1.

a) Packet switching involves nodes in a network to store the packets that it receives, determine the best available route and forward the packet to the next node in the route.

It helps to maximize bandwidth efficiency. It results in a more robust connection as packet transfer is not dependent on a single route.

IMPs (Interface Message Processors) are minicomputers connected by 56-Kbps transmission lines.

Each IMP is connected with atleast 2 IMPs.

When a host connects to an IMP and sends a message to to the IMP, it splits the message into packets that are forwarded independently.

b) IETF (Internet Engineering Task force):-

→ develops & promotes voluntary Internet standards, in particular the standards that comprise the Internet protocol suite (TCP/IP).

→ RFC documents including those that describe "Internet Standards" are issued by IETF.

IESG (Internet Engineering Steering Group):-

→ provides the final technical review of Internet Standards.

→ responsible for day to day management of the IETF.

Md. Sahil , 0017105101029

2. a) Consider a request or reply flow -

the size of the payload in the flow may not be an exact multiple of the MSS.

This last packet will not be transmitted until the previous packet is acknowledged.

In the best case, the penultimate package represents an even numbered package, triggering an immediate acknowledgement from the receiver which in turn "releases" the final small packet.

→ In this case, the Nagle penalty is equal to one network round-trip for the entire flow.

If the penultimate package is an odd-numbered packet, it will not be acknowledged by the receiver until the delayed ACK timer expires.

→ In this case the penalty becomes one network round-trip path approximate 200 ms.

b) Silly window syndrome occurs when the buffer in the receiver's side becomes full. When the buffer is full the receiver signals the sender to stop till space is available in the buffer.

But the problem occurs when a small part of the buffer is cleared and the window size is updated.

This results in small packets transmission and the cycle continues.

This leads to poor channel utilization because the overhead ratio increases with small packets.

It can be easily solved using clark's solution. Ack for the packet is sent right away but window size is not updated until a decent amount of window size is available.

3.

a) TCP's mechanism for congestion control is implemented at the sender's side.

The window side at the sender is set as:

Send window = MIN (flow control window, congestion window)

where flow control window is advertised by the receiver & congestion window is adjusted based on feedback from the network.

The congestion control is governed by two parameters :-

→ congestion window (cwnd)

→ slow-start threshhold value (ssthresh).

It works on two modes :-

→ slow start (cwnd < ssthresh)

Each time an ACK is received,

cwnd = cwnd + 1

→ congestion avoidance (cwnd ≥ ssthresh)

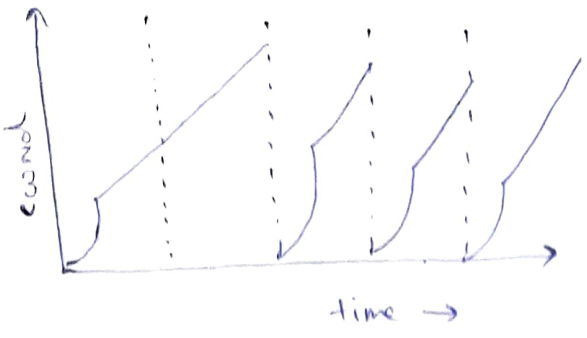
Each time an ACK is received

cwnd = cwnd + 1/cwnd.

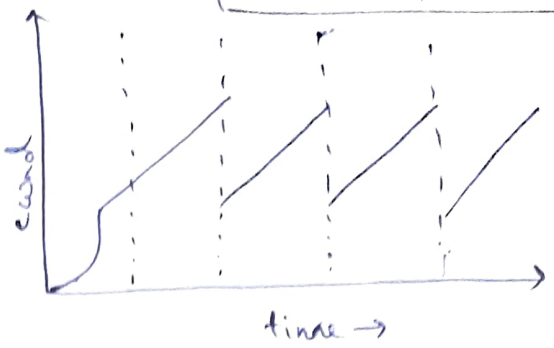
b) Fast recovery avoids slow start after a fast retransmit.

The main intuition behind it is that if duplicate ACKs are received by that sender, it implies that data is getting through.

Avoiding the slow start helps in improving bandwidth utilization of the channel.



without fast recovery



with fast recovery.

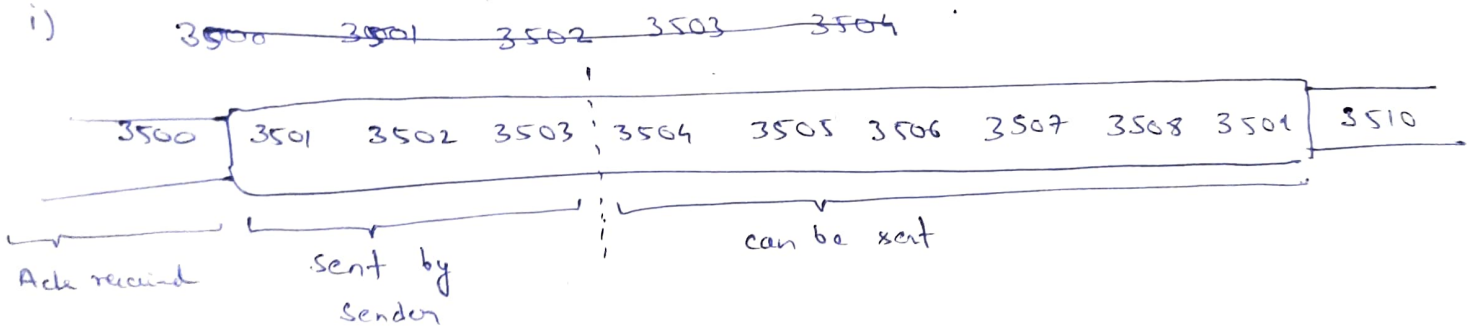
4.

a) Ack sent for 3501.

rwnd = 9 & cwnd = 10

∴ sender window size = min(9, 10) = 9

i)



The sender can send 3504 - 3509 bytes.

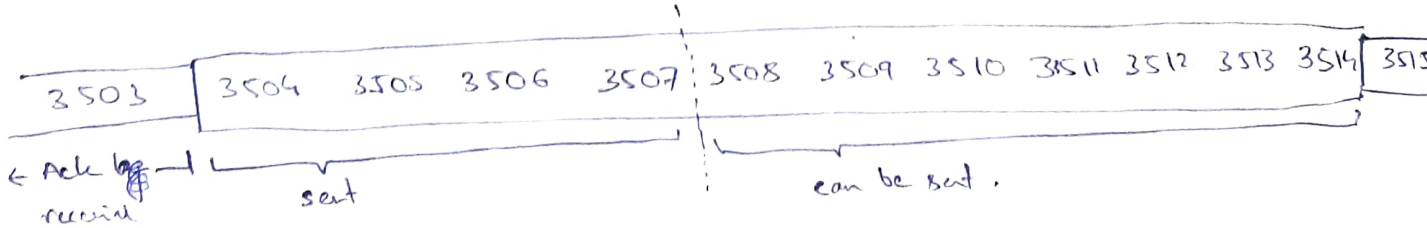
ii) Sender has sent till 3507

Ack recvd for 3504.

rwnd = 12

As single ack received cwnd = 10 + 1 = 11

Sender window = min(12, 11) = 11



& cwnd = 11

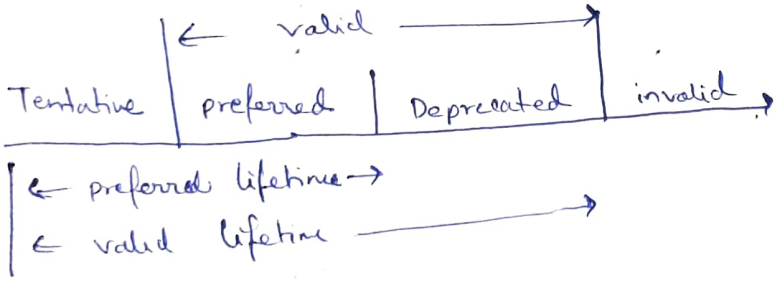
5. In auto configuration the host sends a Router Solicitation message. If a router Advertisement message is received, the information is set on the host which includes ~~prefix~~ prefix.

For each stateless autoconfiguration address prefix 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~~ verified.

→ Valid & preferred lifetimes are set based on information included in the router advertisement message.



Steps for address auto configuration :-

1) Link-local address generation
2) & uniqueness test.
3) address assignment
4) router contact & router direction
5) Global address configuration.

Md. Sahil, 001710501024

6.

The following methods are used to handle loss in media streaming :-

i) Forward error correction :-

→ Redundant encoded chunks are sent after every n normal chunks.

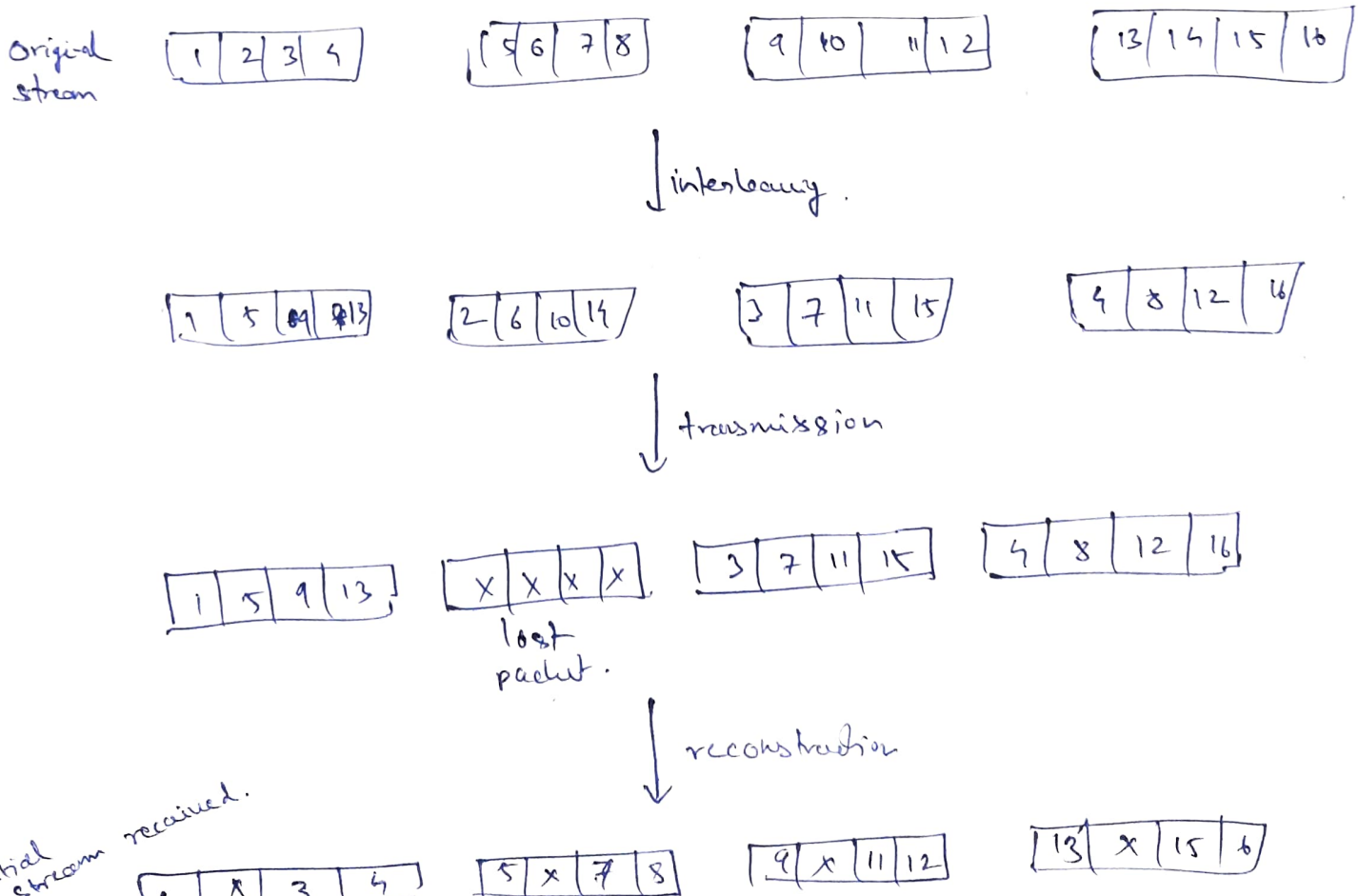
This redundant chunk is found by XORing the first n chunks.

If a packet in this group is lost, it can be reconstructed chunk. If more than 1 packet is lost it is irrecoverable.

ii) Interleaving :-

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

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



The missing data can be reconstructed by either using packet repition & or interpolation.

9.

a) A web application runs on a remote server in contrast to a normal application running in the local machine.

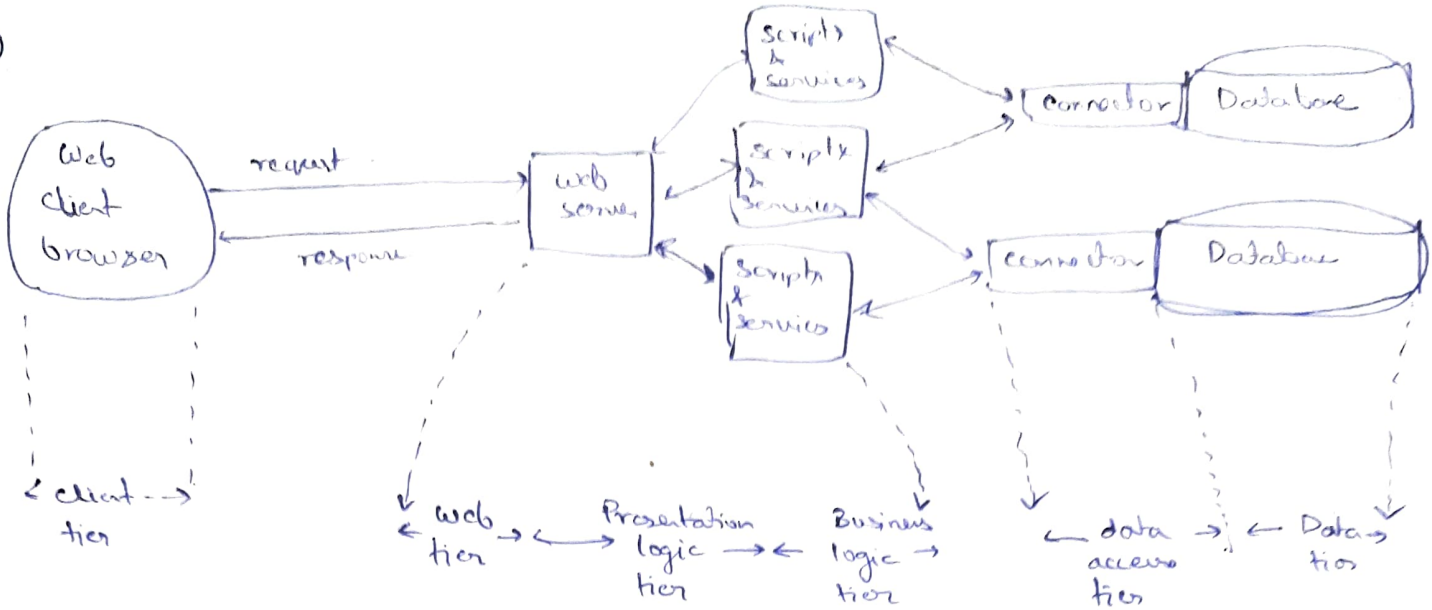
A web application can serve multiple clients and can consist of multiple server side & client side scripts spread across multiple server machines.

The client can avail services by connecting with the server over the world wide web that is built on top of the internet.

b)

HTTP	Websocket
i) stateless	i) statefull
ii) client driven	ii) both server & client & initiate communication
iii) unidirectional in nature	iii) bidirectional in nature.
iv) half duplex	iv) full duplex
v) low overhead per request.	v) moderate overhead per req (to maintain connection)
vi) a new connection is created for every request-response cycle	vii) the same connection persists.

c)



→ Each tier is responsible to implement one concern of the system. i.e. seperation of concern is achieved.

→ Eny tier communicates with its adjacent tiers. So entire tiers can be changed without crumbling the system.

MD. Sakil , 0017105010249

10.

```
class Product {
    private int pid;
    private String name;
    Private float price;
    Private string brand;
    /* getter and setters are available */
}

class Products {
    private static List<Product> products = ..... ;
    private static List<Product> getProducts (
                String name,
                String brand,
                float priceL,
                float priceH ) {
        List<Product> p = new List<Products> (products);
        if (name != null) p.stream().filter(e -> e.getName().equals(name))
        if (brand != null) p.stream().filter

        return products.stream().filter(e -> e.getName().equals(name))
                .filter(e -> e.getBrand().equals(brand))
                .filter(e -> e.getPrice() >= priceL))
                .filter(e -> e.getPrice() <= priceH)).
                .collect(Collectors.toList());
    }
}
```

| Md. Sabit, 01710540024

```
@WebServlet("SearchProductServlet", urlPatterns = "/search.products")
class SearchProductServlet extends HttpServlet {

    @Override
    protected

    public void doGet(HttpServletRequest req, HttpServletResponse res) {

        HttpSession session = req.getSession();
        if (session.isNew())
            req.getRequestDispatcher("searchproduct.jsp").forward(req, res);

        String name = (String) session.getAttribute("name");
        String brand = (String) session.getAttribute("brand");
        Float priceL = (Float) session.getAttribute("priceL");
        Float priceH = (Float) session.getAttribute("priceH");

        if (name == null || brand == null || priceL == null || priceH == null)
        {
            session.setAttribute("msg", "please insert all search params");
            req.getRequestDispatcher("searchproduct.jsp").forward(req, res);
        } else {

            List<Product> products = Products.getProducts(name, brand,
                                                priceL, priceH);

            if (products.empty())
            { session.setAttribute("msg", "Sorry. No items are available");
              session.setAttribute("products", products);

            req.getRequestDispatcher("searchproduct.jsp").forward(req, res);

        }

}
```

b) Users is can be identified using HTTP session via Session management.

In general, the server provides a unique token value called as session-ID to the client each client. On successive requests, this ID can be used to identify returning users.

The server sets a cookie on the client side with the token value. The browser automatically includes the value while making requests to the web server.

The session-ID can also be encoded in the URL itself.

<hostname>/... ; <session ID> ... ...

c) Attributes are used to share objects among servlets / JSPs. We have the following type of attributes :-

1. Request attribute :-

These are scoped to a particular request-response cycle.

We use setAttribute / getAttribute HttpServletRequests to access.

2. Session attributes :-

These are ~~stopped to the~~ scoped to the current user session.

We use setAttribute / getAttribute of HttpSession to access.

3. Context Attribute :-

These are scoped to the ~~server's app~~ server's application context.

We use setAttribute / getAttribute of ServletContext to access.

MD. Sakib, 0017105016, 9

d). The request dispatcher is-

The RequestDispatcher interface provides the facility of dispatching the request to another resource, it may be 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.

Md. Sahil, 0017105010 2a

12. Validation of data can be achieved using filters.

Filters in a servlet-based web application.

A request filter is used to validate the data sent by clients.

The initial request first goes to the filter class and then to the servlet class to be executed. The filter mapping can be declared using annotations or write it in deployment descriptor (web.xml). Url mapping is the same as that of the servlet.

Example :-

A client sends a numerical value in its request. The valid data range corresponds to 1 to 10.

The filter can be written as follows:-

```
@WebFilter("/getmultiple")
Public Class FilterData implements Filter {
    pubic void init(FilterConfig fconfig) throws ServletException { }
    public void doFilter(ServletRequest req, ServletResponse res, FilterChain fchain)
                throws IOException, ServletException {
        int value = Integer.ParseInt(req.getParameter("data"));
        if (val > 1 && val <10)
        {
            chain.doFilter(req, res);
        } else {
PrintWriter out = res.getWriter();
            out.println("Erroraus input");
        }
```