QUESTION BANK

Subject: COMPUTER NETWORKS

Class: TY AIML A & B

SR NO	QUESTION	COs	BL	POs
1	Explain Design issues in data link Layer?	3	2	1
2	Explain Elementary data link layer protocols?	3	2	2
3	Explain Simplex stop and wait protocol?	3	2	1
4	Explain sliding window protocol with i) one bit protocol ii) Go back ARQ iii) Selective Repeat ARQ	3	2	2
5	Explain static and dynamic Channel allocation?	3	2	1
6	Explain ALOHA, Pure ALOHA and slotted ALOHA in detail ?	3	2	1
7	Describe in detail CSMA Protocol ?	3	2	2
8	Describe in detail CSMA/CA and CSMA/CD Protocol With Protocol?	3	2	2
9	Explain IEEE 802.11 in detail?	3	2	1
10	Differentiate between Token Ring and Token Bus?	3	4	1
11	Explain Design issues of Network Layer?	4	2	1
12	Explain comparison between virtual circuit and datagram subnets?	4	2	3
13	Explain shortest path routing algorithm?	4	2	3
14	Explain Distance Vector Routing Algorithm?	4	2	3
15	Explain OSPF in detail?	4	2	1
16	Explain BGP in detail?	4	2	1
17	Explain IP Header format in detail?	4	2	1
18	Illustrate with the diagram the five address format in IP?	4	3	1
19	Differentiate IPv4 and IPv6?	4	4	1
20	Differentiate Broadcast and Multicast Routing?	4	4	2

1.

1. Services provided to the network layer –
The data link layer act as a service interface to the network layer. The principle service is transferring data from network layer on sending machine to the network layer on destination machine. This transfer also takes place via DLL (Data link-layer).

It provides three types of services:

1. Unacknowlwdged and connectionless services.
2. Acknowledged and connectionless services.
3. Acknowledged and connection-oriented services

Unacknowledged and connectionless services.

- Here the sender machine sends the independent frames without any acknowledgement from the sender.
- There is no logical connection established.

Acknowledged and connectionless services.

- There is no logical connection between sender and receiver established.
- Each frame is acknowledged by the receiver.
- If the frame didn't reach the receiver in a specific time interval it has to be sent again.
- It is very useful in wireless systems.

Acknowledged and connection-oriented services

- A logical connection is established between sender and receiver before data is trimester.
- Each frame is numbered so the receiver can ensure all frames have arrived and exactly once.

2. Frame synchronization –
The source machine sends data in the form of blocks called frames to the destination machine. The starting and ending of each frame should be identified so that the frame can be recognized by the destination machine.

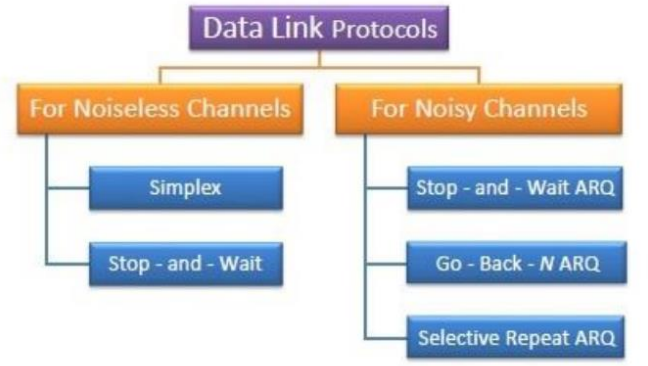
3. Flow control –
Flow control is done to prevent the flow of data frame at the receiver end. The source machine must not send data frames at a rate faster than the capacity of destination machine to accept them.

4. Error control –
Error control is done to prevent duplication of frames. The errors introduced during transmission from source to destination machines must be detected and corrected at the destination machine

2.

Elementary Data Link Protocols Protocols in the data link layer are designed so that this layer can perform its basic functions: framing, error control and flow control. Framing is the process of dividing bit - streams from physical layer into data frames whose size ranges from a few hundred to a few thousand bytes. Error control mechanisms deals with transmission errors and retransmission of corrupted and lost frames. Flow control regulates speed of delivery and so that a fast sender does not drown a slow receiver.

Types of Data Link Protocols Data link protocols can be broadly divided into two categories, depending on whether the transmission channel is noiseless or noisy.



Simplex Protocol

The Simplex protocol is hypothetical protocol designed for unidirectional data transmission over an ideal channel, i.e. a channel through which transmission can never go wrong. It has distinct procedures for sender and receiver. The sender simply sends all its data available onto the channel as soon as they are available its buffer. The receiver is assumed to process all incoming data instantly. It is hypothetical since it does not handle flow control or error control.

Stop – and – Wait Protocol

Stop – and – Wait protocol is for noiseless channel too. It provides unidirectional data transmission without any error control facilities. However, it provides for flow control so that a fast sender does not drown a slow receiver. The receiver has a finite buffer size with finite processing speed. The sender can send a frame only when it has received indication from the receiver that it is available for further data processing.

Stop – and – Wait ARQ

Stop – and – wait Automatic Repeat Request (Stop – and – Wait ARQ) is a variation of the above protocol with added error control mechanisms, appropriate for noisy channels. The sender keeps a copy of the sent frame. It then waits for a finite time to receive a positive acknowledgement from receiver. If the timer expires or a negative acknowledgement is received, the frame is retransmitted. If a positive acknowledgement is received then the next frame is sent.

Go – Back – N ARQ

Go – Back – N ARQ provides for sending multiple frames before receiving the acknowledgement for the first frame. It uses the concept of sliding window, and so is also called sliding window protocol. The frames are sequentially numbered and a finite number of frames are sent. If the acknowledgement of a frame is not received within the time period, all frames starting from that frame are retransmitted.

Selective Repeat ARQ

This protocol also provides for sending multiple frames before receiving the acknowledgement for the first frame. However, here only the erroneous or lost frames are retransmitted, while the good frames are received and buffered.

3.

Simplex Protocol

The Simplex protocol is hypothetical protocol designed for unidirectional data transmission over an ideal channel, i.e. a channel through which transmission can never go wrong. It has distinct procedures for sender and receiver. The sender simply sends all its data available onto the channel as soon as they are available its buffer. The receiver is assumed to process all incoming data instantly. It is hypothetical since it does not handle flow control or error control.

Stop – and – Wait Protocol

Stop – and – Wait protocol is for noiseless channel too. It provides unidirectional data transmission without any error control facilities. However, it provides for flow control so that a fast sender does not drown a slow receiver. The receiver has a finite buffer size with finite processing speed. The sender can send a frame only when it has received indication from the receiver that it is available for further data processing.

4.

The **Sliding Window Protocol** is a flow control method used at the Data Link layer to efficiently manage the transmission of data frames between a sender and a receiver. The "window" refers to a range of acceptable sequence numbers for frames that can be sent (at the sender) or received (at the receiver) without needing an acknowledgment for every single frame.

1. Sliding Window with One-Bit Protocol (Stop-and-Wait)

In the **One-Bit Sliding Window Protocol** (also known as **Stop-and-Wait Protocol**), the window size is fixed at **1**. This is the simplest form of sliding window, where only one frame can be sent at a time. The sender must wait for an acknowledgment (ACK) for each frame before sending the next one.

Key Points:

- **Sender's Window**: Only 1 frame can be "in-flight" (unacknowledged).
- **Receiver's Window**: The receiver expects the next frame in sequence and sends an ACK for every frame received.
- **Behavior**:
    1. The sender sends one frame and waits.
    2. If the receiver receives the frame correctly, it sends an acknowledgment (ACK).
    3. The sender, after receiving the ACK, can then send the next frame.
    4. If the sender doesn't receive an ACK within a timeout period, it retransmits the frame.

Pros:

- Simple to implement.
- Reliable for small networks or networks with low latency.

Cons:

- **Inefficiency**: Since the sender must wait for the ACK after sending every frame, it results in idle time, making the protocol inefficient for networks with high latency or bandwidth.

---

2. Sliding Window with Go-Back-N ARQ

In **Go-Back-N ARQ**, the sliding window allows the sender to transmit up to **N frames** before waiting for an acknowledgment. If an error occurs (or an ACK is missing), the sender **goes back** and retransmits the erroneous frame and all subsequent frames, even if they were received correctly.

Key Points:

- **Sender's Window**: Can send multiple frames (up to N frames) without waiting for individual ACKs.
- **Receiver's Window**: Can only receive frames in order. If a frame is out of order or lost, it discards that frame and all subsequent frames.

- **Behavior**:
    1. The sender sends multiple frames sequentially without waiting for an ACK.
    2. The receiver checks each frame. If correct, it sends an ACK with the sequence number of the next expected frame.
    3. If the receiver detects an error or a lost frame, it discards that frame and any subsequent frames and sends an ACK with the sequence number of the last correctly received frame.
    4. The sender, upon receiving a NACK or a timeout, retransmits the erroneous frame and all subsequent frames.

Example:

- Sender sends frames 1, 2, 3, 4, 5.
- If frame 3 is lost, the receiver discards frames 3, 4, and 5.
- The sender, upon timeout or receiving a NACK for frame 3, retransmits frames 3, 4, and 5.

Pros:

- More efficient than Stop-and-Wait since multiple frames can be sent before waiting for an ACK.
- Simpler than Selective Repeat.

Cons:

- **Inefficiency** in retransmission: If a single frame is lost or corrupted, Go-Back-N requires retransmission of all subsequent frames, even if they were received correctly.

---

3. Sliding Window with Selective Repeat ARQ

Selective Repeat ARQ improves on Go-Back-N by allowing the receiver to accept frames even if they arrive out of order. The receiver can buffer out-of-order frames and send ACKs individually for each frame received. Only the **lost or corrupted frames** are retransmitted, making it more efficient.

Key Points:

- **Sender's Window**: Can send multiple frames (up to N frames) without waiting for individual ACKs.
- **Receiver's Window**: Can receive frames out of order and buffer them until the missing frames are retransmitted.
- **Behavior**:
    1. The sender sends multiple frames sequentially without waiting for an ACK.
    2. The receiver acknowledges each frame individually.
    3. If a frame is lost or corrupted, the receiver sends a NACK or no acknowledgment at all for that specific frame.

4. The sender only retransmits the specific frame that was lost or corrupted.
5. Once the missing frame is received, the receiver can assemble the frames in the correct order.

Example:

- Sender sends frames 1, 2, 3, 4, 5.
- Frame 3 is lost, but the receiver accepts and buffers frames 4 and 5.
- The sender retransmits only frame 3 when it detects the error or gets a NACK.
- Once frame 3 is received, the receiver reassembles the frames in order.

Pros:

- **Efficiency**: Only the erroneous frame is retransmitted, reducing redundant retransmissions.
- Suitable for networks with high error rates and high latency.

Cons:

- **Complexity**: The protocol requires more complex logic for both the sender and receiver, especially for managing the buffers and handling out-of-order frames.

5.

1. **Static Channel Allocation in LANs and MANs:**

It is the classical or traditional approach of allocating a single channel among multiple competing users FDM. If there are N users, the bandwidth is divided into N equal sized portions each user being assigned one portion.

Since each user has a private frequency band, there is no interface between users.

$$T = 1/ (U*C - L)$$

$$T(FDM) = N * T (1/U (C/N) – L/N)$$

Where

T = mean time delay

C = capacity of the channel,

L = arrival rate of frames

I/U = bits/frame

N= number of sub channels

It is not efficient to divide into fixed number of chunks.

T(FDM) = Frequency Division Multiplexing

2. Dynamic Channel Allocation:

Possible assumptions include:

1.**Station Model:**

Assumes that each of N stations independently produce frames. The probability of producing a packet in the interval IDt where I is the constant arrival rate of new frames.

2. **Single Channel Assumption:**

In this allocation all stations are equivalent and can send and receive on that channel.

3. **Collision Assumption:**

If two frames overlap in time-wise, then that's collision. Any collision is an error, and both frames must re transmitted. Collisions are only possible error.

4. **Time** can be divided into Slotted or Continuous.
5. **Stations** can sense a channel is busy before they try it.

6.

(a) ALOHA – Aloha was designed for wireless LAN which is applicable for shared medium. In this, multiple stations can transmit data at the same time and can hence lead to collision and data being garbled.

1. **Pure Aloha:**

When a station sends data it waits for an acknowledgement.

- If the acknowledgement doesn't come within the allotted-time then the station waits for a random amount of time called back-off time (Tb) and re-sends the data.
- Since different stations wait for different amount of time, the probability of further collision decreases.

Vulnerable time = 2 * Frame transmission time
Throughput = G exp{-2 * G}
Maximum throughput = 0.184 for G=0.5

2. **Slotted Aloha:**

This is similar to pure aloha except that we divide time into slots and sending of data is allowed only at the beginning of these slots. If a station misses out the allowed time, it must wait for the next slot. This reduces the probability of collision.

Vulnerable time = Frame transmission time
Throughput = G exp{-*G}
Maximum throughput = 0.368 for G=1.

7.

CSMA is a network access method which is used on shared network topologies such as Ethernet to control access to the network.

Devices attached to the network cable listen (carrier sense) before transmitting. If the channel is in use, devices wait before transmitting.

MA (Multiple Access) indicates that many devices can connect to and share the same network.

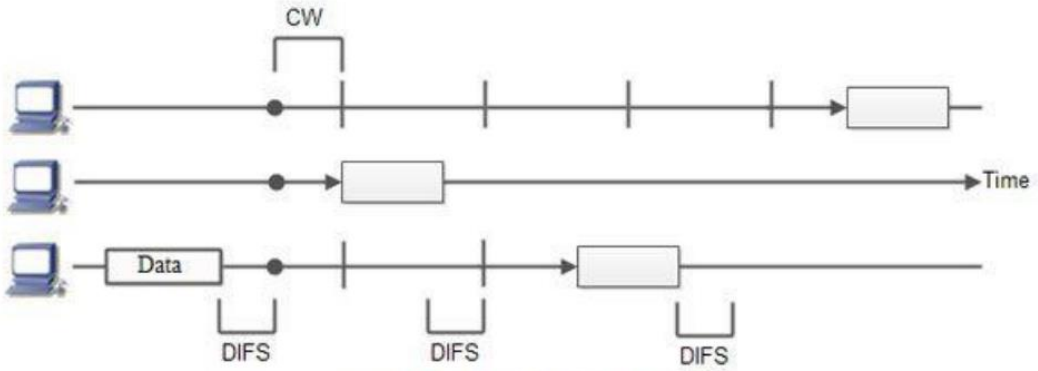
All devices have equal access to use the network when it is clear.

When a station wants to communicate “listen” first on the media communication and awaits a “silence” of a pre-set time (called the Distributed Inter Frame Space or DIFS).

After this compulsory period, the station starts a countdown for a random period considered. The maximum duration of this countdown is called the collision window (Window Collision, CW).

Consider the figure here the waiting time random has the advantage of allowing a statistically equitable distribution of speaking time between the various network equipment, while making little unlikely that both devices speak exactly the same time.

The countdown system prevents a station waiting too long before issuing its package. It's like a bit what place in a meeting room when no master session expected a silence then after few moments before speaking is allows time for someone else to speak.



Sharing time with CSMA method.

CSMA protocol was developed to overcome the problem found in ALOHA to minimize the chances of collision, so as to improve the performance. CSMA protocol is based on the principle of 'carrier sense'. The station senses the carrier or channel before transmitting a frame. It means the station checks the state of channel, whether it is idle or busy.

Even though devices attempt to sense whether the network is in use, there is a good chance that two stations will attempt to access it at the same time. On large networks, the transmission time between one end of the cable and another is enough that one station may access the cable even though another has already just accessed it.

The chances of collision still exist because of propagation delay. The frame transmitted by one station takes some time to reach other stations. In the meantime, other stations may sense the channel to be idle and transmit their frames. This results in the collision.

8.

Carrier Sense Multiple Access (CSMA) is a network protocol used to manage how devices on a shared medium (such as a local area network, or LAN) access the channel to avoid collisions. There are two main variations of this protocol: **CSMA/CA** (Collision Avoidance) and **CSMA/CD** (Collision Detection). These protocols ensure that devices can transmit data without excessive collisions, but they handle potential collisions in different ways.

1. CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

CSMA/CD is primarily used in **wired networks**, particularly in Ethernet. The goal of CSMA/CD is to minimize collisions and, when they occur, to detect them and handle them efficiently.

How CSMA/CD Works:

1. **Carrier Sensing (CS)**:
    - Before a device (node) transmits data, it listens to the shared communication channel (like a coaxial cable or twisted-pair Ethernet) to check if it is free or busy.
    - If the channel is free, the device proceeds to transmit its data.
    - If the channel is busy, the device waits for a random amount of time (backoff period) before sensing the channel again.
2. **Collision Detection (CD)**:
    - While transmitting data, the device continues to listen to the channel to detect any collisions (simultaneous transmissions by other devices).
    - If a collision occurs, the signal strength changes (due to interference from multiple signals on the wire). The device detects this and stops transmitting its data.
3. **Collision Handling**:
    - Upon detecting a collision, the device sends a **jam signal** to inform all devices on the network of the collision.
    - The devices then stop transmitting, wait for a random backoff time (using an exponential backoff algorithm), and retry transmitting.
4. **Transmission Resumption**:
    - After the backoff period, the devices sense the channel again. If it's free, they try to transmit. This process continues until the data is successfully transmitted.

Protocol Operation Summary:

1. Sense the channel.
2. If the channel is free, transmit the data.
3. If a collision occurs, stop transmitting, send a jam signal, and back off for a random time.
4. After the backoff, sense the channel again and repeat.

Example of CSMA/CD Use:

- **Ethernet (IEEE 802.3)**: CSMA/CD was extensively used in early Ethernet networks (10BASE-T) where multiple devices shared a common transmission medium. It is not commonly used in modern Ethernet networks because they now rely on switches, which prevent collisions by allowing full-duplex communication.

Pros of CSMA/CD:

- **Simple and effective**: Works well in networks where traffic is moderate.
- **Less idle time**: After collisions are detected, the protocol quickly recovers.

Cons of CSMA/CD:

- **Collision overhead**: Although collisions are detected, they still occur, leading to retransmissions and increased latency.
- **Not suitable for wireless networks**: In wireless networks, it's harder to detect collisions because the transmitter cannot listen while sending data.

---

2. CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)

CSMA/CA is used primarily in **wireless networks** (like Wi-Fi, IEEE 802.11). Unlike CSMA/CD, which deals with collisions after they occur, **CSMA/CA** attempts to **avoid collisions** altogether, as detecting collisions in wireless networks is more difficult.

How CSMA/CA Works:

1. **Carrier Sensing (CS)**:
    - Like in CSMA/CD, the device first listens to the channel to check if it is free.
    - If the channel is free, the device does not immediately transmit but initiates a process to avoid collisions.
2. **Collision Avoidance (CA)**:
    - Instead of immediately transmitting, the device waits for a random backoff time, even if the channel is free. This helps reduce the chances of two devices transmitting at the same time after sensing a free channel.
    - After the backoff period, the device senses the channel again. If it’s still free, the device sends a **Request to Send (RTS)** signal to the receiver.
3. **Clear to Send (CTS)**:
    - The receiver responds with a **Clear to Send (CTS)** signal if it is ready to receive data. This exchange informs other devices on the network that the sender and receiver are about to communicate, helping avoid collisions with other devices.
    - After receiving the CTS, the sender begins transmitting data.
4. **Transmission**:
    - The device sends its data. Once the data is received successfully, the receiver sends an **acknowledgment (ACK)** signal to the sender.

5. **Collision Handling**:
    - Collisions are less frequent, but if no ACK is received, the sender assumes a collision occurred and retries after a backoff period.

Protocol Operation Summary:

1. Sense the channel.
2. If the channel is free, wait for a random backoff time.
3. After backoff, sense again and send an RTS signal.
4. Upon receiving the CTS signal from the receiver, transmit the data.
5. Wait for an acknowledgment (ACK) from the receiver to confirm successful transmission.

Example of CSMA/CA Use:

- **Wi-Fi (IEEE 802.11)**: Wireless networks like Wi-Fi use CSMA/CA because it's difficult for a wireless transmitter to detect collisions while transmitting (a problem known as the "hidden node problem").

Pros of CSMA/CA:

- **Collision avoidance**: Reduces the chances of collisions happening in the first place, making it ideal for wireless networks.
- **Efficient in wireless environments**: RTS/CTS mechanisms help handle hidden nodes, improving network performance.

Cons of CSMA/CA:

- **Higher overhead**: The RTS/CTS handshake introduces additional delay before data transmission.
- **Increased latency**: The backoff times and the RTS/CTS mechanism can slow down data transmission, especially in congested networks.

9.

IEEE 802.11

IEEE 802.11 was the original version released in 1997. It provided 1 Mbps or 2 Mbps data rate in the 2.4 GHz band that used either frequency-hopping spread spectrum (FHSS) or direct-sequence spread spectrum (DSSS).

IEEE 802.11a

802.11a was published in 1999 as a modification to 802.11 with orthogonal frequency division multiplexing (OFDM) based air interface in physical layer instead of FHSS or DSSS of 802.11. It provides a maximum data rate of 54 Mbps operating in 5 GHz band. Besides, it provides error correcting code. As 2.4 GHz band is crowded, relatively sparsely used 5 GHz imparts additional advantage to 802.11a.

IEEE 802.11b

802.11b is a direct extension of the original 802.11 standard that appeared in early 2000. It uses the same modulation technique as 802.11 that is DSSS and operates in 2.4 GHz band. It has higher data rate of 11 Mbps as compared to 2 Mbps of 802.11, due to which it was rapidly adopted in wireless LANs. However since 2.4 GHz band is pretty crowded 802.11b devices faces interference from other devices.

IEEE 802.11g

802.11g was indorsed in 2003. It operates in the 2.4 GHz band (as in 802.11b) and provides a average throughput of 22 Mbps. It uses OFDM technique (as in 802.11a). It is fully backward compatible with 802.11b. 802.11g devices also faces interference from other devices operating in 2.4 GHz band.

IEEE 802.11n

802.11n was approved and published in 2009 that operates on both the 2.4 GHz and the 5 GHz bands. It has variable data rate ranging from 54 Mbps to 600 Mbps. It provides marked improvement over previous standards 802.11 by incorporating multiple-input multiple-output antennas (MIMO antennas).

IEEE 802.11p

802.11 includes wireless access in vehicular environments (WAVE) to support Intelligent Transportation Systems (ITS). They include network communications between vehicles moving at high speed and the environment. They have a data rate of 27 Mbps and operate in 5.9 GHz band.

3.6 Wireless LANS 802.11 standards

IEEE 802.11 is known as Wi-Fi which lays down the architecture and specifications of wireless LANs (WLANs). Wi-Fi or WLAN uses high frequency radio waves for connecting the nodes.

There are several standards of IEEE 802.11 WLANs. The prominent among them are 802.11, 802.11a, 802.11b, 802.11g, 802.11n and 802.11p. All the standards use carrier-sense multiple access with collision avoidance (CSMA/CA). They also have support for both centralised base- station based as well as ad-hoc networks.

10.

Difference between the Token Bus and the Token Ring:

S. No.	Token Bus Network	Token Ring Network
1.	In the token bus network, the token is passed along a virtual ring.	While in the token ring network the token is passed over a physical ring.
2.	The token bus network is simply designed for large factories.	While the token ring network is designed for the offices.
3.	The token bus network is defined by the IEEE 802.4 standard.	While the token ring network is defined by the IEEE 802.5 standard.
4.	Token bus network provides better bandwidth.	While the token ring network does not provide better bandwidth as compared to the token bus.
5.	In a token bus network, Bus topology is used.	While in token ring network, Star topology is used.
6.	The maximum time it takes to reach the last station in a token bus network cannot be calculated.	While the maximum time to reach the last station in the token ring network can be calculated.
7	In a token bus network, coaxial cable is used	In token ring network, twisted pair and fiber optic is used.
8	In a token bus network, the cable length is 200m to 500m.	In a token ring network, the cable length is 50m to 1000m.
9.	In token bus network, distributed algorithm provide maintenance.	In a token ring network, a designated monitor station performs station maintenance.
10.	The priority handling mechanism is not associated with the transmission of data through workstations with this network.	The priority handling mechanism is associated with the transmission of data through workstations with this network.
11.	These networks are not much reliable.	These networks are reliable.
12.	It does not keep routing details.	It keeps the information of routing.
13.	The network is less expensive compared to the Token Ring network.	It is expensive.

11.

The **network layer** of the OSI model is responsible for routing, forwarding, and addressing data packets to ensure that they are delivered from source to destination across multiple networks. Several design issues arise in the network layer, primarily because of its complexity and the requirements for handling data efficiently, reliably, and securely. Key design issues include:

1. **Routing**

- **Challenge**: The network layer must determine the best path for data to travel between the source and destination across potentially multiple interconnected networks.

- **Issues**:

  - **Optimal Path Selection**: The layer should select the shortest or most efficient path, often based on metrics like hop count, latency, or bandwidth.

  - **Dynamic vs. Static Routing**: Should the routing decisions be fixed (static routing) or adapt dynamically to network conditions (dynamic routing)?

  - **Scalability**: The routing algorithm should work well for small networks but also be scalable to large, global networks like the Internet.

2. **Addressing**

- **Challenge**: Each device must have a unique address (IP address) to ensure correct delivery of packets.

- **Issues**:

  - **Global Uniqueness**: Ensuring that all devices in the world have unique IP addresses, which led to the transition from IPv4 (limited address space) to IPv6 (expanded address space).

  - **Address Mapping**: Translating logical addresses (IP addresses) to physical addresses (MAC addresses) is necessary for communication between devices in the same network (handled by ARP).

  - **Hierarchical Addressing**: Proper management of address hierarchies to optimize routing decisions and reduce complexity.

3. **Packet Forwarding**

- **Challenge**: The network layer must forward packets based on the destination address, potentially across multiple routers.

- **Issues**:

- **Fragmentation and Reassembly**: Large packets may need to be broken into smaller fragments to pass through networks with smaller maximum transmission units (MTUs), and these fragments must be reassembled correctly at the destination.

- **Congestion Control**: Ensuring that intermediate routers do not become overloaded with traffic, which can cause packet loss and reduced network performance.

- **Efficient Forwarding**: Forwarding packets quickly and efficiently is critical to maintain high network throughput.

4. **Error Handling**

- **Challenge**: Packets can be lost, corrupted, or delayed due to various network issues.

- **Issues**:

- **Error Detection and Correction**: While the data link layer handles local error detection, the network layer may also need mechanisms to detect and, in some cases, recover from transmission errors.

- **Timeouts and Retransmissions**: The network must have mechanisms for dealing with lost packets, typically via retransmission.

5. **Quality of Service (QoS)**

- **Challenge**: Different applications (e.g., voice, video, file transfer) require different levels of service from the network.

- **Issues**:

- **Bandwidth Allocation**: Ensuring that applications with high bandwidth needs (e.g., video streaming) get enough bandwidth without starving other applications.

- **Latency and Jitter**: Real-time applications like voice and video conferencing require low latency and minimal jitter.

- **Traffic Prioritization**: The network layer may need to prioritize certain types of traffic over others to maintain performance levels.

6. **Interoperability Between Networks**

- **Challenge**: The network layer must enable communication between different types of networks, potentially with different technologies and addressing schemes.

- **Issues**:

- **Heterogeneous Networks**: Ensuring data can pass seamlessly across networks using different technologies (e.g., Ethernet, Wi-Fi, MPLS).

- **Tunneling**: Techniques like tunneling may be required to transport packets across incompatible networks (e.g., IPv6 packets over IPv4 networks).

7. **Security**

- **Challenge**: The network layer is vulnerable to attacks such as packet sniffing, spoofing, and denial of service (DoS).

- **Issues**:

  - **IP Spoofing and DDoS Attacks**: Attackers can forge the source IP address or flood the network with traffic to disrupt services.

  - **Data Integrity and Confidentiality**: The network layer must provide mechanisms (like IPsec) to ensure data is not tampered with and remains confidential while in transit.

These design challenges must be addressed to ensure efficient, scalable, and secure data transmission in a network.

12.

Aspect	Virtual Circuit Subnets	Datagram Subnets
Connection Type	Connection-oriented	Connectionless
Path Setup	A fixed path (virtual circuit) is established before data transmission.	No path setup; each packet is routed independently.
Routing	Fixed route per connection; all packets follow the same path.	Dynamic routing; packets can take different paths.
Overhead	Lower per packet after setup (since the route is fixed).	Higher per packet, as routing decisions are made for each packet.
Addressing	Only the virtual circuit identifier is used after setup.	Full destination and source address are included in each packet.
Reliability	Generally more reliable due to fixed routes and controlled traffic.	Less reliable; packets may be lost or arrive out of order.
Congestion Control	Easier to control congestion as traffic follows fixed paths.	More difficult, as congestion can occur dynamically due to varying paths.
Example	Frame Relay, ATM, MPLS	IP networks (such as the Internet)
Resource Allocation	Resources (like bandwidth) can be reserved during connection setup.	No resource reservation; best-effort delivery.
Packet Delivery Guarantee	Delivery order is maintained, and packets are not reordered.	No guarantee on order; packets may arrive out of sequence.
Use Case	Suitable for applications requiring consistent, ordered data transfer (e.g., voice calls, video conferencing).	Suitable for applications tolerant to out-of-order delivery (e.g., email, web browsing).

13.

The **Shortest Path Routing Algorithm** is a method used in computer networks to determine the most efficient path between two nodes (routers) in a network. It aims to find the route that has the least cost, where cost can be based on various metrics like distance, delay, bandwidth, or hop count. Two of the most common shortest path algorithms are **Dijkstra's Algorithm** and the **Bellman-Ford Algorithm**.

Key Concepts:

1. **Graph Representation**:
    - The network is represented as a graph, where:
        - **Nodes** represent routers or devices.
        - **Edges** represent communication links between routers.
        - **Weights/Costs** on edges represent the cost of traveling that link (e.g., based on latency, hop count, or bandwidth).
2. **Shortest Path**:
    - The shortest path between two nodes is the route that minimizes the total cost of traveling from one node to another, considering the weights assigned to the links.

1. Dijkstra's Algorithm (Commonly used for non-negative weights)

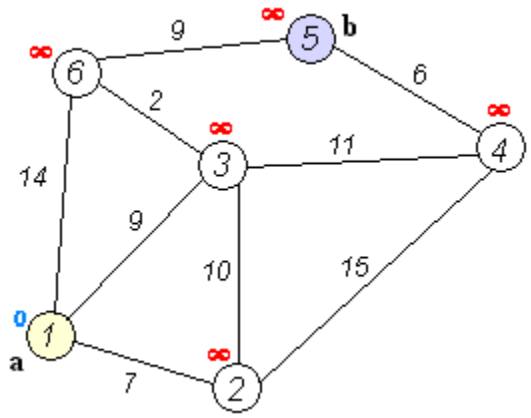
Overview: This algorithm is widely used in link-state routing protocols like OSPF (Open Shortest Path First). It finds the shortest path from a single source node to all other nodes in the graph.

Steps:

1. **Initialization**:
    - Set the distance to the source node as 0 and to all other nodes as infinity.
    - Mark all nodes as unvisited.
2. **Select the Node with the Smallest Distance**:
    - Start at the source node and visit the neighboring nodes.
    - For each neighboring node, update its distance if a shorter path is found via the current node.
3. **Update the Shortest Path**:
    - Continue visiting the unvisited node with the smallest known distance and repeat the process.
    - Once all nodes have been visited, the shortest path to each node is known.
4. **Termination**:
    - The algorithm terminates once all nodes have been visited and their shortest paths calculated.

Example:

- Assume a network where the distance between routers is given in kilometers:



-
    - Start at Node A. Calculate the shortest distance to neighboring nodes, and update distances iteratively until the shortest path from A to all other nodes is known.

Time Complexity:

- For a graph with V nodes and E edges, the time complexity is typically O(V^2) using an array but can be reduced to O((V + E) log V) using a priority queue.

2. Bellman-Ford Algorithm (Handles negative weights)

Overview: The Bellman-Ford algorithm is capable of handling negative edge weights, making it more flexible than Dijkstra's algorithm. It is used in distance-vector routing protocols like RIP (Routing Information Protocol).

Steps:

1. **Initialization**:
    - Set the distance to the source node as 0 and to all other nodes as infinity.
2. **Relax All Edges**:
    - For each edge in the graph, update the distance if a shorter path is found. Repeat this process for all edges V-1 times (where V is the number of nodes).
3. **Negative Cycle Check**:
    - After V-1 iterations, check if you can relax any edge further. If so, a negative cycle exists (meaning there is no valid shortest path).
4. **Termination**:
    - After V-1 iterations, the shortest path is known for all nodes, assuming no negative cycles.

Time Complexity:

- The time complexity is O(V * E).

14.

Distance-vector routing

- The distance-vector routing (DVR) is designed to periodically update the routing data in the network model based on the Bellman-Ford algorithm.
- The distance vector routing protocol is applied to assign the best and the shortest route for the data.
- In this network protocol, the distance refers to the distance (vector) between neighbouring nodes, and the routing refers to the established route.

Key Features of the protocol

Information About the Network

Every router is responsible for sharing the network knowledge in the channel more responsibly with the neighbouring nodes.

Routing Pattern of the Network

Sharing of routing information is done only between directly connected network nodes in the channel.

Sharing of Data Periodically

Each node in the connection is designed to share the updated routing data with each of the nodes in the network.

Algorithm Applied in Distance Vector Routing

The basis of distance vector routing is designed on the working of the Bellman-Ford Algorithm.

According to the algorithm, each of the nodes in the network is designed to maintain a distance-vector table carrying the distance between itself and its direct neighbouring nodes in the connection.

$$d_x(y) = \min_v\{c(x, v) + d_v(y)\}$$

Then using the algorithm, we can deduce the following points for the routing algorithm in the network:

The sharing of distance vectors between neighbouring nodes is done using routing tables.

Each routing table is shared with the latest distance-vector values in the network.

Each routing table data is shared at regular intervals in the network, with an update of the distance vector beyond the neighbouring nodes.

Where,

$d_x(y)$ - The least distance from x to y.

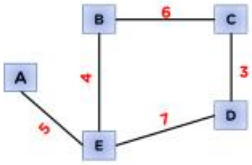
c(x, v) - Node x's cost from each of its neighbour v.

$d_v(y)$ - Distance of each neighbor from initial node.

min_v - Selecting the minimum distance for the data packet.

Network Model

To better understand the working of the routing protocol, we can do so by using the following steps in the example:



Initial Step

In the first step, we will design the routing table for node A, using the only neighboring node, i.e., node E.

Node A

Destination	Vector	Hop
A	0	A
B	∞	-
C	∞	-
D	∞	-
E	5	E

Next, we will go through the same step for node E, i.e.,

Node E

Destination	Vector	Hop
A	5	A
B	4	B
C	∞	-
D	7	D
E	0	E

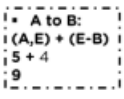
Similarly, we repeat the previous step and design the routing table for each of the nodes.

Update Step

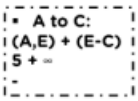
Now, after performing the initial step, we will perform the update step for all the nodes; for this, let's look into node A,

To update node A, we will use all the distance vectors from the nodes,

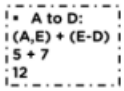
From node A to node B,



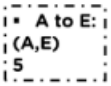
From node A to node C,



From node A to node D,



From node A to node E.



The final updated table for node A we get would be this,

Node A

Destination	Vector	Hop
A	0	A
B	9	E
C	∞	-
D	12	E
E	5	E

Similarly, we can perform the update step for all the nodes in the model, and this update step is to be followed for (n-1) iterations where n - Number of nodes. Going by our example model, we will perform the update step atleast four times, i.e., (5-1) = 4.

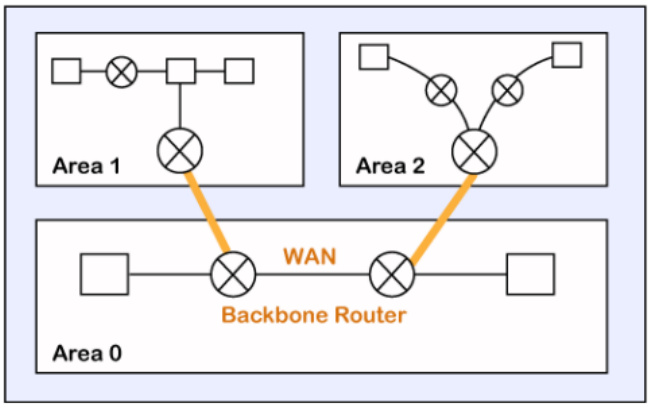
At the end of the update step, we will get the most efficient routing data for each node in the network model, where the sharing of routing data at regular intervals will still continue in the network.

15.

OSPF Protocol

- The OSPF stands for **Open Shortest Path First**.
- It is a widely used and supported routing protocol.
- It is an intradomain protocol, which means that it is used within an area or a network.
- It is an interior gateway protocol that has been designed within a single autonomous system.
- It is based on a link-state routing algorithm in which each router contains the information of every domain, and based on this information, it determines the shortest path.
- The goal of routing is to learn routes. The OSPF achieves by learning about every router and subnet within the entire network. Every router contains the same information about the network.
- The way the router learns this information by sending LSA (Link State Advertisements). These LSAs contain information about every router, subnet, and other networking information. Once the LSAs have been flooded, the OSPF stores the information in a link-state database known as LSDB. The main goal is to have the same information about every router in an LSDBs.

OSPF Areas



OSPF divides the autonomous systems into areas where the area is a collection of networks, hosts, and routers. Like internet service providers divide the internet into a different autonomous system for easy management and OSPF further divides the autonomous systems into Areas.

All the areas inside an autonomous system are connected to the backbone routers, and these backbone routers are part of a primary area. The role of a primary area is to provide communication between different areas.

How does OSPF work?

There are three steps that can explain the working of OSPF:

Step 1: The first step is to become OSPF neighbors. The two connecting routers running OSPF on the same link creates a neighbor relationship.

Step 2: The second step is to exchange database information. After becoming the neighbors, the two routers exchange the LSDB information with each other.

Step 3: The third step is to choose the best route. Once the LSDB information has been exchanged with each other, the router chooses the best route to be added to a routing table based on the calculation of SPF.

Types of links in OSPF

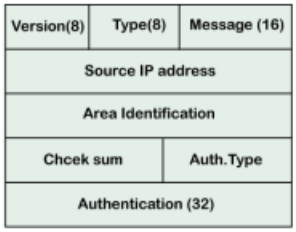
A link is basically a connection, so the connection between two routers is known as a link.

There are four types of links in OSPF:

1. **Point-to-point link:** The point-to-point link directly connects the two routers without any host or router in between.
2. **Transient link:** When several routers are attached in a network, they are known as a transient link.
3. **Stub link:** It is a network that is connected to the single router. Data enters to the network through the single router and leaves the network through the same router.
4. **Virtual link:** If the link between the two routers is broken, the administration creates the virtual path between the routers, and that path could be a long one also.

OSPF Message Format

The following are the fields in an OSPF message format:



- **Version:** It is an 8-bit field that specifies the OSPF protocol version.
- **Type:** It is an 8-bit field. It specifies the type of the OSPF packet.
- **Message:** It is a 16-bit field that defines the total length of the message, including the header. Therefore, the total length is equal to the sum of the length of the message and header.
- **Source IP address:** It defines the address from which the packets are sent. It is a sending routing IP address.
- **Area identification:** It defines the area within which the routing takes place.
- **Checksum:** It is used for error correction and error detection.

- **Authentication type:** There are two types of authentication, i.e., 0 and 1. Here, 0 means for none that specifies no authentication is available and 1 means for pwd that specifies the password-based authentication.
- **Authentication:** It is a 32-bit field that contains the actual value of the authentication data.

OSPF Packets

There are five different types of packets in OSPF:

1. Hello packet

The Hello packet is used to create a neighborhood relationship and check the neighbor's reachability. Therefore, the Hello packet is used when the connection between the routers need to be established.

2. Database Description

After establishing a connection, if the neighbor router is communicating with the system first time, it sends the database information about the network topology to the system so that the system can update or modify accordingly.

3. Link state request

The link-state request is sent by the router to obtain the information of a specified route. Suppose there are two routers, i.e., router 1 and router 2, and router 1 wants to know the information about the router 2, so router 1 sends the link state request to the router 2. When router 2 receives the link state request, then it sends the link-state information to router 1.

4. Link state update

The link-state update is used by the router to advertise the state of its links. If any router wants to broadcast the state of its links, it uses the link-state update.

5. Link state acknowledgment

The link-state acknowledgment makes the routing more reliable by forcing each router to send the acknowledgment on each link state update. For example, router A sends the link state update to the router B and router C, then in return, the router B and C sends the link-state acknowledgment to the router A, so that the router A gets to know that both the routers have received the link-state update.

17.

IP Address Format and Table

- IP address is a short form of "Internet Protocol Address."
- It is a unique number provided to every device connected to the internet network, such as Android phone, laptop, Mac, etc.
- An IP address is represented in an integer number separated by a dot (.), for example, 192.167.12.46.

Types of IP Address

An IP address is categorized into two different types based on the number of IP address it contains. These are:

- IPv4 (Internet Protocol version 4)
- IPv6 (Internet Protocol version 6)

What is IPv4?

- IPv4 is version 4 of IP. It is a current version and the most commonly used IP address.
- It is a 32-bit address written in four numbers separated by a dot (.), i.e., periods.
- This address is unique for each device. For example, 66.94.29.13

What is IPv6?

- IPv4 produces 4 billion addresses, and the developers think that these addresses are enough, but they were wrong.
- IPv6 is the next generation of IP addresses.
- The main difference between IPv4 and IPv6 is the address size of IP addresses.
- The IPv4 is a 32-bit address, whereas IPv6 is a 128-bit hexadecimal address.
- IPv6 provides a large address space, and it contains a simple header as compared to IPv4.

IPv4 Datagram Header

- **VERSION*:*** Version of the IP protocol (4 bits), which is 4 for IPv4
- **HLEN:** IP header length (4 bits), which is the number of 32 bit words in the header. The minimum value for this field is 5 and the maximum is 15.
- **Type of service:** Low Delay, High Throughput, Reliability (8 bits)
- **Total Length:** Length of header + Data (16 bits), which has a minimum value 20 bytes and the maximum is 65,535 bytes.
- **Identification*:*** Unique Packet Id for identifying the group of fragments of a single IP datagram (16 bits)
- **Flags*:*** 3 flags of 1 bit each : reserved bit (must be zero), do not fragment flag, more fragments flag (same order)
- **Fragment Offset:** Represents the number of Data Bytes ahead of the particular fragment in the particular Datagram. Specified in terms of number of 8 bytes, which has the maximum value of 65,528 bytes.

- **Time to live*:*** Datagram's lifetime (8 bits), It prevents the datagram to loop through the network by restricting the number of Hops taken by a Packet before delivering to the Destination.
- **Protocol:** Name of the protocol to which the data is to be passed (8 bits)
- **Header Checksum*:*** 16 bits header checksum for checking errors in the datagram header
- **Source IP address*:*** 32 bits IP address of the sender
- **Destination IP address*:*** 32 bits IP address of the receiver
- **Option:** Optional information such as source route, record route. Used by the Network administrator to check whether a path is working or not.

18.

IP Address Format

- Originally IP addresses were divided into five different categories called **classes**.
- These divided IP classes are class A, class B, class C, class D, and class E.
- Out of these, classes A, B, and C are most important.
- Each address class defines a different number of bits for its **network prefix (network address)** and **host number (host address)**.
- The starting address bits decide from which class an address belongs.

Network Address: The network address specifies the unique number which is assigned to your network. In the above figure, the network address takes two bytes of IP address.

Host Address: A host address is a specific address number assigned to each host machine. With the help of the host address, each machine is identified in your network. The network address will be the same for each host in a network, but they must vary in host address.

Address Format IPv4

The address format of IPv4 is represented into **4-octets** (32-bit), which is divided into three different classes, namely class A, class B, and class C.

The above diagram shows the address format of IPv4. An IPv4 is a 32-bit decimal address. It contains four octets or fields separated by 'dot,' and each field is 8-bit in size. The number that each field contains should be in the range of 0-255.

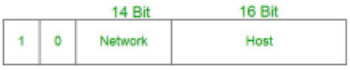
Class A

- **Class A** address uses only first higher order octet (byte) to identify the network prefix, and remaining three octets (bytes) are used to define the individual host addresses.
- The class A address ranges between 0.0.0.0 to 127.255.255.255.
- The first bit of the first octet is always set to 0 (zero), and next 7 bits determine network address, and the remaining 24 bits determine host address. So the first octet ranges from 0 to 127 (00000000 to 01111111).

Class A

Class B

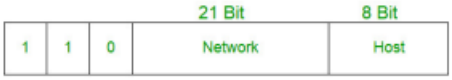
- **Class B** addresses use the initial two octets (two bytes) to identify the network prefix, and the remaining two octets (two bytes) define host addresses.
- The class B addresses are range between 128.0.0.0 to 191.255.255.255.
- The first two bits of the first higher octet is always set to 10 (one and zero bit), and next 14 bits determines the network address and remaining 16 bits determines the host address.
- So the first octet ranges from 128 to 191 (10000000 to 10111111).



Class B

Class C

- **Class C** addresses use the first three octets (three bytes) to identify the network prefix, and the remaining last octet (one byte) defines the host address.
- The class C address ranges between 192.0.0.0 to 223.255.255.255.
- The first three bit of the first octet is always set to 110, and next 21 bits specify network address and remaining 8 bits specify the host address.
- Its first octet ranges from 192 to 223 (11000000 to 11011111).



Class C

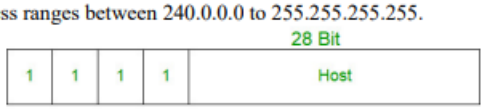
Class D

- **Class D** IP address is reserved for multicast addresses.
- Its first four bits of the first octet are always set to 1110, and the remaining bits determine the host address in any IP address.
- The first higher octet bits are always set to 1110, and the remaining bits specify the host address.
- The class D address ranges between 224.0.0.0 to 239.255.255.255.
- In multicasting, data is not assigned to any particular host machine, so it is not require to find the host address from the IP address, and also, there is no subnet mask present in class D.

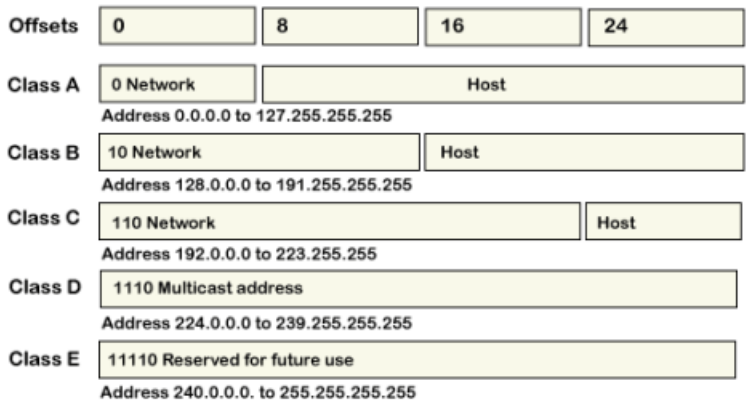
Class D

Class E

- **Class E** IP address is reserved for experimental purposes and future use.
- It does not contain any subnet mask in it.
- The first higher octet bits are always set to 1111, and next remaining bits specify the host address.
- Class E address ranges between 240.0.0.0 to 255.255.255.255.



Class E



IP Address Format IPv6

- All IPv6 addresses are 128-bit hexadecimal addresses, written in 8 separate sections having each of them have 16 bits.
- As the IPv6 addresses are represented in a hexadecimal format, their sections range from 0 to FFFF.
- Each section is separated by colons (:). 'It also allows to removes the starting zeros (0) of each 16-bit section.
- If two or more consecutive sections 16-bit contains all zeros (0 : 0), they can be compressed using double colons (::).

IPv6 addresses are consist of 8 different sections, each section has a 16-bit hexadecimal values separated by colon (:).

IPv6 addresses are represented as following format:

xxxx : xxxx : xxxx : xxxx : xxxx : xxxx : xxxx : xxxx

Each "xxxx" group contains a 16-bit hexadecimal value, and each "x" is a 4-bit hexadecimal value. For example:

FDEC : BA98 : 0000 : 0000 : 0600 : BDFF : 0004 : FFFF

19.

Sr.No.	IPv4	IPv6
1	IPv4 has a 32-bit address length	IPv6 has a 128-bit address length
2	It Supports Manual and DHCP address configuration	It supports Auto and renumbering address configuration
3	In IPv4 end to end, connection integrity is Unachievable	In IPv6 end-to-end, connection integrity is Achievable
4	It can generate 4.29×10 9 address space	The address space of IPv6 is quite large it can produce 3.4×10 38 address space
5	The Security feature is dependent on the application	IPSEC is an inbuilt security feature in the IPv6 protocol
6	Address representation of IPv4 is in decimal	Address representation of IPv6 is in hexadecimal
7	Fragmentation performed by Sender and forwarding routers	In IPv6 fragmentation is performed only by the sender
8	In IPv4 Packet flow identification is not available	In IPv6 packet flow identification are Available and uses the flow label field in the header
9	In IPv4 checksum field is available	In IPv6 checksum field is not available
	It has a broadcast Message Transmission Scheme	In IPv6 multicast and anycast message transmission scheme is available

Sr.No.	IPv4	IPv6
10	In IPv4 Encryption and Authentication facility not provided	In IPv6 Encryption and Authentication are provided
11	IPv4 has a header of 20-60 bytes.	IPv6 has a header of 40 bytes fixed
12	IPv4 can be converted to IPv6	Not all IPv6 can be converted to IPv4
13	IPv4 consists of 4 fields which are separated by addresses dot (.)	IPv6 consists of 8 fields, which are separated by a colon (:)
14	IPv4's IP addresses are divided into five different classes. Class A , Class B, Class C, Class D , Class E.	IPv6 does not have any classes of the IP address.
15	IPv4 supports VLSM(Variable Length subnet mask).	IPv6 does not support VLSM.
16	Example of IPv4: 66.94.29.13	Example of IPv6: 2001:0000:3238:DFE1:0063:0000:0000:FEFB

20.

Sr. No.	Broadcast	Multicast
1	It has one sender and multiple receivers.	It has one or more senders and multiple receivers.
2	It sent data from one device to all the other devices in a network.	It sent data from one device to multiple devices.
3	It works on star and bus topology.	It works on star, mesh, tree and hybrid topology.
4	It scale well across large networks.	It does not scale well across large networks.
5	Its bandwidth is wasted.	It utilizes bandwidth efficiently.
6	It has one-to-all mapping.	It has one-to-many mapping.
7	Hub is an example of a broadcast device.	Switch is an example of a multicast device.
8	It increases network traffic because the data packets are sent to every other node in the network.	It doesn't increase network traffic.
9	The message to be sent should be tripled checked as some sensitive or confidential information shouldn't be distributed to everyone in the network.	No such issue, because the message is target to only selected people.