

**Notes
of
Data communication and networking**

UNIT I

1.1 INTRODUCTION TO NETWORKS & DATA COMMUNICATIONS

The term telecommunication means communication at a distance. The word data refers to information presented in whatever form is agreed upon by the parties creating and using the data. Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable. For Data Communication to occur, the communicating devices must be a part of a communication system made up of a combination of hardware and software.

The effectiveness of a data communication system depends on four fundamental characteristics:

1. Delivery
2. Accuracy
3. Timeliness
4. Jitter

There are five components of data communication as shown in Fig. 1.1 below:

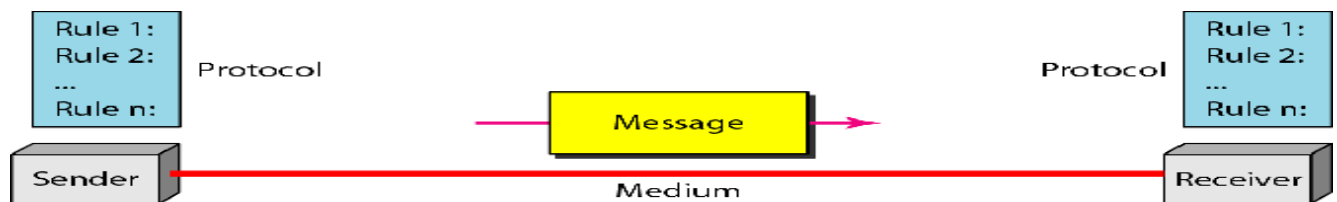


Fig 1.1 Components of Data Communication

- (a) **Sender:** is the device that sends the data message.
- (b) **Message:** is the information (data) to be communicated. Eg: text, numbers etc.
- (c) **Transmission Medium:** is the physical path by which a message travels from sender to receiver. Eg: twisted pair cable, fiber-optic cable etc.
- (d) **Receiver:** is the device that receives the message.
- (e) **Protocols:** is a set of rules that govern the data communication. It represents an agreement between the communicating devices.

Moreover, Data can flow in three different ways namely Simplex, Half- Duplex and Full Duplex. In simplex mode, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive. In half-duplex mode, each station can both transmit and receive, but not at the same time. i.e. When one device is sending, the other can only receive, and vice versa. Whereas, in full-duplex mode (also called duplex), both stations can transmit and receive simultaneously.

A network is a set of devices (often referred to as nodes) connected by communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.

A network must be able to meet these three criteria's:

1. Performance: can be measured using Transit time and Response time:
 - (a) Transit Time: is the time required for a message to travel from one device to another.
 - (b) Response Time: is the elapsed time between an inquiry and a response.
2. Reliability: is measured by the frequency of failure i.e the time it takes a link to recover from a failure.
3. Security: issues include protecting data from unauthorized access and losses.

Furthermore, there are two types of connection: Point to Point and Multipoint. In Point -to-Point: Connection provides a dedicated link between two devices. Whereas, in Multi-Point: Connection is one in which more than two devices share a single link.

Network Categories: The category into which a network falls is determined by its size. Network can be categorized as: LAN, WAN, MAN, Wireless Network and Internetwork.

LANs are designed to allow resources to be shared between personal computers or workstations. The resources to be shared can include hardware (e.g., a printer), software (e.g., an application program), or data. LANs are distinguished from other types of networks by their transmission media and topology. In general, a given LAN will use only one type of transmission medium. The most common LAN topologies are bus, ring, and star. Early LANs had data rates in the 4 to 16 megabits per second (Mbps) range. Today, however, speeds are normally 100 or 1000 Mbps

A wide area network (WAN) provides long-distance transmission of data, image, audio, and video information over large geographic areas that may comprise a country, a continent, or even the whole world. A WAN can be as complex as the backbones that connect the Internet or as simple as a dial-up line that connects a home computer to the Internet. The switched WAN connects the end systems, which usually comprise a router (internetworking connecting device) that connects to another LAN or WAN. The point-to-point WAN is normally a line leased from a telephone or cable TV provider that connects a home computer or a small LAN to an Internet service provider (ISP). This type of WAN is often used to provide Internet access.

A metropolitan area network (MAN) is a network with a size between a LAN and a WAN. It normally covers the area inside a town or a city. It is designed for customers who need a high-speed connectivity, normally to the Internet, and have endpoints spread over a city or part of city. A good example of a MAN is the part of the telephone company network that can provide a high-speed DSL line to the customer. Another example is the cable TV network that originally was designed for cable TV, but today can also be used for high-speed data connection to the Internet.

1.2 THE INTERNET

The Internet is a global, interconnected computer network in which every computer connected to it can exchange data with any other connected computer. Rather than moving through geographical space, it moves your ideas and information through cyberspace – the space of electronic movement of ideas and information.

Significance of an Internet are as follows:

- It's the first mass medium that involves computers and uses digitized data.
- It provides the potential for media convergence, the unification of all media.
- It's transforming how we communicate, obtain information, learn, seek jobs, and maintain professional growth.
- Businesses find it an indispensable tool for their needs.

In the late 1950's the Advanced Research Projects Agency (ARPA) was founded in the United States with the primary focus of developing information technologies that could survive a nuclear attack. Scientists and military experts were especially concerned about what might happen in the event of a Soviet attack on the nation's telephone system. Just one missile, they feared, could destroy the whole network of lines and wires that made efficient long-distance communication possible. In 1962, a scientist from M.I.T. and ARPA named J.C.R. Licklider proposed a solution to this problem: a "galactic network" of computers that could talk to one another. Such a network would enable government leaders to communicate even if the Soviets destroyed the telephone system. In 1965, another M.I.T. scientist developed a way of sending information from one computer to another that he called "packet switching." Packet switching breaks data down into blocks, or packets, before sending it to its destination. That way, each packet can take its own route from place to place. In 1967 ARPA university and private sector contractors met with representatives of the Department of Defense to discuss possible protocols for sharing information via computers. Thus in 1969, ARPAnet delivered its first message: a "node-to-node" communication from one computer to another. It was a connection of computers at UCLA, Stanford, UCSB, Univ. of Utah.

The Internet has come a long way since the 1960s. The Internet today is not a simple hierarchical structure. It is made up of many wide- and local-area networks joined by connecting devices and switching stations. It is difficult to give an accurate representation of the Internet because it is continually changing-new networks are being added, existing networks are adding addresses, and networks of defunct companies are being removed. Today most end users who want Internet connection use the services of Internet service providers (ISPs). There are: International service providers, National service providers, Regional service providers and Local service providers. The Internet today is run by private companies, not by the government.

Hierarchical organization of the Internet is shown in Fig. 1.2 below as (a) structure of a national ISP and (b) Interconnection of national ISPs.

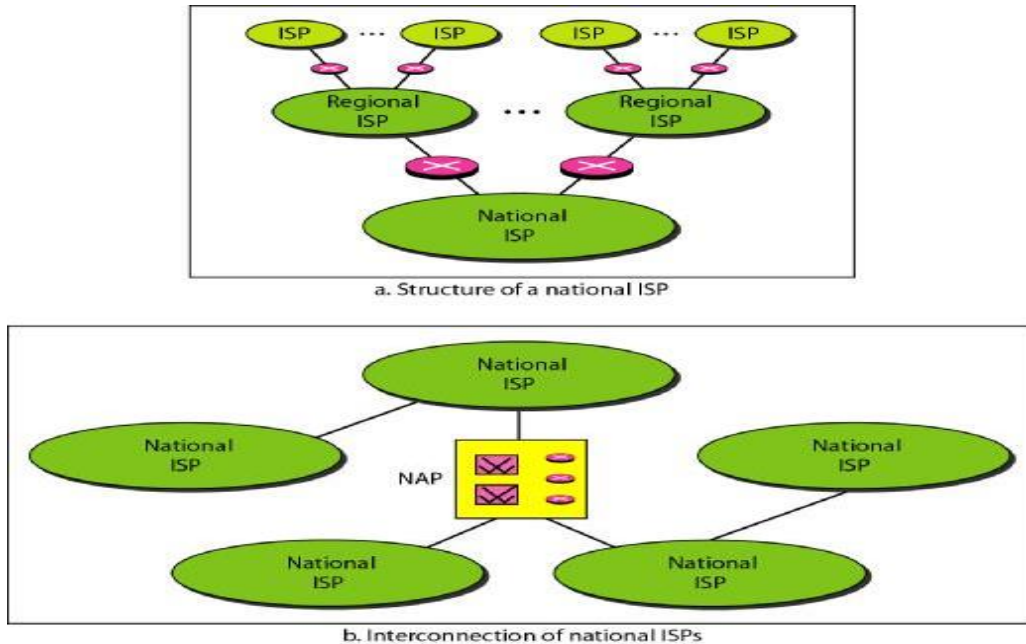


Fig 1.2 (a) Structure of a National ISP and (b) Interconnection of National ISPs

International Internet Service Providers: At the top of the hierarchy are the international service providers that connect nations together.

National Internet Service Providers: The national Internet service providers are backbone networks created and maintained by specialized companies. There are many national ISPs operating in North America; some of the most well known are Sprint Link, PSI Net, UUNet Technology, AGIS, and internet Mel. To provide connectivity between the end users, these backbone networks are connected by complex switching stations (normally run by a third party) called network access points (NAPs). Some national ISP networks are also connected to one another by private switching stations called peering points. These normally operate at a high data rate (up to 600 Mbps).

Regional Internet Service Providers: Regional internet service providers or regional ISPs are smaller ISPs that are connected to one or more national ISPs. They are at the third level of the hierarchy with a smaller data rate.

Local Internet Service Providers: Local Internet service providers provide direct service to the end users. The local ISPs can be connected to regional ISPs or directly to national ISPs. Most end users are connected to the local ISPs.

1.3 PROTOCOLS & STANDARDS

In computer networks, communication occurs between entities in different systems. An entity is anything capable of sending or receiving information. However, two entities cannot simply send bit streams to each other and expect to be understood. Thus, for communication to occur, the entities must agree on a protocol. Therefore, a protocol is a set of rules that govern data communications. A protocol defines: what is communicated, how it is communicated, & when it is communicated.

There are three elements of a protocol:

- **Syntax:** The term syntax refers to the structure or format of the data, meaning the order in which they are presented.
- **Semantics:** The word semantics refers to the meaning of each section of bits. How is a particular pattern to be interpreted, and what action is to be taken based on that interpretation?
- **Timing:** The term timing refers to two characteristics: when data should be sent and how fast they can be sent.

Standards provide guidelines to manufacturers, vendors, government agencies, and other service providers to ensure the kind of interconnectivity necessary in today's marketplace and in international communication. Standards are developed through the cooperation of standards creation committees, forums, and government regulatory agencies. The various standard creation committees are:

- **International Organization for Standardization (ISO):** The ISO is active in developing cooperation in the realms of scientific, technological, and economic activity.
- **International Telecommunication Union-Telecommunication Standards Sector (ITU-T):** By the early 1970s, a number of countries were defining national standards for telecommunications, but there was still little international compatibility. The United Nations responded by forming, as part of its International Telecommunication Union (ITU), a committee, the Consultative Committee for International Telegraphy and Telephony (CCITT). This committee was devoted to the research and establishment of standards for telecommunications in general and for phone and data systems in particular. On March 1, 1993, the name of this committee was changed to the International Telecommunication Union Telecommunication Standards Sector (ITU-T).
- **American National Standards Institute (ANSI):** Despite its name, the American National Standards Institute is a completely private, nonprofit corporation not affiliated with the U.S. federal government. However, all ANSI activities are undertaken with the welfare of the United States and its citizens occupying primary importance.
- **Institute of Electrical and Electronics Engineers (IEEE):** The Institute of Electrical and Electronics Engineers is the largest professional engineering society in the world. International in scope, it aims to advance theory, creativity,

and product quality in the fields of electrical engineering, electronics, and radio as well as in all related branches of engineering.

- **Electronic Industries Association (EIA):** Aligned with ANSI, the Electronic Industries Association is a nonprofit organization devoted to the promotion of electronics manufacturing concerns. Its activities include public awareness education and lobbying efforts in addition to standards development.

1.4 LAYERED TASKS : OSI MODEL

We use the concept of layers in our daily life. As an example, let us consider two friends who communicate through postal mail. The process of sending a letter to a friend would be complex if there were no services available from the post office. Figure 1.3 below shows tasks involved in sending a letter:

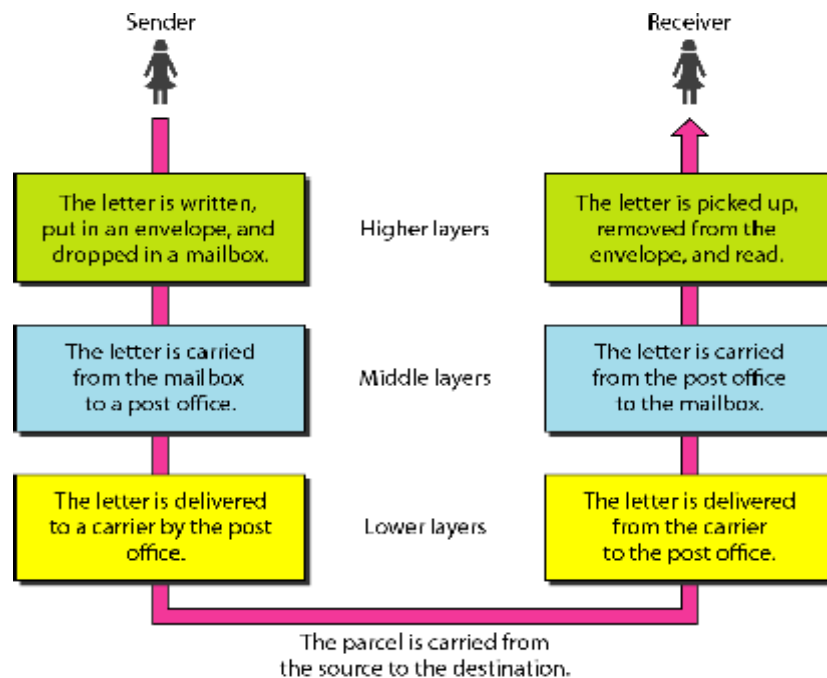


Fig 1.3 Layered Tasks

Thus from above figure it is clearly understood that layer architecture simplifies the network design. It is easy to debug network applications in a layered architecture network. There are two layered Models namely OSI Model and TCP/IP Model.

OSI MODEL: OPEN SYSTEM FOR INTERCONNECTION

International Standard Organization (ISO) established a committee in 1977 to develop architecture for computer communication. Open Systems Interconnection (OSI) reference model is the result of this effort.

In 1984, the Open Systems Interconnection (OSI) reference model was approved as an international standard for communications architecture. Term "open" denotes the ability to

connect any two systems which conform to the reference model and associated standards. The purpose of OSI Model is to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. The OSI model is now considered the primary Architectural model for inter-computer communications. The OSI model describes how information or data makes its way from application programmes (such as spreadsheets) through a network medium (such as wire) to another application programme located on another network. The OSI reference model divides the problem of moving information between computers over a network medium into SEVEN smaller and more manageable problems. This separation into smaller more manageable functions is known as layering. Figure below shows interaction between layers in the OSI model:

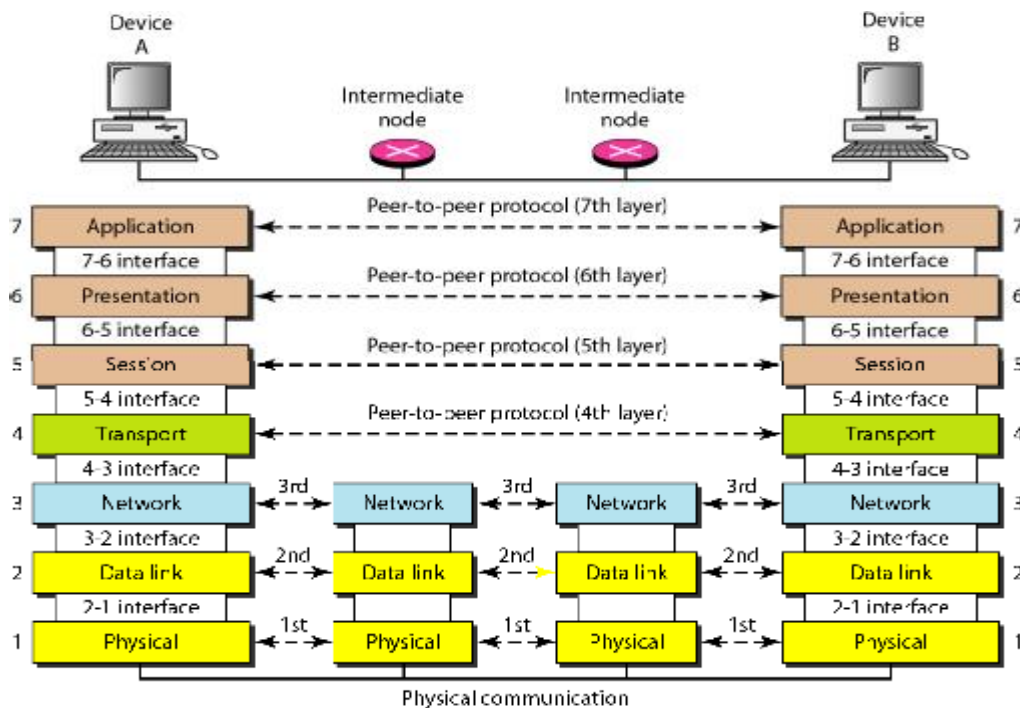


Fig 1.4 OSI Model

The process of breaking up the functions or tasks of networking into layers reduces complexity. Each layer provides a service to the layer above it in the protocol specification. Each layer communicates with the same layer's software or hardware on other computers. The lower 4 layers (transport, network, data link and physical —Layers 4, 3, 2, and 1) are concerned with the flow of data from end to end through the network. The upper four layers of the OSI model (application, presentation and session—Layers 7, 6 and 5) are orientated more toward services to the applications. Data is encapsulated with the necessary protocol information as it moves down the layers before network transit. A message begins at the top application layer and moves down the OSI layers to the bottom physical layer. As the message descends, each successive OSI model layer adds a header to it. A header is layer-specific information that basically explains what functions the layer carried out. Conversely, at the receiving end, headers are striped from the message as it travels up the corresponding layers as shown in Fig.1.5.

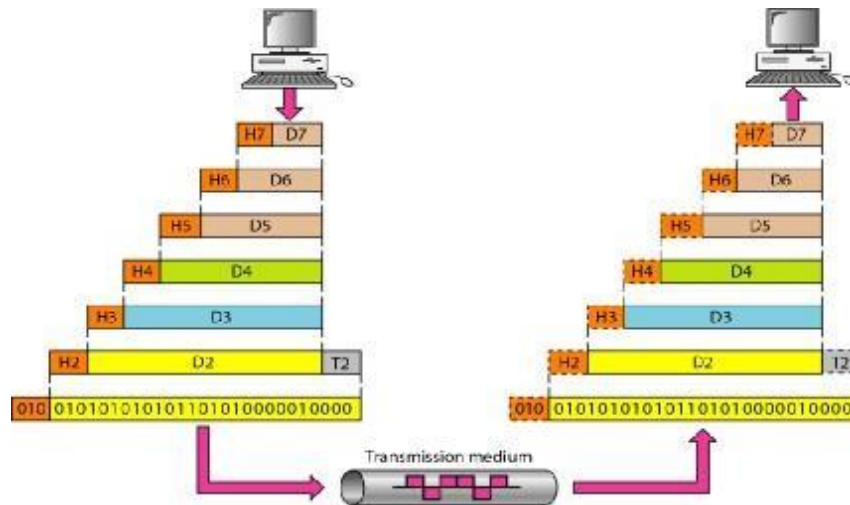


Fig 1.5 Working Principle

A) PHYSICAL LAYER

- Provides physical interface for transmission of information.
- Defines rules by which bits are passed from one system to another on a physical communication medium.
- Covers all - mechanical, electrical, functional and procedural - aspects for physical communication.
- Such characteristics as voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, physical connectors, and other similar attributes are defined by physical layer specifications.
- Concerned with line configuration, physical topology and transmission mode.

B) DATA LINK LAYER

- Data link layer attempts to provide reliable communication over the physical layer interface.
- Breaks the outgoing data into frames and reassemble the received frames.
- Create and detect frame boundaries.
- Handle errors by implementing an acknowledgement and retransmission scheme.
- Implement flow control.
- Responsible for Error Control.
- Supports points-to-point as well as broadcast communication.
- Supports simplex, half-duplex or full-duplex communication.

C) NETWORK LAYER

- Implements routing of frames (packets) through the network.
- Defines the most optimum path the packet should take from the source to the destination.
- Defines logical addressing so that any endpoint can be identified.

- Handles congestion in the network.
- The network layer also defines how to fragment a packet into smaller packets to accommodate different media.

D) TRANSPORT LAYER

- Purpose of this layer is to provide a reliable mechanism for the exchange of data between two processes in different computers.
- Ensures that the data units are delivered error free.
- Ensures that data units are delivered in sequence.
- Ensures that there is no loss or duplication of data units.
- Provides connectionless or connection oriented service.
- Provides for the connection management.
- Multiplex multiple a connection over a single channel.

E) SESSION LAYER

- Session layer provides mechanism for controlling the dialogue between the two end systems.
- It defines how to start, control and end conversations (called sessions) between applications.
- This layer requests for a logical connection to be established on an end-user's request.
- Any necessary log-on or password validation is also handled by this layer.
- Session layer is also responsible for terminating the connection.
- This layer provides services like dialogue discipline which can be full duplex or half duplex.
- Session layer can also provide check-pointing mechanism such that if a failure of some sort occurs between checkpoints, all data can be retransmitted from the last checkpoint.

F) PRESENTATION LAYER

- Presentation layer defines the format in which the data is to be exchanged between the two communicating entities.
- Also handles data compression and data encryption (cryptography).

G) APPLICATION LAYER

- Application layer interacts with application programs and is the highest level of OSI model.
- Application layer contains management functions to support distributed applications.
- Examples of application layer are applications such as file transfer, electronic mail, remote login etc.

Fig. 1.6 below shows summary of all layers of OSI Model:

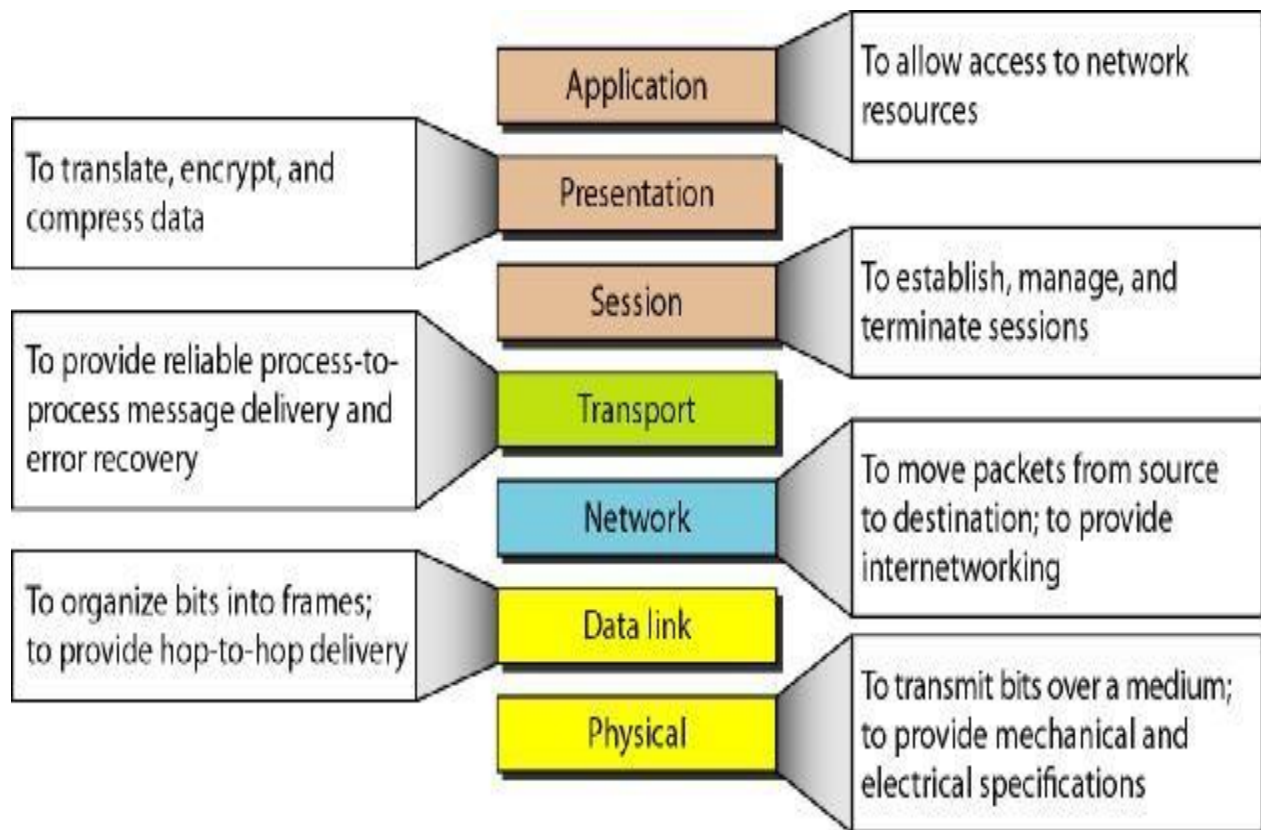


Fig 1.6 Summary of OSI Model

1.5 TCP/IP MODEL: (TRANSMISSION CONTROL PROTOCOL/ INTERNET PROTOCOL)

The layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application. However, when TCP/IP is compared to OSI, we can say that the TCP/IP protocol suite is made of five layers: physical, data link, network, transport, and application. Fig. 1.7 below shows TCP/IP layers in comparison to OSI Model:

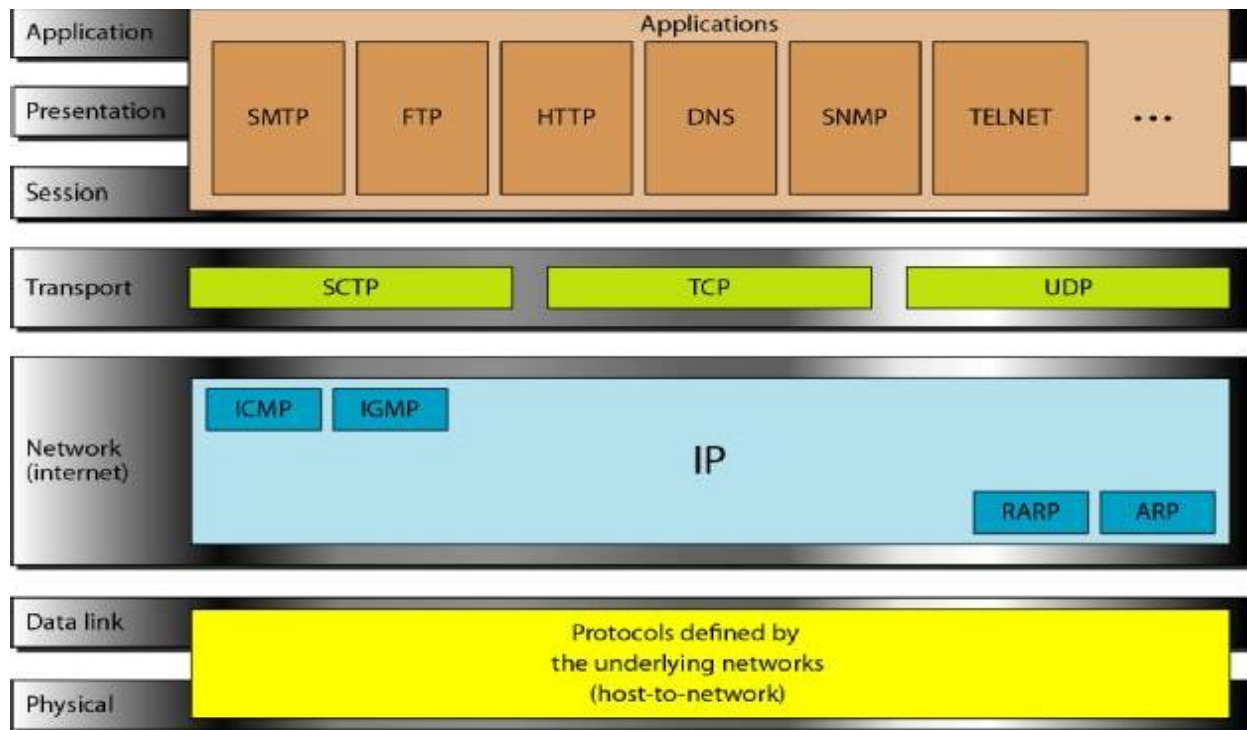


Fig 1.7 TCP/IP Model Vs OSI Model

A) APPLICATION LAYER

Application layer protocols define the rules when implementing specific network applications. It relies on the underlying layers to provide accurate and efficient data delivery. Typical protocols are:

- FTP – File Transfer Protocol: For file transfer
- Telnet – Remote terminal protocol: For remote login on any other computer on the network
- SMTP – Simple Mail Transfer Protocol: For mail transfer
- HTTP – Hypertext Transfer Protocol: For Web browsing

B) TRANSPORT LAYER

Transport Layer protocols define the rules of dividing a chunk of data into segments and then reassemble segments into the original chunk. Typical protocols are: TCP – Transmission Control Protocol: Provide functions such as reordering and data resend.

- UDP – User Datagram Service: Use when the message to be sent fit exactly into a datagram and Use also when a more simplified data format is required.
- SCTP - Stream Control Transmission Protocol: The Stream Control Transmission Protocol (SCTP) provides support for newer applications such as voice over the Internet.

C) NETWORK LAYER

Network layer protocols define the rules of how to find the routes for a packet to the destination. It only gives best effort delivery. Packets can be delayed, corrupted, lost, duplicated, out-of-order.

- IP – Internet Protocol: Provide packet delivery
- ARP – Address Resolution Protocol: Define the procedures of network address / MAC address translation i.e The Address Resolution Protocol (ARP) is used to associate a logical address with a physical address. ARP is used to find the physical address of the node when its Internet address is known.
- RARP – Reverse Address Resolution Protocol: The Reverse Address Resolution Protocol (RARP) allows a host to discover its Internet address when it knows only its physical address.
- ICMP – Internet Control Message Protocol: The Internet Control Message Protocol (ICMP) is a mechanism used by hosts and gateways to send notification of datagram problems back to the sender.
- IGMP – Internet Control Message Protocol: The Internet Group Message Protocol (IGMP) is used to facilitate the simultaneous transmission of a message to a group of recipients.

D) PHYSICAL AND DATA LINK LAYER

At the physical and data link layers, TCP-IP does not define any specific protocol. Rather, it supports all the standard protocols.

1.6 ADDRESSING

There are four levels of addresses are used in an internet employing the TCP/IP protocols: physical, logical, port, and specific. Figure 1.8 below shows relationship of layers and addresses in TCP/IP:

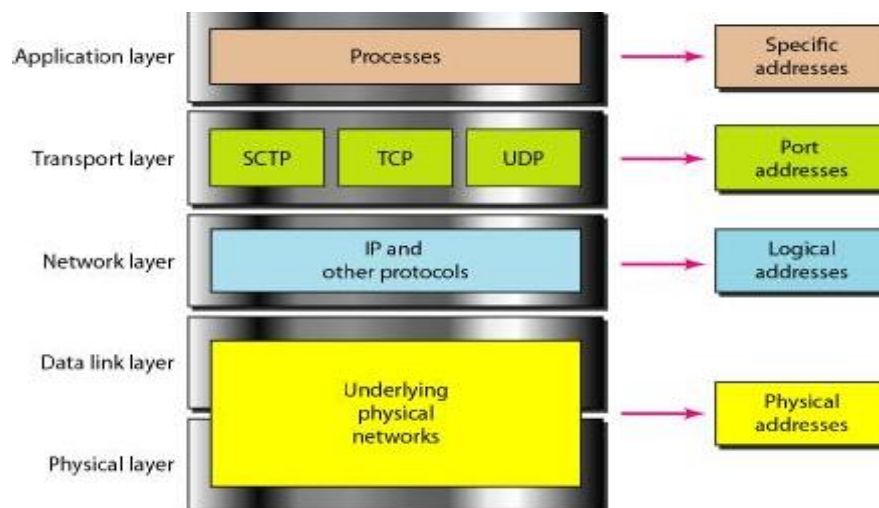


Fig 1.8 TCP/IP Addressing

A) PHYSICAL ADDRESSING:

The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN. It is included in the frame used by the data link layer. It is the lowest-level address. The size and format of these addresses vary depending on the network. Example of Physical Addressing: A node with physical address 10 sends a frame to a node with physical address 87. The two nodes are connected by a link (bus topology LAN). As the figure 1.9 shows, the computer with physical address 10 is the sender, and the computer with physical address 87 is the receiver.

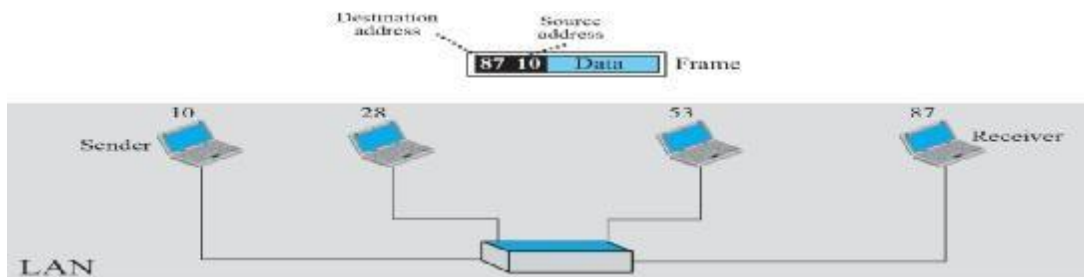


Fig 1.9 Example of Physical Addressing

B) LOGICAL ADDRESSING

Logical addresses are necessary for universal communications that are independent of underlying physical networks. Physical addresses are not adequate in an internetwork environment where different networks can have different address formats. A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network. The logical addresses are designed for this purpose. A logical address in the Internet is currently a 32-bit address that can uniquely define a host connected to the Internet. No two publicly addressed and visible hosts on the Internet can have the same IP address. Example of Logical addressing: Figure 1.10 shows a part of an internet with two routers connecting three LANs.

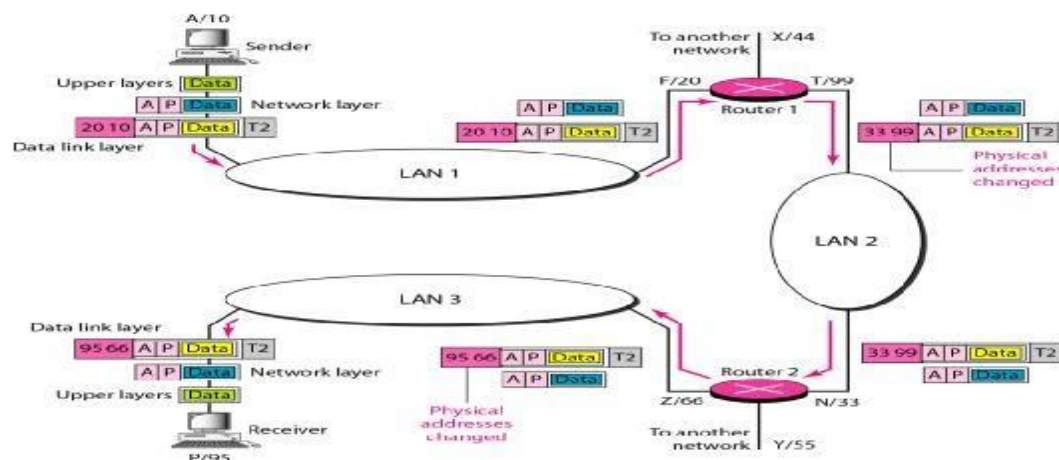


Fig 1.10 Example of Logical Addressing

C) PORT ADDRESSING:

The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. A system that sends nothing but data from one computer to another is not complete. Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process. For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes. In other words, they need addresses. In the TCPIIP architecture, the label assigned to a process is called a port address. A port address in TCPIIP is 16 bits in length. Example of Port addressing: Figure 1.11 shows two computers communicating via the Internet. The sending computer is running three processes at this time with port addresses a, b, and c. The receiving computer is running two processes at this time with port addresses j and k. Process a in the sending computer needs to communicate with process j in the receiving computer. Note that although both computers are using the same application, FTP, for example, the port addresses are different because one is a client program and the other is a server program.

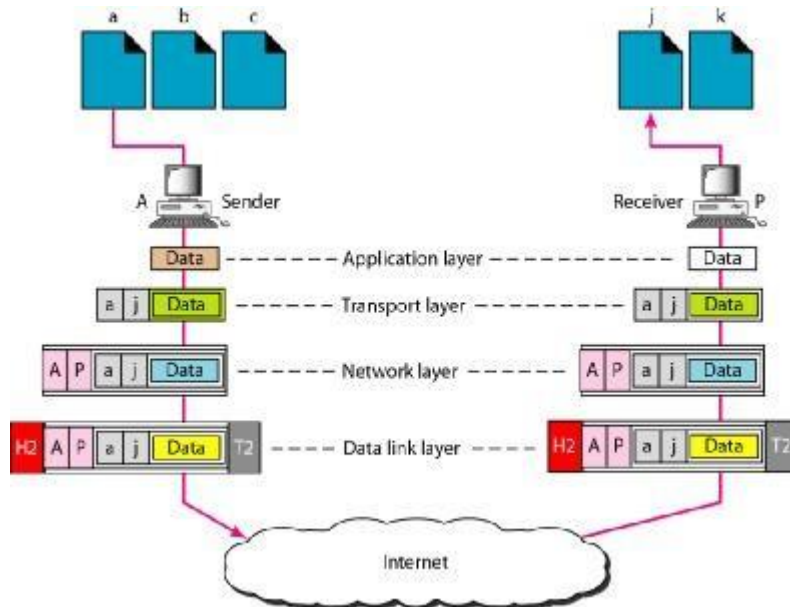


Fig 1.11 Example of Port Addressing

D) SPECIFIC ADDRESSING:

Some applications have user-friendly addresses that are designed for that specific address. Examples include the e-mail address (for example, forouzan@fhda.edu): defines the recipient of an e-mail and the Universal Resource Locator (URL) (for example, www.mhhe.com): used to

find a document on the World Wide Web. These addresses, however, get changed to the corresponding port and logical addresses by the sending computer.

1.7 LINE CODING REVIEW

Line coding is the process of converting digital data to digital signals. 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. We can roughly divide line coding schemes into five broad categories as shown below in Fig.1.12:

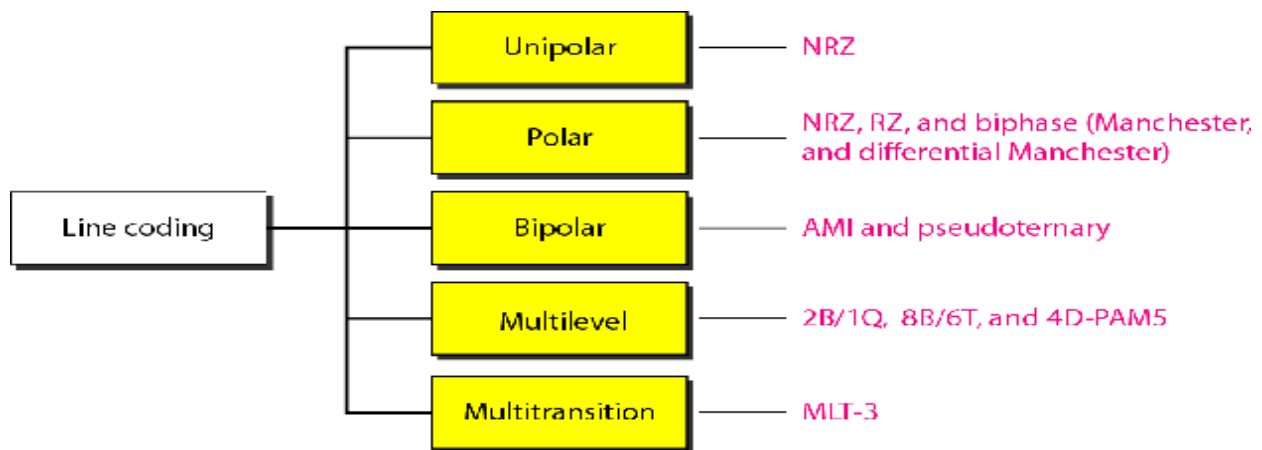


Fig 1.12 Line Coding Schemes

Unipolar Scheme: 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 as shown in Fig 1.13.

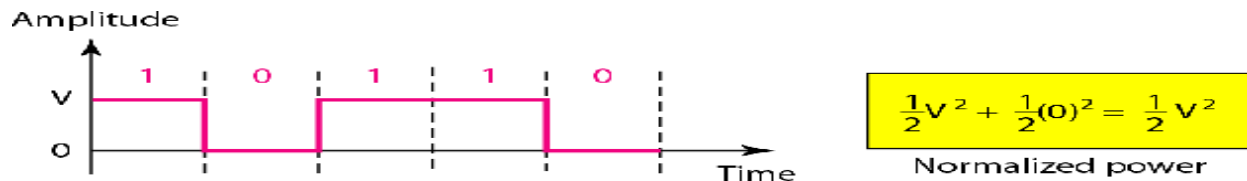


Fig 1.13 Unipolar Schemes

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 as shown in Fig 1.14.

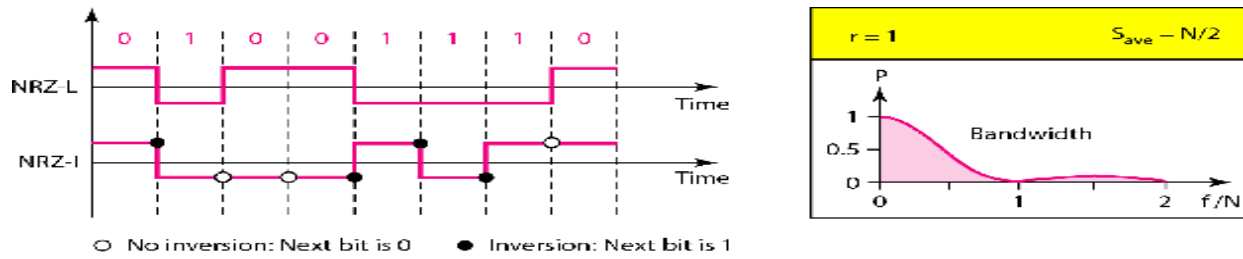


Fig 1.14 Polar Schemes

Biphase: Manchester and Differential Manchester: 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 as shown in Fig 1.15.

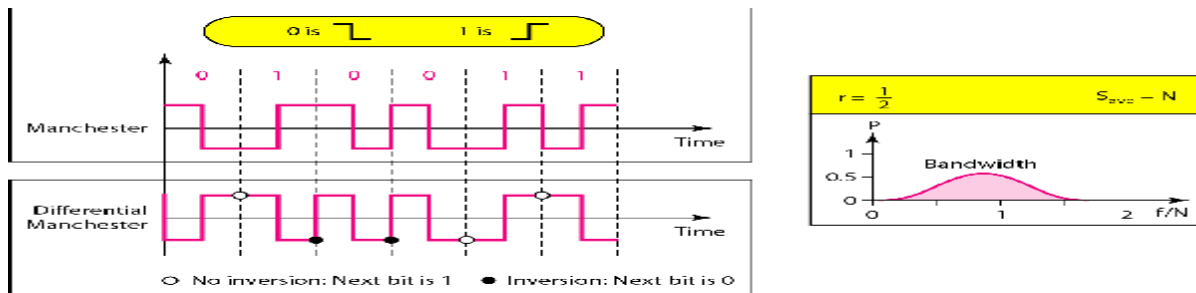


Fig 1.15 Polar Schemes

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 as shown in Fig 1.16.

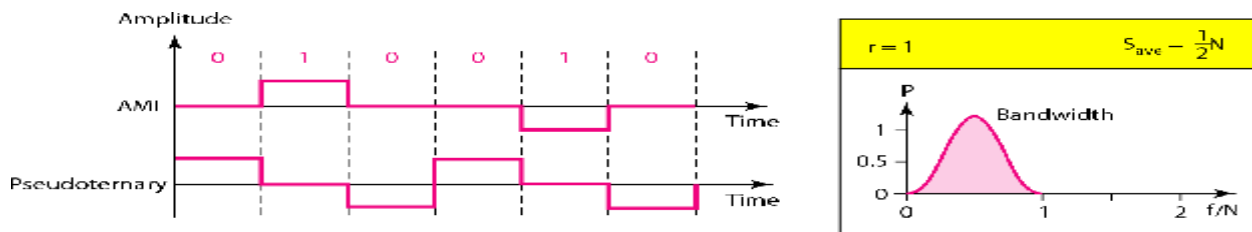


Fig 1.16 Polar Schemes

Multilevel Schemes: The desire to increase the data speed or decrease the required bandwidth has resulted in the creation of many schemes. The 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. We only have 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 as shown in Fig.1.17.

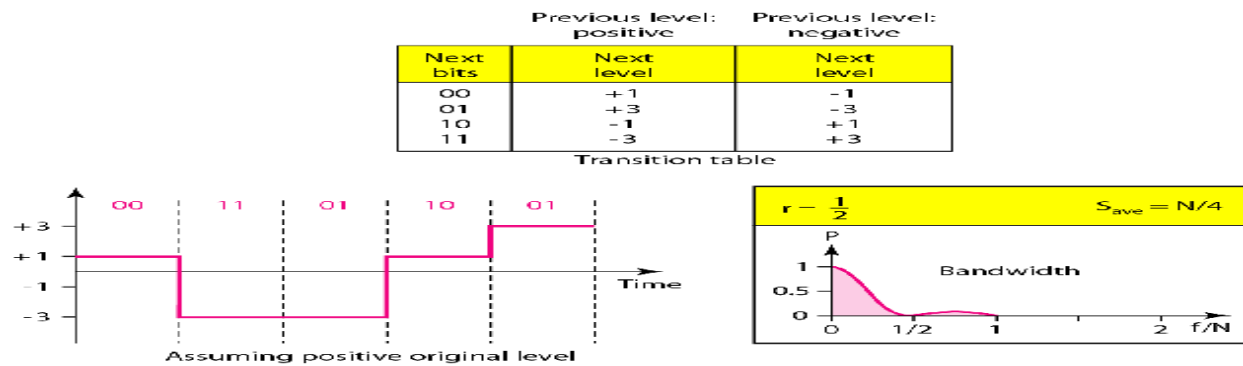


Fig 1.17 Polar Schemes

Summary of Line Coding Schemes have been shown in Table 1.1

Table 1.1 below shows the summary of Line Coding Schemes:

Category	Scheme	Bandwidth (average)	Characteristics
Unipolar	NRZ	$B = N/2$	Costly, no self-synchronization if long 0s or 1s, DC
Unipolar	NRZ-L	$B = N/2$	No self-synchronization if long 0s or 1s, DC
	NRZ-I	$B = N/2$	No self-synchronization for long 0s, DC
	Biphase	$B = N$	Self-synchronization, no DC, high bandwidth
Bipolar	AMI	$B = N/2$	No self-synchronization for long 0s, DC
Multilevel	2B1Q	$B = N/4$	No self-synchronization for long same double bits
	8B6T	$B = 3N/4$	Self-synchronization, no DC
	4D-PAM5	$B = N/8$	Self-synchronization, no DC
Multiline	MLT-3	$B = N/3$	No self-synchronization for long 0s

1.8 TRANSMISSION MEDIA

A transmission medium can be broadly defined as anything 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. The air can also be used to convey the message in a smoke signal or semaphore. 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. It can be classified as shown in Fig 1.18: Guided and Unguided as shown below:

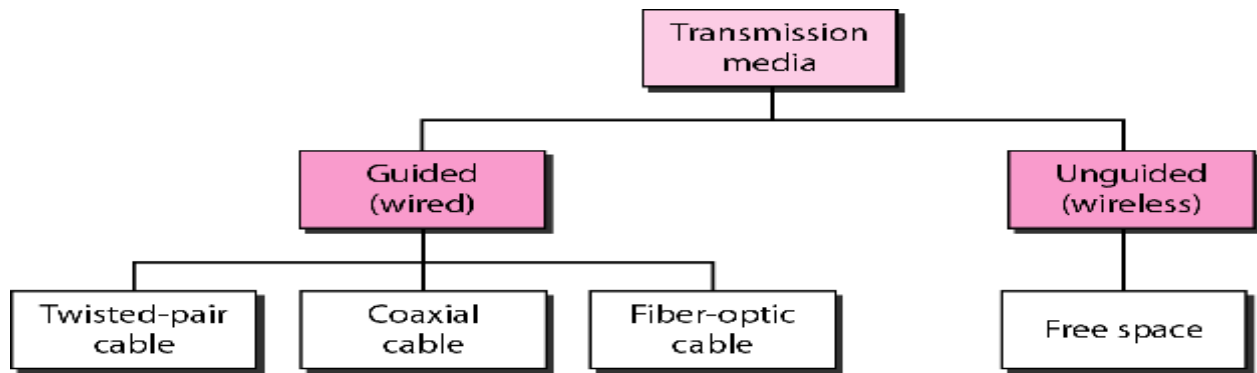


Fig 1.18 Transmission Media

1.8.1 Guided Media

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

Twisted Pair Cable: A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together. 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. Twisted-pair cables are used in telephone lines to provide voice and data channels. The local loop—the line that connects subscribers to the central telephone office commonly consists of unshielded twisted-pair cables. The DSL lines that are used by the telephone companies to provide high-data-rate connections also use the high-bandwidth capability of unshielded twisted-pair cables. Twisted pair Cable is shown below in Fig. 1.19:



Fig 1.19 Twisted Pair Cable

Coaxial Cable: Coaxial cable (or coax) as shown in Fig1.20 carries signals of higher frequency ranges than those in twisted pair cable, in part because the two media are constructed quite differently. Instead of having two wires, coax has a central core conductor of solid or stranded wire (usually copper) enclosed in an insulating sheath, which is, in turn, encased in an outer conductor of metal foil, braid, or a combination of the two. The outer metallic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit. This outer conductor is also enclosed in an insulating sheath, and the whole cable is protected by a plastic cover as shown in figure below:

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

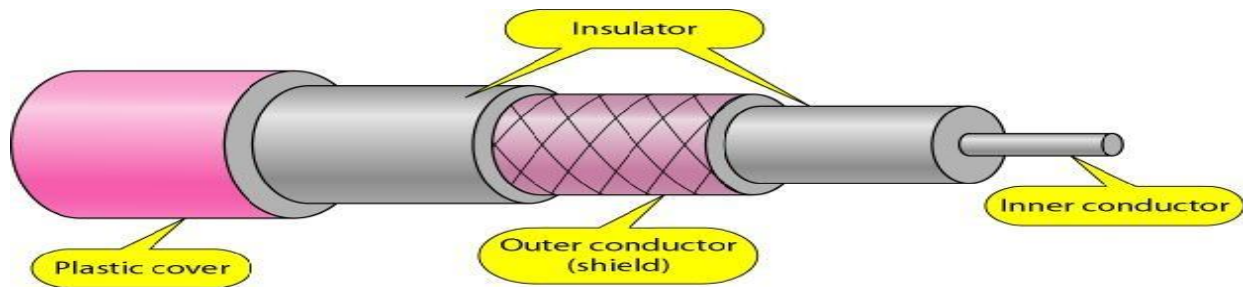


Fig 1.20 Coaxial Cable

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 as shown in Fig.1.21.

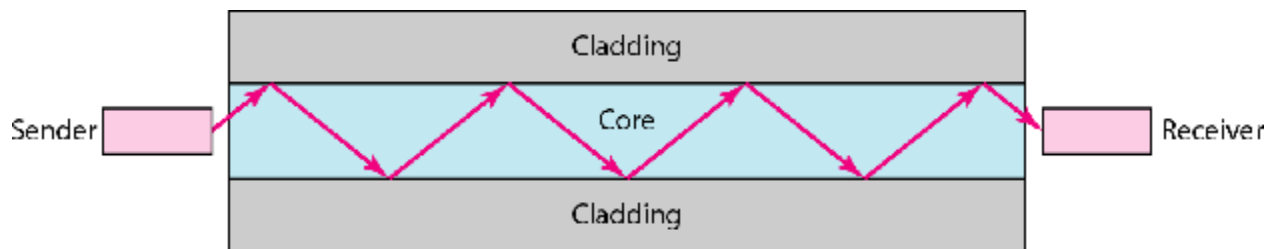


Fig 1.21 Fiber Optic Cable

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

1.8.2 Unguided Media

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. Unguided signals can travel from the source to destination in several ways: ground propagation, sky propagation, and line-of-sight propagation. Wireless transmission is of three types as shown below in Fig 1.22:

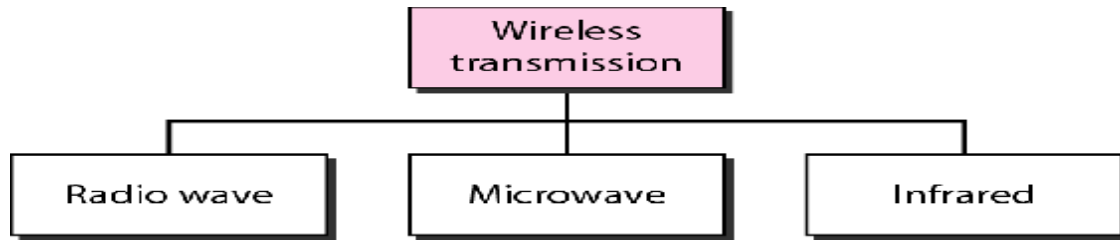


Fig 1.22 Wireless Transmission

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 use omni-directional antennas that send out signals in all directions. 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, cordless phones, and paging are examples of multicasting.

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. 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, satellites and wireless LANs.

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. 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.

UNIT II

2.1 SWITCHING

In large networks we need some means to allow one-to-one communication between any two nodes. In LANs this is achieved using one of three methods:

- Direct point-to-point connection (mesh)
- Via central controller (star)
- Connection to common bus in a multipoint configuration (bus/hub)

None of the previous works in larger networks with large physical separation or consisting of a large number of computers because it requires too much infrastructure and majority of those links would be idle for most of the time. Thus, better solution is a switching network. It consists of a series of interlinked nodes called switched. Switches are capable to create temporary connections between two or more devices. Some of these nodes are connected to the end system and others are used only for routing. End systems can be computers or telephones. Switching network has been shown in Figure 2.1

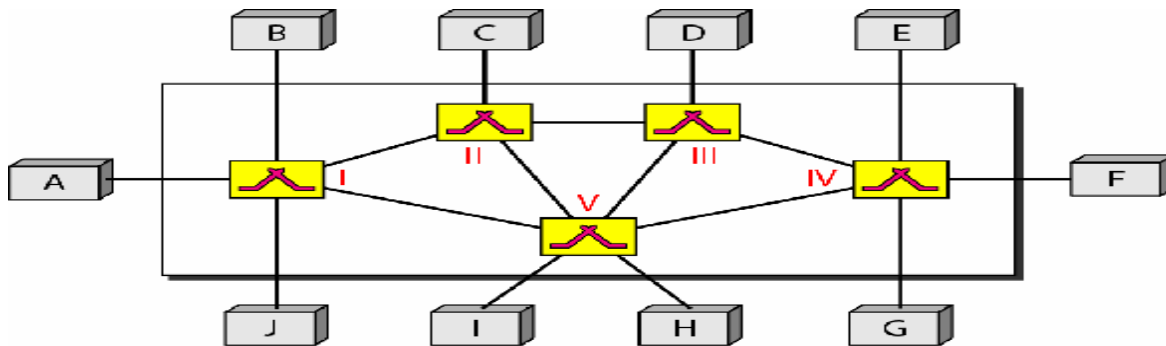


Fig 2.1 Switching Network

There are three types of Switched Network namely Circuit Switched Network, Packet Switched Network and Message Switched Network as shown in Fig 2.2

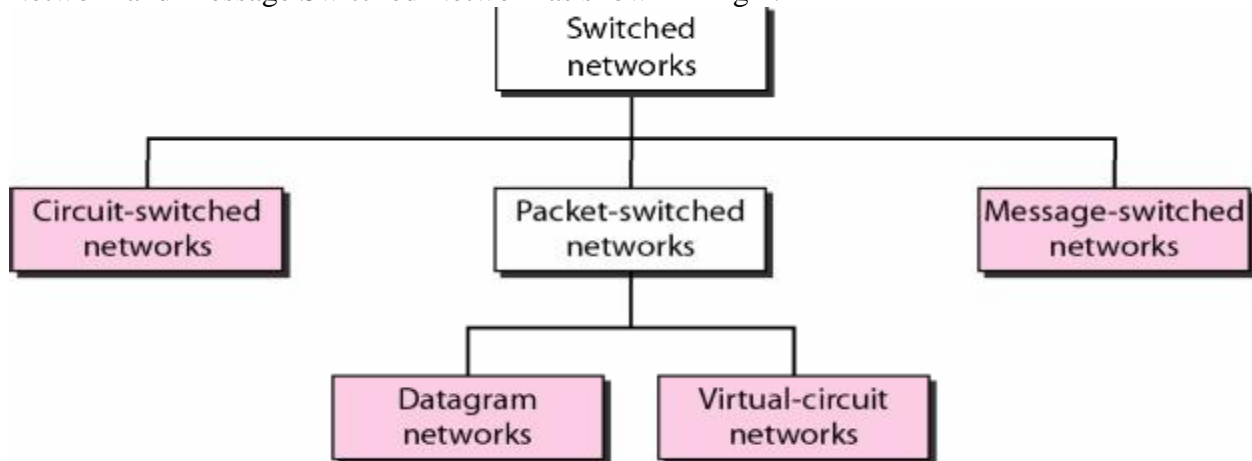


Fig 2.2 Types of Circuit Switching

A) CIRCUIT SWITCHING

A circuit-switched network (as shown in Fig 2.3) 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. Each connection uses only one dedicated channel on each link. Each link is normally divided into n channels by using FDM or TDM. The link can be permanent (leased line) or temporary (telephone).

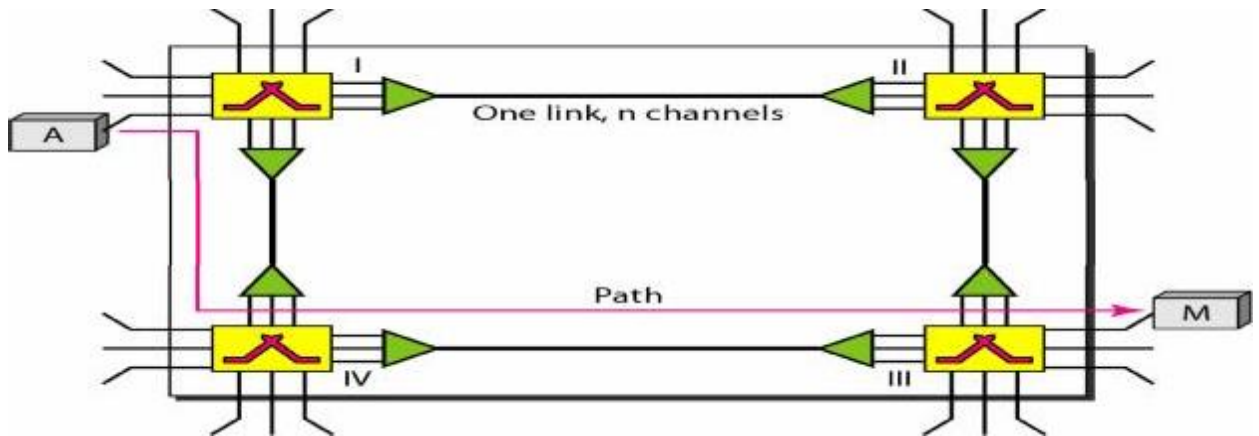


Fig 2.3 Circuit Switching

Switching takes place at physical layer. Resources can be bandwidth in FDM and time slot in TDM, switch buffer, switch processing time or switch I/O ports. Data transferred are not packetized, but it is a continuous flow. No addressing involved during data transfer. There are three transmission phases in circuit switching namely Setup phase, data transfer phase and tear down phase.

It can be argued that circuit-switched networks are not as efficient as the other two types of networks 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. The total delay in this circuit switching is shown in Fig 2.4

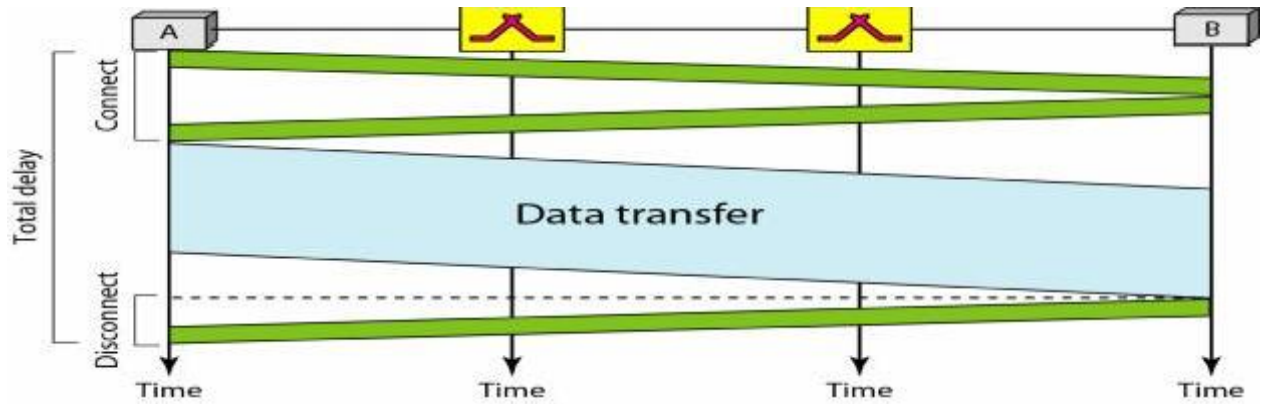


Fig 2.4 Delay in Circuit Switching

B) PACKET SWITCHING

In packet Switching, flow of data is not continuous rather it flows in the form of packets. The size of the packet is determined by the network and the governing protocol. This type of switching further classify into datagram networks and virtual circuit networks.

2.1.1 Datagram Networks

Data are transmitted in discrete units called packets. Size of the packet depends on the protocol and network. Packets switched networks are connectionless, hence no resource allocation. Connectionless means the switch does not keep information about the connection state. Datagram switching is done at network layer as shown in Fig 2.5

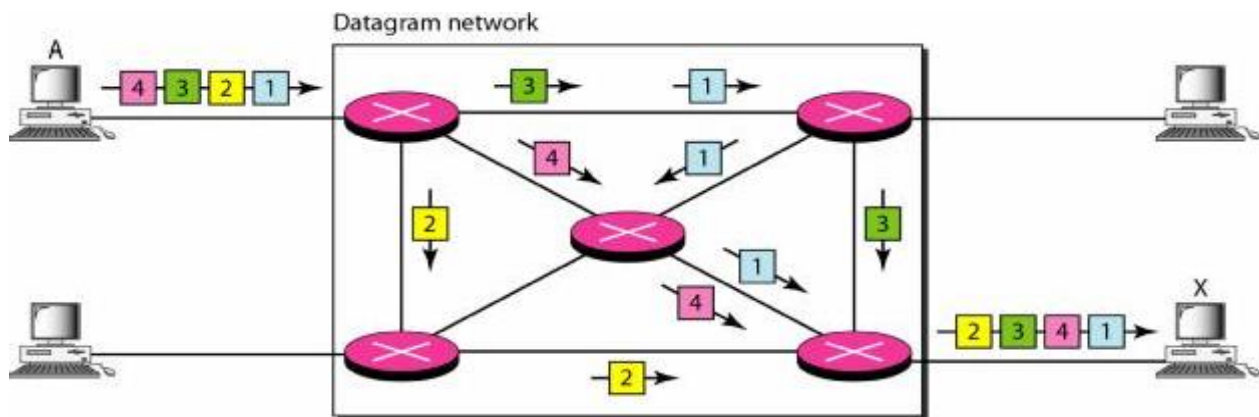


Fig 2.5 Datagram Networks

A switch in a datagram network uses a routing table that is based on the destination address. The destination address in the header of a packet in a datagram network remains the same during the entire journey of the packet. The total delay is shown in Fig 2.6

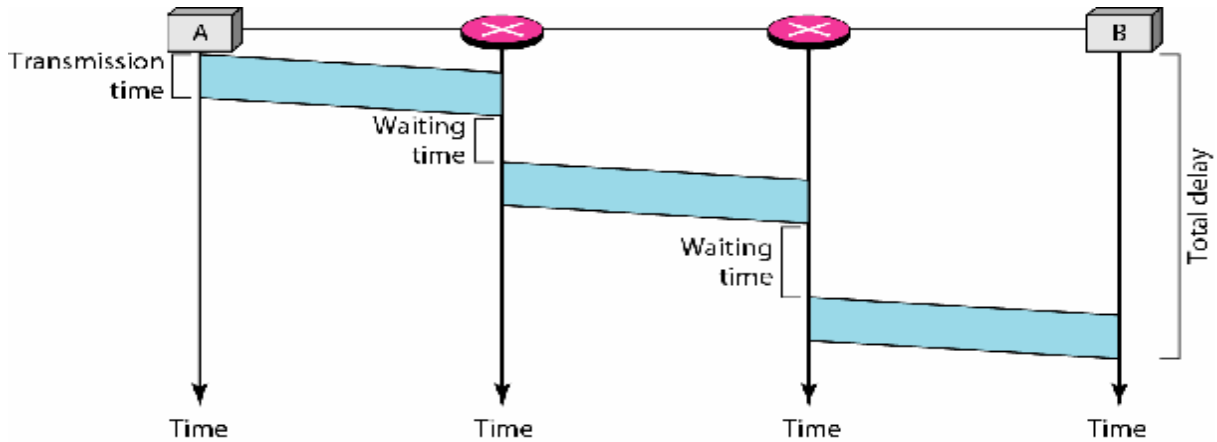


Fig 2.6 Delay in Datagram Networks

2.1.2 Virtual Circuit Networks

A virtual-circuit network is a cross between a circuit-switched network and a datagram network. The virtual-circuit shares characteristics of both. Packets form a single message travel along the same path. Following are the characteristics of virtual circuit networks:

- Three phases to transfer data
- Resources can be allocated during setup phase
- Data are packetized and each packet carries an address in the header
- All packets follow the same path
- Implemented in data link layer

A virtual-circuit network uses a series of special temporary addresses known as virtual circuit identifiers (VCI). The VCI at each switch is used to advance the frame towards its final destination. The switch has a table with 4 columns as shown in Fig 2.7 i.e. Inputs half: Input Port Number and Input VCI and Outputs half: Output Port Number and Output VCI.

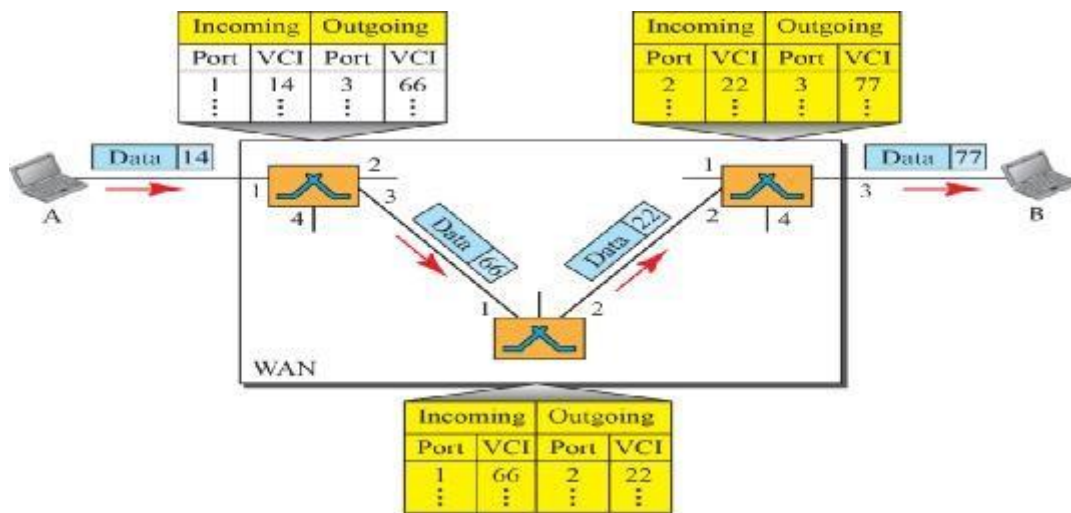


Fig 2.7 Virtual Circuit Identifier

The VCN behaves like a circuit switched net because there is a setup phase to establish the VCI entries in the switch table. There is also a data transfer phase and teardown phase. The total delay in virtual circuit network is shown in Fig 2.8

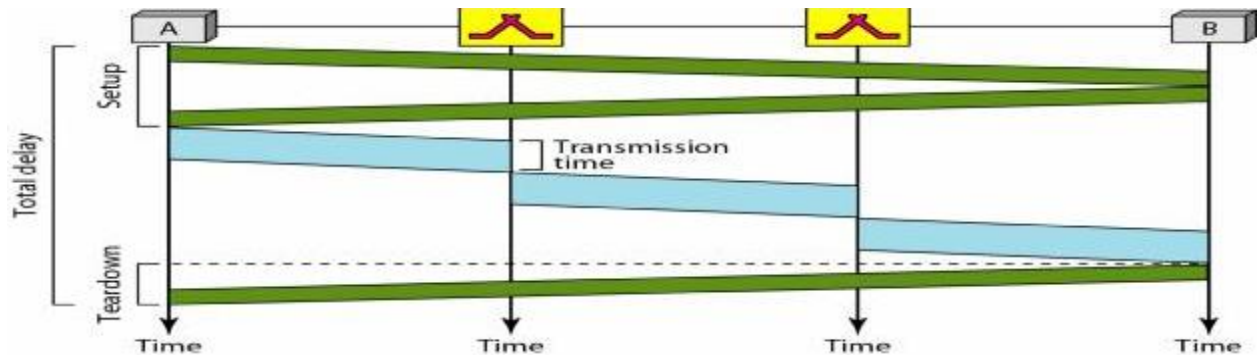


Fig 2.8 Delay in Virtual Circuit Identifier

2.1.3 Structure of a Switch

We use switches in circuit-switched and packet-switched networks. There are two structures of a switch named as space division switch and time division switch.

In space-division switching as shown in Fig 2.9, 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.

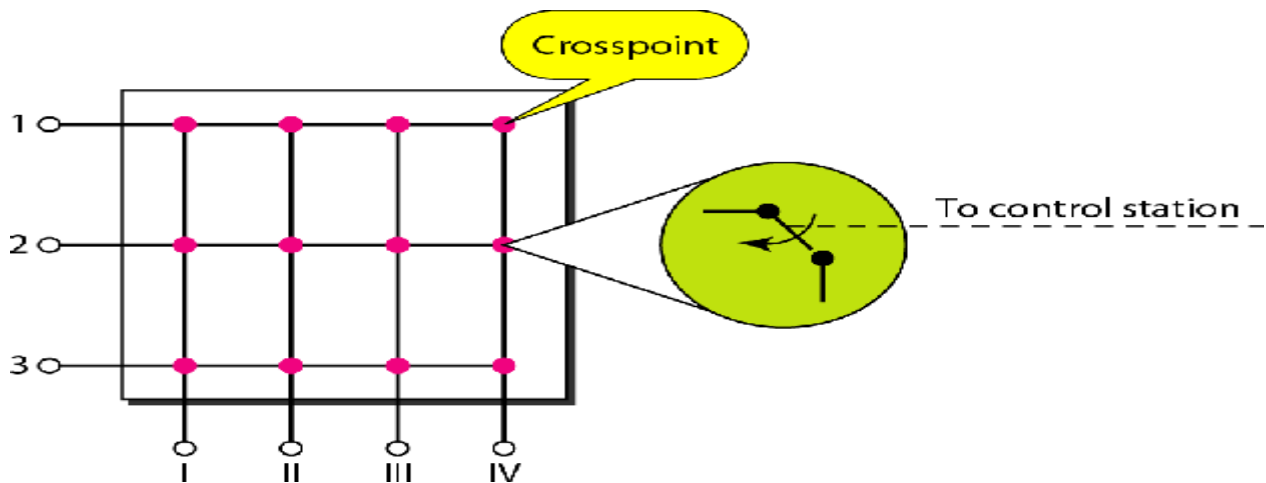


Fig 2.9 Space Division Switching: Crossbar Switch

Time Division Switching as shown in Fig 2.10 consists of a TDM Multiplexer, a TDM Demultiplexer, 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.

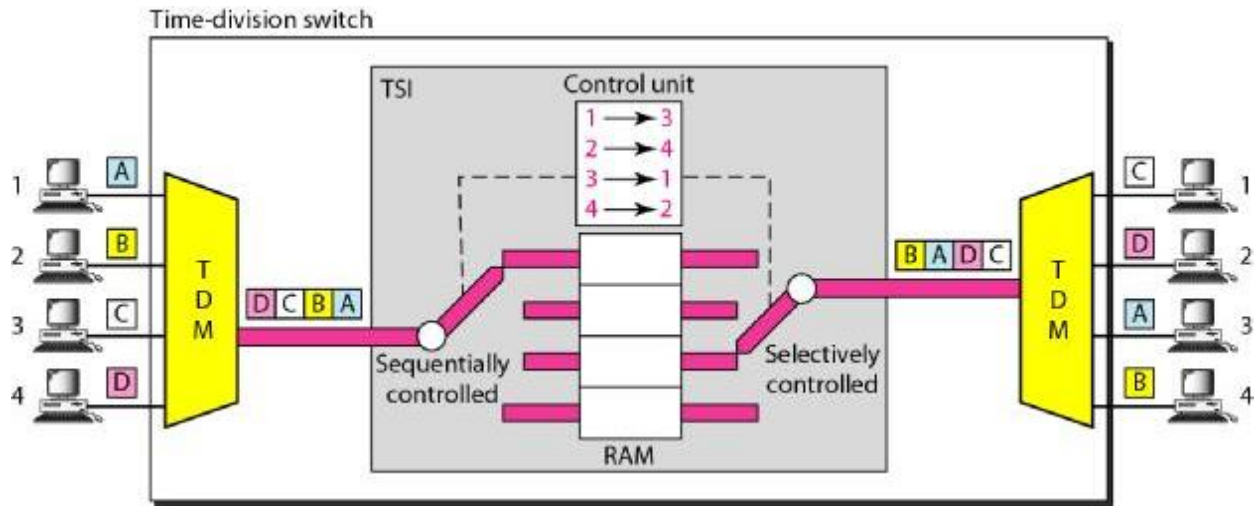


Fig 2.10 Time Division Switching

2.1.4 Ethernet Physical Layer

Telephone networks use circuit switching. The telephone network had its beginnings in the late 1800s. The entire network, which is referred to as the plain old telephone system (POTS), was originally an analog system using analog signals to transmit voice. There are three major components of telephone system namely local loops, trunks and switching offices as shown in Fig 2.11.

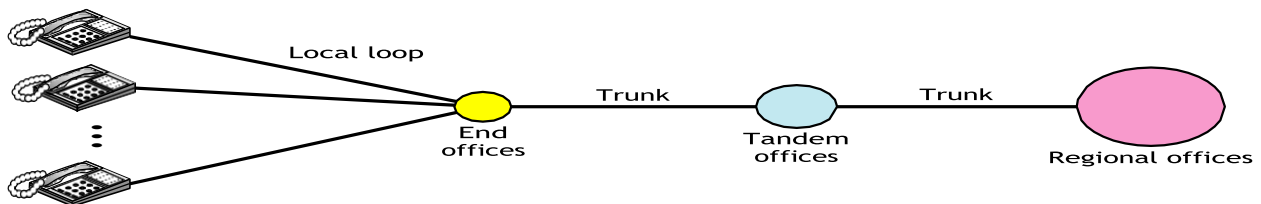


Fig 2.11 Telephone Network

Local Loops: One component of the telephone network is the local loop, a twisted-pair cable that connects the subscriber telephone to the nearest end office or local central office. **Trunks:** Trunks are transmission media that handle the communication between offices. A trunk normally handles hundreds or thousands of connections through multiplexing. **Switching Offices:** To avoid having a permanent physical link between any two subscribers, the telephone company has switches located in a switching office. A switch connects several local loops or trunks and allows a connection between different subscribers. The telephone network has several levels of switching offices such as end offices, tandem offices, and regional offices.

Signaling can be defined as the information exchange concerning the establishment and control of a telecommunication circuit and the management of the network. There are two types: In-Band and Out-Band. In in-band signaling, the same circuit can be used for both signaling and voice communication. In out-of-band signaling, a portion of the voice channel bandwidth was used for signaling; the voice bandwidth and the signaling bandwidth were separate. The signaling system was required to perform other tasks such as: providing dial tone, ring tone, and busy tone, transferring telephone numbers between offices, and providing other functions such as caller ID, voice mail etc. These complex tasks resulted in the provision of a separate network for signaling. This means that a telephone network today can be thought of as two networks: a signaling network and a data transfer network. The protocol that is used in signaling network is SS7 (Signaling System Seven) as shown in Fig 2.12.

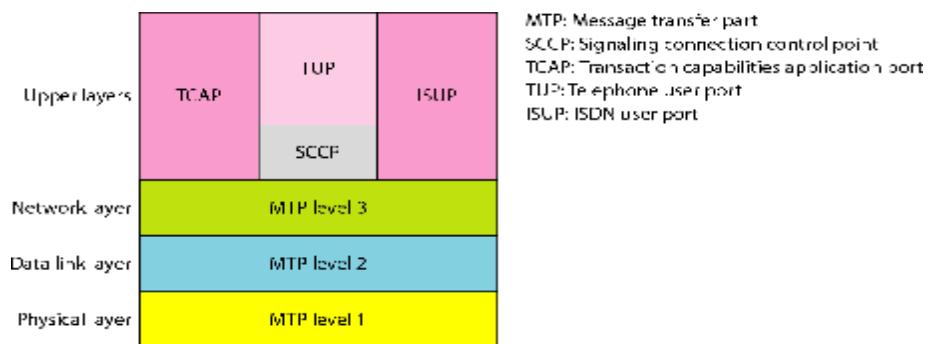


Fig 2.12 SS7 (Signaling System Seven)

2.2 DATA LINK LAYER

Data can be corrupted during transmission. Some applications require that errors be detected and corrected. Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. Thus, we say that error had occurred. There are two types of error: single-bit error and burst error. In a single-bit error, only 1 bit in the data unit has changed whereas, in burst error means that 2 or more bits in the data unit have changed as shown in Fig 2.13.

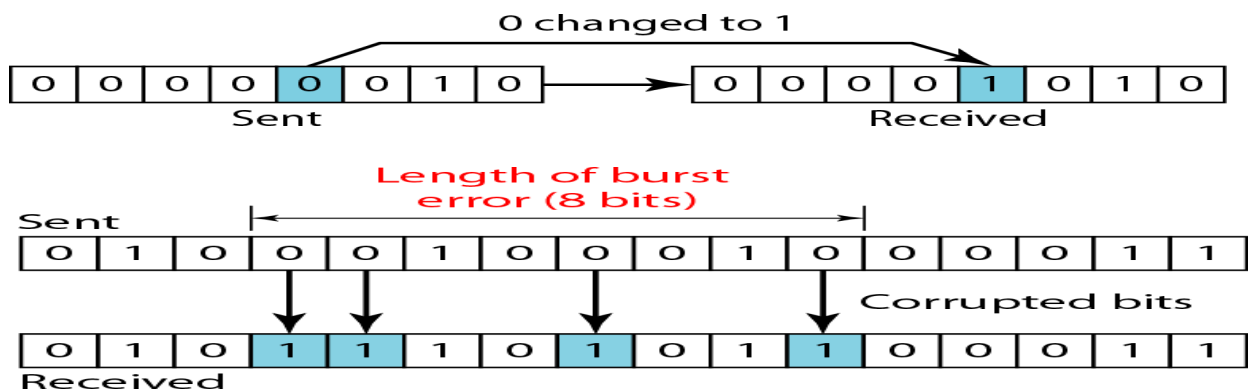


Fig 2.13 Single-bit and Burst Error

The central concept in detecting or correcting errors is redundancy. To be able to detect or correct errors, we need to send some extra bits with our data. These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits. 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. Coding schemes into two broad categories: block coding and convolution coding.

Block Coding: In block coding, we divide our message into blocks, each of k bits, called datawords. We add r redundant bits to each block to make the length $n = k + r$. The resulting n -bit blocks are called codewords as shown in Fig 2.14.

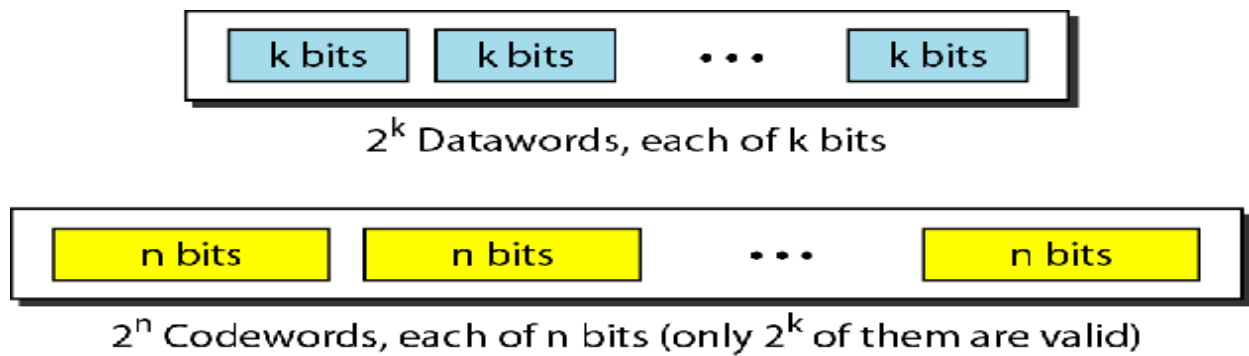


Fig 2.14 Datawords and Codewords

2.2.1 Error Detection

An error-detecting code can detect only the types of errors for which it is designed; other types of errors may remain undetected. Fig 2.15 shows the process of error detection in block coding.

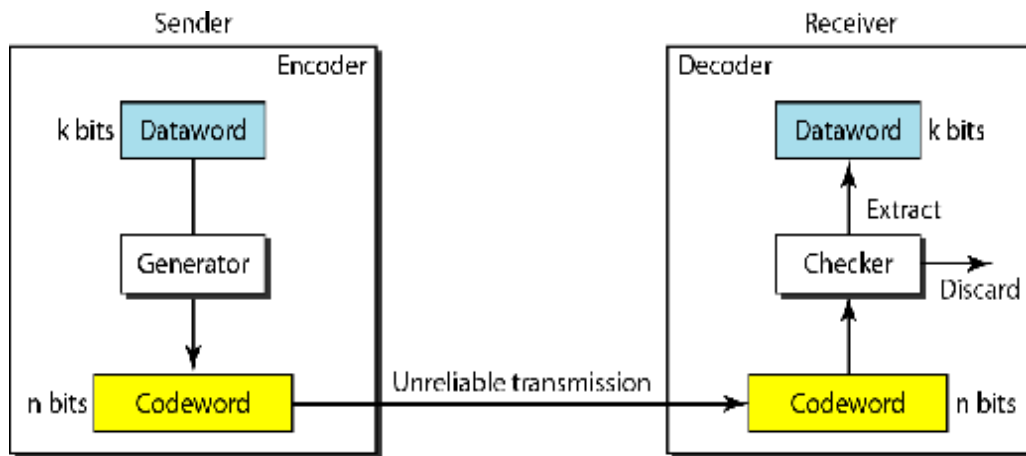


Fig 2.15 Process of Error Detection

2.2.2 Error Correction

Once an error has been detected, it has to be corrected. Fig 2.16 shows the process of error correction.

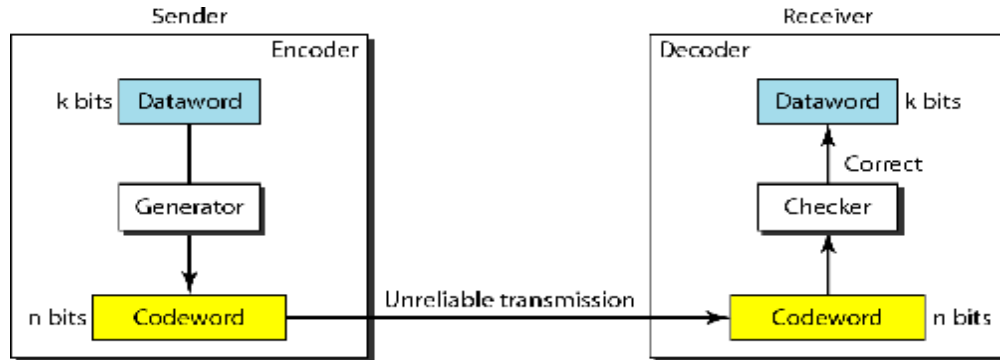


Fig 2.16 Process of Error Correction

One of the central concepts in coding for error control is the idea of the Hamming Distance. 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 $d_{\min} = s + 1$.
- To guarantee correction of up to t errors in all cases, the minimum Hamming distance in a block code must be $d_{\min} = 2t + 1$.

Linear Block Codes: Almost all block codes used today belong to a subset called linear block codes. A linear block code is a code in which the exclusive OR (addition modulo-2) of two valid codewords creates another valid codeword. A single parity-check code is of linear block code. 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 as shown in Fig 2.17.

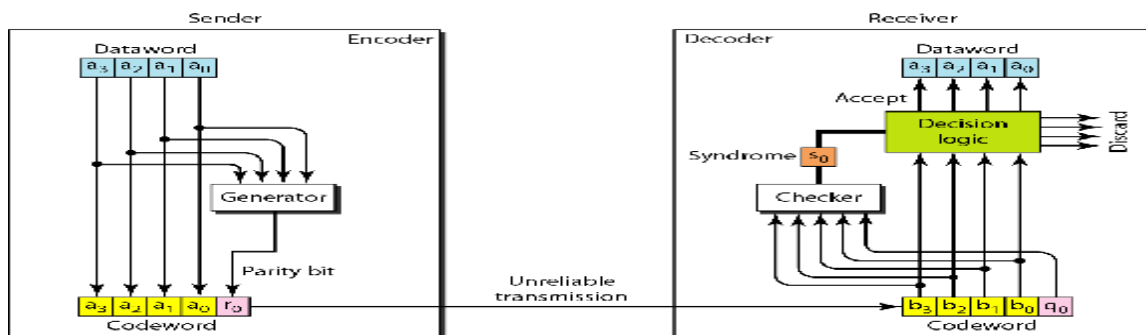


Fig 2.17 Single Parity Check Code

The syndrome is 0 when the number of 1s in the received codeword is even; otherwise, it is 1. If the syndrome is 0, there is no error in the received codeword; the data portion of the received codeword is accepted as the dataword and if the syndrome is 1, the data portion of the received codeword is discarded.

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 as shown in Fig 2.18.

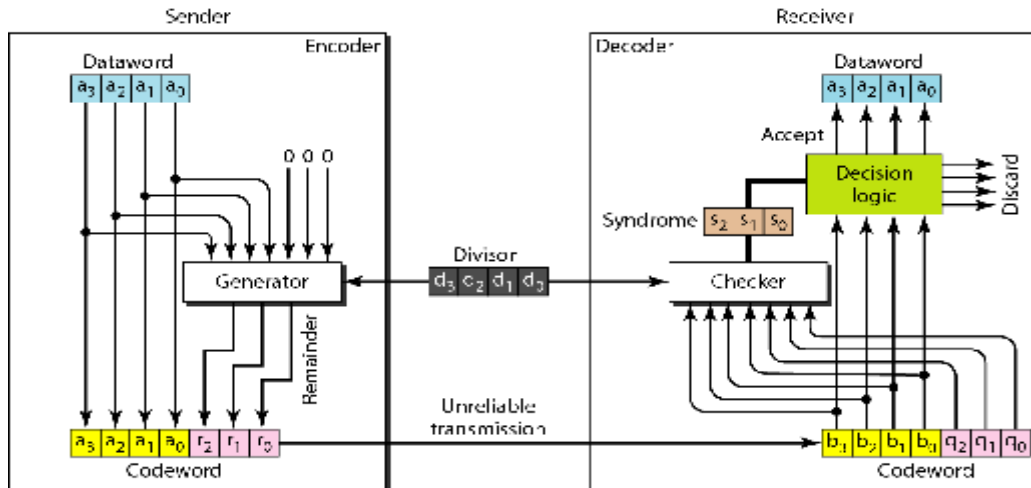


Fig 2.18 Cyclic Code

In cyclic code, concept of long division has been used. The divisor in a cyclic code is normally called the generator polynomial or simply the generator as shown in Fig 2.19.

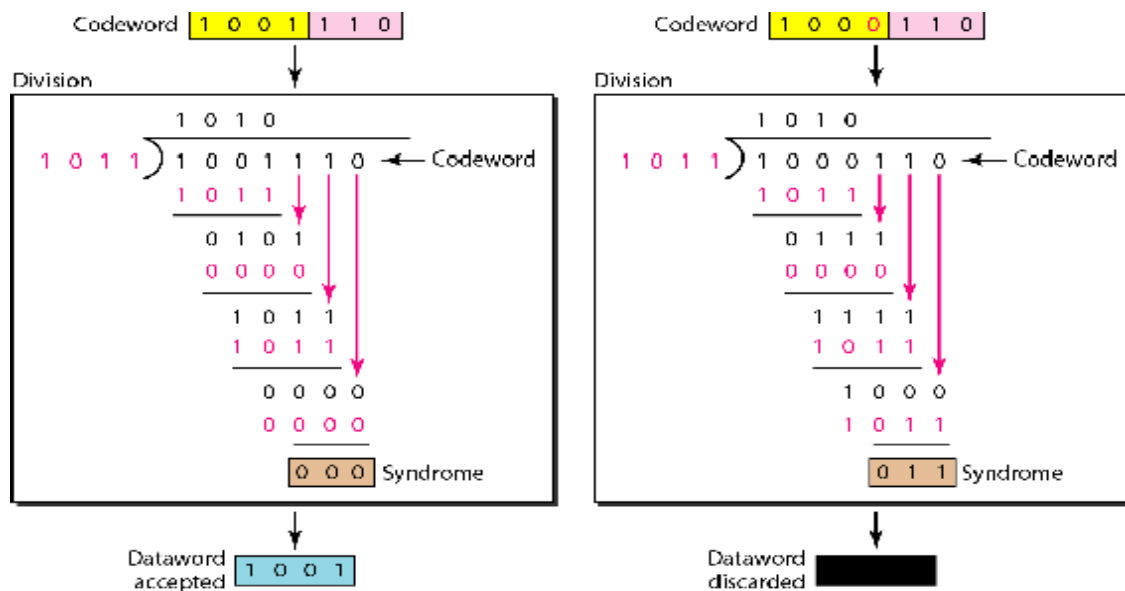


Fig 2.19 Division in Cyclic Code

In a cyclic code, following cases exist:

- If syndrome $s(x) \neq 0$, one or more bits is corrupted.
- If syndrome $s(x) = 0$, either
 - No bit is corrupted
 - Some bits are corrupted, but the decoder failed to detect them.

2.3 DATA LINK CONTROL

The data link layer needs to pack bits into frames, so that each frame is distinguishable from another.

2.3.1 Framing

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. There are two types of framing namely fixed size framing and variable size framing. In fixed size framing, we send fixed size of frames so there is no need to specify the boundaries of the frame. In variable size framing, size of frame is not fixed rather it is variable so we need to define boundaries of frame i.e ending of one frame and beginning of next frame. Thus, we use two techniques namely character oriented and bit oriented. In character oriented, we use the concept of byte stuffing as shown in Fig. 2.20. Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.

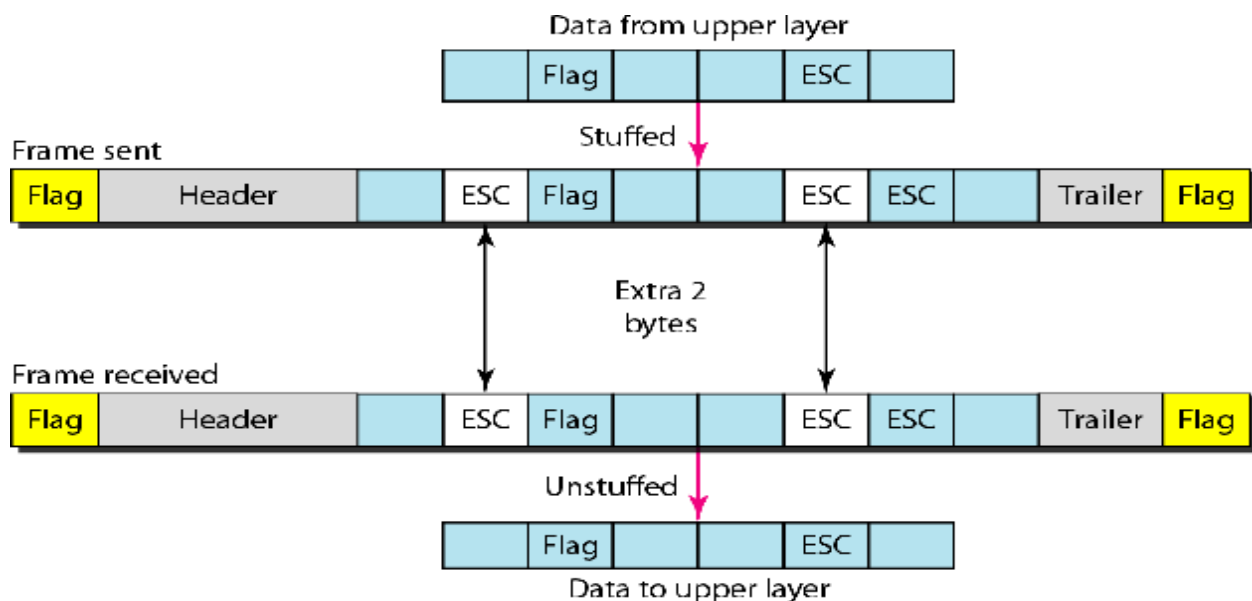


Fig 2.20 Byte stuffing and unstuffing

In bit oriented, we use bit stuffing as shown in Fig 2.21. 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 011110 for a flag.

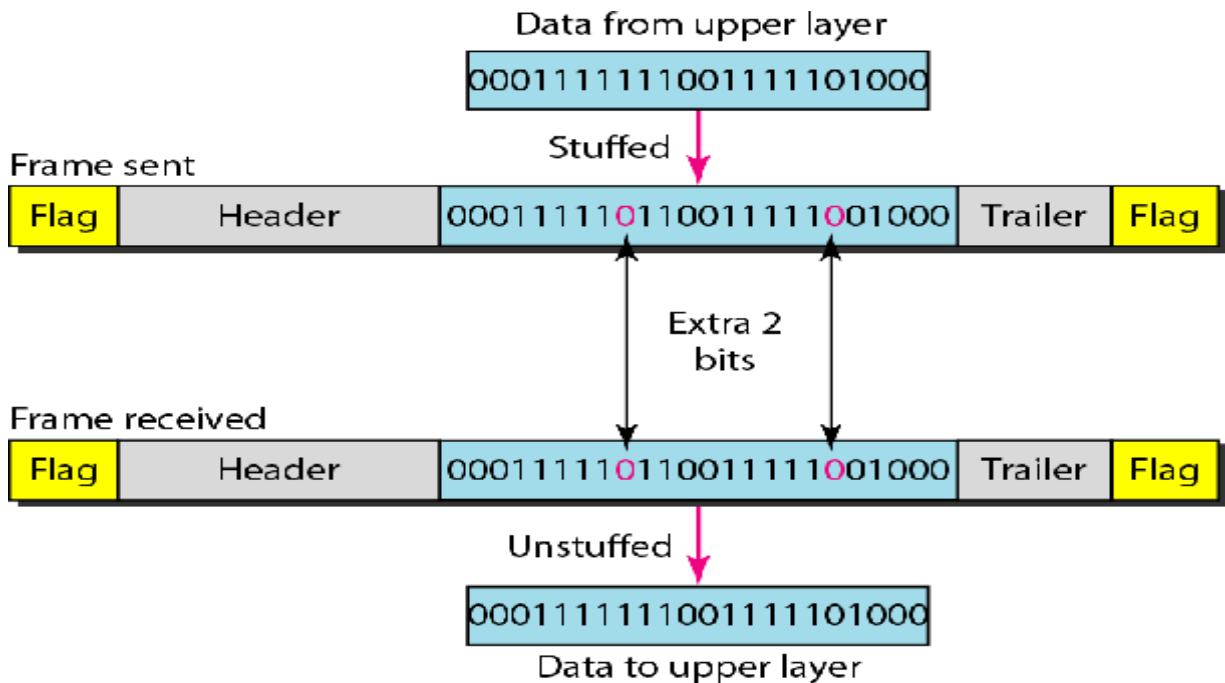


Fig 2.21 Bit stuffing and unstuffing

2.3.2 Flow

The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link 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.

2.3.3 Error Control Protocol

Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.

The data link layer can 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 broadly classified into two categories as shown in Fig.2.22: Noiseless channel and Noisy Channel.

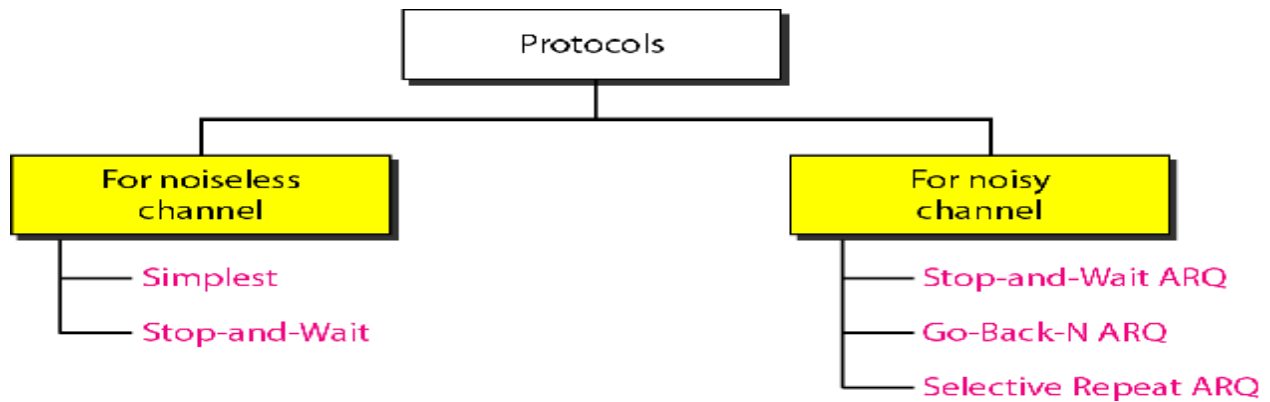


Fig 2.22 Type of Protocols

There is a difference between the protocols in real networks. All the protocols we discuss are unidirectional in the sense that the data frames travel from one node, called the sender, to another node, called the receiver. Although special frames, called acknowledgment (ACK) and negative acknowledgment (NAK) can flow in the opposite direction for flow and error control purposes, data flow in only one direction. In a real-life network, the data link protocols are implemented as bidirectional; data flow in both directions. 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.

2.4 NOISELESS CHANNEL AND NOISY CONTROL PROTOCOL

Noiseless protocols take channel as an ideal one in which no frames are lost, duplicated, or corrupted. There are two types of protocols used for noiseless channels namely simplest and stop & wait protocol. The first is a protocol that does not use flow control; the second is the one that does. Of course, neither has error control because we have assumed that the channel is a perfect noiseless channel.

Simplest protocol: is one that has no flow or error control. In simplest protocol, data frames are sent continuously, there is no concept of acknowledgement as shown in Fig 2.23. The data link layer at the sender site gets data from its network layer, makes a frame out of the data, and sends it. The data link layer at the receiver site receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer.

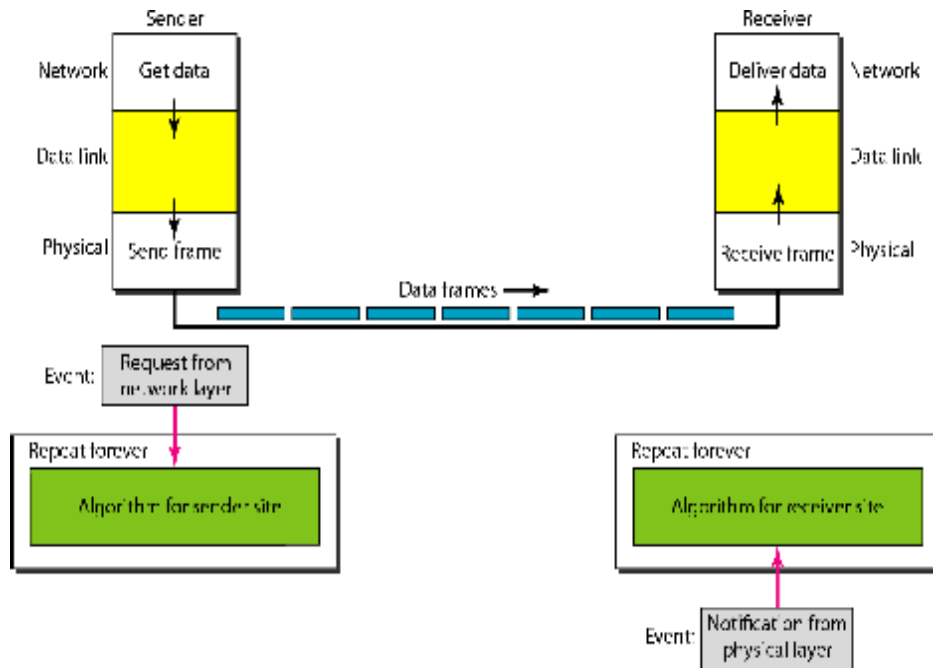


Fig 2.23 The design of the simplest protocol with no flow or error control

Flow diagram has been shown in Fig 2.24. The sender sends a sequence of frames without even thinking about the receiver. To send three frames, three events occur at the sender site and three events at the receiver site.

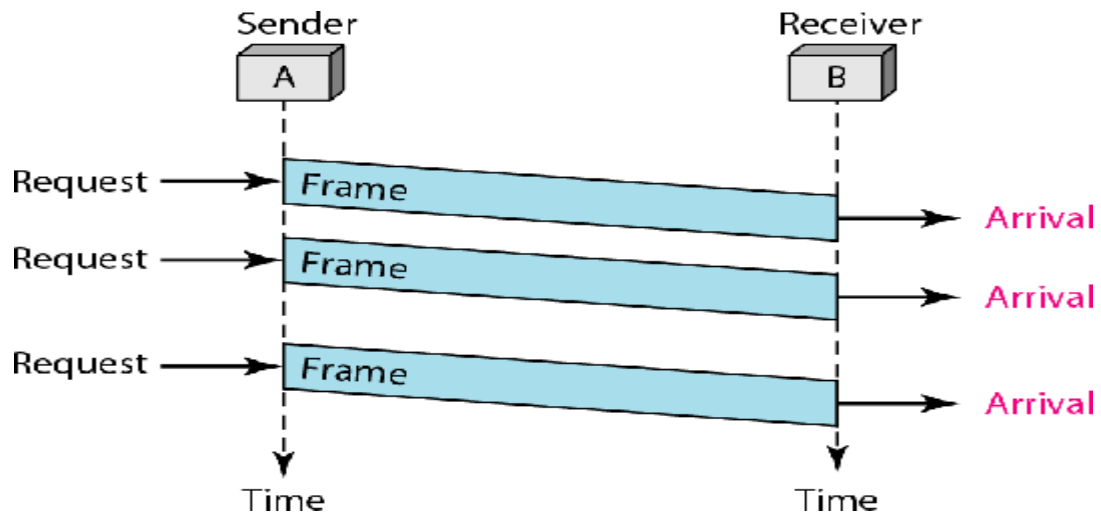


Fig 2.24 Flow diagram of Simplest Protocol

Stop & 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. Thus, in stop and wait protocol, one frame is sent at a time and it waits for acknowledgement of that frame. Once it receives acknowledgement, sender sends another frame as shown in Fig. 2.25.

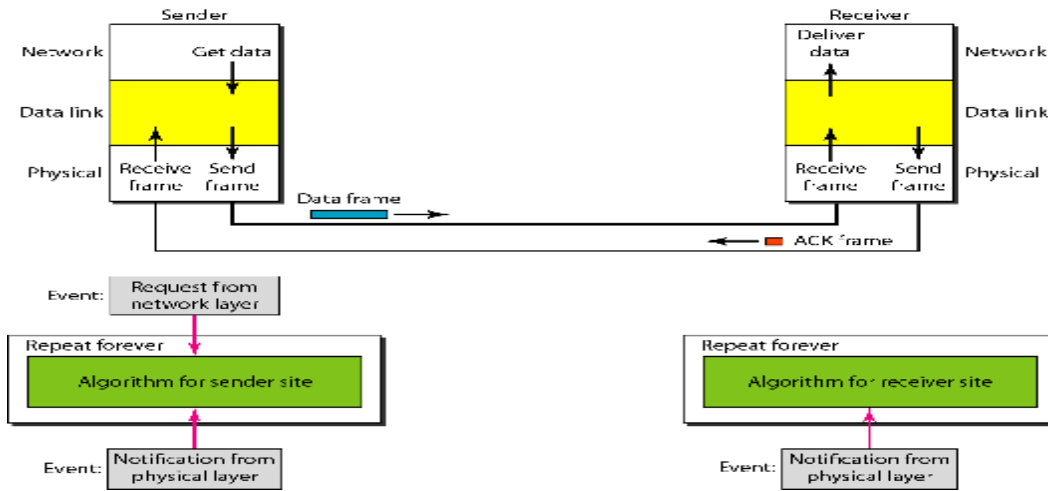


Fig 2.25 Design of Stop-and-Wait Protocol

Flow diagram has been shown in Fig 2.26. 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.

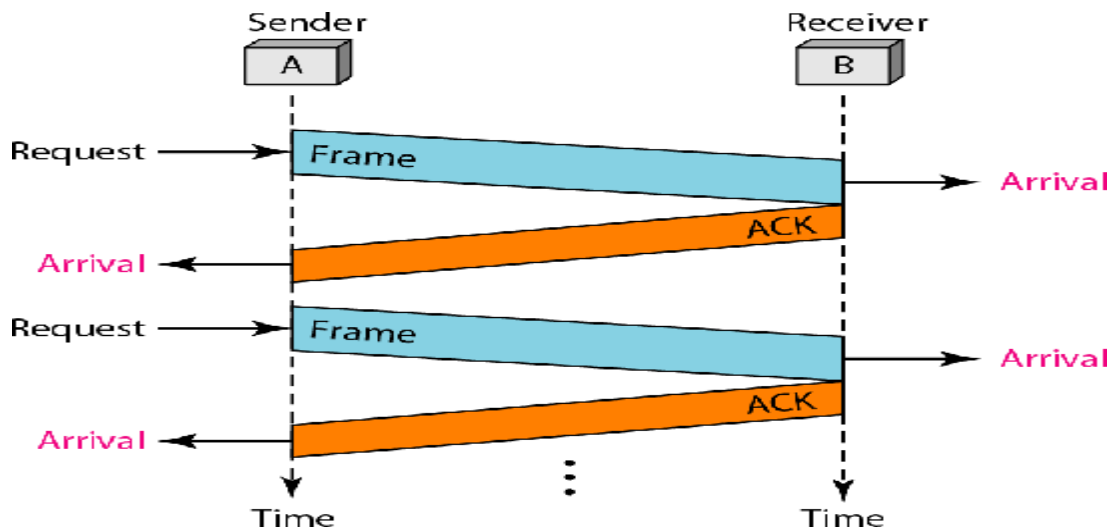


Fig 2.26 Flow diagram of Simplest Protocol

Noisy Channel Protocol

Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We can ignore the error or we need to add error control to our protocols. There are three protocols used in case of noisy channels namely: stop &

wait automatic repeat request, go-back- n automatic repeat request and selective automatic repeat request.

Stop & Wait Automatic Repeat Request (ARQ): 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 as shown in Fig. 2.27. 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. The completed and lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend? 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 the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK even though several copies of the same frame can be in the network.

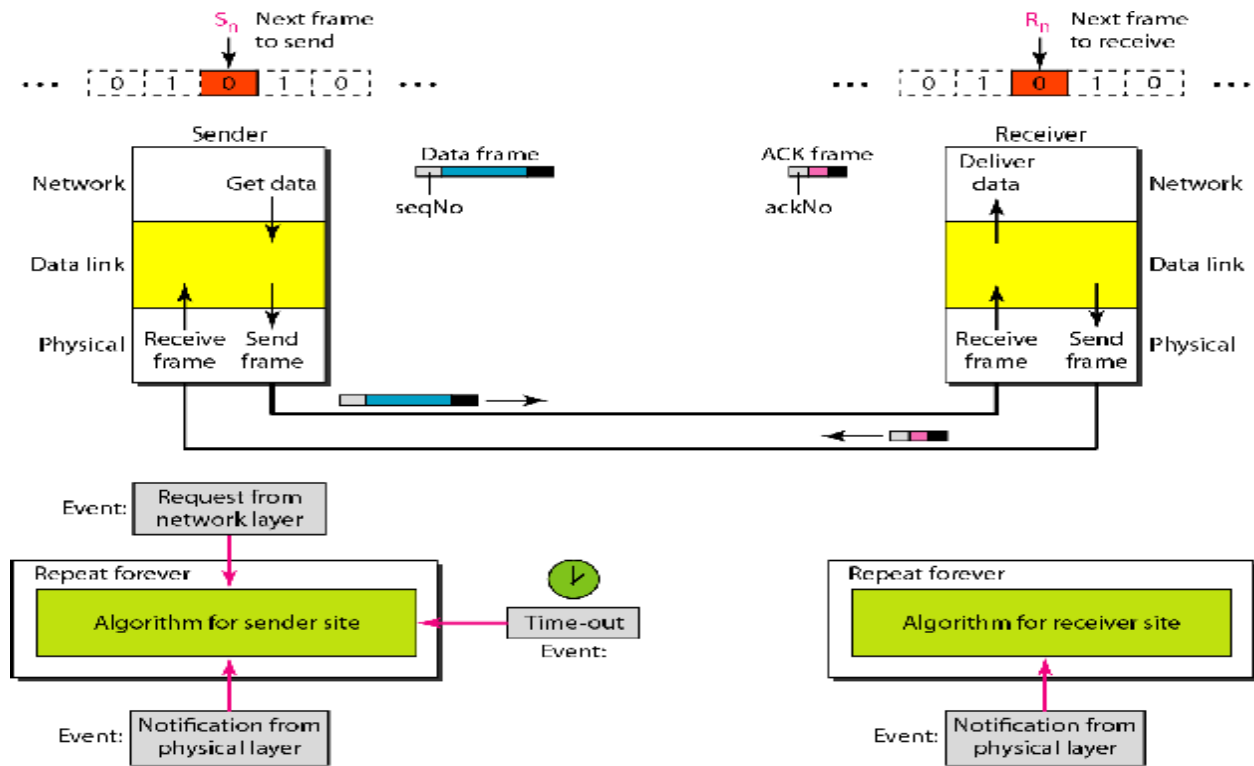


Fig 2.27 Stop & Wait Automatic Repeat Request

Go-Back-N Automatic Repeat Request: To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment. In other words, we need to let more than one frame be outstanding to keep the channel busy while the sender is waiting for acknowledgment. One protocol that can achieve this goal is called Go-Back-N Automatic Repeat Request as shown in Fig. 2.28. In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive. In the Go-Back-N Protocol, the sequence numbers are modulo 2^m , where m is the size of the sequence number field in bits.

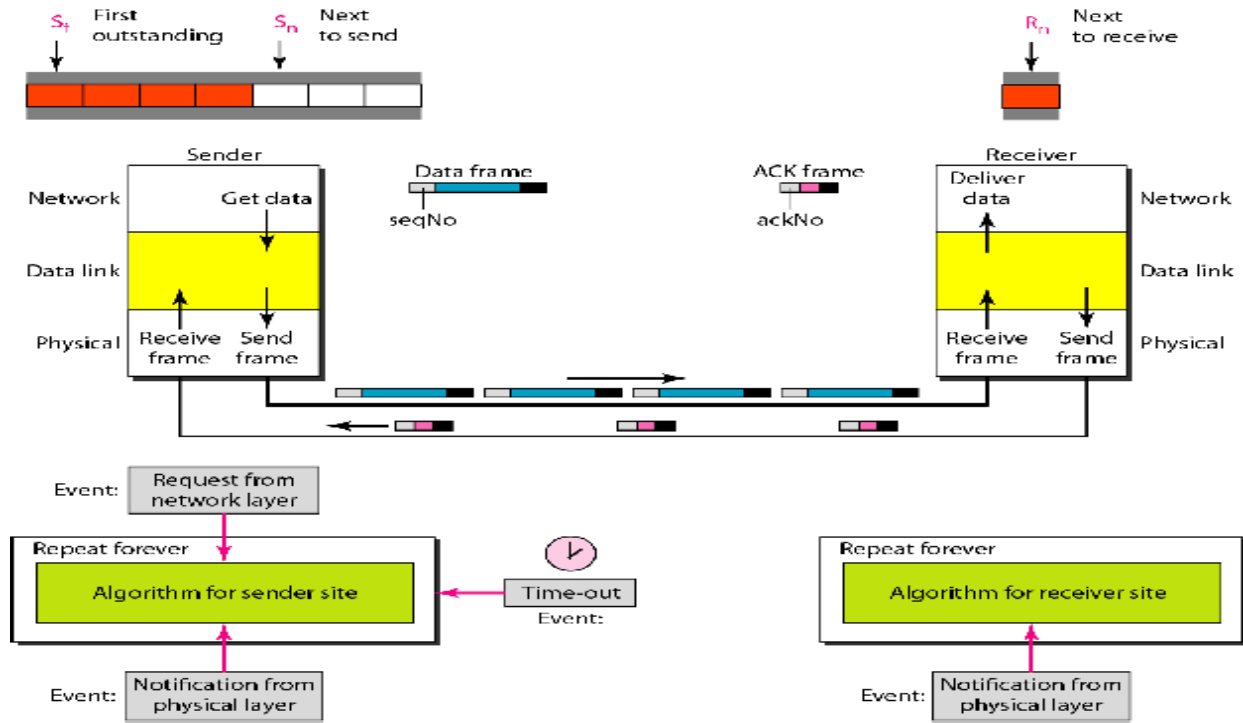


Fig 2.28 Go-Back-n Automatic Repeat Request

This protocol makes use of sliding window at sender and at receiver site. 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} . The send window can slide one or more slots when a valid acknowledgment arrives as shown in Fig.2.29.

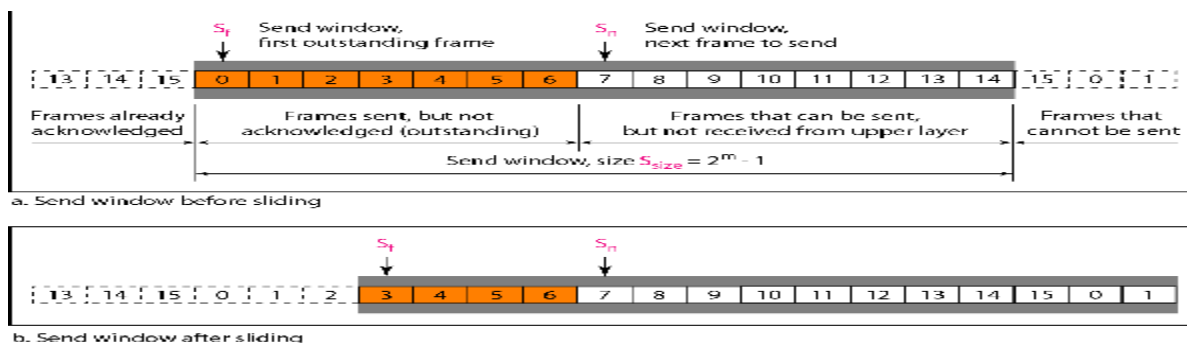


Fig 2.29 Send window (a) before and (b) after sliding

The receive window is an abstract concept defining an imaginary box of size 1 with one single variable R_n . The window slides when a correct frame has arrived; sliding occurs one slot at a time as shown in Fig. 2.30.

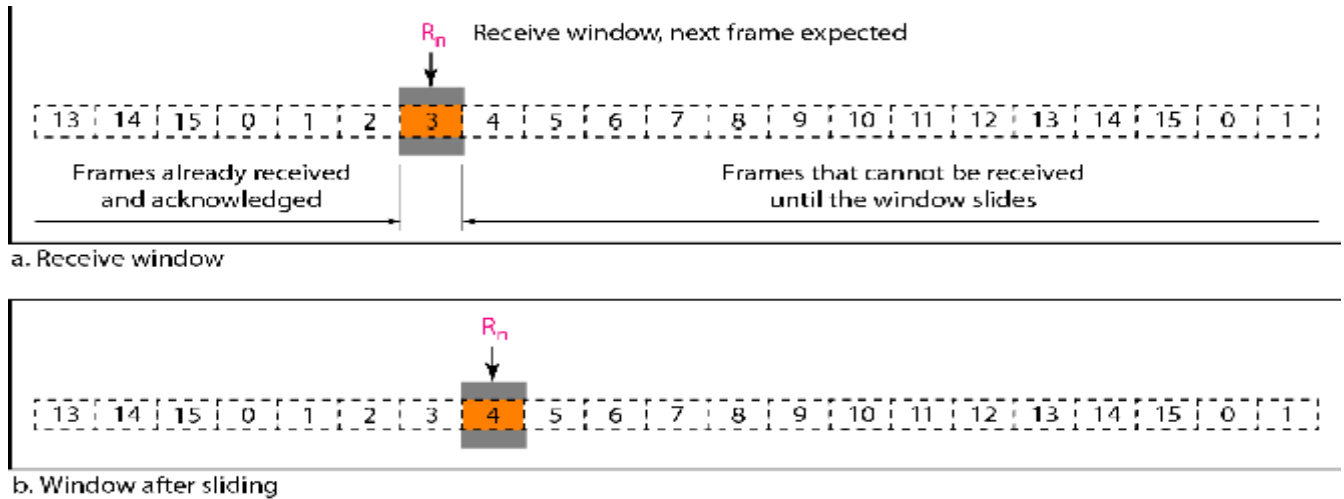


Fig 2.30 Receive window (a) before and (b) after sliding

Selective Repeat Automatic Repeat Request: Go-back-n ARQ simplifies the process at the receiver site. The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link. In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission. For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective Repeat ARQ as shown in Fig. 2.31. It is more efficient for noisy links, but the processing at the receiver is more complex.

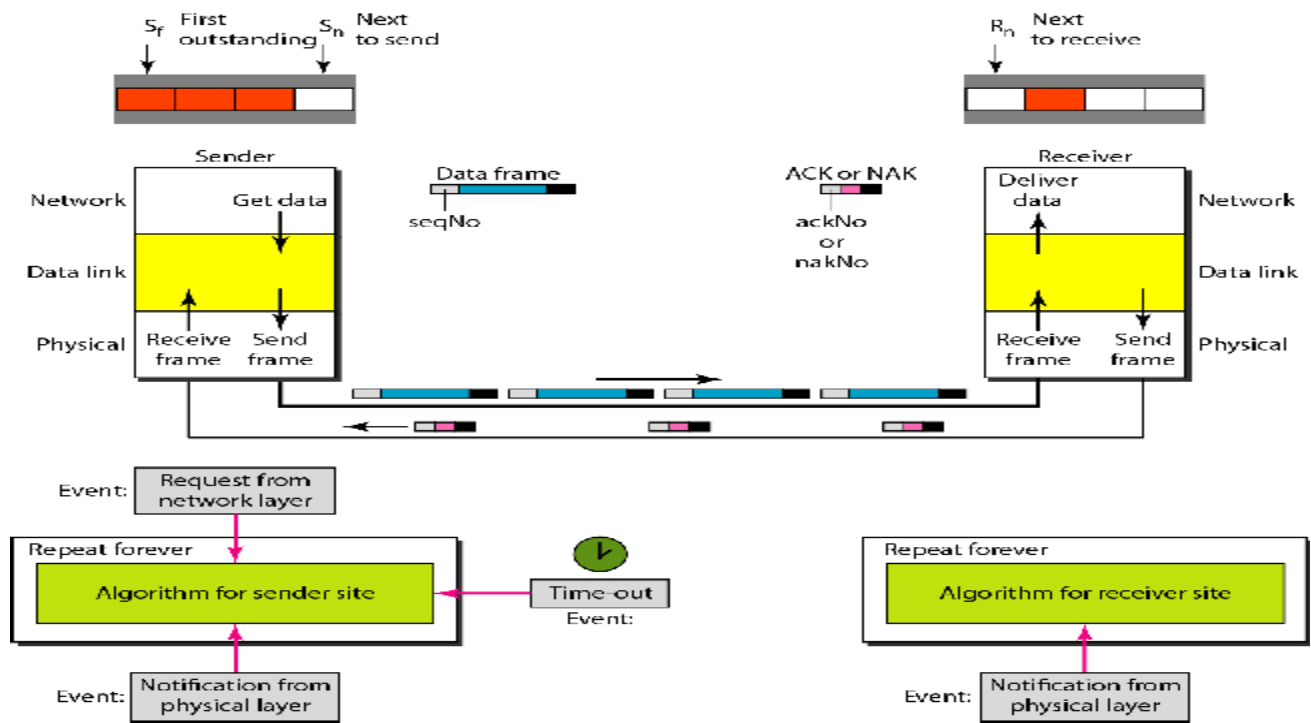


Fig 2.31 Selective Repeat Automatic Repeat Request

2.5 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. HDLC provides two common transfer modes that can be used in different configurations: normal response mode (NRM) and asynchronous balanced mode (ABM). HDLC frame format has been shown in Fig.2.32.

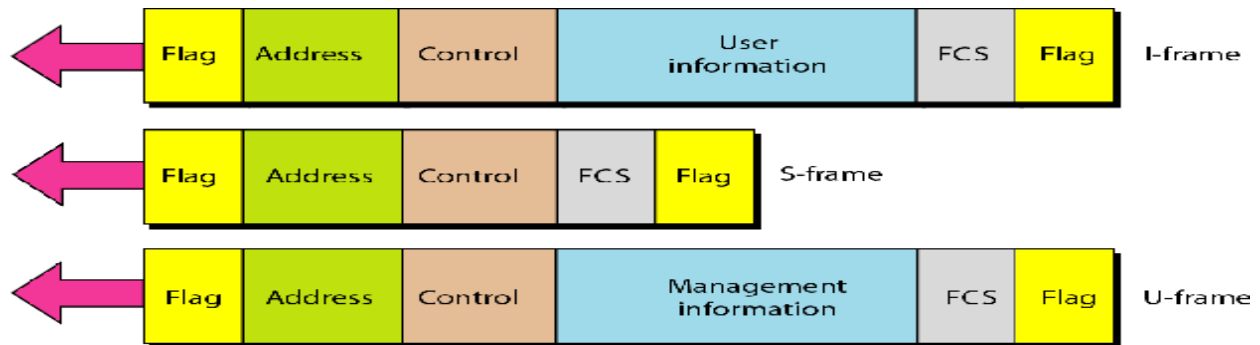


Fig 2.32 HDLC Frames

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 Fig.2.33.

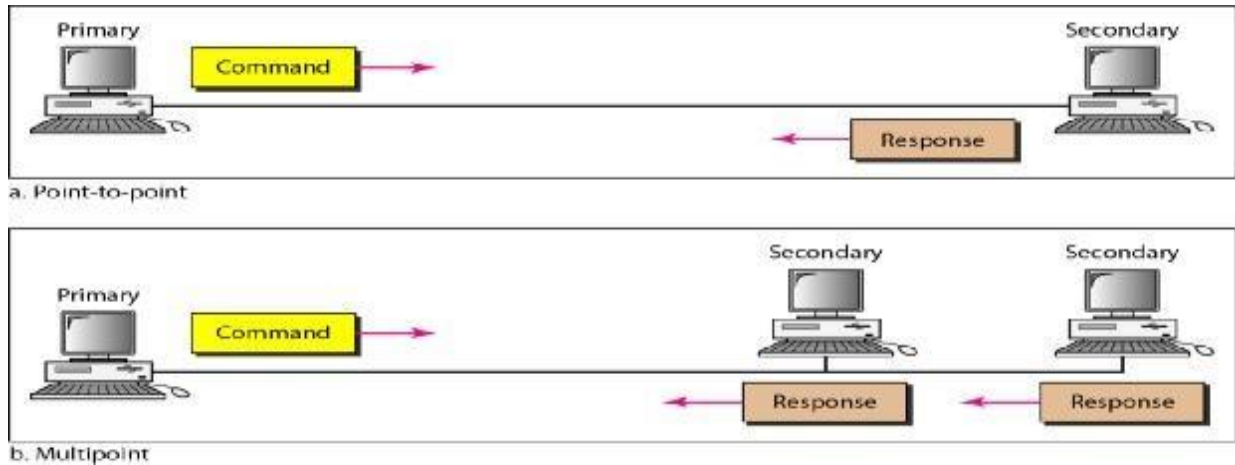


Fig 2.33 Normal Response Mode

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 as shown in Fig 2.34.

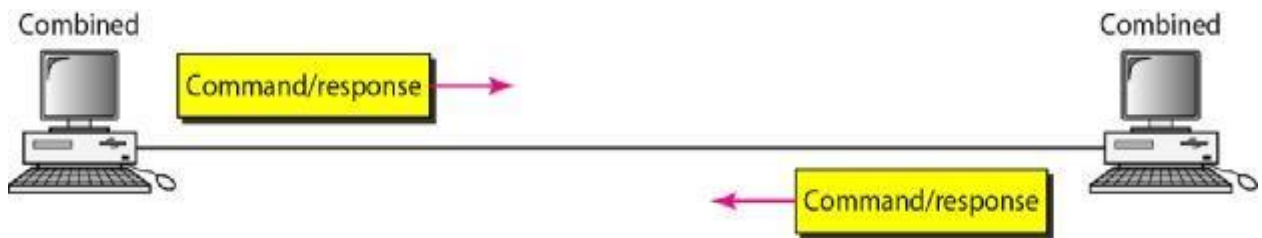


Fig 2.34 Asynchronous Balance Mode

2.6 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 frame format has been shown in Fig. 2.35.

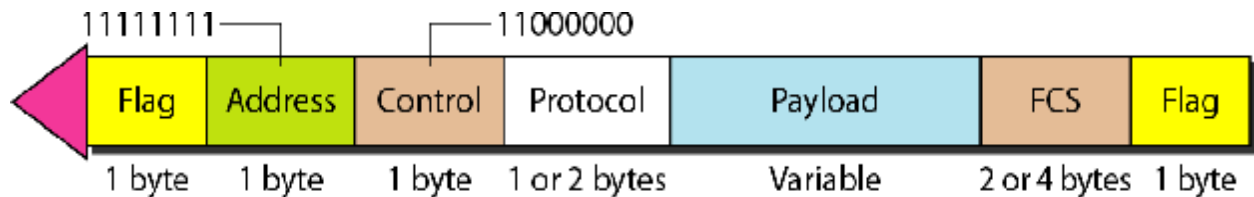


Fig 2.35 PPP Frame Format

PPP provides several services:

- PPP defines the format of the frame to be exchanged between devices.
- PPP defines how two devices can negotiate the establishment of the link and the exchange of data.
- PPP defines how network layer data are encapsulated in the data link frame.
- PPP defines how two devices can authenticate each other.
- PPP provides multiple network layer services supporting a variety of network layer protocols.
- PPP provides connections over multiple links.
- 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:

- PPP does not provide flow control. A sender can send several frames one after another with no concern about overwhelming the receiver.
- 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.
- PPP does not provide a sophisticated addressing mechanism to handle frames in a multipoint configuration.

UNIT-III

3.1 MULTIPLE ACCESS

Data link control is a mechanism which provides a link with reliable communication. In the protocols we assumed that there is an available dedicated link (or channel) between the sender and the receiver. This assumption may or may not be true. If, indeed, we have a dedicated link, as when we connect to the Internet using PPP as the data link control protocol, then the assumption is true and we do not need anything else.

On the other hand, if we use our cellular phone to connect to another cellular phone, the channel (the band allocated to the vendor company) is not dedicated. A person a few feet away from us may be using the same channel to talk to her friend. We can consider the data link layer as two sublayers. The upper sublayer is responsible for data link control, and the lower sublayer is responsible for resolving access to the shared media. If the channel is dedicated, we do not need the lower sublayer. Fig.3.1 shows these two sublayers in the data link layer

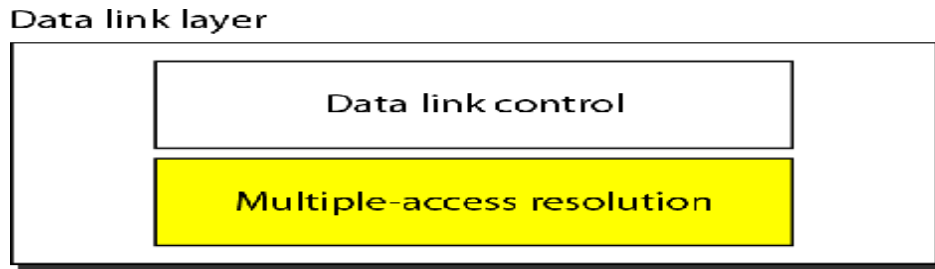


Fig. 3.1 Data link layer divided into two functionality-oriented sublayers

The upper sublayer that is responsible for flow and error control is called the logical (Data) link control (LLC) layer; the lower sublayer that is mostly responsible for multipleaccess resolution is called the media access control (MAC) layer. When nodes or stations 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. The problem of controlling the access to the medium is similar to the rules of speaking in an assembly. The procedures guarantee that the right to speak is upheld and ensure that two people do not speak at the same time, do not interrupt each other, do not monopolize the discussion, and so on. The situation is similar for multipoint networks. Many formal protocols have been devised to handle access to a shared link. We categorize them into three groups. Protocols belonging to each group are shown in Fig. 3.2.

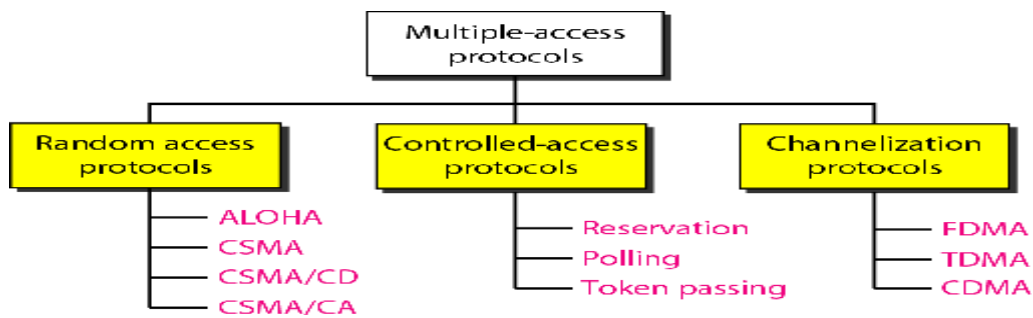


Fig. 3.2 Taxonomy of multiple-access protocols

3.1.1 RANDOM ACCESS

In random access or contention methods, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send. At each instance, a station 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). In other words, each station can transmit when it desires on the condition that it follows the predefined procedure, including the testing of the state of the medium. Two features give this method its name. First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called *random access*. Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called *contention* methods.

In a random access method, each station has the right to the medium without being controlled by any other station. However, if more than one station 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 station follows a procedure that answers the following questions:

1. When can the station access the medium?
2. What can the station do if the medium is busy?
3. How can the station determine the success or failure of the transmission?
4. What can the station do if there is an access conflict?

The random access methods have evolved from a very interesting 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 station to sense the medium before transmitting. This was called carrier sense multiple access. This method later evolved into two parallel methods: carrier sense multiple access with collision detection (CSMA/CD) and carrier sense multiple access with collision avoidance (CSMA/CA). CSMA/CD tells the station what to do when a collision is detected. CSMA/CA tries to avoid the collision.

A) ALOHA

ALOHA, the earliest random access method, was developed at the University of Hawaii in early 1970. It was designed for a radio (wireless) LAN, but it can be used on any shared medium. It is obvious that there are potential collisions in this arrangement. The medium is shared between the stations. When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled.

B) Pure ALOHA

The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station 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 stations. Fig.3.3 shows an example of frame collisions in pure ALOHA.

There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. The figure shows that each station 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. Fig.3.3 shows that only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3. We need to mention that even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed. 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 station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.

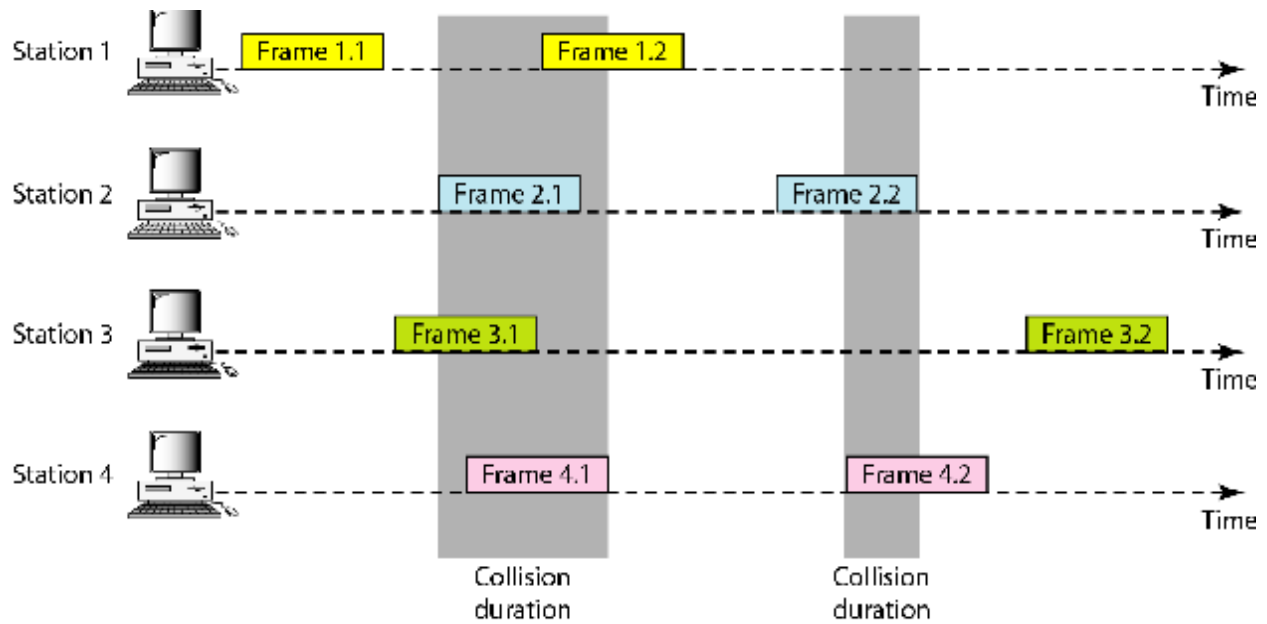


Fig. 3.3 Frames in a pure ALOHA network

A collision involves two or more stations. If all these stations 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 station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. We call this time 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 retransmission attempts K_{max} , a station must give up and try later. Fig.3.4 shows the procedure for pure ALOHA based on the above strategy.

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 stations ($2 \times T_p$). 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 $2^K - 1$ is randomly chosen and multiplied by T_p (maximum propagation time) or T_{rr} (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.

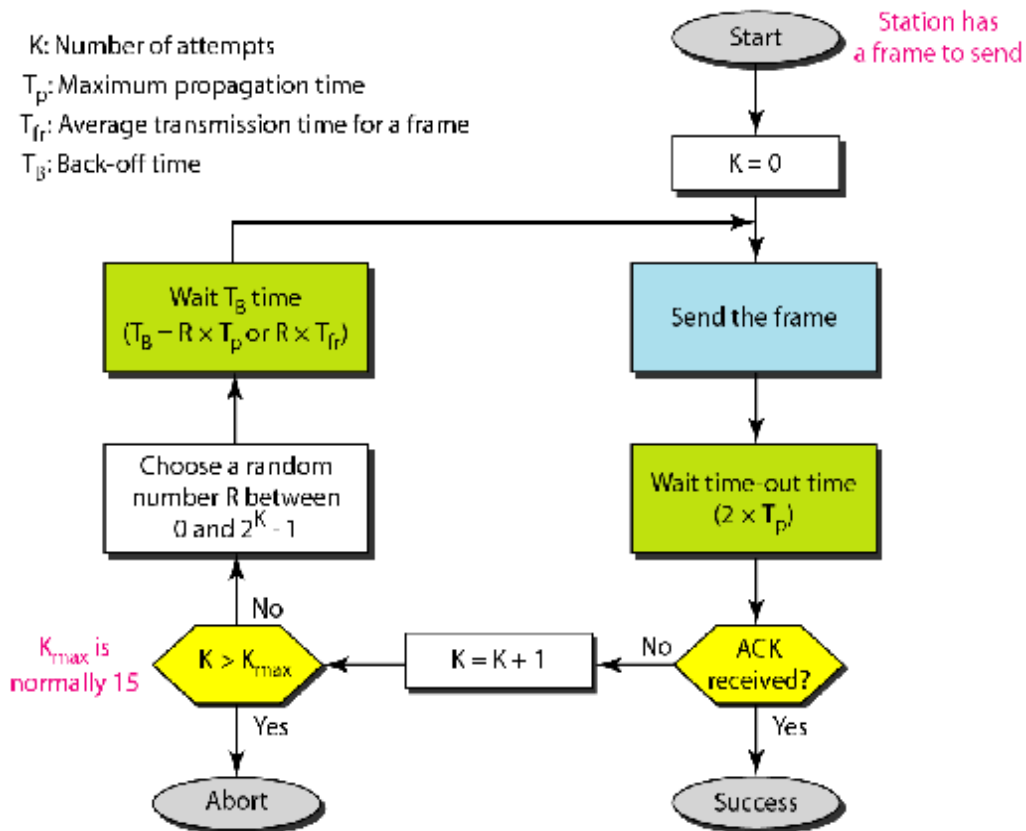


Fig. 3.4 Procedure for pure ALOHA protocol

Pure ALOHA vulnerable time = $2 \times T_{fr}$

The throughput for pure ALOHA is $S = G \times e^{-2G}$.

The maximum throughput $S_{max} = 0.184$ when $G = (1/2)$.

C) 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 station can send. A station may send soon after another station has started or soon before another station 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} s and force the station to send only at the beginning of the time slot. Fig. 3.5 shows an example of frame collisions in slotted ALOHA.

Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station 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 stations 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} . Fig. 3.5 shows the situation

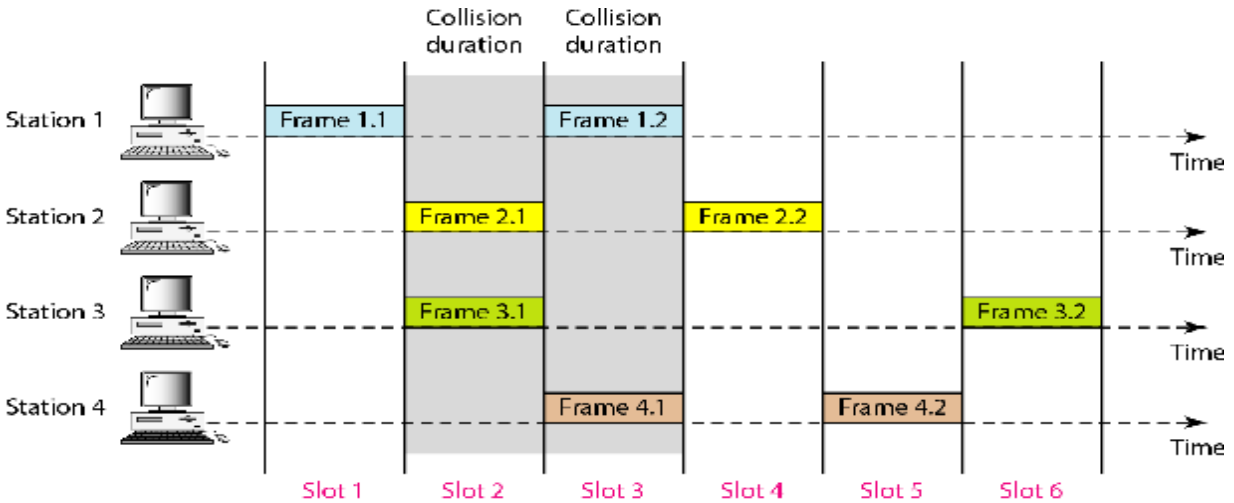


Fig. 3.5 Frames in a slotted ALOHA network

Fig.3.5 shows that the vulnerable time for slotted ALOHA is one-half that of pure ALOHA.

Slotted ALOHA vulnerable time = T_{fr}

The throughput for slotted ALOHA is $S = G \times e^{-G}$.

The maximum throughput $S_{max} = 0.368$ when $G=1$

3.1.2 CDMA

Code division multiple access (CDMA) is a channel access method used by various radio communication technologies. CDMA is an example of multiple access, in which several transmitters can send information simultaneously over a single communication channel. This allows several users to share a band of frequencies. To permit this to be achieved without undue interference between the users, CDMA employs spread-spectrum technology and a special coding scheme (where each transmitter is assigned a code). CDMA is used as the access method in many mobile phone standards such as cdmaOne, CDMA2000 (the 3G evolution of cdmaOne), and WCDMA (the 3G standard used by GSM carriers), which are often referred to as simply *CDMA*.

CDMA is a spread-spectrum multiple access technique. A spread spectrum technique spreads the bandwidth of the data uniformly for the same transmitted power. A spreading code is a pseudo-random code that has a narrow ambiguity function, unlike other narrow pulse codes. In CDMA a locally generated code runs at a much higher rate than the data to be transmitted. Data for transmission is combined via bitwise XOR (exclusive OR) with the faster code. The data signal with pulse duration of T_b (symbol period) is XOR'ed with the code signal with pulse duration of T_c (chip period). (Note: bandwidth is proportional to $1/T$, where T = bit time.) Therefore, the bandwidth of the data signal is $1/T_b$ and the bandwidth of the spread spectrum signal is $1/T_c$. Since T_c is much smaller than T_b , the bandwidth of the spread spectrum signal is much larger than the bandwidth of the original signal. The ratio T_b/T_c is called the spreading factor or processing gain and determines to a certain extent the upper limit of the total number of users supported simultaneously by a base station.

Generation of a CDMA signal

Each user in a CDMA system uses a different code to modulate their signal. Choosing the codes used to modulate the signal is very important in the performance of CDMA systems. The best performance will occur when there is good separation between the signal of a desired user and the signals of other users. The separation of the signals is made by correlating the received signal with the locally generated code of the desired user. If the signal matches the desired user's code then the correlation function will be high and the system can extract that signal. If the desired user's code has nothing in common with the signal the correlation should be as close to zero as possible (thus eliminating the signal); this is referred to as cross-correlation. If the code is correlated with the signal at any time offset other than zero, the correlation should be as close to zero as possible. This is referred to as auto-correlation and is used to reject multi-path interference.

An analogy to the problem of multiple access is a room (channel) in which people wish to talk to each other simultaneously. To avoid confusion, people could take turns speaking (time division), speak at different pitches (frequency division), or speak in different languages (code division). CDMA is analogous to the last example where people speaking the same language can understand each other, but other languages are perceived as noise and rejected. Similarly, in radio CDMA, each group of users is given a shared code. Many codes occupy the same channel, but only users associated with a particular code can communicate.

3.1.3 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 station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station 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 stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide. In Fig.3.6, stations A and C are involved in the collision.

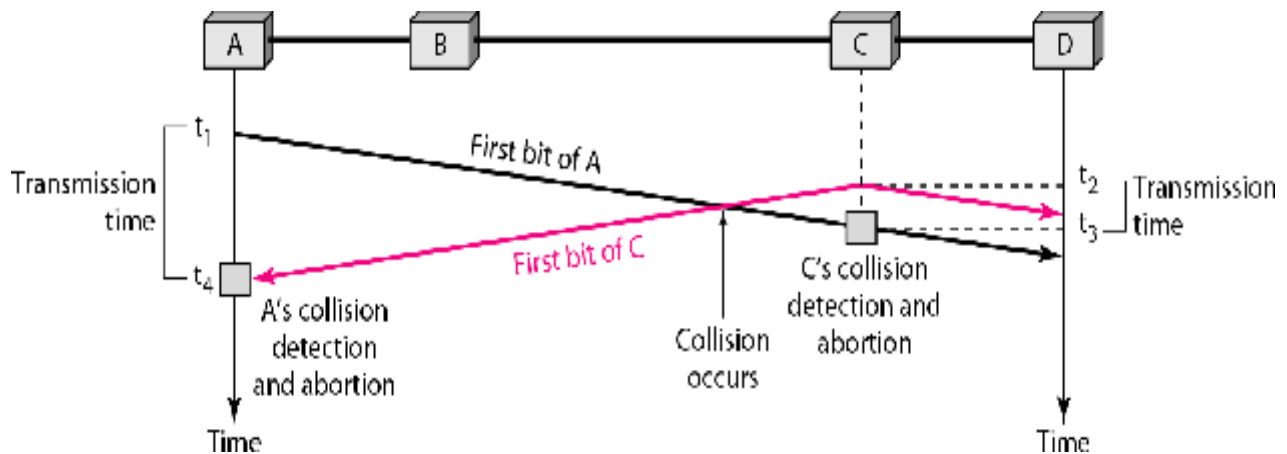


Fig. 3.6 Collision of the first bit in CSMA/CD

At time t_1 , station A has executed its persistence procedure and starts sending the bits of its frame. At time t_2 , station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right. The collision occurs sometime after time t_2' . Station C detects a collision at time t_3 when it receives the first bit of A's frame. Station C immediately (or after a short time, but we assume immediately) aborts transmission. Station A detects collision at time t_4 when it receives the first bit of C's frame; it also immediately aborts transmission. Looking at the figure, we see that A transmits for the duration $t_4 - t_1$; C transmits for the duration $t_3 - t_2'$. Later we show that, for the protocol to work, the length of any frame divided by the bit rate in this protocol must be more than either of these durations. At time t_4 , the transmission of A's frame, though incomplete, is aborted; at time t_3 , the transmission of B's frame, though incomplete, is aborted. Now that we know the time durations for the two transmissions, we can show a more complete graph in Fig. 3.7.

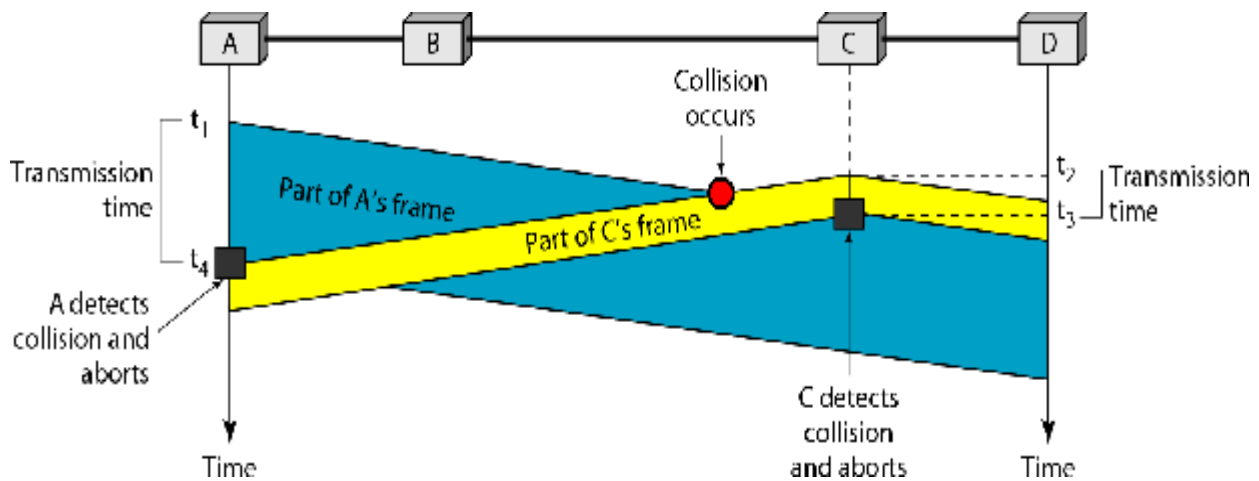


Fig. 3.7 Collision and abortion in CSMA/CD

Procedure

Now let us look at the flow diagram for *CSMA/CD* in Fig. 3.8. It is similar to the one for the ALOHA protocol, but there are differences. The first difference is the addition of the persistence process. We need to sense the channel before we start sending the frame by using one of the persistence processes. The second difference is the frame transmission. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In *CSMA/CD*, transmission and collision detection is a continuous process. We do not send the entire frame and then look for a collision. The station transmits and receives continuously and simultaneously (using two different ports). We use a loop to show that transmission is a continuous process. We constantly monitor in order to detect one of two conditions: either transmission is finished or a collision is detected. Either event stops transmission. When we come out of the loop, if a collision has not been detected, it means that transmission is complete; the entire frame is transmitted. Otherwise, a collision has occurred. The third difference is the sending of a short jamming signal that enforces the collision in case other stations have not yet sensed the collision. The throughput of *CSMA/CD* is greater than that of pure or slotted ALOHA. The maximum throughput occurs at a different value of G and is based on the persistence method and the value of p in the p -persistent approach.

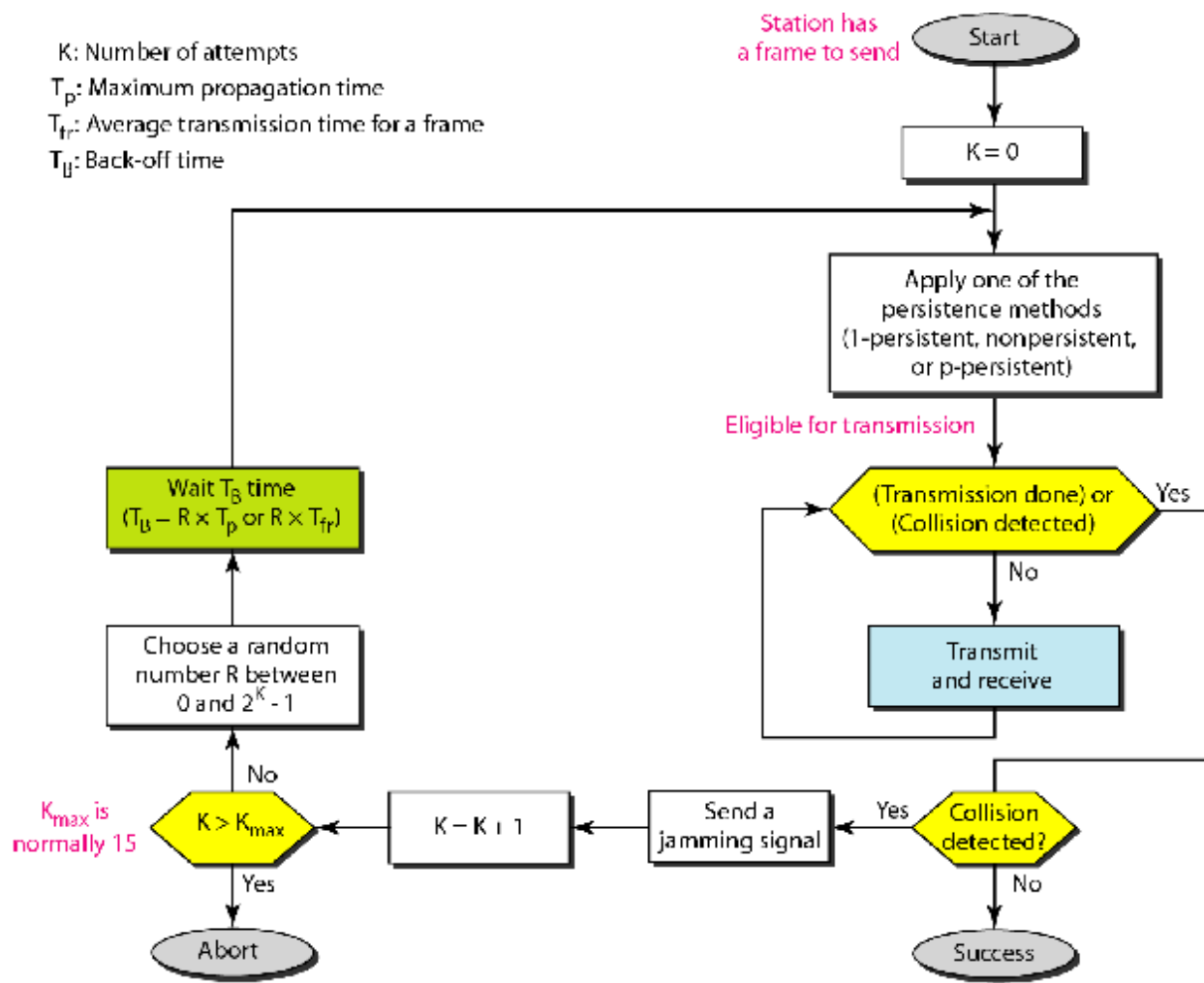


Fig. 3.8 Flow diagram for the CSMA/CD

3.1.4 Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

The basic idea behind *CSMA/CD* is that a station needs to be able to receive while transmitting to detect a collision. When there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station. 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 station needs to add a significant amount of energy to the one created by the first station. In a wired network, the received signal has almost the same energy as the sent signal because either the length of the cable is short or there are repeaters that amplify the energy between the sender and the receiver. This means that in a collision, the detected energy almost doubles. However, in a wireless network, much of the sent energy is lost in transmission. The received signal has very little energy. Therefore, a collision may add only 5 to 10 percent additional energy. This is not useful for effective collision detection. We need to avoid collisions on wireless networks because they cannot be detected. Carrier sense multiple access with collision avoidance (*CSMA/CA*) was

invented for this network. Collisions are avoided through the use of CSMA/CA's three strategies: the interframe space, the contention window, and acknowledgments, as shown in Fig. 3.9.

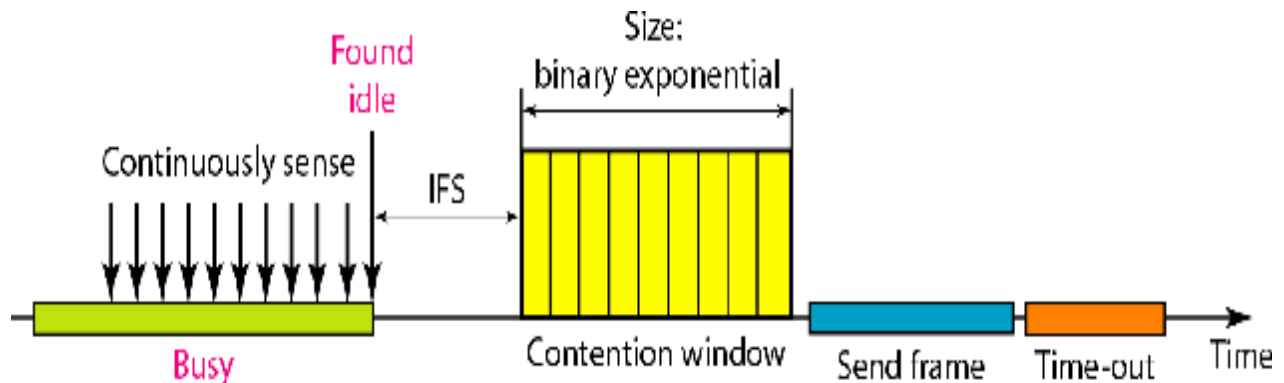


Fig. 3.9 Timing in CSMA/CA

Interframe Space (IFS)

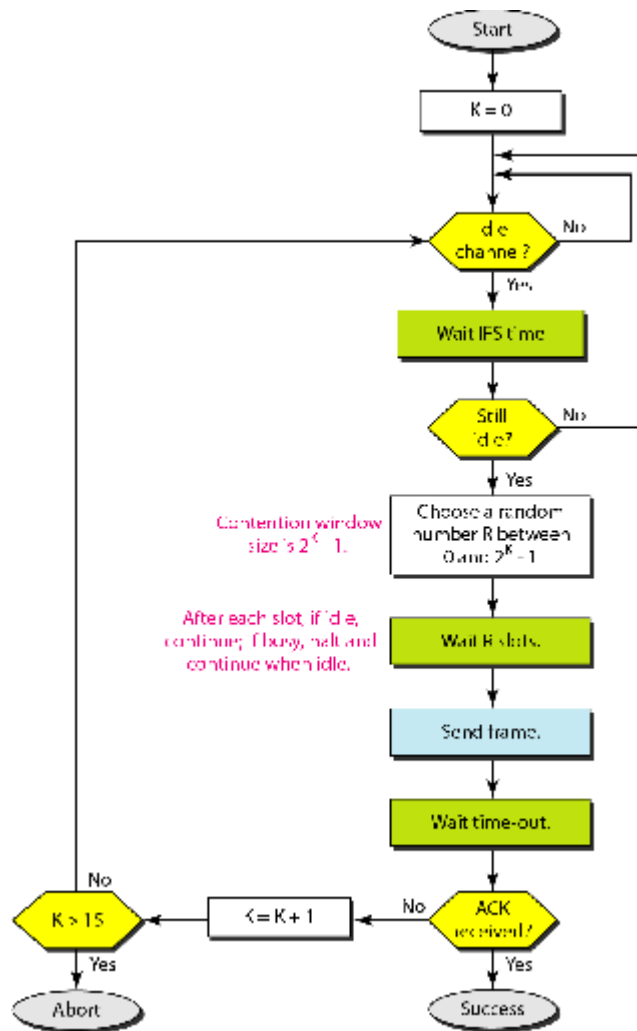
First, collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the interframe space or IFS. Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting. The distant station's signal has not yet reached this station. The IFS time allows the front of the transmitted signal by the distant station to reach this station. The IFS variable can also be used to prioritize stations or frame types. For example, a station that is assigned a shorter IFS has a higher priority.

- In CSMA/CA, the IFS can also be used to define the priority of a station or a frame.
- In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle.

With all these precautions, there still may be a collision resulting in destroyed data. In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.

Procedure

Fig. 3.10 shows the procedure. Note that the channel needs to be sensed before and after the IFS. The channel also needs to be sensed during the contention time. For each time slot of the contention window, the channel is sensed. If it is found idle, the timer continues; if the channel is found busy, the timer is stopped and continues after the timer becomes idle again.



Fog. 3.10 Flow diagram for CSMA/CA

3.2 CONTROLLED ACCESS

In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations. We discuss three popular controlled-access methods.

A) RESERVATION

In the reservation method, a station 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 stations in the system, there are exactly N reservation mini slots in the reservation frame. Each mini slot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own mini slot. The stations that have made reservations can send their data frames after the reservation frame. Fig. 3.11 shows a situation with five stations and a five-mini slot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.

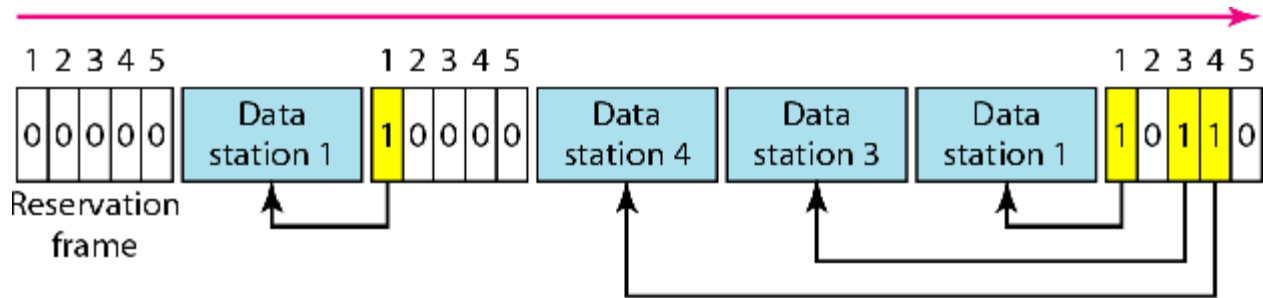


Fig. 3.11 Reservation access method

B) POLLING

Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations. 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 (see Fig. 3.12).

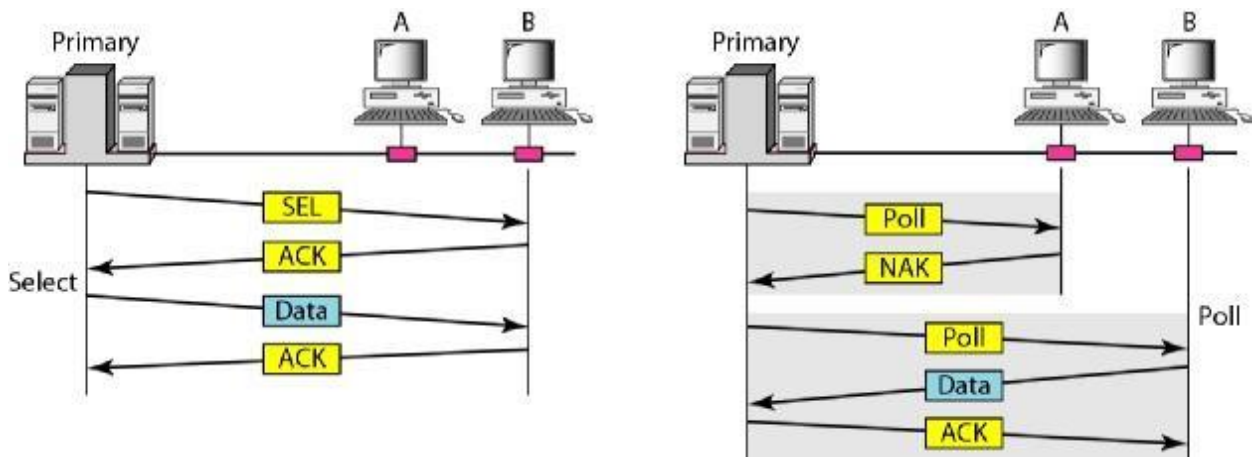


Fig. 3.12 Select and poll functions in polling access method

If the primary wants to receive data, it asks the secondary's 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.

C) TOKEN PASSING

In the token-passing method, the stations in a network are organized in a logical ring. In other words, for each station, there is a *predecessor* and a *successor*. The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring. The current station is the one that is accessing the channel now. The right to this access has been passed from the predecessor to the current station. The right will be passed to the successor when the current station has no more data to send. But how is the right to access the channel passed from one station to another? In this method, a special packet called a token circulates through the ring. The possession of the token gives the station the right to access the channel and send its data.

When a station 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 station has no more data to send, it releases the token, passing it to the next logical station in the ring. The station cannot send data until it receives the token again in the next round. In this process, when a station receives the token and has no data to send, it just passes the data to the next station. Token management is needed for this access method. Stations 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 station that is holding the token fails, the token will disappear from the network. Another function of token management is to assign priorities to the stations and to the types of data being transmitted. And finally, token management is needed to make low-priority stations release the token to high priority stations.

Logical Ring

In a token-passing network, stations do not have to be physically connected in a ring; the ring can be a logical one. In the physical ring topology, when a station sends the token to its successor, the token cannot be seen by other stations; 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 stations fails, the whole system fails. The dual ring topology uses a second (auxiliary) ring which operates in the reverse direction compared with the main ring. The second ring is for emergencies only (such as a spare tire for a car). If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring. After the failed link is restored, the auxiliary ring becomes idle again. Note that for this topology to work, each station needs to have two transmitter ports and two receiver ports. The high-speed Token Ring networks called FDDI (Fiber Distributed Data Interface) and CDDI (Copper Distributed Data Interface) use this topology. In the bus ring

topology, also called a token bus, the stations are connected to a single cable called a bus. They, however, make a logical ring, because each station knows the address of its successor (and also predecessor for token management purposes).

When a station has finished sending its data, it releases the token and inserts the address of its successor in the token. Only the station 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 stations are connected to this ring through the two wire connections. This topology makes the network less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate. Also adding and removing stations from the ring is easier. This topology is still used in the Token Ring LAN designed by IBM.

3.3 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 stations. In this section, we discuss three channelization protocols: FDMA, TDMA, and CDMA.

A) Frequency-Division Multiple Access (FDMA)

In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands. Each station is allocated a band to send its data. In other words, each band is reserved for a specific station, and it belongs to the station all the time. Each station also uses a bandpass filter to confine the transmitter frequencies. To prevent station interferences, the allocated bands are separated from one another by small *guard bands*. Fig. 3.13 shows the idea of FDMA

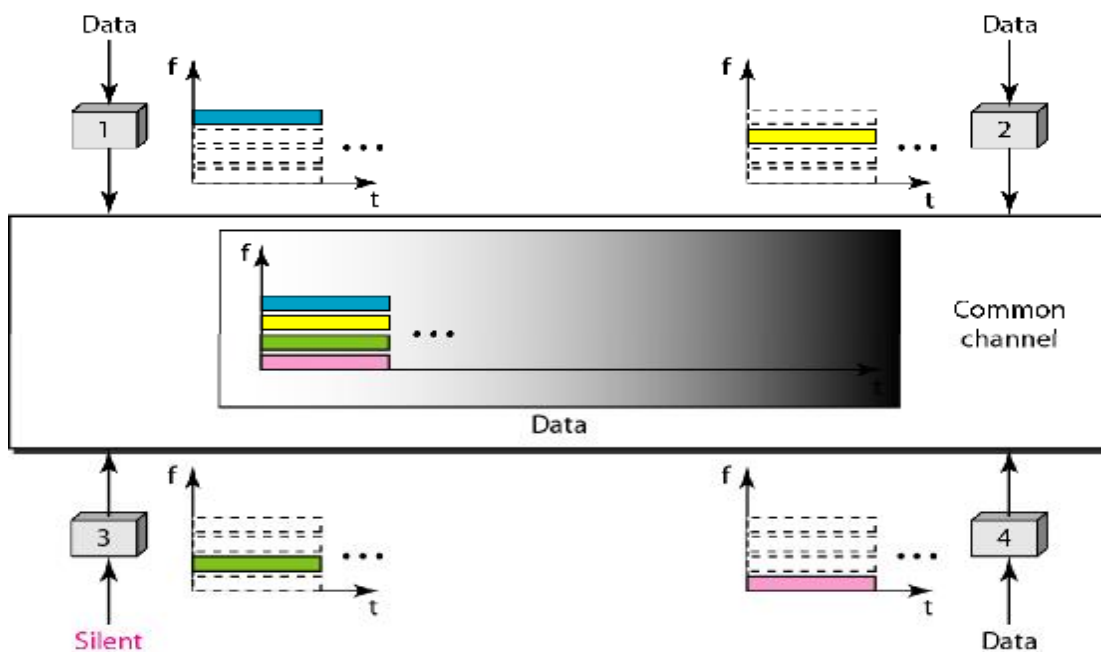


Fig. 3.13 Frequency-division multiple access (FDMA)

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.

We need to emphasize that although FDMA and FDM conceptually seem similar, there are differences between them. 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 bandpass signal. The bandwidth of each channel is shifted by the multiplexer. FDMA, on the other hand, is an access method in the data link layer. The data link layer in each station tells its physical layer to make a bandpass 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 station are automatically bandpass-filtered. They are mixed when they are sent to the common channel.

B) Time-Division Multiple Access (TDMA)

In time-division multiple access (TDMA), the stations share the bandwidth of the channel in time. Each station is allocated a time slot during which it can send data. Each station transmits its data in its assigned time slot. Fig. 3.14 shows the idea behind TDMA.

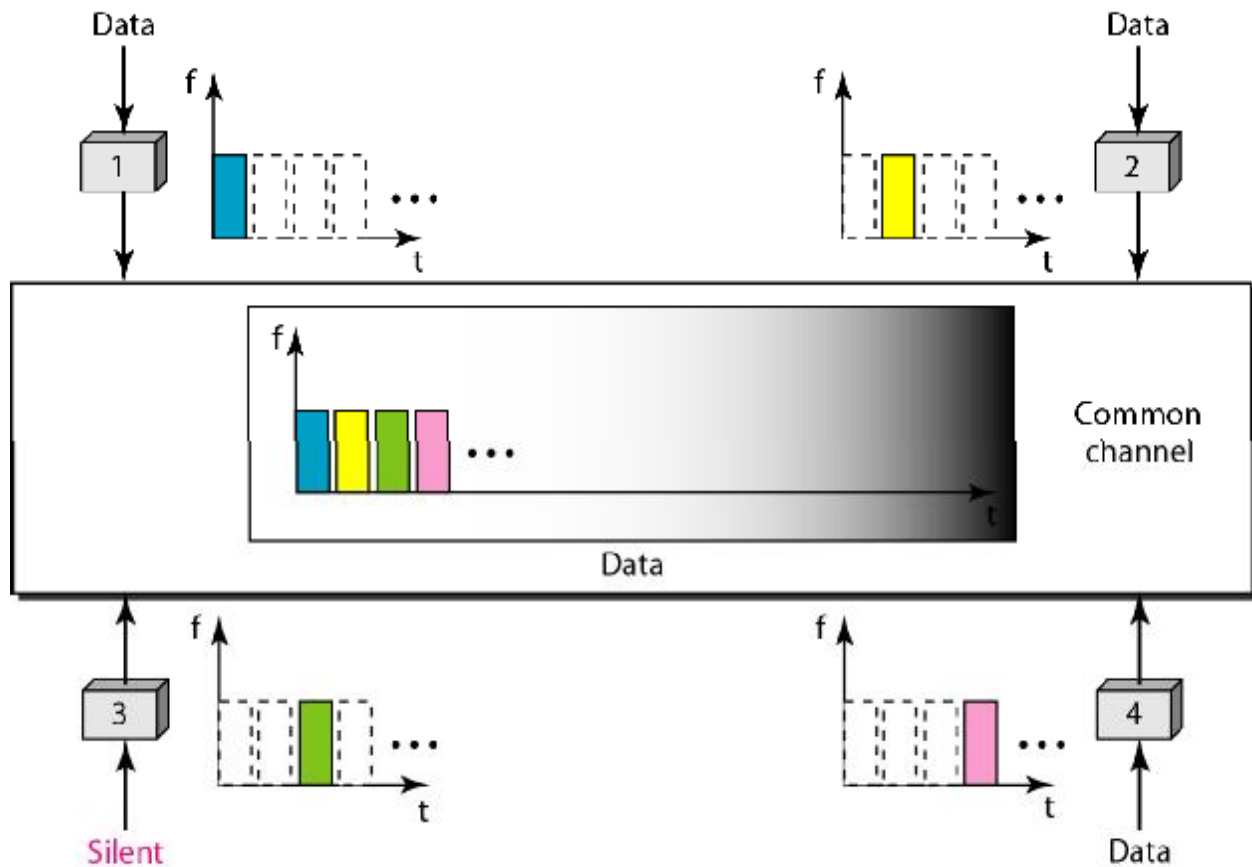


Fig. 3.14 Time-division multiple access (TDMA)

The main problem with TDMA lies in achieving synchronization between the different stations. Each station 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 stations are spread over a large area. To compensate for the delays, we can insert *guard times*. Synchronization is normally accomplished by having some synchronization bits (normally referred to as preamble bits) at the beginning of each slot. We also need to emphasize that although TDMA and TDM conceptually seem the same, there are differences between them. TDM, is a physical layer technique that combines the data from slower channels and transmits them by using a faster channel. The process uses a physical multiplexer that interleaves data units from each channel.

TDMA, on the other hand, is an access method in the data link layer. The data link layer in each station tells its physical layer to use the allocated time slot. There is no physical multiplexer at the physical layer.

C) 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 stations can send data simultaneously; there is no timesharing. Let us first give an analogy. 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). Let us assume we have four stations 1, 2, 3, and 4 connected to the same channel. The data from station 1 are d_1 , from station 2 are d_2 , and so on. The code assigned to the first station is c_1 , to the second is c_2 , and so on. We assume that the assigned codes have two properties.

1. If we multiply each code by another, we get 0.
2. If we multiply each code by itself, we get 4 (the number of stations).

With these two properties in mind, let us see how the above four stations can send data using the same common channel, as shown in Fig. 3.15. Station 1 multiplies (a special kind of multiplication, as we will see) its data by its code to get $d_1 \cdot c_1$. Station 2 multiplies its data by its code to get $d_2 \cdot c_2$. And so on. The data that go on the channel are the sum of all these terms, as shown in the box. Any station that wants to receive data from one of the other three multiplies the data on the channel by the code of the sender. For example, suppose stations 1 and 2 are talking to each other. Station 2 wants to hear what station 1 is saying. It multiplies the data on the channel by c_1 . Because $(c_1 \cdot c_1)$ is 4, but $(c_2 \cdot c_1)$, $(c_3 \cdot c_1)$, and $(c_4 \cdot c_1)$ are all 0s, station 2 divides the result by 4 to get the data from station 1.

$$\begin{aligned} \text{data} &= (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4) \cdot c_1 \\ &= d_1 \cdot c_1 \cdot c_1 + d_2 \cdot c_2 \cdot c_1 + d_3 \cdot c_3 \cdot c_1 + d_4 \cdot c_4 \cdot c_1 = 4 \times d_1 \end{aligned}$$

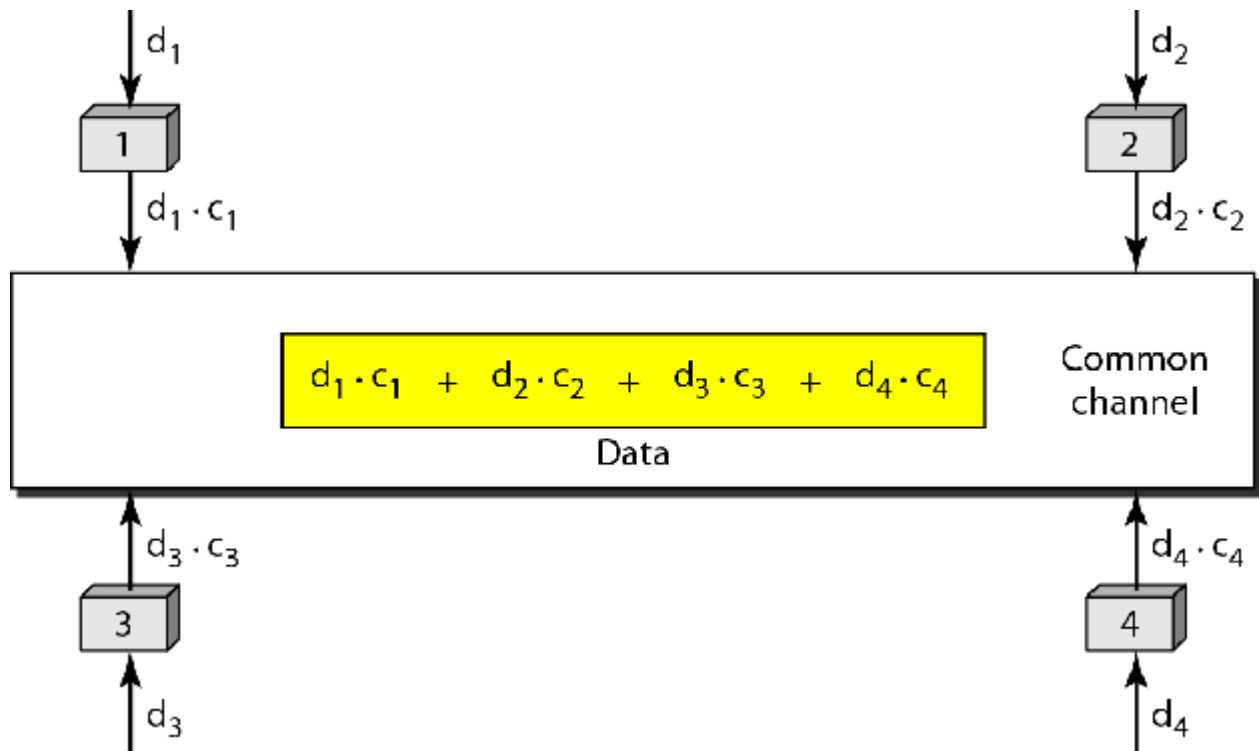


Fig. 3.15 Simple idea of communication with code

Chips

CDMA is based on coding theory. Each station is assigned a code, which is a sequence of numbers called chips. For now, we need to know that we did not choose the sequences randomly; they were carefully selected. They are called orthogonal sequences and have the following properties:

1. Each sequence is made of N elements, where N is the number of stations.
2. If we multiply a sequence by a number, every element in the sequence is multiplied by that element. This is called multiplication of a sequence by a scalar.
3. If we multiply two equal sequences, element by element, and add the results, we get N , where N is the number of elements in the each sequence. This is called the inner product of two equal sequences.
4. If we multiply two different sequences, element by element, and add the results, we get 0. This is called inner product of two different sequences.
5. Adding two sequences means adding the corresponding elements. The result is another sequence.

3.4 WIRED LAN

In this we learned that 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. Some of these technologies survived for a while, but Ethernet is by far the dominant technology. In this, we first briefly discuss the IEEE Standard Project 802, designed to regulate the manufacturing and interconnectivity between different LANs. We then concentrate on the Ethernet LANs. Although Ethernet has gone through a four-generation evolution during the last few decades, the main concept has remained. Ethernet has changed to meet the market needs and to make use of the new technologies.

3.4.1 IEEE STANDARDS

In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers. Project 802 does not seek to replace any part of the OSI or the Internet model. Instead, it is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.

The standard was adopted by the American National Standards Institute (ANSI). In 1987, the International Organization for Standardization (ISO) also approved it as an international standard under the designation ISO 8802. The relationship of the 802 Standard to the traditional OSI model is shown in Fig. 3.16. The IEEE has subdivided the data link layer into two sublayers: logical link control (LLC) and media access control (MAC). IEEE has also created several physical layer standards for different LAN protocols.

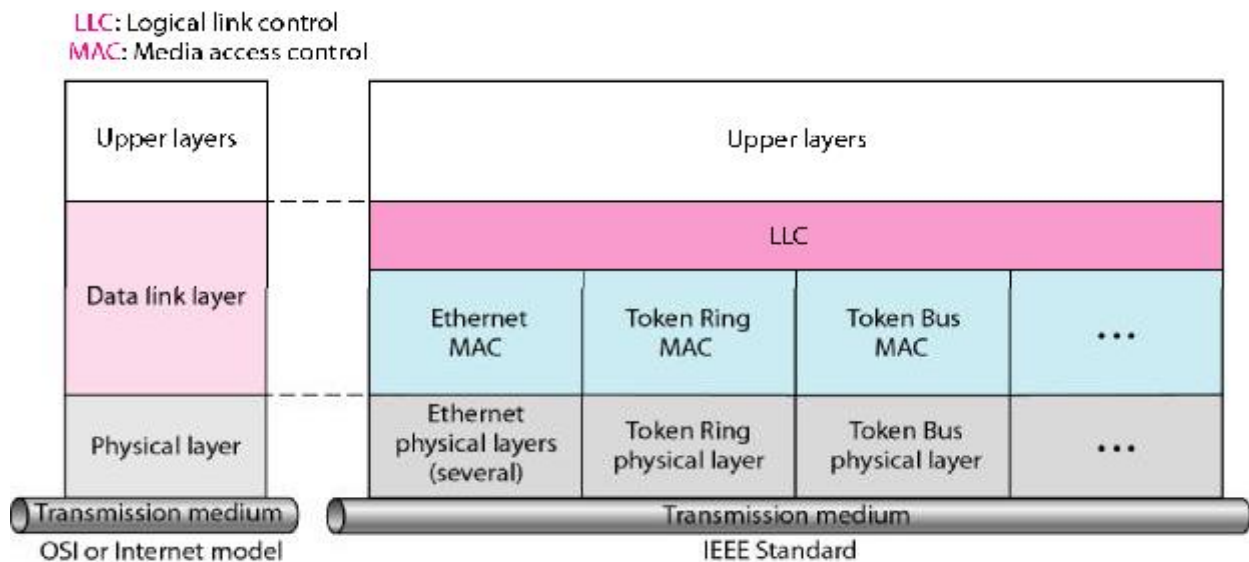


Fig. 3.16 IEEE standard for LANs

A) DATA LINK LAYER

As we mentioned before, the data link layer in the IEEE standard is divided into two sublayers: LLC and MAC.

Logical Link Control (LLC)

We have discussed about data link control in UNIT-2. We said that data link control handles framing, flow control, and error control. In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sublayer called the logical link control. Framing is handled in both the LLC sublayer and the MAC sublayer. 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 sublayer, which provides different protocols for different LANs. A single LLC protocol can provide interconnectivity between different LANs because it makes the MAC sublayer transparent. Fig. 3.16 shows one single LLC protocol serving several MAC protocols. Framing LLC defines a protocol data unit (PDU) that is somewhat similar to that of HDLC. The header contains a control field like the one in HDLC; this field is used for flow and error control. The two other header fields define the upper-layer protocol at the source and destination that uses LLC. These fields are called the destination service access point (DSAP) and the source service access point (SSAP). The other fields defined in a typical data link control protocol such as HDLC are moved to the MAC sublayer. In other words, a frame defined in HDLC is divided into a PDU at the LLC sublayer and a frame at the MAC sublayer, as shown in Fig. 3.17. Need for LLC The purpose of the LLC is to provide flow and error control for the upper-layer protocols that actually demand these services. For example, if a LAN or several LANs are used in an isolated system, LLC may be needed to provide flow and error control for the application layer protocols. However, most upper-layer protocols such as IP, do not use the services of LLC.

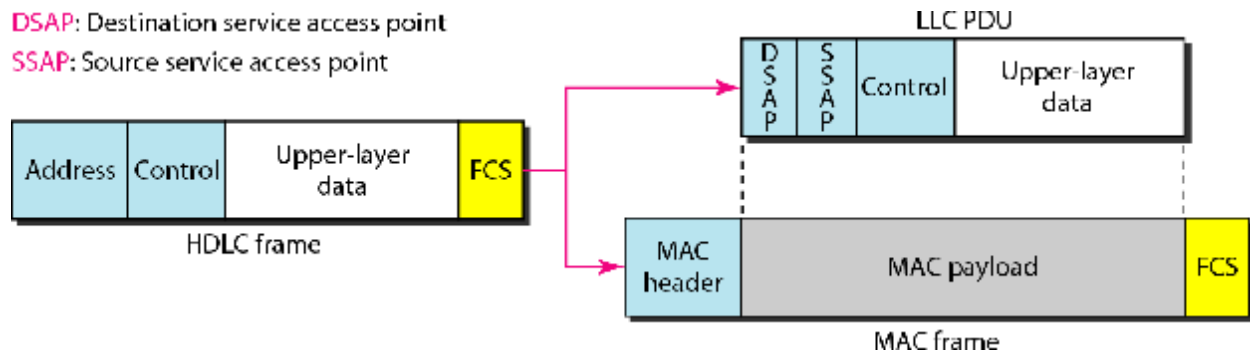


Fig. 3.17 HDLC frame compared with LLC and MAC frames

Media Access Control (MAC)

We have already discussed multiple access methods including random access, controlled access, and channelization. IEEE Project 802 has created a sublayer called media access control that 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. As we discussed in the previous section, part of the framing function is also handled by the MAC layer. In contrast to the LLC sublayer, the MAC sublayer contains a number of distinct modules; each defines the access method and the framing format specific to the corresponding LAN protocol.

B) PHYSICAL LAYER

The physical layer is dependent on the implementation and type of physical media used. IEEE defines detailed specifications for each LAN implementation. For example, although there is only one MAC sublayer for Standard Ethernet, there is a different physical layer specification for each Ethernet implementations.

3.4.2 STANDARD ETHERNET

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations: Standard Ethernet (10 Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (1 Gbps), and Ten-Gigabit Ethernet (10 Gbps), as shown in Fig. 3.18.

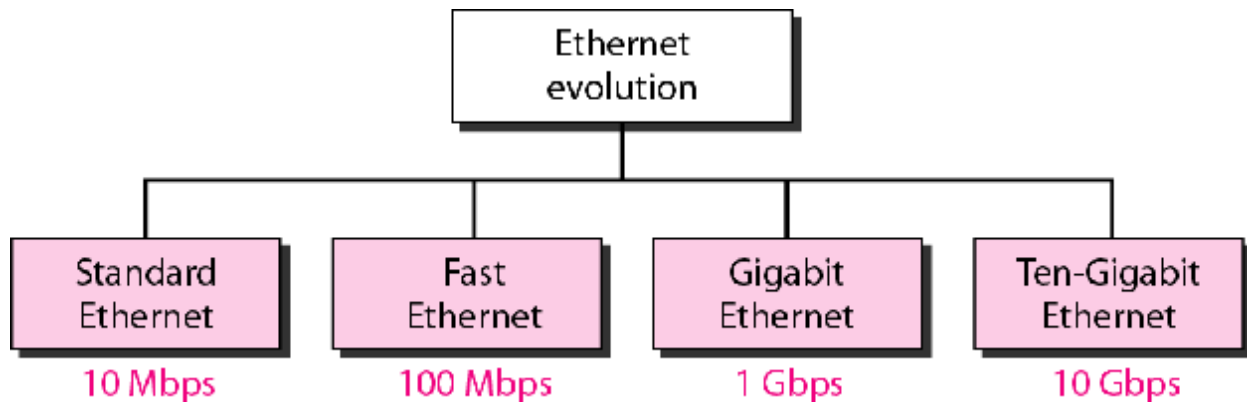


Fig. 3.18 Ethernet evolution through four generations

A) MAC SUBLAYER

In Standard Ethernet, the MAC sublayer governs the operation of the access method. It also frames data received from the upper layer and passes them to the physical layer.

Frame Format

The Ethernet frame contains seven fields: preamble, SFD, DA, SA, length or type of protocol data unit (PDU), upper-layer data, and the CRe. Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium.

Acknowledgments must be implemented at the higher layers. The format of the MAC frame is shown in Fig. 3.19.

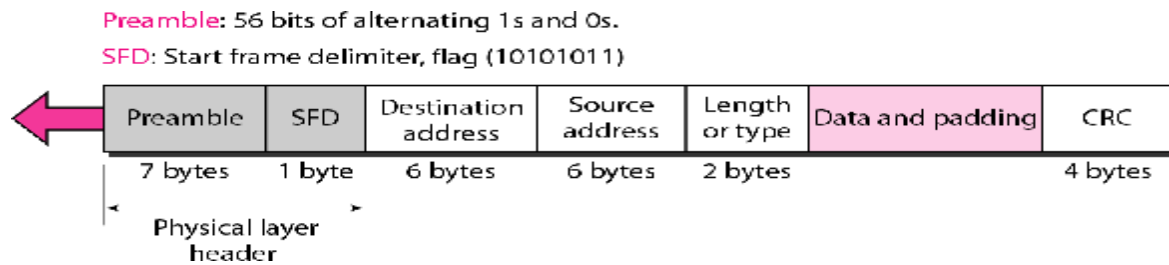


Fig. 3.19 802.3 MAC frame

- **Preamble.** The first field of the 802.3 frame contains 7 bytes (56 bits) of alternating 0s and 1s that alerts the receiving system to the coming frame and enables it to synchronize its input timing. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The preamble is actually added at the physical layer and is not (formally) part of the frame.
- **Start frame delimiter (SFD).** The second field (1 byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits is 11 and alerts the receiver that the next field is the destination address.
- **Destination address (DA).** The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet. We will discuss addressing shortly.
- **Source address (SA).** The SA field is also 6 bytes and contains the physical address of the sender of the packet. We will discuss addressing shortly.
- **Length or type.** This field is defined as a type field or length field. The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame. The IEEE standard used it as the length field to define the number of bytes in the data field. Both uses are common today.
- **Data.** This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes,.
- **CRC.** The last field contains error detection information, in this case a CRC-32

Frame Length

Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame. An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes. Part of this length is the header and the trailer. If we count 18 bytes of header and trailer (6 bytes of source address, 6 bytes of destination address, 2 bytes of length or type, and 4 bytes of CRC), then the minimum length of data from the upper layer is $64 - 18 = 46$ bytes. If the upper-layer packet is less than 46 bytes, padding is added to make up the difference. The standard defines the maximum length of a frame (without preamble and SFD field) as 1518 bytes. If we subtract the 18 bytes of header and trailer, the maximum length of the payload is 1500 bytes. The maximum length restriction has two historical reasons. First, memory was very expensive when Ethernet was designed: a maximum length restriction helped to reduce the size of the buffer. Second, the maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.

Addressing

Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC). The NIC fits inside the station and provides the station with a 6-byte physical address. The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes. Unicast, Multicast, and Broadcast Addresses A source address is always a unicast address-the frame comes from only one station. The destination address, however, can be unicast, multicast, or broadcast. If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast. A unicast destination address defines only one recipient; the relationship between the sender and the receiver is one-to-one. A multicast destination address defines a group of addresses; the relationship between the sender and the receivers is one-to-many. The broadcast address is a special case of the multicast address; the recipients are all the stations on the LAN. A broadcast destination address is forty-eight Is.

Standard Ethernet uses I-persistent CSMA/CD

B) PHYSICAL LAYER

The Standard Ethernet defines several physical layer implementations; four of the most common, are shown in Fig. 3.20.

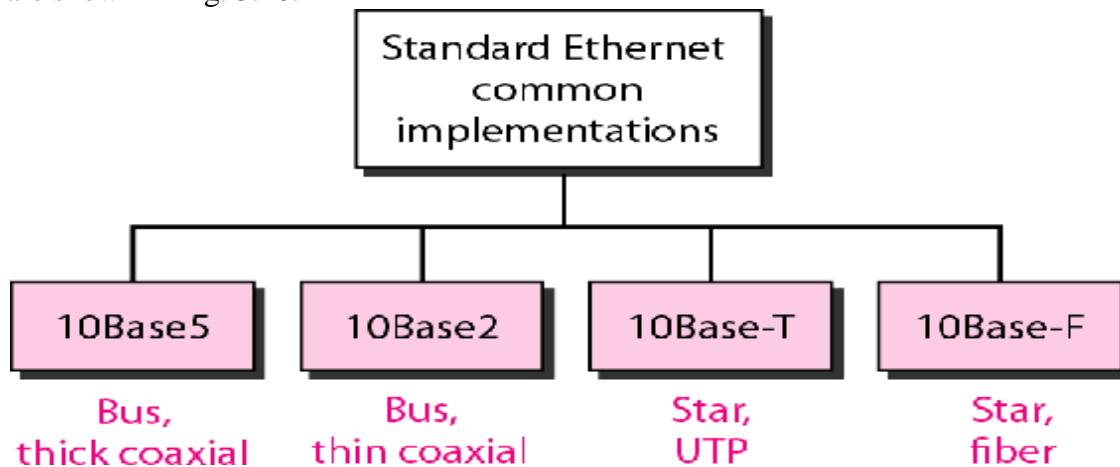


Fig. 3.20 Categories of Standard Ethernet

Encoding and Decoding

All standard implementations use digital signaling (baseband) at 10 Mbps. At the sender, data are converted to a digital signal using the Manchester scheme; at the receiver, the received signal is interpreted as Manchester and decoded into data. Table 3.1 shows a summary of Standard Ethernet implementations.

Table 3.1 Summary of Standard Ethernet implementations

<i>Characteristics</i>	<i>10Base5</i>	<i>10Base2</i>	<i>10Base-T</i>	<i>10Base-F</i>
Media	Thick coaxial cable	Thin coaxial cable	2 UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m
Line encoding	Manchester	Manchester	Manchester	Manchester

3.4.3 FAST ETHERNET

Fast Ethernet was designed to compete with LAN protocols such as FDDI or Fiber Channel (or Fibre Channel, as it is sometimes spelled). IEEE created Fast Ethernet under the name 802.3u. Fast Ethernet is backward-compatible with Standard Ethernet, but it can transmit data 10 times faster at a rate of 100 Mbps. The goals of Fast Ethernet can be summarized as follows:

1. Upgrade the data rate to 100 Mbps.
2. Make it compatible with Standard Ethernet.
3. Keep the same 48-bit address.
4. Keep the same frame format.
5. Keep the same minimum and maximum frame lengths.

A) MAC SUBLAYER

A main consideration in the evolution of Ethernet from 10 to 100 Mbps was to keep the MAC sublayer untouched. However, a decision was made to drop the bus topologies and keep only the star topology. For the star topology, there are two choices, as we saw before: half duplex and full duplex. In the half-duplex approach, the stations are connected via a hub; in the full-duplex approach, the connection is made via a switch with buffers at each port. The access method is the same (*CSMA/CD*) for the half-duplex approach; for full duplex Fast Ethernet, there is no need for *CSMA/CD*. However, the implementations keep *CSMA/CD* for backward compatibility with Standard Ethernet.

Autonegotiation

A new feature added to Fast Ethernet is called autonegotiation. It allows a station or a hub a range of capabilities. Autonegotiation allows two devices to negotiate the mode or data rate of operation. It was designed particularly for the following purposes:

- To allow incompatible devices to connect to one another. For example, a device with a maximum capacity of 10 Mbps can communicate with a device with a 100 Mbps capacity (but can work at a lower rate).
- To allow one device to have multiple capabilities.
- To allow a station to check a hub's capabilities.

B) PHYSICAL LAYER

The physical layer in Fast Ethernet is more complicated than the one in Standard Ethernet. We briefly discuss some features of this layer.

Topology

Fast Ethernet is designed to connect two or more stations together. If there are only two stations, they can be connected point-to-point. Three or more stations need to be connected in a star topology with a hub or a switch at the center.

Implementation

Fast Ethernet implementation at the physical layer can be categorized as either two-wire or four-wire. The two-wire implementation can be either category 5 UTP (100Base-TX) or fiber-optic cable (100Base-FX). The four-wire implementation is designed only for category 3 UTP (100Base-T4). See Fig. 3.21.

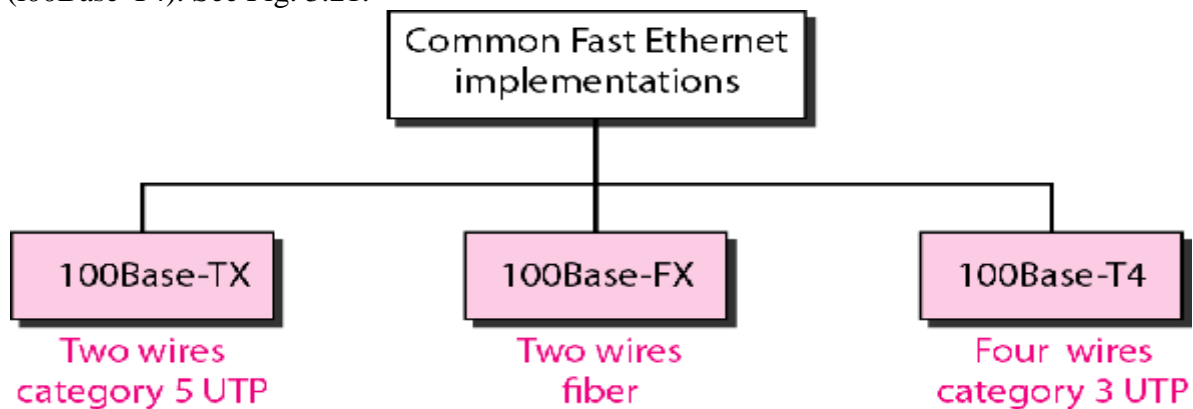


Fig. 3.21 Fast Ethernet implementations

Encoding

Manchester encoding needs a 200-Mbaud bandwidth for a data rate of 100 Mbps, which makes it unsuitable for a medium such as twisted-pair cable. For this reason, the Fast Ethernet designers sought some alternative encoding/decoding scheme. However, it was found that one scheme would not perform equally well for all three implementations. Therefore, three different encoding schemes were chosen. Table 3.2 is a summary of the Fast Ethernet implementations.

Table 3.2 Summary of Fast Ethernet implementations

<i>Characteristics</i>	<i>100Base-TX</i>	<i>100Base-FX</i>	<i>100Base-T4</i>
Media	Cat 5 UTP or STP	Fiber	Cat 4 UTP
Number of wires	2	2	4
Maximum length	100 m	100 m	100 m
Block encoding	4B/5B	4B/5B	
Line encoding	MLT-3	NRZ-I	8B/6T

3.4.4 GIGABIT ETHERNET

The need for an even higher data rate resulted in the design of the Gigabit Ethernet protocol (1000 Mbps). The IEEE committee calls the Standard 802.3z. 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 autonegotiation as defined in Fast Ethernet.

A) MAC SUBLAYER

A main consideration in the evolution of Ethernet was to keep the MAC sublayer untouched. However, to achieve a data rate 1 Gbps, this was no longer possible. Gigabit Ethernet has two distinctive approaches for medium access: half-duplex and full-duplex. Almost all implementations of Gigabit Ethernet follow the full-duplex approach. However, we briefly discuss the half-duplex approach to show that Gigabit Ethernet can be compatible with the previous generations.

Full-Duplex Mode

In full-duplex mode, there is a central switch connected to all computers or other switches. In this mode, each switch has buffers for each input port in which data are stored until they are transmitted. There is no collision in this mode, as we discussed before. This means that *CSMA/CD* is not used. Lack of collision implies that the maximum length of the cable is determined by the signal attenuation in the cable, not by the collision detection process.

Half-Duplex Mode

Gigabit Ethernet can also be used in half-duplex mode, although it is rare. In this case, a switch can be replaced by a hub, which acts as the common cable in which a collision might occur. The half-duplex approach uses *CSMA/CD*. However, as we saw before, the maximum length of the network in this approach is totally dependent on the minimum frame size. Three methods have been defined: traditional, carrier extension, and frame bursting. Traditional In the traditional approach, we keep the minimum length of the frame as in traditional Ethernet (512 bits). However, because the length of a bit is 11100 shorter in Gigabit Ethernet than in 10-Mbps Ethernet, the slot time for Gigabit Ethernet is $512 \text{ bits} \times 111000 \text{ JIS}$, which is equal to 0.512 JIS. The reduced slot time means that collision is detected 100 times earlier. This means that the maximum length of the network is 25 m. This length may be suitable if all the stations are in one room, but it may not even be long enough to connect the computers in one single office.

Carrier Extension To allow for a longer network, we increase the minimum frame length. The carrier extension approach defines the minimum length of a frame as 512 bytes (4096 bits). This means that the minimum length is 8 times longer. This method forces a station to add extension

bits (padding) to any frame that is less than 4096 bits. In this way, the maximum length of the network can be increased 8 times to a length of 200 m. This allows a length of 100 m from the hub to the station.

Frame Bursting Carrier extension is very inefficient if we have a series of short frames to send; each frame carries redundant data. To improve efficiency, frame bursting was proposed. Instead of adding an extension to each frame, multiple frames are sent. However, to make these multiple frames look like one frame, padding is added between the frames (the same as that used for the carrier extension method) so that the channel is not idle. In other words, the method deceives other stations into thinking that a very large frame has been transmitted.

B) PHYSICAL LAYER

The physical layer in Gigabit Ethernet is more complicated than that in Standard or Fast Ethernet. We briefly discuss some features of this layer.

Topology

Gigabit Ethernet is designed to connect two or more stations. If there are only two stations, they can be connected point-to-point. Three or more stations need to be connected in a star topology with a hub or a switch at the center. Another possible configuration is to connect several star topologies or let a star topology be part of another.

Implementation

Gigabit Ethernet can be categorized as either a two-wire or a four-wire implementation. The two-wire implementations use fiber-optic cable (1000Base-SX, short-wave, or 1000Base-LX, long-wave), or STP (1000Base-CX). The four-wire version uses category 5 twisted-pair cable (1000Base-T). In other words, we have four implementations, as shown in Figure 3.22. 1000Base-T was designed in response to those users who had already installed this wiring for other purposes such as Fast Ethernet or telephone services

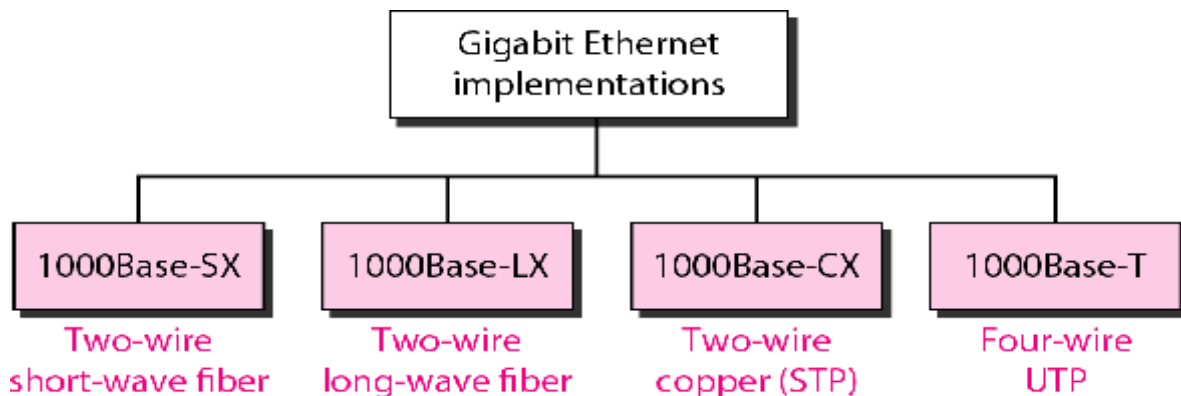


Fig. 3.22 Gigabit Ethernet implementations

Encoding

Gigabit Ethernet cannot use the Manchester encoding scheme because it involves a very high bandwidth (2 GBaud). The two-wire implementations use an NRZ scheme, but NRZ does not self-synchronize properly. To synchronize bits, particularly at this high data rate, 8B10B block encoding, is used. This block encoding prevents long sequences of Os or Is in the stream, but the resulting stream is 1.25 Gbps. Note that in this implementation, one wire (fiber or STP) is used for sending and one for receiving.

In the four-wire implementation it is not possible to have 2 wires for input and 2 for output, because each wire would need to carry 500 Mbps, which exceeds the capacity for category 5 UTP. As a solution, 4D-PAM5 encoding, is used to reduce the bandwidth. Thus, all four wires are involved in both input and output; each wire carries 250 Mbps, which is in the range for category 5 UTP cable. Table 3.3 is a summary of the Gigabit Ethernet implementations:

Table 3.2 Summary of Gigabit Ethernet implementations

<i>Characteristics</i>	<i>1000Base-SX</i>	<i>1000Base-LX</i>	<i>1000Base-CX</i>	<i>1000Base-T</i>
Media	Fiber short-wave	Fiber long-wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum length	550 m	5000 m	25 m	100 m
Block encoding	8B/10B	8B/10B	8B/10B	
Line encoding	NRZ	NRZ	NRZ	4D-PAM5

3.5 WIRELESS LAN

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. In this we 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. Although both protocols need several layers to operate, we concentrate mostly on the physical and data link layers.

3.5.1 IEEE 802.11

IEEE has defined the specifications for a wireless LAN, called IEEE 802.11, which covers the physical and data link layers.

A) ARCHITECTURE

The standard defines two kinds of services: the basic service set (BSS) and the extended service set (ESS).

Basic Service Set

IEEE 802.11 defines the basic service set (BSS) as the building block of a wireless LAN. A basic service set is made of stationary or mobile wireless stations and an optional central base station, known as the access point (AP). Fig. 3.23 shows two sets in this standard. The BSS without an AP is a stand-alone network and cannot send data to other BSSs. It is called an *ad hoc architecture*. In this architecture, stations can form a network without the need of an AP; they can locate one another and agree to be part of a BSS. A BSS with an AP is sometimes referred to as an *infrastructure network*.

BSS: Basic service set

AP: Access point

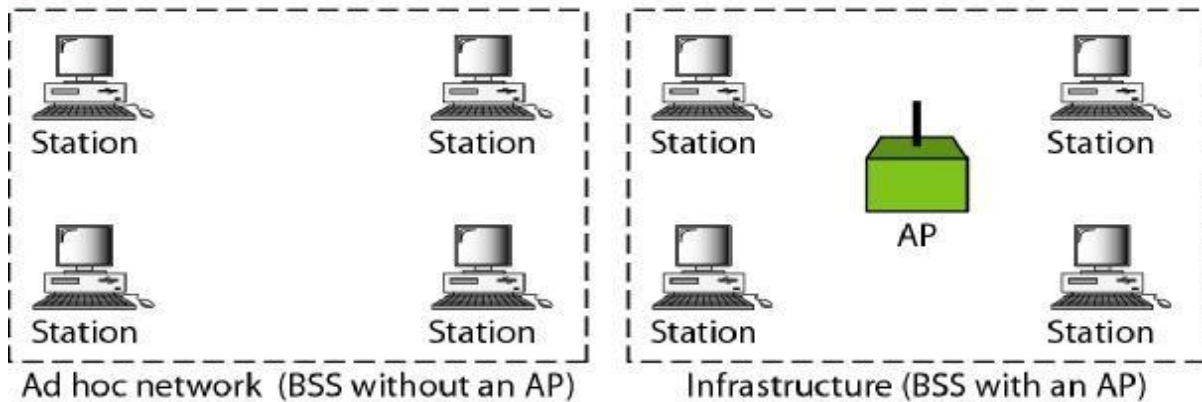


Fig. 3.23 Basic service sets (BSSs)

Extended Service Set

An extended service set (ESS) is made up of two or more BSSs with APs. In this case, the BSSs are connected through a *distribution system*, which is usually a wired LAN. The distribution system connects the APs in the BSSs. IEEE 802.11 does not restrict the distribution system; it can be any IEEE LAN such as an Ethernet. Note that the extended service set uses two types of stations: mobile and stationary. The mobile stations are normal stations inside a BSS. The stationary stations are AP stations that are part of a wired LAN. Fig. 3.24 shows an ESS.

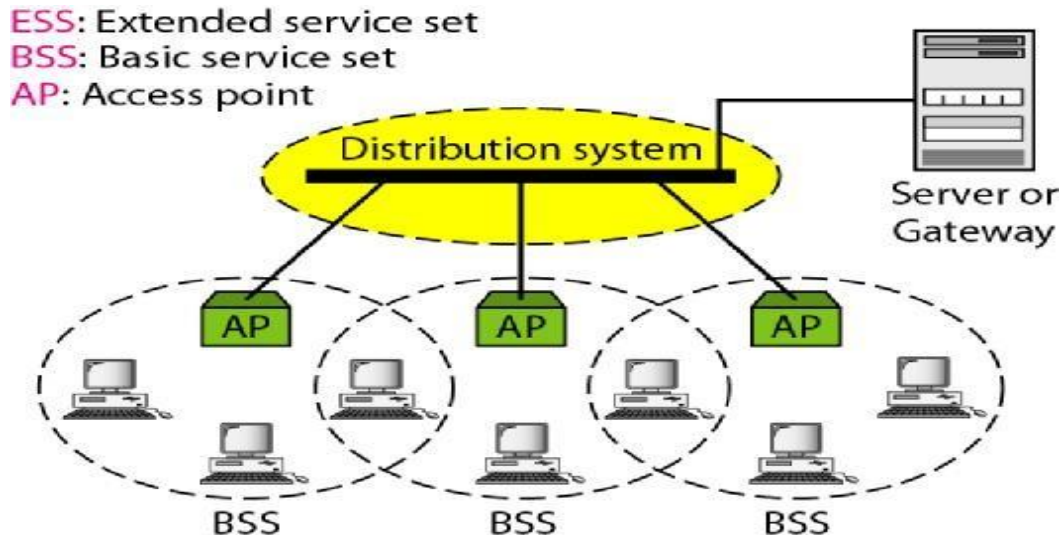


Fig. 3.24 Extended service sets (ESSs)

When BSSs are connected, the stations within reach of one another can communicate without the use of an AP. However, communication between two stations in two different BSSs usually occurs via two APs. The idea is similar to communication in a cellular network if we consider each BSS to be a cell and each AP to be a base station. Note that a mobile station can belong to more than one BSS at the same time. 802.11 and 802.11x refers to a family of specifications developed by the IEEE for *wireless LAN* (WLAN) technology. 802.11 specify an over-the-air interface between a wireless client and a base station or between two wireless clients. The IEEE accepted the specification in 1997.

There are several specifications in the 802.11 family:

- **802.11** — applies to wireless LANs and provides 1 or 2 Mbps transmission in the 2.4 GHz band using either frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS).
- **802.11a** — an extension to 802.11 that applies to wireless LANs and provides up to 54-Mbps in the 5GHz band. 802.11a uses an orthogonal frequency division multiplexing encoding scheme rather than FHSS or DSSS.
- **802.11b** (also referred to as 802.11 High Rate or Wi-Fi) — an extension to 802.11 that applies to wireless LANS and provides 11 Mbps transmission (with a fallback to 5.5, 2 and 1-Mbps) in the 2.4 GHz band. 802.11b uses only DSSS. 802.11b was a 1999 ratification to the original 802.11 standard, allowing wireless functionality comparable to Ethernet.
- **802.11e** — a wireless draft standard that defines the *Quality of Service* (QoS) support for LANs, and is an enhancement to the 802.11a and 802.11b wireless LAN (WLAN) specifications. 802.11e adds QoS features and multimedia support to the existing IEEE 802.11b and IEEE 802.11a wireless standards, while maintaining full backward compatibility with these standards.
- **802.11g** — applies to wireless LANs and is used for transmission over short distances at up to 54-Mbps in the 2.4 GHz bands.
- **802.11n** — 802.11n builds upon previous 802.11 standards by adding *multiple-input multiple-output* (MIMO). The additional transmitter and receiver antennas allow for

increased data throughput through spatial multiplexing and increased range by exploiting the spatial diversity through coding schemes like Alamouti coding. The real speed would be 100 Mbit/s (even 250 Mbit/s in PHY level), and so up to 4-5 times faster than 802.11g.

- **802.11ac** — 802.11ac builds upon previous 802.11 standards, particularly the 802.11n standard, to deliver data rates of 433Mbps per spatial stream, or 1.3Gbps in a three-antenna (three stream) design. The 802.11ac specification operates only in the 5 GHz frequency range and features support for wider channels (80MHz and 160MHz) and beam forming capabilities by default to help achieve its higher wireless speeds.
- **802.11ac Wave 2** — 802.11ac Wave 2 is an update for the original 802.11ac spec that uses MU-MIMO technology and other advancements to help increase theoretical maximum wireless speeds for the spec to 6.93 Gbps.
- **802.11ad** — 802.11ad is a wireless specification under development that will operate in the 60GHz frequency band and offer much higher transfer rates than previous 802.11 specs, with a theoretical maximum transfer rate of up to 7Gbps (Gigabits per second).
- **802.11r** - 802.11r, also called *Fast Basic Service Set* (BSS) Transition, supports VoWi-Fi handoff between access points to enable VoIP roaming on a Wi-Fi network with 802.1X authentication.
- **802.1X** — Not to be confused with 802.11x (which is the term used to describe the family of 802.11 standards) 802.1X is an IEEE standard for port-based Network Access Control that allows network administrators to restricted use of IEEE 802 LAN service access points to secure communication between authenticated and authorized devices.

3.5.2 Bluetooth IEEE 802.15.1

Bluetooth is a wireless LAN technology designed to connect devices of different functions such as telephones, notebooks, computers (desktop and laptop), cameras, printers, coffee makers, and so on. A Bluetooth LAN is an ad hoc network, which means that the network is formed spontaneously; the devices, sometimes called gadgets, find each other and make a network called a piconet. A Bluetooth LAN can even be connected to the Internet if one of the gadgets has this capability. A Bluetooth LAN, by nature, cannot be large. If there are many gadgets that try to connect, there is chaos. Bluetooth technology has several applications. Peripheral devices such as a wireless mouse or keyboard can communicate with the computer through this technology. Monitoring devices can communicate with sensor devices in a small health care center. Home security devices can use this technology to connect different sensors to the main security controller. Conference attendees can synchronize their laptop computers at a conference. Bluetooth was originally started as a project by the Ericsson Company. It is named for Harald Blaatand, the king of Denmark (940-981) who united Denmark and Norway. *Blaatand* translates to *Bluetooth* in English. Today, Bluetooth technology is the implementation of a protocol defined by the IEEE 802.15 standard. The standard defines a wireless personal-area network (PAN) operable in an area the size of a room or a hall.

A) ARCHITECTURE

Bluetooth defines two types of networks: piconet and scatternet.

Piconets

A Bluetooth network is called a piconet, or a small net. A piconet can have up to eight stations, one of which is called the primary the rest are called secondary. All the secondary stations synchronize their clocks and hopping sequence with the primary. Note that a piconet can have only one primary station. The communication between the primary and the secondary can be one-to-one or one-to-many. Fig. 3.25 shows a piconet.

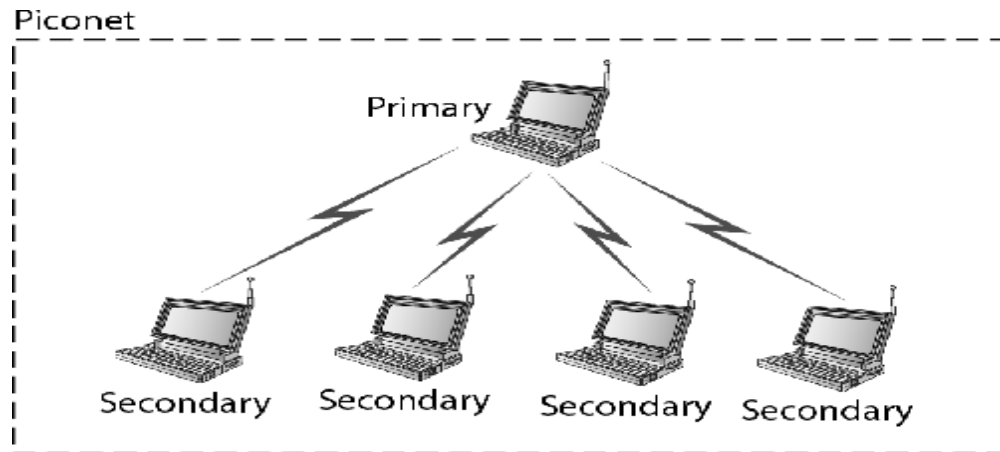


Fig. 3.25 Piconet

Although a piconet can have a maximum of seven secondaries, an additional eight secondaries can be in the *parked state*. A secondary in a parked state is synchronized with the primary, but cannot take part in communication until it is moved from the parked state. Because only eight stations can be active in a piconet, activating a station from the parked state means that an active station must go to the parked state.

Scatternet

Piconets can be combined to form what is called a scatternet. A secondary station in one piconet can be the primary in another piconet. This station can receive messages from the primary in the first piconet (as a secondary) and, acting as a primary, deliver them to secondaries in the second piconet. A station can be a member of two piconets. Fig. 3.26 illustrates a scatternet.

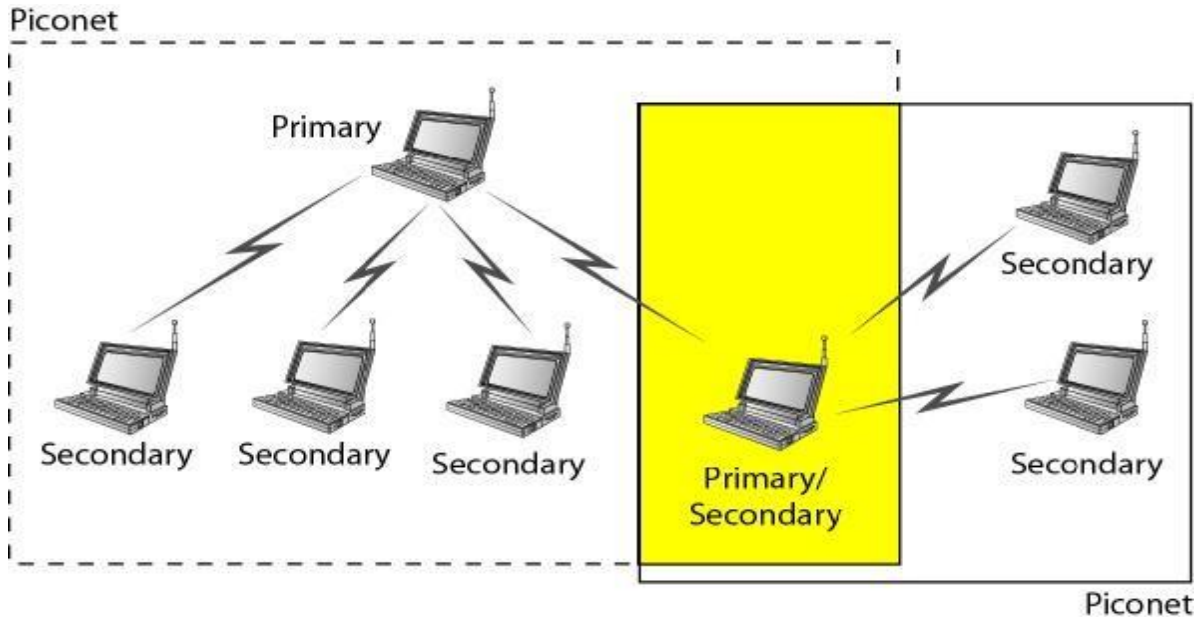


Fig. 3.26 Scatternet

B) BLUETOOTH LAYERS

Bluetooth uses several layers that do not exactly match those of the Internet model Fig. 3.27 shows these layers.

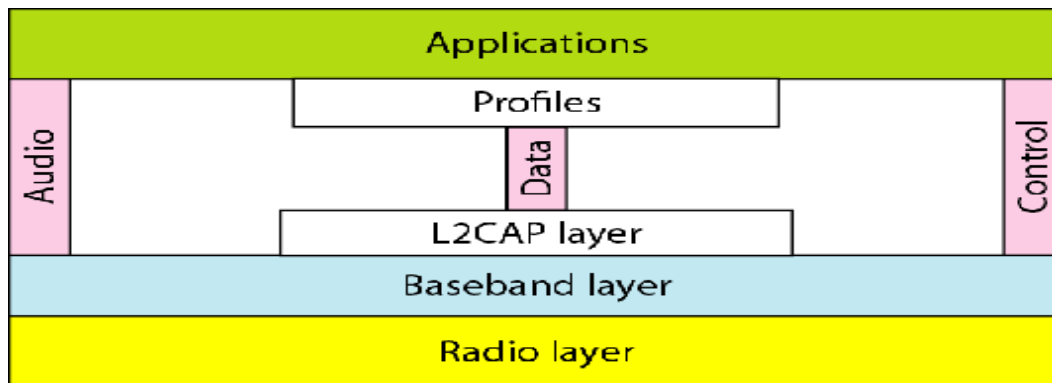


Fig. 3.27 Bluetooth layers

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. Bluetooth uses a 2.4-GHz ISM band divided into 79 channels of 1 MHz each. Bluetooth uses the frequency-hopping spread spectrum (FHSS) method in the physical layer to avoid interference from other devices or other networks. Bluetooth hops 1600 times per second, which means that each device changes its modulation frequency 1600 times per second. To transform bits to a signal, Bluetooth uses a sophisticated version of FSK, called GFSK.

Baseband Layer

The baseband layer is roughly equivalent to the MAC sublayer in LANs. The access method is TDMA. 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 sublayer in LANs. It is used for data exchange on an ACL link; SCQ channels do not use L2CAP. Figure 14.25 shows the format of the data packet at this level. The I6-bit length field 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 (see below). The L2CAP has specific duties: multiplexing, segmentation and reassembly, quality of service (QoS), and group management.

UNIT – IV

4.1 NETWORK LAYER

4.1 .1 DESIGN ISSUES:

Draw crude map. How to get from one host to another? If each link delivers reliably then is the whole route reliable a router may fail, limited but space (may have to throw packets on the door).Data Link Layer deals with machine-to-machine communication Network Layer lowest layer that deals with host-to-host communication, call this end-to-end communication.

Four issues:

1. Interface between the host and the network (the network layer is typically the boundary between the host and subnet)
2. Routing
3. Congestion and deadlock

When more packets enter an area than can be processed, delays increase and performance decreases. If the situation continues, the subnet may have no alternative but to discard packets.

- If the delay increases, the sender may (incorrectly) retransmit, making a bad situation even worse.
 - Overall, performance degrades because the network is using (wasting) resources processing packets that eventually get discarded.
4. Internetworking (A path may traverse different network technologies(e.g., ethernet, point-to-point links, etc.) packets may travel through many different networks each network may have a different frame format some networks may be connectionless, other connection oriented

4.1.2 ROUTING ALGORITHMS

A) NON-HIERARCHICAL ROUTING

In this type of routing, interconnected networks are viewed as a single network, where bridges, routers and gateways are just additional nodes.

- Every node keeps information about every other node in the network
- In case of adaptive routing, the routing calculations are done and updated for all the nodes.

The above two are also the disadvantages of non-hierarchical routing, since the table sizes and the routing calculations become too large as the networks get bigger. So this type of routing is feasible only for small networks.

B) HIERARCHICAL ROUTING

This is essentially a 'Divide and Conquer' strategy. The network is divided into different regions and a router for a particular region knows only about its own domain and other routers. Thus, the network is viewed at two levels:

1. The Sub-network level, where each node in a region has information about its peers in the same region and about the region's interface with other regions. Different regions may have different 'local' routing algorithms. Each local algorithm handles the traffic between

nodes of the same region and also directs the outgoing packets to the appropriate interface.

2. The Network Level, where each region is considered as a single node connected to its interface nodes. The routing algorithms at this level handle the routing of packets between two interface nodes, and is isolated from intra-regional transfer.

Networks can be organized in hierarchies of many levels; e.g. local networks of a city at one level, the cities of a country at a level above it, and finally the network of all nations.

In Hierarchical routing, the interfaces need to store information about:

- All nodes in its region which are at one level below it.
- Its peer interfaces.
- At least one interface at a level above it, for outgoing packages.

Advantages of Hierarchical Routing:

- Smaller sizes of routing tables.
- Substantially lesser calculations and updates of routing tables.

Disadvantage:

- Once the hierarchy is imposed on the network, it is followed and possibility of direct paths is ignored. This may lead to sub optimal routing.

C) SOURCE ROUTING

Source routing is similar in concept to virtual circuit routing. It is implemented as under:

- Initially, a path between nodes wishing to communicate is found out, either by flooding or by any other suitable method.
- This route is then specified in the header of each packet routed between these two nodes. A route may also be specified partially, or in terms of some intermediate hops.

Advantages:

- Bridges do not need to look up their routing tables since the path is already specified in the packet itself.
- The throughput of the bridges is higher, and this may lead to better utilization of bandwidth, once a route is established.

Disadvantages:

- Establishing the route at first needs an expensive search method like flooding.
- To cope up with dynamic relocation of nodes in a network, frequent updates of tables are required; else all packets would be sent in wrong direction. This too is expensive.

D) POLICY BASED ROUTING

In this type of routing, certain restrictions are put on the type of packets accepted and sent. e.g.. The IIT- K router may decide to handle traffic pertaining to its departments only, and reject packets from other routes. This kind of routing is used for links with very low capacity or for security purposes.

E) SHORTEST PATH ROUTING

Here, the central question dealt with is 'How to determine the optimal path for routing ?' Various algorithms are used to determine the optimal routes with respect to some predetermined criteria. A network is represented as a graph, with its terminals as nodes and the links as edges. A 'length' is associated with each edge, which represents the cost of using the link for transmission. Lower the cost, more suitable is the link. The cost is determined depending upon the criteria to be optimized. Some of the important ways of determining the cost are:

- **Minimum number of hops:** If each link is given a unit cost, the shortest path is the one with minimum number of hops. Such a route is easily obtained by a breadth first search method. This is easy to implement but ignores load, link capacity etc.
- **Transmission and Propagation Delays:** If the cost is fixed as a function of transmission and propagation delays, it will reflect the link capacities and the geographical distances. However these costs are essentially static and do not consider the varying load conditions.
- **Queuing Delays:** If the cost of a link is determined through its queuing delays, it takes care of the varying load conditions, but not of the propagation delays.

Ideally, the cost parameter should consider all the above mentioned factors, and it should be updated periodically to reflect the changes in the loading conditions. However, if the routes are changed according to the load, the load changes again. This feedback effect between routing and load can lead to undesirable oscillations and sudden swings.

ROUTING ALGORITHMS

As mentioned above, the shortest paths are calculated using suitable algorithms on the graph representations of the networks. Let the network be represented by graph $G (V, E)$ and let the number of nodes be 'N'. For all the algorithms discussed below, the costs associated with the links are assumed to be positive. A node has zero cost w.r.t itself. Further, all the links are assumed to be symmetric, i.e. if $d_{i,j}$ = cost of link from node i to node j, then $d_{i,j} = d_{j,i}$. The graph is assumed to be complete. If there exists no edge between two nodes, then a link of infinite cost is assumed. The algorithms given below find costs of the paths from all nodes to a particular node; the problem is equivalent to finding the cost of paths from a source to all destinations.

a) Bellman-Ford Algorithm

This algorithm iterates on the number of edges in a path to obtain the shortest path. Since the number of hops possible is limited (cycles are implicitly not allowed), the algorithm terminates giving the shortest path.

Notation:

- $d_{i,j}$ = Length of path between nodes i and j, indicating the cost of the link.
h = Number of hops.

$D[i, h]$ = Shortest path length from node i to node 1, with upto ' h ' hops.
 $D[1, h] = 0$ for all h .

Algorithm :

Initial condition : $D[i, 0] = \text{infinity}$, for all i ($i \neq 1$)
Iteration : $D[i, h+1] = \min \{ d_{i,j} + D[j, h] \}$ over all values of j .
Termination : The algorithm terminates when
 $D[i, h] = D[i, h+1]$ for all i .

Principle:

For zero hops, the minimum length path has length of infinity, for every node. For one hop the shortest-path length associated with a node is equal to the length of the edge between that node and node 1. Hereafter, we increment the number of hops allowed, (from h to $h+1$) and find out whether a shorter path exists through each of the other nodes. If it exists, say through node ' j ', then its length must be the sum of the lengths between these two nodes (i.e. $d_{i,j}$) and the shortest path between j and 1 obtainable in upto h paths. If such a path doesn't exist, then the path length remains the same. The algorithm is guaranteed to terminate, since there are utmost N nodes, and so $N-1$ paths. It has time complexity of $O(N^3)$.

b) Dijkstra's Algorithm

Notation:

D_i = Length of shortest path from node ' i ' to node 1.
 $d_{i,j}$ = Length of path between nodes i and j .

Algorithm

Each node j is labeled with D_j , which is an estimate of cost of path from node j to node 1. Initially, let the estimates be infinity, indicating that nothing is known about the paths. We now iterate on the length of paths, each time revising our estimate to lower values, as we obtain them. Actually, we divide the nodes into two groups; the first one, called set P contains the nodes whose shortest distances have been found, and the other Q containing all the remaining nodes. Initially P contains only the node 1. At each step, we select the node that has minimum cost path to node 1. This node is transferred to set P . At the first step, this corresponds to shifting the node closest to 1 in P . Its minimum cost to node 1 is now known. At the next step, select the next closest node from set Q and update the labels corresponding to each node using:

$$D_j = \min [D_j, D_i + d_{j,i}]$$

Finally, after $N-1$ iterations, the shortest paths for all nodes are known, and the algorithm terminates.

Principle

Let the closest node to 1 at some step be i . Then i is shifted to P . Now, for each node j , the

closest path to 1 either passes through i or it doesn't. In the first case D_j remains the same. In the second case, the revised estimate of D_j is the sum $D_i + d_{i,j}$. So we take the minimum of these two cases and update D_j accordingly. As each of the nodes get transferred to set P, the estimates get closer to the lowest possible value. When a node is transferred, its shortest path length is known. So finally all the nodes are in P and the D_j 's represent the minimum costs. The algorithm is guaranteed to terminate in $N-1$ iterations and its complexity is $O(N^2)$.

c) The Floyd Warshall Algorithm

This algorithm iterates on the set of nodes that can be used as intermediate nodes on paths. This set grows from a single node (say node 1) at start to finally all the nodes of the graph. At each iteration, we find the shortest path using given set of nodes as intermediate nodes, so that finally all the shortest paths are obtained.

Notation

$D_{i,j}[n]$ = Length of shortest path between the nodes i and j using only the nodes 1,2,..n as intermediate nodes.

Initial Condition

$D_{i,j}[0] = d_{i,j}$ for all nodes i,j .

Algorithm

Initially, $n = 0$. At each iteration, add next node to n. i.e. For $n = 1,2,\dots,N-1$,

$D_{i,j}[n+1] = \min \{ D_{i,j}[n], D_{i,n+1}[n] + D_{n+1,j}[n] \}$

Principle

Suppose the shortest path between i and j using nodes 1,2,..n is known. Now, if node n+1 is allowed to be an intermediate node, then the shortest path under new conditions either passes through node n+1 or it doesn't. If it does not pass through the node n+1, then $D_{i,j}[n+1]$ is same as $D_{i,j}[n]$. Else, we find the cost of the new route, which is obtained from the sum, $D_{i,n+1}[n] + D_{n+1,j}[n]$. So we take the minimum of these two cases at each step. After adding all the nodes to the set of intermediate nodes, we obtain the shortest paths between all pairs of nodes together. The complexity of Floyd-Warshall algorithm is $O(N^3)$.

It is observed that all the three algorithms mentioned above give comparable performance, depending upon the exact topology of the network.

4.1.3 CONGESTION CONTROL ALGORITHMS

An important issue in a packet-switched network 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 internetwork occurs because routers and switches have queues-buffers that hold the packets before and after processing. A router, for example, has an input queue and an output queue for each interface. When a packet arrives at the incoming interface, it undergoes three steps before departing, as shown in Fig. 4.1

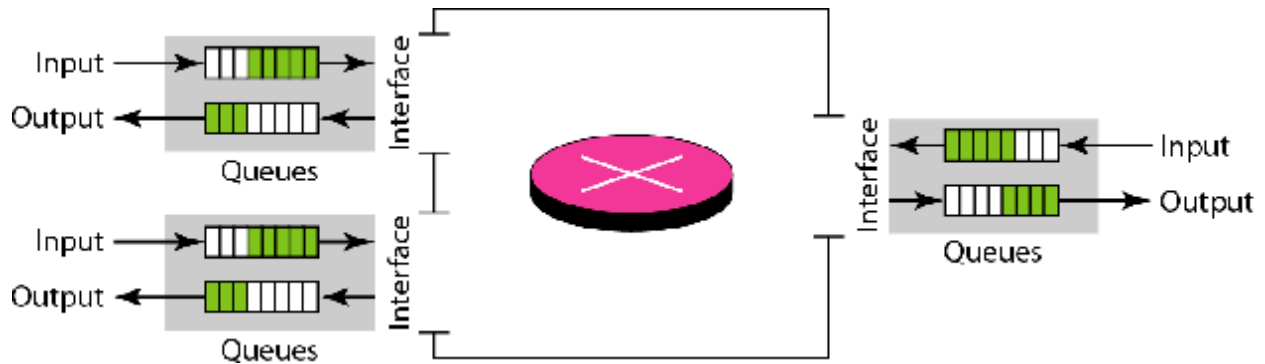


Fig. 4.1 Queue in Router

1. The packet is put at the end of the input queue while waiting to be checked.
2. The processing module of the router removes the packet from the input queue once it reaches the front of the queue and uses its routing table and the destination address to find the route.
3. The packet is put in the appropriate output queue and waits its turn to be sent.

We need to be aware of two issues. First, if the rate of packet arrival is higher than the packet processing rate, the input queues become longer and longer. Second, if the packet departure rate is less than the packet processing rate, the output queues become longer and longer.

CONGESTION CONTROL

Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. In general, we can divide congestion control mechanisms into two broad categories: open-loop congestion control (prevention) and closed-loop congestion control (removal) as shown in Fig. 4.2

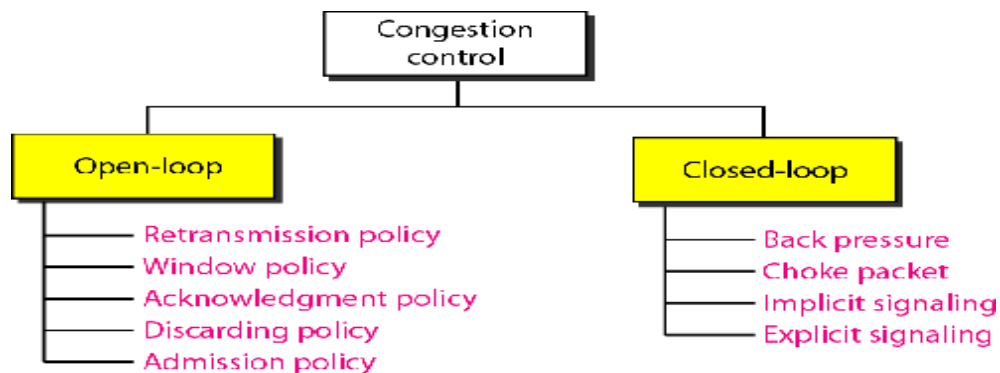


Fig. 4.2

A) 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. We give a brief list of policies that can prevent congestion.

a) 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. For example, the retransmission policy used by TCP (explained later) is designed to prevent or alleviate congestion.

b) Window Policy

The type of window at the sender may also affect congestion. The Selective Repeat window is better than the Go-Back-N window for congestion control. In the *Go-Back-N* window, when the timer for a packet times out, several packets may be resent, although some may have arrived safe and sound at the receiver. This duplication may make the congestion worse. The Selective Repeat window, on the other hand, tries to send the specific packets that have been lost or corrupted.

c) 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 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 fewer loads on the network.

d) Discarding Policy

A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission. For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented or alleviated.

e) Admission Policy

An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual-circuit networks. Switches in a flow first check the resource requirement of a flow before admitting it to the network. A router can deny establishing a virtual circuit connection if there is congestion in the network or if there is a possibility of future congestion.

B) CLOSED-LOOP CONGESTION CONTROL

Closed-loop congestion control mechanisms try to alleviate congestion after it happens.

Several mechanisms have been used by different protocols.

a) Backpressure

The technique of *backpressure* refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes. This may cause the upstream node or nodes to become congested, and they, in turn, reject data from their upstream nodes or nodes. And so on. Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source. The backpressure technique can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is coming. Fig. 4.3 shows the idea of backpressure.

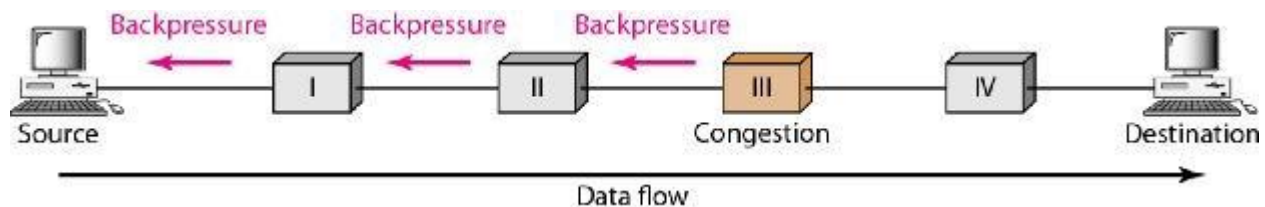


Fig. 4.2 Backpressure method for alleviating congestion

Node III in the figure has more input data than it can handle. It drops some packets in its input buffer and informs node II to slow down. Node II, in turn, may be congested because it is slowing down the output flow of data. If node II is congested, it informs node I to slow down, which in turn may create congestion. If so, node I inform the source of data to slow down. This, in time, alleviates the congestion. Note that the *pressure* on node III is moved backward to the source to remove the congestion. It was, however, implemented in the first virtual-circuit network, X.25. The technique cannot be implemented in a datagram network because in this type of network, a node (router) does not have the slightest knowledge of the upstream router.

b) Choke Packet

A choke packet is a packet sent by a node to the source to inform it of congestion. Note the difference between the backpressure and choke packet methods. In backpressure, the warning is from one node to its upstream node, although the warning may eventually reach the source station. In the choke packet method, the warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet has traveled are not warned. We have seen an example of this type of control in ICMP. When a router in the Internet is over-whelmed with IP datagram's, it may discard some of them; but it informs the source host, using a source quench ICMP message. The warning message goes directly to the source station; the intermediate routers, and does not take any action. Fig. 4.4 shows the idea of a choke packet.

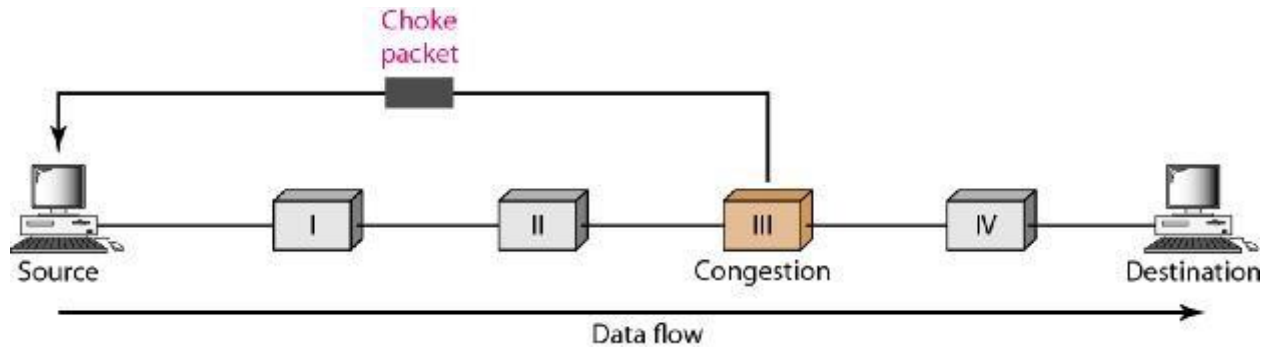


Fig. 4.4 Choke packet

c) Implicit Signaling

In implicit signaling, there is no communication between the congested node or nodes and the source. The source guesses that there is congestion somewhere in the network from other symptoms. For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested. The delay in receiving an acknowledgment is interpreted as congestion in the network; the source should slow down.

d) Explicit Signaling

The node that experiences congestion can explicitly send a signal to the source or destination. The explicit signaling method, however, is different from the choke packet method. In the choke packet method, a separate packet is used for this purpose; in the explicit signaling method, the signal is included in the packets that carry data. Explicit signaling, as we will see in Frame Relay congestion control, can occur in either the forward or the backward direction.

e) Backward Signaling

A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.

f) Forward Signaling

A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can use policies, such as slowing down the acknowledgments, to alleviate the congestion.

4.2 IPv4 ADDRESSING

An IPv4 address is a 32-bit address that *uniquely* and *universally* defines the connection of a device (for example, a computer or a router) to the Internet. IPv4 addresses are unique. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address at the same time. By using some strategies, an address may be assigned to a device for a time period and then taken away and assigned to another device. On the other hand, if a device operating at the network layer has m connections to the Internet, it needs to have m addresses. The IPv4 addresses are universal in the

sense that the addressing system must be accepted by any host that wants to be connected to the Internet.

ADDRESS SPACE

A protocol such as IPv4 that defines addresses has an address space. An address space is the total number of addresses used by the protocol. If a protocol uses N bits to define an address, the address space is 2^N because each bit can have two different values (0 or 1) and N bits can have 2^N values. IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than 4 billion). This means that, theoretically, if there were no restrictions, more than 4 billion devices could be connected to the Internet. We will see shortly that the actual number is much less because of the restrictions imposed on the addresses.

Notations

There are two prevalent notations to show an IPv4 address: binary notation and dotted decimal notation.

Binary Notation

In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte. So it is common to hear an IPv4 address referred to as a 32-bit address or a 4-byte address. The following is an example of an IPv4 address in binary notation:

01110101 10010101 00011101 00000010

Dotted-Decimal Notation

To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. The following is the dotted~decimal notation of the above address:

117.149.29.2

Fig. 4.5 shows an IPv4 address in both binary and dotted-decimal notation. Note that because each byte (octet) is 8 bits, each number in dotted-decimal notation is a value ranging from 0 to 255.

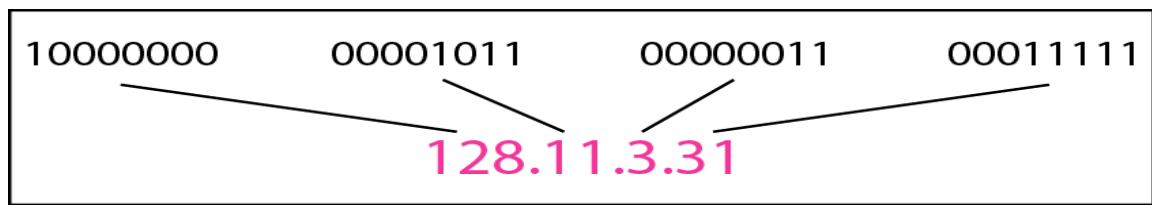


Fig. 4.5 Dotted-decimal notation and binary notation for an IPv4 address

A) CLASSFUL ADDRESSING

IPv4 addressing, at its inception, used the concept of classes. This architecture is called classful addressing. Although this scheme is becoming obsolete, we briefly discuss it here to show the

rationale behind classless addressing. In classful addressing, the address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the address space.

We can find the class of an address when given the address in binary notation or dotted-decimal notation. If the address is given in binary notation, the first few bits can immediately tell us the class of the address. If the address is given in decimal-dotted notation, the first byte defines the class. Both methods are shown in Fig. 4.6

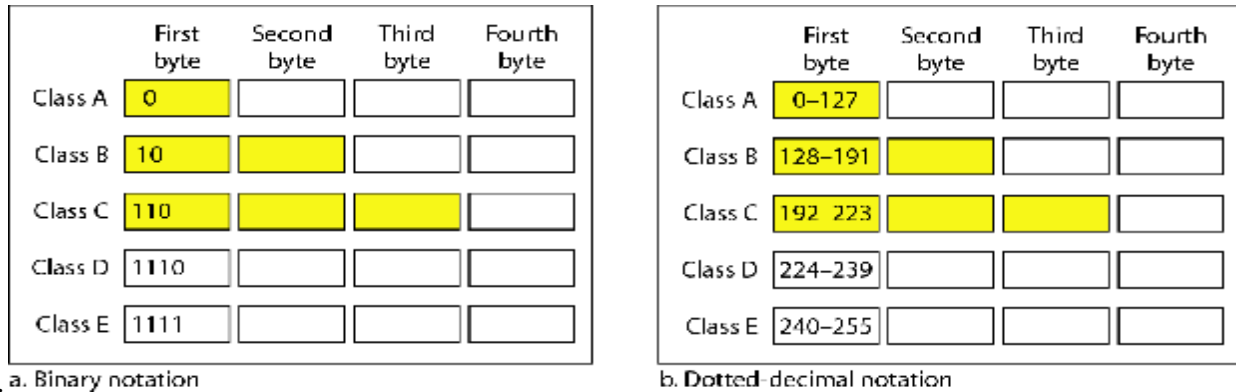


Fig. 4.6 Finding the classes in binary and dotted-decimal notation

Classes and Blocks

One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size as shown in Table 4.1.

Table 4.1 Number of blocks and block size in classful IPv4 addressing

Class	Number of Blocks	Block Size	Application
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Let us examine the table. Previously, when an organization requested a block of addresses, it was granted one in class A, B, or C. Class A addresses were designed for large organizations with a large number of attached hosts or routers. Class B addresses were designed for midsize organizations with tens of thousands of attached hosts or routers. Class C addresses were designed for small organizations with a small number of attached hosts or routers.

We can see the flaw in this design. A block in class A address is too large for almost any organization. This means most of the addresses in class A were wasted and were not used. A block in class B is also very large, probably too large for many of the organizations that received a class B block. A block in class C is probably too small for many organizations. Class D addresses were designed for multicasting. Each address in this class is used to define one group

of hosts on the Internet. The Internet authorities wrongly predicted a need for 268,435,456 groups. This never happened and many addresses were wasted here too. And lastly, the class E addresses were reserved for future use; only a few were used, resulting in another waste of addresses.

Netid and Hostid

In classful addressing, an IP address in class A, B, or C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address. Fig. 4.6 shows some netid and hostid bytes. The netid is in color, the hostid is in white. Note that the concept does not apply to classes D and E. In class A, one byte defines the netid and three bytes define the hostid. In class B, two bytes define the netid and two bytes define the hostid. In class C, three bytes define the netid and one byte defines the hostid.

Mask

Although the length of the netid and hostid (in bits) is predetermined in classful addressing, we can also use a mask (also called the default mask), a 32-bit number made of contiguous 1s followed by contiguous 0s. The masks for classes A, B, and C are shown in Table 4.2. The concept does not apply to classes D and E

Table 4.2 Default masks for classful addressing

<i>Class</i>	<i>Binary</i>	<i>Dotted-Decimal</i>	<i>CIDR</i>
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

The mask can help us to find the netid and the hostid. For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid. The last column of Table 4.2 shows the mask in the form *In* where *n* can be 8, 16, or 24 in classful addressing. This notation is also called slash notation or Classless Interdomain Routing (CIDR) notation. The notation is used in classless addressing, which we will discuss later. We introduce it here because it can also be applied to classful addressing.

Subnetting

During the era of classful addressing, subnetting was introduced. If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called subnets) or, in rare cases, share part of the addresses with neighbors. Subnetting increases the number of 1s in the mask,

Supernetting

The time came when most of the class A and class B addresses were depleted; however, there was still a huge demand for midsize blocks. The size of a class C block with a maximum number

of 256 addresses did not satisfy the needs of most organizations. Even a midsize organization needed more addresses. One solution was supernetting. In supernetting, an organization can combine several class C blocks to create a larger range of addresses. In other words, several networks are combined to create a supernet or a supernet. An organization can apply for a set of class C blocks instead of just one. For example, an organization that needs 1000 addresses can be granted four contiguous class C blocks. The organization can then use these addresses to create one supernet. Supernetting decreases the number of 1s in the mask. For example, if an organization is given four class C addresses, the mask changes from /24 to /22.

Address Depletion

The flaws in classful addressing scheme combined with the fast growth of the Internet led to the near depletion of the available addresses. Yet the number of devices on the Internet is much less than the 232 address space. We have run out of class A and B addresses, and a class C block is too small for most midsize organizations. One solution that has alleviated the problem is the idea of classless addressing.

B) CLASSLESS ADDRESSING

To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented. In this scheme, there are no classes, but the addresses are still granted in blocks.

Address Blocks

In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block (the number of addresses) varies based on the nature and size of the entity. For example, a household may be given only two addresses; a large organization may be given thousands of addresses. An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve. Restriction To simplify the handling of addresses, the Internet authorities impose three restrictions on classless address blocks:

1. The addresses in a block must be contiguous, one after another.
2. The number of addresses in a block must be a power of 2 (1, 2, 4, 8, ...).
3. The first address must be evenly divisible by the number of addresses.

Network Addresses

A very important concept in IP addressing is the network address. When an organization is given a block of addresses, the organization is free to allocate the addresses to the devices that need to be connected to the Internet. The first address in the class, however, is normally (not always) treated as a special address. The first address is called the network address and defines the organization network. It defines the organization itself to the rest of the world.

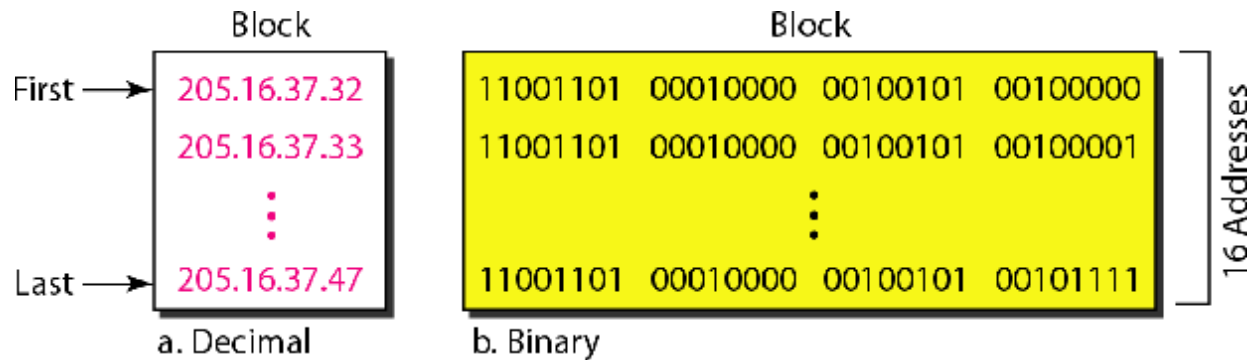


Fig. 4.7 A network configuration for the block 205.16.37.32/28

The organization network is connected to the Internet via a router. The router has two addresses. One belongs to the granted block; the other belongs to the network that is at the other side of the router. We call the second address $x.y.z.t/n$ because we do not know anything about the network it is connected to at the other side. All messages destined for addresses in the organization block (205.16.37.32 to 205.16.37.47) are sent, directly or indirectly, to $x.y.z.t/n$. We say directly or indirectly because we do not know the structure of the network to which the other side of the router is connected.

Address Allocation

The next issue in classless addressing is address allocation. How are the blocks allocated? The ultimate responsibility of address allocation is given to a global authority called the *Internet Corporation for Assigned Names and Addresses* (ICANN). However, ICANN does not normally allocate addresses to individual organizations. It assigns a large block of addresses to an ISP. Each ISP, in turn, divides its assigned block into smaller subblocks and grants the subblocks to its customers. In other words, an ISP receives one large block to be distributed to its Internet users. This is called address aggregation: many blocks of addresses are aggregated in one block and granted to one ISP.

4.3 CONNECTING DEVICES

The communication media used to link devices to form a computer network include electrical cable (HomePNA, power line communication, G.hn), optical fiber (fiber-optic communication), and radio waves (wireless networking). In the OSI model, these are defined at layers 1 and 2 — the physical layer and the data link layer.

A widely adopted *family* of communication media used in local area network (LAN) technology is collectively known as Ethernet. The media and protocol standards that enable communication between networked devices over Ethernet are defined by IEEE 802.3. Ethernet transmit data over both copper and fiber cables. Wireless LAN standards (e.g. those defined by IEEE 802.11) use radio waves, or others use infrared signals as a transmission medium. Power line communication uses a building's power cabling to transmit data.

Various devices are used to connect network of a computer The most common devices are:

A) Routers

- B) Gateways
- C) Repeaters
- D) Bridges
- E) Hub
- F) Modem

A) ROUTER

Routers are devices which connect two or more networks that use similar protocol. A router consists of hardware and software. Hardware can be a computer or a specific device. Software consists of a special management program that controls the flow of data between networks. Routers operate at a network layer of the OSI model. Routers use logical and physical addresses to connect two or more logically separate networks. They make this connection by organizing the large network into logical network segments (sometimes small sub-networks or sub-nets). Each of these sub-nets is given a logical address. Data is grouped into packets or blocks of data. Each packet, in addition to having a physical device address, has a logical address. The network address allows routers to calculate more accurately and efficiently the path of the computer.

Advantages of Router

- They use a high level of intelligence to route data
- Routers can also act as a bridge to handle non-routable protocols such as NetBEUI (Network BIOS Extended User Interface)

Disadvantages of Router:

- High level of intelligence takes more processing time which can affect performance
- Routers are very complicated which makes installation and maintenance difficult.

B) GATEWAYS

Gateways are devices which connect two or more networks that use different protocols. They are similar in function to routers but they are more powerful and intelligent devices. A gateway can actually convert data so that a network with an application on a computer on the other side of the gateway, e.g., a gateway can receive email messages in one format and convert them into another format. A gateway can operate at all seven layers of the OSI model. Since gateways perform data conversion, they are slower in speed and very expensive devices.

C) REPEATERS

Repeaters are used within a network to extend the length of communication. Data processes through transmission media in the form of waves or signals. The transmission media weakens signals that move through it. The weakening of a signal is called attenuation. If the data is to be transmitted beyond the maximum length of a communication media, signals are amplified. The devices used to amplify the signals are called repeaters. Repeaters work at the physical layer of the OSI model. Repeaters are normally two-port boxes that connect two segments. As a signal comes in

one port, it is Regenerated and send out to the other port. The signal is read as 1s and 0s. As 1s and 0s are transmitted, the noise can be cleaned out.

Advantages of Repeater

- Repeaters easily extend the length of network.
- They require no processing overhead, so very little if any performance degradation occurs.
- It can connect signals from the same network type that use different types of cables.

Disadvantages of Repeaters

- Repeaters can not be used to connect segments of different network types.
- They cannot be used to segment traffic on a network to reduce congestion .
- Many types of network have a limit on the number of network s that can be used at once .

D) BRIDGES

Bridges are used to connect similar network segments. A bridge does not pass or signals it receives. When a bridge receive a signal, it determines its destination by looking at its destination and it sends the signals towards it. For example in a above figure a bridge has been used to join two network segments A AND B. When the bridge receives the signals it read address of both sender and receiver. If the sender is a computer in segment A and the receiver is also segment A, it would not pass the signals to the segments B. It will however pass signals if the sender is in one segment and the receiver in other segment. Bridge works at the data link layer of O.S.I model.

Advantages of Bridges

- Bridge extends network segments by connecting them together to make one logical network.
- They can affect the segment traffic between networks by filtering data if it does not need to pass.
- Like repeaters they can connect similar network types with different cabling.

Disadvantages of Bridges

- Bridge possess information about the data they receive with can slow performance.

E) HUB

Hubs are basically multi ports repeaters for U.T.P cables. Some hubs have ports for other type of cable such as coaxial cable. Hubs range in size from four ports up to and for specific to the network types. These are some hubs which are

I. Passive Hub

II. Active Hub

III. Switch/ Intelligent Hub

I Passive Hub

It provides no signal regeneration. They are simply cables connected together so that the signal is broken out to other nodes without regeneration. These are not used often today because of loss of cable length that is allowed.

II Active Hub

It acts as repeaters and regenerates the data signals to all ports. They have no real intelligence to tell whether the signal needs to go to all ports that is blindly repeated.

III Switch Hub

Switches are multi ports bridges. They filter traffic between the ports on the switch by using the address of computers transmitting to them.

Switches can be used when data performance is needed or when collision need to be reduce.

Advantages of Hub

- Hubs need almost no configuration.
- Active hub can extend maximum network media distance.

No processing is done at the hub to slow down performance_

Disadvantages of Hub

- Passive hubs can greatly limit maximum media distance.
- Hubs have no intelligence to filter traffic so all data is send out on all ports whether it is need or not.

Since hubs can act as repeaters the network using them must follow the same rules as repeaters

F) MODEM

The device that converts digital signals into analog signals and analog signals to digital signals is called Modem. The word modem stands for modulation and demodulation. The process of converting digital signals to analog signals is called modulation. The process of converting analog signals to digital signals is called demodulation. Modems are used with computers to transfer data from one computer to another computer through telephone lines.

Modems have two connections these are.

I Analog connection

II Digital connection

I Analog connection.

The connection between the modem and the telephone line is called analog connection.

Types of Modem

THERE ARE TWO TYPES OF MODEMS

- Internal modem
- External modem

Digital connection.

The connection of modem to computer is called digital connection

INTERNAL MODEM

It fits into expansion slots inside the computer. It is directly linked to the telephone lines through the telephone jack. It is normally less expensive than external modem. Its transmission speed is also less external modem.

EXTERNAL MODEM

It is the external unit of computer and is connected to the computer through serial port. It is also linked to the telephone line through a telephone jack. External modems are expensive and have more operation features and high transmission speed.

Advantages of Modem

- Inexpensive hardware and telephone lines.
- Easy to setup and maintain.

Disadvantages of Modem

- Very slow performance.

4.4 Virtual LAN IPV6 Addresses

Despite all short-term solutions, such as classless addressing, Dynamic Host Configuration Protocol (DHCP), and NAT, address depletion is still a long-term problem for the Internet. This and other problems in the IP protocol itself such as lack of accommodation for real-time audio and video transmission, and encryption and authentication of data for some applications, have been the motivation for IPv6.

Structure

An IPv6 address consists of 16 bytes (octets); it is 128 bits long.

Hexadecimal Colon Notation

To make addresses more readable, IPv6 specifies hexadecimal colon notation. In this notation,

128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon, as shown in Fig. 4.8.

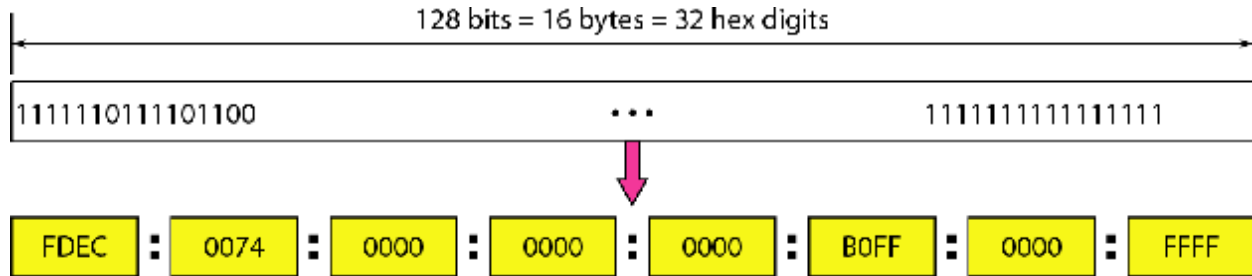


Fig. 4.8 IPv6 address in binary and hexadecimal colon notation

Abbreviation

Although the IP address, even in hexadecimal format, is very long, many of the digits are zeros. In this case, we can abbreviate the address. The leading zeros of a section (four digits between two colons) can be omitted. Only the leading zeros can be dropped, not the trailing zeros (see Fig. 4.9).

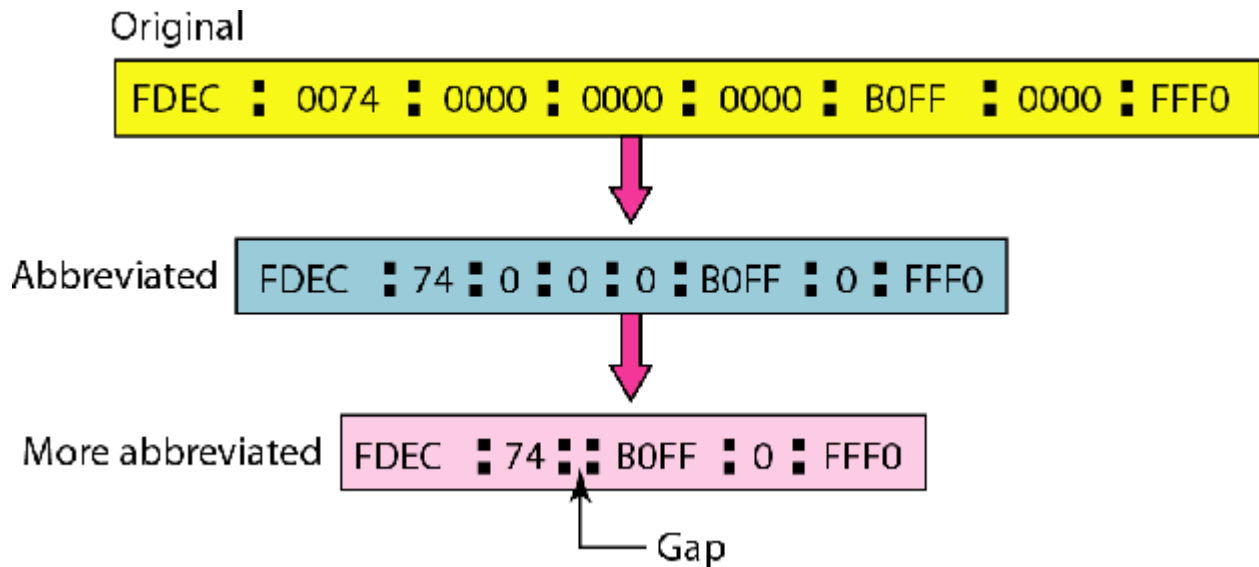


Fig. 4.9 Abbreviated IPv6 addresses

Using this form of abbreviation, 0074 can be written as 74, 000F as F, and 0000 as O. Note that 3210 cannot be abbreviated. Further abbreviations are possible if there are consecutive sections consisting of zeros only. We can remove the zeros altogether and replace them with a double semicolon. Note that this type of abbreviation is allowed only once per address. If there are two runs of zero sections, only one of them can be abbreviated. Re expansion of the abbreviated address is very simple: Align the unabbreviated portions and insert zeros to get the original expanded address.

Address Space

IPv6 has a much larger address space; 2¹²⁸ addresses are available. The designers of IPv6 divided the address into several categories. A few leftmost bits, called the *type prefix*, in each address define its category. The type prefix is variable in length, but it is designed such that no code is identical to the first part of any other code. In this way, there is no ambiguity; when an address is given, the type prefix can easily be determined. Table 4.3 shows the prefix for each type of address. The third column shows the fraction of each type of address relative to the whole address space.

Table 4.3 Type prefixes for IPv6 addresses

<i>Type Prefix</i>	<i>Type</i>	<i>Fraction</i>
0000 0000	Reserved	1/256
0000 0001	Unassigned	1/256
0000 001	ISO network addresses	1/128
0000 010	IPX (Novell) network addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Reserved	1/16
001	Reserved	1/8
010	Provider-based unicast addresses	1/8

<i>Type Prefix</i>	<i>Type</i>	<i>Fraction</i>
011	Unassigned	1/8
100	Geographic-based unicast addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 1110 0	Unassigned	1/512
1111 1110 10	Link local addresses	1/1024
1111 1110 11	Site local addresses	1/1024
1111 1111	Multicast addresses	1/256

A) UNICAST ADDRESSES

A **unicast address** defines a single computer. The packet sent to a unicast address must be delivered to that specific computer. IPv6 defines two types of unicast addresses: geographically based and provider-based. We discuss the second type here; the first type is left for future definition. The provider-based address is generally used by a normal host as a unicast address. The address format is shown in Fig. 4.10

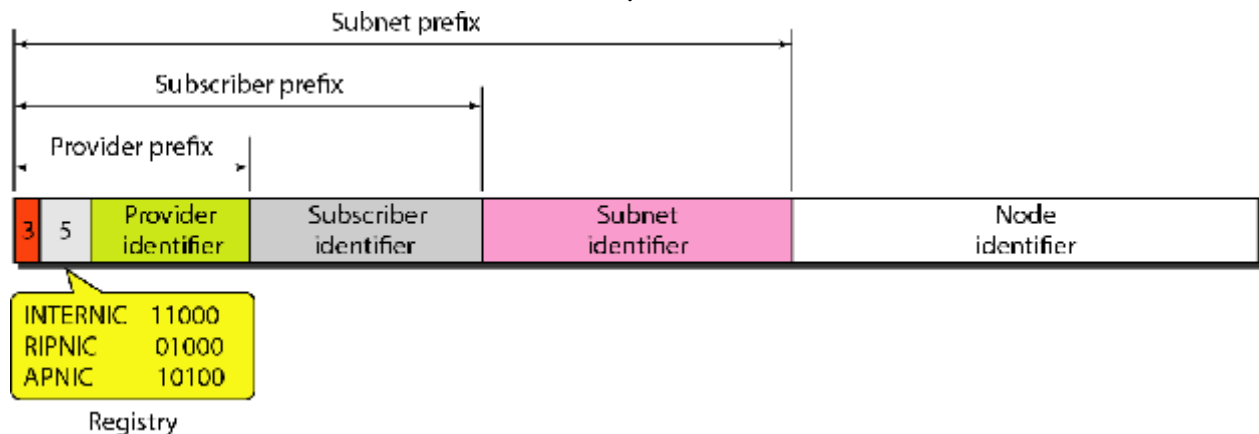


Fig. 4.10 Prefixes for provider-based unicast address

Fields for the provider-based address are as follows:

- Type identifier.** This 3-bit field defines the address as a provider-based address.
- Registry identifier.** This 5-bit field indicates the agency that has registered the address. Currently three registry centers have been defined. INTERNIC (code 11000) is the center for North America; RIPNIC (code 01000) is the center for European registration; and APNIC (code 10100) is for Asian and Pacific countries.
- Provider identifier.** This variable-length field identifies the provider for Internet access (such as an ISP). A 16-bit length is recommended for this field.
- Subscriber identifier.** When an organization subscribes to the Internet through a provider, it is assigned a subscriber identification. A 24-bit length is recommended for this field.
- Subnet identifier.** Each subscriber can have many different subnetworks, and each subnetwork can have an identifier. The subnet identifier defines a specific subnetwork under the territory of the subscriber. A 32-bit length is recommended for this field.
- Node identifier.** The last field defines the identity of the node connected to a subnet. A length of 48 bits is recommended for this field to make it compatible with the 48-bit link (physical) address used by Ethernet.

B) MULTICAST ADDRESSES

Multicast addresses are used to define a group of hosts instead of just one. A packet sent to a multicast address must be delivered to each member of the group. Fig. 4.11 shows the format of a multicast address.

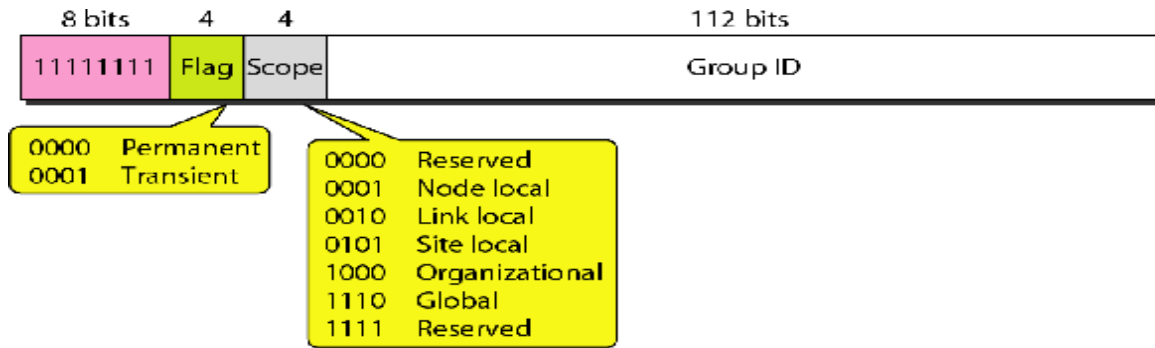


Fig. 4.11 Multicast address in IPv6

The second field is a flag that defines the group address as either permanent or transient. A permanent group address is defined by the Internet authorities and can be accessed at all times. A transient group address, on the other hand, is used only temporarily. Systems engaged in a teleconference, for example, can use a transient group address. The third field defines the scope of the group address. Many different scopes have been defined, as shown in Fig. 4.11.

C) ANYCAST ADDRESSES

IPv6 also defines anycast addresses. An anycast address, like a multicast address, also defines a group of nodes. However, a packet destined for an anycast address is delivered to only one of the members of the anycast group, the nearest one (the one with the shortest route). Although the definition of an anycast address is still debatable, one possible use is to assign an anycast address to all routers of an ISP that covers a large logical area in the Internet. The routers outside the ISP deliver a packet destined for the ISP to the nearest ISP router. No block is assigned for anycast addresses.

D) RESERVED ADDRESSES

Another category in the address space is the reserved address. These addresses start with eight Os (type prefix is 00000000). A few subcategories are defined in this category, as shown in Fig. 4.12.

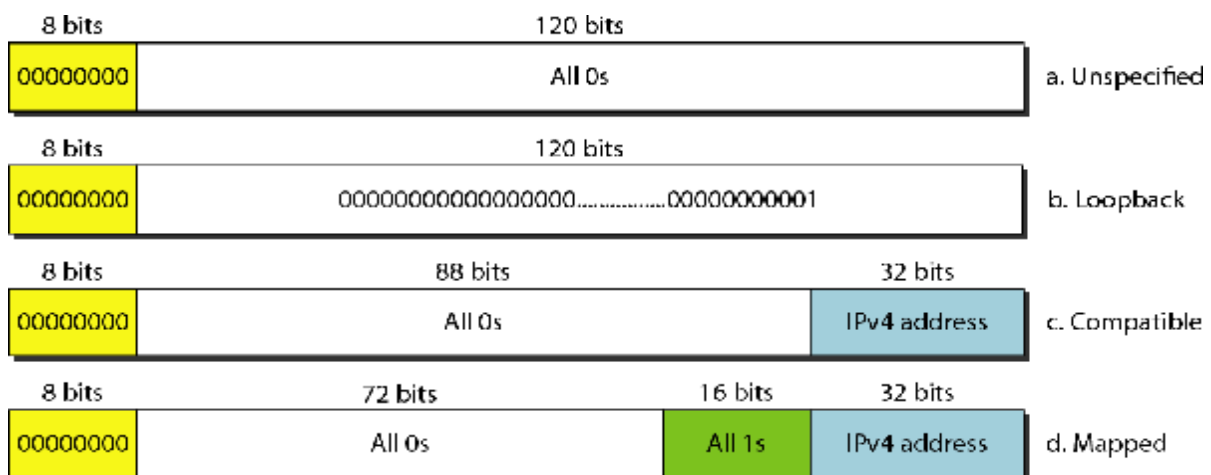


Fig. 4.12 Reserved addresses in IPv6

An unspecified address is used when a host does not know its own address and sends an inquiry to find its address. A loopback address is used by a host to test itself without going into the network. A compatible address is used during the transition from IPv4 to IPv6. It is used when a computer using IPv6 wants to send a message to another computer using IPv4, but the message needs to pass through a part of the network that still operates in IPv4. A mapped address is also used during transition. However, it is used when a computer that has migrated to IPv6 wants to send a packet to a computer still using IPv4.

E) LOCAL ADDRESSES

These addresses are used when an organization wants to use IPv6 protocol without being connected to the global Internet. In other words, they provide addressing for private networks. Nobody outside the organization can send a message to the nodes using these addresses. Two types of addresses are defined for this purpose, as shown in Fig. 4.13

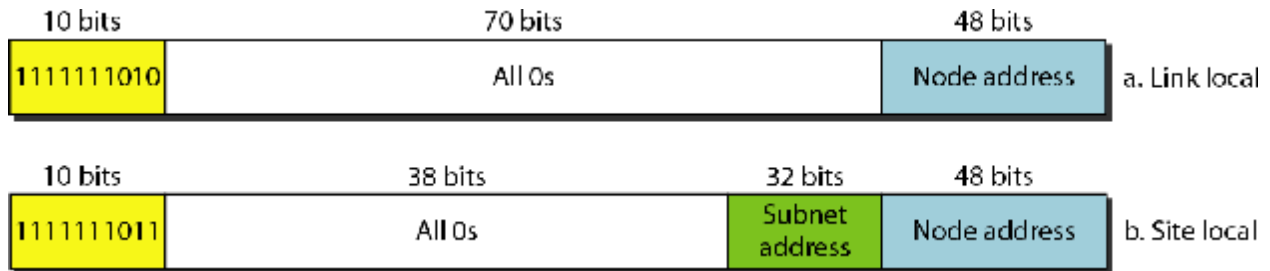


Fig 4.13 Local addresses in IPv6

A link local address is used in an isolated subnet; a site local address is used in an isolated site with several subnets

4.5 INTERNET PROTOCOL

IP specifies the format of packets, also called *datagrams*, and the addressing scheme. Most networks combine IP with a higher-level protocol called *Transmission Control Protocol (TCP)*, which establishes a virtual connection between a destination and a source. IP by itself is something like the postal system. It allows you to address a package and drop it in the system, but there's no direct link between you and the recipient. TCP/IP, on the other hand, establishes a connection between two hosts so that they can send messages back and forth for a period of time. The current version of IP is *IPv4*. A new version, called *IPv6* In internet protocol we discusse two internet protocols i.e.- IPv4 and IPv6

A) IPv4

The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols. Internet Protocol Version 4 (IPv4) is the fourth revision of the IP and a widely used protocol in data communication over different kinds of networks. IPv4 is a connectionless protocol used in packet-switched layer networks, such as Ethernet. It provides the logical connection between network devices by providing identification for each device. There are many ways to configure IPv4 with all kinds of devices - including manual and automatic configurations - depending on the network type.

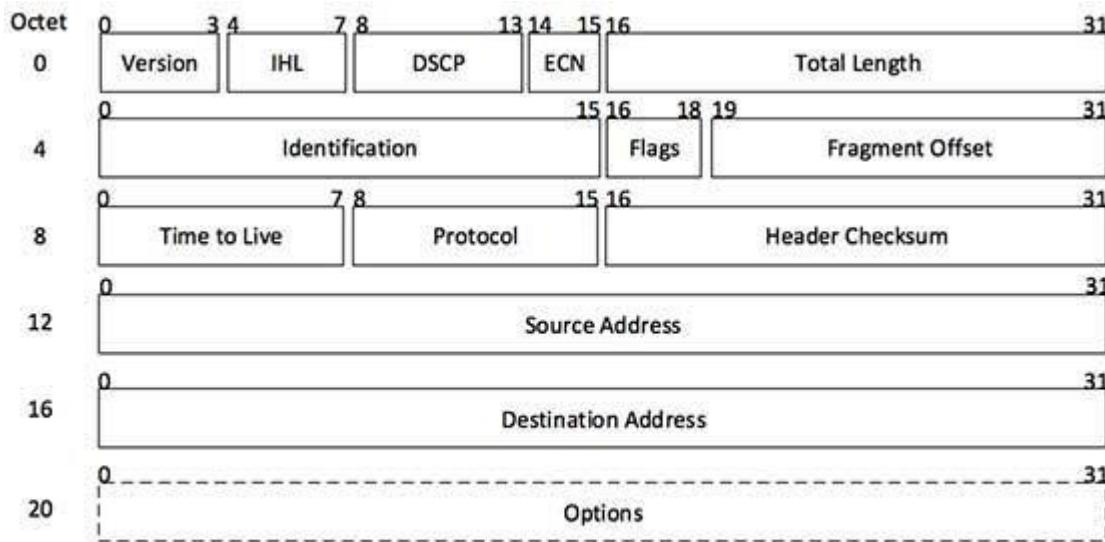
IPv4 is based on the best-effort model. This model guarantees neither delivery nor avoidance of duplicate delivery; these aspects are handled by the upper layer transport. IPv4 is defined and specified in IETF publication RCF 791. It is used in the packet-switched link layer in the OSI model.

IPv4 uses 32-bit addresses for Ethernet communication in five classes, named A, B, C, D and E. Classes A, B and C have a different bit length for addressing the network host. Class D addresses are reserved for multicasting, while class E addresses are reserved for future use.

Class A has subnet mask 255.0.0.0 or /8, B has subnet mask 255.255.0.0 or /16 and class C has subnet mask 255.255.255.0 or /24. For example, with a /16 subnet mask, the network 192.168.0.0 may use the address range of 192.168.0.0 to 192.168.255.255. Network hosts can take any address from this range; however, address 192.168.255.255 is reserved for broadcast within the network. The maximum number of host addresses IPv4 can assign to end users is 232. IPv6 presents a standardized solution to overcome IPv4's limitations. Because of its 128-bit address length, it can define up to 2,128 addresses.

Internet Protocol being a layer-3 protocol (OSI) takes data Segments from layer-4 (Transport) and divides it into what's called packet. IP packet encapsulates data unit received from above layer and adds its own header information.

The encapsulated data is referred to as IP Payload. IP header contains all the necessary information to deliver the packet at the other end.



[Image: IP Header]

Fig. 4.14 Frame format of IPv4

IP header includes many relevant information including Version Number, which, in this context, is 4. Other details are as follows:

- **Version:** Version no. of Internet Protocol used (e.g. IPv4)
- **IHL:** Internet Header Length, Length of entire IP header
- **DSCP:** Differentiated Services Code Point, This is Type of Service.

- **ECN:** Explicit Congestion Notification, carries information about the congestion seen in the route.
- **Total Length:** Length of entire IP Packet (including IP header and IP Payload)
- **Identification:** If IP packet is fragmented during the transmission, all the fragments contain same identification no. to identify original IP packet they belong to.
- **Flags:** As required by the network resources, if IP Packet is too large to handle these 'flags' tell that if they can be fragmented or not. In this 3-bit flag, the MSB is always set to '0'.
- **Fragment Offset:** This offset tells the exact position of the fragment in the original IP Packet.
- **Time to Live:** To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded.
- **Protocol:** Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol. For example protocol number of ICMP is 1, TCP is 6 and UDP is 17.
- **Header Checksum:** This field is used to keep checksum value of entire header which is then used to check if the packet is received error-free.
- **Source Address:** 32-bit address of the Sender (or source) of the packet.
- **Destination Address:** 32-bit address of the Receiver (or destination) of the packet.
- **Options:** This is optional field, which is used if the value of IHL is greater than 5. These options may contain values for options such as Security, Record Route, Time Stamp etc.

B) IPv6

The network layer protocol in the TCPIIP protocol suite is currently IPv4 (Internetworking Protocol, version 4). IPv4 provides the host-to-host communication between systems in the Internet. Although IPv4 is well designed, data communication has evolved since the inception of IPv4 in the 1970s. IPv4 has some deficiencies (listed below) that make it unsuitable for the fast-growing Internet.

- Despite all short-term solutions, such as subnetting, classless addressing, and NAT, address depletion is still a long-term problem in the Internet.
- The Internet must accommodate real-time audio and video transmission. This type of transmission requires minimum delay strategies and reservation of resources not provided in the IPv4 design.
- The Internet must accommodate encryption and authentication of data for some applications. No encryption or authentication is provided by IPv4.

To overcome these deficiencies, IPv6 (Internetworking Protocol, version 6), also known as IPng (Internetworking Protocol, next generation), was proposed and is now a standard. In IPv6, the Internet protocol was extensively modified to accommodate the unforeseen growth of the Internet. The format and the length of the IP address were changed along with the packet format. Related protocols, such as ICMP, were also modified. Other protocols in the network layer, such as ARP, RARP, and IGMP, were either deleted or included in the ICMPv6 protocol, Routing protocols, such as RIP and OSPF were also slightly modified to accommodate these changes.

Communications experts predict that IPv6 and its related protocols will soon replace the current IP version. In this section first we discuss IPv6. Then we explore the strategies used for the transition from version 4 to version 6. The adoption of IPv6 has been slow. The reason is that the original motivation for its development, depletion of IPv4 addresses, has been remedied by short-term strategies such as classless addressing and NAT. However, the fast-spreading use of the Internet, and new services such as mobile IP, IP telephony, and IP-capable mobile telephony, may eventually require the total replacement of IPv4 with IPv6.

Advantages

The next-generation IP, or IPv6, has some advantages over IPv4 that can be summarized as follows:

- Larger address space. An IPv6 address is 128 bits long. Compared with the 32-bit address of IPv4, this is a huge (296) increase in the address space.
- Better header format. IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- New options. IPv6 has new options to allow for additional functionalities.
- Allowance for extension. IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- Support for resource allocation. In IPv6, the type-of-service field has been removed, but a mechanism (called *flow label*) has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
- Support for more security. The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

An Internet Protocol version 6 (IPv6) data packet comprises of two main parts: the header and the payload. The first 40 bytes/octets (40x8 = 320 bits) of an IPv6 packet comprise of the header (see Fig. 4.15) that contains the following fields:

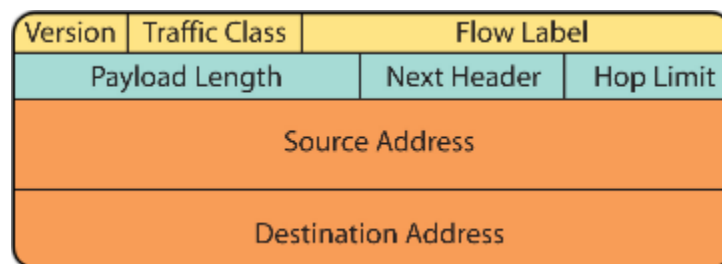


Fig. 4.15 Format of IPv6

Source address (128 bits) The 128-bit source address field contains the IPv6 address of the originating node of the packet. It is the address of the originator of the IPv6 packet.

Destination address (128 bits) The 128-bit contains the destination address of the recipient node of the IPv6 packet. It is the address of the intended recipient of the IPv6 packet.

Version/IP version (4-bits) The 4-bit version field contains the number 6. It indicates the version of the IPv6 protocol. This field is the same size as the IPv4 version field that contains the number 4. However, this field has a limited use because IPv4 and IPv6 packets are not distinguished based on the value in the version field but by the protocol type present in the layer 2 envelope.

Packet priority/Traffic class (8 bits) The 8-bit Priority field in the IPv6 header can assume different values to enable the source node to differentiate between the packets generated by it by associating different delivery priorities to them. This field is subsequently used by the originating node and the routers to identify the data packets that belong to the same traffic class and distinguish between packets with different priorities.

Flow Label/QoS management (20 bits) The 20-bit flow label field in the IPv6 header can be used by a source to label a set of packets belonging to the same flow. A flow is uniquely identified by the combination of the source address and of a non-zero Flow label. Multiple active flows may exist from a source to a destination as well as traffic that are not associated with any flow. The IPv6 routers must handle the packets belonging to the same flow in a similar fashion. The information on handling of IPv6 data packets belonging to a given flow may be specified within the data packets themselves or it may be conveyed by a control protocol such as the RSVP. When routers receive the first packet of a new flow, they can process the information carried by the IPv6 header, Routing header, and Hop-by-Hop extension headers, and store the result (e.g. determining the retransmission of specific IPv6 data packets) in a cache memory and use the result to route all other packets belonging to the same flow (having the same source address and the same Flow Label), by using the data stored in the cache memory.

Payload length in bytes(16 bits) The 16-bit payload length field contains the length of the data field in octets/bits following the IPv6 packet header. The 16-bit Payload length field puts an upper limit on the maximum packet payload to 64 kilobytes. In case a higher packet payload is required, a Jumbo payload extension header is provided in the IPv6 protocol. A Jumbo payload (Jumbogram) is indicated by the value zero in the Payload Length field. Jumbograms are frequently used in supercomputer communication using the IPv6 protocol to transmit heavy data payload.

Next Header (8 bits) The 8-bit Next Header field identifies the type of header immediately following the IPv6 header and located at the beginning of the data field (payload) of the IPv6 packet. This field usually specifies the transport layer protocol used by a packet's payload. The two most common kinds of Next Headers are TCP (6) and UDP (17), but many other headers are also possible. The format adopted for this field is the one proposed for IPv4 by RFC 1700. In case of IPv6 protocol, the Next Header field is similar to the IPv4 Protocol field.

Time To Live (TTL)/Hop Limit (8 bits) The 8-bit Hop Limit field is decremented by one, by each node (typically a router) that forwards a packet. If the Hop Limit field is decremented to zero, the packet is discarded. The main function of this field is to identify and to discard packets that are stuck in an indefinite loop due to any routing information errors. The 8-bit field also puts an upper limit on the maximum number of links between two IPv6 nodes. In this way, an IPv6 data packet is allowed a maximum of 255 hops before it is eventually discarded. An IPv6 data packet can pass through a maximum of 254 routers before being discarded.

In case of IPv6 protocol, the fields for handling fragmentation do not form a part of the basic header. They are put into a separate extension header. Moreover, fragmentation is exclusively handled by the sending host. Routers are not employed in the Fragmentation process.

4.6 Hardware Addressing versus IP Addressing

IP networks require two types of addresses : MAC and IP. Each station stores it's MAC address and IP address in it's own IP stack. It stores MAC and IP addresses of other stations on it's LAN or subnet in the ARP cache.

When the packet is being sent out to a station that is on the same network LAN segment, only the MAC address is needed. When the packet goes beyond, to different networks and travels through routers, the MAC address is still contained in the packet, but only the IP address is used by the routers.

Pairs of matched MAC/IP addresses are stored in the ARP cache.

NOTE: The order that Source/Destination MAC and IP addresses are sent, is reversed:

- packets are sent serially, and they send the Dest MAC first, then source MAC
- packets are sent serially, and they send the Source IP first, then the Dest IP

A) MAC ADDRESS (Layer2 - Data Link layer)

The hard coded 48-bit (6 byte) address, burned into the ROM of the NIC (Network Interface Card) - it is also called the *Hardware address*, or *Ethernet address*. They are expressed as six pairs of hexadecimal digits. The following is a MAC address for a Linksys Ethernet NIC :

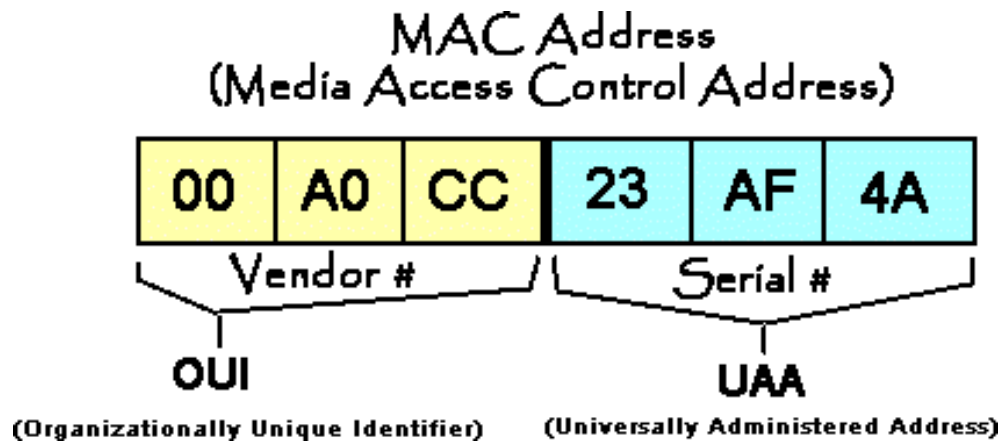


Fig. 4.16 MAC Address

The first 3 bytes are vendor numbers (also called the OUI - Organizationally Unique Identifier), and the last 3 are NIC serial numbers, or station address. This gives a theoretical 281,474,976,710,656 addresses. This is more than 56,000 MAC addresses for each person on the planet!

OUI -This is the identifier for a particular Vendor. The example above shows that all NIC's from Linksys have MAC addresses that begin with 00A0CC. Other vendor OUI's are:

00000C Cisco

00001D Cabletron

0020AF 3COM

08001B Data General

08002B DEC etc

UAA - since all cards from a given manufacturer have the same first 3 bytes - sometimes the MAC addresses are given as only the last 3 bytes (the station address) and it is called a UAA (Universally Administered Address).

LAA - Some low-level protocols can add an overlay to those 3 bytes, basically renaming them with 6 bytes store in software, but keeping a table for cross-referencing (some printer sharing Ethernet boxes do this) and then they are called an LAA (Locally Administered Address). This allows a network administrator to number the NIC's with meaningful digits . . .the last 3 bytes of the UAA address can be renamed to an LAA and contain information about the building, department, room, machine, wiring circuit, or owner's telephone number.

B) SPECIAL BITS

There are two special bits in a MAC address. They are the first two bits sent out on the wire in the MAC - the two least significant bits. They are shown as the last bits of the first byte of the MAC. Ethernet bytes are transmitted big-endian but the bits are transmitted little-endian.

The first bit (bit 0) is used by Ethernet II. The second bit is used by IEEE 802.3.

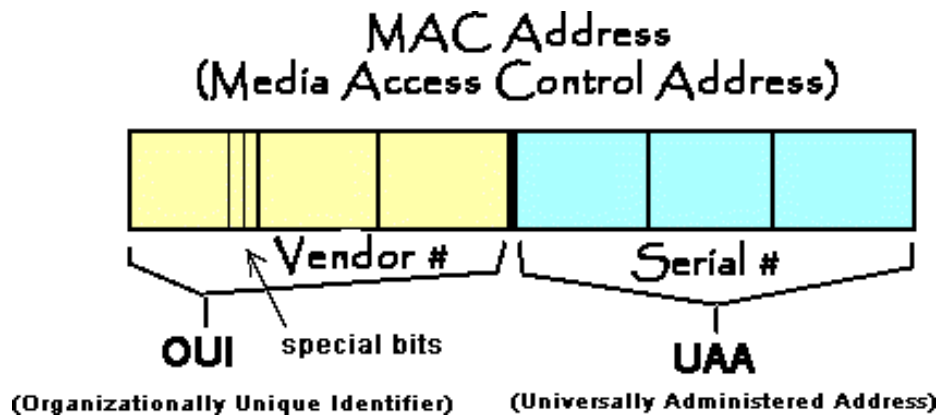


Fig. 4.17 MAC Address using special bits

Ethernet II special bit - if the first bit (bit 0, the LSB) is 0, then it is a physical MAC address (unicast), if it is 1, then it is a multicast address.

IEEE 802.3 special bit - if the second bit (bit 1) is 0, then the UAA address is used. If the bit is 1, then the LAA is used. In any case, a MAC address with an odd first byte is clearly not legal as a source address. Sources can never be multicast.

The IP address (layer 3 - Network layer)

A 32-bit (4 byte) software stored address, and is assigned to represent the same NIC as MAC address represents. The 32-bit IP address is like a shorter nickname for the 48-bit MAC address.

We will discuss IP addressing in detail in the later sections. But the main point in differentiating IP from MAC addresses, is this:

- direct-connected transmission uses Layer 2 - MAC addresses for frame delivery.
- routed transmission uses Layer 3 - IP addresses for packet delivery

4.7 IP DATAGRAM

Data transmitted over an internet using IP is carried in messages called *IP datagrams*. Like all network protocol messages, IP uses a specific format for its datagrams. We are of course looking here at IP version 4 and so we will examine the IPv4 datagram format, which was defined in RFC 791 along with the rest of IPv4.

The IPv4 datagram is conceptually divided into two pieces: the *header* and the *payload*. The header contains addressing and control fields, while the payload carries the actual data to be sent over the internetwork. Unlike some message formats, IP datagrams do not have a footer following the payload.

Even though IP is a relatively simple, connectionless, “unreliable” protocol, the IPv4 header carries a fair bit of information, which makes it rather large. At a minimum, it is 20 bytes long, and with options can be significantly longer. The IP datagram format is described in Table 4.4 and illustrated in Fig. 4.18.

Field Name	Size (bytes)	Description
<i>Version</i>	1/2 (4 bits)	Version: Identifies the version of IP used to generate the datagram. For IPv4, this is of course the number 4. The purpose of this field is to ensure compatibility between devices that may be running different versions of IP. In general, a device running an older version of IP will reject datagrams created by newer implementations, under the assumption that the older version may not be able to interpret the newer datagram correctly.
<i>IHL</i>	1/2 (4 bits)	Internet Header Length (IHL): Specifies the length of the IP header, in 32-bit words. This includes the length of any options fields and padding. The normal value of this field when no options are used is 5

		(5 32-bit words = 5*4 = 20 bytes). Contrast to the longer <i>Total Length</i> field below.												
TOS	1	Type Of Service (TOS): A field designed to carry information to provide quality of service features, such as prioritized delivery, for IP datagrams. It was never widely used as originally defined, and its meaning has been subsequently redefined for use by a technique called <i>Differentiated Services (DS)</i> . See below for more information.												
TL	2	Total Length (TL): Specifies the total length of the IP datagram, in bytes. Since this field is 16 bits wide, the maximum length of an IP datagram is 65,535 bytes, though most are much smaller.												
Identification	2	Identification: This field contains a 16-bit value that is common to each of the fragments belonging to a particular message; for datagrams originally sent unfragmented it is still filled in, so it can be used if the datagram must be fragmented by a router during delivery. This field is used by the recipient to reassemble messages without accidentally mixing fragments from different messages. This is needed because fragments may arrive from multiple messages mixed together, since IP datagrams can be received out of order from any device.												
Flags	3/8 (3 bits)	<p>Flags: Three control flags, two of which are used to manage fragmentation (as described in the topic on fragmentation), and one that is reserved:</p> <table border="1"> <thead> <tr> <th>Subfield Name</th> <th>Size (bytes)</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>Reserved</i></td> <td>1/8 (1 bit)</td> <td><i>Reserved:</i> Not used.</td> </tr> <tr> <td><i>DF</i></td> <td>1/8 (1 bit)</td> <td>Don't Fragment: When set to 1, specifies that the datagram should not be fragmented. Since the fragmentation process is generally "invisible" to higher layers, most protocols don't care about this and don't set this flag. It is, however, used for testing the maximum transmission unit (MTU) of a link.</td> </tr> <tr> <td><i>MF</i></td> <td>1/8 (1 bit)</td> <td>More Fragments: When set to 0, indicates the last fragment in a message; when set to 1, indicates that more fragments are yet to come in the fragmented message. If no fragmentation is used for a message, then of course there is only one "fragment" (the whole message), and this flag is 0. If fragmentation is used, all fragments but the last set this flag to 1 so the recipient knows when all fragments have been sent.</td> </tr> </tbody> </table>	Subfield Name	Size (bytes)	Description	<i>Reserved</i>	1/8 (1 bit)	<i>Reserved:</i> Not used.	<i>DF</i>	1/8 (1 bit)	Don't Fragment: When set to 1, specifies that the datagram should not be fragmented. Since the fragmentation process is generally "invisible" to higher layers, most protocols don't care about this and don't set this flag. It is, however, used for testing the maximum transmission unit (MTU) of a link.	<i>MF</i>	1/8 (1 bit)	More Fragments: When set to 0, indicates the last fragment in a message; when set to 1, indicates that more fragments are yet to come in the fragmented message. If no fragmentation is used for a message, then of course there is only one "fragment" (the whole message), and this flag is 0. If fragmentation is used, all fragments but the last set this flag to 1 so the recipient knows when all fragments have been sent.
Subfield Name	Size (bytes)	Description												
<i>Reserved</i>	1/8 (1 bit)	<i>Reserved:</i> Not used.												
<i>DF</i>	1/8 (1 bit)	Don't Fragment: When set to 1, specifies that the datagram should not be fragmented. Since the fragmentation process is generally "invisible" to higher layers, most protocols don't care about this and don't set this flag. It is, however, used for testing the maximum transmission unit (MTU) of a link.												
<i>MF</i>	1/8 (1 bit)	More Fragments: When set to 0, indicates the last fragment in a message; when set to 1, indicates that more fragments are yet to come in the fragmented message. If no fragmentation is used for a message, then of course there is only one "fragment" (the whole message), and this flag is 0. If fragmentation is used, all fragments but the last set this flag to 1 so the recipient knows when all fragments have been sent.												
Fragment	1 5/8	Fragment Offset: When fragmentation of a message occurs, this field												

<i>Offset</i>	(13 bits)	specifies the offset, or position, in the overall message where the data in this fragment goes. It is specified in units of 8 bytes (64 bits). The first fragment has an offset of 0..																																	
<i>TTL</i>	1	<p>Time To Live (TTL): Short version: Specifies how long the datagram is allowed to “live” on the network, in terms of router hops. Each router decrements the value of the TTL field (reduces it by one) prior to transmitting it. If the TTL field drops to zero, the datagram is assumed to have taken too long a route and is discarded.</p> <p>See below for the longer explanation of <i>TTL</i>.</p>																																	
<i>Protocol</i>	1	<p>Protocol: Identifies the higher-layer protocol (generally either a transport layer protocol or encapsulated network layer protocol) carried in the datagram. The values of this field were originally defined by the IETF “Assigned Numbers” standard, RFC 1700, and are now maintained by the Internet Assigned Numbers Authority (IANA):</p> <table border="1"> <thead> <tr> <th>Value (Hexadecimal)</th> <th>Value (Decimal)</th> <th>Protocol</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>0</td> <td>Reserved</td> </tr> <tr> <td>01</td> <td>1</td> <td>ICMP</td> </tr> <tr> <td>02</td> <td>2</td> <td>IGMP</td> </tr> <tr> <td>03</td> <td>3</td> <td>GGP</td> </tr> <tr> <td>04</td> <td>4</td> <td>IP-in-IP Encapsulation</td> </tr> <tr> <td>06</td> <td>6</td> <td>TCP</td> </tr> <tr> <td>08</td> <td>8</td> <td>EGP</td> </tr> <tr> <td>11</td> <td>17</td> <td>UDP</td> </tr> <tr> <td>32</td> <td>50</td> <td>Encapsulating Security Payload (ESP) Extension Header</td> </tr> <tr> <td>33</td> <td>51</td> <td>Authentication Header (AH) Extension Header</td> </tr> </tbody> </table> <p>Note that the last two entries are used when IPsec inserts additional headers into the datagram: the AH or ESP headers.</p>	Value (Hexadecimal)	Value (Decimal)	Protocol	00	0	Reserved	01	1	ICMP	02	2	IGMP	03	3	GGP	04	4	IP-in-IP Encapsulation	06	6	TCP	08	8	EGP	11	17	UDP	32	50	Encapsulating Security Payload (ESP) Extension Header	33	51	Authentication Header (AH) Extension Header
Value (Hexadecimal)	Value (Decimal)	Protocol																																	
00	0	Reserved																																	
01	1	ICMP																																	
02	2	IGMP																																	
03	3	GGP																																	
04	4	IP-in-IP Encapsulation																																	
06	6	TCP																																	
08	8	EGP																																	
11	17	UDP																																	
32	50	Encapsulating Security Payload (ESP) Extension Header																																	
33	51	Authentication Header (AH) Extension Header																																	
<i>Header Checksum</i>	2	<p>Header Checksum: A checksum computed over the header to provide basic protection against corruption in transmission. This is not the more complex CRC code typically used by data link layer technologies such as Ethernet; it's just a 16-bit checksum. It is calculated by dividing the header bytes into words (a word is two bytes) and then adding them together. The data is not check summed, only the header. At each hop the device receiving the datagram does the same checksum calculation and on a mismatch, discards the</p>																																	

		datagram as damaged.
Source Address	4	Source Address: The 32-bit IP address of the originator of the datagram. Note that even though intermediate devices such as routers may handle the datagram, they do not normally put their address into this field—it is always the device that originally sent the datagram.
Destination Address	4	Destination Address: The 32-bit IP address of the intended recipient of the datagram. Again, even though devices such as routers may be the intermediate targets of the datagram, this field is always for the ultimate destination.
Options	Variable	Options: One or more of several types of options may be included after the standard headers in certain IP datagrams.
Padding	Variable	Padding: If one or more options are included, and the number of bits used for them is not a multiple of 32, enough zero bits are added to “pad out” the header to a multiple of 32 bits (4 bytes).
Data	Variable	Data: The data to be transmitted in the datagram, either an entire higher-layer message or a fragment of one.

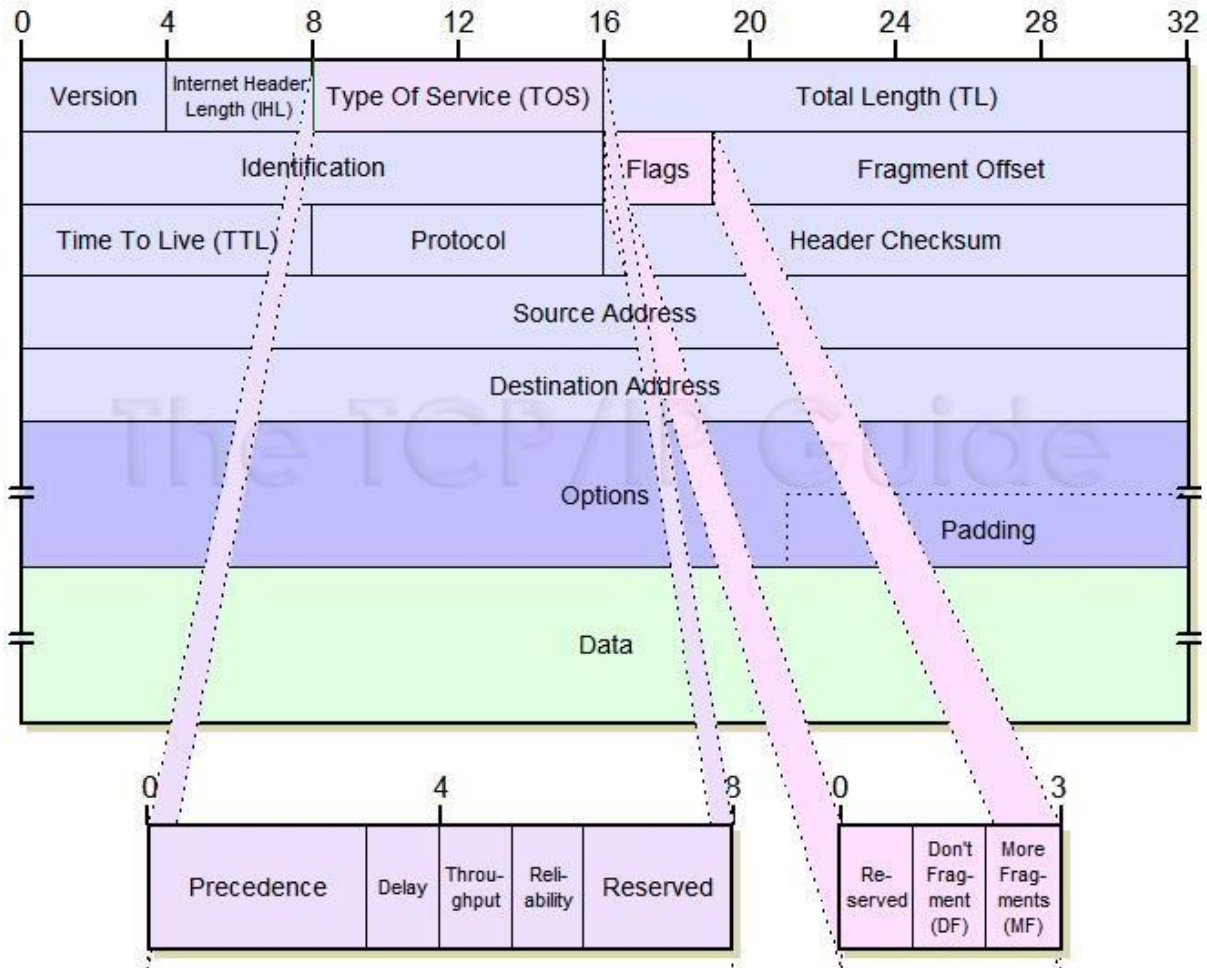


Fig. 4.18 : Internet Protocol Version 4 (IPv4) Datagram Format

This diagram shows graphically the all-important IPv4 datagram format. The first 20 bytes are the fixed IP header, followed by an optional Options section, and a variable-length Data area. Note that the Type Of Service field is shown as originally defined in the IPv4 standard.

UNIT- V

5.1 TRANSPORT LAYER PROTOCOL

The data link layer is responsible for delivery of frames between two neighboring nodes over a link. This is called *node-to-node delivery*. The network layer is responsible for delivery of datagrams between two hosts. This is called *host-to-host delivery*. Communication on the Internet is not defined as the exchange of data between two nodes or between two hosts. Real communication takes place between two processes (application programs). We need process-to-process delivery. However, at any moment, several processes may be running on the source host and several on the destination host. To complete the delivery, we need a mechanism to deliver data from one of these processes running on the source host to the corresponding process running on the destination host. The transport layer is responsible for process-to-process delivery-the delivery of a packet, part of a message, from one process to another. Two processes communicate in a client/server relationship. The original TCP/IP protocol suite specifies two protocols for the transport layer: UDP and TCP. We first focus on UDP, the simpler of the two, before discussing TCP. A new transport layer protocol, SCTP, has been designed, which we also discuss in this chapter. Figure 5.1 shows the position of these protocols in the TCP/IP protocol suite.

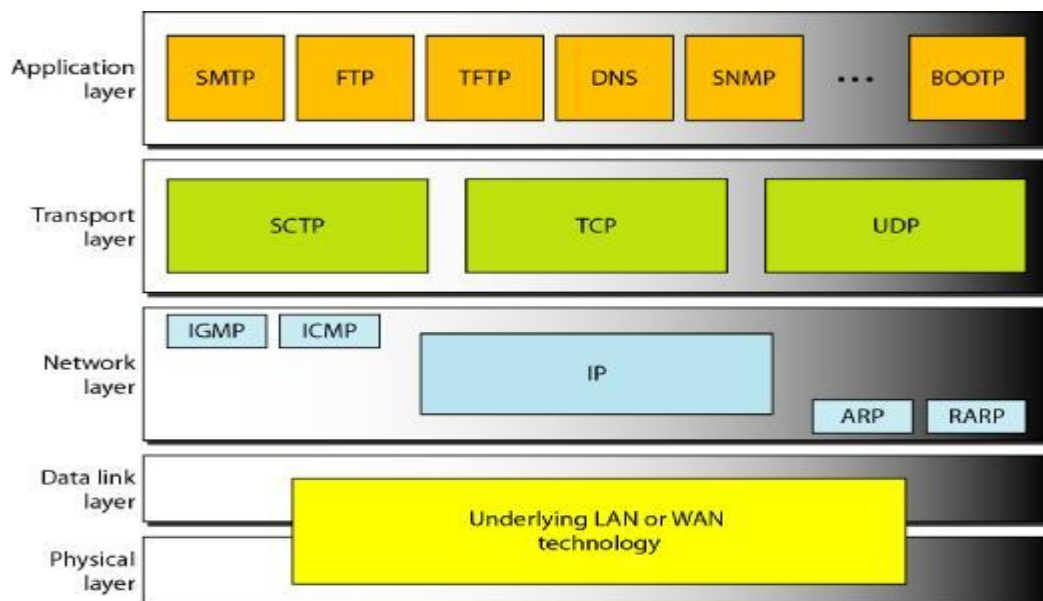


Fig.5.1 Position of UDP, TCP, and SCTP in TCP/IP suite

5.1.1 USER DATAGRAM PROTOCOL (UDP)

The User Datagram Protocol (UDP) is called a connectionless, 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? With the disadvantages come some advantages. 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.

Well-Known Ports for UDP

Table 5.1 shows some well-known port numbers used by UDP. Some port numbers can be used by both UDP and TCP


Table 5.1 Well Known Ports used in 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	Nameserver	Domain Name Service
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)

USER DATAGRAM

UDP packets, called user datagrams, have a fixed-size header of 8 bytes. Figure 5.2 shows the format of a user datagram.

The fields are as follows:

-  **Source port number.** This 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. If the source host is the client (a client sending a request), the port number, in most cases, is an ephemeral port number requested by the process and chosen by the UDP software running on the source host. If the source host is the server (a server sending a response), the port number, in most cases, is a well-known port number.

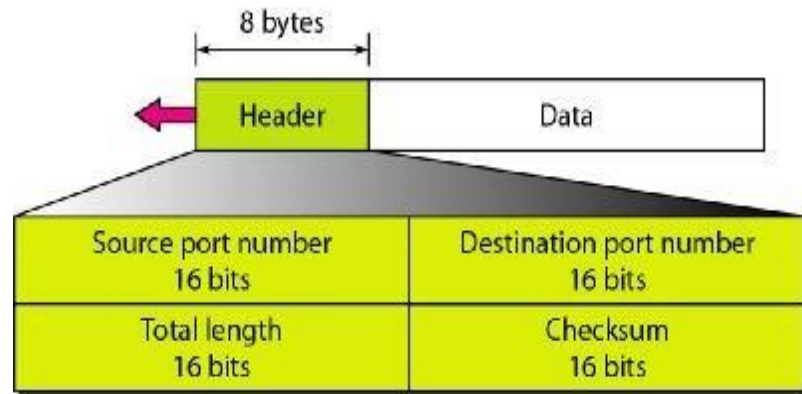


Fig. 5.2 User datagram format

- ▣ **Destination port number.** This is the port number used by the process running on the destination host. It is also 16 bits long. If the destination host is the server (a client sending a request), the port number, in most cases, is a well-known port number. If the destination host is the client (a server sending a response), the port number, in most cases, is an ephemeral port number. In this case, the server copies the ephemeral port number it has received in the request packet.
- ▣ **Length.** This 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 a total length of 65,535 bytes. The length field in a UDP user datagram is actually not necessary. A user datagram is encapsulated in an IP datagram. There is a field in the IP datagram that defines the total length. There is another field in the IP datagram that defines the length of the header. So if we subtract the value of the second field from the first, we can deduce the length of a UDP datagram that is encapsulated in an IP datagram.
- ▣ However, the designers of the UDP protocol felt that it was more efficient for the destination UDP to calculate the length of the data from the information provided in the UDP user datagram rather than ask the IP software to supply this information. We should remember that when the IP software delivers the UDP user datagram to the UDP layer, it has already dropped the IP header.
- ▣ **Checksum.** This field is used to detect errors over the entire user datagram (header plus data)

UDP Operation

UDP uses concepts common to the transport layer. These concepts will be discussed here briefly, and then expanded in the next section on the TCP protocol.

Connectionless Services

As mentioned previously, UDP provides a connectionless service. This 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 and going to the same destination program. The user datagrams are not numbered.

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.

One of the ramifications of being connectionless is that the process that uses UDP cannot send a stream of data to UDP and expect UDP to chop them into different related user datagrams. Instead each request must be small enough to fit into one user datagram. Only those processes sending short messages should use UDP.

Flow and Error Control

UDP is a very simple, unreliable transport protocol. There is no flow control and hence no window mechanism. The receiver may overflow with incoming messages. There is no error control mechanism in UDP except for the checksum. This means that the sender does not know if a message has been lost or duplicated. When the receiver detects an error through the checksum, the user datagram is silently discarded. The lack of flow control and error control means that the process using UDP should provide these mechanisms.

Encapsulation and Decapsulation

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.

Queuing

We have talked about ports without discussing the actual implementation of them. In UDP, queues are associated with ports (see Figure 5.3).

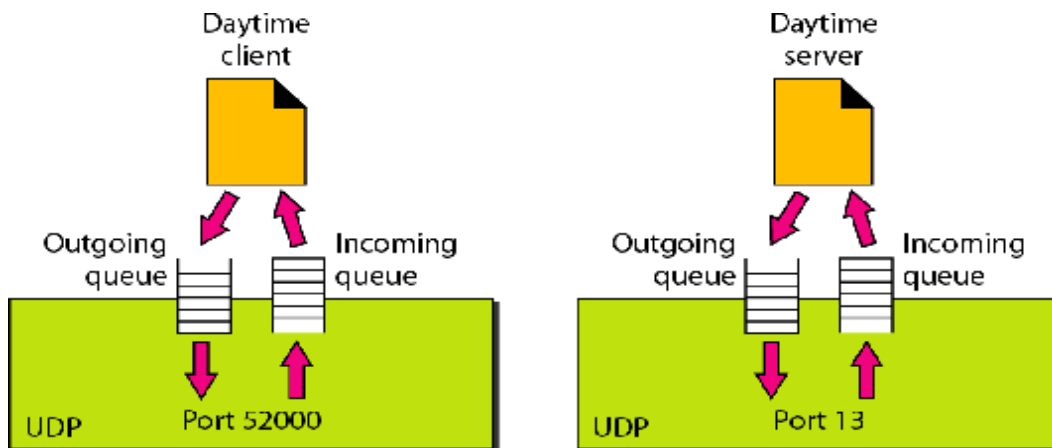


Fig. 5.3 Queue in UDP

At the client site, when a process starts, it requests a port number from the operating system. Some implementations create both an incoming and an outgoing queue associated with each process. Other implementations create only an incoming queue associated with each process. Note that even if a process wants to communicate with multiple processes, it obtains only one port number and eventually one outgoing and one incoming queue. The queues opened by the

client are, in most cases, identified by ephemeral port numbers. The queues function as long as the process is running. When the process terminates, the queues are destroyed.

The client process can send messages to the outgoing queue by using the source port number specified in the request. UDP removes the messages one by one and, after adding the UDP header, delivers them to IP. An outgoing queue can overflow. If this happens, the operating system can ask the client process to wait before sending any more messages.

When a message arrives for a client, UDP checks to see if an incoming queue has been created for the port number specified in the destination port number field of the user datagram. If there is such a queue, UDP sends the received user datagram to the end of the queue. If there is no such queue, UDP discards the user datagram and asks the ICMP protocol to send a *port unreachable* message to the server. All the incoming messages for one particular client program, whether coming from the same or a different server, are sent to the same queue. An incoming queue can overflow. If this happens, UDP drops the user datagram and asks for a port unreachable message to be sent to the server.

At the server site, the mechanism of creating queues is different. In its simplest form, a server asks for incoming and outgoing queues, using its well-known port, when it starts running. The queues remain open as long as the server is running. When a message arrives for a server, UDP checks to see if an incoming queue has been created for the port number specified in the destination port number field of the user datagram. If there is such a queue, UDP sends the received user datagram to the end of the queue. If there is no such queue, UDP discards the user datagram and asks the ICMP protocol to send a port unreachable message to the client. All the incoming messages for one particular server, whether coming from the same or a different client, are sent to the same queue. An incoming queue can overflow. If this happens, UDP drops the user datagram and asks for a port unreachable message to be sent to the client.

When a server wants to respond to a client, it sends messages to the outgoing queue, using the source port number specified in the request. UDP removes the messages one by one and, after adding the UDP header, delivers them to IP. An outgoing queue can overflow. If this happens, the operating system asks the server to wait before sending any more messages.

Use of UDP

The following lists some uses of the UDP protocol:

- UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control. It is not usually used for a process such as FrP that needs to send bulk data.
- 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. 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 .
- UDP is used for some route updating protocols such as Routing Information Protocol(RIP)

5.1.2 TCP

The second transport layer protocol we discuss in this chapter is called Transmission Control Protocol (TCP). TCP, like UDP, is a process-to-process (program-to-program) protocol. TCP, therefore, like UDP, uses port numbers. Unlike UDP, TCP is a connection oriented protocol; it creates a virtual connection between two TCPs to send data. In addition, TCP uses flow and error control mechanisms at the transport level. In brief, TCP is called a *connection-oriented, reliable* transport protocol. It adds connection-oriented and reliability features to the services of IP.

Connection-Oriented Service

TCP, unlike UDP, is a connection-oriented protocol. When a process at site A wants to send and receive data from another process at site B, the following occurs:

1. The two TCPs establish a 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 use 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. The situation is similar to creating a bridge that spans multiple islands and passing all the bytes from one island to another in one single connection.

Reliable Service

TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data. We will discuss this feature further in the section on error control.

A TCP Connection

TCP is connection-oriented. A connection-oriented transport protocol establishes a virtual path between the source and destination. All the segments belonging to a message are then sent over this virtual path. Using a single virtual pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames. You may wonder how TCP, which uses the services of IP, a connectionless protocol, can be connection-oriented. The point is that a TCP connection is virtual, not physical. TCP operates at a higher level. TCP uses the services of IP to deliver individual segments to the receiver, but it controls the connection itself. If a segment is lost or corrupted, it is retransmitted. Unlike TCP, IP is unaware of this retransmission. If a segment arrives out of order, TCP holds it until the missing segments arrive; IP is unaware of this reordering. In TCP, connection-oriented transmission requires three phases: connection establishment, data transfer, and connection termination.

Connection Establishment

TCP transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously. This implies that each party must initialize

communication and get approval from the other party before any data are transferred. **Three-Way Handshaking** The connection establishment in TCP is called three way handshaking. In our example, an application program, called the client, wants to make a connection with another application program, called the server, using TCP as the transport layer protocol.

The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This is called a request for a *passive open*. Although the server TCP is ready to accept any connection from any machine in the world, it cannot make the connection itself.

The client program issues a request for an *active open*. A client that wishes to connect to an open server tells its TCP that it needs to be connected to that particular server. TCP can now start the three-way handshaking process as shown in Figure 5.4. To show the process, we use two time lines: one at each site. Each segment has values for all its header fields and perhaps for some of its option fields, too. However, we show only the few fields necessary to understand each phase. We show the sequence number, the acknowledgment number, the control flags (only those that are set), and the window size, if not empty. The three steps in this phase are as follows.

1. The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers. It consumes one sequence number. When the data transfer starts, the sequence number is incremented by 1. We can say that the SYN segment carries no real data, but we can think of it as containing 1 imaginary byte.

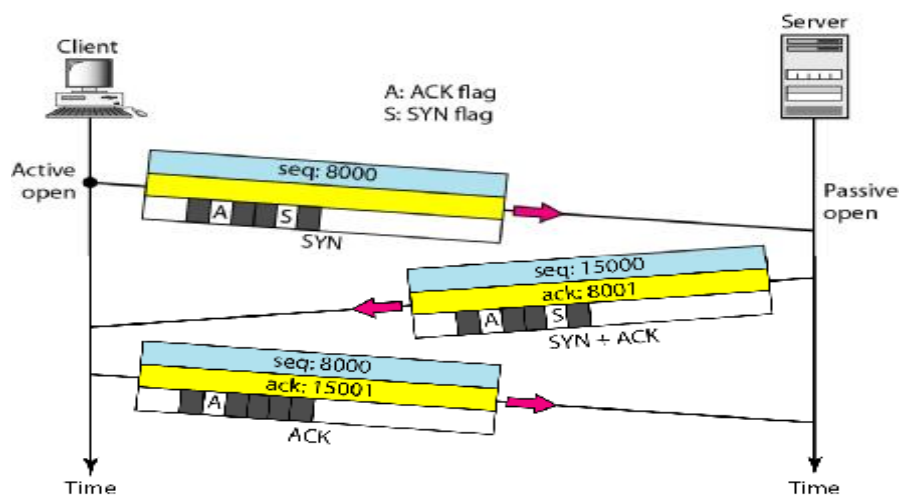


Fig. 5.4 Connection establishment using three-way handshaking

2. The server sends the second segment, a SYN +ACK segment, with 2 flag bits set: SYN and ACK. This segment has a dual purpose. It is a SYN segment for communication in the other direction and serves as the acknowledgment for the SYN segment. It consumes one sequence number.
3. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field. Note that the sequence number in this segment is the same as the one in the SYN segment; the ACK segment does not consume any sequence numbers.

Data Transfer

After connection is established, bidirectional data transfer can take place. The client and server can both send data and acknowledgments.. The acknowledgment is piggybacked with the data. Figure 5.5 shows an example. In this example, after connection is established (not shown in the figure), the client sends 2000 bytes of data in two segments. The server then sends 2000 bytes in one segment. The client sends one more segment. The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgment because there are no more data to be sent. Note the values of the sequence and acknowledgment numbers. The data segments sent by the client have the PSH (push) flag set so that the server TCP knows to deliver data to the server process as soon as they are received. The segment from the server, on the other hand, does not set the push flag. Most TCP implementations have the option to set or not set this flag. Pushing Data We saw that the sending TCP uses a buffer to store the stream of data coming from the sending application program. The sending TCP can select the segment size. The receiving TCP also buffers the data when they arrive and delivers them to the application program when the application program is ready or when it is convenient for the receiving TCP. This type of flexibility increases the efficiency of TCP. However, on occasion the application program has no need for this flexibility_ For example, consider an application program that communicates interactively with another application program on the other end. The application program on one site wants to send a keystroke to the application at the other site and receive an immediate response. Delayed transmission and delayed delivery of data may not be acceptable by the application program. TCP can handle such a situation. The application program at the sending site can request a *push* operation. This means that the sending TCP must not wait for the window to be filled. It must create a segment and send it immediately. The sending TCP must also set the push bit (PSH) to let the receiving TCP know that the segment includes data that must be delivered to the receiving application program as soon as possible and not to wait for more data to come

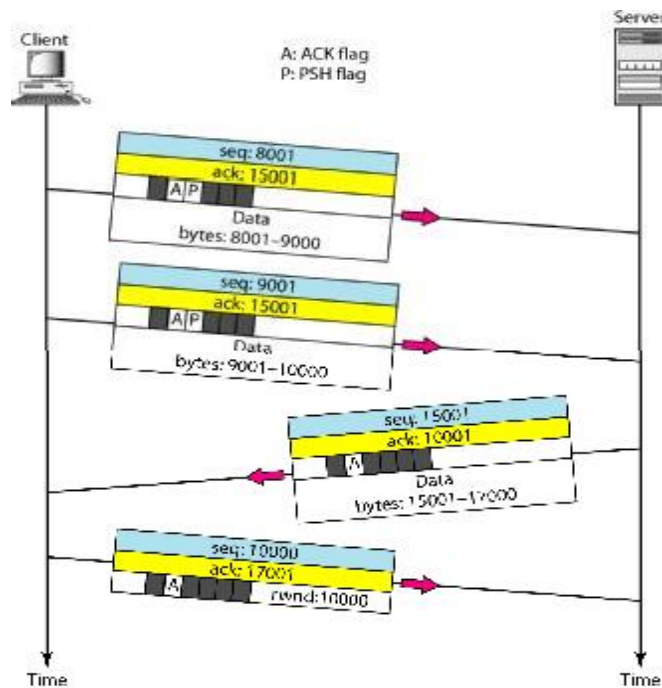


Fig. 5.5 Data Transfer

Connection Termination

Any of the two parties involved in exchanging data (client or server) can close the connection, although it is usually initiated by the client. Most implementations today allow two options for connection termination: three-way handshaking and four-way handshaking with a half-close option.

Three-Way Handshaking Most implementations today allow *three-way handshaking* for connection termination as shown in Figure 5.6.

1. In a normal situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set. Note that a FIN segment can include the last chunk of data sent by the client, or it can be just a

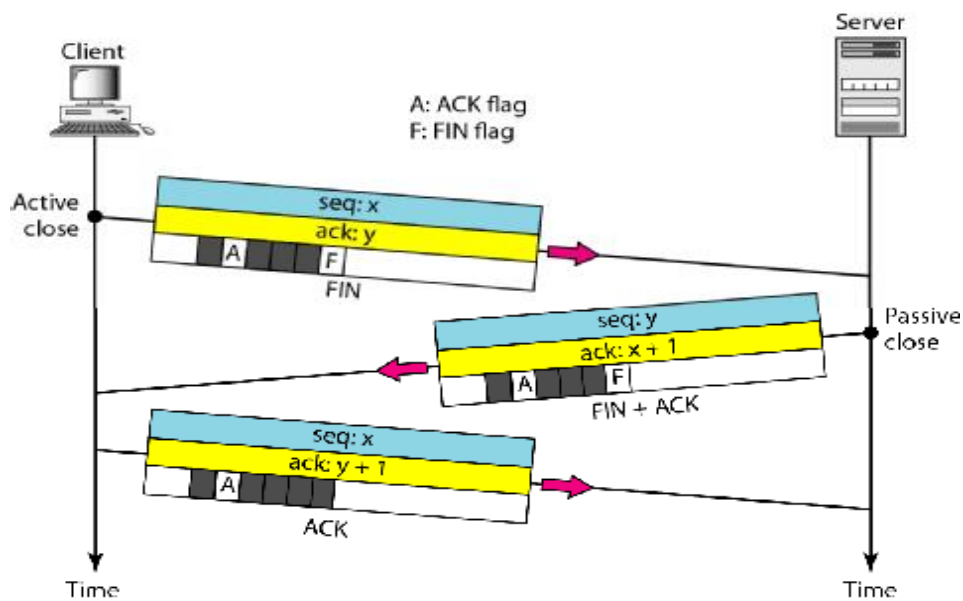


Fig 5.6 Connection Termination

control segment as shown in Figure 5.6. If it is only a control segment, it consumes only one sequence number.

2. The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN +ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction. This segment can also contain the last chunk of data from the server. If it does not carry data, it consumes only one sequence number.
3. The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server. This segment contains the acknowledgment number, which is 1 plus the sequence number received in the FIN segment from the server. This segment cannot carry data and consumes no sequence numbers.

5.2 ATM

Asynchronous Transfer Mode (ATM) is the cell relay protocol designed by the ATM Forum and adopted by the ITU-T. The combination of ATM and SONET will allow high-speed interconnection of all the world's networks. In fact, ATM can be thought of as the "highway" of the information superhighway.

DESIGN GOALS

Among the challenges faced by the designers of ATM, six stand out.

1. Foremost is the need for a transmission system to optimize the use of high-data-rate transmission media, in particular optical fiber. In addition to offering large bandwidths, newer transmission media and equipment are dramatically less susceptible to noise degradation. A technology is needed to take advantage of both factors and thereby maximize data rates.
2. The system must interface with existing systems and provide wide-area interconnectivity between them without lowering their effectiveness or requiring their replacement.
3. The design must be implemented inexpensively so that cost would not be a barrier to adoption. If ATM is to become the backbone of international communications, as intended, it must be available at low cost to every user who wants it.
4. The new system must be able to work with and support the existing telecommunications hierarchies (local loops, local providers, long-distance carriers, and so on).
5. The new system must be connection-oriented to ensure accurate and predictable delivery.
6. Last but not least, one objective is to move as many of the functions to hardware as possible (for speed) and eliminate as many software functions as possible (again for speed).

Cell Networks

Many of the problems associated with frame internetworking are solved by adopting a concept called cell networking. A cell is a small data unit of fixed size. In a **cell** network, which uses the **cell** as the basic unit of data exchange, all data are loaded into identical cells that can be transmitted with complete predictability and uniformity. As frames of different sizes and formats reach the cell network from a tributary network, they are split into multiple small data units of equal length and are loaded into cells. The cells are then multiplexed with other cells and routed through the cell network. Because each cell is the same size and all are small, the problems associated with multiplexing different-sized frames are avoided

Asynchronous TDM

ATM uses asynchronous time-division multiplexing—that is why it is called Asynchronous Transfer Mode—to multiplex cells coming from different channels. It uses fixed-size slots (size of a cell). ATM multiplexers fill a slot with a cell from any input channel that has a cell; the slot is empty if none of the channels has a cell to send. Figure 5.7 shows how cells from three inputs are multiplexed. At the first tick of the clock: channel 2 has no cell (empty input slot), so the

multiplexer fills the slot with a cell from the third channel. When all the cells from all the channels are multiplexed, the output slots are empty.

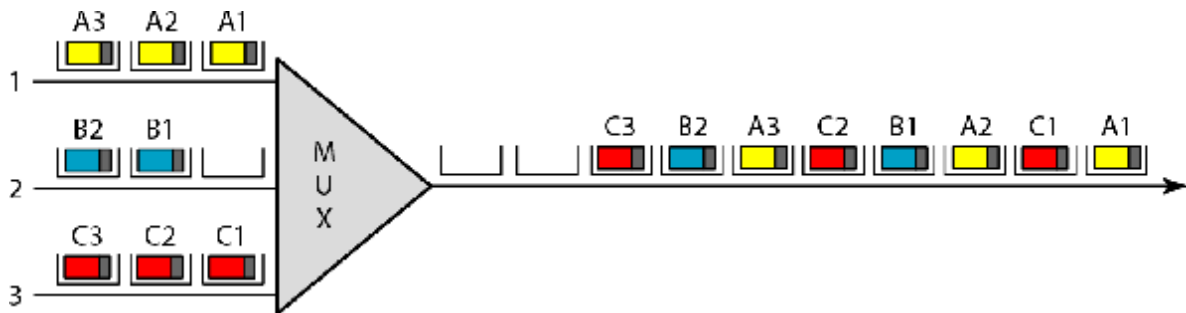


Fig. 5.7 ATM multiplexing

Architecture

ATM is a cell-switched network. The user access devices, called the endpoints, are connected through a user-to-network interface (UNI) to the switches inside the network. The switches are connected through network-to-network interfaces (NNIs). Figure 5.8 shows an example of an ATM network.

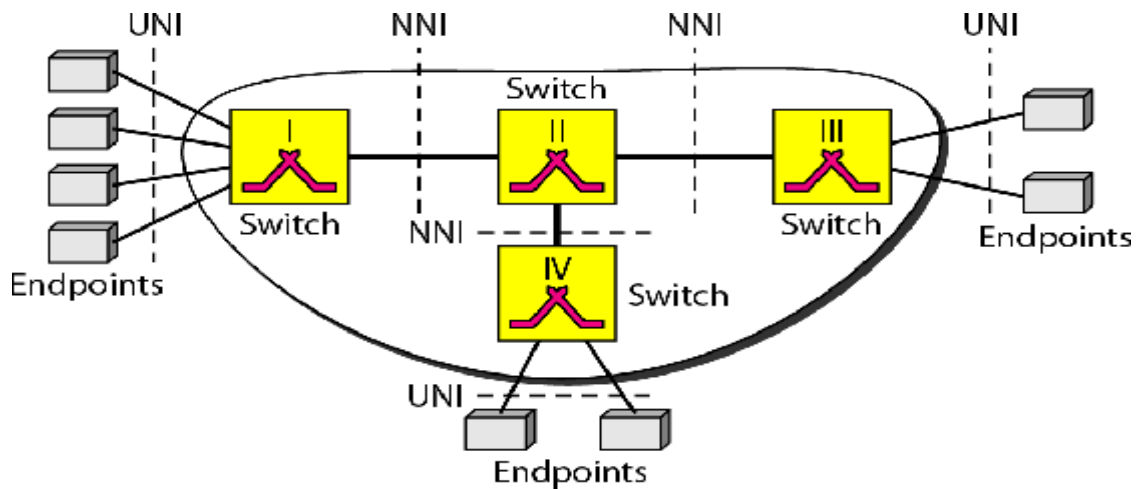


Fig 5.8 Architecture of an ATM network

Virtual Connection

Connection between two endpoints is accomplished through transmission paths (TPs), virtual paths (YPs), and virtual circuits (YCs). A transmission path (TP) is the physical connection (wire, cable, satellite, and so on) between an endpoint and a switch or between two switches. Think of two switches as two cities. A transmission path is the set of all highways that directly connect the two cities. A transmission path is divided into several virtual paths. A virtual path (VP) provides a connection or a set of connections between two switches. Think of a virtual path as a highway that connects two cities. Each highway is a virtual path; the set of all highways is the transmission path.

Cell networks are based on virtual circuits (VCs). All cells belonging to a single message follow the same virtual circuit and remain in their original order until they reach their destination. Think of a virtual circuit as the lanes of a highway (virtual path).

Figure 5.9 shows the relationship between a transmission path (a physical connection), virtual paths (a combination of virtual circuits that are bundled together because parts of their paths are the same), and virtual circuits that logically connect two points.

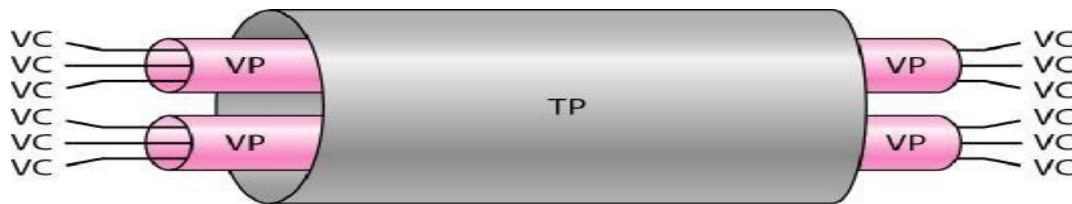


Fig. 5.9 Tp, VPs, and VCs

ATM LAYERS:

The ATM standard defines three layers. They are, from top to bottom, the application adaptation layer, the ATM layer, and the physical layer. The endpoints use all three layers while the switches use only the two bottom layers.

A) PHYSICAL LAYER

Like Ethernet and wireless LANs, ATM cells can be carried by any physical layer carrier. SONET The original design of ATM was based on *SONET* as the physical layer carrier. SONET is preferred for two reasons. First, the high data rate of SONET's carrier reflects the design and philosophy of ATM. Second, in using SONET, the boundaries of cells can be clearly defined. SONET specifies the use of a pointer to define the beginning of a payload. If the beginning of the first ATM cell is defined, the rest of the cells in the same payload can easily be identified because there are no gaps between cells. Just count 53 bytes ahead to find the next cell. Other Physical Technologies ATM does not limit the physical layer to SONET. Other technologies, even wireless, may be used. However, the problem of cell boundaries must be solved. One solution is for the receiver to guess the end of the cell and apply the CRC to the 5-byte header. If there is no error, the end of the cell is found, with a high probability, correctly. Count 52 bytes back to find the beginning of the cell.

B) ATM LAYER

The ATM layer provides routing, traffic management, switching, and multiplexing services. It processes outgoing traffic by accepting 48-byte segments from the AAL sublayers and transforming them into 53-byte cells by the addition of a 5-byte header

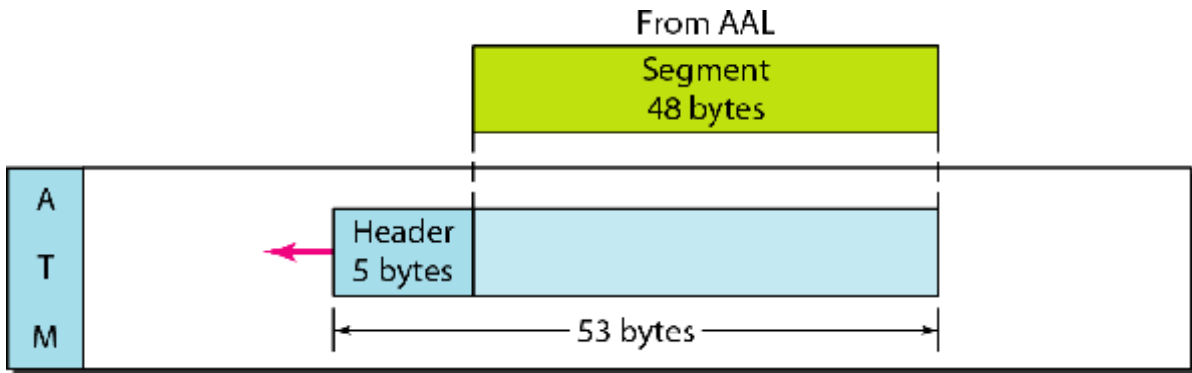


Fig. 5.10 ATM Layer

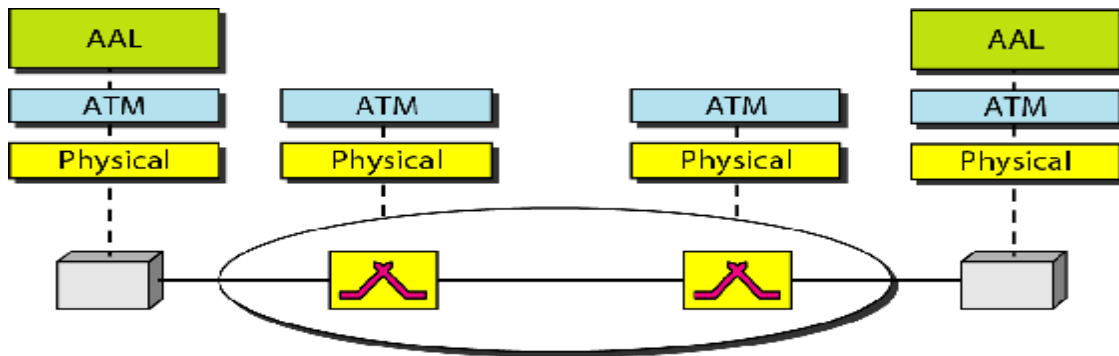


Fig. 5.11 ATM layers in endpoint devices and switches

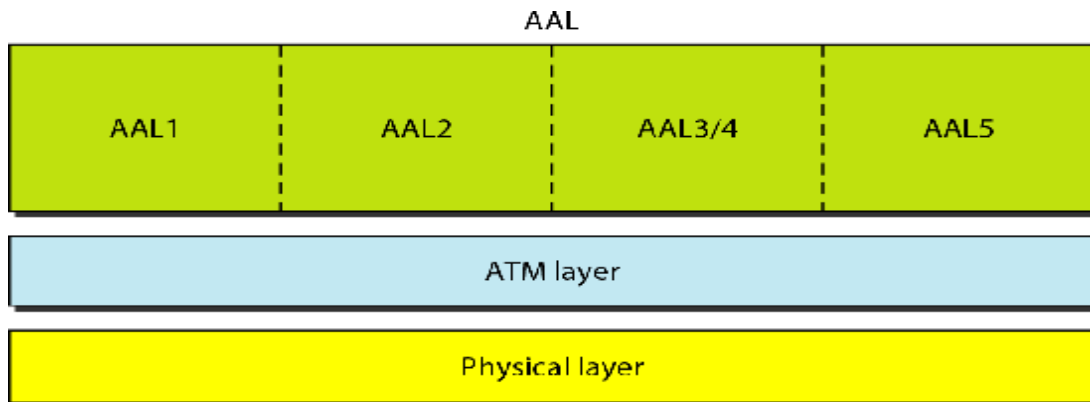


Fig. 5.12 ATM Layer in detail

C) APPLICATION ADAPTATION LAYER

The application adaptation layer (AAL) was designed to enable two ATM concepts. First, ATM must accept any type of payload, both data frames and streams of bits. A data frame can come from an upper-layer protocol that creates a clearly defined frame to be sent to a carrier network such as ATM. A good example is the Internet. ATM must also carry multimedia payload. It can accept continuous bit streams and break them into chunks to be encapsulated into a cell at the ATM layer. AAL uses two sublayers to accomplish these tasks. Whether the data are a data frame or a stream of bits, the

payload must be segmented into 48-byte segments to be carried by a cell. At the destination, these segments need to be reassembled to recreate the original payload. The AAL defines a sublayer, called a segmentation and reassembly (SAR) sublayer, to do so. Segmentation is at the source; reassembly, at the destination. Before data are segmented by SAR, they must be prepared to guarantee the integrity of the data. This is done by a sublayer called the convergence sublayer (CS). ATM defines four versions of the AAL: AAL1, AAL2, AAL3/4, and AAL5. The first is used in streaming audio and video communication; the second, in data communications. AAL1 supports applications that transfer information at constant bit rates, such as video and voice. It allows ATM to connect existing digital telephone networks such as voice channels and T lines.

AAL3/4 Initially, AAL3 was intended to support connection-oriented data services and AAL4 to support connectionless services. As they evolved, however, it became evident that the fundamental issues of the two protocols were the same. They have therefore been combined into a single format called AAL3/4.

AALS AAL3/4 provides comprehensive sequencing and error control mechanisms that are not necessary for every application. For these applications, the designers of ATM have provided a fifth AAL sublayer, called the simple and efficient adaptation layer (SEAL). AALS assumes that all cells belonging to a single message travel sequentially and that control functions are included in the upper layers of the sending application.

5.3 CRYPTOGRAPHY

Does increased security provide comfort to paranoid people? Or does security provide some very basic protections that we are naive to believe that we don't need? During this time when the Internet provides essential communication between tens of millions of people and is being increasingly used as a tool for commerce, security becomes a tremendously important issue to deal with. There are many aspects to security and many applications, ranging from secure commerce and payments to private communications and protecting passwords. But it is important to note that while cryptography is *necessary* for secure communications, it is not by itself *sufficient*.

This paper has two major purposes. The first is to define some of the terms and concepts behind basic cryptographic methods, and to offer a way to compare the myriad cryptographic schemes in use today. The second is to provide some real examples of cryptography in use today.

The Purpose of Cryptography

Cryptography is the science of writing in secret code and is an ancient art; the first documented use of cryptography in writing dates back to circa 1900 B.C. when an Egyptian scribe used non-standard hieroglyphs in an inscription. Some experts argue that cryptography appeared spontaneously sometime after writing was invented, with applications ranging from diplomatic missives to war-time battle plans. It is no surprise, then, that new forms of cryptography came soon after the widespread development of computer communications. In data and telecommunications, cryptography is necessary when communicating over any untrusted medium, which includes just about *any* network, particularly the Internet. Within the context of any application-to-application communication, there are some specific security requirements, including:

- Authentication: The process of proving one's identity. (The primary forms of host-to-host authentication on the Internet today are name-based or address-based, both of which are notoriously weak.)
- Privacy/confidentiality: Ensuring that no one can read the message except the intended receiver.
- Integrity: Assuring the receiver that the received message has not been altered in any way from the original.
- Non-repudiation: A mechanism to prove that the sender really sent this message.

Cryptography, then, not only protects data from theft or alteration, but can also be used for user authentication. There are, in general, three types of cryptographic schemes typically used to accomplish these goals: secret key (or symmetric) cryptography, public-key (or asymmetric) cryptography, and hash functions, each of which is described below. In all cases, the initial unencrypted data is referred to as *plaintext*. It is encrypted into *ciphertext*, which will in turn (usually) be decrypted into usable plaintext. In many of the descriptions below, two communicating parties will be referred to as Alice and Bob; this is the common nomenclature in the crypto field and literature to make it easier to identify the communicating parties. If there is a third or fourth party to the communication, they will be referred to as Carol and Dave. Mallory is a malicious party, Eve is an eavesdropper, and Trent is a trusted third party.

Types of Cryptographic Algorithms

There are several ways of classifying cryptographic algorithms. For purposes of this paper, they will be categorized based on the number of keys that are employed for encryption and decryption, and further defined by their application and use. The three types of algorithms are (Figure 5.13):

- Secret Key Cryptography (SKC): Uses a single key for both encryption and decryption
- Public Key Cryptography (PKC): Uses one key for encryption and another for decryption
- Hash Functions: Uses a mathematical transformation to irreversibly "encrypt" information

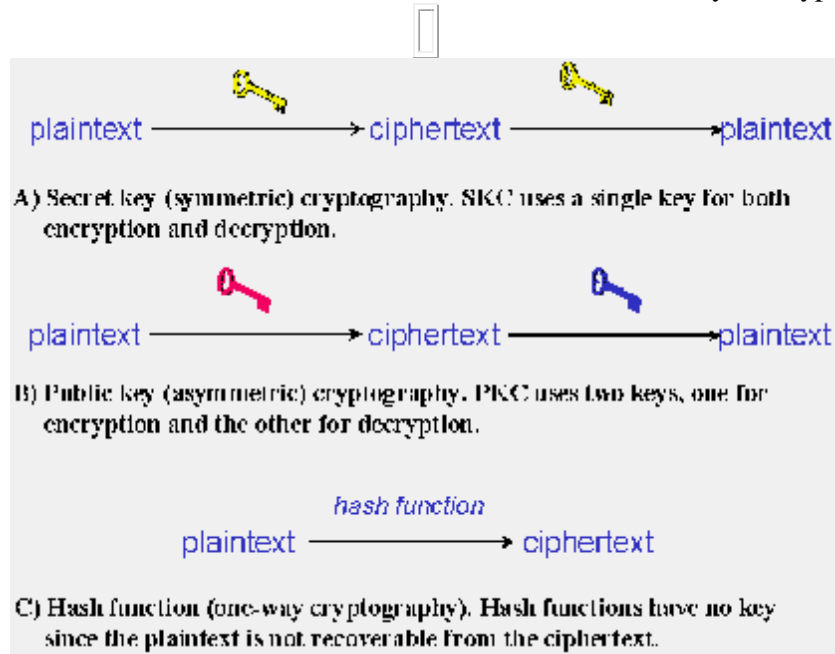


Fig 5.13: Three types of cryptography: secret-key, public key, and hash function

5.4 Network security

It consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or are assigned an ID and password or other authenticating information that allows them access to information and programs within their authority. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions. It does as its title explains: It secures the network, as well as protecting and overseeing operations being done. The most common and simple way of protecting a network resource is by assigning it a unique name and a corresponding password.

NETWORK SECURITY CONCEPTS

Network security starts with authenticating, commonly with a username and a password. Since this requires just one detail authenticating the user name —i.e. the password— this is sometimes termed one-factor authentication. With two-factor authentication, something the user 'has' is also used (e.g. a security token or 'dongle', an ATM card, or a mobile phone); and with three-factor authentication, something the user 'is' is also used (e.g. a fingerprint or retinal scan). Once authenticated, a firewall enforces access policies such as what services are allowed to be accessed by the network users. Though effective to prevent unauthorized access, this component may fail to check potentially harmful content such as computer worms or Trojans being transmitted over the network. Anti-virus software or an intrusion prevention system (IPS) help detect and inhibit the action of such malware. An anomaly-based intrusion detection system may also monitor the network like wire shark traffic and may be logged for audit purposes and for later high-level analysis. Communication between two hosts using a network may be encrypted to maintain privacy. Honeypots, essentially decoy network-accessible resources, may be deployed in a network as surveillance and early-warning tools, as the honeypots are not normally accessed for legitimate purposes. Techniques used by the attackers that attempt to compromise these decoy resources are studied during and after an attack to keep an eye on new exploitation techniques. Such analysis may be used to further tighten security of the actual network being protected by the honeypot. A honeypot can also direct an attacker's attention away from legitimate servers. A honeypot encourages attackers to spend their time and energy on the decoy server while distracting their attention from the data on the real server. Similar to a honeypot, a honeynet is a network set up with intentional vulnerabilities. Its purpose is also to invite attacks so that the attacker's methods can be studied and that information can be used to increase network security. A honeynet typically contains one or more honeypots.

SECURITY MANAGEMENT

Security management for networks is different for all kinds of situations. A home or small office may only require basic security while large businesses may require high-maintenance and advanced software and hardware to prevent malicious attacks from hacking and spamming.