DATA COMMUNICATIONS
&
COMPUTER NETWORK

LECTURE NOTES(Part 2)

By
R.P.Nayak

Department of Computer Science and
Engineering, Kalahandi,
Bhawanipatna
766002

Course of Studies

Module – I (12 Hrs)

Overview of Data Communications and Networking.

Physical Layer : Analog and Digital, Analog Signals, Digital Signals, Analog versus Digital, Data Rate Limits, Transmission Impairment, More about signals.

Digital Transmission: Line coding, Block coding, Sampling, Transmission mode.

Analog Transmission: Modulation of Digital Data; Telephone modems, modulation of Analog signals.

Multiplexing : FDM , WDM , TDM.

Transmission Media: Guided Media, Unguided Media (wireless).

Circuit switching and Telephone Network: Circuit switching, Telephone network.

Module –II (12 Hrs)

Data Link Layer

Error Detection and Correction: Types of Errors, Detection, Error Correction

Data Link Control and Protocols:

Flow and Error Control, Stop-and-wait ARQ. Go-Back-N ARQ, Selective Repeat ARQ, HDLC.

Point-to –Point Access: PPP ,Point –to- Point Protocol, PPP Stack,

Multiple Access: Random Access, Controlled Access, Channelization.

Local area Network: Ethernet, Traditional Ethernet, Fast Ethernet, Gigabit Ethernet. Token bus, Token ring.

Wireless LANs: IEEE 802.11, Bluetooth

Virtual Circuits: Frame Relay and ATM.

Module – III (12 Hrs)

Network Layer:

Host to Host Delivery: Internetworking, addressing and Routing.

Network Layer Protocols: ARP, IPV4, ICMP, IPV6 and ICMPV6.

Transport Layer: Process to Process Delivery: UDP, TCP, Congestion Control and Quality of Service.

Application Layer :

Client Server Model, Socket Interface, Domain Name System (DNS), Electronic Mail (SMTP) and File Transfer (FTP), HTTP and WWW.

Text Books:

1. Data Communications and Networking: Behrouz A. Forouzan, Tata McGraw-Hill, 4th Ed

2. Computer Networks: A. S. Tannenbum, D. Wetherall, Prentice Hall, Imprint of Pearson 5th Ed

Reference Book : .

1. Computer Networks:A system Approach:Larry L, Peterson and Bruce S. Davie,Elsevier, 4th Ed

2. Computer Networks: Natalia Olifer, Victor Olifer, Willey India

3. Data and Computer Communications: William Stallings, Prentice Hall, Imprint of Pearson, 9th Ed.

4. Data communication & Computer Networks: Gupta, Prentice Hall of India

5. Network for Computer Scientists & Engineers: Zheng, Oxford University Press

6. Data Communications and Networking: White, Cengage Learning

Module-ii & Module -iii

Important topics discussed:

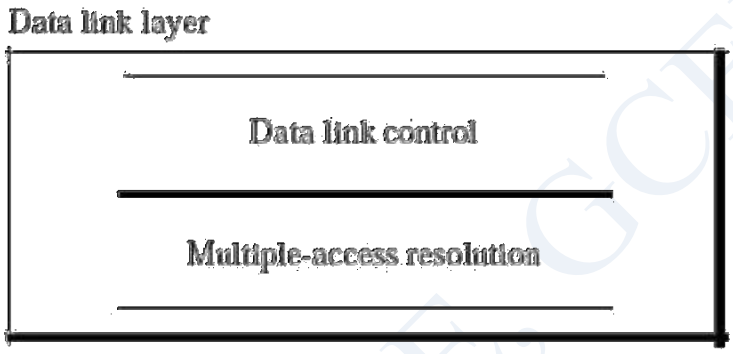
1. ALOHA, CSMA, CSMA/CD, CSMA/CA
2. FDMA,TDMA,CDMA
3. IEEE 802.11
4. BLUETOOTH Architecture
5. IP Addressing, IPv4 Datagram
6. TCP, UDP, SCTP
7. Congestion Control and Quality of Service
8. DNS, ELECTRONIC MAIL, FTP, WWW, HTTP, SNMP

This page is left vacant intentionally

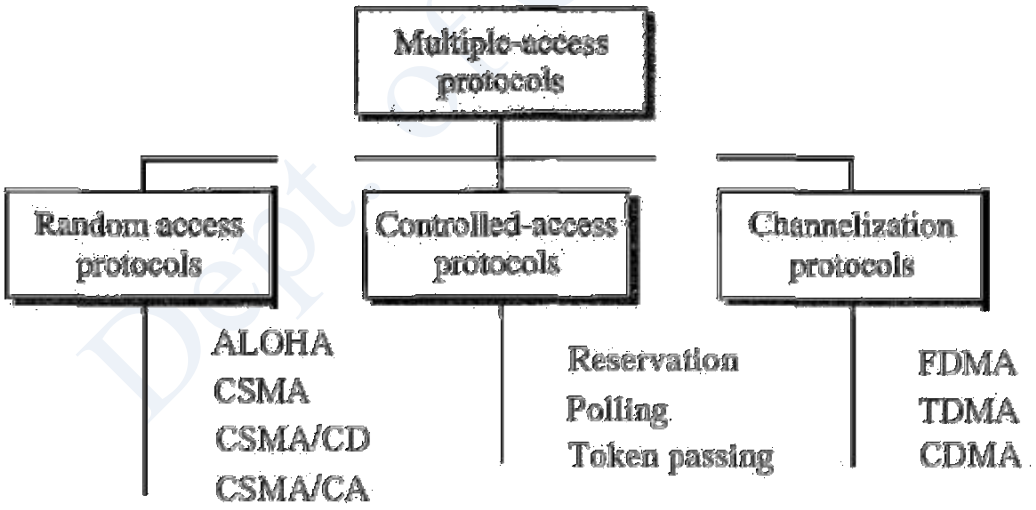
MULTIPLE ACCESS

If we use our cellular phone to connect to another cellular phone, the channel (the band allocated to the vendor company) is not dedicated. A person a few feet away from us may be using the same channel to talk to her friend.

We can consider the data link layer as two sublayers. The upper sublayer is responsible for data link control, and the lower sublayer is responsible for resolving access to the shared media. If the channel is dedicated, we do not need the lower sublayer.



Many formal protocols have been devised to handle access to a shared link. We categorize them into three groups. Protocols belonging to each group are shown in below figure:



RANDOM ACCESS

In random access or contention methods, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send.

At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision depends on the state of the medium (idle or busy). In other words, each station can transmit when it desires on the condition that it follows the predefined procedure, including the testing of the state of the medium.

Two features give this method its name. First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called random access. Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called contention methods.

In a random access method, each station has the right to the medium without being controlled by any other station. However, if more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified. To avoid access conflict or to resolve it when it happens, each station follows a procedure that answers the following questions:

When can the station access the medium?

What can the station do if the medium is busy?

How can the station determine the success or failure of the transmission? What can the station do if there is an access conflict?

ALOHA

ALOHA, the earliest random access method was developed at the University of Hawaii in early 1970. It was designed for a radio (wireless) LAN, but it can be used on any shared medium.

Pure ALOHA

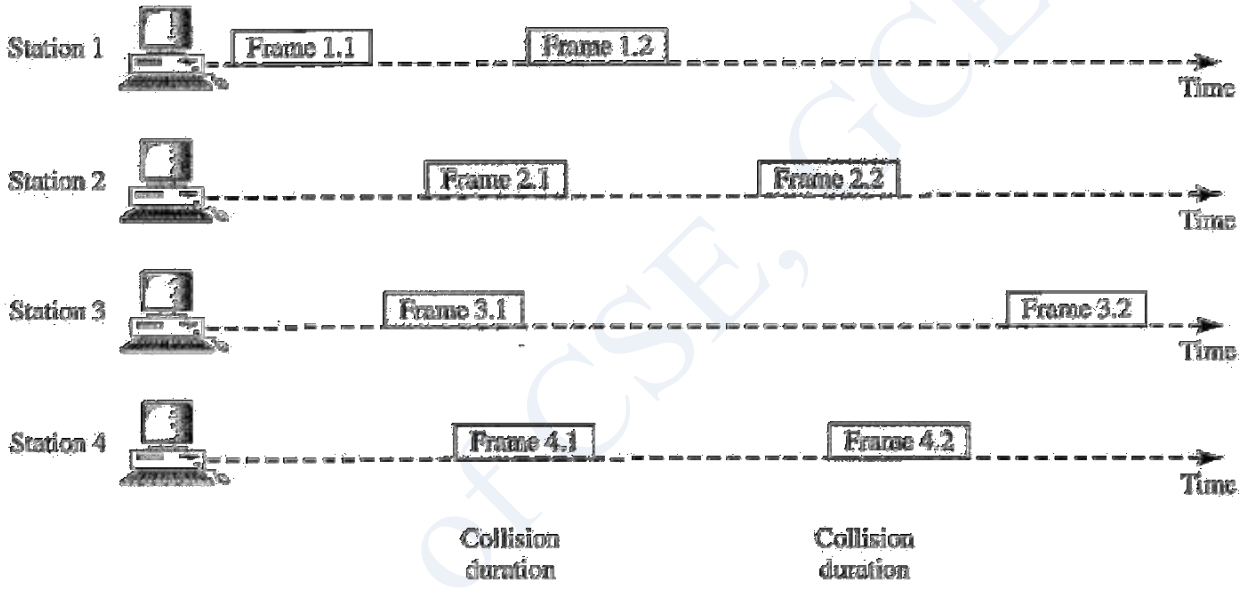
The original ALOHA protocol is called *pure ALOHA*. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send.

However, since there is only one channel to share, there is the possibility of collision between frames from different stations.

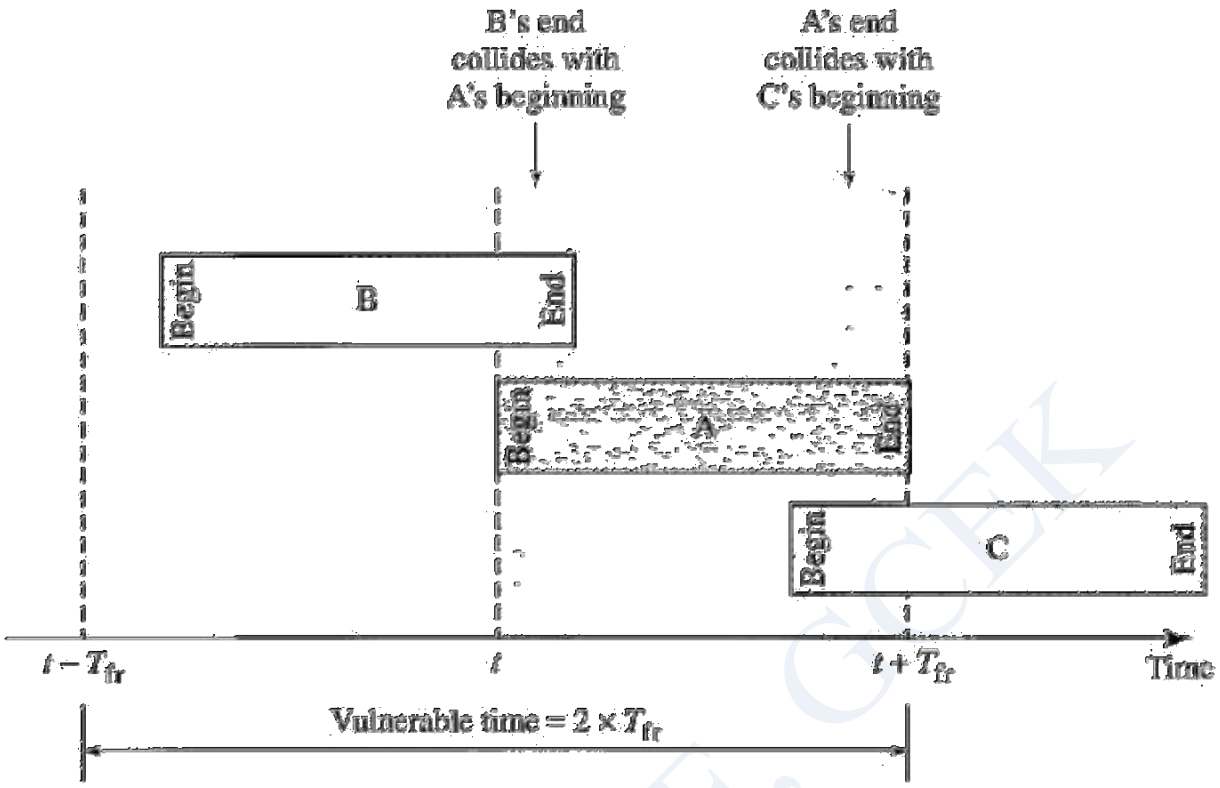
It is obvious that we need to resend the frames that have been destroyed during transmission. The pure ALOHA protocol relies on acknowledgments from the receiver.

When a station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.

There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. The figure shows that each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel. Below figure shows that only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3. We need to mention that even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed.



Vulnerable time: Let us find the length of time, the **vulnerable time,** in which there is a possibility of collision. We assume that the stations send fixed-length frames with each frame taking T_{fr} s to send.



Pure ALOHA vulnerable time = 2 x T_{fr}

Example: A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution:

Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms. The vulnerable time is 2 x

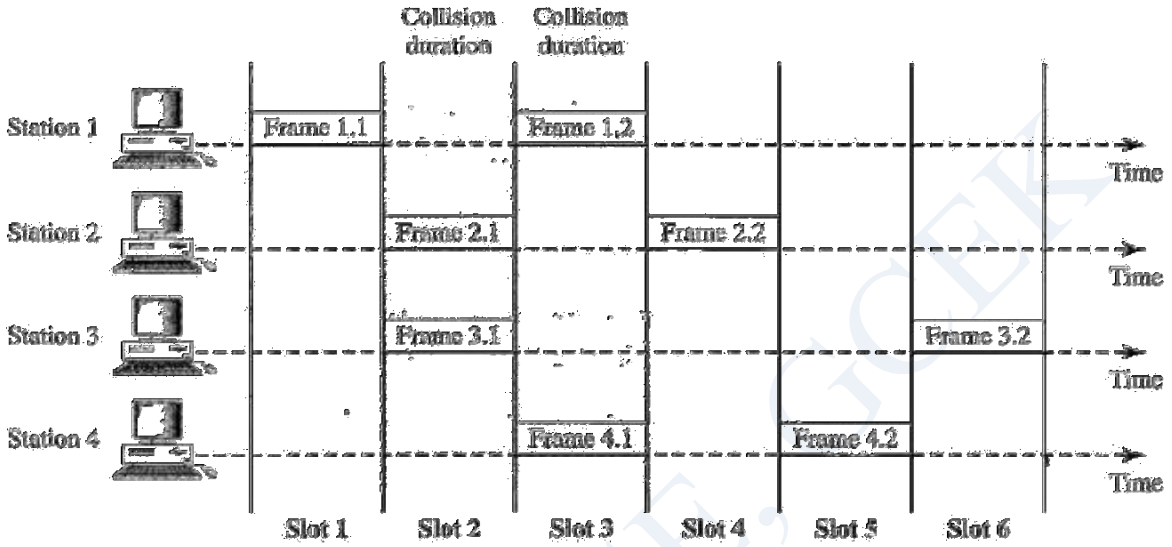
1 ms =2ms. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the one 1ms period that this station is sending.

Throughput: Let us call G the average number of frames generated by the system during one frame transmission time. Then it can be proved that the average number of successful transmissions for pure ALOHA is S = G x e^{-2G}. The maximum throughput Smax is 0.184, for G = 1/2. In other words, if one-half a frame is generated during one frame transmission time (in other words, one frame during two frame transmission times), then 18.4 percent of these frames reach their destination successfully.

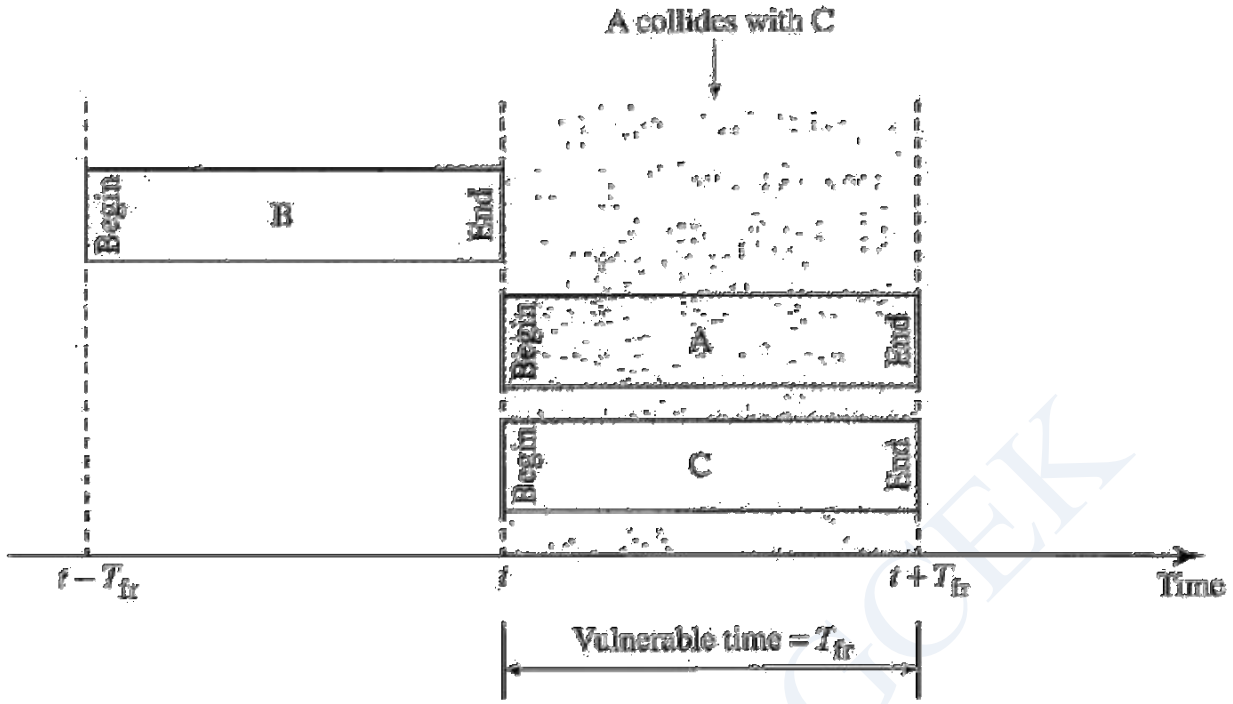
The throughput for pure ALOHA is S =G x e^{-2G}.
The maximum throughput Smax =0.184 when G =(1/2).

Slotted Aloha

Slotted ALOHA was invented to improve the efficiency of pure ALOHA. In slotted ALOHA we divide the time into slots of T_{fr} s and force the station to send only at the beginning of the time slot.



Slotted ALOHA vulnerable time = T_{fr}



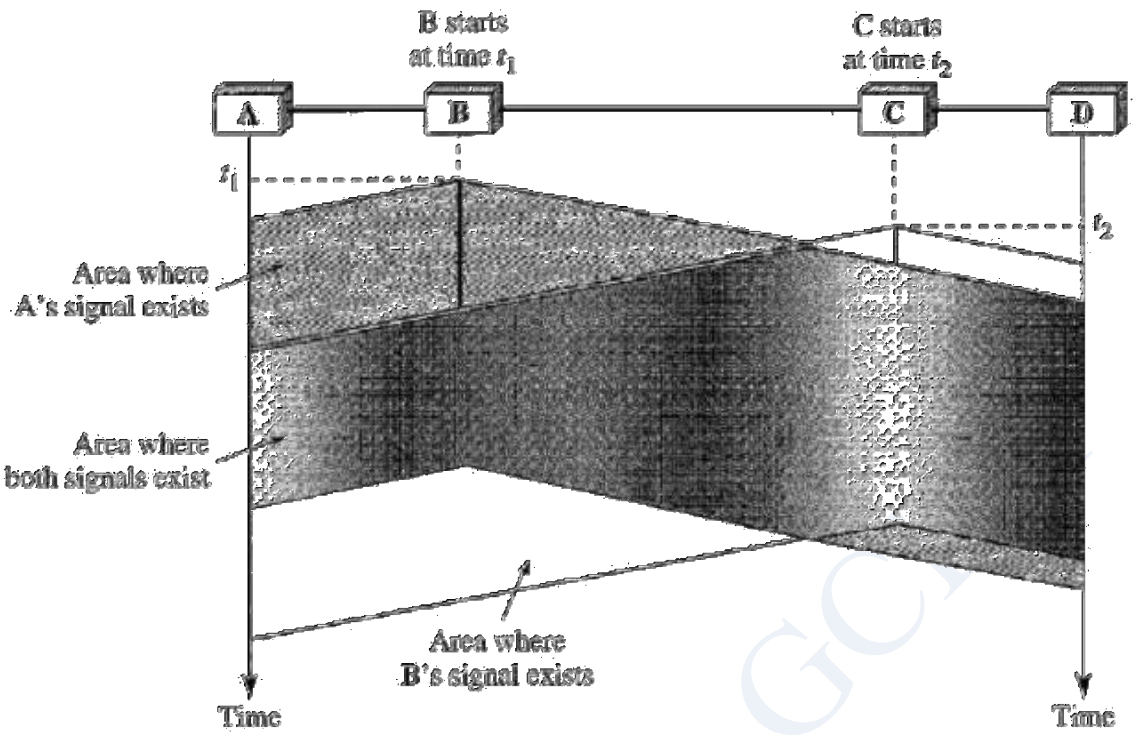
The throughput for slotted ALOHA is S =G x e^{-G}.
The maximum throughput Smax =0.368when G =1.

Carrier Sense Multiple Access (CSMA)

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed.

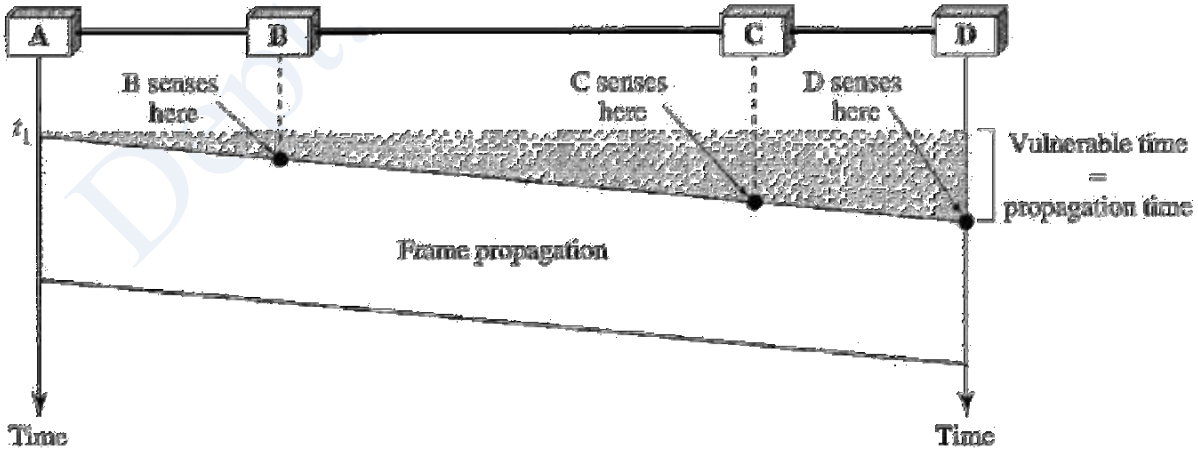
The chance of collision can be reduced if a station senses the medium before trying to use it.

Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk." CSMA can reduce the possibility of collision, but it cannot eliminate it.



Vulnerable Time

The vulnerable time for CSMA is the propagation time T_p . This is the time needed for a signal to propagate from one end of the medium to the other. When a station sends a frame, and any other station tries to send a frame during this time, a collision will result. But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending.



Persistence Methods

What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions: *the I-persistent method, the nonpersistent method, and the p-persistent method.* Below figure shows the behavior of three persistence methods when a station finds a channel busy.

1-Persistent: The **1-persistent method** is simple and straightforward. In this method, after the station finds the line idle, it sends its frame immediately (with probability I). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.

Non-persistent: In the **non-persistent method,** a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. The non-persistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

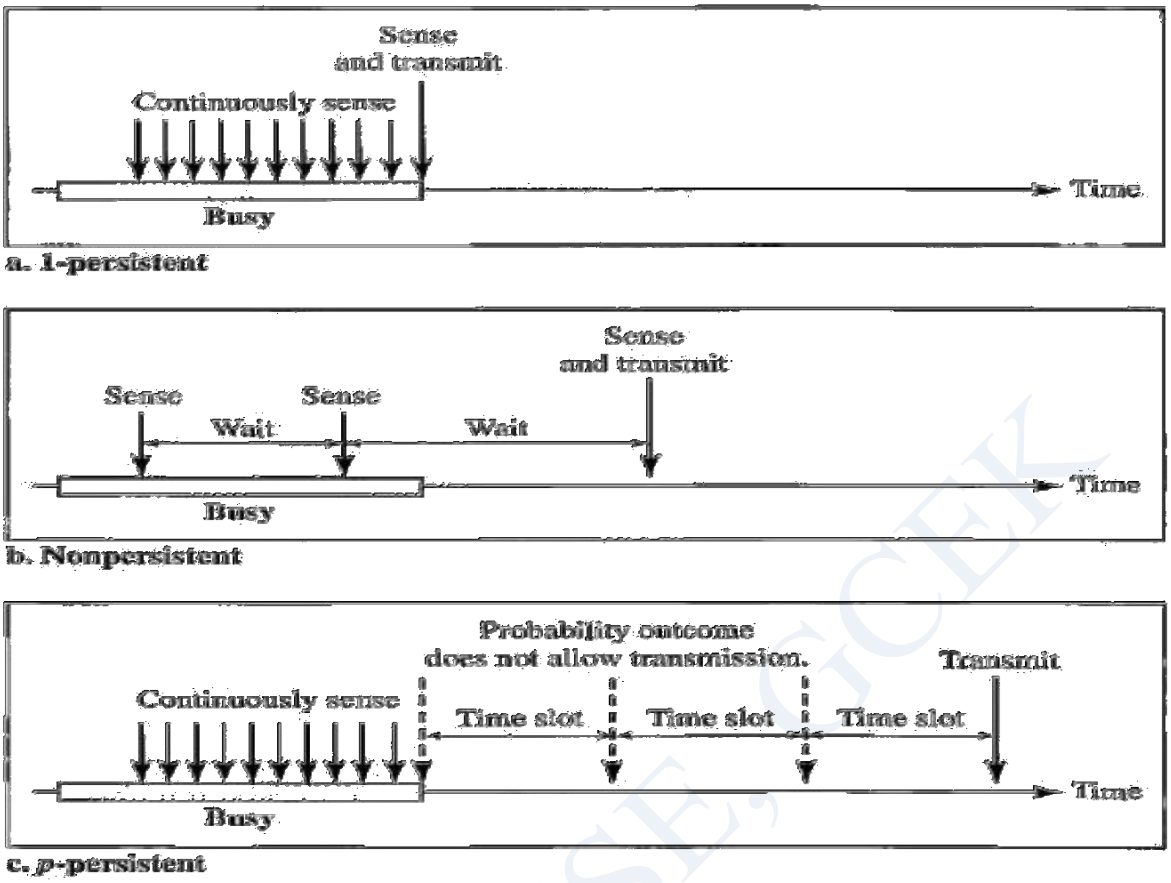
p-Persistent: The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency. In this method, after the station finds the line idle it follows these steps:

1. With probability p, the station sends its frame.

2. With probability q = 1 - p, the station waits for the beginning of the next time slot and checks the line again.

a. If the line is idle, it goes to step 1.

b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

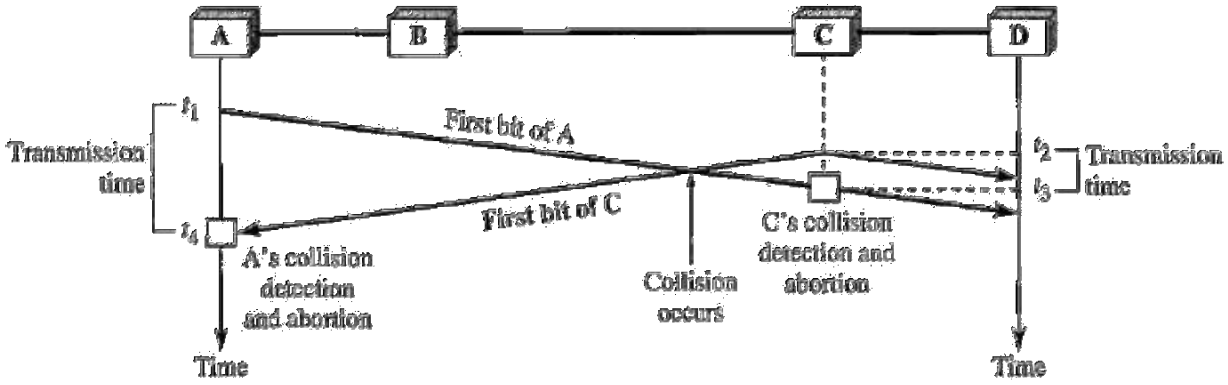


Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

The CSMA method does not specify the procedure following a collision. Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.

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

The basic idea behind CSMA/CD is that a station needs to be able to receive while transmitting to detect a collision. When there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station. To distinguish between these two cases, the received signals in these two cases must be significantly different. In other words, the signal from the second station needs to add a significant amount of energy to the one created by the first station. In a wired network, the received.

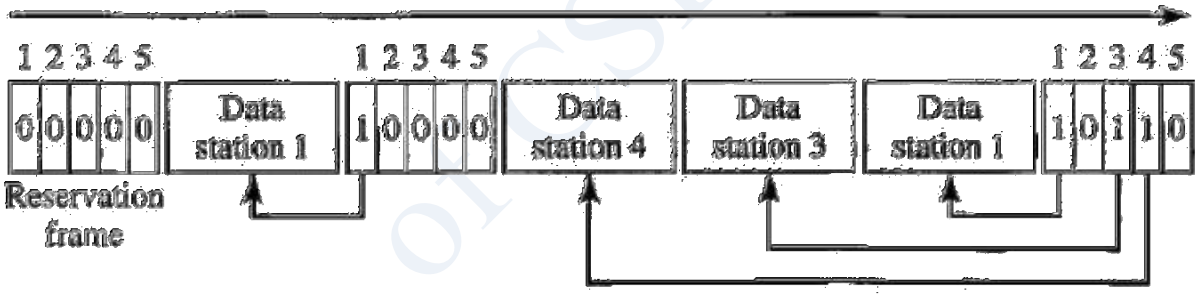


CONTROLLED ACCESS

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.

There are 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.



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.

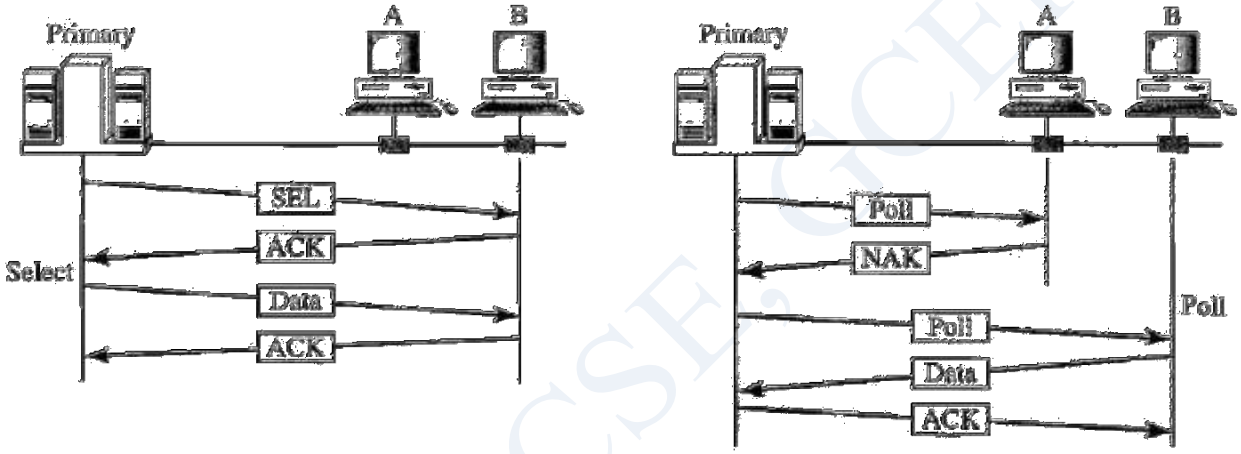
If the primary wants to receive data, it asks the secondary if they have anything to send; this is called poll function. If the primary wants to send data, it tells the secondary to get ready to receive; this is called select function.

1. 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.

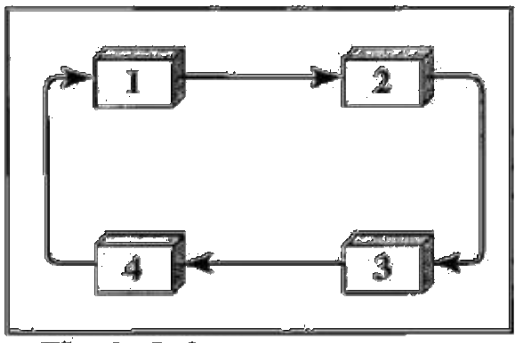
2. Poll

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.

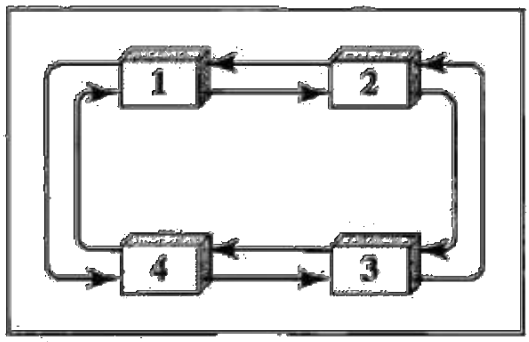


Token Passing: In the token-passing method, the stations in a network are organized in a logical ring. In other words, for each station, there is a *predecessor* and a *successor.* The predecessor is the station which is logically before the station in the ring; the successor is the station which is 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.

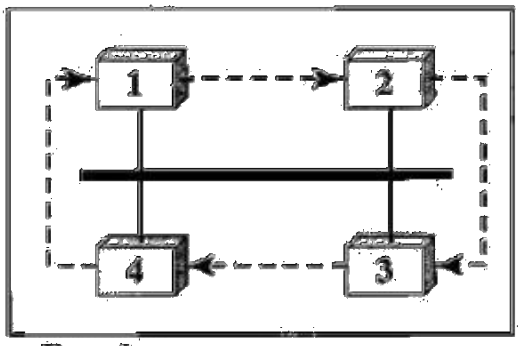
This page is left vacant intentionally



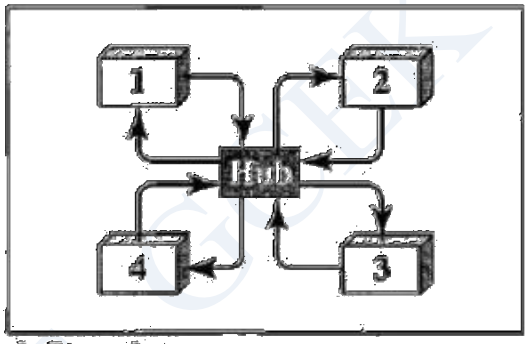
a. Physical ring



b. Dual ring



c. Bus ring



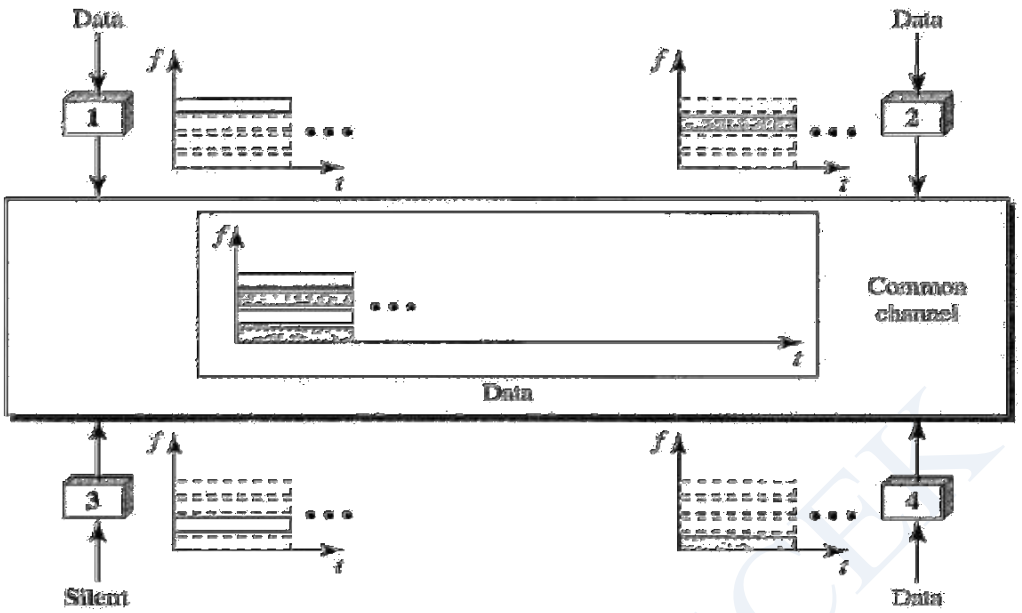
d. Star ring

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.

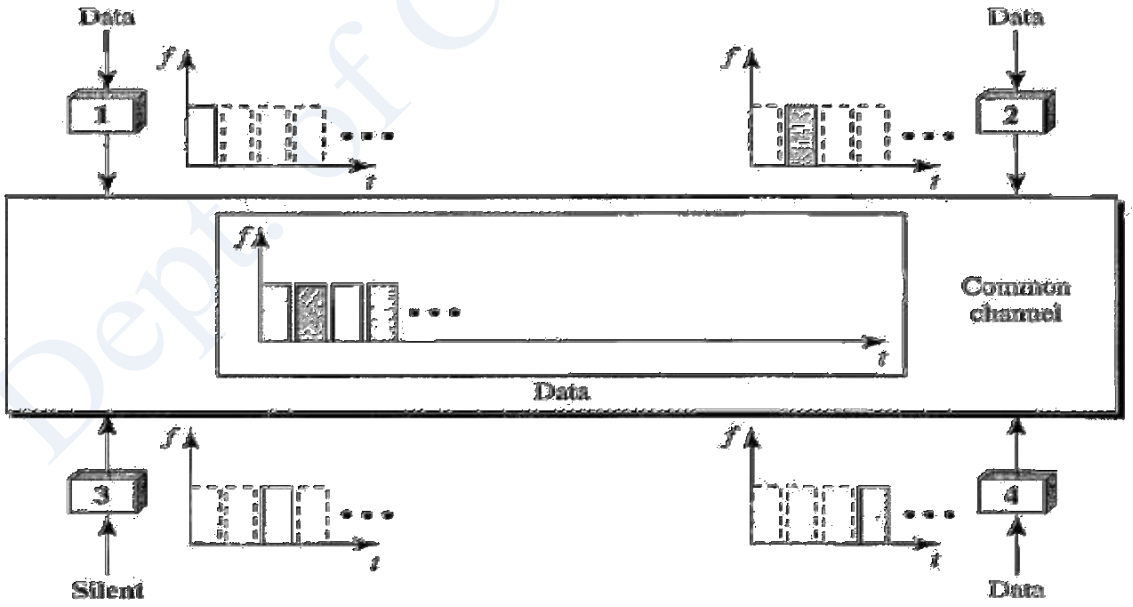
Frequency-Division Multiple Access (FDMA)

In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands. Each station is allocated a band to send its data.



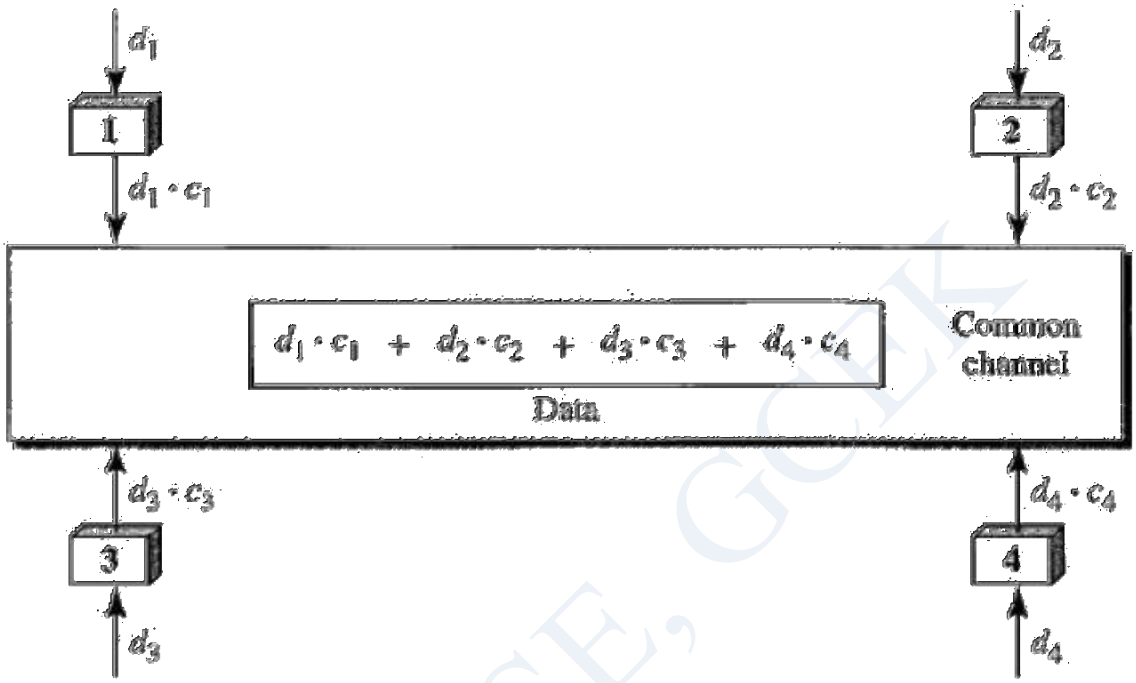
Time-Division Multiple Access (TDMA)

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.



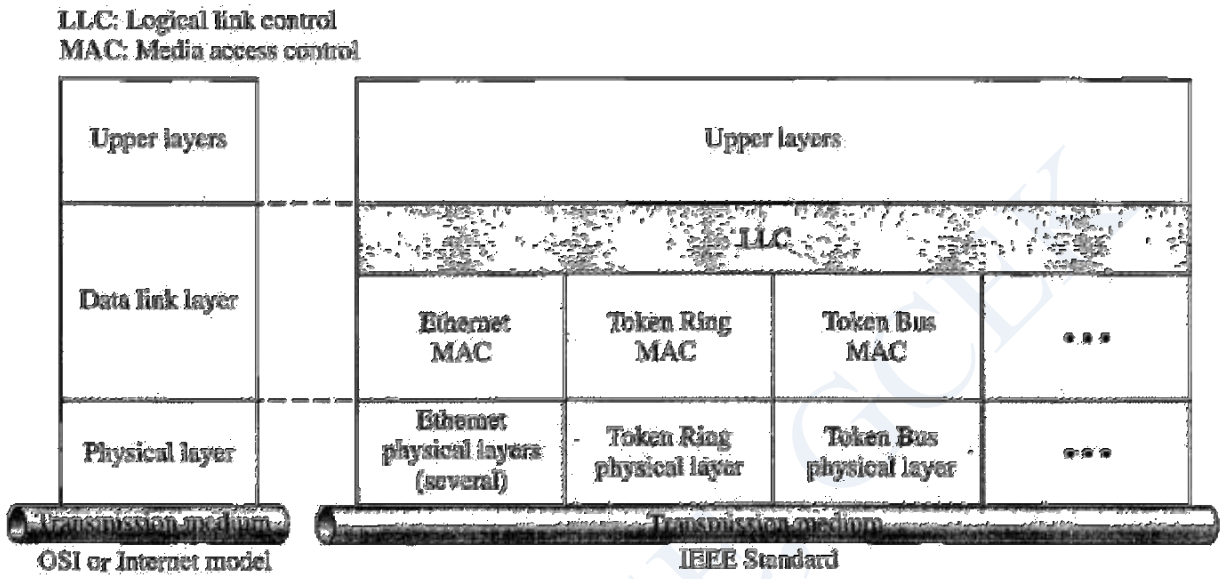
Code-Division Multiple Access (CDMA)

CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link. It differs from TDMA because all stations can send data simultaneously; there is no timesharing.

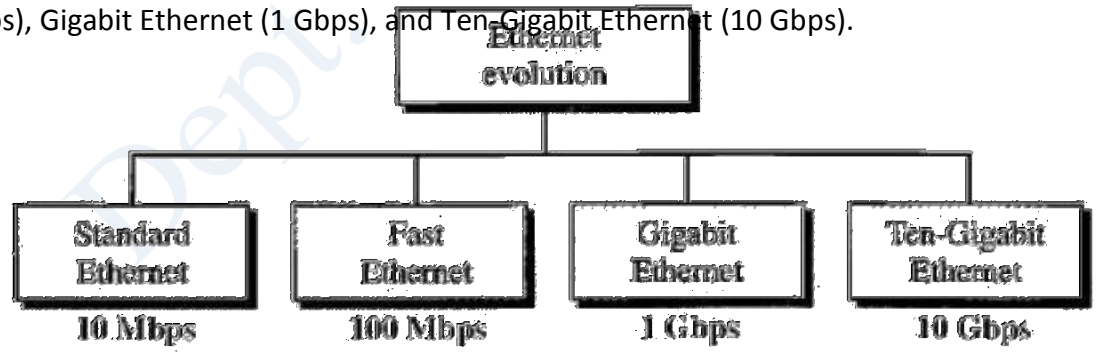


WIRED LANS: Ethernet

The IEEE has subdivided the data link layer into two sub layers: logical link control (LLC) and media access control (MAC). IEEE has also created several physical layer standards for different LAN protocols.



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

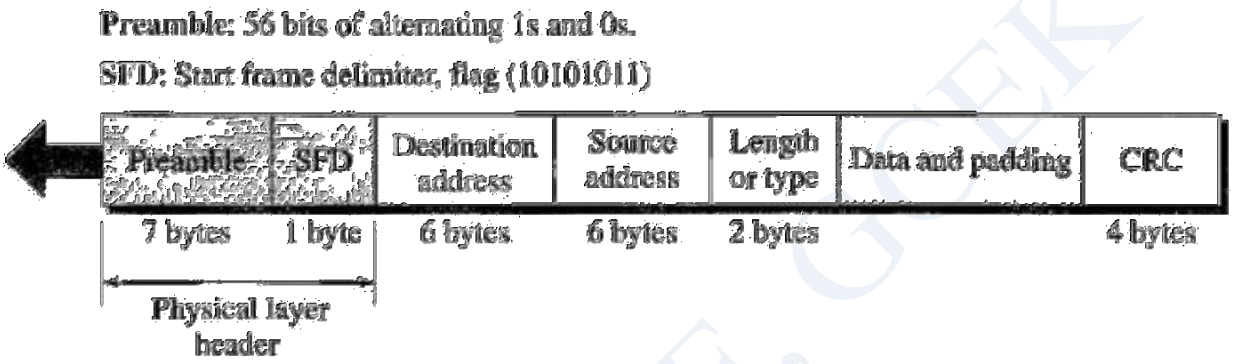


MAC Sublayer

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

Frame Format

The Ethernet frame contains seven fields: preamble, SFD, DA, SA, length or type of protocol data unit (PDU), upper-layer data, and the CRe. Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium. Acknowledgments must be implemented at the higher layers. The format of the MAC frame is shown in below figure:



Preamble: The first field of the 802.3 frame contains 7 bytes (56 bits) of alternating 0s and 1s that alerts the receiving system to the coming frame and enables it to synchronize its input timing.

Start frame delimiter (SFD): The second field (1byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits is 11 and alerts the receiver that the next field is the destination address.

Destination address (DA): The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet. We will discuss addressing shortly.

Source address (SA): The SA field is also 6 bytes and contains the physical address of the sender of the packet. We will discuss addressing shortly.

Length or type: This field is defined as a type field or length field. The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame. The IEEE standard used it as the length field to define the number of bytes in the data field. Both uses are common today.

Data: This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes.

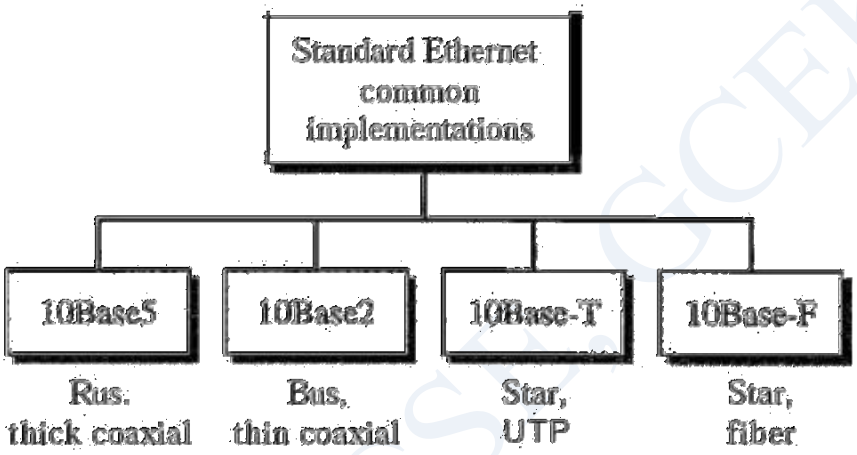
CRC: The last field contains error detection information, in this case a CRC-32.

Physical Layer

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

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.



lOBase5: Thick Ethernet

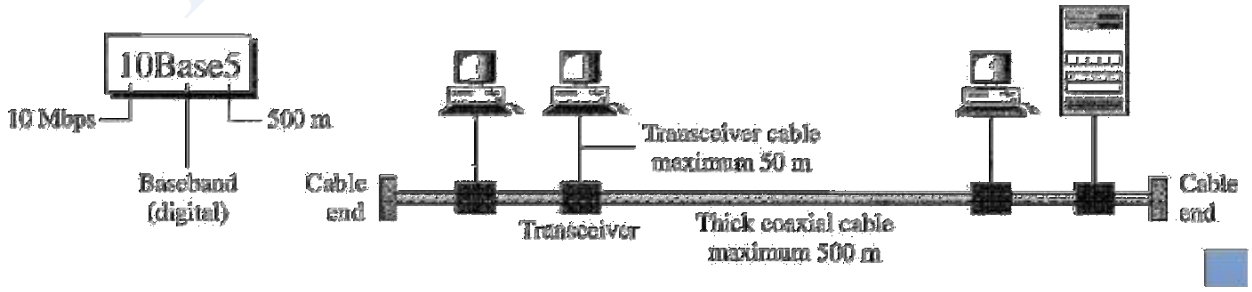
The first implementation is called **10Base5, thick Ethernet, or Thicknet.**

10Base5 was the first Ethernet specification to use a bus topology with an external **transceiver**

(transmitter/receiver) connected via a tap to a thick coaxial cable.

The transceiver is responsible for transmitting, receiving, and detecting collisions.

The transceiver is connected to the station via a transceiver cable that provides separate paths for sending and receiving.



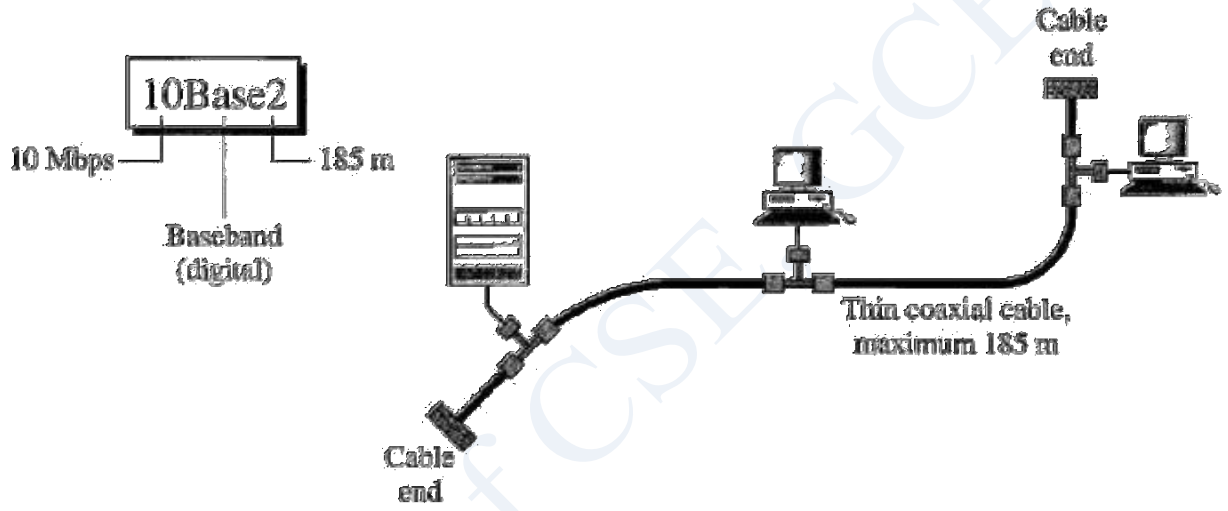
The maximum length of the coaxial cable must not exceed *500* m, otherwise, there is excessive degradation of the signal.

10Base2: Thin Ethernet

The second implementation is called lOBase2, **thin** Ethernet, or Cheapernet.

10Base2 also uses a bus topology, but the cable is much thinner and more flexible.

The cable can be bent to pass very close to the stations. In this case, the transceiver is normally part of the network interface card (NIC), which is installed inside the station.



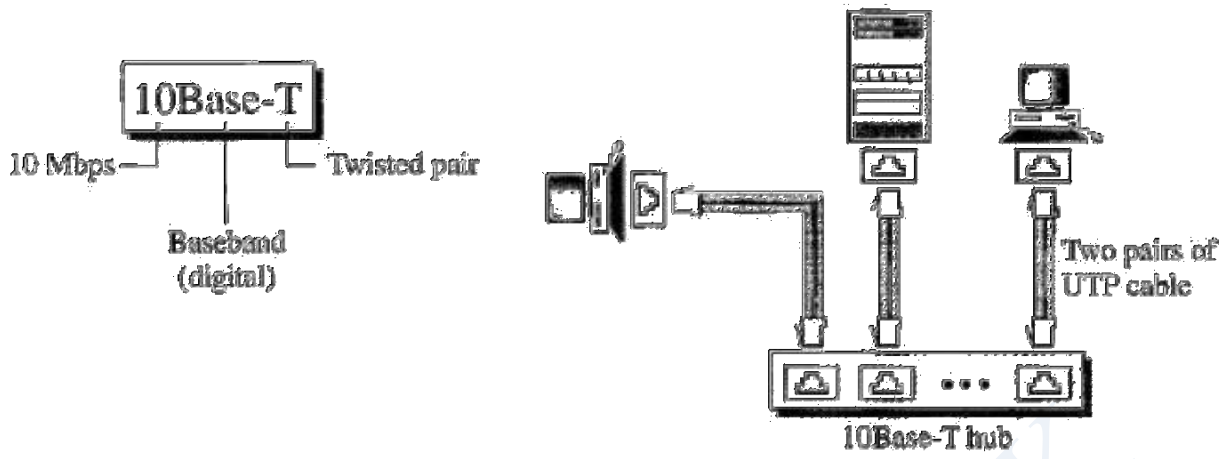
This implementation is more cost effective than 10Base5 because thin coaxial cable is less expensive than thick coaxial and the tee connections are much cheaper than taps.

Installation is simpler because the thin coaxial cable is very flexible.

However, the length of each segment cannot exceed *185* m (close to 200 m) due to the high level of attenuation in thin coaxial cable.

10Base-T: Twisted-Pair Ethernet

The third implementation is called 10Base-T or twisted-pair Ethernet. 10Base-T uses a physical star topology.



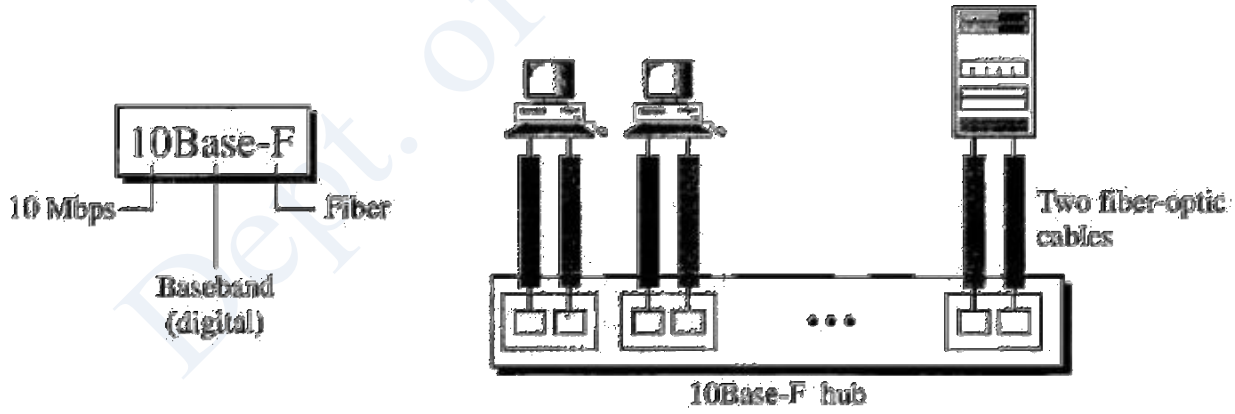
Any collision here happens in the hub.

Compared to l0BaseS or l0Base2, we can see that the hub actually replaces the coaxial cable as far as a collision is concerned.

The maximum length of the twisted cable here is defined as 100 m, to minimize the effect of attenuation in the twisted cable.

10Base-F: Fiber Ethernet

Although there are several types of optical fiber lO-Mbps Ethernet, the most common is called 10Base-F. 10Base-F uses a star topology to connect stations to a hub.



Fast Ethernet

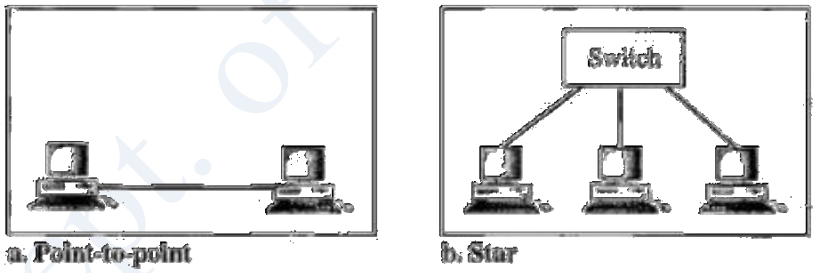
Fast Ethernet was designed to compete with LAN protocols such as FDDI or Fiber Channel (or Fibre Channel, as it is sometimes spelled). IEEE created Fast Ethernet under the name 802.3u. Fast Ethernet is backward-compatible with Standard Ethernet, but it can transmit data 10 times faster at a rate of 100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

Physical Layer

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

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



[Fast Ethernet Topology]

Encoding: Manchester encoding needs a 200-Mbaud bandwidth for a data rate of 100 Mbps, which makes it unsuitable for a medium such as twisted-pair cable. For this reason, the Fast Ethernet designers sought some alternative encoding/decoding scheme.

Gigabit Ethernet

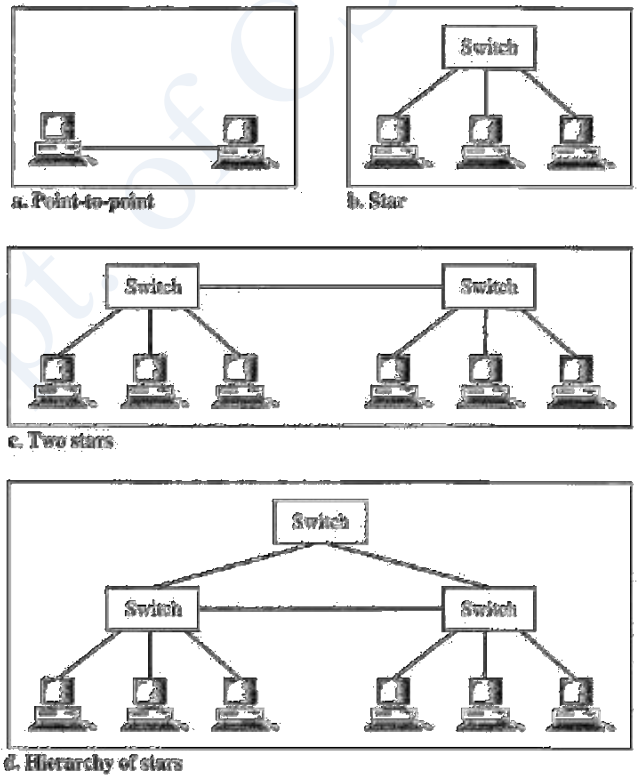
The need for an even higher data rate resulted in the design of the Gigabit Ethernet protocol (1000 Mbps). The IEEE committee calls the Standard 802.3z. The goals of the Gigabit Ethernet design can be summarized as follows:

1. Upgrade the data rate to 1 Gbps.
2. Make it compatible with Standard or Fast Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format.
5. Keep the same minimum and maximum frame lengths.
6. To support auto negotiation as defined in Fast Ethernet.

Physical Layer

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

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



a. Point-to-point
b. Star
c. Two stars
d. Hierarchy of stars

Encoding: Gigabit Ethernet cannot use the Manchester encoding scheme because it involves a very high bandwidth. The two-wire implementations use an NRZ scheme, but NRZ does not self-synchronize properly.

Token Bus (IEEE 802.4)

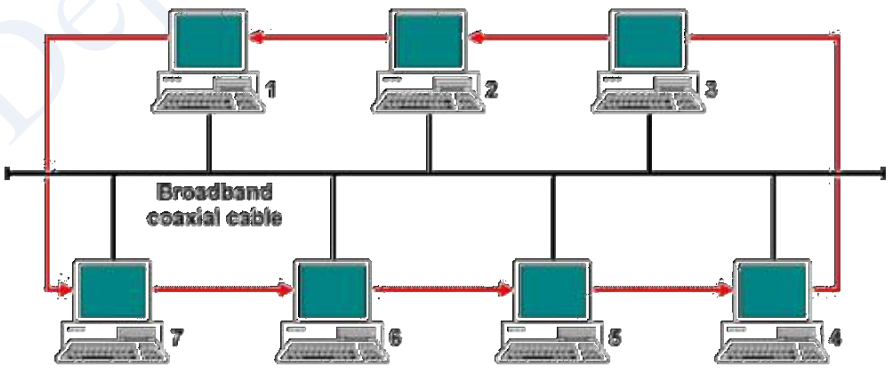
Token bus is a network implementing the token ring protocol over a "virtual ring" on a coaxial cable. A token is passed around the network nodes and only the node possessing the token may transmit. If a node doesn't have anything to send, the token is passed on to the next node on the virtual ring. Each node must know the address of its neighbor in the ring, so a special protocol is needed to notify the other nodes of connections to, and disconnections from, the ring.

Token bus was standardized by IEEE standard 802.4. It is mainly used for industrial applications. Token bus was used by General Motors for their Manufacturing Automation Protocol (MAP) standardization effort. This is an application of the concepts used in token ring networks. The main difference is that the endpoints of the bus do not meet to form a physical ring.

Token Bus suffered from two limitations. Any failure in the bus caused all the devices beyond the failure to be unable to communicate with the rest of the network. Second, adding more stations to the bus was somewhat difficult. Any new station that was improperly attached was unlikely to be able to communicate and all devices beyond it were also affected. Thus, token bus networks were seen as somewhat unreliable and difficult to expand and upgrade.

In order to guarantee the packet delay and transmission in Token bus protocol, a modified Token bus was proposed in Manufacturing Automation Systems and flexible manufacturing system (FMS).

A means for carrying Internet Protocol over token bus was developed.



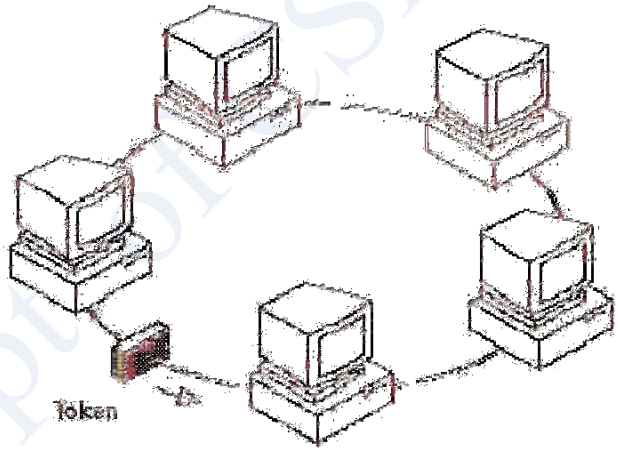
Token ring (IEEE 802.5)

Token ring local area network (LAN) technology is a local area network protocol which resides at the data link layer (DLL) of the OSI model. It uses a special three-byte frame called a token that travels around the ring. Token-possession grants the possessor permission to transmit on the medium. Token ring frames travel completely around the loop.

Stations on a token ring LAN are logically organized in a ring topology with data being transmitted sequentially from one ring station to the next with a control token circulating around the ring controlling access. This token passing mechanism is shared by ARCNET, token bus, and FDDI, and has theoretical advantages over the stochastic CSMA/CD of Ethernet.

Physically, a token ring network is wired as a star, with 'hubs' and arms out to each station and the loop going out-and-back through each.

Each station passes or repeats the special token frame around the ring to its nearest downstream neighbor. This token-passing process is used to arbitrate access to the shared ring media. Stations that have data frames to transmit must first acquire the token before they can transmit them. Token ring LANs normally use differential Manchester encoding of bits on the LAN media.



Token Ring does come with a higher price tag because token ring hardware is more complex and more expensive to manufacture. As a network technology, token ring is passing out of use because it has a maximum speed of 16 Mbps which is slow by today's gigabit Ethernet standards.

Wireless LANs

Wireless communication is one of the fastest-growing technologies. The demand for connecting devices without the use of cables is increasing everywhere. Wireless LANs can be found on college campuses, in office buildings, and in many public areas.

Frame Relay

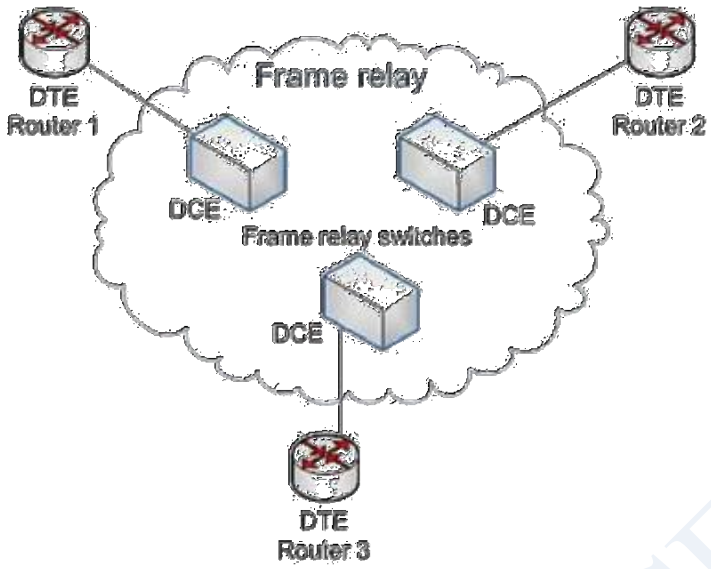
Frame Relay is a standardized wide area network technology that specifies the physical and logical link layers of digital telecommunications channels using a packet switching methodology. Originally designed for transport across Integrated Services Digital Network (ISDN) infrastructure, it may be used today in the context of many other network interfaces.

Frame Relay has its technical base in the older X.25 packet-switching technology, designed for transmitting data on analog voice lines. Unlike X.25, whose designers expected analog signals, Frame Relay offers a fast packet technology, which means that the protocol does not attempt to correct errors. When a Frame Relay network detects an error in a frame, it simply drops that frame. The end points have the responsibility for detecting and retransmitting dropped frames. (However, digital networks offer an incidence of error extraordinarily small relative to that of analog networks.)

Frame Relay often serves to connect local area networks (LANs) with major backbones as well as on public wide-area networks (WANs) and also in private network environments with leased lines over T-1 lines. It requires a dedicated connection during the transmission period. Frame Relay does not provide an ideal path for voice or video transmission, both of which require a steady flow of transmissions. However, under certain circumstances, voice and video transmission do use Frame Relay.

Frame Relay originated as an extension of Integrated Services Digital Network (ISDN). Its designers aimed to enable a packet-switched network to transport the circuit-switched technology. The technology has become a stand-alone and cost-effective means of creating a WAN.

Frame Relay switches create virtual circuits to connect remote LANs to a WAN. The Frame Relay network exists between a LAN border device, usually a router, and the carrier switch. The technology used by the carrier to transport data between the switches is variable and may differ among carriers (i.e. to function, a practical Frame Relay implementation need not rely solely on its own transportation mechanism).



Protocol data unit

Each Frame Relay Protocol data unit (PDU) consists of the following fields:

1. **Flag Field**. The flag is used to perform high-level data link synchronization which indicates the beginning and end of the frame with the unique pattern 01111110. To ensure that the 01111110 pattern does not appear somewhere inside the frame, bit stuffing and destuffing procedures are used.
2. **Address Field**. Each address field may occupy octet 2 to 3, octet 2 to 4, or octet 2 to 5, depending on the range of the address in use. A two-octet address field comprises the EA=ADDRESS FIELD EXTENSION BITS and the C/R=COMMAND/RESPONSE BIT.
    1. **DLCI**-Data Link Connection Identifier Bits. The DLCI serves to identify the virtual connection so that the receiving end knows which information connection a frame belongs to. Note that this DLCI has only local significance. A single physical channel can multiplex several different virtual connections.
    2. **FECN, BECN, DE** bits. These bits report congestion:

        FECN=Forward Explicit Congestion Notification bit

        BECN=Backward Explicit Congestion Notification bit

        DE=Discard Eligibility bit
3. **Information Field**. A system parameter defines the maximum number of data bytes that a host can pack into a frame. Hosts may negotiate the actual maximum frame length at call set-up time. The standard specifies the maximum information field size (supportable by any network)

as at least 262 octets. Since end-to-end protocols typically operate on the basis of larger information units, Frame Relay recommends that the network support the maximum value of at least 1600 octets in order to avoid the need for segmentation and reassembling by end-users.

4. **Frame Check Sequence (FCS) Field**. Since one cannot completely ignore the bit error-rate of the medium, each switching node needs to implement error detection to avoid wasting bandwidth due to the transmission of *err*ed frames. The error detection mechanism used in Frame Relay uses the cyclic redundancy check (CRC) as its basis.

Frame Relay versus X.25

X.25 provides quality of service and error-free delivery, whereas, Frame Relay was designed to relay data as quickly as possible over low error networks. Frame Relay eliminates a number of the higher-level procedures and fields used in X.25. Frame Relay was designed for use on links with error-rates far lower than available when X.25 was designed.

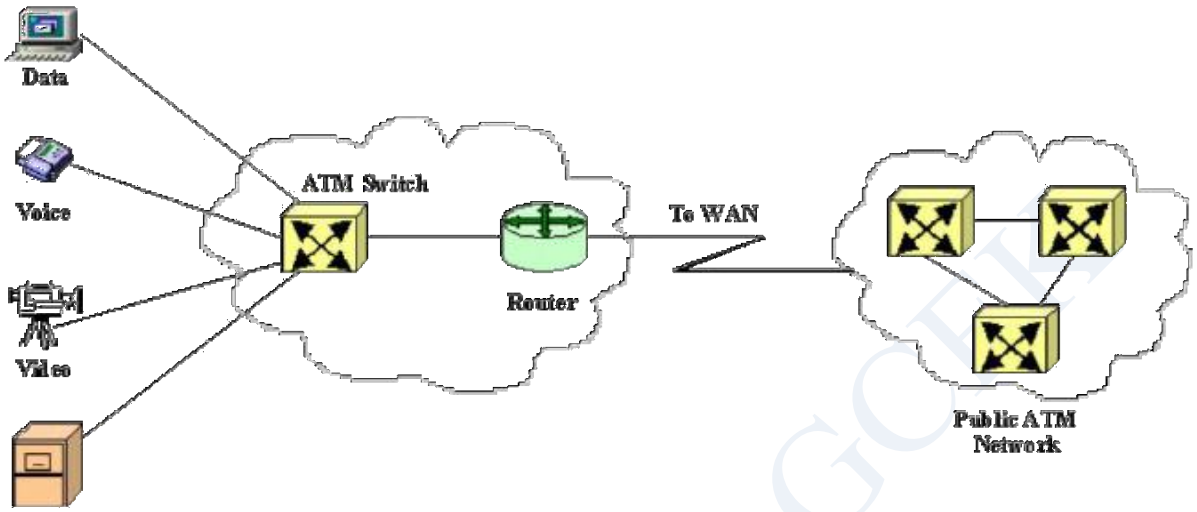
X.25 prepares and sends packets, while Frame Relay prepares and sends frames. X.25 packets contain several fields used for error checking and flow control, most of which are not used by Frame Relay. The frames in Frame Relay contain an expanded link layer address field that enables Frame Relay nodes to direct frames to their destinations with minimal processing. The elimination of functions and fields over X.25 allows Frame Relay to move data more quickly, but leaves more room for errors and larger delays should data need to be retransmitted.

X.25 packet switched networks typically allocated a fixed bandwidth through the network for each X.25 access, regardless of the current load. This resource allocation approach, while apt for applications that require guaranteed quality of service, is inefficient for applications that are highly dynamic in their load characteristics or which would benefit from a more dynamic resource allocation. Frame Relay networks can dynamically allocate bandwidth at both the physical and logical channel level.

ATM

Asynchronous Transfer Mode (ATM) is a standard switching technique, designed to unify telecommunication and computer networks. It uses asynchronous time-division multiplexing, and it encodes data into small, fixed-sized cells. This differs from approaches such as the Internet Protocol or Ethernet that use variable sized packets or frames. ATM provides data link layer services that run over a wide range of OSI physical Layer links. ATM has functional similarity with both circuit

switched networking and small packet switched networking. It was designed for a network that must handle both traditional high-throughput data traffic (e.g., file transfers), and real-time, low-latency content such as voice and video. ATM uses a connection-oriented model in which a virtual circuit must be established between two endpoints before the actual data exchange begins.



Network Layer

Communication at the network layer is host-to-host (computer-to-computer); a computer somewhere in the world needs to communicate with another computer somewhere else in the world.

Usually, computers communicate through the Internet. The packet transmitted by the sending computer may pass through several LANs or WANs before reaching the destination computer.

The Internet addresses are 32 bits in length; this gives us a maximum of 2^{32} addresses. These addresses are referred to as IPv4 (IP version 4) addresses or simply IP addresses if there is no confusion.

The need for more addresses, in addition to other concerns about the IP layer, motivated a new design of the IP layer called the new generation of IP or IPv6 (IP version 6). In this version, the Internet uses 128-bit addresses that give much greater flexibility in address allocation. These addresses are referred to as IPv6 (IP version 6) addresses.

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.

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.

Address Space

A protocol such as IPv4 that defines addresses has an address space.

An address space is the total number of addresses used by the protocol. If a protocol uses N bits to define an address, the address space is 2^N because each bit can have two different values (0 or 1) and N bits can have 2^N values.

IPv4 uses 32-bit addresses, which means that the address space is 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.

Notations

Each TCP/IP host is identified by a logical IP address. This address is unique for each host that communicates by using TCP/IP. Each 32-bit IP address identifies a location of a host system on the network in the same way that a street address identifies a house on a city street.

Just as a street address has a standard two-part format (a street name and a house number), each IP address is separated internally into two parts--a network ID and a host ID:

- The network ID, also known as a network address, identifies a single network segment within a larger TCP/IP internetwork (a network of networks). All the systems that attach and share access to the same network have a common network ID within their full IP address. This ID is also used to uniquely identify each network within the larger internetwork.

- The host ID, also known as a host address, identifies a TCP/IP node (a workstation, server, router, or other TCP/IP device) within each network. The host ID for each device identifies a single system uniquely within its own network.

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

Example 1

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

Solution:

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation.

a. 129.11.11.239

b. 193.131.27.255

Example 2

Change the following IPv4 addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78

b. 221.34.7.82

Solution :

We replace each decimal number with its binary equivalent (see Appendix B).

a. 01101111 00111000 00101101 01001110

b. 11011101 00100010 00000111 01010010

Example 3

Find the error, if any, in the following IPv4 addresses.

a. 111.56.045.78

b. 221.34.7.8.20

c. 75.45.301.14

d. 11100010.23.14.67

Solution :

a. There must be no leading zero (045).

b. There can be no more than four numbers in an IPv4 address.

c. Each number needs to be less than or equal to 255 (301 is outside this range).

d. A mixture of binary notation and dotted-decimal notation is not allowed.

Classful Addressing

IPv4 addressing, at its inception, used the concept of classes. This architecture is called classful 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.

[w.x.y.z]

Class	Value of *w*	Network ID	Host ID	Number of networks	Number of hosts per network
A	0-127	*w*	*x.y.z*	126	16,777,214
B	128-191	*w.x*	*y.z*	16,384	65,534
C	192-223	*w.x.y*	*z*	2,097,152	254
D	224-239	Reserved for multicast addressing	N/A	N/A	N/A
E	240-255	Reserved for experimental use	N/A	N/A	N/A

Example 4

Find the class of each address.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

Solution:

a. The first bit is O. This is a class A address.

b. The first 2 bits are 1; the third bit is O. This is a class C address.

c. The first byte is 14 (between 0 and 127); the class is A.

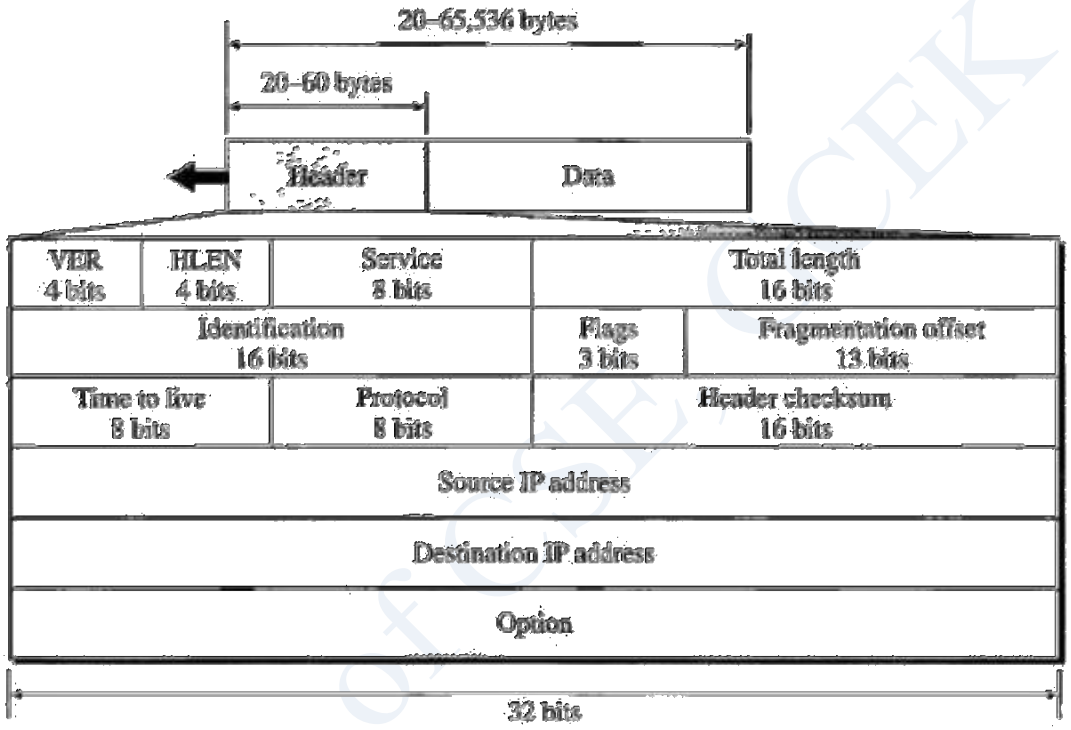
d. The first byte is 252 (between 240 and 255); the class is E.

IPv4

IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service. The term best-effort means that IPv4 provides no error control or flow control (except for error detection on the header). IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees.

Datagram:

Packets in the IPv4 layer are called datagrams. Below figure shows the IPv4 datagram format.



Version (VER)

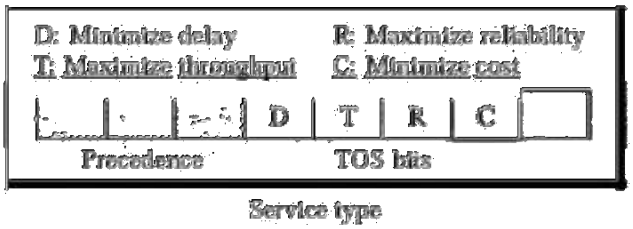
This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4. However, version 6 (or IPng) may totally replace version 4 in the future.

Header length (HLEN)

This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are no options, the header length is 20 bytes, and the value of this field is 5 (5 x 4 = 20). When the option field is at its maximum size, the value of this field is 15 (15 x 4 = 60).

Services

This field, previously called service type, is now called differentiated services.



Service Type

In this interpretation, the first 3 bits are called precedence bits. The next 4 bits are called type of service (TOS) bits, and the last bit is not used.

Precedence is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary). The precedence defines the priority of the datagram in issues such as congestion. If a router is congested and needs to discard some datagrams, those datagrams with lowest precedence are discarded first.

TOS bits is a 4-bit subfield with each bit having a special meaning. Although a bit can be either 0 or 1, one and only one of the bits can have the value of 1 in each datagram. The bit patterns and their interpretations are given in below table.

TOS	Bits Description
0000	Normal (default)
0001	Minimize Cost
0010	Maximize Reliability
0100	Maximize Throughput
1000	Minimize Delay

Total length

This is a 16-bit field that defines the total length (header plus data) of the IPv4 datagram in bytes.
Length of data =total length - header length

Identification

This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IPv4 address must uniquely define a datagram as it leaves the source host.

Flags

This is a 3-bit field. The first bit is reserved. The second bit is called the do not fragment bit. If its value is 1, the machine must not fragment the datagram. The third bit is called the more fragment bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

Fragmentation offset

This 13-bit field shows the relative position of this fragment with respect to the whole datagram.

Time to live

A datagram has a limited lifetime in its travel through an internet. This field was originally designed to hold a timestamp, which was decremented by each visited router. The datagram was discarded when the value became zero.

Protocol

This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer. An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IPv4 datagram is delivered.

Checksum

The checksum concept and its calculation are discussed later in this chapter.

Source address

This 32-bit field defines the IPv4 address of the source. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

Destination address

This 32-bit field defines the IPv4 address of the destination. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

Options

The header of the IPv4 datagram is made of two parts: a fixed part and a variable part. The fixed part is 20 bytes long and was discussed in the previous section. The variable part comprises the options that can be a maximum of 40 bytes.

IPv6

IPv6 (Internetworking Protocol, version 6), also known as **IPng** (Internetworking Protocol, next generation), was proposed and is now a standard. In IPv6, the Internet protocol was extensively modified to accommodate the unforeseen growth of the Internet. The format and the length of the IP address were changed along with the packet format.

Advantages

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

Larger address space: An IPv6 address is 128 bits long. Compared with the 32-bit address of IPv4, this is a huge (296) increase in the address space.

Better header format: IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.

New options: IPv6 has new options to allow for additional functionalities.

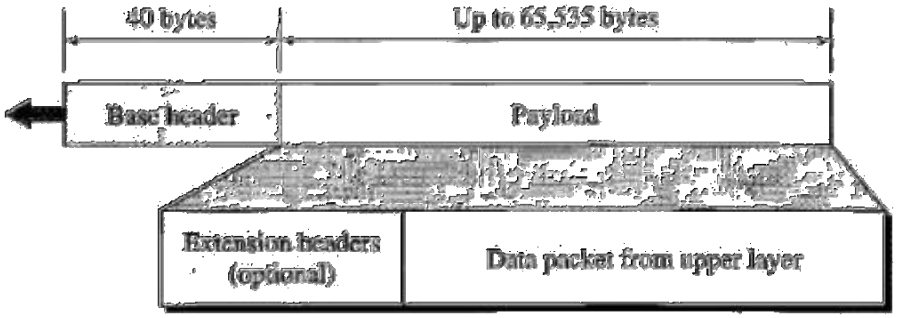
Allowance for extension: IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.

Support for resource allocation: In IPv6, the type-of-service field has been removed, but a mechanism has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.

Support for more security: The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

Packet Format

The IPv6 packet is shown in below Figure. Each packet is composed of a mandatory base header followed by the payload. The payload consists of two parts: optional extension headers and data from an upper layer. The base header occupies 40 bytes, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information.



[IPv6 datagram header and payload]

Base Header

Below figure shows the base header with its eight fields.

These fields are as follows:

Version: This 4-bit field defines the version number of the IP. For IPv6, the value is 6.

Priority: The 4-bit priority field defines the priority of the packet with respect to traffic congestion.

Flow label. The flow label is a 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data.

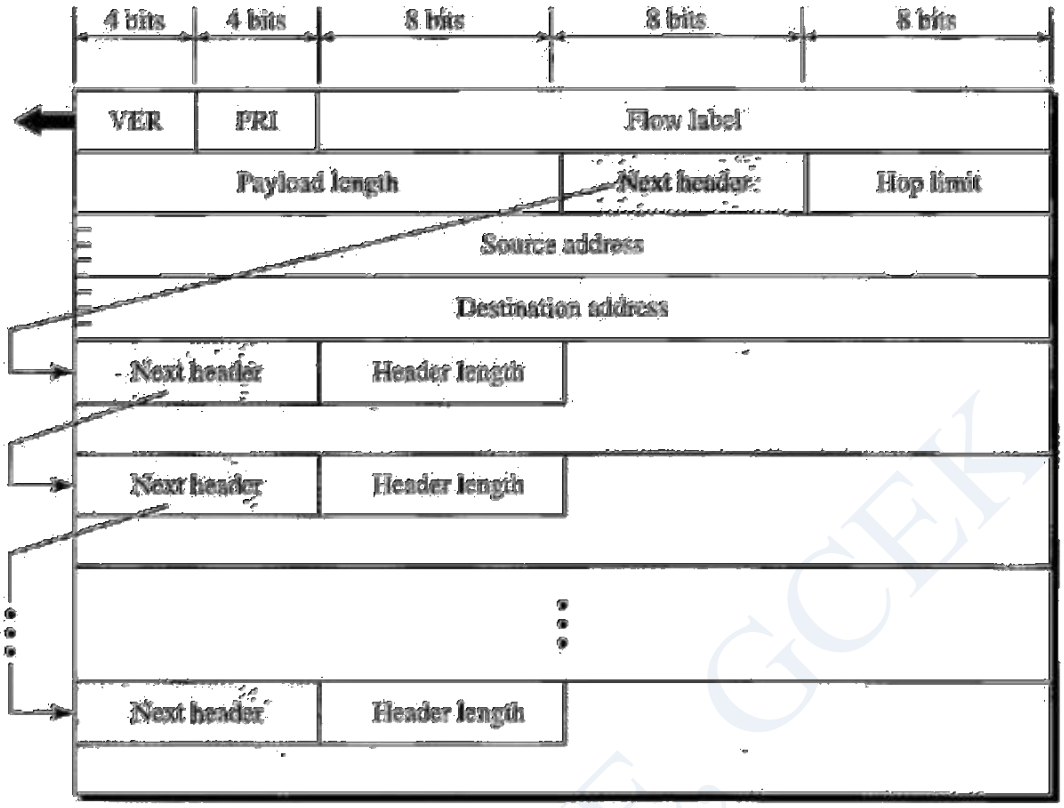
Payload length. The 2-byte payload length field defines the length of the IP datagram excluding the base header.

Next header. The next header is an 8-bit field defining the header that follows the base header in the datagram. The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP. Each extension header also contains this field. Table 20.6 shows the values of next headers. Note that this field in version 4 is called the *protocol.*

Hop limit. This 8-bit hop limit field serves the same purpose as the TTL field in IPv4.

Source address. The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram.

Destination address: The destination address field is a 16-byte (128-bit) Internet address that usually identifies the final destination of the datagram. However, if source routing is used, this field contains the address of the next router.



[Format of an IPv6 datagram]

Addressing

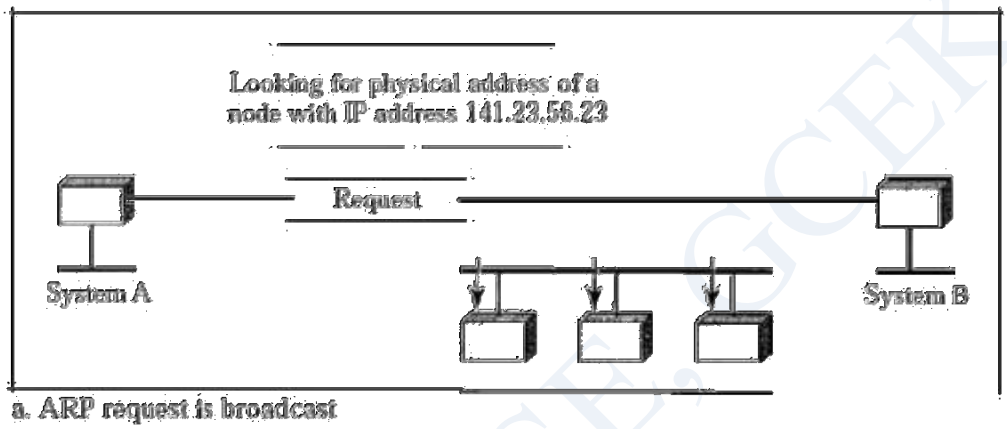
Delivery of a packet to a host or a router requires two levels of addressing: logical and physical. We need to be able to map a logical address to its corresponding physical address and vice versa. These can be done by using either static or dynamic mapping.

Static mapping involves a table that is stored in each machine on the network. Each machine that knows, for example, the IP address of another machine but not its physical address can look it up in the table.

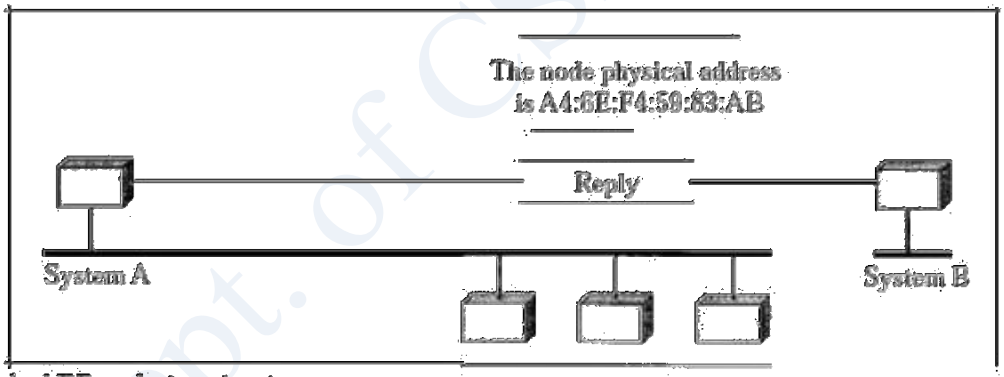
In dynamic mapping each time a machine knows one of the two addresses (logical or physical), it can use a protocol to find the other one.

Mapping Logical to Physical Address: ARP

Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver. The logical (IP) address is obtained from the DNS if the sender is the host or it is found in a routing table if the sender is a router. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver. The host or the router sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network.

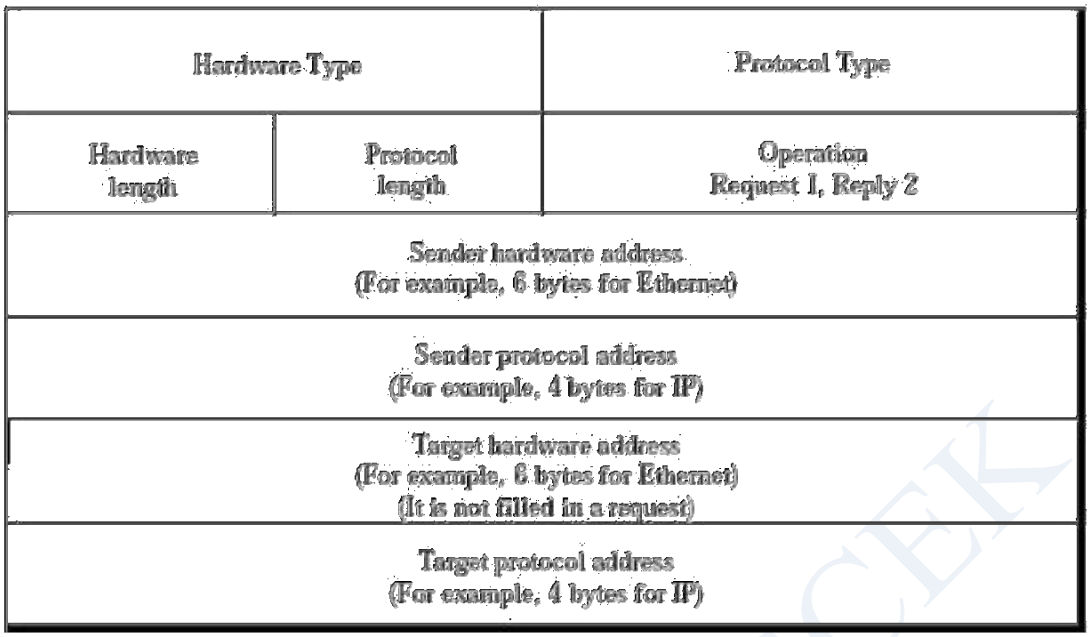


a. ARP request is broadcast



b. ARP reply is unicast

Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and physical addresses. The packet is unicast directly to the inquirer by using the physical address received in the query packet.



The fields are as follows:

- **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) and ARP 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 all 0s 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.

RARP

Reverse Address Resolution Protocol (RARP) finds the logical address for a machine that knows only its physical address. Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the physical (hardware) address of the machine. 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.

ICMP

The IP protocol is a best-effort delivery service that delivers a datagram from its original source to its final destination. However, it has two deficiencies: lack of error control and lack of assistance mechanisms.

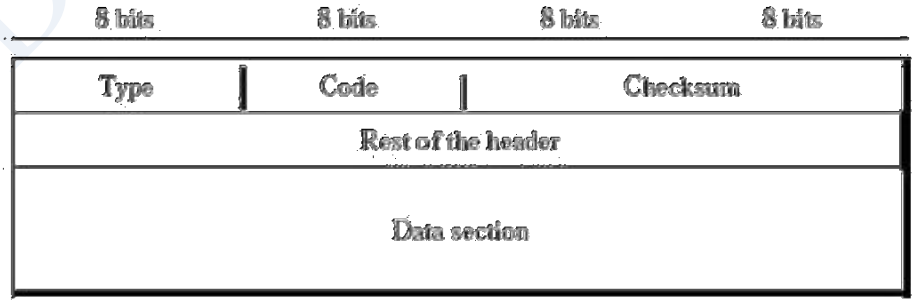
The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol.

Types of Messages

ICMP messages are divided into two broad categories: error-reporting messages and query messages.

The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.

The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host.



An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all.

One of the main responsibilities of ICMP is to report errors. Although technology has produced increasingly reliable transmission media, errors still exist and must be handled. Five types of errors are handled: destination unreachable, source quench, time exceeded, parameter problems, and redirection.

Destination Unreachable: When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a destination-unreachable message back to the source host that initiated the datagram.

Source-quench: The source-quench message in ICMP was designed to add a kind of flow control to the IP. When a router or host discards a datagram due to congestion, it sends a source-quench message to the sender of the datagram. This message has two purposes. First, it informs the source that the datagram has been discarded.

Second, it warns the source that there is congestion somewhere in the path and that the source should slow down (quench) the sending process.

Time Exceeded: When the datagram is discarded, a time-exceeded message must be sent by the router to the original source. Second, a time-exceeded message is also generated when not all fragments that make up a message arrive at the destination host within a certain time limit.

Parameter Problem: Any ambiguity in the header part of a datagram can create serious problems as the datagram travels through the Internet. If a router or the destination host discovers an ambiguous or missing value in any field of the datagram, it discards the datagram and sends a parameter-problem message back to the source.

Redirection: When a router needs to send a packet destined for another network, it must know the IP address of the next appropriate router. The same is true if the sender is a host. Both routers and hosts, then, must have a routing table to find the address of the router or the next router.

ICMPV6

The ARP and IGMP protocols in version 4 are combined in ICMPv6. Just as in ICMPv4, we divide the ICMP messages into two categories. However, each category has more types of messages than before.

Error Reporting: As we saw in our discussion of version 4, one of the main responsibilities of ICMP is to report errors. Five types of errors are handled: destination unreachable, packet too big, time exceeded parameter problems, and redirection.

Destination Unreachable: The concept of the destination-unreachable message is exactly the same as described for ICMP version 4.

Packet Too Big: This is a new type of message added to version 6. If a router receives a datagram that is larger than the maximum transmission unit (MTU) size of the network through which the datagram should pass, two things happen. First, the router discards the datagram and then an ICMP error packet-a packet-too-big message-is sent to the source.

Time Exceeded: This message is similar to the one in version 4.

Parameter Problem: This message is similar to its version 4 counterpart.

Redirection: The purpose of the redirection message is the same as described for version 4.

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 datagram between two hosts. This is called **host-to-host delivery**.

Real communication takes place between two processes (application programs) in a network. This is called process-to process delivery.

The **transport layer** is responsible for **process-to-process delivery**-the delivery of a packet, part of a message, from one process to another.

At the transport layer, we need a transport layer address, called a port number, to choose among multiple processes running on the destination host.

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.

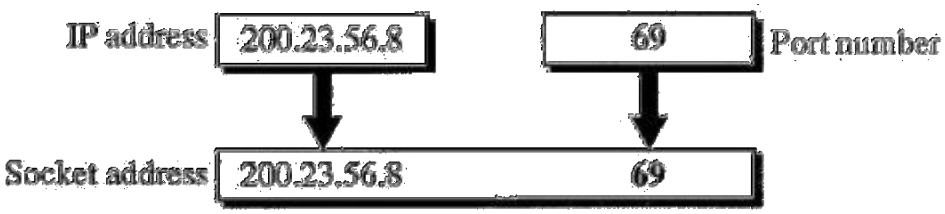
The server process must also define itself with a port number. This port number, however, cannot be chosen randomly.

Socket Addresses

Process-to-process delivery needs two identifiers, IP address and the port number, at each end to make a connection.

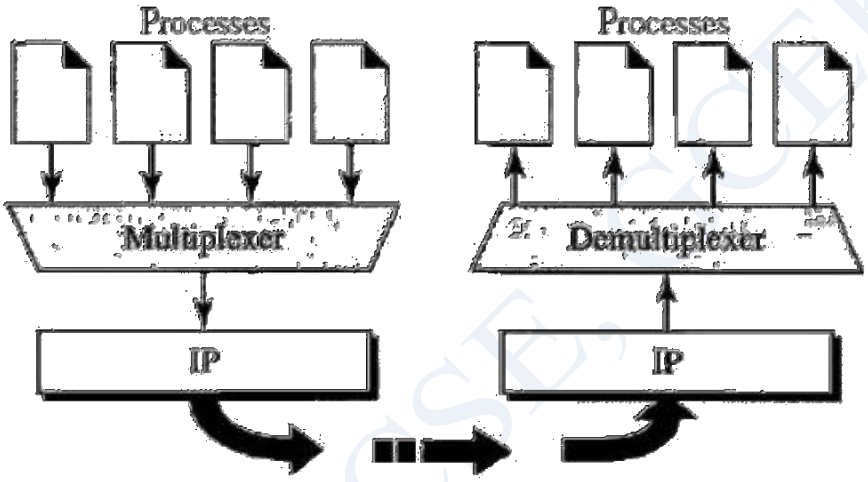
The combination of an IP address and a port number is called a socket address. The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely.

A transport layer protocol needs a pair of socket addresses: the client socket address and the server socket address.



Multiplexing and Demultiplexing

The addressing mechanism allows multiplexing and demultiplexing by the transport layer.



Multiplexing

At the sender site, there may be several processes that need to send packets. However, 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

A transport layer protocol can either be connectionless or connection-oriented.

Connectionless 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.

Connection Oriented Service

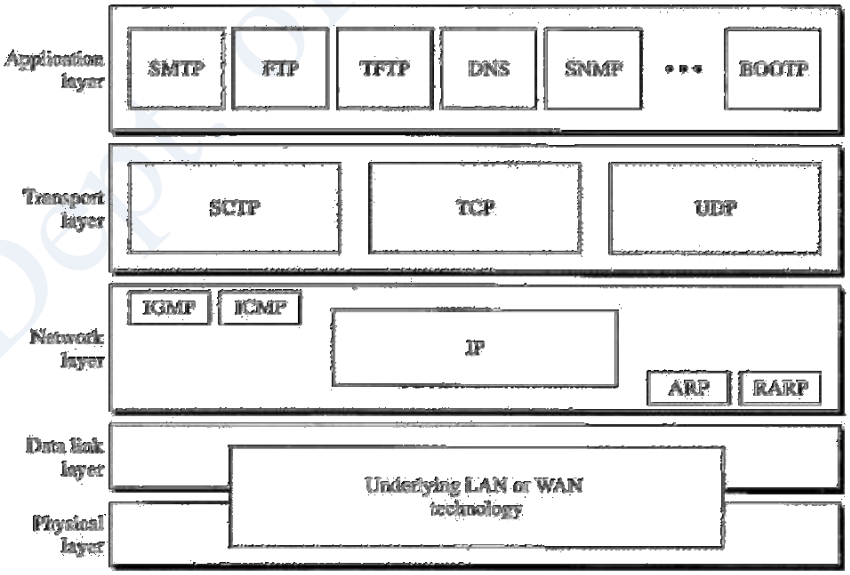
In a connection-oriented service, a connection is first established between the sender and the receiver. Data are transferred. At the end, the connection is released. We will see shortly that TCP and SCTP are connection-oriented protocols.

Reliable Versus Unreliable

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.



[Position of TCP and UDP]

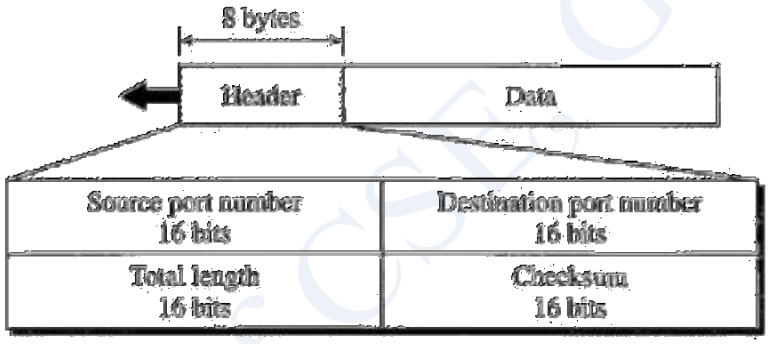
User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication. Also, it performs very limited error checking.

If UDP is so powerless, why would a process want to use it? With the disadvantages come some advantages. UDP is a very simple protocol using a minimum of overhead. If a process wants to send a small message and does not care much about reliability, it can use UDP. Sending a small message by using UDP takes much less interaction between the sender and receiver than using TCP.

User Datagram

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



[User Datagram Format]

The fields are as follows:

Source port number

This is the port number used by the process running on the source host. It is 16 bits long, which means that the port number can range from 0 to 65,535.

If the source host is the client (a client sending a request), the port number, in most cases, is an ephemeral port number requested by the process and chosen by the UDP software running on the source host.

If the source host is the server (a server sending a response), the port number, in most cases, is a well-known port number.

Destination port number

This is the port number used by the process running on the destination host. It is also 16 bits long.

If the destination host is the server (a client sending a request), the port number, in most cases, is a well-known port number.

If the destination host is the client (a server sending a response), the port number, in most cases, is an ephemeral port number. In this case, the server copies the ephemeral port number it has received in the request packet.

Length

This is a 16-bit field that defines the total length of the user datagram, header plus data. The 16 bits can define a total length of 0 to 65,535 bytes.

Checksum

This field is used to detect errors over the entire user datagram (header plus data).

UDP Operation

UDP uses concepts common to the transport layer.

Connectionless Services: As mentioned previously, UDP provides a connectionless service. This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program. The user datagrams are not numbered. Also, there is no connection establishment and no connection termination, as is the case for TCP. This means that each user datagram can travel on a different path.

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

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

Queuing: In UDP, queues are associated with ports. At the client site, when a process starts, it requests a port number from the operating system. Some implementations create both an incoming and an outgoing queue associated with each process. Other implementations create only an incoming queue associated with each process.

Use of UDP

The following lists some uses of the UDP protocol:

UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control.

UDP is suitable for a process with internal flow and error control mechanisms. For example, the Trivial File Transfer Protocol (TFTP) process includes flow and error control

UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.

UDP is used for management processes such as SNMP.

UDP is used for some route updating protocols such as Routing Information Protocol (RIP).

TCP

TCP is called a connection-oriented, reliable transport protocol.

It adds connection-oriented and reliability features to the services of IP.

TCP Services

Before we discuss TCP in detail, let us explain the services offered by TCP to the processes at the application layer.

Process-to-Process Communication

Like UDP, TCP provides process-to-process communication using port numbers.

Stream Delivery Service

TCP, unlike UDP, is a stream-oriented protocol. In UDP, a process (an application program) sends messages, with predefined boundaries, to UDP for delivery. UDP adds its own header to each of these messages and delivers them to IP for transmission. Each message from the process is called a user datagram and becomes, eventually, one IP datagram. Neither IP nor UDP recognizes any relationship between the datagrams.

TCP, on the other hand, allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes. TCP creates an environment in which the

two processes seem to be connected by an imaginary "tube" that carries their data across the Internet.

Full-Duplex Communication

TCP offers full-duplex service, in which data can flow in both directions at the same time. Each TCP then has a sending and receiving buffer, and segments move in both directions.

Connection-Oriented Service

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

1. The two TCPs establish a connection between them.
2. Data are exchanged in both directions.
3. The connection is terminated.

TCP Segment Format

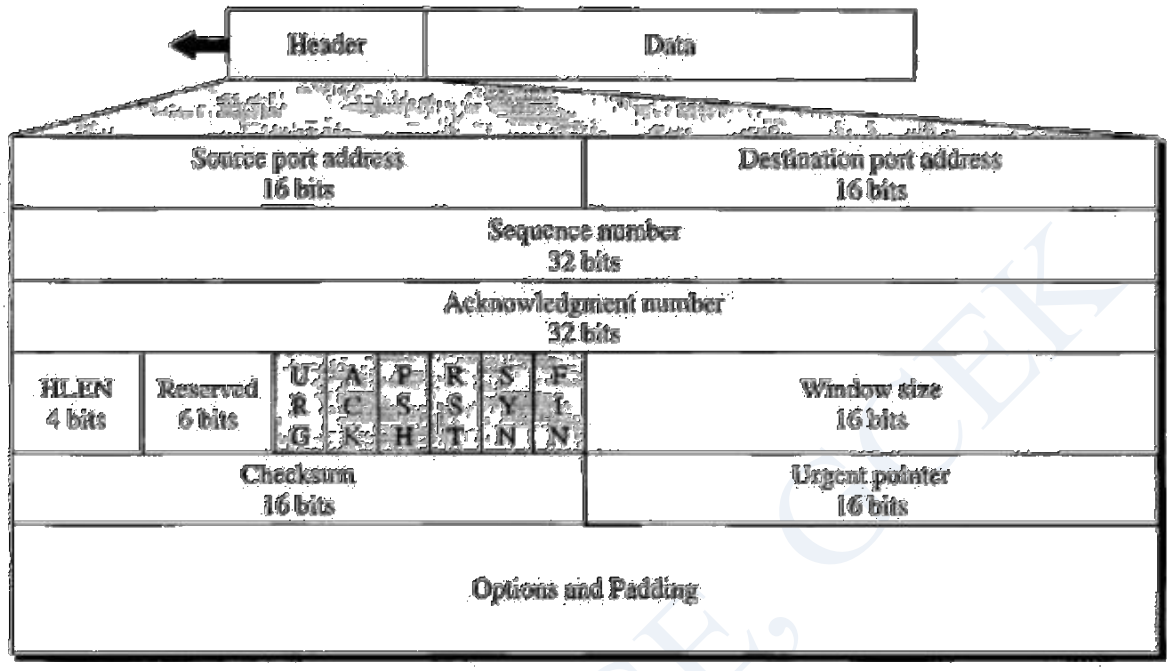
The segment consists of a 20- to 60-byte header, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.

Source port address: This is a 16-bit field that defines the port number of the application program in the host that is sending the segment. This serves the same purpose as the source port address in the UDP header.

Destination port address: This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment. This serves the same purpose as the destination port address in the UDP header.

Sequence number: This 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence comprises the first byte in the segment. During

connection establishment, each party uses a random number generator to create an initial sequence number (ISN), which is usually different in each direction.



Acknowledgment number: This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number *x* from the other party, it defines *x* + I as the acknowledgment number. Acknowledgment and data can be piggybacked together.

Header length: This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field can be between 5 (5 x 4 =20) and 15 (15 x 4 =60).

Reserved: This is a 6-bit field reserved for future use.

Control: This field defines 6 different control bits or flags. One or more of these bits can be set at a time.

URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.

PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection

Window size: This field defines the size of the window, in bytes, that the other party must maintain.

Checksum:

This 16-bit field contains the checksum. The calculation of the checksum for TCP follows the same procedure as the one described for UDP. However, the inclusion of the checksum in the UDP datagram is optional, whereas the inclusion of the checksum for TCP is mandatory.

Urgent pointer: This l6-bit field, which is valid, only if the urgent flag is set, is used when the segment contains urgent data. It defines the number that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment.

Options: There can be up to 40 bytes of optional information in the TCP header.

Congestion Control and Quality of Service

Congestion control and quality of service are two issues so closely bound together that improving one means improving the other and ignoring one usually means ignoring the other. Most techniques to prevent or eliminate congestion also improve the quality of service in a network.

Congestion

An important issue in a packet-switched network is **congestion.**

Congestion in a network may occur if the **load** on the network-the number of packets sent to the network-is greater than the capacity of the network-the number of packets a network can handle.

Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity.

QUALITY OF SERVICE

Flow Characteristics

Traditionally, four types of characteristics are attributed to a flow: reliability, delay, jitter, and bandwidth.

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. For example, it is more important that electronic mail, file transfer, and Internet access have reliable transmissions than telephony or audio conferencing.

Delay: Source-to-destination delay is another flow characteristic. Again applications can tolerate delay in different degrees. In this case, telephony, audio conferencing, video conferencing, and remote log-in need minimum delay, while delay in file transfer or e-mail is less important.

Jitter: Jitter is the variation in delay for packets belonging to the same flow. 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. 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.

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.

Application Layer: DNS

The application layer consists of various applications. Out of those one is DNS, which stands for **Domain Name System**. The very first question arise: 'what is the need of this application?'.

To begin with let's start with a real world example. There are many identifiers to be a unique person in the world, such as SSN, name, and Passport number along with the county who issued it, etc. In the similar fashion, every computer or host and router in the world has a unique identifying 32-bit 'IP' address. Say if we need some information that is on other part of the world. We need to know the IP address of that machine.

Remembering IP addresses is difficult, as it contains all numbers. To remember IP addresses of more than one host becomes cumbersome. Therefore a name has been assigned to almost every IP address which makes it easier for humans to remember.

DNS provides mapping of **IP address and Domain name.**

DNS Services

1. Host name to IP address translation

The primary purpose of DNS is to provide translation of host name to IP address and vice versa. The backward facility (translating IP address to domain name) is known as Reverse DNS.

2. Host aliasing

Host aliasing is referred to another name given to the same machine on the network. It is used because a hostname may have a complicated name instead of that a simple term may be used.

3. Mail server aliasing

It is highly desirable that an email address should contain simple letters, or should be something that can be easy to remember. E.g. richard@gmail.com can be remembered easily but if the original mail server address, say la4.mail1.google.com, were to be used it would be difficult to remember.

4. Load distribution

A set of IP address is provided to one canonical name which prevents the load to be present only on one server. “When the request comes to the DNS server to resolve the domain name, it gives out one of the several canonical names in a rotated order. This redirects the request to one of the several servers in a server group. Once the BIND feature of DNS resolves the domain to one of the servers, subsequent requests from the same client is sent to the same server.”

Problems that arise when we try to centralize DNS:

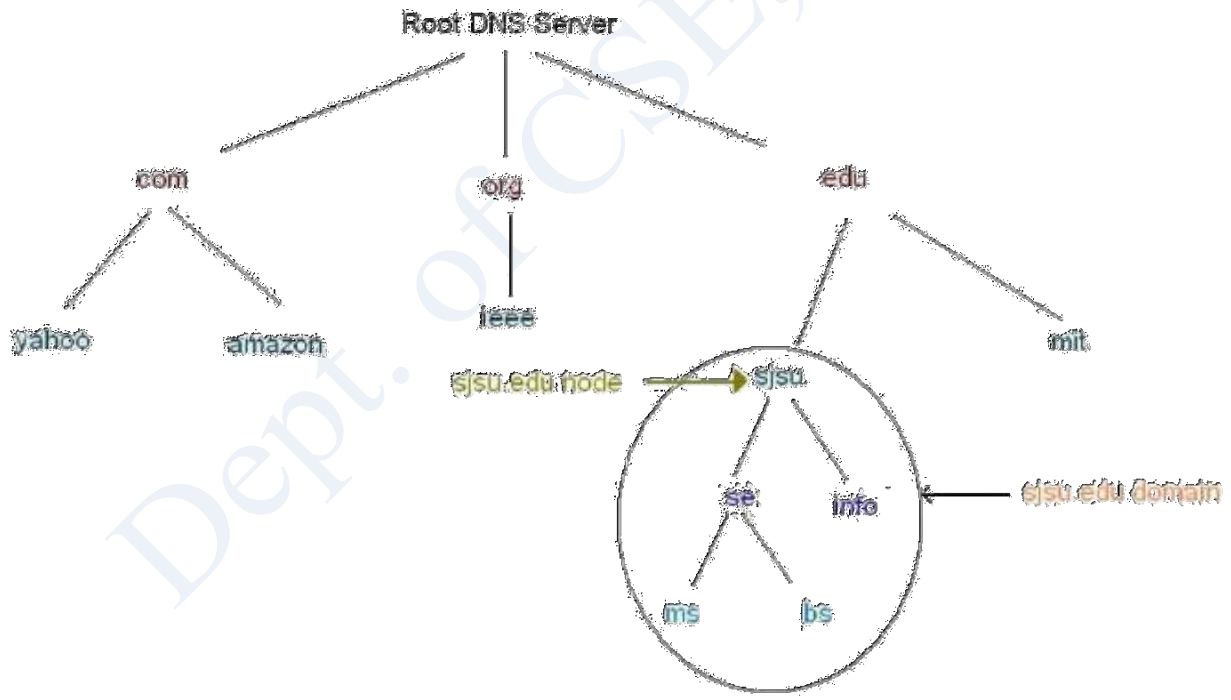
1. Single point of failure

2. Increase in traffic volume

3. Distant centralized database

4. Maintenance

As centralized DNS does not scale because of the reasons mentioned above, a need arose to implement DNS in a distributed manner. The DNS is a distributed system, implemented in a hierarchy of many name servers. The decentralized administration is achieved through delegation.



A domain may contain many sub-domains inside it. To identify if domain is a sub-domain of another domain, you need to compare the domain name with its parent domain name. E.g. se.sjsu.edu is a sub-

domain of the sjsu.edu domain. The other way to determine the sub-domains is through looking at the levels of the tree.

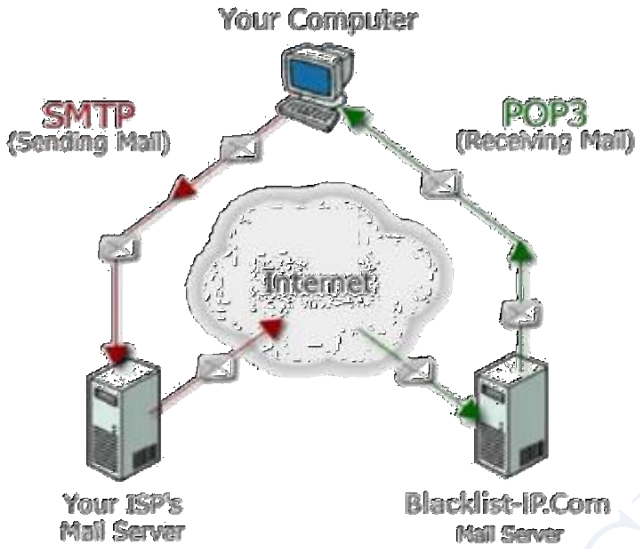
Simple Mail Transfer Protocol (SMTP)

Simple Mail Transfer Protocol (**SMTP**) is an Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks.

SMTP is a connection-oriented, text-based protocol in which a mail sender communicates with a mail receiver by issuing command strings and supplying necessary data over a reliable ordered data stream channel, typically a Transmission Control Protocol (TCP) connection. An SMTP session consists of commands originated by an SMTP client (the initiating agent, sender, or transmitter) and corresponding responses from the SMTP server (the listening agent, or receiver) so that the session is opened, and session parameters are exchanged. A session may include zero or more SMTP transactions. An SMTP transaction consists of three command/reply sequences (see example below.) They are:

1. **MAIL** command, to establish the return address, a.k.a. Return-Path, 5321.From, mfrom, or envelope sender. This is the address for bounce messages.
2. **RCPT** command, to establish a recipient of this message. This command can be issued multiple times, one for each recipient. These addresses are also part of the envelope.
3. **DATA** to send the message text. This is the content of the message, as opposed to its envelope. It consists of a message header and a message body separated by an empty line. DATA is

   actually a group of commands, and the server replies twice: once to the DATA command proper, to acknowledge that it is ready to receive the text, and the second time after the end-of-data sequence, to either accept or reject the entire message.



FTP

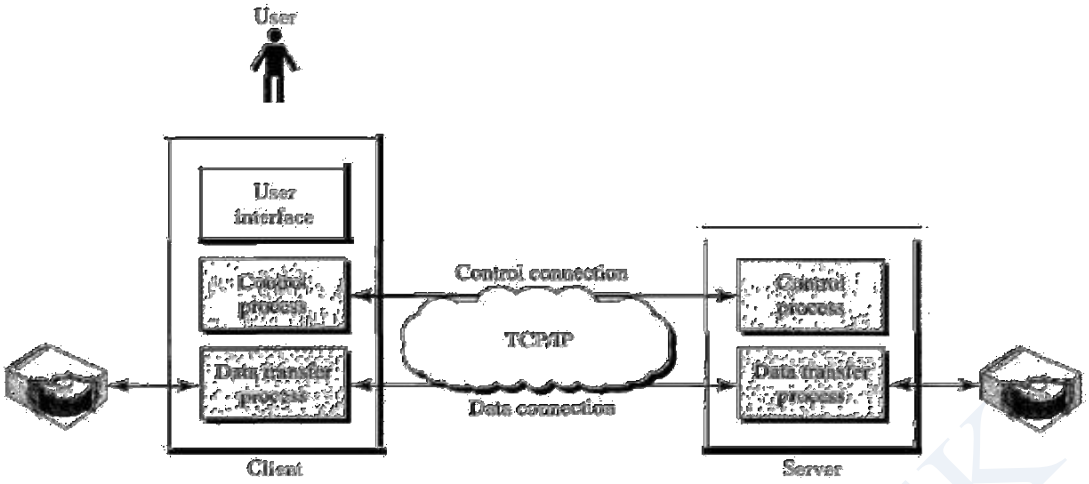
File Transfer Protocol (FTP)

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.



WWW

The **World Wide Web** (WWW) is a repository of information linked together from points all over the world. The WWW has a unique combination of flexibility, portability, and user-friendly features that distinguish it from other services provided by the Internet.

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.

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. The client protocol can be one of the protocols described previously such as FTP or HTTP (described later in the chapter). The interpreter can be HTML, Java, or JavaScript, depending on the type of document.

.

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. A server can also become more efficient through multithreading or multiprocessing. In this case, a server can answer more than one request at a time.

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.

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.

Protocol://host:port/path

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.

This page is left vacant intentionally