CMR INSTITUTE OF TECHNOLOGY

* UGC AUTONOMOUS *

* Approved by AICTE * Accredited by NAAC with 'A' Grade * All B.Tech programs Accredited by NBA*

(Kandlakoya (v),Medchal Road,Hyderabad - 501 401)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

II – B.Tech – II – Sem (CSE ‘C’ & CSE – DS ‘A’)

COMPUTER NETWORKS (20-CS-PC-224)
(R20 Regulations)

By
Dr.A.Nirmal Kumar

COMPUTER NETWORKS

SYLLABUS

Unit	Title/Topics	Hours
I	**Overview of the Internet, Physical layer and Data link layer**	10
	Overview of the Internet: Protocols and standards, Layering scenario, TCP/IP Protocol Suite, The OSI model, Internet history and administration, Comparison of the OSI and TCP/IP reference model. **Physical layer:** Transmission Media, Guided Media, wireless transmission Media. **Data link layer**: Design issues, CRC Codes, Elementary Data Link layer Protocols, sliding Window Protocol. ***Task:*** *Write a program to compute CRC code for the polynomials.*	
II	**Multiple Access protocols**	9
	Multiple Access protocols-Aloha, CSMA, Collision free protocols, Ethernet –Physical layer, Ethernet Mac sub layer, Data link layer switching and use of bridges, learning bridges ,Spanning tree bridges, repeaters, hubs, bridges, switches ,routers and gateways. ***Task:*** *Write a program for 1 bit collision free protocol.*	
III	**Network layer and Routing Algorithms**	5+5=10
	Part-A: Network layer: Network layer Design issues, store and forward packet switching connection less and connection oriented networks. ***Task:*** *Write a program to implement i) Character stuffing ii) Bit stuffing.*	
	Part-B: Routing Algorithms: Optimality principle, shortest path, flooding, distance vector routing, count to infinity problem, hierarchical routing, congestion control algorithms and admission control. ***Task:*** *Implement distance vector routing algorithm for obtaining routing tables at each node.*	
IV	**Internetworking and Transport Layer**	9
	Internetworking: Tunneling, internetwork Routing, Packet fragmentation, IPV4, IPV6 Protocol, IP addresses, CIDR, ICMP, ARP, RARP, DHCP. **Transport Layer**: Services provided to the upper layers elements of transport protocol-addressing connection establishment, connection release. ***Task:*** *Write a program to demonstrate ARP.*	
V	**TCP/IP and Application Layer**	10
	TCP/IP: The internet Transport protocols UD-RPC, Real time Transport protocols, The internet Transport protocols-Introduction to TCP, The TCP services model ,The TCP segment Header, The connection Establishment, The TCP Connection release, The TCP Connection management modeling, The TCP Sliding Window, The TCP Congestion Control. **Application Layer**: Introduction, Providing services, Applications layer paradigms, HTTP, FTP, electronic mail, DNS, SSH. ***Task:*** *Write a program to implement RPC.*	

TEXT BOOKS & REFERENCES

Textbooks:
1. Data Communications and Networking – Behrouz A Forouzan, Fourth Edition, TMH. 2. Computer Networks - Andrew S Tanenbaum, 4th Edition. Pearson Education/PHI
References:
1. Introduction to Data communication and Networking, Tamasi, Pearson Education 2. Computer Networking: A Top-Down Approach Featuring the Internet, James F. Kurose, Keith W. Ross, 3rd Edition, Pearson.

COURSE OUTCOMES

Upon completion of the course, the student will be able

CO 1: To outline the basics of computer networks and various layers **(Unit – I)**

CO 2: To demonstrate multiple access protocols **(Unit – II)**

CO 3: To interpret network layer and routing algorithms (Unit – III)

CO 4: To illustrate internetworking and various transport protocols **(Unit – IV)**

CO 5: To make use of various protocols of application layer **(Unit – V)**

UNIT – III

Part-B: Routing Algorithms: Optimality principle, shortest path, flooding, distance vector routing, count to infinity problem, hierarchical routing, congestion control algorithms and admission control.

Task: *Implement Distance Vector Routing Algorithm for obtaining routing tables at each node.*

Routing algorithm

- 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.

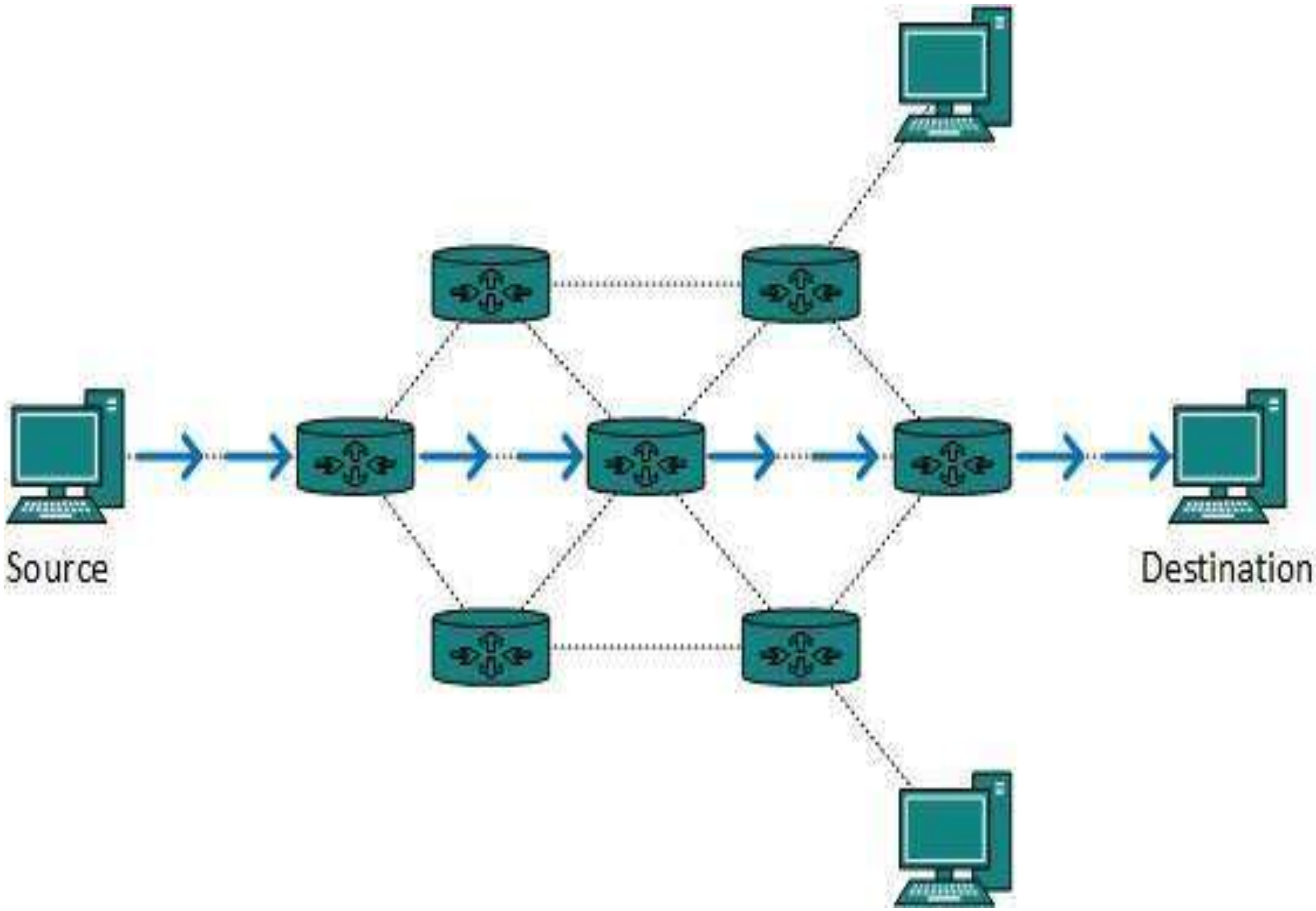
- **Random walks:** In case of random walks, a packet sent by the node to one of its neighbors randomly. An advantage of using random walks is that it uses the alternative routes very efficiently.

Network Layer Routing

- When a device has multiple paths to reach a destination, it always selects one path by preferring it over others. This selection process is termed as Routing. Routing is done by special network devices called routers or it can be done by means of software processes. The software based routers have limited functionality and limited scope.
- A router is always configured with some default route. A default route tells the router where to forward a packet if there is no route found for specific destination. In case there are multiple path existing to reach the same destination, router can make decision based on the following information:
  - Hop Count
  - Bandwidth
  - Metric
  - Prefix-length
  - Delay

Unicast routing

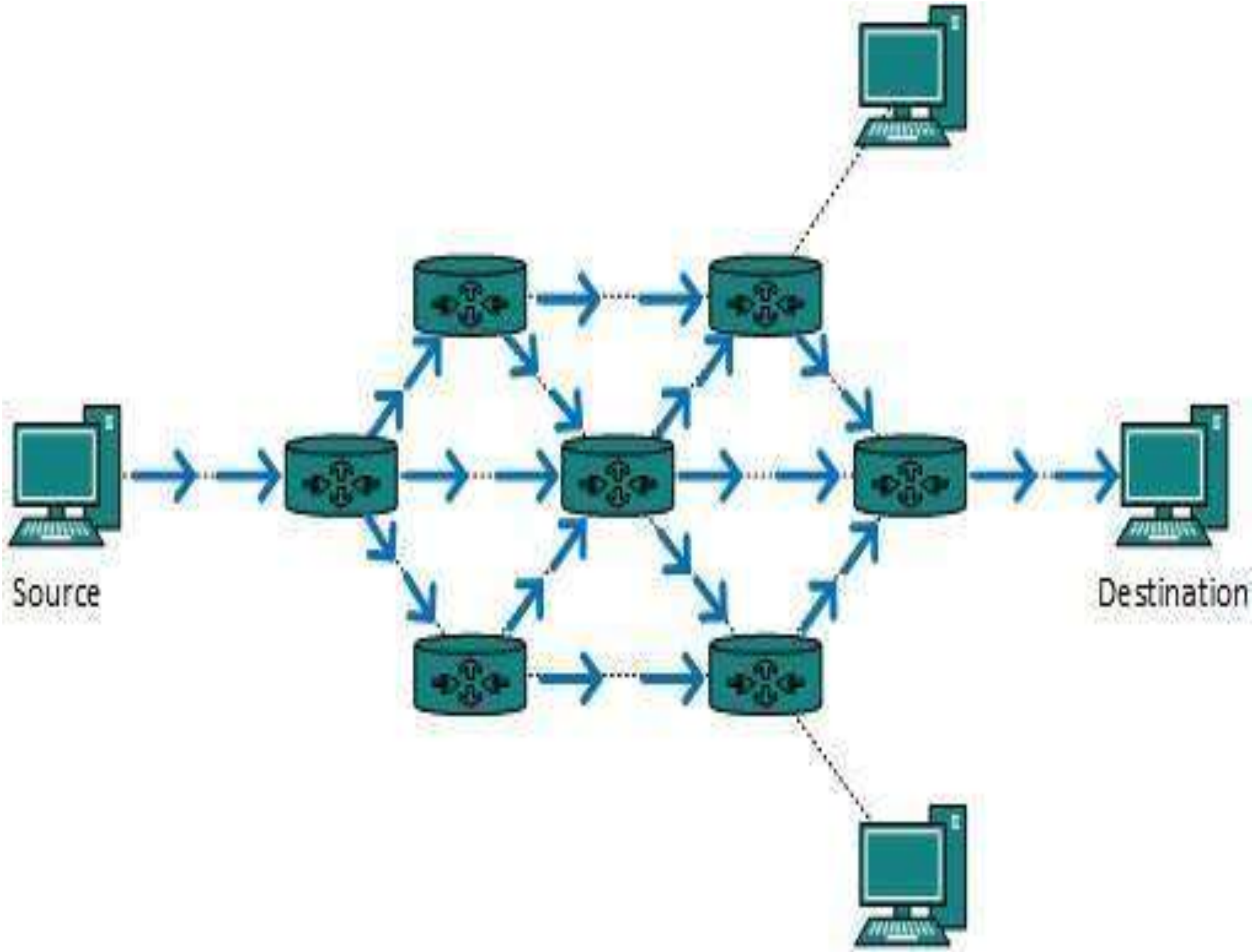
- Most of the traffic on the internet and intranets known as unicast data or unicast traffic is sent with specified destination. Routing unicast data over the internet is called unicast routing. It is the simplest form of routing because the destination is already known. Hence the router just has to look up the routing table and forward the packet to next hop.


Source
Destination

Broadcast routing

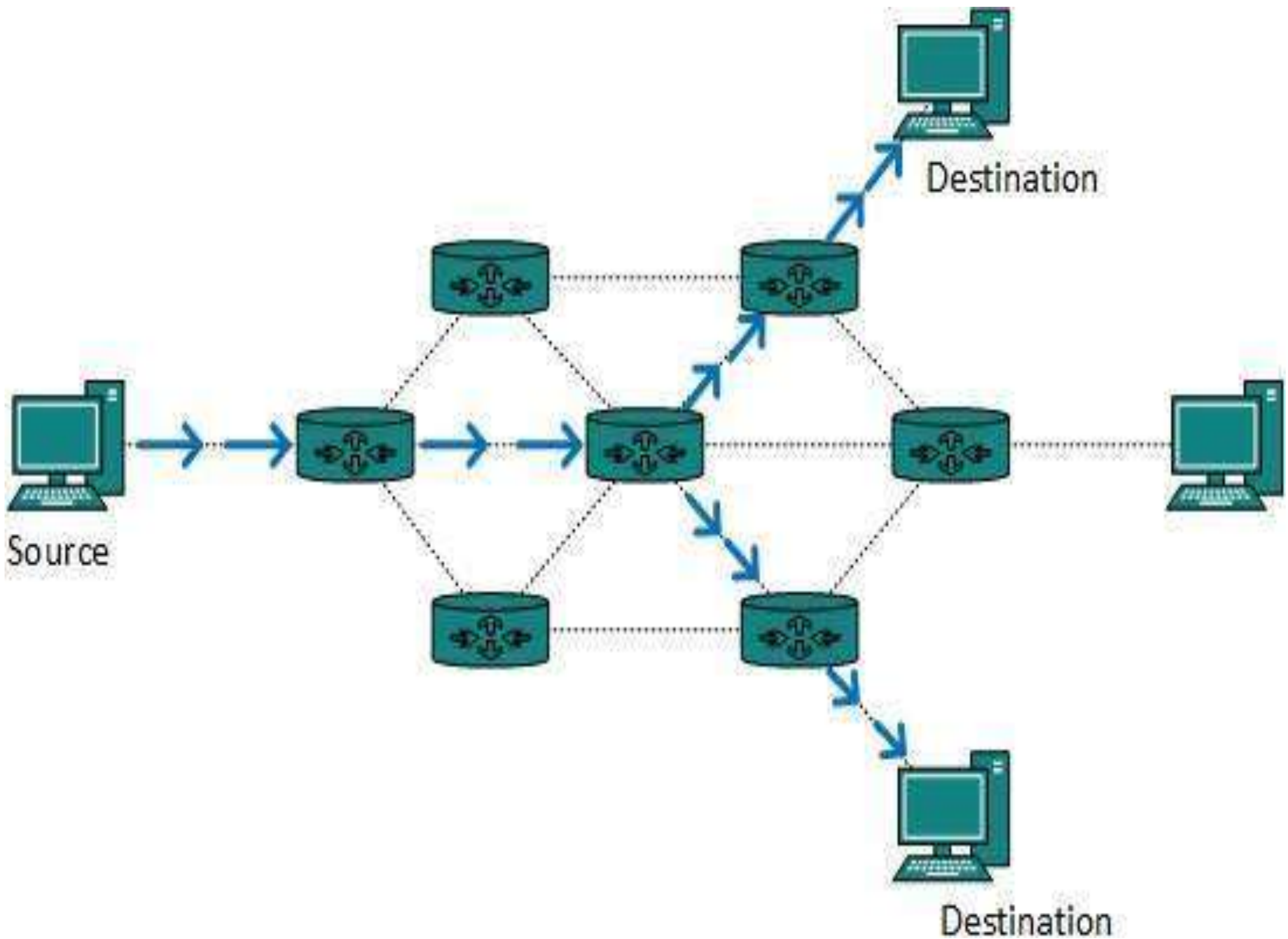
- By default, the broadcast packets are not routed and forwarded by the routers on any network. Routers create broadcast domains. But it can be configured to forward broadcasts in some special cases. A broadcast message is destined to all network devices.
- Broadcast routing can be done in two ways (algorithm):

- A router creates a data packet and then sends it to each host one by one. In this case, the router creates multiple copies of single data packet with different destination addresses. All packets are sent as unicast but because they are sent to all, it simulates as if router is broadcasting.
- This method consumes lots of bandwidth and router must destination address of each node.
- Secondly, when router receives a packet that is to be broadcasted, it simply floods those packets out of all interfaces. All routers are configured in the same way.


Source
Destination

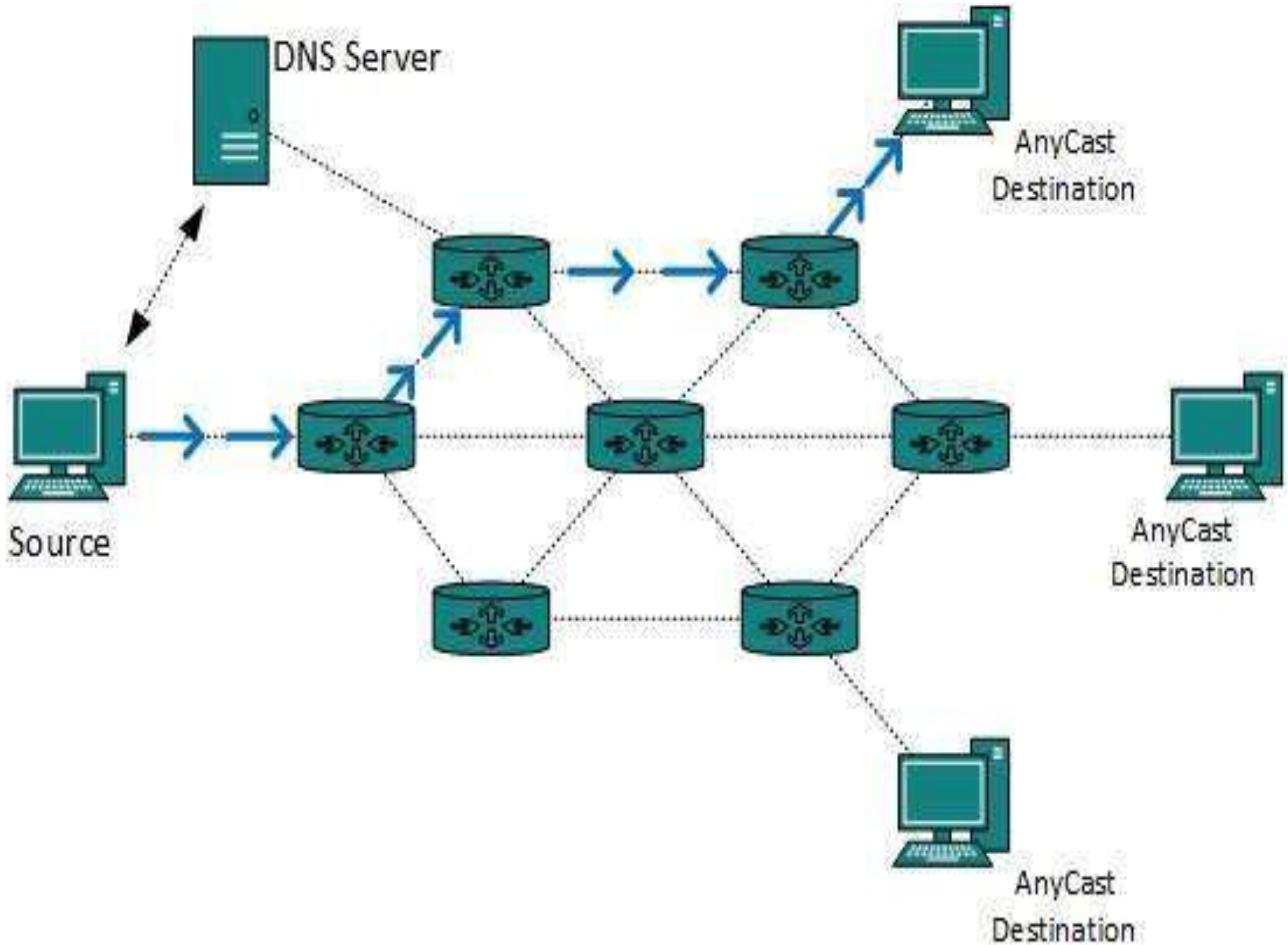
Multicast Routing

- 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 technique, to detect and discard duplicates and loops.


Destination
Source
Destination

Anycast Routing

- Anycast packet forwarding is a mechanism where multiple hosts can have same logical address. When a packet destined to this logical address is received, it is sent to the host which is nearest in routing topology.
- Anycast routing is done with help of DNS server. Whenever an Anycast packet is received it is enquired with DNS to where to send it. DNS provides the IP address which is the nearest IP configured on it.


DNS Server
AnyCast Destination
Source
AnyCast Destination
AnyCast Destination

Unicast Routing Protocols

- There are two kinds of routing protocols available to route unicast packets:

Distance Vector Routing Protocol

- Distance Vector is simple routing protocol which takes routing decision on the number of hops between source and destination. A route with less number of hops is considered as the best route. Every router advertises its set best routes to other routers. Ultimately, all routers build up their network topology based on the advertisements of their peer routers,For example Routing Information Protocol (RIP).

Link State Routing Protocol

- Link State protocol is slightly complicated protocol than Distance Vector. It takes into account the states of links of all the routers in a network. This technique helps routes build a common graph of the entire network. All routers then calculate their best path for routing purposes.for example, Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (ISIS).

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

Protocol Independent Multicast is commonly used now. It has two flavors:

- **PIM Dense Mode**

  This mode uses source-based trees. It is used in dense environment such as LAN.

- **PIM Sparse Mode**

  This mode uses shared trees. It is used in sparse environment such as WAN.

Routing Algorithms

The routing algorithms are as follows:

- Flooding

  Flooding is simplest method packet forwarding. 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.

Time to Live (TTL) can be used to avoid infinite looping of packets. There exists another approach for flooding, which is called Selective Flooding to reduce the overhead on the network. In this method, the router does not flood out on all the interfaces, but selective ones.

Shortest Path

- Routing decision in networks, are mostly taken on the basis of cost between source and destination. Hop count plays major role here. Shortest path is a technique which uses various algorithms to decide a path with minimum number of hops.

Common shortest path algorithms are:

- Dijkstra's algorithm
- Bellman Ford algorithm
- Floyd Warshall algorithm

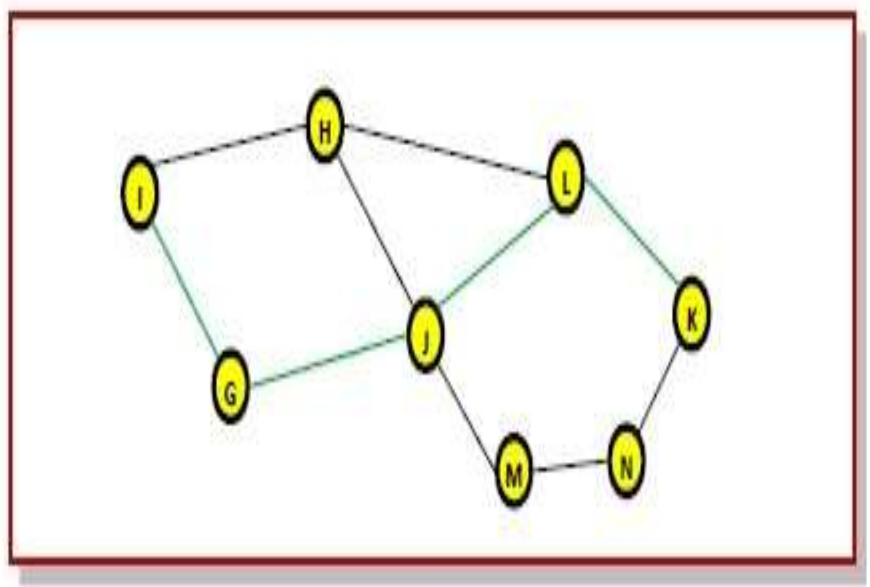
Optimality Principle

- 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 –

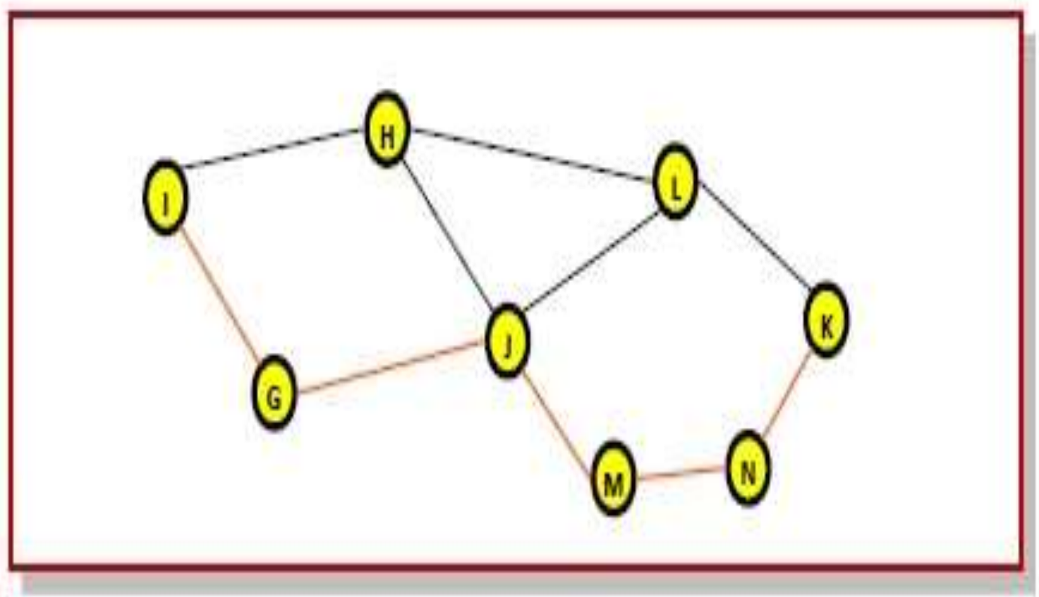
If a better route could be found between router J and router K, the path from router I to router K via J would be updated via this route. Thus, the optimal path from J to K will again lie on the optimal path from I to K.

Example

Consider a network of routers, {G, H, I, J, K, L, M, N} as shown in the figure. Let the optimal route from I to K be as shown via the green path, i.e. via the route I-G-J-L-K. According to the optimality principle, the optimal path from J to K with be along the same route, i.e. J-L-K.



Now, suppose we find a better route from J to K is found, say along J-M-N-K. Consequently, we will also need to update the optimal route from I to K as I-GJ- M-N-K, since the previous route ceases to be optimal in this situation. This new optimal path is shown line orange lines in the following figure –



Distance Vector Routing Algorithm - Introduction

Three Keys to understand the working of Distance Vector Routing Algorithm:

- **Knowledge about the whole network:** Each router shares its knowledge through the entire network. The Router sends its collected knowledge about the network to its neighbors.

- **Routing only to neighbors:** The router sends its knowledge about the network to only those routers which have direct links. The router sends whatever it has about the network through the ports. The information is received by the router and uses the information to update its own routing table.
- **Information sharing at regular intervals:** Within 30 seconds, the router sends the information to the neighboring routers.

Distance Vector Routing Algorithm

Let $d_x(y)$ be the cost of the least-cost path from node x to node y. The least costs are related by Bellman-Ford equation,

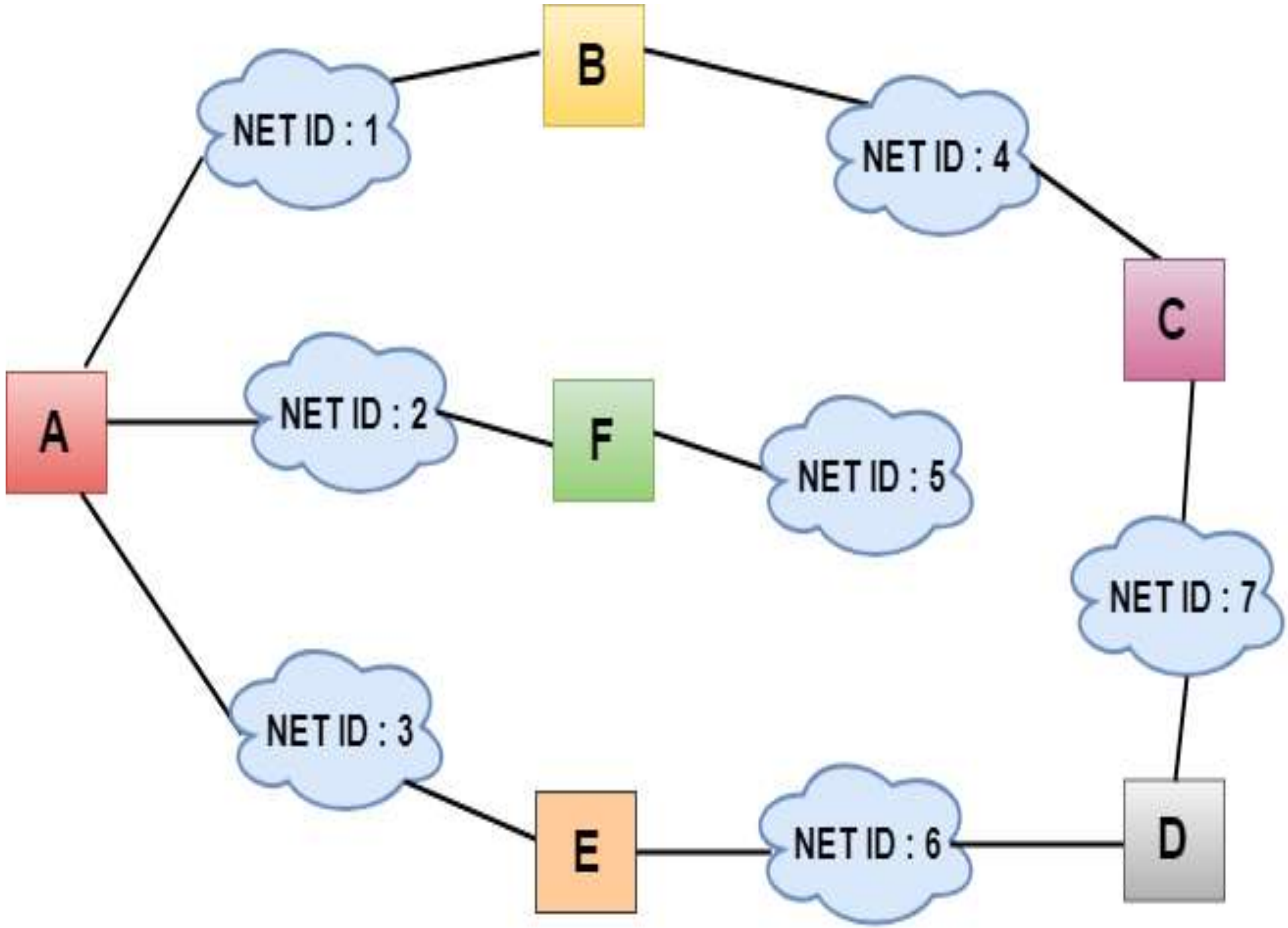
$$d_x(y) = \min_v\{c(x,v) + d_v(y)\}$$

Where the minv is the equation taken for all x neighbors. After traveling from x to v, if we consider the least-cost path from v to y, the path cost will be $c(x,v)+d_v(y)$. The least cost from x to y is the minimum of $c(x,v)+d_v(y)$ taken over all neighbors.

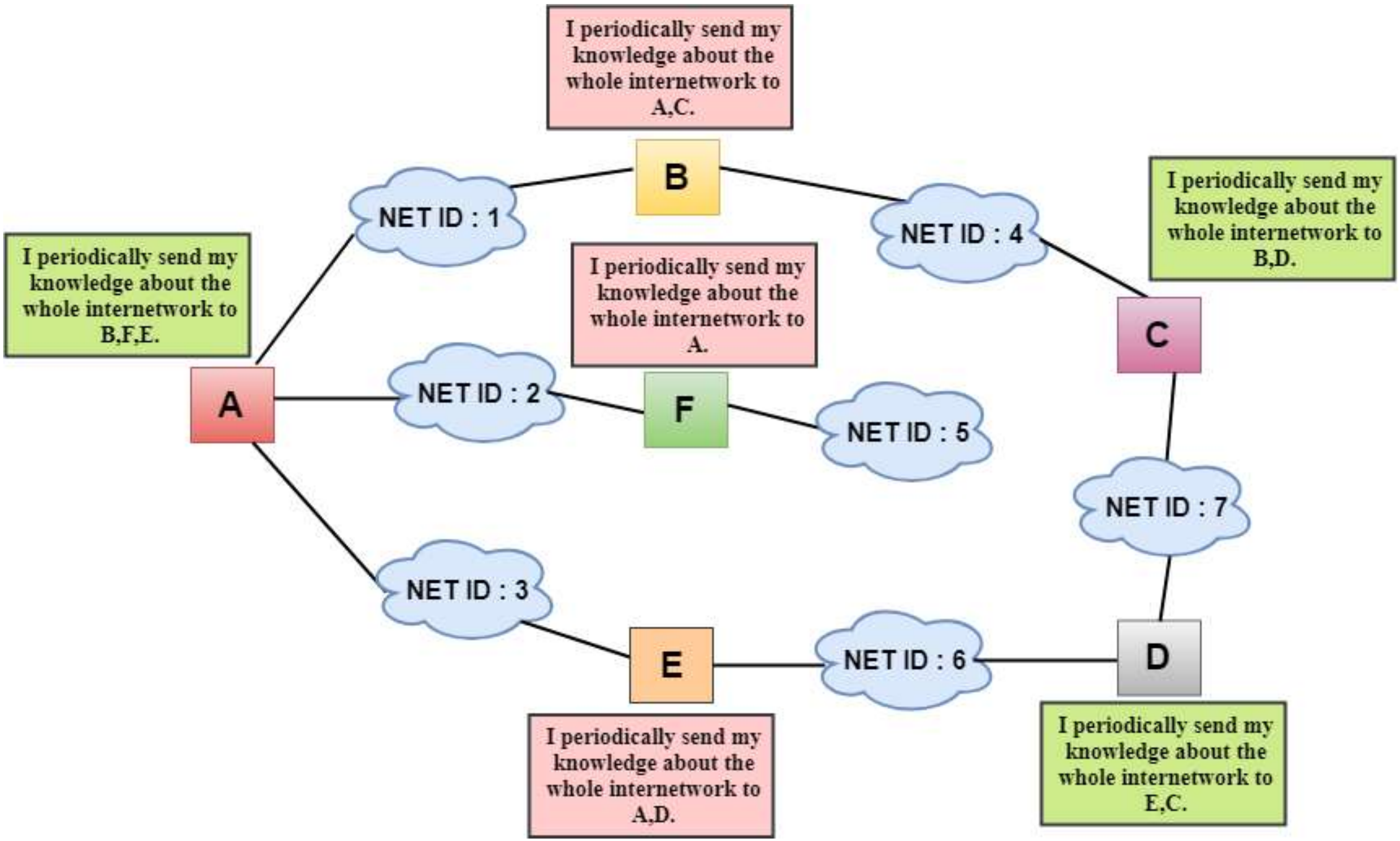
- With the Distance Vector Routing algorithm, the node x contains the following routing information:
- For each neighbor v, the cost c(x,v) is the path cost from x to directly attached neighbor, v.
- The distance vector x, i.e., D_x = [$D_x(y)$: y in N], containing its cost to all destinations, y, in N.
- The distance vector of each of its neighbors, i.e., D_v = [$D_v(y)$: y in N] for each neighbor v of x.

Distance vector routing is an asynchronous algorithm in which node x sends the copy of its distance vector to all its neighbors. When node x receives the new distance vector from one of its neighboring vector, v, it saves the distance vector of v and uses the Bellman-Ford equation to update its own distance vector.

The node x has updated its own distance vector table by using the above equation and sends its updated table to all its neighbors so that they can update their own distance vectors.


A
B
C
D
E
F
NET ID : 1
NET ID : 2
NET ID : 3
NET ID : 4
NET ID : 5
NET ID : 6
NET ID : 7

- In the above figure, each cloud represents the network, and the number inside the cloud represents the network ID.
- All the LANs are connected by routers, and they are represented in boxes labeled as A, B, C, D, E, F.
- Distance vector routing algorithm simplifies the routing process by assuming the cost of every link is one unit. Therefore, the efficiency of transmission can be measured by the number of links to reach the destination.
- In Distance vector routing, the cost is based on hop count.


I periodically send my knowledge about the whole internetwork to A,C.
B
NET ID : 1
NET ID : 4
I periodically send my knowledge about the whole internetwork to B,D.
I periodically send my knowledge about the whole internetwork to B,F,E.
I periodically send my knowledge about the whole internetwork to A.
C
A
NET ID : 2
F
NET ID : 5
NET ID : 7
NET ID : 3
E
NET ID : 6
D
I periodically send my knowledge about the whole internetwork to A,D.
I periodically send my knowledge about the whole internetwork to E,C.

In the above figure, we observe that the router sends the knowledge to the immediate neighbors. The neighbors add this knowledge to their own knowledge and sends the updated table to their own neighbors. In this way, routers get its own information plus the new information about the neighbors.

Routing Table

Two process occurs:

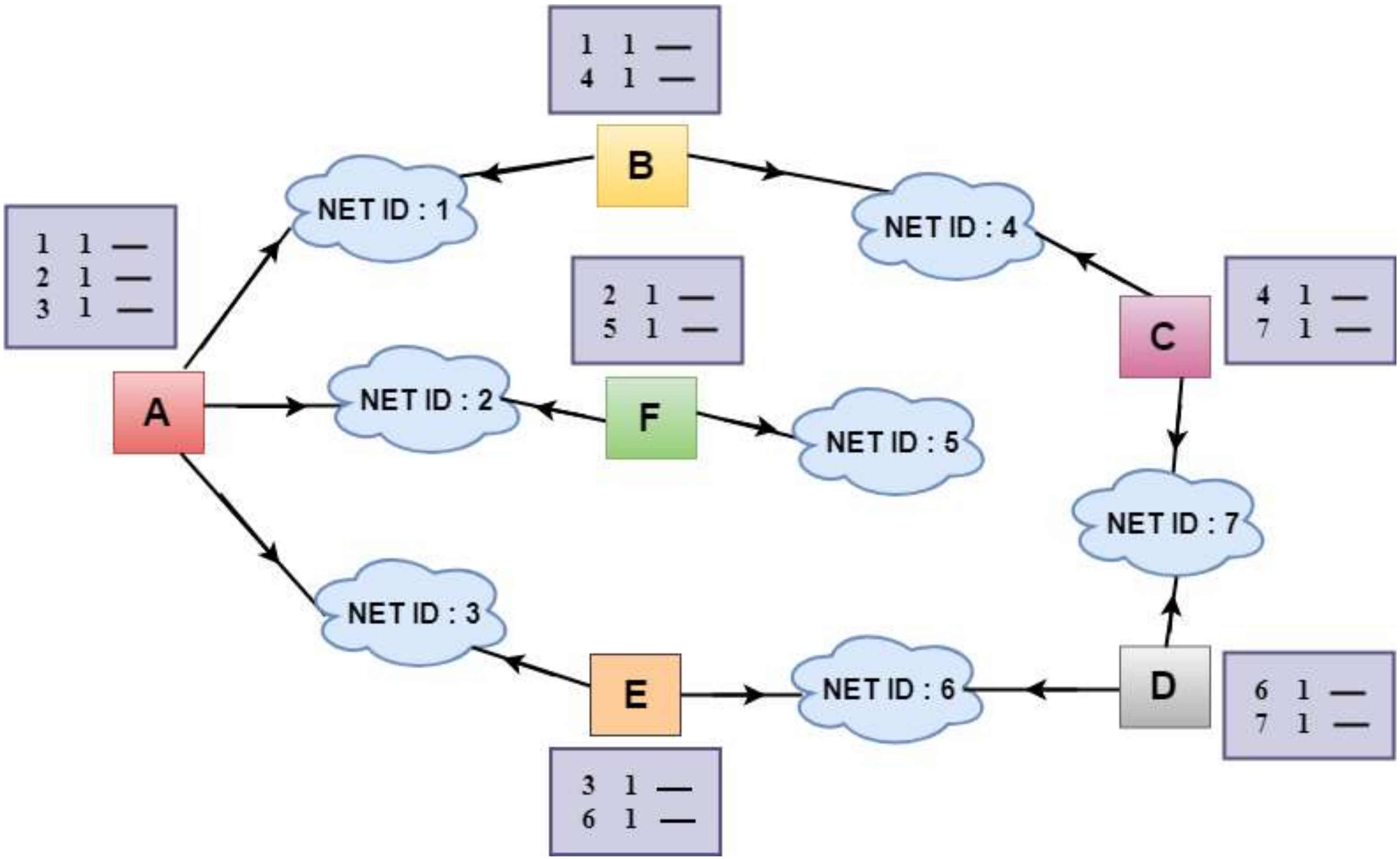
- Creating the Table
- Updating the Table

Creating the Table

- Initially, the routing table is created for each router that contains atleast three types of information such as Network ID, the cost and the next hop.

NET ID	Cost	Next Hop
- - - - -	- - - -	- - - - -
- - - - -	- - - -	- - - - -
- - - - -	- - - -	- - - - -
- - - - -	- - - -	- - - - -
- - - - -	- - - -	- - - - -

NET ID: The Network ID defines the final destination of the packet.
Cost: The cost is the number of hops that packet must take to get there.
Next hop: It is the router to which the packet must be delivered.


1 1 —
2 1 —
3 1 —
1 1 —
4 1 —
2 1 —
5 1 —
4 1 —
7 1 —
6 1 —
7 1 —
3 1 —
6 1 —
A
B
C
D
E
F
NET ID : 1
NET ID : 2
NET ID : 3
NET ID : 4
NET ID : 5
NET ID : 6
NET ID : 7

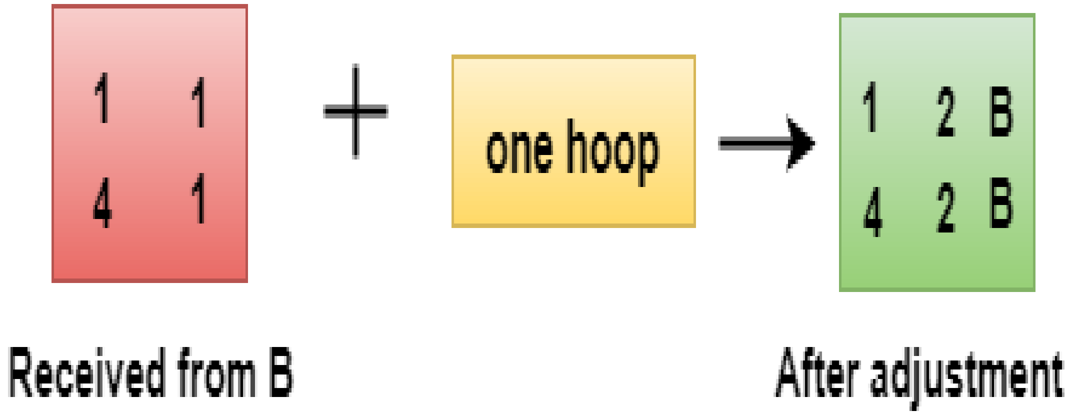
- In the above figure, the original routing tables are shown of all the routers. In a routing table, the first column represents the network ID, the second column represents the cost of the link, and the third column is empty.
- These routing tables are sent to all the neighbors.

For Example:

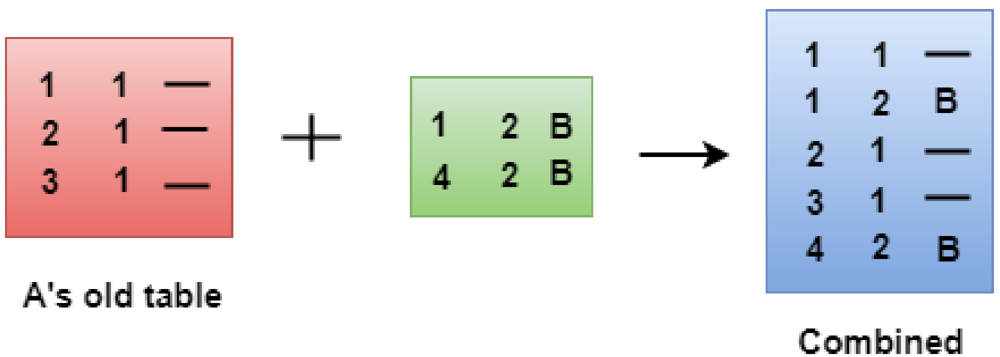
- A sends its routing table to B, F & E.
- B sends its routing table to A & C.
- C sends its routing table to B & D.
- D sends its routing table to E & C.
- E sends its routing table to A & D.
- F sends its routing table to A.

Updating the Table

- When A receives a routing table from B, then it uses its information to update the table.
- The routing table of B shows how the packets can move to the networks 1 and 4.
- The B is a neighbor to the A router, the packets from A to B can reach in one hop. So, 1 is added to all the costs given in the B's table and the sum will be the cost to reach a particular network.



After adjustment, A then combines this table with its own table to create a combined table.



The combined table may contain some duplicate data. In the above figure, the combined table of router A contains the duplicate data, so it keeps only those data which has the lowest cost. For example, A can send the data to network 1 in two ways. The first, which uses no next router, so it costs one hop. The second requires two hops (A to B, then B to Network 1). The first option has the lowest cost, therefore it is kept and the second one is dropped.

1	1	—
1	2	B
2	1	—
3	1	—
4	2	B

Combined

1	1	—
2	1	—
3	1	—
4	2	B

A's new table

The process of creating the routing table continues for all routers. Every router receives the information from the neighbors, and update the routing table.

Final routing tables of all the routers are given below:

Router A

6	2	E
1	1	–
3	1	–
4	2	B
7	3	E
2	1	–
5	2	F

Router B

6	3	E
1	1	–
3	2	A
4	1	–
7	2	C
2	2	A
5	3	A

Router C

6	2	D
1	2	B
3	3	D
4	1	–
7	1	–
2	3	B
5	4	B

Router D

6	1	–
1	3	E
3	2	E
4	2	C
7	1	–
2	3	E
5	4	E

Router E

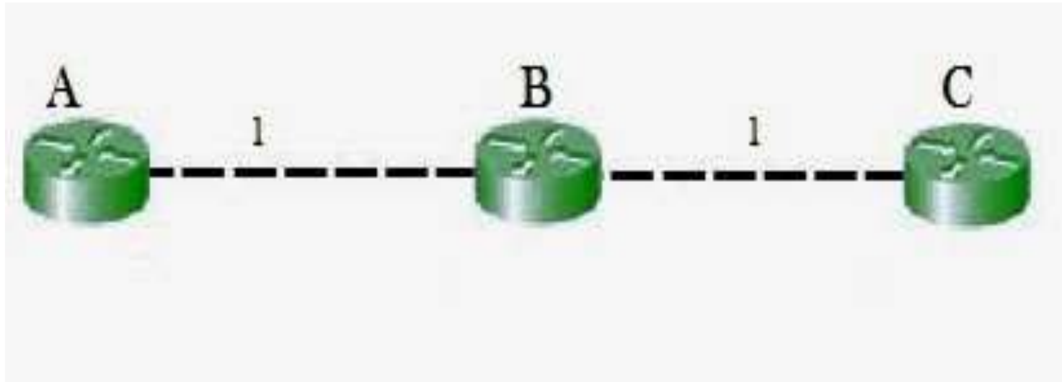
6	1	–
1	2	A
3	1	–
4	3	A
7	2	D
2	2	A
5	3	A

Router F

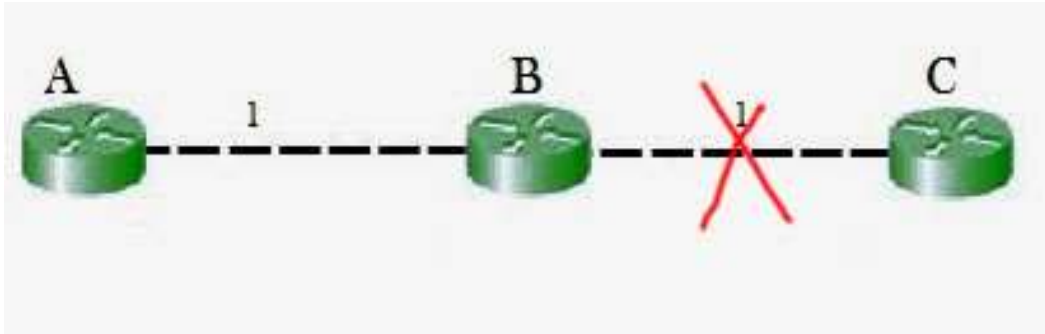
6	3	A
1	2	A
3	2	A
4	3	A
7	4	A
2	1	–
5	1	–

Count to infinity problem in Routing

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.



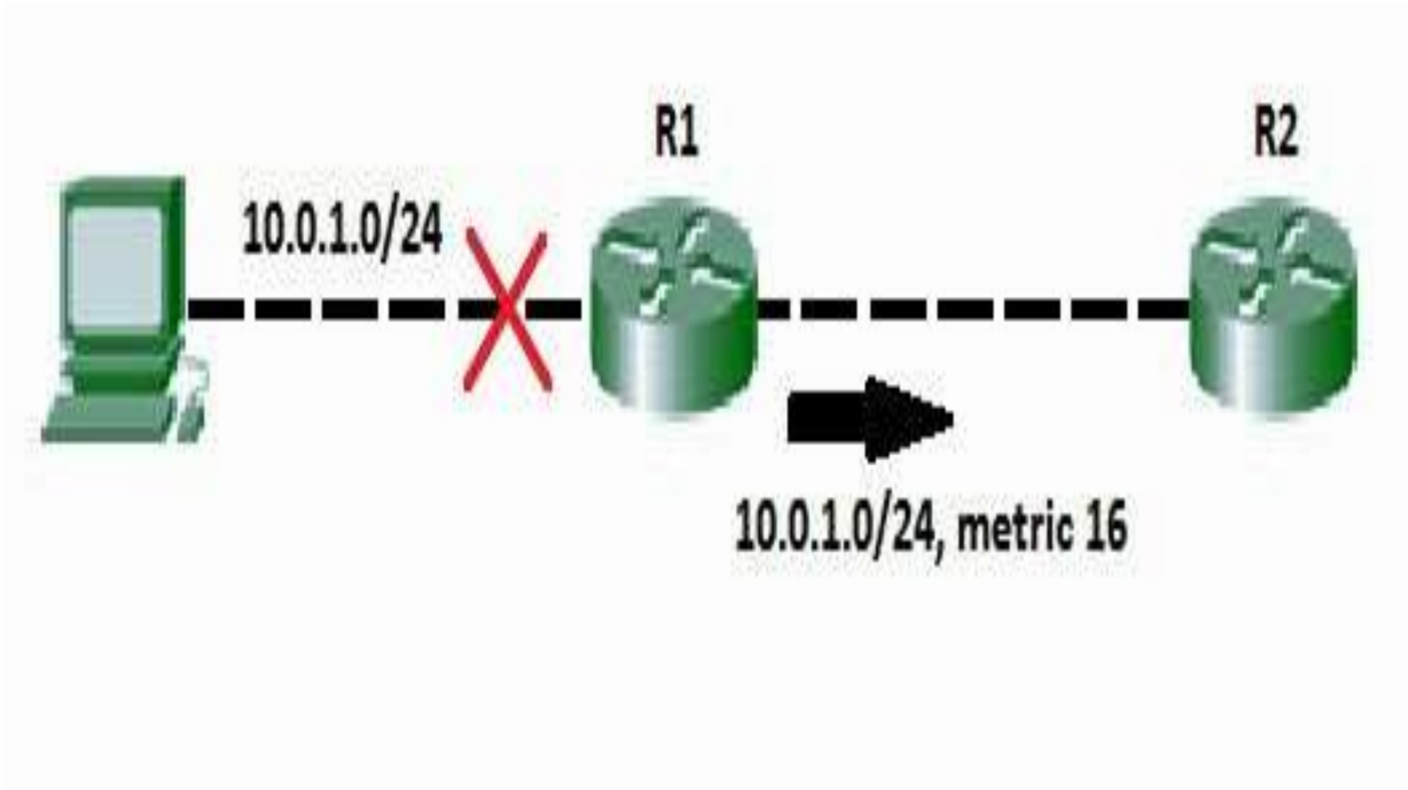
So in this example, the Bellman-Ford algorithm will converge for each router, they will have entries for each other. B will know that it can get to C at a cost of 1, and A will know that it can get to C via B at a cost of 2.



If the link between B and C is disconnected, then B will know that it can no longer get to C via that link and will remove it from its table. Before it can send any updates it’s possible that it will receive an update from A which will be advertising that it can get to C at a cost of 2. B can get to A at a cost of 1, so it will update a route to C via A at a cost of 3. A will then receive updates from B later and update its cost to 4. They will then go on feeding each other bad information toward infinity which is called as **Count to Infinity problem**.

Solution for Count to Infinity problem:-

- **Route Poisoning:** When a route fails, distance vector protocols spread the *bad news* about a route failure by poisoning the route. Route poisoning refers to the practice of advertising a route, but with a special metric value called Infinity. Routers consider routes advertised with an infinite metric to have failed. Each distance vector routing protocol uses the concept of an actual metric value that represents infinity. RIP defines infinity as 16. The main disadvantage of poison reverse is that it can significantly increase the size of routing announcements in certain fairly common network topologies.


R1
R2
10.0.1.0/24
10.0.1.0/24, metric 16

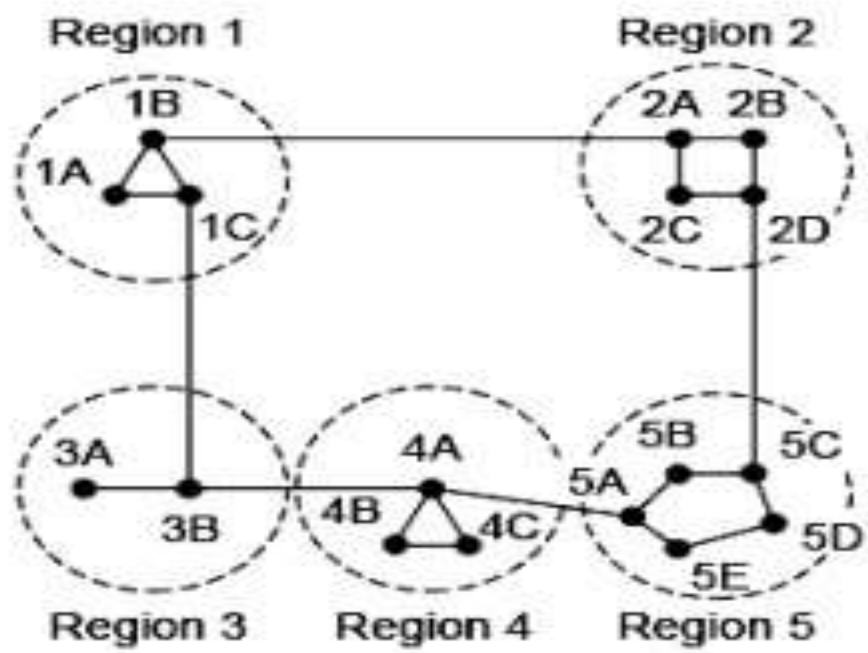
Hierarchical Routing

- In hierarchical routing, the routers are divided into regions. Each router has complete details about how to route packets to destinations within its own region. But it does not have any idea about the internal structure of other regions.
- As we know, in both LS and DV algorithms, every router needs to save some information about other routers. When network size is growing, the number of routers in the network will increase. Therefore, the size of routing table increases, then routers cannot handle network traffic as efficiently. To overcome this problem we are using hierarchical routing.

- In hierarchical routing, routers are classified in groups called regions. Each router has information about the routers in its own region and it has no information about routers in other regions. So, routers save one record in their table for every other region.
- For huge networks, a two-level hierarchy may be insufficient hence, it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups and so on.

- **Example**

Consider an example of two-level hierarchy with five regions as shown in figure –



Let see the full routing table for router 1A which has 17 entries, as shown below –

Full Table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4

When routing is done hierarchically then there will be only 7 entries as shown below –

Hierarchical Table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Congestion Control

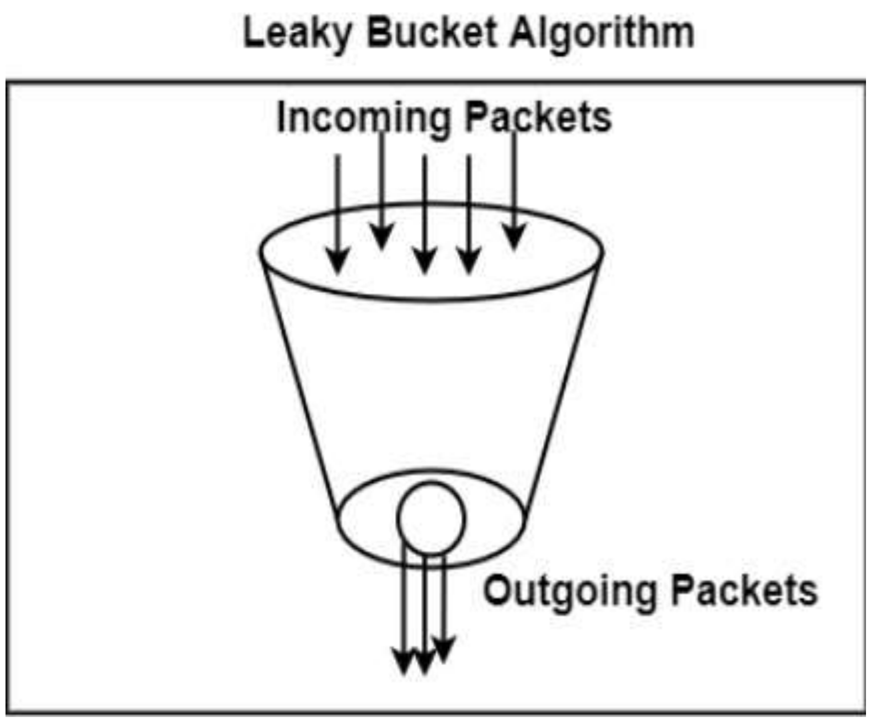
What is **congestion**?

- A state occurs in the network layer when the message traffic is so heavy that it slows down network response time.

Leaky Bucket

- The leaky bucket algorithm discovers its use in the context of network traffic shaping or rate-limiting. The algorithm allows controlling the rate at which a record is injected into a network and managing burstiness in the data rate.

- A leaky bucket execution and a token bucket execution are predominantly used for traffic shaping algorithms. This algorithm is used to control the rate at which traffic is sent to the network and shape the burst traffic to a steady traffic stream.



- In this algorithm, a bucket with a volume of, say, b bytes and a hole in the Notes bottom is considered. If the bucket is null, it means b bytes are available as storage. A packet with a size smaller than b bytes arrives at the bucket and will forward it. If the packet's size increases by more than b bytes, it will either be discarded or queued. It is also considered that the bucket leaks through the hole in its bottom at a constant rate of r bytes per second.
- The outflow is considered constant when there is any packet in the bucket and zero when it is empty. This defines that if data flows into the bucket faster than data flows out through the hole, the bucket overflows.

Token Bucket Algorithm

- The leaky bucket algorithm has a rigid output design at the average rate independent of the bursty traffic. In some applications, when large bursts arrive, the output is allowed to speed up. This calls for a more flexible algorithm, preferably one that never loses information. Therefore, a token bucket algorithm finds its uses in network traffic shaping or rate-limiting.
- It is a control algorithm that indicates when traffic should be sent. This order comes based on the display of tokens in the bucket. The bucket contains tokens. Each of the tokens defines a packet of predetermined size. Tokens in the bucket are deleted for the ability to share a packet.

- When tokens are shown, a flow to transmit traffic appears in the display of tokens. No token means no flow sends its packets. Hence, a flow transfers traffic up to its peak burst rate in good tokens in the bucket.
- Thus, the token bucket algorithm adds a token to the bucket each 1 / r seconds. The volume of the bucket is b tokens. When a token appears, and the bucket is complete, the token is discarded. If a packet of n bytes appears and n tokens are deleted from the bucket, the packet is forwarded to the network.

- When a packet of n bytes appears but fewer than n tokens are available. No tokens are removed from the bucket in such a case, and the packet is considered non-conformant. The non-conformant packets can either be dropped or queued for subsequent transmission when sufficient tokens have accumulated in the bucket.
- They can also be transmitted but marked as being non-conformant. The possibility is that they may be dropped subsequently if the network is overloaded.