6/3/2014

COMPUTER NETWORKS

CSE-2011 B.Sc. (Honours) Question Solution

S M TALHA JUBAED
DEPT. OF COMPUTER SCIENCE & ENGINEERING (CSE)
UNIVERSITY OF RAJSHAHI (RU)
HOTLINE# +088-01911-088 706

Question-01: What do you mean by computer network and distributed system? 3 Marks CSE-2011 CSE-2006

Computer Network:

A computer network or data network is a telecommunications network that allows computers to exchange data. In other words, computer networks means a collection of "autonomous" computers interconnected by a single technology.

In computer networks, networked computing devices pass data to each other along data connections. The connections (network links) between nodes are established using either cable media or wireless media. The best-known computer network is the Internet.

Network computer devices that originate, route and terminate the data are called network nodes. Nodes can include hosts such as personal computers, phones, servers as well as networking hardware. Two such devices are said to be networked together when one device is able to exchange information with the other device, whether or not they have a direct connection to each other.

Computer networks support applications such as access to the World Wide Web, shared use of application and storage servers, printers, and fax machines, and use of email and instant messaging applications. Computer networks differ in the physical media used to transmit their signals, the communications protocols to organize network traffic, the network's size, topology and organizational intent.

Examples of Computer Network:

1. Ethernet
2. The Internet
3. Connection-Oriented Networks: X.25, Frame Relay, and ATM
4. Wireless LANs: 802:11

Distributed system:

In a distributed system, a collection of independent computers appears to its users as a single coherent system. Usually, it has a single model or paradigm that it presents to the users. Often a layer of software on top of the operating system, called middleware, is responsible for implementing this model.

A well-known example of a distributed system is the World Wide Web, in which everything looks like a document (Web page).

Question: Define Network Architecture. 2 Marks CSE-2011

Network Architecture:

A set of layers and protocols is called a network architecture. To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. The purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it. Basically, **a protocol is an agreement between the communicating parties on how communication is to proceed.**

The specification of an architecture must contain enough information to allow an implementer to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol. Neither the details of the implementation nor the specification of the interfaces is part of the architecture because these are hidden away inside the machines and not visible from the outside. It is not even necessary that the interfaces on all machines in a network be the same, provided that each machine can correctly use all the protocols. A list of protocols used by a certain system, one protocol per layer, is called a protocol stack.

Question: Define Physical Address, Logical Address, Port Address, and application-specific address. 3 Marks CSE-2011

Physical Address:

The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN. It is included in the frame used by the data link layer. It is the lowest-level address. The physical addresses have authority over the link (LAN or WAN). The size and format of these addresses vary depending on the network. For example, Ethernet uses a 6-byte (48-bit) physical address that is imprinted on the network interface card (NIC). LocalTalk (Apple), however, has a 1-byte dynamic address that changes each time the station comes up. **The physical addresses will change from hop to hop.**

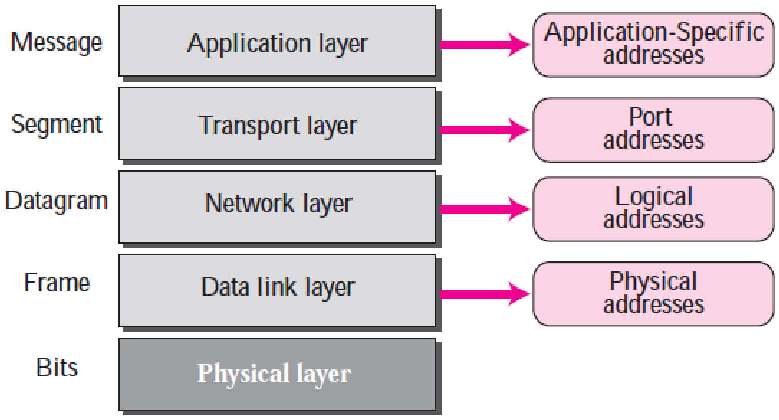
Physical addresses can be either unicast (one single recipient), multicast (a group of recipients), or broadcast (to be received by all systems in the network). Some networks support all three addresses. For example, Ethernet supports the unicast physical addresses (6 bytes), the multicast addresses, and the broadcast addresses. Some networks do not support the multicast or broadcast physical addresses. If a frame must be sent to a group of recipients or to all systems, the multicast or broadcast address must be simulated using unicast addresses. This means that multiple packets are sent out using unicast addresses.

Example of Physical Address:

Most local area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below:

07:01:02:01:2C:4B
A 6-byte (12 hexadecimal digits) physical address

Figure 2.15 *Addresses in the TCP/IP Protocol Suite*



Logical Addresses:

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

The logical addresses can be either unicast (one single recipient), multicast (a group of recipients), or broadcast (all systems in the network). There are limitations on broadcast addresses.

Port Addresses:

The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. A system that sends nothing but data from one computer to another is not complete. Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process. For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes. In other words, they need addresses. In the TCP/IP architecture, the label assigned to a process is called a port address. A port address in TCP/IP is 16 bits in length. **From hop to hop, the port addresses remain the same.**

Example of port address:

A port address is a 16-bit address represented by one decimal number as shown.

753

A 16-bit port address represented as one single number

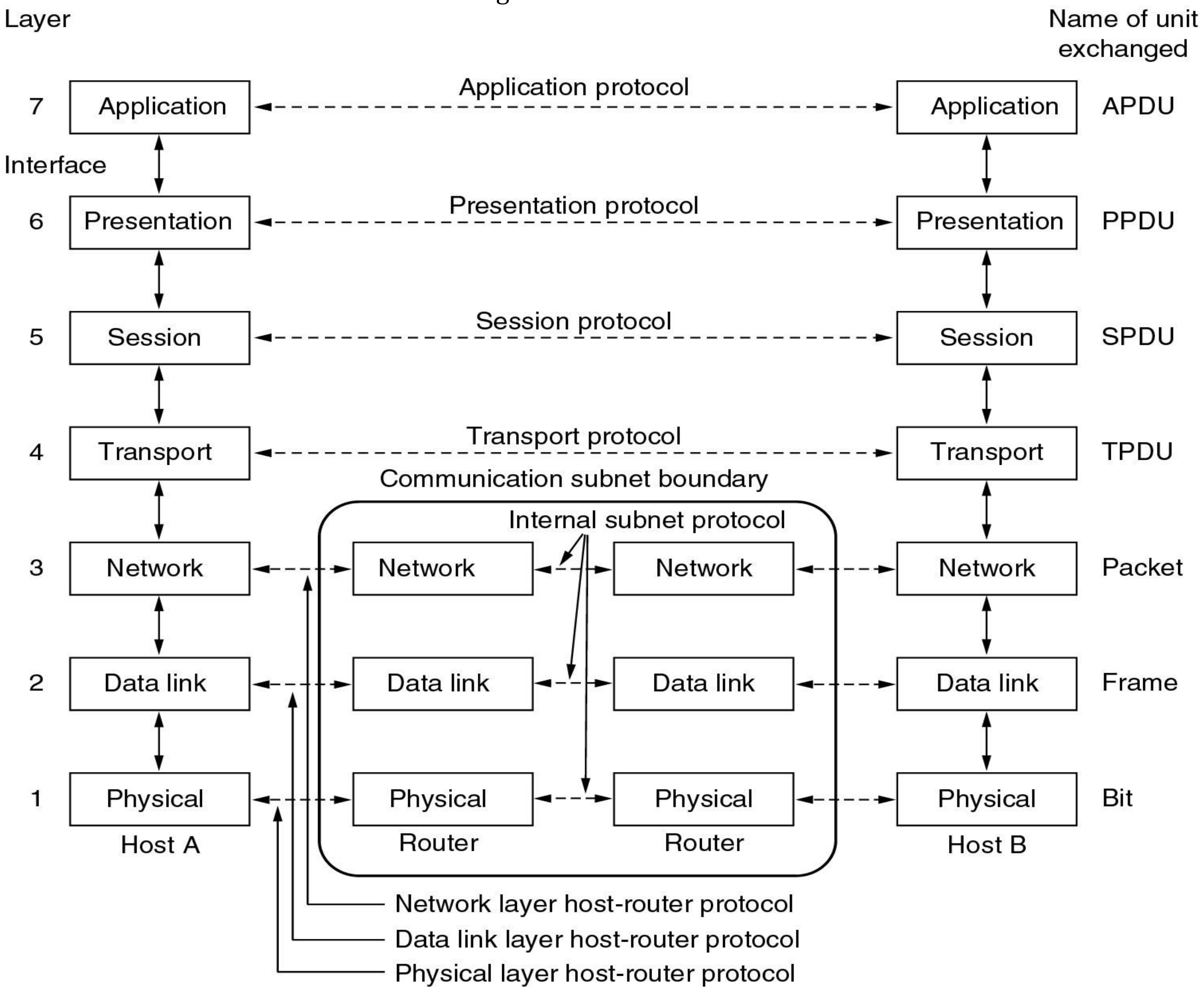
Application-Specific Addresses

Some applications have user-friendly addresses that are designed for that specific application. Examples include the e-mail address (for example, forouzan@fhda.edu) and the Universal Resource Locator (URL) (for example, www.mhhe.com). The first defines the recipient of an e-mail; the second is used to find a document on the World Wide Web. These addresses, however, get changed to the corresponding port and logical addresses by the sending computer.

Question-04: Draw the architecture of OSI reference Model. 6 Marks CSE-2011

Architecture of OSI reference Model

The architecture of OSI reference model is given below:



Question-04: Describe TCP/IP reference model. 6 Marks CSE-2011

TCP/IP reference model:

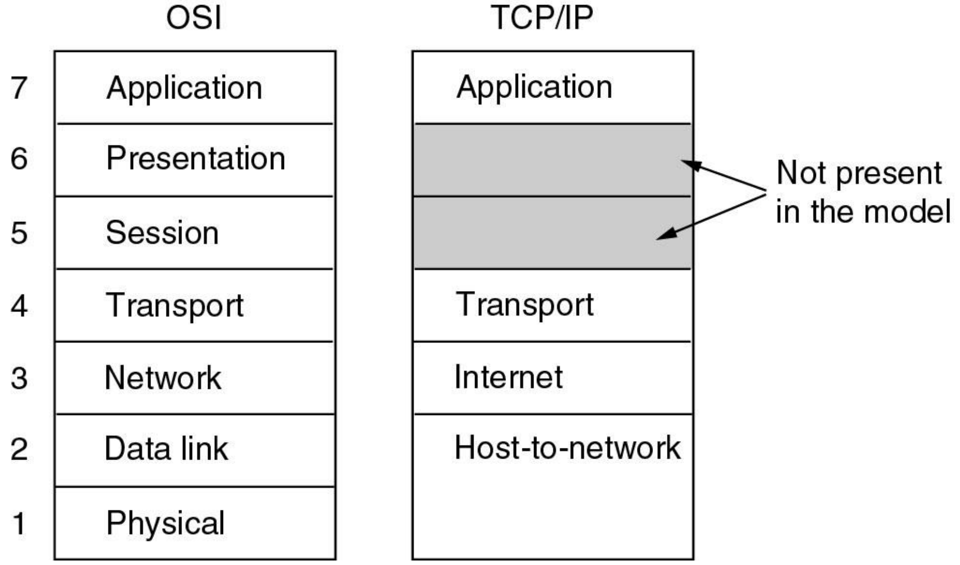
TCP/IP reference model is composed of four layers: Internet Layer, Transport Layer, Application Layer and Host-To-Host Layer. These four layers are described below:

The Internet Layer

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

The analogy here is with the (snail) mail system. A person can drop a sequence of international letters into a mail box in one country, and with a little luck, most of them will be delivered to the correct address in the destination country. Probably the letters will travel through one or more international mail gateways along the way, but this is transparent to the users. Furthermore, that each country (i.e., each network) has its own stamps, preferred envelope sizes, and delivery rules is hidden from the users.

The internet layer defines an official packet format and protocol called IP (Internet Protocol). The job of the internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly the major issue here, as is avoiding congestion. For these reasons, it is reasonable to say that the TCP/IP internet layer is similar in functionality to the OSI network layer. Figure 1-21 shows this correspondence.



The Transport Layer

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

The second protocol in this layer, UDP (User Datagram Protocol), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP, and UDP is shown in **Fig. 1-22.** Since the model was developed, IP has been implemented on many other networks.

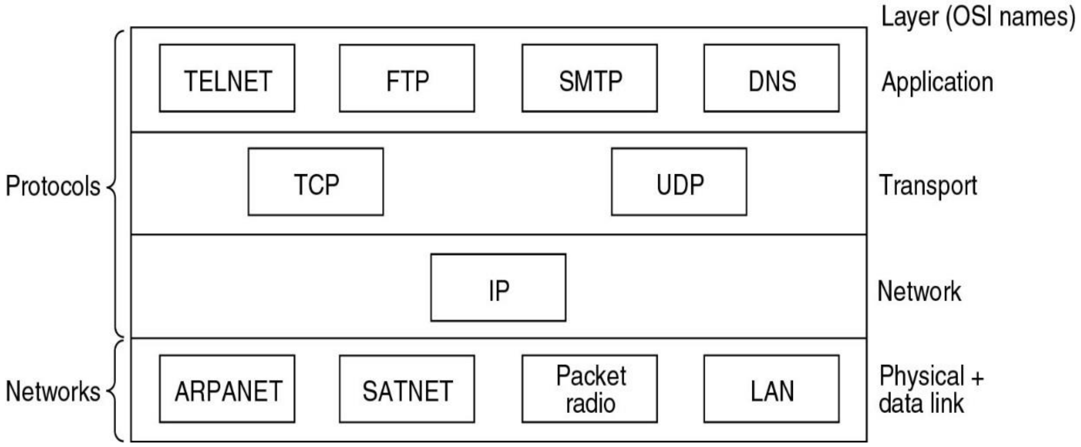


Fig. 1-22 Protocols and networks in the TCP/IP model initially.

The Application Layer

The TCP/IP model does not have session or presentation layers. No need for them was perceived, so they were not included. Experience with the OSI model has proven this view correct: they are of little use to most applications.

On top of the transport layer is the application layer. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP), as shown in Fig. 1-22. The virtual terminal protocol allows a user on one machine to log onto a distant machine and work there. The file transfer protocol provides a way to move data efficiently from one machine to another. Electronic mail was originally just a kind of file transfer, but later a specialized protocol (SMTP) was developed for it. Many other protocols have been added to these over the years: the Domain Name System (DNS) for mapping host names onto their network addresses, NNTP, the protocol for moving USENET news articles around, and HTTP, the protocol for fetching pages on the World Wide Web, and many others.

The Host-to-Network Layer

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

Question-06: Describe Circuit Switching network with example. 4 Marks CSE-2011

Circuit Switching network:

The passage of a message from a source to a destination involves many decisions. When a message reaches a connecting device, a decision needs to be made to select one of the output ports through which the packet needs to be send out. In other words, the connecting device acts as a switch that connects one port to another port. This selection process of one of the output ports through which the packet needs to be send out is called switching. **One solution to the switching is referred to as circuit switching, in which a physical circuit (or channel) is established between the source and destination of the message before the delivery of the message. After the circuit is established, the entire message, is transformed from the source to the destination. The source can then inform the network that the transmission is complete, which allows the network to open all switches and use the links and connecting devices for another connection.** The circuit switching was never implemented at the network layer; it is mostly used at the physical layer. **The network that uses circuit switching technique is called Circuit switched network.**

Circuit switching networks require a circuit to be established. The circuit functions as if the nodes were physically connected as with an electrical circuit. In circuit switching, network links are dedicated to one communication session at a time, guarantees the quality of service. In circuit switching, the bit delay is constant during a connection. While circuit switching is commonly used for connecting voice circuits, the concept of a dedicated path persisting between two communicating parties or nodes can be extended to signal content other than voice. Its advantage is that it provides for continuous transfer without the overhead associated with packets making maximal use of available bandwidth for that communication. No circuit can be degraded by competing users because it is protected from use by other callers until the circuit is released and a new connection is set up. Even if no actual communication is taking place, the channel remains reserved and protected from competing users.

Basically, Circuit switching is a methodology of implementing a telecommunications network in which two network nodes establish a dedicated communications channel (circuit) through the network before the nodes may communicate. The circuit guarantees the full bandwidth of the channel and remains connected for the duration of the communication session. The circuit functions as if the nodes were physically connected as with an electrical circuit. In circuit switching, the whole message is sent from the source to the destination without being divided into packets. The circuit switching was never implemented at the network layer; it is mostly used at the physical layer.

Circuit switching is old and expensive, and it is what PSTN uses. Circuit switching can be relatively inefficient because capacity is guaranteed on connections which are set up but are not in continuous use, but rather momentarily. However, unused capacity guaranteed to a connection cannot be used by other connections on the same network.

Examples of circuit-switched networks:

1. The defining example of a circuit-switched network is the early analog telephone network
2. Public switched telephone network (PSTN)
3. ISDN B-channel
4. Circuit Switched Data (CSD) and High-Speed Circuit-Switched Data (HSCSD) service in cellular systems such as GSM
5. Datakit
6. X.21 (Used in the German DATEX-L and Scandinavian DATEX circuit switched data network)
7. Optical mesh network

Circuit-switched networks এর উধাহরন নিম্নভাবেও দেওয়া যেতে পারেঃ

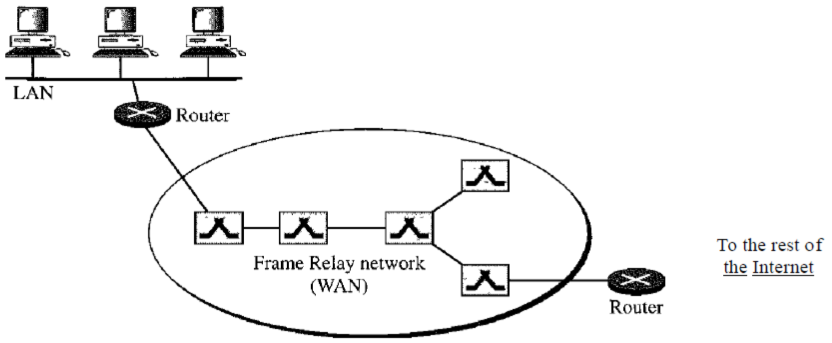
Example:

A good example of a circuit-switched network is the early telephone systems in which the path was established between a caller and a callee when the telephone number of the callee was dialled by the caller. When the callee responded to the call, the circuit was established. The voice message could now flow between the two parties, in both directions, while all of the connecting devices maintained the circuit. When the caller or callee hung up, the circuit was disconnected. The telephone network is not totally a circuit-switched network today.

Question: Draw the architecture of frame relay network. 4 Marks CSE-2011

Architecture of frame relay network:

Figure 18.1 *Frame Relay network*

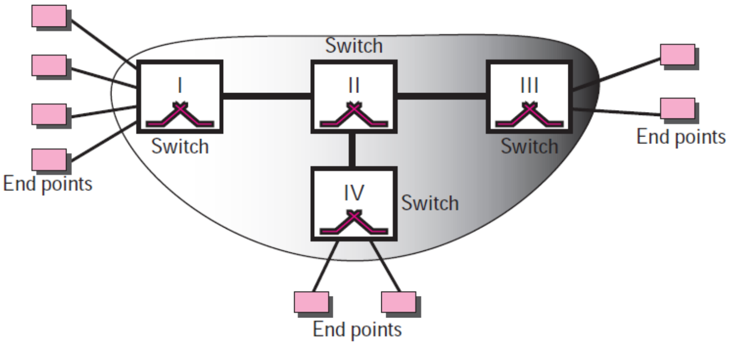


Question: Draw the architecture of ATM Network. 5 Marks CSE-2011

Architecture of ATM Network:

ATM is a switched network. The user access devices, called the end points, are connected to the switches inside the network. The switches are connected to each other using high-speed communication channels. Figure 3.33 shows an example of an ATM network.

Figure 3.33 *Architecture of an ATM network*



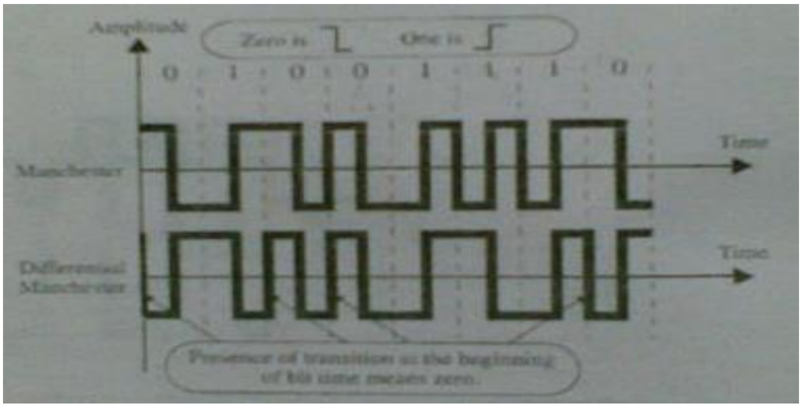
Question: Define Manchester and Differential Manchester encoding techniques. 3 Marks CSE-2011

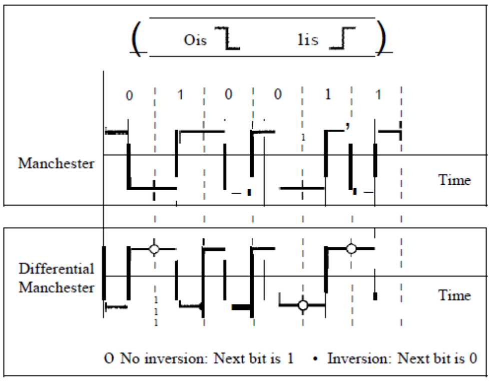
Manchester and Differential Manchester Encoding Techniques:

Manchester encoding technique is a polar biphase line coding scheme in which the idea of Return-to-Zero (transition at the middle of the bit) and the idea of Not-Return_to_Zero-Level (NRZ-L) are combined. In Manchester encoding, the duration of the bit is divided into two halves. The voltage remains at one level during the first half and moves to the other level in the second half. The transition at the middle of the bit provides synchronization.

Differential Manchester encoding technique is a polar biphase line coding scheme in which the idea of Return-to-Zero (transition at the middle of the bit) and the idea of Not-Return_to_Zero-Invert (NRZ-I) are combined. In differential Manchester encoding technique, there is always a transition at the middle of the bit, but the bit values are determined at the beginning of the bit. If the next bit is 0, there is a transition; if the next bit is 1, there is none.

Figure 4.8 shows both Manchester and differential Manchester encoding.





দুটি ফিগার একই । অস্বচ্ছতার জন্য এখানে দুটি ফিগার দেওয়া হলো । পরিচ্ছন্ন ছবি, নিজে একে নিন ।

Question: Describe the control field format of I-Frame, S-frame and U-Frame. 4 Marks CSE-2011

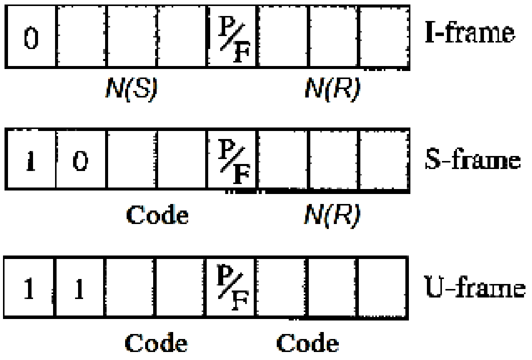
Control Field

The control field determines the type of frame and defines its functionality. So let us discuss the format of this field in greater detail. The format is specific for the type of frame, as shown in Figure 11.28.

Control Field for I-Frames

I-frames are designed to carry user data from the network layer. In addition, they can include flow and error control information (piggybacking). The subfields in the control field are used to define these functions. The first bit defines the type. If the first bit of the control field is 0, this means the frame is an I-frame. The next 3 bits, called N(S), define the sequence number of the frame. Note that with 3 bits, we can define a sequence number between and 7; but in the extension format, in which the control field is 2 bytes, this field is larger. The last 3 bits, called N(R), correspond to the acknowledgment number when piggybacking is used. The single bit between N(S) and N(R) is called the P/F bit. The P/F field is a single bit with a dual purpose. It has meaning only when it is set (bit = 1) and can mean poll or final. It means poll when the frame is sent by a primary station to a secondary (when the address field contains the address of the receiver). It means final when the frame is sent by a secondary to a primary (when the address field contains the address of the sender).

Figure 11.28 *Control field format for the different frame types*



<u>**Control field format of S-frame:**</u>

Supervisory frames are used for flow and error control whenever piggybacking is either impossible or inappropriate (e.g., when the station either has no data of its own to send or needs to send a command or response other than an acknowledgment). S-frames do not have information fields. If the first 2 bits of the control field is 10, this means the frame is an S-frame. The last 3 bits, called N(R), corresponds to the acknowledgment number (ACK) or negative acknowledgment number (NAK) depending on the type of S-frame. The 2 bits called code is used to define the type of S-frame itself. With 2 bits, we can have four types of S-frames, as described below:

- **Receive ready (RR).** If the value of the code subfield is 00, it is an RR S-frame. This kind of frame acknowledges the receipt of a safe and sound frame or group of frames. In this case, the value N(R) field defines the acknowledgment number.

- **Receive not ready (RNR).** If the value of the code subfield is 10, it is an RNR S-frame. This kind of frame is an RR frame with additional functions. It acknowledges the receipt of a frame or group of frames, and it announces that the receiver is busy and cannot receive more frames. It acts as a kind of congestion control mechanism by asking the sender to slow down. The value of NCR) is the acknowledgment number.

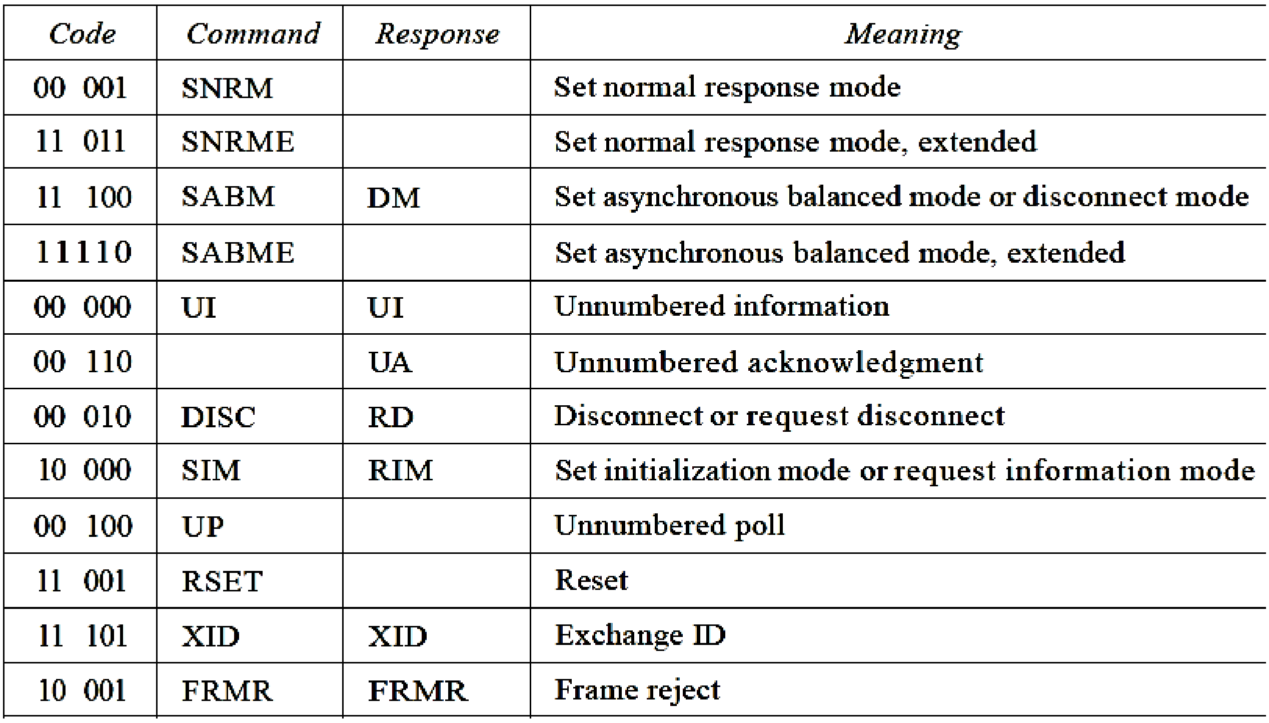
- **Reject (REJ).** If the value of the code subfield is 01, it is a REJ S-frame. This is a NAK frame, but not like the one used for Selective Repeat ARQ. It is a NAK that can be used in Go-Back-N ARQ to improve the efficiency of the process by informing the sender, before the sender time expires, that the last frame is lost or damaged. The value of NCR) is the negative acknowledgment number.

- **Selective reject (SREJ).** If the value of the code subfield is 11, it is an SREJ S-frame. This is a NAK frame used in Selective Repeat ARQ. The HDLC Protocol uses the term selective reject instead of selective repeat. The value of N(R) is the negative acknowledgment number.

Control Field for U-Frames:

Unnumbered frames are used to exchange session management and control information between connected devices. Unlike S-frames, U-frames contain an information field, but one used for system management information, not user data. As with S-frames, however, much of the information carried by U-frames is contained in codes included in the control field. U-frame codes are divided into two sections: a 2-bit prefix before the PtF bit and a 3-bit suffix after the P/F bit. Together, these two segments (5 bits) can be used to create up to 32 different types of U-frames. Some of the more common types are shown in Table 11.1.

Table 11.1 *U-frame control command and response*



Code	*Command*	*Response*	*Meaning*
00 001	SNRM		Set normal response mode
11 011	SNRME		Set normal response mode, extended
11 100	SABM	DM	Set asynchronous balanced mode or disconnect mode
11110	SABME		Set asynchronous balanced mode, extended
00 000	UI	UI	Unnumbered information
00 110		UA	Unnumbered acknowledgment
00 010	DISC	RD	Disconnect or request disconnect
10 000	SIM	RIM	Set initialization mode or request information mode
00 100	UP		Unnumbered poll
11 001	RSET		Reset
11 101	XID	XID	Exchange ID
10 001	FRMR	FRMR	Frame reject

Question: What is Flow Control? Why it is necessary? 4 Marks CSE-2011

Flow control:

If the rate at which the data is absorbed by the receiver is less than the rate produced at the sender, the data link layer imposes a mechanism to prevent overwhelming the receiver. This mechanism is called flow control. Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

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

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

<u>Why it is necessary:</u>

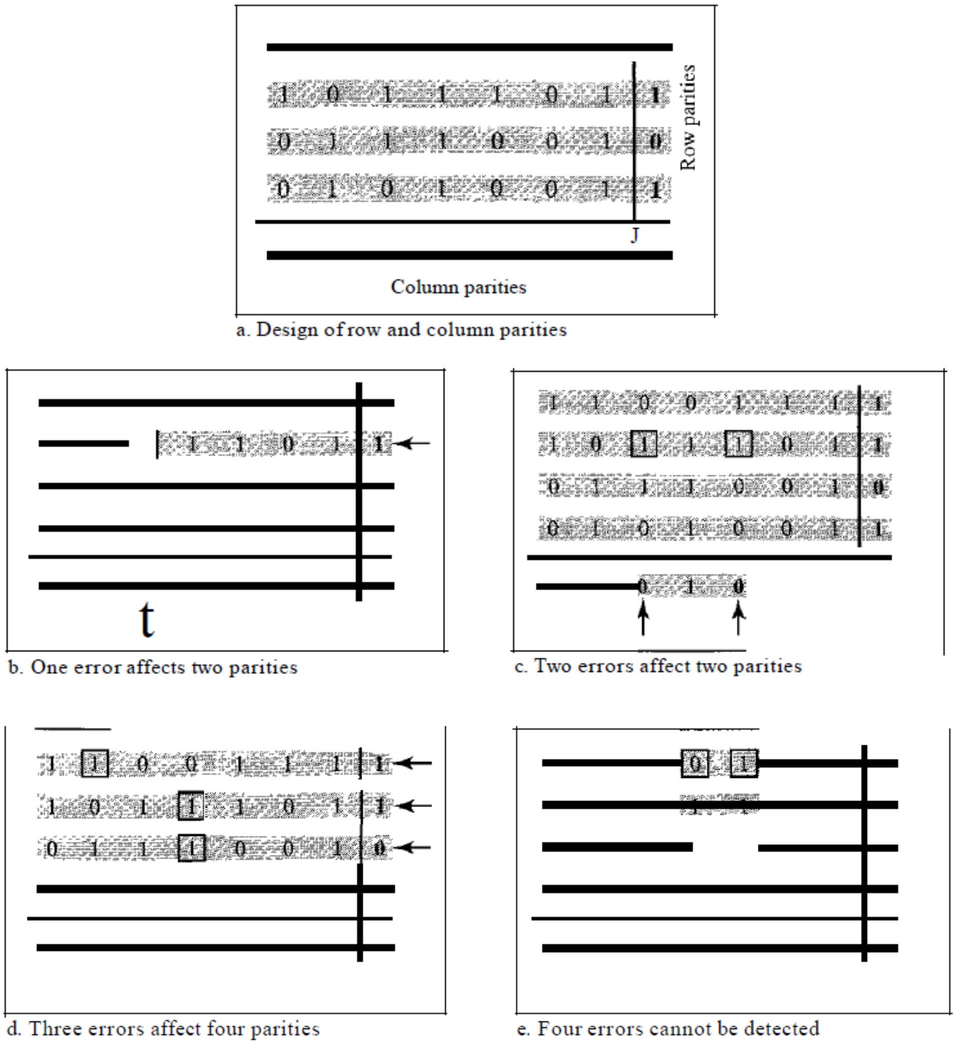
Flow control is necessary when data is being sent faster than it can be processed by receiver. When a processing program is limited in capacity, data transfer from one computer to another computer should be controlled. Flow control is needed since the sending entity should not overwhelm the receiving entity. Recipient needs some time to process incoming packets. If sender sends faster than recipient processes, then buffer overflow occurs. Flow control prevents buffer overflow.

Question: Explain the error checking process of 2D parity checking technique with an example. 5 Marks CSE-2011

<u>Error checking process of 2D parity checking technique:</u>

2D Parity checking technique is an error-detecting code. In two-dimensional parity check, the dataword is organized in a table (rows and columns). In Figure 10.11, the data to be sent, five 7-bit bytes, are put in separate rows. For each row and each column, 1 parity-check bit is calculated. The whole table is then sent to the receiver, which finds the syndrome for each row and each column. As Figure 10.11 shows, the two-dimensional parity check can detect up to three errors that occur anywhere in the table (arrows point to the locations of the created nonzero syndromes). However, errors affecting 4 bits may not be detected.

Figure 10.11 *Two-dimensional parity-check code*



a. Design of row and column parities

b. One error affects two parities

c. Two errors affect two parities

d. Three errors affect four parities

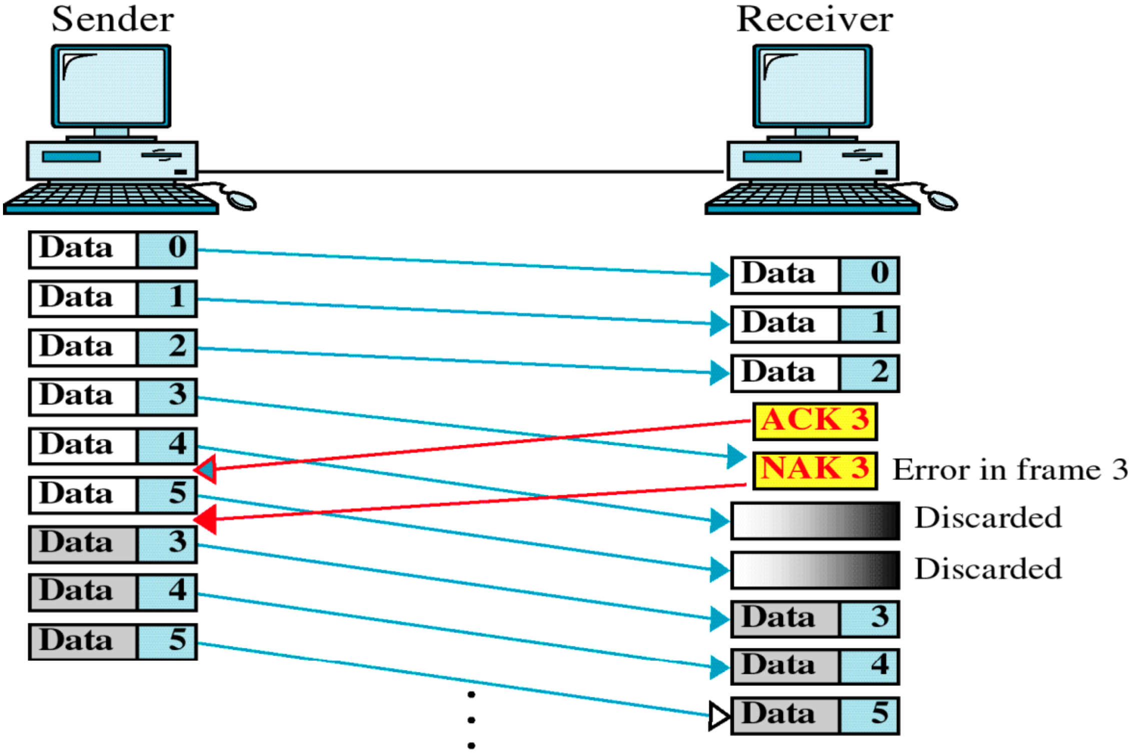
e. Four errors cannot be detected

Question: Describe the delayed acknowledgement operation of Go-Back-N ARQ. 5 Marks CSE-2012

<u>Delayed acknowledgement operation of Go-Back-N ARQ</u>

Although there can be a timer for each frame that is sent, in Go-Back-N ARQ protocol we use only one. The reason is that the timer for the first outstanding frame always expires first; we send all outstanding frames when this timer expires. The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order. If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting. The silence of the receiver causes the timer of the unacknowledged frame at the sender site to expire.

This, in turn, causes the sender to go back and resend all frames, beginning with the one with the expired timer. The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgment for several frames. When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3, 4,5, and 6 again. That is why the protocol is called Go-Back-N ARQ.



Question: What are the drawbacks of STOP-AND-WAIT ARQ? 2 Marks CSE-2011

Drawbacks of STOP-AND-WAIT ARQ:

1. **Inefficient:** The Stop-and-Wait ARQ is very inefficient if our channel is thick and long. By thick, we mean that our channel has a large bandwidth; by long, we mean the round-trip delay is long. The product of these two is called the bandwidth delay product. We can think of the channel as a pipe. The bandwidth-delay product then is the volume of the pipe in bits. The pipe is always there. If we do not use it, we are inefficient. The bandwidth-delay product is a measure of the number of bits we can send out of our system while waiting for news from the receiver.
2. Stop-and-wait ARQ is inefficient compared to other ARQs, because the time between packets, if the ACK and the data are received successfully, is twice the transit time (assuming the turnaround time can be zero).
3. The problem of "Stop and Wait ARQ" is not be able to send multiple frames.

Question: Define two procedures, one for sender and another for receiver to allow unidirectional dataflow over an unreliable channel. 5 Marks CSE-2011

Question: What is Piggybacking? Explain in brief. What are the advantages and disadvantages of piggybacking? 4 Marks CSE-2011 CSE-2006 CSE-2005

What is piggybacking?

In all practical situations, the transmission of data needs to be bi-directional. This is called as full-duplex transmission. We can achieve this full duplex transmission *i.e.* by having two separate channels-one for forward data transfer and the other for separate transfer *i.e.* for acknowledgements. A better solution would be to use each channel (forward & reverse) to transmit frames both ways, with both channels having the same capacity. If A and B are two users. Then the data frames from A to Bare intermixed with the acknowledgements from A to B. One more improvement that can be made is piggybacking. The concept is explained as follows:

In two way communication, whenever a data frame is received, the received waits and does not send the control frame (acknowledgement) back to the sender immediately. The receiver waits until its network layer passes in the next data packet. The delayed acknowledgement is then attached to this outgoing data frame. This technique of temporarily delaying the acknowledgement so that it can be hooked with next outgoing data frame is known as piggybacking.

<u>Advantages of piggybacking:</u>

1. The major advantage of piggybacking is better use of available channel bandwidth.
2. It improves the efficiency of the bidirectional protocols.

- **<u>The disadvantages of piggybacking are:</u>**

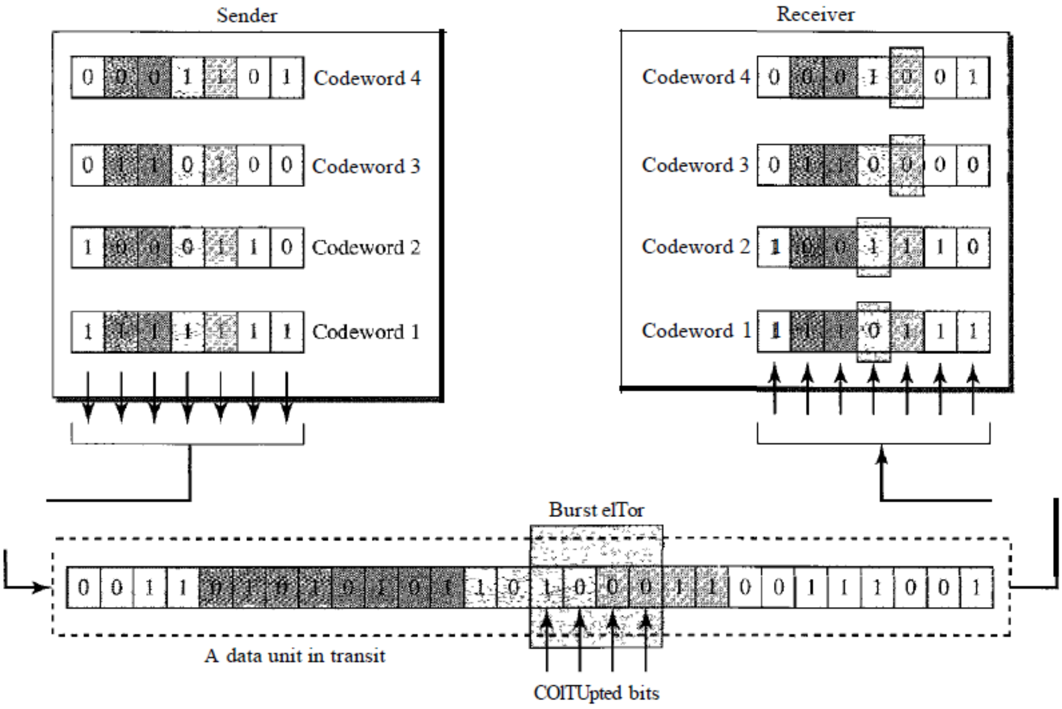
1. Additional complexity.
2. If the data link layer waits too long before transmitting the acknowledgement, then retransmission of frame would take place.

Question: Explain how to use Hamming code to correct burst errors. 4 Marks CSE-2011

<u>Hamming code to correct burst errors:</u>

A Hamming code can only correct a single error or detect a double error. However, there is a way to make it detect a burst error, as shown in Figure 10.13. The key is to split a burst error between several codewords, one error for each codeword. In data communications, we normally send a packet or a frame of data. To make the Hamming code respond to a burst error of size N, we need to make N codewords out of our frame. Then, instead of sending one codeword at a time, we arrange the codewords in a table and send the bits in the table a column at a time. In Figure 10.13, the bits are sent column by column (from the left). In each column, the bits are sent from the bottom to the top. In this way, a frame is made out of the four codewords and sent to the receiver. Figure 10.13 shows that when a burst error of size 4 corrupts the frame, only 1 bit from each codeword is corrupted. The corrupted bit in each codeword can then easily be corrected at the receiver.

Figure 10.13 *Burst error correction using Hamming code*

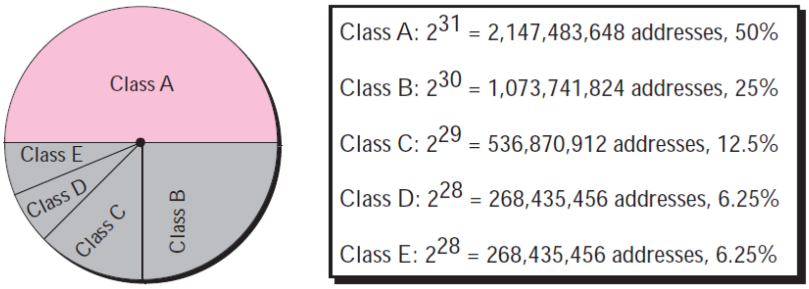


Question: Define and explain in brief Classful addressing and Classless addressing. 4+4 Marks CSE-2011

Classful addressing:

IPv4 addressing, at its inception, used the concept of classes. This architecture is called classful addressing where the IP address space is divided into five **classes: A, B, C, D,** and **E.** Each class occupies some part of the whole address space. Figure 5.5 shows the class occupation of the address space.

Figure 5.5 *Occupation of the address space*



Class A: 2^{31} = 2,147,483,648 addresses, 50%

Class B: 2^{30} = 1,073,741,824 addresses, 25%

Class C: 2^{29} = 536,870,912 addresses, 12.5%

Class D: 2^{28} = 268,435,456 addresses, 6.25%

Class E: 2^{28} = 268,435,456 addresses, 6.25%

Recognizing Classes

We can find the class of an address when the address is given either in binary or dotted decimal notation. In the binary notation, the first few bits can immediately tell us the class of the address; in the dotted-decimal notation, the value of the first byte can give the class of an address (Figure 5.6).

Figure 5.6 *Finding the class of an address*

	Octet 1	Octet 2	Octet 3	Octet 4
Class A	0........			
Class B	10......			
Class C	110.....			
Class D	1110....			
Class E	1111....			

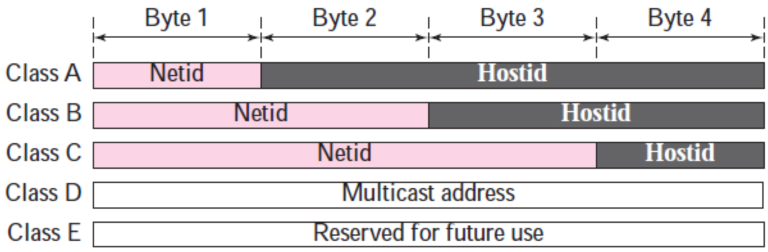
Binary notation

	Byte 1	Byte 2	Byte 3	Byte 4
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–299			
Class E	240–255			

Dotted-decimal notation

In classful addressing, an IP address in classes A, B, and C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address. Figure 5.8 shows the netid and hostid bytes. Classes D and E are not divided into netid and hosted. In class A, 1 byte defines the netid and 3 bytes define the hostid. In class B, 2 bytes define the netid and 2 bytes define the hostid. In class C, 3 bytes define the netid and 1 byte defines the hostid.

Figure 5.8 *Netid and hostid*



Classless Addressing:

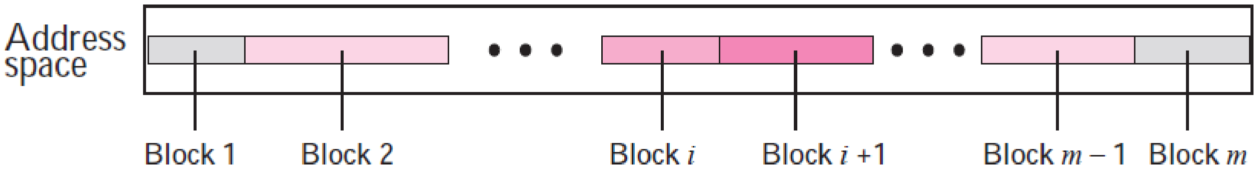
A scheme of IP Addressing to overcome address depletion and give more organizations access to the Internet where there are no classes, but the addresses are still granted in blocks is called Classless IP Addressing. Subnetting and supernetting in classful addressing did not really solve the address depletion problem and made the distribution of addresses and the routing process more difficult. With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. The larger address space, however, requires that the length of IP addresses to be increased, which means the format of the IP packets

needs to be changed. Although the long-range solution has already been devised and is called IPv6, a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization. The short-term solution still uses IPv4 addresses, but it is called classless addressing. In other words, the class privilege was removed from the distribution to compensate for the address depletion.
In classless addressing, variable-length blocks are used that belong to no classes. We can have a block of 1 address, 2 addresses, 4 addresses, 128 addresses, and so on.

Variable-Length Blocks:

In classless addressing, the whole address space is divided into variable length blocks. Theoretically, we can have a block of $2^0, 2^1, 2^2, \ldots, 2^{32}$ addresses. The only restriction, as we discuss later, is that the number of addresses in a block needs to be a power of 2. An organization can be granted one block of addresses. Figure 5.27 shows the division of the whole address space into non-overlapping blocks.

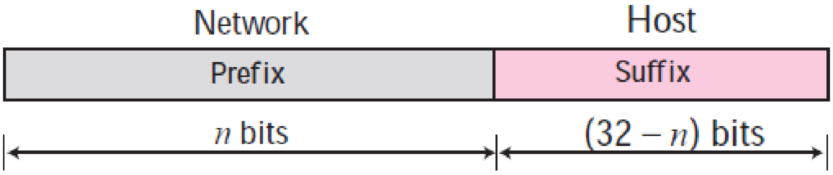
Figure 5.27 *Variable-length blocks in classless addressing*



Two-Level Addressing:

In classless addressing. When an organization is granted a block of addresses, the block is actually divided into two parts, the prefix and the suffix. The prefix plays the same role as the netid; the suffix plays the same role as the hosted used in classfull addressing. All addresses in the block have the same prefix; each address has a different suffix. Figure 5.28 shows the prefix and suffix in a classless block. In classless addressing, the prefix defines the network and the suffix defines the host. In classless addressing, the length of the prefix, n, depends on the size of the block; it can be 0, 1, 2, 3, . . . , 32. In classless addressing, the value of n is referred to as prefix length; the value of 32 − n is referred to as suffix length. The prefix length in classless addressing can be 1 to 32.

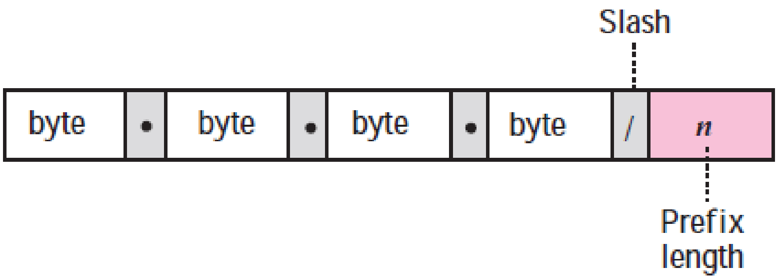
Figure 5.28 *Prefix and suffix*



Slash Notation

The prefix length in classless addressing play a very important role when we need to extract the information about the block from a given address in the block. In classless addressing, the prefix length cannot be found if we are given only an address in the block. The given address can belong to a block with any prefix length. In classless addressing, we need to include the prefix length to each address if we need to find the block of the address. In this case, the prefix length, n, is added to the address separated by a slash. The notation is informally referred to as slash notation. An address in classless addressing can then be represented as shown in Figure 5.29. The slash notation is formally referred to as **classless interdomain routing** or **CIDR** (pronounced cider) notation. **In classless addressing, we need to know one of the addresses in the block and the prefix length to define the block.**

Figure 5.29 *Slash notation*



Network Mask

The idea of network mask in classless addressing is the same as the one in classful addressing. A network mask is a 32-bit number with the n leftmost bits all set to 0s and the rest of the bits all set to 1s.

Extracting Block Information

An address in slash notation (CIDR) contains all information we need about the block: the first address (network address), the number of addresses, and the last address. These three pieces of information can be found as follows:

❑ The number of addresses in the block can be found as:

$$\boldsymbol{N = 2^{32} - n}$$

in which n is the prefix length and N is the number of addresses in the block.

❑ The first address (network address) in the block can be found by ANDing the address with the network mask:

First address = (any address) AND (network mask)

Alternatively, we can keep the n leftmost bits of any address in the block and set the 32− n bits to 0s to find the first address.

❑ The last address in the block can be found by either adding the first address with the number of addresses or, directly, by ORing the address with the complement (NOTing) of the network mask:

Last address = (any address) OR [NOT (network mask)]

Alternatively, we can keep the n leftmost bits of any address in the block and set the 32 − n bits to 1s to find the last address.

Block Allocation

The next issue in classless addressing is block allocation. The ultimate responsibility of block allocation is given to a global authority called the Internet Corporation for Assigned Names and Addresses (ICANN). However, ICANN does not normally allocate addresses to individual Internet users. It assigns a large block of addresses to an ISP (or a larger organization that is considered an ISP in this case). For the proper operation of the CIDR, three restrictions need to be applied to the allocated block.

1. The number of requested addresses, N, needs to be a power of 2. This is needed to provide an integer value for the prefix length, n (see the second restriction). The number of addresses can be 1, 2, 4, 8, 16, and so on.
2. The value of prefix length can be found from the number of addresses in the block. Since N = 232 - n, then n = log2 (232/N) = 32 - log2N. That is the reason why N needs to be a power of 2.
3. The requested block needs to be allocated where there are a contiguous number of unallocated addresses in the address space. However, there is a restriction on choosing the beginning addresses of the block. The beginning address needs to be divisible by the number of addresses in the block. To see this restriction, we can show that the beginning address can be calculated as X × 2n - 32 in which X is the decimal value of the prefix. In other words, the beginning address is X × N.

Question: How can we recognize the different classes of IP addresses? 3 Marks CSE- 2011

Recognition the different classes of IP addresses:

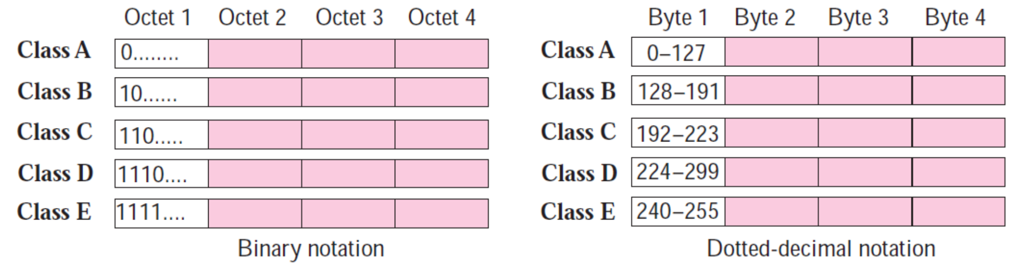
We can recognize the class of an address when the address is given either in binary or dotted decimal notation. In the binary notation, the first few bits can immediately tell us the class of the address; in the dotted-decimal notation, the value of the first byte can give the class of an address (Figure 5.6).

If the address is given in binary notation, then for class A IP addresses, first will be starts with 0, for class B, first two bits will be 1,0 respectively, for class C, First three bits will be 1,1,0, for class D, first four bits will be 1,1,1,0 and finally for class E, first four bits will be 1,1,1,1.

If the address is given in dotted decimal notation, then, the value of the first byte will be 0-127 for class-A, 128-191 for class-B, 192-223 for class-C, 244-299 for class-D and 240-255 for class-E IP addresses.

In classful addressing, an IP address in classes A, B, and C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address. Figure 5.8 shows the netid and hostid bytes. Classes D and E are not divided into netid and hosted. In class A, 1 byte defines the netid and 3 bytes define the hostid. In class B, 2 bytes define the netid and 2 bytes define the hostid. In class C, 3 bytes define the netid and 1 byte defines the hostid.

Figure 5.6 *Finding the class of an address*



	Octet 1	Octet 2	Octet 3	Octet 4
Class A	0........			
Class B	10......			
Class C	110.....			
Class D	1110....			
Class E	1111....			

Binary notation

	Byte 1	Byte 2	Byte 3	Byte 4
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–299			
Class E	240–255			

Dotted-decimal notation

Question: Describe the connection establishment operation with example. 6 Marks CSE-2011

Connection establishment operation

TCP transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously. This implies that each party must initialize communication and get approval from the other party before any data are transferred.

Three-Way Handshaking

The connection establishment in TCP is called three-way handshaking. In our example, an application program, called the client, wants to make a connection with another application program, called the server, using TCP as the transport layer protocol. The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This request is called a passive open. Although the server TCP is ready to accept a connection from any machine in the world, it cannot make the connection itself.

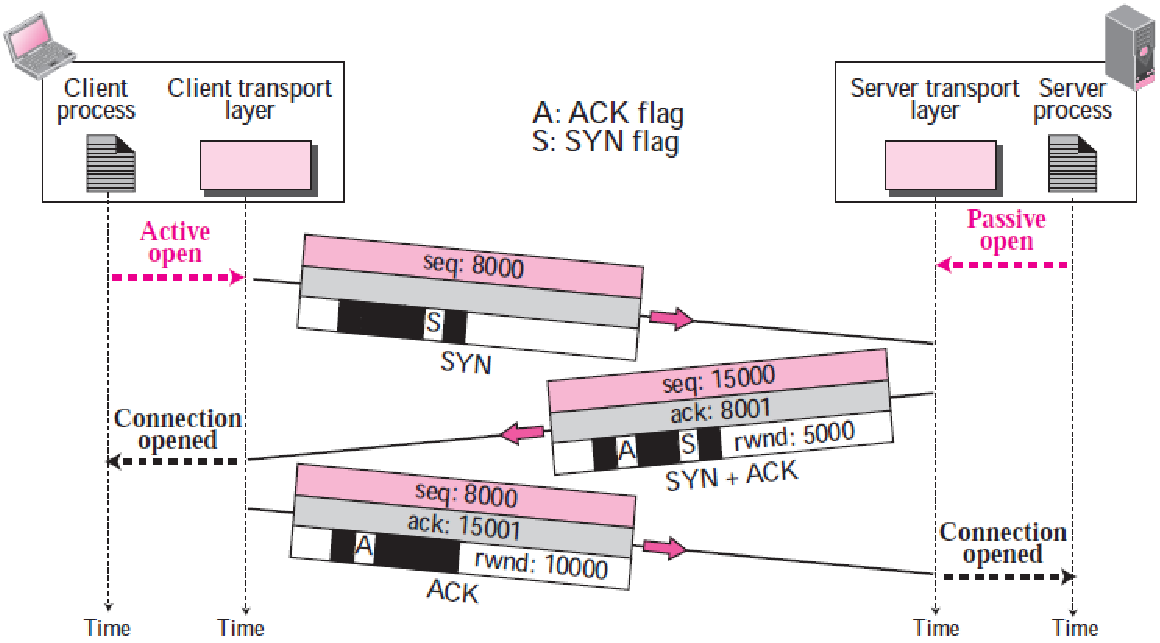
The client program issues a request for an active open. A client that wishes to connect to an open server tells its TCP to connect to a particular server. TCP can now start the three-way handshaking process as shown in Figure 15.9. To show the process we use time lines. Each segment has values for all its header fields and perhaps for some of its option fields too. However, we show only the few fields necessary to understand each phase. We show the sequence number, the acknowledgment number, the control flags (only those that are set), and window size if relevant. The three steps in this phase are as follows.

1. The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers. The client in our example chooses a random number as the first sequence number and sends this number to the server. This sequence number is called the initial sequence number (ISN). Note that this segment does not contain an acknowledgment number. It does not define the window size either; a window size definition makes sense only when a segment includes an acknowledgment. The segment can also include some options that we discuss later in the chapter. Note that the SYN segment is a control segment and carries no data.

However, it consumes one sequence number. When the data transfer starts, the ISN is incremented by 1. We can say that the SYN segment carries no real data, but we can think of it as containing one imaginary byte. A SYN segment cannot carry data, but it consumes one sequence number.

2. The server sends the second segment, a SYN + ACK segment with two flag bits set: SYN and ACK. This segment has a dual purpose. First, it is a SYN segment for communication in the other direction. The server uses this segment to initialize a sequence number for numbering the bytes sent from the server to the client. The server also acknowledges the receipt of the SYN segment from the client by setting the ACK flag and displaying the next sequence number it expects to receive from the client. Because it contains an acknowledgment, it also needs to define the receive window size, rwnd (to be used by the client). A SYN + ACK segment cannot carry data, but does consume one sequence number.

Figure 15.9 *Connection establishment using three-way handshaking*



1. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field. Note that the sequence number in this segment is the same as the one in the SYN segment; the ACK segment does not consume any sequence numbers. The client must also define the server window size. Some implementations allow this third segment in the connection phase to carry the first chunk of data from the client. In this case, the third segment must have a new sequence number showing the byte number of the first byte in the data. In general, the third segment usually does not carry data and consumes no sequence numbers. An ACK segment, if carrying no data, consumes no sequence number.

Question: What do you mean by Cryptology and Cryptography, Cryptanalysis, Digital signal? 8 Marks CSE-2011 CSE-2009

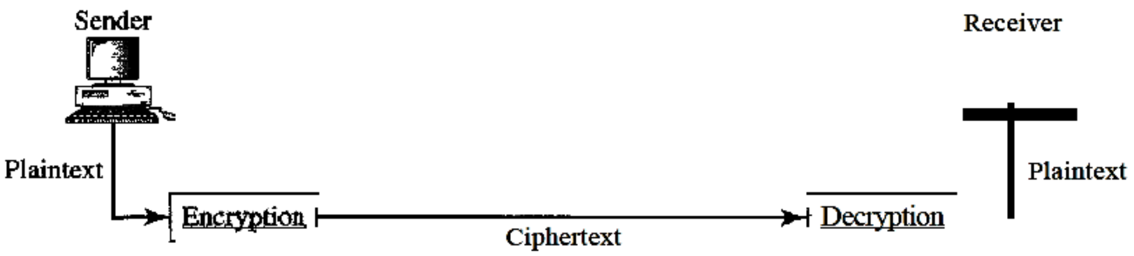
Cryptanalysis:

The messages to be encrypted, known as the **plaintext**, are transformed by a function that is parameterized by a **key**. The output of the encryption process, known as the **ciphertext**, is then transmitted, often by messenger or radio. We assume that the enemy, **or intruder**, hears and accurately copies down the complete ciphertext. However, unlike the intended recipient, he does not know what the decryption key is and so cannot decrypt the ciphertext easily. Sometimes the intruder can not only listen to the communication channel (passive intruder) but can also record messages and play them back later, inject his own messages, or modify legitimate messages before they get to the receiver (active intruder). **The art of breaking ciphers is called cryptanalysis**.

Cryptography:

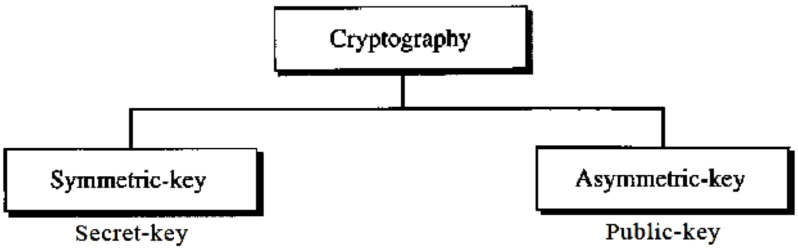
The art of devising ciphers is called cryptography. Cryptography, a word with Greek origins, means “secret writing.” However, we use the term to refer to the science and art of transforming messages to make them secure and immune to attacks. Figure 30.1 shows the components involved in cryptography.

Figure 30.1 *Cryptography components*



We can divide all the cryptography algorithms (ciphers) into two groups: symmetric key (also called secret-key) cryptography algorithms and asymmetric (also called public-key) cryptography algorithms. Figure 30.2 shows the taxonomy.

Figure 30.2 *Categories ofcryptography*



Cryptology:

The art of breaking ciphers, called cryptanalysis, and **the art of devising ciphers (cryptography)** is collectively known as **cryptology.**

Digital signature:

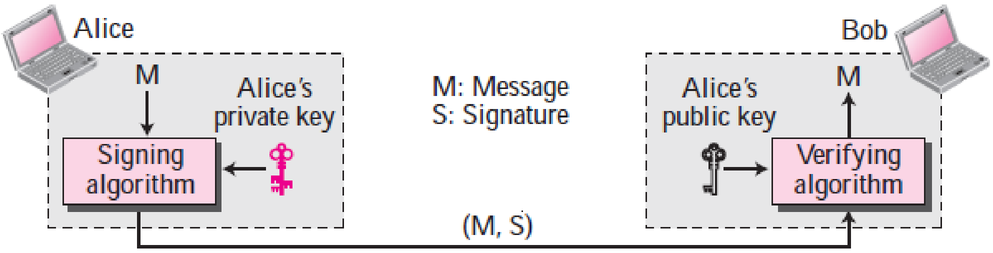
Digital signature is a way to provide message integrity and message authentication and some more security services. A MAC uses a secret key to protect the digest; a digital signature uses a pair of private-public keys.

When we sign a document digitally, we send the signature as a separate document. For a digital signature, the recipient receives the message and the signature. A copy of the signature is not stored anywhere. The recipient needs to apply a verification technique to the combination of the message and the signature to verify the authenticity. For a digital signature, there is a one-to-one relationship between a signature and a message. Each message has its own signature. The signature of one message cannot be used in another message. Each message needs a new signature.

When Alice sends a message to Bob, Bob needs to check the authenticity of the sender; he needs to be sure that the message comes from Alice and not Eve. Bob can ask Alice to sign the message electronically. In other words, an electronic signature can prove the authenticity of Alice as the sender of the message. We refer to this type of signature as a digital signature.

Figure 29.18 shows the digital signature process. The sender uses a signing algorithm to sign the message. The message and the signature are sent to the receiver. The receiver receives the message and the signature and applies the verifying algorithm to the combination. If the result is true, the message is accepted; otherwise, it is rejected.

Figure 29.18 *Digital signature process*



In a digital signature, the signer uses her private key, applied to a signing algorithm, to sign the document. The verifier, on the other hand, uses the public key of the signer, applied to the verifying algorithm, to verify the document. A digital signature needs a public-key system. The signer signs with her private key; the verifier verifies with the signer's public key. A cryptosystem uses the private and public keys of the receiver where as a digital signature uses the private and public keys of the sender.

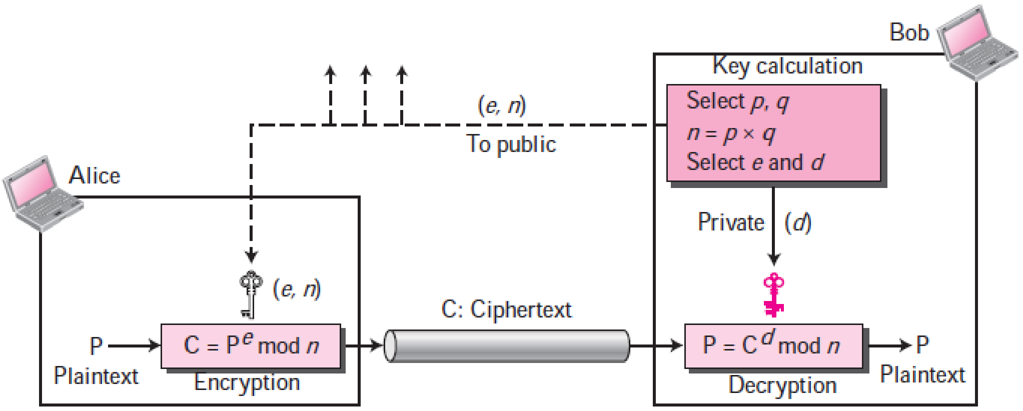
বিশেষ নির্দেশনাঃ ডিজিটাল সিগনেচার সংক্রান্ত প্রশ্ন টীকা আকারে আসতে পারে পরীক্ষায়, সেজন্য ডিজিটাল সিগনেচার সম্পর্কে বিস্তারিত উল্লেখ করা হয়েছে, যাতে পরীক্ষায় ডিজিটাল সিগনেচারের উপর যেকোনোভাবে প্রশ্ন আসলেও উত্তর করা যায় ।

Question: Describe RSA algorithm with example. 6 Marks CSE-2011 CSE-2009

<u>RSA algorithm:</u>

The most common public key algorithm is RSA, named for its inventors Rivest, Shamir, and Adleman (RSA). It uses two numbers, e and d, as the public and private keys, as shown in Figure 29.15.

Figure 29.15 *Encryption, decryption, and key generation in RSA*



In RSA, e and n are announced to the public; d and ϕ are kept secret. The two keys, *e* and *d*, have a special relationship to each other. We show how to calculate the keys without proof.

Selecting Keys

Bob use the following steps to select the private and public keys:

1. Bob chooses two very large prime numbers p and q. A prime number is one that can be divided evenly only by 1 and itself.

2. Bob multiplies the above two primes to find n, the modulus for encryption and decryption. In other words, $n = p \times q$.

3. Bob calculates another number $\phi = (p - 1) \times (q - 1)$.

4. Bob chooses a random integer e. He then calculates d so that $d \times e = 1 \, mod \, \phi$

5. Bob announces e and n to the public; he keeps ϕ and d secret.

Encryption

Anyone who needs to send a message to Bob can use n and e. For example, if Alice needs to send a message to Bob, she can change the message, usually a short one, to an integer. This is the plaintext. She then calculates the ciphertext, using e and n.

$$\boldsymbol{C = P^e \, (mod \, n)}$$

Alice sends C, the ciphertext, to Bob.

Decryption

Bob keeps ϕ and d private. When he receives the ciphertext, he uses his private key d to decrypt the message:

$$P = C^d \, (mod \, n)$$

Restriction

For RSA to work, the value of P must be less than the value of n. If P is a large number, the plaintext needs to be divided into blocks to make P less than n.

Example:

Bob chooses 7 and 11 as p and q and calculates n = 7 . 11 = 77. The value of ϕ = (7 - 1) (11 - 1) or 60. Now he chooses two keys, e and d. If he chooses e to be 13, then d is 37. Now imagine Alice sends the plaintext 5 to Bob. She uses the public key 13 to encrypt 5.

Plaintext: 5
$\boldsymbol{C = 5^{13} = 26 \, mod \, 77}$
Ciphertext: 26

Bob receives the ciphertext 26 and uses the private key 37 to decipher the ciphertext:

Ciphertext: 26
$P = 26^{37} = 5 \, mod \, 77$
Plaintext: 5 **Intended message sent by Alice.**

The plaintext 5 sent by Alice is received as plaintext 5 by Bob.

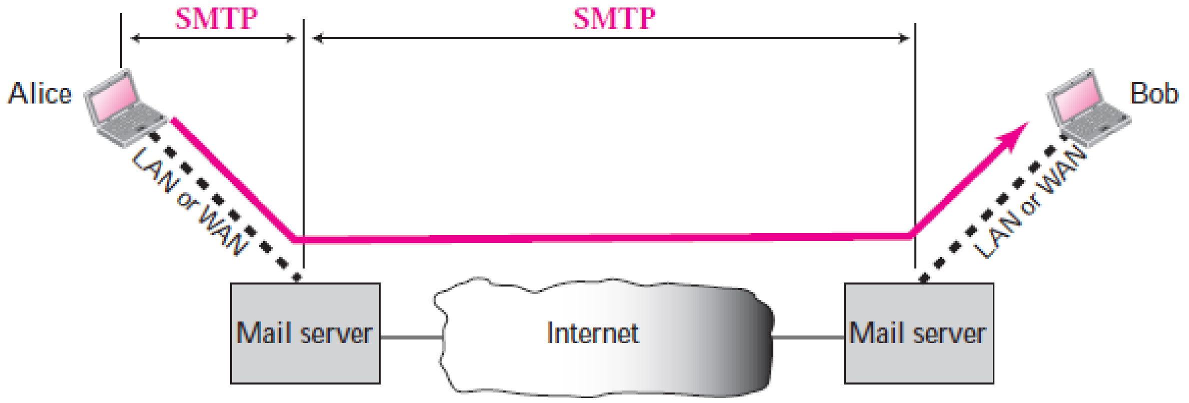
Question: What is SMTP? Why it is used? 4 Marks CSE-2011

Simple Mail Transfer Protocol (SMTP)

The actual mail transfer is done through message transfer agents (MTAs). To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The formal protocol that defines the MTA client and server in the Internet is called Simple Mail Transfer Protocol (SMTP).

Two pairs of MTA client-server programs are used in the most common situation (fourth scenario). Figure 23.8 shows the range of the SMTP protocol in this scenario.

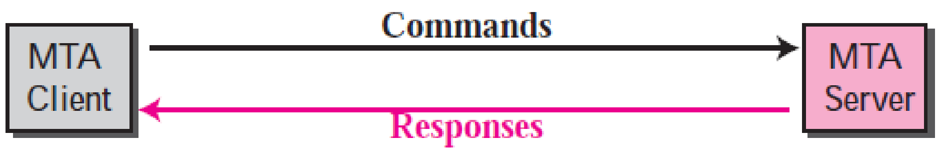
Figure 23.8 *SMTP range*



SMTP is used two times, between the sender and the sender's mail server and between the two mail servers. Another protocol is needed between the mail server and the receiver. SMTP simply defines how commands and responses must be sent back and forth. Each network is free to choose a software package for implementation.

SMTP uses commands such as **HELO, MAIL FROM, RCPT TO** etc. and responses such as **211 for** System status or help reply, **214 for** Help message etc. to transfer messages between an MTA client and an MTA server (see Figure 23.9). Each command or reply is terminated by a two-character (carriage return and line feed) end-of-line token.

Figure 23.9 *Commands and responses*



Why SMTP is used:

We use SMTP to send an e-mail and simulate the commands and responses. The actual mail transfer is done through message transfer agents (MTAs). To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. SMTP protocol is used to define the MTA client and server in the Internet. SMTP is used two times, between the sender and the sender's mail server and between the two mail servers. SMTP simply defines how commands and responses must be sent back and forth. The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination. After a client has made a TCP connection to the well-known port 25, the SMTP server starts the connection phase. After connection has been established between the SMTP client and server, a single message between a sender and one or more recipients can be exchanged. **After the message is transferred successfully, the client terminates the connection by sending QUIT command. Simple mail Transfer Protocol is used in the** process of transferring a mail message .

[নিচের প্রশ্নের কোন গ্রহণযোগ্য উত্তর না পাওয়ায় ফাইল ট্রান্সফার প্রটোকল সম্পর্কে যা যা পরীক্ষার জন্য পড়া লাগবে, তার সবকিছু এখানে সংযুক্ত করা হলো ।]

Question: Describe the working principle of FTP. 6 Marks CSE-2011 CSE-2009

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 of these problems have been solved by FTP in a very simple and elegant approach.

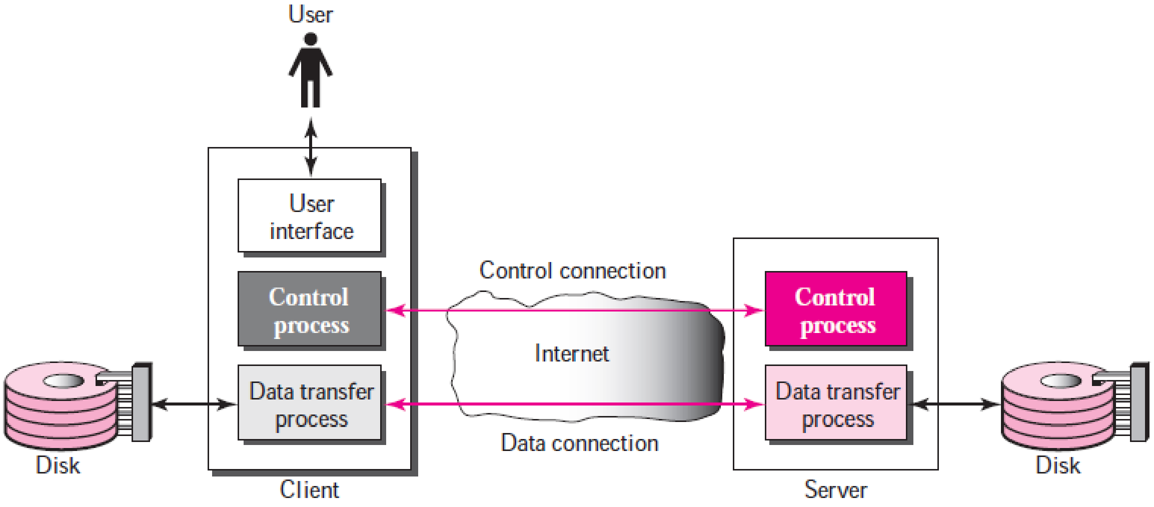
FTP differs from other client-server applications in that it establishes two connections between the hosts. One connection is used for data transfer, the other for control information (commands and responses). Separation of commands and data transfer makes FTP more efficient. The control connection uses 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.

FTP uses two well-known TCP ports: Port 21 is used for the control connection, and port 20 is used for the data connection. Figure 21.1 shows the basic model of FTP. The client has three components: user interface, client control process, and the client data transfer process. The server has two components: the server control process and the server data transfer process. The control connection is made between the control processes. The data connection is made between the data transfer processes. The control connection remains connected during the entire interactive FTP session. The data connection is opened and then closed for each file transferred. It opens each time commands that involve transferring files are used, and it closes when the file is transferred. In other words, when a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.

Connections:

The two FTP connections, control and data, use different strategies and different port numbers.

Figure 21.1 *FTP*



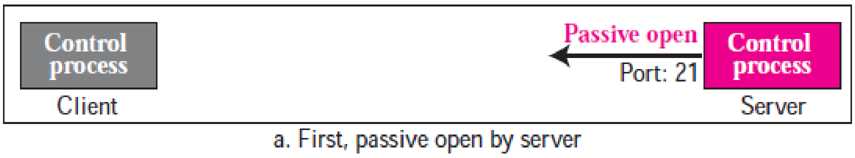
Control Connection:

The control connection is created in the same way as other application programs described so far. There are two steps:

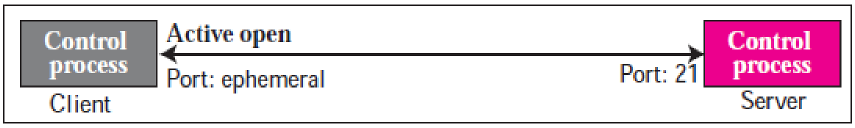
1. The server issues a passive open on the well-known port 21 and waits for a client.
2. The client uses an ephemeral port and issues an active open.

The connection remains open during the entire process. The service type, used by the IP protocol, is minimize delay because this is an interactive connection between a user (human) and a server. The user types commands and expects to receive responses without significant delay. Figure 21.2 shows the initial connection between the server and the client.

Figure 21.2 *Opening the control connection*



a. First, passive open by server



b. Later, active open by client

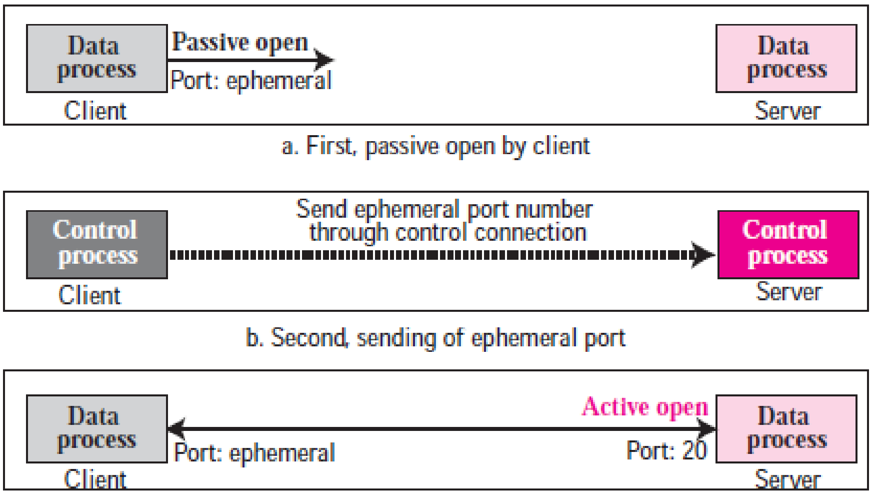
Data Connection:

The data connection uses the well-known port 20 at the server site. However, the creation of a data connection is different from what we have seen so far. The following shows how FTP creates a data connection:

1. The client, not the server, issues a passive open using an ephemeral port. This must be done by the client because it is the client that issues the commands for transferring files.
2. The client sends this port number to the server using the PORT command.
3. The server receives the port number and issues an active open using the wellknown port 20 and the received ephemeral port number.

The steps for creating the initial data connection are shown in Figure 21.3.

Figure 21.3 *Creating the data connection*



a. First, passive open by client

b. Second, sending of ephemeral port

c. Third, active open by server

Communication:

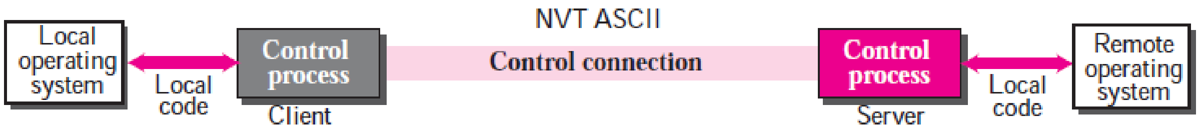
The FTP client and server, which run on different computers, must communicate with each other. These two computers may use different operating systems, different character sets, different file structures, and different file formats. FTP must make this heterogeneity compatible.

FTP has two different approaches, one for the control connection and one for the data connection. We will study each approach separately.

Communication over Control Connection:

FTP uses the same approach as TELNET or SMTP to communicate across the control connection. It uses the NVT ASCII character set (see Figure 21.4). Communication is achieved through commands and responses. This simple method is adequate for the control connection because we send one command (or response) at a time. Each command or response is only one short line so we need not worry about file format or file structure. Each line is terminated with a two-character (carriage return and line feed) end-of-line token.

Figure 21.4 *Using the control connection*

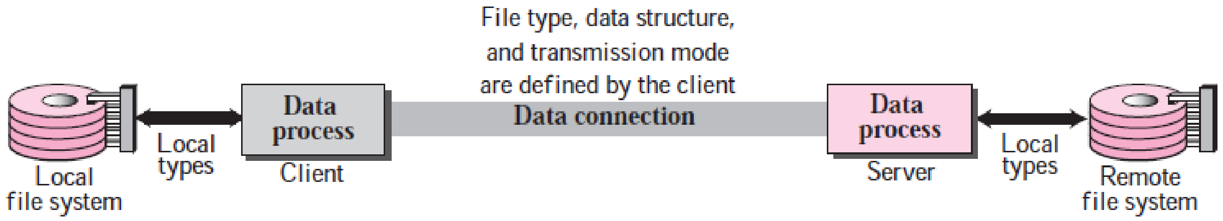


.

Communication over Data Connection:

The purpose and implementation of the data connection are different from that of the control connection. We want to transfer files through the data connection. The client must define the type of file to be transferred, the structure of the data, and the transmission mode. Before sending the file through the data connection, we prepare for transmission through the control connection. The heterogeneity problem is resolved by defining three attributes of communication: file type, data structure, and transmission mode (see Figure 21.5).

Figure 21.5 *Using the data connection*



File Type:

FTP can transfer one of the following file types across the data connection:

1. ASCII file.
2. EBCDIC file.
3. Image file
4. Nonprint
5. TELNET.

Data Structure:

FTP can transfer a file across the data connection using one of the following interpretations about the structure of the data:

1. File structure (default).
2. **Record structure.**
3. Page structure

Transmission Mode:

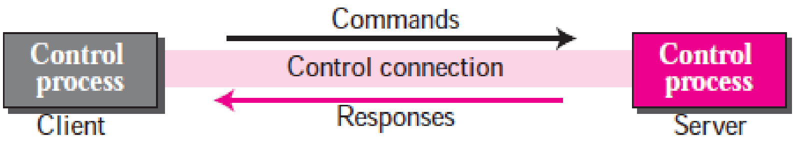
FTP can transfer a file across the data connection using one of the following three transmission modes:

1. Stream mode.
2. Block mode
3. Compressed mode

Command Processing:

FTP uses the control connection to establish a communication between the client control process and the server control process. During this communication, the commands are sent from the client to the server and the responses are sent from the server to the client (see Figure 21.6). Commands, which are sent from the FTP client control process, are in the form of ASCII uppercase, which may or may not be followed by an argument. We can roughly divide the commands into six groups: access commands, file management commands, data formatting commands, port defining commands, file transferring commands, and miscellaneous commands. Every FTP command generates at least one response. A response has two parts: a three-digit number followed by text. The numeric part defines the code; the text part defines needed parameters or extra explanations. We represent the three digits as xyz.

Figure 21.6 *Command processing*



File Transfer

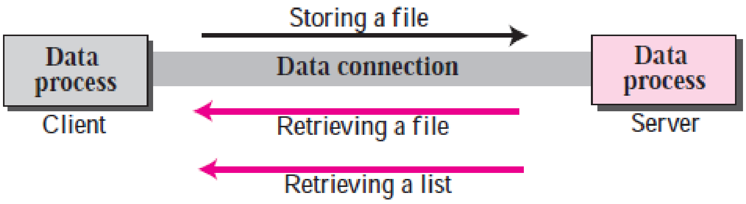
File transfer occurs over the data connection under the control of the commands sent over the control connection. However, we should remember that file transfer in FTP means one of three things (see Figure 21.7).

❑ A file is to be copied from the server to the client (download). This is called *retrieving a file*. It is done under the supervision of the RETR command.

❑ A file is to be copied from the client to the server (upload). This is called *storing a file*. It is done under the supervision of the STOR command.

❑ A list of directory or file names is to be sent from the server to the client. This is done under the supervision of the LIST command. Note that FTP treats a list of directory or file names as a file. It is sent over the data connection.

Figure 21.7 *File transfer*



Question: Write about BOOTP, and DHCP. 4 Marks CSE-2011 CSE-2009

Bootstrap Protocol (BOOTP):

The Bootstrap Protocol (BOOTP) is the prerunner of DHCP. It provides configuration information from a table (file). It is a client/server protocol designed to provide physical address to logical address mapping.

The Bootstrap Protocol (BOOTP) is a client/server protocol that configures a diskless computer or a computer that is booted for the first time. BOOTP provides the IP address, net mask, the address of a default router, and the address of a name server.

It overcomes the two deficiencies of the RARP protocol. First, since it is a client/server program, the BOOTP server can be anywhere in the Internet. Second, it can provide all pieces of information (i.e. The IP address of the computer, the subnet mask of the computer, the IP address of a router, the IP address of a name server) including the IP address. To provide the four pieces of information described above, it removes all restriction about the RARP protocol. BOOTP, however, is a static configuration protocol. When a client requests its IP address, the BOOTP server consults a table that matches the physical address of the client with its IP address. This implies that the binding between the physical address and the IP address of the client already exists. The binding is predetermined. There are some situations in which we need a dynamic configuration protocol. For example, when a host moves from one physical network to another, its physical address changes. As another example, there are occasions when a host wants a temporary IP address to be used for a period of time. BOOTP cannot handle these situations because the binding between the physical and IP addresses is static and fixed in a table until changed by the administrator. As we will see shortly, DHCP has been devised to handle these shortcomings.

BOOTP is an application layer protocol. The administrator may put the client and the server on the same network or on different networks, as shown in Figure 21.7. BOOTP messages are encapsulated in a UDP packet, and the UDP packet itself is encapsulated in an IP packet.

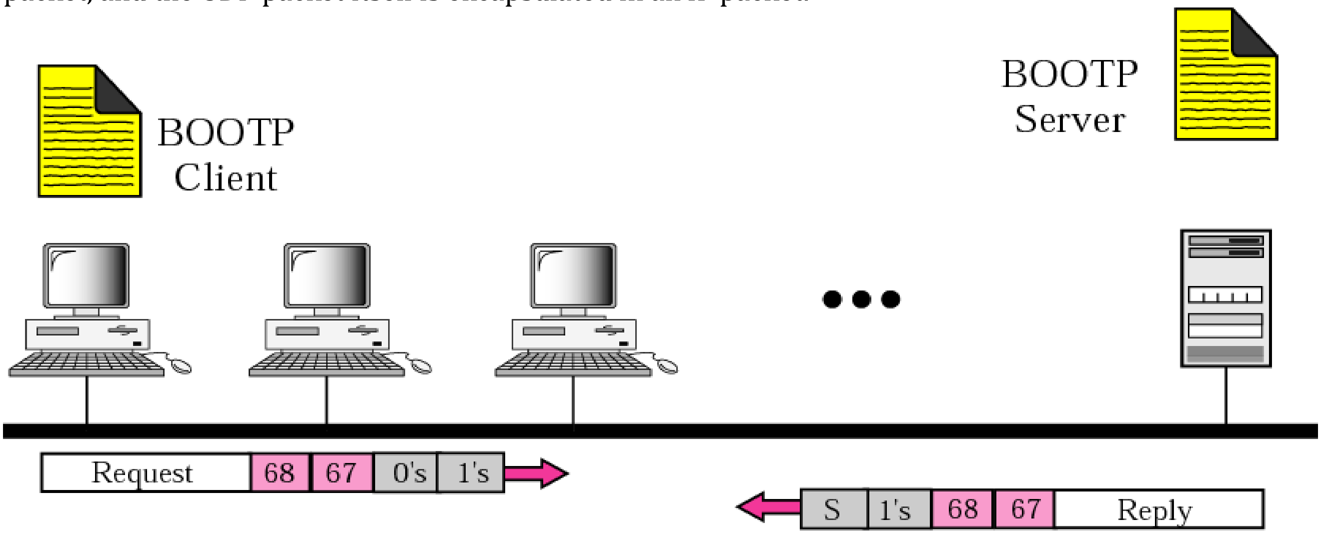


Figure 21.7 (a) Client and server on the same network

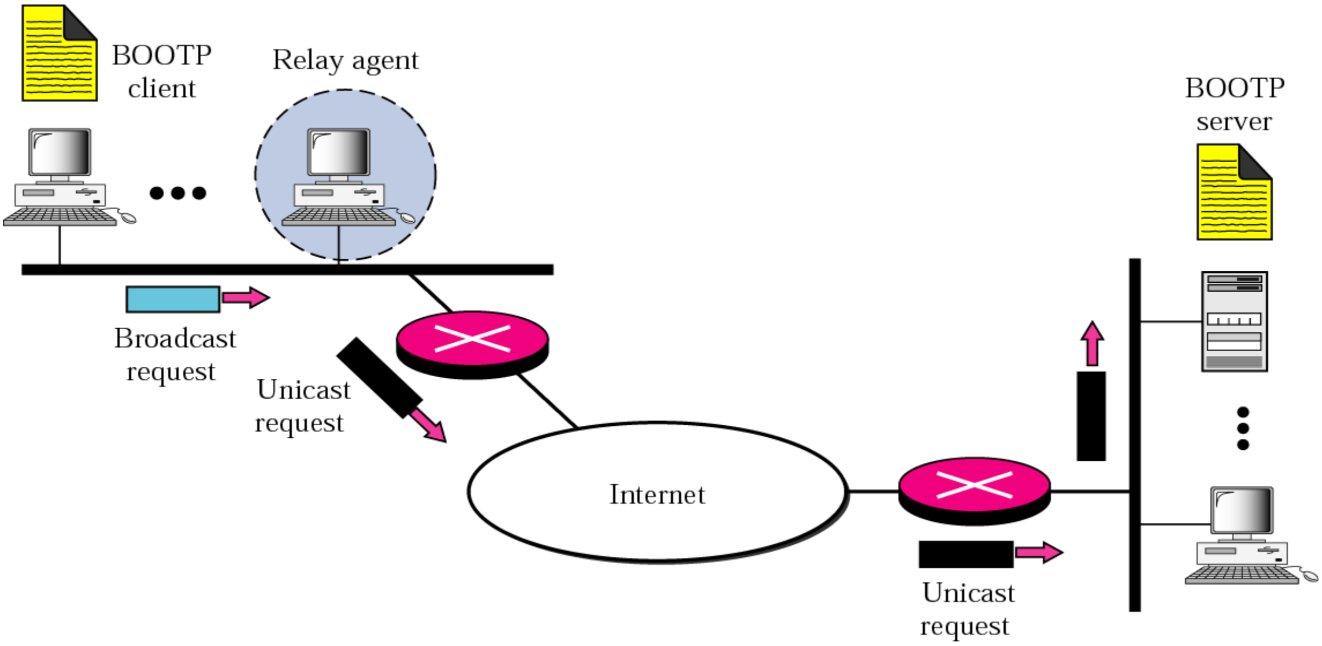


Figure 21.7 (b) Client and server on two different networks

The reader may ask how a client can send an IP datagram when it knows neither its own IP address (the source address) nor the server's IP address (the destination address). The client simply uses all as as the source address and allIs as the destination address.

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

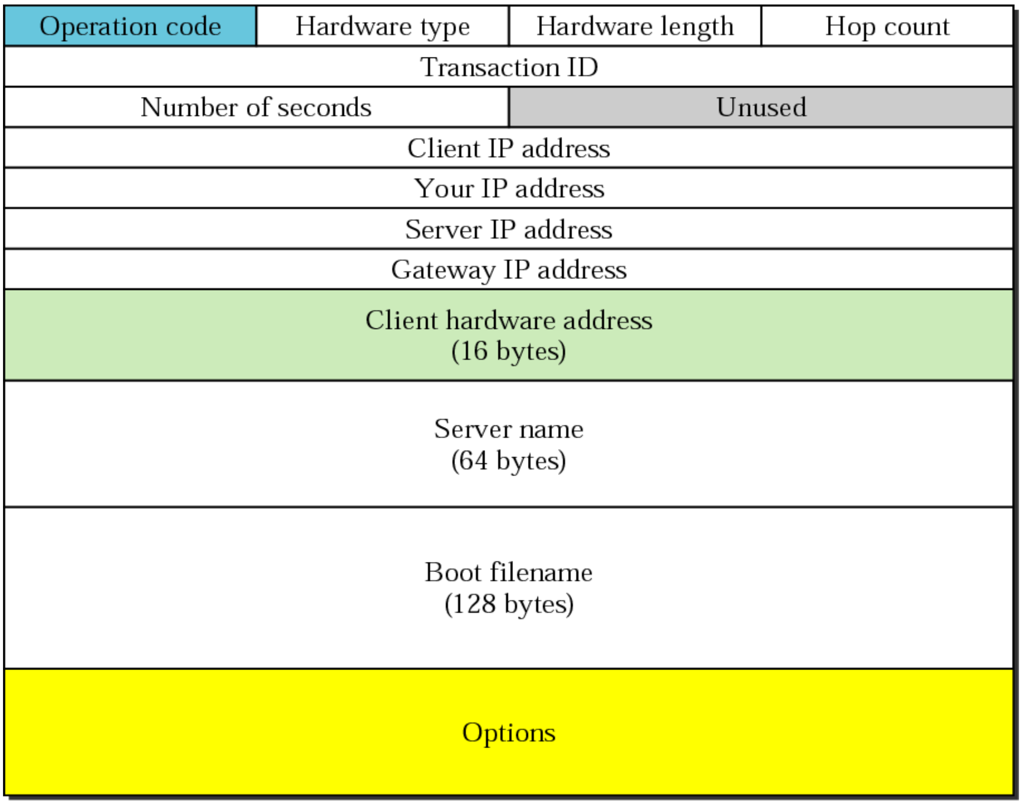


Figure: BOOTP Packet Format

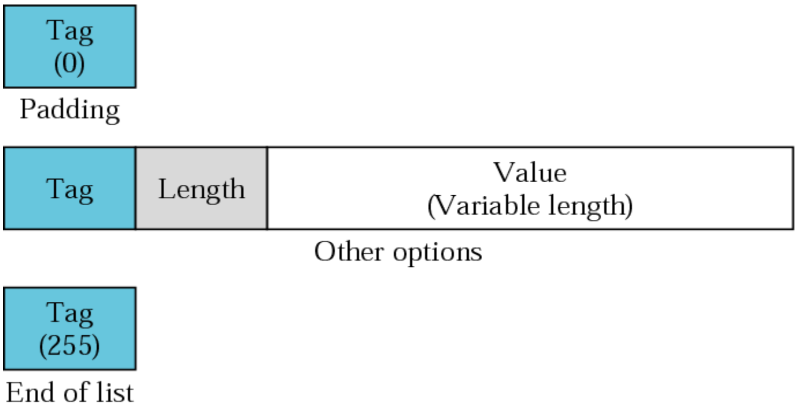
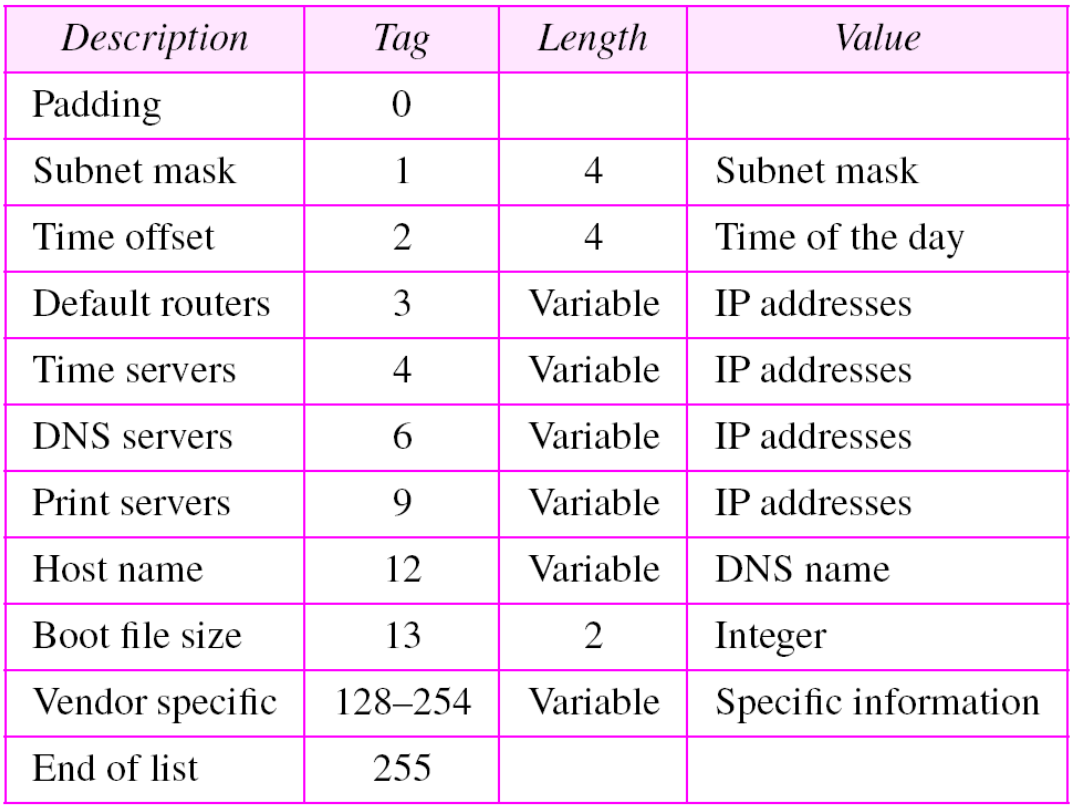


Figure: OPTION format



Description	*Tag*	*Length*	*Value*
Padding	0		
Subnet mask	1	4	Subnet mask
Time offset	2	4	Time of the day
Default routers	3	Variable	IP addresses
Time servers	4	Variable	IP addresses
DNS servers	6	Variable	IP addresses
Print servers	9	Variable	IP addresses
Host name	12	Variable	DNS name
Boot file size	13	2	Integer
Vendor specific	128–254	Variable	Specific information
End of list	255		

Figure: OPTIONS FOR BOOTP

Dynamic Host Configuration Protocol (DHCP):

The Dynamic Host Configuration Protocol (DHCP) is a client/server protocol designed to provide the four pieces of information for a diskless computer or a computer that is booted for the first time. DHCP is a successor to BOOTP and is backward compatible with it. **DHCP provides static and dynamic address allocation that can be manual or automatic.**

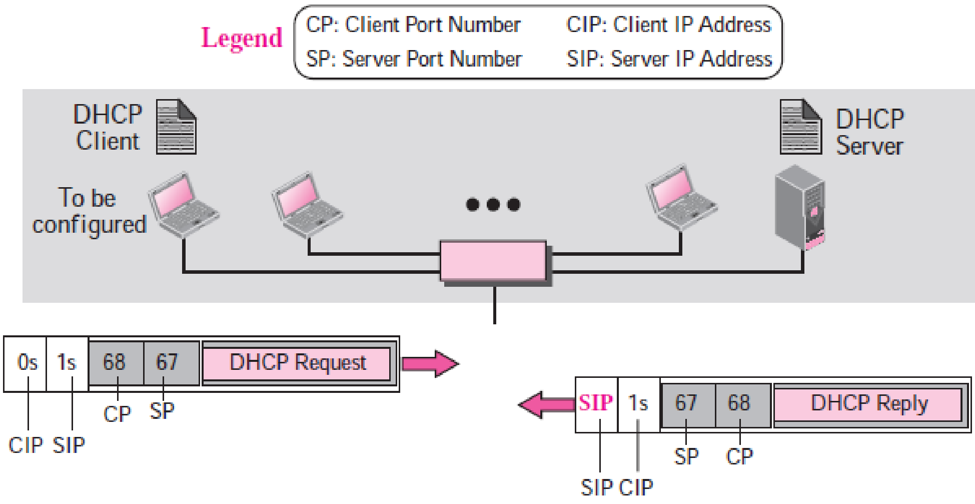
DHCP Operation:

The DHCP client and server can either be on the same network or on different networks. Let us discuss each situation separately.

Same Network

Although the practice is not very common, the administrator may put the client and the server on the same network as shown in Figure 18.1.

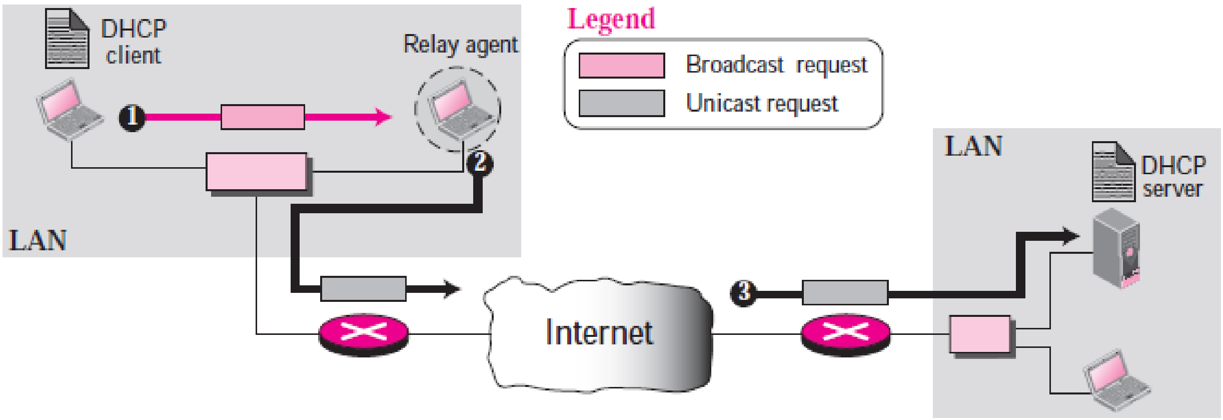
Figure 18.1 *Client and server on the same network*



Different Networks

As in other application-layer processes, a client can be in one network and the server in another, separated by several other networks. Figure 18.2 shows the situation.

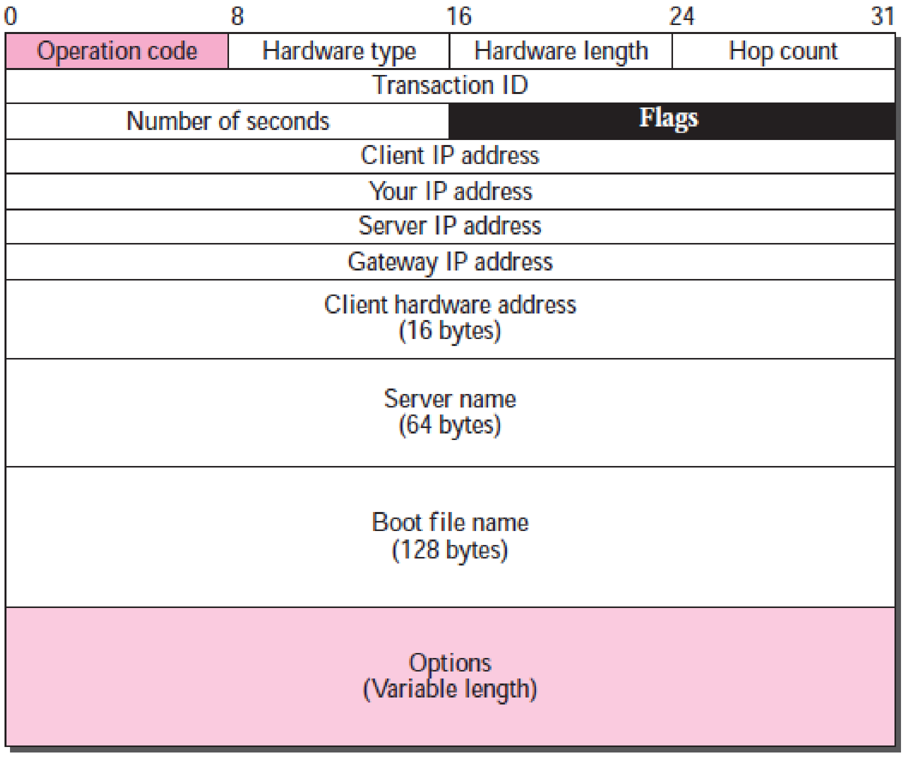
Figure 18.2 *Client and server on two different networks*



Packet Format

Figure 18.4 shows the format of a DHCP packet.

Figure 18.4 *DHCP packet format*



0	8	16	24 31
Operation code	Hardware type	Hardware length	Hop count
Transaction ID			
Number of seconds		Flags	
Client IP address			
Your IP address			
Server IP address			
Gateway IP address			
Client hardware address (16 bytes)			
Server name (64 bytes)			
Boot file name (128 bytes)			
Options (Variable length)			

Configuration:

The DHCP has been devised to provide static and dynamic address allocation.

Static Address Allocation

In this capacity, a DHCP server has a database that statically binds physical addresses to IP addresses. When working in this way, DHCP is backward compatible with the deprecated protocol BOOTP.

Dynamic Address Allocation

DHCP has a second database with a pool of available IP addresses. This second database makes DHCP dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time. When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned. On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database. The dynamic aspect of DHCP is needed when a host moves from network to network or is connected and disconnected from a network (for example, a subscriber to a service provider). DHCP provides temporary IP addresses for a limited period of time. The addresses assigned from the pool are temporary addresses. The DHCP server issues a **lease** for a specific period of time. When the lease expires, the client must either stop using the IP address or renew the lease. The server has the choice to agree or disagree with the renewal. If the server disagrees, the client stops using the address.

যেসকল প্রশ্নের উত্তর প্রদান করা সম্ভব হয়নি, সেগুলোর তালিকা দেওয়া হলোঃ

Question: You are given a datagram of 226 bytes. Write the transfer process of the datagram through ATM Network. 4 Marks CSE-2011

Question: A company is granted the site address 201.70.64.0 (Class C). The company needs six subnets. Design the subnets. 6 Marks CSE-2011

Question: How many layers are considered in a Router and why? 3 Marks CSE-2011

Question: What are the purposes of using IHL and type of service fields in IPv4? 4 Marks CSE-2011

Question: Define Run length Coding and Frequency depended encoding? 4 Marks CSE-2011