4.1 Routing in the internet and CIDR
4.2 Multicasting
4.3 Unidirectional Link Routing
4.4 RIPng
4.5 OSPF for IPv6
4.6 PIM-SM and DVMRP for IPv6

Routing is the process of selecting a path for traffic in a network, or between or across multiple networks. Router decides best path with help of path of traffic to travel in which network, with less traffic, less collision, more secure, less chance of hacking etc.

- Unicast – one to one
- Multicast – one to many e.g. conference, news to multiple receipts
- Broadcast – one to all
- Anycast – may be assigned to routers only. One to one of many e.g. used to identify a set of routers of an ISP, in a subnets etc.

4.1 Routing in the internet and CIDR (Classless Inter-Domain Routing)

CIDR (Classless Inter-Domain Routing, sometimes called Supernetting) is a way to allow more flexible allocation of Internet Protocol (IP) addresses than was possible with the original system of IP address classes. As a result, the number of available Internet addresses was greatly increased, which along with widespread use of network address translation (NAT), has significantly extended the useful life of IPv4.

4.2 Multicasting

- One to many addressing i.e. delivery of packets to many destinations e.g. location of servers by clients.
- An IPv6 multicast group is an arbitrary group of receivers that want to receive a particular data stream. This group has no physical or geographical boundaries--receivers can be located anywhere on the Internet or in any private network. Receivers that are interested in receiving data flowing to a particular group must join the group by signaling their local router. This signaling is achieved with the MLD (Multicast Listener Discovery) protocol.
- Routers use the MLD protocol to learn whether members of a group are present on their directly attached subnets. Hosts join multicast groups by sending MLD report messages. The network then delivers data to a potentially unlimited number of receivers, using only one copy of the multicast data on each subnet. IPv6 hosts that wish to receive the traffic are known as group members.
- Packets delivered to group members are identified by a single multicast group address. Multicast packets are delivered to a group using best-effort reliability, just like IPv6 unicast packets.
- The multicast environment consists of senders and receivers. Any host, even it is a member of a group, can send to a group. However, only the members of a group receive the message.
- A multicast address is chosen for the receivers in a multicast group. Senders use that address as the destination address of a datagram to reach all members of the group.
- Membership in a multicast group is dynamic; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time.
- How active a multicast group is, its duration, and its membership can vary from group to group and from time to time. A group that has members may have no activity.
- Inside a corporate network, Multicast can deliver live video to multiple nodes without having to have massive bandwidth on the part of the server delivering the video feed. This way you can have a video server feeding a 720p stream on only a 100Mb connection, and yet still serve that feed to 3000 clients.
- Multicast is... a need. Well, at least in some scenarios. If you have information (a lot of information, usually) that should be transmitted to various (but usually not all) hosts over an internet, then Multicast is the answer.
- One common situation in which it is used is when distributing real time audio and video to the set of hosts which have joined a distributed conference.
- When sending to multiple receivers?
    - Better bandwidth utilization
    - Fast processing
    - Lower router cycles required

Multicast applications

- Any Applications with multiple receivers - 1-to-many or many-to-many
- Live Video distribution e.g. Video Conference
- Periodic Data Delivery i.e. "Push" technology e.g. stock quotes, sports scores, magazines, newspapers, Advertisements
- Server/Web-site replication
- Reducing Network/Resource Overhead - more efficient to establish multicast tree rather than multiple point-to point links
- Distributed Interactive Simulation (DIS) e.g. war games, virtual reality

Multicast addresses

- Multicast addresses always start with FF00::/8

Multicast types

- **All Nodes Address: FF02 :: 1/8 –** All nodes or devices including routers are identified by this address on the local network segment can get multicast messages.
- **All Routers Address: FF01 :: 2/8** – All routers on a local network segment can hear multicast messages with this address as the destination. Used by devices to communicate with an IPv6 Router.

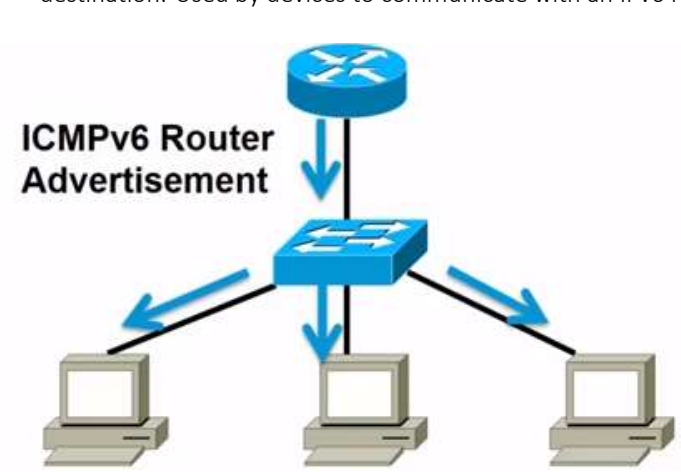


Fig. All nodes address process

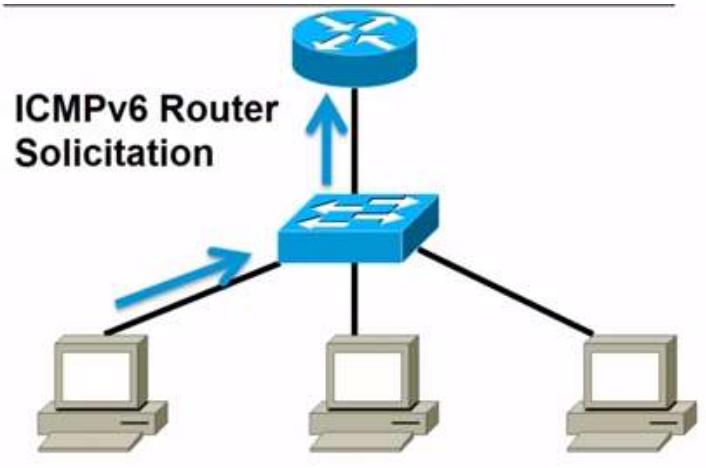


Fig. All routers address process

Multicast Listener Discovery Protocol (MLD - IPv6 Multicasting)

The purpose of Multicast Listener Discovery (MLD) is to enable each IPv6 router to discover the presence of multicast listeners (that is, nodes wishing to receive multicast packets) on its directly attached links, and to discover specifically which multicast addresses are of interest to those neighbouring nodes. This information is then provided to whichever multicast routing protocol is being used by the router, in order to ensure that multicast packets are delivered to all links where there are interested receivers.

MLD is an asymmetric protocol, specifying different behaviours for multicast listeners and for routers. For those multicast addresses to which a router itself is listening, the router performs both parts of the protocol, including responding to its own messages.

If a router has more than one interface to the same link, it need perform the router part of MLD over only one of those interfaces. Listeners, on the other hand, must perform the listener part of MLD on all interfaces from which an application or upper-layer protocol has requested reception of multicast packets.

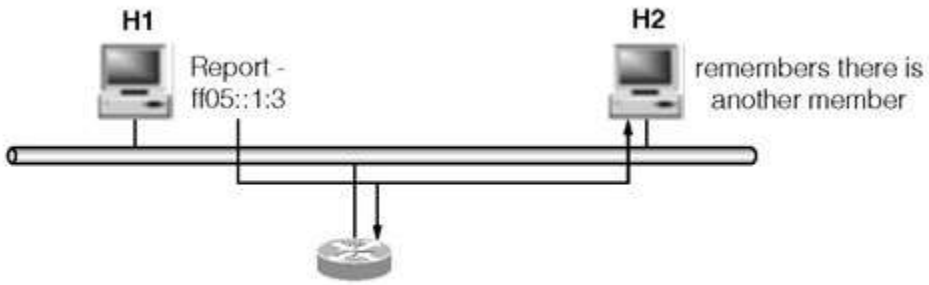
MLD Join Process

An IPv6 host joins a multicast group by sending a Multicast Listener Report message destined to the multicast address of interest. A multicast router is configured to accept all multicast packets from the link. The router recognizes the MLD packets through the Hop-by-Hop Router Alert option and passes these MLD messages to the upper layer. The multicast routing process within the router accepts and starts forwarding these multicast packets to the reported group according to the routing protocol it deploys.

A host that is a member of a group also receives Report messages sent to the group by other hosts joining the group. The receiving host then remembers that there is another host joining the group. The existence of the other group members affects the leave process described in Section 2.3.8.

Figure 2-3 gives an example of the join process. In this example, host H2 has already joined the multicast group ff05::1:3. Host H1 is starting up and joins the same group by sending out the Report message indicating its interest in the ff05::1:3 group. Both the router and H2 receive the Report message, and H2 remembers there is another group member.

FIGURE 2-3



Joining a group.

MLD Leave Process

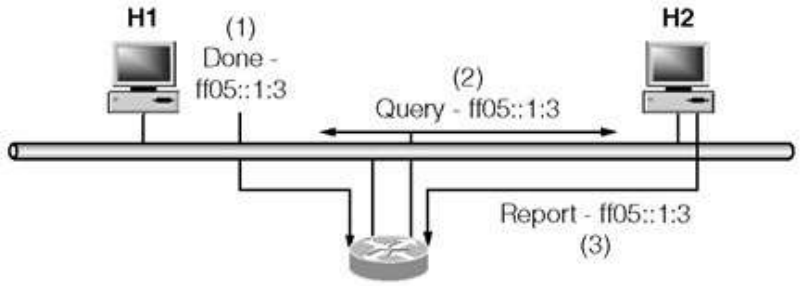
When a host is going to leave a multicast group and is the last node that sent a Multicast Listener Report message for the group (this can be detected by whether or not the host has heard a Report from another node for the group), it notifies the router of the departure by sending a Multicast Listener Done message to the All-Routers multicast group address (ff02::2).

Assume in the previous example that H1 did not receive any Report message from other nodes. Then H1 is the last member that sent a Report for ff05::1:3. Since H2 is aware of the presence of another listener (H1) being a member in that group, it will not generate a Done message if it leaves the group first. On the other hand, if H1 leaves the group before H2, it will notify the router with a Done message sent to the All-Routers multicast address (ff02::2).

The router responds to the Done message from H1 with a multicast-address specific Query. H2 responds to the Query with a Report message, indicating there is still a listener present. The router thus continues the multicast packet forwarding for this group. Figure 2-4 illustrates this scenario.

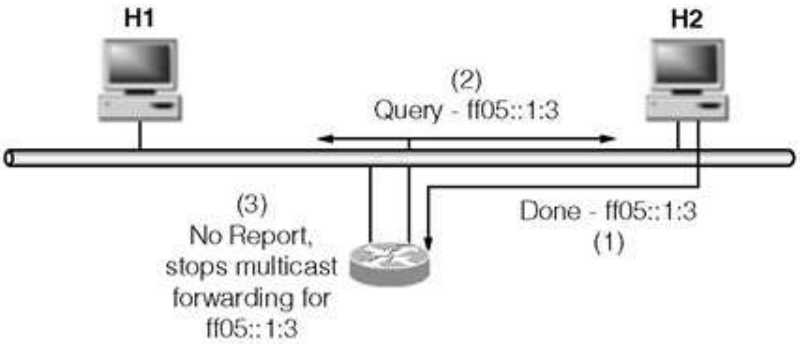
The router responds to the Done message by sending out Last Listener Query Count number of multicast-address specific Query messages, each separated by the Last Listener Query Interval seconds. If the departing node is the last member of the group, the router will not receive any Report. The router will cease forwarding multicast packets for the group once the transmissions complete without receiving any Report message. Continuing with the previous example, this progression takes place when H2 leaves the group as depicted in Figure 2-5.

FIGURE 2-4



Leaving a group.

FIGURE 2-5



Last node leaving a group.

4.3 Unidirectional Link Routing (UDLR)

UDLR simulates a bi-directional link by tunneling over the return path

- •Supports both IP unicast and multicast
- •Supports all IP routing protocols (including IS-IS for IP)
- •Supports address resolution via ARP and NHRP

Basic UDLR Tunnel Operation

- Traffic from the downstream network (B or C) destined for A's unidirectional interface will traverse the unidirectional tunnel.
- Packets are encapsulated by B in a GRE IP packet and forwarded to upstream router A.
- A decapsulates the GRE packet, and places the original packet on the input queue belonging to A's unidirectional link.
- Access filters limit traffic on the upstream and downstream tunnel interfaces
- IP, ARP, NHRP, and CLNS (for IS-IS IP routing) are supported.
- Upstream router A's routing protocol must never advertise the tunnel subnet over the unidirectional link interface - this will cause a recursive tunnel in downstream routers.

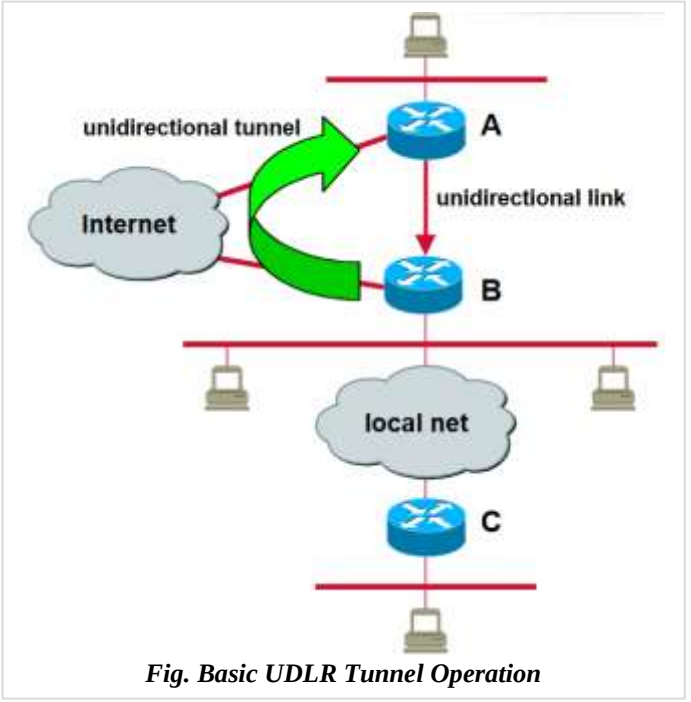


Fig. Basic UDLR Tunnel Operation

Routing protocols support unidirectional links only if the unidirectional links emulate bidirectional links because routing protocols expect to send and receive traffic through the same interface. Unidirectional links are advantageous because when you transmit mostly unacknowledged unidirectional high-volume traffic (for example, a video broadcast stream) over a high-capacity full-duplex bidirectional link, you use both the link from the source to the receiver and the equally high-capacity reverse-direction link, called the "back channel," that carries the few acknowledgements from the receiver back to the source. UDE and UDLR support use of a high-capacity unidirectional link for the high-volume traffic without consuming a similar high-capacity link for the back channel. UDE provides a high-capacity unidirectional link. UDLR provides the back channel through a tunnel that is configured over a regular-capacity link, and also provides bidirectional link emulation by transparently making the back channel appear to be on the same interface as the high-capacity unidirectional link.

4.4 RIPng (RFC 2080)

Unicast Routing – mechanism of sending the information from a single sender to a single receiver. It is a point to point communication between sender and receiver. E.g. HTTP, FTP request.

- Unicast Routing Protocol consists of:
  - RIP (Routing Information Protocol)
  - OSPF (Open Shortest Path First)
  - BGP (Border Gateway Protocol)
- Unicast routing is a process that enable sender to send a unicast IP packets to the destination node.
- 1 router or more intermediate routers may be used, depending to the destination of the node. (Figure 1)
- Unicast routing protocol is a set of rules of forwarding unicast traffic from a source to a destination on an internetwork.

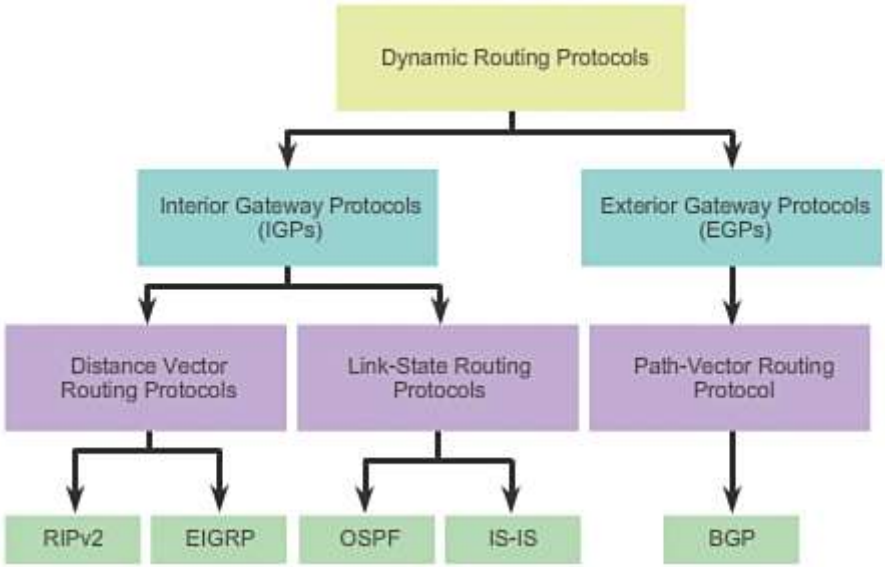


Fig. 2. Routing Protocol Classification

For example, IPv4 routing protocols are classified as follows:

- RIPv2: IGP, distance vector, classless protocol

- **EIGRP**: IGP, distance vector, classless protocol developed by Cisco
- **OSPF**: IGP, link-state, classless protocol
- **IS-IS**: IGP, link-state, classless protocol
- **BGP**: EGP, path-vector, classless protocol

An ***autonomous system (AS)*** is a collection of routers under a common administration such as a company or an organization. An AS is also known as a routing domain. Typical examples of an AS are a company's internal network and an ISP's network.
The Internet is based on the AS concept; therefore, two types of routing protocols are required:

- ***Interior Gateway Protocols (IGP)***: Used for routing within an AS. It is also referred to as intra-AS routing. Companies, organizations, and even service providers use an IGP on their internal networks. IGPs include RIP, EIGRP, OSPF, and IS-IS.
- ***Exterior Gateway Protocols (EGP)***: Used for routing between autonomous systems. It is also referred to as inter-AS routing. Service providers and large companies may interconnect using an EGP. The Border Gateway Protocol (BGP) is the only currently viable EGP and is the official routing protocol used by the Internet.

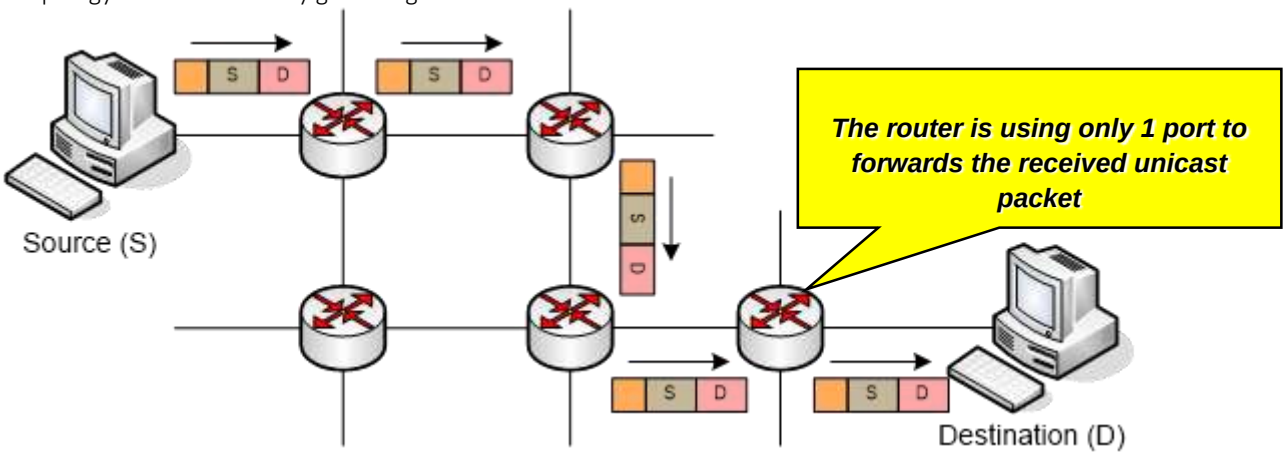
Distance Vector Routing Protocols

Distance vector means that routes are advertised by providing two characteristics:

- **Distance**: Identifies how far it is to the destination network and is based on a metric such as the hop count, cost, bandwidth, delay, and more.
- **Vector**: Specifies the direction of the next-hop router or exit interface to reach the destination.

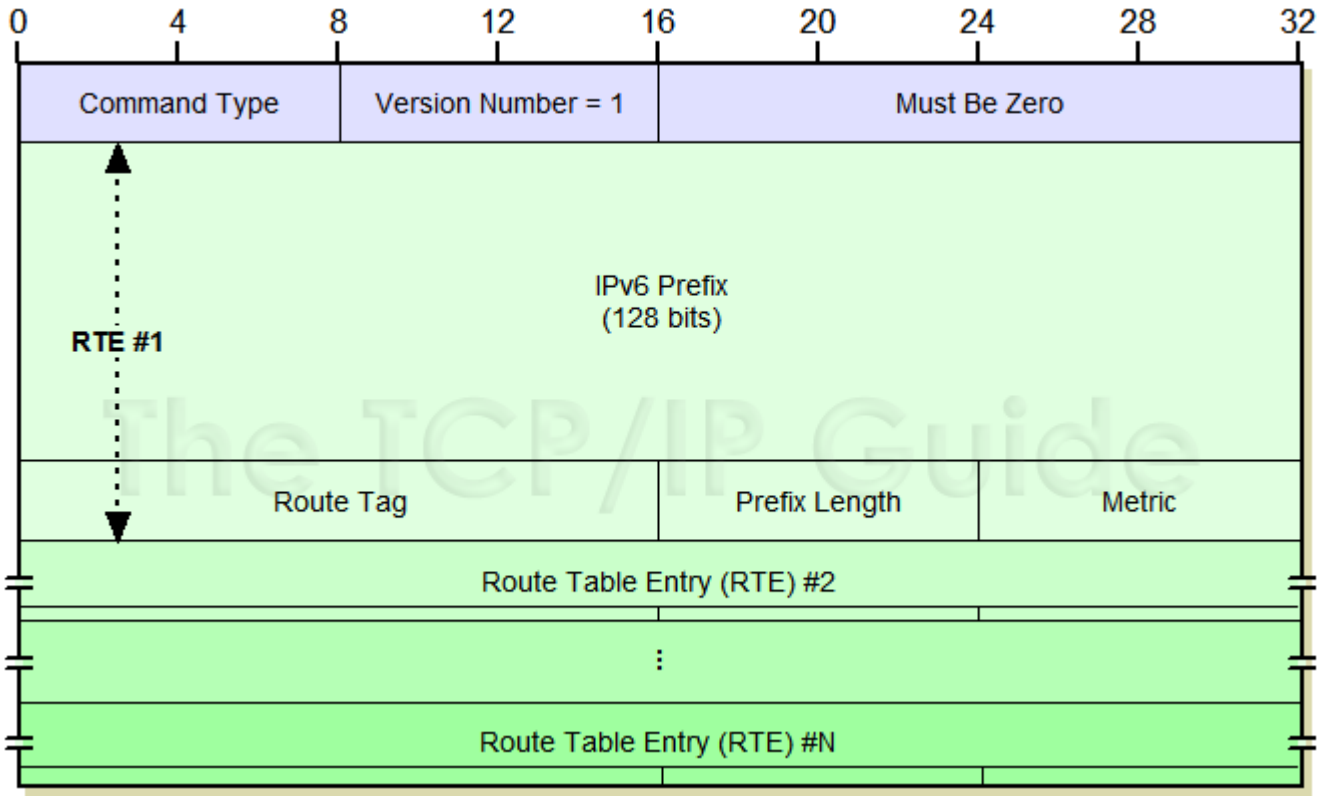
Link-State Routing Protocols

In contrast to distance vector routing protocol operation, a router configured with a ***link-state routing protocol*** can create a complete view or topology of the network by gathering information from all of the other routers.



- RIPng is designed to allow routers to exchange information to compute routes in IPv6-enabled network. RIPng relies on certain information about each of the networks, mainly the metric. RIPng metric is a value between 1 and 15, inclusive. The maximum path limit is 15, after which the network is considered unreachable. RIPng supports multiple IPv6 addresses on each interface.
- RIPng for IPv6 is a simple routing protocol with a periodic route advertising mechanism designed for use in small to midsize IPv6 networks. RIPng for IPv6 does not scale well to a large or very large IPv6 network
- RIPng uses a simple mechanism to determine the metric (cost) of a route. It basically counts the number of routers (hops) to the destination. Each router counts as one hop. Routes with a distance greater than or equal to 16 are considered to be unreachable. The router periodically distributes information about its routes to its directly connected neighbors using **RIPng response messages.** Upon receiving RIPng response messages from its neighbor, the router adds the distance between the neighbor and itself (usually one, as in one hop) to the metric of each route received.

RIPng Message Formats



Field Name	Size (bytes)	Description
Command	1	***Command Type:*** Identifies the type of RIPng message being sent. **1** = *RIPng Request*, **2** = *RIPng Response*.
Version	1	***Version Number:*** Set to 1 (not 6, since this is the first version of the new protocol RIPng.)
Must Be Zero	2	***Reserved:*** Field reserved; value must be set to all zeroes.
Route Table Entries (RTEs)	Variable	***Route Table Entries (RTEs):*** The body of an RIPng message consists of a variable number of *Route Table Entries (RTEs)* that contain information about routes. Each entry is 20 bytes long and has the following subfields:

Subfield Name	Size (bytes)	Description
IPv6 Prefix	16	***IPv6 Prefix:*** The 128-bit IPv6 address of the network whose information is contained in this *RTE*.
Route Tag	2	***Route Tag:*** Additional information to be carried with this route, as defined in RIP-2.
Prefix Len	1	***Prefix Length:*** The number of bits of the IPv6 address that is the network portion (the remainder being the host portion). This is the number that normally would appear after the "slash" when specifying an IPv6 network address, and is analogous to an IPv4 subnet mask. See the description of IPv6 prefix notation for more details.
Metric	1	***Metric:*** The distance for the network indicated by the IP address, as in RIP-1. Values of 1 to 15 indicate the number of hops to reach the network (as described in the general discussion of the RIP algorithm) while a value of 16 represents "infinity" (an unreachable destination).

RIPng Routing Table:

Router keeps the following entries in the routing table

- **IPv6 Route :** Address prefix and prefix length of the destination address
- **Next Hop Address :** The IPv6 Address (link-local) of the first router along the path
- **Next Hop Interface :** The physical interface used to reach the next hop
- **Metric :** Number indicating the total distance to the destination. RIPng advertizes directly connected routes with the configured outgoing metric of 1.
- **Timer**: Amount of time since the information about the route was last updated
- **Route change flag**: Set to control triggered routing updates
- **Route Source**: Entity to provide route information ◦ eg: Ripng, OSPF etc..

4.5 OSPF for IPv6 (OSPFv3) RFC 5340

- **Open Shortest Path First** (**OSPF**) is a routing protocol for Internet Protocol (IP) networks. It uses a link state routing (LSR) algorithm and falls into the group of interior gateway protocols (IGPs), operating within a single autonomous system (AS). It is defined as OSPF Version 2 in RFC 2328 (1998) for IPv4.[1] The updates for IPv6 are specified as OSPF Version 3 in RFC 5340 (2008).
- **OSPF** is perhaps the most widely used IGP in large enterprise networks. Intermediate System to Intermediate System (IS-IS), another link-state dynamic routing protocol, is more common in large service provider networks.
- **OSPF** has had a long history as a premier IP routing protocol. OSPF has continued to stay relevant by adapting to IPv6 and is now evolved into a fully dual-protocol multi-AF routing protocol. Organizations now have multiple options for deploying OSPF. Organizations can stick with OSPFv2 for IPv4, and then use OSPFv3 for IPv6-only for a configuration that separates the control planes and the forwarding planes. Organizations can now combine the configuration of IPv4 and IPv6 into a single OSPFv3 process that can work equally well for both IP protocols.
- **OSPF** cost of each router link is a unit less number that the network administrator assigns, and it can include delay, bandwidth, and other cost factors. The accumulated cost between network segments in an OSPF network must be less than 65,535.
- It was designed to overcome some of the limitations introduced by RIP, such as the small diameter, long convergence time, and a metric that does not reflect the characteristics of the network. In addition, OSPF handles a much larger routing table to accommodate large number of routes.

How OSPFv3 Works

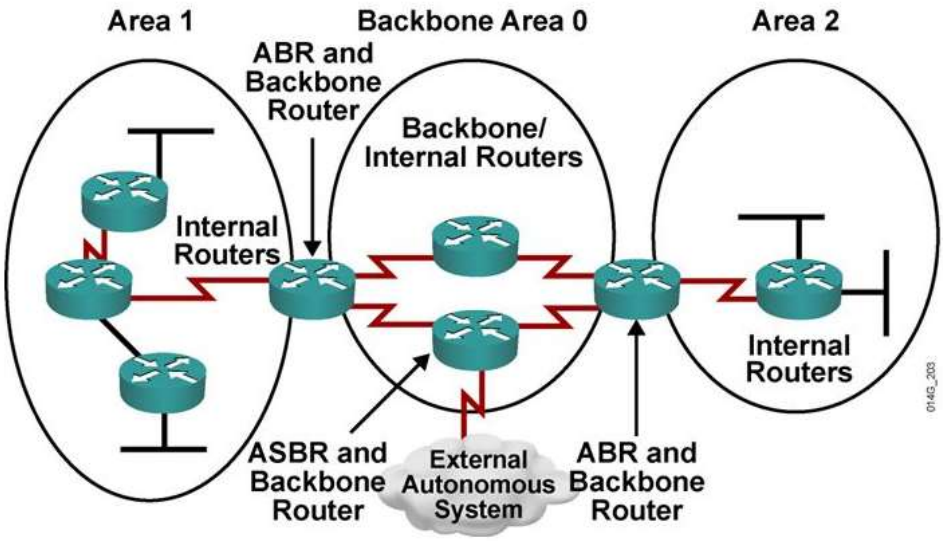
- OSPFv3 is a routing protocol for IPv4 and IPv6. It is a link-state protocol, as opposed to a distance-vector protocol. Think of a link as being an interface on a networking device. A link-state protocol makes its routing decisions based on the states of the links that connect source and destination machines. The state of a link is a description of that interface and its relationship to its neighboring networking devices. The interface information includes the IPv6 prefix of the interface, the network mask, the type of network it is connected to, the devices connected to that network, and so on. This information is propagated in various type of link-state advertisements (LSAs).
- A device's collection of LSA data is stored in a link-state database. The contents of the database, when subjected to **the Dijkstra algorithm**, result in the creation of the OSPF routing table. The difference between the database and the routing table is that the database contains a complete collection of raw data; the routing table contains a list of shortest paths to known destinations via specific device interface ports.
- OSPFv3, which is described in RFC 5340, supports IPv6 and IPv4 unicast AFs.

Comparison of OSPFv3 and OSPF Version 2

- **Protocol Processing Per-Link, Not Per-Subnet**
  - Pv6 uses the term "link" to indicate "a communication facility or medium over which nodes can communicate at the link layer" ([IPV6]).
  - "Interfaces" connect to links. Multiple IPv6 subnets can be assigned to a single link, and two nodes can talk directly over a single link, even if they do not share a common IPv6 subnet (IPv6 prefix).
- **Removal of Addressing Semantics**
  - In OSPF for IPv6, addressing semantics have been removed from the OSPF protocol packets and the main LSA types, leaving a network-protocol-independent core.
- **Addition of Flooding Scope**
  - Flooding scope for LSAs has been generalized and is now explicitly coded in the LSA's LS type field.
- **Explicit Support for Multiple Instances per Link**
  - OSPF now supports the ability to run multiple OSPF protocol instances on a single link. For example, this may be required on a NAP segment shared between several providers. Providers may be supporting separate OSPF routing domains that wish to remain separate even though they have one or more physical network segments (i.e., links) in common.

- Use of Link-Local Addresses
  - IPv6 link-local addresses are for use on a single link, for purposes of neighbor discovery, auto-configuration, etc. IPv6 routers do not forward IPv6 datagrams having link-local source addresses [IP6ADDR].
  - Link-local unicast addresses are assigned from the IPv6 address **range FE80/10.**
- Authentication Changes
  - In OSPF for IPv6, authentication has been removed from the OSPF protocol. The "AuType" and "Authentication" fields have been removed from the OSPF packet header, and all authentication-related fields have been removed from the OSPF area and interface data structures.
- Packet Format Changes
  - OSPF for IPv6 runs directly over IPv6. Aside from this, all addressing semantics have been removed from the OSPF packet headers, making it essentially "network-protocol-independent". All addressing information is now contained in the various LSA types only.
- LSA Format Changes
  - **LSA-Link State Acknowledge**, All addressing semantics have been removed from the LSA header, router-LSAs, and network-LSAs. These two LSAs now describe the routing domain's topology in a network-protocol-independent manner. New LSAs have been added to distribute IPv6 address information and data required for next-hop resolution. The names of some of IPv4's LSAs have been changed to be more consistent with each other.
- Handling Unknown LSA Types
  - Handling of unknown LSA types has been made more flexible so that, based on the LS type, unknown LSA types are either treated as having link-local flooding scope, or are stored and flooded as if they were understood. This behaviour is explicitly coded in the LSA Handling bit of the link state header's LS type field
- Stub/ NSSA Area Support
  - **NSSSA (Not So Stubby Area)** - An NSSA autonomous system boundary router (ASBR) generates this LSA and an NSSA area border router (ABR) translates it into a type 5 LSA (*External Link – Generated by an ASBR and flooded throughout the AS to advertise a route external to OSPF.), which* gets propagated into the OSPF domain.
  - **A stub area** that only allows routes internal to the area and restricts Type 3 LSAs (Network Summary - *This type of LSA are generated by ABR, they represent networks from an area and are sent to the rest of the area in OSPF domain.) from* entering the stub area is often called a totally stubby area.
  - In OSPF for IPv4, stub and NSSA areas were designed to minimize link-state database and routing table sizes for the areas' internal routers. This allows routers with minimal resources to participate in even very large OSPF routing domains.
- Identifying Neighbors by Router ID
  - In OSPF for IPv6, neighboring routers on a given link are always identified by their OSPF Router ID. This contrasts with the IPv4 behavior where neighbors on point-to-point networks and virtual links are identified by their Router IDs while neighbors on broadcast, NBMA, and point-to-multipoint links are identified by their IPv4 interface addresses.

Types of OSPF Router



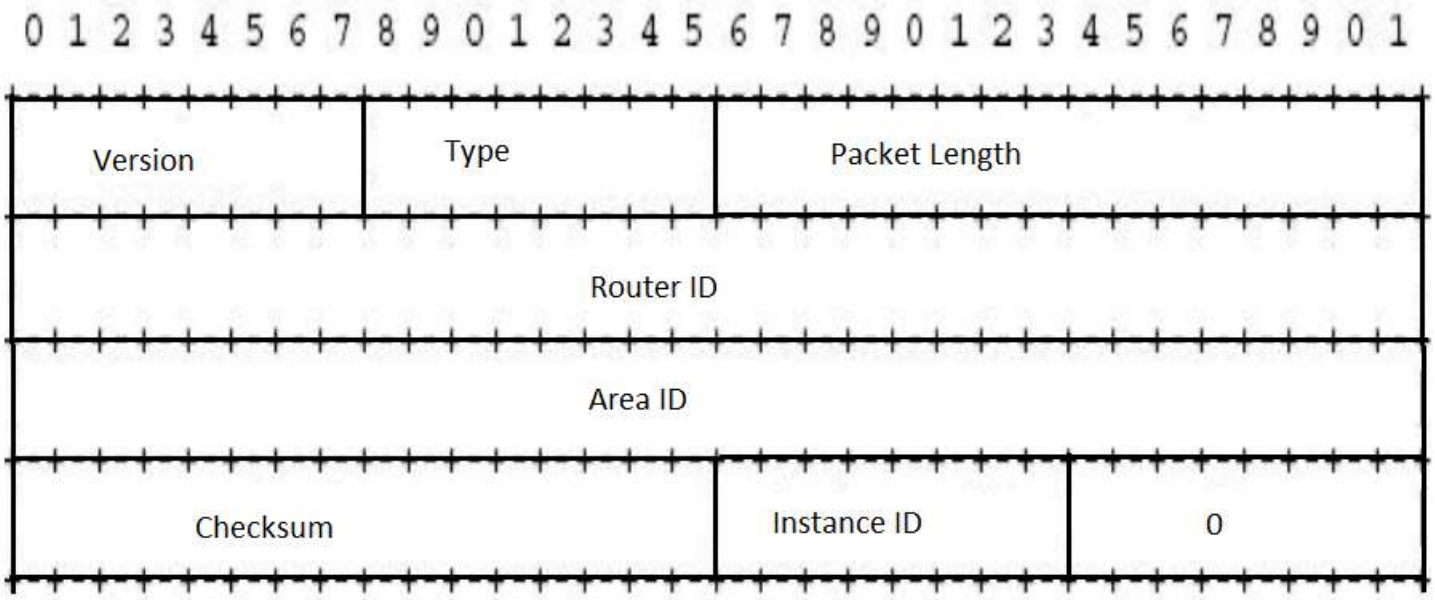
Internal routers (IRs) are routers whose directly connected networks all belong to the same OSPF area.
Area Border Routers (ABRs) are attached to multiple OSPF areas, so there can be multiple ABRs within a network.
Autonomous System Boundary System (ASBRs) are connected to more than one AS and exchange routing information with routers in another AS.

Backbone Routers: Routers whose interfaces connect them only to the backbone area are considered backbone routers (BRs). BRs do not have an interface to the other OSPF areas, because if they did, they would be considered ABRs.

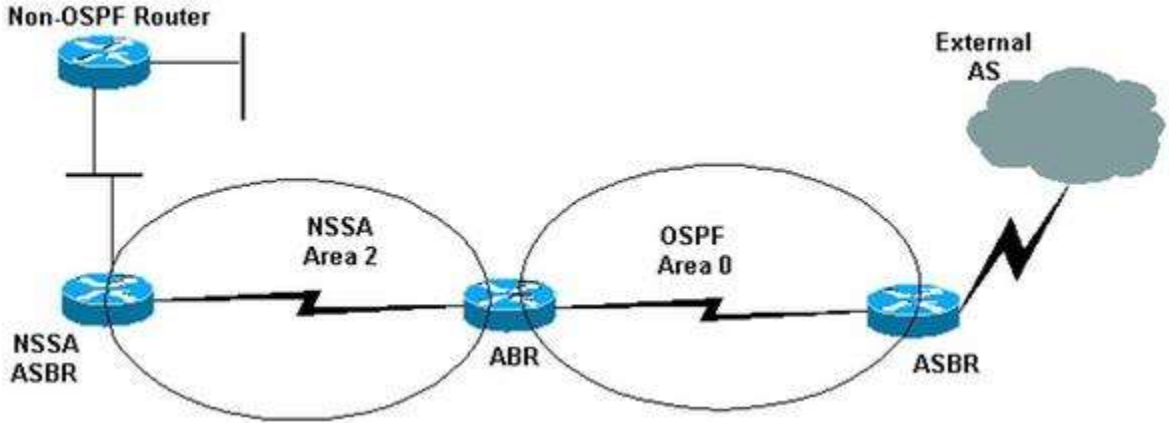
OSPFv3 Packet Types

- **Hello Packet**: discovers neighbours and builds adjacencies between them.
- **Database Description (DBD):** checks for database synchronization between routers.
- **Link State Router (LSR)**: request specific link-state records from router to router.
- **Link State Update (LSU)**: sends specifically requested link state records.
- **Link State Acknowledgement (LSA)**: acknowledges the other packet types.

OSPFv3 Header Format



- **Version**: 3
- **Type**: Type of OSPF packet
  1- Hello Message, 2- Database Description message, 3- Link State Request, 4- Link State Update, 5- Link State Acknowledgement
- **Packet Length**: Length of OSPF packet in bytes. It includes the standard 16 bytes as well.
- **Router ID**: The 32-bit Router ID of the packet source
- **Area ID**: as **Sub-domains**, A 32-bit Area ID indicating the area that this packet belongs to. Every packet belongs to a single area.
- **Checksum**: Standard 16-bit checksum
- **InstanceID**: Enables multiple instances of OSPF to be run over a single link. It has local significance only. Received packets whose Instance ID is not equal to the receiving interface's Instance ID, are discarded.



4.6 PIM-SM and DVMRP for IPv6

- PIM (Protocol Independent Multicast) can function without being dependent on any specific routing protocol i.e. does not use its own mechanism of topology discovery in applications such as video and audio conferencing.
- PIM does not build its own routing tables. PIM uses the unicast routing table.

- A group specified as **dense** is not mapped to an RP. Instead, data packets destined for that group are forwarded by means of PIM dense-mode rules. A group specified as **sparse** is mapped to an RP, and data packets are forwarded by means of PIM sparse-mode rules.
- **PIM-SM (Sparse Mode)** is based on a "**join protocol**" routing flow step by step, where traffic is not forwarded on a segment unless an explicit request originates (typically through IGMP) from the network segment.
- **PIM-DM (Dense Mode)** is based on a "**flood and prune**" approach, where everyone receives traffic until they explicitly inform (through the PIM-DM prune mechanism) that they do not want that particular stream. Thus, PIM-DM is typically deployed in topologies where listeners are densely populated. And PIM-SM is typically deployed where the receivers are sparsely populated over the network, so that most of the network segments' bandwidth is conserved. You can configure dense mode or sparse mode on a per-interface basis. After they are enabled, some interfaces can run dense mode, while others run sparse mode.

Types of Multicast Routing Protocols

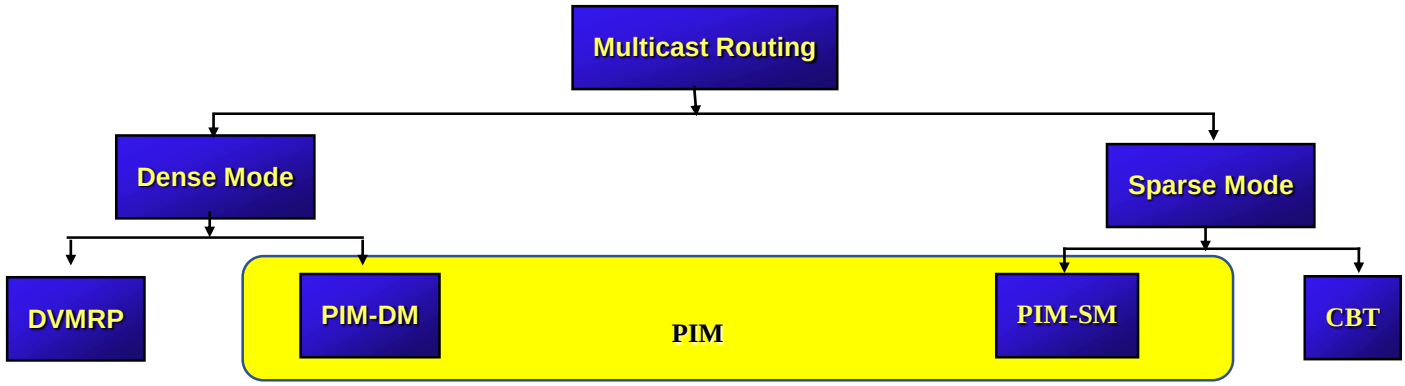


Fig. 2. Types of Multicast Routing Protocol

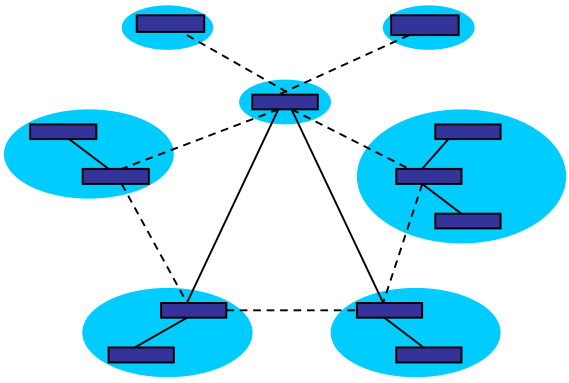


Fig. Sparse Mode

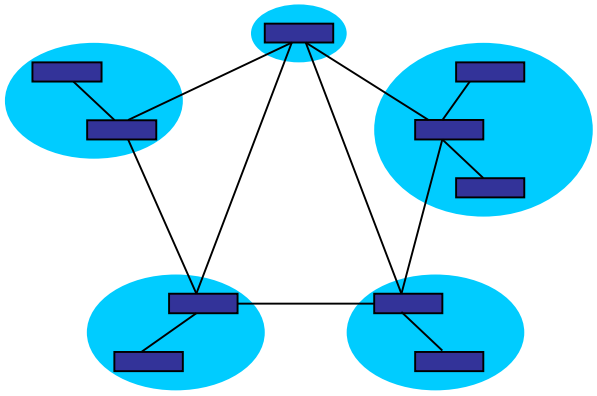


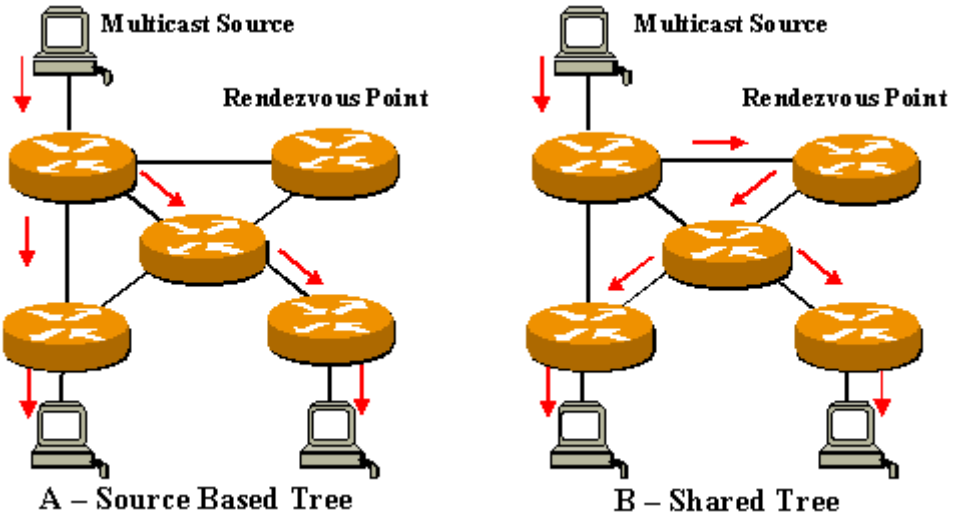
Fig. Dense Mode

PIM-SM

Receiver are scattered over a large area.

- Multicast protocols build trees for transmitting group information G from a source S to any number of interested receivers.
- The source application only transmits the data stream to the group multicast address G without knowing any detail about the underlying tree construction process.
- **Source distribution trees** are identified in the multicast routing table as (S, G) and provide optimal routing from the Source to the Receiver at the cost of more state on the network.
- **Shared distribution trees** are identified in the multicast routing table as (*, G) and usually doesn't provide optimal paths from the Source to the Receivers because the RP is now the root of the SPT and may not be in the optimal path to the Source.
- **3 phases of PIM** – SM to build a multicast distribution tree
    - RP Tree: Shared tree
    - Register Stop
    - Shortest Path Tree
- **Source tree** multicast is the traditional multicast with its root at the top and branches forming a spanning tree through the network to the receivers. for efficient direct delivery along the "shortest" path from source to receiver.

- **Shared trees** use a single (central) common root placed at some chosen point in the network. This shared root is called a rendezvous point (RP)



Rendezvous Points Tree (RPT)

- A **Rendezvous Point (RP)** is a router in a multicast network domain that acts as a shared root for a multicast shared tree. Any number of routers can be configured to work as RPs and they can be configured to cover different group ranges.
- A **shared tree** is a tree with the root at the RP. With source tree, the tree root is located at the source.
- **RP** is the meeting point between multicast sources **S** and receivers interested in receiving the information sent to some group **G**. Once the interested receiver discover some source **S** sending information to a group of interest **G**, if the shared tree is not the most optimal path to the source, a source tree is built.
  - In shared tree: passing through the RP
    - Root of the distribution tree is a router, not a host
  - In PIM-SM multicast routing protocol
    - the core router at the root of the shared tree is the *rendezvous point (RP*)
    - The traffic from upstream and join/prune message from downstream routers "rendezvous" at this core router.
    - To join the Shared Tree (host want to receive multicast traffic)
    - Router or DR executes an RPF check on the RP address in its routing table
    - Produces the interface closest to the RP.
    - Send a join message (*,G) out on this interface.
    - These upstream routers
    - Repeat the those 3 processes until it reaches the RP.
    - It is building the shared tree or RPT as it goes until it reaches the RP.

Register Stop

- Sources of multicast traffic don't necessarily join the group to which they are sending data
  - First Hop Router (FHR) or DR can receive the traffic without knowing the information on how to send the traffic to the RP through the tree.
  - It encapsulates the packet and send to RP as unicast packet.(Register Msg)
  - RP de-encapsulates the Register message and forwards the extracted data packet to downstream members on the RPT
- Encapsulation at DR and decapsulation for Register message at RP is not efficient
  - RP initiates a (S, G) Join toward S and the path to source is established.
  - DR starts sending traffic from S using both native multicast and Register-encapsulated messages
  - RP detected a duplicated multicast packets, it will send a "Register Stop" message to tell DR stop sending Register Message.
  - DR stop sending the Register message and RP now only receive the packet from native multicast packet.

Shortest Path Tree (SPT)

- RP is a place for a source and receivers to meet
  - But if there is too many multicast group "rendezvous" there, it might become a bottleneck
  - Hence, establishing SPT might solve this problem and also reduce the path delay from Source to receives
  - SPT can be accomplish by specifying an SPT-Threshold in terms of bandwidth.
  - If this threshold is exceeded, the last-hop DR joins the SPT
- Build the SPT
  - Router executes an RPF check on the source address in its routing table to find the interface closest to the source.
  - Issues an (S,G) Join to the RPF next router toward S.

- Each upstream router repeats this process, until
  - Arrives at the subnet of S
  - Router that already has (S,G) Join state.
- The DR at the Source subnet then starts forwarding packets onto the source tree to the receiver
- Now, the receiver's DR, it receives packets from Shared Tree (RP) and Source Tree (SPT). To stop receiving duplicating traffics,
  - Receiver's DR sends a PIM Prune message towards the RP router.
  - This message is known as (S,G,rpt) Prune, to tell RP this particular traffic coming in from the RPT are no longer needed
- PIM Prune message is received by the RP router, then it stops sending this particular multicast traffic down to the receiver's DR.

PIM Dense mode – routing flow

- flood and prune approach
- PIM sparse mode uses a source based tree between the source router 'S' and the last hop router 'G'. the state in the multicast table will look like (S,G)

1) Multicast source starts flooding traffic through its PIM enabled interfaces
2) If multiple routers are forwarding the traffic, a PIM assert message is used to determine the PIM forwarder. The router with the better metric (default highest IP address) wins the election.
3) If the routers do not have multicast receivers then it will send a Prune message to their upstream router requesting the distribution tree be pruned. If there is another router on that broadcast segment and it has multicast receivers then the prune message is ignored (join override)
4) Between the source and receiver builds the source based tree (S,G). Remember that dense mode is a "flood-and-prune" behaviour which repeats every 3 minutes which does not scale well…..which means hello sparse mode.

How does PIM sparse mode work?

1) The receiving router (last hop router) sends a IGMP (Internet Group Message Protocol) join message to the RP creating a (*,G) along the shared tree between the RP and the last hop router.
2) The source (sender of the traffic) then builds its source based between (S,G) between it and the RP. Before the source tree is completely established the source sends a unicast register message to the RP.
3) Once the RP receives a multicast packet over the source tree it sends a register stop message to the source. This will tell the source to stop sending the multicast traffic inside a register message.
4) We now have two trees built. We have the shared tree (*,G) between RP and the last hop router. And we have the sourced based (S,G) tree from the first hop router to the RP.
5) We now are going to start the communicating between the last hop router and the first hop router. The last hop router sends a join message directly to the first hop router to form an optimal path (source path tree) between the sender and the receiver.
6) Now since we have communication directly between the last hop (receiver) router and the first hop router (sender) the last hop router sends a (S,G) prune bit message to the RP asking it to stop sending multicast packets.
7) Now with he shared tree built the last hop router and the RP is pruned the RP doesn't need to receive multicast traffic from the first hop router anymore.
8) The RP now sends a (S,G) prune message to the first hop router. Now we have multicast traffic flowing just between the first hop router and the last hop router. This path from switching over from the shared + source based tree to the RP is called the SPT switchover (shortest path tree)

DVMRP (Distance Vector Multicast Routing Protocol)

It is the oldest routing protocol that has been used to support multicast data transmission over networks. The protocol sends multicast data in the form of unicast packets that are reassembled into multicast data at the destination.
DVMRP is a dense-mode multicasting protocol and therefore uses a broadcast and prune mechanism. The protocol builds a source-rooted tree (SRT) in a similar way to PIM dense mode. DVMRP routers flood datagrams to all interfaces except the one that provides the shortest unicast route to the source. DVMRP uses pruning to prevent unnecessary sending of multicast messages through the SRT.

DVMRP Operation

The protocol is based on the Routing Information Protocol (RIP). The router generates a routing table with the multicast group of which it has knowledge with corresponding distances (i.e. number of devices/routers between the router and the destination). When a multicast packet is received by a router, it is forwarded by the router's interfaces specified in the routing table.

DVMRP operates via a reverse path flooding technique, sending a copy of a received packet (specifically IGMP messages for exchanging routing information with other routers) out through each interface except the one at which the packet arrived. If a router (i.e. a LAN which it borders) does not wish to be part of a particular multicast group, it sends a "prune message" along the source path of the multicast.

A DVMRP router **sends prune messages** to its neighbors if it discovers that:

- The network to which a host is attached has no active members of the multicast group.
- All neighbors, except the next-hop neighbor connected to the source, have pruned the source and the group.

- When a neighbor receives a prune message from a DVMRP router, it removes that neighbor from its (S,G) pair table, which provides information to the multicast forwarding table.
- When a host on a previously pruned branch attempts to join a multicast group, it sends an **IGMP** message to its first-hop router. The first-hop router then sends a graft message upstream.

Identifying Neighbors

In this implementation of DVMRP, a *neighbor* is a directly connected DVMRP router. When you enable DVMRP on an interface, the associated VR adds information about local networks to its DVMRP routing table. The VR then sends probe messages periodically to learn about neighbors on each of its interfaces. To ensure compatibility with other DVMRP routers that do not send probe messages, the VR also updates its DVMRP routing table when it receives route report messages from such routers.

Advertising Routes

As its name suggests, DVMRP uses a distance-vector routing algorithm. Such algorithms require that each router periodically inform its neighbors of its routing table. DVMRP routers advertise routes by sending DVMRP report messages. For each network path, the receiving router picks the neighbor advertising the lowest cost and adds that entry to its routing table for future advertisement.

The cost, or metric, for this routing protocol is the hop count back to the source. The hop count for a network device is the number of routers on the route between the source and that network device.

Table 41 shows an example of the routing table for a DVMRP router.

Table 41: Sample Routing Table for a DVMRP Router

Source Subnet	Subnet Mask	From Router	Metric	Time Before Entry Is Deleted from Routing Table	Input Port	Output Port
143.2.0.0	255.255.0.0	143.32.44.12	4	85	3/0	4/0, 4/1
143.3.0.0	255.255.0.0	143.2.55.23	2	80	3/1	4/0, 4/1
143.4.0.0	255.255.0.0	143.78.6.43	3	120	3/1	4/0, 4/1

The DVMRP router maintains an (S, G) pair table that provides information to the multicast forwarding table. The (S,G) pair table is based on:

- Information from the DVMRP routing table
- Information learned from prune messages
- If IGMP and DVMRP are on the same interface, group information learned from IGMP

The (S,G) pair table includes a route from each subnetwork that contains a source to each multicast group of which that source is a member. These routes can be static or learned routes. Table 42 shows an example of the (S,G) pair table for DVMRP. A dash (–) in the Input Port column indicates that the interface is associated with a protocol other than DVMRP.

Table 42: Sample DVMRP (S,G) Pair Table

Source Subnet	Multicast Group	Time Before Entry Is Deleted from Routing Table	Input Port	Output Port
143.2.0.0	230.1.2.3	85	3/0	4/0, 4/1
	230.2.3.4	75	3/0	4/0, 4/1
	230.3.4.5	60	3/0	4/1
	230.4.5.6	90	—	4/0
143.3.0.0	230.1.2.3	80	3/1	4/0, 4/1

The basic operation of DVMRP consists of four processes :

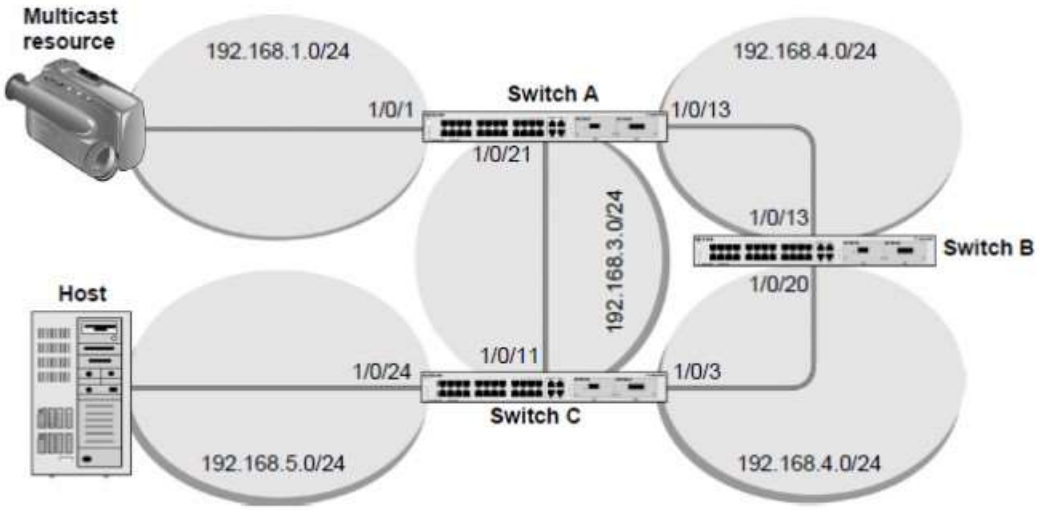
(i) Neighbor discovery, which is used to find other DVMRP capable routers attached to a common network.
(ii) Route exchange, similar to RIP.
(iii) Graft messages, used to add networks to the forwarding list.
(iv) Prune messages, used to remove networks from the forwarding list

The DVMRP is used for multicasting over IP networks without routing protocols to support multicast. The DVMRP is based on the RIP protocol but more complicated than RIP. DVRMP maintains a link-state database to keep track of the return paths to the source of multicast packages.

The DVMRP operates as follows:

- The first message for any source-group pair is forwarded to the entire multicast network, with respect to the time-to-live (TTL) of the packet.
- TTL restricts the area to be flooded by the message.
- All the leaf routers that do not have members on directly attached sub-networks send back prune messages to the upstream router.
- The branch that transmitted a prune message is deleted from the delivery tree.
- The delivery tree, which is spanning to all the members in the multicast group, is constructed.

In the figure below, DVMRP is running on switches A, B, and C. IGMP is also running on Switch C, which is connected to the host directly. After the host sends an IGMP report to switch C, multicast streams are sent from the multicast resource to the host along the path built by DVMRP.



DVRMP's main tasks include:

- Tracks multicast datagram source paths
- Encapsulates packets as Internet Protocol (IP) datagrams
- Supports multicast IP datagram tunneling via unsupported encapsulated and addressed unicast packet routers
- Generates dynamic multicast IP delivery trees via reverse path multicasting and a distributed routing algorithm
- Exchanges routing datagrams made up of small, fixed-length headers and tagged data streams via Internet Group Management Protocol
- Handles tunnel and physical interfacing according to broadcast routing exchange source trees produced during truncated tree branch removal
- Manages reverse path forwarding for multicast traffic forwarding to downstream interfaces