line editing (echoing, character erasing, line erasing and so on) is done by the client. The client then sends the whole line to the server.

16.6 SECURE SHELL (SSH)

Another most popular application program used for remote login is Secure Shell (SSH). SSH, like TELNET, uses TCP's service, but SSH is more secure and provides more services than TELNET.

Versions: There are two versions of SSH - SSH-1 and SSH-2, which are totally incompatible. The first version, SSH-1 is has some of security flaws in it. So here we discuss only SSH-2.

Components

SSH is an application-layer protocol with four components, as shown in Figure 16.21.

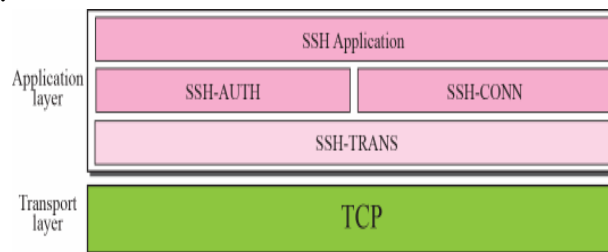


Figure 16.21: SSH – components

SSH Transport-Layer Protocol (SSH-TRANS): SSH first uses a protocol that creates a secured channel on the top of TCP. This new layer is an independent protocol called as SSH-TRANS. When the software implementing this protocol is called, the client and server first use the TCP protocol to establish an insecure pro-connection. Then they exchange several security parameters to establish a secure channel on the top of the TCP. Services provided by SSH-TRANS protocol are message confidentiality, data integrity, authenticity and compression.

SSH Authentication Protocol (SSH-AUTH): Once a secure channel is established between the client and the server and the server is authenticated for the client, SSH can call another software that can authenticate the client for the server referred as SSH – AUTH protocol.

SSH Connection Protocol (SSH-CONN): Once the secured channel is established and both server and client are authenticated for each other, SSH can call a piece of software that implements the third protocol, SSHCONN. One of the services provided by the SSH-CONN protocol is to do multiplexing. SSH-CONN takes the secure channel established by the two previous protocols and lets the client create multiple logical channels over it.

SSH Applications: After the connection phase is completed, SSH allows several application programs to use the connection. Each application can create a logical channel as described above and then benefit from the secured connection.

In other words, remote login is one of the services that can use the SSH-CONN protocols; other applications, such as a file transfer application can use one of the logical channels for this purpose.

Port Forwarding

Other interesting service provided by the SSH protocol is to provide *port forwarding*. We can use the secured channels available in SSH to access an application programs (such as TELNET and SMTP) that does not provide security services. SSH port forwarding mechanism creates a tunnel through which the messages belonging to other protocol can travel. For this reason, this mechanism is sometimes referred to as **SSH tunneling**. Figure 16.22 shows the concept of port forwarding.

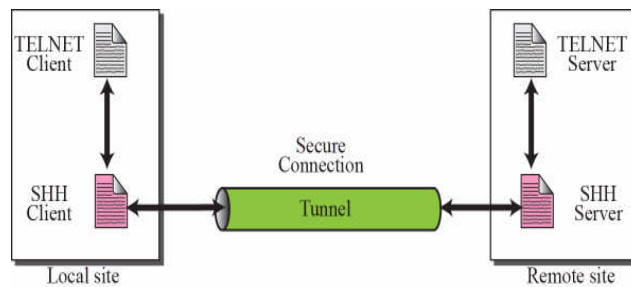


Figure 16.22: Port forwarding

We can change a direct, but insecure, connection between the TELNET client and the TELNET server by port forwarding. The TELNET client can use the SSH client on the local site to make a secure connection with the SSH server on the remote site. Any request from the TELNET client to the TELNET server is carried through the tunnel provided by the SSH client and server. Any response from the TELNET server to the TELNET client is also carried through the tunnel provided by the SSH client and server.

Domain Name System (DNS)

Every host connected to an internet has a unique IP address. IP address of that host is used by other computers to find and connect to that host. But people prefer usually host names instead of IP address of the host. Therefore we need a mechanism or system that can translate or map the host name to IP address or IP address to host name. In the internet such mechanism or system is provided by one of the application layer protocol called as Domain Name System (DNS).

Now we discuss how actually DNS works to map host name to IP address. In Figure 16.23, a user wants to use a FTP client to access the corresponding FTP server running on a remote host. The user knows only

the FTP server name, such as ftp.gnu.org. However, the TCP/IP suite present on user's FTP client needs the IP address of the FTP server to make the connection. For mapping the FTP server's name to IP address following are steps.

1. The user passes the FTP server name to the FTP client.
2. FTP client passes the FTP server to the DNS client.
3. We know that each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the FTP server name using the known IP address of the DNS server.
4. Once query message received by the DNS server, responds to the DNS client with the response message having DNS record (IP address) of the desired FTP server.
5. The DNS client passes the IP address to the FTP client.
6. Now the FTP client uses the received IP address to access the FTP server.

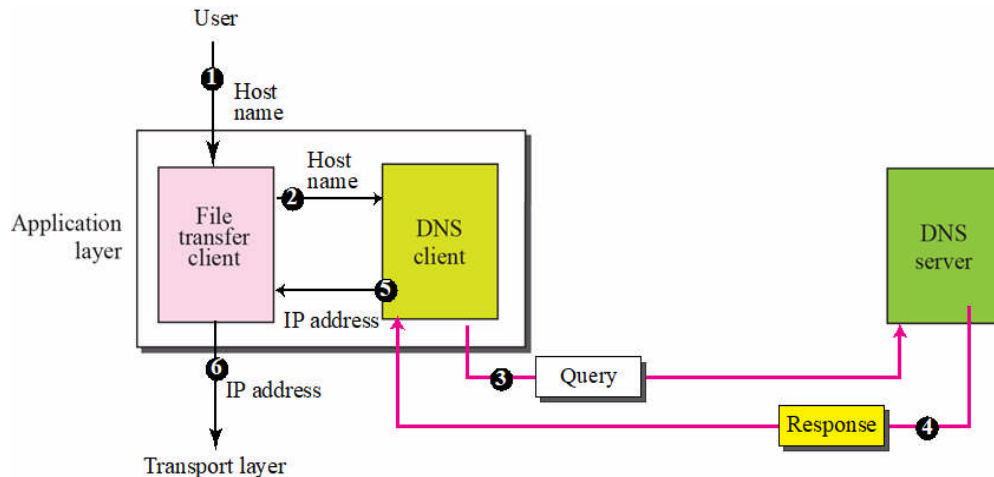


Figure 16.23: Working of DNS

Name Space

Internet is divided into over 200 top level domains. Each domain is divided into sub-domains, which are further partitioned. All domains can be represented by a tree. The leaves of the tree represent domains that have no sub-domains (but contain machines). A leaf domain may contain a single host or represent a company and contain thousands of hosts. Top level domains could be generic and country domains as shown in the Figure 16.24.

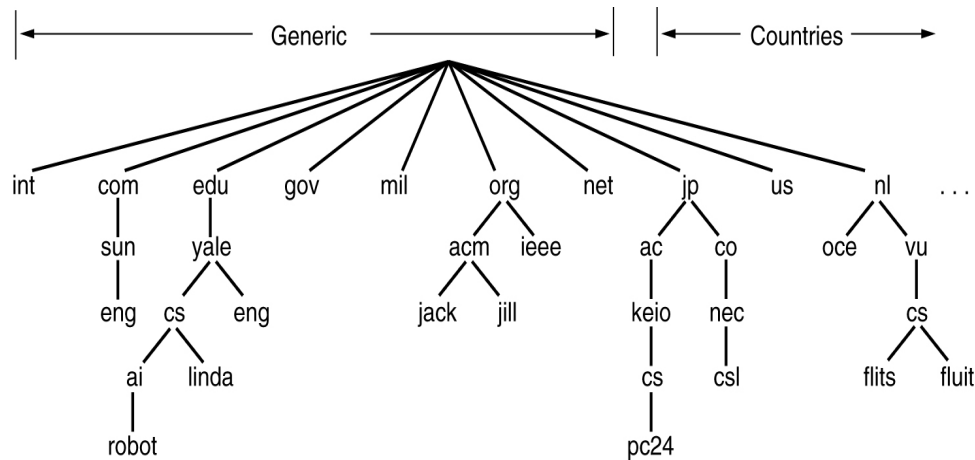


Figure 16.24: Domain Name space

The namespace needs to be made hierarchical to be able to scale.

The idea is to name objects based on

- Location (within country, set of organizations, set of companies, etc).
- Unit within that location (company within set of company, etc).
- Object within unit (name of person in company).

A domain name is the sequence of labels from a node to the root, separated by dots (".s), read from left to right. The name space has a maximum depth of 127 levels. Domain names are limited to 255 characters in length.

A node's domain name identifies its position in the name space. Each domain controls how it allocates the domains under it i.e. Japan makes a domains ac.jp and co.jp that may be different than edu and com. To create a new domain, permission is required from the domain that will include it; once created, it can create sub-domains without having to ask permission from the higher up domains.

Fully Qualified Domain Name (FQDN): If a label is terminated by a null string, it is called as fully qualified domain name (FQDN). An FQDN is a domain name that contains the full name of a host. A DNS server can only match an FQDN to an address. Note that the name must end with a null label, but because null means nothing, the label ends with a dot (.).

Partially Qualified Domain Name (PQDN): If a label is not terminated by a nullstring, it is called a partially qualified domain name (PQDN). A PQDN starts from a node, but it does not reach the root. Example of FQDN and PQDN are shown in the figure 16.25.

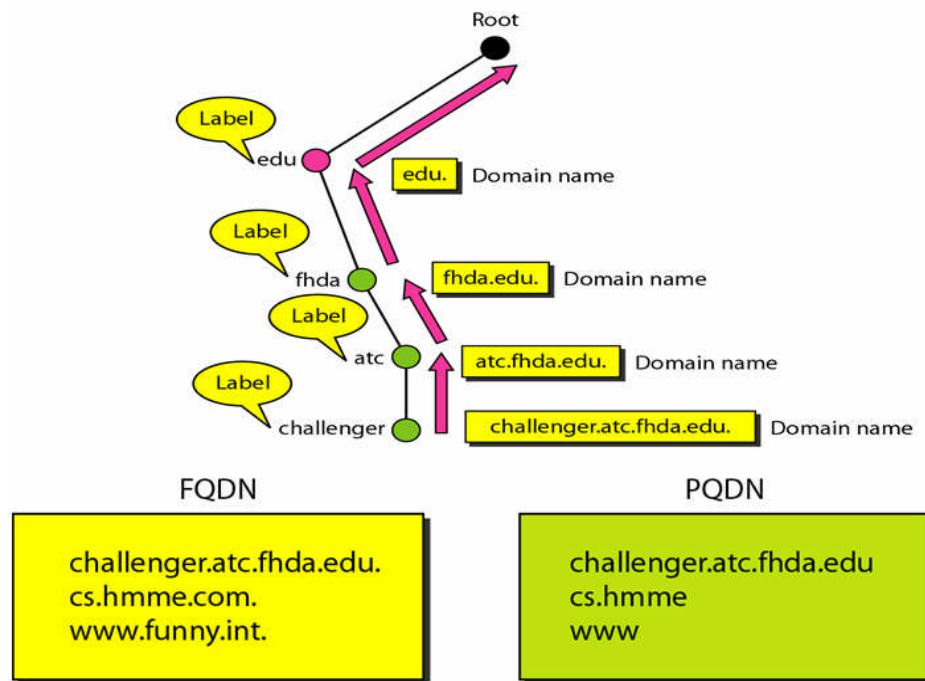


Figure 16.25: FQDN and PQDN

Distribution of Name Space

Storing the information comprised in the domain name space on one single computer is very inefficient and also not reliable because it is a huge amount of information. It is inefficient because responding to requests from all over the world places a heavy load on the system. It is not reliable because any failure makes the data inaccessible.

Hierarchy of Name Servers

The solution to this problem is to distribute the information among many computers called *DNS servers*. One way to do this is to divide the whole space into many domains based on the first level. In other words, we let the root stand alone and create as many domains (sub trees) as there are first-level nodes. Because a domain created this way could be very large, DNS allows domains to be divided further into smaller domains (sub domains). Each server can be responsible (authoritative) for either a large or small domain. In other words, we have a hierarchy of servers in the same way that we have a hierarchy of names (see Figure 16.26).

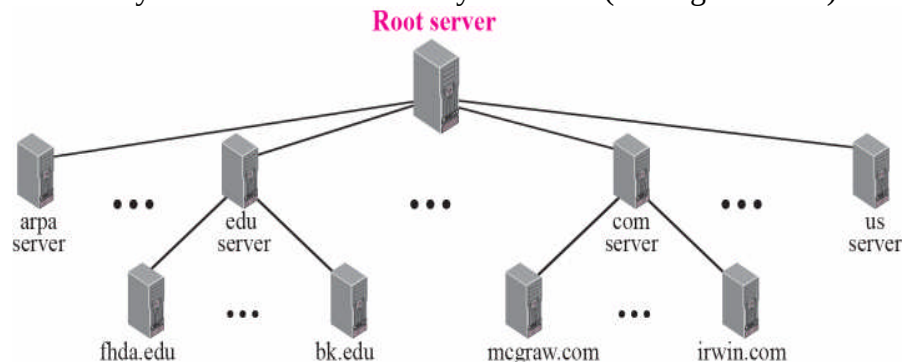


Figure 16.26: Hierarchy of Name Servers

Zone

Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers. What a server is responsible for or has authority over is called a **zone**. We can define a zone as a contiguous part of the entire tree. If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the “domain” and the “zone” refer to the same thing. The server makes a database called a *zone file* and keeps all the information for every node under that domain.

However, if a server divides its domain into sub domains and delegates part of its authority to other servers, “domain” and “zone” refer to different things. The information about the nodes in the sub domains is stored in the servers at the lower levels, with the original server keeping some sort of reference to these lower-level servers (see Figure 16.27).

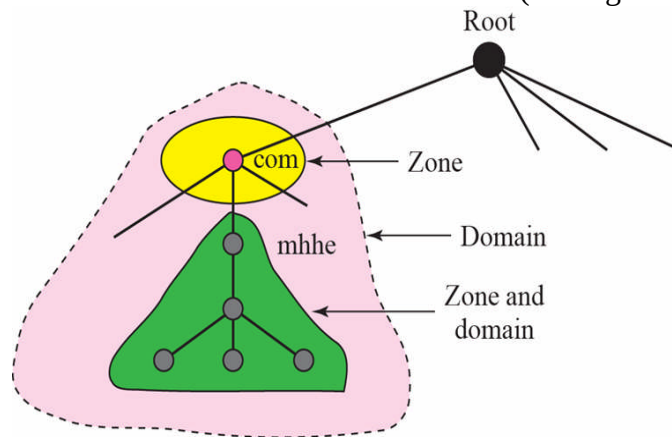


Figure 16.27: Zones and Domains

Root Server

A **root server** is a server whose zone consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers. There are several root servers, each covering the whole domain name space. The root servers are distributed all around the world.

Primary and Secondary Servers

DNS defines two types of servers: primary and secondary. A *primary server* is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.

A *secondary server* is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If updating is required, it must be done by the primary server, which sends the updated version to the secondary. The primary and secondary servers are both authoritative for the zones they serve.

Resolution

Mapping of Domain name to IP address or IP address to Domain name is called resolution.

Resolver

A host that needs to map a name to an address or an address to a name calls a DNS client called as resolver. The resolver accesses the closest DNS server with a mapping request.

Mapping Names to Addresses

When, the resolver gives a domain name to the server and asks for the corresponding address. In this case, the server checks the generic domains or the country domains to find the mapping.

Mapping Addresses to Names

When, the resolver gives an IP address to the server and asks for the corresponding domain name; this type of query is called as *PTR* query. To answer queries of this kind, DNS uses the inverse domain.

Recursive Resolution

The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer. If the server is the authority for the domain name, it checks its database and responds. If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response. If the parent is the authority, it responds; otherwise, it sends the query to yet another server. When the query is finally resolved, the response travels back until it finally reaches the requesting client (see Figure 16.28).

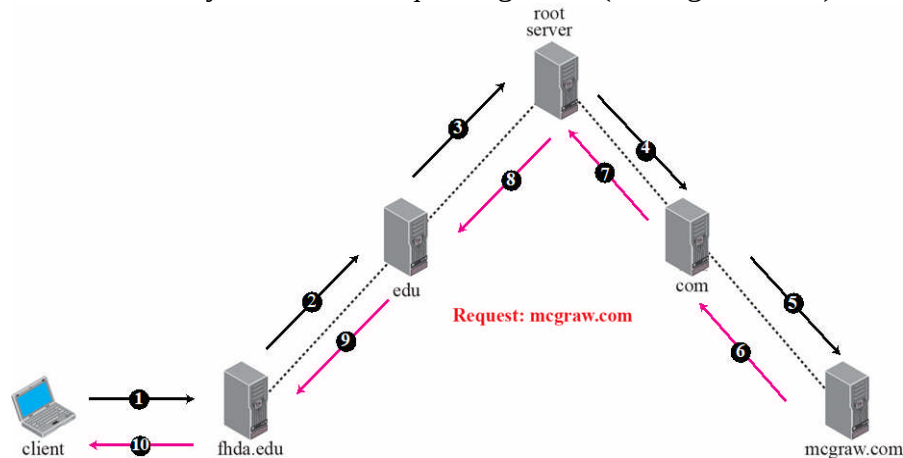


Figure 16.28: Recursive Resolution

Iterative Resolution

If the client does not ask for a recursive answer, the mapping can be done iteratively. If the server is an authority for the name, it sends the answer. If it is not, it returns (to the client) the IP address of the server that it thinks can resolve the query. The client is responsible for repeating the query to this second server. If the newly addressed server can resolve the

problem, it answers the query with the IP address; otherwise, it returns the IP address of a new server to the client. Now the client must repeat the query to the third server. This process is called *iterative* because the client repeats the same query to multiple servers (see Figure 16.29).

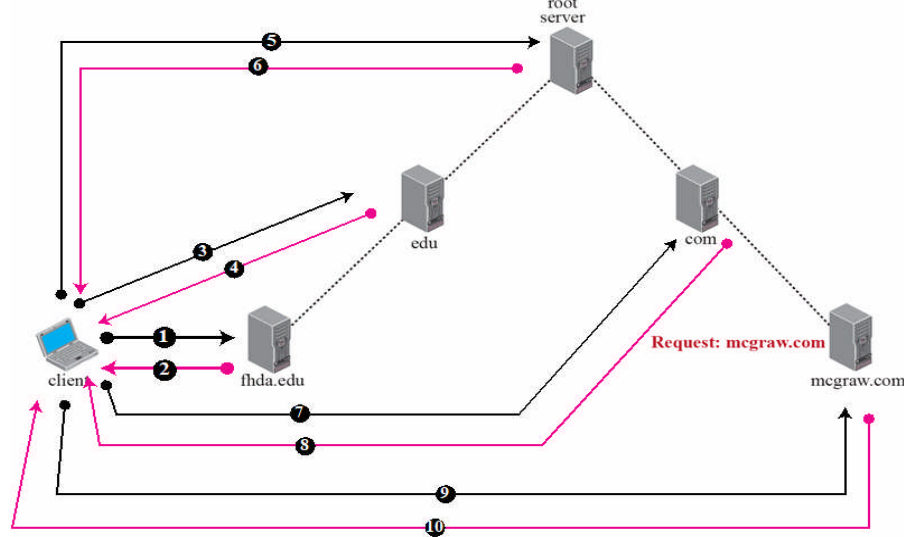


Figure 16.29: Iterative Resolution

Caching

Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address. Reduction of this search time would increase efficiency. DNS handles this with a mechanism called **caching**. When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client. If the same or another client asks for the same mapping, it can check its cache memory and resolve the problem. However, to inform the client that the response is coming from the cache memory and not from an authoritative source, the server marks the response as *unauthoritative*.

Caching speeds up resolution, but it can also be problematic. If a server caches a mapping for a long time, it may send an outdated mapping to the client. To solve this, two techniques are used. First, the authoritative server always adds information to the mapping called *time-to-live* (TTL). It defines the time in seconds that the receiving server can cache the information. After that time, the mapping is invalid and any query must be sent again to the authoritative server. Second, DNS requires that each server keep a TTL counter for each mapping it caches. The cache memory must be searched periodically and those mappings with an expired TTL must be purged.

DNS Messages

DNS has two types of messages: query and response. Both of them have the same format. The query message consists of a header and question records whereas the response message consists of a header,

question records, answer records, authoritative records, and additional records (see Figure 16.30).

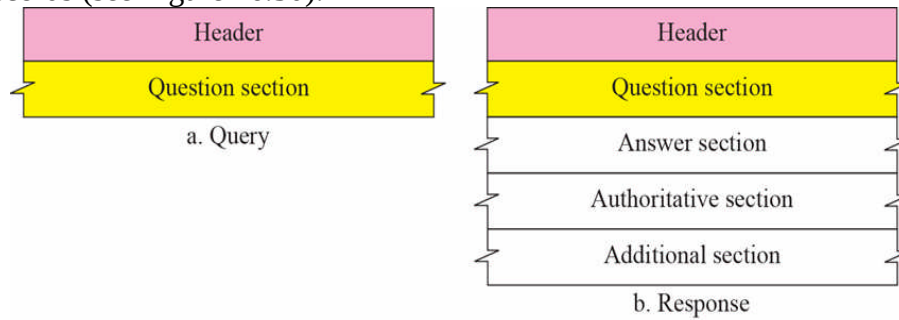


Figure 16.30: DNS – Query and Response message

Header: Both query and response messages have the same header format with some fields set to zero for the query messages. The header is 12 bytes and its format is shown in Figure 16.31.

Identification	Flags
Number of question records	Number of answer records (All 0s in query message)
Number of authoritative records (All 0s in query message)	Number of additional records (All 0s in query message)

Figure 16.31: Header format

The header fields are as follows:

- **Identification:** This is a 16-bit field used by the client to match the response with the query. The client uses a different identification number each time it sends a query. The server duplicates this number in the corresponding response.
- **Flags:** This is a 16-bit field consisting of the subfields that defines the type of message, type of answer requested, the type of desired resolution, and so on.
- **Number of question records:** This is a 16-bit field containing the number of queries in the question section of the message.
- **Number of answer records:** This is a 16-bit field containing the number of answer records in the answer section of the response message. Its value is zero in the query message.
- **Number of authoritative records:** This is a 16-bit field containing the number of authoritative records in the authoritative section of a response message. Its value is zero in the query message.
- **Number of additional records:** This is a 16-bit field containing the number of additional records in the additional section of a response message. Its value is zero in the query message.

Question Section: This is a section consisting of one or more question records. It is present on both query and response messages.

Answer Section: This is a section consisting of one or more resource records. It is present only on response messages. This section includes the answer from the server to the client (resolver). **Authoritative Section:** This is a section consisting of one or more resource records. It is present only on response messages. This section gives information (domain name) about one or more authoritative servers for the query.

Additional Information Section: This is a section consisting of one or more resource records. It is present only on response messages. This section provides additional information that may help there solver. For example, a server may give the domain name of an authoritative server to the resolver in the authoritative section, and include the IP address of the same authoritative server in the additional information section.

Resource Record

Each domain name is associated with a record called as *resource record*. The DNS server database consists of resource records. Resource records are also what is returned by the server to the client. Figure 16.32 shows the format of are source record.

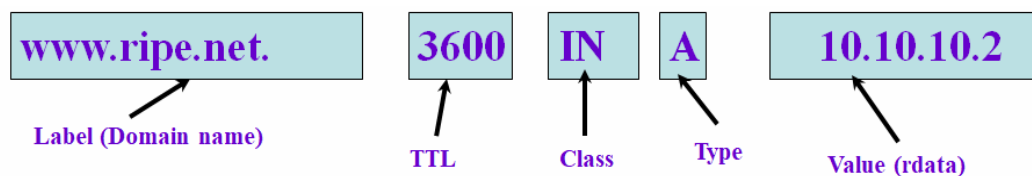


Figure 16.32: DNS – Resource Record

A resource record has five parts namely Domain name, Time to Live (TTL), Class, Type and Value.

- ❖ **Domain Name:** The Domain name tells the domain to which this record applies. Normally many records exist for each domain and each copy of the database holds information about multiple domains. This field is the primary search key used to satisfy queries. The order of the records in the database is not important.
- ❖ **Time to Live (TTL):** Time to live field gives an indication of how stable the record is. Information that is highly stable is assigned a large value, such as 86400 (number of seconds in a day). Information that is highly volatile is assigned a small value, such as 60 seconds (1 minute).
- ❖ **Class:** Class field is always *IN* for Internet information.
- ❖ **Type:** Type field tells what kind of record this is (see the Table 16.11).
- ❖ **Value:** Value field gives the resource data value (IP address for type 'A' record).

Resource Record types: there are eight types of DNS records as shown in the Table 16.11.

- **Start of Authority (SOA):** SOA record provides the name of the primary source of information about the name server's zone, the e-mail address of its administrator, a unique serial number and various flags and timeouts.

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

Table 16.11: Resource Record types

- **Address (A):** Address record is the most important record. It holds a 32 bit IP address for some host; Every Internet host must have at least one IP address; some hosts have two or more IP addresses (being connected to multiple networks, having one type A resource record per network connection); DNS can be made to cycle through those (for first request to return first record, for second request to return the second A type record).
- **Mail Exchange (MX):** MX record specifies the name of the host prepared to accept e-mail for the specified domain; it is used because not every machine is prepared to accept e-mail. If someone wants to send e-mail to *bill@microsoft.com*, the sending host needs to find a mail server at microsoft.com that is willing to accept e-mail. MX record can provide this information.
- **Name Server (NS):** The NS record specifies Name Servers i.e., every DNS database normally has an NS record for each of the top-level domains.
- **Canonical Name (CNAME):** The CNAME records allow aliases to be created. In example: *cs.mit.edu 86400 IN CNAME 1cs.mit.edu* creates an alias for 1cs.mit.edu (real domain name).
- **Pointer (PTR):** The PTR record used to associate a name with an IP address to allow lookups of the IP address and return the name of the corresponding machine. This is called reverse lookup.

16.8 SUMMARY

- In Client-server architecture, client is the consumer of the services provided by the server. We discussed different client-server application programs such as HTTP, FTP, SMTP, POP3, IMAP4, MIME, TELNET, Secure shell and DNS.

- World Wide Web is *repository of information* linked together from various points all over the world is called as World Wide Web (WWW). It consists of different components such client (browser), server, URL, cookies and web documents
- HTTP is used in WWW to transfer web pages from one host to other over internet. In HTTP, the commands from the client to the server are embedded in *a request message*. The contents of the requested file or other information are embedded in *a response message*. HTTP uses the services of TCP on well-known port 80.
- FTP is standard mechanism provided by TCP/IP for copying a file from one host to another. FTP is different than other client-server programs. It establishes two connections between hosts. One is used for data transfer and other is used for control information (exchange of Commands and Responses). FTP uses the services of TCP, it needs two connections: well-known *port 21* is used for the Control connection and *port 20* is used for the Data connection.
- *An Electronic - mail* is one of the most popular Internet services. At the beginning of the internet era, the messages sent by email were *short* and consisted of *text* only. Today email is much more complex, it allows a message to include text, audio, and video. It also allows one message to be sent to one or more recipients.
- This is the most common email architecture used today in which when both sender and receiver are connected to their mail servers via a LAN or a WAN, we need two UAs and two pair of MTAs (client and server), and pair of MAAs (client and server).
- **User Agent (UA):** it provides service to the user to make the process of sending and receiving a message easier. Two types of UAs – Command driven UA and GUI based UA.
- Multipurpose Internet Mail Extensions (MIME), which is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client MTA to be sent through the Internet. The message at the receiving site is transformed back to the original data.
- **Message Transfer Agents (MTA):** a client-server program used to transfer the message across the internet. To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The protocol that defines the MTA client and server in the Internet is called as Simple Mail Transfer Protocol (SMTP).SMTP uses Commands and Responses to transfer the mail messages between SMTP client and SMTP server.
- **Message Access Agent (MAA):** a client-server program that pulls the stored email messages.POP3 and IMAP4 are two MAAs that are used to pull the stored emails.

- The *TELNET* is an abbreviation for *TERminal NETwork*. It is the standard TCP/IP protocol for virtual terminal service as proposed by the International Organization for Standards (ISO). TELNET enables the establishment of a connection with a remote computer in such a way that the local terminal appears to be a terminal at the remote computer.
- Another most popular application program used for remote login is Secure Shell (SSH). SSH, like TELNET, uses TCP's service, but SSH is more secure and provides more services than TELNET.
- We need a mechanism or system that can translate or map the host name to IP address or IP address to host name. In the internet such mechanism or system is provided by one of the application layer protocol called as Domain Name System (DNS).

16.9 REFERENCE FOR FURTHER READING

For more details about topics discussed in this chapter, we recommend the following books.

1. *Data Communication and Networking* by Behrouz A. Forouzan, McGraw-Hill, 2007.
2. *TCP/IP Protocol Suite* by Behrouz A. Forouzan, McGraw-Hill, 2010.

16.10 MODEL QUESTIONS

1. What is WWW and how the HTTP is related to WWW?
2. How is HTTP similar to SMTP?
3. How is HTTP similar to FTP?
4. What is a URL and what are its components?
5. What are the three types of Web documents?
6. What is remote log-in in TELNET?
7. How are the control and data characters distinguished in NVT?
8. How are options negotiated in TELNET?
9. How Secure Shell is different than TELNET?
10. In electronic mail, what are the tasks of a user agent?
11. What is MIME?
12. Why do we need POP3 or IMAP4 for electronic mail?
13. What is the purpose of FTP?
14. Describe the functions of the two FTP connections.
15. What kinds of file types can FTP transfer?
16. What is anonymous FTP?
17. What is a purpose of DNS?

18. What is the Role of a primary server and a secondary server in DNS?
19. How does recursive resolution differ from iterative resolution?
20. What are FQDN and a PQDN?
21. How does caching increase the efficiency of name resolution?
22. What are the types of DNS messages?

