Name - Rakesh Suthar (22BCA10178)
Subject - Computer Networks
Slot - A11 + A12 + A13
Faculty - Dr. Vikas Kumar Jain

Term End Examination

1)

b) In a circuit switched network, end to end addressing is crucial during the setup and teardown phases for several reasons, while it is not needed during the data transfer phase

Need For End to End Addressing

Setup Phase

Established a Dedicated Path - During the setup phase a dedicated communication path (or circuit) must be established between the source and destination devices. This involves configuring switches and reserving bandwidth along the entire route. End to End Addressing are necessary to identify the specific devices involved in the communication and to ensure that the correct path is set up through network switches.

Resource Reservation - The network needs to allocate resources (such as bandwidth) for this connection. The addresses help in reserving these resources along the specified path, ensuring that they are not used by other communications during the call.

Teardown Phase -

Ending the Connection - At the end of a communication session, a teardown message is sent from one endpoint to terminate the established circuit. The end to end addresses are required to identify which specific common connection is being terminated, allowing the network to release the reserved resources properly. This ensures that other potential communications can utilize those resources afterward.

No Addresses Needed During Data Transfer

* Established Circuit - Once the circuit is established all data packets flow through the predetermined path without needing to reference addresses. The connection's dedicated nature means that all devices involved know how to route data along this fixed path without needing additional addressing information.

* Consistent Data Flow - The Data Transfer occurs continously over this dedicated link, allowing for a smooth and reliable exchange of information without interruptions or delays caused by address resolution. The absence of address checks during this phase contributes to lower latency and higher efficiency in data transmission.

Note - End to End Addressing in circuit switched networks is essential for establishing and terminating connections but necessary during actual data transfer due to Dedicated nature of communication path.

2)

b) To illustrate the three phase connection management used in process to process communication between two servers (Server 1 and Server 2), we can refer to TCP three way hand shake mechanism.

Three Phase Connection Management -

1) Connection Establishment

Step 1 SYN (Synchronize) - Server 1 initiates the connection by sending a SYN packet to Server 2. This packet includes an initial sequence number, indicating that Server 1 is ready to establish a connection.

Step 2 SYN-ACK (Synchronize-Acknowledge) - Upon receiving the SYN packet, Server 2 respond with a SYN-ACK packet. This packet acknowledges the receipt of the SYN from Server 1 and includes its own initial sequence number.

Step 3 Ack (Acknowledge) - Finally, Server 1 sends an ACK packet back to server 2, confirming that it received the SYN-ACK. At this point both servers have synchronized their sequence numbers and established a reliable connection.

2) Data Transfer - Once the connection is established data can be transferred between Server 1 and Server 2. The data is segmented into packets which are sent over the established connection. Each packet included sequence numbers for tracking and ensuring that data is received in order.

3) Connection Termination -

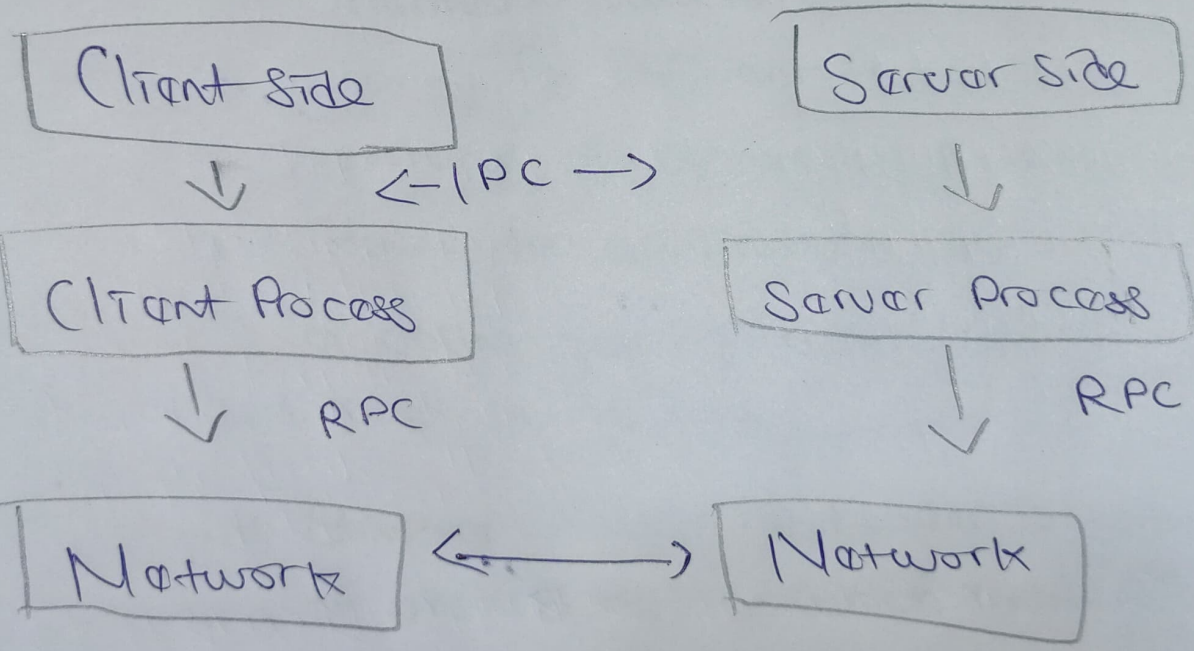
* FIN (Finish) - Either Server can initiate termination by sending a FIN packet to indicate that it has finished sending data.

* ACK For FIN - The receiving server acknowledges the FIN with an ACK

* FIN From Receiver - The receiving server may also send its own FIN packet if it has no more data to send.

* Final Ack - The initiating server acknowledges this final FIN with an ACK, completing the termination process.

Diagram -



3)

a)

UDP-

* Connectionless Protocol - UDP does not establish a connection before data transfer. Each packet or diagram, is sent independently, which makes it faster but less reliable.

* Diagram-Oriented - Data is sent in discrete packets, allowing for efficient transmission without the overhead of connection management. This is suitable for applications like video streaming or online gaming where speed is prioritized over reliability.

* Header Structure - The UDP header is minimal consisting of only 8 bytes, which includes source port, destination port, length, and check sum fields.

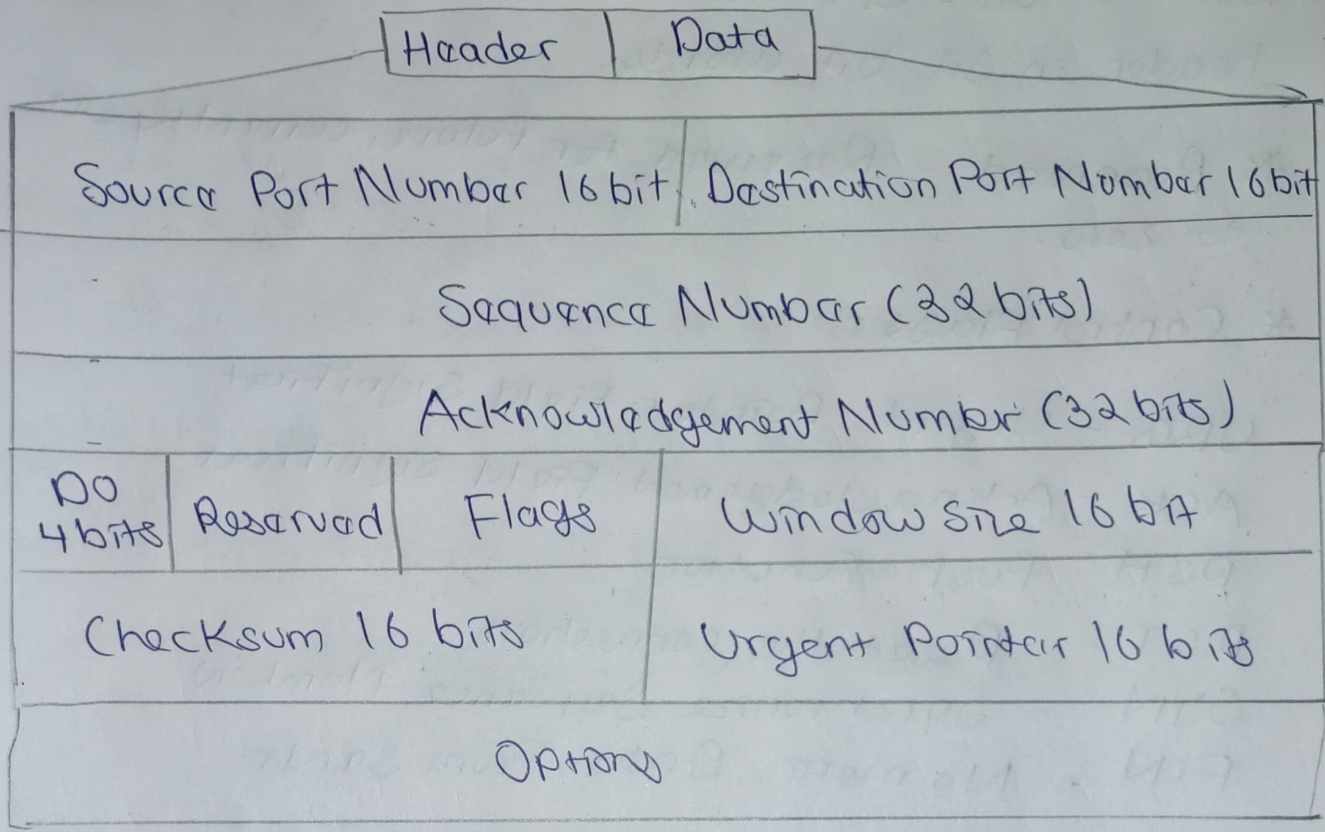
TCP -

* Connection-Oriented Protocol - TCP establishes a connection through a handshake process before data transfer begins. This ensures that both sender and receiver are ready for communication.

* Reliable Transmission - TCP guarantees the delivery of data packets in correct order. It uses acknowledgements and retransmissions to ensure data integrity.

* Header Structure - The TCP header is more complex, ranging 20 to 60 bytes depending on options used. It includes fields for sequence numbers, acknowledgment numbers, flags and more.

TCP Header Format



* Source Port - Identifies the Port Number of Sending Application
* Destination Port - Identifies the Port Number of Receiving Application.
* Sequence Number - Indicates the order of bytes sent from the sender.
* Acknowledgement Number - Contains the next sequence number that the sender expects to receive from the receiver.

* Data offset - Specifies the size of the TCP header in 32 bit words.

* Reserved - Reserved for future, currently set to zero.

* Control Flags -

URG - Urgent Pointer field significant
ACK - Acknowledgement field significat
PSH - Push function
RST - Reset the connection
SYN - Synchronize Sequence Numbers
FIN - No more Data From Sender.

* Window size - Specifies the size of the senders receive window

* Checksum - Used for Error checking the header and data.

* Options - Provides additional feature such as maximum Segment size (MSS)

Ex - A User Request webpage from server

Source Port - 49152
Destination Port - 80
Sequence Number 1
Acknowledgment Number - 0

4)

$$\text{Efficiency}(\eta) = \frac{1}{1+2a}$$

a = Ratio of Propagation Delay

Bit Rate (R) = 4 Kbps = 4000 bps

One way Propagation Delay (T_p) = 20ms = 0.02s

Desired Efficiency (η) $\geq$ 50% or 0.5

$$0.5 = \frac{1}{1+2a}$$

$$1+2a = 2$$
$$\Rightarrow 2a = 1$$
$$a = 0.5$$

Calculating Transmision Delay (T_t)

$$a = \frac{T_p}{T_t}$$

$$0.5 = \frac{0.02}{T_t}$$

$T_t = 0.04$ seconds $= 40$ ms

Minimum Frame Size (L)

$$T_t = \frac{L}{R}$$

$$40\text{ms} = \frac{L}{4000}$$

$L = 4000 \times 0.04 = 160$ bits

5)

When you enter url like www.vitbhopal.ac.in into your web browser, the Domain Name System (DNS) plays a crucial role in translating the human readable address into machine readable IP Address

* Query Initiation - The process begins when you type the domain name into your browser. The browser first checks its local cache to see if it has recently resolved this domain and already knows the corresponding IP Address. If the IP Address is found in cache, it is used immediately, by passing further DNS query.

* DNS Resolver Request - If the IP Address is not cached, the browser sends a DNS query to a DNS Resolver, which is typically provided by your ISP. This resolver acts as an intermediatery between your device and the DNS system.

* Root Name Server - The DNS resolver first queries one of the root name servers. There are 13 root name servers globally, and they provide information about where to find top level domain (TLD) servers (like .com).

* TLD Server - Upon receiving the response from the root server, the resolver then queries the appropriate TLD server for .in in this case. The TLD server contains information about all domains under its extension and directs the resolver to the authoritative name server for vithopal.ac.in

* Authoritative Name Server - The resolver then queries the authoritative name server for vithopal.ac.in. This server holds the actual DNS records that map domain names to their corresponding IP Addresses. If it finds a match it returns the IP Address back to resolver

* Response to Client - Finally the Resolver sends the IP Address back to your web browser - The browser can now establish a connection with the web server at the IP Address using Protocols such as TCP/IP

* Establishing Connection - With the IP Address in hand, your browser establishes a TCP connection with the web server and sends an HTTP request to retrieve the webpage.