COMPUTER NETWORKS (R20A0510)

LECTURE NOTES

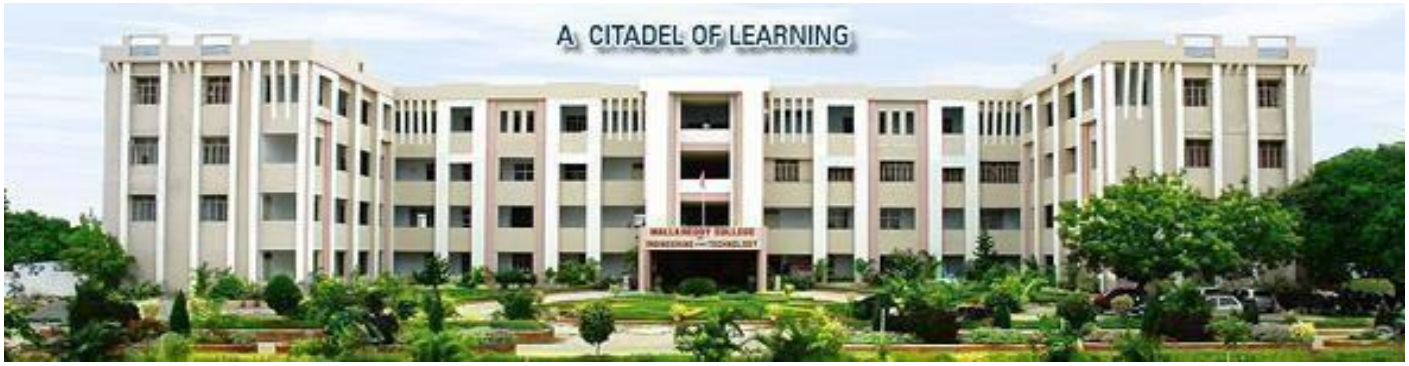
III B.TECH- I SEM ECE
2023-2024

Prepared by:

Mr M RAMANJANEYULU

Mr .CH. KIRAN KUMAR

Mr KDK AJAY



Department of Electronics & Communication Engineering

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

Recognized under 2(f) and 12 (B) of UGC ACT 1956

(Affiliated to JNTUH, Hyderabad, Approved by AICTE-Accredited by NBA&NAAC–'A'Grade-ISO9001:2015Certified)

Maisammaguda, Dhulapally (Post Via. Kompally), Secunderabad–500100,Telangana State, India

MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

III Year B.Tech. ECE- I Sem **L/T/P/C**
3/-/-/3

CORE ELECTIVE – I
(R20A0510) COMPUTER NETWORKS

COURSE OBJECTIVES:

1) Build an understanding of the fundamental concepts of computer networking.
2) Familiarize the student with the basic taxonomy and terminology of the computer networking area.
3) Introduce the student to advanced networking concepts, preparing the student for entry Advanced Courses in computer networking.
4) Allow the student to gain expertise in some specific areas of networking such as the design and Maintenance of individual networks.
5) To understand about Electronic mail, FTP, WWW, HTTP, Multimedia and Network security.

UNIT I

Introduction: Introduction to networks, Internet, Protocols and Standards, The OSI model, Layers in OSI Model, TCP/IP Suite, Addressing, Analog & Digital Signals

Physical Layer: Physical Layer Introduction, Digital Transmission, multiplexing, Transmission media, Circuit switched networks, Datagram networks, Virtual circuit networks, Switch& telephone network

UNIT II:

Data link layer: Introduction, Block coding, Cyclic codes, checksum, Framing, Flow and error control, Noiseless & Noisy channels, HDLC, Point to point protocols

Media Access Sub Layer: Random Access, Controlled access, channelization, IEEE Standards.

UNIT III:

Ethernet, Fast Ethernet, Giga bit Ethernet, wireless LANS, Connecting LANs, Backbone networks, Virtual LANs, Wireless WANs, SONET, frame relay, ATM.

UNIT IV:

Network Layer: Logical addressing, internetworking, tunneling, address mapping, ICMP, IGMP, Forwarding, Unicast routing protocols, multicast routing protocols.

UNIT V:

Transport Layer: Process to process delivery, TCP and UDP protocols, SCTP ,Data traffic , congestion, Congestion Control, QoS, integrated services, Differentiated services, QoS in Switched networks.

Application Layer: Domain name space, DNS in internet , Electronic Mail, FTP, WWW, HTTP, SNMP, Multi Media, Network Security.

TEXT BOOKS:

1) Data Communications and Networking- Behrouz A Forouzan Fourth Edition TMH, 2006.
2) Computer Networks- Andrew S Tanenbaum, 4th Edition, Pearson Education

REFERENCE BOOKS:

1) An Engineering approach to computer Networks- S.Keshav, 2nd Edition, Pearson Education
2) Computer and communication Networks- Nader F Mir, Pearson Education
3) Data and Computer Communications, G.S.Hura and M. Singhal, CRC Press, Taylor and Francis Group.
4) Data Communications and Computer Networks, P.C.Gupta, PHI
5) Computer Networking : A top-down Approach Featuring the Internet, James F.Kurose, K.W.Rose, 3rd Edition, Pearson Education

COURSE OUTCOMES:

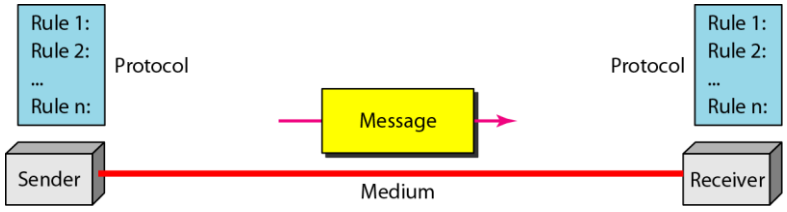
1) Have a good understanding of the OSI Reference Model and in particular have a good knowledge of Layers 1-3.
2) Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies
3) Specify and identify deficiencies in existing protocols, and then go onto formulate new and better protocols
4) Have an understanding of the issues surrounding Mobile and Wireless Networks.
5) Have a working knowledge of datagram and internet socket programming.

Introduction to Computer Networks

1.1 Data Communication: When we communicate, we are sharing information. This sharing can be local or remote. Between individuals, local communication usually occurs face to face, while remote communication takes place over distance.

1.1.1 Components:

A data communications system has five components.



1. Message. The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.

2. Sender. The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.

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

4. Transmission medium. The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves

5. Protocol. A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.

1.1.2 Data Representation:

Information today comes in different forms such as text, numbers, images, audio, and video.

Text:

In data communications, text is represented as a bit pattern, a sequence of bits (Os or Is). Different sets of bit patterns have been designed to represent text symbols. Each set is called a code, and the process of representing symbols is called coding. Today, the prevalent coding system is called Unicode, which uses 32 bits to represent a symbol or character used in any language in the world. The American Standard Code for Information Interchange (ASCII), developed some decades ago in the United States, now constitutes the first 127 characters in Unicode and is also referred to as Basic Latin.

Numbers:

Numbers are also represented by bit patterns. However, a code such as ASCII is not used to represent numbers; the number is directly converted to a binary number to simplify mathematical operations. Appendix B discusses several different numbering systems.

Images:

Images are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels (picture elements), where each pixel is a small dot. The size of the pixel depends on the *resolution.* For example, an image can be divided into 1000 pixels or 10,000 pixels. In the second case, there is a better representation of the image (better resolution), but more memory is needed to store the image. After an image is divided into pixels, each pixel is assigned a bit pattern. The size and the value of the pattern depend on the image. For an image made of only blackand- white dots (e.g., a chessboard), a I-bit pattern is enough to represent a pixel. If an image is not made of pure white and pure black pixels, you can increase the size of the bit pattern to include gray scale. For example, to show four levels of gray scale, you can use 2-bit patterns. A black pixel can be represented by 00, a dark gray pixel by 01, a light gray pixel by 10, and a white pixel by 11. There are several methods to represent color images. One method is called RGB, so called because each color is made of a combination of three primary colors: *red*, green, and blue. The intensity of each color is measured, and a bit pattern is assigned to it. Another method is called YCM, in which a color is made of a combination of three other primary colors: yellow, cyan, and magenta.

Audio:

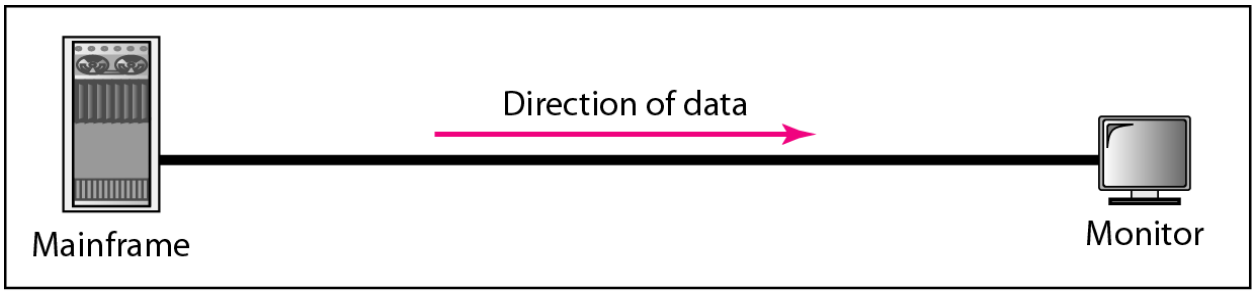
Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images. It is continuous, not discrete. Even when we use a microphone to change voice or music to an electric signal, we create a continuous signal. In Chapters 4 and 5, we learn how to change sound or music to a digital or an analog signal.

Video:

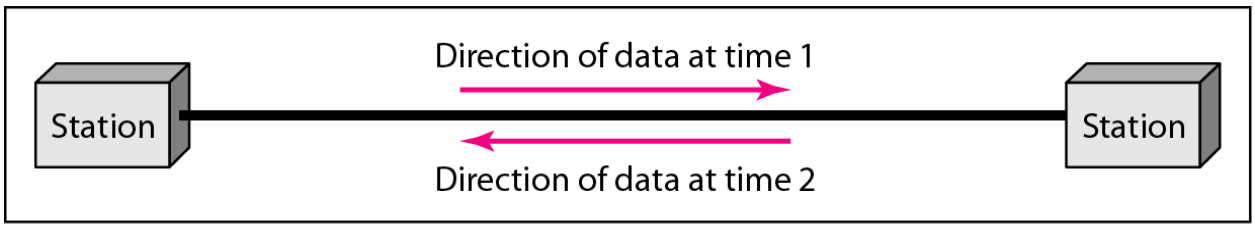
Video refers to the recording or broadcasting of a picture or movie. Video can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion. Again we can change video to a digital or an analog signal.

1.1.3 Data Flow

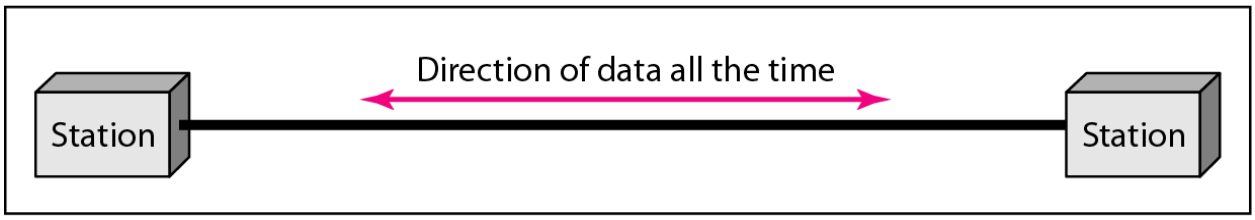
Communication between two devices can be simplex, half-duplex, or full-duplex as shown in Figure



a. Simplex



b. Half-duplex



c. Full-duplex

Simplex:

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 (see Figure a). Keyboards and traditional monitors are examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output. The simplex mode can use the entire capacity of the channel to send data in one direction.

Half-Duplex:

In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa The half-duplex mode is like a one-lane road with traffic allowed in both directions.

When cars are traveling in one direction, cars going the other way must wait. In a half-duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time. Walkie-talkies and CB (citizens band) radios are both half-duplex systems.

The half-duplex mode is used in cases where there is no need for communication in both directions at the same time; the entire capacity of the channel can be utilized for each direction.

Full-Duplex:

In full-duplex both stations can transmit and receive simultaneously (see Figure c). The full-duplex mode is like a tW<D-way street with traffic flowing in both directions at the same time. In full-duplex mode, si~nals going in one direction share the capacity of the link: with signals going in the other din~c~on. This sharing can occur in two ways: Either the link must contain two physically separate t:nmsmissiIDn paths, one for sending and the other for receiving; or the capacity of the ch:arillilel is divided between signals traveling in both directions. One common example of full-duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time. The full-duplex mode is used when communication in both directions is required all the time. The capacity of the channel, however, must be divided between the two directions.

1.2 NETWORKS

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.

1.2.1 Distributed Processing

Most networks use distributed processing, in which a task is divided among multiple computers. Instead of one single large machine being responsible for all aspects of a process, separate computers (usually a personal computer or workstation) handle a subset.

1.2.2 Network Criteria

A network must be able to meet a certain number of criteria. The most important of these are performance, reliability, and security.

Performance:

Performance can be measured in many ways, including transit time and response time.Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between an inquiry and a response. The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hardware, and the efficiency of the software. Performance is often evaluated by two networking metrics: throughput and delay. We often need more throughput and less delay. However, these two criteria are often contradictory. If we try to send more data to the network, we may increase throughput but we increase the delay because of traffic congestion in the network.

Reliability:

In addition to accuracy of delivery, network reliability is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.

Security:

Network security issues include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses.

1.2.3 Physical Structures:

Type of Connection

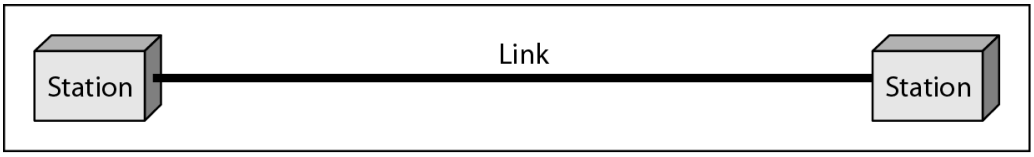
A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another. For visualization purposes, it is simplest to imagine any link as a line drawn between two points. For communication to occur, two devices must be connected in some way to the same link at the same time. There are two possible types of connections: point-to-point and multipoint.

Point-to-Point

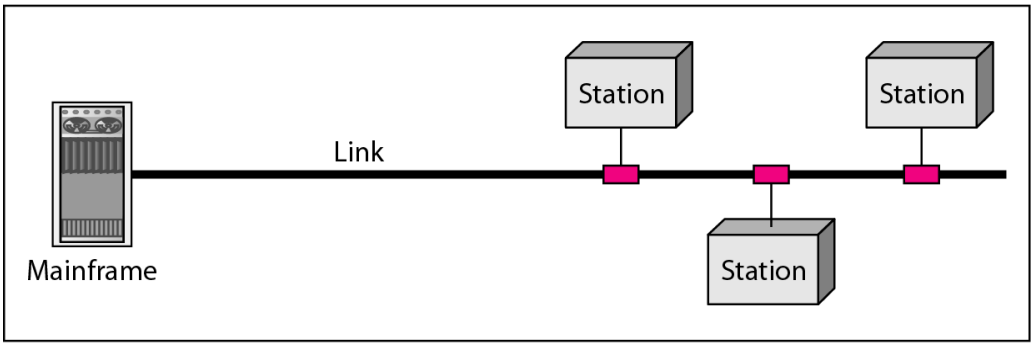
A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible. When you change television channels by infrared remote control, you are establishing a point-to-point connection between the remote control and the television's control system.

Multipoint

A multipoint (also called multidrop) connection is one in which more than two specific devices share a single link. In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a *spatially shared* connection. If users must take turns, it is a *timeshared* connection.



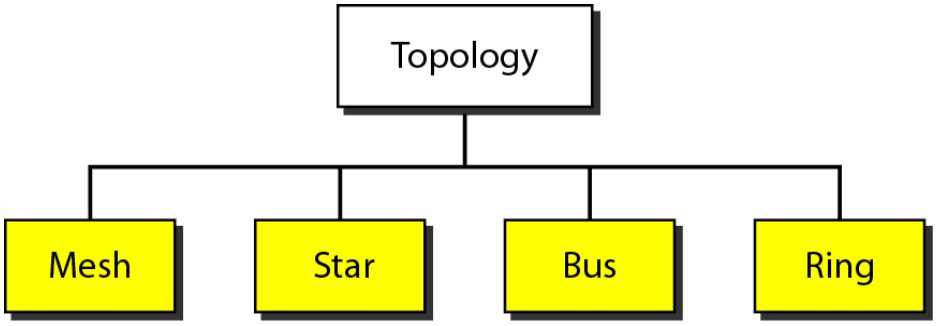
a. Point-to-point



b. Multipoint

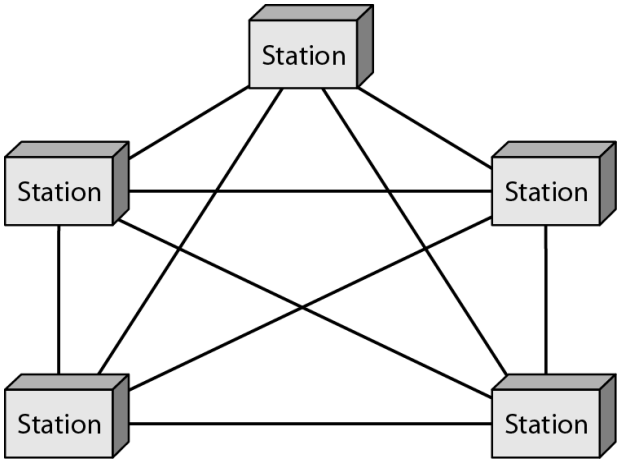
1.2.3.1 Physical Topology

The term *physical topology* refers to the way in which a network is laid out physically. One or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another. There are four basic topologies possible: mesh, star, bus, and ring



Mesh: In a mesh topology, every device has a dedicated point-to-point link to every other device. The term *dedicated* means that the link carries traffic only between the two devices it connects. To find the number of physical links in a fully connected mesh network with *n* nodes, we first consider that each node must be connected to every other node. Node 1 must be connected to *n* - I nodes, node 2 must be connected to *n* – 1 nodes, and finally node *n* must be connected to *n* - 1 nodes. We need *n(n* - 1) physical links. However, if each physical link allows communication in both directions (duplex mode), we can divide the number of links by 2. In other words, we can say that in a mesh topology, we need *n(n* -1) /2 duplex-mode links.

To accommodate that many links, every device on the network must have *n* – 1 input/output *(VO)* ports to be connected to the other *n* - 1 stations.



Advantages:

1. The use of dedicated links guarantees that each connection can carry its own data load, thus eliminating the traffic problems that can occur when links must be shared by multiple devices.
2. A mesh topology is robust. If one link becomes unusable, it does not incapacitate the entire system. Third, there is the advantage of privacy or security. When every message travels along a dedicated line, only the intended recipient sees it. Physical boundaries prevent other users from gaining access to messages. Finally, point-to-point links make fault identification and fault isolation easy. Traffic can be routed to avoid links with suspected problems. This facility enables the network manager to discover the precise location of the fault and aids in finding its cause and solution.

Disadvantages:

1. Disadvantage of a mesh are related to the amount of cabling because every device must be connected to every other device, installation and reconnection are difficult.
2. Second, the sheer bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate. Finally, the hardware required to connect each link (I/O ports and cable) can be prohibitively expensive.

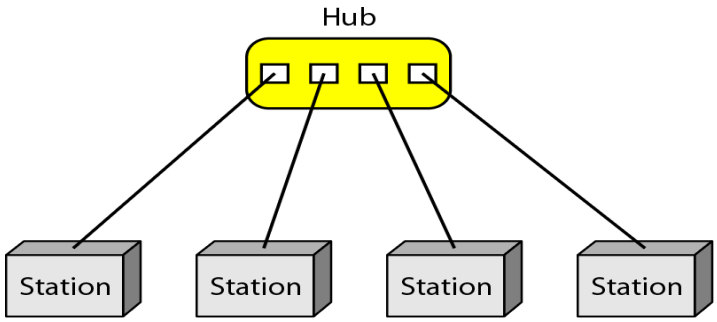
For these reasons a mesh topology is usually implemented in a limited fashion, for example, as a backbone connecting the main computers of a hybrid network that can include several other topologies.

Star Topology:

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

A star topology is less expensive than a mesh topology. In a star, each device needs only one link and one I/O port to connect it to any number of others. This factor also makes it easy to install and reconfigure. Far less cabling needs to be housed, and additions, moves, and deletions involve only one connection: between that device and the hub.

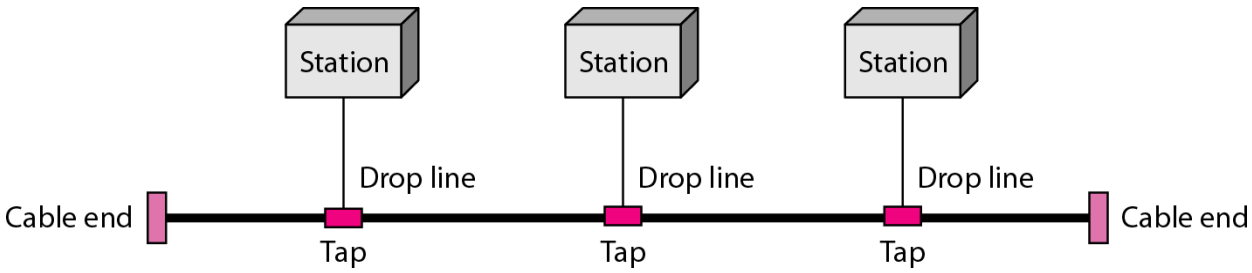
Other advantages include robustness. If one link fails, only that link is affected. All other links remain active. This factor also lends itself to easy fault identification and fault isolation. As long as the hub is working, it can be used to monitor link problems and bypass defective links.



One big disadvantage of a star topology is the dependency of the whole topology on one single point, the hub. If the hub goes down, the whole system is dead. Although a star requires far less cable than a mesh, each node must be linked to a central hub. For this reason, often more cabling is required in a star than in some other topologies (such as ring or bus).

Bus Topology:

The preceding examples all describe point-to-point connections. A **bus topology,** on the other hand, is multipoint. One long cable acts as a **backbone** to link all the devices in a network



Nodes are connected to the bus cable by drop lines and taps. A drop line is a connection running between the device and the main cable. A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core. As a signal travels along the backbone, some of its energy is transformed into heat. Therefore, it becomes weaker and weaker as it travels farther and farther. For this reason there is a limit on the number of taps a bus can support and on the distance between those taps.

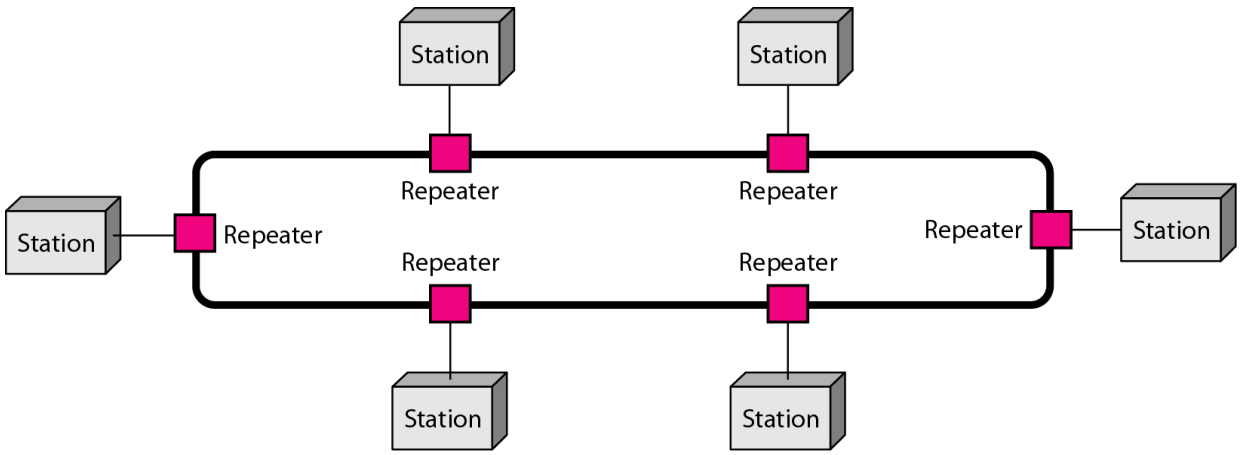
Advantages of a bus topology include ease of installation. Backbone cable can be laid along the most efficient path, then connected to the nodes by drop lines of various lengths. In this way, a bus uses less cabling than mesh or star topologies. In a star, for example, four network devices in the same room require four lengths of cable reaching all the way to the hub. In a bus, this redundancy is eliminated. Only the backbone cable stretches through the entire facility. Each drop line has to reach only as far as the nearest point on the backbone.

Disadvantages include difficult reconnection and fault isolation. A bus is usually designed to be optimally efficient at installation. It can therefore be difficult to add new devices. Signal reflection at the taps can cause degradation in quality. This degradation can be controlled by limiting the number and spacing of devices connected to a given length of cable. Adding new devices may therefore require modification or replacement of the backbone.

In addition, a fault or break in the bus cable stops all transmission, even between devices on the same side of the problem. The damaged area reflects signals back in the direction of origin, creating noise in both directions.

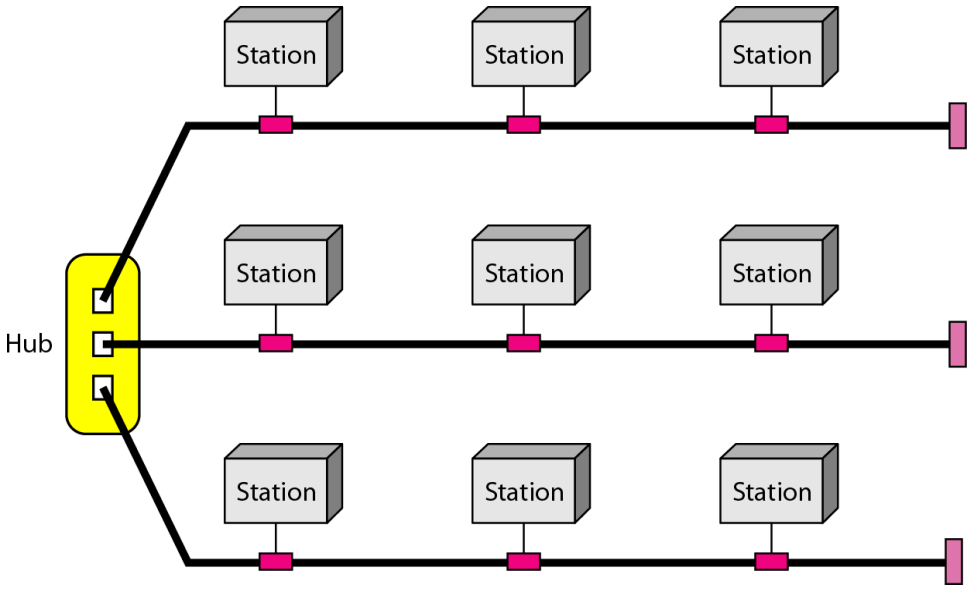
Bus topology was the one of the first topologies used in the design of early local area networks. Ethernet LANs can use a bus topology, but they are less popular.

Ring Topology In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along



A ring is relatively easy to install and reconfigure. Each device is linked to only its immediate neighbors (either physically or logically). To add or delete a device requires changing only two connections. The only constraints are media and traffic considerations (maximum ring length and number of devices). In addition, fault isolation is simplified. Generally in a ring, a signal is circulating at all times. If one device does not receive a signal within a specified period, it can issue an alarm. The alarm alerts the network operator to the problem and its location.

However, unidirectional traffic can be a disadvantage. In a simple ring, a break in the ring (such as a disabled station) can disable the entire network. This weakness can be solved by using a dual ring or a switch capable of closing off the break. Ring topology was prevalent when IBM introduced its local-area network Token Ring. Today, the need for higher-speed LANs has made this topology less popular. Hybrid Topology A network can be hybrid. For example, we can have a main star topology with each branch connecting several stations in a bus topology as shown in Figure



1.2.4 Categories of Networks

Local Area Networks:

Local area networks, generally called LANs, are privately-owned networks within a single building or campus of up to a few kilometres in size. They are widely used to connect personal computers and workstations in company offices and factories to share resources (e.g., printers) and exchange information. LANs are distinguished from other kinds of networks by three characteristics:

(1) Their size,

(2) Their transmission technology, and

(3) Their topology.

LANs are restricted in size, which means that the worst-case transmission time is bounded and known in advance. Knowing this bound makes it possible to use certain kinds of designs that would not otherwise be possible. It also simplifies network management. LANs may use a transmission technology consisting of a cable to which all the machines are attached, like the telephone company party lines once used in rural areas. Traditional LANs run at speeds of 10 Mbps to 100 Mbps, have low delay (microseconds or nanoseconds), and make very few errors. Newer LANs operate at up to 10 Gbps Various topologies are possible for broadcast LANs. Figure1 shows two of them. In a bus (i.e., a linear cable) network, at any instant at most one machine is the master and is allowed to transmit. All other machines are required to refrain from sending. An arbitration mechanism is needed to resolve conflicts when two or more machines want to transmit simultaneously. The arbitration mechanism may be centralized or distributed. IEEE 802.3, popularly called Ethernet, for example, is a bus-based broadcast network with decentralized control, usually operating at 10 Mbps to 10 Gbps. Computers on an Ethernet can transmit whenever they want to; if two or more packets collide, each computer just waits a random time and tries again later.

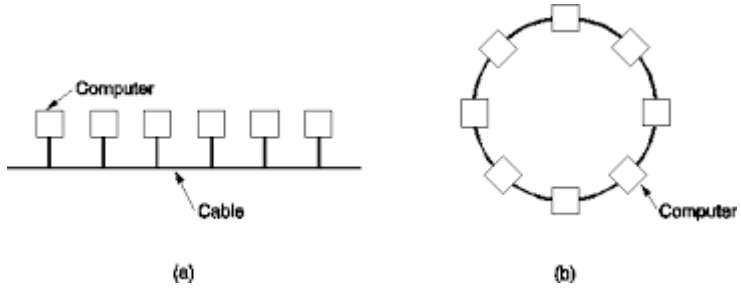


Fig.1: Two broadcast networks . (a) Bus. (b) Ring.

A second type of broadcast system is the ring. In a ring, each bit propagates around on its own, not waiting for the rest of the packet to which it belongs. Typically, each bit circumnavigates the entire ring in the time it takes to transmit a few bits, often before the complete packet has even been transmitted. As with all other broadcast systems, some rule is needed for arbitrating simultaneous accesses to the ring. Various methods, such as having the machines take turns, are in use. IEEE 802.5 (the IBM token ring), is a ring-based LAN operating at 4 and 16 Mbps. FDDI is another example of a ring network.

Metropolitan Area Network (MAN):

Metropolitan Area Network:

A metropolitan area network, or MAN, covers a city. The best-known example of a MAN is the cable television network available in many cities. This system grew from earlier community antenna systems used in areas with poor over-the-air television reception. In these early systems, a large antenna was placed on top of a nearby hill and signal was then piped to the subscribers' houses. At first, these were locally-designed, ad hoc systems. Then companies began jumping into the business, getting contracts from city governments to wire up an entire city. The next step was television programming and even entire channels designed for cable only. Often these channels were highly specialized, such as all news, all sports, all cooking, all gardening, and so on. But from their inception until the late 1990s, they were intended for television reception only. To a first approximation, a MAN might look something like the system shown in Fig. In this figure both television signals and Internet are fed into the centralized head end for subsequent distribution to people's homes. Cable television is not the only MAN. Recent developments in high-speed wireless Internet access resulted in another MAN, which has been standardized as IEEE 802.16.

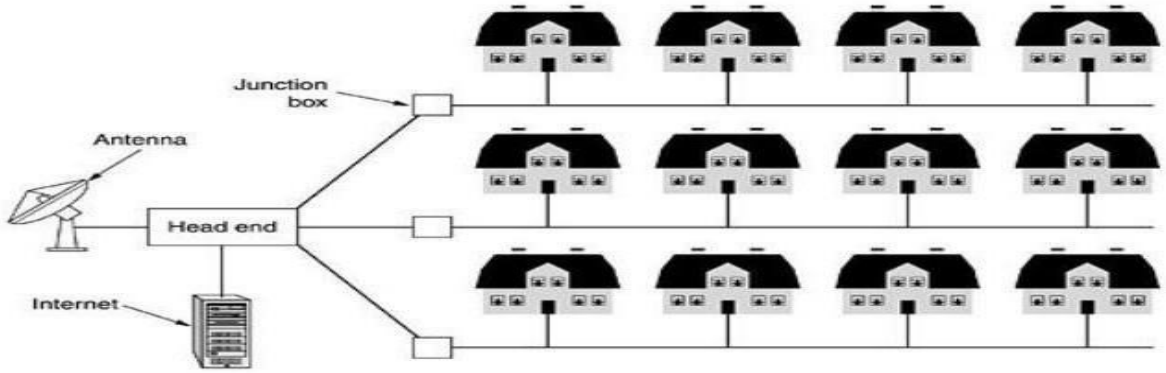


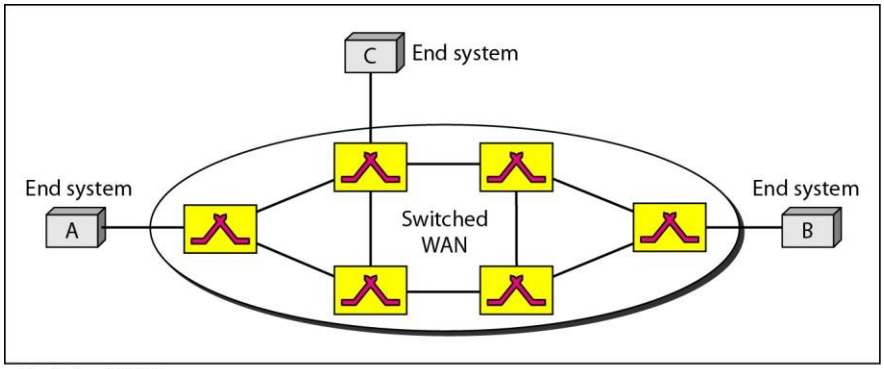
Fig.2: Metropolitan area network based on cable TV.

A MAN is implemented by a standard called DQDB (Distributed Queue Dual Bus) or IEEE 802.16. DQDB has two unidirectional buses (or cables) to which all the computers are attached.

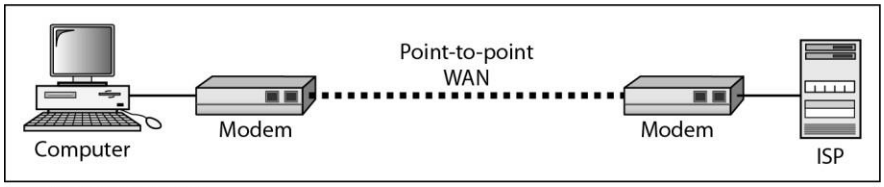
Wide Area Network (WAN).

Wide Area Network:

A wide area network, or WAN, spans a large geographical area, often a country or continent. It contains a collection of machines intended for running user (i.e., application) programs. These machines are called as hosts. The hosts are connected by a communication subnet, or just subnet for short. The hosts are owned by the customers (e.g., people's personal computers), whereas the communication subnet is typically owned and operated by a telephone company or Internet service provider. The job of the subnet is to carry messages from host to host, just as the telephone system carries words from speaker to listener.



a. Switched WAN



b. Point-to-point WAN

Separation of the pure communication aspects of the network (the subnet) from the application aspects (the hosts), greatly simplifies the complete network design. In most wide area networks, the subnet consists of two distinct components: transmission lines and switching elements.

Transmission lines move bits between machines. They can be made of copper wire, optical fiber,or even radio links. In most WANs, the network contains numerous transmission lines, each one connecting a pair of routers. If two routers that do not share a transmission line wish to communicate, they must do this indirectly, via other routers. When a packet is sent from one

router to another via one or more intermediate routers, the packet is received at each intermediate router in its entirety, stored there until the required output line is free, and then forwarded. A subnet organized according to this principle is called a store-and-forward or packet-switched subnet. Nearly all wide area networks (except those using satellites) have store-and-forward subnets. When the packets are small and all the same size, they are often called cells.

The principle of a packet-switched WAN is so important. Generally, when a process on some host has a message to be sent to a process on some other host, the sending host first cuts the message into packets, each one bearing its number in the sequence. These packets are then injected into the network one at a time in quick succession. The packets are transported individually over the network and deposited at the receiving host, where they are reassembled into the original message and delivered to the receiving process. A stream of packets resulting from some initial message is illustrated in Fig.

In this figure, all the packets follow the route ACE, rather than ABDE or ACDE. In some networks all packets from a given message must follow the same route; in others each packed is routed separately. Of course, if ACE is the best route, all packets may be sent along it, even if each packet is individually routed.

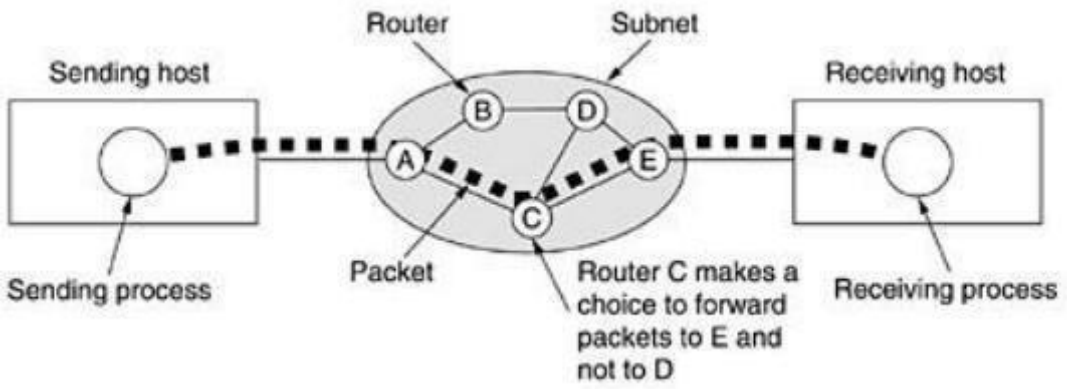


Fig.3.1: A stream of packets from sender to receiver.

Not all WANs are packet switched. A second possibility for a WAN is a satellite system. Each router has an antenna through which it can send and receive. All routers can hear the output from the satellite, and in some cases they can also hear the upward transmissions of their fellow routers to the satellite as well. Sometimes the routers are connected to a substantial point-to-point subnet, with only some of them having a satellite antenna. Satellite networks are inherently broadcast and are most useful when the broadcast property is important.

1.3 THE INTERNET

The Internet has revolutionized many aspects of our daily lives. It has affected the way we do business as well as the way we spend our leisure time. Count the ways you've used the Internet recently. Perhaps you've sent electronic mail (e-mail) to a business associate, paid a utility bill, read a newspaper from a distant city, or looked up a local movie schedule-all by using the Internet. Or maybe you researched a medical topic, booked a hotel reservation, chatted with a fellow Trekkie, or comparison-shopped for a car. The Internet is a communication system that has brought a wealth of information to our fingertips and organized it for our use.

A Brief History

A network is a group of connected communicating devices such as computers and printers. An internet (note the lowercase letter i) is two or more networks that can communicate with each other. The most notable internet is called the Internet (uppercase letter I), a collaboration of more than hundreds of thousands of interconnected networks. Private individuals as well as various organizations such as government agencies, schools, research facilities, corporations, and libraries in more than 100 countries use the Internet. Millions of people are users. Yet this extraordinary communication system only came into being in 1969.

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

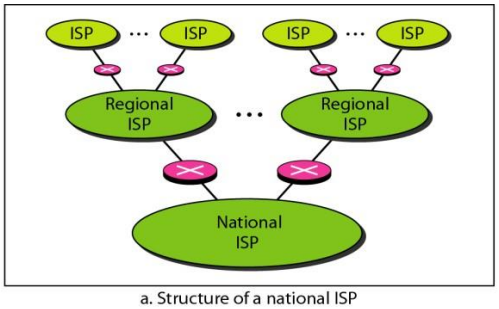
In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for ARPANET, a small network of connected computers. The idea was that each host computer (not necessarily from the same manufacturer) would be attached to a specialized computer, called an *inteiface message processor* (IMP). The IMPs, in tum, would be connected to one another. Each IMP had to be able to communicate with other IMPs as well as with its own attached host. By 1969, ARPANET was a reality. Four nodes, at the University of California at Los Angeles (UCLA), the University of California at Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah, were connected via the IMPs to form a network. Software called the *Network Control Protocol* (NCP) provided communication between the hosts.

In 1972, Vint Cerf and Bob Kahn, both of whom were part of the core ARPANET group,

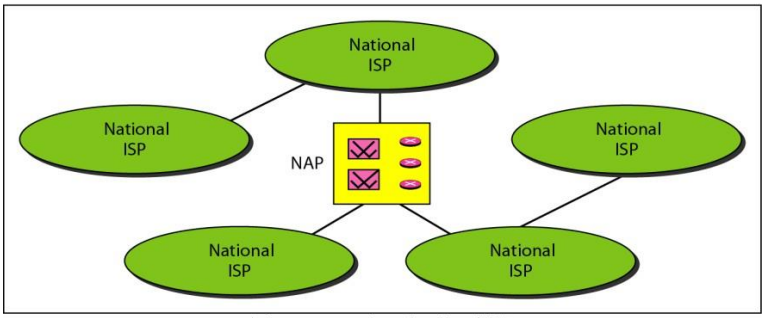
collaborated on what they called the *Internetting Projec1*. Cerf and Kahn's landmark 1973 paper outlined the protocols to achieve end-to-end delivery of packets. This paper on Transmission Control Protocol (TCP) included concepts such as encapsulation, the datagram, and the functions of a gateway. Shortly thereafter, authorities made a decision to split TCP into two protocols: Transmission Control Protocol (TCP) and Internetworking Protocol (lP). IP would handle datagram routing while TCP would be responsible for higher-level functions such as segmentation, reassembly, and error detection. The internetworking protocol became known as TCPIIP.

The Internet Today

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 (lSPs). There are international service providers, national service providers, regional service providers, and local service providers. The Internet today is run by private companies, not the government. Figure 1.13 shows a conceptual (not geographic) view of the Internet.



a. Structure of a national ISP



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 SprintLink, PSINet, 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. Note that in this sense, a local ISP can be a company that just provides Internet services, a corporation with a network that supplies services to its own employees, or a nonprofit organization, such as a college or a university, that runs its own network. Each of these local ISPs can be connected to a regional or national service provider.

1.4 PROTOCOLS AND STANDARDS

Protocols:

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. For communication to occur, the entities must agree on a protocol. A protocol is a set of rules that govern data communications. A protocol defines what is communicated, how it is communicated, and when it is communicated. The key elements of a protocol are syntax, semantics, and timing.

o Syntax. The term *syntax* refers to the structure or format of the data, meaning the order in which they are presented. For example, a simple protocol might expect the first 8 bits of data to be the address of the sender, the second 8 bits to be the address of the receiver, and the rest of the stream to be the message itself.

o 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? For example, does an address identify the route to be taken or the final destination of the message?

o Timing. The term *timing* refers to two characteristics: when data should be sent and how fast they can be sent. For example, if a sender produces data at 100 Mbps but the receiver can process data at only 1 Mbps, the transmission will overload the receiver and some data will be lost.

Standards

Standards are essential in creating and maintaining an open and competitive market for equipment manufacturers and in guaranteeing national and international interoperability of data and telecommunications technology and processes. 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 communications.

Data communication standards fall into two categories: *de facto* (meaning "by fact" or "by convention") and *de jure* (meaning "by law" or "by regulation").

o De facto. Standards that have not been approved by an organized body but have been adopted as standards through widespread use are de facto standards. De facto standards are often established originally by manufacturers who seek to define the functionality of a new product or technology.

o De jure. Those standards that have been legislated by an officially recognized body are de jure standards.

Standards are developed by cooperation among standards creation committees, forums, and government regulatory agencies.

Standards Creation Committees:

a) International Standards Organization (ISO)

b) International Telecommunications Union (ITU)

c) American National Standards Institute (ANSI)
d) Institute of Electrical and Electronics Engineers (IEEE)
e) Electronic Industries Association (EIA)

a) International Standards Organization (ISO)

A multinational body whose membership is drawn mainly from the standards creation committees of various governments throughout the world. Dedicated to worldwide agreement on international standards in a variety field. Currently includes 82 memberships industrialized nations. Aims to facilitate the international exchange of goods and services by providing models for compatibility, improved quality, increased quality, increased productivity and decreased prices.

b) International Telecommunications Union (ITU)

Also known as International Telecommunications Union-Telecommunication Standards Sector (ITU-T). An international standards organization related to the United Nations that develops standards for telecommunications. Two popular standards developed by ITU-T are:

i) V series – transmission over phone lines

ii) X series – transmission over public digital networks, email and directory services and ISDN.

c) American National Standards Institute (ANSI)

A non-profit corporation not affiliated with US government. ANSI members include professional societies, industry associations, governmental and regulatory bodies, and consumer groups. Discussing the internetwork planning and engineering, ISDN services, signaling, and architecture and optical hierarchy.

d) Institute of Electrical and Electronics Engineers (IEEE)

The largest national professional group involved in developing standards for computing, communication, electrical engineering, and electronics. Aims to advance theory, creativity and product quality in the fields of electrical engineering, electronics and radio. It sponsored an important standard for local area networks called Project 802 (eg. 802.3, 802.4 and 802.5 standards.)

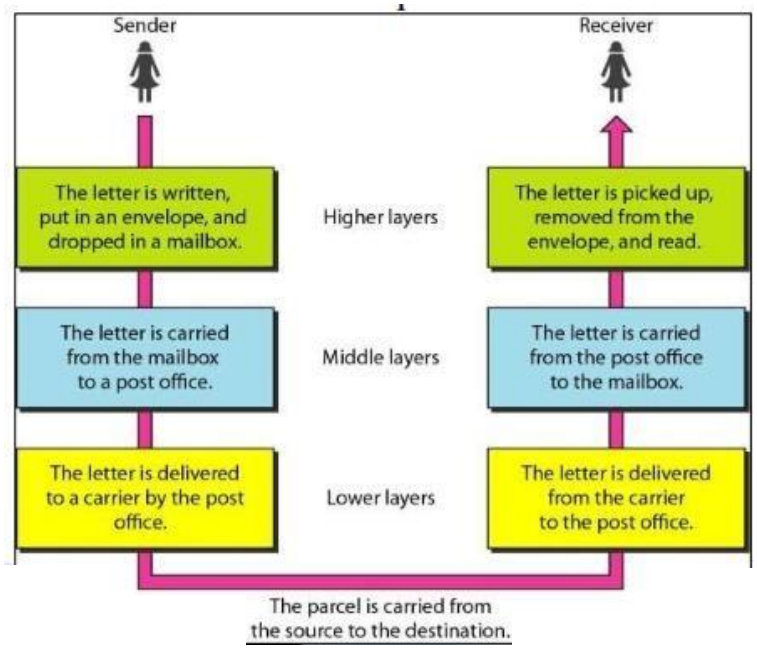
e) Electronic Industries Association (EIA)

An association of electronics manufacturers in the US. Provide activities include public awareness education and lobbying efforts in addition to standards development. Responsible for developing the EIA-232-D and EIA-530 standards.

INTERNET STANDARDS

An Internet standard is a thoroughly tested specification that is useful to and adhered to by those who work with the Internet. It is a formalized regulation that must be followed. There is a strict procedure by which a specification attains Internet standard status. A specification begins as an Internet draft. An Internet draft is a working document (a work in progress) with no official status and a six-month lifetime. Upon recommendation from the Internet authorities, a draft may be published as a Request for Comment (RFC). Each RFC is edited, assigned a number, and made available to all interested parties. RFCs go through maturity levels and are categorized according to their requirement level.

1.5 LAYERED TASKS



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. Below Figure shows the steps in this task.

Sender, Receiver, and Carrier

In Figure we have a sender, a receiver, and a carrier that transports the letter. There is a hierarchy of tasks.

At the Sender Site

Let us first describe, in order, the activities that take place at the sender site.

o Higher layer. The sender writes the letter, inserts the letter in an envelope, writes the sender and receiver addresses, and drops the letter in a mailbox.

o Middle layer. The letter is picked up by a letter carrier and delivered to the post office. o Lower layer. The letter is sorted at the post office; a carrier transports the letter.

0n the Way: The letter is then on its way to the recipient. On the way to the recipient's local post office, the letter may actually go through a central office. In addition, it may be transported by truck, train, airplane, boat, or a combination of these.

At the Receiver Site

- o Lower layer. The carrier transports the letter to the post office.
- o Middle layer. The letter is sorted and delivered to the recipient's mailbox.
- o Higher layer. The receiver picks up the letter, opens the envelope, and reads it.

1.6 The OSI Reference Model

The OSI model (minus the physical medium) is shown in Fig. This model is based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various layers (Day and Zimmermann, 1983). It was revised in 1995(Day, 1995). The model is called the ISO-OSI (Open Systems Interconnection) Reference Model because it deals with connecting open systems—that is, systems that are open for communication with other systems.

The OSI model has seven layers. The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.

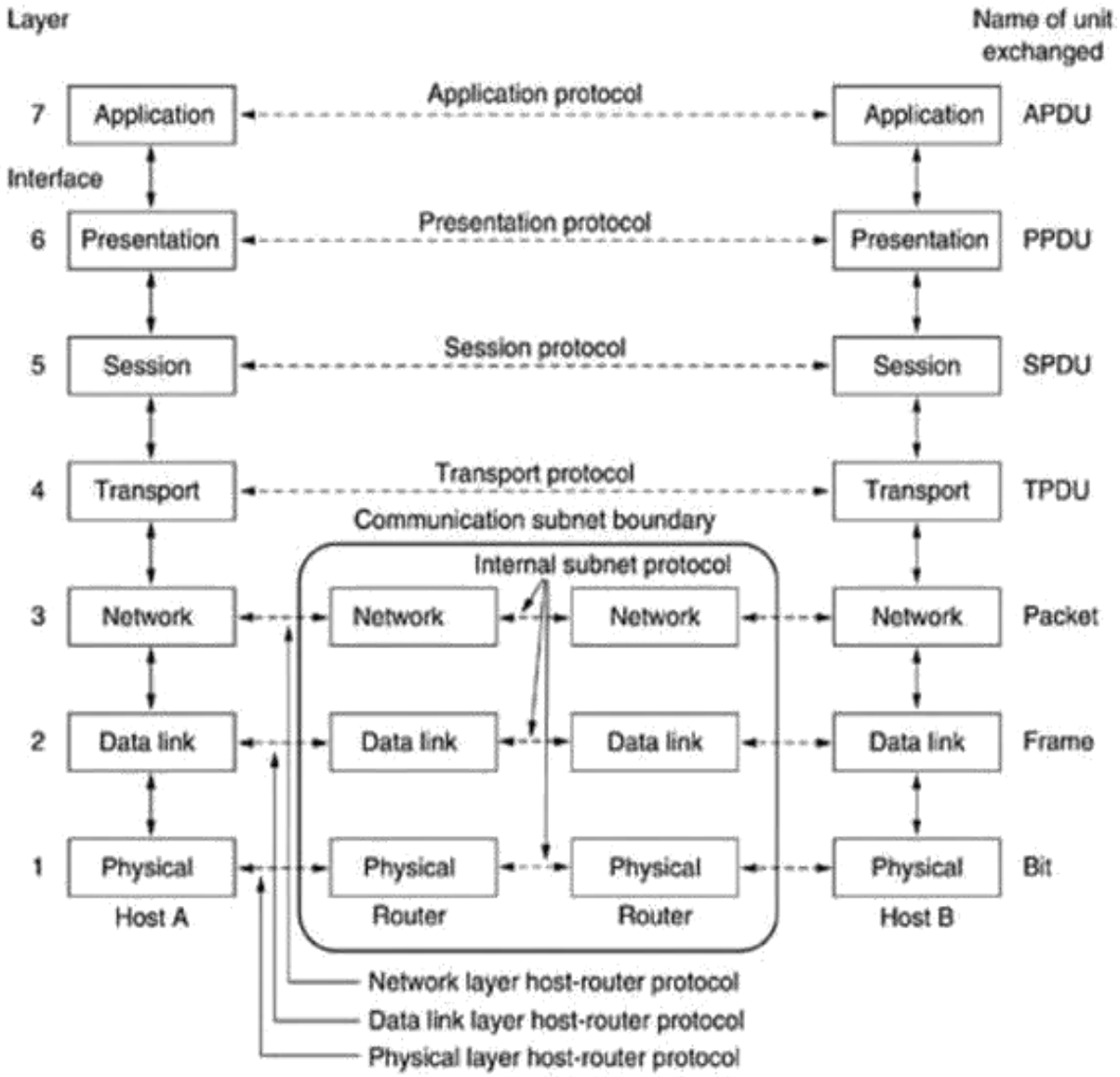


Fig.4: The OSI reference model

The physical layer:

The physical layer is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit, it is received by the other side as a 1 bit, not as a 0 bit.

- The physical layer coordinates the functions required to carryabit stream over a physical medium.
- It deals with the mechanical and electrical specifications oftheinterface and transmission medium.
- It also defines the procedures and functions that physical devicesand interfaces have to perform for transmission to Occur.

Physical Layer also performs following responsibility

1) **Physical characteristics of interfaces and medium.** The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.
2) **Representation of bits**. The physical layer data consists of a stream of bits (sequence of Os or 1s) . The physical layer defines the type of encoding(how Os and Is are changed to signals).
3) **Data rate:** The transmission rate-the number of bits sent each second-is also defined by the physical layer..
4) **Synchronization of bits**. Sender & Receiver must have same bit rate **5.Line configuration**. point-to-point configuration . multipoint configuration
5) **Physical topology**. mesh topology (every device is connected to every other device), a star topology (devices are connected through a central device), a ring topology (each device is connected to the next, forming aring), a bus topology (every device is on a common link), or a hybrid topology
6) **Transmission mode** simplex, half-duplex, or full-duplex.

The Data Link Layer:

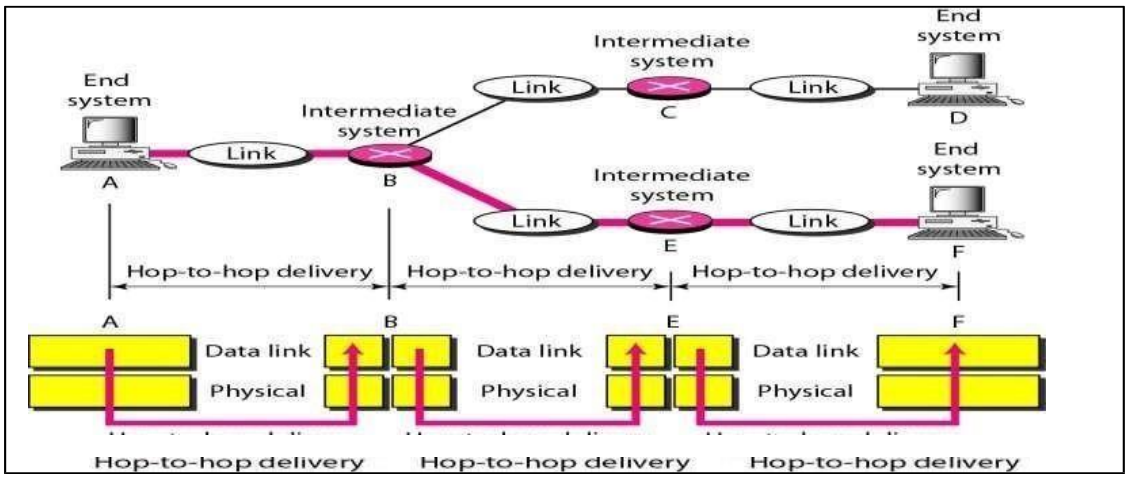
The main task of the data link layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer. It accomplishes this task by having the sender break up the input data into data frames (typically a few hundred or a few

thousand bytes) and transmits the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by sending back an acknowledgement frame.

Another issue that arises in the data link layer (and most of the higher layers as well) is how to keep a fast transmitter from drowning a slow receiver in data. Some traffic regulation mechanism is often needed to let the transmitter know how much buffer space the receiver has at the moment. Frequently, this flow regulation and the error handling are integrated.

- The data link layer transforms the physical layer, a raw transmission facility, to a reliable link.
- It makes the physical layer appear error-free to the upper layer
- The data link layer is responsible for moving frames from one hop (node) to the next

- Hop-to-hop delivery



- As the figure shows, communication at the data link layer occurs between two adjacent nodes.
- To send data from A to F, three partial deliveries are made. First, the data link layer at A sends a frame to the data link layer at B (a router).
- Second, the data link layer at B sends a new frame to the data link layer at E.

- Finally, the data link layer at E sends a new frame to the data link layer at F.
- Other responsibilities of the Data Link layer include the following:

1) Framing. The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
2) Physical addressing. If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.
3) Flow control. If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.
4) Error control. The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames.
5) Access control. When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

The Network Layer:

The network layer controls the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be based on static tables that are "wired into" the network and rarely changed. They can also be determined at the start of each conversation, for example, a terminal session (e.g., a login to a remote machine). Finally, they can be highly dynamic, being determined anew for each packet, to reflect the current network load.

If too many packets are present in the subnet at the same time, they will get in one another's way, forming bottlenecks. The control of such congestion also belongs to the network layer. More generally, the quality of service provided (delay, transit time, jitter, etc.) is also a network layer issue.

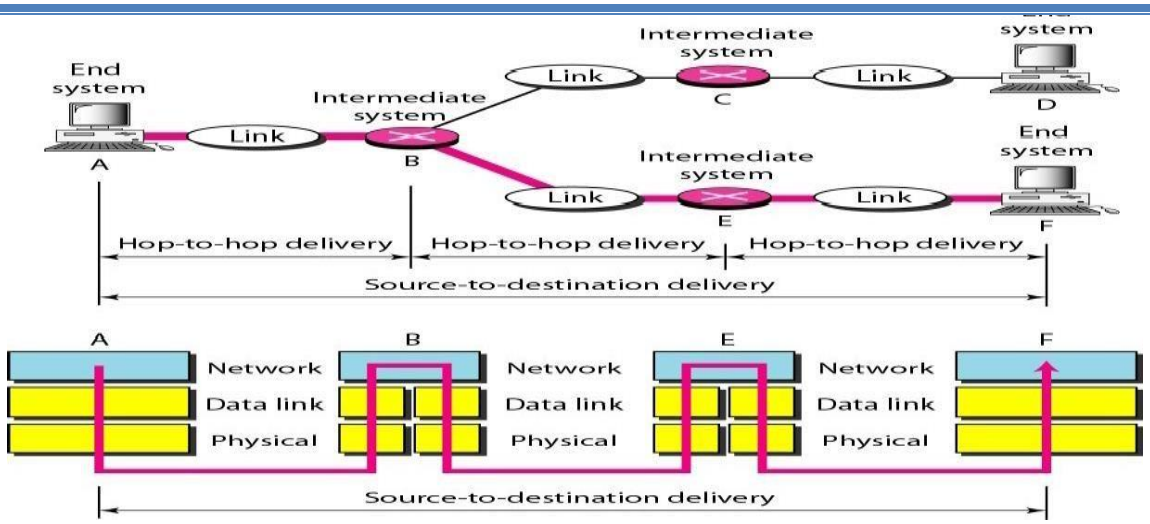
When a packet has to travel from one network to another to get to its destination, many problems

can arise. The addressing used by the second network may be different from the first one. The second one may not accept the packet at all because it is too large. The protocols may differ, and so on. It is up to the network layer to overcome all these problems to allow heterogeneous networks to be interconnected. In broadcast networks, the routing problem is simple, so the network layer is often thin or even nonexistent.

- The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links).
- If two systems are connected to the same link, there is usually no need for a network layer.
- However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery

Other responsibilities of the network layer include the following:

1) Logical addressing: The network layer adds a header to the packet coming from the upper layer that, among other things, includes the logical addresses of the sender and receiver. We discuss logical addresses later in this chapter.
2) Routing. When independent networks or links are connected to create internetworks (network of networks) or a large network, the connecting devices (called routers or switches) route or switch the packets to their final destination. One of the functions of the network layer is to provide this mechanism. Source-to-destination delivery



The network layer at A sends the packet to the network layer at B. When the packet arrives at router B, the router makes a decision based on the final destination (F) of the packet.

The Transport Layer:

The basic function of the transport layer is to accept data from above, split it up into smaller units if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. Furthermore, all this must be done efficiently and in a way that isolates the upper layers from the inevitable changes in the hardware technology. The transport layer also determines what type of service to provide to the session layer, and, ultimately, to the users of the network. The most popular type of transport connection is an error-free point-to-point channel that delivers messages or bytes in the order in which they were sent. However, other possible kinds of transport service are the transporting of isolated messages, with no guarantee about the order of delivery, and the broadcasting of messages to multiple destinations. The type of service is determined when the connection is established.

The transport layer is a true end-to-end layer, all the way from the source to the destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages. In the lower layers, the protocols are between each machine and its immediate neighbors, and not between the ultimate source and destination machines, which may be separated by many routers.

The transport layer is responsible for process-to-process delivery of the entire message. A process is an application program running on a host.

- The transport layer ensures that the whole message arrives intact and in order, overseeing both

error control and flow control at the

Other responsibilities of the transport layer include the following:

Service-point addressing. Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header must therefore include a type of address called a service-point address (or port address). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.

Segmentation and reassembly. A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.

Connection control. The transport layer can be either connectionless or connection oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated

Flow control. Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.

Error control. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission.

The Session Layer:

The session layer allows users on different machines to establish sessions between them. Sessions offer various services, including dialog control (keeping track of whose turn it is to transmit), token management (preventing two parties from attempting the same critical operation at the same time), and synchronization (check pointing long transmissions to allow them to

continue from where they were after a crash).

- The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes.
- The session layer is the network dialog controller.
- It establishes, maintains, and synchronizes the interaction among communicating systems.

Specific responsibilities of the session layer include the following:

Dialog control: It allows the communication between two processes to take place in either half duplex (one way at a time) or full-duplex (two ways at a time) mode.

Synchronization: The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent. layers.

The Presentation Layer:

The presentation layer is concerned with the syntax and semantics of the information transmitted. In order to make it possible for computers with different data representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used "on the wire." The presentation layer manages these abstract data structures and allows higher-level data structures (e.g., banking records), to be defined and exchanged.

- The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems.
- The presentation layer is responsible for translation, compression, and encryption.

Specific responsibilities of the presentation layer include the following:

- **Translation.** The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The infomation must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.

- **Encryption.** To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.

Compression. Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

The Application Layer:

The application layer contains a variety of protocols that are commonly needed by users. One widely-used application protocol is HTTP (Hypertext Transfer Protocol), which is the basis for the World Wide Web. When a browser wants a Web page, it sends the name of the page it wants to the server using HTTP. The server then sends the page back. Other application protocols are used for file transfer, electronic mail, and network news.

1.7 The TCP/IP Reference Model:

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

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

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

1. Host-to-Network Layer
2. Internet Layer

3. Transport Layer

4. Application Layer

Host-to-Network Layer:

The TCP/IP reference model does not really say much about what happens here, except to point out that the host has to connect to the network using some protocol so it can send IP packets to it. This protocol is not defined and varies from host to host and network to network.

Internet Layer:

This layer, called the internet layer, is the linchpin that holds the whole architecture together. Its job is to permit hosts to inject packets into any network and have they travel independently to the destination (potentially on a different network). They may even arrive in a different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired. Note that "internet" is used here in a generic sense, even though this layer is present in the Internet.

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

The Transport Layer:

The layer above the internet layer in the TCP/IP model is now usually called the transport layer. It is designed to allow peer entities on the source and destination hosts to carry on a conversation, just as in the OSI transport layer. Two end-to-end transport protocols have been defined here. The first one, TCP (Transmission Control Protocol), is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet. It fragments the incoming byte stream into discrete messages and passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.

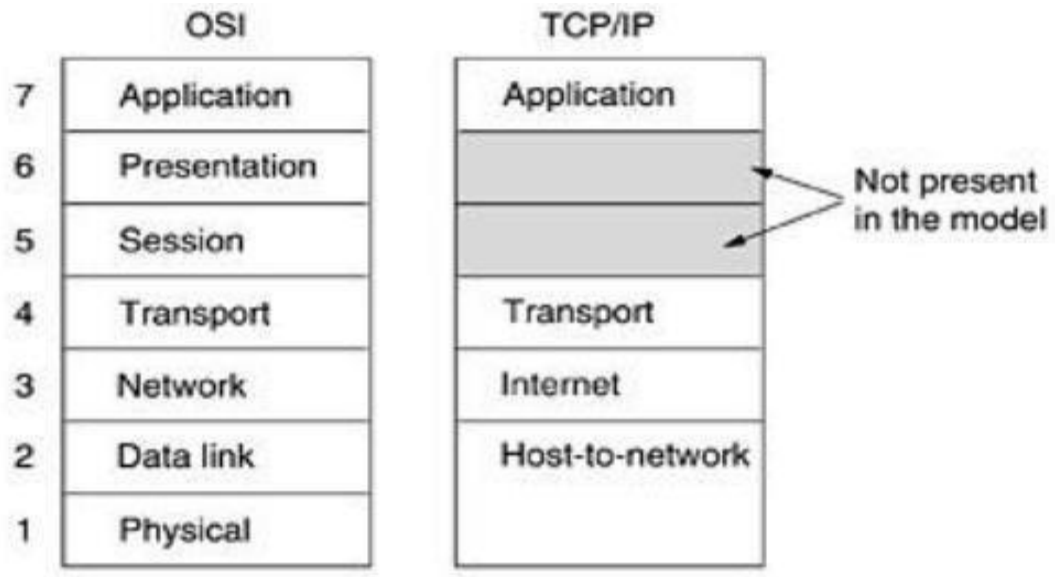


Fig.1: The TCP/IP reference model.

The second protocol in this layer, UDP (User Datagram Protocol), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP, and UDP is shown in Fig.2. Since the model was developed, IP has been implemented on many other networks.

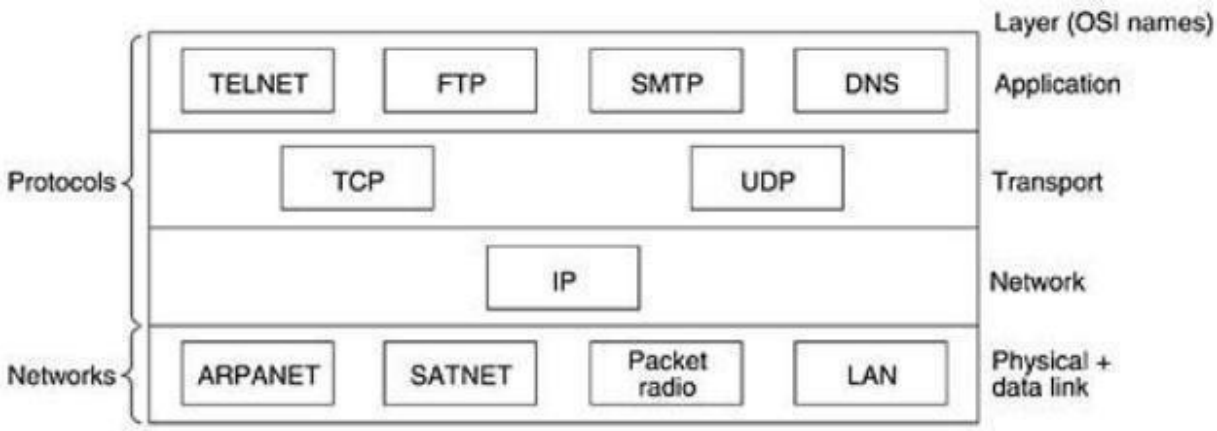


Fig.2: Protocols and networks in the TCP/IP model initially.

The Application Layer:

The TCP/IP model does not have session or presentation layers. On top of the transport layer is the application layer. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP), as shown in Fig.6.2. The

virtual terminal protocol allows a user on one machine to log onto a distant machine and work there. The file transfer protocol provides a way to move data efficiently from one machine to another. Electronic mail was originally just a kind of file transfer, but later a specialized protocol (SMTP) was developed for it. Many other protocols have been added to these over the years: the Domain Name System (DNS) for mapping host names onto their network addresses, NNTP, the protocol for moving USENET news articles around, and HTTP, the protocol for fetching pages on the World Wide Web, and many others.

Comparison of the OSI and TCP/IP Reference Models:

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

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

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

A layer's interface tells the processes above it how to access it. It specifies what the parameters are and what results to expect. It, too, says nothing about how the layer works inside.

Finally, the peer protocols used in a layer are the layer's own business. It can use any protocols it wants to, as long as it gets the job done (i.e., provides the offered services). It can also change them at will without affecting software in higher layers.

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

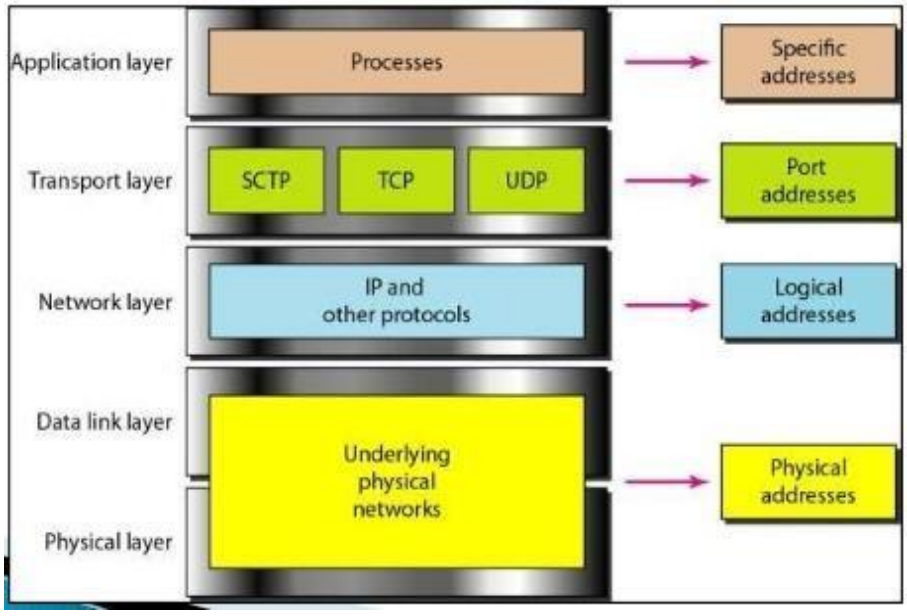
PACKET.

As a consequence, the protocols in the OSI model are better hidden than in the TCP/IP model and can be replaced relatively easily as the technology changes. Being able to make such changes is one of the main purposes of having layered protocols in the first place. The OSI reference model was devised before the corresponding protocols were invented. This ordering means that the model was not biased toward one particular set of protocols, a fact that made it quite general. The downside of this ordering is that the designers did not have much experience with the subject and did not have a good idea of which functionality to put in which layer.

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

1.8. ADDRESSING

The TCP/IP protocols employed in today internet basically uses four levels of addressing. They are: Physical Address, Logical Address, Port Address andSpecific Address and each address is related to specific layer in TCP/IP Arachitecture as shown in below figure.



Physical Address: Physical address is the address of the node defined by its LAN. This physical address is included in the frame used by data link layer. The physical address is also known as link address and this is the lowest level address. The size and format of the physical address depends on the network. For e.g. the physical address used by Ethernet is of 6 byte length which is printed on NIC (Network Interface card). The physical address can be either unicast, multicast or broadcast.

The physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below: A 6-byte (12 hexadecimal digits) physical address **07:01:02:01:2C:4B**

Logical Addresses: The physical addresses discussed in above are not adequate to identify host as internet consists wide number of networks and each network uses different size and format. Thus it is necessary to design a new addressing scheme to identify a host uniquely. This job of identifying a host uniquely is done by logical addresses. The physical address will change from hop to hop where as logical address remains same. The logical address can be either unicast, multicast or broadcast.

Currently the length of logical address is of 32 bit length. The format of an internet address in IPv4 is in decimal numbers **132.24.75.9**

Port Address: Today it is well known fact that the computers are nothing but devices which can run multiple processes at a given time. The port addressing is a way to identify a specific process to which data is to be forwarded when it reaches to the destination. The port address is of 16 bit length. The physical address will change from hop to hop where as port address remains same.

For eg. Let us assume computer A is communicating with computer C using telnet and at the same time computer A is also communicating with computer B using FTP. Now as both of them are different processes the computer A has to give two different port addresses, one for computer B and the other for computer C which will identify the process. A port address is a 16-bit address represented by one decimal number **753**

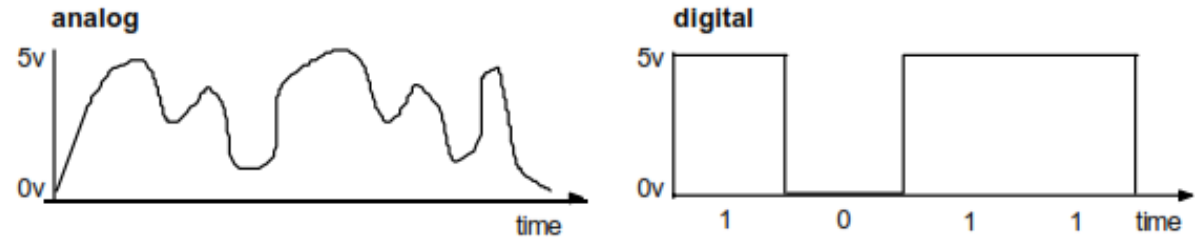
Specific Address: Some applications have user-friendly addresses that are designed for that specific application.

Examples include the e-mail address (for example, co_sci@yahoo.com) and the Universal Resource Locator (URL) (for example, www.mhhe.com). The first defines the recipient of an e-mail; the second is 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.

1.9 ANALOG AND DIGITAL SIGNALS:

Like the data they represent, signals can be either analog or digital. An analog signal has infinitely many levels of intensity over a period of time. As the wave moves from value *A* to value *B*, it passes through and includes an infinite number of values along its path. A digital signal, on the other hand, can have only a limited number of defined values. Although each value can be any number, it is often as simple as 1 and O. The simplest way to show signals is by plotting them on a pair of perpendicular axes. The vertical axis represents the value or strength of a signal. The horizontal axis represents time. Figure below illustrates an analog signal and a digital signal. The curve representing the analog signal passes through an infinite number of points. The vertical lines of the digital signal, however, demonstrate the sudden jump that the signal makes from value to value.

) Analog and digital signals.



Periodic and Nonperiodic Signals:

A periodic signal completes a pattern within a measurable time frame, called a period, and repeats that pattern over subsequent identical periods. The completion of one full pattern is called a cycle. A nonperiodic signal changes without exhibiting a pattern or cycle that repeats over time.

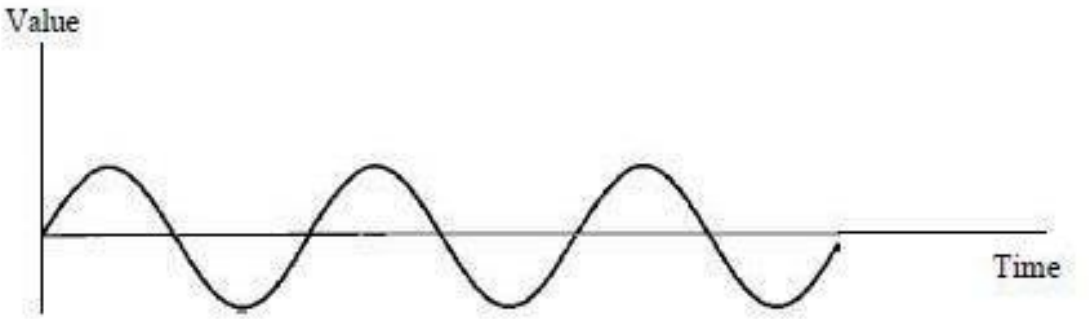
PERIODIC ANALOG SIGNALS:

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

Sine Wave

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

A sine wave

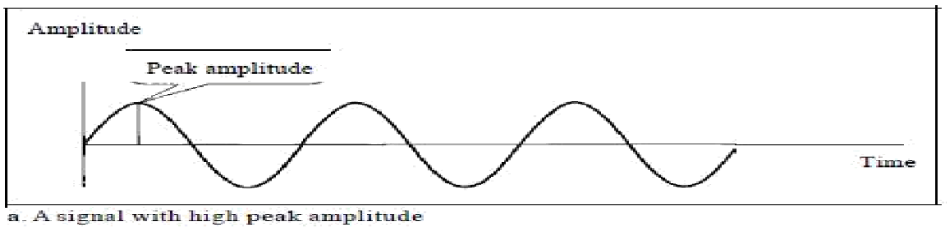


Characteristics of Signals:

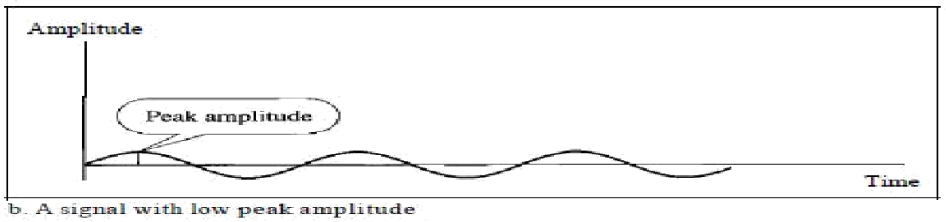
1. Peak Amplitude

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

Two signals with the same phase and frequency, but different amplitudes



a. A signal with high peak amplitude



b. A signal with low peak amplitude

2. eriod and Frequency

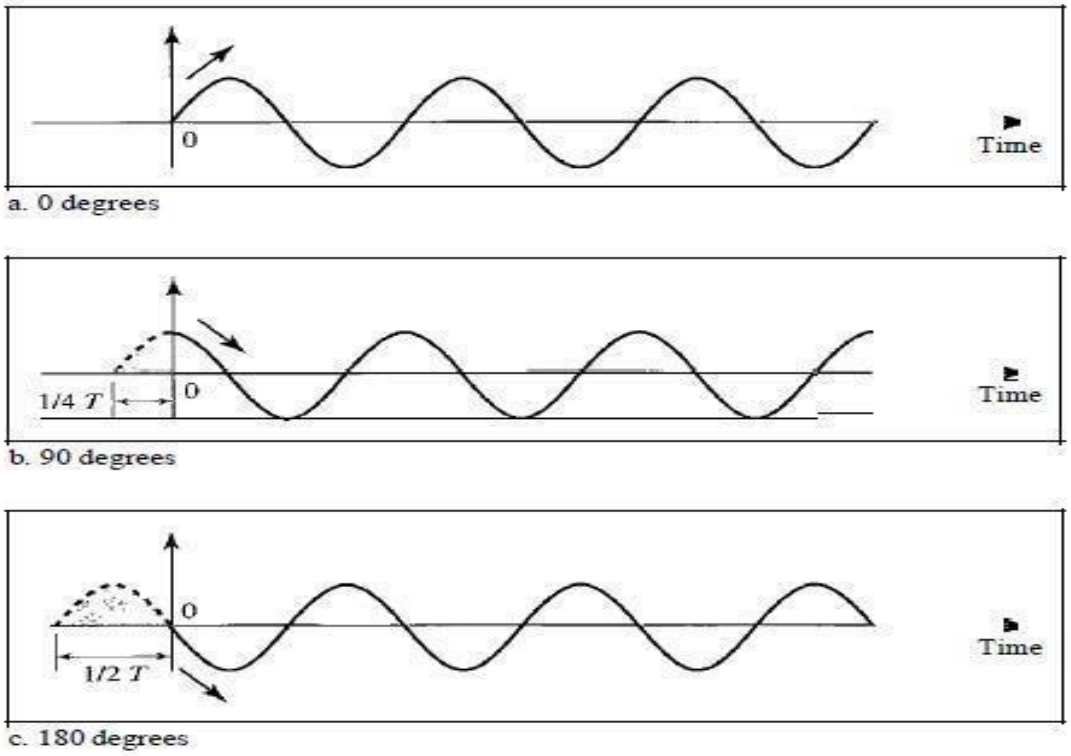
Period refers to the amount of time, in seconds, a signal needs to complete 1 cycle.

Frequency refers to the number of periods in I s. Note that period and frequency are just one characteristic defined in two ways. Period is the inverse of frequency, and frequency is the inverse of period, as the following formulas show.

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

Period is formally expressed in seconds. Frequency is formally expressed in Hertz (Hz), which is cycle per second.

Three sine waves with the same amplitude andfrequency, but different phases



3. ***Phase***

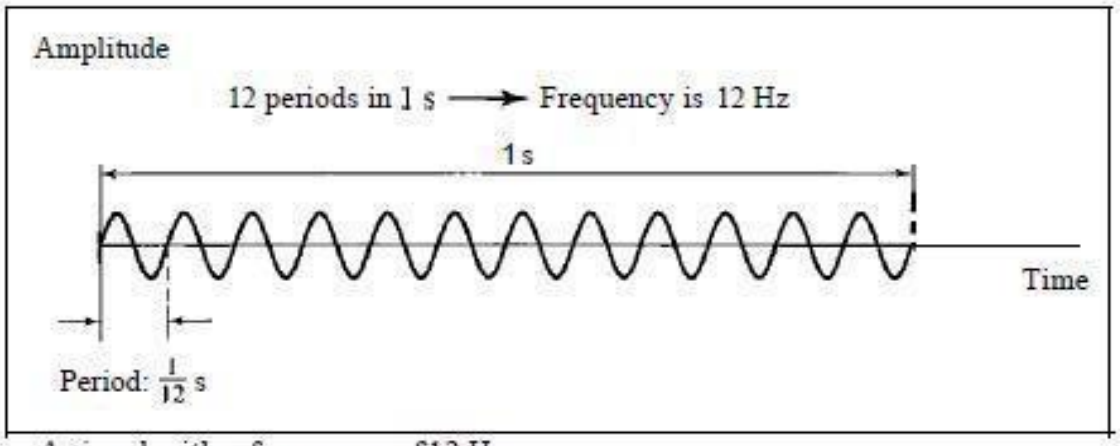
The term phase describes the position of the waveform relative to time O. If we think of the wave as something that can be shifted backward or forward along the time axis, phase describes the amount of that shift. It indicates the status of the first cycle. Phase is measured in degrees or radians [360° is *2n* rad; 1° is *2n/360* rad, and 1 rad is *360/(2n)]*. A phase shift of 360° corresponds to a shift of a complete period; a phase shift of 180° corresponds to a shift of one-half of a period; and a phase shift of 90° corresponds to a shift of one-quarter of a period.

I. A sine wave with a phase of 0° starts at time 0 with a zero amplitude. The amplitude is increasing.

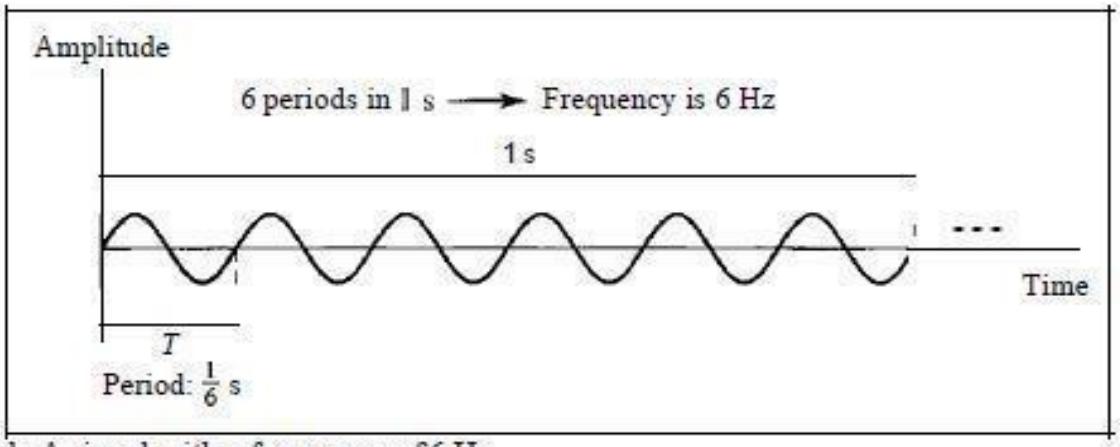
II. A sine wave with a phase of 90° starts at time 0 with a peak amplitude. The amplitude is decreasing.

III. A sine wave with a phase of 180° starts at time 0 with a zero amplitude. The amplitude is decreasing.

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



a. A signal with a frequency of 12 Hz



b. A signal with a frequency of 6 Hz

4. *Wavelength*

Wavelength is another characteristic of a signal traveling through a transmission medium. Wavelength binds the period or the frequency of a simple sine wave to the propagation speed of the medium. While the frequency of a signal is independent of the medium, the wavelength depends on both the frequency and the medium. Wavelength is a property of any type of signal. In data communications, we often use wavelength to describe the transmission of light in an optical fiber. The wavelength is the distance a simple signal can travel in one period. Wavelength can be calculated if one is given the propagation speed (the speed of light) and the period of the signal. However, since period and frequency are related to each other, if we represent wavelength by λ, propagation speed by c (speed of light), and frequency by *f*, we get

Wavelength=Propagation speed * Period = propagation speed/frequency

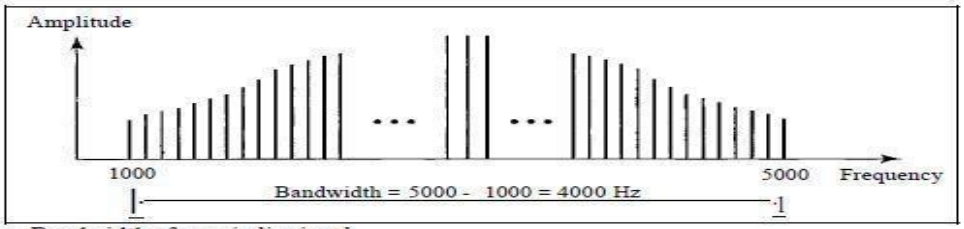
$$\lambda = c/f$$

The wavelength is normally measured in micrometers (microns) instead of meters.

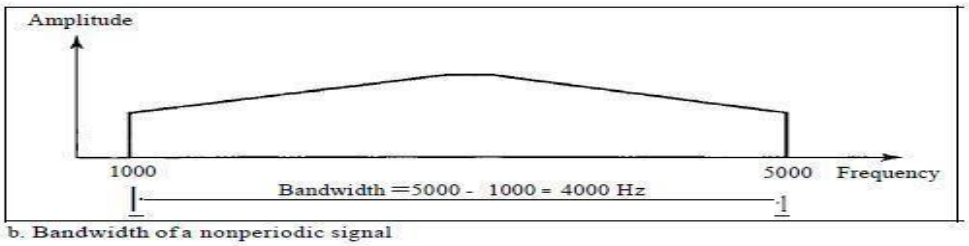
Bandwidth

The range of frequencies contained in a composite signal is its bandwidth. The bandwidth is normally a difference between two numbers. For example, if a composite signal contains frequencies between 1000 and 5000, its bandwidth is 5000 - 1000, or 4000. Figure 3.12 shows the concept of bandwidth. The figure depicts two composite signals, one periodic and the other nonperiodic. The bandwidth of the periodic signal contains all integer frequencies between 1000 and 5000 (1000, 100 I, 1002, ...). The bandwidth of the nonperiodic signals has the same range, but the frequencies are continuous.

Figure 3.12 *The bandwidth ofperiodic and nonperiodic composite signals*



a. Bandwidth of a periodic signal



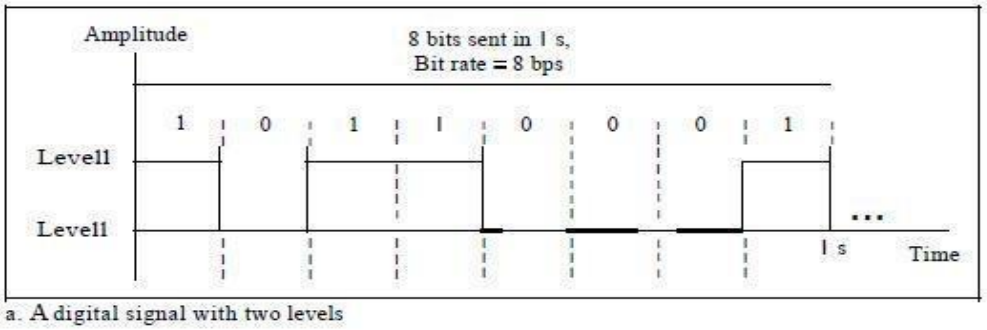
b. Bandwidth of a nonperiodic signal

DIGITAL SIGNALS

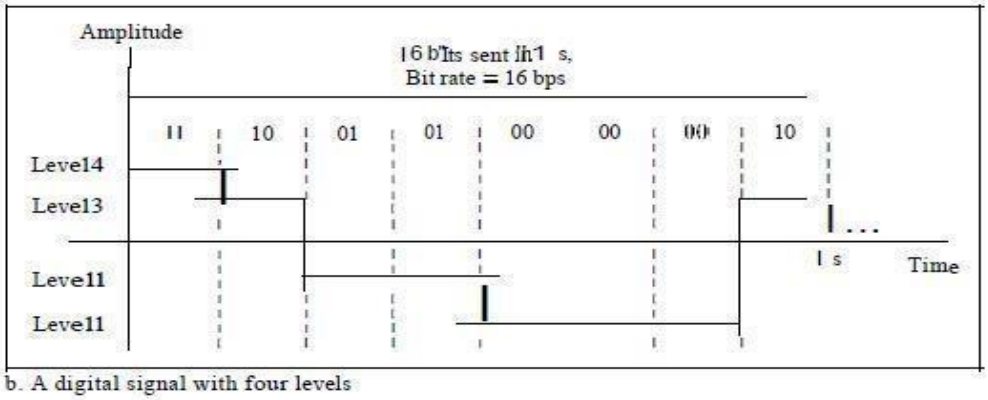
In addition to being represented by an analog signal, information can also be represented by a

digital signal. For example, a 1 can be encoded as a positive voltage and a 0 as zero voltage. A digital signal can have more than two levels. In this case, we can send more than 1 bit for each level. Figure 3.16 shows two signals, one with two levels and the other with four.

Figure 3.16 *Two digital signals: one with two signal levels and the other with four signal levels*



a. A digital signal with two levels



b. A digital signal with four levels

We send 1 bit per level in part a of the figure and 2 bits per level in part b of the figure. In general, if a signal has *L* levels, each level needs *log2L* bits.

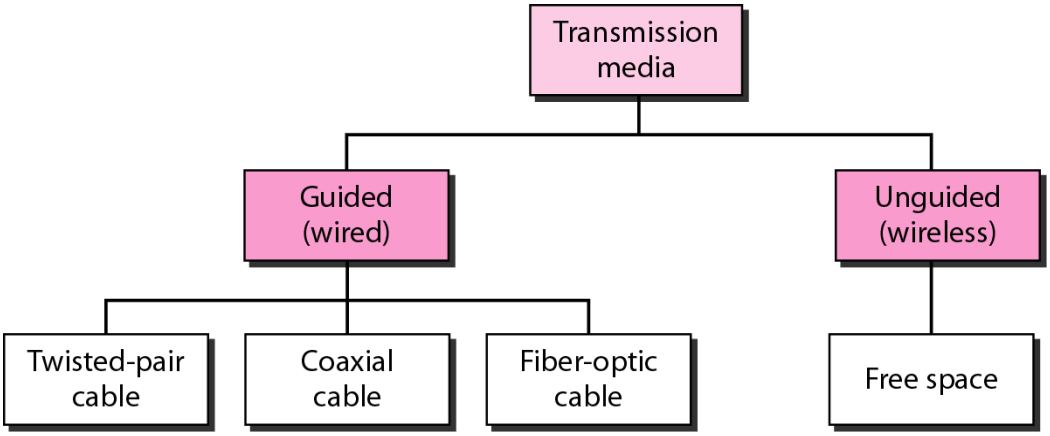
Bit Rate

Most digital signals are nonperiodic, and thus period and frequency are not appropriate characteristics. Another *term-bit rate is* used to describe digital signals. The bit rate is the number of bits sent in 1s, expressed in bits per second (bps). Figure 3.16 shows the bit rate for two signals.

1.10 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 optical cable. The information is usually a signal that is the result of conversion of data from another form.

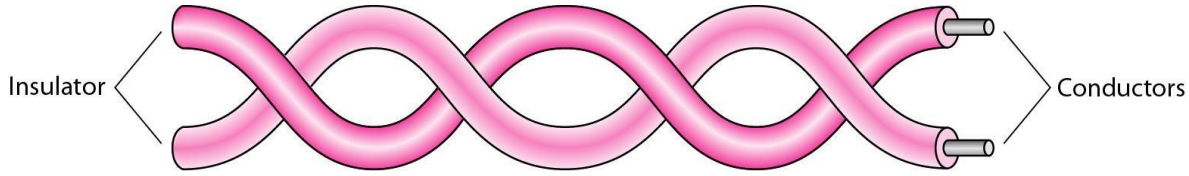


Guided Media

Guided media, which are those that provide a conduit from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable. A signal traveling along any of these media is directed and contained by the physical limits of the medium. Twisted-pair and coaxial cable use metallic (copper) conductors that accept and transport signals in the form of electric current. Optical fiber is a cable that accepts and transports signals in the form of light.

1. Twisted-Pair Cable

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



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

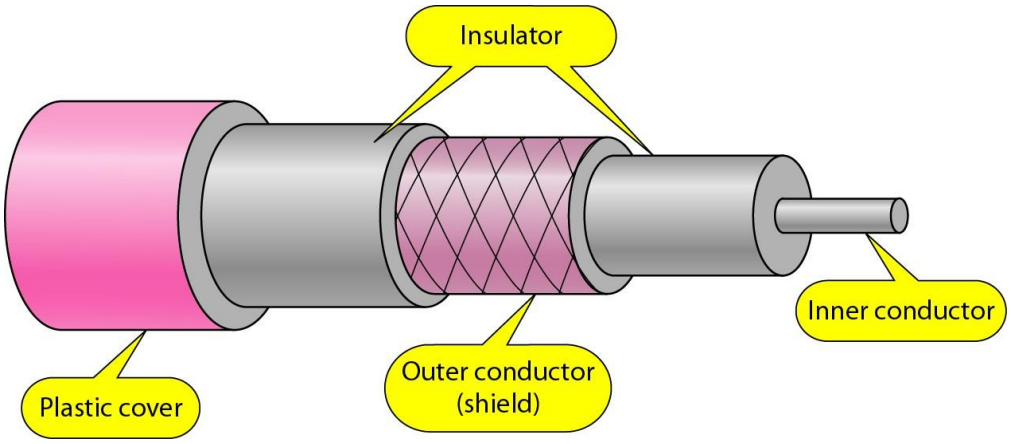
Applications

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 line that are used by the telephone companies to provide high-data-rate connections also use the high-bandwidth capability of unshielded twisted-pair cables. Local-area networks, such as lOBase-T and lOOBase-T, also use twisted-pair cables.

2. <u>Coaxial Cable</u>

Coaxial cable (or *coax)* 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

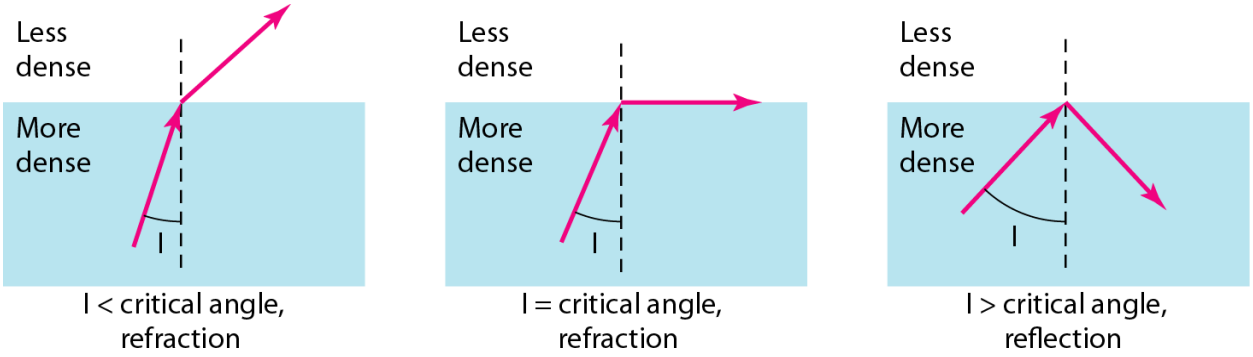
insulating sheath, which is, in turn, encased in an outer conductor of metal foil, braid, or a combination of the two. The outer metallic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit. This outer conductor is also enclosed in an insulating sheath, and the whole cable is protected by a plastic cover (below figure).



Applications

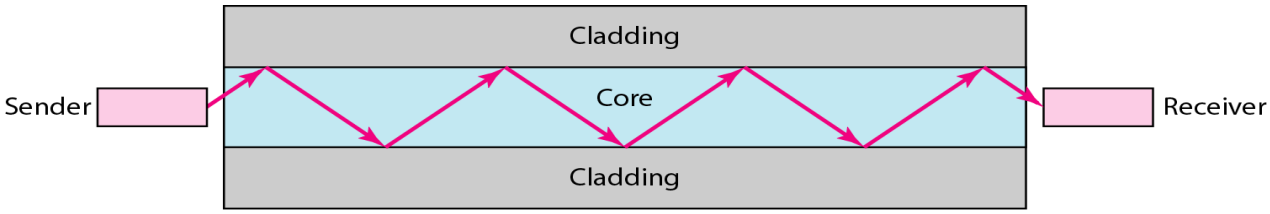
Coaxial cable was widely used in analog telephone networks where a single coaxial network could carry 10,000 voice signals. Later it was used in digital telephone networks where a single coaxial cable could carry digital data up to 600 Mbps. However, coaxial cable in telephone networks has largely been replaced today with fiber-optic cable. Cable TV networks also use coaxial cables. In the traditional cable TV network, the entire network used coaxial cable. Later, however, cable TV providers replaced most of the media with fiber-optic cable; hybrid networks use coaxial cable only at the network boundaries, near the consumer premises. Cable TV uses RG-59 coaxial cable. Another common application of coaxial cable is in traditional Ethernet LANs. Because of its high bandwidth, and consequently high data rate, coaxial cable was chosen for digital transmission in early Ethernet LANs.

3. **Fiber Optic Cable:** A fiber-optic cable is made of glass or plastic and transmits signals in the form of light. To understand optical fiber, we first need to explore several aspects of the nature of light. Light travels in a straight line as long as it is moving through a single uniform If a ray of light traveling through one substance suddenly enters another substance (of a different density), the ray changes direction. Figure 7.10 shows how a ray of light changes direction when going from a more dense to a less dense substance.



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

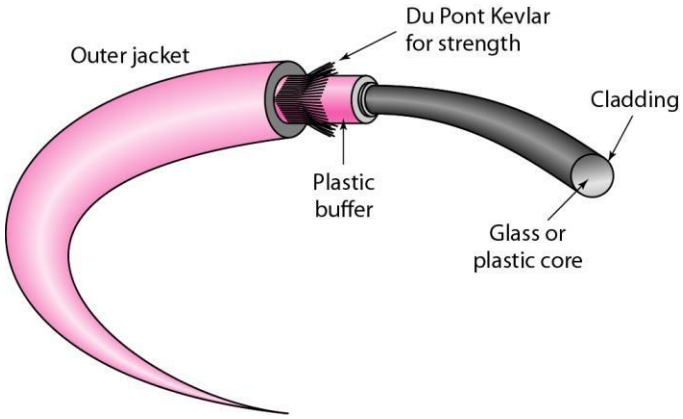
Optical fibers use reflection to guide light through a channel. A glass or plastic core is surrounded by a cladding of less dense glass or plastic. The difference in density of the two materials must be such that a beam of light moving through the core is reflected off the cladding instead of being refracted into it. See Figure below.



Cable Composition

Figure 7.14 shows the composition of a typical fiber-optic cable. The outer jacket is made of either PVC or Teflon. Inside the jacket are Kevlar strands to strengthen the cable. Kevlar is a strong material used in the fabrication of bulletproof vests. Below the Kevlar is another plastic

coating to cushion the fiber. The fiber is at the center of the cable, and it consists of cladding and core.



Applications

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. The SONET network provides such a backbone. Some cable TV companies use a combination of optical fiber and coaxial cable, thus creating a hybrid network. Optical fiber provides the backbone structure while coaxial cable provides the connection to the user premises. This is a cost-effective configuration since the narrow bandwidth requirement at the user end does not justify the use of optical fiber. Local-area networks such as 100Base-FX network (Fast Ethernet) and 1000Base-X also use fiber-optic cable.

Advantages and Disadvantages of Optical Fiber

Advantages

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

1. **Higher bandwidth**. Fiber-optic cable can support dramatically higher bandwidths (and hence data rates) than either twisted-pair or coaxial cable. Currently, data rates and bandwidth utilization over fiber-optic cable are limited not by the medium but by the signal generation and reception technology available.

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

3. **Immunity to electromagnetic interference**. Electromagnetic noise cannot affect fiber-optic cables.

4. **Resistance to corrosive materials**. Glass is more resistant to corrosive materials than copper.

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

6. **Greater immunity to tapping**. Fiber-optic cables are more immune to tapping than copper cables. Copper cables create antenna effects that can easily be tapped.

Disadvantages

There are some disadvantages in the use of optical fiber.

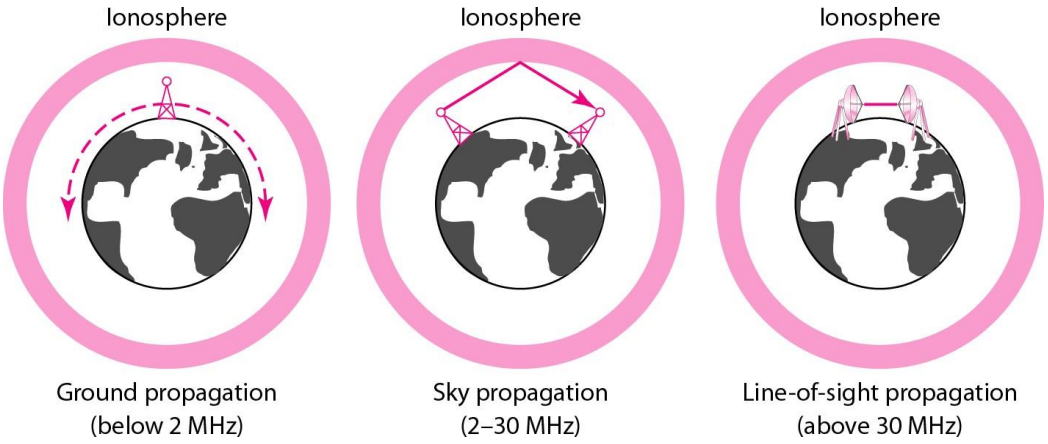
1. **Installation and maintenance**. Fiber-optic cable is a relatively new technology. Its installation and maintenance require expertise that is not yet available everywhere.

2. **Unidirectional light propagation**. Propagation of light is unidirectional. If we need bidirectional communication, two fibers are needed.

3. **Cost**. The cable and the interfaces are relatively more expensive than those of other guided media. If the demand for bandwidth is not high, often the use of optical fiber cannot be justified.

UNGUIDED MEDIA: WIRELESS

Unguided media transport electromagnetic waves without using a physical conductor. This type of communication is often referred to as wireless communication. Signals are normally broadcast through free space and thus are available to anyone who has a device capable of receiving them.



Unguided signals can travel from the source to destination in several ways: ground propagation, sky propagation, and line-of-sight propagation, as shown in Figure 7.18. In ground propagation, radio waves travel through the lowest portion of the atmosphere, hugging the earth. These low-frequency signals emanate in all directions from the transmitting antenna and follow the curvature of the planet. Distance depends on the amount of power in the signal: The greater the power, the greater the distance. In sky propagation, higher-frequency radio waves radiate upward into the ionosphere where they are reflected back to earth. This type of transmission allows for greater distances with lower output power. In line of sight propagation, very high frequency signals are transmitted in straight lines directly from antenna to antenna. Antennas must be directional, facing each other, and either tall enough or close enough together not to be affected by the curvature of the earth. Line-of-sight propagation is tricky because radio transmissions cannot be completely focused.

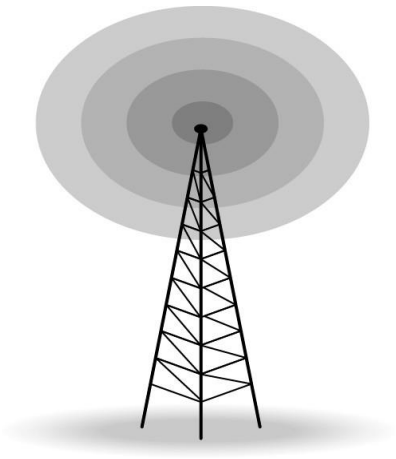
1. <u>Radio Waves</u>

Waves ranging in frequencies between 3 kHz and 1 GHz are called radio waves. Radio waves, for the most part, are omnidirectional. When an antenna transmits radio waves, they are propagated in all directions. This means that the sending and receiving antennas do not have to be aligned. A sending antenna sends waves that can be received by any receiving antenna. The omnidirectional property has a disadvantage, too. The radio waves transmitted by one antenna

are susceptible to interference by another antenna that may send signals using the same frequency or band. Radio waves, particularly those waves that propagate in the sky mode, can travel long distances. This makes radio waves a good candidate for long-distance broadcasting such as AM radio. Radio waves, particularly those of low and medium frequencies, can penetrate walls. This characteristic can be both an advantage and a disadvantage. It is an advantage because, for example, an AM radio can receive signals inside a building. It is a disadvantage because we cannot isolate a communication to just inside or outside a building. The radio wave band is relatively narrow, just under 1 GHz, compared to the microwave band. When this band is divided into sub bands, the sub bands are also narrow, leading to a low data rate for digital communications.

Omnidirectional Antenna

Radio waves use omnidirectional antennas that send out signals in all directions. Based on the wavelength, strength, and the purpose of transmission, we can have several types of antennas. Below figure 7.20 shows an omnidirectional antenna.



Applications

The omnidirectional characteristics of radio waves make them useful for multicasting, in which there is one sender but many receivers. AM and FM radio, television, maritime radio, cordless

phones, and paging are examples of multicasting.

2. Microwaves

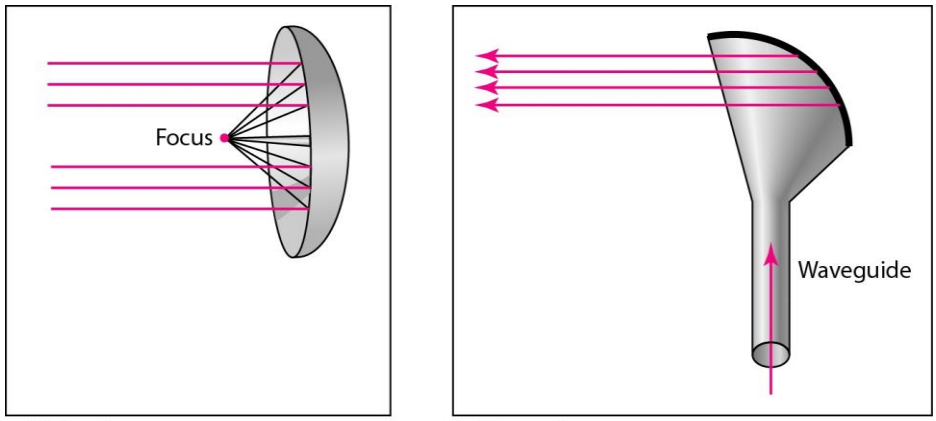
Electromagnetic waves having frequencies between I and 300 GHz are called microwaves. Microwaves are unidirectional. When an antenna transmits microwave waves, they can be narrowly focused. This means that the sending and receiving antennas need to be aligned. The unidirectional property has an obvious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

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

Unidirectional Antenna

Microwaves need unidirectional antennas that send out signals in one direction. Two types of antennas are used for microwave communications: the parabolic dish and the horn (see below figure). A parabolic dish antenna is based on the geometry of a parabola: Every line parallel to the line of symmetry (line of sight) reflects off the curve at angles such that all the lines intersect in a common point called the focus. The parabolic dish works as a funnel, catching a wide range of waves and directing them to a common point. In this way, more of the signal is recovered than would be possible with a single-point receiver. Outgoing transmissions are broadcast through a

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



a. Dish antenna b. Horn antenna

3. Infrared

Infrared waves, with frequencies from 300 GHz to 400 THz (wavelengths from 1 mm to 770 nm), can be used for short-range communication. Infrared waves, having high frequencies, cannot penetrate walls. This advantageous characteristic prevents interference between one system and another; a short-range communication system in one room cannot be affected by another system in the next room. When we use our infrared remote control, we do not interfere with the use of the remote by our neighbors. However, this same characteristic makes infrared signals useless for long-range communication. In addition, we cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication.

Applications

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

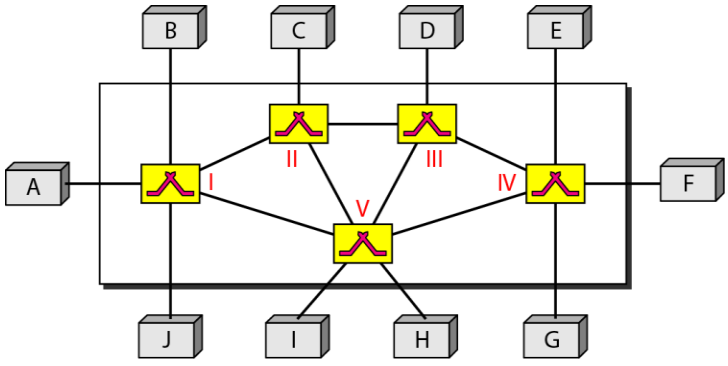
that allows a wireless keyboard to communicate with a PC. The standard originally defined a data rate of 75 kbps for a distance up to 8 m. The recent standard defines a data rate of 4 Mbps.

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

SWITCHING

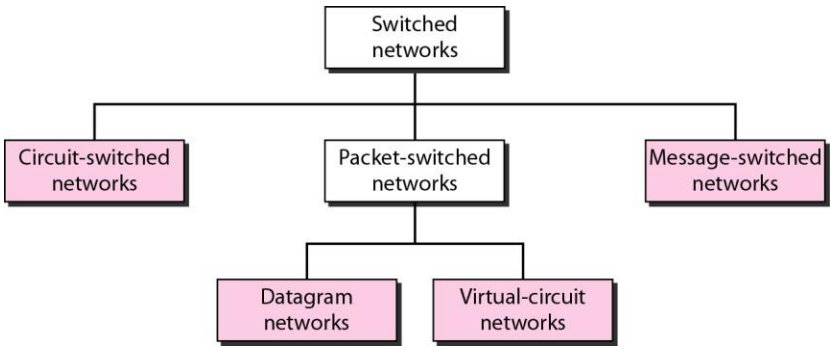
A network is a set of connected devices. Whenever we have multiple devices, we have the problem of how to connect them to make one-to-one communication possible. One solution is to make a point-to-point connection between each pair of devices (a mesh topology) or between a central device and every other device (a star topology). These methods, however, are impractical and wasteful when applied to very large networks. The number and length of the links require too much infrastructure to be cost-efficient, and the majority of those links would be idle most of the time. Other topologies employing multipoint connections, such as a bus, are ruled out because the distances between devices and the total number of devices increase beyond the capacities of the media and equipment.

A better solution is **switching**. A switched network consists of a series of interlinked nodes, called switches. Switches are devices capable of creating temporary connections between two or more devices linked to the switch. In a switched network, some of these nodes are connected to the end systems (computers or telephones, for example). Others are used only for routing. Figure 8.1 shows a switched network.



The end systems (communicating devices) are labeled A, B, C, D, and so on, and the switches are labeled I, II, III, IV, and V. Each switch is connected to multiple links. Traditionally, three

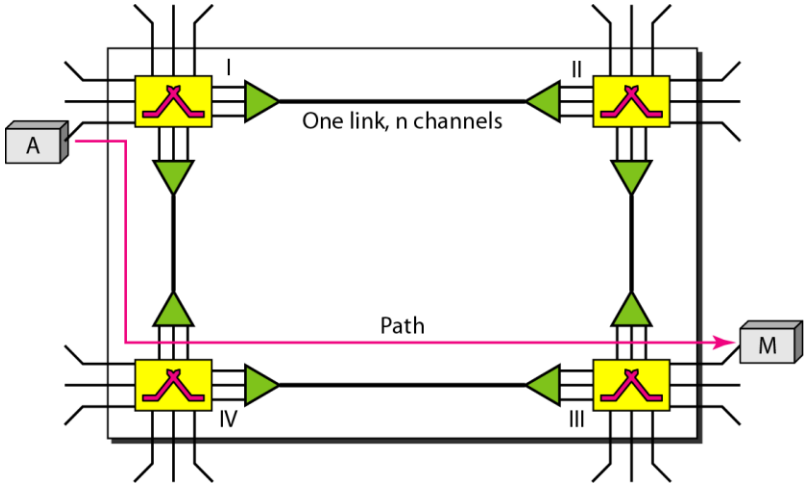
methods of switching have been important: circuit switching, packet switching, and message switching. The first two are commonly used today. The third has been phased out in general communications but still has networking applications. We can then divide today's networks into three broad categories: circuit-switched networks, packet-switched networks, and message-switched.



CIRCUIT-SWITCHED NETWORKS

A circuit-switched network consists of a set of switches connected by physical links. A connection between two stations is a dedicated path made of one or more links. However, each connection uses only one dedicated channel on each link. Each link is normally divided into *n* channels by using FDM or TDM. Figure 8.3 shows a trivial circuit-switched network with four switches and four links. Each link is divided into *n (n* is 3 in the figure) channels by using FDM or TDM.

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



Three Phases

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

Setup Phase

Before the two parties (or multiple parties in a conference call) can communicate, a dedicated circuit (combination of channels in links) needs to be established. The end systems are normally connected through dedicated lines to the switches, so connection setup means creating dedicated channels between the switches. For example, in Figure 8.3, when system A needs to connect to system M, it sends a setup request that includes the address of system M, to switch I. Switch I finds a channel between itself and switch IV that can be dedicated for this purpose. Switch I then sends the request to switch IV, which finds a dedicated channel between itself and switch III. Switch III informs system M of system A's intention at this time. In the next step to making a connection, an acknowledgment from system M needs to be sent in the opposite direction to system A. Only after system A receives this acknowledgment is the connection established.

Data Transfer Phase

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

Teardown Phase

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

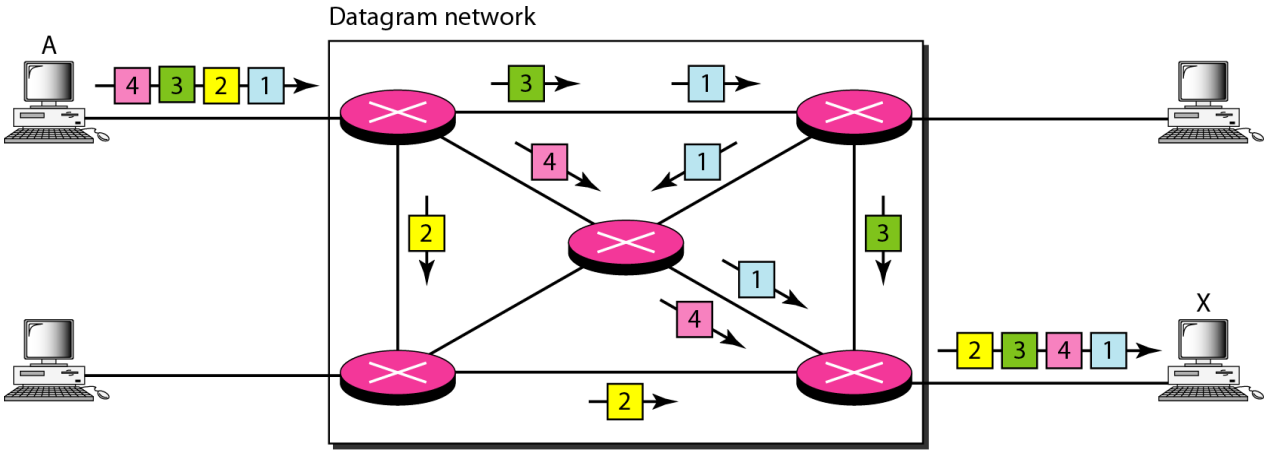
1. PACKET SWITCHED NETWORK

In a Computer Network, the communication between two ends is done in blocks of data called

packets. So instead of continuous communication the exchange takes place in the form of individual packets between the two computers. This allows us to make the switches function for both storing and forwarding because a packet is an independent entity that can be stored and sent later.

a. DATAGRAM NETWORKS

In data communications, we need to send messages from one end system to another. If the message is going to pass through a packet-switched network, it needs to be divided into packets of fixed or variable size. The size of the packet is determined by the network and the governing protocol. In packet switching, there is no resource allocation for a packet. This means that there is no reserved bandwidth on the links, and there is no scheduled processing time for each packet. Resources are allocated on demand. The allocation is done on a first come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed. As with other systems in our daily life, this lack of reservation may create delay. For example, if we do not have a reservation at a restaurant, we might have to wait. In a datagram network, each packet is treated independently of all others. Even if a packet is part of a multipacket transmission, the network treats it as though it existed alone. Packets in this approach are referred to as datagrams. Datagram switching is normally done at the network layer. The switches in a datagram network are traditionally referred to as routers.



In this example, all four packets (or datagrams) belong to the same message, but may travel different paths to reach their destination. This is so because the links may be involved in carrying packets from other sources and do not have the necessary bandwidth available to carry all the packets from A to X. This approach can cause the datagrams of a transmission to arrive at their

destination out of order with different delays between them packets. Packets may also be lost or dropped because of a lack of resources. In most protocols, it is the responsibility of an upper-layer protocol to reorder the datagrams or ask for lost datagrams before passing them on to the application. The datagram networks are sometimes referred to as connectionless networks. The term *connectionless* here means that the switch (packet switch) does not keep information about the connection state. There are no setup or teardown phases. Each packet is treated the same by a switch regardless of its source or destination.

b. VIRTUAL-CIRCUIT NETWORKS

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

4. As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.

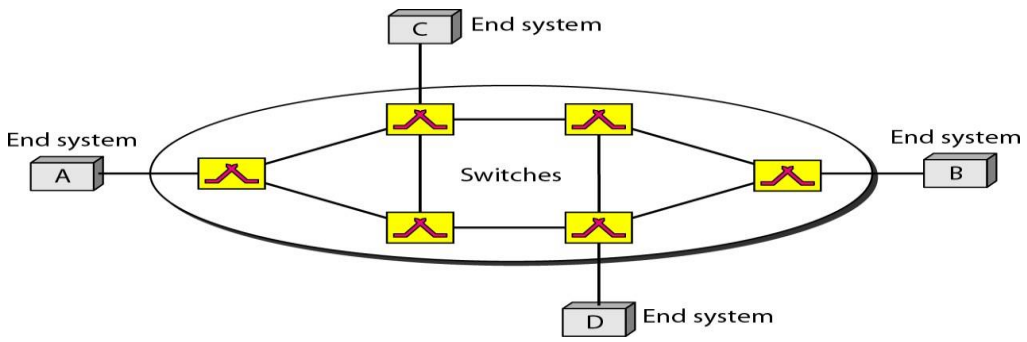
5. Resources can be allocated during the setup phase, as in a circuit-switched network, or on demand, as in a datagram network.

6. As in a datagram network, data are packetized and each packet carries an address in the header. However, the address in the header has local jurisdiction , not end-to-end jurisdiction. The reader may ask how the intermediate switches know where to send the packet if there is no final destination address carried by a packet.

7. As in a circuit-switched network, all packets follow the same path established during the connection.

8. A virtual-circuit network is normally implemented in the data link layer, while a circuit-switched network is implemented in the physical layer and a datagram network in the network layer. But this may change in the future.

Figure 8.10 is an example of a virtual-circuit network. The network has switches that allow traffic from sources to destinations. A source or destination can be a computer, packet switch, bridge, or any other device that connects other networks.



UNIT-II
DATA LINK LAYER

2.1 Introduction

The data link layer transforms the physical layer, a raw transmission facility, to a link responsible for node-to-node (hop-to-hop) communication. Specific responsibilities of the data link layer include *framing, addressing, flow control, error control, and media access control.*

2.1.1. DATA LINK LAYER DESIGN ISSUES

The following are the data link layer design issues

1. Services Provided to the Network Layer

The network layer wants to be able to send packets to its neighbors without worrying about the details of getting it there in one piece.

2. Framing

Group the physical layer bit stream into units called frames. Frames are nothing more than "packets" or "messages". By convention, we use the term "frames" when discussing DLL.

3. Error Control

Sender checksums the frame and transmits checksum together with data. Receiver re-computes the checksum and compares it with the received value.

4. Flow Control

Prevent a fast sender from overwhelming a slower receiver.

2.1.2 Services Provided to the Network Layer

The function of the data link layer is to provide services to the network layer. The principal service is transferring data from the network layer on the source machine to the network layer on the destination machine.

The data link layer can be designed to offer various services. The actual services offered can vary from system to system. Three reasonable possibilities that are commonly provided are

1) **Unacknowledged Connectionless service**
2) **Acknowledged Connectionless service**
3) **Acknowledged Connection-Oriented service**

In **Unacknowledged connectionless service** consists of having the source machine send independent frames to the destination machine without having the destination machine acknowledge them.

No logical connection is established beforehand or released afterward. If a frame is lost due to noise on the line, no attempt is made to detect the loss or recover from it in the data link layer.

This class of service is appropriate when the error rate is very low so that recovery is left to higher layers. It is also appropriate for real-time traffic, such as voice, in which late data are worse than bad data. Most LANs use unacknowledged connectionless service in the data link layer.

When **Acknowledged Connectionless service** is offered, there are still no logical connections used, but each frame sent is individually acknowledged.

In this way, the sender knows whether a frame has arrived correctly. If it has not arrived within a specified time interval, it can be sent again. This service is useful over unreliable channels, such as wireless systems.

Adding Ack in the DLL rather than in the Network Layer is just an optimization and not a requirement. If individual frames are acknowledged and retransmitted, entire packets get through much faster. On reliable channels, such as fiber, the overhead of a heavyweight data link

protocol may be unnecessary, but on wireless channels, with their inherent unreliability, it is well worth the cost.

In Acknowledged Connection-Oriented service, the source and destination machines establish a connection before any data are transferred. Each frame sent over the connection is numbered, and the data link layer guarantees that each frame sent is indeed received. Furthermore, it guarantees that each frame is received exactly once and that all frames are received in the right order.

When connection-oriented service is used, transfers go through three distinct phases.

In the first phase, the connection is established by having both sides initialize variables and counters needed to keep track of which frames have been received and which ones have not.

In the second phase, one or more frames are actually transmitted.

In the third and final phase, the connection is released, freeing up the variables, buffers, and other resources used to maintain the connection

2.2 CYCLIC CODES

The cyclic codes are special class of linear block codes which has property of generating a new code word when the given codeword is shifted cyclically. For e.g., if we assume the bits of first word as a_0 to a_6 and bits in the second word can be obtained by shifting as shown below.

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

2.2.1 CYCLIC REDUNCANCY CHECK (CRC)

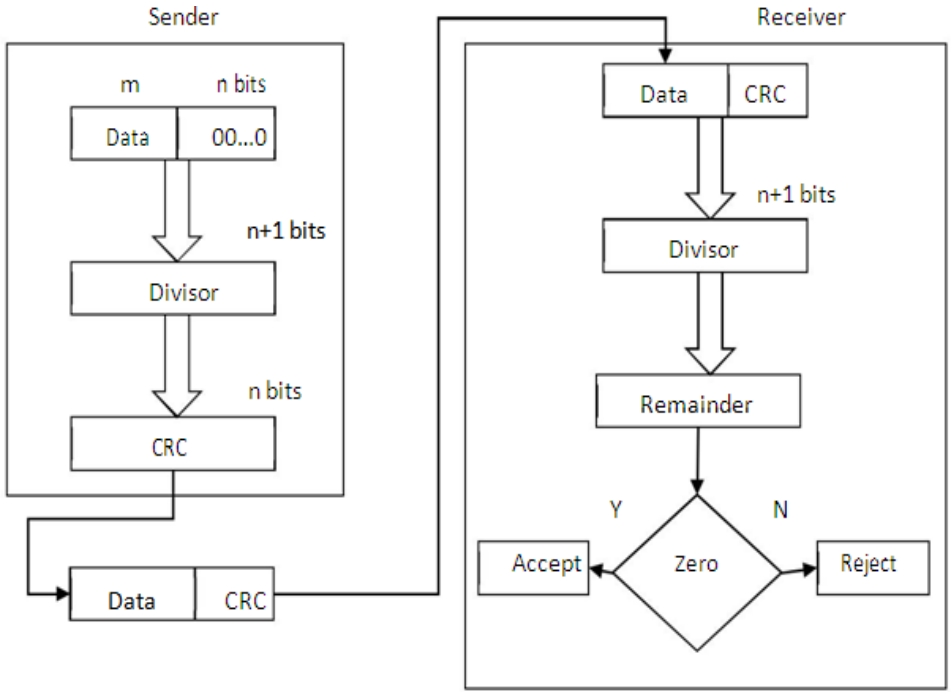
The cyclic redundancy check codes are popularly employed in LANs and WANs for error correction. The principle of operation of CRC encoders and decoders can be better explained with the following examples.

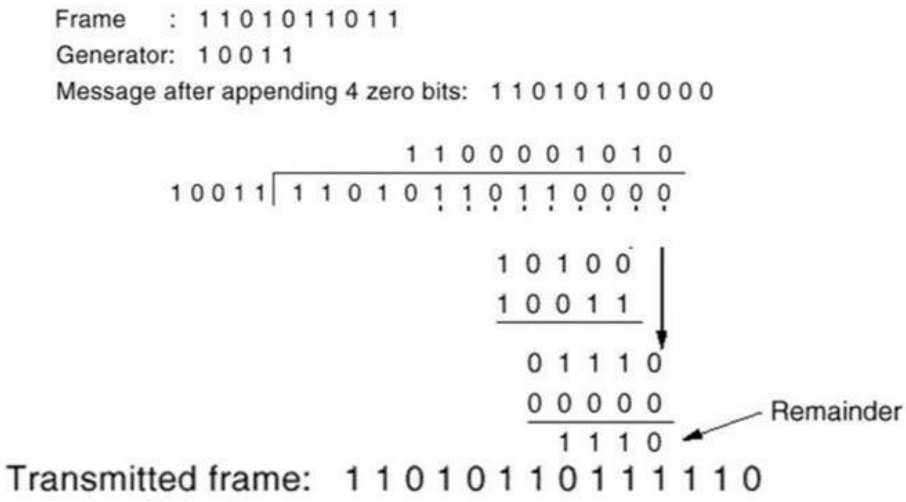
CRC is the most powerful and easy to implement technique.CRC is based on *binary division*. In CRC, a sequence of redundant bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.

At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted. A remainder indicates that the data unit has been damaged in transit and therefore must be rejected. The binary number, which is (r+1) bit in length, can also be considered as the coefficients of a polynomial, called *Generator Polynomial*.

PERFORMANCE OF CRC

CRC is a very effective error detection technique. If the divisor is chosen according to the previously mentioned rules, its performance can be summarized as follows. CRC can detect all single-bit errors and double bit errors (three 1's). CRC can detect any odd number of errors (X+1) and it can also detect all burst errors of less than the degree of the polynomial.





2.3 CHECKSUM

Checksum is also very popular error detection method. Checksum is not used in Data Link Layer, but used by another protocols. Checksum is also based upon the concept of Redundancy.

Ex: Consider a list of 4 numbers to be transferred from sender to destination. What we do is , we send the list of 4 No's along with the sum of these numbers.

1) When the sender sends this list, the receiver adds the 4 no's and compares the result with the last no's. if it is matched, data is transferred properly, else not transferred and assumes error is occurred.
2) We can make the job of the receiver easier if we send the negative (complement) of the sum, called "Checksum". In this case we send (7, 11, 12, 6, -36). The Receiver can add no's received (including checksum)
    - If the result=0, it assumes no error.
    - If the result=1, an error is occurred.

Sender site		Receiver site
7		7
11		11
12		12
0	7, 11, 12, 0, 6, 9	0
6	Packet	6
0		9
Sum = 36		Sum = 45
Wrapped sum = 6		Wrapped Sum = 15
Checksum = 9		Checksum = 0

Detail of Wrapping and Complementing

1 0 0 1 0 0	36		1 0 1 1 0 1	45
1 0			1 0	
0 1 1 0	6		1 1 1 1	15
1 0 0 1	**9**	Complementing	**0 0 0 0**	**0**

Internet Checksum:

The Internet has been using a 16-bit checksum.

The sender calculates the checksum by following these steps:

Sender site:

1. The message is divided into 16-bit words.
2. The value of the checksum word is set to 0.

3. All words including the checksum are added using one's complement addition.
4. The sum is complemented and becomes the checksum.
5. The checksum is sent with the data.

The receiver uses the following steps for error detection.

Receiver site:

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

Example:

Let us calculate the checksum for a text of 8 characters ("Forouzan"). The text needs to be divided into 2 byes (16-bit) word.

1 0 1 3 Carries	1 0 1 3 Carries
4 6 6 F (Fo)	4 6 6 F (Fo)
7 2 6 F (ro)	7 2 6 F (ro)
7 5 7 A (uz)	7 5 7 A (uz)
6 1 6 E (an)	6 1 6 E (an)
0 0 0 0 Checksum(initial)	7 0 3 8 Checksum(received)
8 F C 6 Sum(Partial)	F F F E Sum(Partial)
1	1
8 F C 7 Sum	F F F F Sum
7 0 3 8 Checksum(to send)	0 0 0 0 Checksum(new)
a. Checksum at the sender site	b. Checksum at the receiver site

Performance:

The performance of checksum is not strong as the CRC in error-checking capability. The tendency in the internet, particularly in designing new protocols, is to replace checksum with a

CRC.

2.4 FRAMING

To provide service to the network layer, the data link layer must use the service provided to it by the physical layer. What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination. This bit stream is not guaranteed to be error free. The number of bits received may be less than, equal to, or more than the number of bits transmitted, and they may have different values. It is up to the data link layer to detect and, if necessary, correct errors. The usual approach is for the data link layer to break the bit stream up into discrete frames and compute the checksum for each frame. When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame and possibly also sending back an error report).

Breaking the bit stream up into frames is more difficult than it at first appears. One way to achieve this framing is to insert time gaps between frames, much like the spaces between words in ordinary text. However, networks rarely make any guarantees about timing, so it is possible these gaps might be squeezed out or other gaps might be inserted during transmission. Since it is too risky to count on timing to mark the start and end of each frame, other methods have been devised. We will look at four methods:

1. Character count.
2. Flag bytes with byte stuffing.
3. Starting and ending flags, with bit stuffing.
4. Physical layer coding violations.

The first framing method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is. This technique is shown in below Fig. (a) for four frames of sizes 5, 5, 8, and 8 characters, respectively.

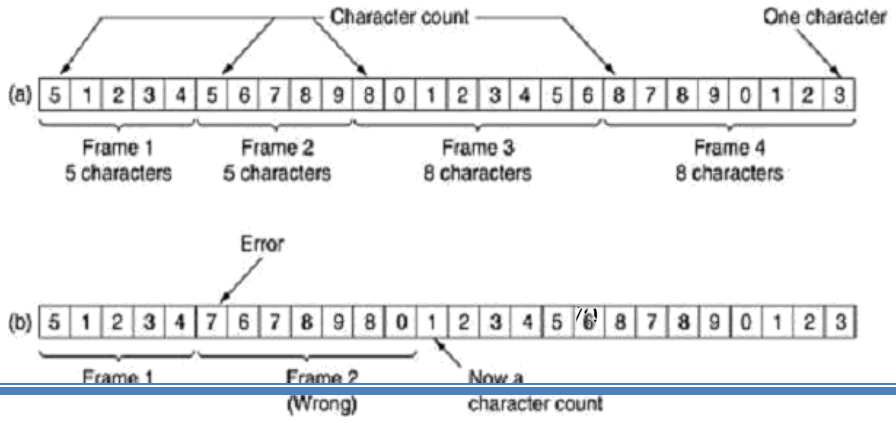


Fig. A character stream. (a) Without errors. (b) With one error.

The trouble with this algorithm is that the count can be garbled by a transmission error. For example, if the character count of 5 in the second frame of Fig. 3.1(b) becomes a 7, the destination will get out of synchronization and will be unable to locate the start of the next frame. Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts. Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission. For this reason, the character count method is rarely used anymore.

The second framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes. In the past, the starting and ending bytes were different, but in recent years most protocols have used the same byte, called a flag byte, as both the starting and ending delimiter, as shown in below Fig. (a) as FLAG. In this way, if the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame. Two consecutive flag bytes indicate the end of one frame and start of the next one.

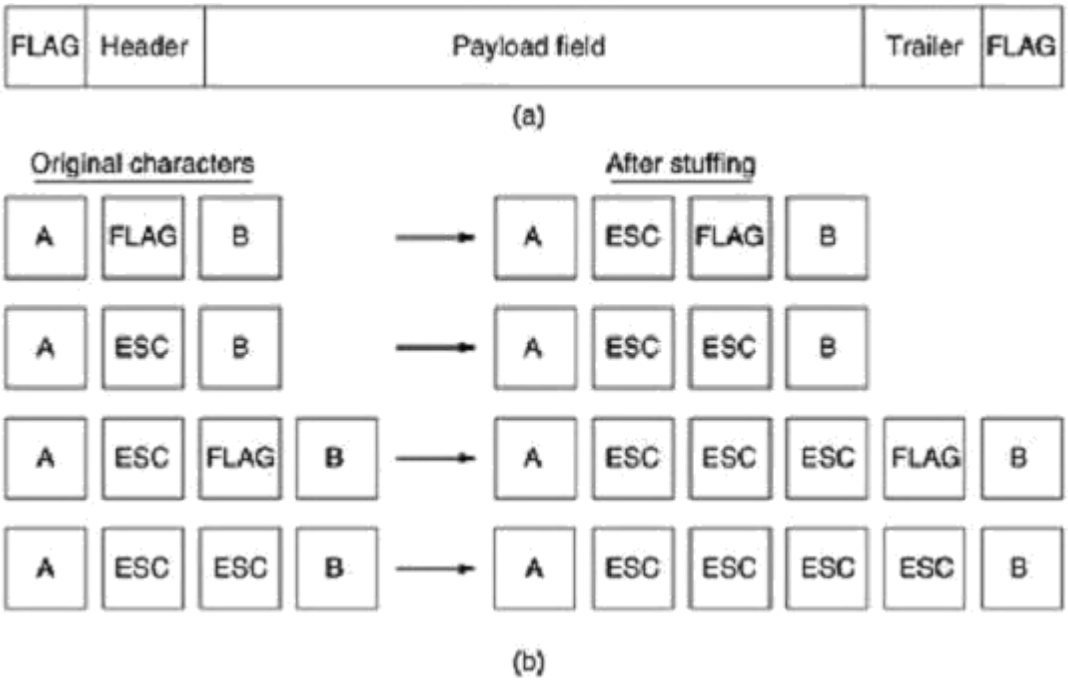


Fig. (a) A frame delimited by flag bytes (b) Four examples of byte sequences before and after byte stuffing.

A serious problem occurs with this method when binary data, such as object programs or floating-point numbers, are being transmitted. It may easily happen that the flag byte's bit pattern occurs in the data. This situation will usually interfere with the framing. One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each "accidental" flag byte in the data. The data link layer on the receiving end removes the escape byte before the data are given to the network layer. This technique is called byte stuffing or character stuffing. Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.

Of course, the next question is: What happens if an escape byte occurs in the middle of the data? The answer is that it, too, is stuffed with an escape byte. Thus, any single escape byte is part of an escape sequence, whereas a doubled one indicates that a single escape occurred naturally in the data. Some examples are shown in Fig.(b). In all cases, the byte sequence delivered after de stuffing is exactly the same as the original byte sequence.

The byte-stuffing scheme depicted in below Fig. is a slight simplification of the one used in the PPP protocol that most home computers use to communicate with their Internet service provider. A major disadvantage of using this framing method is that it is closely tied to the use of 8-bit characters. Not all character codes use 8-bit characters. For example UNICODE uses 16-bit characters, As networks developed, the disadvantages of embedding the character code length in the framing mechanism became more and more obvious, so a new technique had to be developed to allow arbitrary sized characters.

The new technique allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character. It works like this. Each frame begins and ends with a special bit pattern, 01111110 (in fact, a flag byte). Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically de

stuffs (i.e., deletes) the 0 bit. Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing. If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110.

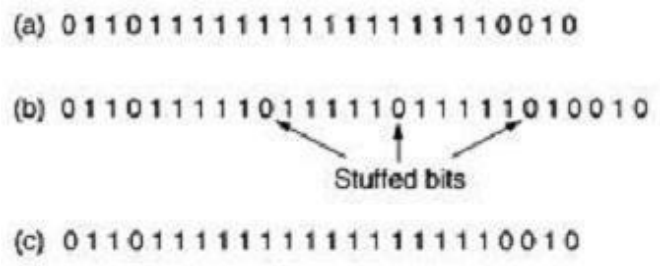


Figure. Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern. Thus, if the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data. The last method of framing is only applicable to networks in which the encoding on the physical medium contains some redundancy. For example, some LANs encode 1 bit of data by using 2 physical bits. Normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair. The scheme means that every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries. The combinations high-high and low-low are not used for data but are used for delimiting frames in some protocols.

As a final note on framing, many data link protocols use combination of a character count with one of the other methods for extra safety. When a frame arrives, the count field is used to locate the end of the frame. Only if the appropriate delimiter is present at that position and the checksum is correct is the frame accepted as valid. Otherwise, the input stream is scanned for the next delimiter.

2.5 FLOW AND ERROR CONTROL

Data communication requires at least two devices working together, one to send and the other to

receive. Even such a basic arrangement requires a great deal of coordination for an intelligible exchange to occur. The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.

2.5.1 FLOW CONTROL

Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer. In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver. The flow of data must not be allowed to overwhelm the receiver. Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data. The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily. Incoming data must be checked and processed before they can be used. The rate of such processing is often slower than the rate of transmission. For this reason, each receiving device has a block of memory, called a *buffer,* reserved for storing incoming data until they are processed. If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.

2.5.2 ERROR CONTROL

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

Noiseless Channels:

Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted.

1. Simplest Protocol

It has no flow or error control. It is a unidirectional protocol in which data frames are traveling in only one direction-from the sender to receiver. The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately.

Design

The sender site cannot send a frame until its network layer has a data packet to send. The receiver site cannot deliver a data packet to its network layer until a frame arrives.

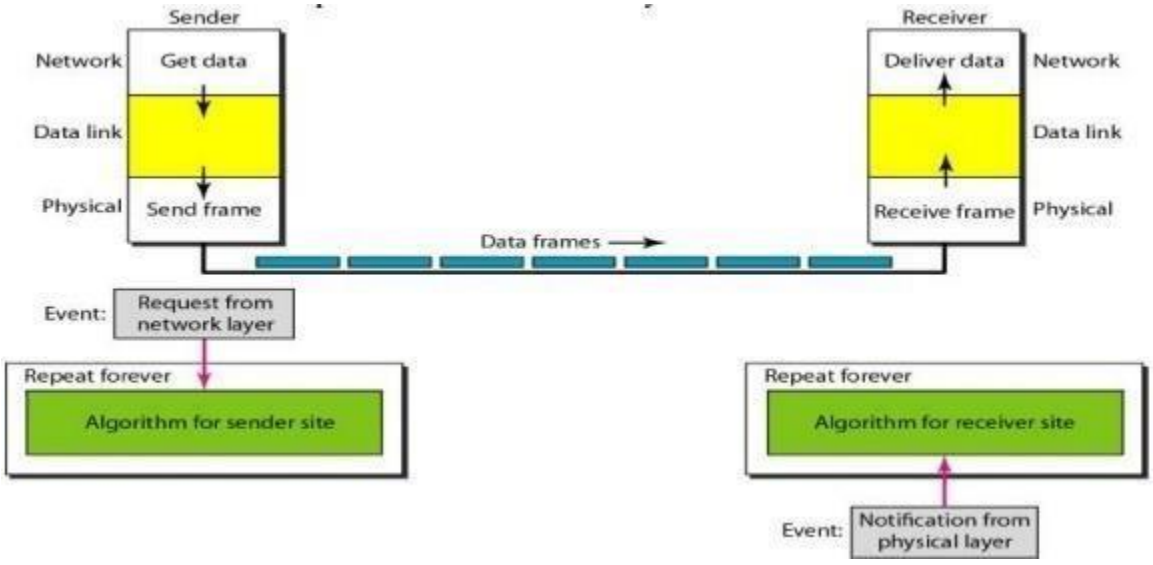


Figure 2.6 The design of the simplest protocol with no flow or error control

If the protocol is implemented as a procedure, we need to introduce the idea of events in the protocol. The procedure at the sender site is constantly running; there is no action until there is a request from the network layer. The procedure at the receiver site is also constantly running, but there is no action until notification from the physical layer arrives.

Example 2.1

It is very simple. 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. Note that the data frames are shown by tilted boxes; the height of the box defines the transmission time difference between the first bit and the last bit in the frame.

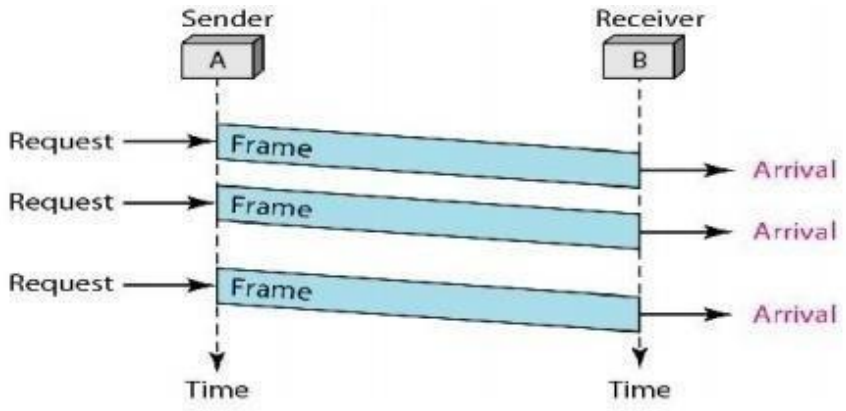


Figure 2.7 Flow diagram for Example 2.1

2. Stop-and-Wait Protocol

If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use.

In Stop-and-Wait Protocol the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame.

Design

Figure 2.8 illustrates the mechanism.

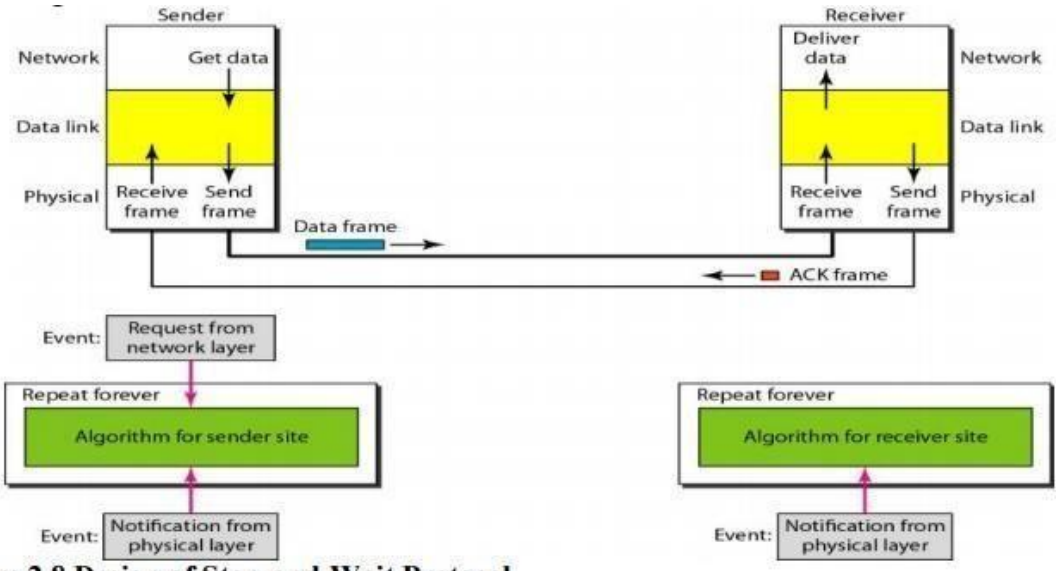


Figure 2.8 Design of Stop-and-Wait Protocol

Comparing this figure with Figure 2.6, we can see the traffic on the forward channel (from sender to receiver) and the reverse channel. At any time, there is either one data frame on the forward channel or one ACK frame on the reverse channel. We therefore need a half-duplex link.

Example 2.2

Figure 2.9 shows an example of communication using this protocol. It is still very simple. The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame. Note that sending two frames in the protocol involves the sender in four events and the receiver in two events.

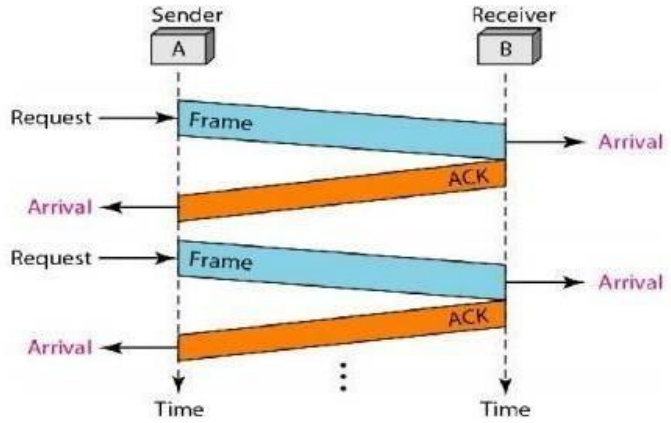


Figure 2.9 Flow diagram for Example 2.2

1. Stop-and-Wait Automatic Repeat Request

The Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ), adds a simple error control mechanism to the Stop-and-Wait Protocol. To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver.

Sequence Numbers

The protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame. For example, if we decide that the field is *m* bits long, the sequence numbers start from 0, go to *2m* - 1, and then are repeated.

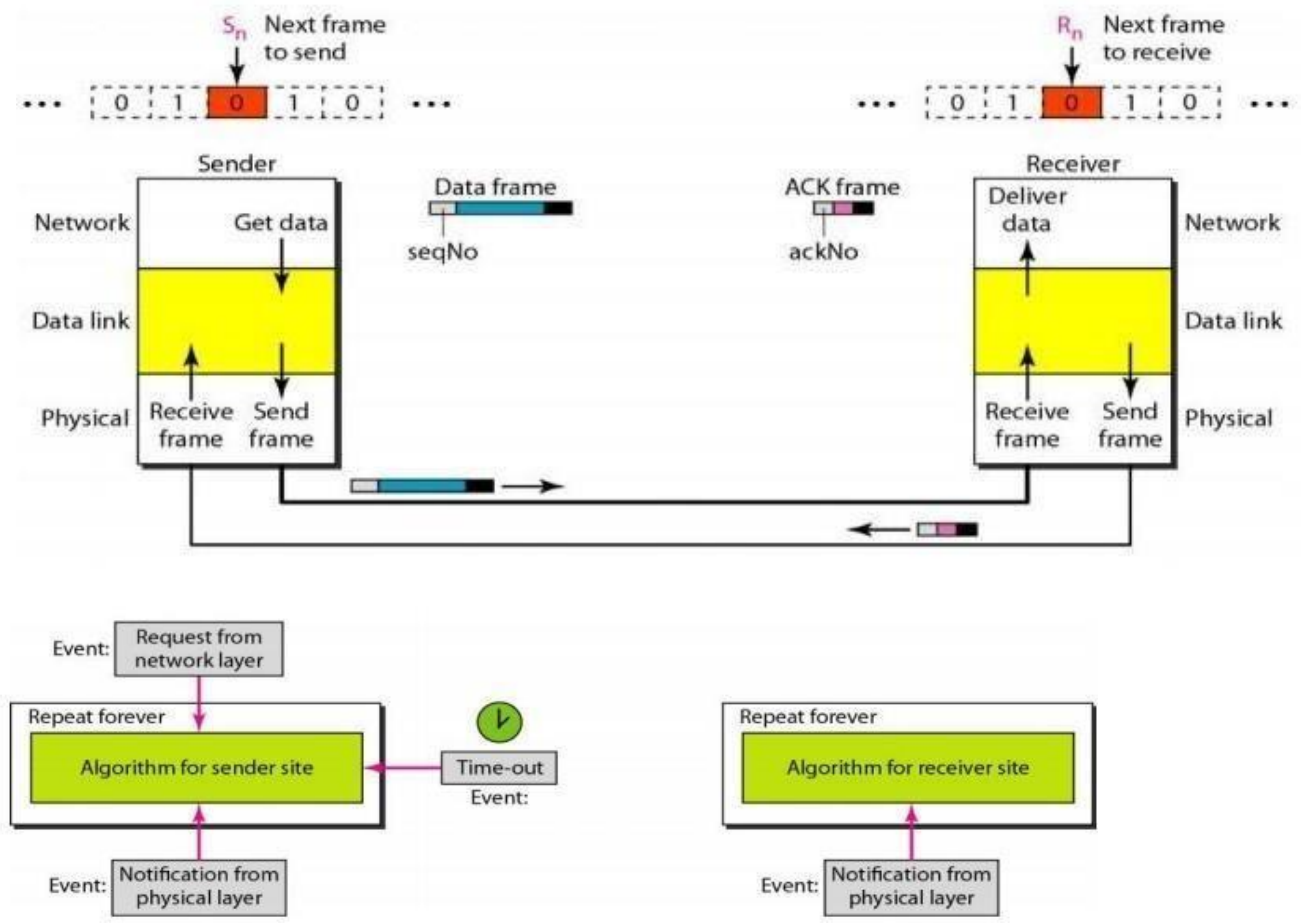


Figure 2.10 Design of the Stop-and-wait ARQ Protocol

Acknowledgment Numbers

Since the sequence numbers must be suitable for both data frames and ACK frames, we use this convention: The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver. For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1 (meaning frame 1 is expected next). If frame 1 has

arrived safe and sound, the receiver sends an ACK frame with acknowledgment 0 (meaning frame 0 is expected).

Design

Figure 2.10 shows the design of the Stop-and-Wait ARQ Protocol. The sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame. A data frames uses a seq No (sequence number); an ACK frame uses an ack No (acknowledgment number). The sender has a control variable, which we call *Sn* (sender, next frame to send), that holds the sequence number for the next frame to be sent (0 or 1).

The receiver has a control variable, which we call *Rn* (receiver, next frame expected), that holds the number of the next frame expected. When a frame is sent, the value of *Sn* is incremented (modulo-2), which means if it is 0, it becomes 1 and vice versa. When a frame is received, the value of *Rn* is incremented (modulo-2), which means if it is 0, it becomes 1 and vice versa. Three events can happen at the sender site; one event can happen at the receiver site. Variable *Sn* points to the slot that matches the sequence number of the frame that has been sent, but not acknowledged; *Rn* points to the slot that matches the sequence number of the expected frame.

Example 2.3

Frame 0 is sent and acknowledged. Frame 1 is lost and resent after the time-out. The resent frame 1 is acknowledged and the timer stops. Frame 0 is sent and acknowledged, but the acknowledgment is lost. The sender has no idea if the frame or the acknowledgment is lost, so after the time-out, it resends frame 0, which is acknowledged.

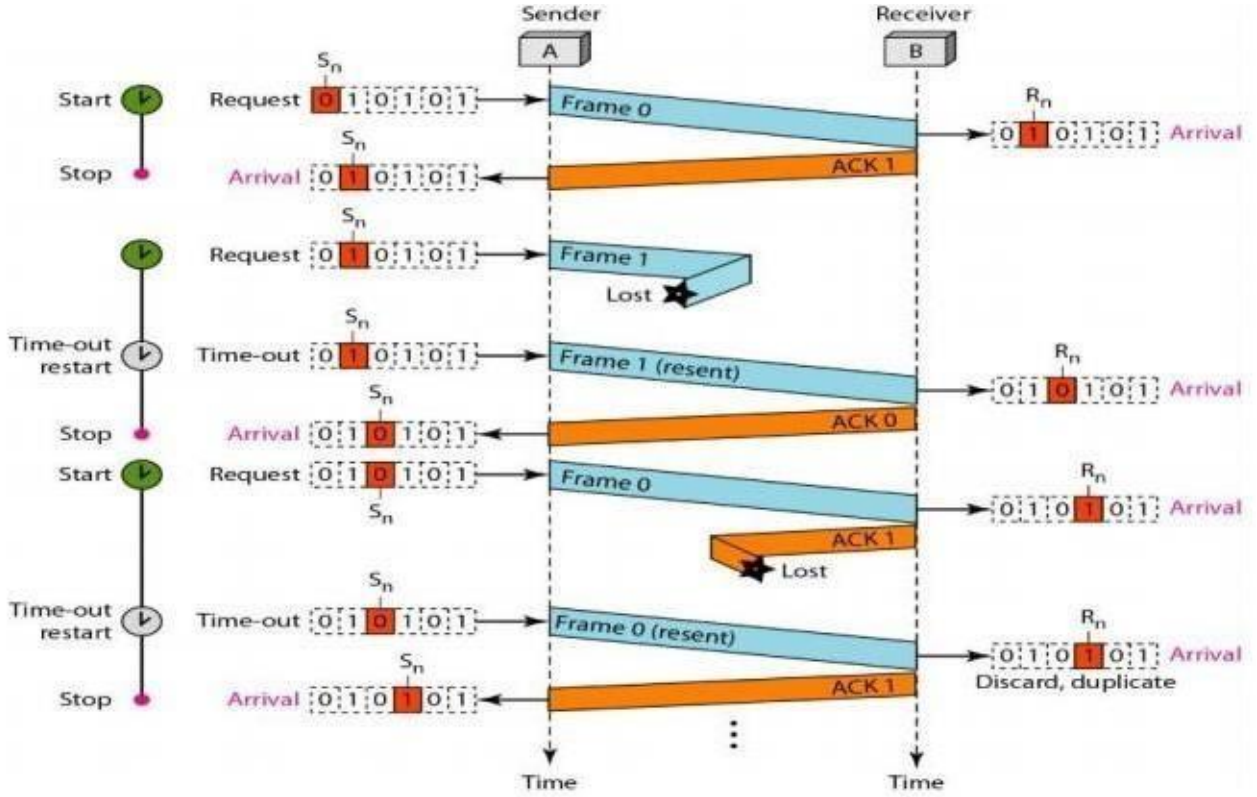


Figure 2.11 Flow diagram for Example 2.3

Example 2.4

Assume that, in a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 ms to make a round trip. What is the bandwidth-delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

Solution

The bandwidth-delay product is (1x106)x(20x10-3) =20,000bit

Pipelining

In networking and in other areas, a task is often begun before the previous task has ended. This is known as pipelining. There is no pipelining in Stop-and-Wait ARQ because we need to wait for a frame to reach the destination and be acknowledged before the next frame can be sent. However, pipelining does apply to our next two protocols because several frames can be sent before we receive news about the previous frames. Pipelining improves the efficiency of the

transmission if the number of bits in transition is large with respect to the bandwidth-delay product.

2. Go-Back-N Automatic Repeat Request

In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.

Sequence Numbers

Frames from a sending station are numbered sequentially. However, because we need to include the sequence number of each frame in the header, we need to set a limit. If the header of the frame allows m bits for the sequence number, the sequence numbers range from 0 to 2m - 1.

Sliding Window

The sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver. In other words, the sender and receiver need to deal with only part of the possible sequence numbers. The range which is the concern of the sender is called the send sliding window; the range that is the concern of the receiver is called the receive sliding window.

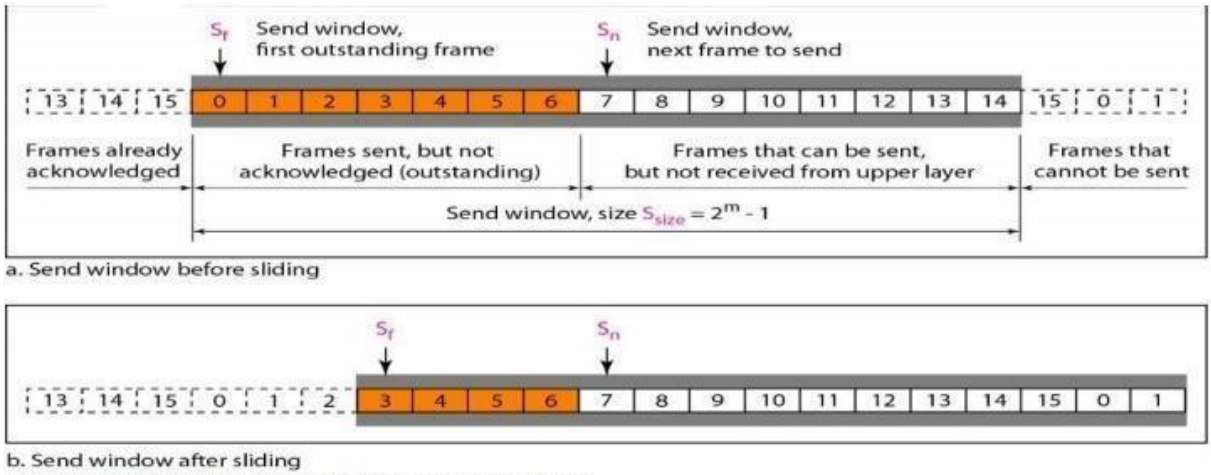


Figure 2.12 Send window for Go-Back-N ARQ

The sender does not worry about these frames and keeps no copies of them. The second region, colored in Figure 2.12 a, defines the range of sequence numbers belonging to the frames that are sent and have an unknown status.

The window itself is an abstraction; three variables define its size and location at any time. We call these variables *Sf(send* window, the first outstanding frame), *Sn* (send window, the next frame to be sent), and Ssize (send window, size). The variable *Sf* defines the sequence number of the first (oldest) outstanding frame. The variable *Sn* holds the sequence number that will be assigned to the next frame to be sent. Finally, the variable Ssize defines the size of the window, which is fixed in our protocol.

The receive window makes sure that the correct data frames are received and that the correct acknowledgments are sent. The size of the receive window is always 1.

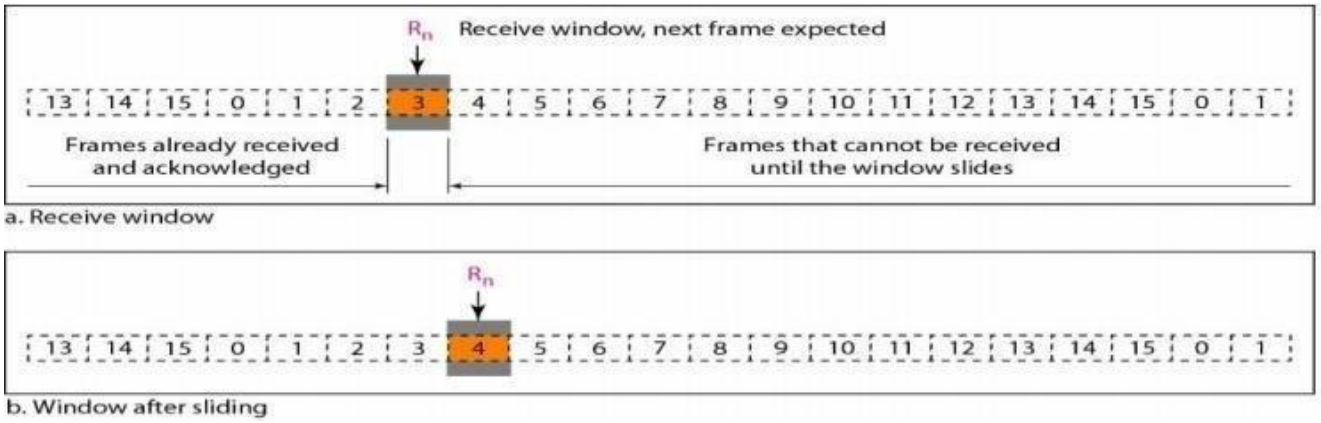


Figure 2.13 Receive window for Go-Back-N ARQ

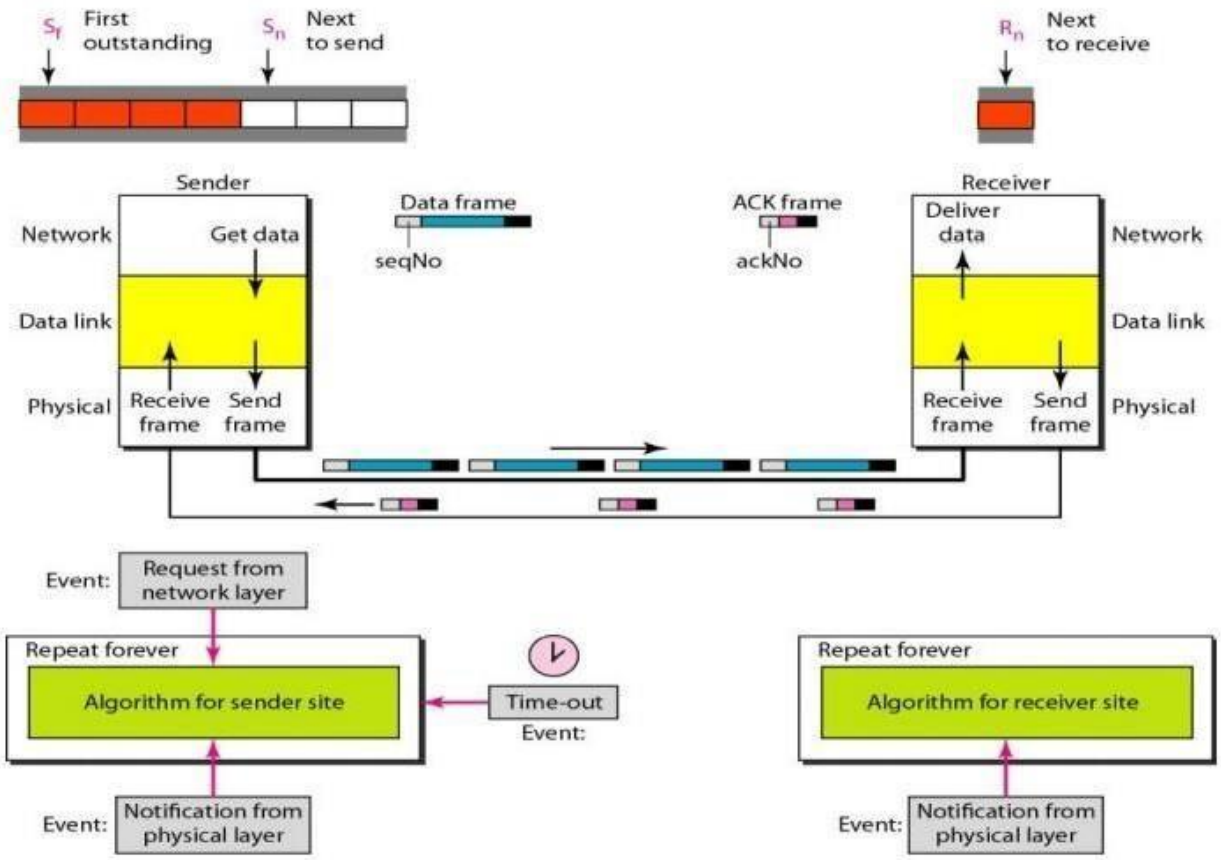


Figure 2.14 Design of Go-Back-N ARQ

Note that we need only one variable Rn (receive window, next frame expected) to define this abstraction. The sequence numbers to the left of the window belong to the frames already

received and acknowledged; the sequence numbers to the right of this window define the frames that cannot be received. Any received frame with a sequence number in these two regions is discarded. Only a frame with a sequence number matching the value of Rn is accepted and acknowledged. The receive window also slides, but only one slot at a time.

Design

Figure 2.14 shows the design for this protocol. As we can see, multiple frames can be in transit in the forward direction, and multiple acknowledgments in the reverse direction. The idea is

similar to Stop-and-Wait ARQ; the difference is that the send window allows us to have as many frames in transition as there are slots in the send window.

Send Window Size

We can now show why the size of the send window must be less than *2m.* As an example, we choose *m* =2, which means the size of the window can be *2m*- 1, or 3. Figure 2.15 compares a window size of 3 against a window size of 4. If the size of the window is 3 (less than 22) and all three acknowledgments are lost, the frame 0 timer expires and all three frames are resent. The receiver is now expecting frame 3, not frame 0, so the duplicate frame is correctly discarded.

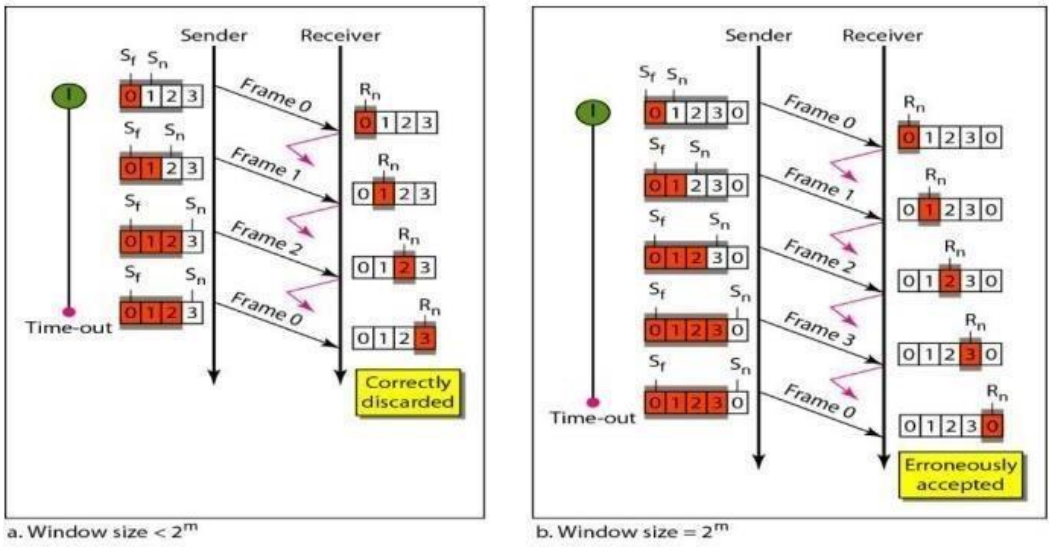


Figure 2.15 Window size for Go-Back-N ARQ

Example 2.4

Figure 2.16 shows an example of Go-Back-*N.* This is an example of a case where the forward channel is reliable, but the reverse is not. No data frames are lost, but some ACKs are delayed and one is lost. The example also shows how cumulative acknowledgments can help if acknowledgments are delayed or lost.

After initialization, there are seven sender events. Request events are triggered by data from the network layer; arrival events are triggered by acknowledgments from the physical layer. There is no time-out event here because all outstanding frames are acknowledged before the timer

expires. Note that although ACK 2 is lost, ACK 3 serves as both ACK 2 and ACK3. There are four receiver events, all triggered by the arrival of frames from the physical layer.

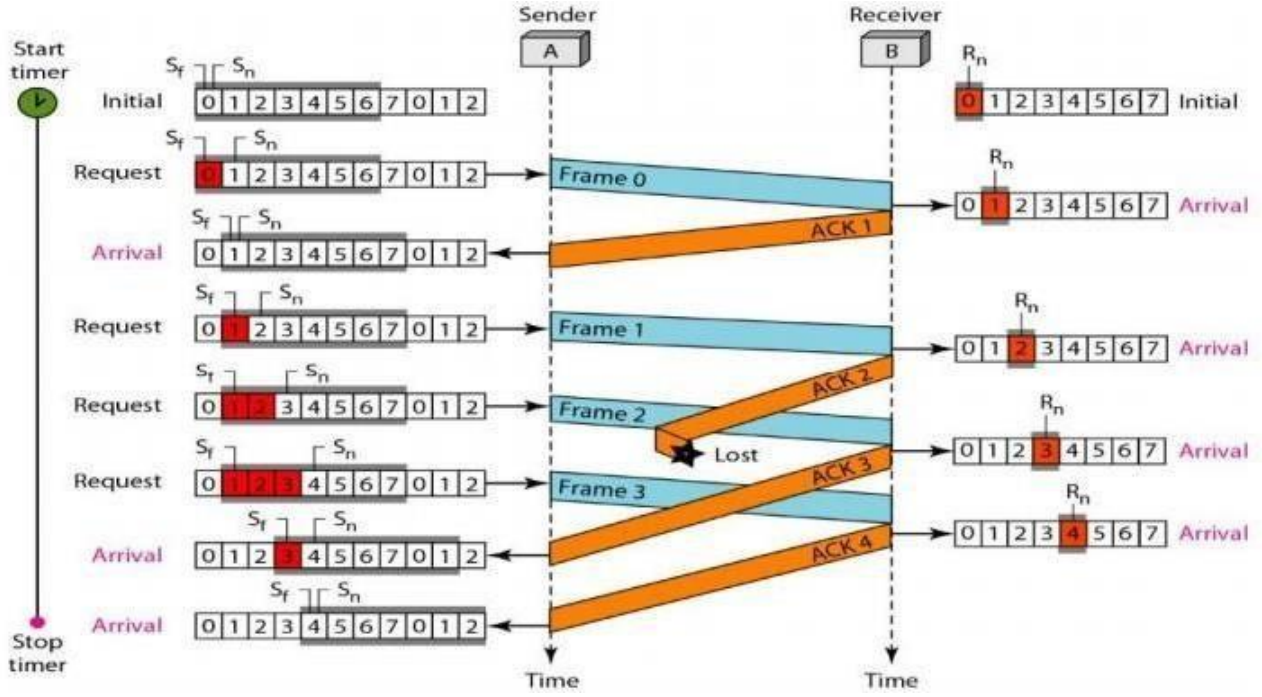


Figure 2.16 Flow diagram for Example 2.4

3. 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.

Windows

The Selective Repeat Protocol also uses two windows: a send window and a receive window. First, the size of the send window is much smaller; it is 2m- 1. Second, the receive window is the same size as the send window. The send window maximum size can be 2m- 1. For example, if m = 4, the sequence numbers go from 0 to 15, but the size of the window is just 8 (it is 15 in the Go-Back-N Protocol). The smaller window size means less efficiency in filling the pipe, but the fact that there are fewer duplicate frames can compensate for this.

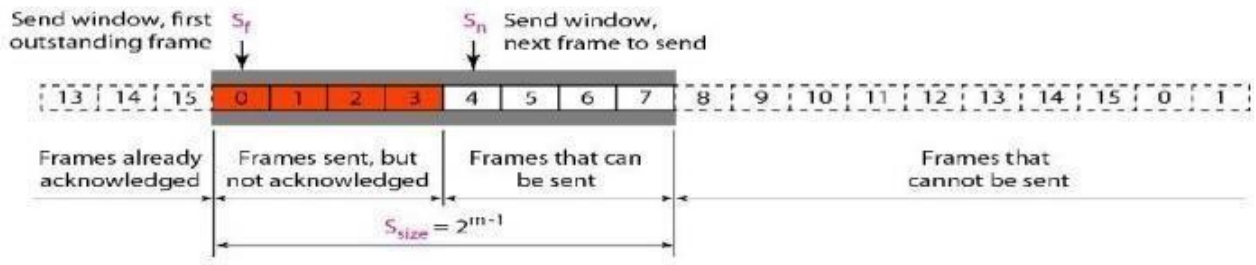


Figure 2.17 Send window for Selective Repeat ARQ

The receive window in Selective Repeat is totally different from the one in Go Back- N. First, the size of the receive window is the same as the size of the send window (2m- 1). Figure 2.18 shows the receive window in this protocol. Those slots inside the window that are colored define frames that have arrived out of order and are waiting for their neighbors to arrive before delivery to the network layer.

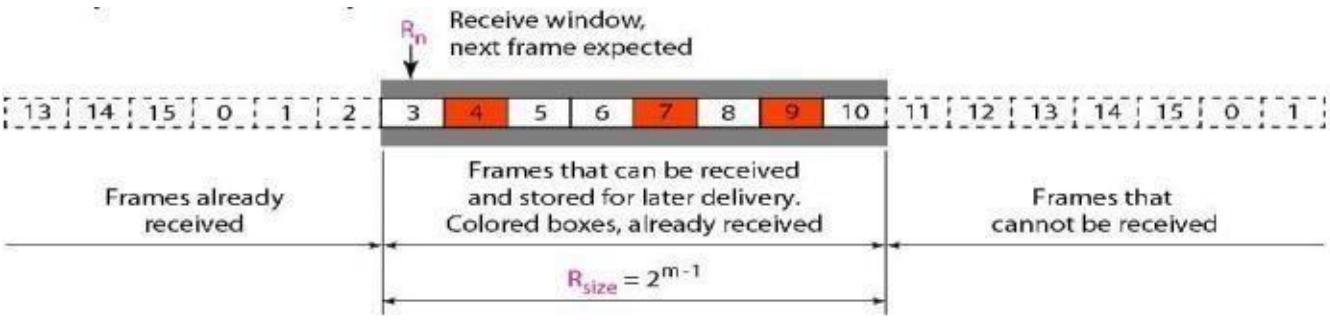


Figure 2.18 Receive window for Selective Repeat ARQ

Design

The design in this case is to some extent similar to the one we described for the 00Back-N, but more complicated, as shown in Figure 2.19.

Window Sizes

We can now show why the size of the sender and receiver windows must be at most on half of *2m.* For an example, we choose *m* = 2, which means the size of the window is *2m/2,* or 2. If thesize of the window is 2 and all acknowledgments are lost, the timer for frame 0 expires and

frame 0 is resent. However, this time, the window of the receiver expects to receive frame 0 (0 is part of the window), so it accepts frame 0, not as a duplicate, but as the first frame in the next cycle. This is clearly an error. In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of *2m*

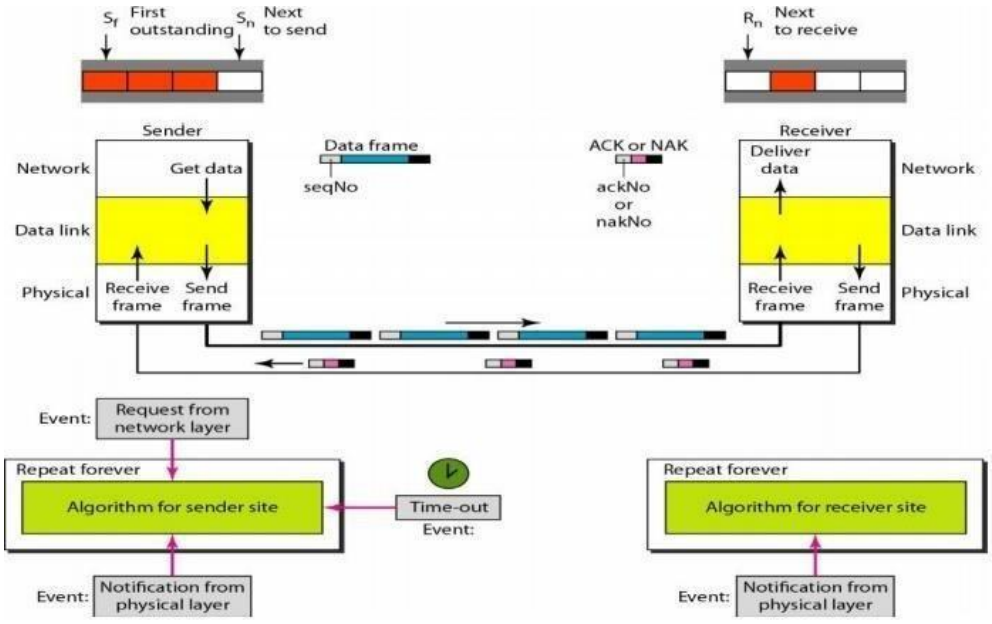


Figure 2.19 Design of Selective Repeat ARQ

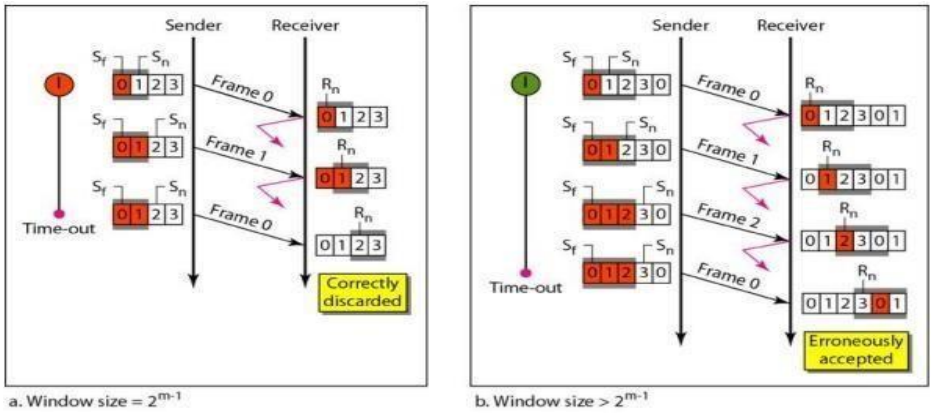


Figure 2.20 Selective Repeat ARQ, Window size

Example 2.5

Frame 1 is lost. We show how Selective Repeat behaves in this case.

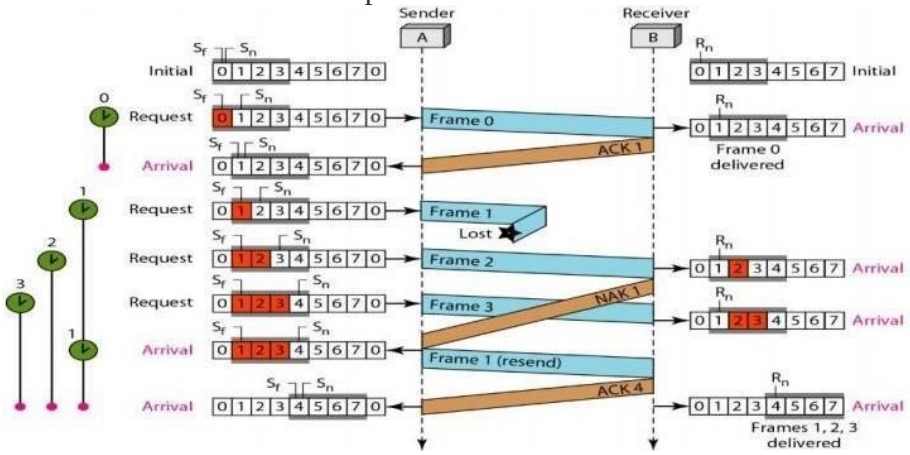


Figure 2.21 Flow diagram for Example 2.5

One main difference is the number of timers. Here, each frame sent or resent needs a timer, which means that the timers need to be numbered (0, 1, 2, and 3). The timer for frame 0 starts at the first request, but stops when the ACK for this frame arrives. The timer for frame 1 starts at the second request restarts when a NAK arrives, and finally stops when the last ACK arrives. The other two timers start when the corresponding frames are sent and stop at the last arrival event.

Piggybacking

The three protocols we discussed in this section are all unidirectional: data frames flow in only one direction although control information such as ACK and NAK frames can travel in the other direction. In real life, data frames are normally flowing in both directions: from node A to node B and from node B to node A. This means that the control information also needs to flow in both directions.

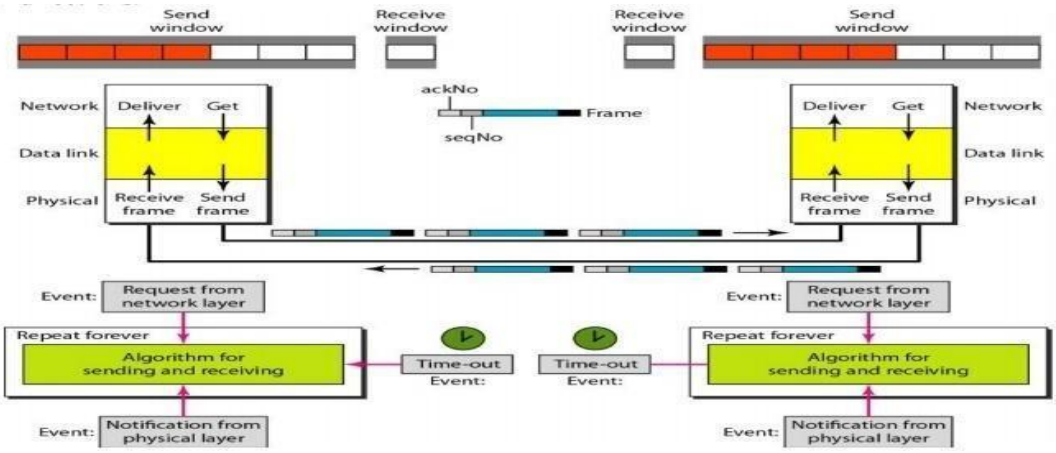


Figure 2.22 Design of Piggybacking in Go-Back-N ARQ

A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

2.6 HDLC—HIGH-LEVEL DATA LINK CONTROL

These are a group of closely related protocols that are a bit old but are still heavily used. They are all derived from the data link protocol first used in the IBM mainframe world: SDLC (Synchronous Data Link Control) protocol. After developing SDLC, IBM submitted it to ANSI and ISO for acceptance as U.S. and international standards, respectively. ANSI modified it to become ADCCP (Advanced Data Communication Control Procedure), and ISO modified it to become HDLC (High-level Data Link Control). CCITT then adopted and modified HDLC for its LAP (Link Access Procedure) as part of the X.25 network interface standard but later modified it again to LAPB, to make it more compatible with a later version of HDLC. The nice thing about standards is that you have so many to choose from. Furthermore, if you do not like any of them, you can just wait for next year's model. These protocols are based on the same principles. All are bit oriented, and all use bit stuffing for data transparency. They differ only in minor, but nevertheless irritating, ways. The discussion of bit-oriented protocols that follows is intended as a general introduction. For the specific details of any one protocol, please consult the appropriate definition.

All the bit-oriented protocols use the frame structure shown in below Fig. The Address field is primarily of importance on lines with multiple terminals, where it is used to identify one of the terminals. For point-to-point lines, it is sometimes used to distinguish commands from responses.

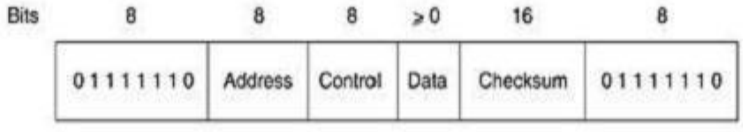


Fig. Frame format for bit-oriented protocols

The Control field is used for sequence numbers, acknowledgements, and other purposes, as discussed below.

The Data field may contain any information. It may be arbitrarily long, although the efficiency of the checksum falls off with increasing frame length due to the greater probability of multiple burst errors.

The Checksum field is a cyclic redundancy code. The frame is delimited with another flag sequence (01111110). On idle point-to-point lines, flag sequences are transmitted continuously. The minimum frame contains three fields and totals 32 bits, excluding the flags on either end. There are three kinds of frames: Information, Supervisory, and Unnumbered.

The contents of the Control field for these three kinds are shown in below Fig. The protocol uses a sliding window, with a 3-bit sequence number. Up to seven unacknowledged frames may be outstanding at any instant. The Seq field in below (a) is the frame sequence number. The Next field is a piggybacked acknowledgement. However, all the protocols adhere to the convention that instead of piggybacking the number of the last frame received correctly, they use the number of the first frame not yet received (i.e., the next frame expected). The choice of using the last frame received or the next frame expected is arbitrary; it does not matter which convention is used, provided that it is used consistently.

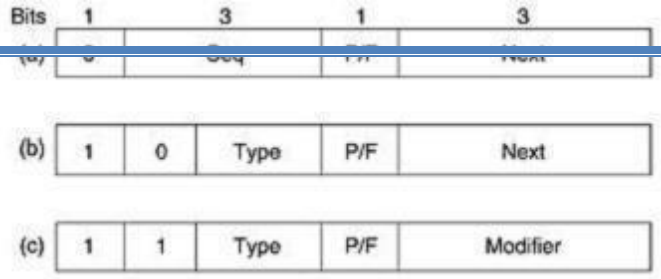


Fig. Control field of (a) an information frame, (b) a supervisory frame, (c) an unnumbered frame

The P/F bit stands for Poll/Final. It is used when a computer (or concentrator) is polling a group of terminals. When used as P, the computer is inviting the terminal to send data. All the frames sent by the terminal, except the final one, have the P/F bit set to P. The final one is set to F. In some of the protocols, the P/F bit is used to force the other machine to send a Supervisory frame immediately rather than waiting for reverse traffic onto which to piggyback the window information. The bit also has some minor uses in connection with the Unnumbered frames.

The various kinds of Supervisory frames are distinguished by the Type field. Type 0 is an acknowledgement frame (officially called RECEIVE READY) used to indicate the next frame expected. This frame is used when there is no reverse traffic to use for piggybacking.

Type 1 is a negative acknowledgement frame (officially called REJECT). It is used to indicate that a transmission error has been detected. The Next field indicates the first frame in sequence not received correctly (i.e., the frame to be retransmitted). The sender is required to retransmit all outstanding frames starting at Next. This strategy is similar to our protocol 5 rather than our protocol 6.

Type 2 is RECEIVE NOT READY. It acknowledges all frames up to but not including Next, just as RECEIVE READY does, but it tells the sender to stop sending. RECEIVE NOT READY is intended to signal certain temporary problems with the receiver, such as a shortage of buffers, and not as an alternative to the sliding window flow control. When the condition has been repaired, the receiver sends a RECEIVE READY, REJECT, or certain control frames.

Type 3 is the SELECTIVE REJECT. It calls for retransmission of only the frame specified. In this sense it is like our protocol 6 rather than 5 and is therefore most useful when the sender's window size is half the sequence space size, or less. Thus, if a receiver wishes to buffer out- of-sequence frames for potential future use, it can force the retransmission of any specific frame using Selective Reject. HDLC and ADCCP allow this frame type, but SDLC and LAPB do not allow it (i.e., there is no Selective Reject), and type 3 frames are undefined. The third class of frame is the Unnumbered frame. It is sometimes used for control purposes but can also carry data

when unreliable connectionless service is called for. The various bit-oriented protocols differ considerably here, in contrast with the other two kinds, where they are nearly identical. Five bits are available to indicate the frame type, but not all 32 possibilities are used.

2.7 PPP-THE POINT-TO-POINT PROTOCOL:

The Internet needs a point-to-point protocol for a variety of purposes, including router-to-router traffic and home user-to-ISP traffic. This protocol is PPP (Point-to-Point Protocol), which is defined in RFC 1661 and further elaborated on in several other RFCs (e.g., RFCs 1662 and 1663). PPP handles error detection, supports multiple protocols, allows IP addresses to be negotiated at connection time, permits authentication, and has many other features.

PPP provides three features:

1. A framing method that unambiguously delineates the end of one frame and the start of the next one. The frame format also handles error detection.
2. A link control protocol for bringing lines up, testing them, negotiating options, and bringing them down again gracefully when they are no longer needed. This protocol is called LCP (Link Control Protocol). It supports synchronous and asynchronous circuits and byte-oriented and bit-oriented encodings.
3. A way to negotiate network-layer options in a way that is independent of the network layer protocol to be used. The method chosen is to have a different NCP (Network Control Protocol) for each network layer supported.

To see how these pieces fit together, let us consider the typical scenario of a home user calling up an Internet service provider to make a home PC a temporary Internet host. The PC first calls the provider's router via a modem. After the router's modem has answered the phone and established a physical connection, the PC sends the router a series of LCP packets in the payload field of one or more PPP frames. These packets and their responses select the PPP parameters to be used.

Once the parameters have been agreed upon, a series of NCP packets are sent to configure the network layer. Typically, the PC wants to run a TCP/IP protocol stack, so it needs an IP address. There are not enough IP addresses to go around, so normally each Internet provider gets a block of them and then dynamically assigns one to each newly attached PC for the duration of its login session. If a provider owns n IP addresses, it can have up to n machines logged in simultaneously, but its total customer base may be many times that. The NCP for IP assigns the IP address. At this point, the PC is now an Internet host and can send and receive IP packets, just as hardwired hosts can. When the user is finished, NCP tears down the network layer connection and frees up the IP address. Then LCP shuts down the data link layer connection. Finally, the computer tells the modem to hang up the phone, releasing the physical layer connection.

The PPP frame format was chosen to closely resemble the HDLC frame format, since there was no reason to reinvent the wheel. The major difference between PPP and HDLC is that PPP is character oriented rather than bit oriented. In particular, PPP uses byte stuffing on dial-up modem lines, so all frames are an integral number of bytes. It is not possible to send a frame consisting of 30.25 bytes, as it is with HDLC. Not only can PPP frames be sent over dialup telephone lines, but they can also be sent over SONET or true bit-oriented HDLC lines (e.g., for router-router connections). The PPP frame format is shown in below fig.

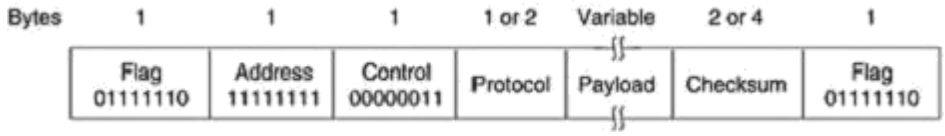


Fig. The PPP full frame format for unnumbered mode operation

All PPP frames begin with the standard HDLC flag byte (01111110), which is byte stuffed if it occurs within the payload field. Next comes the Address field, which is always set to the binary value 11111111 to indicate that all stations are to accept the frame. Using this value avoids the issue of having to assign data link addresses.

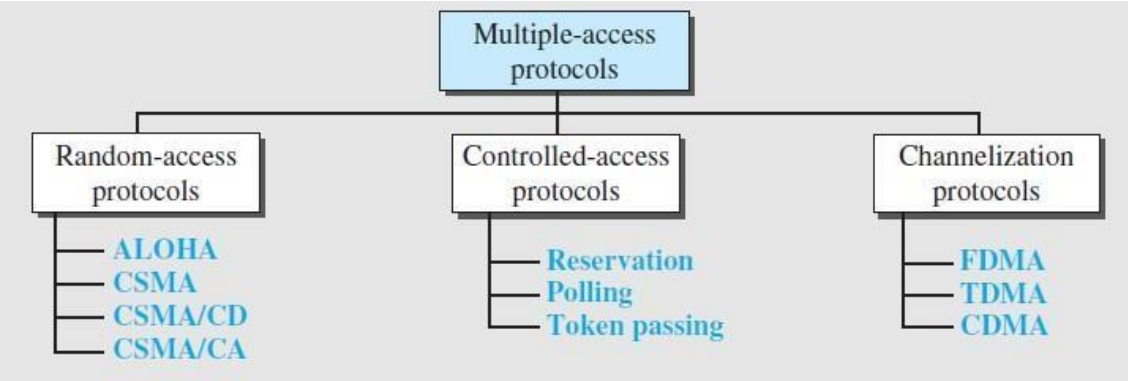
The Address field is followed by the Control field, the default value of which is 00000011. This value indicates an unnumbered frame. In other words, PPP does not provide reliable transmission using sequence numbers and acknowledgements as the default. In noisy environments, such as wireless networks, reliable transmission using numbered mode can be used. The exact details are

defined in RFC 1663, but in practice it is rarely used. Since the Address and Control fields are always constant in the default configuration, LCP provides the necessary mechanism for the two parties to negotiate an option to just omit them altogether and save 2 bytes per frame.

The fourth PPP field is the Protocol field. Its job is to tell what kind of packet is in the Payload field. Codes are defined for LCP, NCP, IP, IPX, AppleTalk, and other protocols. Protocols starting with a 0 bit are network layer protocols such as IP, IPX, OSI CLNP, XNS. Those starting with a 1 bit are used to negotiate other protocols. These include LCP and a different NCP for each network layer protocol supported. The default size of the Protocol field is 2 bytes, but it can be negotiated down to 1 byte using LCP. The Payload field is variable length, up to some negotiated maximum. If the length is not negotiated using LCP during line setup, a default length of 1500 bytes is used. Padding may follow the payload if need be. After the Payload field comes the Checksum field, which is normally 2 bytes, but a 4-byte checksum can be negotiated. In summary, PPP is a multiprotocol framing mechanism suitable for use over modems, HDLC bit-serial lines, SONET, and other physical layers. It supports error detection, option negotiation, header compression, and, optionally, reliable transmission using an HDLC type frame format.

2.8 CATEGEORIES MULTIPLE ACCESS PROTOCOLS

The multiple access protocols can be broadly classified into three categories namely Random access Protocols, Controlled access Protocols and Channelization Protocols (as given in below figure). Let us discuss in detail about the different protocols which are classified and as shown in below figure.



2.8.1 Random Access:

ALOHA:

In the 1970s, Norman Abramson and his colleagues at the University of Hawaii devised a new and elegant method to solve the channel allocation problem. Their work has been extended by many researchers since then (Abramson, 1985).

Although Abramson's work, called the ALOHA system, used ground-based radio broadcasting, the basic idea is applicable to any system in which uncoordinated users are competing for the use of a single shared channel. There are two versions of ALOHA: pure and slotted. They differ with respect to whether time is divided into discrete slots into which all frames must fit. Pure ALOHA does not require global time synchronization; slotted ALOHA does.

Pure ALOHA:

The basic idea of an ALOHA system is simple: let users transmit whenever they have data to be sent. There will be collisions, of course, and the colliding frames will be damaged. However, due to the feedback property of broadcasting, a sender can always find out whether its frame was destroyed by listening to the channel, the same way other users do. With a LAN, the feedback is immediate; with a satellite, there is a delay of 270 msec before the sender knows if the transmission was successful. If listening while transmitting is not possible for some reason, acknowledgements are needed. If the frame was destroyed, the sender just waits a random amount of time and sends it again. The waiting time must be random or the same frames will collide over and over, in lockstep. Systems in which multiple users share a common channel in a way that can lead to conflicts are widely known as contention systems.

A sketch of frame generation in an ALOHA system is given in below fig. We have made the frames all the same length because the throughput of ALOHA systems is maximized by having a uniform frame size rather than by allowing variable length frames.

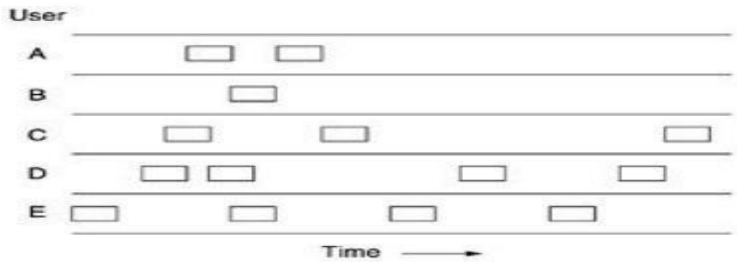


Fig. In pure ALOHA, frames are transmitted at completely arbitrary times.

Whenever two frames try to occupy the channel at the same time, there will be a collision and both will be garbled. If the first bit of a new frame overlaps with just the last bit of a frame almost finished, both frames will be totally destroyed and both will have to be retransmitted later. The checksum cannot (and should not) distinguish between a total loss and a near miss.

Let the "frame time" denote the amount of time needed to transmit the standard, fixed length frame (i.e., the frame length divided by the bit rate). At this point we assume that the infinite population of users generates new frames according to a Poisson distribution with mean N frames per frame time. (The infinite-population assumption is needed to ensure that N does no decrease as users become blocked.) If N > 1, the user community is generating frames at a higher rate than the channel can handle, and nearly every frame will suffer a collision. For reasonable throughput we would expect 0 < N < 1. In addition to the new frames, the stations also generate retransmissions of frames that previously suffered collisions. Let us further assume that the probability of k transmission attempts per frame time, old and new combined, is also Poisson, with mean G per frame time. Clearly, G≥N. At low load (i.e., N 0), there will be few collisions, hence few retransmissions, so G N. At high load there will be many collisions, so G > N. Under all loads, the throughput, S, is just the offered load, G, times the probability, P0, of a transmission succeeding—that is, S = GP0, where P0 is the probability that a frame does not suffer a collision.

A frame will not suffer a collision if no other frames are sent within one frame time of its start, as shown in below fig.

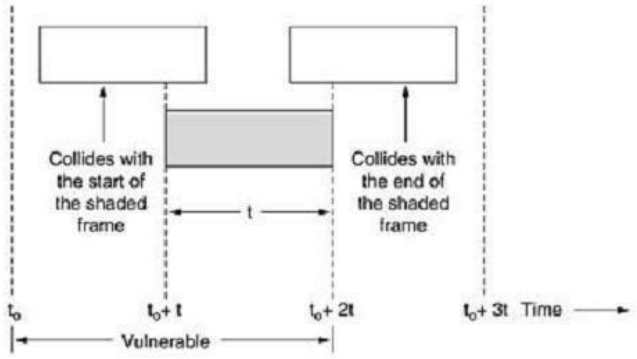


Fig. Vulnerable period for the shaded frame

Under what conditions will the shaded frame arrive undamaged? Let t be the time required to send a frame. If any other user has generated a frame between time t0 and t0+tr, the end of that frame will collide with the beginning of the shaded one. In fact, the shaded frame's fate was

already sealed even before the first bit was sent, but since in pure ALOHA a station does not listen to the channel before transmitting, it has no way of knowing that another frame was already underway. Similarly, any other frame started between t0+t and t0+2t will bump into the end of the shaded frame.

The probability that k frames are generated during a given frame time is given by the Poisson distribution:

Equation

$$\Pr[k] = \frac{G^k e^{-G}}{k!}$$

so the probability of zero frames is just e-G. In an interval two frame times long, the mean number of frames generated is 2G. The probability of no other traffic being initiated during the entire vulnerable period is thus given by P0 = e -2G. Using S = GP0, we get

$$S = Ge^{-2G}$$

The relation between the offered traffic and the throughput is shown in Fig. 4-3. The maximum throughput occurs at G = 0.5, with S = 1/2e, which is about 0.184. In other words, the best we can hope for is a channel utilization of 18 per cent. This result is not very encouraging, but with everyone transmitting at will, we could hardly have expected a 100 per cent success rate.

Slotted ALOHA:

In 1972, Roberts published a method for doubling the capacity of an ALOHA system (Robert, 1972). His proposal was to divide time into discrete intervals, each interval corresponding to one frame. This approach requires the users to agree on slot boundaries. One way to achieve synchronization would be to have one special station emit a pip at the start of each interval, like a clock.

In Roberts' method, which has come to be known as slotted ALOHA, in contrast to Abramson's pure ALOHA, a computer is not permitted to send whenever a carriage return is typed. Instead, it is required to wait for the beginning of the next slot. Thus, the continuous pure ALOHA is turned into a discrete one. Since the vulnerable period is now halved, the probability of no other traffic during the same slot as our test frame is e-G which leads to

\Equation

$$S = Ge^{-G}$$

As you can see from Fig.3, slotted ALOHA peaks at G = 1, with a throughput of S=1/e or about 0.368, twice that of pure ALOHA. If the system is operating at G = 1, the probability of an empty slot is 0.368. The best we can hope for using slotted ALOHA is 37 percent of the slots empty, 37 percent successes, and 26 percent collisions. Operating at higher values of G reduces the number of empties but increases the number of collisions exponentially.

To see how this rapid growth of collisions with G comes about, consider the transmission of a test frame. The probability that it will avoid a collision is e-G, the probability that all the other users are silent in that slot. The probability of a collision is then just 1 – e-G. The probability of a transmission requiring exactly k attempts, (i.e., k – 1 collisions followed by one success) is

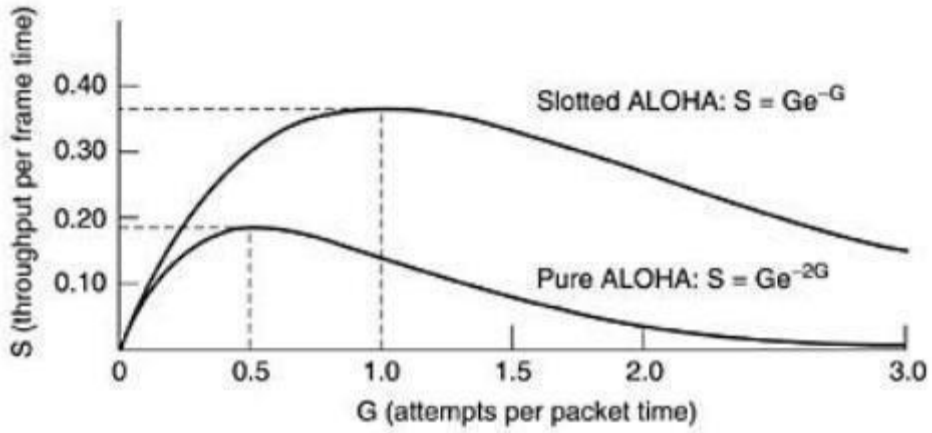


Fig.3 Throughput versus offered traffic for ALOHA systems.

$$P_k = e^{-G}(1 - e^{-G})^{k-1}$$

The expected number of transmissions, E, per carriage return typed is then

$$E = \sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} ke^{-G}(1 - e^{-G})^{k-1} = e^{G}$$

As a result of the exponential dependence of E upon G, small increases in the channel load can drastically reduce its performance.

CSMA

Carrier Sense Multiple Access Protocols:

With slotted ALOHA the best channel utilization that can be achieved is 1/e. This is hardly surprising, since with stations transmitting at will, without paying attention to what the other stations are doing, there are bound to be many collisions. In local area networks, however, it is possible for stations to detect what other stations are doing, and adapt their behaviour accordingly. These networks can achieve a much better utilization than 1/e. In this section we will discuss some protocols for improving performance. Protocols in which stations listen for a carrier (i.e., a transmission) and act accordingly are called carrier sense protocols. A number of them have been proposed. Kleinrock and Tobagi (1975) have analysed several such protocols in detail. Below we will mention several versions of the carrier sense protocols.

1.1-persistent CSMA:

The first carrier sense protocol that we will study here is called **1-persistent CSMA** (Carrier Sense Multiple Access). When a station has data to send, it first listens to the channel to see if anyone else is transmitting at that moment. If the channel is busy, the station waits until it becomes idle. When the station detects an idle channel, it transmits a frame. If a collision occurs, the station waits a random amount of time and starts all over again. The protocol is called 1-persistent because the station transmits with a probability of 1 when it finds the channel idle.

The propagation delay has an important effect on the performance of the protocol. There is a small chance that just after a station begins sending, another station will become ready to send and sense the channel. If the first station's signal has not yet reached the second one, the latter will sense an idle channel and will also begin sending, resulting in a collision. The longer the propagation delay, the more important this effect becomes, and the worse the performance of the protocol. Even if the propagation delay is zero, there will still be collisions. If two stations become ready in the middle of a third station's transmission, both will wait politely until the transmission ends and then both will begin transmitting exactly simultaneously, resulting in a collision. If they were not so impatient, there would be fewer collisions. Even so, this protocol is far better than pure ALOHA because both stations have the decency to desist from interfering with the third station's frame. Intuitively, this approach will lead to a higher performance than pure ALOHA. Exactly the same holds for slotted ALOHA.

2. Non-persistent CSMA:

A second carrier sense protocol is **nonpersistent CSMA**. In this protocol, a conscious attempt is made to be less greedy than in the previous one. Before sending, a station senses the channel. If no one else is sending, the station begins doing so itself. However, if the channel is already in use, the station does not continually sense it for the purpose of seizing it immediately upon detecting the end of the previous transmission. Instead, it waits a random period of time and then repeats the algorithm. Consequently, this algorithm leads to better channel utilization but longer delays than 1-persistent CSMA.

3. P-persistent CSMA:

The last protocol is **p-persistent CSMA**. It applies to slotted channels and works as follows. When a station becomes ready to send, it senses the channel. If it is idle, it transmits with a probability p. With a probability q = 1 - p, it defers until the next slot. If that slot is also idle, It either transmits or defers again, with probabilities p and q. This process is repeated until either the frame has been transmitted or another station has begun transmitting. In the latter case, the unlucky station acts as if there had been a collision (i.e., it waits a random time and starts again). If the station initially senses the channel busy, it waits until the next slot and applies the above algorithm. Figure 4 shows the computed throughput versus offered traffic for all three protocols, as well as for pure and slotted ALOHA.

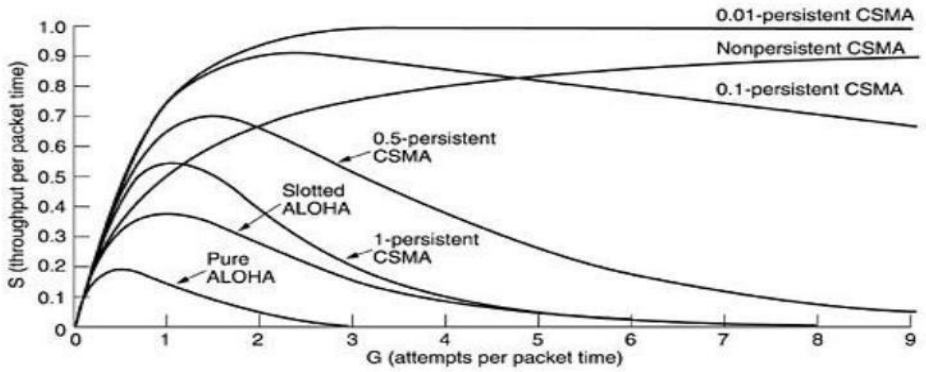


Fig. Comparison of the channel utilization versus load for various random access Protocols.

4. CSMA with Collision Detection:

Persistent and non persistent CSMA protocols are clearly an improvement over ALOHA because they ensure that no station begins to transmit when it senses the channel busy. Another improvement is for stations to abort their transmissions as soon as they detect a collision. In other words, if two stations sense the channel to be idle and begin transmitting simultaneously, they will both detect the collision almost immediately. Rather than finish transmitting their frames, which are irretrievably garbled anyway, they should abruptly stop transmitting as soon as the collision is detected. Quickly terminating damaged frames saves time and bandwidth.

This protocol, known as CSMA/CD (CSMA with Collision Detection) is widely used on LANs in the MAC sublayer. In particular, it is the basis of the popular Ethernet LAN, so it is worth devoting some time to looking at it in detail. CSMA/CD, as well as many other LAN protocols, uses the conceptual model of below fig. At the point marked t0, a station has finished transmitting its frame. Any other station having a frame to send may now attempt to do so. If two or more stations decide to transmit simultaneously, there will be a collision. Collisions can be detected by looking at the power or pulse width of the received signal and comparing it to the transmitted signal.

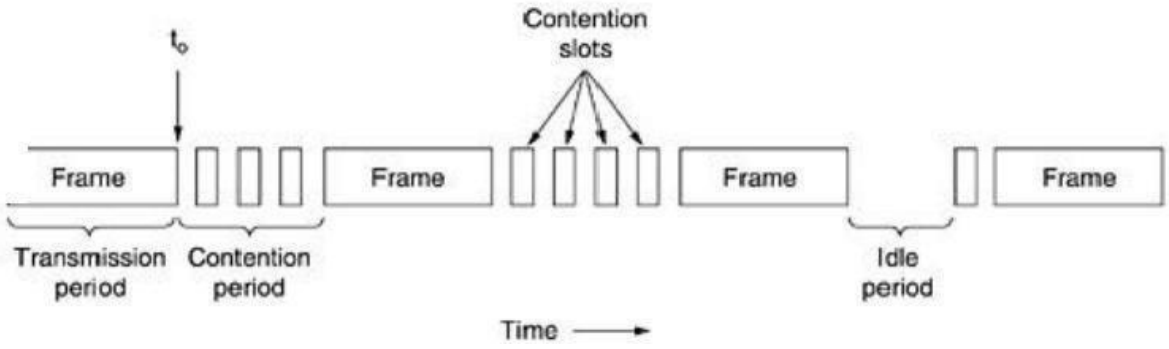


Fig. CSMA/CD can be in one of three states: contention, transmission, or idle

After a station detects a collision, it aborts its transmission, waits a random period of time, and then tries again, assuming that no other station has started transmitting in the meantime. Therefore, our model for CSMA/CD will consist of alternating contention and transmission periods, with idle periods occurring when all stations are quiet (e.g., for lack of work).

Now let us look closely at the details of the contention algorithm. Suppose that two stations both begin transmitting at exactly time t0. How long will it take them to realize that there has been a

collision? The answer to this question is vital to determining the length of the contention period and hence what the delay and throughput will be. The minimum time to detect the collision is then just the time it takes the signal to propagate from one station to the other.

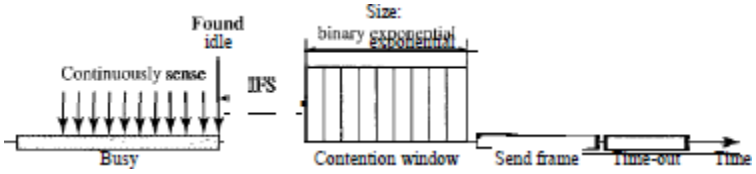
Based on this reasoning, you might think that a station not hearing a collision for a time equal to the full cable propagation time after starting its transmission could be sure it had seized the cable. By "seized," we mean that all other stations knew it was transmitting and would not interfere. This conclusion is wrong. Consider the following worst-case scenario. Let the time for a signal to propagate between the two farthest stations be . At t0, one station begins transmitting. At , an instant before the signal arrives at the most distant station, that station also begins transmitting. Of course, it detects the collision almost instantly and stops, but the little noise burst caused by the collision does not get back to the original station until time . In other words, in the worst case a station cannot be sure that it has seized the channel until it has transmitted for without hearing a collision. For this reason we will model the contention interval as a slotted ALOHA system with slot width . On a 1-km long coaxial cable, . For simplicity we will assume that each slot contains just 1 bit. Once the channel has been seized, a station can transmit at any rate it wants to, of course, not just at 1 bit per sec.

CSMA with Collision Avoidance:

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 *(CSMAICA)* was invented for this network. Collisions are avoided through the use of CSMAICA's three strategies: the inter frame space, the contention window, and acknowledgments, as shown in below figure.



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 inter frame 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. If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time (described next). 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.

Contention Window

The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy. This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time. This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting station. One interesting point about the contention window is that the station needs to sense the channel after each time slot. However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle. This gives priority to the station with the longest waiting time.

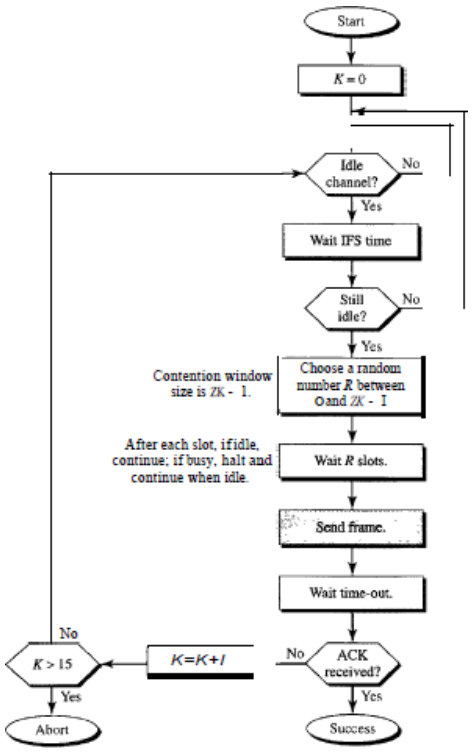
Acknowledgment

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

Figure 12.17 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.

CSMAICA and Wireless Networks. CSMAICA was mostly intended for use in wireless networks. The procedure described above, however, is not sophisticated enough to handle some particular issues related to wireless networks, such as hidden terminals or exposed terminals. We will see how these issues are solved by augmenting the above protocol with hand-shaking features.



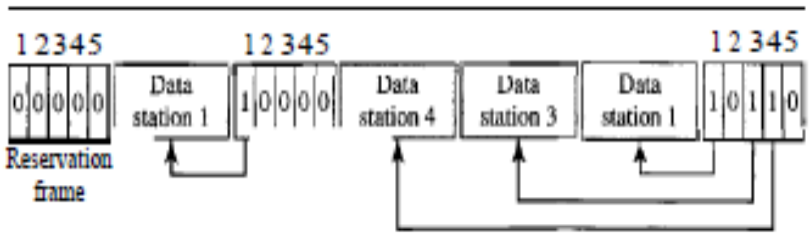
2.8.2 CONTROLLED ACCESS

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.

Reservation

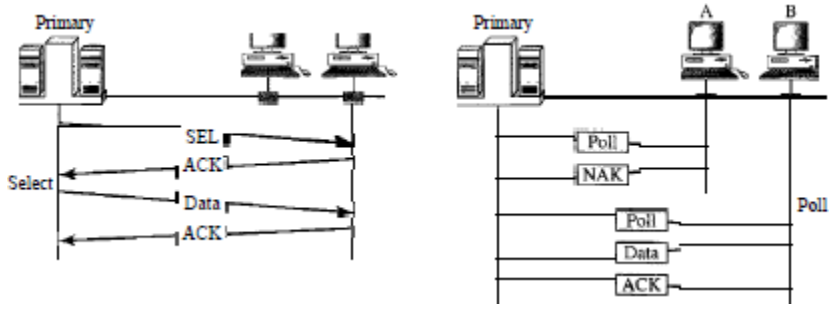
In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.

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. Below figure 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.



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 below figure).



If the primary wants to receive data, it asks the secondaries 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.

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.

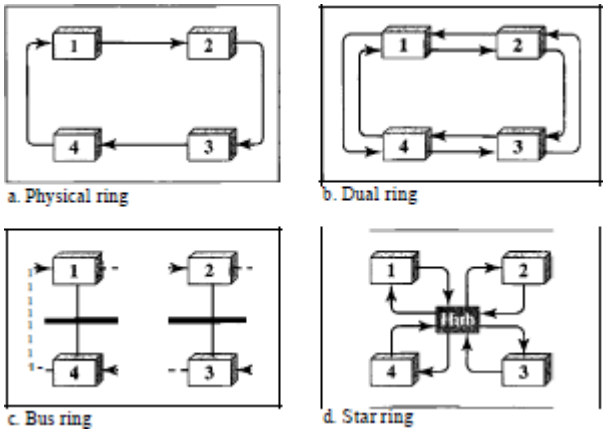
But how is the right to access the channel passed from one station to another? In this method, a special packet called a token circulates through the ring. The possession of the token gives the station the right to access the channel and send its data. When a station has some data to send, it waits until 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 then ext 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. Figure 12.20 show four different physical topologies that can create a logical ring.



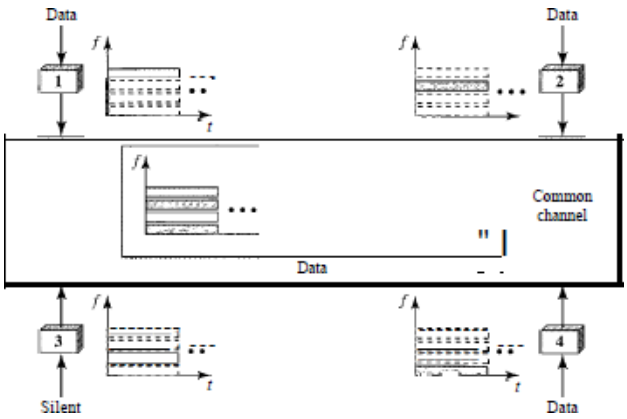
a. Physical ring
b. Dual ring
c. Bus ring
d. Star ring

2.8.3 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.

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 band pass filter to confine the transmitter frequencies. To prevent station interferences, the allocated bands are separated from one another by small *guard bands.* Below figure shows the idea of 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 will see in Chapter 16 how this feature can be used in cellular telephone systems.

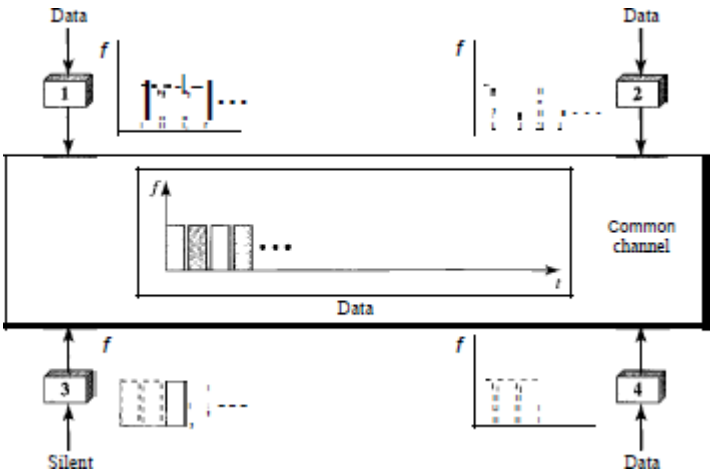
We need to emphasize that although FDMA and FDM conceptually seem similar, there are differences between them. FDM, as we saw in Chapter 6, is a physical layer technique that combines the loads from low-bandwidth channels and transmits them by using a high-bandwidth channel. The channels that are combined are low-pass. The multiplexer modulates the signals,

combines them, and creates a band pass signal. The bandwidth of each channel is shifted by the multiplexer.

FDMA, 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.

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 is assigned time slot. Below figure shows the idea behind 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 *guardtimes.* Synchronization is normally accomplished by having some synchronization bits (normally refened 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.

CDMA: Code Division Multiple Access

While TDM and FDM assign time slots and frequencies, respectively to the nodes, CDMA assigns a different code to each node. Each node then uses its unique code to encode the data bits it sends. If the codes are chosen carefully, CDMA networks have the wonderful property that different nodes can transmit simultaneously and yet have their respective receivers correctly receive a sender's encoded data bits in spite of interfering transmissions by other nodes.

CDMA has been used in military systems for some time and now has widespread civilian use, particularly in cellular telephony. Because CDMA's use is so tightly tied to wireless channels. It will suffice to know that CDMA codes, like time slots in TDM and frequencies in FDM, can be allocated to the multiple access channel users.

UNIT-III
EATHERNET

3.0 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 Figure 1.

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.

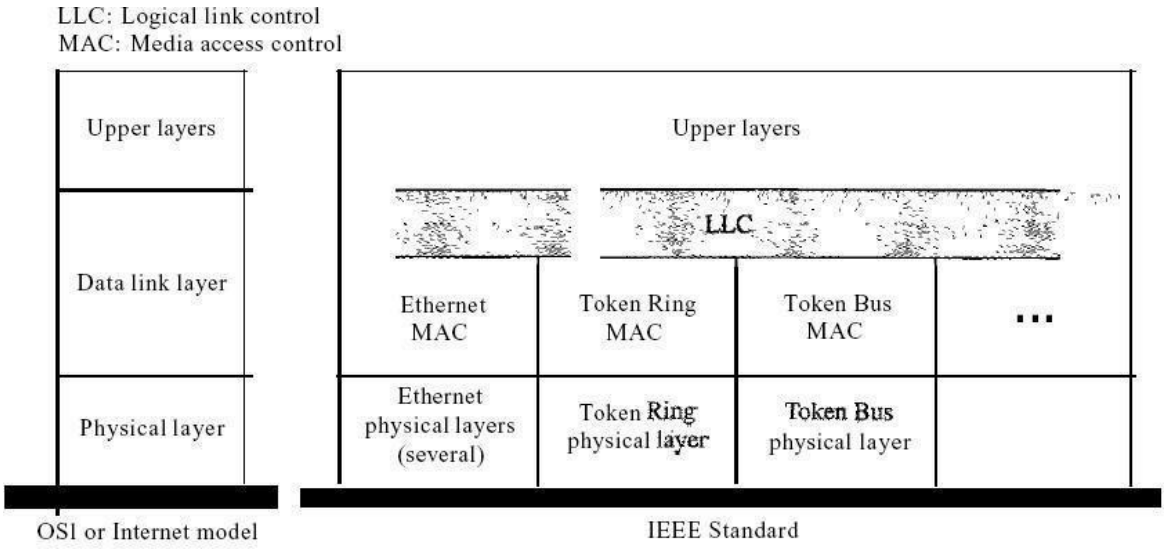


Figure 1 *IEEE standardfor LANs*

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 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. Figure 1 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 Figure 2.

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.

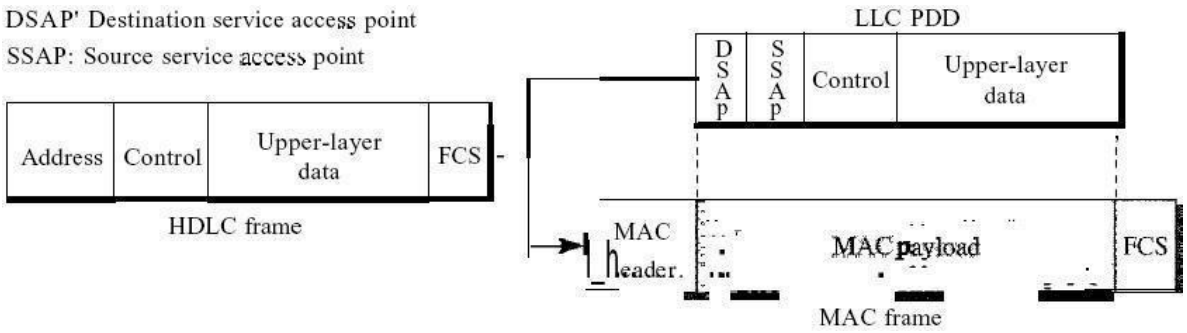


Figure 13.2 *HDLCframe compared with LLC and MAC frames*

Media Access Control (MAC)

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 tokenpassing 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.

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 specifications for each Ethernet implementations.

3.1 Standard Ethernet

The Ethernet has under gone four evolutions so far as depicted in the following figure. The detailed description of different evolutions of ether has given below.

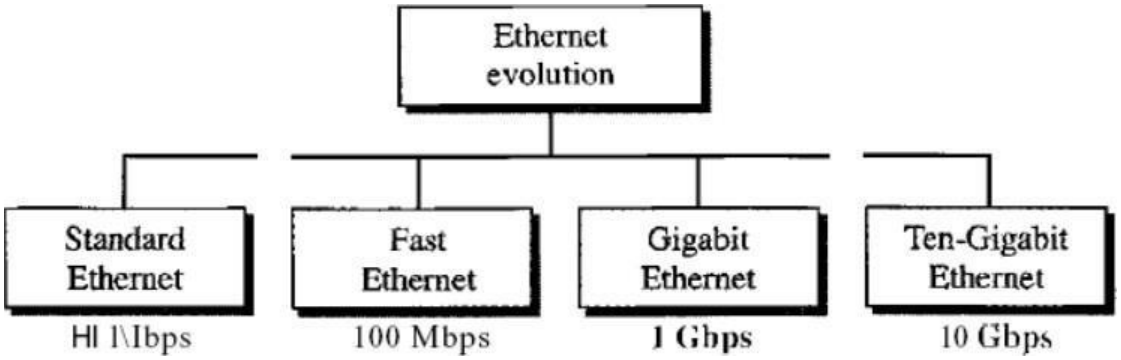


Figure 1 Ethernet evolution through four generations

3.1.1 MAC Sublayer

In Standard Ethernet, the MAC sub layer 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 CRC. 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 Figure 2.

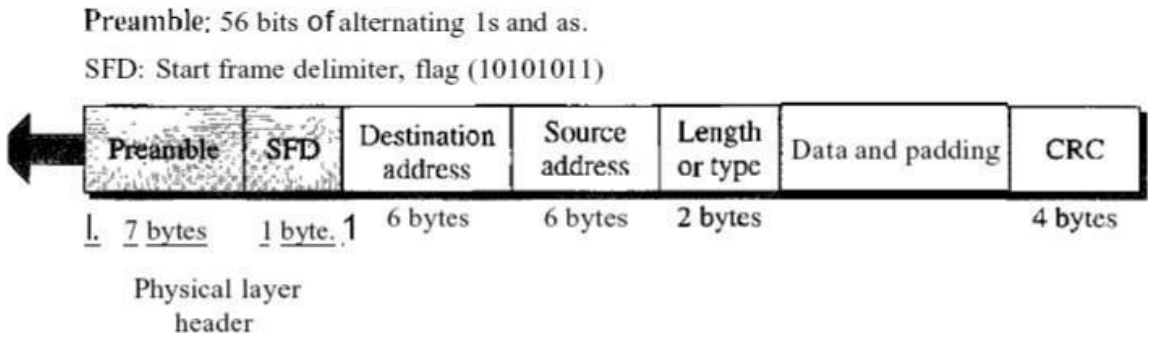


Figure 2 802.3 MAC frame

- D Preamble. The first field of the 802.3 frame contains 7 bytes (56 bits) of alternating Os and Is 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.
- D Start frame delimiter (SFD). The second field (l 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.
- 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, as shown in fig.

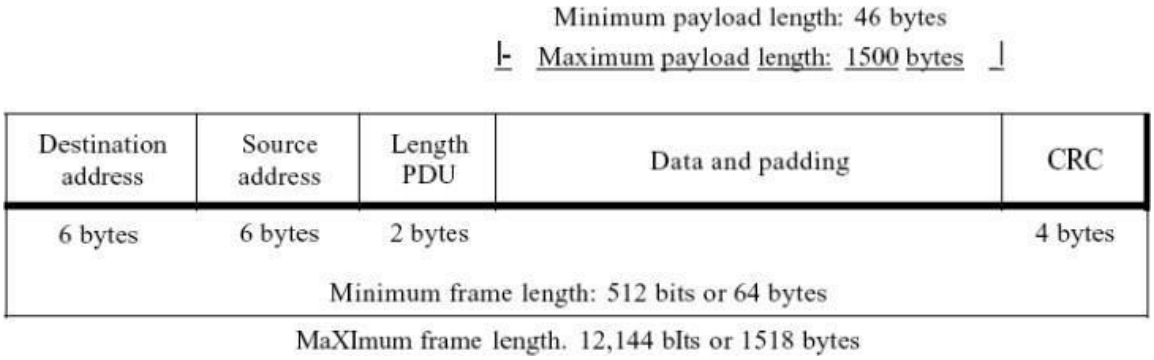


Figure 3 Minimum and maximum lengths

The minimum length restriction is required for the correct operation of CSMA/CD. 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.

Frame length:

Minimum: 64 bytes (512 bits)

Maximum: 1518 bytes (12,144 bits)

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 stationwith a 6-byte 87 physical address. As shown in Fig 6, the Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes.

06:01 :02:01:2C:4B

6 bytes =12 hex digits =48 bits.

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.

Figure 5 shows how to distinguish a unicast address from a multicast address. If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast.

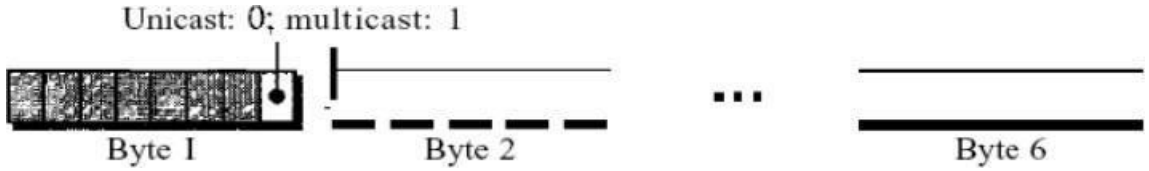


Figure 5 Unicast and multicast addresses

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.

3.1.2 Physical Layer

The Standard Ethernet defines several physical layer implementations; four of the most common, are shown in Figure 8.

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. Manchester encoding is self-synchronous, providing a transition at each bit interval. Figure 6 shows the encoding scheme for Standard Ethernet.

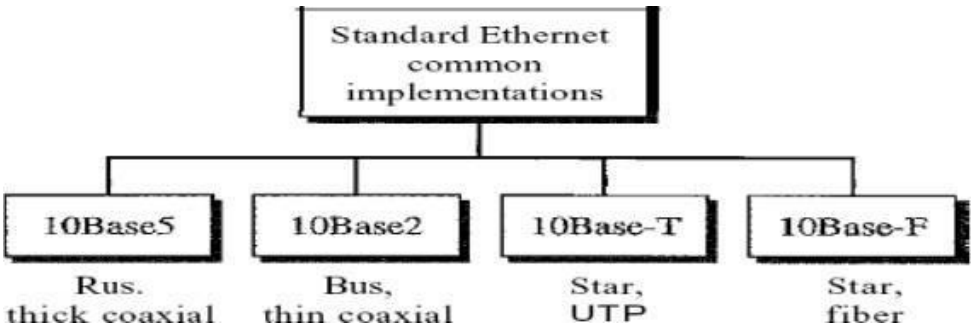


Figure 6 Categories of Standard Ethernet

3.2 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.

- MAC Sublayer
- Physical Layer

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: 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.

Physical layer:

The Physical layer in fast Ethernet is more complicated than the one in standard Ethernet.

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 centre.

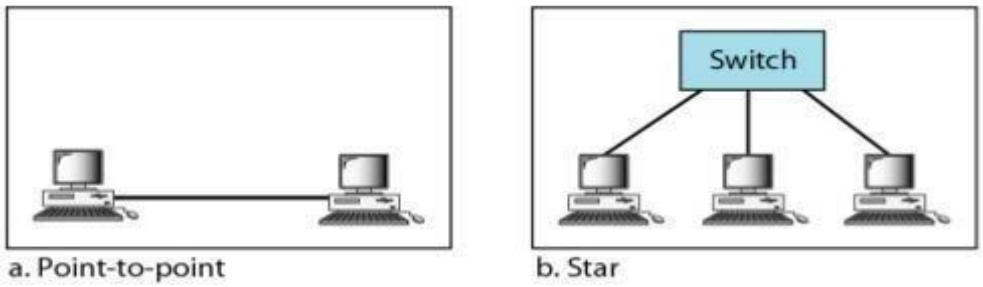


Fig. Fast Ethernet topology

Implementation:

Fast Ethernet implementation at the physical layer can be categorised as either two wire or four wire. The two wire implementation can be either category five UTP(100Base-TX) or Fibre optic cable(100Base-FX). The four wire implementation is designed only for category three UTP (100Base-T4).

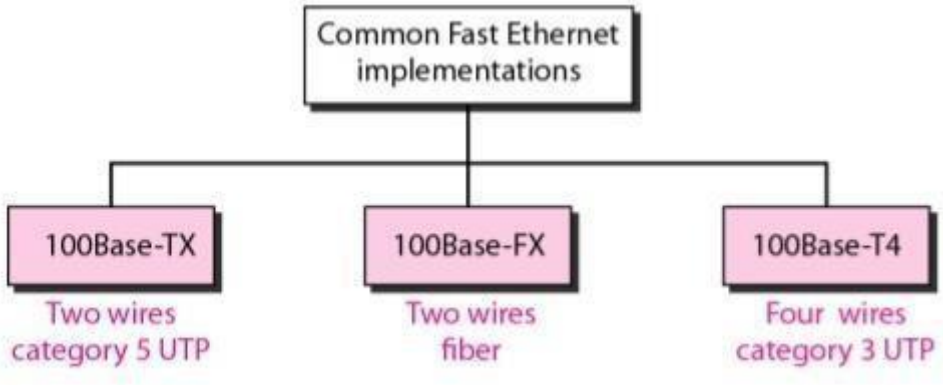
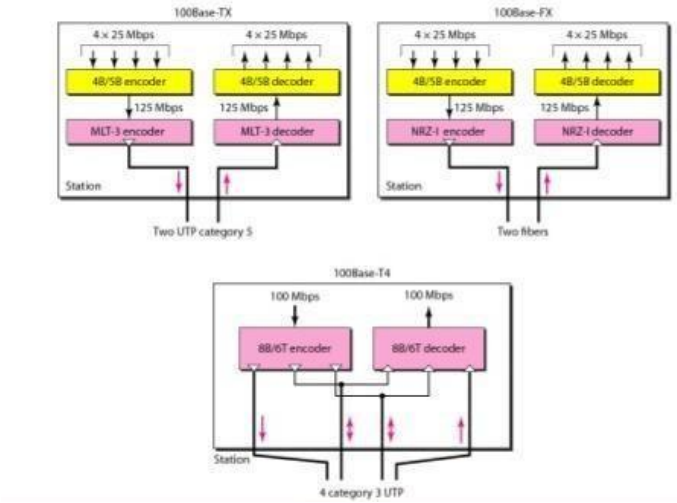


Fig. Fast Ethernet Implementations

Encoding:

Manchester encoding needs a 200-Mbaud bandwidth for a data rate of 100Mbps, 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:



100 Base-TX:

(a) 100Base-TX uses two pairs of category 5 unshielded twisted pair (UTP) or two pairs of shielded twisted pair (STP) cables to connect a station to the hub.

(b) One pair is used to carry frames from the station to the hub and the other to carry frames from the hub to the station.

(c) The distance between hub and station should be less than 100 meters.

(d) For this implementation, the MLT-3 scheme is used. However as MLT-3 is not a self synchronous line coding scheme, 4B/5B block coding is used to provide bit synchronization.

(e) This creates a data rate of 125 Mbps, which is fed into MLT-3 for encoding.

100 Base-FX*:*

(a) It users two pairs fiber-optic cables.

(b) One pair carries frame from the station to the hub and the other from hub to the station.

(c) The distance between the station and the hub (or switch) should be less than 2000 meters.

(d) It makes use of NRZ-I encoding scheme.

(e) As NRZ-I has a bit synchronization problem for long sequences, 100Base-FX uses 4B/5B block encoding that increases the bit rate from 100 to 125 mbps.

100 Base-T4:

(a) It uses four pairs of category 3 UTP.

(b) Two of the four pairs are bi-directional, the other two are unidirectional.

(c) In each direction, three pairs are used at the same time to carry data as shown in fig.

(d) Encoding/decoding in 100Base-T4 is more complicated.

(e) As this implementation uses category 3 UTP, each twisted pair cannot easily handle more than 25 Mbaud.

(f) As one pair switches between sending and receiving, three pairs of UTP category 3 can handle only 75 Mbaud (25 Mbaud each).

(g) Thus it requires an encoding scheme that converts 100 Mbps to a 75 Mbaud signal. This is done by using 8B/6T (eight binary/six ternary) encoding scheme.

Table: Summary of Fast Ethernet implementations

Characteristics	*100Base-TX*	*100Base-FX*	*100Base-T4*
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

3.3 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 it the Standard 802.3z. The goals of the Gigabit Ethernet were to upgrade the data rate to 1 Gbps, but keep the address length, the frame

format, and the maximum and minimum frame length the same. 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. Support auto negotiation as defined in Fast Ethernet.

3.3.1 MAC Sub layer

A main consideration in the evolution of Ethernet was to keep the MAC sub layer untouched. However, to achieve a data rate of 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, so we mostly ignore the half-duplex mode.

Full-Duplex Mode

In full-duplex mode, there is a central switch connected to all computers or other switches. In this mode, for each input port, each switch has buffers in which data are stored until they are transmitted. Since the switch uses the destination address of the frame and sends a frame out of the port connected to that particular destination, there is no collision. 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 1/100 shorter in Gigabit Ethernet than in 10-Mbps Ethernet, the slot time for Gigabit Ethernet is 512 bits □□1/1000 μs, which is equal to 0.512 μs. The reduced slot time means that collision is detected100 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.

3.3.2 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 one 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. 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.

Encoding

Figure 13.17 shows the encoding/decoding schemes for the four implementations. 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, 8B/10B block encoding, discussed in Chapter 4, is used. This block encoding prevents long sequences of 0s or 1s 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, as discussed in Chapter 4, 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.

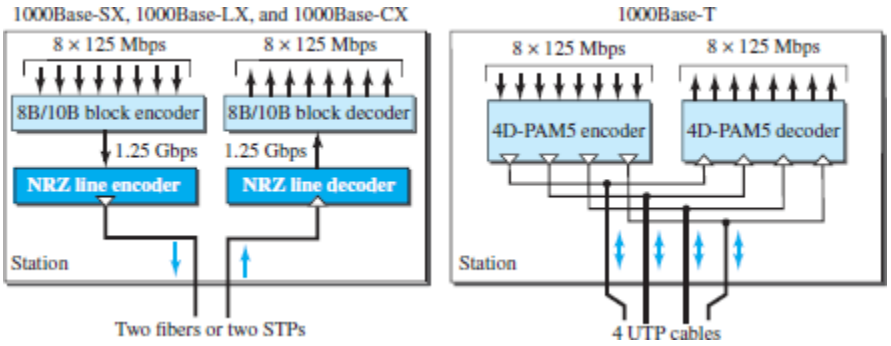


Fig. Encoding of Giga bit Ethernet

Implementation	*Medium*	*Medium Length*	*Wires*	*Encoding*
1000Base-SX	Fiber S-W	550 m	2	8B/10B + NRZ
1000Base-LX	Fiber L-W	5000 m	2	8B/10B + NRZ
1000Base-CX	STP	25 m	2	8B/10B + NRZ
1000Base-T4	UTP	100 m	4	4D-PAM5

Fig. Summary Giga bit Ethernet

3.4 Wireless LANs

Wireless communication is one of the fastest growing technologies. The demand for connecting devices without the use of cables has been increasing everywhere. Here we will discuss about two promising wireless technologies namely wireless LANs also refereed as IEEE802.11 or wireless Ethernet and Bluetooth.

3.4.1 IEEE 802.11

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

Architecture of 802.11

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). Figure 9 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.

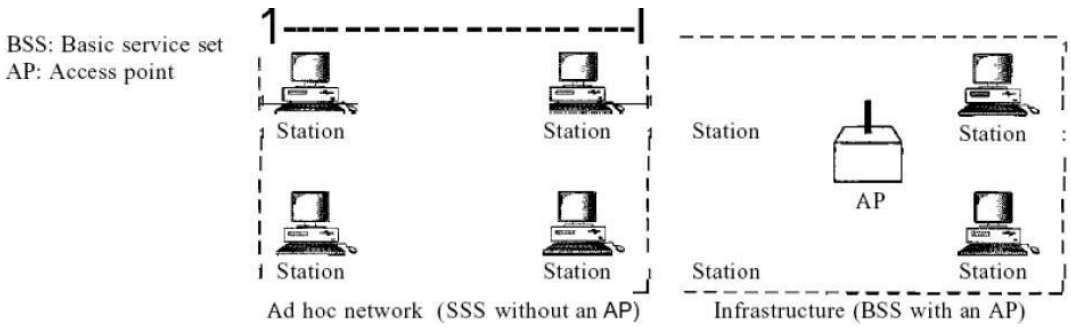


Figure.7 *Basic service sets (BSSs)*

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. Figure 10 shows an ESS.

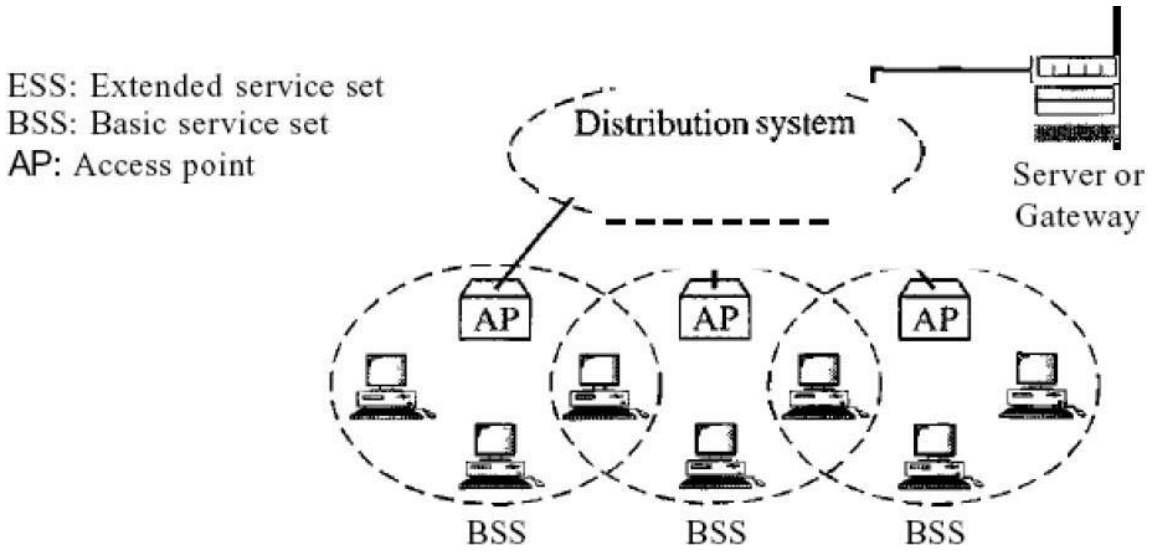
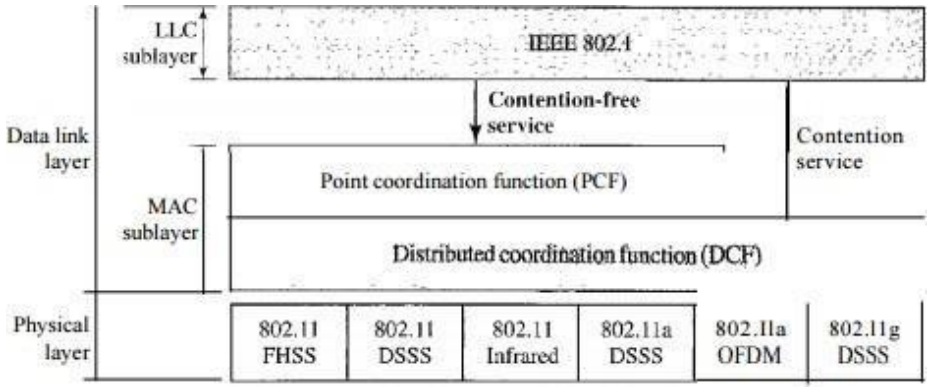


Figure 8 ***Extended service sets (ESSs)***

3.4.2 MAC SUBLAYER

IEEE 802.11 defines two MAC sublayers: the distributed coordination function (DCF) and point coordination function (PCF). Below figure shows the relationship between the two MAC sublayers, the LLC sublayer, and the physical layer. We discuss the physical layer implementations later in the chapter and will now concentrate on the MAC sublayer.



Distributed Coordination Function

One of the two protocols defined by IEEE at the MAC sublayer is called the distributed coordination function (DCF). DCF uses CSMAICA as the access method. Wireless LANs cannot implement CSMAfCD for three reasons:

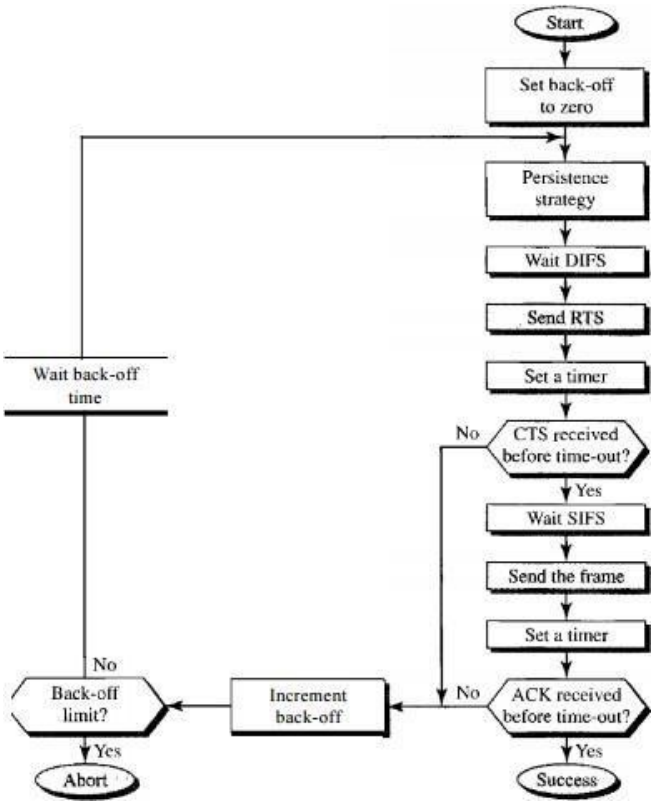
I. For collision detection a station must be able to send data and receive collision signals at the same time. This can mean costly stations and increased bandwidth requirements.

2. Collision may not be detected because of the hidden station problem. We will discuss this problem later in the chapter.

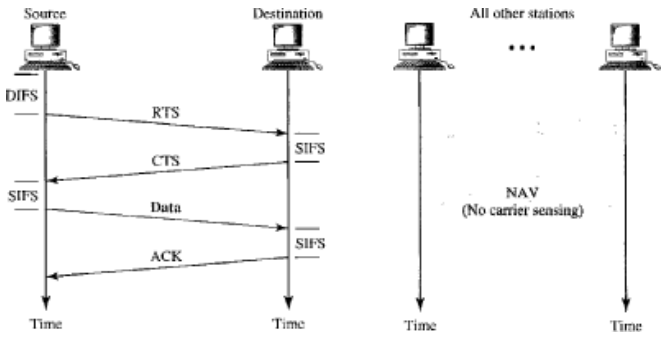
3. The distance between stations can be great. Signal fading could prevent a station at one end from hearing a collision at the other end.

Process Flowchart is given in below figure which shows the process flowchart for CSMAICA as used in wireless LANs. We will explain the steps shortly.

.



Frame Exchange Time Line Below Figure shows the exchange of data and control frames in time



I. Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency.

a. The channel uses a persistence strategy with back-off until the channel is idle.

b. After the station is found to be idle, the station waits for a period of time called the distributed interframe space (DIFS); then the station sends a control frame called the request to send (RTS).

2. After receiving the RTS and waiting a period of time called the short interframe space (SIFS), the destination station sends a control frame, called the clear to send (CTS), to the source station. This control frame indicates that the destination station is ready to receive data.

3. The source station sends data after waiting an amount of time equal to SIFS.

4. The destination station, after waiting an amount of time equal to SIFS, sends an acknowledgment to show that the frame has been received. Acknowledgment is needed in this protocol because the station does not have any means to check for the successful arrival of its data at the destination. On the other hand, the lack of collision in CSMA/CD is a kind of indication to the source that data have arrived.

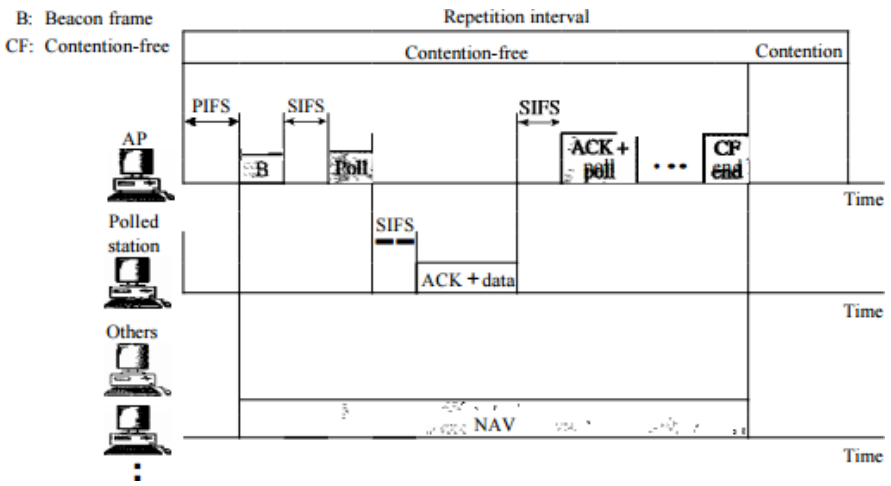
Network Allocation Vector How do other stations defer sending their data if one station acquires access? In other words, how is the collision avoidance aspect of this protocol accomplished? The key is a feature called NAV. When a station sends an RTS frame, it includes the duration of time that it needs to occupy the channel. The stations that are affected by this transmission create a timer called a network allocation vector (NAV) that

shows how much time must pass before these stations are allowed to check the channel for idleness. Each time a station accesses the system and sends an RTS frame, other stations start their NAV. In other words, each station, before sensing the physical medium to see if it is idle, first checks its NAV to see if it has expired. Above figure shows the idea of NAV.

Collision During Handshaking What happens if there is collision during the time when RTS or CTS control frames are in transition, often called the handshaking period? Two or more stations may try to send RTS frames at the same time. These control frames may collide. However, because there is no mechanism for collision detection, the sender assumes there has been a collision if it has not received a CTS frame from the receiver. The back-off strategy is employed, and the sender tries again.

Point Coordination Function (PCP) The point coordination function (PCF) is an optional access method that can be implemented in an infrastructure network (not in an ad hoc network). It is implemented on top of the DCF and is used mostly for time-sensitive transmission. PCF has a centralized, contention-free polling access method. The AP performs polling for stations that are capable of being polled. The stations are polled one after another, sending any data they have to the AP. To give priority to PCF over DCF, another set of interframe spaces has been defined: PIFS and SIFS. The SIFS is the same as that in DCF, but the PIFS (PCF IFS) is shorter than the DIFS. This means that if, at the same time, a station wants to use only DCF and an AP wants to use PCF, the AP has priority.

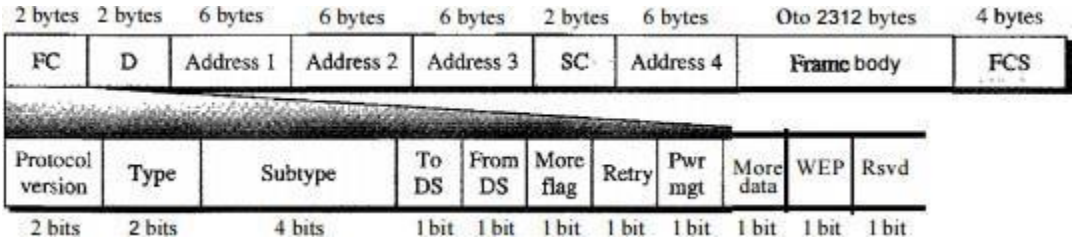
Due to the priority of PCF over DCF, stations that only use DCF may not gain access to the medium. To prevent this, a repetition interval has been designed to cover both contention-free (PCF) and contention-based (DCF) traffic. The repetition interval, which is repeated continuously, starts with a special control frame, called a beacon frame. When the stations hear the beacon frame, they start their NAV for the duration of the contention-free period of the repetition interval. Below figure 14.6 shows an example of a repetition interval.



During the repetition interval, the PC (point controller) can send a poll frame, receive data, send an ACK, receive an ACK, or do any combination of these (802.11 uses piggybacking). At the end of the contention-free period, the PC sends a CF end (contention-free end) frame to allow the contention-based stations to use the medium.

Fragmentation The wireless environment is very noisy; a corrupt frame has to be retransmitted. The protocol, therefore, recommends fragmentation-the division of a large frame into smaller ones. It is more efficient to resend a small frame than a large one.

Frame Format The MAC layer frame consists of nine fields, as shown in below figure.



Frame control (FC). The FC field is 2 bytes long and defines the type of frame and some control information. Table below describes the subfields of frame control field.

Field	*Explanation*
Version	Current version is 0
Type	Type of information: management (00), control (01), or data (10)
Subtype	Subtype of each type (see Table 14.2)
ToDS	Defined later
FromDS	Defined later
More flag	When set to 1, means more fragments
Retry	When set to 1, means retransmitted frame
Pwr mgt	When set to 1, means station is in power management mode
More data	When set to 1, means station has more data to send
WEP	Wired equivalent privacy (encryption implemented)
Rsvd	Reserved

D. In all frame types except one, this field defines the duration of the transmission that is used to set the value of NAY. In one control frame, this field defines the ID of the frame.

Addresses. There are four address fields, each 6 bytes long. The meaning of each address field depends on the value of the To DS and From DS subfields and will be discussed later.

Sequence control. This field defines the sequence number of the frame to be used in flow control.

Frame body. This field, which can be between 0 and 2312 bytes, contains information based on the type and the subtype defined in the FC field.

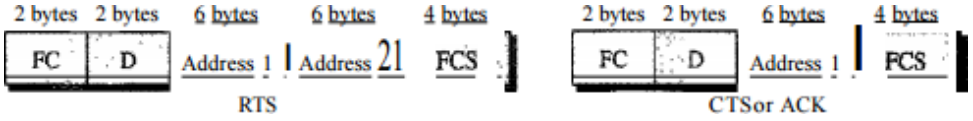
FCS. The FCS field is 4 bytes long and contains a CRC-32 error detection sequence.

Frame Types

A wireless LAN defined by IEEE 802.11 has three categories of frames: management frames, control frames, and data frames.

Management Frames Management frames are used for the initial communication between stations and access points.

Control Frames Control frames are used for accessing the channel and acknowledging frames. Below figure shows the format.



For control frames the value of the type field is 0 I; the values of the subtype fields for frames we have discussed are shown in below table

Subtype	*Meaning*
1011	Request to send (RTS)
1100	Clear to send (CTS)
1101	Acknowledgment (ACK)

Data Frames Data frames are used for carrying data and control information.

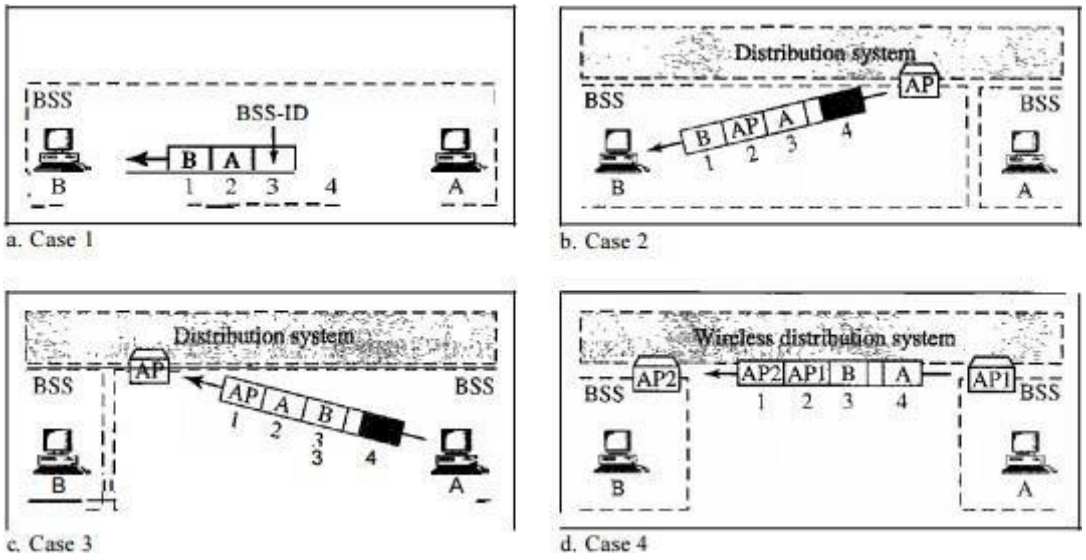
3.4.3 ADDRESSING MECHANISM

The IEEE 802.11 addressing mechanism specifies four cases, defined by the value of the two flags in the FC field, To DS and From DS. Each flag can be either 0 or I, resulting in four different situations. The interpretation of the four addresses (address I to address 4) in the MAC frame depends on the value of these flags, as shown in below table.

To DS	*From* DS	*Address* 1	*Address* 2	*Address* 3	*Address* 4
0	0	Destination	Source	BSS ID	N/A
0	1	Destination	SendingAP	Source	N/A
1	0	Receiving AP	Source	Destination	N/A
1	1	Receiving AP	SendingAP	Destination	Source

Note that address 1 is always the address of the next device. Address 2 is always the address of the previous device. Address 3 is the address of the final destination station if it is not defined by address I. Address 4 is the address of the original source station if it is not the same as address 2.

Case 1: 00 In this case, To DS = 0 and From DS = O. This means that the frame is not going to a distribution system (To DS = 0) and is not coming from a distribution. system (From DS =0). The frame is going from one station in a BSS to another without passing through the distribution system. The ACK frame should be sent to the original sender. The addresses are shown in below figure.



Case 2: 01 In this case, To DS = 0 and From DS = 1. This means that the frame is coming from a distribution system (From DS = 1). The frame is coming from an AP and going to a station. The ACK should be sent to the AP. The addresses are as shown in Figure 14.9. Note that address 3 contains the original sender of the frame (in another BSS).

Case 3: 10 In this case, To DS =1 and From DS = O. This means that the frame is going to a distribution system (To DS = 1). The frame is going from a station to an AP. The ACK is sent to the original station. The addresses are as shown in above figure Note that address 3 contains the final destination of the frame (in another BSS).

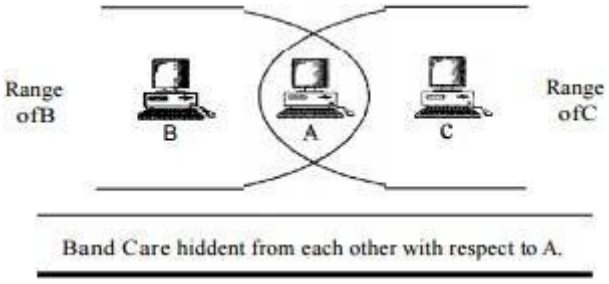
Case 4:11 In this case, To DS =1 and From DS =1. Thus is the case in which the distribution system is also wireless. The frame is going from one AP to another AP in a wireless distribution system. We do not need to define addresses if the distribution system is a wired LAN because the frame in these cases has the format of a wired LAN frame (Ethernet, for example). Here, we need four addresses to define the original sender, the final destination, and two intermediate APs. Above figure shows the situation.

Hidden and Exposed Station Problems

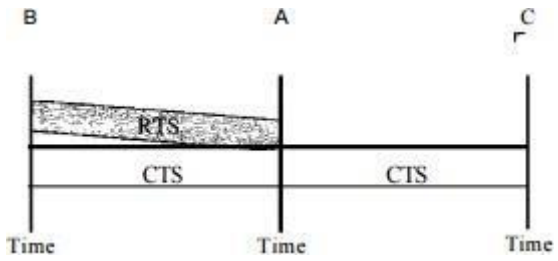
We referred to hidden and exposed station problems in the previous section. It is time now to dicuss these problems and their effects. Hidden Station Problem Figure 14.10 shows an example of the hidden station problem. Station B has a transmission range shown by the left oval (sphere in space); every station in this range can hear any signal transmitted by station B. **Station C** has a transmission range shown by the right oval (sphere in space); every

station located in this range can hear any signal transmitted by C. Station C is outside the transmission range of B; likewise, station B is outside the transmission range of C. Station A, however, is in the area covered by both Band C; it can hear any signal transmitted by B or C.

Assume that station B is sending data to station A. In the middle of this transmission, station C also has data to send to station A. However, station C is out of B's range and transmissions from B cannot reach C. Therefore C thinks the medium is free. Station C sends its data to A, which results in a collision at A because this station is receiving data from both B and C. In this case, we say that stations Band C are hidden from each other with respect to A. Hidden stations can reduce the capacity of the network because of the possibility of collision.



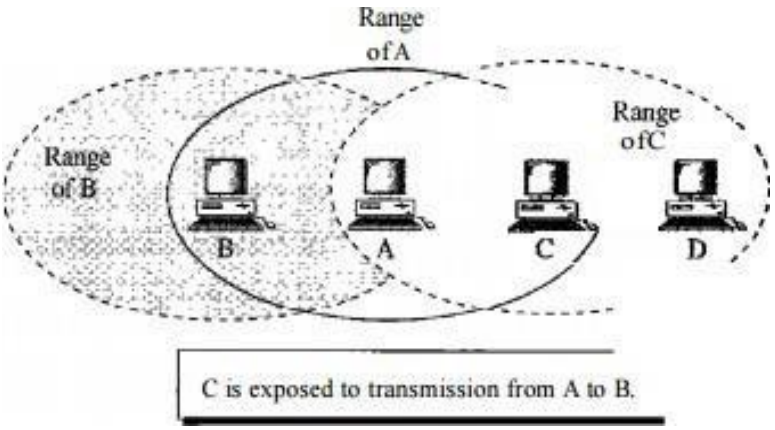
The solution to the hidden station problem is the use of the handshake frames (RTS and CTS) that we discussed earlier. Figure below shows that the RTS message from B reaches A, but not C. However, because both Band C are within the range of A, the CTS message, which contains the duration of data transmission from B to A reaches C. Station C knows that some hidden station is using the channel and refrains from transmitting until that duration is over.



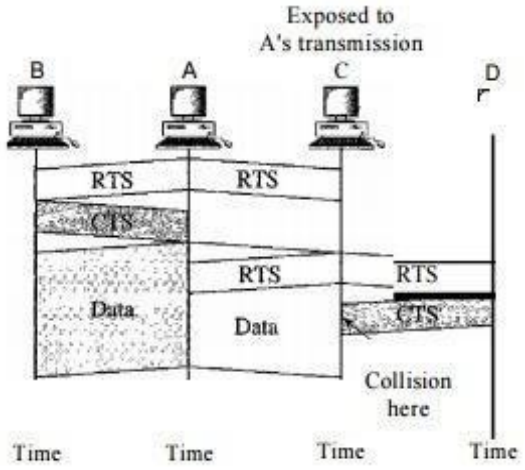
Exposed Station Problem

Now consider a situation that is the inverse of the previous one: the exposed station problem. In this problem a station refrains from using a channel when it is, in fact, available. In below figure, station A is transmitting to station B. Station C has some data to send to station D,

which can be sent without interfering with the transmission from A to B. However, station C is exposed to transmission from A; it hears what A is sending and thus refrains from sending. In other words, C is too conservative and wastes the capacity of the channel.



The handshaking messages RTS and CTS cannot help in this case, despite what you might think. Station C hears the RTS from A, but does not hear the CTS from B. Station C, after hearing the RTS from A, can wait for a time so that the CTS from B reaches A; it then sends an RTS to D to show that it needs to communicate with D. Both stations B and A may hear this RTS, but station A is in the sending state, not the receiving state. Station B, however, responds with a CTS. The problem is here. If station A has started sending its data, station C cannot hear the CTS from station D because of the collision; it cannot send its data to D. It remains exposed until A finishes sending its data as in below figure.

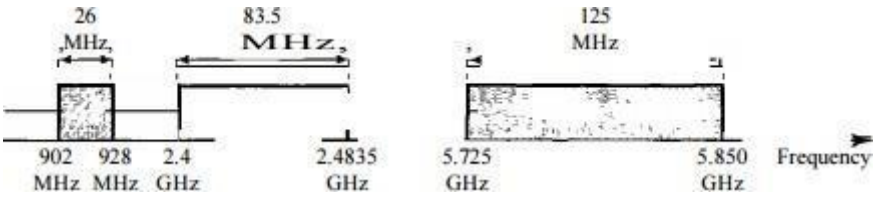


3.4.4 PHYSICAL LAYER

Here we discuss six specifications, as shown in Table below.

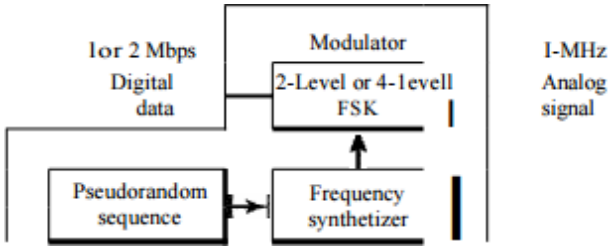
IEEE	*Technique*	*Band*	*Modulation*	*Rate (Mbps)*
802.11	FHSS	2.4 GHz	FSK	1 and 2
	DSSS	2.4 GHz	PSK	1 and 2
		Infrared	PPM	1 and 2
802.11a	OFDM	5.725 GHz	PSKorQAM	6 to 54
802.11b	DSSS	2.4 GHz	PSK	5.5 and 11
802.1Ig	OFDM	2.4 GHz	Different	22 and 54

All implementations, except the infrared, operate in the industrial, scientific, and medical (ISM) band, which defines three unlicensed bands in the three ranges 902-928 MHz, 2.400--4.835 GHz, and 5.725-5.850 GHz, as shown in below figure.



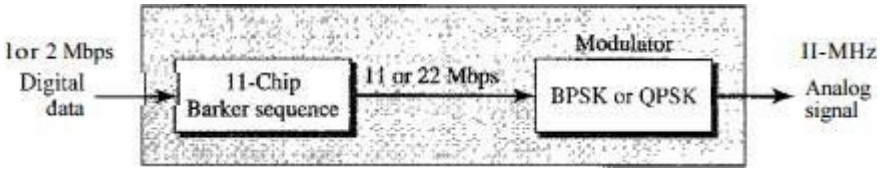
IEEE 802.11 FHSS

IEEE 802.11 FHSS uses the frequency-hopping spread spectrum (FHSS) method as discussed in Chapter 6. FHSS uses the 2.4-GHz ISM band. The band is divided into 79 sub bands of 1 MHz (and some guard bands). A pseudorandom number generator selects the hopping sequence. The modulation technique in this specification is either two-level FSK or four-level FSK with I or 2 bits/baud, which results in a data rate of 1 or 2 Mbps, as shown in below figure.



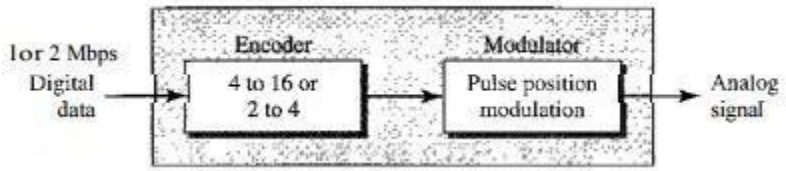
IEEE 802.11 DSSS

IEEE 802.11 DSSS uses the direct sequence spread spectrum (DSSS) method as discussed in Chapter 6. DSSS uses the 2.4-GHz ISM band. The modulation technique in this specification is PSK at 1Mbaud/s. The system allows 1 or 2 bits/baud (BPSK or QPSK), which results in a data rate of 1 or 2 Mbps, as shown in below figure.



IEEE 802.11 Infrared

IEEE 802.11 infrared uses infrared light in the range of 800 to 950 nm. The modulation technique is called pulse position modulation (PPM). For a I-Mbps data rate, a 4-bit sequence is first mapped into a 16-bit sequence in which only one bit is set to 1 and the rest are set to O. For a 2-Mbps data rate, a 2-bit sequence is first mapped into a 4-bit sequence in which only one bit is set to 1 and the rest are set to O. The mapped sequences are then converted to optical signals; the presence of light specifies 1, the absence of light specifies O. See below figure.

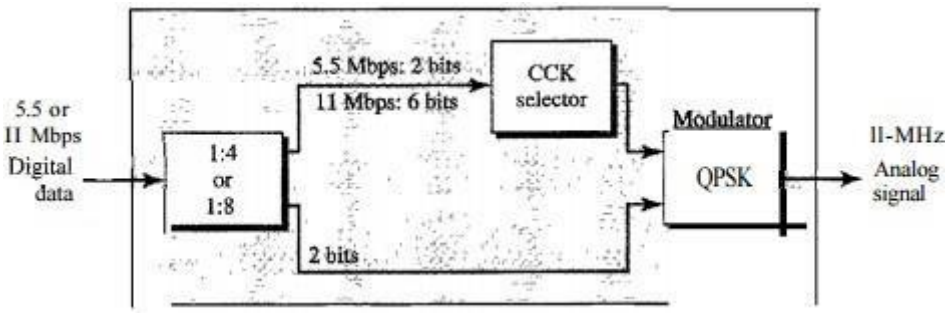


IEEE 802.lla OFDM

IEEE 802.Ila OFDM describes the orthogonal frequency-division multiplexing (OFDM) method for signal generation in a 5-GHz ISM band. OFDM is similar to FDM, with one major difference: All the subbands are used by one source at a given time. Sources contend with one another at the data link layer for access. The band is divided into 52 subbands, with 48 subbands for sending 48 groups of bits at a time and 4 subbands for control information. Dividing the band into subbands diminishes the effects of interference. Ifthe subbands are used randomly, security can also be increased. OFDM uses PSK and QAM for modulation. The common data rates are 18 Mbps (PSK) and 54 Mbps (QAM).

IEEE 802.llb DSSS

IEEE 802.11 b DSSS describes the high-rate direct sequence spread spectrum (HRDSSS) method for signal generation in the 2.4-GHz ISM band. HR-DSSS is similar to DSSS except for the encoding method, which is called complementary code keying (CCK). CCK encodes 4 or 8 bits to one CCK symbol. To be backward compatible with DSSS, HR-DSSS defines four data rates: 1,2, 5.5, and 11 Mbps. The first two use the same modulation techniques as DSSS. The 5.5-Mbps version uses BPSK and transmits at 1.375 M bauds with 4-bit CCK encoding. The II-Mbps version uses QPSK and transmits at 1.375 Mbps with 8-bit CCK encoding. Below figure shows the modulation technique for this standard.



IEEE 802.11g

This new specification defines forward error correction and OFDM using the 2.4-GHz ISM band. The modulation technique achieves a 22- or 54-Mbps data rate. It is backward compatible with 802.11b, but the modulation technique is OFDM.

3.5 Bluetooth:

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.

3.5.1 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;t the rest are called secondaries. 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. Figure 1 shows a piconet.

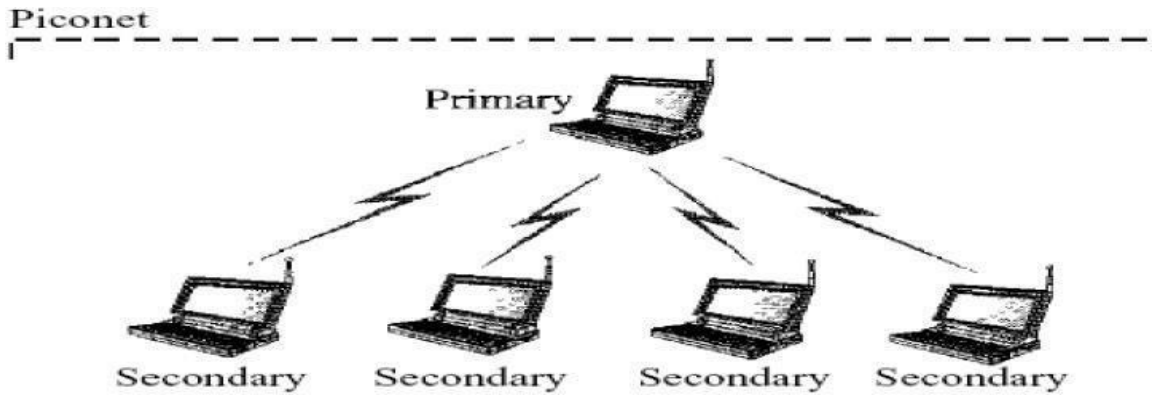


Figure 9 *Piconet*

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.

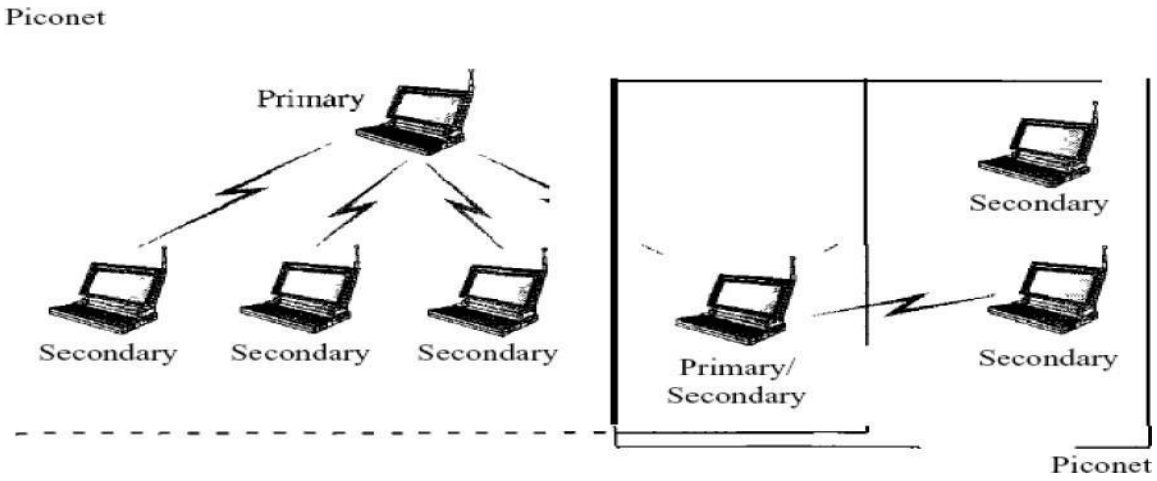


Figure 10 illustrates a scatternet.

Bluetooth Devices

A Bluetooth device has a built-in short-range radio transmitter. The current data rate is 1 Mbps with a 2.4-GHz bandwidth. This means that there is a possibility of interference between the IEEE 802.11b wireless LANs and Bluetooth LANs.

3.5.2 Bluetooth Layers

Bluetooth uses several layers that do not exactly match those of the Internet model we have defined in this book. Figure 3 shows these layers.

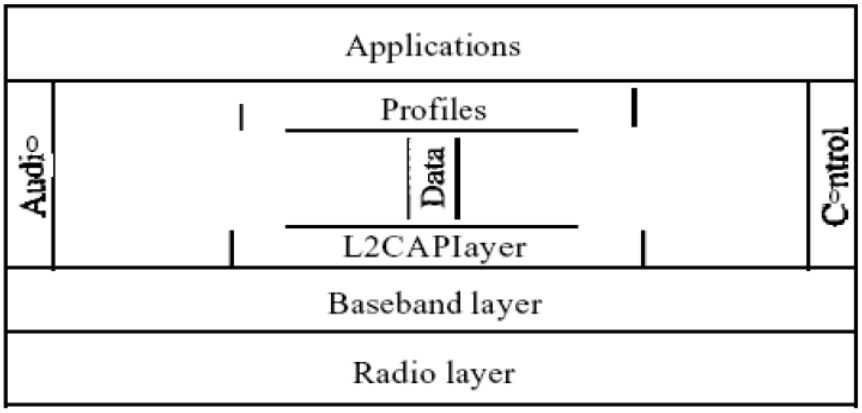


Figure 11 *Bluetooth layers*

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.

Band

Bluetooth uses a 2.4-GHz ISM band divided into 79 channels of 1 MHz each.

FHSS

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. A device uses a frequency for only 625 Ils (1/1600 s) before it hops to another frequency; the dwell time is 625 *Ils.*

Modulation

To transform bits to a signal, Bluetooth uses a sophisticated version of FSK, called GFSK (FSK with Gaussian bandwidth filtering; a discussion of this topic is beyond the scope of this book). GFSK has a can'ier frequency. Bit 1 is represented by a frequency deviation above the carrier; bit ais represented by a frequency deviation below the carrier. The frequencies, in megahertz, are defined according to the following formula for each channel:
fc=2402+n n =0, 1,2,3, ..., 78

For example, the first channel uses carrier frequency 2402 MHz (2.402 GHz), and the second channel uses carrier frequency 2403 MHz (2.403 GHz).

Baseband Layer

The baseband layer is roughly equivalent to the MAC sub layer 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 µs. 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.

TDMA

Bluetooth uses a form of TDMA (see Chapter 12) that is called TDD-TDMA (timedivision duplex TDMA). TDD-TDMA is a kind of half-duplex communication in which the secondary and receiver send and receive data, but not at the same time (halfduplex); however, the communication for each direction uses different hops. This is similar to walkie-talkies using different carrier frequencies.

Single-Secondary Communication If the piconet has only one secondary, the TDMA operation is very simple. The time is divided into slots of 625 Ils. The primary uses evennumbered slots (0, 2, 4, ...); the secondary uses odd-numbered slots (1, 3, 5, ...). TDD-TDMA allows the primary and the secondary to communicate in half-duplex mode.
In slot 0, the primary sends, and the secondary receives; in slot 1, the secondary sends, and the primary receives. The cycle is repeated. Figure 12 shows the concept.

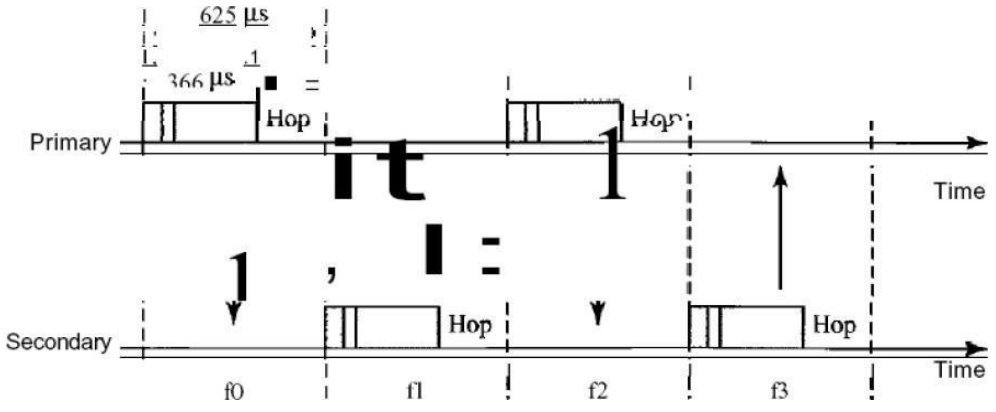
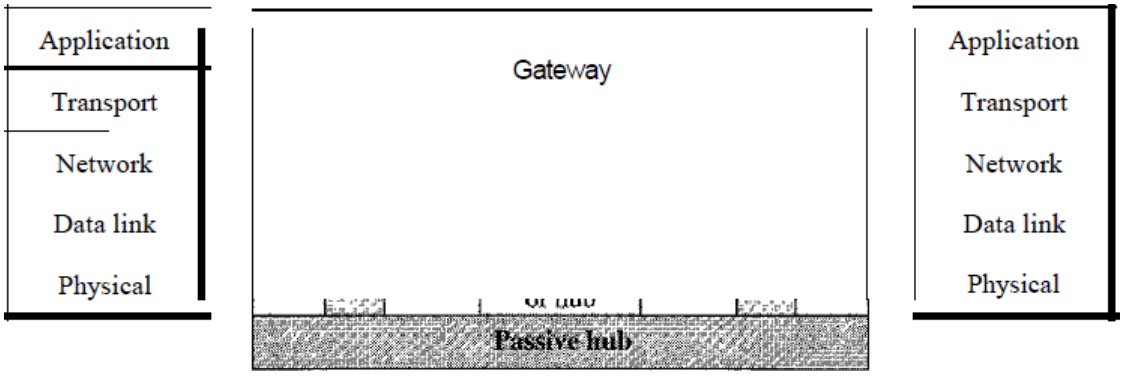


Figure 12 ***Single-secondary communication***

3.6 CONNECTING DEVICES

In this section, we divide **connecting devices** into five different categories based on the layer **in** which they operate **in** a network, as shown **in following f**igure.



The five categories contain devices which can be defined as

1. Those which operate below the physical layer such as a passive hub.
2. Those which operate at the physical layer (a repeater or an active hub).
3. Those which operate at the physical and data link layers (a bridge or a two-layer switch).
4. Those which operate at the physical, data link, and network layers (a router or a three-layer switch).
5. Those which can operate at all five layers (a gateway).

3.6.1 Passive Hubs

A passive hub is just a connector. It connects the wires coming from different branches. In a star-topology Ethernet LAN, a passive hub is just a point where the signals coming from different stations collide; the hub is the collision point. This type of a hub is part of the media; its location in the Internet model is below the physical layer.

3.6.2 Repeaters

A repeater is a device that operates only in the physical layer. Signals that carry information within a network can travel a fixed distance before attenuation endangers the integrity of the data. A repeater receives a signal and, before it becomes too weak or corrupted, regenerates

the original bit pattern. The repeater then sends the refreshed signal. A repeater can extend the physical length of a LAN, as shown in Figure 15.2.

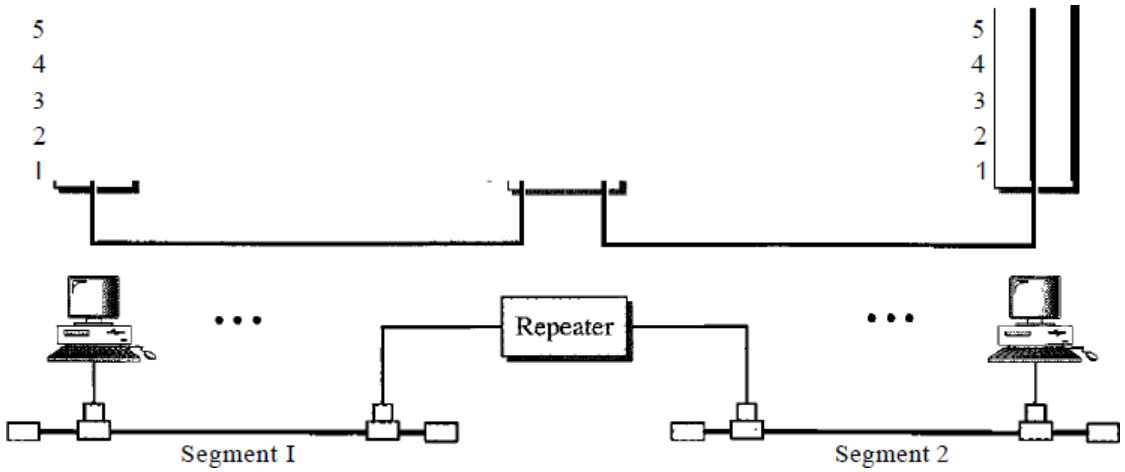


Figure *A repeater connecting two segments of a LAN*

A repeater does not actually connect two LANs; it connects two segments of the same LAN. The segments connected are still part of one single LAN. A repeater is not a device that can connect two LANs of different protocols.

A repeater can overcome the 10Base5 Ethernet length restriction. In this standard, the length of the cable is limited to 500 m. To extend this length, we divide the cable into segments and install repeaters between segments. Note that the whole network is still considered one LAN, but the portions of the network separated by repeaters are called segments. The repeater acts as a two-port node, but operates only in the physical layer. When it receives a frame from any of the ports, it regenerates and forwards it to the other port.

It is tempting to compare a repeater to an amplifier, but the comparison is inaccurate. An amplifier cannot discriminate between the intended signal and noise; it amplifies equally everything fed into it. A repeater does not amplify the signal; it regenerates the signal. When it receives a weakened or corrupted signal, it creates a copy, bit for bit, at the original strength.

The location of a repeater on a link is vital. A repeater must be placed so that a signal reaches it before any noise changes the meaning of any of its bits. A little noise can alter the precision of a bit's voltage without destroying its identity. If the corrupted bit travels much farther, however, accumulated noise can change its meaning completely. At that point, the original voltage is not recoverable, and the error needs to be corrected. A repeater placed on

the line before the legibility of the signal becomes lost can still read the signal well enough to determine the intended voltages and replicate them in their original form.

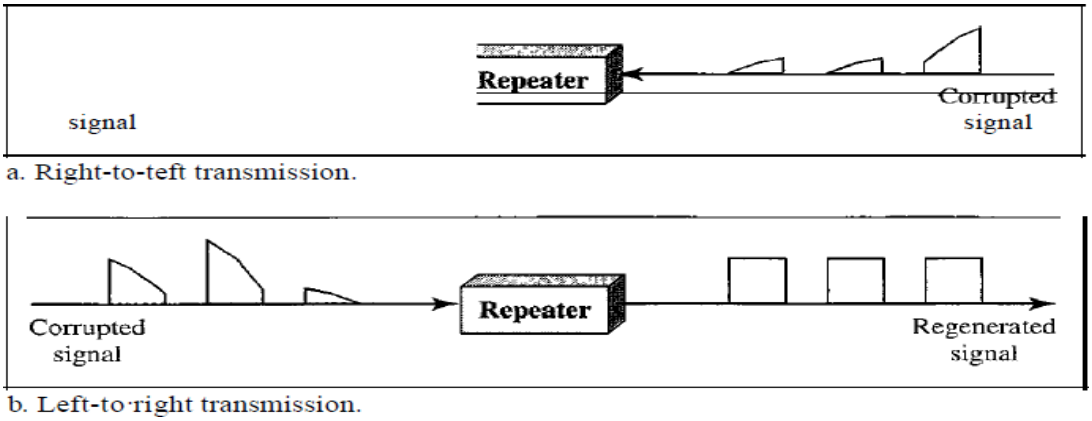


Figure *Function of a repeater*

3.6.3 Active Hubs

An active hub is actually a multipart repeater. It is normally used to create connections between stations in a physical star topology. We have seen examples of hubs in some Ethernet implementations (lOBase-T, for example). However, hubs can also be used to create multiple levels of hierarchy, as shown in below figure. The hierarchical use of hubs removes the length limitation of 10Base-T (100 m).

3.6.4 Bridges

A bridge operates in both the physical and the data link layer. As a physical layer device, it regenerates the signal it receives. As a data link layer device, the bridge can check the physical (MAC) addresses (source and destination) contained in the frame.

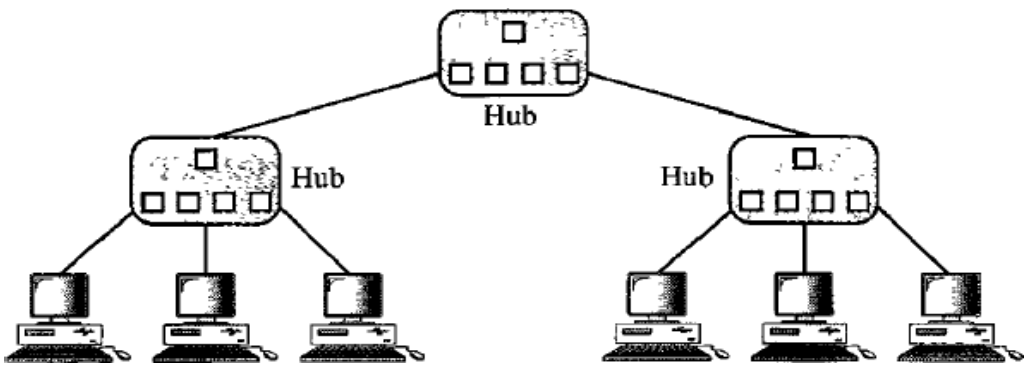


Figure: *A hierarchy of hubs*

Filtering

One may ask, What is the difference in functionality between a bridge and a repeater? A bridge has filtering capability. It can check the destination address of a frame and decide if the frame should be forwarded or dropped. If the frame is to be forwarded, the decision must specify the port. A bridge has a table that maps addresses to ports.

Let us give an example. In below figure, two LANs are connected by a bridge. If a frame destined for station 712B13456142 arrives at port 1, the bridge consults its table to find the departing port. According to its table, frames for 7l2B13456142 leave through port 1; therefore, there is no need for forwarding, and the frame is dropped. On the other hand, if a frame for 712B13456141 arrives at port 2, the departing port is port 1 and the frame is forwarded. In the first case, LAN 2 remains free of traffic; in the second case, both LANs have traffic. In our example, we show a two-port bridge; in reality a bridge usually has more ports. Note also that a bridge does not change the physical addresses contained in the frame.

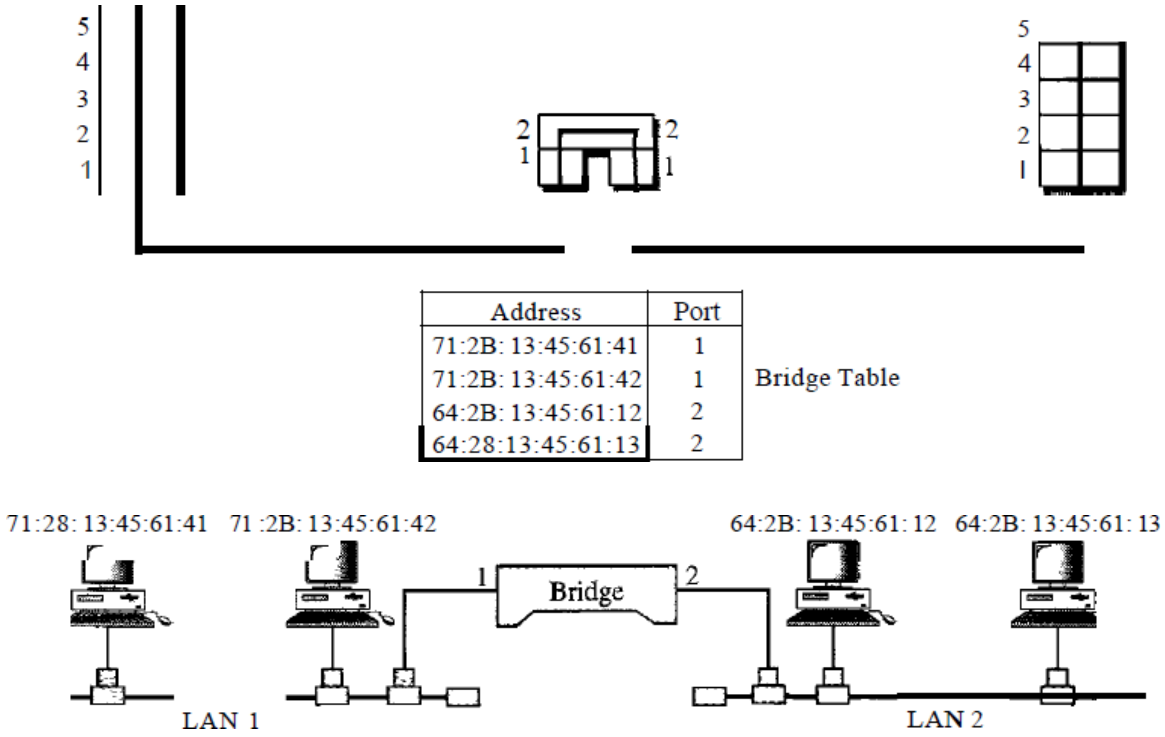


Figure. *A bridge connecting two LANs*

Transparent Bridges

A transparent bridge is a bridge in which the stations are completely unaware of the bridge's existence. If a bridge is added or deleted from the system, reconfiguration of the stations is unnecessary. According to the IEEE 802.1 d specification, a system equipped with transparent bridges must meet three criteria:

I. Frames must be forwarded from one station to another.

2. The forwarding table is automatically made by learning frame movements in the network.

3. Loops in the system must be prevented.

Forwarding A transparent bridge must correctly forward the frames, as discussed in the previous section.

Learning The earliest bridges had forwarding tables that were static. The systems administrator would manually enter each table entry during bridge setup. Although the process was simple, it was not practical. If a station was added or deleted, the table had to be modified manually. The same was true if a station's MAC address changed, which is not a rare event. For example, putting in a new network card means a new MAC address.

A better solution to the static table is a dynamic table that maps addresses to ports automatically. To make a table dynamic, we need a bridge that gradually learns from the frame movements. To do this, the bridge inspects both the destination and the source addresses. The destination address is used for the forwarding decision the source address is used for adding entries to the table and for updating purposes.

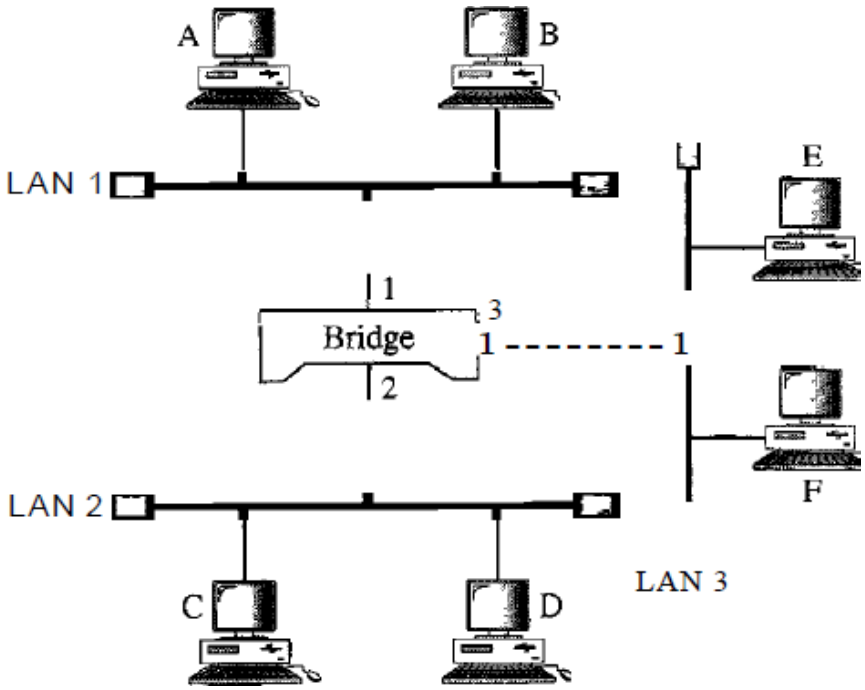
Let us elaborate on this process by using below figure.

1. When station A sends a frame to station D, the bridge does not have an entry for either D or A. The frame goes out from all three ports; the frame floods the network. However, by looking at the source address, the bridge learns that station A must be located on the LAN connected to port 1. This means that frames destined for A, in the future, must be sent out through port 1. The bridge adds this entry to its table. The table has its first entry now.

2. When station E sends a frame to station A, the bridge has an entry for A, so it forwards the frame only to port 1. There is no flooding. In addition, it uses the source address of the frame, E, to add a second entry to the table.

3. When station B sends a frame to C, the bridge has no entry for C, so once again it floods the network and adds one more entry to the table.

4. The process of learning continues as the bridge forwards frames.



Address	Port

a. Original

Address	Port
A	1

b. After A sends a frame to D

Address	Port
A	1
E	3

c. After E sends a frame to A

Address	Port
A	1
E	3
B	1

d. After B sends a frame to C

Figure. A learning bridge and the process of learning

Loop Problem Transparent bridges work fine as long as there are no redundant bridges in the system. Systems administrators, however, like to have redundant bridges (more than one bridge between a pair of LANs) to make the system more reliable. If a bridge fails, another bridge takes over until the failed one is repaired or replaced. Redundancy can create loops in the system, which is very undesirable. Figure below shows a very simple example of a loop created in a system with two LANs connected by two bridges.

1. Station A sends a frame to station D. The tables of both bridges are empty. Both forward the frame and update their tables based on the source address A.

2. Now there are two copies of the frame on LAN 2. The copy sent out by bridge 1 is received by bridge 2, which does not have any information about the destination

address D; it floods the bridge. The copy sent out by bridge 2 is received by bridge 1 and is sent out for lack of information about D. Note that each frame is handled separately because bridges, as two nodes on a network sharing the medium, use an access method such as CSMA/CD. The tables of both bridges are updated, but still there is no information for destination D.

3. Now there are two copies of the frame on LAN 1. Step 2 is repeated, and both copies flood the network.

4. The process continues on and on. Note that bridges are also repeaters and regenerate frames. So in each iteration, there are newly generated fresh copies of the frames.

To solve the looping problem, the IEEE specification requires that bridges use the spanning tree algorithm to create a loop less topology.

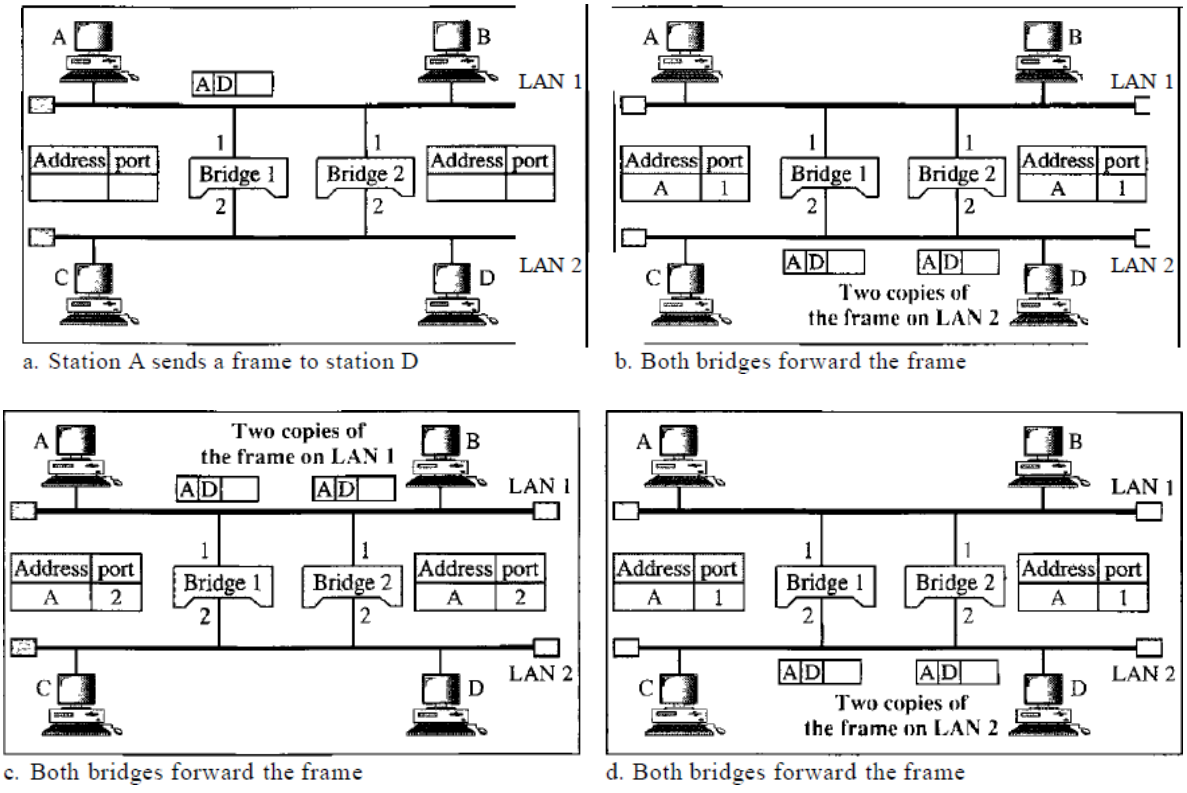


Figure. Loop problem in a learning bridge

3.7 BACKBONE NETWORKS

Some connecting devices discussed in this chapter can be used to connect LANs in a backbone network. A backbone network allows several LANs to be connected. In a backbone network, no station is directly connected to the backbone; the stations are part of a LAN, and the backbone connects the LANs. The backbone is itself a LAN that uses a LAN protocol such as Ethernet; each connection to the backbone is itself another LAN. Although many different architectures can be used for a backbone, we discuss only the two most common: the bus and the star.

3.7.1 Bus Backbone

In a bus backbone, the topology of the backbone is a bus. The backbone itself can use one of the protocols that support a bus topology such as lOBase5 or lOBase2. Bus backbones are normally used as a distribution backbone to connect different buildings in an organization. Each building can comprise either a single LAN or another backbone (normally a star backbone). A good example of a bus backbone is one that connects single- or multiple-floor buildings on a campus. Each single-floor building usually has a single LAN. Each multiple-floor building has a backbone (usually a star) that connects each LAN on a floor. A bus backbone can interconnect these LANs and backbones. Below Figure shows an example of a bridge-based backbone with four LANs.

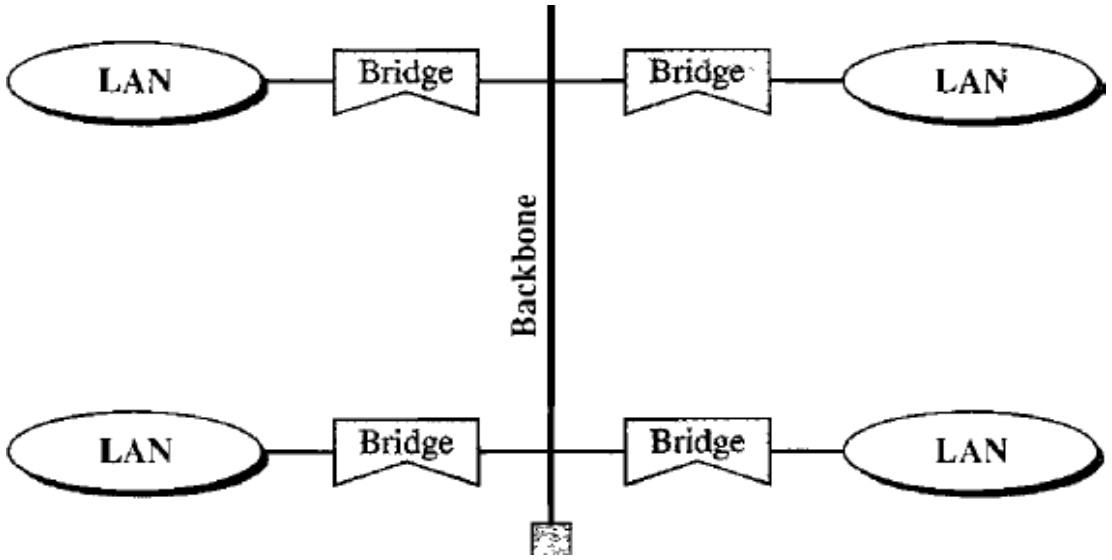


Figure. Bus backbone

In above figure, if a station in a LAN needs to send a frame to another station in the same LAN, the corresponding bridge blocks the frame; the frame never reaches the backbone. However, if a station needs to send a frame to a station in another LAN, the bridge passes the frame to the backbone, which is received by the appropriate bridge and is delivered to the destination LAN. Each bridge connected to the backbone has a table that shows the stations on the LAN side of the bridge. The blocking or delivery of a frame is based on the contents of this table.

3.7.2 Star Backbone

In a star backbone, sometimes called a collapsed or switched backbone, the topology of the backbone is a star. In this configuration, the backbone is just one switch (that is why it is called, erroneously, a collapsed backbone) that connects the LANs. In a star backbone, the topology of the backbone is a star; the backbone is just one switch. Below figure shows a star backbone. Note that, in this configuration, the switch does the job of the backbone and at the same time connects the LANs.

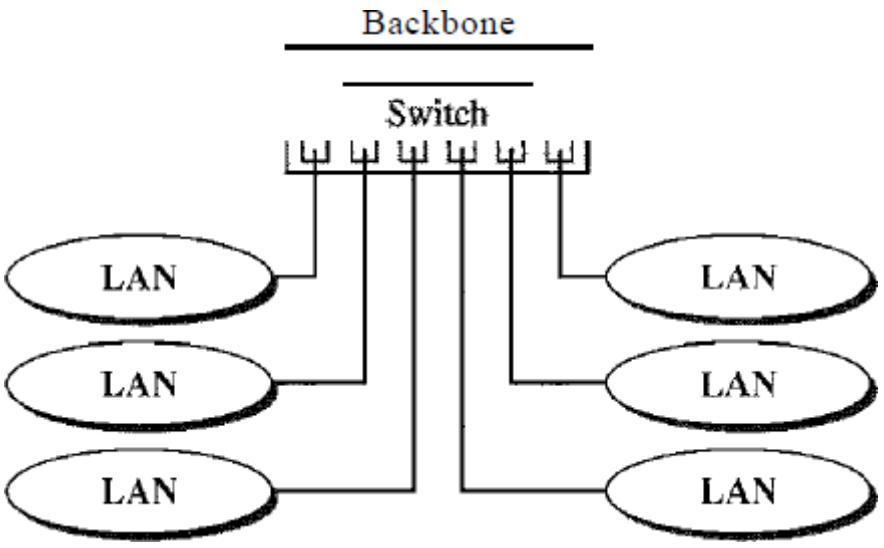


Figure. *Star backbone*

Star backbones are mostly used as a distribution backbone inside a building. In a multi floor building, we usually find one LAN that serves each particular floor. A star backbone connects these LANs. The backbone network, which is just a switch, can be installed in the basement or the first floor, and separate cables can run from the switch to each LAN. If the individual LANs have a physical star topology, either the hubs (or switches) can be installed in a closet on the corresponding floor, or all can be installed close to the switch.

We often find a rack or chassis in the basement where the backbone switch and all hubs or switches are installed.

3.7.3 Connecting Remote LANs

Another common application for a backbone network is to connect remote LANs. This type of backbone network is useful when a company has several offices with LANs and needs to connect them. The connection can be done through bridges, sometimes called remote bridges. The bridges act as connecting devices connecting LANs and point-to-point networks, such as leased telephone lines or ADSL lines. The point-to-point network in this case is considered a LAN without stations. The point-to-point link can use a protocol such as PPP. Below figure shows a backbone connecting remote LANs.

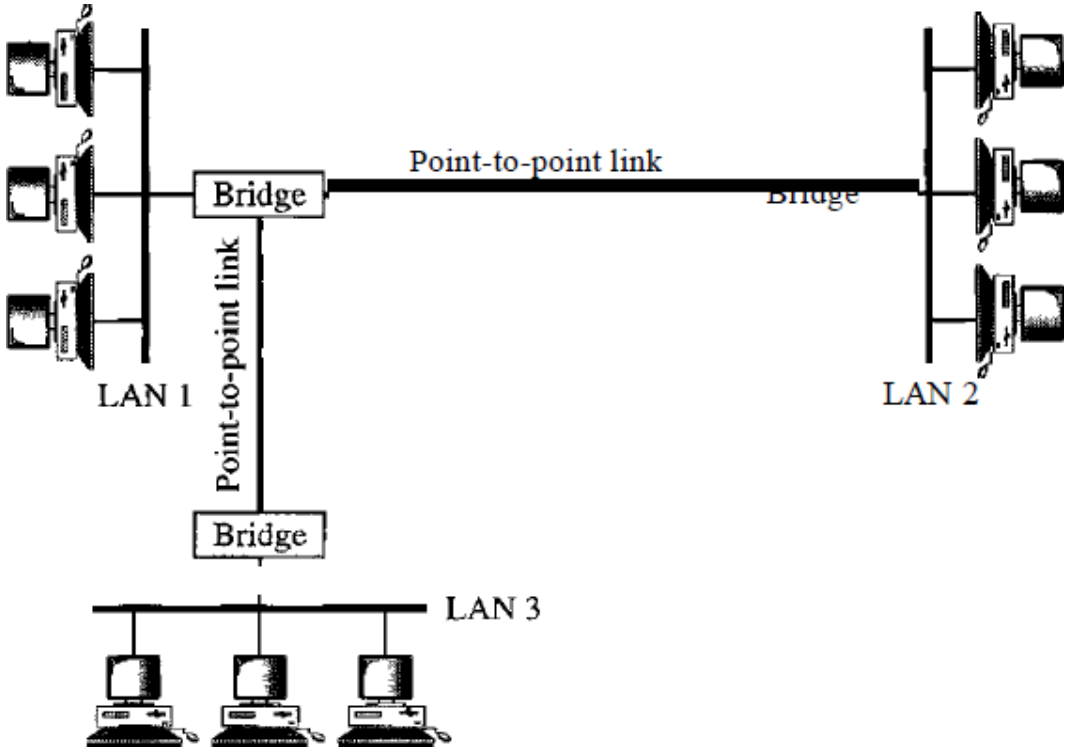


Figure. Connecting remote IANs with bridges

3.8 VIRTUAL LANs

A station is considered part of a LAN if it physically belongs to that LAN. The criterion of membership is geographic. What happens if we need a virtual connection between two stations belonging to two different physical LANs? We can roughly define a virtual local area network (VLAN) as a local area network configured by software, not by physical wiring.

Let us use an example to elaborate on this definition. Below figure shows a switched LAN in an engineering firm in which 10 stations are grouped into three LANs that are connected by a switch. The first four engineers work together as the first group, the next three engineers work together as the second group, and the last three engineers work together as the third group. The LAN is configured to allow this arrangement.

But what would happen if the administrators needed to move two engineers from the first group to the third group, to speed up the project being done by the third group? The LAN configuration would need to be changed. The network technician must rewire. The problem is repeated if, in another week, the two engineers move back to their previous group. In a switched LAN, changes in the work group mean physical changes in the network configuration.

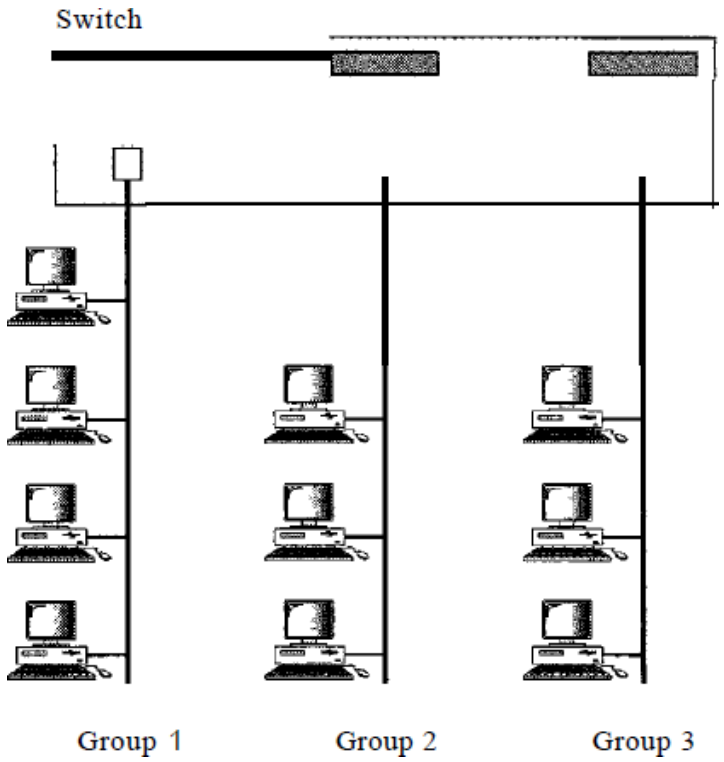


Figure. A switch connecting three LANs

Below figure shows the same switched LAN divided into VLANs. The whole idea of VLAN technology is to divide a LAN into logical, instead of physical, segments. A LAN can be divided into several logical LANs called VLANs. Each VLAN is a work group in the organization. If a person moves from one group to another, there is no need to change the physical configuration. The group membership in VLANs is defined by software, not hardware. Any station can be logically moved to another VLAN. All members belonging to a VLAN can receive broadcast messages sent to that particular VLAN.

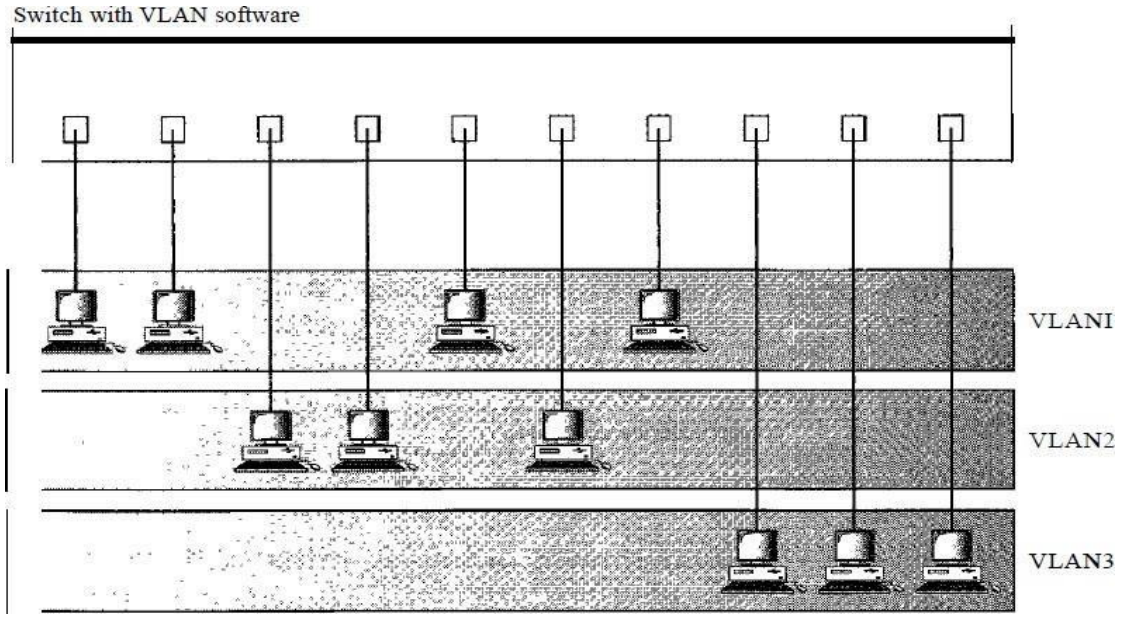


Figure. *A switch using VLAN software*

This means if a station moves from VLAN 1 to VLAN 2, it receives broadcast messages sent to VLAN 2, but no longer receives broadcast messages sent to VLAN 1. It is obvious that the problem in our previous example can easily be solved by using VLANs. Moving engineers from one group to another through software is easier than changing the configuration of the physical network. VLAN technology even allows the grouping of stations connected to different switches in a VLAN. Below Figure shows a backbone local area network with two switches and three VLANs. Stations from switches A and B belong to each VLAN.

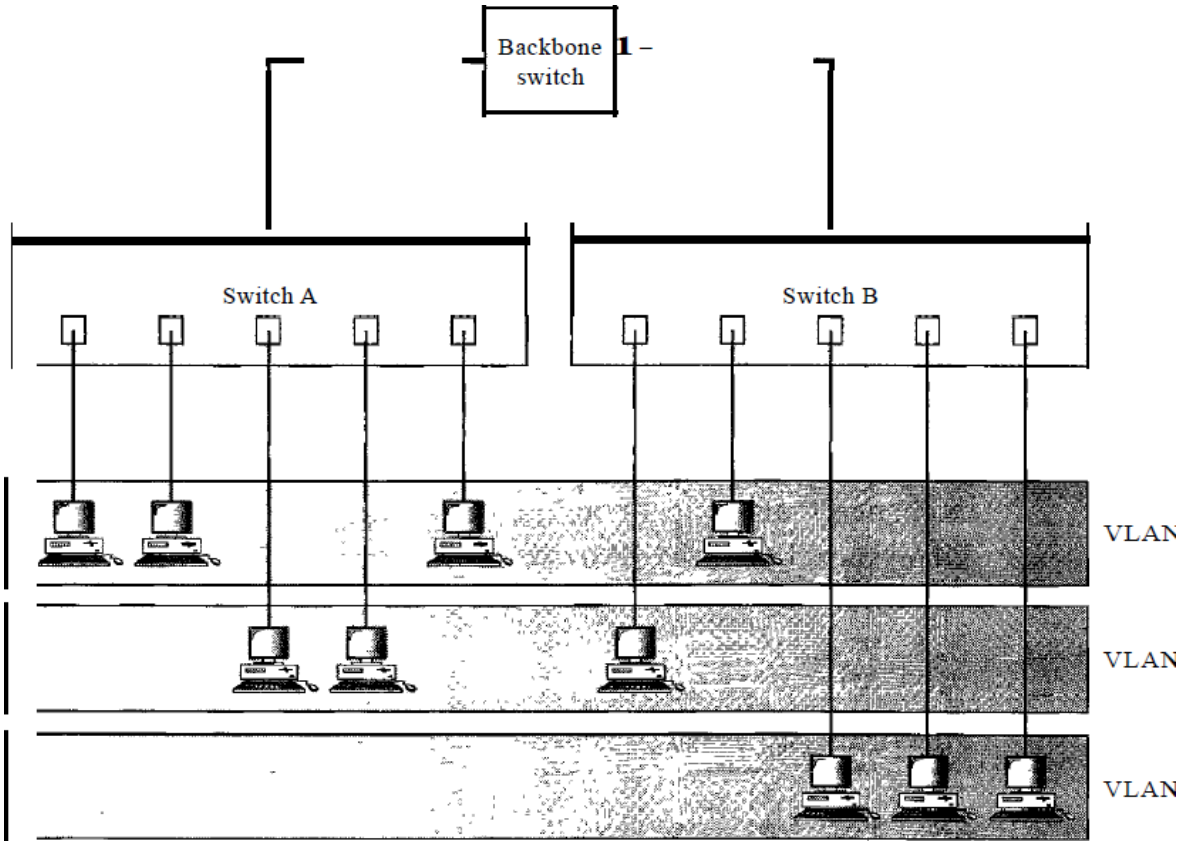


Figure. Two switches in a backbone using VLAN software

This is a good configuration for a company with two separate buildings. Each building can have its own switched LAN connected by a backbone. People in the first building and people in the second building can be in the same work group even though they are connected to different physical LANs.

From these three examples, we can define a VLAN characteristic: VLANs group stations belonging to one or more physical LANs into broadcast domains. The stations in a VLAN communicate with one another as though they belonged to a physical segment.

Membership

What characteristic can be used to group stations in a VLAN? Vendors use different characteristics such as port numbers, MAC addresses, IP addresses, IP multicast addresses, or a combination of two or more of these.

Port Numbers

Some VLAN vendors use switch port numbers as a membership characteristic. For example, the administrator can define that stations connecting to ports 1, 2, 3, and 7 belong to VLAN 1; stations connecting to ports 4, 10, and 12 belong to VLAN 2; and so on.

MAC Addresses

Some VLAN vendors use the 48-bit MAC address as a membership characteristic. For example, the administrator can stipulate that stations having MAC addresses E21342A12334 and F2A123BCD341belong to VLAN 1.

IP Addresses

Some VLAN vendors use the 32-bit IP address (see Chapter 19) as a membership characteristic.

For example, the administrator can stipulate that stations having IP addresses 181.34.23.67, 181.34.23.72, 181.34.23.98, and 181.34.23.112 belong to VLAN 1.

Multicast IP Addresses

Some VLAN vendors use the multicast IP address (see Chapter 19) as a membership characteristic. Multicasting at the IP layer is now translated to multicasting at the data link layer.

Combination

Recently, the software available from some vendors allows all these characteristics to be combined. The administrator can choose one or more characteristics when installing the software. In addition, the software can be reconfigured to change the settings.

Configuration

How are the stations grouped into different VLANs? Stations are configured in one of three ways: manual, semiautomatic, and automatic.

Manual Configuration

In a manual configuration, the network administrator uses the VLAN software to manually assign the stations into different VLANs at setup. Later migration from one VLAN to another is also done manually. Note that this is not a physical configuration; it is a logical configuration. The term *manually* here means that the administrator types the port numbers, the IP addresses, or other characteristics, using the VLAN software.

Automatic Configuration

In an automatic configuration, the stations are automatically connected or disconnected from a VLAN using criteria defined by the administrator. For example, the administrator can define the project number as the criterion for being a member of a group. When a user changes the project, he or she automatically migrates to a new VLAN.

Semiautomatic Configuration

A semiautomatic configuration is somewhere between a manual configuration and an automatic configuration. Usually, the initializing is done manually, with migrations done automatically.

Communication Between Switches

In a multi switched backbone, each switch must know not only which station belongs to which VLAN, but also the membership of stations connected to other switches. For example, in Figure 15.17, switch A must know the membership status of stations connected to switch B, and switch B must know the same about switch A. Three methods have been devised for this purpose: table maintenance, frame tagging, and time-division multiplexing.

Table Maintenance

In this method, when a station sends a broadcast frame to its group members, the switch creates an entry in a table and records station membership. The switches send their tables to one another periodically for updating.

Frame Tagging

In this method, when a frame is travelling between switches, an extra header is added to the MAC frame to define the destination VLAN. The frame tag is used by the receiving switches to determine the VLANs to be receiving the broadcast message.

Time-Division Multiplexing (TDM)

In this method, the connection (trunk) between switches is divided into timeshared channels (see TDM in Chapter 6). For example, if the total number of VLANs in a backbone is five, each trunk is divided into five channels. The traffic destined for VLAN 1 travels in channel, the traffic destined for VLAN 2 travels in channel 2, and so on. The receiving switch determines the destination VLAN by checking the channel from which the frame arrived.

IEEE Standard

In 1996, the IEEE 802.1 subcommittee passed a standard called 802.1Q that defines the format for frame tagging. The standard also defines the format to be used in multiswitched backbones and enables the use of multivendor equipment in VLANs. IEEE 802.1Q has opened the way for further standardization in other issues related to VLANs. Most vendors have already accepted the standard.

Advantages

There are several advantages to using VLANs.

Cost and Time Reduction

VLANs can reduce the migration cost of stations going from one group to another. Physical reconfiguration takes time and is costly. Instead of physically moving one station to another segment or even to another switch, it is much easier and quicker to move it by using software.

Creating Virtual Work Groups

VLANs can be used to create virtual work groups. For example, in a campus environment, professors working on the same project can send broadcast messages to one another without the necessity of belonging to the same department. This can reduce traffic if the multicasting capability of IP was previously used.

Security

VLANs provide an extra measure of security. People belonging to the same group can send broadcast messages with the guaranteed assurance that users in other groups will not receive these messages.

3.9 Wireless WANs:

3.9.1 Cellular Telephony: A cellular system comprises the following basic components:

Mobile Stations (MS): Mobile handsets, which is used by an user to communicate with another user

Cell: Each cellular service area is divided into small regions called cell (5 to 20 Km)

Base Stations (BS): Each cell contains an antenna, which is controlled by a small office.

Mobile Switching Center (MSC): Each base station is controlled by a switching office, called mobile switching center

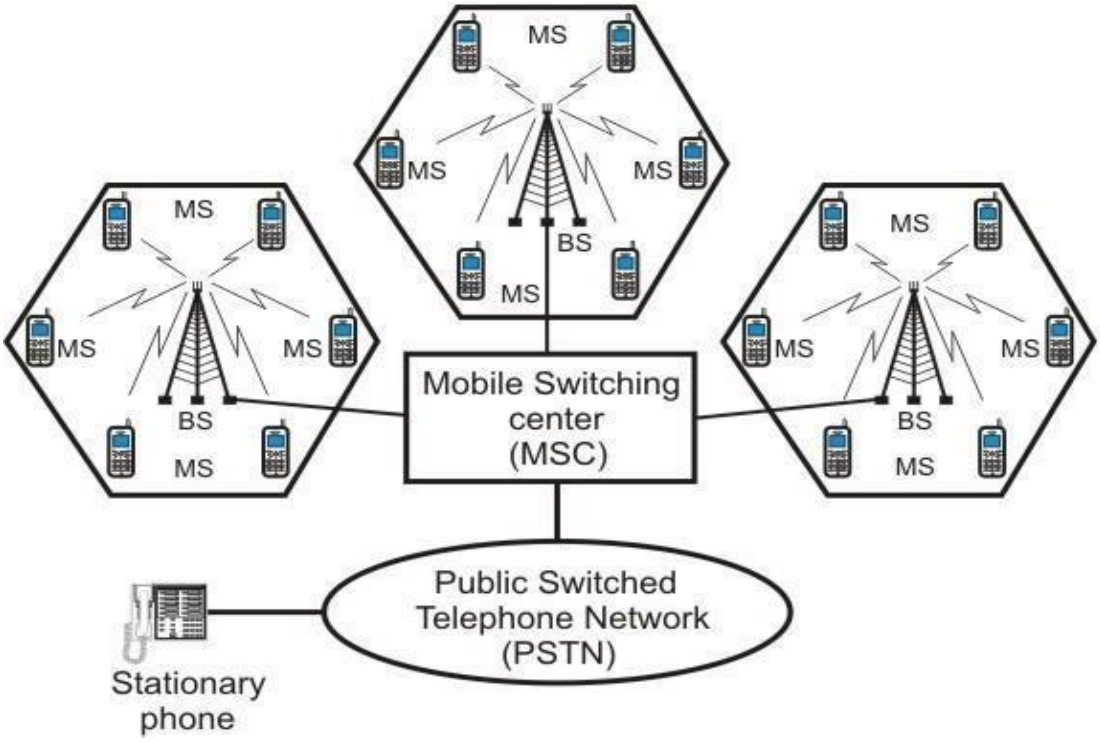


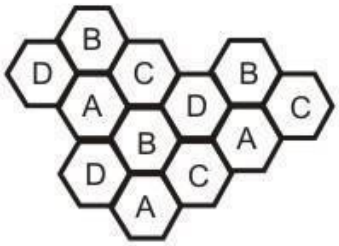
Figure. Schematic diagram of a cellular telephone system

3.9.1.1 Frequency Reuse Principle

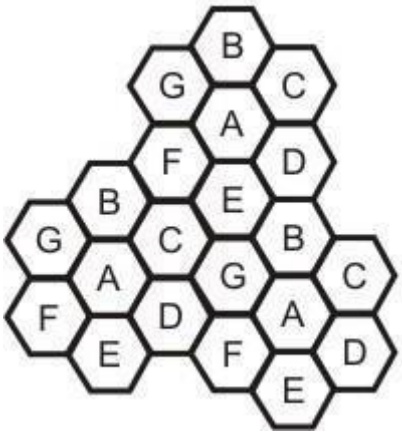
Cellular telephone systems rely on an intelligent allocation and reuse of channels. Each base station is given a group of radio channels to be used within a cell. Base stations in neighbouring cells are assigned completely different set of channel frequencies. By limiting the coverage areas, called *footprints*, within cell boundaries, the same set of channels may be used to cover different cells separated from one another by a distance large enough to keep interference level within tolerable limits. Cells with the same letter use the same set of

frequencies, called *reusing cells*. N cells which collectively use the available frequencies (S = k.N) is known as cluster. If a cluster is replicated M times within a system, then total number duplex channels (capacity) is C = M.k.N= M.S.

Reuse factor: Fraction of total available channels assigned to each cell within a cluster is 1/N. Example showing reuse factor of ¼ is shown in Fig. (a) and Fig. (b) shows reuse factor of 1/7.



(a) Cells showing reuse factor of ¼



(b) Cells showing reuse factor of 1/7

As the demand increases in a particular region, the number of stations can be increased by replacing a cell with a cluster as shown in below Fig. Here cell C has been replaced with a cluster. However, this will be possible only by decreasing the transmitting power of the base stations to avoid interference.

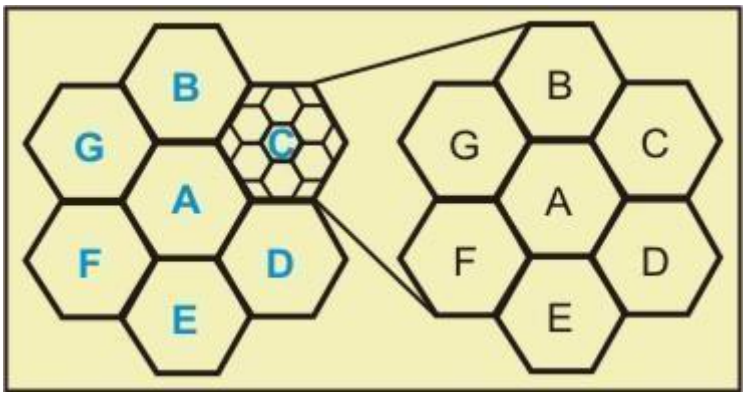


Figure. A cell is replaced by a cluster as demand increases

3.9.1.2 Transmitting and Receiving

Basic operations of transmitting and receiving in a cellular telephone network are discussed in this section.

Transmitting involves the following steps:

1. A caller enters a 10-digit code (phone number) and presses the send button.
2. The MS scans the band to select a free channel and sends a strong signal to send the number entered.
3. The BS relays the number to the MSC.
4. The MSC in turn dispatches the request to all the base stations in the cellular system.
5. The Mobile Identification Number (MIN) is then broadcast over all the forward control channels throughout the cellular system. It is known as *paging*.
6. The MS responds by identifying itself over the reverse control channel.
7. The BS relays the acknowledgement sent by the mobile and informs the MSC about the handshake.
8. The MSC assigns an unused voice channel to the call and call is established.

Receiving involves the following steps:

1. All the idle mobile stations continuously listens to the paging signal to detect messages directed at them.
2. When a call is placed to a mobile station, a packet is sent to the callee's home MSC to find out where it is.

3. A packet is sent to the base station in its current cell, which then sends a broadcast on the paging channel.
4. The callee MS responds on the control channel.
5. In response, a voice channel is assigned and ringing starts at the MS.

Roaming: Two fundamental operations are associated with Location Management; *location update* and *paging*. When a Mobile Station (MS) enters a new Location Area, it performs a location updating procedure by making an association between the foreign agent and the home agent. One of the BSs, in the newly visited Location Area is informed and the home directory of the MS is updated with its current location. When the home agent receives a message destined for the MS, it forwards the message to the MS via the foreign agent. An authentication process is performed before forwarding the message.

3.9.1.3 First Generation System

The first generation was designed for voice communication. One example is Advanced Mobile Phone System (AMPS) used in North America. AMPS is an analog cellular phone system. It uses 800 MHz ISM band and two separate analog channels; forward and reverse analog channels. The band between 824 to 849 MHz is used for reverse communication from MS to BS. The band between 869 to 894 MHz is used for forward communication from BS to MS. Each band is divided in to 832 30-KHz channels as shown in below Fig. 5.9.8.

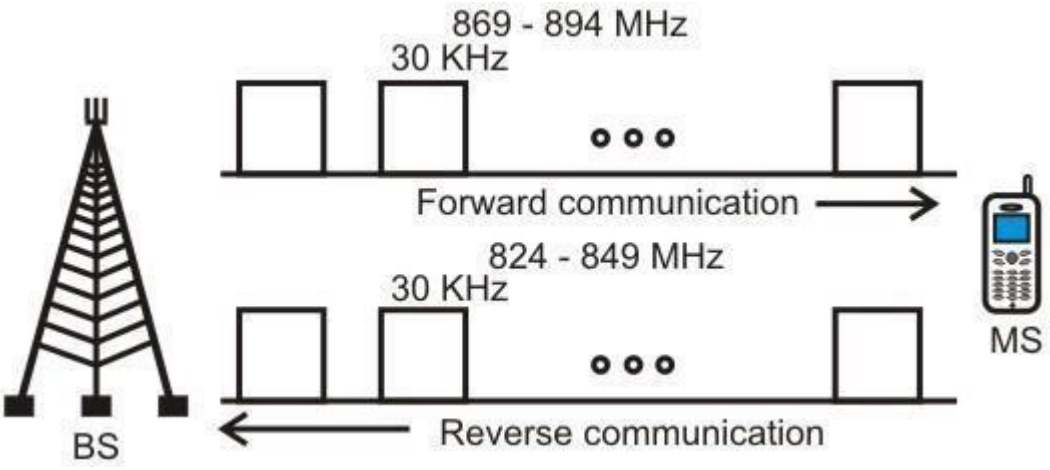


Figure Frequency bands used in AMPS system

As each location area is shared by two service providers, each provider can have 416 channels, out of which 21 are used for control. AMPS uses Frequency Division Multiple Access (FDMA) to divide each 25-MHz band into 30-KHz channels as shown in below Fig.

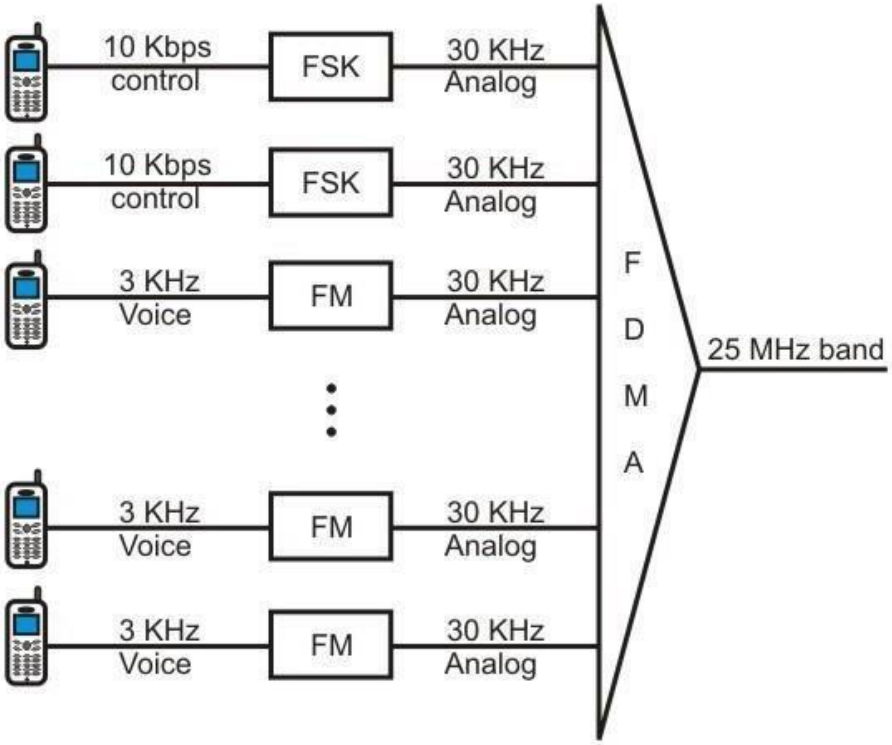


Figure. FDMA medium access control technique used in AMPS

3.9.1.3 Second Generation

The first generation cellular network was developed for analog voice communication. To provide better voice quality, the second generation was developed for digitized voice communication. Three major systems were evolved, as shown in below Fig.

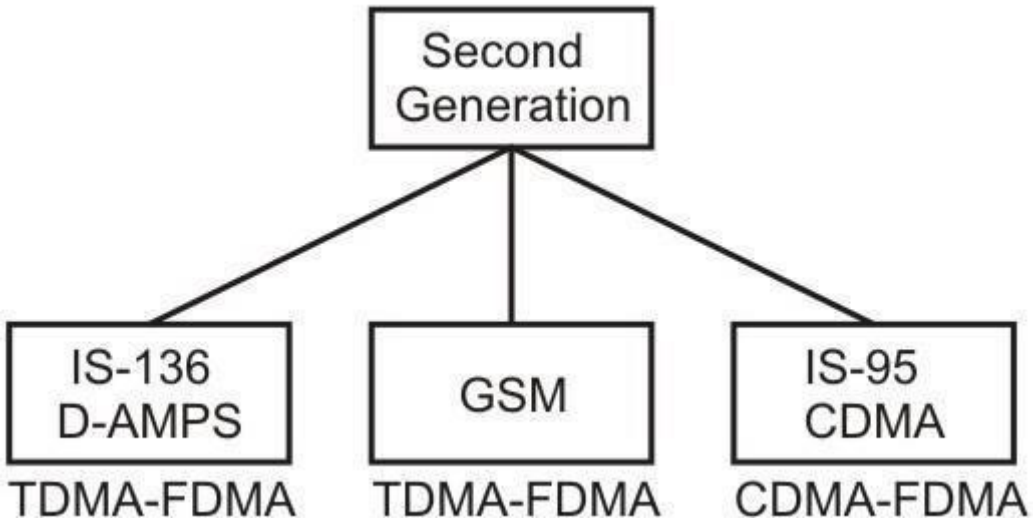


Figure. FDMA medium access control technique used in AMPS

D-AMPS: D-AMPS is essentially a digital version of AMPS and it is backward compatible with AMPS. It uses the same bands and channels and uses the frequency reuse factor of 1/7. 25 frames per second each of 1994 bits, divided in 6 slots shared by three channels. Each slot has 324 bits-159 data, 64 control, 101 error-correction as shown in below Fig. As shown in the figure, it uses both TDMA and FDMA medium access control techniques.

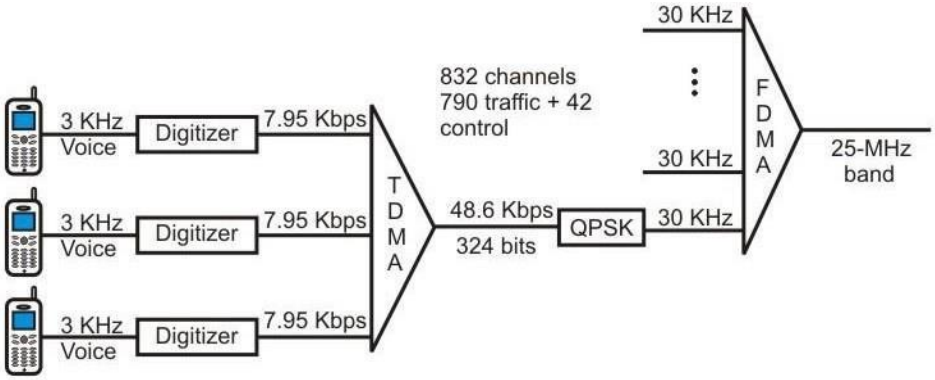


Figure. D-AMPS

GSM: The Global System for Mobile (GSM) communication is a European standard developed to replace the first generation technology. Uses two bands for duplex communication. Each voice channel is digitized and compressed to a 13Kbps digital signal. Each slot carries 156.25 bits, 8 slots are multiplexed together creating a FDM frame, 26 frames are combined to form a multi frame, as shown in below Fig. For medium access control, GSM combines both TDMA and FDMA. There is large amount of overhead in TDMA, 114 bits are generated by adding extra bits for error correction. Because of complex error correction, it allows a reuse factor as low as 1/3.

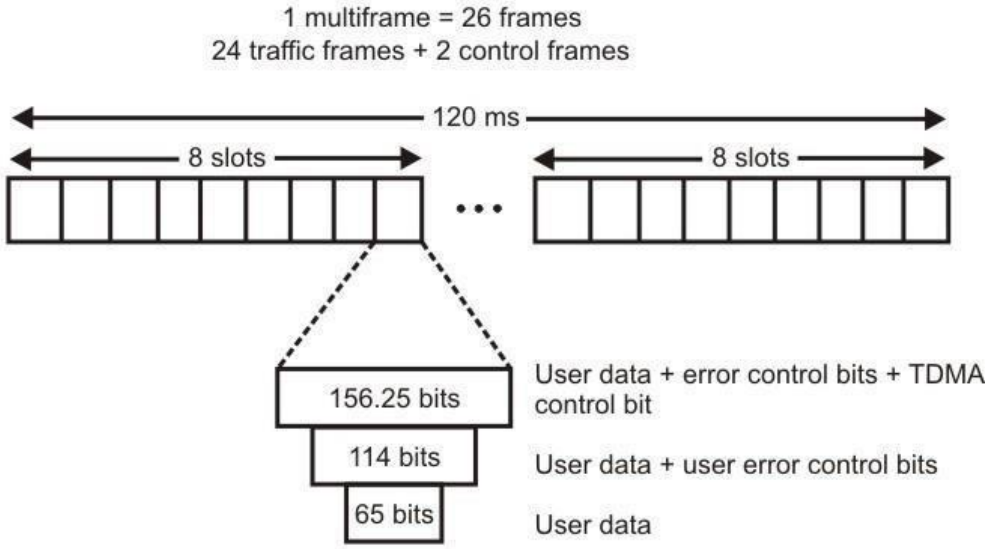


Figure. Multiframe components

IS-95 CDMA: IS-95 is based on CDMA/DSSS and FDMA medium access control technique. The forward and backward transmissions are shown in below figures (a) and (b) respectively.

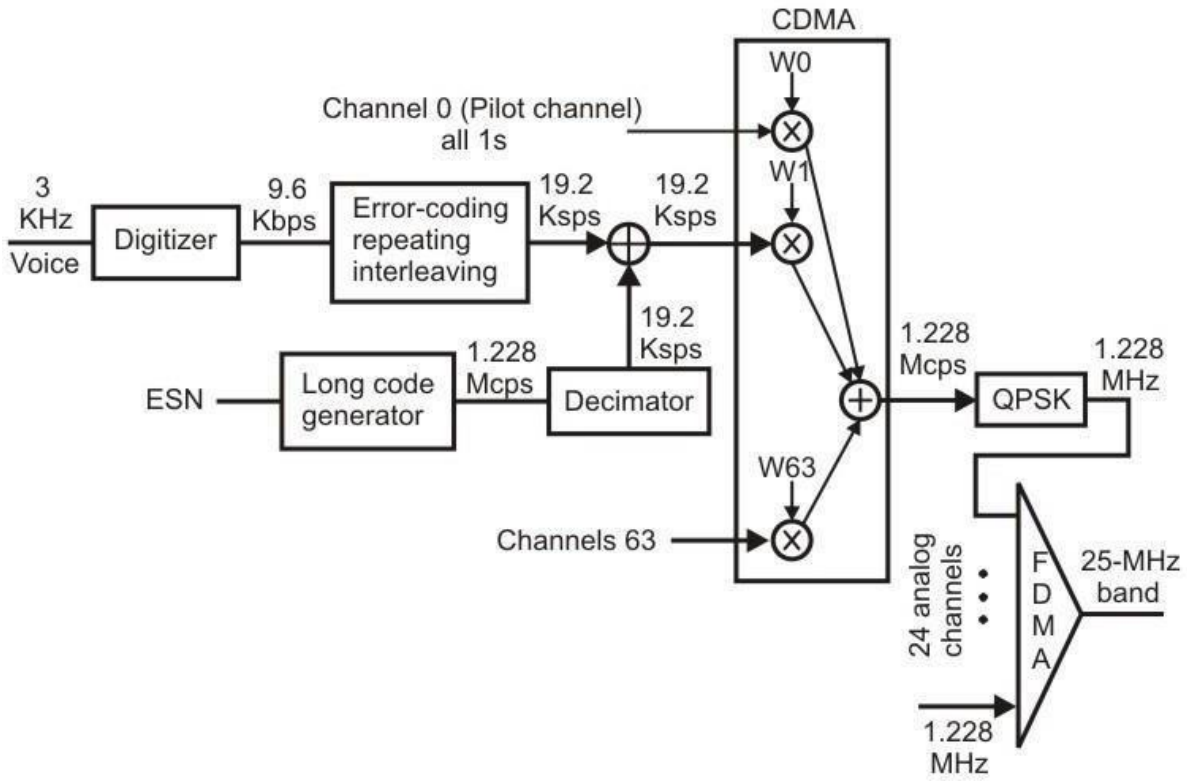


Figure (a). Forward transmission in IS-95 CDMA

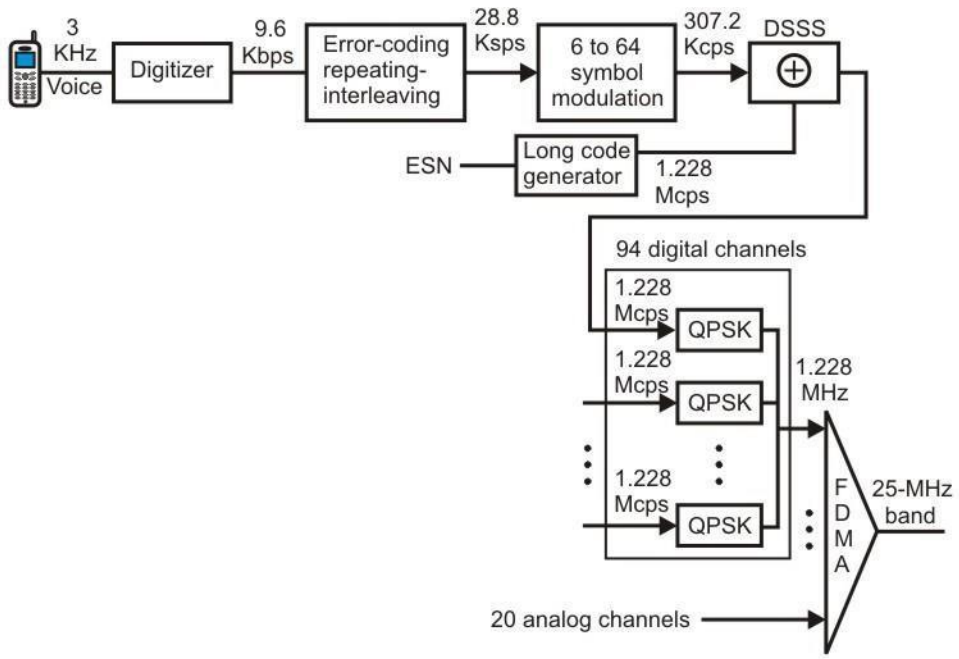


Figure (b). Backward transmission in IS-95 CDMA

3.9.1.3 Third Generation

We are presently using the second generation technologies and the development of the third generation technologies are in progress. Goals of the third generation (3G) technologies are mentioned below:

1. Allow both digital data and voice communication.
2. To facilitate universal personnel communication.
3. Listen music, watch movie, access internet, video conference, etc.

Criteria for 3G Technologies are:

1. Voice quality: Same as present PSTN network.
2. Data rate: 144Kbps (car), 384 (pedestrians) and 2Mbps (stationary).
3. Support for packet-switched and circuit-switched data services.
4. Bandwidth of 2 MHz.
5. Interface to the internet.

ITU developed a blueprint called Internet Mobile Communication for year 2000 (IMT-2000). All five Radio Interfaces adopted by IMT-2000 evolved from the second generation technologies as shown in below Fig.

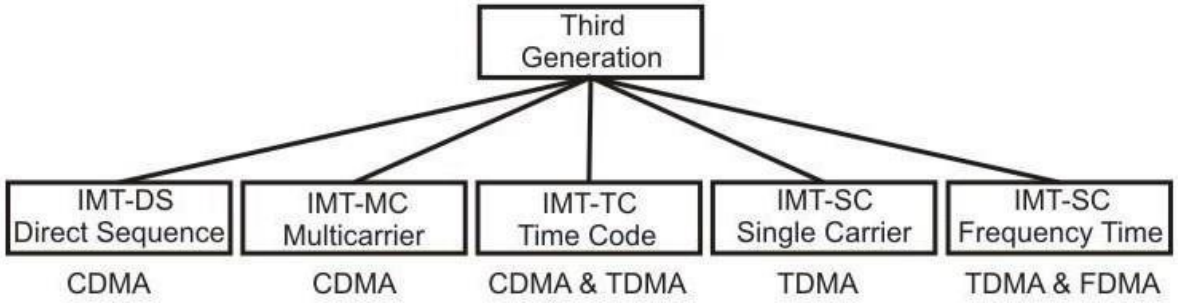


Figure. Third generation cellular technologies

3.9.2 Satellite Networks

3.9.2.1 Introduction

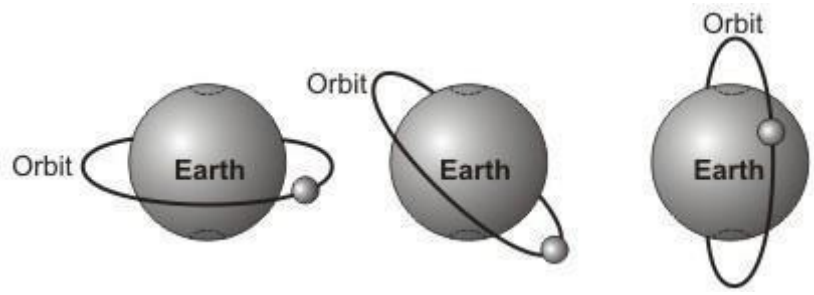
Microwave frequencies, which travel in straight lines, are commonly used for wideband communication. The curvature of the earth results in obstruction of the signal between two *earth stations* and the signal also gets attenuated with the distance it traverses. To overcome both the problems, it is necessary to use a *repeater*, which can receive a signal from one earth station, amplify it, and retransmit it to another earth station. Larger the height of a repeater from the surface of the earth, longer is the distance of line-of-sight communication. Satellite networks were originally developed to provide long-distance telephone service. So, for communication over long distances, satellites are a natural choice for use as *repeaters in the sky*. In this lesson, we shall discuss different aspects of satellite networks.

3.9.2.2 Orbits of Satellites

Artificial satellites deployed in the sky rotate around the earth on different orbits. The orbits can
be categorized into three types as follows:

- Equatorial
- Inclined
- Polar

Time required to make a complete trip around the earth, known as period, is determined by Kepler's Law of period: T2 = (4π2/GM) r3, where T is the period, G is the gravitational constant, M is the mass of the central body and r is the radius.

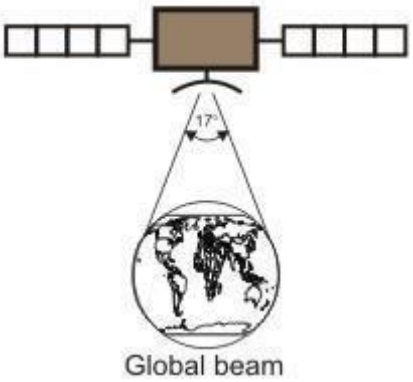


(a) **(b)** **(c)**

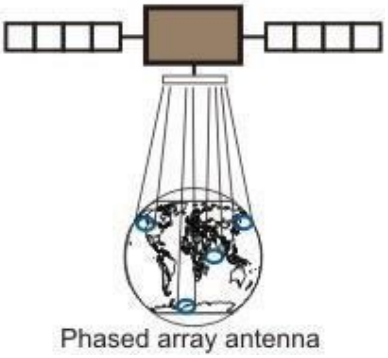
Figure. Three different orbits of satellites; (a) equatorial, (b) inclined and (c) polar

3.9.2.2 Footprint of Satellites

Signals from a satellite is normally aimed at a specific area called the *footprint*. Power is maximum at the center of the footprint. It decreases as the point moves away from the footprint center. The amount of time a beam is pointed to a given area is known as *dwell time.*



(a) Footprint using a global beam



(b) Footprint using a phased array antenna

3.9.2.3 Categories of Satellites

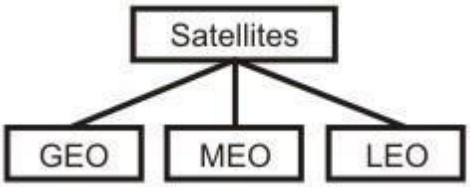
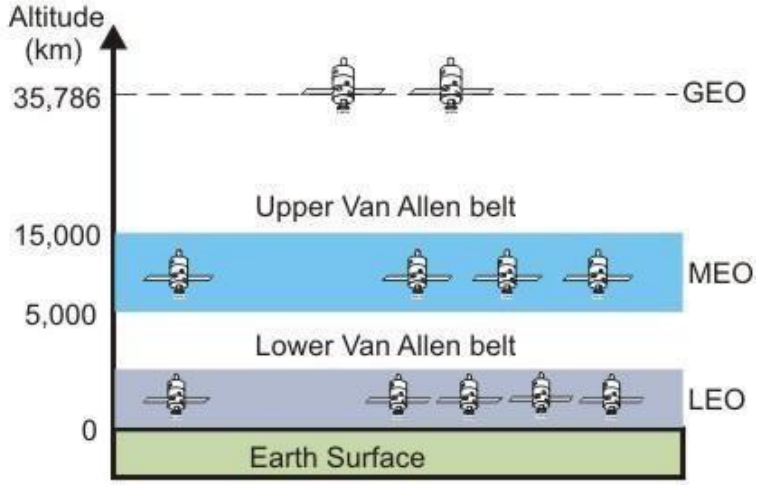


Figure Categories of satellites

The satellites can be categorized into three different types , based on the location of the orbit. These orbits are chosen such that the satellites are not destroyed by the high-energy charged particles present in the two *Van Allen belts*, as shown in below Fig.



The Low Earth Orbit (LEO) is below the lower Van Allen belt in the altitude of 500 to 2000 Km. The Medium Earth Orbit (MEO) is in between the lower Van Allen belt and upper Van Allen belt in the altitude of 5000 to 15000 Km. The Medium Earth Orbit (MEO) is in between the lower Van Allen belt and upper Van Allen belt in the altitude of 5000 to 15000 Km. Above the upper Van Allen belt is the Geostationary Earth Orbit (GEO) at the altitude of about 36,000 Km. Below the Geostationary Earth Orbit and above the upper Van Allen belt is Global Positioning System (GPS) satellites at the altitude of 20,000 Km. The orbits of these satellite systems are shown in below Fig.

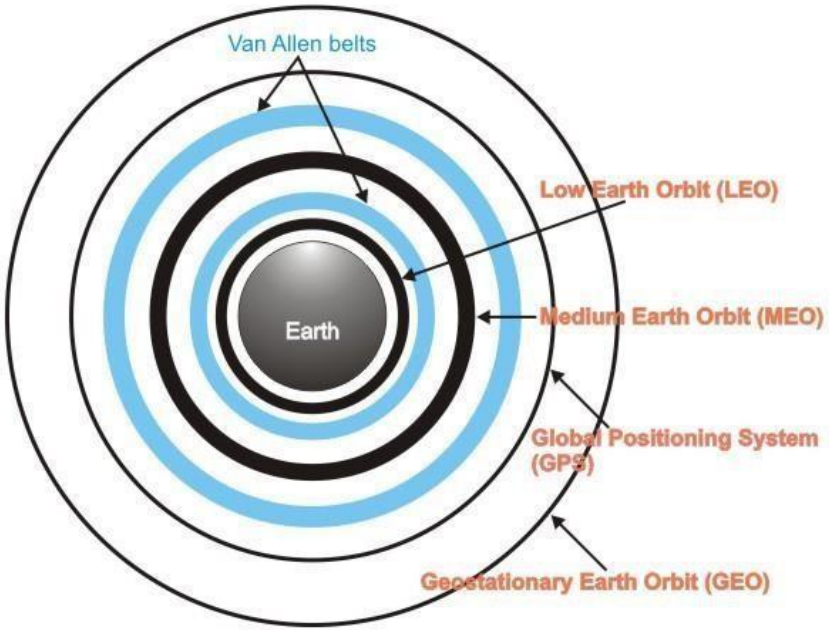


Figure. Orbits of the satellites of different categories

3.9.2.4 Frequency Bands

Two frequencies are necessary for communication between a ground station and a satellite; one for communication from the ground station on the earth to the satellite called *uplink frequency* and another frequency for communication from the satellite to a station on the earth, called *downlink frequency*. These frequencies, reserved for satellite communication, are divided in several bands such as L, S, Ku, etc are in the gigahertz (microwave) frequency range as shown in below table. Higher the frequency, higher is the available bandwidth.

Table. Frequency bands for satellite communication

Band	Downlink Frequency (GHz)	Uplink Frequency (GHz)	Bandwidth(MHz)
L	1.5	1.6	15
S	1.9	2.2	70
C	4	6	500
Ku	11	14	500
Ka	20	30	3500

3.9.2.5 Low Earth Orbit Satellites

The altitude of LEO satellites is in the range of 500 to 1500 Km with a rotation period of 90 to 120 min and round trip delay of less than 20 ms. The satellites rotate in polar orbits with a rotational speed of 20,000 to 25,000 Km. As the footprint of LEO satellites is a small area of about 8000 Km diameter, it is necessary to have a constellation of satellites, as shown in below Fig., which work together as a network to facilitate communication between two earth stations anywhere on earth's surface.

Figure. LEO satellite network

The satellite system is shown in below Fig. Each satellite is provided with three links; the User Mobile Link (UML) for communication with a mobile station, the Gateway Link (GWL) for communication with a earth station and the Inter-satellite Link (ISL) for communication between two satellites, which are close to each other. Depending on the frequency bands used by different satellites, these can be broadly categorized into three types; the little LEOs operating under 1 GHz and used for low data rate communication, the big LEOs operating in the range 1 to 3 GHz and the Broadband and the broadband LEOs provide communication capabilities similar to optical networks.

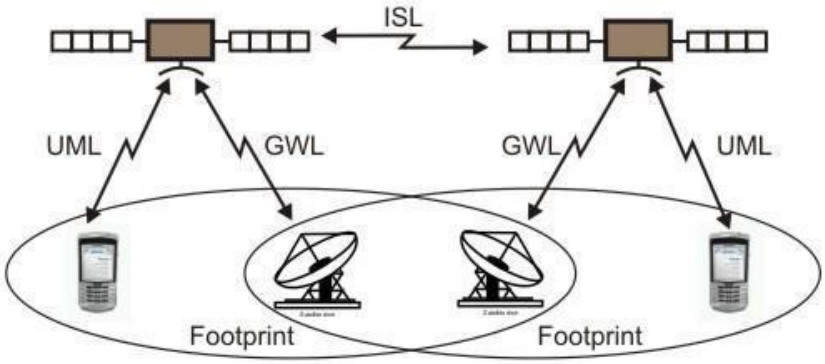


Figure. LEO satellite system

Iridium System

The Iridium system was a project started by Motorola in 1990 with the objective of providing worldwide voice and data communication service using handheld devices. It took 8 years to materialize using 66 satellites. The 66 satellites are divided in 6 polar orbits at an altitude of 750 Km. Each satellite has 48 spot beams (total 3168 beams). The number of active spot beams is about 2000. Each spot beam covers a cell as shown in below Fig.

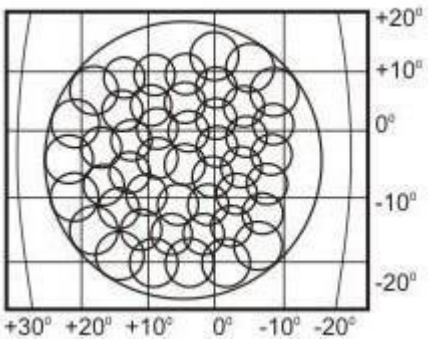


Figure. Overlapping spot beams of the Iridium system

The Teledesic System

The Teledesic project started in 1990 by Craig McCaw and Bill Gates in 1990 with the objective of providing fiber-optic like communication (Internet-in-the-sky). It has 288 satellites in 12 polar orbits, each orbit having 24 satellites at an altitude of 1350 Km. Three types of communications that are allowed in Teledasic are as follows;

ISL: Intersatellite communication allows eight neighbouring satellites to communicate with each other

GWL: Communication between a satellite and a gateway

UML: Between an user and a satellite

The surface of the earth is divided into thousands of cells and each satellite focuses it beams to a cell during dwell time. It uses Ka band communication with data rates of 155Mbps uplink and 1.2Gbps downlink.

3.9.2.6 Medium Earth Orbit Satellites

MEO satellites are positioned between two Van Allen Belts at an height of about 10,000 Km with a rotation period of 6 hours. One important example of the MEO satellites is the Global Positioning System (GPS) as briefly discussed below:

GPS

The Global Positioning System (GPS) is a satellite-based navigation system. It comprises a network of 24 satellites at an altitude of 20,000 Km (Period 12 Hrs) and an inclination of 55° as shown in below Fig. Although it was originally intended for military applications and deployed by the Department of Defence, the system is available for civilian use since 1980.

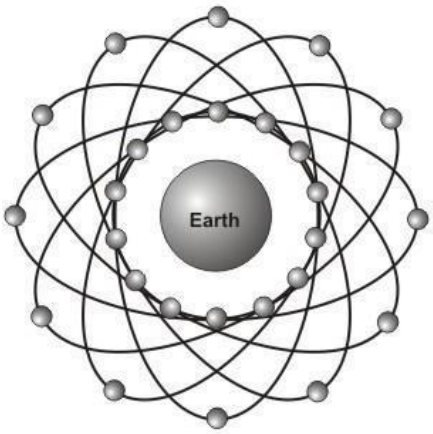


Figure. Global positioning system

It allows land, sea and airborne users to measure their position, velocity and time. It works in any weather conditions, 24 hrs a day. Positioning is accurate to within 15 meters. It is used for land and sea navigation using the principle of triangulation as shown in below Fig. It requires that at any time at least 4 satellites to be visible from any point of earth. A GPS receiver can find out the location on a map.

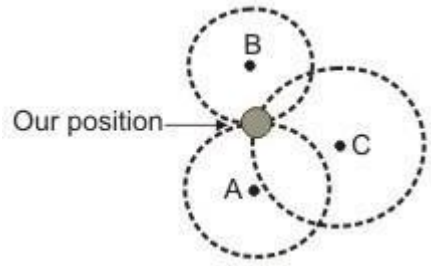


Figure. Triangulation approach used to find the position of an object

3.9.2.7 GEO Satellites

Back in 1945, the famous science fiction writer Arthur C. Clarke suggested that a radio relay satellite in an equatorial orbit with a period of 24 h would remain stationary with respect to the earth's surface and that can provide radio links for long distance communication. Although the rocket technology was not matured enough to place satellites at that height in those days, later it became the basis of Geostationary (GEO) satellites. To facilitate constant communication, the satellite must move at the same speed as earth, which are known as Geosynchronous. GEO satellites are placed on equatorial plane at an Altitude of 35786Km. The radius is 42000Km with the period of 24 Hrs. With the existing technology, it is possible to have 180 GEO satellites in the equatorial plane. But, only three satellites are required to provide full global coverage as shown in below fig.

Long round-trip propagation delay is about 270 msec between two ground stations. Key features of the GEO satellites are mentioned below:

Inherently broadcast media: It does not cost much to send to a large number of stations

- Lower privacy and security: Encryption is essential to ensure privacy and security
- Cost of communication is independent of distance

The advantages are best exploited in VSATs as discussed in the following section.

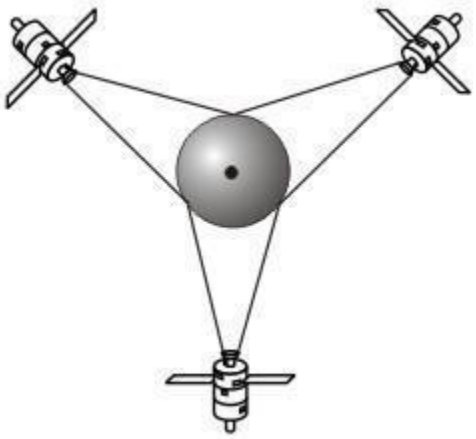


Figure Three satellites providing full global coverage in GEO system

VSAT Systems:

VSAT stands for Very Small Aperture Terminal. It was developed to make access to the satellite more affordable and without any intermediate distribution hierarchy. Most VSAT systems operate in Ku band with antenna diameter of only 1 to 2 meters and transmitting power of 1 to 2 watts. Possible implementation approaches are: *One-way*, *Split two-way and two-way*. One-way VSAT configuration is shown in below Fig.

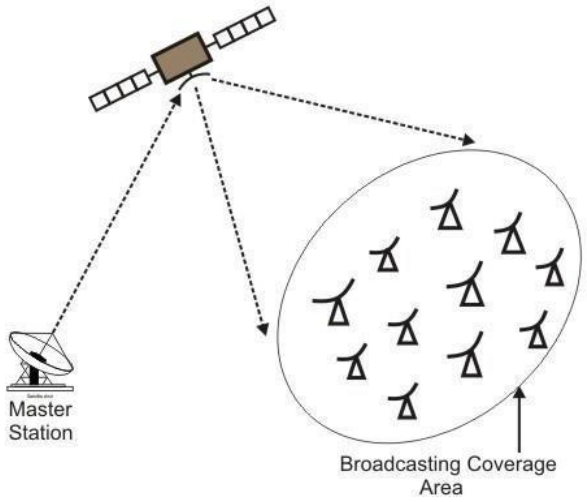


Figure. One-way satellite configurations

In this case, there is a master station and there can be many narrow-banding groups within a large broadcasting area of the satellite. This configuration is used in Broadcast Satellite Service (BSS). Other applications of one-way VSAT system are the Satellite Television Distribution system and Direct to Home (DTH) service as shown in below Fig. which has become very popular in recent times.

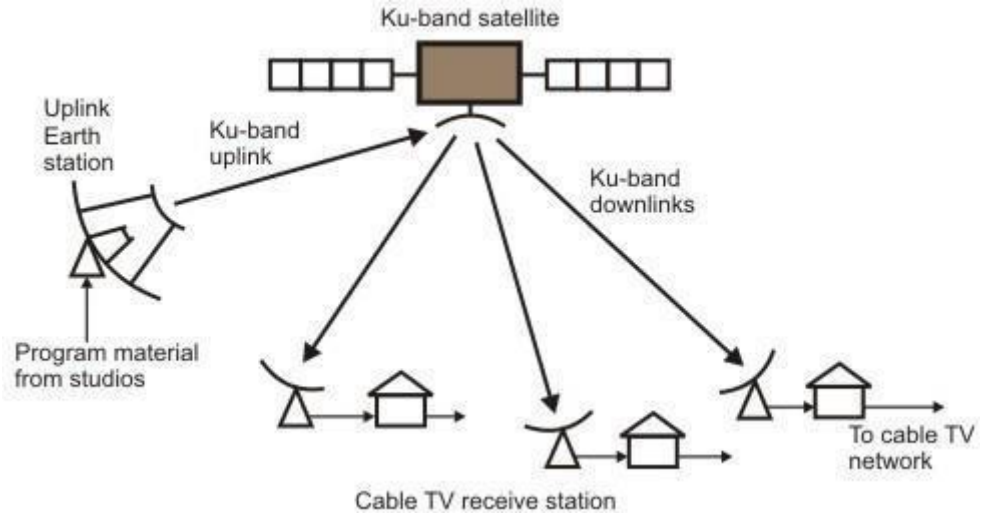
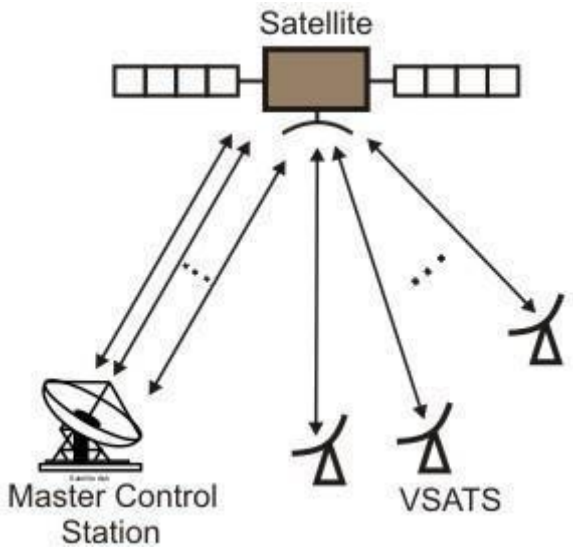
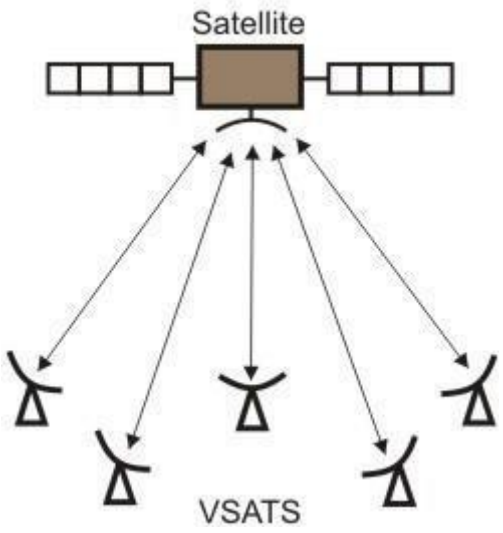


Figure. Satellite Television distribution system

In case of two-way configuration, there are two possible topologies: star and mesh. In the first case, all the traffic is routed through the master control station as shown in below Fig. (a). On the other hand, each VSAT has the capability to communicate directly with any other VSAT stations in the second case, as shown in below Fig. (b). In case of split two-way system, VSAT does not require uplink transmit capability, which significantly reduces cost.





(a) Two-way VSAT configuration with star topology, (b) Two-way VSAT configuration with mesh topology

UNIT IV

NETWORK LAYER

1. LOGICAL ADDRESSING

IPv4 ADDRESSES

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.

An IPv4 address is 32 bits long.

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. We will see later that, 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. We will see later that a router is such a device.

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.

The IPv4 addresses are unique and universal.

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 *2N* because each bit can have two different values (0 or 1) and *N* bits can have *2N* values. IPv4 uses 32-bit addresses, which means that the address space is 232 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.

The address space of IPv4 is 232 or 4,294,967,296.

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

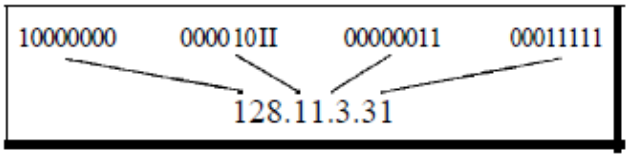
To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. The following is the dotted decimal notation of the above address:

117.149.29.2

Figure 19.1 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.

Figure 19.1 *Dotted-decimal notation and binary notation for an IPv4 address*



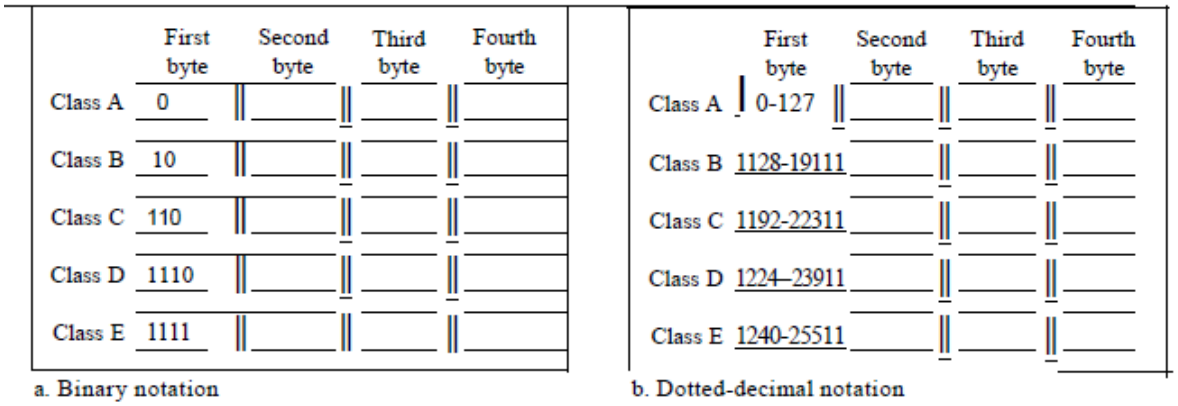
Classful Addressing

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.

In classful addressing, the address space is divided into five classes:A, B, C, D, and E. 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 Figure 19.2.

Figure 19.2 *Finding the classes in binary and dotted-decimal notation*



Limitations of classful addressing:

- 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 which means 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

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 19.1.

Table 19.1 *Number ofblocks and block size in classfulIPv4 addressing*

Class	*Number ofBlocks*	*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 as we will see in a later chapter. 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. « In c1assfnl addressing, a large part of the available addresses were wasted.

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. Figure 19.2 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

Table 19.2 *Default masks for classful addressing*

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	11111111 00000000 00000000 00000000	255.0.0.0	*18*
B	11111111 11111111 00000000 00000000	255.255.0.0	*116*
C	11111111 11111111 11111111 00000000	255.255.255.0	*124*

Although the length of the netid and hostid is predetermined in we can also use a mask, which is a 32-bit number made of contiguous 1s followed by contiguous 0s.

- The masks for classes A, B, and C are shown in below table.
- The mask can help us to find the netid and the hostid.

SUBNETTING

- Subnetting was introduced for classful addressing.
- If an organization was granted a large block in class A or B,
- Then, 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 Is in the
- mask, as we will see later when we discuss classless addressing.

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 supernetwork or a supemet.
- An organization can apply for a set of class C blocks instead of just one.
- Ex: An organization that needs 1000 addresses can be granted four contiguous class C blocks. The organization can then use these addresses to create one supernetwork.
- 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

CLASSLESS ADRESSING:

- 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.

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

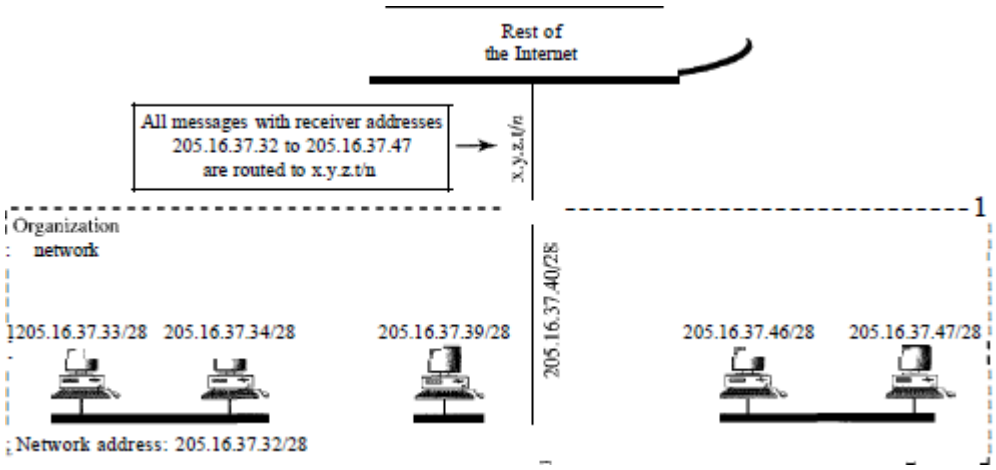
MASK:

- A better way to define a block of addresses is to select any address in the block and the mask.
- As we discussed before, a mask is a 32-bit number in which the n leftmost bits are 1s and the 32 - n rightmost bits are Os.
- However, in classless addressing the mask for a block can take any value from 0 to 32.
- It is very convenient to give just the value of n preceded by a slash (CIDR notation).
- The address and the In notation completely define the whole block (the first address, the last address, and the number of addresses

In IPv4 addressing, a block of addresses can be defined as x.y.z.t /*n* in which x.y.z.t defines one of the addresses and the /*n* defines the mask.

NETWORK ADDRESSES:

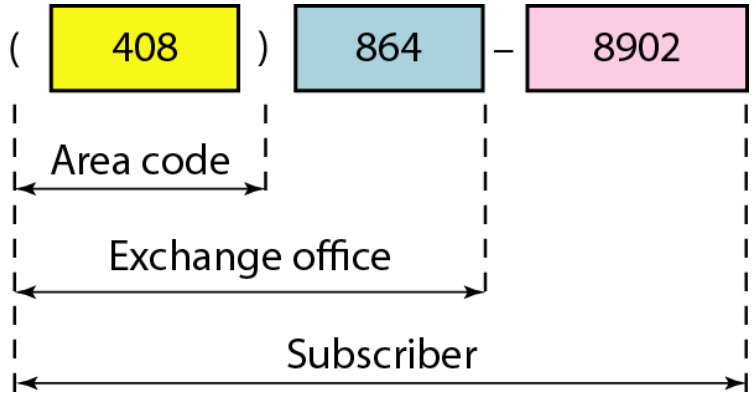
- When an organization is given a block of addresses, the organization is free to allocate the addresses to the devices that need to be connected to the Internet.
- The first address in the class, however, is normally (not always) treated as a special address.
- The first address is called the network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



HIERARCHY

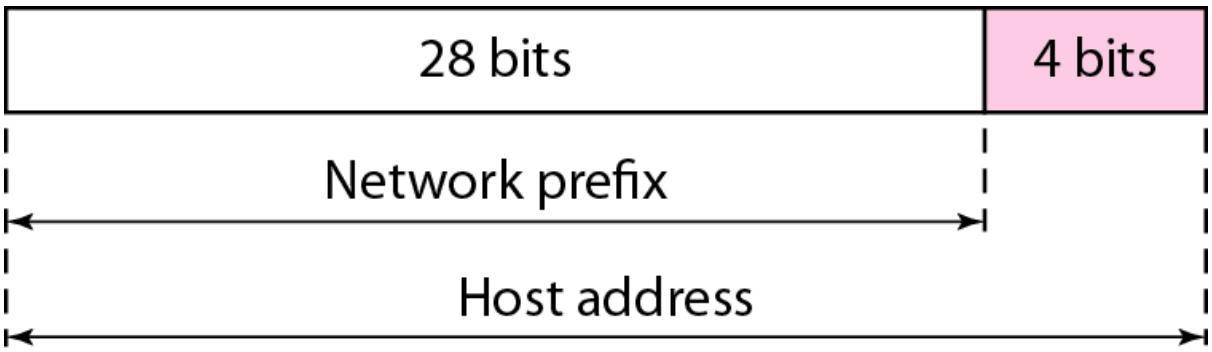
- IP addresses, like other addresses or identifiers can have levels of hierarchy.
- For example, a telephone network in North America has three levels of hierarchy.

- The leftmost three digits define the area code, the next three digits define the exchange, the last four digits define the connection of the local loop to the central office. Figure 19.5 shows the structure of a hierarchical telephone number



TWO-LEVEL HIERARCHY

- An IP address can define only two levels of hierarchy when not subnetted.
- The n leftmost bits of the address x.y.z.tJn define the network (organization network); the 32 – n rightmost bits define the particular host (computer or router) to the network.
- The two common terms are prefix and suffix.
- The part of the address that defines the network is called the prefix; the part that defines the host is called the suffix.



THREE-LEVEL HIERARCHY

- An organization that is granted a large block of addresses may want to create clusters of networks (called subnets) and divide the addresses between the different subnets.

- The rest of the world still sees the organization as one entity; however, internally there are several subnets.
- All messages are sent to the router address that connects the organization to the rest of the Internet; the router routes the message to the appropriate subnets.
- The organization, however, needs to create small subblocks of addresses, each assigned to specific subnets.
- The organization has its own mask; each subnet must also have its own.

Figure 19.7 ***Configuration and addresses in a subnetted network***

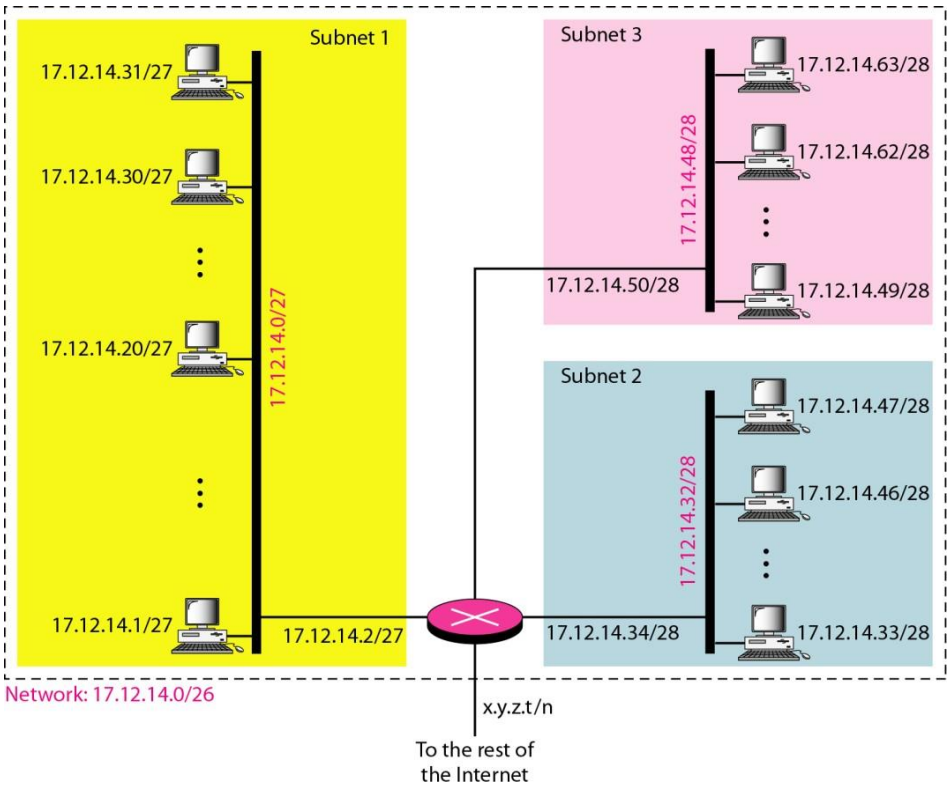
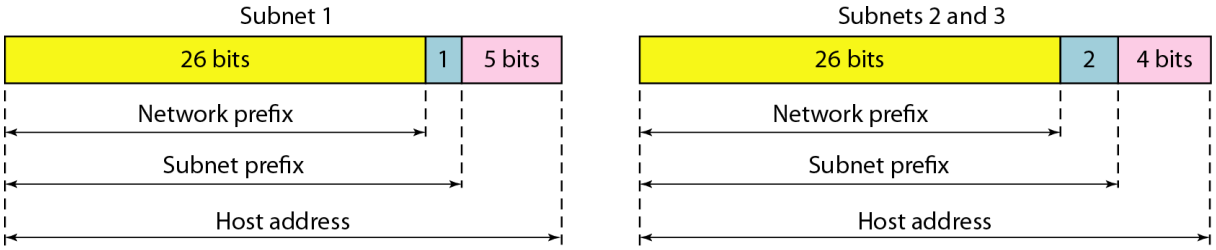


Figure 19.8 ***Three-level hierarchy in an IPv4 address***



MORE LEVELS OF HIERARCHY:

The structure of classless addressing does not restrict the number of hierarchical levels.

An organization can divide the granted block of addresses into subblocks.

Each subblock can in turn be divided into smaller subblocks. And so on.

One example of this is seen in the ISPs. A national ISP can divide a granted large block into smaller blocks and assign each of them to a regional ISP. A regional ISP can divide the block received from the national ISP into smaller blocks and assign each one to a local ISP. A local ISP can divide the block received from the regional ISP into smaller blocks and assign each one to a different organization. Finally, an organization can divide the received blockand make several subnets out of it.

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.

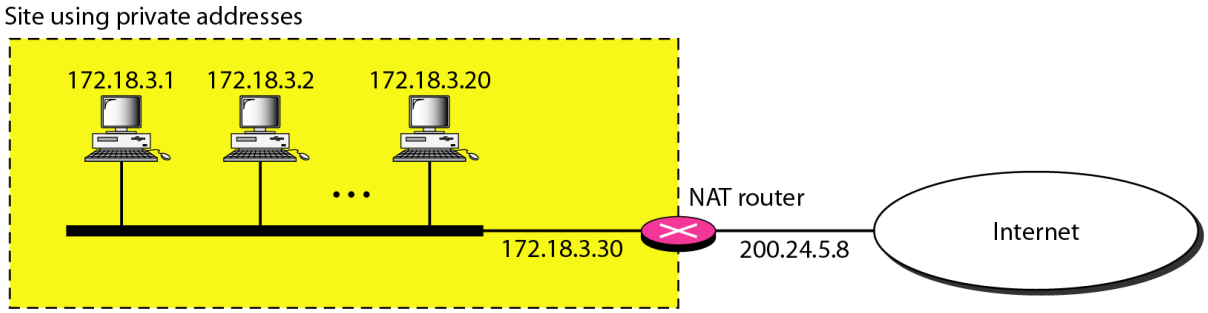
- Each ISP, in turn, divides its assigned block into smaller subblocks and grants the subblocks to its customers.
- In other words, an ISP receives one large block to be distributed to its Internet users.
- This is called address aggregation: many blocks of addresses are aggregated in one block and granted to one ISP.

NETWORK ADDRESS TRANSLATION:

- The number of home users and small businesses that want to use the Internet is ever increasing.
- In the beginning, a user was connected to the Internet with a dial-up line, which means that she was connected for a specific period of time.
- An ISP with a block of addresses could dynamically assign an address to this user. An address was given to a user when it was needed. But the situation is different today.
- Home users and small businesses can be connected by an ADSL line or cable modem.
- In addition, many are not happy with one address; many have created small networks with several hosts and need an IP address for each host.
- With the shortage of addresses, this is a serious problem-solution to this problem is called network address translation (NAT).
- NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally.
- The traffic inside can use the large set; the traffic outside, the small set.
- To separate the addresses used inside the home or business and the ones used for the Internet, the Internet authorities have reserved three sets of addresses as private addresses, shown in below table.

Range			Total
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

- Any organization can use an address out of this set without permission from the Internet authorities.
- Everyone knows that these reserved addresses are for private networks.
- They are unique inside the organization, but they are not unique globally.
- No router will forward a packet that has one of these addresses as the destination address.
- The site must have only one single connection to the global Internet through a router that runs the NAT software. Below fig. shows a simple implementation of NAT.

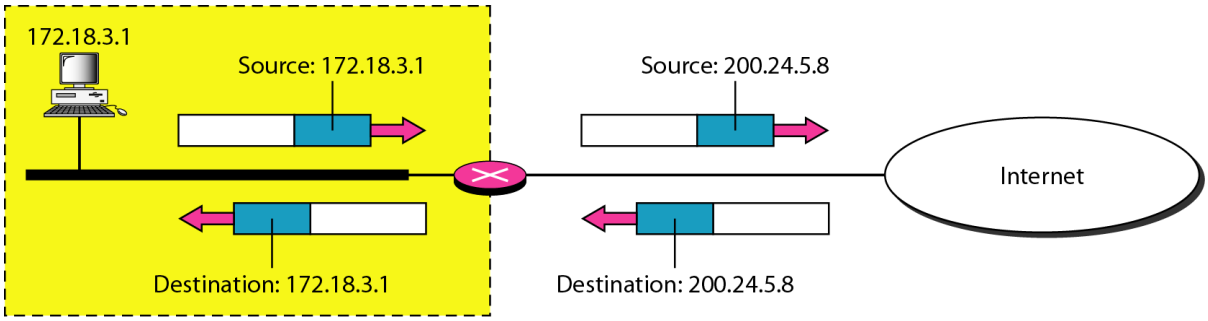


- The private network uses private addresses.
- The router that connects the network to the global address uses one private address and one global address.
- The private network is transparent to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8

ADDRESS TRANSLATION:

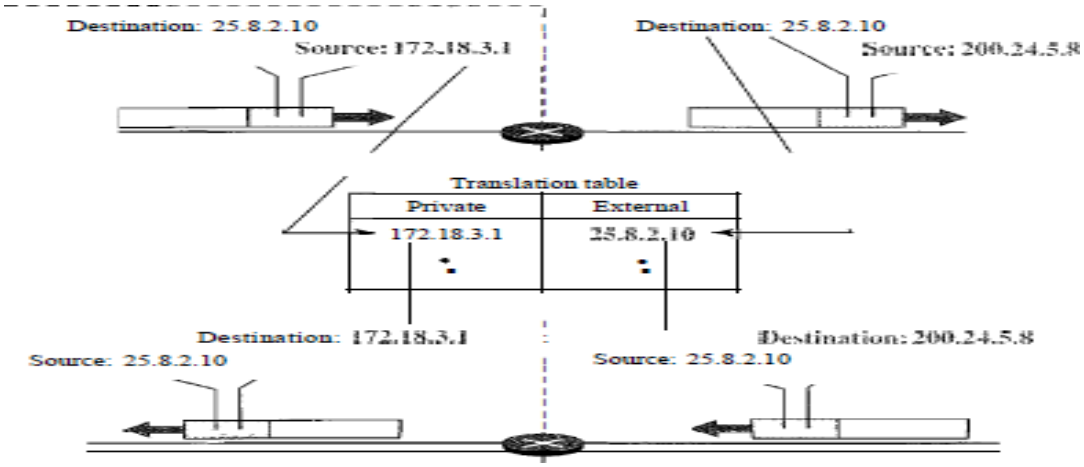
- All the outgoing packets go through the NAT router, which replaces the source address in the packet with the global NAT address.
- All incoming packets also pass through the NAT router, which replaces the destination address in the packet (the NAT router global address) with the appropriate private address.

Below fig. shows an example of address translation



TRANSLATION TABLE:

- Translation table has only two columns: the private' address and the external address (destination address of the packet).
- When the router translates the source address of the outgoing packet, it also makes note of the destination address-where the packet is going.
- When the response comes back from the destination, the router uses the source address of the packet (as the external address) to find the private address of the packet.



POOL OF IP ADDRESSES

- As NAT router has only one global address, only one private network host can access the same external host.

To remove this restriction, the NAT router uses a pool of global addresses

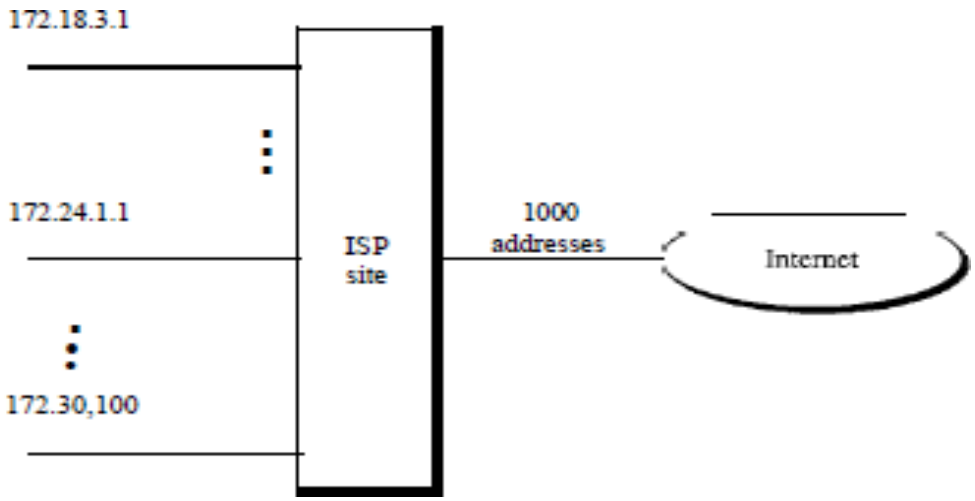
USING BOTH IP ADDRESSES AND PORT NUMBERS

- To allow a many-to-many relationship between private-network hosts and external server programs, we need more information in the translation table.
- For example, suppose two hosts with addresses 172.18.3.1 and 172.18.3.2 inside a private network need to access the HTTP server on external host 25.8.3.2.
- If the translation table has five columns, instead of two, that include the source and destination port numbers of the transport layer protocol, the ambiguity is eliminated.

Private Address	*Private Port*	*External Address*	*External Port*	*Transport Protocol*
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
. . .	. . .	. . .	. . .	. . .

NAT AND ISP

- An ISP that serves dial-up customers can use NAT technology to conserve addresses.
- For example, suppose an ISP is granted 1000 addresses, but has 100,000 customers. Each of the customers is assigned a private network address.
- The ISP translates each of the 100,000 source addresses in outgoing packets to one of the 1000 global addresses; it translates the global destination address in incoming packets to the corresponding private address. Below fig illustrates this concept.



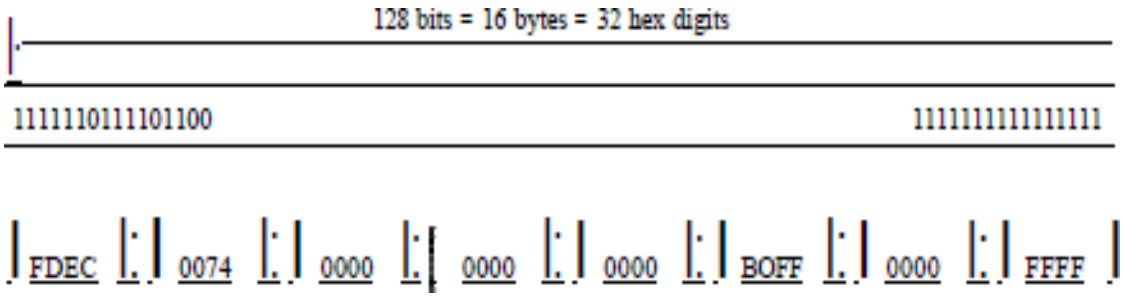
19-2 IPv6 ADDRESSES

Despite all short-term solutions, address depletion is still a long-term problem for the Internet. This and other problems in the IP protocol itself have been the motivation for IPv6.

Note: An IPv6 address is 128 bits long.

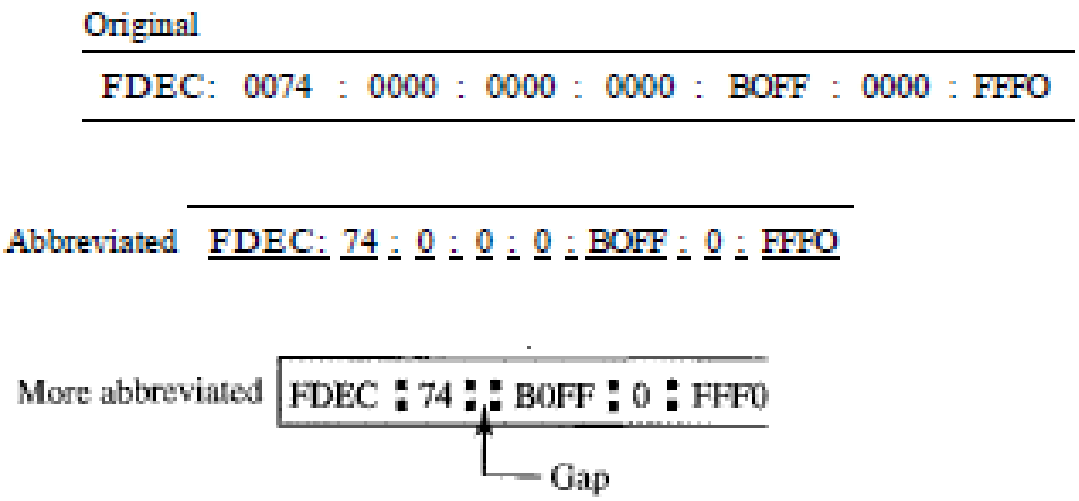
HEXADECIMAL COLON NOTATION

- 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 below fig.



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 below fig.)



ADDRESS SPACE:

- IPv6 has a much larger address space; $\mathbf{2^{128}}$ 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 below shows the prefix for each type of address.

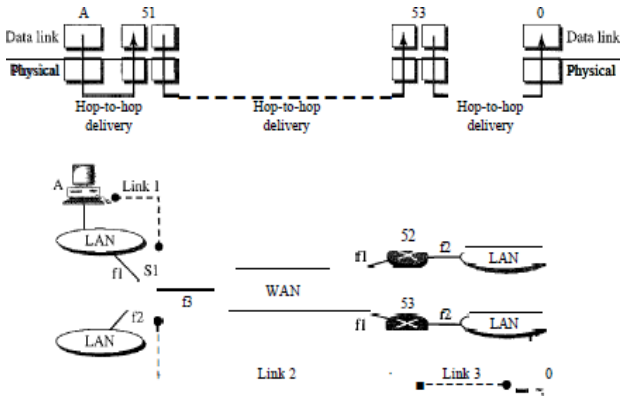
- The third column shows the fraction of each type of address relative to the whole address space.

Type Prefix	*Type*	*Fraction*
00000000	Reserved	1/256
00000001	Unassigned	1/256
0000001	ISO network addresses	1/128
0000010	IPX (Novell) network addresses	1/128
0000011	Unassigned	1/128
00001	Unassigned	1/32
0001	Reserved	*1/16*
001	Reserved	1/8
010	Provider-based unicast addresses	*1/8*

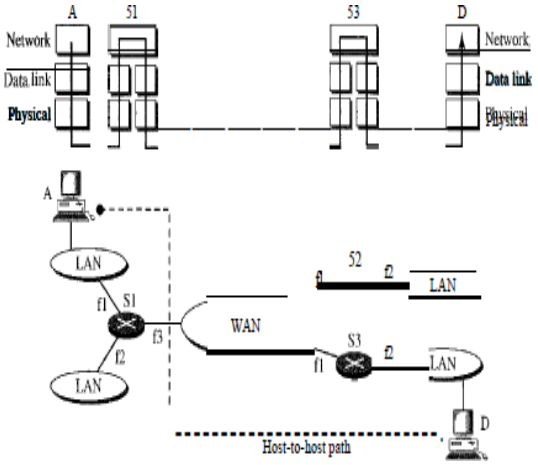
Type Prefix	*Type*	*Fraction*
011	Unassigned	1/8
100	Geographic-based unicast addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1116
11110	Unassigned	1132
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
11111110 a	Unassigned	1/512
1111 111010	Link local addresses	111024
1111 1110 11	Site local addresses	1/1024
11111111	Multicast addresses	1/256

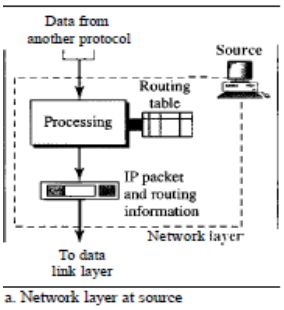
2. INTER NETWORKING

- Physical and data link layers are jointly responsible for data delivery on the network from one node to the next node. Consider following figure, Assume a packet is being sent to D from A

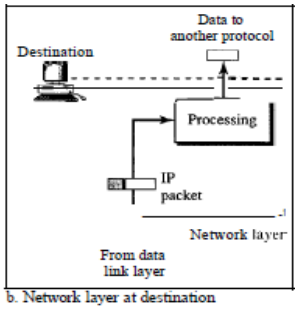


- How does interface of S3 know that the packet to be forwarded to f3
- This creates the necessity of network layer, which builds logical address in the packet, that gives routing information,

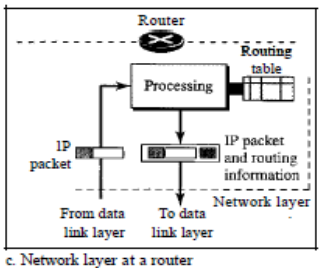




a. Network layer at source



b. Network layer at destination



c. Network layer at a router

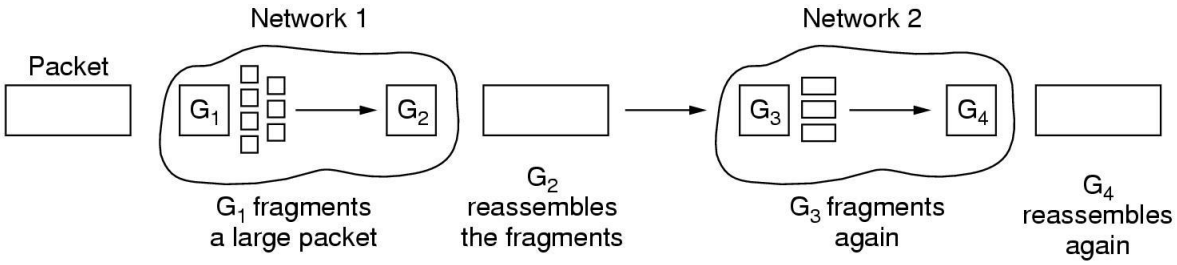
INTERNETWORKING AS DATAGRAM NETWORK

- The Internet, at the network layer, is a packet-switched network.
- In general, switching can be divided into three broad categories: circuit switching, packet switching, and message switching.
- Packet switching uses either the virtual circuit approach or the datagram approach.
- The Internet has chosen the datagram approach to switching in the network layer.
- It uses the universal addresses defined in the network layer to route packets from the source to the destination.

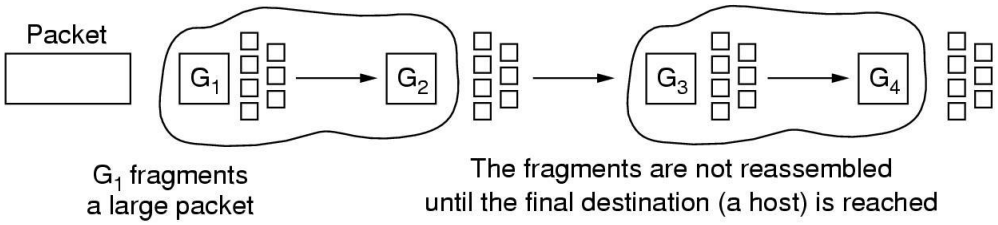
INTERNETWORKING: FRAGMENTATION

- ❑ Transparent fragmentation
    - o Strategy

- Gateway breaks large packet into fragments
- Each fragment addressed to same exit gateway
- Exit gateway does reassembly



(a)



(b)

  - Simple, but some problems
    - Gateway must know when it has all pieces
    - Performance loss: all fragments through same gateway
    - Overhead: repeatedly reassemble and refragment
  - Example: ATM segmentation

- Non transparent fragmentation
  - Strategy
    - Gateway breaks large packet into fragments
    - Each fragment is forwarded to destination
  - problems

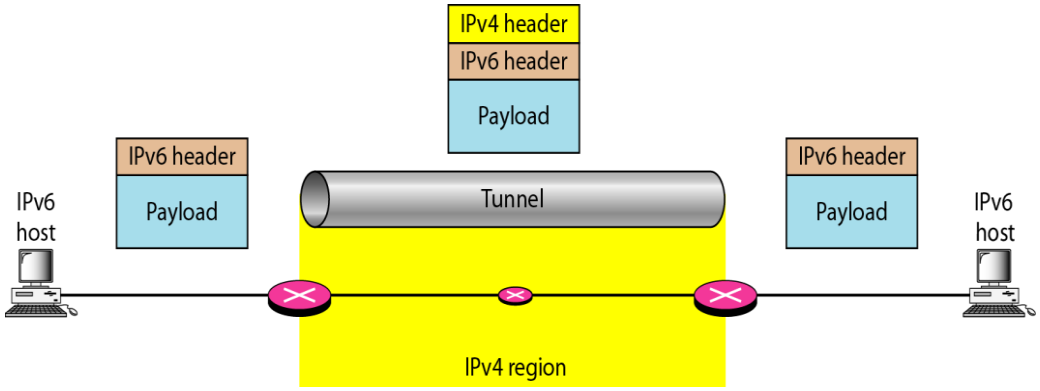
- Every host must be able to reassembly
- More headers

o Example: IP fragmentation

INTERNETWORKING AS CONNECTION LESS NETWORK

- In connection oriented circuit, there is a logical relationship between the packets, as they move in order across the network
- The order in which packets received is same as in order, how the source has emitted the packets.
- In conneetionless service, the network layer protocol treats each packet independently, with each packet having no relationship to any other packet.
- The packets in a message mayor may not travel the same path to their destination.
- This type of service is used in the datagram approach to packet switching. The Internet has chosen this type of service at the network layer.
- The reason for this decision is that the Internet is made of so many heterogeneous networks and it is not possible to create a connection from the source to the destination without knowing the nature of the networks.

3. TUNNELING:



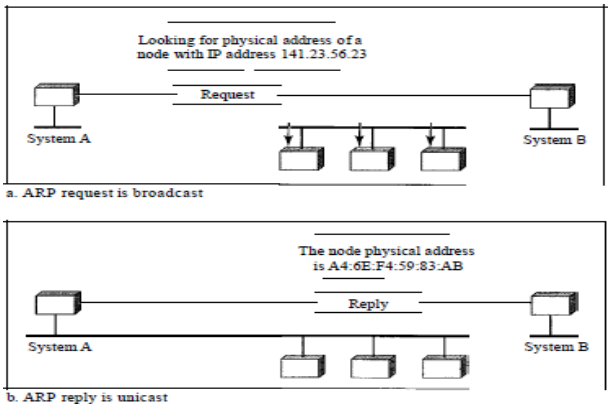
Tunneling is a strategy used when two computers using IPv6 want to communicate with each other and the packet must pass through a region that uses IPv4. To pass through this region, the packet must have an IPv4 address. So the IPv6 packet is encapsulated in an IPv4 packet when it enters the region, and it leaves its capsule when it exits the region. It seems as if the IPv6 packet goes through a tunnel at one end and emerges at the other end. To make it clear that the IPv4 packet is carrying an IPv6packet as data, the protocol value is set to 41. Tunneling is shown in above Figure.

4. TYPES OF ADDRESSES IN INTERNET (ARP)

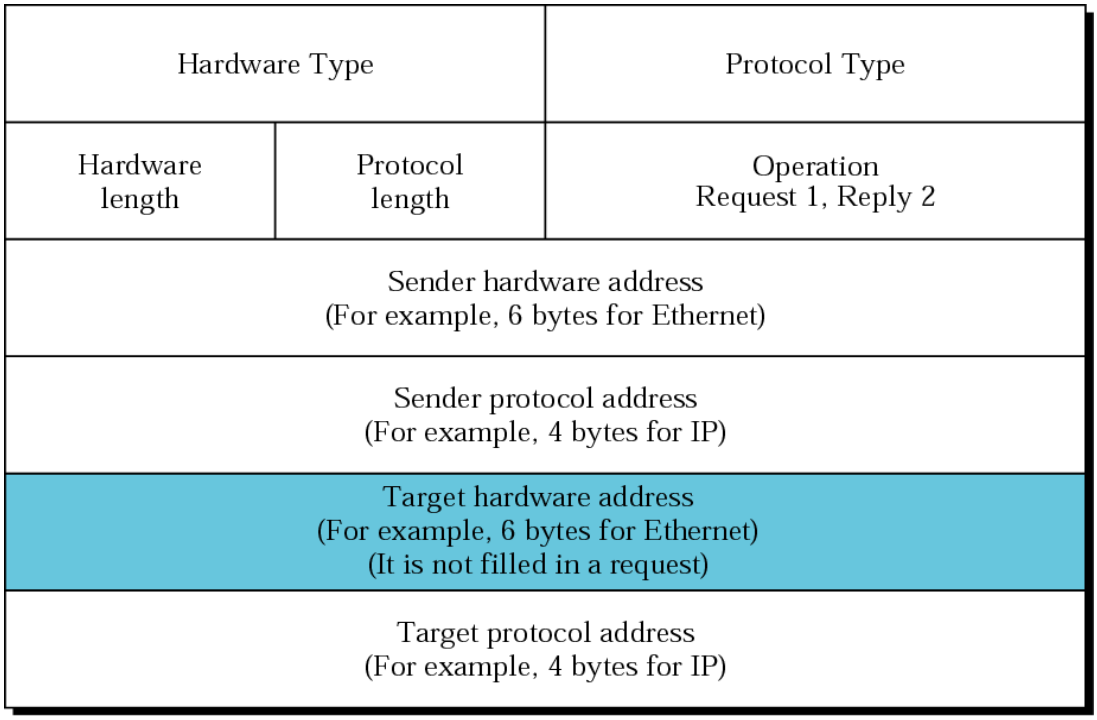
- Media Access Control (MAC) addresses in the network access layer
    - Associated w/ network interface card (NIC)
    - 48 bits or 64 bits
- IP addresses for the network layer
    - 32 bits for IPv4, and 128 bits for IPv6
    - E.g., 123.4.56.7
- IP addresses + ports for the transport layer
    - E.g., 123.4.56.7:80
- IP addresses are chosen by the local system administrator to suit the local network
- Ethernet addresses are built into the interface hardware by the manufacturer
- The two addresses bear absolutely no relationship to one another (as we would expect from the layering principles)

- Primarily ARP is used to translate IP addresses to Ethernet MAC addresses
    - The device drive for Ethernet NIC needs to do this to send a packet
- Suppose want to send a packet over (say) an Ethernet.
- We only know the destination's IP address to build the Ethernet frame we have to know the Ethernet address that the destination has.
- This is what ARP does: Find the hardware address corresponding to an IP address

MAPPING:



ARP PACKET:

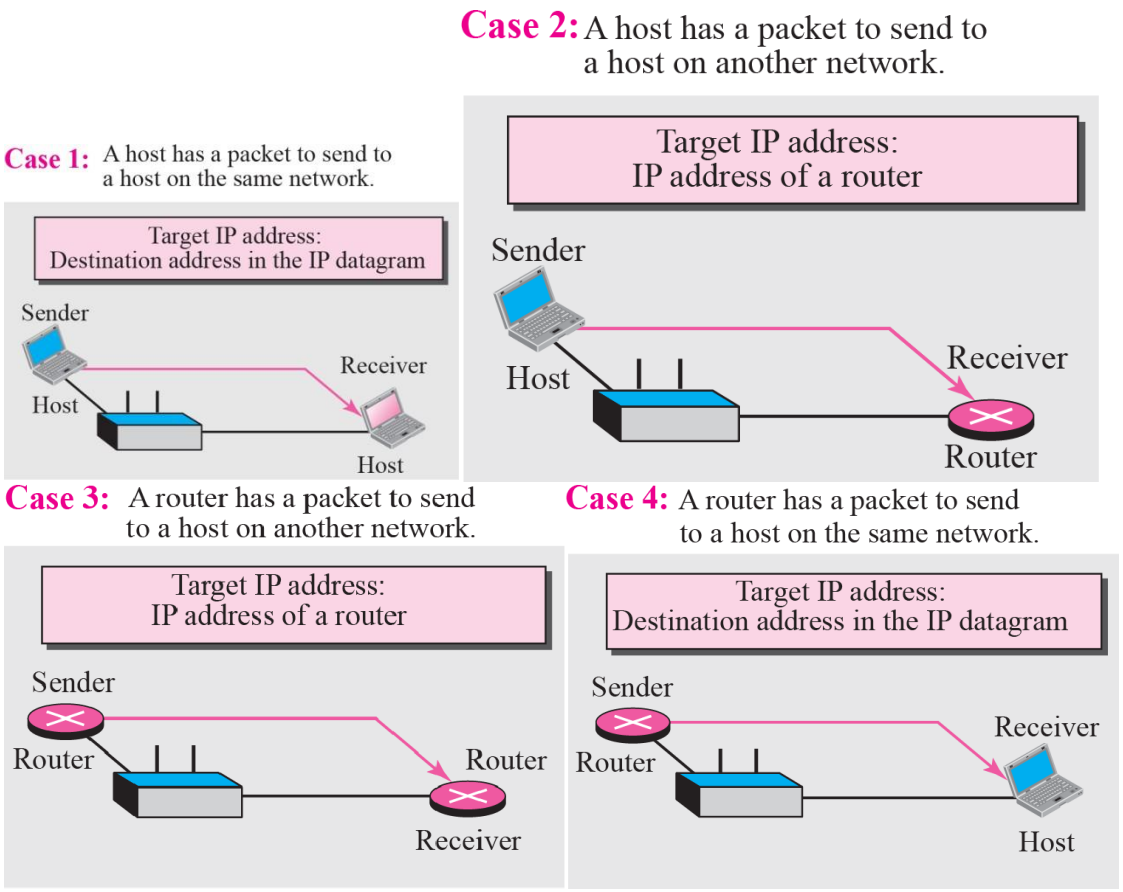


HEADER:

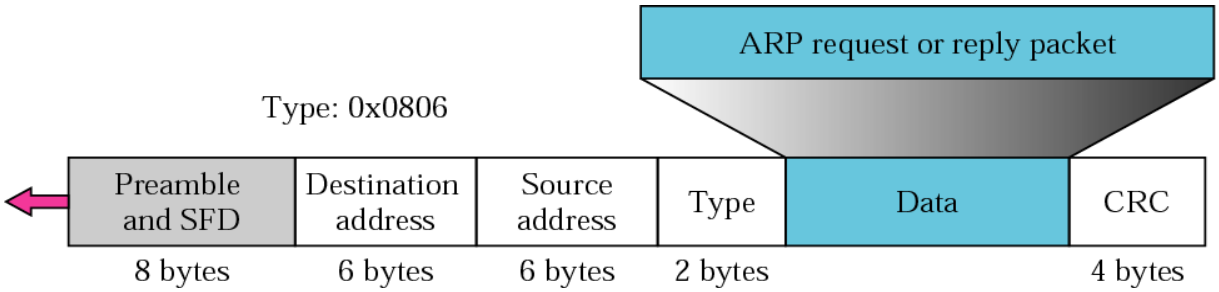
- **Hardware type.** This is a 16-bit field defining the type of the network on which ARP is running. Each LAN has been assigned an integer based on its type. For example, Ethernet is given type 1. ARP can be used on any physical network.
- **Protocol type**. This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 080016, ARP can be used with any higher-level protocol.
- **Hardware length.** This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.
- **Protocol length**. This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.
- Operation. This is a 16-bit field defining the type of packet. Two packet types are defined: ARP request (1) andARP reply (2).
- **Sender hardware address**. This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.

- **Sender protocol address**. This is a variable-length field defining the logical (for example, IP) address of the sender. For the IP protocol, this field is 4 bytes long.
- **Target hardware address**. This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long.
- For an ARP request message, this field is alIOs because the sender does not know the physical address of the target.
- **Target protocol address**. This is a variable-length field defining the logical (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long.

CASES OF ARP:



ARP ENCAPSULATION

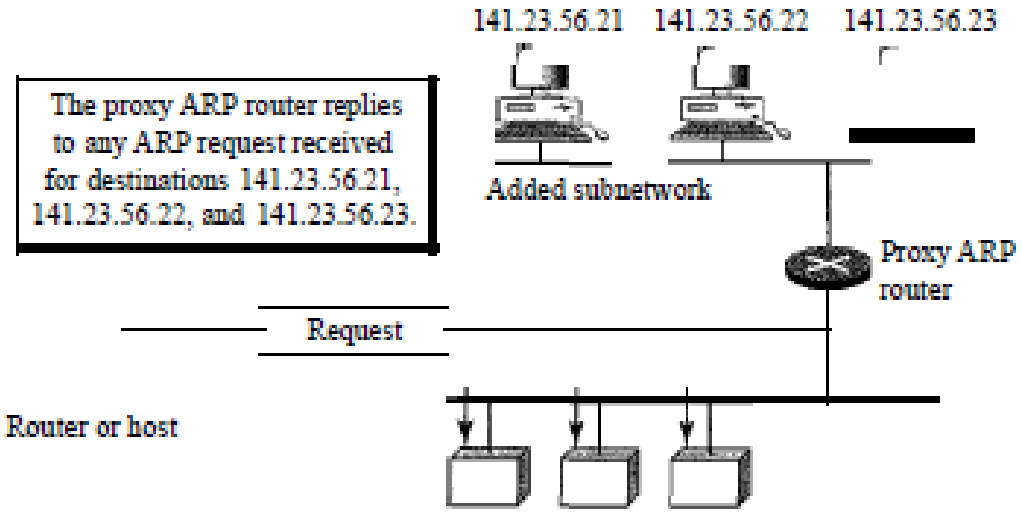


ARP CACHE:

- For every outgoing packet sending ARP request and waiting for responses is inefficient
- Requires more bandwidth
- Consumes Time
- ARP cache maintained at each node
- Size limit = 512 entries (timer)

PROXY ARP:

- A technique called proxy ARP is used to create a subnetting effect. A proxy ARP is an ARP that acts on behalf of a set of hosts.
- Whenever a router running a proxy ARP receives an ARP request looking for the IP address of one of these hosts, the router sends an ARP reply announcing its own hardware (physical) address.

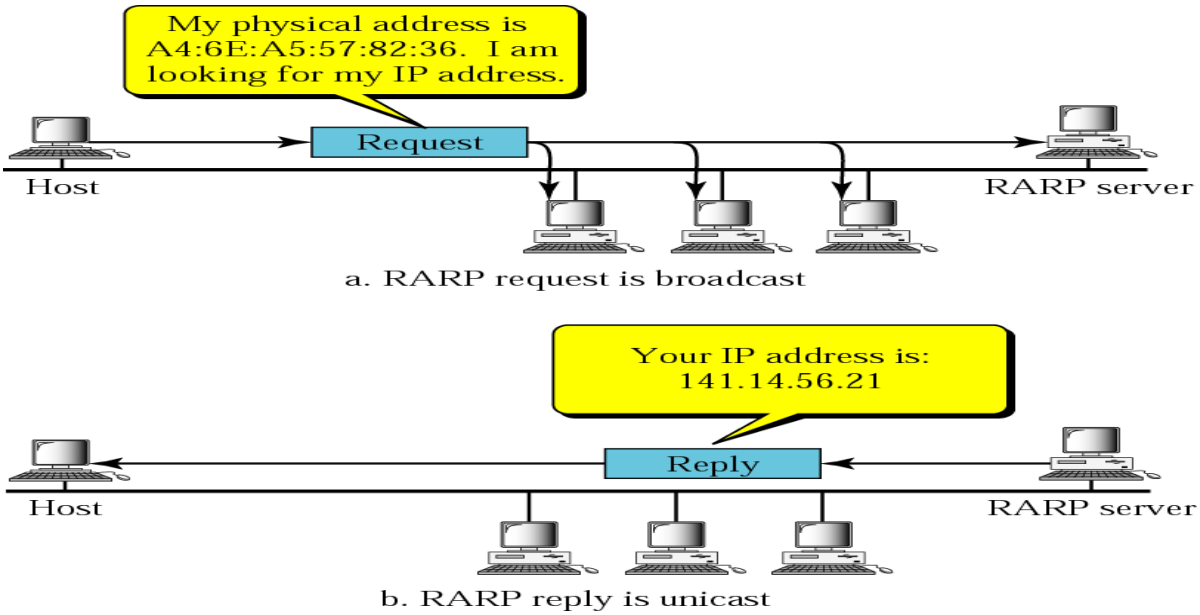


- However, the administrator may need to create a subnet without changing the whole system to recognize subnetted addresses.
- One solution is to add a router running a proxy ARP.
- In this case, the router acts on behalf of all the hosts installed on the subnet.
- When it receives an ARP request with a target IP address that matches the address of one of its members (141.23.56.21, 141.23.56.22, or 141.23.56.23), it sends an ARP reply and announces its hardware address as the target hardware address.
- When the router receives the IP packet, it sends the packet to the appropriate host.

RARP:

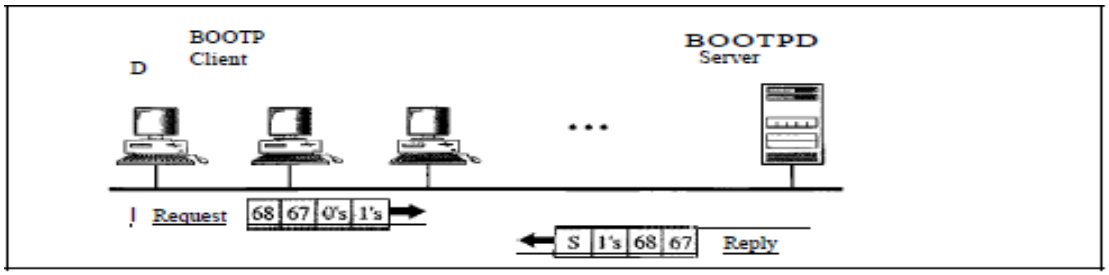
- Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the hardware
- To create an IP datagram, a host or a router needs to know its own IP address or addresses.
- The IP address of a machine is usually read from its configuration file stored on a disk file.
- However, a diskless machine is usually booted from ROM, which has minimum booting information.

- The ROM is installed by the manufacturer. It cannot include the IP address because the IP addresses on a network are assigned by the network administrator.
- The machine can get its physical address which is unique locally.
- It can then use the physical address to get the logical address by using the RARP protocol.
- RARP finds the logical address for a machine that only knows its physical address.
- The RARP request packets are broadcast;
- The RARP reply packets are unicast.

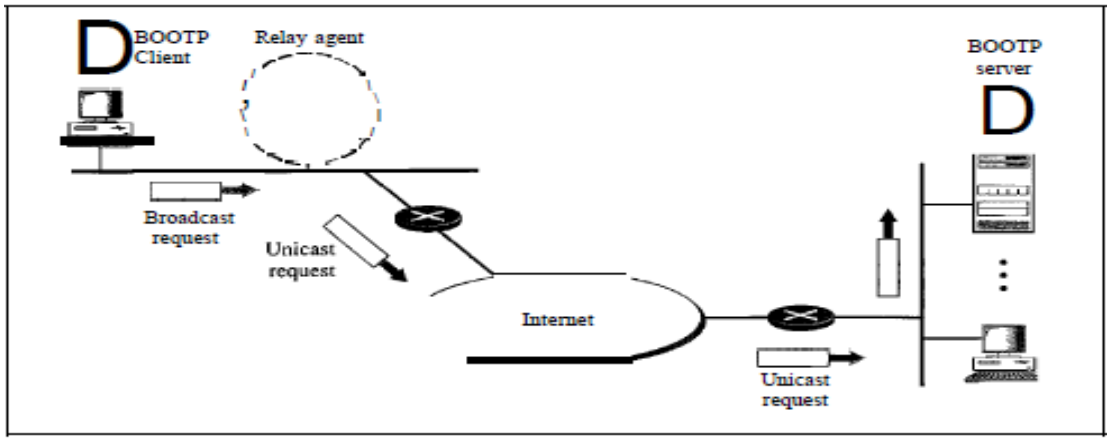


BOOTP:

- The Bootstrap Protocol (BOOTP) is an application layer, client/server protocol designed to provide physical address to logical address mapping.
- BOOTP messages are encapsulated in a UDP packet, and the UDP packet itself is encapsulated in an IP packet.
- The client simply uses all 0's as the source address and all 1's as the destination address.



a. Client and server on the same network



b. Client and server on different networks

- One of the advantages of BOOTP over RARP is that the client and server are application-layer processes.
- The BOOTP request is broadcast because the client does not know the IP address of the server. A broadcast IP datagram cannot pass through any router.
- To solve the problem, there is a need for an intermediary. One of the hosts can be used as a relay.
- The host in this case is called a relay agent. The relay agent knows the unicast address of a BOOTP server. When it receives this type of packet, it encapsulates the message in a unicast datagram and sends the request to the BOOTP server.
- The packet carrying a unicast destination address, is routed by any router and reaches the BOOTP server.
- The BOOTP server knows the message comes from a relay agent because one of the fields in the request message defines the IP address of the relay agent.

- The relay agent, after receiving the reply, sends it to the BOOTP client.

DHCP:

- Dynamic Host Configuration Protocol
- It is a method for assigning Internet Protocol (IP) addresses permanently or to individual computers in an organization's network
- DHCP lets a network administrator supervise and distribute IP addresses from a central point and automatically sends a new IP address when a computer is plugged into a different place in the network

Types of IP Addresses

- DHCP is used to assign IP addresses to hosts or workstations on the network
- Two types of IP addresses:
    - ✓ Static
        - ❖ Is a number that is assigned to a computer by an Internet service provider (ISP) to be its permanent address on the Internet
    - ✓ Dynamic
        - ❖ The temporary IP address is called a dynamic IP address

Importance of DHCP:

- Important when it comes to adding a machine to a network
- When computer requests an address, the administrator would have to manually configure the machine
    - ✓ Mistakes are easily made
    - ✓ Causes difficulty for both administrator as well as neighbors on the network

- DHCP solves all the hassle of manually adding a machine to a network.

WORKING OF DHCP

- When a client needs to start up TCP/IP operations, it broadcasts a request for address information.
- The DHCP server will not reallocate the address during the lease period and will attempt to return the same address every time the client requests an address.
- The client can extend its lease or send a message to the server before the lease expires it that it no longer needs the address so it can be released and assigned to another client on the network.

Advantages of DHCP

- DHCP minimizes the administrative burden
- By using DHCP there is no chance to conflict IP address
- By using DHCP relay agent you provide IP address to another network

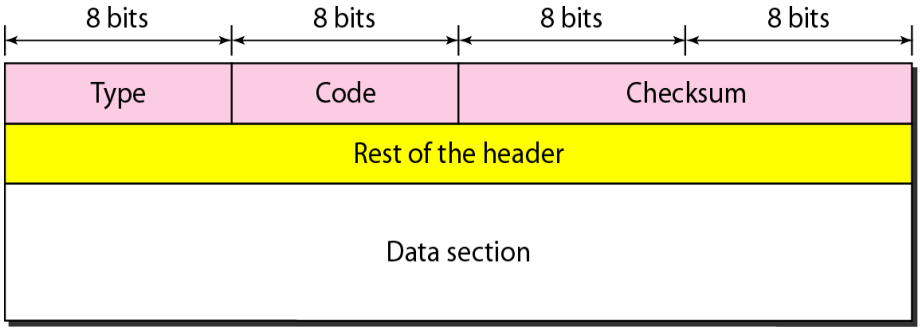
Disadvantages of DHCP

- When DHCP server is unavailable, client is unable to access enterprises network.
- Your machine name does not change when you get a new IP address.

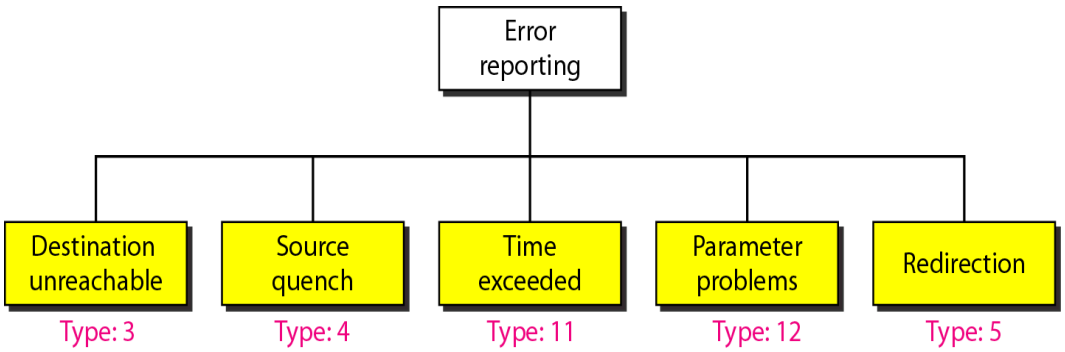
5. INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

- IP is an unreliable method for delivery of network data.
- It has no built-in processes to ensure that data is delivered in the event that problems exist with network communication.
- If an intermediary device such as a router fails, or if a destination device is disconnected from the network, data cannot be delivered.
- Additionally, nothing in its basic design allows IP to notify the sender that a data transmission has failed.

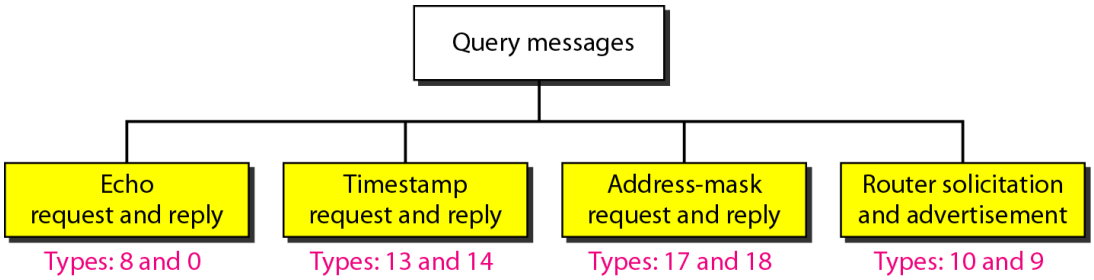
- Internet Control Message Protocol (ICMP) is the component of the TCP/IP protocol stack that addresses this basic limitation of IP.
- ICMP does not overcome the unreliability issues in IP.
- Reliability must be provided by upper layer protocols if it is needed.
- ICMP is an error reporting protocol for IP.
- When datagram delivery errors occur, ICMP is used to report these errors back to the source of the datagram.
- ICMP does not correct the encountered network problem; it merely reports the problem.
- ICMP reports on the status of the delivered packet only to the source device.
- It does not propagate information about network changes to routers.
- The general format of ICMP is given below



ERROR REPORTING MESSAGES



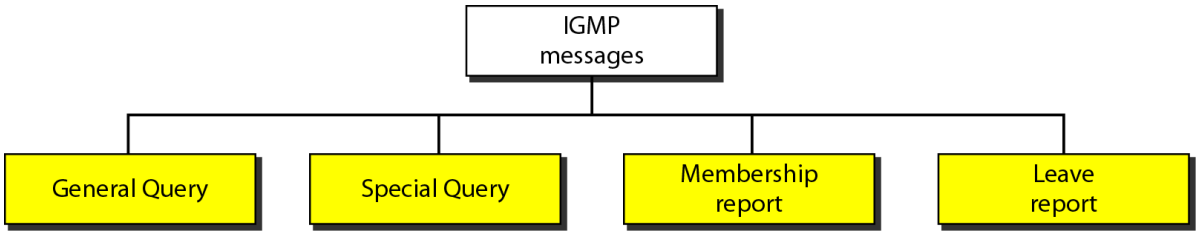
QUERY MESSAGES



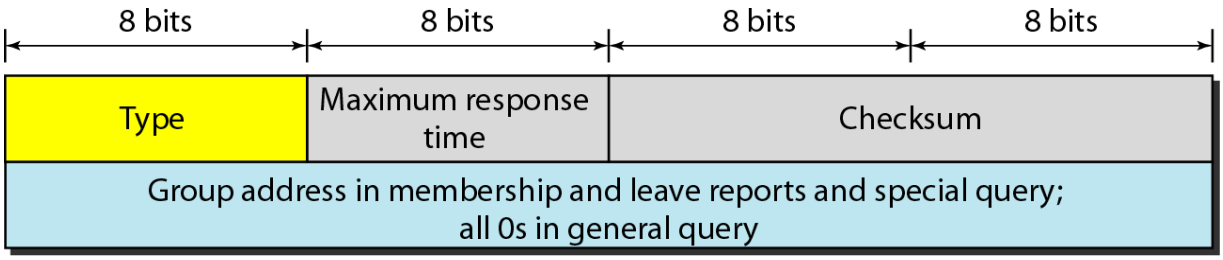
Internet Group Management Protocol (IGMP)

- The IP protocol can be involved in two types of communication: unicasting and multicasting.
- The Internet Group Management Protocol (IGMP) is one of the necessary, but not sufficient, protocols that is involved in multicasting.
- IGMP is a companion to the IP protocol.
- IGMP is not a multicasting routing protocol; it is a protocol that manages group membership.
- The IGMP protocol gives the multicast routers information about the membership status of hosts (routers) connected to the network
- A multicast router may receive thousands of multicast packets every day for different groups. If a router has no knowledge about the membership status of the hosts, it must broadcast all these packets.
- This creates a lot of traffic and consumes bandwidth.
- A better solution is to keep a list of groups in the network for which there is at least one loyal member.
- IGMP helps the multicast router create and update this list.

IGMP MESSAGES:



IGMP MESSAGE FORMAT:



- Type: This 8-bit field defines the type of message, as shown in Table.

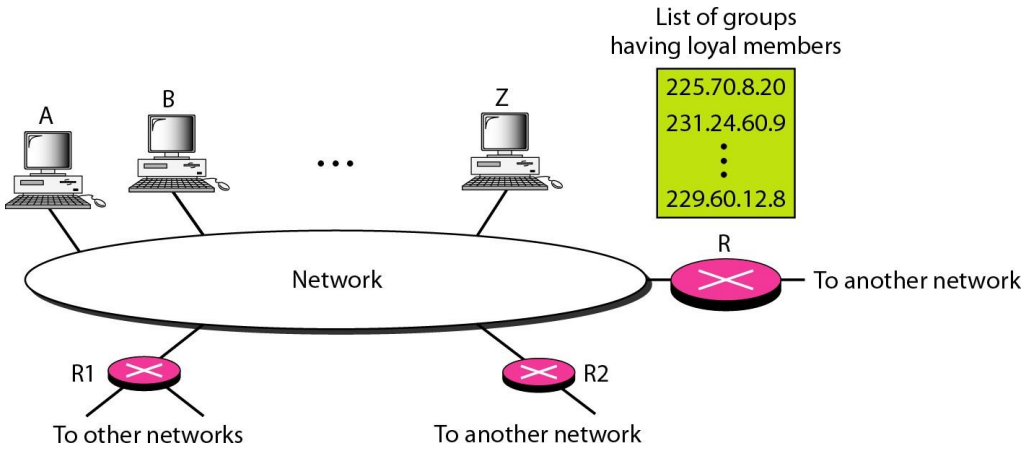
Type	*Value*
General or special query	0x11 or 00010001
Membership report	0x16 or 00010110
Leave report	0x17 or 00010111

- **Maximum Response Time**: This 8-bit field defines the amount of time in which a query must be answered. The value is in tenths of a second.
- **Checksum**: This is a 16-bit field carrying the checksum. The checksum is calculated over the 8-byte message.

- **Group address**: The value of this field is 0 for a general query message. The value defines the groupid (multicast address of the group) in the special query, the membership report, and the leave report messages.

IGMP OPERATION

- A multicast router connected to a network has a list of multicast addresses of the groups with at least one loyal member in that network



- For each group, there is one router that has the duty of distributing the multicast packets destined for that group.

Joining a Group:

A host or a router can join a group. A host maintains a list of processes that have membership in a group. When a process wants to join a new group, it sends its request to the host. The host adds the name of the process and the name of the requested group to its list.

- If this is the first entry for this particular group, the host sends a membership report message. If this is not the first entry, there is no need to send the membership report since the host is already a member of the group; it already receives multicast packets for this group.
- The protocol requires that the membership report be sent twice, one after the other within a few moments. In this way, if the first one is lost or damaged, the second one replaces it.

Leaving a Group:

When a host sees that no process is interested in a specific group, it sends a leave report. Similarly, when a router sees that none of the networks connected to its interfaces is interested in a specific group, it sends a leave report about that group.

- However, when a multicast router receives a leave report, it cannot immediately purge that group from its list because the report comes from just one host or router; there may be other hosts or routers that are still interested in that group.
- To make sure, the router sends a special query message and inserts the groupid, or multicast address, related to the group.

The router allows a specified time for any host or router to respond. If, during this time, no interest is received, the router assumes that there are no loyal members in the network and purges the group from its list

Monitoring membership:

A host or router can join a group by sending a membership report message. It can leave a group by sending a leave report message.

- What happens when a system shuts down suddenly?
- To handle this, router periodically (by default, every 125 s) sends a general query message.
    - In this message, the group address field is set to 0.0.0.0.
    - This means the query for membership continuation is for all groups in which a host is involved, not just one.

The router expects response for maxiumum time of 10 s. A system can respond by sending member id for continuing in group

Delayed Response:

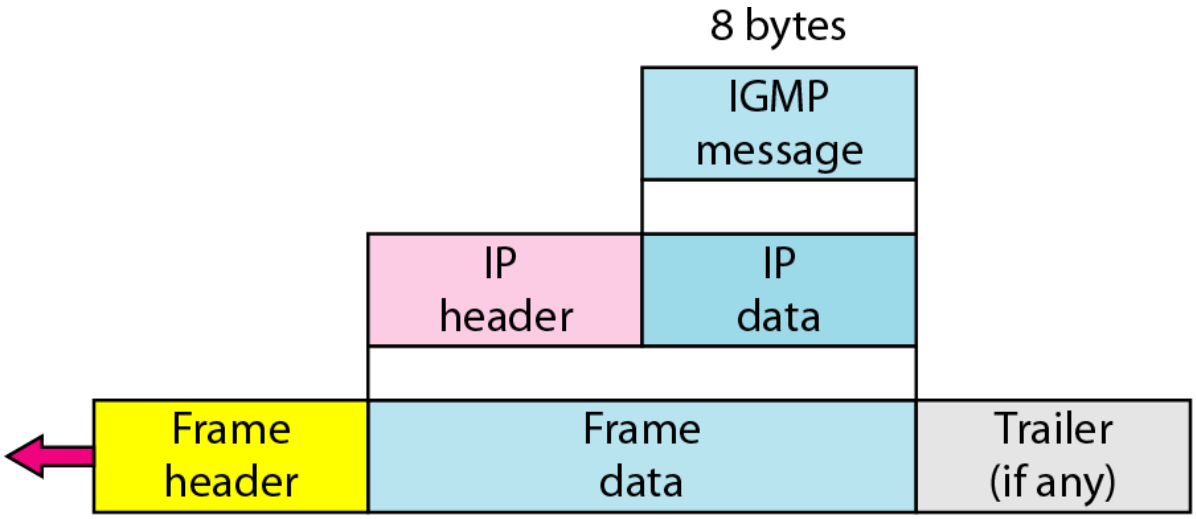
To prevent unnecessary traffic, IGMP uses a delayed response strategy.

- When a host or router receives a query message, it does not respond immediately; it delays the response.
- Each host or router uses a random number to create a timer, which expires between 1 and l0s. The expiration time can be in steps of I s or less.
- Query Router Query messages may create a lot of responses.
- To prevent unnecessary traffic, IGMP designates one router as the query router for each network.
- Only this designated router sends the query message, and the other routers are passive (they receive responses and update their lists).

ENCAPSULATION

NETWORK LAYER

- The IGMP message is encapsulated in an IP datagram, which is itself encapsulated in a frame. See Figure below.



- The value of the protocol field is 2 for the IGMP protocol. Every IP packet carrying this value in its protocol field has data delivered to the IGMP protocol. When the message is encapsulated in the IP datagram, the value of TTL must be 1.

- No IGMP message must travel beyond the LAN. A TTL value of 1 guarantees that the message does not leave the LAN since this value is decremented to 0 by the next router and, consequently, the packet is discarded.
- Table below shows the destination IP address for each type of message

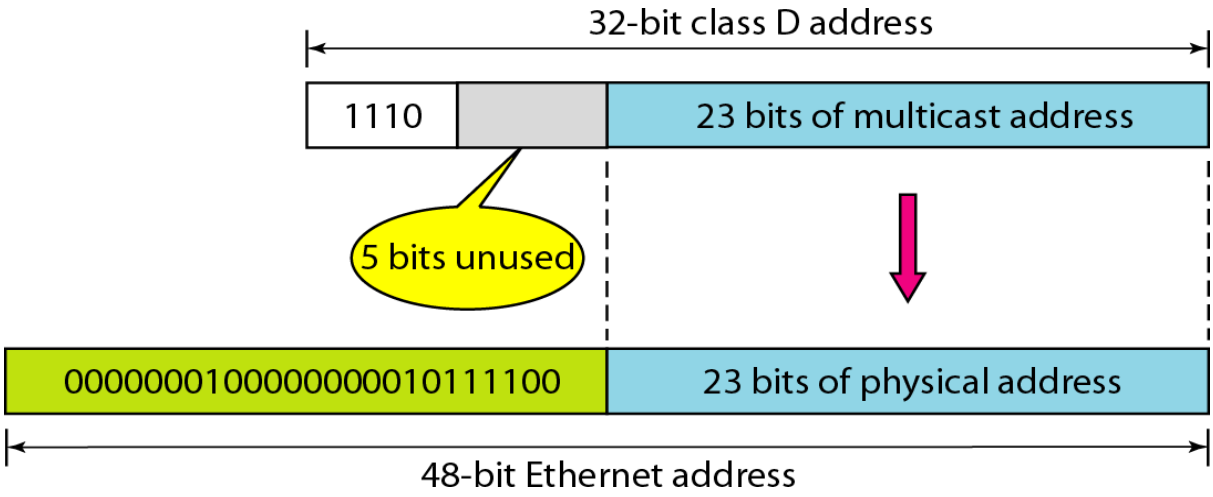
Type	*IP Destination Address*
Query	224.0.0.1 All systems on this subnet
Membership report	The multicast address of the group
Leave report	224.0.0.2 All routers on this subnet

- A query message is multicast by using the multicast address 224.0.0.1. All hosts and all routers will receive the message.
- A membership report is multicast using a destination address equal to the multicast address being reported (groupid). Every station that receives the packet can immediately determine (the group for which a report has been sent.
- A leave report message is multicast using the multicast address 224.0.0.2. So that routers receive this type of message. Hosts receive this message too, but disregard it

DATALINK LAYER

- Most LANs support physical multicast addressing including ethernet.
- An Ethernet physical address (MAC address) is six octets (48 bits) long. If the first 25 bits in an Ethernet address are 00000001000000000010111100, this identifies a physical multicast address for the TCP/IP protocol.
- The remaining 23 bits can be used to define a group. To convert an IP multicast address into an Ethernet address, the multicast router extracts the least significant 23 bits of a class

D IP address and inserts them into a multicast Ethernet physical address (shown in below fig).



- However, the group identifier of a class D IP address is 28 bits long, which implies that 5 bits is not used.
- This means that 32 (25) multicast addresses at the IP level are mapped to a single multicast address.
- In other words, the mapping is many-to-one instead of one-to-one. If the 5 leftmost bits of the group identifier of a class D address are not all zeros, a host may receive packets that do not really belong to the group in which it is involved.
- Thus, the host must check the IP address and discard any packets that do not belong to it.
- Other LANs support the same concept but have different methods of mapping.

FORWARDING:

Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.

Next-Hop Method Versus Route Method

- One of the techniques that reduces the contents of a routing table is next-hop method.
- In this technique, the routing table holds only the address of the next hop instead of information about the complete route (route method).
- The entries of a routing table must be consistent with one another. Below fig. shows how routing tables.

a. Routing tables based on route

Destination	Route
Host B	R1, R2, host B

Routing table for host A

Destination	Route
Host B	R2, host B

Routing table for R1

Destination	Route
Host B	Host B

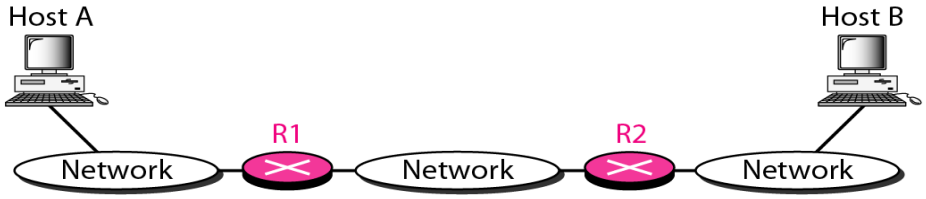
Routing table for R2

b. Routing tables based on next hop

Destination	Next hop
Host B	R1

Destination	Next hop
Host B	R2

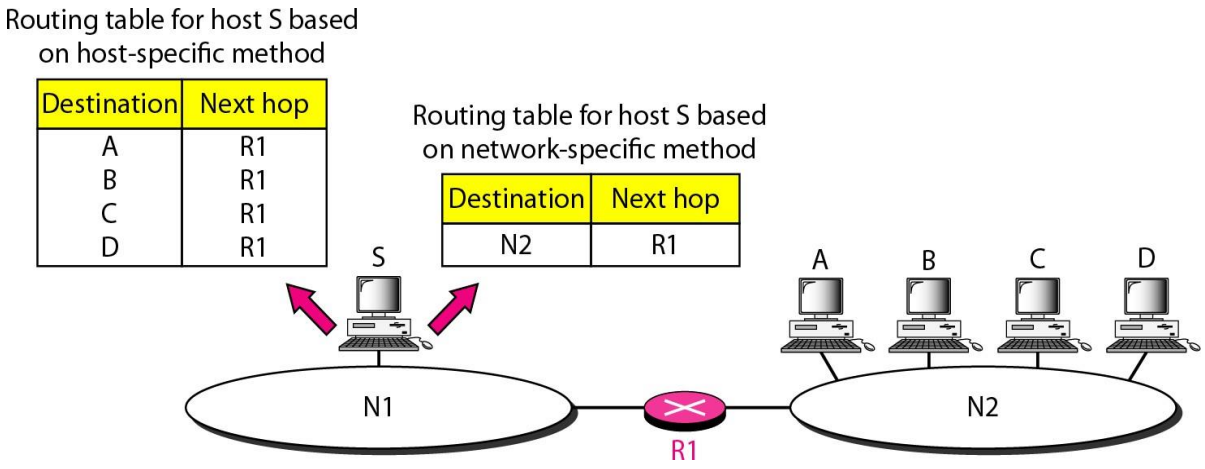
Destination	Next hop
Host B	---



Network Specific Method Vs Host Specific Method:

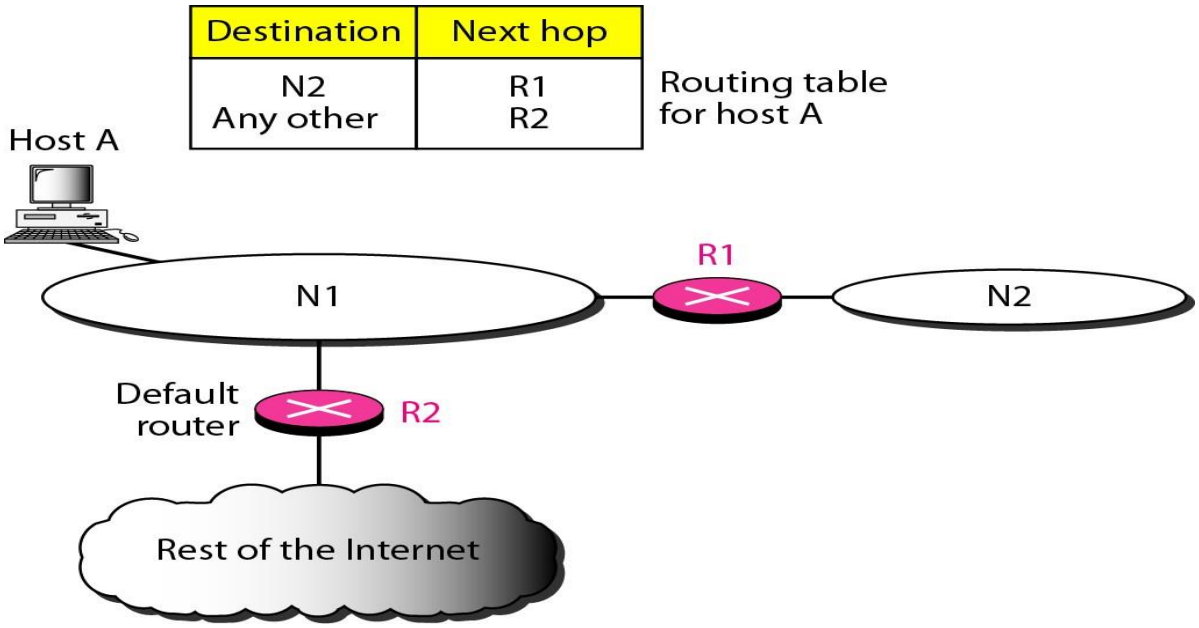
- A second technique to reduce the routing table is called the network-specific method.
- Here, instead of having an entry for every destination host connected to the same physical network (host-specific method), we have only one entry that defines the address of the destination network itself. Below fig. shows the concept.

Host-specific routing is used for purposes such as checking the route or providing security measures



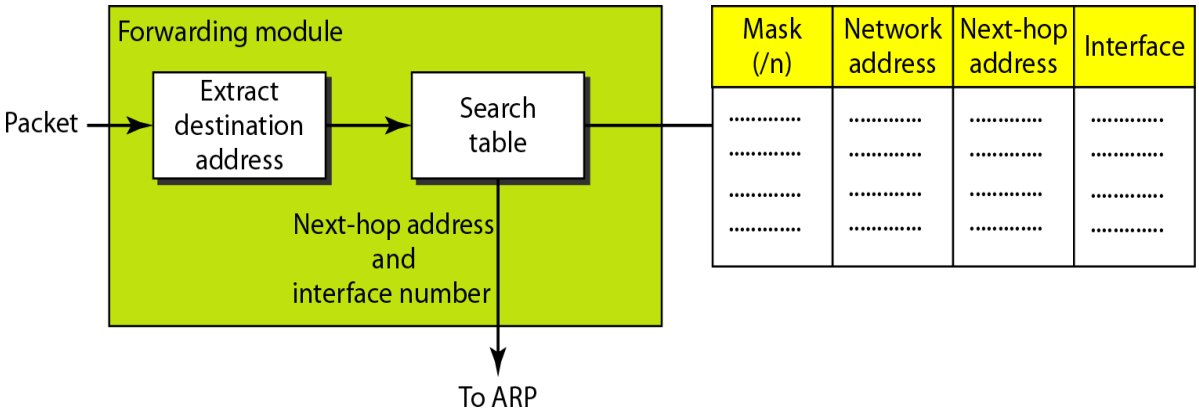
Default method:

- The next technique to simplify routing is called the default method.
- In below fig., host A is connected to a network with two routers.
- Router Rl routes the packets to hosts connected to network N2. However, for the rest of the Internet, router R2 is used.
- So instead of listing all networks in the entire Internet, host A can just have one entry called the default (normally defined as network address 0.0.0.0).



FORWARDING PROCESS:

- Assume that hosts and routers use classless addressing.
- In classless addressing, the routing table needs to have one row of information for each block involved.
- The table needs to be searched based on the network address. Unfortunately, the destination address in the packet gives no clue about the network address.
- To solve the problem, we need to include the mask (/n) in the table; we need to have an extra column that includes the mask for the corresponding block. Below fig. shows a simple forwarding module for classless addressing.



ROUTING TABLE:

- A host or a router has a routing table with an entry for each destination, or a combination of destinations, to route IP packets.

The routing table can be either static or dynamic

STATIC ROUTING TABLE

- A static routing table contains information entered manually. The administrator enters the route for each destination into the table. When a table is created, it cannot update automatically when there is a change in the Internet.
- The table must be manually altered by the administrator.

A static routing table can be used in a small internet that does not change very often, or in an experimental internet for troubleshooting. It is poor strategy to use a static routing table in a big internet such as the Internet

DYNAMIC ROUTING TABLE

- A dynamic routing table is updated periodically by using one of the dynamic routing protocols such as RIP, OSPF, or BGP.
- Whenever there is a change in the Internet, such as a shutdown of a router or breaking of a link, the dynamic routing protocols update all the tables in the routers (and eventually in the host) automatically.

- The routers in a big internet such as the Internet need to be updated dynamically for efficient delivery of the IP packets.

TABLE FORMAT:

- The routing table for classless addressing has a minimum of four columns.
- However, some of today's routers have even more columns.
- The number of columns is vendor-dependent, and not all columns can be found in all routers. Below fig. shows some common fields in today's routers.

Mask	Network address	Next-hop address	Interface	Flags	Reference count	Use
.....	.....	.....	.....	.....	.....	.....

- **Mask:** This field defines the mask applied for the entry.
- **Network address:** This field defines the network address to which the packet is finally delivered. In the case of host-specific routing, this field defines the address of the destination host.
- **Next-hop address:** This field defines the address of the next-hop router to which the packet is delivered
- **Interface:** This field shows the name of the interface.
- **Flags:** This field defines up to five flags. Flags are on/off switches that signify either presence or absence. The five flags are U (up), G (gateway), H (host-specific), D (added by redirection), and M (modified by redirection).
- **U (up):** The U flag indicates the router is up and running. Ifthis flag is not present, it means that the router is down. The packet cannot be forwarded and is discarded.

- **G (gateway):** The G flag means that the destination is in another network. The packet is delivered to the next-hop router for delivery (indirect delivery). When this flag is missing, it means the destination is in this network (direct delivery).
- **H (host-specific):** The H flag indicates that the entry in the network address field is a host-specific address. When it is missing, it means that the address is only the network address of the destination.
- **D (added by redirection):** The D flag indicates that routing information for this destination has been added to the host routing table by a redirection message from ICMP.
- **M (modified by redirection):** The M flag indicates that the routing information for this destination has been modified by a redirection message from ICMP.

Reference count: This field gives the number of users of this route at the moment. For example, if five people at the same time are connecting to the same host from this router, the value of this column is 5.

Use: This field shows the number of packets transmitted through this router for the corresponding destination.

22-3 UNICAST ROUTING PROTOCOLS

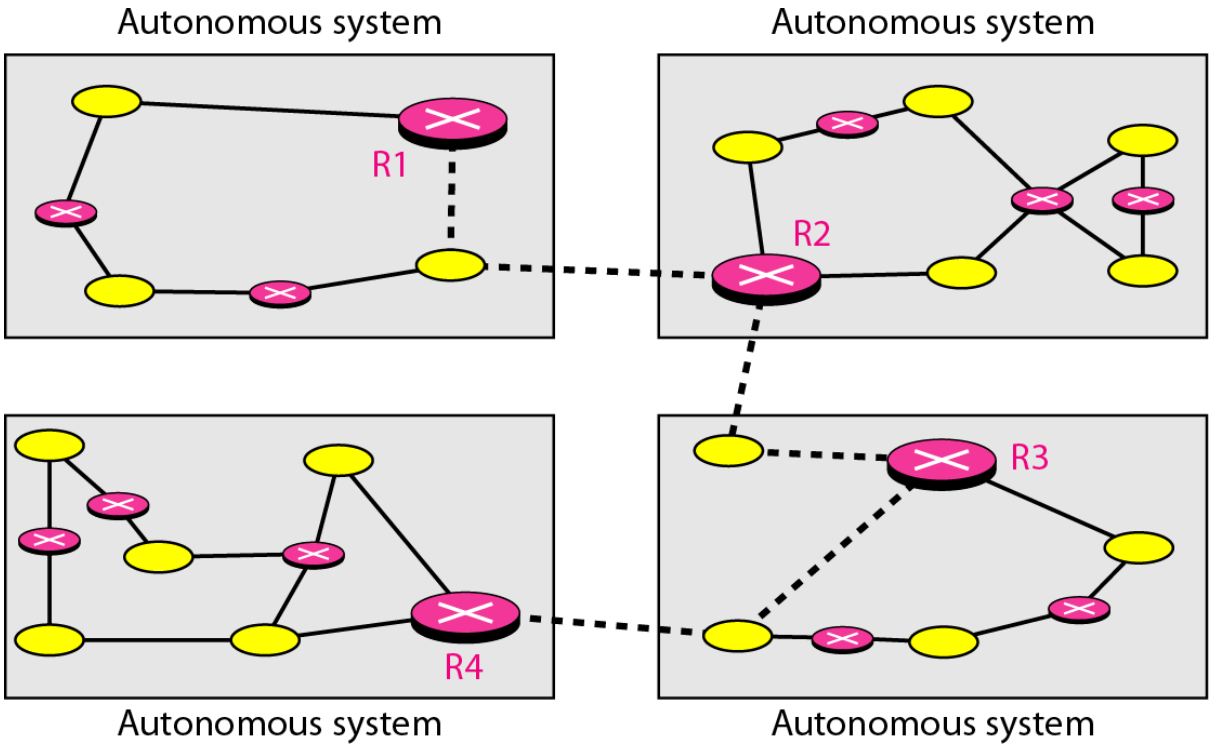
A routing table can be either static or dynamic. A static table is one with manual entries. A dynamic table is one that is updated automatically when there is a change somewhere in the Internet. A routing protocol is a combination of rules and procedures that lets routers in the Internet inform each other of changes.

INTRA- AND INTERDOMAIN ROUTING

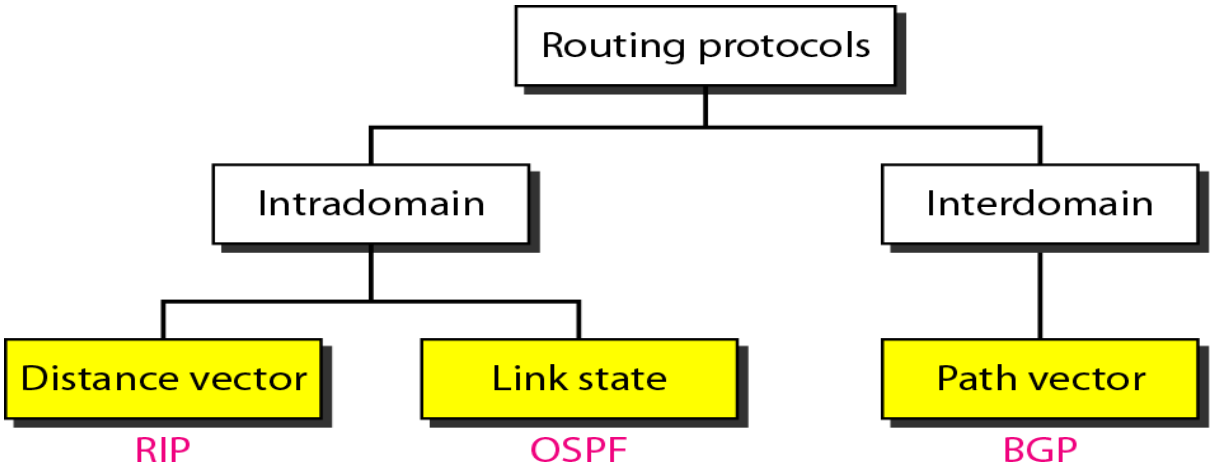
- Today, an internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers. For this reason, an internet is divided into autonomous systems.
- An autonomous system (AS) is a group of networks and routers

- under the authority of a single administration. Routing inside an autonomous system is referred to as intradomain routing.
- Routing between autonomous systems is referred to as interdomain routing.
- Each autonomous system can choose one or more intradomain
- routing protocols to handle routing inside the autonomous system.
- However, only one interdomain routing protocol handles routing between autonomous systems.

Figure 22.12 ***Autonomous systems***

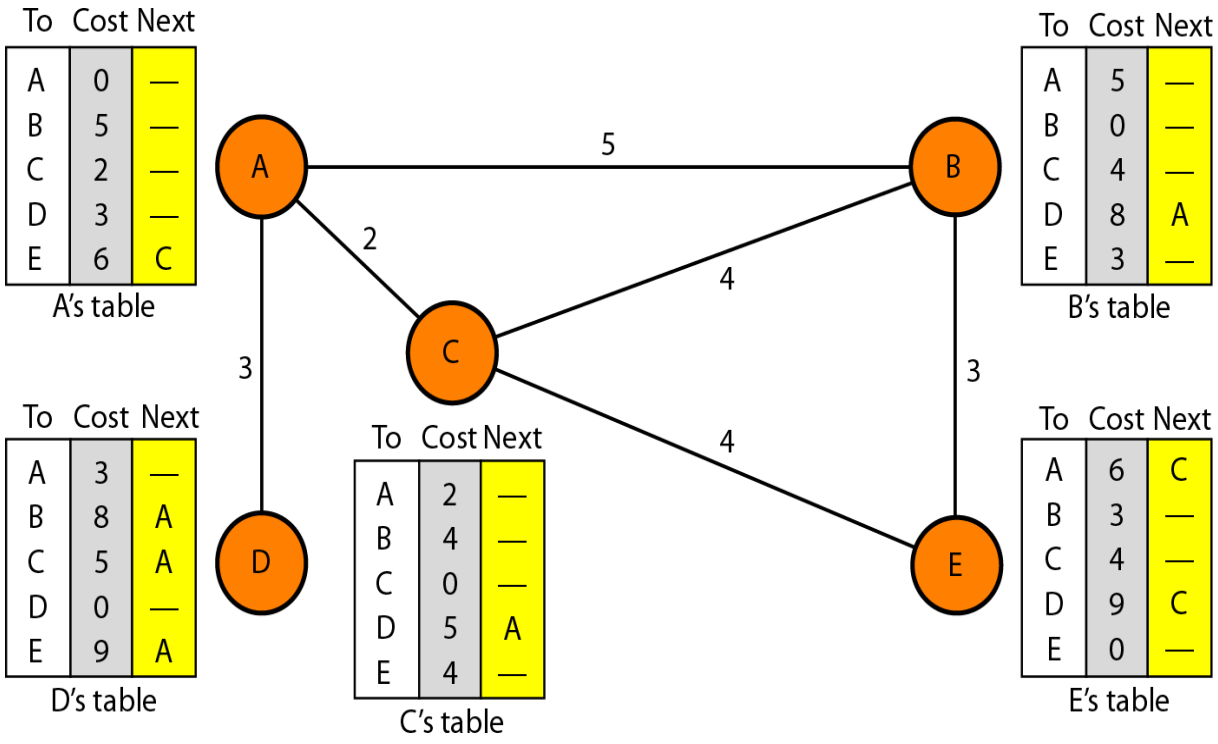


- Several intradomain and interdomain routing protocols are in use. Here, we discuss two intradomain routing protocols and one interdomain routing protocol
- Routing Information Protocol (RIP) is an implementation of the distance vector protocol.
- Open Shortest Path First (OSPF) is an implementation of the link state protocol.
- Border Gateway Protocol (BGP) is an implementation of the path vector protocol.



DISTANCE VECTOR RIP:

- In distance vector routing, the least-cost route between any two nodes is the route with minimum distance.
- In this protocol, each node maintains a vector (table) of minimum distances to every node.
- The table at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing)



INITIALIZATION:

- The tables in below fig. are stable; each node knows how to reach any other node and the cost.
- Each node can know only the distance between itself and its immediate neighbors, those directly connected to it.
- Assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors.
- Below fig. shows the initial tables for each node. The distance for any entry that is not a neighbor is marked as infinite (unreachable).

SHARING:

- The whole idea of distance vector routing is the sharing of information between neighbors.

- Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E.
- On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D.
- Now the question is how much of the table must be shared with each neighbor? A node is not aware of a neighbor's table.
- The best solution is **send its entire table to the neighbor** and let the neighbor decide what part to use and what part to discard.
- Here, the column next stop of table is not useful for the neighbor.
- Thus a node therefore can send only the first two columns of its table to any neighbor.
- For better understanding, sharing here means sharing only the first two columns.

UPDATING:

- When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

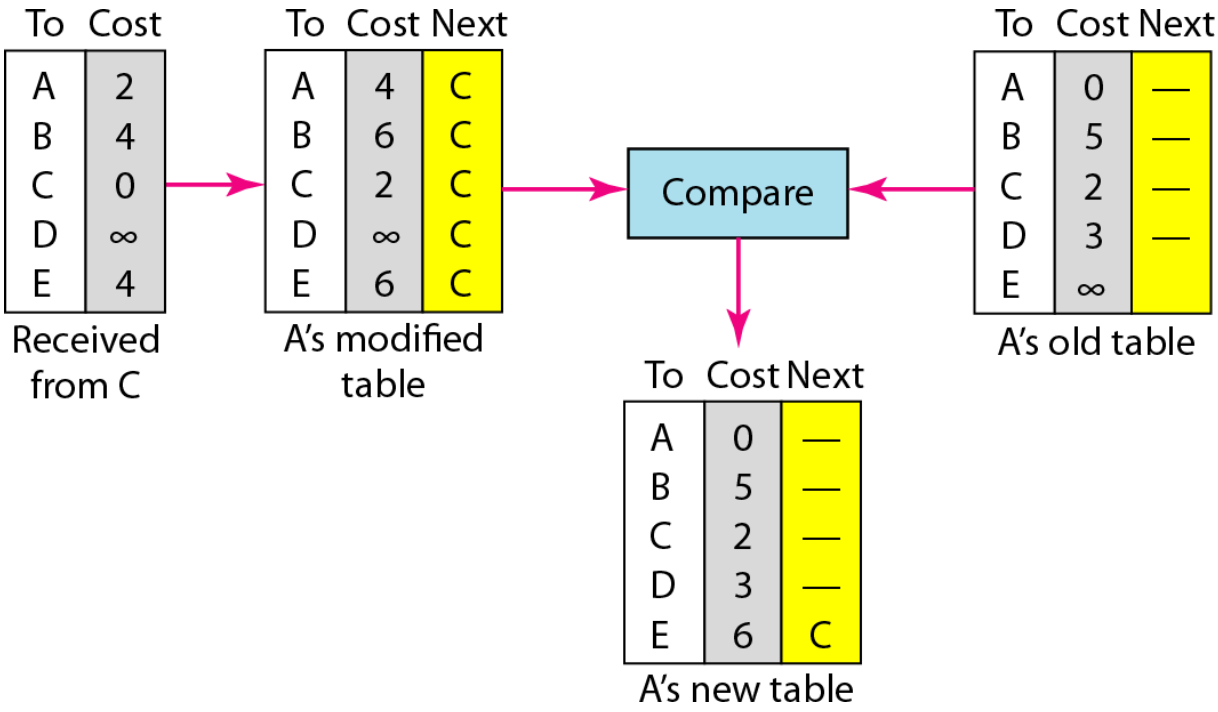
1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. The logic is clear. If node C claims that its distance to a destination is x mi, and the distance between A and C is y mi, then the distance between A and that destination, via C, is x + y mi.

2. The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.

3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.

b) If the next-node entry is the same, the receiving node chooses the new row. For example, suppose node C has previously advertised a route to node X with distance. Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist any more. The new route has a distance of infinity.



PROBLEMS TO BE ADDRESSED

- There are **several points** that are needed to be addressed here.
- **First,** as we know from mathematics, when we add any number to infinity, the result is still infinity.
- **Second,** the modified table shows how to reach A from A via C. If A needs to reach itself via C, it needs to go to C and come back, a distance of 4.

- Third, the only benefit from this updating of node A is the last entry, how to reach E. Previously, node A did not know how to reach E (distance of infinity); now it knows that the cost is 6 via C.
- Each node can update its table by using the tables received from other nodes.
- In a short time, if there is no change in the network itself, such as a failure in a link, each node reaches a stable condition in which the contents of its table remains the same.

When to share

- The question here is, When does a node should send its partial routing table to all its immediate neighbors? The table is sent both periodically and when there is a change in the table.
- **Periodic Update** A node sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing.
- **Triggered Update** A node sends its two-column routing table to its neighbors anytime there is a change in its routing table. This is called a triggered update. The change can result from the following.

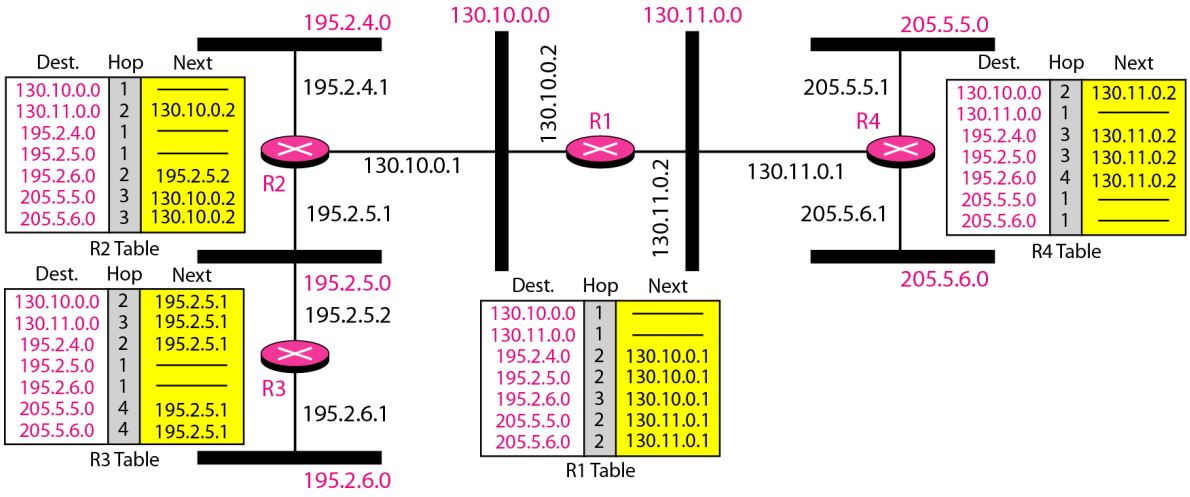
1. A node receives a table from a neighbor, resulting in changes in its own table after updating.

2. A node detects some failure in the neighboring links which results in a distance change to infinity.

ROUTING INFORMATION PROTOCOL

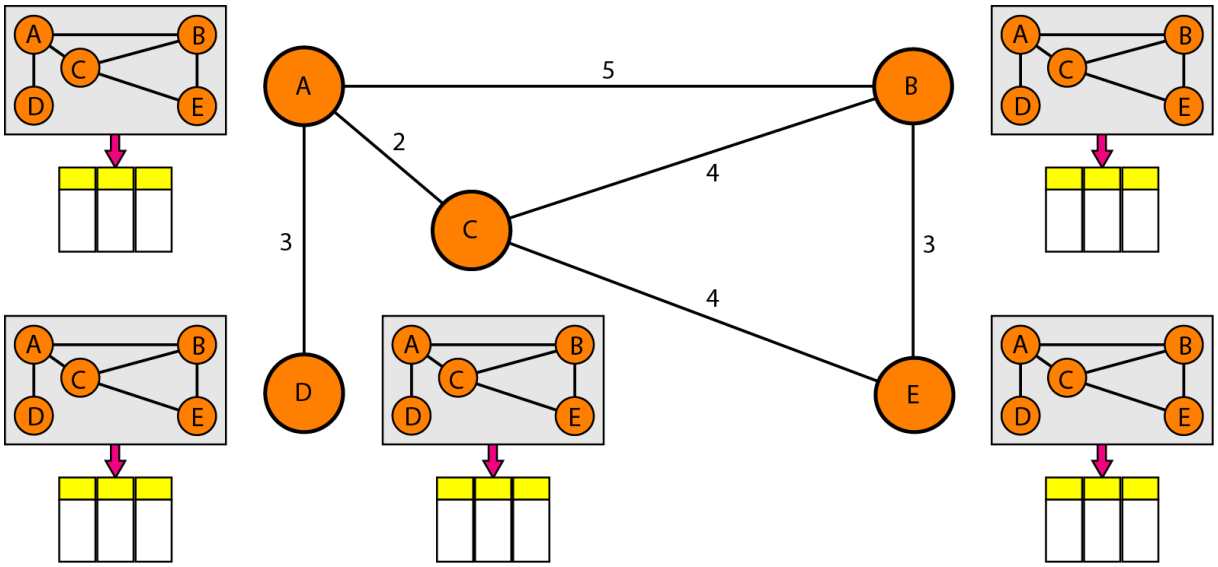
- The Routing Information Protocol (RIP) is an intradomain routing protocol used inside an autonomous system.
- It is a very simple protocol based on distance vector routing. RIP implements distance vector routing directly with some considerations:
- 1. In an autonomous system, we are dealing with routers and networks (links). The routers have routing tables; networks do not.

- 2. The destination in a routing table is a network, which means the first column defines a network address.
- 3. The metric used by RIP is very simple; the distance is defined as the number of links (networks) to reach the destination. For this reason, the metric in RIP is called a hop count.
- 4. Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
- 5. The next-node column defines the address of the router to which the packet is to be sent to reach its destination
- Figure 22.19 shows an autonomous system with seven networks and four routers. The table of each router is also shown.
- Let us look at the routing table for Rl. The table has seven entries to show how to reach each network in the autonomous system. Router Rl is directly connected to networks 130.10.0.0 and 130.11.0.0, which means that there are no next-hop entries for these two networks.
- To send a packet to one of the three networks at the far left, router Rl needs to deliver the packet to R2. The next-node entry for these three networks is the interface of router R2 with IP address 130.10.0.1. To send a packet to the two networks at the far right, router Rl needs to send the packet to the interface of router R4 with IP address 130.11.0.1. The other tables can be explained similarly.



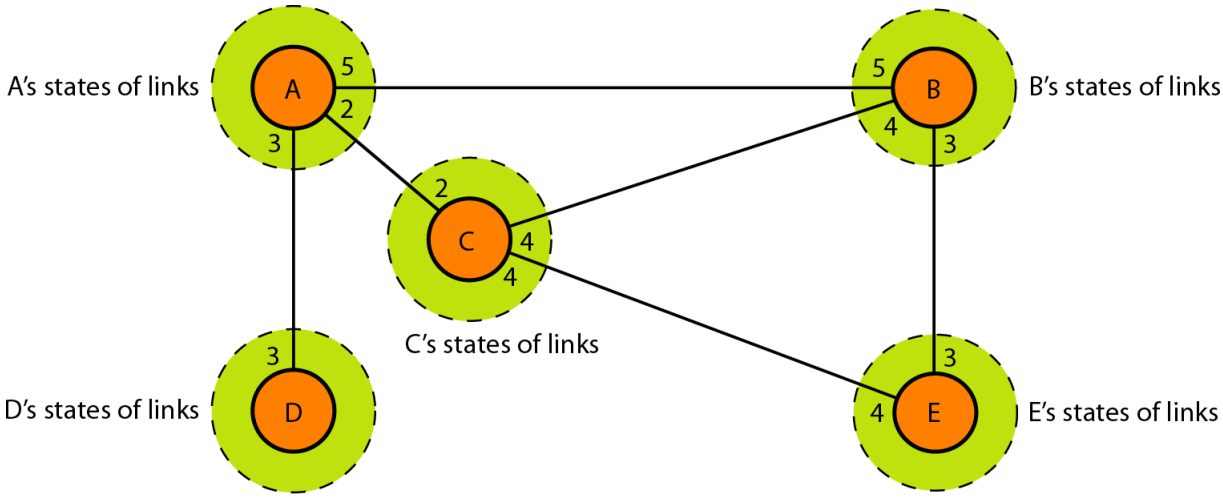
LINK STATE ROUTING

- Link state routing has a different philosophy from that of distance vector routing.
- In link state routing, if each node in the domain has the entire topology of the domain the list of nodes and links, how they are connected including the type, cost (metric), and condition of the links (up or down)-the node can use Dijkstra's algorithm to build a routing table. Below fig. shows the concept.



- The figure shows a simple domain with five nodes. Each node uses the same topology to create a routing table, but the routing table for each node is unique because the calculations are based on different interpretations of the topology.
- This is analogous to a city map. While each person may have the same map, each needs to take a different route to reach her specific destination.
- The topology must be dynamic, representing the latest state of each node and each link. If there are changes in any point in the network (a link is down, for example), the topology must be updated for each node.
- How can a common topology be dynamic and stored in each node? No node can know the topology at the beginning or after a change somewhere in the network.

- Link state routing is based on the assumption that, although the global knowledge about the topology is not clear, each node has partial knowledge: it knows the state (type, condition, and cost) of its links.
- In other words, the whole topology can be compiled from the partial knowledge of each node. Fig. below indicates the part of the knowledge belonging to each node.



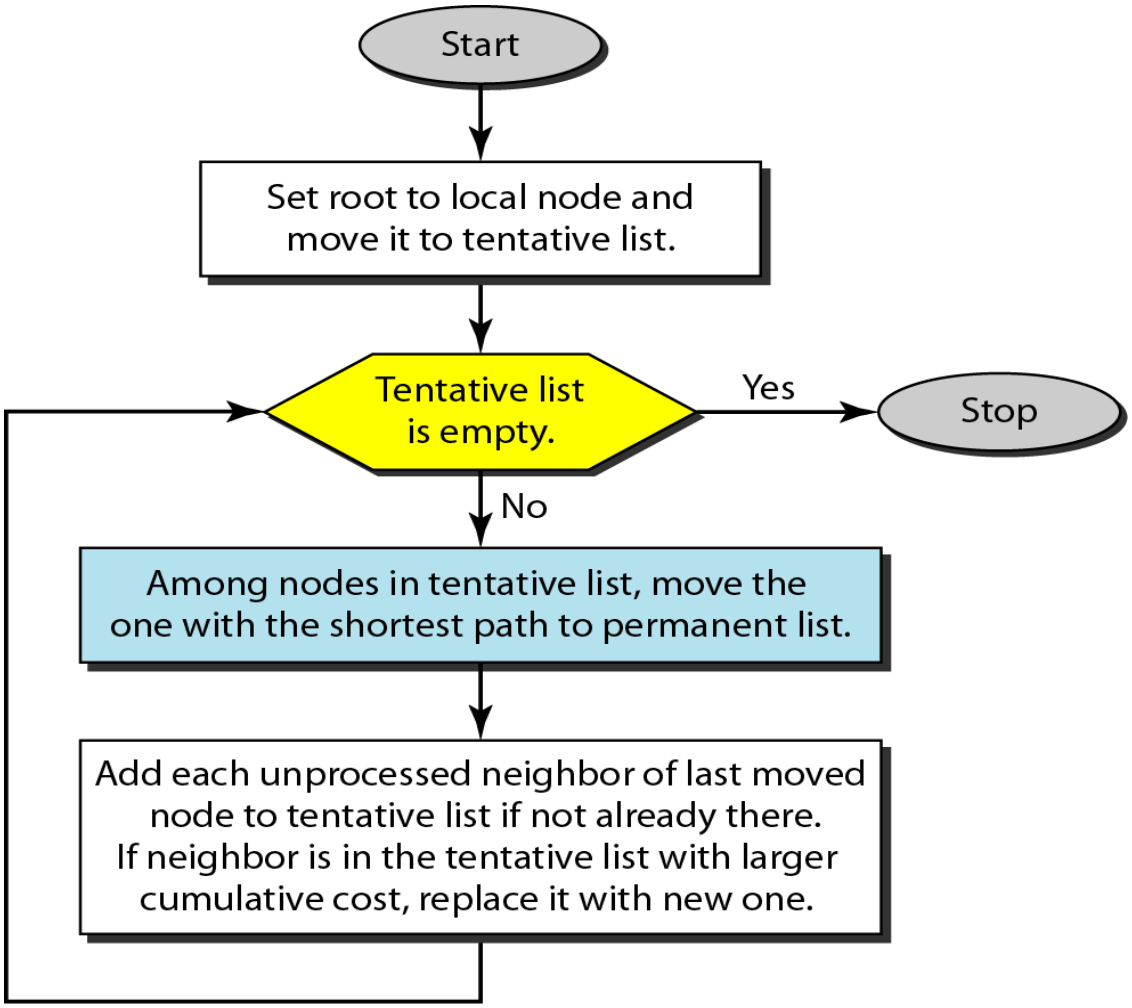
- Node A knows that it is connected to node B with metric 5, to node C with metric 2, and to node D with metric 3.
- Node C knows that it is connected to node A with metric 2, to node B with metric 4, and to node E with metric 4.
- Node D knows that it is connected only to node A with metric 3. And so on.
- Although there is an overlap in the knowledge, the overlap guarantees the creation of a common topology-a picture of the whole domain for each node.

BUILDING ROUTING TABLES

- In link state routing, four sets of actions are required to ensure that each node has the routing table showing the least-cost node to every other node.
- 1. Creation of the states of the links by each node, called the link state packet (LSP).
- 2. Dissemination of LSPs to every other router, called flooding, in an efficient and reliable way.

- 3. Formation of a shortest path tree for each node.
- 4. Calculation of a routing table based on the shortest path tree.
- **Creation of Link State Packet (LSP)** A link state packet can carry a large amount of information. For the moment, however, we assume that it carries a minimum amount of data: the node identity, the list of links, a sequence number, and age.
- The first two, node identity and the list of links, are needed to make the topology.
- The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones.
- The fourth, age, prevents old LSPs from remaining in the domain for a long time. LSPs are generated on two occasions:
- 1. When there is a **change in the topology** of the domain. Triggering of LSP dissemination is the main way of quickly informing any node in the domain to update its topology.
- 2. **On a periodic basis**: The period in this case is much longer compared to distance vector routing. The timer set for periodic dissemination is normally in the range of 60 min or 2 h based on the implementation.
- **Flooding of LSPs** After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbors. The process is called flooding and based on the following:
- 1. The creating node sends a copy of the LSP out of each interface.
- 2. A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP.
- If it is newer, the node does the following:
    - a. It discards the old LSP and keeps the new one.
    - b. It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

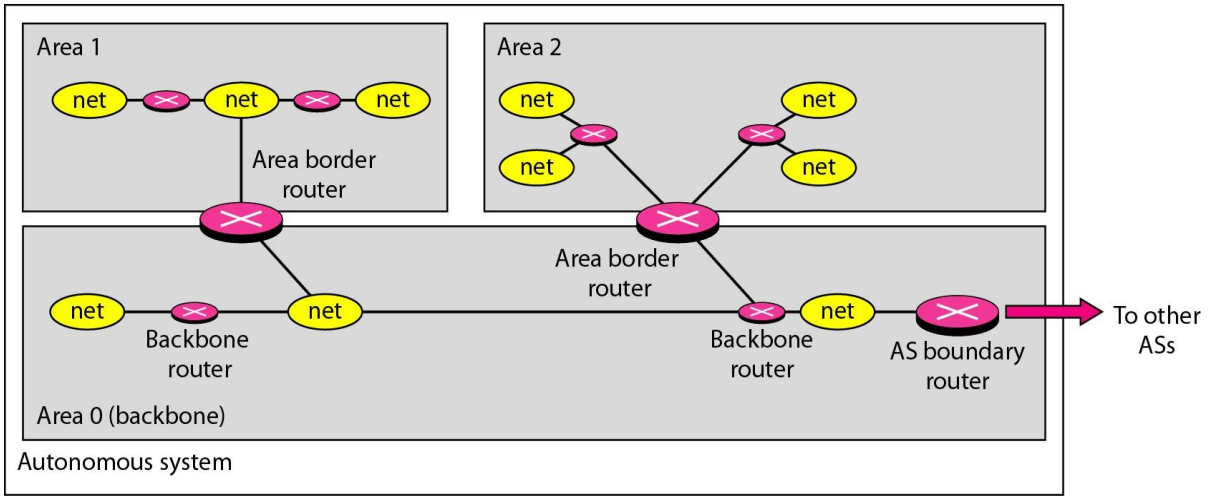
- **Formation of Shortest Path Tree:** Dijkstra Algorithm After receiving all LSPs, each node will have a copy of the whole topology.
- However, the topology is not sufficient to find the shortest path to every other node; a shortest path tree is needed.
- A tree is a graph of nodes and links; one node is called the root. All other nodes can be reached from the root through only one single route.
- A shortest path tree is a tree in which the path between the root and every other node is the shortest.
- What we need for each node is a shortest path tree with that node as the root.
- The Dijkstra algorithm creates a shortest path tree from a graph. The algorithm divides the nodes into two sets: tentative and permanent.
- It finds the neighbors of a current node, makes them tentative, examines them, and if they pass the criteria, makes them permanent. The process is illustrated in flow chart



Open Shortest Path First (OSPF)

- The Open Shortest Path First or OSPF protocol is an intradomain routing protocol based on link state routing.
- **Areas:** To handle routing efficiently and in a timely manner, OSPF divides an autonomous system into areas.
- An area is a collection of networks, hosts, and routers all contained within an autonomous system.
- Routers inside an area flood the area with routing information. At the border of an area, special routers called area border routers summarize the information about the area and send it to other areas.

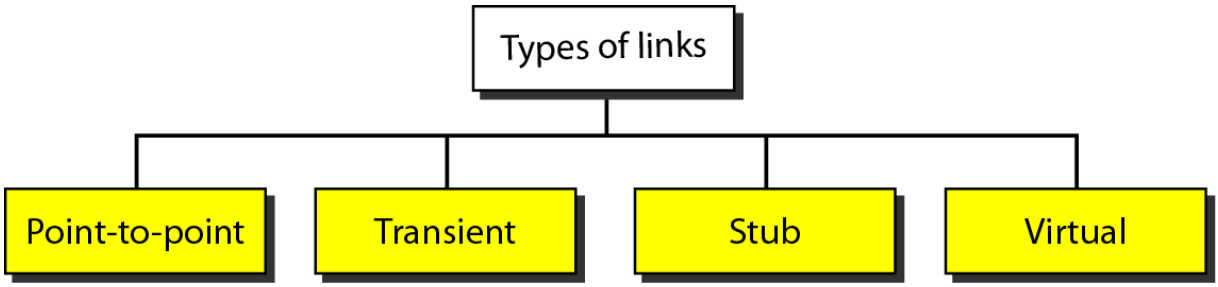
- Among the areas inside an autonomous system is a special area called the backbone; all the areas inside an autonomous system must be connected to the backbone.
- Here, the backbone serves as a primary area and the other areas as secondary areas. The routers inside the backbone are called the backbone routers.
- If the connectivity between a backbone and an area is broken, a virtual link between routers must be created by an administrator to allow continuity of the functions of the backbone as the primary area.
- Each area has an area identification. The area identification of the backbone is zero. Below Fig. shows an autonomous system and its areas.



- **Metric** The OSPF protocol allows the administrator to assign a cost, called the metric, to each route. The metric can be based on a type of service (minimum delay, maximum throughput, and so on).

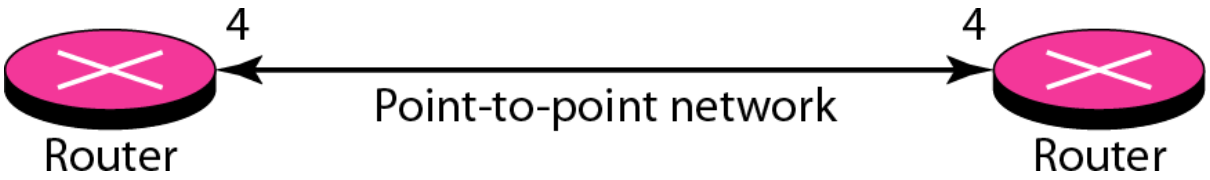
A router can have multiple routing tables, each based on a different type of service

- **Types of Links:** In OSPF terminology, a connection is called a link. Four types of links have been defined: point-to-point, transient, stub, and virtual as shown in fig.

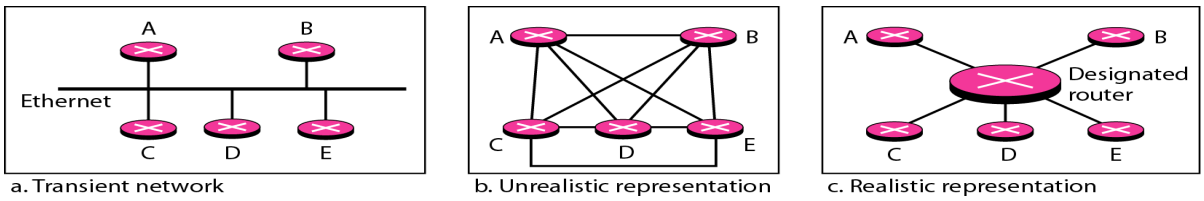


- A **point-to-point** link connects two routers without any other host or router in between. Graphically, the routers are represented by nodes, and the link is represented by a bidirectional edge connecting the nodes.

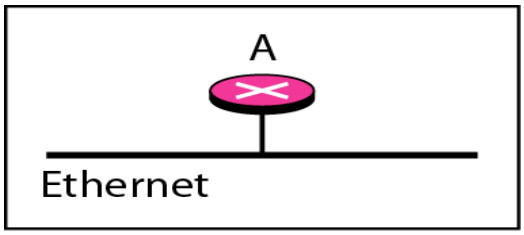
The metrics, which are usually the same, are shown at the two ends, one for each direction. In other words, each router has only one neighbor at the other side of the link (see Fig. below)



- A **transient link** is a network with several routers attached to it. The data can enter through any of the routers and leave through any router.
- All LANs and some WANs with two or more routers are of this type. In this case, each router has many neighbors.



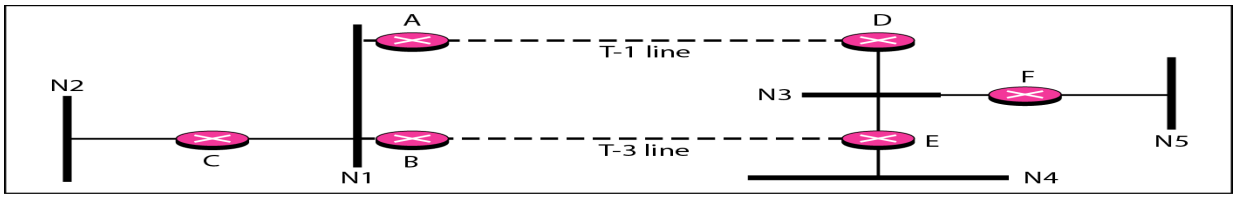
- A stub link is a network that is connected to only one router. The data packets enter the network through this single router and leave the network through this same router.
- This is a special case of the transient network. We can show this situation using the router as a node and using the designated router for the network. However, the link is only one-directional, from the router to the network (see Figure below).



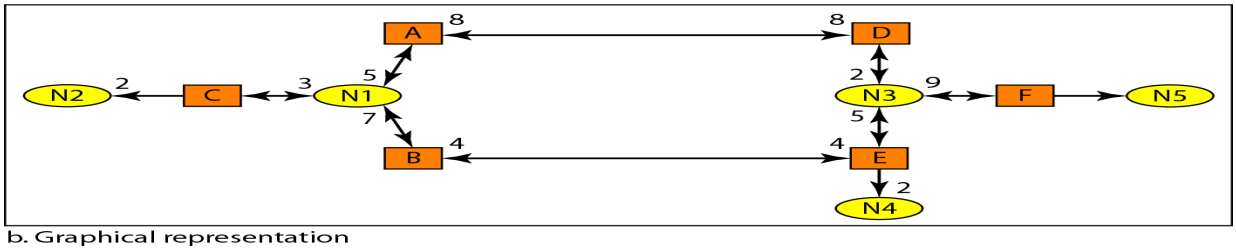
a. Stub network

b. Representation

- When the link between two routers is broken, the administration may create a **virtual link** between them, using a longer path that probably goes through several routers.
- **Graphical Representation** Let us now examine how an AS can be represented graphically. Two of the networks are point-to-point networks. We use symbols such as Nl and N2 for transient and stub networks. There is no need to assign an identity to a point-to-point network. The figure also shows the graphical representation of the AS as seen by OSPF.
- We have used square nodes for the routers and ovals for the networks (represented by designated routers). However, OSPF sees both as nodes. Note that we have three stub networks.



a. Autonomous system



b. Graphical representation

PATH VECTOR ROUTING

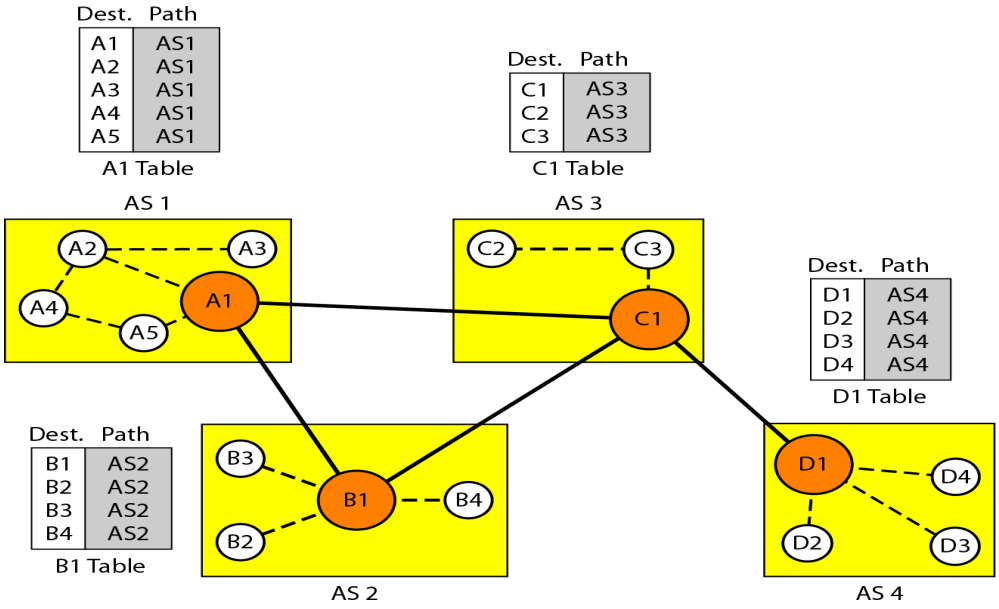
- Distance vector and link state routing are both intradomain routing protocols. They can be used inside an autonomous system, but not between autonomous systems.

Draw backs of distance vector and Link state routing:

- Distance vector routing is subject to instability if there are more than a few hops in the domain of operation. Link state routing needs a huge amount of resources to calculate routing tables.
- There is a need for a third routing protocol which we call **path vector routing**.
- **Path vector routing** proved to be useful for interdomain routing. The principle of path vector routing is similar to that of distance vector routing.
- In path vector routing, we assume that there is one node in each autonomous system that acts on behalf of the entire autonomous system and let us refer it as speaker node.
- The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring ASs.
- The idea is the same as for distance vector routing except that only speaker nodes in each AS can communicate with each other.
- However, what is advertised is different. A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems.

Initialization

- At the beginning, each speaker node can know only the reachability of nodes inside its autonomous system. Below fig. shows the initial tables for each speaker node in a system made of four ASs.
- Node Al is the speaker node for ASl, Bl for AS2, Cl for AS3, and Dl for AS4. Node Al creates an initial table that shows Al to A5 are located in ASI and can be reached through it. Node Bl advertises that Bl to B4 are located in AS2 and can be reached through Bl. And so on.



- **Sharing** Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors.
- In Figure, node Al shares its table with nodes Bl and Cl. Node Cl shares its table with nodes Dl, Bl, and Al. Node Bl shares its table with Cl and Al. Node Dl shares its table with Cl.
- **Updating** When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table.

After a while each speaker has a table and knows how to reach each node in other ASs. Here, figure shows the tables for each speaker node after the system is stabilized.

- **Stabilization Of Routing Table**: According to the figure, if router Al receives a packet for nodes A3, it knows that the path is in ASI (the packet is at home); but if it receives a packet for Dl, it knows that the packet should go from ASl, to AS2, and then to AS3. The routing table shows the path completely. On the other hand, if node Dl in AS4 receives a packet for node A2, it knows it should go through AS4, AS3, and AS 1
- **Loop prevention**: The instability of distance vector routing and the creation of loops can be avoided in path vector routing. When a router receives a message, it checks to see if its

autonomous system is in the path list to the destination. If it is, looping is involved and the message is ignored

- **Policy routing:** Policy routing can be easily implemented through path vector routing. When a router receives a message, it can check the path. If one of the autonomous systems listed in the path is against its policy, it can ignore that path and that destination. It does not update its routing table with this path, and it does not send this message to its neighbors.
- **Optimum path:** We cannot include metrics in this route because each autonomous system that is included in the path may use a different criterion for the metric. One system may use, internally, RIP, which defines hop count as the metric; another may use OSPF with minimum delay defined as the metric. The optimum path is the path that fits the organization. For the tables, we chose the one that had the smaller number of autonomous systems, but this is not always the case. Other criteria, such as security, safety, and reliability, can also be applied

Dest.	Path
A1	AS1
…	
A5	AS1
B1	AS1-AS2
…	…
B4	AS1-AS2
C1	AS1-AS3
…	…
C3	AS1-AS3
D1	AS1-AS2-AS4
…	…
D4	AS1-AS2-AS4

A1 Table

Dest.	Path
A1	AS2-AS1
…	
A5	AS2-AS1
B1	AS2
…	…
B4	AS2
C1	AS2-AS3
…	…
C3	AS2-AS3
D1	AS2-AS3-AS4
…	…
D4	AS2-AS3-AS4

B1 Table

Dest.	Path
A1	AS3-AS1
…	
A5	AS3-AS1
B1	AS3-AS2
…	…
B4	AS3-AS2
C1	AS3
…	…
C3	AS3
D1	AS3-AS4
…	…
D4	AS3-AS4

C1 Table

Dest.	Path
A1	AS4-AS3-AS1
…	
A5	AS4-AS3-AS1
B1	AS4-AS3-AS2
…	…
B4	AS4-AS3-AS2
C1	AS4-AS3
…	…
C3	AS4-AS3
D1	AS4
…	…
D4	AS4

D1 Table

BORDER GATEWAY PROTOCOL (BGP)

- Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector routing. It first appeared in 1989 and has gone through four versions.
- **Types of Autonomous Systems:** We can divide autonomous systems into three categories: stub, multihomed, and transit.

- **Stub AS:** A stub AS has only one connection to another AS. The interdomain data traffic in a stub AS can be either created or terminated in the AS.
- The hosts in the AS can send data traffic to other ASs. The hosts in the AS can receive data coming from hosts in other ASs. Data traffic, however, cannot pass through a stub AS.
- A stub AS is either a source or a sink. A good example of a stub AS is a small corporation or a small local ISP.
- **Multihomed AS:** A multihomed AS has more than one connection to other Ass, but it is still only a source or sink for data traffic. It can receive data traffic from more than one AS.
- It can send data traffic to more than one AS, but there is no transient traffic. It does not allow data coming from one AS and going to another AS to pass through. A good example of a multihomed AS is a large corporation that is connected to more than one regional or national AS that does not allow transient traffic.
- **Transit AS:** A transit AS is a multihomed AS that also allows transient traffic. Good examples of transit ASs are national and international ISPs (Internet backbones).

PATH ATTRIBUTES

- The path is a list of autonomous systems, in fact, it is a list of attributes. Each attribute gives some information about the path.
- The list of attributes helps the receiving router make a more-informed decision when applying its policy.
- Attributes are divided into two broad categories: well known and optional.
- A wellknown attribute is one that every BGP router must recognize.
- An optional attribute is one that needs not be recognized by every router.
- Well-known attributes are themselves divided into two categories: mandatory and discretionary.
- A well-known mandatory attribute is one that must appear in the description of a route.

- A well-known discretionary attribute is one that must be recognized by each router, but is not required to be included in every update message.

EX:One wellknown mandatory attribute is ORIGIN. This defines the source of the routing information (RIP, OSPF, and so on). Another well-known mandatory attribute is AS_PATH, next hop and so on.

- The optional attributes can also be subdivided into two categories: transitive and nontransitive.
- An optional transitive attribute is one that must be passed to the next router by the router that has not implemented this attribute.
- An optional nontransitive attribute is one that must be discarded if the receiving router has not implemented it.

22-4 MULTICAST ROUTING PROTOCOLS

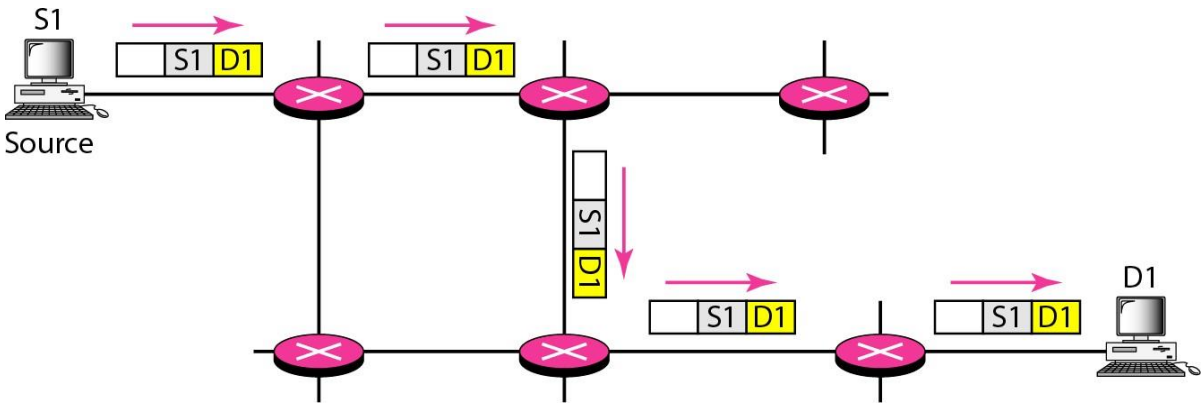
In this section, we discuss multicasting and multicast routing protocols.

Unicasting

- In unicast communication, there is one source and one destination.

The relationship between the source and the destination is one-to-one.

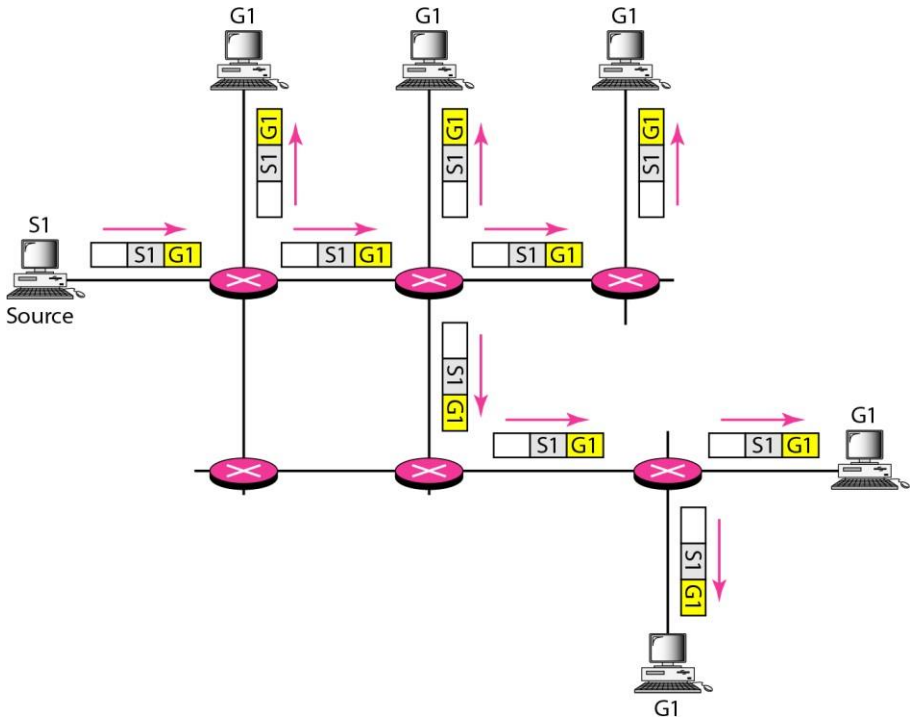
- In this type of communication, both the source and destination addresses, in the IP datagram, are the unicast addresses assigned to the hosts
- In below fig. A unicast packet starts from the source S1 and passes through routers to reach the destination D1.
- Note that in unicasting, when a router receives a packet, it forwards the packet through only one of its interfaces as defined in the routing table.
- The router may discard the packet if it cannot find the destination address in its routing table.



Multicasting

- In multicast communication, there is one source and a group of destinations.
- The relationship is one-to-many. In this type of communication, the source address is a unicast address, but the destination address is a group address, which defines one or more destinations.
- The group address identifies the members of the group. Below fig. shows the idea behind multicasting.
- A multicast packet starts from the source S1 and goes to all destinations that belong to group G1.

In multicasting, when a router receives a packet, it may forward it through several of its interfaces.



BROADCASTING

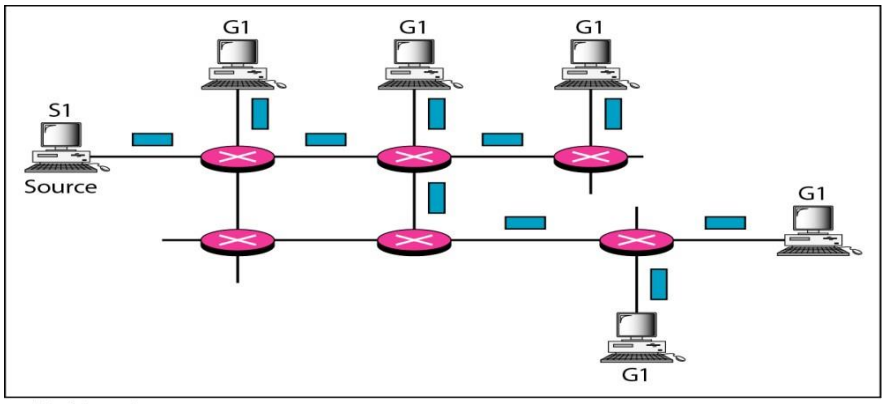
- In broadcast communication, the relationship between the source and the destination is one-to-all.
- There is only one source, but all the other hosts are the destinations. The Internet does not explicitly support broadcasting because of the huge amount of traffic it would create and because of the bandwidth it would need.
- Imagine the traffic generated in the Internet if one person wanted to send a message to everyone else connected to the Internet.

MULTICASTING VS MULTIPLE UNICASTING

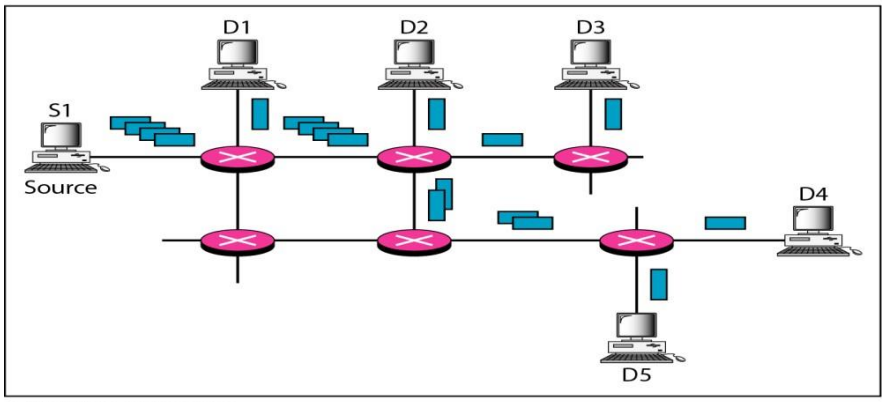
- Multicasting starts with one single packet from the source that is duplicated by the routers.
- The destination address in each packet is the same for all duplicates. Note that only one single copy of the packet travels between any two routers.
- In multiple unicasting, several packets start from the source. If there are five destinations, for example, the source sends five packets, each with a different unicast destination address.
- Note that there may be multiple copies traveling between two routers.
- For example, when a person sends an e-mail message to a group of people, this is multiple unicasting.
- The e-mail software creates replicas of the message, each with a different destination address and sends them one by one. This is not multicasting; it is multiple unicasting.

Emulation ofMulticasting with Unicasting

- You might wonder why we have a separate mechanism for multicasting, when it can be emulated with unicasting. There are two obvious reasons for this.
- 1. Multicasting is more efficient than multiple unicasting. Below fig shows how multicasting requires less bandwidth than does multiple unicasting. In multiple unicasting, some of the links must handle several copies.
- 2. In multiple unicasting, the packets are created by the source with a relative delay between packets. If there are 1000 destinations, the delay between the first and the last packet may be unacceptable. In multicasting, there is no delay because only one packet is created by the source.



a. Multicasting



b. Multiple unicasting

APPLICATIONS OF MULTICATING;

- Multicasting has many applications today such as access to distributed databases, information dissemination, teleconferencing, and distance learning

Access to Distributed Databases

- Most of the large databases today are distributed. That is, the information is stored in more than one location, usually at the time of production.
- The user who needs to access the database does not know the location of the information. A user's request is multicast to all the database locations, and the location that has the information responds.

Information Dissemination

- Businesses often need to send information to their customers. If the nature of the information is the same for each customer, it can be multicast.

- In this way a business can send one message that can reach many customers.

For example, a software update can be sent to all purchasers of a particular software packages.

Dissemination ofNews

- In a similar manner news can be easily disseminated through multicasting. One single message can be sent to those interested in a particular topic. For example, the statistics of the championship high school basketball tournament can be sent to the sports editors of many newspapers.

Teleconferencing

- Teleconferencing involves multicasting. The individuals attending a teleconference all need to receive the same information at the same time. Temporary or permanent groups can be formed for this purpose.
- For example, an engineering group that holds meetings every Monday morning could have a permanent group while the group that plans the holiday party could form a temporary group.

Distance Learning

- One growing area in the use of multicasting is distance learning. Lessons taught by one single professor can be received by a specific group of students.
- This is especially convenient for those students who find it difficult to attend classes on campus.

MULTICASTING ROUTING;

- In this section, we first discuss the idea of optimal routing, common in all multicast protocols. We then give an overview of multicast routing protocols.

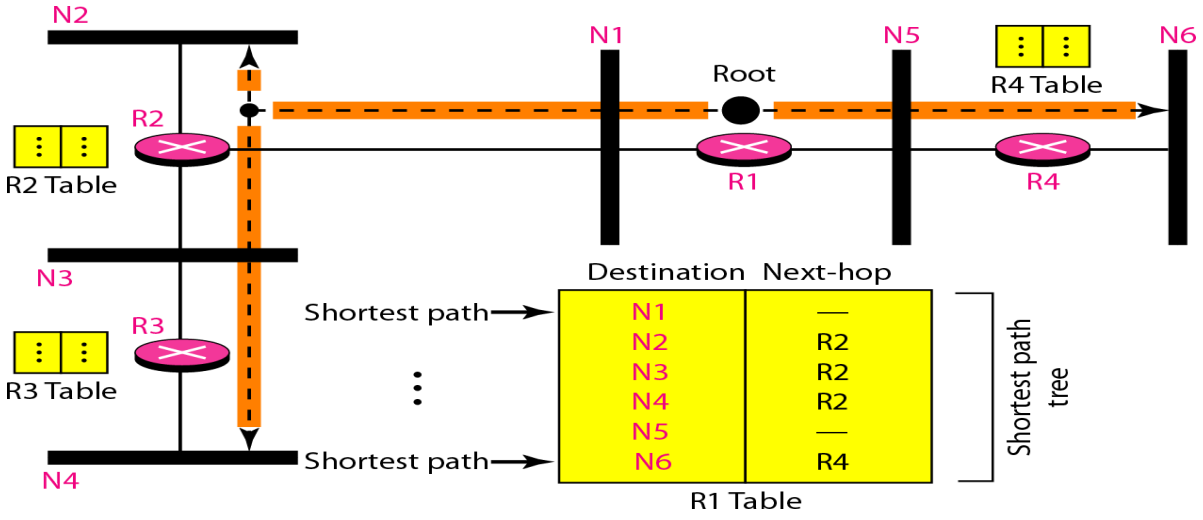
Optimal Routing: Shortest Path Trees

- The process of optimal interdomain routing eventually results in the finding of the shortest path tree. The root of the tree is the source, and the leaves are the potential destinations. The

path from the root to each destination is the shortest path. However, the number of trees and the formation of the trees in unicast and multicast routing are different.

Unicast Routing In unicast routing, when a router receives a packet to forward, it needs to find the shortest path to the destination of the packet. The router consults its routing table for that particular destination.

- The next-hop entry corresponding to the destination is the start of the shortest path. The router knows the shortest path for each destination, which means that the router has a shortest path tree to optimally reach all destinations.
- That is, each line of the routing table is a shortest path; the whole routing table is a shortest path tree.
- In unicast routing, each router needs only one shortest path tree to forward a packet; however, each router has its own shortest path tree. Below fig. shows the situation.

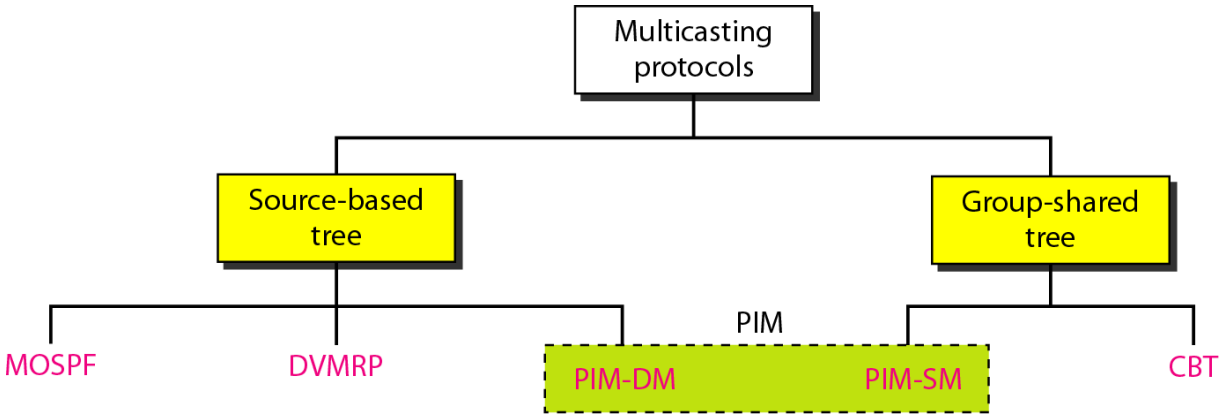


- The figure shows the details of the routing table and the shortest path tree for router Rl.
- Each line in the routing table corresponds to one path from the root to the corresponding network. The whole table represents the shortest path tree.

ROUTING PROTOCOLS;

- During the last few decades, several multicast routing protocols have emerged.

- Some of these protocols are extensions of unicast routing protocols; others are totally new.



Multicast Link State Routing: Multicast link state routing is a direct extension of unicast routing and uses a source-based tree approach.

- Recall that in unicast routing, each node needs to advertise the state of its links. For multicast routing, a node needs to revise the interpretation of state. A node advertises every group which has any loyal member on the link.
- The information about the group comes from IGMP. Each router running IGMP solicits the hosts on the link to find out the membership status.
- When a router receives all these LSPs, it creates n (n is the number of groups) topologies, from which n shortest path trees are made by using Dijkstra's algorithm.
- So each router has a routing table that represents as many shortest path trees as there are groups.
- The only problem with this protocol is the time and space needed to create and save the many shortest path trees. The solution is to create the trees only when needed and the result can be cached in case there are additional packets for that destination

Multicast Open Shortest Path First (MOSPF) protocol is an extension of the OSPF protocol that uses multicast link state routing to create source-based trees.

- The protocol requires a new link state update packet to associate the unicast address of a host with the group address or addresses the host is sponsoring. This packet is called the group-membership LSA.
- In this way, we can include in the tree only the hosts (using their unicast addresses) that belong to a particular group.
- In other words, we make a tree that contains all the hosts belonging to a group, but we use the unicast address of the host in the calculation.
- For efficiency, the router calculates the shortest path trees on demand (when it receives the first multicast packet). In addition, the tree can be saved in cache memory for future use by the same source/group pair.
- MOSPF is a data-driven protocol; the first time an MOSPF router sees a datagram with a given source and group address, the router constructs the Dijkstra shortest path tree.

Multicast Distance Vector Routing: Unicast distance vector routing is very simple; extending it to support multicast routing is complicated.

- Multicast routing does not allow a router to send its routing table to its neighbors.
- The idea is to create a table from scratch by using the information from the unicast distance vector tables.
- Multicast distance vector routing uses source-based trees, but the router never actually makes a routing table.
- When a router receives a multicast packet, it forwards the packet as though it is consulting a routing table. After its use (after a packet is forwarded) the table is destroyed.
- To accomplish this, the multicast distance vector algorithm uses a process based on four decision-making strategies. They are: Flooding, Reverse Path Forwarding, Reverse Path Boosting, Reverse Path Multicasting.
- Each strategy is built on its predecessor.

- Let us discuss one by one and see how each strategy can improve the shortcomings of the previous one.

Flooding:

Flooding is the first strategy that comes to mind. A router receives a packet and, without even looking at the destination group address, sends it out from every interlace except the one from which it was received.

- Flooding accomplishes the first goal of multicasting: every network with active members receives the packet. However, so will networks without active members.
- This is a broadcast, not a multicast. There is another problem: it creates loops. A packet that has left the router may come back again from another interlace or the same interlace and be forwarded again.
- Some flooding protocols keep a copy of the packet for a while and discard any duplicates to avoid loops. The next strategy, reverse path forwarding, corrects this defect.

Reverse Path Forwarding (RPF): RPF is a modified flooding strategy. To prevent loops, only one copy is forwarded; the other copies are dropped.

- In RPF, a router forwards only the copy that has traveled the shortest path from the source to the router. To find this copy, RPF uses the unicast routing table.
- The router receives a packet and extracts the source address (a unicast address). It consults its unicast routing table as though it wants to send a packet to the source address. The routing table tells the router the next hop.
- If the multicast packet has just come from the hop defined in the table, the packet has travelled the shortest path from the source to the router because the shortest path is reciprocal in unicast distance vector routing protocols.
- If the path from A to B is the shortest, then it is also the shortest from B to A. The router forwards the packet if it has travelled from the shortest path; it discards it otherwise.

- This strategy prevents loops because there is always one shortest path from the source to the router. If a packet leaves the router and comes back again, it has not traveled the shortest path. To make the point clear, let us refer fig. in next slide

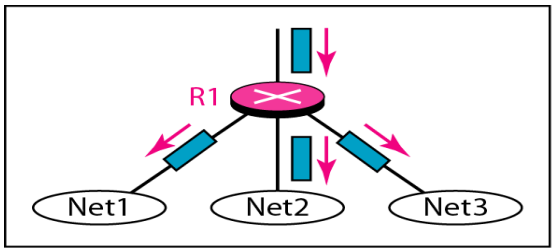
Reverse Path Broadcasting (RPB): RPF guarantees that each network receives a copy of the multicast packet without formation of loops.

- However, RPF does not guarantee that each network receives only one copy; a network may receive two or more copies.
- The reason is that RPF is not based on the destination address (a group address); forwarding is based on the source address.
- To visualize the problem, let us look at Fig.
- Net3 in this fig. receives two copies of the packet even though each router just sends out one copy from each interface.
- Net3 in this fig. receives two copies of the packet even though each router just sends out one copy from each interface.
- There is duplication because a tree has not been made; instead of a tree we have a graph.
- Net3 has two parents: routers R2 andR4.
- To eliminate this, we must define only one parent router for each network

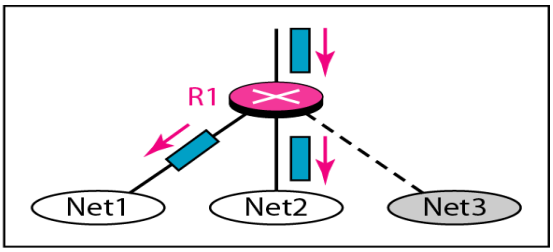
Reverse Path Multicasting (RPM): As you may have noticed, RPB does not multicast the packet, it broadcasts it. This is not efficient.

- To increase efficiency, the multicast packet must reach only those networks that have active members for that particular group. This is called reverse path multicasting (RPM).
- To convert broadcasting to multicasting, the protocol uses two procedures, pruning and grafting.

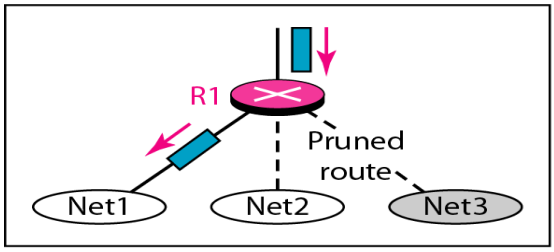
Figure here shows the idea of pruning and grafting



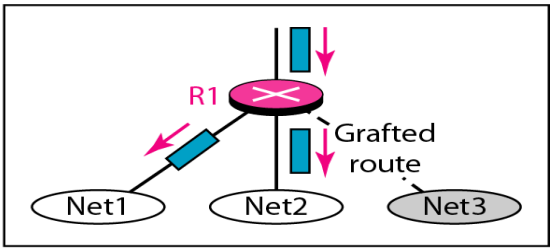
a.RPF



b. RPB



c. RPM (after pruning)



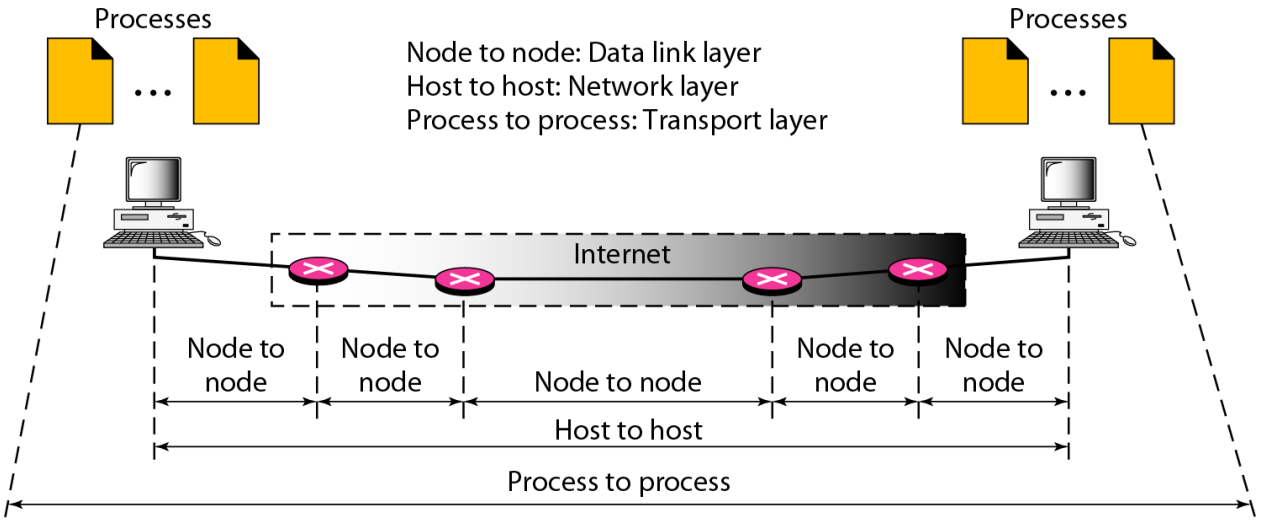
d. RPM (after grafting)

UNIT-5

TRANSPORT LAYER

PROCESS-TO-PROCESS DELIVERY

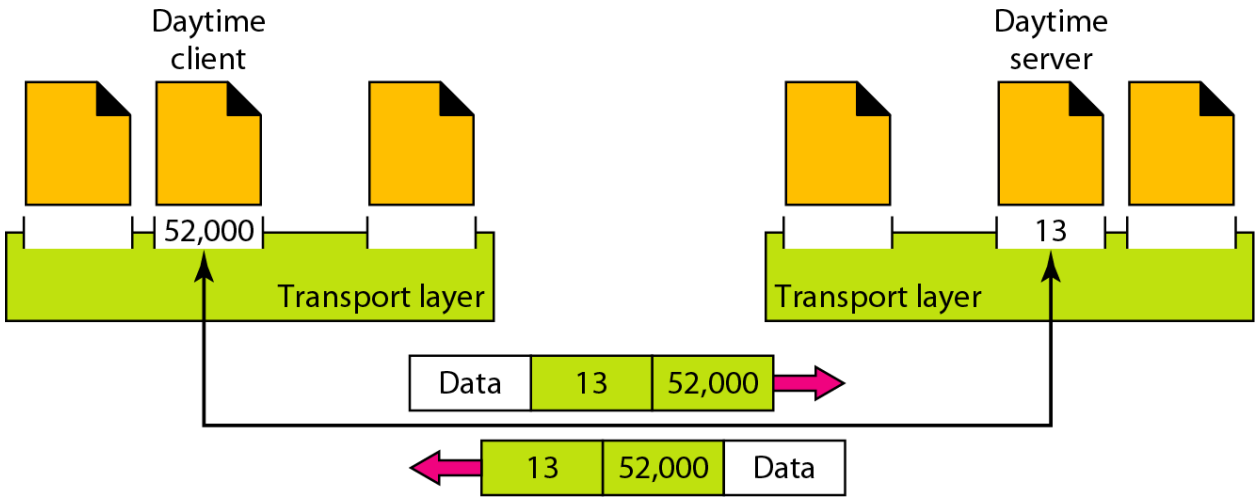
- The data link layer is responsible for delivery of frames between two neighboring nodes over a link. This is called **node-to-node delivery**.
- The network layer is responsible for delivery of datagrams between two hosts. This is called **host-to-host delivery**.
- The real communication takes place between two processes (application programs). We need process-to-process delivery.
- 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



CLIENT/SERVER PARADIGM

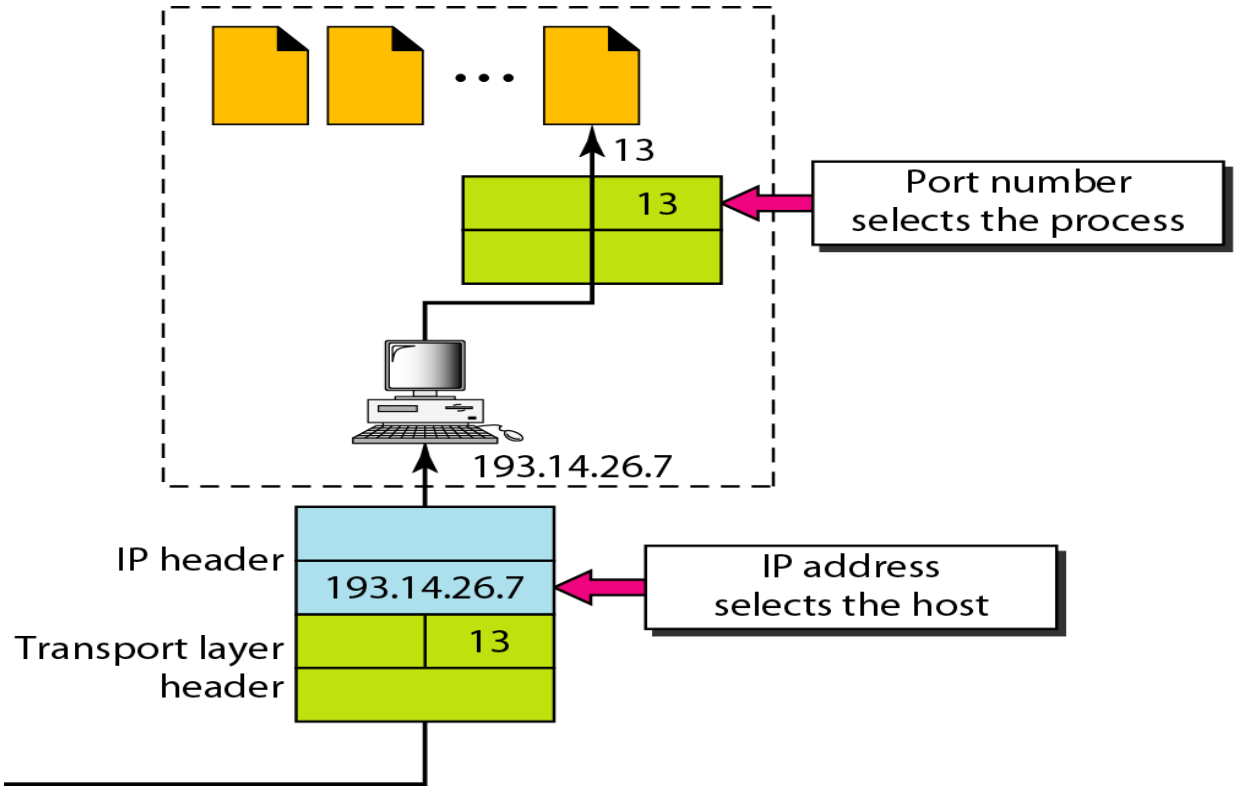
- A process on the local host, called a client, needs services from a process usually on the remote host, called a server. Both processes (client and server) have the same name.

- For example, to get the day and time from a remote machine, we need a Daytime client process running on the local host and a Daytime server process running on a remote machine.
- Whenever we need to deliver something to one specific destination among many, we need an address. At the transport layer, we need a transport layer address, called a port number, to choose among multiple processes running on the destination host.
- The destination port number is needed for delivery; the source port number is needed for the reply.
- In the Internet model, the port numbers are 16-bit integers between 0 and 65,535.
- The client program defines itself with a port number, chosen randomly by the transport layer software running on the client host. This is the ephemeral port number.
- The server process must also define itself with a port number. However this cannot be chosen randomly.
- If the port number is chosen randomly then the process at the client site that wants to access that server and use its services will not know the port number.
- Of course, one solution would be to send a special packet and request the port number of a specific server, but this requires more overhead.
- To overcome the drawback of additional overhead, the internet has chosen to assign a universal port number called well known port number to the server program.



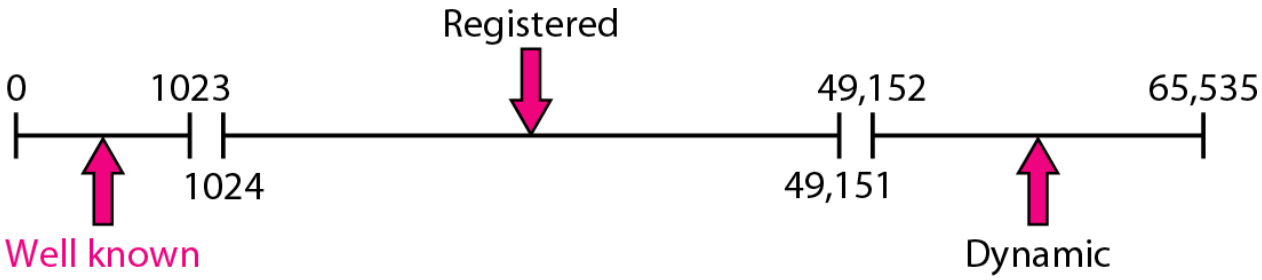
IP Address Vs Port Address: The destination IP address defines the host among the different hosts

in the world. After the host has been selected, the port number defines one of the processes on this particular host.



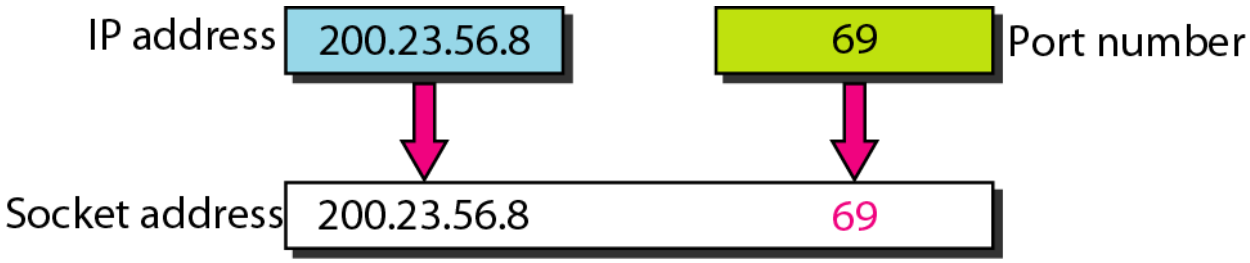
IANA RANGES

- The IANA (Internet Assigned Number Authority) has divided the port numbers into three ranges: well known, registered, and dynamic (or private).
- Well-known ports: The ports ranging from 0 to 1023 are assigned and controlled by IANA. These are the well-known ports.
- Registered ports: The ports ranging from 1024 to 49,151 are not assigned or controlled by IANA. They can only be registered with IANA to prevent duplication.
- Dynamic ports: The ports ranging from 49,152 to 65,535 are neither controlled nor registered. They can be used by any process. These are the ephemeral ports



SOCKET ADDRESSES

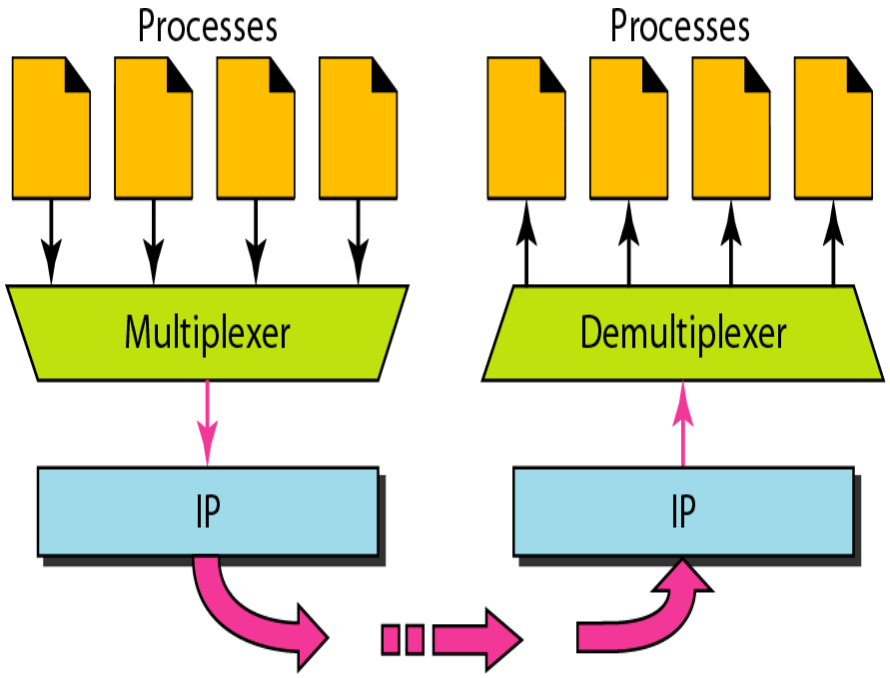
- Process to Process delivery needs two identifiers, IP address, and port number, at each end to make a connection.
- The combination of an IP address and a port number is called a socket address.
- The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely.
- A Transport Layer protocol needs a pair of socket addresses; the client socket address and the server socket address for processes to process communication.



MULTIPLEXING & DEMULTIPLEXING

- Multiplexing: At the sender site, there may be several processes that need to send packets. But, there is only one transport layer protocol at any time. This is a many-to-one relationship and requires multiplexing.
- The protocol accepts messages from different processes, differentiated by their assigned port numbers. After adding the header, the transport layer passes the packet to the network layer.
- Demultiplexing: At the receiver site, the relationship is one-to-many and requires demultiplexing.

- The transport layer receives datagrams from the network layer. After error checking and dropping of the header, the transport layer delivers each message to the appropriate process based on the port number.



CONNECTIONLESS VERSUS CONNECTION-ORIENTED SERVICE

- In a connectionless service, the packets are sent from one party to another with no need for connection establishment or connection release. The packets are not numbered; they may be delayed or lost or may arrive out of sequence. There is no acknowledgment either. We will see shortly that one of the transport layer protocols in the Internet model, UDP, is connectionless.

- In a connection-oriented service, a connection is first established between the sender and the receiver. Data are transferred. At the end, the connection is released. We will see shortly that TCP and SCTP are connection-oriented protocols.

RELIABLE VERSUS UNRELIABLE

- Reliable: The transport layer service can be reliable or unreliable. If the application layer program needs reliability, we use a reliable transport layer protocol by implementing flow and error control at the transport layer.

- This means a slower and more complex service. On the other hand, if the application program does not need reliability because it uses its own flow and error control mechanism

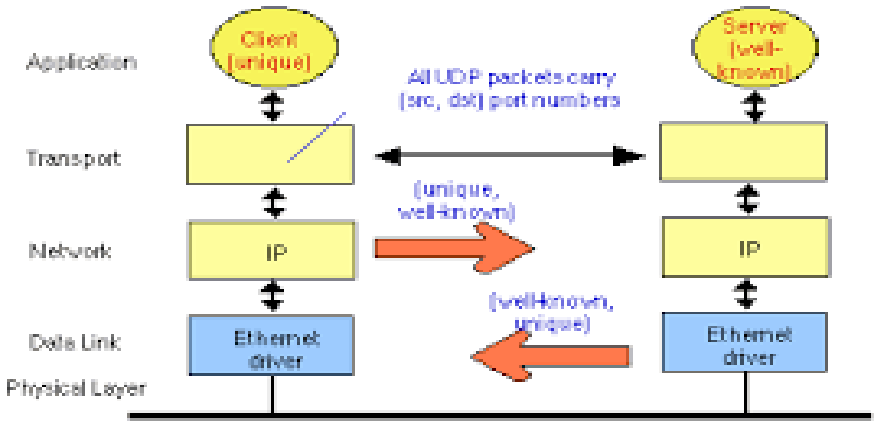
or it needs fast service or the nature of the service does not demand flow and error control (real-time applications), then an unreliable protocol can be used.

- Note : *If the data link layer is reliable and has flow and error control, do we need this at the transport layer, too? The answer is yes. Reliability at the data link layer is between two nodes; we need reliability between two ends.*

UDP:

The **User Datagram Protocol** (**UDP**) is one of the core members of the Internet protocol suite. The protocol was designed by David P. Reed in 1980 and formally defined in RFC 768.

UDP uses a simple connectionless transmission model with a minimum of protocol mechanism. It has no handshaking dialogues, and thus exposes any unreliability of the underlying network protocol to the user's program. There is no guarantee of delivery, ordering, or duplicate protection. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram.



With UDP, computer applications can send messages, in this case referred to as datagram, to other hosts on an Internet Protocol (IP) network without prior communications to set up special transmission channels or data paths. UDP is suitable for purposes where error checking and correction is either not necessary or is performed in the application, avoiding the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system.[1] If error correction facilities are needed at the network interface

level, an application may use the Transmission Control Protocol (TCP) or Stream Control Transmission Protocol (SCTP)

UDP (User Datagram Protocol) is an alternative communications protocol to Transmission Control Protocol (TCP) used primarily for establishing low-latency and loss tolerating connections between applications on the Internet. Both UDP and TCP run on top of the Internet Protocol (IP) and are sometimes referred to as UDP/IP or TCP/IP. Both protocols send short packets of data, called datagram.

UDP provides two services not provided by the IP layer. It provides port numbers to help distinguish different user requests and, optionally, a checksum capability to verify that the data arrived intact.

TCP has emerged as the dominant protocol used for the bulk of Internet connectivity owing to services for breaking large data sets into individual packets, checking for and resending lost packets and reassembling packets into the correct sequence. But these additional services come at a cost in terms of additional data overhead, and delays called latency.

UDP is an ideal protocol for network applications in which perceived latency is critical such as gaming, voice and video communications, which can suffer some data loss without adversely affecting perceived quality. In some cases, forward error correction techniques are used to improve audio and video quality in spite of some loss.

UDP can also be used in applications that require lossless data transmission when the application is configured to manage the process of retransmitting lost packets and correctly arranging received packets. This approach can help to improve the data transfer rate of large files compared with TCP.

Attributes

A number of UDP's attributes make it especially suited for certain applications.

- It is transaction-oriented, suitable for simple query-response protocols such as the Domain Name System or the Network Time Protocol.
- It provides datagram, suitable for modeling other protocols such as in IP tunneling or Remote Procedure Call and the Network File System.
- It is simple, suitable for bootstrapping or other purposes without a full protocol stack, such as the DHCP and Trivial File Transfer Protocol.
- The lack of retransmission delays makes it suitable for real-time applications such as Voice over IP, online games, and many protocols built on top of the Real Time Streaming
- Protocol.
  It is stateless, suitable for very large numbers of clients, such as in streaming media applications for example IPTV
- Works well in unidirectional communication, suitable for broadcast information such as in many kinds of service discovery and shared information such as broadcast time or Routing Information Protocol
- UDP provides application multiplexing (via port numbers) and integrity verification (via checksum) of the header and payload.[4] If transmission reliability is desired, it must be implemented in the user's application.

- The UDP header consists of 4 fields, each of which is 2 bytes (16 bits).[1] The use of the fields "Checksum" and "Source port" is optional in IPv4 (pink background in table). In IPv6 only the source port is optional (see below).
- Source port number
- This field identifies the sender's port when meaningful and should be assumed to be the port to reply to if needed. If not used, then it should be zero. If the source host is the client, the port number is likely to be an ephemeral port number. If the source host is the server, the port number is likely to be a well-known port number.[2]
- Destination port number
- This field identifies the receiver's port and is required. Similar to source port number, if the client is the destination host then the port number will likely be an ephemeral port number and if the destination host is the server then the port number will likely be a well- known port number.[2]
- Length
- A field that specifies the length in bytes of the UDP header and UDP data. The minimum length is 8 bytes because that is the length of the header. The field size sets a theoretical limit of 65,535 bytes (8 byte header + 65,527 bytes of data) for a UDP datagram. The practical limit for the data length which is imposed by the underlying IPv4 protocol is 65,507 bytes (65,535 − 8 byte UDP header − 20 byte IP header).[2]
- In IPv6 Jumbo grams it is possible to have UDP packets of size greater than 65,535 bytes.[5] RFC 2675 specifies that the length field is set to zero if the length of the UDP header plus UDP data is greater than 65,535.
- Checksum
- The checksum field is used for error-checking of the header and data. If no checksum is generated by the transmitter, the field uses the value all-zeros.[6] This field is not optional for IPv6.[7]

Reliable byte stream (TCP):

A **reliable byte stream** is a common service paradigm in computer networking; it refers to a byte stream in which the bytes which emerge from the communication channel at the recipient are exactly the same, and in exactly the same order, as they were when the sender inserted them into the channel.

The classic example of a reliable byte stream communication protocol is the Transmission Control Protocol, one of the major building blocks of the Internet.

A reliable byte stream is not the only reliable service paradigm which computer network communication protocols provide, however; other protocols (e.g. SCTP) provide a reliable message stream, i.e. the data is divided up into distinct units, which are provided to the consumer of the data as discrete objects.

Connection-oriented (TCP):

- Flow control: keep sender from overrunning receiver
- Congestion control: keep sender from overrunning network

Characteristics of TCP Reliable Delivery:

TCP provides a **reliable, byte-stream, full-duplex inter-process communications service** to application programs/processes. The service is **connection-oriented** and uses the concept of **port numbers** to identify processes.

Reliable

All data will be delivered correctly to the destination process, without errors, even though the underlying packet delivery service (IP) is unreliable -- see later.

Connection-oriented

Two process which desire to communicate using TCP must first request a **connection**. A connection is closed when communication is no longer desired.

Byte-stream

An application which uses the TCP service is unaware of the fact that data is broken into **segments** for transmission over the network.

Full-duplex

Once a TCP connection is established, application data can flow in both directions simultaneously -- note, however, that many application protocols do not take advantage of this.

Port Numbers

Port numbers identify processes/connections in TCP.

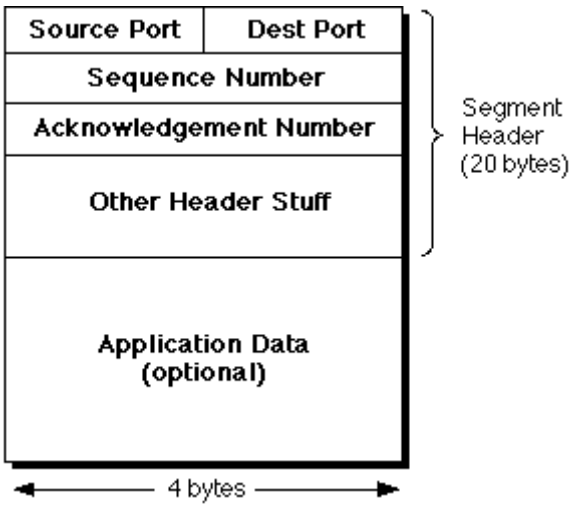
Edge Systems and Reliable Transport

1. An **edge system** is any computer (host, printer, even a toaster...) which is "connected to" the Internet -- that is, it has access to the Internet's packet delivery system, but doesn't itself form part of that delivery system.
2. A **transport service** provides communications between application processes running on edge systems. As we have already seen, application processes communicate with each another using **application protocols** such as HTTP and SMTP. The interface between an application process and the transport service is normally provided using the **socket** mechanism.

Most application protocols require **reliable data transfer**, which in the Internet is provided by the **TCP** transport service/protocol. Note: some applications **do not** require reliability, so the unreliable **UDP** transport service/protocol is also provided as an alternative

TCP Segments

TCP slices (dices?) the incoming byte-stream data into **segments** for transmission across the Internet. A segment is a highly-structured data package consisting of an administrative **header** and some **application data**.



Source and Destination Port Numbers

We have already seen that TCP server processes wait for connections at a pre-agreed port number. At connection establishment time, TCP first allocates a **client port number** -- a port number by which the client, or initiating, process can be identified. Each segment contains both port numbers.

Segment and Acknowledgment Numbers

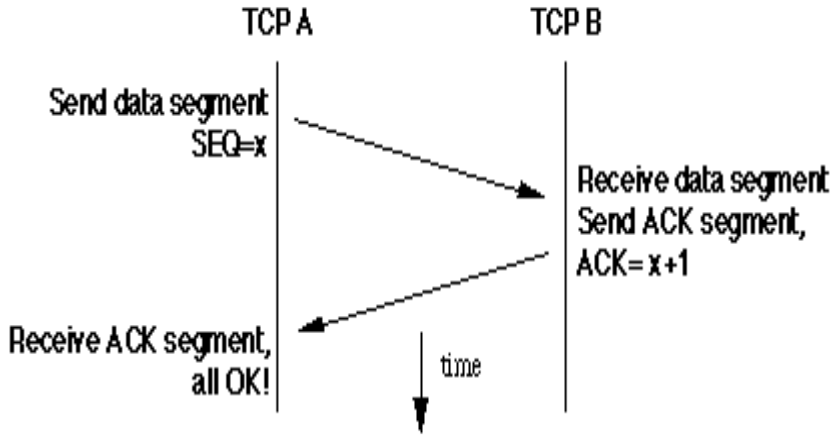
Every transmitted segment is identified with a 32-bit **Sequence number**[2], so that it can be explicitly acknowledged by the receipient. The Acknowledgment Number identifies the last segment recived by the originator of this segment.

Application Data

Optional because some segments convey only **control information** -- for example, an ACK segment has a valid acknowledgment number field, but no data. The data field can be any size up to the currently configured **MSS** for the whole segment.

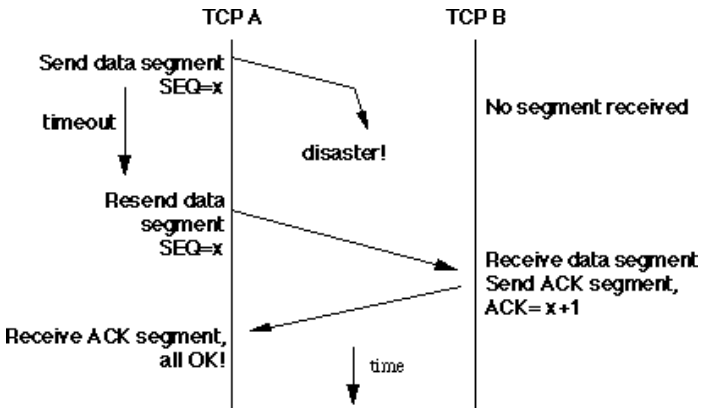
TCP Operation

When a segment is received correct and intact at its destination, an **acknowledgment** (ACK) segment is returned to the sending TCP. This ACK contains the sequence number of the last byte correctly received, incremented by 1[3]. ACKs are cumulative -- a single ACK can be sent for several segments if, for example, they all arrive within a short period of time.



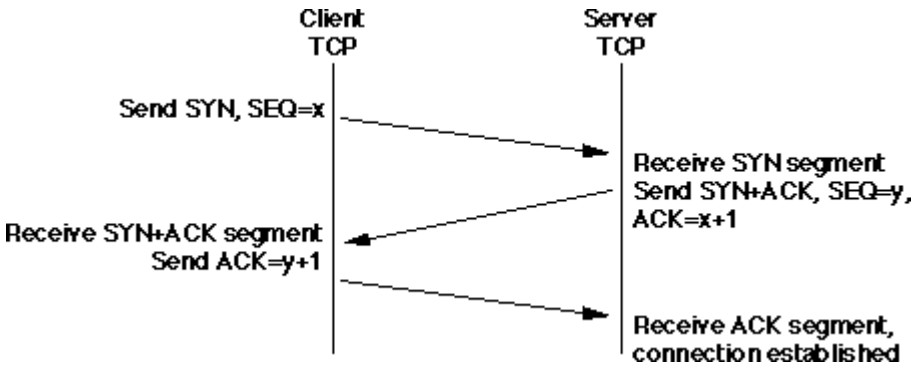
The network service can fail to deliver a segment. If the sending TCP waits for **too long**[4] for an acknowledgment, it times out and resends the segment, on the assumption that the datagram has been lost.

In addition, the network can potentially deliver duplicated segments, and can deliver segments out of order. TCP buffers or discards out of order or duplicated segments appropriately, using the byte count for identification.



TCP Connections:

An application process requests TCP to establish, or open, a (reliable) connection to a server process running on a specified edge-system, and awaiting connections at a known port number. After allocating an unused client-side port number[5], TCP initiates an exchange of connection establishment "control segments":



- This exchange of segments is called a **3-way handshake** (for obvious reasons), and is necessary because any one of the three segments can be lost, etc. The **ACK** and **SYN** segment names refer to "control bits" in the TCP header: for example, if the **ACK** bit is set, then this is an **ACK** segment.
- Each TCP chooses an **random initial sequence number** (the **x** and **y** in this example). This is crucial to the protocol's operation if there's a small chance that "old" segments (from a closed connection) might be interpreted as valid within the current connection.
- A connection is **closed** by another 3-way handshake of control segments. It's possible for a connection to be **half open** if one end requests close, and the other doesn't respond with an appropriate segment.

Optional: TCP Flow Control, Congestion Control and Slow Start

TCP attempts to make the best possible use of the underlying network, by sending data at the highest possible rate that won't cause segment loss. There are two aspects to this:

Flow Control

The two TCPs involved in a connection each maintain a **receive window** for the connection, related to the size of their **receive buffers**. For TCP "**A**", this is the maximum number of bytes that TCP "**B**" should send to it before "blocking" and waiting for an ACK. All TCP segments contain a **window** field, which is used to inform the other TCP of the sender's receive window size -- this is called "advertising a window size". At any time, for example, TCP **B** can have multiple segments "**in-flight**" -- that is, sent but not yet ACK'd -- up to TCP **A**'s advertised window.

Congestion Avoidance and Control

When a connection is initially established, the TCPs know nothing at all about the speed, or capacity, of the networks which link them. The built-in "**slow start**" algorithm controls the rate at which segments are initially sent, as TCP tentatively discovers reasonable numbers for the connection's **Round Trip Time (RTT)** and its variability. TCP also slowly increases the number of segments "in-flight", since this increases the utilisation of the network.

Every TCP in the entire Internet is attempting to make full use of the available network, by increasing the number of "in-flight" segments it has outstanding. Ultimately there will come a point where the sum of the traffic, in some region of the network exceeds one or more router's buffer space, at which time segments will be dropped. When TCP "times out", and has to resend a dropped segment, it takes this as an indication that it (and all the other TCPs) have pushed the network just a little too hard. TCP immediately reduces its **congestion window** to a low value, and slowly, slowly allows it to increase again as ACKs are received.

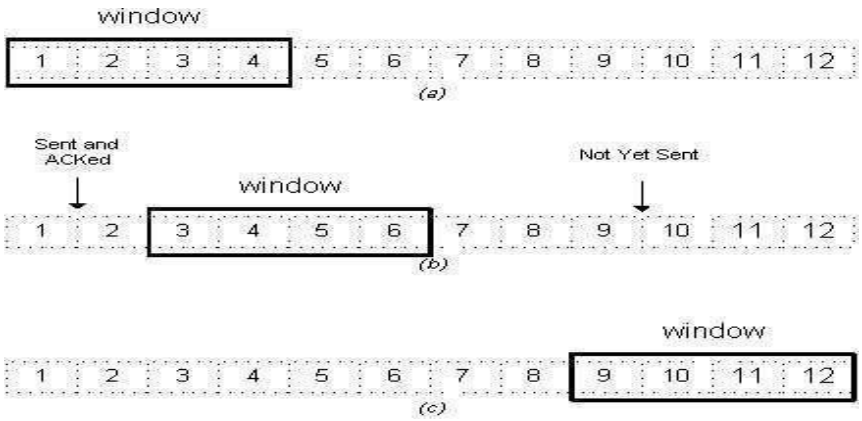
Flow control:

- Based on window mechanism

One of TCP's primary functions is to properly match the transmission rate of the sender to that of the receiver and the network. It is important for the transmission to be at a high enough rate to ensure good performance, but also to protect against overwhelming the network or receiving host.

TCP's 16-bit window field is used by the receiver to tell the sender how many bytes of data the receiver is willing to accept. Since the window field is limited to a maximum of 16 bits, this provides for a maximum window size of 65,535 bytes.

The window size advertised by the receiver tells the sender how much data, starting from the current position in the TCP data byte stream can be sent without waiting for further acknowledgements. As data is sent by the sender and then acknowledged by the receiver, the window slides forward to cover more data in the byte stream. This concept is known as a "sliding window".



The window boundary is eligible to be sent by the sender. Those bytes in the stream prior to the window have already been sent and acknowledged. Bytes ahead of the window have not been sent and must wait for the window to "slide" forward before they can be transmitted by the sender. A receiver can adjust the window size each time it sends acknowledgements to the sender. The maximum transmission rate is ultimately bound by the receiver's ability to accept and process data.

ERROR CONTROL:

Retransmission:

TCP is relegated to rely mostly upon implicit signals it learns from the network and remote host. TCP must make an educated guess as to the state of the network and trust the information from the remote host in order to control the rate of data flow. This may seem like an awfully tricky problem, but in most cases TCP handles it in a seemingly simple and straightforward way.

A sender's implicit knowledge of network conditions may be achieved through the use of a **timer**. For each TCP segment sent the sender expects to receive an acknowledgement within some period of time otherwise an error in the form of a timer expiring signals that that something is wrong.

Somewhere in the end-to-end path of a TCP connection a segment can be lost along the way. Often this is due to congestion in network routers where excess packets must be dropped. TCP not only must correct for this situation, but it can also **learn** something about network conditions from it.

Whenever TCP transmits a segment the sender starts a timer which keeps track of how long it takes for an acknowledgment for that segment to return. This timer is known as the **retransmission timer**. If an acknowledgement is returned before the timer expires,
which by default is often initialized to 1.5 seconds, the timer is reset with no consequence. If however an acknowledgement for the segment does not return within the timeout period, the sender would retransmit the segment and double the retransmission timer value for each consecutive timeout up to a maximum of about 64 seconds. If there are serious network problems, segments may take a few minutes to be successfully transmitted before the sender eventually times out and generates an error to the sending application.

Fundamental to the timeout and retransmission strategy of TCP is the measurement of the **round-trip time** between two communicating TCP hosts. The round-trip time may vary during the TCP connection as network traffic patterns fluctuate and as routes become available or unavailable.

A TCP option negotiated in the TCP connection establishment phase sets the number of bits by which the window is right-shifted in order to increase the value of the window. TCP keeps track of when data is sent and at what time acknowledgements covering those sent bytes are returned. TCP uses this information to calculate an estimate of round trip time. As packets are sent and acknowledged, TCP adjusts its round-trip time estimate and uses this information to come up with a reasonable timeout value for packets sent. If acknowledgements return quickly, the round-trip time is short and the retransmission timer is thus set to a lower value. This allows TCP to quickly retransmit data when network response time is good, alleviating the need for a long delay between the occasional lost segment. The converse is also true. TCP does not retransmit data too quickly during times when network response time is long.

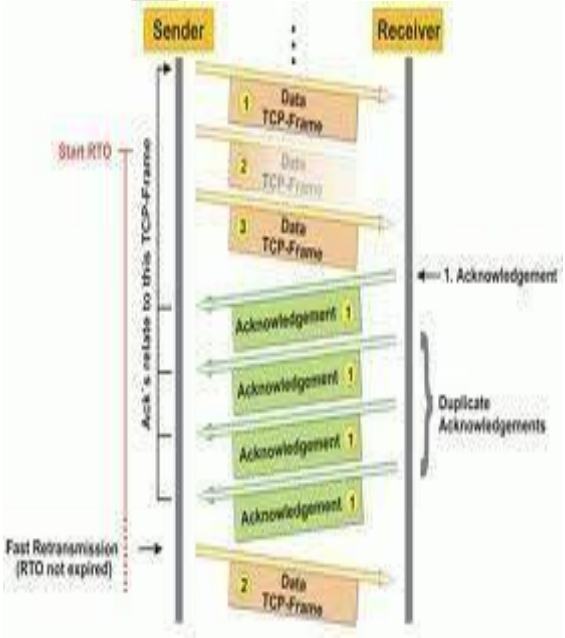
If a TCP data segment is lost in the network, a receiver will never even know it was once sent. However, the sender is waiting for an acknowledgement for that segment to return. In one case, if an acknowledgement doesn't return, the sender's retransmission timer expires which causes a retransmission of the segment. If however the sender had sent at least one additional segment after the one that was lost and that later segment is received correctly, the receiver does not send an acknowledgement for the later, out of order segment.

The receiver cannot acknowledgement out of order data; it must acknowledge the last contiguous byte it has received in the byte stream prior to the lost segment. In this case, the receiver will send an acknowledgement indicating the last contiguous byte it has received. If that last contiguous byte was already acknowledged, we call this a duplicate ACK. The reception of duplicate ACKs can implicitly tell the sender that a segment may have been lost or delayed. The sender knows this because the receiver only generates a duplicate ACK when it receives other, out of order segments. In fact, the Fast Retransmit algorithm described later uses duplicate ACKs as a way of speeding up the retransmission process.

A TCP sender uses a timer to recognize lost segments. If an acknowledgement is not received for a particular segment within a specified time (a function of the estimated Round-trip delay time), the sender will assume the segment was lost in the network, and will retransmit the segment.

Duplicate acknowledgement is the basis for the fast retransmit mechanism which works as follows: after receiving a packet (e.g. with sequence number 1), the receiver sends an acknowledgement by adding 1 to the sequence number (i.e., acknowledgement number 2) which means that the receiver receives the packet number 1 and it expects packet number 2 from the sender. Let's assume that three subsequent packets have been lost. In the meantime the receiver receives packet numbers 5 and 6. After receiving packet number 5, the receiver sends an acknowledgement, but still only for sequence number 2. When the receiver receives packet number 6, it sends yet another acknowledgement value of 2. Because the sender receives more than one acknowledgement with the same sequence number (2 in this example) this is called duplicate acknowledgement.

The fast retransmit enhancement works as follows: if a TCP sender receives a specified number of acknowledgements which is usually set to three duplicate acknowledgements with the same acknowledge number (that is, a total of four acknowledgements with the same acknowledgement number), the sender can be reasonably confident that the segment with the next higher sequence number was dropped, and will not arrive out of order. The sender will then retransmit the packet that was presumed dropped before waiting for its timeout.



TCP Congestion control:

The standard fare in TCP implementations today can be found in RFC 2581 [2]. This reference document specifies four standard congestion control algorithms that are now in common use. Each of the algorithms noted within that document was actually designed long before the standard was published [9], [11]. Their usefulness has passed the test of time.

The four algorithms, Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery are described below.

Slow Start

Slow Start, a requirement for TCP software implementations is a mechanism used by the sender to control the transmission rate, otherwise known as sender-based flow control. This is accomplished through the return rate of acknowledgements from the receiver. Inother words, the rate of acknowledgements returned by the receiver determine the rate at which the sender can transmit data.

When a TCP connection first begins, the Slow Start algorithm initializes a **congestion window** to one segment, which is the maximum segment size (MSS) initialized by the receiver during the connection establishment phase. When acknowledgements are returned by the receiver, the congestion window increases by one segment for each acknowledgement returned. Thus, the sender can transmit the minimum of the congestion window and the advertised window of the receiver, which is simply called the **transmission window**.

Slow Start is actually not very slow when the network is not congested and network response time is good. For example, the first successful transmission and acknowledgement of a TCP segment increases the window to two segments. After successful transmission of these two segments and acknowledgements completes, the window is increased to four segments. Then eight segments, then sixteen segments and so on, doubling from there on out up to the maximum window size advertised by the receiver or until congestion finally does occur.

At some point the congestion window may become too large for the network or network conditions may change such that packets may be dropped. Packets lost will trigger a timeout at the sender. When this happens, the sender goes into congestion avoidance mode as described in the next section.

SCTP

Stream Control Transmission Protocol (SCTP) is a new reliable, message-oriented transport layer protocol. SCTP, however, is mostly designed for Internet applications that have recently been introduced. These new applications need a more sophisticated service than TCP can provide. SCTP combines the best features of UDP and TCP. SCTP is a reliable message-oriented protocol.

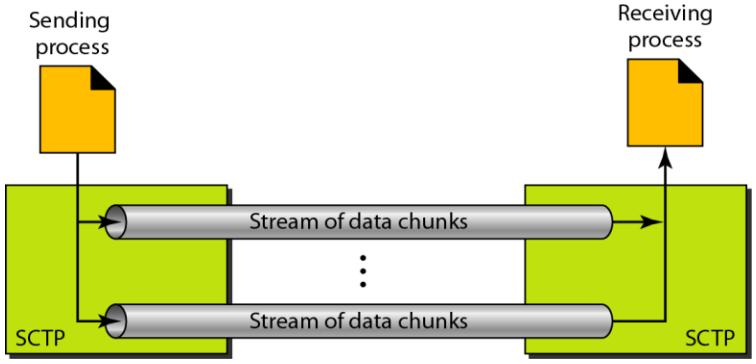
SCTP SERVICES

Process-to-Process Communication

SCTP uses all well-known ports in the TCP space. Table 23.4 lists some extra port numbers used by SCTP.

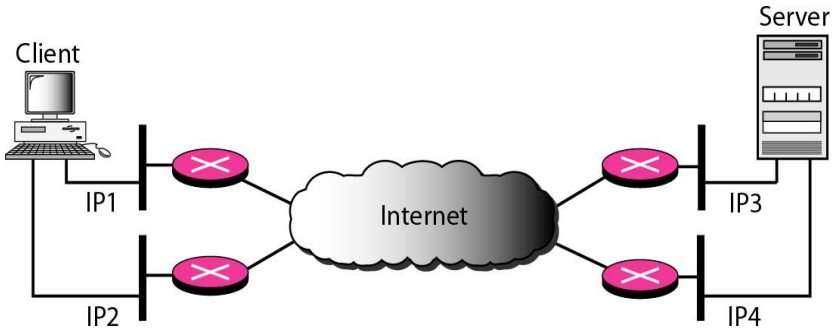
Multiple Streams

We learned in the previous section that TCP is a stream-oriented protocol. Each connection between a TCP client and a TCP server involves one single stream. The problem with this approach is that a loss at any point in the stream blocks the delivery of the rest of the data. This can be acceptable when we are transferring text; it is not when we are sending real-time data such as audio or video. SCTP allows multistream service in each connection, which is called association in SCTP terminology. If one of the streams is blocked, the other streams can still deliver their data.



Multihoming

The sending and receiving host can define multiple IP addresses in each end for an association. In this fault-tolerant approach, when one path fails, another interface can be used for data delivery without interruption. This fault-tolerant feature is very helpful when we are sending and receiving a real-time payload such as Internet telephony. Figure below shows the idea of multihoming.



In Figure above, the client is connected to two local networks with two IP addresses. The server is also connected to two networks with two IP addresses. The client and the server can make an association, using four different pairs of IP addresses

Full-Duplex Communication

Like TCP, SCTP offers full-duplex service, in which data can flow in both directions at the same time. Each SCTP then has a sending and receiving buffer, and packets are sent in both directions.

Connection-Oriented Service

Like TCP, SCTP is a connection-oriented protocol. However, in SCTP, a connection is called an association. When a process at site A wants to send and receive data from another process at site B, the following occurs:

1. The two SCTPs establish an association between each other.
2. Data are exchanged in both directions.
3. The association is terminated.

Reliable Service

SCTP, like 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.

SCTP FEATURES

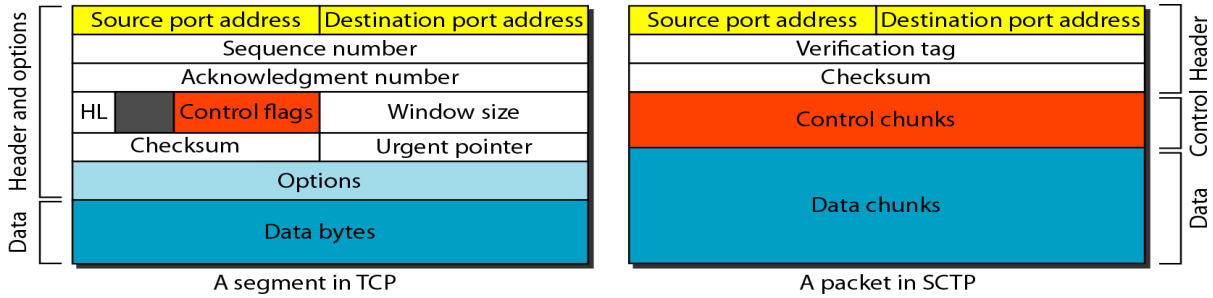
- Transmission Sequence Number: The unit of data in SCTP is a DATA chunk, Data transfer in SCTP is controlled by numbering the data chunks. SCTP uses a transmission sequence number (TSN) to number the data chunks (same as sequence number in TCP).
- Stream Identifier: In SCTP, there may be several streams in each association. Each stream in SCTP needs to be identified by using a stream identifier (SI). Each data chunk must carry the SI in its header

- Stream Sequence Number: When a data chunk arrives at the destination SCTP, it is delivered to the appropriate stream and in the proper order. This means that, in addition to an SI, SCTP defines each data chunk in each stream with a stream sequence number (SSN).

SCTP PACKETS

Data are carried as data chunks, control information is carried as control chunks. The SCTP packets are discussed in comparison with TCP packets.

1. The control information in TCP is part of the header; the control information in SCTP is included in the control chunks. There are several types of control chunks; each is used for a different purpose.



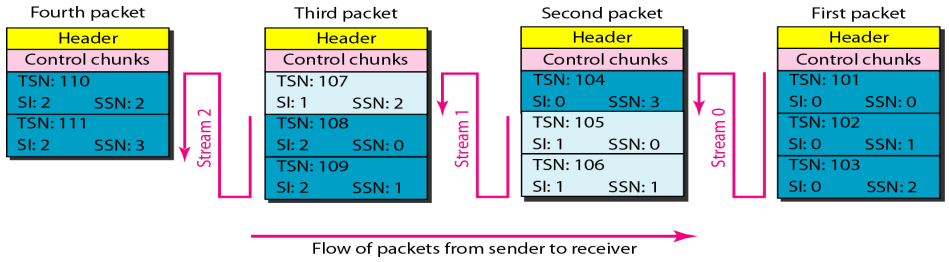
2. The data in a TCP segment treated as one entity; an SCTP packet can carry several data chunks; each can belong to a different stream.
3. The options section, which can be part of a TCP segment, does not exist in an SCTP packet. Options in SCTP are handled by defining new chunk types.
4. The mandatory part of the TCP header is 20 bytes, while the general header in SCTP is only 12 bytes. The SCTP header is shorter due to the following:
a. An SCTP sequence number (TSN) belongs to each data chunk and hence is located in the chunk's header.
b. The acknowledgment number and window size are part of each control chunk.
c. There is no need for a header length field (shown as HL in the TCP segment) because there are no options to make the length of the header variable; the SCTP header length is fixed (12 bytes).
d. There is no need for an urgent pointer in SCTP.
5. The checksum in TCP is 16 bits; in SCTP, it is 32 bits.
6. The verification tag in SCTP is an association identifier, which does not exist in TCP. In TCP, the combination of IP and port addresses defines a connection; in SCTP we may have multihorning using different IP addresses. A unique verification tag is needed to define each association.

7. TCP includes one sequence number in the header, which defines the number of the first byte in the data section. An SCTP packet can include several different data chunks. TSNs, SIs, and SSNs define each data chunk.
8. Some segments in TCP that carry control information (such as SYN and FIN) need to consume one sequence number; control chunks in SCTP never use a TSN, SI, or SSN. These three identifiers belong only to data chunks, not to the whole packet.

In SCTP, we have data chunks, streams, and packets. An association may send many packets, a packet may contain several chunks, and chunks may belong to different streams. To make the definitions of these terms clear, let us suppose that process A needs to send 11 messages to process B in three streams. The first four messages are in the first stream, the second three messages are in the second stream, and the last four messages are in the third stream.

Although a message, if long, can be carried by several data chunks, we assume that each message fits into one data chunk. Therefore, we have 11 data chunks in three streams. The application process delivers 11 messages to SCTP, where each message is earmarked for the appropriate stream. Although the process could deliver one message from the first stream and then another from the second, we assume that it delivers all messages belonging to the first stream first, all messages belonging to the second stream next, and finally, all messages belonging to the last stream.

We also assume that the network allows only three data chunks per packet, which means that we need. four packets as shown in Figure 23.30. Data chunks in stream 0 are carried in the first packet and part of the second packet; those in stream 1 are carried in the second and third packets; those in stream 2 are carried in the third and fourth packets.



Note that each data chunk needs three identifiers: TSN, SI, and SSN. TSN is a cumulative number and is used, as we will see later, for flow control and error control. SI defines the stream to which the chunk belongs. SSN defines the chunk's order in a particular stream. In our example, SSN starts from 0 for each stream.

Acknowledgment Number

TCP acknowledgment numbers are byte-oriented and refer to the sequence numbers. SCTP acknowledgment numbers are chunk-oriented. They refer to the TSN. A second difference between TCP and SCTP acknowledgments is the control information. Recall that this information is part of the segment header in TCP. To acknowledge segments that carry only control information, TCP uses a sequence number and acknowledgment number (for example, a SYN segment needs to be acknowledged by an ACK segment).

In SCTP, however, the control information is carried by control chunks, which do not need a TSN. These control chunks are acknowledged by another control chunk of the appropriate type (some need no acknowledgment). For example, an INIT control chunk is acknowledged by an INIT ACK chunk. There is no need for a sequence number or an acknowledgment number.

Flow Control

Like TCP, SCTP implements flow control to avoid overwhelming the receiver.

Error Control

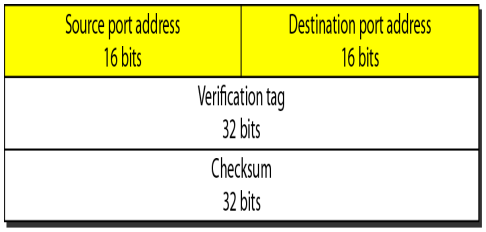
Like TCP, SCTP implements error control to provide reliability. TSN numbers and acknowledgment numbers are used for error control.

Congestion Control

Like TCP, SCTP implements congestion control to determine how many data chunks can be injected into the network.

PACKETS FORMAT

- An SCTP packet has a mandatory general header and a set of blocks called chunks. There are two types of chunks: control chunks and data chunks. A control chunk controls and maintains the association; a data chunk carries user data. The following fig shows the general header.
- **General Header:** The general header (packet header) defines the endpoints of each association to which the packet belongs, guarantees that the packet belongs to a particular association, and preserves the integrity of the contents of the packet including the header itself. The fonnat of the general header is given here.



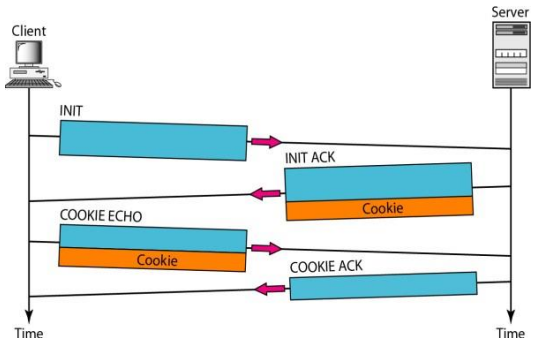
- **Source port address**. This is a 16-bit field that defines the port number of the process sending the packet.
- **Destination port address**. This is a 16-bit field that defines the port number of the process receiving the packet.
- **Verification tag**. This is a number that matches a packet to an association. This prevents a packet from a previous association from being mistaken as a packet in this association. It serves as an identifier for the association; it is repeated in every packet during the association. There is a separate verification used for each direction in the association.
- **Checksum.** This 32-bit field contains a CRC-32 checksum. Note that the size of the checksum is increased from 16 (in UDP, TCP, and IP) to 32 bits to allow the use of the CRC-32 checksum

An SCTP Association

SCTP, like TCP, is a connection-oriented protocol. However, a connection in SCTP is called an *association* to emphasize multihoming. An Association establishment in SCTP requires a four-way handshake. In this procedure, a process, normally a client, wants to establish an association with another process, normally a server, using SCTP as the transport layer protocol. Similar to TCP, the SCTP server needs to be prepared to receive any association (passive open). Association establishment, however, is initiated by the client (active open). SCTP association establishment is shown in Figure fig. The steps, in a normal situation, are as follows:

1. The client sends the first packet, which contains an INIT chunk.
2. The server sends the second packet, which contains an INIT ACK chunk.
3. The client sends the third packet, which includes a COOKIE ECHO chunk. This is a very simple chunk that echoes, without change, the cookie sent by the server. SCTP allows the inclusion of data chunks in this packet.
4. The server sends the fourth packet, which includes the COOKIE ACK chunk that acknowledges the receipt of the COOKIE ECHO chunk. SCTP allows the inclusion of data chunks with this packet.

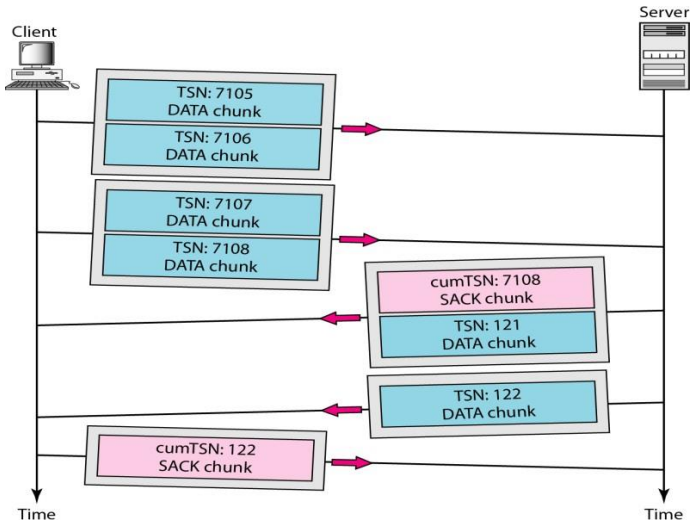
Cookie: To overcome sync flooding (malicious attack) designer has chosen cookie. After receiving init from client. The sever sends cookie to the client, The client runs the cookie and sends the response of cookie along with cookie for its identification.



Data Transfer

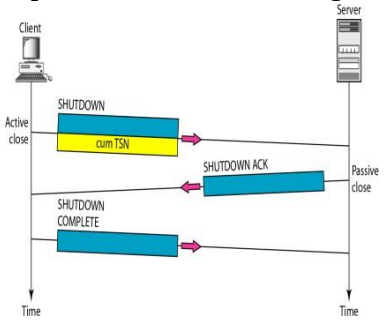
The whole purpose of an association is to transfer data between two ends. After the association is established, bidirectional data transfer can take place. The client and the server can both send data. Like TCP, SCTP supports piggybacking. For better illustration consider following figure.

1. The client sends the first packet carrying two DATA chunks with TSNs 7105 and 7106.
2. The client sends the second packet carrying two DATA chunks with TSNs 7107 and 7108.
3. The third packet is from the server. It contains the SACK chunk needed to acknowledge the receipt of DATA chunks from the client. Contrary to TCP, SCTP acknowledges the last in-order TSN received, not the next expected. The third packet also includes the first DATA chunk from the server with TSN 121.
4. After a while, the server sends another packet carrying the last DATA chunk with TSN 122, but it does not include a SACK chunk in the packet because the last DATA chunk received from the client was already acknowledged.
5. Finally, the client sends a packet that contains a SACK chunk acknowledging the receipt of the last two DATA chunks from the server.



Association Termination

In SCTP, like TCP, either of the two parties involved in exchanging data (client or server) can close the connection. However, unlike TCP, SCTP does not allow a halfclose situation. If one end closes the association, the other end must stop sending new data. If any data are left over in the queue of the recipient of the termination request, they are sent and the association is closed. Association **termination** uses three packets, as shown in Figure below.



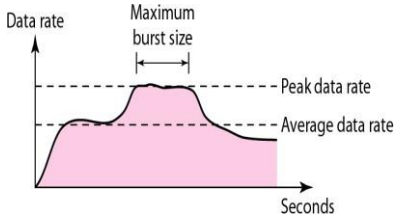
Note: Flow control and error control same as in TCP IP (Refer TCP)

DATA TRAFFIC

- The main focus of congestion control and quality of service is data traffic. In congestion control we try to avoid traffic congestion. In quality of service, we try to create an appropriate environment for the traffic. So, before talking about congestion control and quality of service.

PARAMETERS OF DATA TRAFFIC

- **Traffic Descriptor:** Traffic descriptors are qualitative values that represent a data flow

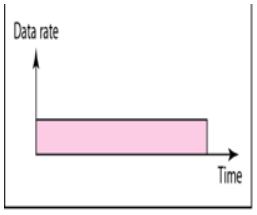


- **Average Data Rate:** The average data rate is the number of bits sent during a period of time, divided by the number of seconds in that period.
- **Peak Data Rate:** The peak data rate defines the maximum data rate of the traffic. In Figure above it is the maximum y axis value.
- **Maximum Burst Size:** Although the peak data rate is a critical value for the network, it can usually be ignored if the duration of the peak value is very short.

- **Effective Bandwidth:** The effective bandwidth is the bandwidth that the network needs to allocate for the flow of traffic. The effective bandwidth is a function of three values: average data rate, peak data rate, and maximum burst size.

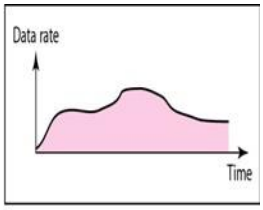
TRAFFIC PROFILES

- **Constant Bit Rate:** A constant-bit-rate (CBR), or a fixed-rate, traffic model has a data rate that does not change. In this type of flow, the average data rate and the peak data rate are the same. The maximum burst size is not applicable This type of traffic is very easy to handle since network knows how much bandwidth to allocate for this type of flow.



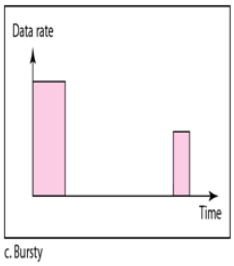
a. Constant bit rate

- In the **variable-bit-rate (VBR),** the rate of the data flow changes in time, with the changes smooth instead of sudden and sharp. Here the type of flow, the average data rate and the peak data rate are different. The maximum burst size is usually a small value This type of traffic is more difficult to handle than constant-bit-rate traffic, but it normally does not need to be reshaped.



b. Variable bit rate

- In the **bursty data** category, the data rate changes suddenly in a very short time. It may jump from zero, for example, to 1 Mbps in a few microseconds and vice versa. The average bit rate and the peak bit rate are very different values in this type of flow. The maximum burst size is significant. This is the most difficult type of traffic for a network to handle because the profile is very unpredictable. To handle this type of traffic, the network normally needs to reshape it, using reshaping techniques unpredictable.



CONGESTION

- An important issue in a packet-switched network is congestion. Congestion in a network may occur if the number of packets sent to the network-is greater than 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

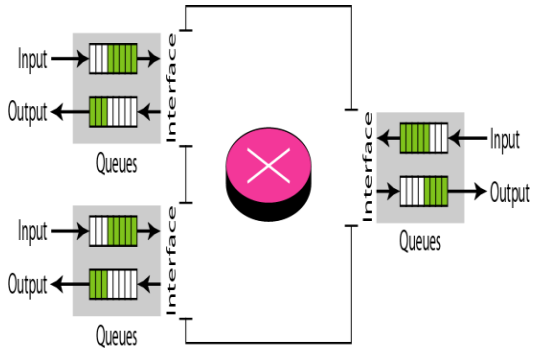
Reasons for existence of congestion:

- congestion happens on a freeway because any abnonnality in the flow.
- Congestion in a network or internetwork occurs because routers and switches have queues-buffers that hold the packets before and after processing
- When a packet arrives at the incoming interface, it undergoes three steps before departing, as shown in Figure.

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.

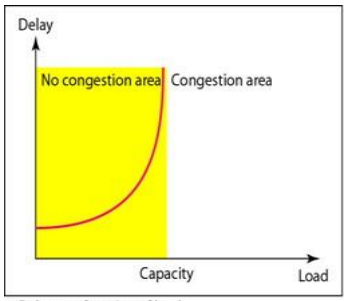


Awareness:

- 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

NETWORKPER FORMANCE

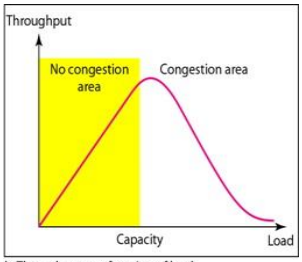
- Note that when the load is much less than the capacity of the network, the delay is at a minimum. This minimum delay is composed of propagation delay and processing delay, both of which are negligible.
- However, when the load reaches the network capacity, the delay increases sharply because we now need to add the waiting time in the queues (for all routers in the path) to the total delay. Note that the delay becomes infinite when the load is greater than the capacity
- Delay has a negative effect on the load and consequently the congestion.



a. Delay as a function of load

- We can define throughput in a network as the number of packets passing through the network in a unit of time.
- Notice that when the load is below the capacity of the network, the throughput increases proportionally with the load. We expect the throughput to remain constant after the load reaches the capacity, but instead the throughput declines sharply.
- The reason is the discarding of packets by the routers. When the load exceeds the capacity, the queues become full and the routers have to discard some packets.

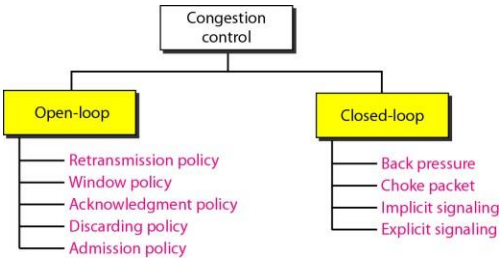
- Discarding packets does not reduce the number of packets in the network because the sources retransmit the packets, using time-out mechanisms, when the packets do not reach the destinations.



b. Throughput as a function of load

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).



OPEN LOOP CONTROL

- **Retransmission Policy:** Retransmission is sometimes unavoidable. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. Retransmission in general may increase congestion in the network. However, a good retransmission policy can prevent congestion. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion.

- **Window Policy:** The type of window at the sender may also affect congestion. 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.

- **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 less load on the network.

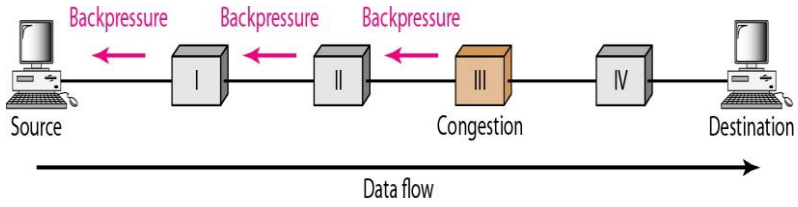
- **Discarding Policy:** A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission. For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound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

- **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.

CLOSED LOOP CONTROL

- 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.

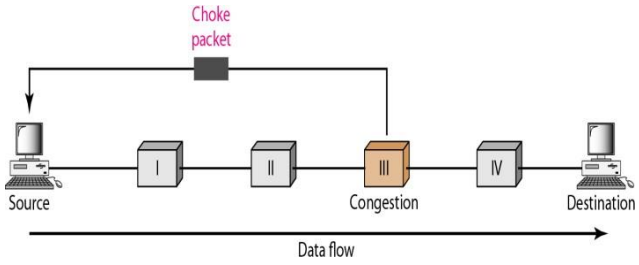
- **Backpressure** is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source. The backpressure technique can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is corning. Figure 24.6 shows the idea of backpressure.



- **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.

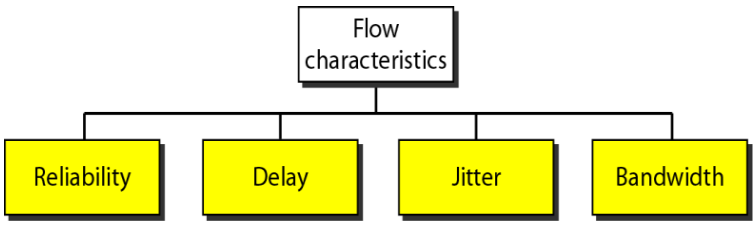
- In the choke packet method, the warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet has traveled are not warned.



- **Implicit Signaling:** In implicit signaling, there is no communication between the congested node or nodes and the source.
- The source guesses that there is a 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.
- **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, congestion control, can occur in either the forward or the backward direction.
- **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.
- **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.

QUALITY OF SERVICE

Quality of service (QoS) is an internetworking issue that has been discussed more than defined. We can informally define quality of service as something a flow seeks to attain. The QoS can be judged by flow characteristics and flow class. The following fig. shows different flow characteristics.



- **Reliability:** Reliability is a characteristic that a flow needs. Lack of reliability means losing a packet or acknowledgment, which entails retransmission. However, the sensitivity of application programs to reliability is not the same.
- **Delay:** Source-to-destination delay is another flow characteristic. Again applications can tolerate delay in different degrees.
- **Jitter:** Jitter is the variation in delay for packets belonging to the same flow. Jitter is defined as the variation in the packet delay. High jitter means the difference between delays is large; low jitter means the variation is small.
    - For example, if four packets depart at times 0, 1, 2, 3 and arrive at 20, 21, 22, 23, all have the same delay, 20 units of time. On the other hand, if the above four packets arrive at 21, 23, 21, and 28, they will have different delays: 21,22, 19, and 24.
- **Bandwidth:** Different applications need different bandwidths. In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an e-mail may not reach even a million.
- **Flow Classes-** based flow characteristic, we can classify flows into groups, with each group similar characteristics. These classes will be used by other protocols such as ATM (Not there in syllabus)

1 INTEGRATED SERVICES

- Two models have been designed to provide quality of service in the Internet: Integrated Services and Differentiated Services. Both models emphasize the use of quality of service at the network layer (IP).

- IP was originally designed for best-effort delivery. Which means that every user receives the same level of services. This type of delivery does not guarantee the minimum of a service, such as bandwidth, to applications such as real-time audio and video
- If such an application accidentally gets extra bandwidth, it may be detrimental to other applications, resulting in congestion.
- Integrated Services, sometimes called IntServ, is afiow-based QoS model, which means that a user needs to create a flow, a kind of virtual circuit, from the source to the destination
- Signaling: IP is a connectionless, datagram, packet-switching protocol. we can implement a flow-based model over a IP signaling protocol for making a reservation. This protocol is called Resource Reservation Protocol (RSVP).
- Flow Specification: When a source makes a reservation, it needs to define a flow specification. A flow spedfication has two parts: Rspec (resource specification) and Tspec (traffic specification).
- Rspec defines the resource that the flow needs to reserve (buffer, bandwidth, etc.).
- Tspec defines the traffic characterization of the flow.
- Admission: After a router receives the flow specification from an application, it decides to admit or deny the service.
- The decision is based on the previous commitments of the router and the current availability of the resource.

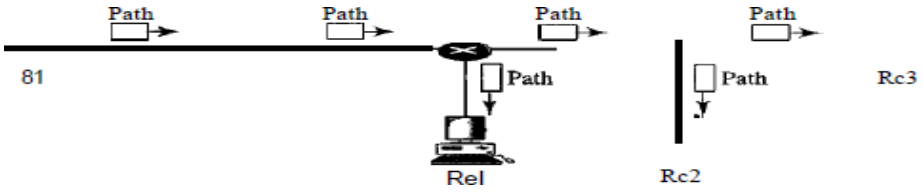
Service Classes

- Guaranteed Service Class: This type of service is designed for real-time traffic that needs a guaranteed minimum end-to-end delay. The end-to-end delay is the sum of the delays in the routers.
- Only the first, the sum of the delays in the routers, can be guaranteed by the router. This type of service guarantees the packets within delivery time. (sum of delays of routers= Tspec)
- Controlled-Load Service Class: This type of service is designed for applications that can accept some delays, but are sensitive to an overloaded network and to the danger of losing packets.

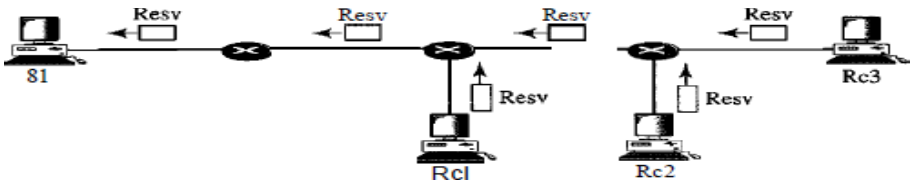
- Good examples of these types of applications are file transfer, e-mail, and Internet access. The controlled load service is a qualitative type of service in that the application requests the possibility of low-loss or no-loss packets.

RSVP

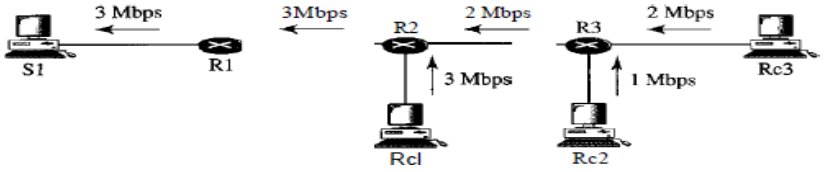
- The Resource Reservation Protocol (RSVP) is a signaling protocol to help IP create a flow and consequently make a resource reservation.
- Multicast Trees: RSVP is different signaling system designed for multicasting. It can also used for unicasting. The reason for this design is to enable RSVP to provide resource reservations for all kinds of traffic including multimedia which often uses multicasting.
- Receiver-Based Reservation: In RSVP, the receivers, not the sender, make the reservation. This strategy matches the other multicasting protocols
- RSVP Messages: RSVP has several types of messages. However, for our purposes, we discuss only two of them: Path and Resv.
- Path Messages: Receivers make the reservation using RSVP. However, the receivers do not know the path traveled by packets before the reservation is made. The path is needed for the reservation.
- To solve the problem, RSVP uses Path messages. A Path message travels from the sender and reaches all receivers in the multicast path. On the way, a Path message stores the necessary information for the receivers.
- A Path message is sent in a multicast environment; a new message is created when the path diverges. Figure below shows path messages.



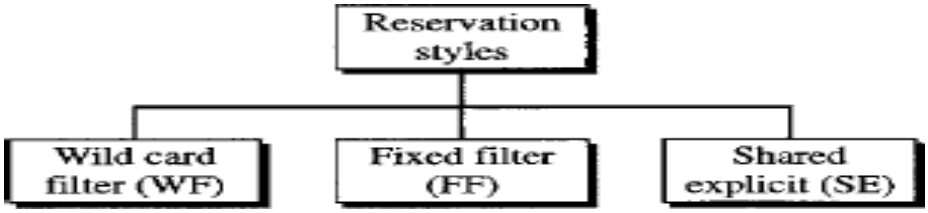
- Resv Messages: After a receiver has received a Path message, it sends a Resv message.
- The Resv message travels toward the sender (upstream) and makes a resource reservation on the routers that support RSVP.
- If a router does not support RSVP on the path, it routes the packet based on the best-effort delivery. Below Fig. shows the Resv messages.



- Reservation Merging: In RSVP, the resources are not reserved for each receiver in a flow; the reservation is merged.
- In Figure, Rc3 requests a 2-Mbps bandwidth while Rc2 requests a I-Mbps bandwidth. Router R3, which needs to make a bandwidth reservation, merges the two requests.
- The reservation is made for 2 Mbps, the larger of the two, because a 2-Mbps input reservation can handle both requests. The same situation is true for R2.



- Reservation Styles: When there is more than one flow, the router needs to make a reservation to accommodate all of them. RSVP defines three types of reservation styles, as shown in Fig.



- Wild Card Filter: Style In this style, the router creates a single reservation for all senders. The reservation is based on the largest request. This type of style is used when the flows from different senders do not occur at the same time.
- Fixed Filter Style: In this style, the router creates a distinct reservation for each flow. This means that if there are n flows, n different reservations are made.
- This type of style is used when there is a high probability that flows from different senders will occur at the same time.
- Shared Explicit: Style In this style, the router creates a single reservation which can be shared by a set of flows.

- Soft State: The reservation information (state) stored in every node for a flow needs to be refreshed periodically. This is referred to as a soft state. The default interval for refreshing is currently 30 s.

Problems with Integrated Services

- There are at least two problems with Integrated Services that may prevent its full implementation in the Internet: scalability and service-type limitation.
- Scalability: The Integrated Services model requires that each router keep information for each flow. As the Internet is growing every day, this is a serious problem.
- Service-Type Limitation: The Integrated Services model provides only two types of services, guaranteed and control-load. Those opposing this model argue that applications may need more than these two types of services.

2 DIFFERENTIATED SERVICES

- Differentiated Services (DS or Diffserv) was introduced by the IETF (Internet Engineering Task Force) to handle the shortcomings of Integrated Services. Two fundamental changes were made:

1. The main processing was moved from the core of the network to the edge of the network. This solves the scalability problem.

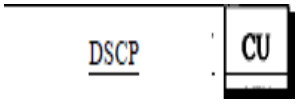
-The routers do not have to store information about flows. The applications, or hosts, define the type of service they need each time they send a packet.

2. The per-flow service is changed to per-class service. The router routes the packet based on the class of service defined in the packet, not the flow.

-This solves the service-type limitation problem. We can define different types of classes based on the needs of applications.

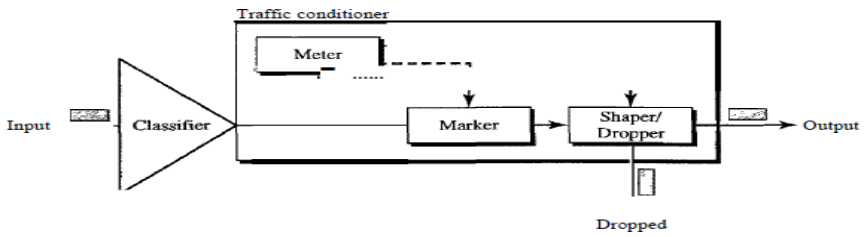
DS Field

- In Diffserv, each packet contains a field called the DS field. The value of this field is set at the boundary of the network by the host or the first router designated as the boundary router as shown in Fig.



- The DS field contains two subfields: DSCP and CU.

- The DSCP (Differentiated Services Code Point) is a 6-bit subfield that defines the per-hop behavior (PHB).
- The 2-bit CU (currently unused) subfield is not currently used.
- The Diffserv capable node (router) uses the DSCP 6 bits as an index to a table defining the packet-handling mechanism for the current packet being processed.
- Per-Hop Behavior: The Diffserv model defines per-hop behaviors (PHBs) for each node that receives a packet. So far three PHBs are defined: DE PHB, EF PHB, and AF PHB.
- DE PHB: The DE PHB (default PHB) is the same as best-effort delivery, which is compatible with TOS.
- EF PHB The EF PHB (expedited forwarding PHB) provides the services like low loss, low latency ensured bandwidth. This is the same as having a virtual connection between the source and destination.
- AF PHB The AF PHB (assured forwarding PHB) delivers the packet with a high assurance as long as the class traffic does not exceed the traffic profile of the node. The users of the network need to be aware that some packets may be discarded.
- Traffic Conditioner: To implement Oiffserv, the OS node uses traffic conditioners such as meters, markers, shapers, and droppers, as shown in Fig.

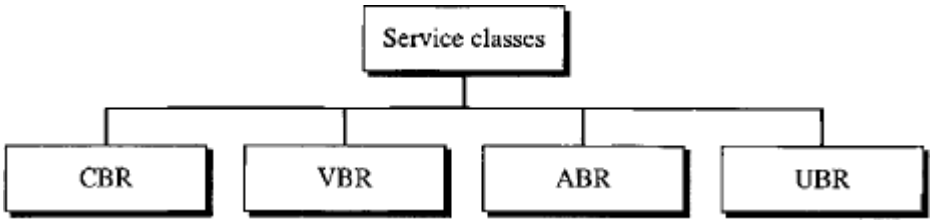


- Meters: The meter checks to see if the incoming flow matches the negotiated traffic profile. The meter also sends this result to other components. The meter can use several tools such as a token bucket to check the profile.
- Marker: A marker can remark a packet that is using best-effort delivery (OSCP: 000000) or down-mark a packet based on information received from the meter. Downmarking (lowering the class of the flow) occurs if the flow does not match the profile. A marker does not up-mark (promote the class) a packet.
- Shaper: A shaper uses the information received from the meter to reshape the traffic if it is not compliant with the negotiated profile.

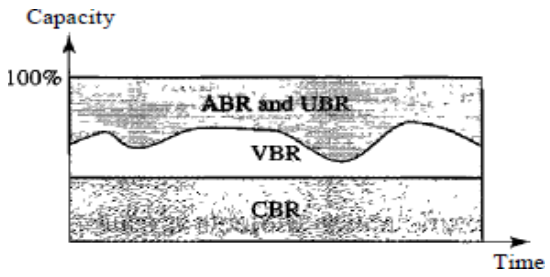
- Dropper: A dropper, which works as a shaper with no buffer, discards packets if the flow severely violates the negotiated profile.

QoSinATM

- The QoS in ATM is based on the class, user-related attributes, and network-related attributes.
- The ATM Forum defines four service classes: CBR, VBR, ABR, and UBR



- CBR The constant-bit-rate (CBR) class is designed for customers who need realtime audio or video services.
- VBR The variable-bit-rate (VBR) class is divided into two subclasses: real-time(VBR-RT) and non-real-time (VBR-NRT).
- VBR-RT is designed for those users who need real-time services (such as voice and video transmission) and use compression techniques to create a variable bit rate. VBR-NRT is designed for those users who do not need real-time services but use compression techniques to create a variable bit rate.
- ABR The available-bit-rate (ABR) class delivers cells at a minimum rate. If more network capacity is available, this minimum rate can be exceeded. ABR is particularly suitable for applications that are bursty.
- UBR The unspecified-bit-rate (UBR) class is a best-effort delivery service that does not guarantee anything. Below fig. Show relationship between different classes to the capacity of the network



ATM Attributes

- ATM defines two sets of attributes. User-related attributes and network related attributes.
- User-related attributes are those attributes that define how fast the user wants to send data. These are negotiated at the time of contract between a user and a network. Few of user-related attributes are given
- SCR The sustained cell rate (SCR) is the average cell rate over a long time interval. The actual cell rate may be lower or higher than this value, but the average should be equal to or less than the SCR
- PCR The peak cell rate (PCR) defines the sender's maximum cell rate. The user‘s cell rate can sometimes reach this peak, as long as the SCR is maintained.
- MCR The minimum cell rate (MCR) defines the minimum cell rate acceptable to the sender.
- CVDT The cell variation delay tolerance (CVDT) is a measure of the variation in cell transmission times.
- Network-Related Attributes: The network-related attributes are those that define characteristics of the network. The following are some network-related attributes
- CLR The cell loss ratio (CLR) defines the fraction of cells lost (or delivered so late that they are considered lost) during transmission.

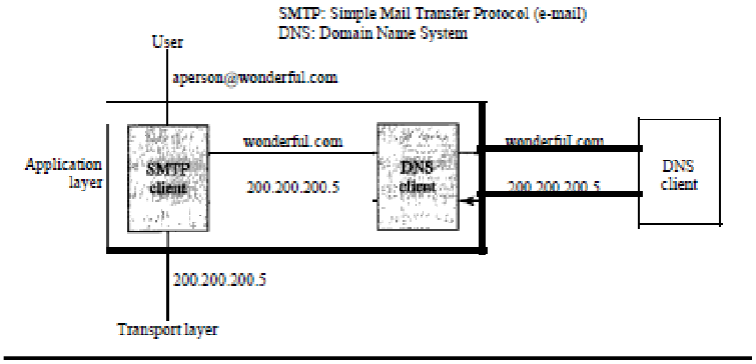
$$CLR = \frac{1}{100} = 10^{-2}$$

- CTD The cell transfer delay (CTD) is the average time needed for a cell to travel from source to destination. The maximum CTD and the minimum CTD are also considered attributes.
- CDV The cell delay variation (CDV) is the difference between the CTD maximum and the CTD minimum.
- CER The cell error ratio (CER) defines the fraction of the cells delivered in error.

APPLICATION LAYER

1. DOMAIN NAME SYSTEM

The Domain Name System (DNS) is a supporting program that is used by other programs such as e-mail. Figure 25.1 shows an example of how a DNS client/server program can support an e-mail program to find the IP address of an e-mail recipient. A user of an e-mail program may know the e-mail address of the recipient; however, the IP protocol needs the IP address. The DNS client program sends a request to a DNS server to map the e-mail address to the corresponding IP address.



NAME SPACE: A name space that maps each address to a unique name can be organized in two ways: fiat or hierarchical.

Flat Name Space

In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure. The names mayor may not have a common section; if they do, it has no meaning. The main disadvantage of a fiat name space is that it cannot be used in a large System such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

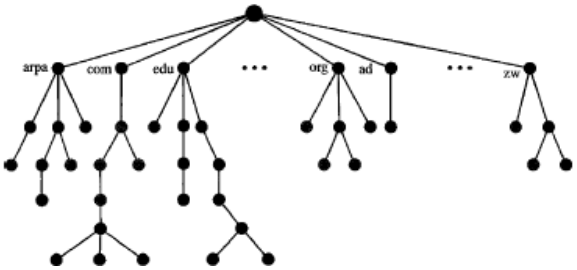
Hierarchical Name Space

In a hierarchical name space, each name is made of several parts. The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on

DOMAIN NAME SPACE:

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127 (see Figure 25.2).

Figure 25.2 *Domain name space*



Label

Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

Domain Names

Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.). The domain names are always read from the node up to the root. The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing. Figure 25.3 shows some domain names.

Fully Qualified Domain Name

If a label is terminated by a null string, it is called a fully qualified domain name (FQDN). An FQDN is a domain name that contains the full name of a host. For example, the domain name: challenger.ate.tbda.edu.

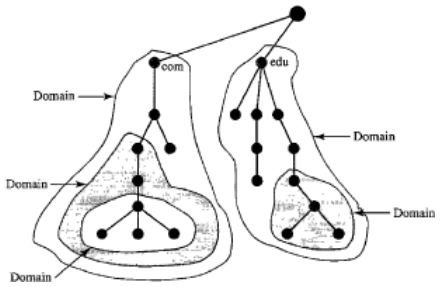
Partially Qualified Domain Name

If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN). A PQDN starts from a node, but it does not reach the root. It is used when the name to be resolved belongs to the same site as the client. For example, if a user at the *jhda.edu.* site wants to get the IP address of the challenger computer, he or she can define the partial name

Figure 25.4 shows some FQDNs and PQDNs.

Figure 25.4 *FQDN and PQDN*

Domain: A **domain** is a subtree of the domain name space. The name of the domain is the domain name of the node at the top of the subtree. Figure 25.5 shows some domains. Note that a domain may itselfbe divided into domains (or **subdomains** as they are sometimes called).

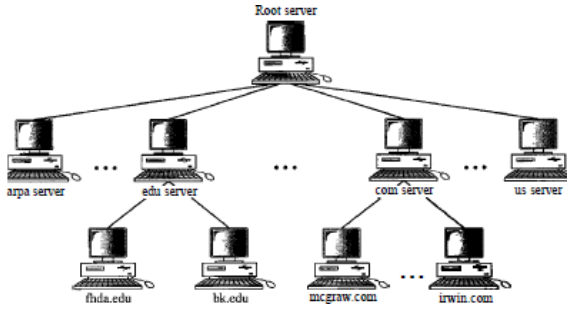


DISTRIBUTION OF NAME SPACE

The information contained in the domain name space must be stored. However, it is very inefficient and also unreliable to have just one computer store such a huge amount of information. It is inefficient because responding to requests from all over the world places a heavy load on the system. It is not unreliable because any failure makes the data inaccessible.

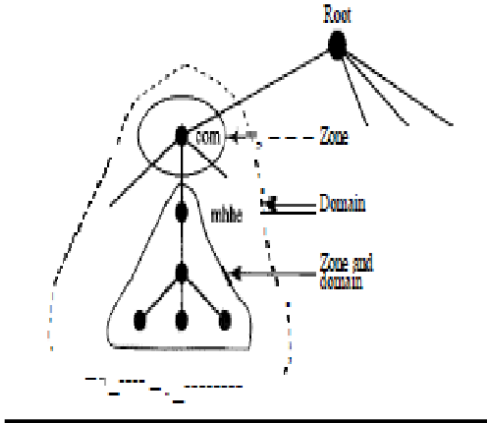
Hierarchy of Name Servers

The solution to these problems is to distribute the information among many computers called DNS servers. One way to do this is to divide the whole space into many domains based on the first level. In other words, we let the root stand alone and create as many domains (subtrees) as there are first-level nodes. Because a domain created in this way could be very large, DNS allows domains to be divided further into smaller domains (subdomains). Each server can be responsible (authoritative) for either a large or a small domain. In other words, we have a hierarchy of servers in the same way that we have a hierarchy of names (see Figure 25.6).



Zone

Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers. What a server is responsible for or has authority over is called a zone. We can define a zone as a contiguous part of the entire tree. If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the *domain* and the *zone* refer to the same thing. The server makes a database called a *zone file* and keeps all the information for every node under that domain. However, if a server divides its domain into subdomains and delegates part of its authority to other servers, *domain* and *zone* refer to different things. The information about the nodes in the subdomains is stored in the servers at the lower levels, with the original server keeping some sort of reference to these lower-level servers. Of course the original server does not free itself from responsibility totally: It still has a zone, but the detailed information is kept by the lower-level servers (see Figure 25.7). A server can also divide part of its domain and delegate responsibility but still keep part of the domain for itself. In this case, its zone is made of detailed information for the part of the domain that is not delegated and references to those parts that are delegated.



Root Server

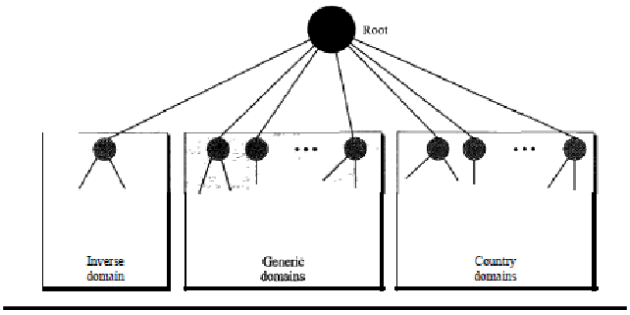
A root server is a server whose zone consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers. There are several root servers, each covering the whole domain name space. The servers are distributed all around the world.

Primary and Secondary Servers

DNS defines two types of servers: primary and secondary. A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk. A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If updaJing is required, it must be done by the primary server, which sends the updated version to the secondary. The primary and secondary servers are both authoritative for the zones they serve. The idea is not to put the secondary server at a lower level of authority but to create redundancy for the data so that if one server fails, the other can continue serving clients. Note also that a server can be a primary server for a specific zone and a secondary server for another zone. Therefore, when we refer to a server as a primary or secondary server, we should be careful to which zone we refer

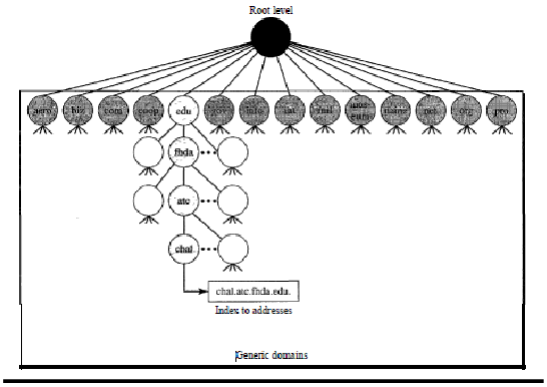
2. DNS IN THE INTERNET

DNS is a protocol that can be used in different platforms. In the Internet, the domain name space (tree) is divided into three different sections: generic domains, country domains, and the inverse domain.



Generic Domains

The **generic domains** define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database (see Figure 25.9).



Looking at the tree, we see that the first level in the generic domains section allows 14 possible labels. These labels describe the organization types as listed in Table 25.1.

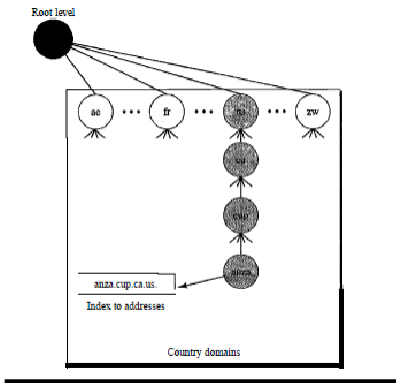
Table 25.1 *Generic domain labels*

Label	*Description*
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other nonprofit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

Country Domains

The country domains section uses two-character country abbreviations (e.g., us for United States). Second labels can be organizational, or they can be more specific, national designations. The United States, for example, uses state abbreviations as a subdivision of us (e.g., ca.us.).

figure 25.10 shows the country domains section. The address *anza.cup.ca.us* can be translated to De Anza College in Cupertino, California, in the United States.



Inverse Domain

The inverse domain is used to map an address to a name. This may happen, for example, when a server has received a request from a client to do a task. Although the server has a file that contains a list of authorized clients, only the IP address of the client (extracted from the received IP packet) is listed. The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the authorized list. This type of query is called an inverse or pointer (PTR) query. To handle a pointer query, the inverse domain is added to the domain name space with the first-level node called *arpa* (for historical reasons). The second level is also one single node named *in-addr* (for inverse address). The rest of the domain defines IP addresses. The servers that handle the inverse domain are also hierarchical. This means the netid part of the address should be at a higher level than the subnetid part, and the subnetid part higher than the hostid part. In this way, a server serving the whole site is at a higher level than the servers serving each subnet. This configuration makes the domain look inverted when compared to a generic or country domain. To follow the convention of reading the domain labels from the bottom to the top, an IF address such as 132.34.45.121 (a class B address with netid 132.34) is read as 121.45.34.132.in-addr. arpa. See Figure 25.11 for an illustration of the inverse domain configuration.

3. ELECTRONIC MAIL

One of the most popular Internet services is electronic mail (e-mail). The designers of the Internet probably never imagined the popularity of this application program. At the beginning of the Internet era, the messages sent by electronic mail were short and consisted of text only; they

let people exchange quick memos. Today, electronic mail is much more complex. It allows a message to include text, audio, and video. It also allows one message to be sent to one or more recipients.

Architecture

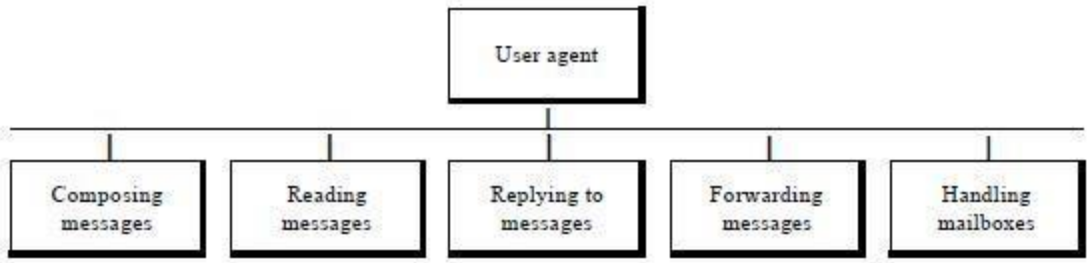
User Agent

The first component of an electronic mail system is the user agent. It provides service to the user to make the process of sending and receiving a message easier.

Services Provided by a User Agent

A user agent is a software package (program) that composes, reads, replies to, and forwards messages. It also handles mailboxes. Figure 26.11 shows the services of a typical user agent.

Figure 26.11 *Services of user agent*



Composing Messages

A user agent helps the user compose the e-mail message to be sent out. Most user agents provide a template on the screen to be filled in by the user. Some even have a built-in editor that can do spell checking, grammar checking, and other tasks expected from a sophisticated word processor. A user, of course, could alternatively use his or her favorite text editor or word processor to create the message and import it, or cut and paste it, into the user agent template.

Reading Messages

The second duty of the user agent is to read the incoming messages. When a user invokes a user agent, it first checks the mail in the incoming mailbox. Most user agents show a one-line summary of each received mail. Each e-mail contains the following fields.

1. A number field.
2. A flag field that shows the status of the mail such as new, already read but not replied to, or read and replied to.

3. The size of the message.

4. The sender.

5. The optional subject field.

Replying to Messages

After reading a message, a user can use the user agent to reply to a message. A user agent usually allows the user to reply to the original sender or to reply to all recipients of the message. The reply message may contain the original message and the new message.

Forwarding Messages

Replying is defined as sending a message to the sender or recipients of the copy. *Forwarding* is defined as sending the message to a third party. A user agent allows the receiver to forward the message, with or without extra comments, to a third party.

Handling Mailboxes

A user agent normally creates two mailboxes: an inbox and an outbox. Each box is a file with a special format that can be handled by the user agent. The inbox keeps all the received e-mails until they are deleted by the user. The outbox keeps all the sent e-mails until the user deletes them. Most user agents today are capable of creating customized mailboxes.

User Agent Types

There are two types of user agents: command-driven and GUI-based.

Command-Driven: Command-driven user agents belong to the early days of electronic mail. They are still present as the underlying user agents in servers. A command-driven user agent normally accepts a one-character command from the keyboard to perform its task. For example,

a user can type the character r, at the command prompt, to reply to the sender of the message, or type the character R to reply to the sender and all recipients. Some examples of command-driven user agents are *mail, pine,* and *elm.*

GUI-Based: Modern user agents are GUI-based. They contain graphical-user interface (GUI) components that allow the user to interact with the software by using both the keyboard and the mouse. They have graphical components such as icons, menu bars, and windows that make the services easy to access. Some examples of GUI-based user agents are Eudora, Microsoft's Outlook, and Netscape.

Addresses

To deliver mail, a mail handling system must use an addressing system with unique addresses. In the Internet, the address consists of two parts: a local part and a domain name, separated by an @

sign

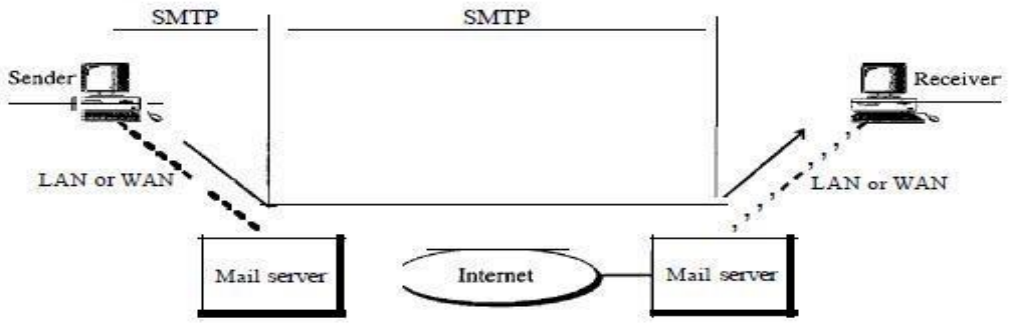
Local Part: The local part defines the name of a special file, called the user mailbox, where all the mail received for a user is stored for retrieval by the message access agent.

Domain Name: The second part of the address is the domain name. An organization usually selects one or more hosts to receive and send e-mail; the hosts are sometimes called *mail servers* or *exchangers*. The domain name assigned to each mail exchanger either comes from the DNS database or is a logical name (for example, the name of the organization).

SMTP

The actual mail transfer is done through message transfer agents. To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The formal protocol that defines the MTA client and server in the Internet is called the Simple Mail Transfer Protocol (SMTP).

Figure 26.16 *SMTP range*

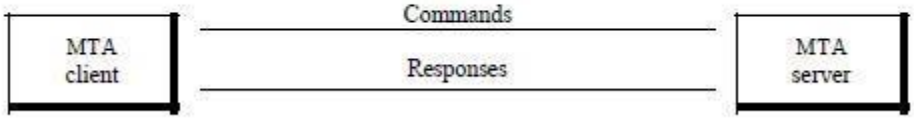


SMTP is used two times, between the sender and the sender's mail server and between the two mail servers. SMTP simply defines how commands and responses must be sent back and forth.

Commands and Responses

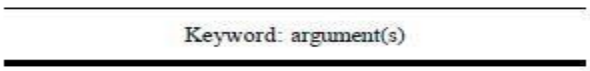
SMTP uses commands and responses to transfer messages between an MTA client and an MTA server (see Figure 26.17).

Figure 26.17 *Commands and responses*



Commands: Commands are sent from the client to the server. The format of a command is shown in Figure 26.18. It consists of a keyword followed by zero or more arguments. SMTP defines 14 commands. The first five are mandatory; every implementation must support these five commands. The next three are often used and highly recommended. The last six are seldom used.

Figure 26.18 *Commandformat*



The commands are listed in Table 26.7.

Table 26.7 *Commands*

Keyword	*Argument(s)*
HELO	Sender's host name
MAIL FROM	Sender of the message
RCPTTO	Intended recipient of the message
DATA	Body of the mail
QUIT	
RSET	
VRFY	Name of recipient to be verified
NOOP	
TURN	
EXPN	Mailing list to be expanded
HELP	Command name

Responses: Responses are sent from the server to the client. A response is a three digit code that may be followed by additional textual information. Table 26.8 lists some of the responses.

Table 26.8 *Responses*

Code	*Description*
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input

Mail Transfer Phases

The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

POP3

Post Office Protocol, version 3 (POP3) is simple and limited in functionality. The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server. Mail access starts with the client when the user needs to download e-mail from the mailbox on the mail server. The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox. The user can then list and retrieve the mail messages, one by one.

POP3 has two modes: the delete mode and the keep mode. In the delete mode, the mail is deleted from the mailbox after each retrieval. In the keep mode, the mail remains in the mailbox after retrieval. The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying. The keep mode is normally used when the user accesses her mail away from her primary computer (e.g., a laptop). The mail is read but kept in the system for later retrieval and organizing.

IMAP

Another mail access protocol is Internet Mail Access Protocol, version 4 (IMAP4). IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex.

POP3 is deficient in several ways. It does not allow the user to organize her mail on the server; the user cannot have different folders on the server. In addition, POP3 does not allow the user to partially check the contents of the mail before downloading. IMAP4 provides the following extra functions:

o A user can check the e-mail header prior to downloading.

o A user can search the contents of the e-mail for a specific string of characters prior to downloading.

o A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.

o A user can create, delete, or rename mailboxes on the mail server.

o A user can create a hierarchy of mailboxes in a folder for e-mail storage.

2. FTP

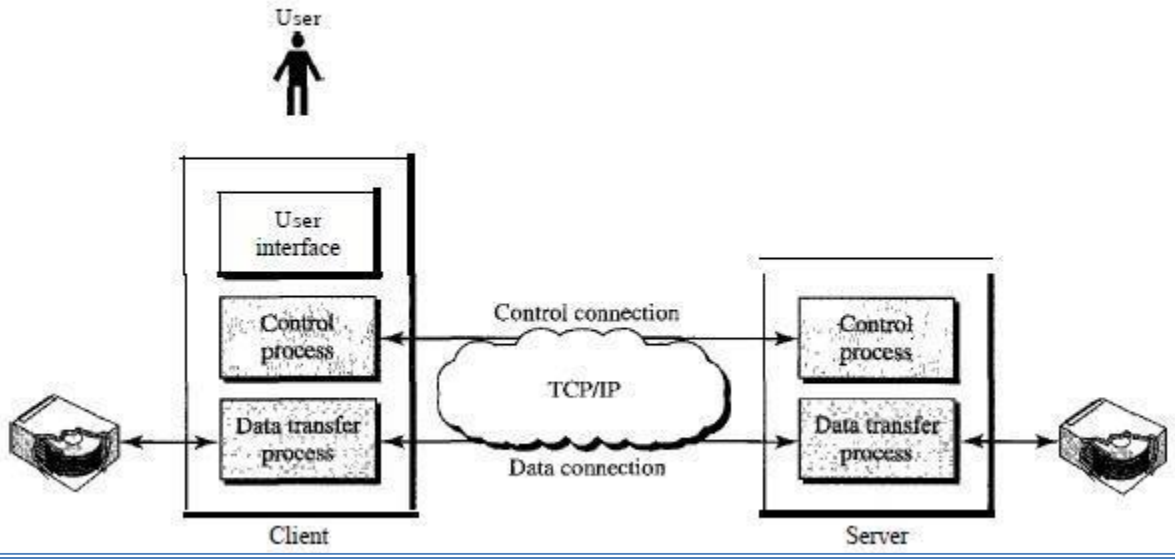
File Transfer Protocol (FTP) is the standard mechanism provided by *TCP/IP* for copying a file from one host to another. Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first. For example, two systems may use different file name conventions. Two systems may have different ways to represent text and data.

Two systems may have different directory structures. All these problems have been solved by FTP in a very simple and elegant approach.

FTP differs from other client/server applications in that it establishes two connections between the hosts. One connection is used for data transfer, the other for control information (commands and responses). Separation of commands and data transfer makes FTP more efficient. The control connection uses very simple rules of communication. We need to transfer only a line of command or a line of response at a time. The data connection, on the other hand, needs more complex rules due to the variety of data types transferred. However, the difference in complexity is at the FTP level, not TCP. For TCP, both connections are treated the same. FTP uses two well-known TCP ports: Port 21 is used for the control connection, and port 20 is used for the data connection.

Figure 26.21 shows the basic model of FTP. The client has three components: user interface, client control process, and the client data transfer process. The server has two components: the server control process and the server data transfer process. The control connection is made between the control processes. The data connection is made between the data transfer processes.

Figure 26.21 *FTP*



The control connection remains connected during the entire interactive FTP session. The data connection is opened and then closed for each file transferred. It opens each time commands that involve transferring files are used, and it closes when the file is transferred. In other words, when a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.

Transmission Mode: FTP can transfer a file across the data connection by using one of the following three transmission modes: stream mode, block mode, and compressed mode. The stream mode is the default mode. Data are delivered from FTP to TCP as a continuous stream of bytes. TCP is responsible for chopping data into segments of appropriate size. If the data are simply a stream of bytes (file structure), no end-of-file is needed. End-of-file in this case is the closing of the data connection by the sender. If the data are divided into records (record structure), each record will have a 1-byte end-of-record (EOR) character and the end of the file will have a 1-byte end-of-file (EOF) character. In block mode, data can be delivered from FTP to TCP in blocks. In this case, each block is preceded by a 3-byte header. The first byte is called the *block descriptor;* the next 2 bytes define the size of the block in bytes. In the compressed mode, if the file is big, the data can be compressed. The compression method normally used is run-length encoding. In this method, consecutive appearances of a data unit are replaced by one occurrence and the number of repetitions. In a text file, this is usually spaces (blanks). In a binary file, null characters are usually compressed.

3. HTTP

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. HTTP functions as a combination of FTP and SMTP. It is similar to FTP because it transfers files and uses the services of TCP. However, it is much simpler than FTP because it uses only one TCP connection. There is no separate control connection; only data are transferred between the client and the server. HTTP is like SMTP because the data transferred between the client and the server look like SMTP messages. In addition, the format of the messages is controlled by MIME-like headers. Unlike SMTP, the HTTP messages are not destined to be read by humans; they are read and interpreted by the HTTP server and HTTP client (browser). SMTP messages are stored and forwarded, but HTTP messages are delivered immediately. The commands from the client to the server are embedded in a request message. The contents of the requested file or other information are embedded in a response message. HTTP uses the services

of TCP on well-known port 80.

HTTP Transaction

Figure 27.12 illustrates the HTTP transaction between the client and server. Although HTTP uses the services of TCP, HTTP itself is a stateless protocol. The client initializes the transaction by sending a request message. The server replies by sending a response.

Messages

The formats of the request and response messages are similar; both are shown in Figure 27.13. A request message consists of a request line, a header, and sometimes a body. A response message consists of a status line, a header, and sometimes a body.

Figure 27.12 *HTTP transaction*

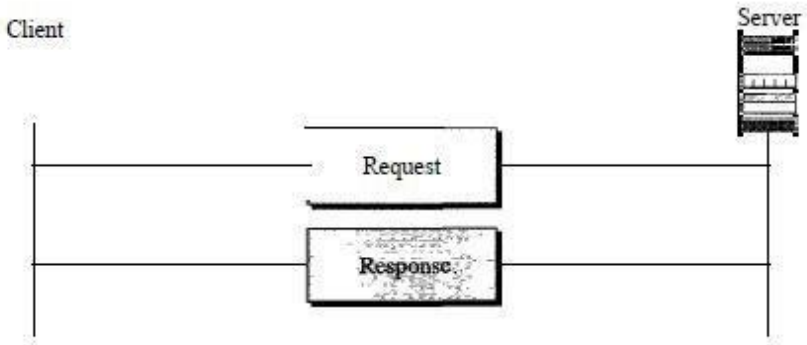
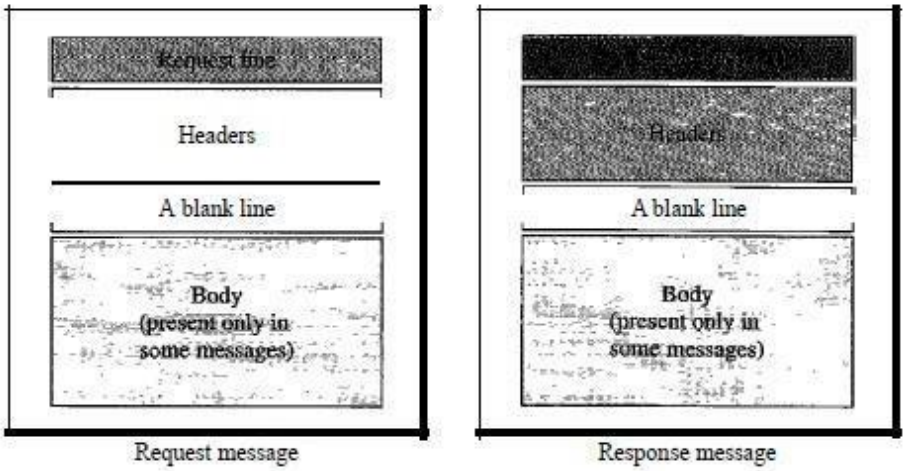
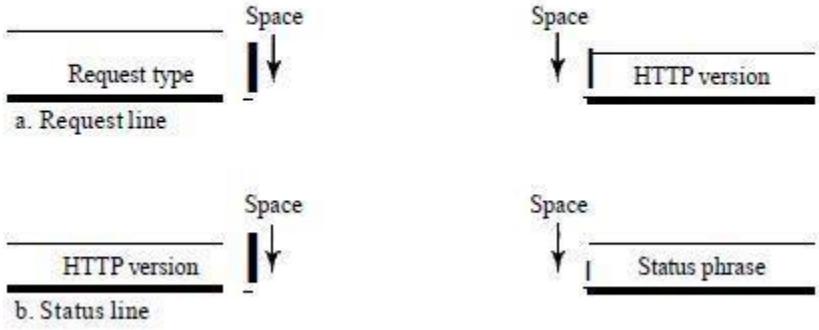


Figure 27.13 *Request and response messages*



Request and Status Lines: The first line in a request message is called a request line; the first line in the response message is called the status line. There is one common field, as shown in Figure 27.14.

Figure 27.14 *Request and status lines*



a. **Request type**. This field is used in the request message. In version 1.1 of HTTP, several request types are defined. The request type is categorized into *methods* as defined in Table 27.1.

Table 27.1 *Methods*

Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires about available options

2. **URL**. URL means Uniform Resource Locator

3. **Version**. The current version of HTTP

4. **Status code**. This field is used in the response message. The status code field is similar to those in the FTP and the SMTP protocols. It consists of three digits. Whereas the codes in the 100 range are only informational, the codes in the 200 range indicate a successful request. The codes in the 300 range redirect the client to another URL, and the codes in the 400 range indicate

an error at the client site. Finally, the codes in the 500 range indicate an error at the server site. We list the most common codes in Table 27.2.

e. **Status phrase**. This field is used in the response message. It explains the status code in text form. Table 27.2 also gives the status phrase.

Table 27.2 *Status codes*

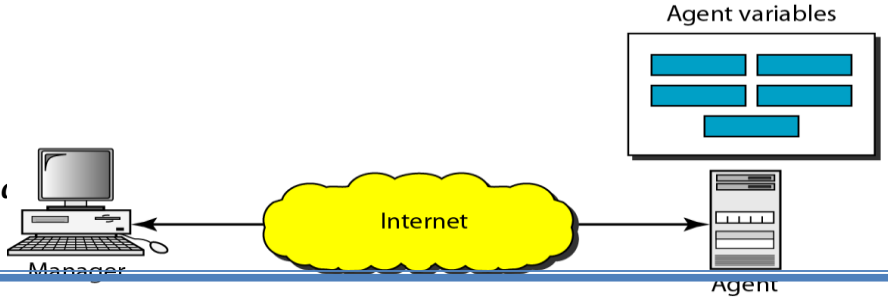
Code	*Phrase*	*Description*
	Informational	
100	Continue	The initial part of the request has been received, and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
	Success	
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon.
204	No content	There is no content in the body.

Simple Network Management Protocol (SNMP)

The Simple Network Management Protocol (SNMP) is a framework for managing devices in an internet using the TCPIIP protocol suite. It provides a set of fundamental operations for monitoring and maintaining an internet.

Concept

SNMP uses the concept of manager and agent. That is, a manager, usually a host, controls and monitors a set of agents, usually routers (see Figure below). SNMP is an application-level protocol in which a few manager stations control a set of agents. The protocol is designed at the application level so that it can monitor devices made by different manufacturers and installed on different physical networks.



Managers a

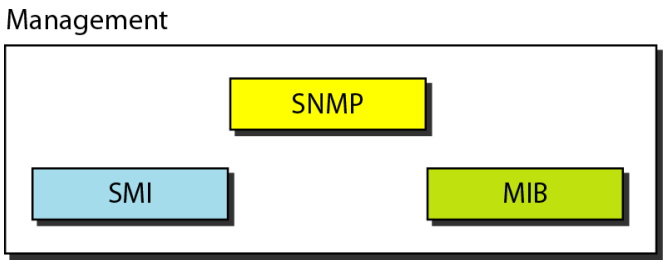
A management station, called a manager, is a host that runs the SNMP client program. A managed station, called an agent, is a router (or a host) that runs the SNMP server program. Management is achieved through simple interaction between a manager and an agent. The agent keeps performance information in a database. The manager has access to the values in the database. Agents can also contribute to the management process. The server program running on the agent can check the environment, and if it notices something unusual, it can send a warning message, called a trap, to the manager. warning message, called a trap, to the manager.

In other words, management with SNMP is based on three basic ideas:

1. A manager checks an agent by requesting information that reflects the behavior of the agent.
2. A manager forces an agent to perform a task by resetting values in the agent database.
3. An agent contributes to the management process by warning the manager of an unusual situation.

Management Components

To do management tasks, SNMP uses two other protocols: Structure of Management Information (SMI) and Management Information Base (MIB).



RoleofSNMP

SNMP has some very specific roles in network management. SNMP defines the format of packets exchanged between a manager and an agent. It reads and changes the status (values) of objects (variables) in SNMP packets.

Role ofSMI

SMI defines the general rules for naming objects, defining object types (including range and length), and showing how to encode objects and values. SMI does not define the number of objects an entity

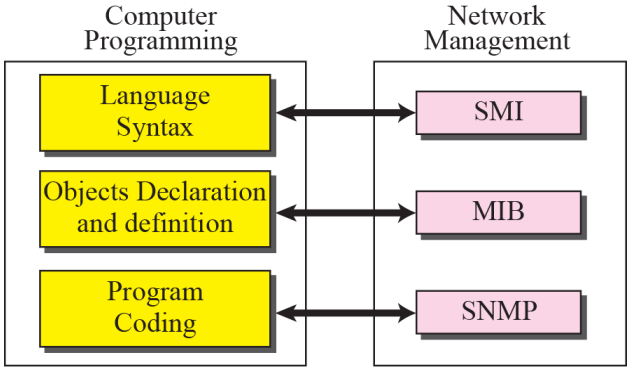
should manage or name the objects to be managed or define the association between the objects and their values.

Role ofMIB

We need some protocol to define the number of objects, name them according to the rules defined by SMI, and associate a type to each named object and thus the role of MIB creates a collection of named objects, their types, and their relationships to each other in an entity to be managed.

ANALOGY

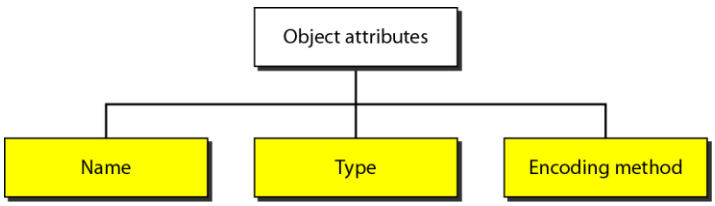
The functions of SNMP is similar to the functions of any of the programming. The role of SMI is similar to the rules followed by the programming language. The job of MIB lies in defining and declaring the various objects.



Structure of Management Information

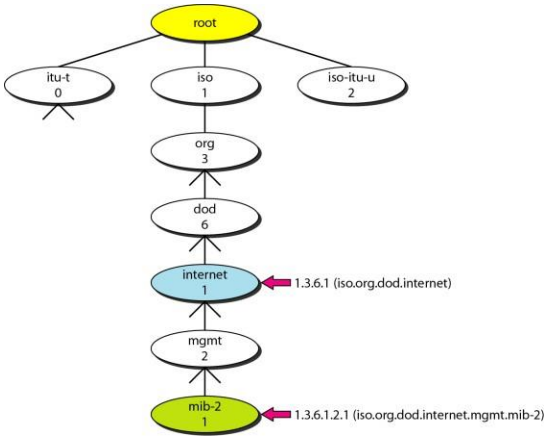
The Structure of Management Information, version 2 (SMIv2) is a component for network management. Its functions are

1. To name objects
2. To define the type of data that can be stored in an object
3. To show how to encode data for transmission over the network



1. Naming of Objects:

SMI requires that each managed object (such as a router, a variable in a router, a value) have a unique name. To name objects globally, SMI uses an object identifier, which is a hierarchical identifier based on a tree structure (see Figure below).

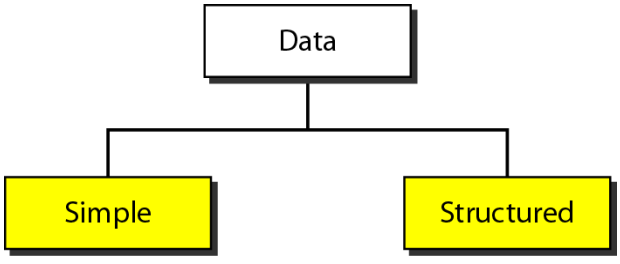


The tree structure starts with an unnamed root. Each object can be defined by using a sequence of integers separated by dots. The tree structure can also define an object by using a sequence of textual names separated by dots. The integer-dot representation is used in SNMP. The name-dot notation is used by people. For example, the following shows the same object in two different notations:

iso.org.dod.internet.mgmt.mib-2 ... 1.3.6.1.2.1

2. Type of the data:

The second attribute of an object is the type of data stored in it. SMI has two broad categories of data type: *simple* and *structured*



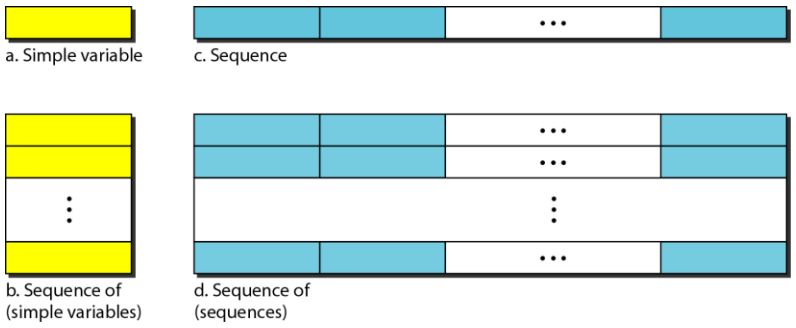
Simple Type The simple data types are atomic data types. Some of them are taken directly from ASN.l; others are added by SMI. Table 28.1. The first five are from ASN.l; the next seven are defined by SMI.

Type	*Size*	*Description*
INTEGER	4 bytes	An integer with a value between -2^{31} and $2^{31} - 1$
Integer32	4 bytes	Same as INTEGER
Unsigned32	4 bytes	Unsigned with a value between 0 and $2^{32} - 1$
OCTET STRING	Variable	Byte string up to 65,535 bytes long
OBJECT IDENTIFIER	Variable	An object identifier
IPAddress	4 bytes	An IP address made of four integers
Counter32	4 bytes	An integer whose value can be incremented from 0 to 2^{32}; when it reaches its maximum value, it wraps back to 0.
Counter64	8 bytes	64-bit counter
Gauge32	4 bytes	Same as Counter32, but when it reaches its maximum value, it does not wrap; it remains there until it is reset
TimeTicks	4 bytes	A counting value that records time in $\frac{1}{100}$ s
BITS		A string of bits
Opaque	Variable	Uninterpreted string

Structured Type By combining simple and structured data types, we can make new structured data types. SMI defines two structured data types: *sequence* and *sequence of*

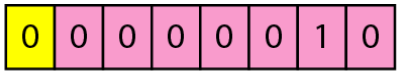
Sequence. A *sequence* data type is a combination of simple data types, not necessarily of the same type. It is analogous to the concept of a *struct* or a *record* used in programming languages such as C.

Sequence of. A *sequence of*data type is a combination of simple data types all of the same type or a combination of sequence data types all of the same type. It is analogous to the concept of an *array* used in programming languages such as C.

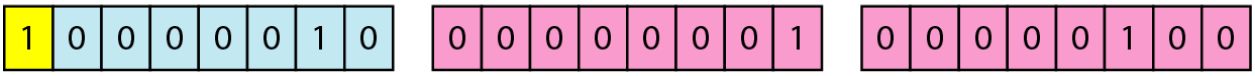


3. Encoding Method

SMI uses another standard, Basic Encoding Rules (BER), to encode data to be transmitted over the network. BER specifies that each piece of data be encoded in triplet format: tag, length, and value, as illustrated in Figure below.



a. The colored part defines the length (2).



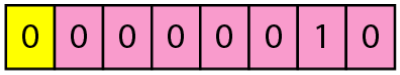
b. The shaded part defines the length of the length (2 bytes); the colored bytes define the length (260 bytes).

Tag. The tag is a I-byte field that defines the type of data. It is composed of three subfields: *class* (2 *bits),fonnat* (1 bit), and *number* (5 bits). The class subfield defines the scope of the data. Four classes are defined: universal (00), applicationwide (01), context-specific (10), and private (11). The universal data types are those taken from ASN.l (INTEGER, OCTET STRING, and ObjectIdentifier). The applicationwide data types are those added by SMI (IPAddress, Counter, Gauge, and TimeTicks). The five context-specific data types have meanings that may change from one protocol to another. The private data types are vendor-specific.

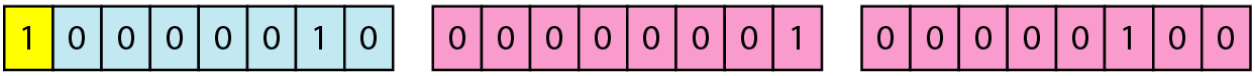
The **format** subfield indicates whether the data are simple (0) or structured (1). The number subfield further divides simple or structured data into subgroups. For example, in the universal class, with simple format, INTEGER has a value of 2, OCTET STRING has a value of 4, and so on. Table 28.2 shows the data types we use in this chapter and their tags in binary and hexadecimal numbers.

Data Type	*Class*	*Format*	*Number*	*Tag (Binary)*	*Tag (Hex)*
INTEGER	00	0	00010	**00000010**	**02**
OCTET STRING	00	0	00100	**00000100**	**04**
OBJECT IDENTIFIER	00	0	00110	**00000110**	**06**
NULL	00	0	00101	**00000101**	**05**
Sequence, sequence of	00	1	10000	**00110000**	**30**
IPAddress	01	0	00000	**01000000**	**40**
Counter	01	0	00001	**01000001**	**41**
Gauge	01	0	00010	**01000010**	**42**
TimeTicks	01	0	00011	**01000011**	**43**
Opaque	01	0	00100	**01000100**	**44**

Length. The length field is I or more bytes. If it is 1 byte, the most significant bit must be O. The other 7 bits define the length of the data. If it is more than 1 byte, the most significant bit of the first byte must be 1. The other 7 bits of the first byte define the number of bytes needed to define the length. See Figure 28.10 for adepiction of the length field.



a. The colored part defines the length (2).



b. The shaded part defines the length of the length (2 bytes);
the colored bytes define the length (260 bytes).

Management Information Base (MIB)

The Management Information Base, version 2 (MIB2) is the second component used in network management. Each agent has its own MIB2, which is a collection of all the objects that the manager can manage. The objects in MIB2 are categorized under 10 different groups as shown in figure.

sys This object *(system)* defines general information about the node (system), such as the name, location, and lifetime.

if This object *(inteiface)* defines information about all the interfaces of the node including interface number, physical address, and IP address.

at This object *(address translation)* defines the information about the ARP table.

ip This object defines information related to IP, such as the routing table and the IP address.

icmp This object defines information related to ICMP, such as the number of packets sent and received and total errors created.

tcp This object defines general information related to TCP, such as the connection table, time-out value, number of ports, and number of packets sent and received.

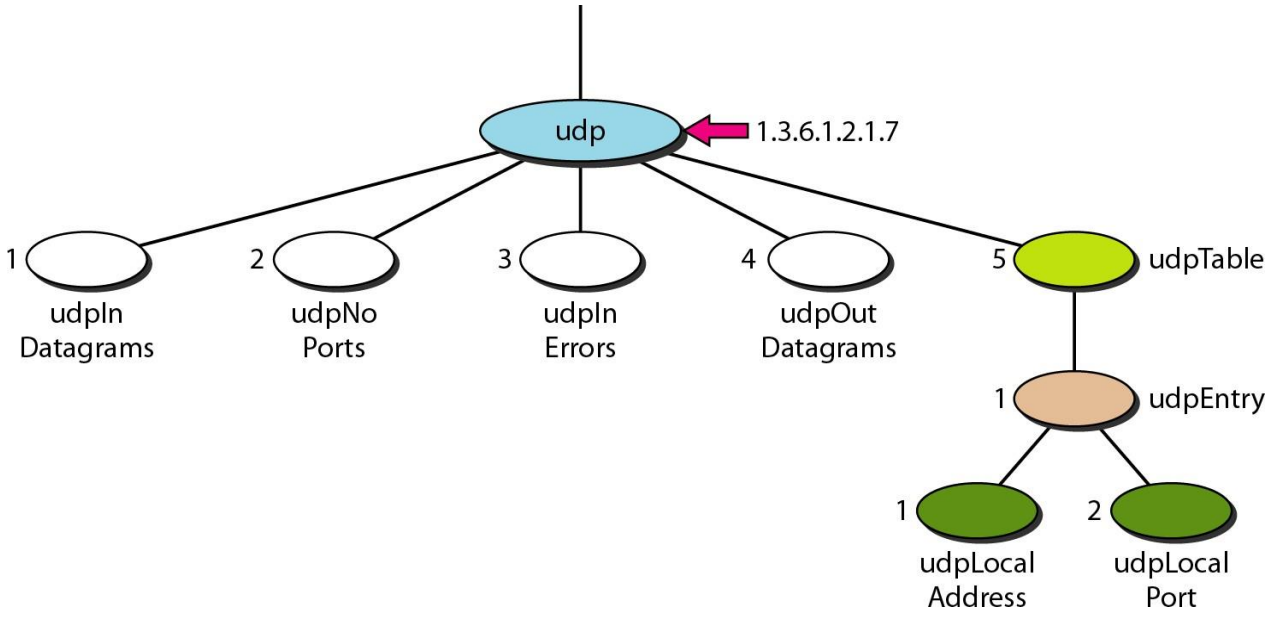
udp This object defines general information related to UDP, such as the number of ports and number of packets sent and received.

snmp This object defines general information related to SNMP itself.

Accessing MIB Variables

To show how to access different variables, we use the udp group as an example. There are four simple variables in the udp group and one sequence of (table of) records. Figure 28.16 shows the variables and the table. We will show how to access each entity.

Simple Variables To access any of the simple variables, we use the id of the group (1.3.6.1.2.1.7) followed by the id of the variable. The following shows how to access each variable.



udpInDatagrams.0	1.3.6.1.2.1.7.1.0
udpNoPorts.0	1.3.6.1.2.1.7.2.0
udpInErrors.0	1.3.6.1.2.1.7.3.0
udpOutDatagrams.0	1.3.6.1.2.1.7.4.0

udpTable ... 1.3.6.1.2.1.7.5

udpEntry ... 1.3.6.1.2.1.7.5.1

udpLocalAddress 1.3.6.1.2.1.7.5.1.1
udpLocalPoo' 1.3.6.1.2.1.7.5.1.2

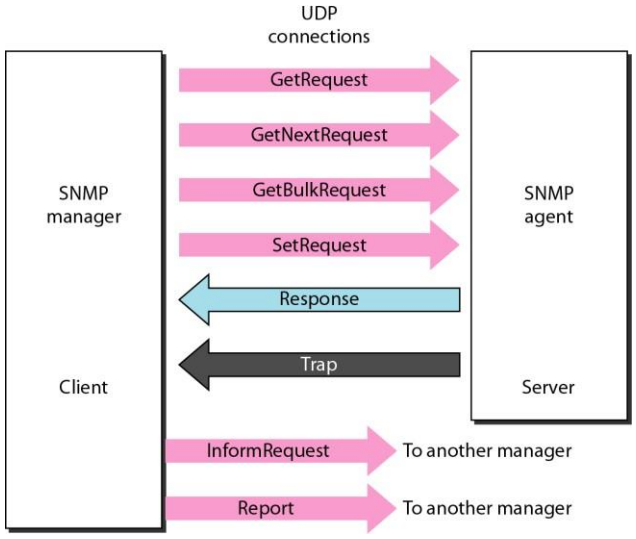
SNMP

SNMP uses both SMI and MIB in Internet network management. It is an application program that allows

1. A manager to retrieve the value of an object defined in an agent
2. A manager to store a value in an object defined in an agent
3. An agent to send an alarm message about an abnormal situation to the manager

PDUs

SNMPv3 defines eight types of packets (or PDUs): GetRequest, GetNextRequest, GetBulkRequest,
SetRequest, Response, Trap, InformRequest, and Report (see Figure below).



GetRequest The GetRequest PDU is sent from the manager (client) to the agent (server) to retrieve the value of a variable or a set of variables.

GetNextRequest The GetNextRequest PDU is sent from the manager to the agent to retrieve the value of a variable. The retrieved value is the value of the object following the defined Objectid in the PDD. It is mostly used to retrieve the values ofthe entries in a table. If the manager does

not know the indexes of the entries, it cannot retrieve the values. However, it can use GetNextRequest and define the ObjectId of the table. Because the first entry has the ObjectId immediately after the ObjectId of the table, the value of the first entry is returned. The manager can use this ObjectId to get the value of the next one, and so on.

GetBulkRequest The GetBulkRequest POD is sent from the manager to the agent to retrieve a large amount of data. It can be used instead of multiple GetRequest and GetNextRequest PODs.

SetRequest The SetRequest PDD is sent from the manager to the agent to set (store) a value in a variable.

Response The Response PDD is sent from an agent to a manager in response to GetRequest or GetNextRequest. It contains the value(s) of the variable(s) requested by the manager.

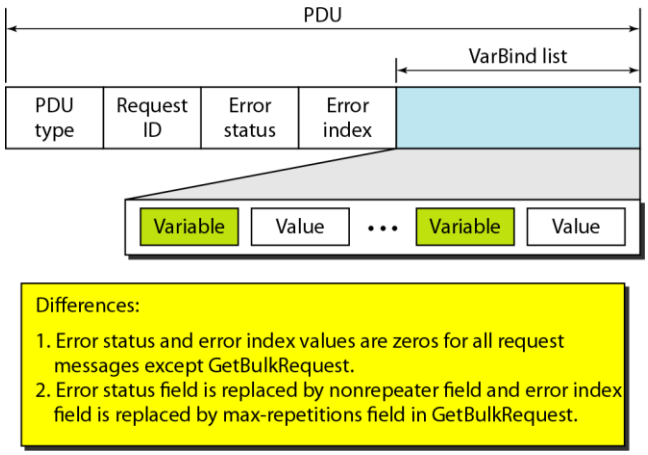
Trap The Trap (also called SNMPv2 Trap to distinguish it from SNMPv1 Trap)

POD is sent from the agent to the manager to report an event. For example, if the agent is rebooted, it infonns the manager and reports the time of rebooting. InformRequest The InfonnRequest POD is sent from one manager to another remote manager to get the value of some variables from agents under the control of the remote manager. The remote manager responds with a Response POD.

Report The Report POD is designed to report some types of errors between managers. It is not yet in use.

Format

The fonnat for the eight SNMP PODs is shown in Figure 28.21. The GetBulkRequest POD differs from the others in two areas, as shown in the figure.



PDU type. This field defines the type of the POD (see Table 28.4).

Request ID. This field is a sequence number used by the manager in a Request POD and repeated by the agent in a response. It is used to match a request to a response. Error status. This is an integer that is used only in Response PDUs to show the types of errors reported by the agent. Its value is 0 in Request PDUs. Table 28.3 lists the types of errors that can occur.

Status	*Name*	*Meaning*
0	noError	No error
1	tooBig	Response too big to fit in one message
2	noSuchName	Variable does not exist
3	badValue	The value to be stored is invalid
4	readOnly	The value cannot be modified
5	genErr	Other errors

Nonrepeaters. This field IS used only in GetBulkRequest and replaces the error status field, which is empty in Request PDUs.

Error index. The error index is an offset that tells the manager which variable caused the errOr.

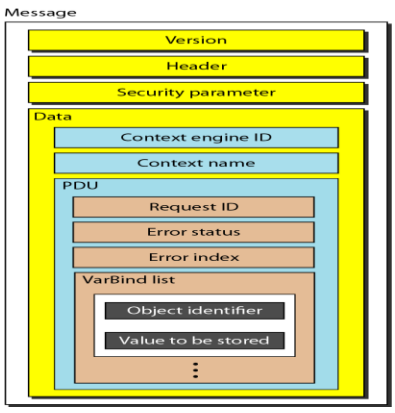
Max-repetition. This field is also used only in GetBulkRequest and replaces the error index field, which is empty in Request PDUs.

VarBind list. This is a set of variables with the corresponding values the manager wants to retrieve or set. The values are null in GetRequest and GetNextRequest. In a Trap PDU, it shows the variables and values related to a specific PDU.

Messages

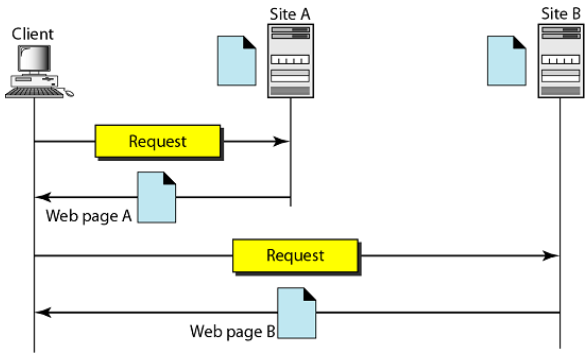
SNMP does not send only a PDU, it embeds the PDU in a message. A message in SNMPv3 is made of four elements: version, header, security parameters, and data (which include the encoded PDU), as shown in Figure 28.22.

Because the length of these elements is different from message to message, SNMP uses BER to encode each element. Remember that BER uses the tag and the length to define a value. The *version* defines the current version (3). The *header* contains values for message identification, maximum message size (the maximum size of the reply), message flag (one octet of data type OCTET STRING where each bit defines security type, such as privacy or authentication, Or other information), and a message security model (defining the security protocol). The message *security parameter* is used to create a message digest (see Chapter 31). The data contain the PDU. If the data are encrypted, there is information about the encrypting engine (the manager program that did the encryption) and the encrypting context (the type of encryption) followed by the encrypted PDU. If the data are not encrypted, the data consist of just the PDU. To define the type ofPDU, SNMP uses a tag. The class is context-sensitive (10), the format is structured (1), and the numbers are 0, 1,2, 3, 5, 6, 7, and 8 (see Table 28.4). Note that SNMPvl defined A4 for Trap, which is obsolete today.



WWW

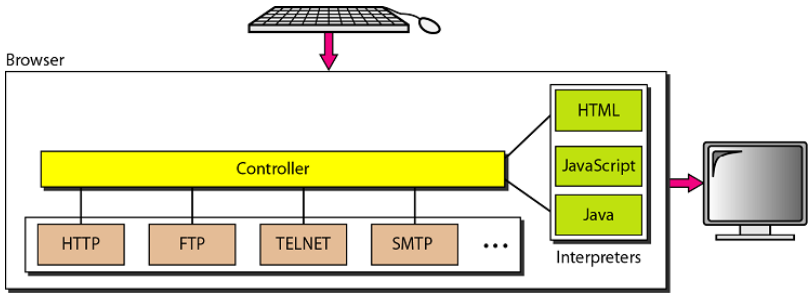
TheWWW today is a distributed clientJserver service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called *sites*, as shown **in** Figure below



Each site holds one or more documents, referred to as *Web pages.* Each Web page can contain a link to other pages in the same site or at other sites. The pages can be retrieved and viewed by using browsers. Let us go through the scenario shown in above Figure. The client needs to see some information that it knows belongs to site A. It sends a request through its browser, a program that is designed to fetch Web documents. The request, among other information, includes the address of the site and the Web page, called the URL, which we will discuss shortly. The server at site A finds the document and sends it to the client. When the user views the document, she finds some references to other documents, including a Web page at site B. The reference has the URL for the new site. The user is also interested in seeing this document. The client sends another request to the new site, and the new page is retrieved.

Client (Browser)

A variety of vendors offer commercial browsers that interpret and display a Web document, and all use nearly the same architecture. Each browser usually consists of three parts: a controller, client protocol, and interpreters. The controller receives input from the keyboard or the mouse and uses the client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on the screen.

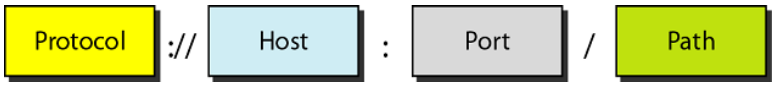


Server

The Web page is stored at the server. Each time a client request arrives, the corresponding document is sent to the client. To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than disk.

Uniform Resource Locator

A client that wants to access a Web page needs the address. To facilitate the access of documents distributed throughout the world, HTTP uses locators. The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet. The URL defines four things: protocol, host computer, port, and path (see Figure below).



The ***protocol*** is the client/server program used to retrieve the document. Many different protocols can retrieve a document; among them are FTP or HTTP. The most common today is HTTP.

The **host** is the computer on which the information is located, although the name of the computer can be an alias. Web pages are usually stored in computers, and computers are given alias names that usually begin with the characters "www". This is not mandatory, however, as the host can be any name given to the computer that hosts the Web page. The URL can optionally contain the port number of the server.

If the ***port*** is included, it is inserted between the host and the path, and it is separated from the host by a colon.

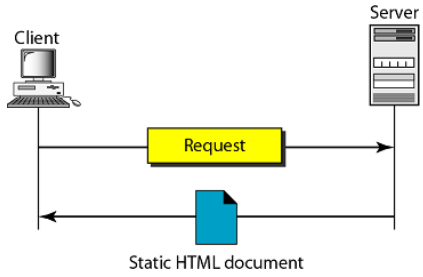
Path is the pathname of the file where the information is located. Note that the path can itself contain slashes that, in the UNIX operating system, separate the directories from the subdirectories and files.

WEB DOCUMENTS

The documents in the WWW can be grouped into three broad categories: static, dynamic, and active.

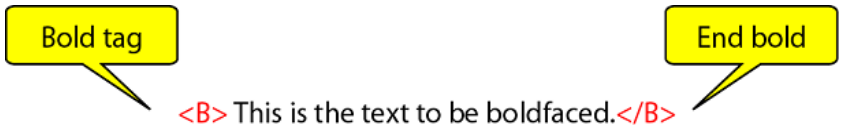
Static Documents

Static documents are fixed-content documents that are created and stored in a server. The client can get only a copy of the document. In other words, the contents of the file are determined when the file is created, not when it is used. Of course, the contents in the server can be changed, but the user cannot change them. When a client accesses the document, a copy of the document is sent. The user can then use a browsing program to display the document (see Figure below).

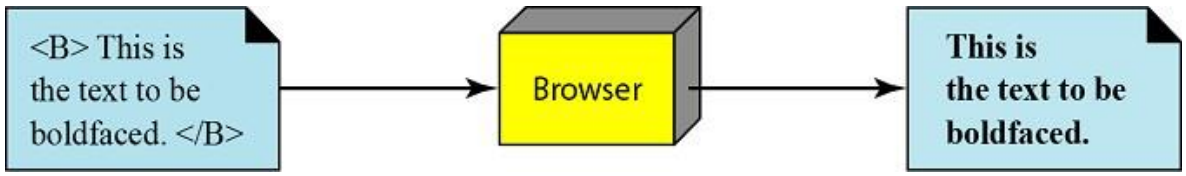


HTML

For creating web documents we use HTML. Hypertext Markup Language (HTML) is a language for creating Web pages. The term *markup language* comes from the book publishing industry. To make part of a text displayed in boldface with HTML, we put beginning and ending boldface tags (marks) in the text, as shown in Figure 27.5. The two tags <B> and </B> are instructions for the browser.



When the browser sees these two marks, it knows that the text must be boldfaced (see Figure above). A markup language such as HTML allows us to embed formatting instructions in the file itself. The instructions are included with the text. In this way, any browser can read the instructions and format the text according to the specific workstation



HTML lets us use only ASCII characters for both the main text and formatting instructions. In this way, every computer can receive the whole document as an ASCII document. The main text is the data, and the formatting instructions can be used by the browser to format the data. A Web page is made up of two parts: the head and the body. The head is the first part of a Web page. The head contains the title of the page and other parameters that the browser will use. The actual contents of a page are in the body, which includes the text and the tags. Whereas the text is the actual infonnation contained in a page, the tags define the appearance of the document. Every HTML tag is a name followed by an optional list of attributes, all enclosed between less-than and greater-than symbols « and >).

An **attribute**, if present, is followed by an equals sign and the value of the attribute. Some tags can be used alone; others must be used in pairs. Those that are used in pairs are called *beginning* and *ending* tags. The beginning tag can have attributes and values and starts with the name of the tag. The ending tag cannot have attributes or values but must have a slash before the name of the tag.

a. Beginning tag

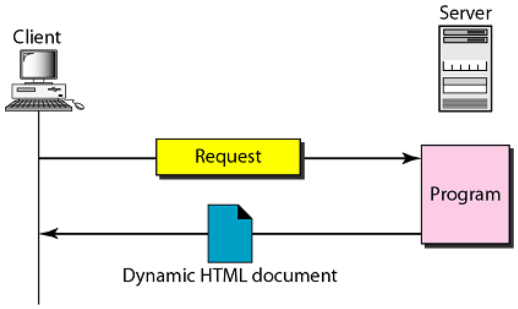
b. Ending tag

Dynamic Documents

A **dynamic document** is created by a Web server whenever a browser requests the document. When a request arrives, the Web server runs an application program or a script that creates the dynamic document. The server returns the output of the program or script as a response to the browser that requested the document. Because a fresh document is created for each request, the contents of a dynamic document can vary from one request to another. very simple example of a dynamic document is the retrieval of the time and date from a server.

Common Gateway Interface (CGI)

The **Common** Gateway **Interface** (CGI) is a technology that creates and handles dynamic documents. CGI is a set of standards that defines how a dynamic document is written, how data are input to the program, and how the output result is used. CGI is not a new language; instead, it allows programmers to use any of several languages such as C, C++, Boume Shell, Kom Shell, C Shell, Tcl, or Perl. The only thing that CGI defines is a set of rules and tenns that the programmer must follow.



Input In traditional programming, when a program is executed, parameters can be passed to the program. Parameter passing allows the programmer to write a generic program that can be used in different situations

*For **example,** a generic copy program can be written to copy any file to another. A user can use the program to copy a file named x to another file named y by passing x and y as parameters. The input from a browser to a server is sent by using aform. If the infonnation in a fonn is small (such as a word), it can be appended to the URL after a question mark. For example, the following URL is carrying fonn infonnation (23, a value):*

http://www.deanzalcgi-binlprog.pl?23

When the server receives the URL, it uses the part of the URL before the question mark to access the program to be run, and it interprets the part after the question mark (23) as the input sent by

the client. It stores this string in a variable. When the CGI program is executed, it can access this value.

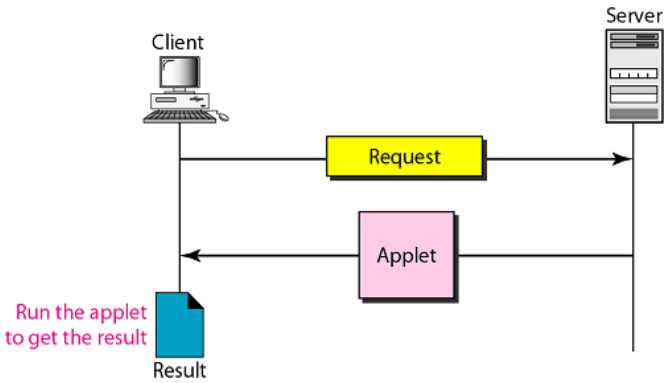
Output The whole idea of CGI is to execute a CGI program at the server site and send the output to the client (browser). The output is usually plain text or a text with HTML structures; however, the output can be a variety of other things. It can be graphics or binary data, a status code, instructions to the browser to cache the result, or instructions to the server to send an existing document instead of the actual output.

Active Documents

For many applications, we need a program or a script to be run at the client site. These are called active documents. For example, suppose we want to run a program that creates animated graphics on the screen or a program that interacts with the user. The program definitely needs to be run at the client site where the animation or interaction takes place. When a browser requests an active document, the server sends a copy of the document or a script. The document is then run at the client (browser) site.

Java Applets

One way to create an active document is to use Java applets. Java is a combination of a high-level programming language, a run-time environment, and a class library that allows a programmer to write an active document (an applet) and a browser to run it. It can also be a stand-alone program that doesn't use a browser.



JavaScript

The idea of scripts in dynamic documents can also be used for active documents. If the active part of the document is small, it can be written in a scripting language; then it can be interpreted and run by the client at the same time and the same is illustrated in following fig.