B.TECH.

SIXTH SEMESTER EXAMINATION, 2006-2007

COMPUTER NETWORKING

Time – 3 hours | Total Marks – 100

Note: (i) Attempt *ALL* questions.
(ii) All questions carry equal marks.
(iii) In case of numerical problems assume data wherever not provided.
(iv) Be precise in your answer.

Q. 1. Attempt any four part of the following– (4×5=20)

Q.(a). What is Signal to Noise Ratio required to achieve Channel capacity of 20 Mbps with 3 MHz Bandwidth?

Ans. According to question

Singal to Noise ratio, $\frac{S}{N} = ?$

Data rate, $r = 20 \times 10^6$ bps

Bandwidth, $B = 3 \times 10^6$ Hz

We know that

$$r = 2B \log \left(1+\frac{S}{N}\right)$$

$$20 \times 10^6 = 2 \times 3 \times 10^6 \log_2 \left(1+\frac{S}{N}\right)$$

$$10 = 3 \log_2 \left(1+\frac{S}{N}\right)$$

$$\left(1+\frac{S}{N}\right) = 2^{3.33}$$

$$\frac{S}{N} = 2^{3.33} - 1 \quad \text{Ans.}$$

Q.1.(b). Differentiate the followings–

(i) Base band and Broadband

(ii) Synchronous and asynchronous transmission.

Ans. (i) Base Band and Broadband

Baseband Signalling:

1) Uses digital signalling
2) No frequency-division multiplexing
3) Bi-directional transmission
4) Signal travels over short distances

Broadband Signalling:

1) Uses analog signalling
2) Unidirectional transmission
3) Frequency-division multiplexing is possible
4) Signal can travel over long distances before being attenuated

(ii) Asynchronous and Synchronous Transmission

Asynchronous transmission uses start and stop bits to signify the beginning bit ASCII character would actually be transmitted using 10 bits e.g.: A "0100 0001" would become "1 0100 0001 0". The extra one (or zero depending on parity bit) at the start and end of the transmission tells the receiver first that a character is coming and secondly that the character has ended. This method of transmission is used when data is sent intermittently as opposed to in a solid stream. In the previous example the start and stop bits are in bold. The start and stop bits must be of opposite polarity. This allows the receiver to recognize when the second packet of

information is being sent.

Synchronous transmission uses no start and stop bits but instead synchronizes transmission speeds at both the receiving and sending end of the transmission using clock signals built into each component. A continual stream of data is then sent between the two nodes. Due to there being no start and stop bits the data transfer rate is quicker although more errors will occur, as the clocks will eventually get out of sync, and the receiving device would have the wrong time that had been agreed in protocol (computing) for sending/receiving data, so some bytes could become corrupted (by losing bits). Ways to get around this problem include re-synchronization of the clocks and use of check digits to ensure the byte is correctly interpreted and received.

Q.1.(c). Compare and Contrast the twisted pair and Coaxial cable and optical fibre transmission medium.

Ans. Compare Contrast Twisted pair, Coaxial Cable and Optical fiber

Twisted pair cabling is a form of wiring in which two conductors are wound together for the purposes of canceling out electromagnetic interference (EMI) from external sources, electromagnetic radiation from the UTP cable, and crosstalk between neighboring pairs.

Twisting wires decreases interference because the loop area between the wires is reduced. In balanced pair operation, the two wires typically carry equal and opposite signals (differential mode) which are combined by addition at the destination. The common-mode noise from the two wires (mostly) cancel each other in this addition because the two wires have similar amounts of EMI that are 180 degrees out of phase. This results in the same effect as subtraction. Differential mode also reduces electromagnetic radiation from the cable, along with the attenuation that it causes.

Coaxial cable is an electrical cable consisting of a round conducting wire, surrounded by an insulating spacer, surrounded by a cylindrical conducting sheath, usually surrounded by a final insulating layer (jacket). It is used as a high-frequency transmission line to carry a high-frequency or broadband signal. Because the electromagnetic field carrying the signal exists (ideally) only in the space between the inner and outer conductors, it cannot interfere with or suffer interference from external electromagnetic fields.

Optical fiber (or "fiber optic") refers to the medium and the technology associated with the transmission of information as light pulses along a glass or plastic wire or fiber. Optical fiber carries much more information than conventional copper wire and is in general not subject to electromagnetic interference and the need to retransmit signals. Most telephone company long-distance lines are now of optical fiber.

Transmission on optical fiber wire requires repeaters at distance intervals. The glass fiber requires more protection within an outer cable than copper. A type of fiber known as single mode fiber is used for longer distances; multimode fiber fiber is used for shorter distances.

Q.1.(d). With the help of neat diagram explain the working principal and merits and demerits of the following topologies

(i) Token Bus

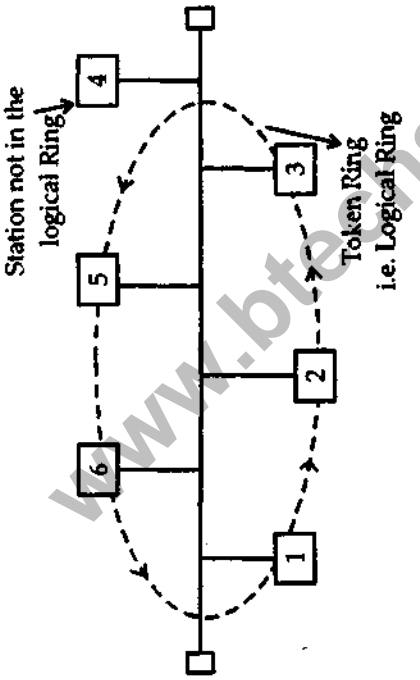
(ii) Star Topology

Ans. (i) Token Bus

Token bus is a network implementing the

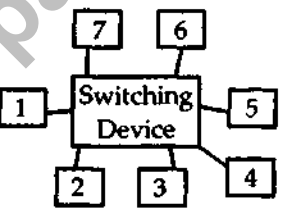
token ring protocol over a "virtual ring" on a coaxial cable. A token is passed around the network nodes and only the node possessing the token may transmit. If a node doesn't have anything to send, the token is passed on to the next node on the virtual ring. Each node must know the address of its neighbour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

Token bus was standardized by the IEEE 802.4 Working Group. It is mainly used for industrial applications. Token bus was used by GM (General Motors) for their Manufacturing Automation Protocol (MAP) standardization effort.



(ii) Star Topology

It is type of network topology in which each of the nodes of the network is connected to a central node with a point-to-point link in a 'hub' and 'spoke' fashion, the central node being the 'hub' and the nodes that are attached to the central node being the 'spokes' (e.g., a collection of point-to-point links from the peripheral nodes that converge at a central node) – all data that is transmitted between nodes in the network is transmitted to this central node, which is usually some type of device that then retransmits the data to some or all of the other nodes in the network, although the central node may also be a simple common connection point (such as a 'punch-down' block) without any active device to repeat the signals.



Q.1.(e). A frame is consisting two characters each of two bits long is transmitted over the communication channel having the bit error rate of 10^{-3}. Give the probability at receiver containing at least one error.

Ans. According to question

Bit error rate $B_r = 10^{-3}$

Frame size = 2 characters $= 2 \times 2$

$= 4$ bits

Probability at receiver $= 4 \times 10^{-3} = 0.004$

Ans.

Q.1.(f). A digital signal system is required to operate at 9600 bps, if the signal element encodes a 4 bit word, what is the minimum required bandwidth of the channel?

Ans. According to question

Bandwidth of channel, W = ?

Data rate, r = 9600 bps.

Signal level, V = 4

We know that nyquist rate

$r = 2\omega \log_2 V$

(∵ ω = band width of channel)

$9600 = 2\omega \log_2 4 = 2\ w \log_2 2^2$

$= 4\omega \cdot 1$

$\omega = \frac{9600}{4} = 2400$ Hz

$\omega = 2.4$ KHz

Q. 2. Attempt any four of the following— (4×5=20)

Q.2.(a). Measurement of slotted ALOHA channel with an infinite number of users show that 10 percent of the slots are idle—

(i) What is the channel load?

(ii) What is the throughput?

Ans. According to question

idle slot = 10% = 0.1

So, busy slot = offered load

= 90% = 0.9

G = 0.9

Thus (i) Cannel load = offered load

(ii) Throughput for slotted aloha

$S = Ge^{-G}$

$= 0.9 * (2.713)^{-0.9}$

$S = \frac{0.9}{(2.713)^{0.9}}$

$= \frac{0.9}{2.455} = 0.3665$

Thus S = 36.65%

Q.2.(b). For each of the statement given below state whether it is true or false. If there is any ambiguity in the question then you may justify your answer in one or two sentences otherwise just give TRUE or FALSE—

(a) Aloha performance is not dependent on a.

(b) Random access protocols can be made stable using back-off parameters.

(c) Selective Reject protocol will have, on an average, lesser retransmission than GO Back N protocol when there are frame errors.

(d) If the probability of a bit received with an error on a link is b and L is the packet length in bits, then the probability that the packet is received without error is $(1-(1-b^L)$.

(e) Ethernet can provide deterministic delay guarantee to its packets.

Ans. (a) False; Depend on offered load.

(b) True.

(c) False; Because due to error in frame , same frame must be retransmitted again.

(d) True.

(e) False; Because it uses CSMA/CD protocol and Bus topology.

Q.2.(c). Consider builiding a CSMA/CD network running at 1 Gbps over a 1-km cable with no repeaters. The signal spped in the cable is 2,00,000 km/s. What is the minimum frame size?

Ans. As per question

Data rate, $r = 1 \times 10^9$ bps

Cable length $d = 1 \times 10^3$ m

Signal speed $v = 2 \times 10^5$ km/sec

Thus, time taken inreaching bit at receiving node $t = \frac{d}{v} = \frac{1\times10^3}{2\times10^8}$

$t = 5 \times 10^{-6}$ sec.

Thus, as per CSMA/CD protocol

Frame size = data rate × t

$= 1 \times 10^9$ bps $\times 5 \times 10^{-6}$ sec

$= 5 \times 10^3$ bits

Q.2.(d). A system has to be transmit a message 1110111101 over a communication link using the polynomial generator shown in Fig. 1. Determine the message that should be transmitted.

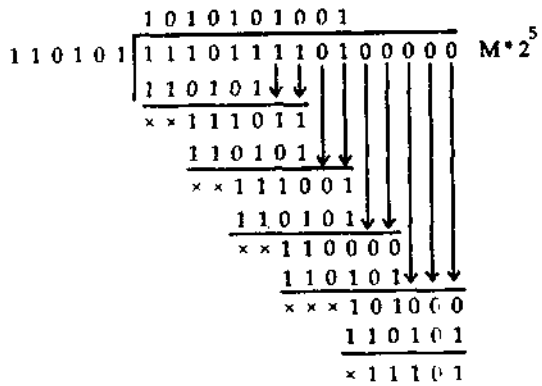
Ans. As per quesion

given message, m = 1110111101

golynomial generator is calculated through given circuit is $p(x) = x^5 + x^4 + x^2 + 1$

p = 110101

so, by CRC method



Thus, transmitted bit is given by

$T = M * 2^5 + R$

$T = 111011110111101$ **Ans.**

Q.2.(e). A radio station is using 14.44 Kbps channel for message transmission and the sending message packets are 100 bits long. Calculate the maximum throughput for the channel using slotted aloha.

Ans. According to question

Data rate, $r = 14.44 \times 10^3$ bps.

Frame length = 100 bits

Throughput $S = ?$

Now, Slot size is given by,

$$\Delta t = \frac{100}{14.44 \times 10^3} = \frac{1}{144.4}$$

$$= 6.92 \times 10^{-3} \text{ sec}$$

So, offered load

$$G = K\Delta T = 100 \times 6.92 \times 10^{-3}$$

$$G = 0.692$$

Throughput

$$S = Ge^{-G} = \frac{0.692}{(2.713)^{0.692}} = \frac{0.692}{1.9950}$$

$$S = 0.3468$$

$$S = 34.68\%$$

Q.2.(f). Explain with the help of neat diagram the basic and extended frame associated with HDLC protocol. Also mention the control fields and sub fields of each case.

Ans. HDLC frame, including the flag, are

Flag		8 bits
FCS		16 or 32 bits
Information		Variable length, 0 or more bits, in multiples of 8
Control		8 or 16 bits
Address		8 bits
Flag		8 bits

Note that the end flag of one frame can be (but does not have to be) the beginning (start) flag of the next frame. Note that the data comes in groups of 8 bits. The FCS is the Frame Check Sequence, and is a more sophisticated version of the parity bit. The field contains the result of a binary calculation that uses the bit sequences that make up the 'Address', 'Control' and 'Information' fields.

Frame Types

I-Frames (user data)

I frames transport user data from the network layer. In addition they can also include flow and error control information piggybacked on data. The subfields in the control field define these functions.

S-Frames (control)

Supervisory Frames are used for flow and error control whenever piggypacking is impossible or inappropriate , for example when primary field has to send only command or response or acknowledge and not data. S-frame do not have information fields.

The first 2 bits (10) mean it is S-frame

U-Frames

U-Frames are used for link management. They exchange session management and control information between connected devices.

U-frames contain an information field used for system management information and not user data.

Q. 3. Attempt any two of the following: (2×10=20)

Q.3.(a). A network consisting of five routers is shown below. The cost of the link connecting the routers is mentioned along with the link. If the network is running using link state routing protocol, given the updates node. A will produce and propel and to whom?

Ans. The updated node is given by

F	Seq.	Age	C	5	2
			E		

E	Seq.	Age	F	2	1
			D		

D	Seq.	Age	C	3	1	1	2
			E			A	B

C	Seq.	Age	D	3	5	3	5
			F			B	A

B	Seq.	Age	C	3	2	2
			D			A

A	Seq.	Age	B	2	5	1
			C		D	

A will propel to B, C and D.

Q.3.(b). Consider a router in an IP network with the following table–

Subnet Number	Subnet Mask	Next HOP
203.187.152.0	255.255.248.0	Interface 0
203.187.128.0	255.255.252.0	R2
203.96.0.0	255.255.192.0	R3
203.187.130.0	255.255.254.0	R5
Default		R4

Find the next hops for the following addresses–

(i) 203.187.131.25

(ii) 203.96.130.186

(iii) 203.187.155.138

(iv) 203.96.16.234

(v) 203.187.129.146

Ans. Next hops are

(i) R_5

(ii) R_4

(iii) Interface 0

(iv) R_3

(v) R_2

Q.3.(c). Answer the following related to Internet protocol–

(i) Explain the Header and Routing table explosion limitation associated with IPv4.

(ii) What are the extension headers available in IPv6?

(iii) Describe with example the use of following IPv6 Address schemes

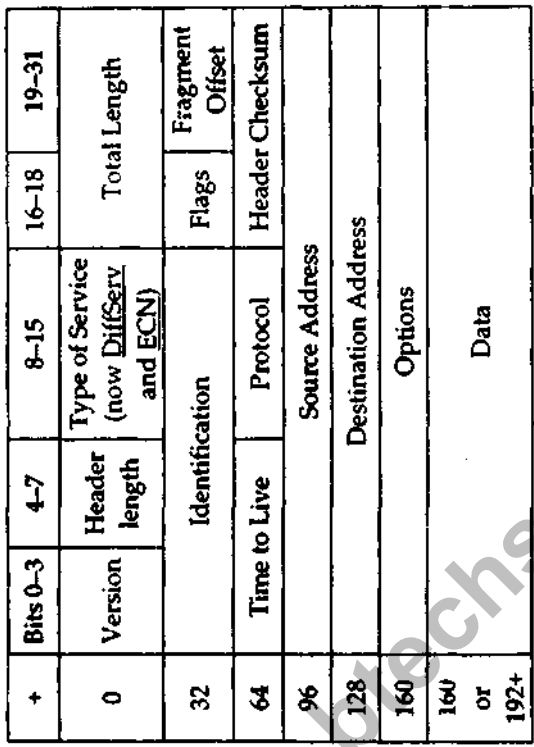
(a) IPv4 Mapped and IPv4 Compatible addressing.

(b) Link local and Site local addressing.

Ans. (i) Header

The header consists of 13 fields, of which only 12 are required. The 13th field is optional (red background in table) and aptly named: options. The fields in the header are packed with the most significant byte first (big endian), and for the

diagram and discussion, the most significant bits are considered to come first. The most significant bit is numbered 0, so the version field is actually found in the 4 most significant bits of the first byte, for example.



Version

The first header field in an IP packet is the 4-bit version field. For IPv4, this has a value of 4 (hence the name IPv4).

Internet Header Length (IHL)

The second field is a 4-bit Internet Header Length (IHL) telling the number of 32-bit words in the header. Since an IPv4 header may contain a variable number of options, this field specifies the size of the header (this also coincides with the offset to the data). The minimum value for this field is 5 (rfc791), which is a length of 5×32 = 160 bits. Being a 4-bit field the maximum length is 15 words or 480 bits.

Type of Service (TOS)

The following 8 bits were allocated to a Type of Service (TOS) field:

- bits 0-2: precedence
- bit 3: 0 = Normal Delay, 1 = Low Delay
- bit 4: 0 = Normal Throughput, 1 = High Throughput
- bit 5: 0 = Normal Reliability, 1 = High Reliability
- bits 6-7: Reserved for future use

This field is now used for DiffServ and ECN. The original intention was for a sending host to specify a preference for how the datagram would be handled as it made its way through an internetwork. For instance, one host could set its IPv4 datagrams' TOS field value to prefer low delay, while another might prefer high reliability. In practice, the TOS field has not been widely implemented. However, a great deal of experimental, research and deployment work has focused on how to make use of these eight bits.

Total Length

This 16-bit field defines the entire datagram size, including header and data, in bytes. The minimum-length datagram is 20 bytes (20 bytes header + 0 bytes data) and the maximum is 65,535 — the maximum value of a 16-bit word. The minimum size datagram that any host is required to be able to handle is 576 bytes, but most modern hosts handle much larger packets. Sometimes subnetworks impose further restrictions on the size, in which case datagrams must be fragmented. Fragmentation is handled in either the host or packet switch in IPv4 .

Identification

This field is an identification field and is

primarily used for uniquely identifying fragments of an original IP datagram. Some experimental work has suggested using the ID field for other purposes, such as for adding packet-tracing information to datagrams in order to help trace back datagrams with spoofed source addresses.

Flags

A 3-bit field follows and is used to control or identify fragments. They are (in order, from high order to low order):

- Reserved; must be zero.
- Don't Fragment (DF)
- More Fragments (MF)

If the DF flag is set and fragmentation is required to route the packet then the packet will be dropped. This can be used when sending packets to a host that does not have sufficient resources to handle fragmentation.

When a packet is fragmented all fragments have the MF flag set except the last fragment, which does not have the MF flag set. The MF flag is also not set on packets that are not fragmented — clearly an unfragmented packet can be considered the last fragment.

Fragment Offset

The fragment offset field, measured in units of 8-byte blocks, is 13-bits long and specifies the offset of a particular fragment relative to the beginning of the original unfragmented IP datagram. The first fragment has an offset of 0. This allows a maximum offset of 65,528 () which would exceed the maximum IP packet length of 65,535 with the header length included.

Time To Live (TTL)

An 8-bit time to live (TTL) field helps prevent datagrams from persisting (e.g. going in circles) on an inter network. Historically the TTL field limited a datagram's lifetime in seconds, but has come to be a hop count field. Each packet switch (or router) that a datagram crosses decrements the TTL field by one. When the TTL field hits zero, the packet is no longer forwarded by a packet switch and is discarded. Typically, an ICMP message (specifically the time exceeded) is sent back to the sender that it has been discarded. The reception of these ICMP messages is at the heart of how traceroute works.

Protocol

This field defines the protocol used in the data portion of the IP datagram. The Internet Assigned Numbers Authority maintains a list of Protocol numbers. Common protocols and their decimal values are shown below Header

Checksum

The 16-bit checksum field is used for error-checking of the header. At each hop, the checksum of the header must be compared to the value of this field. If a header checksum is found to be mismatched, then the packet is discarded. Note that errors in the data field are up to the encapsulated protocol to handle — indeed, both UDP and TCP have checksum fields.

Source address

An IP address is a group of 4 8-bit octets for a total of 32 bits. The value for this field is determined by taking the binary value of each octet and concatenating them together to make a single 32-bit value.

Destination address

Identical to the source address field but indicates the receiver of the packet.

Options

Additional header fields may follow the destination address field, but these are not often used. Note that the value in the IHL field must include enough extra 32-bit words to hold all

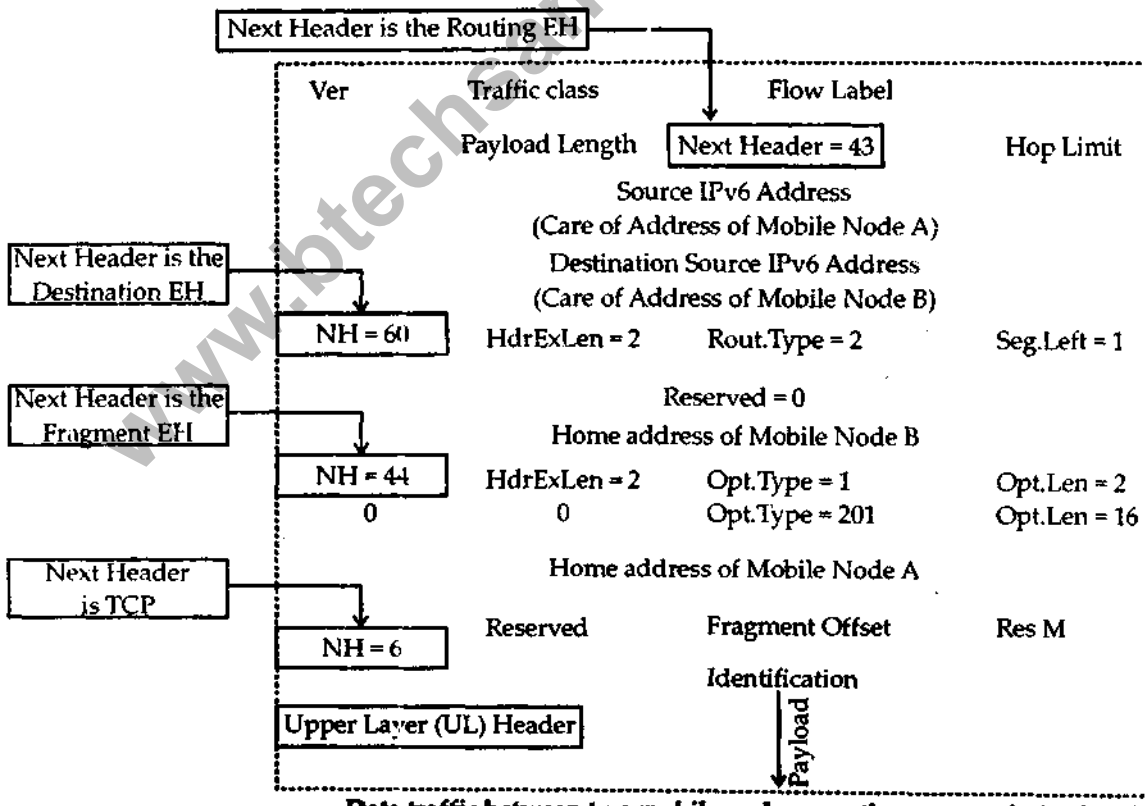
the options (plus any padding needed to ensure that the header contains an integral number of 32-bit words). The list of options may be terminated with an EOL (End of Options List) option; this is only necessary if the end of the options would not otherwise coincide with the end of the header.

(ii) Extension Header available in IPv6

Commonly Used Extension Headers

The Extension Header should not be viewed as an esoteric feature of IPv6 that would be encountered only at later stages of the network and service deployment. Extension headers are an intrinsic part of the IPv6 protocol and they support some basic functions and certain services. The following is a list of circumstances where EHs are commonly used:

- Hop-by-Hop EH is used for the support of Jumbo-grams or, with the Router Alert option, it is an integral part in the operation of MLD. Router Alert [3] is an integral part in the operations of IPv6 Multicast through Multicast Listener Discovery (MLD) and RSVP for IPv6.
- Destination EH is used in IPv6 Mobility as well as support of certain applications.
- Routing EH is used in IPv6 Mobility and in Source Routing. It may be necessary to disable "IPv6 source routing" on routers to protect against DDoS.
- Fragmentation EH is critical in support of communication using fragmented packets (in IPv6, the traffic source must do fragmentation-routers do not perform fragmentation of the packets they forward)



Data traffic between two mobile nodes over the route optimized path

• Mobility EH is used in support of Mobile IPv6 service

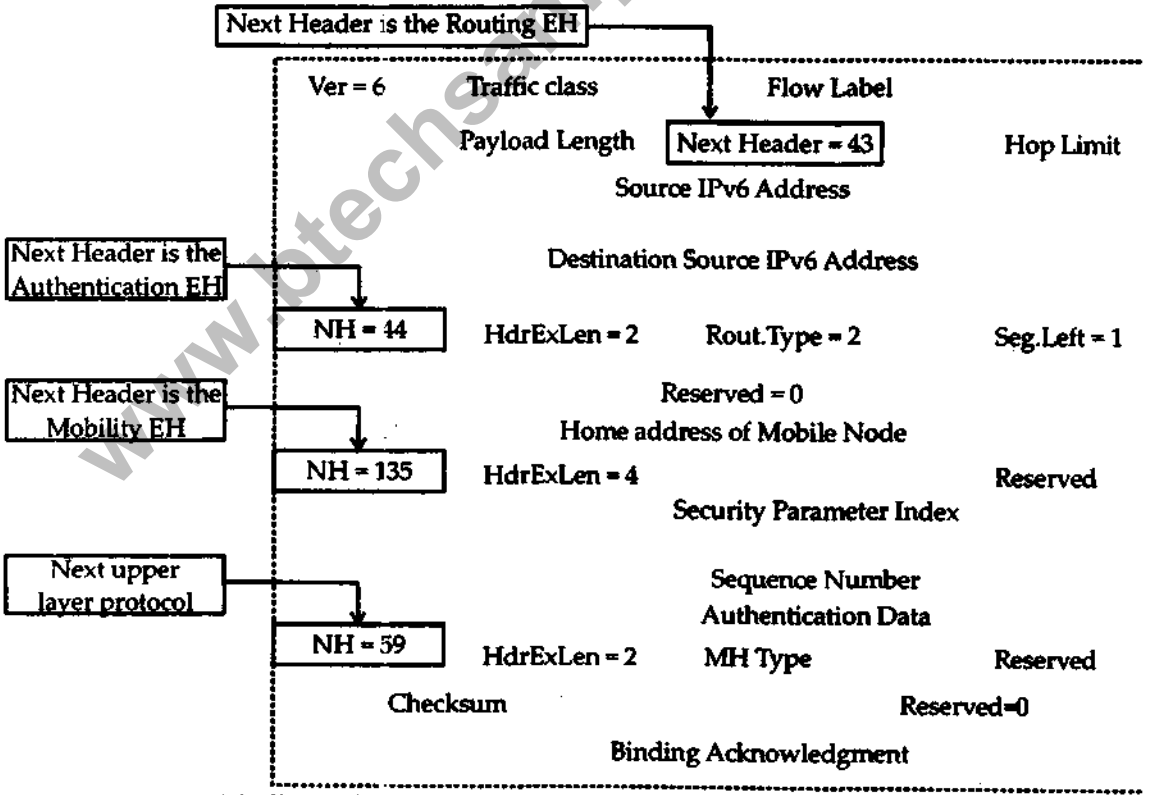
• Authentication EH is similar in format and use to the IPv4 authentication header defined in RFC2402 [4].

• Encapsulating Security Payload EH is similar in format and use to the IPv4 ESP header defined in RFC2406 [5]. All information following the Encapsulating Security Header (ESH) is encrypted and for that reason, it is inaccessible to intermediary network devices. The ESH can be followed by an additional Destination Options EH and the upper layer datagram.

Figure 3 presents the structure of IPv6 data plane packets using extension headers. In this example, the packet is sent from Mobile Node A to Mobile Node B over the route optimized path [6], hence the use of the Routing EH (43) and the Destination Options EH (60). It is sent over a path that has an Maximum Transmission Unit (MTU) smaller than that of Mobile Nodes (MNs) access link, hence the use of the Fragmentation EH (44).

The length of the Extension Header chain in this case is 56 bytes. By comparison, the IPv6 packet between a Mobile Node and Correspondent Node over the route optimized path would have either the Routing or the Destination Options EH (not both) leading to a shorter EH chain.

To exemplify the case of an IPv6 packet with a longer EH chain, Figure 4 depicts the structure of a Binding Acknowledgement sent from the Home Agent to the Mobile Node. The first header



Binding Acknowledgment Sent from a Correspondent Node to a Mobile Node

in the chain is the Routing EH (43), the second is the Authentication EH (51) and finally the Mobility EH (135).

This packet, built to emphasize special cases where multiple extension headers might be used, has an EH chain length of 72 bytes. Since the length of most individual extension headers is variable, the length of EH chains can be even larger. Note however that this size is typically driven by the need to carry certain information in addition to that in the main header. The larger EH chains (whether due to many EH or to long individual EH) are used for control plane traffic. Packets carrying user data such as the one showed in Figure 3 generally have shorter EH chains.

The common use of IPv6 EH makes it important to analyze and understand the way in which network devices (routers, layer 3 switches and generally devices that forward traffic based on layer 3 information) process the extension headers.

IPV6 EXTENSION HEADERS PROCESSING

This section describes the way in which various Extension Header types must be processed by network devices under basic forwarding conditions or in the context of advanced features such as Access Lists. It identifies the protocol requirements that must be observed.

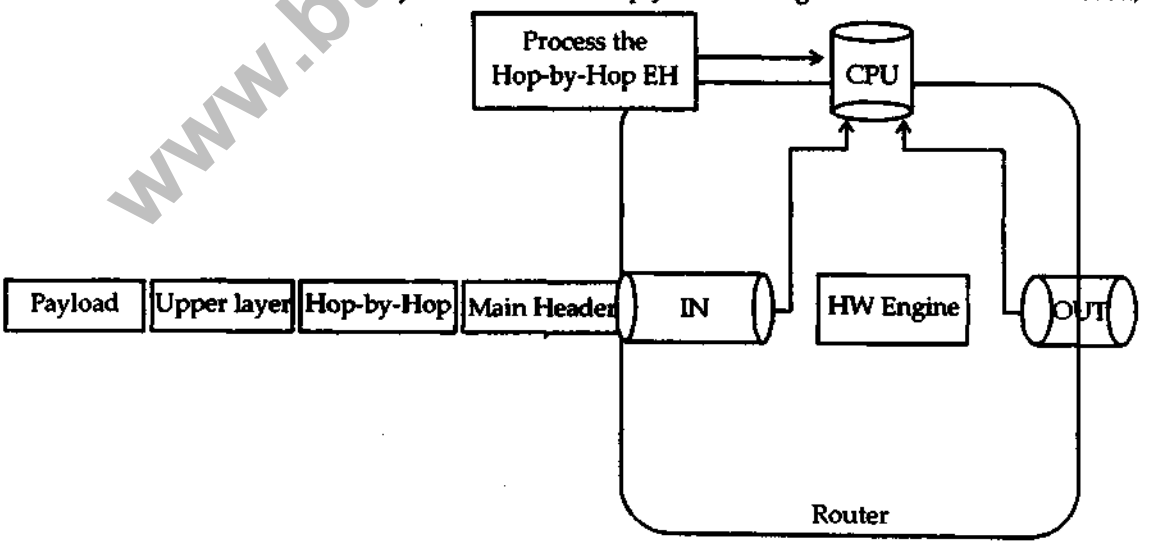
Hop-by-Hop Extension Header

The Hop-by-Hop Extension Header is the ONLY EH that MUST be fully processed by all network devices as shown in Figure 5. From this perspective, the Hop-by-Hop EH is similar to the IPv4 options. This explains the reason why this EH MUST be the first in a chain of extension headers.

Because the Hop-by-Hop EH must be fully processed, it is handled by the CPU1[1] and the IPv6 traffic that contains a Hop-by-Hop EH will go through the slow forwarding path. This rule applies to all vendors. Hardware forwarding is not feasible in this case.

Other Extension Headers

Network devices are not required to process any of the other IPv6 extension headers when simply forwarding the traffic. For this reason,



Forwarding IPv6 Packets with the Hop-by-Hop Extension Header

IPv6 traffic with one or more EHs other than Hop-by-Hop can be forwarded in hardware as shown in Figure.

Network devices might however process some EHs if specifically configured to do so while supporting certain services such as IPv6 Mobility.

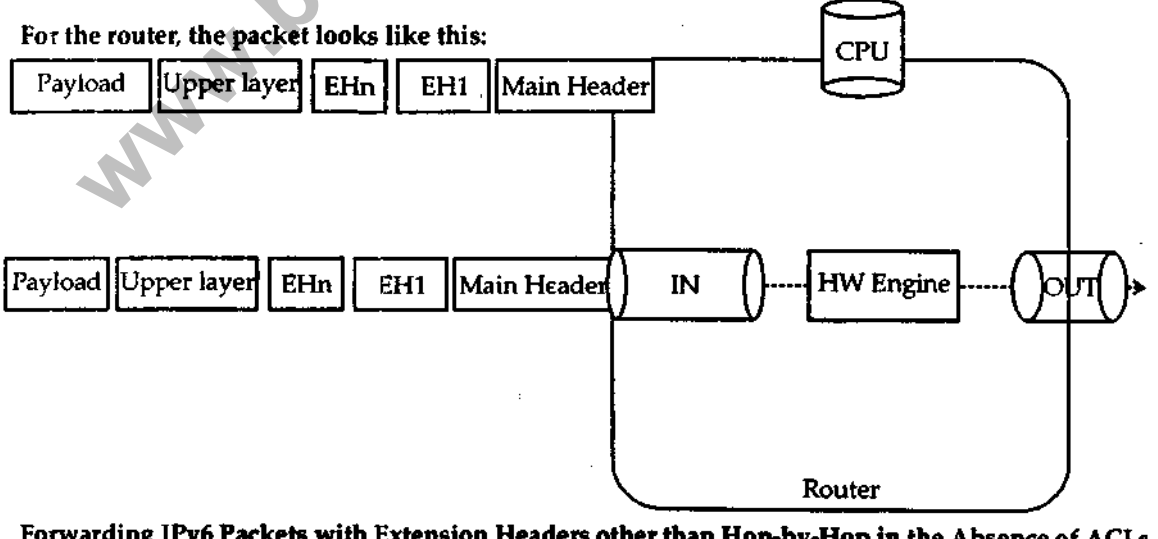
The extensions headers used to secure the IP communication between two hosts, Authentication and Encapsulating Security Payload Headers, are also ignored by the intermediary network devices while forwarding traffic. These EHs are relevant only to the source and destination of the IP packet. It is important however to remember that all information following the ESH is encrypted and not available for inspection by an intermediary device, if that is required.

Extension Headers and Access Lists

In the absence of the Hop-by-Hop EH, based on the processing rules described in the previous Hop-by-Hop Extension Header and Other Extension Headers sections, as long as a router is concerned exclusively with layer 3 information and it is not specifically instructed to process certain EH (for certain services it is supporting), it can forward IPv6 traffic without analyzing the extension headers. An IPv6 packet can have an arbitrary number of EH (other than Hop-by-Hop) and the router would ignore them and simply forward the traffic based on the main header. Under these conditions, routers can forward the IPv6 traffic in hardware despite the EHs. Access Lists (ACL) applied on router interfaces however, can change the router's IPv6 forwarding performance characteristics when extension headers are present.

(a) IPv4 Mapped addressing

IPv4 mapped IPv6 addresses constitute a special class of IPv6 addresses. Such an IPv6 address has its first 80 bits set to zero, the next 16 set to one, while its last 32 bits represents an IPv4 address. For example, ::ffff:c000:280 is the mapped IPv6 address for 192.0.2.128.



Forwarding IPv6 Packets with Extension Headers other than Hop-by-Hop in the Absence of ACLs

Notation

As a special exception to IPv6 addresses notation, IPv4 mapped addresses are commonly represented with their last 32 bits notated as an IPv4 address. As such, ::ffff:c0C0:280 would typically be notated ::ffff:192.0.2.128 instead.

Usage

IPv4 mapped addresses are normally used by the IP stack to represent IPv4 addresses to IPv6 applications. It allows the transparent use of transport layer protocols (TCP or UDP) over IPv4 through the IPv6 networking API. It is therefore considered an IPv6 transition mechanism for dual-stack hosts.

The big advantage of this mechanism is in allowing a server application to only use a single listening socket to handle connections from client via both IPv6 and IPv4 protocols. In that case, IPv6 clients will be handled as usual, and IPv4 clients appear as IPv6 clients with an adequate mapped IPv6 address. It can also be used to establish IPv4 connections actively with an IPv6 socket, but that feature is rather rarely used.

While the actual packets on the network will be IPv4, the logical connection will be presented as an IPv6 one to the application.

Limitations

Because of the significant internal differences between IPv4 and IPv6 at the network layer, some of the lower level functionality that may be exposed by the IPv6 stack might not work with IPv4 mapped addresses, if there is no direct translation to IPv4.

(b) IPv4 Compatible addressing:

IPv4-compatible IPv6 addresses constitute a special class of IPv6 addresses. Such an IPv6 address has its first 96 bits set to zero, while its last 32 bits represents an IPv4 address.

IPv6 transition mechanisms no longer use IPv4-compatible addresses. The only remaining use of these addresses is to represent an IPv4 address in a table with fixed size members that must also be able to store an IPv6 address.

The undefined IPv6 address :: and the loopback IPv6 address ::1 are not really IPv4-compatible addresses, even though they are included in the IPv6 address space ::/96.

Q. 4. Attempt any two of the following— **(2×10=20)**

Q.4.(a). Describe the Nagle's alogrithm and Karl – Partridge algorithm and mention their merits and demerits.

Ans. (a) Nagle's Algorithm

Nagle's algorithm is a means of improving the efficiency of TCP/IP networks by reducing the number of packets that need to be sent over the network. Nagle's algorithm works by coalescing a number of small outgoing messages, and sending them all at once. Specifically, as long as there is a sent packet for which the sender has received no acknowledgment, the sender should keep buffering its output until it has a full packet's worth of output, so that output can be sent all at once.

Algorithm

if there is new data to send

if the window size >= MSS and available data is >= MSS

send complete MSS size segment now

else

if there is unconfirmed data still in the pipe enqueue data in the buffer until an acknowledge is received

else

send data immediately

where MSS = Maximum segment size

This algorithm interacts badly with TCP delayed acknowledgments, a feature introduced into TCP at roughly the same time in the early 1980s, but by a different group. With both algorithms enabled, applications which do two successive writes to a TCP connection, followed by a read, experience a constant delay of up to 500 milliseconds, the "ACK delay". For this reason, TCP implementations usually provide applications with an interface to disable the

Nagle algorithm. This is typically called the TCP_NODELAY option. The first major application to run into this problem was the X Window System.

Karl – Partridge algorithm

One problem that occurs with the dynamic estimation of RTT (Round Trip time) is what to do when a segment times out and is sent again. When the acknowledgement comes in, it is unclear whether the acknowledgement refers to the first transmission or a later one. Guessing wrong can seriously contaminate the estimate of RTT. Phl Karn discovered this problem the hard way. He is an amateur radio enthusiast interested in transmitting TCP/IP packets by ham radio, a notoriously unreliable medium. He made a simple proposal : do not update RTT on any segments that have been retransmitted. Instead the timeout is doubled on each failure until the segments get through the first time. This fix is called Karn's algorthm.

Most TCP implements use it.

Q.4.(b). Assume for the TCP adaptive retransmission machenism the Estimate RTT is 4.0 at some point and subsequent measurement RTT all are 1.0. How long does it take before the time out value as calculated by the Jacobson/ Karels algorithms fall below 4.0? Assume initial value of deviation 20, $\delta = 1/8$, $\alpha = 1$, $\beta = 4$.

Ans. We know that, by Jacobson/Karels algorithm

$$\text{Timeout} = \text{RTT} + 4 \times D \quad \ldots(i)$$

where $D = \alpha D_0 + (1 - \alpha)|RTT - M|$

Here $\alpha = 1$, $D = D_0 + 0 = D_0 = 20$

($\because$ intial value of deviation $D_0 = 20$)

Here from (i)

Time out $= 1 + (4 \times 20) = 81$ **Ans.**

Q.4.(c). What are the services provided by the transport layer? Explain with the help of neat diagram the three way handshake process for the connection establishment done by TCP prior to send the data over the transmission channel.

Ans. Services provided by Transport Layer

(i) Connection-oriented Service: connection-oriented service is built on top of network layer that in the Transport layer. It works in three phases. , Connection establishment, data transfer and connection termination.

(ii) Same Order Delivery : the Transport layer provides it. The simplest way of doing this is to give each packet a number, and allow the receiver to reorder the packets.

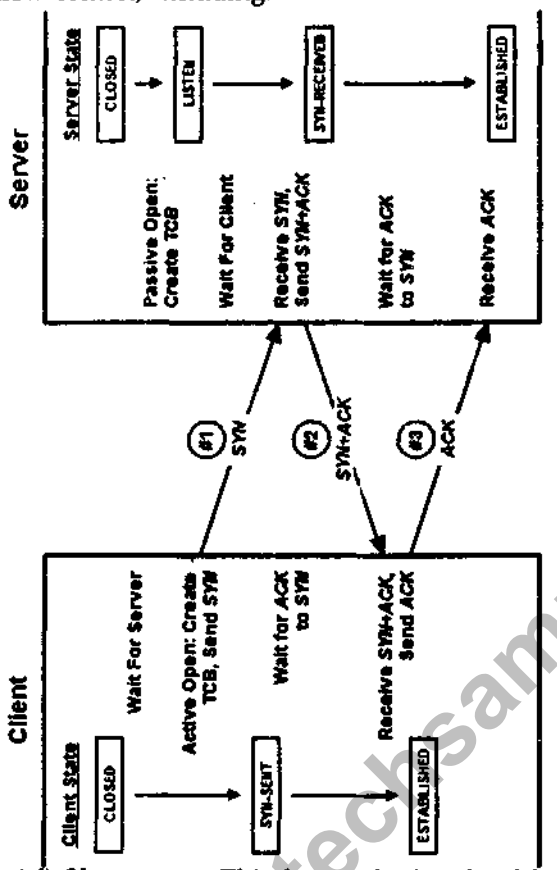
(iii) Reliable Data : Packets may be lost in routers, switches, bridges and hosts due to network congestion, when the packet queues are filled and the network nodes have to delete packets. Packets may be lost or corrupted in for example Ethernet due to interference and noise, since Ethernet does not retransmit corrupt packets. Packets may be delivered in the wrong order by an underlying network. Some transport layer protocols, for example TCP, can fix this. By means of an error detection code, for example a checksum, the transport protocol may check that the data is not corrupted, and verify that by sending an ACK message to the sender. Automatic repeat request schemes may be used to retransmit lost or corrupted data. By introducing segment numbering in the transport layer packet headers, the packets can be sorted in order. Of course, error free is impossible, but it is possible to substantially reduce the numbers of undetected errors.

(iv) Flow Control : The amount of memory on a computer is limited, and without flow control a larger computer might flood a computer with so much information that it can't hold it all before dealing with it. Nowadays, this is not a big issue, as memory is cheap while bandwidth is comparatively expensive, but in earlier times it was more important. Flow control allows the receiver to say "Whoa!" before it is overwhelmed. Sometimes this is already provided by the network, but where it is not, the Transport layer may add it on.

(v) Congestion avoidance : Network congestion occurs when a queue buffer of a network node is full and starts to drop packets. Automatic repeat request may keep the network in a congested state. This situation can be

TCP "Three-Way Handshake" Connection Establishment Procedure					
Client			**Server**		
Start State	**Action**	**Move To State**	**Start State**	**Action**	**Move To State**
CLOSED	The client cannot do anything until the server has performed a passive *OPEN* and is ready to accept a connection. (Well, it can try, but nothing will be accomplished until the server is ready.)	—	*CLOSED*	The server performs a passive *OPEN*, creating a transmission control block (TCB) for the connection and readying itself for the receipt of a connection request (*SYN*) from a client.	*LISTEN*
CLOSED	**Step #1 Transmit:** The client performs an active OPEN, creating a transmission control block (TCB) for the connection and sending a *SYN* message to the server.	*SYN-SENT*	*LISTEN*	The server waits for contact from a client.	—
SYN-SENT	The client waits to receive an *ACK* to the *SYN* it has sent, as well as the server's *SYN*.	—	*LISTEN*	**Step #1 Receive, Step #2 Transmit:** The server receives the *SYN* from the client. It sends a single *SYN+ACK* message back to the client that contains an *ACK* for the client's *SYN*, and the server's own *SYN*.	*SYN-RECEIVED*
CLOSED	**Step #1 Transmit:** The client performs an active OPEN, creating a transmission control block (TCB) for the connection and sending a *SYN* message to the server.	*SYN-SENT*	*LISTEN*	The server waits for contact from a client.	—
SYN-SENT	The client waits to receive an *ACK* to the *SYN* it has sent, as well as the server's *SYN*.	—	*LISTEN*	**Step #1 Receive, Step #2 Transmit:** The server receives the *SYN* from the client. It sends a single *SYN+ACK* message back to the client that contains an *ACK* for the client's *SYN*, and the server's own *SYN*.	*SYN-RECEIVED*
SYN-SENT	**Step #2 Receive, Step #3 Transmit:** The client receives from the server the *SYN+ACK* containing the *ACK* to the client's *SYN*, and the *SYN* from the server. It sends the server an *ACK* for the server's *SYN*. The client is now done with the connection establishment.	*ESTABLI-SHED*	*SYN-RECEIVED*	The server waits for an *ACK* to the *SYN* it sent previously.	—
ESTABLISHED	The client is waiting for the server to finish connection establishment so they can operate normally.		*SYN-RECEIVED*	**Step #3 Receive:** The server receives the *ACK* to its *SYN* and is now done with connection establishment.	*ESTABLISH-ED*
ESTABLISHED	The client is ready for normal data transfer operations.		*ESTABLIS HED*	The server is ready for normal data transfer operations.	

avoided by adding congestion avoidance to the flow control, including.



(vi) Slow-start : This keeps the bandwidth consumption at a low level in the beginning of the transmission, or after packet retransmission.

(vii) Byte orientation : Rather than dealing with things on a packet-by-packet basis, the Transport layer may add the ability to view communication just as a stream of bytes. This is nicer to deal with than random packet sizes, however, it rarely matches the communication model which will normally be a sequence of messages of user defined sizes.

(viii) Ports : Ports are essentially ways to address multiple entities in the same location. For example, the first line of a postal address is a kind of port, and distinguishes between different occupants of the same house. Computer applications will each listen for information on their own ports, which is why you can use more than one network-based application at the same time.

Part II

Three Way handshake process for connection establishment done by TCP:

It is adapted from the table describing the TCP finite state machine, but shows what happens for both the server and the client over time. Each row shows the state the device begins in, what action it takes in that state and the state to which it transitions. The transmit and receive parts of each of the three steps of the handshake process are shown in the table.

TCP "Three-Way Handshake" Connection Establishment Procedure

This diagram illustrates how a conventional connection is established between a client and server, showing the three messages sent during the process and how each device transitions from the CLOSED state through intermediate states until the session is ESTABLISHED.

Q. 5. Attempt any two of the following— **(10×2=20)**

Q.5.(a). What is socket? Explain with the help of diagram using one application the client server communication using TCP socket.

Ans. Socket

It is a communication end-point unique to a machine communicating on an Internet Protocol-based network, such as the Internet.

An Internet socket is composed of the following:

- Protocol (TCP, UDP, raw IP)
- Local IP address
- Local port
- Remote IP address
- Remote port

The remote address can be any valid IP address, or 0.0.0.0 for a listening socket, or 255.255.255.255 for a broadcasting socket.

Operating systems combine sockets with a running process or processes (which use the socket to send and receive data over the network), and a transport protocol (i.e. TCP or UDP) with which the process(es) communicate to the remote host.

Usually sockets are implemented over TCP but this is not required. They can be implemented over any transport protocol, such as SNA [1]. The concept of a socket is an entity that implements an API, regardless of the implementation.

Two widely used Internet socket types are:
1. Datagram sockets, which use UDP
2. Stream Sockets, which use TCP

Part II

Simultaneous Open Connection Establishment in TCP Socket

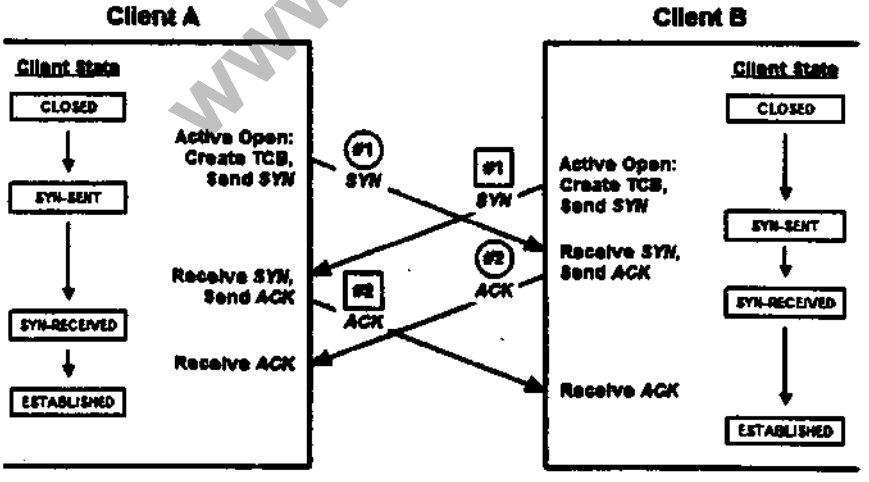
TCP is also set up to handle the situation where both devices perform an active OPEN instead of one doing a passive OPEN. This may occur if two clients are trying to reach each other instead of a client and a server. It is uncommon, however, and only happens under certain circumstances. Simultaneous connection establishment can also only happen if a well-known port is used as the source port for one of the devices.

In this case, the steps are different for both devices. Each client will perform an active OPEN and then proceed through both the SYN-SENT and SYN-RECEIVED states until their SYNs are acknowledged. This means there isn't a "three-way handshake" any more as shown in Table 152. Instead, it is like two simultaneous "two-way handshakes". Each client sends a SYN, receives the other's SYN and ACKs it, and then waits for its own ACK. The transaction, simplified, is described in Table and Figure.

Q.5.(b). Answer the following related to DNS—

TCP Simultaneous Open Connection Establishment Procedure

Client A			Client B		
Start State	Action	Move To State	Start State	Action	Move To State
CLOSED	Client A Step #1 Transmit: Client A performs an active OPEN, creating a TCB and sending a SYN message to the server.	SYN-SENT	CLOSED	Client B Step #1 Transmit: Client B performs an active OPEN, creating a TCB and sending a SYN to the server.	SYN-SENT
SYN-SENT	Client B Step #1 Receive and Step #2 Transmit: Client A receives Client B's SYN and sends it an ACK. It is still waiting for an ACK to its own SYN.	SYN-RECEIVED	SYN-SENT	Client A Step #1 Receive and Step #2 Transmit: Client B receives Client A's SYN and sends it an ACK. It is still waiting for an ACK to its own SYN.	SYN-RECEIVED
SYN-RECEIVED	Client A Step #2 Receive: Client A receives the ACK from Client B for its SYN and is done with connection establishment.	ESTABLISHED	SYN-RECEIVED	Client B Step #2 Receive: Client B receives the ACK from Client A for its SYN and is done with connection establishment.	ESTABLISHED



1. Suppose a host sce.iit.ac.in wants IP address of www.microsoft.com. How does the DNS perform this operation?

2. Give an example of Domain name hierarchy.

3. Explain with example the DNS iterative and recursive query.

4. Draw the message format of DNS protocol and explain its fields.

Ans. TCP Simultaneous Open Connection Establishment Procedure

This diagram shows what happens when two devices try to open a connection to each other at the same time. In this case instead of a three-way handshake, each sends a SYN and receives an ACK. They each follow the same sequence of states, which differs from both sequences in the normal three-way handshake.

(2) Example of domain name hierarchy:

(i) .com

The granddaddy of the internet boom, .com is the TLD to have - assuming it's not already been taken, of course!

(ii) .net & .org

The two other non-country specific TLDs are always nice to get, if you can - they lack the familiarity of a .com, and for some sites that simply won't do, but for certain applications or audiences a good, short & easy to remember .net or .org address can be very effective.

(iii) .co.uk

If you're targeting a UK (or any other relevant country, for that matter), then the country specific commercial domain is another good choice - good enough for Modern Life, in fact - and in the UK at least, the .co.uk domain has a similar level of saturation in the commercial sector as the global .com.

(iv) .org.uk

Although relatively new, the .org.uk domain is quite desirable - not as much as .co.uk, nor .org, but compared to the rest of the newer TLDs- this one has potential.

(v) .us

Made popular by del.icio.us, the .us TLD is great for those targeting a primarily US-based market, but lacks the appeal of a good .com.

(vi) .eu

Although not in widespread use, the newer .eu domain is limited in usefulness to countries within Europe - but if your business trades exclusively in this region, the .eu TLD is short and easy to remember - and it's a lot easier to get hold of a good .eu domain than a half-decent .com!

(vii).biz, .info

Unfortunately, early adoption by spammers and other less reputable sites have sullied the .biz and .info domains somewhat. I'm surprised no-one has registered 'made-for-adsense.info' - perhaps that's too obvious?

(viii) .me.uk

As with the more global .name (see below), .me.uk is intended for individuals, and for that purpose it's ideal - unfortunately, it's suitable for only that purpose.

(3) DNS Recursive and Iterative Queries

With a recursive name query , the DNS client requires that the DNS server respond to the client with either the requested resource record or an error message stating that the record or domain name does not exist. The DNS server cannot just refer the DNS client to a different DNS server.

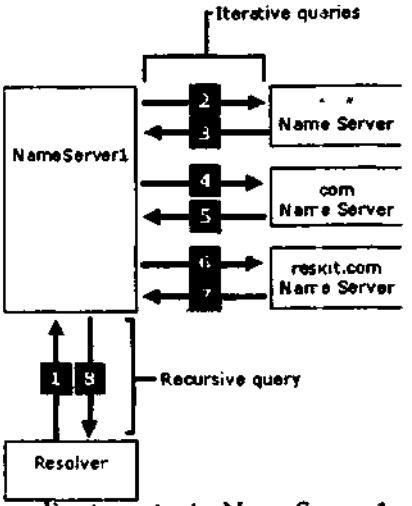
Thus, if a DNS server does not have the requested information when it receives a recursive query, it queries other servers until it gets the information, or until the name query fails.

Recursive name queries are generally made by a DNS client to a DNS server, or by a DNS server that is configured to pass unresolved name queries to another DNS server, in the case of a DNS server configured to use a forwarder.

An iterative name query is one in which a DNS client allows the DNS server to return the best answer it can give based on its cache or zone data. If the queried DNS server does not have an exact match for the queried name, the best possible information it can return is a referral (that is, a pointer to a DNS server authoritative for a lower level of the domain namespace). The DNS client can then query the DNS server for which it obtained a referral. It continues this process until it locates a DNS server that is authoritative for the queried name, or until an error or time-out condition is met.

This process is sometimes referred to as "walking the tree," and this type of query is typically initiated by a DNS server that attempts to resolve a recursive name query for a DNS client.

Figure shows an example of iterative and recursive queries. This example assumes that none of the servers have the requested information in their caches.



1. The client contacts NameServer1 with a recursive query for noam.reskit.com. The server must now return either the answer or an error message.

2. NameServer1 checks its cache and zones for the answer, but does not find it, so it contacts a server authoritative for the Internet (that is, a root server) with an iterative query for noam.reskit.com.

3. The server at the root of the Internet does not know the answer, so it responds with a referral to a server authoritative for the .com domain.

4. NameServer1 contacts a server authoritative for the .com domain with an iterative query for noam.reskit.com.

5. The server authoritative for the .com domain does not know the exact answer, so it responds with a referral to a server authoritative for the reskit.com domain.

6. NameServer1 contacts the server authoritative for the reskit.com domain with an iterative query for noam.reskit.com.

7. The server authoritative for the reskit.com domain does know the answer. It responds with the requested IP address.

8. NameServer1 responds to the client query with the IP address for noam.reskit.com.

(4) Message format of DNS Protocol :

DNS protocol is utilized to identify servers by their IP addresses and aliases given their registered name. The request is usually simple, including just the name of the server. The response however is usually very complex because it contains all the addresses and aliases that the server might have. Because of this a compression algorithm is utilized in all cases to reduce the number of redundant data and the size of the datagrams. UDP is utilized to send and receive DNS requests.

DNS MESSAGE FORMAT

Header
Question
Answer
Authority
Additional

DNS HEADER FORMAT

OCTET 1,2 ID
OCTET 3,4 QR(1 bit) + OPCODE(4 bit)+ AA(1 bit) + TC(1 bit) + RD(1 bit)+ RA(1 bit) + Z(3 bit) + RCODE(4 bit)
OCTET 5,6 QDCOUNT
OCTET 7,8 ANCOUNT
OCTET 9,10 NSCOUNT
OCTET 11,12 ARCOUNT

QUESTION FORMAT

OCTET 1,2,...n QNAME
OCTET n+1,n+2 QTYPE
OCTET n+3,n+4 QCLASS

ANSWER, AUTHORITY, ADDITIONAL FORMAT

OCTET 1,2,..n	NAME
OCTET n+1,n+2	TYPE
OCTET n+3,n+4	CLASS
OCTET n+5,n+6,n+7,n+8	TTL
OCTET n+9,n+10	RDLENGTH
OCTET n+11,n+12,.....	RDATA

Q.5.(c). What is firewall? What are the different type of firewall? How does it works? Mention thelimitations of firewalls.

Ans. Firewall

A firewall is a hardware or software device which is configured to permit, deny, or proxy data through a computer network which has different levels of trust.

Types of Firewall

There are several classifications of firewalls depending on where the communication is taking place, where the communication is intercepted and the state that is being traced.

(i) Network layer and packet filters :

Network layer firewalls, also called packet filters, operate at a relatively low level of the TCP/IP protocol stack, not allowing packets to pass through the firewall unless they match the established ruleset. The firewall administrator may define the rules; or default rules may apply. The term packet filter originated in the context of BSD operating systems.

Network layer firewalls generally fall into two sub-categories, stateful and stateless. Stateful firewalls maintain context about active sessions, and use that "state information" to speed up packet processing. Any existing network connection can be described by several properties, including source and destination IP address, UDP or TCP ports, and the current stage of the connection's lifetime (including session initiation, handshaking, data transfer, or completion connection). If a packet does not match an existing connection, it will be evaluated according to the ruleset for new connections. If a packet matches an existing connection based on comparison with the firewall's state table, it will be allowed to pass without further processing.

Stateless firewalls have packet-filtering capabilities, but cannot make more complex decisions on what stage communications between hosts have reached.

Modern firewalls can filter traffic based on many packet attributes like source IP address, source port, destination IP address or port, destination service like WWW or FTP. They can filter based on protocols, TTL values, netblock of originator, domain name of the source, and many other attributes.

Commonly used packet filters on various versions of Unix are ipf (various), ipfw (FreeBSD/ Mac OS X), pf (OpenBSD, and all other BSDs), iptables/ipchains (Linux).

(ii) Application-layer

Application-layer firewalls work on the application level of the TCP/IP stack (i.e., all browser traffic, or all telnet or ftp traffic), and may intercept all packets traveling to or from an application. They block other packets (usually dropping them without acknowledgement to the sender). In principle, application firewalls can prevent all unwanted outside traffic from reaching protected machines.

On Inspecting all packets for improper content, firewalls can restrict or prevent outright the spread of networked computer worms and trojans. In practice, however, this becomes so complex and so difficult to attempt (given the variety of applications and the diversity of content each may allow in its packet traffic) that comprehensive firewall design does not generally attempt this approach.

The XML firewall exemplifies a more recent kind of application-layer firewall.

(iii) Proxies

A proxy device (running either on dedicated hardware or as software on a general-purpose machine) may act as a firewall by responding to input packets (connection requests, for example) in the manner of an application, whilst blocking other packets.

Proxies make tampering with an internal system from the external network more difficult and misuse of one internal system would not necessarily cause a security breach exploitable from outside the firewall (as long as the application proxy remains intact and properly configured). Conversely, intruders may hijack a publicly-reachable system and use it as a proxy for their own purposes; the proxy then masquerades as that system to other internal machines. While use of internal address spaces enhances security, crackers may still employ methods such as IP spoofing to attempt to pass packets to a target network.

(iv) Network address translation

Firewalls often have network address translation (NAT) functionality, and the hosts protected behind a firewall commonly have addresses in the "private address range". Firewalls often have such functionality to hide the true address of protected hosts. Originally, the NAT function was developed to address the limited amount of IPv4 routable addresses that could be used or assigned to companies or individuals as well as reduce both the amount and therefore cost of obtaining enough public addresses for every computer in an organization. Hiding the addresses of protected devices has become an increasingly important defense against network reconnaissance.