CST 303 COMPUTER NETWORKS

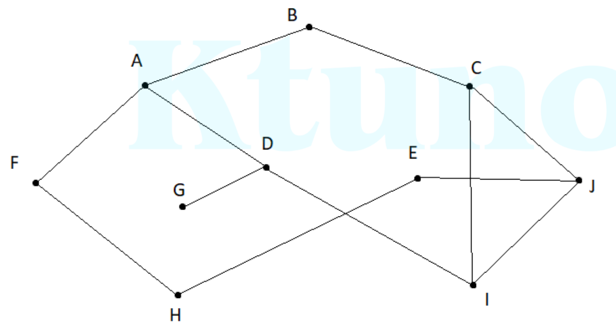
Module 3 Important Questions

Q1:Explain the optimality principle. What is a sink tree?

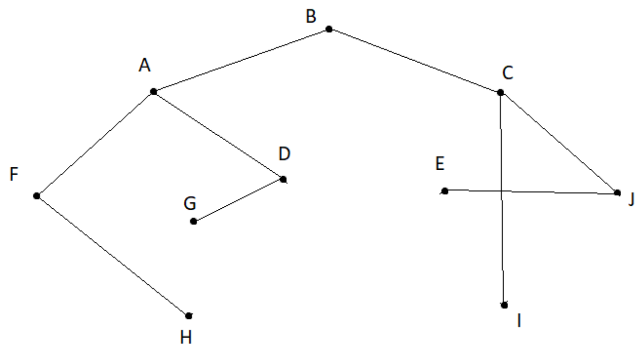
A:

- Optimality refers to the capability of a routing algorithm to select the best route.
- Optimality principle states that if router *J* is on the optimal path from router *I* to router *K*, then the optimal path from *J* to *K* also falls along the same route.
- To see this, we call the part of the route from *I* to *J* as *r*1 and the rest of the route as *r*2. If a route better than *r*2 existed from *J* to *K*, it could be concatenated with *r*1 to improve the route from *I* to *K*, contradicting our statement that *r*1*r*2 is optimal.
- The set of optimal routes from all sources to a given destination form a tree rooted at the destination, such a tree is called a **sink tree,** where the distance metric is the number of hops.
- Note that a sink tree is not necessarily unique; other trees with the same path lengths may exist.
- The goal of all routing algorithms is to discover and use the sink trees for all routers.

Eg:



a) A subnet



b) A sink tree for router B

Q2: Discuss the shortest path routing. Also explain the Dijkstra's algorithm in detail

A:

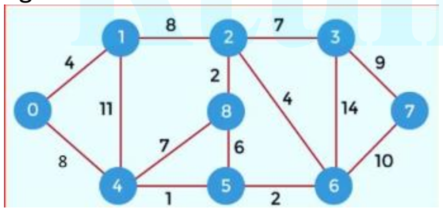
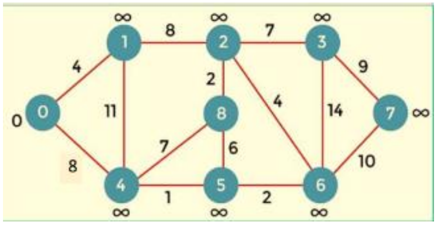
-It is one of the simple static routing algorithms that are widely used for routing in the network.
-The basic idea is to build a graph with each node representing a router and each line representing a communication link.
- A path selected can be called shortest in many contexts. If cost is selected as the criteria then the shortest path is the route which is least expensive. The cost is determined depending upon the criteria to be optimized, they include:

- **Minimum number of hops:** If each link is given a unit cost, the shortest path is the one with minimum number of hops. Such a route is easily obtained by a breadth first search method. This is easy to implement but ignores load, link capacity etc.
- **Transmission and Propagation Delays:** If the cost is fixed as a function of transmission and propagation delays, it will reflect the link capacities and the geographical distances. However these costs are essentially static and do not consider the varying load conditions.
- **Queuing Delays:** If the cost of a link is determined through its queuing delays, it takes care of the varying load conditions, but not of the propagation delays.

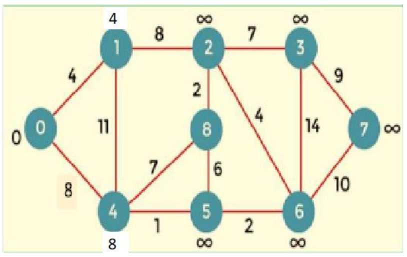
Dijkstra's algorithm

It is a single-source shortest path algorithm, i.e. only one source is given, and we have to find the shortest path from the source to all the nodes.

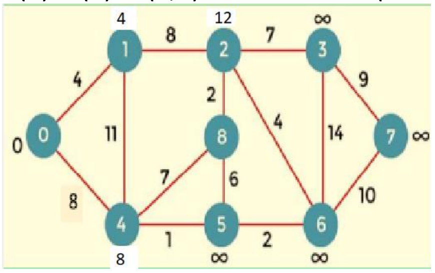
STEPS:

1. **Consider any vertex as the source vertex**. Here, let's take 0 as the source.
   eg:
   
2. **Set the distance of all vertices from the source as infinity**.
   
3. **Find the distance from source to the directly connected vertices and update them.**
   Use the formula:
   If d(x) + c(x,y) < d(y) Then, d(y)= d(x) + c(x,y)
   {here, d(x) is the distance between source and x. c(x,y) is the cost or distance between the vertices x and y}
   Here in the example, 1 and 4 are directly connected to the source 0.
   so , d(1)=d(0) + c(0,1) < ∞
   I.e. d(1)= 0+4=4 < ∞
   Similarly, d(4)= d(0) + c(0,4) <∞
   i.e d(4)=0+8=8 <∞

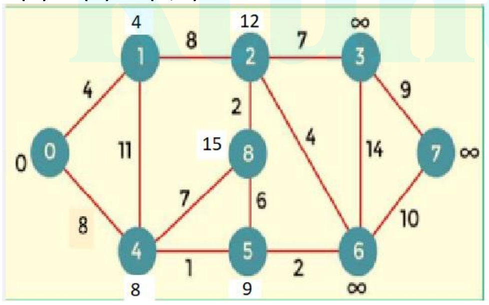
Ktunotes.in



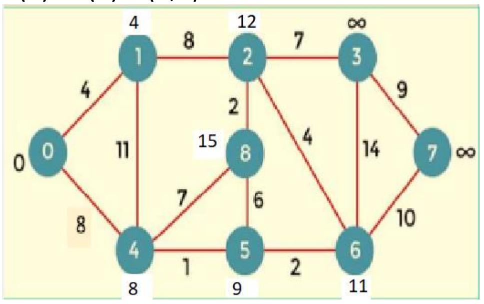
4. **Compare the distance of vertices except the source vertex and select the vertex with minimum distance.** Here, we select vertex 1.
5. **Then consider the vertices directly connected to the selected vertex**. Here, we have 2 paths from vertex 1 : to vertex 2 and 4. **Calculate the distance using the formula and update the graph.**
   d(2)= d(1) +c(1,2) = 4 + 8=12 < ∞
   d(4)=d(1) +c(1,4)=4+11=15 > 8 (no updation)



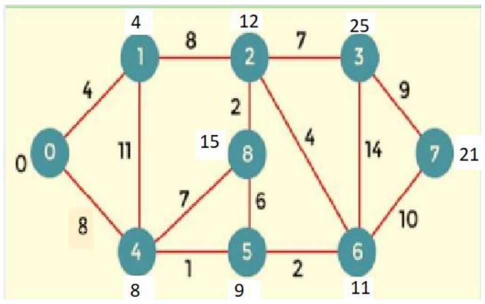
6. Now compare the distances of vertices which were not selected previously and select the vertex with minimum distance. Here, we select the vertex 4. Now repeat step 5.
   d(8)=d(4)+c(4,8) = 8+7=15 < ∞
   d(5)=d(4) +c(4,5)= 8+1=9 <∞



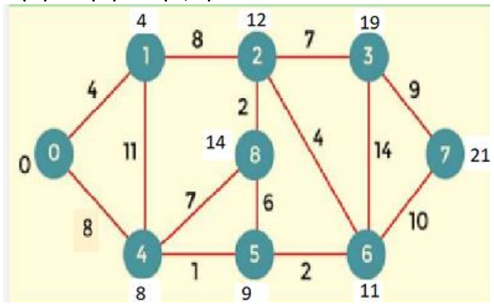
7. Select vertex 5 and repeat step 5.
   d(8)=d(5)+c(5,8)= 9+6=15 (no updation)
   d(6)= d(5)+c(5,6)= 9+2= 11 <∞



8. Select vertex 6
   d(2)=d(6)+c(6,2)=11+4=15 > 12 (no updation)
   d(3)= d(6)+c(6,3)=11+14=25 <∞
   d(7)= d(6) +c(6,7)=11+10=21<∞



9. Select vertex 2
   d(3)=d(2)+c(2,3)=12+7=19 <25
   d(8)= d(2) + c(2,8)=12+2=14 <15



10. Select vertex 3
    d(7)=d(3)+c(3,7)=19+9=28 >21 (no updation)
11. Now we are left to select only one vertex, i.e. vertex 7. By here, we stop the algorithm. Now we have the minimum distance of each vertex from the source vertex as shown in the figure above.

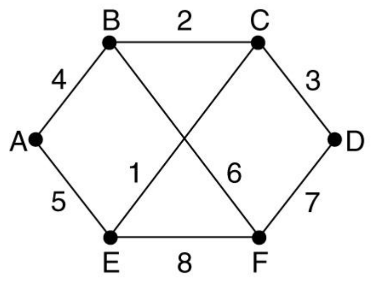
Q3: Explain link State routing algorithm.

A:

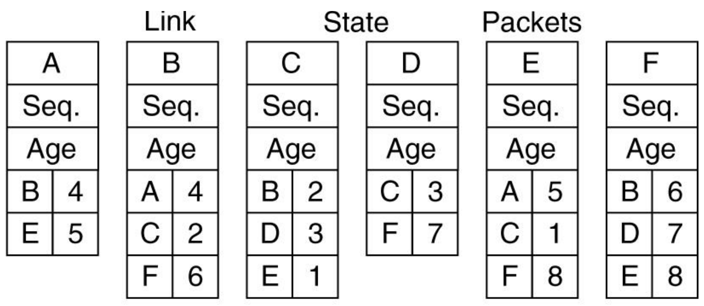
-Link State Routing is dynamic routing.

-Each router must do the following:

1. Discover its neighbors, learn their network address.
   - When a router is booted, it first learns its immediate neighbors.
   - It sends a HELLO packet on each point-to-point line. The router on the other end will send a reply telling who it is.
   - Each router has a global unique name.
   - If two or more routers are connected by a LAN, we can model the LAN as a node.
2. Measure the delay or cost to each of its neighbors.
   - It sends an ECHO packet, measures the round trip delay, and divides it by two. This is repeated several times to have a better estimation.
3. Construct a packet telling all it has just learned.
   - Build the link state packet containing: node ID, sequence number, age, a list of neighbors and the delay to the neighbor.



(a)



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

(b)

4. Send this packet to all other routers.
   - The concept of flooding is used to distribute the link state packets.
   - To keep the flood in check, each packet contains a sequence number that is increased by one for each new packet.
   - The age decreases by one per second. The packet is discarded when age = 0.
   - When a link state packet arrives, the router checks if it is new.
     - If Yes ▯ forward it to all outgoing lines except the one it arrived.
     - If No (duplicated or with low sequence number) ▯ discard it.
5. Compute the shortest path to every other router.
   - Once a router has accumulated a full set of link state packets, it knows all nodes and links, thus can construct the subnet graph.
   - Run Dijkstra algorithm to find the shortest paths from the source to all other nodes.

Q4: Describe the distance vector routing algorithm in detail (include count- to- infinity topic)

A:

- 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.

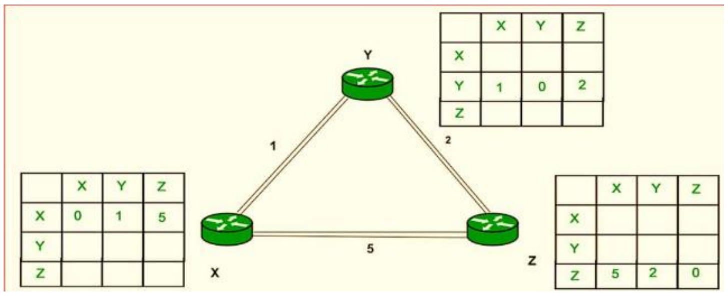
Distance Vector Routing algorithm – working

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

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

Example

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



	X	Y	Z
X			
Y	1	0	2
Z			

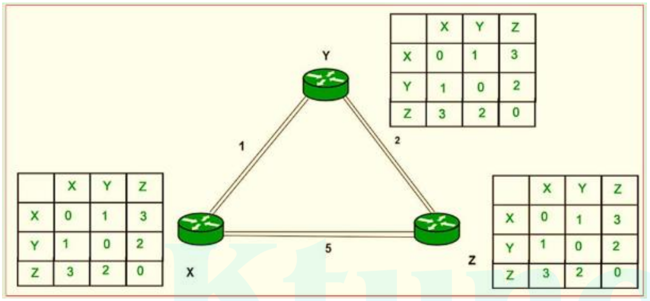
	X	Y	Z
X	0	1	5
Y			
Z			

	X	Y	Z
X			
Y			
Z	5	2	0

• Consider router X , X will share its 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 an intermediate node(hop) so it will be updated in routing table X. Similarly for Z also. Finally the routing table for all nodes will be:



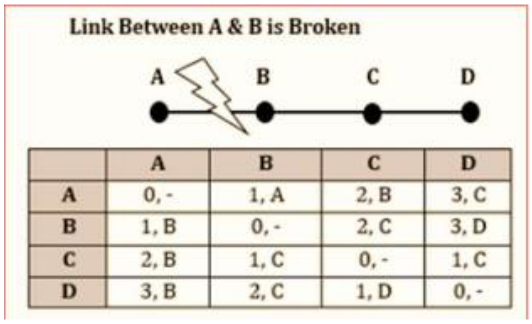
	X	Y	Z
X	0	1	3
Y	1	0	2
Z	3	2	0

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

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

Counting 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.
- Consider a network with a graph as shown above in figure . In this graph, there is only one link between A and the other parts of the network.



	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, -

- 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.
- **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.

Q5: Explain the flood based routing algorithm

A:

- Flooding is a **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.
- 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 neighbors.
- **Controlled flooding** − They use some methods to control the transmission of packets to the neighboring 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 approximately in the right direction, instead of every available path.

Q6: Explain the routing for mobile hosts

A:

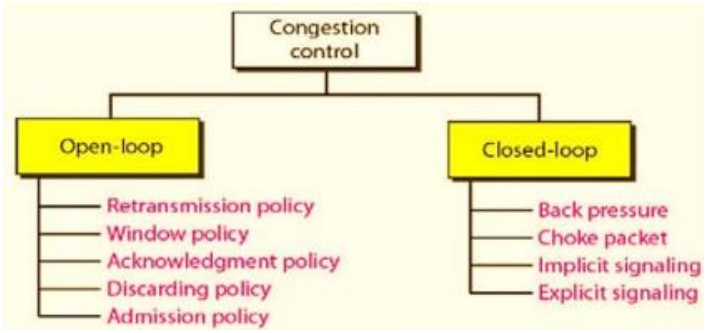
- **All hosts that are away from home and still want to be connected are known as mobile hosts.**
- All hosts are assumed to have a permanent home location that never changes.
- **The routing goal in systems with mobile hosts is to make it possible to send packets to mobile hosts** using their home addresses and have the packets efficiently reach them **wherever they may be**.
- The 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.
- 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.

Q7: What is meant by congestion in networks? What are the different congestion control techniques /what is open loop and closed loop congestion control?

A:

- 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.



<u>Open-Loop Congestion Control:</u>

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.

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 congestion**.

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.

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**.

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.

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.

Backpressure

- In backpressure, the **warning is from one node to its upstream node**, although the warning may eventually reach the source station.

Choke Packet

- A choke packet is a **packet sent by a node to the source to inform it of congestion**.
- 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.

Implicit Signaling

- In implicit signaling, there is **no communication between the congested node or nodes and the**

source. **The source guesses that there is congestion somewhere in the network from other symptoms.**

- For example, when a source sends several packets and there is no acknowledgement 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.

Explicit Signaling

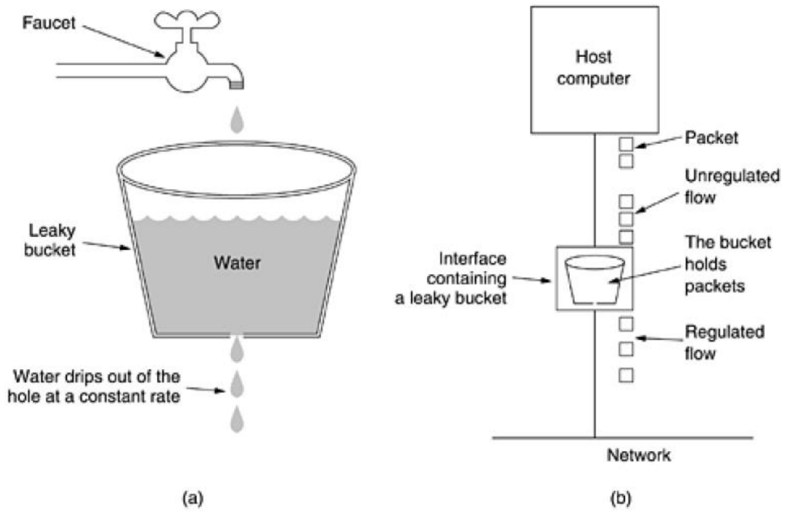
- The node that experiences congestion can explicitly send a signal to the source or destination.
- **Explicit signaling can occur in either the forward or the backward direction.**
- **Backward Signaling -** A bit can be set in a packet moving in the direction opposite to the congestion. This bit can 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

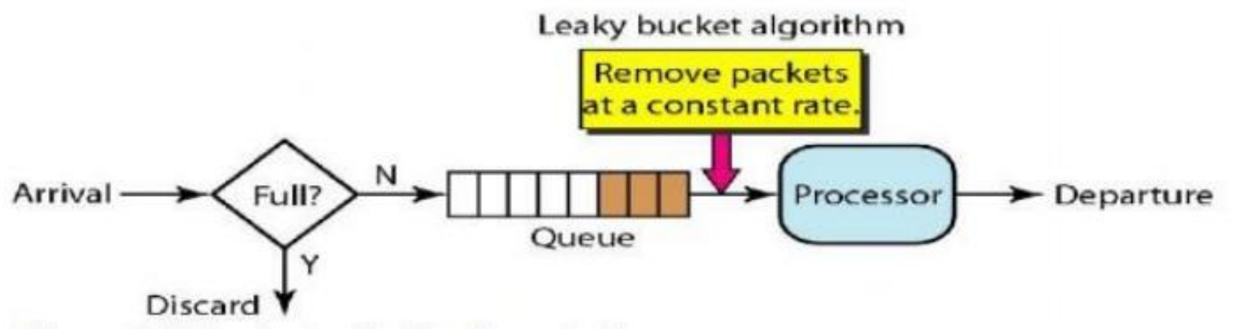
Q8: List and explain various congestion control algorithms/define leaky bucket and token bucket algorithm for congestion control.

A:

<u>**Leaky Bucket Algorithm**</u>

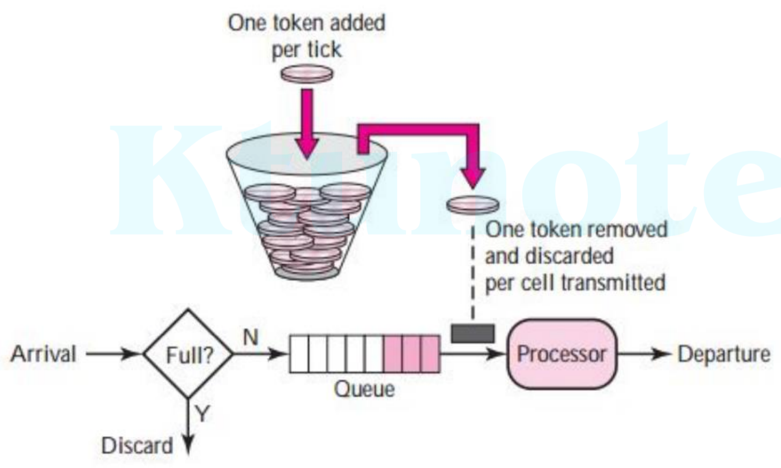
- Each host is connected to the network by an interface containing a leaky bucket, that is, a finite internal queue.
- No matter the rate at which water enters the bucket, the outflow is at a constant rate.
- If a packet arrives at the queue when it is full, the packet is discarded.





Token Bucket Algorithm

- The bucket holds tokens instead of packets
- Tokens are generated and placed into the token bucket at a constant rate
- When a packet arrives at the token bucket, it is transmitted if there is a token available. Otherwise it is buffered until a token becomes available.
- The token bucket has a fixed size, so when it becomes full, subsequently generated tokens are discarded.
- When the number of available tokens is reduced to zero, all new requests are denied.



Q9: What is meant by QoS? Explain the techniques for achieving good QoS.

A:

- Quality of Service (QoS) refers to the capability of a network to provide better service to selected network traffic over various technologies, including Frame Relay, Asynchronous Transfer Mode (ATM),Ethernet etc.
- Four parameters that determine QoS are:
    - **Reliability**: No bits must be delivered incorrectly.
    - **Delay:** The time it takes for a packet to go from its source to its end destination.
    - **Jitter:** The irregular speed of packets on a network as a result of congestion, which can result in packets arriving late and out of sequence.
    - **Bandwidth :** Bandwidth management mechanisms measure and control traffic flows on the network to avoid exceeding its capacity and the resulting network congestion that occurs.

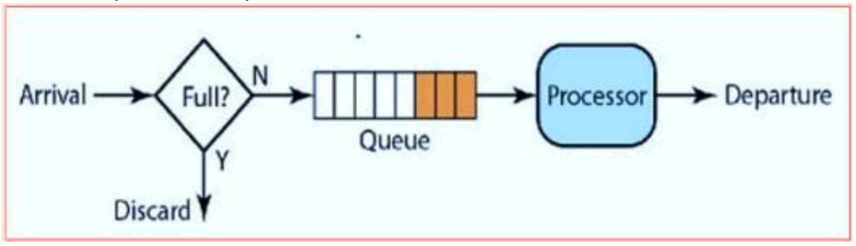
Techniques for achieving good Quality of Services:

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 techniques:

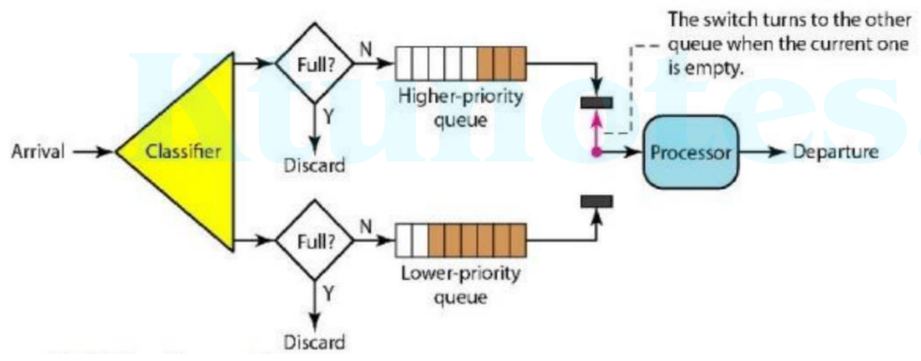
a) FIFO Queuing

- 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.



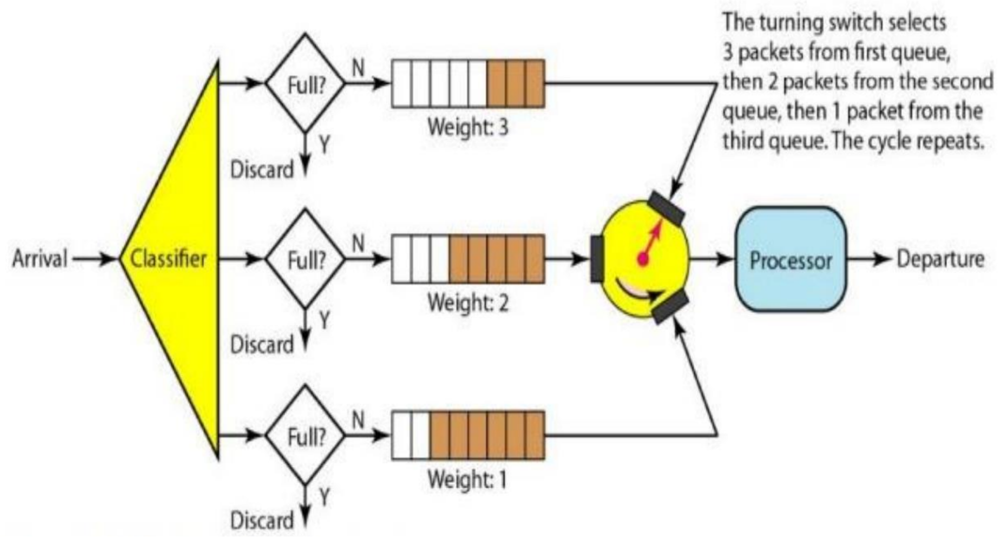
b) Priority Queuing

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

- 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.

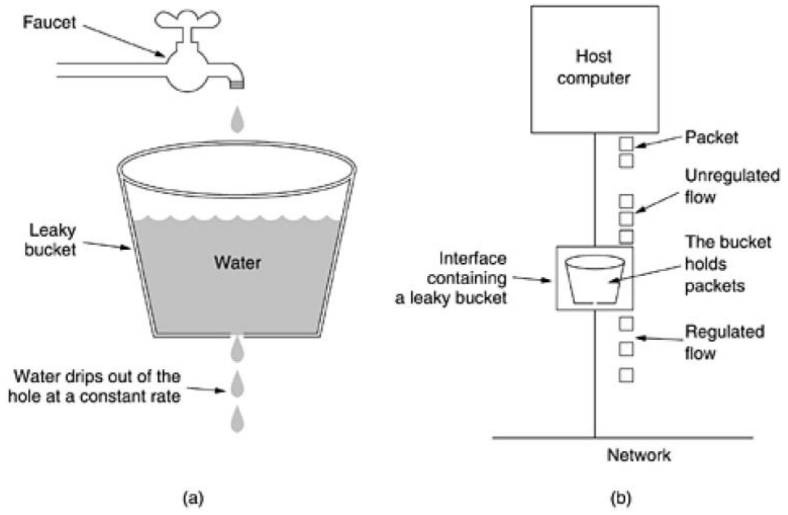

The turning switch selects 3 packets from first queue, then 2 packets from the second queue, then 1 packet from the third queue. The cycle repeats.
Arrival
Classifier
Full?
N
Y
Discard
Weight: 3
Weight: 2
Weight: 1
Processor
Departure

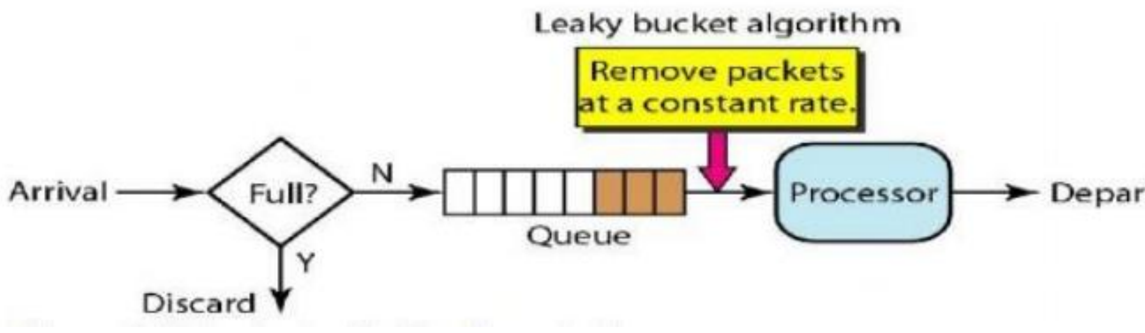
2. Traffic shaping

- It is a mechanism to control the amount and the rate of the traffic sent to the network.
- Traffic shaping is about regulating the average rate (and burstiness) of data transmission.
- It reduces congestion.
- There are 2 techniques: Leaky bucket & Token bucket algorithms

i) Leaky Bucket Algorithm

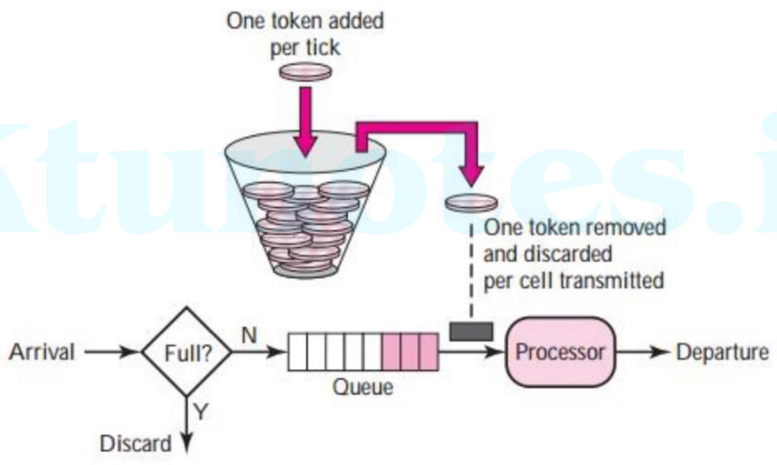
- Each host is connected to the network by an interface containing a leaky bucket, that is, a finite internal queue.
- No matter the rate at which water enters the bucket, the outflow is at a constant rate.
- If a packet arrives at the queue when it is full, the packet is discarded.





ii) Token Bucket Algorithm

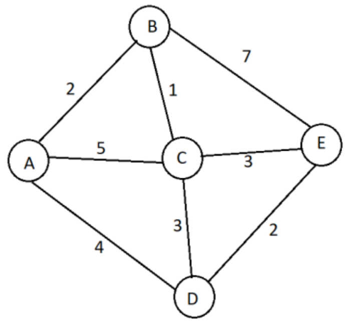
- The bucket holds tokens instead of packets
- Tokens are generated and placed into the token bucket at a constant rate
- When a packet arrives at the token bucket, it is transmitted if there is a token available. Otherwise it is buffered until a token becomes available.
- The token bucket has a fixed size, so when it becomes full, subsequently generated tokens are discarded.
- When the number of available tokens is reduced to zero, all new requests are denied.



3. **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.

4. **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.

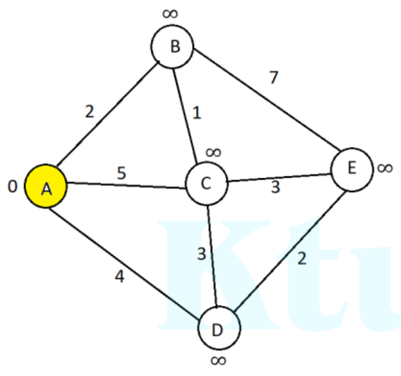
Q10: Find the shortest path from node A to all other nodes using Dijkstra's algorithm.



A:

Source vertex: A

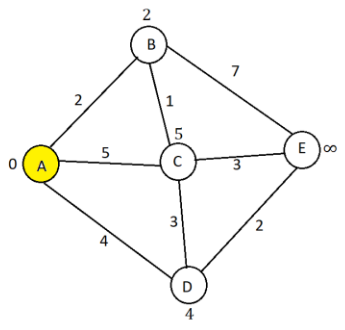
Assign distance infinity to all vertices, except source.



d(B)=d(A)+c(A,B)=0+2=2 < ∞

d(C)=d(A)+c(A,C)=0+5=5 < ∞

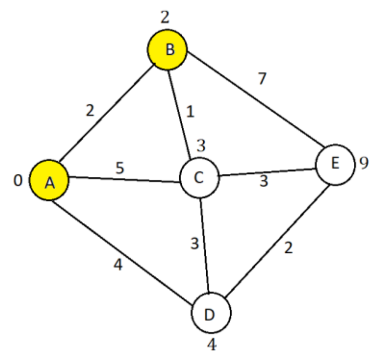
d(D)=d(A)+c(A,D)=0+4=4 <∞



Select vertex B

d(C)=d(B)+c(B,C)=2+1=3 < 5

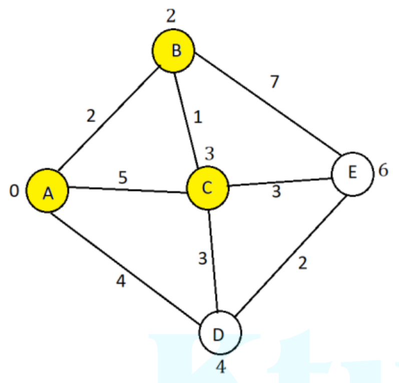
d(E)=d(B)+c(B,E)=2+7=9<∞



Select vertex C

d(E)=d(C)+c(C,E)=3+3=6<9

d(D)=d(C)+c(C,D)=3+3=6>4 (no updation)



Select vertex D

d(E)=d(D)+c(D,E)=4+2=6 (no updation)

Therefore, shortest distance from source vertex A to all other vertices is given as

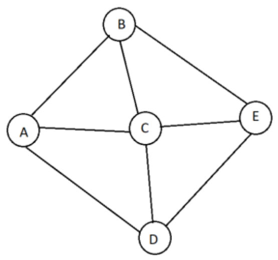
A to B=2

A to C=3

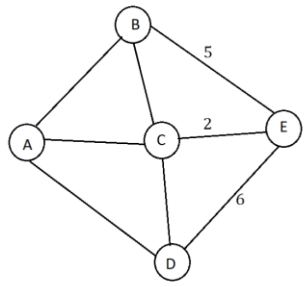
A to D=4

A to E=6

Q11: Consider the following subnet. Distance vector routing is used and the following have just come in to router E: from B: (13,0,6,10,4), from C: (11,7,0,10,1) and from D: (3,16,3,0,9). The measured delays to B,C and D are 5,2,6 respectively. What is E's new routing table? Give both the outgoing line to use and the expected delay.



A:



B's TABLE

	A	B	C	D	E
A					
B	13	0	6	10	4
C					
D					
E					

C's TABLE

	A	B	C	D	E
A					
B					
C	11	7	0	10	1
D					
E					

D's TABLE

	A	B	C	D	E
A					
B					
C					
D	3	16	3	0	9

E					

Measured delay for B=5, So add 5 to B’s table
Outgoing line through B: (18,5,11,15,9)
Measured delay for C=2, so add 2 to C’s table
Outgoing line through C: (13,9,2,12,3)
Measured delay for D=6, so add 6 to D’s table
Outgoing line through D: (9,22,9,6,15)
Taking minimum for each destination (except E):
E’s table: (9,5,2,6,0)
Outgoing lines are: (D,B,C,D,-)

E’s TABLE

	A	B	C	D	E
A					
B	18	5	11	15	9
C	13	9	2	12	3
D	9	22	9	6	15
E	9	5	2	6	0