CST 303 COMPUTER NETWORKS

MODULE 3

Network layer design issues. Routing algorithms - The Optimality Principle, Shortest path routing, Flooding, Distance Vector Routing, Link State Routing, Multicast routing, Routing for mobile hosts. Congestion control algorithms. **Quality of Service (QoS) -** requirements, Techniques for achieving good QoS.

NETWORK LAYER: INTRODUCTION

The network Layer is the third layer in the OSI model of computer networks. Its main function is to transfer network packets from the source to the destination. It involves both the source host and the destination host.

Key among these services are addressing, packetizing, routing, and forwarding. Packetizing involves encapsulating data into packets suitable for transmission. Routing determines the optimal path for these packets through the network, ensuring they navigate through multiple nodes and networks efficiently. Forwarding is the process of directing these packets to their next hop along the selected path.

Services Offered by Network Layer

The **services** which are offered by the network layer protocol are as follows:

- Addressing
- Packetizing
- Routing
- Forwarding

Addressing

There are two types of addressing performed in the network: logical addressing and physical addressing. The data link layer performs the physical addressing, while the network layer does the logical addressing in the OSI model. Logical addressing is also used to distinguish between the source and destination system. The network layer adds a header to the packet, which includes the logical addresses of both the sender and the receiver.

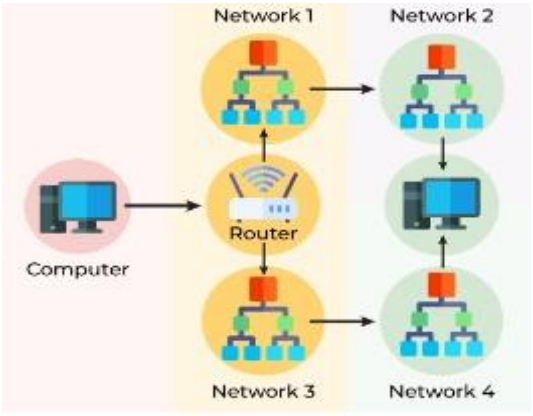
Packetizing

- The process of encapsulating the data received from the upper layers of the network (also called payload) in a network layer packet at the source and decapsulating the payload from the network layer packet at the destination is known as packetizing.
- The source host adds a header that contains the source and destination address and some other relevant information required by the network layer protocol to the payload received from the upper layer protocol and delivers the packet to the data link layer.
- The destination host receives the network layer packet from its data link layer,

decapsulates the packet, and delivers the payload to the corresponding upper layer protocol. The routers in the path are not allowed to change either the source or the destination address. The routers in the path are not allowed to decapsulate the packets they receive unless they need to be fragmented.

Routing

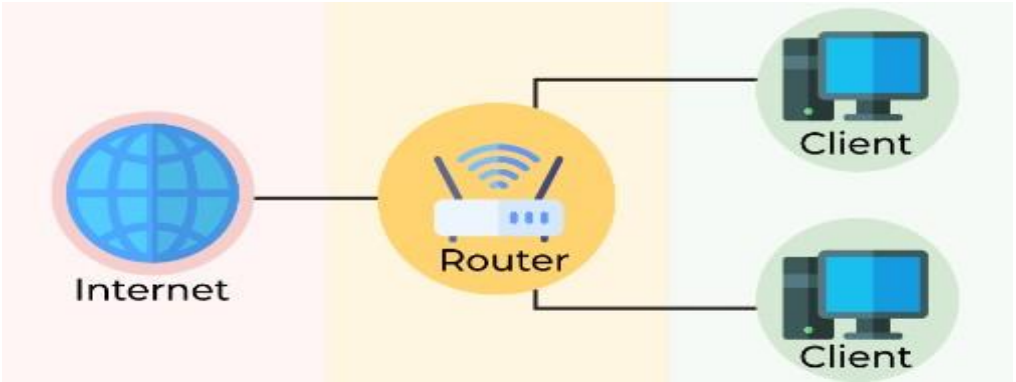
Routing is the process of moving data from one device to another device. These are two other services offered by the network layer. In a network, there are a number of routes available from the source to the destination. The network layer specifies some strategies which find out the best possible route. This process is referred to as routing. There are a number of routing protocols that are used in this process and they should be run to help the routers coordinate with each other and help in establishing communication throughout the network.



Routing

Forwarding

Forwarding is simply defined as the action applied by each router when a packet arrives at one of its interfaces. When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network (unicast routing) or to some attached networks (in the case of multicast routing). Routers are used on the network for forwarding a packet from the local network to the remote network. So, the process of routing involves packet forwarding from an entry interface out to an exit interface.



DATALINK LAYER DESIGN ISSUES

The network layer comes with some design issues that are described as follows:

1. **Store and Forward packet switching**

The host sends the packet to the nearest router. This packet is stored there until it has fully arrived once the link is fully processed by verifying the checksum then it is forwarded to the next router till it reaches the destination. This mechanism is called “Store and Forward packet switching.”

2. Services provided to the Transport Layer

Through the network/transport layer interface, the network layer transfers its **patterns** services to the transport layer. These services are described below. But before providing these services to the transfer layer, the following goals must be kept in mind:-

- Offering services must not depend on router technology.
- The transport layer needs to be protected from the type, number, and topology of the available router.
- The network addresses for the transport layer should use uniform numbering patterns**,** also at LAN and WAN connections.

Based on the connections there are 2 types of services provided :

- **Connectionless –** The routing and insertion of packets into the subnet are done individually. No added setup is required.
- **Connection-Oriented –** Subnet must offer reliable service and all the packets must be transmitted over a single route.

3. Implementation of Connectionless Service

Packets are termed as “datagrams” and corresponding subnets as “datagram subnets”. When the message size that has to be transmitted is 4 times the size of the packet, then the network layer divides into 4 packets and transmits each packet to the router via. a few protocols. Each data packet has a destination address and is routed independently irrespective of the packets.

4. Implementation of Connection-Oriented service:

To use a connection-oriented service, first, we establish a connection, use it, and then release it. In connection-oriented services, the data packets are delivered to the receiver in the same order in which they have been sent by the sender. It can be done in either two ways :

- **Circuit Switched Connection –** A dedicated physical path or a circuit is established between the communicating nodes and then the data stream is transferred.
- **Virtual Circuit Switched Connection –** The data stream is transferred over a packet switched network, in such a way that it seems to the user that there is a dedicated path from the sender to the receiver. A virtual path is established here. While, other connections may also be using the same path.

VIRTUAL CIRCUITS AND DATAGRAM CIRCUITS

Virtual Circuit is a connection-oriented service in which resources like buffers, CPU, bandwidth, etc. are used for creating a data transfer session. Virtual Circuit is also known as **connection-oriented switching**. In virtual circuits, the path that is followed by the first data packet would get fixed and all other data packets will also use the same path and consume the same resources. Consequently, a common and same header is used by all the data packets. Due to all these reasons, virtual circuits are comparatively less complex and more reliable for data transmission, however they are expensive to install and maintain.

Datagram networks are connectionless services for data transmission, in which no resources like CPU, buffer, bandwidth, etc. are required for data transmission. In datagram networks, the path for data transmission is not fixed. Therefore, the data packets are free to decide the path on any intermediate router on the way by dynamically changing the routing tables on the routers. Since the data packets follow different paths, they have different headers with information of the data packets. Due to dynamic resource allocation and dynamic path, datagram networks are error prone and less reliable. However, datagram networks are cheaper to install and maintain.

ROUTING ALGORITHMS

- In order to transfer the packets from source to the destination, the network layer must determine the best route through which packets can be transmitted.
- Whether the network layer provides datagram service or virtual circuit service, the main job of the network layer is to provide the best route. The routing protocol provides this job.
- The routing protocol is a routing algorithm that provides the best path from the source to the destination. The best path is the path that has the "least-cost path" from source to the destination.
- Routing is the process of forwarding the packets from source to the destination but the best route to send the packets is determined by the routing algorithm.

Classification of a Routing algorithm

The Routing algorithm is divided into two categories:

- Adaptive Routing algorithm
- Non-adaptive Routing algorithm

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.

An adaptive routing algorithm can be classified into three parts:

- **Centralized algorithm:** It is also known as global routing algorithm as it computes the least-cost path between source and destination by using complete and global knowledge about the network. This algorithm takes the connectivity between the nodes and link cost as input, and this information is obtained before actually performing any calculation. **Link state algorithm** is referred to as a centralized algorithm since it is aware of the cost of each link in the network.
- **Isolation algorithm:** It is an algorithm that obtains the routing information by using local information rather than gathering information from other nodes.
- **Distributed algorithm:** It is also known as decentralized algorithm as it computes the least-cost path between source and destination in an iterative and distributed manner. In the decentralized algorithm, no node has the knowledge about the cost of all the network links. In the beginning, a node contains the information only about its own directly attached links and through an iterative process of calculation computes the least-cost path to the destination. A Distance vector algorithm is a decentralized algorithm as it never knows the complete path from source to the destination, instead it knows the direction through which the packet is to be forwarded along with the least cost path.

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.

The Non-Adaptive Routing algorithm is of two types:

Flooding: In case of flooding, every incoming packet is sent to all the outgoing links except the one from it has been reached. The disadvantage of flooding is that node may contain several copies of a particular packet.

Random walks: In case of random walks, a packet sent by the node to one of its neighbours randomly. An advantage of using random walks is that it uses the alternative routes very efficiently.

Characteristics of routing algorithms

Regardless of whether routes are chosen independently for each packet or only when new connections are established, certain properties are desirable in a routing algorithm:

• correctness, simplicity,

• robustness, stability,

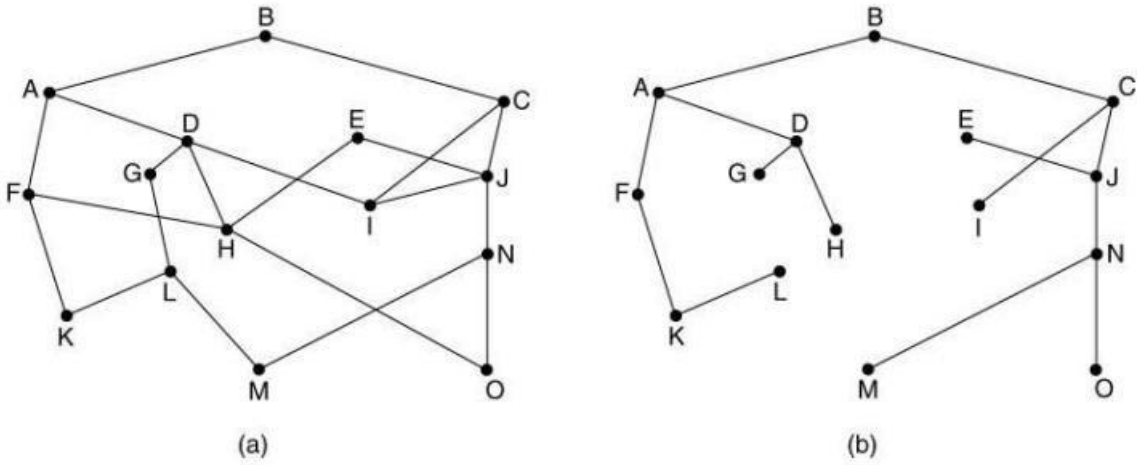
• fairness, and optimality.

Correctness and simplicity hardly require comment. **Robustness**: the routing algorithm should be able to cope with changes in topology and traffic without requiring all jobs in all hosts to be aborted and the network to be rebooted every time some router crashes. **Stability** is also an important goal for the routing algorithm. A stable algorithm reaches equilibrium and stays there. **Fairness and optimality** may sound obvious – surely no reasonable person would oppose them – but as it turn out, they are often contradictory goals.

The Optimality Principle

- One can make a general statement about optimal routes without regard to network topology or traffic.
- This statement is known as the optimality principle.
- It 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
- As a direct consequence of the optimality principle, we can see that 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.

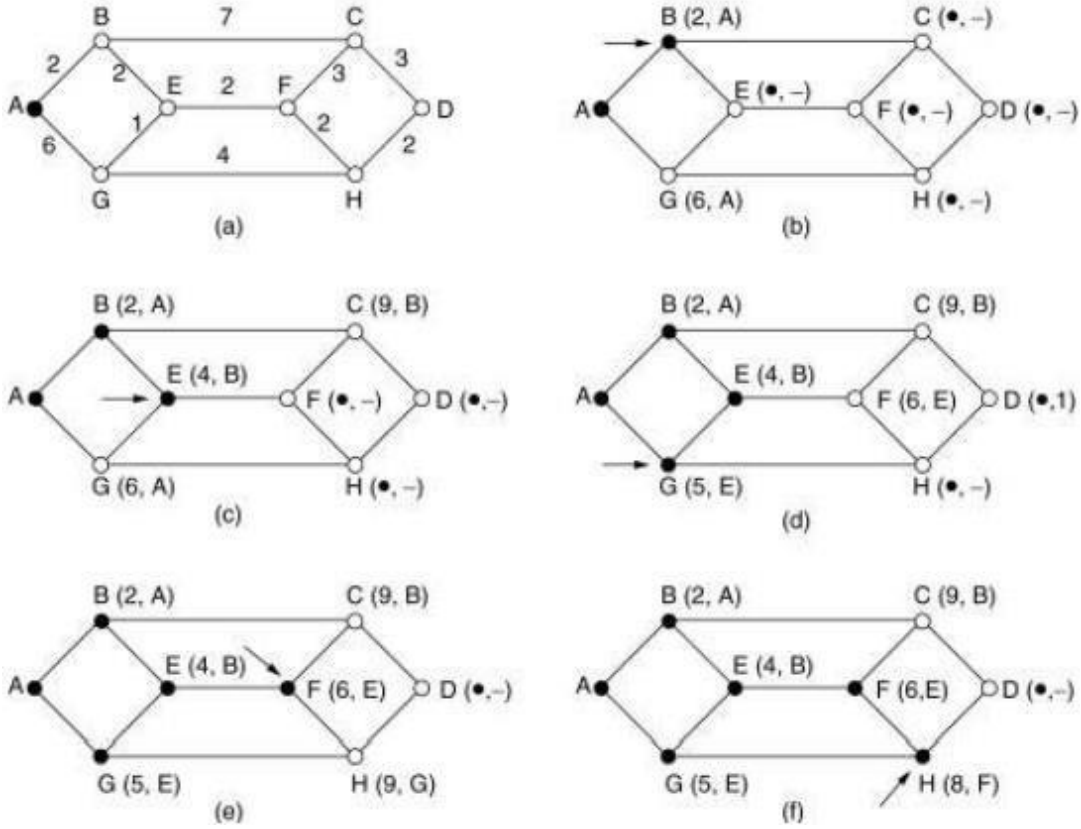
- The goal of all routing algorithms is to discover and use the sink trees for all routers



(a) A subnet. (b) A sink tree for router B.

SHORTEST PATH ROUTING

The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line or link. To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.



The first 5 steps used in computing the shortest path from A to D. The arrows indicate the working node

Many other metrics besides hops and physical distance are also possible. For example, each arc could be labelled with the mean queuing and transmission delay for some standard test packet as determined by hourly test runs. With this graph labelling, the shortest path is the fastest path rather than the path with the fewest arcs or kilometres. In the general case, the labels on the arcs could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measured delay, and other factors. By changing the weighting function, the algorithm would then compute the "shortest" path measured according to any one of a number of criteria or to a combination of criteria.

Flooding

- Another static algorithm is flooding, in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.
- One such measure is to have a hop counter contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero. Ideally, the hop counter should be initialized to the length of the path from source to destination. A variation of flooding that is slightly more practical is selective flooding. • In this algorithm the routers do not send every incoming packet out on every line, only on those lines that are going approximately in the right direction.
- Flooding is not practical in most applications

DISTANCE VECTOR ROUTING ALGORITHM

Distance vector is the **"Dynamic Routing"** protocol. Distance vector routing algorithm is also called as **Bellman-Ford algorithm** or **Ford Fulkerson algorithm** as this algorithm is used to find the shortest route from one node to another node in the network. The routing protocol is used to calculate the best route from source to destination based on the distance or hops as its primary metric to define an optimal path. The distance vector refers to the distance to the neighbor nodes, where routing defines the routes to the established node.

The **Distance Vector routing algorithm**(DVR) shares the information of the routing table with the other routers in the network and keeps the information up-to-date to select an optimal path from source to destination

The Bellman-Ford algorithm is defined as:

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

where, dx(y)= The least distance from x to y

c(x,v)= Node x's cost from each of its neighbor v

dv(y)= Distance to each node from initial node

minv= selecting shortest distance

Each router in the network will share the distance information with the neighboring router. All the information is gathered from the neighbor routers. With each router's information, an optimal distance is calculated and stored in the routing table. This way, the process of calculating the optimal path is done using the distant vector routing protocol.

WORKING

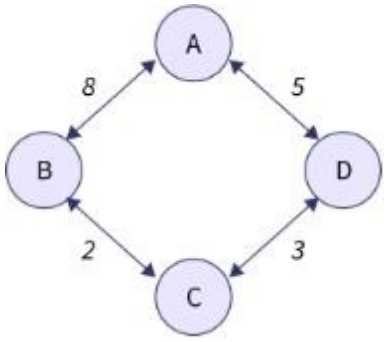
The distance vector routing algorithm works by having each router maintain a routing table, giving the best-known distance from source to destination and which route is used to get there.

These tables are updated by exchanging the information with the neighbor having a direct link. Tables contain one entry for each route, this entry contains two parts, the preferred outgoing line used to reach the destination or an estimate of the time or distance to that destination.

The metric used can be the number of hops required to reach from source to destination. Time delay in milliseconds, the router can measure it with a special echo signal which the receiver can timestamp and send as soon as possible.

The router exchanges the network topology information periodically with its neighboring node and updates the information in the routing table. The cost of reaching the destination is estimated based on the metric, and an optimal path is obtained to send data packets from source to destination.

Eg:



Step - 1

As we can see in the above diagram of the DVR network, the routers in the network start sharing their information with the neighboring routers in the network.

Routing table of A:

Destination	distant	Hop
A	0	A
B	8	B
C	infinity	-
D	5	D

Routing table of B:

Destination	distant	Hop
A	8	A
B	0	B
C	2	C
D	infinity	D

Routing table of C:

Destination	distant	Hop
A	infinity	-
B	2	B
C	0	C
D	3	D

Routing table of D :

Destination	distant	Hop
A	5	A
B	infinity	B
C	3	C
D	0	D

Step - 2

After creating the separate local table this information is shared with the neighboring node having a direct link.

For Router A:

The router A has a direct connection to neighboring routers B and D.

Destination	Vector B	Vector D
A	8	5
B	0	infinity
C	2	3
D	infinity	0

So based on the vector information of the neighboring router, the value of the router is updated.

Here the optimal distance is been calculated to reach the specific destination. First, the distance from A to B is identified, in our case which is 8, ie cost(A−>B)=8 cost(A−>D)=5

The distance to reach a destination B from router A is:

Since the cost is min from neighbor B so the router chooses the path from B. It then updates the routing information with entries (8,B).

To find a cost to reach destination C from router A we will use a similar approach.

Distance to reach destination D from A

Consequently, A's new routing table is:

Destination	distant	Hop
A	0	A
B	8	B
C	8	D
D	5	D

Similarly changes the routing table of B, C and D

Step - 3

After this, the router again exchanges the distance vector obtained in step 2 with its neighboring router.

After exchanging the distance vector, the router prepares a new routing table.

For router A new routing table is:

Destination	distant	Hop
A	0	A
B	8	B
C	8	D
D	5	D

LINK STATE ROUTING

A method in which each router shares its neighbourhood information with every other router in the internetwork.

• Each node has a complete map of the topology

• Used in packet switching networks

• Link state routing protocol: OSPF, IS-IS

• Idea behind LSR is simple and can be stated as 5 parts.

Each router must do the following

1. Discover its neighbors and learn their network addresses.
2. Measure the distance or cost metric to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to and receive packets from 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.

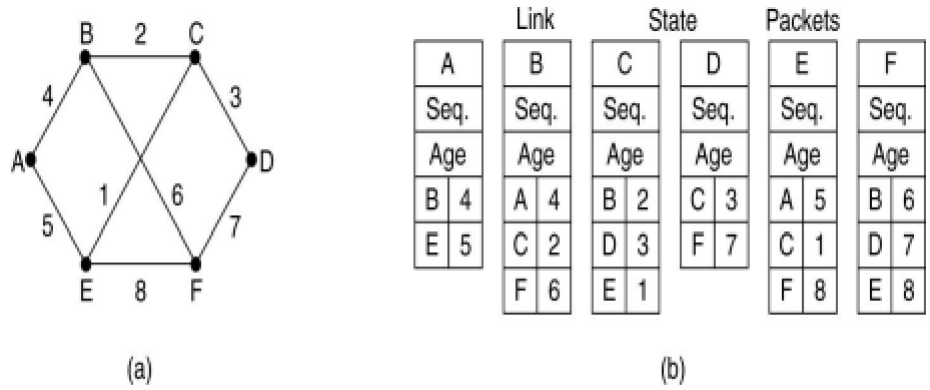
Measuring Link Cost

- Each router must know a reasonable estimate of the delay to each of its neighbors.
- The delay of links may be factored into cost
- The most direct way to determine this delay is to send a special ECHO packet over to neighbors 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.
- Issue is whether to take the load into account when measuring the delay. To factor the load in, the round-trip timer must be started when the ECHO packet is queued. To ignore the load, the timer should be started when the ECHO packet reaches the front of the queue.

Building Link State Packets(LSP)

- Once the information 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 neighbours.

- The link state packets are built either at regular intervals or when some significant event occurs
- An example network is presented with costs shown as labels on the lines.
- The corresponding link state packets for all six routers are shown.
- 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.



(a) A subnet. (b) The link state packets for this subnet.

Distributing Link State Packets

- Link state packets are to be distributed reliably.
- As the packets are distributed and installed, the routers getting the first ones will change their routes. Consequently, the different routers may be using different versions of the topology. This can lead to inconsistencies, loops, unreachable machines, and other problems.
- Flooding is used 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.

- If a packet with a sequence number lower than the highest one seen so far ever arrives, it is rejected since the router has more recent data.

Computing the shortest Routes

- Once a router has accumulated a full set of link state packets, it can construct the entire subnet graph because every link is represented. Every link is represented twice, once for each direction. The two values can be averaged or used separately.
- Dijkstra’s algorithm can be run at each router to find the shortest path to every other router.
- 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.
- For a subnet with n routers, each of which has k neighbors, the memory required to store the input data is proportional to kn.
- the computation time can be an issue. but, in many practical situations, link state routing works well.

MULTICAST ROUTING

Multicasting-

- In multicast communication, there is one source and a group of destinations.
- The relationship is one-to-many.
- In this type of communication, the source address is a unicast address, but the destination address is a group address, which defines one or more destinations. The group address identifies the members of the group.
- Some applications require that widely-separated processes work together in groups, for example, a group of processes implementing a distributed database system. In these situations, it is frequently necessary for one process to send a message to all the other members of the group. If the group is small, it can just send each other member a point-to point message. If the group is large, this strategy is expensive.
- Thus, we need a way to send messages to well-defined groups that are numerically large in size but small compared to the network as a whole.

- Sending a message to such a group is called multicasting, and its routing algorithm is called multicast routing. Multicasting requires group management. Some way is needed to create and destroy groups, and to allow processes to join and leave groups.
- How these tasks are accomplished is not based on the routing algorithm. It is based on the fact that when a process joins a group, it informs its host of this fact. It is important that routers know which of their hosts belong to which groups
- Either hosts must inform their routers about changes in group membership, or routers must query their hosts periodically. Either way, routers learn about which of their hosts are in which groups. Routers tell their neighbors, so the information propagates through the subnet.
- Multicasting- applications: distributed database, teleconferencing, distance learning, information dissemination etc

ROUTING FOR MOBILE HOST

- Millions of people have portable computers nowadays, and they generally want to read their email 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.
- Hosts that never move are said to be stationary. They are connected to the network by copper wires or fiber optics. In contrast, we can distinguish two other kinds of hosts. Migratory hosts are basically stationary hosts who move from one fixed site to another from time to time but use the network only when they are physically connected to it. Roaming hosts actually compute on the run and want to maintain their connections as they move around. We will use the term mobile hosts to mean either of the latter two categories, that is, 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.
- The world is divided up (geographically) into small units. Let us call them 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 or just wandering into the cell, his computer must register itself with the foreign agent there.

The registration procedure typically works like this:

1. Periodically, each foreign agent broadcasts a packet announcing its existence and address. A newly arrived mobile host may wait for one these messages, but if one arrives quickly enough, the mobile host can broadcast a packet saying:" are there any foreign agent around? "
2. The mobile host just register with the foreign agent, giving its home address, current data link layer address, and some security information.
3. The foreign agent contacts the mobile hosts home agent and says, one of your hosts is over here, It also includes the security information, to convince the home agent that the mobile hosts are really there.
4. 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.
5. When the foreign agent gets the acknowledgement from the home agent, it makes an entry in its table and informs the mobile hosts that it is now registered. When a packet is sent to a mobile host, it is routed to the host's home LAN because that is what the address says should be done. Here the sender, in the northwestcity of Seattle, wants to send a packet to a host normally across the United States in New York. Packets sent to the mobile host on its home LAN in New York are intercepted by thehome agent there. The home agent then looks up the mobile host's new (temporary) location and finds the address of the foreign agent handling the mobile host, in Los Angeles.

The home agent then does two things.

- First, it encapsulates the packet in the payload field of an outer packet and sends the latter to the foreign agent.
- This mechanism is called tunneling;
- After getting the encapsulated packet, the foreign agent removes the original packet from the payload field and sends it to the mobile host as a data link frame.

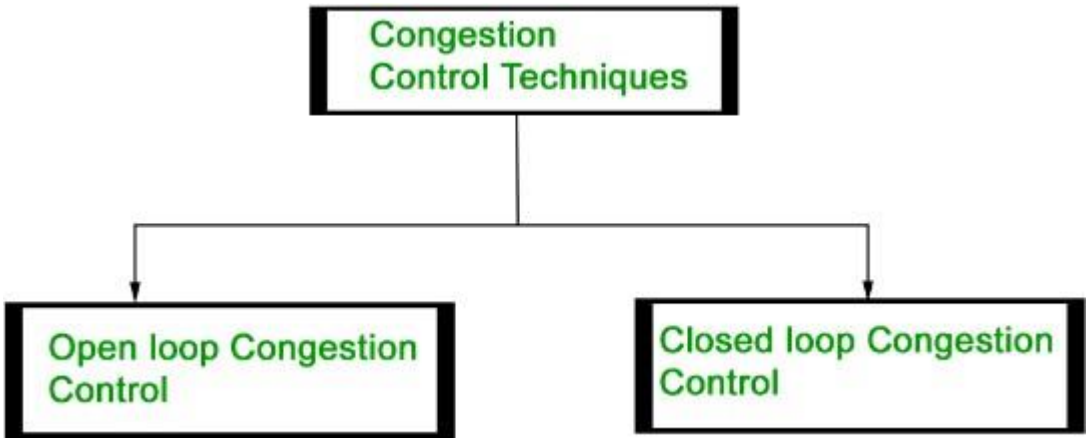
- Second, the home agent tells the sender to henceforth send packets to the mobile host by encapsulating them in the payload of packets explicitly addressed to the foreign agent instead of just sending them to the mobile host's home address.
- Subsequent packets can now be routed directly to the host via the foreign agent, bypassing the home location entirely.

SREELA CHANDRAN, AP

CONGESTION CONTROL

Congestion control is a crucial concept in computer networks. It refers to the methods used to prevent network overload and ensure smooth data flow. When too much data is sent through the network at once, it can cause delays and data loss. Congestion control techniques help manage the traffic. Congestion in a computer network happens when there is too much data being sent at the same time, causing the network to slow down.

Congestion control refers to the techniques used to control or prevent congestion. Congestion control techniques can be broadly classified into two categories:



OPEN LOOP CONGESTION CONTROL

- Policies are applied to prevent congestion before it happens.
- Congestion control is handled by either the source or the destination.
- Policies are;
    - Retransmission Policy
    - Window Policy
    - Acknowledgment Policy
    - Discarding Policy
    - Admission Policy

Retransmission Policy

- If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted.
- Retransmission increases congestion.
- The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion.

Window Policy

- The Selective Repeat window is better than the Go-Back-N window for congestion control.
- The Selective Repeat window sends the specific packets that have been lost or corrupted.

Acknowledgment Policy

- If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion.

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

- 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

► Policies are applied to alleviate congestion after it happens.

► Policies are;

- Backpressure
- Choke Packet
- Implicit Signalling
- Explicit Signalling

Backpressure

- A node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source.
- Method:
- A congested node stops receiving data from the immediate upstream node or nodes.

- This may cause the upstream node or nodes to become congested, they, in turn, reject data from their upstream node or nodes, and so on.
- Only used in Virtual Circuit network in which each node knows the upstream node

Choke Packet

- A choke packet is a packet sent by a node to the source to inform it of congestion.
- When a router in the Internet is overwhelmed with IP datagrams, it may discard some of them, but it informs the source host directly, using a source quench ICMP message

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.
- Eg: Delay in acknowledgement, or no acknowledgement for some time.

Thus the source slows down.

Explicit Signaling

- The node that experiences congestion can explicitly send a signal to the source or destination.
- Here the signal is included in the packets that carry data.
- Can occur in either the forward or the backward direction.
- Mostly in ATM (Asynchronous Transfer Mode) networks

Quality of Service (QoS.

 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 4 primary parameters: reliability, delay, jitter, and bandwidth. Together these determine the QoS (Quality of Service) the flow requires.

The needs of each flow can be characterized by 4 primary parameters: reliability, delay, jitter, and bandwidth. Together these determine the QoS (Quality of Service) the flow requires.

Techniques for Achieving Good Quality of Service

- Overprovisioning
- Buffering
- Traffic Shaping
- Leaky bucket algorithm
- Token bucket algorithm
- Resource reservation
- Proportional routing

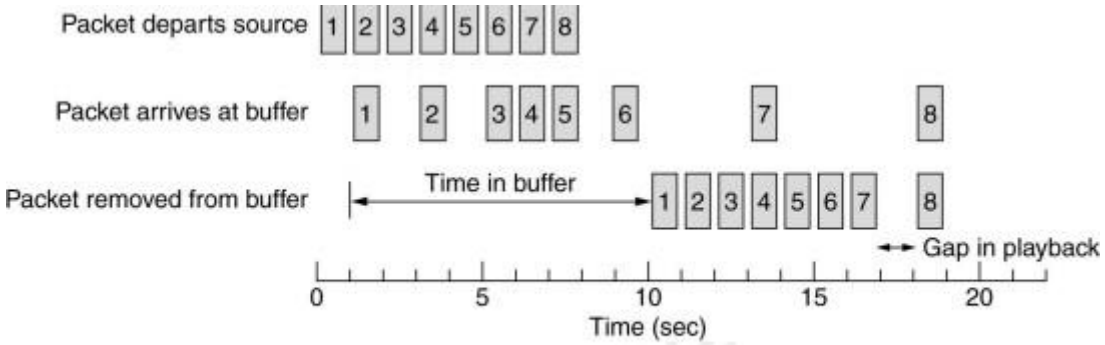
- Admission control
- Packet Scheduling

Over provisioning:

- To provide so much router capacity, buffer space, &bandwidth that the packets just fly through easily.
- The trouble with this solution is that it is expensive.
- As time goes on and designers have a better idea of how much is enough, this technique may even become practical.

Buffering:

- Flows can be buffered on the receiving side before being delivered.
- Buffering them does not affect the reliability or bandwidth, and increases the delay, but it smooths out the jitter. For audio and video on demand, jitter is the main problem, so this technique helps a lot.



Smoothing the output stream by buffering packets.

Traffic Shaping:

- One of the main causes of congestion is that traffic is often bursty.
- Traffic shaping is about regulating the average rate (and burstiness) of data transmission.
- Monitoring a traffic flow is called traffic policing.
- Agreeing to a traffic shape and policing it afterward are easier with virtual-circuit subnets than with datagram subnets.
- When a connection is set up, the user and the subnet(i.e., the customer and the carrier) agree on a certain traffic attern (i.e., shape) for that circuit. Sometimes this is called a service level agreement.
- Traffic shaping is a mechanism to control the amount and rate of traffic sent to the network
- The 2 traffic shaping techniques are :-
  1. Leaky bucket
  2. Token bucket

Leaky bucket algorithm: -

- It is the algorithm used to control congestion in network traffic
- Its working is similar to a Leaky bucket and hence the name

- Leaky bucket is a bucket with a hole at bottom
- Flow of water from bucket is at a constant rate which is independent of water entering the bucket
- If bucket is full, any additional water entering in the bucket is thrown out
- Same technique is applied to control congestion in network traffic
- Every host in the network is having a buffer with finite queue length
- Packets which are put in the buffer when buffer is full are thrown away. The buffer may drain onto the subnet either by some no. of packets per unit time or by some total number of unit time Imagine a bucket with a small hole in the bottom.
- No matter the rate at which water enters the bucket, the outflow is at a constant rate when there is any water in the bucket and zero when the bucket is empty.
- Also, once the bucket is full, any additional water entering it spills over the sides and is lost. In practice the bucket is a finite queue that outputs at a finite rate. Each host is connected to the network by an interface containing a leaky bucket, that is, a finite internal queue. If a packet arrives at the queue when it is full, the packet is discarded. In other words, if ne or more processes within the host try to send a packet when the maximum number is already queued, the new packet is discarded. This arrangement can be built into the hardware interface or simulated by the host operating system.
- Host is allowed to put one packet per clock tick onto the network.
  - This mechanism turns an uneven flow of packets from the user processes inside the host into an even flow of packets onto the network, smoothing out bursts and greatly reducing the chances of congestion. The Leaky Bucket algorithm can be implemented for packets or a constant amount of bytes, send within each time interval.
- Conceptually each network interface contains a leaky bucket. And the following steps are performed:
  - When the host has to send a packet, the packet is thrown into the bucket.
  - The bucket leaks at a constant rate, meaning the network interface transmits packets at a constantrate.
  - Burst traffic is converted to a uniform traffic by the leaky bucket.

Token bucket algorithm

- A variant on the leaky bucket
- Similar to the leaky bucket but it allows for varying output rates
- This is useful when larger burst of traffic arrive
- In this approach , a token bucket is used to manage the queue regulator that controls the rate of packet flow into the network

- Each token grants the ability to transmit a fixed no. of bytes, if the token bucket fills,newly generated tokens are discarded
- If the flow delivers more packets than the queue can store, the excess packets are discarded
- The leaky bucket algorithm enforces a rigid output pattern at the average rate, no matter how bursty the traffic is. For many applications, it is better to allow the output to speed up somewhat when large bursts arrive. One such algorithm is the token bucket algorithm. In this algorithm, the leaky bucket holds tokens, generated by a clock at the rate of one token every T sec.
- In Fig. 4 (a) a bucket is holding three tokens, with five packets waiting to be transmitted.
- For a packet to be transmitted, it must capture and destroy one token.
- In Fig. (b) three of the five packets have gotten through, but the other two are stuck waiting for two more tokens to be generated.

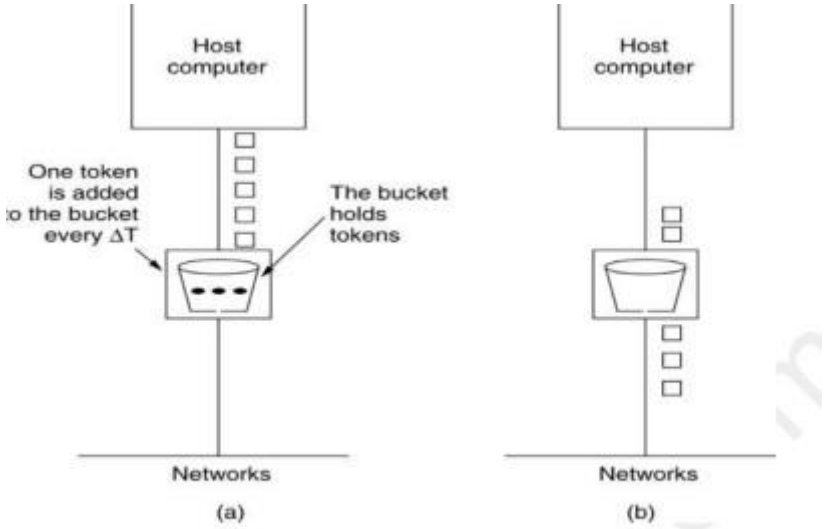


Figure: The token bucket algorithm. (a) Before. (b) After.

Resource reservation

• Once we have a specific route for a flow, it becomes possible to reserve resources along that route to make sure the needed capacity is available. Three different kinds of resources can potentially be reserved:

1. Bandwidth.

2. Buffer space.

3. CPU cycles.

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.

Proportional routing

- Most routing algorithms try to find the best path for each destination and send all traffic to that destination over the best path.
- A different approach that has been proposed to provide a higher quality of service is to split the traffic for each destination over multiple paths.
- Since routers generally do not have a complete overview of network-wide traffic, the only feasible way to split traffic over multiple routes is to use locally-available information.
- A simple method is to divide the traffic equally or in proportion to the capacity of the outgoing links.

Packet 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.
- Several scheduling techniques are designed to improve the quality of service.
- Three of them are:
    1. FIFO queuing,
    2. priority queuing,
    3. weighted fair 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.

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. Note that the system does not stop serving a queue until it is empty.

In **weighted fair queuing technique**, the packets are still assigned to different classes and admitted to different queues. The queues, however, 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.

Routing Information Protocol (RIP)

Routing Information Protocol (RIP) is a dynamic routing protocol that uses hop count as a routing metric to find the best path between the source and the destination network. It is a distance-vector routing protocol. Its works on the Network layer of the OSI model. RIP uses

port number 520.

Hop Count

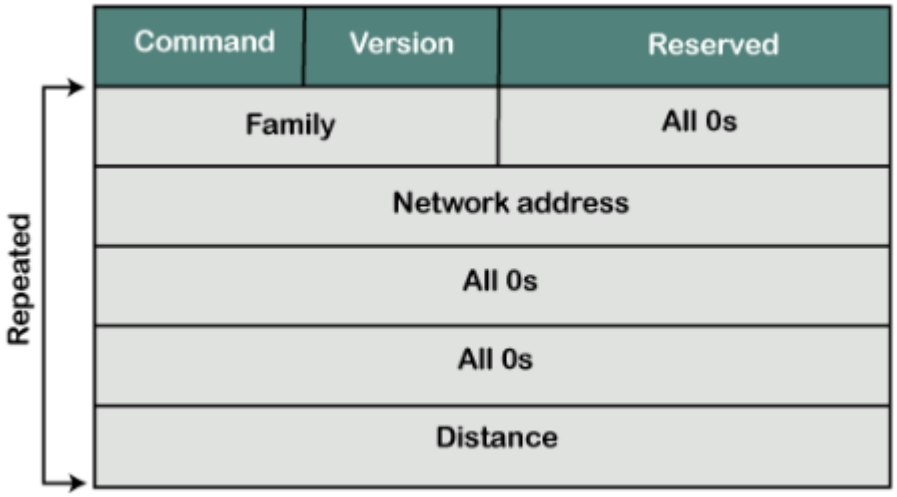
Hop count is the number of routers occurring in between the source and destination network. The path with the lowest hop count is considered as the best route to reach a network and therefore placed in the routing table. RIP prevents routing loops by limiting the number of hops allowed in a path from source and destination. The maximum hop count allowed for RIP is 15 and a hop count of 16 is considered as network unreachable.

Features of RIP

1. Updates of the network are exchanged periodically.
2. Updates (routing information) are always broadcast.
3. Full routing tables are sent in updates.
4. Routers always trust routing information received from neighbor routers. This is also known as *Routing on* rumors.

RIP Message Format

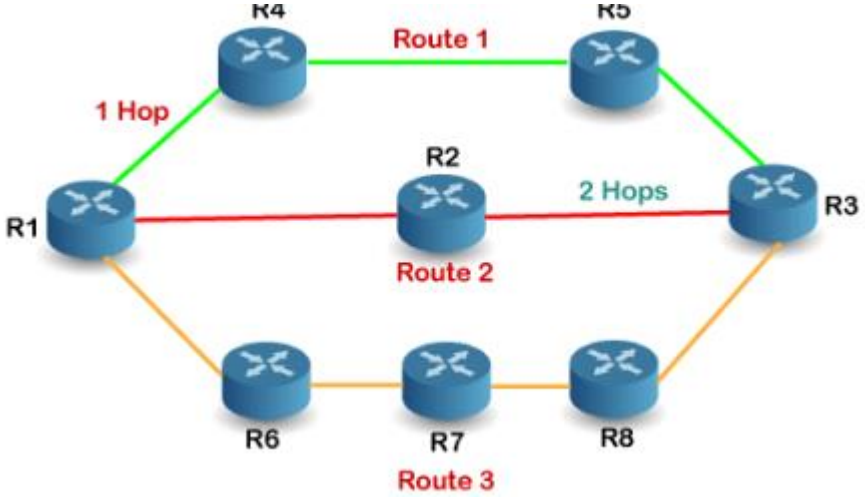
Now, we look at the structure of the RIP message format. The message format is used to share information among different routers. The RIP contains the following fields in a message:



- Command: It is an 8-bit field that is used for request or reply. The value of the request is 1, and the value of the reply is 2.
- Version: Here, version means that which version of the protocol we are using. Suppose we are using the protocol of version1, then we put the 1 in this field.
- Reserved: This is a reserved field, so it is filled with zeroes.
- Family: It is a 16-bit field. As we are using the TCP/IP family, so we put 2 value in this field.
- Network Address: It is defined as 14 bytes field. If we use the IPv4 version, then we use 4 bytes, and the other 10 bytes are all zeroes.
- Distance: The distance field specifies the hop count, i.e., the number of hops used to

reach the destination

How does the RIP work?



If there are 8 routers in a network where Router 1 wants to send the data to Router 3. If the network is configured with RIP, it will choose the route which has the least number of hops. There are three routes in the above network, i.e., Route 1, Route 2, and Route 3. The Route 2 contains the least number of hops, i.e., 2 where Route 1 contains 3 hops, and Route 3 contains 4 hops, so RIP will choose Route 2.

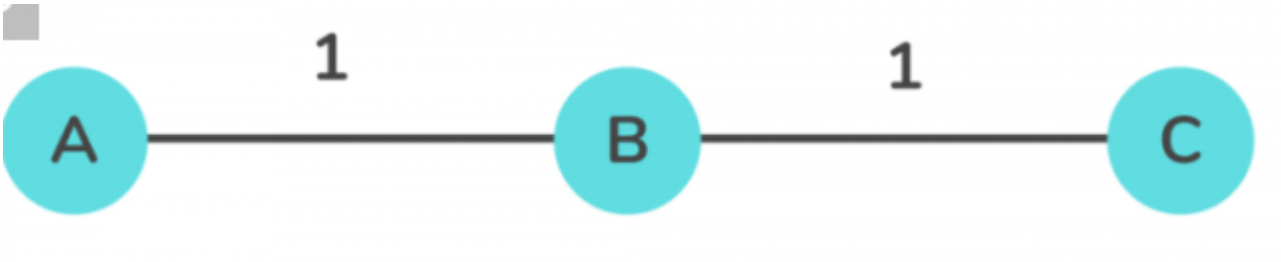
The following are the disadvantages of RIP:

- In RIP, the route is chosen based on the hop count metric. If another route of better bandwidth is available, then that route would not be chosen.
- The RIP is a classful routing protocol, so it does not support the VLSM (Variable Length Subnet Mask). The classful routing protocol is a protocol that does not include the subnet mask information in the routing updates.
- It broadcasts the routing updates to the entire network that creates a lot of traffic. In RIP, the routing table updates every 30 seconds. Whenever the updates occur, it sends the copy of the update to all the neighbors except the one that has caused the update. The sending of updates to all the neighbors creates a lot of traffic. This rule is known as a split-horizon rule.
- It faces a problem of Slow convergence. Whenever the router or link fails, then it often takes minutes to stabilize or take an alternative route; This problem is known as Slow convergence.
- RIP supports maximum 15 hops which means that the maximum 16 hops can be configured in a RIP.

Counting to infinity problem

The main issue with **D**istance Vector **R**outing (DVR) protocols is Routing Loops since Bellman-Ford Algorithm cannot prevent loops. This routing loop in the DVR network causes the Count to Infinity Problem. Routing loops usually occur when an interface goes down or two routers send updates at the same time.

Consider the above diagram, for this setup, the Bellman-Ford algorithm will work such that for each router, they will have entries for each other. Router A will infer that it can reach B at a cost of 2 units, and B will infer that it can reach C at a cost of 1 unit.



Consider the case in the above diagram, where the connection between B and C gets disconnected. In this case, B will know that it cannot get to C at a cost of 1 anymore and update its table accordingly. However, it can be possible that A sends some information to B that it is possible to reach C from A at a cost of 2. Then, since B can reach A at a cost of 1, B will erroneously update its table that it can reach C via A at a cost of 1 + 2 = 3 units. A will then receive updates from B and update its costs to 4, and so on. Thus, the process enters into a loop of bad feedback and the cost shoots towards infinity. This entire situation is called the Count to Infinity problem.