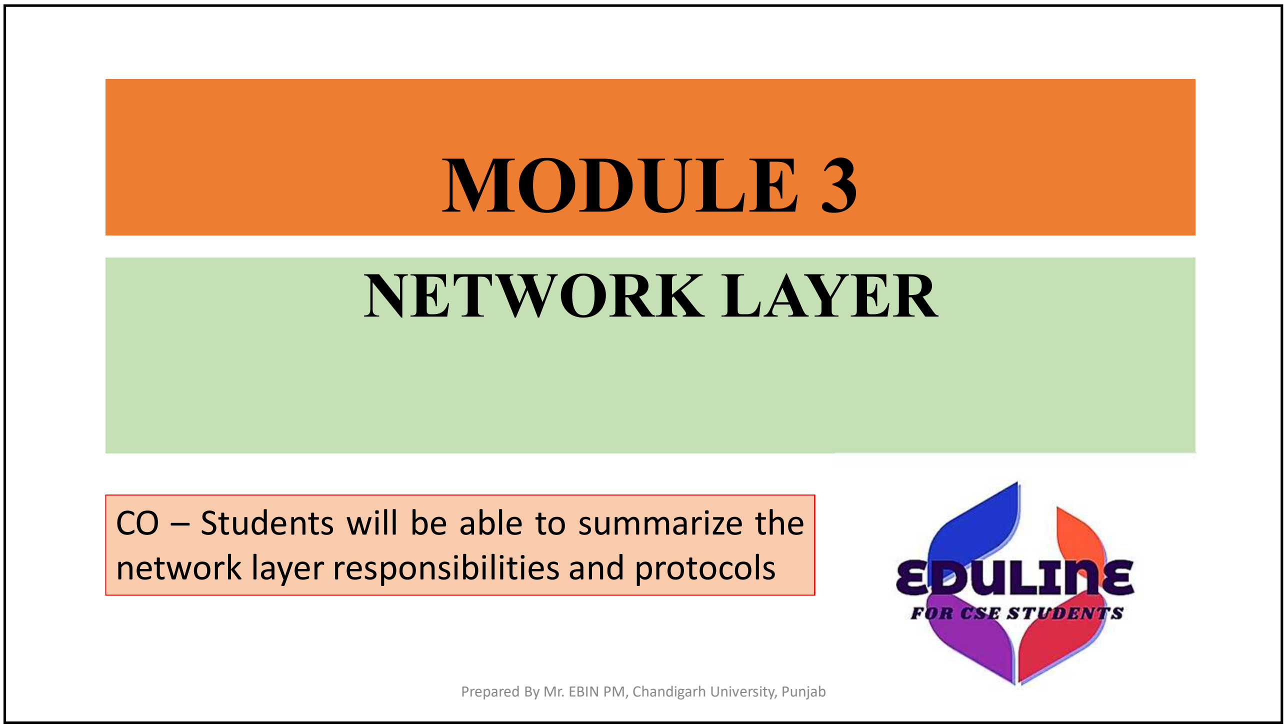
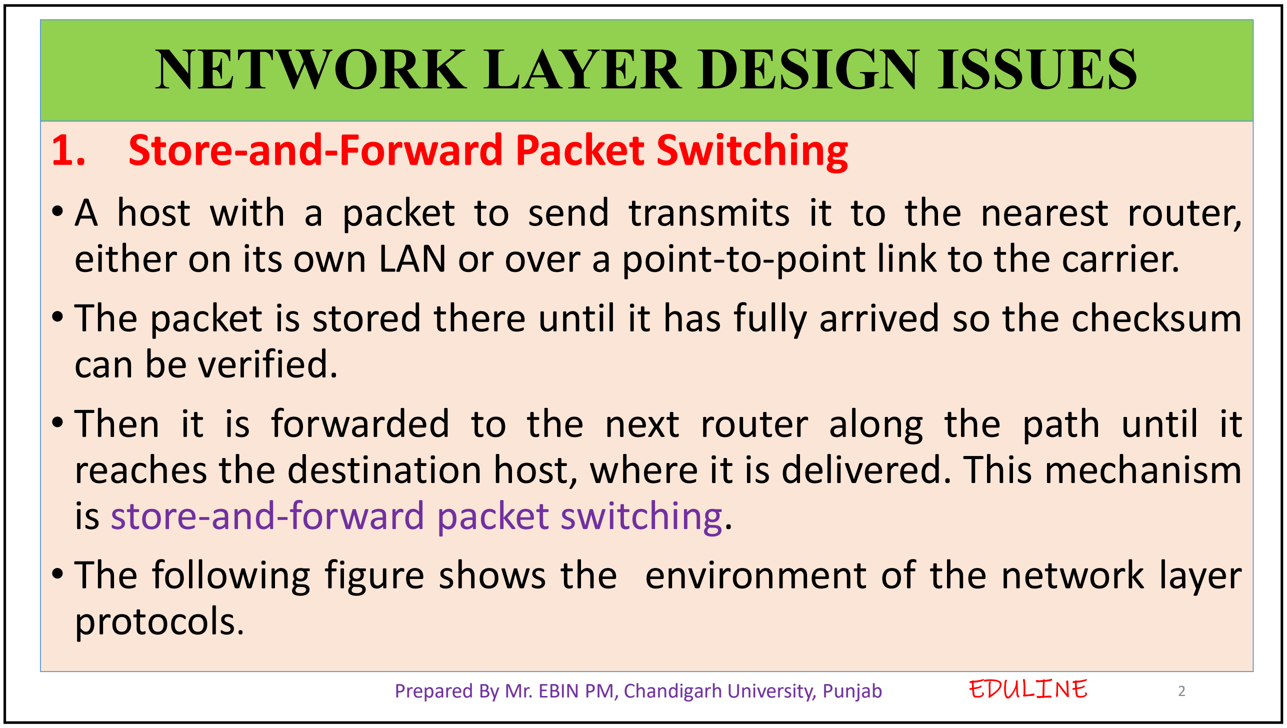


MODULE 3

NETWORK LAYER

CO – Students will be able to summarize the network layer responsibilities and protocols

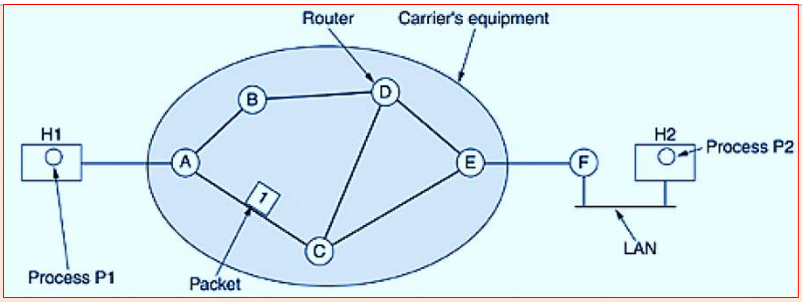
EDULINE FOR CSE STUDENTS



NETWORK LAYER DESIGN ISSUES

1. Store-and-Forward Packet Switching

- A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the carrier.
- The packet is stored there until it has fully arrived so the checksum can be verified.
- Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is store-and-forward packet switching.
- The following figure shows the environment of the network layer protocols.



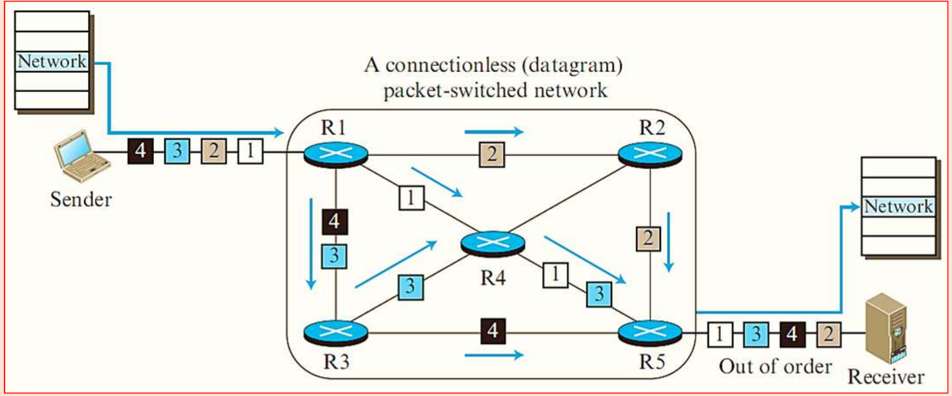
- The major components of the system are the carrier's equipment (routers connected by transmission lines) and the customers' equipment, shown outside the oval.
- Host H1 is directly connected to one of the carrier's routers, A, by a leased line. In contrast, H2 is on a LAN with a router, F, owned and operated by the customer. This router also has a leased line to the carrier's equipment.

- The connecting devices in a packet-switched network still need to decide how to route the packets to the final destination.
- Packet-switched network can use two different approaches to route the packets: the datagram approach and the virtual circuit approach

a) Datagram Approach: Connectionless Service

- The network-layer protocol treats each packet independently, with each packet having no relationship to any other packet.
- The idea was that the network layer is only responsible for delivery of packets from the source to the destination. In this approach, the packets in a message may or may not travel the same path to their destination.

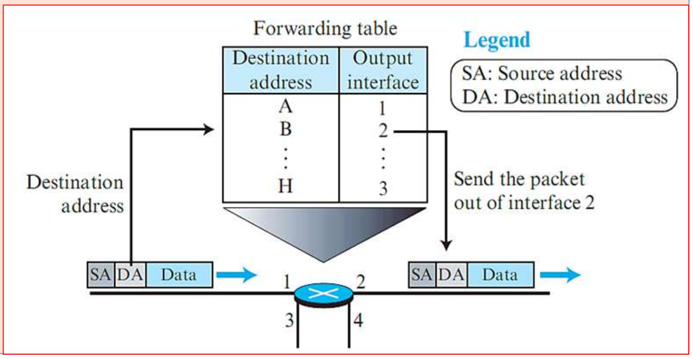
A connectionless packet-switched network



- When the network layer provides a connectionless service, each packet traveling in the Internet is an independent entity; there is no relationship between packets belonging to the same message.
- The switches in this type of network are called routers

- Each packet is routed based on the information contained in its header: source and destination addresses.
- The destination address defines where it should go; the source address defines where it comes from. The router in this case routes the packet based only on the destination address. The source address may be used to send an error message to the source if the packet is discarded

Forwarding process in a router when used in a Datagram approach

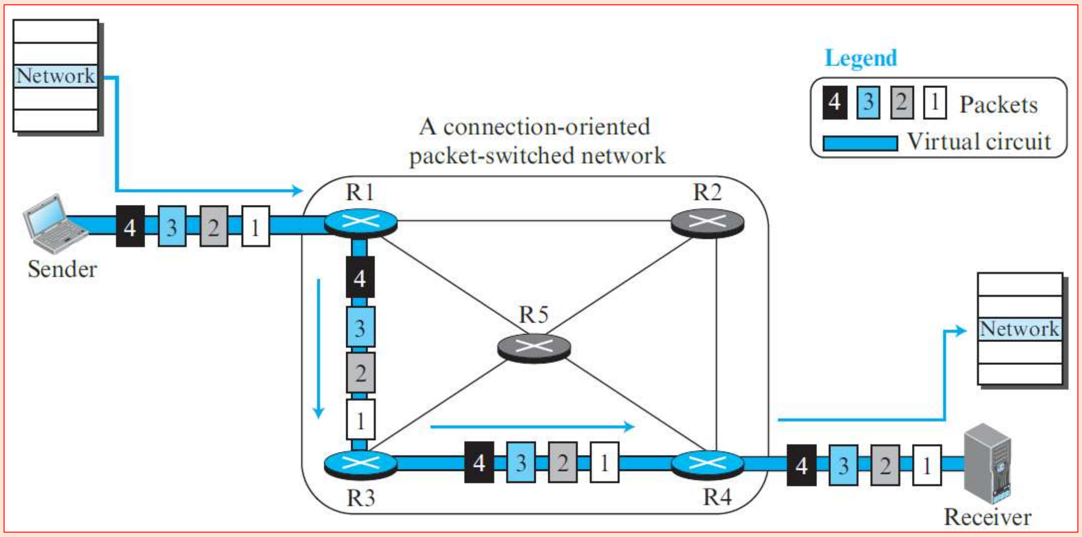

Forwarding table

Destination address	Output interface
A	1
B	2
⋮	⋮
H	3

b) Virtual-Circuit Approach: Connection-Oriented Service

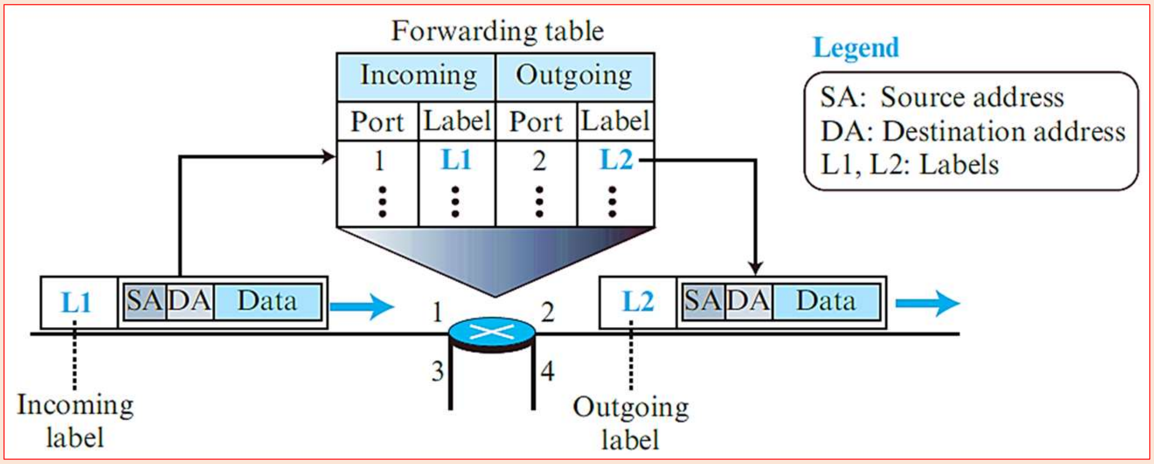
- In a connection-oriented service (also called virtual-circuit approach), there is a relationship between all packets belonging to a message.
- Before all datagrams in a message can be sent, a virtual connection should be set up to define the path for the datagrams.
- After connection setup, the datagrams can all follow the same path.
- In this type of service, not only must the packet contain the source and destination addresses, it must also contain a **flow label**, **a virtual circuit identifier (VCI)** that defines the virtual path the packet should follow.

A virtual-circuit packet-switched network



- Each packet is forwarded based on the label in the packet

Forwarding process in a router when used in a virtual-circuit network



Incoming		Outgoing	
Port	Label	Port	Label
1	L1	2	L2
⋮	⋮	⋮	⋮

- To create a connection-oriented service, a three-phase process is used:

1. **Setup**
2. **Data transfer**
3. **Teardown**

- In the setup phase, the source and destination addresses of the sender and receiver are used to make table entries for the connection-oriented service.
- In the teardown phase, the source and destination inform the router to delete the corresponding entries.
- Data transfer occurs between these two phases.

2. Services Provided to the Transport Layer

➢The network layer services have been designed with the following goals in mind.

1. The services should be independent of the router technology.
2. The transport layer should be shielded from the number, type, and topology of the routers present.
3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

➢Connection Less & Connection oriented service

- If connectionless service is offered, packets are injected into the subnet individually and routed independently of each other. No advance setup is needed. In this context, the packets are frequently called datagrams and the subnet is called a datagram subnet.
- If connection-oriented service is used, a path from the source router to the destination router must be established before any data packets can be sent. This connection is called a VC (virtual circuit), and the subnet is called a virtual-circuit subnet.
- The Internet offers connectionless network-layer service
- ATM networks offer connection-oriented network-layer service.

NETWORK LAYER SERVICES

1. Packetizing

- Packetizing means encapsulating the payload (data received from upper layer) in a network-layer packet at the source and decapsulating the payload from the network-layer packet at the destination.
- In other words, one duty of the network layer is to carry a payload from the source to the destination without changing it or using it.
- The routers are not allowed to change source and destination addresses either. They just inspect the addresses for the purpose of forwarding the packet to the next network on the path.

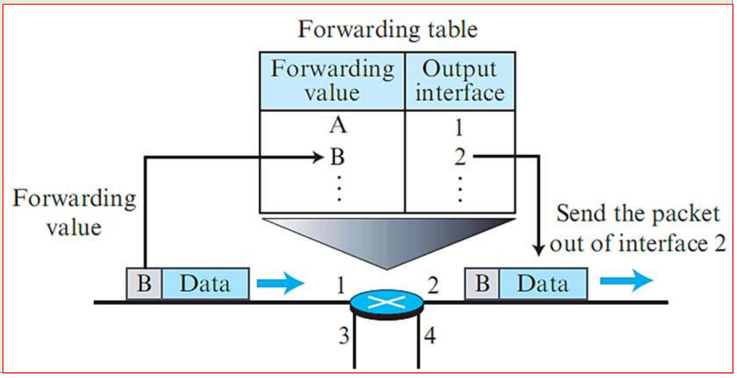
2. Routing

- The network layer is responsible for routing the packet from its source to the destination.
- There is more than one route from the source to the destination. The network layer is responsible for finding the best one among these possible routes.
- The network layer needs to have some specific strategies for defining the best route.
- In the Internet today, this is done by running some routing protocols to help the routers coordinate their knowledge about the neighborhood and to come up with consistent tables to be used when a packet arrives.

3. Forwarding

- Forwarding can be defined as the action applied by each router when a packet arrives at one of its interfaces.
- The decision-making table a router normally uses for applying this action is sometimes called the forwarding table and sometimes the routing table.

Forwarding process



4. Error Control

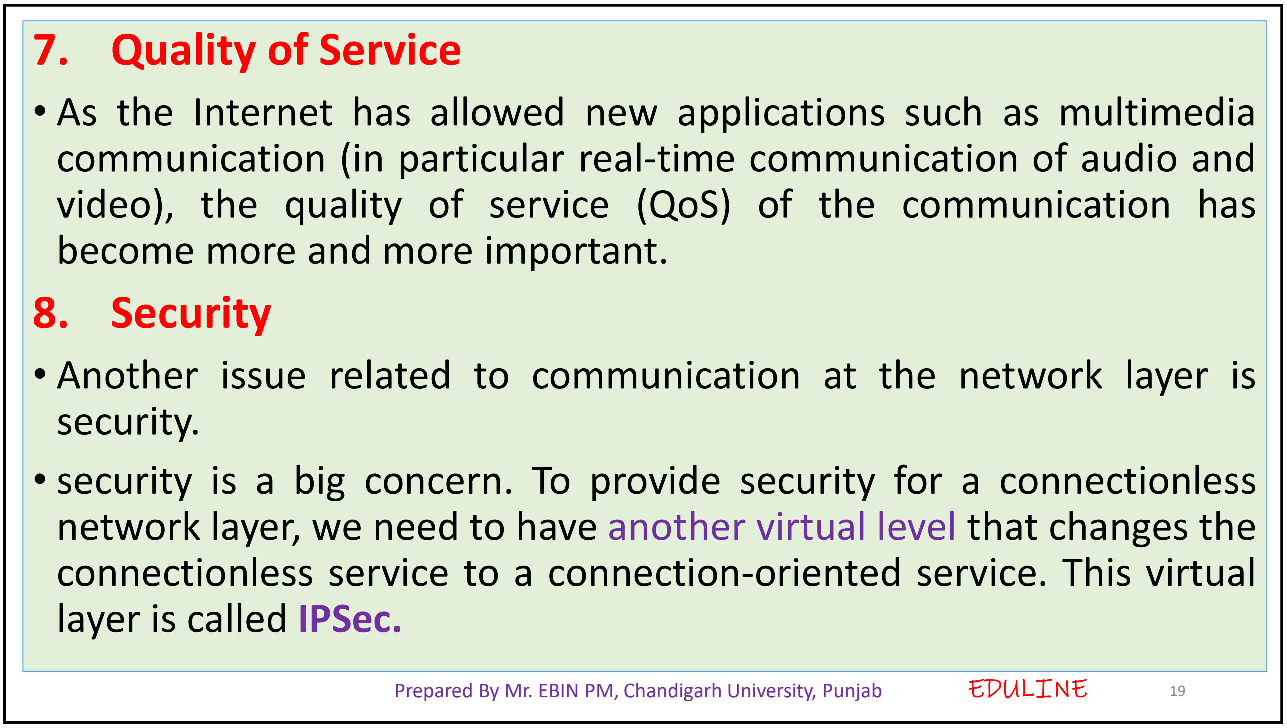
- The designers of the network layer have added a checksum field to the datagram to control any corruption in the header, but not the whole datagram.
- This checksum may prevent any changes or corruptions in the header of the datagram between two hops and from end to end
- The network layer in the Internet does not directly provide error control, the Internet uses an auxiliary protocol, ICMP, that provides some kind of error control if the datagram is discarded or has some unknown information in the header.

5. Flow Control

- Flow control regulates the amount of data a source can send without overwhelming the receiver.
- If the upper layer at the source computer produces data faster than the upper layer at the destination computer can consume it, the receiver will be overwhelmed with data.
- To control the flow of data, the receiver needs to send some feedback to the sender to inform the latter that it is overwhelmed with data.

6. Congestion Control

- Congestion in the network layer is a situation in which too many datagrams are present in an area of the Internet.
- Congestion may occur if the number of datagrams sent by source computers is beyond the capacity of the network or routers. In this situation, some routers may drop some of the datagrams.
- However, as more datagrams are dropped, the situation may become worse because, due to the error control mechanism at the upper layers, the sender may send duplicates of the lost packets.

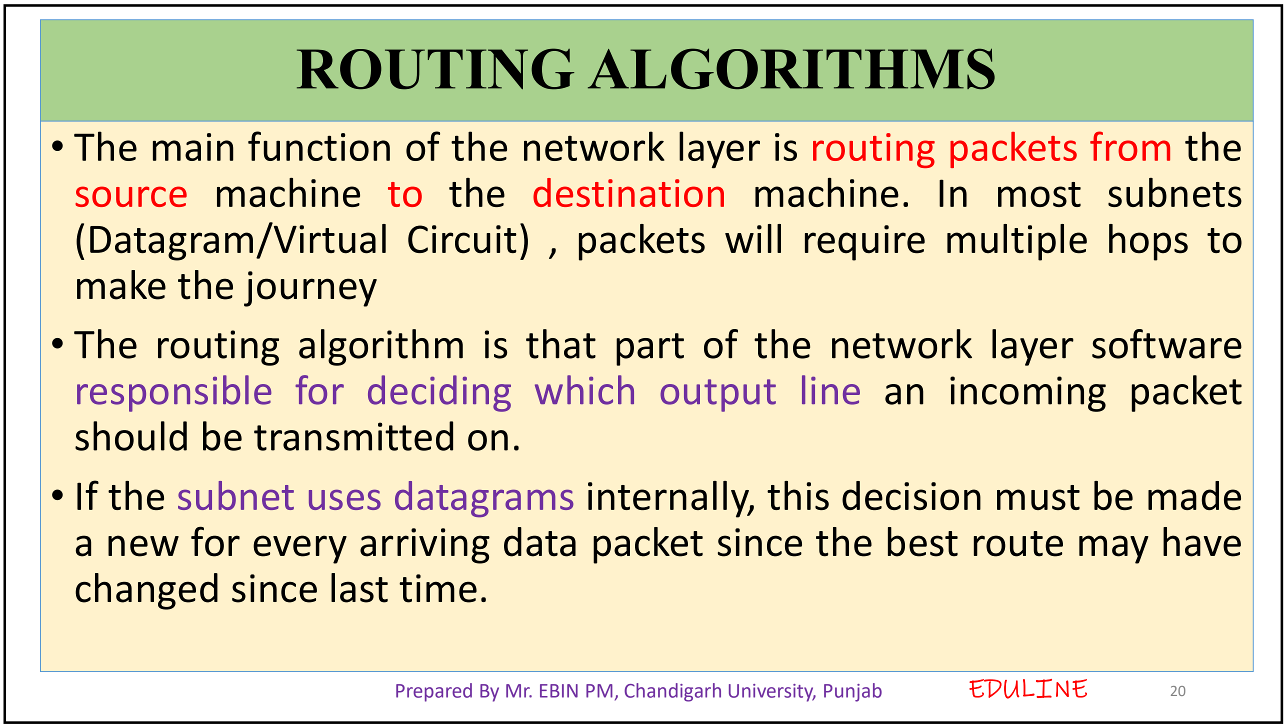


7. Quality of Service

- As the Internet has allowed new applications such as multimedia communication (in particular real-time communication of audio and video), the quality of service (QoS) of the communication has become more and more important.

8. Security

- Another issue related to communication at the network layer is security.
- security is a big concern. To provide security for a connectionless network layer, we need to have another virtual level that changes the connectionless service to a connection-oriented service. This virtual layer is called **IPSec.**



ROUTING ALGORITHMS

- The main function of the network layer is routing packets from the source machine to the destination machine. In most subnets (Datagram/Virtual Circuit) , packets will require multiple hops to make the journey
- The routing algorithm is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.
- If the subnet uses datagrams internally, this decision must be made a new for every arriving data packet since the best route may have changed since last time.

- If the subnet uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up. Thereafter, data packets just follow the previously-established route. This is sometimes called session routing because a route remains in force for an entire user session.
- A router is having two processes inside it.
- One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing tables. This process is forwarding.
- The other process is responsible for filling in and updating the routing tables. That is where the routing algorithm comes into play.

- Certain properties are desirable in a routing algorithm: correctness, simplicity, robustness, stability, fairness, and optimality.
- The routing algorithm should be able to cope with changes in the topology and traffic without requiring all jobs in all hosts to be aborted and the network to be rebooted every time some router crashes.

➢Routing algorithms can be grouped into two major classes:

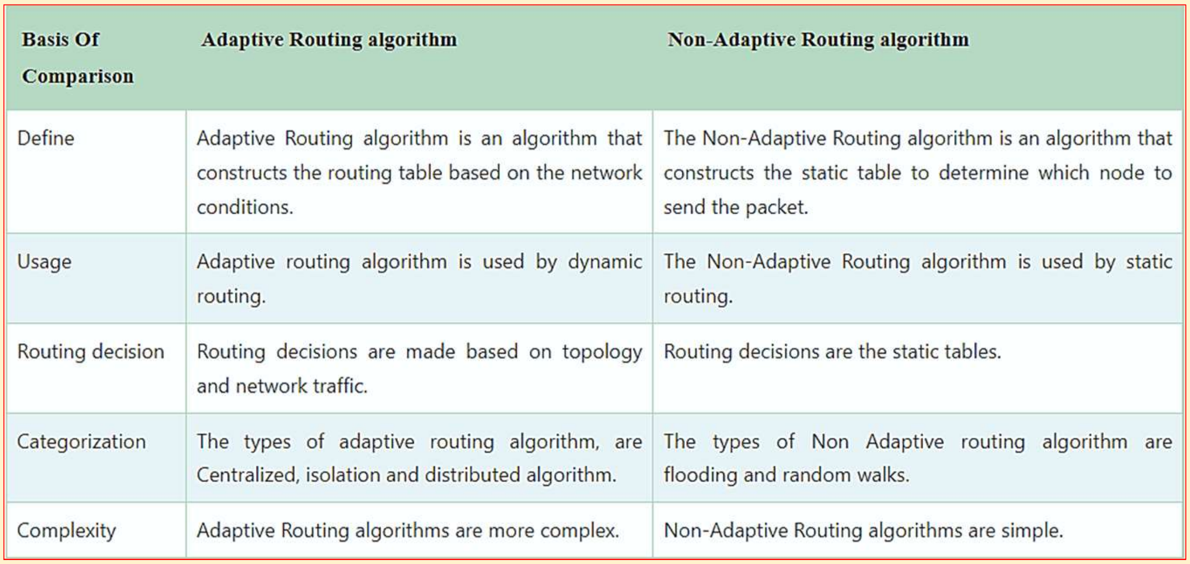
1. **Nonadaptive**
2. **Adaptive.**

➢Adaptive Routing algorithm

- An adaptive routing algorithm is also known as dynamic routing algorithm.
- This algorithm makes the routing decisions based on the topology and network traffic.
- The main parameters related to this algorithm are hop count, distance and estimated transit time.

➢Non-Adaptive Routing algorithm

- Non Adaptive routing algorithm is also known as a static routing algorithm.
- When booting up the network, the routing information stores to the routers.
- Non Adaptive routing algorithms do not take the routing decision based on the network topology or network traffic.



Basis Of Comparison	Adaptive Routing algorithm	Non-Adaptive Routing algorithm
Define	Adaptive Routing algorithm is an algorithm that constructs the routing table based on the network conditions.	The Non-Adaptive Routing algorithm is an algorithm that constructs the static table to determine which node to send the packet.
Usage	Adaptive routing algorithm is used by dynamic routing.	The Non-Adaptive Routing algorithm is used by static routing.
Routing decision	Routing decisions are made based on topology and network traffic.	Routing decisions are the static tables.
Categorization	The types of adaptive routing algorithm, are Centralized, isolation and distributed algorithm.	The types of Non Adaptive routing algorithm are flooding and random walks.
Complexity	Adaptive Routing algorithms are more complex.	Non-Adaptive Routing algorithms are simple.

OPTIMALITY PRINCIPLE

- The purpose of a routing algorithm at a router is to decide which output line an incoming packet should go. The optimal path from a particular router to another may be the least cost path, the least distance path, the least time path, the least hops path or a combination of any of the above.
- The optimality principle can be logically proved as follows –

➢ If a better route could be found between router J and router K, the path from router I to router K via J would be updated via this route. Thus, the optimal path from J to K will again lie on the optimal path from I to K.

SHORTEST PATH ROUTING

- It is one of the simple static routing algorithms that are widely used for routing in the network.
- The basic idea of it is to build a graph with each node representing a router and each line representing a communication link.
- To choose a route between any two nodes in the graph the algorithm simply finds the shortest path between the nodes.
- Shortest Path means that the path in which anyone or more metrics is minimized. The metric may be distance, bandwidth, average traffic, communication cost, mean queue length, measured delay or any other factor

❖Common Shortest Path Algorithms

➢Bellman Ford's Algorithm

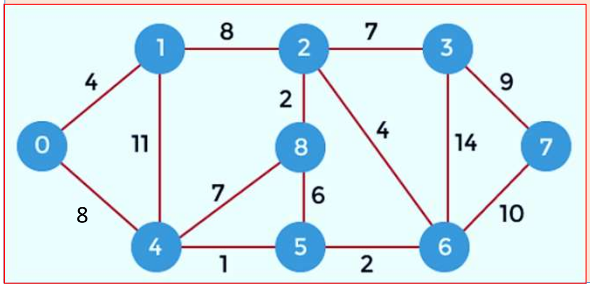
➢Dijkstra's Algorithm

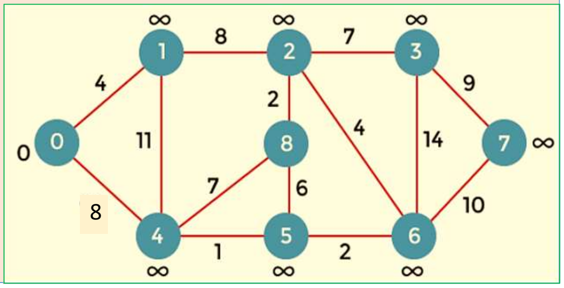
➢Floyd Warshall's Algorithm

Dijkstra's Algorithm

- Dijkstra's algorithm is a single-source shortest path algorithm.
- Here, single-source means that only one source is given, and we have to find the shortest path from the source to all the nodes.
- Let's understand the working of Dijkstra's algorithm. Consider the below graph.

- First, we have to consider any vertex as a source vertex.
- Here we assume that 0 as a source vertex, and distance to all the other vertices is infinity.
- Initially, we do not know the distances. First, we will find out the vertices which are directly connected to the vertex 0.
- As we can observe in the below graph that two vertices are directly connected to vertex 0.

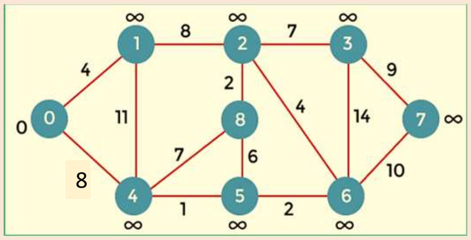




❖The formula for calculating the distance between the vertices:

if($d(u) + c(u, v) < d(v)$) Then

$d(v) = d(u) + c(u, v)$



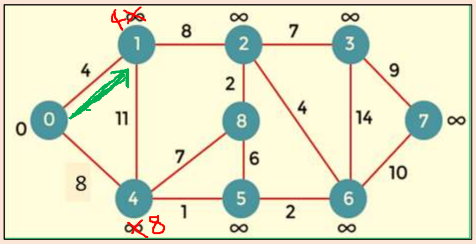
- Let's assume that the vertex 0 is represented by 'x' and the vertex 1 is represented by 'y'. The distance between the vertices can be calculated by using the below formula:

$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (0 + 4) < \infty$$
$$= 4 < \infty$$

Since 4<∞ so we will update d(y) from ∞ to 4.

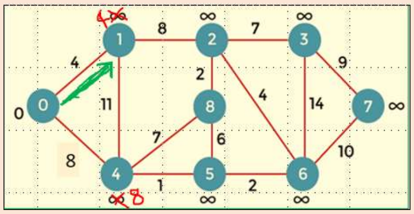
Now we consider vertex 0 same as 'x' and vertex 4 as 'y'.

$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (0 + 8) < \infty$$
$$= 8 < \infty$$



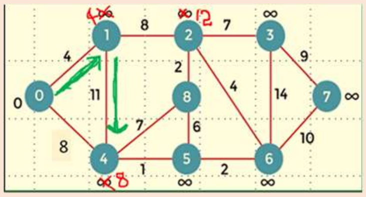
- Therefore, the value of d(y) is 8. We replace the infinity value of vertices 1 and 4 with the values 4 and 8 respectively.
- Now, we have found the shortest path from the vertex 0 to 1 and 0 to 4. Therefore, vertex 0 is selected. Now, we will compare all the vertices except the vertex 0.
- Since vertex 1 has the lowest value, i.e., 4; therefore, vertex 1 is selected.

- Since vertex 1 is selected, so we consider the path from 1 to 2, and 1 to 4. First, we calculate the distance between the vertex 1 and 2. Consider the vertex 1 as 'x', and the vertex 2 as 'y'.



$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (4 + 8) < \infty$$
$$= 12 < \infty$$

- Since 12<∞ so we will update d(2) from ∞ to 12.
- Now, we calculate the distance between the vertex 1 and vertex 4. Consider the vertex 1 as 'x' and the vertex 4 as 'y'.



$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (4 + 11) < 8$$
$$= 15 < 8$$

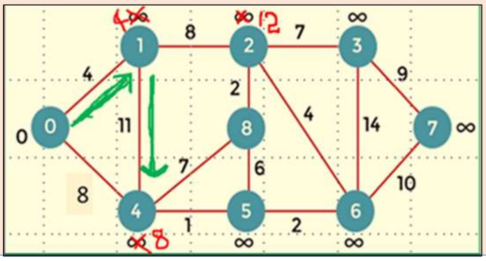
Since 15 is not less than 8, we will not update the value d(4) from 8 to 15.

- Till now, two nodes have been selected, i.e., 0 and 1.
- Now we have to compare the nodes except the node 0 and 1.
- The node 4 has the minimum distance, i.e., 8. Therefore, vertex 4 is selected.
- Since vertex 4 is selected, so we will consider all the direct paths from the vertex 4.
- The direct paths from vertex 4 are 4 to 0, 4 to 1, 4 to 8, and 4 to 5.
- Since the vertices 0 and 1 have already been selected so we will not consider the vertices 0 and 1.
- We will consider only two vertices, i.e., 8 and 5.

- First, we consider the vertex 8. First, we calculate the distance between the vertex 4 and 8.
- Consider the vertex 4 as 'x', and the vertex 8 as 'y'.

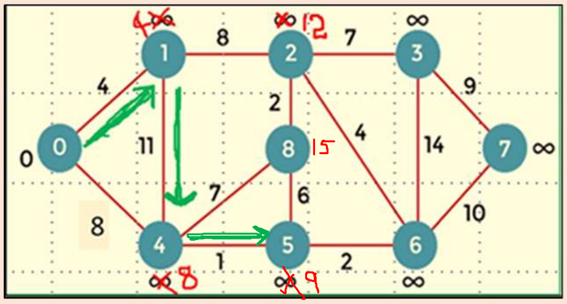
$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (8 + 7) < \infty$$
$$= 15 < \infty$$

- Since 15 is less than the infinity so we update d(8) from infinity to 15.



- Now, we consider the vertex 5. First, we calculate the distance between the vertex 4 and 5.
- Consider the vertex 4 as 'x', and the vertex 5 as 'y'.

$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (8 + 1) < \infty$$
$$= 9 < \infty$$



- Since 5 is less than the infinity, we update d(5) from infinity to 9.
- The node 5 has the minimum value, i.e., 9. Therefore, vertex 5 is selected.

- Since the vertex 5 is selected, so we will consider all the direct paths from vertex 5. The direct paths from vertex 5 are 5 to 8, and 5 to 6.
- First, we consider the vertex 8. First, we calculate the distance between the vertex 5 and 8. Consider the vertex 5 as 'x', and the vertex 8 as 'y'.

$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (9 + 15) < 15$$
$$= 24 < 15$$

- Since 24 is not less than 15 so we will not update the value d(8) from 15 to 24.

- Now, we consider the vertex 6. First, we calculate the distance between the vertex 5 and 6. Consider the vertex 5 as 'x', and the vertex 6 as 'y'.

$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (9 + 2) < \infty$$
$$= 11 < \infty$$

- Since 11 is less than infinity, we update d(6) from infinity to 11
- Till now, nodes 0, 1, 4 and 5 have been selected.
- We will compare the nodes except the selected nodes.
- The node 6 has the lowest value as compared to other nodes. Therefore, vertex 6 is selected.

- Since vertex 6 is selected, we consider all the direct paths from vertex 6. The direct paths from vertex 6 are 6 to 2, 6 to 3, and 6 to 7.
- First, we consider the vertex 2. Consider the vertex 6 as 'x', and the vertex 2 as 'y'.

$$d(x, y) = d(x) + c(x, y) < d(y)$$

$$= (11 + 4) < 12$$

$$= 15 < 12$$

- Since 15 is not less than 12, we will not update d(2) from 12 to 15

- Now we consider the vertex 3. Consider the vertex 6 as 'x', and the vertex 3 as 'y'.

$$d(x, y) = d(x) + c(x, y) < d(y)$$

$$= (11 + 14) < \infty$$

$$= 25 < \infty$$

- Since 25 is less than ∞, so we will update d(3) from ∞ to 25.
- Now we consider the vertex 7. Consider the vertex 6 as 'x', and the vertex 7 as 'y'.

$$d(x, y) = d(x) + c(x, y) < d(y)$$

$$= (11 + 10) < \infty$$

$$= 22 < \infty$$

- Since 22 is less than ∞ so, we will update d(7) from ∞ to 22.

- Till now, nodes 0, 1, 4, 5, and 6 have been selected. Now we have to compare all the unvisited nodes, i.e., 2, 3, 7, and 8. Since node 2 has the minimum value, i.e., 12 among all the other unvisited nodes. Therefore, node 2 is selected.
- Since node 2 is selected, so we consider all the direct paths from node 2. The direct paths from node 2 are 2 to 8, 2 to 6, and 2 to 3.
- First, we consider the vertex 8. Consider the vertex 2 as 'x' and 8 as 'y'.

$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (12 + 2) < 15$$
$$= 14 < 15$$

- Since 14 is less than 15, we will update d(8) from 15 to 14.

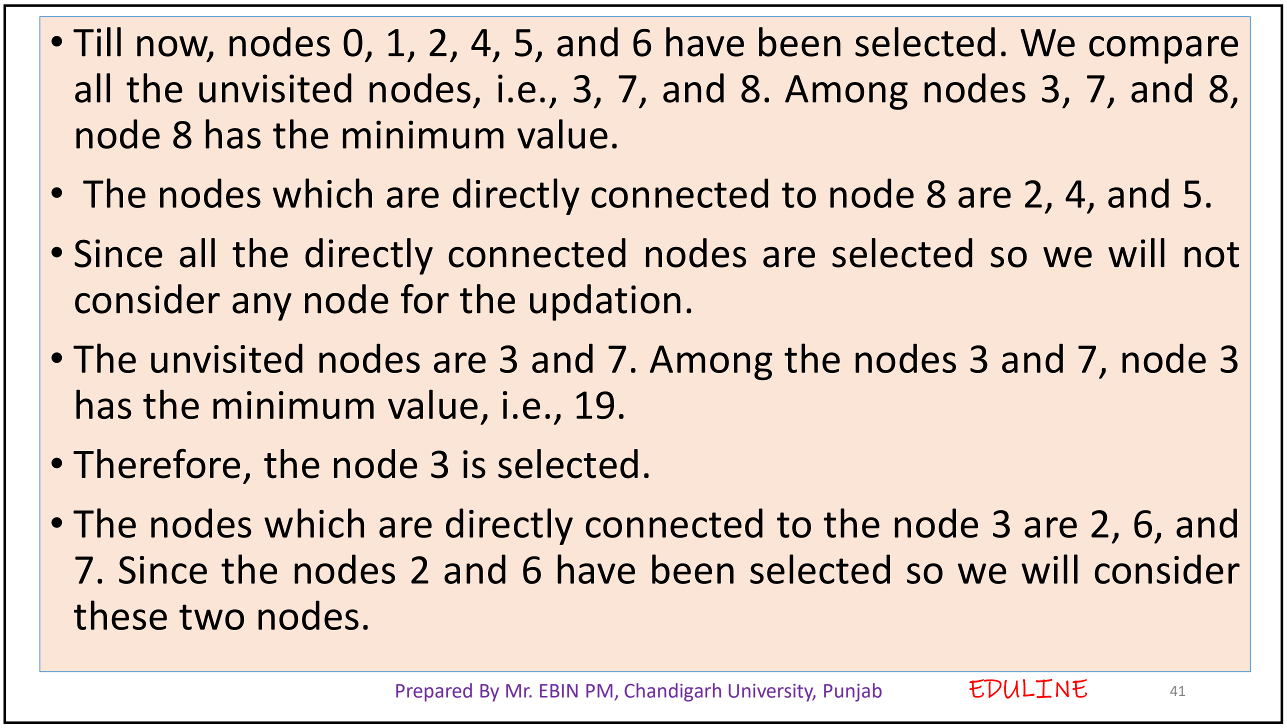
- Now, we consider the vertex 6. Consider the vertex 2 as 'x' and 6 as 'y'.

$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (12 + 4) < 11$$
$$= 16 < 11$$

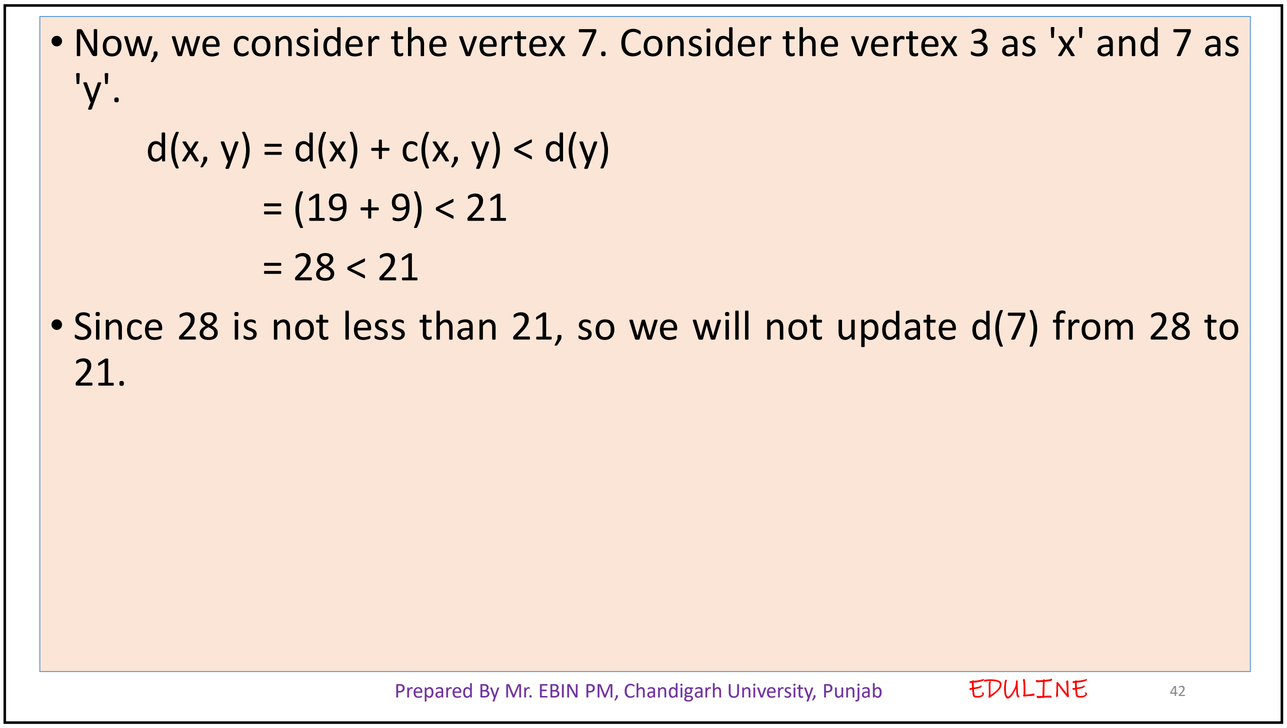
- Since 16 is not less than 11 so we will not update d(6) from 11 to 16.
- Now, we consider the vertex 3. Consider the vertex 2 as 'x' and 3 as 'y'.

$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (12 + 7) < 25$$
$$= 19 < 25$$

- Since 19 is less than 25, we will update d(3) from 25 to 19.



- Till now, nodes 0, 1, 2, 4, 5, and 6 have been selected. We compare all the unvisited nodes, i.e., 3, 7, and 8. Among nodes 3, 7, and 8, node 8 has the minimum value.
- The nodes which are directly connected to node 8 are 2, 4, and 5.
- Since all the directly connected nodes are selected so we will not consider any node for the updation.
- The unvisited nodes are 3 and 7. Among the nodes 3 and 7, node 3 has the minimum value, i.e., 19.
- Therefore, the node 3 is selected.
- The nodes which are directly connected to the node 3 are 2, 6, and 7. Since the nodes 2 and 6 have been selected so we will consider these two nodes.



- Now, we consider the vertex 7. Consider the vertex 3 as 'x' and 7 as 'y'.

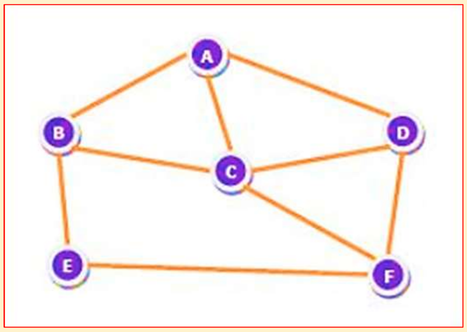
$$d(x, y) = d(x) + c(x, y) < d(y)$$
$$= (19 + 9) < 21$$
$$= 28 < 21$$

- Since 28 is not less than 21, so we will not update d(7) from 28 to 21.

FLOODING

- Flooding is the static routing algorithm. In this algorithm, every incoming packet is sent on all outgoing lines except the line on which it has arrived
- Flooding is a way to distribute routing information updates quickly to every node in a large network. It is also sometimes used in multicast packets.
- Flooding, which is similar to broadcasting, occurs when source packets (without routing data) are transmitted to all attached network nodes. Because flooding uses every path in the network, the shortest path is also used.
- The flooding algorithm is easy to implement.

- When a packet is received, the routers send it to all the interfaces except the one on which it was received. This creates too much burden on the network and lots of duplicate packets wandering in the network.
- Requires no network information like topology, load condition, cost of different paths



❖Types of Flooding

- **Uncontrolled flooding** – Here, each router unconditionally transmits the incoming data packets to all its neighbours.
- **Controlled flooding** – They use some methods to control the transmission of packets to the neighbouring nodes. The two popular algorithms for controlled flooding are Sequence Number Controlled Flooding (SNCF) and Reverse Path Forwarding (RPF).
- **Selective flooding** – Here, the routers don't transmit the incoming packets only along those paths which are heading towards approximately in the right direction, instead of every available paths.

❖Advantages of Flooding

- It is very simple to setup and implement, since a router may know only its neighbours.
- It is extremely robust. Even in case of malfunctioning of a large number routers, the packets find a way to reach the destination.
- All nodes which are directly or indirectly connected are visited. So, there are no chances for any node to be left out. This is a main criteria in case of broadcast messages.
- The shortest path is always chosen by flooding.

➢Limitations of Flooding

- Flooding tends to create an infinite number of duplicate data packets, unless some measures are adopted to damp packet generation.
- It is wasteful if a single destination needs the packet, since it delivers the data packet to all nodes irrespective of the destination.
- The network may be clogged with unwanted and duplicate data packets. This may hamper delivery of other data packets.

DISTANCE VECTOR ROUTING

- The Distance vector algorithm is a dynamic algorithm.
- It is also called Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm
- It is mainly used in ARPANET, and RIP.
- Each router maintains a distance table known as Vector.
- Each node receives information from one or more of its directly attached neighbors, performs calculation and then distributes the result back to its neighbors.
- Information sharing at regular intervals - Within 30 seconds, the router sends the information to the neighboring routers.

❖Distance Vector (DV) Algorithm

1. A router transmits its distance vector to each of its neighbors in a routing packet.
2. Each router receives and saves the most recently received distance vector from each of its neighbors.
3. A router recalculates its distance vector when:

- It receives a distance vector from a neighbor containing different information than before.
- It discovers that a link to a neighbor has gone down.

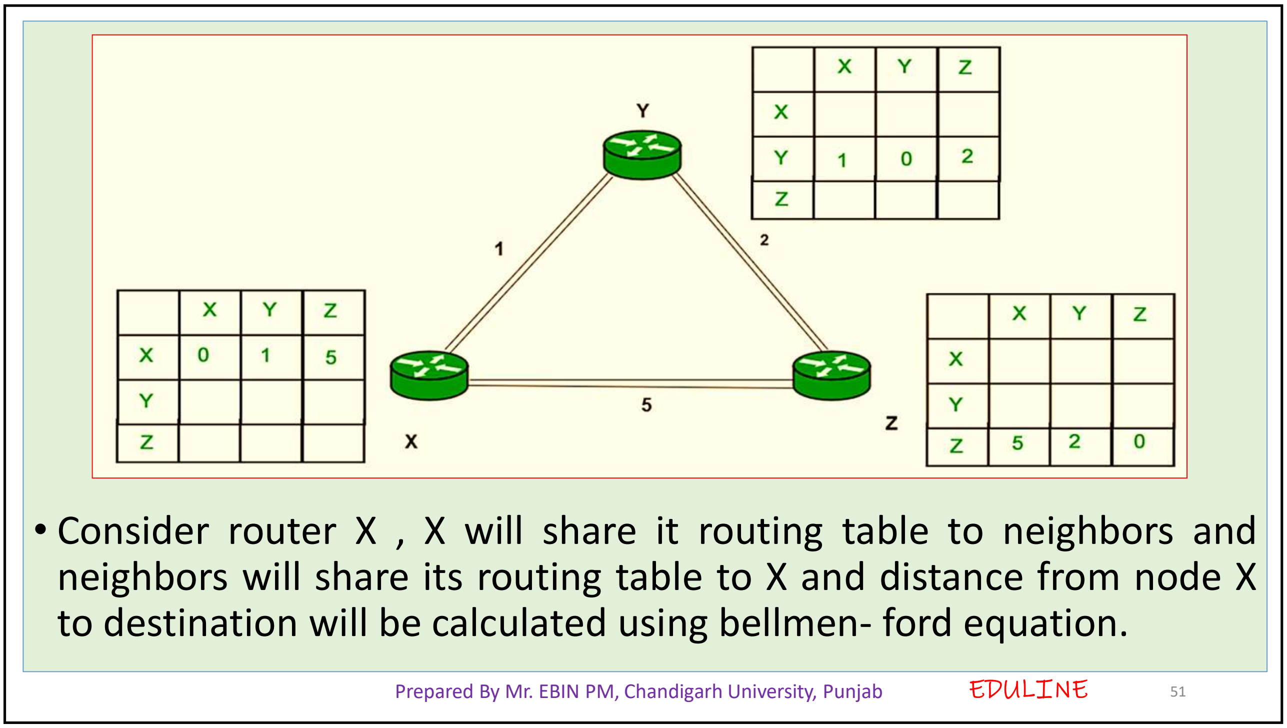
➢The DV calculation is based on minimizing the cost to each destination

- From time-to-time, each node sends its own distance vector estimate to neighbors.
- When a node x receives new DV estimate from any neighbor v, it saves v's distance vector and it updates its own DV using B-F equation:

$$D_x(y) = \min \{ C(x,v) + D_v(y), D_x(y) \} \text{ for each node } y \in N$$

Example

- Consider 3-routers X, Y and Z as shown in figure. Each router have their routing table. Every routing table will contain distance to the destination nodes.

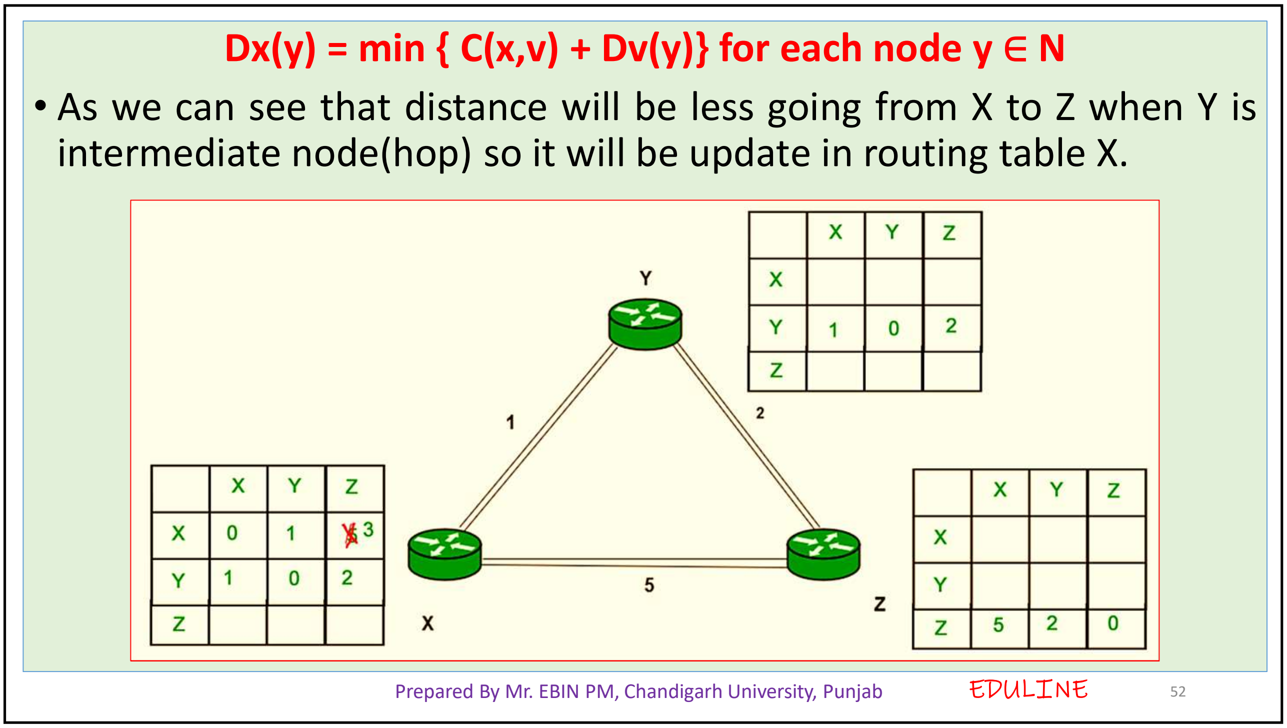


	X	Y	Z
X			
Y	1	0	2
Z			

	X	Y	Z
X	0	1	5
Y			
Z			

	X	Y	Z
X			
Y			
Z	5	2	0

- Consider router X , X will share it routing table to neighbors and neighbors will share its routing table to X and distance from node X to destination will be calculated using bellmen- ford equation.



Dx(y) = min { C(x,v) + Dv(y)} for each node y ∈ N

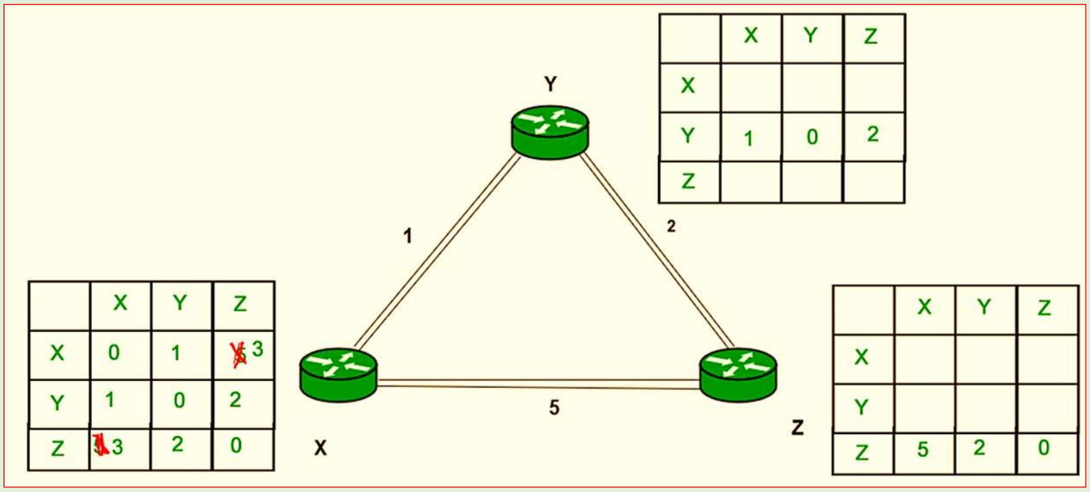
- As we can see that distance will be less going from X to Z when Y is intermediate node(hop) so it will be update in routing table X.

	X	Y	Z
X			
Y	1	0	2
Z			

	X	Y	Z
X	0	1	~~5~~ 3
Y	1	0	2
Z			

	X	Y	Z
X			
Y			
Z	5	2	0

Similarly for Z also



Y:

	X	Y	Z
X			
Y	1	0	2
Z			

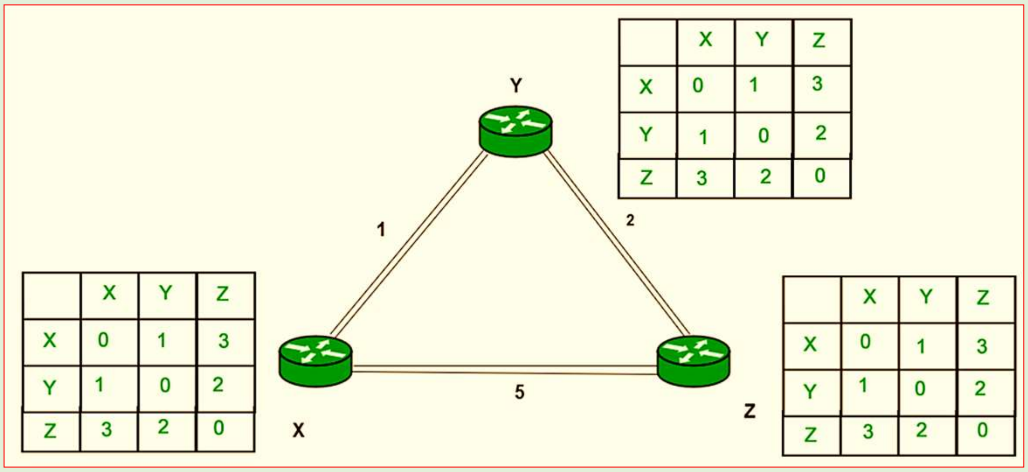
X:

	X	Y	Z
X	0	1	~~5~~ 3
Y	1	0	2
Z	~~5~~ 3	2	0

Z:

	X	Y	Z
X			
Y			
Z	5	2	0

Finally the routing table for all



Y:

	X	Y	Z
X	0	1	3
Y	1	0	2
Z	3	2	0

X:

	X	Y	Z
X	0	1	3
Y	1	0	2
Z	3	2	0

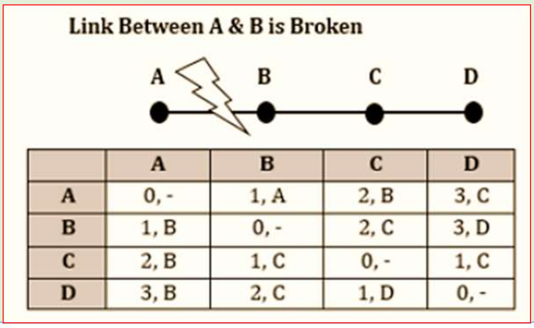
Z:

	X	Y	Z
X	0	1	3
Y	1	0	2
Z	3	2	0

- Distance Vector routing uses UDP(User datagram protocol) for transportation

COUNT TO INFINITY PROBLEM

- Counting to infinity is just another name for a routing loop.
- In distance vector routing, routing loops usually occur when an interface goes down.
- It can also occur when two routers send updates to each other at the same time.

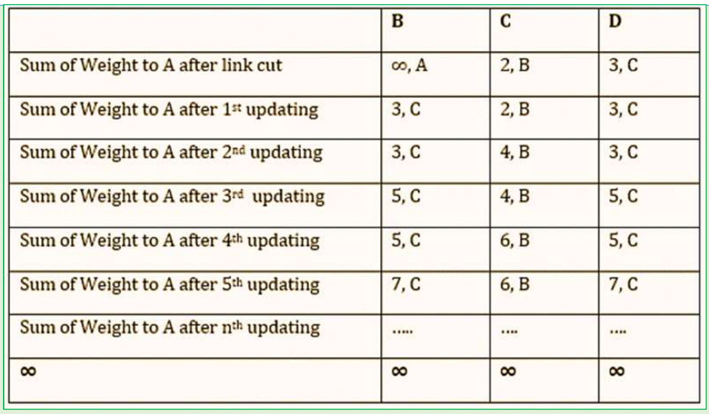


Link Between A & B is Broken

	A	B	C	D
A	0, -	1, A	2, B	3, C
B	1, B	0, -	2, C	3, D
C	2, B	1, C	0, -	1, C
D	3, B	2, C	1, D	0, -

- Imagine a network with a graph as shown above in figure .As you see in this graph, there is only one link between A and the other parts of the network.
- Now imagine that the link between A and B is cut. At this time, B corrects its table.
- After a specific amount of time, routers exchange their tables, and so B receives C's routing table.
- Since C doesn't know what has happened to the link between A and B, it says that it has a link to A with the weight of 2 (1 for C to B, and 1 for B to A -- it doesn't know B has no link to A).
- B receives this table and thinks there is a separate link between C and A, so it corrects its table and changes infinity to 3 (1 for B to C, and 2 for C to A, as C said).

- Once again, routers exchange their tables.
- When C receives B's routing table, it sees that B has changed the weight of its link to A from 1 to 3, so C updates its table and changes the weight of the link to A to 4 (1 for C to B, and 3 for B to A, as B said).
- This process loops until all nodes find out that the weight of link to A is infinity.
- This situation is shown in the table below.
- In this way, Distance Vector Algorithms have a slow convergence rate.



	B	C	D
Sum of Weight to A after link cut	∞, A	2, B	3, C
Sum of Weight to A after 1st updating	3, C	2, B	3, C
Sum of Weight to A after 2nd updating	3, C	4, B	3, C
Sum of Weight to A after 3rd updating	5, C	4, B	5, C
Sum of Weight to A after 4th updating	5, C	6, B	5, C
Sum of Weight to A after 5th updating	7, C	6, B	7, C
Sum of Weight to A after nth updating	.....	....	....
∞	∞	∞	∞

- One way to solve this problem is for routers to send information only to the neighbors that are not exclusive links to the destination.
- For example, in this case, C shouldn't send any information to B about A, because B is the only way to A.

LINK STATE ROUTING

- The idea behind link state routing is simple and can be stated as five parts

1. Discover its neighbors and learn their network addresses.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

Learning about the Neighbors

- When a router is booted, its first task is to learn who its neighbors are.
- It accomplishes this goal by sending a special **HELLO packet** on each point-to-point line.
- The router on the other end is expected to send back a reply telling who it is.

Measuring Line Cost

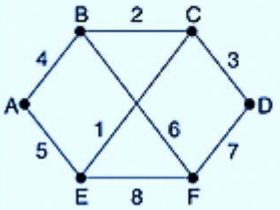
- The link state routing algorithm requires each router to know, or at least have a reasonable estimate of, the delay to each of its neighbors.

- The most direct way to determine this delay is to send over the line a special **ECHO packet** that the other side is required to send back immediately
- By measuring the round-trip time and dividing it by two, the sending router can get a reasonable estimate of the delay.
- For even better results, the test can be conducted several times, and the average used.

➢Building Link State Packets

- Once the information needed for the exchange has been collected, the next step is for each router to build a packet containing all the data.

- The packet starts with the identity of the sender, followed by a sequence number and age ,and a list of neighbors.
- Building the link state packets is easy. The hard part is determining when to build them.
- One possibility is to build them periodically, that is, at regular intervals.
- Another possibility is to build them when some significant event occurs, such as a line or neighbor going down or coming back up again or changing its properties appreciably.



Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

➢Distributing the Link State Packets

- The fundamental idea is to use flooding to distribute the link state packets.
- To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent. Routers keep track of all the (source router, sequence) pairs they see.

- When a new link state packet comes in, it is checked against the list of packets already seen. If it is new, it is forwarded on all lines except the one it arrived on. If it is a duplicate, it is discarded.
- Include the age of each packet after the sequence number and decrement it once per second. When the age hits zero, the information from that router is discarded.

➢Computing the New Routes

- Dijkstra's algorithm can be run locally to construct the shortest path to all possible destinations.
- The results of this algorithm can be installed in the routing tables, and normal operation resumed.

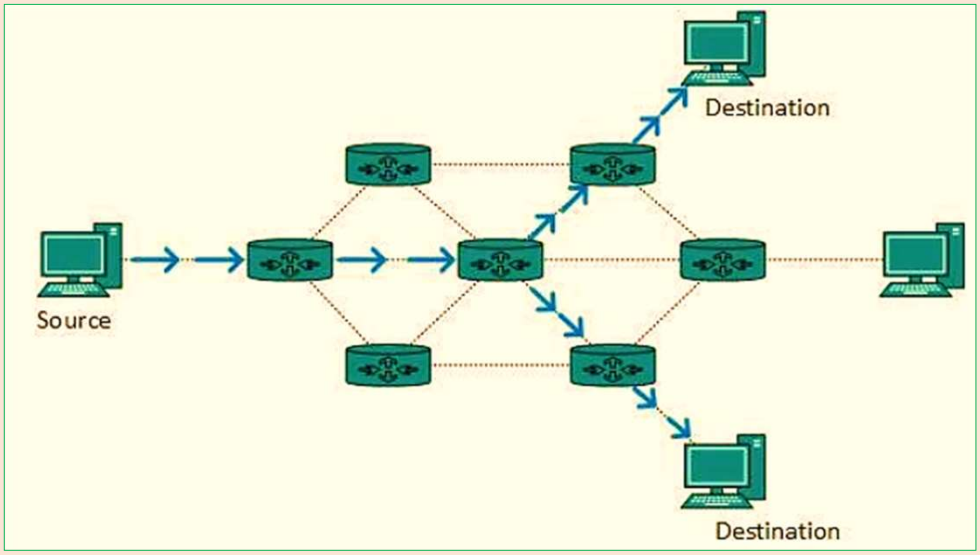
MULTICAST ROUTING

- Sending a message to a group is called multicasting, and its routing algorithm is called multicast routing
- Sending a packet to all destinations simultaneously is called broadcasting.
- Multicast routing is special case of broadcast routing with significance difference and challenges.
- In broadcast routing, packets are sent to all nodes even if they do not want it. But in Multicast routing, the data is sent to only nodes which wants to receive the packets.

- The router must know that there are nodes, which wish to receive multicast packets (or stream) then only it should forward. Multicast routing works spanning tree protocol to avoid looping.
- Multicast routing also uses Reverse Path Forwarding (RPF) technique, to detect and discard duplicates and loops.
- To do multicast routing, each router computes a spanning tree covering all other routers
- When a process sends a multicast packet to a group, the first router examines its spanning tree and prunes it, removing all lines that do not lead to hosts that are members of the group.

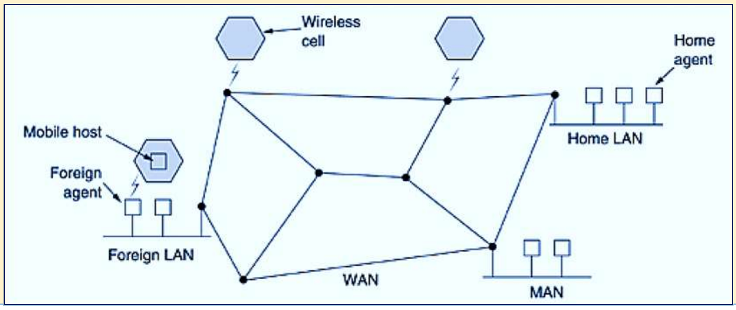
❖Multicast Routing Protocols

- Unicast routing protocols use graphs while Multicast routing protocols use trees, i.e. spanning tree to avoid loops.
- The optimal tree is called shortest path spanning tree.
- **DVMRP** - Distance Vector Multicast Routing Protocol
- **MOSPF** - Multicast Open Shortest Path First
- **CBT** - Core Based Tree
- **PIM** - Protocol independent Multicast



ROUTING FOR MOBILE HOSTS

- Millions of people have portable computers nowadays, and they generally want to read their e-mail and access their normal file systems wherever in the world they may be.
- These mobile hosts introduce a new complication: to route a packet to a mobile host, the network first has to find it.



❖General overview of Hosts:

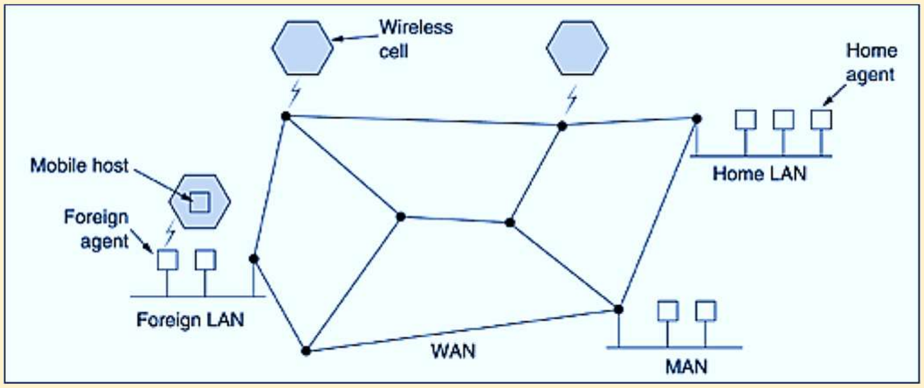
- The model of the world that network designers typically use is shown in the above figure.
- Here we have a WAN consisting of routers and hosts.
- Connected to the WAN are LANs, MANs, and wireless cells.
- Hosts that never move are said to be **stationary.**
- They are connected to the network by copper wires or fiber optics.

❖Mobile Host:

- By the term mobile host, all hosts that are away from home and still want to be connected.
- All hosts are assumed to have a permanent home location that never changes.
- The routing goal in systems with mobile hosts is to make it possible to send packets to mobile hosts using their home addresses and have the packets efficiently reach them wherever they may be.
- The trick, of course, is to find them.

➢Model of the World:

- In the model of following Figure , the world is divided up (geographically) into small units called areas, where an area is typically a LAN or wireless cell.



- Each area has one or more foreign agents, which are processes that keep track of all mobile hosts visiting the area.
- In addition, each area has a home agent, which keeps track of hosts whose home is in the area, but who are currently visiting another area.
- When a new host enters an area, either by connecting to it (e.g., plugging into the LAN) or just wandering into the cell, his computer must register itself with the foreign agent there.

➢Registration process:

The registration procedure typically works like this:

- Periodically, each foreign agent broadcasts a packet announcing its existence and address. A newly-arrived mobile host may wait for one of these messages, but if none arrives quickly enough, the mobile host can broadcast a packet saying: Are there any foreign agents around?
- The mobile host registers with the foreign agent, giving its home address, current data link layer address, and some security information.

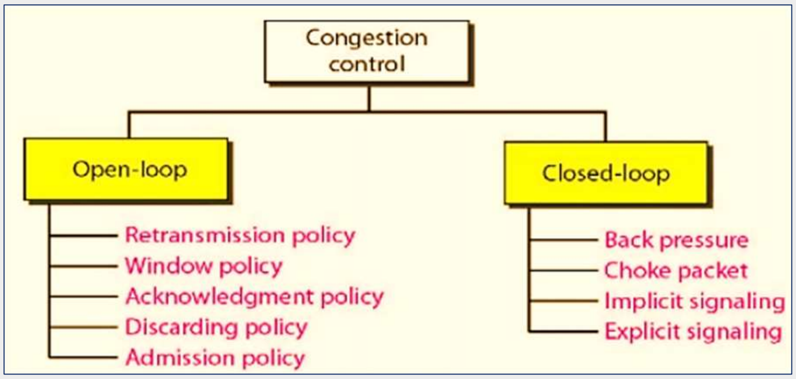
- The foreign agent contacts the mobile host's home agent and says: One of your hosts is over here. The message from the foreign agent to the home agent contains the foreign agent's network address. It also includes the security information to convince the home agent that the mobile host is really there.
- The home agent examines the security information, which contains a timestamp, to prove that it was generated within the past few seconds. If it is happy, it tells the foreign agent to proceed.
- When the foreign agent gets the acknowledgement from the home agent, it makes an entry in its tables and informs the mobile host that it is now registered.

CONGESTION CONTROL

- When too many packets are present in (a part of) the subnet, performance degrades. This situation is called **congestion**
- Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened.
- In general, we can divide congestion control mechanisms into two broad categories:

➢ **open-loop congestion control (prevention)**

➢ **closed-loop congestion control (removal)**



❖Open-Loop Congestion Control

- In open-loop congestion control, policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination

1. Retransmission Policy

- Retransmission is sometimes unavoidable. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted.
- Retransmission in general may increase congestion in the network. However, a good retransmission policy can prevent congestion.
- The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion.
- For example, the retransmission policy used by TCP is designed to prevent or alleviate congestion.

2. Window Policy

- The type of window at the sender may also affect congestion.
- The Selective Repeat window is better than the Go-Back-N window for congestion control.
- In the Go-Back-N window, when the timer for a packet times out, several packets may be resent, although some may have arrived safe and sound at the receiver. This duplication may make the congestion worse.
- The Selective Repeat window, on the other hand, tries to send the specific packets that have been lost or corrupted.

3. Acknowledgment Policy

- The acknowledgment policy imposed by the receiver may also affect congestion.
- If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion.
- Several approaches are used in this case. A receiver may send an acknowledgment only if it has a packet to be sent or a special timer expires.
- A receiver may decide to acknowledge only N packets at a time. We need to know that the acknowledgments are also part of the load in a network.
- Sending fewer acknowledgments means imposing less load on the network.

4. Discarding Policy

- A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission.
- For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented or alleviated.

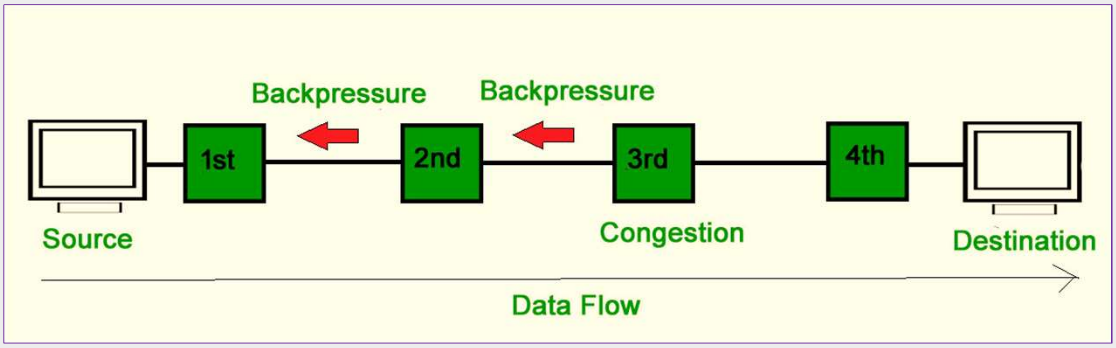
5. Admission Policy

- An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual-circuit networks.
- Switches in a flow first check the resource requirement of a flow before admitting it to the network.
- A router can deny establishing a virtual circuit connection if there is congestion in the network or if there is a possibility of future congestion.

❖Closed-Loop Congestion Control

- Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols.

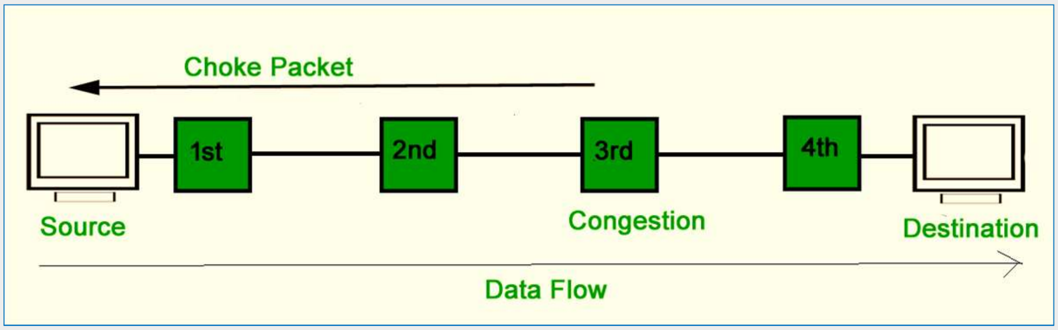
1. Backpressure



- Node 3 in the figure has more input data than it can handle. It drops some packets in its input buffer and informs node 2 to slow down.
- Node 2, in turn, may be congested because it is slowing down the output flow of data. If node 2 is congested, it informs node 1 to slow down, which in turn may create congestion.
- If so, node 1 informs the source of data to slow down. This, in time, alleviates the congestion.
- The pressure on node 3 is moved backward to the source to remove the congestion.

2. Choke Packet

- A choke packet is a packet sent by a node to the source to inform it of congestion.
- In backpressure, the warning is from one node to its upstream node, although the warning may eventually reach the source station.
- In the choke packet method, the warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet has traveled are not warned.



3. Implicit Signaling

- In implicit signaling, there is no communication between the congested node or nodes and the source. The source guesses that there is a congestion somewhere in the network from other symptoms.

- For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested.
- The delay in receiving an acknowledgment is interpreted as congestion in the network; the source should slow down.

4. Explicit Signaling

- The node that experiences congestion can explicitly send a signal to the source or destination.
- In the choke packet method, a separate packet is used for this purpose; in the explicit signaling method, the signal is included in the packets that carry data.
- Explicit signaling can occur in either the forward or the backward direction.

- Backward Signaling - A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.
- Forward Signaling - A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can use policies, such as slowing down the acknowledgments, to alleviate the congestion.

QUALITY OF SERVICE

- A stream of packets from a source to a destination is called a flow. In a connection-oriented network, all the packets belonging to a flow follow the same route; in a connectionless network, they may follow different routes. The needs of each flow can be characterized by four primary parameters:

1. **reliability**
2. **delay**
3. **Jitter**
4. **bandwidth**

- Together these determine the QoS (Quality of Service) the flow requires

- **Reliability** - Reliability is a characteristic that a flow needs. Lack of reliability means losing a packet or acknowledgment, which entails retransmission.
- **Delay** - Source-to-destination delay is another flow characteristic.
- **Jitter** -Jitter is the variation in delay for packets belonging to the same flow. For example, if four packets depart at times 0, 1, 2, 3 and arrive at 20, 21, 22, 23, all have the same delay, 20 units of time. On the other hand, if the above four packets arrive at 21, 23, 21, and 28, they will have different delays: 21,22, 19, and 24.
- **Bandwidth** - Different applications need different bandwidths. In video conferencing we need to send millions of bits per second to refresh a color screen.

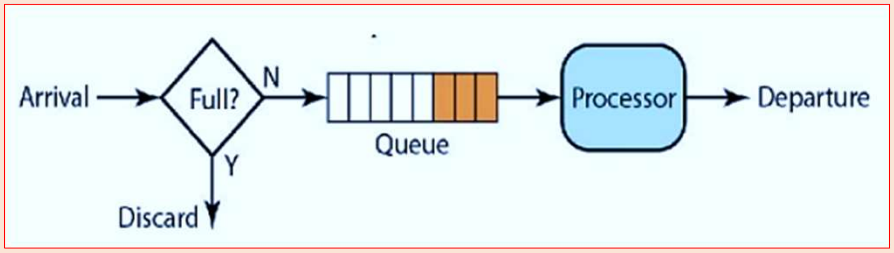
TECHNIQUES FOR ACHIEVING GOOD QUALITY OF SERVICE

1. Scheduling

- Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner. Following are some scheduling technique

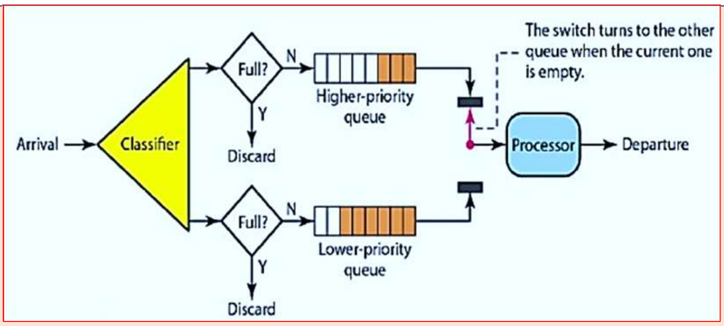
a) FIFO Queuing

- In first-in, first-out (FIFO) queuing, packets wait in a buffer (queue) until the node (router or switch) is ready to process them.
- If the average arrival rate is higher than the average processing rate, the queue will fill up and new packets will be discarded.



b) Priority Queuing

- In priority queuing, packets are first assigned to a priority class. Each priority class has its own queue.
- The packets in the highest-priority queue are processed first.
- Packets in the lowest-priority queue are processed last.

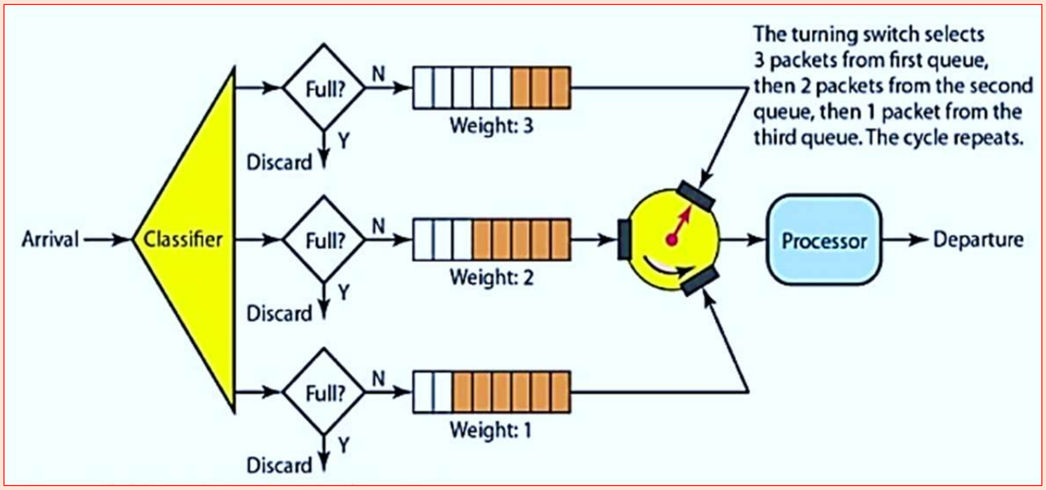


- A priority queue can provide better QoS than the FIFO queue because higher priority traffic, such as multimedia, can reach the destination with less delay.
- If there is a continuous flow in a high-priority queue, the packets in the lower-priority queues will never have a chance to be processed. This is a condition called starvation.

c) *Weighted Fair Queuing*

- In this technique, the packets are still assigned to different classes and admitted to different queues. The queues are weighted based on the priority of the queues; higher priority means a higher weight.
- The system processes packets in each queue in a round-robin fashion with the number of packets selected from each queue based on the corresponding weight.
- For example, if the weights are 3, 2, and 1, three packets are processed from the first queue, two from the second queue, and one from the third queue. If the system does not impose priority on the classes, all weights can be equal

Fig: Weighted fair queuing



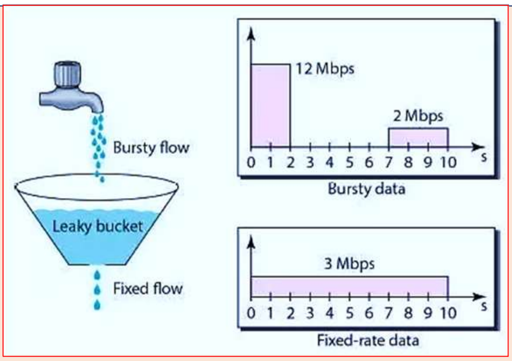
The turning switch selects 3 packets from first queue, then 2 packets from the second queue, then 1 packet from the third queue. The cycle repeats.

2. Traffic Shaping

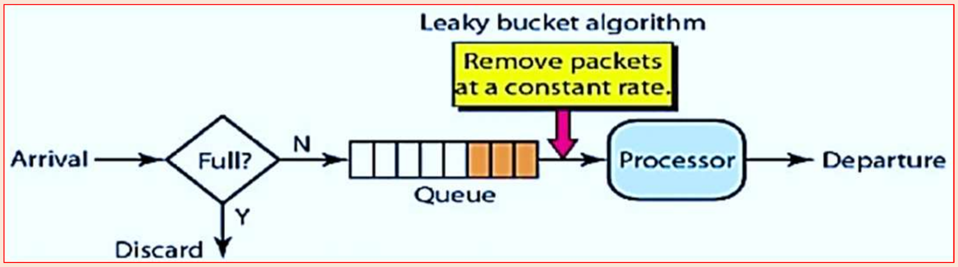
- Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques can shape traffic: leaky bucket and token bucket.

a) Leaky Bucket

- If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket.
- The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty.
- The input rate can vary, but the output rate remains constant.



- In Figure , the host sends a burst of data at a rate of 12 Mbps for 2 s, for a total of 24 Mbits of data. The host is silent for 5 s and then sends data at a rate of 2 Mbps for 3 s, for a total of 6 Mbits of data. In all, the host has sent 30 Mbits of data in 10s. The leaky bucket smooth's the traffic by sending out data at a rate of 3 Mbps during the same 10 s.



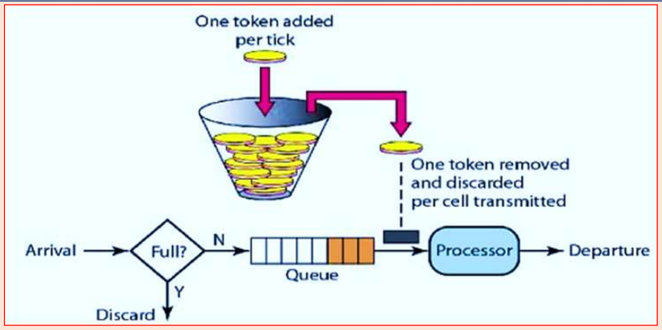
- A FIFO queue holds the packets. If the traffic consists of fixed-size packets (e.g., cells in ATM networks), the process removes a fixed number of packets from the queue at each tick of the clock.
- If the traffic consists of variable-length packets, the fixed output rate must be based on the number of bytes or bits.

➢The following is an algorithm for variable-length packets:

- Initialize a counter to n at the tick of the clock.
- If n is greater than the size of the packet, send the packet and decrement the counter by the packet size. Repeat this step until n is smaller than the packet size.
- Reset the counter and go to step 1.
- A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.

b) Token Bucket

- The leaky bucket is very restrictive. It does not credit an idle host. For example, if a host is not sending for a while, its bucket becomes empty.
- Now if the host has bursty data, the leaky bucket allows only an average rate. The time when the host was idle is not taken into account.
- On the other hand, the token bucket algorithm allows idle hosts to accumulate credit for the future in the form of tokens. For each tick of the clock, the system sends n tokens to the bucket.
- The system removes one token for every cell (or byte) of data sent. For example, if n is 100 and the host is idle for 100 ticks, the bucket collects 10,000 tokens.



- The token bucket can easily be implemented with a counter. The token is initialized to zero. Each time a token is added, the counter is incremented by 1. Each time a unit of data is sent, the counter is decremented by 1. When the counter is zero, the host cannot send data.
- The token bucket allows bursty traffic at a regulated maximum rate.

➢Combining Token Bucket and Leaky Bucket

- The two techniques can be combined to credit an idle host and at the same time regulate the traffic.
- The leaky bucket is applied after the token bucket; the rate of the leaky bucket needs to be higher than the rate of tokens dropped in the bucket.

c) Resource Reservation

- A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on. The quality of service is improved if these resources are reserved beforehand.
- One QoS model called Integrated Services, which depends heavily on resource reservation to improve the quality of service.

d) Admission Control

- Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications.
- Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.