PIM Sparse Mode Protocol Mechanics

Module 3
Agenda

- PIM Neighbor Discovery
- PIM State
- PIM SM Forwarding
- PIM SM Joining
- PIM SM Registering
- PIM SM SPT-Switchover
- PIM SM Pruning
- PIM SM Special Cases
PIM Neighbor Discovery

- PIMv2 Hellos are periodically multicast to the “All-PIM-Routers” (224.0.0.13) group address. (Default = 30 seconds)
  - Note: PIMv1 multicasts PIM Query messages to the “All-Routers” (224.0.0.2) group address.
- If the “DR” times-out, a new “DR” is elected.
- The “DR” is responsible for sending all Joins and Register messages for any receivers or senders on the network.

- PIM Neighbor Discovery
  - PIM Hellos are sent periodically to discover the existence of other PIM routers on the network and to elect the Designated Router.
  - For Multi-Access networks (e.g. Ethernet), the PIM Hello messages are multicast to the “All-PIM-Routers” (224.0.0.13) multicast group address.

- Designated Router (DR)
  - For multi-access networks, a Designated Router (DR) is elected. In PIM Sparse mode networks, the DR is responsible for sending Joins to the RP for members on the multi-access network and for sending Registers to the RP for sources on the multi-access network. For Dense mode, the DR has no meaning. The exception to this is when IGMPv1 is in use. In this case, the DR also functions as the IGMP Querier for the Multi-Access network.

- Designated Router (DR) Election
  - To elect the DR, each PIM node on a multi-access network examines the received PIM Hello messages from its neighbors and compares the IP Address of its interface with the IP Address of its PIM Neighbors. The PIM Neighbor with the highest IP Address is elected the DR.
  - If no PIM Hellos have been received from the elected DR after some period (configurable), the DR Election mechanism is run again to elect a new DR.
### PIM Neighbor Discovery

#### Command Output

```plaintext
wan-gw8>show ip pim neighbor
PIM Neighbor Table
Neighbor Address  Interface          Uptime    Expires   Mode
171.68.0.70       FastEthernet0      2w1d      00:01:24  Sparse
171.68.0.91       FastEthernet0      2w6d      00:01:01  Sparse (DR)
171.68.0.82       FastEthernet0      7w0d      00:01:14  Sparse
171.68.0.86       FastEthernet0      7w0d      00:01:13  Sparse
171.68.0.80       FastEthernet0      7w0d      00:01:02  Sparse
171.68.28.70      Serial2.31         22:47:11  00:01:16  Sparse
171.68.28.50      Serial2.33         22:47:22  00:01:08  Sparse
171.68.27.74      Serial2.36         22:47:07  00:01:21  Sparse
171.68.28.170     Serial0.70         1d04h     00:01:06  Sparse
171.68.27.2       Serial1.51         1w4d      00:01:25  Sparse
171.68.28.110     Serial3.56         1d04h     00:01:20  Sparse
171.68.28.58      Serial3.102        12:53:25  00:01:03  Sparse
```

- **Neighbor Address** - the IP address of the PIM Neighbor
- **Interface** - the interface where the PIM Hello of this neighbor was received.
- **Uptime** - the period of time that this PIM Neighbor has been active.
- **Expires** - the period of time after which this PIM Neighbor will no longer be considered as active. (Reset by the receipt of another PIM Query.)
- **Mode** - PIM mode (Sparse, Dense, Sparse/Dense) that the PIM Neighbor is using.
- **“(DR)”** - Indicates that this PIM Neighbor is the Designated Router for the network.
Agenda

- PIM Neighbor Discovery
- **PIM SM State**
  - PIM SM Forwarding
  - PIM SM Joining
  - PIM SM Registering
  - PIM SM SPT-Switchover
  - PIM SM Pruning
  - PIM SM Special Cases
PIM State

• Describes the “state” of the multicast distribution trees as understood by the router at this point in the network.

• Represented by entries in the multicast routing (mroute) table
  – Used to make multicast traffic forwarding decisions
  – Composed of (*, G) and (S, G) entries
  – Each entry contains RPF information
    • Incoming (i.e. RPF) interface
    • RPF Neighbor (upstream)
  – Each entry contains an Outgoing Interface List (OIL)
    • OIL may be NULL

• PIM State
  – In general, Multicast State basically describes the multicast distribution tree as it is understood by the router at this point in the network.
  – However to be completely correct, “Multicast State” describes the multicast traffic “forwarding” state that is used by the router to forward multicast traffic.

• Multicast Routing (mroute) Table
  – Multicast “state” is stored in the multicast routing (mroute) table and which can be displayed using the show ip mroute command.
  – Entries in the mroute table are composed of (*, G) and (S, G) entries each of which contain:
    • RPF Information consisting of an Incoming (or RPF) interface and the IP address of the RPF (i.e. upstream) neighbor router in the direction of the source. (In the case of PIM-SM, this information in a (*, G) entry points toward the RP. PIM-SM will be discussed in a later module.)
    • Outgoing Interface List (OIL) which contains a list of interfaces that the multicast traffic is to be forwarded. (Multicast traffic must arrive on the Incoming interface before it will be forwarded out this interfaces. If multicast traffic does not arrive on the Incoming interface, it is simply discarded.)
• **PIM-SM State Example**

  – **(*, G) Entry** - The (*, 224.1.1.1) entry shown in sample output of the `show ip mroute` command is the (*, G) entry. If there is no matching entry for a particular (S, G) entry, this entry is used to forward traffic down the Shared Tree.

    - The **Expires** countdown timer in the first line of the (*, G) entry which shows when the entry will expire and be deleted. This entry will remain at roughly 3 minutes as long as there is an interface in the Outgoing Interface list.

    - The **Incoming interface** information is used to RPF check arriving (*, G) multicast traffic and is computed in the direction of the RP (in this case, 10.1.5.1). The **Incoming interface** information is used to RPF check arriving (*, G) multicast traffic and is computed in the direction of the RP (in this case, 10.1.5.1).

    - The **Outgoing Interface list** which reflects the interfaces where (*,G) Joins have been received or where directly connected members of group “G” reside. Traffic flowing down the Shared Tree are forwarded out these interfaces. The **Expires** countdown timers on these interfaces are reset to 3 minutes by the receipt of periodic (*, G) Joins. If the count ever reaches zero, the entry in the OIL is deleted.

  – **(S, G) Entry** - The (128.9.160.43/32, 224.1.1.1) entry is an example of an (S, G) entry in the mroute table. This entry is used to forward any multicast traffic sent by source 128.9.160.43 to group 224.1.1.1. Notice the following:

    - The **Expires** countdown timer in the first line of the (S, G) entry which shows when the entry will expire and be deleted. This entry is reset to 3 minutes whenever an (S, G) multicast packet is forwarded.

    - The **Incoming interface** information is used to RPF check arriving (S, G) multicast traffic. If a packet does not arrive via this interface, the packet is discarded.

    - The **Outgoing Interface list** which reflects the interfaces where (S,G) packets are to be forwarded.
• PIM-SM (*,G) State Rules
  – A (*, G) entry is created when a (*, G) Join or an IGMP Report is received
    • The later condition can be simulated by manually configuring the interface to join the group.
  – (*, G) entries are also automatically created whenever an (S, G) entry for the group must be created.
    • The (*, G) entry is created first and then the (S, G) entry. The reason for this will become clear shortly.
  – The IIF reflects the RPF interface/neighbor in the direction of the RP.
  – The OIL of a PIM-SM (*, G) entry reflects interfaces that:
    • Have received a (*, G) Join or
    • Where a directly connected member has joined the group
    • The interface was manually configured to join the group. (Note: This may be accomplished using the `ip igmp static-group <group>` command.)
  – (*, G) entries are deleted when its Expires timer counts down to zero. This will only occur when:
    • The OIL is Null and
    • No child (S, G) entry exists
PIM-SM (S,G) State Rules

- **(S,G) creation**
  - By receipt of (S,G) Join or Prune or
  - By “Register” process
  - Parent (*,G) created (if doesn’t exist)

- **(S,G) reflects forwarding of “S” to “G”**
  - IIF = RPF Interface normally toward source
    - RPF toward RP if “RP-bit” set
  - OIL = Initially, copy of (*,G) OIL minus IIF

- **(S,G) deletion**
  - By normal (S,G) entry timeout

---

- **PIM-SM (S, G) Rules**
  - In PIM-SM, (S, G) state is created as a result of:
    - The receipt of an (S, G) Join or Prune or
    - The PIM-SM Register process which is triggered by a first-hop router receiving a packet from a directly connected source.
  - When an (S, G) entry must be created, the following steps occur:
    - If a corresponding (*, G) entry does not exist, it is created first.
    - The RPF Information is computed for the source “S”. This information is stored in the (S, G) entry as the Incoming interface and the RPF neighbor (i.e. the PIM neighbor in the direction of the source).
      - The exception to this rule is if the RP-bit is set in the (S, G) entry, the RPF interface is pointed up the Shared Tree. This mechanism allows duplicate (S, G) traffic to be blocked from flowing down the Shared Tree after a downstream router has switched to the Shortest Path Tree. (More on this later.)
    - The OIL of the (S, G) entry is populated with a copy of the OIL from the parent (*, G) entry less the Incoming interface. (The Incoming interface must not appear in the OIL otherwise a multicast route loop could occur.)
  - In PIM-SM, (S, G) entries are deleted when their Expires timer counts down to zero. The Expires timer is reset whenever an (S, G) packet is received and forwarded.
PIM-SM OIL Rules

• Interfaces in OIL added
  – By receipt of Join message
    • Interfaces added to (*,G) are added to all (S,G)’s

• Interfaces in OIL removed
  – By receipt of Prune message
    • Interfaces removed from (*,G) are removed from all (S,G)’s
  – Interface Expire timer counts down to zero
    • Timer reset (to 3 min.) by receipt of periodic Join or
    • By IGMP membership report

• PIM-SM Outgoing Interface List Rules
  – Adding an interface
    • Interfaces are added to an (S, G) OIL when a (S, G) Join message is received on an interface.
    • Interfaces are added to the (*, G) OIL when a (*, G) Join message is received on an interface.
    • Anytime an interface is added to the (*, G) OIL, the interface is added to the OIL of all associated (S, G) OIL’s. (Note: A check is always made to prevent the IIF from appearing in the OIL.)
  – Removing an interface
    • Interfaces are removed from the OIL of a (*, G) or (S, G) entry if the interface’s Expires timer counts down to zero.
      Note: The interface Expires timer is reset to 3 minutes by the receipt of periodic Join messages sent by downstream routers once per minute or by an IGMP Report sent by a directly connected member on the interface.
    • Interfaces are removed from the OIL if an Prune message is received (and it is not overridden by another router if the interface is a multi-access network).
    • Interfaces removed from a (*, G) OIL, are removed from the OIL of all associated (S, G) OIL’s.
PIM-SM State Flags

- **S** = Sparse Mode
- **C** = Directly Connected Host
- **L** = Local (Router is member)
- **P** = Pruned (All intfcs in OIL = Prune)
- **T** = Forwarding via SPT
  - Indicates at least one packet was forwarded

**PIM-SM State Flags**

- **“S”** Flag ((*, G) entries only)
  - Indicates the group is operating in Sparse mode. (Appears only on (*, G) entries.)
- **“C”** Flag
  - Indicates that there is a member of the group directly connected to the router.
- **“L”** Flag
  - Indicates the router itself is a member of this group and is receiving the traffic.
  (This would be the case for the Auto-RP Discovery group 224.0.1.40 which all Cisco routers join automatically.)
- **“P”** Flag
  - Set whenever all interfaces in the outgoing interface list of an entry are Pruned (or the list is Null). This general means that the router will send Prune messages to the RPF neighbor to try to shut off this traffic.
- **“T”** Flag ((S, G) entries only)
  - Indicates that at least one packet was received via the SPT
PIM-SM State Flags (cont.)

• J = Join SPT
  – In (*, G) entry
    • Indicates SPT-Threshold is being exceeded
    • Next (S,G) received will trigger join of SPT
  – In (S, G) entry
    • Indicates SPT joined due to SPT-Threshold
    • If rate < SPT-Threshold, switch back to Shared Tree

• F = Register
  – In (S,G) entry
    • “S” is a directly connected source
    • Triggers the Register Process
  – In (*, G) entry
    • Set when “F” set in at least one child (S,G)

• PIM-SM State Flags
  – “J” Flag (Join SPT)
    • When this flag is set in a (*, G) entry, it indicates that the rate of traffic flowing down
      the Shared Tree is above the SPT-Threshold and will cause a switch to the SPT for
      the next packet received down the shared tree. (More on this later.)
    • When this flag is set in an (S, G) entry, it indicates that the (S, G) entry (and hence
      the SPT) was created as a result of the SPT-Threshold being exceeded. If the rate
      of this (S, G) traffic drops back below the SPT, the router will attempt to switch this
      traffic flow back to the Shared Tree.
  – “F” Flag (Register)
    • This flag is set on an (S, G) entry when source “S” is directly connected to the
      router. This indicates that this router is a “first-hop” router and triggers it to send
      Register messages to the RP to inform the RP of this active source.
    • This flag can also be set for arriving (S, G) entries created at a border router such
      as a router that borders on a DVMRP or other dense mode cloud. This causes the
      router to perform a proxy-register operation and send (S, G) Register messages to
      the RP on behalf of the downstream DVMRP routers. This proxy-register operation
      follows the same rules as for directly connected sources.
    • The “F” flag is also set on a (*, G) entry if any associated (S, G) entries have the
      “F” flag set.
PIM-SM State Flags (cont.)

- **R = RP bit**
  - (S, G) entries only
  - Set by (S,G)RP-bit Prune
  - Indicates info is applicable to Shared Tree
  - Used to prune (S,G) traffic from Shared Tree
    - Initiated by Last-hop router after switch to SPT
  - Modifies (S,G) forwarding behavior
    - IIF = RPF toward RP (i.e. up the Shared Tree)
    - OIL = Pruned accordingly

- **PIM-SM Flags**
  - “R” Flag (RP-Bit)
    - This flag is set on (S, G) entries only and indicates that the (S, G) forwarding information in the entry is applicable to (S, G) traffic flowing down the Shared Tree.
    - The “R” flag is set on an (S, G) entry by the receipt of an (S, G)RP-bit Prune message. These messages are sent by downstream routers on the Shared Tree that are requesting that this specific (S, G) traffic flow be pruned off of the Shared Tree. This is done to eliminate duplicate (S, G) traffic after a downstream router has switched to the (S, G) Shortest-Path Tree.
    - Whenever the “R” flag is set on an (S, G) entry, the RPF information must be changed to point toward the RP instead of pointing at source “S”. This is done because the (S, G) entry is now applicable to (S, G) traffic arriving down the Shared Tree. As a result, the RPF information must point up the Shared Tree in order for arriving (S, G) packets to RPF correctly. (This should be made clear later.)
• **X = Proxy Join Timer flag**
  - (S, G) entries only
  - Indicates Proxy Join Timer is running
  - Used to handle turn-around router case
    - More on this in another Module
  - When Proxy Join Timer is running
    - (S, G) Joins are sent toward the source
    - The sending of (S, G) Prunes are suppressed
      - Even if the OIL list is NULL

• **PIM-SM Flags**
  - "X" Flag (Proxy Join Timer Running)
    - This flag is set on (S, G) entries only and is used to indicate that the “Proxy Join Timer” is running. When this timer is running, the router will continue to send (S, G) Joins in the direction of the source even if the OIL is NULL.
    - This is used to handle the special turn-around router situation which occurs when the SPT to the RP and the Shared Tree merge. (More on this special scenario will be presented in another module.)
• **M = MSDP Created bit**
  - (S, G) entries only
  - Set when (S, G) learned via an MSDP SA msg

• **A = MSDP Advertise bit**
  - (S, G) entries only
  - (S, G) *may* be advertised in an MSDP SA msg
    - Presence of certain filters can affect this.
  - Indicates source is in local SM domain
    - Received a PIM (S,G) Register or
    - Source is directly connected or
    - (S,G) traffic was received on a DM interface
      - via the RPF interface

• **PIM-SM Flags**
  - “M” Flag (MSDP Created)
    - This flag only appears on (S, G) entries and only on the router that is the active RP for group “G”.
    - The flag indicates that the RP has learned of this particular source via an MSDP “Source Active” message. (MSDP is addressed in more detail in another module.)
  - “A” Flag (Advertise Flag)
    - This flag only appears on (S, G) entries and only on the router that is the active RP for group “G”.
    - The “A” flag indicates that this source is in the local PIM-SM domain and that it is a candidate for being announced to RP’s in other networks via MSDP “Source Active” messages.
    - A source is considered to be in the local domain if an (S, G) Register message was received for this source or the source is directly connected to the RP or the (S, G) traffic was received on a Dense mode interface that has been designated as a dense mode boundary interface.
Agenda

- PIM Neighbor Discovery
- PIM SM State
- PIM SM Forwarding
- PIM SM Joining
- PIM SM Registering
- PIM SM SPT-Switchover
- PIM SM Pruning
- PIM SM Special Cases
• **PIM SM Forwarding**
  
  - In PIM Sparse mode, incoming multicast packets for group “G” are forwarded out all interfaces that are in the Multicast Route table entry’s “outgoing interface list” (oilist).
  - Once again, an attempt is made to forward using a matching (S,G) entry if one exists. In that case, the “oilist” of the (S,G) entry is used to control forwarding of the multicast packet. If no (S, G) entry exists, then the “oilist” of the (*,G) entry is used to control the forwarding of the multicast packet.

• **Outgoing Interface Lists**
  
  - Interfaces are placed in the (*, G) “oilist” if and only if:
    - The router has received a PIM “Join” for this Group from another PIM Neighbor via this interface or
    - A host on this interface has joined the Group (via IGMP) or
    - The router’s interface has been manually configured to join the group.
Agenda

• PIM Neighbor Discovery
• PIM State
• PIM SM Forwarding
• **PIM SM Joining**
  • PIM SM Registering
  • PIM SM SPT-Switchover
  • PIM SM Pruning
  • PIM SM Special Cases
PIM SM Joining

- **Leaf routers send a (\(^*,G\)) Join to toward RP**
  - Joins sent hop-by-hop along path toward RP
- **Each router along path creates (\(^*,G\)) state**
  - IF no (\(^*,G\)) state,
    - Create it and send a Join toward RP
  - ELSE
    - Join process complete. Reached the Shared Tree.

- **Leaf (last-hop) routers join the Shared Tree (RPT)**
  - When a last-hop router wishes to begin receiving multicast traffic for group “G”, it sends a PIM (\(^*,G\)) Join message to its up-stream PIM Neighbor in the direction of the RP.
  - While the Join is multicast to the “All-Routers” (224.0.0.2) multicast address, the up-stream PIM Neighbor’s IP address is indicated in the body of the PIM Join Message. This permits all PIM routers on a Multi-Access network to be aware of the Join but only the indicated up-stream PIM Neighbor will perform the Join.

- **Routers up the Shared Tree (RPT) create (\(^*,G\)) state**
  - When a PIM router receives a (\(^*,G\)) Join for group “G” from one of its down-stream PIM Neighbors, it will check to see if any (\(^*,G\)) state exists for group “G” in its Multicast Routing table.
    - If (\(^*,G\)) state for group “G” already exists, then the interface from which the Join was received is placed on the (\(^*,G\)) olist.
    - If no (\(^*,G\)) state for group “G” exists, a (\(^*,G\)) entry is created, the interface from which the Join was received is placed in the (\(^*,G\)) olist and a (\(^*,G\)) Join is sent towards the RP.
  - The end result of the above mechanism is to create (\(^*,G\)) state all the way from the last-hop router to the RP so that group “G” multicast traffic will flow down the Shared Tree (RPT) to the last-hop router.
• **PIM SM Joining Example**
  - 1) Receiver “A” wishes to receive group “G” multicast traffic and therefore sends an IGMP Host Membership message (sometimes loosely referred to as an IGMP Join) which is received by “rtr-b”.
  “rtr-b” has no existing (*, G) state for group “G” and therefore creates an entry. (See next slide.)
• State in “rtr-b” after Joining (*, 224.1.1.1)
  – (*, 224.1.1.1)
    • indicates the (*, G) entry.
  – 00:00:05/00:02:54
    • indicates that the entry has existed for 5 seconds and will expire in 2 minutes and
      54 seconds.
  – RP 10.1.5.1
    • is the IP Address of the Rendezvous Point for Group 224.1.1.1
  – flags: SC
    • indicates that this is a Sparse mode group (S) and that there is a member of this
      group directly connected (C) to the router.
  – Incoming interface: Ethernet0, RPF nbr 10.1.2.1
    • indicates the Incoming interface (up the Shared Tree toward RP) and
    • the RPF neighbor’s IP address (in the direction of the RP) is 10.1.2.1
  – Outgoing interface list:
    • lists the interfaces that are in the outgoing interface list for this entry.
  – Ethernet1, Forward/Sparse, 00:00:05/00:02:54
    • indicates Ethernet 1 is in the oillist; it’s in the “Forward” state; Sparse mode and that
      it has been in the list for 5 seconds and will expire in 2 minutes and 54 seconds if
      no further (*, G) Join or IGMP Report is received on this interface.
1. “Rcvr A” wishes to receive group G traffic. Sends IGMP Join for G.
2. “rtr-b” sends (*,G) Join towards RP.

- PIM SM Joining Example
  2) Because the OIL of the (*, G) transitioned from Null to non-Null (when “rtr-b” added Ethernet 1 to the OIL of the newly created entry), a PIM (*, G) Join is sent to rtr-b’s upstream PIM neighbor (rtr-a) in the direction of the RP.
  When “rtr-a” receives the (*, G) Join it creates (*, G) state. (See next slide.)
• **State in “rtr-a” after Joining (*, 224.1.1.1)**
  - (*, 224.1.1.1)
    • indicates the (*, G) entry.
  - 00:00:05/00:02:54
    • indicates that the entry has existed for 5 seconds and will expire in 2 minutes and 54 seconds.
  - **RP 10.1.5.1**
    • is the IP Address of the Rendezvous Point for Group 224.1.1.1
  - **flags: S**
    • indicates that this is a Sparse mode group (S).
  - **Incoming interface: Serial0, RPF nbr 10.1.4.1**
    • indicates the Incoming interface (up the Shared Tree toward RP) and
    the RPF neighbor’s IP address (in the direction of the RP) is 10.1.4.1
  - **Outgoing interface list:**
    • lists the interfaces that are in the outgoing interface list for this entry.
  - **Ethernet0, Forward/Sparse, 00:00:05/00:02:54**
    • indicates Ethernet 0 is in the oillist; it’s in the “Forward” state; Sparse mode and that
    it has been in the list for 5 seconds and will expire in 2 minutes and 54 seconds if
    no further (*, G) Join or IGMP Report is received on this interface.
**PIM SM Joining**

1. “Rcvr A” wishes to receive group G traffic. Sends IGMP Join for G.
2. “rtr-b” sends (*,G) Join towards RP.
3. “rtr-a” sends (*,G) Join towards RP.
4. Shared tree is built all the way back to the RP.

**PIM SM Joining Example**

3) Because the OIL of the (*, G) transitioned from Null to non-Null (when “rtr-a” added Ethernet 0 to the OIL of the newly created entry), a PIM (*, G) Join is sent to rtr-a’s up-stream PIM neighbor in the direction of the RP.

When the upstream router receives the (*, G) Join it too creates (*, G) state and creates a branch of the Shared Tree.

4) This process continues all the way back to the RP (or until a router is reached that is already on the Shared Tree and therefore already has a (*, G) entry.)
Agenda

- PIM Neighbor Discovery
- PIM SM State
- PIM SM Forwarding
- PIM SM Joining
- **PIM SM Registering**
- PIM SM SPT-Switchover
- PIM SM Pruning
- PIM SM Special Cases
PIM SM Registering

- Senders begin sourcing Multicast Traffic
  - Senders don’t necessarily perform IGMP group joins.

- 1st-hop router unicasts “Registers” to RP
  - A Mcast packet is encapsulated in each Register msg
  - Registers messages follow unicast path to RP

- RP receives “Register” messages
  - De-encapsulates Mcast packet inside Register msg
  - Forwards Mcast packet down Shared Tree
  - Sends (S,G) Join toward Source / 1st-Hop router to build an (S,G) SPT between Source and RP

- All Senders are not necessarily Receivers and vice versa.
  - It is not a requirement that all sources be receivers. In the case of a source-only host, it is permissible for the host to simply begin sending multicast traffic without ever joining the group via IGMP.

- 1st-hop router sends “Registers” to the RP
  - In PIM Sparse mode, when a 1st-hop router receives the first multicast packet from directly connected source “S” for group “G”, it creates (S, G) state and sets the “F” bit in the (S, G) entry to indicate that it is a directly connected “Source” and also sets the “Registering” flag to indicate that it’s in the process of “Registering”.
  - Next, the 1st-hop router encapsulates the original multicast packet in a PIM Register message and unicasts it to the RP. (Any subsequent multicast packets received from directly connected source “S” for group “G” are also encapsulated in a Register message and unicasts to the RP. This continues until an (S, G) “Register-Stop” message is received from the RP.)

- RP receives “Register messages
  - When the RP receives a “Register” message it will de-encapsulated the message. If this packet is to a Group for which the RP has (*, G) state, the RP will:
    • Forward the original packet out all interfaces in the the (*, G) entry’s “oilist”.
    • If it hasn’t already done so, the RP creates (S, G) state and sends an (S, G) Join back towards the Source in order to join the Shortest-path Tree (SPT) to Source “S”.

Copyright © 1999, 2000, Cisco Systems, Inc.

Module3.ppt

Cisco.com
**PIM SM Registering (cont.)**

- **1st-hop router receives (S,G) Join**
  - SPT between Source and RP now built.
  - Begins forwarding traffic down (S,G) SPT to RP
  - (S,G) Traffic temporarily flowing down 2 paths to RP

- **RP receives traffic down native (S,G) SPT**
  - Sends a “Register-Stop” msg to Source / 1st-Hop router.

- **1st-hop router receives “Register-Stop”**
  - Stops encapsulating traffic in “Register” messages
  - (S,G) Traffic now flowing down single SPT to RP

- **1st-hop router receives (S, G) Join**
  - When the 1st-hop router receives the (S, G) Join (sent hop-by-hop from the RP), it processes it normally by adding the interface, from which the Join was received, to the “oilst” of the existing (S, G) entry. (This entry was originally created when the 1st-hop router received the first multicast packet from directly connected Source “S”.) This completes the building of the Shortest-Path Tree (SPT) from the Source to the RP.
  - The 1st-hop router now begins forwarding Source “S” multicast traffic down the newly built Shortest-Path Tree (SPT) to the RP.
  - Note: (S, G) traffic temporarily flows to the RP via two methods; via Register messages (until a Register-Stop message is received) and the “native” Shortest-Path Tree (SPT).

- **RP begins receiving traffic down the (S, G) SPT.**
  - As soon as the RP begins receiving (S, G) traffic “natively” (i.e. not encapsulated in Register messages) down the SPT, the RP will set the “T” bit in the (S, G) entry to denote that traffic is successfully flowing down the Shortest-Path Tree (SPT).
  - Now when any (S, G) Register messages are received by the RP, it sees that the “T” bit is set in the (S, G) entry will respond by sending a PIM (S, G) Register-Stop message to the 1st-hop router. This notifies the 1st-hop router that traffic is now being received “natively” down the SPT.

- **1st-hop router receives “Register-Stop” message**
  - When the (S, G) Register-Stop message is received by the 1st-hop router, it clears the “Registering” flag in the (S, G) entry and stops encapsulating (S,G) traffic in Register messages. Traffic is now flowing only down the SPT to the RP.
PIM SM Register Examples

• Receivers Join Group First
  • Source Registers First
  • Receivers along the SPT

• PIM SM Register Examples
  – Depending on whether there are any existing Receivers for group “G” on the Shared Tree (RPT), the RP hands the Register process a little different.

  In the following examples we will consider the Register process for the cases when:
  • Receivers join group “G” first;
  • The Source Registers first.
  • Receivers along the SPT.
• **State in RP before Registering (Rcvr’s on Shared Tree)**
  - Pay particular attention to the following in the (*, G) entry:
    - The “Incoming interface:” is NULL and the “RPF nbr” is 0.0.0.0. This indicates that this router is the RP.
    - The “Outgoing interface list:” contains Serial0 and Serial1 which are assumed to be the only two active branches of the Shared Tree (RPT).
• **State in “rtr-b” before source registers**
  – Note that there is no group state information for this Group yet.
- State in 1st-hop router (rtr-a) before source registers
  - Note that there is no group state information for this Group yet.

```
rtr-a>sh ip mroute 224.1.1.1
No such group.
```
“Source” begins sending group G traffic.

• Receivers Join Group First Example
  1) Source “S” begins sending traffic to group “G”.
  2) 1st-hop router (“rtr-a”) creates (*, G) and (S, G) state; encapsulates the multicast packets in PIM Register message(s) and unicasts it(them) to the RP.
  3) The RP (“rtr-c”) de-encapsulates the packets and sees that the packet is for group “G” for which it already has (*, G) state. It then forwards the packets down the Shared Tree.
**PIM SM Registering**

**Receiver Joins Group First**

1. “Source” begins sending group G traffic.
2. “rtr-a” encapsulates packets in Registers; unicasts to RP.

**1st-hop router (rtr-a) creates (S, G) state**

- A (*, G) entry must be created before the (S, G) entry can be created. Note that:
  - The RPF information for this entry points up the Shared Tree via Serial0 with the RPF neighbor of 171.68.28.191. (Serial 0 of “rtr-b”.)
  - Because in this example no members have joined the group (the sender is only sending), the OIL of the (*, G) entry is Null.
  - The “P” flag (Pruned) is set since the OIL is Null.
- The (S, G) entry is then created. Pay particular attention to the following:
  - The RPF information for this entry points towards the source via Ethernet0. The RPF neighbor is 0.0.0.0 because the source is directly connected.
  - The (S, G) OIL receives a copy of the (*, G) OIL. (Which is Null.)
  - The “F” flags are set in the (S, G) entry which indicates that this is a directly connected Source.
  - The “Registering” flag is set in the (S, G) entry which indicates that we are still sending Register messages to the RP for this Source.
  - The “P” flag (Pruned) is set since the OIL is Null.

2) The 1st-hop router encapsulates the multicast packets in PIM Register message(s) and unicasts them to the RP.
• The RP creates \((S, G)\) state
  – As a result of the Register message that was received from “rtr-a”, the RP creates \((S, G)\) state as follows:
    • The RPF information is calculated using the source address contained in the multicast packet encapsulated inside of the register message. This results in an IIF of Serial3 and an RPF neighbor of 171.68.28.139.
    • Next, the OIL of the parent \((*, G)\) entry is copied into the OIL of the new \((S,G)\) entry. (An additional check is made to insure that the IIF does not appear in the OIL. If it does, it is removed to prevent a route loop.)
  – Now the router is ready to forward the \((S, G)\) packet that was encapsulated in the Register message using the newly created \((S, G)\) state. (Note that traffic is always forwarded using the matching \((S, G)\) entry if one exists. Otherwise, the \((*, G)\) entry is used.) This is accomplished as follows:
    • Because this packet was received inside of a Register message, the RPF check is skipped.
    • Next, the router forwards a copy of the packet out all interfaces in the \((S, G)\) OIL. In this case a copy is sent out Serial0 and Serial1 which corresponds to the two branches of the Shared Tree.
    • The “T” flag is not yet set in the \((S, G)\) entry. However, when the first \((S, G)\) packet is received natively (via the Incoming interface) and forwarded using this entry, the “T” flag will be set.
RP sends (S,G) Join toward Source to build SPT.

Receivers Join Group First Example (cont.)

4) Because RP has existing (*., G) state (i.e. Receivers already waiting on the Shared Tree), it sends an (S, G) Join toward source “S” to build a Shortest-Path Tree (SPT) from source “S” to the RP.
• “rtr-b” processes the (S, G) Join and creates state
  – A (*, G) entry must be created before the (S, G) entry can be created. Note that:
    • The RPF information for the (*, G) entry points up the Shared Tree via Serial1 with
      the RPF neighbor of 171.68.28.140. (Serial 3 of the RP.)
    • Because in this example no members have joined the group, the OIL of the (*, G)
      entry is Null.
    • The “P” flag (Pruned) is set since the OIL is Null.
  – The (S, G) entry is then created. Pay particular attention to the following:
    • The RPF information for this entry points towards the source via Serial0. The RPF
      neighbor is 171.68.28.190. (Serial 0 of “rtr-a”.)
    • The (S, G) OIL initially receives a copy of the (*, G) OIL. (Which is Null.)
    • Interface Serial1 (which is the interface that received the (S, G) Join) is added to
      the (S, G) OIL.
    • The “T” flag is not yet set in the (S, G) entry. However, when the first (S, G) packet
      is forwarded using this entry, the flag will be “T” set.
  5) Because the OIL of the (S, G) transitioned from Null to non-Null (when “rtr-b” added
     Serial1 to the OIL of the newly created entry), a PIM (S, G) Join is sent to rtr-a’s to
     continue the process of joining the SPT.
"rtr-a" processes the (S, G) Join; adds Serial0 to OIL

- Because an (S, G) entry already existed, "rtr-a" simply added the interface on which it received the (S, G) join to the OIL. This results in the following:
  - Serial0 is now listed in the “Outgoing interface list” (OIL) since the RP joined the SPT via this interface.
  - The “P” flag (Pruned) is cleared since the OIL is no longer Null.
RP begins receiving (S,G) traffic down SPT.

RP sends “Register-Stop” to “rtr-a”.

- **A branch of the (S,G) SPT has been built to the RP.**
  
  6) Now that the SPT has been built from source “S” to the RP, traffic begins flowing down the Shortest-Path Tree (SPT). At this point, the RP is receiving the (S, G) traffic “natively” down the SPT. (This causes the “T” flags to be set in the (S, G) entries along this path including in the RP.)

  7) The RP then sends an (S, G) “Register-Stop” to the 1st-hop router to inform it that the encapsulated group “G” Register messages from source “S” are no longer necessary.
• “rtr-a” stops sending Register messages
  – When the 1st-hop router (rtr-a) receives the (S, G) Register-Stop message it ceases sending encapsulated Register messages for (S, G) traffic.
    • Notice that the “Registering” flag on the second line of the (S, G) entry is no longer being displayed indicating that “rtr-a” is not sending Registers.
    • This is the final state in “rtr-a” after the Registration process.

8) (S, G) traffic is now only flowing down the Shortest-Path Tree (SPT).
• **Final state in “rtr-b” after the Registration process**
  – Pay particular attention to the following in the (S, G) entry:
    • The “T” flag is now set indicating that (S, G) traffic is flowing along this path.
    • The (*, G) entry still has a Null OIL and the “P” flag is still set.
    • This is because there are no members that have joined the Shared Tree.

**Diagram:**

1. **Mcast Packets**
   - Source: 171.68.37.121
   - RP: 171.68.28.140

2. **Shared Tree**
   - (*, 224.1.1.1)
   - Mcast Traffic
   - (171.68.37.121, 224.1.1.1)

3. **Receiver Joins Group First**
   - PIM SM Registering
   - S0 S1

4. **(E0)**
   - rtr-a
   - Serial0, Forward/Sparse, 00:04:28/00:01:32

5. **rtr-b**
   - (*, 224.1.1.1), 00:04:28/00:01:32, RP 171.68.28.140, flags: SP
   - Incoming interface: Serial1, RPF nbr 171.68.28.140,
   - Outgoing interface list: Null

6. **S0 S1**
   - (*, 224.1.1.1), 00:04:28/00:01:32, flags: T
   - Incoming interface: Serial1, RPF nbr 171.68.28.190
   - Outgoing interface list:
     - Serial1, Forward/Sparse, 00:04:28/00:01:32
Final state in the RP after the Registration process

- Pay particular attention to the following in the newly created (S, G) entry:
  - The “T” flag is now set indicating that (S, G) traffic is flowing along this path.
PIM SM Register Examples

• Receivers Join Group First
• Source Registers First
• Receivers along the SPT

• PIM SM Register Examples
  – Depending on whether there are any existing Receivers for group “G” on the Shared Tree (RPT), the RP hands the Register process a little different.

In the following examples we will consider the Register process for the cases when:
  • Receivers join group “G” first;
  • The Source Registers first.
• State in RP before Registering (w/o Rcvr’s on Shared Tree)
  – Notice that no state for group “G” exists since there are no Receivers on the Shared Tree yet.
• **State in “rtr-b” before source registers**
  – Note that there is no group state information for this Group yet.
• State in 1st-hop router (rtr-a) before source registers
  – Note that there is no group state information for this Group yet.
• Source Registers First Example

1) Source “S” begins sending traffic to group “G”.
2) 1st-hop router (“rtr-a”) creates (*, G) and (S, G) state; encapsulates the multicast packets in PIM Register message(s) and unicasts it(them) to the RP.
3) The RP (“rtr-c”) de-encapsulates the (S, G) packet and creates (*, G) and (S, G) state. Since no one has joined the Shared Tree yet, the OIL’s of these entries will be NULL. Because the OIL of the (S, G) entry (just created) is NULL, the packet is discarded.
**1st-hop router (rtr-a) creates (S, G) state**

- A (*, G) entry must be created before the (S, G) entry can be created. Note that:
  - The RPF information for this entry points up the Shared Tree via Serial0 with the RPF neighbor of 171.68.28.191. (Serial 0 of “rtr-b”.)
  - Because in this example no members have joined the group (the sender is only sending), the OIL of the (*, G) entry is Null.
  - The "P" flag (Pruned) is set since the OIL is Null.
- The (S, G) entry is then created. Pay particular attention to the following:
  - The RPF information for this entry points towards the source via Ethernet0. The RPF neighbor is 0.0.0.0 because the source is directly connected.
  - The (S, G) OIL receives a copy of the (*, G) OIL. (Which is Null.)
  - The "F" flags are set in the (S, G) entry which indicates that this is a directly connected Source.
  - The "Registering" flag is set in the (S, G) entry which indicates that we are still sending Register messages to the RP for this Source.
  - The "P" flag (Pruned) is set since the OIL is Null.

2) The 1st-hop router encapsulates the multicast packets in PIM Register message(s) and unicasts them to the RP.
• The RP creates (S, G) state
  – As a result of the Register message that was received from “rtr-a”, the RP creates (*, G) and (S, G) state. However, because no previous (*, G) state existed, it must be created before the (S,G) entry can be created.
    • This (*, G) entry is created as shown above. Notice that the (*, G) OIL is NULL. This is because the RP has not yet received any (*, G) Joins for this group. (Remember, in this example, the source registers first.)
  – Next, the (S, G) entry can be created and is accomplished as follows:
    • The RPF information is calculated using the source address contained in the multicast packet encapsulated inside of the register message. This results in an IIF of Serial3 and an RPF neighbor of 171.68.28.139.
    • Next, the OIL of the parent (*, G) entry is copied into the OIL of the new (S,G) entry. Since the OIL of the (*, G) entry is NULL, this results in a NULL (S, G) OIL.
  – Now the router is ready to forward the (S, G) packet that was encapsulated in the Register message using the newly created (S, G) state. This is accomplished as follows:
    • Because this packet was received inside of a Register message, the RPF check is skipped.
    • Next, the router forwards a copy of the packet out all interfaces in the matching (S, G) OIL. However, because the (S, G) OIL is NULL (i.e. there are no branches of the Shared Tree), the packet is simply discarded.
4) Since the RP has no (*, G) state and hence no receivers on the Shared Tree, it does not need the (S, G) traffic. Therefore the RP sends an (S, G) "Register-Stop" message to the 1st-hop router so it will stop sending Register messages.

- **Source Registers First Example**
  
  3 “rtr-c” (RP) has no receivers on Shared Tree; discards packet.
  4 RP sends “Register-Stop” to “rtr-a”.

---

Copyright © 1999, 2000, Cisco Systems, Inc.
• Source Registers First Example

5) The 1st-hop router receives the (S, G) Register-Stop message and stops sending Register messages for (S, G) traffic.

Note: Eventually, the original (S, G) entry will time out (approx. 3 min.) and be deleted. The Register process will start over again when the 1st-hop router receives the next multicast packet from directly connected source “S”. The RP will again respond with a Register-Stop which will prevent the (S,G) traffic from flowing to the RP until it is needed.
State in 1st-hop router after Registering (w/o Rcvr’s on Shared Tree)

- Pay particular attention to the following in the (S, G) entry:
  - The “Registering” flag is now cleared.
  - The “Outgoing interface list” is still Null since the RP did not join the SPT.
  - The “P” flag (Pruned) is still set since the oilist is still Null.
  - The “00:01:32” Expiration time value will count down to zero at which time the (S, G) entry will be deleted. (The Register process will begin all over again when the next multicast packet is received from source “S”.)
• State in “rtr-b” after Registering (w/o Rcvr’s on Shared Tree)
  – Notice that no state exists in “rtr-b” at this point in time.
State in RP after Registering (w/o Rcvr’s on Shared Tree)

- Pay particular attention to the following in the newly created (S, G) entry:
  - The “RPF nbr” is the IP Address of “rtr-b”.
  - The “Incoming interface:” is Serial3 which is the RPF interface towards source “S” via “rtr-b”.
  - The “Outgoing interface list:” is Null since the (*,G) OIL is also Null. (Indicates there are no Receivers on the Shared Tree yet.)
  - The “P” flag (Pruned) is set since the OIL is Null.
- The (S,G) state will remain in the RP as long as the source is still actively sending. This is accomplished by fact that the first-hop route will continue sending periodic Register messages to the RP as long as the first-hop router is receiving traffic from the source.
• Source Registers First Example

6) The RP now begins receiving (*, G) Joins from Last-hop routers with Receivers that wish to join the Shared Tree.
• The RP process the (*, G) Join
  – In the (*, G) entry:
    • **Serial1** has been added to the (*, G) entry since a (*,G) Join was received on this interface which is the only active branch of the Shared Tree (RPT).
  – In the (S, G) entry:
    • **Serial1** has also been added to the (S, G) OIL because the OIL’s of all (S,G) entries are always kept in sync with their parent (*, G).
      Note: When the (S, G) OIL’s are synchronized with the OIL of their parent (*.G) OIL, a check is made to insure that the IIF of the (S, G) does not appear in the OIL of the (S, G). This could result in a route loop.

7) The transitioning of the (*, G) OIL from Null to non-Null triggers the RP to scan its list of (S, G) entries for group “G” and send (S, G) Joins towards all sources. (This will cause a SPT to be built from each active source back to the RP which will eventually start the flow of (S, G) traffic to the RP.)
“rtr-b” processes the (S, G) Join and creates state
- A (*, G) entry must be created before the (S, G) entry can be created. Note that:
  - The RPF information for the (*, G) entry points up the Shared Tree via Serial1 with the RPF neighbor of 171.68.28.140. (Serial 3 of the RP.)
  - Because in this example no members have joined the group, the OIL of the (*, G) entry is Null.
  - The “P” flag is set since the OIL is Null.
- The (S, G) entry is then created. Pay particular attention to the following:
  - The RPF information for this entry points towards the source via Serial0. The RPF neighbor is 171.68.28.190. (Serial 0 of “rtr-a”).
  - The (S, G) OIL initially receives a copy of the (*, G) OIL. (Which is Null.)
  - Interface Serial1 (which is the interface that received the (S, G) Join) is added to the (S, G) OIL.
  - The “T” flag is not yet set in the (S, G) entry. However, when the first (S, G) packet is forwarded using this entry, the flag will be “T” set.
- Because the OIL of the (S, G) transitioned from Null to non-Null (when “rtr-b” added Serial1 to the OIL of the newly created entry), a PIM (S, G) Join is sent to rtr-a’s to continue the process of joining the SPT.
• 1st-hop router (rtr-a) processes the (S, G) Join
  – The (S, G) Join is processed as follows:
    • Serial0 is added to the “Outgoing interface list” (OIL). (This is the interface on which the (S, G) Join arrived.)
    • The “P” flag (Pruned) is cleared since the OIL is no longer Null.
9) As a result of Serial0 being added to the (S, G) OIL, traffic begins to flow down the SPT from the source to the RP.
10) The RP then forwards all incoming (S, G) traffic to the Receivers down the Shared Tree.
State in “rtr-b” after Receivers Join
- Pay particular attention to the following:
  - Both (*, G) and (S, G) state was created as a result of the (S, G) Join received from the RP.
  - The “P” flag set in the (*, G) entry since there are no receivers on the Shared Tree at this point in the network.
  - The “T” flag is set in the (S, G) entry indicating that traffic is flowing down the Shortest-Path Tree.
  - The “RPF nbr” is the IP Address of “rtr-a”.
  - **Serial0** is the “Incoming interface” of the (S, G) entry since this is the RPF interface for source “S” via “rtr-a”.
  - **Serial1** is listed in the “Outgoing interface list” of the (S, G) entry since the RP joined the SPT via this interface.
State in RP after Receivers Join

- In the (*, G) entry:
  - **Serial1** has been added to the (*, G) entry since a (*,G) Join was received on this interface which is the only active branch of the Shared Tree (RPT).

- In the (S, G) entry:
  - **Serial1** has also been added to the (S, G) OIL because the OIL’s of all (S,G) entries are always kept in sync with their parent (*, G).

  Note: When the (S, G) OIL’s are synchronized with the OIL of their parent (*, G) OIL, a check is made to insure that the IIF of the (S, G) does not appear in the OIL of the (S, G). This could result in a route loop.
• PIM SM Register Examples
  – Depending on whether there are any existing Receivers for group “G” on the Shared Tree (RPT), the RP hands the Register process a little different.

    In the following examples we will consider the Register process for the cases when:
    • Receivers join group “G” first;
    • The Source Registers first.
    • Receivers along the SPT.
**State in “rtr-b” with traffic flowing on the SPT**

- Pay particular attention to the following:
  - Both (*, G) and (S, G) state was created as a result of the (S, G) Join received from the RP.
  - The “P” flag set in the (*, G) entry since there are no receivers on the Shared Tree at this point in the network.
  - The “T” flag is set in the (S, G) entry indicating that traffic is flowing down the Shortest-Path Tree.
  - The “RPF nbr” is the IP Address of “rtr-a”.
  - **Serial0** is the “Incoming interface” of the (S, G) entry since this is the RPF interface for source “S” via “rtr-a”.
  - **Serial1** is listed in the “Outgoing interface list” of the (S, G) entry since the RP joined the SPT via this interface.
• **State in the RP with traffic flowing on the SPT**
  
  – Pay particular attention to the following:
  
  • The (*, G) entry only has **Serial1** in its outgoing interface list.
  
  • In the (S, G) entry, **Serial0** is the **“Incoming interface”** since this is the RPF interface for source “S” via “rtr-b”.

  • **Serial1** is listed in the **“Outgoing interface list”** of the (S, G) entry because the OIL of the (S, G) entry is always kept in sync with the (*, G) OIL.
“Rcvr A” wishes to receive group G traffic. Sends IGMP Join for G.

- Receivers along the SPT
  - Step 1: A host directly connected to “rtr-b”, Receiver “A”, joins multicast group 224.1.1.1 by sending an IGMP Report.
Receivers along the SPT

- As a result of the IGMP Report sent by Receiver A for group 224.1.1.1, “rtr-b” updates its state for this group as follows.
  - Ethernet0 is added to the OIL of the (*, G) entry. This is done to permit any (*, 224.1.1.1) traffic flowing down the Shared Tree to be forwarded to Receiver “A”.
  - Next, the OIL’s of all child (S, G) entries are synchronized with the OIL change just made to the OIL of the (*, G). This results in Ethernet0 being added to the OIL of the (171.68.37.121/32, 224.1.1.1) entry. This permits traffic from this source to be “picked off” as it flows along the SPT through “rtr-b” on its way to the RP. (Note that this traffic does not flow to the RP and then back out the same interface to reach “rtr-b”. This is a common misperception.)
**PIM SM Registering**

**Receivers along the SPT**

- **Step 2:** Because the OIL of the (*, G) entry in “rtr-b” transitioned from NULL to non-Null (Ethernet0 is now in the (*, G) OIL), a (*, G) Join message is triggered. This message is sent up the Shared Tree so that “rtr-b” will be placed on a branch of the Shared Tree.
• Receivers along the SPT
  – When the RP receives the (*, G) Join sent by “rtr-b”, it adds Serial3 to the (*, G) OIL.
  – Next, the RP synchronizes the OIL’s of all (S, G) entries by adding Serial3 to each (S, G) OIL. However in this case, Serial3 is the Incoming interface for the (171.68.37.121/32, 224.1.1.1) entry and is therefore not added to the OIL. (If it were, Serial3 would appear in both the incoming and outgoing interface list which could cause a route loop.)
3 Group G traffic begins to flow to “Rcvr A”.
(Note: 171.68.37.121 traffic doesn’t flow to RP then back down to rtr-b)

- Receivers along the SPT
  - **Step 3:** Traffic from source 171.68.37.121 is now being “picked off” by “rtr-b” and forwarded out Ethernet0 as the traffic flows down the SPT to the RP.
  - Again, it is important to note that this source traffic does not flow to the RP and then turn around and come back out on the same interface that it arrived. (Refer to the state in the RP shown on the previous page.)
Agenda

• PIM Neighbor Discovery
• PIM SM State
• PIM SM Forwarding
• PIM SM Joining
• PIM SM Registering
  • PIM SM SPT-Switchover
• PIM SM Pruning
• PIM SM Special Cases
• SPT Thresholds may be set for any Group
  – Access Lists may be used to specify which Groups
  – Default Threshold = 0kbps (i.e. immediately join SPT)
  – Threshold = “infinity” means “never join SPT”.
• Threshold triggers Join of Source Tree
  – Sends an (S,G) Join up SPT for next “S” in “G” packet received.
• Pros
  – Reduces Network Latency
• Cons
  – More (S,G) state must be stored in the routers.

• SPT Thresholds
  – In PIM Sparse mode, SPT Thresholds may be configured to control when to switch to the Shortest-Path Tree (SPT).
  – SPT Thresholds are specified in Kbps and can be used with Access List to specify to which Group(s) the Threshold applies.
  – The default SPT-Threshold is 0Kbps. This means that any and all sources are immediately switched to the Shortest-Path Tree.
  – If an SPT-Threshold of “Infinity” is specified for a group, the sources will not be switched to the Shortest-Path Tree (SPT) and will remain on the Shared Tree.

• Exceeding the Threshold
  – When the Group’s SPT-Threshold is exceed in a last-hop router, the next received packet for the group will cause an (S, G) Join to be sent toward the source of the packet. This builds a Shortest-Path Tree from the source “S” to the last-hop router.

• PROS
  – By switching to the Shortest-Path Tree (SPT), the most optimal (usually) path is used to deliver the multicast traffic. Depending on the location of the source in relation to the RP, this switch to the SPT can reduce network latency substantially.

• CONS
  – In networks with large numbers of senders (remember most multicast applications such as IP/TV Client, send RTCP multicast packets in the background and are therefore senders), an increased amount of state must be kept in the routers. In some cases, an Infinity threshold may be used to force certain groups to remain on the Shared Tree when latency is not an issue.
PIM SM SPT-Switchover

SPT-Switchover Mechanism

Once each second
- Compute new (∗, G) traffic rate
- If threshold exceeded, set “J” flag in (∗, G)

For each (Si, G) packet received:
- If “J” flag set in (∗, G)
  • Join SPT for (Si, G)
  • Mark (Si, G) entry with “J” flag
  • Clear “J” flag in (∗, G)

• SPT-Threshold Myth
  - This is a frequently misunderstood mechanism. Many people think that the the traffic rates of the sources in the group are monitored and compared against the SPT-Threshold. THIS IS NOT THE CASE. Instead, the total aggregate rate of Group traffic flowing down the Shared Tree (RPT) is calculated once per second. If this total aggregate rate is exceed, then the next Group packet received causes that source to be switched to the Shortest-Path Tree (SPT).

• SPT-Switchover Mechanism
  - Once each second, the aggregate (∗, G) traffic rate is computed and checked against the SPT-Threshold. If the aggregate rate of all group traffic flowing down the Shared Tree (RPT) exceeds the threshold, then the “J” flag is set in the (∗, G) entry.
  - As each multicast packet is received on the Shared Tree, the “J” bit is checked in the (∗, G) entry.
    • If the “J” flag is set, a new (S, G) entry is created for the source of the packet.
    • An (S, G) Join is sent towards the source in order to join the SPT.
    • The “J” flag is set in the (S, G) entry to denote that this entry was created as a result of the SPT-Threshold switchover.
    • The “J” flag in the (∗, G) is reset. (It will be set in one second if the aggregate rate on the Shared Tree is still over the SPT-Threshold.)
  - This mechanism can sometimes result in low rate sources being switched to the SPT erroneously. However, the RPT-switchback mechanism will correct this situation and eventually only the high rate sources will be received via SPTs while low rate sources will remain on the Shared Tree.
• PIM-SM SPT-Switchover Example
  – Receivers A & B have joined multicast group 224.1.1.1 which has resulted in traffic flowing down the Shared Tree as shown by the solid arrows.
  – The state is “rtr-c” prior to the switchover is as follows:
    • The IIF of the (*, G) entry points toward the RP via Serial0.
    • The OIL of the (*, G) entry contains Serial1 and Serial2 as a result of (*, G) Joins that were sent up the Shared Tree by “rtr-a” and “rtr-d”, respectively.
• **PIM-SM SPT-Switchover Example**
  – The state is “rtr-d” prior to the switchover is as follows:
    • The IIF of the (*, G) entry points toward the RP via Serial0.
    • The OIL of the (*, G) entry contains Ethernet0 as a result of the IGMP Reports for group 224.1.1.1 that are sent by Receiver “B”.

---

*S, G* Traffic Flow

- **Shared (RPT) Tree**
- **SPT Tree**

**State in “rtr-d” before switch**

(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: SC
Incoming interface: Serial0, RPF nbr 10.1.4.8,
Outgoing interface list:
  Ethernet0, Forward/Sparse, 00:01:43/00:02:11
• **PIM-SM SPT-Switchover Example**
  - The state is “rtr-a” prior to the switchover is as follows:
    - The IIF of the (*, G) entry points toward the RP via Serial0.
    - The OIL of the (*, G) entry contains Ethernet0 as a result of (*, G) Joins that were sent up the Shared Tree by “rtr-b”.

(S, G) Traffic Flow
- Shared (RPT) Tree
- SPT Tree
• **PIM-SM SPT-Switchover Example**
  – The state is “rtr-b” prior to the switchover is as follows:
    • The IIF of the (\(*, G\)) entry points toward the RP via Ethernet0.
    • The OIL of the (\(*, G\)) entry contains Ethernet1 as a result of the IGMP Reports for group 224.1.1.1 that are sent by Receiver “A”.

\[
\text{State in “rtr-b” before switch}
\]

\[
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: SC
\]

\[
\text{Incoming interface: Ethernet0, RPF nbr 10.1.2.1,}
\]

\[
\text{Outgoing interface list:}
\]

\[
\text{Ethernet1, Forward/Sparse, 00:01:43/00:02:11}
\]
**PIM SM SPT-Switchover**

1. Group “G” rate exceeds SPT Threshold at “rtr-b”;
2. Set J Flag in (*) G) and wait for next (S; G) packet.

**PIM-SM SPT-Switchover Example**

- Step 1: The total amount of all traffic flowing down the Shared Tree begins to exceed the SPT-Threshold configured at “rtr-b”.
- Step 2: As a result, “rtr-b” sets the “J” flag in the (*, G) entry to denote that the rate is above the SPT-Threshold for this group.
### PIM SM SPT-Switchover Example

- **Step 3:** The very next packet to arrive via the Shared Tree happens to be from source \((S_i, G)\). Because there is a member directly connected to this router (denoted by the “C” flag) and the traffic rate is above the SPT-Threshold (denoted by the “J” flag), “rtr-b” initiates a switch to the SPT for \((S_i, G)\) traffic.

- **Step 4:** The “J” flag in the \((*, G)\) entry is first cleared and an new traffic rate measurement interval (1 second) is started. Next, \((S_i, G)\) state is created for source “Si” sending to group “G”.

---

(S,G) packet arrives down Shared tree.
1. Clear J Flag in the (*,G) & create \((S_i,G)\) state.
**PIM SM SPT-Switchover Example**

- The `(171.68.37.121/32, 224.1.1.1)` entry shown above is created as follows:
  - To denote that this entry was created as a result of the SPT Switchover mechanism, the “J” flag is set on the (S, G) entry. (The “J” flag being set will cause “rtr-b” to monitor the rate of the (S, G) traffic and if the rate of this traffic drops below the SPT Threshold for over one minute, “rtr-b” will attempt to switch this traffic flow back to the Shared Tree.)
  - The RPF information is calculated in the direction of source “S_i”. This results in an Incoming interface of `Ethernet0`, and an RPF neighbor address of `10.1.2.1`. (Note: That the RPF information for the (S, G) entry is the same as the (*, G) entry. This indicates that the Shared Tree and the SPT are following the same path at this point.)
  - The OIL for the (S, G) entry is constructed by copying the OIL from the (*, G) entry and then removing the IIF from this list to prevent a possible route loop. This results in an (S, G) OIL containing only `Ethernet1`. 
PIM SM SPT-Switchover Example

- Step 5: Once state has been created for (Si, G), an (S, G) Join is sent toward source “Si” to build a branch of the SPT to “rtr-b”. These (Si, G) Joins will continue to be sent periodically (once a minute) as long as the (Si, G) entry is not Pruned (i.e. does not have a Null OIL).
**PIM SM SPT-Switchover**

- **PIM-SM SPT-Switchover Example**
  - When the (Si, G) Join is received by “rtr-a”, the (171.68.37.121/32, 224.1.1.1) entry shown above is created as follows:
    - The RPF information is calculated in the direction of source “Si”. This results in an Incoming interface of Serial1, and an RPF neighbor address of “10.1.9.1.”. (Note: That the RPF information for the (S, G) entry is not the same as the (*, G) entry. This indicates that the paths of the Shared Tree and the SPT diverge at this point.)
    - The OIL for the (S, G) entry is constructed by copying the OIL from the (*, G) entry and then removing the IIF from this list to prevent a possible route loop. This results in an (S, G) OIL containing only Ethernet0.


**PIM SM SPT-Switchover**

- **Step 6**: When the \((S_i, G)\) state is created at “rtr-a”, an \((S_i, G)\) Join is sent toward source “\(S_i\)”. These \((S_i, G)\) Joins will continue to be sent periodically (once a minute) as long as the \((S_i, G)\) entry is not Pruned (i.e. does not have a Null OIL).
- **Step 7**: Because the paths of the Shared Tree and the SPT diverge at “rtr-a”, (note the difference in RPF information on the previous page), this causes “rtr-a” to begin sending \((S_i, G)\)RP-bit Prune messages up the Shared Tree to stop the flow of redundant \((S_i, G)\) traffic down the Shared Tree.

**PIM-SM SPT-Switchover Example**

- “rtr-a” forwards \((S_i, G)\) Join toward \(S_i\).
- SPT & RPT diverge, triggering \((S_i, G)\)RP-bit Prunes toward RP.
(S_i, G) traffic begins flowing down SPT tree.

- PIM-SM SPT-Switchover Example
  - Step 8: When the (Si, G) Joins reach the first-hop router directly connected to source “Si”, a complete branch of the SPT has been built (shown by the dashed arrows). This permits (Si, G) traffic to flow via the SPT to “rtr-b” and receiver “A”.
**PIM SM SPT-Switchover**

- **PIM-SM SPT-Switchover Example**
  - When the (Si, G)RP-bit Prune reaches “rtr-c”, the (171.68.37.121/32, 224.1.1.1) entry shown above is created as follows:
    - Because this (S, G) entry was created as a result of the receipt of an (S,G)RP-bit Prune, the “R” bit is set to denote that this forwarding state is applicable to traffic flowing down the Shared Tree and not the Source Tree.
    - Because the “R” bit is set, the RPF information is calculated in the direction of the RP instead of source “Si”. (Remember, this entry is applicable to (Si,G) traffic flowing down the Shared Tree and therefore the RPF information must point up the Shared Tree.) This results in an Incoming interface of **Serial0**, and an RPF neighbor address of "10.1.5.1".
    - The OIL for the (S, G) entry is constructed by copying the OIL from the (*) G entry minus the interface that the (Si, G)RP-bit Prune was received. Next, the IIF is removed from the OIL to prevent a possible route loop. These steps results in an (S, G) OIL containing only **Serial2**.
  - At this point, (Si, G) traffic flowing down the Shared Tree will be forwarded using the (Si, G) entry. (Si, G) traffic arriving at “rtr-a” will RPF correctly because the RPF information in the (Si, G) entry is pointing up the Shared Tree (as a result of the “R” bit) and will then be forwarded out all interfaces in the (Si, G) OIL. In this case, only **Serial2** remains in the (Si, G) OIL and therefore (Si, G) traffic will be sent to “rtr-d” but **not** “rtr-a”. This successfully prunes the redundant (Si, G) traffic from the branch of the Shared Tree between “rtr-c” and “rtr-a".
PIM SM SPT-Switchover

Unnecessary \((S_i, G)\) traffic is pruned from the Shared tree.

- **PIM-SM SPT-Switchover Example**
  - Step 9: At this point, the redundant \((S_i, G)\) traffic is pruned from the Shared Tree branch from “rtr-c” to “rtr-a”. \((S_i, G)\) traffic is reaching receiver “A” via the SPT through “rtr-a” and “rtr-b”.

[Diagram showing network topology and traffic flow]

Copyright © 1999, 2000, Cisco Systems, Inc.
Unnecessary (Si, G) traffic is pruned from the Shared tree.
(Si, G) traffic still flows via other branches of the Shared tree.

**PIM-SM SPT-Switchover Example**

- Step 10: (Si, G) traffic is still reaching receiver “B” via a branch of the Shared Tree through “rtr-c” and “rtr-d”. This is because the (Si, G) state in “rtr-c” still has Serial2 in its OIL.
PIM SM SPT-Switchover

Shared Tree Switchback Mechanism

- Once each minute
  - If “J” flag set in \((S_i, G)\) entry
    - Compute new \((S_i, G)\) traffic rate
    - If rate < SPT-threshold
      - Rejoin \((*, G)\) Tree for \((S_i, G)\) traffic
      - Send \((S_i, G)\) prune up SPT toward \(S_i\)
      - Delete \((S_i, G)\) entry

Shared Tree Switchback
- The Shared Tree Switchback (for lack of a better term) mechanism is used to switch sources back to the Shared Tree when their traffic rate falls below the SPT-Threshold.

Switchback Algorithm
- The Switchback mechanism runs once a minute. (This helps prevent Sources from cycling between Shared Tree and Shortest-Path Tree too rapidly.)
- For each \((S_i, G)\) entry in the Multicast Routing Table that has the “J” flag set, the mechanism computes the traffic rate for source \(S_i\).
- If the rate has fallen below the SPT-Threshold, a switchback to the Shared Tree is initiated by the last-hop router by:
  - Sending a Join/Prune message that contains a \((*, G)\) Join without a \((S_i, G)\)RP-bit Prune, up the Shared Tree (RPT). (This will cause the \((S_i, G)\) Prune state along the RPT to be deleted which will permit \((S_i, G)\) traffic to begin flowing down the RPT again.)
  - Deleting its \((S_i, G)\) entry in the Multicast Routing Table.
  - Send \((S_i, G)\) Prune up the Shortest-Path Tree (SPT) to stop traffic from flowing down the SPT.
- Note that this Switchback Algorithm is broken in older versions of IOS.
Agenda

- PIM Neighbor Discovery
- PIM SM State
- PIM SM Forwarding
- PIM SM Joining
- PIM SM Registering
- PIM SM SPT-Switchover
- PIM SM Pruning
- PIM SM State Maintenance
PIM SM Pruning

- **IGMP group times out / last host sends Leave**
  - Interface removed from all (*,G) & (S,G) entries
    - IF all interfaces in “oilist” for (*,G) are pruned;
      THEN send Prune up shared tree toward RP
    - Any (S, G) state allowed to time-out

- **Each router along path “prunes” interface**
  - IF all interfaces in “oilist” for (*,G) are pruned;
    THEN send Prune up shared tree toward RP
  - Any (S, G) state allowed to time-out

- **SM Pruning**
  - Locally connected host sends an IGMP Leave (or IGMP state times out in the router) for group “G”.
  - The interface is removed from the (*, G) and all (S, G) entries in the Multicast Routing Table.
    - If the (*, G) “Outgoing Interface list” is now Null, then send a (*, G) Prune up the Shared Tree (RPT) towards the RP.
    - Any remaining (S, G) entries are allowed to timeout and be deleted from the Multicast Routing Table.
  - When the routers up the Shared Tree receive the (*, G) Prune, they remove the interface on which the Prune was received from their (*, G) “Outgoing interface list”.
    - If as a result of removing the interface the (*, G) “Outgoing Interface list” becomes Null, then forward a (*, G) Prune up the Shared Tree (RPT) towards the RP.
    - Any remaining (S, G) entries are allowed to timeout and be deleted from the Multicast Routing Table.
PIM SM Pruning

- IGMP group times out / last host sends Leave
- Interface removed from all (*,G) & (S,G) entries
  - IF all interfaces in “oilist” for (*,G) are pruned; THEN send Prune up shared tree toward RP
  - Any (S, G) state allowed to time-out
- Each router along path “prunes” interface
  - IF all interfaces in “oilist” for (*,G) are pruned; THEN send Prune up shared tree toward RP
  - Any (S, G) state allowed to time-out

SM Pruning
- Locally connected host sends an IGMP Leave (or IGMP state times out in the router) for group “G”.
- The interface is removed from the (*, G) and all (S, G) entries in the Multicast Routing Table.
  - If the (*, G) “Outgoing Interface list” is now Null, then send a (*, G) Prune up the Shared Tree (RPT) towards the RP.
  - Any remaining (S, G) entries are allowed to timeout and be deleted from the Multicast Routing Table.
- When the routers up the Shared Tree receive the (*, G) Prune, they remove the interface on which the Prune was received from their (*, G) “Outgoing interface list”.
  - If as a result of removing the interface the (*, G) “Outgoing Interface list” becomes Null, then forward a (*, G) Prune up the Shared Tree (RPT) towards the RP.
  - Any remaining (S, G) entries are allowed to timeout and be deleted from the Multicast Routing Table.
• **State in “rtr-b” before Pruning**
  – Pay particular attention to the following:
    • Traffic is flowing down the Shared Tree. (Denoted by the existence of only the (*, G) entry.)
    • The “Incoming interface” is Ethernet0.
    • The “Outgoing interface list” contains Ethernet1.
    • The “C” flag is set in the (*, G) which denotes that there is a locally connected host for this group. (Rcvr A)
• **State in “rtr-a” before Pruning — RPT Case**
  
  – Pay particular attention to the following:
    
    • Traffic is flowing down the Shared Tree. (Denoted by the existance of only the (*, G) entry.)
    
    • The “Incoming interface” is Serial0.
    
    • The “Outgoing interface list” contains Ethernet0.
• PIM SM Pruning Example — RPT Case

1) The last-hop or Leaf router (rtr-b) receives an IGMP Group Leave message from “Rcvr A” for group “G”. After performing the normal IGMP Leave processing and finding that “Rcvr A” was the last host to leave, the IGMP state for group “G” on interface “E1” is deleted.

2) This causes interface “E1” to be removed from the “Outgoing interface list” of the (*, G) entry and any (Si, G) entries (in this case there are none) in the Multicast Routing Table. Because “E1” was the only interface in the (*, G) entry, its outgoing interface list becomes Null.

3) Because the (*, G) “Outgoing interface list” is now Null, a (*, G) Prune is sent up the Shared Tree (RPT) via “E0” toward the RP.
• PIM SM Pruning Example — RPT Case (cont.)

4) The (*) G Prune is received by "rtr-a" which causes interface “E0” to be removed from the “Outgoing interface list” of the (*, G) entry in the Multicast Routing Table.

(Note: “rtr-a” delayed Pruning E0 from the (*, G) entry for 3 seconds since this is a Multi-Access network and it needed to wait for a possible overriding Join from another PIM neighbor. Since none was received, the interface was pruned.)

5) Because the (*, G) “Outgoing interface list” is now Null, a (*, G) Prune is forwarded on up the Shared Tree (RPT) via “S0” toward the RP.

6) This pruning continues back toward the RP or until a router is reached whose (*, G) “Outgoing interface list” doesn’t go to Null as a result of the Prune.
• State in “rtr-b” before Pruning — SPT Case
  – Pay particular attention to the following:
    • Both a (*, G) and (S, G) entries exist.
    • The “J” flag is set in the (S, G) entry. This indicates that the (S, G) state was created as a result of the SPT-Threshold being exceeded.
    • The “T” flag is set in the (S, G) entry. This indicates that (S, G) traffic is being successfully received via the Shortest-Path Tree (SPT).
    • The “Incoming interface” is the same for the (*, G) and the (S, G) entry. This indicates that Shared Tree and the Shortest-Path tree are the same at this point.
• **State in “rtr-b” before Pruning — SPT Case**
  
  – Pay particular attention to the following:
    
    • Both a (*, G) and (S, G) entries exist.
    
    • The “T” flag is set in the (S, G) entry. This indicates that (S, G) traffic is being successfully received via the Shortest-Path Tree (SPT).
    
    • The “Incoming interface” is different for the (*, G) and the (S, G) entry. This indicates that Shared Tree and the Shortest-Path tree diverge at this point.
“rtr-b” is a Leaf router. Last host “Rcvr A”, leaves group G.

2. “rtr-b” removes E1 from (*,G) and any (Si,G) “oillists”.

3. “rtr-b” (*,G) “oillist” now empty; sends (*,G) Prune toward RP.

**PIM SM Pruning Example — SPT Case**

1) The last-hop or Leaf router (rtr-b) receives an IGMP Group Leave message from “Rcvr A” for group “G”. After performing the normal IGMP Leave processing and finding that “Rcvr A” was the last host to leave, the IGMP state for group “G” on interface “E1” is deleted.

2) This causes interface “E1” to be removed from the “Outgoing interface list” of the (*, G) entry and any (Si, G) entries in the Multicast Routing Table. Because “E1” was the only interface in the (*, G) and the (Si, G) entries, their outgoing interface lists become Null.

3) Because the (*, G) “Outgoing interface list” is now Null, a (*, G) Prune is sent up the Shared Tree (RPT) via “E0” toward the RP.
1. "rtr-b" is a Leaf router. Last host "Rcvr A", leaves group G.
2. "rtr-b" removes E1 from (*,G) and any (Si,G) "oilists".
3. "rtr-b" (*,G) "oilist" now empty; sends (*,G) Prune toward RP.
4. "rtr-b" stops sending periodic (S, G) joins.

**PIM SM Pruning Example — SPT Case (cont.)**

4) Because the (Si, G) “Outgoing interface list” is now Null, “rtr-b” stops sending Periodic (Si, G) Join messages up the Shortest-Path Tree (SPT).
**PIM SM Pruning Example — SPT Case (cont.)**

5) The (*) Prune is received by “rtr-a” which causes interface “E0” to be removed from the “Outgoing interface list” of the (*) entry in the Multicast Routing Table.

(Note: “rtr-a” delayed Pruning E0 from the (*) entry for 3 seconds since this is a Multi-Access network and it needed to wait for a possible overriding Join from another PIM neighbor. Since none was received, the interface was pruned.)

6) Because the (*) “Outgoing interface list” is now Null, a (*) Prune is forwarded on up the Shared Tree (RPT) via “S0” toward the RP.

7) Because “rtr-a” is no longer receiving (Si, G) Join messages from “rtr-b”, the (Si, G) state eventually times out. This causes a (Si, G) Prune to be sent up the Shortest-Path Tree (SPT) towards the source “Si”.

8) Traffic stops flowing down the Shortest-Path Tree (SPT).
5) The (*, G) Prune is received by “rtr-a” which causes interface “E0” to be removed from the “Outgoing interface list” of the (*, G) entry in the Multicast Routing Table.

(Note: “rtr-a” delayed Pruning E0 from the (*, G) entry for 3 seconds since this is a Multi-Access network and it needed to wait for a possible overriding Join from another PIM neighbor. Since none was received, the interface was pruned.)

6) Because the (*, G) “Outgoing interface list” is now Null, a (*, G) Prune is forwarded on up the Shared Tree (RPT) via “S0” toward the RP.

7) Because “rtr-a” is no longer receiving (Si, G) Join messages from “rtr-b”, the (Si, G) state eventually times out. This causes a (Si, G) Prune to be sent up the Shortest-Path Tree (SPT) towards the source “Si”.

8) Traffic stops flowing down the Shortest-Path Tree (SPT).
**PIM SM Pruning Example — SPT Case (cont.)**

5) The (*, G) Prune is received by “rtr-a” which causes interface “E0” to be removed from the “Outgoing interface list” of the (*, G) entry in the Multicast Routing Table.

(Note: “rtr-a” delayed Pruning E0 from the (*, G) entry for 3 seconds since this is a Multi-Access network and it needed to wait for a possible overriding Join from another PIM neighbor. Since none was received, the interface was pruned.)

6) Because the (*, G) “Outgoing interface list” is now Null, a (*, G) Prune is forwarded on up the Shared Tree (RPT) via “S0” toward the RP.

7) Because “rtr-a” is no longer receiving (Si, G) Join messages from “rtr-b”, the (Si, G) state eventually times out. This causes a (Si, G) Prune to be sent up the Shortest-Path Tree (SPT) towards the source “Si”.

8) Traffic stops flowing down the Shortest-Path Tree (SPT).
Another (S_i,G) data packet arrives via Serial1.

rtr-a responds by sending an (S_i,G) Prune toward source.

PIM SM Pruning Example — SPT Case (cont.)

5) The (*, G) Prune is received by "rtr-a" which causes interface "E0" to be removed from the "Outgoing interface list" of the (*, G) entry in the Multicast Routing Table.

(Note: "rtr-a" delayed Pruning E0 from the (*, G) entry for 3 seconds since this is a Multi-Access network and it needed to wait for a possible overriding Join from another PIM neighbor. Since none was received, the interface was pruned.)

6) Because the (*, G) "Outgoing interface list" is now Null, a (*, G) Prune is forwarded on up the Shared Tree (RPT) via "S0" toward the RP.

7) Because "rtr-a" is no longer receiving (Si, G) Join messages from "rtr-b", the (Si, G) state eventually times out. This causes a (Si, G) Prune to be sent up the Shortest-Path Tree (SPT) towards the source "Si".

8) Traffic stops flowing down the Shortest-Path Tree (SPT).
7) Another (S_i,G) data packet arrives via Serial1.
8) ‘rtr-a’ responds by sending an (S_i,G) Prune toward source.
9) (S_p,G) traffic ceases flowing down SPT.

**PIM SM Pruning Example — SPT Case (cont.)**

5) The (*, G) Prune is received by “rtr-a” which causes interface “E0” to be removed from the “Outgoing interface list” of the (*, G) entry in the Multicast Routing Table.

(Note: “rtr-a” delayed Pruning E0 from the (*, G) entry for 3 seconds since this is a Multi-Access network and it needed to wait for a possible overriding Join from another PIM neighbor. Since none was received, the interface was pruned.)

6) Because the (*, G) “Outgoing interface list” is now Null, a (*, G) Prune is forwarded on up the Shared Tree (RPT) via “S0” toward the RP.

7) Because “rtr-a” is no longer receiving (Si, G) Join messages from “rtr-b”, the (Si, G) state eventually times out. This causes a (Si, G) Prune to be sent up the Shortest-Path Tree (SPT) towards the source “Si”.

8) Traffic stops flowing down the Shortest-Path Tree (SPT).
Agenda

- PIM Neighbor Discovery
- PIM SM State
- PIM SM Forwarding
- PIM SM Joining
- PIM SM Registering
- PIM SM SPT-Switchover
- PIM SM Pruning
- PIM SM Special Cases
• **RP on a Stick**

  This is a special situation that occurs under the following conditions:
  
  • All branches of the Shared Tree are out a single interface on the RP (i.e. there is only a single interface in the \((*, G)\) OIL at the RP.)
  
  • All sources for the group are out the same interface. (This would result in \((S, G)\) entries with Null OIL’s since the incoming interface can never appear in the OIL of an entry.)

• **Unusually State results in this condition**

  – Special PIM rules had to be created that were not in the original PIMv2 specification in order to avoid situations where:
    
    • \((S, G)\) traffic flows were incorrectly pruned.
    
    • \((S, G)\) traffic continued to flow to the RP only to be dropped.
    
    • \((S, G)\) state would get stuck in the RP and the First-Hop router even when the source has long since stopped sending.

  – Problem was solved in IOS 12.0 by:
    
    • Special “Proxy Join” Timer and
    
    • Introduction of “Atomic” and “Non-Atomic” \((*, G)\) Joins
• **RP-on-a-Stick Example**
  – Consider that above network topology where both “rtr-b” and “rtr-c” share a common Ethernet segment with the RP.
RP-on-a-Stick Example

- When a host behind “rtr-c” joins group 224.1.1.1, a branch of the Shared Tree is created (shown by the solid arrow) which results in the following state on the RP:

  - (*, 224.1.1.1), 00:01:21/00:02:59, RP 10.1.4.1, flags: S
  - Incoming interface: Null, RPF nbr 0.0.0.0,
  - Outgoing interface list:
    - Ethernet0, Forward/Sparse, 00:01:21/00:02:39
**RP on a Stick**

- **RP-on-a-Stick Example**
  - This also results in the following state on “rtr-c”:

  - 
    - 
      - (*, 224.1.1.1), 00:01:21/00:02:59, RP 10.1.4.1, flags: SC
      - Incoming interface: Ethernet1, RPF nbr 10.1.4.1,
      - Outgoing interface list:
        - Ethernet0, Forward/Sparse, 00:01:21/00:02:39

• **RP-on-a-Stick Example**
  
  Now assume that source 192.1.1.1 behind “rtr-b” begins sending to group 224.1.1.1. After the normal Register process has completed, a branch of the SPT (shown by the heavy dashed arrow) is built from “rtr-b” to the RP. This allows traffic to flow to the members as shown by the thin dashed arrows.
• **RP-on-a-Stick Example**
  
  The creation of the SPT results in the following state on “rtr-b”:

  - (*, 224.1.1.1), 00:01:21/00:02:59, RP 10.1.4.1, flags: SP
  - Incoming interface: Ethernet1, RPF nbr 10.1.4.1,
  - Outgoing interface list:
    - (192.1.1.1/32, 224.1.1.1), 00:00:49/00:02:46, flags: T
    - Incoming interface: Ethernet0, RPF nbr 0.0.0.0,
    - Outgoing interface list:
      - Ethernet0, Forward/Sparse, 00:00:49/00:02:11
**RP-on-a-Stick Example**

- The creation of the SPT also results in the following state on the RP:

  
  
  (\(*, \ 224.1.1.1\), 00:02:43/00:02:59, RP 10.1.4.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list:
    Ethernet0, Forward/Sparse, 00:02:43/00:02:17
  
  (192.1.1.1/32, 224.1.1.1), 00:00:49/00:02:46, flags: PT
  Incoming interface: Ethernet0, RPF nbr 10.1.4.2,
  Outgoing interface list:

- Notice that the OIL of the (S, G) entry is Null which, in turn, results in the “P” flag being set. Normally, this would cause the RP to send an (S, G) Prune toward the source to shut off the flow of (S, G) traffic. However in this case, that would starve the host behind “rtr-c” of the desired group traffic. Obviously, something else must be done to prevent this.
RP on a Stick

- Solution requires three new concepts
  - Atomic & Non-Atomic Joins
  - Proxy Join Timer/Flag
  - Header-only Registers

RP-on-a-Stick Solution
- In order to deal of this special situation, several new concepts were added to the 12.0 implementation of PIM. These are:
  - Atomic vs. Non-Atomic (*, G) Joins
  - The Proxy Join Timer (and its flag) on (S, G) entries
  - Header-only Registers (aka Data-less Registers)
- Each of the above are discussed in the following pages
RP on a Stick

• Non-Atomic Joins
  – Contains (*, G) Join for group “G” only
    • This is the type of (*, G) Join we are familiar with

• Atomic Joins
  – Contains (*, G) Join for group “G” followed by
    – (S, G)RP-bit Prunes for all sources in group “G”
      • Used to prune unnecessary (S, G) traffic from the Shared Tree after switchover to the SPT.
  – All in the same PIM Join/Prune message

• Non-Atomic Joins
  – This is a PIM Join/Prune message that contains only a (*, G) Join for group “G” in the Join list without any associated (S, G)RP-bit Prunes for group “G” in the Prune list.
    • This is the typical (*, G) Join that has been described in most of the examples in Module 5, “PIM-SM”.

• Atomic Joins
  – This is a PIM Join/Prune message that contains a (*, G) Join for group “G” in the Join list AND a complete list of all (S, G)RP-bit Prunes for group “G” in the Prune list.
    • Remember, these (S, G)RP-bit Prunes are used to Prune specific (S, G) traffic off of the Shared Tree after a router has joined the SPT directly toward the source.
**Example: Atomic (**, G) Join**

- In this example, a (**, G) entry for group 224.1.1.1 in the Join list of the PIM Join/Prune message. (The WC (wildcard) and RP (RP-Tree) bits tell us that this entry is a (**, G) Join to RP 10.1.4.1.)

- In addition, there is an (S, G) entry for group 224.1.1.1 (192.1.1.1, 224.1.1.1) with the RP-bit set in the Prune list. (I.e. an (S, G)RP-bit Prune exists for group 224.1.1.1.)

- Because both a (**, G) Join along with one or more (S, G)RP-bit Prunes exist in this Join/Prune message for group 224.1.1.1, it is said to contain an Atomic (**, G) Join for group 224.1.1.1.
### Example: Non-Atomic (*, G) Join

- Also in this example, is a (*, G) entry for group 224.1.0.5 in the Join list of the PIM Join/Prune message. (The WC (wildcard) and RP (RP-Tree) bits tell us that this entry is a (*, G) Join to RP 10.1.4.1).
- In addition, there are no (S, G) entries for group 224.1.0.5 with the RP-bit set in the Prune list. (i.e. there are no (S, G)RP-bit Prunes for group 224.1.0.5).
- Because only a (*, G) Join exists in this Join/Prune message for group 224.0.1.5 without any corresponding (S, G)RP-bit Prunes, it is said to contain an Non-Atomic (*, G) Join for group 224.0.1.5.

---

**PIM Join/Prune Message**

<table>
<thead>
<tr>
<th>Join List</th>
<th>Prune List</th>
</tr>
</thead>
<tbody>
<tr>
<td>(10.1.4.1, 224.1.0.5)</td>
<td>WC, RP</td>
</tr>
<tr>
<td>(10.1.4.1, 224.1.1.1)</td>
<td>WC, RP</td>
</tr>
<tr>
<td>(10.1.4.1, 224.2.2.2)</td>
<td>WC, RP</td>
</tr>
<tr>
<td>(10.1.19.21, 224.2.2.2)</td>
<td>RP</td>
</tr>
<tr>
<td>(10.1.19.21, 224.1.0.5)</td>
<td>WC, RP</td>
</tr>
<tr>
<td>(10.1.4.1, 224.1.0.5)</td>
<td>WC, RP</td>
</tr>
<tr>
<td>(10.1.4.1, 224.3.3.3)</td>
<td>WC, RP</td>
</tr>
<tr>
<td>(192.1.1.1, 224.1.1.1)</td>
<td>RP</td>
</tr>
<tr>
<td>(192.1.4.2, 224.2.2.2)</td>
<td>RP</td>
</tr>
</tbody>
</table>

(*) Join

+ NO (S, G)RP-bit Prunes

= Non-Atomic Join
**Proxy Join Timer**

The Proxy Join Timer only exists on (S, G) entries in the Mroute table. Its purpose is to attract (S, G) traffic to the router even when the OIL of the (S, G) entry is Null. This maintains the flow of (S, G) traffic in cases such as the RP-on-a-Stick.

**Proxy Join Timer Rules**

- The Proxy Join Timer is started when the RP receives the first (S, G) Register message when a source goes active if:
  - The OIL is null in resulting (S, G) entry AND the OIL is non-Null in the (*, G) entry. (This is the RP-on-a-Stick condition.)
- The Proxy Join Timer is started whenever the router receives a Non-Atomic (*,G) Join and an (S, G) entry already exists.
  - This timer runs for 2 minutes unless restarted by the receipt of another Non-Atomic (*, G) Join.
  - When this timer is running on an (S, G) entry, the "X" flag will be displayed in the flags field of the entry.
- When the Proxy Join Timer is running, the router will:
  - Send periodic (S, G) Joins toward the source even though the OIL is Null.
  - Suppress the sending of (S, G) Prunes toward the source even though the OIL is Null.
RP on a Stick

• **Header-only Registers**
  - Used to keep (S, G) state alive in the RP
  - Sent every 2 minutes by First-hop router
    - As long as source is still active
    - Continues sending until a Register-Stop is received
  - Register Messages contains null (S,G) data packet
    - Processed by the RP
    - Resets (S, G) entry timer at the RP
    - RP doesn’t send Null packet down Shared Tree

• **Header-only (Data-less) Registers**
  Normally, the Expire timer of an (S, G) entry is reset to 3 minutes every time the router forwards a packet associated with that entry. However, in the RP-on-a-Stick case, the (S, G) entry has a Null OIL and is therefore not forwarding any packets. This would normally result in the (S, G) entry timing out at the RP. This can not be allowed to happen as it is possible that another member somewhere in the network could join the Shared Tree via another interface. If the (S, G) entry was allowed to timeout, the RP would not be able to trigger the “Batch-Join” to rejoin the SPT when this new member joined. (Because there wouldn’t be any (S, G) entry to tell the RP of the active source.)

To prevent this from happening, the behavior of the First-Hop DR was changed in IOS 12.0 so that (S, G) Header-only (aka Data-less) Registers would be sent periodically (every 2 minutes) to the RP. These Header-only Registers cause the RP to reset the Expire timer in the (S, G) entry thereby preventing it from timing out.

• **Contents of Header-only Registers**
  - Header-only Registers contain a specially formatted null or “data-less” (S, G) packet.
  - These “null” (S, G) packets are not forwarded down the Shared Tree by the RP.
• **RP-on-a-Stick Example**

  - In this example, “rtr-c” is sending Non-Atomic (*, G) Joins to the RP to keep on the Shared Tree. (Note that “rtr-c” has not joined the SPT at this point. This could be due to the SPT-Threshold being set to *Infinity*.)

  - The RP is now running version 12.0 or later of IOS. Therefore, when the Non-Atomic (*, G) Join for group 224.1.1.1 is received, the RP starts the Proxy Join Timer in all (S, G) entries for group 224.1.1.1. This results in the following state in the RP:

    ```plaintext
    (*, 224.1.1.1), 00:02:43/00:02:59, RP 10.1.4.1, flags: S
    Incoming interface: Null, RPF nbr 0.0.0.0,
    Outgoing interface list:
      Ethernet0, Forward/Sparse, 00:02:43/00:02:17
    ```
    
    ```plaintext
    (192.1.1.1/32, 224.1.1.1), 00:00:49/00:02:46, flags: FXT
    Incoming interface: Ethernet0, RPF nbr 10.1.4.2,
    Outgoing interface list:
    ```

  - Notice the “X” flag is set in the above example. This causes the RP to continue sending (S, G) Joins toward the source (even though the OIL is Null) which, in turn, will keep the traffic flowing to the member across the common Ethernet segment.

  ```python
  Proxy Join Timer started/restarted by Non-Atomic Joins
  ```
• RP-on-a-Stick Example
  – The First-hop router (rtr-b) is also running version 12.0 or later of IOS and it will therefore send periodic Header-only (S, G) Register messages to the RP.
  – When RP receives these Header-only (S, G) Registers, (roughly every 2 minutes), it resets the Expire timer in the corresponding (S, G) entry in the Mroute table. This results in the following state in the RP:

```
(*, 224.1.1.1), 00:02:43/00:02:59, RP 10.1.4.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list:
    Ethernet0, Forward/Sparse, 00:02:43/00:02:17
(192.1.1.1/32, 224.1.1.1), 00:00:49/00:02:59, flags: PXT
  Incoming interface: Ethernet0, RPF nbr 10.1.4.2,
  Outgoing interface list:

(Notice the Expire timer in the (S, G) entry has been reset.)
```
• Turnaround Router
  – As it turns out, the RP-on-a-Stick problem is actually a special case of another problem referred to the Turnaround Router problem. This situation occurs whenever:
    • There is only a single branch of the Shared Tree and
    • the Shared Tree and a SPT share a common path to the RP.
  – We want to have the (S, G) traffic flow along the SPT toward the RP and “turnaround” at the appropriate router in the network and flow back down the Shared Tree without sending unnecessary traffic to the RP.

• Turnaround Router Solution
  – Once again, the new concepts of
    • Proxy Join Timer
    • Atomic and Non-Atomic Joins
    • Header-only Registers
  permit the routers to solve this problem.
• Turnaround Router Example
  – In this example, we once again have a single branch of the 224.1.1.1 Shared Tree at the RP.
  – The SPT for source (192.1.1.1, 224.1.1.1) merges with the single branch of the 224.1.1.1 Shared Tree at "rtr-x". This router is referred to as the Turnaround Router because it is here that we want the (S, G) traffic to turnaround and flow back down the Shared Tree to the members of group 224.1.1.1.
  – Additionally, we do not want the (S, G) traffic flow to all the way to the RP as it is unnecessary traffic because there is only the single branch of the Shared Tree. In cases where the number of hops between the Turnaround Router and the RP is large or where the links along this path are congested, the flow of traffic to the RP would simply waste precious network resources.
  – Instead, we want the traffic to only flow as shown by the thin dashed arrows in the drawing above.
• Turnaround Router — Step-by-Step
  – Step 1
    • The host connected to “rtr-c” joins group 224.1.1.1. This causes “rtr-c” to create (*, G) state and sends a Non-Atomic (*, G) Join toward the RP.
  – Step 2
    • The Turnaround Router (rtr-x) receives this Non-Atomic (*, G) Join and it too creates (*, G) state and sends a Non-Atomic (*, G) Join to the RP.
  – Step 3
    • The RP receives the (*, G) Join and creates (*, G) state with only Serial0 in the OIL.
• Turnaround Router — Step-by-Step

  – Step 4
  
  • The source, 192.1.1.1, begins sending to group 224.1.1.1. This causes the first-hop router (rtr-b) to send an (S, G) Register to the RP.

  – Step 5
  
  • The RP processes the Register message and creates an (S, G) entry. Because the OIL of the newly created (S, G) entry is Null and the OIL of the (*, G) entry is non-Null, the RP starts the Proxy Timer in the (S, G) entry and sends an (S, G) Join toward the source.

  At this point, the state in the RP is as follows:

  (*, 224.1.1.1), 00:02:43/00:02:59, RP 10.1.3.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list:
  Serial0, Forward/Sparse, 00:02:43/00:02:17

  (192.1.1.1/32, 224.1.1.1), 00:00:49/00:02:59, flags: FXT
  Incoming interface: Serial0, RPF nbr 10.1.3.2,
  Outgoing interface list:

  Notice that the Proxy Join Timer is running (note the “X” flag in the (S,G) entry.)
  • While the Proxy Join Timer is running, the RP will continue to send periodic (S, G) Joins toward the source.
  • The Proxy Join Timer will be restarted each time the RP receives another Non-Atomic Join from “rtr-x”.

Copyright © 1999, 2000, Cisco Systems, Inc.
• Turnaround Router — Step-by-Step
  – Step 6
    • The (S, G) Join travels hop-by-hop building the SPT from the source to the RP. At this point, the state in the Turnaround Router (rtr-x) is as follows:

\[(*, 224.1.1.1), 00:02:43/00:02:59, RP 10.1.3.1, \text{flags: } S\]
Incoming interface: Serial0, RPF nbr 10.1.3.1,
Outgoing interface list:
  Ethernet0, Forward/Sparse, 00:02:43/00:02:17

\[(192.1.1.1/32, 224.1.1.1), 00:00:49/00:02:59, \text{flags: } T\]
Incoming interface: Ethernet0, RPF nbr 10.1.4.2,
Outgoing interface list:
  Serial0, Forward/Sparse, 00:00:48/00:02:12
• **Turnaround Router — Step-by-Step**
  – Once “rtr-b” receives the (S, G) Join, traffic begins to flow as shown above.
• Turnaround Router — Step-by-Step
  – Step 7
  • Router “rtr-x” detects that the paths of the SPT and the Shared Tree diverge at this point. As a result, “rtr-x” begins sending periodic (S,G)RP-bit Prunes up the Shared Tree in the same Join/Prune message with the periodic (*, G) Joins. In other words, it begins sending Atomic Joins to the RP instead of Non-Atomic Joins!
  (Note: Router “rtr-x” knows that the SPT and Shared Tree paths have diverged at this point because the RPF information (Incoming Interface and/or RPF neighbor) of the (S, G) entry is different than the (*,G) entry.)
• Turnaround Router — Step-by-Step

Because the RP is no longer receiving Non-Atomic Joins, the Proxy Join Timer for the (S, G) entry is no longer being restarted and it eventually times out. This is indicated by the “X” flag being clear in the (S, G) entry shown below:

(*, 224.1.1.1), 00:02:43/00:02:59, RP 10.1.4.1, flags: S
Incoming interface: Null, RPF nbr 0.0.0.0,
Outgoing interface list:
Serial0, Forward/Sparse, 00:02:43/00:02:17

(192.1.1.1/32, 224.1.1.1), 00:00:49/00:02:59, flags: PT
Incoming interface: Serial0, RPF nbr 10.1.3.2,
Outgoing interface list:
• Turnaround Router — Step-by-Step
  
  – Step 8
    • Now that the Proxy Join Timer is no longer running, the RP resumes its normal behavior and sends an (S, G) Prunes toward the source in response to the arrival of (S, G) packets.

  – Step 9
    • When “rtr-x” receives the (S, G) Prune, it removes Serial0 from its outgoing interface list.
• **Turnaround Router — Step-by-Step**
  
  – As a result of Serial0 being removed from the (S, G) OIL, the flow of traffic to the RP is shutoff.
  – Step 10
    
    • Non-Atomic Joins arriving at "rtr-x" now start the Proxy Join Timer. (Note the “X” flag in the (S, G) entry.) This causes the Turnaround Router (rtr-x) to suppress sending (S, G) Prunes and instead, send (S, G) Joins toward the source. This keeps the traffic flowing as shown.
• Turnaround Router
  – Step 11
    • Finally, Header-only Registers sent by the First-hop router (rtr-b) continue to reset the Expire timer in the (S, G) entry at the RP. This prevents the (S, G) entry from timing out and being deleted at the RP.
• **Turnaround Router**
  – As a result of the Header-only Registers, the state in the RP will be as follows as long as the source and member remain active:

    `(*, 224.1.1.1), 00:02:43/00:02:59, RP 10.1.3.1, flags: S
    Incoming interface: Null, RPF nbr 0.0.0.0,
    Outgoing interface list:
        Serial0, Forward/Sparse, 00:02:43/00:02:17`

    `(192.1.1.1/32, 224.1.1.1), 00:00:49/00:02:59, flags: PT
    Incoming interface: Serial0, RPF nbr 10.1.3.2,
    Outgoing interface list:`

    `(192.1.1.1/32, 224.1.1.1), 00:00:49/00:02:59, flags: PT
    Incoming interface: Serial0, RPF nbr 10.1.3.2,
    Outgoing interface list:`

    `(*, 224.1.1.1), 00:02:43/00:02:59, RP 10.1.3.1, flags: S
    Incoming interface: Null, RPF nbr 0.0.0.0,
    Outgoing interface list:
        Serial0, Forward/Sparse, 00:02:43/00:02:17`
• Turnaround Router
  – As a result of the Non-Atomic Joins, the state in the Turnaround router will be as follows as long as the source and member remain active:

    (*, 224.1.1.1), 00:02:43/00:02:59, RP 10.1.3.1, flags: S
    Incoming interface: Serial0, RPF nbr 10.1.3.1,
    Outgoing interface list: Ethernet0, Forward/Sparse, 00:02:43/00:02:17

    (192.1.1.1/32, 224.1.1.1), 00:00:49/00:02:59, flags: PXT
    Incoming interface: Ethernet0, RPF nbr 10.1.4.2,
    Outgoing interface list: