Q.1 Attempt any TWO question of the following : [10]

Q.1(a) Find the errors and rewrite the following IPV4 addresses : [5]

(i) 427.45.12.47
(ii) 40.040.35.7.8
(iii) A0.37.27.255
(iv) 27.01010101256.23
(v) 27.56.78.256

(A) (i) Each byte should be less than or equal to 255. 427 is outside this range.
(ii) There should be no leading zeroes in dotted-decimal notation. (040) and we may not have more than 4 bytes in an IPv4 addresses.
(iii) A0 not allowed. Either binary or dotted decimal notation is only allowed.
(iv) A mixture of binary notation and dotted-decimal notation is not allowed.
(v) Each number in the dotted-decimal notation is between 0 and 255. (256 not allowed)

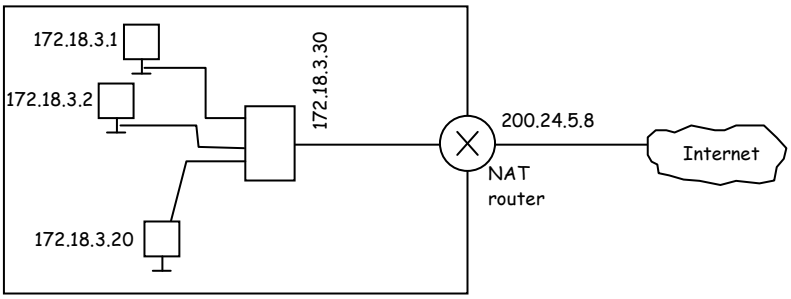
Q.1(b) Compare IPv4 with IPv6. [5]

(A) The following shows the comparison between IPv4 and IPv6 headers.

- The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
- The service type field is eliminated in IPv6. The traffic class and flow label fields together take over the function of the service type field.
- The total length field is eliminated in IPv6 and replaced by the payload length field.
- The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
- The TTL field is called hop limit in IPv6.
- The protocol field is replaced by the next header field.
- The header checksum is eliminated because the checksum is provided by upper layer protocols; it is therefore not needed at this level.
- The option fields in IPv4 are implemented as extension headers in IPv6.

Q.1(c) Explain the concept of Network Address Translation. [5]

(A)



The distribution of addresses through ISPs has created a new problem. Assume, that an ISP has granted a small range of addresses to a small business or a household. If the business grows or the household needs a large range so, network address translation (NAT). The technology allows a site to use a set of private addresses for internet communication & a set of global internal address for the communication with the rest of the world. The site must have only one single connection to the global internet through a NAT capable route that runs NAT software the private network uses private addresses. The route that connects the network to the global address uses one private address & one global address. The private network is transparent to the rest of the internet the rest of the internet sees only the NAT route with the address 200.24.5.8.

- **Address translation** : All the outgoing packets go through the NAT route, which replaces the source address in the packet with the global NAT address. All incoming packets also pass through the NAT route, which replaces the destination address in the packet with the appropriate private address.
- **Translation table** : The reader may have noticed that translating the source address for an outgoing packet is straight forward. They are 10's of private IP addresses, each belonging to one specific host. The problem is solved if the NAT route has a translation table.
- **Using one IP address** : It has private address and the external address. When the router translates the source address of the outgoing packet, it also makes note of the destination address when the packet is going.
- **Using a pool of IP address** : It using 4 address (200.24.5.8, 200.24.5.9, 200.24.5.10 and 200.24.5.11) to external server programs.
  Using both IP addresses and port addresses.
  It allows many–to–many relationship between private network hosts and external server programs when the response from HTTP comes back, the combination of source address & destination port address defines the private network host to which the response should be directed.

Q.1(d) Draw a neat labeled diagram and explain IPv4 datagram header format. [5]

(A)

- **Packets in the network (internet) layer are called datagrams.**
- Following Figure shows the IP datagram format. A datagram is a variable-length packet consisting of two parts: header and data. The header is 20 to 60 bytes in length and contains information essential to routing and delivery. It is customary in TCP/IP to show the header in
  4-byte sections.

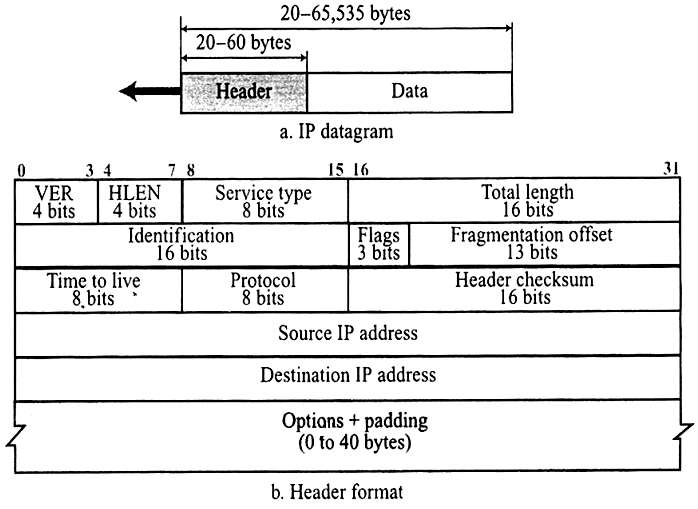


Fig. : IP datagram

- **Version (VER):** This 4-bit field defines the version of the IP protocol. Currently the version is 4. This field tells the IP software running in the processing machine that the datagram has the format of version 4. If the machine is using some other version of IP, the datagram is discarded rather than interpreted incorrectly.
- **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).
- **Service type:** In the original design of IP header, this field was referred to as type of service **(TOS),** which defined how the datagram should be handled. Part of the field was used to define the precedence of the datagram; the rest defined the type of ser-

vice (low delay, high throughput, and so on). IETF has changed the interpretation of this 8-bit field. This field now defines a set of differentiated services. The new interpretation is shown in Figure.

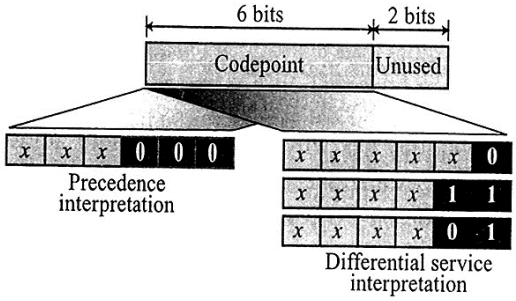


Fig. : Service type

In this interpretation, the first 6 bits make up the code point subfield and the last 2 bits are not used.

- **Total length** : This is a 16-bit field that defines the total length (header plus data) of the IP datagram in bytes. To find the length of the data coming from the upper layer, subtract the header length from the total length. The header length can be found by multiplying the value in the HLEN field by four.
  Length of data = total length - header length

- **Identification** : This field is used in fragmentation.

- **Flags** : This field is used in fragmentation.

- **Fragmentation offset** : This field is used in fragmentation (discussed in the next section).

- **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.

  This field is needed because routing tables in the Internet can become corrupted. A datagram may travel between two or more routers for a long time without ever getting delivered to the destination host. This field limits the lifetime of a datagram.

  Another use of this field is to intentionally limit the journey of the packet.
- **Protocol** : This 8-bit field defines the higher-level protocol that uses the services of the IP layer. An IP 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 IP datagram should be delivered. In other words, since the IP protocol multiplexes and demultiplexes data from different higher-level protocols, the value of this field helps in the demultiplexing process when the datagram arrives at its final destination (see Figure).

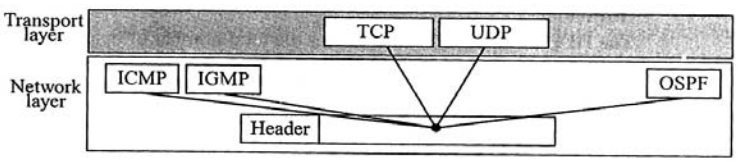


Fig. : Multiplexing

Some of the value of this field for different higher-level protocols is shown in Table 1.

Table 1 : Protocols

Value	Protocol	Value	Protocol
1	ICMP	17	UDP
2	IGMP	89	OSPF
6	TCP		

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

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

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

Q.2 Attempt any TWO question of the following : [10]

Q.2(a)Describe 3 phases of communication between remote host and mobiles host. [5]

(A) To communicate with a remote host, a mobile host goes through three phases: agent discovery, registration, and data transfer, as shown in figure 1.

The first phase, agent discovery, involves the mobile host, the foreign agent, and the home agent. The second phase, registration, also involves the mobile host and the two agents. Finally, in the third phase, the remote host is also involved.

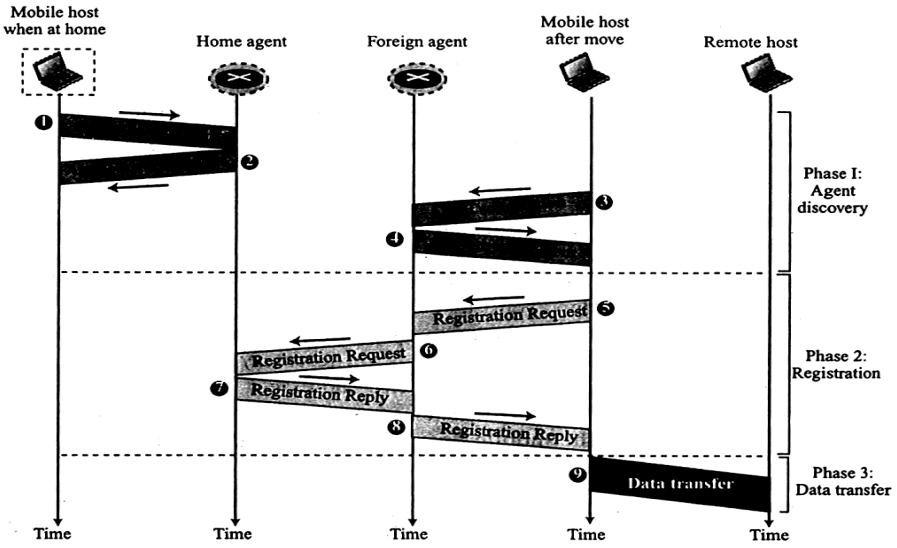


Fig. 1: Remote host and mobile host communication

Phase 1 - Agent Discovery

The first phase in mobile communication, agent discovery, consists of two subphases. A mobile host must discover (learn the address of) a home agent before it leaves its home network. A mobile host must also discover a foreign agent after it has moved to a foreign network. This discovery consists of learning the care-of address as well as the foreign agent's address.
The discovery involves two types of messages: advertisement and solicitation.

Agent Advertisement

- When a router advertises its presence on a network using an ICMP router advertisement, it can append an agent advertisement to the packet if it acts as an agent. Figure 2 shows how an agent advertisement is piggybacked to the router advertisement packet.
  Mobile IP does not use a new packet type for agent advertisement; it uses the router advertisement packet of ICMP, and appends an agent advertisement message.

ICMP Advertisement message			
Type	Length	Sequence number	
Lifetime		Code	Reserved
Care-of addresses (foreign agent only)			

Fig. 2 : Agent advertisement

Agent Solicitation

When a mobile host has moved to a new network and has not received agent advertisements, it can initiate an agent solicitation. It can use the ICMP solicitation message to inform an agent that it needs assistance.

Mobile IP does not use a new packet type for agent solicitation; it uses the router solicitation packet of ICMP.

Phase 2 – Registration

The second phase in mobile communication is registration. After a mobile host has moved to a foreign network and discovered the foreign agent, it must register. There are four aspects of registration:

1) The mobile host must register itself with the foreign agent.
2) The mobile host must register itself with its home agent. This is normally done by the foreign agent on behalf of the mobile host.
3) The mobile host must renew registration if it has expired.
4) The mobile host must cancel its registration (deregistration) when it returns home.

Request and Reply

To register with the foreign agent and the home agent, the mobile host uses a registration request and a registration reply as shown in Figure 3.

Registration Request

A registration request is sent from the mobile host to the foreign agent to register its care-of address and also to announce its home address and home agent address. The foreign agent, after receiving and registering the request, relays the message to the home agent. Note that the home agent now knows the address of the foreign agent because the IP packet that is used for relaying has the IP address of the foreign agent as the source address. Figure 23 shows the format of the registration request.

Type	Flag	Lifetime
Home address		
Home agent address		
Care-of address		
Identification		
Extensions …		

Fig. 3 : Registration request format

Registration Reply : A registration reply is sent from the home agent to the foreign agent and then relayed to the mobile host. The reply confirms or denies the registration request. Figure 4 shows the format of the registration reply.

Type	Code	Lifetime
Home address		
Home agent address		
Identification		
Extensions …		

Fig. 4: Registration reply format

The fields are similar to those of the registration request with the following exceptions. The value of the type field is 3. The code field replaces the flag field and shows the result of the registration request (acceptance or denial). The care-of address field is not needed.

Phase 3 - Data Transfer

After agent discovery and registration, a mobile host can communicate with a remote host. Figure 5 shows the idea.

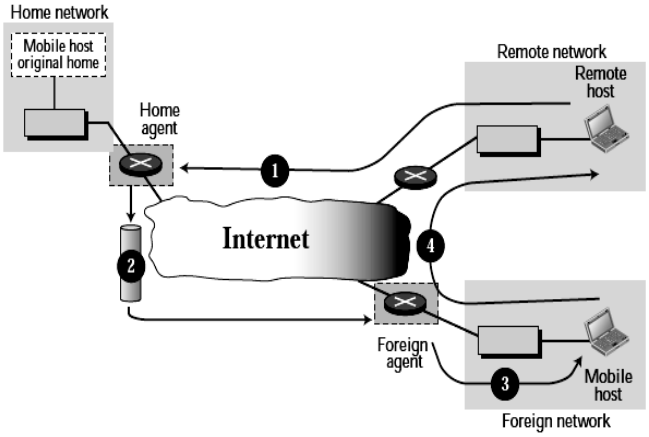


Fig.5 : Data transfer

From Remote Host to Home Agent

When a remote host wants to send a packet to the mobile host, it uses its address as the source address and the home address of the mobile host as the destination address. In other words, the remote host sends a packet as though the mobile host is at its home network. The packet, however, is intercepted by the home agent, which pretends it is the mobile host. Path 1 of Figure 5 shows this step.

From Home Agent to Foreign Agent

After receiving the packet, the home agent sends the packet to the foreign agent using the tunneling. The home agent encapsulates the whole IP packet inside another IP packet using its address as the source and the foreign agent's address as the destination. Path 2 of Figure 5 shows this step.

From Foreign Agent to Mobile Host

When the foreign agent receives the packet, it removes the original packet. However, since the destination address is the home address of the mobile host, the foreign agent consults a registry table to find the care-of address of the mobile host. (Otherwise, the packet would just be sent back to the home network.) The packet is then sent to the care-of address. Path 3 of Figure 5 shows this step.

From Mobile Host to Remote Host

When a mobile host wants to send a packet to a remote host (for example, a response to the packet it has received), it sends as it does normally. The mobile host prepares a packet with its home address as the source, and the address of the remote host as the destination. Although the packet comes from the foreign network, it has the home address of the mobile host. Path 4 of Figure 5 shows this step.

Transparency

In this data transfer process, the remote host is unaware of any movement by the mobile host. The remote host sends packets using the home address of the mobile host as the destination address; it receives packets that have the home address of the mobile host as the source address. The movement is totally transparent. The rest of the Internet is not aware of the mobility of the moving host.

Q.2(b)Explain the cache control module and the cache table in the ARP package. [5]

(A)

- The cache-control module is responsible for maintaining the cache table. It periodically (for example, every 5 s) checks the cache table, entry by entry.
- If the state of the entry is FREE, it continues to the next entry. If the state is PENDING, the module increments the value of the attempts field by 1. It then checks the value of the attempts field. If this value is greater than the maximum number of attempts allowed, the state is changed to FREE and the corresponding queue is destroyed. However, if the number of attempts is less than the maximum, the module creates and sends another ARP request.
- If the state of the entry is RESOLVED, the module decrements the value of the time-out field by the amount of time elapsed since the last check. If this value is less than or equal to zero, the state is changed to FREE and the queue is destroyed.

Cache-Control Module

1. Sleep until the periodic timer matures.
2. For every entry in the cache table
    1. If (the state is FREE)
        1. Continue.
    2. If (the state is PENDING)
        1. Increment the value of attempts by 1.
        2. If (attempts greater than maximum)
            1. Change the state to FREE.
            2. Destroy the corresponding queue.
        3. Else
            1. Send an ARP request.
        4. Continue.
    3. If (the state is RESOLVED)
        1. Decrement the value of time-out by the value of elapsed time.
        2. If (time-out less than or equal to zero)
            1. Change the state to FREE.
            2. Destroy the corresponding queue.
3. Return.

Q.2(c)How Bellman-Ford algorithm helps to find least cost between any two nodes? Explain. [5]

(A) Bellman-Ford Algorithm

The algorithm is based on the fact that if all neighbors of node. I know the shortest distance to node j, then the shortest distance between node I and j can be found by adding the distance between node i and each neighbor to the neighbor's shortest distance to node j and then select the minimum.

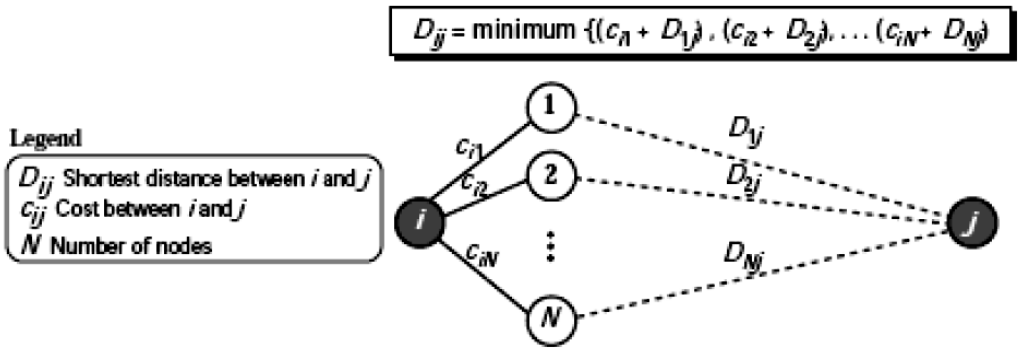


Fig. 1

We create a shortest distance table (vector) for each node using the following steps:

(i) The shortest distance and the cost between a node and itself is initialized to 0.

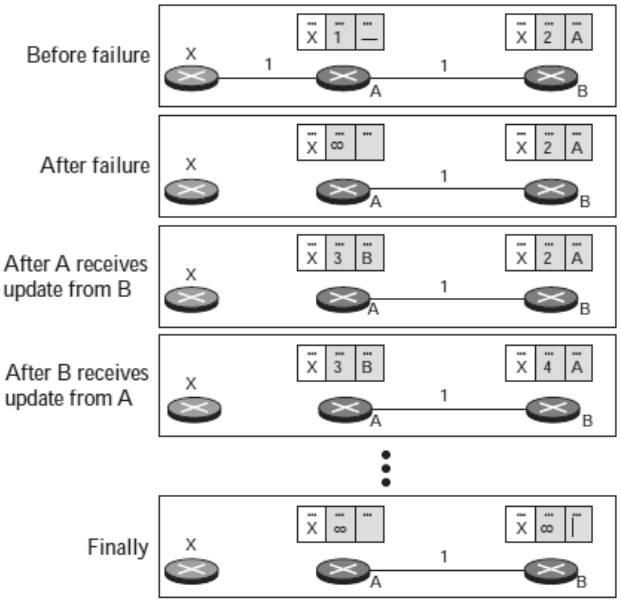
(ii) The shortest distance between a node and any other node is set to infinity. The cost between a node and any other node should be given (can be infinity if the nodes are not connected).

(iii) The algorithm repeates until there is no more changes in the shortest distance vector.

Q.2(d)State and explain the solutions to the two node instability in RIP. [5]

(A) Two-Node Loop

One example of count to infinity is the two-node loop problem. To understand the problem, look at the scenario depicted in Figure



Two-node instability

The figure shows a system with three nodes.

- At the beginning, both nodes A and B know how to reach node X. But suddenly, the link between A and X fails. Node A changes its table. If A can send its table to B immediately, everything is fine. However, the system becomes unstable if B sends its routing table to A before receiving A's routing table.
- Node A receives the update and, assuming that B has found a way to reach X, immediately updates its routing table. Now A sends its new update to B. Now B thinks that something has been changed around A and updates its routing table. The cost of reaching X increases gradually until it reaches infinity.
- At this moment, both A and B know that X cannot be reached. However, during this time the system is not stable. Node A thinks that the route to X is via B; node B thinks that the route to X is via A.
- If A receives a packet destined for X, it goes to B and then comes back to A. Similarly, if B receives a packet destined for X, it goes to A and comes back to B. Packets bounce between A and B, creating a two-node loop problem. A few solutions have been proposed for instability of this kind.

Q.3 Attempt any TWO question of the following : [10]

Q.3(a) What are the options available in TCP? Explain each with its packet format. [5]

(A) (i) End of Option (EOP)

It is a 1-byte option used for padding at the end of the option section. EOP

0 0 0 0 0 0 0 0

(ii) **No Operation (NOP)**

It is 1-byte option used as a filler.

0 0 0 0 0 0 0 1

(iii) Maximum Segment Size (MSS)

It defines the size of the biggest unit of data that can be received by the destination of the TCP segment.

(iv) Window Scale Factor

The window size field in the header defines the size of the sliding window. This field is 16 bits long, which means that the window can range from 0 to 65,535 bytes. To increase the window size, a window scale factor is used.

(v) Timestamp

This is a 10-byte option with the format. Note that the end with the active open announces a timestamp in the connection request segment (SYN). If it receives a timestamp in the next segment (SYN + ACK) from the other end, it is allowed to use the timestamp.

(vi) SACK-Permitted and SACK Options

Selective acknowledgment allows the sender to have a better idea of which segments are actually lost and which have arrived out of order. The new proposal even includes a list for duplicate packets.

The SACK-permitted option of two bytes is used only during connection establishment.

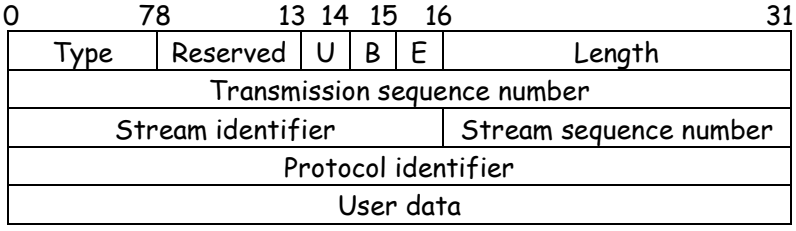
Q.3(b) Explain Data Chunk in detail. [5]

(A) Data chunk :

It carries the user data. A packet may contain zero or more data chunks. The descriptions of the common fields are the same.

The type field has a value of 0. The flag field has 5 reserved bits and 3 defined bits; U, B and E. The U field when set to 1, signals, unordered data. The B (beginning) and E (end) bit together define the position of a chunk in a message that is fragmented when B = 1 and E = 1 then is no fragmentation when B = 1 and E = 0 it is the first fragment. When B = 0 and E = 1, it is the last fragment. When B = 0 and E = 0, it is a middle fragment.

- Transmission segment number (TSN) : This 32–bit field defines the transmission sequence number. It is a sequence number that is initialized in an INIT chunk for one direction & in the INIT ACK chunk for the opposite direction.



Data Chunk :

- **Stream identifier (SI) :** This 16-bit field defines each stream is an association. All chunks belonging to the same stream is one direction carry the same stream identifier.

- **Stream sequence number (SSN) :** This 16–bit defines a chunk in a particular stream is one direction.

- **Protocol identifier :** This 32–bit field can be used by the application program to define the type of data. It is ignored by the SCTP layer.

- **User data :** This field carries the actual user data. SCTP has some specific rules about the user data field. First, no chunk can carry data belonging to more than one message, but a message can be spread our several data chunks. Second, cannot be empty, third if the data cannot end at a 32–bit boundary, padding must be added. The padding bytes are not included in the value of the length.

Q.3(c) Explain in detail UDP Package. **[5]**

(A)

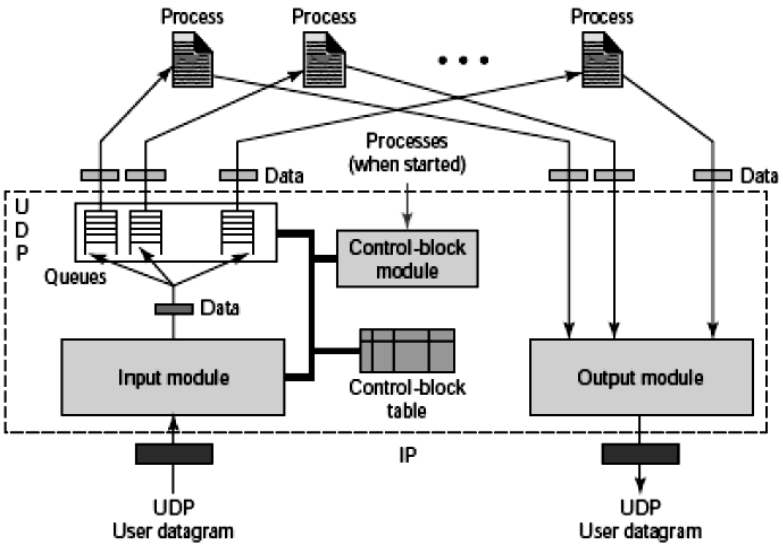


Fig. 1:

Control-Block Table

It keep track of the open ports. The state can be FREE or IN-USE, the process ID, the port number, and the corresponding queue number.

Input Queues

It has a set of input queues, one for each process.

Control-Block Module

It is responsible for the management of the control-block table. When a process starts, it asks for a port number from the operating system.

Input Module

It receives a user datagram from the IP. It searches the control-block table to find an entry having the same port number as this user datagram.

Output Module

It is responsible for creating and sending user datagrams.

Q.3(d) Find the netid of the following IP address : **[5]**

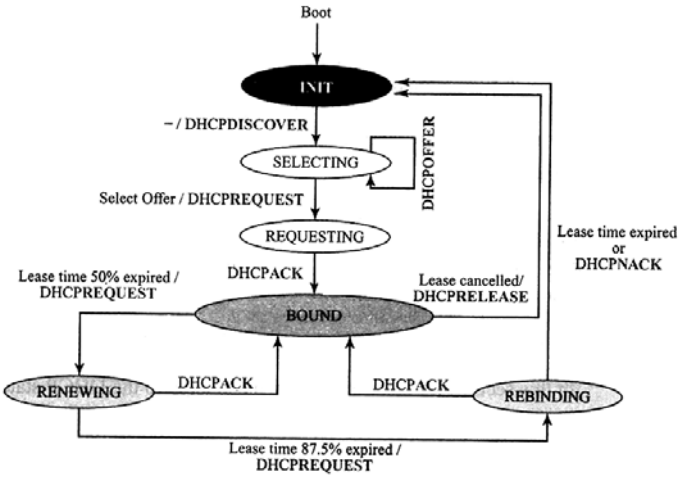
(i) 114.34.2.8
(ii) 132.56.8.6
(iii) 208.34.54.12
(iv) 251.34.98.5
(v) 129.14.6.8

(A) (i) 114.34.2.8 Net id: 114 (ii) 132.56.8.6 Net id: 132.56
(iii) 208.34.54.12 Net id: 208.34.54 (iv) 251.34.98.5 Class E ; No Net id
(v) 129.14.6.8 Net id: 129.14

Q.4 Attempt any TWO question of the following : [10]

Q.4(a) Draw and explain the DHCP Client Transition diagram. [5]

(A) DHCP client transition diagram :



INIT State : When the DHCP client first states it is in the INIT state. The client broadcast a DHCP Discover message using port 67.

Selecting state :

After sending the DHCP discover message the client goes to the selecting state those servers that can provide this type of service respond with a DHCP after message. The client chosen one of the offer and sends a DHCP Request message to the selected server. If there is no reply to any of the DHCP discovers the client sleeps for sleeps for 5 minutes before trying again.

- **Requesting State :** The client remains in the requesting state until it receives a DHCPPACK message from the server that creates the binding because the client physical address & its IP address.
- **BOUND State :** When 50 percent of the lease period is reached, the client sends another DHCP Request to ask for renewal.
- **Renewing State :** The client remains in the renewing state until one of two events happens. It can receive a DHCPPACK, which renews the lease agreement if a DHCPPACK is not received & 87.5 percent of the lease time expires, the client goes to the rebinding state.
- **Rebinding State :** The client remains in the rebinding state until one of the 3 clients happens. If the client receiver a DHCPAPK it goes to the bound state and resets the time.

Q.4(b) How do you established an association in SCTP? Explain. [5]

(A) SCTP requires a four-way handshake. In this procedure, a process, normally a client, wants to establish an association with another process, normally a server, using SCTP as the transport layer protocol.

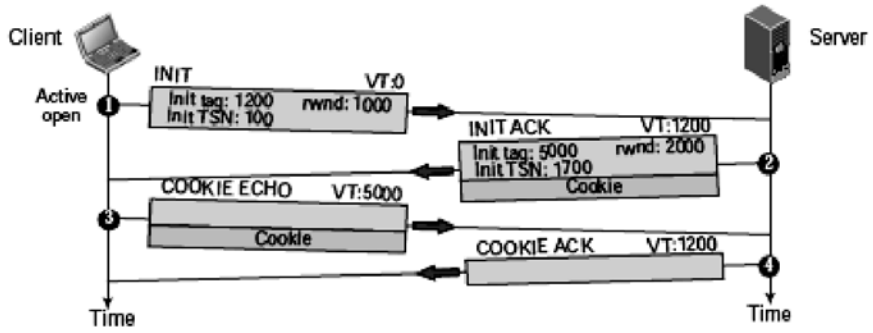


Fig. 1: Four-way handshaking

(1) The client sends the first packet, which contains an INIT chunk. The verification tag (VT) of this packet is 0 because no verification tag has yet been defined for this direction.

(2) The server sends the second packet, which contains an INIT ACK chunk. The verification tag is the value of the initial tag field in the INIT chunk. This chunk initiates the tag to be used in the other direction

(3) The client sends the third packet, which includes a COOKIE ECHO chunk. This is a very simple chunk that echoes, without change, the cookie sent by the server.

(4) The server sends the fourth packet, which includes the COOKIE ACK chunk that acknowledges the receipt of the COOKIE ECHO chunk.

Q.4(c) Explain the features of Stream Control Transmission Protocol. [5]

(A) SCTP Features

Transmission Sequence Number (TSN)

- The unit of data in TCP is a byte. Data transfer in TCP is controlled by numbering bytes using a sequence number.
- On the other hand, the unit of data in SCTP is a data chunk, which may or may not have a one-to-one relationship with the message coming from the process because of fragmentation.
- Data transfer in SCTP is controlled by numbering the data chunks.
- SCTP uses a transmission sequence number (TSN) to number the data chunks.
- In other words, the TSN in SCTP plays the analogous role as the sequence number in TCP.
- TSNs are 32 bits long and randomly initialized between 0 and 232 - 1. Each data chunk must carry the corresponding TSN in its header.
- In SCTP, a data chunk is numbered using a TSN.

Stream Identifier (SI)

- In TCP, there is only one stream in each connection.
- In SCTP, there may be several streams in each association. Each stream in SCTP needs to be identified using a stream identifier (SI).
- Each data chunk must carry the SI in its header so that when it arrives at the destination, it can be properly placed in its stream.
- The SI is a 16-bit number starting from 0.
- To distinguish between different streams, SCTP uses an SI.

Stream Sequence Number (SSN)

- When a data chunk arrives at the destination SCTP, it is delivered to the appropriate stream and in the proper order.
- This means that, in addition to an SI, SCTP defines each data chunk in each stream with a stream sequence number (SSN).
- To distinguish between different data chunks belonging to the same stream, SCTP uses SSNs.

Packets

- In TCP, a segment carries data and control information.
- Data are carried as a collection of bytes; control information is defined by six control flags in the header.
- The design of SCTP is totally different: data are carried as data chunks, control information as control chunks.
- Several control chunks and data chunks can be packed together in a packet.

- A packet in SCTP plays the same role as a segment in TCP. Figure 2 compares a segment in TCP and a packet in SCTP.
- TCP has segments; SCTP has packets.

Acknowledgment Number

- TCP acknowledgment numbers are byte-oriented and refer to the sequence numbers.
- SCTP acknowledgment numbers are chunk-oriented.
- They refer to the TSN.
- A second difference between TCP and SCTP acknowledgments is the control information. This information is part of the segment header in TCP.
- In SCTP, acknowledgment numbers are used to acknowledge only data chunks; control chunks are acknowledged by other control chunks if necessary.

Flow Control

- SCTP implements flow control to avoid overwhelming the receiver.

Error Control

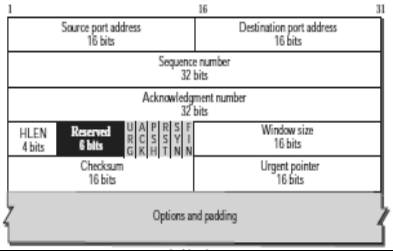
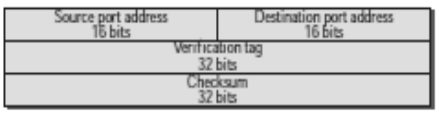
- SCTP implements error control to provide reliability. TSN numbers and acknowledgment numbers are used for error control.

Congestion Control

- SCTP implements congestion control to determine how many data chunks can be injected into the network.

Q.4(d) Differentiate between a TCP header and a SCTP header. [5]

(A)

Points	TCP header	SCTP header
Source port	**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.	**Source port address.** This is a 16-bit field that defines the port number of the process sending the packet.
Destination port	**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.	**Destination port address.** This is a 16-bit field that defines the port number of the process receiving the packet.
Check Sum	This 16-bit field contains the checksum.	**Checksum.** This 32-bit field contains a CRC-32 checksum (see Appendix D). Note that the size of the checksum is increased from 16 bits (in UDP, TCP, and IP) to 32 bits in SCTP to allow the use of the CRC-32 checksum.
Diagram		

Q.5 Attempt any TWO question of the following : [10]

Q.5(a) What are the types of TFTP messages? What is the purpose of each one? [5]

(A) TFTP Messages

There are five types of TFTP messages, RRQ, WRQ, DATA, ACK, and ERROR, as shown in Figure 1.

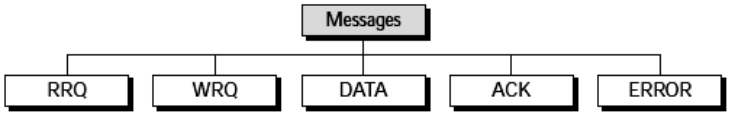


Fig.1 : Message categories

RRQ

The read request (RRQ) message is used by the client to establish a connection for reading data from the server.

Fig.2 : RRQ format

WRQ

The write request (WRQ) message is used by the client to establish a connection for writing data to the server. The format is the same as RRQ except that the OpCode is 2.

Fig.3 : WRQ format

DATA

The data (DATA) message is used by the client or the server to send blocks of data.

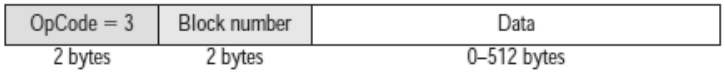


Fig.4 : Data format

ACK

The acknowledge (ACK) message is used by the client or server to acknowledge the receipt of a data block. The message is only 4 bytes long.

Fig.5 : ACK format

ERROR

The ERROR message is used by the client or the server when a connection cannot be established or when there is a problem during data transmission. It can be sent as a negative response to RRQ or WRQ. It can also be used if the next block cannot be transferred during the actual data transfer phase. The error message is not used to declare a damaged or duplicated message. These problems are resolved by error-control mechanisms.

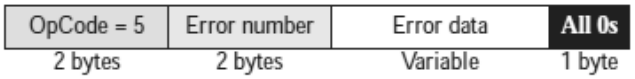


Fig.6 : ERROR format

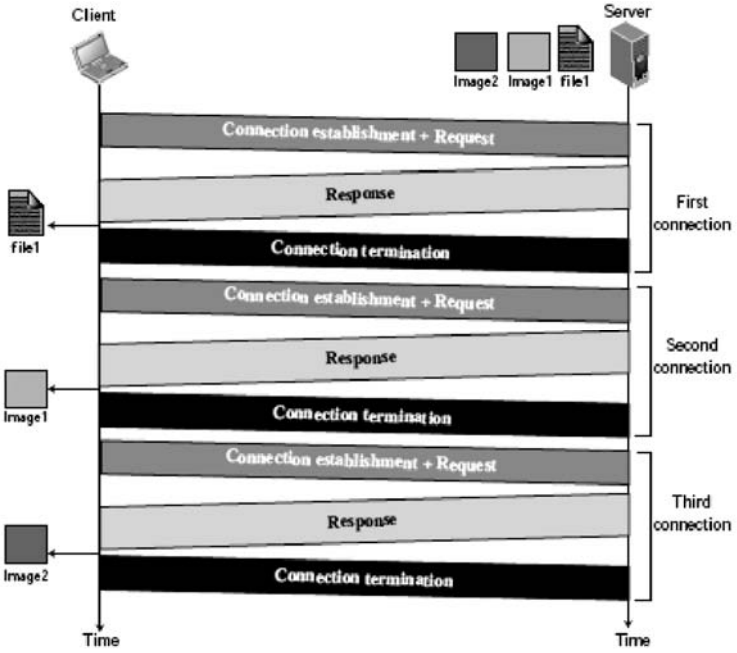
Q.5(b) Explain Persistence and non-persistence connection of HTTP. [5]

(A) Non-persistent Connection

In this, one TCP connection is made for each request/response. steps

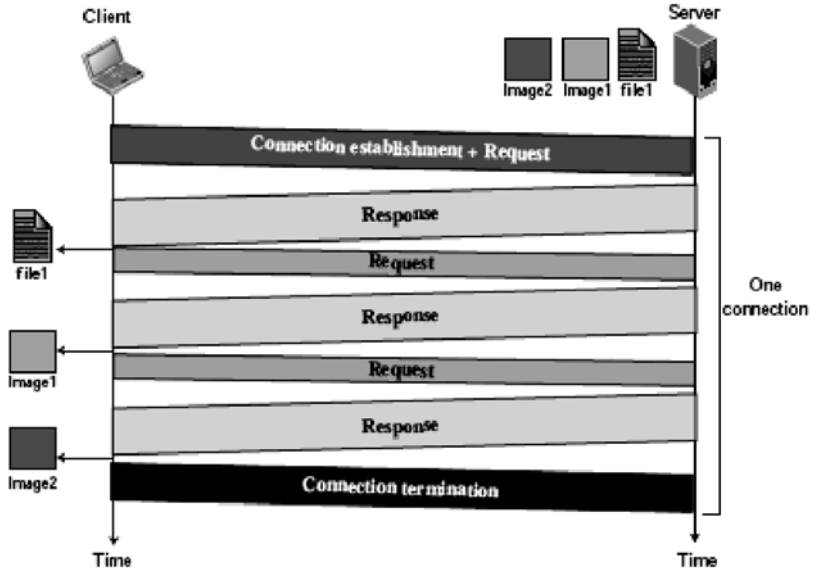
(1) The client opens a TCP connection and sends a request.

(2) The server sends the response and closes the connection.
(3) The client reads the data until it encounters an end-of-file marker; it then closes the connection.



Persistent Connection

The server leaves the connection open for more requests after sending a response. The server can close the connection at the request of the client or if a time-out has been reached.



Q.5(c) Describe NVT character set for option negotiation. **[5]**

(A) **Option Negotiation** : To use any of the options first requires option negotiation between the client and the server. Four control characters are used for this purpose; these are shown in Table 1.

Table 1 : NVT character set for option negotiation

Character	Code	Meaning 1	Meaning 2	Meaning 3
WILL	251	Offering to enable	Accepting to enable	
WONT	252	Rejecting to enable	Offering to disable	Accepting to disable
DO	253	Approving to enable	Requesting to enable	
DONT	254	Disapproving to enable	Approving to disable	Requesting to disable

Enabling an Option : Some options can only be enabled by the server, some only by the client, and some by both. An option is enabled either through an offer or a request.

Offer to Enable : A party can offer to enable an option if it has the right to do so. The offering can be approved or disapproved by the other party. The offering party sends the WILL command, which means "Will I enable the option?" The other party sends either the DO command, which means "Please do," or the DONT command, which means "Please don't." See Figure 1.

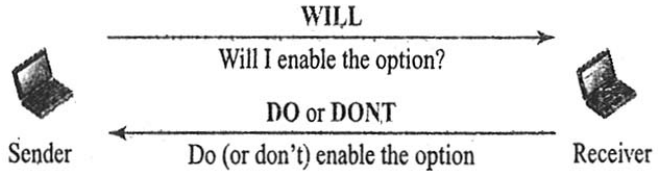


Fig. 1 : Offer to enable an option

Request to Enable : A party can request from the other party the enabling of an , option. The request can be accepted or refused by the other party. The requesting party sends the DO command, which means "Please do enable the option." The other party sends either the WILL command, which means "I will," or the WONT command, which means "I won't." See Figure 2.

Disabling an Option : An option that has been enabled can be disabled by one of the parties. An option is disabled either through an offer or a request.

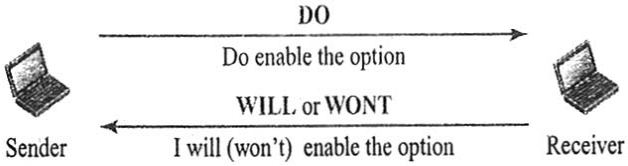


Fig. 2 : Request to enable an option

Offer to Disable : A party can offer to disable an option. The other party must approve the offering; it cannot be disapproved. The offering party sends the WONT command, which means "I won't use this option anymore." The answer must be the DONT command, which means "Don't use it anymore." Figure 3 shows an offer to disable an option.

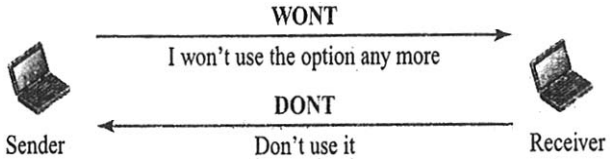


Fig. 3 : Offer to disable an option

Request to Disable : A party can request from another party the disabling of an option. The other party must accept the request; it cannot be rejected. The requesting party sends the DONT command, which means "Please don't use this option anymore." The answer must be the WONT command, which means "I won't use it anymore." Figure 4 shows a request to disable an option.

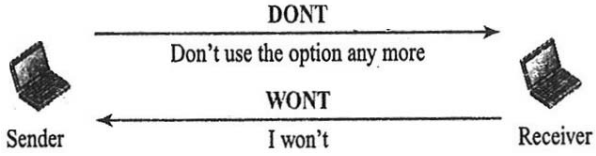


Fig. 4 : Request to disable an option

Q.5(d) List and explain the components of SSH. [5]

(A) SECURE SHELL (SSH)

Another popular remote login application program is Secure Shell (SSH). SSH, like TELNET, uses TCP as the underlying transport protocol, but SSH is more secure and provides more service than TELNET.

Components

SSH is a proposed application–layer protocol with four components, as shown in Figure.

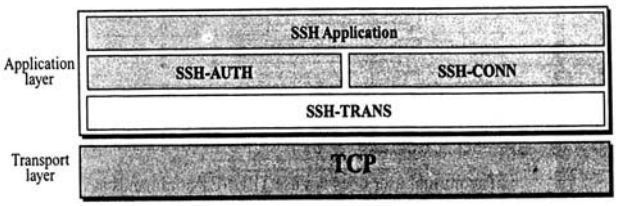


Fig. : Components of SSH.

SSH Transport–Layer Protocol (SSH–TRANS)

- Since TCP is not a secured transport layer protocol, SSH first uses a protocol that creates a secured channel on the top of TCP.
- This new layer is an independent protocol referred to as SSH–TRANS.
- When the software implementing this protocol is called, the client and server first use the TCP protocol to establish an insecure proconnection.
- Then they exchange several security parameters to establish a secure channel on the top of the TCP.
- The services provided by this protocol:
  - Privacy or confidentiality of the message exchanged.
  - Data integrity, which means that it is guaranteed that the messages exchanged between the client and server are not changed by an intruder.
  - Server authentication, which means that the client is now sure that the server is the one that it claims to be.
  - Compression of the messages that improve the efficiency of the system and makes attack more difficult.

SSH Authentication Protocol (SSH–AUTH)

After a secure channel is established between the client and the server and the server is authenticated for the client, SSH can call another software that can authenticate the client for the server.

SSH Connection Protocol (SSH–CONN)

After the secured channel is established and both server and client are authenticated for each other, SHH can call a piece of software that implements the third protocol, SSH–CONN. One of the services by the SSH–CONN protocol is to do multiplexing. SSH–CONN takes the secure channel established by the two previous protocols and lets the client create multiple logical channels over it.

Q.6 Attempt any TWO question of the following : [10]

Q.6(a) Explain the following email scenarios with the help of diagrams: [5]

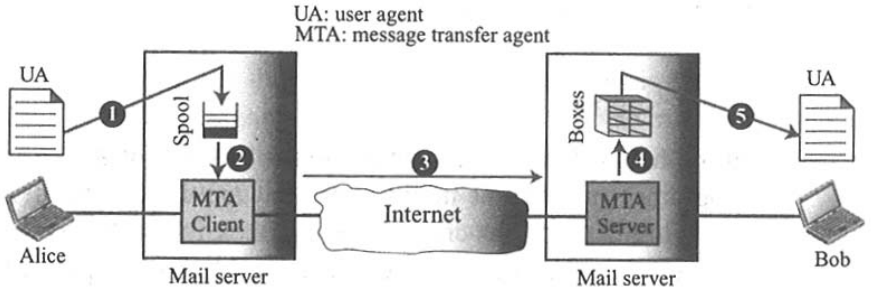
(i) When the sender and the receiver of an e-mail are on different mail servers.

(ii) When sender is connected to the mail server via a LAN or a WAN,

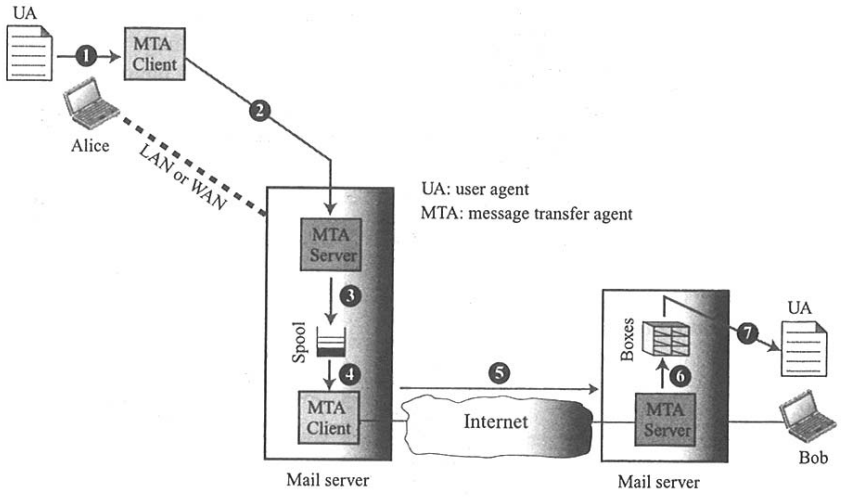
(A) (i) Second scenario :

The sender and the receiver of the e-mail are users on 2 different mail servers. The message needs to be sent over the internet. We need user agents (VAS) and message transfer agents (MTAS). Alice needs to use a user agent program to send her message

to the mail server at her own site. The mail server at site uses sequence to store messages waiting to be sent. Bob also needs a user agent program to retrieve messages stored in the mail box of the system at this site. When the sender two message transfer agents are needed : one client and one server. Like most client–server programs on the internet, the server needs to run all of the time because it does not know when a client will ask for a connection. The client, on the other hand, can be triggered by the system when there is a message in the queue to be sent.

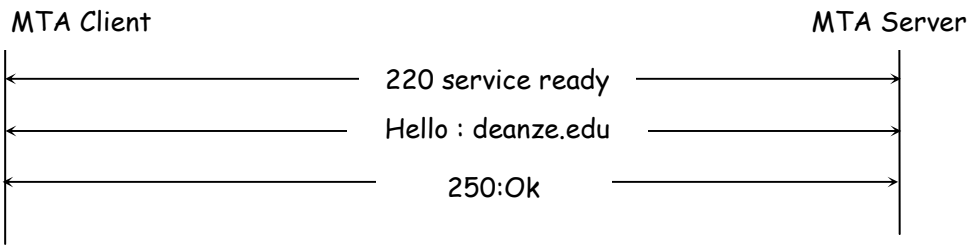


(ii) Third Scenario : Since Alice is separated from her mail server. Alice is either connected to the mail server via point–to–point WAN–such as a dial up modem a DSL or a cable modem-or she is connected to a LAN in an organisation that uses one mail server for handling emails. She is then needs to send the message through the LAN or WAN. Whenever Alice has a message to send, she calls the user agent which in turn, calls the MTA client. The MTA client established a connection with the MTA server on the system which is running all the time. The system at Alice site queues all messages received. It then uses an MTA client to send the messages to the system at Bob's site. The system receives the message & stores it in Bob's mailbox.

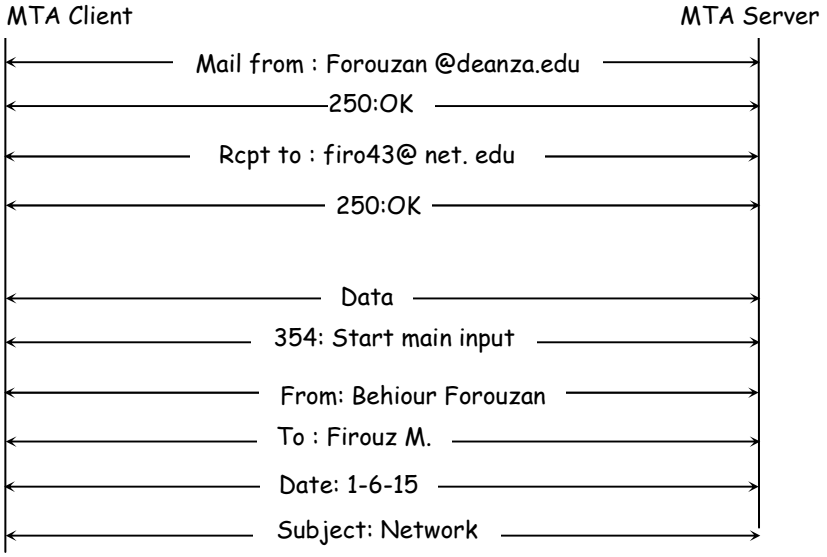


Q.6(b) Explain the phases of mail transfer. **[5]**

(A) (i) Connection Establishment : After a client has made a TCP connection to the well-known port 25, the SMTP server starts the connection phase.

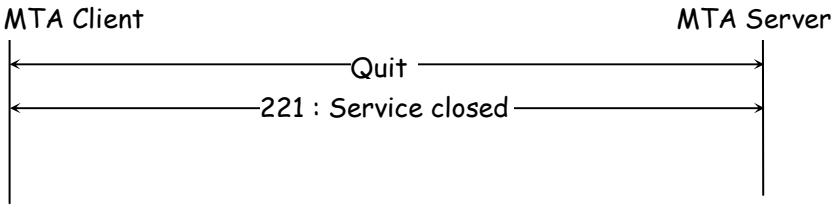


(ii) Message Transfer : A single message between a sender and one or more recipients can be exchanged.



Connection Termination

After the message is transferred successfully, the client terminates the connection.



Q.6(c) Write the functions of IMAP4 that are not carried out by POP3. [5]

(A) IMAP4 provides the following extra functions which are not carried out by POP3:

- A user can check the e–mail header prior to downloading.
- A user can search the contents of the e–mail for a specific string of characters prior to downloading.
- A user can partially download e–mail. This is especially useful if bandwidth is limited and the e–mail contains multimedia with high bandwidth requirements.
- A user can create, delete, or rename mailboxes on the mail server.
- A user can create a hierarchy of mailboxes in a folder for e–mail storage.

Q.6(d) Explain the two techniques used for traffic shaping to improve Quality of Service. [5]

(A)

- **Leaky Bucket.**
  - If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket. The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty. The input rate can vary, but the output rate remains constant.
  - Similarly, in networking, a technique called **leaky bucket** can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate. Figure 1 shows a leaky bucket and its effects.
  - In the figure, we assume that the network has committed a bandwidth of 3 Mbps for a host. The use of the leaky bucket shapes the input traffic to make it conform to this commitment. In Figure 25.32 the host sends a burst of data at a rate of 12 Mbps for 2 s, for a total of 24 Mbits of data. The host is silent for 5 s and then sends data at a rate of 2 Mbps for 3 s, for a total of 6 Mbits of data. In all, the host has sent 30 Mbits of data in 10 s. The leaky bucket smooths the traffic by sending out data at a rate of 3 Mbps during the same 10 s. Without the leaky bucket, the beginning burst may have hurt the network by consuming more bandwidth than is set aside for this host. We can also see that the leaky bucket may prevent congestion. As an analogy, consider the freeway during rush hour (bursty

traffic). If, instead, commuters could stagger their working hours, congestion on our freeways could be avoided.

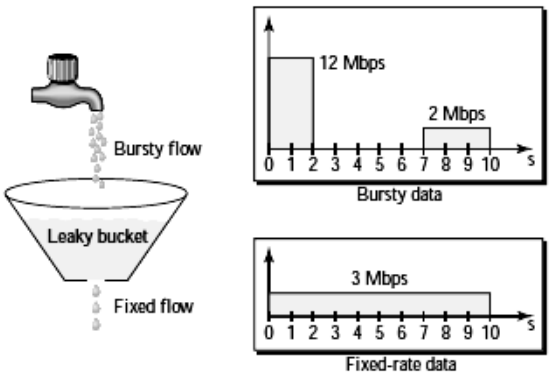


Fig. 1 : Leaky bucket

- A simple leaky bucket implementation is shown in Figure 2. A FIFO queue holds the packets. If the traffic consists of fixed-size packets (e.g., cells in ATM networks), the process removes a fixed number of packets from the queue at each tick of the clock. If the traffic consists of variable-length packets, the fixed output rate must be based on the number of bytes or bits.

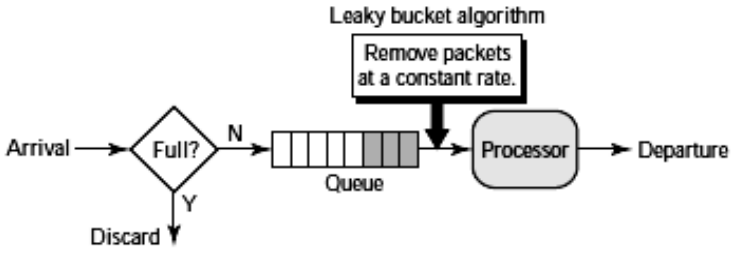


Fig. 2 : Leaky bucket implementation

The following is an algorithm for variable-length packets:

1. Initialize a counter to *n* at the tick of the clock.
2. If *n* is greater than the size of the packet, send the packet and decrement the counter by the packet size. Repeat this step until *n* is smaller than the packet size.
3. Reset the counter and go to step 1.

- **Token Bucket :** The leaky bucket is very restrictive. It does not credit an idle host. For example, if a host is not sending for a while, its bucket becomes empty. Now if the host has bursty data, the leaky bucket allows only an average rate. The time when the host was idle is not taken into account. On the other hand, the token bucket algorithm allows idle hosts to accumulate credit for the future in the form of tokens. For each tick of the clock, the system sends *n* tokens to the bucket. The system removes one token for every cell (or byte) of data sent. For example, if *n* is 100 and the host is idle for 100 ticks, the bucket collects 10,000 tokens. Now the host can consume all these tokens in one tick with 10,000 cells, or the host takes 1,000 ticks with 10 cells per tick. In other words, the host can send bursty data as long as the bucket is not empty.

  The token bucket can easily be implemented with a counter. The token is initialized to zero. Each time a token is added, the counter is incremented by 1. Each time a unit of data is sent, the counter is decremented by 1. When the counter is zero, the host cannot send data.

  Combining Token Bucket and Leaky Bucket : The two techniques can be combined to credit an idle host and at the same time regulate the traffic. The leaky bucket is applied after the token bucket; the rate of the leaky bucket needs to be higher than the rate of tokens dropped in the bucket.

Q.7 Attempt any THREE question of the following : [15]

Q.7(a) Explain the extension headers in IPv6. [5]

(A) (i) Hop-by-Hop Option

The hop-by-hop option is used when the source needs to pass information to all routers visited by the datagram. Routers must be informed about certain management, debugging, or control functions. Or, if the length of the datagram is more than the usual 65,535 bytes.

<table>
<tr><td colspan="2">VER Traffic class</td><td colspan="2">Flow label</td></tr>
<tr><td colspan="2">Payload Length</td><td>Next</td><td>Hop limit</td></tr>
<tr><td colspan="4">Source Address</td></tr>
<tr><td colspan="4">Destination Address</td></tr>
<tr><td>Next Header</td><td>Header Length</td><td colspan="2"></td></tr>
<tr><td colspan="4"></td></tr>
<tr><td>Next Header</td><td>Header Length</td><td colspan="2"></td></tr>
<tr><td colspan="4"></td></tr>
<tr><td>Next Header</td><td>Header Length</td><td colspan="2"></td></tr>
<tr><td colspan="4"></td></tr>
</table>

(ii) Fragmentation

In IPv6, only the original source can fragment. A source must use a Path MTU Discovery technique to find the smallest MTU supported by any network on the path. The source then fragments the datagram using this knowledge.

(iii) Authentication

It validates the message sender and ensures the integrity of data. The former is needed so the receiver can be sure that a message is from the genuine sender and not from an imposter.

(iv) Destination option:

It is used when source needs to pass information to the destination only. Intermediate routers are not permitted access to the information.

(v) Source Routing :

The source routing extension header combines the concepts of the strict source route check that the data is not altered in transition by same hacker.

(vi) Encrypted Security Payload

The encrypted security payload (ESP) is an extension that provides confidentiality and guards against eavesdropping.

Q.7(b) Explain the timers used in Routing Information Protocol. [5]

(A) Timers in Routing Information Protocol

- RIP uses three times to support its operation (see Figure). The periodic timer controls the sending of messages, the expiration timer governs the validity of a route, and the garbage collection timer advertises the failure of a route.

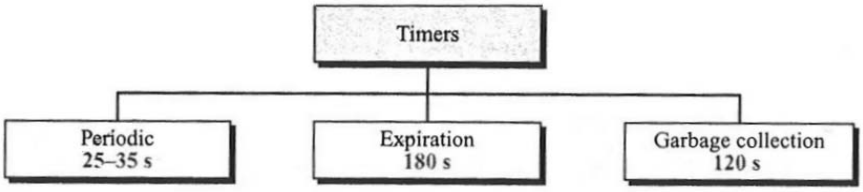


Fig. : RIP timers

Periodic Timer

- The periodic timer controls the advertising of regular update messages. Although the protocol specifies that this timer must be set to 30 s. the working model uses a random number between 25 and 35 s. This is to prevent any possible synchronization and therefore overload on an internet if routers update simultaneously.
- Each router has one periodic timer that is randomly set to a number between 25 and 35. It counts down; when zero is reached, the update message is sent, and the timer is randomly set once again.

Expiration Timer

- The expiration timer governs the validity of a route. When a router receives update information for a route, the expiration timer is set to 180 s for that particular route. Every time a new update for the route is received, the timer us reset. In normal situations this occurs every 30 s. However, if there is a problem on an internet and no update is received within the allotted 180 s, the route is considered expired and the hop count of the route is set to 16, which means the destination is unreachable. Every route has its own expiration timer.

Garbage Collections Timer

- When the information about a route becomes invalid, the router does not immediately purge that route from its table. Instead, it continues to advertise the route with a metric value of 16. At the same time, a timer called the garbage collection timer is set to 120 s for that route. When the count reaches zero, the route is purged from the table. This timer allows neighbors to become aware of the invalidity of a route prior to purging.

Q.7(c) List and explain various features of TCP. [5]

(A) Various features
Numbering system
Flow control
Error control
Congestion control

Numbering System

Although the TCP software keeps track of the segment being transmitted or received, there is no field for a segment number value. Instead, there are two fields called the sequence number and the acknowledgment number. These two fields refer to the byte number not the segment number.

Byte Numbers

TCP numbers all data bytes that are transmitted in a connection. Numbering is independent in each direction. When TCP receives bytes of data from the process and stores them in the sending buffer, it numbers them. The numbering does not necessarily start from 0 ; it starts randomly. TCP generates a random number between 0 and 2 - 1 for the number of the first byte. For example, if the random number happens to be 1,057 and the total data to be sent is 6,000 bytes, the bytes are numbered from 1,057 to 7,056. We will see that byte numbering is used for flow and error control.

The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number.

Sequence Number

After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent. The sequence number for each segment is the number of the first byte carried in that segment.

Acknowledgment Number

Communication in TCP is full duplex; when a connection is established, both parties can send and receive data at the same time. Each party numbers the bytes, usually with a different starting byte number. The sequence number in each direction shows the number of the first byte carried by the segment. Each party also uses an acknowledgment number to confirm the bytes it has received. However, the acknowledgment number defines the number of the next byte that the party expects to receive. In addition, the acknowledgment number is cumulative, which means that the party takes the number of the last byte that it has received, safe and sound, adds 1 to it, and announces this sum as the acknowledgment number. The term cumulative here means that if a party uses 5,643 as an acknowledgment number, it has received all bytes from the beginning up to 5,642. Note that this does not mean that the party has received 5,642 bytes because the first byte number does not have to start from 0.

Flow control

TCP provides flow control. The sending TCP controls how much data can be accepted from the sending process; the receiving TCP controls how much data can be sent by the sending TCP. This is done to prevent the receiver from being overwhelmed with data. the numbering system allows TCP to use a byte oriented flow control.

Error Control

To provide reliable service, TCP implements an error control mechanism. Although error control considers as the unit of data for error detection, error control is byte-oriented.

Congestion Control

TCP takes into account congestion in the network. The amount of data sent by the sender is not only controlled by the receiver, but is also determined by the level of congestion, if any, in the network.

Q.7(d) Explain the features of Stream Control Transmission Protocol. [5]

(A) SCTP Features

Transmission Sequence Number (TSN)

- The unit of data in TCP is a byte. Data transfer in TCP is controlled by numbering bytes using a sequence number.
- On the other hand, the unit of data in SCTP is a data chunk, which may or may not have a one-to-one relationship with the message coming from the process because of fragmentation.
- Data transfer in SCTP is controlled by numbering the data chunks.
- SCTP uses a transmission sequence number (TSN) to number the data chunks.
- In other words, the TSN in SCTP plays the analogous role as the sequence number in TCP.
- TSNs are 32 bits long and randomly initialized between 0 and 232 - 1. Each data chunk must carry the corresponding TSN in its header.
- In SCTP, a data chunk is numbered using a TSN.

Stream Identifier (SI)

- In TCP, there is only one stream in each connection.
- In SCTP, there may be several streams in each association. Each stream in SCTP needs to be identified using a stream identifier (SI).
- Each data chunk must carry the SI in its header so that when it arrives at the destination, it can be properly placed in its stream.
- The SI is a 16-bit number starting from 0.
- To distinguish between different streams, SCTP uses an SI.

Stream Sequence Number (SSN)

- When a data chunk arrives at the destination SCTP, it is delivered to the appropriate stream and in the proper order.
- This means that, in addition to an SI, SCTP defines each data chunk in each stream with a stream sequence number (SSN).
- To distinguish between different data chunks belonging to the same stream, SCTP uses SSNs.

Packets

- In TCP, a segment carries data and control information.
- Data are carried as a collection of bytes; control information is defined by six control flags in the header.
- The design of SCTP is totally different: data are carried as data chunks, control information as control chunks.
- Several control chunks and data chunks can be packed together in a packet.
- A packet in SCTP plays the same role as a segment in TCP. Figure 2 compares a segment in TCP and a packet in SCTP.
- TCP has segments; SCTP has packets.

Acknowledgment Number

- TCP acknowledgment numbers are byte-oriented and refer to the sequence numbers.
- SCTP acknowledgment numbers are chunk-oriented.
- They refer to the TSN.
- A second difference between TCP and SCTP acknowledgments is the control information. This information is part of the segment header in TCP.
- In SCTP, acknowledgment numbers are used to acknowledge only data chunks; control chunks are acknowledged by other control chunks if necessary.

Flow Control

- SCTP implements flow control to avoid overwhelming the receiver.

Error Control

- SCTP implements error control to provide reliability. TSN numbers and acknowledgment numbers are used for error control.

Congestion Control

- SCTP implements congestion control to determine how many data chunks can be injected into the network.

Q.7(e) What is the concept of out–of-band signaling? [5]

(A)

- To make control characters effective in special situations, TELNET uses out-of-band signaling.
- In out-of-band signaling, the control characters are preceded by IAC and are sent to the remote process.
- Imagine a situation in which an application program running at the server site has gone into an infinite loop and does not accept any more input data. The user wants to interrupt the application program, but the program does not read data from the buffer. The TCP at the server site has found that the buffer is full and has sent a segment specifying that the client window size should be zero. In other words, the TCP at the server site is announcing that no more regular traffic is accepted. To remedy such a situation, an urgent TCP segment should be sent from the client to the server. The urgent segment overrides the regular flow-control mechanism. Although TCP is not accepting normal segments, it must accept an urgent segment.
- When a TELNET process (client or server) wants to send an out-of-band sequence of characters to the other process (client or server), it embeds the sequence in the data

stream and inserts a special character called a DM (data mark). However, to inform the other party, it creates a TCP segment with the urgent bit set and the urgent pointer pointing to the DM character. When the receiving process receives the data, it reads the data and discards any data preceding the control characters (IAC and IP, for example). When it reaches the DM character, the remaining data are handled normally. In other words, the DM character is used as a synchronization character that switches the receiving process from the urgent mode to the normal mode and resynchronizes the two ends (see Figure).

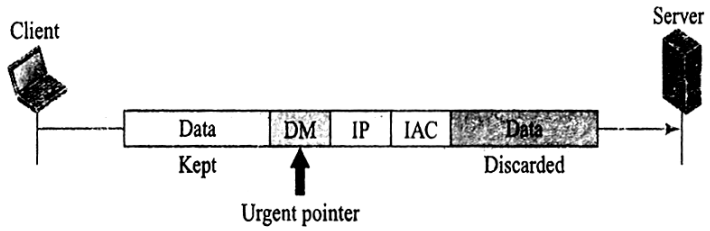


Fig. : Out-of-band signaling

In this way, the control character (IP) is delivered out of band to the operating system, which uses the appropriate function to interrupt the running application program.

Q.7(f) List and explain the Positive Completion Responses of SMTP. **[5]**

(A)

Code	*Description*
	Positive Completion Reply
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded

❑ ❑ ❑ ❑ ❑