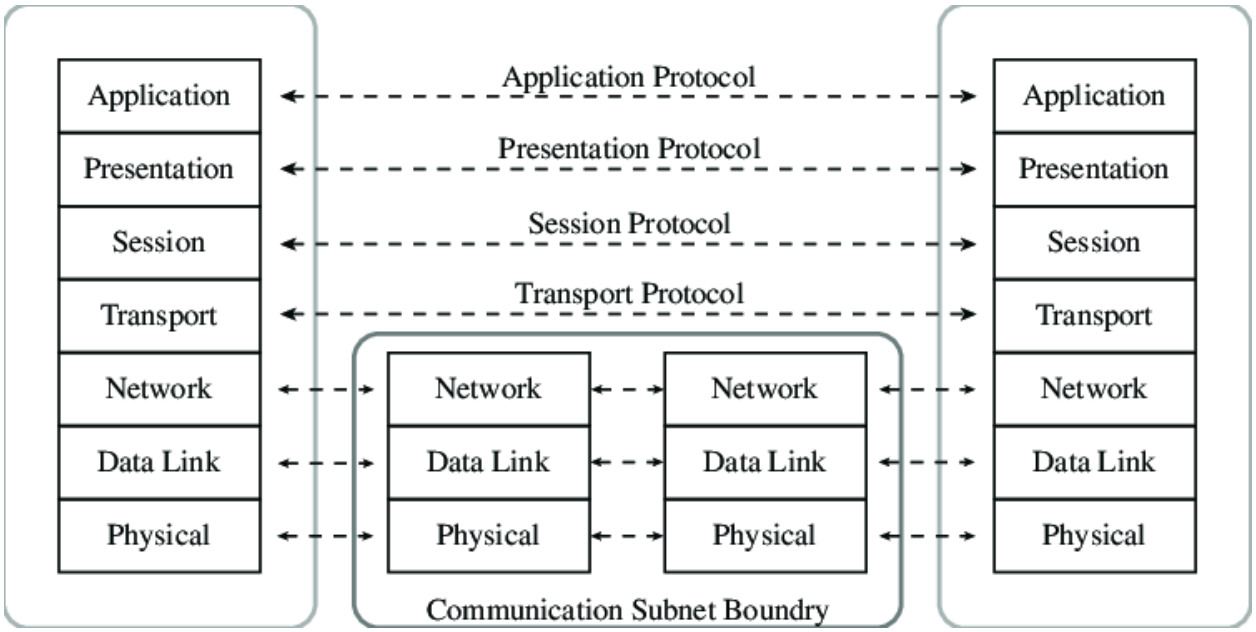
Q1:-explain ISO-OSI reference model with diagram / design issues of OSI layers(10m/5m)



1. Physical Layer

- **Signal Attenuation**: Signal strength decreases over distance, which can affect data integrity.
- **Data Rate**: Determining how fast data should be transmitted over the network medium.
- **Physical Connections**: Deciding on connectors, cable types, and physical standards for effective data transfer.

2. Data Link Layer

- **Error Detection and Correction**: Ensuring that errors introduced during transmission are identified and corrected.
- **Flow Control**: Managing data flow between devices to prevent faster devices from overwhelming slower ones.
- **Medium Access Control**: Determining how multiple devices share a common communication medium (e.g., Ethernet).

3. Network Layer

- **Routing**: Finding the best path for data to travel across networks.

- **Congestion Control**: Managing data flow to avoid network congestion, which can cause delays or data loss.
- **Addressing**: Assigning unique network addresses to devices for identification and routing.

4. Transport Layer

- **Reliability**: Ensuring complete and accurate data transfer, often through acknowledgment and retransmission of lost packets.
- **Flow Control and Error Handling**: Managing data flow and detecting/correcting errors for end-to-end communication.
- **Multiplexing**: Allowing multiple applications to use the network simultaneously by separating data streams.

5. Session Layer

- **Session Management**: Establishing, maintaining, and terminating sessions between applications.
- **Synchronization**: Managing checkpoints to ensure data can be recovered if a session is interrupted.
- **Dialog Control**: Handling bidirectional data flow, determining whether communication is half-duplex or full-duplex.

6. Presentation Layer

- **Data Translation**: Converting data formats to ensure compatibility between different systems.
- **Encryption and Decryption**: Securing data by encoding and decoding to protect privacy.
- **Data Compression**: Reducing data size to optimize transmission speed and storage.

7. Application Layer

- **Application Protocol Standardization**: Ensuring different applications can communicate by following standard protocols (e.g., HTTP, FTP).
- **User Authentication and Authorization**: Validating user identity and managing access to network services.

- **Service Quality**: Ensuring network services meet specific performance standards, such as response time and reliability.

design issues of OSI layers are as follow:

1. Reliability

- **Definition**: Making sure data is sent accurately and arrives at the destination without errors.
- **Explanation**: Reliability involves checking for errors and, if data is lost or damaged, resending it to ensure that what the receiver gets is exactly what was sent.

2. Addressing

- **Definition**: Giving each device a unique identifier so data knows where to go.
- **Explanation**: Just like a house address, addressing allows data packets to find the correct device on a network, helping ensure that information reaches the right place.

3. Error Control

- **Definition**: Detecting and correcting errors in transmitted data.
- **Explanation**: Errors can happen when data travels across a network. Error control methods find these errors and fix them, ensuring that the receiver gets the correct information.

4. Flow Control

- **Definition**: Managing the rate of data transmission so that fast senders don't overwhelm slow receivers.
- **Explanation**: It's like pouring water slowly into a glass to avoid spilling—flow control regulates how fast data is sent, ensuring the receiver can handle it without being overwhelmed.

5. Multiplexing and Demultiplexing

- **Multiplexing**: Combining multiple data streams into one for transmission.

- **Demultiplexing**: Separating the combined data stream back into individual streams for each application.
- **Explanation**: Think of multiplexing as packing several letters into one envelope for mailing. Demultiplexing is when the recipient opens the envelope and sorts each letter to its intended reader.

6. Scalability

- **Definition**: The ability of a network or system to handle a growing amount of work or users.
- **Explanation**: A scalable network can expand to support more devices and traffic without performance issues. It's like adding more seats to a theater without crowding or discomfort.

7. Routing

- **Definition**: Finding the best path for data to travel from sender to receiver across networks.
- **Explanation**: Routing works like a GPS, directing data packets through the quickest and most efficient paths to their destination.

8. Confidentiality

- **Definition**: Ensuring that only authorized people can access the data.
- **Explanation**: Confidentiality is like locking private information in a safe so only those with the key (authorized people) can see it.

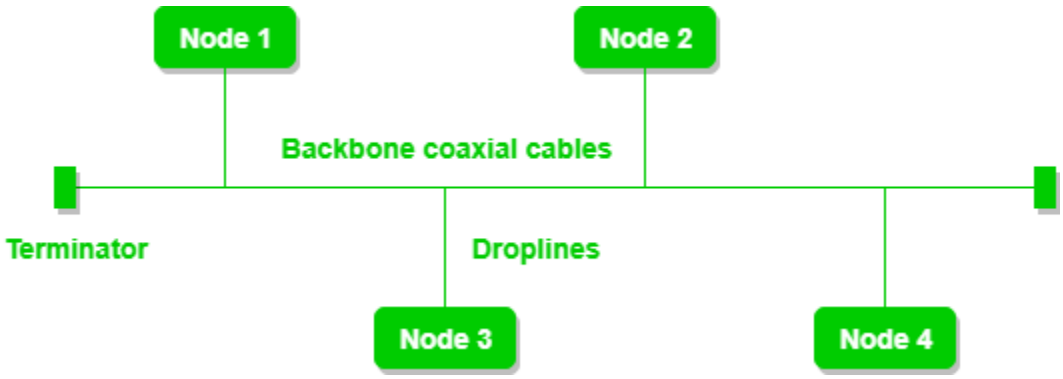
9. Integrity

- **Definition**: Ensuring data isn't altered or tampered with during transmission.
- **Explanation**: Integrity is like sealing an envelope so the recipient knows the contents haven't been changed. If the seal is broken, they know something is wrong.

Q2:-Network Topologies (Bus, Ring, Mesh, etc.)

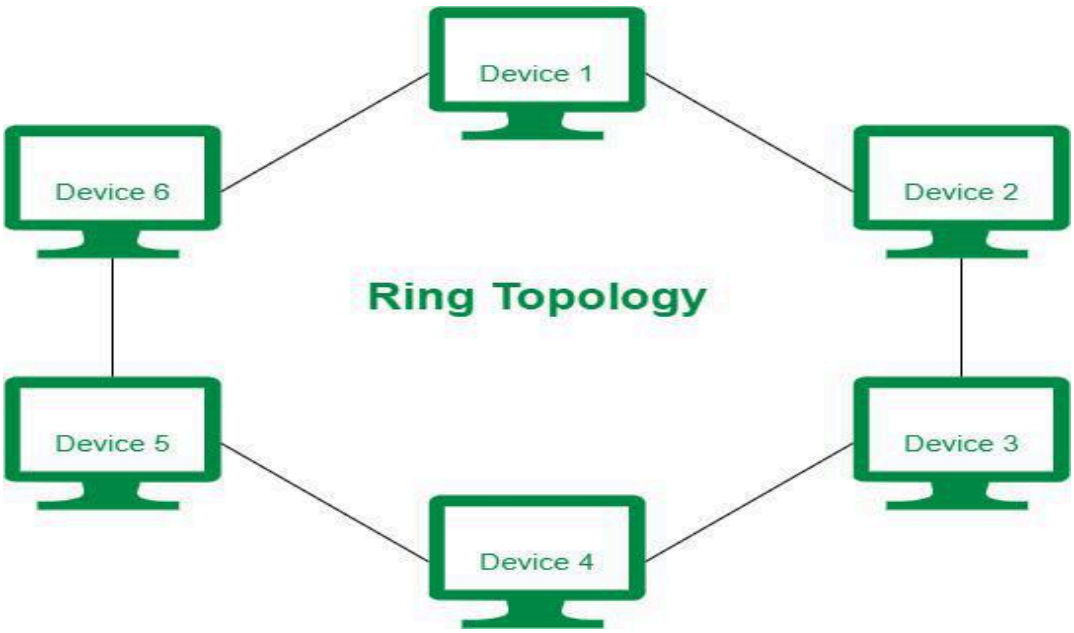
1. Bus Topology

- **Structure**: All devices are connected to a single central cable called the "bus."
- **Features**: Simple to set up, but if the main cable fails, the entire network goes down. Each device shares the same communication line, so only one device can send data at a time.



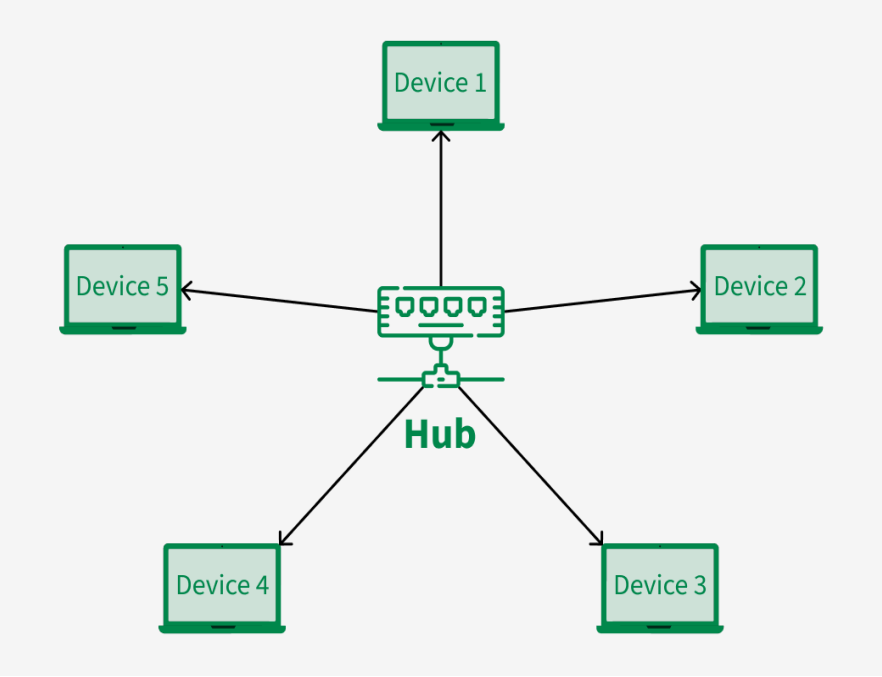
2. Ring Topology

- **Structure**: Each device is connected to two other devices, forming a circular or "ring" structure. Data travels in one direction (or in both directions in a dual ring).
- **Example**: The second image you uploaded represents a ring topology, with devices arranged in a circular pattern.
- **Pros**: Easy to troubleshoot and predictable data travel path.
- **Cons**: If one device fails, it can disrupt the entire network unless it’s a dual ring.



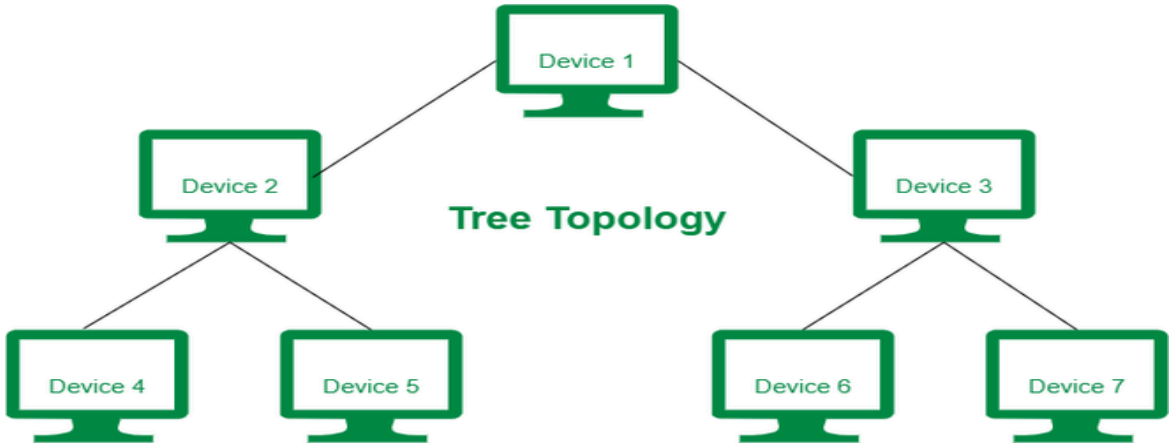
3. Star Topology

- **Structure**: All devices are connected to a central hub or switch. Data flows through this central point.
- **Pros**: If one cable fails, it doesn't affect the rest of the network; easy to add new devices.
- **Cons**: If the central hub fails, the entire network goes down; requires more cable than a bus topology.



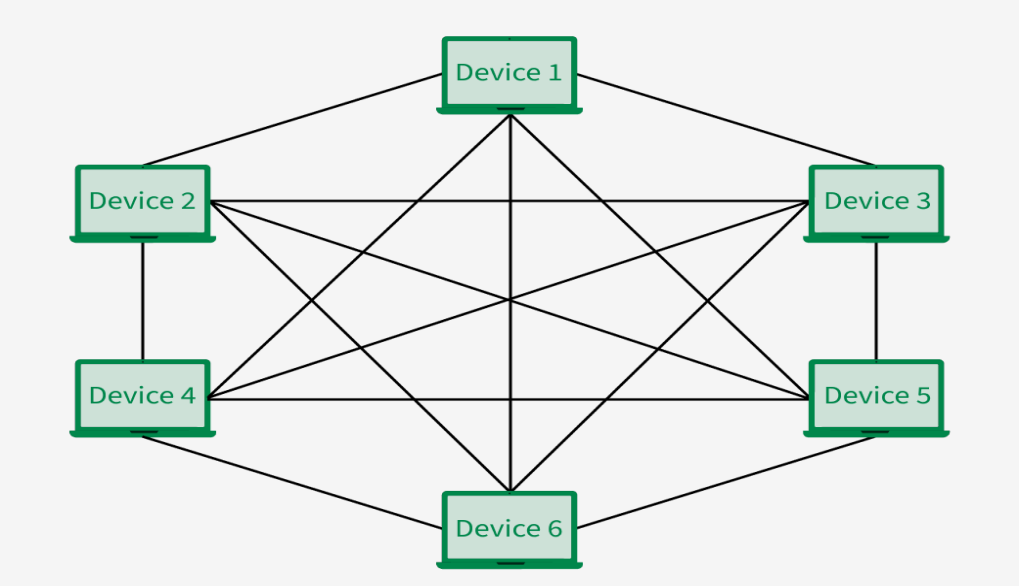
4. Tree Topology

- **Structure**: A combination of bus and star topologies, where groups of star-configured devices are connected to a central backbone.
- **Pros**: Supports scalable networks and is easier to expand.
- **Cons**: If the backbone fails, segments connected to it are also affected; requires more cabling



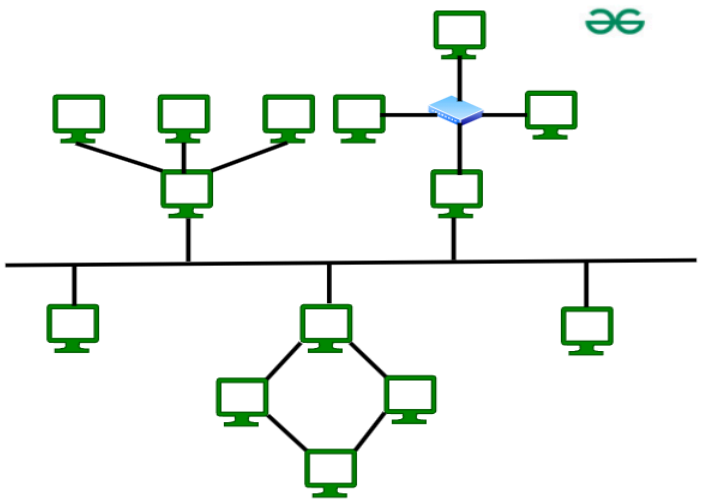
5. Mesh Topology

- **Structure**: Every device is connected to every other device, creating multiple paths for data to travel.
- **Pros**: Highly reliable, as there are multiple routes for data to travel; if one connection fails, data can still be rerouted.
- **Cons**: Expensive and difficult to set up due to the large number of connections and cables required.



6. Hybrid Topology

- **Structure**: A combination of two or more topologies (e.g., a mix of star and ring or star and bus topologies).
- **Pros**: Flexible, scalable, and can adapt to different needs; can combine the benefits of different topologies.
- **Cons**: Complex to design and manage, as it combines various types of connections.



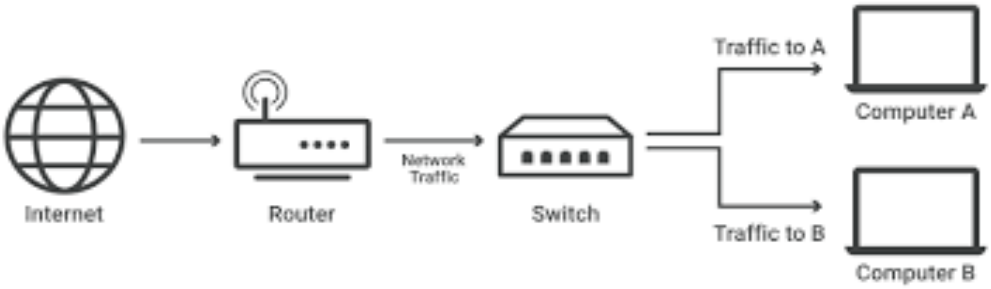
Q3:-Network/Interconnecting Devices (Hub, Router, etc.)

1. Hub

- **Function**: Connects multiple devices in a network.
- **How It Works**: When data arrives at one port, the hub copies it and sends it to all other ports, so all connected devices receive the data.
- **Limitations**: Can create network congestion since it sends data to all devices, even if only one device needs it.

2. Switch

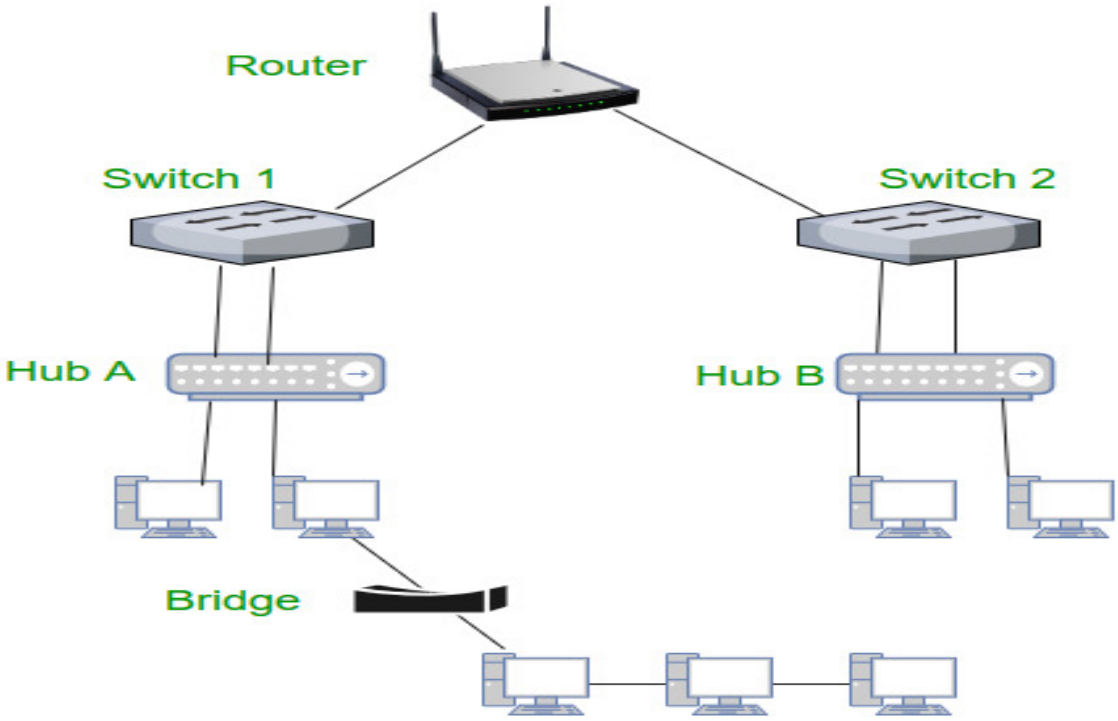
- **Function**: Connects multiple devices in a network, like a hub, but smarter.
- **How It Works**: Unlike a hub, a switch sends data only to the specific device that needs it, not to all connected devices. It learns which devices are connected to which ports.
- **Benefits**: Reduces network traffic and improves efficiency by only sending data to the correct device.



3. Router

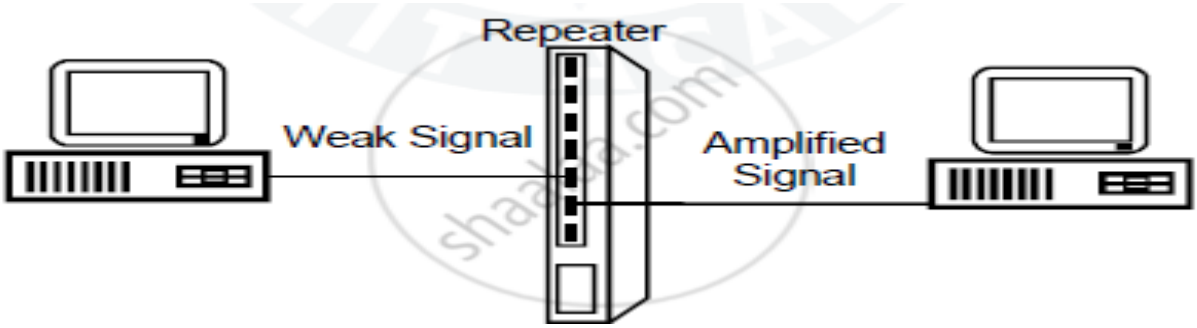
- **Function**: Connects different networks together and directs data between them.

- **How It Works**: Routers use IP addresses to determine the best route for data to travel between devices on different networks, such as between your home network and the internet.
- **Benefits**: Allows multiple networks to communicate with each other, manages traffic efficiently, and provides internet access.



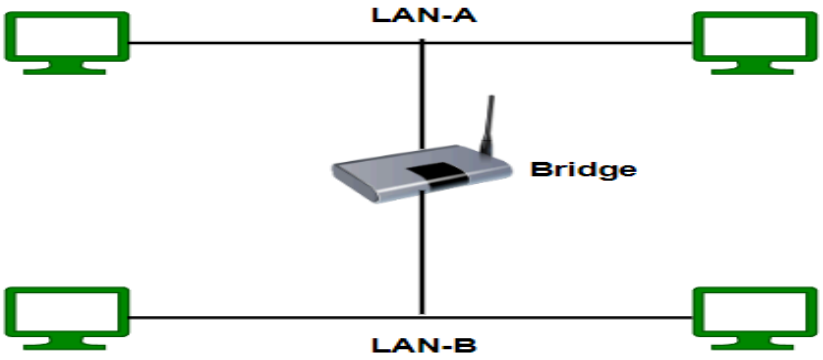
4. Repeater

- **Function**: Extends the range of a network.
- **How It Works**: A repeater amplifies or regenerates weak signals so they can travel longer distances. It's useful in large buildings or outdoor areas.
- **Benefits**: Ensures data can travel farther without losing signal quality.



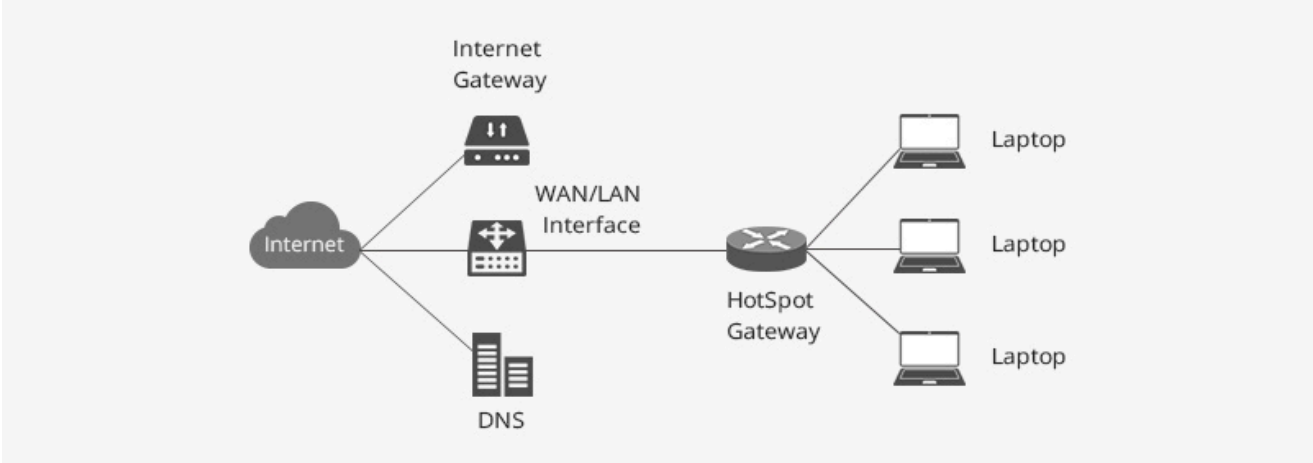
5. Bridge

- **Function**: Connects two or more network segments, like two separate local area networks (LANs).
- **How It Works**: Bridges allow data to pass between segments based on MAC addresses. They help divide a network into smaller parts, reducing traffic and improving performance.
- **Benefits**: Reduces congestion by segmenting large networks and allows communication between different network segments.



6. Gateway

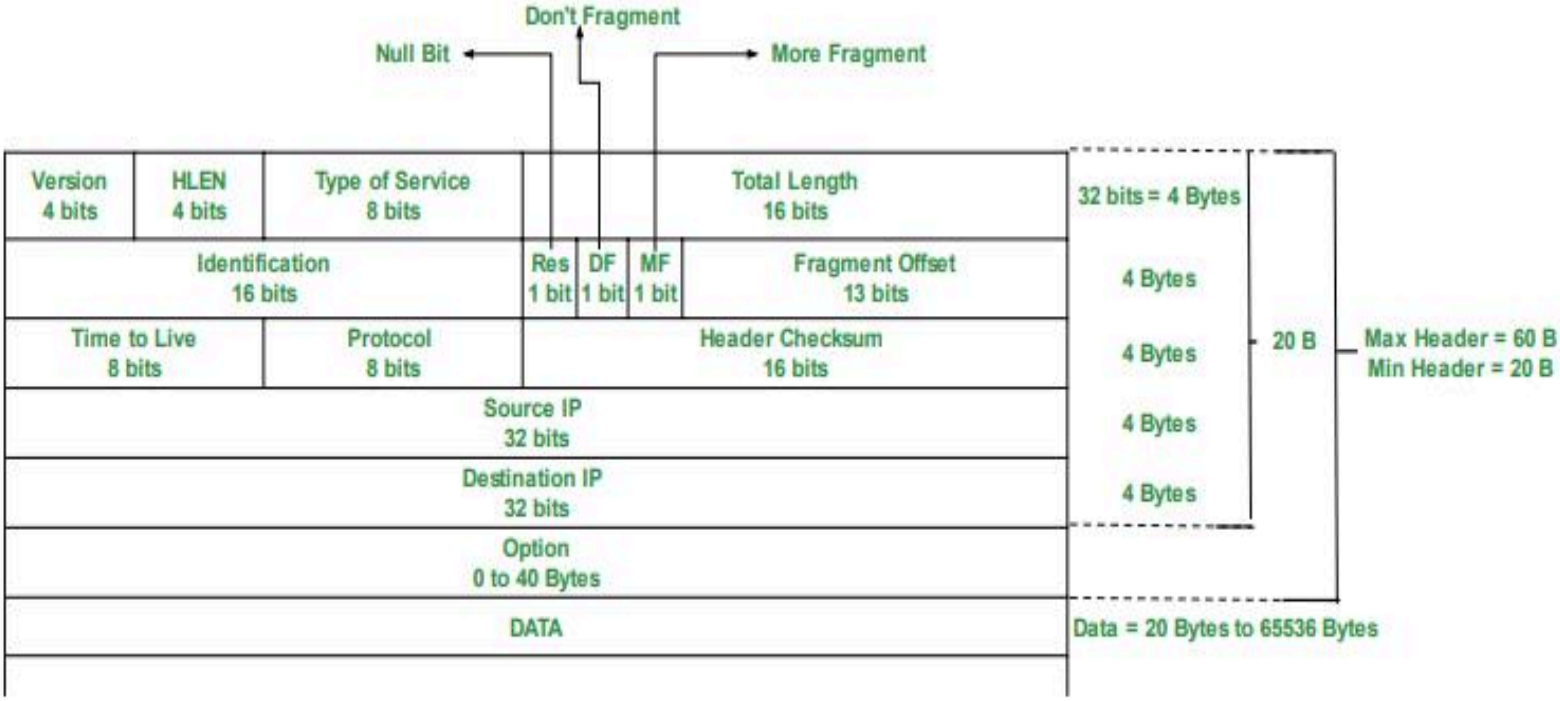
- **Function**: Acts as an entry and exit point to connect different networks, often those with different protocols.
- **How It Works**: Gateways can translate protocols (rules for data exchange) to allow communication between different types of networks, such as a local network and the internet.
- **Benefits**: Enables communication between different types of networks that otherwise couldn't connect, such as a LAN and the internet.



Q4:-Connection-less vs Connection-oriented Services

Connection-oriented Service	Connection-less Service
Related to the telephone system.	Related to the postal system.
Preferred for long and steady communication.	Preferred by bursty communication.
Necessary for certain types of communication.	Not compulsory.
Feasible and reliable.	Not always feasible or reliable.
Congestion is not possible.	Congestion is possible.
Guarantees reliability.	Does not guarantee reliability.
Packets follow the same route.	Packets do not follow the same route.
Requires a high bandwidth range.	Requires a low bandwidth range.
Example: TCP (Transmission Control Protocol).	Example: UDP (User Datagram Protocol).
Requires authentication.	Does not require authentication.

Q5:-Explain IPv4 Header Format(10m/5m)



VERSION: Indicates the IP protocol version (4 bits). For IPv4, this value is 4.

HLEN: Header length (4 bits), which is the number of 32-bit words in the header. It can range from 5 to 15.

Type of Service: Specifies the priority of the packet (8 bits), such as Low Delay, High Throughput, or Reliability.

Total Length: The total length of the header and data combined (16 bits), with a minimum of 20 bytes and a maximum of 65,535 bytes.

Identification: A unique ID for each packet (16 bits), used to identify fragments that belong to the same data packet.

Flags: Contains three 1-bit flags:

- Reserved bit (must always be zero),
- "Do Not Fragment" flag,
- "More Fragments" flag (indicates more fragments are coming).

Fragment Offset: Shows the position of the fragment within the original data packet. It's measured in units of 8 bytes, with a maximum value of 65,528 bytes.

Time to Live (TTL): Sets the packet's lifetime (8 bits), limiting the number of hops a packet can take to prevent it from endlessly looping in the network.

Protocol: Specifies the protocol (8 bits) to which the data should be passed (e.g., TCP or UDP).

Header Checksum: A 16-bit value used to detect errors in the header.

Source IP Address: The 32-bit IP address of the sender.

Destination IP Address: The 32-bit IP address of the receiver.

Option: Optional field for additional information, like source route or record route, often used by network administrators to verify network paths.

Q6:-Explain Classful & Classless IPv4(10m)

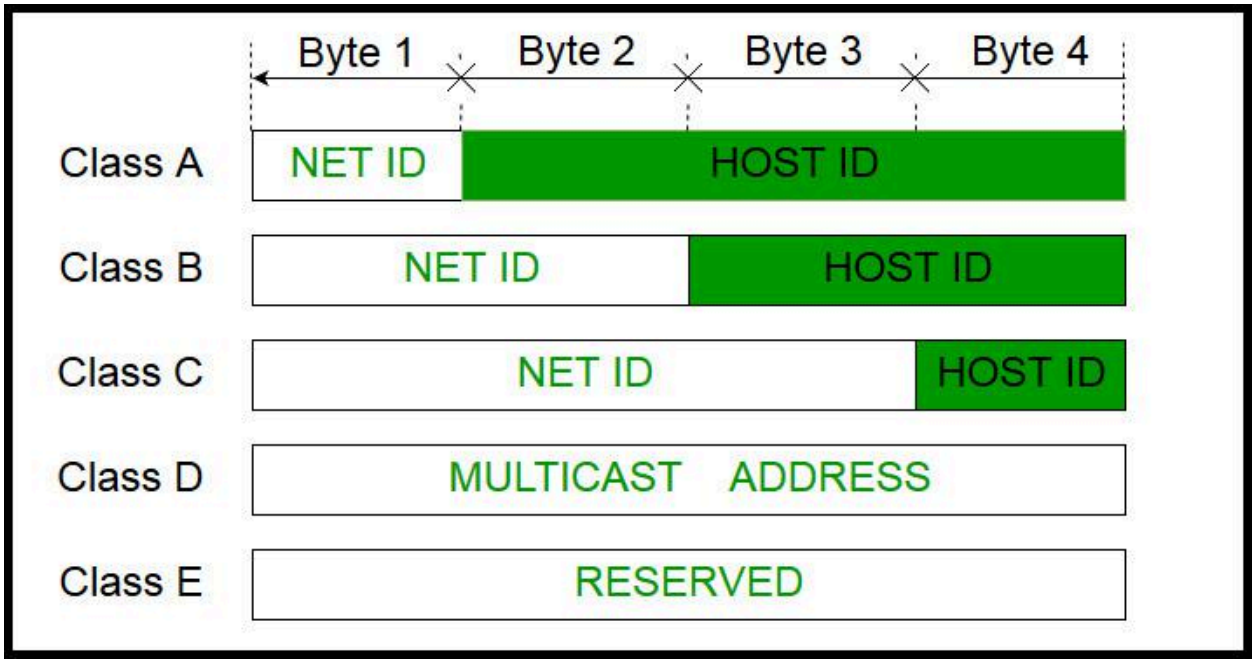
Define:

Classful addressing was introduced in 1981, dividing IPv4 addresses into five classes (A to E), each with a specific range. The class of an IP address decides how to separate the network part from the host part using a specific subnet mask for that class. However, classful addressing was not flexible and caused problems with how addresses were allocated. This led to the creation of classless addressing (CIDR) to use IP address space more efficiently.

```
Classes A-C: unicast addresses

Class D: multicast addresses

Class E: reserved for future use
```



Class A

- **Structure**: In a Class A address, the first bit of the first octet is always '0'.
- **Range**: Class A addresses range from 0.0.0.0 to 127.255.255.255 (because 01111111 in binary converts to 127 in decimal).
- **Network and Host Portion**: The first 8 bits (or the first octet) represent the network portion, and the remaining 24 bits (or 3 octets) represent the host portion.
- **Subnet Mask**: The default subnet mask for Class A is 255.0.0.0.

- **Example**: 10.0.0.1 is a Class A address, where '10' represents the network, and '0.0.1' represents the host.

Class B

- **Structure**: In a Class B address, the first two bits of the first octet are always '10'.
- **Range**: Class B addresses range from 128.0.0.0 to 191.255.255.255 (because 10111111 in binary converts to 191 in decimal).
- **Network and Host Portion**: The first 16 bits (or 2 octets) represent the network portion, and the remaining 16 bits (or 2 octets) represent the host portion.
- **Subnet Mask**: The default subnet mask for Class B is 255.255.0.0.
- **Example**: 172.16.0.1 is a Class B address, where '172.16' represents the network, and '0.1' represents the host.

Class C

- **Structure**: In a Class C address, the first three bits of the first octet are always '110'.
- **Range**: Class C addresses range from 192.0.0.0 to 223.255.255.255 (because 11011111 in binary converts to 223 in decimal).
- **Network and Host Portion**: The first 24 bits (or 3 octets) represent the network portion, and the last 8 bits (or 1 octet) represent the host portion.
- **Subnet Mask**: The default subnet mask for Class C is 255.255.255.0.
- **Example**: 192.168.1.1 is a Class C address, where '192.168.1' represents the network, and '1' represents the host.

Class D

- **Structure**: In a Class D address, the first four bits of the first octet are always '1110'.
- **Range**: Class D addresses range from 224.0.0.0 to 239.255.255.255 (because 11101111 in binary converts to 239 in decimal).
- **Usage**: Class D addresses are reserved for multicast, which is used to send data to multiple devices at once.
- **Subnet Mask**: Class D does not use a standard subnet mask.
- **Example**: 224.0.0.1 is a Class D address, used for multicast.

Class E

- **Structure**: In a Class E address, the first four bits of the first octet are always '1111'.
- **Range**: Class E addresses range from 240.0.0.0 to 255.255.255.255.
- **Usage**: Class E addresses are reserved for experimental purposes and research; they are not used for general internet traffic.
- **Subnet Mask**: Class E does not use a standard subnet mask.
- **Example**: 250.1.1.1 is a Class E address, reserved for testing or experimental use.

Q7:-Dijkstra Algorithm(10m)

Dijkstra's algorithm is a popular algorithm used to find the shortest path between nodes in a graph.

Dijkstra's Algorithm Steps:

1. **Initialize**:
    - Start with the source (starting) node.
    - Set the distance to the source node as 0 and all other nodes as infinity (∞).
    - Mark all nodes as unvisited.
2. **Visit the Unvisited Node with the Smallest Known Distance**:
    - From the source node, select the unvisited node with the smallest known distance.
3. **Update Distances**:
    - For the current node, calculate the distance to each of its neighboring nodes.
    - If the calculated distance is smaller than the known distance, update it.
4. **Mark the Current Node as Visited**:
    - Mark the current node as visited, meaning it won't be checked again.
5. **Repeat**:
    - Repeat steps 2–4 until all nodes have been visited or the shortest path to the destination node is found.
6. **Construct the Shortest Path** (Optional):

    - To find the shortest path, backtrack from the destination node to the source node by following the nodes with the minimum distance values.
7. **Example of your's**

Q8:-Explain Link State Routing(LSR)(10m)

Link State Routing (LSR) is a strong and efficient routing method used in computer networks.

- **How it works**: LSR uses a link state database (LSDB) to keep track of the status of all links in the network. It then uses Dijkstra's shortest path algorithm to find the best route for data to travel.
- **When it's useful**: LSR is particularly effective in large networks that change frequently, as it doesn't have the "count-to-infinity" issue found in some other routing methods.
- **Drawbacks**: LSR needs more memory and processing power compared to Distance Vector Routing (DVR) and is less scalable for extremely large networks.

To understand the Link State Routing algorithm, here are the three main ideas:

1. **Knowledge about the neighborhood**: Instead of sharing a full routing table, each router only shares information about its neighboring connections. It broadcasts the identity and cost of the directly connected links to other routers.
2. **Flooding**: Each router sends this information to every other router in the network, except its direct neighbors. This process is called "flooding." Every router that receives the packet sends copies to all its neighbors, ensuring that every router eventually gets the same information.
3. **Information sharing**: A router only sends out new information to other routers when there's a change in its link information.

EXAMPLE:refer next pdf

Advantages of the Link State Routing (LSR) Algorithm:

- **Quick Convergence and Adaptability**: LSR only needs to know about the links it is directly connected to, unlike Distance Vector Routing (DVR), which needs to understand the entire network. This allows LSR to quickly adjust to changes in the network, making it very useful in large networks where connections change frequently.
- **No Count-to-Infinity Problem**: LSR does not have the "count-to-infinity" issue found in DVR. In DVR, if two routers have incorrect distance information to a destination, they might keep updating each other indefinitely, causing a loop. With LSR, routers only share information about their direct links, so this problem doesn't occur.

Disadvantages of the Link State Routing (LSR) Algorithm:

- **Higher Memory and Processing Requirements**: LSR needs more memory and processing power compared to DVR. Each router must keep an updated link state database (LSDB) to track changes in the network, which can use a lot of resources.
- **Limited Scalability**: LSR is not as scalable as DVR and can be challenging to implement in very large networks with thousands of routers.

Q9:-ARP(Address resolution protocol) and RARP(Reverse Address resolution protocol) (10m)

The **Address Resolution Protocol (ARP)** is used in computer networks to map IP addresses to physical (MAC) addresses. Here's how it works in simple terms:

1. **Purpose**: Each device on a network has an IP address for identifying it on a larger network (like the Internet) and a MAC address, which is used for sending data on the local network (like Ethernet). Since Ethernet doesn't understand IP addresses, ARP translates (or resolves) an IP address into a MAC address so data can be properly routed on the local network.
2. **Process**:
    - Suppose a computer (let's call it Host A) wants to communicate with another computer (Host B) on the same local network and knows Host B's IP address.

    - Host A sends a broadcast ARP request on the network, asking, "Who has this IP address?"
    - Host B sees this request and replies with its MAC address.
    - Host A can now use this MAC address to send data directly to Host B on the Ethernet network.
3. **Caching**: After learning the MAC address, Host A temporarily stores it in its ARP cache. This way, if Host A needs to communicate with Host B again soon, it doesn't need to send another ARP request.
4. **Proxy ARP**: Sometimes, a router responds to ARP requests on behalf of a device on a different network, allowing a device to appear as if it's on the local network even if it isn't. This is useful in certain network configurations, like for mobile devices.

RARP

RARP is a computer networking protocol that helps a device (like a computer) on a local network find out its IP address by using its physical (MAC) address. This is the reverse of the Address Resolution Protocol (ARP), which finds a device's MAC address based on its IP address. In this article, we'll go over everything about RARP.

What is Reverse Address Resolution Protocol (RARP)?

RARP is a protocol that a device on a local network uses to request its IP address from a central router or server. The network administrator sets up a table on the router, linking each device's MAC address to an IP address. When a new device is connected or if a device does not have memory to store an IP address, it sends out a "RARP request" containing its MAC address in both the sender and receiver fields.

This allows the device to find out its IP address without needing to remember it or know which server will respond to the request. The server with the table of MAC and IP addresses will respond to this request, assigning the IP address to the device. RARP is limited in that it only assigns IP addresses, without offering other services.

How Does RARP Work?

RARP works at the lowest layer of the network, allowing devices to communicate. Each device has two types of addresses: an IP address (which is assigned by software) and a MAC(---moaz faqih—) address (which is built into the hardware). The RARP server in the network has a list of MAC addresses and their assigned IP addresses.

When a device needs an IP address, it sends a broadcast message, meaning it reaches all devices in the network, using its MAC address. The RARP server recognizes the request and responds with the IP address associated with that MAC address.

Disadvantages of RARP

- The RARP server must be on the same local network.
- RARP works on the lowest network layer, so routers cannot forward these requests across different networks.
- RARP cannot handle subnetting since it doesn't send subnet mask information. If a network has multiple subnets, each needs a separate RARP server.
- It doesn't support advanced configurations used in modern networks, like Ethernet.

Q10:-Explain Distance vector routing(10m)

Computer networks typically use complex, efficient dynamic routing algorithms to find the shortest paths for the current network layout. Two popular dynamic algorithms are *distance vector routing* and *link state routing*. Here, we'll explain distance vector routing.

What is Distance Vector Routing?

Distance vector routing works by having each router maintain a table (or vector) that shows the best-known distance to each destination and which link to use to get there. Routers update these tables by exchanging information with their neighbors, so eventually, each router knows the best path to each destination in the network.

Another name for this algorithm is the *distributed Bellman-Ford routing algorithm*, named after its inventors, and it was the original algorithm used in the ARPANET and Internet under the name RIP (Routing Information Protocol).

How Does Distance Vector Routing Work?

In distance vector routing, each router keeps a routing table that lists every other router in the network. Each entry in this table has two parts:

1. The preferred outgoing link to use for that destination.
2. An estimate of the distance to that destination.

The distance can be measured in *hops* (the number of steps between routers) or *delay* (the time it takes for data to travel between routers). The router knows the distance to each of its neighboring routers. If the distance metric is hops, the distance to each neighbor is just one hop. If it's based on delay, the router measures it directly by sending special "echo" packets that return as fast as possible.

For example, if delay is the metric, the router measures the delay to each neighbor. Then, every so often (for example, every T milliseconds), each router sends its estimated delays to each destination to its neighbors and receives similar information back.

An Example of Distance Vector Routing

This updating process is illustrated in Fig. 5-9. Part (a) shows a network. The first four columns of part (b) show the delay vectors received from the neighbors of router *J*. *A* claims to have a 12-msec delay to *B*, a 25-msec delay to *C*, a 40-msec delay to *D*, etc. Suppose that *J* has measured or estimated its delay to its neighbors, *A*, *I*, *H*, and *K*, as 8, 10, 12, and 6 msec, respectively.

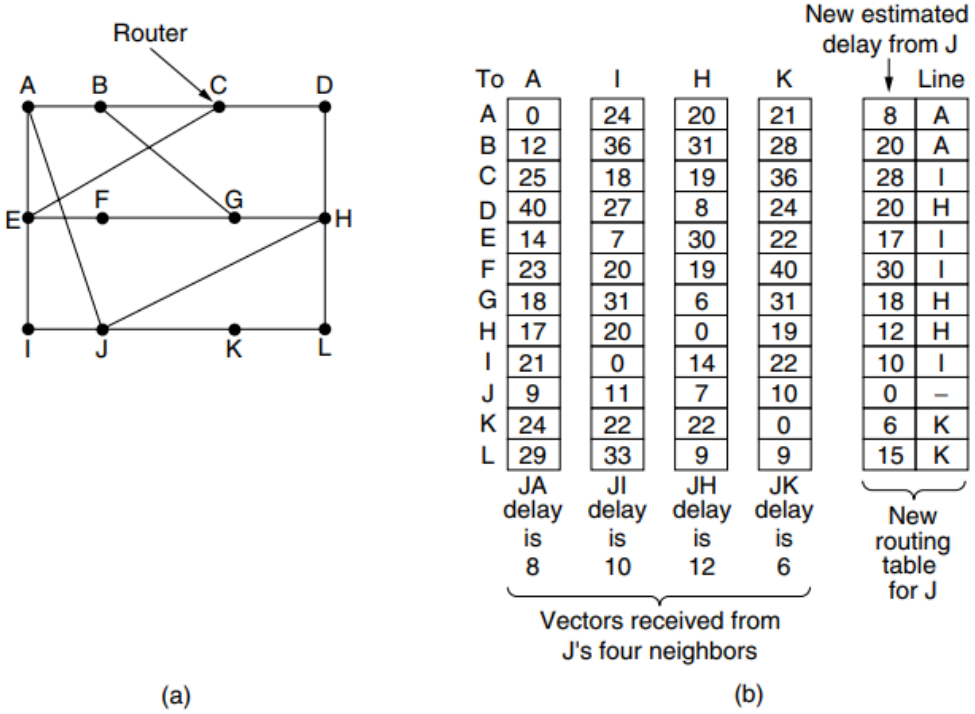


Figure 5-9. (a) A network. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

Consider how *J* computes its new route to router *G*. It knows that it can get to *A* in 8 msec, and furthermore *A* claims to be able to get to *G* in 18 msec, so *J* knows it can count on a delay of 26 msec to *G* if it forwards packets bound for *G*

to *A*. Similarly, it computes the delay to *G* via *I*, *H*, and *K* as 41 (31 + 10), 18 (6 + 12), and 37 (31 + 6) msec, respectively. The best of these values is 18, so it makes an entry in its routing table that the delay to *G* is 18 msec and that the route to use is via *H*. The same calculation is performed for all the other destinations, with the new routing table shown in the last column of the figure.