UNIT 4
ROUTING
1.ROUTING

Routing is the process of path selection in any network. A computer network is made of many machines, called *nodes*, and paths or links that connect those nodes.

Communication between two nodes in an interconnected network can take place through many different paths. Routing is the process of selecting the best path using some predetermined rules.

The network layer is responsible for routing the packet from its source to the destination. The network layer is responsible for finding the best one among these possible routes.

The network layer needs to have some specific strategies for defining the best route. Routing is the concept of applying strategies and running routing protocols to create the decision-making tables for each router. These tables are called as routing tables.

TYPES

There are two different types of routing, which are based on how the router creates its routing tables:

Static routing

In static routing, a network administrator uses static tables to manually configure and select network routes. Static routing is helpful in situations where the network design or parameters are expected to remain constant. The static nature of this routing technique comes with expected drawbacks, such as network congestion. While administrators can configure fallback paths in case a link fails, static routing generally decreases the adaptability and flexibility of networks, resulting in limited network performance.

Dynamic routing

In dynamic routing, routers create and update routing tables at runtime based on actual network conditions. They attempt to find the fastest path from the source to the destination by using a dynamic routing protocol, which is a set of rules that create, maintain, and update the dynamic routing table.

The biggest advantage of dynamic routing is that it adapts to changing network conditions, including traffic volume, bandwidth, and network failure. Routing algorithms are software programs that implement different routing protocols.

They work by assigning a cost number to each link; the cost number is calculated using various network metrics.

Every router tries to forward the data packet to the next best link with the lowest cost.

DIFFERENCE BETWEEN STATIC AND DYNAMIC ROUTING:

S.NO	Static Routing	Dynamic Routing
1.	In static routing routes are user-defined.	In dynamic routing, routes are updated according to the topology.
2.	Static routing does not use complex routing algorithms.	Dynamic routing uses complex routing algorithms.
3.	Static routing provides high or more security.	Dynamic routing provides less security.
4.	Static routing is manual.	Dynamic routing is automated.
5.	Static routing is implemented in small networks.	Dynamic routing is implemented in large networks.
6.	In static routing, additional resources are not required.	In dynamic routing, additional resources are required.

2.ROUTING DESIGN GOALS

Routing algorithms, while different in nature, all have the same basic objectives. While some algorithms are better than others, all routing protocols have their advantages and disadvantages. Routing algorithms are designed with the following objectives and goals:

- Optimal Routing
- Stability
- Ease of Use
- Flexibility
- Rapid Convergence

Optimal Routing

One of the primary goals of all routing protocols is to select the most optimal path through the network from the source subnet or host to the destination subnet or host. The most optimal route depends on the metrics used by the routing protocols. A route that may be considered the best by one protocol may

not necessarily be the most optimal route from the perspective of another protocol. For example, RIP might consider a path that is only two hops long as the most optimal path to a destination network, even though the links were 64Kbps links, while advanced protocols such as OSPF and EIGRP might determine that the most optimal path to that same destination is the one traversing four routers, but using 10Gbps links, for example.

Stability

Network stability, of a lack thereof, is another major objective for routing algorithms. Routing algorithms should be stable enough to accommodate unforeseen network events, such hardware failures and even incorrect implementations. While this is typically a characteristic of all routing algorithms, the manner and time in which they respond to such events makes some better than other and thus more preferred in modern day networks.

Ease of Use

Routing algorithms are designed to be as simple as possible. In addition to also providing the capability to support complex internetwork deployments, routing protocols should take into consideration the resources required to run the algorithm. Some routing algorithms require more hardware or software resources, e.g. CPU and memory, to run than others; however, they are capable of providing more functionality than alternative simple algorithms.

Flexibility

In addition to providing routing functionality, routing algorithms should also be feature-rich, allowing them to support the different requirements encountered in different networks. It should be noted that this capability typically comes at the expense of other features, such as convergence which is described next.

Rapid Convergence

Rapid convergence is another primary objective of all routing algorithms. As stated earlier in this chapter, convergence occurs when all routers in the network have the same view and agree on optimal routes. When convergence takes a long time to occur, intermittent packet loss and loss of connectivity may be experienced between remote networks. In addition to these problems, slow convergence can result in network routing loops and outright network outages.

Optimally principle

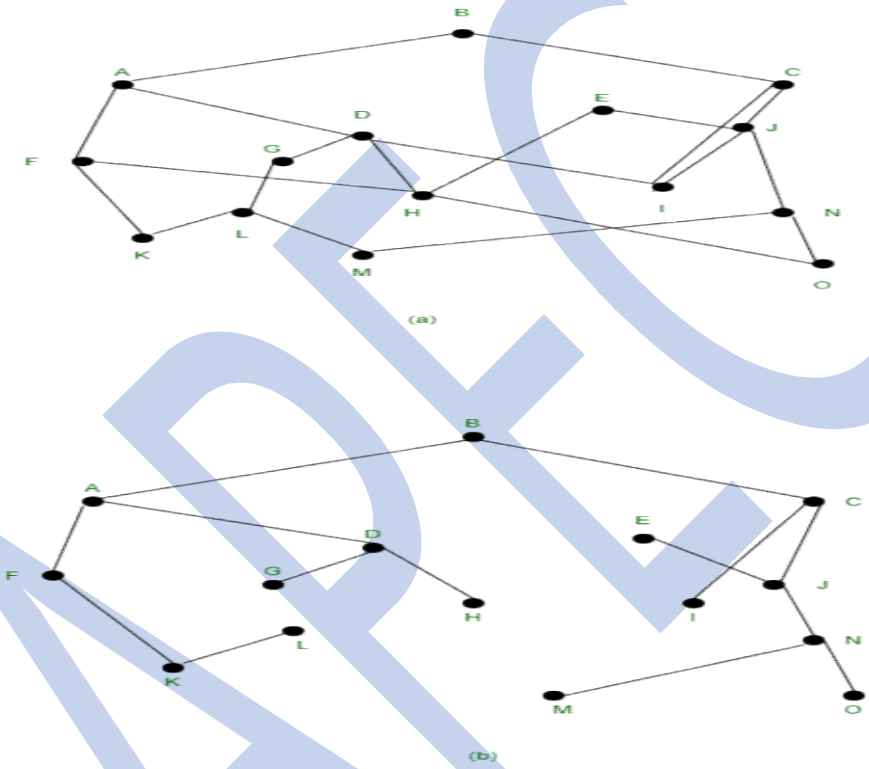
It states that if the router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route. Call the route from I to J *r1* and the rest of the route *r2*. it could be concatenated with *r1* to improve the route from I to K, contradicting our statement that *r1r2* is optimal only if a route better than r2 existed from J to K.

Sink Tree for routers :

We can see that the set of optimal routes from all sources to a given destination from a tree rooted at the destination as a directed consequence of

the optimality principle. This tree is called a **sink tree** and is illustrated in fig(1).

The sink tree is not unique also other trees with the same path lengths may exist. If we allow all of the possible paths to be chosen, the tree becomes a more general structure called a **DAG (Directed Acyclic Graph).** DAGs have no loops. We will use sink trees as a convenient shorthand for both cases. we will take technical assumption for both cases that the paths do not interfere with each other so, for example, a traffic jam on one path will not cause another path to divert.

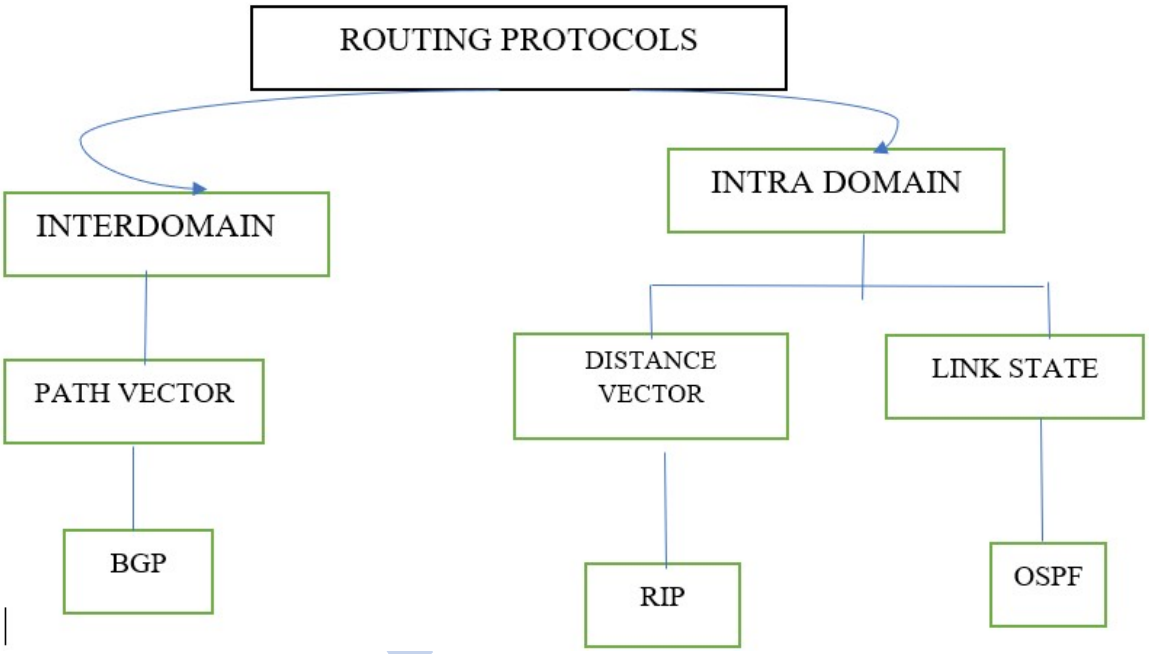


3.UNICAST ROUTING PROTOCOLS

Unicast means the transmission from a single sender to a single receiver. It is a point-to-point communication between the sender and receiver. There are various unicast protocols such as TCP, HTTP, etc. TCP is the most commonly used unicast protocol. A protocol is more than an algorithm. A protocol needs to define its domain of operation, the messages exchanged, communication between routers, and interaction with protocols in other domains.

- A routing protocol specifies how routers communicate with each other, distributing information that enables them to select routes between any two nodes on a computer network.
- Routers perform the "traffic directing" functions on the Internet; data packets are forwarded through the networks of the internet from router to router until they reach their destination computer.

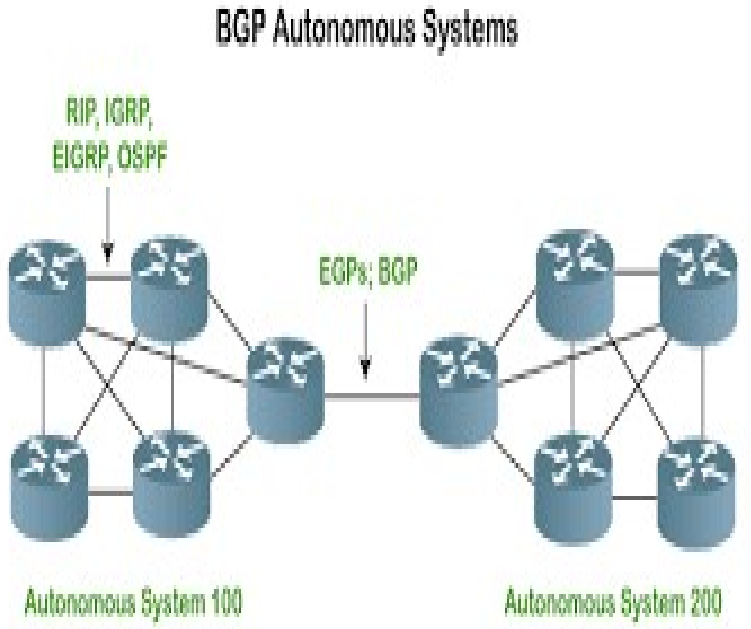
- Routing algorithms determine the specific choice of route.
- Each router has a prior knowledge only of networks attached to it directly.
- A routing protocol shares this information first among immediate neighbors,
- and then throughout the network. This way, routers gain knowledge of the
- topology of the network.
- The ability of routing protocols to dynamically adjust to changing conditions such as disabled data lines and computers and route data around obstructions is what gives the Internet its survivability and reliability.
- The specific characteristics of routing protocols include the manner in which they avoid routing loops, the manner in which they select preferred routes, using information about hop costs, the time they require to reach routing convergence, their scalability, and other factors.



INTRA AND INTER DOMAIN ROUTING

Intra-domain Routing is different from **Inter-domain Routing**. Intra domain is any protocol in which Routing algorithm works only within domains on the other hand Inter domain is any protocol in which Routing algorithm works within and between domains

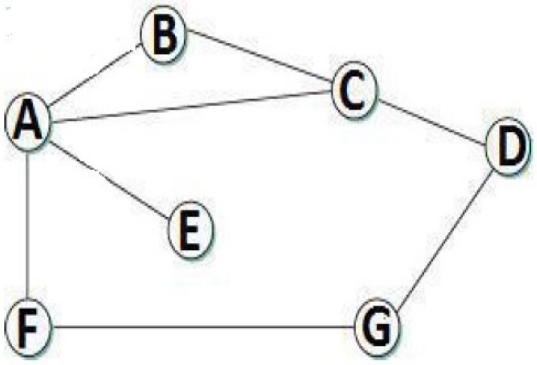
S.No	Intradomain Routing	Interdomain Routing
1.	Routing algorithm works only within domains.	Routing algorithm works within and between domains.
2.	It needs to know only about other routers within their domain.	It needs to know only about other routers within and between their domain.
3.	Protocols used in intradomain routing are known as Interior-gateway protocols.	Protocols used in interdomain routing are known as Exterior-gateway protocols.
4.	In this Routing, routing takes place within an autonomous network.	In this Routing, routing takes place between the autonomous networks.
5.	Intradomain routing protocols ignores the internet outside the AS (autonomous system).	Interdomain routing protocol assumes that the internet contains the collection of interconnected AS (autonomous systems).
6.	Some Popular Protocols of this routing are RIP (routing information protocol) and OSPF (open shortest path first).	Popular Protocols of this routing is BGP (Border Gateway Protocol) used to connect two or more AS (autonomous system).



4.DISTANCE VECTOR ROUTING

Distance vector routing is *distributed*, i.e., algorithm is run on all nodes. Each node *knows* the distance (cost) to each of its directly connected neighbors. Nodes construct a *vector* (Destination, Cost, Next Hop) and distributes to its neighbors. Nodes compute routing table of *minimum* distance to every other node via Next Hop using information obtained from its neighbors.

Initial State



In given network, cost of each link is 1 hop.

- Each node sets a distance of 1 (hop) to its immediate neighbors and cost to itself as 0.
- Distance for non-neighbors is marked as unreachable with value ∞ (infinity).

- For node A, nodes B, C, E and F are reachable, whereas nodes D and G are unreachable.

Destination	Cost	NextHop
A	0	A
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

Node A's initial table

Destination	Cost	NextHop
A	1	A
B	1	B
C	0	C
D	1	D
E	∞	—
F	∞	—
G	∞	—

Node C's initial table

Destination	Cost	NextHop
A	1	A
B	∞	—
C	∞	—
D	∞	—
E	∞	—
F	0	F
G	1	G

Node F's initial table

The initial table for all the nodes is given below

Initial Distances Stored at Each Node (Global View)							
Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Each node *sends* its initial table (distance vector) to neighbors and receives their estimate. Node *A* sends its table to nodes *B*, *C*, *E* & *F* and receives tables from nodes *B*, *C*, *E* & *F*. Each node *updates* its routing table by comparing with each of its neighbor's.
For each destination, Total Cost is computed as:

***Total Cost* = Cost (*Node* to *Neighbor*) + Cost (*Neighbor* to *Destination*)**

If Total Cost < Cost then
Cost* = Total Cost and Next Hop = *Neighbor

Node A learns from C's table to reach node D and from F's table to reach node G. Total Cost to reach node D via C = Cost (A to C) + Cost (C to D)
Cost = 1 + 1 = 2. Since 2 < ∞, entry for destination D in A's table is changed to (D, 2, C)
Total Cost to reach node G via F = Cost (A to F) + Cost (F to G) = 1 + 1 = 2

Since 2 < ∞, entry for destination G in A's table is changed to (G, 2, F). Each node builds complete routing table after few exchanges amongst its neighbors.

Node A's final routing table

Destination	Cost	NextHop
A	0	A
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

System stabilizes when all nodes have complete routing information, i.e., **convergence.** Routing tables are exchanged *periodically or* in case of *triggered update*. The final distances stored at each node is given below:

Final Distances Stored at Each Node (Global View)

Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

Updation of Routing Tables

There are two different circumstances under which a given node decides to send a routing update to its neighbors.

1. ***Periodic Update***

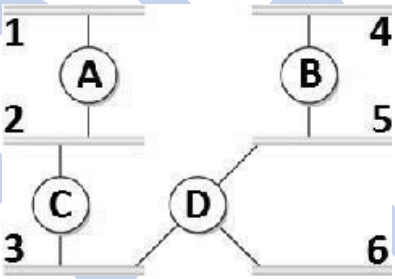
   In this case, each node automatically sends an update message every so often, even if nothing has changed. The frequency of these periodic updates varies from protocol to protocol, but it is typically on the order of several seconds to several minutes.

2. ***Triggered Update***

   In this case, whenever a node notices a link failure or receives an update from one of its neighbors that causes it to change one of the routes in its routing
   table. Whenever a node's routing table changes, it sends an update to its neighbors, which may lead to a change in their tables, causing them to send an update to their neighbors.

ROUTING INFORMATION PROTOCOL (RIP)

RIP is an intra-domain routing protocol based on distance-vector algorithm.



Routers *advertise* the cost of reaching networks. Cost of reaching each link is 1 hop. For example, router *C* advertises to *A* that it can reach network 2, *3* at cost 0 (directly connected), networks *5*, *6* at cost 1 and network *4* at cost 2. Each router *updates* cost and next hop for each network number. Infinity is defined as 16, i.e., any route cannot have more than 15 hops.
Therefore, RIP can be implemented on small-sized networks only.

Advertisements are sent every 30 seconds or in case of triggered update.

Command - It indicates the packet type. Value 1 represents a request packet. Value 2 represents a response packet.
Version - It indicates the RIP version number. For RIPv1, the value is 0x01.
Address Family Identifier - When the value is 2, it represents the IP protocol.
IP Address - It indicates the destination IP address of the route. It can be the addresses of only the natural network segment.
Metric - It indicates the hop count of a route to its destination.

Count-To-Infinity (or) Loop Instability Problem

Suppose link from node *A* to *E* goes *down*.

Node *A* advertises a distance of ∞ to *E* to its neighbors. Node B receives periodic update from C before A's update reaches B. Node *B* updated by *C*,

concludes that *E* can be reached in 3 hops via *C*. Node *B* advertises to *A* as 3 hops to reach *E*. Node *A* in turn updates *C* with a distance of 4 hops to *E* and so on. Thus, nodes update each other until cost to *E* reaches *infinity*, i.e., *no convergence*. Routing table does not stabilize. This problem is called *loop instability* or *count to infinity*

Solution to Count-To-Infinity (or) Loop Instability Problem:

Infinity is redefined to a small number, say 16. Distance between any two nodes can be 15 hops maximum. Thus, distance vector routing *cannot be used* in large networks. When a node updates its neighbors, it does not send those routes it learned from each neighbor back to that neighbor. This is known as **split horizon**. **Split horizon with poison reverse** allows nodes to advertise routes it learnt from a node back to that node, but with a warning message

5.LINK STATE ROUTING

Each node knows *state* of link to its neighbors and *cost*. Nodes create an update packet called *link-state packet* (LSP) that contains:

- ID of the node
- List of neighbors for that node and associated cost
- 64-bit Sequence number
- Time to live

Link-State routing protocols rely on two mechanisms:

Reliable flooding of link-state information to all other nodes

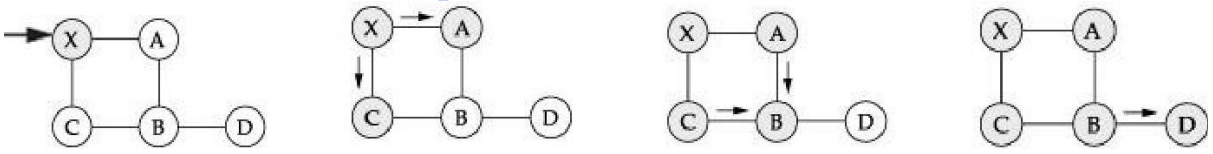
Route calculation from the accumulated link-state knowledge

Reliable Flooding

Each node *sends* its LSP out on each of its directly connected links. When a node receives LSP of another node, checks if it has an LSP already for that node. If not, it stores and forwards the LSP on all other links except the incoming

one. Else if the received LSP has a *bigger* sequence number, then it is stored and

forwarded. Older LSP for that node is *discarded*. Otherwise discard the received LSP, since it is not latest for that node. Thus, recent LSP of a node eventually *reaches* all nodes, i.e., reliable *flooding*.



Flooding of LSP in a small network is as follows:

When node *X* receives *Y*'s LSP (*fig a*), it floods onto its neighbors *A* and *C* (*fig b*). Nodes *A* and *C* forward it to *B*, but does not send it back to *X* (*fig c*). Node *B* receives two copies of LSP with same sequence number. Accepts one

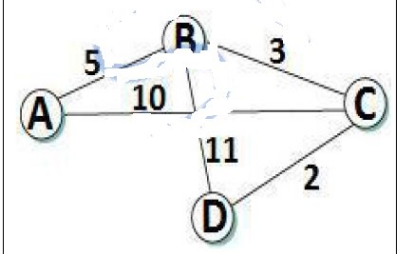
LSP and forwards it to *D* (*fig d*). Flooding is complete. LSP is generated either *periodically* or when there is a *change* in the topology.

Route Calculation

Each node knows the entire topology, once it has LSP from every other node. Forward search algorithm is used to compute routing table from the received LSPs. Each node maintains two lists, namely Tentative and Confirmed with entries of the form (Destination, Cost, Next Hop).

DIJKSTRA'S SHORTEST PATH ALGORITHM
(FORWARD SEARCH ALGORITHM)

1. Each host maintains two lists, known as ***Tentative*** and ***Confirmed***
2. Initialize the Confirmed list with an entry for the Node (Cost = 0).
3. Node just added to Confirmed list is called Next. Its LSP is examined.
4. For each neighbor of Next, calculate cost to reach each neighbor as Cost (Node to Next) + Cost (Next to Neighbor).
   a. If Neighbor is neither in Confirmed nor in Tentative list, then add
   (Neighbor, Cost, Next Hop) to Tentative list.
   b. If Neighbor is in Tentative list, and Cost is less than existing then
   replace the entry with (Neighbor, Cost, Next Hop).
5. If Tentative list is empty then *stop*, otherwise move *least* cost entry from
   Tentative list to Confirmed list. Go to *Step 2*.



Step	Confirmed	Tentative	Comments
1	(D,0,–)		Since D is the only new member of the confirmed list, look at its LSP.
2	(D,0,–)	(B,11,B) (C,2,C)	D's LSP says we can reach B through B at cost 11, which is better than anything else on either list, so put it on Tentative list; same for C.
3	(D,0,–) (C,2,C)	(B,11,B)	Put lowest-cost member of Tentative (C) onto Confirmed list. Next, examine LSP of newly confirmed member (C).
4	(D,0,–) (C,2,C)	(B,5,C) (A,12,C)	Cost to reach B through C is 5, so replace (B,11,B). C's LSP tells us that we can reach A at cost 12.
5	(D,0,–) (C,2,C) (B,5,C)	(A,12,C)	Move lowest-cost member of Tentative (B) to Confirmed, then look at its LSP.
6	(D,0,–) (C,2,C) (B,5,C)	(A,10,C)	Since we can reach A at cost 5 through B, replace the Tentative entry.
7	(D,0,–) (C,2,C) (B,5,C) (A,10,C)		Move lowest-cost member of Tentative (A) to Confirmed, and we are all done.

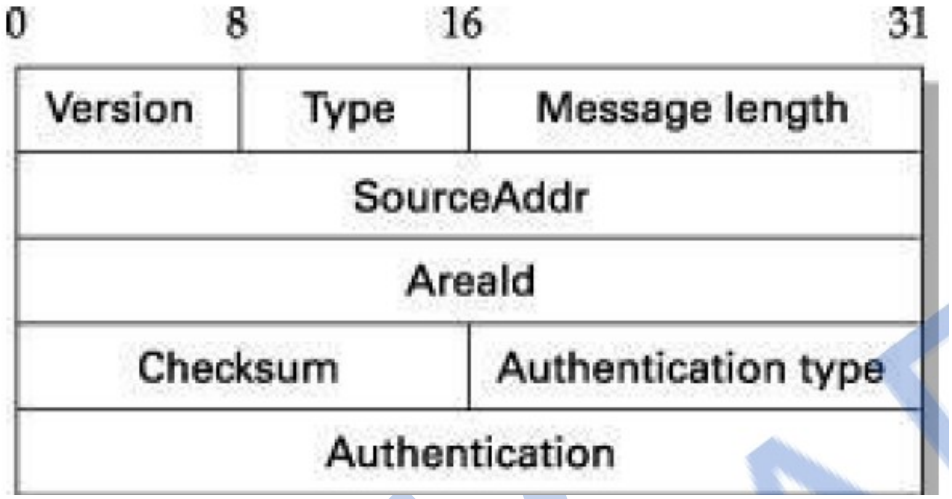
OPEN SHORTEST PATH FIRST PROTOCOL (OSPF)

OSPF is a non-proprietary widely used link-state routing protocol. OSPF Features are:

Authentication—Malicious host can collapse a network by advertising to reach every host with cost 0. Such disasters are averted by authenticating routing updates.

Additional hierarchy—Domain is partitioned into areas, i.e., OSPF is more scalable.

Load balancing—Multiple routes to the same place are assigned same cost. Thus, traffic is distributed evenly.



Link State Packet Format

***Version* —** represents the current version, i.e., 2.

***Type* —** represents the type (1–5) of OSPF message.

T*ype 1* - "hello" message,

T*ype 2* - request,

T*ype 3* – send,

T*ype 4* - acknowledge the receipt of link state messages,

T*ype 5* - reserved

***Source Addr* —** identifies the sender

***Area Id* —** 32-bit identifier of the area in which the node is located

***Checksum* —** 16-bit internet checksum

PATH VECTOR ROUTING (PVR)

Path-vector routing is an asynchronous and distributed routing algorithm. The Path-vector routing is not based on least-cost routing. The best route is determined by the source using the policy it imposes on the route. In other words, the source can control the path. Path-vector routing is not actually used in an internet, and is mostly designed to route a packet between ISPs.

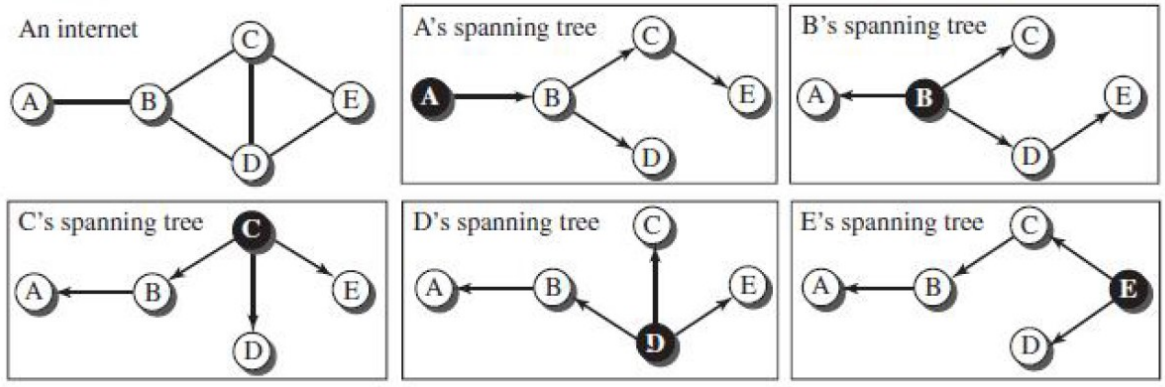
Spanning Trees

- In path-vector routing, the path from a source to all destinations is determined by the *best* spanning tree.
- The best spanning tree is not the least-cost tree.
- It is the tree determined by the source when it imposes its own policy.
- If there is more than one route to a destination, the source can choose the route that meets its policy best.
- A source may apply several policies at the same time. One of the common policies uses the minimum number of nodes to be visited.
- Another common policy is to avoid some nodes as the middle node in a route.

- The spanning trees are made, gradually and asynchronously, by each node.
- When a node is booted, it creates a *path vector* based on the information it can obtain about its immediate neighbor.
- A node sends greeting messages to its immediate neighbors to collect these pieces of information. Each node, after the creation of the initial path vector, sends it to all its immediate neighbors.
- Each node, when it receives a path vector from a neighbor, updates its path
- vector using the formula
- The policy is defined by selecting the *best* of multiple paths.
- Path-vector routing also imposes one more condition on this equation.
- If Path (**v**, ***y***) includes ***x***, that path is discarded to avoid a loop in the path.
- In other words, ***x*** does not want to visit itself when it selects a path to ***y***.

Example:

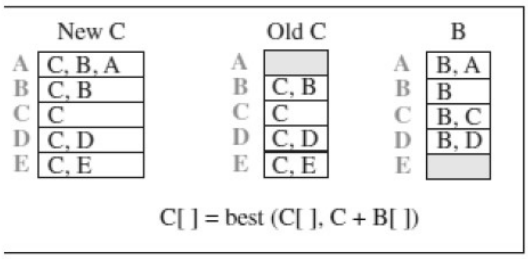
- The Figure below shows a small internet with only five nodes.
- Each source has created its own spanning tree that meets its policy.
- The policy imposed by all sources is to use the minimum number of nodes to reach a destination.
- The spanning tree selected by A and E is such that the communication does not pass through D as a middle node.
- Similarly, the spanning tree selected by B is such that the communication does not pass through C as a middle node.



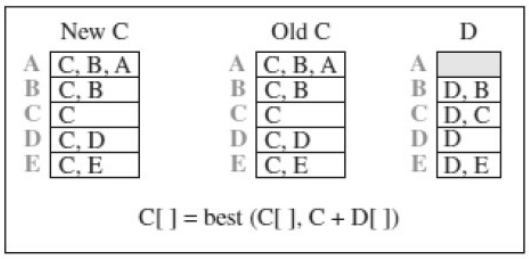
Updating Path Vectors

- The Figure below shows the path vector of node C after two events.
- In the first event, node C receives a copy of B's vector, which improves its vector: now it knows how to reach node A.
- In the second event, node C receives a copy of D's vector, which does not change its vector.

- The vector for node C after the first event is stabilized and serves as its forwarding table.



Event 1: C receives a copy of B's vector



Event 2: C receives a copy of D's vector

BORDER GATEWAY PROTOCOL (BGP)

It is used to Exchange routing information for the internet and is the protocol used between ISP which are different. The protocol can connect together any internetwork of autonomous system using an arbitrary topology. The only requirement is that each AS have at least one router that is able to run BGP and that is router connect to at least one other AS's BGP router. BGP's main function is to exchange network reach-ability information with other BGP systems. Border Gateway Protocol constructs an autonomous systems' graph based on the information exchanged between BGP routers.

Characteristics of Border Gateway Protocol (BGP):

- Inter-Autonomous System Configuration: The main role of BGP is to provide communication between two autonomous systems. BGP supports Next-Hop Paradigm.
- Coordination among multiple BGP speakers within the AS (Autonomous System).
- Path Information: BGP advertisement also include path information, along with the reachable destination and next destination pair.
- Policy Support: BGP can implement policies that can be configured by the administrator. For ex:- a router running BGP can be configured to distinguish between the routes that are known within the AS and that which are known from outside the AS.
- Runs Over TCP.
- BGP conserve network Bandwidth.
- BGP supports CIDR.
- BGP also supports Security.

Functionality of Border Gateway Protocol (BGP):

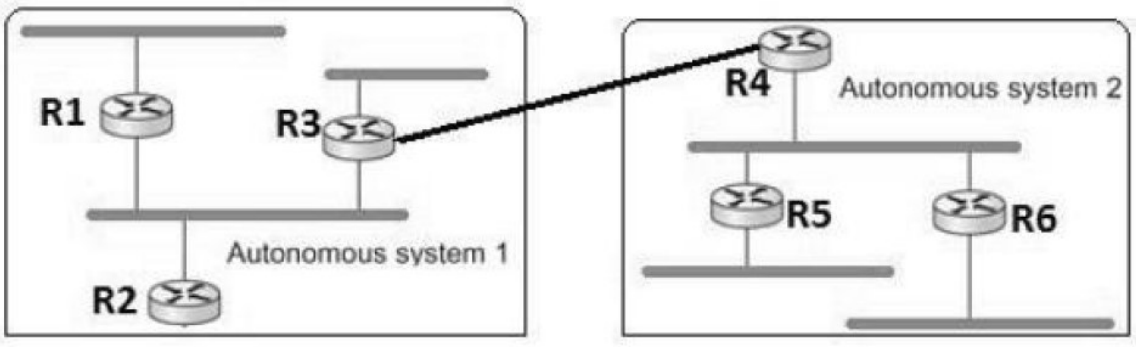
BGP peers performs 3 functions, which are given below.

- The first function consists of initial peer acquisition and authentication. both the peers established a TCP connection and perform message exchange that guarantees both sides have agreed to communicate.
- The second function mainly focus on sending negative or positive reach-ability information.
- The third function verifies that the peers and the network connection between them are functioning correctly.

BGP Route Information Management Functions:

- **Route Storage:** Each BGP stores information about how to reach other networks.
- **Route Update:** In this task, Special techniques are used to determine when and how to use the information received from peers to properly update the routes.
- **Route Selection:** Each BGP uses the information in its route databases to select good routes to each network on the internet network.
- **Route advertisement:** Each BGP speaker regularly tells its peer what is known about various networks and methods to reach them.

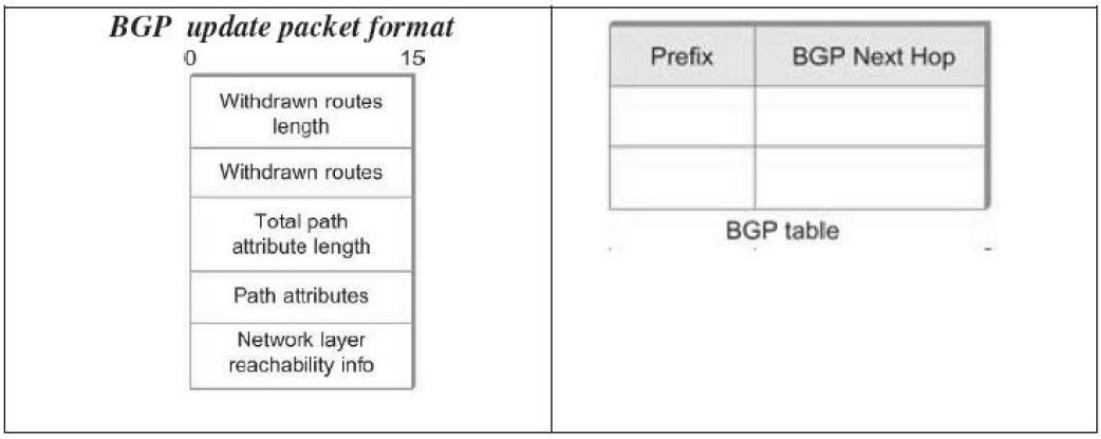
The Border Gateway Protocol version (BGP) is the only interdomain routing protocol used in the Internet today. BGP4 is based on the path-vector algorithm. It provides information about the reachability of networks in the Internet. BGP views internet as a set of autonomous systems interconnected Arbitrarily



- Each AS have a *border router* (gateway), by which packets enter and leave that AS. In above figure, *R3* and *R4* are border routers. One of the routers in each autonomous system is designated as BGP *speaker*.
- BGP Speaker *exchange* reachability information with other BGP speakers, known as *external* BGP session. BGP advertises complete *path* as enumerated list of AS (path vector) to reach a particular network.
- Paths must be without any *loop,* i.e., AS list is unique. For *example*, backbone network advertises that network 128.96 and 192.4.153 can be

reached along the path <*AS1*, *AS2, AS4*>. If there are *multiple* routes to a destination, BGP speaker chooses one based on policy.

- Speakers *need not* advertise any route to a destination, even if one exists. Advertised paths can be *cancelled*, if a link/node on the path goes down. This negative advertisement is known as *withdrawn* route. Routes are not repeatedly sent. If there is no change, *keep alive* messages are sent.



iBGP - interior BGP

- A Variant of BGP
- Used by routers to update routing information learnt from other speakers to routers inside the autonomous system.

- Each router in the AS is able to determine the appropriate next hop for all prefixes.

6.GLOBAL INTERNET

A global routing algorithm computes the least-cost path between a source and destination using complete, global knowledge about the network. That is, the algorithm takes the connectivity between all nodes and all link costs as inputs. This then requires that the algorithm somehow obtain this information before actually performing the calculation. The calculation itself can be run at one site (a centralized global routing algorithm) or replicated at multiple sites. The key distinguishing feature here, however, is that a global algorithm has complete information about connectivity and link costs. In practice, algorithms with global state information are often referred to as link-state (LS) algorithms, since the algorithm must be aware of the cost of each link in the network.

COMPARISION OF RIP AND OSPF

RIP	OSPF
RIP Stands for Routing	OSPF stands for Open Shortest Path

RIP	OSPF
Information Protocol.	First.
RIP works on the Bellman-Ford algorithm.	OSPF works on Dijkstra algorithm.
It is a Distance Vector protocol and it uses the distance or hops count to determine the transmission path.	It is a link-state protocol and it analyzes different sources like the speed, cost and path congestion while identifying the shortest path.
It is used for smaller size organizations.	It is used for larger size organizations in the network.
It allows a maximum of 15 hops.	There is no such restriction on the hop count.
It is not a more intelligent dynamic routing protocol.	It is a more intelligent routing protocol than RIP.
The networks are classified as areas and tables here.	The networks are classified as areas, sub-areas, autonomous systems, and backbone areas here.
Its administrative distance is 120.	Its administrative distance is 110.
RIP uses UDP (User Datagram Protocol) Protocol.	OSPF works for IP (Internet Protocol) Protocol.
It calculates the metric in terms of Hop Count.	It calculates the metric in terms of bandwidth.

7.MULTICASTING BASICS

In multicasting, there is one source and a group of destinations. Multicast supports efficient delivery to multiple destinations. The relationship is one to many or many-to-many.

- **One-to-Many (Source Specific Multicast)**
  - Radio station broadcast
  - Transmitting news, stock-price
  - Software updates to multiple hosts
- **Many-to-Many (Any Source Multicast)**
  - Multimedia teleconferencing
  - Online multi-player games
  - Distributed simulations

In this type of communication, the source address is a unicast address, but the destination address is a group address. The group address defines the members of the group. In multicasting, a multicast router may have to send out copies of the same datagram through more than one interface. Hosts that are members of a group receive copies of any packets sent to that group's multicast address

- A host can be in multiple groups
- A host can join and leave groups
- A host signals its desire to join or leave a multicast group by communicating with its local router using a special protocol.
- In IPv4, the protocol is Internet Group Management Protocol (IGMP)
- In IPv6, the protocol is Multicast Listener Discovery (MLD)

IGMP OR MLD PROTOCOL

- Hosts communicate their desire to *join* / *leave* a multicast group to a router using Internet Group Message Protocol (IGMP) in IPv4 or Multicast Listener Discovery (MLD) in IPv6.
- Provides multicast routers with information about the membership status of hosts connected to the network.
- Enables a multicast router to create and update list of loyal members for each group.

MULTICAST ADDRESSING

- Multicast address is associated with a group, whose members are dynamic.
- Each group has its own IP multicast address. IP addresses reserved for multicasting are Class D in IPv4 (Class D 224.0.0.1 to 239.255.255.255), 1111 1111 prefix in IPv6.
- Hosts that are members of a group receive copy of the packet sent when destination contains group address.

MULTICASTING VERSUS MULTIPLE UNICASTING

Multicasting starts with a single packet from the source that is duplicated by the routers. The destination address in each packet is the same for all duplicates. Only a single copy of the packet travels between any two routers

- In **multiple unicasting**, several packets start from the source.
- If there are three destinations, for example, the source sends three packets, each with a different unicast destination address.
- There may be multiple copies traveling between two routers

NEED FOR MULTICAST

A source needs to send a separate packet with the identical data to each member of the group. Source needs to keep track of the IP address of each member in the group. *Using IP multicast.* Sending host does not send multiple copies of the packet. A host sends a single copy of the packet addressed to the group's multicast address. The sending host does not need to know the individual unicast IP address of each member

TYPES OF MULTICASTING

- ***Source-Specific Multicast*** - In *source-specific* multicast (one-to-many model), receiver specifies multicast group and sender from which it is interested to receive packets. Example: Internet radio broadcasts.
- ***Any Source Multicast*** - Supplements *any source* multicast (many-to-many model).

MULTICAST APPLICATIONS

- Access to Distributed Databases
- Information Dissemination
- Teleconferencing.
- Distance Learning

8.MULTICAST ROUTING

- To support multicast, a router must additionally have multicast forwarding tables that indicate, based on multicast address, which links to use to forward the multicast packet.
- Unicast forwarding tables collectively specify a set of paths.
- Multicast forwarding tables collectively specify a set of trees -Multicast distribution trees.
- Multicast routing is the process by which multicast distribution trees are
- determined.
- To support multicasting, routers *additionally* build multicast forwarding
- tables.
- Multicast forwarding table is a tree structure, known as ***multicast distribution trees.***

- Internet multicast is implemented on physical networks that support broadcasting by *extending* forwarding functions.

MULTICAST ROUTING PROTOCOLS

Internet multicast is implemented on physical networks that support broadcasting by *extending forwarding functions.* Major multicast routing protocols are:

1. Distance-Vector Multicast Routing Protocol (DVMRP)
2. Protocol Independent Multicast (PIM)

1. Distance Vector Multicast Routing Protocol

- The DVMRP, is a routing protocol used to share information between routers to facilitate the transportation of IP multicast packets among networks.
- It formed the basis of the Internet's historic multicast backbone.
- Distance vector routing for unicast is extended to support multicast routing.
- Each router maintains a routing table for all destination through exchange of distance vectors.
- DVMRP is also known as ***flood-and-prune protocol***.
- DVMRP consists of two major components:
- A conventional distance-vector routing protocol, like RIP
- A protocol for determining how to forward multicast packets, based on the
- routing table
- DVMRP router forwards a packet if the packet arrived from the link used to reach the source of the packet
- If downstream links have not pruned the tree DVMRP protocol uses the **basic packet types** as follows:

- **DVMRP Probes**
  - for DVMRP Neighbor Discovery
- **DVMRP Reports**
  - for Multicast Route Exchange
- **DVMRP Prunes**
  - for pruning multicast delivery trees
- **DVMRP Grafts**
  - for grafting multicast delivery trees
- **DVMRP Graft Ack's**
  - for acknowledging graft msgs

Source Subnet	Multicast Group	TTL	InPort	OutPorts
128.1.0.0	224.1.1.1	200	1 Pr	2p 3p
	224.2.2.2	100	1	2p 3
	224.3.3.3	250	1	2
128.2.0.0	224.1.1.1	150	2	2p 3

Multicasting is added to distance-vector routing in four stages.

- Flooding
- Reverse Path Forwarding (RPF)
- Reverse Path Broadcasting (RPB)
- Reverse Path Multicast (RPM)

Flooding

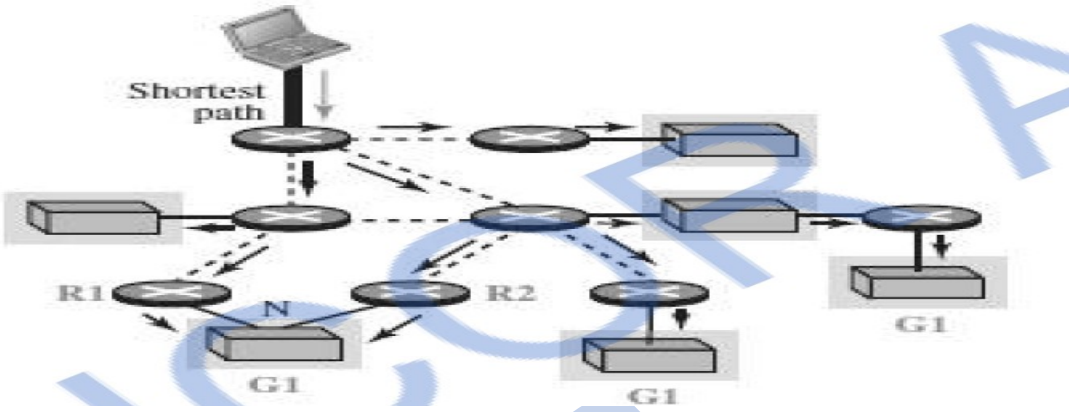
Router on receiving a multicast packet from source *S* to a Destination from Next Hop, *forwards* the packet on all out-going links. Packet is flooded and looped back to *S*. The drawbacks are:

It floods a network, even if it has *no members* for that group.

Packets are forwarded by each router connected to a LAN, i.e., *duplicate flooding*

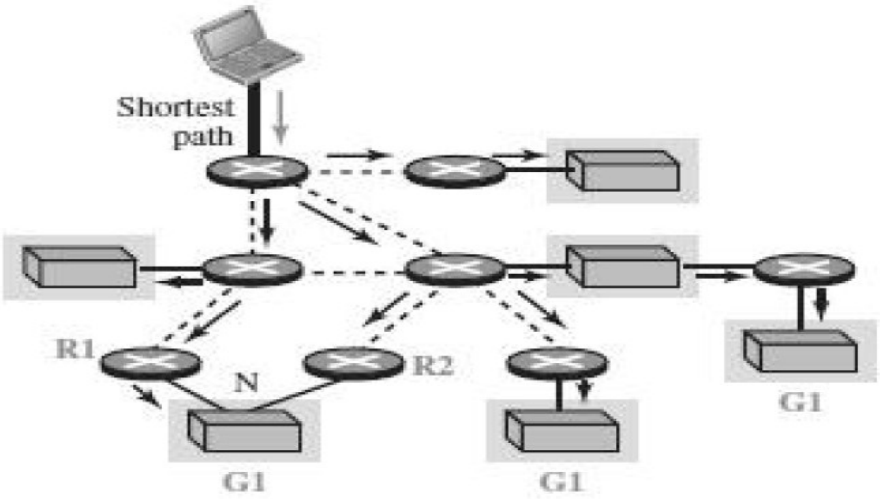
Reverse Path Forwarding (RPF)

- RPF eliminates the looping problem in the flooding process.
- Only one copy is forwarded and the other copies are discarded.
- RPF forces the router to forward a multicast packet from one specific interface: the one which has come through the shortest path from the source to the router. Packet is flooded but not looped back to *S.*



Reverse-Path Broadcasting (RPB)

- RPB does not multicast the packet, it broadcasts it. RPB creates a shortest path broadcast tree from the source to each destination. It guarantees that each destination receives one and only one copy of the packet.
- We need to prevent each network from receiving more than one copy of the packet.
- If a network is connected to more than one router, it may receive a copy of the packet from each router.
- One router identified as parent called designated Router (DR). Only parent router *forwards* multicast packets from source *S to the attached network.*
- When a router that is not the parent of the attached network receives a multicast packet, it simply drops the packet.

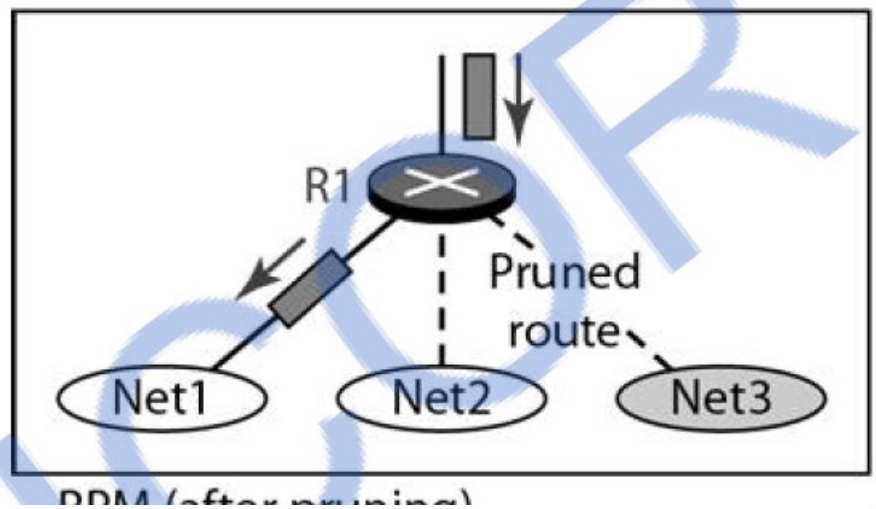


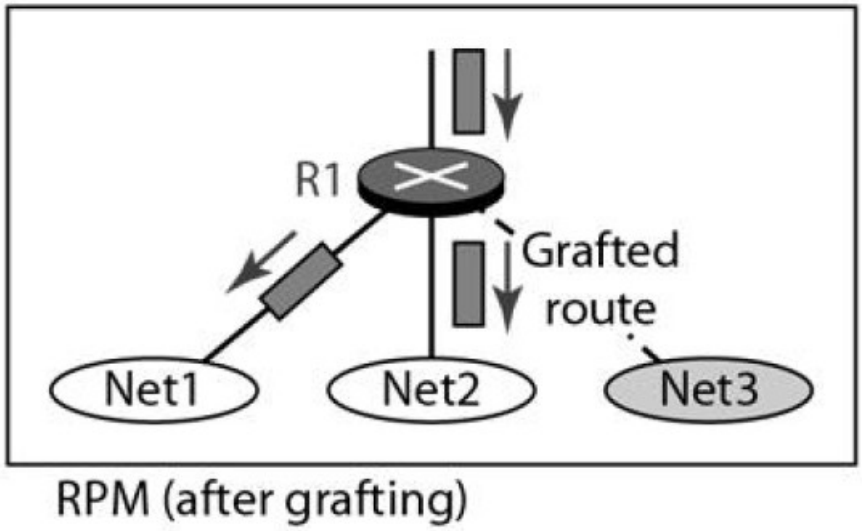
Using RPB, N receives only one copy.

Reverse-Path Multicasting (RPM)

- To increase efficiency, the multicast packet must reach only those networks that have active members for that particular group. RPM adds pruning and grafting to RPB to create a multicast shortest path tree that supports dynamic membership changes.

- ***Pruning:*** Sent from routers receiving multicast traffic for which they have no active group members. "Prunes" the tree created by DVMRP. Stops needless data from being sent





RPM (after grafting)

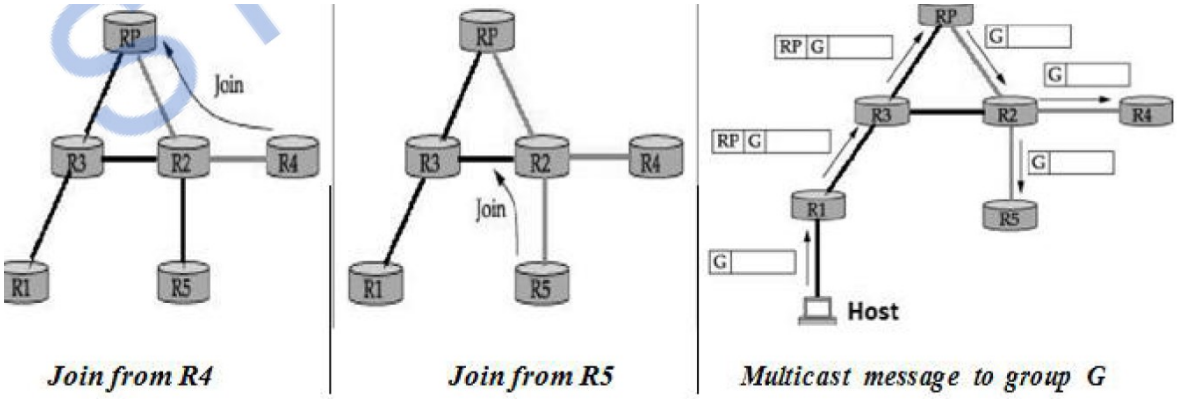
Protocol Independent Multicast (PIM)

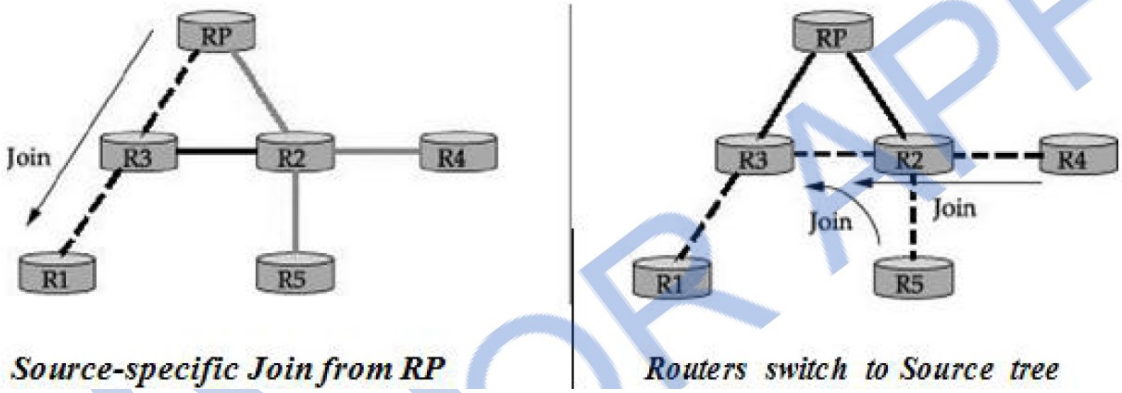
PIM divides multicast routing problem into *sparse* and *dense* mode. PIM sparse mode (PIM-SM) is widely used. PIM does not rely on any type of unicast routing protocol, hence protocol independent. Routers explicitly join and leave multicast group using ***Join and Prune messages***. One of the router is designated as *rendezvous point* (RP) for each group in a domain to receive PIM messages. Multicast forwarding *tree* is built as a result of routers sending Join messages to RP. Two types of trees to be constructed:

- ***Shared tree*** - used by all senders
- ***Source-specific* tree** - used only by a specific sending host.

Example

- Router *R4* sends Join message for group *G* to rendezvous router RP. Join message is received by router *R2*. It makes an entry (*, *G*) in its table and forwards the message to *RP*. When *R5* sends Join message for group *G*, *R2* does not forwards the Join. It *adds* an outgoing interface to the forwarding table created for that group. As routers send Join message for a group, branches are *added* to the tree, i.e., shared.





Analysis of PIM

- Protocol independent because, tree is based on Join messages via *shortest* path.
- Shared trees are more *scalable* than source-specific trees.
- Source-specific trees enable *efficient* routing than shared trees.