# UNIT III ROUTING

Routing – Network as Graph - Distance Vector – Link State – Global Internet –Subnetting - Classless Routing (CIDR) - BGP- IPv6 – Multicast routing - DVMRP- PIM.

## ROUTING:

- In a network there are multiple routes available between a source and a destination **The process of finding the shortest route from the source to a destination is defined as routing.**
- The routing table in a router is the table which contains the shortest route to reach a destination.
- It is built up by the routing algorithms. It generally contains mappings from network numbers to next hops.

## Forwarding versus Routing

**Forwarding**:

Used to o select an output port based on destination address and routing table

**Routing**:

Process by which routing table is built

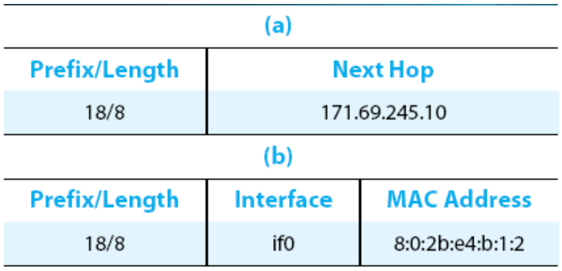
## Forwarding table VS Routing table

**Forwarding table**

- Used when a packet is being forwarded and so must contain enough information to accomplish the forwarding function
- A row in the forwarding table contains the mapping from a network number to an outgoing interface and some MAC information, such as Ethernet Address of the next hop. (Shown in fig. b)

**Routing table**

- Built by the routing algorithm
- Generally, contains mapping from network numbers to next hops.(shown in fig. a)



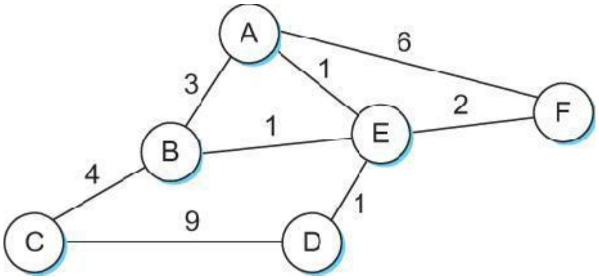
(a)

| Prefix/Length | Next Hop |
|---|---|
| 18/8 | 171.69.245.10 |

(b)

| Prefix/Length | Interface | MAC Address |
|---|---|---|
| 18/8 | if0 | 8:0:2b:e4:b:1:2 |

## Network as a Graph



- The basic problem of routing is to find the lowest-cost path between any two nodes
- Where the cost of a path equals the sum of the costs of all the edges that make up the path
- For a simple network, we can calculate all shortest paths and load them into some nonvolatile storage on each node.
- Such a static approach has several shortcomings
- It does not deal with node or link failures
- It does not consider the addition of new nodes or links
- It implies that edge costs cannot change

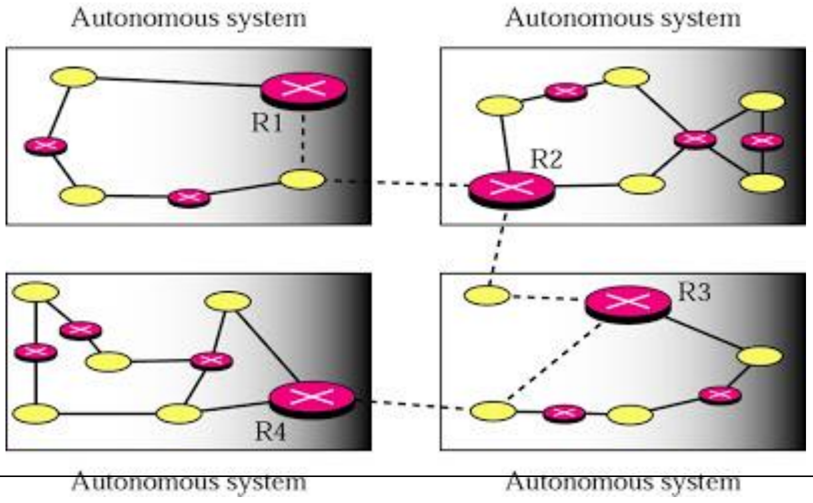
Hence we need a solution in the form of dynamic protocols (implemented with routing algorithms) to find shortest route between nodes.

Before starting with it, we need to know that Internet is a Collection of Millions of Networks and we can't represent the entire Internet as a single entity (single network/graph) to find the shortest routes between nodes. The Internet is divided into blocks termed as Autonomous Systems.

## Grouping of Collection of Networks in Internetworks: Autonomous system

An autonomous system (AS) is a network or a collection of networks that are all managed and supervised by a single entity or organization.It is a group of networks and routers under authority of a single administrator. An AS is a heterogeneous network typically governed by a large enterprise.

The number of unique autonomous networks in the routing system of the Internet exceeded 5,000 in 1999, 30,000 in late 2008, 35,000 in mid-2010, 42,000 in late 2012, 54,000 in mid-2016 and 60,000 in early 2018.



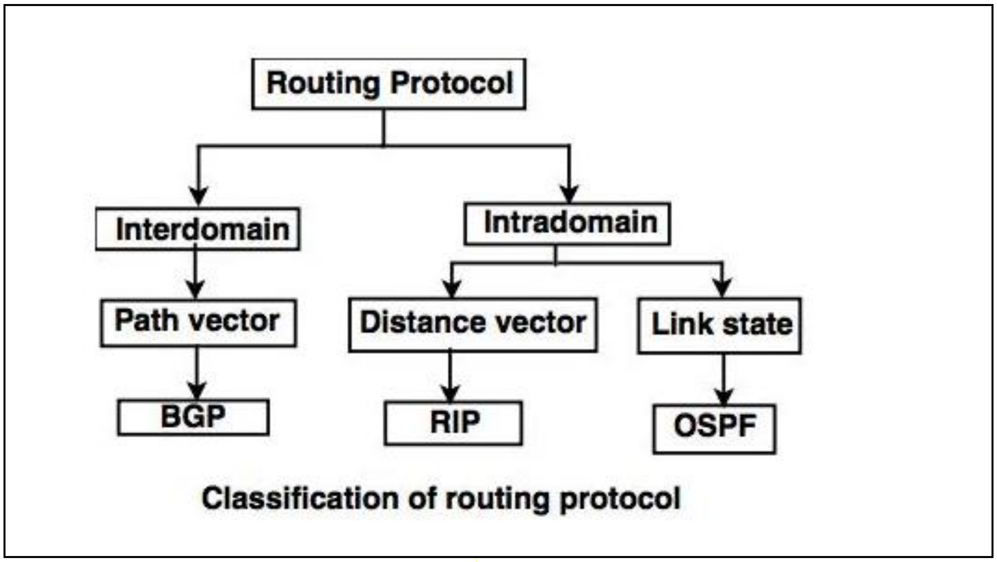
Since the Internet is divided into Autonomous Systems(AS), it is necessary to perform the routing process within an AS and between ASs.

**Intra-Domain Routing Protocols (Interior Gateway Protocols):**

- Used to construct routing table within an Autonomous System.
- RIP (Routing Information Protocol) is based on Distance Vector Routing
- OSPF (Open Shortest Path First Protocol is based on Link State Routing

**Inter-Domain Routing Protocols (Exterior Gateway Protocols):**

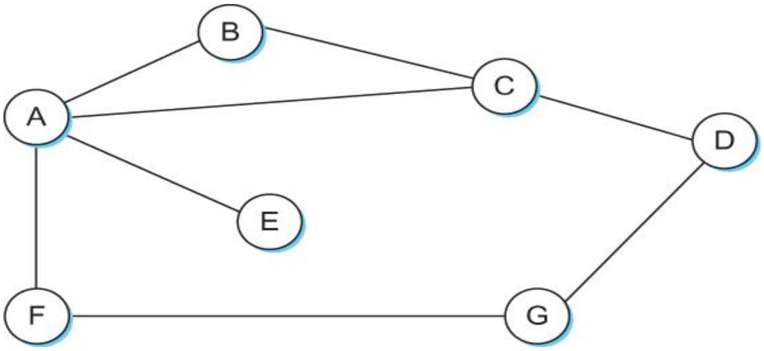
- Used to construct routing table between Autonomous Systems
- BGP(Border Gateway Protocol) is based on Path Vector Routing.



Classification of routing protocol

# DISTANCE VECTOR ROUTING

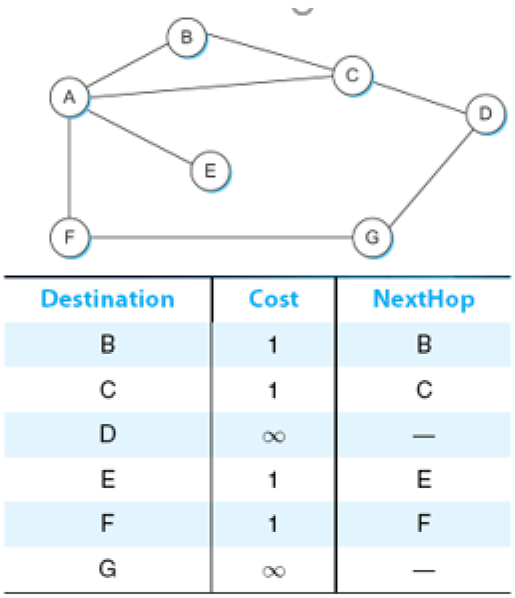
- Working Principle: Each node(router) shares the entire routing information (entire AS) to its neighbors periodically.
- That is, each node constructs a one dimensional array (a vector) containing the "distances" (costs) to all other nodes and distributes that vector to its immediate neighbors
- Starting assumption is that each node knows the cost of the link to each of its directly connected neighbors.
- Consider the below example,



The below table shows the global view of all vectors at each node, first row is the initial vector at router A and so on for each router.

| 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 |

The Initial Routing Table of Router A is Shown below,



| Destination | Cost | NextHop |
|---|---|---|
| B | 1 | B |
| C | 1 | C |
| D | ∞ | — |
| E | 1 | E |
| F | 1 | F |
| G | ∞ | — |

Initial routing table at node A

Whenever a node receives the routing information from another node, it applies an updating algorithm to update its routing table,

Before applying the algorithm, it does two steps,

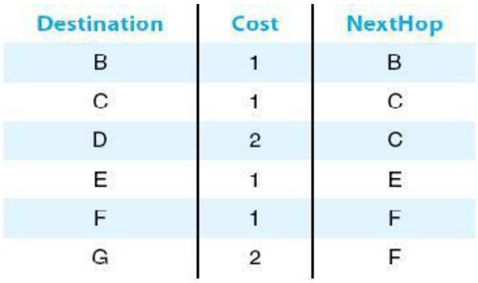
- Let ‘D’ represent each entry in the received Routing Information.
- **Increment the cost of each entry by 1.**
- Let ‘c’ be the node from which the routing information came.
- **Add a column with a next hop value to be ‘c’ from which the routing information came.**
- Let ‘A’ represent the Router.

The Router updates its own table according to the following three rules:

1. New destination: D is a previously unknown destination. Router A adds ⟨D,cost,c⟩ to its routing table.
2. Lower cost: D is a known destination with entry ⟨D,cost(old),c⟩, but the new total cost c is less than the old. A switches to the cheaper route, updating its entry for D to ⟨D,cost(new),c⟩. otherwise it retains the old entry.
3. Next_hop increase: A has an existing entry ⟨D,cost(old),c⟩and the new total cost c is greater than the old cost. Because this is a cost increase from the neighbor c that A is currently using to reach D, A must incorporate the increase in its table. A updates its

Eg:(follow class notes)

After receiving the distance vector of its neighbors the routing table at node A is as follows,

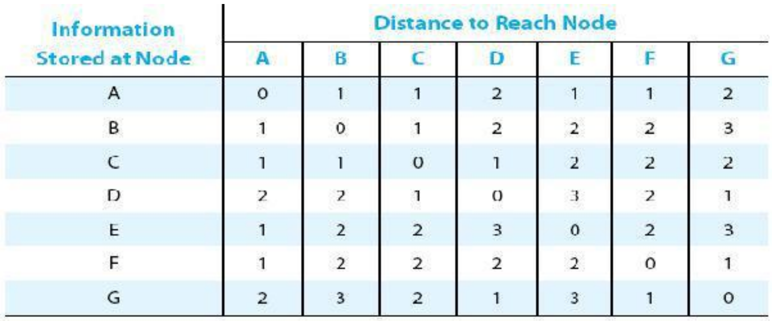


| Destination | Cost | NextHop |
|---|---|---|
| B | 1 | B |
| C | 1 | C |
| D | 2 | C |
| E | 1 | E |
| F | 1 | F |
| G | 2 | F |

Final routing table at node A

After a few exchanges of information between neighbors, all nodes have consistent routing table with correct distance information. The process of getting consistent routing information to all the nodes is calledconvergence.

The following figure shows the final distances stored at each node.



| 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 |

Final distances stored at each node (global view)

There are two different conditions under which a node decides to send a routing table to neighbors

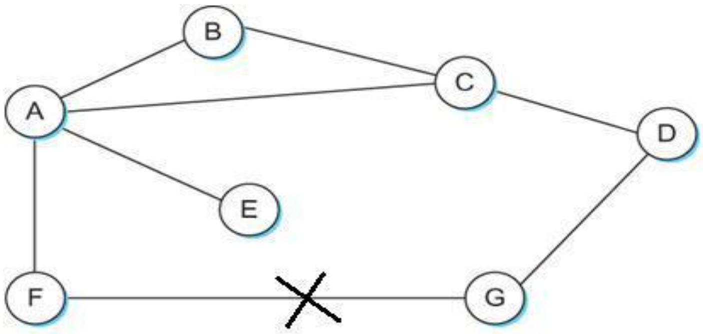
1. **Periodic update**

   Each node automatically sends an update message every seconds orminutes even though there is no change

2. **Triggered update**
   - whenever a node's routing table changes, it sends its updated distance information to its neighbors.

   - Triggered updates are sent generally, whenever a link fails and causes changes in the routing table.

## Link Failure example

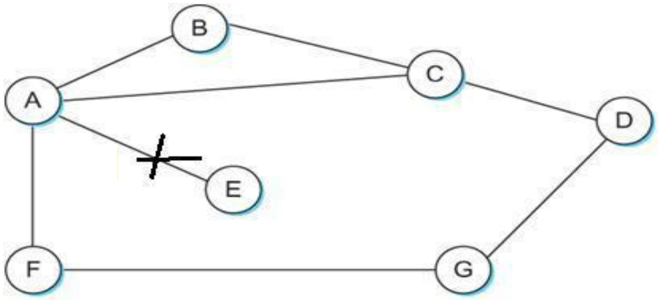


Consider, in this F detects its link to G is failed.

When a node detects a link failure (G)

1. F detects that link to G has failed
2. F sets distance to G to infinity and sends update to A
3. A sets distance to G to infinity since it uses F to reach G
4. A receives periodic update from C with 2-hop path to G
5. A sets distance to G to 3 and sends update to F
6. F decides it can reach G in 4 hops via A

## Link Failure with Count-to-infinity problem



In this example, consider the link A-E fails.

1. Slightly different circumstances can prevent the network from stabilizing
   a. Suppose the link from A to E goes down
   b. In the next round of updates, A advertises a distance of infinity to E, but B and C advertise a distance of 2 to E
2. Depending on the exact timing of events, the following might happen
   a. Node B, upon hearing that E can be reached in 2 hops from C, concludes that it can reach E in 3 hops and advertises this to A
   b. Node A concludes that it can reach E in 4 hops and advertises this to C
   c. Node C concludes that it can reach E in 5 hops; and so on.
   d. This cycle stops only when the distances reach some number that is large enough to be considered infinite

## Partial solutions to Count to infinity problems:

1. Use some relatively small number as an **approximation of infinity.**
   - For example, the maximum number of hops to get across a certain network is never going to be **more than 16.**

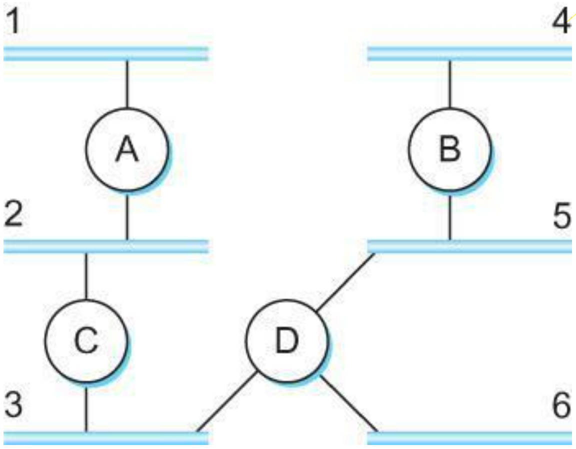
2. One technique to improve the time to stabilize routing is called ***splithorizon*** *(means*Never sent same information back to the interface itcame from)
   - When a node sends a routing update to its neighbors, it does not send those routes it learned from each neighbor back to that neighbor
   - For example, if B has the route (E, 2, A) in its table, then it knows it must have learned this route from A, and so whenever B sends a routing update to A, it does not include the route (E, 2) in that update

3. In a stronger version of split horizon, called ***split horizon withpoison reverse.***
   - B actually sends that back route to A, but it puts negative information in the route to ensure that A will not eventually use B to get to E
   - For example, B sends the route (E, ∞) to A

## Routing Information Protocol (RIP)

- Widely used routing protocol in IP networks
- It is a routing protocol built on distance vector algorithm
- Routers running RIP actually advertise distances to networks. They send periodic updates every 30 seconds and triggered updates when the routing table changes.
- RIP uses link costs equal to 1 and uses 16 to represent infinity. So a network running RIP must have a maximum hops of 16.

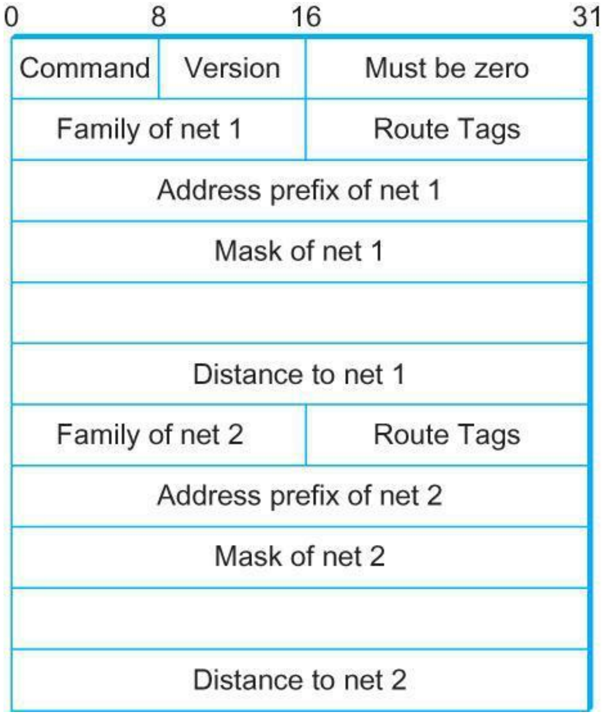
The following figure shows an example network running on RIP.



In this example each network, Router C advertises to router A that it can reach,
Nw 2 and 3 at cost 0
Nw 5 and 6 at cost 1
Nw 4 at cost 2

## RIP Packet Format

RIP version 2 supports CIDR.
RIP messages are encapsulated in a UDP datagram
RIP uses the services of UDP on well-known port 520



Command - request or response
version - 2
must be zero - unused
Address - IP address
Family - network address designed to carry information to different Protocols
Distance - metric value that determines how many hops to reach its destination
(1 to 15 are valid routes, 16 is unreachable)
mask - subnet masking
next hop - indicates the IP address of the next hop

Route tag- Distinguishes between internal & external routes (internal routes are learned by diff protocols like RIP, OSPF etc)
*(external routes are learn using only one protocol. Eg: BGP)*

## LINK STATE ROUTING

- Link-state routing is the second major class of intra-domain routing protocol.
- The starting assumptions for link-state routing are rather similar to those for distance vector routing.
- The basic idea behind link-state protocols is very simple: Everynode nowshow to reach its directly connected neighbors, and if we make sure that the totality of this knowledge is disseminated to every node, then every node will have enough knowledge of the network to build a complete map of the network.
- **Working Principle: Each node shares the Routing Information about the neighbors to all the other nodes periodically.**

**LINK STATE PACKET:**

- Each router creates a link state packet (LSP) which contains names (e.g. network addresses) and cost to each of its neighbours.
- Information in LSP:
    - id of the node that created the LSP
    - cost of link to each directly connected neighbor
    - sequence number (SEQNO)
    - time-to-live (TTL) for this packet
- The LSP is transmitted to all other routers, who each update their own records
- When a router receives LSPs from all routers, it can use (collectively) that information to construct the routing table.

## How LSP is shared to all routers? Reliable Flooding

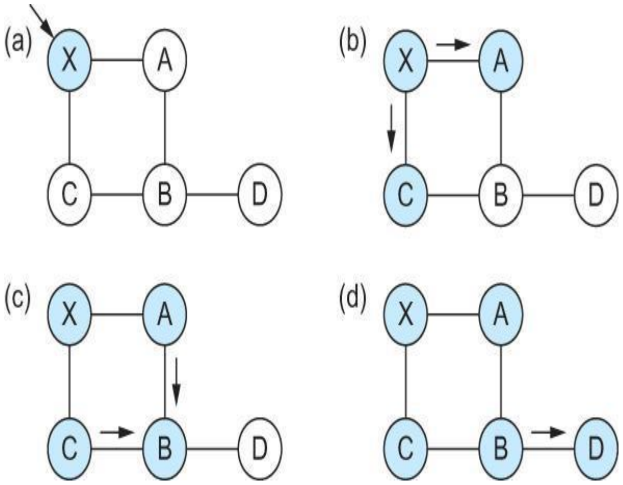
- *Reliable flooding* is the process of making sure that all the nodesparticipating in the routing protocol get a copy of the link-state information from all the other nodes.
    - As the term —flooding suggests, the basic idea is for a node to send its link-state information out on all of its directly connected links, with each node that receives this information forwarding it out on all of *its* links.
- Initially each node knows only the state of the link to each of its neighbor.
- The information of each node is put into update packet called as Link State Packet (LSP) and its flooded to all other packets. Ie This process continues until the information has reached all the nodes in the network.

In short the process is can be summarized as,

- store most recent LSP from each node
- forward LSP to all nodes but one that sent it
- generate new LSP periodically; increment SEQNO
- start SEQNO at 0 when reboot
- decrement TTL of each stored LSP; discard when TTL=0

**In flooding, each node forwards the LSP to all neighbors except the one from which the LSP was received.**

**EXAMPLE: LSP FLOODED IN A SMALL NETWORK**



Flooding of link-state packets.
(a) LSP arrives at node X;
(b) X floods LSP to A and C;
(c) A and C flood LSP to B (but not X);
(d) flooding is complete

Each node becomes shaded as it stores the new LSP. In Figure (a) the LSP arrives at node X, which sends it to neighbors A and in Figure (b) A and C do not send it back to X, but send it on to B. Since B receives two identical copies of the LSP, it will accept whichever arrived first and ignore the second as a duplicate. It then passes the LSP on to D, who has no neighbors to flood it to, and the process is complete.

Just as in RIP, each node generates LSPs under two circumstances. Either the expiry of a periodic timer or a change in topology can cause a node to generate a new LSP.

LSP contains sequence numbers that make it possible to distinguish new LSP from old one.

## Route Calculation-SHORTEST PATH ALGORITHM

In practice, each router computes its routing table directly from the LSPs it has collected.

Once a node has a copy of LSP from every other node, it can compute a complete map of network topology and finds the shortest route (routing table) using Dijkstra's algorithm called the *forward search* algorithm.

- Specifically, each router maintains two lists, known as
  1. **Tentative**
  2. **Confirmed**
- Each of these lists contains a set of entries of the form

  (Destination, Cost, NextHop)

## The algorithm

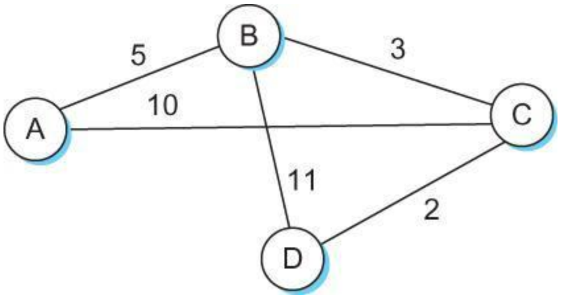
1. Initialize the **Confirmed** list with an entry for myself; this entry has a cost of 0
2. For the node just added to the **Confirmed** list in the previous step, call it node **Next**, select its LSP
3. For each neighbor (Neighbor) of **Next**, calculate the cost (Cost) to reach this Neighbor as the sum of the cost from myself to Next and from Next to Neighbor

a) If Neighbor is currently on neither the **Confirmed** nor the **Tentative** list, then add (Neighbor, Cost, Nexthop) to the **Tentative** list, where Nexthop is the direction I go to reach Next.

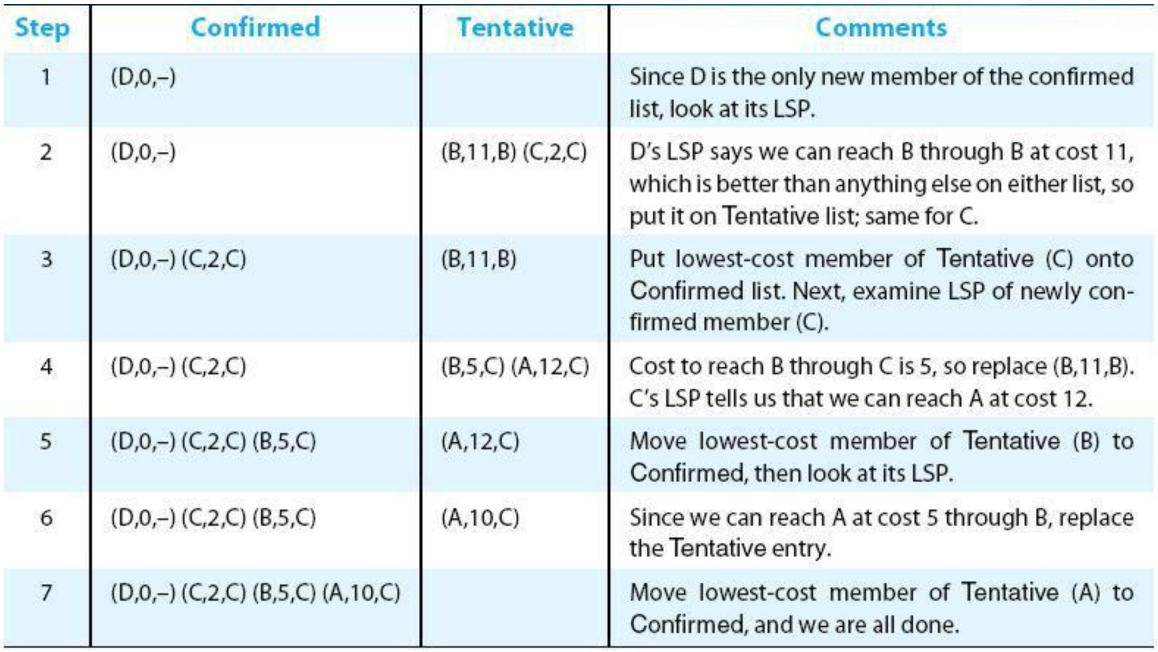
b) If Neighbor is currently on the **Tentative** list, and the Cost is less than the currently listed cost for the Neighbor, then replace the current entry with **(Neighbor, Cost, Nexthop)** where Nexthop is the direction I go to reach Next.

4. If the **Tentative** list is empty, stop. Otherwise, pick the entry from the **Tentative** list with the lowest cost, move it to the **Confirmed** list, and return to Step 2.

Consider an example network for link-state routing



The following table list the steps for building routing table for **node D.**



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

**The difference between distance vector and link state routing algorithm is**– in distance vector, each node talks only to its directly connected neighbors about the distance to all nodes. In link state, each node talks to all other nodes only about the state of its directly connected links.

## OSPF:OPEN SHORTEST PATH FIRST PROTOCOL

- One of the most widely used link-state routing protocols is OSPF.
- The first word, —Open, refers to the fact that it is an open, nonproprietary standard, created under the auspices of the IETF. The —SPF part comes from an alternative name for link state routing

OSPF adds quite a number of features to the basic link-state algorithm described above, including the following:

- Authentication of routing messages
- Additional hierarchy
- Load balancing

Authentication of routing messages

Protection against misconfigured routers by providing 8 byte password for authentication

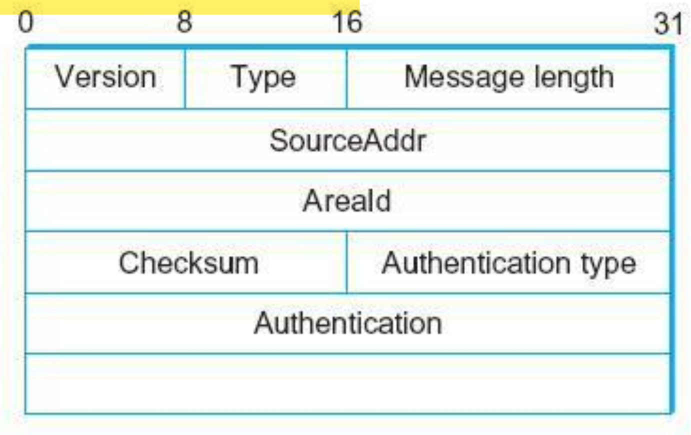
Additional hierarchy

- ➢ Introduces additional hierarchy by allowing arouting domain to be portioned into areas.
- ➢ Reduces the amount of information that must be transmitted to and stored in each node.
- ➢ Router doesn't need to know how to reach each network in its domain; it may know how to get to right area.

Load balancing

OSPF allows distributing traffic among multiple routes of same cost.

## OSPF Header Format



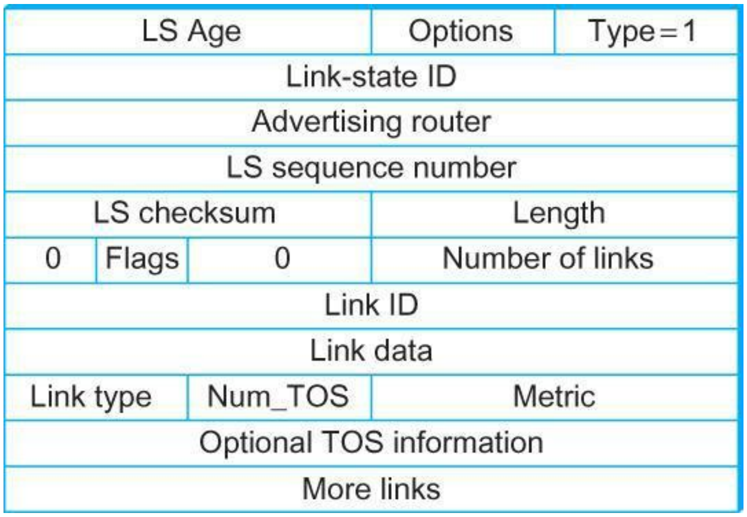
| 0 | 8 | 16 | 31 |
|---|---|---|---|
| Version | Type | Message length | |
| SourceAddr | | | |
| AreaId | | | |
| Checksum | | Authentication type | |
| Authentication | | | |
| | | | |

- Version - set to 2
- Type – may take the values 1 through 5
- SourceAddr - identifies the sender of the message
- Authentication type
  - 0 if authentication is used
  - 1 if password is used
  - 2 if cryptographic authentication checksum is used
- Area id – 32 bit identifier of the area in which node is located
- Authentication - password or cryptographic checksum Checksum - the entire packet, except the authentication data, is protected by a 16-bit checksum using the same algorithm as the IP header.

Type - OSPF Message Types
Type 1 -> “hello” msg (notficationmsg to nofity that it is alive)
Type 2 -> request
Type 3 -> response
Type 4 –> send
Type 5 -> acknowledge the receipt of link state msg

The basic building blocks of link state messages is known as link state advertisement (LSA). One message may contain one or many LSAs.

The packet format for type1 link-state advertisement is as follows:-



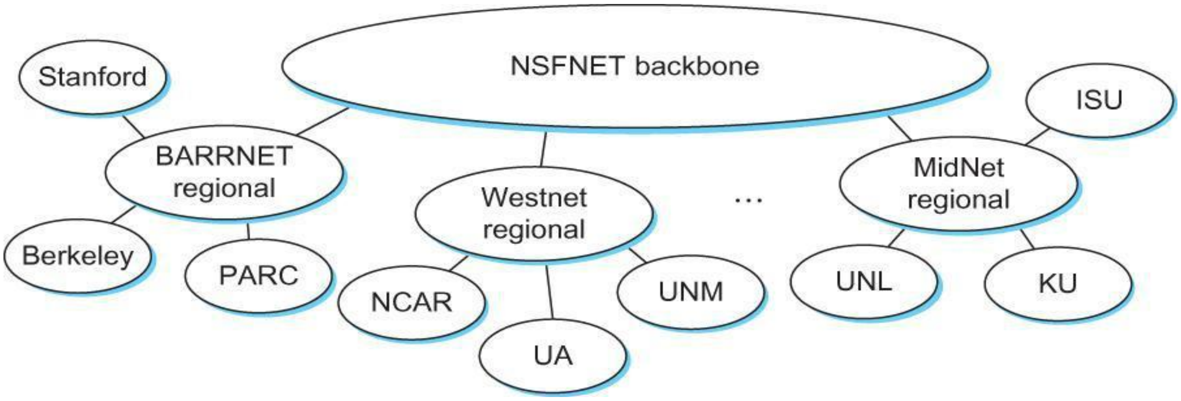
| | | | |
|---|---|---|---|
| LS Age | | Options | Type = 1 |
| Link-state ID | | | |
| Advertising router | | | |
| LS sequence number | | | |
| LS checksum | | Length | |
| 0 | Flags | 0 | Number of links |
| Link ID | | | |
| Link data | | | |
| Link type | Num_TOS | Metric | |
| Optional TOS information | | | |
| More links | | | |

**OSPF Link State Advertisement**

- LSAge – Equivalent to time to live(TTL), LSA expires when the age reaches the maximum value ( difference is TTL counts down, LSAge counts Up)
- Type = 1 LSAs advertise the cost of a link between routers.
- [ Type= 2 are used to advertise the networks to which the advertising routers are connected
- Other Types are used to support additional hierarchy]
- Link state ID & Advertising router
  -identical in Type 1 LSA, it should be unique in routing domain
  - 32 bit identifier for a router that created this LSA
- LS sequence number – to detect old or duplicate LSAs
- LS Checksum – similar to the other checksum ,
  - used to verify data
  - covers all the fields except LSAge
  - no need to compute checksum everytimeLSAge is incremented
- Length – length in bytes of complete LSA
- TOS - Type of Service
- Link Type
  - Type 1 point to point
  - Type 2 Transient ( in between nodes )
  - Type 3Stub
  - Type 4 Virtual
- Metric - cost of link

# GLOBAL INTERNET

- Internetworking is a heterogeneous of networks with tens of thousands of networks connected to it.
- The Global Internet



The tree structure of the Internet in 1990

- As the internet grows it poses,

- large no of network numbers are used, routers have to maintain a large routing table

Global internet is not just a random interconnection of Ethernets, but instead it takes on a shape that reflects the fact that it interconnects many different organizations.

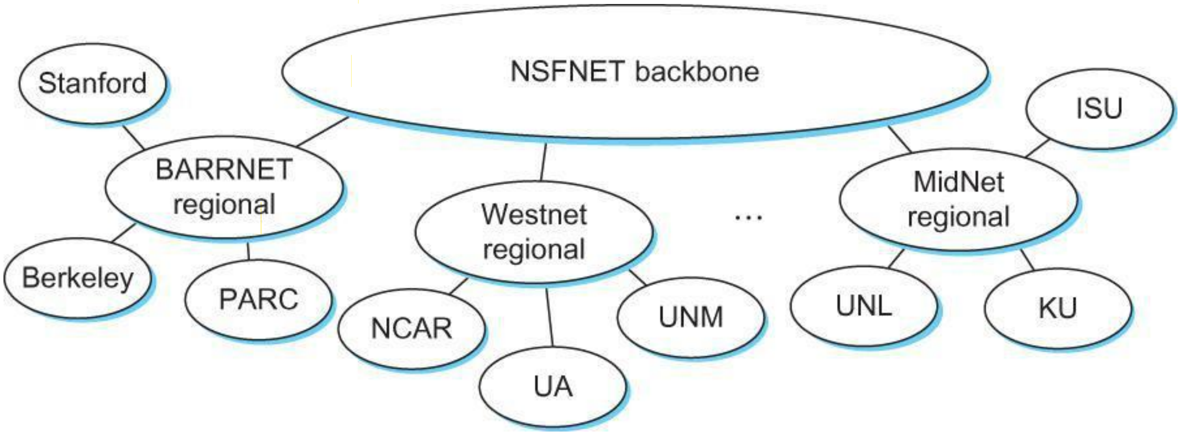


Figure : The tree structure of the Internet in 1990

Salient features of this topology is,

- It has end user sites that connect to service provider etwork
- Eg: user site – Stanford university
  Service provider network - e.g., BARRNET was a provider network that served sites in that area

Many providers served a limited geographic region and known as RegionalNetworks.

These Regional networks were in turn connected by Nation Wide Backbone –was funded by **NSF (National Science Foundation). Therefore, its called as NSFNET backbone.**

This has some significant consequence in routing.
Eg: AS decide the best routing protocol used in their network.

**As the internet grows it poses**

- large no of network numbers are used
- routers have to maintain a large routing table
- large amount of IP address space are wasted

**To tackle the 2 issues of scalability we use,**

- Supernetting or classless routing
- Subnetting

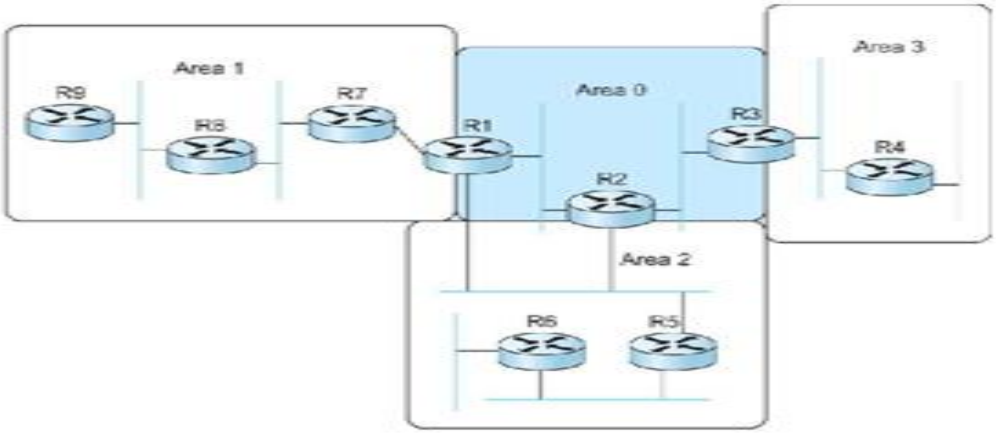
**ROUTING AREAS:**

As a first example of using hierarchy to scale up the routing system, we'll examine how link-state routing protocols (such as OSPF and IS-IS) can be used to partition a routing domain into
**subdomains called *areas*.**

By adding this extra level of hierarchy, we enable single domains to grow larger without overburdening the routing protocols or resorting to the more complex interdomain routing protocols

Area is a set of routers that are administratively configured to exchange link-state information with each other.
There is one special area—the backbone area, also known as area 0.

**An example of a routing domain divided into areas is shown in Figure**



- ✓ Routers R1, R2, and R3 are members of the backbone area. They are also members of at least one nonbackbone area.
- ✓ R1 is actually a member of both area 1 and area 2.
- ✓ A router that is a member of both the backbone area and a nonbackbone area is an area border router (ABR). The routers that are at the edge of an AS, which are referred to as AS border routers

**Routing within a single area**

- ✓ All the routers in the area send link-state advertisements to each other and thus develop a complete, consistent map of the area

- ✓ The link-state advertisements of routers that are not area border routers do not leave the area in which they originated.

- ✓ This has the effect of making the flooding androute calculation processes considerably more scalable.

- ✓ For example, router R4 in area 3 will never see a link-state advertisement from router R8 in area 1. As a consequence, it will know nothing about the detailed topology of areas other than its own.

**How does a router in one area determine the right next hop for a packet destined to a network in another area?**

The answer to this becomes clear if we imagine the path of a packet that has to travel from one nonbackbone area to another as being split into three parts.

1. First, it travels from its source network to the backbone area
2. then it crosses the backbone,
3. then it travels from the backbone to the destination network

To make this work, the area border routers summarize routing information that they have learned from one area and make it available in their advertisements to other areas.

For example, R1 receives link-state advertisements from all the routers in area 1 and can thus determine the cost of reaching any network in area 1.

When R1 sends link-state advertisements into area 0, it advertises the costs of reaching the networks in area 1 much as if all those networks were directly connected to R1.

This enables all the area 0 routers to learn the cost to reach all networks in area 1. The area border routers (ABR) then summarize this information and advertise it into the non-backbone areas. Thus, all routers learn how to reach all networks in the domain.

**In the case of area 2, there are two ABRs and that routers in area 2 will thus have to make a choice as to which one they use to reach the backbone.**

**Scalability & optimality**

When dividing a domain into areas, the network administrator makes a tradeoff between scalability and optimality of routing.

The use of areas forces all packets traveling from one area to another to go via the back- bone area, even if a shorter path might have been available.

For example, even if R4 and R5 were directly connected, packets would not flow between them because they are in different non-backbone areas.

It turns out that the need for scalability is often more important than the need to use the absolute shortest path.

Finally, we note that there is a trick by which network administrators can more flexibly decide which routers go in area 0. This trick uses the idea of a *virtual link* between routers.Such a virtual link is obtained by configuring a router that is not directly connected to area 0 to exchange backbone routing information with a router that is.

For example, a virtual link could be configured from R8 to R1, thus making R8 part of the backbone. R8 would now participate in link-state advertisement flooding with the other routers in area 0.

The cost of the virtual link from R8 to R1 is determined by the exchange of routing information that takes place in area 1. This technique can help to improve the optimality of routing

## INTER-DOMAIN ROUTING

- Inter domain routing is used for complex network.
- Internet is organized as autonomous systems (AS) each of which is under the control of a single administrative entity.
- Autonomous System (AS)
  - corresponds to an administrative domain
  - examples: University, company, backbone network

A corporation's internal network might be a single AS, as may the network of a single Internet service provider.

**Challenges in Inter-Domain Routing**

Perhaps the most important challenge of inter-domain routing today is the need for each AS to determine its own routing*policies*. A key design goal of inter-domain routing is that policies and much more complex ones, should be supported by the inter-domain routing system. We are concerned with reachability than optimality.

- Finding path anywhere close to optimal is considered to be a great achievement.

- Scalability: An Internet backbone router must be able to forward any packet destined anywhere in the Internet

  - ✓ Having a routing table that will provide a match for any valid IP address

- Autonomous nature of the domains
  - ✓ It is impossible to calculate meaningful path costs for a path that crosses multiple ASs
  - ✓ A cost of 1000 across one provider might imply a great path but it might mean an unacceptable bad one from another provid

- Issues of trust
  - ✓ Provider A might be unwilling to believe certain advertisements from provider B

**There are two Inter-domain Routing Protocols**

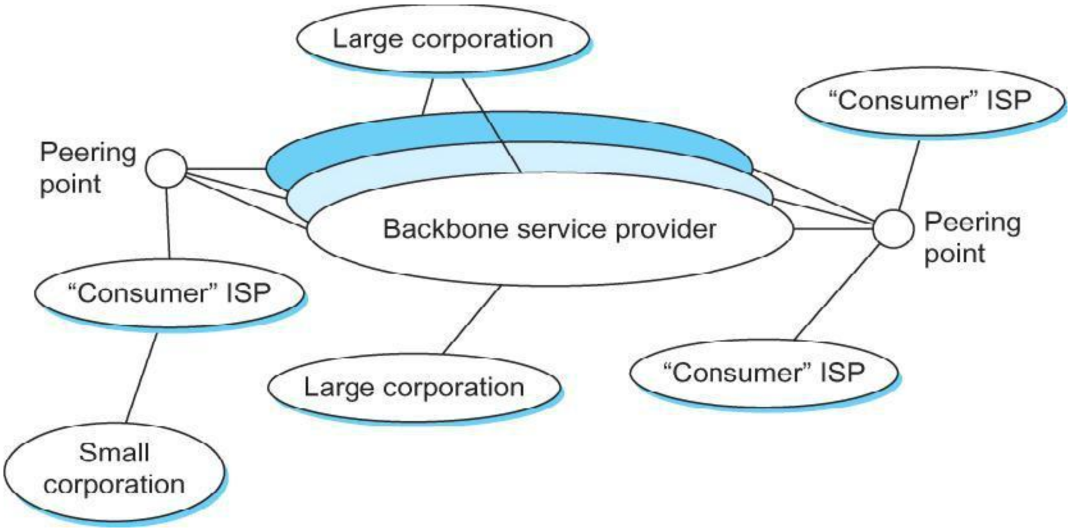
1. Exterior Gateway Protocol (EGP)
2. Border Gateway Protocol (BGP)

- **Exterior Gateway Protocol (EGP)**
  - ✓ Forced a tree-like topology onto the Internet
  - ✓ Did not allow for the topology to become general
    - ➢ Tree like structure: there is a single backbone and autonomous systems are connected only as parents and children and not as peers
- **Border Gateway Protocol (BGP)**
  - ✓ BGP version 4 is often regarded as one of the more complex parts of the internet.
  - ✓ BGP makes virtually no assumptions about how autonomous systems are interconnected—they form an arbitrary graph.

**BORDER GATEWAY PROTOCOL:**

- ➢ BGP is used for routing interconnected set of Autonomous Systems(ASs)
- ➢ It Assumes that the Internet is an arbitrarily interconnected set of ASs.

Figure shows today's multibackbone Internet. today's Internet consists of a richly interconnected set of networks, mostly operated by private companies (ISPs) rather than governments. Many Internet Service Providers (ISPs) exist mainly to provide service to "consumers" (i.e., individuals with computers in their homes), while others offer something more like the old backbone service, interconnecting other providers and sometimes larger corporations. Often, many providers arrange to interconnect with each other at a single *peering point*.



It defines **two types of traffic**

1) *local traffic*

- as traffic that originates at or terminates on nodes within an AS

2) *transit traffic*

- as traffic that passes through an AS.

**Three types of AS are,**

- ✓ *Stub AS*: an AS that has only a single connection to one other AS;such an AS will only carry local traffic (*small corporation in the figure of the previous page*).
- ✓ *Multihomed AS*: an AS that has connections to more than one otherAS, but refuses to carry transit traffic (*large corporation at the top inthe figure of the previous page*).
- ✓ *Transit AS*: an AS that has connections to more than one other AS, andis designed to carry both transit and local traffic (*backbone providers inthe figure of the previous page*).

The *goal* of Inter-domain routing is

- ➢ To find any path to the intended destination that is loop free.
- ➢ Paths must be compliant with policies of various ASs along the paths.

## Basics of BGP

1. Each AS has one or more border routers through which packets enter and leave the AS.
2. A Border Router is simple an IP router that is charged with the task of forwarding packet between autonomous systems.

*Each AS has*

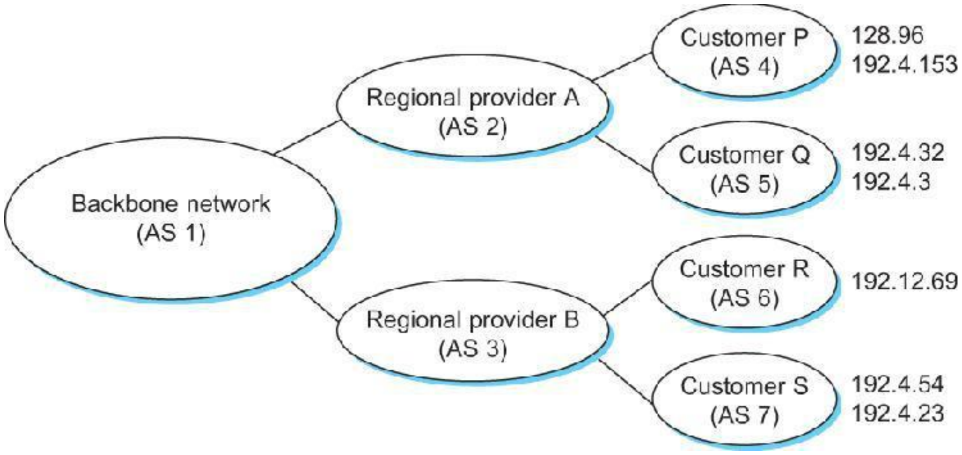
- One BGP *speaker* that advertises:
  - ✓ local networks
  - ✓ other reachable networks (transit AS only)
  - ✓ gives *path* information

- In addition to the BGP speakers, the AS has one or more border "gateways" which need not be the same as the speakers
- The border gateways are the routers through which packets enter and leave the AS
- BGP does not belong to either of the two main classes of routing protocols (distance vectors and link-state protocols).

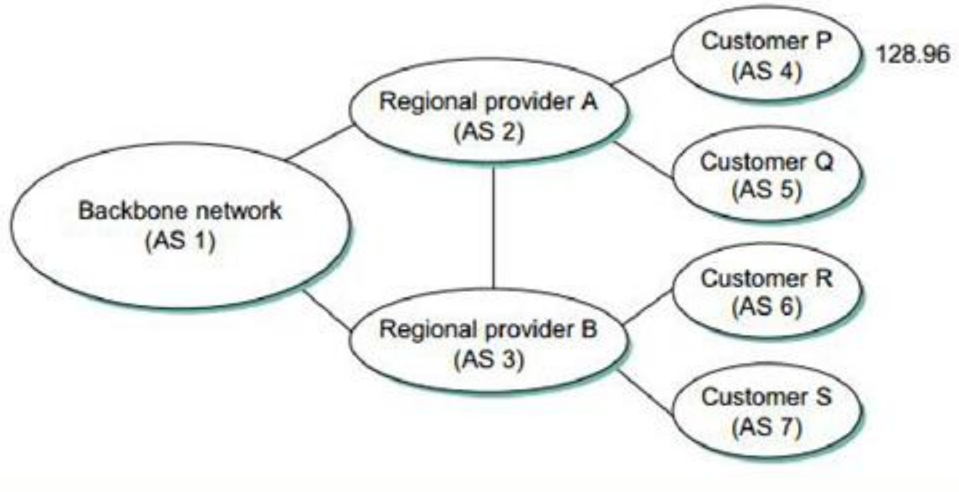
- BGP advertises *complete paths* as an enumerated lists of ASs to reach a particular network. It is sometimes called a *path-vector* protocol for this reason.

  consider the very simple example network in Figure. Assume that the providers are transit networks, while the customer networks are stubs.

An Example of a network running BGP



- A BGP speaker for the AS of provider A (AS 2) advertises reachability to P and Q
  - ✓ Network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS 2.
- Speaker for backbone network then advertises
  - ✓ Networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path <AS 1, AS 2>.
- Speaker can also cancel previously advertised paths
- An important job of BGP is to prevent the establishment of looping paths.



In figure only in the addition of an extra link between AS 2 and AS 3, but the effect now is that the graph of autonomous systems has a loop in it.

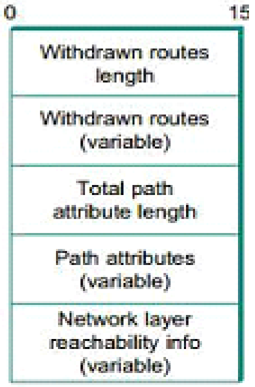
Suppose AS 1 learns that it can reach network 128.96 through AS 2, so it advertises this fact to AS 3, who in turn advertises it back to AS 2. In the absence of any loop prevention mechanism, AS 2 could now decide that AS 3 was the preferred route for packets destined for 128.96.

If AS 2 starts sending packets addressed to 128.96 to AS 3, AS 3 would send them to AS 1; AS 1 would send them back to AS 2; and they would loop forever. This is prevented by carrying the complete AS path in the routing messages.

In this case, the advertisement for a path to 128.96 received by AS 2 from AS 3 would contain an AS path of <AS 3, AS 1, AS 2, AS 4>. AS 2 sees itself in this path, and thus concludes that this is not a useful path for it to use.

In order for this loop prevention technique to work, the AS numbers carried in BGP clearly need to be unique. For example, AS 2 can only recognize itself in the AS path in the above example if no other AS identifies itself in the same way. AS numbers have until recently been 16-bit numbers, and they are assigned by a central authority to assure uniqueness.

### BGP-4 Update Packet Format


0
15
Withdrawn routes length
Withdrawn routes (variable)
Total path attribute length
Path attributes (variable)
Network layer reachability info (variable)

Given that links fail and policies change, BGP speakers need to be able to cancel previously advertised paths.

This is done with a form of negative advertisement known as a *withdrawn route*. Both positive and negative reachability information are carried in a BGP update message, the format of which is shown in Figure.

### Common AS Relationships and Policies

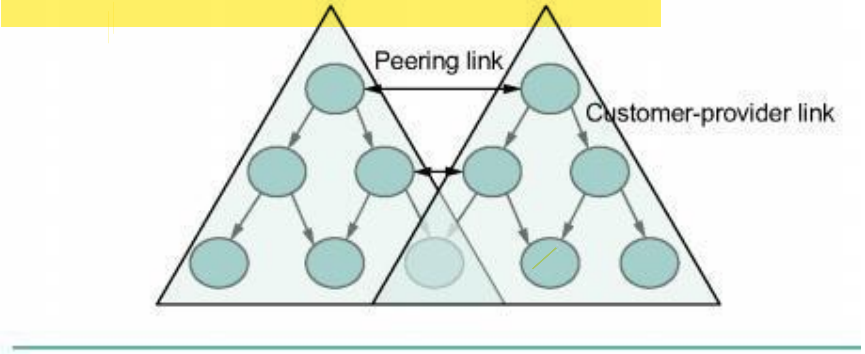


FIGURE 4.8 Common AS relationships.

**Three common relationships and the policies are,**

**1. Provider-Customer**

Providers are in business of connecting their customers to rest of internet. A customer might be s smaller ISP. so the common policy is to advertise all the routes I know to my customer, and advertise routes I learn from my customer to everyone.

**2. Customer-Provider**

Advertise my own prefixes and routes learned from my provider to my customers to my provider, advertises routes learned from my providers to my customers , but don't advertise routes learned from one provider to another provider.

**3. Peer**

Third option is a symmetrical peering between autonomous systems. policy here is to advertise routes learned from my customers to my peer, advertise routes learned from my peer to my customers, but don't advertise routes from my peer to any provider or vice versa.

## Integrating Inter-domain and Intra-domain Routing

Consider, for example, the border router of a provider AS that connects to a customer AS. That router could learn that the network prefix 192.4.54/24 is located inside the customer AS, either through BGP or because the information is con- figured into the border router. It could inject a route to that prefix into the routing protocol running inside the provider AS. This would be an advertisement of the sort, “I have a link to 192.4.54/24 of cost X.”

This would cause other routers in the provider AS to learn that this border router is the place to send packets destined for that prefix.

The routers in a backbone network use a variant of BGP called ***interior BGP*(iBGP)**to effectively redistribute the informationthat is learned by the BGP speakers at the edges of the AS to all the other routers in the AS. ( The other variant of BGP, discussed above, runs between autonomous systems and is called ***exterior BGP*, or eBGP**).

iBGP enables any router in the AS to learn the best border router to use when sending a packet to any address. At the same time, each router in the AS keeps track of how to get to each border router using a conventional intradomain protocol with no injected information. By combining these two sets of information, each router in the AS is able to determine the appropriate next hop for all prefixes.

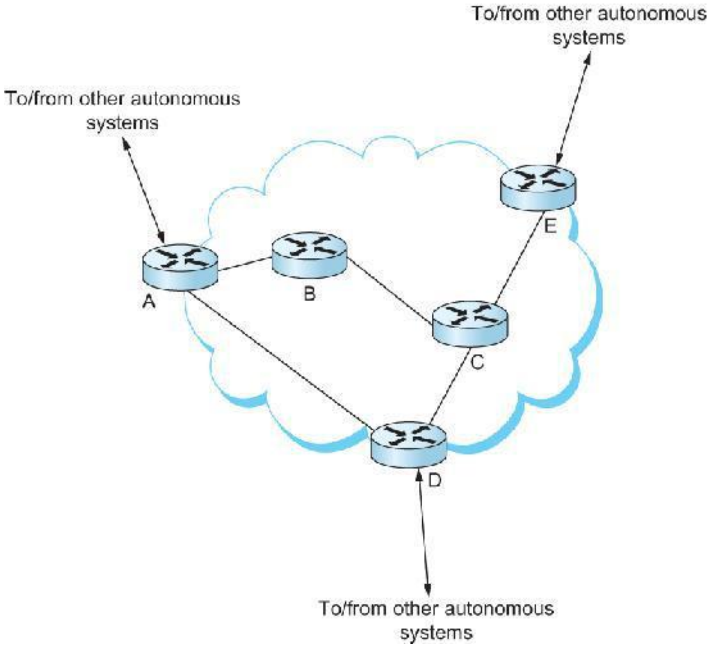

To/from other autonomous systems
To/from other autonomous systems
E
A
B
C
D
To/from other autonomous systems

FIGURE 4.9 Example of interdomain and intradomain routing. All routers run iBGP and an intradomain routing protocol. Border routers A, D, and E also run eBGP to other autonomous systems.

Consider the simple example network, rep- resenting a single AS, in Figure 4.9. The three border routers, A, D, and E, speak eBGP to other autonomous systems and learn how to reach variousprefixes. These three border routers communicate with other and with the interior routers B and C by building a mesh of iBGP sessions among all the routers in the AS. Let's now focus in on how router B builds up its complete view of how to forward packets to any prefix.

Look at the table at the top left of Figure 4.10 which shows the information that router B learns from its iBGP sessions. It learns that some prefixes are best reached via router A, some via D, and some via E. At the same time, all the routers in the AS are also running some intra-domain routing protocol such as Routing Information Protocol (RIP) or Open Shortest Path First (OSPF).

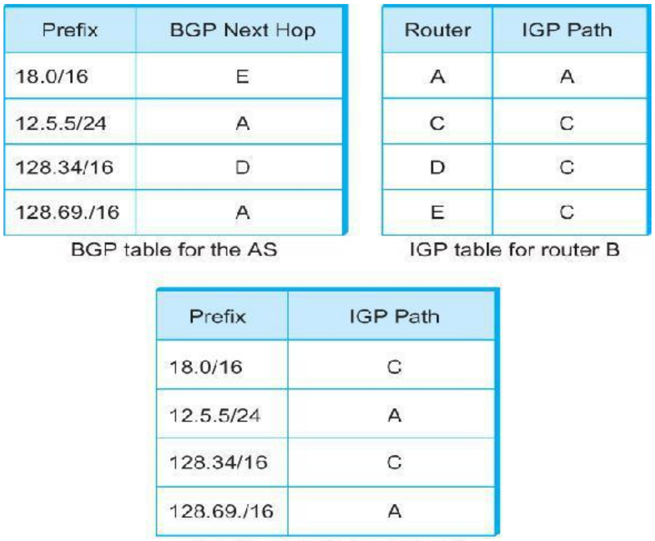
(A generic term for intra-domain protocols is an interior gateway protocol, or IGP.) From this completely separate protocol, B learns how to reach other nodes *inside* the domain, as shown in the top right table.

For example, to reach router E, B needs to send packets toward router C. Finally, in the bottom table, B puts the whole picture together, combining the information about external prefixes learned from iBGP with the infor- mation about interior routes to the border routers learned from the IGP.

Thus, if a prefix like 18.0/16 is reachable via border router E, and the best interior path to E is via C, then it follows that any packet destined for 18.0/16 should be forwarded toward C. In this way, any router in the AS can build up a complete routing table for any prefix that is reachable via some border router of the AS.

All routers run iBGP and an intradomain routing protocol. Border routers (A, D, E) also run eBGP to other ASs

BGP routing table, IGP routing table, and combined table at router B is,



| Prefix | BGP Next Hop |
|---|---|
| 18.0/16 | E |
| 12.5.5/24 | A |
| 128.34/16 | D |
| 128.69./16 | A |

BGP table for the AS

| Router | IGP Path |
|---|---|
| A | A |
| C | C |
| D | C |
| E | C |

IGP table for router B

| Prefix | IGP Path |
|---|---|
| 18.0/16 | C |
| 12.5.5/24 | A |
| 128.34/16 | C |
| 128.69./16 | A |

Combined table for router B

FIGURE 4.10 BGP routing table, IGP routing table, and combined table at router B.

# IPV6

**The motivation for new version of IP is**

- To deal with scaling problem
- To achieve 100 % address utilization efficiency

## Historical perspective

- IEFT (Internet Engg Task Force), which is responsible for specification of standards and protocols looked into the problem in 1991
- Since the IP address is carried in the header of every IP packet, increasing the size of the address dictates a change in the packet header.
- The effort to define a new version of IP was known as IP NextGeneration, or IPng.
- An official IP version number was assigned, so IPng is now known as IPv6

In addition to deal with scalable routing and addressing, IPv6 should also

- support for real-time services
- supports multicast
- security support
- auto configuration (i.e., the ability of hosts to automatically configure themselves with such information as their own IP address and domain name)
- End-to-end fragmentation
- Enhanced routing functionality, including support for mobile hosts.

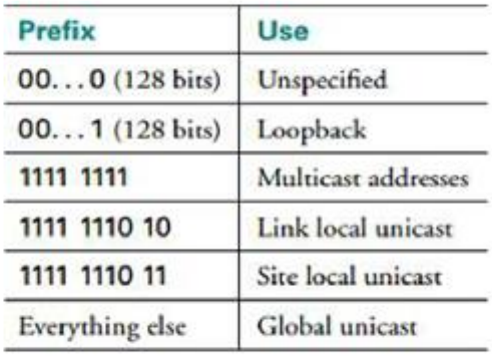
## Addresses and Routing

- **IPv6 provides a 128-bit address space.**
- IPv6 can address $3.4\times10_{38}$ nodes,
- IPv6 address space is predicted to provide over 1500 addresses per square foot of the earth's surface

## Address Space Allocation

- **IPv6 also follows CIDR like IPv4, so IPv6 addresses are classless**
- The address prefix assignments for IPv6 is as follows,

At the time of writing, IPv6 unicast addresses are being allocated from the block that begins 001, with the remaining address space—about 87%—being reserved for future use.



| Prefix | Use |
|---|---|
| 00...0 (128 bits) | Unspecified |
| 00...1 (128 bits) | Loopback |
| 1111 1111 | Multicast addresses |
| 1111 1110 10 | Link local unicast |
| 1111 1110 11 | Site local unicast |
| Everything else | Global unicast |

**Fig : Address Prefix Assignments for IPv6**

**Multicast address**

- Similar to class D address in IPV4
- The multicast address space is for multicast,
- Start with a byte of all 1s.

**Link local unicast**

- Enable a host to construct an address that will work on the network to which it is connected, without being concerned about the global uniqueness of the address.
- This may be useful for autoconfiguration

**Site local unicast**

- Intended to allow valid address to be constructed on site.
- (e.g., a private corporate network) that is not connected to the larger Internet;

**global unicast address**

- Some important special types of addresses.
- Two special address types have uses in the IPv4-to-IPv6 transition.
  1. IPv4 compatible IPv6 address
  2. IPv4 mapped IPv6 address

**IPv4 compatible IPv6 address**

- One type, the *IPv4-compatible IPv6 address*, is used for devices that are compatible with both IPv4 and IPv6;
- Begins with 96 bits zeros then followed by 32 bits IPv4 address

**IPv4 mapped IPv6 address**

- A node that is only capable of understanding IPv4 can be assignedan IPv4-mapped IPv6 address by prefixing the 32-bit IPv4 address with 2 bytes of all 1s and then zero-extending the result to 128 bits.
- Begins with 80 bits zeros, followed by 16 bits of ones and 32 bits IPv4 address

## Address Notation

The standard representation is

x:x:x:x:x:x:x:x

where each —x is a hexadecimal representation of a 16-bit piece of the

## Example IPV6 Address

47CD:1234:4422:ACO2:0022:1234:A456:0124

**If we have an address with a large number of contiguous 0s can be written more compactly by omitting all the 0 fields.**

**Example1**

47CD:0000:0000:0000:0000:0000:A456:0124

could be written as 47CD::A456:0124

**Example 2**

3FFE:085B:1F1F:0000:0000:0000:00A9:1234

could be written as 3FFE:85B:1F1F::A9:1234

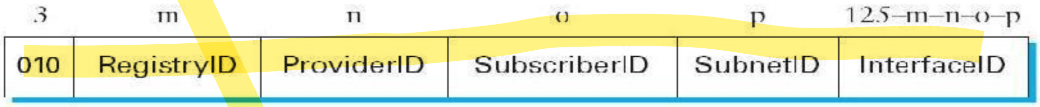
- 8 groups of 16-bit hexadecimal numbers separated by “:” & Leading zeros can be removed. **::** = all zeros in one or more group of 16-bit hexadecimal numbers
- Clearly, this formof shorthand can only be used for one set of contiguous 0s in an address to avoid ambiguity.

The two types of IPv6 addresses that contain an embedded IPv4 address have their own special notation that makes extraction of the IPv4 address easier.

For example, the IPv4 -mapped IPv6 address of a host whose IPv4 address was 128.96.33.81 could be written as:: FFFF:128.96.33.81

That is, the last 32 bits are written in IPv4 notation, rather than as a pair of hexadecimal numbers separated by a colon.Note that the double colon at the front indicates the leading 0s.

**IPv6 provider-based unicast address is as follows:**

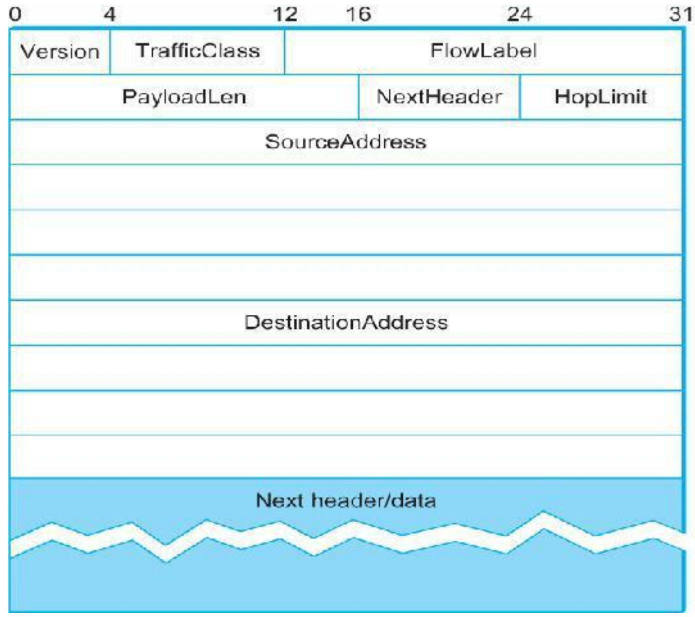


| 3 | m | n | o | p | 125–m–n–o–p |
|---|---|---|---|---|---|
| 010 | RegistryID | ProviderID | SubscriberID | SubnetID | InterfaceID |

The Registry ID might be an identifier assigned to a European address registry, with different IDs assigned to other continents or countries.

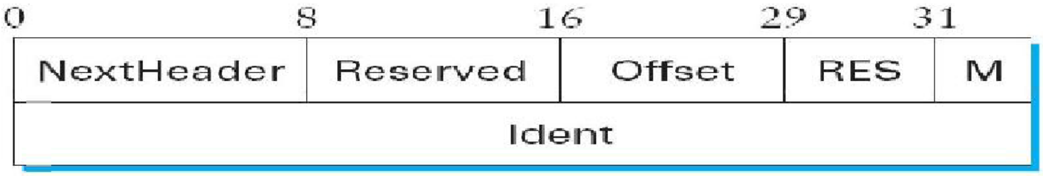
## Packet Format

The IPV6 packet header format is as follows



- ✓ Version - set to 6 for IPv6.
- ✓ TrafficClass and FlowLabel– deals with quality of service issues.
- ✓ PayloadLen -Length of the packet excluding the IPv6 header,measured in bytes.
- ✓ NextHeader – if special headers follow IP header, it indicates option.
- ✓ If there are no special headers, it indicates the highlevel protocols running over IP. Eg: TCP and UDP
- ✓ HopLimit - field is simply the TTL of IPv4.

Each Option has its own type of extension header. The IPv6 fragmentationextension header is as follows,



Assuming it is the only extension header present, then the NextHeader field of the IPv6 header would contain the value 44, which is the value assigned to indicate the fragmentation header

- The NextHeader field of the fragmentation header itself contains a value describing the header that follows it. Again, assuming no other extension headers are present, then the next header might be the TCP header, which results in NextHeader containingthe value 6, just as the Protocol field would in IPv4.
- If the fragmentation header were followed by, say, an authentication header, then the fragmentation header's NextHeaderfield would contain the value 51.

# MULTICASTING

- IP multicast is a method of sending Internet Protocol (IP) datagrams to a group of interested receivers in a single transmission.
- It is often employed for streaming media applications on the Internet and private networks. The method is the IP-specific version of the general concept of multicast networking.

## Overview

- One-to-many
  - ✓ Radio station broadcast
  - ✓ Transmitting news, stock-price
  - ✓ Software updates to multiple hosts
- Many-to-many
  - ✓ Multimedia teleconferencing
  - ✓ Online multi-player games
  - ✓ Distributed simulations

## Without support for multicast

- ✓ A source needs to send a separate packet with the identical data to each member of the group
  - ➢ This redundancy consumes more bandwidth
  - ➢ Redundant traffic is not evenly distributed, concentrated near the sending host
- ✓ Source needs to keep track of the IP address of each member in the group
  - ➢ Group may be dynamic

- **To support many-to-many and one-to-many IP provides an IP-level multicast**

- Basic IP multicast model is many-to-many based on multicast groups
  - ✓ Each group has its own IP multicast address
  - ✓ Hosts that are members of a group receive copies of any packets sent to that group's multicast address
  - ✓ A host can be in multiple groups
  - ✓ A host can join and leave groups
- Using IP multicast to send the identical packet to each member of the group
  - ✓ A host sends a single copy of the packet addressed to the group's multicast address
  - ✓ The sending host does not need to know the individual unicast IP address of each member
  - ✓ Sending host does not send multiple copies of the packet
- IP's original many-to-many multicast has been supplemented with support for a form of one-to-many multicast

**One-to-many multicast**

**Source specific multicast (SSM)**

- ✓ Single sender, multiple receivers

- ✓ A receiving host specifies both a multicast group and a specific sending host
  E.g. radio stations, TV stations

**Many-to-many model**

**Any source multicast (ASM)**

- ✓ Some or all nodes can become sender
  E.g. teleconferencing, online video games

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

- The router has the responsibility for making multicast behave correctly with regard to the host

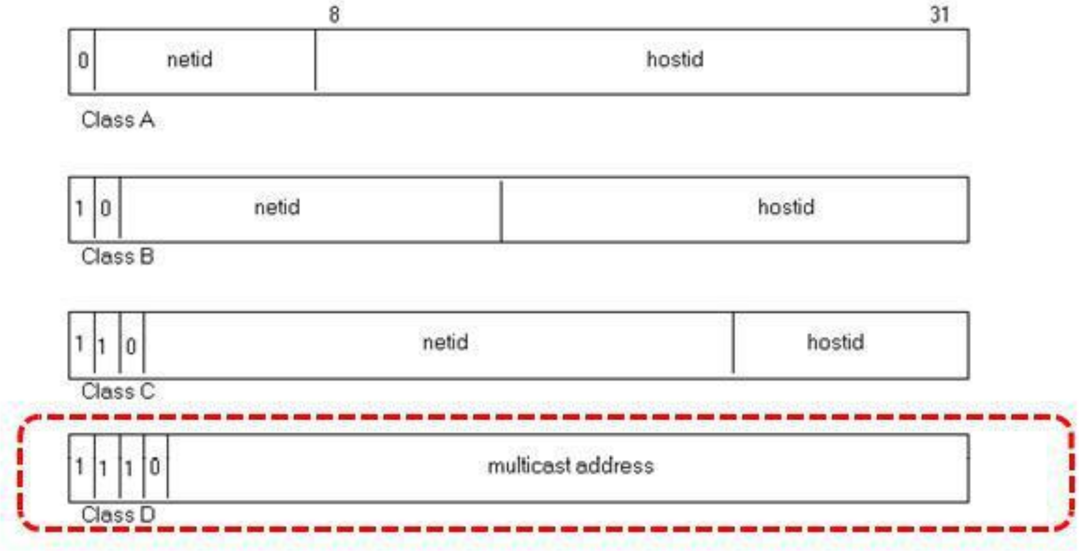
**Multicast address**

- ➢ A **multicast address** is a logical identifier for a group of hosts in a computer network, that are available to process datagrams or frames intended to be multicast for a designatednetwork service.
- ➢ IPv6 has a portion of address space 11111111 reserved for multicast group addresses.
- ➢ When a host on the Ethernet joins an IP multicast group, it configures its Ethernet interface to receive packets with corresponding Ethernet multicast address.
- ➢ This causes the receiving host to receive not only the multicast traffic but also traffic sent to any of the other multicast groups. Thus, IP header of any receiving host examine IP header of any multicast packet to determine whether the packet belongs to desire group.

**Multicasting with IPv4**

- ■ Multicast addressing can be used in the Link Layer (Layer 2 in the OSI model), such as Ethernet multicast, and at the InternetLayer (Layer 3 for OSI) for Internet Protocol Version 4 (IPv4) or Version 6 (IPv6) multicast.
- ■ IPv4 multicast addresses are defined by the leading address bits of *1110*, originating from the classful network design of the early Internet when this group of addresses was designated as *Class D*.

- The Classless Inter-Domain Routing (CIDR) prefix of this group is 224.0.0.0/4
- The group includes the addresses from 224.0.0.0 to 239.255.255.255.



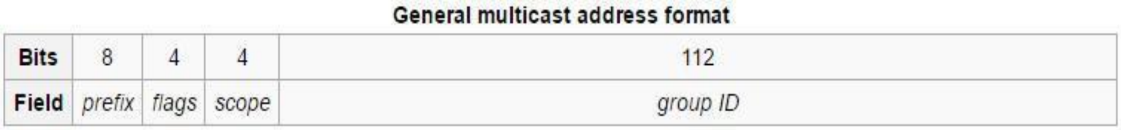
**Example list of notable well-known IPv4 addresses that are reserved for IP multicasting and that are registered with the Internet Assigned Numbers Authority(IANA).**

| | |
|---|---|
| 224.0.0.1 | The *All Hosts* multicast group addresses all hosts on the same network segment. |
| 224.0.0.2 | The *All Routers* multicast group addresses all routers on the same network segment. |
| 224.0.0.4 | This address is used in the Distance Vector Multicast RoutingProtocol (DVMRP) to address multicast routers. |
| 224.0.0. | The Open Shortest Path First (OSPF) *All OSPF Routers* address is used to send Hello packets to all OSPF routers on a network segment. |

. .

## Multicasting with IPv6

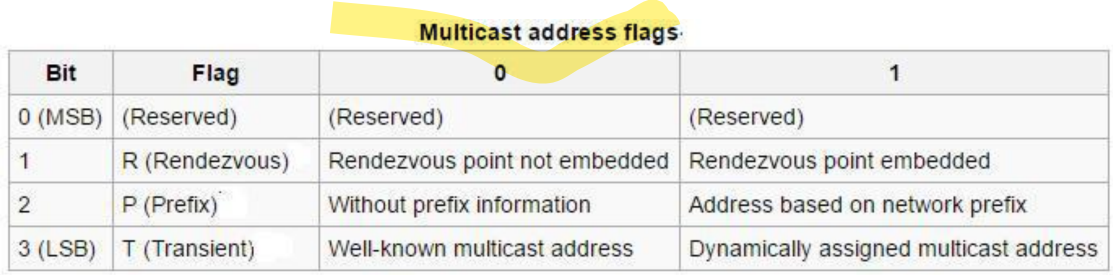
- Multicast addresses in IPv6 have the prefix ff00::/8. IPv6 multicast addresses are generally formed from four bit groups, illustrated as follows:



General multicast address format

| Bits | 8 | 4 | 4 | 112 |
|---|---|---|---|---|
| Field | *prefix* | *flags* | *scope* | *group ID* |

- The *prefix* holds the binary value 11111111 for any multicast address. Currently, 3 of the 4 flag bits in the *flags* field are defined; the most-significant flag bit is reserved for future use. The other three flags are known as *R*, *P* and *T*.



Multicast address flags

| Bit | Flag | 0 | 1 |
|---|---|---|---|
| 0 (MSB) | (Reserved) | (Reserved) | (Reserved) |
| 1 | R (Rendezvous) | Rendezvous point not embedded | Rendezvous point embedded |
| 2 | P (Prefix) | Without prefix information | Address based on network prefix |
| 3 (LSB) | T (Transient) | Well-known multicast address | Dynamically assigned multicast address |

## Some Well-known IPv6 multicast addresses

| Address | Description |
|---|---|
| ff02::1 | All nodes on the local network segment |
| ff02::2 | All routers on the local network segment |
| ff02::5 | OSPFv3 All SPF routers |
| ff02::6 | OSPFv3 All DR routers |
| ff02::9 | RIP routers |
| ff02::d | PIM routers |
| ff0x::114 | Used for experiments |

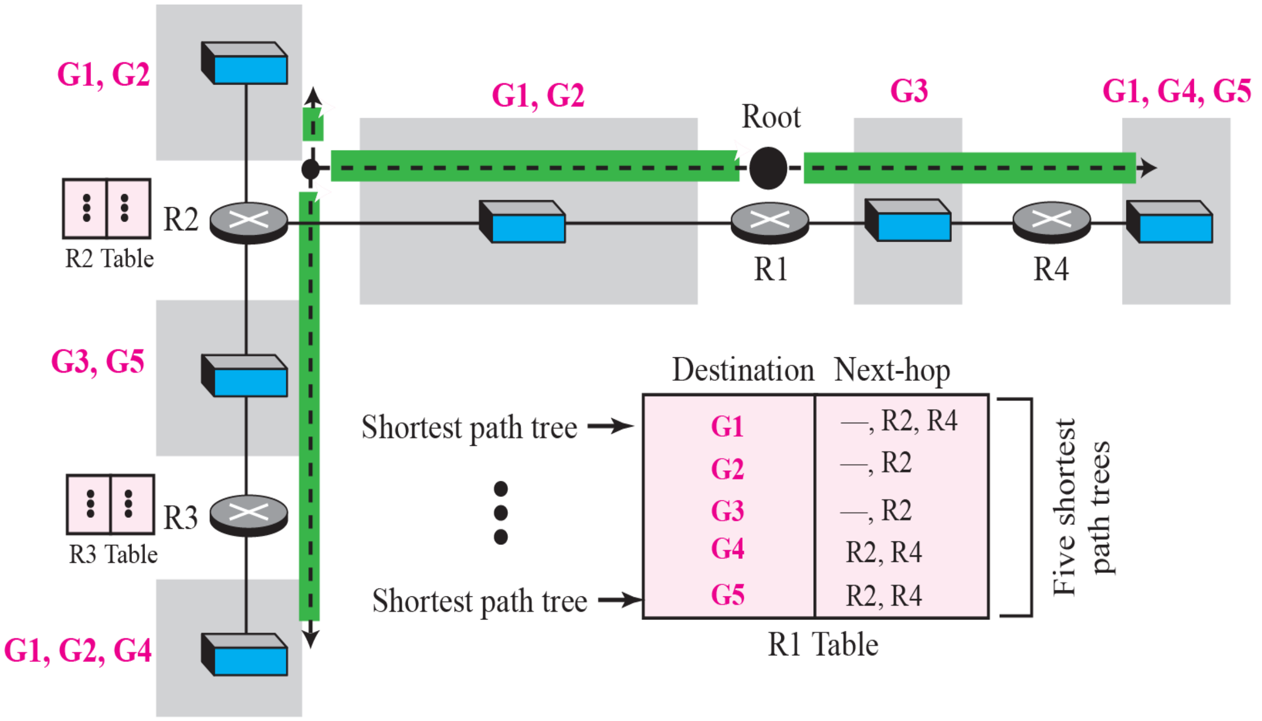
# MULTICAST ROUTING

- A router's unicast forwarding tables indicate for any IP address, which link to use to forward the unicast packet
- To support multicast, a router must additionally have multicastforwarding tables that indicate, based on multicast address, which links to use to forward the multicast packet
- Unicast forwarding tables collectively specify a set of paths
- Multicast forwarding tables collectively specify a set of trees

### ✓ Multicast distribution trees

- ■ *In multicast routing, each involved router needs to construct a shortest path tree for each group.*
- ■ In the source-based tree approach, each router needs to have one shortest path tree for each group and source.
- ■ In the group-shared tree approach, only the core router, which has a shortest path tree for each group, is involved in multicasting.

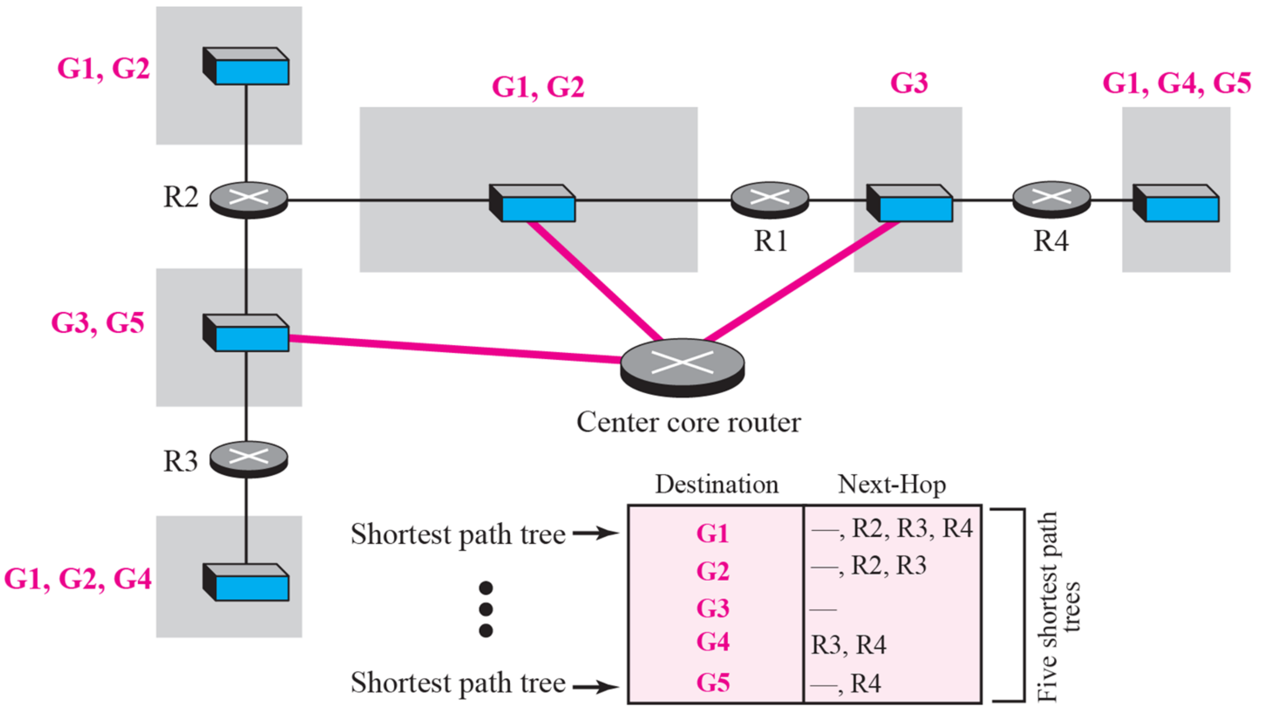
## SOURCE BASED APPROACH



| Destination | Next-hop |
|---|---|
| G1 | —, R2, R4 |
| G2 | —, R2 |
| G3 | —, R2 |
| G4 | R2, R4 |
| G5 | R2, R4 |

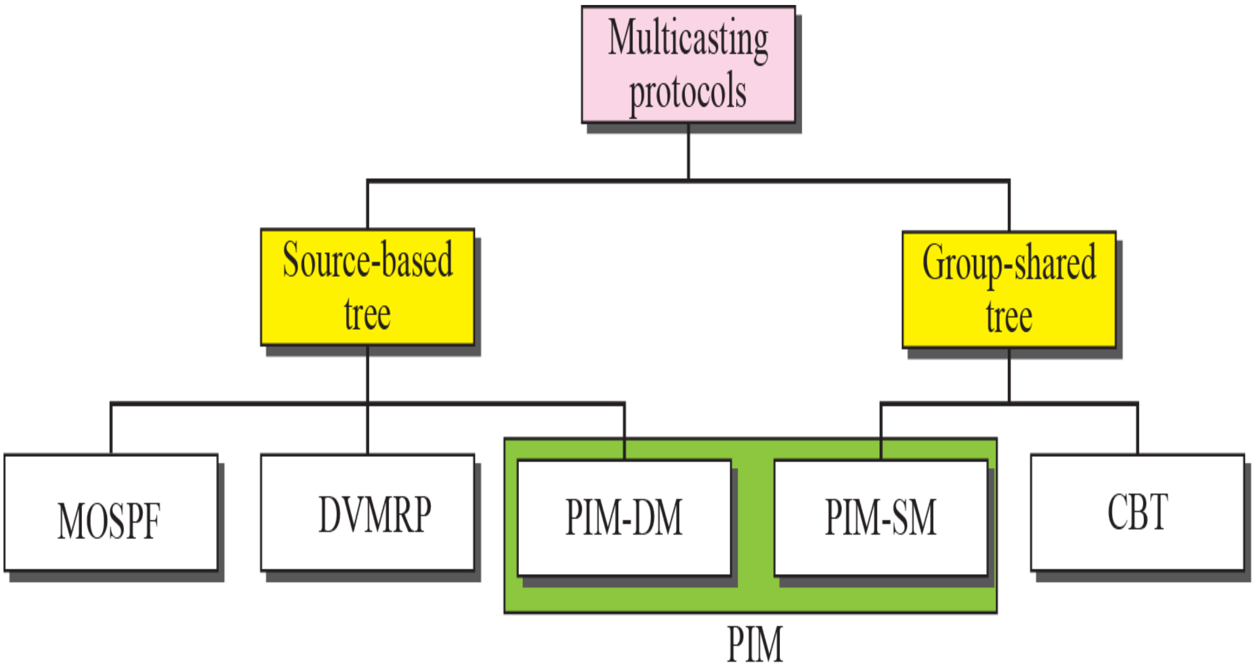
R1 Table

## GROUP SHARED TREE APPROACH:



| Destination | Next-Hop |
|---|---|
| G1 | —, R2, R3, R4 |
| G2 | —, R2, R3 |
| G3 | — |
| G4 | R3, R4 |
| G5 | —, R4 |

Core Router Table



## DISTANCE VECTOR MRP:

Using DVMRP, multicasting is a two stage process, they are

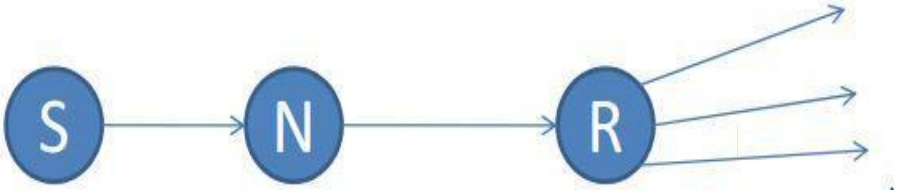
1. Flood: design a broadcast that can forward packets to all nodes on the Internet
2. Prune: refine broadcast to prune networks that have no nodes in multicast group

**Decision-making strategies**

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

**Reverse Path Forwarding (RPF):**

- Each router already knows that shortest path to source S goes through router N.
- When receive multicast packet from S, forward on all outgoing links (except the one on which the packet arrived), if packet arrived from N. flood to all links except to the link connected to S.



**Two shortcomings**

1. It truly floods the network.
   - ✓ Cannot avoid flooding to LANS with no multicast group participants
2. A packet is forwarded to a LAN by ALL routers connected to it.

**Two shortcomings**

3. It truly floods the network.
   - ✓ Cannot avoid flooding to LANS with no multicast group participants
4. A packet is forwarded to a LAN by ALL routers connected to it.

**Solution to shortcoming1: REVERSE PATH BROADCAST (RPB)**

Reverse Path?We are considering shortest path towards rootwhen making forwarding decisions– (unicast look at shortest path to destination).

Goal: **Prune networks that have no hosts in group G**
– Done in 2 steps.

**Step 1: Determine if LAN is a *leaf* with no members in G**
– **leaf** if parent is only router on the LAN
– determine if any hosts are members of G using IGMP (nodes in G periodically announce membership in network)

**Parent router decide to forward multicast packet to LAN based on membership announcements**

**Step 2: Propagate "no members of G here" information**

– LAN send Parent router a **set of groups for which thisnetwork is interested in receiving multicast packets**.

– This information is propagated from router to router

– Now each router know for each of its links, for what groups it should forward multicast packets.

**Solution to shortcoming2**

- **Eliminate duplicate broadcast packets**
- Allow only the designated router (parent router) forward packets to LAN
- Parent Router selection.
  - Router with shortest path to S
  - Use smallest address to break ties
- A router can learn if it is the parent for the LAN for a given source.

## PROTOCOL INDEPENDENT MULTICAST:

Protocol Independent Multicast (PIM) is developed

i. To solve scaling issues.

ii. To limit the traffic received by each group.

PIM is divided into two groups,

1. PIM – DM (Dense Mode)
2. PIM – SM (Sparse Mode)

**PIM – DM**

- PIM – Dense mode uses flood and prune algorithm (used in DVMRP).
- Suffers from scaling issues

**PIM - SM**

- Routers explicitly join and leave the group
  - Uses "Join" and "Leave" messages.
- PIM assigns a representative node called the "RendezvousPoint" (or RP)to each multicast group.
- RP's IP address is known to all the routers in a domain.
- PIM-SM defines a set of procedures by which all routers in a domain can agree to use RP for a given group.
- Multicast forwarding tree is built as a result of routers sending join messages to RP.

- PIM-SM use join message to build two kinds of trees
  1. Shared Tree: Used by all senders
  2. Source-Specific Tree: Used by only a specific sending host.

Under **normal course of operations**:

– A shared tree is built first.

– 1 or more source specific trees are constructed if there is enough traffic towarrant this.

PIM-SM first creates shared tree first followed by one or more source specific trees.

When a router sends a join message towards RP for a group G, it is sent as normal IP unicast transmission.

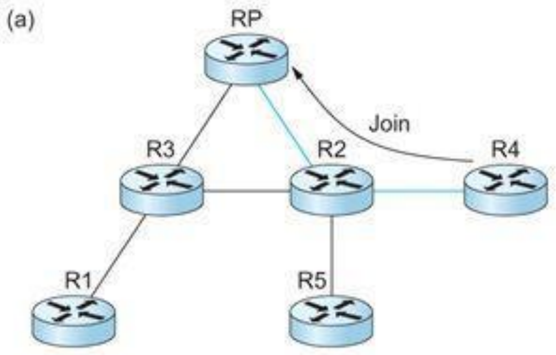


**Fig : (a) R4 sends Join to RP and joins shared tree**

Join message has to pass through a sequence of routers before reaching RP. Each router along the path look at join and create a forwarding table entry for shared tree. In the figure a) the Join message from R4 passes through R2 to reach RP.

As more routers send join messages towards RP, they form new branches to be added to tree as follows,

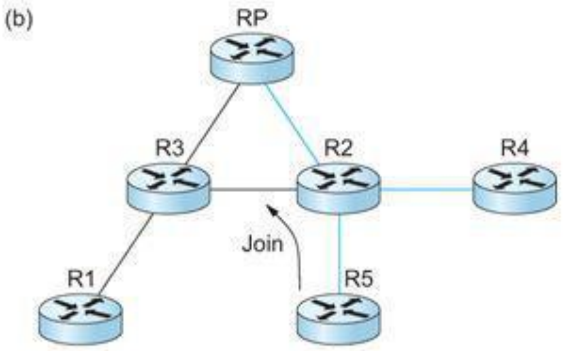


Fig : (b) R5 joins shared tree

If R1 wants to build a source-specific tree, RP sends a join message to R1.

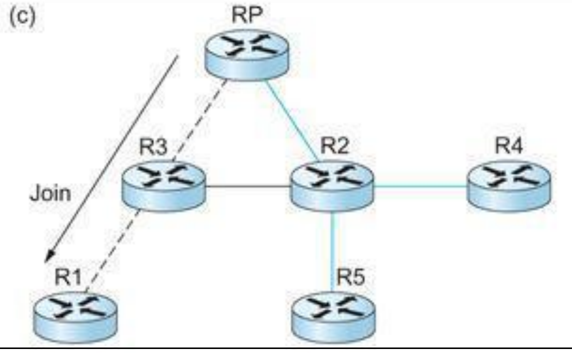


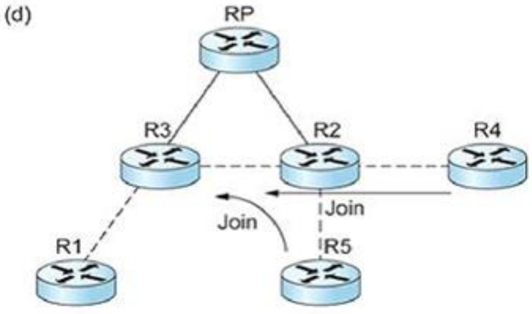
Fig: (c) RP builds source-specific tree to R1 by sending Join to R1

RP = Rendezvous point

—— Shared tree

- - - - Source-specific tree for source R1

Fig : (d) R4 and R5 can build a source – specific tree to R1 by sending joins to R1.



**Traditional multicast**

– A group address is a single IP address taken from a reserved range (224.0.0.0/4 for IPv4, FF00::/8 for IPv6)

- **Source-Specific Multicast (SSM) -**RFC4607

  – An SSM group, called a *channel*, is identified as (S,G)

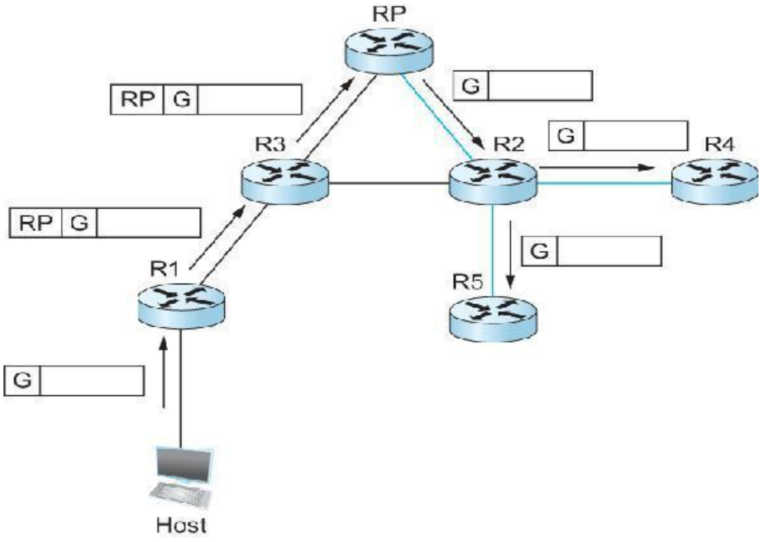
  - S: source address , G: group address
  - ASM multicast route written as (*,G)

  – reserved IPv4 address range 232.0.0.0/8 and the IPv6 range FF3x::/32

**Advantages of SSM**

1. Because an SSM channel is defined by **both a sourceand a group address**,group addresses can be re-usedby multiple sources while keeping**channels unique**.
   - E.g. SSM channel (192.168.45.7, **232.7.8.9**) is different than (192.168.3.104, **232.7.8.9**)
   - hosts subscribed to one will not receive traffic from the other
   - hosts will only receive traffic from explicitly requested sources
2. SSM does not rely on the designation of a rendezvous point (RP)to establish a multicast tree
   - **Why?**
   - because the **source of an SSM channel is alwaysknown in advance**, multicast trees are efficientlybuilt from channel hosts toward the source (based on the **unicast routing topology**) without the need for an RP.

The delivery of packet along a shared tree is as follows,



- **R1** **tunnels** **the packet to the RP** (as R1 has no state in multicastgroup with RP)

  -R1 encapsulate multicast packet inside a PIM registermessage and send to RP

- **RP sends to R2** which forwards it along the shared tree to **R4 and R5**.