Model Question Paper-1 with effect from 2018-19 (CBCS Scheme)

COMPUTER NETWORKS AND SECURITY

Module - 1

1(A) - Explain client-server and Peer-to-Peer architecture

1.Client-Server Architecture

• In this architecture, there is a server and many clients distributed over the network (Figure 1.1a).
• The server is always-on while a client can be randomly run.
• The server is listening on the network and a client initializes the communication.
• Upon the requests from a client, the server provides certain services to the client.
• Usually, there is no communication between two clients.
• The server has a fixed IP address.
• A client contacts the server by sending a packet to the server's IP address.
• A server is able to communicate with many clients.
• The applications such as FTP, telnet, Web, e-mail etc use the client-server architecture.

2 P2P Architecture

• There is no dedicated server (Figure 1.1b).
• Pairs of hosts are called peers.
• The peers communicate directly with each other.
• The peers are not owned by the service-provider. Rather, the peers are laptops controlled by users.
• Many of today's most popular and traffic-intensive applications are based on P2P architecture.
• Examples include file sharing (BitTorrent), Internet telephone (Skype) etc.
• Main feature of P2P architectures: self-scalability.
• For ex: In a P2P file-sharing system,
☐ Each peer generates workload by requesting files.
☐ Each peer also adds service-capacity to the system by distributing files to other peers.
• Advantage: Cost effective '.' Normally, server-infrastructure & server bandwidth are not required.
• Three challenges of the P2P applications:

1) ISP Friendly

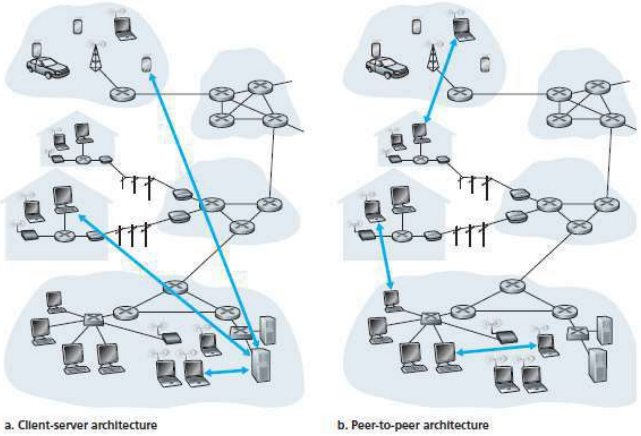
☐ Most residential ISPs have been designed for asymmetrical bandwidth usage.
☐ Asymmetrical bandwidth means there is more downstream-traffic than upstream-traffic.
☐ But P2P applications shift upstream-traffic from servers to residential ISPs, which stress on the ISPs.

2) Security

☐ Since the highly distribution and openness, P2P applications can be a challenge to security.

3) Incentive

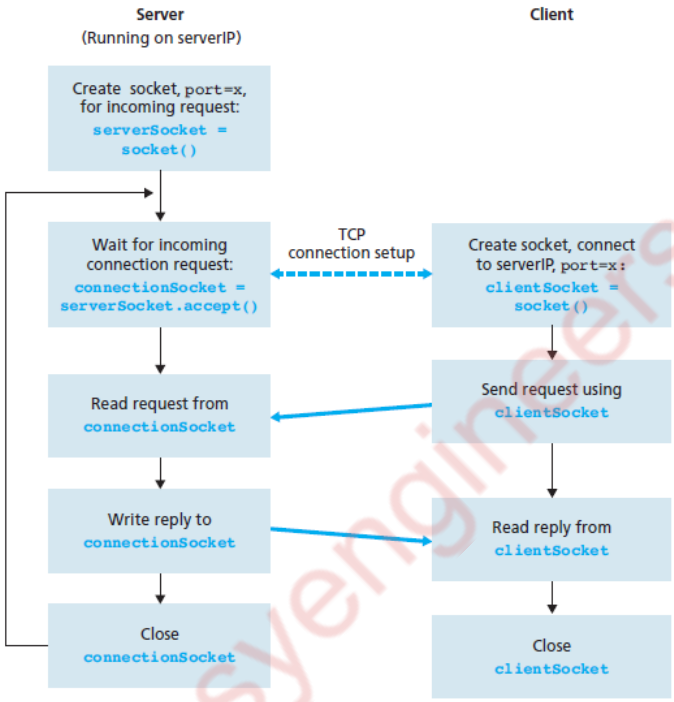
☐ Success of P2P depends on convincing users to volunteer bandwidth & resources to the applications.



a. Client-server architecture b. Peer-to-peer architecture

1(B) - Define Socket. Demonstrate the working of TCP Socket

- A typical network application consists of a pair of programs—a client program and a server program—residing in two different end systems.
- When these two programs are executed, a client process and a server process are created, and these processes communicate with each other by reading from, and writing to, sockets.
- When creating a network application, the developer's main task is therefore to write the code for both the client and server programs.



• The client-side of the application is as follows

```
from socket import *
serverName = 'servername'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort)) // This line initiates TCP connection b/w
client & server
sentence = raw_input('Input lowercase sentence:')
clientSocket.send(sentence)
modifiedSentence = clientSocket.recv(1024)
print 'From Server:', modifiedSentence
clientSocket.close()
```

• The server-side of the application is as follows:

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind(('',serverPort))
serverSocket.listen(1) // This line specifies no. of connection-requests from the client to
server
print 'The server is ready to receive'
while 1:
```

```
connectionSocket, addr=serverSocket.accept() //allows server to accept connection request
from client
sentence = connectionSocket.recv(1024)
capitalizedSentence = sentence.upper()
connectionSocket.send(capitalizedSentence)
connectionSocket.close()
```

1(C) - Explain the working of BitTorrent for file distribution

- The collection of all peers participating in the distribution of a particular file is called a torrent.
- Peers download equal-size chunks of the file from one another. Chunk size = 256 KBytes.
- The peer also uploads chunks to other peers.
- Once a peer has acquired the entire file, the peer may leave the torrent or remain in the torrent.
- Each torrent has an infrastructure node called tracker.
- Here is how it works (Figure):
    1) When a peer joins a torrent, the peer
        → registers itself with the tracker and
        → periodically informs the tracker that it is in the torrent.
    2) When a new peer joins the torrent, the tracker
        → randomly selects a subset of peers from the set of participating peers and
        → sends the IP addresses of these peers to the new peer.
    3) Then, the new peer tries to establish concurrent TCP connections with all peers on this list.

☐ All peers on the list are called neighboring-peers.

4) Periodically, the new peer will ask each of the neighboring-peers for the set of chunks.

- To choose the chunks to download, the peer uses a technique called rarest-first.
- Main idea of rarest-first:
  → Determine the chunks that are the rarest among the neighbors and
  → Request then those rarest chunks first.

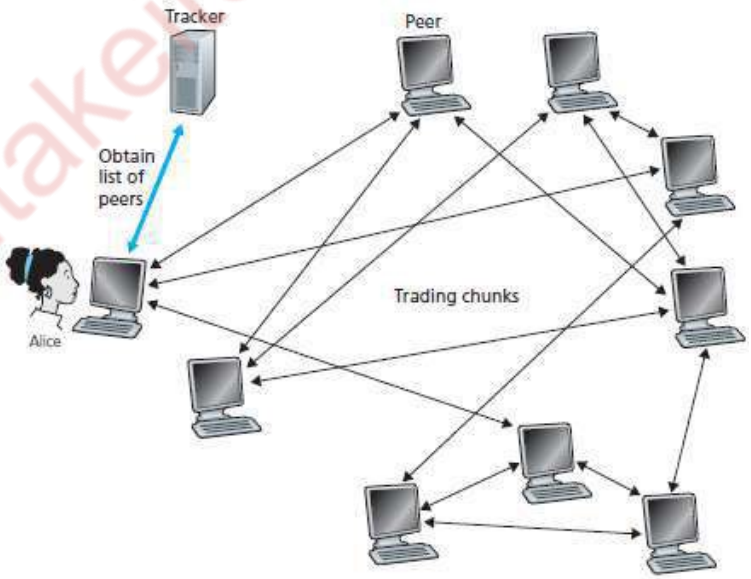


Fig. 2 File distribution with BitTorrent

2(A) - Describe in detail the services offered by DNS and explain DNS message format.

Services Provided by DNS

- The DNS is
  1) A distributed database implemented in a hierarchy of DNS servers.
  2) An application-layer protocol that allows hosts to query the distributed database.
- DNS servers are often UNIX machines running the BIND software.
- The DNS protocol runs over UDP and uses port 53. (BIND ☐ Berkeley Internet Name Domain)
- DNS is used by application-layer protocols such as HTTP, SMTP, and FTP.
- Assume a browser requests the URL www.someschool.edu/index.html.
- Next, the user's host must first obtain the IP address of www.someschool.edu
- This is done as follows:
  1) The same user machine runs the client-side of the DNS application.
  2) The browser
     → extracts the hostname "www.someschool.edu" from the URL and
     → passes the hostname to the client-side of the DNS application.

3) The client sends a query containing the hostname to a DNS server.
4) The client eventually receives a reply, which includes the IP address for the hostname.
5) After receiving the IP address, the browser can initiate a TCP connection to the HTTP server.

- DNS also provides following services:

1) Host Aliasing

A host with a complicated hostname can have one or more alias names.

2) Mail Server Aliasing

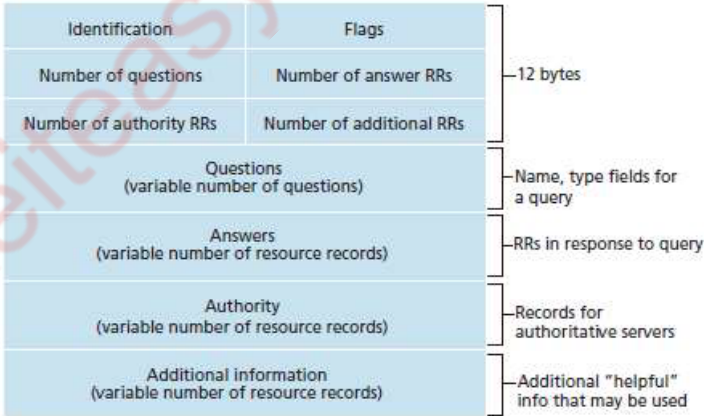
For obvious reasons, it is highly desirable that e-mail addresses be mnemonic.

3) Load Distribution

DNS is also used to perform load distribution among replicated servers.
Busy sites are replicated over multiple servers & each server runs on a different system.

DNS Messages

- Two types of DNS messages: 1) query and 2) reply.
- Both query and reply messages have the same format.



- The various fields in a DNS message are as follows

1) Header Section

- The first 12 bytes is the header-section.
- This section has following fields:

i) Identification

☐ This field identifies the query.
☐ This identifier is copied into the reply message to a query.
☐ This identifier allows the client to match received replies with sent queries.

ii) Flag

☐ This field has following 3 flag-bits:

a) Query/Reply

¤ This flag-bit indicates whether the message is a query (0) or a reply (1).

b) Authoritative

¤ This flag-bit is set in a reply message when a DNS server is an authoritative-server.

c) Recursion Desired

¤ This flag-bit is set when a client desires that the DNS server perform recursion.

iii) Four Number-of-Fields

☐ These fields indicate the no. of occurrences of 4 types of data sections that follow the header.

2) Question Section

- This section contains information about the query that is being made.
- This section has following fields:

i) Name

☐ This field contains the domain-name that is being queried.

ii) Type

☐ This field indicates the type of question being asked about the domain-name.

3) Answer Section

- This section contains a reply from a DNS server.
- This section contains the resource-records for the name that was originally queried.
- A reply can return multiple RRs in the answer, since a hostname can have multiple IP addresses.

4) Authority Section

- This section contains records of other authoritative-servers.

5) Additional Section

- This section contains other helpful records.

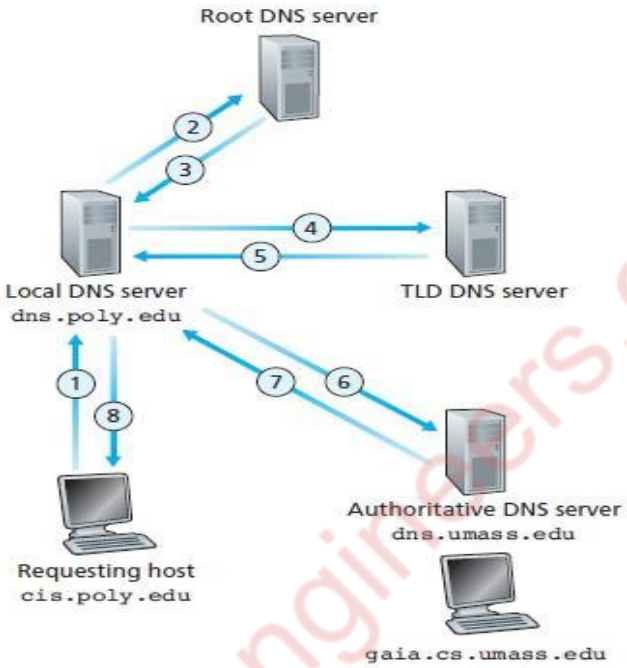
2(B) - Compare HTTP and SMTP

HTTP	SMTP
Pull Protocol- someone loads information on a Web server and users use HTTP to pull the information from the server at their convenience.	Push Protocol- the sending mail server pushes the file to the receiving mail server.
HTTP does not mandates data to be in 7-bit ASCII format.	SMTP requires each message, including the body of each message, to be in 7-bit ASCII format.
HTTP encapsulates each object in its own HTTP response message.	Internet mail places all of the message's objects into one message.

2(C) - With a diagram explain the interaction of the various DNS servers

Two type of Interaction:

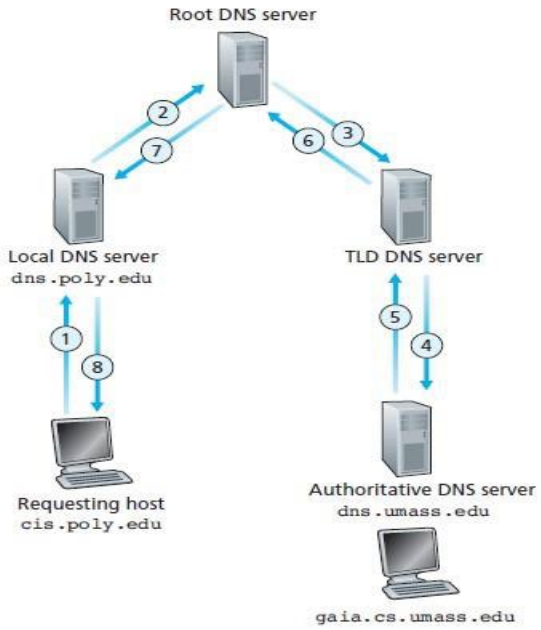
1) **Recursive Queries:**



Here DNS query is sent to local DNS server then to root server, then to TLD server and finally to authoritative DNS server. DNS response arrives in the reverse order.

2) **Iterative Queries:**

Here DNS query will be sent to Local DNS server, then to root server. Root server sends the IP address of TLD server. Now local DNS server sends query to TLD DNS server. TLD DNS server sends the IP address
of authoritative DNS server to local DNS server. Now Local DNS server sends query to authoritative DNS server. Authoritative DNS server sends the IP address of host to local DNS server. Local DNS server sends it to the host.



Module – 2

3(A) - Explain the concept of transport layer Multiplexing and De-multiplexing.

Multiplexing and Demultiplexing

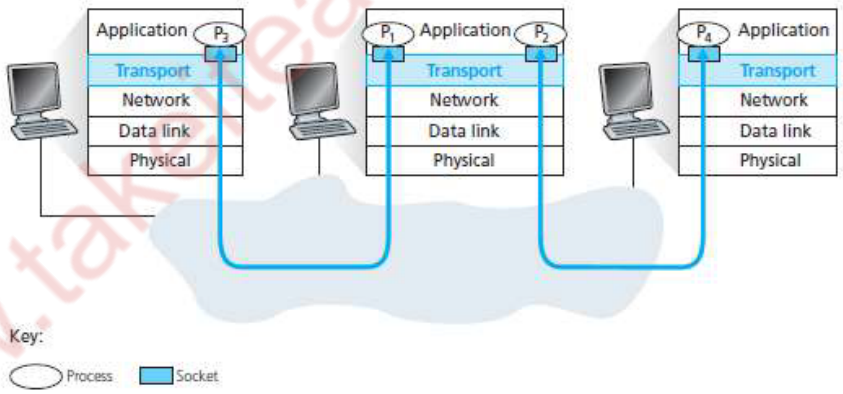
• A process can have one or more sockets.
• The sockets are used to pass data from the network to the process and vice versa.

1) Multiplexing

☐ At the sender, the transport-layer
→ gathers data-chunks at the source-host from different sockets
→ encapsulates data-chunk with header to create segments and
→ passes the segments to the network-layer.
☐ The job of combining the data-chunks from different sockets to create a segment is called multiplexing.

2) Demultiplexing

☐ At the receiver, the transport-layer
→ examines the fields in the segments to identify the receiving-socket and
→ directs the segment to the receiving-socket.
☐ The job of delivering the data in a segment to the correct socket is called demultiplexing.
• In Figure
☐ In the middle host, the transport-layer must demultiplex segments arriving from the networklayer
to either process P1 or P2.
☐ The arriving segment's data is directed to the corresponding process's socket.



3(B) - With neat diagram, explain TCP segment structure and its fields.

TCP Segment Structure

• The segment consists of header-fields and a data-field.
• The data-field contains a chunk-of-data.
• When TCP sends a large file, it breaks the file into chunks of size MSS.

• Figure 3.B shows the structure of the TCP segment.
• The fields of TCP segment are as follows:

1) Source and Destination Port Numbers

- These fields are used for multiplexing/demultiplexing data from/to upper-layer applications.

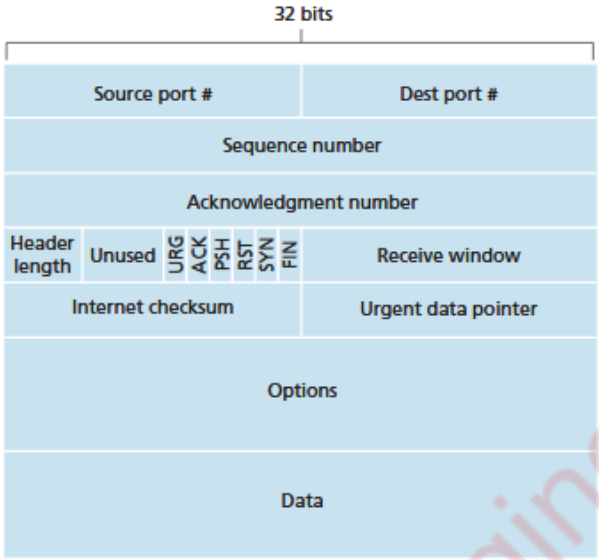


Figure 3.B: TCP segment structure

2) Sequence Number & Acknowledgment Number

- These fields are used by sender & receiver in implementing a reliable data-transfer-service.

3) Header Length

- This field specifies the length of the TCP header.

4) Flag

- This field contains 6 bits.

i) ACK

¤ This bit indicates that value of acknowledgment field is valid.

ii) RST, SYN & FIN

¤ These bits are used for connection setup and teardown.

iii) PSH

¤ This bit indicates the sender has invoked the push operation.

iv) URG

¤ This bit indicates the segment contains urgent-data.

5) Receive Window

- This field defines receiver's window size
- This field is used for flow control.

6) Checksum

- This field is used for error-detection.

7) Urgent Data Pointer

- field indicates the location of the last byte of the urgent data.

8) Options This

- This field is used when a sender & receiver negotiate the MSS for use in high-speed networks.

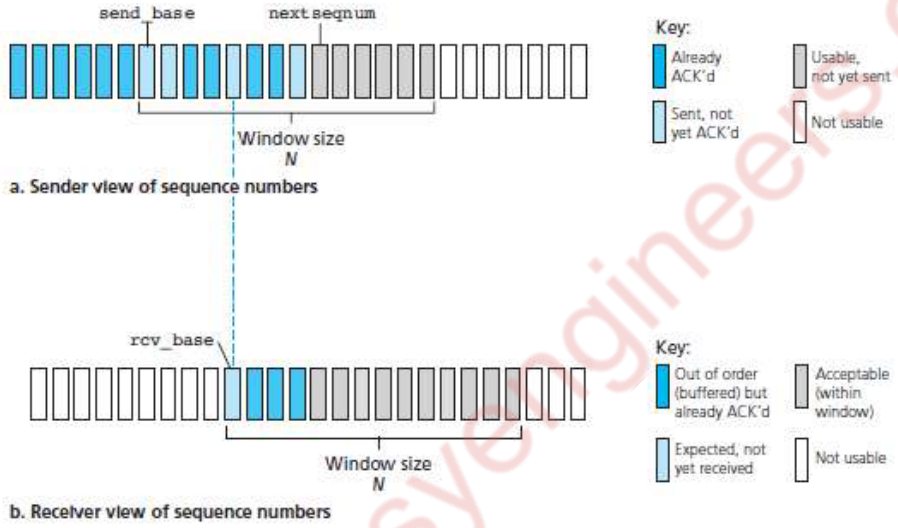
3(C) - Explain in brief, TCP congestion control mechanism.

• TCP has congestion-control mechanism.
• TCP uses end-to-end congestion-control rather than network-assisted congestion-control
• Here is how it works:
➢ Each sender limits the rate at which it sends traffic into its connection as a function of perceived congestion.
i) If sender perceives that there is little congestion, then sender increases its data-rate.
ii) If sender perceives that there is congestion, then sender reduces its data-rate.
• This approach raises three questions:
1) How does a sender limit the rate at which it sends traffic into its connection?
2) How does a sender perceive that there is congestion on the path?
3) What algorithm should the sender use to change its data-rate?
• The sender keeps track of an additional variable called the congestion-window (cwnd).
• The congestion-window imposes a constraint on the data-rate of a sender.
• The amount of unacknowledged-data at a sender will not exceed minimum of (cwnd & rwnd), that is:
• The sender's data-rate is roughly cwnd/RTT bytes/sec.
• Explanation of Loss event:
➢ A "loss event" at a sender is defined as the occurrence of either
→ timeout or
→ receipt of 3 duplicate ACKs from the receiver.
➢ Due to excessive congestion, the router-buffer along the path overflows. This causes a datagram to be dropped.
➢ The dropped datagram, in turn, results in a loss event at the sender.
➢ The sender considers the loss event as an indication of congestion on the path.
• How congestion is detected?
➢ Consider the network is congestion-free.
➢ Acknowledgments for previously unacknowledged segments will be received at the sender.
➢ TCP
→ will take the arrival of these acknowledgments as an indication that all is well and
→ will use acknowledgments to increase the window-size (& hence data-rate).
➢ TCP is said to be self-clocking because
→ acknowledgments are used to trigger the increase in window-size
➢ Congestion-control algorithm has 3 major components:
1) Slow start
2) Congestion avoidance and
3) Fast recovery.

4(B) - With neat diagram, explain Selective Repeat protocol.

Selective Repeat (SR)

- Problem with GBN:
  - GBN suffers from performance problems.
  - When the window-size and bandwidth-delay product are both large, many packets can be in the pipeline.
  - Thus, a single packet error results in retransmission of a large number of packets.
- Solution: Use Selective Repeat (SR).



- The sender retransmits only those packets that it suspects were erroneous.
- Thus, avoids unnecessary retransmissions. Hence, the name "selective-repeat".
- The receiver individually acknowledge correctly received packets.
- A window-size N is used to limit the no. of outstanding, unacknowledged packets in the pipeline.

4(C) - Explain in brief, TCP connection Management process.

1. Connection Setup & Data Transfer

- To setup the connection, three segments are sent between the two hosts. Therefore, this process is referred to as a three-way handshake.
- Suppose a client-process wants to initiate a connection with a server-process.
- Figure 2.33 illustrates the steps involved:

Step 1: Client sends a connection-request segment to the Server

- The client first sends a connection-request segment to the server.
- The connection-request segment contains:

1) SYN bit is set to 1.
2) Initial sequence-number (client_isn).

- The SYN segment is encapsulated within an IP datagram and sent to the server.

Step 2: Server sends a connection-granted segment to the Client

- Then, the server
  → extracts the SYN segment from the datagram
  → allocates the buffers and variables to the connection and
  → sends a connection-granted segment to the client.
- The connection-granted segment contains:

1) SYN bit is set to 1.
2) Acknowledgment field is set to client_isn+1.
3) Initial sequence-number (server_isn).

Step 3: Client sends an ACK segment to the Server

- Finally, the client
  → allocates buffers and variables to the connection and
  → sends an ACK segment to the server
- The ACK segment acknowledges the server.
- SYN bit is set to zero, since the connection is established.

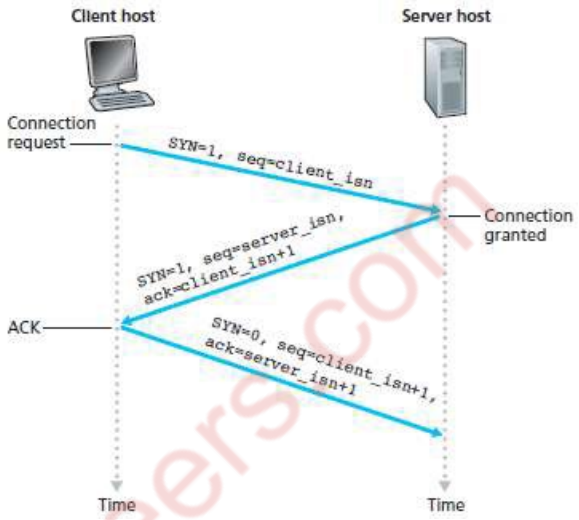


fig. TCP three-way handshake: segment exchange

2. Connection Release

- Either of the two processes in a connection can end the connection.
- When a connection ends, the "resources" in the hosts are de-allocated.
- Suppose the client decides to close the connection.
- Figure illustrates the steps involved:

1) The client-process issues a close command.
   ¤ Then, the client sends a shutdown-segment to the server.
   ¤ This segment has a FIN bit set to 1.
2) The server responds with an acknowledgment to the client.
3) The server then sends its own shutdown-segment.
   ¤ This segment has a FIN bit set to 1.
4) Finally, the client acknowledges the server's shutdown-segment.

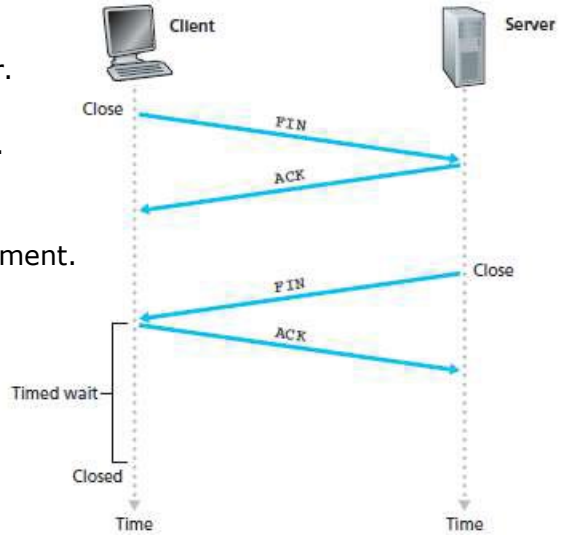


Fig . Closing a TCP connection

Module – 3

5(A) - Explain the three switching techniques

Switching

• Three types of switching fabrics (Figure 5.A):

1) Switching via memory
2) Switching via a bus and
3) Switching via an interconnection network.

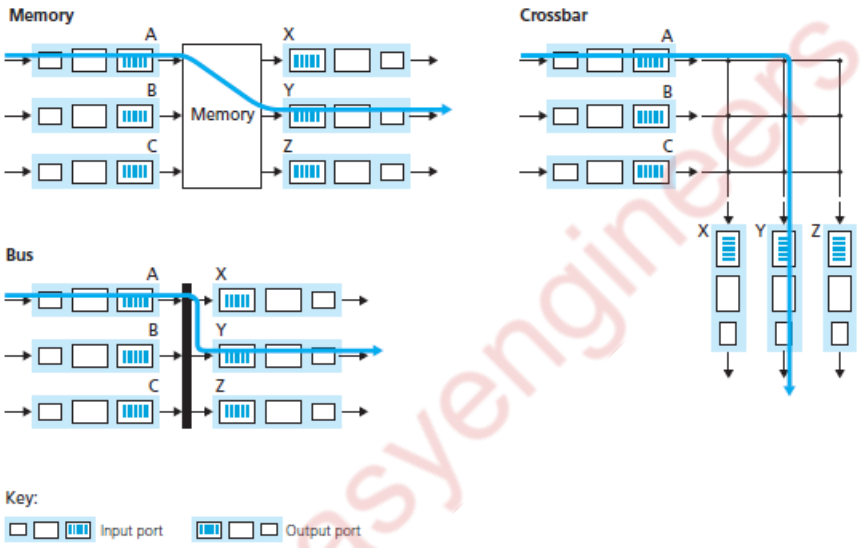


Figure 5.A: Three switching techniques

Switching via Memory

• Switching b/w input-ports & output-ports is done under direct control of CPU i.e. routing-processor.

• Input and output-ports work like a traditional I/O devices in a computer.

• Here is how it works (Figure 3.7a):

i) On arrival of a packet, the input-port notifies the routing-processor via an interrupt.
ii) Then, the packet is copied from the input-port to processor-memory.
iii) Finally, the routing-processor
→ extracts the destination-address from the header
→ looks up the appropriate output-port in the forwarding-table and
→ copies the packet into the output-port's buffers.

• Let memory-bandwidth = B packets per second.

Thus, the overall forwarding throughput must be less than B/2.

• Disadvantage:

- Multiple packets cannot be forwarded at the same time. This is because

→ only one memory read/write over the shared system bus can be done at a time.

Switching via a Bus

• Switching b/w input-ports & output-ports is done without intervention by the routing-processor.
• Here is how it works

i) The input-port appends a switch-internal label (header) to the packet.
   - The label indicates the local output-port to which the packet must be transferred.

ii) Then, the packet is received by all output-ports.
   - But, only the port that matches the label will keep the packet.

iii) Finally, the label is removed at the output-port.

• Disadvantages:
   i) Multiple packets cannot be forwarded at the same time. This is because
      → only one packet can cross the bus at a time.
   ii) The switching speed of the router is limited to the bus-speed.

Switching via an Interconnection Network

• A crossbar switch is an interconnection network.
• The network consists of 2N buses that connect N input-ports to N output-ports.
• Each vertical bus intersects each horizontal bus at a Crosspoint.
• The Crosspoint can be opened or closed at any time by the switch-controller.
• Here is how it works

1) To move a packet from port A to port Y, the switch-controller closes the Crosspoint at the intersection of buses A and Y.
2) Then, port A sends the packet onto its bus, which is picked up by bus Y.

• Advantage:
   - Crossbar networks are capable of forwarding multiple packets in parallel.
   - For ex: A packet from port B can be forwarded to port X at the same time. This is because
     → A-to-Y and B-to-X packets use different input and output buses.

• Disadvantage:
- If 2 packets have to use same output-port, then one packet has to wait. This is because
→ only one packet can be sent over any given bus at a time.

5(B) - Explain distance vector algorithm.

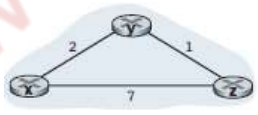
• Distance vector (DV) algorithm is 1) iterative, 2) asynchronous, and 3) distributed.

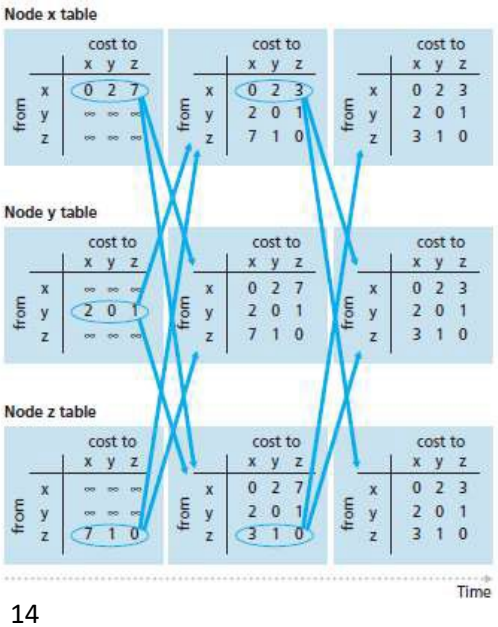
1) It is distributed. This is because each node
   → receives some information from one or more of its directly attached neighbors
   → performs the calculation and
   → distributes then the results of the calculation back to the neighbors.
2) It is iterative. This is because
   → the process continues on until no more info is exchanged b/w neighbors.
3) It is asynchronous. This is because
   → the process does not require all of the nodes to operate in lockstep with each other.

Distance-Vector (DV) Algorithm

At each node, x:

```
1  Initialization:
2     for all destinations y in N:
3        Dx(y) = c(x,y)    /* if y is not a neighbor then c(x,y) = ∞ */
4     for each neighbor w
5        Dw(y) = ? for all destinations y in N
6     for each neighbor w
7        send distance vector Dx = [Dx(y): y in N] to w
8
9  loop
10    wait (until I see a link cost change to some neighbor w or
11          until I receive a distance vector from some neighbor w)
12
13    for each y in N:
14       Dx(y) = minv{c(x,v) + Dv(y)}
15
16    if Dx(y) changed for any destination y
17       send distance vector Dx = [Dx(y): y in N] to all neighbors
18
19 forever
```





The operation of the algorithm is illustrated in a synchronous manner. Here, all nodes simultaneously

→ receive distance vectors from their neighbours

→ compute their new distance vectors, and

→ inform their neighbours if their distance vectors have changed.

• The table in the upper-left corner is node x's initial routing-table.

• In this routing-table, each row is a distance vector.

• The first row in node x's routing-table is $D_x = [D_x(x), D_x(y), D_x(z)] = [0, 2, 7]$.

• After initialization, each node sends its distance vector to each of its two neighbours.

• This is illustrated in Figure by the arrows from the first column of tables to the second column of tables.

• For example, node x sends its distance vector Dx = [0, 2, 7] to both nodes y and z. After receiving
the updates, each node recomputes its own distance vector.

• For example, node x computes

$D_x(x) = 0$

$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} = \min\{2 + 0, 7 + 1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} = \min\{2 + 1, 7 + 0\} = 3$

• The second column therefore displays, for each node, the node's new distance vector along with
distance vectors just received from its neighbours.

• Note, that node x's estimate for the least cost to node z, $D_x(z)$, has changed from 7 to 3.

• The process of receiving updated distance vectors from neighbours, recomputing routing-table
entries, and informing neighbours of changed costs of the least-cost path to a destination continues
until no update messages are sent.

• The algorithm remains in the quiescent state until a link cost changes.

4) And so on. . . .

5) When the LS algorithm terminates,

We have, for each node, its predecessor along the least-cost path from the source.

6(A) - With general format, explain various fields of IPv6.

• The format of the IPv6 datagram is shown in Figure 6.A

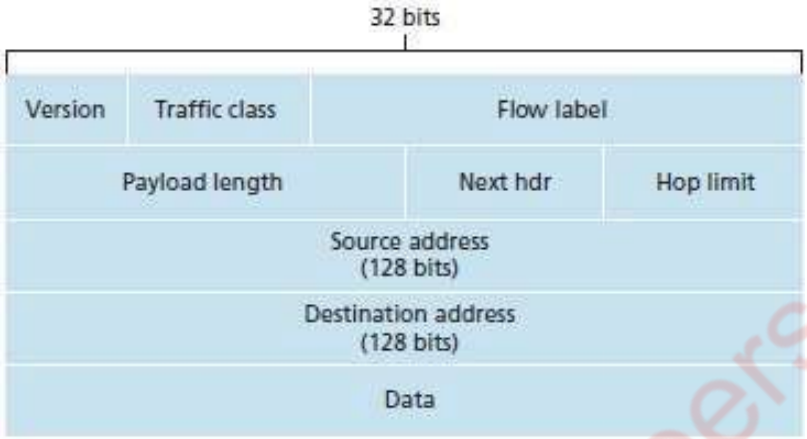


Fig: 6.A

• The following fields are defined in IPv6:

1) Version

- This field specifies the IP version, i.e., 6.

2) Traffic Class

- This field specifies priority of packet based on parameters such as delay, throughput, reliability & cost.

3) Flow Label

- This field is used to provide special handling for a particular flow of data.

4) Payload Length

- This field shows the length of the IPv6 payload.

5) Next Header

- This field identifies type of extension header that follows the basic header.

6) Hop Limit

- This field shows the maximum number of routers the packet can travel.
- The contents of this field are decremented by 1 by each router that forwards the datagram.
- If the hop limit count reaches 0, the datagram is discarded.

7) Source & Destination Addresses

- These fields show the addresses of the source & destination of the packet.

8) Data

- This field is the payload portion of the datagram.
- When the datagram reaches the destination, the payload will be
  → removed from the IP datagram and
  → passed on to the upper layer protocol (TCP or UDP).

6(B) - List the broadcast routing algorithms. Explain any two of them.

Broadcast-routing means delivering a packet from a source-node to all other nodes in the network.

Broadcast Routing Algorithms are:

- **N-way Unicast**
- **Uncontrolled Flooding**
- **Controlled Flooding**
- **Spanning - Tree Broadcast**

Spanning - Tree Broadcast

• This is another approach to providing broadcast. (MST → Minimum Spanning Tree).
• Spanning-tree is a tree that contains each and every node in a graph.
• A spanning-tree whose cost is the minimum of all of the graph's spanning-trees is called a MST.
• Here is how it works (Figure 3.33):

1) Firstly, the nodes construct a spanning-tree.
2) The node sends broadcast-packet out on all incident links that belong to the spanning-tree.
3) The receiving-node forwards the broadcast-packet to all neighbors in the spanning-tree.

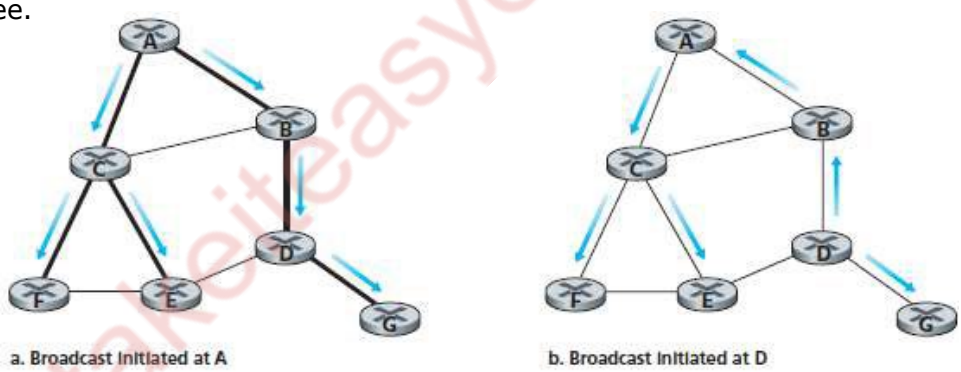


Fig: Broadcast along a spanning-tree

• Disadvantage:
Complex: The main complexity is the creation and maintenance of the spanning-tree.

Center Based Approach

• This is a method used for building a spanning-tree.
• Here is how it works:
1) A center-node (rendezvous point or a core) is defined.
2) Then, the nodes send unicast tree-join messages to the center-node.
3) Finally, a tree-join message is forwarded toward the center until the message either
→ arrives at a node that already belongs to the spanning-tree or
→ arrives at the center.

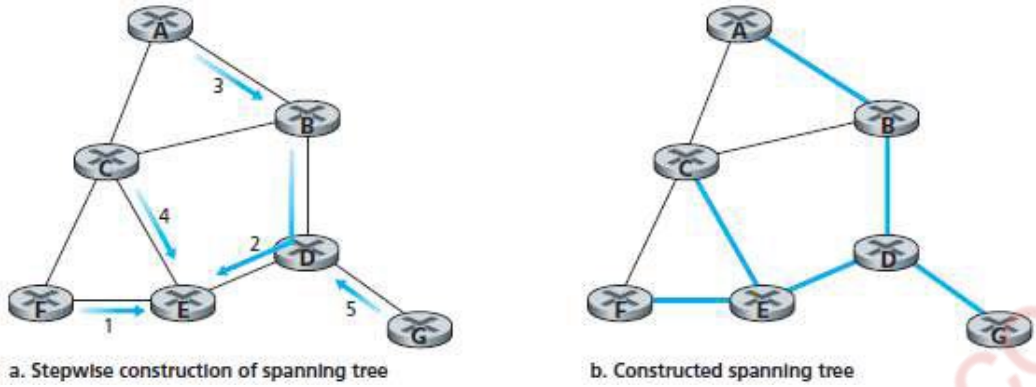


Fig: Center-based construction of a spanning-tree

6(C) - Explain the intra-AS routing protocol in detail.

Intra-AS Routing Protocol

• The routing-algorithm running within an autonomous-system is called intra-AS routing protocol.

• All routers within the same AS must run the same intra-AS routing protocol. For ex: RIP and OSPF

• Figure 6.C provides a simple example with three ASs: AS1, AS2, and AS3.

• AS1 has four routers: 1a, 1b, 1c, and 1d. These four routers run the intra-AS routing protocol.

• Each router ination within AS1.

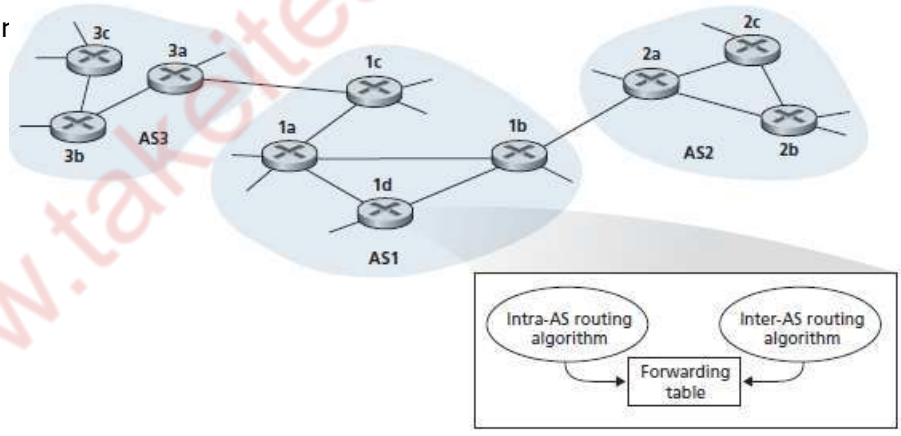


Fig: An example of interconnected autonomous-systems

Module – 4

7(A) - What are the elements of network security? Explain the threats to network security.

Elements of Network Security

1. *Confidentiality*. Information should be available only to those who have rightful access to it.

2. *Authenticity and integrity*. The sender of a message and the message itself should be verified at the receiving point.

Threats to Network Security

Internet infrastructure *attacks* are broadly classified into four categories, as follows:

1. DNS hacking
2. Routing table poisoning
3. Packet mistreatment
4. Denial of service

1. DNS hacking - Domain Name Server (DNS) hijacking, also named DNS redirection, is a type of DNS attack in which DNS queries are incorrectly resolved in order to unexpectedly redirect users to malicious sites. To perform the attack, perpetrators either install malware on user computers, take over routers, or intercept or hack DNS communication.

1. An *information-level attack* forces a server to correspond with other than the correct answer. With cache poisoning, a hacker tricks a remote name server into caching the answer for a third-party domain by providing malicious information for the domain's authorized servers. Hackers can then redirect traffic to a preselected site.

2. In a *masquerading attack*, the adversary poses as a trusted entity and obtains all the secret information. In this guise, the attacker can stop any message from being transmitted further or can change the content or redirect the packet to bogus servers. This action is also known as a *middle-ma-attack.*

3. The attacker normally sends queries to each host and receives in reply the DNS host name. In an *information leakage attack*, the attacker sends queries to all hosts and identifies which IP addresses are not used. Later on, the intruder can use those IP addresses to make other types of attacks.

4. Once a domain name is selected, it has to be registered. Various tools are available to register domain names over the Internet. If the tools are not smart enough, an invader might obtain secure information and use it to highjack the domain later. In the *domain highjacking attack*, whenever a user enters a domain address, she/he is forced to enter into the attacker's Web site. This can be very irritating and can cause a great loss of Internet usage ability.

2. Routing table poisoning –

A *routing table poisoning attack* is the undesired modification of routing tables. An attacker can do this by maliciously modifying the routing information update packets sent by routers. This is a challenging and important problem, as a routing table is the basis of routing in the Internet. Any false entry in a routing table could lead to significant consequences, such as congestion, an overwhelmed host, looping, illegal access to data, and network partition.

3. Packet-Mistreatment Attacks -

A *packet-mistreatment attack* can occur during any data transmission. A hacker may capture certain data packets and mistreat them. This type of attack is very difficult to detect. The attack may result in congestion, lowering throughput, and denial-of-service attacks. Similar to routing table poisoning attacks, packet-mistreatment attacks can also be subclassified into *link attacks* and *router attacks*. The link attack causes interruption, modification, or replication of data packets. A router attack can misroute all packets and may result in congestion or denial of service.

4. Denial of service –

A *denial-of-service attack* is a type of security breach that prohibits a user from accessing normally provided services. The denial of service does not result in information theft or any kind of information loss but can nonetheless be very dangerous, as it can cost the target person a large amount of time and money. Denial-of-service attacks affect the destination rather than a data packet or router.

7(B) - Briefly explain the steps of DES algorithm.

With the *Data Encryption Standard* (DES), plaintext messages are converted into 64-bit blocks, each encrypted using a key. The key length is 64 bits but contains only 56 usable bits; thus, the last bit of each 8 byte in the key is a parity bit for the corresponding byte. DES consists of 16 identical rounds of an operation, as shown in Figure 7.B. The details of the algorithm on each 64-bit block of message at each round *i* of operation are as follows.

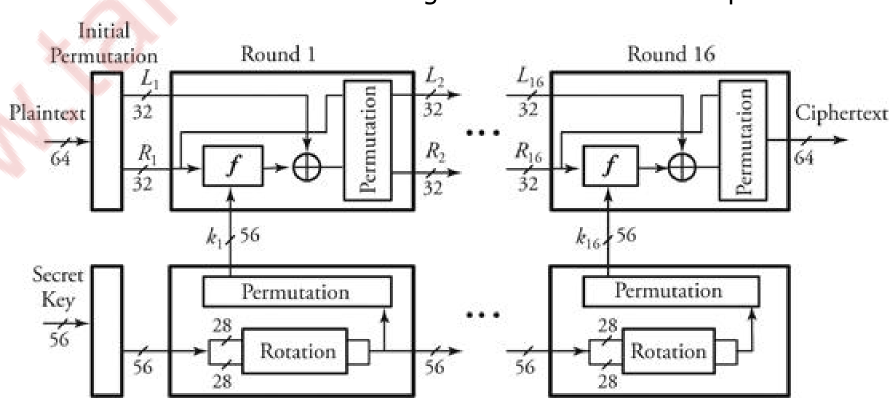


Figure 7.B: The Data Encryption Standard (DES)

1. Initialize. Before round 1 begins, all 64 bits of an incoming message and all 56 bits of the secret key are separately permuted (shuffled).

2. Each incoming 64-bit message is broken into two 32-bit halves denoted by *Li* and *Ri*, respectively.

3. The 56 bits of the key are also broken into two 28-bit halves, and each half is rotated one or two bit positions, depending on the round.
4. All 56 bits of the key are permuted, producing version *ki* of the key on round *i*.

5. In this step, is a logic Exclusive-OR, and the description of function *F*() appears next. Then, *Li* and *Ri* are determined by $R_i = L_{i-1} \oplus F(R_{i-1}, k_i).$

6. All 64 bits of a message are permuted.

The operation of function *F*() at any round *i* of DES is as follows.
1. Out of 52 bits of *ki*, function *F*() chooses 48 bits.

2. The 32-bit *Ri*−1 is expanded from 32 bits to 48 bits so that it can be combined with 48-bit *ki*. The expansion of *Ri*−1 is carried out by first breaking *Ri*−1 into eight 4-bit chunks and then expanding each chunk by copying the leftmost bit and the rightmost bit from left and right adjacent chunks, respectively.

3. Function *F*() also partitions the 48 bits of *ki* into eight 6-bit chunks.

4. The corresponding eight chunks of *Ri*−1 and eight chunks of *ki* are combined as follows

$$R_{i-1} = R_{i-1} \oplus k_i.$$

At the receiver, the same steps and the same key are used to reverse the encryption. It is now apparent that the 56-bit key length may not be sufficient to provide full security. This argument is still controversial. Triple DES provides a solution for this controversy: three keys are used, for a total of 168 bits. It should also be mentioned that DES can be implemented more efficiently in hardware than in software.

7(C) - Discuss about (i) Cryptographic techniques (ii) Authentication techniques.

Cryptographic Techniques
Cryptography has a long and fascinating history. Centuries ago, cryptography was used as a tool to protect national secrets and strategies. Today, network engineers focus on *cryptography* methods for computer communication networks. Cryptography is the process of tranforming a piece of information or message shared by two parties into some sort of code. The message is scrambled before transmission so that it is undetectable by outside watchers. This kind of message needs to be decoded at the receiving end before any further processing.

(ii) Authentication techniques.

Encryption methods offer the assurance of message confidentiality. However, a networking system must be able to verify the authenticity of the message and the sender of the message. These forms of security techniques in computer networks are known as *authentication techniques* and are categorized as *authentication with message digest* and *authentication with digital signature.* Message authentication protects a user in a network against data falsification and ensures data integrity. These methods do not necessarily use keys.

8(A) - Explain RSA algorithm. Using RSA algorithm encrypt a message m=9. Assume p=3 and q=11. Find the public and private keys and also show the cipher text.

Clearly, $n = pq = 33$. We select $x = 3$, which is relatively prime to $(p - 1)(q - 1) = 20$. Then, from $xy \bmod (p - 1)(q - 1) = 3y \bmod 20 = 1$, we can get $y = 7$. Consequently, the public key and the private key should be {3, 33} and {7, 33}, respectively. If we encrypt the message, we get $c = m^x \bmod n = 9^3 \bmod 33 = 3$. The decryption process is the reverse of this action, as $m = c^y \bmod n = 3^7 \bmod 33 = 9$.

8(B) - Discuss the Secure Hash Algorithm.

The *Secure Hash Algorithm* (SHA) was proposed as part of the digital signature standard. SHA-1, the first version of this standard, takes messages with a maximum length of 224 and produces a 160-bit digest. With this algorithm, SHA-1 uses five registers, *R*1 through *R*5, to maintain a "state" of 20 bytes.

The first step is to pad a message *m* with length *lm*. The message length is forced to $lm = 448 \bmod 512$. In other words, the length of the padded message becomes 64 bits less than the multiple of 512 bits. The number of padding bits can be as low as 1 bit and as high as 512 bits. The padding includes a 1 bit and as many 0 bits as required. Therefore, the least-significant 64 bits of the message length are appended to convert the padded message to a word with a multiple of 512 bits.

After padding, the second step is to expand each block of 512-bit (16 32 bits) words {*m*0, *m*1, ..., *m*15} to words of 80 32 bits using:

$$w_i = m_i \text{ for } 0 \le i \le 15$$

and

$$w_i = w_{i-3} \oplus w_{i-8} \oplus w_{i-14} \oplus w_{i-16} \hookleftarrow 1 \text{ for } 16 \le i \le 79,$$

Where *j* means left rotation by *j* bits. This way, bits are shifted several times if the incoming block is mixed with the state. Next, bits from each block of *wi* are mixed into the state in four steps, each maintaining 20 rounds. For any values of *a*, *b*, and *c*, and bit number *i*, we define a function *Fi* (*a*, *b*, *c*) as follows:

$$F_i(a,b,c) = \begin{cases} (a \cap b) \cup (\bar{a} \cap c) & 0 \le i \le 19 \\ a \oplus b \oplus c & 20 \le i \le 39 \\ (a \cap b) \cup (a \cap c) \cup (b \cap c) & 40 \le i \le 59 \\ a \oplus b \oplus c & 60 \le i \le 79 \end{cases}.$$

Then, the 80 steps ($i = 0, 1, 2, ..., 79$) of the four rounds are described as

$$\delta = (R_1 \leftrightarrow 5) + F_i(R_2, R_3, R_4) + R_5 + w_i + C_i$$

$$R_5 = R_4$$

$$R_4 = R_3$$

$$R_3 = R_2 \leftrightarrow 30$$

$$R_2 = R_1$$

$$R_1 = \delta,$$

where *Ci* is a constant value specified by the standard for round *i*. The message digest is produced by concatenation of the values in *R*1 through *R*5.

8(C) - Write a note on firewalls.

- Firewall is a security barrier between two networks that screens traffic coming in and out of the gate of one network to accept or reject connections and services according to a set of rules.

- A firewall is like a secretary for a network which examines requests for access to the network. It decides whether they pass a reasonableness test. If they pass it they are allowed through and if not they are refused.
- If a man wants to meet the chair of the community department, the secretary does a certain level of filtering but if the man wants to meet the President of the country, the secretary will perform a much different level of filtering.
- A network firewall is placed between the internal network, which might be considered safe and the external network or the Internet which is known to be unsafe.
- The job of the firewall is to determine what to let into and out of the internal network. In this way, a firewall provides access control for the network.
- There are essentially three types of firewalls. Each type of firewall filters packets by examining the data up to a particular layer of the network protocol stack.
- The firewalls are:
    - A packet filter is a firewall that operates at the network layer.
    - A stateful packet filter is a firewall that lives at the transport layer.
    - An application proxy is a firewall that operates at the application layer where it functions as a proxy.

Module – 5

9(A) - Briefly explain the properties of Audio and Video

Properties of Video

1) High Bit Rate

- Video distributed over the Internet use
  → 100 kbps for low-quality video conferencing.
  → 3 Mbps for streaming high-definition (HD) movies.
- The higher the bit-rate,
  → better the image quality and
  → better the overall user viewing experience.

2) Video Compression

- A video can be compressed, thereby trading off video-quality with bit-rate.
- A video is a sequence of images, displayed at a constant rate.
  For example: 24 or 30 images per second.
- An uncompressed digital image consists of an array of pixels.
- Each pixel is encoded into a number of bits to represent luminance and color.
- There are two types of redundancy in video:

1) Spatial Redundancy

- An image that consists of mostly white space has a high degree of redundancy.
- These images can be efficiently compressed without sacrificing image quality.

2) Temporal Redundancy

- Temporal redundancy reflects repetition from image to subsequent image.
- For exanple:

If image & subsequent image are same, re-encoding of subsequent image can be avoided.

Properties of Audio

- PCM (Pulse Code Modulation) is a technique used to change an analog signal to digital data (digitization).
- PCM consists of 1) Encoder at the sender and 2) Decoder at the receiver.

PCM Encoder

- Digital audio has lower bandwidth requirements than video.
- Consider how analog audio is converted to a digital-signal:
- The analog audio-signal is sampled at some fixed rate. This operation is referred to as sampling.
- For example: 8000 samples per second.
- The value of each sample is an arbitrary real number.
- Each sample is then rounded to one of a finite number of values. This process is called quantization.
- The number of such finite values is called as quantization-values.
- The number of quantization-values is typically a power of 2. For ex: 256(2_8) quantization-values.
- Each of the quantization-values is represented by a fixed number of bits.

• For example:
If there are 256(2_8) quantization-values, then each value is represented by 8 bits.
• Bit representations of all values are then concatenated to form digital representation of the signal.
This process is called encoding.

• For example:
If an analog-signal is sampled at 8000 samples per second & each sample is represented by 8 bits,
then the digital-signal will have a rate of 64000 bits per second (8000*8=64000).

PCM Decoder

• For playback through audio speakers, the digital-signal can be converted back to an analog-signal.
This process is called decoding.
• However, the decoded analog-signal is only an approximation of the original signal.
• The sound quality may be noticeably degraded.
• The decoded signal can better approximate the original analog-signal by increasing
i) sampling rate and
ii) number of quantization-values,
• Thus, there is a trade-off between
→ quality of the decoded signal and
→ bit-rate & storage requirements of the digital-signal.

9(B) - List the categories of streaming of stored video. Explain any one of them.

Streaming Stored Video

• Prerecorded videos are placed on servers.
• Users send requests to these servers to view the videos on-demand.
• The media is prerecorded, so the user may pause, reposition or fast-forward through video-content.
• Three categories of applications:
1) UDP streaming
2) HTTP streaming and
3) Adaptive HTTP streaming.

HTTP Streaming

• The video is stored in an HTTP server as an ordinary file with a specific URL.
• Here is how it works:
1) When a user wants to see the video, the client
→ establishes a TCP connection with the server and
→ issues an HTTP GET request for that URL.
2) Then, the server responds with the video file, within an HTTP response message.
3) On client side, the bytes are collected in a client application buffer.

4) Once no. of bytes in this buffer exceeds a specific threshold, the client begins playback.

• Advantages:

1) Not Costly & Complex

- Streaming over HTTP avoids the need for a media control server (RTSP).
- This reduces the cost of deploying a large-scale application.

2) No Firewall Problem

- The use of HTTP over TCP also allows the video to traverse firewalls and NATs more easily.

3) Prefetching Video

- The client downloads the video at a rate higher than the consumption rate.
- Thus, prefetching video-frames that are to be consumed in the future.
  - This prefetched video is stored in the client application buffer

• Nowadays, most video-streaming applications use HTTP streaming. For example: YouTube

9(C) - Explain the RTP protocol header fields.

RTP Packet Header Fields

• Four header fields of RTP Packet (Figure 5.6):

1) Payload type

2) Sequence number

3) Timestamp and

4) Source identifier.

• Header fields are illustrated in Figure

Figure9.C: RTP header fields

Payload-Type Number	Audio Format	Sampling Rate	Rate
0	PCM μ-law	8 kHz	64 kbps
1	1016	8 kHz	4.8 kbps
3	GSM	8 kHz	13 kbps
7	LPC	8 kHz	2.4 kbps
9	G.722	16 kHz	48–64 kbps
14	MPEG Audio	90 kHz	—
15	G.728	8 kHz	16 kbps

Table 9.C.1 : Audio payload types supported by RTP

Payload-Type Number	Video Format
26	Motion JPEG
31	H.261
32	MPEG 1 video
33	MPEG 2 video

Table 9.C.2 : Some video payload types supported by RTP

1) Payload Type

i) For an audio-stream, this field is used to indicate type of audio encoding that is being used.

- For example: PCM, delta modulation.
- Table 9.C.1 lists some of the audio payload types currently supported by RTP.

ii) For a video stream, this field is used to indicate the type of video encoding.

- For example: motion JPEG, MPEG.
- Table 9.C.2 lists some of the video payload types currently supported by RTP.

2) Sequence Number

• This field increments by one for each RTP packet sent.

• This field may be used by the receiver to detect packet loss and to restore packet sequence.

3) Timestamp

• This field reflects the sampling instant of the first byte in the RTP data packet.

• The receiver can use timestamps

→ to remove packet jitter in the network and

→ to provide synchronous playout at the receiver.

• The timestamp is derived from a sampling clock at the sender.

4) Source Identifier (SRC)

• This field identifies the source of the RTP stream.

• Typically, each stream in an RTP session has a distinct SRC.

10(A) - With neat diagram explain CDN operation.

When a browser wants to retrieve a specific video, the CDN intercepts the request.

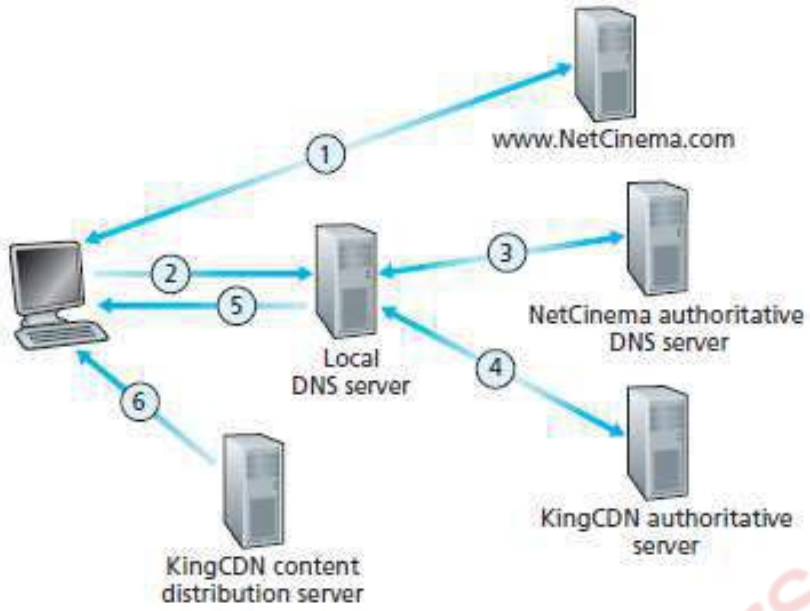
• Then, the CDN

1) determines a suitable server-cluster for the client and

2) redirects the client"s request to the desired server.

• Most CDNs take advantage of DNS to intercept and redirect requests.

• CDN operation is illustrated in Figure



• Suppose a content provider "NetCinema" employs the CDN company "KingCDN" to distribute videos.
• Let URL = http://video.netcinema.com/6Y7B23V
• Six events occur as shown in Figure 5.2:
1) The user visits the Web page at NetCinema.
2) The user clicks on the following link:
http://video.netcinema.com/6Y7B23V,
☐ Then, the user"s host sends a DNS query for "video.netcinema.com".
3) The user"s local-DNS-server (LDNS) forwards the DNS-query to an authoritative-DNS-server
"NetCinema".
☐ The server "NetCinema" returns to the LDNS a hostname in the KingCDN"s domain.
☐ For example: "a1105.kingcdn.com".
4) The user"s LDNS then sends a second query, now for "a1105.kingcdn.com".
☐ Eventually, KingCDN"s DNS system returns the IP addresses of a "KingCDN" server to LDNS.
5) The LDNS forwards the IP address of the "KingCDN" server to the user"s host.
6) Finally, the client
→ establishes a TCP connection with the server
→ issues an HTTP GET request for the video.

10(B) - Discuss the following (i) Adaptive Streaming (ii) DASH

(i)Adaptive Streaming - is a technique used in streaming multimedia over computer networks. While in the past most video or audio streaming technologies utilized streaming protocols such as RTP with RTSP, today's adaptive streaming technologies are almost exclusively based on HTTP and designed to work efficiently over large distributed HTTP networks such as the Internet. It works by detecting a user's bandwidth and CPU capacity in real time and adjusting the quality of the media stream accordingly. It requires the use of an encoder which can encode a single source media (video or audio) at multiple bit rates. The player client switches between streaming the different encodings depending on available resources. "The result: very little buffering, fast start time and a good experience for both high-end and low-end connections."

(ii) DASH - Dynamic Adaptive Streaming over HTTP (**DASH**), also known as **MPEG-DASH**, is an adaptive bitrate streaming technique that enables high quality streaming of media content over the Internet delivered from conventional HTTP web servers. Similar to Apple's HTTP Live Streaming (HLS) solution, MPEG-DASH works by breaking the content into a sequence of small segments, which are served over HTTP. Each segment contains a short interval of playback time of content that is potentially many hours in duration, such as a movie or the live broadcast of a sports event. The content is made available at a variety of different bit rates, i.e., alternative segments encoded at different bit rates covering aligned short intervals of playback time. While the content is being played back by an MPEG-DASH client, the client uses a bit rate adaptation (ABR) algorithm to automatically select the segment with the highest bit rate possible that can be downloaded in time for playback without causing stalls or re-buffering events in the playback. The current MPEG-DASH reference client dash.js offers both buffer-based (BOLA) and hybrid (DYNAMIC) bit rate adaptation algorithms. Thus, an MPEG-DASH client can seamlessly adapt to changing network conditions and provide high quality playback with few stalls or re-buffering events.

10(C) - Give the limitations of best effort IP service.

Limitations of the Best-Effort IP Service

- The Internet"s network-layer protocol IP provides best-effort service.
- The IP makes best effort to move each datagram from source to destination.
- But IP does not guarantee deliver of the packet to the destination.
- Three main challenges to the design of real-time applications:

1) Packet-loss
2) Packet delay and
3) Packet jitter.

Packet Loss

- By default, most existing VoIP applications run over UDP.
- The UDP segment is encapsulated in an IP datagram.
- The datagram passes through router buffers in the path from sender to receiver
- Problem:
  - There is possibility that one or more buffers are full.

- In this case, the arriving IP datagram may be discarded.

• Possible solution:
- Loss can be eliminated by sending the packets over TCP rather than over UDP.
- However, retransmissions are unacceptable for real-time applications „." they increase delay.
- Packet-loss results in a reduction of sender"s transmission-rate, leading to buffer starvation.

End-to-End Delay

• End-to-end delay is the sum of following delays:
1) Transmission, processing, and queuing delays in routers.
2) Propagation delays in links and
3) Processing delays in end-systems.

• For VoIP application,
→ delays smaller than 150 msecs are not perceived by a human listener.
→ delays between 150 and 400 msecs can be acceptable but are not ideal and
→ delays exceeding 400 msecs can seriously hinder the interactivity in voice conversations.

• Typically, the receiving-side will discard any packets that are delayed more than a certain threshold.
• For example: more than 400 msecs.

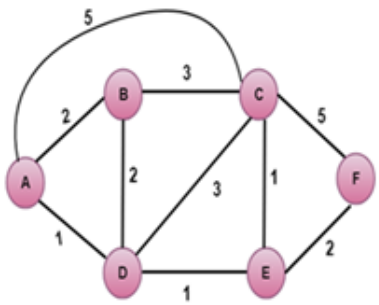
Packet Jitter

• Jitter refers to varying queuing delays that a packet experiences in the network"s routers.
• If the receiver
→ ignores the presence of jitter and
→ plays out audio-chunks,
then the resulting audio-quality can easily become unintelligible.
• Jitter can often be removed by using sequence numbers, timestamps, and a playout delay.

Answers to below questions are not available in this doc.

4(A) - Explain the stop and wait protocol with FSM representation rdt2.1

5(C) - Write the link state algorithm and apply it to the following graph with source node is 'A'



Model Question Paper-1 with effect from 2019-20 (CBCS Scheme)

Computer Networks and Security

Module – 1

1(A) - Differentiate between i) HTTP & FTP ii) SMTP & HTTP iii) UDP & TCP

i) HTTP & FTP

BASIS FOR COMPARISON	HTTP	FTP
Basic	HTTP is used to access websites.	FTP transfers file from one one host to another.
Connection	HTTP establishes data connection only.	FTP establishes two connection one for data and one for the control connection.
TCP ports	HTTP uses TCP's port number 80.	FTP uses TCP's port number 20 and 21.
URL	If you are using HTTP, http will appear in URL.	If you are using FTP, ftp will appear in URL.
Efficient	HTTP is efficient in transferring smaller files like web pages.	FTP is efficient in transferring larger files.
Authentication	HTTP does not require authentication.	FTP requires a password.
Data	The content transferred to a device using HTTP is not saved to the memory of that device.	The file transferred to the host device using FTP is saved in the memory of that host device.

ii) SMTP & HTTP

HTTP	SMTP
Pull Protocol- someone loads information on a Web server and users use HTTP to pull the information from the server at their convenience.	Push Protocol- the sending mail server pushes the file to the receiving mail server.
HTTP does not mandates data to be in 7-bit ASCII format.	SMTP requires each message, including the body of each message, to be in 7-bit ASCII format.
HTTP encapsulates each object in its own HTTP response message.	Internet mail places all of the message's objects into one message.

iii) UDP & TCP

TCP	UDP
It is a connection-oriented protocol.	It is a connectionless protocol.
TCP reads data as streams of bytes, and the message is transmitted to segment boundaries.	UDP messages contain packets that were sent one by one. It also checks for integrity at the arrival time.
TCP messages make their way across the internet from one computer to another.	It is not connection-based, so one program can send lots of packets to another.
TCP rearranges data packets in the specific order.	UDP protocol has no fixed order because all packets are independent of each other.
The speed for TCP is slower.	UDP is faster as error recovery is not attempted.
Header size is 20 bytes	Header size is 8 bytes.
TCP is heavy-weight. TCP needs three packets to set up a socket connection before any user data can be sent.	UDP is lightweight. There are no tracking connections, ordering of messages, etc.
TCP does error checking and also makes error recovery.	UDP performs error checking, but it discards erroneous packets.
Acknowledgment segments	No Acknowledgment segments
Using handshake protocol like SYN, SYN-ACK, ACK	No handshake (so connectionless protocol)
TCP is reliable as it guarantees delivery of data to the destination router.	The delivery of data to the destination can't be guaranteed in UDP.
TCP offers extensive error checking mechanisms because it provides flow control and acknowledgment of data.	UDP has just a single error checking mechanism which is used for checksums.

1(B) - Explain cookies and web caching with diagram.

Cookies

• Cookies refer to a small text file created by a Web-site that is stored in the user's computer.
• Cookies are stored either temporarily for that session only or permanently on the hard disk.
• Cookies allow Web-sites to keep track of users.
• Cookie technology has four components:
  1) A cookie header-line in the HTTP response-message.
  2) A cookie header-line in the HTTP request-message.
  3) A cookie file kept on the user's end-system and managed by the user's browser.
  4) A back-end database at the Web-site.

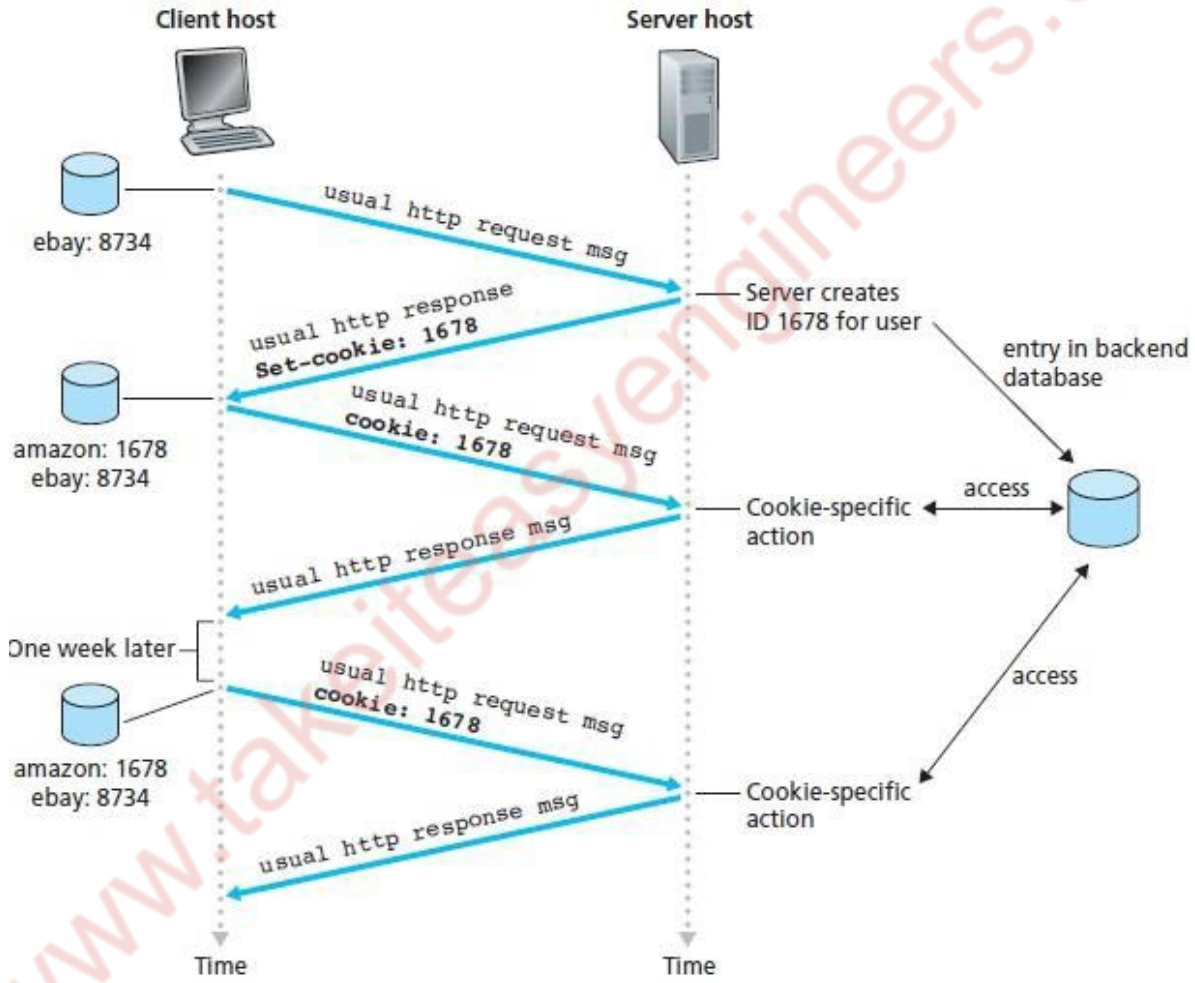


Figure 1.B: Keeping user state with cookies

• Here is how it works (Figure 1.B):
1) When a user first time visits a site, the server
 → creates a unique identification number (1678) and
 → creates an entry in its back-end database by the identification number.
2) The server then responds to user's browser.
 ☐ HTTP response includes Set-cookie: header which contains the identification number (1678)

3) The browser then stores the identification number into the cookie-file.
4) Each time the user requests a Web-page, the browser
→ extracts the identification number from the cookie file, and
→ puts the identification number in the HTTP request.
5) In this manner, the server is able to track user's activity at the web-site.

Web Caching

• A Web-cache is a network entity that satisfies HTTP requests on the behalf of an original Web-server.
• The Web-cache has disk-storage.
• The disk-storage contains copies of recently requested-objects.

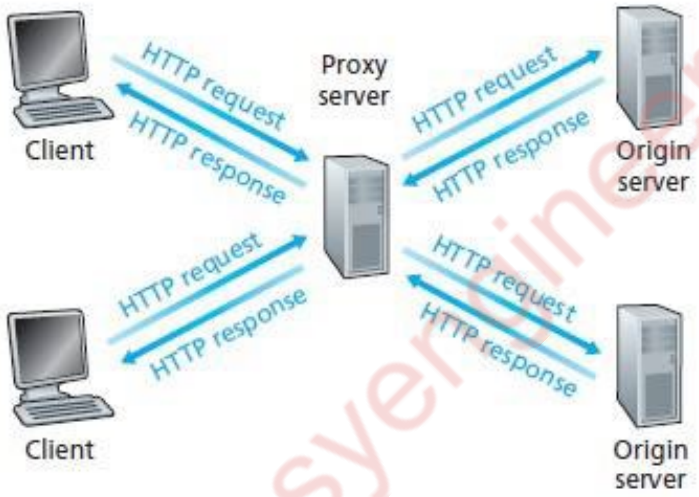


Figure 1.B.1: Clients requesting objects through a Web-cache (or Proxy Server)

• Here is how it works (Figure 1.B.1)
1) The user's HTTP requests are first directed to the web-cache.
2) If the cache has the object requested, the cache returns the requested-object to the client.
3) If the cache does not have the requested-object, then the cache
→ connects to the original server and
→ asks for the object.
4) When the cache receives the object, the cache
→ stores a copy of the object in local-storage and
→ sends a copy of the object to the client.
• A cache acts as both a server and a client at the same time.
1) The cache acts as a server when the cache
→ receives requests from a browser and
→ sends responses to the browser.
2) The cache acts as a client when the cache
→ requests to an original server and
→ receives responses from the origin server.
• Advantages of caching:
1) To reduce response-time for client-request.
2) To reduce traffic on an institution's access-link to the Internet.
3) To reduce Web-traffic in the Internet.

2(A) - Discuss the working of Domain Name Service.

All the hosts connected to network is identified by IP address. But it is difficult for human beings to remember these IP address to access a particular host. Hence hosts are identified by hostnames. Ex: google.com

But the routers require IP address to forward the packet.

In order to map hostname with the IP address DNS is used.

Overview of How DNS Works

☐ Suppose that some application running in a user's host needs to translate a hostname to an IP address. The application will invoke the client side of DNS, specifying the hostname that needs to be translated.

☐ DNS in the user's host then takes over, sending a query message into the network.

☐ All DNS query and reply messages are sent within UDP datagrams to port 53. After a delay, ranging from milliseconds to seconds, DNS in the user's host receives a DNS reply message that provides the desired mapping. This mapping is then passed to the invoking application.

In this centralized design, clients simply direct all queries to the single DNS server, and the DNS server responds directly to the querying clients. Although the simplicity of this design is attractive, it is inappropriate for today's Internet, with its vast (and growing) number of hosts. The problems with a centralized design include:

A Single Point of Failure

☐ If the DNS server crashes then the entire Internet will not stop.

Traffic Volume

☐ A Single DNS Server cannot handle the huge global DNS traffic.

☐ But with distributed system, the traffic is distributed and reduces overload on server.

Distant Centralized Database

☐ A single DNS server cannot be "close to" all the querying clients.

☐ If we put the single DNS server in Mysore,

then all queries from USA must travel to the other side of the globe.

☐ This can lead to significant delays.

Maintenance

☐ The single DNS server would have to keep records for all Internet hosts.

☐ This centralized database has to be updated frequently to account for every new host.

2(B) - Demonstrate client server socket programming application using TCP.

Ans --- Refer 1(B) – Page | 2

Module – 2

3(A) - Illustrate TCP & UDP segment structure with a help of diagram.

Ans --- Refer 3(B) – Page | 7 & 8 (TCP)

UDP Segment Structure

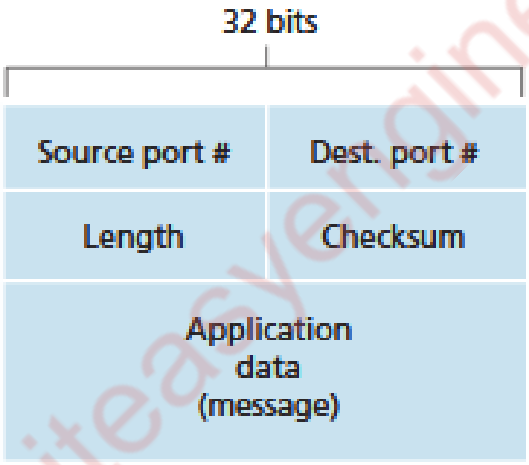


Fig 3.A: UDP Segment structure

• UDP Segment contains following fields (Figure 3.A):
1) Application Data: This field occupies the data-field of the segment.
2) Destination Port No: This field is used to deliver the data to correct process running on the
destination-host. (i.e. demultiplexing function).
3) Length: This field specifies the number of bytes in the segment (header plus data).
4) Checksum: This field is used for error-detection.

4(A) - Describe TCP connection management with a help of diagram.

Ans --- Refer 4(C) – Page | 10 & 11

Module - 3

5(A) - With a help of neat diagram explain virtual circuit diagram and Datagram network.

Virtual Circuit Networks

- A VC consists of
  1) A path between the source and destination.
  2) VC number: This is one number for each link along the path.
  3) Entries in the forwarding-table in each router.
- A packet belonging to a virtual-circuit will carry a VC number in its header.
- At intervening router, the VC number of traversing packet is replaced with a new VC number.
- The new VC number is obtained from the forwarding-table.

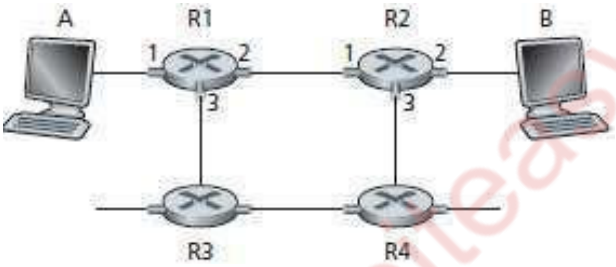


Figure 5.A: A simple virtual-circuit network

Incoming Interface	Incoming VC #	Outgoing Interface	Outgoing VC #
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87

Table 5.A.1: Forwarding-table in R1

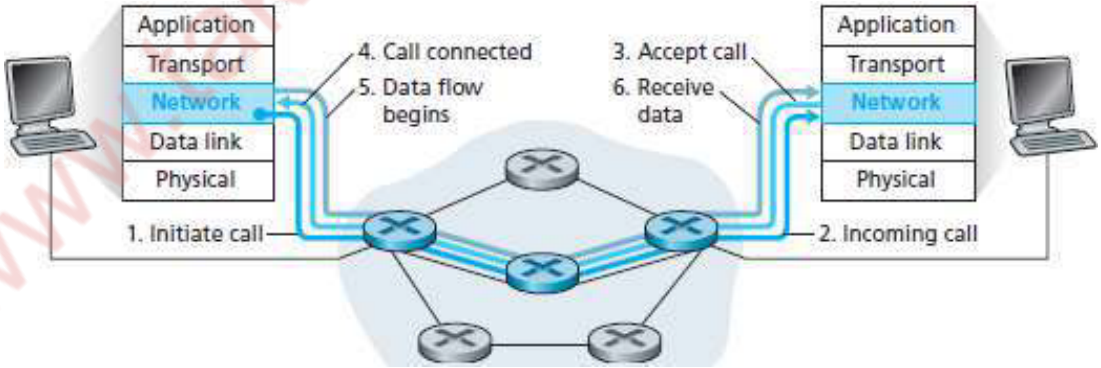


Figure 5.A.3: Virtual-circuit setup

Why a packet does not use the same VC number on each link along the path?
Answer:

1) Replacing the number from link to link reduces length of the VC field in the packet-header.
2) VC setup is simplified by permitting a different VC number at each link along the path.

• Disadvantage:
The routers must maintain connection state information for the ongoing connections.
• Three phases in a virtual-circuit (Figure 5.A.3):

1) VC Setup

- During the setup phase, the sending transport-layer
  → contacts the network-layer
  → specifies the receiver's address and
  → waits for the network to set-up the VC.
- The network-layer determines the path between sender and receiver.
- The network-layer also determines the VC number for each link along the path.
- Finally, the network-layer adds an entry in the forwarding-table in each router.
- During VC setup, the network-layer may also reserve resources.

2) Data Transfer

- Once the VC has been established, packets can begin to flow along the VC.

3) VC Teardown

- This is initiated when the sender/receiver wants to terminate the VC.
- The network-layer
  → informs the other end-system of the call termination and
  → removes the appropriate entries in the forwarding-table in each router.

Datagram Networks

• The source attaches the packet with the address of the destination.
• The packets are injected into the network.
• The packets are routed independent of each other.
• No advance circuit setup is needed. So, routers do not maintain any connection state information.
• As a packet is transmitted from source to destination, it passes through a series of routers.
• Each router uses the packet's destination-address to forward the packet.

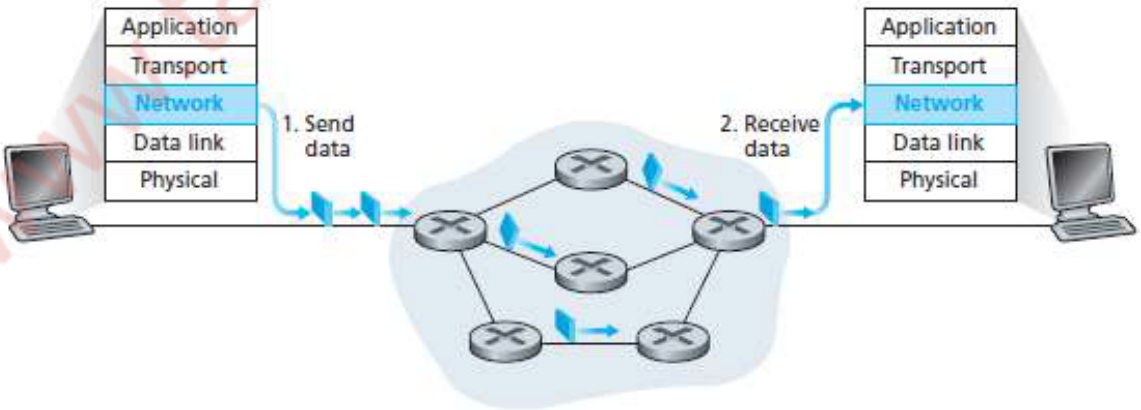


Figure 5.A.4: Datagram network

Comparison of Virtual Circuit & Datagram

Issue	Datagram	Virtual Circuit
Connection Setup	None	Required
Addressing	Packet contains full source and destination-address	Packet contains short virtual-circuit number identifier
State Information	None other than router table containing destination-network	Each virtual-circuit number entered to table on setup, used for routing
Routing	Packets routed independently	Route established at setup, all packets follow same route
Effect of Router Failure	Only on packets lost during crash	All virtual circuits passing through failed router terminated

5(B) - Explain router architecture.

• The router is used for transferring packets from an incoming-links to the appropriate outgoing-links.

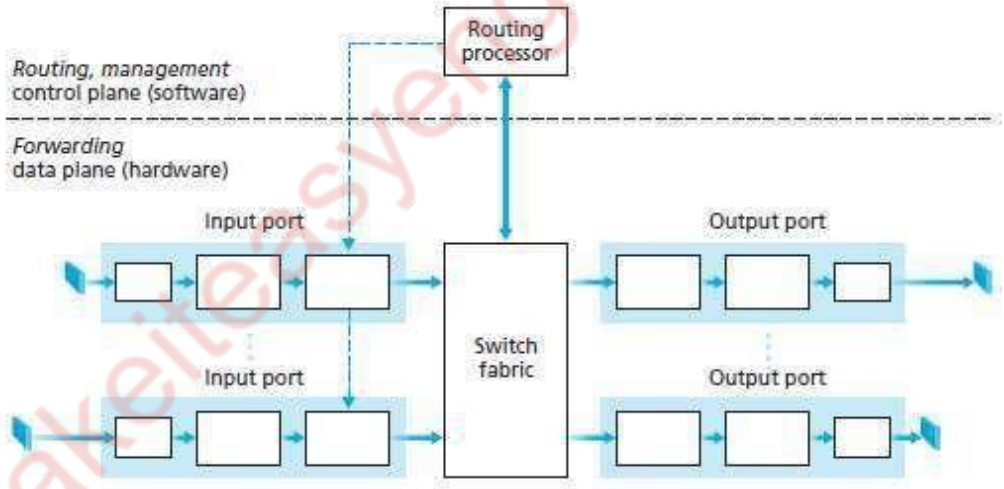


Figure 5.B: Router architecture

• Four components of router (Figure 5.B):

1) Input Ports

• An input-port is used for terminating an incoming physical link at a router (Figure 3.6).
• It is used for interoperating with the link layer at the other side of the incoming-link.
• It is used for lookup function i.e. searching through forwarding-table looking for longest prefix match.
• It contains forwarding-table.
• Forwarding-table is consulted to determine output-port to which arriving packet will be forwarded.
• Control packets are forwarded from an input-port to the routing-processor.
• Many other actions must be taken:
  i) Packet's version number, checksum and time-to-live field must be checked.

ii) Counters used for network management must be updated.

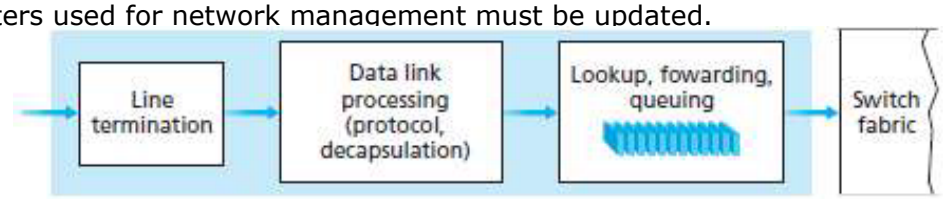


Figure 5.B.1: Input port processing

2) Switching Fabric

• The switching fabric connects the router's input-ports to its output-ports.
• In fabric, the packets are switched (or forwarded) from an input-port to an output-port.
• In fact, fabric is a network inside of a router.
• A packet may be temporarily blocked if packets from other input-ports are currently using the fabric.
• A blocked packet will be queued at the input-port & then scheduled to send at a later point in time.

3) Output Ports

• An output-port
→ stores packets received from the switching fabric and
→ transmits the packets on the outgoing-link.
• For a bidirectional link, an output-port will typically be paired with the input-port.

4) Routing Processor

• The routing-processor
→ executes the routing protocols
→ maintains routing-tables & attached link state information and
→ computes the forwarding-table.
• It also performs the network management functions.

5(C) - Illustrate the following i)IPv4 Addressing ii)IP fragmentation iii)Subnet Addressing.

i)IPv4 Addressing -

• IP address is a numeric identifier assigned to each machine on the internet.
• IP address consists of two parts: network ID(NID) and host ID(HID).
1) NID identifies the network to which the host is connected. All the hosts connected to the
same network have the same NID.
2) HID is used to uniquely identify a host on that network.
• HID is assigned by the network-administrator at the local site.
NID for an organization may be assigned by the ISP (Internet Service Provider).
• IPv4 uses 32-bit addresses, i.e., approximately 4 billion addresses (2_{32}).

• IP addresses are usually written in dotted-decimal notation. The address is broken into four bytes.
For example, an IP address of
10000000 10000111 01000100 00000101
is written as
128.135.68.5
• IP address can be classified as
1) Classful IP addressing &
2) Classless IP addressing (CIDR → Classless Inter Domain Routing)

ii)IP fragmentation –

• Each network imposes a restriction on maximum size of packet that can be carried. This is called the
MTU (maximum transmission unit).
• For example:
MTU Ethernet = 1500 bytes
MTU FDDI = 4464 bytes
• Fragmentation means
"The datagram is divided into smaller fragments when size of a datagram is larger than MTU"
• Each fragment is routed independently (Figure 5.C.1).
• A fragmented datagram may be further fragmented, if it encounters a network with a smaller MTU.
• Source/router is responsible for fragmentation of original datagram into the fragments. Only destination is responsible for reassembling the fragments into the original datagram.

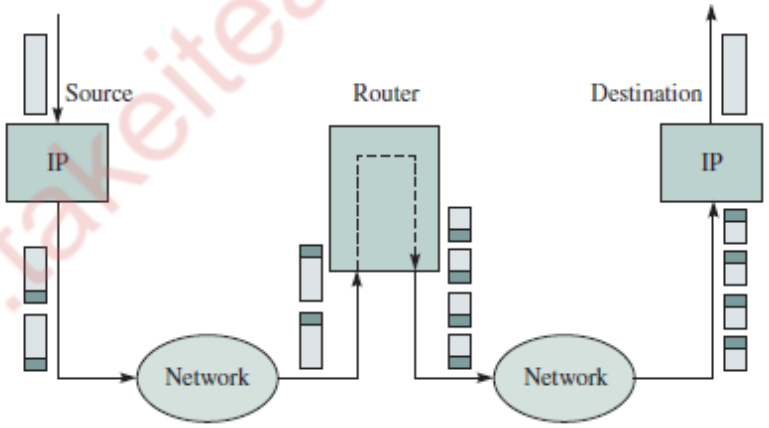


Figure 5.C.1- IP fragmentation and reassembly

iii) Subnet Addressing -

Subnetting reduces the total number of network-numbers by assigning a single network-number to
many adjacent physical networks.

• Each adjacent physical network is referred to as subnet. (Figure 5.C.2).
• All nodes on a subnet are configured with a subnet mask. For example: 255.255.255.0.
• The 1's in the subnet-mask represent the positions that refer to the network or subnet-numbers.
The 0's represent the positions that refer to the host part of the address.
• The bitwise AND of IP address and its subnet mask gives the subnet number.
• Advantage:
  - The subnet-addressing scheme is oblivious to the network outside the organization.
  - Inside the organization the network-administrator is free to choose any combination of lengths for the subnet & host ID fields.

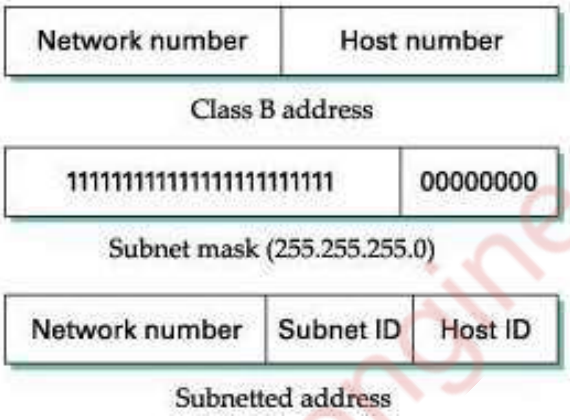


Figure 5.C.2: Subnet addressing

6(A) - Explain Dijkstra's algorithm with example.

• Dijkstra's algorithm computes the least-cost path from one node to all other nodes in the network.
• Let us define the following notation:
  1) u: source-node
  2) D(v): cost of the least-cost path from the source u to destination v.
  3) p(v): previous node (neighbor of v) along the current least-cost path from the source to v.
  4) N': subset of nodes; v is in N' if the least-cost path from the source to v is known.

Link-State (LS) Algorithm for Source Node u

```
1  Initialization:
2    N' = {u}
3    for all nodes v
4      if v is a neighbor of u
5        then D(v) = c(u,v)
6      else D(v) = ∞
7
8  Loop
9    find w not in N' such that D(w) is a minimum
10   add w to N'
11   update D(v) for each neighbor v of w and not in N':
12        D(v) = min( D(v), D(w) + c(w,v) )
13   /* new cost to v is either old cost to v or known
14    least path cost to w plus cost from w to v */
15 until N'= N
```

• Example: Consider the network in Figure 6.A and compute the least-cost paths from u to all possible destinations.

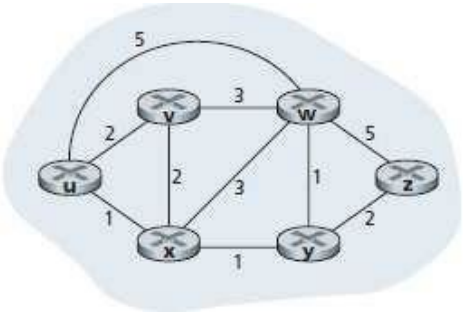


Figure 6.A: Abstract graph model of a computer network

Solution:

• Let's consider the few first steps in detail.

1) In the initialization step, the currently known least-cost paths from u to its directly attached neighbors, v, x, and w, are initialized to 2, 1, and 5, respectively.

2) In the first iteration, we
   → look among those nodes not yet added to the set N' and
   → find that node with the least cost as of the end of the previous iteration.

3) In the second iteration,
   → nodes v and y are found to have the least-cost paths (2) and
   → we break the tie arbitrarily and
   → add y to the set N' so that N' now contains u, x, and y.

4) And so on. . . .

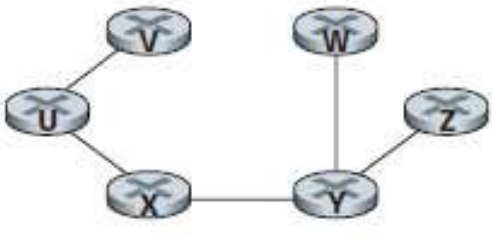
5) When the LS algorithm terminates,

We have, for each node, its predecessor along the least-cost path from the source.

• A tabular summary of the algorithm's computation is shown in Table 6.A.1

Table 6.A.1

step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Destination	Link
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)

Figure 6.A.1: Least cost path and forwarding-table for node u

6(B) - Explain various broadcast routing algorithms.

N-way Unicast

• Given N destination-nodes, the source-node
→ makes N copies of the packet and
→ transmits then the N copies to the N destinations using unicast routing (Figure 3.31).

• Disadvantages:

1) Inefficiency

☐ If source is connected to the n/w via single link, then N copies of packet will traverse this link.

2) More Overhead & Complexity

☐ An implicit assumption is that the sender knows broadcast recipients and their addresses.

☐ Obtaining this information adds more overhead and additional complexity to a protocol.

3) Not suitable for Unicast Routing

☐ It is not good idea to depend on the unicast routing infrastructure to achieve broadcast.

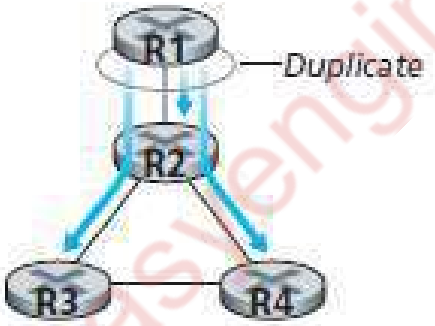


Figure 6.B.1: Duplicate creation/transmission

For other broadcast routing algorithms refer --- 6(B) Page | 17 & 18

Module - 4

7(B) - Explain RSA Algorithm with an example.

Rivert, Shamir, and Aldeman developed the RSA public-key encryption and signature scheme. This was the first practical public-key encryption algorithm. RSA is based on the intractability of factoring large integers. Assume that a plaintext *m* must be encrypted to a ciphertext *c*. The RSA algorithm has three phases for this: *key generation*, *encryption*, and *decryption*.

Key Generation

In the RSA scheme, the key length is typically 512 bits, which requires an enormous computational power. A plaintext is encrypted in blocks, with each block having a binary

value less than some number n. Encryption and decryption are done as follows, beginning with the generation of a public key and a private key.

Begin Key Generation Algorithm

1. Choose two roughly 256-bit prime numbers, a and b, and derive $n = ab$. (A number is prime if it has factors of 1 and itself.)

2. Find x. Select encryption key x such that x and $(a - 1)(b - 1)$ are relatively prime. (Two numbers are relatively prime if they have no common factor greater than 1.)

3. Find y. Calculate decryption key y:

(10.5) $$xy \bmod (a-1)(b-1) = 1.$$

4. At this point, a and b can be discarded.

5. The public key = {x, n}.

6. The private key = $\{y, n\}$.

In this algorithm, x and n are known to both sender and receiver, but only the receiver must know y. Also, a and b must be large and about the same size and both greater than 1,024 bits. The larger these two values, the more secure the encryption.

Encryption

Both sender and receiver must know the value of n. The sender knows the value of x, and only the receiver knows the value of y. Thus, this is a public-key encryption, with the public key $\{x, n\}$ and the private key $\{y, n\}$. Given $m < n$, ciphertext c is constructed by

(10.6) $$c = m^x \bmod n.$$

Note here that if a and b are chosen to be on the order of 1,024 bits, $n \approx 2, 048$. Thus, we are not able to encrypt a message longer than 256 characters.

Decryption

Given the ciphertext, c, the plaintext, m, is extracted by (10.7) $m = c^y \bmod n.$

In reality, the calculations require a math library, as numbers are typically huge. One can see easily how Equations (10.6) and (10.7) work.

Example. For an RSA encryption of a 4-bit message of 1,000, or $m = 9$, we choose $a = 3$ and $b = 11$. Find the public and the private keys for this security action, and show the ciphertext.

Solution. Clearly, $n = ab = 33$. We select $x = 3$, which is relatively prime to $(a - 1)(b - 1) = 20$. Then, from $xy \bmod (a - 1)(b - 1) = 3y \bmod 20 = 1$, we can get $y = 7$. Consequently, the public key and the private key should be {3, 33} and {7, 33}, respectively. If we encrypt the message, we get $c = mx \bmod n = 93 \bmod 33 = 3$. The decryption process is the reverse of this action, as $m = cy \bmod n = 37 \bmod 33 = 9$.

8(A) - Explain Diffie-Hellman Key-Exchange Protocol.

In the *Diffie-Hillman key-exchange* protocol, two end users can agree on a shared secret code without any information shared in advance. Thus, intruders would not be able to access the transmitted communication between the two users or discover the shared secret code. This protocol is normally used for *virtual private networks* (VPNs), explained in Chapter 16. The essence of this protocol for two users, 1 and 2, is as follows. Suppose that user 1 selects a prime a, a random integer number $x1$, and a generator g and creates $y1$ $\{1, 2, ..., a - 1\}$ such that

$$y_1 = g^{x_1} \bmod a. \quad (10.8)$$

In practice, the two end users agree on a and g ahead of time. User 2 performs the same function and creates y_2:

$$y_2 = g^{x_2} \bmod a. \quad (10.9)$$

User 1 then sends y_1 to user 2. Now, user 1 forms its key, k_1, using the information its partner sent as

$$k_1 = y_2^{x_1} \bmod a, \quad (10.10)$$

and user 2 forms its key, k_2, using the information its partner sent it as

$$k_2 = y_1^{x_2} \bmod a. \quad (10.11)$$

It can easily be proved that the two Keys k_1 and k_2 are equal. Therefore, the two users can now encrypt their messages, each using its own key created by the other one's information.

8(B) - With a help of neat diagram explain computation of SHA-1.

Ans --- Refer 8(B) - Page | 22 & 23

8(C) - Explain different types of Firewall.

Packet-Filtering Firewalls
This is the oldest firewall type out there. They are designed to create checkpoints at individual routers or switches. The packet-filtering firewalls will check the data packets that try to come through, without inspecting the contents. If the information trying to come through looks suspicious, it cannot get through the network. This is a simple firewall that does not impact network performance too much.

Circuit-Level Gateways
Circuit-level gateways are much like packet-filtering firewalls in that they quickly and easily check and approve or deny traffic. They do it without being heavy on resources, too. Circuit-level gateways work by verifying the transmission control protocol handshake. It doesn't check the packet directly, so there is a risk of malware getting through. These are not the best ones to protect your business.

Stateful Inspection Firewalls
A combination of the two firewalls above, the stateful inspection firewalls offer a higher level of protection for your business. The problem with these is that they take up more resources, which can slow down the legitimate packet transfer.

Proxy Firewalls (Application-Level Gateways/Cloud Firewalls)
If you want firewalls that operate at the application layer to filter traffic, proxy firewalls do the job. These are cloud-based most of the time, and they establish traffic connections and examine data packets coming through. The difference between these and the stateful inspection firewalls is that the proxy firewalls can also do a more in-depth inspection to check the packet contents. The drawback to these is that they can create a network slowdown because of all the extra steps – but it's all in the name of the security for your business.

Next-Generation Firewalls
There's no real insight into what makes a firewall today "next-generation" besides the time it was created. There are commonalities between these firewalls and the originals, and those include TCP handshakes and packet inspections. Next-generation firewalls also use IPS – intrusion prevention systems – to stop network attacks.

Software Firewalls
These are any firewalls installed on local devices. The biggest draw for these in that they can create a useful, in-depth defense path. Maintaining these on more than one device is not easy, though, so you may need more than one for each asset.

Hardware Firewalls
Hardware firewalls use physical appliances, and they act like a traffic router. The intercept data packets before they are connected to a network server. The weakness here is that they can be easily bypassed, which goes against your need for a firewall.

Cloud Firewalls

Cloud solutions are also called FaaS – firewalls as a service. They often go hand in hand with proxy firewalls, and the most significant benefit to these is that they grow with your business. They work to filter large amounts of traffic away from your company, where it's malicious.

Module – 5

9(A) - Explain the properties of audio and video.

Ans --- Refer 9(A) – Page | 24 & 25

9(B) - With a help of neat diagram explain streaming stored video over HTTP/TCP.

Ans --- Refer 9(B) – Page | 25 & 26

9(C) - Explain CDN Operation.

Ans --- Refer 10(A) – Page | 27 & 28

10(A) - Explain Interleaving mechanism.

- A VoIP application can send interleaved audio.
- The sender resequences units of audio-data before transmission.
- Thus, originally adjacent units are separated by a certain distance in the transmitted-stream.
- Interleaving can mitigate the effect of packet-losses.
- Interleaving is illustrated in Figure 5.5.
- For example:

If units are 5 msecs in length and chunks are 20 msecs (that is, four units per chunk), then

→ the first chunk contains the units 1, 5, 9, and 13

→ the second chunk contains the units 2, 6, 10 & 14 and so on.

- Advantages:
  1) Improves the perceived quality of an audio-stream.
  2) Low overhead.
  3) Does not increase the bandwidth requirements of a stream.
- Disadvantage:
  1) Increases latency. This limits use for VoIP applications.

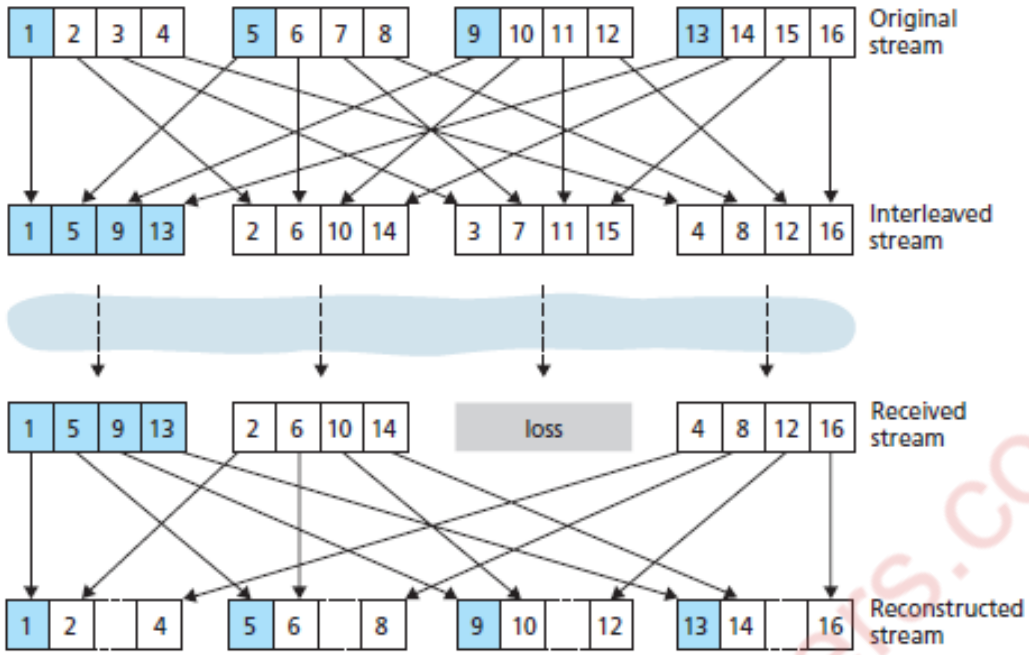


Figure 10.A : Sending interleaved audio

10(B) - Explain RTP Basics and RTP Packet.

RTP Basics

• RTP runs on top of UDP.

• The RTP packet is composed of i) RTP header & ii) audio chunk

• The header includes

i) Type of audio encoding

ii) Sequence number and

iii) Timestamp.

• The application appends each chunk of the audio-data with an RTP header.

• Here is how it works:

1) At sender-side:

i) A media chunk is encapsulated within an RTP packet.

ii) Then, the packet is encapsulated within a UDP segment.

iii) Finally, the UDP segment is handed over to IP.

2) At receiving-side:

i) The RTP packet is extracted from the UDP segment.

ii) Then, the media chunk is extracted from the RTP packet.

iii) Finally, the media chunk is passed to the media-player for decoding and rendering

• If an application uses RTP then the application easily interoperates with other multimedia applications

For RTP Packet refer 9(C) – Page | 26 & 27

10(C) - With a diagram, explain SIP call establishment.

• SIP (Session Initiation Protocol) is an open and lightweight protocol.
• Main functions of SIP:
1) It provides mechanisms for establishing calls b/w a caller and a callee over an IP network.
2) It allows the caller to notify the callee that it wants to start a call.
3) It allows the participants to agree on media encodings.
4) It also allows participants to end calls.
5) It provides mechanisms for the caller to determine the current IP address of the callee.
6) It provides mechanisms for call management, such as
→ adding new media streams during the call
→ changing the encoding during the call
→ inviting new participants during the call,
→ call transfer and
→ call holding.

Setting up a Call to a Known IP Address
• SIP call-establishment process is illustrated below Figure.

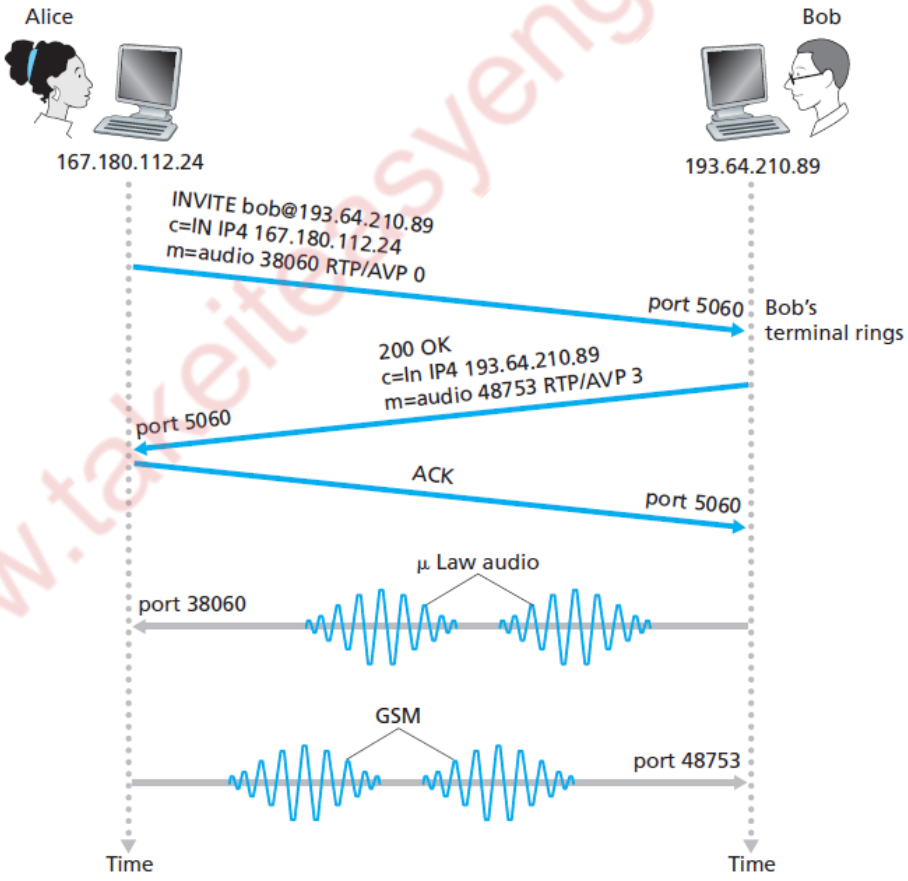


Figure 10.C : SIP call establishment when Alice knows Bob"s IP address

• Consider an example: Alice wants to call Bob.
• Alice“s & Bob“s PCs are both equipped with SIP-based software for making and receiving phone calls.
• The following events occur:

1) An SIP session begins when Alice sends Bob an INVITE message.

- This INVITE message is sent over UDP to the well-known port 5060 for SIP.
- The INVITE message includes

i) An identifier for Bob (bob@193.64.210.89)
ii) An indication of Alice“s current IP address
iii) An indication that Alice desires to receive audio, which is encoded in format AVP 0.

2) Then, Bob sends an SIP response message (which resembles an HTTP response message).

- The response message is sent over UDP to the well-known port 5060 for SIP.
- The response message includes

i) 200 OK
ii) An indication of Bob“s current IP address
iii) An indication that Bob desires to receive audio, which is encoded in format AVP 3.

3) Then, Alice sends Bob an SIP acknowledgment message.
4) Finally, Bob and Alice can talk.
• Three key characteristics of SIP:

1) SIP is an out-of-band protocol
   - The SIP message & the media-data use different sockets for sending and receiving.
2) The SIP messages are ASCII-readable and resemble HTTP messages.
3) SIP requires all messages to be acknowledged, so it can run over UDP or TCP.

Answers to below questions are not available in this doc.

3(B) - With a neat diagram, demonstrate the working of GO-BACK-N protocol.

4(B) - Interpret the FSM of TCP congestion control.

7(A) - Explain Feistel structure of DES Algorithm.