UNIT 3

1. IPv4 Datagram

1. Introduction

Internet Protocol (IP) is the communications protocol in the network layer. Its main operation is routing. It forwards packets originating at a host to its destination, one router at a time.

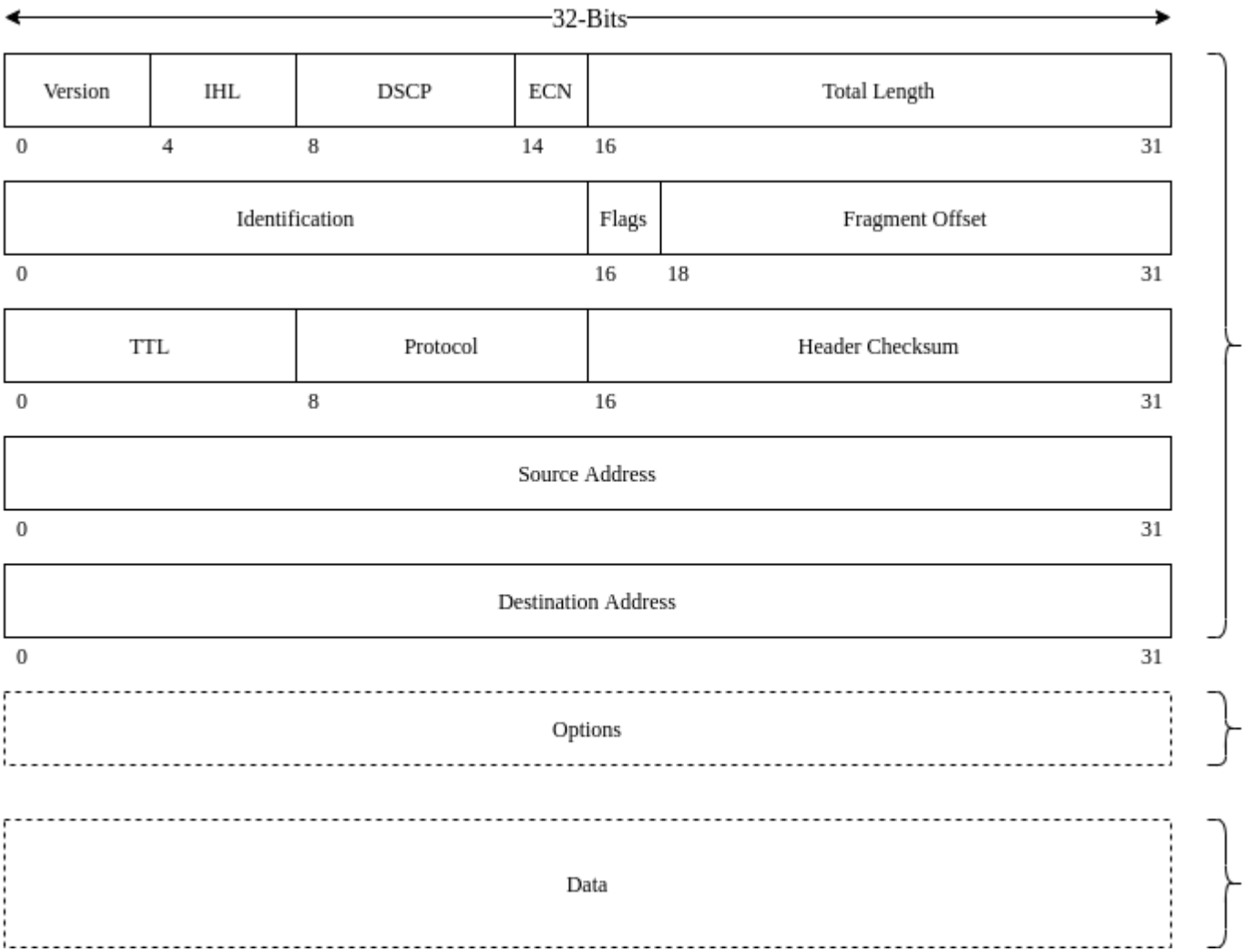
In summary, the transport layer breaks up data streams into IP packets. Then, IP routers forward these packets through the Internet. At the destination, the network layer reassembles all the packets into the original stream. Finally, the transport layer takes and processes the data.

The Internet is made up of different networks. The hardware for these networks supports varying bandwidths. As a result, sometimes, the routers have to break the IP packets into smaller parts along the way. **This is called fragmentation.**

2. IPv4 Datagram

IPv4 was developed in 1981, and it's still widely used. It uses 32-bit address space.

An IPv4 packet is called a datagram. It is made up of a header and a data part:



IPv4 header contains a 20-byte fixed mandatory part, followed by optional fields. **Hence, the minimum size of an IPv4 header is 20 bytes.**

The optional part contains options and padding. It can grow up to 40 bytes in size. **So, the maximum header size is 60 bytes.**

A datagram can carry up to 65,535 bytes of data ( ), at least 20 bytes of it being the header. Hence, the maximum amount of data it carries is 65,515 bytes.

The header part contains 14 fields. The first 13 are mandatory. The last one is the optional field called options. Let's look at the individual header fields.

- **Version** − Version no. of Internet Protocol used (e.g. IPv4).
- **IHL** − Internet Header Length; Length of entire IP header.
- **DSCP** − Differentiated Services Code Point; this is Type of Service.
- **ECN** − Explicit Congestion Notification; It carries information about the congestion seen in the route.
- **Total Length** − Length of entire IP Packet (including IP header and IP Payload).

- **Identification** − If IP packet is fragmented during the transmission, all the fragments contain same identification number. to identify original IP packet they belong to.
- **Flags** − As required by the network resources, if IP Packet is too large to handle, these 'flags' tells if they can be fragmented or not. In this 3-bit flag, the MSB is always set to '0'.
- **Fragment Offset** − This offset tells the exact position of the fragment in the original IP Packet.
- **Time to Live** − To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded.
- **Protocol** − Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol. For example protocol number of ICMP is 1, TCP is 6 and UDP is 17.
- **Header Checksum** − This field is used to keep checksum value of entire header which is then used to check if the packet is received error-free.
- **Source Address** − 32-bit address of the Sender (or source) of the packet.
- **Destination Address** − 32-bit address of the Receiver (or destination) of the packet.
- **Options** − This is optional field, which is used if the value of IHL is greater than 5. These options may contain values for options such as Security, Record Route, Time Stamp, etc.

What is IPv4 datagram fragmentation?

The Internet is made up of different networks. The hardware for these networks supports varying bandwidths. As a result, sometimes, **the routers have to break the IP packets into smaller parts along the way**. This is called fragmentation

IPv4 Datagram Fragmentation

Why IPv4 Datagram Fragmentation required?

Different Networks may have different maximum transmission unit (MTU), for example due to differences in LAN technology. When one network wants to transmit datagrams to a network with a smaller MTU, the routers on path may fragment and reassemble datagrams.

How is Fragmentation done?

When a packet is received at the router, destination address is examined and MTU is determined. If size of the packet is bigger than the MTU, and the 'Do not Fragment (DF)' bit is set to 0 in header, then the packet is fragmented into parts and sent one by one. The maximum size of each fragment is the MTU minus the IP header size (Minimum 20 bytes and Maximum 60 bytes).
Each fragment is converted to a packet and the following changes happen in the datagram header:

1. The total length field is changed to the size of the fragment.
2. The More Fragment bit (MF bit) is set for all the fragment packets except the last one.
3. The fragment offset field is set, based on the number of fragment that is being set and the MTU.
4. Header Checksum is re-calculated.

Example: For a data packet of 4000 bytes and MTU of 1500 bytes, we have actual data of 3980 bytes that is to be transmitted and 1480 bytes is the maximum data size that is permissible to be sent. So, there would be 3 fragments:
For the first fragment, data size = 1480 bytes, offset = 0 and MF flag = 1
For the second fragment, data size = 1480 bytes, offset = 185(1480/ 8) and MF flag = 1
For the third fragment, data size = 1020 bytes, offset = 370 (2960/8) and MF flag = 0

An important point to be noted here is that all fragments would be having same identification number, thus indicating that all the fragments belong to the same parent data packet.

Delays –
Processing delay: Time taken by the routers to process the data packet header.
Queuing delay: Time taken by the data packet in routing queues.

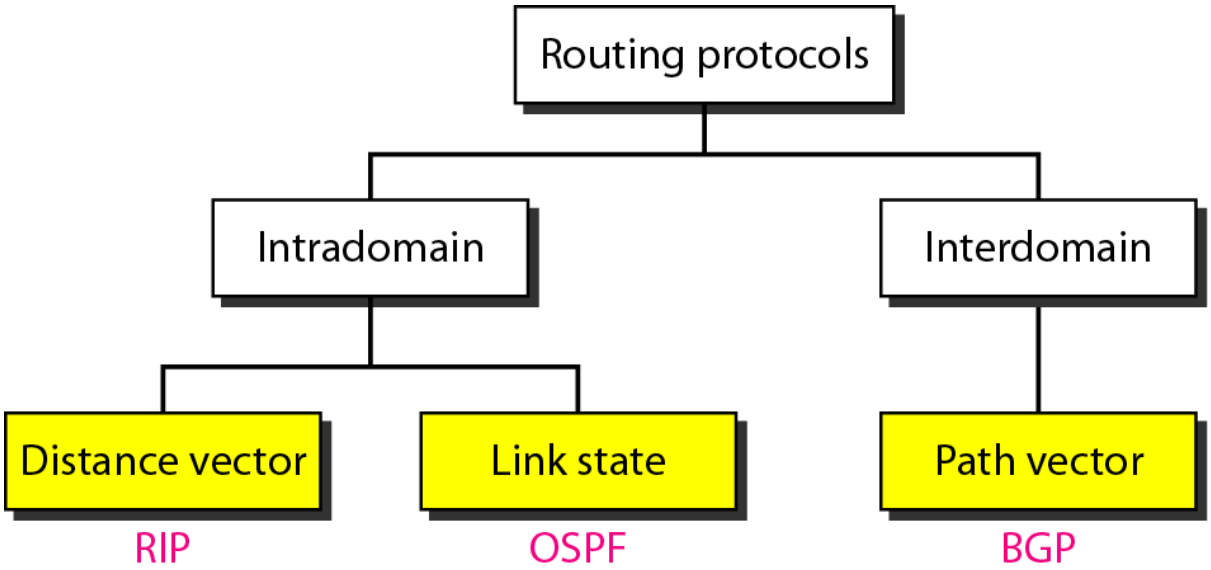
Transmission delay: Time taken to load a data packet onto the transmission channel $D_t = N/R$,
N: Number of bits to be transmitted
R: Rate or transmission speed of the channel
Propagation delay – Time taken by the data packet to reach from source to destination

$D_p = D/S$,
D: Distance between the source and the destination
S: is the speed of propagation

Unicast Routing and Least-Cost Routing :

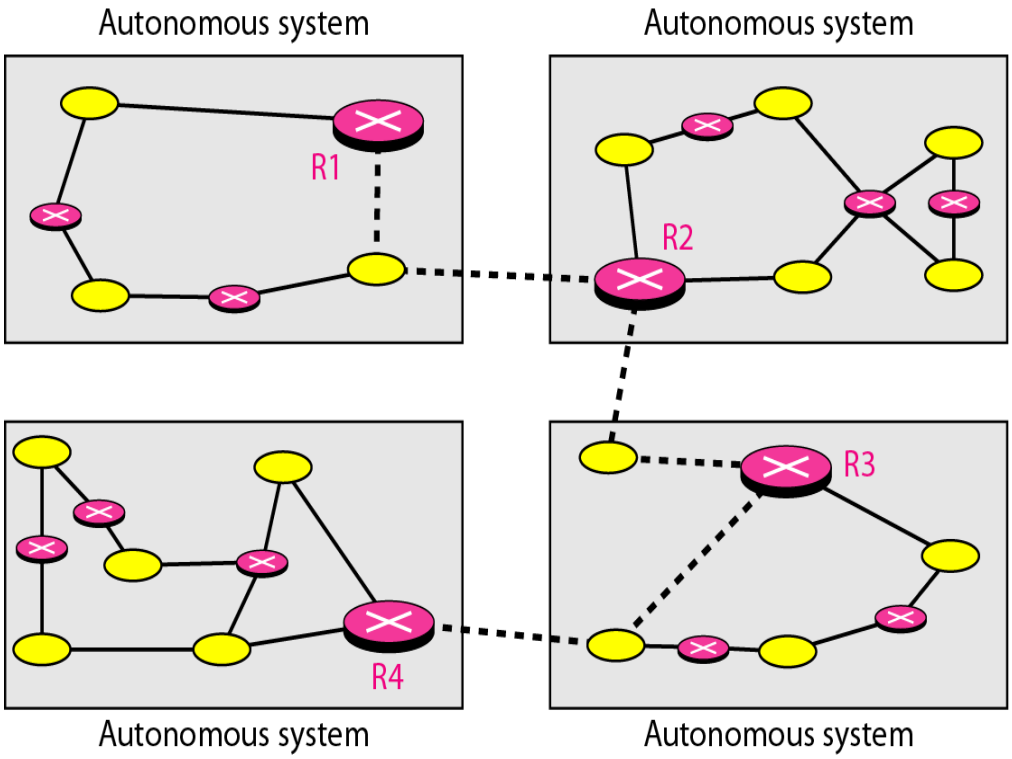
Unicast routing is the process of forwarding data packets from a source to a single destination over a network. In unicast routing, each packet is individually addressed and delivered to the specific destination specified in the packet header.
The general idea behind unicast routing is to find the most efficient path for the data packets to travel from the source to the destination. This is typically done using a routing algorithm that takes into account various factors such as the distance between the source and destination, the speed and capacity of the available network links, and the current load on the network.

One common approach to unicast routing is least-cost routing, where the goal is to find the path that has the lowest cost in terms of network resources (e.g. bandwidth, transmission time, etc.). To determine the least-cost path, the routing algorithm may use metrics such as the hop count (the number of intermediate nodes the packet must pass through), the delay time (the time it takes for the packet to reach the destination), or the cost of using each link in the network.
Unicast routing is an important aspect of networking, as it enables data to be delivered efficiently and reliably from one specific location to another. It is used in a wide range of applications, including the Internet, local area networks (LANs), and wide area networks (WANs).



- Interdomain Routing allows hosts inside one domain to exchange data with hosts in another domain. This is done using Path Vector Routing.

- Intradomain Routing is the routing that is concerned with data transfer only within a specific domain. This is done using either Distance Vector Routing or Link State Routing



Distance Vector Routing

In distance-vector routing (DVR), each router is required to inform the topology changes to its neighboring routers periodically. Historically it is known as the old ARPNET routing algorithm or Bellman-Ford algorithm.

How the DVR Protocol Works

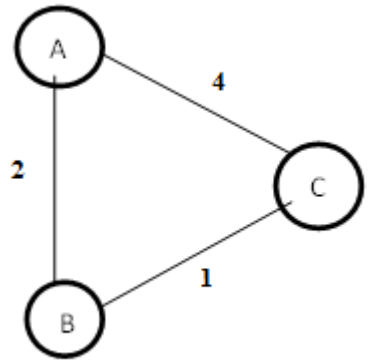
- In DVR, each router maintains a routing table. It contains only one entry for each router. It contains two parts − a preferred outgoing line to use for that destination and an estimate of time (delay). Tables are updated by exchanging the information with the neighbor's nodes.

- Each router knows the delay in reaching its neighbors (Ex − send echo request).
- Routers periodically exchange routing tables with each of their neighbors.
- It compares the delay in its local table with the delay in the neighbor's table and the cost of reaching that neighbor.
- If the path via the neighbor has a lower cost, then the router updates its local table to forward packets to the neighbor.

Example − Distance Vector Router Protocol

In the network shown below, there are three routers, A, B, and C, with the following weights − AB =2, BC =3 and CA =5.

Step 1 − In this DVR network, each router shares its routing table with every neighbor. For example, A will share its routing table with neighbors B and C and neighbors B and C will share their routing table with A.



Form A	A	B	C
A	0	2	3
B			
C			

Form B	A	B	C
A			
B	2	0	1
C			

Form C	A	B	C
A			
B			
C	3	1	0

Step 2 − If the path via a neighbor has a lower cost, then the router updates its local table to forward packets to the neighbor. In this table, the router updates the lower cost for A and C by updating the new weight from 4 to 3 in router A and from 4 to 3 in router C.

Form A	A	B	C
A	0	2	3
B			
C			

Form B	A	B	C
A			
B	2	0	1
C			

Form C	A	B	C

A			
B			
C	3	1	0

Step 3 − The final updated routing table with lower cost distance vector routing protocol for all routers A, B, and C is given below −

Router A

Form A	A	B	C
A	0	2	3
B	2	0	1
C	3	1	0

Router B

Form B	A	B	C
A	0	2	3
B	2	0	1
C	3	1	0

Router C

Form C	A	B	C
A	0	2	3
B	2	0	1
C	3	1	0

Link-State Routing

Link state routing is a technique in which each router shares the knowledge of its neighbourhood with every other router in the internetwork.

The three keys to understand the Link State Routing algorithm:

- **Knowledge about the neighbourhood:** Instead of sending its routing table, a router sends the information about its neighbourhood only. A router broadcast its identities and cost of the directly attached links to other routers.
- **Flooding:** Each router sends the information to every other router on the internetwork except its neighbours. This process is known as Flooding. Every router that receives the packet sends the copies to all its neighbours. Finally, each and every router receives a copy of the same information.
- **Information sharing:** A router sends the information to every other router only when the change occurs in the information.

Link State Routing has two phases:

Reliable Flooding

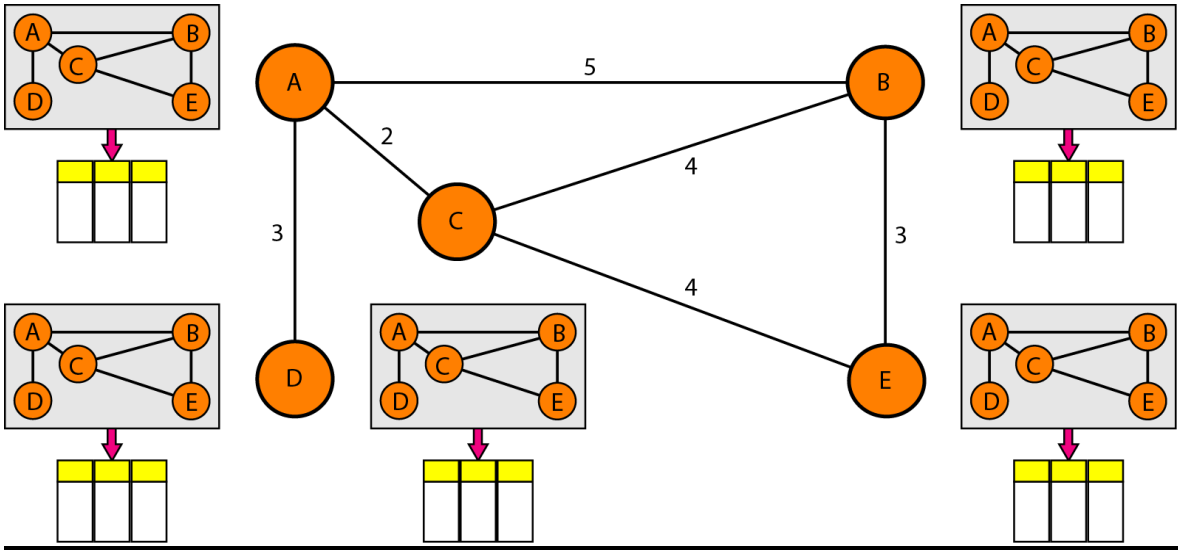
- **Initial state:** Each node knows the cost of its neighbours.
- **Final state:** Each node knows the entire graph.

Route Calculation

Each node uses Dijkstra's algorithm on the graph to calculate the optimal routes to all nodes.

- The Link state routing algorithm is also known as Dijkstra's algorithm which is used to find the shortest path from one node to every other node in the network.

FIG : 22.20 Concept of link state routing

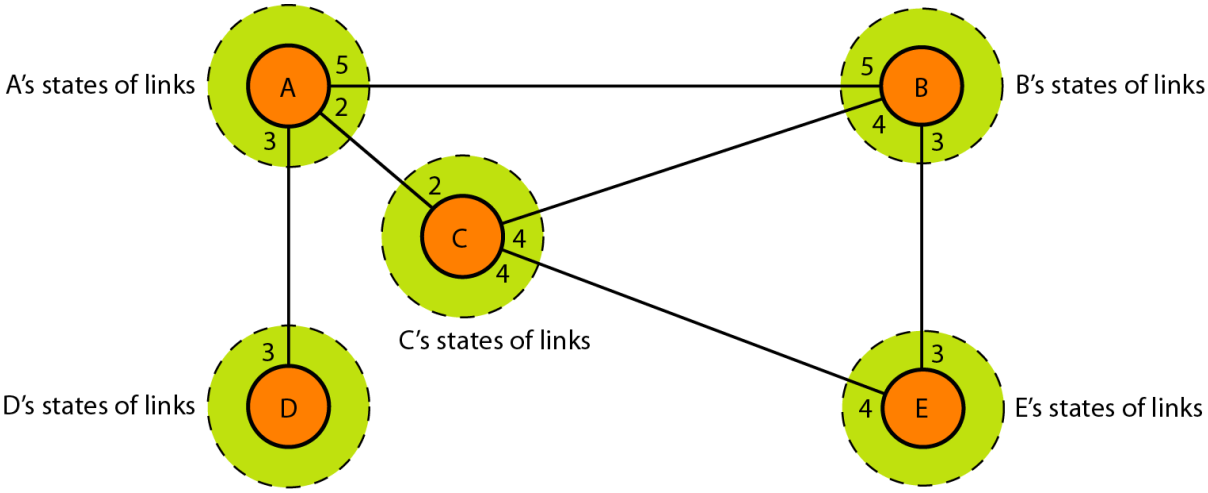


The figure shows a simple domain with five nodes. Each node uses the same topology to create a routing table, but the routing table for each node is unique because the calculations are based on different interpretations of the topology. This is analogous to a city map. While each person may have the same map, each needs to take a different route to reach her specific destination.

The topology must be dynamic, representing the latest state of each node and each link. If there are changes in any point in the network (a link is down, for example), the topology must be updated for each node.

How can a common topology be dynamic and stored in each node? No node can know the topology at the beginning or after a change somewhere in the network. Link state routing is based on the assumption that, although the global knowledge about the topology is not clear, each node has partial knowledge: it knows the state (type, condition, and cost) of its links. In other words, the whole topology can be compiled from the partial knowledge of each node. Figure 22.21 shows the same domain as in Figure 22.20, indicating the part of the knowledge belonging to each node.

Figure 22.21 Link *state knowledge*



Node A knows that it is connected to node B with metric 5, to node C with metric 2, and to node D with metric 3. Node C knows that it is connected to node A with metric 2, to node B with metric 4, and to node E with metric 4. Node D knows that it is connected only to node A with metric 3. And so on. Although there is an overlap in the knowledge, the overlap guarantees the creation of a common topology-a picture of the whole domain for each node.

Building Routing Tables

In link state routing, four sets of actions are required to ensure that each node has the routing table showing the least-cost node to every other node.

1.. Creation of the states of the links by each node, called the link state packet (LSP).

2. Dissemination of LSPs to every other router, called flooding, in an efficient and reliable way.
3. Formation of a shortest path tree for each node.
4. Calculation of a routing table based on the shortest path tree.

Creation of Link State Packet (LSP)A link state packet can carry a large amount of information. For the moment, however, we assume that it carries a minimum amount of data: the node identity, the list of links, a sequence number, and age.

The first two, node identity and the list of links, are needed to make the topology.

The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones.

The fourth, age, prevents old LSPs from remaining in the domain for a long time. LSPs are generated on two occasions:

1. When there is a change in the topology of the domain. Triggering of LSP dissemination is the main way of quickly informing any node in the domain to update its topology.

2. On a periodic basis. The period in this case is much longer compared to distance vector routing. As a matter of fact, there is no actual need for this type of LSP dissemination. It is done to ensure that old information is removed from the domain. The timer set for periodic dissemination is normally in the range of 60 min or 2 h based on the implementation. A longer period ensures that flooding does not create too much traffic on the network.

Flooding of LSPs After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbours. The process is called flooding and based on the following:

1. The creating node sends a copy of the LSP out of each interface.
2. A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following:
   a. It discards the old LSP and keeps the new one.
   b. It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

Formation of Shortest Path Tree:

Dijkstra Algorithm After receiving all LSPs, each node will have a copy of the whole topology. However, the topology is not sufficient to find the shortest path to every other node; a shortest path tree is needed.

A tree is a graph of nodes and links; one node is called the root. All other nodes can be reached from the root through only one single route. A shortest path tree is a tree in which the path between the root and every other node is the shortest. What we need for each node is a shortest path tree with that node as the root.

The Dijkstra algorithm creates a shortest path tree from a graph. The algorithm divides the nodes into two sets: tentative and permanent. It finds the neighbors of a current node, makes them tentative, examines them, and if they pass the criteria,

makes them permanent. We can informally define the algorithm by using the flowchart in Figure 22.22.

Let us apply the algorithm to node A of our sample graph in Figure 22.23. To find the shortest path in each step, we need the cumulative cost from the root to each node, which is shown next to the node.

The following shows the steps. At the end of each step, we show the permanent (filled circles) and the tentative (open circles) nodes and lists with the cumulative costs.

Figure 22.22 ***Dijkstra algorithm***

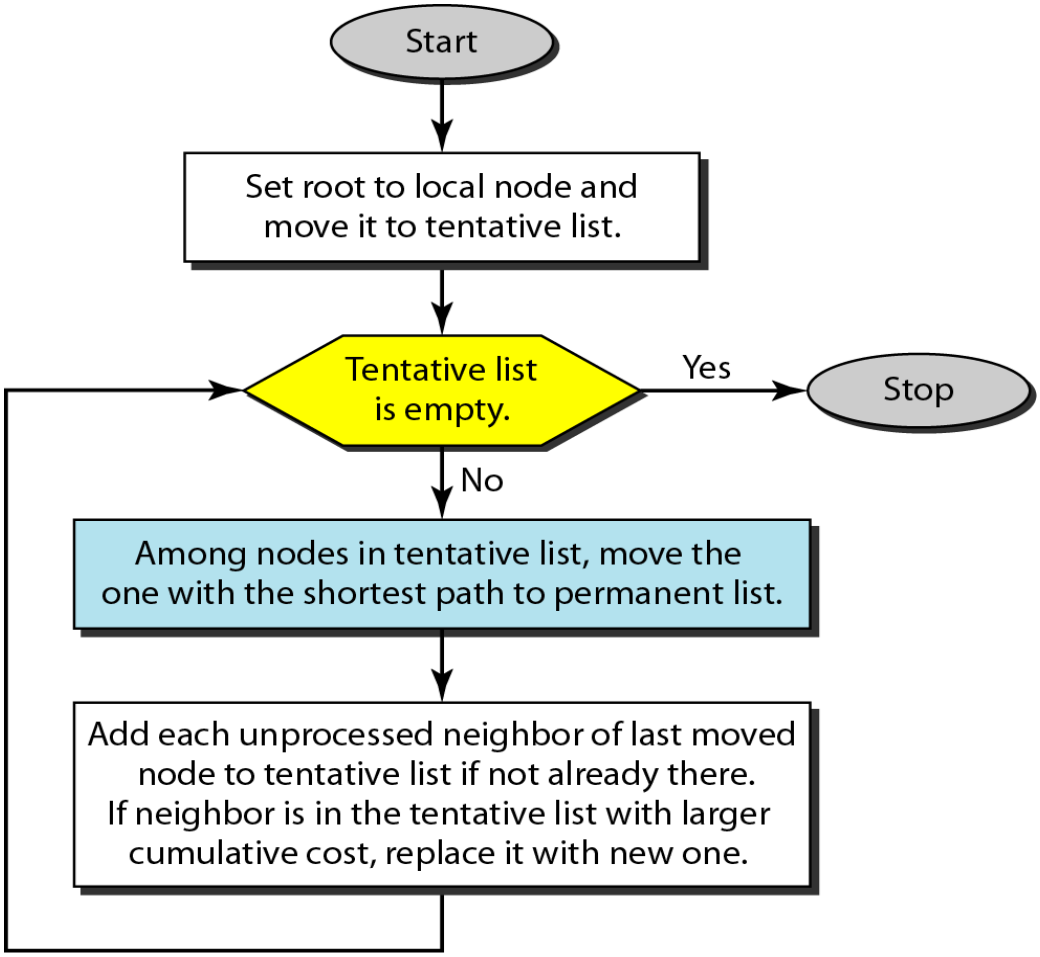


Figure 22.23 *Example of formation of shortest path tree*

1. We make node A the root of the tree and move it to the tentative list. Our

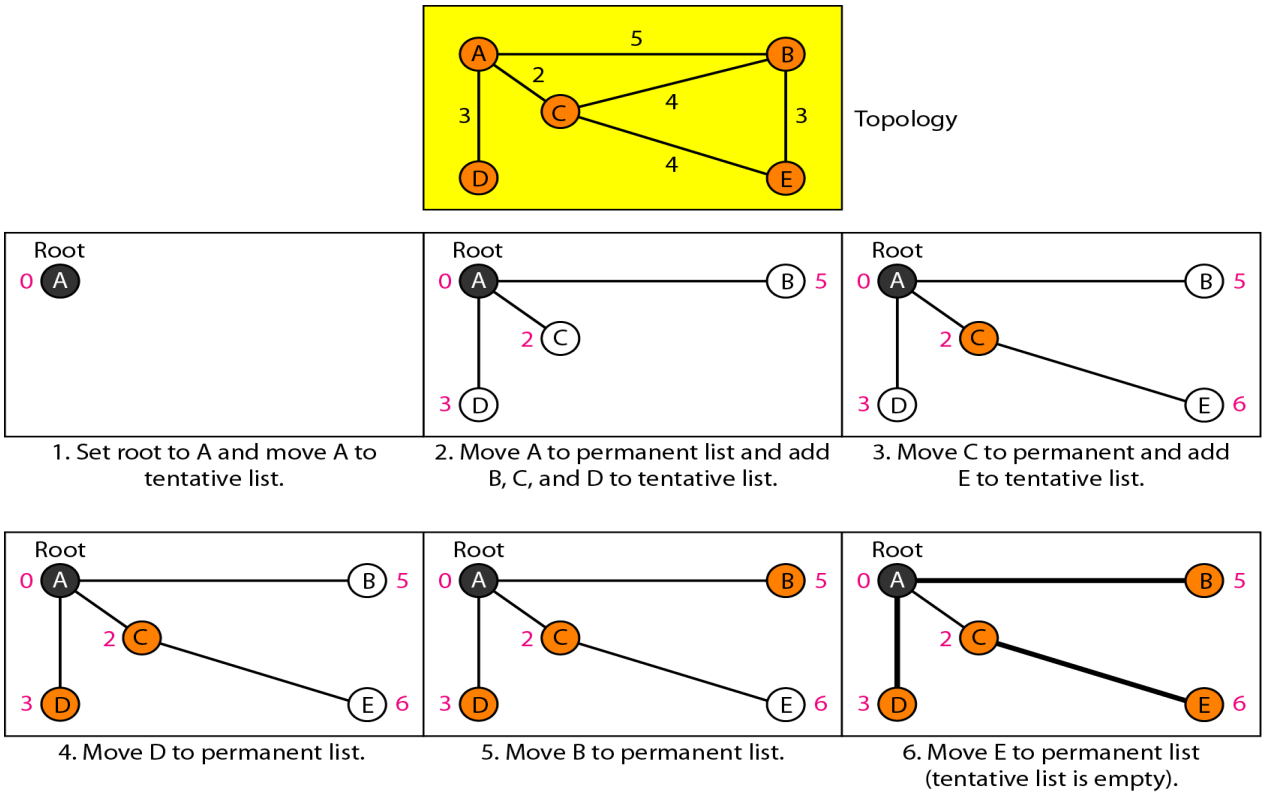
two lists are

Permanent list: empty Tentative list: A(O)

2. Node A has the shortest cumulative cost from all nodes in the tentative list. We move A to the permanent list and add all neighbors of A to the tentative list. Our new lists are

Permanent list: A(O) Tentative list: B(5), C(2), D(3)

Figure 22.23



3. Node C has the shortest cumulative cost from all nodes in the tentative list. We move C to the permanent list. Node C has three neighbors, but node A is already pro• cessed, which makes the unprocessed neighbors just B and E. However, B is already in the tentative list with a cumulative cost of 5. Node A could also reach node B through C with a cumulative cost of 6. Since 5 is less than 6, we keep node B with a cumulative cost of 5 in the tentative list and do not replace it. Our new lists are

Permanent list: A(O), e(2) Tentative list: B(5), 0(3), E(6)

4. Node D has the shortest cumulative cost of all the nodes in the tentative list. We move D to the permanent list. Node D has no unprocessed neighbor to be added to the tentative list. Our new lists are

Permanent list: A(O), C(2), 0(3) Tentative list: B(5), E(6)

5. Node B has the shortest cumulative cost of all the nodes in the tentative list. We move B to the permanent list. We need to add all unprocessed neighbors of B to the tentative list (this is just node E). However, E(6) is already in the list with a smaller cumulative cost. The cumulative cost to node E, as the neighbor of B, is 8. We keep node E(6) in the tentative list. Our new lists are

Permanent list: A(O), B(5), C(2), 0(3) Tentative list: E(6)

6. Node E has the shortest cumulative cost from all nodes in the tentative list. We move E to the permanent list. Node E has no neighbor. Now the tentative list is empty. We stop; our shortest path tree is ready. The final lists are

Permanent list: A(0), B(5), C(2), D(3), E(6) Tentative list: empty

Calculation of Routing Table from Shortest Path Tree Each node uses the shortest path tree protocol to construct its routing table. The routing table shows the cost of reaching each node from the root. Table 22.2 shows the routing table for node A.

Table 22.2 ***Routing table for node A***

Node	*Cost*	*Next Router*
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

Path Vector Routing

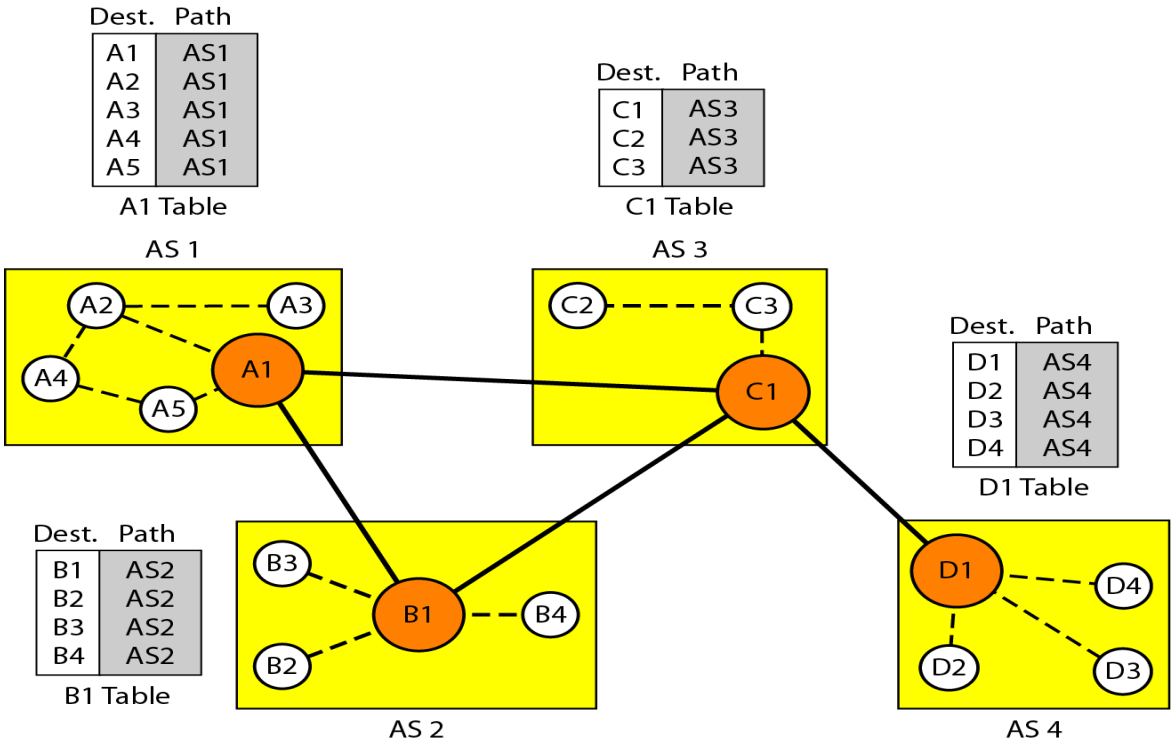
Distance vector and link state routing are both intradomain routing protocols. They can be used inside an autonomous system, but not between autonomous systems. These two protocols are not suitable for interdomain routing mostly because of scalability. Both of these routing protocols become intractable 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 proved to be 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 for our conceptual discussion) in each autonomous system that acts on behalf of the entire autonomous system. Let us call it the speaker node. The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring ASs. The idea is the same as for distance vector routing except that only speaker nodes in each AS can communicate with each other. However, what is advertised is different. A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems.

Initialization

At the beginning, each speaker node can know only the reachability of nodes inside its autonomous system. Figure 22.30 shows the initial tables for each speaker node in a system made of four ASs.

Figure 22.30 ***Initial routing tables in path vector routing***



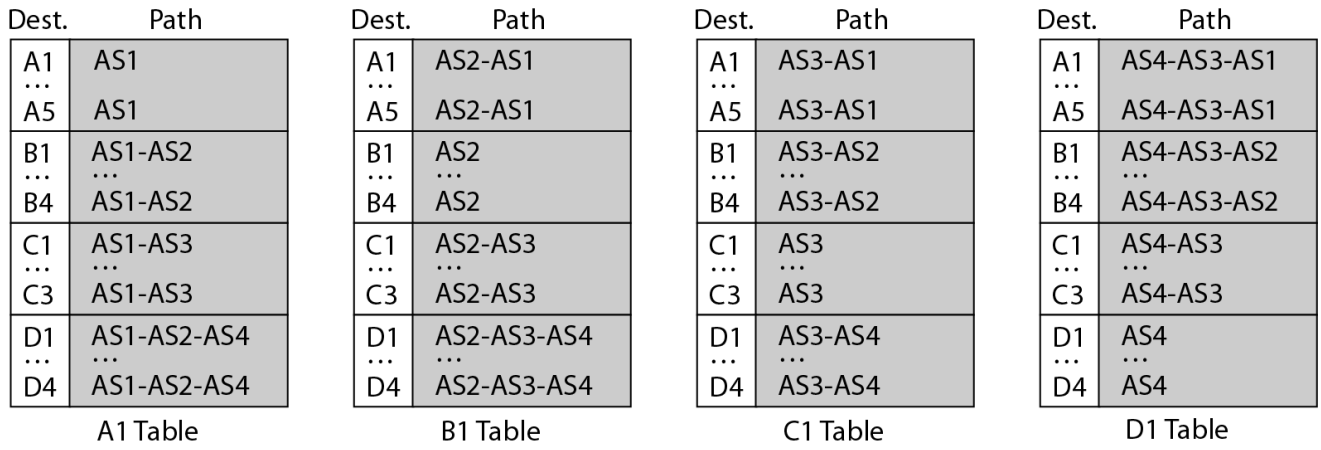
Node Al is the speaker node for ASl, Bl for AS2, Cl for AS3, and Dl for AS4. Node Al creates an initial table that shows Al to A5 are located in ASI and can be reached through it. Node Bl advertises that Bl to B4 are located in AS2 and can be reached through Bl. And so on.

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 22.30, node Al shares its table with nodes Bl and Cl. Node Cl shares its table with nodes Dl, Bl, andAl. Node Bl shares its table with Cl andAl. Node Dl shares its table with Cl.

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 ASs. Figure 22.31 shows the tables for each speaker node after the system is stabilized.

According to the figure, if router Al receives a packet for nodes A3, it knows that the path is in ASI (the packet is at home); but if it receives a packet for Dl, it knows that the packet should go from ASl, to AS2, and then to AS3. The routing table shows the path completely. On the other hand, if node Dl in AS4 receives a packet for node A2, it knows it should go through AS4, AS3, and AS 1.

Figure 22.31 ***Stabilized tables for three autonomous systems***



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.

What is the optimum path in path vector routing? We are looking

for a path to a destination that is the best for the organization that runs the autonomous system. We definitely cannot include metrics in this route because each autonomous system that is included in the path may use a different criterion for the metric. One system may use, internally, RIP, which defines hop count as the metric; another may use OSPF with minimum delay defined as the metric. The optimum path is the path that fits the organization. In our previous figure, each autonomous system may have more than one path to a destination. For example, a path from AS4 to ASI can be AS4-AS3-AS2-AS1, or it can be AS4-AS3-ASI. For the tables, we chose 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.

UNICAST ROUTING PROTOCOLS: RIP, OSPF, BGP

RIP Protocol

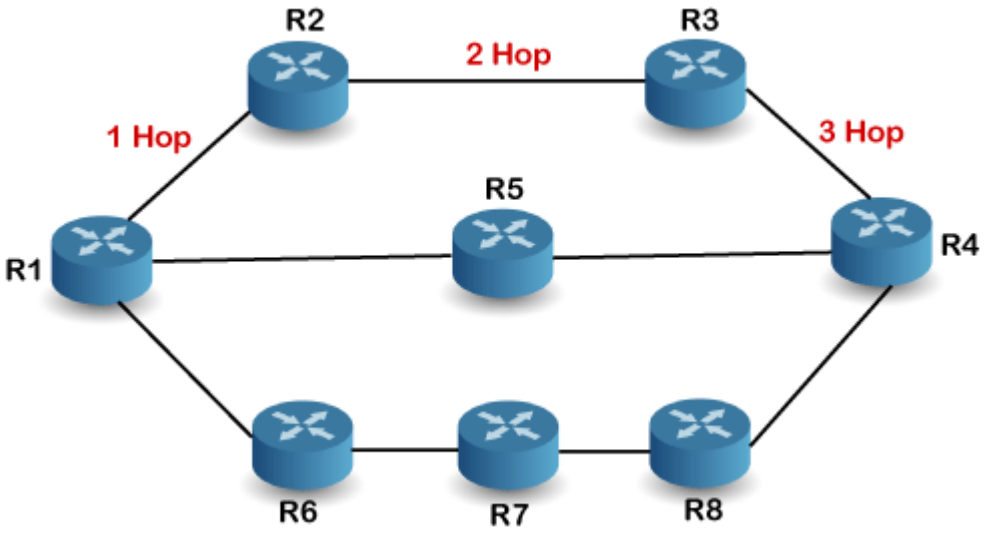
RIP stands for Routing Information Protocol. RIP is an intra-domain routing protocol used within an autonomous system. Here, intra-domain means routing the packets in a defined domain, for example, web browsing within an institutional area. To understand the RIP protocol, our main focus is to know the structure of the packet, how many fields it contains, and how these fields determine the routing table.

Before understanding the structure of the packet, we first look at the following points:

- RIP is based on the distance vector-based strategy, so we consider the entire structure as a graph where nodes are the routers, and the links are the networks.
- In a routing table, the first column is the destination, or we can say that it is a network address.
- The cost metric is the number of hops to reach the destination. The number of hops available in a network would be the cost. The hop count is the number of networks required to reach the destination.
- In RIP, infinity is defined as 16, which means that the RIP is useful for smaller networks or small autonomous systems. The maximum number of hops that RIP can contain is 15 hops, i.e., it should not have more than 15 hops as 16 is infinity.
- The next column contains the address of the router to which the packet is to be sent to reach the destination.

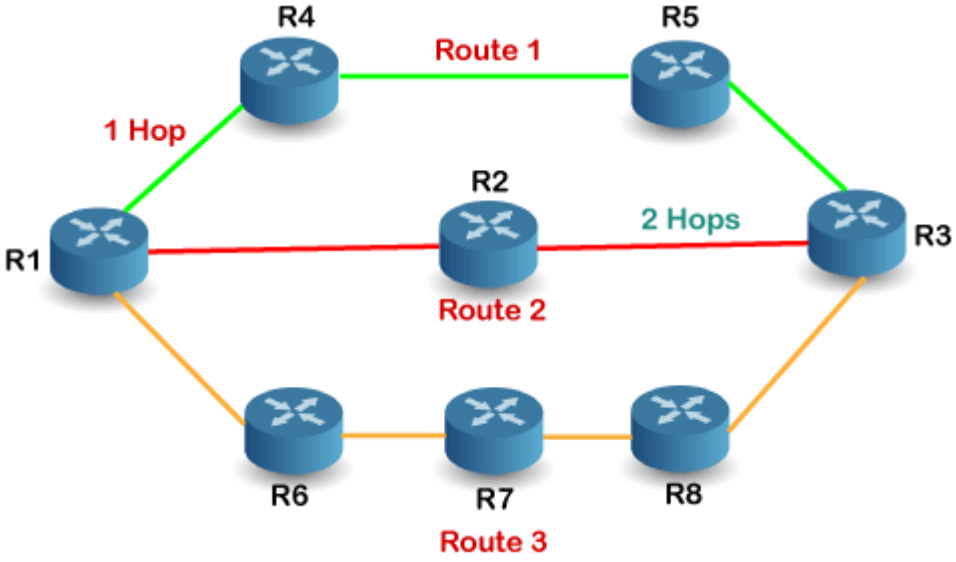
How is hop count determined?

When the router sends the packet to the network segment, then it is counted as a single hop.



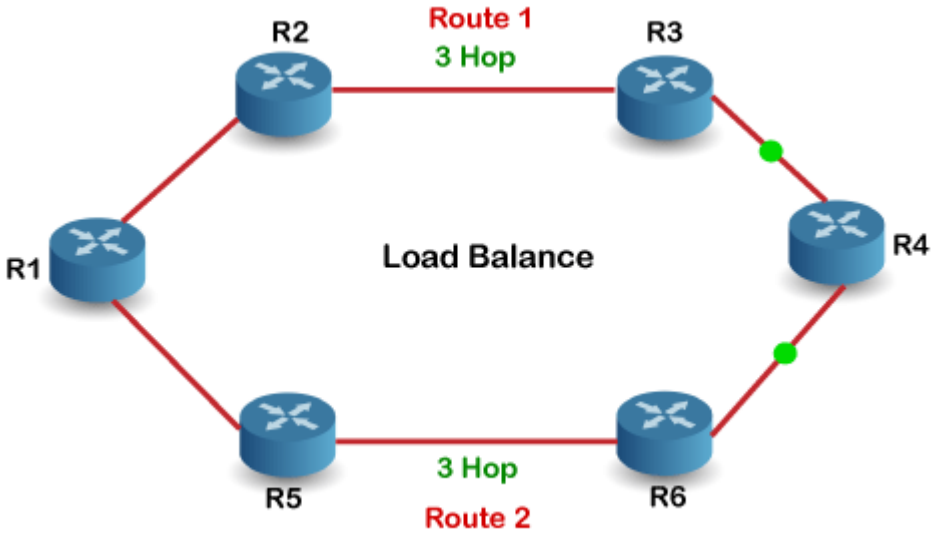
In the above figure, when the router 1 forwards the packet to the router 2 then it will count as 1 hop count. Similarly, when the router 2 forwards the packet to the router 3 then it will count as 2 hop count, and when the router 3 forwards the packet to router 4, it will count as 3 hop count. In the same way, RIP can support maximum upto 15 hops, which means that the 16 routers can be configured in a RIP.

How does the RIP work?



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

Let's look at another example.



Suppose R1 wants to send the data to R4. There are two possible routes to send data from r1 to r2. As both the routes contain the same number of hops, i.e., 3, so RIP will send the data to both the routes simultaneously. This way, it manages the load balancing, and data reach the destination a bit faster.

OSPF Protocol

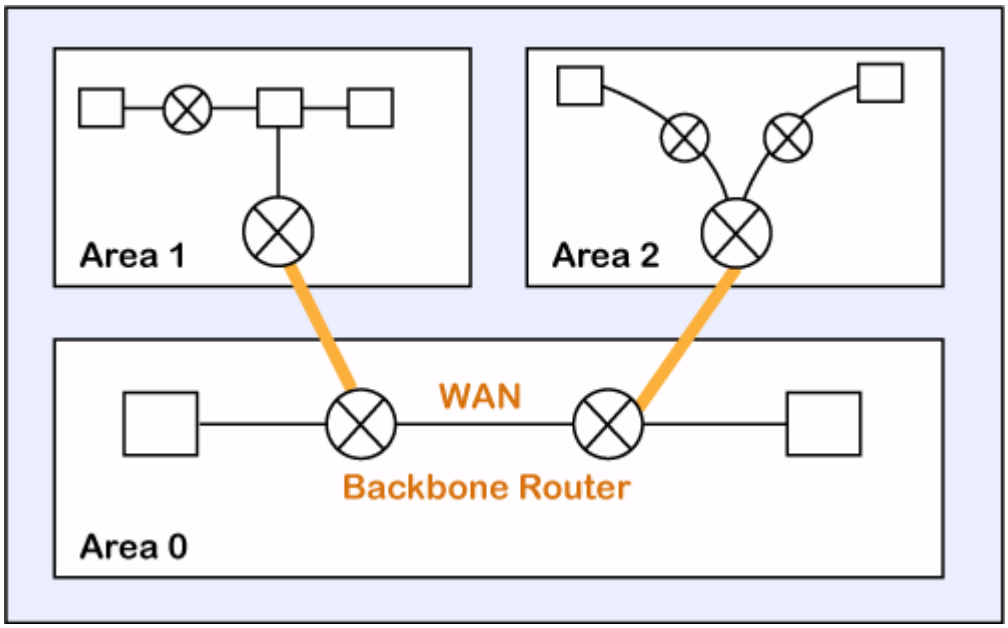
The OSPF stands for **Open Shortest Path First**. It is a widely used and supported routing protocol. It is an intradomain protocol, which means that it is used within an area or a network. It is an interior gateway protocol that has been designed within a single autonomous system.

It is based on a link-state routing algorithm in which each router contains the information of every domain, and based on this information, it determines the shortest path. The goal of routing is to learn routes.

The OSPF achieves by learning about every router and subnet within the entire network. Every router contains the same information about the network. The way the router learns this information by sending LSA (Link State Advertisements). These LSAs contain information about every router, subnet, and other networking information. Once the LSAs have been flooded, the OSPF

stores the information in a link-state database known as LSDB. The main goal is to have the same information about every router in an LSDBs.

OSPF Areas



OSPF divides the autonomous systems into areas where the area is a collection of networks, hosts, and routers. Like internet service providers divide the internet into a different autonomous system for easy management and OSPF further divides the autonomous systems into Areas.

Routers that exist inside the area flood the area with routing information

In Area, the special router also exists. The special routers are those that are present at the border of an area, and these special routers are known as Area Border Routers. This router summarizes the information about an area and shares the information with other areas.

All the areas inside an autonomous system are connected to the backbone routers, and these backbone routers are part of a primary area. The role of a primary area is to provide communication between different areas.

How does OSPF work?

There are three steps that can explain the working of OSPF:

Step 1: The first step is to become OSPF neighbors. The two connecting routers running OSPF on the same link creates a neighbor relationship.

Step 2: The second step is to exchange database information. After becoming the neighbors, the two routers exchange the LSDB information with each other.

Step 3: The third step is to choose the best route. Once the LSDB information has been exchanged with each other, the router chooses the best route to be added to a routing table based on the calculation of SPF.

Types of links in OSPF

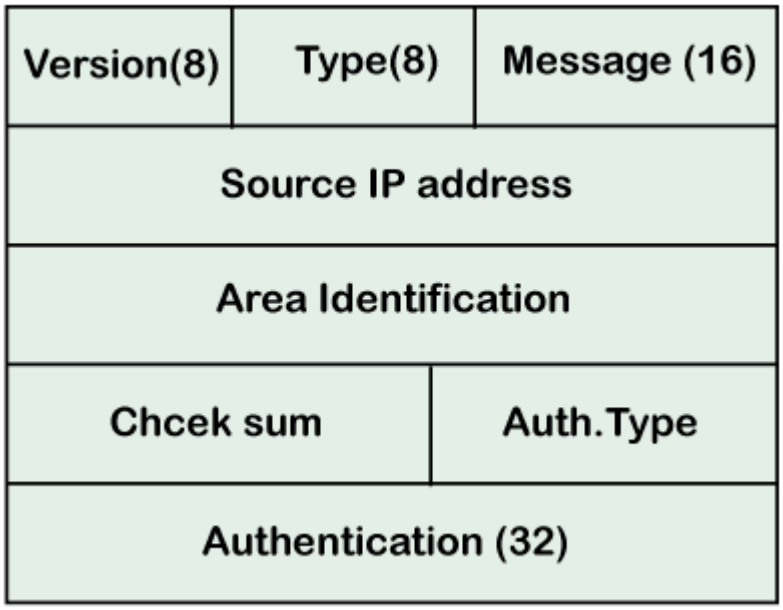
A link is basically a connection, so the connection between two routers is known as a link.

There are four types of links in OSPF:

1. **Point-to-point link:** The point-to-point link directly connects the two routers without any host or router in between.
2. **Transient link:** When several routers are attached in a network, they are known as a transient link.
   The transient link has two different implementations:
   Unrealistic topology: When all the routers are connected to each other, it is known as an unrealistic topology.
   Realistic topology: When some designated router exists in a network then it is known as a realistic topology. Here designated router is a router to which all the routers are connected. All the packets sent by the routers will be passed through the designated router.
3. **Stub link:** It is a network that is connected to the single router. Data enters to the network through the single router and leaves the network through the same router.
4. **Virtual link:** If the link between the two routers is broken, the administration creates the virtual path between the routers, and that path could be a long one also.

OSPF Message Format

The following are the fields in an OSPF message format:



- **Version:** It is an 8-bit field that specifies the OSPF protocol version.
- **Type:** It is an 8-bit field. It specifies the type of the OSPF packet.
- **Message:** It is a 16-bit field that defines the total length of the message, including the header. Therefore, the total length is equal to the sum of the length of the message and header.
- **Source IP address:** It defines the address from which the packets are sent. It is a sending routing IP address.
- **Area identification:** It defines the area within which the routing takes place.
- **Checksum:** It is used for error correction and error detection.
- **Authentication type:** There are two types of authentication, i.e., 0 and 1. Here, 0 means for none that specifies no authentication is available and 1 means for pwd that specifies the password-based authentication.
- **Authentication:** It is a 32-bit field that contains the actual value of the authentication data.

OSPF Packets

There are five different types of packets in OSPF:

- Hello
- Database Description

- Link state request
- Link state update
- Link state Acknowledgment

Let's discuss each packet in detail.

1. Hello packet

The Hello packet is used to create a neighborhood relationship and check the neighbor's reachability. Therefore, the Hello packet is used when the connection between the routers need to be established.

2. Database Description

After establishing a connection, if the neighbor router is communicating with the system first time, it sends the database information about the network topology to the system so that the system can update or modify accordingly.

3. Link state request

The link-state request is sent by the router to obtain the information of a specified route. Suppose there are two routers, i.e., router 1 and router 2, and router 1 wants to know the information about the router 2, so router 1 sends the link state request to the router 2. When router 2 receives the link state request, then it sends the link-state information to router 1.

4. Link state update

The link-state update is used by the router to advertise the state of its links. If any router wants to broadcast the state of its links, it uses the link-state update.

5. Link state acknowledgment

The link-state acknowledgment makes the routing more reliable by forcing each router to send the acknowledgment on each link state update. For example, router A sends the link state update to the router B and router C, then in return, the router B and C sends the link-state acknowledgment to the router A, so that the router A gets to know that both the routers have received the link-state update.

OSPF States

The device running the OSPF protocol undergoes the following states:

- **Down:** If the device is in a down state, it has not received the HELLO packet. Here, down does not mean that the device is physically down; it means that the OSPF process has not been started yet.

- **Init:** If the device comes in an init state, it means that the device has received the HELLO packet from the other router.
- **2WAY:** If the device is in a 2WAY state, which means that both the routers have received the HELLO packet from the other router, and the connection gets established between the routers.
- **Exstart:** Once the exchange between the routers get started, both the routers move to the Exstart state. In this state, master and slave are selected based on the router's id. The master controls the sequence of numbers, and starts the exchange process.
- **Exchange:** In the exchange state, both the routers send a list of LSAs to each other that contain a database description.
- **Loading:** On the loading state, the LSR, LSU, and LSA are exchanged.
- **Full:** Once the exchange of the LSAs is completed, the routers move to the full state.

BGP4 overview

Border Gateway Protocol version 4 (BGP4) is an exterior gateway protocol that performs inter-autonomous system (AS) or inter-domain routing. It peers to other BGP-speaking systems over TCP to exchange network reachability and routing information.

BGP primarily performs two types of routing: inter-AS routing, and intra-AS routing. BGP peers belonging to different autonomous systems use the inter-AS routing, referred as Exterior BGP (eBGP). On the other hand, within an AS BGP can be used to maintain a consistent view of network topology, to provide optimal routing, or to scale the network.

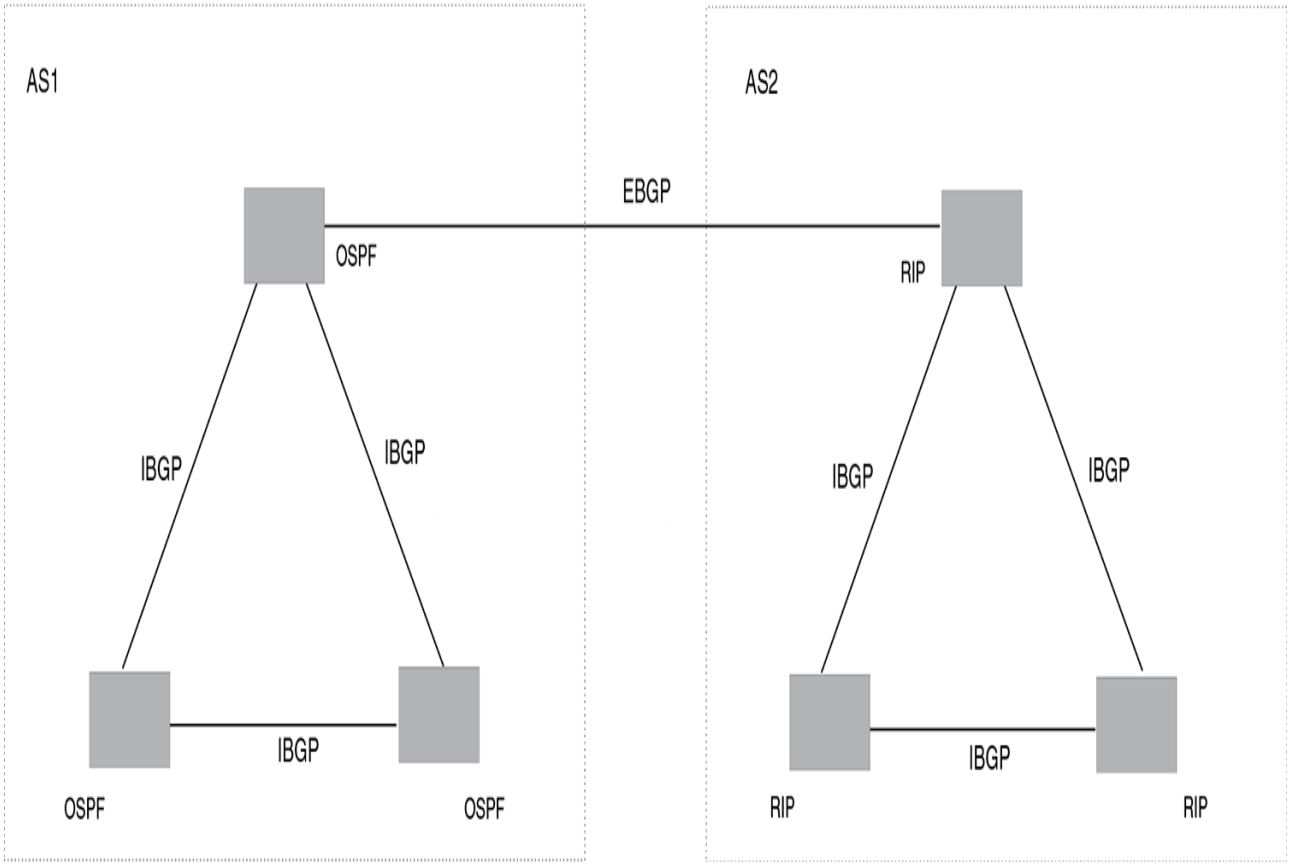
BGP is a path vector protocol and implements this scheme on large scales by treating each AS as a single point on the path to any given destination. For each route (destination), BGP maintains the AS path and uses this to detect and prevent loops between autonomous systems.

Devices within an AS can use different Interior Gateway Protocols (IGPs) such as RIP and OSPF to communicate with one another.

However, for devices in different autonomous systems to communicate, they need to use an EGP. BGP4 is the standard EGP used by Internet devices and therefore is the EGP implemented on Brocade devices.

This is a simple example of two BGP4 ASs. Each AS contains three BGP4 devices. All of the BGP4 devices within an AS communicate using iBGP. BGP4 devices communicate with other autonomous systems using eBGP. Notice that each of the devices also is running an Interior Gateway Protocol (IGP). The devices in AS1 are running OSPF and the devices in AS2 are running RIP. The device can be configured to redistribute routes among BGP4, RIP, and OSPF. They also can redistribute static routes.

FIGURE 29 BGP4 autonomous systems



Difference between EBGP and IBGP

1. **External Border Gateway Protocol (EBGP) :**
   EBGP is used between autonomous systems. It is used and implemented at the

edge or border router that provides inter-connectivity for two or more autonomous system. It functions as the protocol responsible for interconnection of networks from different organizations or the Internet.

2. Internal Border Gateway Protocol (IBGP) :
IBGP is used inside the autonomous systems. It is used to provide information to your internal routers. It requires all the devices in same autonomous systems to form full mesh topology or either of Route reflectors and Confederation for prefix learning.