Fifth Semester B.E. Degree Examination, Feb./Mar. 2022
Computer Networks and Security

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. Differentiate between :
(i) HTTP and FTP (ii) SMTP and HTTP (iii) UDP and TCP (10 Marks)
b. Explain Cookies and Web Caching with diagram. (10 Marks)

OR

2 a. Describe in detail the services offered by DNS and explain DNS message format. (08 Marks)
b. Compare HTTP and SMTP. (04 Marks)
c. Define Socket. Demonstrate the working of TCP Socket. (08 Marks)

Module-2

3 a. With the help of FSM, describe the two states of the sender side and one state of the receiver side of rdt2.0 (10 Marks)
b. With a neat diagram, demonstrate the working of Go-BACK-N protocol. (10 Marks)

OR

4 a. Describe TCP connection management with help of diagram. (10 Marks)
b. Interpret the FSM to TCP congestion control. (10 Marks)

Module-3

5 a. Explain the Implementation of virtual circuit services in Computer Network. (07 Marks)
b. Explain the three Switching Techniques. (06 Marks)
c. Explain Distance vector algorithm using three nodes network. (07 Marks)

OR

6 a. Explain Dijkstra's algorithm with example. (10 Marks)
b. Explain various broadcast routing algorithms. (10 Marks)

Module-4

7 a. Explain Feistel structure of DES Algorithm. (10 Marks)
b. Explain RSA Algorithm with an example. (10 Marks)

OR

8 a. In the Diffie - Hellman key exchange protocol prove that the two keys k_1 and k_2 are equal. (10 Marks)

b. Discuss the following :
(i) Secure Hash Algorithm (ii) Firewalls. (10 Marks)

Module-5

9 a. Explain briefly how DNS redirects a users request to a CDN server. (10 Marks)
b. With neat diagram explain the naïve-architecture for audio/video streaming. (10 Marks)

OR

10 a. Write a short notes on :
(i) Netflix video streaming platform (ii) VOIP with Skype. (10 Marks)
b. With neat diagram explain the RTP header fields. (10 Marks)

* * * * *

18CS52- COMPUTER NETWORKS AND SECURITY

1. a. Differentiate between the following

i. HTTP and FTP

FTP	HTTP
• FTP sends connection information out-of-band as it uses two parallel TCPs to transfer files. The **control connection** is used to send information like passwords, and the **data connection** is used to send actual data. • The FTP server maintains state information like the user's current directory for a particular session. • The client-side needs to authenticate itself in order to transfer information. • The server keeps track of a users' state, which constrains the total number of sessions that FTP can maintain simultaneously.	• HTTP transfers control information in-band as it uses the same connection to transfer data as it does to control information. • HTTP is stateless. The server does not need to keep track of any user's state. • The client may or may not authenticate themselves, i.e., client authentication is not mandatory. • Because HTTP is stateless, it can easily maintain multiple sessions simultaneously.

ii. HTTP and SMTP

SMTP vs HTTP

SMTP
- persistent connections
- 7-bit ASCII request/response + status codes
- CRLF.CRLF for end of message
- Push
- Multiple objects sent in multipart message

HTTP
- persistent or non-persistent
- ASCII request/response + status codes
- CRLF or CRLFCRLF for end of message
- Pull
- Single object encapsulated in its own response message

iii. TCP and UDP

Sr. No.	Key	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
1	Definition	It is a communications protocol, using which the data is transmitted between systems over the network. In this, the data is transmitted into the form of packets. It includes error-checking, guarantees the delivery and preserves the order of the data packets.	It is same as the TCP protocol except this doesn't guarantee the error-checking and data recovery. If you use this protocol, the data will be sent continuously, irrespective of the issues in the receiving end.
2	Design	TCP is a connection oriented protocol.	UDP is a connection less protocol.
3	Reliable	As TCP provides error checking support and also guarantees delivery of data to the destination router this make it more reliable as compared to UDP.	While on other hand UDP does provided only basic error checking support using checksum so the delivery of data to the destination cannot be guaranteed in UDP as compared to that in case of TCP.
4	Data transmission	In TCP the data is transmitted in a particular sequence which means that packets arrive in-order at the receiver.	On other hand there is no sequencing of data in UDP in order to implement ordering it has to be managed by the application layer.
5	Performance	TCP is slower and less efficient in performance as compared to UDP. Also TCP is heavy-weight as compared to UDP.	On other hand UDP is faster and more efficient than TCP.
6	Retransmission	Retransmission of data packets is possible in TCP in case packet get lost or need to resend.	On other hand retransmission of packets is not possible in UDP.

1 b. Explain cookies and web caching with diagram

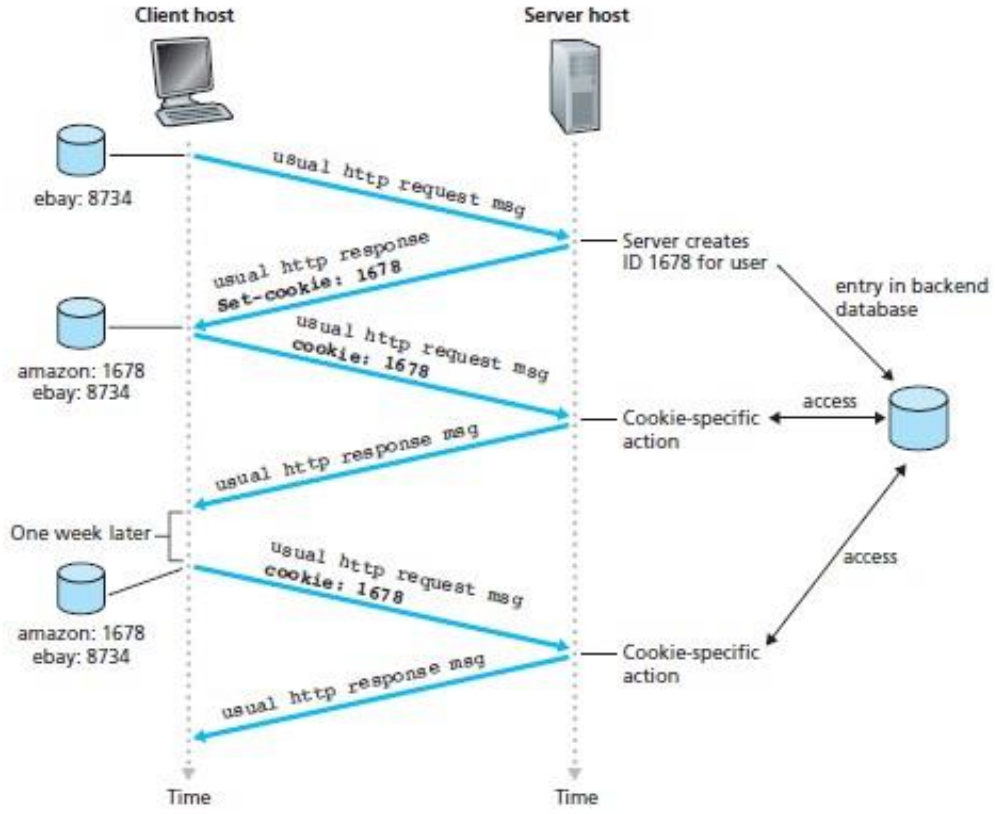
Cookies:

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.

Suppose a user, who always accesses the Web using Internet Explorer from her home PC, contacts
Amazon.com for the first time. Let us suppose that in the past he has already visited the eBay site.
When the request comes into the Amazon Web server, the server creates a unique identification number and creates an entry in its back-end database that is indexed by the identification number.

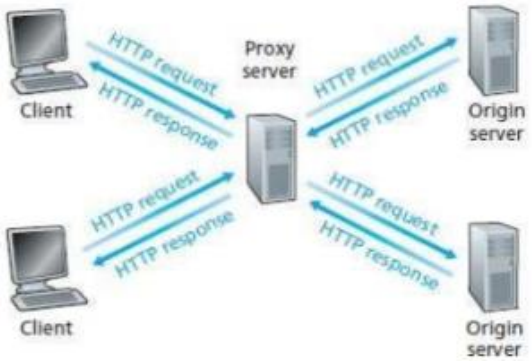
The Amazon Web server then responds to Susan's browser, including in the HTTP response a Set-
cookie: header, which contains the identification number.

For example, the header line might be:
Set-cookie: 1678
When users browser receives the HTTP response message, it sees the Set-cookie: header. The browser then appends a line to the special cookie file that it manages. This line includes the hostname of the server and the identification number in the Set-cookie: header.As user continues to browse the Amazon site, each time he requests a Web page, his browser
consults his cookie file, extracts his identification number for this site, and puts a cookie header line
that includes the identification number in the HTTP request. Specifically, each of his HTTP requests to the Amazon server includes the header line:
Cookie: 1678



Web Caching:

- A Web cache—also called a proxy server—is a network entity that satisfies HTTP requests on the behalf of an origin Web server.
- The Web cache has its own disk storage and keeps copies of recently requested objects in this storage.
- A user's browser can be configured so that all of the user's HTTP requests are first directed to the Web cache.



Ex: Suppose a browser is requesting the object http://www.someschool.edu/campus.gif. Here is what happens:

1. The browser establishes a TCP connection to the Web cache and sends an HTTP request for the object to the Web cache.
2. The Web cache checks to see if it has a copy of the object stored locally. If it does, the Web cache returns the object within an HTTP response message to the client browser.
3. If the Web cache does not have the object, the Web cache opens a TCP connection to the origin server, that is, to www.someschool.edu. The Web cache then sends an HTTP request for the object into the cache-to-server TCP connection.
4. After receiving this request, the origin server sends the object within an HTTP response to the Web cache.
5. When the Web cache receives the object, it stores a copy in its local storage and sends a copy, within an HTTP response message, to the client browser (over the existing TCP connection between the client browser and the Web cache).

- When web cache receives requests from and sends responses to a browser, it is a server. When it sends requests to and receives responses from an origin server, it is a client.
- Typically a Web cache is purchased and installed by an ISP. For example, a university might install a cache on its campus network and configure all of the campus browsers to point to the cache. Or a major residential ISP (such as AOL) might install one or more caches in its network and pre configure its shipped browsers to point to the installed caches.
- Web caching has seen deployment in the Internet for two reasons. First, a Web cache can substantially reduce the response time for a client request. Second, Web caches can substantially reduce traffic on an institution's access link to the Internet.

2a. Describe in detail the services offered by DNS and explain the DNS message format

DNS provides a few other important services in addition to translating hostnames to IP addresses:

Host aliasing: A host with a complicated hostname can have one or more alias names. For example, a hostname such as relay1.west-coast.enterprise.com could have, say, two aliases such as enterprise.com and www.enterprise.com.

Mail server aliasing :
(for example, the canonical hostname might be something like relay1.west-coast.hotmail.com). DNS can be invoked by a mail application to obtain the canonical hostname for a supplied alias hostname as well as the IP address of the host

Load distribution: DNS is also used to perform load distribution among replicated servers, such as replicated Web servers.

DNS Message Format:

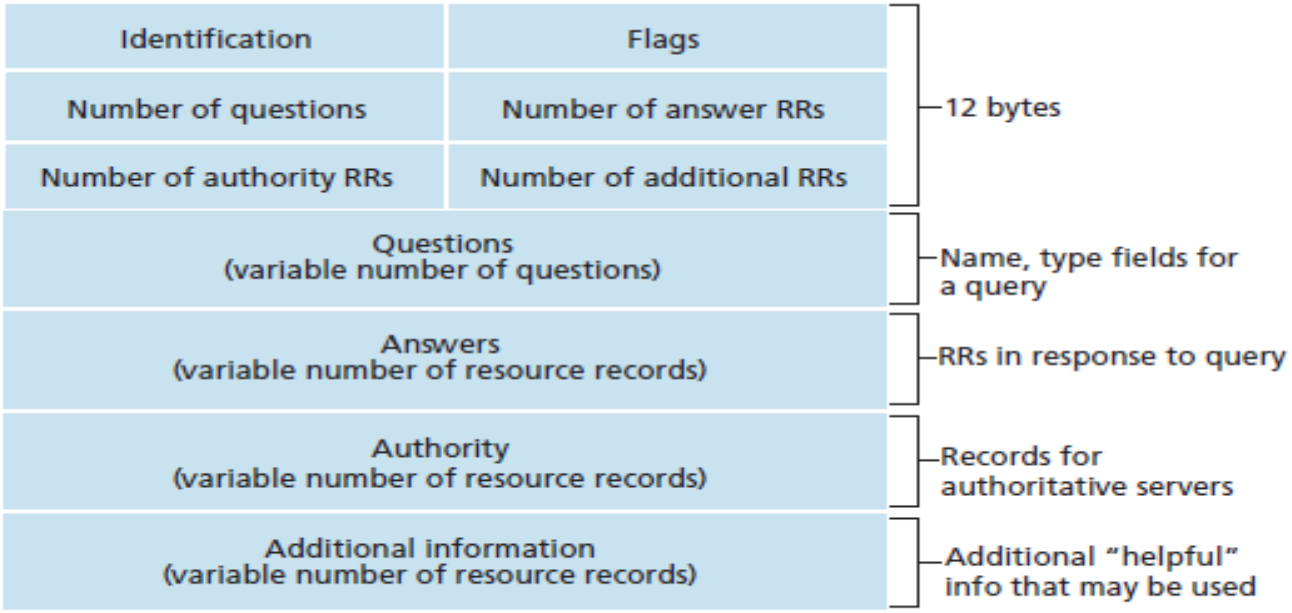


Figure 2.23 ♦ DNS message format

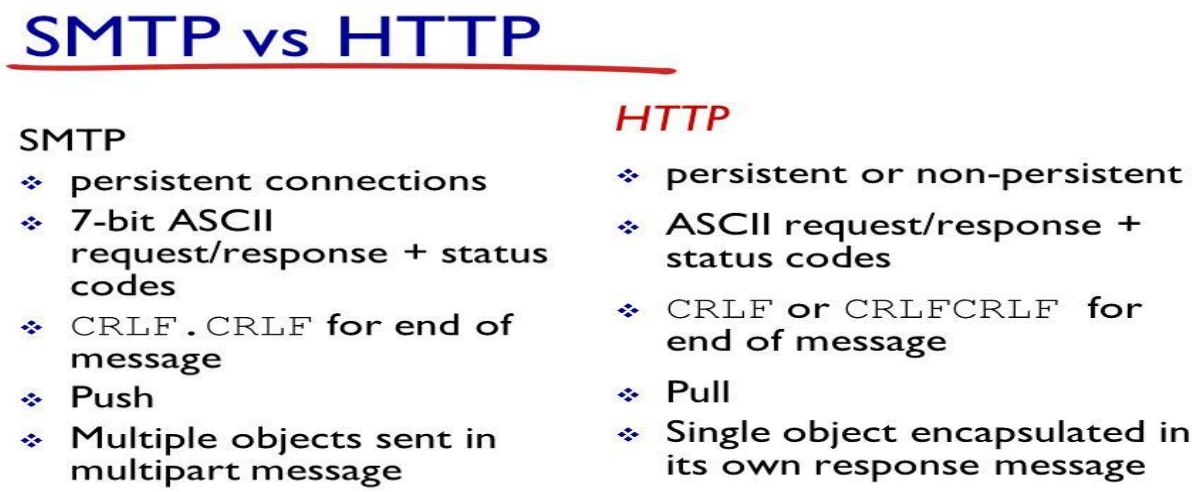
- The first 12 bytes is the header section, which has a number of fields.
- The first field is a 16-bit number that identifies the query. This identifier is copied into the reply message to a query, allowing the client to match received replies with sent queries.
- There are a number of flags in the flag field.

  A 1-bit query/reply flag indicates whether the message is a query (0) or a reply (1). A1-bit authoritative flag is set in a reply message when a DNS server is an authoritative server for a queried name.

  A 1-bit recursion-desired flag is set when a client (host or DNS server) desires that the DNS server perform recursion when it doesn't have the record.

  A 1-bit recursion available field is set in a reply if the DNS server supports recursion.
- In the header, there are also four number-of fields. These fields indicate the number of occurrences of the four types of data sections that follow the header.
- The **question** section contains information about the query that is being made. This section includes (1) a name field that contains the name that is being queried, and (2) a type field that indicates the type of question being asked about the name
- In a reply from a DNS server, the **answer** section contains the resource records for the name that was originally queried.

2b. Compare HTTP and SMTP



2c. Define socket. Demonstrate the working of TCP Socket

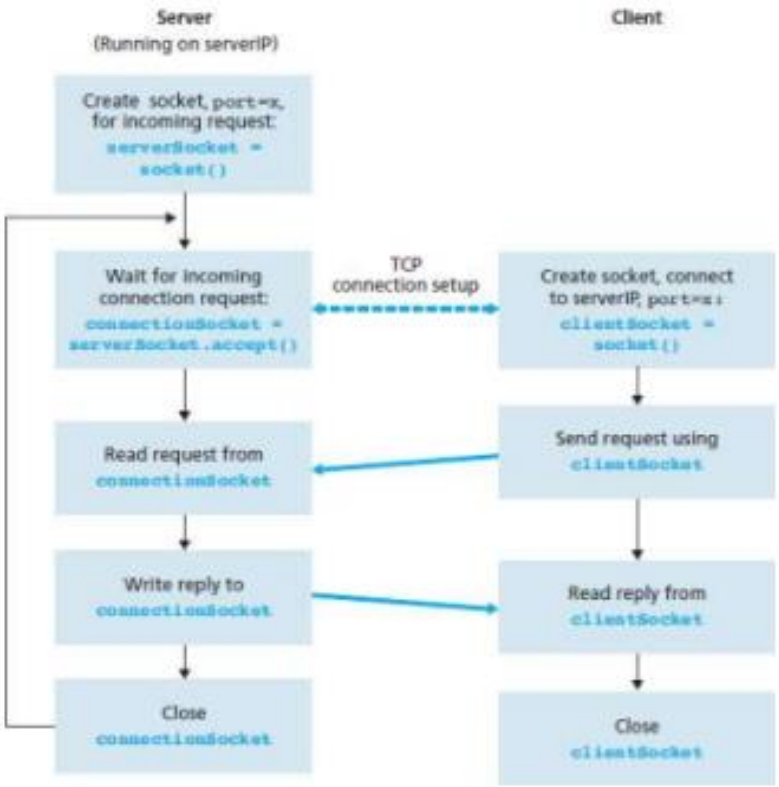
A socket is one endpoint of a two-way communication link between two programs running on the network

TCP Socket Programming:
client.py

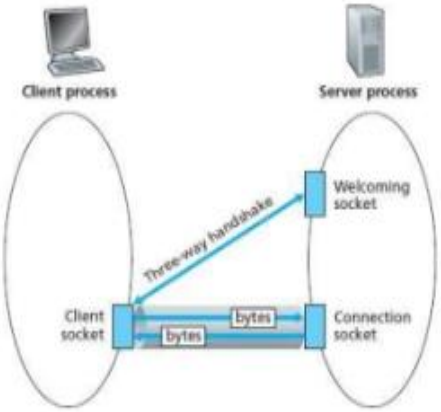
```
from socket import *
serverName = 'servername'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = raw_input('Input lowercase sentence:')
clientSocket.send(sentence)
modifiedSentence = clientSocket.recv(1024)
print 'From Server:', modifiedSentence
clientSocket.close()
```

server.py

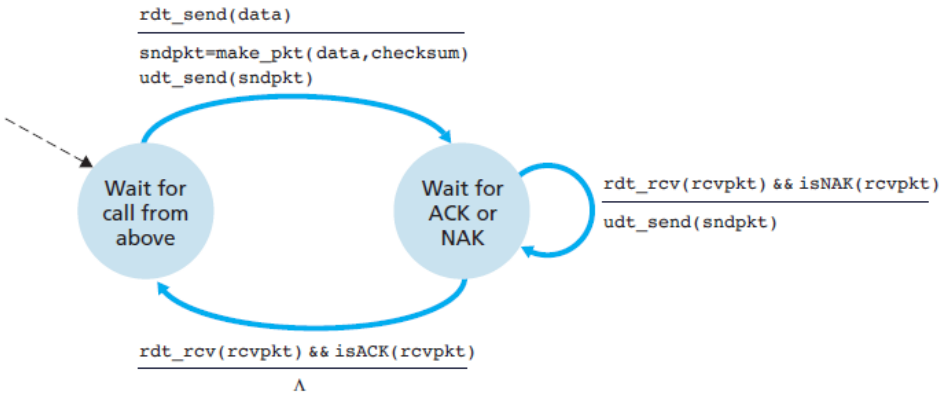
```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((‘’,serverPort))
serverSocket.listen(1)
print ‘The server is ready to receive’
while 1:
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024)
capitalizedSentence = sentence.upper()
connectionSocket.send(capitalizedSentence)
connectionSocket.close()
```



- One end of the TCP connection is attached to the client socket and the other end is attached to a server socket.
- When creating the TCP connection, we associate with it the client socket address (IP address and port number) and the server socket address (IP address and port number). With the TCP connection established, when one side wants to send data to the other side, it just drops the data into the TCP connection via its socket. This is different from UDP, for which the server must attach a destination address to the packet before dropping it into the socket.
- During the three-way handshake, the client process knocks on the welcoming door of the server process. When the server "hears" the knocking, it creates a new door— more precisely, a new socket that is dedicated to that particular client.



3a. With the help of FSM, describe the two states of the sender and one state of the receiver of rdt2.0



a. rdt2.0: sending side

b. rdt2.0: receiving side

3b. With a neat diagram, demonstrate the working of GoBackN Protocol

- The sender is allowed to transmit multiple packets without waiting for an acknowledgment.
- But, the sender is constrained to have at most N unacknowledged packets in the pipeline.
  Where N = window-size which refers maximum no. of unacknowledged packets in the pipeline
- GBN protocol is called a sliding-window protocol.
- Figure 2.17 shows the sender's view of the range of sequence-numbers.

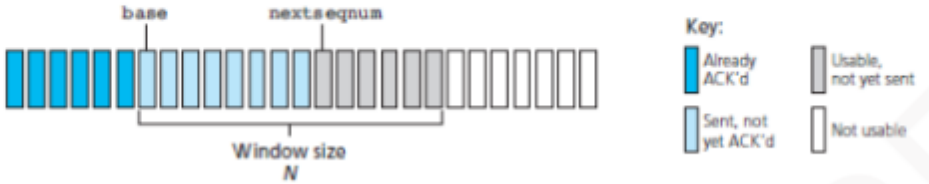


Figure 2.17: Sender's view of sequence-numbers in Go-Back-N

2.4.3.1 GBN Sender

- The sender must respond to 3 types of events:

1) Invocation from above.

➢ When rdt_send() is called from above, the sender first checks to see if the window is full i.e. whether there are N outstanding, unacknowledged packets.

i) If the window is not full, the sender creates and sends a packet.

ii) If the window is full, the sender simply returns the data back to the upper layer. This is an implicit indication that the window is full.

2) Receipt of an ACK.

➢ An acknowledgment for a packet with sequence-number n will be taken to be a cumulative acknowledgment.

➢ All packets with a sequence-number up to n have been correctly received at the receiver.

3) A Timeout Event.

➢ A timer will be used to recover from lost data or acknowledgment packets.

i) If a timeout occurs, the sender resends all packets that have been previously sent but that have not yet been acknowledged.

ii) If an ACK is received but there are still additional transmitted but not yet acknowledged packets, the timer is restarted.

iii) If there are no outstanding unacknowledged packets, the timer is stopped.

2.4.3.2 GBN Receiver

- If a packet with sequence-number n is received correctly and is in order, the receiver
  → sends an ACK for packet n and
  → delivers the packet to the upper layer.
- In all other cases, the receiver
  → discards the packet and
  → resends an ACK for the most recently received in-order packet.

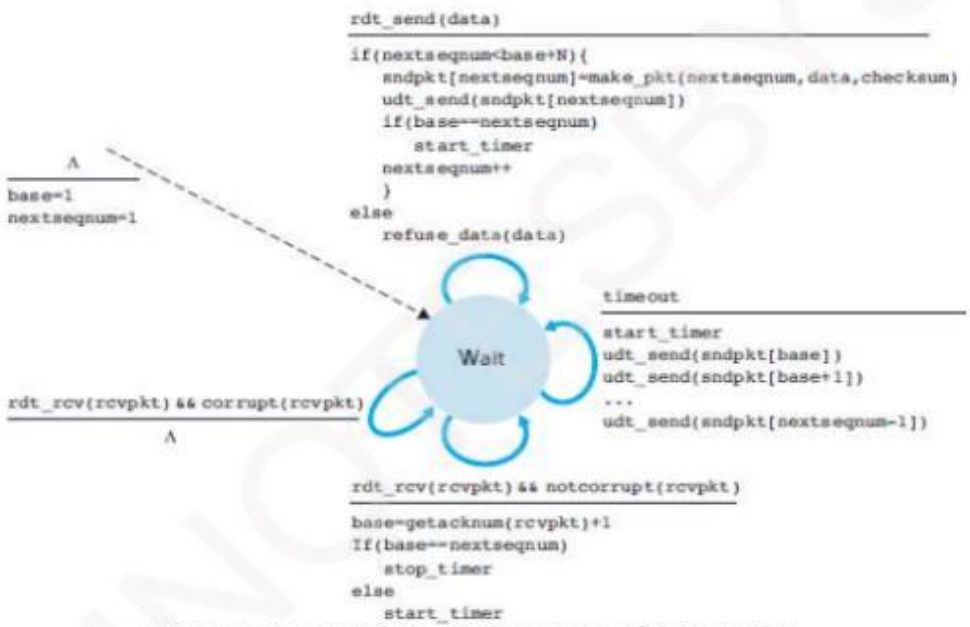


Figure 2.18: Extended FSM description of GBN sender

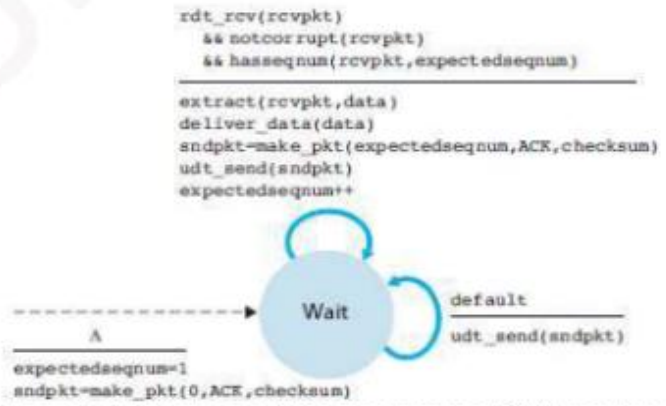


Figure 2.19: Extended FSM description of GBN receiver

4a. Describe TCP Connection Management with the help of diagram

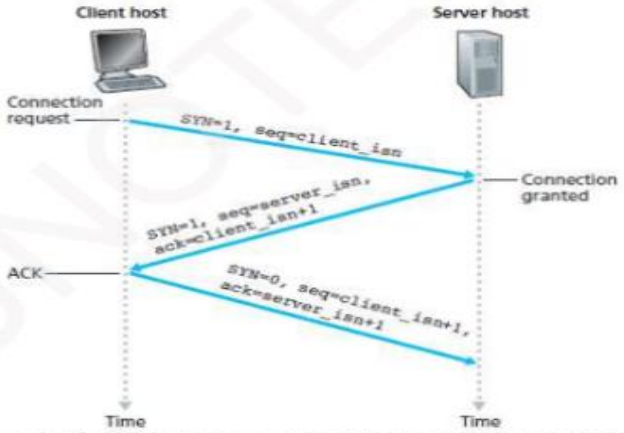


Figure 2.33: TCP three-way handshake: segment exchange

2.5.6 TCP Connection Management

2.5.6.1 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.

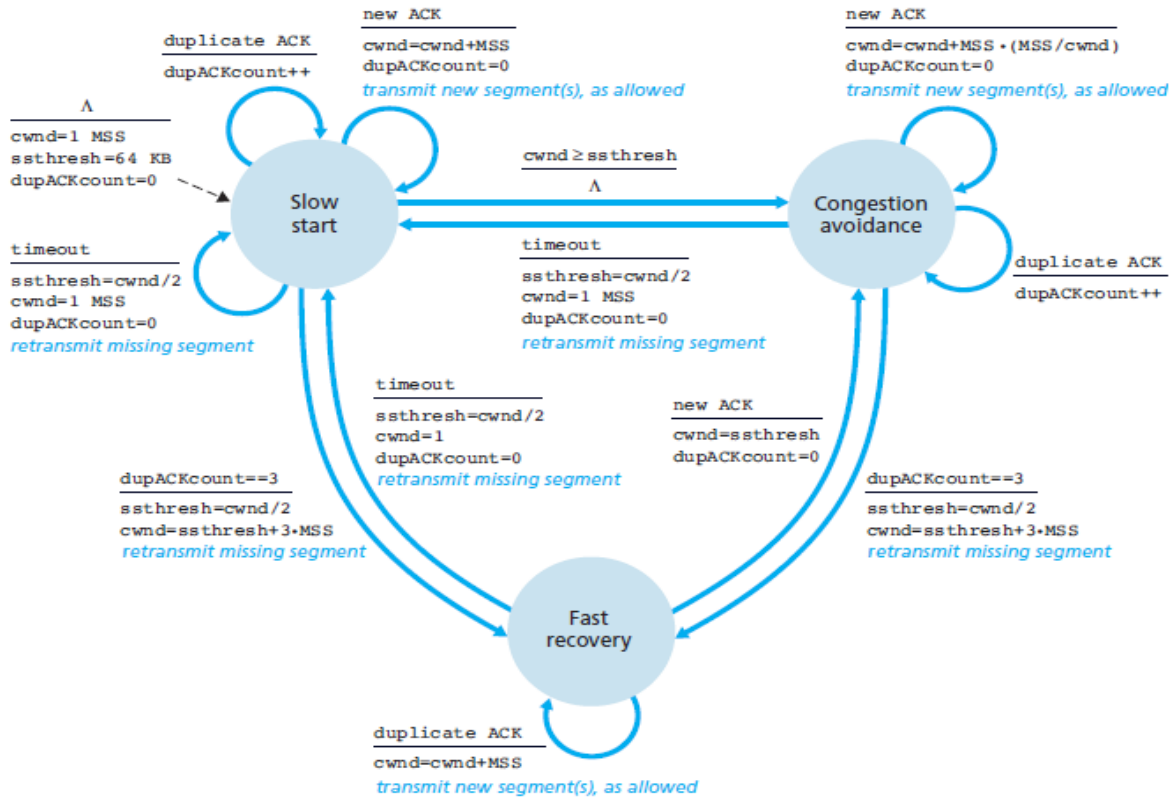
Step 2: Server sends a connection-granted segment to the Client

- 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.

4b. Interpret the FSM for TCP Congestion Control



5a. Explain the implementation of Virtual Circuit Services in Computer 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.

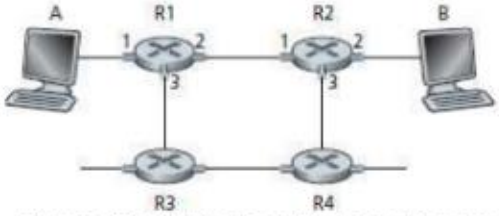


Figure 3.2: A simple virtual-circuit network

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 3.1: Forwarding-table in R1

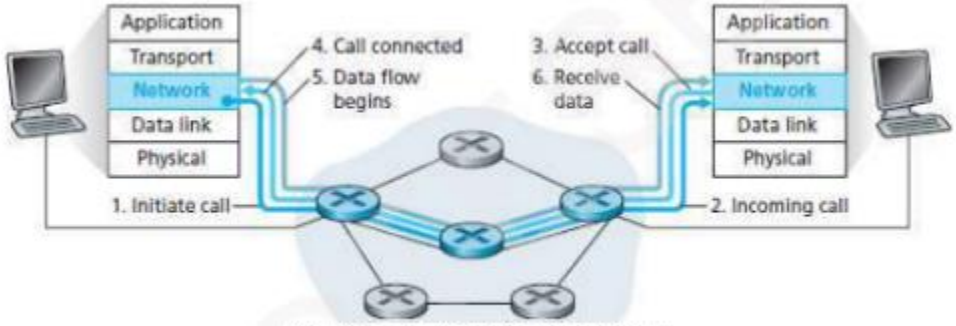


Figure 3.3: Virtual-circuit setup

Three phases in a virtual-circuit (Figure 3.3):

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.

5b. Explain 3 switching techniques

Three types of switching fabrics (Figure 3.7):

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

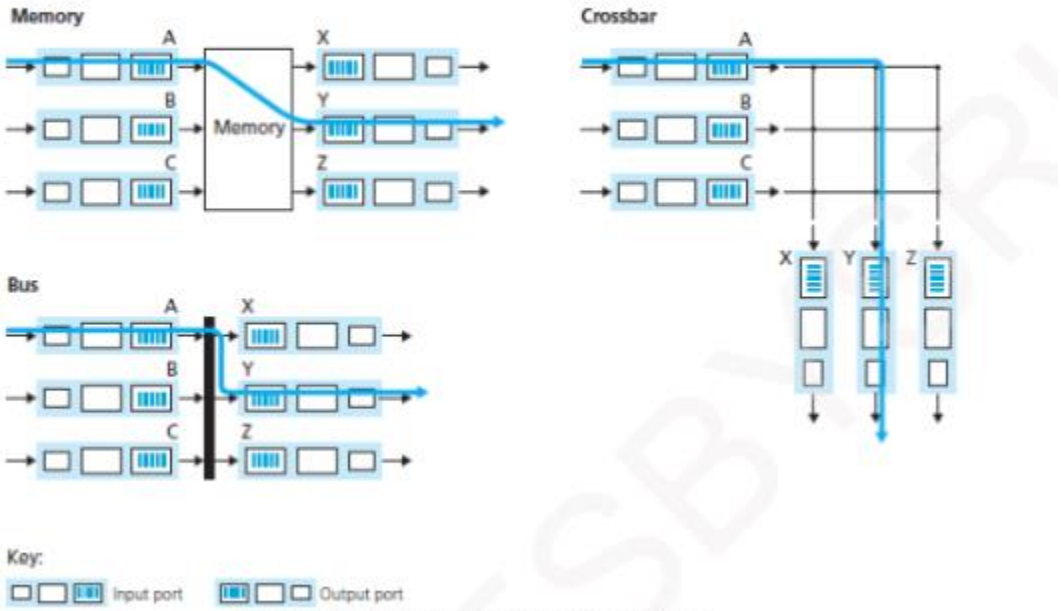


Figure 3.7: Three switching techniques

3.3.1.1 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.

3.3.1.2 Switching via a Bus

- Switching b/w input-ports & output-ports is done without intervention by the routing-processor.
- Here is how it works (Figure 3.7b):
  - 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.

3.3.1.3 Switching via an Interconnection Network

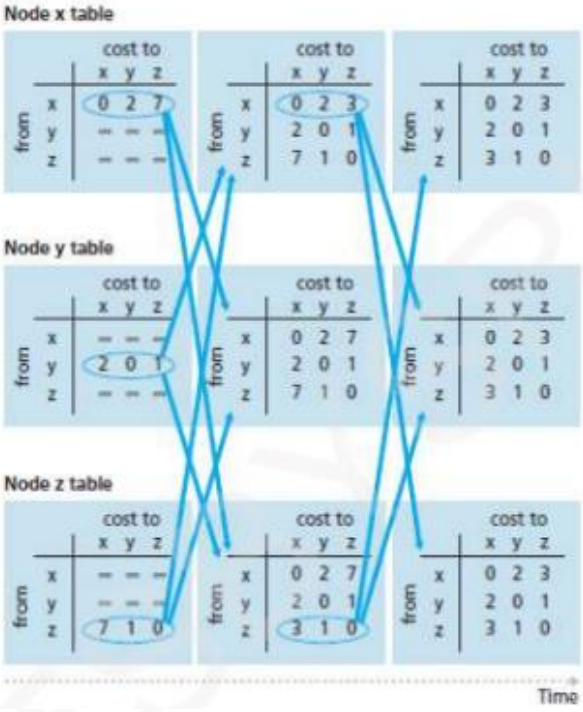
- A crossbar switch is an interconnection network.
- The network consists of 2N buses that connect N input-ports to N output-ports.
- Each vertical bus intersects each horizontal bus at a crosspoint.
- The crosspoint can be opened or closed at any time by the switch-controller.
- Here is how it works (Figure 3.7c):
  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.

5c. Explain DVR Algorithm using 3 nodes network

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 3.24 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.

6a. 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 3.22 and compute the least-cost paths from u to all possible destinations.

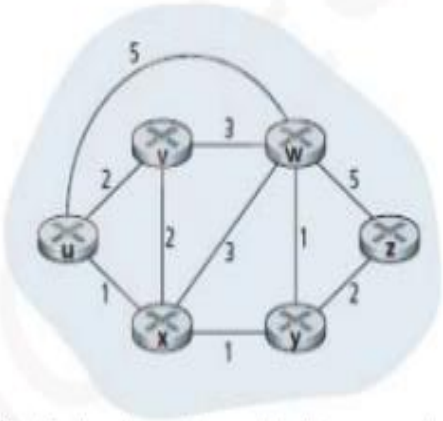


Figure 3.22: Abstract graph model of a computer network

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					

Table 3.5: Running the link-state algorithm on the network in Figure 3.20

- Figure 3.23 shows the resulting least-cost paths for u for the network in Figure 3.22.

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

Figure 3.23: Least cost path and forwarding-table for node u

6b. Explain various broadcast routing algorithms

3.7.1 Broadcast Routing Algorithms

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

3.7.1.1 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.

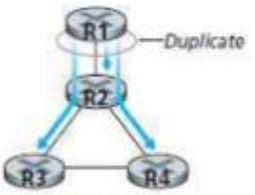


Figure 3.31: Duplicate creation/transmission

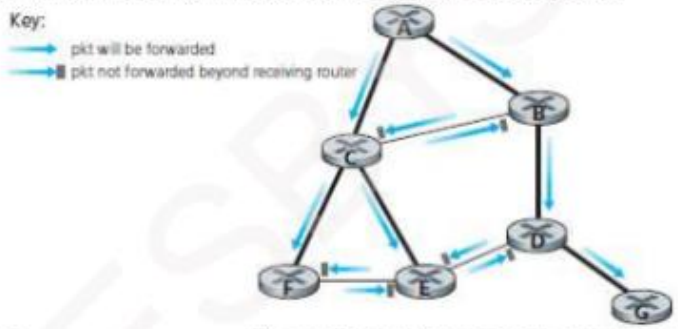


Figure 3.32: Reverse path forwarding

3.7.1.2 Uncontrolled Flooding

- The source-node sends a copy of the packet to all the neighbors.
- When a node receives a broadcast-packet, the node duplicates & forwards packet to all neighbors.
- In connected-graph, a copy of the broadcast-packet is delivered to all nodes in the graph.
- Disadvantages:
  1) If the graph has cycles, then copies of each broadcast-packet will cycle indefinitely.
  2) When a node is connected to 2 other nodes, the node creates & forwards multiple copies of packet
- Broadcast-storm refers to
  "The endless multiplication of broadcast-packets which will eventually make the network useless."

3.7.1.3 Controlled Flooding

- A node can avoid a broadcast-storm by judiciously choosing
  → when to flood a packet and when not to flood a packet.
- Two methods for controlled flooding:

1) Sequence Number Controlled Flooding

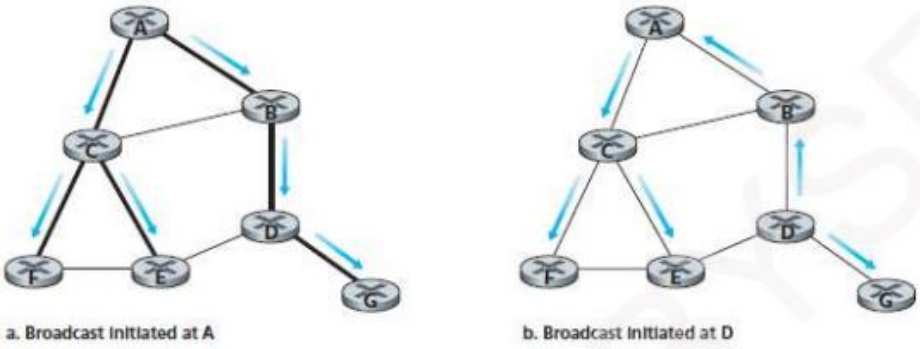
- A source-node
  → puts its address as well as a broadcast sequence-number into a broadcast-packet
  → sends then the packet to all neighbors.
- Each node maintains a list of the source-address & sequence# of each broadcast-packet.
- When a node receives a broadcast-packet, the node checks whether the packet is in this list.
- If so, the packet is dropped; if not, the packet is duplicated and forwarded to all neighbors.

2) Reverse Path Forwarding (RPF)

- If a packet arrived on the link that has a path back to the source;
  Then the router transmits the packet on all outgoing-links.
  Otherwise, the router discards the incoming-packet.
- Such a packet will be dropped. This is because
  → the router has already received a copy of this packet (Figure 3.32).

3.7.1.4 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.



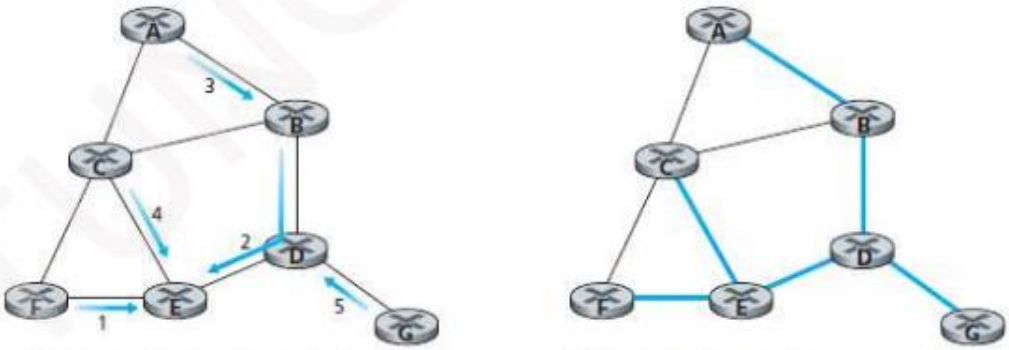
a. Broadcast initiated at A b. Broadcast initiated at D

Figure 3.33: Broadcast along a spanning-tree

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

3.7.1.4.1 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.
- Figure 3.34 illustrates the construction of a center-based spanning-tree.

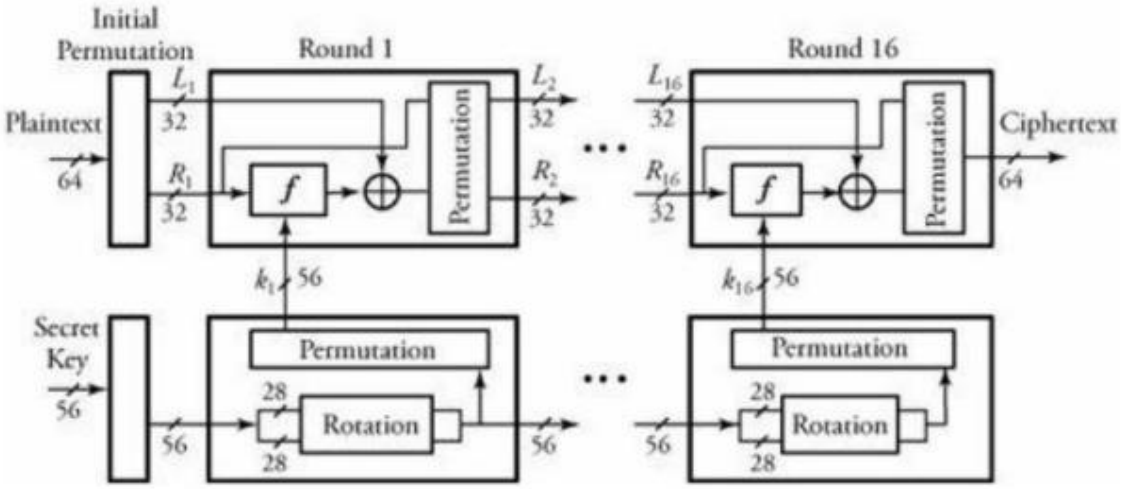


a. Stepwise construction of spanning tree b. Constructed spanning tree

Figure 3.34: Center-based construction of a spanning-tree

7a. Explain Fiestel Structure of DES Algorithm

- Plaintext messages are converted into 64-bit blocks & each block is encrypted using a key.
- The key length is 56 bits.
- DES consists of 16 identical rounds of an operation.



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 roundi.

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

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.

3. 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.

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

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

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

2. 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.

3. 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.

7b. Explain RSA algorithm with example

RSA ALGORITHM

- Assume that a plaintext m must be encrypted to a cipher text c.
- This has three phases: key generation, encryption and decryption.

Key Generation Algorithm

1) Choose two prime numbers a and b and compute n=a.b
2) *Find x*. Select encryption-key x such that x and (a-1)(b-1) are relatively prime.
3) *Find y*. Calculate decryption-key y.

   xy mod (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}

Encryption

1) Both sender and receiver must know the value of n.
2) The sender knows the value of x and only the receiver knows the value of y.
3) Ciphertext c is constructed by

   $c=m^x \bmod n$

Decryption

Given the ciphertext c, the plaintext m is extracted by

$m=c^y \bmod n$.

Example.

F. or 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 mod (a − 1)(b − 1) = 3y mod 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 mod n = 93 mod 33 = 3.

The decryption process is the reverse of this action, as m = c y mod n = 37 mod 33 = 9.

8a. In Diffie-Hellman Key Exchange Protocol, prove that the two keys are equal.

- 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), 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

$$y_1 = g^{x_1} \bmod a.$$

In practice, the two end users agree on a and g ahead of time. User 2 performs the same function and creates y2:

$$y_2 = g^{x_2} \bmod a.$$

User 1 then sends y1 to user 2. Now, user 1 forms its key, k1, using the information its partner sent as

$$k_1 = y_2^{x_1} \bmod a$$

and user 2 forms its key, k2, using the information its partner sent it as

$$k_2 = y_1^{x_2} \bmod a$$

It can easily be proved that the two Keys k1 and k2 are equal. Therefore, the two users can now encrypt their messages

1. Alice and Bob both use public numbers P = 23, G = 5
2. Alice selected private key a = 4, and Bob selected b = 3 as the private key
3. Both Alice and bob now calculate the value of x and y as follows:
   Alice: x = (54 mod 23) = 4
   Bob: y = (53 mod 23) = 10
4. Now, both Alice and Bob exchange public numbers with each other.
5. Alice and Bob now calculate the symmetric keys
   Alice: ka = ya mod p = 104 mod 23 = 18
   Bob: kb = xb mod p = 43 mod 23 = 18
6. 18 is the shared secret key.

8b. Discuss the following

i. 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 2^{24} 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 l_m. The message length is forced to $l_m = 448 \mod 512$. In other words, the length of the padded message becomes 64 bits less than the 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 $\hookleftarrow j$ means left rotation by j bits.
- Then, the 80 steps (i = 0, 1, 2, ..., 79) of the four rounds are described as follows

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

$$R_5 = R_4$$
$$R_4 = R_3$$
$$R_3 = R_2 \hookleftarrow 30$$
$$R_2 = R_1$$
$$R_1 = \delta,$$

Where C_i is a constant value specified by the standard for round i .

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

The message digest is produced by concatenation of the values in R_1 through R_5.

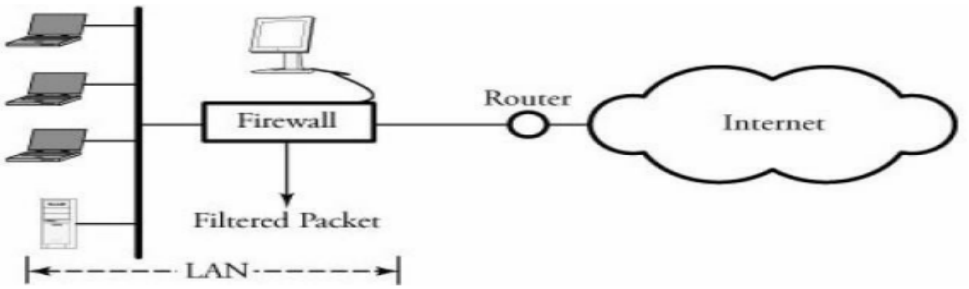
ii. Firewalls

- Firewall is placed between hosts of a certain network and the outside world.
- Firewall is used to protect the network from unwanted web sites and potential hackers.
- The main objective is to monitor and filter packets coming from unknown sources.
- Firewall can also be used to control data traffic.
- Firewall can be a software program or a hardware device.

  1) Software firewalls can be installed in home computers by using an Internet connection with gateways.
- 2) Hardware firewalls are more secure than software firewalls are not expensive.

A firewall controls the flow of traffic by one of the following three methods:

1) Packet filtering: A firewall filters those packets that pass through. If packets can get through the filter, they reach their destinations: otherwise, they are discarded
2) A firewall filters packets based on the source IP address. This filtering is helpful when a host has to be protected from any unwanted external packets.
3) Denial of Service (DOS). This method controls the number of packets entering a network.



9a. Explain briefly how DNS redirects a users request to a CDN server.

- A CDN
  → manages servers in multiple geographically distributed locations
  → stores copies of the videos in its servers, and
  → attempts to direct each user-request to a CDN that provides the best user experience.
- The CDN may be a private CDN or a third-party CDN.

1) Private CDN

- A private CDN is owned by the content provider itself.
- For example:
  Google"s CDN distributes YouTube videos

2) Third Party CDN

- A third-party CDN distributes content on behalf of multiple content providers CDNs.
- Two approaches for server placement:

i) Enter Deep

- The first approach is to enter deep into the access networks of ISPs.
- Server-clusters are deployed in access networks of ISPs all over the world.
- The goal is to get close to end users.
- This improves delay/throughput by decreasing no. of links b/w end user & CDN cluster

ii) Bring Home

- The second approach is to bring the ISPs home.
- Large clusters are built at a smaller number of key locations.
- These clusters are connected using a private high-speed network.
- Typically, clusters are placed at a location that is near the PoPs of many tier-1 ISPs.
  For example: within a few miles of both Airtel and BSNL PoPs in a major city.
- Advantage:
  Lower maintenance and management overhead.
- Disadvantage:
  Higher delay and lower throughput to end users.

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 5.2.

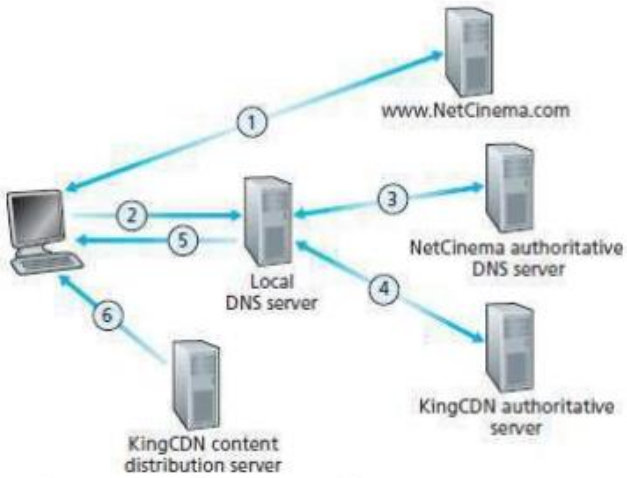


Figure 5.2: DNS redirects a user"s request to a CDN server

- 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.

Cluster Selection Strategies

- Cluster-selection strategy is used for dynamically directing clients to a server-cluster within the CDN.
- The CDN learns the IP address of the client"s LDNS server via the client"s DNS lookup.
- After learning this IP address, the CDN selects an appropriate cluster based on this IP address.
- Three approaches for cluster-selection:

1) Geographically Closest
- The client is assigned to the cluster that is geographically closest.
- Using geo-location databases, each LDNS IP address is mapped to a geographic location.
- When a DNS request is received from LDNS, the CDN chooses geographically closest-cluster.
- Advantage:
  This solution can work reasonably well for a large fraction of the clients.
- Disadvantages: The solution may perform poorly. This is because
  1) Geographically closest-cluster may not be the closest-cluster along the path.
  2) The LDNs location may be far from the client"s location.

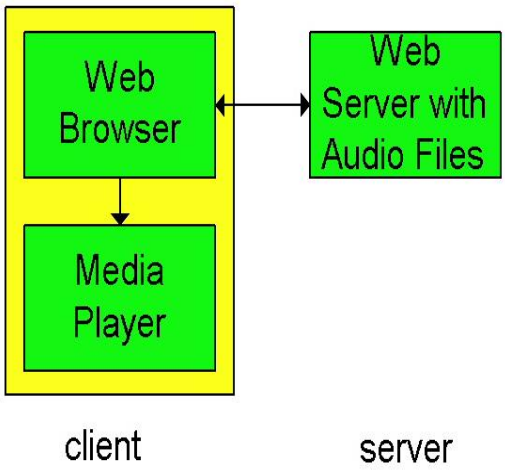
2) Based on Current Traffic Conditions
- The best cluster can be determined for a client based on the current traffic-conditions.
- CDNs perform real-time measurements of delay/loss performance b/w their clusters & clients.
- In a CDN, each cluster periodically sends probes to all of the LDNSs around the world.
- Disadvantage:
  Many LDNSs are configured to not respond to the probes.

3) IP Anycast
- The idea behind IP anycast:
  In Internet, the routers must route the packets to the closest-cluster, as determined by BGP.

9b. With neat diagram explain the naive architecture for audio/video streaming

A naive architecture for audio/video streaming is shown in Figure 6.2.1. In this architecture:

1. The browser process establishes a TCP connection with the Web server and requests the audio/video file with an HTTP request message.
2. The Web server sends to the browser the audio/video file in an HTTP response message.
3. The `content-type:` header line in the HTTP response message indicates a specific audio/video encoding. The client browser examines the content-type of the response message, launches the associated media player, and passes the file to the media player.
4. The media player then renders the audio/video file.



Although this approach is very simple, it has a major drawback: the media player (i.e., the helper application) must interact with the server through the intermediary of a Web browser. This can lead to many problems. In particular, when the browser is an intermediary, the entire object must be downloaded before the browser passes the object to a helper application; the resulting initial delay is typically unacceptable for audio/video clips of moderate length. For this reason, audio/video streaming implementations typically have the server send the audio/video file directly to the media player process. In other words, a direct socket connection is made between the server process and the media player process. As shown in Figure 6.2-2, this is typically done by making use of a **meta file**, which is a file that provides information (e.g., URL, type of encoding, etc.) about the audio/video file that is to be streamed.

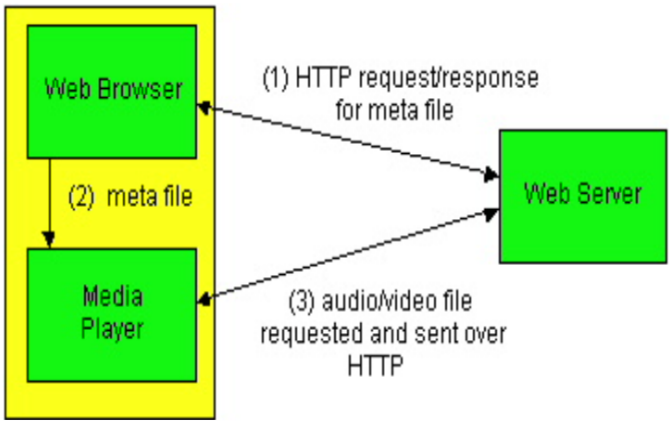


Figure 6.2-2 Web server sends audio/video directly to the media player.

A direct TCP connection between the server and the media player is obtained as follows:

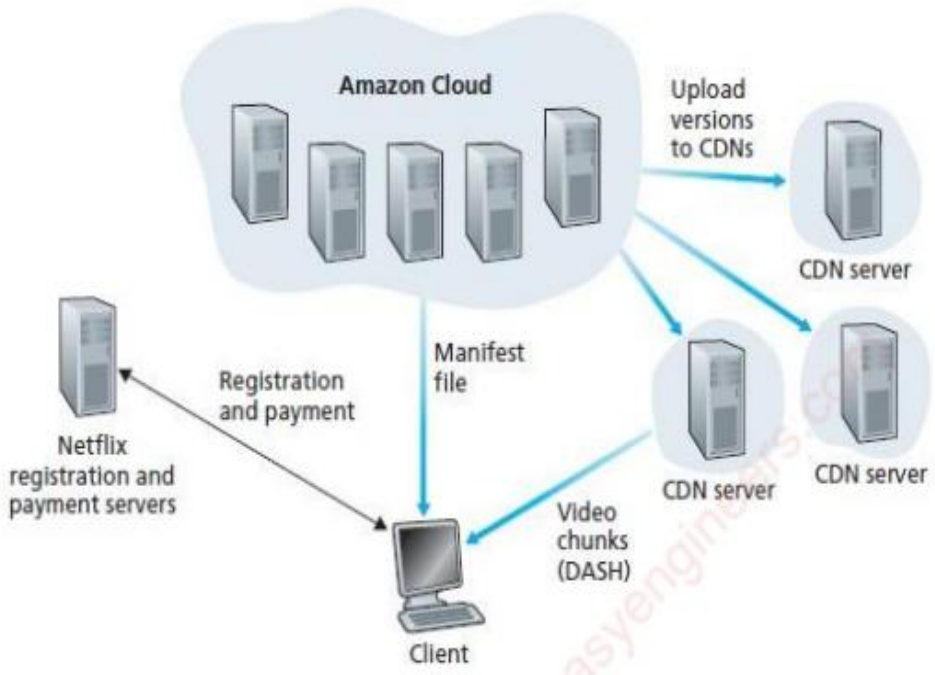
1. The user clicks on a hyperlink for an audio/video file.
2. The hyperlink does not point directly to the audio/video file, but instead to a meta file. The meta file contains the the URL of the actual audio/video file. The HTTP response message that encapsulates the meta file includes a `content-type:` header line that indicates the specific audio/video application.

3. The client browser examines the content-type header line of the response message, launches the associated media player, and passes the entity body of the response message (i.e., the meta file) to the media player.
4. The media player sets up a TCP connection directly with the HTTP server. The media player sends an HTTP request message for the audio/video file into the TCP connection.
5. The audio/video file is sent within an HTTP response message to the media player. The media player streams out the audio/video file.

10a. Write short notes on

i. Netflix video streaming platform

- Basic Architecture:



- Netflix has four major components: the registration and payment servers, the Amazon cloud, multiple CDN providers, and clients.
- In its own hardware infrastructure, Netflix maintains registration and payment servers, which handle registration of new accounts and capture credit-card payment information.
- Netflix runs its online service by employing machines (or virtual machines) in the Amazon cloud. Some of the functions taking place in the Amazon cloud include:
  - **Content ingestion:** Before Netflix can distribute a movie to its customers, it must first ingest and process the movie. Netflix receives studio master versions of movies and uploads them to hosts in the Amazon cloud.
  - **Content processing:** The machines in the Amazon cloud create many different formats for each movie, suitable for a diverse array of client video players running on desktop computers, smartphones, and game consoles connected to televisions. A different version is created for each of these formats and at multiple bit rates, allowing for adaptive streaming over HTTP using DASH.
  - **Uploading versions to the CDNs:** Once all of the versions of a movie have been created, the hosts in the Amazon cloud upload the versions to the CDNs.
- The Web pages for browsing the Netflix video library are served from servers in the Amazon cloud.
- When the user selects a movie to "Play Now," the user's client obtains a manifest file, also from servers in the Amazon cloud. The manifest file includes a variety of information, including a ranked list of CDNs and the URLs for the different versions of the movie, which are used for DASH playback.
- The ranking of the CDNs is determined by Netflix, and may change from one streaming session to the next.
- Typically the client will select the CDN that is ranked highest in the manifest file.
- After the client selects a CDN, the CDN leverages DNS to redirect the client to a specific CDN server.
- The client and that CDN server then interact using DASH.

ii. VOIP with Skype

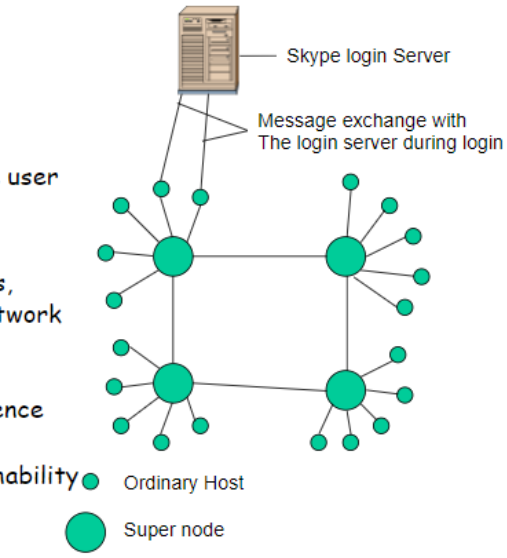
What Is Skype?

- It is a peer-to-peer (P2P) overlay network for VoIP and other applications, developed by the people who did KaZaA
- To a user, it is an Instant Messaging (IM) system that supports P2P VoIP and other applications
- The fee-based SkypeOut service allows calls to regular phone numbers

Overview of Skype P2P Network

Major Entities

- Login Server
  - Single central server
  - User authentication at login (private/public keys)
  - Ensures uniqueness of Skype user names
- Super nodes
  - Nodes with public IP address, enough CPU, memory, and network bandwidth
  - Registrar functionality (distributed directory, presence service)
  - Signaling/media proxy (reachability of ordinary hosts)
- Ordinary nodes
  - Hosts behind FW/NAT



Highlights of Skype

- P2P-based signaling
  - to find and locate users, Skype uses "supernodes" that are running on peer machines (In contrast, traditional VoIP systems use fixed central servers)
  - little management structure and overhead
  - highly scalable
- Strong NAT/firewall traversal capability
  - e.g., use a peer relay to connect clients behind NATs, or use a TCP tunnel to a peer relay to bypass a UDP blocking firewall
- Better voice quality than the MSN and Yahoo IM applications
  - Wideband voice codec: allow frequency between 50-8000Hz to pass through
  - Heavy software-based DSP operations at clients, including codecs, mixer and fancy echo cancellation, e.g., Thinkpad X31, an active Skype session consumes around 10-20% of CPU
- Security feature:
  - encrypts calls end-to-end, and stores user information in a decentralized fashion
- Integrated buddy list, presence information, chat, and audio conferencing

10b. With a neat diagram, explain RTP 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 5.6.

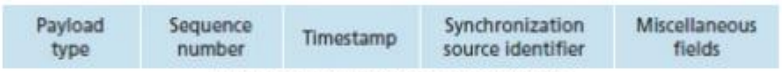


Figure 5.6: RTP header fields

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 5.1 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 5.2 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.