DNS Server Purpose & Working

3 classes of DNS server, purpose and working of dns

The three classes of DNS servers are:

Recursive DNS Servers: These servers are responsible for resolving DNS queries on behalf of client machines. When a client requests a DNS resolution, the recursive server will query other DNS servers to find the answer and return it to the client.

Root DNS Servers: These servers are the first point of contact in the DNS resolution process. They maintain information about the top-level domain names (.com, .net, .org, etc.) and provide referrals to other DNS servers for lower-level domains.

Authoritative DNS Servers: These servers are responsible for maintaining DNS records for specific domains. When a recursive server receives a query for a domain, it will query the authoritative DNS server for that domain to get the answer.

The purpose of DNS Is to translate human-readable domain names into IP addresses that computers can understand. When a user types a domain name into their web browser, the browser sends a DNS query to a recursive DNS server. The recursive server then queries other DNS servers until it finds the authoritative DNS server for the requested domain. The authoritative server then returns the IP address associated with the domain, allowing the browser to connect to the appropriate web server and retrieve the desired content.

Overall, DNS is essential to the functioning of the internet, as it allows users to access websites and services by name rather than by IP address, which is easier to remember and more user-friendly.

Purpose of line coding discuss polar nrz schemes

Line coding is a technique used in digital communication to convert a sequence of digital bits into a waveform that can be transmitted over a physical communication channel, such as a copper wire, fiber optic cable, or wireless channel. The purpose of line coding is to ensure reliable and efficient transmission of digital data while minimizing errors and noise.

Polar NRZ (Non-Return-to-Zero) is a line coding scheme that uses two voltage levels to represent binary 0 and binary 1. In polar NRZ, the voltage level stays constant during the transmission of each bit, either high or low, depending on the bit value. This coding scheme is simple to implement and requires no additional synchronization or clock signal, but it has some disadvantages. For example, a long string of consecutive 0's or 1's can cause synchronization problems, making it difficult to distinguish one bit from another.

Polar NRZ-I (Inverted Non-Return-to-Zero) is a variation of polar NRZ that solves this problem by changing the polarity of the signal for each consecutive 1 bit, so that a transition occurs for each bit. This coding scheme eliminates the problem of long strings of consecutive 0's or 1's and provides better synchronization and timing recovery.

The purpose of line coding Is to ensure that digital data can be transmitted over a physical communication channel with minimal errors and noise. Line coding schemes, such as polar NRZ, are used to convert a sequence of digital bits into a waveform that can be transmitted over a communication channel. The choice of line coding scheme depends on various factors, including the nature of the communication channel, the desired data rate, and the characteristics of the data being transmitted.

Merits and demerits of osi model

The OSI (Open Systems Interconnection) model is a conceptual model that describes the communication functions of a telecommunication or computing system. The model consists of seven layers, each of which has a specific set of functions that work together to enable communication between devices.

Here are some of the merits and demerits of the OSI model:

Merits:

Modular design: The OSI model is designed as a modular architecture, meaning that each layer is independent of the others. This makes it easier to develop, implement, and troubleshoot communication protocols.

StandardizationThe OSI model provides a standard framework for communication protocols. This enables interoperability between different systems and devices, making it easier for them to communicate with each other.

Clear division of responsibilities: The model provides a clear division of responsibilities between different layers. This makes it easier to develop protocols and applications that work within a specific layer, without having to worry about the details of lower or higher layers.

Improved troubleshooting: The model's clear division of responsibilities makes it easier to troubleshoot communication problems. Since each layer is independent, problems can be isolated to a specific layer, making it easier to diagnose and fix.

Demerits:

Complexity: The OSI model can be quite complex, with seven different layers and many different protocols to implement. This can make it difficult to design and implement communication systems based on the model.

Overhead: The OSI model requires a lot of overhead in terms of memory and processing power. This can make it difficult to implement on resource-limited devices or systems.

Limited adoption: While the OSI model provides a standard framework for communication protocols, it has not been widely adopted in practice. Many communication protocols used today do not strictly adhere to the OSI model, and some protocols have been developed outside of the model entirely.

Lack of flexibility: The strict division of responsibilities between different layers in the OSI model can limit its flexibility. Some communication systems may require more fluid or dynamic interactions between layers, which may be difficult to achieve within the OSI model's framework.

In conclusion, while the OSI model provides a useful framework for understanding communication protocols, it has both merits and demerits. Its clear division of responsibilities and standardization are valuable, but its complexity, overhead, limited adoption, and lack of flexibility can be drawbacks.

Different approaches to congestion control in TCP.

TCP (Transmission Control Protocol) is a widely used transport layer protocol that provides reliable, ordered, and error-checked delivery of data packets over IP networks. Congestion control is a crucial component of TCP, as it helps to prevent network congestion and ensure efficient and fair use of network resources. There are different approaches to congestion control in TCP, including:

AIMD (Additive Increase Multiplicative Decrease): This is the most used congestion control algorithm in TCP. It works by increasing the sending rate (the congestion window) linearly until congestion is detected, and then reducing the sending rate exponentially. This approach provides a slow and steady increase in sending rate, while also being responsive to network congestion.

Fast Retransmit/Fast Recovery: This approach is used to reduce the number of retransmitted packets when a packet is lost due to congestion. Instead of waiting for a timeout to occur, TCP retransmits the lost packet as soon as it detects that it is missing, based on duplicate ACKs received from the receiver. This approach helps to improve performance and reduce congestion by avoiding unnecessary retransmissions and reducing the time it takes to recover from congestion.

Explicit Congestion Notification (ECN): This approach allows routers to signal TCP endpoints when congestion is detected, using special bits in IP packets. TCP endpoints can then respond by reducing their sending rate before congestion becomes severe, helping to prevent congestion from occurring in the first place.

Random Early Detection (RED): This is a router-based approach to congestion control that involves randomly dropping packets when the queue in a router reaches a certain threshold. This approach encourages TCP endpoints to reduce their sending rate before congestion becomes severe, helping to prevent network congestion from occurring.

Overall, these approaches to congestion control in TCP are designed to ensure efficient and fair use of network resources, while also maintaining reliable and ordered delivery of data packets. By adapting to changes in network conditions and responding to congestion in a timely manner, TCP helps to ensure that network traffic flows smoothly and without interruption.

How is congestion control perceived detected?

Congestion control is an important mechanism used in network protocols to prevent network congestion, which occurs when the amount of traffic on a network exceeds its capacity. Congestion can cause delays, packet loss, and decreased network performance. There are several ways that congestion control is perceived and detected in network protocols:

Packet Loss: Packet loss occurs when network congestion is severe enough to cause packets to be dropped. Packet loss is an indication that the network is congested and can trigger congestion control mechanisms in the protocol.

Delay: Congestion can cause delays in packet delivery, as packets are queued up at routers and switches. Delay can be measured using metrics such as round-trip time (RTT) and can be used as an indication of network congestion.

Queue Length: The length of queues at routers and switches can be used as an indication of network congestion. If the queue length exceeds a certain threshold, congestion control mechanisms can be triggered to reduce the amount of traffic on the network.

Explicit Signalling: Some network protocols, such as TCP, use explicit signalling to detect and respond to congestion. For example, TCP uses ACK packets to signal the sender about the state of the network, including congestion. TCP can then reduce its sending rate to prevent congestion from becoming more severe.

Feedback from the Network: Some network protocols, such as the Resource Reservation Protocol (RSVP), use feedback from the network to detect and respond to congestion. RSVP uses feedback from routers to adjust its traffic control parameters in response to network conditions.

Overall, congestion control mechanisms rely on various indicators to detect and respond to network congestion. By detecting congestion early and responding in a timely manner, these mechanisms help to prevent network congestion from becoming severe and ensure efficient and fair use of network resources.

How does UDP detect errors in transmitted segments explain with example?

UDP (User Datagram Protocol) is a connectionless transport layer protocol that is used to send and receive datagrams over IP networks. Unlike TCP, UDP does not provide reliable, ordered delivery of packets, and does not perform error checking at the transport layer. Instead, error detection and correction are left to the application layer or the receiving device.

UDP does include a checksum field in the header of each datagram, which can be used to detect errors in the transmitted segments. The checksum is computed by adding the values of all the 16-bit words in the datagram, including the source and destination port numbers, length field, and data payload. The checksum value is then complemented and inserted into the checksum field.

When the receiving device receives a UDP datagram, it computes the checksum using the same algorithm and compares it to the value in the checksum field. If the computed checksum matches the checksum in the datagram header, the datagram is assumed to be error-free. If the computed checksum does not match the checksum in the header, the datagram is assumed to be corrupted and is discarded.

For example, suppose a client sends a UDP datagram to a server. The datagram contains a payload of 100 bytes and a checksum value of 0x1234. When the server receives the datagram, it computes the checksum using the same algorithm and obtains a value of 0x5678. Since the computed checksum does not match the checksum value in the datagram header, the server knows that the datagram has been corrupted during transmission and discards it. The client can then resend the datagram if necessary.

It Is important to note that the UDP checksum is not a reliable mechanism for error detection and correction, as it only detects errors with a certain probability and cannot correct errors. Applications that require reliable data transfer and error correction should use TCP instead of UDP.

Difference between circuit switching and packet switching.

Circuit Switching	Packet Switching
In-circuit switching has there are 3 phases: i) Connection Establishment. ii) Data Transfer. iii) Connection Released.	In Packet switching directly data transfer takes place.
In-circuit switching, each data unit knows the entire path address which is provided by the source.	In Packet switching, each data unit just knows the final destination address intermediate path is decided by the routers.
In-Circuit switching, data is processed at the source system only	In Packet switching, data is processed at all intermediate nodes including the source system.
The delay between data units in circuit switching is uniform.	The delay between data units in packet switching is not uniform.
Resource reservation is the feature of circuit switching because the path is fixed for data transmission.	There is no resource reservation because bandwidth is shared among users.
Circuit switching is more reliable.	Packet switching is less reliable.
Wastage of resources is more in Circuit Switching	Less wastage of resources as compared to Circuit Switching
It is not a store and forward technique.	It is a store and forward technique.
Transmission of the data is done by the source.	Transmission of the data is done not only by the source but also by the intermediate routers.
Congestion can occur during the connection establishment phase because there might be	Congestion can occur during the data transfer phase, a large

Circuit Switching	Packet Switching
a case where a request is being made for a channel but the channel is already occupied.	number of packets comes in no time.
Circuit switching is not convenient for handling bilateral traffic.	Packet switching is suitable for handling bilateral traffic.
In-Circuit switching, the charge depends on time and distance, not on traffic in the network.	In Packet switching, the charge is based on the number of bytes and connection time.
Recording of packets is never possible in circuit switching.	Recording of packets is possible in packet switching.
In-Circuit Switching there is a physical path between the source and the destination	In Packet Switching there is no physical path between the source and the destination
Circuit Switching does not support store and forward transmission	Packet Switching supports store and forward transmission
Call setup is required in circuit switching.	No call setup is required in packet switching.
In-circuit switching each packet follows the same route.	In packet switching packets can follow any route.
The circuit switching network is implemented at the physical layer.	Packet switching is implemented at the datalink layer and network layer
Circuit switching requires simple protocols for delivery.	Packet switching requires complex protocols for delivery.

Difference between persistent and non persistent http

Persistent Connection	Non-Persistent Connection
The Persistent Connection is the second version of the HTTP, and it is also called as HTTP/1.1	The Non-Persistent connection was the first version of HTTP, and it is also called as HTTP/1.0
The Persistent connection will always be in the default mode.	The Non-Persistent connection will always be in the non-default mode.
The Persistent connection uses very less time because all the requests and responses are transferred in a single TCP.	The Non-Persistent connection uses more time when compared to Persistent connection because it uses new TCP for every new request and response.
The Persistent connection requires only one round trip time for all the objects.	The Non-Persistent connection requires two RTT's for every object present in the connection.
The request methods used in the Persistent connection are GET, HEAD, POST, PUT, DELETE, etc.	The request methods used in the non-Persistent connection are HEAD, POST, etc.
For downloading the multiple objects, the Persistent connection only uses a single connection	For downloading the multiple objects, the non-Persistent connection requires multiple connections
The usage of the CPU will be less in the persistent connection because it runs on a single TCP	The usage of the CPU will be more when compared to persistent connection because it runs on the multiple TCP's

Flow of UDP sender and receiver actions

The User Datagram Protocol (UDP) is a simple and connectionless protocol for sending and receiving data over an IP network. The flow of actions between the UDP sender and receiver is as follows:

UDP Sender Actions:

The sender creates a UDP socket and binds it to a local port number.

The sender specifies the IP address and port number of the receiver.

The sender creates a UDP datagram containing the data to be sent and includes the destination IP address and port number in the header.

The sender sends the datagram to the network using the sendto() function.

If an error occurs during the transmission, the sender may retry sending the datagram.

UDP Receiver Actions:

The receiver creates a UDP socket and binds it to a local port number.

The receiver waits for incoming UDP datagrams using the recvfrom() function.

When a datagram arrives, the receiver extracts the data from the datagram and processes it.

The receiver can send a response datagram to the sender, using the sender's IP address and port number from the incoming datagram.

The receiver can continue to wait for incoming datagrams or close the socket.

In summary, the UDP sender creates a UDP datagram, specifies the destination address and port number, and sends the datagram to the network. The UDP receiver creates a UDP socket, waits for incoming datagrams, processes the data, and may send a response datagram to the sender. UDP does not establish a connection between the sender and receiver, so there is no need for a handshake or other connection-oriented protocol actions.

How DHCP assigns IP addresses dynamically to hosts? Explain DHCP client server scenario.

The Dynamic Host Configuration Protocol (DHCP) is a protocol that is used to automatically assign IP addresses and other network configuration parameters to hosts on a network. DHCP works using a client-server model, where a DHCP server manages a pool of IP addresses and other configuration parameters, and leases them to DHCP clients for a specific period.

Here Is the general scenario of DHCP client-server interaction:

The DHCP client sends a broadcast message to the network requesting an IP address and other network configuration parameters.

The DHCP server receives the broadcast message and responds with an offer of an IP address and other configuration parameters.

The DHCP client receives the offer and sends a request for the offered IP address and configuration parameters.

The DHCP server acknowledges the request and assigns the IP address and configuration parameters to the DHCP client.

The DHCP client sends a final message to indicate that it has accepted the assigned IP address and configuration parameters.

The DHCP client-server interaction can be divided into the following steps:

DHCP Discover: The DHCP client sends a broadcast message to the network, requesting an IP address and other network configuration parameters. The message contains the client's MAC address and other identifying information.

DHCP Offer: The DHCP server receives the broadcast message and responds with an offer of an IP address and other configuration parameters. The offer message contains the IP address and other configuration parameters, along with the server's IP address and lease duration.

DHCP Request: The DHCP client receives the offer message and sends a request for the offered IP address and configuration parameters. The request message contains the client's MAC address and the server's IP address.

DHCP Acknowledge: The DHCP server receives the request message and acknowledges it by assigning the IP address and configuration parameters to the DHCP client. The acknowledge message contains the assigned IP address and lease duration.

DHCP Release: When the DHCP lease expires, the DHCP client releases the assigned IP address and other configuration parameters. This allows the DHCP server to reassign the IP address to another client.

In summary, DHCP allows hosts to dynamically obtain IP addresses and other network configuration parameters from a DHCP server. The DHCP client sends a broadcast message to request an IP address, and the DHCP server responds with an offer of an IP address and other configuration parameters. The DHCP client then sends a request to accept the offer, and the DHCP server assigns the IP address and configuration parameters to the client. Finally, the client sends a message to acknowledge the assignment of the IP address and configuration parameters.

PDU used by any four layers of OSI model.

The OSI model has seven layers, and each layer uses a Protocol Data Unit (PDU) to encapsulate the data being transmitted. Here are the PDUs used by any four layers of the OSI model:

Application layer: The PDU at the application layer is the data itself. The application layer does not add any headers or trailers to the data. Examples of protocols at this layer include HTTP, FTP, SMTP, and Telnet.

Transport layer: The PDU at the transport layer is a segment. The transport layer adds a header that contains information such as the source and destination port numbers, sequence numbers, and checksum. Examples of protocols at this layer include TCP and UDP.

Network layer: The PDU at the network layer is a packet. The network layer adds a header that contains information such as the source and destination IP addresses, and a time-to-live (TTL) value. Examples of protocols at this layer include IP, ICMP, and ARP.

Data Link layer: The PDU at the data link layer is a frame. The data link layer adds a header that contains information such as the source and destination MAC addresses, and a trailer that contains a checksum. Examples of protocols at this layer include Ethernet, Wi-Fi, and PPP.

Physical layer: At the physical layer, the PDU is typically referred to as a bit. The physical layer is responsible for the transmission and reception of raw bit streams over a physical medium, such as copper wires, fiber-optic cables, or wireless signals. It deals with the electrical, mechanical, and procedural aspects of physical data transmission.

It's worth noting that the PDUs used by each layer of the OSI model are encapsulated within each other as the data passes down the layers, with the lower layers adding their own headers and trailers to the data.

How is crosstalk minimised in case of twisted pair cables

Crosstalk is the unwanted transfer of signals between adjacent pairs of twisted pair cables. This can cause interference and reduce the quality of the transmitted signals. There are several techniques used to minimize crosstalk in twisted pair cables:

Twisted Pair Design: The twisting of the pairs of wires is done in a way to minimize the crosstalk between the pairs. The twists in the wire pairs change their position along the length of the cable, which helps to reduce the electromagnetic interference (EMI) and the effect of crosstalk.

Shielding: Shielding is an effective way to reduce crosstalk. Shielding involves placing a layer of conductive material around the twisted pair cable, which reduces the amount of external electromagnetic interference that can affect the cable. Shielding can be done using braided or foil shields.

Separation: Separating the twisted pair cables from other cables can also help to minimize crosstalk. This is done by keeping the twisted pair cables away from other cables, especially those that carry high-frequency signals.

Proper termination: Proper termination of the cables can also help to reduce crosstalk. A poorly terminated cable can increase the amount of crosstalk between the wires.

Crosstalk cancellation techniques: Crosstalk can also be minimized by using techniques that cancel out the effect of crosstalk. These techniques include adaptive equalization and echo cancellation.

Overall, proper design, shielding, separation, termination, and the use of crosstalk cancellation techniques can all help to minimize crosstalk in twisted pair cables.

Minimum value of retransmission timer in ARQ protocol

> Given values of `EstimatedRTT` and `DevRTT`, what value should be used for TCP's timeout interval? Clearly, the interval should be greater than or equal to `EstimatedRTT`, or unnecessary retransmissions would be sent. But the timeout interval should not be too much larger than `EstimatedRTT`; otherwise, when a segment is lost, TCP would not quickly retransmit the segment, leading to large data transfer delays. It is therefore desirable to set the timeout equal to the `EstimatedRTT` plus some margin. The margin should be large when there is a lot of fluctuation in the `SampleRTT` values; it should be small when there is little fluctuation. The value of `DevRTT` should thus come into play here. All of these considerations are taken into account in TCP's method for determining the retransmission timeout interval:
>
> `TimeoutInterval = EstimatedRTT + 4 · DevRTT`

Why is delta modulation used in digital communication? How modulator and demodulator work to perform delta modulation

Delta modulation is used in digital communication because it is a simple and efficient technique for encoding analog signals into digital signals. It is a form of analogue-to-digital conversion that samples the analogue signal at a high rate and encodes each sample as a 1-bit digital value. Delta modulation works by comparing the current sample to the previous sample and encoding the difference between the two as a 1-bit value.

The modulator in a delta modulation system takes an analog input signal and samples it at a high rate, typically several times the Nyquist rate. The modulator then compares each sample to the previous sample and encodes the difference between the two as a 1-bit value. The output of the modulator is a stream of 1-bit values that represent the encoded digital signal.

The demodulator in a delta modulation system receives the stream of 1-bit values and reconstructs the original analog signal. The demodulator starts with an initial estimate of the analog signal, and then adds or subtracts the encoded difference value for each sample to update the estimate. The demodulator output is the reconstructed analog signal.

Delta modulation is a simple and efficient technique for encoding analog signals into digital signals, but it has some limitations. One of the limitations is that it can introduce quantization noise, which is caused by the 1-bit encoding of the difference value. This noise can be reduced by using higher bit-depth encoding or by using techniques such as delta-sigma modulation. Another limitation is that it is sensitive to channel noise and can suffer from error propagation if errors occur in the encoded difference values.

Difference between TCP and UDP

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Type of Service	TCP is a connection-oriented protocol. Connection orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.	UDP is the Datagram-oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, or terminating a connection. UDP is efficient for broadcast and multicast types

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
		of network transmission.
Reliability	TCP is reliable as it guarantees the delivery of data to the destination router.	The delivery of data to the destination cannot be guaranteed in UDP.
Error checking mechanism	TCP provides extensive error-checking mechanisms. It is because it provides flow control and acknowledgment of data.	UDP has only the basic error-checking mechanism using checksums.
Acknowledgment	An acknowledgment segment is present.	No acknowledgment segment.
Sequence	Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in order at the receiver.	There is no sequencing of data in UDP. If the order is required, it has to be managed by the application layer.

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Speed	TCP is comparatively slower than UDP.	UDP is faster, simpler, and more efficient than TCP.
Retransmission	Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in the User Datagram Protocol (UDP).
Header Length	TCP has a (20-60) bytes variable length header.	UDP has an 8 bytes fixed-length header.
Weight	TCP is heavy-weight.	UDP is lightweight.
Handshaking Techniques	Uses handshakes such as SYN, ACK, SYN-ACK	It’s a connectionless protocol i.e. No handshake
Broadcasting	TCP doesn’t support Broadcasting.	UDP supports Broadcasting.

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Protocols	TCP is used by HTTP, HTTPs, FTP, SMTP and Telnet.	UDP is used by DNS, DHCP, TFTP, SNMP, RIP, and VoIP.
Stream Type	The TCP connection is a byte stream.	UDP connection is a message stream.
Overhead	Low but higher than UDP.	Very low.
Applications	This protocol is primarily utilized in situations when a safe and trustworthy communication procedure is necessary, such as in email, on the web surfing, and in military services.	This protocol is used in situations where quick communication is necessary but where dependability is not a concern, such as VoIP, game streaming, video, and music streaming, etc.

Compare Go Back N and SR

S.NO	Go-Back-N Protocol	Selective Repeat Protocol
1.	In Go-Back-N Protocol, if the sent frame are find suspected then all the frames are re-transmitted from the lost packet to the last packet transmitted.	In selective Repeat protocol, only those frames are re-transmitted which are found suspected.
2.	Sender window size of Go-Back-N Protocol is N.	Sender window size of selective Repeat protocol is also N.
3.	Receiver window size of Go-Back-N Protocol is 1.	Receiver window size of selective Repeat protocol is N.
4.	Go-Back-N Protocol is less complex.	Selective Repeat protocol is more complex.
5.	In Go-Back-N Protocol, neither sender nor at receiver need sorting.	In selective Repeat protocol, receiver side needs sorting to sort the frames.
6.	In Go-Back-N Protocol, type of Acknowledgement is cumulative.	In selective Repeat protocol, type of Acknowledgement is individual.
7.	In Go-Back-N Protocol, Out-of-Order packets are NOT Accepted (discarded) and the entire window is re-transmitted.	In selective Repeat protocol, Out-of-Order packets are Accepted.
8.	In Go-Back-N Protocol, if Receives a corrupt packet, then also, the entire window is re-transmitted.	In selective Repeat protocol, if Receives a corrupt packet, it immediately sends a negative acknowledgement and hence only

S.NO	Go-Back-N Protocol	Selective Repeat Protocol
		the selective packet is retransmitted.
9.	Efficiency of Go-Back-N Protocol is `N/(1+2*a)`	Efficiency of selective Repeat protocol is also `N/(1+2*a)`

Advantages of fiber optic cable over twisted pair cables

1. **Greater Bandwidth**

Copper cables were originally designed for voice transmission and have a limited bandwidth. Fiber optic cables provide more bandwidth for carrying more data than copper cables of the same diameter. Within the fiber cable family, **singlemode fiber** delivers up to twice the throughput of **multimode fiber**.

2. **Faster Speeds**

Fiber optic cables have a core that carries light to transmit data. This allows fiber optic cables to carry signals at speeds that are only about 31 percent slower than the speed of light—faster than Cat5 or Cat6 copper cables. There is also less signal degradation with fiber cables.

3. **Longer Distances**

Fiber optic cables can carry signals much farther than the typical 328-foot limitation for copper cables. For example, some 10 Gbps singlemode fiber cables can carry signals almost 25 miles. The actual distance depends on the type of cable, the wavelength and the network.

4. **Better Reliability**

Fiber is immune to temperature changes, severe weather and moisture, all of which can hamper the connectivity of copper cable. Plus, fiber does not carry electric current, so it's not bothered by electromagnetic interference (EMI) that can interrupt data transmission. It also does not present a fire hazard like old or worn copper cables can.

5. **Thinner and Sturdier**

Compared to copper cables, fiber optic cables are thinner and lighter in weight. Fiber can withstand more pull pressure than copper and is less prone to damage and breakage.

6. More Flexibility for the Future

Media converters make it possible to incorporate fiber into existing networks. The converters extend UTP Ethernet connections over fiber optic cable. Modular patch panel solutions integrate equipment with 10 Gb, 40 Gb and 100/120 Gb speeds to meet current needs and provide flexibility for future needs. The panels in these solutions accommodate a variety of cassettes for different types of fiber patch cables.

7. Lower Total Cost of Ownership

Although some fiber optic cables may have a higher initial cost than copper, the durability and reliability of fiber can make the total cost of ownership (TCO) lower. And, costs continue to decrease for fiber optic cables and related components as technology advances.

What is the Need of MIME in email service? Draw and discuss MIME header.

Multipurpose Internet Mail Extension (MIME) is a standard that was proposed by Bell Communications in 1991 in order to expand the limited capabilities of email.
MIME is a kind of add-on *or a supplementary protocol* that allows non-ASCII data to be sent through SMTP. It allows the users to exchange different kinds of data files on the Internet: audio, video, images, application programs as well.

Why do we need MIME?

Limitations of Simple Mail Transfer Protocol (SMTP):

1. SMTP has a very simple structure
2. Its simplicity however comes with a price as it only sends messages in NVT 7-bit ASCII format.
3. It cannot be used for languages that do not support 7-bit ASCII format such as French, German, Russian, Chinese and Japanese, etc. so it cannot be transmitted using SMTP. So, in order *to make SMTP more broad, we use MIME*.
4. It cannot be used to send binary files or video or audio data.

Features of MIME –

1. It can send multiple attachments with a single message.
2. Unlimited message length.
3. Binary attachments (executables, images, audio, or video files) may be divided if needed.
4. MIME provided support for varying content types and multi-part messages.

MIME with SMTP and POP –
SMTP transfers the mail being a message transfer agent from the sender's side to the mailbox of the receiver side and stores it and MIME header is added to the original header and provides additional information. while POP being the message access agent organizes the mails from the mail server to the receiver's computer. POP allows the user agent to connect with the message transfer agent.

MIME Header:
It is added to the original e-mail header section to define transformation. There are *five headers* that we add to the original header:

1. **MIME-Version –** Defines the version of the MIME protocol. It must have the parameter *Value 1.0*, which indicates that message is formatted using MIME.
2. **Content-Type –** Type of data used in the body of the message. They are of different types like text data (plain, HTML), audio content, or video content.
3. **Content-Type Encoding –** It defines the method used for encoding the message. Like 7-bit encoding, 8-bit encoding, etc.
4. **Content Id –** It is used for uniquely identifying the message.
5. **Content description –** It defines whether the body is actually an image, video, or audio

Differentiate between RIP and OSPF (10 Points)

RIP	OSPF
RIP Stands for Routing Information Protocol.	OSPF stands for Open Shortest Path First.
RIP works on the Bellman-Ford algorithm.	OSPF works on Dijkstra algorithm.
It is a Distance Vector protocol and it uses the distance or hops count to determine the transmission path.	It is a link-state protocol and it analyses different sources like the speed, cost and path congestion while identifying the shortest path.
It is used for smaller size organizations.	It is used for larger size organizations in the network.

RIP	OSPF
It allows a maximum of 15 hops.	There is no such restriction on the hop count.
It is not a more intelligent dynamic routing protocol.	It is a more intelligent routing protocol than RIP.
The networks are classified as areas and tables here.	The networks are classified as areas, sub-areas, autonomous systems, and backbone areas here.
Its administrative distance is 120.	Its administrative distance is 110.
RIP uses UDP(User Datagram Protocol) Protocol.	OSPF works for IP(Internet Protocol) Protocol.
It calculates the metric in terms of Hop Count.	It calculates the metric in terms of bandwidth.
In RIP, the whole routing table is to be broadcasted to the neighbors every 30 seconds by the routers.	In OSPF, parts of the routing table are only sent when a change has been made to it.
RIP utilizes less memory compared to OSPF but is CPU intensive like OSPF.	OSPF device resource requirements are CPU intensive and memory.
It consumes more bandwidth because of greater network resource requirements in sending the whole routing table.	It consumes less bandwidth as only part of the routing table is to send.
Its multicast address is 224.0.0.9.	OSPF’s multicast addresses are 224.0.0.5 and 224.0.0.6.

Differentiate between Software Defined Network and Traditional Network. (10 Points)

S.No.	SDN	TRADITIONAL NETWORK
01.	Software Defined Network is virtual networking approach.	Traditional network is the old conventional networking approach.
02.	Software Defined Network is centralized control.	Traditional Network is distributed control.
03.	This network is programmable.	This network is non programmable.
04.	Software Defined Network is open interface.	Traditional network is closed interface.
05.	In Software Defined Network data plane and control plane are decoupled by software.	In traditional network data plane and control plane are mounted on same plane.
06.	It supports automatic configuration so it takes less time.	It supports static/manual configuration so it takes more time.
07.	It can prioritize and block specific network packets.	It leads all packets in the same way no prioritization support.
08.	It is easy to program as per need.	It is difficult to program again and to replace existing program as per use.
09.	Cost of Software Defined Network is low.	Cost of Traditional Network is high.
10.	Structural complexity is low in Software Defined Network.	Structural complexity is high in Traditional Network.
11.	Extensibility is high in Software Defined Network.	Extensibility is low in Traditional Network.
12.	In SDN it is easy to troubleshooting and reporting as it is centralized controlled.	In Traditional network it is difficult to troubleshoot and report as it is distributed controlled.
13.	Its maintenance cost is lower than traditional network.	Traditional network maintenance cost is higher than SDN.

Describe how Web caching can reduce the delay in receiving a requested object. Will Web caching reduce the delay for all objects requested by a user or for only some of the objects? Justify?

Web caching is a technique used to store copies of web objects (such as web pages, images, and videos) closer to the user, typically on a caching server located at the edge of the network. By caching frequently requested objects, web caching aims to reduce the delay in receiving those objects when requested by users. Here's how web caching works and its impact on reducing delay:

1. Caching Mechanism: When a user requests a web object, the request first goes to the caching server. The caching server checks if it has a copy of the requested object in its cache. If the object is present, it is served directly from the cache to the user, eliminating the need to retrieve it from the original source server. This significantly reduces the delay in delivering the object since the caching server is typically located closer to the user, resulting in shorter network latency.
2. Reduced Network Traffic: By serving cached objects, web caching reduces the amount of network traffic generated by repeatedly fetching the same object from the source server. This helps alleviate congestion on the network and allows for more efficient utilization of available bandwidth.
3. Localized Storage: Caching servers are strategically placed at various locations within a network, including Internet service provider (ISP) networks, content delivery networks (CDNs), or even within enterprises. Placing caches closer to end-users ensures that frequently accessed objects are readily available nearby, reducing the round-trip time required to fetch those objects from distant servers.
4. Popular and Time-Insensitive Objects: Web caching is most effective for objects that are popular and relatively stable over time. These objects, such as images, style sheets, or static web pages, are frequently requested by multiple users. Caching them significantly reduces the delay for subsequent requests from users, as they can be served directly from the cache. These

objects are less likely to change frequently, ensuring that the cached copies remain valid and up to date.

5. Limited Impact on Dynamic or Personalized Content: Web caching may have limited impact on dynamic or personalized content that is unique to each user or frequently updated. Such content, which includes user-specific pages, dynamically generated content, or real-time data, is typically not suitable for caching. Each user's request for these objects would require personalized or real-time data, which cannot be efficiently served from a cache. In such cases, the delay reduction provided by web caching would be minimal or negligible.

In summary, web caching reduces delay by storing frequently requested objects closer to the user, minimizing the need to fetch those objects from distant servers. However, its effectiveness in reducing delay depends on the popularity and stability of the objects. While it significantly improves the delivery speed of popular and time-insensitive content, dynamic or personalized content may not benefit from web caching and may require direct retrieval from the original source server.

With respect to reliable data transfer(rdt) protocols, describe the need of sequence numbers in rdt2.1 and timers in rdt3.0. Demonstrate with examples.

In reliable data transfer (rdt) protocols, sequence numbers and timers play crucial roles in ensuring accurate and timely delivery of data between a sender and receiver. Let's explore the need for sequence numbers in rdt2.1 and timers in rdt3.0, along with examples:

1. Sequence Numbers in rdt2.1: In rdt2.1, sequence numbers are essential for handling potential packet loss and duplicate packets.

Scenario: Sender S transmits packets to Receiver R.

a. Packet Loss: Sequence numbers allow R to detect lost packets. When R receives packets out of order or with missing sequence numbers, it can request retransmission from S for the missing packets.

Example: S sends packets with sequence numbers: 0, 1, 2, 3, 4. If R receives packets in the order 0, 1, 3, 4 (missing packet 2), it can request retransmission of packet 2.

b. Duplicate Packets: Sequence numbers help R identify duplicate packets. R can discard duplicate packets with the same sequence number to avoid processing or storing redundant data.

Example: S sends packets with sequence numbers: 0, 1, 2, 3. Due to network issues, packet 1 is duplicated, and R receives packets with sequence numbers: 0, 1, 1, 2, 3. R can discard the duplicate packet with sequence number 1, ensuring that only one copy of each packet is processed.

2. Timers in rdt3.0: In rdt3.0, timers are introduced to handle potential packet loss and delays, ensuring reliable data transfer even in the presence of network issues.

Scenario: Sender S transmits packets to Receiver R.

a. Packet Loss: Timers help S detect packet loss. S starts a timer when it sends a packet and waits for an acknowledgment (ACK) from R. If the timer expires before receiving the ACK, S assumes the packet was lost and retransmits it.

Example: S sends packet 0 and starts a timer. If the ACK for packet 0 is not received within a specified time, the timer expires, and S retransmits packet 0.

b. Delayed Packets: Timers prevent indefinite waiting for delayed packets. If a packet is delayed in the network and does not reach R within a certain time, R can detect the delay and request S to retransmit the packet.

Example: S sends packet 0, but it experiences a significant delay in the network. If R does not receive packet 0 within a specified time, it can request S to retransmit the packet.

Overall, timers in rdt3.0 provide a mechanism for the sender to manage retransmissions and handle potential delays in packet delivery, improving the reliability of the data transfer process.

Compare and contrast:

1) Pure ALOHA and Slotted ALOHA

2) Flow control and Congestion control

1. Pure ALOHA and Slotted ALOHA:

   a) Pure ALOHA:

   - In Pure ALOHA, a station can transmit data whenever it has a frame to send, without checking for other ongoing transmissions.
   - Collisions may occur when two or more stations transmit simultaneously, resulting in the loss of data.
   - Stations detect collisions by listening for acknowledgments (ACKs) from other stations. If no ACK is received, a station assumes a collision and retransmits the frame after a random backoff period.

b) Slotted ALOHA:

- Slotted ALOHA divides time into fixed-size slots, and stations are allowed to transmit data only at the beginning of a time slot.
- This synchronized transmission approach reduces the probability of collisions compared to Pure ALOHA.
- Stations listen for ACKs and retransmit if no ACK is received.

Key Differences:

- Pure ALOHA allows random and immediate transmissions, while Slotted ALOHA restricts transmissions to specific time slots.

- In Pure ALOHA, collisions can occur at any time, leading to a higher probability of collisions. Slotted ALOHA reduces collisions by enforcing time slot boundaries.
- Pure ALOHA requires stations to wait for a random backoff period after a collision, while Slotted ALOHA does not require additional backoff since the time slots provide synchronization.

Pure Aloha	Slotted Aloha
In this Aloha, any station can transmit the data at any time. In this, The time is continuous and not globally synchronized. Vulnerable time for Pure Aloha = 2 x Tt	In this, any station can transmit the data at the beginning of any time slot. In this, The time is discrete and globally synchronized. Vulnerable time for Slotted Aloha = Tt
In Pure Aloha, Probability of successful transmission of the data packet = $G \times e^{-2G}$reduce In Pure Aloha, Maximum efficiency = 18.4%	In Slotted Aloha, Probability of successful transmission of the data packet = $G \times e^{-G}$ In Slotted Aloha, Maximum efficiency = 36.8%
Pure Aloha doesn't reduces the number of collisions to half.	Slotted Aloha reduces the number of collisions to half and doubles the efficiency of Pure Aloha.

2. Flow Control and Congestion Control:

   a) Flow Control:

- Flow control is a mechanism to manage the rate of data transmission between a sender and receiver to prevent the receiver from being overwhelmed.
- It ensures that the sender does not transmit data faster than the receiver can process it, avoiding data loss or buffer overflow at the receiver.
- Flow control is typically implemented through techniques such as sliding window protocols, where the receiver advertises its buffer capacity to the sender, allowing the sender to adjust its transmission rate accordingly.

b) Congestion Control:

- Congestion control aims to manage network congestion, which occurs when the demand for network resources exceeds its capacity.
- It prevents network congestion from degrading performance, causing packet loss, increased delays, or reduced throughput.
- Congestion control mechanisms, such as TCP congestion control algorithms, dynamically adjust the sender's transmission rate based on network conditions, packet loss, and other congestion indicators.

Key Differences:

- Flow control operates at the individual sender-receiver level, ensuring smooth data transfer between them. Congestion control operates at a network-wide level, managing overall network traffic.
- Flow control primarily focuses on preventing receiver buffer overflow. Congestion control focuses on preventing network congestion by dynamically adjusting the transmission rate.
- Flow control is typically implemented at the transport layer (e.g., TCP). Congestion control is implemented at both the transport and network layers.

It's important to note that while flow control and congestion control serve different purposes, they both contribute to ensuring reliable and efficient data transfer in computer networks.

S.NO	Flow Control	Congestion Control
1.	Traffic from sender to receiver is controlled, to avoid overwhelming the slow receiver.	Traffic entering the network from a sender is controlled by reducing rate of packets. Here, the sender must control/modulate his own rate to achieve optimal network utilization.
2.	Flow control is typically used in data link layer.	Congestion control is applied in network and transport layer.
3.	In this, Receiver's data is prevented from being overwhelmed.	In this, Network is prevented from congestion.
4.	In flow control, sender needs to take measures to avoid receiver from being overwhelmed depending on feedback from receiver and in absence of any feedback.	In this, many algorithms designed for transport layer/network layer define how endpoints should behave to avoid congestion.
5.	Types of Flow control are 1. Stop and Wait – For every frame transmitted, sender expects ACK from receiver. 2. Sliding Window – ACK needed only after sender transmits data until window is full, which is	Mechanisms designed to prevent network congestions are 1. Network Queue Management 2. Explicit Congestion Notification 3. TCP Congestion control

	allocated initially by receiver.	

Discuss roles and responsibility of the following OSI layers:

1) Networks layer

2) Presentation layer

3) Session layer

4)Physical layer

5) Application layer

1. Network Layer:
    - Responsible for establishing and maintaining logical connections between network devices.
    - Provides routing and forwarding of data packets between different networks.
    - Determines the optimal path for packet delivery based on network conditions and routing protocols.
    - Manages logical addressing and translates logical addresses (IP addresses) to physical addresses (MAC addresses) for data transmission.
    - Examples of protocols operating at this layer include IP (Internet Protocol) and ICMP (Internet Control Message Protocol).

2. Presentation Layer:
    - Handles the syntax and semantics of data exchanged between different systems.
    - Translates data from the format used by the application layer into a common format for transmission and vice versa.
    - Performs data compression and encryption to ensure secure and efficient data transfer.
    - Deals with data representation, including character encoding, data format conversions, and data compression algorithms.
    - Examples of protocols operating at this layer include SSL/TLS (Secure Sockets Layer/Transport Layer Security) and MIME (Multipurpose Internet Mail Extensions).
3. Session Layer:
    - Establishes, maintains, and terminates communication sessions between applications running on different hosts.
    - Synchronizes and manages dialogue control, ensuring orderly data exchange between applications.
    - Handles session checkpointing and recovery to resume interrupted sessions.
    - Manages authentication and authorization processes for secure communication.
    - Examples of protocols operating at this layer include NetBIOS (Network Basic Input/Output System) and SIP (Session Initiation Protocol).
4. Physical Layer:
    - Deals with the physical transmission of raw bit streams over the physical medium.
    - Specifies electrical, mechanical, and timing interfaces for transmitting bits.

- Handles modulation, coding, and signaling schemes used to represent bits as electrical or optical signals.
- Defines physical media characteristics such as voltage levels, transmission rates, and physical connectors.
- Examples of technologies operating at this layer include Ethernet, Wi-Fi, fiber optics, and DSL (Digital Subscriber Line).

5. Application Layer:
    - Provides interface and services to end-user applications for network communication.
    - Implements protocols and standards that enable specific applications to communicate with each other.
    - Handles application-specific functionalities, such as email services (SMTP), web browsing (HTTP), file transfer (FTP), and domain name resolution (DNS).
    - Supports user authentication, data formatting, content negotiation, and application-level security.
    - Examples of protocols operating at this layer include HTTP, FTP, SMTP, DNS, and SSH (Secure Shell).

Each OSI layer plays a distinct role in the communication process, contributing to the overall functionality and efficiency of network communication.

What are

1) Network Addresses

2) Broadcast addresses?

In case Class C IP addressing scheme, explain why one bit masking is treated as in-valid?

1. Network Addresses:

- Network addresses are identifiers assigned to network segments or subnets within an IP network.
- They help in routing data packets to the correct network destination.
- In IPv4, network addresses are represented by the combination of the network ID and host ID.
- The network ID identifies the specific network, while the host ID identifies individual hosts within that network.
- Network addresses are essential for the proper functioning of IP routing and subnetting.

2. Broadcast Addresses:
    - Broadcast addresses are special addresses used to send a message to all devices within a specific network or subnet.
    - When a device sends a packet to a broadcast address, all devices within the network receive and process the packet.
    - Broadcast addresses are used for network-wide announcements, service discovery, and other types of broadcast-based communication.
    - In IPv4, the broadcast address is typically the highest possible address within a network or subnet. For example, in a Class C network (with a subnet mask of 255.255.255.0), the broadcast address would be the IP address ending in ".255".

In the Class C IP addressing scheme, the first three bits are fixed at 110, defining the range of Class C addresses. The remaining 21 bits are used for host identification. Subnetting allows for dividing a Class C network into smaller subnets by borrowing bits from the host portion. However, in Class C, only a maximum of 8 bits can be borrowed for subnetting, ensuring enough bits are left for host addresses. If more than 8 bits are borrowed, it would result in fewer available host addresses, making it invalid. The minimum requirement is to leave 21 bits for host addressing to accommodate enough hosts within the network.

What are port numbers? Discuss with respect to open and reserved ports. UDP is a message-oriented protocol. TCP is a byte-oriented protocol. If an application needs to protect the boundaries of its message, which protocol should be used, UDP or TCP and Why?

Port numbers are identifiers used to specify a particular process or service running on a device within a network. In the context of TCP/IP networking, port numbers are a crucial component of the transport layer protocols, such as TCP and UDP. They allow multiple applications to share the same network interface while ensuring that data packets are delivered to the correct destination process.

Port numbers are 16-bit unsigned integers, ranging from 0 to 65535. They are divided into three ranges:

1. Well-known Ports (0-1023): These ports are reserved for widely used services and protocols. Examples include port 80 for HTTP, port 443 for HTTPS, port 25 for SMTP, and port 22 for SSH. These ports are standardized and assigned by the Internet Assigned Numbers Authority (IANA).
2. Registered Ports (1024-49151): These ports are assigned to specific services or protocols by the IANA upon request. They are often used by well-known applications but are not as widely recognized as well-known ports.
3. Dynamic or Private Ports (49152-65535): These ports are available for temporary or private use by applications. They can be used by client applications for initiating connections and are not registered or standardized.

Now, regarding whether UDP or TCP should be used to protect the boundaries of an application's message:

If an application needs to protect the boundaries of its message and ensure reliable, ordered delivery, TCP is the preferred choice. Here's why:

1. Byte-oriented Protocol: TCP is a byte-oriented protocol, meaning it treats data as a stream of bytes rather than discrete messages. It guarantees in-order delivery of data and provides mechanisms for retransmission and

error recovery. TCP maintains the integrity and sequence of data, which is crucial for applications that require message boundaries to be preserved.

2. Reliability: TCP is a reliable protocol that ensures all data is received and delivered in the correct order. It provides error detection, retransmission, and flow control mechanisms to guarantee the delivery of data. This reliability is essential for applications that require the complete and accurate transfer of messages.

UDP, on the other hand, is a message-oriented protocol that does not provide the same level of reliability and ordering guarantees as TCP. UDP is a connectionless protocol that does not establish a dedicated connection before data transmission. It does not perform error recovery or retransmission. While UDP is useful for applications that prioritize speed and low overhead, it does not inherently provide the necessary mechanisms to protect message boundaries and ensure reliable delivery.

Therefore, if an application needs to protect the boundaries of its messages and requires reliable, ordered delivery, TCP should be used. TCP's byte-oriented nature and reliability mechanisms make it suitable for applications that rely on maintaining message boundaries and ensuring accurate data transfer.

It's 1989. Alice and Bob are 4 hops apart on a datagram packet switched network where each link is 100 mile long. Per-hop processing delay is 10 micro-seconds. Packets are 1500 bytes long. All links have a transmission speed of 56kbit/s (original speed of Internet backbone links in the 80s). The speed of light in the wire is approximately 125,000 miles/s. If Bob sends a 10-packet message to Alice, a) How long will it take Alice to receive the message up to the last bit (measured from the time Bob starts sending)?

b)22 years later, all is the same, except that link transmission speed now is 1Gbit/s. How long will it take Alice to receive the message up to the last bit (measured from the time Bob starts sending)?

a) In 1989:

- Distance between Alice and Bob: 4 hops * 100 miles/hop = 400 miles
- Speed of light in the wire: 125,000 miles/s
- Per-hop processing delay: 10 microseconds
- Packet size: 1500 bytes
- Transmission speed: 56 kbit/s

To calculate the total transmission time for the 10-packet message from Bob to Alice, we need to consider the following components:

1. Transmission time: The time it takes to transmit the packets over each link.
2. Propagation delay: The time it takes for the signal to propagate between hops.
3. Processing delay: The time it takes for each hop to process the packets.

Transmission time per packet: Packet size = 1500 bytes = 12,000 bits Transmission time = Packet size / Transmission speed = 12,000 bits / 56,000 bits/s = 0.2143 seconds

Propagation delay per hop: Propagation delay = Distance / Speed of light = 400 miles / 125,000 miles/s = 0.0032 seconds

Total time for 4 hops (excluding processing delay): Total time = Transmission time per packet * Number of packets + Propagation delay per hop * (Number of packets - 1) = 0.2143 seconds * 10 + 0.0032 seconds * (10 - 1) = 2.143 seconds + 0.0256 seconds = 2.1686 seconds

Adding the per-hop processing delay (10 microseconds * 4 hops = 0.00004 seconds) to the total time: Total time with processing delay = 2.1686 seconds + 0.00004 seconds = 2.16864 seconds

Therefore, it will take approximately 2.16864 seconds for Alice to receive the message up to the last bit in 1989.

b) In 2011 (22 years later):

- Link transmission speed: 1 Gbit/s

Using the same calculations as in part (a), with the updated transmission speed of 1 Gbit/s, the transmission time per packet becomes: Transmission time per packet = Packet size / Transmission speed = 12,000 bits / 1,000,000,000 bits/s = 0.000012 seconds

Total time for 4 hops (excluding processing delay): Total time = Transmission time per packet * Number of packets + Propagation delay per hop * (Number of packets - 1) = 0.000012 seconds * 10 + 0.0032 seconds * (10 - 1) = 0.00012 seconds + 0.0256 seconds = 0.02572 seconds

Adding the per-hop processing delay (10 microseconds * 4 hops = 0.00004 seconds) to the total time: Total time with processing delay = 0.02572 seconds + 0.00004 seconds = 0.02576 seconds

Therefore, it will take approximately 0.02576 seconds (or 25.76 milliseconds) for Alice to receive the message up to the last bit 22 years later, assuming the link transmission speed has increased to 1 Gbit/s.

What is the purpose random access MAC protocols? Explain Slotted ALOHA with its pros and cons.

The purpose of random-access MAC (Media Access Control) protocols is to allow multiple devices or nodes to share a common communication channel in a decentralized manner. These protocols provide a set of rules for devices to access the channel, avoid collisions, and ensure fair and efficient utilization of the available bandwidth.

Slotted ALOHA is one such random access MAC protocol. It operates in discrete time slots and is based on the concept of contention. Here's how Slotted ALOHA works:

1. Time Division: The time is divided into equal slots, where each slot corresponds to the time required to transmit a single data packet.
2. Transmission Attempt: When a device has data to transmit, it waits for the beginning of the next time slot and then transmits the entire packet within that slot.
3. Collision Detection: After transmitting a packet, the device listens for an acknowledgment (ACK) from the receiver. If an ACK is not received within the time slot, the device assumes a collision has occurred.
4. Retransmission: In the event of a collision, the device waits for a random time period and retransmits the packet in a subsequent time slot. The random time period reduces the probability of collisions recurring.

Advantages of Slotted ALOHA:

- **Simplicity:** The Slotted Aloha protocol is relatively simple to implement and understand, making it an easy option for low-complexity networks.
- **Flexibility:** Slotted Aloha can be used in a wide range of network environments, including those with varying numbers of nodes and varying traffic loads
- **Low overhead:** Slotted Aloha does not require complex management or control mechanisms, which can help to reduce the overhead and complexity of the network.

Disadvantages of Slotted ALOHA:

- **Low throughput:** The maximum throughput of the Slotted Aloha protocol is relatively low at around 36.8 %, which can be limiting for high-bandwidth applications.
- **High collision rate:** The high collision rate in slotted ALOHA can result in a high packet loss rate, which can negatively impact the overall performance of the network.

- **Inefficiency:** The protocol is inefficient at high loads, as the efficiency decreases as the number of nodes attempting to transmit increases.

Overall, Slotted ALOHA provides a simple and decentralized approach for sharing a communication channel among multiple devices. While it offers advantages such as ease of implementation and fair utilization of the channel, it also has limitations in terms of channel efficiency and suitability for heavy traffic scenarios

How UDP detect errors in transmitted segment. Explain with an example.

UDP (User Datagram Protocol) does not have built-in error detection or correction mechanisms. It is a simple and lightweight transport protocol that provides a connectionless and unreliable delivery of datagrams. Unlike TCP, which implements error detection and retransmission, UDP leaves the responsibility of error detection and recovery to the application layer.

However, it's worth mentioning that UDP does include a basic checksum mechanism to detect errors in the transmitted segment. The checksum is calculated by the sender and included in the UDP header. When the receiver receives the UDP segment, it recalculates the checksum and compares it with the checksum value in the header. If the calculated checksum and the received checksum do not match, it indicates that errors or corruption may have occurred during transmission.

Here's an example to illustrate how UDP detects errors using the checksum:

1. Sender Side: Let's assume the sender wants to transmit a UDP segment containing a message "Hello" to the receiver.
    - The sender calculates a checksum for the entire UDP segment, including the UDP header and the message.
    - The checksum value is then inserted into the UDP header.
2. Receiver Side: Upon receiving the UDP segment, the receiver performs the following steps:

- The receiver extracts the UDP segment from the received packet.
- It recalculates the checksum for the received segment, including the UDP header and the message.
- The receiver compares the calculated checksum with the checksum value in the UDP header.

If the calculated checksum matches the received checksum, it indicates that the UDP segment is likely error-free. However, if the calculated checksum does not match the received checksum, it suggests that errors or corruption might have occurred during transmission. In such cases, the receiver cannot automatically correct the errors or request retransmission as UDP does not provide these capabilities. Instead, it is up to the application layer to handle error detection and recovery, if necessary.

It's important to note that while UDP's checksum can detect errors, it is not as robust as the error detection and correction mechanisms employed by TCP. UDP's simplicity and lack of error recovery make it suitable for applications where real-time communication and low latency are more important than data reliability, such as streaming media or real-time gaming.

Describe the flow of UDP sender and Receiver actions.

The flow of actions in a UDP (User Datagram Protocol) sender and receiver can be described as follows:

UDP Sender Actions:

1. Create a UDP socket: The sender creates a UDP socket to establish communication with the receiver.
2. Prepare the UDP segment: The sender prepares the UDP segment, which includes the UDP header and the data (payload) to be sent. The UDP header contains the source and destination port numbers, length, and optional checksum.

3. Specify the destination address and port: The sender specifies the destination IP address and port number to which the UDP segment will be sent.
4. Send the UDP segment: The sender uses the socket to send the UDP segment to the specified destination address and port.
5. End transmission: After sending the UDP segment, the sender may close the socket if it no longer needs to send any more data.

UDP Receiver Actions:

1. Create a UDP socket: The receiver creates a UDP socket to listen for incoming UDP segments.
2. Bind the socket to a specific port: The receiver binds the UDP socket to a specific port number so that it can receive UDP segments sent to that port.
3. Receive the UDP segment: The receiver waits for incoming UDP segments on the bound socket. When a UDP segment arrives, the receiver reads it from the socket.
4. Extract the data: The receiver extracts the data (payload) from the received UDP segment for further processing.
5. Process the data: The receiver performs any necessary processing or actions on the received data according to the application's requirements.
6. Optional: Send a response: If a response is required, the receiver can create a new UDP segment and send it back to the sender's address and port specified in the received UDP segment.
7. Continue listening: After processing the received UDP segment, the receiver continues to listen for incoming UDP segments on the bound socket.

It's important to note that UDP is a connectionless protocol, so there is no formal connection setup or teardown process like in TCP. The sender and receiver communicate by exchanging UDP segments, and each segment is treated independently without any inherent ordering or reliability guarantees.

What is socket programming? Explain the working principle of client server model.

Socket programming is a method of network communication that allows processes (programs) running on different computers to exchange data. It provides a programming interface for network communication using sockets, which are endpoints for sending and receiving data across a network.

The client-server model is a common architecture used in socket programming, where there are two entities involved: the client and the server.

Working principle of the client-server model:

1. Server Setup:
    - The server application creates a socket and binds it to a specific port number on the server machine.
    - The server then listens for incoming connections on that port.
2. Client Connection:
    - The client application creates a socket and specifies the IP address and port number of the server it wants to connect to.
    - The client requests a connection to the server by initiating a connection request.
3. Server Acceptance:
    - When the server receives a connection request, it accepts the connection by creating a new socket dedicated to that specific client.
    - This new socket is used for communication with the client.
4. Data Exchange:
    - Once the connection is established, the client and server can exchange data through their respective sockets.
    - The client can send requests to the server, and the server can respond with the requested data or perform the necessary actions.

- The data is typically transmitted in chunks or packets, and both the client and server can send and receive data simultaneously.

5. Connection Termination:
    - Either the client or the server can initiate the termination of the connection when the communication is complete.
    - The initiating party sends a termination request, and the other party acknowledges the termination request.
    - Once the termination process is completed, the connection is closed, and the sockets are released.

Socket programming allows for flexible and customizable network communication between clients and servers. It enables the exchange of data, control messages, and information between distributed applications over a network. By following the client-server model, applications can establish connections, exchange data, and terminate connections as needed to facilitate effective communication.

What are the fragmentation field in IP Datagram. Give importance of these fields

In an IP datagram (packet), the fragmentation field consists of several fields that are used to handle the fragmentation and reassembly of packets when they are too large to fit within the Maximum Transmission Unit (MTU) of a network.

The fragmentation field in an IP datagram includes the following fields:

1. Identification: This field is a unique identifier assigned to each datagram by the sender. It helps the receiving host reassemble the fragmented packets correctly by matching the identification field.
2. Flags: The flags field consists of three bits: Reserved, Don't Fragment (DF), and More Fragments (MF).
    - Reserved: These bits are reserved for future use and are typically set to zero.

- DF (Don't Fragment): When set to 1, it indicates that the packet should not be fragmented and should be discarded if it cannot be delivered without fragmentation.
- MF (More Fragments): When set to 1, it indicates that more fragments of the original packet follow the current fragment.

3. Fragment Offset: This field specifies the position of the fragment within the original packet, measured in 8-byte units (or octets). It indicates the relative position of the current fragment in the entire packet.

Importance of the fragmentation fields:

1. Fragmentation and Reassembly: The fragmentation fields are crucial for breaking down large IP packets into smaller fragments to fit within the MTU of the underlying network. This allows the transmission of packets across networks with different maximum packet sizes.
2. Correct Reassembly: The identification field helps the receiving host correctly reassemble the fragmented packets. It ensures that all the fragments belonging to the same original packet are properly matched and reconstructed.
3. Handling MTU Differences: Different networks may have different MTUs, which can result in packet fragmentation. The DF flag allows a sender to request that packets should not be fragmented, ensuring that they will be either delivered intact or discarded entirely if fragmentation is required.
4. Efficient Data Transmission: Fragmentation allows for more efficient data transmission by adapting packet sizes to network constraints. It minimizes the need for data retransmission and improves overall network performance.
5. Path MTU Discovery: The fragmentation fields are used in the Path MTU Discovery process to determine the maximum MTU size along a path. By determining the maximum MTU, unnecessary fragmentation can be avoided, improving efficiency, and reducing packet loss.

By including fragmentation fields in IP datagrams, the IP protocol provides a mechanism for breaking down and reassembling packets to ensure successful

transmission across networks with different MTUs. This enables efficient data transmission and helps maintain the integrity of transmitted packets.

State hierarchical name space. Give role of any two DNS resolver

Hierarchical name space refers to the organization and structure of domain names in the Domain Name System (DNS). It is based on a hierarchical naming scheme where domain names are divided into meaningful components, separated by periods (dots), creating a tree-like structure.

In a hierarchical name space:

1. Domain Names: Domain names are divided into multiple levels, from right to left. Each level represents a specific domain or subdomain, with the top-level domain (TLD) being the highest level (e.g., .com, .org, .net). Subdomains can be added to create a deeper hierarchy (e.g., subdomain.example.com).
2. Authority and Delegation: Responsibility for managing different levels of the hierarchy is delegated to different entities. Each domain or subdomain can have its own authoritative name servers, which are responsible for storing and providing DNS information for that domain. This hierarchical delegation allows for efficient distribution of management and resolution tasks across the DNS infrastructure.

Role of DNS Resolvers:

1. Recursive Resolver: A recursive resolver is a DNS resolver that performs the full resolution process on behalf of a client. When a client sends a DNS query, the recursive resolver receives the query and starts the resolution process by contacting the authoritative name servers in a recursive manner. It sends queries to the appropriate name servers, retrieves the necessary information, and returns the result to the client. Recursive resolvers play a crucial role in resolving domain names and caching DNS records to improve future query response times.

2. Caching Resolver: A caching resolver is a DNS resolver that stores recently resolved DNS records in its cache. When a client sends a DNS query, the caching resolver first checks its cache for the requested domain name's information. If the information is available, the resolver can quickly retrieve it from the cache and provide the response to the client, without needing to perform a full resolution process. Caching resolvers help reduce the load on authoritative name servers and improve the overall efficiency of DNS resolution.

Both recursive and caching resolvers are important components of the DNS infrastructure. They handle the resolution process, improve response times, and help distribute the workload across the DNS system. Recursive resolvers ensure that clients receive complete and accurate DNS information, while caching resolvers reduce the need for repeated queries by storing commonly accessed records locally. Together, they contribute to the smooth functioning and performance of the DNS.

What do you mean by firewall? how does it control access to a system?

A firewall is a network security device or software that acts as a barrier between an internal network and external networks (such as the internet) to monitor and control incoming and outgoing network traffic. Its primary function is to enforce security policies and control access to a system or network by analysing network packets and applying rules to determine whether to allow or block traffic.

Firewalls control access to a system by employing various techniques:

1. Packet Filtering: Firewalls inspect individual packets of data based on predefined rules. These rules define criteria such as source IP address, destination IP address, port number, and protocol type. Packets that match the specified criteria are allowed to pass through, while those that violate the rules are either dropped or rejected.
2. Stateful Inspection: Stateful firewalls keep track of the state of network connections. They maintain information about established connections,

including source and destination IP addresses, port numbers, and connection status. By examining the context of packets in relation to the established connections, stateful firewalls can make more intelligent decisions about allowing or blocking traffic.

3. Application-Level Gateways (Proxy Firewalls): Proxy firewalls act as intermediaries between internal and external networks. They receive network traffic, terminate the connection with the sender, and establish a new connection on behalf of the recipient. This allows for more granular control over application-specific protocols, as the proxy can inspect and filter traffic at the application layer.
4. Network Address Translation (NAT): Firewalls often incorporate NAT functionality to mask the internal IP addresses of devices from external networks. NAT translates internal private IP addresses to a single public IP address, providing an additional layer of security by hiding the internal network topology.
5. Virtual Private Networks (VPNs): Firewalls may include VPN capabilities to establish secure, encrypted connections between remote networks or individuals. VPNs create a secure tunnel over the internet, allowing authorized users to access internal resources remotely while ensuring confidentiality and data integrity.

By implementing these techniques, firewalls control access to a system by allowing or denying network traffic based on predefined rules and policies. They act as a gatekeeper, inspecting and filtering packets to prevent unauthorized access, protect against malicious activities, and enforce network security policies.

The ethernet protocol is called CSMA/CD because it supports carrier sense, collision detection and backoff. For each of these three capabilities, give a brief explanation of what it is and why it offers an improvement over Slotted Aloha.

CSMA/CD (Carrier Sense Multiple Access with Collision Detection) is an access control method used in Ethernet networks. It offers several improvements over

the Slotted Aloha protocol, mainly by incorporating carrier sense, collision detection, and backoff mechanisms. Here's a brief explanation of each capability and why it enhances network performance compared to Slotted Aloha:

1. Carrier Sense:
    - Carrier sense refers to the ability of a device to sense the medium (the network cable) before transmitting data. Before sending a packet, a device using CSMA/CD listens for any ongoing transmissions on the network. If the medium is idle, it starts transmitting. If it detects another transmission, it waits until the medium becomes idle before sending its own packet.
    - Improvement over Slotted Aloha: Carrier sense helps avoid collisions by ensuring that a device waits until the network is not in use before sending data. This reduces the likelihood of collisions and improves overall network efficiency.
2. Collision Detection:
    - Collision detection enables a device to detect if a collision occurs during data transmission. In Ethernet, collisions occur when two or more devices transmit data simultaneously, resulting in corrupted packets. Each device monitors the network for collisions by comparing the transmitted signal with the received signal.
    - Improvement over Slotted Aloha: Collision detection allows devices to recognize when a collision occurs and take appropriate action. Upon detecting a collision, devices stop transmitting, wait for a random period, and then attempt to retransmit the data. This mechanism helps in recovering from collisions and avoids further collisions by using a backoff algorithm.
3. Backoff:
    - Backoff is a mechanism used in CsMA/CD to manage retransmissions after a collision. When a device detects a collision, it waits for a random period before retransmitting the data. The random backoff

period helps to avoid repeated collisions that may occur if devices keep retransmitting simultaneously.

- Improvement over Slotted Aloha: Slotted Aloha does not incorporate a backoff mechanism, which can lead to repeated collisions if devices immediately retransmit after a collision. Backoff in CsMA/CD helps to distribute retransmissions across time, reducing the chances of multiple devices colliding again and improving the overall efficiency of the network.

In summary, CSMA/CD offers improvements over Slotted Aloha by incorporating carrier sense, collision detection, and backoff mechanisms. Carrier sense ensures that devices wait for an idle medium before transmitting, collision detection allows devices to recognize and handle collisions, and backoff helps avoid repeated collisions by introducing random delays before retransmission. These enhancements increase the efficiency and reliability of Ethernet networks compared to the simpler Slotted Aloha protocol.

The OSI reference model has two additional layers. Where these layers in the stack are and what services do they provide?

The OSI (Open Systems Interconnection) reference model consists of seven layers, but it is important to note that there are two additional layers often referenced outside of the standard seven-layer model. These additional layers are typically referred to as the "upper layers" or "upper-level protocols."

The two additional layers are:

1. Presentation Layer:
    - Position in the OSI Model: The Presentation Layer is the sixth layer in the OSI model, located above the Session Layer and below the Application Layer.
    - Services Provided: The Presentation Layer is responsible for data representation and ensures the compatibility and interoperability of

different systems by handling the syntax and semantics of the information exchanged. It deals with the translation, compression, encryption, and formatting of data to be transmitted, ensuring that the data is properly understood by the receiving system. This layer also provides services such as data compression, encryption, and character encoding/decoding.

2. Session Layer:
    - Position in the OSI Model: The Session Layer is the fifth layer in the OSI model, located above the Transport Layer and below the Presentation Layer.
    - Services Provided: The Session Layer establishes, manages, and terminates communication sessions between network applications. It enables two endpoints (processes) to establish a connection, exchange data in either full-duplex or half-duplex mode, and terminate the session when communication is complete. The Session Layer handles synchronization, checkpointing, and recovery of sessions and supports session multiplexing, allowing multiple sessions to be established and managed simultaneously.

While the standard seven layers of the OSI model (from Physical Layer to Application Layer) are widely recognized and used, the inclusion of the Presentation Layer and Session Layer as additional layers above the Transport Layer is sometimes done to provide a more detailed understanding of the various protocols and services involved in networking.

Describe p2p architecture w.r.t URI persistence, cost, and privacy.

In a peer-to-peer (P2P) architecture, computers (known as peers) in the network share resources directly with each other without the need for a centralized server. P2P networks are decentralized and allow peers to act as both clients and servers, contributing and consuming resources simultaneously. When considering

URI persistence, cost, and privacy, P2P architectures have the following characteristics:

1. URI Persistence:
    - P2P architectures can have varying levels of URI persistence. In some P2P networks, the availability and persistence of URIs (Uniform Resource Identifiers) can be challenging because the resources are distributed across multiple peers. If a peer hosting a particular resource goes offline, the URI to access that resource may become unavailable until another peer in the network hosts it. However, in more robust P2P systems, redundancy and distributed data storage mechanisms can be implemented to enhance URI persistence, ensuring that resources remain accessible even if some peers go offline.
2. Cost:
    - P2P architectures can provide cost advantages compared to traditional client-server architectures. In a P2P network, resources are shared among peers, reducing the need for centralized infrastructure and costly server hardware. P2P systems can leverage the idle resources of individual peers, such as bandwidth and storage capacity, to distribute the load and decrease the overall cost of hosting and delivering content. This decentralized approach can be particularly beneficial for distributing large files or handling high volumes of traffic without the need for significant upfront investments in infrastructure.
3. Privacy:
    - P2P architectures can present privacy challenges due to the distributed nature of the network. In P2P networks, peers directly communicate with each other, which can raise concerns about privacy and security. While P2P systems can employ encryption and authentication mechanisms to protect data during transmission, the decentralized nature of the architecture means that peers may have direct access to the data being shared. This can potentially expose

sensitive information to unauthorized users if proper security measures are not implemented. Privacy considerations are crucial in P2P networks, and protocols and encryption techniques should be employed to protect the confidentiality and integrity of shared data.

Overall, P2P architectures offer benefits in terms of URI persistence, cost efficiency, and resource sharing. However, they also present challenges related to maintaining URI availability, ensuring privacy and security, and addressing potential scalability issues. The design and implementation of a P2P system should consider these factors to provide reliable and secure sharing of resources among peers.

What should the minimum value of the retransmission timer be in an ARQ protocol and why?

> Given values of `EstimatedRTT` and `DevRTT`, what value should be used for TCP's timeout interval? Clearly, the interval should be greater than or equal to `EstimatedRTT`, or unnecessary retransmissions would be sent. But the timeout interval should not be too much larger than `EstimatedRTT`; otherwise, when a segment is lost, TCP would not quickly retransmit the segment, leading to large data transfer delays. It is therefore desirable to set the timeout equal to the `EstimatedRTT` plus some margin. The margin should be large when there is a lot of fluctuation in the `SampleRTT` values; it should be small when there is little fluctuation. The value of `DevRTT` should thus come into play here. All of these considerations are taken into account in TCP's method for determining the retransmission timeout interval:
>
> `TimeoutInterval = EstimatedRTT + 4 · DevRTT`

Compare how exponential backoff algorithm is used in CSMA/CD versus CSMA / CA?

Exponential backoff algorithms are used in both CSMA/CD (Carrier Sense Multiple Access with Collision Detection) and CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) to manage collisions and control access to the shared medium. However, there are some differences in how the exponential backoff algorithm is employed in these two protocols:

CSMA/CD (used in Ethernet networks):

1. Collision Detection:
    - In CSMA/CD, if a collision is detected during data transmission, the transmitting device immediately stops sending and starts the backoff process.
    - The backoff process involves selecting a random waiting time from a range of predefined values based on the number of previous collisions.
    - The waiting time is typically increased exponentially with each collision. For example, if the first collision occurs, the device waits for a random time within a small range. If a second collision occurs, the device waits for a random time within a larger range, and so on.
    - This exponential backoff algorithm in CSMA/CD helps to distribute retransmissions across time, reducing the likelihood of repeated collisions and improving overall network efficiency.

CSMA/CA (used in wireless networks, such as Wi-Fi):

1. Collision Avoidance:
    - In CSMA/CA, collision avoidance is used instead of collision detection because it is difficult to detect collisions reliably in wireless networks.
    - Before transmitting data, a device using CSMA/CA listens to the wireless medium to check for any ongoing transmissions.
    - If the medium is idle, the device sends a small request-to-send (RTS) packet to the intended recipient, indicating its intention to transmit.
    - The recipient then sends a clear-to-send (CTS) packet to acknowledge the request and reserve the medium for the transmission.
    - Other devices within the range of the RTS and CTS packets defer their transmissions during this reservation period, avoiding collisions.
    - If a device does not receive an expected CTS packet or detects a collision during the RTS or CTS exchange, it assumes a collision has occurred and initiates a backoff process like CSMA/CD.

- The backoff process in CSMA/CA also involves selecting a random waiting time, but it typically uses a fixed backoff algorithm rather than an exponential one. The waiting time is still randomized, but the range remains constant for subsequent backoffs.

In summary, while both CSMA/CD and CSMA/CA use an exponential backoff algorithm to manage collisions, the specific implementation differs. CSMA/CD uses collision detection and increases the waiting time exponentially with each collision. On the other hand, CSMA/CA uses collision avoidance and employs a fixed backoff algorithm with randomized waiting times. These differences reflect the distinct characteristics and challenges of wired (CSMA/CD) and wireless (CSMA/CA) networks.

Q /An ISP is granted a block of addresses starting with 130.60.0.0. The ISP wants to distribute these blocks to 100 organizations with each organization receiving just eight addresses. Design the subblocks and give the slash notation for each subblock. Find out how many addresses are still available after these allocations.

Solution:

8=2^3
Host bit=3

Subnet	Net	. Subnet	. Host	Subnet IP
Subnet 0	130.60	00000000 00000	000	130.60.0.0/29
	130.60	00000000 00000	111	130.60.0.7/29
Subnet 98	130.60	00000011 00010	000	130.60.3.16/29
	130.60	00000011 00010	111	130.60.3.23/29
Subnet 99	130.60	00000011 00011	000	130.60.3.24/29
	130.60	00000011 00011	111	130.60.3.31/29

Number of granted addresses to the ISP=2^16=65536

Number of allocated addresses by the ISP=100*8=800

Number of available addresses=65536-800=64736

Subnets: 128

Hosts: 768

8 IPs in each Subnet

6 Useable Host Addresses in each subnet

Address: 120.60.4.0

Netmask: 255.255.252.0 = 22

Wildcard: 0.0.3.255

=>

Network: 120.60.4.0/22

Broadcast: 120.60.7.255

HostMin: 120.60.4.1

HostMax: 120.60.7.254

Hosts/Net: 1022

Subnets

Network: 120.60.4.0/29

Network: 120.60.4.8/29

Network: 120.60.4.16/29

Network: 120.60.4.24/29

Network: 120.60.4.32/29

Network: 120.60.4.40/29

Network: 120.60.4.48/29

Network: 120.60.4.56/29

Network: 120.60.4.64/29

Network: 120.60.4.72/29

Network: 120.60.4.80/29

Network: 120.60.4.88/29

Network: 120.60.4.96/29

.

.

.

.

Network: 120.60.7.152/29

Network: 120.60.7.160/29

Network: 120.60.7.168/29

Network: 120.60.7.176/29

Network: 120.60.7.184/29

Network: 120.60.7.192/29

Network: 120.60.7.200/29

Network: 120.60.7.208/29

Network: 120.60.7.216/29

Network: 120.60.7.224/29

Network: 120.60.7.232/29

Network: 120.60.7.240/29

Network: 120.60.7.248/29

we have a CIDR of /22 so how many address are left ?

i.e. $2^{(32-22)} = 2^{10}$

out of the 10 bits remaining we need to provide 100 Subnets

so we need Ceil(log 100) i.e. 7 bits from the remaining 10 bits for the subnet ID part, but since we use only 100 subnet IDs we are left with 28 IDs in spare becuase using 7 bits we can address 128 Networks but we only used 100,

and since 28 networks are remaining and each network (subnet) has 8 hosts asssociated, we still have 28*8=224 hosts left.

Justify that BGP is an exterior protocol.

BGP (Border Gateway Protocol) is classified as an exterior routing protocol due to the following reasons:

1. Routing between Autonomous Systems (AS): BGP is primarily used for routing between different Autonomous Systems, which are networks under separate administrative control. It enables communication and exchange of routing information between ASs. ASs are typically operated by different organizations or Internet Service Providers (ISPs).
2. Exchange of routing information across boundaries: BGP is designed to exchange routing information between different routing domains, also known as routing boundaries. These routing domains can be different ISPs, organizations, or networks that are geographically dispersed and operate independently.
3. Path selection across AS boundaries: BGP uses a path-vector algorithm to determine the best path for routing between ASs. It considers various factors such as AS path length, policy configurations, and path attributes to make routing decisions across AS boundaries. This distinguishes BGP from interior routing protocols like OSPF or RIP, which operate within a single AS.
4. Scalability and robustness: BGP is specifically designed to handle the scale and complexity of the global Internet routing system. It can manage the exchange of many routing prefixes and adapt to changes in network topology and policies. BGP provides mechanisms for route aggregation, filtering, and policy control to ensure efficient and stable routing across AS boundaries.

Overall, BGP's purpose is to enable routing between different autonomous systems and facilitate the exchange of routing information across external boundaries. This makes it an exterior routing protocol in contrast to interior protocols that operate within a single routing domain.

Compare and analyse the merits of link state routing algorithm over distance vector algorithm

Link state routing algorithms work by having each router in the network maintain a complete map of the network topology. This map includes information about all the other routers in the network and the links between them. Each router shares information about its directly connected neighbors with every other router in the network. This is done using link state advertisements (LSAs), which are small packets that contain information about the sending router's neighbors and their links.

Once a router receives an LSA from another router, it updates its map of the network topology to include the new information. The router then uses this map to independently calculate the shortest path to any destination using algorithms such as Dijkstra's algorithm. This algorithm works by iteratively selecting the closest unvisited node to the source and updating the shortest path to all its neighbors.

One advantage of link state routing over distance vector routing is that link state routing typically converges faster. This means that when there is a change in the network topology, such as a link failure, link state routing algorithms can quickly update their routing tables to reflect the new topology. This is because each router maintains a complete map of the network and can independently calculate the shortest path to any destination.

Another advantage of link state routing is that it is less prone to routing loops than distance vector routing. Routing loops can occur when inconsistent routing information causes packets to be forwarded in a loop between routers. Link state routing algorithms are less susceptible to this problem because they maintain a complete map of the network topology and can detect and avoid loops.

However, link state routing algorithms require more computational power and memory than distance vector routing algorithms. This is because they need to store and process a complete map of the network topology. In contrast, distance vector routing algorithms only need to maintain a simple routing table with information about the distance and direction to destination networks.

Distance vector routing algorithms work by having each router maintain a routing table that contains information about the distance and direction to destination networks. Each router shares its routing table with its immediate neighbors. The

neighbors then use this information to update their own routing tables based on the Bellman-Ford algorithm.

The Bellman-Ford algorithm works by iteratively relaxing the edges of a graph to find the shortest path from a source node to all other nodes. In the context of distance vector routing, this means that each router updates its routing table based on the information it receives from its neighbors. If a neighbor has a shorter path to a destination network, then the router will update its own routing table to reflect this new information.

One disadvantage of distance vector routing compared to link state routing is that it can take longer to converge. This is because it relies on information being propagated from one neighbor to another. If there is a change in the network topology, such as a link failure, it can take several rounds of updates before all routers have consistent information about the new topology.

Another disadvantage of distance vector routing is that it is more prone to routing loops than link state routing. This can happen when inconsistent routing information causes packets to be forwarded in a loop between routers. Distance vector routing algorithms use various techniques, such as split horizon and poison reverse, to try and prevent routing loops, but they are not always successful.

In summary, link state routing has several advantages over distance vector routing, including faster convergence and fewer routing loops. However, it requires more computational power and memory than distance vector routing. The choice between these two types of algorithms depends on the specific requirements of the network.

Justify that TCP is connection-oriented protocol.

TCP (Transmission Control Protocol) is considered a connection-oriented protocol due to the following justifications:

1. Connection Establishment: Before data transmission begins, TCP establishes a connection between the sender and the receiver. This process involves a handshake mechanism known as the TCP three-way handshake, where the client and server exchange SYN (synchronize) and ACK (acknowledge) packets to establish a reliable connection.

2. Reliable Data Delivery: TCP ensures reliable data delivery by implementing various mechanisms. It employs sequencing and acknowledgment numbers to guarantee the ordered and error-free delivery of data packets. If a packet is lost or damaged during transmission, TCP retransmits it until it is successfully received by the destination.
3. Flow Control: TCP implements flow control mechanisms to manage the rate of data transmission between the sender and receiver. It uses a sliding window mechanism to regulate the amount of data that can be sent before receiving acknowledgments. This helps prevent overwhelming the receiver with more data than it can handle.
4. Congestion Control: TCP incorporates congestion control mechanisms to avoid network congestion and prevent data loss. It monitors network conditions and adjusts the transmission rate accordingly to maintain optimal network performance. TCP dynamically adapts to changing network conditions, reducing the likelihood of congestion, and ensuring fair bandwidth utilization.
5. Connection Termination: Once the data transmission is complete, TCP performs a connection termination process to gracefully close the connection. It uses a four-way handshake, involving the exchange of FIN (finish) and ACK packets, to ensure that all data is transmitted and received before terminating the connection.

These features of TCP make it a connection-oriented protocol, as it establishes a reliable connection between communicating parties, guarantees data delivery, manages flow control and congestion control, and provides a structured process for connection establishment and termination.

Why is a connection establishment for mail transfer needed if TCP has already established a connection?

In the context of mail transfer, such as with the Simple Mail Transfer Protocol (SMTP), a separate connection establishment process is needed even though TCP has already established a connection. This is because SMTP operates at a higher

layer of the network stack, specifically the application layer, and requires its own connection for the exchange of email-related information.

Here's why a separate connection establishment for mail transfer is necessary:

1. Application Protocol: SMTP is a specific application protocol used for email communication. While TCP provides the underlying transport connection, SMTP defines the rules and format for exchanging email messages. It has its own commands, such as "HELO" or "EHLO" to initiate the SMTP session and "MAIL FROM" to specify the sender's address. These SMTP commands are not part of the TCP protocol but are required for email communication.

2. Message Exchange: SMTP involves a series of message exchanges between the mail client (sending server) and the mail server (receiving server). These exchanges include commands and responses for various stages of the email transfer process, such as sender identification, recipient specification, and data transmission. These exchanges happen within the context of an SMTP session, which is separate from the TCP connection.

3. Multiple Clients and Sessions: A mail server may handle multiple clients simultaneously, each initiating its own SMTP session. By establishing separate SMTP connections, the mail server can manage and distinguish between different clients and sessions. This allows the server to handle concurrent mail transfers from multiple sources efficiently.

4. Session State and Persistence: The SMTP session maintains a stateful connection, storing information about the ongoing email transfer. This session state includes the sender, recipient(s), message content, and other relevant data. Separating the SMTP session from the underlying TCP connection allows the session state to be maintained independently and persistently, even if the TCP connection is closed or lost.

Therefore, even though TCP provides a reliable connection, a separate connection establishment process for mail transfer is necessary to establish an SMTP session, exchange email-specific commands and responses, handle multiple clients concurrently, and maintain the session state independently of the TCP connection.

Why do we need a DNS system when we can directly use an IP address?

While it is possible to directly use IP addresses to access websites and services on the internet, the Domain Name System (DNS) serves several important purposes that make it essential for efficient and user-friendly network communication. Here are some reasons why we need a DNS system:

1. Human-Readable Naming: IP addresses are numerical and not easy to remember for humans. DNS provides a hierarchical naming system that allows us to assign meaningful and memorable domain names to resources on the internet. Instead of typing a series of numbers, users can simply enter a domain name like "example.com" to access a website or service.

2. Dynamic IP Address Assignment: IP addresses can change for various reasons, such as network reconfigurations, equipment upgrades, or failover mechanisms. DNS allows organizations to assign domain names to resources while easily updating the corresponding IP addresses behind the scenes. This flexibility enables seamless transitions and avoids the need for users to manually update their configurations.

3. Load Balancing and Redundancy: DNS enables load balancing and high availability by allowing multiple IP addresses to be associated with a single domain name. This way, requests can be distributed across multiple servers or locations to optimize performance and handle increased traffic. DNS can also detect and redirect requests in case of server failures, ensuring continuity of service.

4. Centralized Management: DNS provides a centralized management system for domain name assignments and configurations. It allows organizations to control and modify their domain names and associated IP addresses from a single point of control. This simplifies administration and reduces the burden of individually managing IP addresses for each resource.

5. Scalability: DNS is designed to handle the scalability requirements of the internet. It operates in a distributed manner, with multiple DNS servers spread across the globe. This distributed architecture ensures efficient and

reliable resolution of domain names, even in the face of large-scale internet usage and global network traffic.

6. Protocol Independence: DNS is independent of the underlying protocols used for data transfer, such as TCP or UDP. It can resolve names for various services, including web servers, email servers, FTP servers, and more. DNS provides a standardized and consistent method for resolving domain names across different network protocols.

In summary, the DNS system plays a crucial role in providing human-readable naming, dynamic IP address assignment, load balancing, redundancy, centralized management, scalability, and protocol independence. It simplifies network communication, enhances user experience, and enables efficient management of resources on the internet.

A domain name is hello. customer. info. Is this a generic domain or a country domain? Why?

The domain name `hello.customer.info` is a generic top-level domain (gTLD) because it uses the `.info` top-level domain (TLD). A gTLD is a TLD that is used to categorize websites based on their content or purpose, rather than their location. Some common examples of gTLDs include `.com`, `.org`, and `.net`.
In contrast, a country code top-level domain (ccTLD) is a TLD that is used to categorize websites based on their location. Each ccTLD corresponds to a specific country or territory and is typically two letters long. Some common examples of ccTLDs include `.us` for the United States, `.uk` for the United Kingdom, and `.jp` for Japan.
In this case, the `.info` TLD is a gTLD that is intended for informational websites. It is not associated with any specific country or territory, so it is not a ccTLD.

Consider an application that transmits data at a steady rate (for example, the sender generates an N-bit unit of data every k time units, where k is small and fixed). Also, when such an application starts, it will continue running for a relatively long period of time. Answer the following questions, briefly justifying your answer:

a) Would a packet-switched network or a circuit-switched network be more appropriate for this application? Why?

b) Suppose that a packet-switched network is used and the only traffic in this network comes from such applications as described above. Furthermore, assume that the sum of the application data rates is less than the capacities of each link. Is some form of congestion control needed? Why?

a) For the described application that transmits data at a steady rate and continues running for a long period of time, a circuit-switched network would be more appropriate. In a circuit-switched network, dedicated resources are allocated for the duration of the connection, ensuring a continuous and predictable transmission rate. Since the application generates data at a steady rate, having a dedicated circuit would provide a consistent and reliable connection without the overhead of packet switching.

b) Even if the sum of the application data rates is less than the capacities of each link in a packet-switched network, some form of congestion control is still needed. Congestion control mechanisms are necessary to prevent network congestion and ensure fair allocation of network resources. In a packet-switched network, multiple applications share the same network resources, including links, routers, and buffers. Without congestion control, a burst of traffic from one or more applications could overwhelm the network and lead to performance degradation, packet loss, and increased latency.

Congestion control mechanisms, such as traffic shaping, packet prioritization, and flow control, help regulate the flow of packets, prevent congestion, and ensure equitable sharing of network resources. These mechanisms are designed to monitor and manage the utilization of network resources to maintain optimal network performance and prevent any individual application or connection from monopolizing resources to the detriment of others. Therefore, even in a scenario where the application data rates are below link capacities, congestion control is necessary to maintain a stable and efficient packet-switched network.

What is SNAT and DNAT? Identify the basic differences between firewall and DMZ.

SNAT (Source Network Address Translation) and DNAT (Destination Network Address Translation) are techniques used in networking and firewall configurations to modify the source and destination IP addresses respectively.

1. SNAT (Source Network Address Translation):
    - SNAT involves modifying the source IP address of outgoing packets to a different IP address.
    - It is commonly used in network address translation (NAT) scenarios where multiple devices within a private network share a single public IP address.
    - SNAT allows the devices to communicate with external networks by replacing their private IP addresses with the public IP address during outbound traffic.
    - This translation allows the devices to establish connections with external servers and receive responses back, as the public IP address is recognizable and routable on the internet.
2. DNAT (Destination Network Address Translation):
    - DNAT involves modifying the destination IP address of incoming packets to a different IP address.

- It is used to redirect incoming traffic from a public IP address to a specific private IP address or server within a private network.
- DNAT is often employed in scenarios such as port forwarding or load balancing, where incoming requests to a public IP address need to be redirected to a particular internal server.
- By modifying the destination IP address, DNAT ensures that the incoming traffic is directed to the correct internal destination.

Now, let's discuss the basic differences between a firewall and a DMZ (Demilitarized Zone):

1. Firewall:
    - A firewall is a security device or software that monitors and controls network traffic between different network segments or between a network and the internet.
    - It acts as a barrier or gatekeeper, enforcing security policies by allowing or blocking specific types of traffic based on predefined rules.
    - A firewall filters traffic based on factors like source and destination IP addresses, port numbers, protocols, and packet contents.
    - It helps protect the internal network from unauthorized access, external threats, and potential attacks by inspecting and regulating traffic flow.
2. DMZ (Demilitarized Zone):
    - A DMZ is a separate network segment that sits between an internal network and an external network, typically the internet.
    - The DMZ is a semi-trusted zone that houses servers or services that need to be accessible from the internet, such as web servers, email servers, or publicly available resources.
    - By placing these servers in the DMZ, organizations can isolate them from the internal network, reducing the risk of direct access to sensitive data or critical infrastructure.

- The DMZ is usually protected by firewalls, with specific rules and configurations to allow limited access to the servers within it.
- The purpose of the DMZ is to provide an additional layer of security by segregating publicly accessible services from the internal network.

In summary, a firewall is a security device that controls and monitors network traffic, while a DMZ is a separate network segment that houses publicly accessible servers or services. Firewalls enforce security policies, whereas the DMZ provides an isolated zone for hosting external-facing resources.

Firewall	DMZ
A network security system that controls traffic between networks [1]	A network security technique used to protect an organization's internal network from external threats [2]
Protects the entire network [3]	Primarily used to protect specific servers, applications, and other resources on the network [3]
Placed between an organization's internal network and the internet [1]	A physical or logical subnetwork that contains and exposes an organization's external-facing services to an untrusted network, usually a larger network such as the Internet [2]
Can be a device or software [2]	Isolated by a security gateway, such as a firewall, that filters traffic between the DMZ and a LAN [4]
Monitors and controls incoming and outgoing network traffic based on predetermined security rules [2]	Ideally located between two firewalls, ensuring incoming network packets are observed by a firewall before they make it through to the servers hosted in the DMZ [4]

What do you mean by Framing? Explain byte stuffing? A bit stuffing framing protocol uses an 8-bit delimiter pattern 01111110. If the output bit string after stuffing is 01111100101, find the input bit string.

Framing is a process in data communication where a stream of bits is divided into discrete frames, allowing the receiver to identify the boundaries of each frame and extract the transmitted data correctly.

Byte stuffing is a specific technique used in framing protocols to ensure the integrity of data and distinguish frame boundaries. It involves adding special control characters (delimiters) to the data stream to indicate the start and end of a frame.

In the case of bit stuffing, an 8-bit delimiter pattern, such as 01111110, is used. Whenever the delimiter pattern is encountered within the data stream, a specific stuffing rule is applied to ensure it does not confuse the receiver. The stuffing rule states that if five consecutive 1 bits are encountered, a 0 bit is inserted after them to maintain the integrity of the data and prevent the receiver from mistakenly identifying the delimiter.

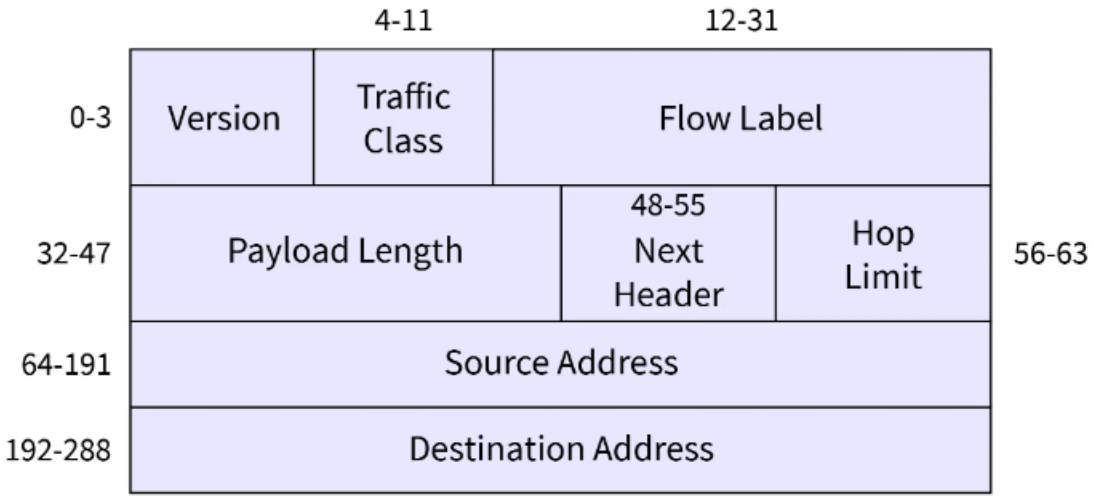
Now, let's analyse the given output bit string after stuffing: 01111100101.

To determine the original input bit string, we need to reverse the process of bit stuffing:

1. Remove the stuffed bits (0s) that follow five consecutive 1s within the output bit string.
    - Original input after removing stuffed bits: 0111111001
2. Remove the delimiter pattern (01111110) from the input bit string to obtain the final input bit string.
    - Final input bit string: 0111

Therefore, the original input bit string, before the bit stuffing process, is 0111.

Draw packet format of IPv6. Give importance of each field and highlight the merits of IPv6 over IPv4



Version (4-bits) :

It shows the version of internet protocol we used, i.e. 0110

Traffic Class (8-bits) :

This is an 8-bit field in which 8 bits are divided into **two** parts. The most significant 6-bit is for the type of service so that the router will get to know about what services need to be provided to the given packet. And for **Explicit Congestion Notification** (ECN), the least significant 2-bit is used.

Flow Label (20-bits) :

This 20-bit is required for maintaining the sequential flow of packets related to a particular communication. This field is also helpful in avoiding the reordering of packets. The source labels the sequence to help the router so that it can identify that a particular packet is related to a specific flow of data. It is generally used for real or streaming media.

Payload Length (16-bits) :

This field is used to help the router in knowing how much information is stored in the payload of a particular packet.

Next Header (8-bits):

This field is used to represent the type of extension header or if the extension header is not present then it shows the Upper Layer **PDU**. The value for Upper Layer PDU is the same as that of values in IPv4.

Hop Limit (8-bits) :

Hop limit is a field in a header that stops the header to go into an infinite loop in the network. It works the same as that of **TTL** in IPv4. When it passes a hop or router its value is **decremented** by 1. The packet is discarded when it reaches 0.

Source Address (128-bits) :

This field provides the address from where the packet originates.

Destination Address (128-bits) :

The destination address is the address of the packet's intended recipient.

6 advantages of IPv6 to IPv4

1. Simpler header format

IPv4 offers 12 header fields whereas IPv6 offers 8 header fields. The introduction of extension headers makes it possible to implement optional information into IPv6 packages much more effectively than with IPv4. Because routers on the delivery path of a package don't process IPv6 extension headers, with IPv6, these are only read at the destination, which means a significant improvement of router performance.

2. More efficient routing

IPv6 reduces the size of routing tables, which makes routing more hierarchical and therefore more efficient. IPv6 also lets ISPs combine the prefixes of their customers' networks into a single prefix. Additionally, with IPv6, the source device, rather than the router,

handles fragmentation, using a protocol for discovering the path's Maximum Transition Unit (MTU).

3. More security

Because IPv6 uses 128-bit addresses, the pool of possible addresses is 340 undecillions (3.4×10^{38}). IPv4's 32-bit addresses allow for only 4.3 billion addresses. A bigger pool of addresses obviously means more scalability, but, less obviously, it also means increased security. That's because under the IPv6 system, host scanning and identification is more difficult for hackers.

When IPv4 was first developed, internet security wasn't as much of an issue as it is today. This explains why the few security protocols available for IPv4 appear to have been developed as an afterthought. But IPv6 was built with security in mind. Many of the IPv4's optional security requirements have been baked into IPv6 as default requirements. For instance, IPv6 automatically encrypts traffic and checks packet integrity. This gives standard internet traffic VPN-level protection.

4. True Quality of Service

Quality of Service (QoS) is a tool that lets you train your router to allocate specific portions of your available bandwidth to different applications. Good Quality of Service means, for example, your computer won't struggle to play a video because it happens to be trying to download a massive file at the same time.

Quality of Service *technically* exists in IPv4, but it doesn't actually work. Packets can technically be assigned different priorities, but routers usually just ignore the QoS flag, and some even mark *all* packets "Highest priority", which, as you might imagine, defeats the purpose.

But IPv6 has an integrated mechanism for securing Quality of Service. It prioritises urgent packages, which makes handling those packages more efficient. It even has specialised fields ("traffic class" and "flow label") that are directly responsible for ensuring Quality of Service.

5. Easier file-sharing

With IPv6, there are two separate address spaces for private addressing. These are called "link-local" and "site-local". A link-local address has lots of useful functions, including hosting auto configuration by simply querying the router (no DHCP necessary!) and setting up ad-hoc LANs *without a router*. This means you can connect PCs and share files without having to bother with file-sharing protocols.

6. No more NAT (Network Address Translation)

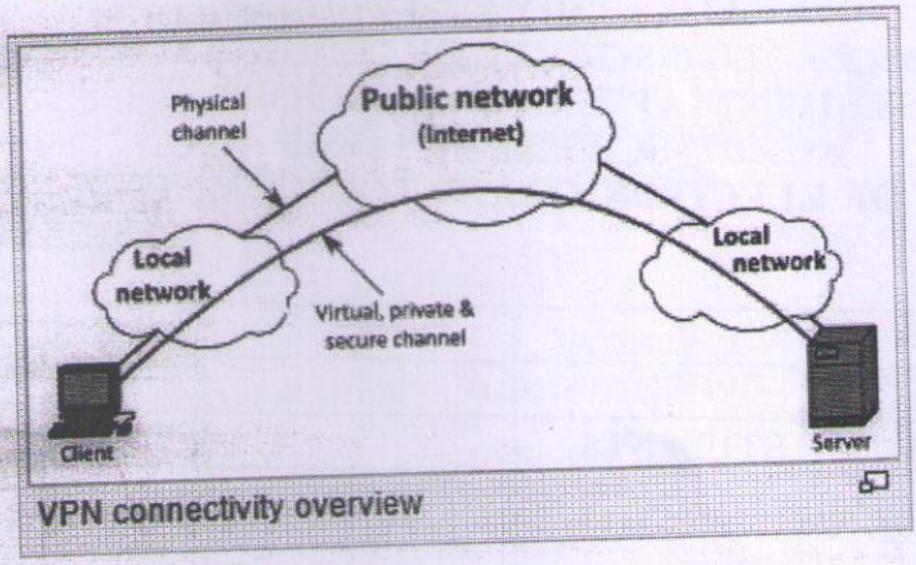
Because there aren't enough IPv4 addresses, much of the internet relies on NAT for connectivity. NAT was great for extending the life of IPv4, but it also forces every packet that enters or leaves your network to be examined and altered. And services that use multiple ports have a particularly hard time, since they need to undergo all sorts of cumbersome adjustments to work. With IPv6, every device

can have its own unique IP address, which eliminates the need for NAT!

Give the concept of vpn with neat diagram.

A **virtual private network** (**VPN**) extends a private network across a public network, and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network. Applications running on a computing device, e.g., a laptop, desktop, smartphone, across a VPN may therefore benefit from the functionality, security, and management of the private network. Encryption is a common, though not an inherent, part of a VPN connection. VPN technology was developed to allow remote users and branch offices to access corporate applications and resources. To ensure security, the private network connection is established using an encrypted layered tunneling protocol, and VPN users use authentication methods, including passwords or certificates, to gain access to the VPN. In other applications, Internet users may secure their connections with a VPN to circumvent geo-restrictions and censorship or to connect to proxy servers to protect personal identity and location to stay anonymous on the Internet. Some websites, however, block access to known VPN technology to prevent the circumvention of their geo-restrictions, and many VPN providers have been developing strategies to get around these roadblocks.A VPN is created by establishing a virtual point-to-point connection through the use of

dedicated circuits or with tunneling protocols over existing networks. A VPN available from the public Internet can provide some of the benefits of a wide area network (WAN). From a user perspective, the resources available within the private network can be accessed remotely.



VPN connectivity overview

Link Layer

https://questions.examside.com/past-years/gate/gate-cse/computer-networks/data-link-layer-and-switching

https://gateoverflow.in/tag/data-link-layer

https://byjus.com/gate/gate-computer-networks-questions/

https://www.computersciencejunction.in/2019/07/31/computer-networks-gate-questions-html/

Algorithms

https://questions.examside.com/past-years/gate/gate-cse/computer-networks/routing-algorithm

https://gateoverflow.in/tag/routing

https://byjus.com/gate/computer-network-gate-questions/

Error detection and correction techniques

https://www.gatevidyalay.com/tag/error-detection-and-correction/

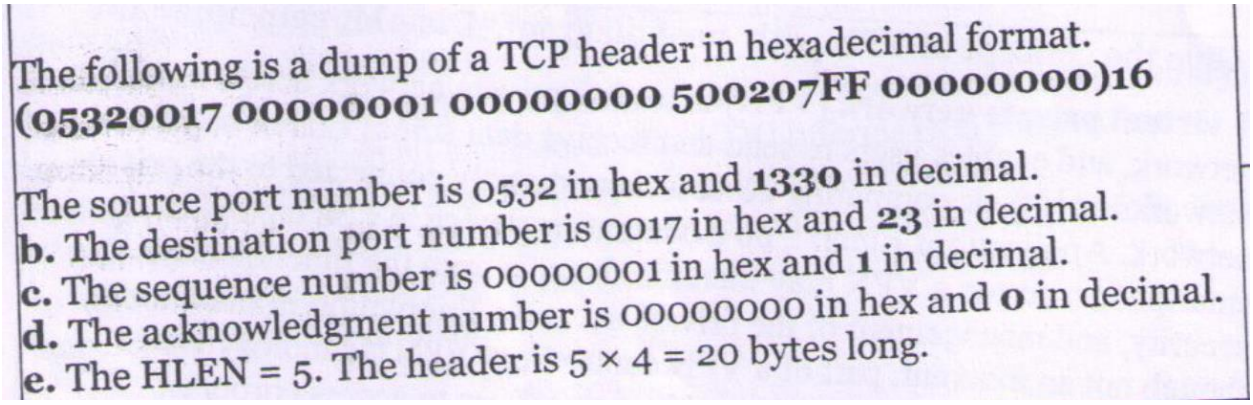
https://gateoverflow.in/tag/error-detection?start=30

Selective repeat

https://www.gatevidyalay.com/selective-repeat-protocol-practice-problems/

https://gateoverflow.in/tag/selective-repeat

https://gateoverflow.in/tag/go-back-n



The following is a dump of a TCP header in hexadecimal format.
(05320017 00000001 00000000 500207FF 00000000)16

The source port number is 0532 in hex and 1330 in decimal.
b. The destination port number is 0017 in hex and 23 in decimal.
c. The sequence number is 00000001 in hex and 1 in decimal.
d. The acknowledgment number is 00000000 in hex and 0 in decimal.
e. The HLEN = 5. The header is 5 × 4 = 20 bytes long.

The following is a dump of a UDP header in hexadecimal format.

0632000D001CE217

(i) What is the source port number?

(ii) What is the destination port number?

(iii) What is the total length of the user datagram?

(iv) What is the length of the data?

(v) Is the packet directed from a client to a server or vice versa?

(vi) What is the client process?

Ans:

The given hexadecimal dump of a UDP header can be interpreted as follows:

0632: The source port number is represented by the first 4 hexadecimal digits. Converting `0632` from hexadecimal to decimal gives `1586`. So, the source port number is **1586**.

000D: The destination port number is represented by the second set of 4 hexadecimal digits. Converting `000D` from hexadecimal to decimal gives `13`. So, the destination port number is **13**.

001C: The total length of the user datagram is represented by the third set of 4 hexadecimal digits. Converting `001C` from hexadecimal to decimal gives `28`. So, the total length of the user datagram is **28 bytes**.

The length of the data can be calculated by subtracting the length of the UDP header (which is always `8` bytes) from the total length of the user datagram. In this case, the length of the data is `28 - 8 = 20 bytes`.

Since the destination port number is `13`, which is a well-known port number for the Daytime Protocol, it can be inferred that this packet is directed from a client to a server.

The client process can be identified by its source port number, which in this case is `1586`. However, without additional information about the processes running on the client machine, it is not possible to determine which specific process is associated with this port number.

Station A needs to send a message consisting of 9 packets to station B using a sliding window (window size 3) and go back-n error control strategy. All packets are ready and immediately available for transmission. If every 5th packet that A transmits gets lost (but no acks from B ever get lost) then what is the number of packets that A will transmit for sending the message to B?

In a sliding window protocol with a window size of `3` and a go-back-n error control strategy, station A can send up to `3` packets at a time before waiting for an acknowledgment from station B. If every `5th` packet that A transmits gets lost, then A will need to retransmit that packet and all subsequent packets in the window until it receives an acknowledgment from B.

In a sliding window protocol with a window size of `3` and a go-back-n error control strategy, station A can send up to `3` packets at a time before waiting for an acknowledgment from station B. If every `5th` packet that A transmits gets lost, then A will need to retransmit that packet and all subsequent packets in the window until it receives an acknowledgment from B.

Here is a step-by-step breakdown of the transmission process:

1. A sends packets `1`, `2`, and `3` to B.
2. B receives packets `1`, `2`, and `3` and sends an acknowledgment for packet `3`.
3. A receives the acknowledgment for packet `3` and sends packets `4`, `5`, and `6` to B.
4. Packet `5` gets lost in transmission. B receives packets `4` and `6` but does not send an acknowledgment because it is still waiting for packet `5`.
5. A times out waiting for an acknowledgment and retransmits packets `5`, `6`, and `7`.
6. B receives packets `5`, `6`, and `7` and sends an acknowledgment for packet `7`.
7. A receives the acknowledgment for packet `7` and sends packets `8` and `9` to B.
8. B receives packets `8` and `9` and sends an acknowledgment for packet `9`.

In total, station A will transmit 11 packets (`9` original packets + 2 retransmissions) to send the message to station B.

A 2 km long broadcast LAN has 10 bps bandwidth and uses CSMA/CD. The signal travels along the wire at 2 × 108 m/s. What is the minimum packet size that can be used on this network?

In a CSMA/CD (Carrier Sense Multiple Access with Collision Detection) network, the minimum packet size is determined by the round-trip time it takes for a signal to travel from one end of the network to the other and back. This time is known as the slot time and is calculated as follows:

Slot time = 2 * (Length of LAN) / (Signal propagation speed) = 2 * (2000 m) / (2 × 10^8 m/s) = 20 μs

The minimum packet size is then calculated by multiplying the slot time by the data rate of the network:

Minimum packet size = Slot time * Data rate = 20 μs * 10 bps = 200 bits

So, the minimum packet size that can be used on this network is **200 bits**.

Transmission time 1ms, propagation delay 1 ms, bandwidth 4Mbps. Calculate efficiency. If n stations are connected to TDM network and each station needs 2Kbps, how many station can be connected to this network?

The efficiency of a communication system can be calculated using the formula:

Efficiency = Transmission time / (Transmission time + 2 * Propagation delay)
In this case, the transmission time is `1 ms`, the propagation delay is `1 ms`, and the bandwidth is `4 Mbps`. Plugging these values into the formula above, we get:

Efficiency = 1 ms / (1 ms + 2 * 1 ms) = 1/3

So, the efficiency of this communication system is **1/3** or approximately **33.33%**.
In a Time Division Multiplexing (TDM) network, the total bandwidth of the network is divided among the connected stations. If each station needs `2 Kbps` and the total bandwidth of the network is `4 Mbps`, then we can calculate the maximum number of stations that can be connected to the network as follows:

Maximum number of stations = Total bandwidth / Bandwidth per station = (4 Mbps) / (2 Kbps) = 4000 / 2 = 2000

So, a maximum of **2000** stations can be connected to this TDM network.

What is the total delay (latency) for a frame of size 5 million bits

that is being sent on a link with 10 routers each having a queuing

time of 2 microseconds and a processing time of 1 Microseconds.

The length of the link is 2000 Km. The speed of signal inside the

link is 2 x 108

m/s. The link has a bandwidth of 5 Mbps?

i. Identify individual components of delay.

ii. Which component of the total delay is dominant?

iii. Which component of delay is negligible?

The total delay (latency) for a frame being sent on a link with multiple routers consists of several components: transmission delay, propagation delay, processing delay, and queuing delay.

(i) The individual components of delay can be calculated as follows:

- Transmission delay: This is the time it takes to transmit the frame onto the link. It is calculated by dividing the size of the frame by the bandwidth of the link. In this case, the transmission delay is `5 million bits / 5 Mbps = 1 second`.
- Propagation delay: This is the time it takes for a signal to travel from one end of the link to the other. It is calculated by dividing the length of the link by the speed of signal propagation. In this case, the propagation delay is `2000 km / (2 x 10^8 m/s) = 0.01 seconds`.
- Processing delay: This is the time it takes for each router to process the frame. In this case, there are `10` routers and each router has a processing time of `1 microsecond`, so the total processing delay is `10 * 1 microsecond = 0.00001 seconds`.
- Queuing delay: This is the time that the frame spends waiting in router queues. In this case, there are `10` routers and each router has a queuing time of `2 microseconds`, so the total queuing delay is `10 * 2 microseconds = 0.00002 seconds`.
- (ii) The dominant component of the total delay in this case is the transmission delay, which is `1 second`. This is much larger than any of the other components of delay.
- (iii) The negligible component of the total delay in this case is the processing delay, which is only `0.00001 seconds`. This is much smaller than any of the other components of delay.
- So, the total delay (latency) for a frame of size `5 million bits` that is being sent on a link with `10` routers each having a queuing time of `2 microseconds` and a processing time of `1 microsecond`, with a length of `2000 km`, a signal speed of `2 x 10^8 m/s`, and a bandwidth of `5 Mbps` would be approximately `1 + 0.01 + 0.00001 + 0.00002 = 1.01003 seconds`.

The Company is provided with classful addresses starting from 131.120.0.0. And it wants to create autonomous systems for each department. Human Resource is the smallest department which requires 32 hosts followed by the Purchase department which wants to configure around 62 hosts. Sales require around 128 IP addresses in its department. Marketing department needs to have 128 connected devices. The Finance department will configure 96 hosts, while the Operation department will need 256 IP addresses in its network. Provide IP range and subnet mask for each department?

Rules:

i. Company assigns Initial IP addresses first keeping IP addresses at the end reserved.

ii. The IP blocks which are allocated are continuous.(No gaps in ranges).

iii. Make sure you assign blocks in such a way that minimum IP addresses are wasted.

iv. So that unused IP addresses can bring maximum profit to ISP in future.

The company has been provided with classful addresses starting from `131.120.0.0`. This is a Class B address range, which means that the first two octets (`131.120`) are fixed and the last two octets (`0.0`) can be used for subnetting and assigning host addresses.

To create autonomous systems for each department while minimizing the number of wasted IP addresses, we can use variable-length subnet masking (VLSM) to assign subnets of different sizes to each department based on their needs.

Here is one possible way to assign IP ranges and subnet masks to each department:

- Human Resource: This department requires `32` hosts. The smallest subnet that can accommodate `32` hosts is a `/27` subnet, which provides `30` usable host addresses (`2^5 - 2 = 30`). We can assign the first `/27` subnet in the address range to the Human Resource department: `131.120.0.0/27`. The IP range for this department would be `131.120.0.1` to `131.120.0.30`, with a subnet mask of `255.255.255.224`.
- Purchase: This department requires `62` hosts. The smallest subnet that can accommodate `62` hosts is a `/26` subnet, which provides `62` usable host addresses (`2^6 - 2 = 62`). We can assign the next `/26` subnet in the address range to the Purchase department: `131.120.0.32/26`. The IP range for this department would be `131.120.0.33` to `131.120.0.94`, with a subnet mask of `255.255.255.192`.
- Sales: This department requires `128` hosts. The smallest subnet that can accommodate `128` hosts is a `/25` subnet, which provides `126` usable host addresses (`2^7 - 2 = 126`). We can assign the next `/25` subnet in the address range to the Sales department: `131.120.0.128/25`. The IP range for this department would be `131.120.0.129` to `131.120.0.254`, with a subnet mask of `255.255.255.128`.
- Marketing: This department requires `128` hosts, so we can assign it the same size subnet as the Sales department: a `/25` subnet with `126` usable host addresses (`2^7 - 2 = 126`). We can assign the next `/25` subnet in the address range to the Marketing department: `131.120.1.<IPAddress>/25`. The IP range for this department would be `131.<IPAddress>.1.<IPAddress>` to `<IPAddress>.<IPAddress>.1.<IPAddress>`, with a subnet mask of `<IPAddress>.<IPAddress>.<IPAddress>.128`

- Finance: This department requires `<IPAddress>` hosts, so we can assign it a `/25` subnet with `<IPAddress>` usable host addresses (`2^7 - 2 = <IPAddress>`). We can assign the next `/25` subnet in the address range to the Finance department: `<IPAddress>.<IPAddress>.1.<IPAddress>/25`. The IP range for this department would be `<IPAddress>.<IPAddress>.1.<IPAddress>` to `<IPAddress>.<IPAddress>.1.<IPAddress>`, with a subnet mask of `<IPAddress>.<IPAddress>.<IPAddress>.128`.
- Operation: This department requires `<IPAddress>` hosts, so we can assign it a `/24` subnet with `<IPAddress>` usable host addresses (`2^8 - 2 = <IPAddress>`). We can assign the next `/24` subnet in the address range to the Operation department: `<IPAddress>.<IPAddress>.2.<IPAddress>/24`. The IP range for this department would be `<IPAddress>.<IPAddress>.2.<IPAddress>` to `<IPAddress>.<IPAddress>.2.<IPAddress>`, with a subnet mask of `<IPAddress>.<IPAddress>.<IPAddress>.0`.

A network with one primary and two secondary stations uses polling. The secondary station 1 wants to send data to secondary station 2. The size of data frame is 1000 bytes and Station 1 has 5 such frames which he would like to send. The Poll, Select, ACK and NAK are of 16 bytes. Primary Station polls secondary station 1 first. How many total bytes are exchanged if there is no limitation on the number of frames a station can send in response to a poll? (Draw communication diagram and show calculation.)

In a network with one primary and two secondary stations that uses polling, the primary station controls the communication by polling each secondary station to see if it has data to send. In this case, the primary station polls secondary station 1 first. Since secondary station 1 has data to send, it will respond to the poll with its data frames.

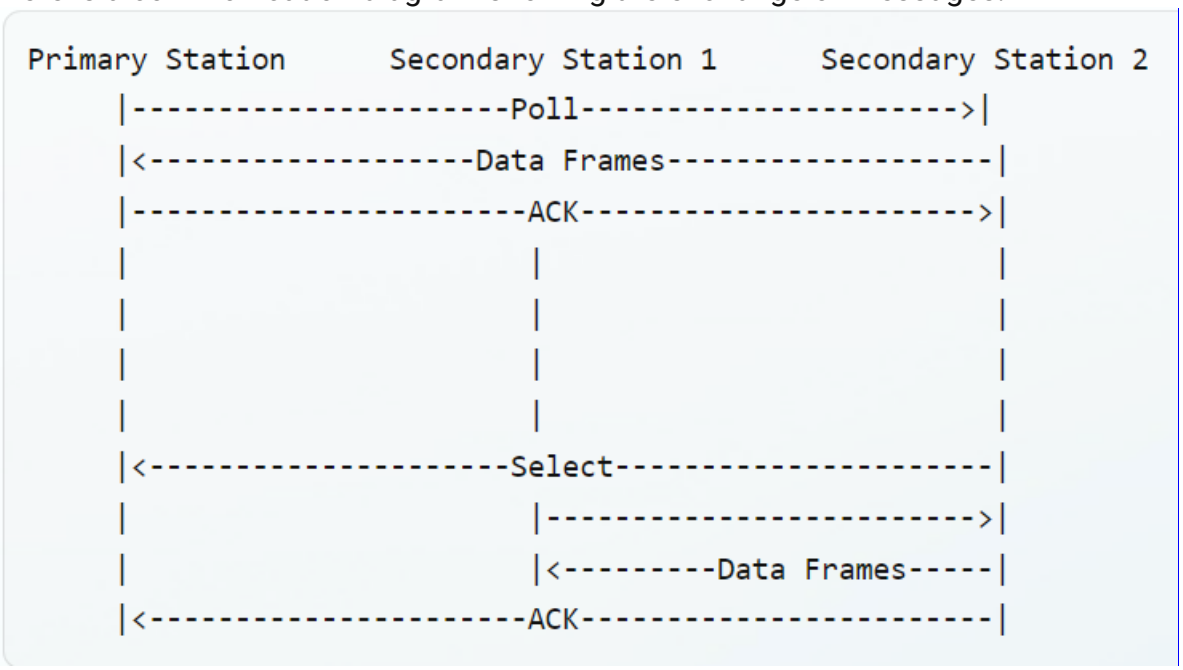
The total number of bytes exchanged can be calculated as follows:

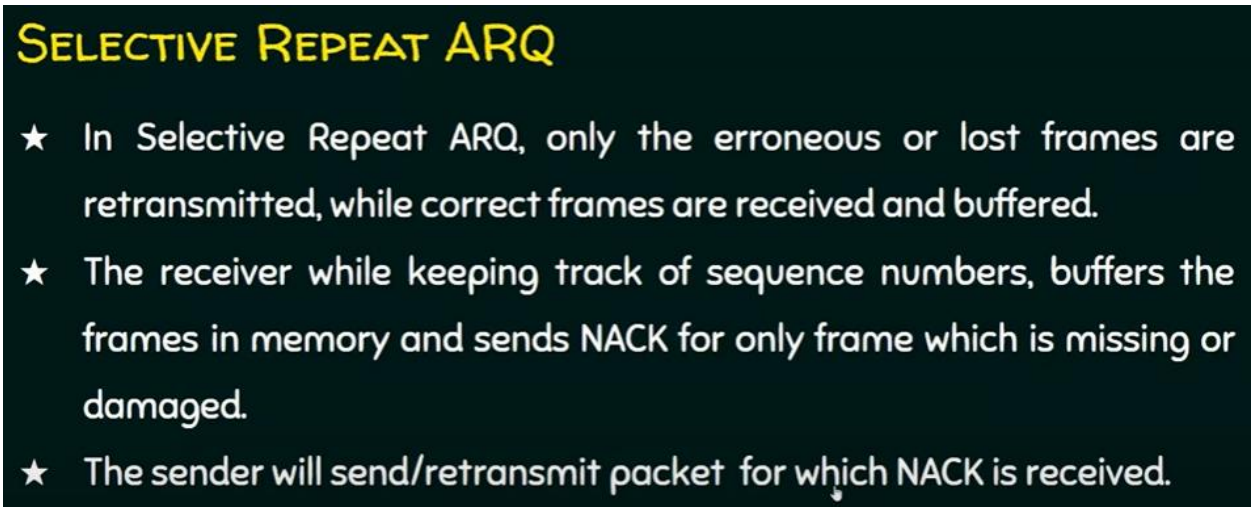
- The primary station sends a poll to secondary station 1: `16 bytes`
- Secondary station 1 responds with its `5` data frames: `5 * 1000 bytes = 5000 bytes`
- The primary station sends an ACK to secondary station 1: `16 bytes`
- The primary station sends a Select to secondary station 2: `16 bytes`

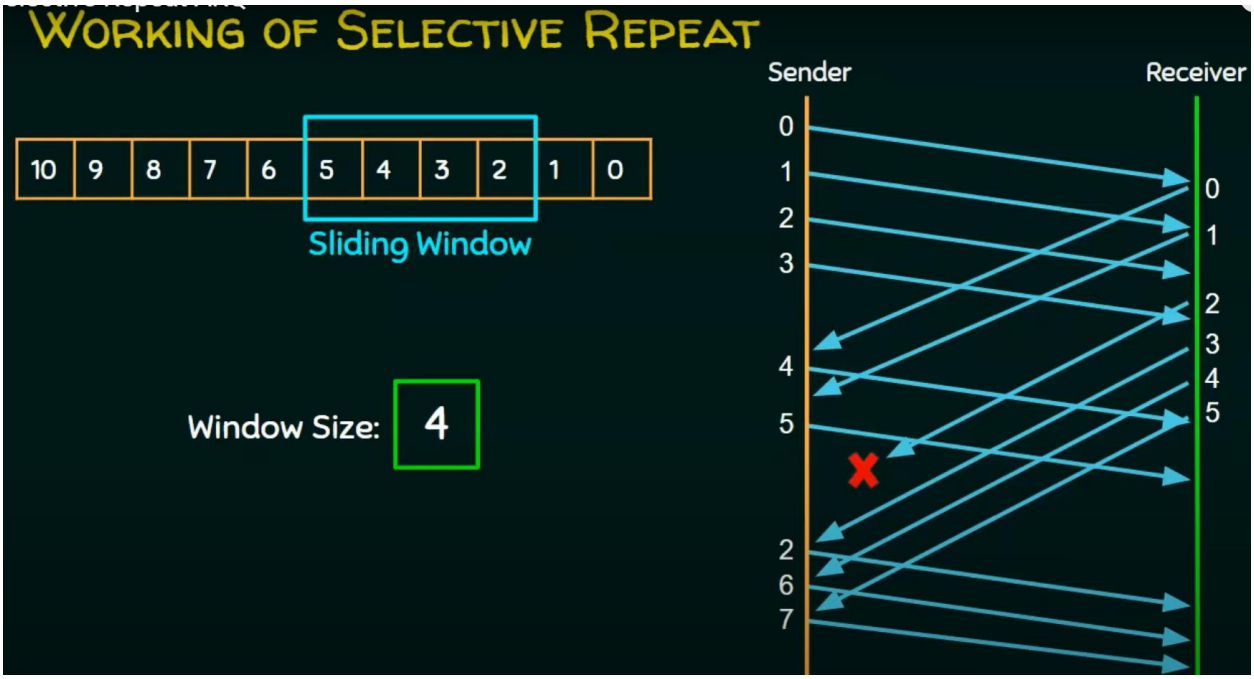
- Secondary station 2 receives the data frames from secondary station 1: `5000 bytes`
- The primary station sends an ACK to secondary station 2: `16 bytes`

Total bytes exchanged = `16 + 5000 + 16 + 16 + 5000 + 16 = 10064 bytes`

Here is a communication diagram showing the exchange of messages:






WORKING OF SELECTIVE REPEAT
10 9 8 7 6 5 4 3 2 1 0
Sliding Window
Window Size: 4
Sender
Receiver
0
1
2
3
4
5
2
6
7
0
1
2
3
4
5