Unit 3: Dynamic Routing

Basic Routing:

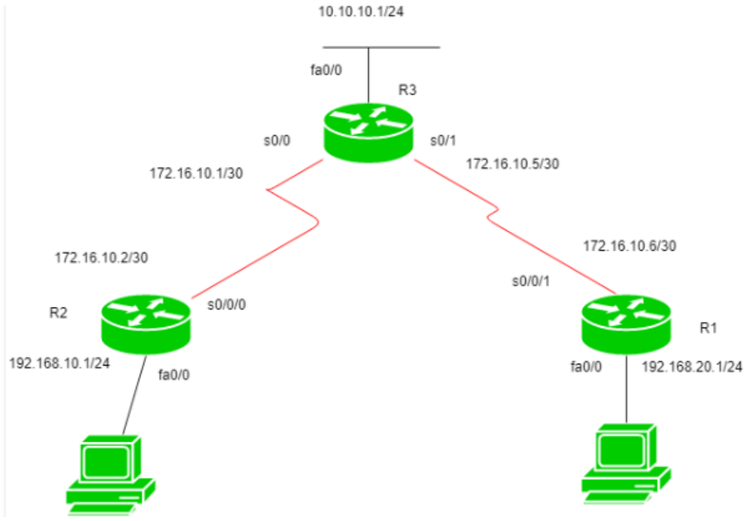
Routing is the process of selecting a path for traffic in a network, or between or across multiple networks. It refers to establishing the routes that data packets take on their way to a particular destination. In general, routing involves the network topology, or the setup of hardware, that can effectively relay data. Standard protocols help to identify the best routes for data and to ensure quality transmission. Individual pieces of hardware such as routers are referred to as "nodes" in the network. Different algorithms and protocols can be used to figure out how to best route data packets, and which nodes should be used. There are 3 types of routing:
Static routing – Static routing is a process in which we have to manually add routes in routing table.

Advantages –

- No routing overhead for router CPU which means a cheaper router can be used to do routing.
- It adds security because only administrator can allow routing to particular networks only.
- No bandwidth usage between routers.

Disadvantage –

- For a large network, it is a hectic task for administrator to manually add each route for the network in the routing table on each router.
- The administrator should have good knowledge of the topology. If a new administrator comes, then he has to manually add each route so he should have very good knowledge of the routes of the topology.



Default Routing –This is the method where the router is configured to send all packets towards a single router (next hop). It doesn't matter to which network the packet belongs, it is forwarded out to router which is configured for default routing. It is generally used with stub routers. A stub router is a router which has only one route to reach all other networks.

Dynamic Routing –Dynamic routing makes automatic adjustment of the routes according to the current state of the route in the routing table. Dynamic routing uses protocols to discover network destinations and the routes to reach it. RIP and OSPF are the best examples of dynamic routing protocol. Automatic adjustment will be made to reach the network destination if one route goes down.

A dynamic protocol have following features:

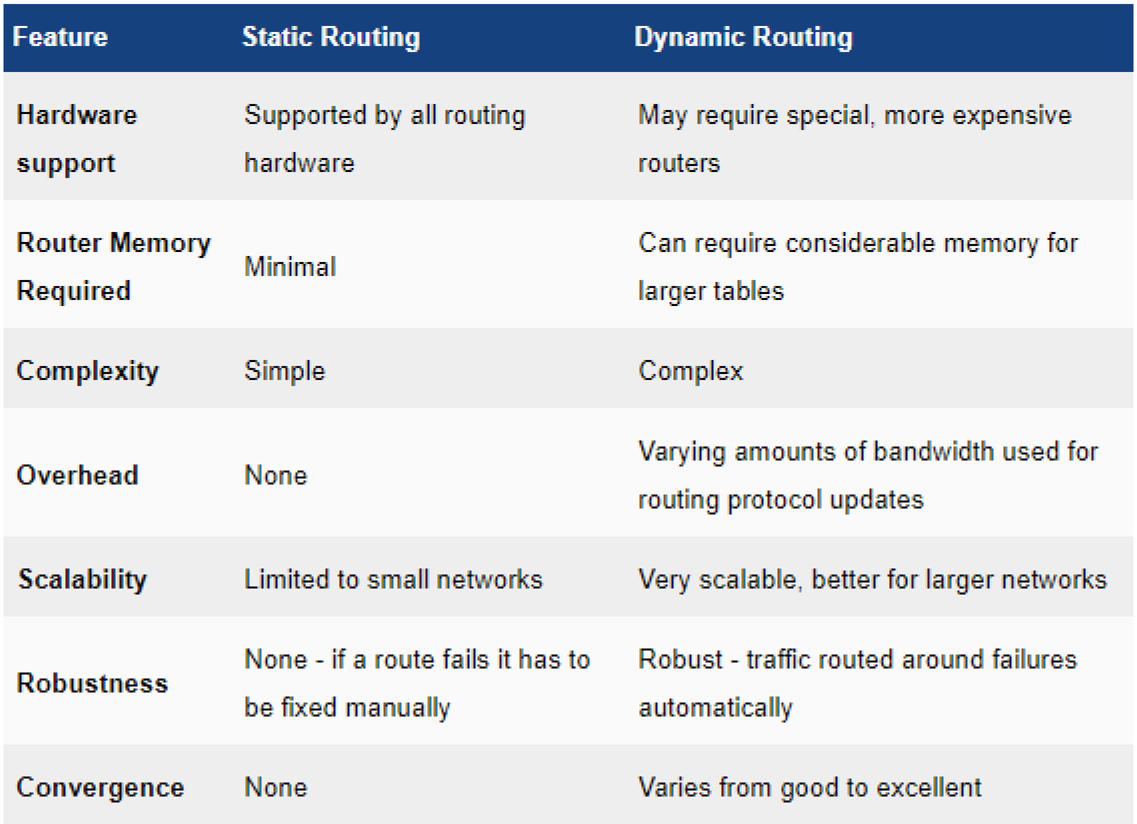
- The routers should have the same dynamic protocol running in order to exchange routes.
- When a router finds a change in the topology then router advertises it to all other routers.

Advantages –

- Easy to configure.
- More effective at selecting the best route to a destination remote network and also for discovering remote network.

Disadvantage –

- Consumes more bandwidth for communicating with other neighbors.
- Less secure than static routing.



Feature	Static Routing	Dynamic Routing
Hardware support	Supported by all routing hardware	May require special, more expensive routers
Router Memory Required	Minimal	Can require considerable memory for larger tables
Complexity	Simple	Complex
Overhead	None	Varying amounts of bandwidth used for routing protocol updates
Scalability	Limited to small networks	Very scalable, better for larger networks
Robustness	None - if a route fails it has to be fixed manually	Robust - traffic routed around failures automatically
Convergence	None	Varies from good to excellent

Dynamic routing is a networking technique that provides optimal data routing. Unlike static routing, dynamic routing enables routers to select paths according to real-time logical network layout changes. In dynamic routing, the routing protocol operating on the router is responsible for the creation, maintenance and updating of the dynamic routing table. In static routing, all these jobs are manually done by the system administrator. The cost of routing is a critical factor for all organizations. The least-expensive routing technology is provided by dynamic routing, which automates table changes and provides the best paths for data transmission.

Typically, dynamic routing protocol operations can be explained as follows:

- The router delivers and receives the routing messages on the router interfaces.
- The routing messages and information are shared with other routers, which use exactly the same routing protocol.
- Routers swap the routing information to discover data about remote networks.
- Whenever a router finds a change in topology, the routing protocol advertises this topology change to other routers.

Dynamic routing is easy to configure on large networks and is more intuitive at selecting the best route, detecting route changes and discovering remote networks. However, because routers share updates, they consume more bandwidth than in static routing; the routers' CPUs and RAM may also face additional loads as a result of routing protocols. Also, dynamic routing is less secure than static routing. Dynamic routing uses multiple algorithms and protocols. The most popular are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF).

Levels of Abstraction: Partitioning AS and areas

The Internet is huge, so it is necessary to divide the routing domain into sub-domains. There are several layers of abstractions. The Internet is partitioned into Autonomous systems (AS), an independent administrative domain. Routing between autonomous systems is called inter-domain routing (external routing). Routing within an AS is called Intra-domain routing (internal routing).

Generally, levels of abstraction in routing are classified into two types: Global routing & Decentralized routing.

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. 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.

In a decentralized routing algorithm, the calculation of the least-cost path is carried out in an iterative, distributed manner. No node has complete information about the costs of all network links. Instead, each node begins with only the knowledge of the costs of its own directly attached links. Then, through an iterative process of calculation and exchange of information with its neighboring nodes (that is, nodes that are at the other end of links to which it itself is attached), a node gradually calculates the least-cost path to a destination or set of destinations. The decentralized routing algorithm is also called a distance-vector (DV) algorithm, because each node maintains a vector of estimates of the costs (distances) to all other nodes in the network.

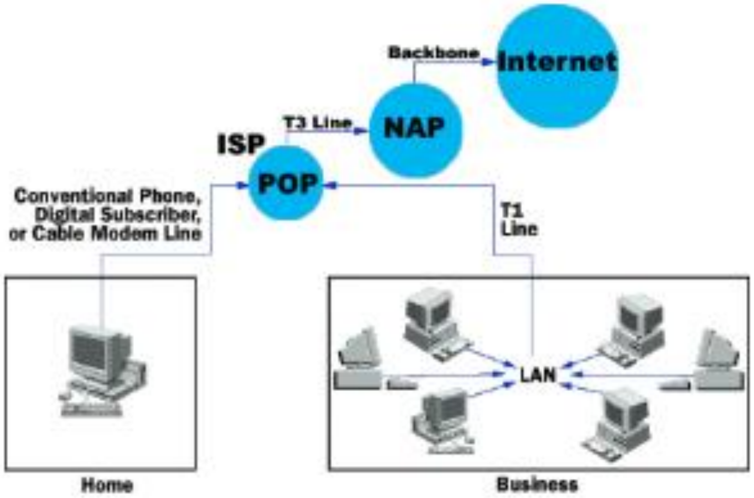


Fig: Levels of abstraction in Internet

Every computer that is connected to the Internet is part of a network, even the one in your home. When you connect to your ISP, you become part of their network. The ISP may then connect to a larger network and become part of their network. The Internet is simply a network of networks. Most large communications companies have their own dedicated backbones connecting various regions. In each region, the company has a Point of Presence (POP). The POP is a place for local users to access the company's network, often through a local phone number or dedicated line. The amazing thing here is that there is no overall controlling network. Instead, there are several high-level networks connecting to each other through Network Access Points or NAPs.

Autonomous Systems-RFC1930

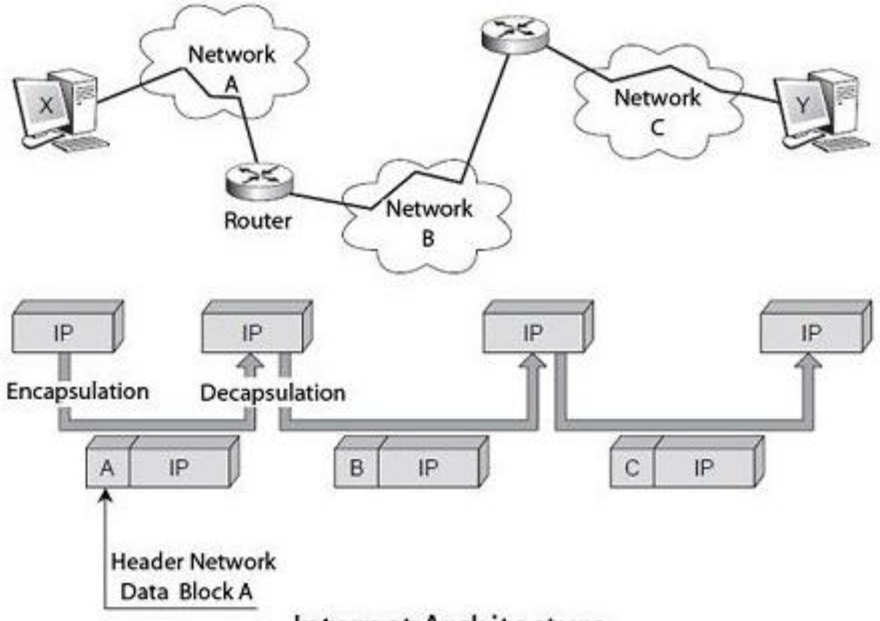
On the Internet, an autonomous system (AS) is the unit of router policy, either a single network or a group of networks that is controlled by a common network administrator (or group of administrators) on behalf of a single administrative entity (such as a university, a business enterprise, or a business division). An autonomous system is also sometimes referred to as a routing domain. An autonomous system is assigned a globally unique number, sometimes called an Autonomous System Number (ASN).

RFC1930 is a standard that discusses when it is appropriate to register and utilize an Autonomous System (AS), and lists criteria for such. It aims the network operators and service providers who need to understand under what circumstances they should make use of an AS.

Simple Internet Architecture:

The internet is a worldwide, publically accessible network of interconnected computer networks that transmit data by using standard Internet Protocol (IP). It is a constantly changing collection of thousands of individual networks intercommunicating with a common protocol.

The Internet architecture is based on a simple idea: all networks want to be part of carrying a single packet type, a specific format the IP protocol. In addition, this IP packet must carry an address defined with sufficient generality in order to identify each computer and terminals scattered throughout the world. This architecture is illustrated in Figure.



Internet Architecture

The user who wishes to make on this internetwork must store its data in IP packets that are delivered to the first network to cross. This first network encapsulates the IP packet in its own packet structure, the package A, which circulates in this form until an exit door, where it is decapsulated so as to retrieve the IP packet. The IP address is examined to locate, using routing algorithm, the next network to cross, and so on until arriving at the destination terminal.

Reachability and Metrics:

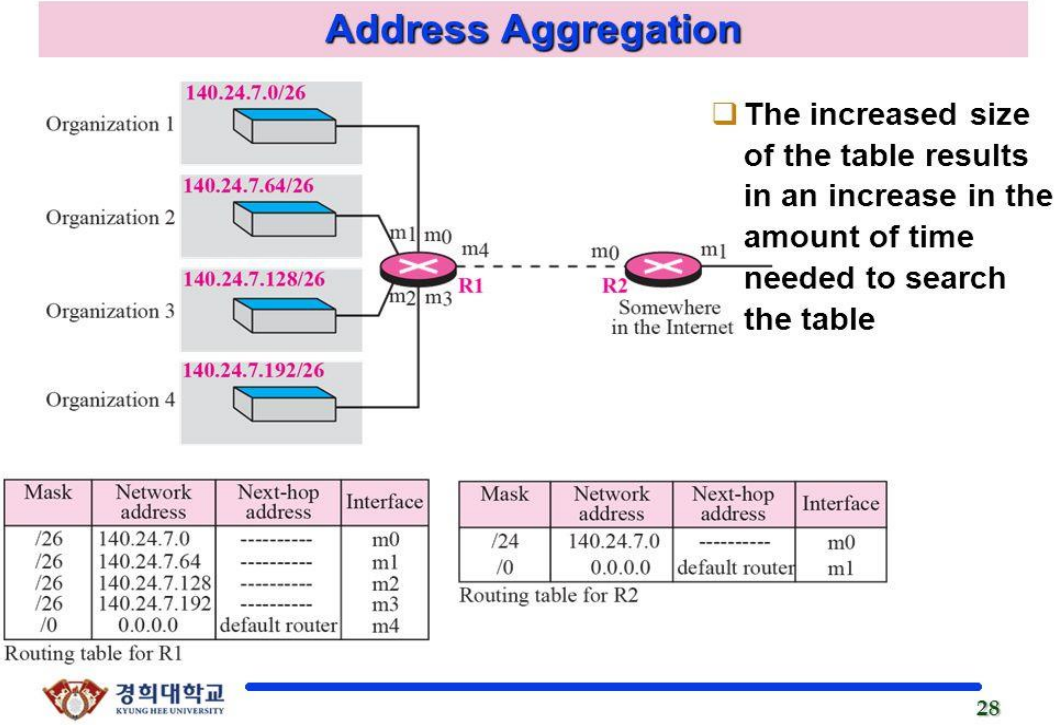
"Reachable" is just what it says; "Can this device see and communicate with the target device?” It can be defined as the network state of the user’s device. PING is a basic, primary tool for reachability.

Router metrics are metrics used by a router to make routing decisions. A routing metric is a unit calculated by a routing algorithm for selecting or rejecting a routing path for transferring data/traffic. It is calculated by routing algorithms when determining the optimal route for sending network traffic. Metrics are assigned to each different route available in the routing table and are calculated using many different techniques and methods based on the routing algorithms in use. Some of the parameters used for calculating a routing metric are as follows:

- Hop count
- Path reliability
- Path speed
  Bandwidth (rate of water flow and width of the pipe, “Bandwidth is the capacity and speed is the transfer rate”)
- Load
- Latency
- Maximum transmission unit

IP Aggregation:

When we use classless addressing, it is likely that the number of routing table entries will increase. This is so because the intent of classless addressing is to divide up the whole address space into manageable blocks. The increased size of the table results in an increase in the amount of time needed to search the table. To resolve this problem, the idea of address aggregation was designed. IP Aggregation serves as a critical tool to reduce the size of routing table and helps reducing the processing power and memory usage of the router.



Address Aggregation

- The increased size of the table results in an increase in the amount of time needed to search the table

Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/26	140.24.7.192	----------	m3
/0	0.0.0.0	default router	m4

Routing table for R1

Mask	Network address	Next-hop address	Interface
/24	140.24.7.0	----------	m0
/0	0.0.0.0	default router	m1

Routing table for R2

Router R1 is connected to networks of four organizations that each use 64 addresses. Router R2 is somewhere far from R1. Router R1 has a longer routing table because each packet must be correctly routed to the appropriate organization. Router R2, on the other hand, can have a very small routing table. For R2, any packet with destination 140.24.7.0 to 140.24.7.255 is sent out from interface m0 regardless of the organization number. This is called address aggregation into one larger block. Router R2 would have a longer routing table if each organization had addresses that could not be aggregated into one block.

Example of IP aggregation:

```
 IP              SUBNET           NEXT HOP
> 129.10.112.0    255.255.255.0    R1
> 129.10.80.0     255.255.255.0    R1
> 129.10.0.0      255.255.0.0      R2
> 129.10.63.0     255.255.255.0    R4
> 129.10.63.0     255.255.255.0    R4
> 129.10.64.0     255.255.192.0    R3
> 129.10.65.0     255.255.255.0    R4
> 129.10.66.0     255.255.255.0    R4
```

What is the proper method to aggregate these entries into the minimum number of entries?

First, separate the entries by next hop. You have to summarize them separately:

```
IP              SUBNET            NEXT HOP
129.10.112.0    255.255.255.0     R1
129.10.80.0     255.255.255.0     R1

129.10.0.0      255.255.0.0       R2

129.10.64.0     255.255.192.0     R3

129.10.63.0     255.255.255.0     R4
129.10.63.0     255.255.255.0     R4
129.10.65.0     255.255.255.0     R4
129.10.66.0     255.255.255.0     R4
```

Then for each next hop, convert all the network addresses to binary. Here is just the first one:

```
10000001.00001010.01110000.00000000 = 129.10.112.0
10000001.00001010.01010000.00000000 = 129.10.80.0
```

Now find all the identical digits, starting from the left. In this case, the digits are all the same up to the 18th position. So your new mask is /18. Now, using either address and the /18 mask, find the network address by ANDing the address and the mask:

```
10000001.00001010.01110000.00000000 = 129.10.112.0
11111111.11111111.11000000.00000000 = /18 (255.255.192.0)
-----------------------------------
10000001.00001010.01000000.00000000 = 129.10.64.0 /18
```

So the best summarization of the first two routes is 129.10.64.0/18.

Redistribution of routing information:

In a router, route redistribution allows a network that uses one routing protocol to route traffic dynamically based on information learned from another routing protocol. While running a single routing protocol throughout our entire IP internetwork is desirable, multi-protocol routing is common for a number of reasons, such as company mergers, multiple departments managed by multiple network administrators, and multi-vendor environments. Running different routing protocols is often part of a network design. In any case, having a multiple protocol environment makes redistribution a necessity. Differences in routing protocol characteristics, such as metrics, administrative distance, classful and classless capabilities can effect redistribution. Consideration must be given to these differences for redistribution to succeed. If we misconfigure Route Redistribution, this will lead to sub-optimal routing and even severe instabilities such as route oscillations and persistent routing loops. We have to convert the metric from one routing protocol to another, and this doesn't happen by default, we have to do it manually. The router should be instructed what metrics to use, and this differs from one routing protocol to another.

Load Balancing:

Load balancing is a technique used to distribute workloads uniformly across servers or other compute resources to optimize network efficiency, reliability and capacity. Modern high-traffic websites must serve hundreds of thousands of concurrent requests from users or clients and return the correct text, images, video, or application data, all in a fast and reliable manner. To cost-effectively scale to meet these high volumes, modern computing best practice generally requires adding more servers. A load balancer acts as the “traffic cop” sitting in front of the servers and routing client requests across all servers capable of fulfilling those requests in a manner that maximizes speed and capacity utilization and ensures that no one server is overworked, which could degrade performance. If a single server goes down, the load balancer redirects traffic to the remaining online servers. When a new server is added to the server group, the load balancer automatically starts to send requests to it. In this manner, a load balancer performs the following functions:

- Distributes client requests or network load efficiently across multiple servers
- Ensures high availability and reliability by sending requests only to servers that are online
- Provides the flexibility to add or subtract servers as demand arises

Popular Routing Algorithms:

A routing algorithm is a set of step-by-step operations used to direct Internet traffic efficiently. When a packet of data leaves its source, there are many different paths it can take to its destination. The routing algorithm is used to determine mathematically the best path to take.

The routing algorithms can be classified as follows:

- Adaptive Routing Algorithm: These algorithms change their routing decisions to reflect changes in the topology and in traffic as well. These get their routing information from adjacent routers or from all routers. The optimization parameters are the distance, number of hops and estimated transit time. This can be further classified as follows:
    i) Centralized: In this type some central node in the network gets entire information about the network topology, about the traffic and about other nodes. This then transmits this information to the respective routers. The advantage of this is that only one node is required to keep the information. The disadvantage is that if the central node goes down the entire network is down, i.e. single point of failure.
    ii) Isolated: In this method the node decides the routing without seeking information from other nodes. The sending node does not know about the status of a particular link. The disadvantage is that the packet may be send through a congested route resulting in a delay. Some examples of this type of algorithm for routing are: Hot Potato, Backward Learning
    iii) Distributed: In this, the node receives information from its neighboring nodes and then takes the decision about which way to send the packet. The disadvantage is that if in between the interval (it receives information and sends the packet) something changes, then the packet may be delayed.
- Non-Adaptive Routing Algorithm: These algorithms do not base their routing decisions on measurements and estimates of the current traffic and topology. Instead the route to be taken in going from one node to the other is computed in advance, off-line, and downloaded to the routers when the network is booted. This is also known as static routing.
    i) Flooding: Flooding adapts the technique in which every incoming packet is sent on every outgoing line except the one on which it arrived. One problem with this method is that packets may go in a loop. As a result of this a node may receive several copies of a particular packet which is undesirable.

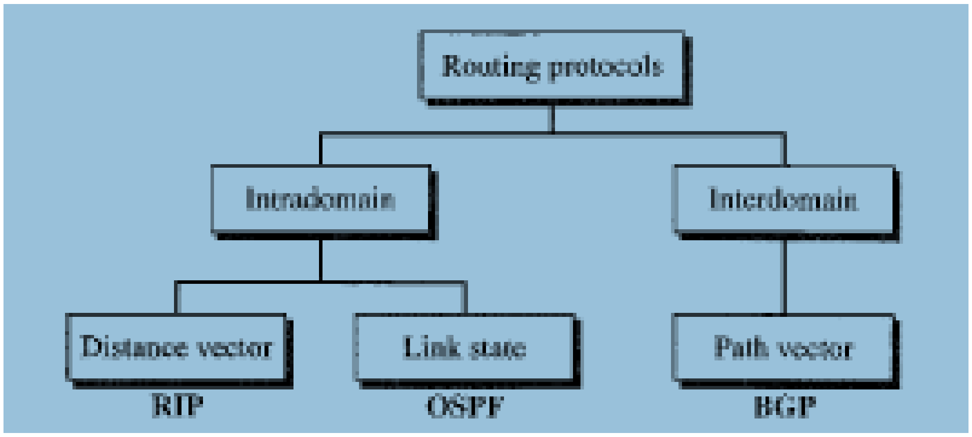
ii) Random Walk: In this method a packet is sent by the node to one of its neighbors randomly. This algorithm is highly robust. When the network is highly interconnected, this algorithm has the property of making excellent use of alternative routes. It is usually implemented by sending the packet onto the least queued link.

Delta Routing: Delta routing is a hybrid of the centralized and isolated routing algorithms. Here each node computes the cost of each line (i.e. some functions of the delay, queue length, utilization, bandwidth etc.) and periodically sends a packet to the central node giving it these values which then computes the k best paths from node i to node j.

Multipath Routing: In the above algorithms it has been assumed that there is a single best path between any pair of nodes and that all traffic between them should use it. In many networks however there are several paths between pairs of nodes that are almost equally good. Sometimes in order to improve the performance multiple paths between single pair of nodes are used. This technique is called multipath routing.

Hierarchical Routing: In this method of routing the nodes are divided into regions based on hierarchy. A particular node can communicate with nodes at the same hierarchical level or the nodes at a lower level and directly under it. Here, the path from any source to a destination is fixed and is exactly one if the hierarchy is a tree.

Dynamic routing algorithms are basically categorized as follows:

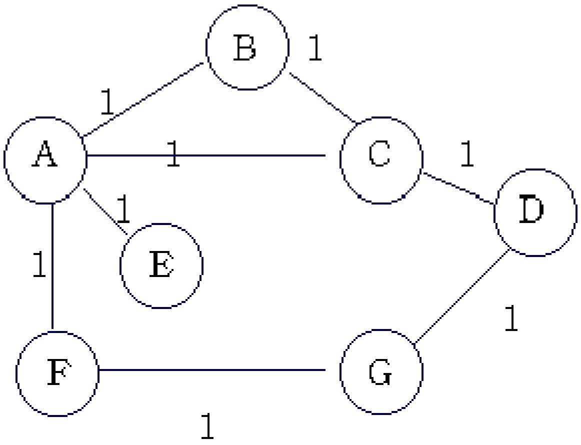


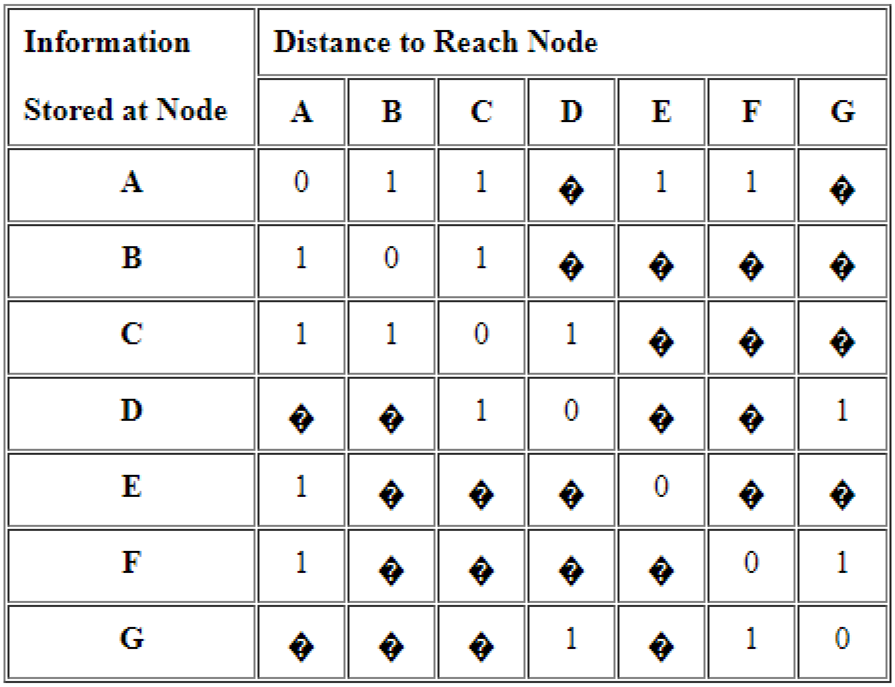
Distance Vector Routing Algorithm:

A distance-vector routing protocol in data networks determines the best route for data packets based on distance. Distance-vector routing protocols measure the distance by the number of routers a packet has to pass, one router counts as one hop. The vector describes the route of the message over a given set of network nodes. To determine the best route across a network routers on which a distance-vector protocol is implemented exchange information with one another, usually routing tables plus hop counts for destination networks and possibly other traffic information. The basic idea here is that each node receives some information from one or more of its directly attached neighbors, performs a calculation, and then distributes the results of its calculation back to its neighbors.

Distance vector routing algorithm is also called Bellman Ford algorithm. Each router maintains a Distance Vector table containing the distance between itself and all possible destination nodes. Distances, based on a chosen metric, are computed using information from the neighbors' distance vectors. The starting assumption for distance-vector routing is that each node knows the cost of the link to each of its directly connected neighbors. A link that is down is assigned an infinite cost.

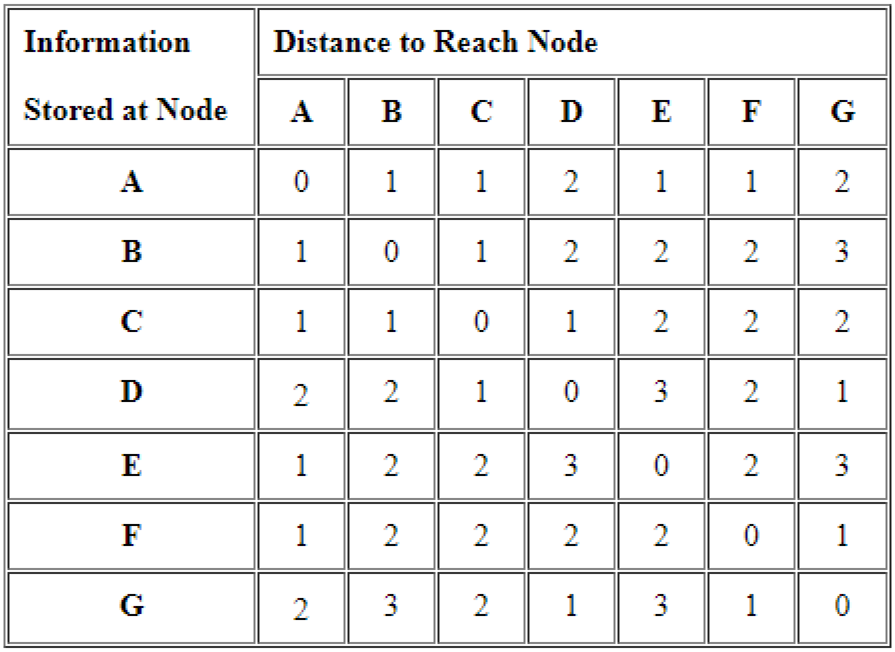
E.g.:





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

Table 1. Initial distances stored at each node(global view).

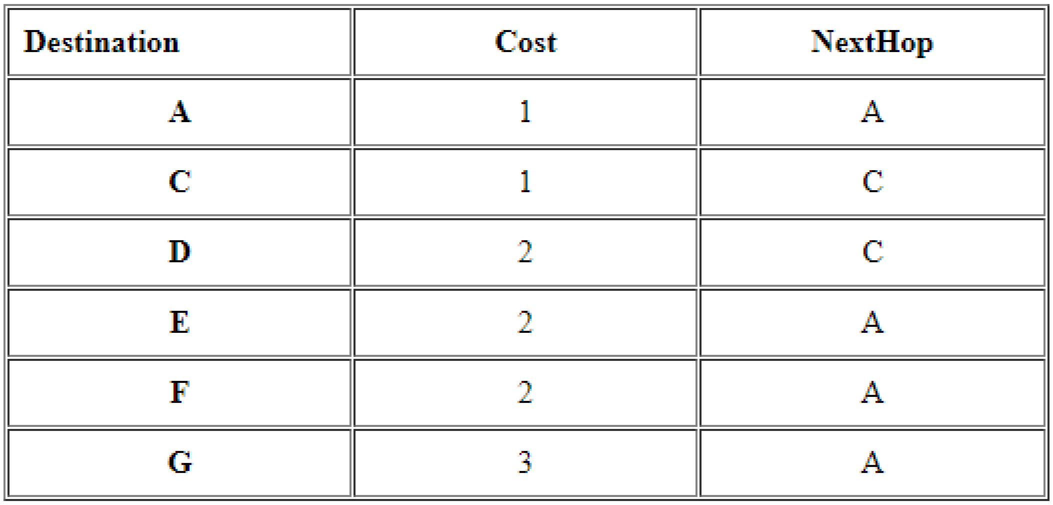


Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

Table 2. final distances stored at each node (global view).

In practice, each node's forwarding table consists of a set of triples of the form: (Destination, Cost, Next Hop).

For example, Table below shows the complete routing table maintained at node B for the network in figure above.



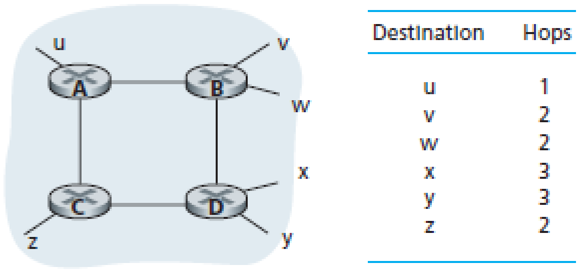
Destination	Cost	NextHop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	3	A

Table 3. Routing table maintained at node B.

RIP (Routing Information Protocol):

Routing Information Protocol (RIP) is a dynamic protocol used to find the best route or path from end-to-end (source to destination) over a network by using a routing metric/hop count algorithm. This algorithm is used to determine the shortest path from the source to destination, which allows the data to be delivered at high speed in the shortest time. RIP plays an important role providing the shortest and best path for data to take from node to node. The hop is the step towards the next existing device, which could be a router, computer or other device. Once the length of the hop is determined, the information is stored in a routing table for future use. RIP is being used in both local and wide area networks and is generally considered to be easily configured and implemented.

Figure below illustrates an AS with six leaf subnets. The table in the figure indicates the number of hops from the source A to each of the leaf subnets.



Destination	Hops
u	1
v	2
w	2
x	3
y	3
z	2

Figure 4.34 ♦ Number of hops from source router A to various subnets

The maximum cost of a path is limited to 15, thus limiting the use of RIP to autonomous systems that are fewer than 15 hops in diameter. Recall that in DV protocols, neighboring routers exchange distance vectors with each other. The distance vector for any one router is the current estimate of the shortest path distances from that router to the subnets in the AS. In RIP, routing updates are exchanged between neighbors approximately every 30 seconds using a RIP response message. The response message sent by a router or host contains a list of up to 25 destination subnets within the AS, as well as the sender's distance to each of those subnets. Response messages are also known as RIP advertisements.

In brief the RIP protocol works as follows:

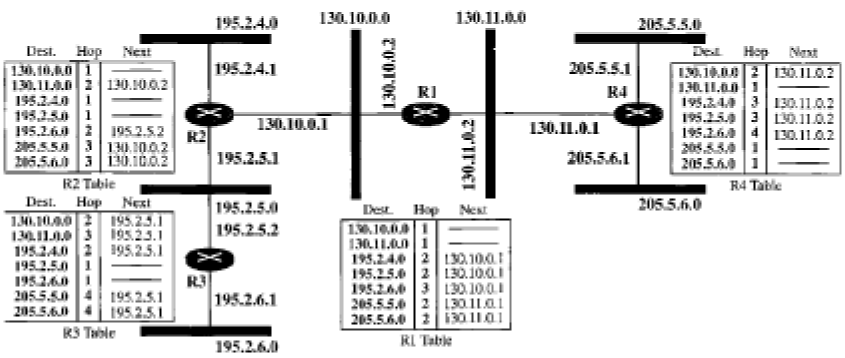
- Each router initializes its routing table with a list of locally connected networks.
- Periodically, each router advertises the entire contents of its routing table over all of its RIP-enabled interfaces.
  - ➢ Whenever a RIP router receives such an advertisement, it puts all of the appropriate routes into its routing table and begins using it to forward packets. This process ensures that every network connected to every router eventually becomes known to all routers.
  - ➢ If a router does not continue to receive advertisements for a remote route, it eventually times out that route and stops forwarding packets over it.
- Every route has a property called a metric, which indicates the "distance" to the route's destination.
  - ➢ Every time a router receives a route advertisement, it increments the metric.
  - ➢ Routers prefer shorter routes to longer routes when deciding which of two versions of a route to program in the routing table.

- The maximum metric permitted by RIP is 16, which means that a route is unreachable. This means that the protocol cannot scale to networks where there may be more than 15 hops to a given destination.

RIP also includes some optimizations of this basic algorithm to improve stabilization of the routing database and to eliminate routing loops.

- When a router detects a change to its routing table, it sends an immediate "triggered" update. This speeds up stabilization of the routing table and elimination of routing loops.
- When a route is determined to be unreachable, RIP routers do not delete it straightaway. Instead they continue to advertise the route with a metric of 16 (unreachable). This ensures that neighbors are rapidly notified of unreachable routes, rather than having to wait for a soft state timeout.
- When router A has learnt a route from router B, it advertises the route back to B with a metric of 16 (unreachable). This ensures that B is never under the impression that A has a different way of getting to the same destination. This technique is known as "split horizon with poison reverse."
- A "Request" message allows a newly-started router to rapidly query all of its neighbors' routing tables.

Figure 22.19 *Example of a domain using RIP*



Dest.	Hop	Next
130.10.0.0	1	—
130.11.0.0	2	130.10.0.2
195.2.4.0	1	—
195.2.5.0	1	—
195.2.6.0	2	195.2.5.2
205.5.5.0	3	130.10.0.2
205.5.6.0	3	130.10.0.2

R2 Table

Dest.	Hop	Next
130.10.0.0	2	195.2.5.1
130.11.0.0	3	195.2.5.1
195.2.4.0	2	195.2.5.1
195.2.5.0	1	—
195.2.6.0	1	—
205.5.5.0	4	195.2.5.1
205.5.6.0	4	195.2.5.1

R3 Table

Dest.	Hop	Next
130.10.0.0	1	—
130.11.0.0	1	—
195.2.4.0	2	130.10.0.1
195.2.5.0	2	130.10.0.1
195.2.6.0	3	130.10.0.1
205.5.5.0	2	130.11.0.1
205.5.6.0	2	130.11.0.1

R1 Table

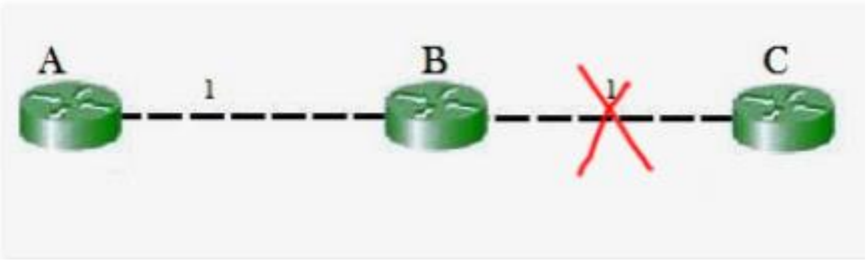
Dest.	Hop	Next
130.10.0.0	2	130.11.0.2
130.11.0.0	1	—
195.2.4.0	3	130.11.0.2
195.2.5.0	3	130.11.0.2
195.2.6.0	4	130.11.0.2
205.5.5.0	1	—
205.5.6.0	1	—

R4 Table

The figure above shows an autonomous system with seven networks and four routers. Table for each router is also shown. Looking at routing table for R1, it has seven entries to show how to reach each network in the autonomous system. Router R1 is directly connected to networks 130.10.0.0 and 130.11.0.0, which means that there are no next hop entries for these two networks. To send a packet to one of the three networks at the far left, router R1 needs to deliver the packet to R2. The next-node entry for these three networks is the interface of router R2 with IP address 130.10.0.1. To send a packet to the two networks at the far right, router R1 needs to send the packet to the interface of router R4 with IP address 130.11.0.1. The other tables can be explained similarly.

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

Counting to infinity problem:



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.

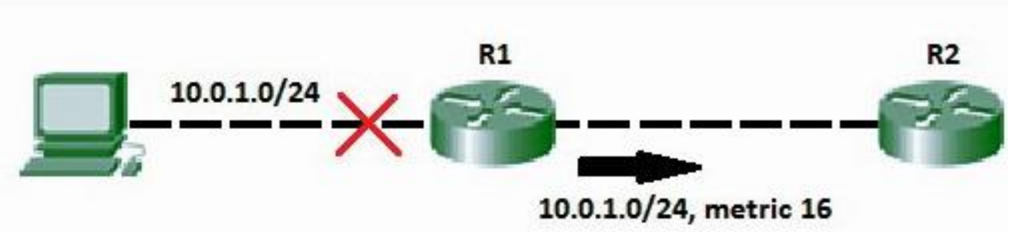
Solution for Count to infinity:

Triggered Update:

A type of Routing Information Protocol (RIP) announcement that occurs when network topology changes is called triggered update. With triggered updates, the update announcing network topology changes is sent almost immediately rather than waiting for the next periodic announcement. Triggered updates deal with count to infinity issues by forcing an update as soon as the link changes. Triggered updates improve the convergence time (the time it takes for a router to update its routing tables) of RIP internetworks, but at the cost of additional broadcast traffic while the triggered updates are propagated.

Route Poisoning:

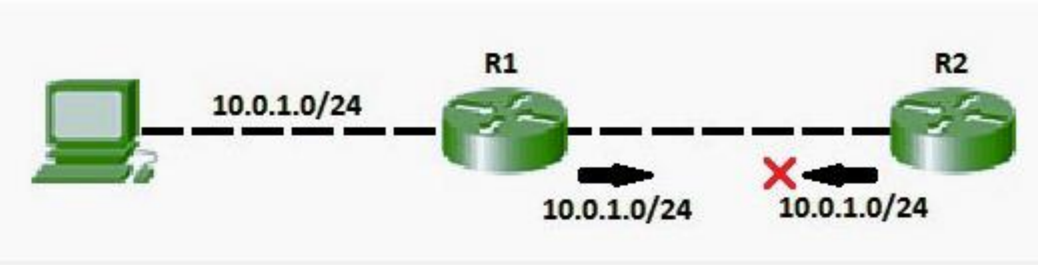
When a route fails, distance vector protocols spread the bad news about a route failure by poisoning the route. Route poisoning refers to the practice of advertising a route, but with a special metric value called Infinity. Routers consider routes advertised with an infinite metric to have failed. Each 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.



Split horizon:

If the link between B and C goes down, and B had received a route from A , B could end up using that route via A. A would send the packet right back to B, creating a loop. But according to Split horizon Rule, Node A does not advertise its route for C (namely A to B to C) back to B. On the surface, this seems redundant since B will never route via node A because the route costs more than the direct route from B to C.

Consider the following network topology showing Split horizon:



In addition to these, we can also use split horizon with route poisoning where above both technique will be used combinely to achieve efficiency and less increase the size of routing announcements.

Split horizon with Poison reverse technique is used by Routing Information Protocol (RIP) to reduce routing loops. Additionally, **Holddown** timers can be used to avoid the formation of loops. Holddown timer immediately starts when the router is informed that attached link is down. Till this time, router ignores all updates of down route unless it receives an update from the router of that downed link. During the timer, if the down link is reachable again, routing table can be updated.

Disadvantage:

- The primary drawback of this algorithm is its vulnerability to the ‘Count-to-Infinity’ problem. Many partial solutions have been proposed but none works under all circumstances.
- Another drawback of this scheme is that it does not take into account link bandwidth.
- Yet another problem with this algorithm is that it takes longer time for convergence as network size grows.
- Increased network traffic: RIP checks with its neighboring routers every 30 seconds, which increases network traffic.
- Maximum hop count: RIP has a maximum hop count of 15, which means that on large networks, other remote routers may not be able to be reached.
- Closest may not be shortest: Choosing the closest path by hop count does not necessarily mean that the fastest route was selected. RIP does not consider other factors when calculating best path.
- RIP only updates neighbors so the updates for non-neighboring routers are not first-hand information

Link State Protocols:

The basic concept of link-state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible destination in the network. Each collection of best paths will then form each node's routing table. While distance vector routers use a distributed algorithm to compute their routing tables, link-state routing uses link-state routers to exchange messages that allow each router to learn the entire network topology. Based on this learned topology, each router is then able to compute its routing table by using a shortest path computation.

Features of link state routing protocols:

- Link state packet – A small packet that contains routing information.
- Link state database – A collection information gathered from link state packet.
- Shortest path first algorithm (Dijkstra algorithm) – A calculation performed on the database results into shortest path
- Routing table – A list of known paths and interfaces.

Calculation of shortest path –

To find shortest path, each node need to run the famous **Dijkstra algorithm**. Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph. This famous algorithm uses the following steps:

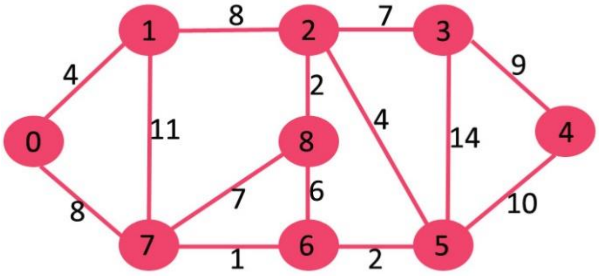
Step-1: The node is taken and chosen as a root node of the tree, this creates the tree with a single node, and now set the total cost of each node to some value based on the information in Link State Database

Step-2: Now the node selects one node, among all the nodes not in the tree like structure, which is nearest to the root, and adds this to the tree. The shape of the tree gets changed.

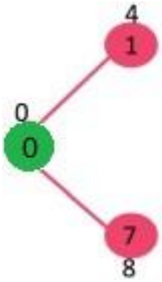
Step-3: After this node is added to the tree, the cost of all the nodes not in the tree needs to be updated because the paths may have been changed.

Step-4: The node repeats the Step 2 and Step 3 until all the nodes are added in the tree.

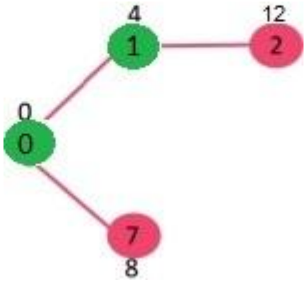
Let us understand with the following example:



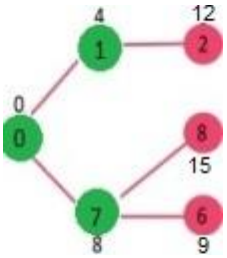
The set sptSet is initially empty and distances assigned to vertices are {0, INF, INF, INF, INF, INF, INF, INF} where INF indicates infinite. Now pick the vertex with minimum distance value. The vertex 0 is picked, include it in sptSet. So sptSet becomes {0}. After including 0 to sptSet, update distance values of its adjacent vertices. Adjacent vertices of 0 are 1 and 7. The distance values of 1 and 7 are updated as 4 and 8. Following subgraph shows vertices and their distance values, only the vertices with finite distance values are shown. The vertices included in SPT are shown in green colour.



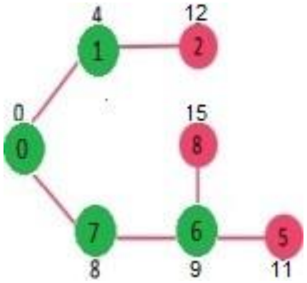
Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). The vertex 1 is picked and added to sptSet. So sptSet now becomes {0, 1}. Update the distance values of adjacent vertices of 1. The distance value of vertex 2 becomes 12.



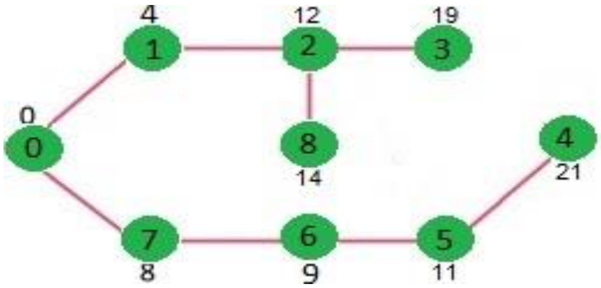
Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). Vertex 7 is picked. So sptSet now becomes {0, 1, 7}. Update the distance values of adjacent vertices of 7. The distance value of vertex 6 and 8 becomes finite (15 and 9 respectively).



Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). Vertex 6 is picked. So sptSet now becomes {0, 1, 7, 6}. Update the distance values of adjacent vertices of 6. The distance value of vertex 5 and 8 are updated.

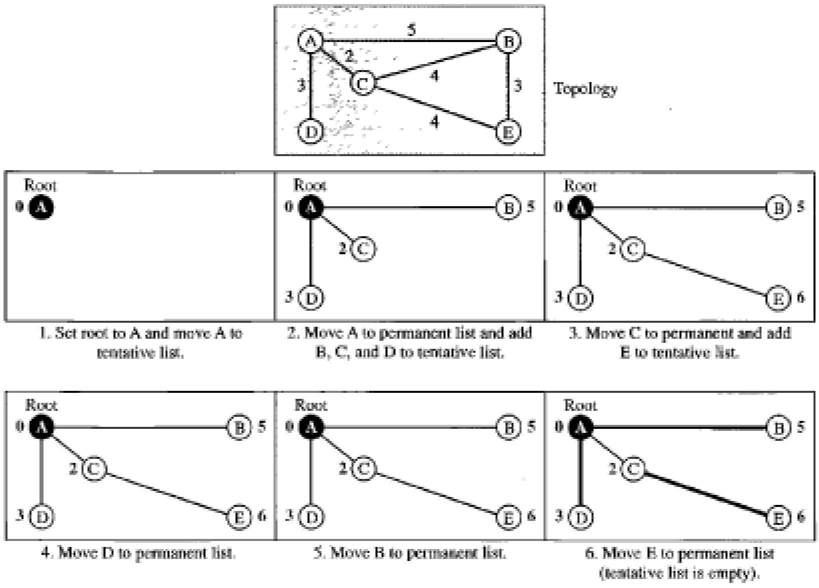


We repeat the above steps until sptSet doesn't include all vertices of given graph. Finally, we get the following Shortest Path Tree (SPT).



Another example for Dijkstra's algorithm is as follows:

Figure 22.23 *Example of formation of shortest path tree*



1. Set root to A and move A to tentative list.

2. Move A to permanent list and add B, C, and D to tentative list.

3. Move C to permanent and add E to tentative list.

4. Move D to permanent list.

5. Move B to permanent list.

6. Move E to permanent list (tentative list is empty).

Overview of OSPF (Open Path Shortest First):

Open Shortest Path First (OSPF) is a link state routing protocol (LSRP) that uses the Shortest Path First (SPF) network communication algorithm (Dijkstra's algorithm) to calculate the shortest connection path between known devices.

The OSPF routing protocol has largely replaced the older Routing Information Protocol (RIP) in corporate networks. Using OSPF, a router that learns of a change to a routing table (when it is reconfigured by network staff, for example) or detects a change in the network immediately multicasts the information to all other OSPF hosts in the network so they will all have the same routing table information. Unlike RIP, which requires routers to send the entire routing table to neighbors every 30 seconds, OSPF sends only the part that has changed and only when a change has taken place. When routes change -- sometimes due to equipment failure -- the time it takes OSPF routers to find a new path between endpoints with no loops (which is called "open") and that minimizes the length of the path is called the convergence time. Rather than simply counting the number of router hops between hosts on a network, as RIP does, OSPF bases its path choices on "link states" that take into account additional network information, including IT-assigned cost metrics that give some paths higher assigned costs. For example, a satellite link may be assigned higher cost than a wireless WAN link, which in turn may be assigned higher cost than a metro Ethernet link.

For example, a person in city A wants to travel to city M and is given two options:

Travel via cities B and C. The route would be ABCM. And the distance (or bandwidth cost in the networking case) for A-B is 10 miles, B-C is 5 miles and C-M is 10 miles.

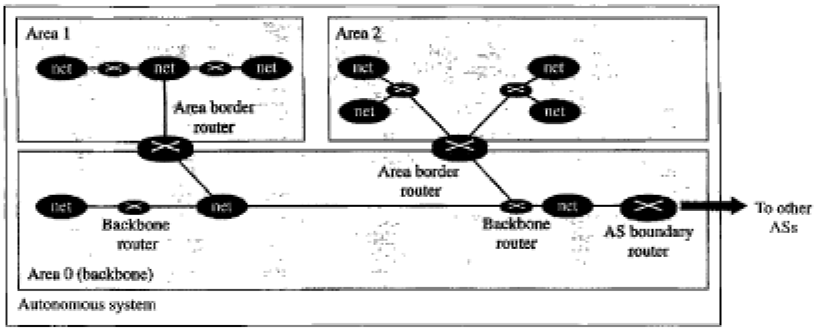
Travel via city F. The route would be AFM. And the distance for A-F is 20 miles and F-M is 10 miles.

The shortest route is always the one with least amount of distance covered in total. Thus, the ABCM route is the better option (10+5+10=25), even though the person has to travel to two cities as the associated total cost to travel to the destination is less than the second option with a single city (20+10=30). OSPF performs a similar algorithm by first calculating the shortest path between the source and destination based on link bandwidth cost and then allows the network to send and receive IP packets via the shortest route.

OSPF Network Topology:

Two routers communicating OSPF to each other exchange information about the routes they know about and the cost for them to get there. When many OSPF routers are part of the same network, information about all of the routes in a network are learned by all of the OSPF routers within that network—technically called an area. Each OSPF router passes along information about the routes and costs they've heard about to all of their adjacent OSPF routers, called neighbors.

Figure 22.24 *Areas in an autonomous system*



An area is a collection of networks, hosts, and routers all contained within an autonomous system. At the border of an area, special routers called area border routers summarize the information about the area and send it to other areas. Among the areas inside an autonomous system is a special area called the backbone; all the areas inside an autonomous system must be connected to the backbone. In other words, the backbone serves as a primary area and the other areas as secondary areas. The routers inside the backbone are called the backbone routers.

OSPF works best in a hierarchical routing environment. When designing an OSPF network, the first and most important task is to determine which routers and links are to be included in the backbone (area 0) and which are to be included in each area. The following are three important characteristics to OSPF to ensure that your OSPF network has a hierarchical routing structure:

- The hierarchical routing structure must exist or be created to effectively use OSPF. The benefits of having a single area include simplicity, ease of troubleshooting, and so on.
- A contiguous backbone area must be present, and all areas must have a connection to the backbone.
- Explicit topology (shortest path) has precedence over any IP addressing schemes that might have been applied; that is, your physical topology takes precedence over a summarized route.

When designing the topology for an OSPF network, consider the following important items:

- Number of routers in an area
- Number of areas connected to an ABR (Area border router)
- Number of neighbors for a router
- Number of areas supported by a router
- Selection of the designated router (DR)
- Size and development of the OSPF LSDB (link state database)

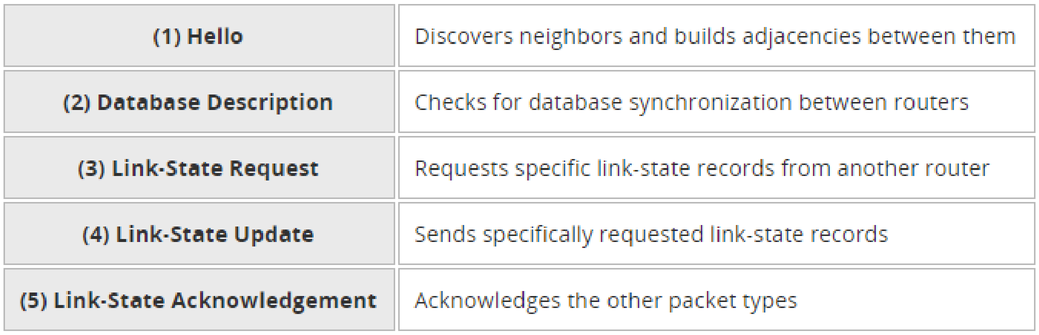
OSPF Protocols (hello, exchange, flooding):

Routers periodically send hello packets on all interfaces to establish and maintain neighbor relationships. Hello packets are multicast on physical networks that have a multicast or broadcast capability, which enables dynamic discovery of neighboring routers.. Hello packets are sent out every 10 seconds which helps to detect failed neighbors. RouterDeadInterval (default 40 seconds) is specified for detecting such neighbors. Also, hello message ensures that link between neighbors is bidirectional. Neighboring routers agree on intervals where hello interval is set so that a link is not accidentally brought down.

OSPF uses hello packets and two timers to check if a neighbor is still alive or not:

Hello interval: this defines how often we send the hello packet.

Dead interval: this defines how long we should wait for hello packets before we declare the neighbor dead.



(1) Hello	Discovers neighbors and builds adjacencies between them
(2) Database Description	Checks for database synchronization between routers
(3) Link-State Request	Requests specific link-state records from another router
(4) Link-State Update	Sends specifically requested link-state records
(5) Link-State Acknowledgement	Acknowledges the other packet types

The Hello message contains a list of information needed to form an OSPF neighbor relation between two neighboring routers, the following a list of information contained the Hello messages:

- OSPF Router ID. The router's ID which is configured or automatically selected by OSPF (analyzed below)
- Hello Interval Timer. Frequency upon which Hello packets are sent.
- Dead Interval Timer. Defines how long we should wait for hello packets before we declare the neighbor dead.
- Subnet Mask
- Router Priority. Used to help determine the Designated Router (DR). Higher priority takes precedence. A configured Priority of 0 means the router will not become a DR or BDR.
- List of reachable OSPF neighbors in the network.
- Area ID
- DR & BDR's IP addresses (if exists)
- Authentication Password (if configured)

The following conditions must be met for two routers to become neighbors:

- They must have the same IP network/subnet
- The Hello and Dead Interval timers must be identical
- Router interfaces connecting two routers must have the same Area ID
- Type of area must be identical (normal or stub area)
- Authentication password (if used) must be identical

Neighboring routers first exchange hellos. A database description packet establishes the sequence number for the each packet. The other router sends LSA (Link State Advertisement) headers and the sequence number is incremented for every pair of database description packets. After examining LSA headers, explicit request are sent for complete LSAs. The exchange of packets is for the communication between routers to discover the link states.

Flooding starts when a router wants to update self-originated LSAs. Link State Update packets are used for updating the link states. Neighbor installs more recent LSAs into its database to adapt to the recent changes in the network. The link state update packets flood out on all interfaces except the one on which it arrived. Reliability-retransmissions are done until acks are received by the router that sent the link state update packet.

Distribution of Link State Advertisement:

The LSAs (Link-State Advertisements) are used by routers running OSPF to exchange topology information. An LSA contains routing and topology information that describe a part of an OSPF network. Routers exchange LSAs and learn the complete topology of the network until all routers have the exact same topology database.

When two neighbors decide to exchange routes, they send each other a list of all LSAa in their respective topology database. Each router then checks its topology database and sends a Link State Request (LSR) requesting all LSAs not found in its topology table. The other router responds with the Link State Update (LSU) that contains all LSAs requested by the neighbor.

There are several different LSA types in OSPF:

Type 1 LSA – also known as router link advertisement (RLA), a Type 1 LSA is sent by every router to other routers in its area. It contains the router ID (RID), interfaces, IP information, and current interface state. Note that Type 1 LSAs are flooded only across their own area.

Type 2 LSA – also known as network link advertisement (NLA), a Type 2 LSA is generated by designated routers (DRs) to send out information about the state of other routers that are part of the same network. Type 2 LSAs are flooded across their own area only.

Type 3 LSA – also known as summary link advertisement (SLA), a Type 3 LSA is generated by area border routers (ABRs) and sent toward the area external to the one where they were generated. It contains the IP information and RID of the ABR that is advertising an LSA Type 3.

Type 4 LSA – informs the rest of the OSPF domain how to get to the ASBR. The link-state ID includes the router ID of the described ASBR.

Type 5 LSA – also known as AS external link advertisements, a Type 5 LSA is sent by autonomous system boundary routers (ASBRs) to advertise routes that are external to the OSPF autonomous system and are flooded everywhere.

IS-IS:

The IS-IS (Intermediate System - Intermediate System) protocol is one of a family of IP Routing protocols, and is an Interior Gateway Protocol (IGP) for the Internet, used to distribute IP routing information throughout a single Autonomous System (AS) in an IP network.

IS-IS is a link-state routing protocol, which means that the routers exchange topology information with their nearest neighbors. The topology information is flooded throughout the AS, so that every router within the AS has a complete picture of the topology of the AS. This picture is then used to calculate end-to-end paths through the AS, normally using a variant of the Dijkstra algorithm. Therefore, in a link-state routing protocol, the next hop address to which data is forwarded is determined by choosing the best end-to-end path to the eventual destination.

Both IS-IS and Open Shortest Path First (OSPF) are link state protocols, and both use the same Dijkstra algorithm for computing the best path through the network. As a result, they are conceptually similar. Both support variable length subnet masks, can use multicast to discover neighboring routers using hello packets, and can support authentication of routing updates.

IS-IS differs from OSPF in the way that "areas" are defined and routed between. IS-IS routers are designated as being: Level 1 (intra-area); Level 2 (inter area); or Level 1–2 (both). Routing information is exchanged between Level 1 routers and other Level 1 routers of the same area, and Level 2 routers can only form relationships and exchange information with other Level 2 routers. Level 1–2 routers exchange information with both levels and are used to connect the inter area routers with the intra area routers.

The major differences between OSPF and IS-IS are:

- IS-IS runs on the data link layer, whereas OSPF runs on the network layer.
- OSPF supports virtual link, whereas IS-IS does not support.

- OSPF defines a backbone area called area 0 for inter-area advertisements, whereas IS-IS categorizes the domain into two layers.
- An OSPF router can belong to multiple areas whereas an IS-IS router can belong to only one area.
- OSPF uses Router ID, whereas IS-IS uses System ID to identify each router on the network

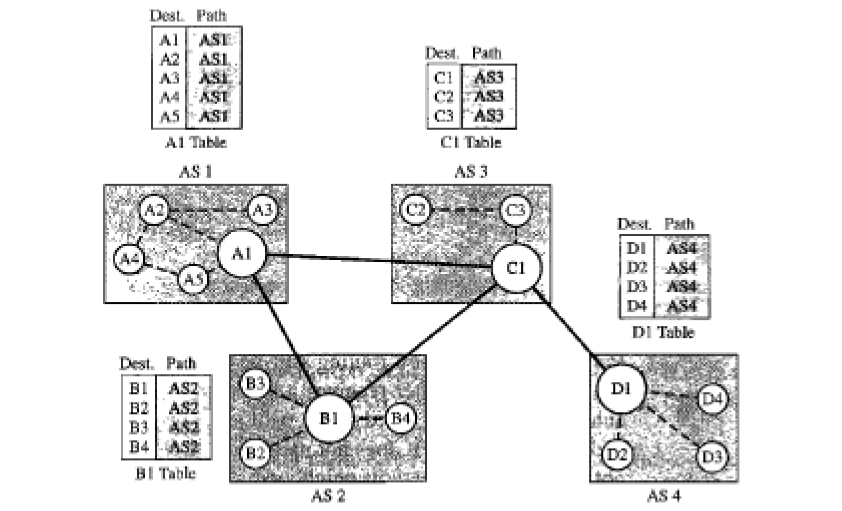
Path Vector:

Distance vector and Link State routing are both intradomain routing protocols. They can be used inside an autonomous system, but not between the autonomous systems. These two protocols are not suitable for interdomain routing mostly because of scalability. Both of these routing protocols become intractable when the domain of operation becomes large. Distance vector routing is subject to instability if there are more than a few hops in the domain of operation. Link state routing needs a huge amount of resources to calculate routing tables. It also creates heavy traffic because of flooding. There is a need for a third routing protocol which we call path vector routing.

Path Vector Routing is a routing algorithm in unicast routing protocol of network layer, and it is useful for interdomain routing. The principle of path vector routing is similar to that of distance vector routing. In path vector routing, we assume that there is one node (there can be more, but one is enough) in each autonomous system that acts on behalf of the entire autonomous system, referred to as speaker node. The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring autonomous systems. A speaker node advertises the path, not the metrics of the nodes, in its autonomous system or other autonomous systems.

Initialization: At the beginning, each speaker node can only know only the reachability of the nodes inside its autonomous system.

Figure 22.30 *Initial routing tables in path vector routing*



Dest.	Path
A1	AS1
A2	AS1
A3	AS1
A4	AS1
A5	AS1

A1 Table

Dest.	Path
C1	AS3
C2	AS3
C3	AS3

C1 Table

Dest.	Path
D1	AS4
D2	AS4
D3	AS4
D4	AS4

D1 Table

Dest.	Path
B1	AS2
B2	AS2
B3	AS2
B4	AS2

B1 Table

Sharing: Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors. In figure, node A1 shares its table with nodes B1 and C1.

Node C1 shares its table with nodes D1, B1 and A1. Node B1 shares its table with C1 and A1. Node D1 shares its table with C1.

Updating: When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table. After a while, each speaker has a table and knows how to reach each node in other autonomous systems. Figure below shows the tables for each speaker node after the system is stabilized.

Loop prevention: The instability of distance vector routing and the creation of loops can be avoided in path vector routing. When a router receives a message, it checks to see if its autonomous system is in the path list to the destination. If it is, looping is involved and the message is ignored.

Policy routing: Policy routing can be easily implemented through path vector routing. When a router receives a message, it can check the path. If one of the autonomous systems listed in the path is against its policy, it can ignore that path and that destination. It does not update its routing table with this path, and it does not send this message to its neighbors.

Optimum path: The optimum path in path vector routing is a path to a destination that is the best for the organization that runs the autonomous system. We cannot use metrics in this route because each autonomous system that is included in the path may use a different criteria for the metric. One system may use RIP which defines hop count as the metric, another may use OSPF with the minimum delay (higher link bandwidth) as the metric. The optimum path is the path that fits the organization. In previous figure, each autonomous system may have more than one path to a destination. For eg: a path from AS4 to AS1 can be AS4-AS3-AS2-AS1 or it can be AS4-AS3-AS1. For the tables, we choose the one that had the smaller number of autonomous systems, but this is not always the case. Other criteria, such as security, safety, and reliability can also be applied.

Border Gateway Protocol:

Border gateway protocol (BGP) is an interdomain routing protocol using path vector routing. BGP is protocol that manages how packets are routed across the internet through the exchange of routing and reachability information between edge routers. BGP directs packets between autonomous systems (AS) -- networks managed by a single enterprise or service provider. Traffic that is routed within a single network AS is referred to as internal BGP, or iBGP. More often, BGP is used to connect one AS to other autonomous systems, and it is then referred to as an external BGP, or eBGP.

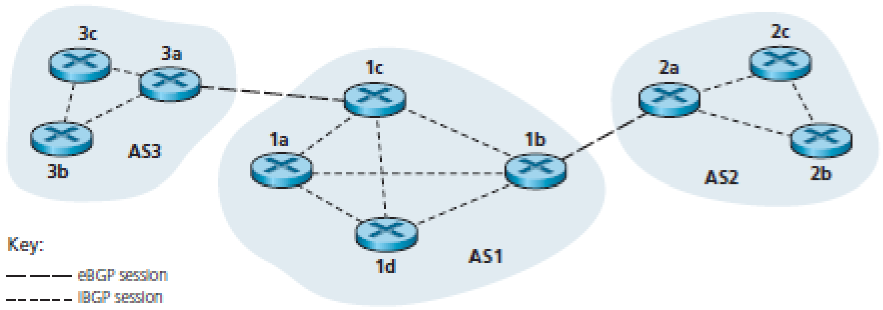


Figure 4.40 ♦ eBGP and iBGP sessions

All other routing protocols are concerned solely with finding the optimal path towards all known destinations. BGP cannot take this simplistic approach because the peering agreements between ISPs almost always result in complex routing policies. To help network operators implement these policies, BGP carries a large number of attributes with each IP prefix:

- Weight – The BGP weight attribute is Cisco-specific and is used to influence how traffic is routed for a specific BGP device. This value does not pass between internal or external BGP neighbors (peers).
- Local Preference – The local preference attribute is used to dictate how traffic prefers to leave a specific BGP ASN. This attribute is passed between neighbors within the same ASN. The highest local preference gets priority.
- Local Routes – Routes which have been sourced from the local router will be preferred over those sourced from other routers.
- Shortest AS_PATH – With BGP, the path is notated by the ASN of the external BGP networks that must be traversed to reach the destination network; e.g. 10 20 30 means that the traffic must pass through ASNs 10, 20, and 30 to reach the destination. If multiple options exist to a specific network, the one with the shortest AS path will be preferred.
- Origin – With origin, BGP is looking for the source of the initial network advertisement, for example if it was redistributed from an IGP, an EGP or through an unknown source. When analyzing this attribute, routes that have originated from an IGP are preferred to those from an EGP, and routes that have originated by an EGP will be preferred over those originated from an unknown source. I < E < ?
- Multi-Exit Discriminator (MED) – The MED is a value that can be injected into a neighboring BGP ASN. This is used when multiple paths exist between two different BGP ASNs. The MED is used to suggest to the neighboring ASN the preferred way to route traffic into their network. The lowest MED value gets priority.
- BGP Neighbor Type – There are two different types of BGP neighborship: internal and external. A BGP neighborship that exists within the same ASN between two devices is considered internal, and a BGP neighborship that exists between devices from different ASNs is considered external. External (or eBGP) routes are preferred to Internal (iBGP) routes.
- IGP metric/next hop – The next attribute uses the IGP metric to the BGP next hop address.
- Oldest External Route – If the contending BGP routes are external then the one which has existed the longest will be preferred
- Lowest Router-ID – The route with the lowest BGP router ID will be preferred
- Lowest Neighbor Address – The route coming through a neighbor with the lowest address will be preferred.

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 of 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 knows about various networks and methods to reach them.

BGP Router Model:

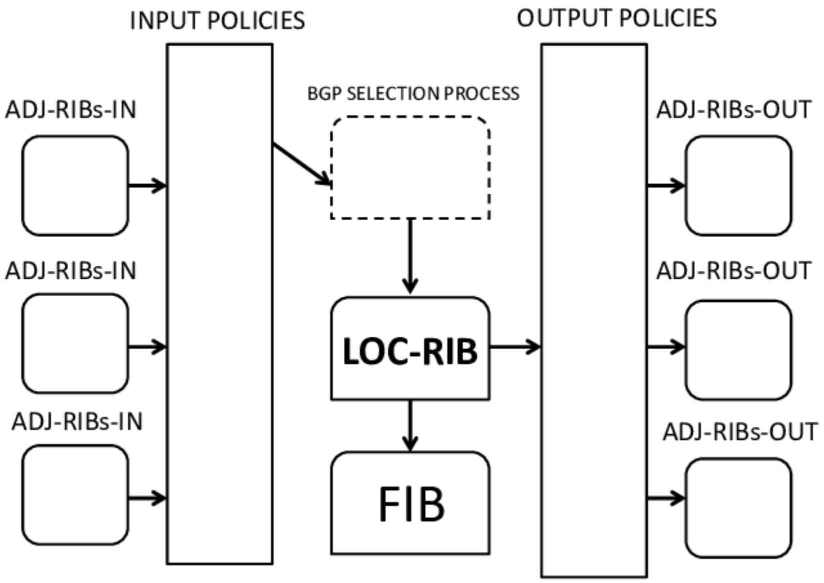
In general, a BGP router or BGP speaker is a router that runs the BGP protocol software, besides other routing protocols. The BGP router maintains the reachability information exchanged between AS in a table named Routing Information Base (RIB). The RIB is one of the most important elements of BGP protocol, organizing all information and considering not only routing, but also the policy relations while storing and sending routes.

To better organize its model in accordance to its functions, a RIB is subdivided in three parts:

- Adj-RIBs-In: This is the part of RIB responsible for storing the routing information received from peers through UPDATEs. For each neighbor of a BGP speaker there exists an associated Adj-RIBs-In instance. When the BGP selection process is triggered, the deployed input policies filter the routes at Adj-RIBs-In that must be available to be selected and installed in Loc-RIB.
- Loc RIB: The Loc RIB contains the selected best paths chosen after the selection process takes place and input policies filter the routes received by Adj-RIBs-In. The routes in RIB are then installed in Forwarding Information Base (FIB), which is a table used by routers to forward the packets to the indicated outputs. Loc RIB is also referred to as main RIB.

- Adj-RIBs-Out: For each neighbor of a BGP speaker there also exists an associated Adj-RIBs-Out instance. After the routes are installed in Loc RIB, output policies are applied to select the routes that will be advertised to each peer. The Adj-RIBs-Out contains the information of the permitted routes after the output policies are filtered. This model is an abstract concept of a BGP router. The real implementation depends on vendors and manufacturers; some implementations keep only one instance of RIB, for example, to save storage and memory.

The BGP Router model with its elements and the relation between them is illustrated in Figure.


INPUT POLICIES
OUTPUT POLICIES
ADJ-RIBs-IN
ADJ-RIBs-IN
ADJ-RIBs-IN
BGP SELECTION PROCESS
LOC-RIB
FIB
ADJ-RIBs-OUT
ADJ-RIBs-OUT
ADJ-RIBs-OUT