Priyank Lohariwal
00171050155

Internet Technologies
BCSE - Final Year 1st Sem Examinations

GROUP A

1. a) i) Packet Switching is ~~was~~ used ~~to design~~ as a more robust communications network using redundancy and digital technology.

ii) Package switching is used to optimize the use of channel capacity and minimize transmission latency. Hence, it is more bandwidth efficient.

~~Internet~~ Interface Message Processors :-

→ These are minicomputers connected by 50kbps transmission lines.

→ Each IMP is connected with (at least) 2 IMPs

→ A host is connected to an IMP. ~~It then sends a~~ It then sends a message to the IMP that is then split into packets consisting of 1008 bits and is then forwarded ~~to~~ independently to destination.

It is basically the first packet-router used as a part of the original ARPAnet.

---

b) Responsibilities of IETF (Internet Engineering Task force) :-

→ Developing and promoting voluntary Internet standards, particularly the standards that comprise of the Internet protocol suite (TCP/IP).

→ ~~Requesting~~ Issuing Request for comments (RFC) documents that include those RFCs ~~that~~ that define "Internet Standards".

Responsibilities of IESG :- (Internet Engineering Steering Group):

→ Providing the final technical review of Internet Standards.

→ Responsible for day-to-day management of the IETF

GROUP B

~~2. a)~~

2. a) Consider a request/response flow -

~~- The size of the segments being sent ma~~

- The size of the data payload being sent in the flow may not be an exact multiple of the MSS.
- ~~That~~ The last packet of the flow may be smaller.
- This last packet will not be transmitted until the previous packet is acknowledged.

We know, that every second packet is ack'd in delayed acknowledgements.

Two cases arise :-

Case 1 :- ~~Be~~ The second last packet represents an even numbered packet, triggering an immediate acknowledgement from the receiver that causes the sender to release the final packet.

~~Neg~~ Nagle's penalty = 1 RTT (per flow)

Case 2 :- The second last packet represents an odd-numbered packet. Not acknowledged by the receiver until delayed ACK timer expires.

Nagle's penalty = 1RTT + 200 ms (per flow)

(200ms being default Delayed ACK timer)

Hence, the delayed acknowledgement increases Nagle's penalty.

b) Silly Window Syndrome:-

Due to poor implementation of TCP, sometimes the ~~sender's window size~~ receiving window size shrinks to zero (silly value).

Causes:-

- Sender window transmitting one byte of data repeatedly. ~~or~~ and/or
- Receiver window accepting one byte of data repeatedly.

Solution:-

1) Clark's Solution
   a i) Receiver should not send a window update for 1 byte. ie, delay the ack until there is a decent
   ii) amount of buffer space available.
   ii) Receiver should then advertise that window size to the sender.

   OR

   Acknowledge receipt right away but don't change the window size until you have atleast half the buffer space available.

3. a) TCP Congestion control mechanism is implemented at the sender. The send window is adjusted as the minimum of flow control window and congestion window where, flow control window (rwnd) is set by the receiver and congestion window (cwnd) is adjusted based on feedback from the network.

This adjustment of the cwnd is based on the mode at which the algorithm is working.
There are 2 modes:—

- slow start
- congestion avoidance

we set a slow-start threshold value (ssth) until which first mode is used.

Slow start

Initially, cwnd = 1
For every acknowledgement, cwnd is increased by 1.
Hence, the name slow start.
But because of the mechanism, cwnd grows exponentially until cwnd < ssth.

Congestion avoidance

For every ack, cwnd is increased by $1/cwnd$
i.e, $cwnd_{new} = cwnd_{old} + \frac{1}{cwnd_{old}}$

Hence, cwnd grows linearly.

→ TCP assumes there is congestion if it detects a packet loss.

Detection in packet loss

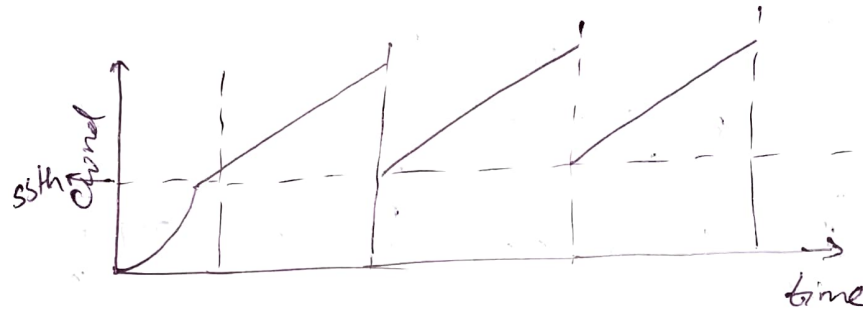
i) Timeout (ssth = cwnd/2)
↳ current cwnd when congestion is detected.

ii) cwnd = 1

iii) ~~Pa~~ Slow start entered.

In this way, congestion control mechanism is entered.

b) 'Fast recovery' is a technique to avoid Slow start after a fast retransmit. The intuition ~~of the~~ is behind this is duplicate ACKs indicate that data is getting through



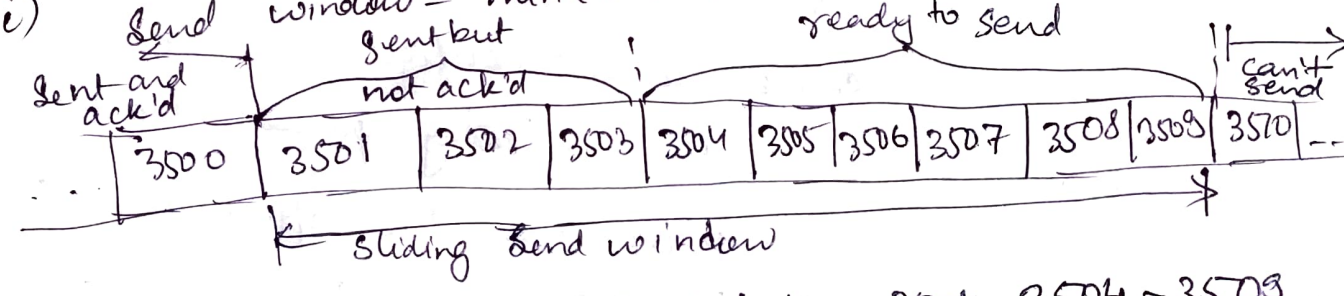
when a congestion is detected, the cwnd is set to ssth instead of 1 to avoid Slow start phase.

This helps in increased utilization of the channel.

---

4. a) rwnd = 9
ACK received = 3501
Sent bytes upto 3503
cwnd = 10

i) Send window = min(rwnd, cwnd) = 9



Hence, the Sender can Send bytes ~~3504~~ 3504 - 3509.

ii) ACK received = 3504

rwnd = 12

cwnd = 10

Bytes Sent = 3507

Assuming ~~only one ack is received~~ ACK 3504 is received just after ACK 3501 and no ACK's are sent in between. Hence, cwnd = cwnd + 1

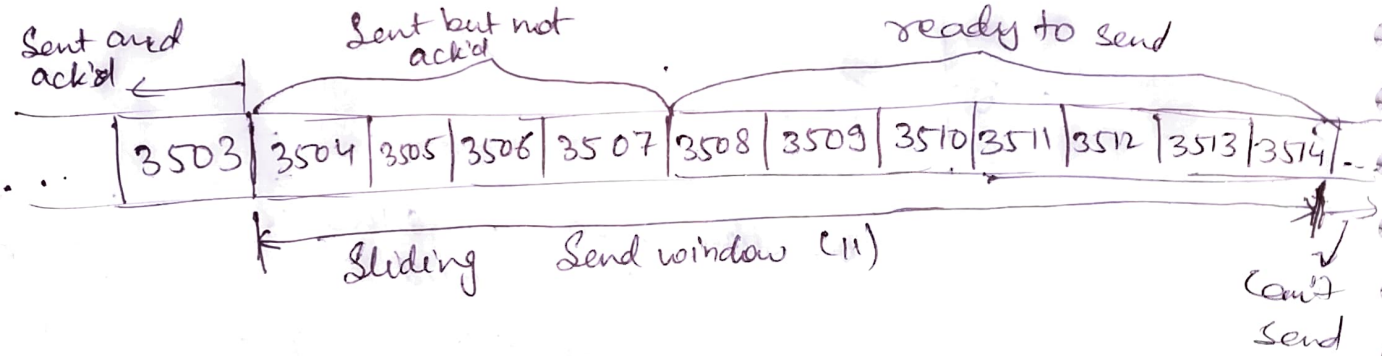
= 10 + 1

= 11

(assuming we are at slow start mode)

∴ Send window = min(rwnd, cwnd)

= min(12, 11) = 11



New value of cwnd = 11

---

5. In auto configuration, the host sends a Router Solicitation message. If a router advertisement message is received, the information is set on the host which includes Prefix and default route

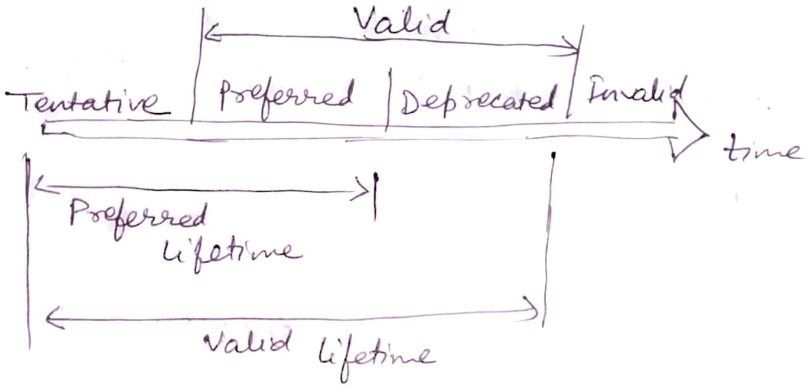
For each stateless auto configuration address prefix that is included, the following processes occur :-

→ The address prefix and the appropriate 64-bit interface identifier are used to derive a tentative address.

→ The uniqueness of the tentative address is verified by using duplicate address detection.

→ Valid and preferred lifetimes are set based on information included in the Router Advertisement message.

States of autoconfiguration



Steps for address autoconfiguration:-

i) Link-local address generation
ii) Uniquess test
iii) Address assignment
iv) Router Contact
v) Router Direction
vi) Global Address configuration

6. The following methods are used to handle Packet loss in multimedia streaming:-

i) Forward error correction ii) Interleaving

i) Forward error correction

→ Redundant encoded chunks are sent after every n chunks. This redundant chunk is found by XORing the first n chunks.

⇒ If a packet in this group is lost, it can be restructured using the redundant chunk.

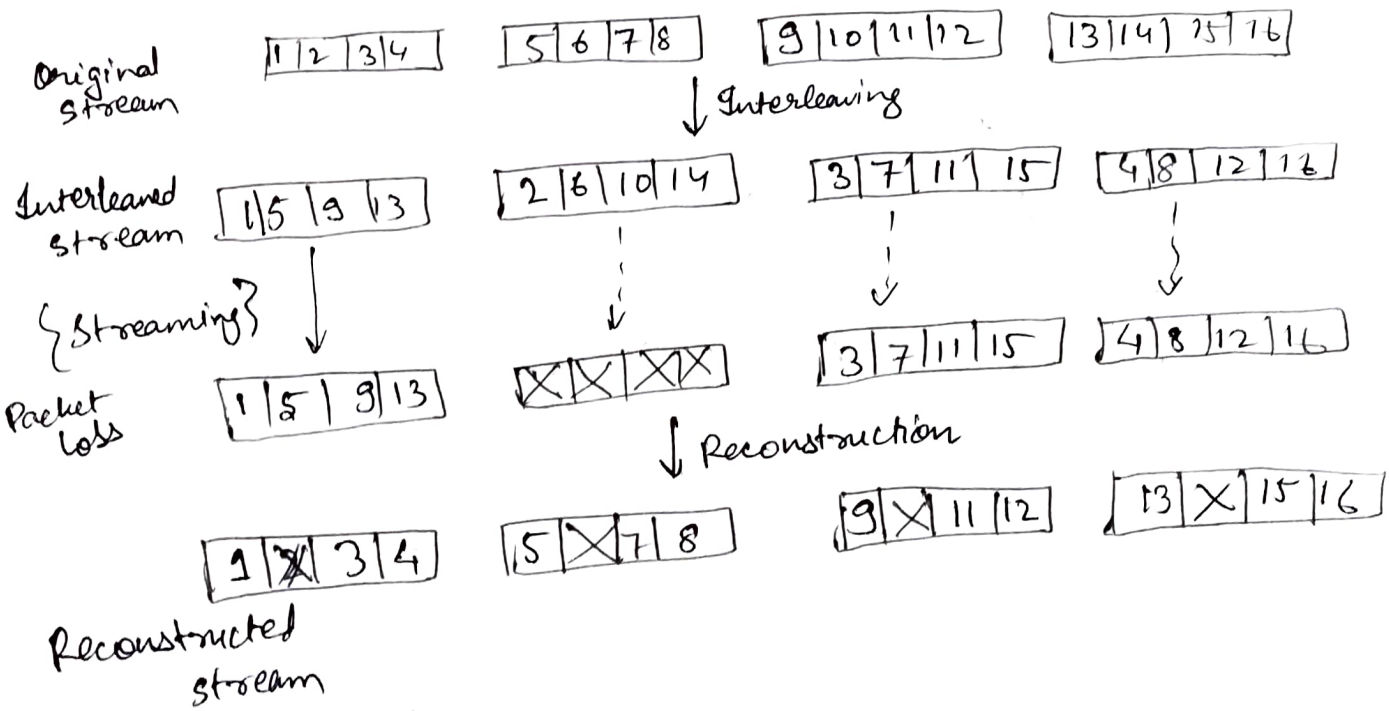
⇒ If more than 1 packet is lost, it is irrecoverable.

~~Disadvantages~~

ii) Interleaving

→ 20 ms of audio data is divided into smaller units of 5 ms each and interleave.

Even if a packet is lost, we would still have a set of partially filled chunks.



We can ~~repair~~ ~~the~~ recover from packet loss by using following methods :-

i) Packet repitition :- Just repeat the last received packet in place of the lost packet.

ii) Interpolation :- Interpolate between the audio before and after the loss to get a suitable packet.

Group C

9. a) A web application is an application software that runs on a web server unlike computer based software programs that are run locally on the OS of the device.

~~The web server is accessed by users via the internet and is built on top of the web that is accessed~~ The web server may be spread across different other servers or throughout the world wide web(www). The world wide web is a system of interlinked documents accessed via the Internet using HTTP. We can say that a webapp is built on www and www is built on top of the Internet.

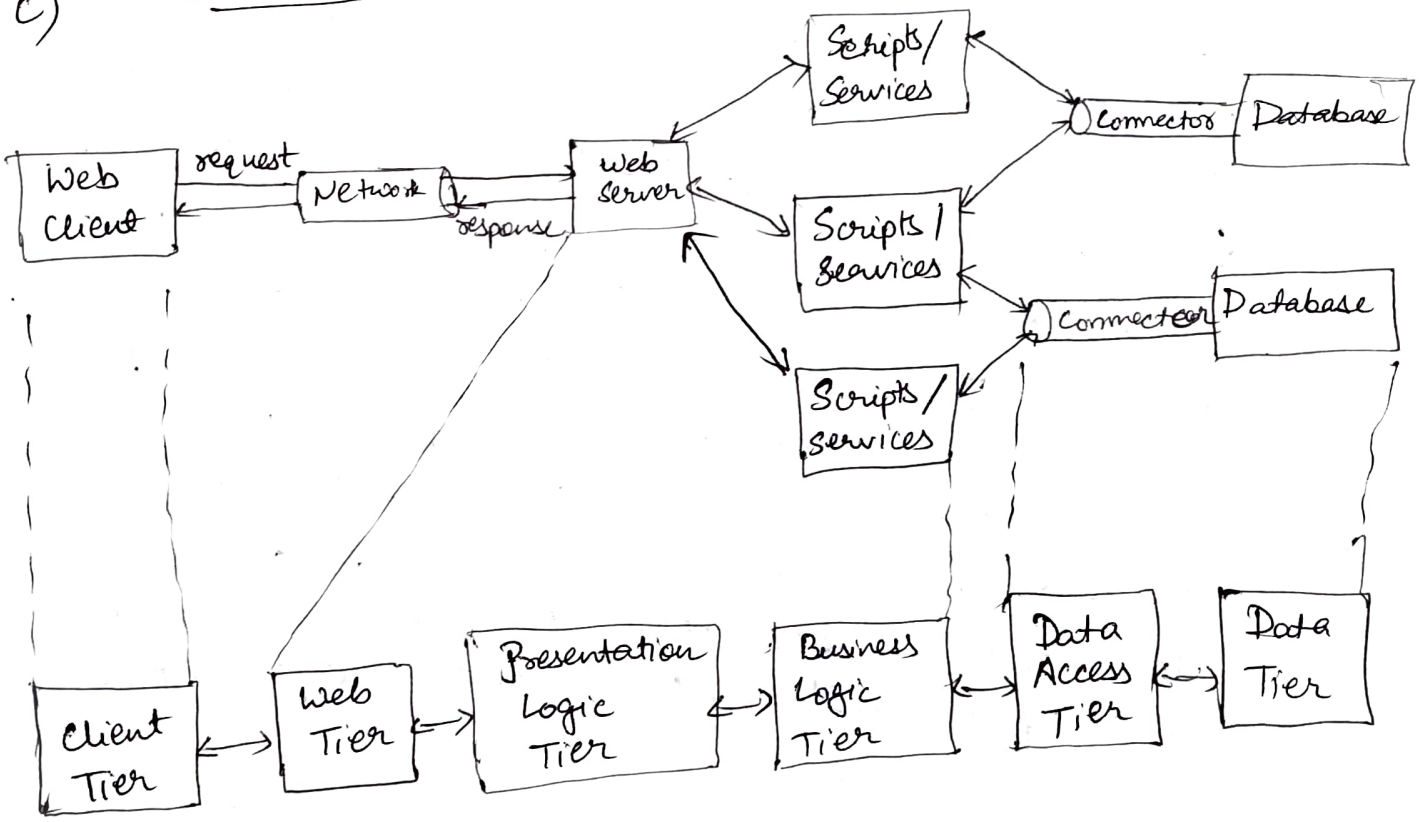
Hence, a web application is related to the Internet in the above mentioned way.

b)

HTTP	WebSockets
i) It is a stateless protocol.	i) It is a stateful protocol.
ii) It is unidirectional in nature and is half-duplex.	ii) It is a bidirectional fully duplex protocol.
iii) HTTP is a client driven protocol. Server cannot initiate a communication.	iii) After a connection is established, both client & server can communicate with each other.

HTTP	Websocket
iv) For every request generated by the client, a new connection is established.	iv) After an initial TCP connection is established, that connection is used for every further communications.
v) moderate overhead per request / connection.	v) Moderate overhead to establish and maintaining the connection, then low overhead per message.

c) 6-Tier Architechture



A famous architectural design pattern is 6-tier architecture.

Concepts:-

- Each tier is responsible to implement one concern of the program i.e, separation of concerns is achieved.
- Every tier communicates with its adjacent tiers only. Hence, as long as a tier knows how to communicate with its adjacent tiers, an entire tier can be changed without crumbling the whole application.

Priyank Lohariwal 00171050105

Group D

10. Model Classes.

Product.java

```
public class Product {
    private int id;
    private String name;
    private String brand;
    private int price;

    Product (int id, String name, String brand, int price) {
        this.id = id;
        this.name = name;
        this.brand = brand;
        this.price = price;
    }
    // Getters and setters
}
```

ProductList.java

~~private ArrayList<Produ~~

```
public class ProductList {
    private ArrayList <Product> products;

    ProductList () {
        // Get the products from the database
    }
    Public ArrayList <Product> Search (
            String name, String brand, int low, int high) {

return products.stream()
        .filter(p -> p.getBrand().equals(brand))
        .filter(p -> p.getName().equals(name))
        .filter(p -> p.getPrice() >= low && p.getPrice() <= high)
```

```
        .collect(Collectors.toList());
    }
    // Getter function for all products
}
```

Priyank Loharwal
00171050105S

Controllers

ProductSearch.java

```
@WebServlet("SearchController", urlPatterns = "/product-search")
public class ProductSearch.java extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req,
                         HttpServletResponse res)
            throws IOException, ServletException {

String name = (String) req.getAttribute("name");
String brand = (String) req.getAttribute("brand");
 int low = (int) req.getAttribute("low");
 int high = (int) req.getAttribute("high");

if (name == null || brand == null || low == null || high == null)
 { req.setAttribute("msg", "Send all details");
   req.getRequestDispatcher("search.jsp").forward(req, res);
 }
    ProductList pl = new ProductList();
ArrayList<Product> products = pl.search(name, brand,
                                        low, high);

if (products.empty()) {
    req.setAttribute("msg", "No items matched your search");
    req.getRequestDispatcher("search.jsp").forward(req,res);
 }
```

res.setAttribute("products", products);
req.getRequestDispatcher("search-results.jsp").forward(req, res);
}

View

Assuming 2 jsp files.

i) search.jsp => User fills the details of the apparels and hits submit button.
This triggers ProductSearch.java.

ii) search-results.jsp => The response from the servlet is shown in this jsp file.

---

b) A web container identifies a returning user by the concept of session management.
Session management can be implemented using Http Sessions.

In general, the server provides a unique token value for a session known as JSESSIONID to the client. This value is stored on the client id.
For subsequent requests, the server uses this token to identify the returning users.

Normally, the server sets a cookie on the client which includes this token. On every subsequent requests, these cookies are sent to the server in the request headers.
If cookies are not supported, the token is encoded in the URL itself at the end following a semicolon(;).

The webserver ensures the uniqueness of session-ids.

c) ~~Objects~~ Attributes are used to share objects among Servlets/JSPs we have the following type of attributes:-

1. Request attribute:-
   These are ~~scp~~ scoped to a particular request-response cycle.
   We use SetAttribute/getAttribute of HttpServletRequest to access.

2. Session attributes:-
   These are scoped to the current user session.
   We use ~~sess~~ setAttribute/getAttribute of HttpSession to access.

3. ContentAttribute:-
   These are scoped to the server's application context.
   We use setAttribute/getAttribute of ServletContext to access.

d) The RequestDispatcher interface provides the facility of dispatching the request to another resource whether it be an html, servlet or jsp. This interface can also be used to include the content of another resource also. It is one of the ways of servlet collaboration.

There are two methods in this interface:-

i) forward(ServletRequest req, ServletResponse ~~Request~~ res)

ii) Include (ServletRequest req, ServletResponse res)

Priyank Lohariwal 00171080105

Group E

12. Filters are used to validate the data sent by the client for servlet based web applications.
More precisely, we use a request filter.
The request from the front-end first goes into the filter classes and after proper validation is forwarded to the actual servlet to be executed. The filter mapping can be declared using either servlet annotations or by writing it in deployment descriptor (web.xml).

Example :- A client sends a username in its request and we need to validate the length of username as >=8 and it starts with a letter.

The filter class code → url mapping same as servlet class

```
@WebFilter("/login")
Public class Validate implements Filter {
@Override
public void init(FilterConfig fconfig) throws
                                    ServletException { }

@Override
public void doFilter(ServletRequest req, ServletResponse res,
                     FilterChain chain)
        throws IOException, ServletException {

String name = req.getParameter("name");
if(name.size() >= 8 &&((name.charAt(0)>='a' &&
   name.charAt(0) <='z')||(name.charAt(0) >='A'&&
   (name.charAt(0) <='Z')))
{ chain.doFilter(req,res); }

else { PrintWriter out = res.getWriter();
       out.println("Wrong username format");
} }
}
```