APRICOT 10: Understanding and Deploying IP Multicast Networks

Srini Irigi, SPG TME,

Cisco Systems, CCIE 6147

Stefan Olofsson, SPG TME

Cisco Systems

welcome to
the human network.

Session Goal

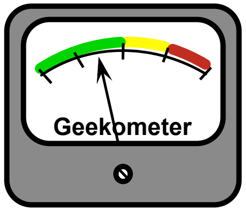
- To provide you with an understanding of the fundamentals of IP Multicast to help maintain your exiting deployment or plan a new multicast deployment.

Today's Agenda

- Multicast Fundamentals
- Multicast Service Models, Distribution Trees, Forwarding
- Multicast Protocol Basics
- Layer2 Multicast
- PIM Mechanics
- SSM
- BiDir
- RPs
- Scoping

Fundamentals of IP Multicast

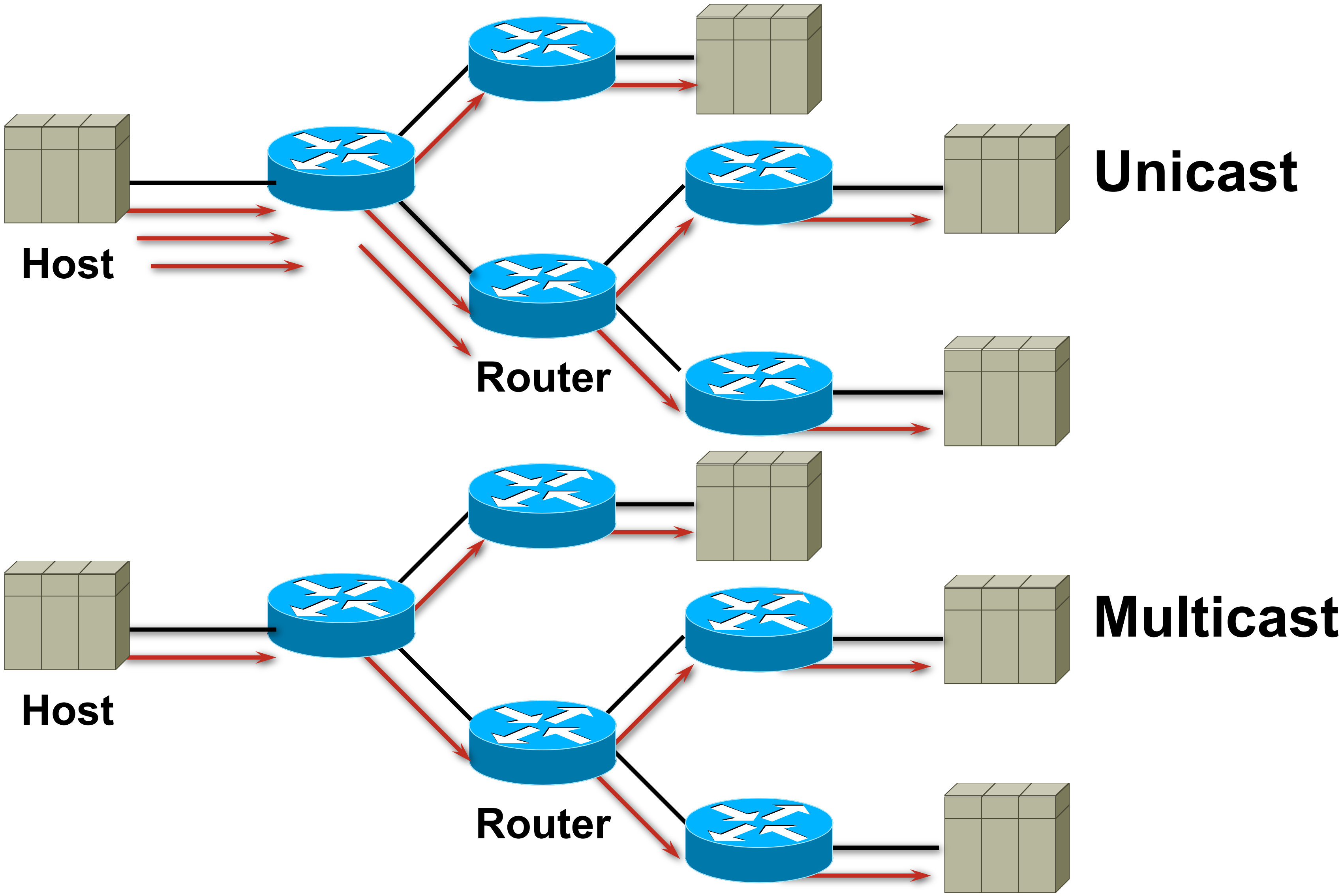
Agenda



- Why Multicast
- Multicast Applications
- Multicast Service Model
- Multicast Distribution Trees
- Multicast Forwarding
- Multicast Protocol Basics

Why Multicast?

Multicast Advantages



Multicast Disadvantages

Multicast Is UDP Based!!!

- Best-effort delivery: Drops are to be expected. Multicast applications should not expect reliable delivery of data and should be designed accordingly. Reliable Multicast is still an area for much research. Expect to see more developments in this area.
- No congestion avoidance: Lack of TCP windowing and “slow-start” mechanisms can result in network congestion. If possible, Multicast applications should attempt to detect and avoid congestion conditions.
- Duplicates: Some multicast protocol mechanisms (e.g. Asserts, Registers and Shortest-Path Tree Transitions) result in the occasional generation of duplicate packets. Multicast applications should be designed to expect occasional duplicate packets.
- Out-of-sequence packets: Various network events can result in packets arriving out of sequence. Multicast applications should be designed to handle packets that arrive in some other sequence than they were sent by the source.

Multicast Applications

Enterprise Primary Multicast Applications

- IP/TV
- Hoot-n-Holler
- VoIP Music-on-Hold
- TIBCO Data Distribution
- Internet Multicast Access

IP/TV

- One-to-many video multicast

  Live or rebroadcast content

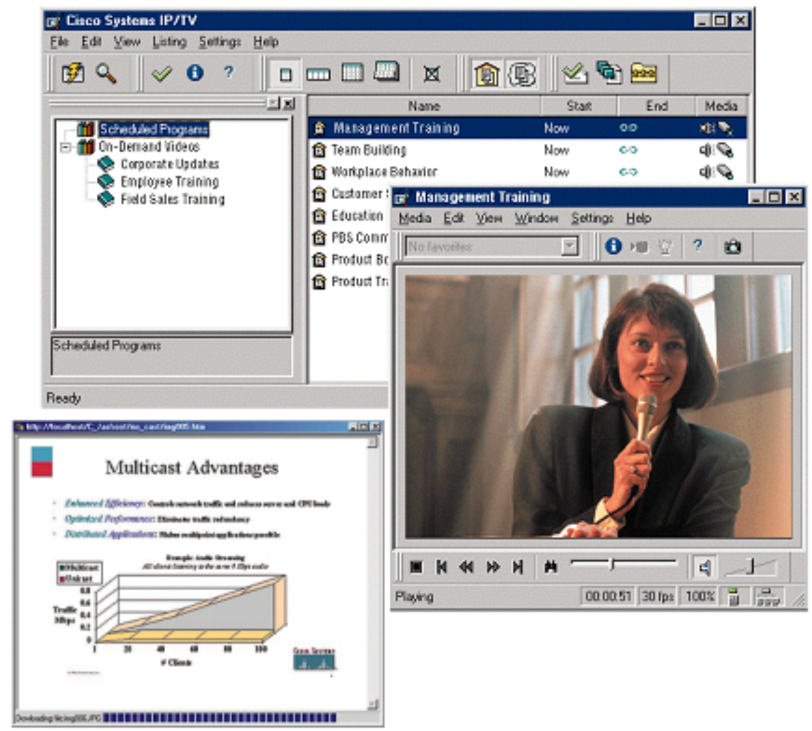
  Synchronized presentations

  Integrated “Question Manager”

  Supports “Source Specific Multicast” (SSM)

  Video-on-Demand (VoD)

    (Unicast only)



Corporate Broadcasts (IP/TV)

- Multicast Protocol: SSM
- IP/TV assigned to an SSM group range
  - **No** RPs, minimal configuration
  - Avoids “Capt. Midnight” problem
- Additional options:
  - Bandwidth-based group scoping

Hoot-n-Holler

Hoot-n-Holler
Turret

Hoot-n-Holler

- Broadcast audio network
- Typically point to multipoint
- Uses specialized analog 4-wire phones (hoot phones) and digital turrets
- Brokerages, utilities, media companies, mass transit, publishing, etc.

TIBCO Data Distribution

- Popular with financial institutions
  - Used to send stock market data to traders
- Uses subscribe/publish model
- Clients multicast subscriptions messages
  - Specifying data flow(s) they wish to receive
- Servers receive subscriptions
  - Build list of all requested data flows
  - Primary server multicast requested flows
  - Backup server takes over if primary fails

Example Multicast Applications

- Old Mbone Multicast Applications

  sdr—session directory

    Lists advertised sessions

    Launches multicast application(s)

  vat—audio conferencing

    PCM, DVI, GSM, and LPC4 compression

  vic—video conferencing

    H.261 video compression

    Motion JPEG

  wb—white board

    Shared drawing tool

    Can import PostScript images

    Uses reliable Multicast

Multicast Service Model

IP Multicast Service Model

- RFC 1112 (Host Ext. for Multicast Support)
- Each multicast group identified by a class-D IP address
- Members of the group could be present anywhere in the Internet
- Members join and leave the group and indicate this to the routers
- Senders and receivers are distinct:

  i.e., a sender need not be a member
- Routers listen to all multicast addresses and use multicast routing protocols to manage groups

IP Multicast Packet

- Source address
  - Unique unicast IP address of the packet source
- Destination address
  - ClassD address range
  - Does NOT represent a unique unicast destination address
  - Used to represent a unique group of receivers

IP Multicast Addressing

- Multicast Group Addresses (224.0.0.0/4)

  Range: 224.0.0.0–239.255.255.255

  Old Class D address range.

  High-order 4 bits are 1110

Multicast Address Ranges

- Link-Local Address Range

  224.0.0.0–224.0.0.255

- Global Address Range

  224.0.1.0–238.255.255.255

- Administratively Scoped Address Range

  239.0.0.0–239.255.255.25

- Scope Relative Address Range

  Top 256 addresses of a Scoped Address Range

Link-Local Address Range

- Assigned by IANA

  224.0.0.0–224.0.0.255

  Local wire multicast

    TTL = 1

  Examples:

    224.0.0.5 = OSPF_DR's

    224.0.0.10 = EIGRP Hello's

    224.0.0.13 = All_PIM_Routers

    224.0.0.22 = All_IGMPv3_Routers

Global Address Range

- Assigned by IANA
  - Address Range: 224.0.1.0–238.255.255.255
  - Generally intended for "global" Internet scope multicast
  - Sometimes assigned to specific protocols
    - Example: Auto-RP (224.0.1.39 and 224.0.1.40)
  - Problem:
    - IANA is coming under increasing pressure from companies to assign them blocks of addresses for their applications or content services
    - **This was never the intent of this block!**
      - GLOP Addressing or SSM should be used instead!

Global Multicast Address Assignment

- Dynamic Group Address Assignment
  - Historically accomplished using SDR application
    - Sessions announced over well-known group(s)
      - Address collisions detected and resolved at session creation time
      - Has problems scaling
  - Other techniques considered
    - Multicast Address Set-Claim (MASC)
      - Hierarchical, dynamic address allocation scheme
      - Unlikely to be deployed
  - No really good dynamic assignment method available for Global multicast
  - But is dynamic assignment really necessary with GLOP and SSM available?

Global Multicast Address Assignment

- Static Group Address Assignment
  - RFC 3180—GLOP Addressing in 233/8
    - Group range: 233.0.0.0–233.255.255.255
      - Your AS number is inserted in middle two octets
      - Remaining low-order octet used for group assignment
    - EGLOP Addresses
      - Make use of private AS numbers
      - Assigned by a Registration Authority

Global Multicast Address Assignment

- Static Group Address Assignment
  - Source Specific Multicast
    - Address range: 232.0.0.0/8
    - Flows based on both Group **and** Source address
      - Two different content flows can share the same Group address without interfering with each other
    - **Provides virtually unlimited address space!**
      - Preferred method for global one-to-many multicast

Private Multicast Address Assignment

- Assigned from the private 239.0.0.0/8 range

  May be subdivided into geographic scopes ranges

  Administration responsibility can be by scope range

- Question:

  “What technology is most often used to manage private multicast assignment?”

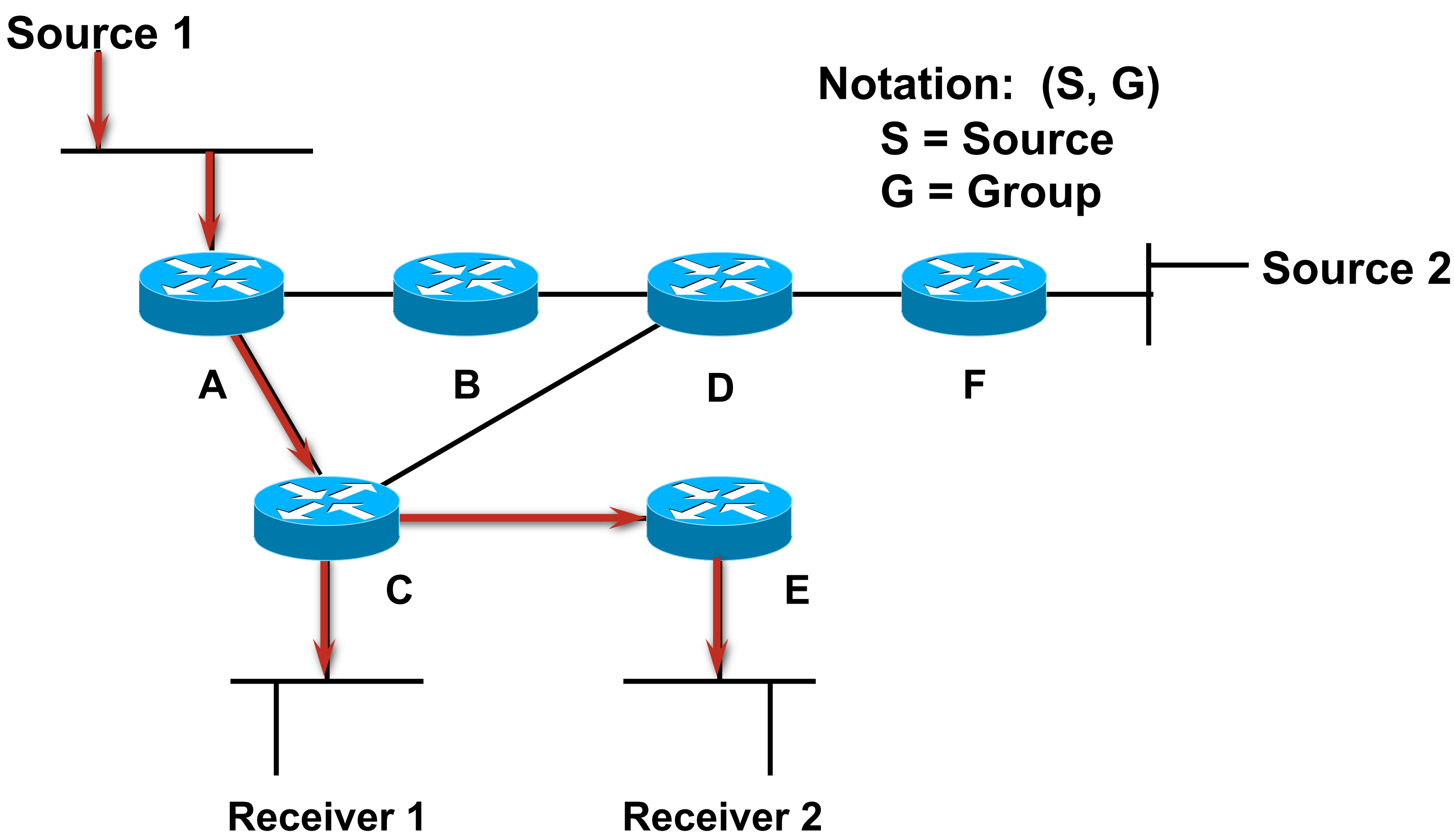
- Answer:

  A spreadsheet

Multicast Distribution Trees

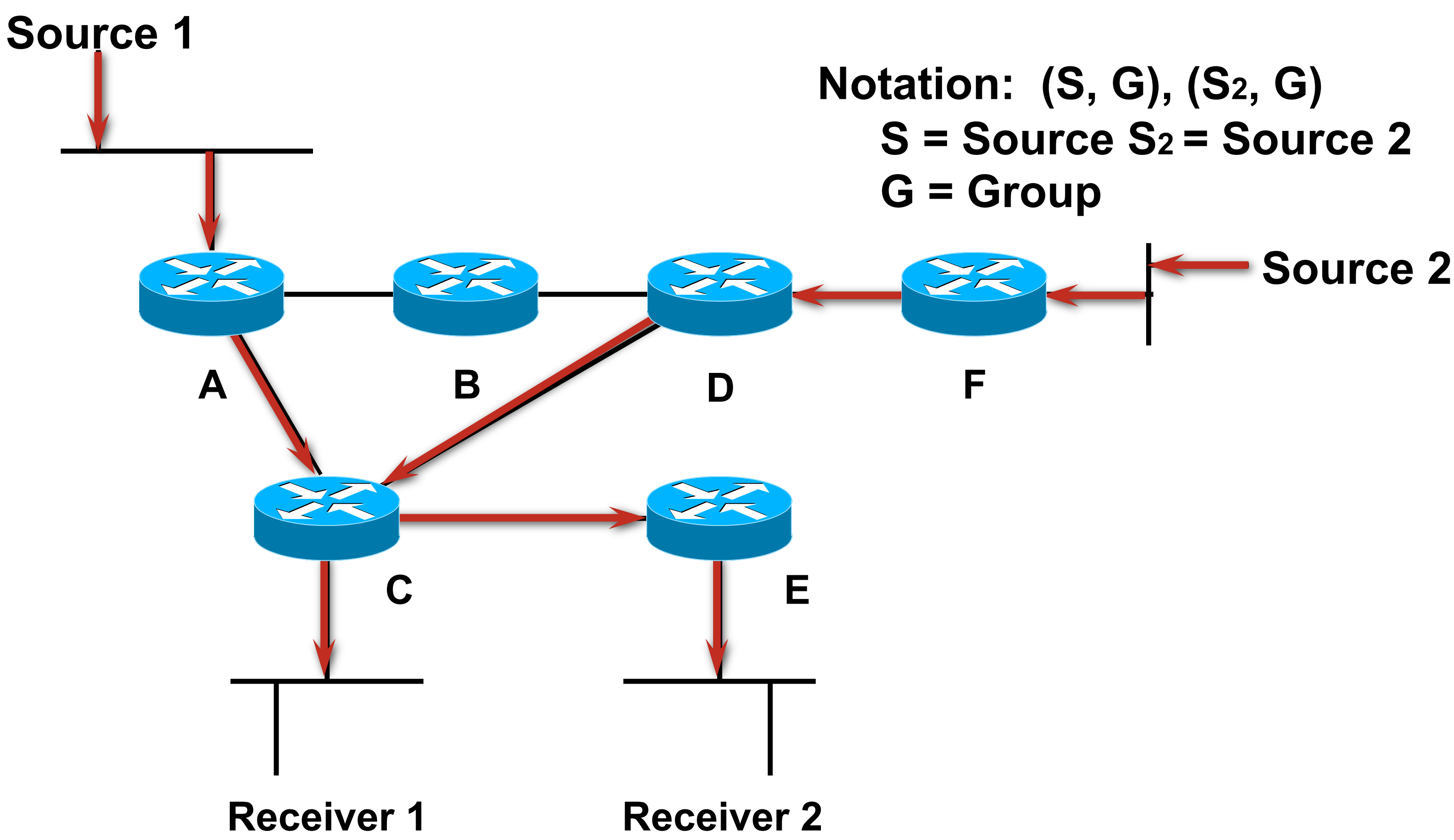
Multicast Distribution Trees

Shortest Path or Source Tree



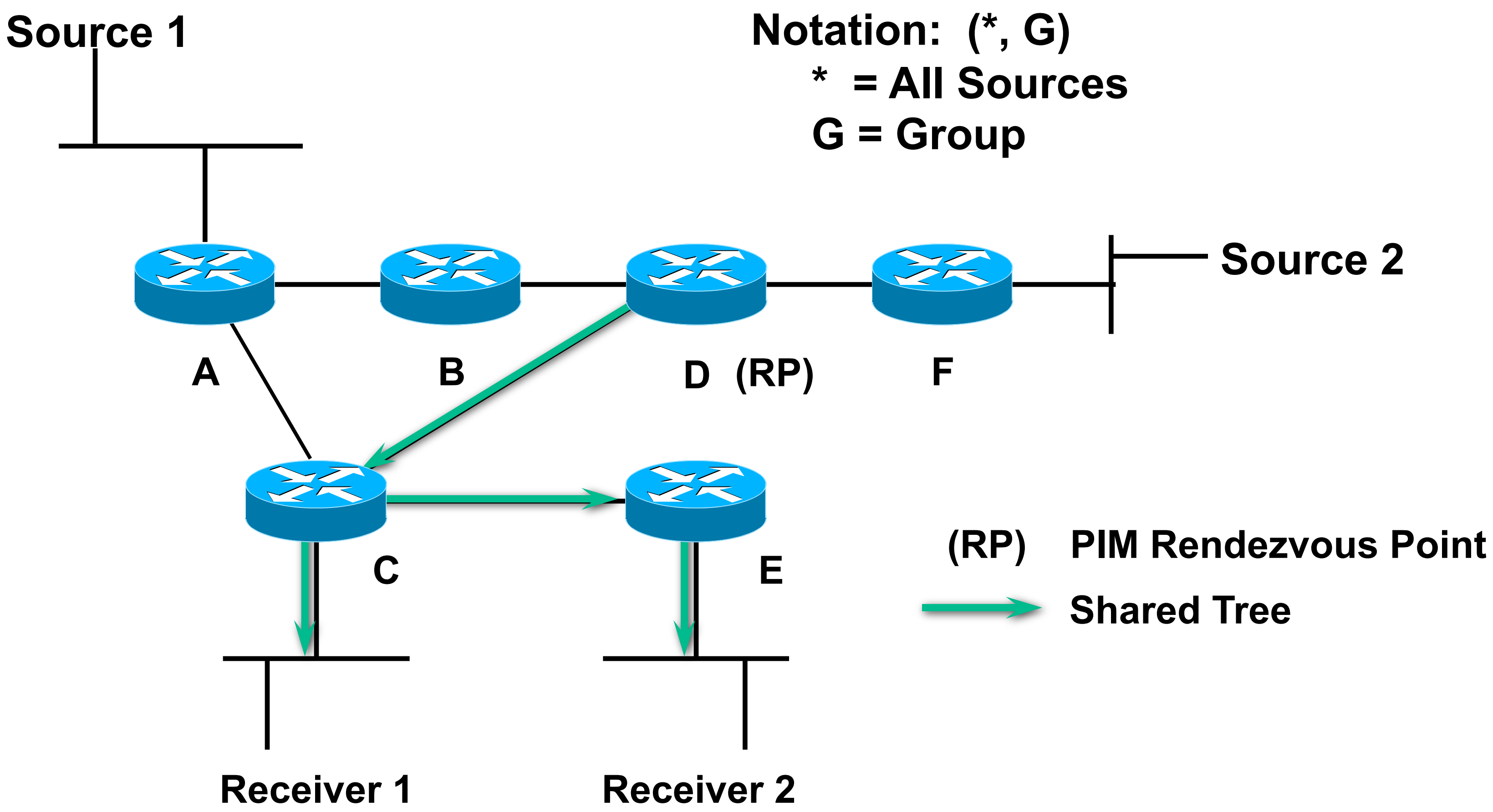
Multicast Distribution Trees

Shortest Path or Source Tree



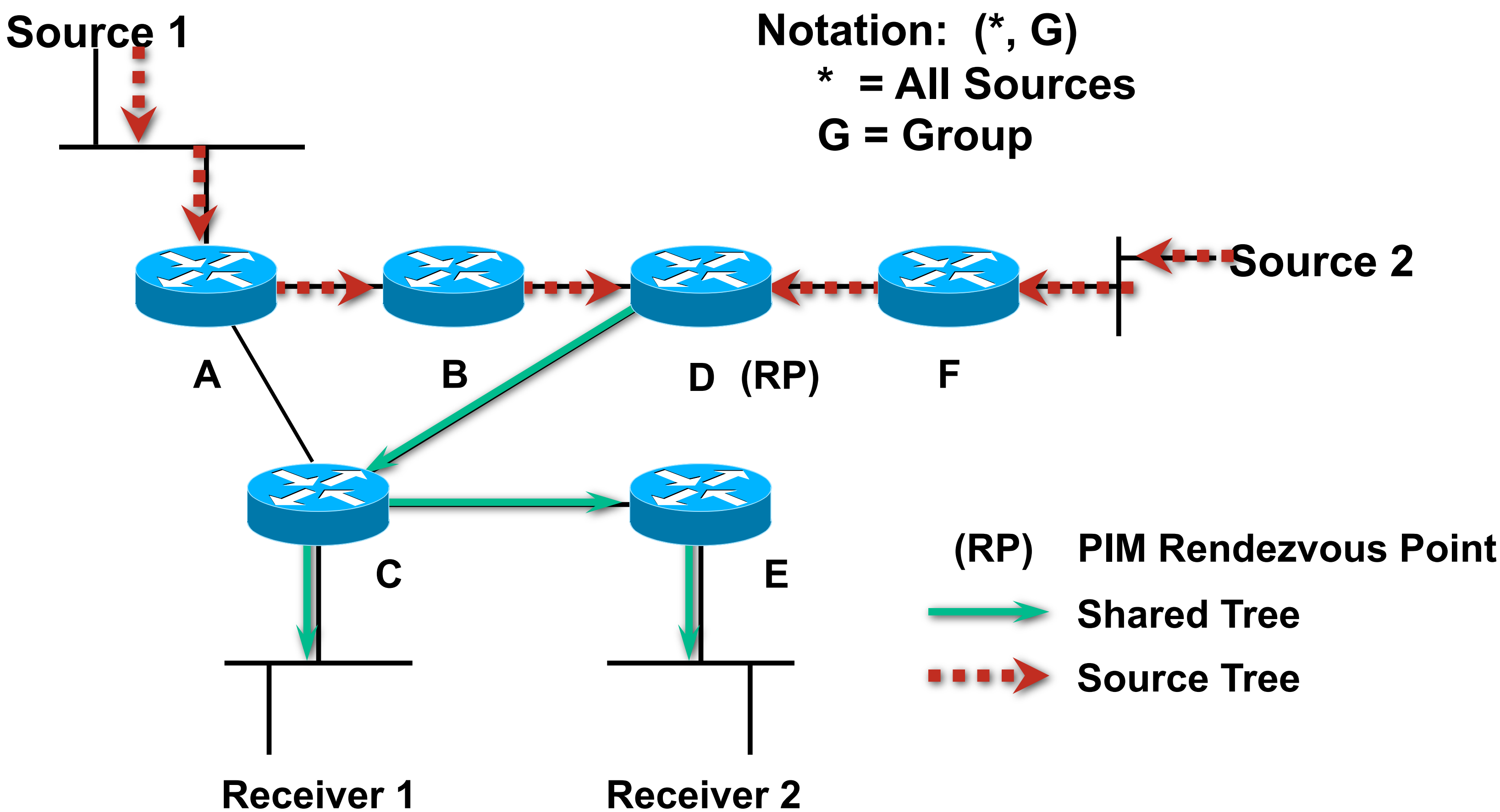
Multicast Distribution Trees

Shared Tree



Multicast Distribution Trees

Shared Tree



Multicast Distribution Trees

Characteristics of Distribution Trees

- Shortest Path trees

  Uses more memory n(S x G) but you get optimal paths from source to all receivers; minimizes delay

- Shared trees

  Uses less memory n(G) but you may get sub-optimal paths from source to all receivers; may introduce extra delay

Multicast Forwarding

Unicast vs. Multicast Forwarding

- Unicast Forwarding
  - Destination IP address directly indicates where to forward packet
  - Forwarding is hop-by-hop
    - Unicast routing table determines interface and next-hop router to forward packet

Unicast vs. Multicast Forwarding

- Multicast Forwarding
  - Destination IP address (group) doesn’t directly indicate where to forward packet
  - Forwarding is connection-oriented
    - Receivers must first be “connected” to the source before traffic begins to flow
      - Connection messages (PIM Joins) follow unicast routing table toward multicast source
      - Build Multicast Distribution Trees that determine where to forward packets
      - Distribution Trees rebuilt dynamically in case of network topology changes

Reverse Path Forwarding (RPF)

- The RPF Calculation

  The multicast source address is checked against the unicast routing table

  This determines the interface and upstream router in the direction of the source to which PIM Joins are sent

  This interface becomes the "Incoming" or RPF interface

  - A router forwards a multicast datagram only if received on the RPF interface

Reverse Path Forwarding (RPF)

- RPF Calculation

  Based on Address of tree root

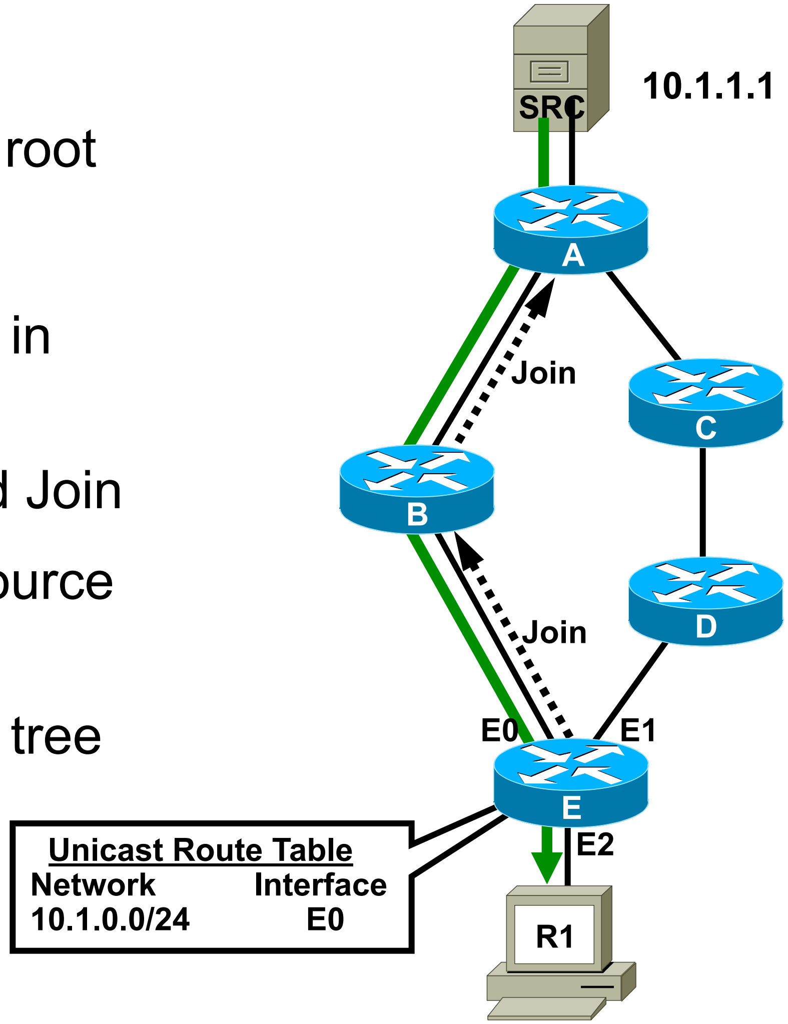
    Source or RP

  Best path to source found in Unicast Route Table

  Determines where to send Join

  Joins continue towards Source to build multicast tree

  Multicast data flows down tree



Reverse Path Forwarding (RPF)

- RPF Calculation

  Based on Address of tree root

    Source or RP

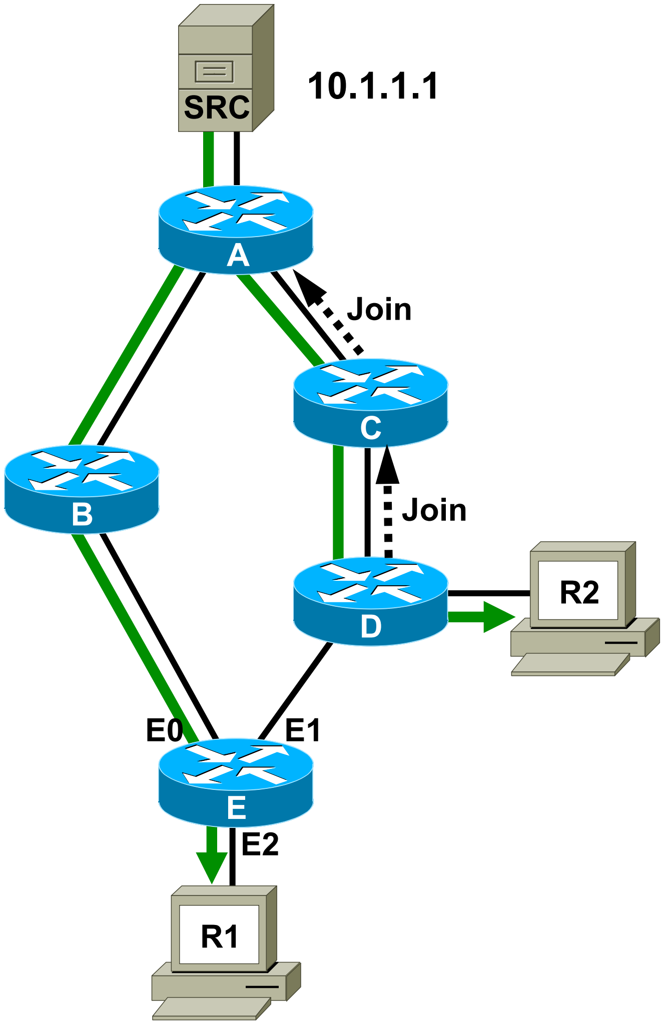
  Best path to source found in Unicast Route Table

  Determines where to send Join

  Joins continue towards Source to build multicast tree

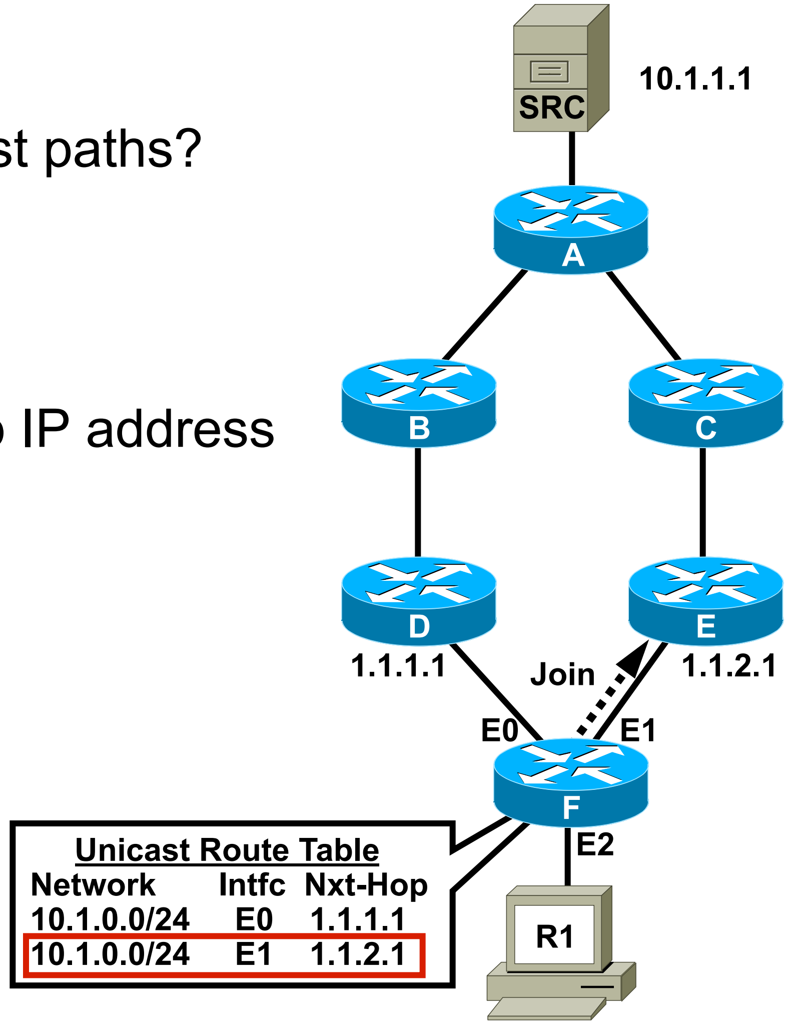
  Multicast data flows down tree

  Repeat for other receivers

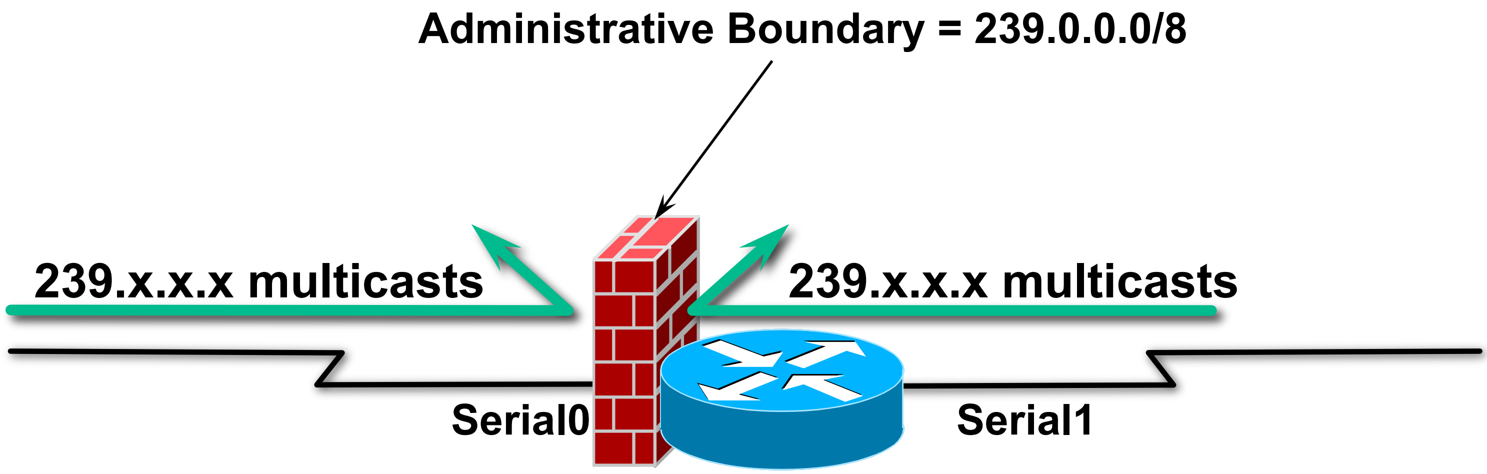


Reverse Path Forwarding (RPF)

- RPF Calculation
  - What if we have equal-cost paths?
    - We can't use both
  - Tie-Breaker
    - Use highest Next-Hop IP address

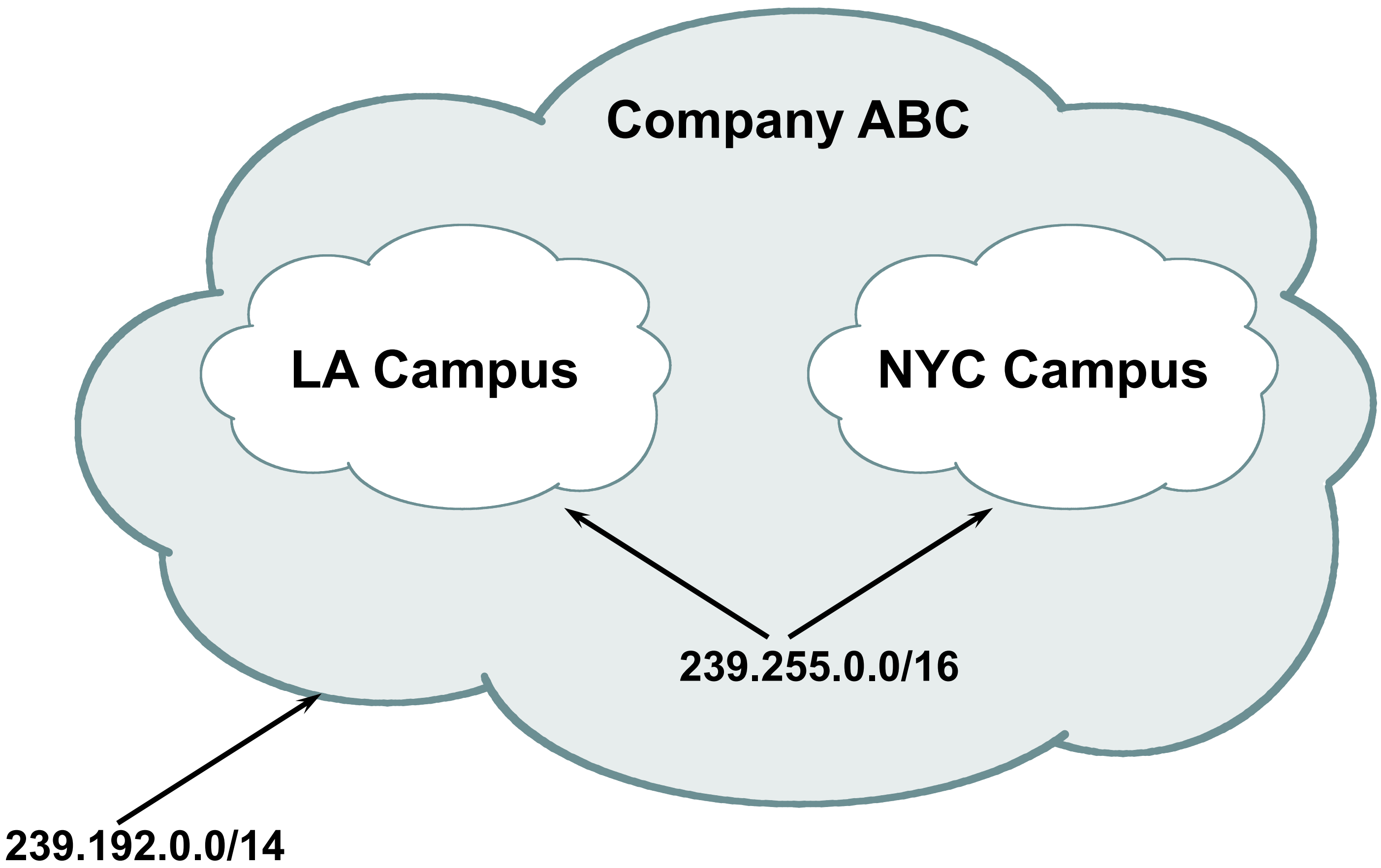


Administrative Boundaries



- Configured using the **ip multicast boundary <acl>** interface command

Administrative Boundaries



Multicast Protocol Basics

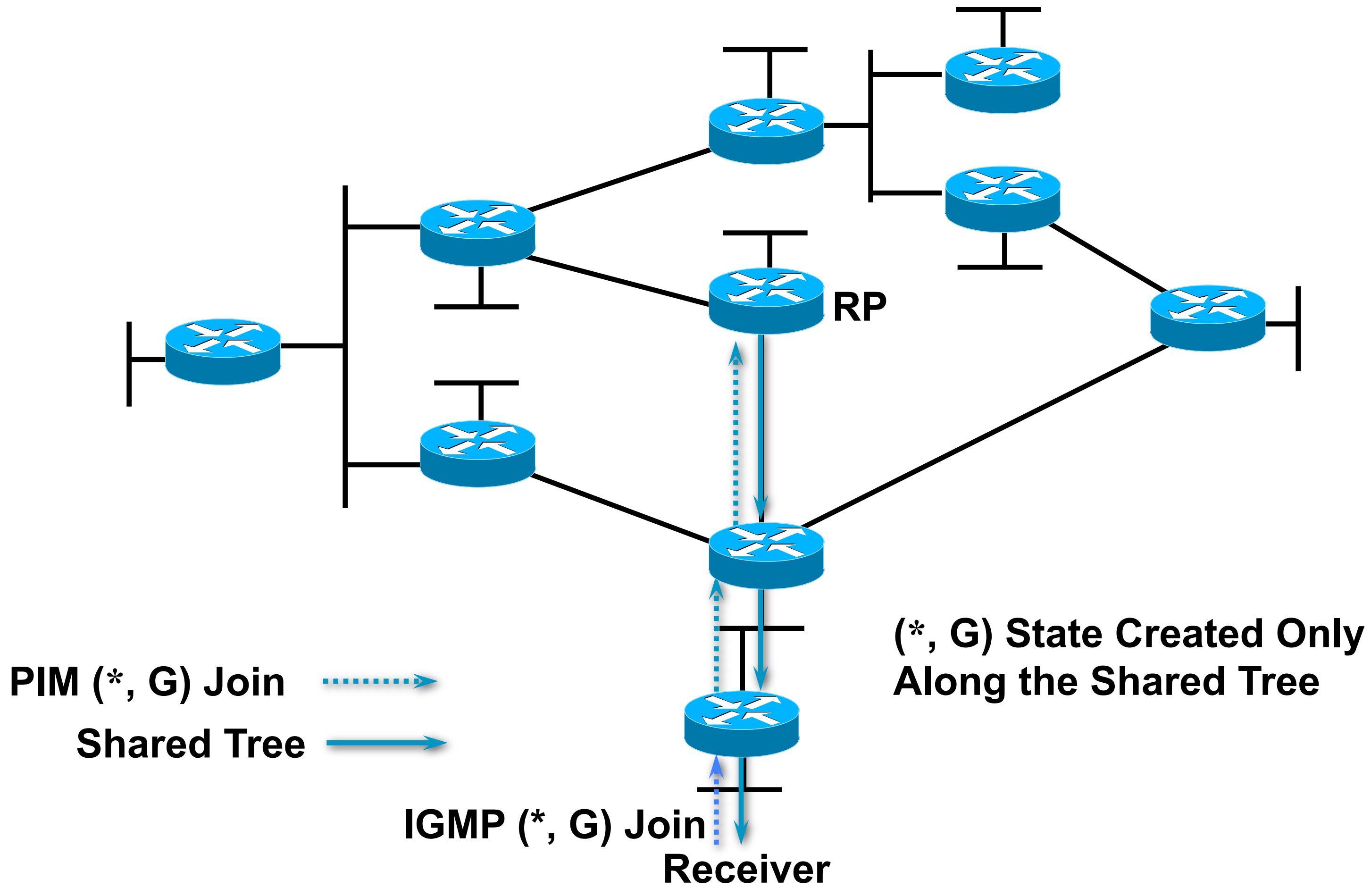
Types of Multicast Protocols

- Dense-mode
  - Uses “Push” model
  - Traffic flooded throughout network
  - Pruned back where it is unwanted
  - Flood and prune behavior (typically every three minutes)
- Sparse-mode
  - Uses “Pull” model
  - Traffic sent only to where it is requested
  - Explicit Join behavior

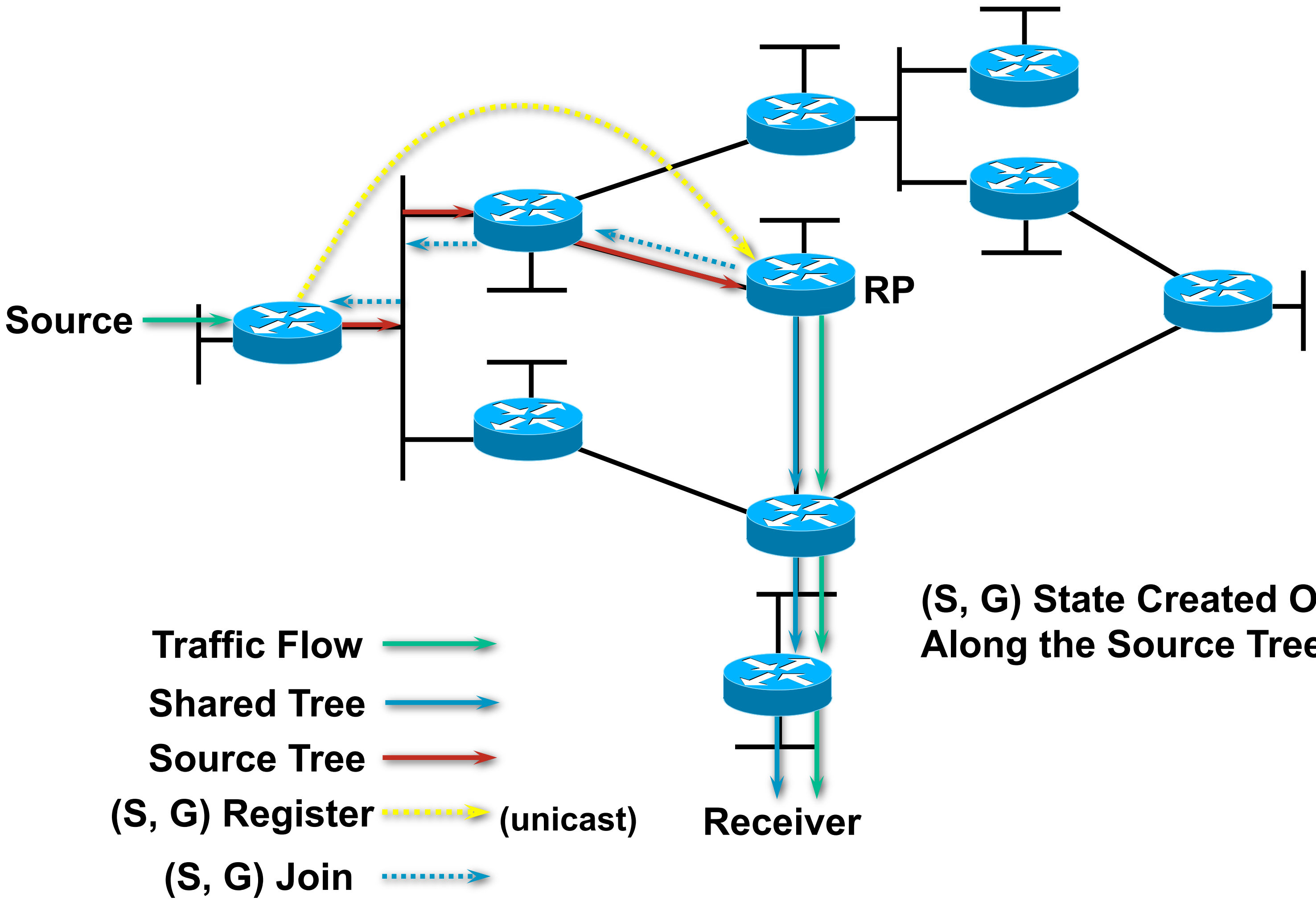
PIM-SM (RFC 4601)

- Supports both source and shared trees
  - Assumes no hosts want multicast traffic unless they specifically ask for it
- Uses a Rendezvous Point (RP)
  - Senders and Receivers “rendezvous” at this point to learn of each others existence
    - Senders are “registered” with RP by their first-hop router
    - Receivers are “joined” to the Shared Tree (rooted at the RP) by their local Designated Router (DR)
- Appropriate for …
  - Wide scale deployment for **both** densely and sparsely populated groups in the enterprise
  - Optimal choice for all production networks regardless of size and membership density

PIM-SM Shared Tree Join

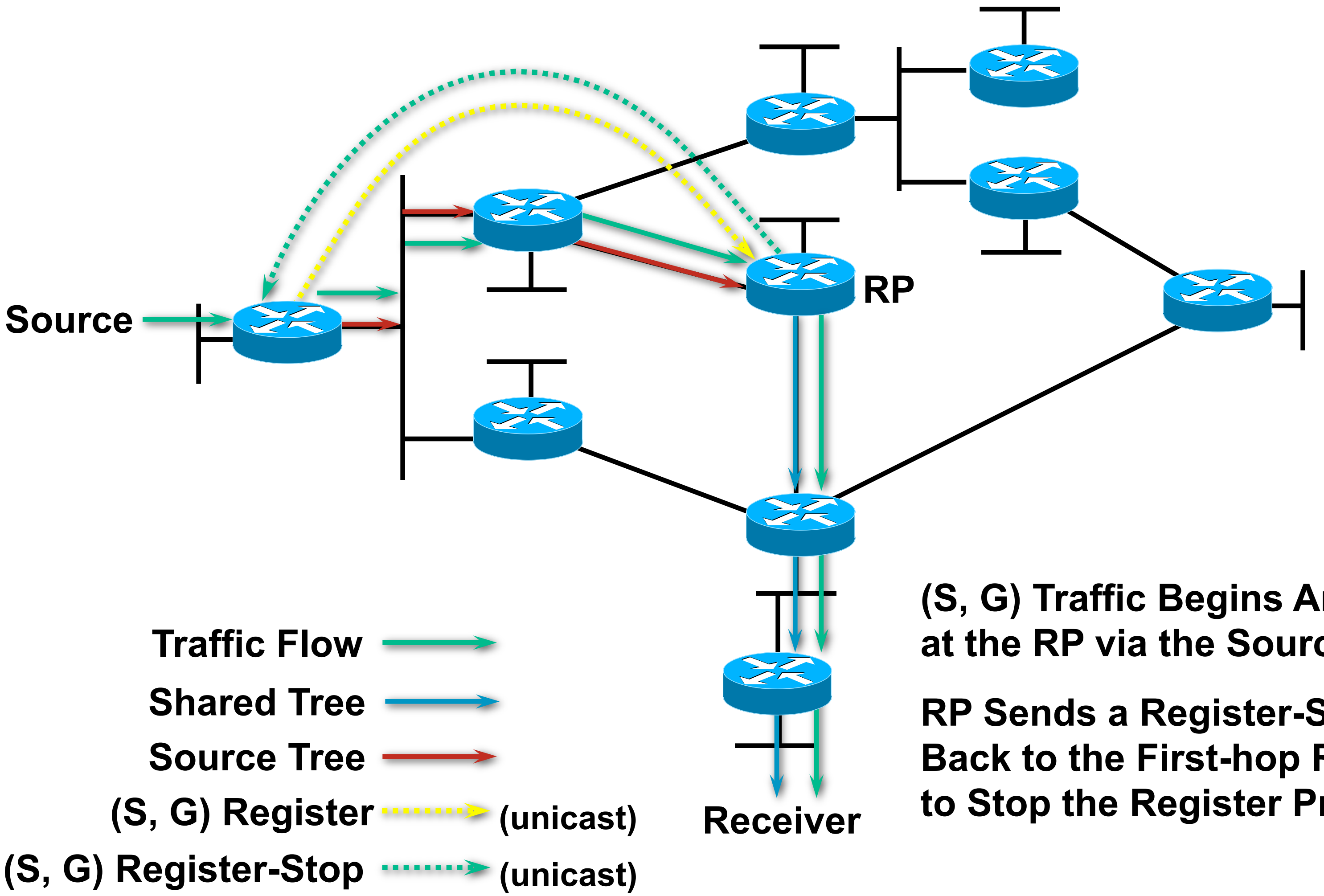


PIM-SM Sender Registration



(S, G) State Created Only Along the Source Tree

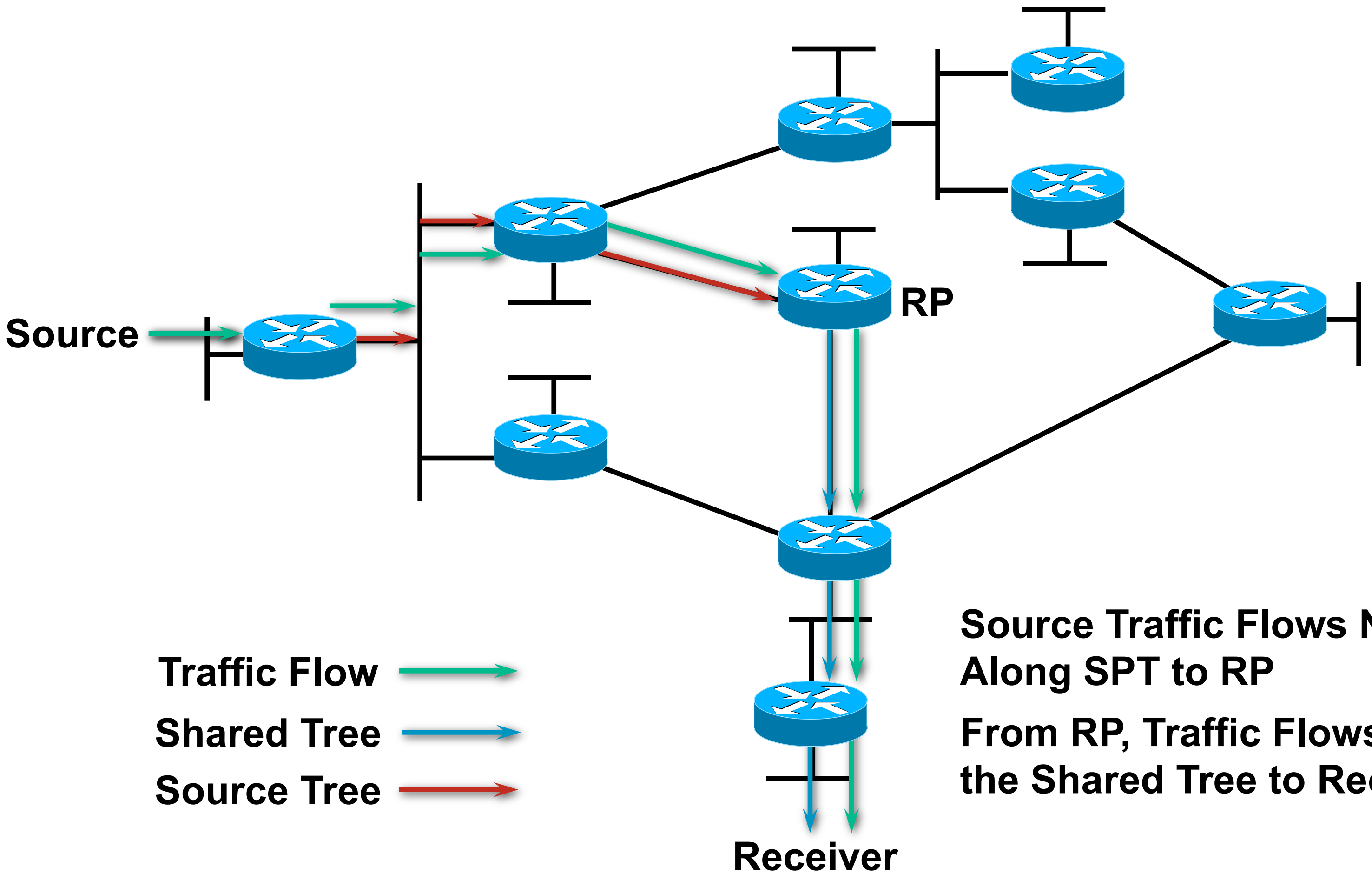
PIM-SM Sender Registration



(S, G) Traffic Begins Arriving at the RP via the Source Tree

RP Sends a Register-Stop Back to the First-hop Router to Stop the Register Process

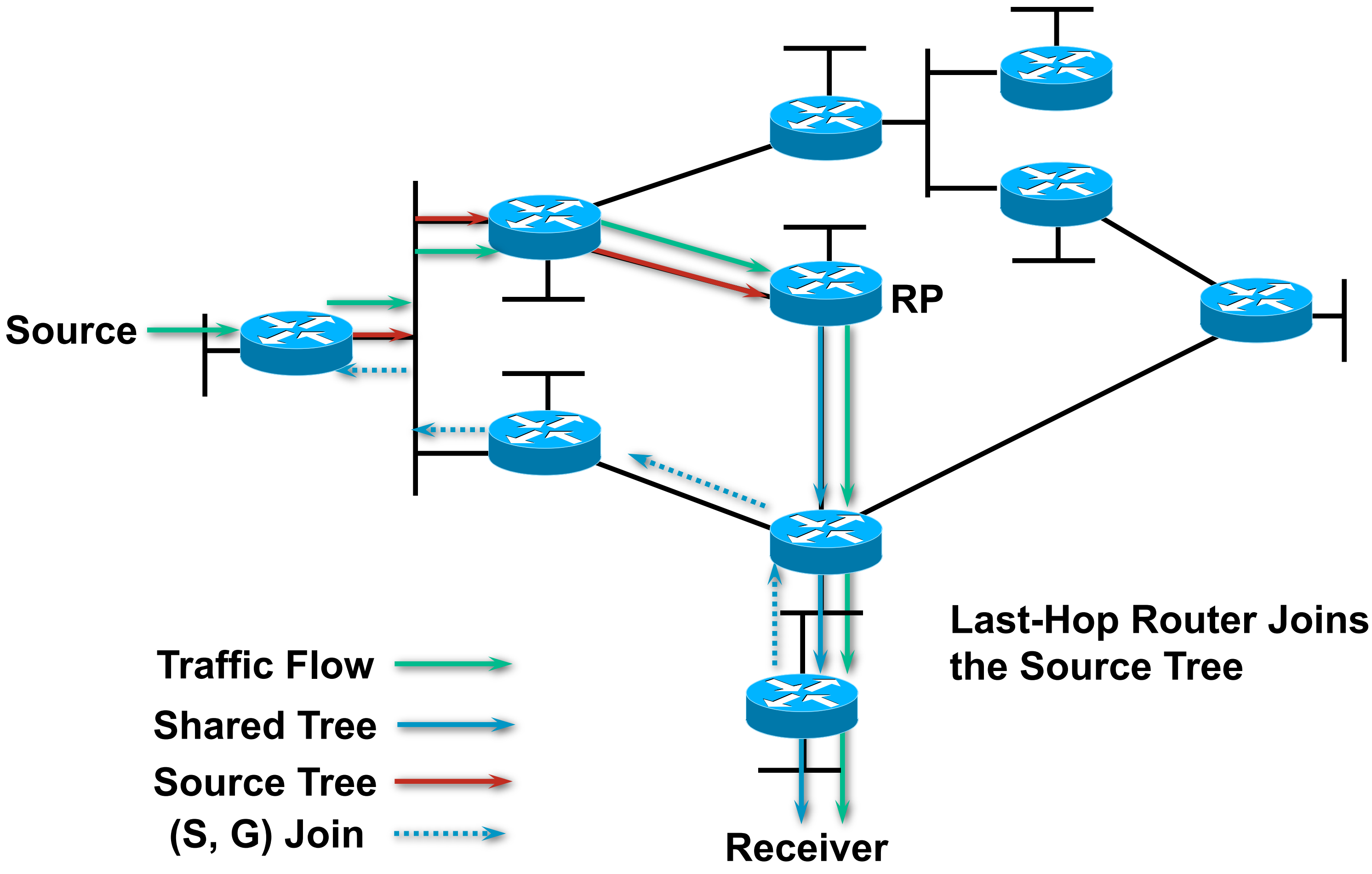
PIM-SM Sender Registration



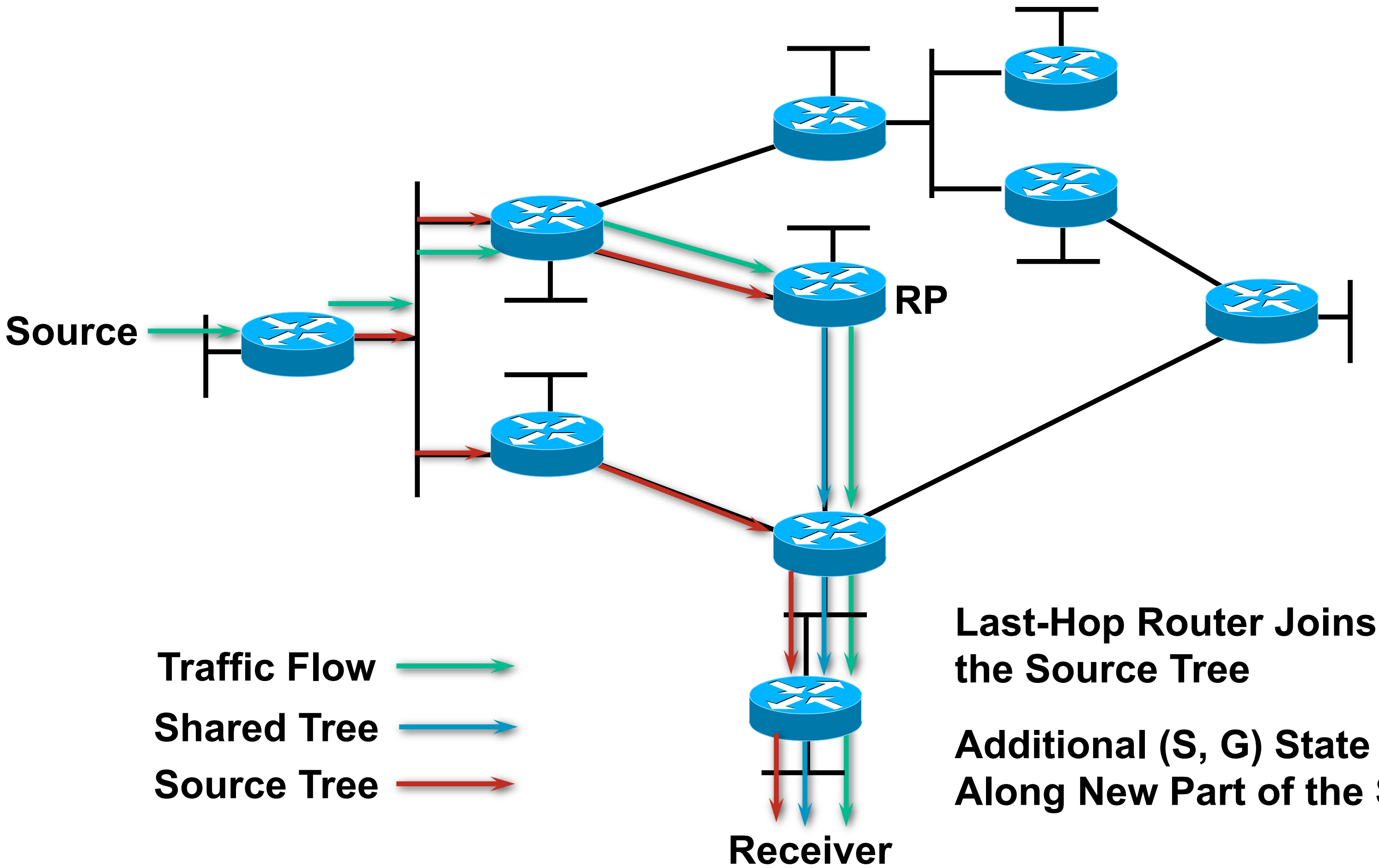
Source Traffic Flows Natively Along SPT to RP

From RP, Traffic Flows Down the Shared Tree to Receivers

PIM-SM SPT Switchover



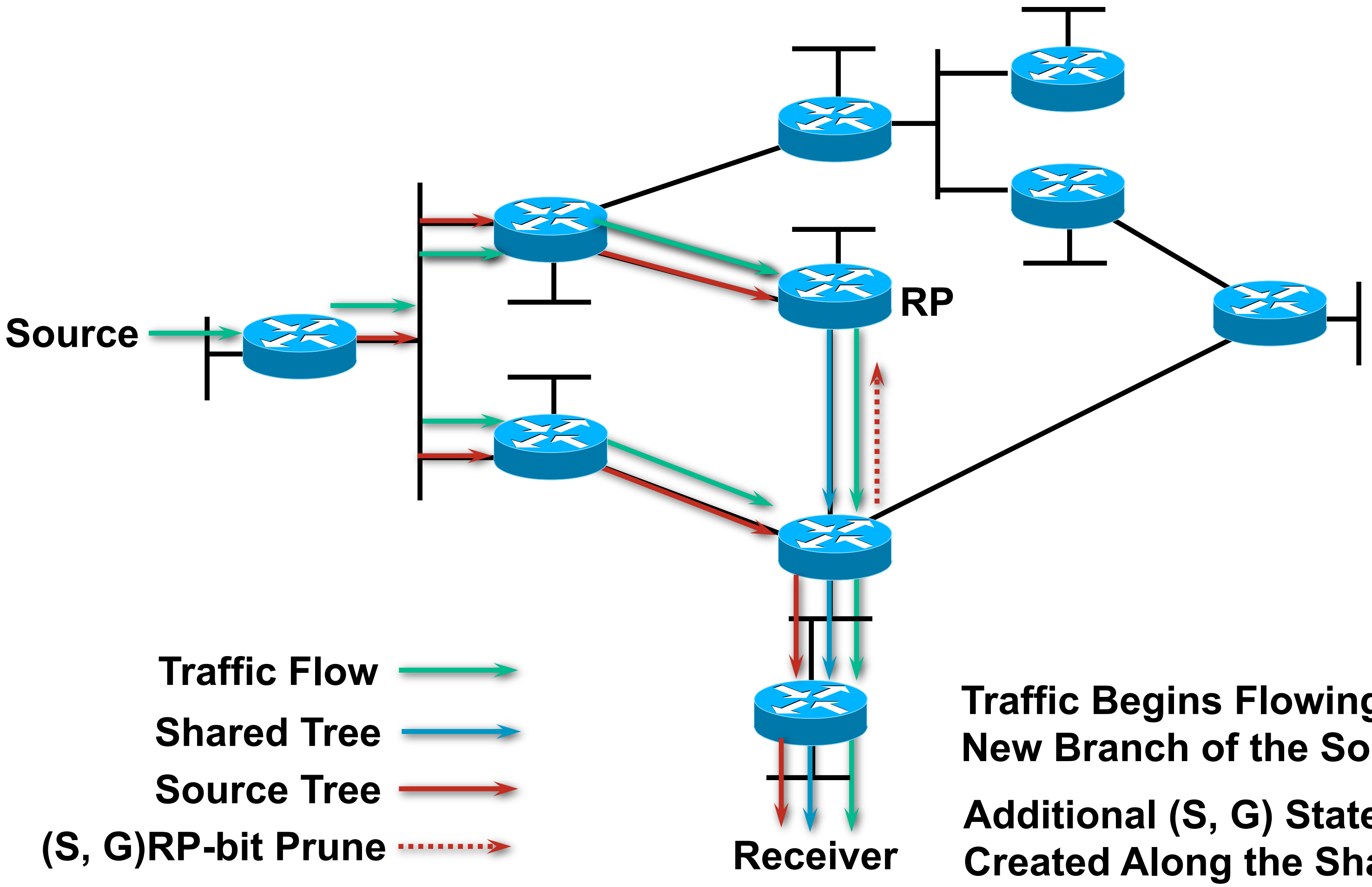
PIM-SM SPT Switchover



Last-Hop Router Joins the Source Tree

Additional (S, G) State Is Created Along New Part of the Source Tree

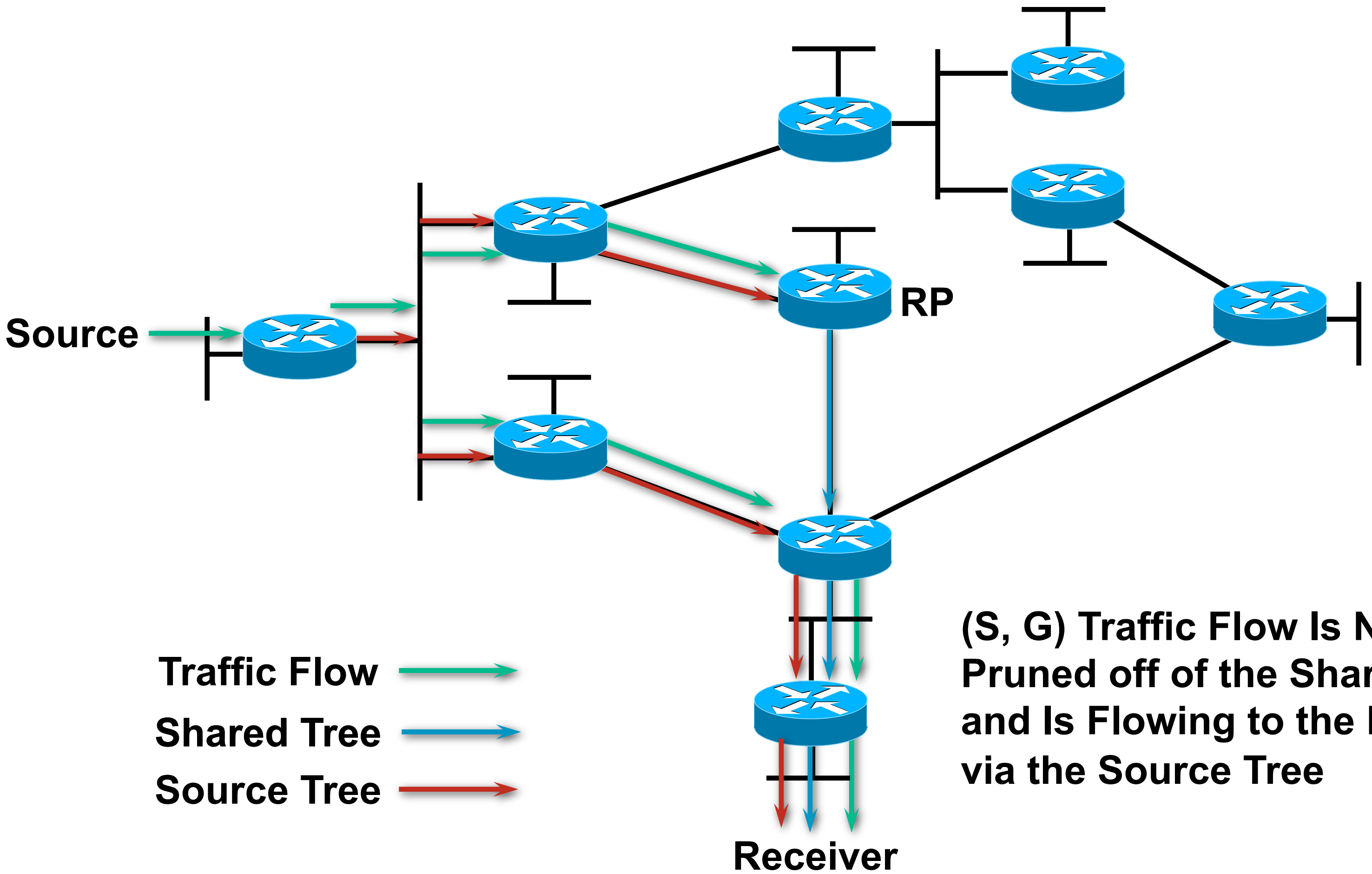
PIM-SM SPT Switchover



Traffic Begins Flowing Down the New Branch of the Source Tree

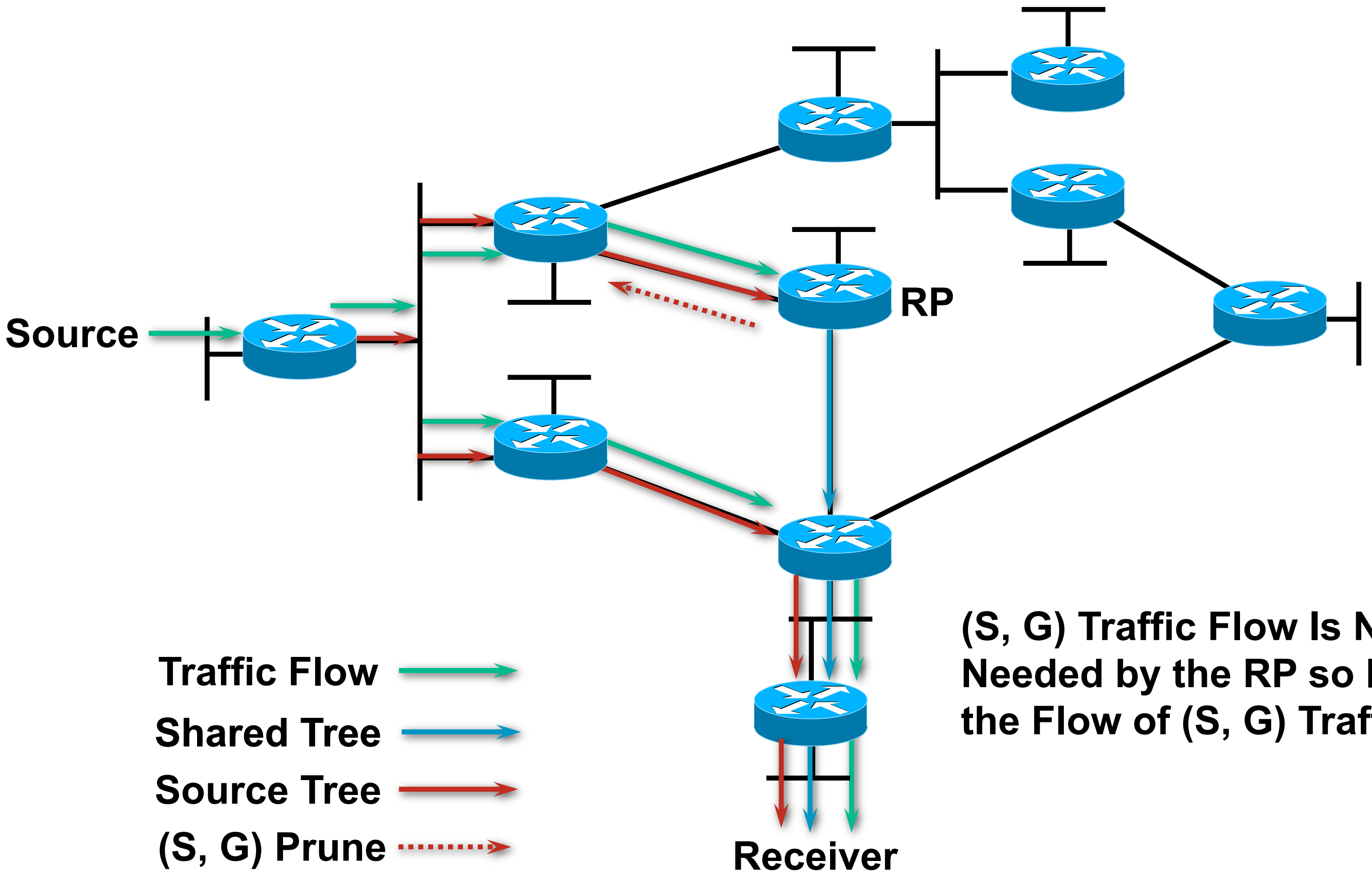
Additional (S, G) State Is Created Along the Shared Tree to Prune off (S, G) Traffic

PIM-SM SPT Switchover



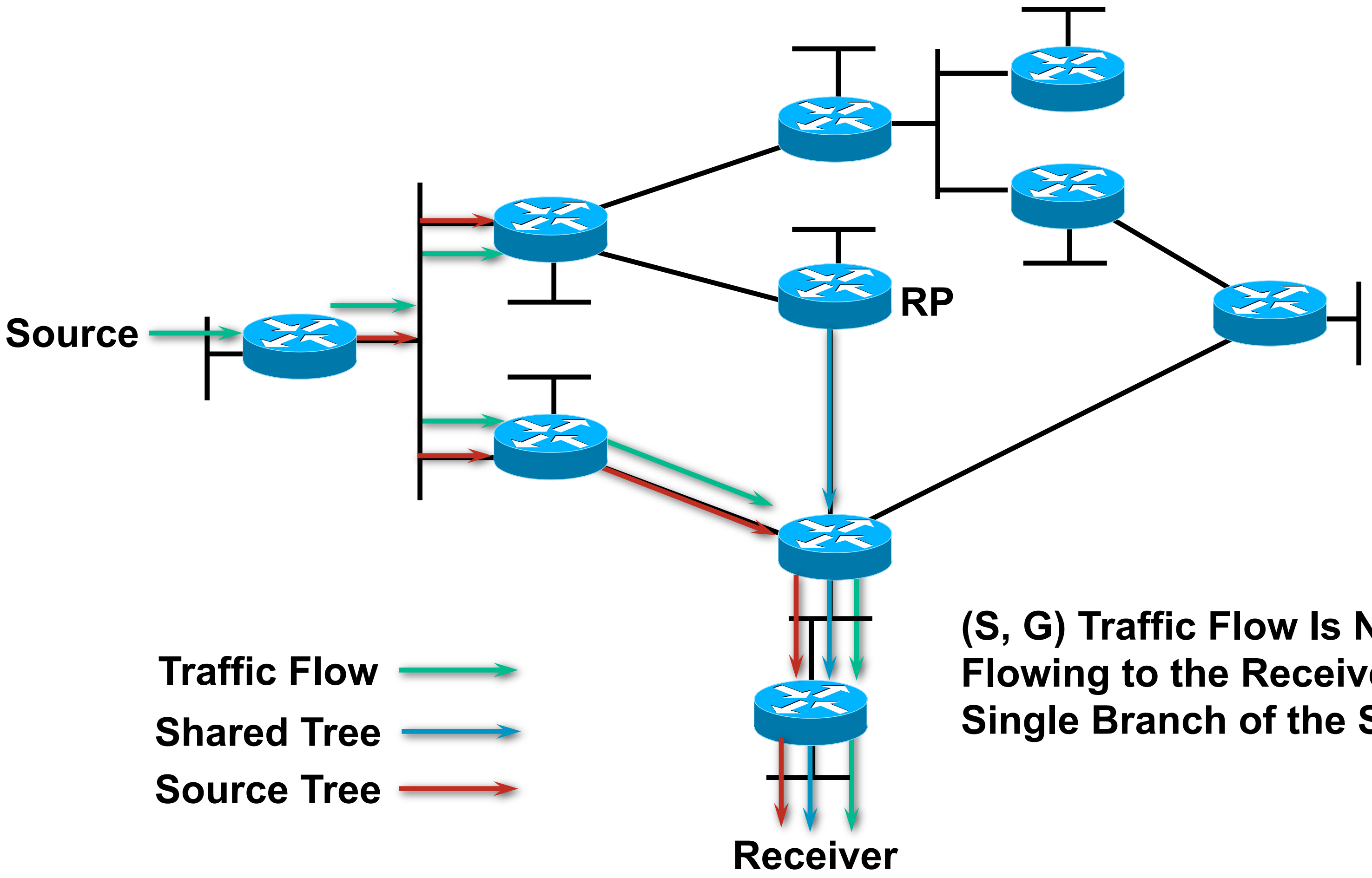
(S, G) Traffic Flow Is Now Pruned off of the Shared Tree and Is Flowing to the Receiver via the Source Tree

PIM-SM SPT Switchover



(S, G) Traffic Flow Is No Longer Needed by the RP so It Prunes the Flow of (S, G) Traffic

PIM-SM SPT Switchover



(S, G) Traffic Flow Is Now Only Flowing to the Receiver via a Single Branch of the Source Tree

PIM-SM FFF

PIM-SM Frequently Forgotten Fact

"The default behavior of PIM-SM is that routers with directly connected members will join the Shortest Path Tree as soon as they detect a new multicast source."

PIM-SM—Evaluation

- Advantages:
  - Traffic only sent down “joined” branches
  - Can switch to optimal source-trees for high traffic sources dynamically
  - Unicast routing protocol-independent
  - Basis for inter-domain multicast routing
    - When used with MBGP and MSDP
- Disadvantages
  - Few if any
- Primary application
  - All production multicast networks with sparse or dense distribution of receivers

Protocol Summary

Conclusion

"Sparse mode good, dense mode bad!"

R. Davis

"The Caveman's Guide to IP Multicast", 2000

IP Multicast at Layer 2

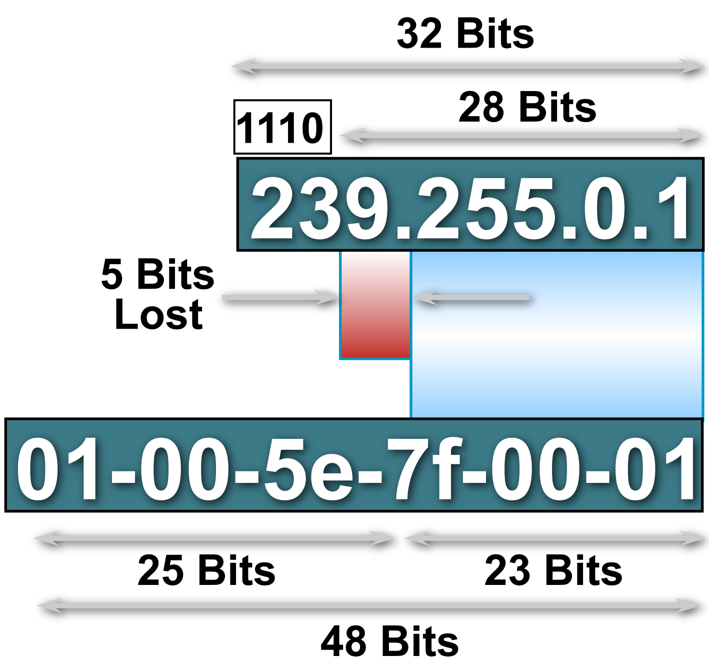
Module Agenda

- MAC Layer Multicast Addresses
- IGMPv2
- IGMPv3
- L2 Multicast Frame Switching
  - IGMP Snooping
  - PIM Snooping

MAC Layer Multicast Addresses

Layer 2 Multicast Addressing

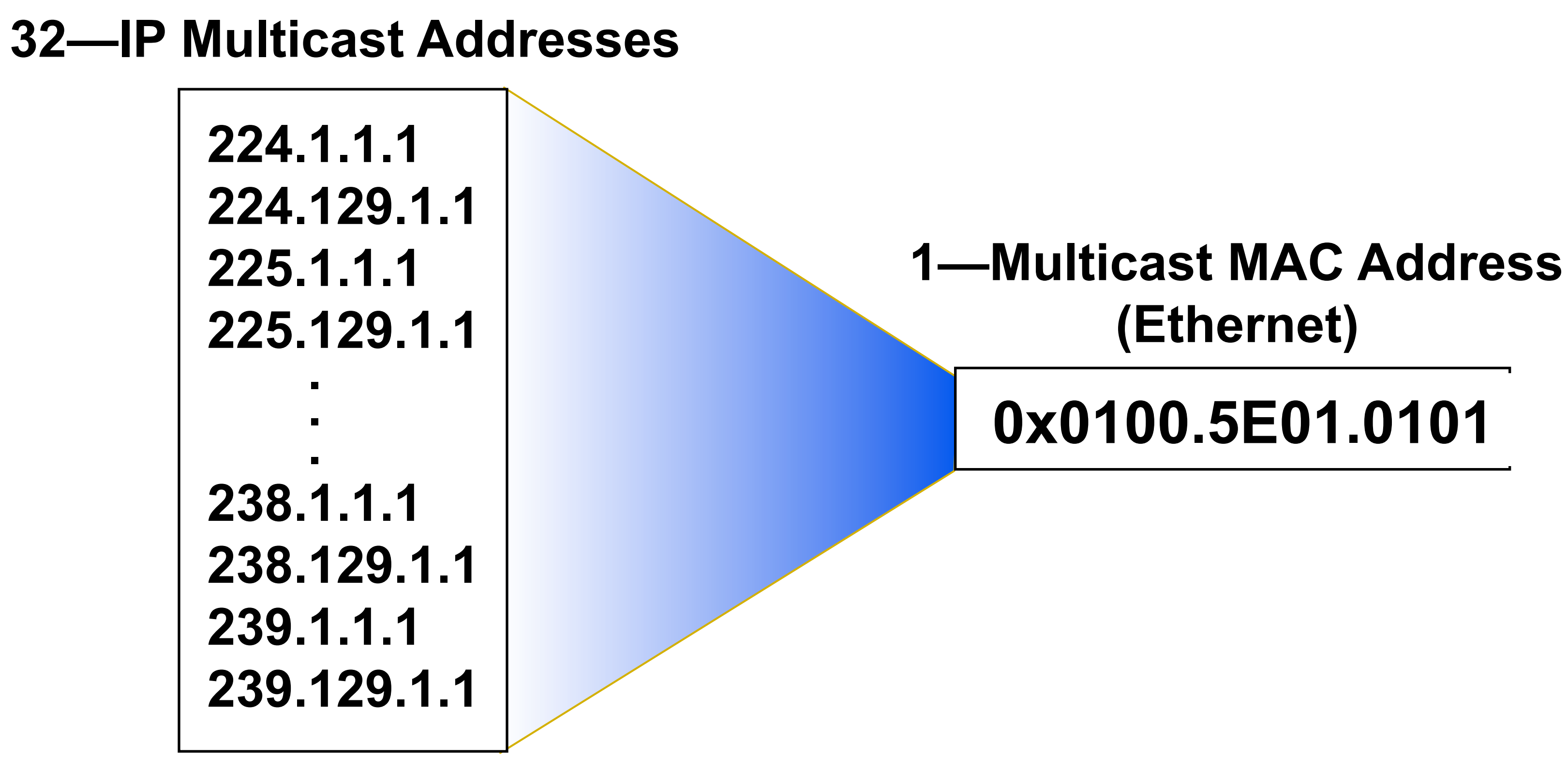
IP Multicast MAC Address Mapping



Layer 2 Multicast Addressing

IP Multicast MAC Address Mapping

Be aware of the 32:1 address overlap



IGMPv2

IGMP

- How hosts tell routers about group membership
- Routers solicit group membership from directly connected hosts
- RFC 1112 specifies first version of IGMP
- RFC 2236 specifies IGMPv2
  - Most widely deployed and supported
- RFC 3376 specifies IGMPv3
  - Growing support (required for SSM)

IGMPv2

RFC 2236

- Membership queries

  Queries sent to 224.0.0.1 with ttl = 1

  One router on LAN is elected to send queries

  Query interval 60–120 seconds

- Membership reports

  IGMP report sent by one host suppresses sending by others

  Restrict to one report per group per LAN

  Unsolicited reports sent by host, when it first joins the group

IGMPv2

RFC 2236

- Group-specific query

  Router sends Group-specific queries to make sure there are no members present before stopping to forward data for the group for that subnet

- Leave Group message

  Host sends leave message if it leaves the group and is the last member (reduces leave latency in comparison to v1)

IGMPv2

RFC 2236

- Querier election mechanism

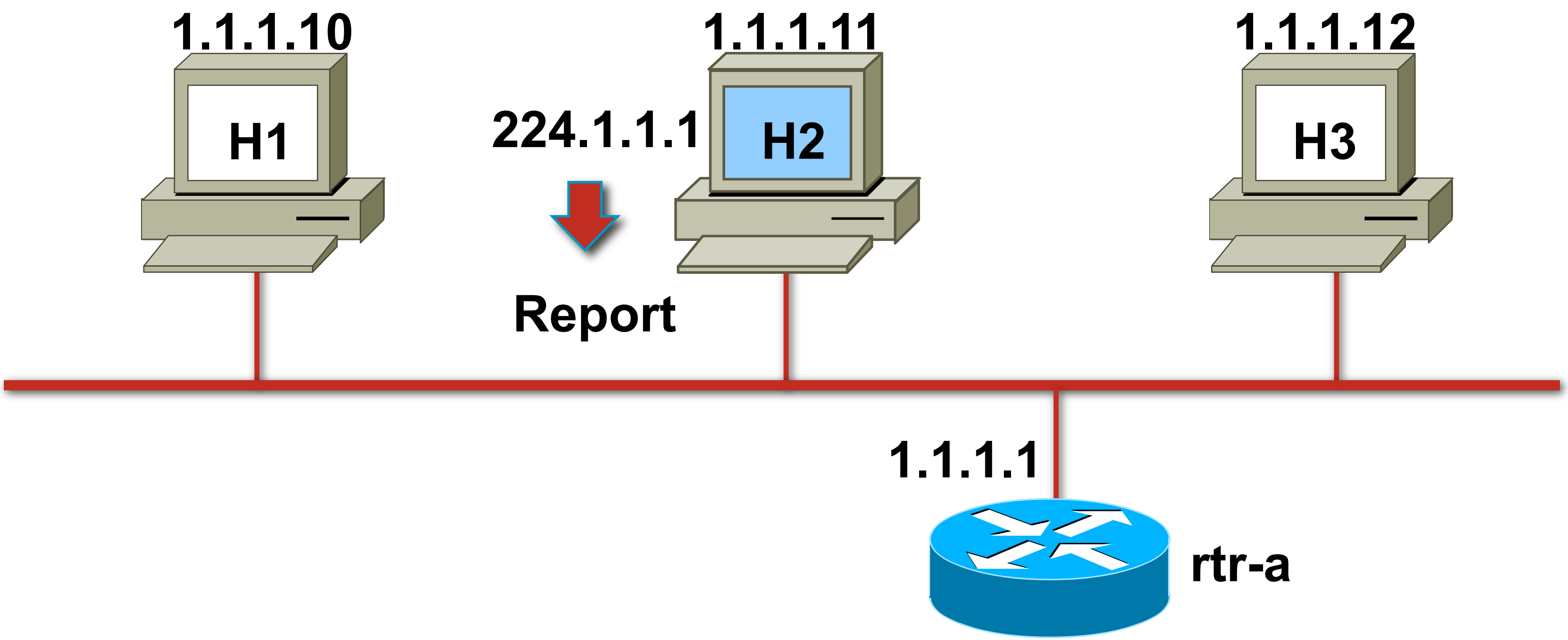
  On multi-access networks, an IGMP querier router is elected based on lowest IP address. Only the querier router sends queries.

- Query-Interval Response Time

  General queries specify “Max. Response Time” which inform hosts of the maximum time within which a host must respond to any query. (improves burstiness of the responses)

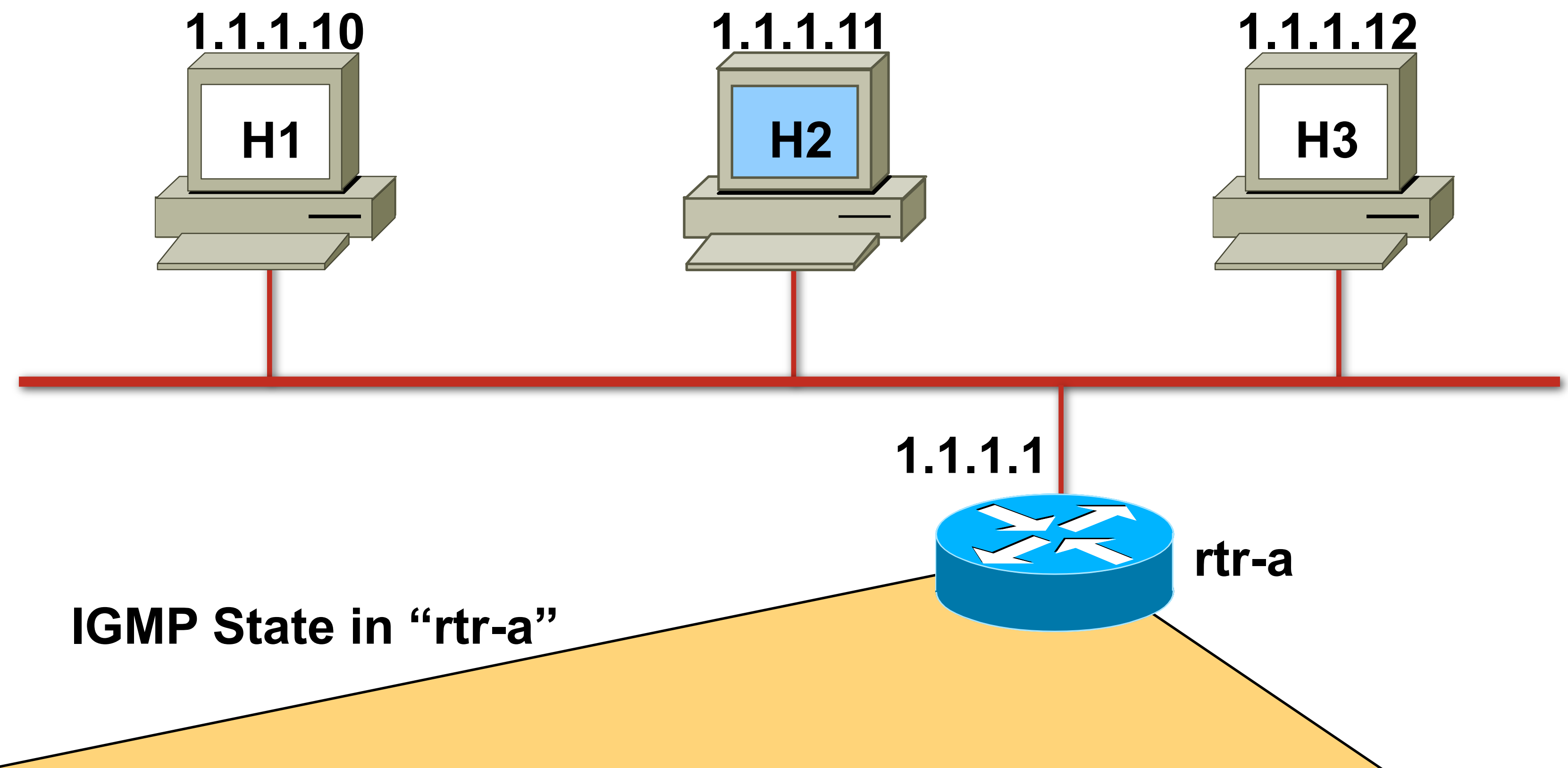
- Backward compatible with IGMPv1

IGMPv2—Joining a Group



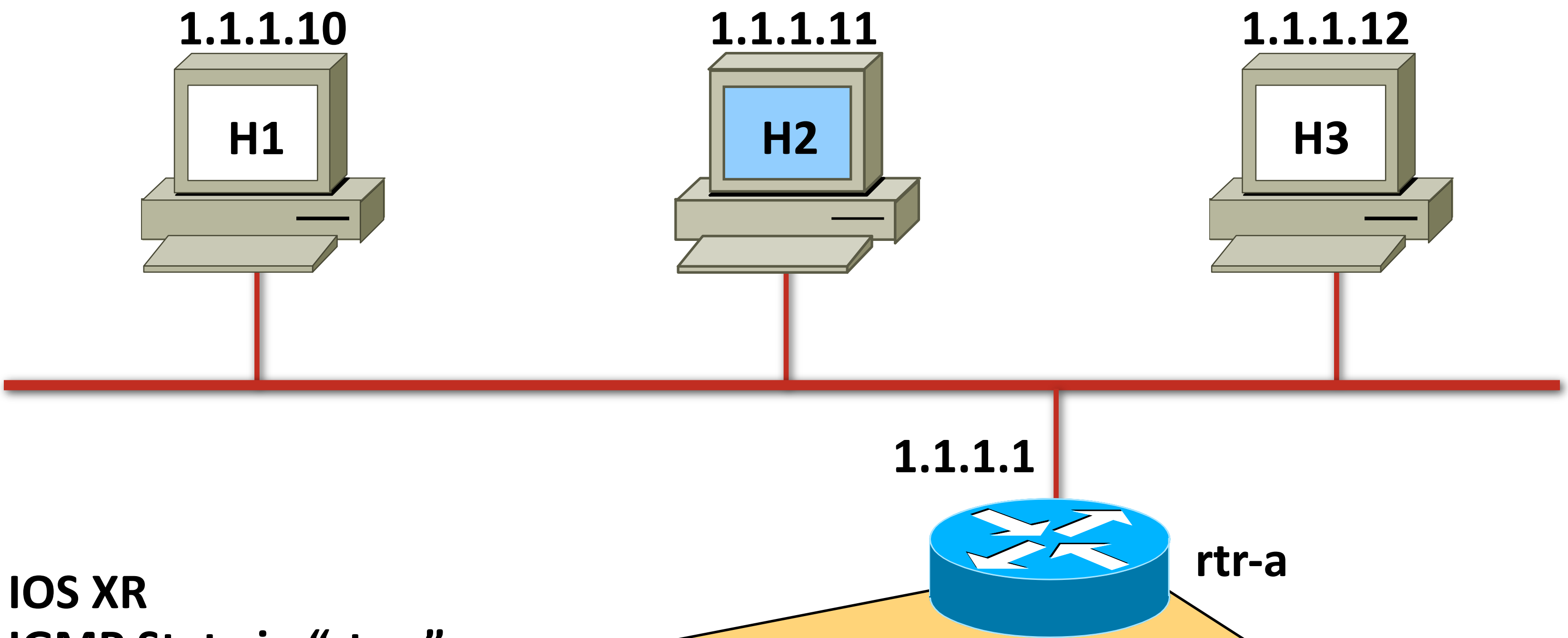
- Joining member sends report to 224.1.1.1 immediately upon joining (same as IGMPv1)

IGMPv2—Joining a Group




```
rtr-a>show ip igmp group
IGMP Connected Group Membership
Group Address     Interface      Uptime     Expires    Last Reporter
224.1.1.1         Ethernet0      6d17h      00:02:31   1.1.1.11
```

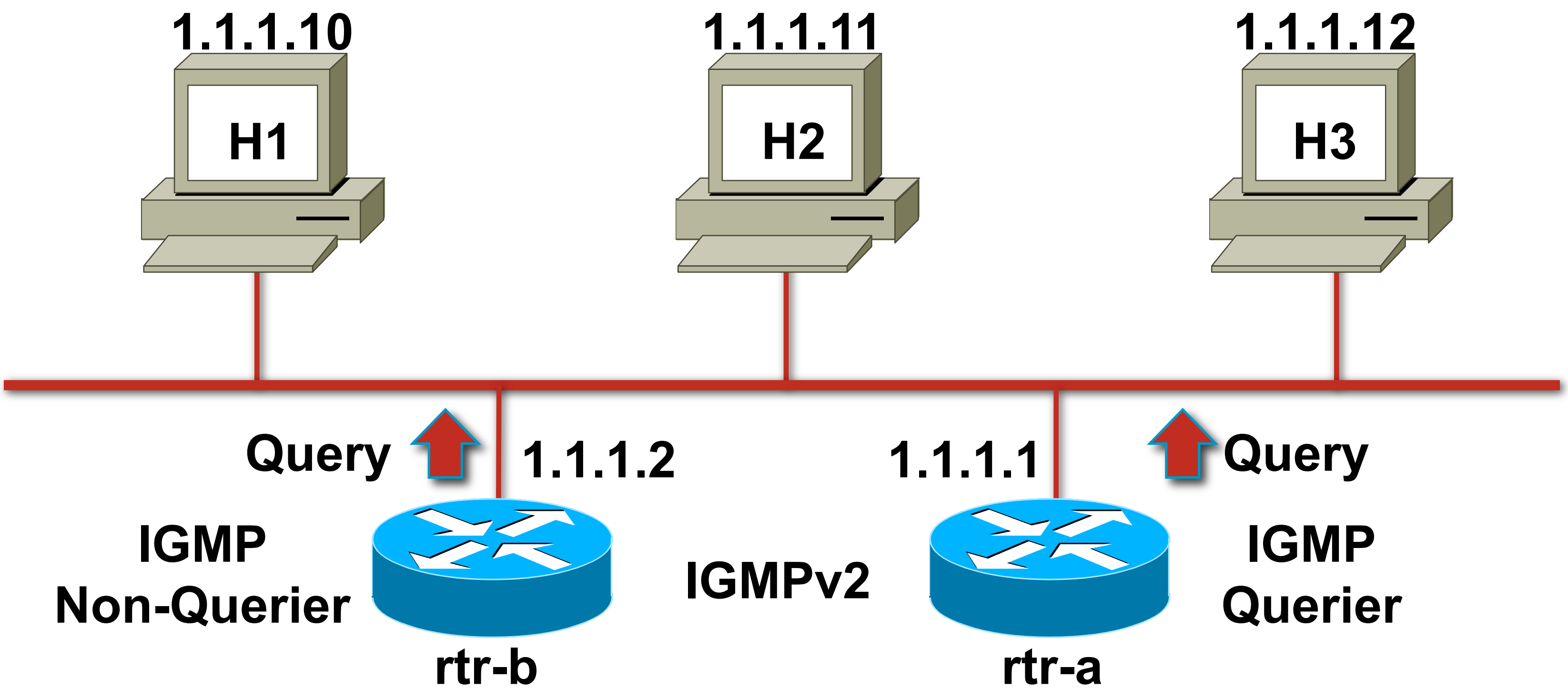
IGMPv2—Joining a Group



IOS XR
IGMP State in "rtr-a"

```
RP/0/RP0/CPU0:rtr-a#show igmp group
IGMP Connected Group Membership
Group Address    Interface                    Uptime    Expires   Last Reporter
224.1.1.1        Ethernet0                    00:00:35  00:01:34  1.1.1.11
```

IGMPv2—Querier Election



- Initially all routers send out a query
- Router with lowest IP address “elected” querier
- Other routers become “non-queriers”

IGMPv2—Querier Election

Determining Which Router Is the IGMP Querier

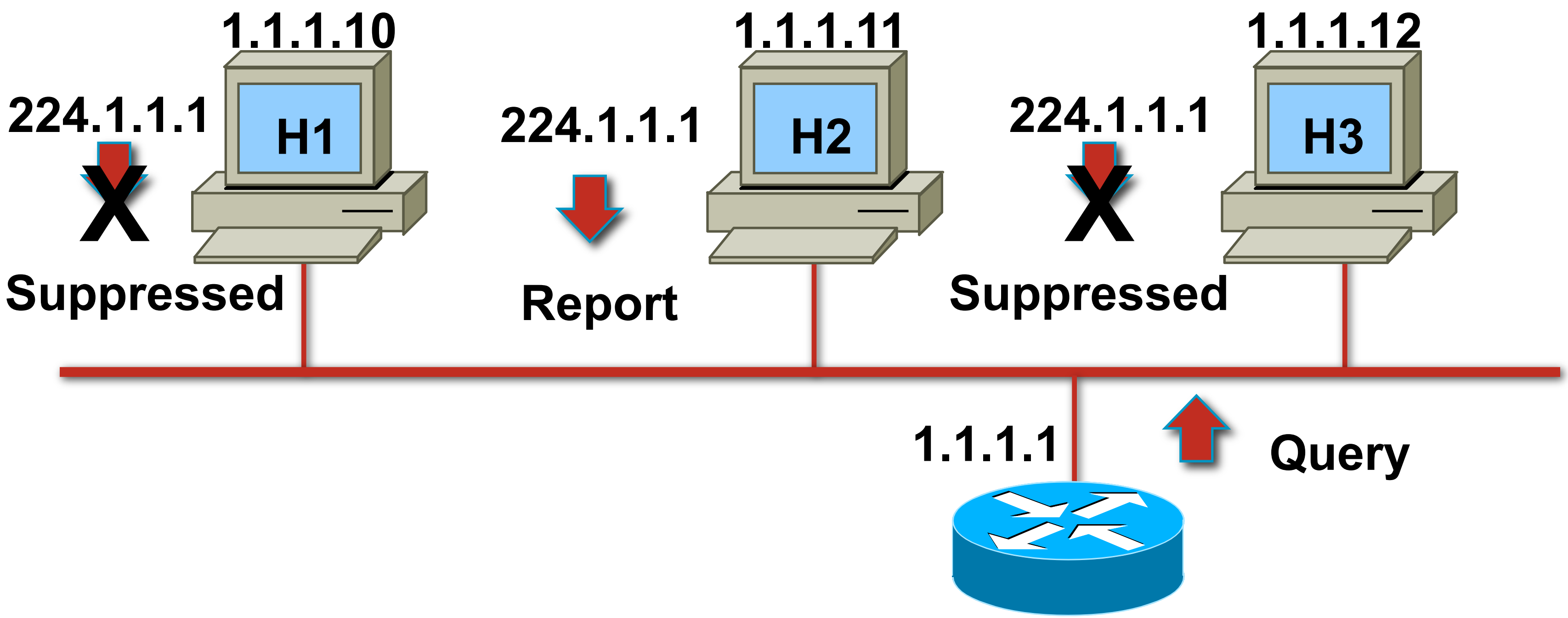
```
rtr-a>show ip igmp interface e0
Ethernet0 is up, line protocol is up
  Internet address is 1.1.1.1, subnet mask is 255.255.255.0
  IGMP is enabled on interface
  Current IGMP version is 2
  CGMP is disabled on interface
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Inbound IGMP access group is not set
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 1.1.1.1 (this system)
  IGMP querying router is 1.1.1.1 (this system)
  Multicast groups joined: 224.0.1.40 224.2.127.254
```

IGMPv2—Querier Election

Determining Which Router Is the IGMP Querier (XR)

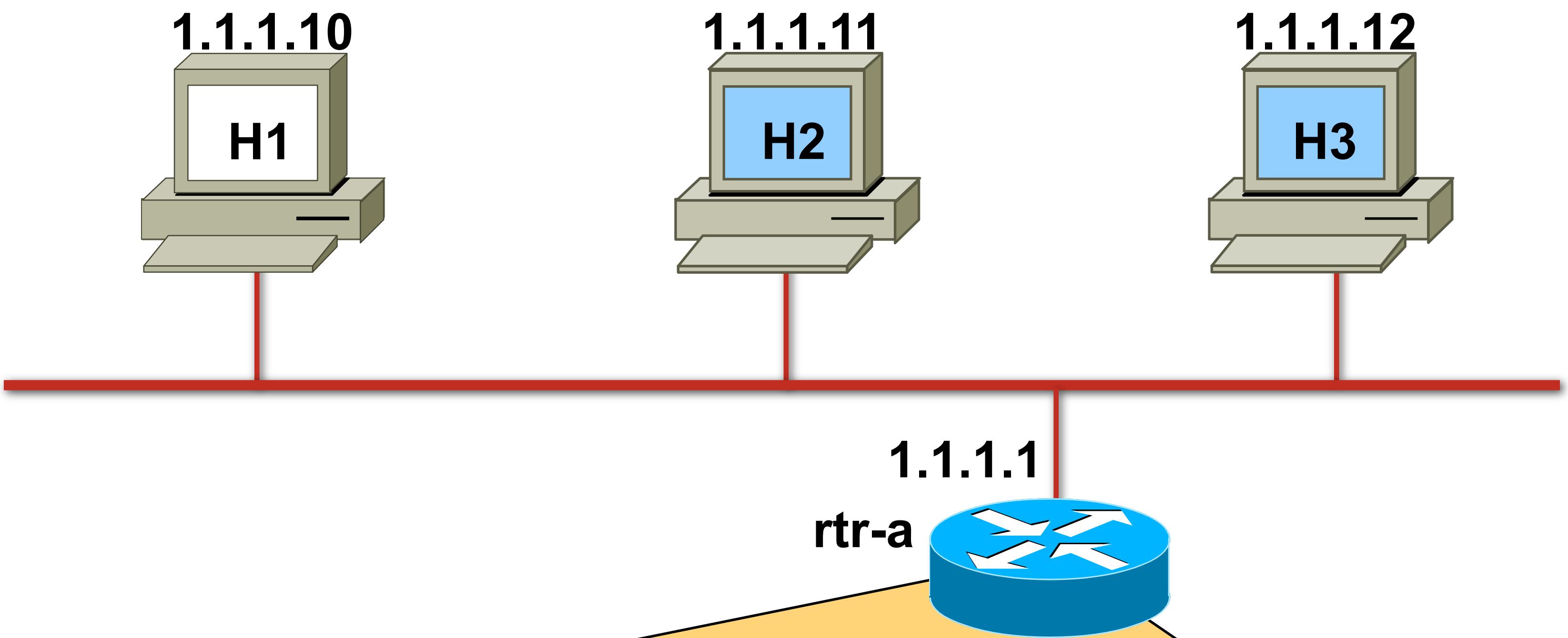
```
RP/0/RP0/CPU0:rtr-a#show igmp interface Ethernet0
Ethernet0 is up, line protocol is up
  Internet address is 1.1.1.1/24
  IGMP is enabled on interface
  Current IGMP version is 2
  IGMP query interval is 60 seconds
  IGMP querier timeout is 125 seconds
  IGMP max query response time is 10 seconds
  Last member query response interval is 1 seconds
  IGMP activity: 4 joins, 0 leaves
  IGMP querying router is 1.1.1.1 (this system)
```

IGMPv2—Maintaining a Group



- Router sends periodic queries
- One member per group per subnet reports
- Other members suppress reports

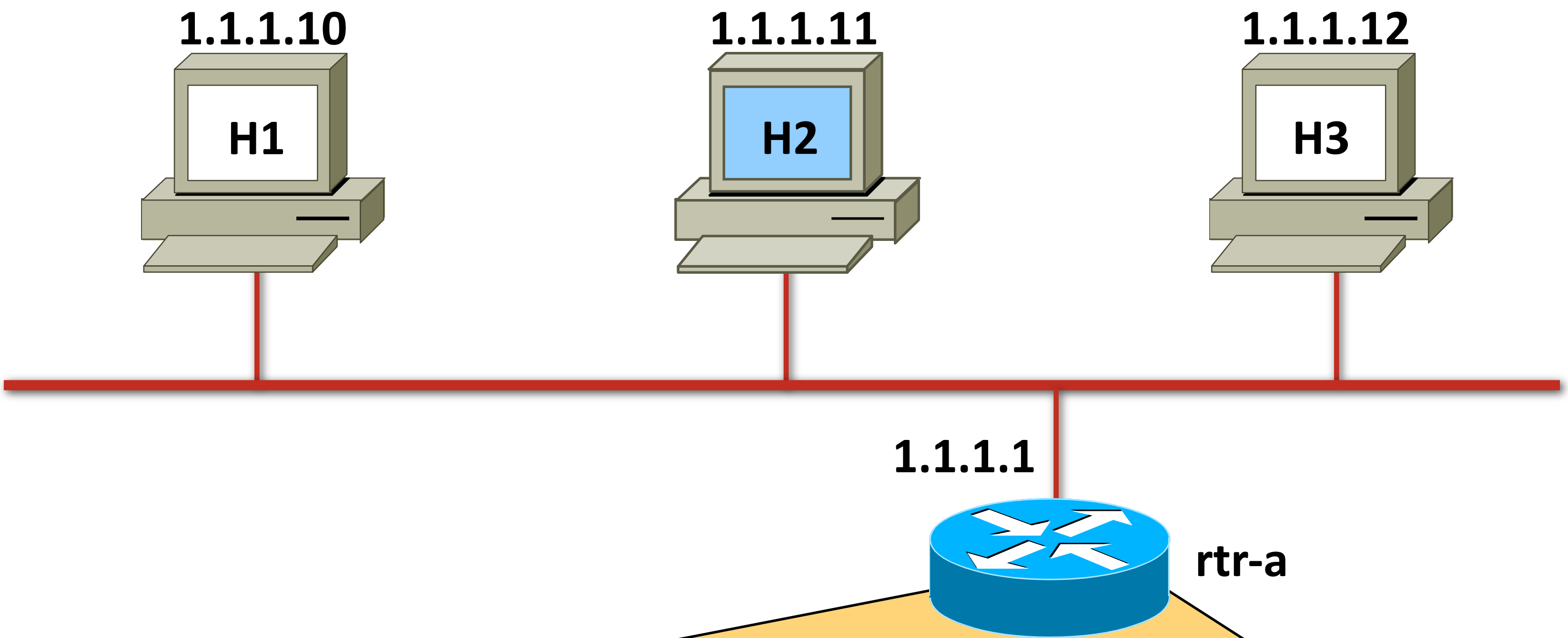
IGMPv2—Leaving a Group



IGMP State in "rtr-a" before Leave

```
rtr-a>sh ip igmp group
IGMP Connected Group Membership
Group Address     Interface       Uptime     Expires    Last Reporter
224.1.1.1         Ethernet0       6d17h      00:02:31   1.1.1.11
```

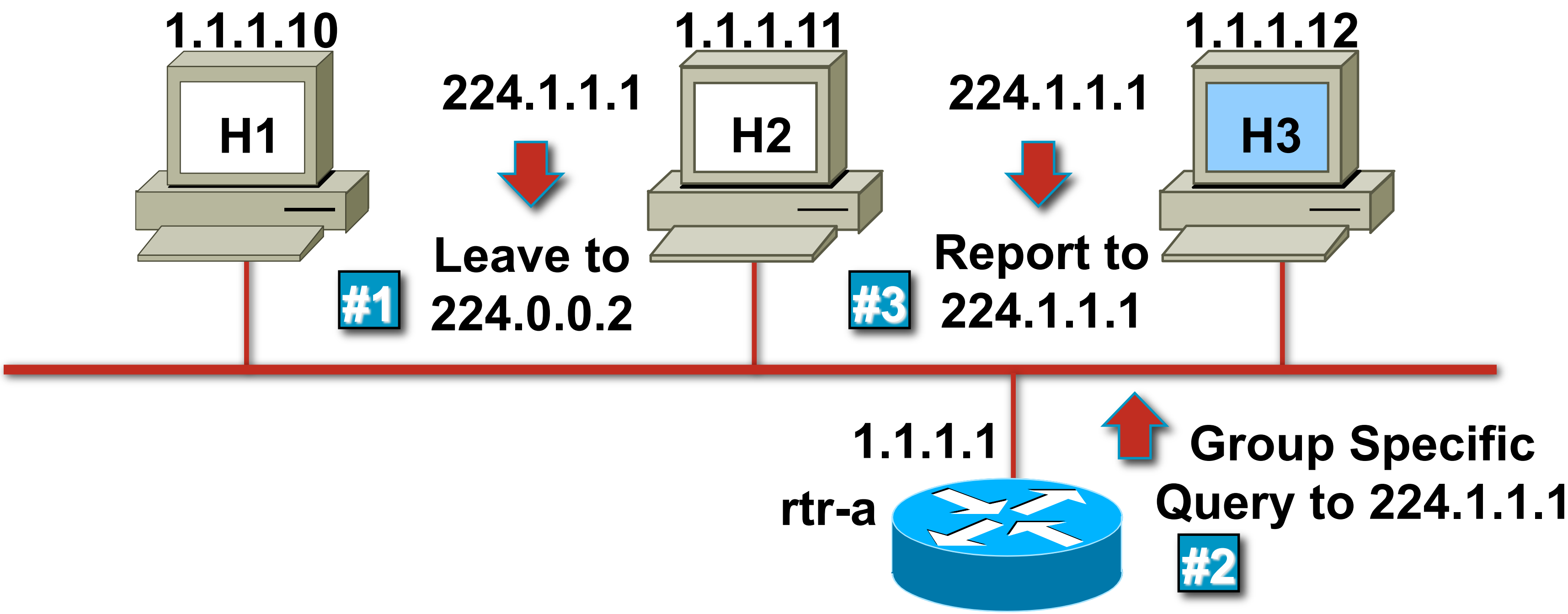
IGMPv2—Joining a Group



IGMP State in “rtr-a” before Leave (XR)

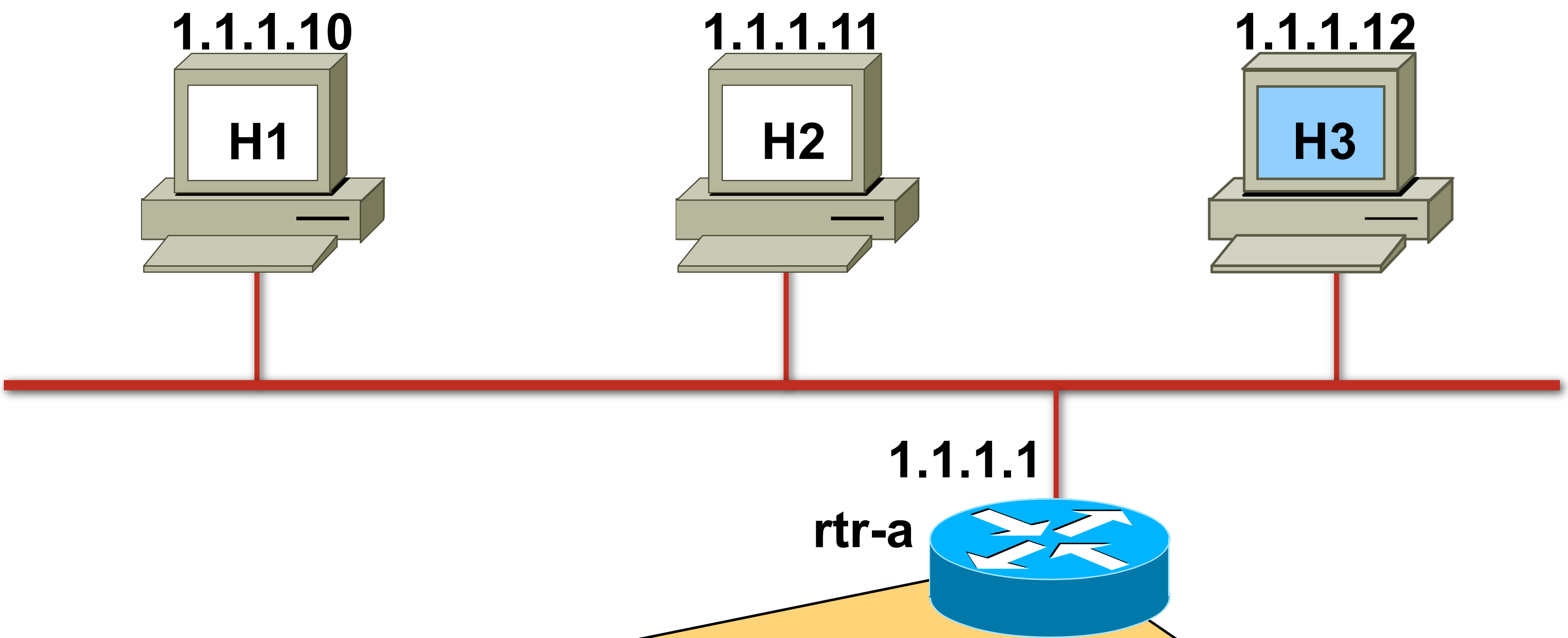
```
RP/0/RP0/CPU0:rtr-a#show igmp group
IGMP Connected Group Membership
Group Address    Interface                    Uptime    Expires   Last Reporter
224.1.1.1        Ethernet0                    00:00:35  00:01:34  1.1.1.11
```


IGMPv2—Leaving a Group



- H2 leaves group; sends Leave message
- Router sends Group specific query
- A remaining member host sends report
- Group remains active

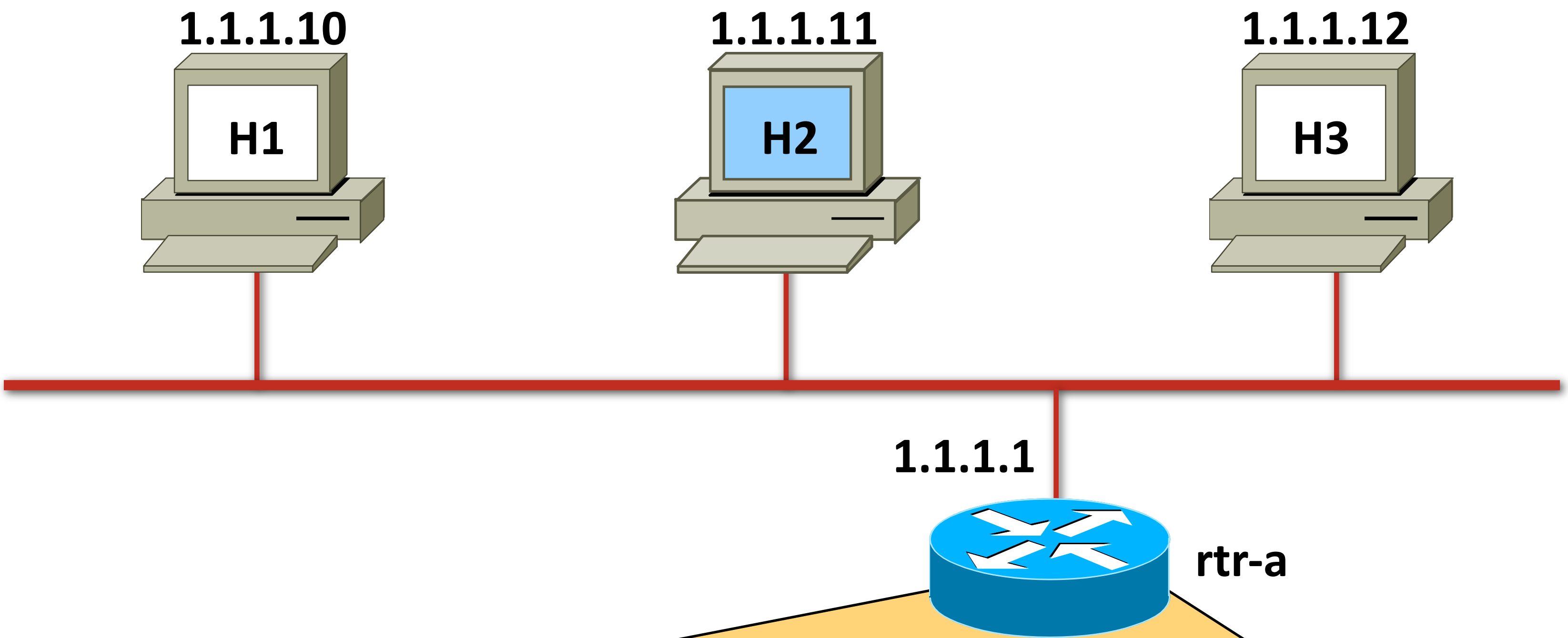
IGMPv2—Leaving a Group



IGMP State in "rtr-a" after H2 Leaves

```
rtr-a>sh ip igmp group
IGMP Connected Group Membership
Group Address      Interface      Uptime    Expires    Last Reporter
224.1.1.1          Ethernet0      6d17h     00:01:47   1.1.1.12
```

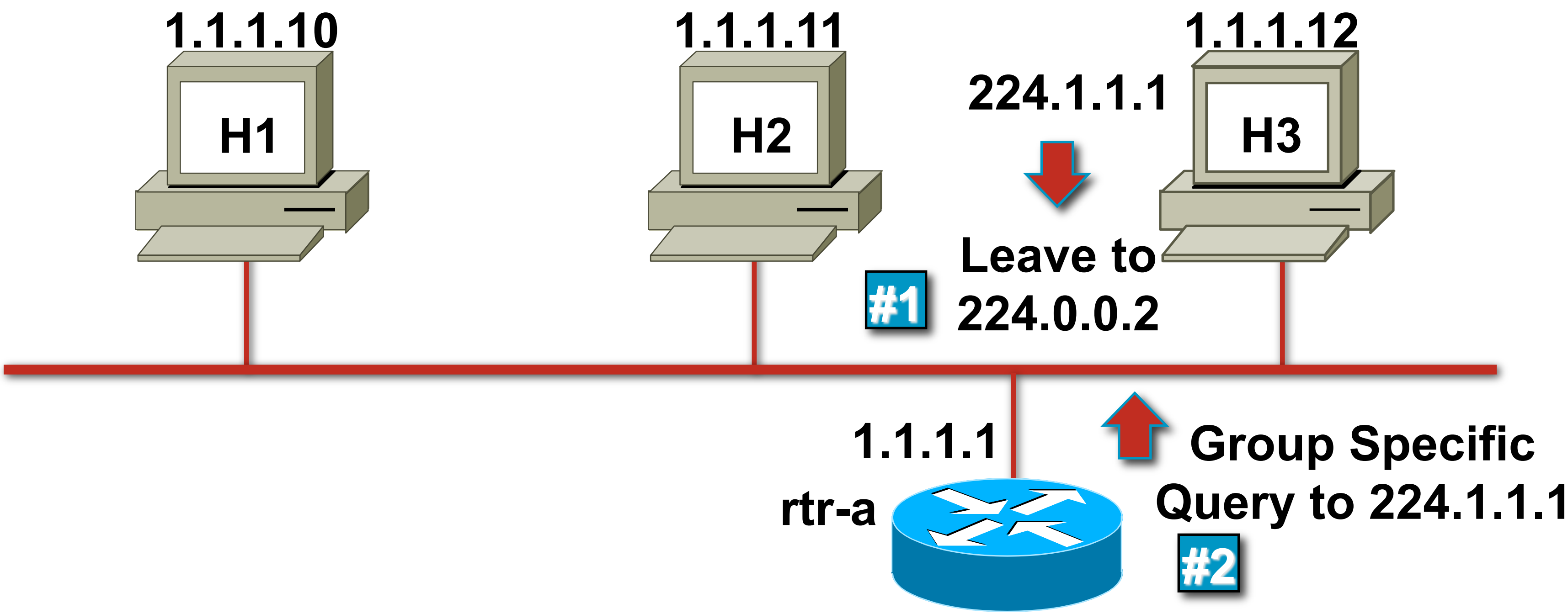

IGMPv2—Joining a Group



IGMP State in "rtr-a" after H2 Leaves (XR)

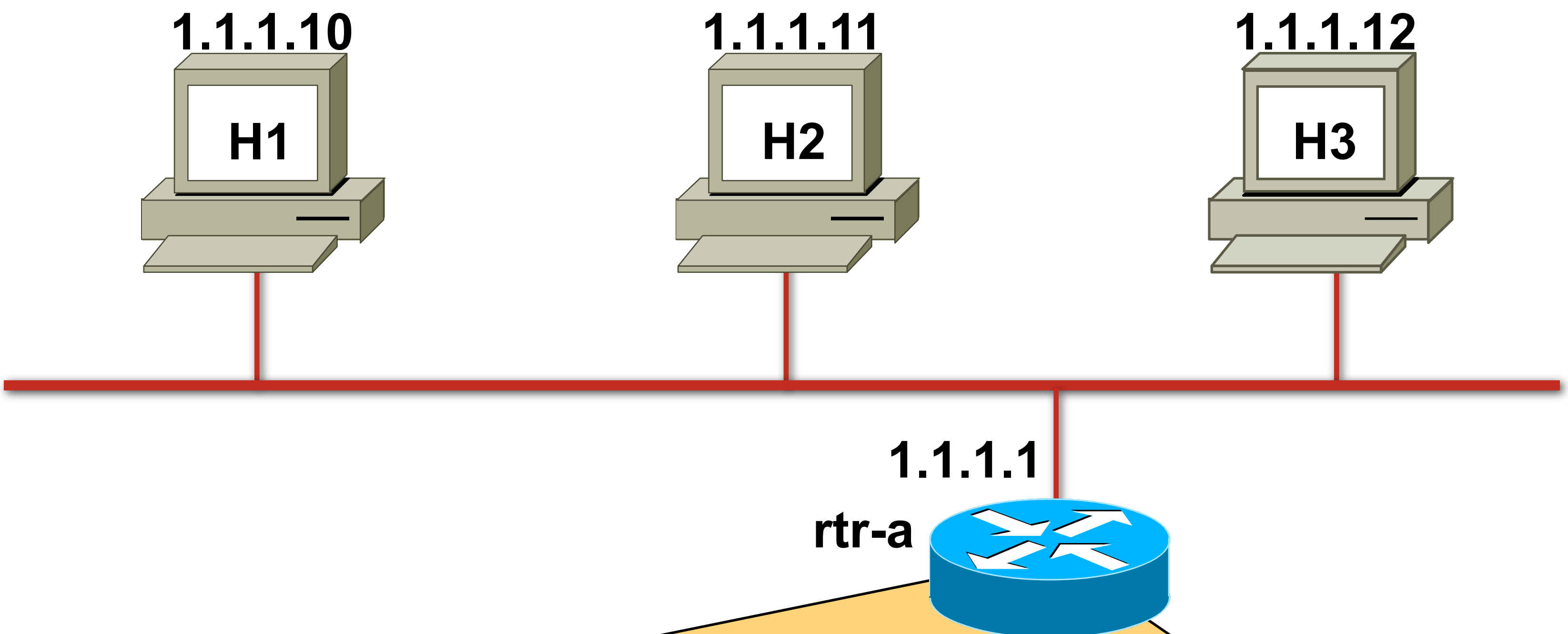
```
RP/0/RP0/CPU0:rtr-a#show igmp group
IGMP Connected Group Membership
Group Address    Interface                   Uptime    Expires   Last Reporter
224.1.1.1        Ethernet0                   00:00:53  00:02:14  1.1.1.12
```


IGMPv2—Leaving a Group



- Last host leaves group; sends Leave message
- Router sends Group specific query
- No report is received
- Group times out

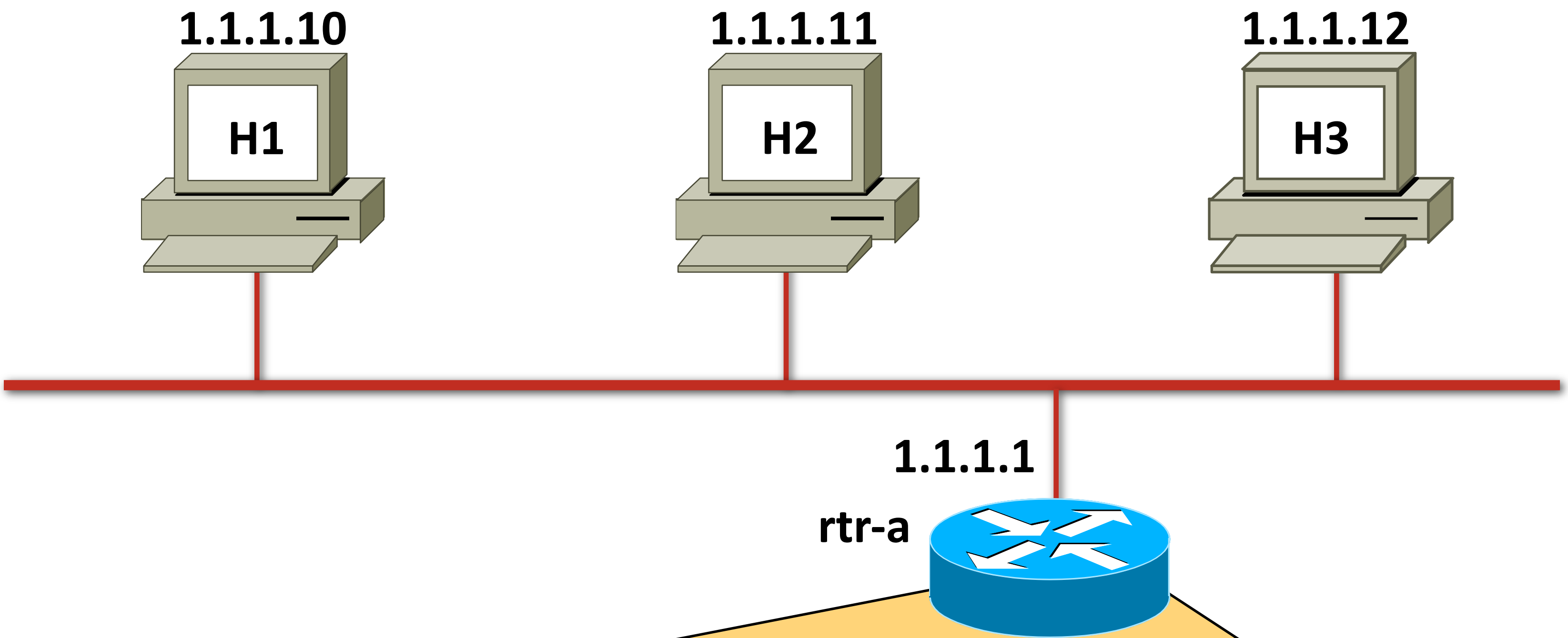
IGMPv2—Leaving a Group



IGMP State in "rtr-a" after H3 Leaves

```
rtr-a>show ip igmp group
IGMP Connected Group Membership
Group Address    Interface      Uptime    Expires    Last Reporter
```


IGMPv2—Leaving a Group



IGMP State in "rtr-a" after H3 Leaves (XR)

```
RP/0/RP0/CPU0:rtr-a#show igmp group
IGMP Connected Group Membership
Group Address    Interface                    Uptime    Expires    Last Reporter
```

IGMPv3

IGMPv3

RFC 3376

- Adds Include/Exclude Source Lists

  Enables hosts to listen only to a specified subset of the hosts sending to the group

  Requires new 'IPMulticastListen' API

    New IGMPv3 stack required in the O/S

  Apps must be rewritten to use IGMPv3 Include/Exclude features

  Available in IOS 12.2, 12.1(3)T and 12.0(15)S

IGMPv3

RFC 3376

- New membership report address
  - 224.0.0.22 (All-IGMPv3-Routers)
    - All IGMPv3 hosts send reports to this address
      - Instead of the target group address as in IGMPv1/v2
    - All IGMPv3 routers listen to this address
    - Hosts do not listen or respond to this address
  - No report suppression
    - All hosts on wire respond to queries
    - Response Interval may be tuned over broad range
      - Useful when large numbers of hosts reside on subnet

IGMPv3—Query Packet Format

Type = 0x11
  IGMP Query

Max. Resp. Time
  Max. time to send a response
   if < 128, Time in 1/10 secs
   if > 128, FP value (12.8 - 3174.4 secs)

Group Address:
  Multicast Group Address
  (0.0.0.0 for General Queries)

S Flag
  Suppresses processing by routers

QRV (Querier Robustness Value)
  Affects timers and # of retries

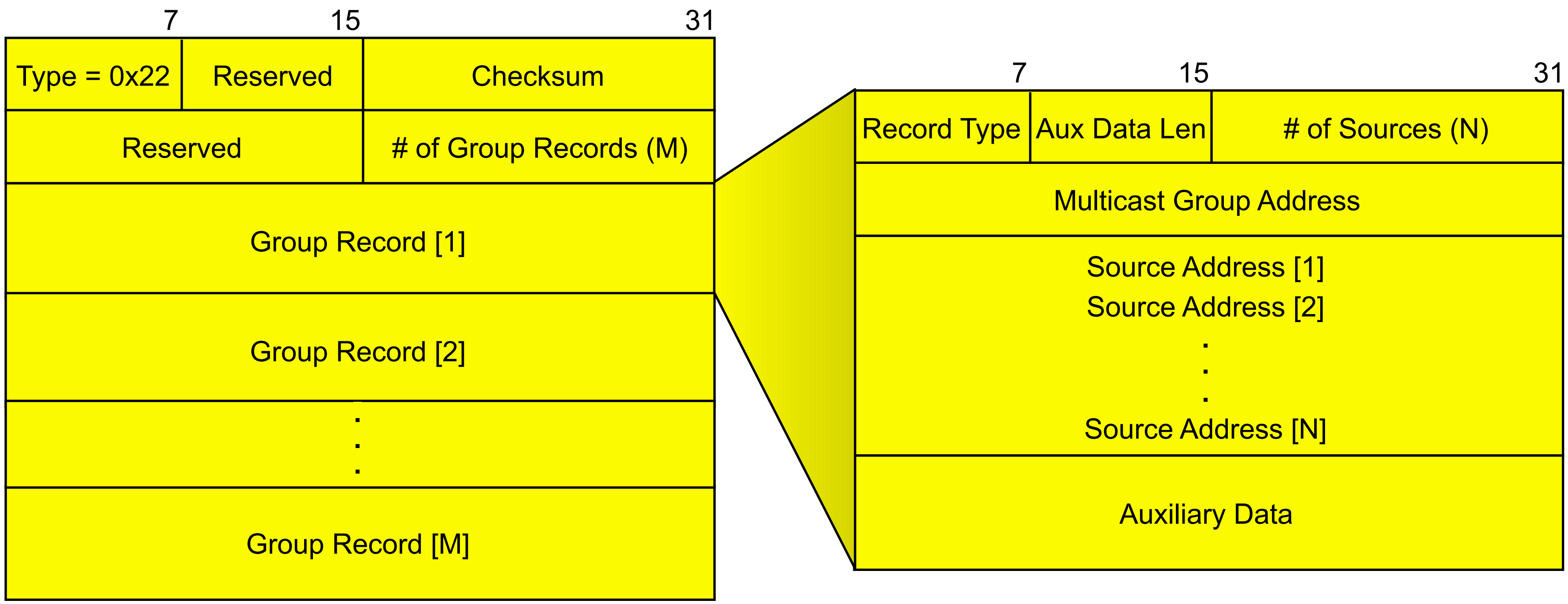
QQIC (Querier's Query Interval)
  Same format as Max. Resp. Time

Number of Sources (N)
  (Non-zero for Group-and-Source Query)

Source Address
  Address of Source

		7	15	31
Type = 0x11			Max. Resp. Code	Checksum
Group Address				
	S	QRV	QQIC	Number of Sources (N)
Source Address [1]				
Source Address [2]				
. . .				
Source Address [N]				

IGMPv3—Report Packet Format



# of Group Records (M)
  Number of Group Records in Report

Group Records 1 - M
  Group address plus list of zero or more sources to Include/Exclude (See Group Record format)

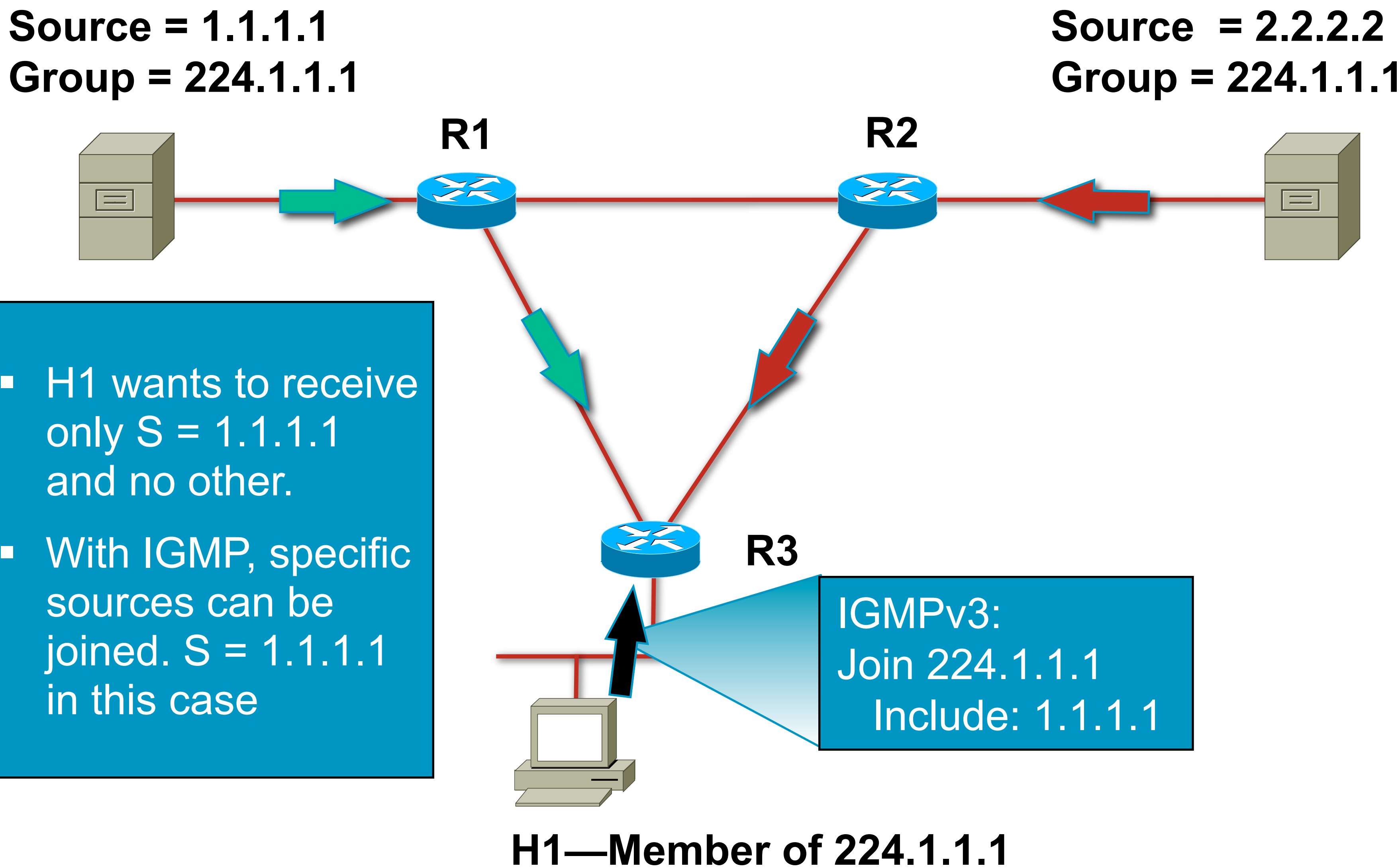
Record Type
  Include, Exclude, Chg-to-Include, Chg-to-Exclude, Add, Remove

# of Sources (N)
  Number of Sources in Record

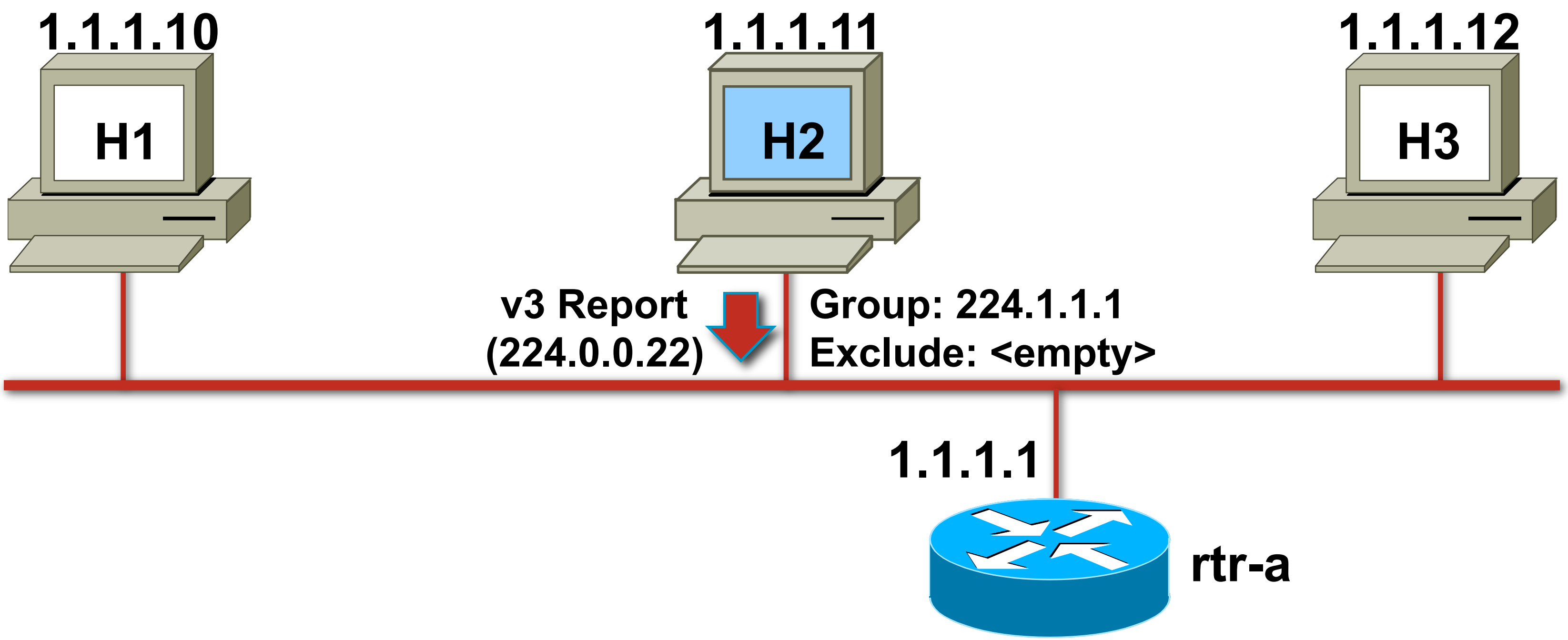
Source Address 1- N
  Address of Source

IGMPv3 Example

- H1 wants to receive only S = 1.1.1.1 and no other.
- With IGMP, specific sources can be joined. S = 1.1.1.1 in this case

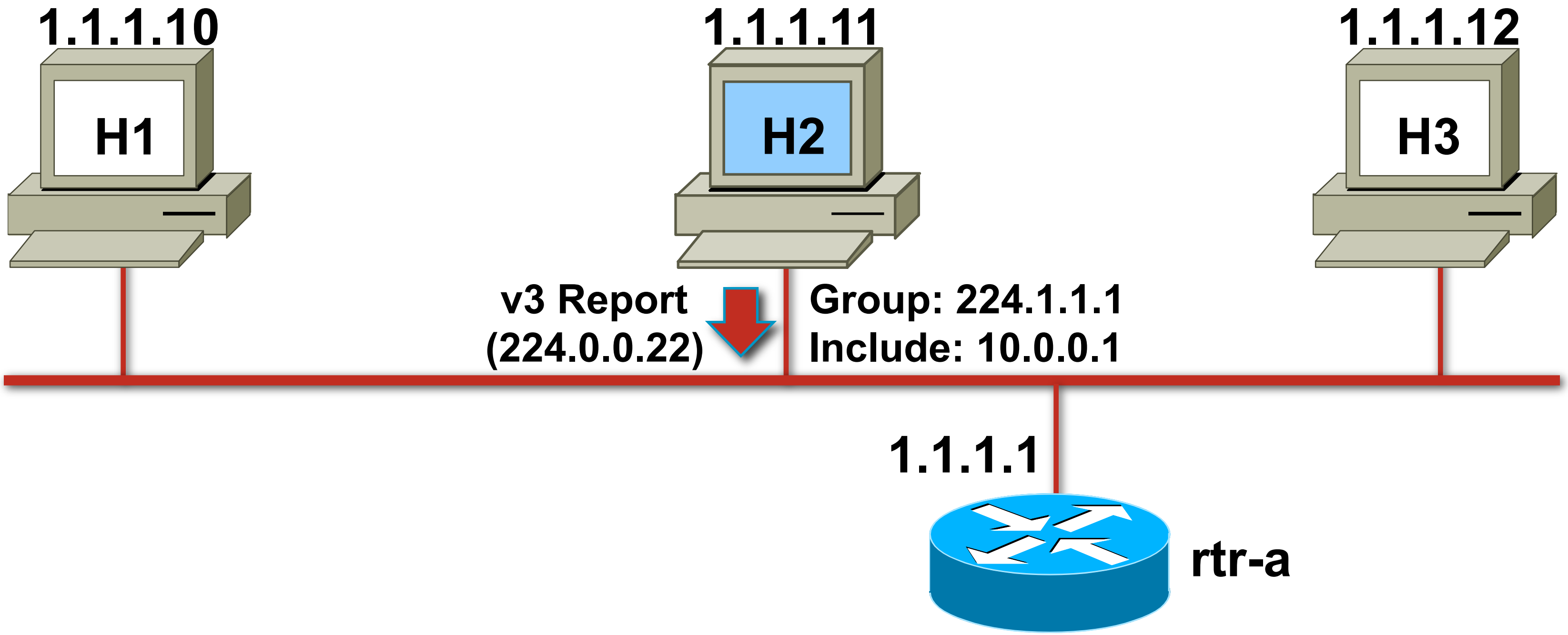


IGMPv3—Joining a Group



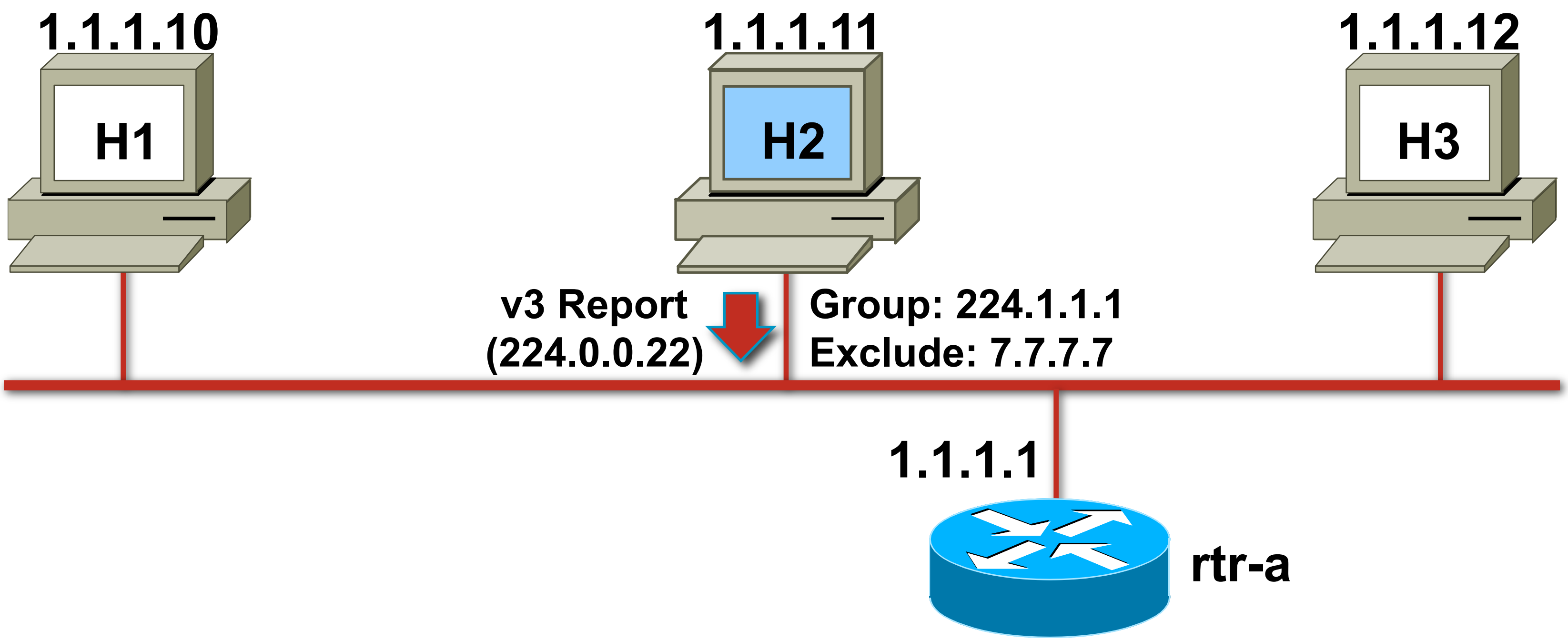
- Joining member sends IGMPv3 Report to 224.0.0.22 immediately upon joining

IGMPv3—Joining Specific Source(s)



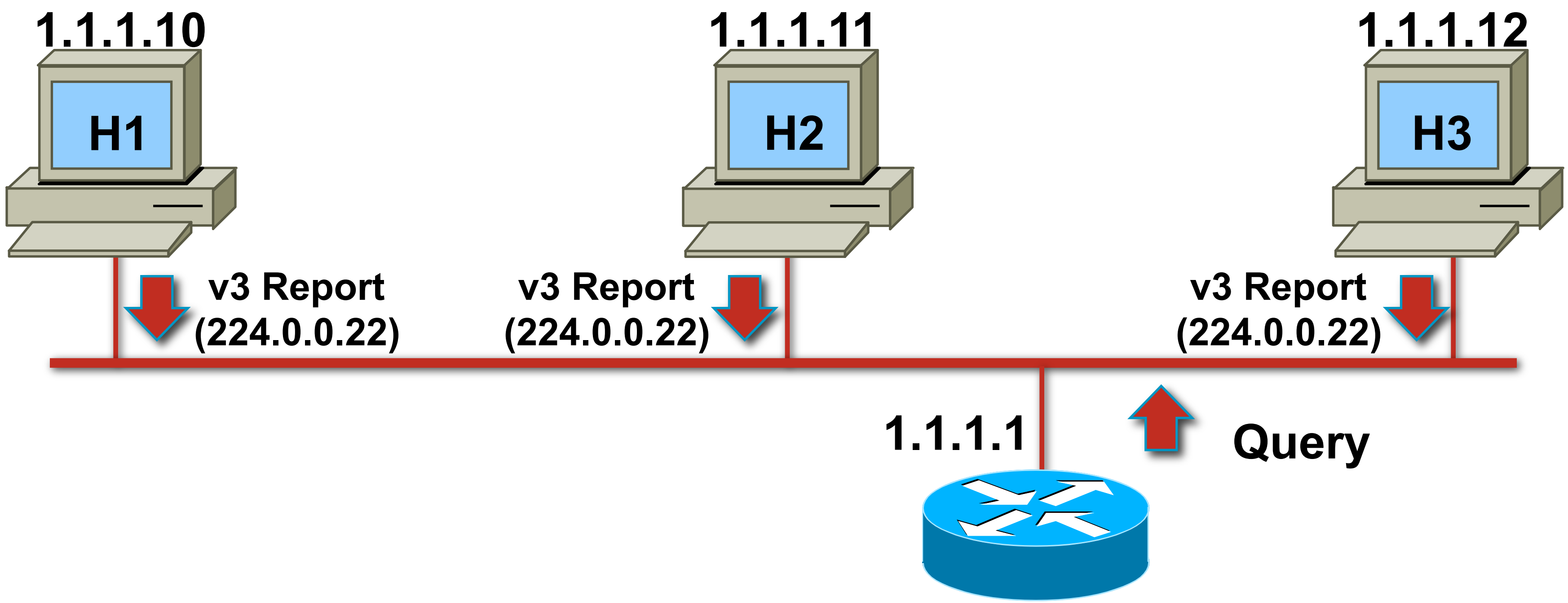
- IGMPv3 report contains desired source(s) in the Include list
- Only "Included" source(s) are joined

IGMPv3—Excluding Specific Source(s)



- IGMPv3 report contains undesired source(s) in the Exclude list
- All sources except “Excluded” source(s) are joined

IGMPv3—Maintaining State



- Router sends periodic queries
- All IGMPv3 members respond

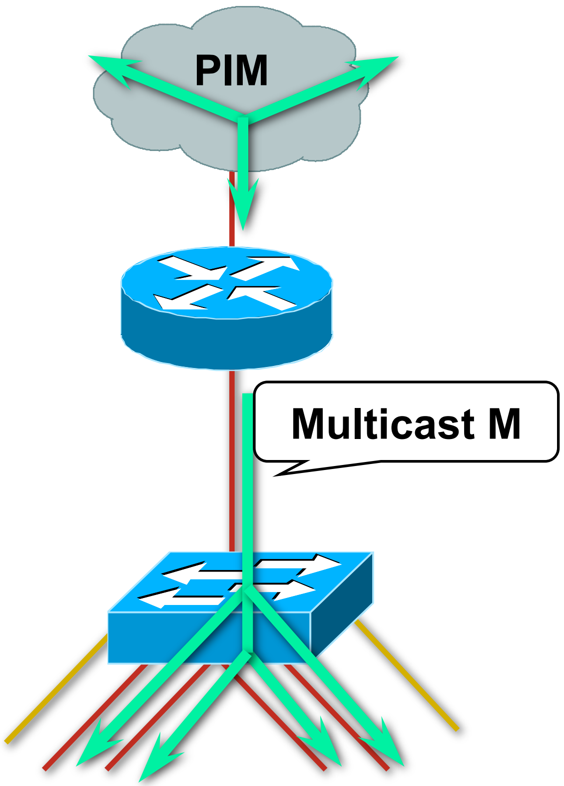
  Reports contain multiple Group state records

L2 Multicast Frame Switching

L2 Multicast Frame Switching

Problem: Layer 2 Flooding of Multicast Frames

- Typical L2 switches treat multicast traffic as unknown or broadcast and must “flood” the frame to every port
- Static entries can sometimes be set to specify which ports should receive which group(s) of multicast traffic
- Dynamic configuration of these entries would cut down on user administration

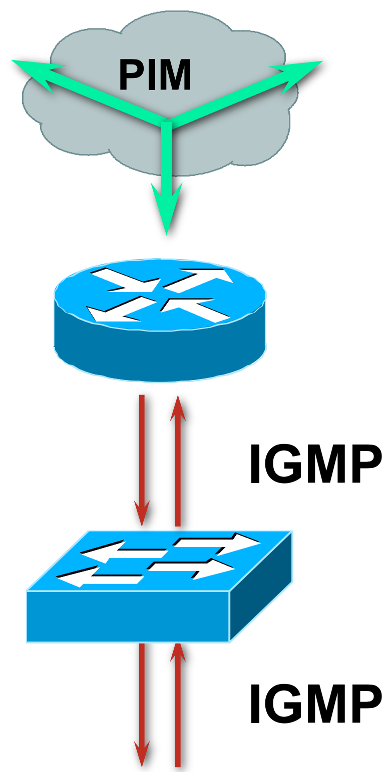


IGMP/PIM Snooping

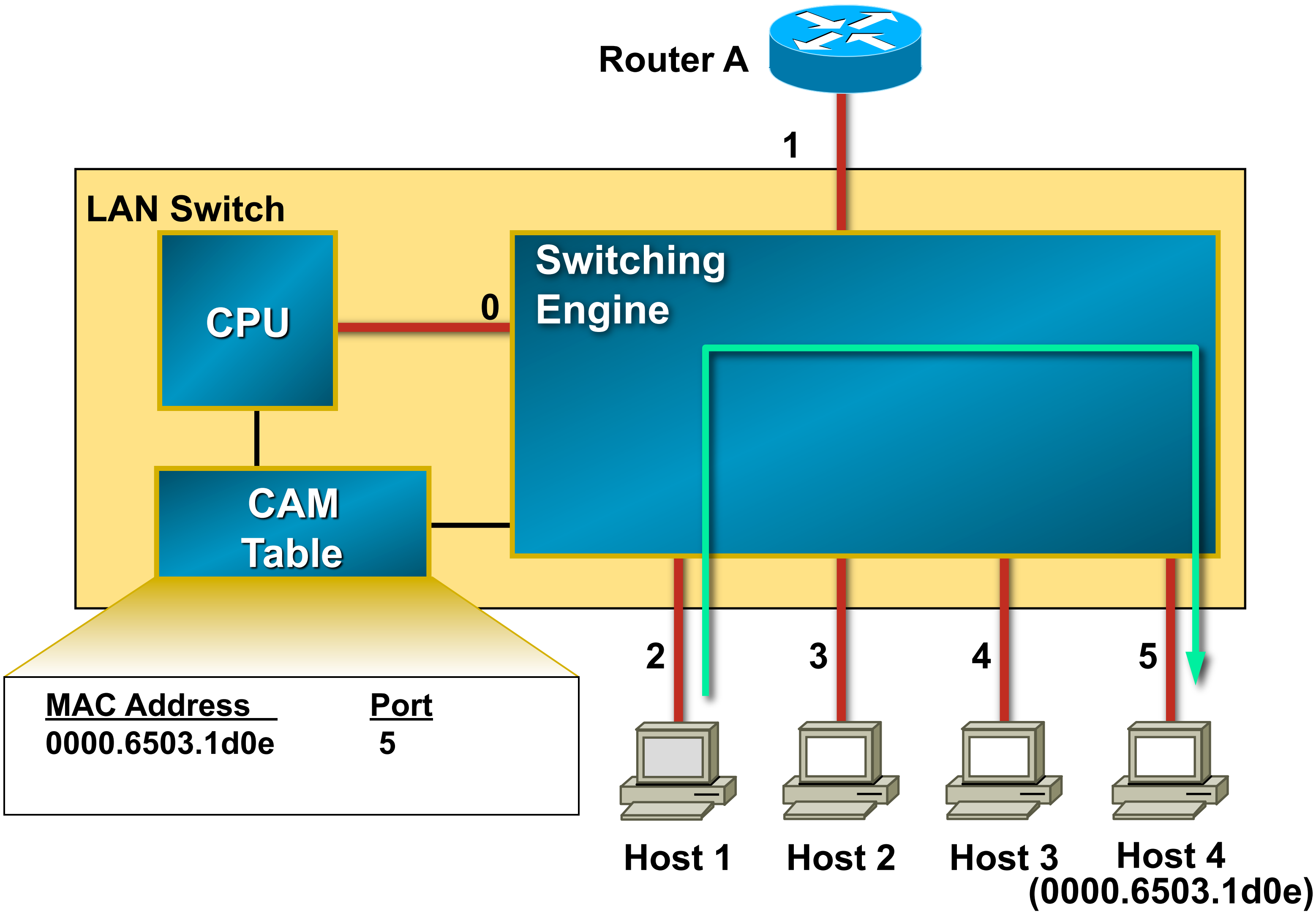
L2 Multicast Frame Switching

Solution 1: IGMPv1/2 Snooping

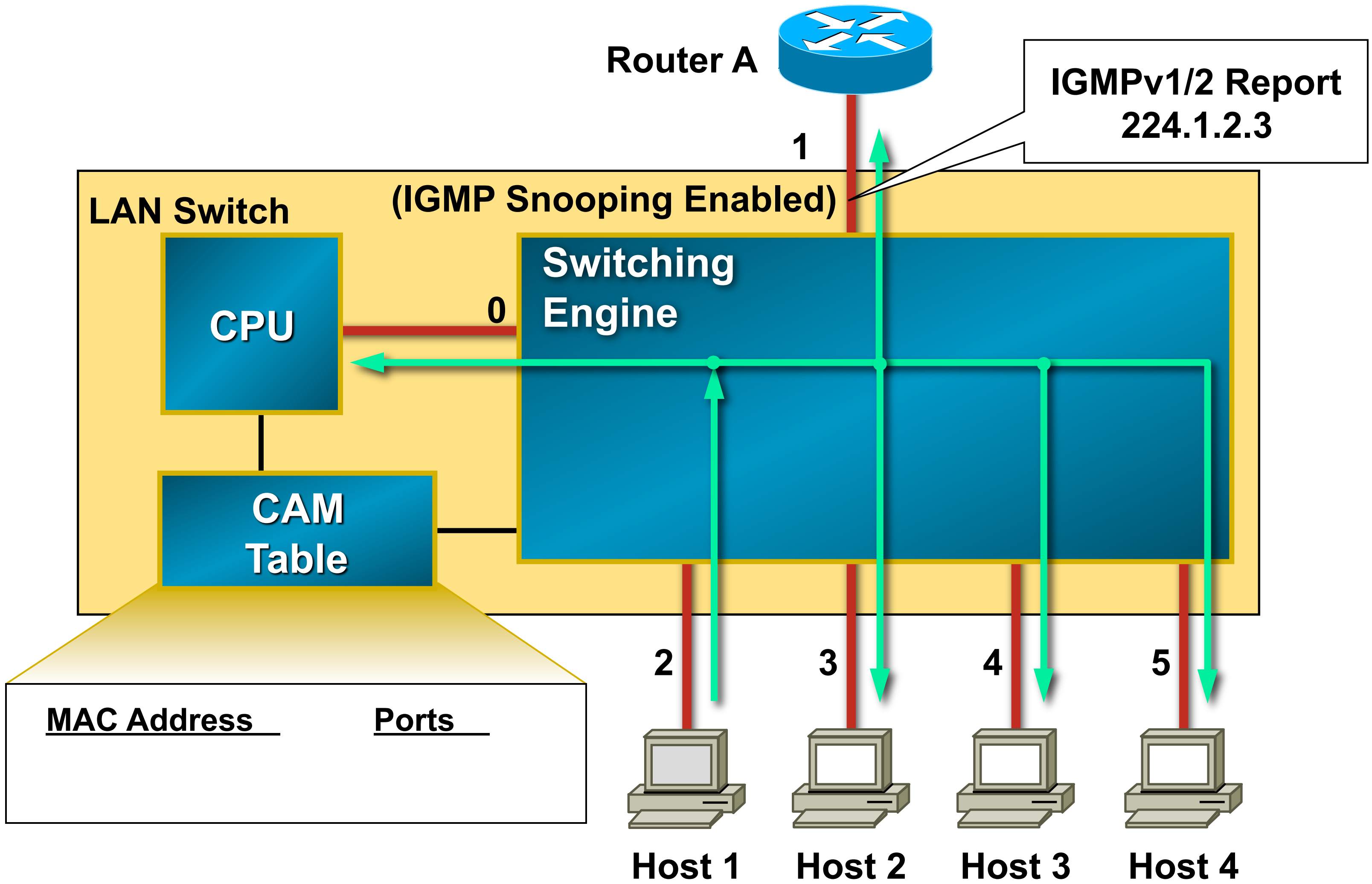
- Switches become “IGMP” aware
- IGMP packets intercepted by the NMP or by special hardware ASICs
- Switch must examine contents of IGMP messages to determine which ports want what traffic
  - IGMP membership reports
  - IGMP leave messages
- Impact on switch:
  - Must process all Layer 2 multicast packets
  - Admin. load increases with multicast traffic load
  - Requires special hardware to maintain throughput



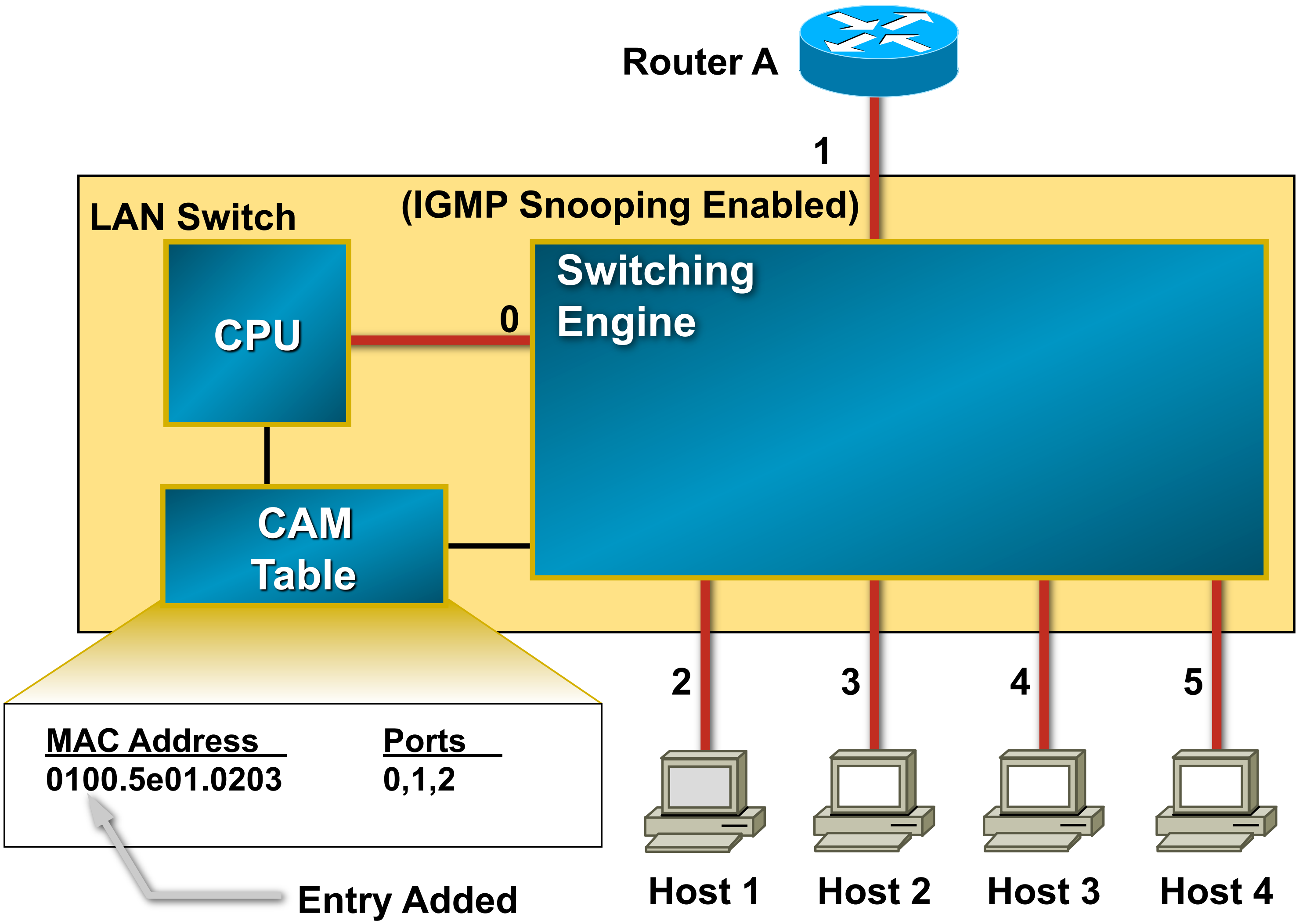
Typical L2 Switch Architecture



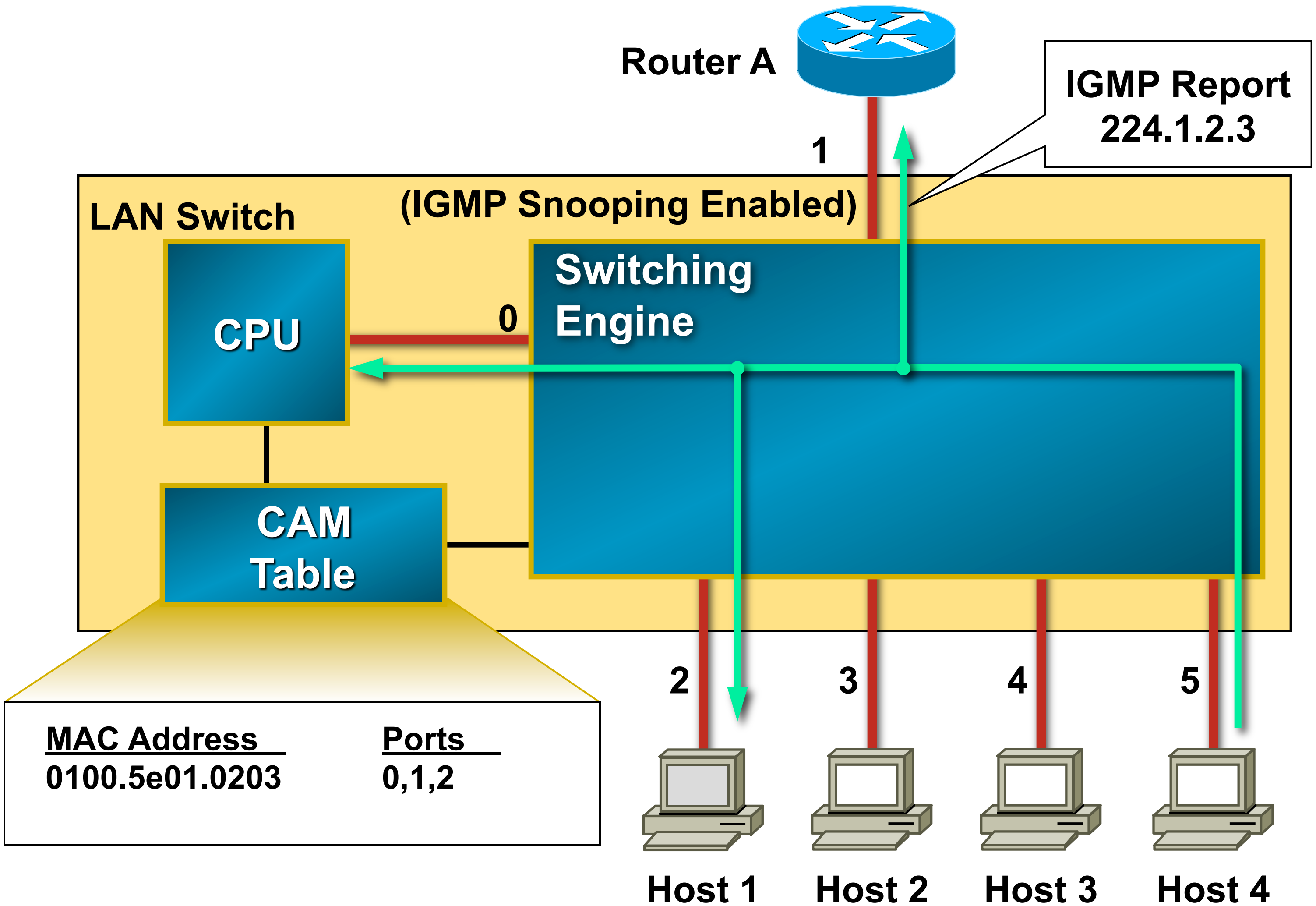
Typical L2 Switch—First Join



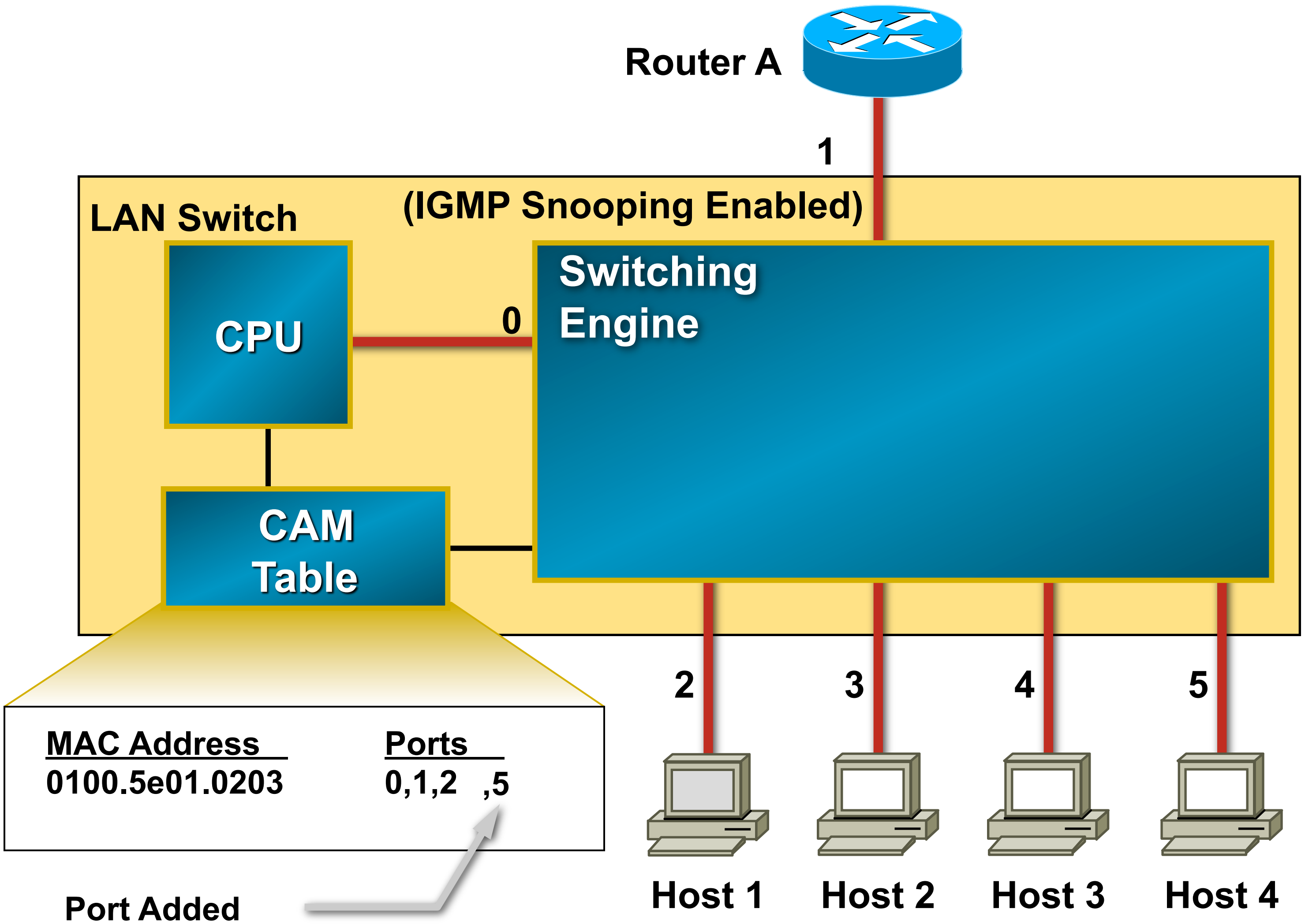
Typical L2 Switch—First Join



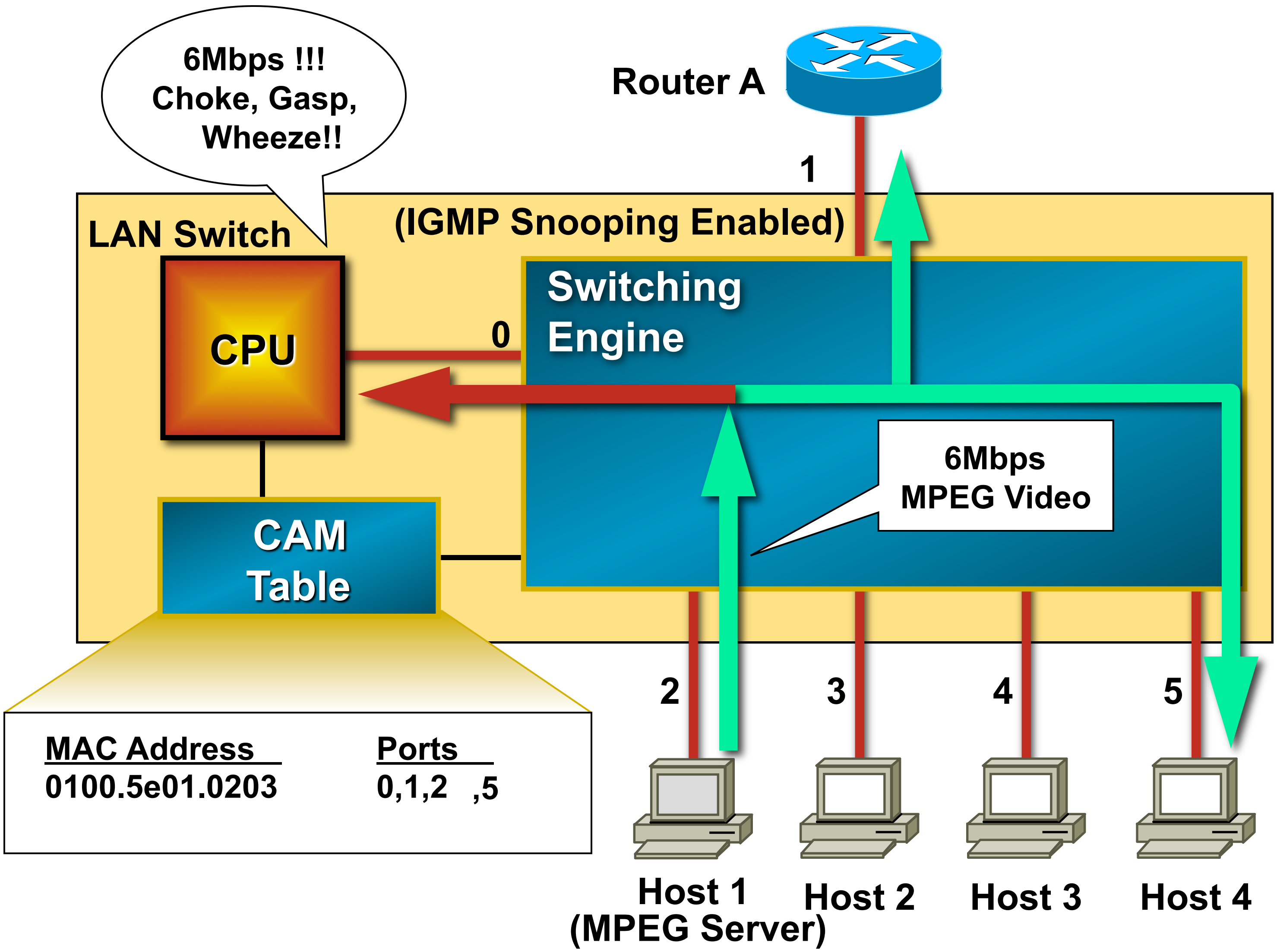
Typical L2 Switch—



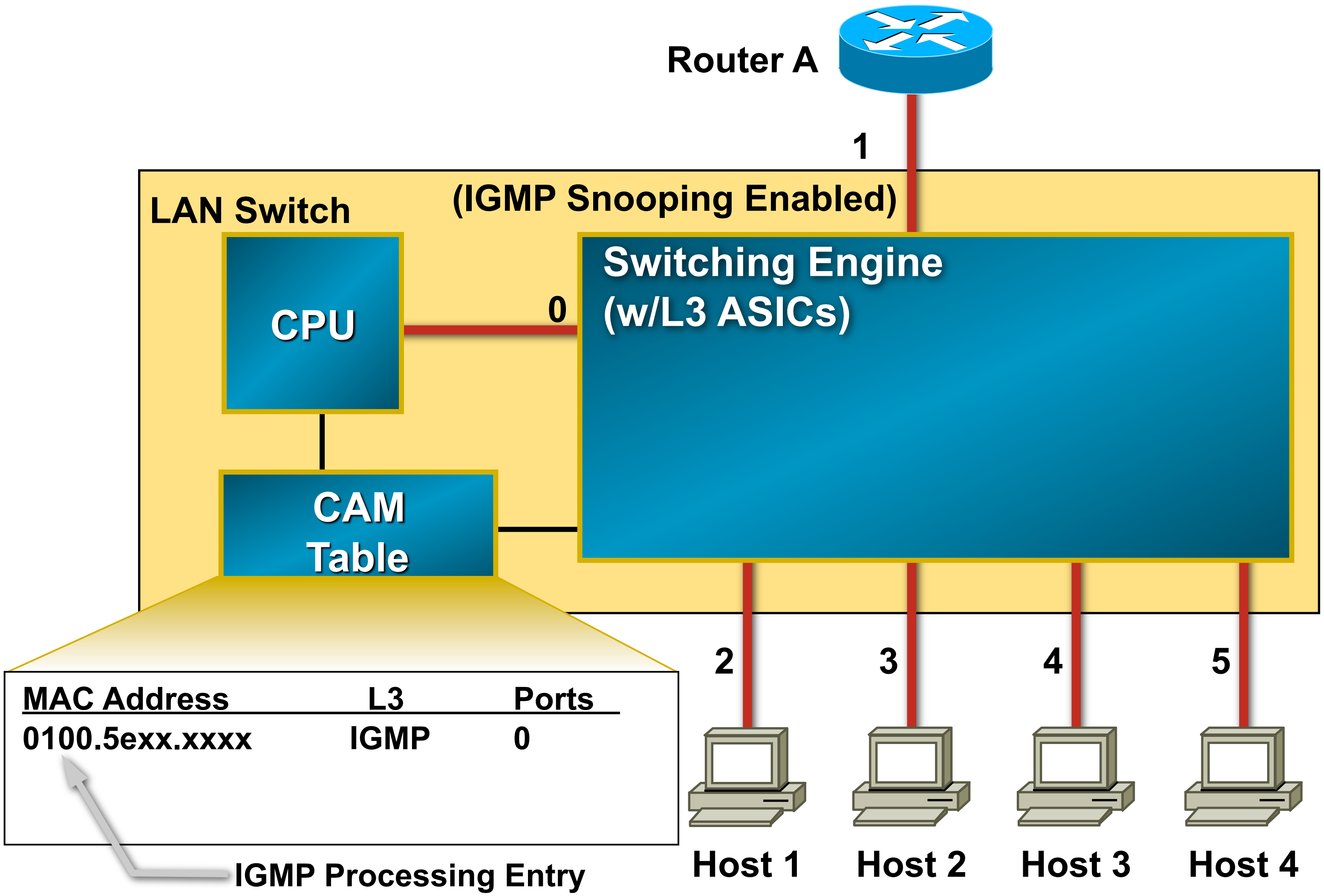
Typical L2 Switch—Second Join



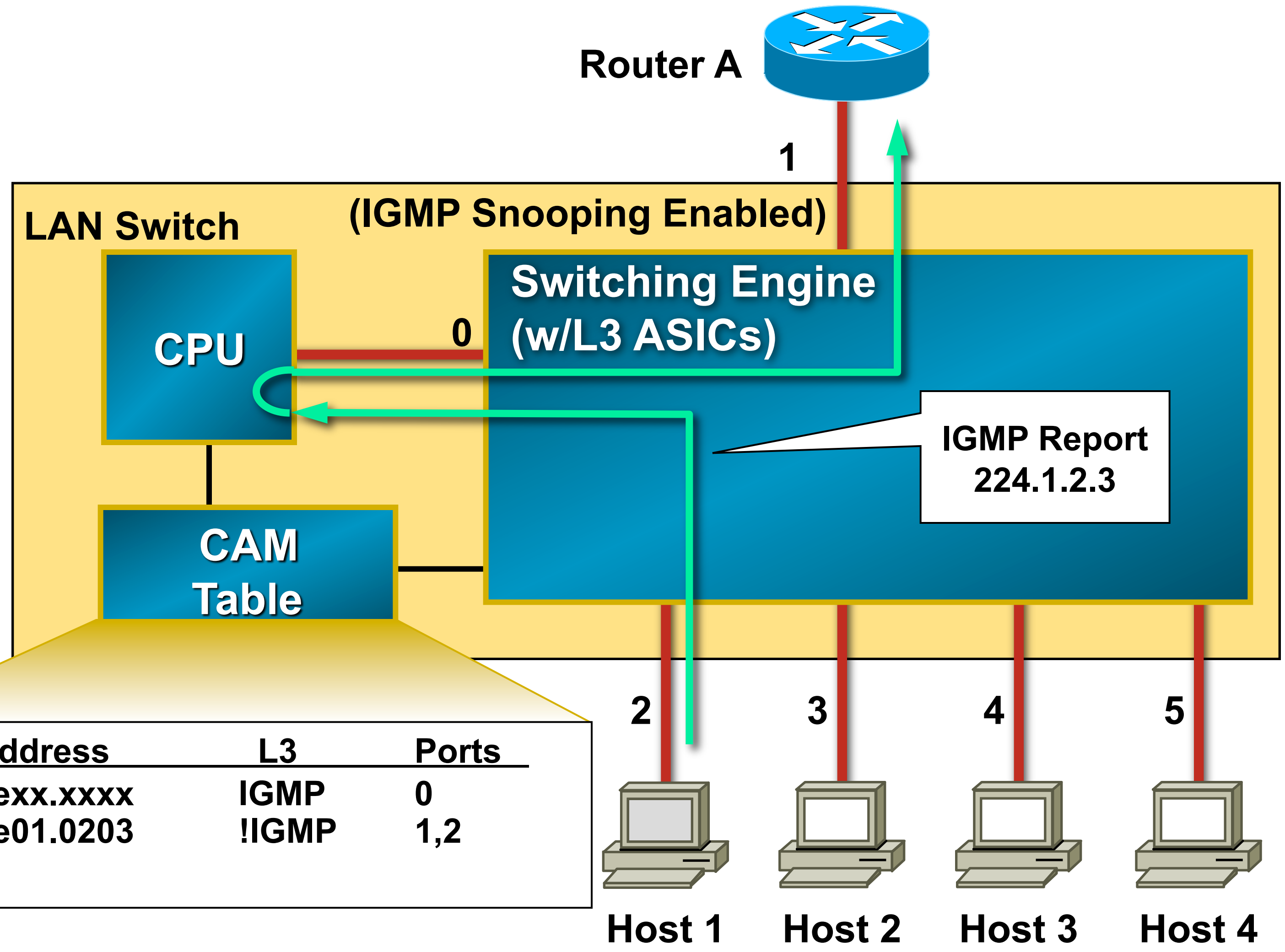
Typical L2 Switch—Meltdown!



L3 Aware Switch

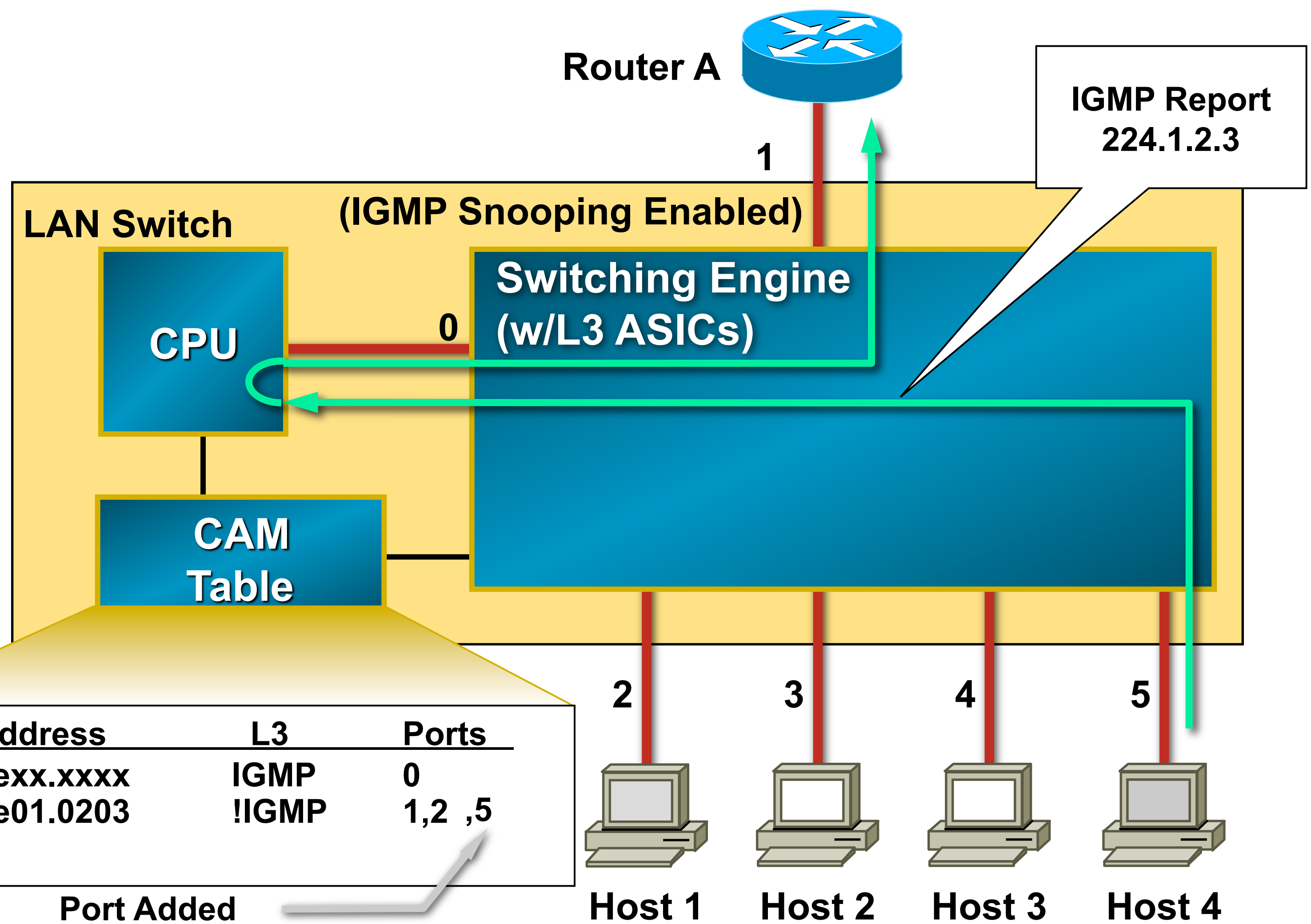


L3 Aware Switch



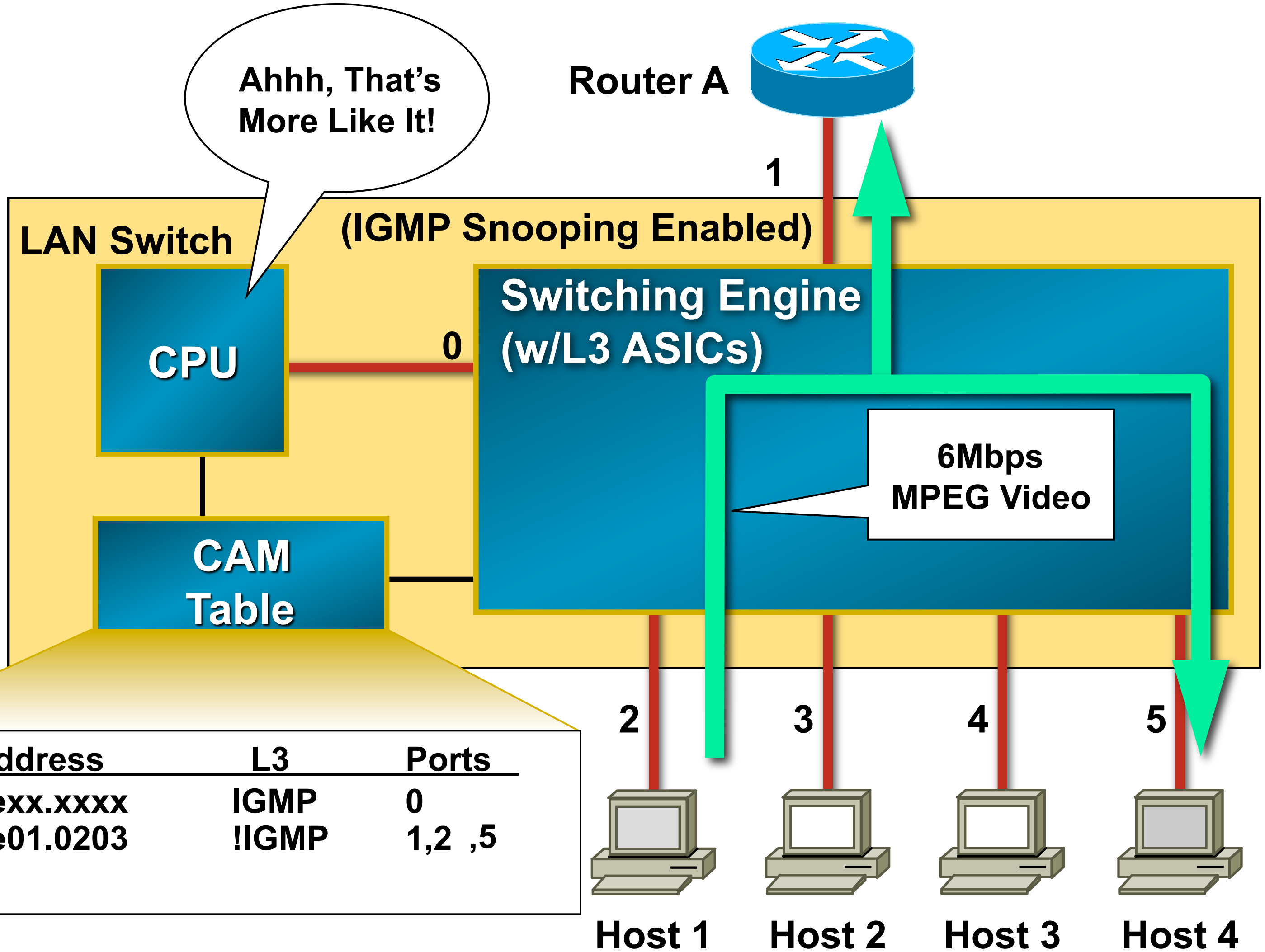
MAC Address	L3	Ports
0100.5exx.xxxx	IGMP	0
0100.5e01.0203	!IGMP	1,2

L3 Aware Switch



MAC Address	L3	Ports
0100.5exx.xxxx	IGMP	0
0100.5e01.0203	!IGMP	1,2 ,5

L3 Aware Switch

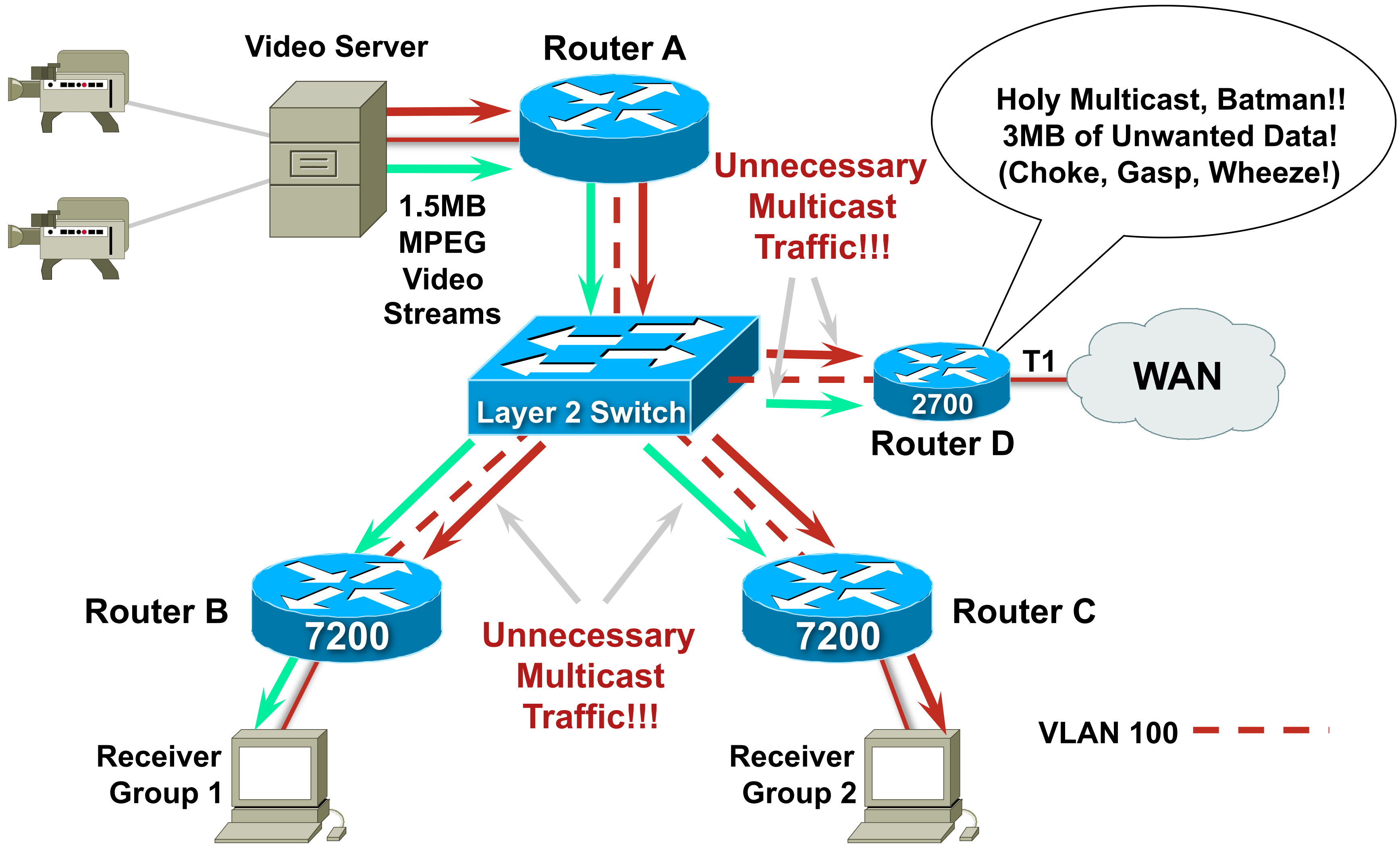


MAC Address	L3	Ports
0100.5exx.xxxx	IGMP	0
0100.5e01.0203	!IGMP	1,2 ,5

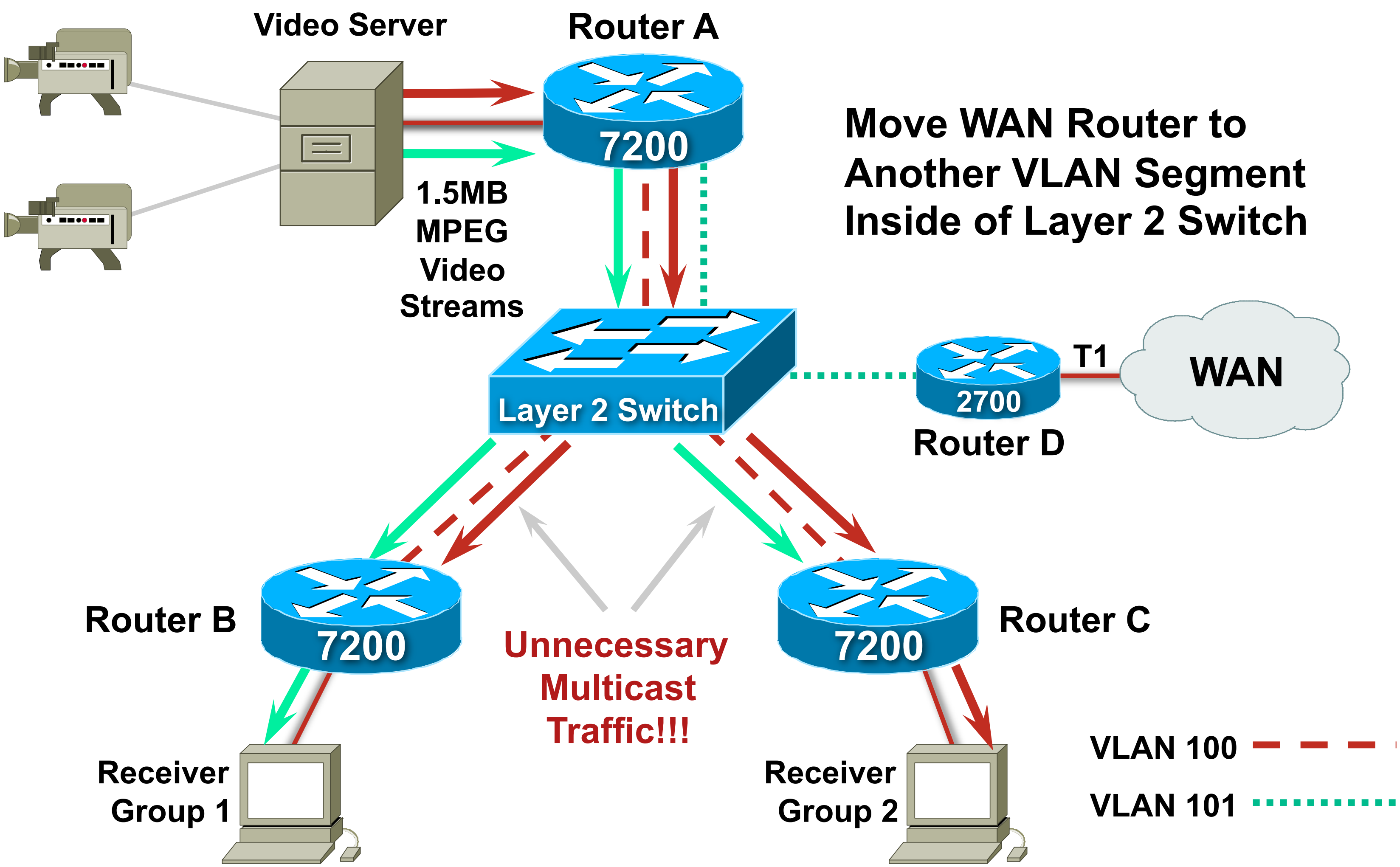
Summary—Frame Switches

- IGMP snooping
  - Switches with Layer 3-aware ASICs
    - High-throughput performance maintained
    - Increases cost of switches
  - Switches without Layer 3-aware ASICs
    - Suffer serious performance degradation
    - Will not be an issue for IGMPv3

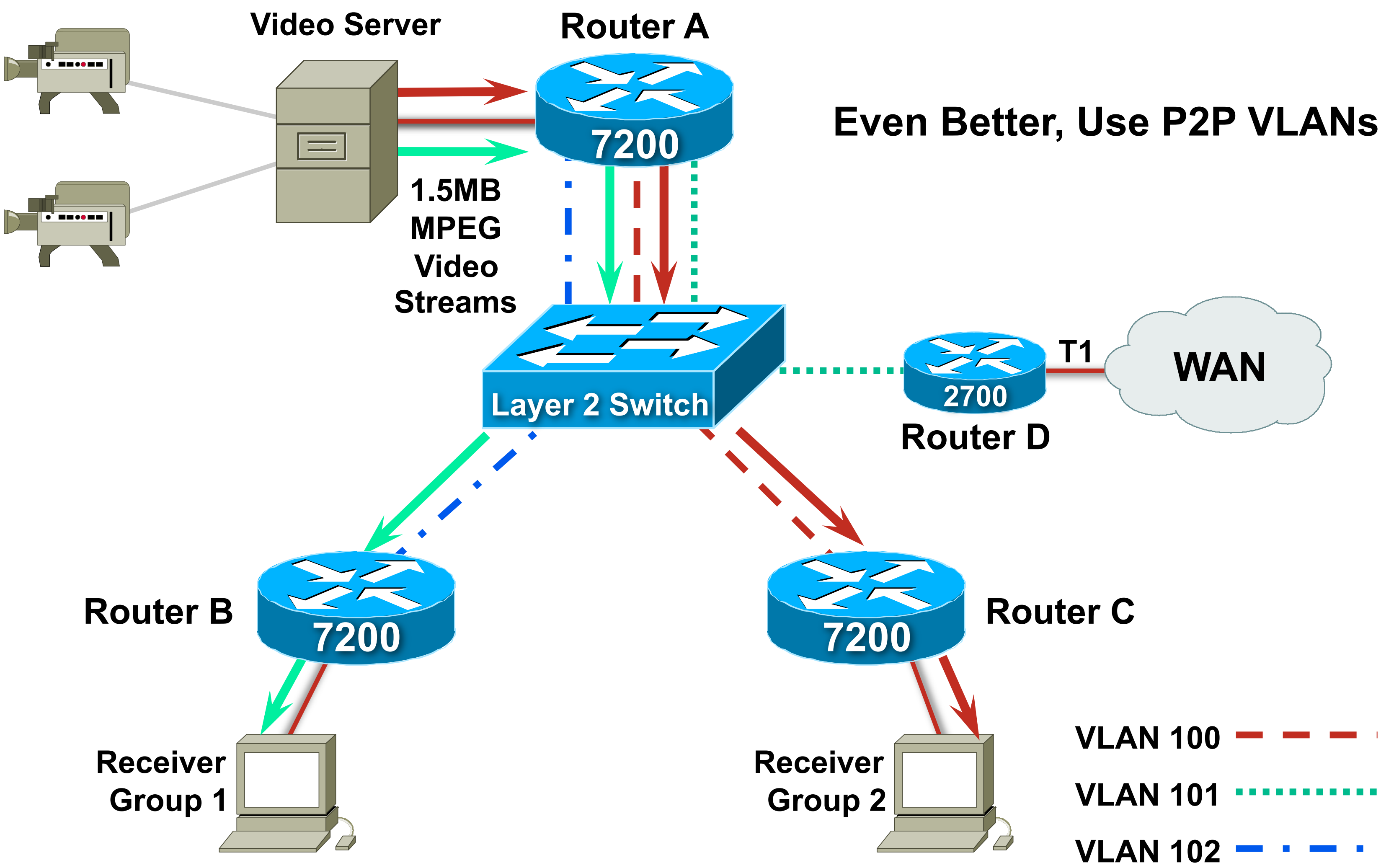
Design Issue—Core Switch



Design Issue—Core Switch



Design Issue—Core Switch



Design Issue—Core Switch

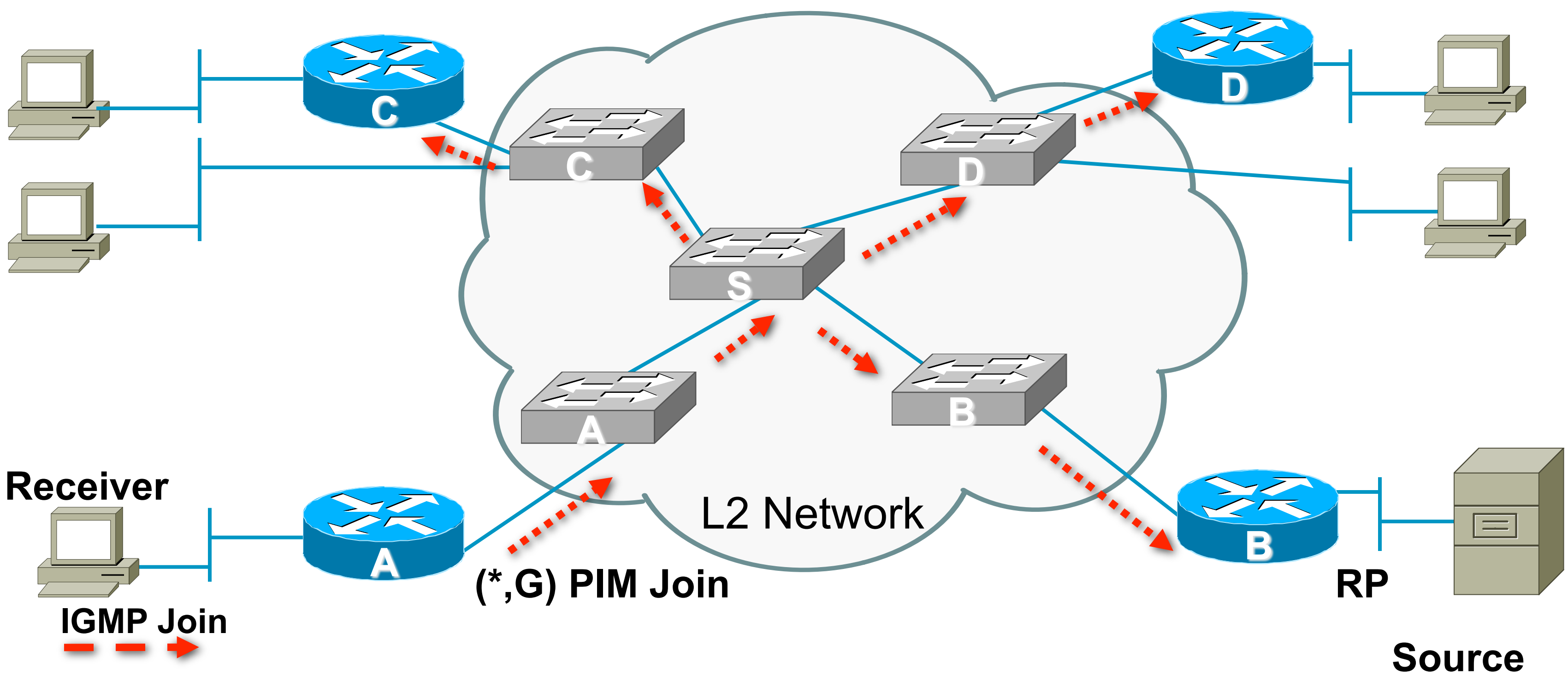
- Problem
  - Routers send PIM Join/Prunes at Layer 3
    - IGMP Join/Leaves not sent by routers
    - Other routers on VLAN can override Prune
  - Switches operate at Layer 2
    - Use IGMP Snooping to constrain multicast
    - Must assume routers want all multicast traffic
  - Need new Layer 2 Join/Prune mechanism
- Solution: PIM Snooping

PIM Snooping

- Constrain multicast traffic among multicast router ports of a VLAN
  - IGMP Snooping constrains on host ports
  - IGMP Snooping floods on multicast router ports
  - Effective in core, IXP, Metro-Ethernet

Without PIM Snooping

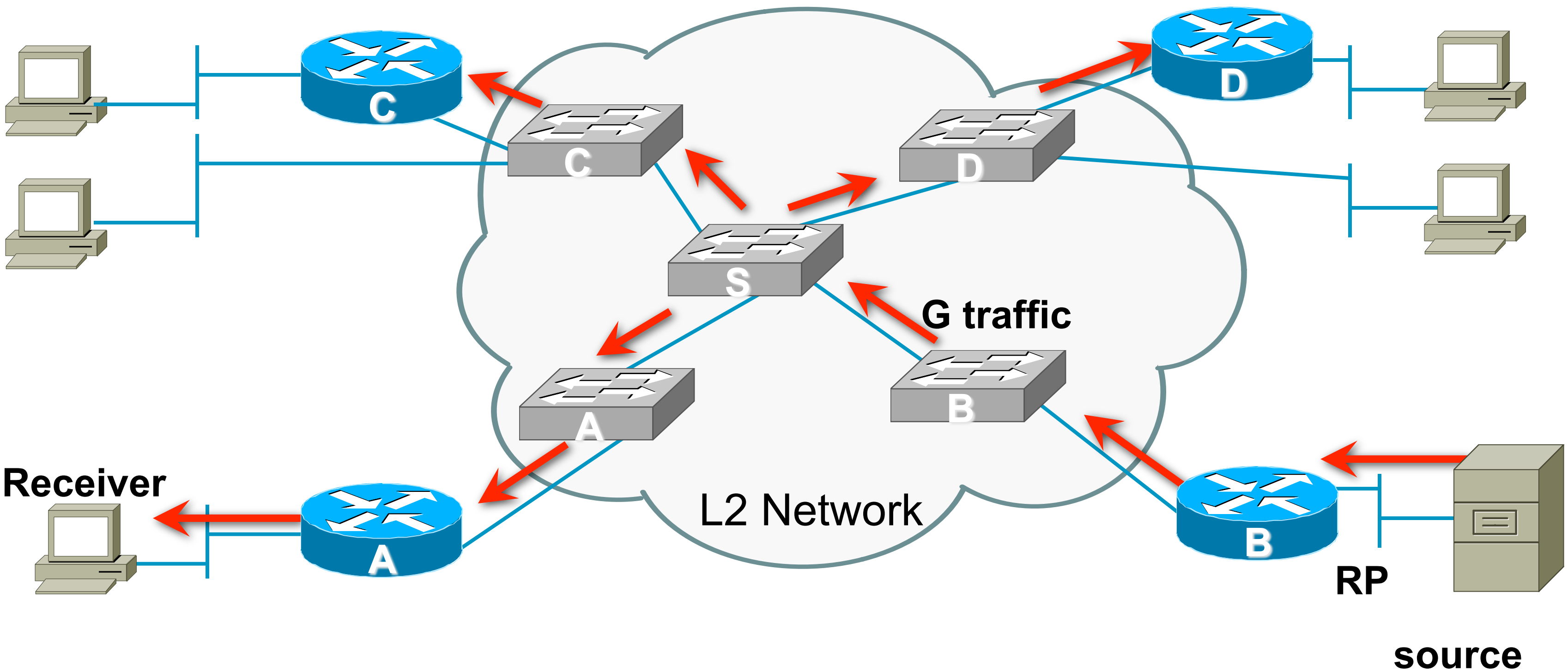
PIM Join Flow



PIM Joins Are Flooded in the vlan

Without PIM Snooping

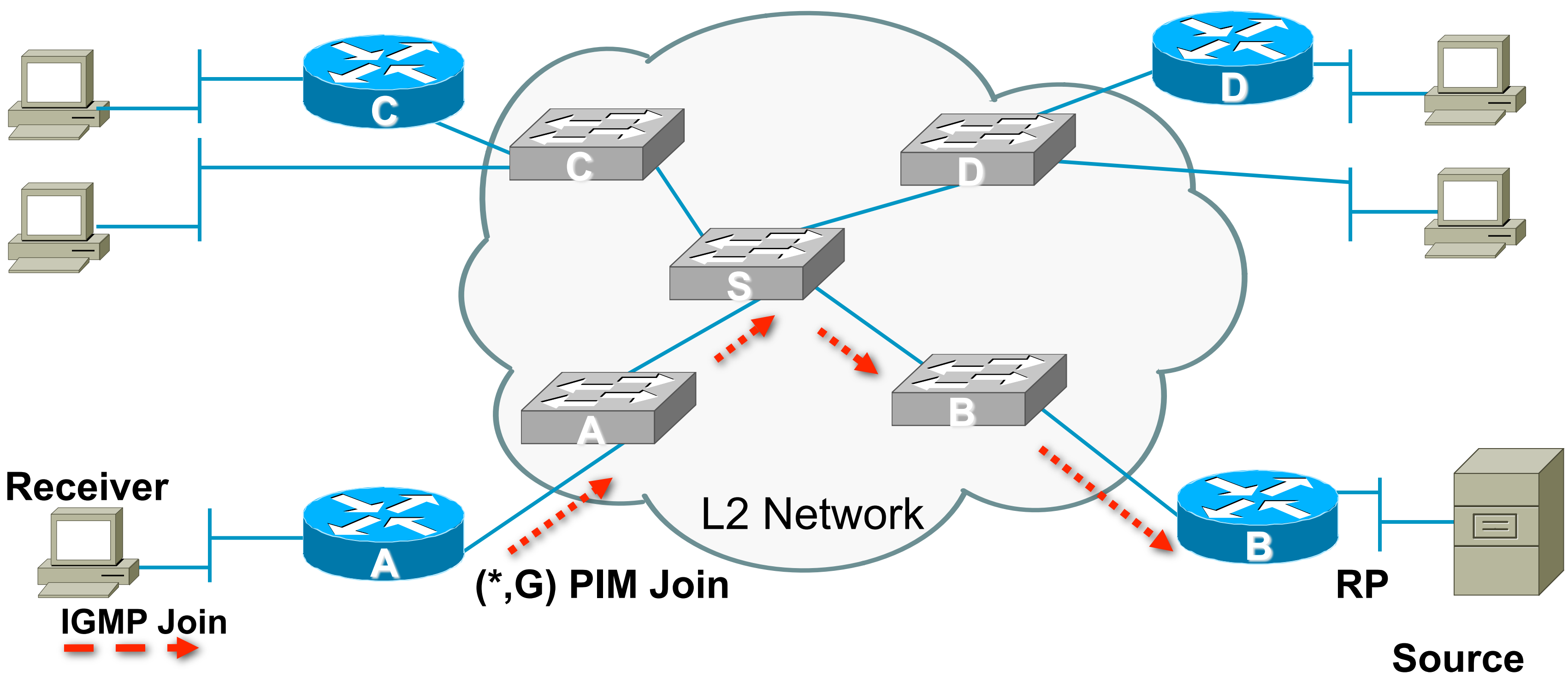
Traffic Flow



Traffic Is Unnecessarily Flooded to Routers C and D Also

With PIM Snooping

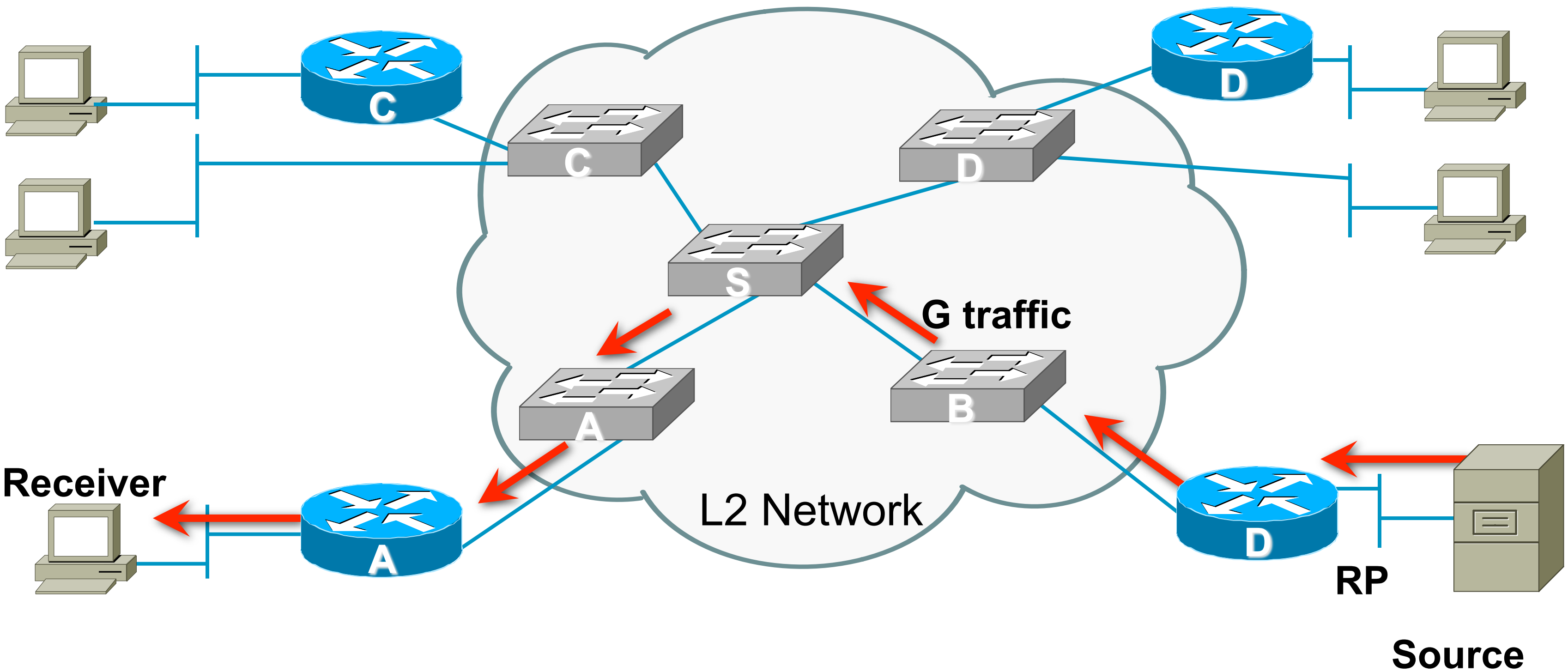
PIM Join Flow



PIM Joins Are Sent Only to the Upstream PIM Router

With PIM Snooping

Traffic Flow



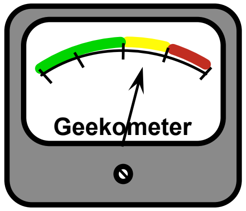
Traffic from Router B Is Sent Only to Router A

Summary—Design Issues

- Pay attention to campus topology
  - Be aware of unwanted flooding over trunks
- Host networks
  - Use IGMP snooping
- Core networks
  - Use p2p VLANs or PIM Snooping
- Address overlap
  - Select group addresses to avoid L2 overlap
  - Avoid x.0.0.x group addresses when possible

PIM Sparse Mode

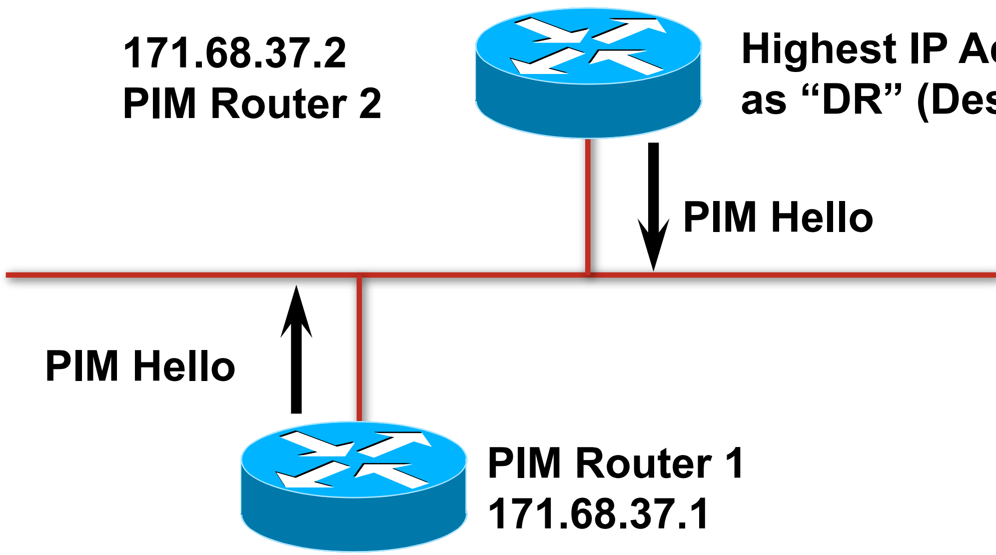
Module Agenda



- PIM Neighbor Discovery
- PIM State
- PIM SM Joining
- PIM SM Registering
- PIM SM SPT-Switchover

PIM Neighbor Discovery

PIM Neighbor Discovery



- PIMv2 Hellos are periodically multicast to the "All-PIM-Routers" (224.0.0.13) group address (default = 30 seconds)
- 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

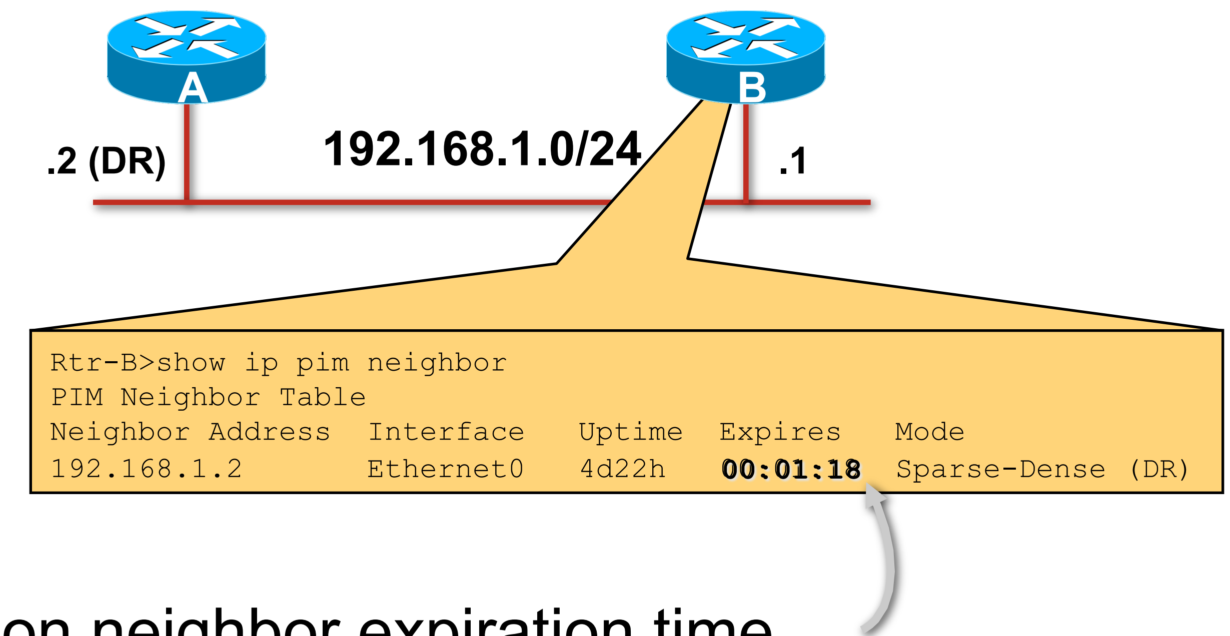
PIM Neighbor Discovery—IOS

```
wan-gw8>show ip pim neighbor
PIM Neighbor Table
Neighbor          Interface          Uptime/Expires       Ver    Mode
Address                                                          Prio/Mode
171.68.0.70       FastEthernet0/0    2w1d/00:01:24        v2     1 / B S
171.68.0.91       FastEthernet0/0    2w6d/00:01:01        v2     1 / B S
171.68.0.82       FastEthernet0/0    7w0d/00:01:14        v2     5 / DR B S
171.68.0.86       FastEthernet0/0    7w0d/00:01:13        v2     1 / B S
171.68.0.80       FastEthernet0/0    7w0d/00:01:02        v2     1 / B S
171.68.28.70      Serial2.31         22:47:11/00:01:16    v2     1 / B S
171.68.28.50      Serial2.33         22:47:22/00:01:08    v2     1 / B S
171.68.27.74      Serial2.36         22:47:07/00:01:21    v2     N /
171.68.28.170     Serial0.70         1d4h/00:01:06        v2     N /
171.68.27.2       Serial1.51         1w4d/00:01:25        v2     1 / B S
171.68.28.110     Serial3.56         1d4h/00:01:20        v2     1 / B S
171.68.28.58      Serial3.102        12:53:25/00:01:03    v2     1 / B S
```

PIM Neighbor Discovery—IOS XR

```
RP/0/5/CPU0:2001#show pim neighbor

PIM neighbors in VRF default

Neighbor Address      Interface                  Uptime    Expires    DR pri Flags

192.2.1.1*            GigabitEthernet0/2/1/1     4d22h     00:01:19 1        B A
192.2.1.2             GigabitEthernet0/2/1/1     00:01:54  00:01:23 1 (DR)
192.101.1.1*          GigabitEthernet0/2/1/0.101 4d22h     00:01:18 1        B A
192.101.1.2           GigabitEthernet0/2/1/0.101 00:01:54  00:01:23 1 (DR)
192.102.1.1*          GigabitEthernet0/2/1/0.102 00:00:57  00:01:22 1        B A
192.102.1.2           GigabitEthernet0/2/1/0.102 00:00:53  00:01:21 1 (DR)
192.103.1.1*          GigabitEthernet0/2/1/0.103 00:00:57  00:01:21 1        B A
192.103.1.2           GigabitEthernet0/2/1/0.103 00:00:54  00:01:20 1 (DR)
192.104.1.1*          GigabitEthernet0/2/1/0.104 00:00:57  00:01:21 1        B A
192.104.1.2           GigabitEthernet0/2/1/0.104 00:00:54  00:01:20 1 (DR)
```

DR Failover



- Depends on neighbor expiration time
- Expiration time sent in PIM query messages

  Expiration time = 3 x <query-interval>

  Default <query-interval> = 30 seconds

  DR failover ~ 90 seconds (worst case) by default

Tuning DR Failover

- Tune PIM query interval
  - Use interface configuration command
    - `ip pim query-interval <period> [msec]`
      - Default <period> = seconds
      - "msec" keyword available beginning with 12.1(11b)E
  - Permits DR failover to be adjusted
    - Sub-second DR failover possible
    - Smaller intervals increase PIM query traffic
      - Increase is usually insignificant

PIM State

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-SM State Example—IOS

```
sj-mbone> show ip mroute
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report
Outgoing interface flags: H - Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 2w1d/00:00:00, RP 172.16.25.1, flags: SJC
  Incoming interface: Serial0/1, RPF nbr 172.16.4.1
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse-Dense, 2w1d/00:01:40
    Serial0/0, Forward/Sparse-Dense, 00:4:52/00:02:08

(172.16.8.2, 224.1.1.1), 00:00:10/00:02:59, flags: CJT
  Incoming interface: Serial0/1, RPF nbr 172.16.4.1
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse-Dense, 00:00:10/00:02:49
    Serial0/0, Forward/Sparse-Dense, 00:4:52/00:02:08
```


PIM-SM State Example—IOS XR

```
RP/0/5/CPU0:rtr#show mrib route

IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
    C - Directly-Connected Check, S - Signal, IA - Inherit Accept,
    IF - Inherit From, D - Drop, MA - MDT Address, ME - MDT Encap,
    MD - MDT Decap, MT - MDT Threshold Crossed, MH - MDT interface handle
    CD - Conditional Decap, MPLS - MPLS Decap, MF - MPLS Encap, EX - Extranet
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
    NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
    II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
    LD - Local Disinterest, DI - Decapsulation Interface
    EI - Encapsulation Interface, MI - MDT Interface, LVIF - MPLS Encap,
    EX - Extranet

(*,225.0.0.0) RPF nbr: 0.0.0.0 Flags: C
  Up: 00:02:13
  Outgoing Interface List
    GigabitEthernet0/2/1/0.102 Flags: F NS, Up: 00:02:13

(192.2.1.2,225.0.0.0) RPF nbr: 192.2.1.2 Flags:
  Up: 00:00:07
  Incoming Interface List
    GigabitEthernet0/2/1/1 Flags: A, Up: 00:00:07
  Outgoing Interface List
    GigabitEthernet0/2/1/0.102 Flags: F NS, Up: 00:00:07
```


PIM-SM (*,G) State Rules

- (*,G) creation
  - Receipt of a (*,G) Join or IGMP Report
  - Automatically if (S,G) must be created
- (*,G) reflects default group forwarding
  - IIF = RPF interface toward RP
  - OIL = interfaces
    - That received a (*,G) Join or
    - With directly connected members or
    - Manually configured
- (*,G) deletion
  - When OIL = NULL and
  - No child (S,G) state exists

PIM-SM (*,G) State Rules in XR

- (*,G) creation
  - Receipt of a (*,G) Join or
  - Receipt of an IGMPv2 Report or
  - IGMPv3 (*,G) Join
- (*,G) reflects default group forwarding
  - IIF = RPF interface toward RP
  - OIL = interfaces
    - That received a (*,G) Join or
    - With directly connected members or
    - Manually configured (static-group)
- (*,G) deletion
  - When OIL = NULL or
  - No child (S,G) state 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) State Rules in XR

- (S,G) creation

  By receipt of (S,G) Join or Prune or

  By receipt of an IGMPv3 (S,G) Join

  By “Register” process

- (S,G) reflects forwarding of “S” to “G”

  IIF = RPF Interface normally toward source

  RPF toward RP if “RP-bit” set

  OIL = Interface on which Join was received

- (S,G) deletion

  By normal (S,G) entry timeout

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 OIL Rules (XR)

- Interfaces in OIL added

  By receipt of Join message

- Interfaces in OIL removed

  By receipt of Prune message

  Interface expire timer counts down to zero

    Timer reset (to 3 min.) by receipt of periodic Join

    or

    By IGMP membership report

PIM-SM Triggered Join/Prune Rules

- Triggering Join/Prune Messages
  - (*,G) Joins are triggered when:
    - The (*,G) OIL transitions from Null to non-Null
  - (*,G) Prunes are triggered when:
    - The (*,G) OIL transitions from non-Null to Null
  - (S,G) Joins are triggered when:
    - The (S,G) OIL transitions from Null to non-Null
  - (S,G) Prunes are triggered when:
    - The (S,G) OIL transitions from non-Null to Null
  - (S,G)RP-bit Prunes are triggered when:
    - The (S,G) RPF info != the (*,G) RPF info

PIM-SM State Flags

- S = Sparse
- 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 (Cont.)

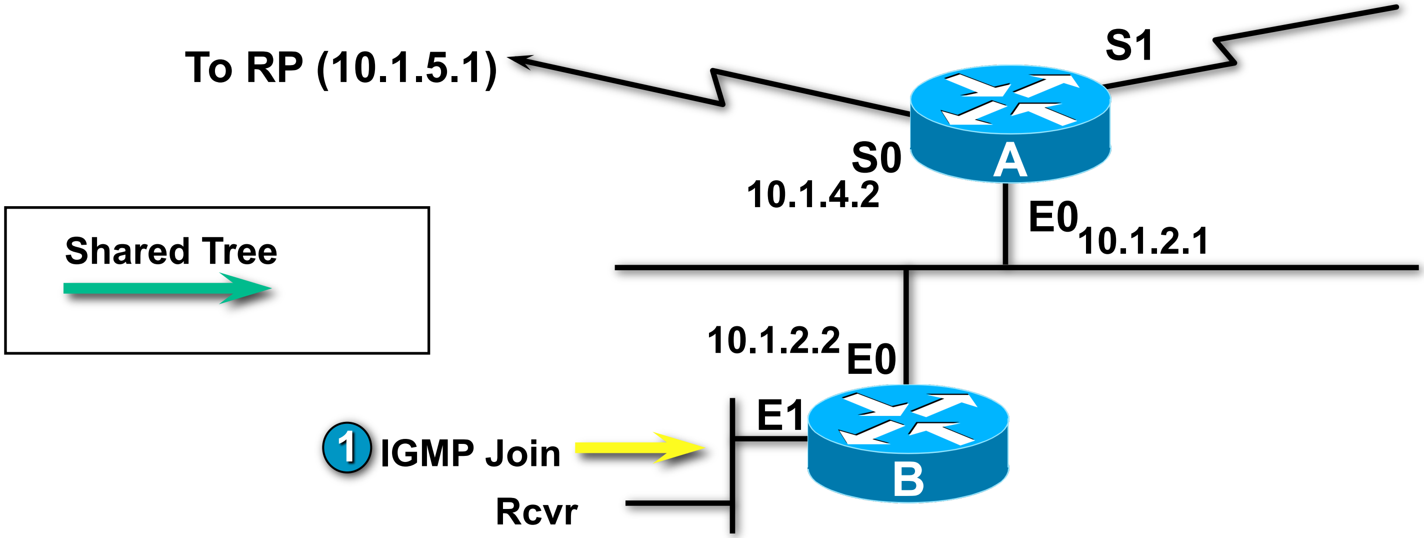
- J = Join SPT
  - In (*, G) entry
    - Indicates SPT-Threshold is being exceeded
    - Next (S,G) received will trigger join of SPT
  - In (S, G) entry
    - Indicates SPT joined due to SPT-Threshold
    - If rate < SPT-Threshold, switch back to Shared Tree
- F = Register/First-Hop
  - In (S,G) entry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

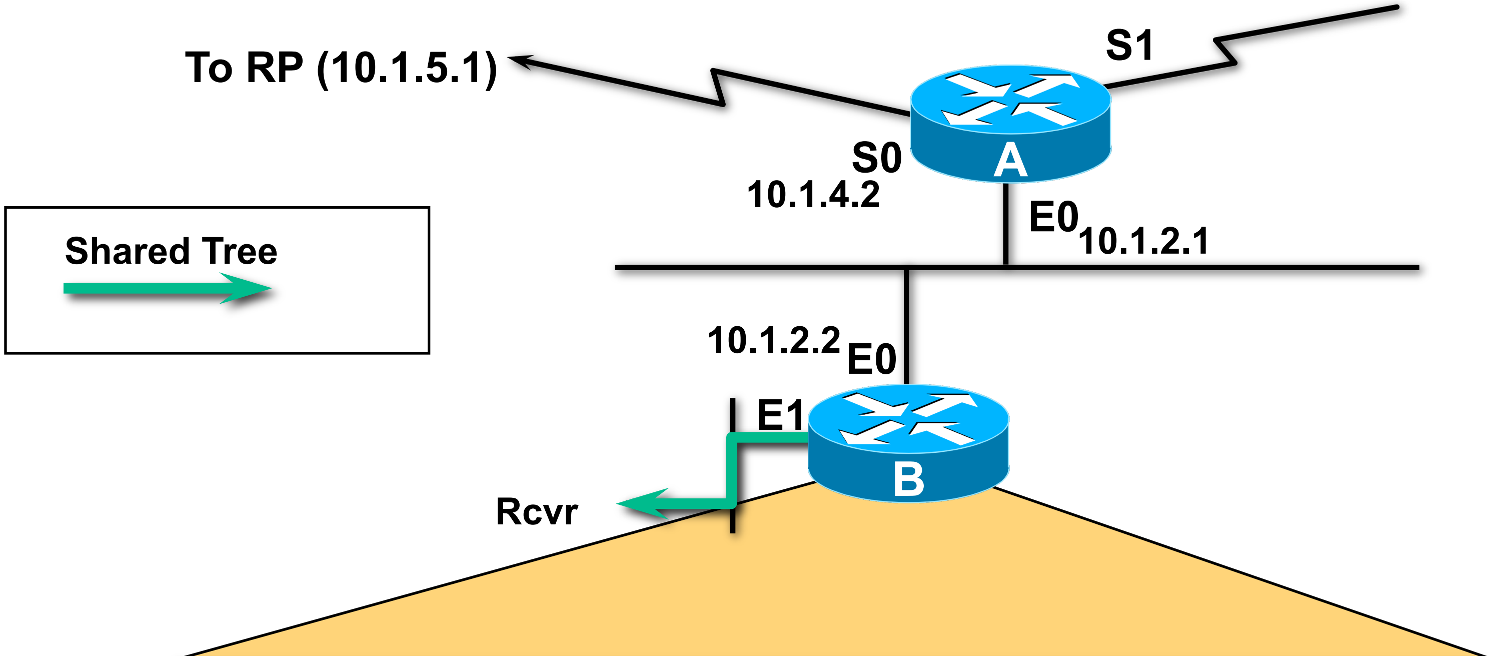
PIM SM Joining

PIM SM Joining



1 Rcvr wishes to receive group G traffic. Sends IGMP Join for G.

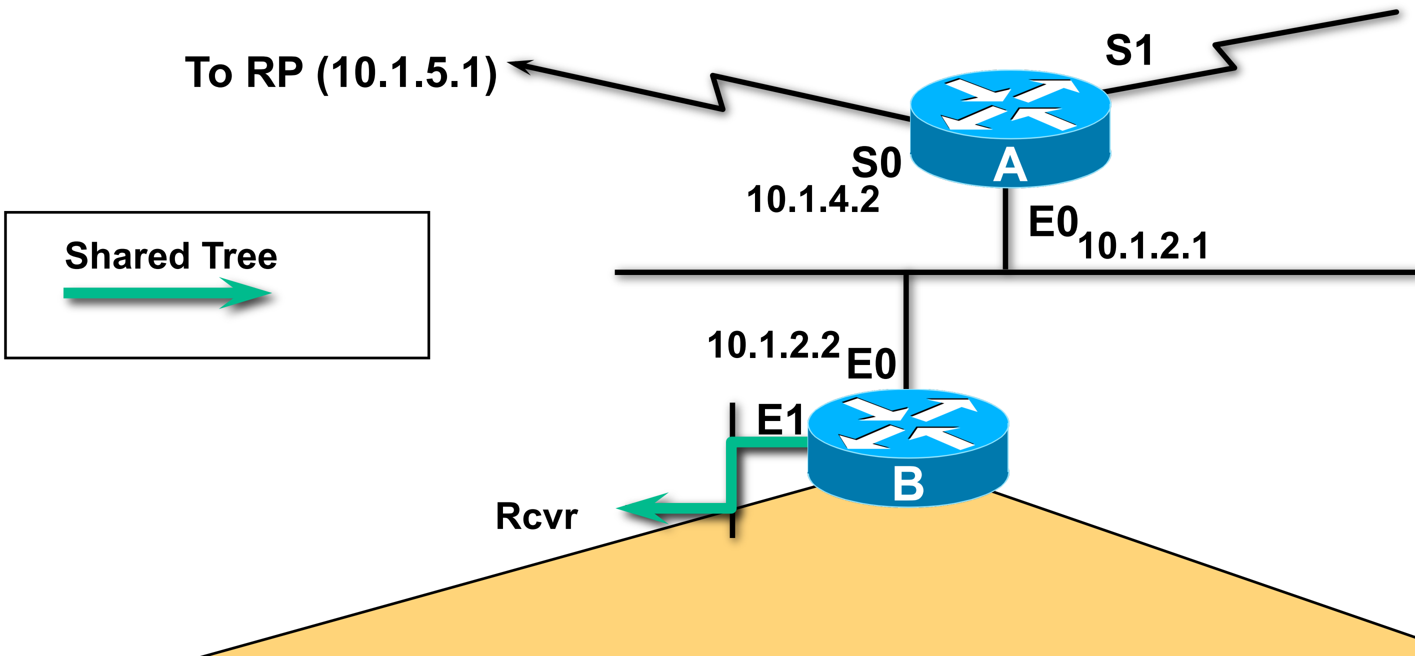
PIM SM Joining




```
(*, 224.1.1.1), 00:00:05/00:00:00, RP 10.1.5.1, flags: SC
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:00:05/00:02:54
```

B Creates (*, 224.1.1.1) State

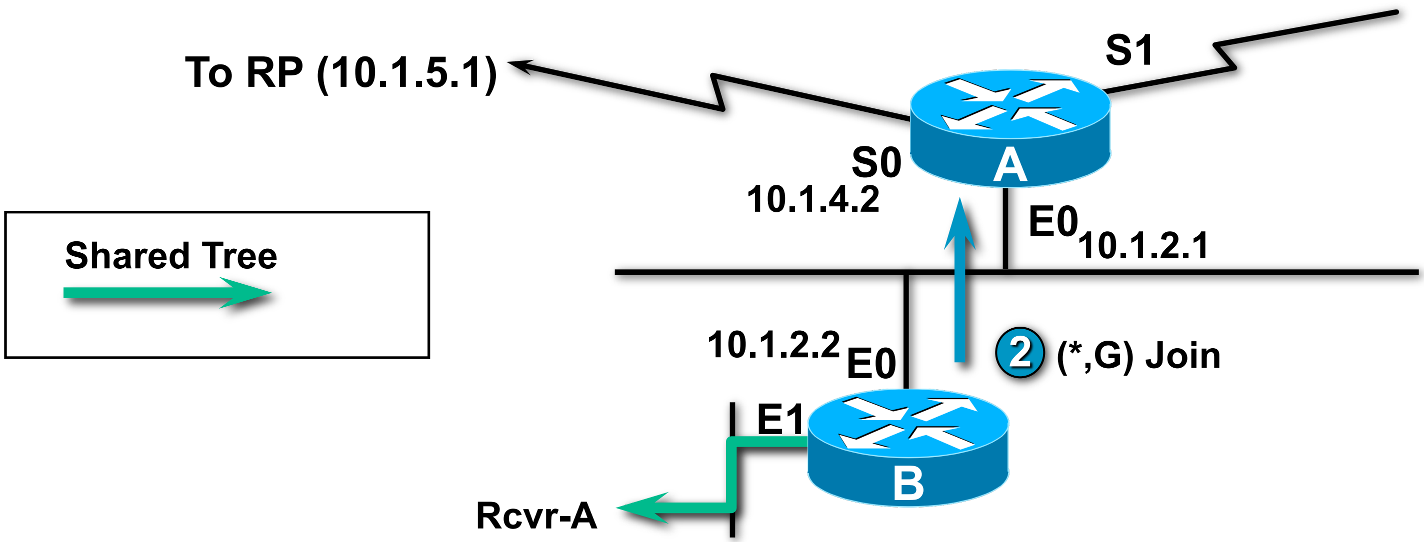
PIM SM Joining (XR)




```
(*,224.1.1.1) RPF nbr: 10.1.2.1 Flags: C
  Up: 00:09:49
  Incoming Interface List
    Ethernet0 Flags: A NS, Up: 00:05:38
  Outgoing Interface List
    Ethernet1 Flags: F NS LI, Up: 00:09:49
```

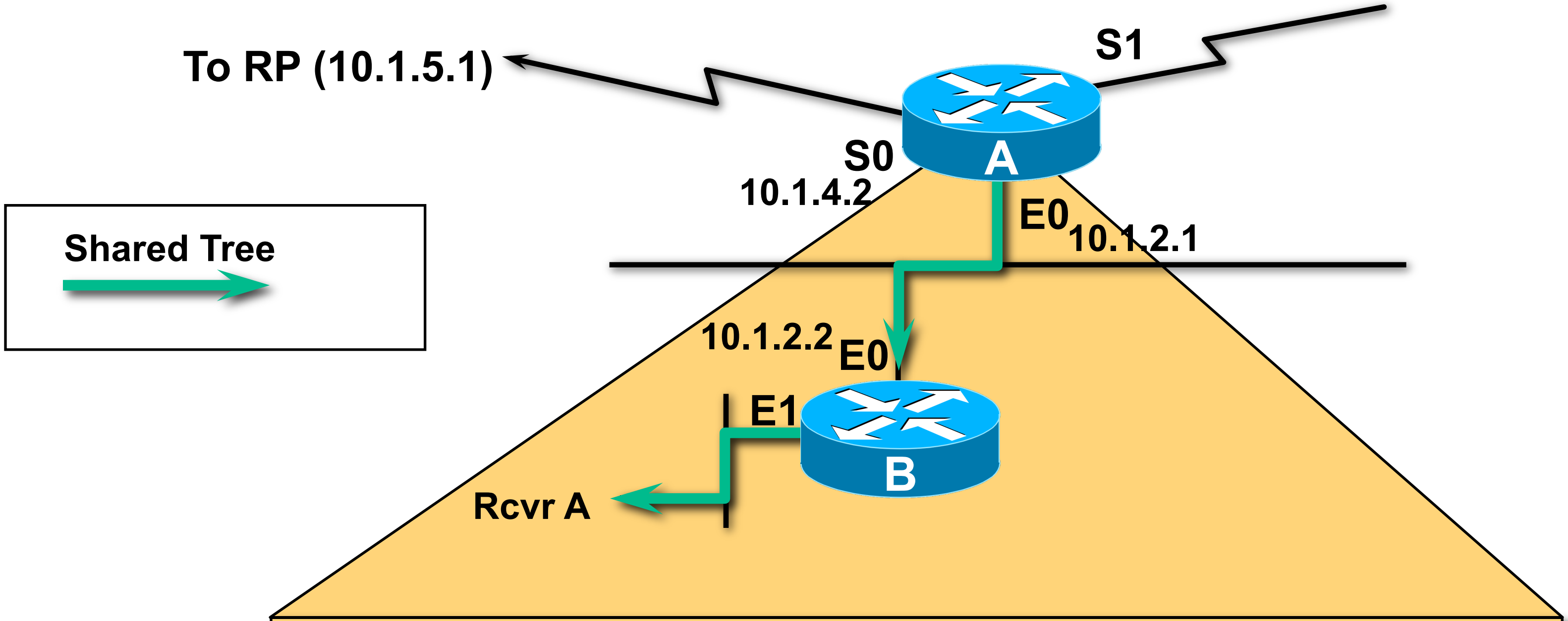
B Creates (*, 224.1.1.1) State

PIM SM Joining



1. Rcvr wishes to receive group G traffic. Sends IGMP Join for G.
2. B sends (*,G) Join towards RP.

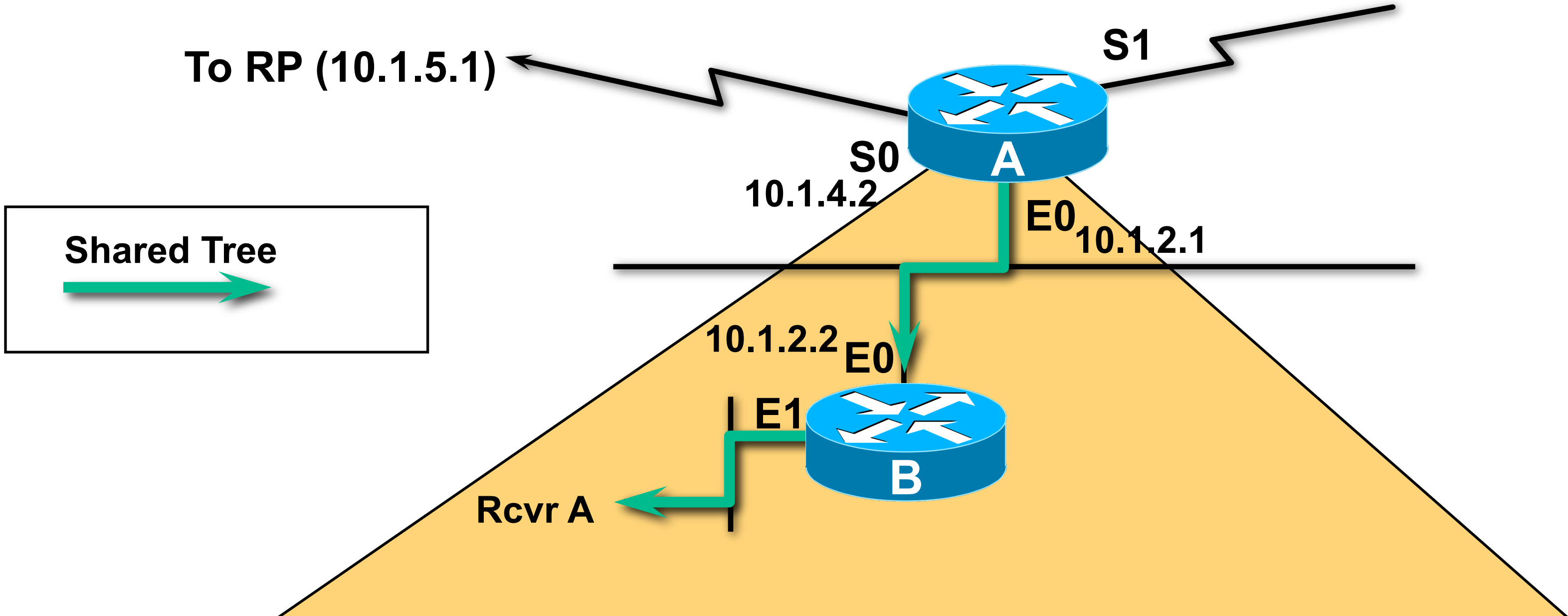
PIM SM Joining




```
(*, 224.1.1.1), 00:00:05/00:03:24, RP 10.1.5.1, flags: S
  Incoming interface: Serial0, RPF nbr 10.1.4.1
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:00:05/00:02:54
```

A Creates (*, 224.1.1.1) State

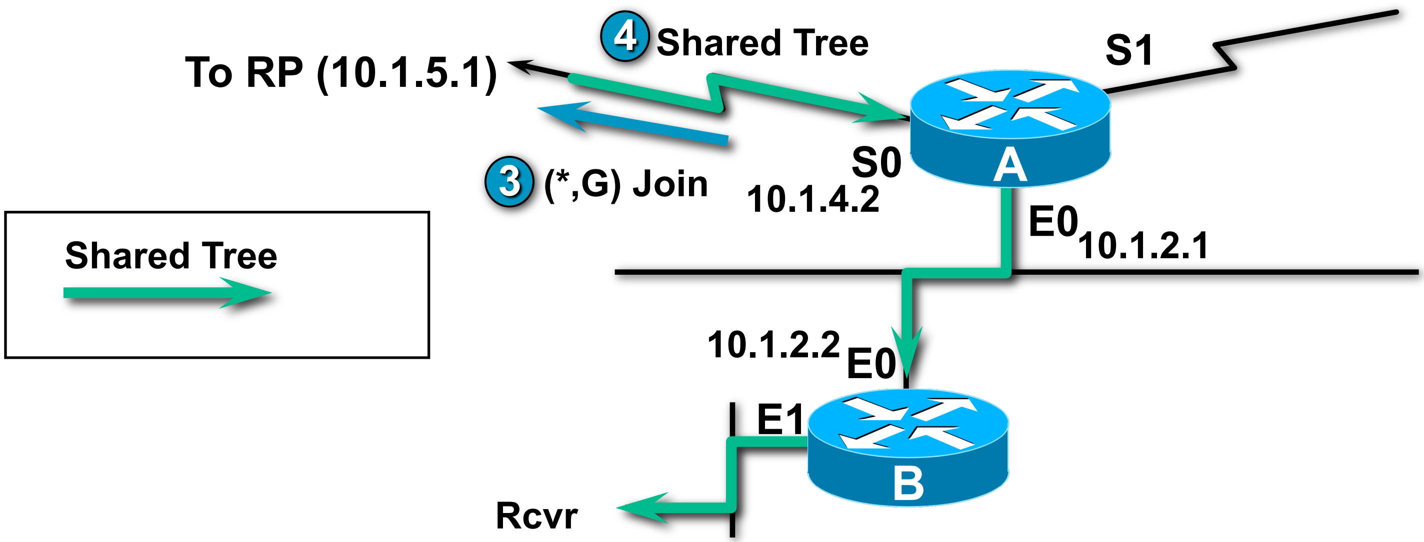
PIM SM Joining (XR)




```
(*,224.1.1.1) RPF nbr: 10.1.4.1 Flags: C
  Up: 00:03:29
  Incoming Interface List
    Serial0 Flags: A, Up: 00:03:29
  Outgoing Interface List
    Ethernet0 Flags: F NS, Up: 00:03:29
```

A Creates (*, 224.1.1.1) State

PIM SM Joining



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

PIM SM Registering

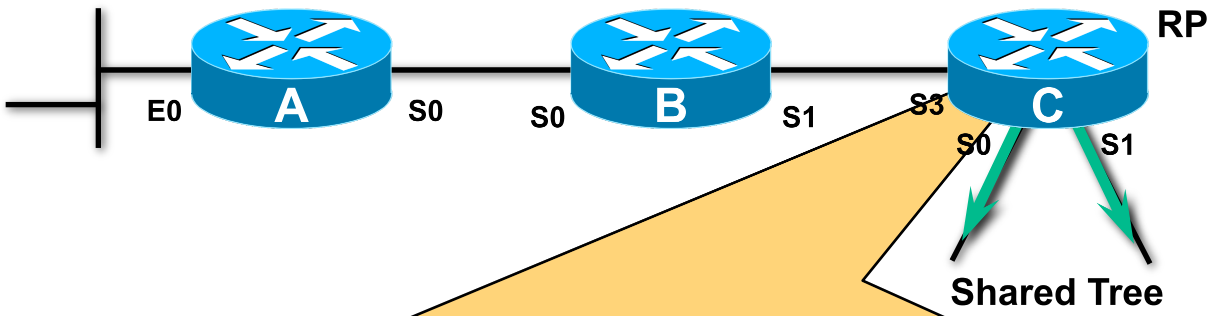
PIM SM Register Scenarios

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

PIM SM Registering: Receiver Joins First

PIM SM Registering

Receiver Joins Group First



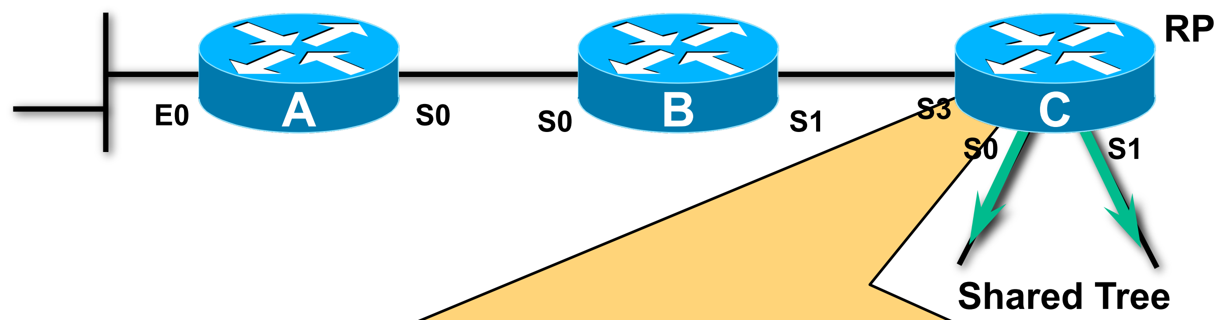

```
(*, 224.1.1.1), 00:03:14/00:02:59, RP 171.68.28.140, flags:S
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list:
    Serial0, Forward/Sparse-Dense, 00:03:14/00:03:15
    Serial1, Forward/Sparse-Dense, 00:03:14/00:03:15
```

State in "RP" Before Any Source Registers

(With Receivers on Shared Tree)

PIM SM Registering (XR)

Receiver Joins Group First



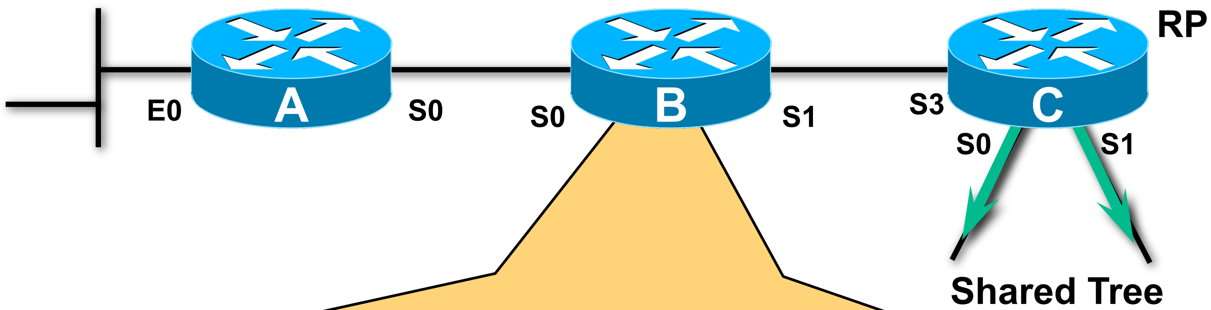

```
(*,224.1.1.1) RPF nbr: 171.68.28.140 Flags: C
  Up: 00:18:46
  Incoming Interface List
    Decapstunnel0 Flags: A, Up: 00:00:58
  Outgoing Interface List
    Serial0 Flags: F NS, Up: 00:18:46
    Serial1 Flags: F NS, Up: 00:18:46
```

State in "RP" Before Any Source Registers

(With Receivers on Shared Tree)

PIM SM Registering

Receiver Joins Group First



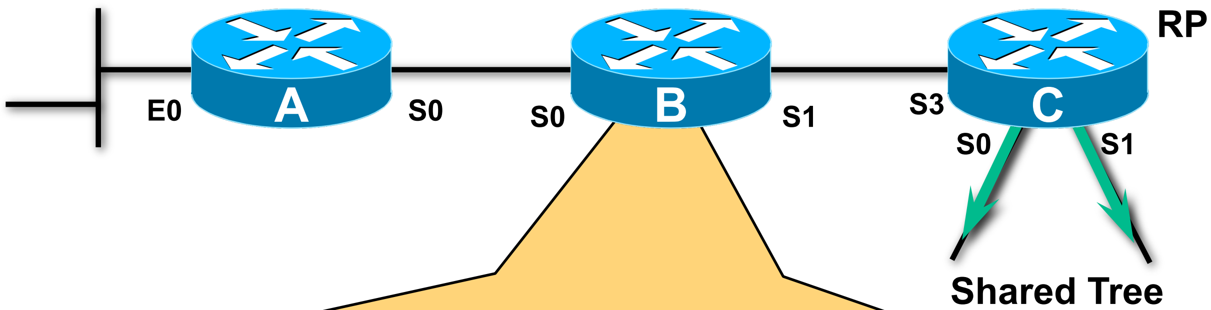

```
rtr-b>sh ip mroute 224.1.1.1

No such group
```

State in B Before Any Source Registers

(With Receivers on Shared Tree)

PIM SM Registering (XR)

Receiver Joins Group First



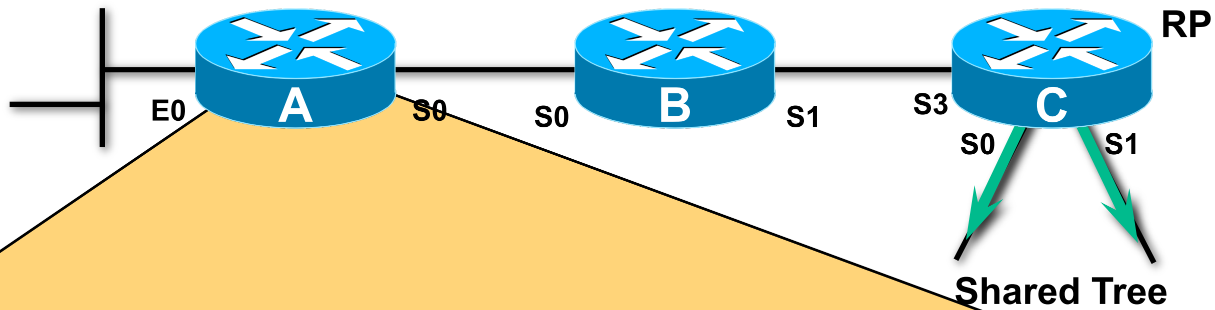

```
RP/0/5/CPU0:rtr-b#show mrib route 224.1.1.1

No matching routes in MRIB route-DB
```

State in B Before Any Source Registers

(With Receivers on Shared Tree)

PIM SM Registering

Receiver Joins Group First



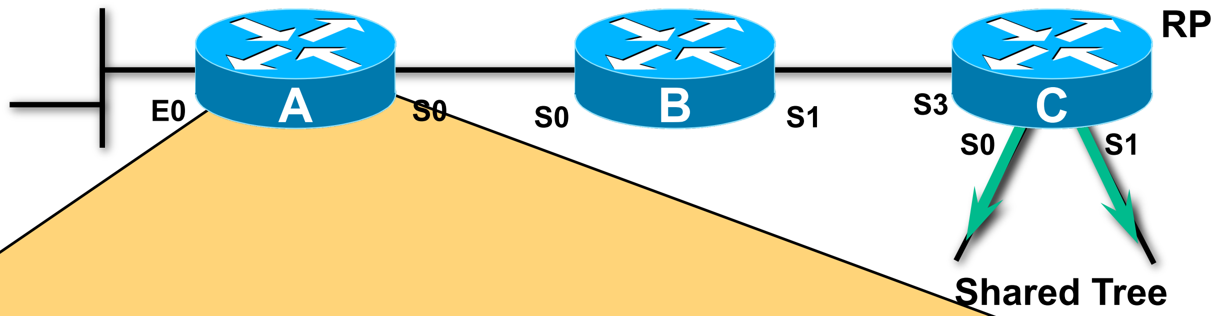

```
rtr-a>sh ip mroute 224.1.1.1

No such group.
```

State in A Before Any Source Registers

(With Receivers on Shared Tree)

PIM SM Registering (XR)

Receiver Joins Group First



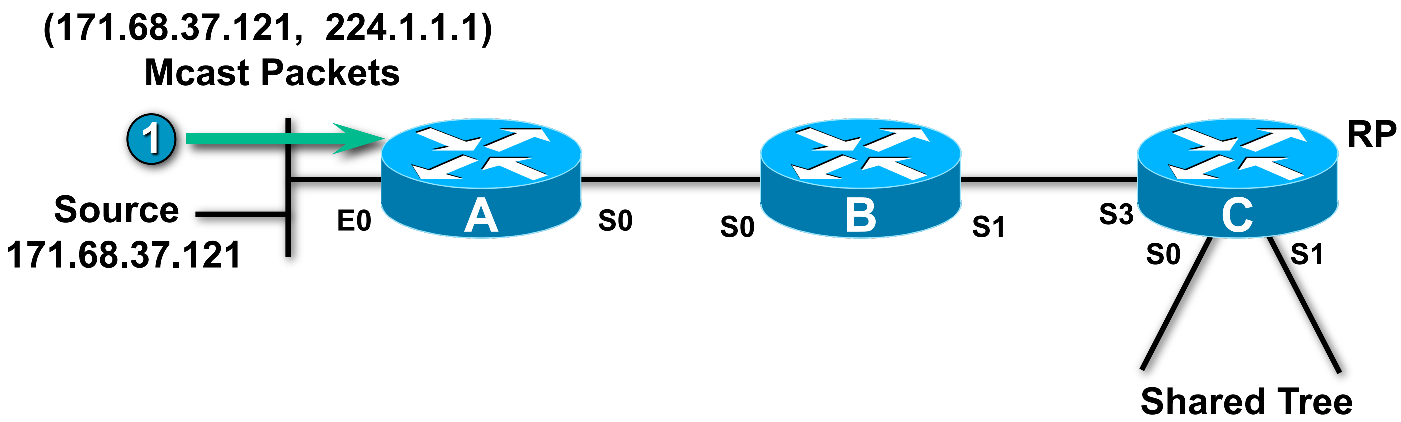

```
RP/0/5/CPU0:rtr-a#show mrib route 224.1.1.1

No matching routes in MRIB route-DB
```

State in A Before Any Source Registers

(With Receivers on Shared Tree)

PIM SM Registering

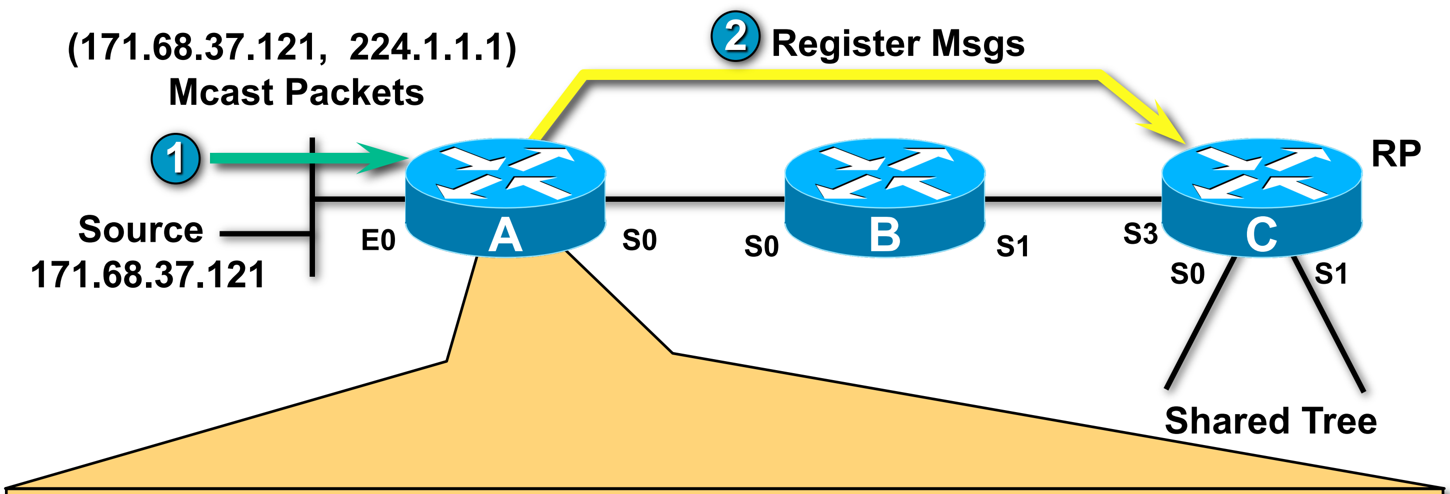
Receiver Joins Group First



1 **Source begins sending group G traffic.**

PIM SM Registering

Receiver Joins Group First




```
(*, 224.1.1.1), 00:00:03/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial0, RPF nbr 171.68.28.191,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:00:03/00:02:56, flags: FPT
  Incoming interface: Ethernet0, RPF nbr 0.0.0.0, Registering
  Outgoing interface list: Null
```

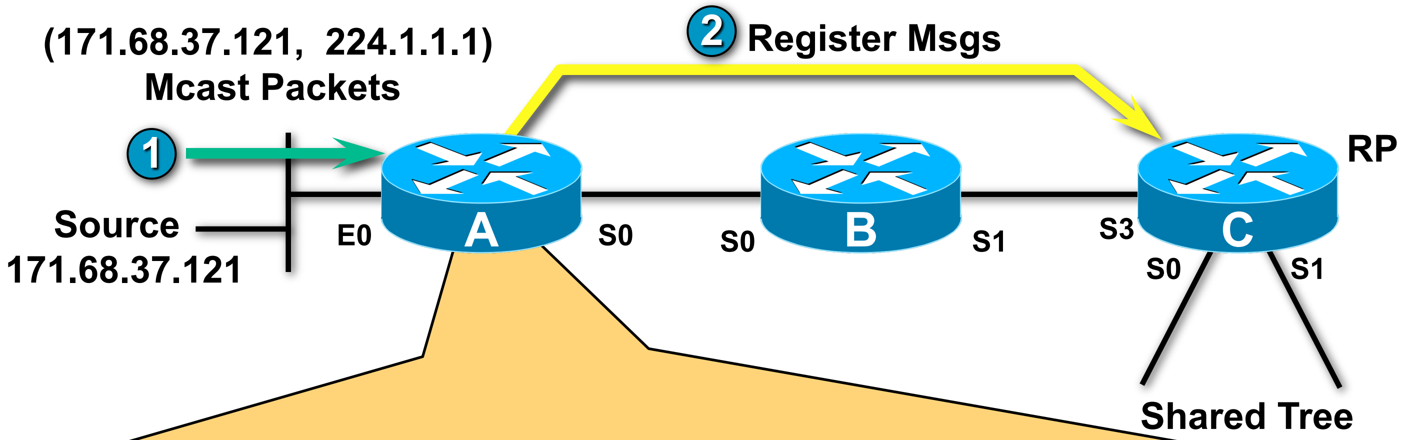
A Creates (S, G) State for Source

(After Automatically Creating a (*, G) entry)

1. **Source begins sending group G traffic.**
2. **A encapsulates packets in Registers; unicasts to RP.**

PIM SM Registering (XR)

Receiver Joins Group First



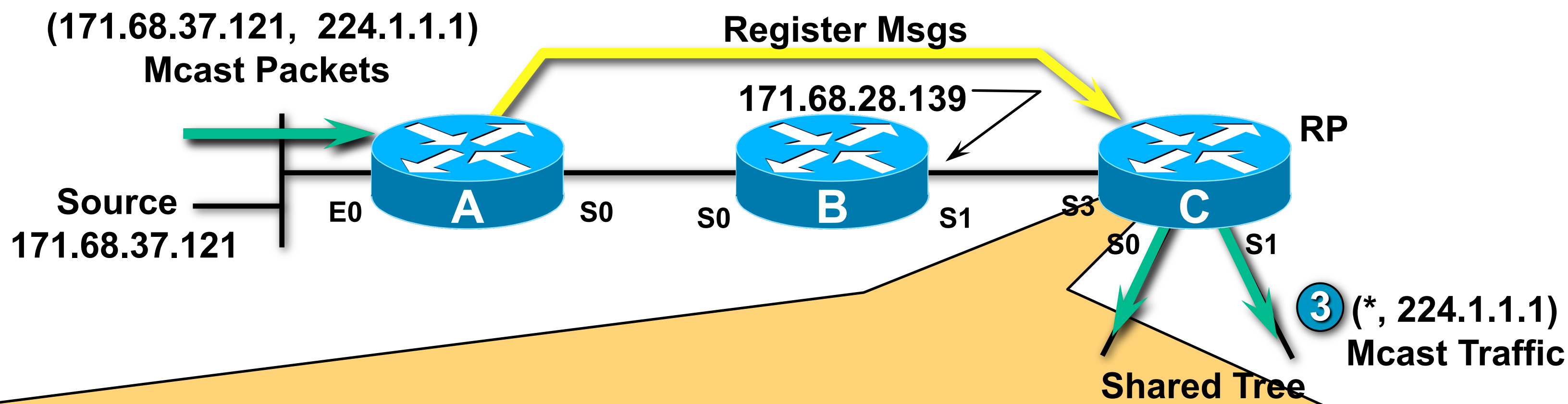

```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.37.121 Flags:
  Incoming Interface List
    Ethernet0 Flags: A, Up: 00:00:04
  Outgoing Interface List
    Encapstunnel0 Flags: F NS EI, Up: 00:00:04
```

A Creates (S, G) State for Source

(Without Automatically Creating a (*, G) entry)

1. **Source begins sending group G traffic.**
2. **A encapsulates packets in Registers; unicasts to RP.**

PIM SM Registering
Receiver Joins Group First



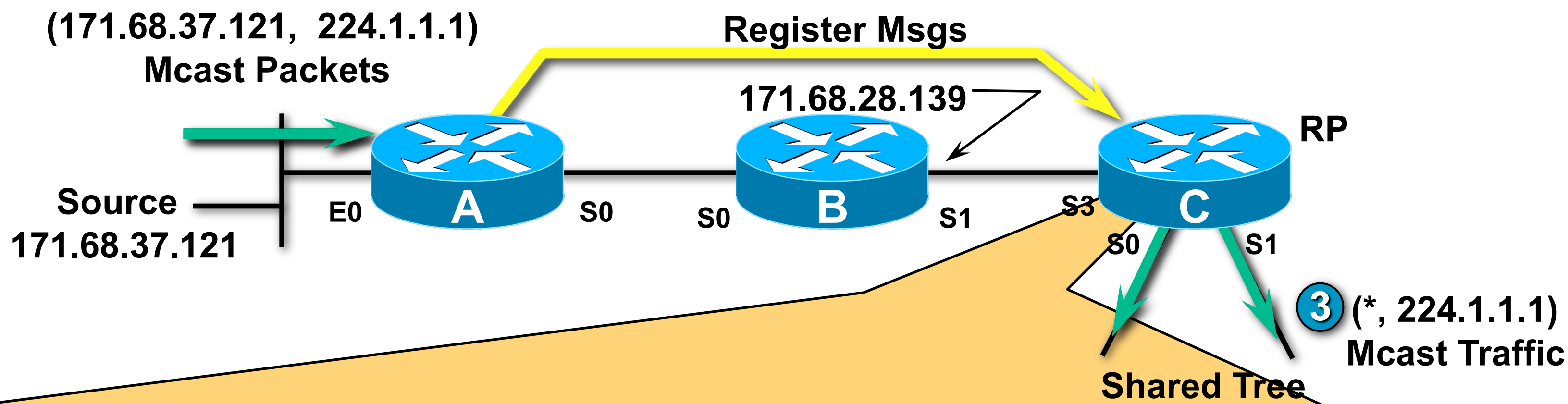

```
(*, 224.1.1.1), 00:09:21/00:00:00, RP 171.68.28.140, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list:
    Serial0, Forward/Sparse-Dense, 00:09:21/00:02:38
    Serial1, Forward/Sparse-Dense, 00:03:14/00:02:46

(171.68.37.121, 224.1.1.1, 00:01:15/00:02:46, flags:
  Incoming interface: Serial3, RPF nbr 171.68.28.139,
  Outgoing interface list:
    Serial0, Forward/Sparse-Dense, 00:00:49/00:02:11
    Serial1, Forward/Sparse-Dense, 00:00:49/00:02:11
```

“RP” Processes Register; Creates (S, G) State

3 RP (C) de-encapsulates packets; forwards down Shared tree.

PIM SM Registering (XR)

Receiver Joins Group First



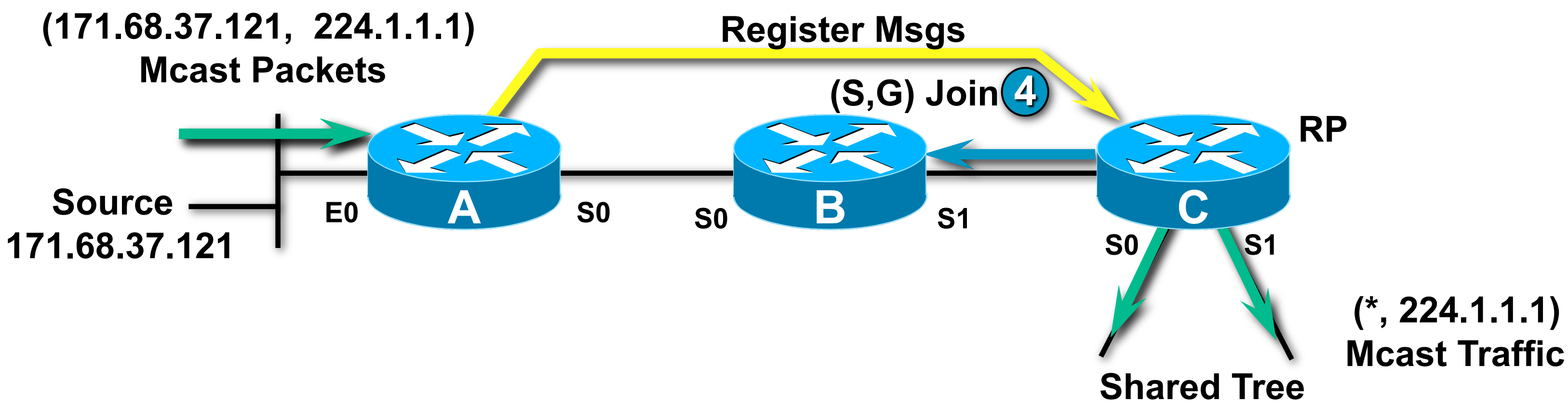

```
(*,224.1.1.1) RPF nbr: 171.68.28.140 Flags:
  Incoming Interface List
    Decapstunnel0 Flags: A NS, Up: 00:04:42
  Outgoing Interface List
    Serial0 Flags: F NS, Up: 12:27:21
    Serial1 Flags: F NS, Up: 12:27:21

(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.140 Flags:
  Incoming Interface List
    Decapstunnel0 Flags: A, Up: 00:00:09
  Outgoing Interface List
    Serial0 Flags: F NS, Up: 00:00:09
    Serial1 Flags: F NS, Up: 00:00:09
```

"RP" Processes Register; Creates (S, G) State

3 RP (C) de-encapsulates packets; forwards down Shared tree.

PIM SM Registering

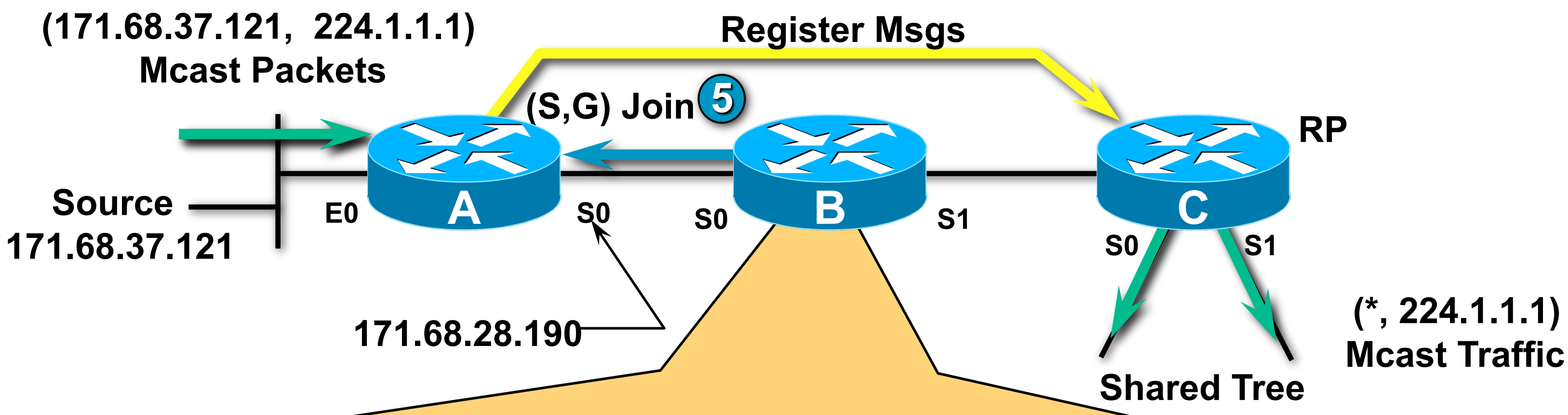
Receiver Joins Group First



4 **RP sends (S,G) Join toward Source to build SPT.**

PIM SM Registering

Receiver Joins Group First




```
(*, 224.1.1.1), 00:04:28/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial1, RPF nbr 171.68.28.140,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:04:28/00:01:32, flags:
  Incoming interface: Serial0, RPF nbr 171.68.28.190
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:04:28/00:01:32
```

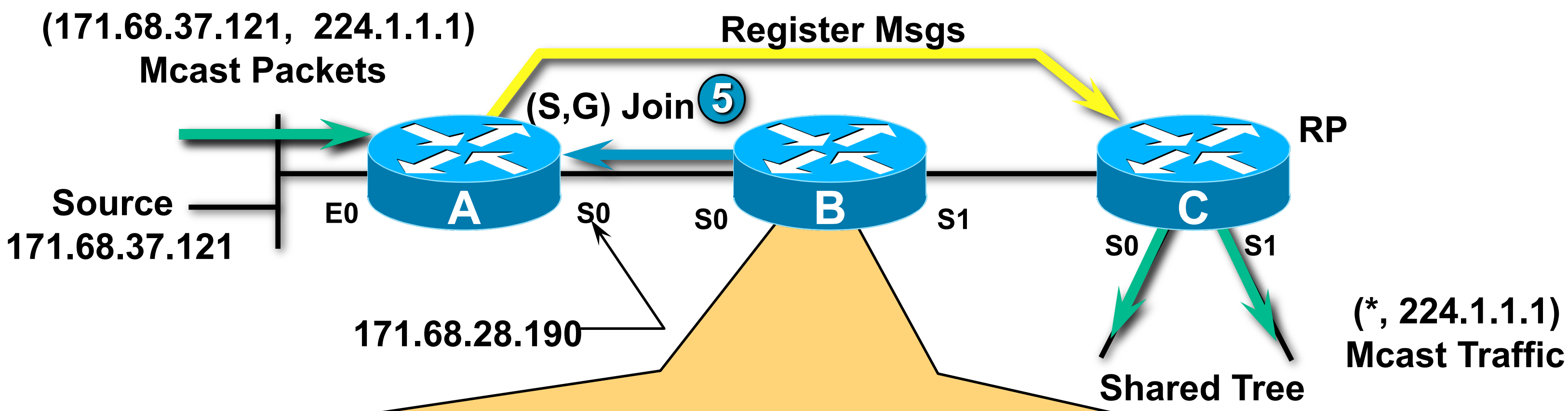
B Processes Join, Creates (S, G) State

(After Automatically Creating the (*, G) Entry)

5 B sends (S,G) Join toward Source to continue building SPT.

PIM SM Registering (XR)

Receiver Joins Group First




```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.190 Flags:
  Up: 00:00:46
  Incoming Interface List
    Serial0 Flags: A, Up: 00:00:46
  Outgoing Interface List
    Serial1 Flags: F NS, Up: 00:00:46
```

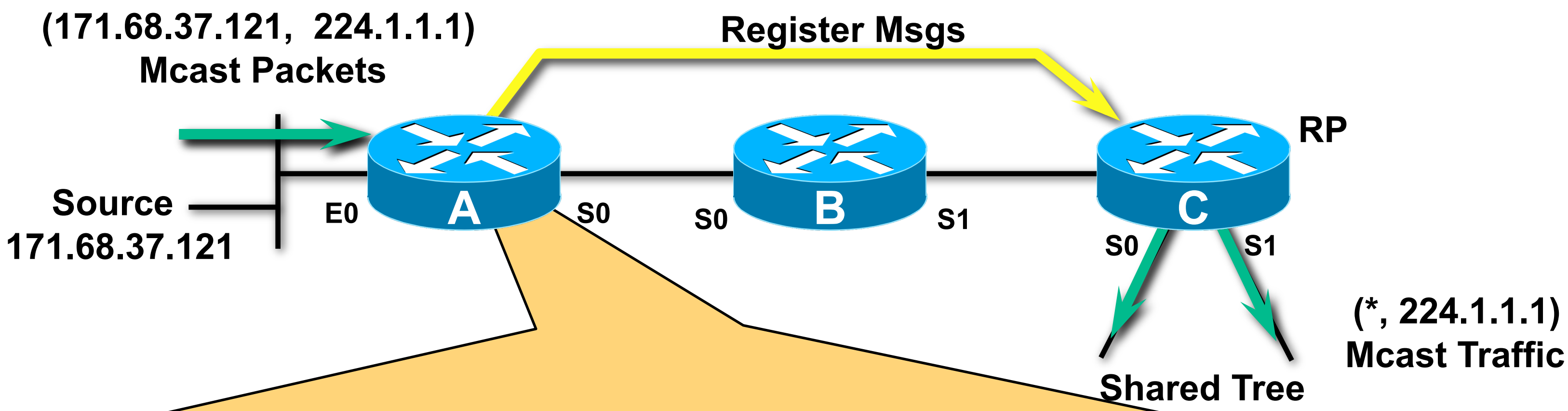
B Processes Join, Creates (S, G) State

(Without Automatically Creating the (*, G) Entry)

5 B sends (S,G) Join toward Source to continue building SPT.

PIM SM Registering

Receiver Joins Group First

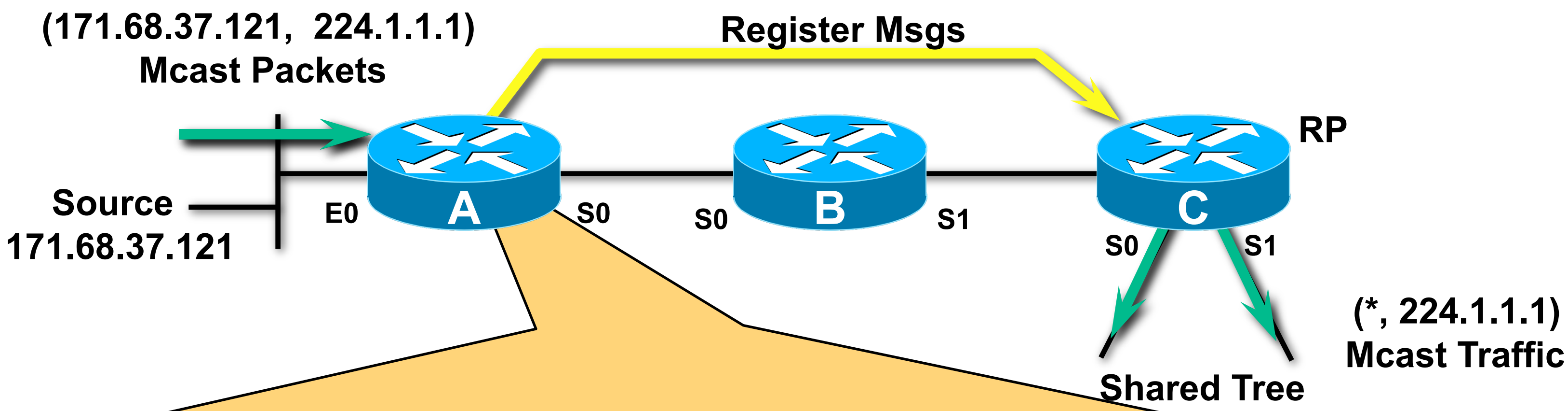



```
(*, 224.1.1.1), 00:04:28/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial0, RPF nbr 171.68.28.191,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:04:28/00:01:32, flags: FT
  Incoming interface: Ethernet0, RPF nbr 0.0.0.0, Registering
  Outgoing interface list:
    Serial0, Forward/Sparse-Dense, 00:04:28/00:01:32
```

A Processes the (S, G) Join; Adds Serial0 to OIL

PIM SM Registering (XR)

Receiver Joins Group First

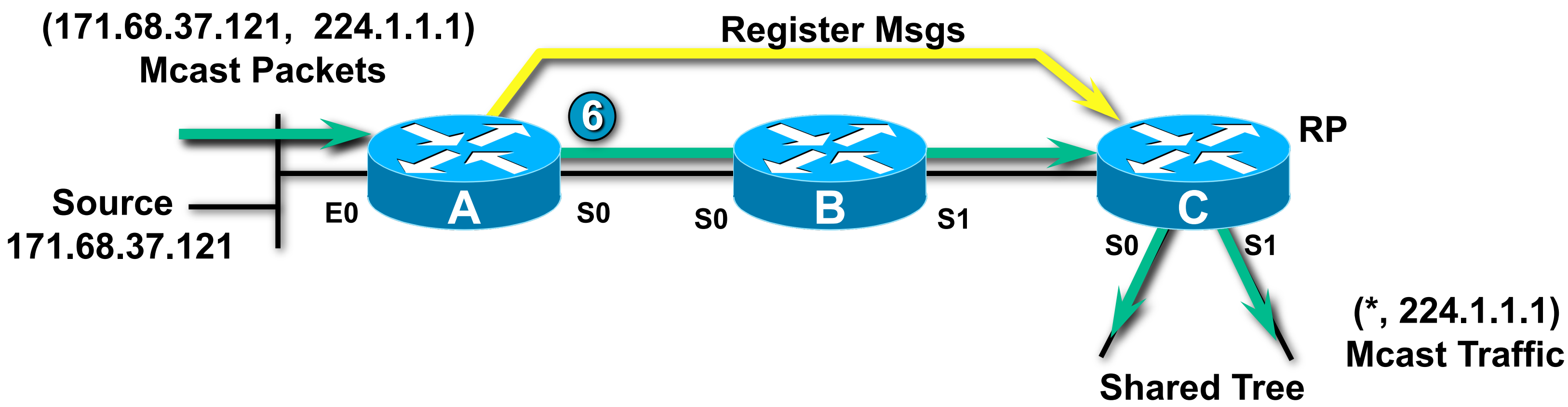



```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.37.121 Flags:
  Up: 00:00:04
  Incoming Interface List
    Ethernet0 Flags: A, Up: 00:00:04
  Outgoing Interface List
    Encapstunnel0 Flags: F NS EI, Up: 00:00:04
    Serial0 Flags: F NS, Up: 00:00:04
```

A Processes the (S, G) Join; Adds Serial0 to OIL

PIM SM Registering

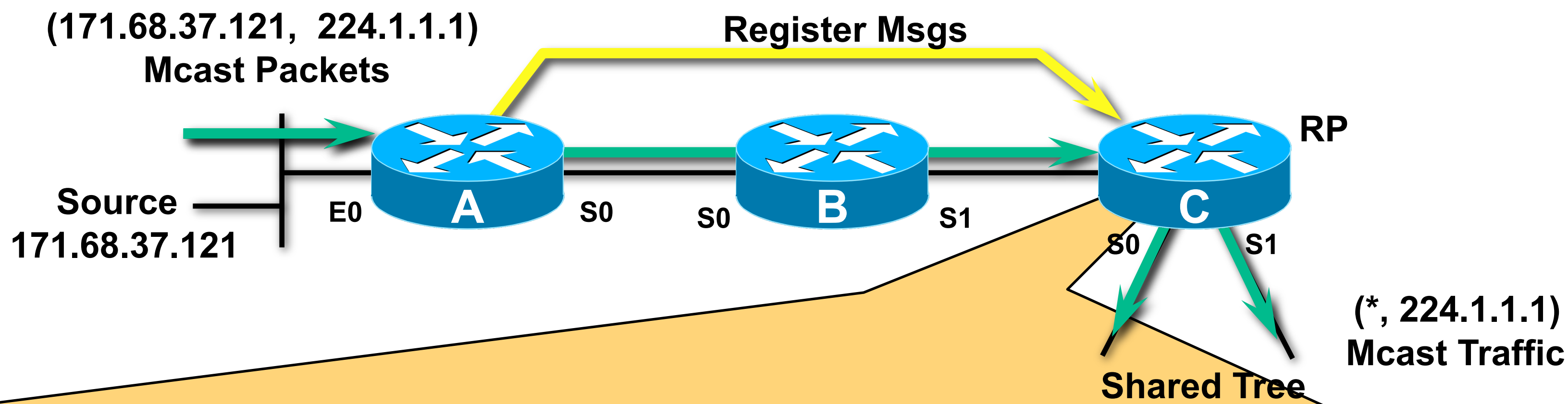
Receiver Joins Group First



6 **RP begins receiving (S,G) traffic down SPT.**

PIM SM Registering

Receiver Joins Group First



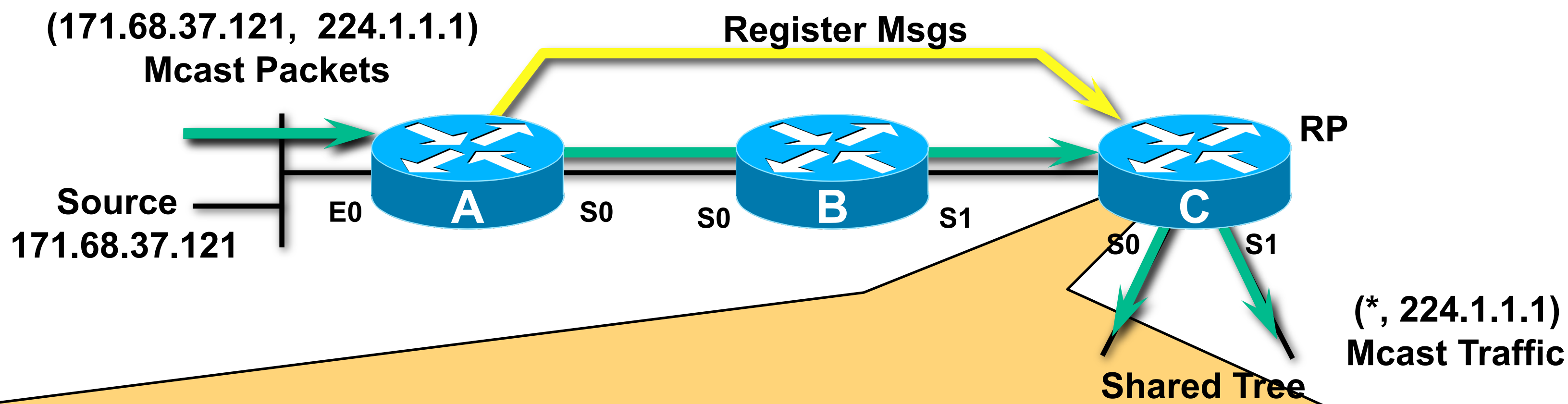

```
(*, 224.1.1.1), 00:09:21/00:00:00, RP 171.68.28.140, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list:
    Serial0, Forward/Sparse-Dense, 00:09:21/00:02:38
    Serial1, Forward/Sparse-Dense, 00:03:14/00:02:46

(171.68.37.121, 224.1.1.1, 00:01:15/00:02:46, flags:T
  Incoming interface: Serial3, RPF nbr 171.68.28.139,
  Outgoing interface list:
    Serial0, Forward/Sparse-Dense, 00:00:49/00:02:11
    Serial1, Forward/Sparse-Dense, 00:00:49/00:02:11
```

Note "T"Flag Is Now Set

Traffic Arriving via SPT Is Forwarded Down Shared Tree (This Causes the "T" Flag to Be Set)

PIM SM Registering (XR)

Receiver Joins Group First

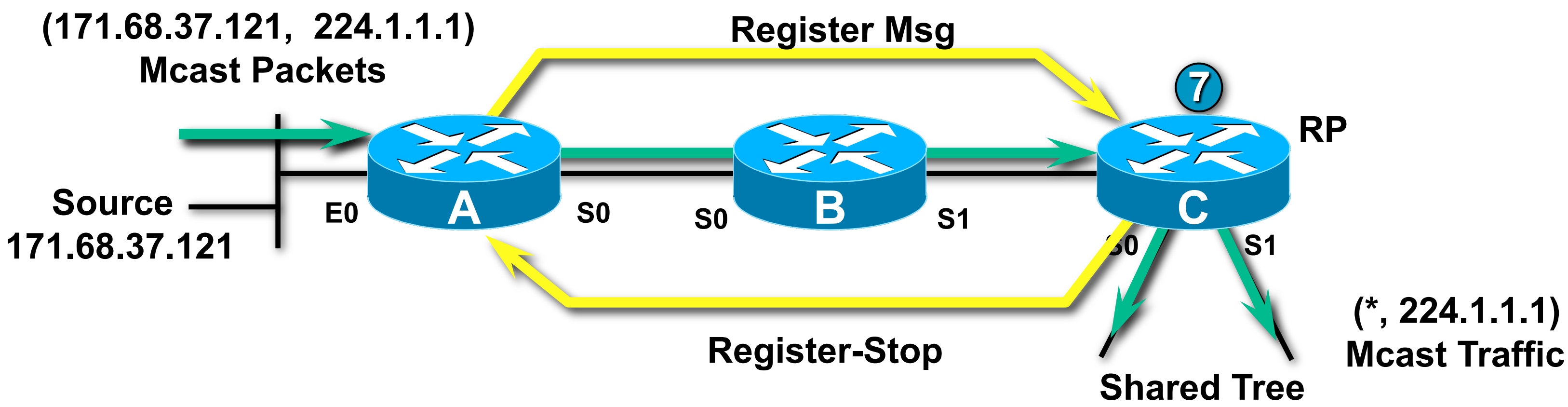



```
(*,224.1.1.1) RPF nbr: 171.68.28.140 Flags: C
  Incoming Interface List
    Decapstunnel0 Flags: A NS, Up: 00:19:40
  Outgoing Interface List
    Serial0 Flags: F NS, Up: 12:42:19
    Serial1 Flags: F NS, Up: 12:42:19

(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.139 Flags:
  Incoming Interface List
    Serial3 Flags: A, Up: 00:00:36
  Outgoing Interface List
    Serial0 Flags: F NS, Up: 00:00:36
    Serial1 Flags: F NS, Up: 00:00:36
```

Traffic Arriving via SPT Is Forwarded Down Shared Tree

PIM SM Registering

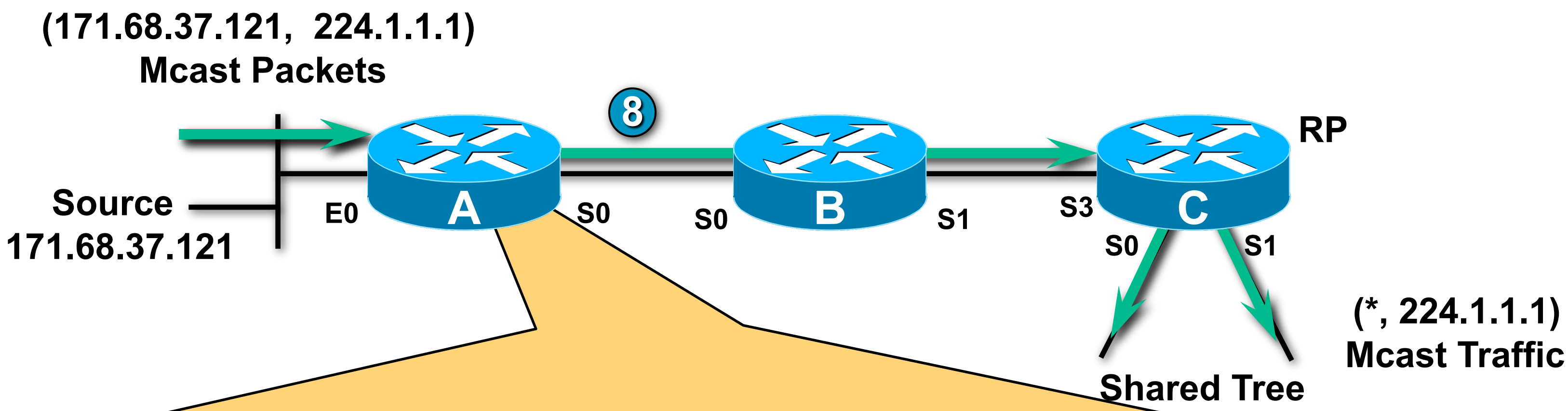
Receiver Joins Group First



⑦ **Once "T" Flag is set, next "Register" causes RP to send back a "Register-Stop" to A**

PIM SM Registering

Receiver Joins Group First




```
(*, 224.1.1.1), 00:04:28/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial0, RPF nbr 171.68.28.191,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:04:28/00:01:32, flags: FT
  Incoming interface: Ethernet0, RPF nbr 0.0.0.0,
  Outgoing interface list:
     Serial0, Forward/Sparse-Dense, 00:04:28/00:01:32
```

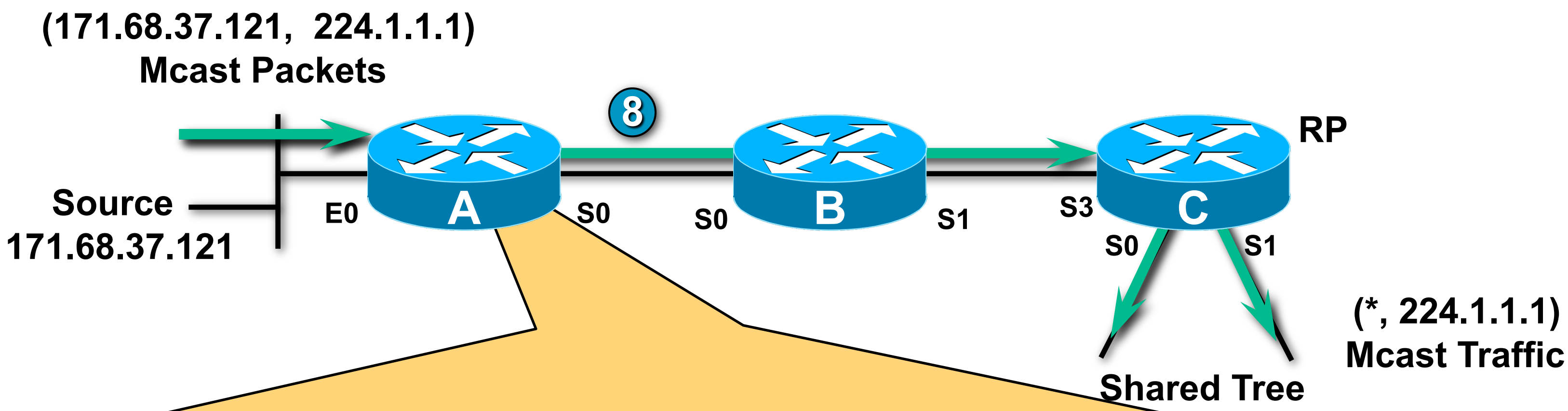
A Stops Sending Register Messages

(Final State in A)

8 **(S,G) Traffic now flowing down a single path (SPT) to RP.**

PIM SM Registering (XR)

Receiver Joins Group First




```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.37.121 Flags:
  Up: 00:00:46
  Incoming Interface List
    Ethernet0 Flags: A, Up: 00:00:46
  Outgoing Interface List
    Serial0 Flags: F NS, Up: 00:00:46
```

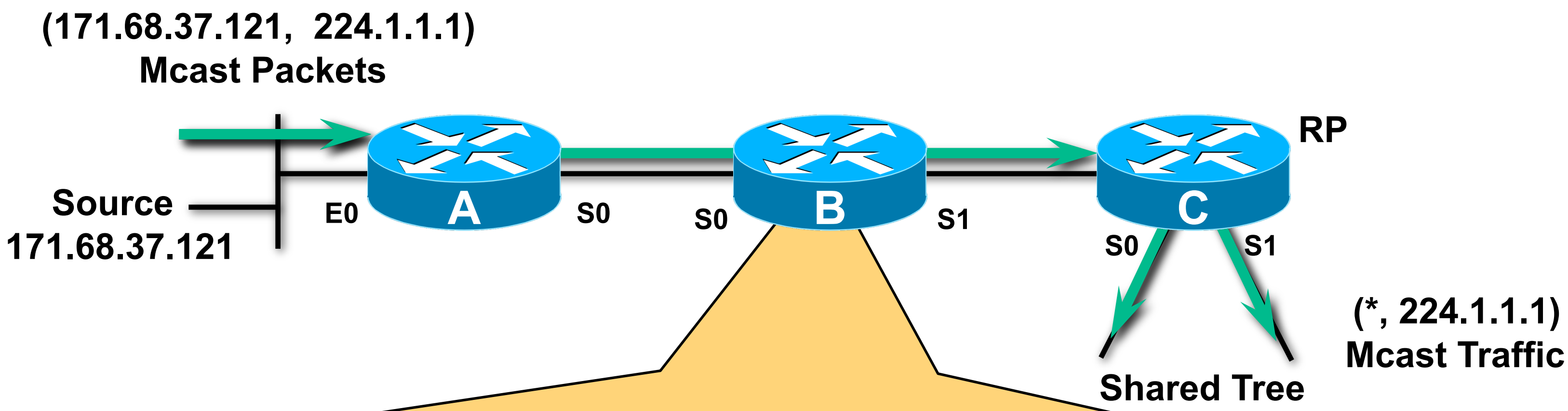
A Stops Sending Register Messages

(Final State in A)

8 (S,G) Traffic now flowing down a single path (SPT) to RP.

PIM SM Registering

Receiver Joins Group First

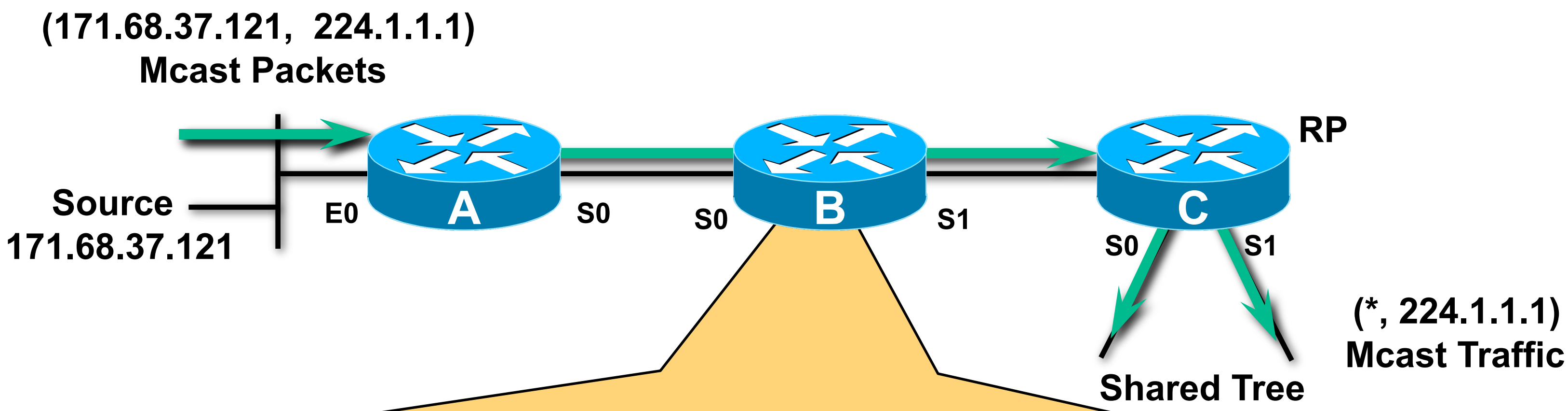



```
(*, 224.1.1.1), 00:04:28/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial1, RPF nbr 171.68.28.140,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:04:28/00:01:32, flags: T
  Incoming interface: Serial0, RPF nbr 171.68.28.190
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:04:28/00:01:32
```

Final State in B

PIM SM Registering (XR)

Receiver Joins Group First

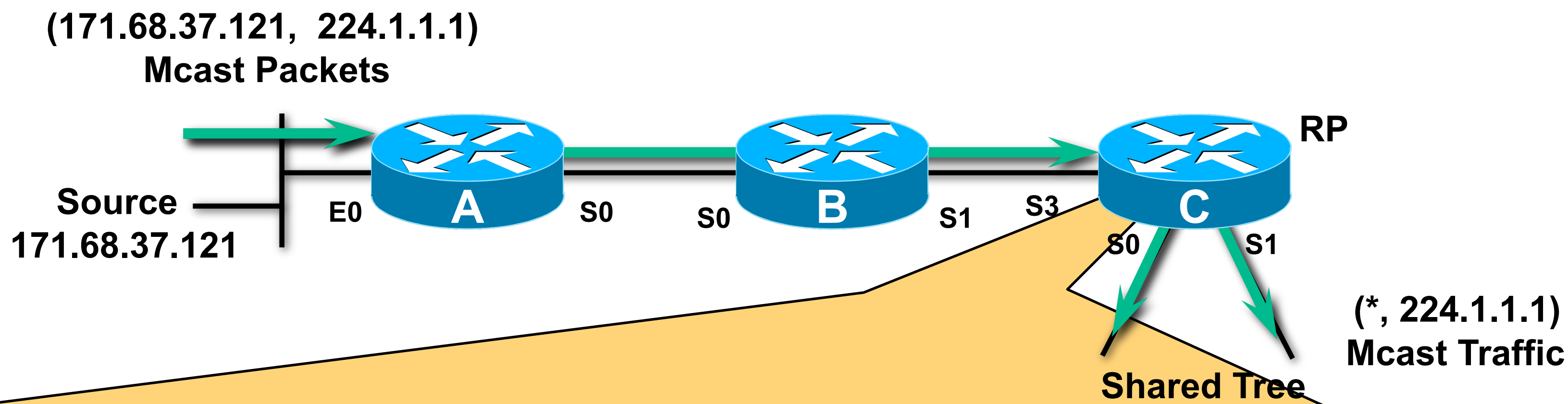



```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.190 Flags:
  Up: 00:00:46
  Incoming Interface List
    Serial0 Flags: A, Up: 00:00:46
  Outgoing Interface List
    Serial1 Flags: F NS, Up: 00:00:46
```

Final State in B

PIM SM Registering

Receiver Joins Group First



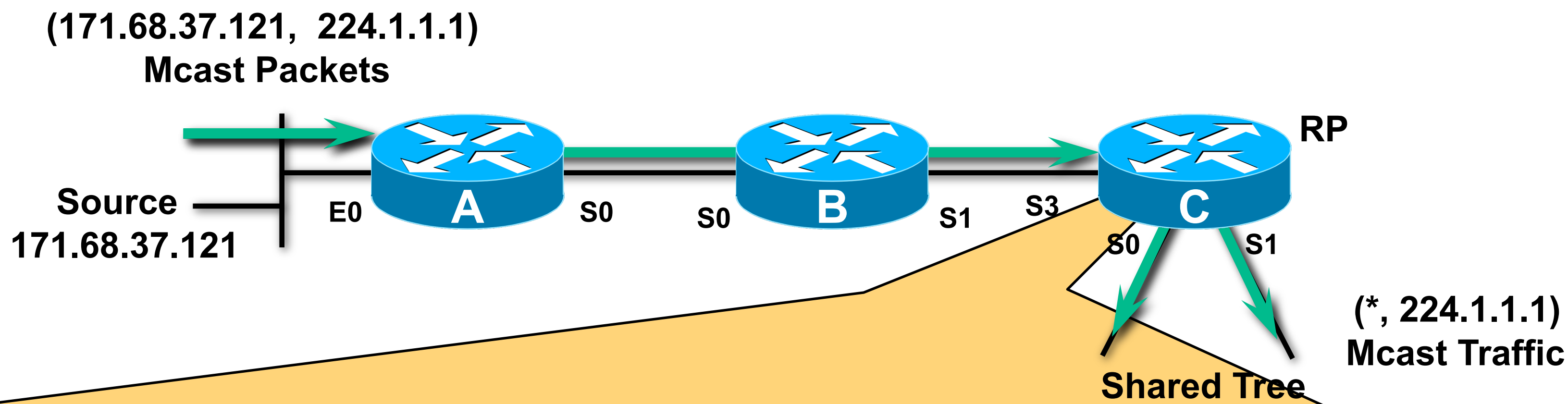

```
(*, 224.1.1.1), 00:09:21/00:00:00, RP 171.68.28.140, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list:
    Serial0, Forward/Sparse-Dense, 00:09:21/00:02:38
    Serial1, Forward/Sparse-Dense, 00:03:14/00:02:46

(171.68.37.121, 224.1.1.1, 00:01:15/00:02:46, flags: T
  Incoming interface: Serial3, RPF nbr 171.68.28.139,
  Outgoing interface list:
    Serial0, Forward/Sparse-Dense, 00:00:49/00:02:11
    Serial1, Forward/Sparse-Dense, 00:00:49/00:02:11
```

Final State in the “RP”

(With Receivers on Shared Tree)

PIM SM Registering (XR)

Receiver Joins Group First



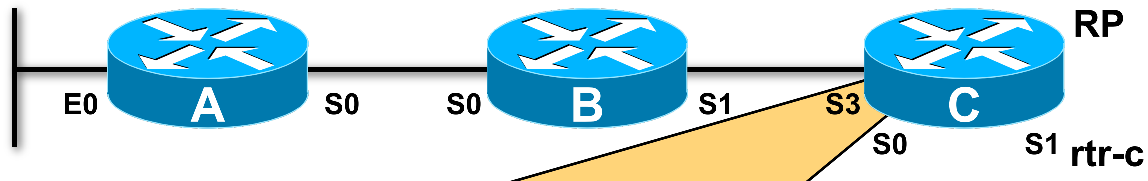

```
(*,224.1.1.1) RPF nbr: 171.68.28.140 Flags: C
  Incoming Interface List
    Decapstunnel0 Flags: A NS, Up: 00:19:40
  Outgoing Interface List
    Serial0 Flags: F NS, Up: 12:42:19
    Serial1 Flags: F NS, Up: 12:42:19

(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.139 Flags:
  Incoming Interface List
    Serial3 Flags: A, Up: 00:00:36
  Outgoing Interface List
    Serial0 Flags: F NS, Up: 00:00:36
    Serial1 Flags: F NS, Up: 00:00:36
```

Final State in the "RP"
(With Receivers on Shared Tree)

PIM SM Registering: Source Registers First

PIM SM Registering

Source Registers First




```
rtr-c>show ip mroute 224.1.1.1

Group 224.1.1.1 not found.
```

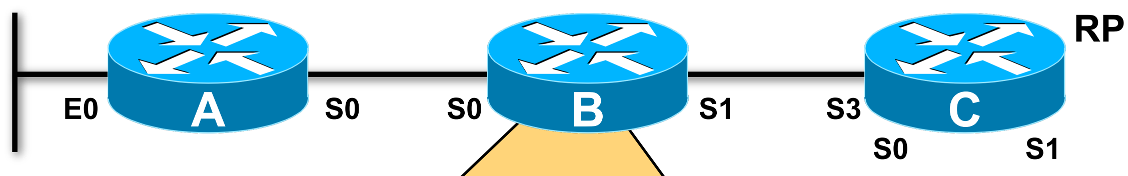
```
RP/0/5/CPU0:rtr-c#show mrib route 224.1.1.1
No matching routes in MRIB route-DB
```

State in "RP" Before Registering

(Without Receivers on Shared Tree)

PIM SM Registering

Source Registers First



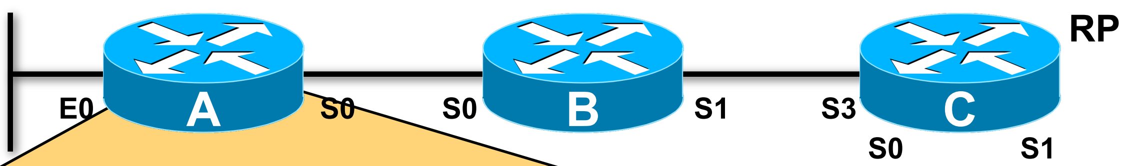

```
rtr-b>show ip mroute 224.1.1.1

Group 224.1.1.1 not found.
```

```
RP/0/5/CPU0:rtr-b#show mrib route 224.1.1.1
No matching routes in MRIB route-DB
```

State in B Before Any Source Registers
(With Receivers on Shared Tree)

PIM SM Registering

Source Registers First



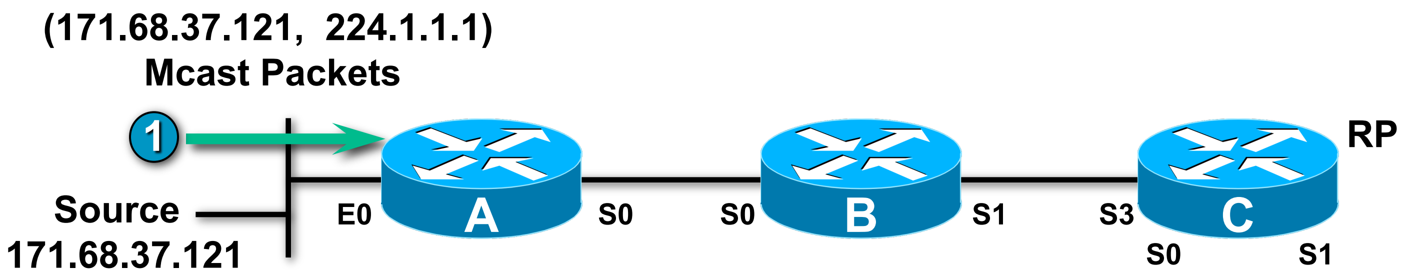

```
rtr-a>show ip mroute 224.1.1.1

Group 224.1.1.1 not found.
```

```
RP/0/5/CPU0:rtr-a#show mrib route 224.1.1.1
No matching routes in MRIB route-DB
```

State in A Before Any Source Registers

(With Receivers on Shared Tree)

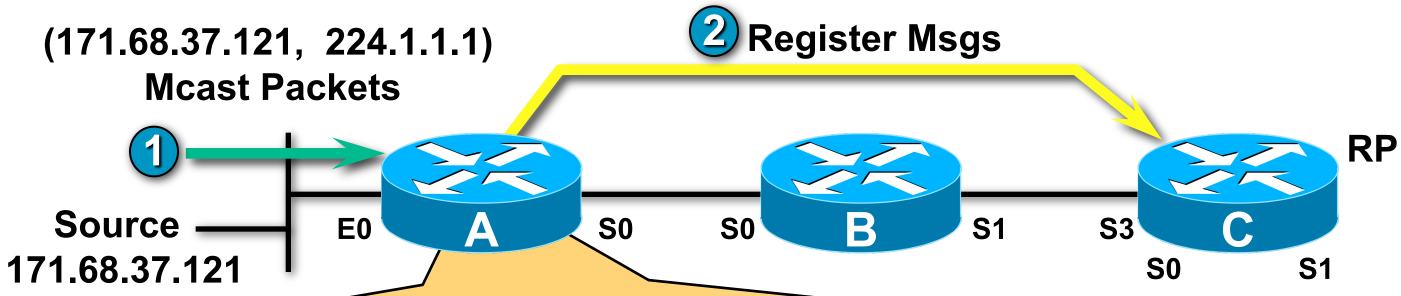
PIM SM Registering
Source Registers First



1 **Source begins sending group G traffic.**

PIM SM Registering

Source Registers First




```
(*, 224.1.1.1), 00:00:03/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial0, RPF nbr 171.68.28.191,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:00:03/00:02:56, flags: FPT
  Incoming interface: Ethernet0, RPF nbr 0.0.0.0, Registering
  Outgoing interface list: Null
```

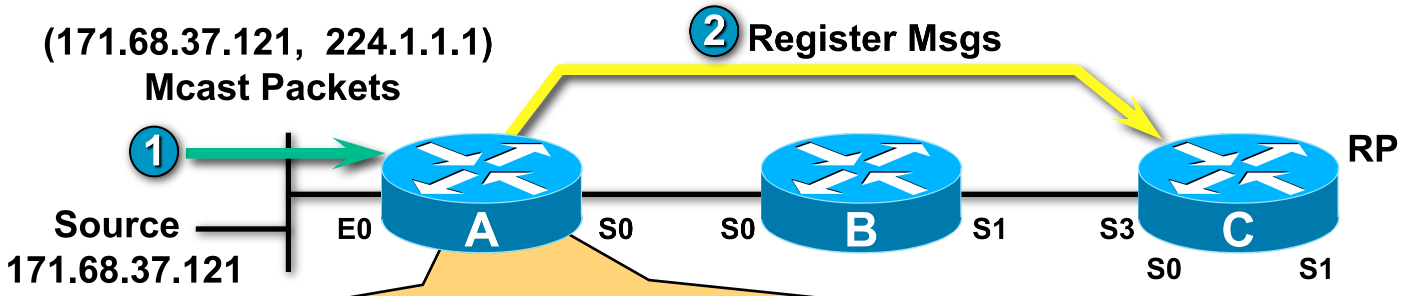
A Creates (S, G) State for Source

(After Automatically Creating a (*, G) Entry)

1. **Source begins sending group G traffic.**
2. **A encapsulates packets in Registers; unicasts to RP.**

PIM SM Registering (XR)

Source Registers First




```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.37.121 Flags:
  Up: 00:00:04
  Incoming Interface List
    Ethernet0 Flags: A, Up: 00:00:04
  Outgoing Interface List
    Encapstunnel0 Flags: F NS EI, Up: 00:00:04
```

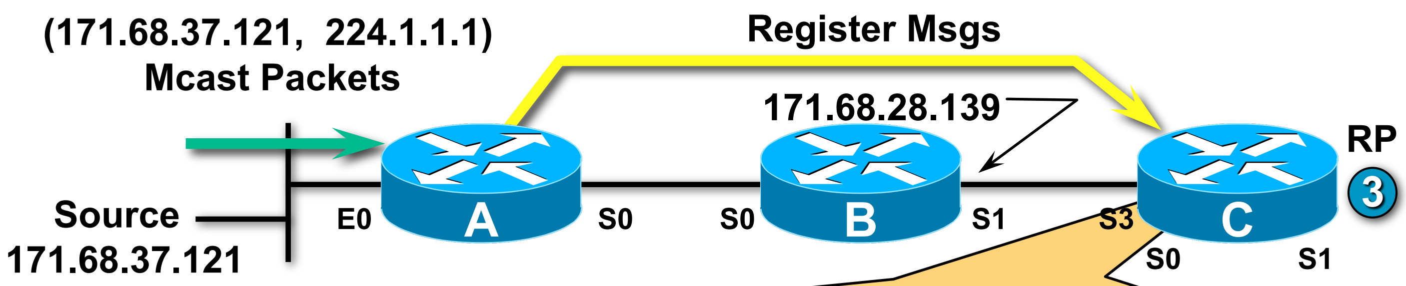
A Creates (S, G) State for Source

(Without Automatically Creating a (*, G) Entry)

1. **Source begins sending group G traffic.**
2. **A encapsulates packets in Registers; unicasts to RP.**

PIM SM Registering

Source Registers First




```
(*, 224.1.1.1), 00:01:15/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:01:15/00:01:45, flags: P
  Incoming interface: Serial3, RPF nbr 171.68.28.139,
  Outgoing interface list: Null
```

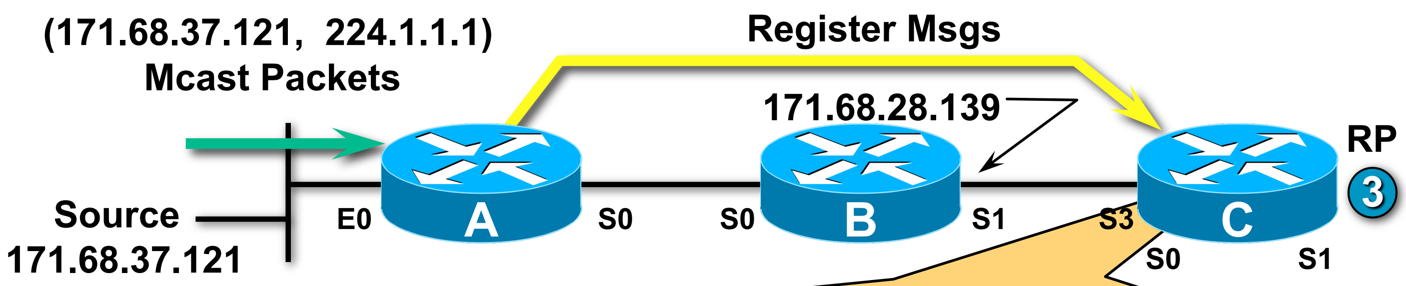
"RP" Processes Register; Creates (S, G) State

(After Automatically Creating the (*, G) Entry)**

3 RP (C) has no receivers on Shared Tree; discards packet.

PIM SM Registering (XR)

Source Registers First




```
(*,224.0.0.0/4) RPF nbr: 171.68.28.140 Flags: C
  Up: 00:50:57
  Outgoing Interface List
    Decapstunnel0 Flags: NS DI, Up: 00:50:57

(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.140 Flags: C
  Up: 00:00:07
  Incoming Interface List
    Decapstunnel0 Flags: A, Up: 00:00:07
```

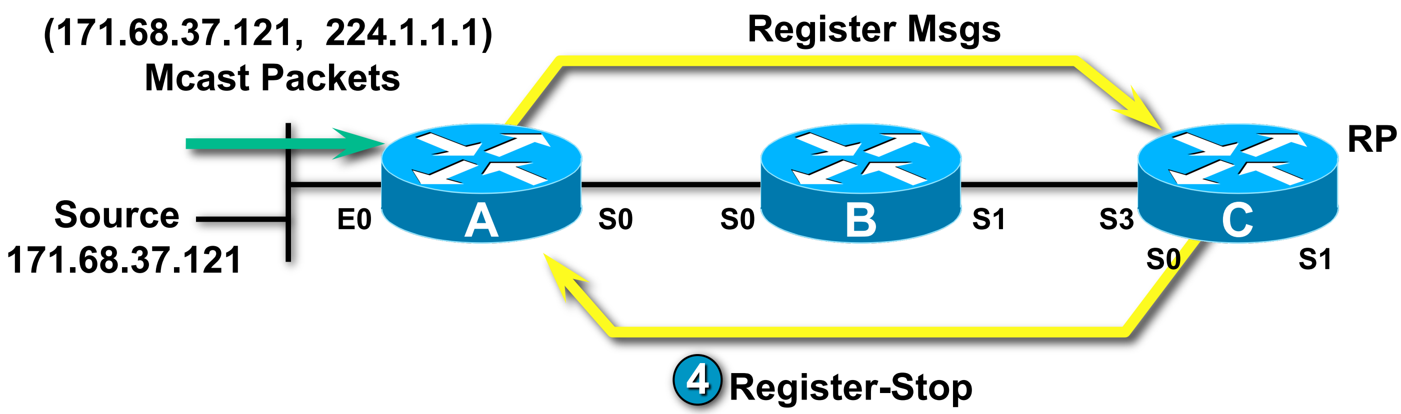
"RP" Processes Register; Creates (S, G) State

(Without Automatically Creating the (*, G) Entry)**

③ RP (C) has no receivers on Shared Tree; discards packet.

PIM SM Registering

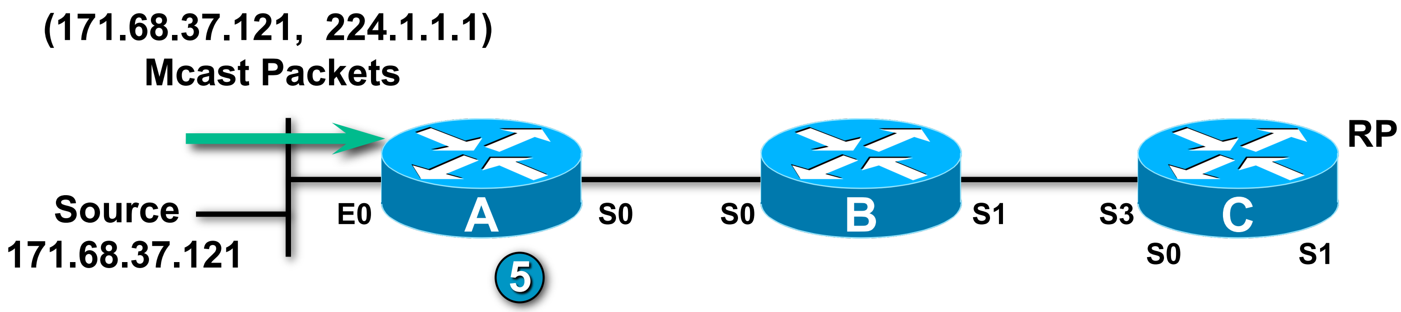
Source Registers First



4 RP sends "Register-Stop" to A.

PIM SM Registering

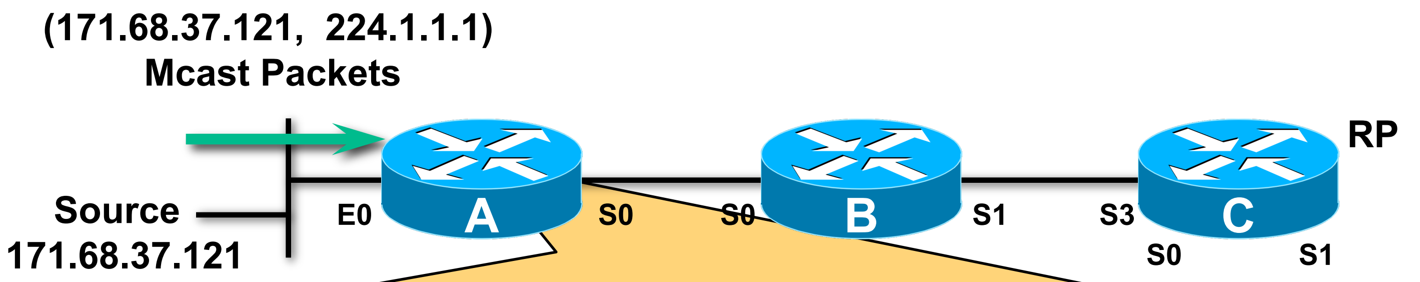
Source Registers First



5 **A stops encapsulating traffic in Register Messages; drops packets from Source.**

PIM SM Registering

Source Registers First



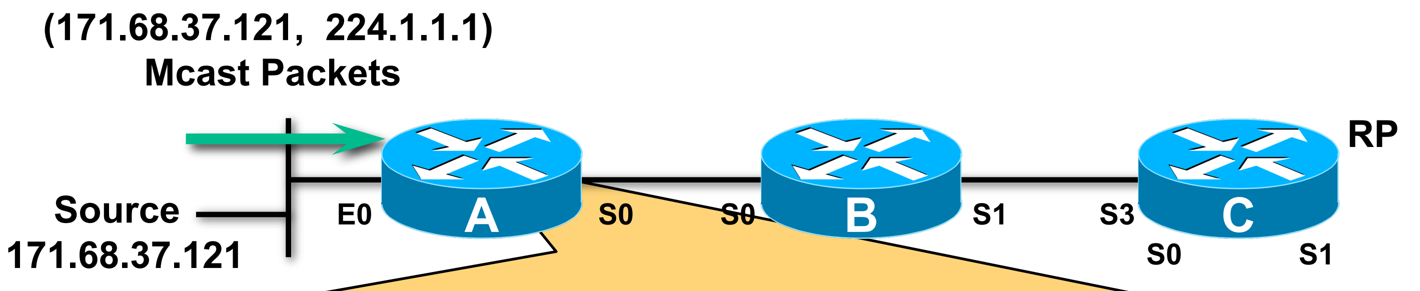

```
(*, 224.1.1.1), 00:01:28/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial0, RPF nbr 171.68.28.191,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:01:28/00:01:32, flags: FPT
  Incoming interface: Ethernet0, RPF nbr 0.0.0.0
  Outgoing interface list: Null
```

State in A After Registering

(Without Receivers on Shared Tree)

PIM SM Registering (XR)

Source Registers First



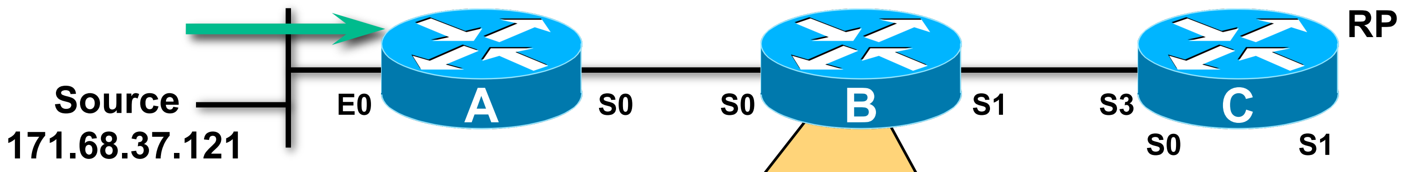

```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.37.121 Flags:
  Up: 00:00:05
  Incoming Interface List
    Ethernet0 Flags: A, Up: 00:00:05
```

State in A After Registering

(Without Receivers on Shared Tree)

PIM SM Registering

Source Registers First

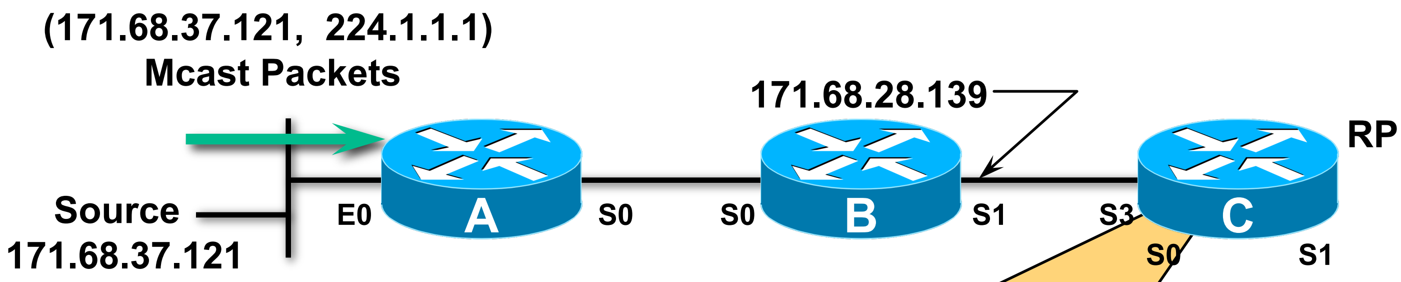



```
rtr-b>show ip mroute 224.1.1.1

Group 224.1.1.1 not found.
```

```
RP/0/5/CPU0:rtr-b#show mrib route 224.1.1.1
No matching routes in MRIB route-DB
```

State in B After A Registers
(Without Receivers on Shared Tree)

PIM SM Registering
Source Registers First

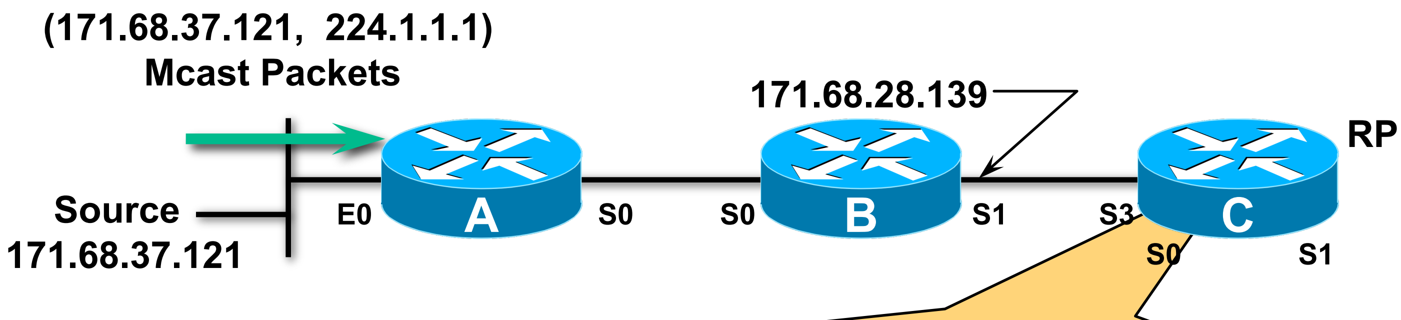



```
(*, 224.1.1.1), 00:01:15/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:01:15/00:01:45, flags: P
  Incoming interface: Serial3, RPF nbr 171.68.28.139,
  Outgoing interface list: Null
```

State in RP After A Registers
(Without Receivers on Shared Tree)

PIM SM Registering (XR)

Source Registers First



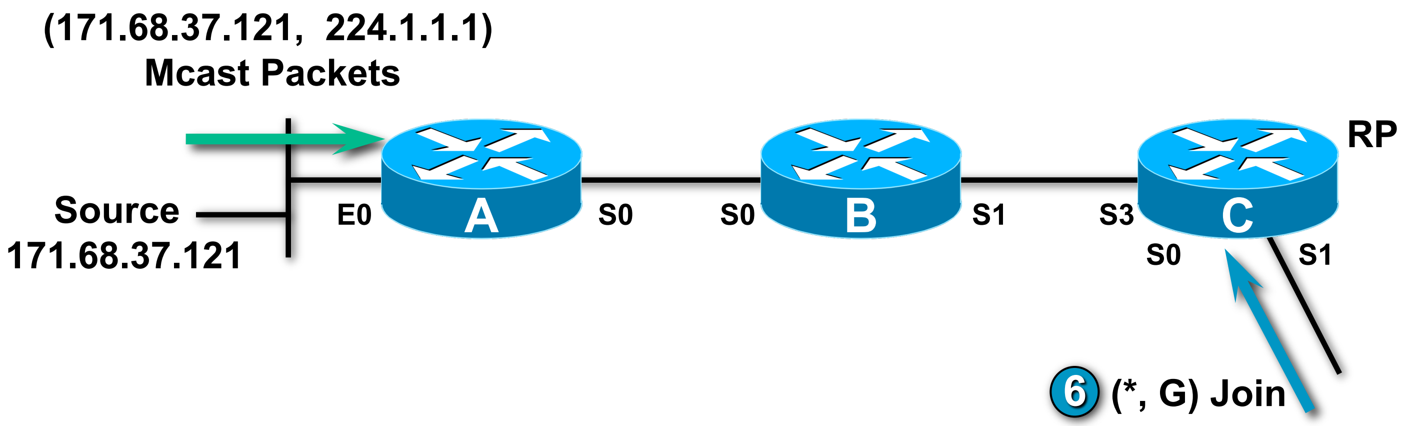

```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.140 Flags: C
  Incoming Interface List
    Decapstunnel0 Flags: A
```

State in RP After A Registers

(Without Receivers on Shared Tree)

PIM SM Registering

Source Registers First

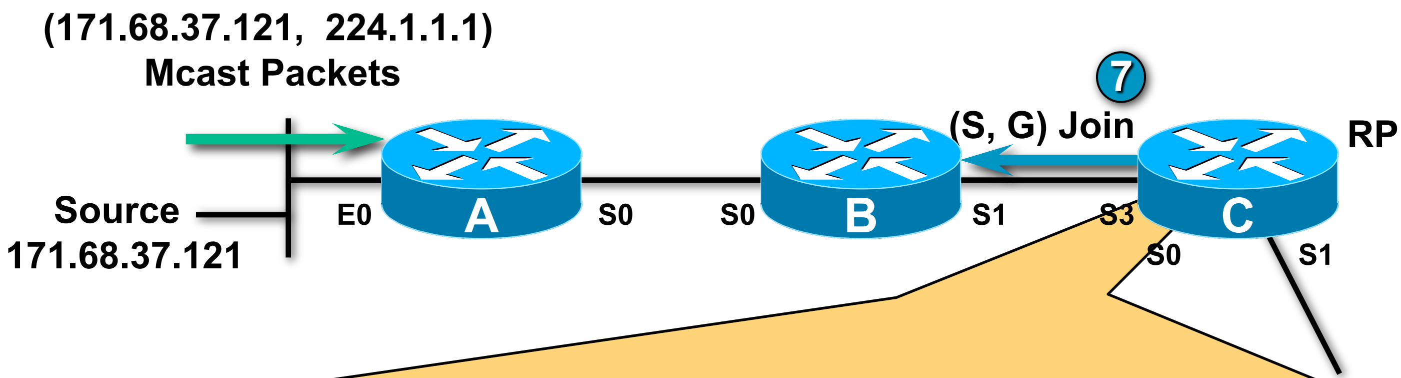


Receivers Begin Joining the Shared Tree

6 **RP (C) receives (*, G) Join from a receiver on Shared Tree.**

PIM SM Registering

Source Registers First




```
(*, 224.1.1.1), 00:09:21/00:00:00, RP 171.68.28.140, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list:
     Serial1, Forward/Sparse-Dense, 00:00:14/00:02:46

(171.68.37.121, 224.1.1.1, 00:01:15/00:02:46, flags: T
  Incoming interface: Serial3, RPF nbr 171.68.28.139,
  Outgoing interface list:
     Serial1, Forward/Sparse-Dense, 00:00:14/00:02:46
```

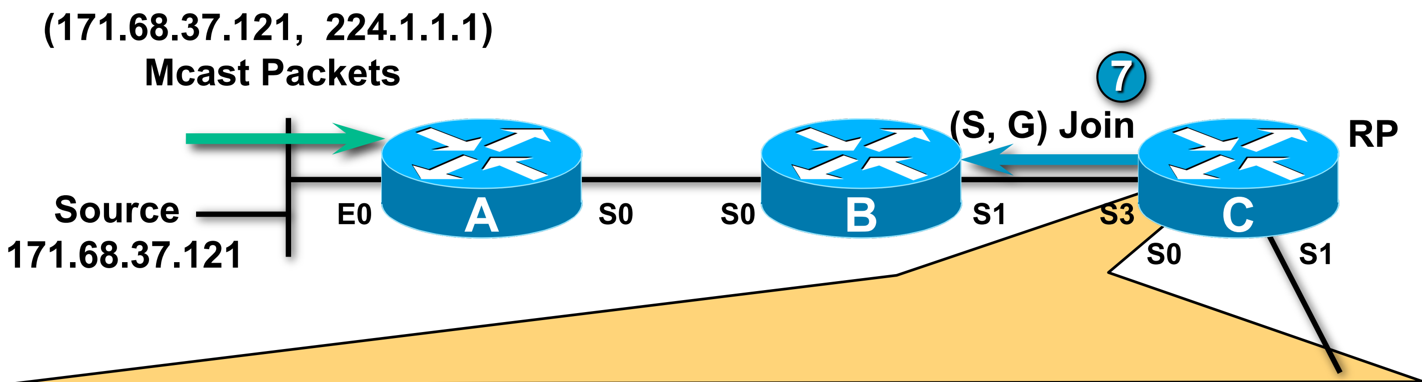
RP Processes (*,G) Join

(Adds Serial1 to Outgoing Interface Lists)

7 RP sends (S,G) Joins for all known Sources in Group.

PIM SM Registering (XR)

Source Registers First




```
(*,224.1.1.1) RPF nbr: 171.68.28.140 Flags: C
  Incoming Interface List
    Decapstunnel0 Flags: A NS, Up: 00:00:05
  Outgoing Interface List
    Serial1 Flags: F NS, Up: 00:00:05

(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.139 Flags:
  Incoming Interface List
    Serial3 Flags: A, Up: 00:00:05
  Outgoing Interface List
    Serial1 Flags: F NS, Up: 00:00:05
```

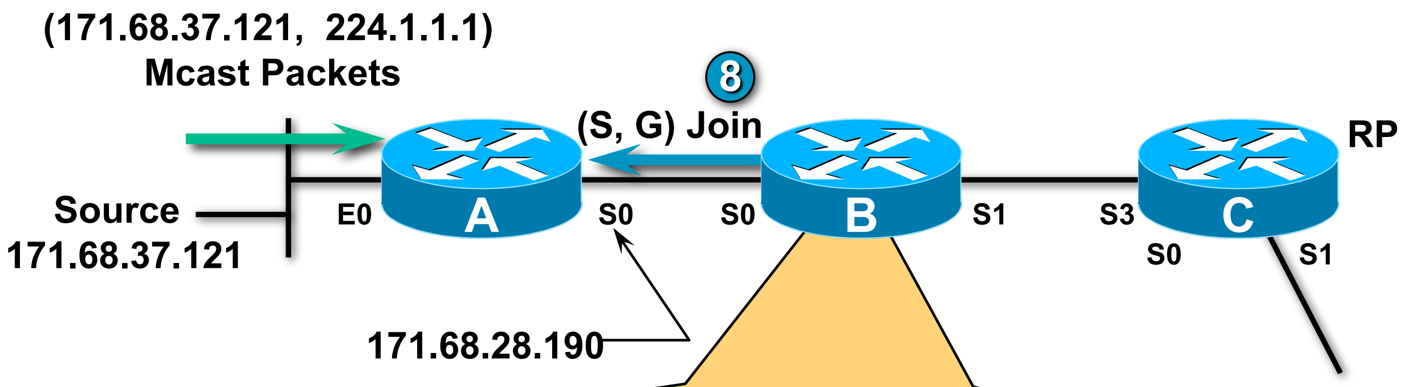
RP Processes (*,G) Join

(Adds (*,G) State and Serial1 to Outgoing Interface Lists)

7 RP sends (S,G) Joins for all known Sources in Group.

PIM SM Registering

Source Registers First




```
(*, 224.1.1.1), 00:04:28/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial1, RPF nbr 171.68.28.140,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:04:28/00:01:32, flags:
  Incoming interface: Serial0, RPF nbr 171.68.28.190
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:04:28/00:01:32
```

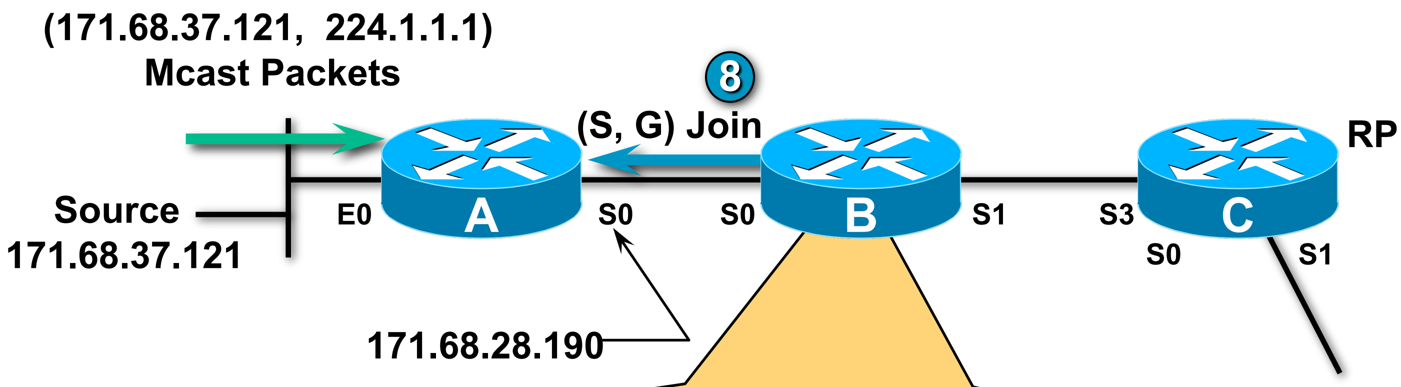
B Processes Join, Creates (S, G) State

(After Automatically Creating the (*, G) Entry)

8 **B sends (S,G) Join toward Source to continue building SPT.**

PIM SM Registering (XR)

Source Registers First




```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.190 Flags:
  Up: 00:00:46
  Incoming Interface List
    Serial0 Flags: A, Up: 00:00:46
  Outgoing Interface List
    Serial1 Flags: F NS, Up: 00:00:46
```

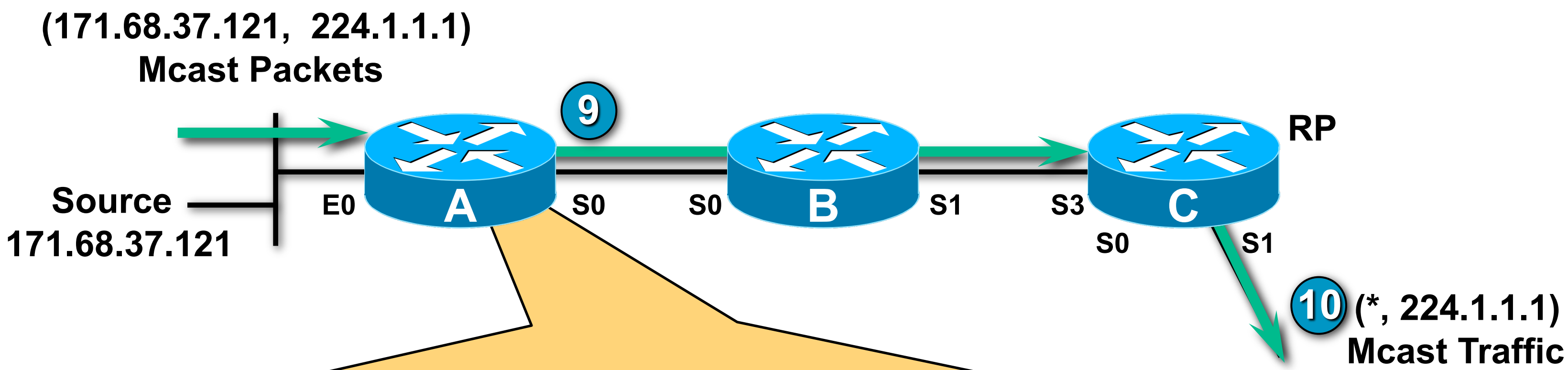
B Processes Join, Creates (S, G) State

(Without Automatically Creating the (*, G) Entry)

8 **B sends (S,G) Join toward Source to continue building SPT.**

PIM SM Registering

Source Registers First




```
(*, 224.1.1.1), 00:04:28/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial0, RPF nbr 171.68.28.191,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:04:28/00:01:32, flags: FT
  Incoming interface: Ethernet0, RPF nbr 0.0.0.0,
  Outgoing interface list:
    Serial0, Forward/Sparse-Dense, 00:04:28/00:01:32
```

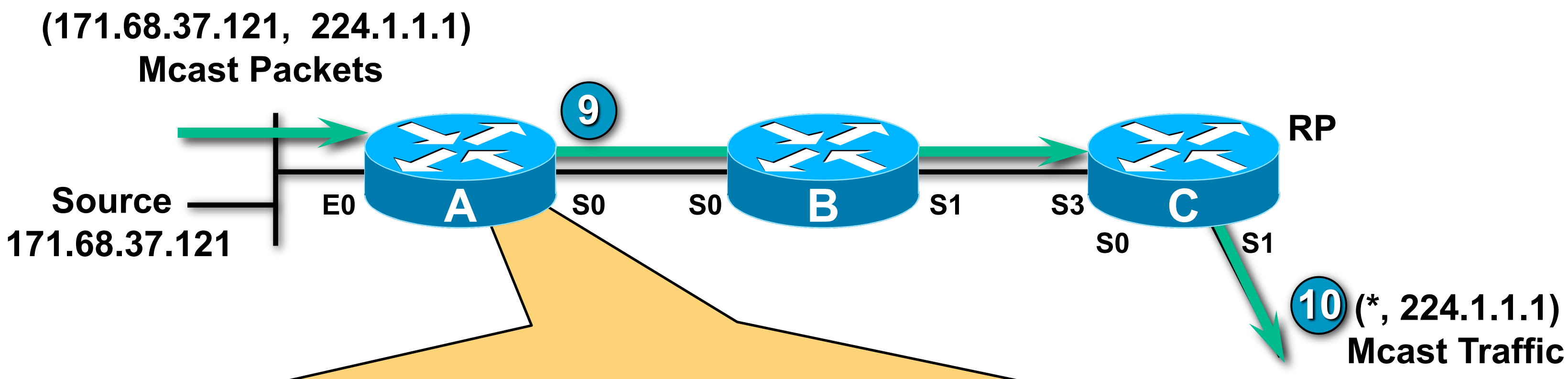
A Processes the (S, G) Join; Adds Serial0 to OIL

9 **RP begins receiving (S,G) traffic down SPT.**

10 **RP forwards (S,G) traffic down Shared Tree to receivers.**

PIM SM Registering (XR)

Source Registers First



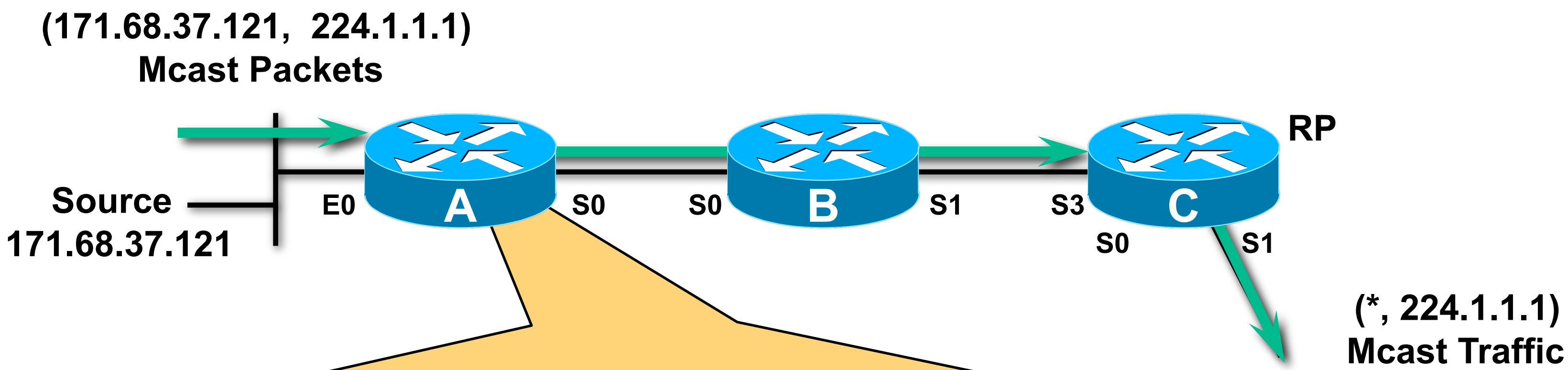

```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.37.121 Flags:
  Up: 00:00:46
  Incoming Interface List
    Ethernet0 Flags: A, Up: 00:00:46
  Outgoing Interface List
    Serial0 Flags: F NS, Up: 00:00:46
```

A Processes the (S, G) Join; Adds Serial0 to OIL

- (9) **RP begins receiving (S,G) traffic down SPT.**
- (10) **RP forwards (S,G) traffic down Shared Tree to receivers.**

PIM SM Registering

Source Registers First

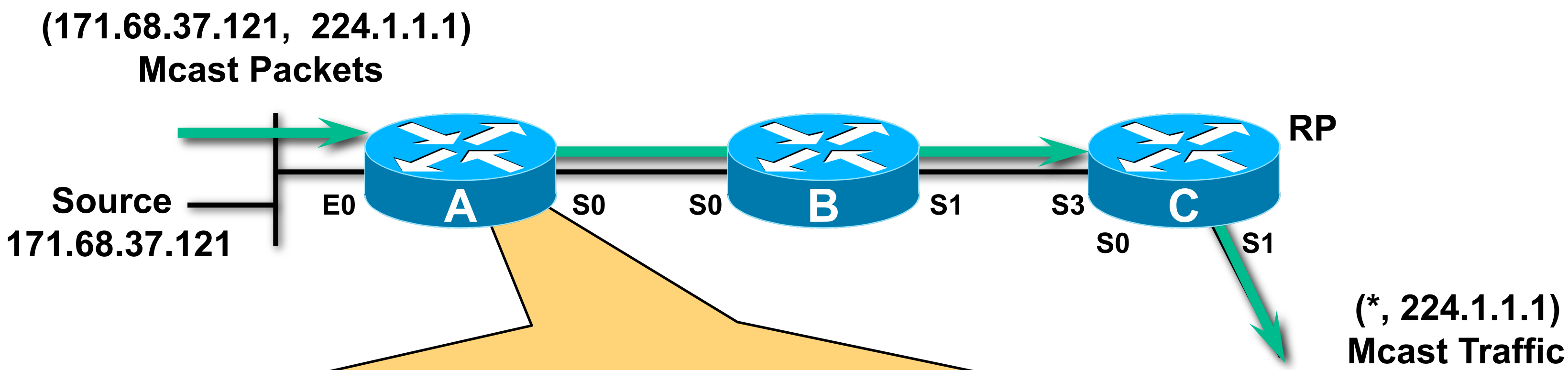



```
(*, 224.1.1.1), 00:04:28/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial0, RPF nbr 171.68.28.191,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:04:28/00:01:32, flags: FT
  Incoming interface: Ethernet0, RPF nbr 0.0.0.0,
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:04:28/00:01:32
```

Final State in Router A (IOS)

PIM SM Registering (XR)

Source Registers First

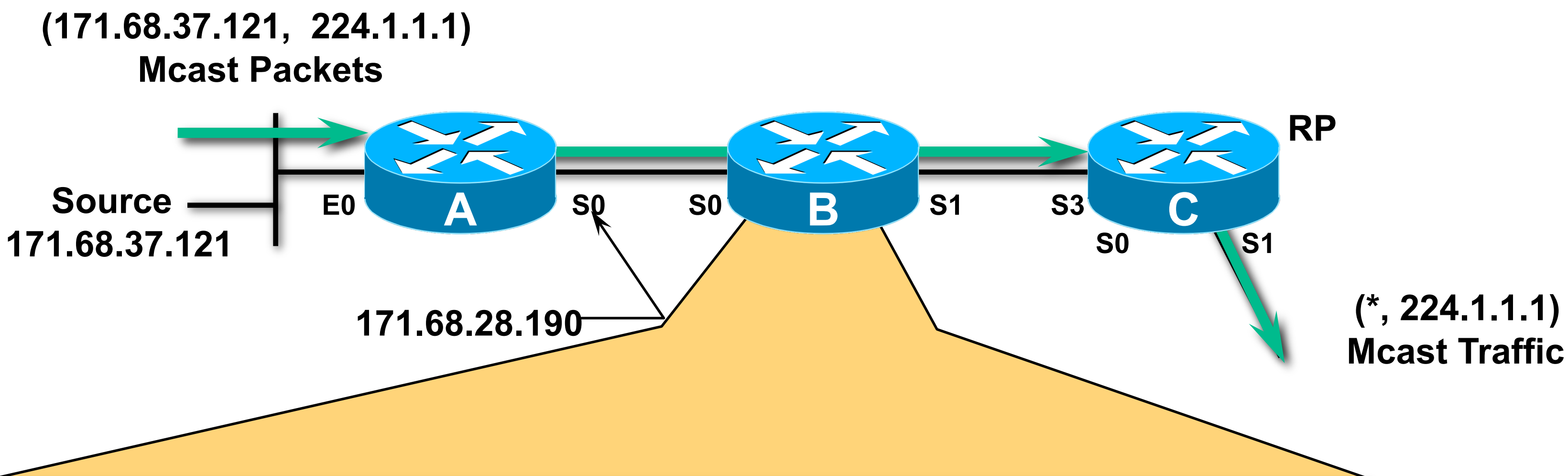



```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.37.121 Flags:
  Incoming Interface List
    Ethernet0 Flags: A
  Outgoing Interface List
    Serial0 Flags: F NS
```

Final State in Router A (IOS XR)

PIM SM Registering

Source Registers First

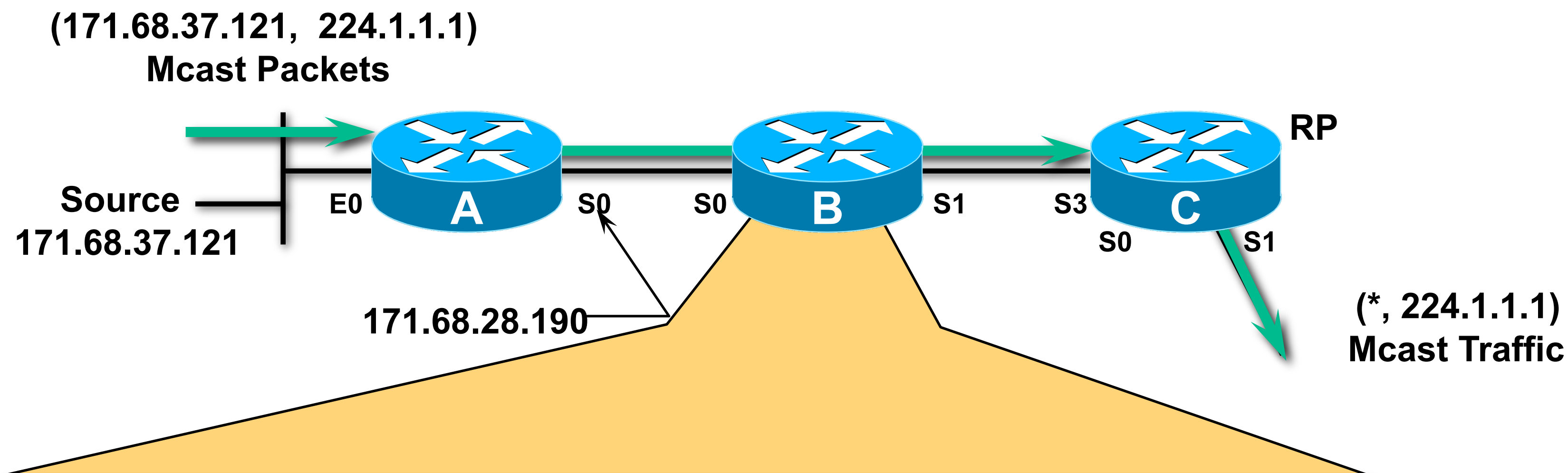



```
(*, 224.1.1.1), 00:04:28/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial1, RPF nbr 171.68.28.140,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:04:28/00:01:32, flags: T
  Incoming interface: Serial0, RPF nbr 171.68.28.190
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:04:28/00:01:32
```

Final State in B After Receivers Join

PIM SM Registering (XR)

Source Registers First

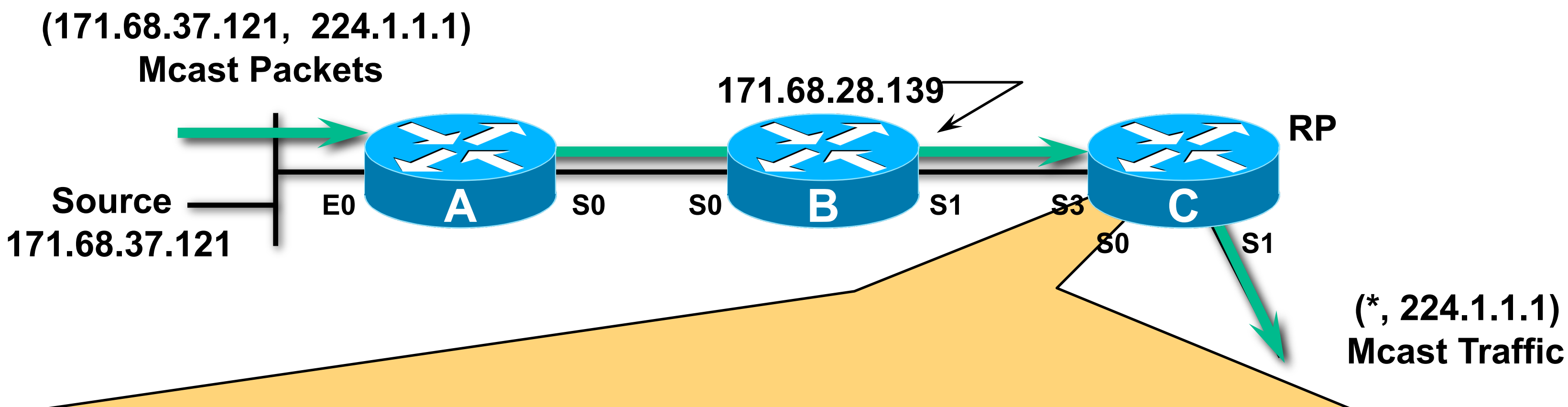



```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.190 Flags:
  Incoming Interface List
    Serial0 Flags: A
  Outgoing Interface List
    Serial1 Flags: F NS
```

Final State in B After Receivers Join (IOS XR)

PIM SM Registering

Source Registers First

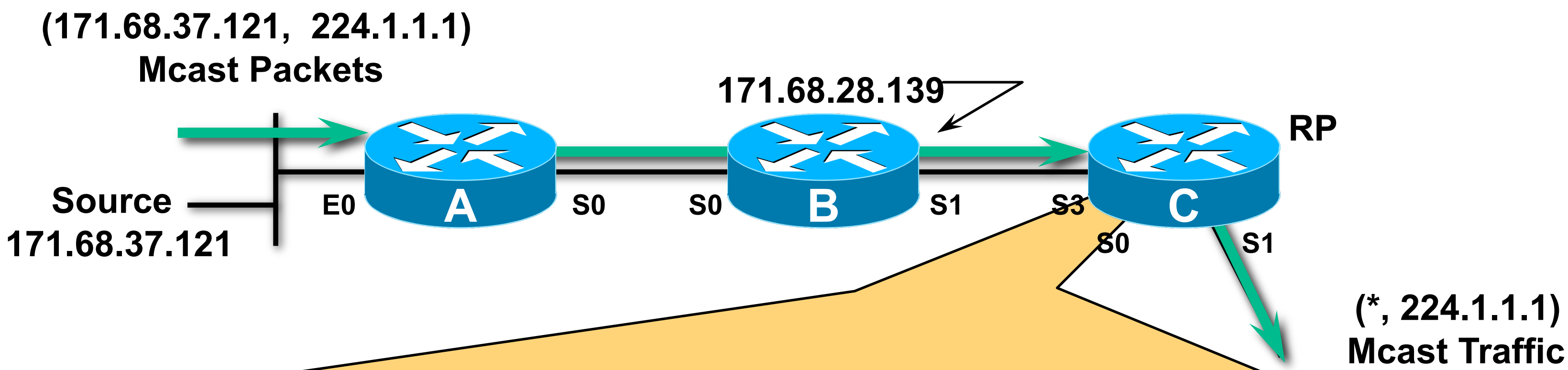



```
(*, 224.1.1.1), 00:09:21/00:00:00, RP 171.68.28.140, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:03:14/00:02:46

(171.68.37.121, 224.1.1.1, 00:01:15/00:02:46, flags: T
  Incoming interface: Serial3, RPF nbr 171.68.28.139,
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:00:49/00:02:11
```

Final State in RP After Receivers Join (IOS)

PIM SM Registering (XR)

Source Registers First



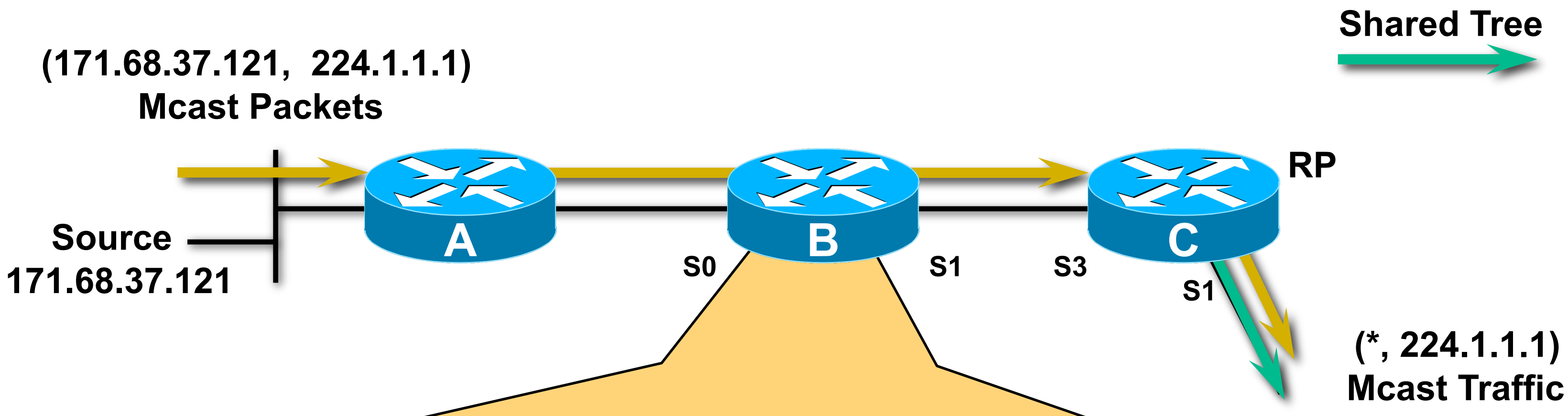

```
(*,224.1.1.1) RPF nbr: 171.68.28.140 Flags: C
  Incoming Interface List
    Decapstunnel0 Flags: A
  Outgoing Interface List
    Serial1 Flags: F NS

(171.68.37.121,224.1.1.1) RPF nbr 171.68.28.139 Flags:
  Incoming Interface List
    Serial3 Flags: A
  Outgoing Interface List
    Serial1 Flags: F NS
```

Final State in RP After Receivers Join (IOS XR)

PIM SM Registering: Receiver Along the SPT

PIM SM Registering

Receivers Along the SPT

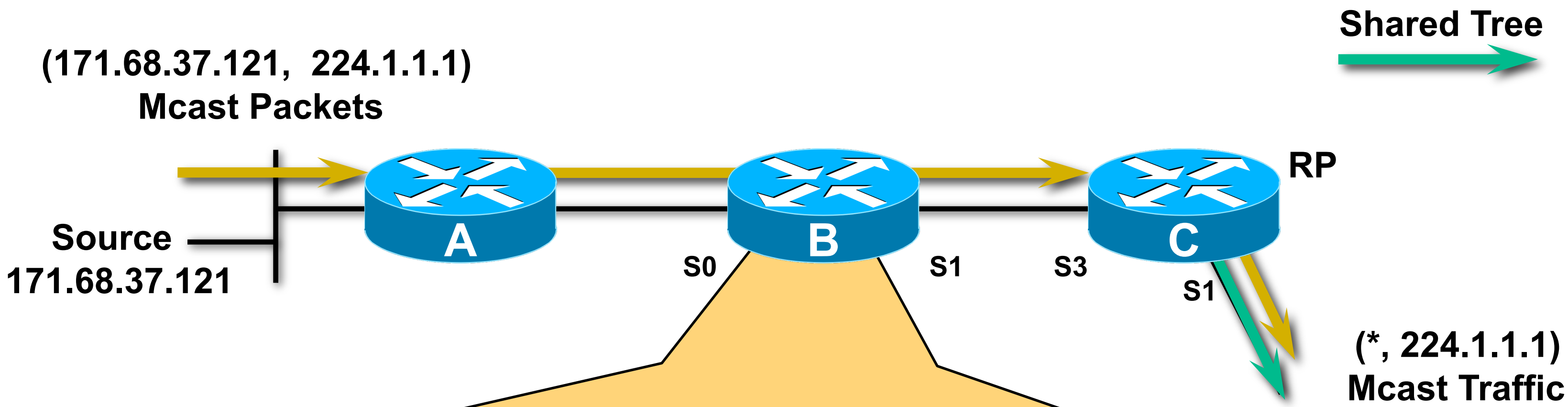



```
(*, 224.1.1.1), 00:04:28/00:00:00, RP 171.68.28.140, flags: SP
  Incoming interface: Serial1, RPF nbr 171.68.28.140,
  Outgoing interface list: Null

(171.68.37.121, 224.1.1.1), 00:04:28/00:01:32, flags: T
  Incoming interface: Serial0, RPF nbr 171.68.28.190
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:04:28/00:01:32
```

Current State in B

PIM SM Registering (XR)

Receivers Along the SPT

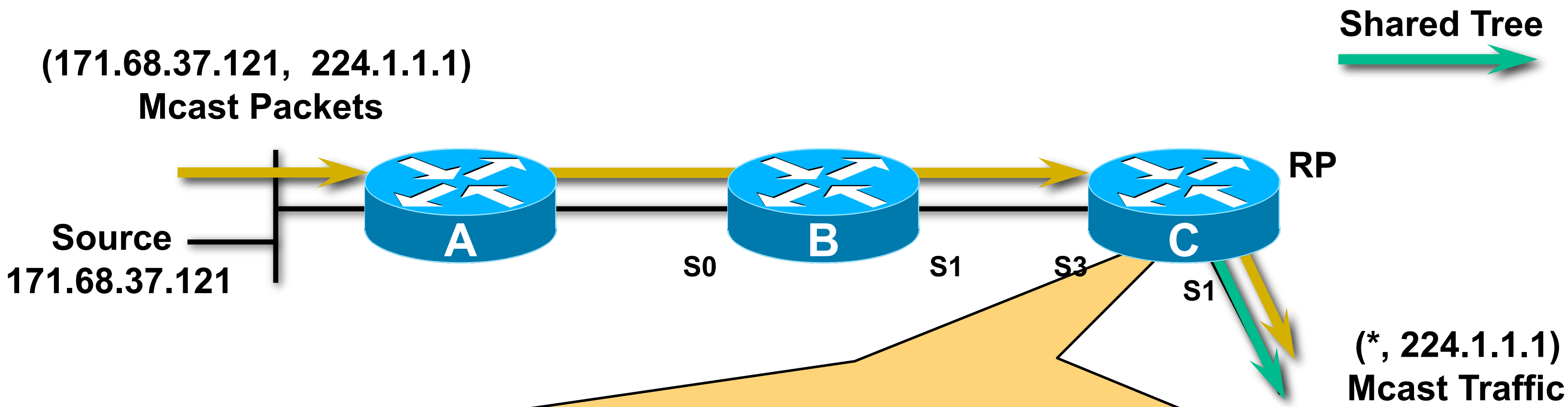



```
(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.190 Flags:
  Incoming Interface List
    Serial0 Flags: A
  Outgoing Interface List
    Serial1 Flags: F NS
```

Current State in B

PIM SM Registering

Receivers Along the SPT

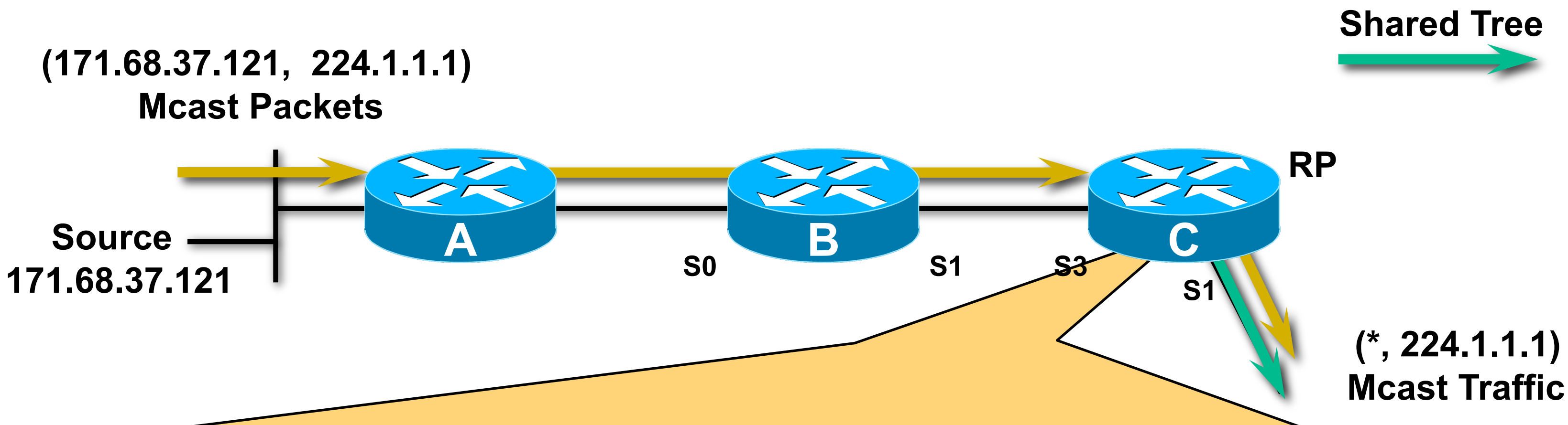



```
(*, 224.1.1.1), 00:09:21/00:00:00, RP 171.68.28.140, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:03:14/00:02:46

(171.68.37.121, 224.1.1.1, 00:01:15/00:02:46, flags: T
  Incoming interface: Serial3, RPF nbr 171.68.28.139,
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:00:49/00:02:11
```

Current State in the RP

PIM SM Registering (XR)

Receivers Along the SPT

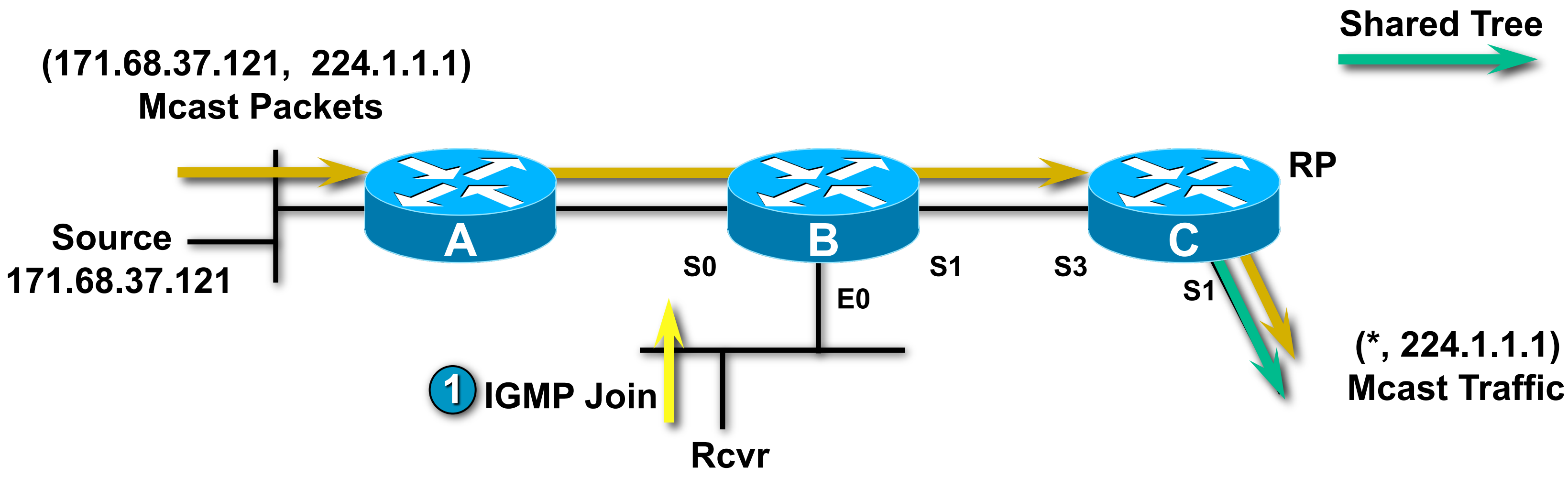



```
(*,224.1.1.1) RPF nbr: 171.68.28.140 Flags: C
  Incoming Interface List
    Decapstunnel0 Flags: A
  Outgoing Interface List
    Serial1 Flags: F NS

(171.68.37.121,224.1.1.1) RPF nbr 171.68.28.139 Flags:
  Incoming Interface List
    Serial3 Flags: A
  Outgoing Interface List
    Serial1 Flags: F NS
```

Current State in the RP

PIM SM Registering

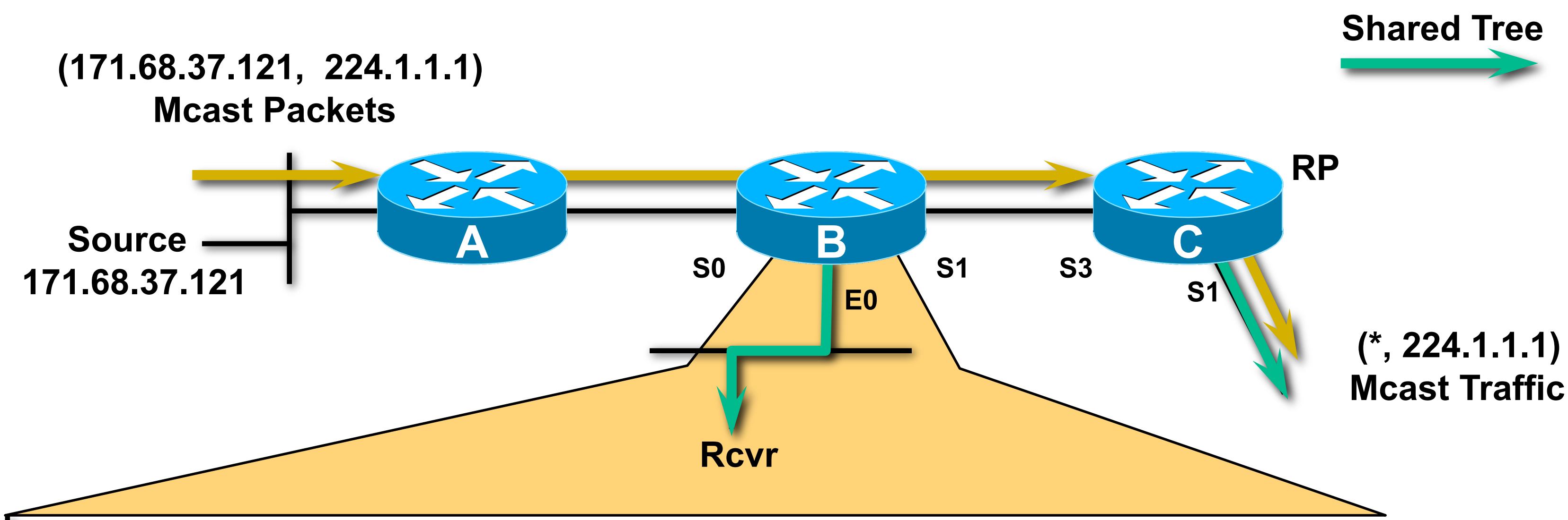
Receivers Along the SPT



1 Rcvr wishes to receive group G traffic. Sends IGMP Join for G.

PIM SM Registering

Receivers Along the SPT

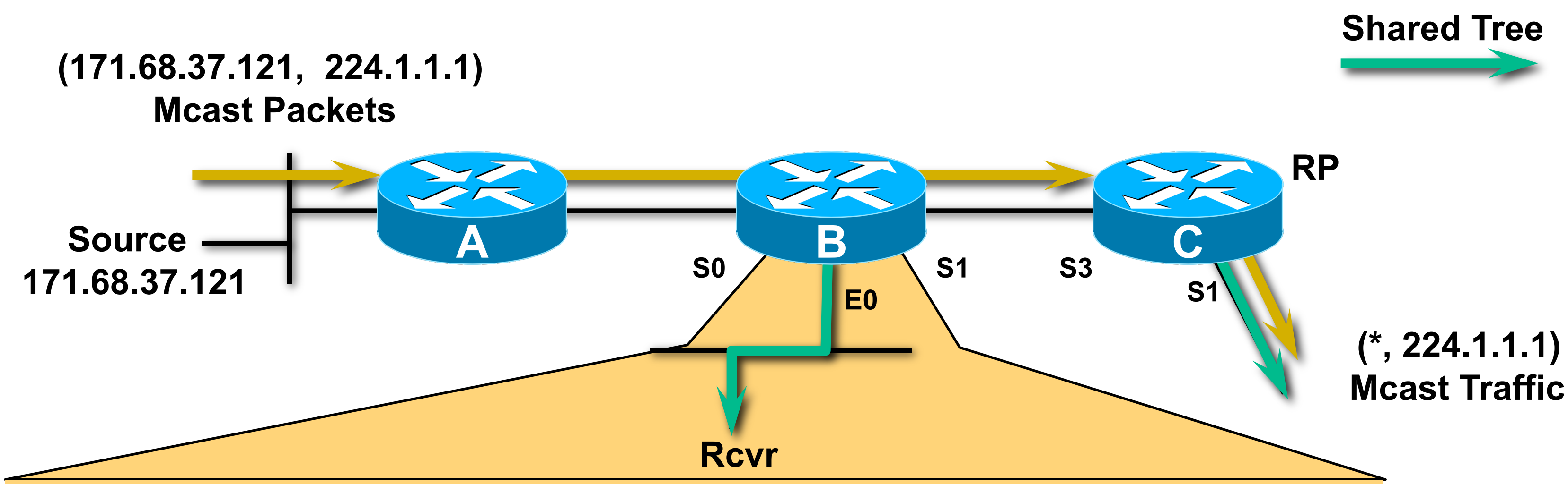



```
(*, 224.1.1.1), 00:04:28/00:00:00, RP 171.68.28.140, flags: SC
  Incoming interface: Serial1, RPF nbr 171.68.28.140,
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:00:30/00:02:30

(171.68.37.121, 224.1.1.1), 00:04:28/00:01:32, flags: CT
  Incoming interface: Serial0, RPF nbr 171.68.28.190
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:04:28/00:01:32
    Ethernet0, Forward/Sparse-Dense, 00:00:30/00:02:30
```

State in B After Rcvr Joins Group

PIM SM Registering (XR)

Receivers Along the SPT

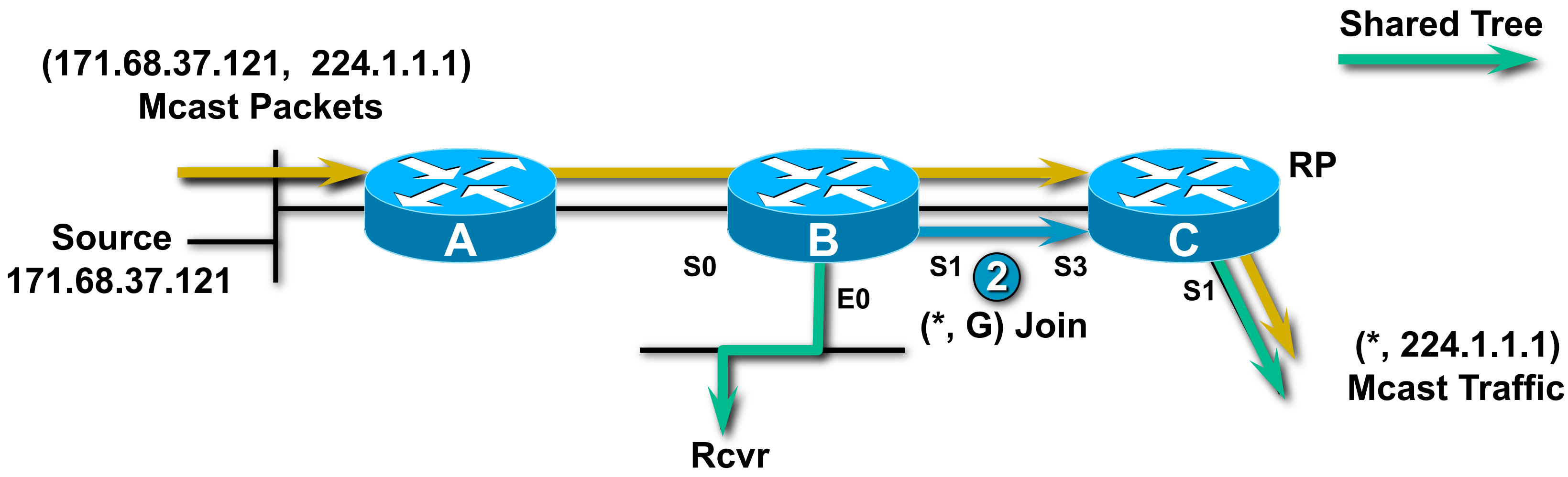



```
(*,224.1.1.1) RPF nbr: 171.68.28.140 Flags: C
  Incoming Interface List
    Serial1 Flags: A NS
  Outgoing Interface List
    Ethernet0 Flags: F NS LI

(171.68.37.121,224.1.1.1) RPF nbr: 171.68.28.190 Flags:
  Incoming Interface List
    Serial0 Flags: A
  Outgoing Interface List
    Serial1 Flags: F NS
    Ethernet0 Flags: F NS LI
```

State in B After Rcvr Joins Group

PIM SM Registering

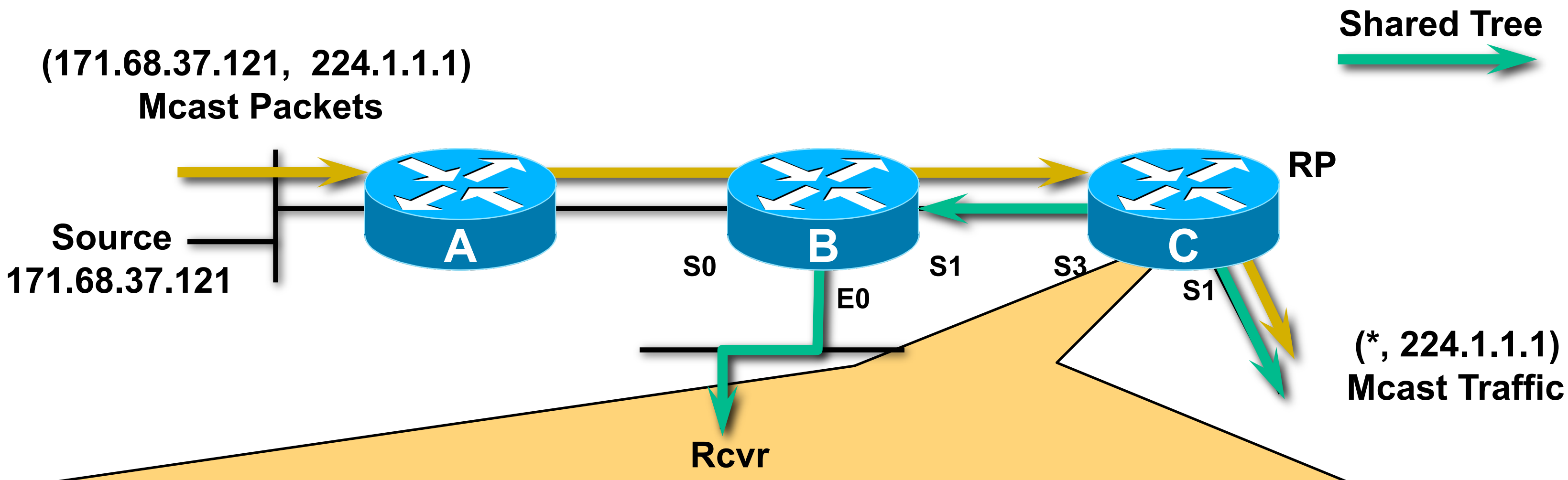
Receivers Along the SPT



② B triggers a (*,G) Join to join the Shared Tree

PIM SM Registering

Receivers Along the SPT

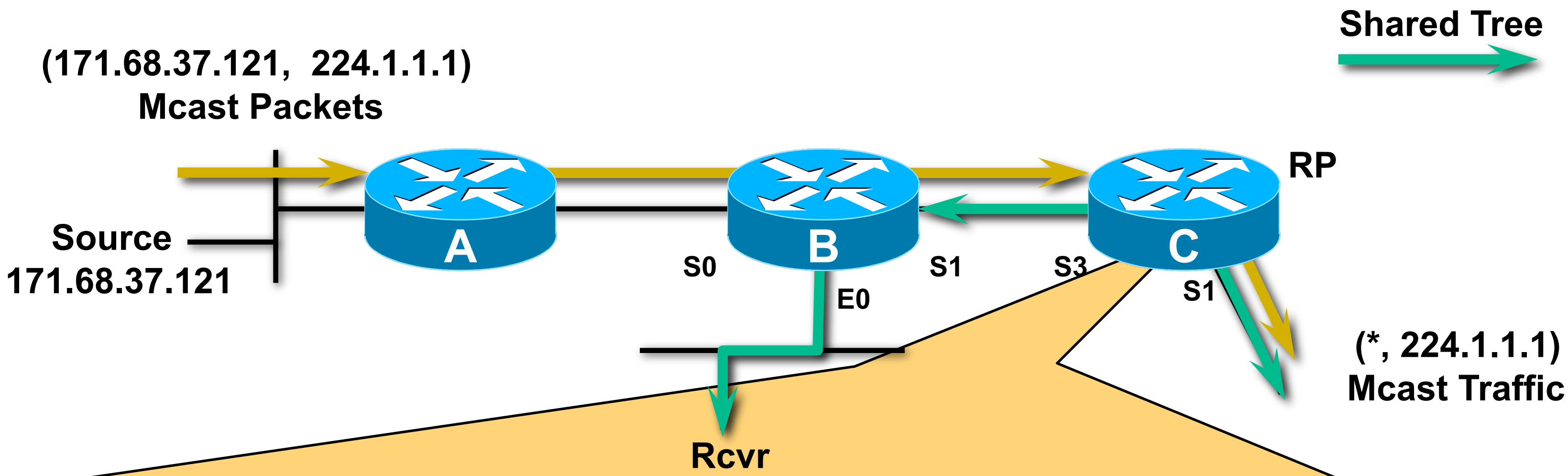



```
(*, 224.1.1.1), 00:09:21/00:00:00, RP 171.68.28.140, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0,
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:03:14/00:02:46
    Serial3, Forward/Sparse-Dense, 00:00:10/00:02:50

(171.68.37.121, 224.1.1.1, 00:01:15/00:02:46, flags: T
  Incoming interface: Serial3, RPF nbr 171.68.28.139,
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:00:49/00:02:11
```

State in RP After B Joins Shared Tree

PIM SM Registering (XR)

Receivers Along the SPT

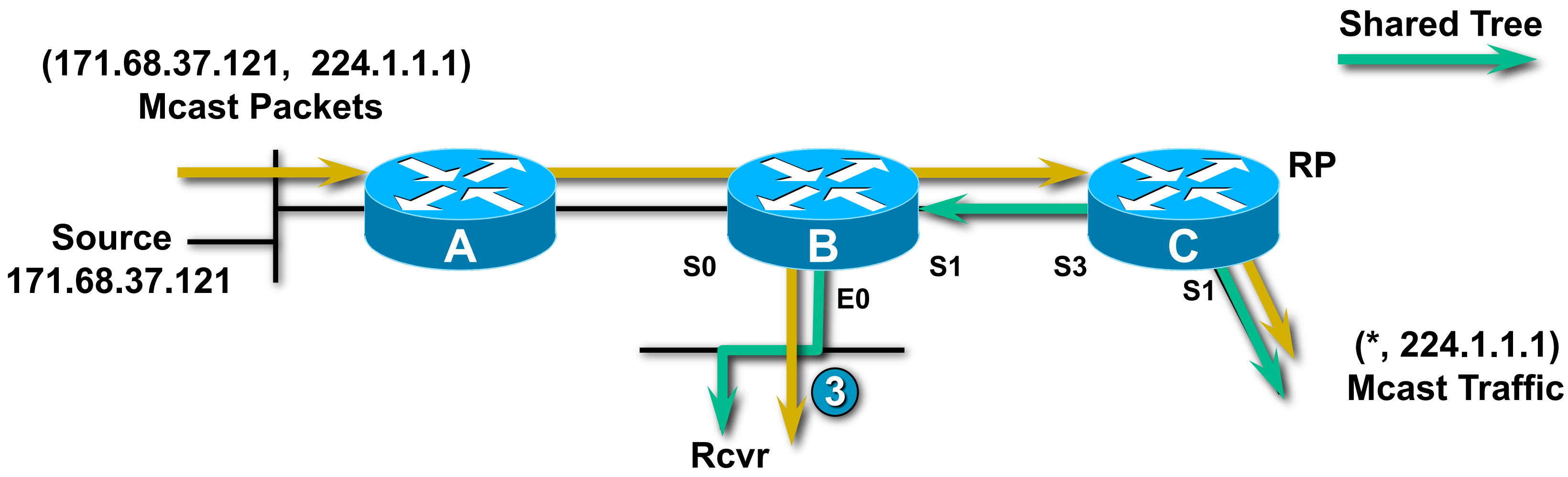



```
(*,224.1.1.1) RPF nbr: 171.68.28.140 Flags: C
  Incoming Interface List
    Decapstunnel0 Flags: A
  Outgoing Interface List
    Serial1 Flags: F NS
    Serial3 Flags: F NS

(171.68.37.121,224.1.1.1) RPF nbr 171.68.28.139 Flags: L
  Incoming Interface List
    Serial3 Flags: A
  Outgoing Interface List
    Serial1 Flags: F NS
```

State in RP After B Joins Shared Tree

PIM SM Registering

Receivers Along the SPT



3 **Group G traffic begins to flow to Rcvr.**

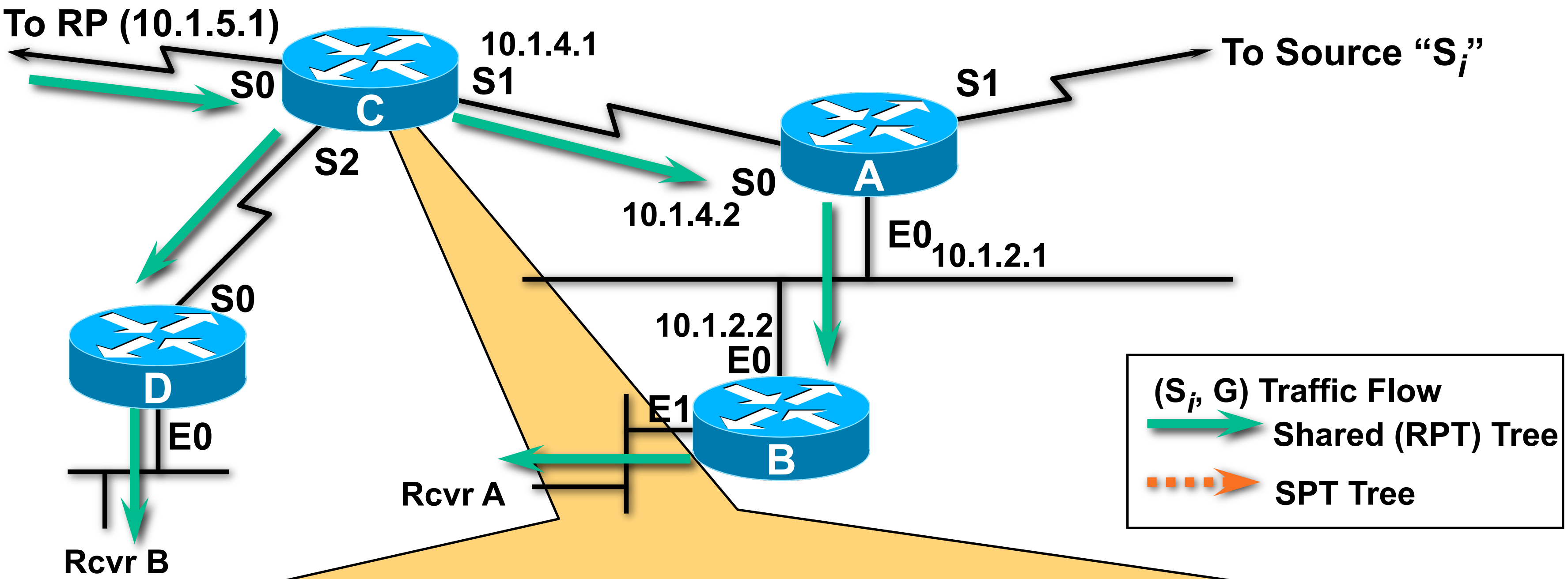
(Note: 171.68.37.121 traffic doesn't flow to RP then back down to B)

PIM SM SPT-Switchover

PIM SM SPT-Switchover

- 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”
  - **Don’t use values in between “0” and “infinity”**
    - **(In IOS XR, “0” and “infinity” are the only options)**
- Threshold triggers Join of Source Tree
  - Sends an (S,G) Join up SPT for next “S” in “G” packet received

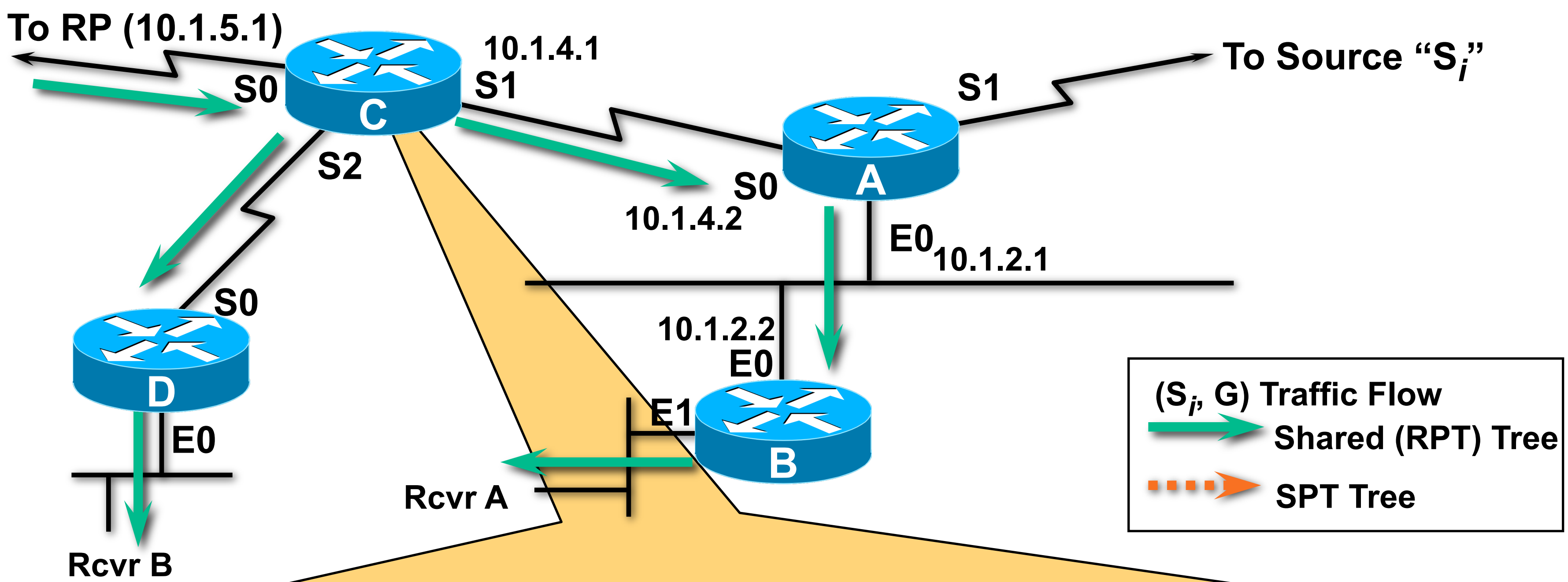
PIM SM SPT-Switchover




```
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: S
  Incoming interface: Serial0, RPF nbr 10.1.5.1,
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:01:43/00:02:11
    Serial2, Forward/Sparse-Dense, 00:00:32/00:02:28
```

State in C Before Switch

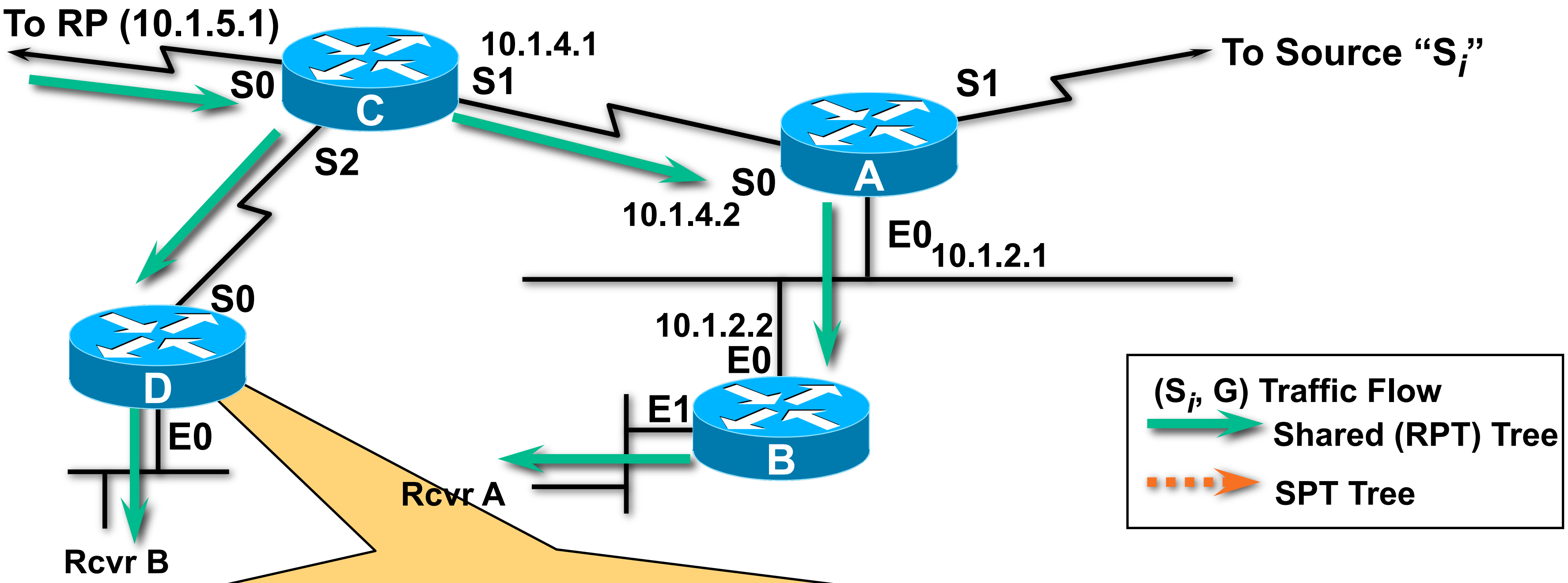
PIM SM SPT-Switchover (XR)




```
(*,224.1.1.1) RPF nbr: 10.1.5.1 Flags: C
  Up: 07:20:03
  Incoming Interface List
    Serial0 Flags: A, Up: 07:20:03
  Outgoing Interface List
    Serial1 Flags: F NS, Up: 07:20:03
    Serial1 Flags: F NS, Up: 07:20:03
```

State in C Before Switch

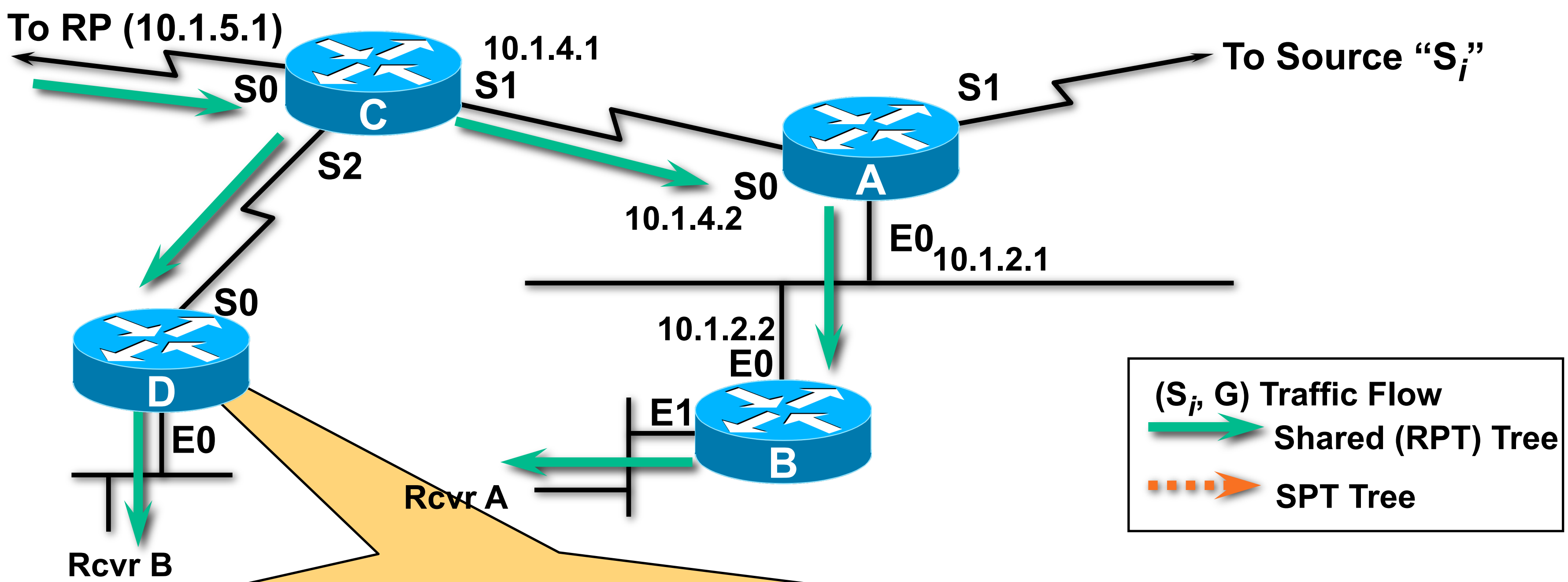
PIM SM SPT-Switchover




```
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: SC
  Incoming interface: Serial0, RPF nbr 10.1.4.9,
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:01:43/00:02:11
```

State in D Before Switch

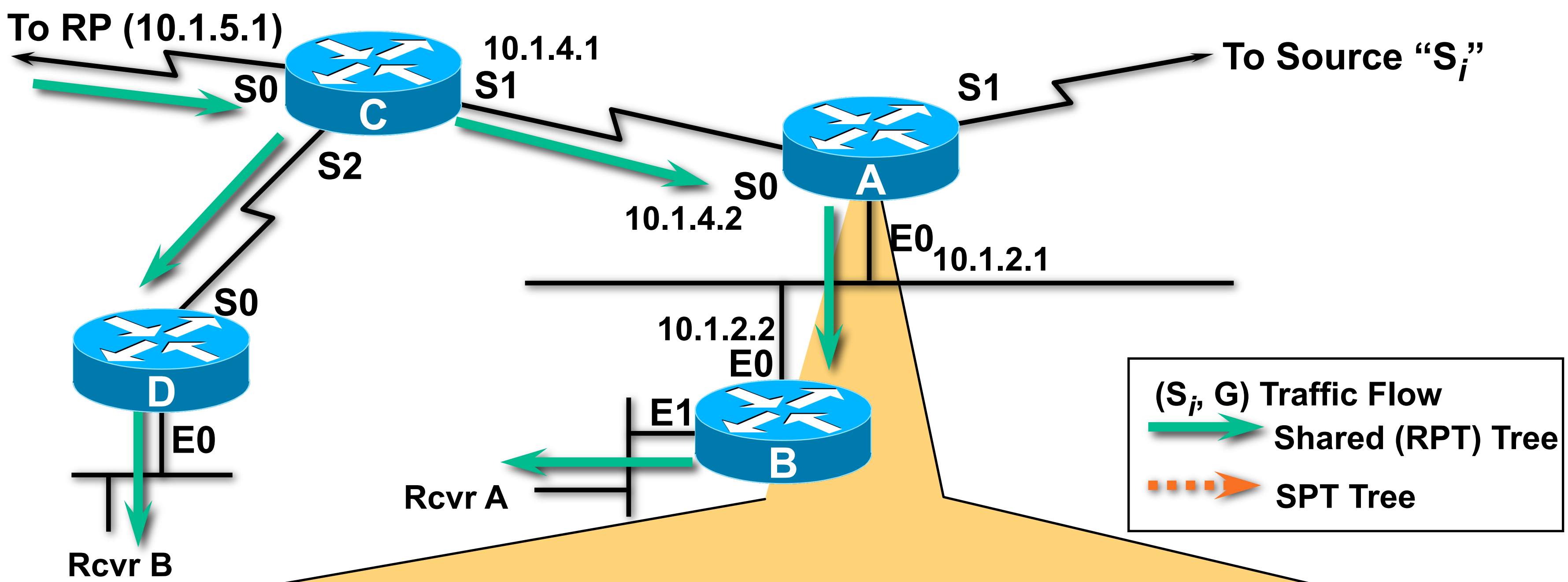
PIM SM SPT-Switchover (XR)




```
(*,224.1.1.1) RPF nbr: 10.1.4.9 Flags: C
  Up: 07:52:21
  Incoming Interface List
    Serial0 Flags: A, Up: 07:34:33
  Outgoing Interface List
    Ethernet0 Flags: F NS LI, Up: 07:52:21
```

State in D Before Switch

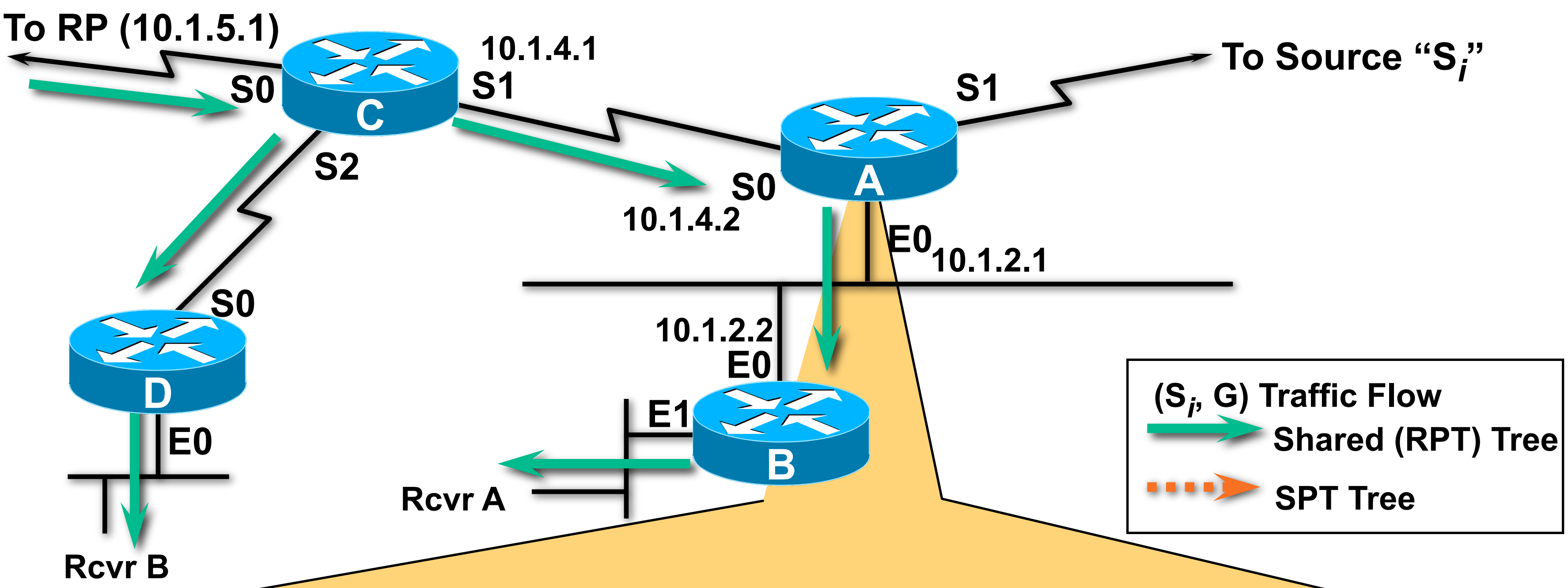
PIM SM SPT-Switchover




```
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: S
  Incoming interface: Serial0, RPF nbr 10.1.4.1,
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:01:43/00:02:11
```

State in A Before Switch

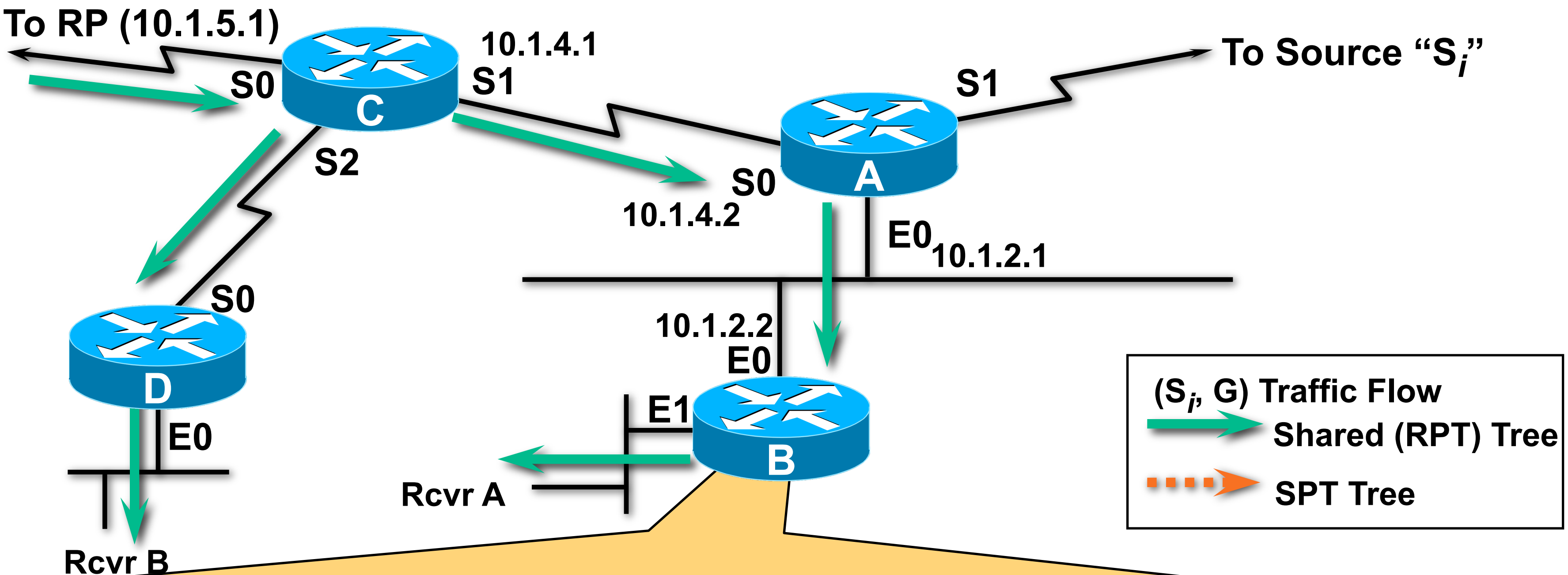
PIM SM SPT-Switchover (XR)




```
(*,224.1.1.1) RPF nbr: 10.1.4.1 Flags: C
  Up: 07:20:03
  Incoming Interface List
    Serial0 Flags: A, Up: 07:20:03
  Outgoing Interface List
    Ethernet0 Flags: F NS, Up: 07:20:03
```

State in A Before Switch

PIM SM SPT-Switchover

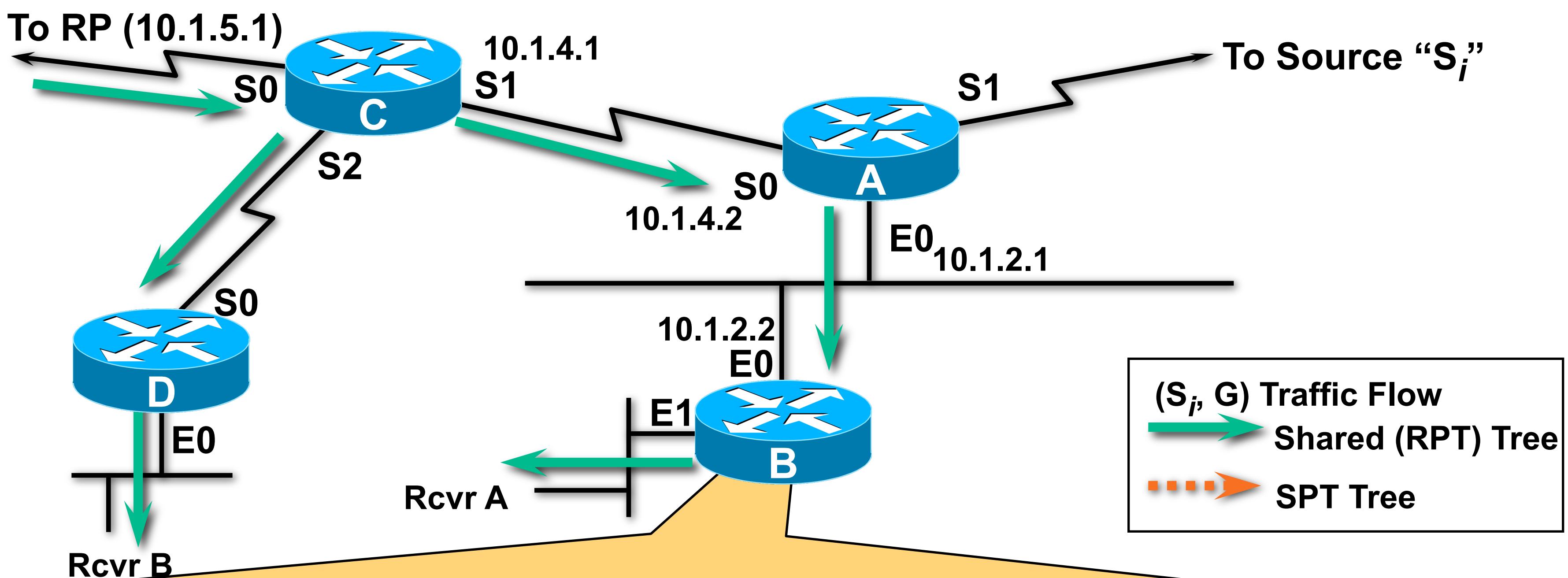



```
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: SCJ
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:01:43/00:02:11
```

Note "J" Flag is set

State in B Before Switch

PIM SM SPT-Switchover (XR)

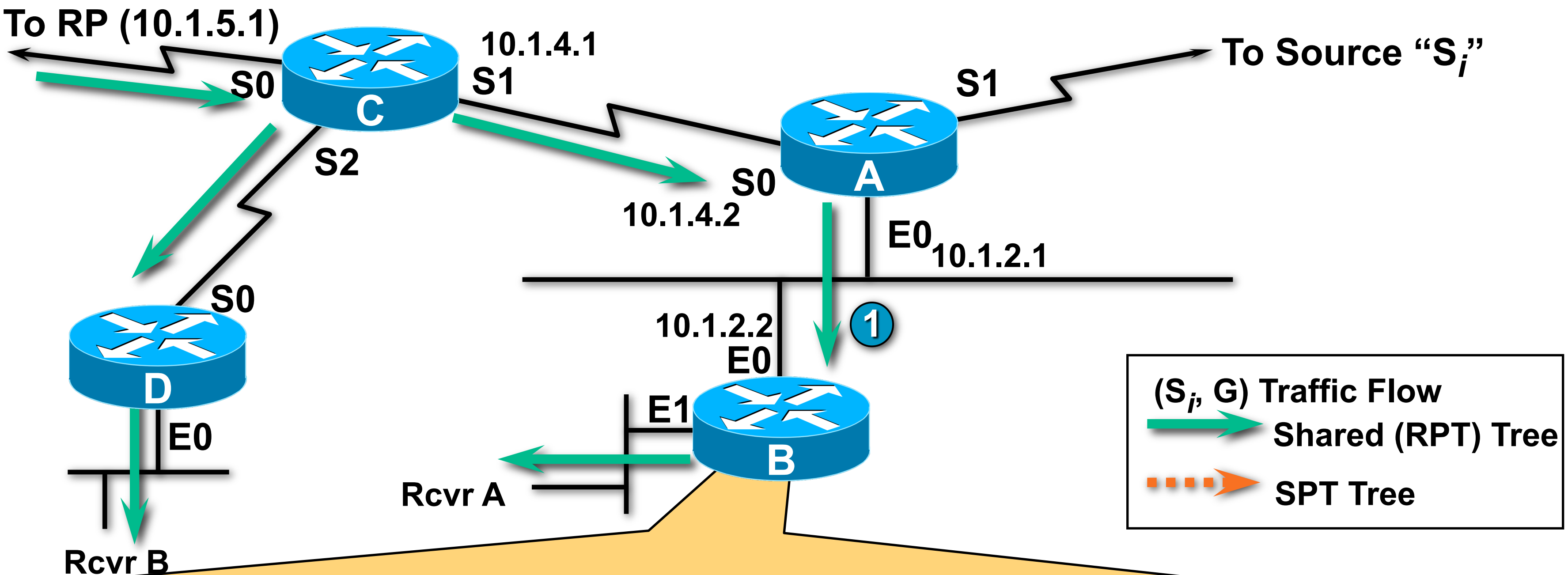



```
(*,224.1.1.1) RPF nbr: 10.1.2.1 Flags: C
  Up: 08:35:34
  Incoming Interface List
    Ethernet0 Flags: A NS, Up: 08:17:46
  Outgoing Interface List
    Ethernet1 Flags: F NS LI, Up: 08:35:34
```

Note "NS" Flag is set

State in B Before Switch

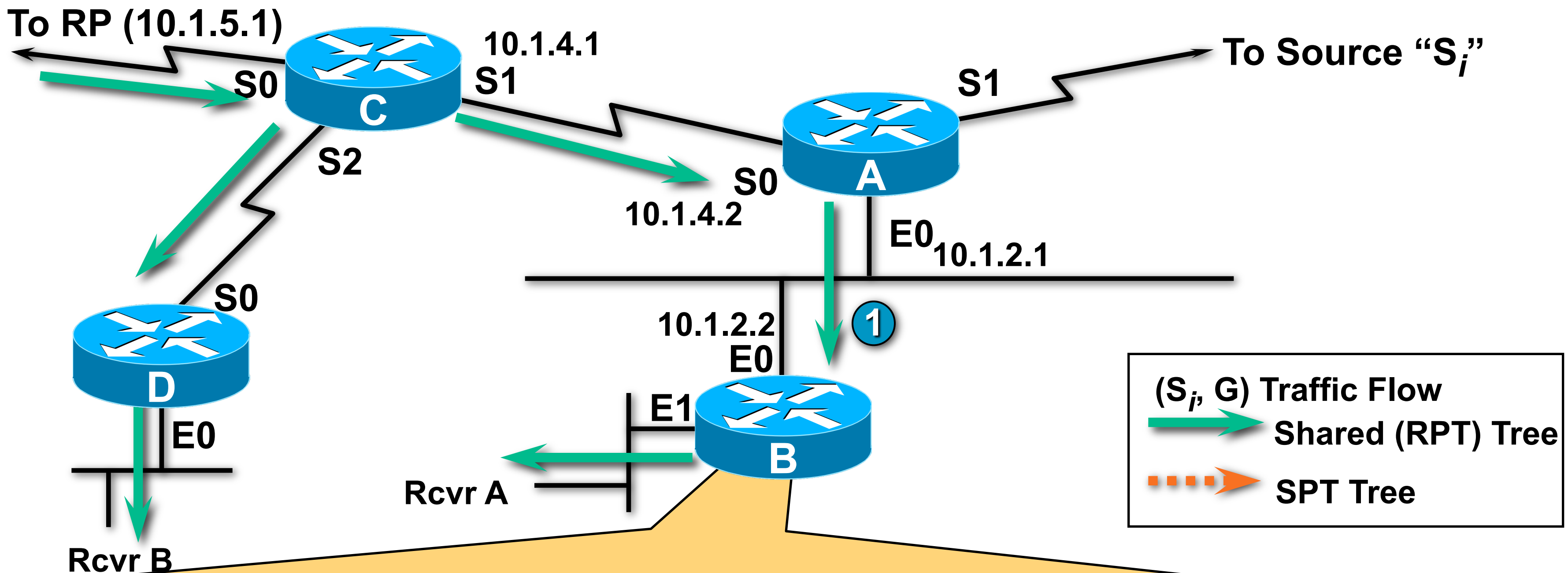
PIM SM SPT-Switchover




```
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: SCJ
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:01:43/00:02:11
```

① **New source (S_i,G) packet arrives down Shared tree.**

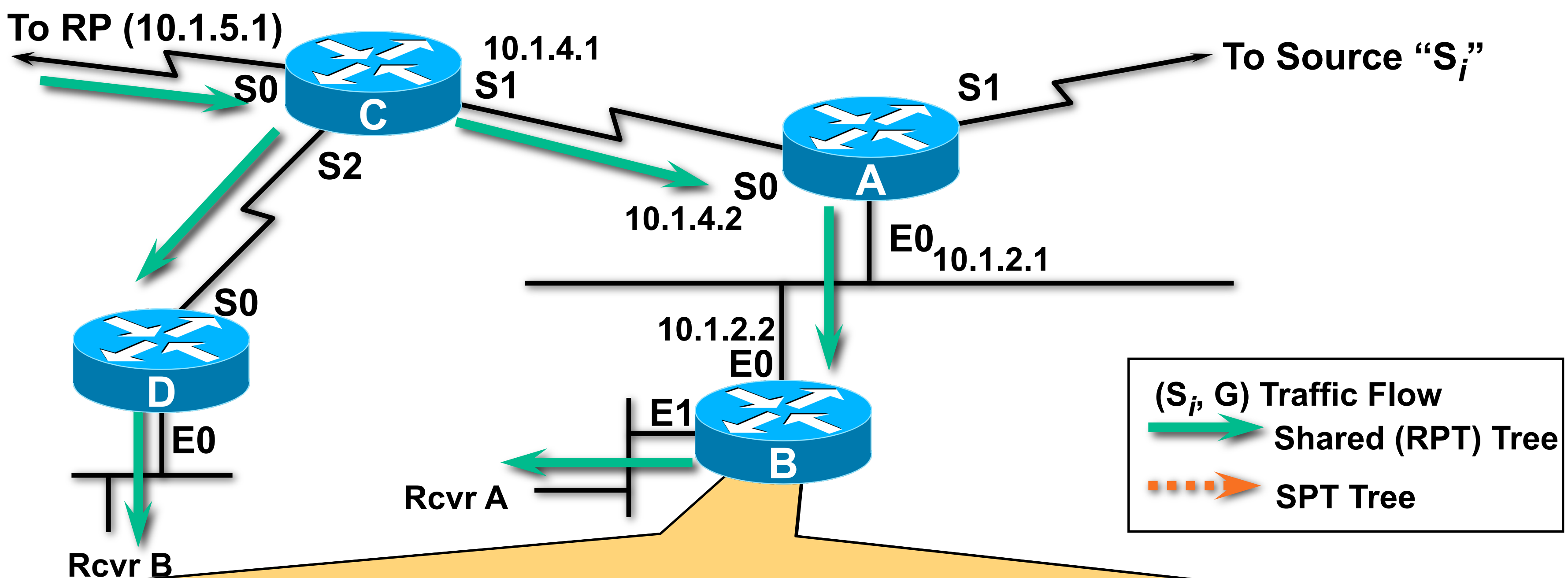
PIM SM SPT-Switchover (XR)




```
(*,224.1.1.1) RPF nbr: 10.1.2.1 Flags: C
  Up: 08:35:34
  Incoming Interface List
    Ethernet0 Flags: A NS, Up: 08:17:46
  Outgoing Interface List
    Ethernet1 Flags: F NS LI, Up: 08:35:34
```

1 **New source (S_i,G) packet arrives down Shared tree.**

PIM SM SPT-Switchover

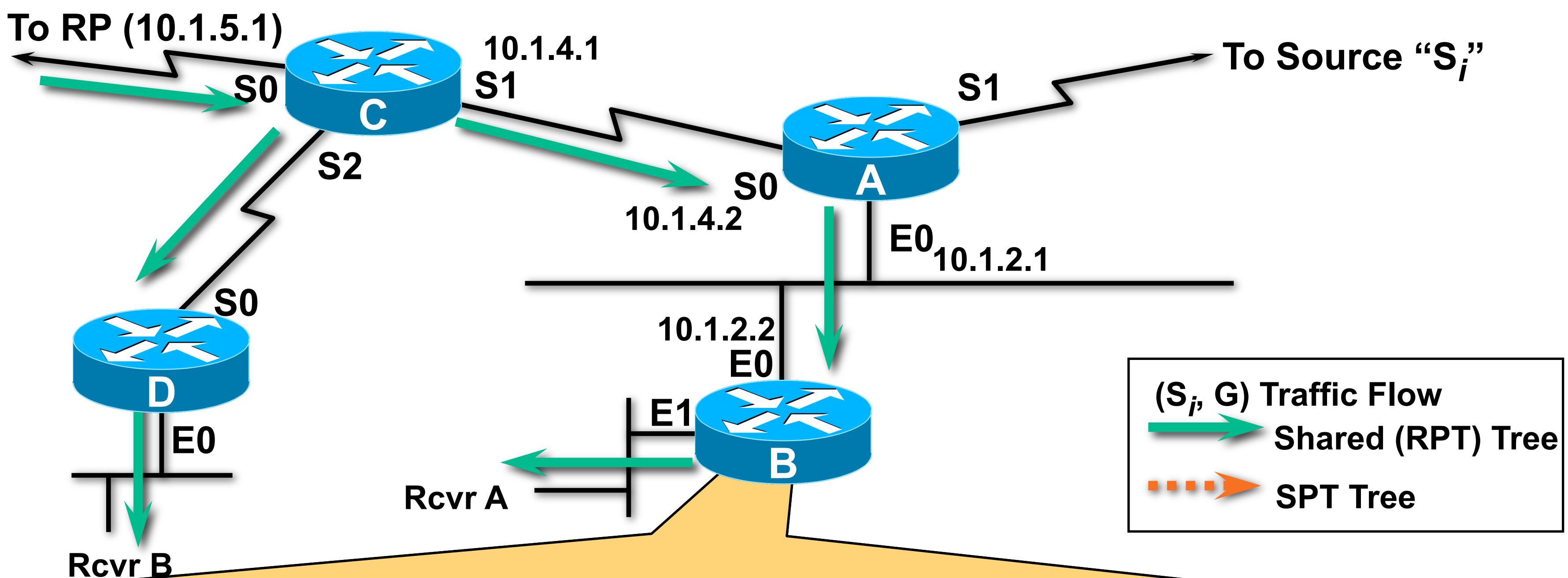



```
(*, 224.1.1.1), 00:01:43/00:00:00, RP 10.1.5.1, flags: SCJ
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:01:43/00:02:11

(171.68.37.121, 224.1.1.1), 00:00:28/00:02:51, flags: CJ
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:00:28/00:02:32
```

← 2

2 B creates (S_i,G) state.

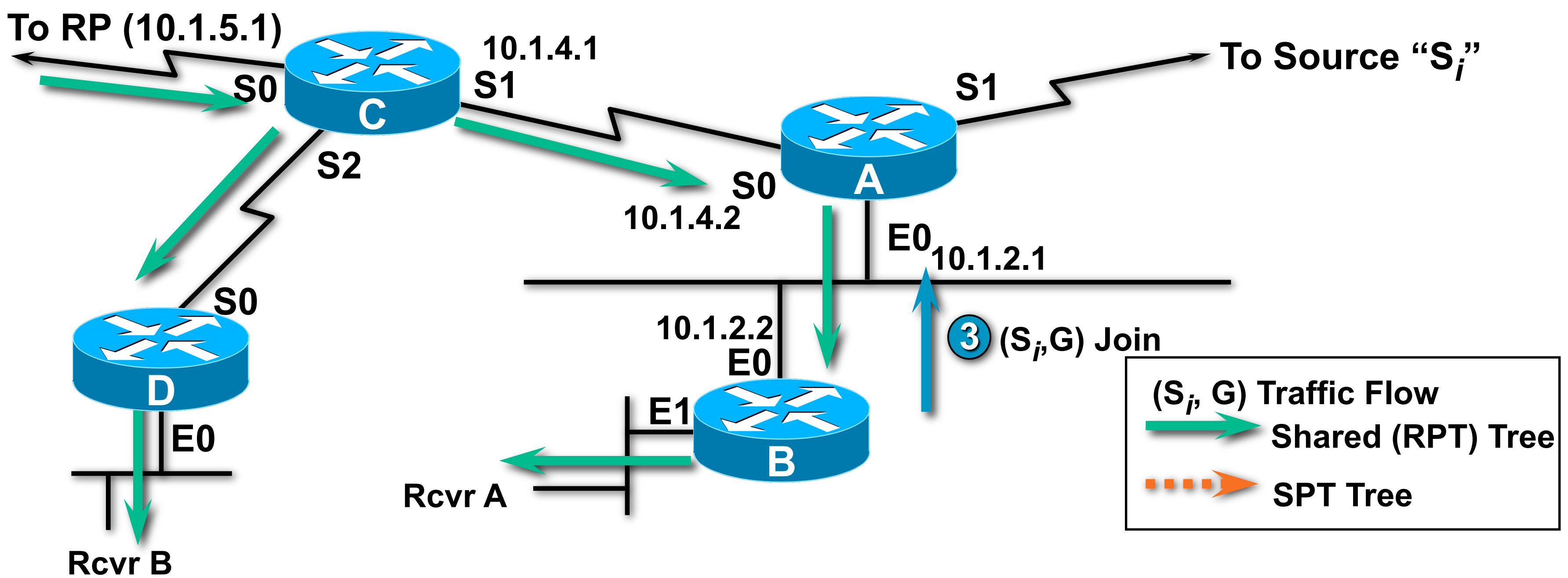
PIM SM SPT-Switchover (XR)




```
(*,224.1.1.1) RPF nbr: 10.1.2.1 Flags: C
  Incoming Interface List
    Ethernet0 Flags: A NS, Up: 08:27:55
  Outgoing Interface List
    Ethernet1 Flags: F NS LI, Up: 08:45:43

(171.68.37.121,224.1.1.1) RPF nbr: 10.1.2.1 Flags:
  Incoming Interface List
    Ethernet0 Flags: A, Up: 00:00:06
  Outgoing Interface List
    Ethernet1 Flags: F NS, Up: 00:00:06
```

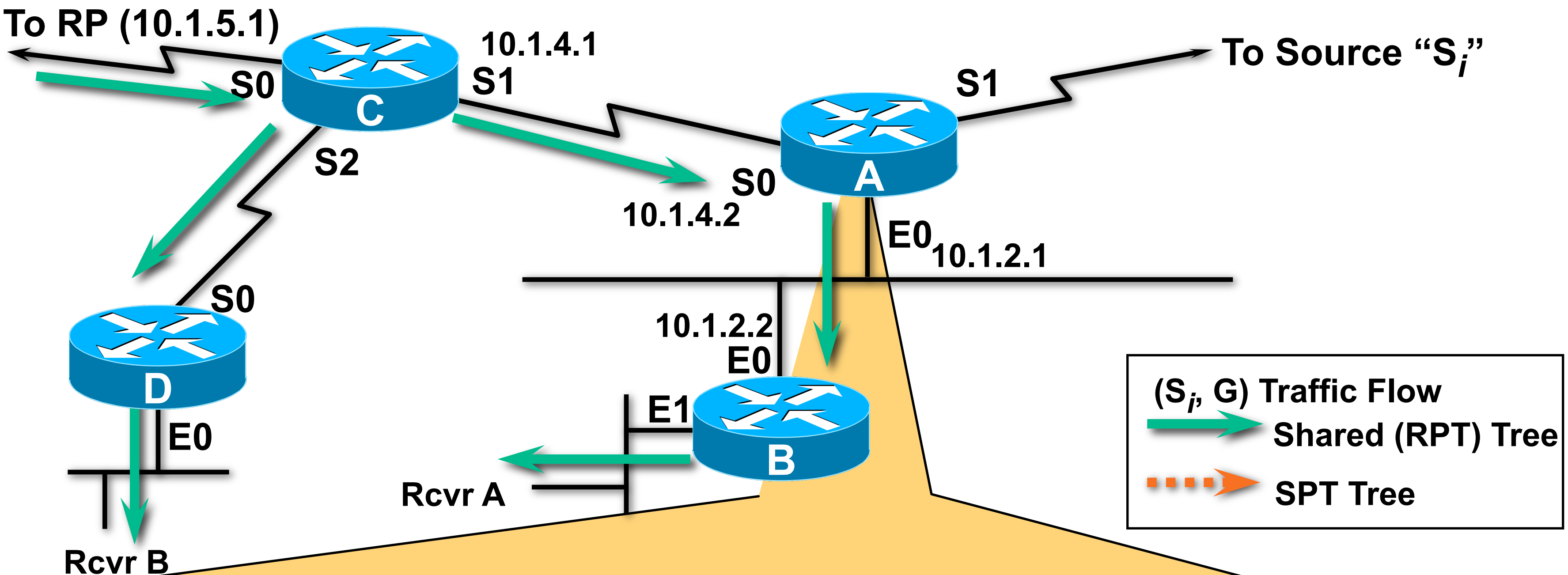
2 B creates (S_i,G) state.

PIM SM SPT-Switchover



3 B sends (S_i,G) Join towards S_i .

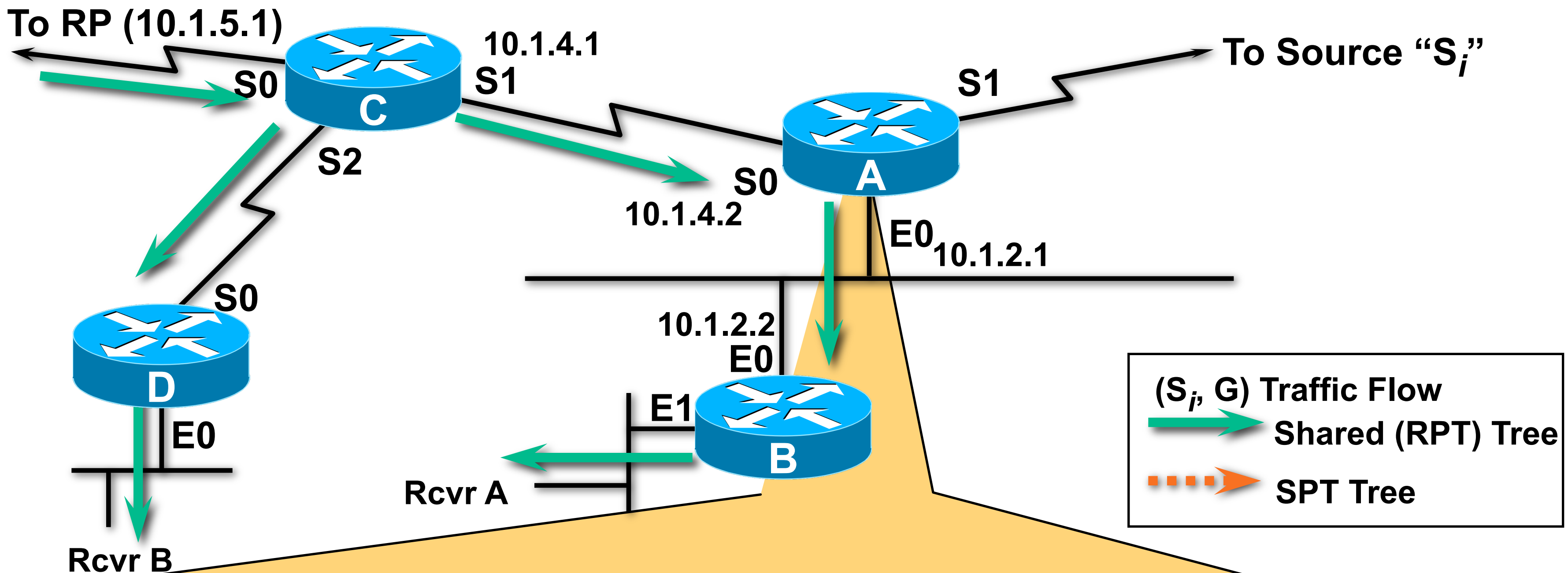
PIM SM SPT-Switchover




```
(*, 224.1.1.1), 00:01:43/00:00:00, RP 10.1.5.1, flags: S
  Incoming interface: Serial0, RPF nbr 10.1.4.1,
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:01:43/00:02:11

(171.68.37.121, 224.1.1.1), 00:13:28/00:02:53, flags:
  Incoming interface: Serial1, RPF nbr 10.1.9.2
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:13:25/00:02:30
```

New State in A

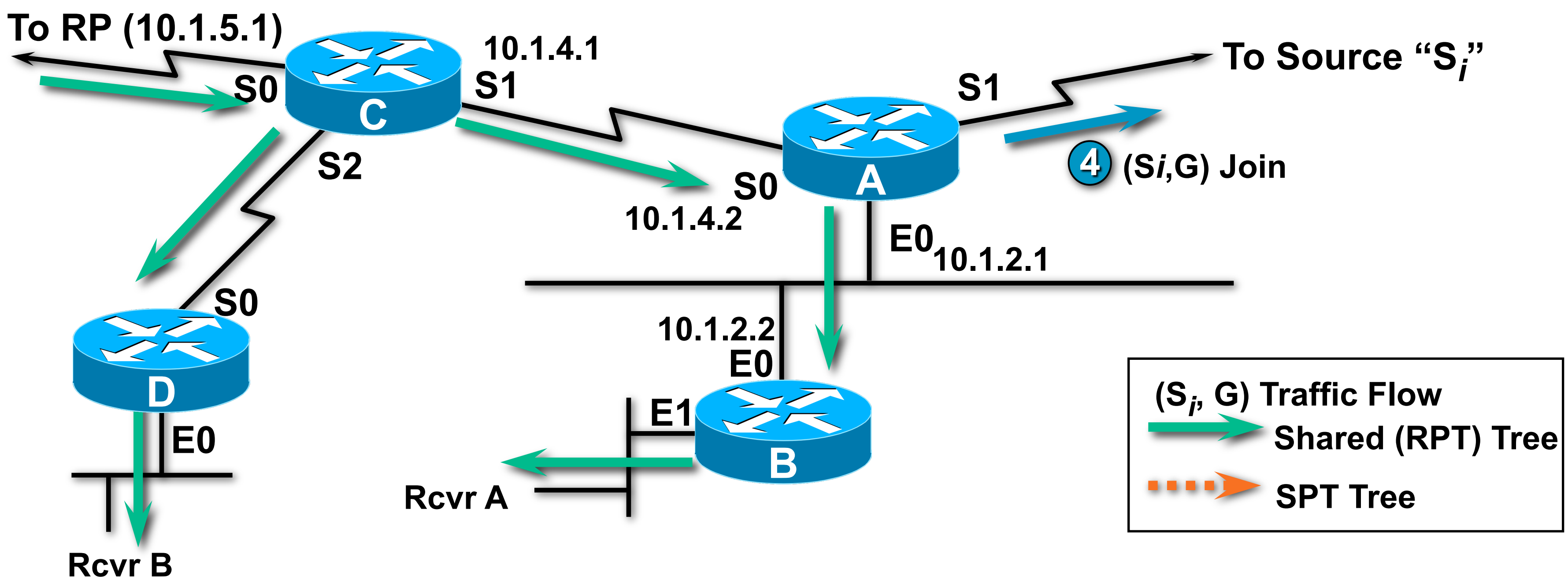
PIM SM SPT-Switchover (XR)




```
(*,224.1.1.1) RPF nbr: 10.1.4.1 Flags: C
  Incoming Interface List
    Serial0 Flags: A, Up: 08:29:24
  Outgoing Interface List
    Ethernet0 Flags: F NS, Up: 08:29:24

(171.68.37.121,224.1.1.1) RPF nbr: 10.1.2.1 Flags:
  Incoming Interface List
    Serial1 Flags: A, Up: 00:02:13
  Outgoing Interface List
    Ethernet0 Flags: F NS, Up: 00:02:13
```

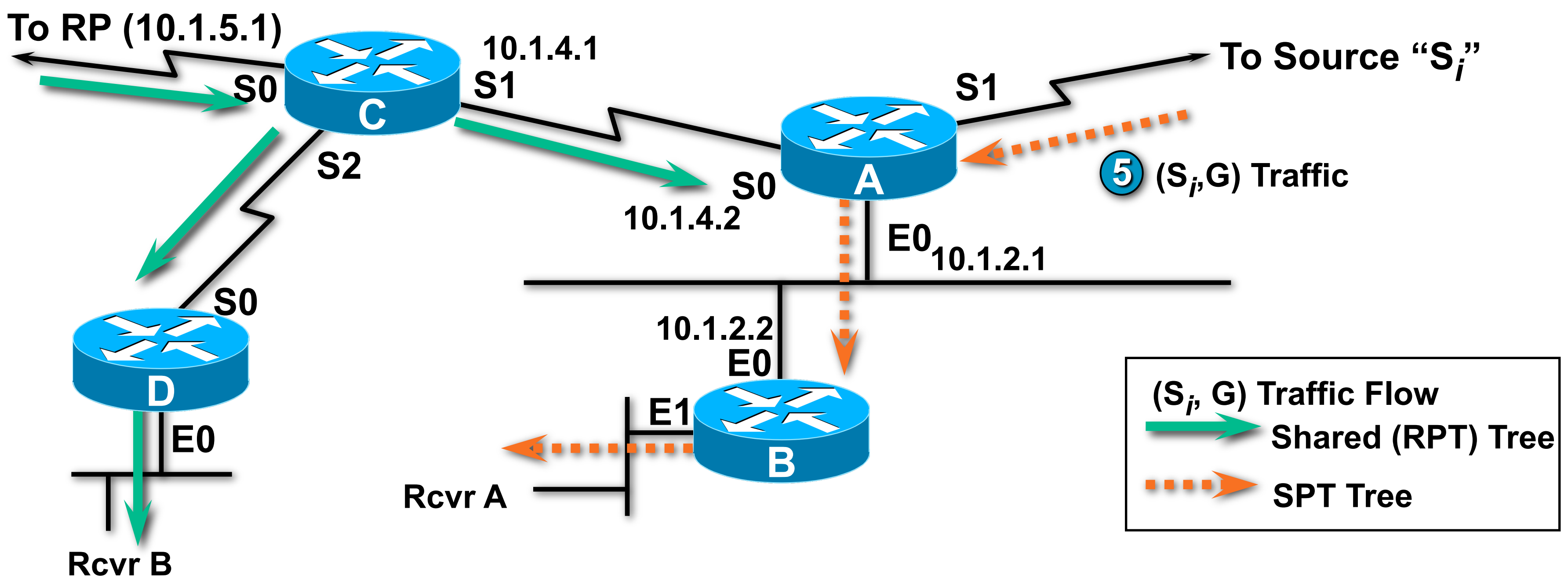
New State in A

PIM SM SPT-Switchover



4 A triggers (S_i,G) Join toward S_i.

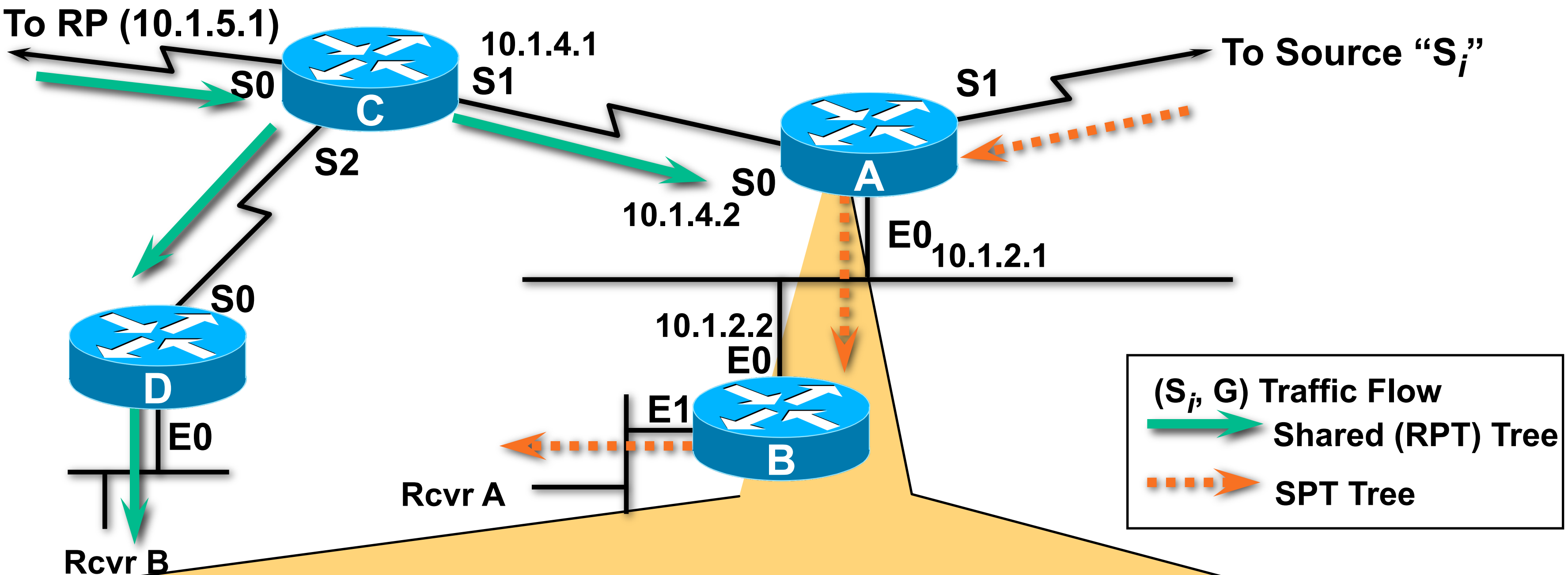
PIM SM SPT-Switchover



4 A triggers (S_i,G) Join toward S_i.

5 (S_i, G) traffic begins flowing down SPT tree.

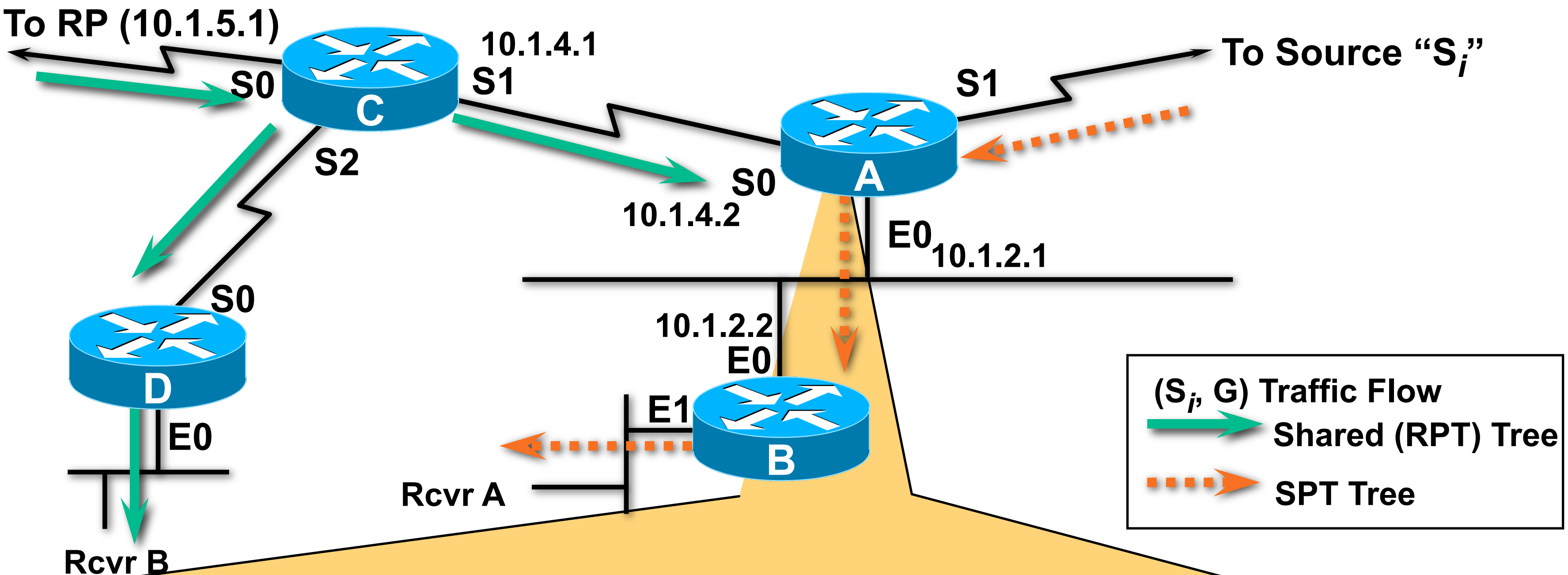
PIM SM SPT-Switchover




```
(*, 224.1.1.1), 00:01:43/00:00:00, RP 10.1.5.1, flags: S
  Incoming interface: Serial0, RPF nbr 10.1.4.1,
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:01:43/00:02:11

(171.68.37.121, 224.1.1.1), 00:13:28/00:02:53, flags: T
  Incoming interface: Serial1, RPF nbr 10.1.9.2
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:13:25/00:02:30
```

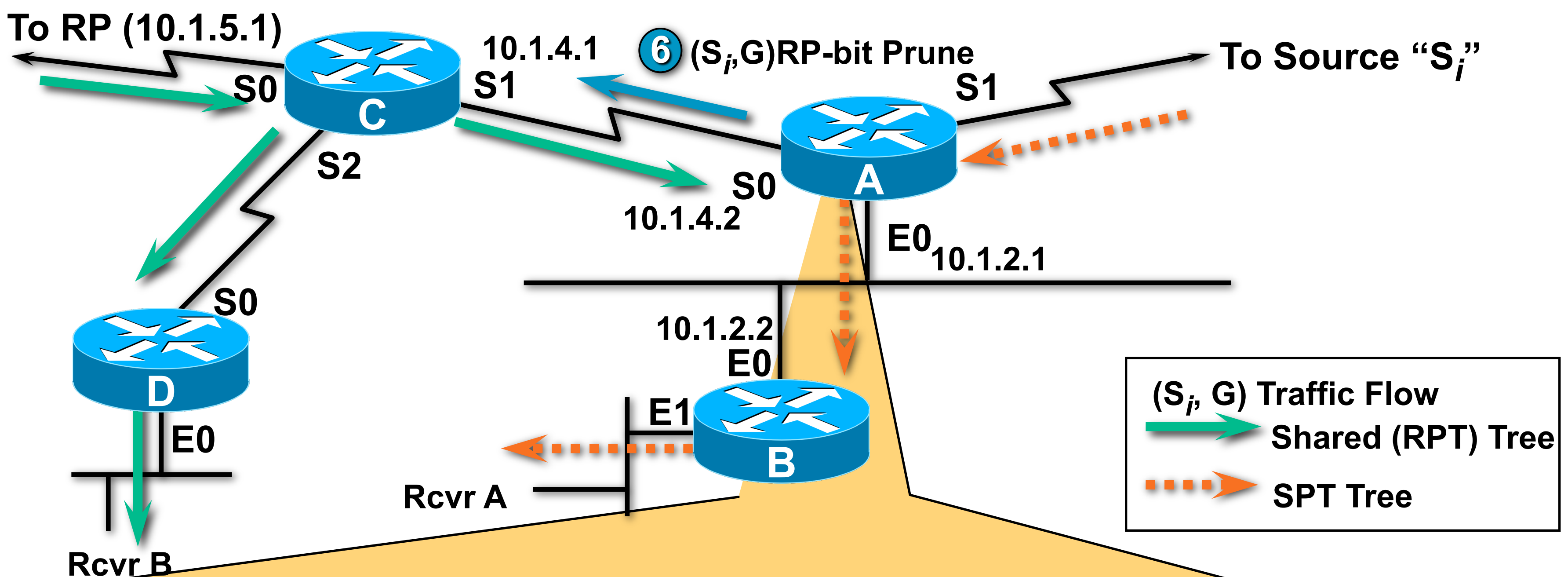
"T"Flag Set by Arriving Traffic on SPT

PIM SM SPT-Switchover (XR)




```
(*,224.1.1.1) RPF nbr: 10.1.4.1 Flags: C
  Incoming Interface List
    Serial0 Flags: A, Up: 08:29:24
  Outgoing Interface List
    Ethernet0 Flags: F NS, Up: 08:29:24

(171.68.37.121,224.1.1.1) RPF nbr: 10.1.9.2 Flags:
  Incoming Interface List
    Serial1 Flags: A, Up: 00:02:13
  Outgoing Interface List
    Ethernet0 Flags: F NS, Up: 00:02:13
```

PIM SM SPT-Switchover

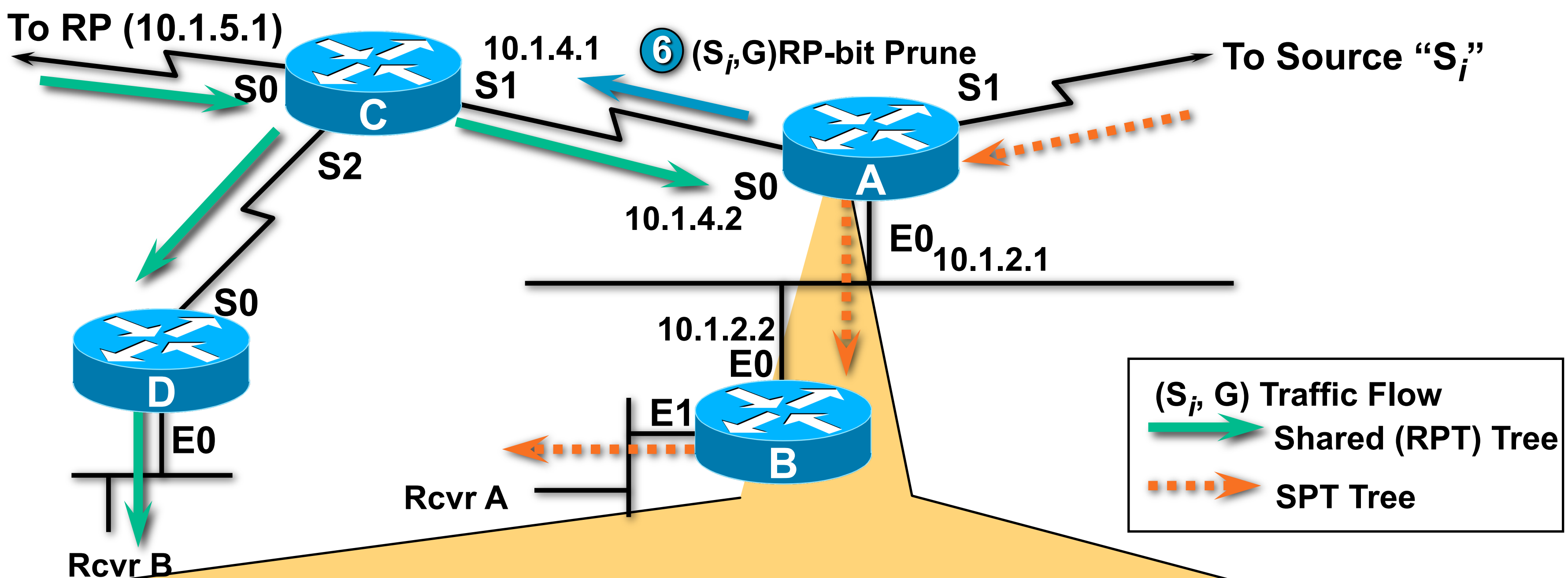



```
(*, 224.1.1.1), 00:01:43/00:00:00, RP 10.1.5.1, flags: S
  Incoming interface: Serial0, RPF nbr 10.1.4.1,
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:01:43/00:02:11

(171.68.37.121, 224.1.1.1), 00:13:28/00:02:53, flags:T
  Incoming interface: Serial1, RPF nbr 10.1.9.2
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:13:25/00:02:30
```

Note RPF Info Does Not Match. This Indicates SPT and RPT Diverge.

6 Once "T" flag is set, A triggers (S$_i$,G)RP-bit Prunes toward RP.

PIM SM SPT-Switchover (XR)

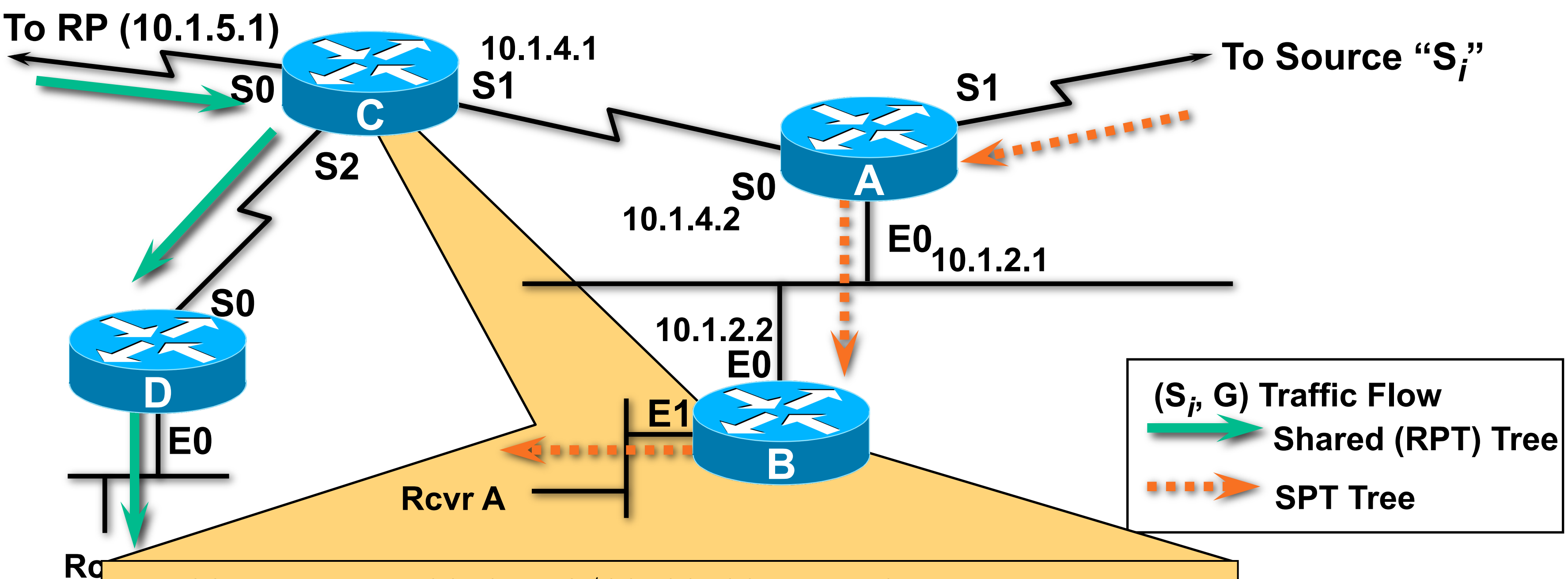



```
(*,224.1.1.1) RPF nbr: 10.1.4.1 Flags: C
  Incoming Interface List
    Serial0 Flags: A, Up: 08:29:24
  Outgoing Interface List
    Ethernet0 Flags: F NS, Up: 08:29:24

(171.68.37.121,224.1.1.1) RPF nbr: 10.1.9.2 Flags:
  Incoming Interface List
    Serial1 Flags: A, Up: 00:02:13
  Outgoing Interface List
    Ethernet0 Flags: F NS, Up: 00:02:13
```

Note RPF Info Does Not Match. This Indicates SPT and RPT Diverge.

6 A sends (S_i,G)RP-bit Prunes toward RP.

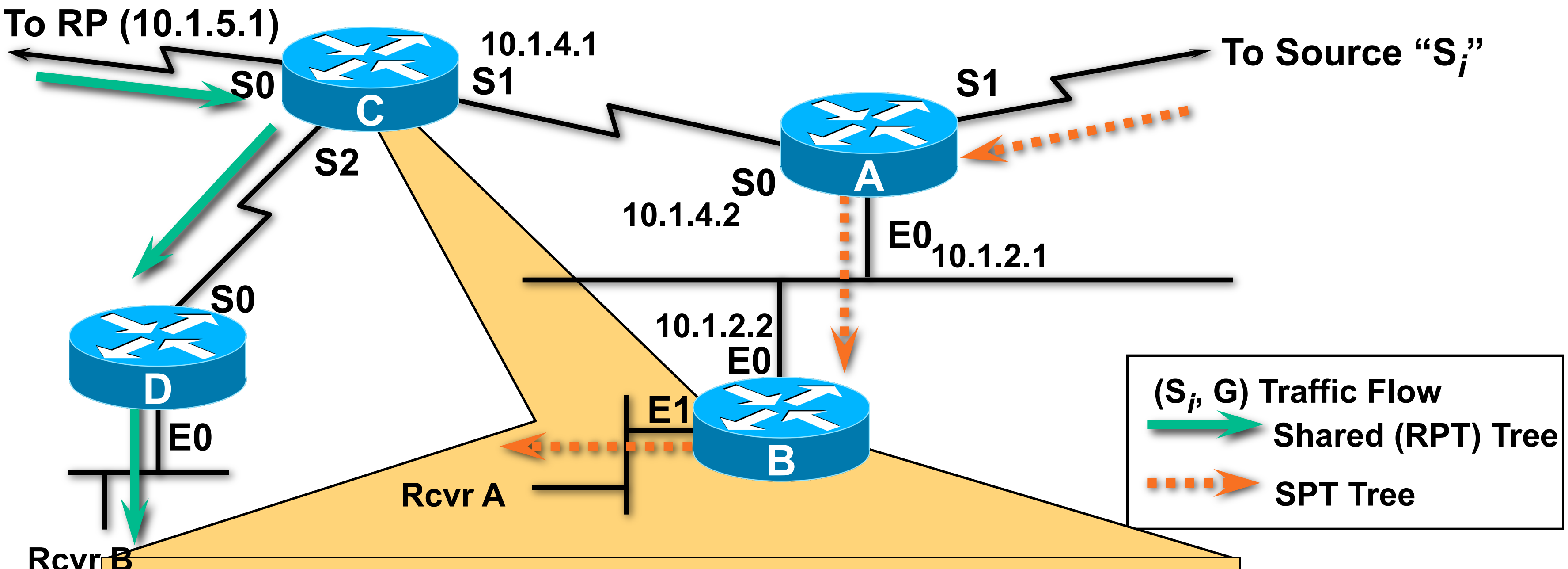
PIM SM SPT-Switchover




```
(*, 224.1.1.1), 00:01:43/00:00:00, RP 10.1.5.1, flags: S
  Incoming interface: Serial0, RPF nbr 10.1.5.1,
  Outgoing interface list:
    Serial1, Forward/Sparse-Dense, 00:01:43/00:02:11
    Serial2, Forward/Sparse-Dense, 00:00:32/00:02:28

(171.68.37.121, 224.1.1.1), 00:13:28/00:02:53, flags: R
  Incoming interface: Serial0, RPF nbr 10.1.5.1
  Outgoing interface list:
    Serial2, Forward/Sparse-Dense, 00:00:32/00:02:28
```

State in C After Receiving the (S_i, G) RP-bit Prune

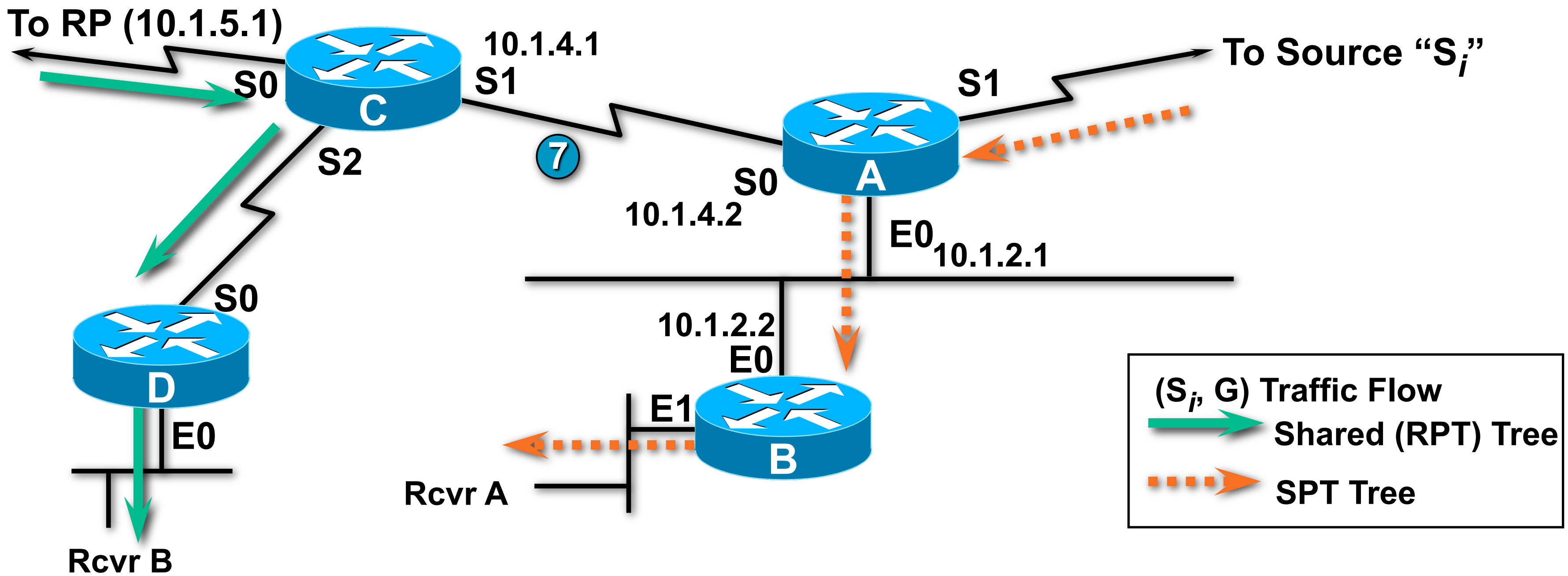
PIM SM SPT-Switchover (XR)




```
(*,224.1.1.1) RPF nbr: 10.1.5.1 Flags: C
  Incoming Interface List
    Serial0 Flags: A, Up: 07:20:03
  Outgoing Interface List
    Serial1 Flags: F NS, Up: 07:20:03
    Serial2 Flags: F NS, Up: 07:20:03

(171.68.37.121,224.1.1.1) RPF nbr: 10.1.5.1 Flags:
  Incoming Interface List
    Serial0 Flags: A, Up: 00:02:13
  Outgoing Interface List
    Serial2 Flags: F NS, Up: 00:02:13
```

State in C After Receiving the (S_i, G) RP-bit Prune

PIM SM SPT-Switchover

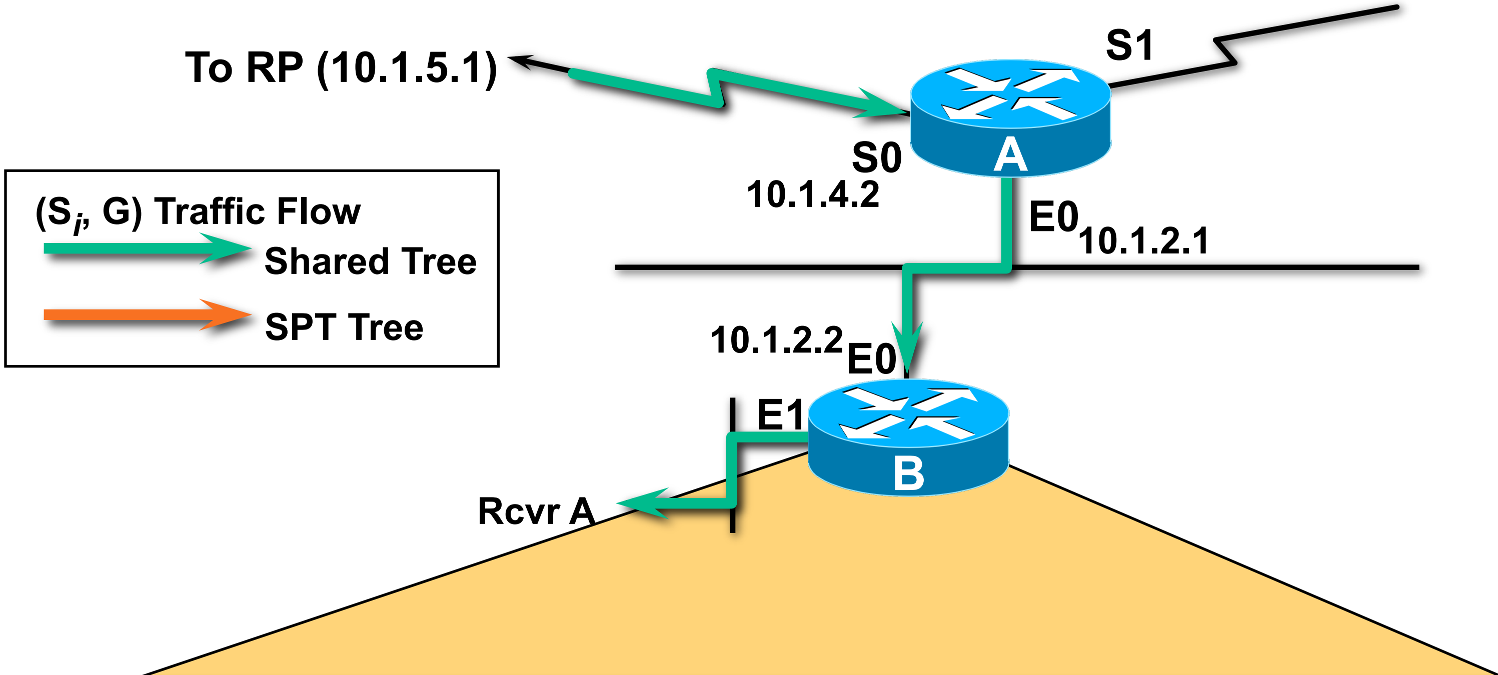


7 **Unnecessary (S_i, G) traffic is pruned from the Shared tree.**

PIM SM Pruning

PIM SM Pruning

Shared Tree Case

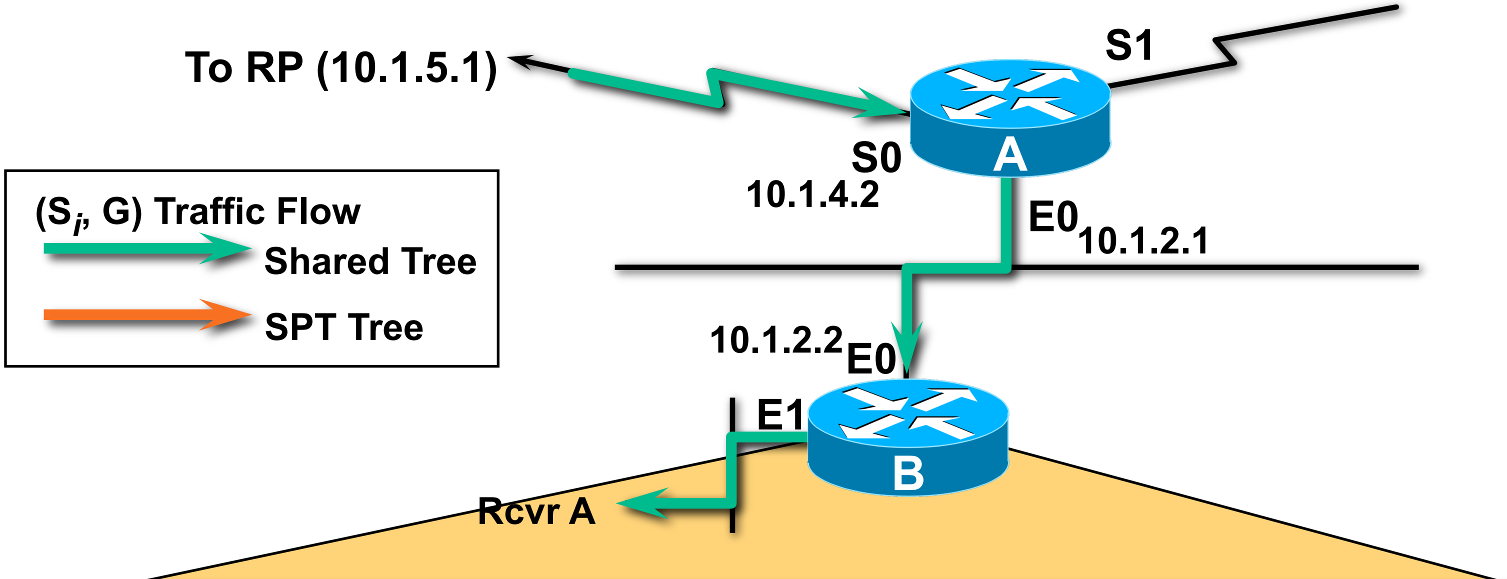



```
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: SC
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:01:43/00:02:11
```

State in B Before Pruning

PIM SM Pruning (XR)

Shared Tree Case

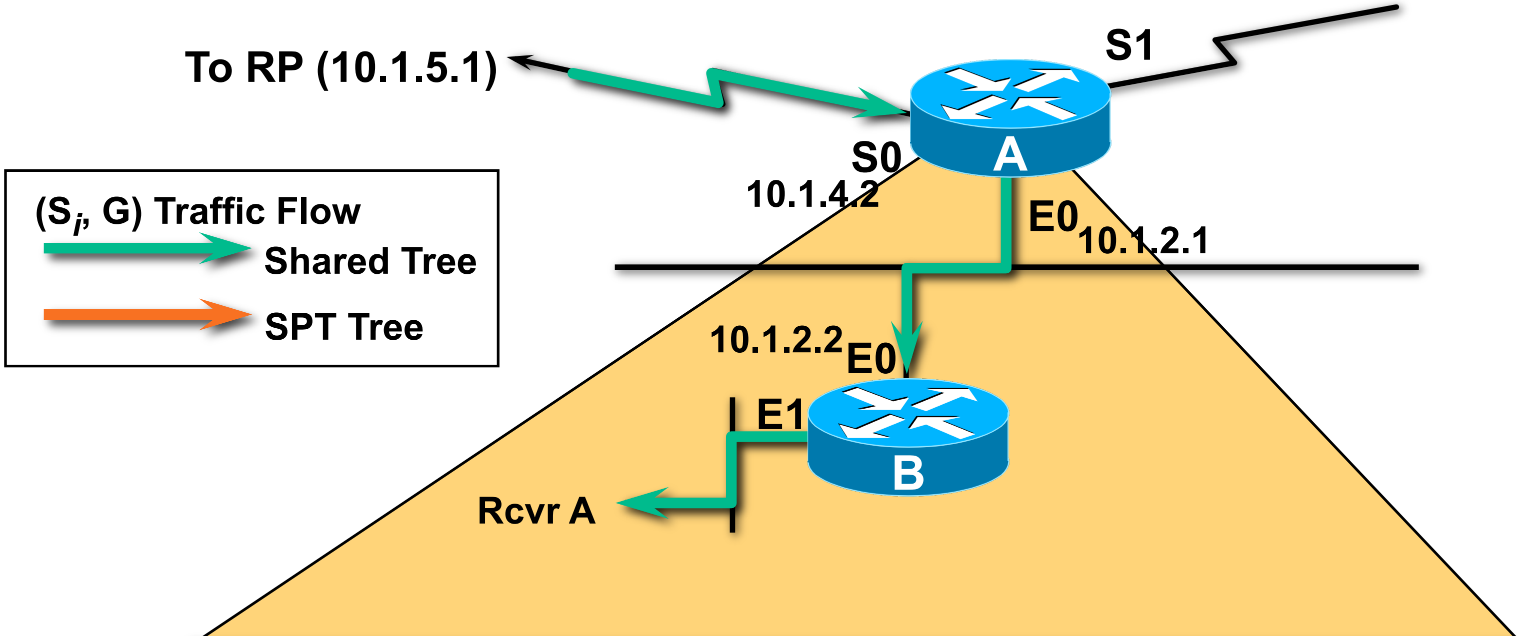



```
(*,224.1.1.1) RPF nbr: 10.1.2.1 Flags: C
  Incoming Interface List
    Ethernet0 Flags: A, Up: 17:37:56
  Outgoing Interface List
    Ethernet1 Flags: F NS LI, Up: 17:55:44
```

State in B Before Pruning

PIM SM Pruning

Shared Tree Case

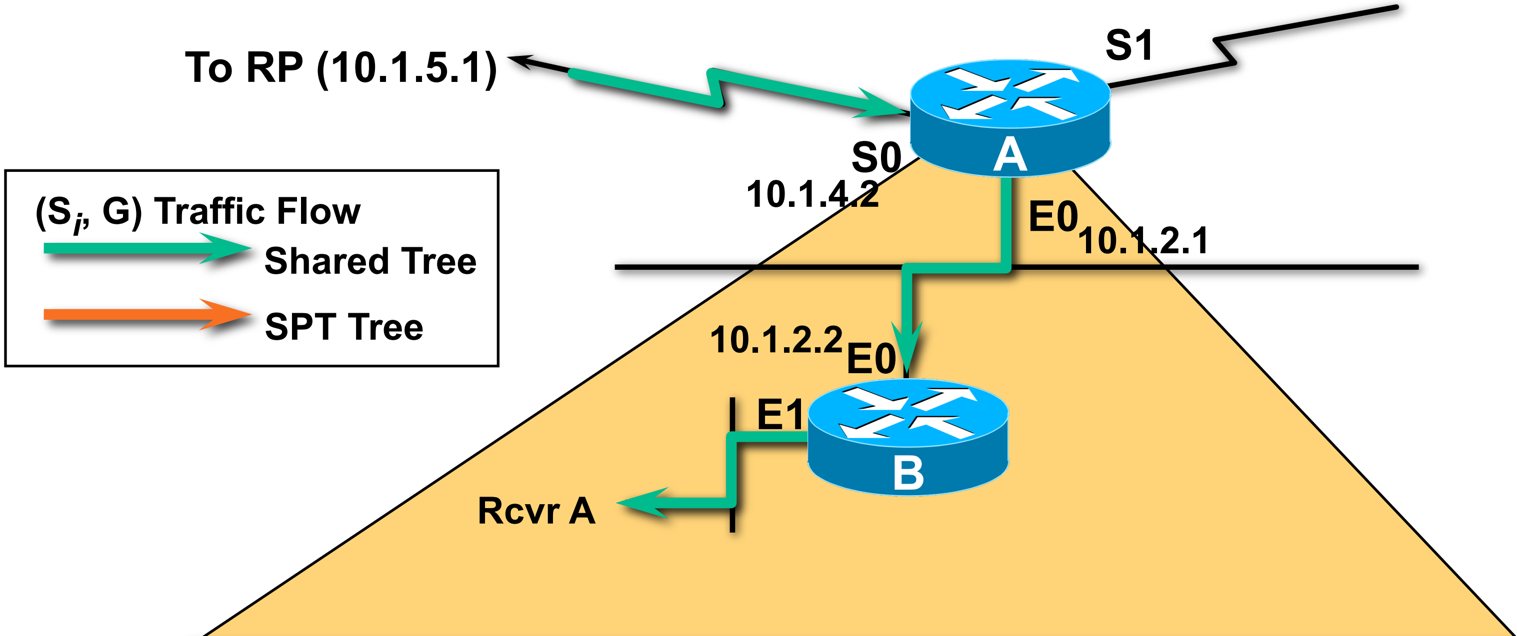



```
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: S
  Incoming interface: Serial0, RPF nbr 10.1.4.1,
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:01:43/00:02:11
```

State in A Before Pruning

PIM SM Pruning (XR)

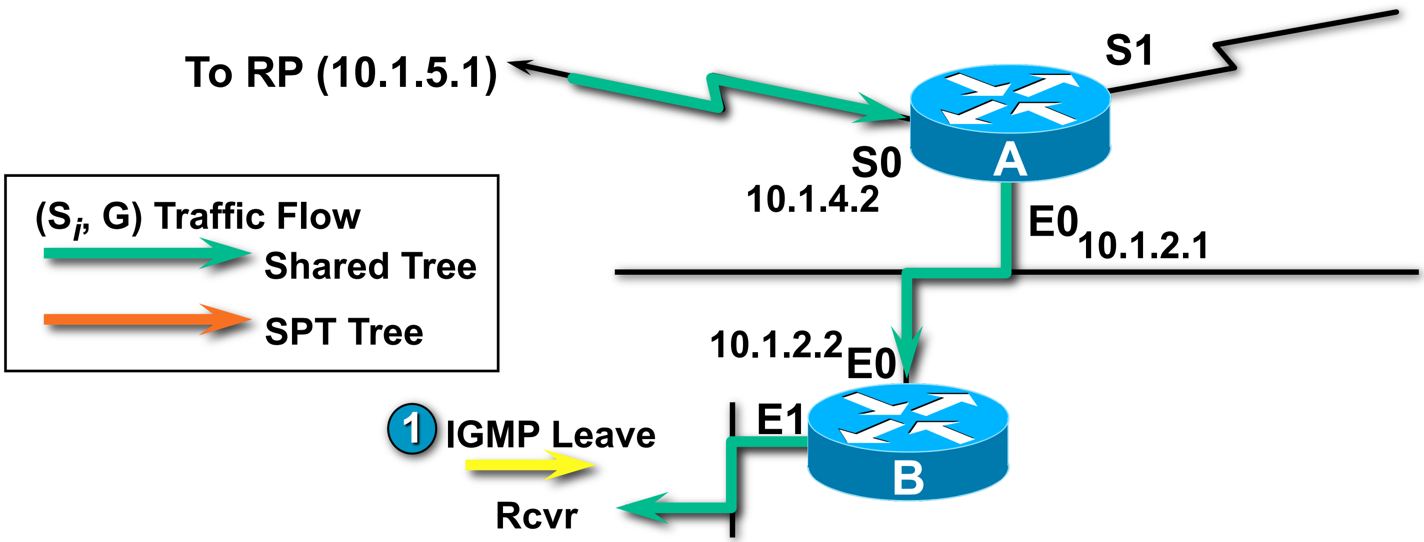
Shared Tree Case




```
(*,224.1.1.1) RPF nbr: 10.1.4.1 Flags: C
  Up: 00:03:37
  Incoming Interface List
    Serial0 Flags: A, Up: 00:03:37
  Outgoing Interface List
    Ethernet0 Flags: F NS, Up: 00:03:37
```

State in A Before Pruning

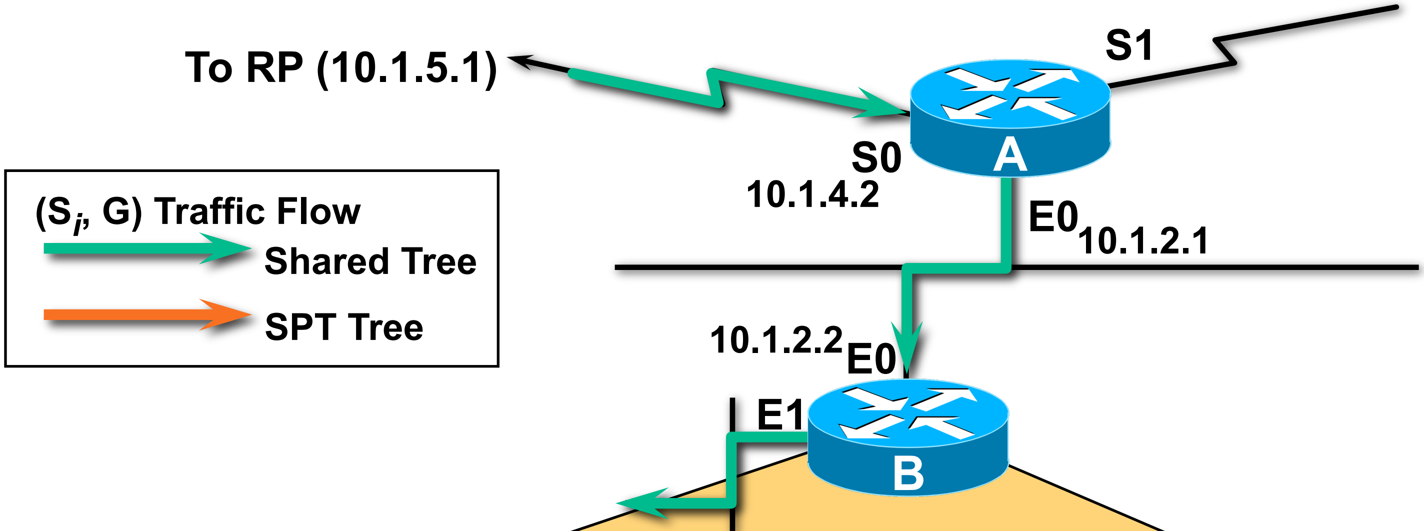
PIM SM Pruning
Shared Tree Case



1 B is a Leaf router. Last Rcvr, leaves group G.

PIM SM Pruning

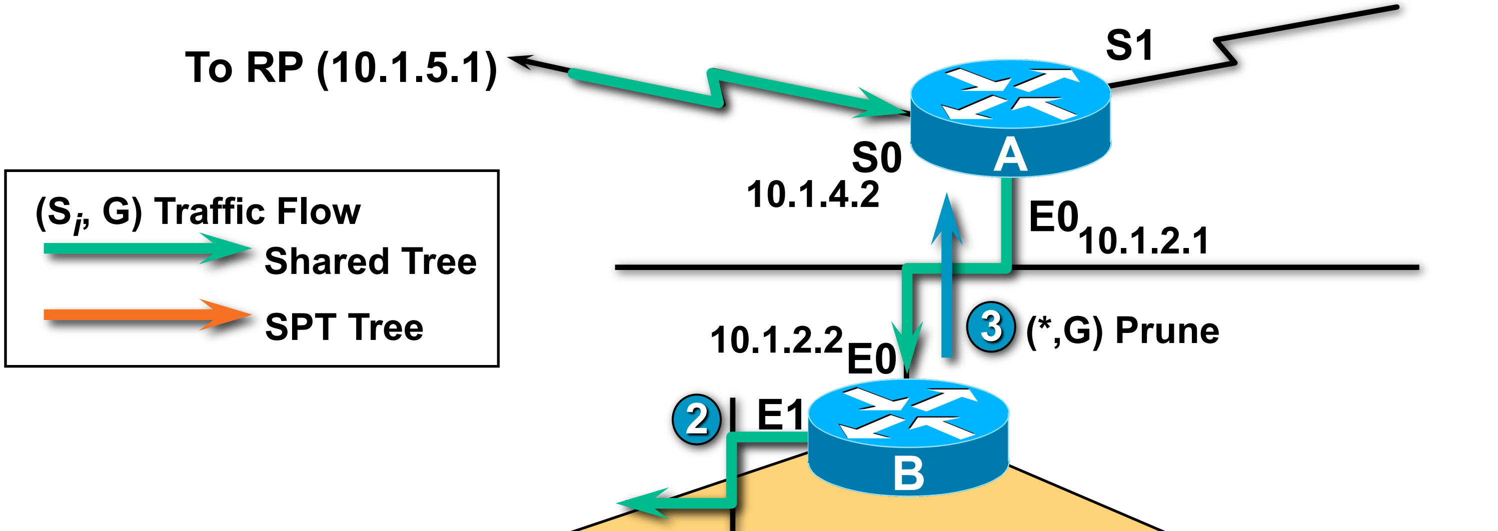
Shared Tree Case




```
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: S
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:01:43/00:02:11
```


PIM SM Pruning

Shared Tree Case

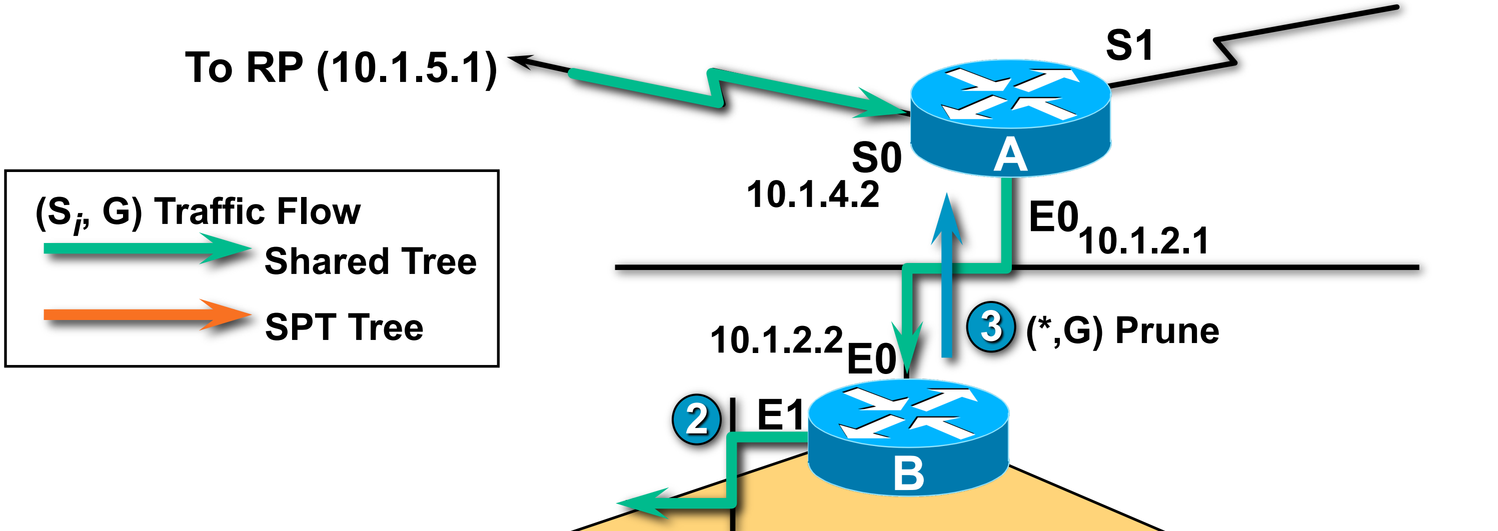



```
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: SP
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:
```

2. **B removes E1 from (*,G) and any (S_i,G) *"oilists"*.**
3. **B's (*,G) *"oilist"* now empty; triggers (*,G) Prune toward RP.**

PIM SM Pruning (XR)

Shared Tree Case

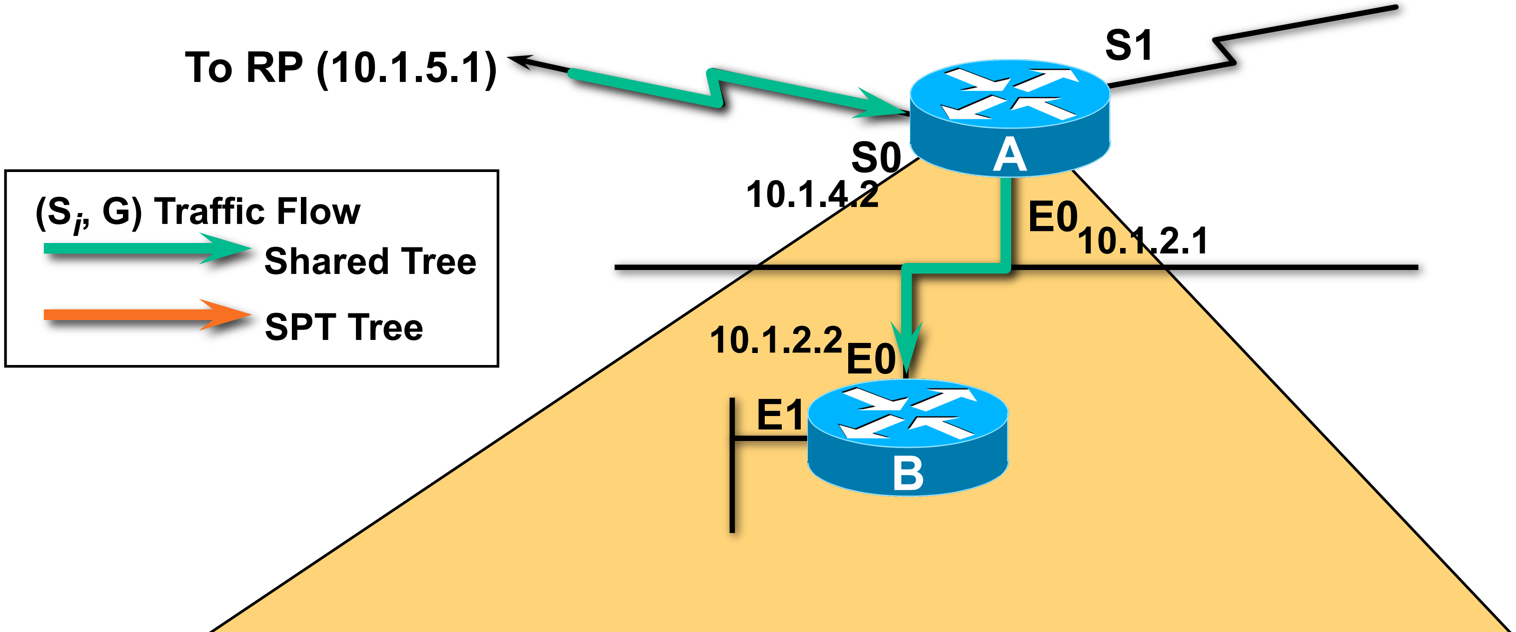



```
(*,224.1.1.1) RPF nbr: 10.1.2.1 Flags: C
  Incoming Interface List
    Ethernet0 Flags: A, Up: 17:37:56
```

2. **B removes E1 from (*,G) and any (S$_i$,G) *"oilists"*.**
3. **B's (*,G) *"oilist"* now empty; triggers (*,G) Prune toward RP.**

PIM SM Pruning

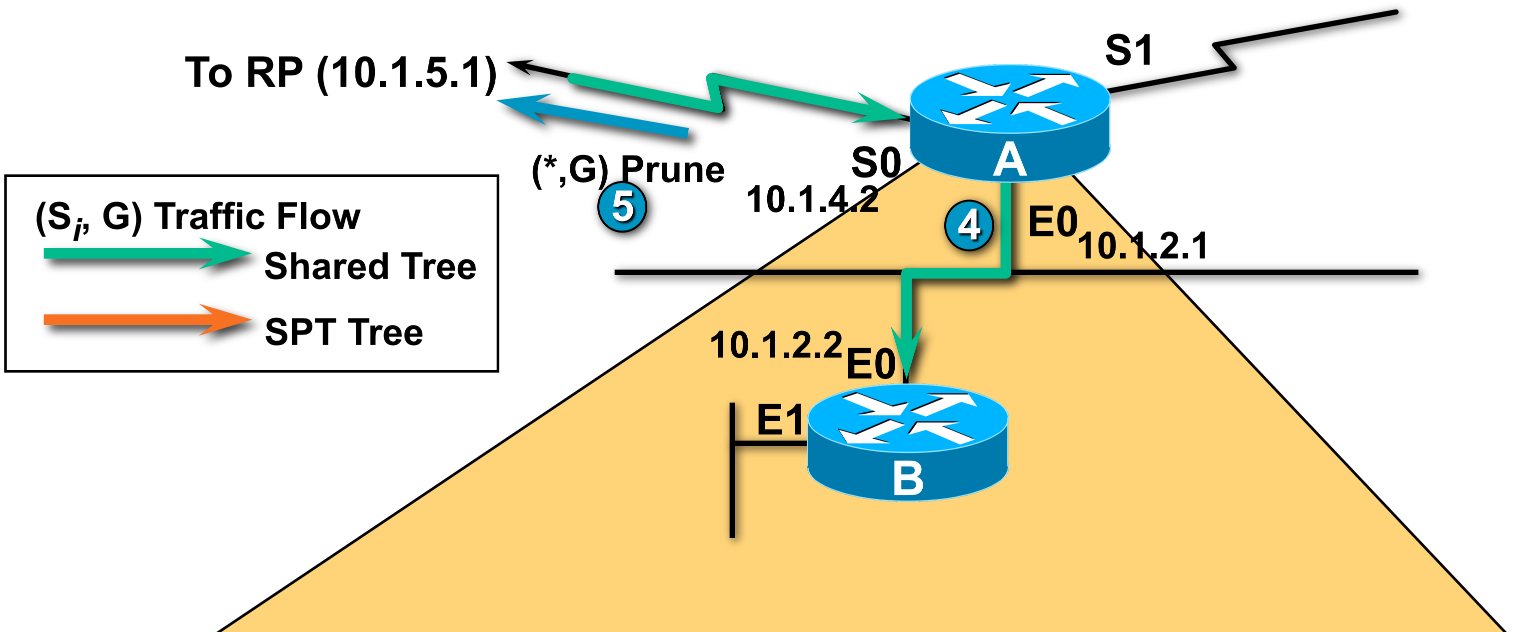
Shared Tree Case




```
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: S
  Incoming interface: Serial0, RPF nbr 10.1.4.1,
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:01:43/00:02:11
```

PIM SM Pruning

Shared Tree Case




```
(*, 224.1.1.1), 00:01:43/00:02:13, RP 10.1.5.1, flags: SP
  Incoming interface: Serial0, RPF nbr 10.1.4.1,
  Outgoing interface list:
```

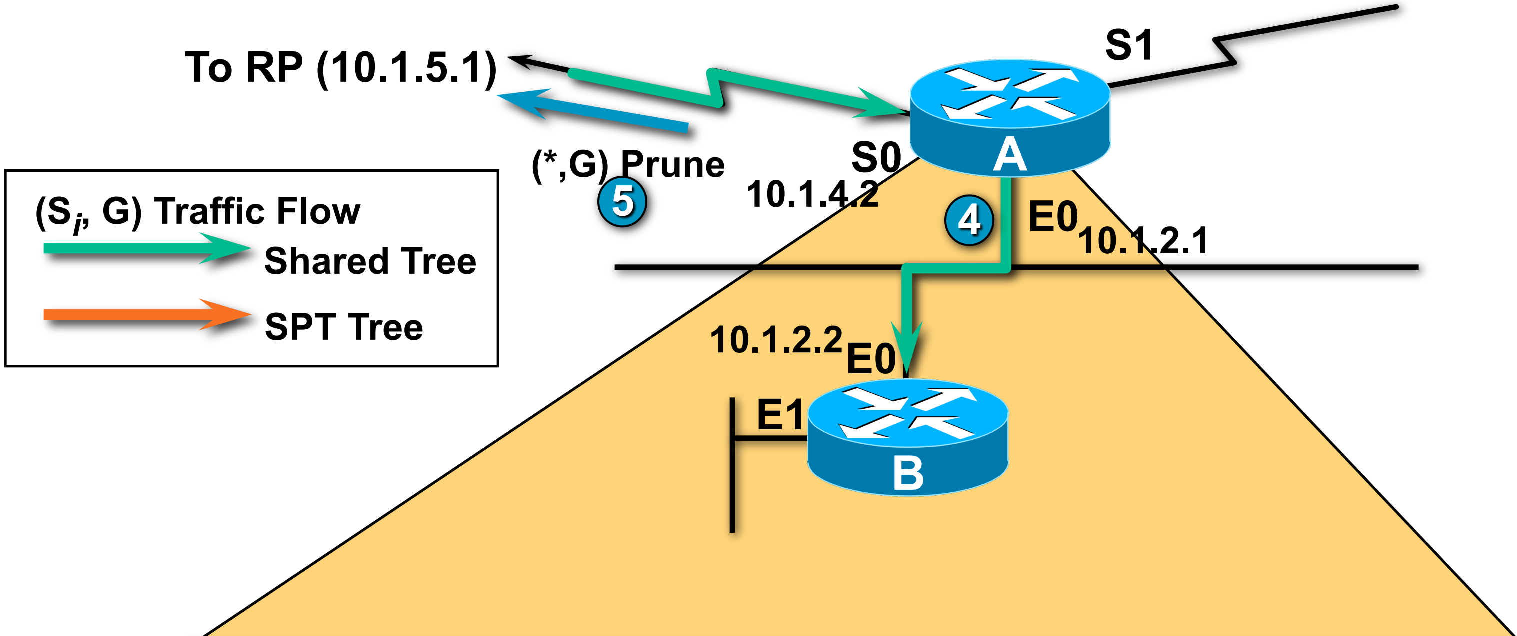
4 **A receives Prune; removes E0 from (*,G) *"oilist"*.**

(After the 3 second Multi-access Network Prune delay.)

5 **A's (*,G) *"oilist"* now empty; triggers (*,G) Prune toward RP.**

PIM SM Pruning (XR)

Shared Tree Case




```
(*,224.1.1.1) RPF nbr: 10.1.4.1 Flags: C
  Up: 00:03:37
  Incoming Interface List
    Serial0 Flags: A, Up: 00:03:37
```

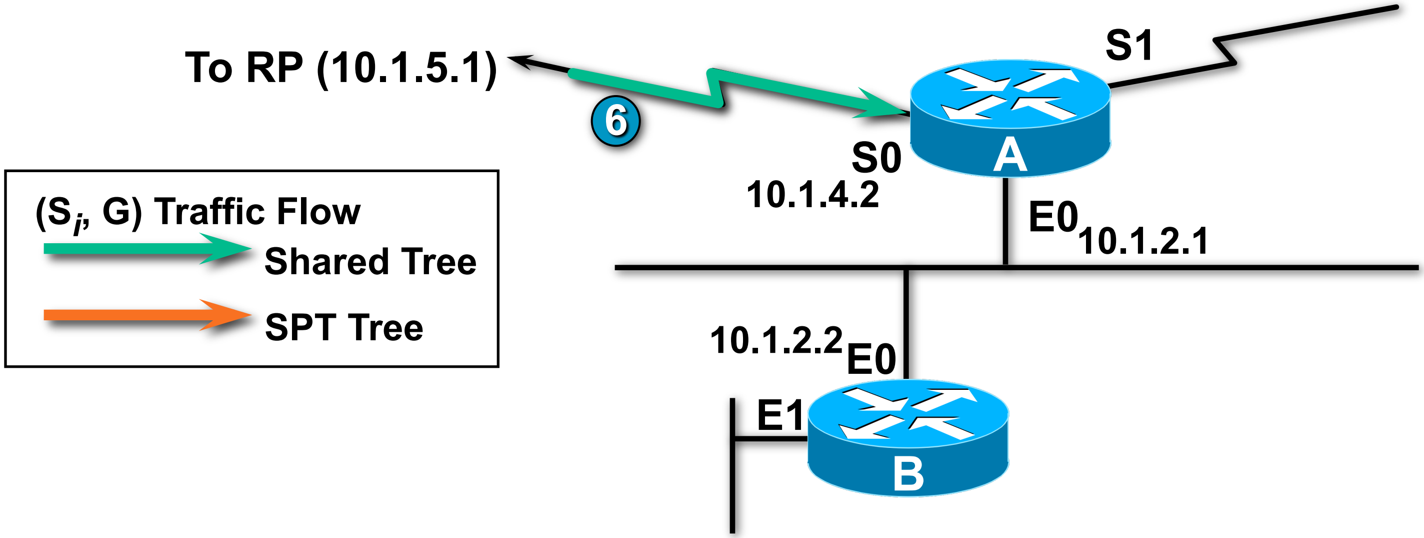
4 **A receives Prune; removes E0 from (*,G) *"oilist"*.**

(After the 3 second Multi-access Network Prune delay.)

5 **A's (*,G) *"oilist"* now empty; triggers (*,G) Prune toward RP.**

PIM SM Pruning

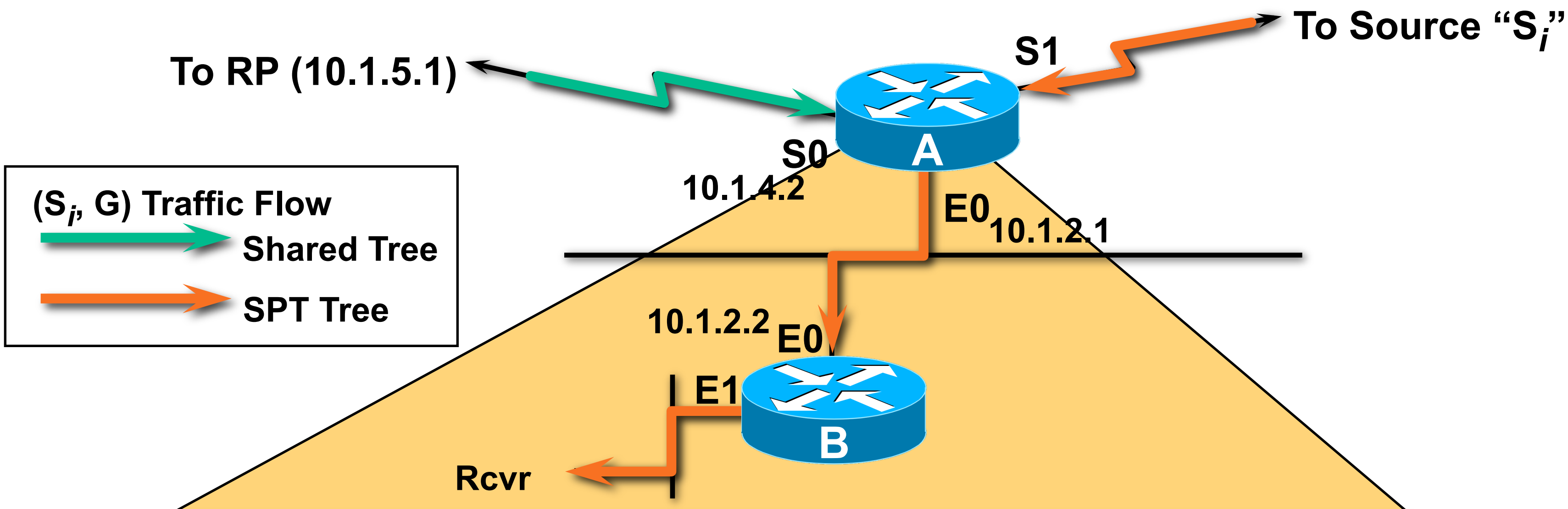
Shared Tree Case



6 **Pruning continues back toward RP.**

PIM SM Pruning

Source (SPT) Case

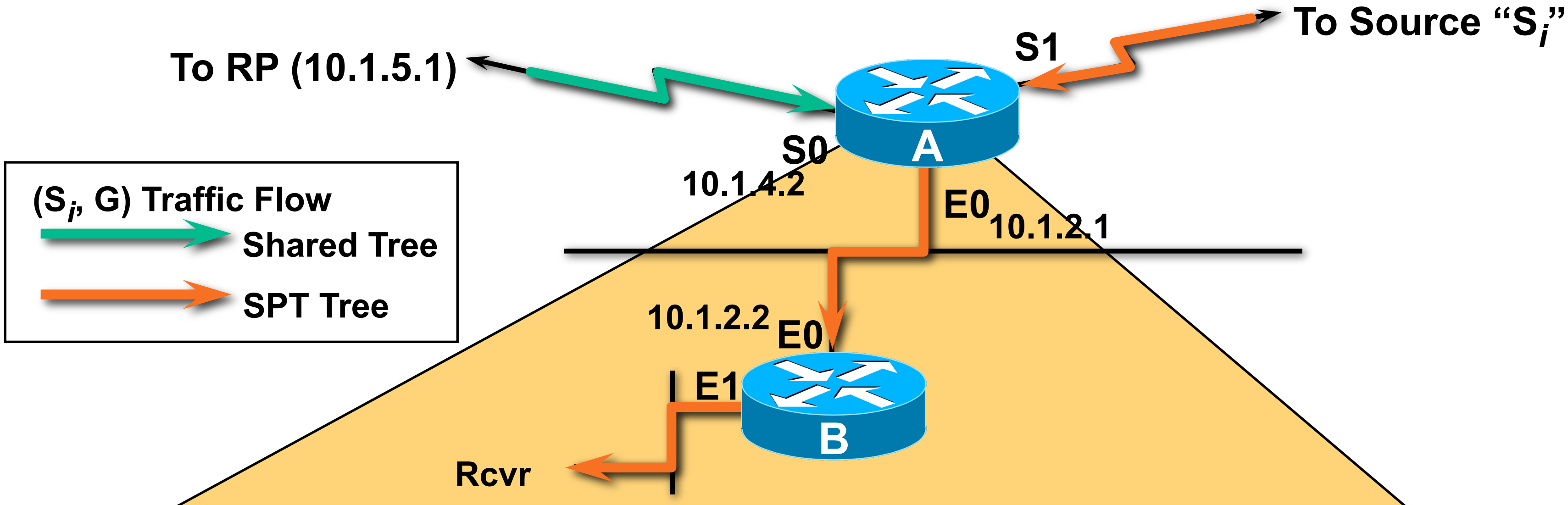



```
(*, 224.1.1.1), 00:01:43/00:00:00, RP 10.1.5.1, flags: S
  Incoming interface: Serial0, RPF nbr 10.1.4.1,
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:01:43/00:02:11

(171.68.37.121, 224.1.1.1), 00:01:05/00:01:55, flags: T
  Incoming interface: Serial1, RPF nbr 10.1.9.2
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:01:05/00:02:55
```

State in A Before Pruning

PIM SM Pruning (XR)

Source (SPT) Case

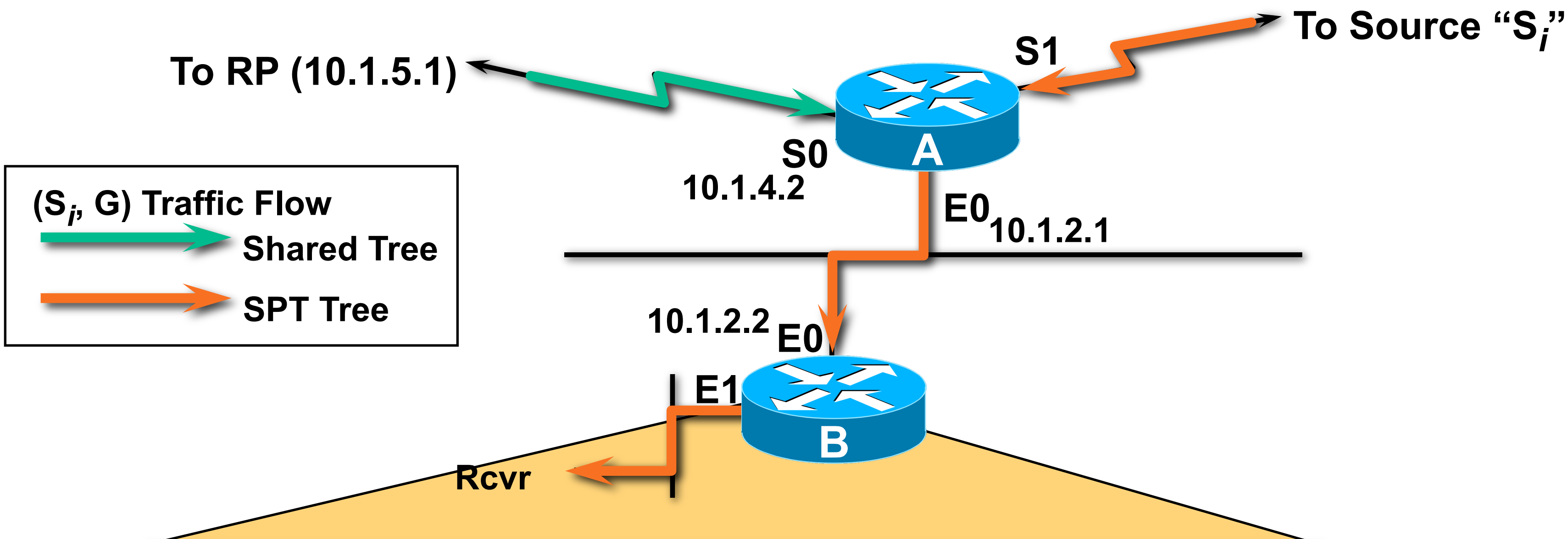



```
(*,224.1.1.1) RPF nbr: 10.1.4.1 Flags: C
  Incoming Interface List
    Serial0 Flags: A, Up: 00:02:03
  Outgoing Interface List
    Ethernet0 Flags: F NS, Up: 00:02:03

(171.68.37.121,224.1.1.1) RPF nbr: 10.1.9.2 Flags:
  Incoming Interface List
    Serial1 Flags: A, Up: 00:02:32
  Outgoing Interface List
    Ethernet0 Flags: F NS, Up: 00:02:03
```

State in A Before Pruning

PIM SM Pruning

Source (SPT) Case

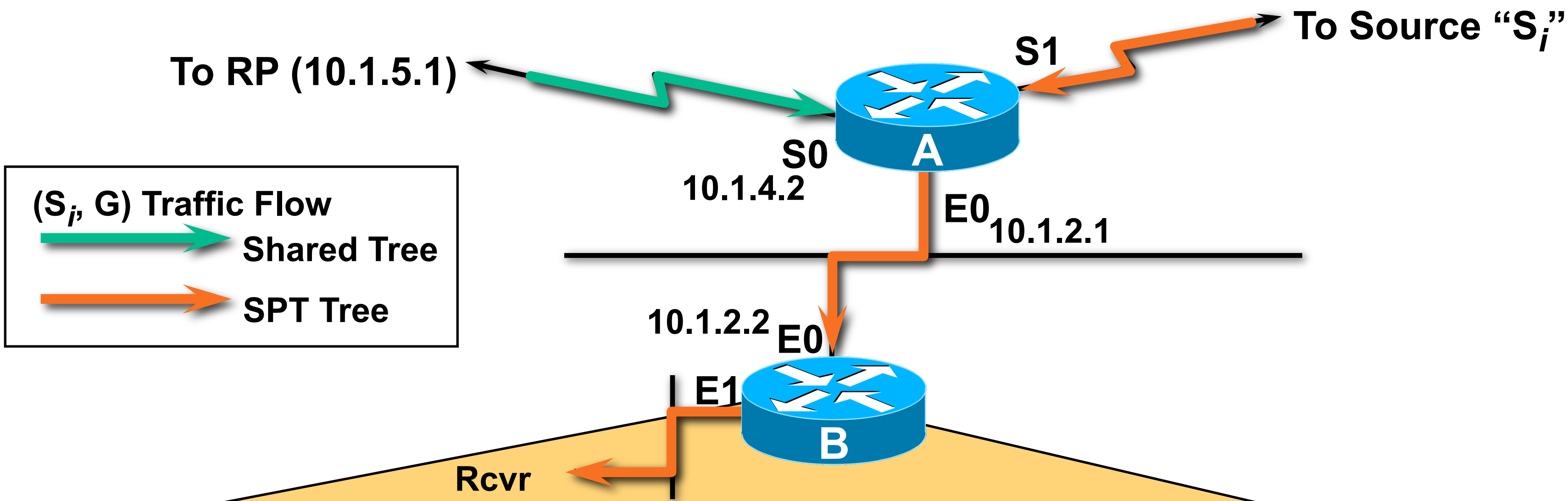



```
(*, 224.1.1.1), 00:01:43/00:00:00, RP 10.1.5.1, flags: SC
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:01:43/00:02:11

(171.68.37.121, 224.1.1.1), 00:01:05/00:01:55, flags: CJT
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:01:05/00:02:55
```

State in B Before Pruning

PIM SM Pruning (XR)

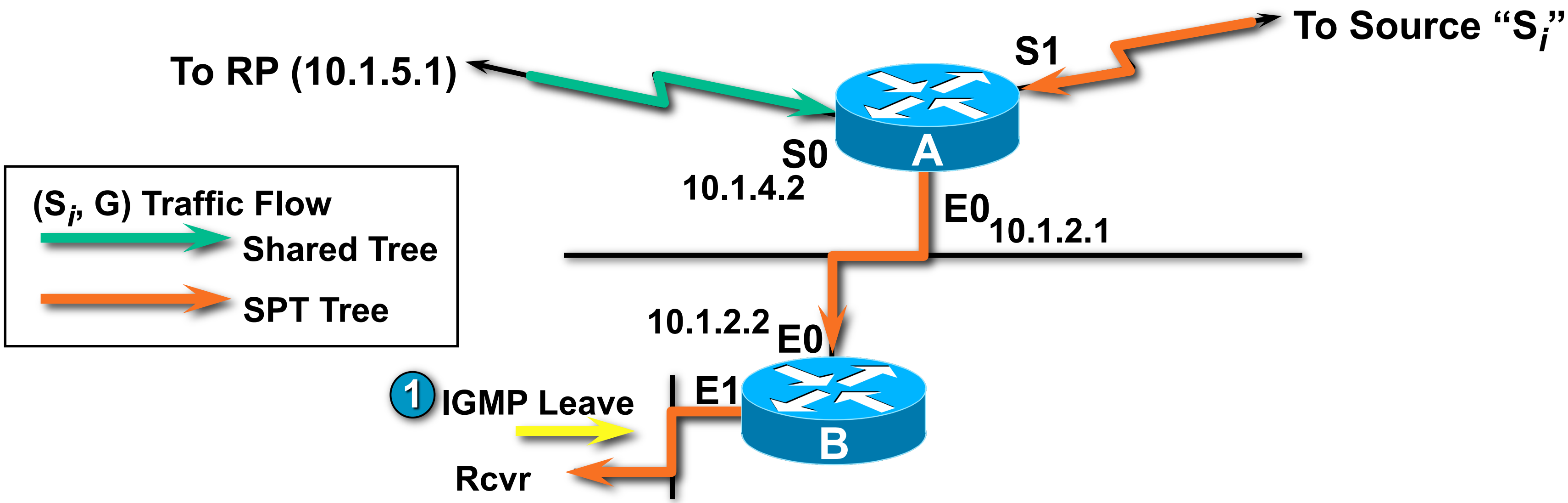
Source (SPT) Case




```
(*,224.1.1.1) RPF nbr: 10.1.2.1 Flags: C
  Incoming Interface List
    Ethernet0 Flags: A, Up: 17:37:56
  Outgoing Interface List
    Ethernet1 Flags: F NS LI, Up: 17:55:44

(171.68.37.121,224.1.1.1) RPF nbr: 10.1.2.1 Flags:
  Incoming Interface List
    Ethernet0 Flags: A, Up: 00:00:05
  Outgoing Interface List
    Ethernet1 Flags: F NS, Up: 00:00:05
```

State in B Before Pruning

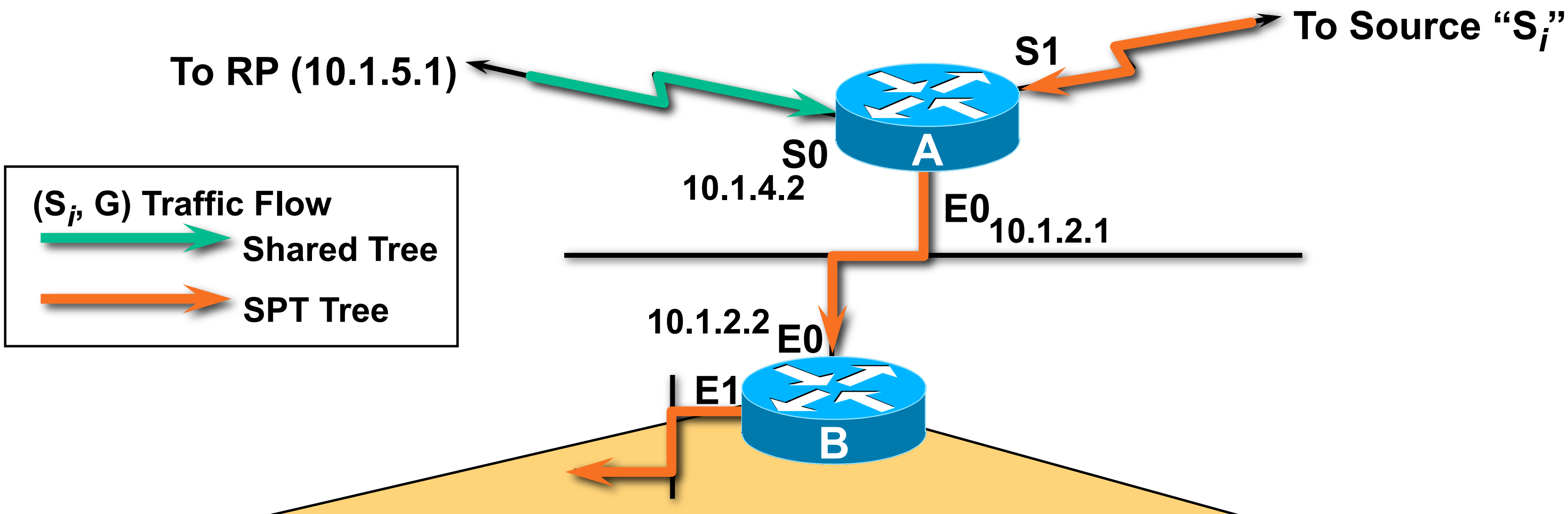
PIM SM Pruning
Source (SPT) Case



1 **B is a Leaf router. Last Rcvr leaves group G.**

PIM SM Pruning

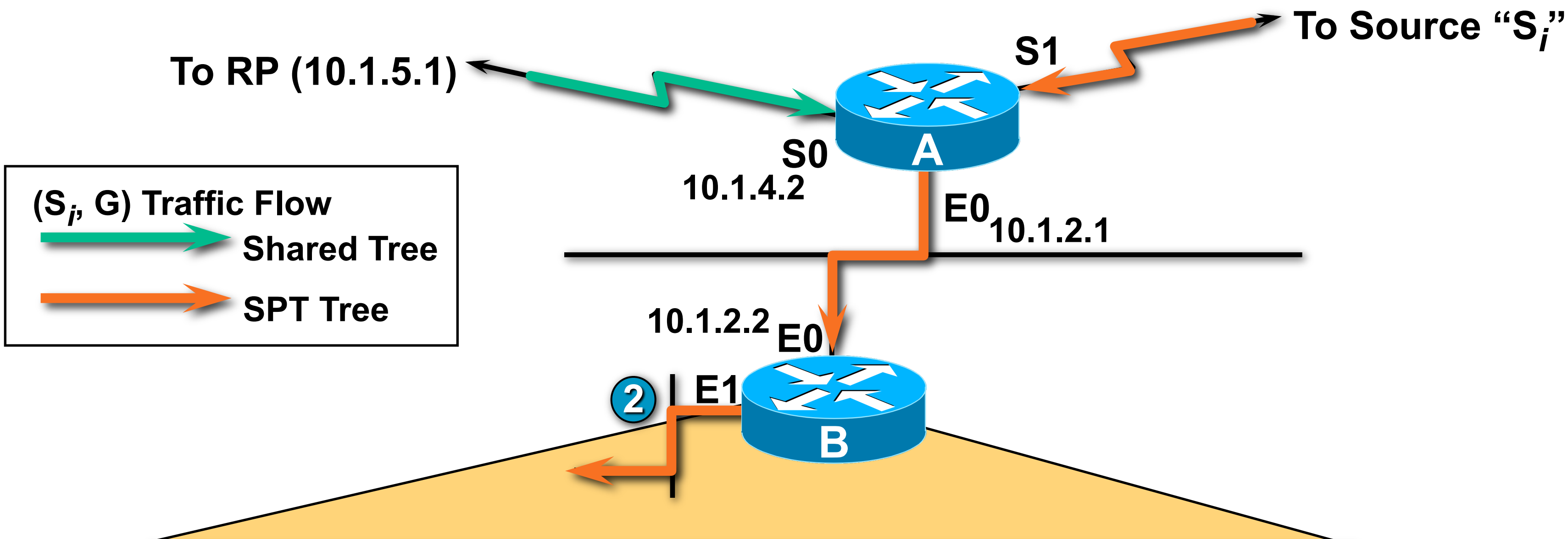
Source (SPT) Case




```
(*, 224.1.1.1), 00:01:43/00:02:59, RP 10.1.5.1, flags: SC
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:01:43/00:02:11
(171.68.37.121, 224.1.1.1), 00:01:05/00:01:55, flags: CJT
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:01:05/00:02:55
```

PIM SM Pruning

Source (SPT) Case

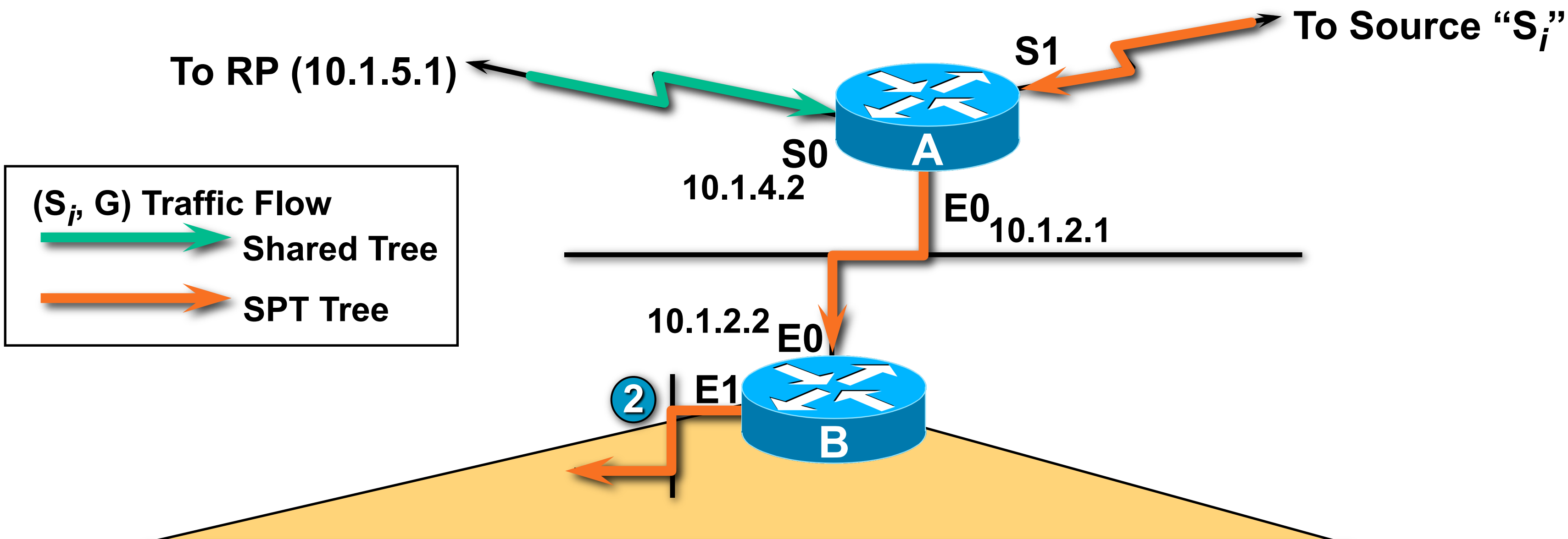



```
(*, 224.1.1.1), 00:01:43/00:00:00, RP 10.1.5.1, flags: SP
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:

(171.68.37.121, 224.1.1.1), 00:01:05/00:01:55, flags: CJPT
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1
  Outgoing interface list:
```

2 B removes E1 from (*,G) and all (S,G) *OILs*.

PIM SM Pruning (XR)

Source (SPT) Case

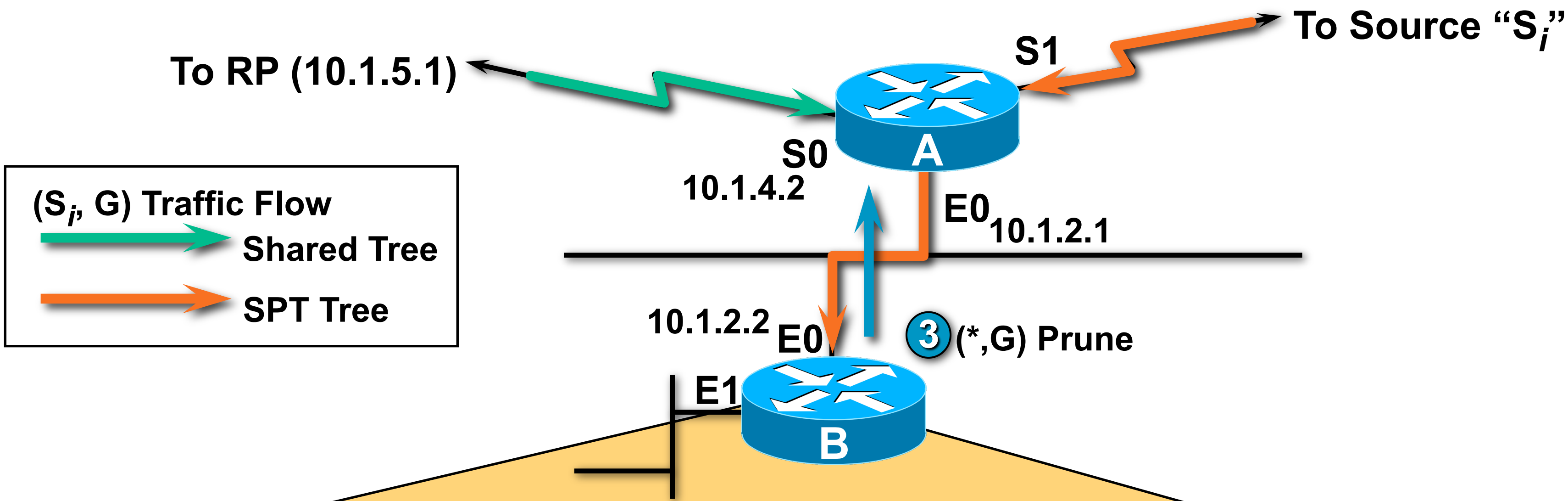



```
(*,224.1.1.1) RPF nbr: 10.1.2.1 Flags: C
  Incoming Interface List
    Ethernet0 Flags: A, Up: 17:37:56

(171.68.37.121,224.1.1.1) RPF nbr: 10.1.2.1 Flags:
  Incoming Interface List
    Ethernet0 Flags: A, Up: 00:14:15
```

2 B removes E1 from (*,G) and all (S,G) *OILs*.

PIM SM Pruning

Source (SPT) Case

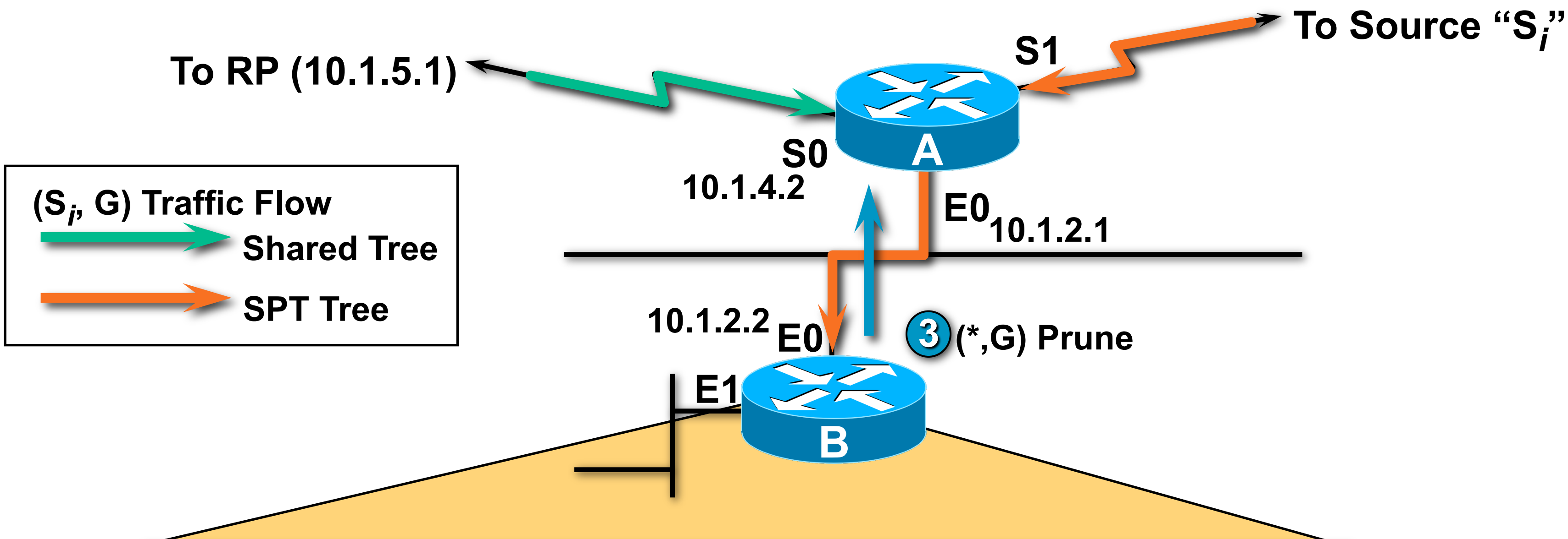



```
(*, 224.1.1.1), 00:01:43/00:00:00, RP 10.1.5.1, flags: SP
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:

(171.68.37.121, 224.1.1.1), 00:01:05/00:01:55, flags: CJPT
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1
  Outgoing interface list:
```

3 B's (*,G) OIL now empty; triggers (*,G) Prune toward RP.

PIM SM Pruning (XR)

Source (SPT) Case

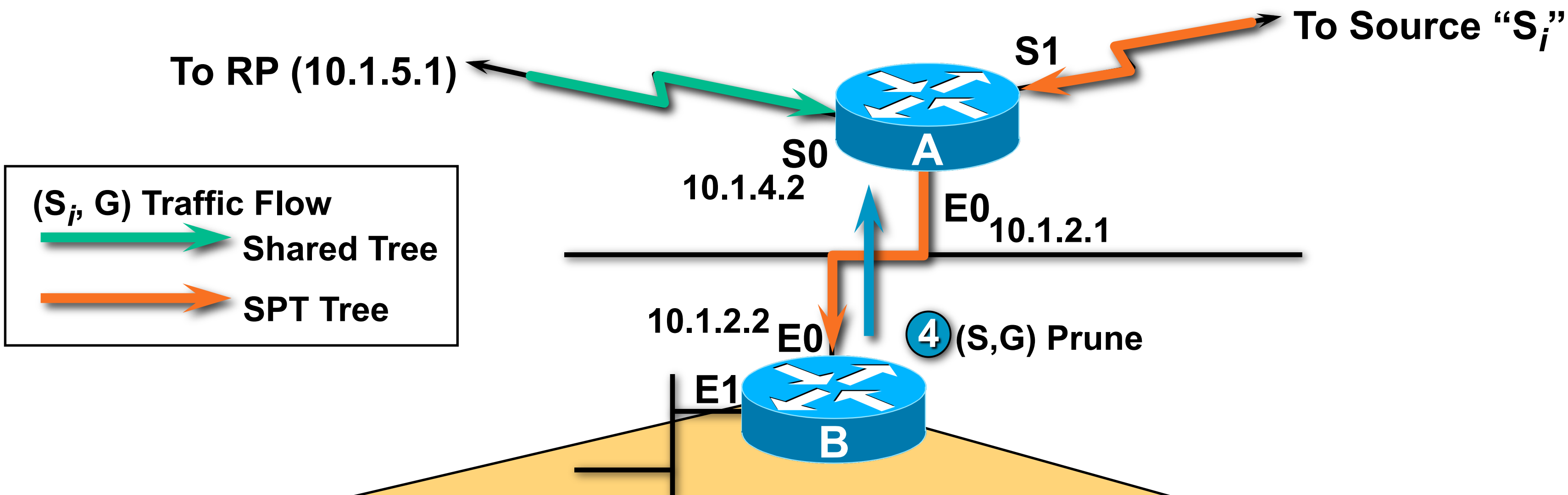



```
(192.2.1.2,224.1.1.1) RPF nbr: 10.1.2.1 Flags:
  Up: 00:14:15
  Incoming Interface List
    Ethernet0 Flags: A, Up: 00:14:15
```

3 B's (*,G) OIL now empty; triggers (*,G) Prune toward RP.

PIM SM Pruning

Source (SPT) Case

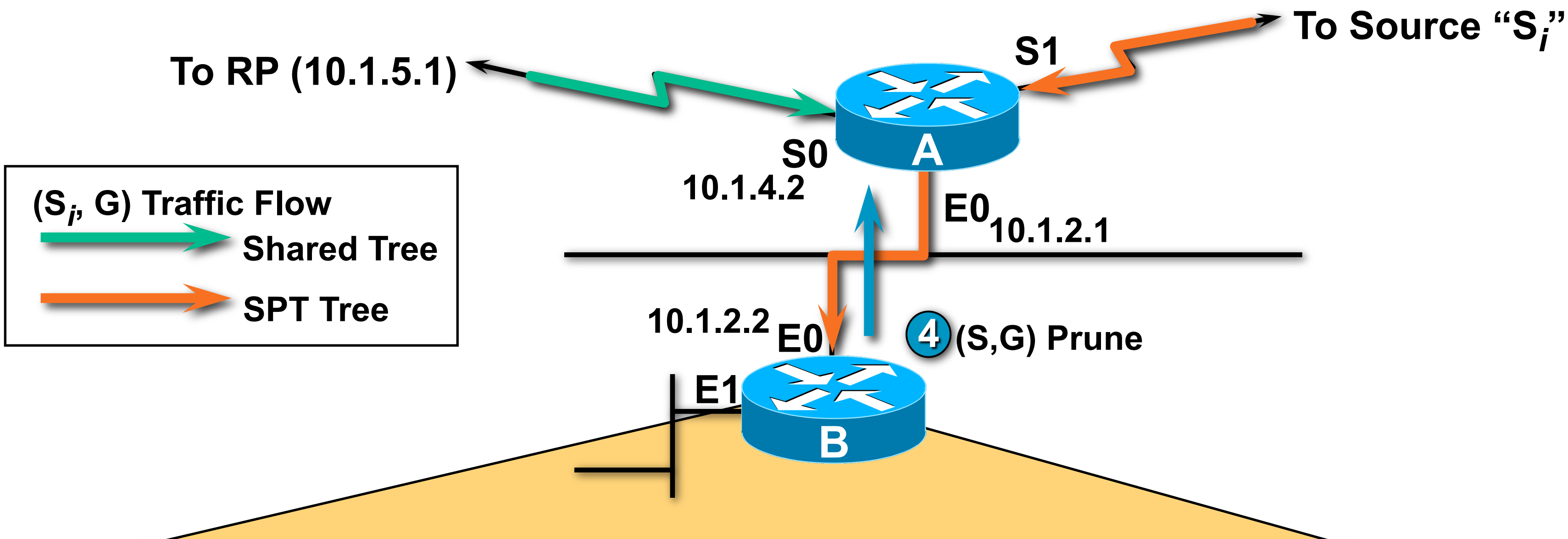



```
(*, 224.1.1.1), 00:01:43/00:00:00, RP 10.1.5.1, flags: SP
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:

(171.68.37.121, 224.1.1.1), 00:01:05/00:01:55, flags: CJPT
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1
  Outgoing interface list:
```

4 B's (S,G) OIL also now empty; triggers (S, G) Prune towards Si .

PIM SM Pruning (XR)

Source (SPT) Case

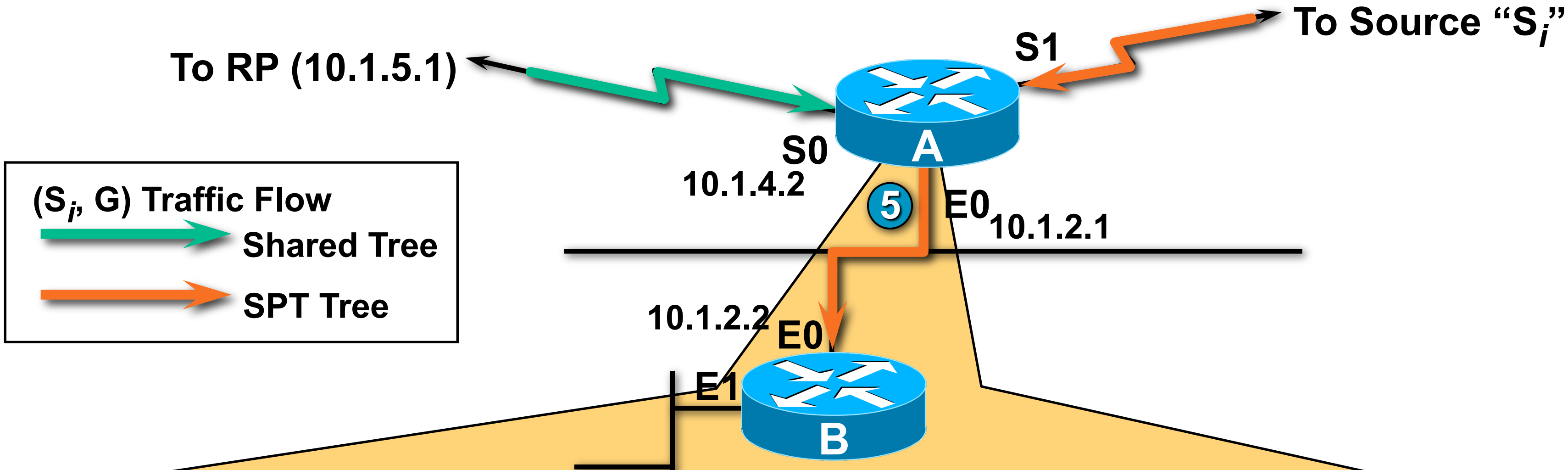



```
(192.2.1.2,224.1.1.1) RPF nbr: 10.1.2.1 Flags:
  Up: 00:14:15
  Incoming Interface List
    Ethernet0 Flags: A, Up: 00:14:15
```

4 **B's (S,G) OIL also now empty; triggers (S, G) Prune towards Si .**

PIM SM Pruning

Source (SPT) Case



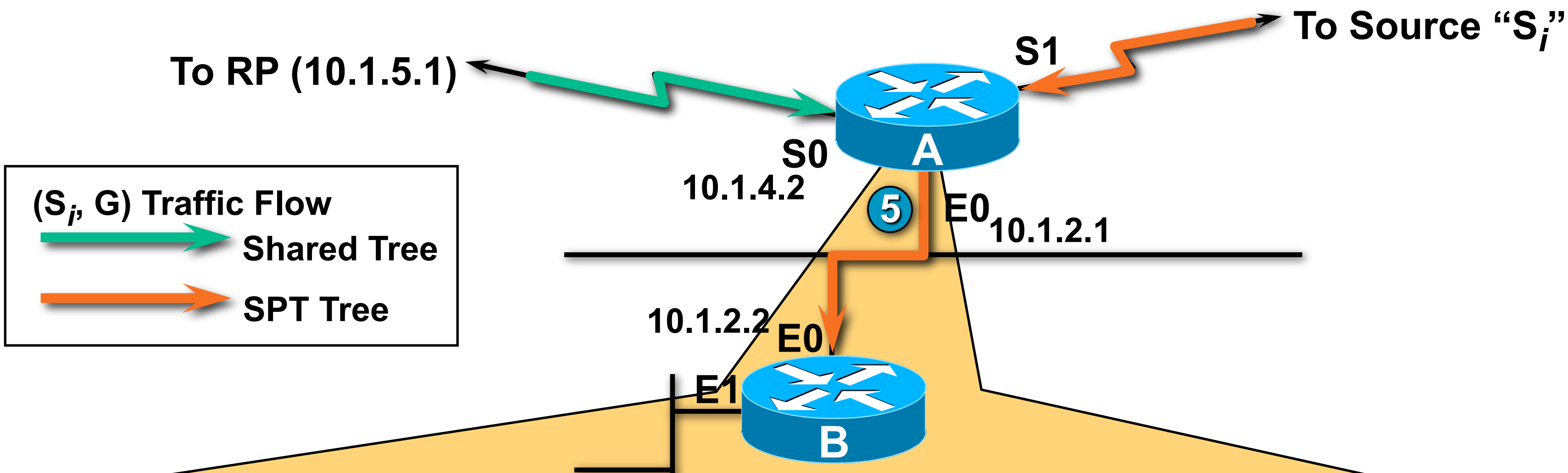

```
(*, 224.1.1.1), 00:02:32/00:00:00, RP 10.1.5.1, flags: SP
  Incoming interface: Serial0, RPF nbr 10.1.4.1,
  Outgoing interface list:

(171.68.37.121, 224.1.1.1), 00:01:56/00:00:53, flags: PT
  Incoming interface: Serial1, RPF nbr 10.1.9.2
  Outgoing interface list:
```

5 A receives (*, G) Prune; removes E0 from (*,G) & (S,G) OILs

(After the 3 second Multi-access Network Prune delay.)

PIM SM Pruning (XR)

Source (SPT) Case



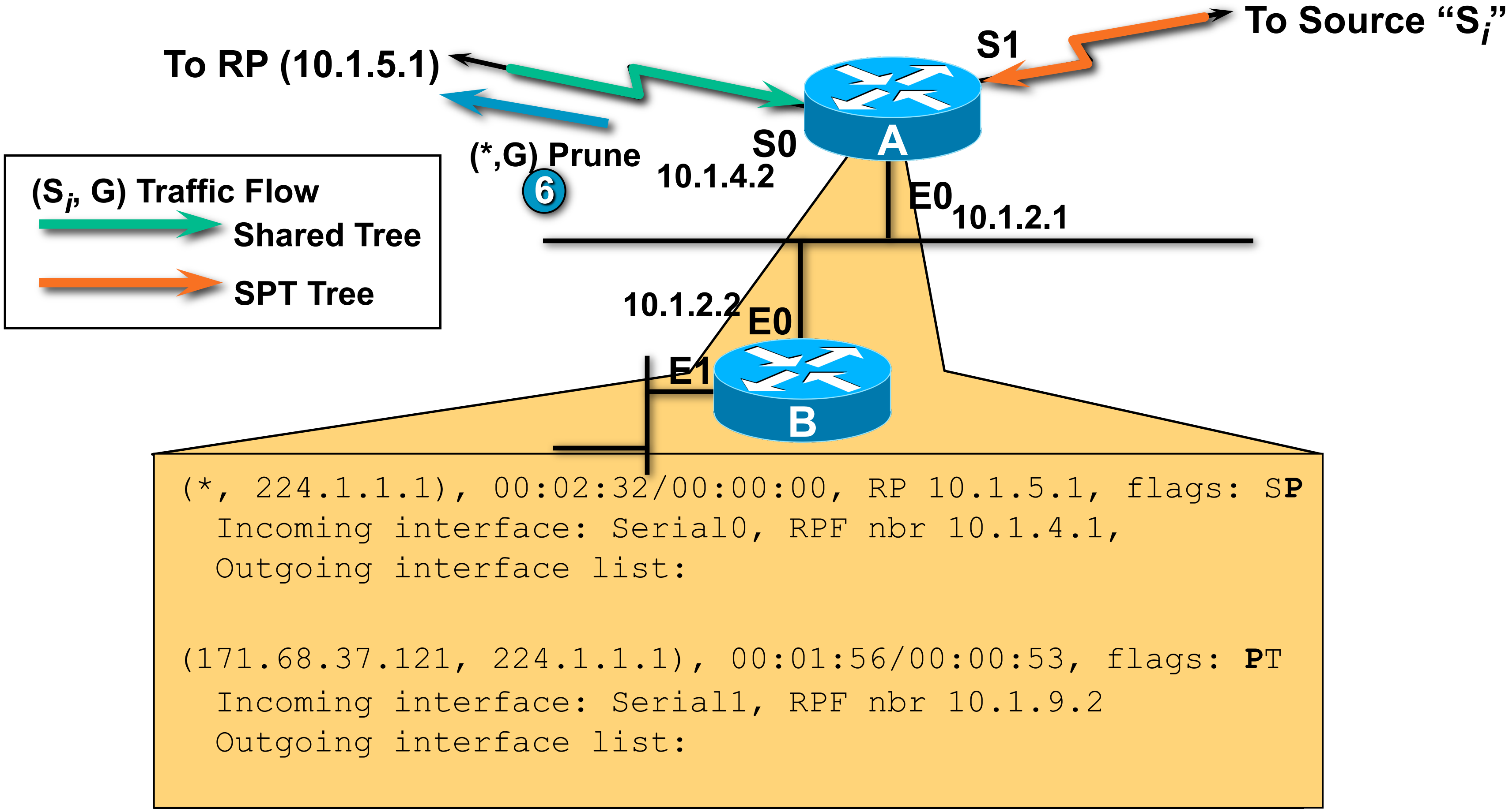

```
(*,224.1.1.1) RPF nbr: 10.1.4.1 Flags:
  Incoming Interface List
    Serial0 Flags: A, Up: 00:14:15

(192.2.1.2,224.1.1.1) RPF nbr: 10.1.9.2 Flags:
  Incoming Interface List
    Serial1 Flags: A, Up: 00:14:15
```

5 A receives (*, G) Prune; removes E0 from (*,G) & (S,G) OILs

(After the 3 second Multi-access Network Prune delay.)

PIM SM Pruning

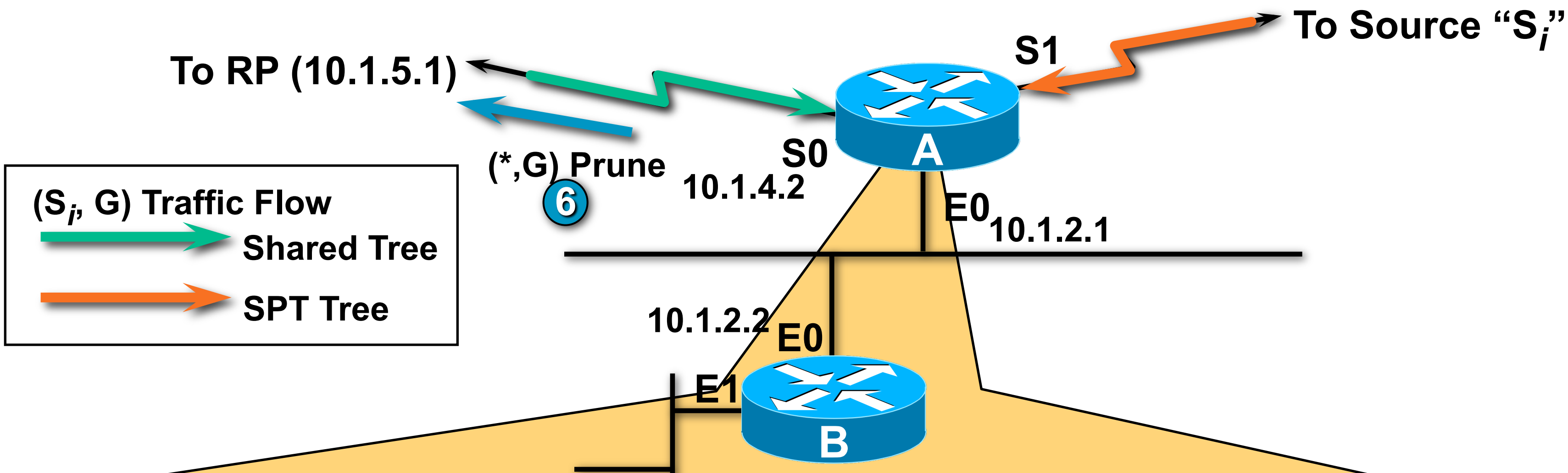
Source (SPT) Case



6 A's (*,G) *OIL* now empty; triggers (*,G) Prune toward RP.

PIM SM Pruning (XR)

Source (SPT) Case

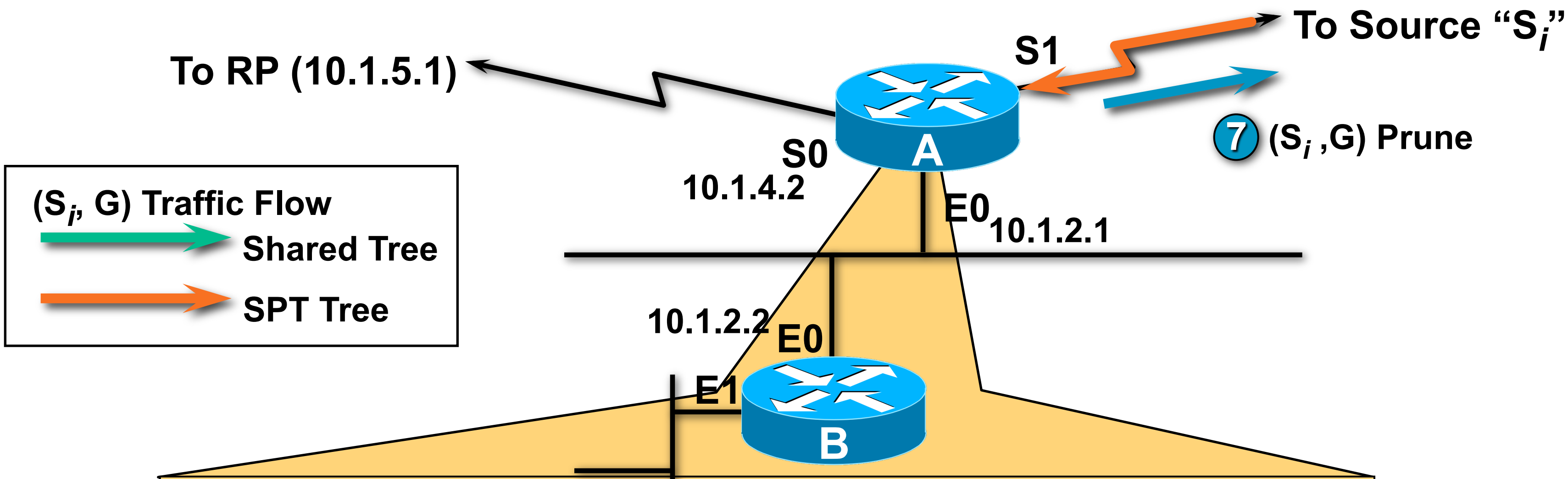



```
(*,224.1.1.1) RPF nbr: 10.1.4.1 Flags:
  Incoming Interface List
    Serial0 Flags: A, Up: 00:14:15

(192.2.1.2,224.1.1.1) RPF nbr: 10.1.9.2 Flags:
  Incoming Interface List
    Serial1 Flags: A, Up: 00:14:15
```

6 A's (*,G) *OIL* now empty; triggers (*,G) Prune toward RP.

PIM SM Pruning

Source (SPT) Case

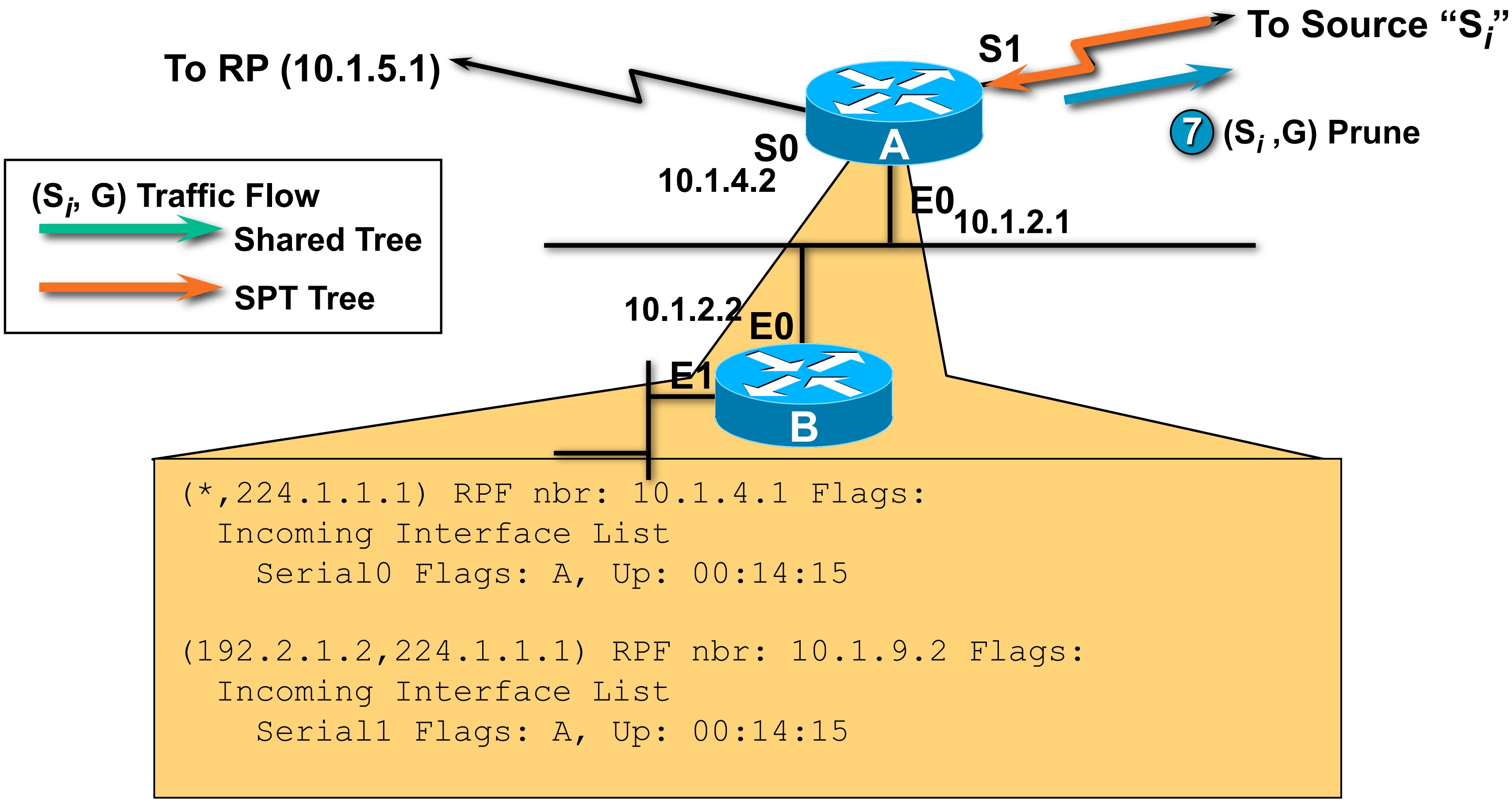



```
(*, 224.1.1.1), 00:02:32/00:00:00, RP 10.1.5.1, flags: SP
  Incoming interface: Serial0, RPF nbr 10.1.4.1,
  Outgoing interface list:

(171.68.37.121, 224.1.1.1), 00:01:56/00:00:53, flags: PT
  Incoming interface: Serial1, RPF nbr 10.1.9.2
  Outgoing interface list:
```

7 A's (S,G) *OIL* also now empty; triggers (S,G) Prune towards S_i.

PIM SM Pruning (XR)

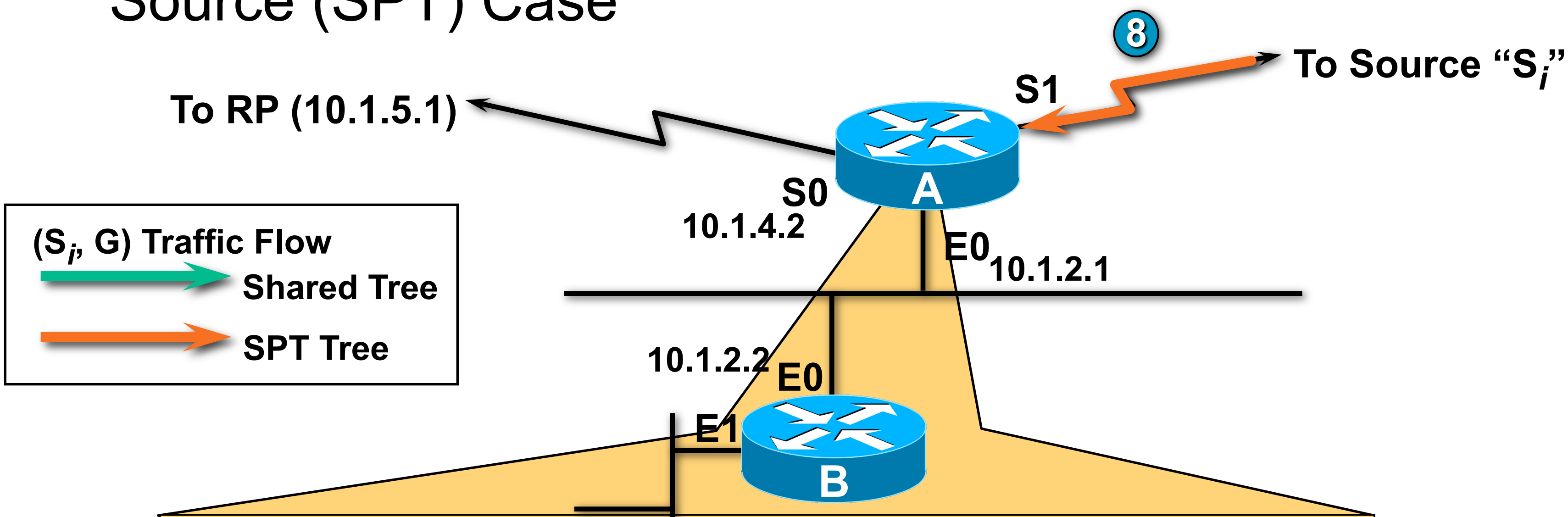
Source (SPT) Case



7 **A's (S,G) *OIL* also now empty; triggers (S,G) Prune towards S_i .**

PIM SM Pruning

Source (SPT) Case

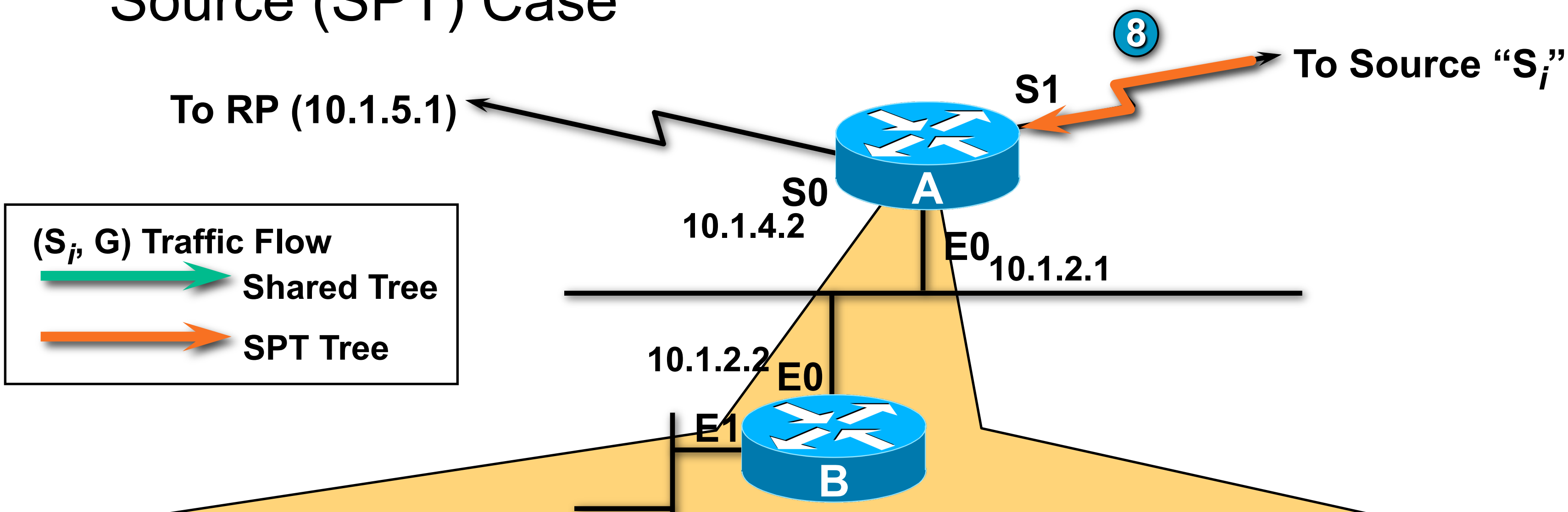



```
(*, 224.1.1.1), 00:02:32/00:00:00, RP 10.1.5.1, flags: SP
  Incoming interface: Serial0, RPF nbr 10.1.4.1,
  Outgoing interface list:

(171.68.37.121, 224.1.1.1), 00:01:56/00:00:53, flags: PT
  Incoming interface: Serial1, RPF nbr 10.1.9.2
  Outgoing interface list:
```

8 **(S_i,G) traffic ceases flowing down SPT.**

PIM SM Pruning (XR)

Source (SPT) Case




```
(*,224.1.1.1) RPF nbr: 10.1.4.1 Flags:
  Incoming Interface List
    Serial0 Flags: A, Up: 00:14:15

(192.2.1.2,224.1.1.1) RPF nbr: 10.1.9.2 Flags:
  Incoming Interface List
    Serial1 Flags: A, Up: 00:14:15
```

8 (S_i,G) traffic ceases flowing down SPT.

Recap: Common Multicast Flags—IOS

- S: Sparse Mode (in contrast to D for Dense Mode)
- s: SSM; only seen on (S,G) entries
- B: Bidir
- F: Register; set on first-hop router
- P: Prune; entry has an empty OIL
- J: Join-SPT; (*,G) traffic exceeds SPT Threshold
- T: SPT; set on (S,G) entries after SPT join
- L: Local; router should receive and process this traffic
- C: Connected; seen primarily with IGMP

Recap: Common Multicast Flags—IOS XR

- F: Forward traffic for the entry on this interface
- A: Accept - RPF interface; F & A are mutually exclusive
- DI: Decap Tunnel; for Register msgs on RP
- EI: Encap Tunnel; for Register msgs on first hop router
- C: Connected; perform check on incoming packets
- NS: Negate Signal; do not signal PIM (internal)
- SP: Signal Present; MFIB signaling to PIM (internal)
- LI: Local Interest; received IGMP report/join for that group
- LD: Local Disinterest; received IGMP join excluding (S,G)
- II: Internal interest; router host stack interested

Source Specific Multicast

Barriers to Multicast Deployment

- Global Multicast Address Allocation
  - Dynamic Address Allocation
    - No adequate dynamic address allocation methods exist
    - SDR—Doesn’t scale
    - MASC—Long ways off!
  - Static Address Allocation (GLOP)
    - Based on AS number
    - Insufficient address space for large Content Providers
- Multicast Content “Jammers”
  - Undesirable sources on a multicast group
    - “Capt. Midnight” sources bogus data/noise to group
    - Can cause DoS attack by congesting low speed links

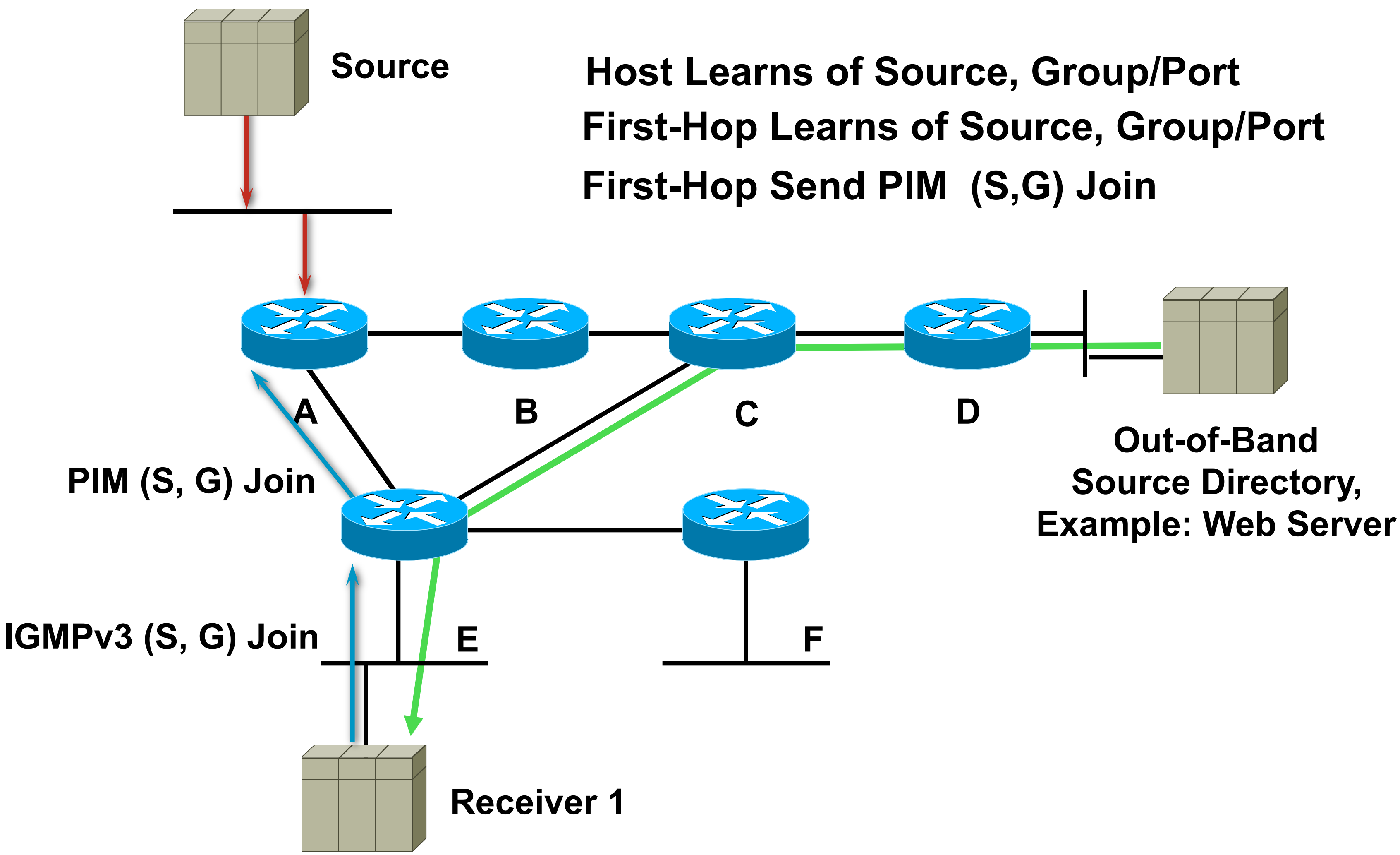
Source Specific Multicast (SSM)

- Uses Source Trees only
- Assumes one-to-many model
  - Most Internet multicast fits this model
  - IP/TV also fits this model
- Hosts responsible for source discovery
  - Typically via some out-of-band mechanism
    - Web page, Content Server, etc.
  - Eliminates need for RP and Shared Trees
  - Eliminates need for MSDP

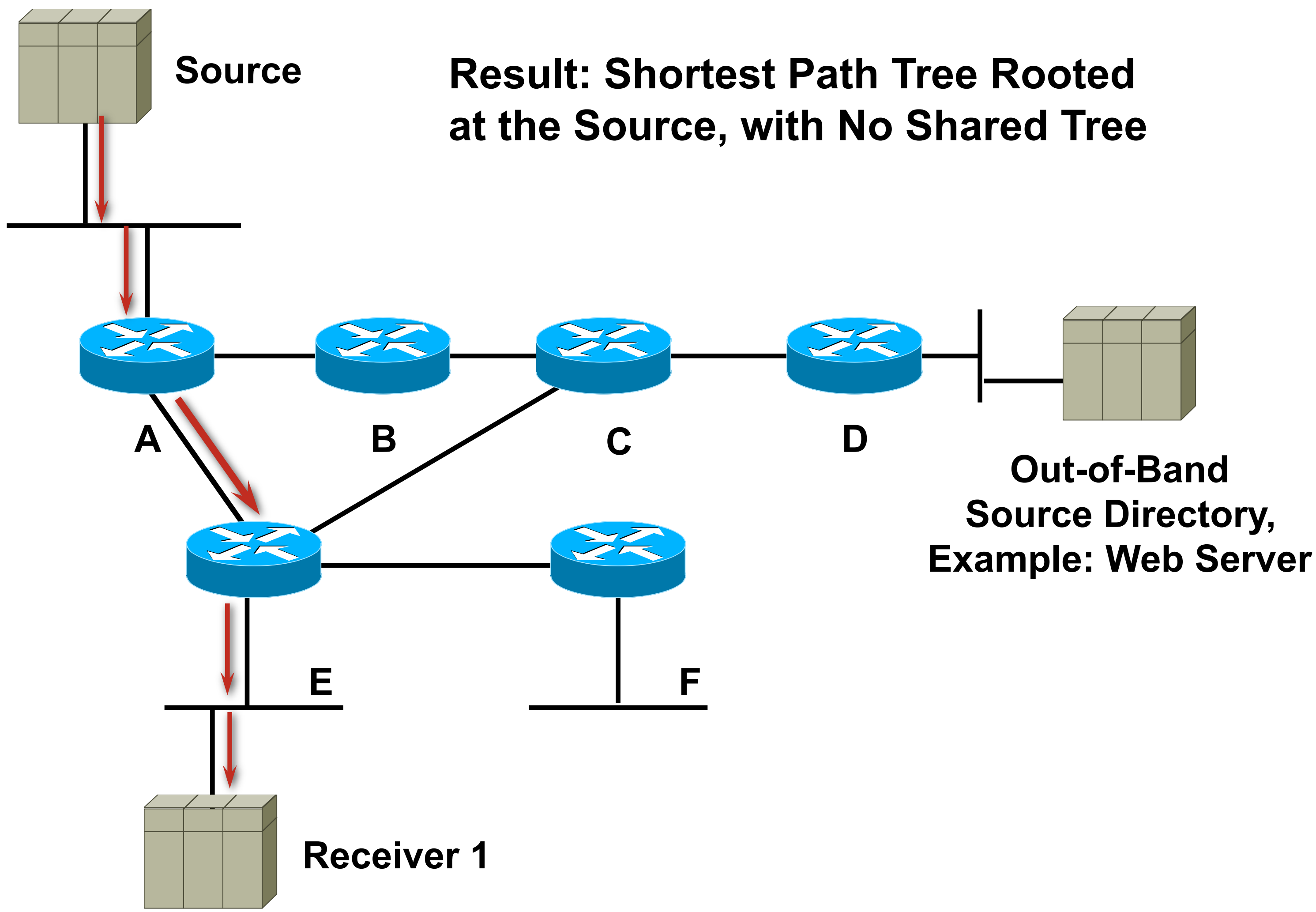
SSM Overview

- Hosts join a specific source within a group
  - Content identified by specific (S,G) instead of (*,G)
  - Hosts responsible for learning (S,G) information
- Last-hop router sends (S,G) join toward source
  - Shared Tree is never Joined or used
  - Eliminates possibility of content Jammers
  - Only specified (S,G) flow is delivered to host
- Eliminates Networked-Based Source Discovery
  - No RPs for SSM groups
- Simplifies address allocation
  - Dissimilar content sources can use same group without fear of interfering with each other

SSM Example


Source
Host Learns of Source, Group/Port
First-Hop Learns of Source, Group/Port
First-Hop Send PIM (S,G) Join
A
B
C
D
Out-of-Band
Source Directory,
Example: Web Server
PIM (S, G) Join
IGMPv3 (S, G) Join
E
F
Receiver 1

SSM Example


Source
Result: Shortest Path Tree Rooted at the Source, with No Shared Tree
A
B
C
D
Out-of-Band Source Directory, Example: Web Server
E
F
Receiver 1

SSM Configuration

- Global command

  `ip pim ssm {default | range <acl>}`

  Defines SSM address range

    Default range = 232.0.0.0/8

    Use ACL for other ranges

  Prevents Shared Tree Creation

    (*, G) Joins never sent or processed

    PIM Registers never sent or processed

  Available in Cisco IOS versions

    12.1(5)T, 12.2, 12.0(15)S, 12.1(8)E

SSM—Summary

- Uses Source Trees only
  - Hosts are responsible for source and group discovery
  - Hosts must signal router which (S,G) to join
- Solves multicast address allocation problems
  - Flows differentiated by **both** source and group
  - Content providers can use same group ranges
    - Since each (S,G) flow is unique
- Helps prevent certain DoS attacks
  - “Bogus” source traffic:
    - Can’t consume network bandwidth
    - Not received by host application

Bidirectional (BiDir) PIM

Multicast Application Categories

- One-to-many applications
  - Video, TV, radio, concerts, stock ticker, etc.
- Few-to-few applications
  - Small (<10 member) video/audio conferences
- Few-to-many applications
  - TIBCO RV servers (publishing)
- Many-to-many applications
  - Stock trading floors, gaming
- Many-to-few applications
  - TIBCO RV clients (subscriptions)

Multicast Application Categories

PIM-SM (S, G) State

- One-to-many applications
  - Single (S,G) entry
- Few-to-few applications
  - Few (<10 typical) (S,G) entries
- Few-to-many applications
  - Few (<10 typical) (S,G) entries
- Many-to-many applications
  - **Unlimited (S,G) entries**
- Many-to-few applications
  - **Unlimited (S,G) entries**

Many-to-Any State Problem

- Creates huge amounts of (S,G) state
  - State maintenance workloads skyrocket
    - High OIL fan-outs make the problem worse
  - Router performance begins to suffer
- Using Shared-Trees only
  - Provides some (S,G) state reduction
    - Results in (S,G) state only along SPT to RP
    - Frequently still too much (S,G) state
    - Need a solution that only uses (*,G) state

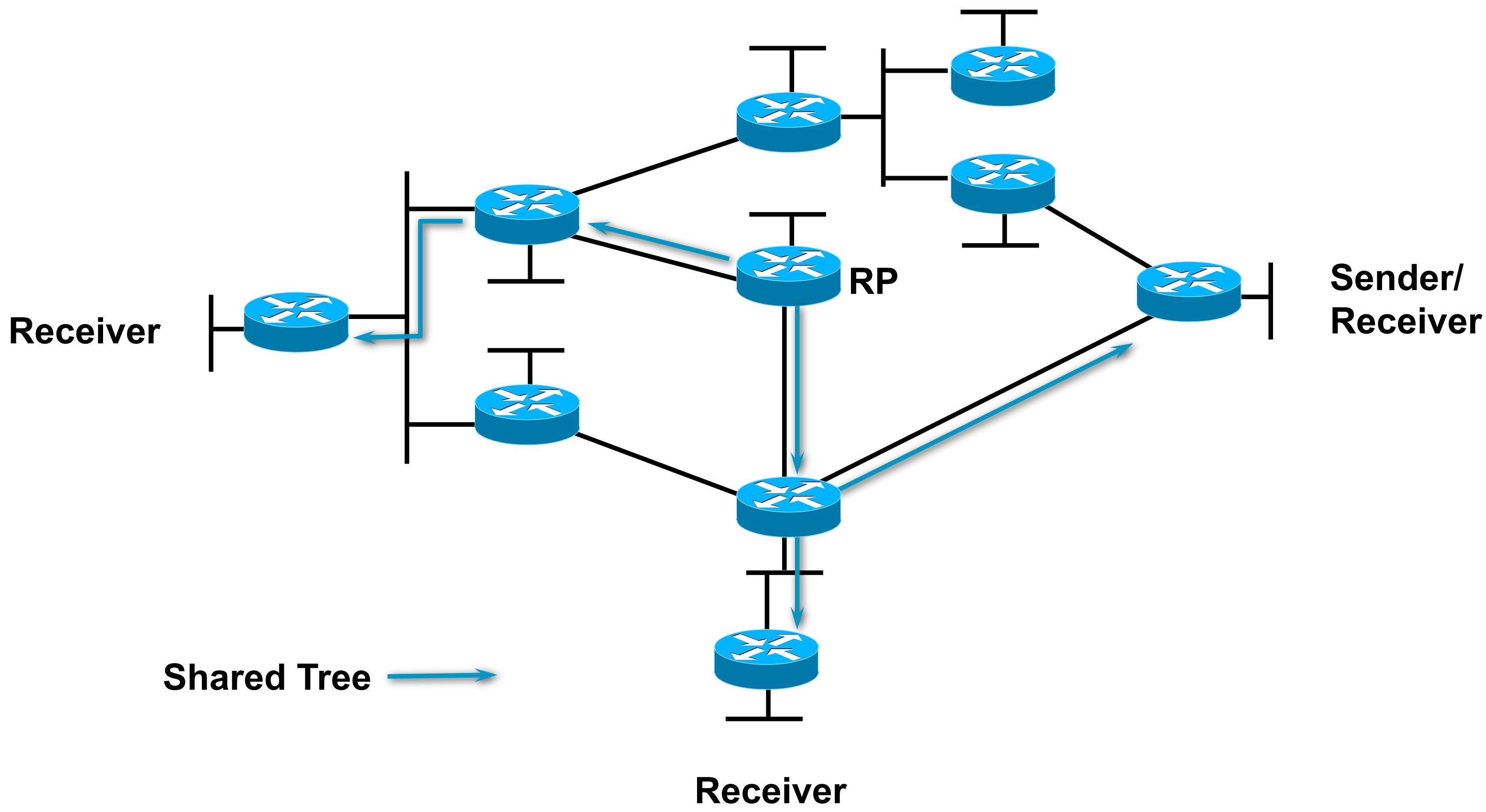
Bidirectional (BiDir) PIM

- Idea:
  - Use the same tree for traffic from sources towards RP and from RP to receivers
- Benefits:
  - Less state in routers
    - Only (*, G) state is used
    - Source traffic follows the Shared Tree
      - Flows up the Shared Tree to reach the RP
      - Flows down the Shared Tree to reach all other receivers

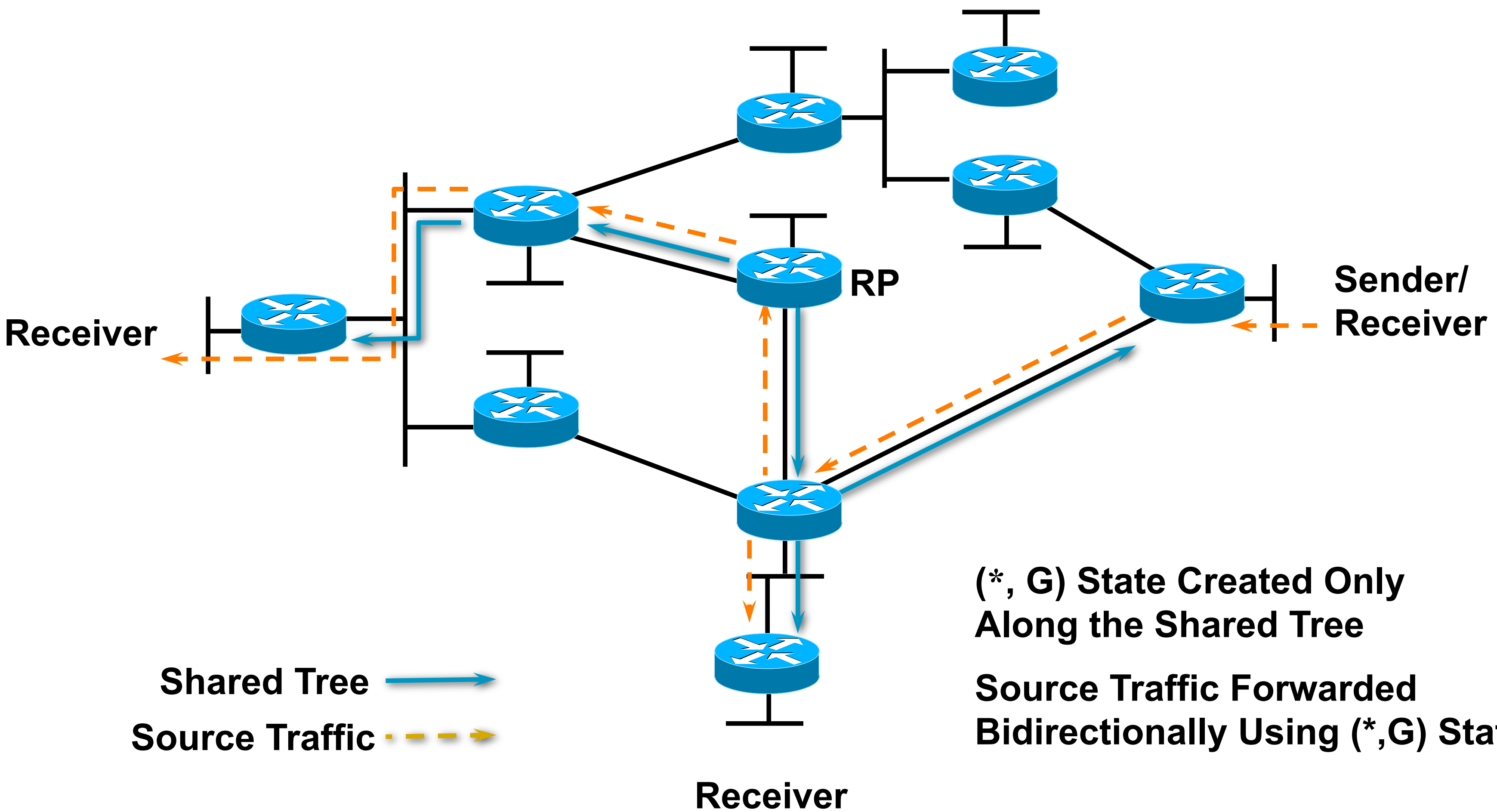
Bidirectional (BiDir) PIM

- Bidirectional Shared-Trees
  - Violates current (*,G) RPF rules
    - Traffic often accepted on **outgoing** interfaces
    - Care must be taken to avoid multicast loops
  - Requires a Designated Forwarder (DF)
    - Responsible for forwarding traffic up Shared Tree
      - DFs will accept data on the interfaces in their OIL
      - Then send it out all other interfaces (including the IIF)

Bidirectional PIM—Overview



Bidirectional PIM—Overview



(*, G) State Created Only Along the Shared Tree

Source Traffic Forwarded Bidirectionally Using (*,G) State

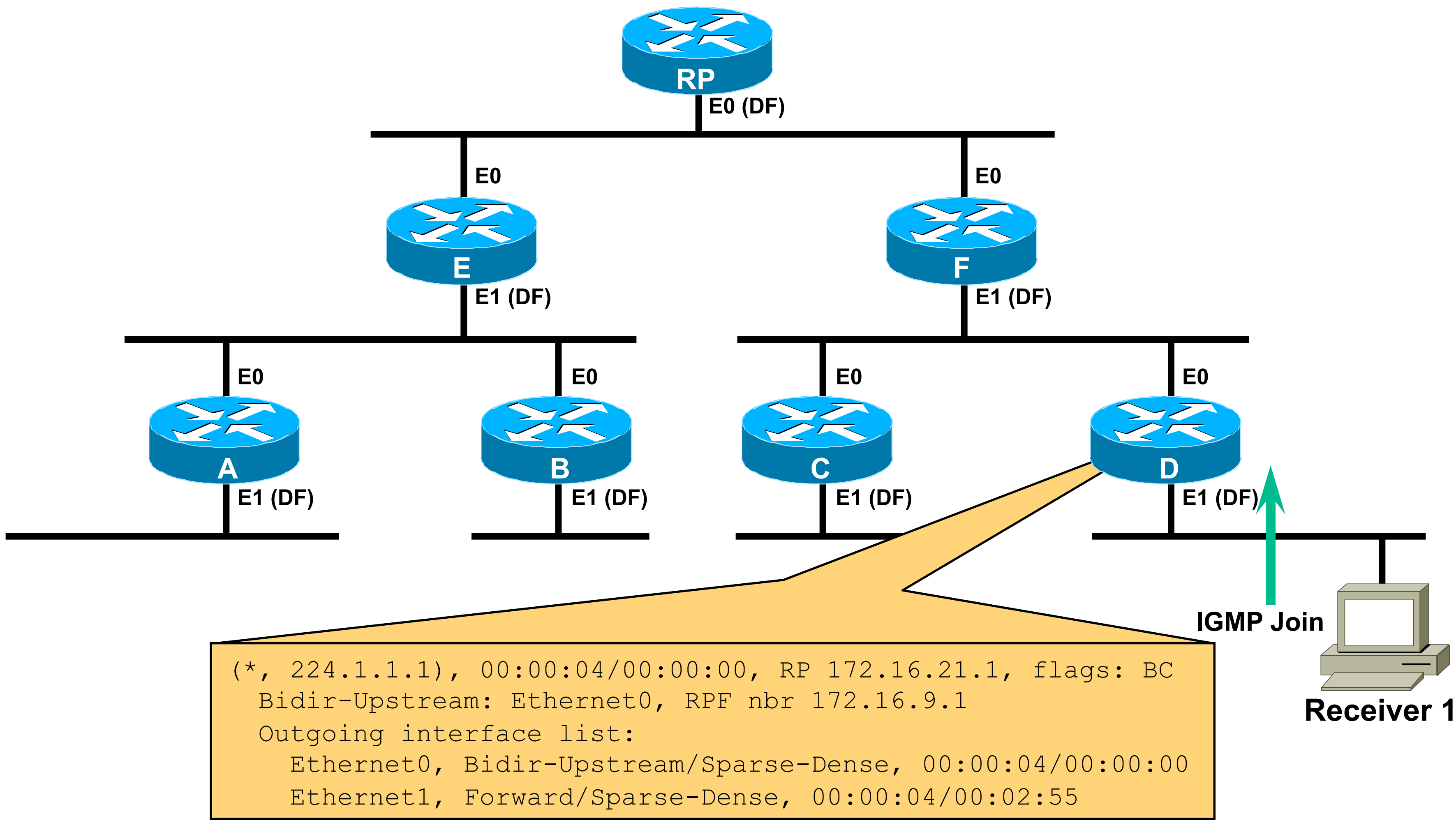
PIM Modifications for BiDir Operation

- Designated Forwarders (DF)
  - One DF per link
  - Router with best path to the RP is elected DF
    - Note: Designated Routers (DR) are not used for bidir groups
  - In addition to normal (*,G) forwarding rules:
    - Accepts traffic on outgoing interfaces
    - Forwards traffic out all other interfaces

Designated Forwarder Election

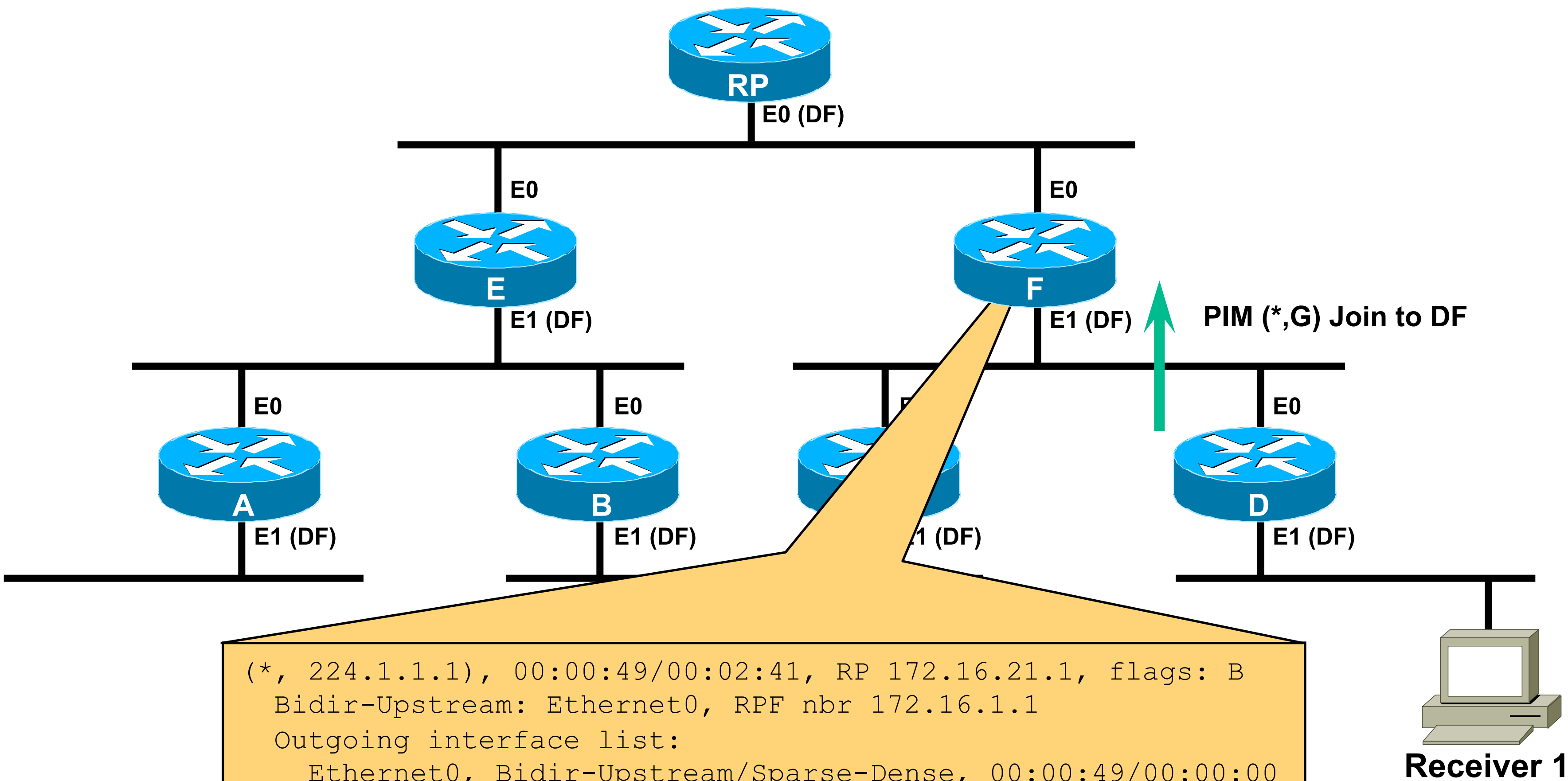
- Automatically performed on every link
  - When Bidir Group-range/RP is learned or configured
  - Router with the best path to the RP elected DF
  - Uses assert-like metric comparison to pick best path
- Purpose:
  - Ensures all routers on link agree on who is DF
    - Prevents route loops from forming

Forwarding/Tree Building



Receiver 1 Joins Group Causing Router "D" to Create (*, G) State

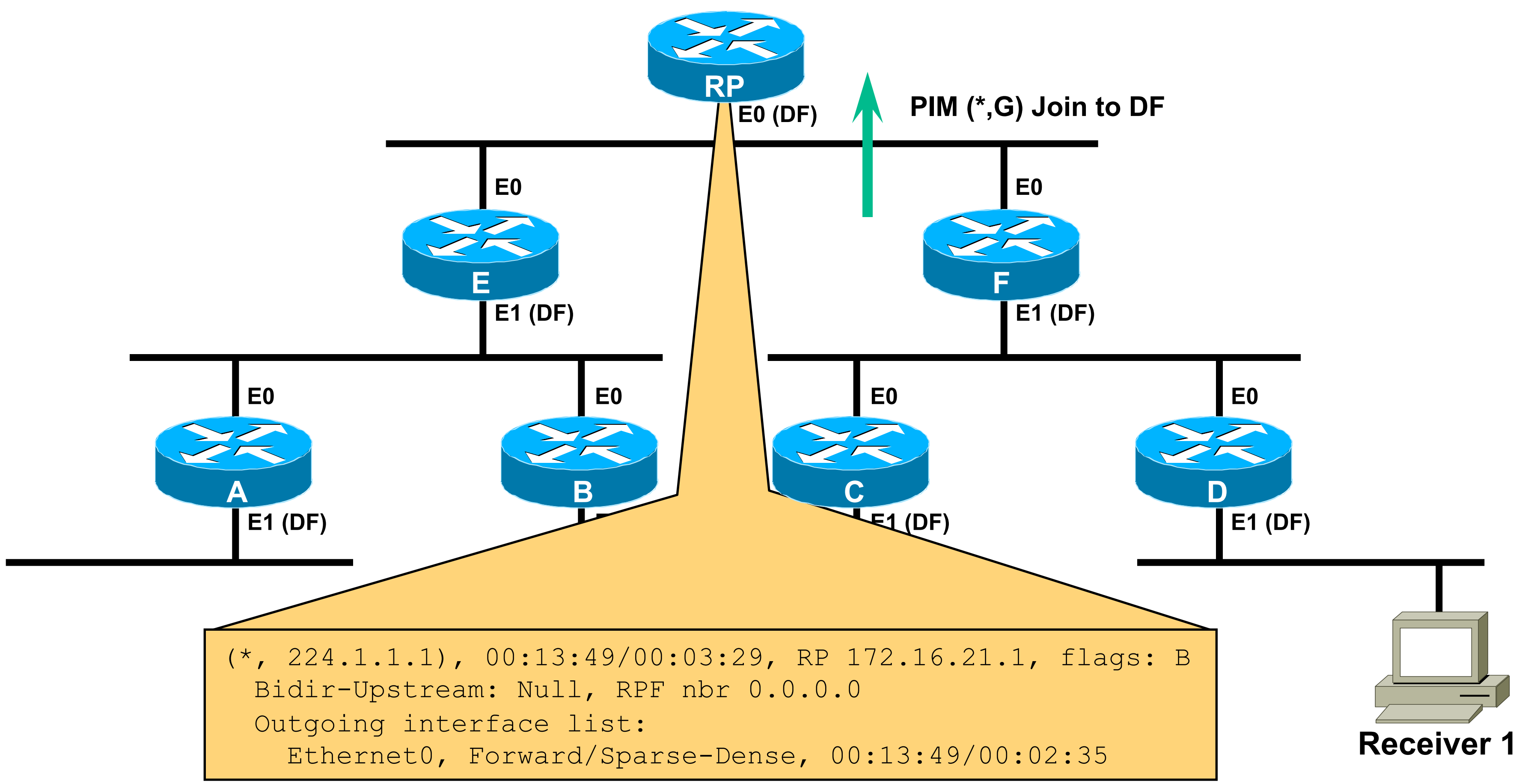
Forwarding/Tree Building




```
(*, 224.1.1.1), 00:00:49/00:02:41, RP 172.16.21.1, flags: B
  Bidir-Upstream: Ethernet0, RPF nbr 172.16.1.1
  Outgoing interface list:
    Ethernet0, Bidir-Upstream/Sparse-Dense, 00:00:49/00:00:00
    Ethernet1, Forward/Sparse-Dense, 00:00:49/00:02:41
```

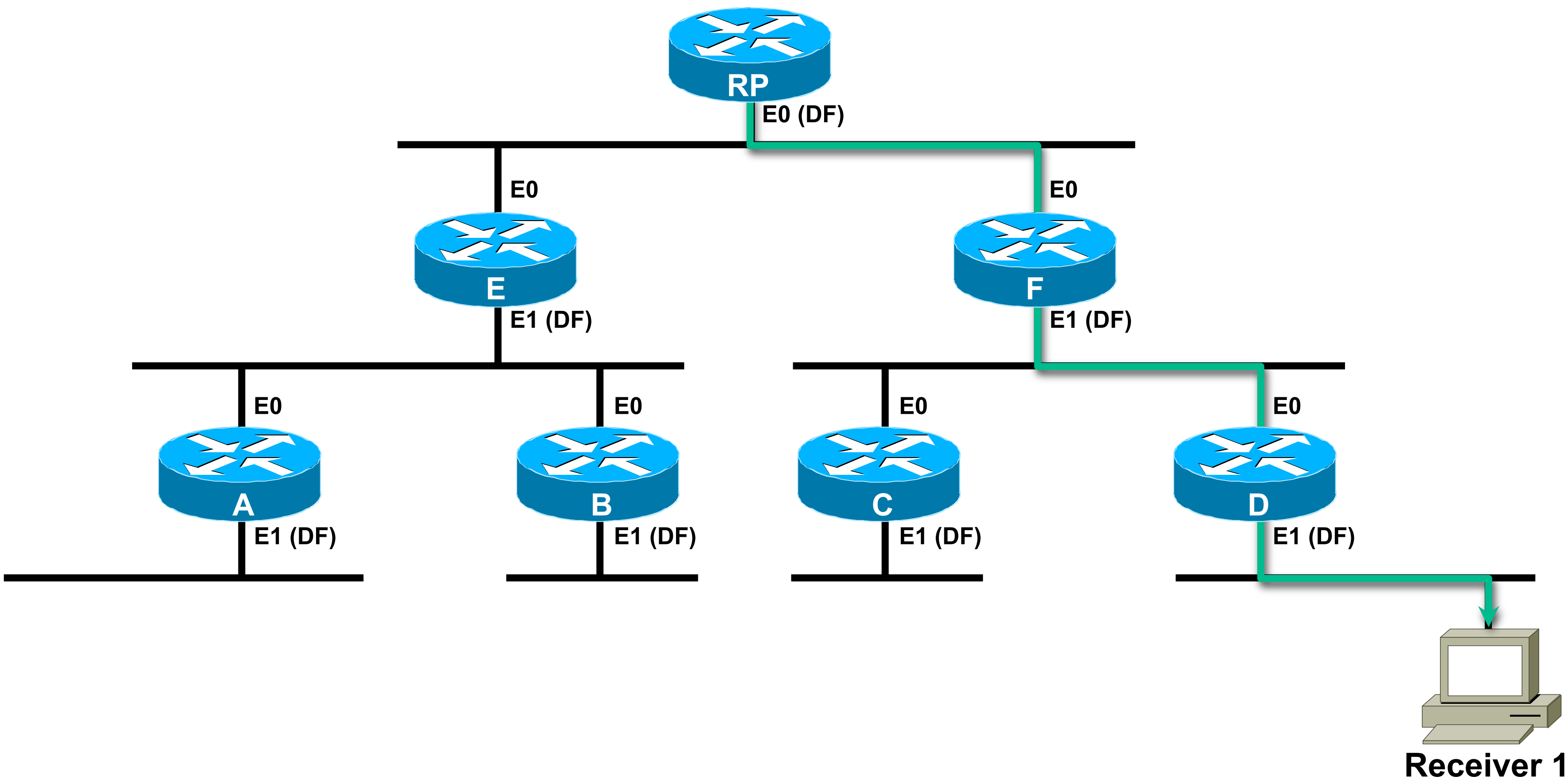
Router “D” Sends (*, G) Join to Router “F” (DF) Causing It to Create (*, G) State

Forwarding/Tree Building



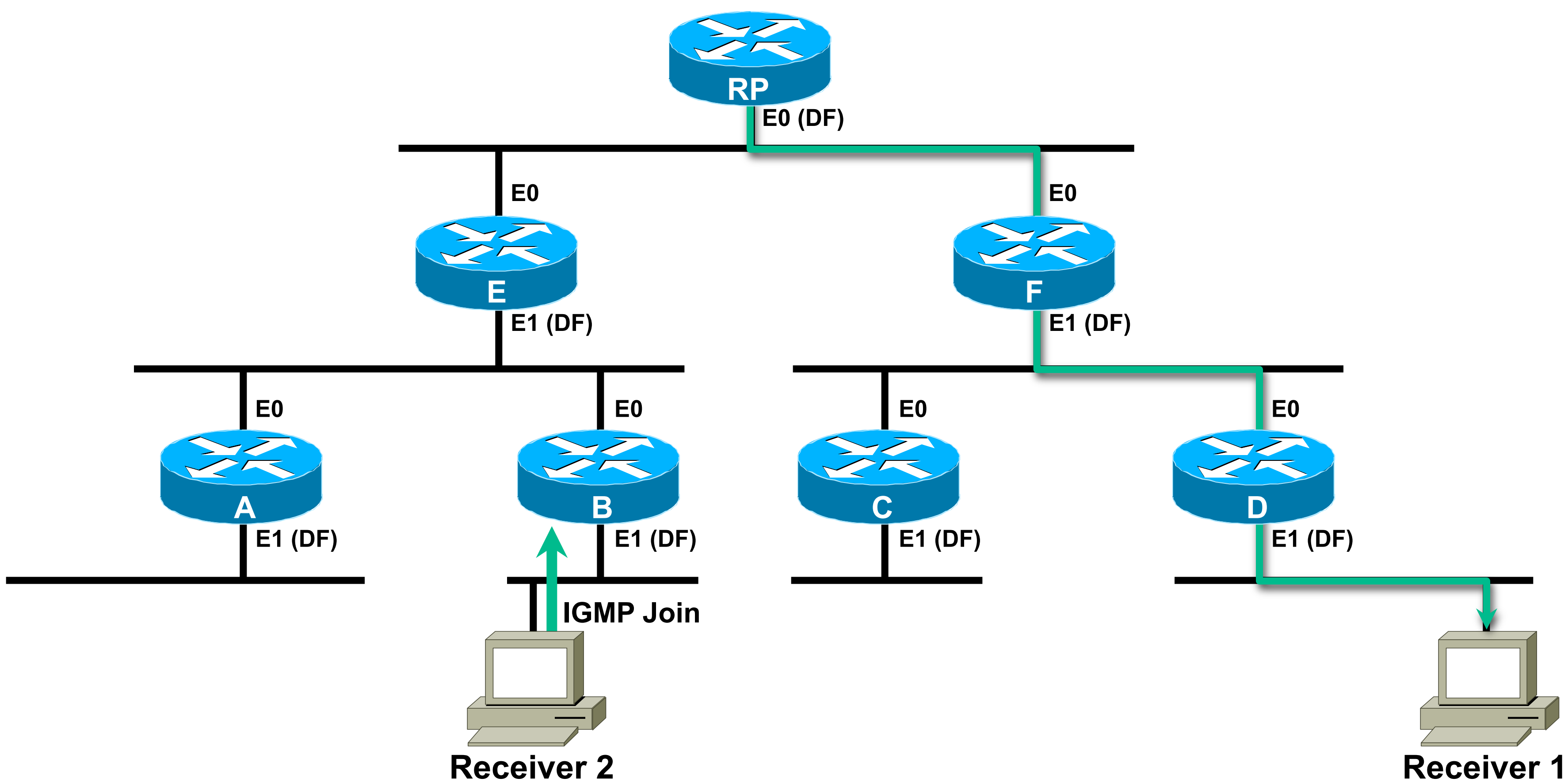
Router “F” Sends (*, G) Join to “RP” Causing It to Create (*, G) State

Forwarding/Tree Building



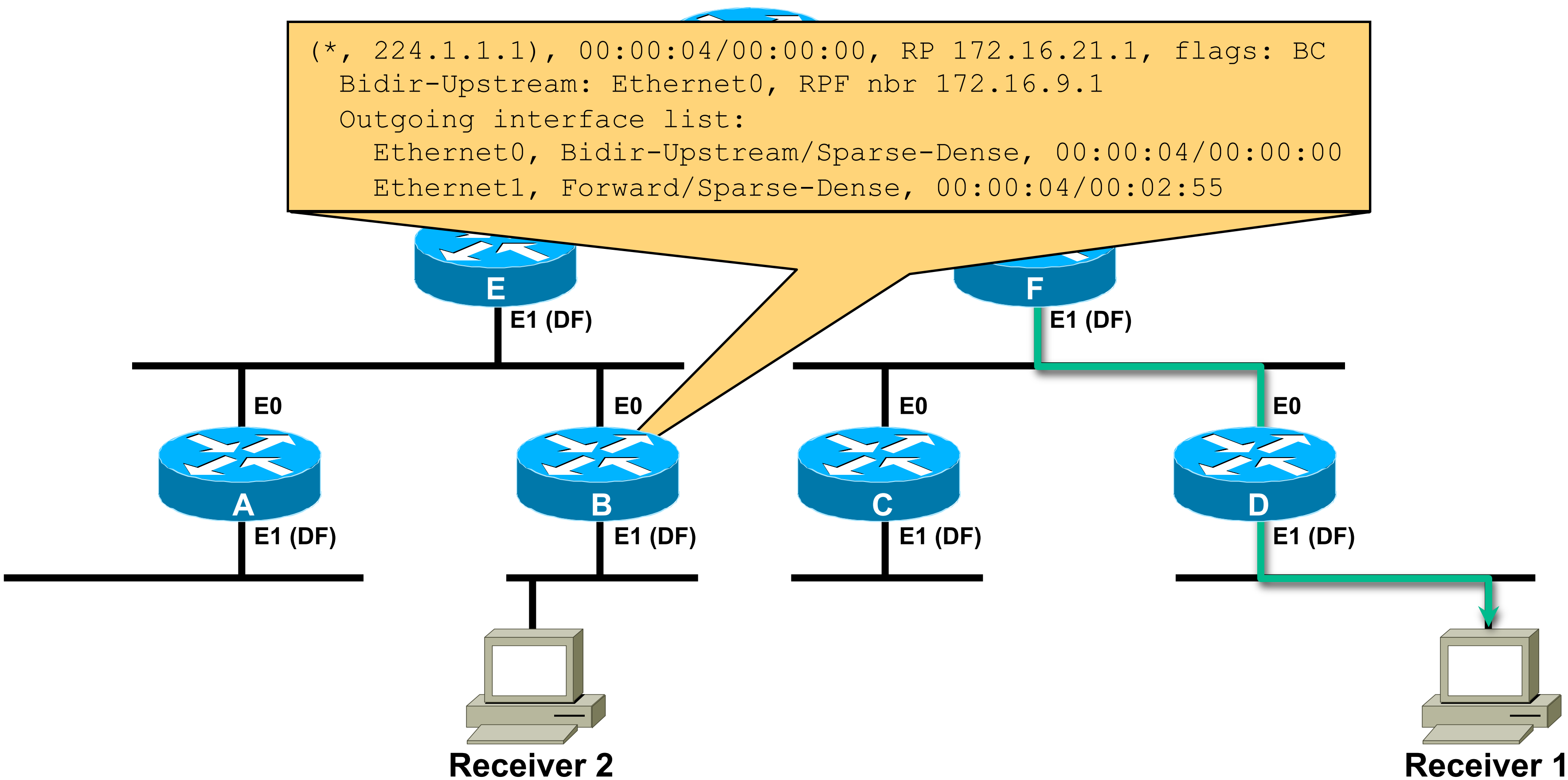
Branch of Shared Tree Is Now Built Down to Receiver 1

Forwarding/Tree Building



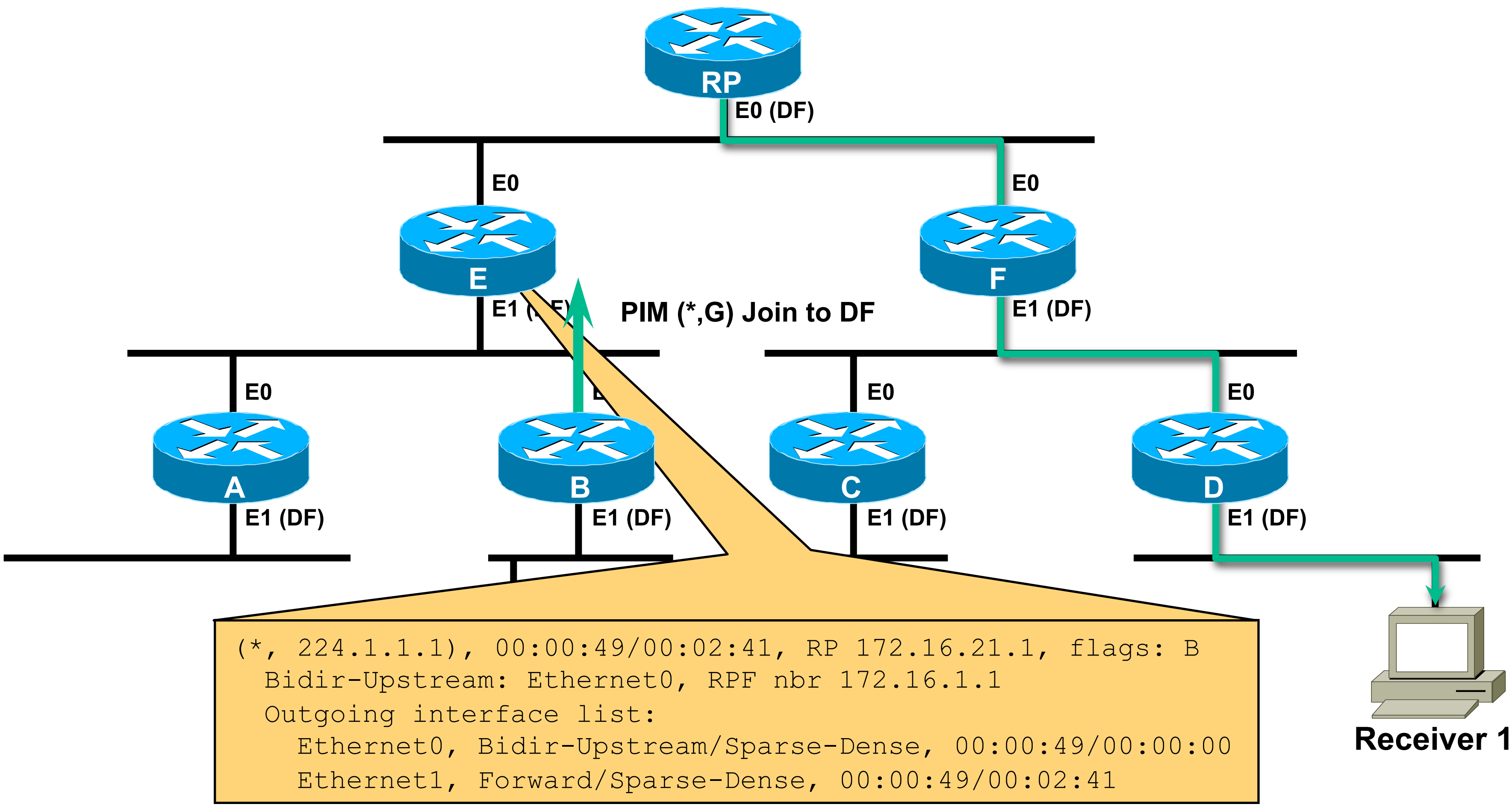
Receiver 2 Also Joins Group

Forwarding/Tree Building



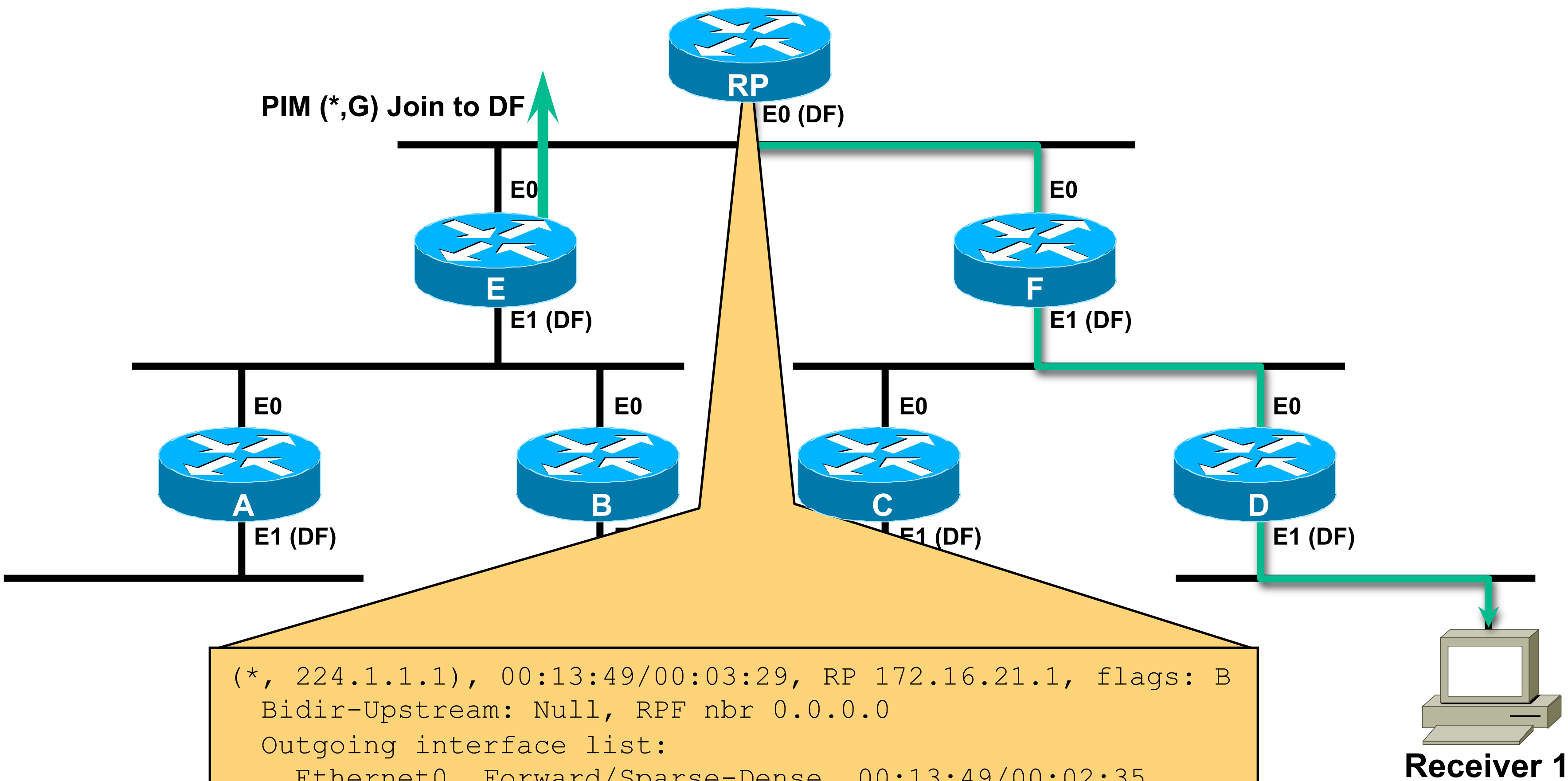
Router "B" Creates (*, G) State

Forwarding/Tree Building



Router "B" Sends (*, G) Join to "E" (DF) Causing It to Create (*, G) State

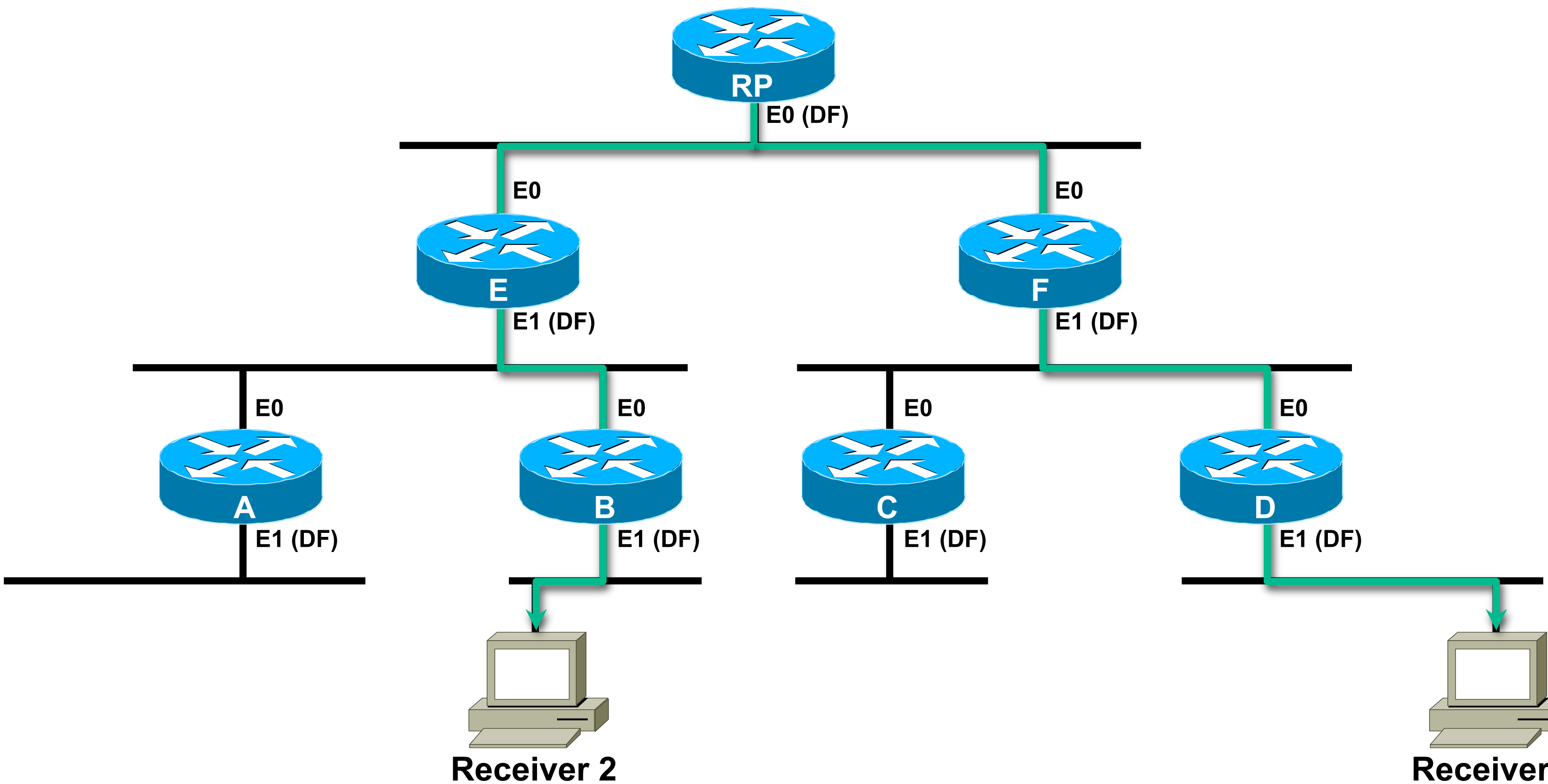
Forwarding/Tree Building




```
(*, 224.1.1.1), 00:13:49/00:03:29, RP 172.16.21.1, flags: B
  Bidir-Upstream: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:13:49/00:02:35
```

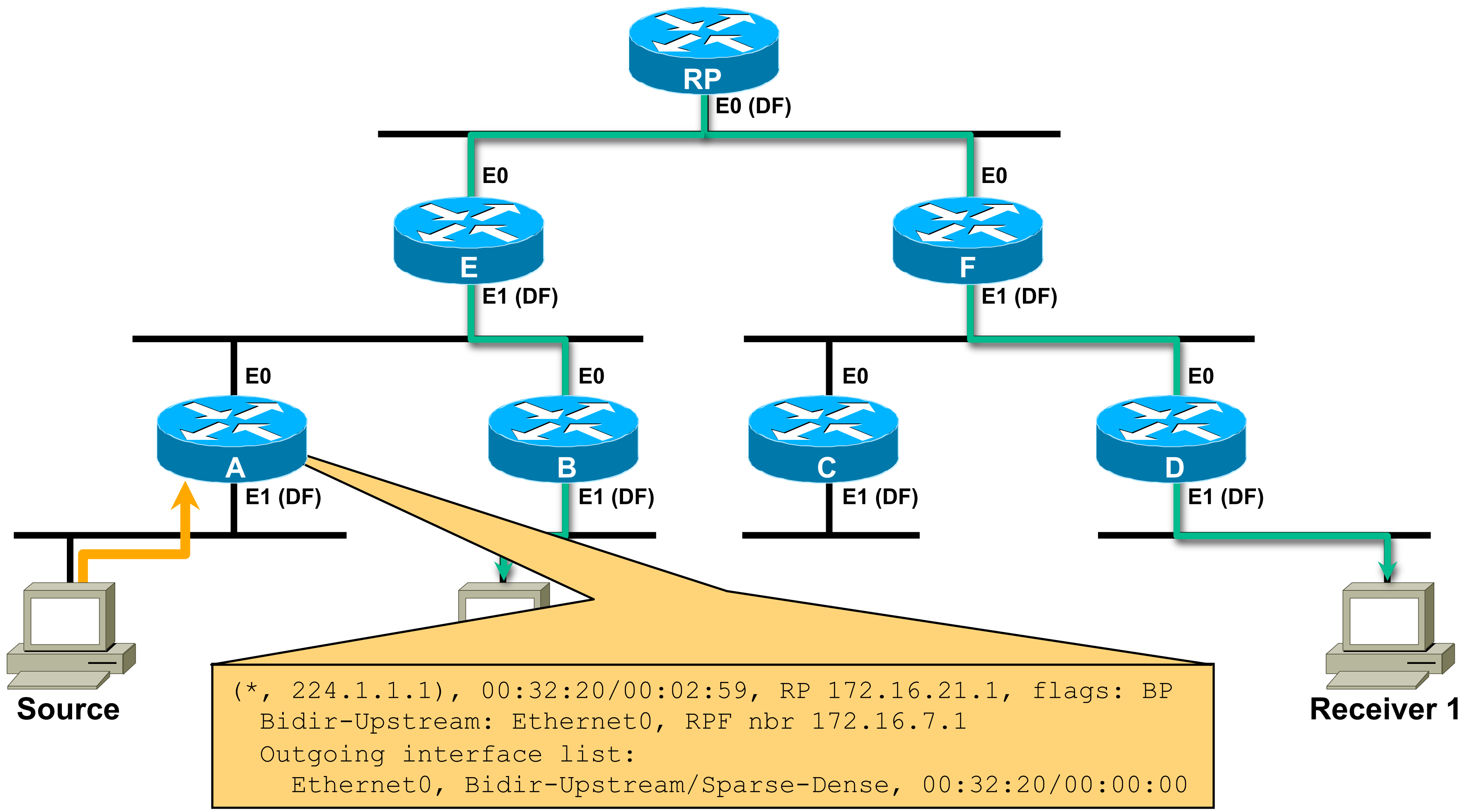
Router "E" Sends (*, G) Join to "RP" (State on RP Remains Unchanged)

Forwarding/Tree Building



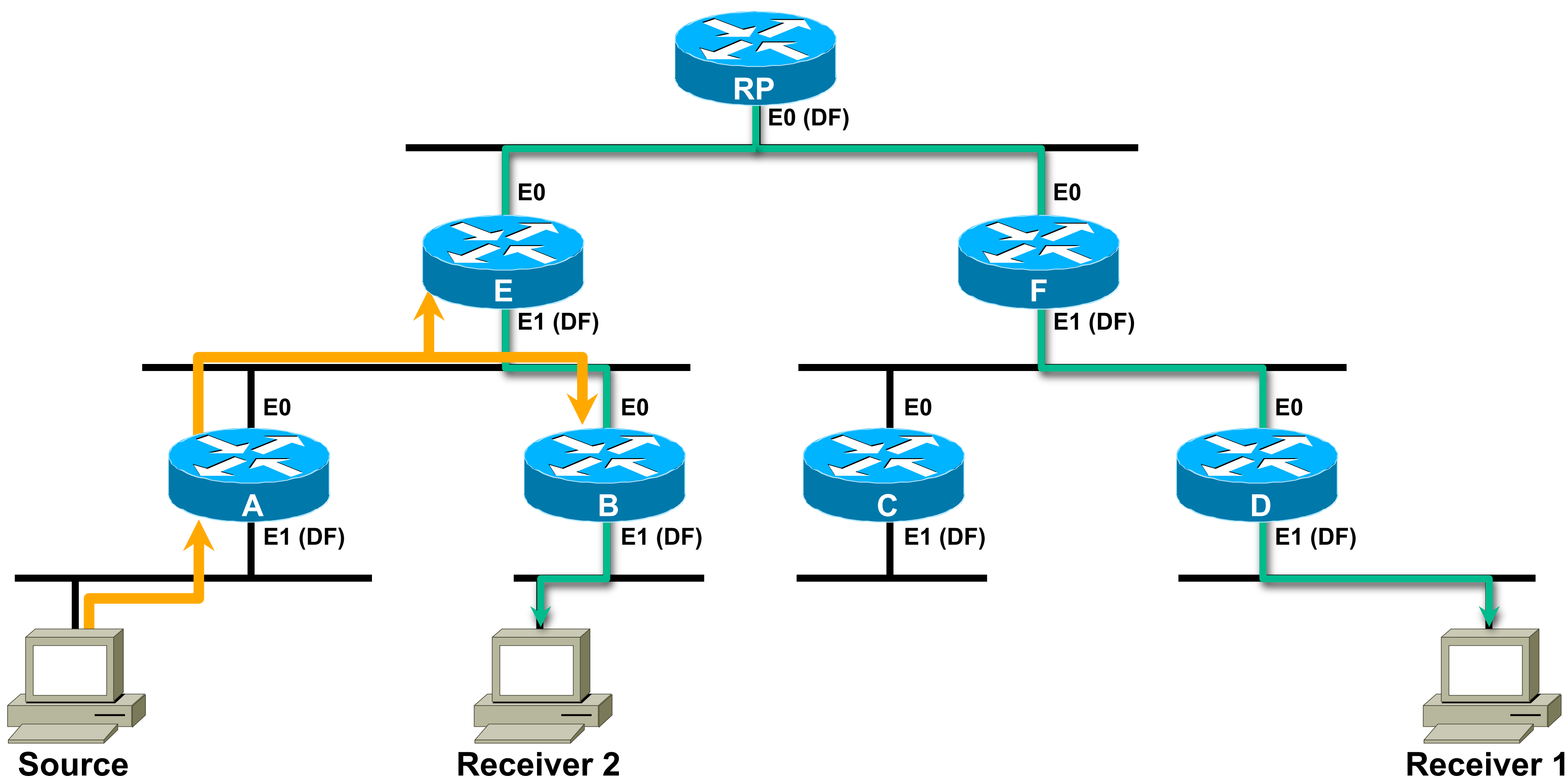
New Branch of Shared Tree Is Built to Receiver 2

Forwarding/Tree Building



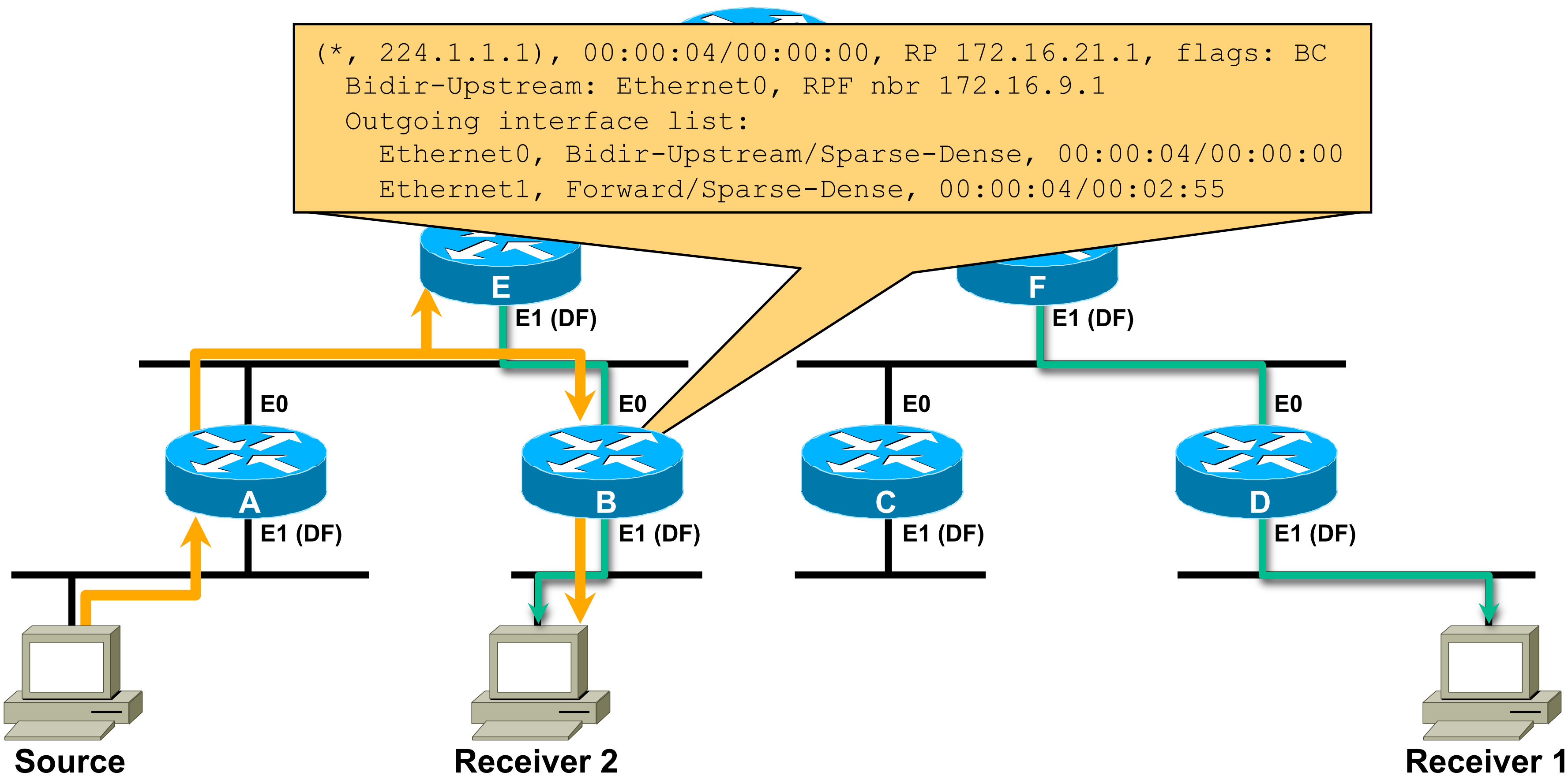
Arriving Traffic from Source Causes Router "A" to Create (*, G) State

Forwarding/Tree Building



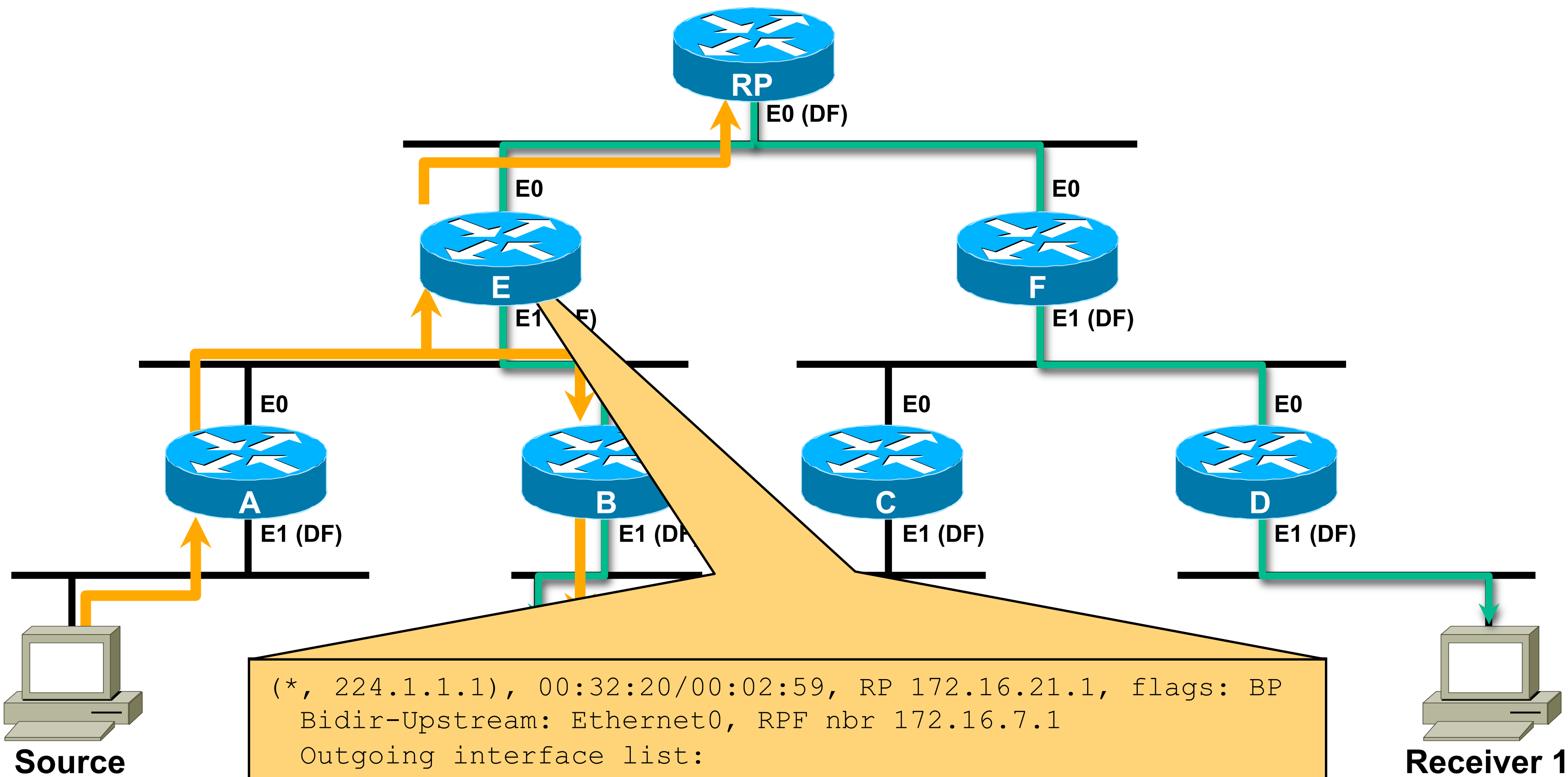
Traffic Is Forwarded Toward Router "E" and Also Arrives at IIF of Router "B"

Forwarding/Tree Building



Router "B" Forwards Traffic Back Down Shared Tree ala Normal PIM-SM

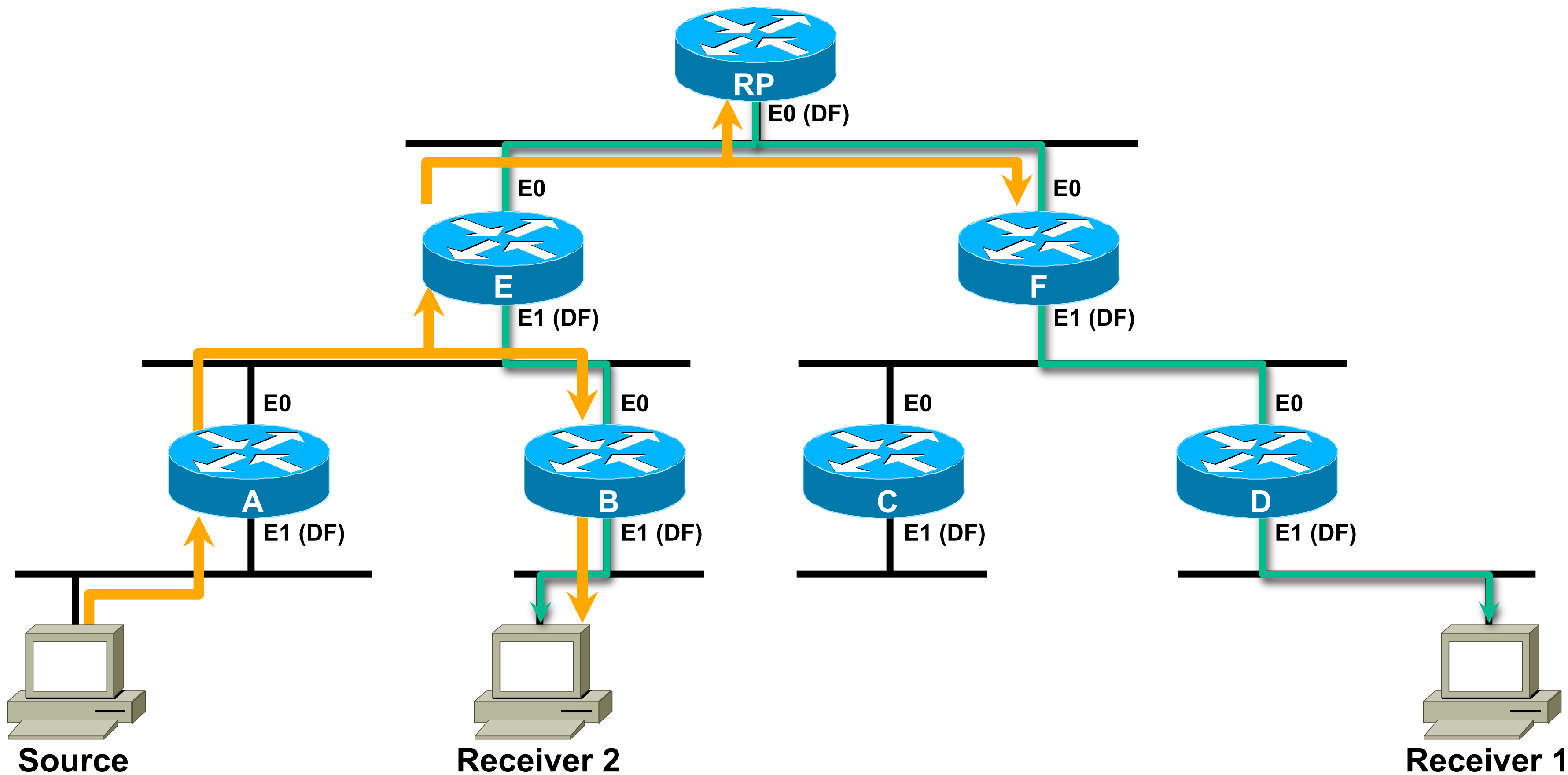
Forwarding/Tree Building




```
(*, 224.1.1.1), 00:32:20/00:02:59, RP 172.16.21.1, flags: BP
  Bidir-Upstream: Ethernet0, RPF nbr 172.16.7.1
  Outgoing interface list:
    Ethernet0, Bidir-Upstream/Sparse-Dense, 00:32:20/00:00:00
    Ethernet1, Forward/Sparse-Dense, 00:00:04/00:02:55
```

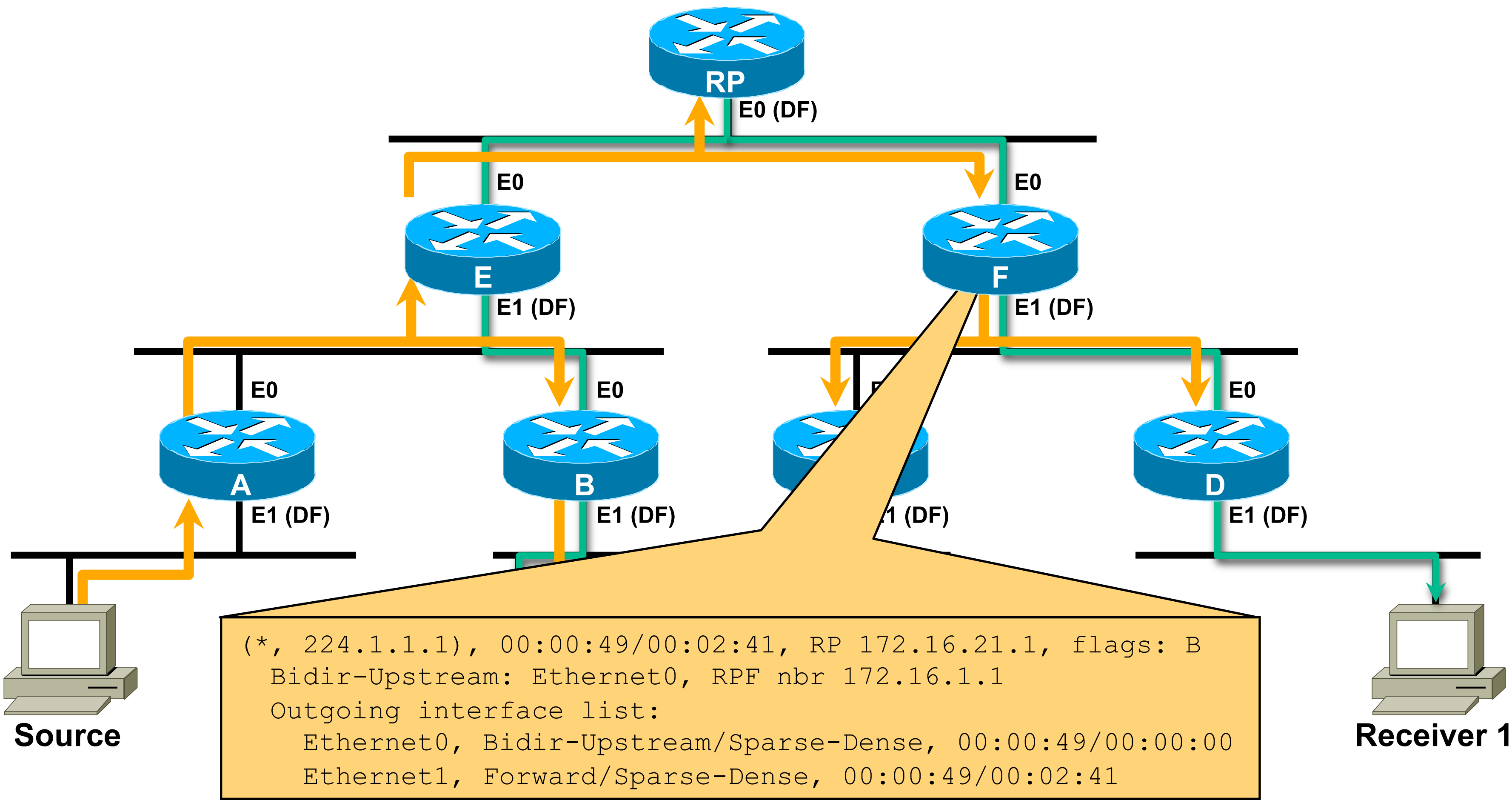
Router "E" Forwards Traffic on Toward RP

Forwarding/Tree Building



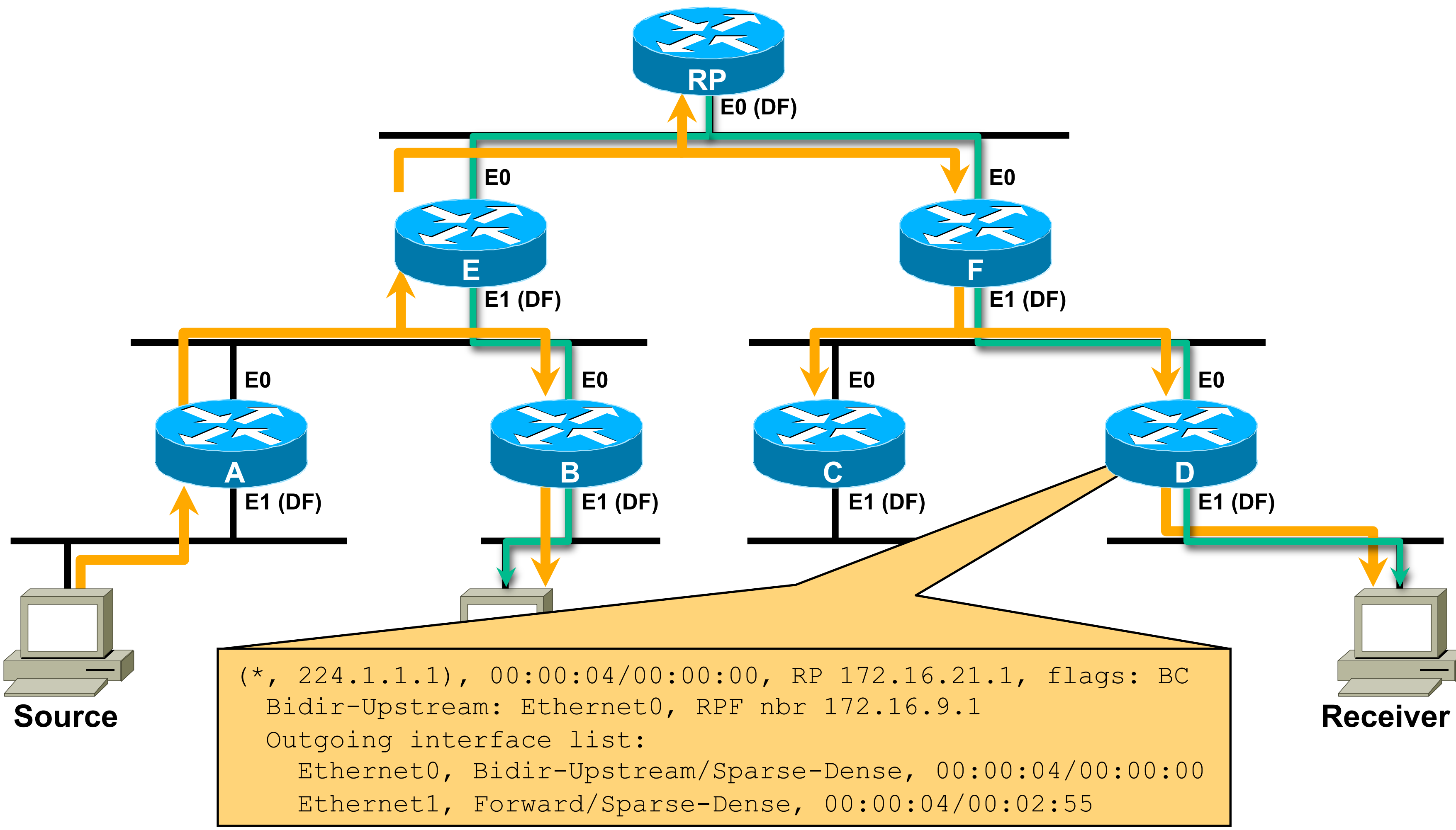
Traffic Forwarded Toward RP Also Arrives at the IIF of Router "F"

Forwarding/Tree Building



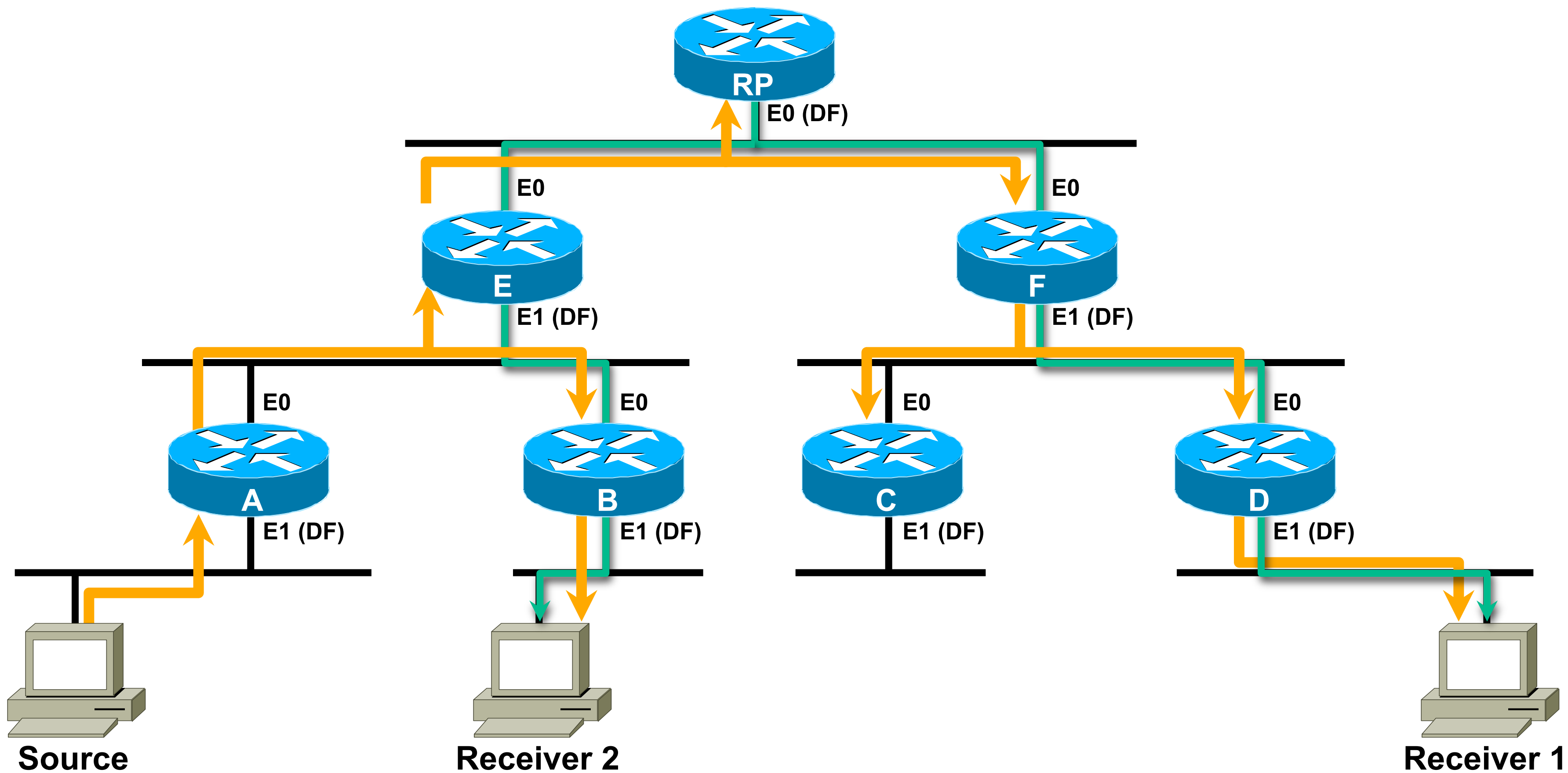
Router "F" Forwards Traffic on Down the Shared Tree ala Normal PIM-SM

Forwarding/Tree Building



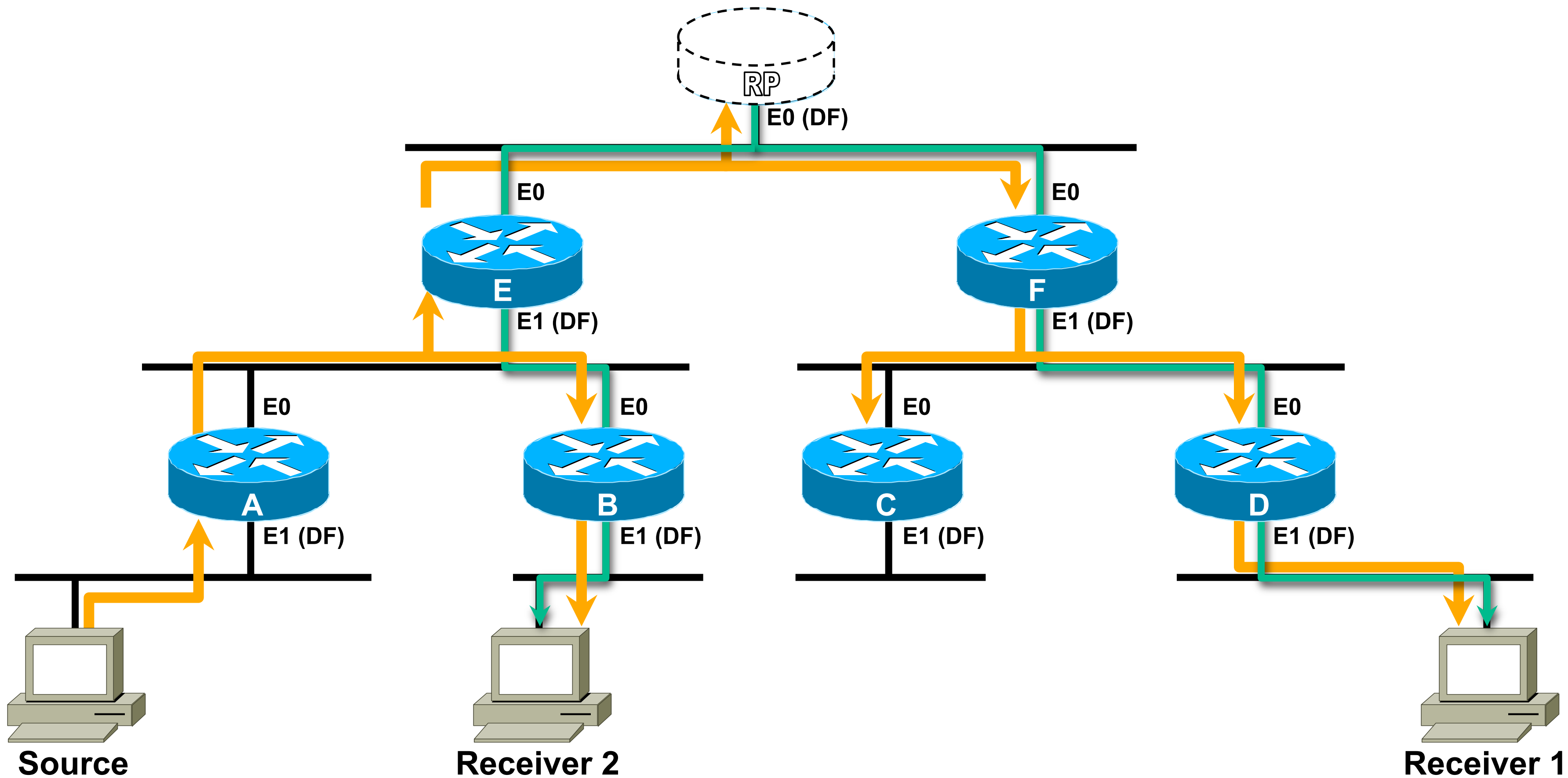
Router “D” Forwards Traffic to Receiver 1 via the Shared Tree

Forwarding/Tree Building



Question: Does the RP even have to physically exist?

Forwarding/Tree Building



Question: Does the RP even have to physically exist?

Answer: No. It can just be a phantom address.

Bidir PIM—Summary

- Uses Shared Trees only
  - Single (*, G) forwarding entry per group
  - Source traffic flows up and down Shared Tree
- Drastically reduces network mroute state
  - Eliminates ALL (S,G) state in the network
    - By eliminating SPT between source and RP
  - Allows many-to-any applications to scale
    - Permits virtually an unlimited number of sources

Rendezvous Points

Module Agenda

- Static RPs
- Auto RP
- PIMv2 BSR
- Anycast RPs
- Tuning RP Operations

Static RPs

Static RPs

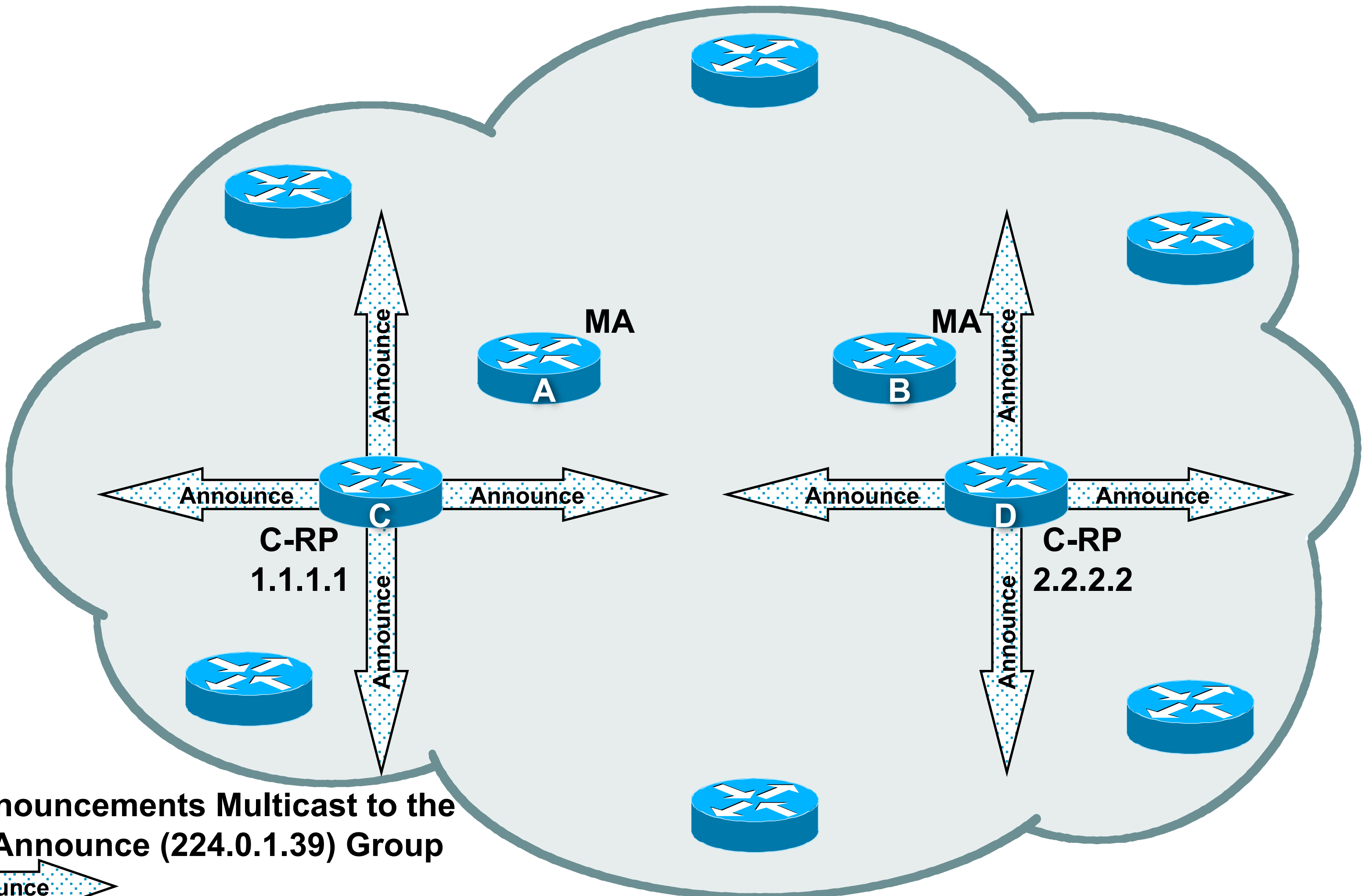
- Hard-configured RP address
  - RP assigned to a loopback (Lo1) on the RP router
  - Same static RP configuration must be on all routers
  - RP can be relocated to another router through configuration
    - Set loopback of “new” RP to the domains RP address
  - Failover using Anycast RPs (Later...)
- Command
  - `ip pim rp-address <address> [group-list <acl>] [override]`
  - Optional group list specifies group range
    - Default: Range = 224.0.0.0/4 **(Includes Auto-RP Groups!!!!)**
  - Override keyword “overrides” Auto-RP information
    - Default: Auto-RP learned info takes precedence

Auto-RP

Auto-RP Overview

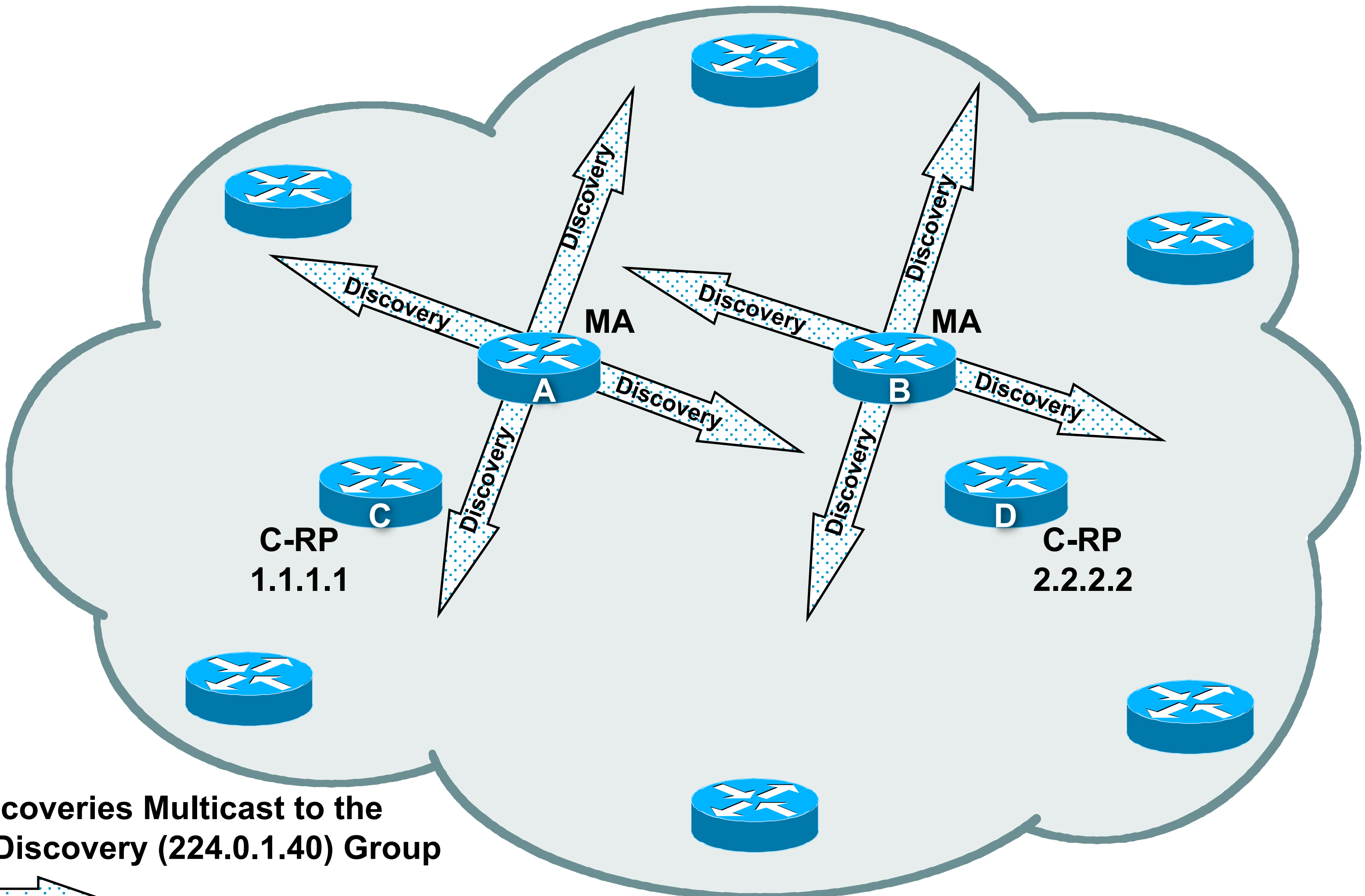
- All routers automatically learn RP address
  - Configuration requirements:
    - Candidate RPs
    - Mapping Agents
    - PIM Sparse-dense on all router interfaces
- Makes use of Multicast to distribute info
  - Two specially IANA assigned Groups used
    - Cisco-Announce—224.0.1.39
    - Cisco-Discovery—224.0.1.40
  - Typically Dense mode is used to forward these groups
- Backup RPs supported as part of the protocol
- Can be used with Admin-Scoping

Auto-RP—From 10,000 Feet



RP-Announcements Multicast to the Cisco Announce (224.0.1.39) Group

Auto-RP—From 10,000 Feet



RP-Discoveries Multicast to the Cisco Discovery (224.0.1.40) Group

Auto-RP Fundamentals

- Candidate RPs
  - Multicast RP-Announcement messages
    - Sent to Cisco-Announce (224.0.1.39) group
    - Sent every rp-announce-interval (default: 60 sec)
  - RP-Announcements contain:
    - Group Range (default = 224.0.0.0/4)
    - Candidate’s RP address
    - Holdtime = 3 x <rp-announce-interval>
  - Configured via global config command

    ```
    ip pim send-rp-announce <intfc> scope <ttl> [group-list acl]
    ```
  - ‘Deny’ in group-list has variable meaning
    - Before 12.0(1.1) Deny = “I’m not C-RP for this group-range”
    - After 12.0(1.1) Deny = “Force group-range to always be DM”

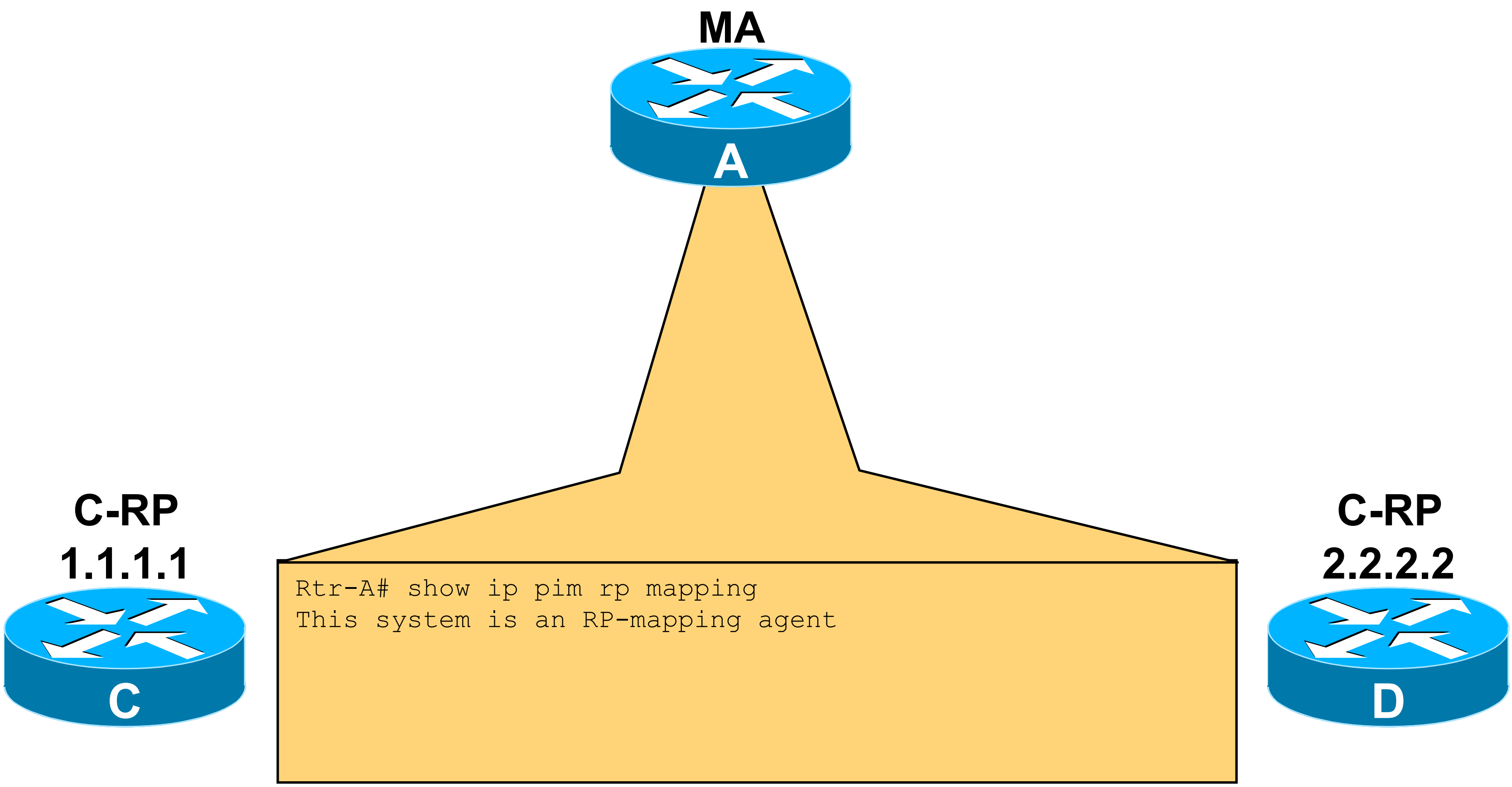
Auto-RP Fundamentals

- Mapping agents
  - Receive RP-Announcements
    - Stored in Group-to-RP Mapping Cache with holdtimes
    - Elects highest C-RP IP address as RP for group range
  - Multicast RP-Discovery messages
    - Sent to Cisco-Discovery (224.0.1.40) group
    - Sent every 60 seconds or when changes detected
  - RP-Discovery messages contain:
    - Elected RPs from MA's Group-to-RP Mapping Cache
  - Configured via global config command
    - **ip pim send-rp-discovery [<interface>] scope <ttl>**
    - Source address of packets set by '<interface>' (12.0)
      - If not specified, source address = output interface address
      - Results in the appearance of multiple MA's. (one/interface)

Auto-RP Fundamentals

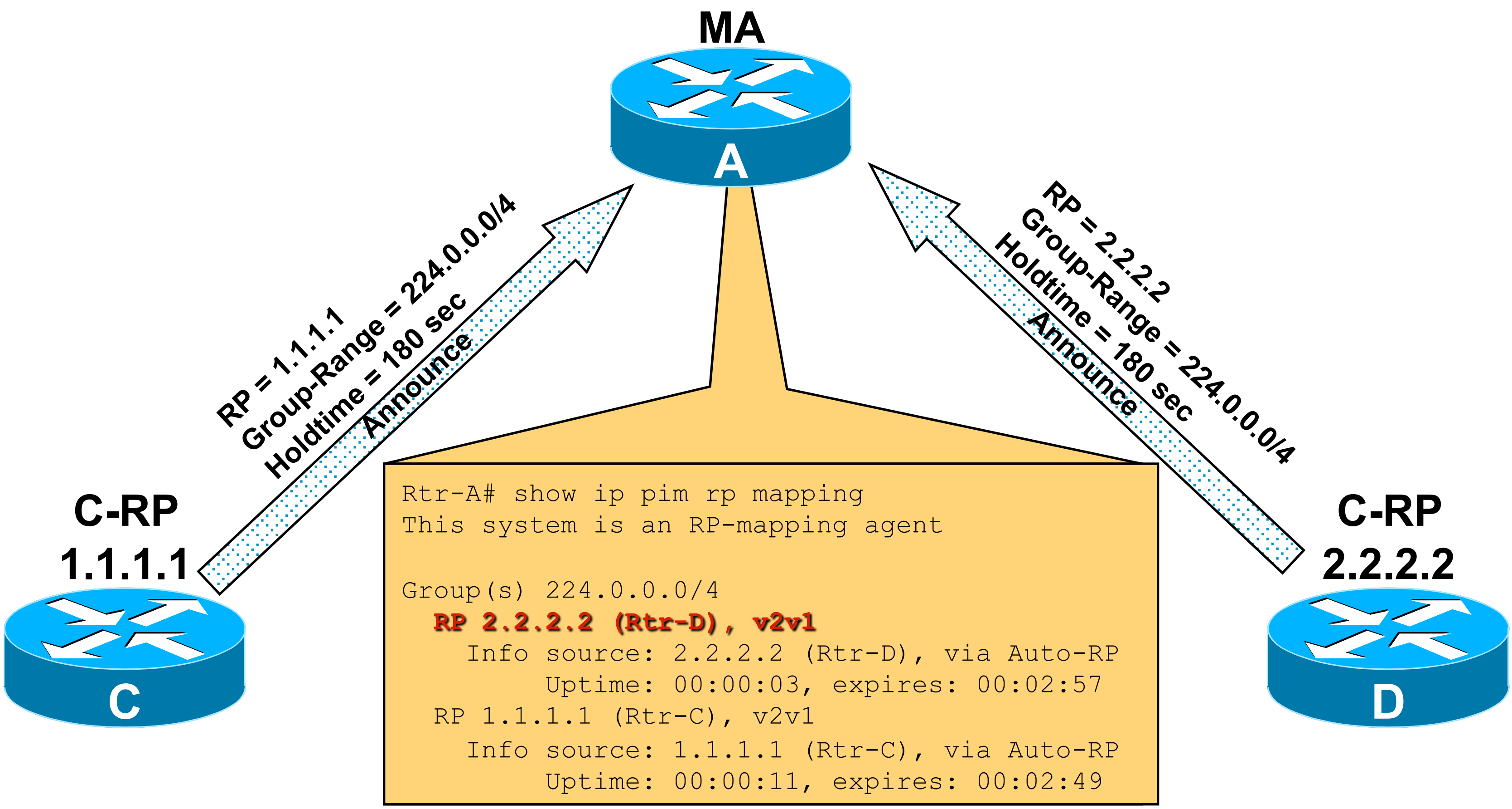
- All Cisco routers
  - Join Cisco-Discovery (224.0.1.40) group
    - Automatic
    - No configuration necessary
  - Receive RP-Discovery messages
    - Stored in local Group-to-RP Mapping Cache
    - Information used to determine RP for group range

Auto-RP—A Closer Look



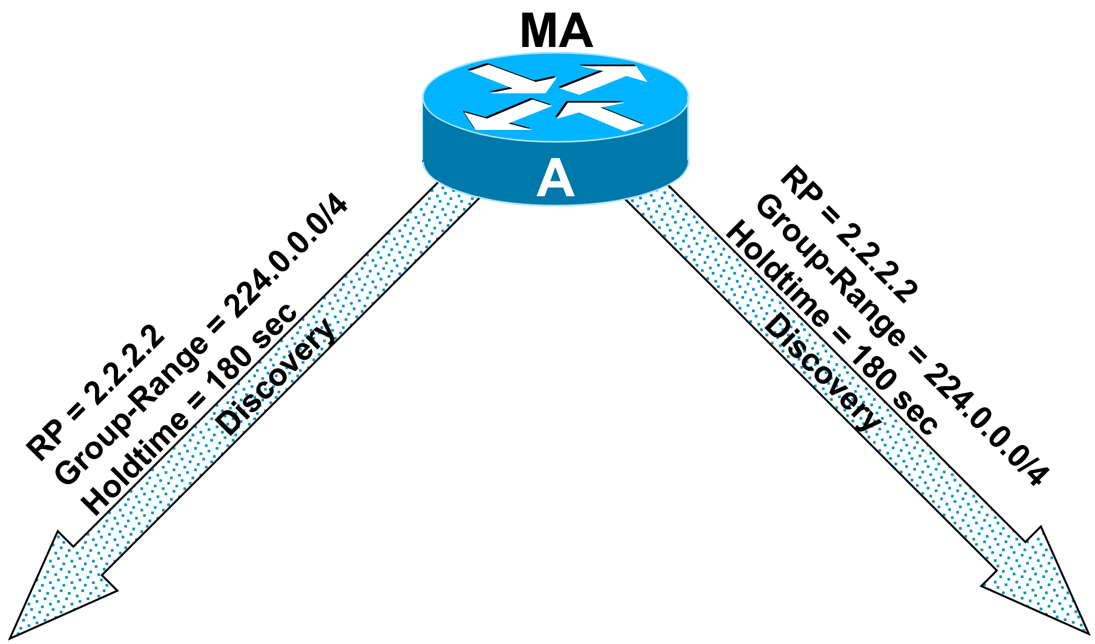
Initial Cache State in the Mapping Agent

Auto-RP—A Closer Look



- C-RP information is stored in MA's Group-to-RP Mapping Cache
- Mapping Agent elects highest IP Address as RP

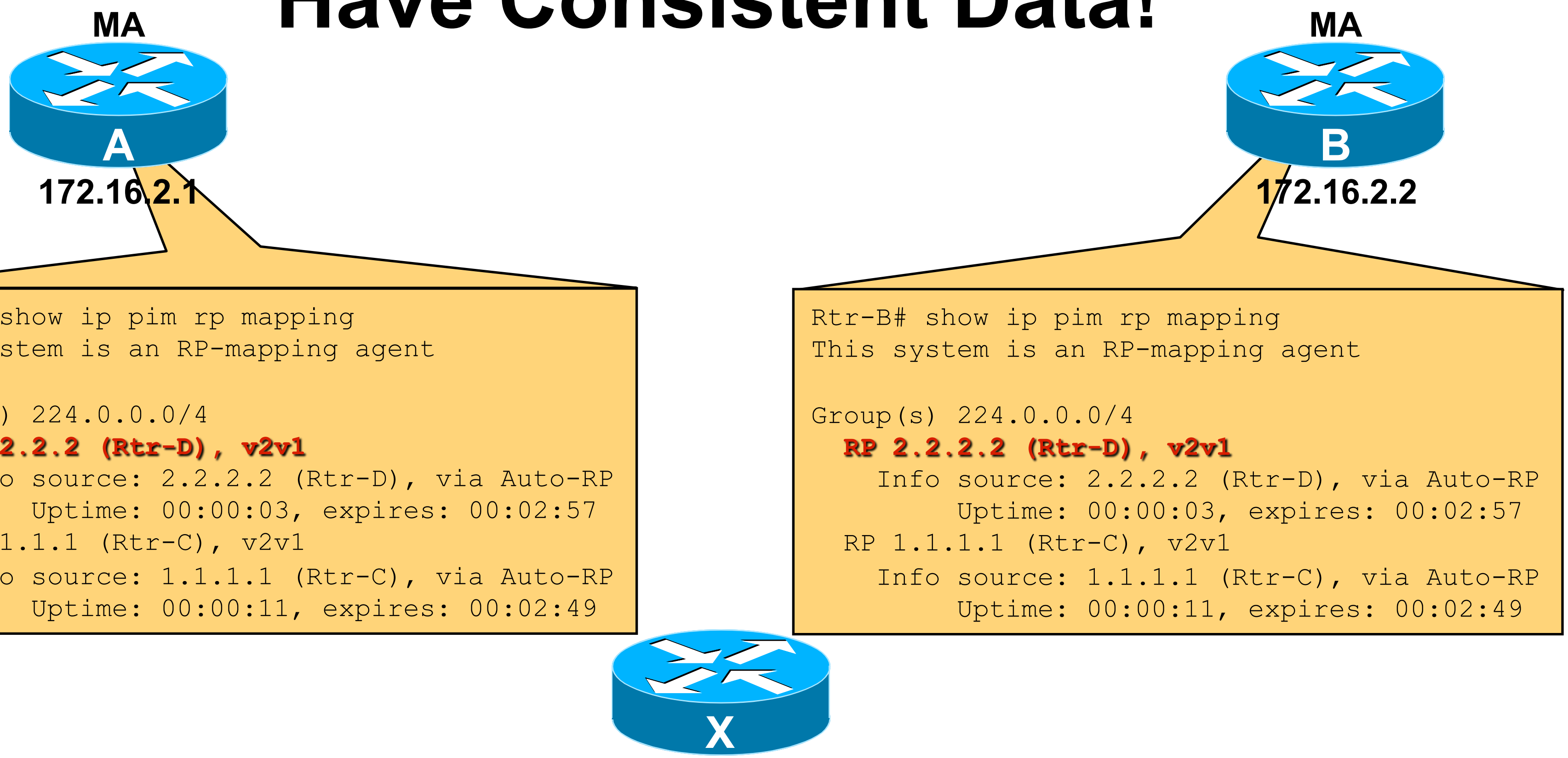
Auto-RP—A Closer Look



- Mapping Agent advertises elected RP via Discovery messages

Auto-RP—A Closer Look

All Mapping Agents **Must** Have Consistent Data!

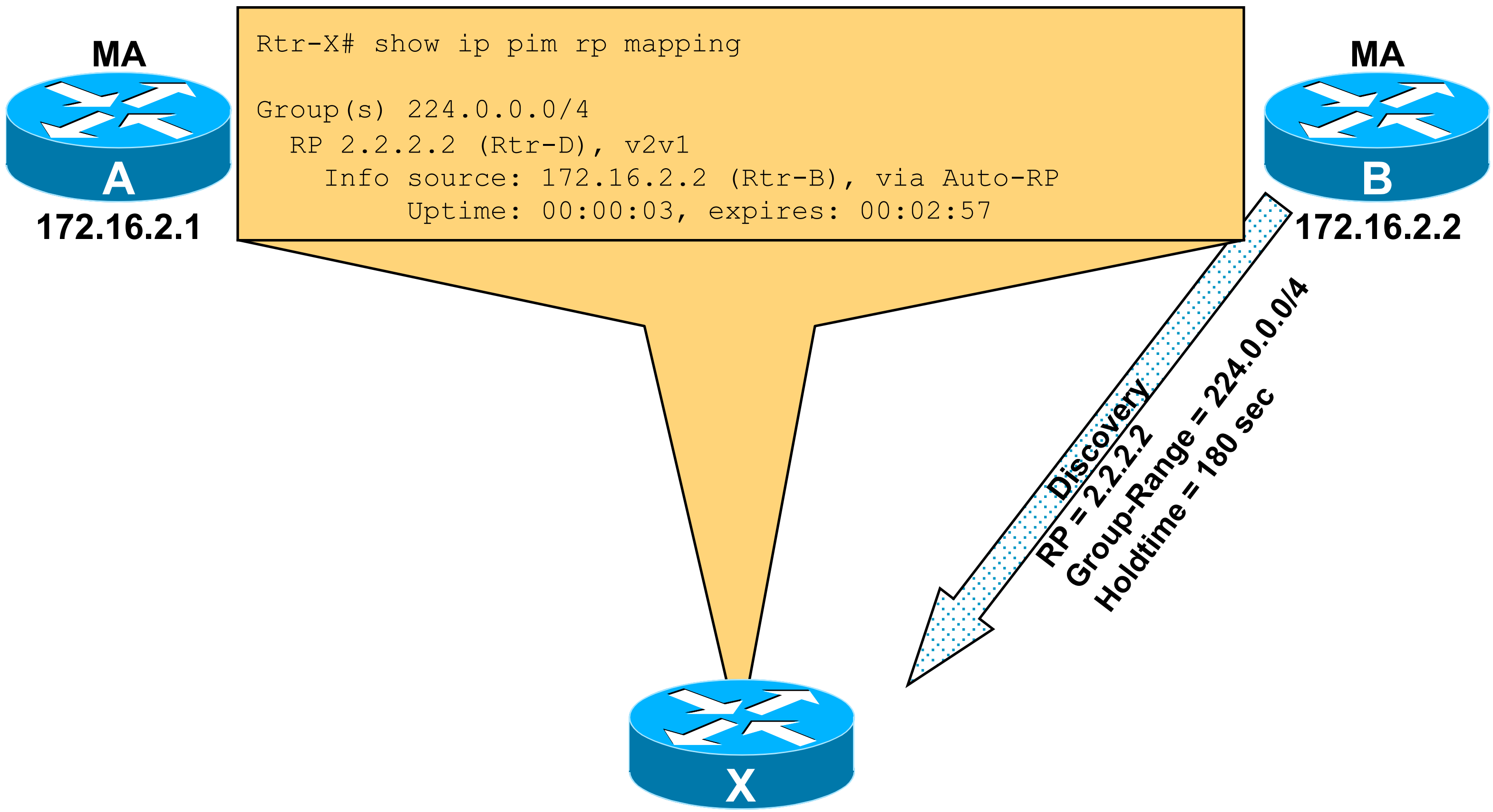



```
Rtr-A# show ip pim rp mapping
This system is an RP-mapping agent

Group(s) 224.0.0.0/4
  RP 2.2.2.2 (Rtr-D), v2v1
    Info source: 2.2.2.2 (Rtr-D), via Auto-RP
         Uptime: 00:00:03, expires: 00:02:57
  RP 1.1.1.1 (Rtr-C), v2v1
    Info source: 1.1.1.1 (Rtr-C), via Auto-RP
         Uptime: 00:00:11, expires: 00:02:49
```

```
Rtr-B# show ip pim rp mapping
This system is an RP-mapping agent

Group(s) 224.0.0.0/4
  RP 2.2.2.2 (Rtr-D), v2v1
    Info source: 2.2.2.2 (Rtr-D), via Auto-RP
         Uptime: 00:00:03, expires: 00:02:57
  RP 1.1.1.1 (Rtr-C), v2v1
    Info source: 1.1.1.1 (Rtr-C), via Auto-RP
         Uptime: 00:00:11, expires: 00:02:49
```

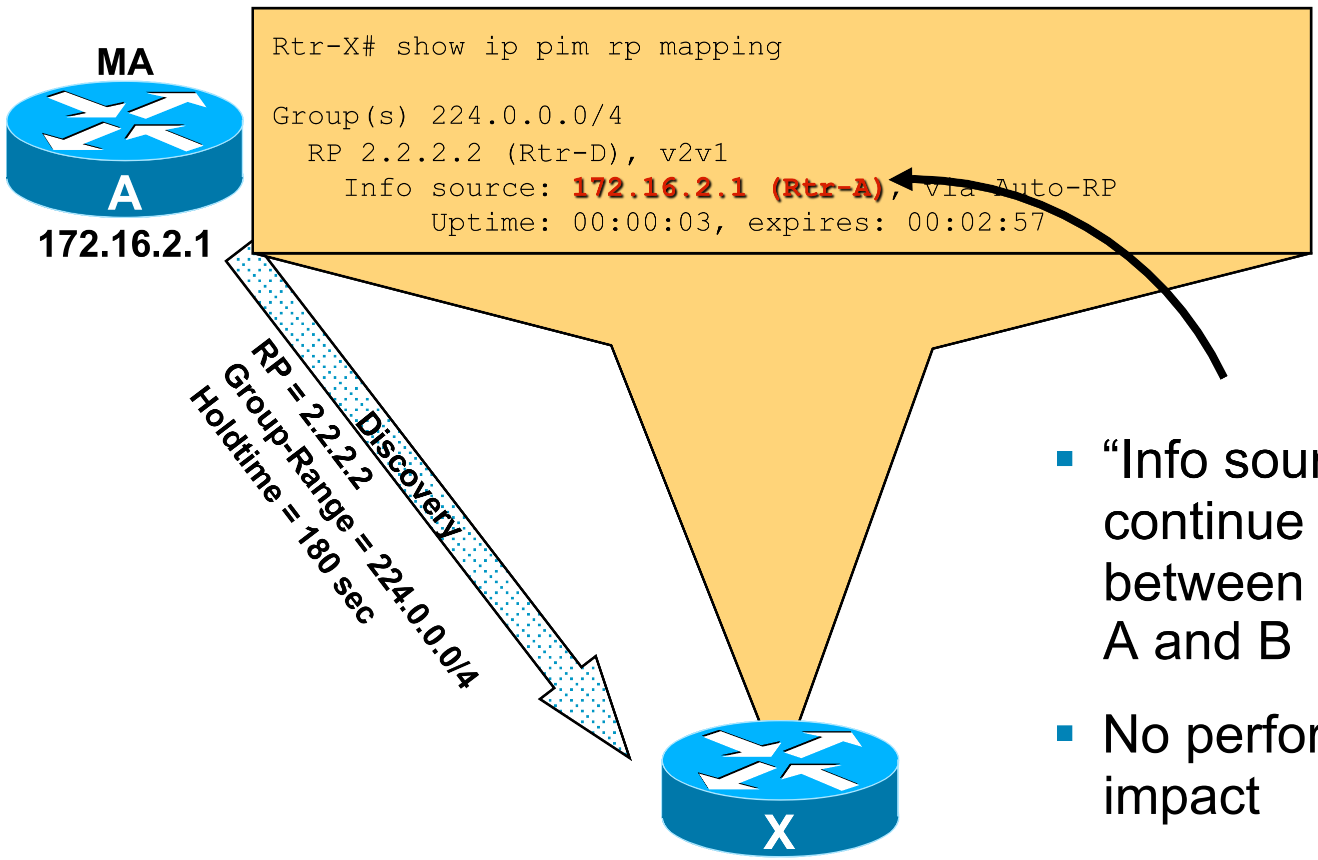
Auto-RP—A Closer Look

Local Cache Initially Loaded from Router "B"



Auto-RP—A Closer Look

Identical Info Received From Router "A"



- "Info source" will continue to flip-flop between routers A and B
- No performance impact

Auto-RP Failover

- RP failover time
  - Function of ‘Holdtime’ in C-RP Announcement
    - Holdtime = 3 x <rp-announce-interval>
    - Default < rp-announce-interval> = 60 seconds
    - Default Failover ~ 3 minutes
- Minimizing impact of RP failure
  - Use SPTs to reduce impact
    - Traffic on SPTs not affected by RP failure
    - Immediate switch to SPTs is on by default
    - New and/or bursty sources still a problem

Tuning Auto-RP Failover

- Tune Candidate RPs
- Use 'interval' clause to control failover times

```
ip pim send-rp-announce <intfc> scope <ttl>
                              [group-list acl]
                              [interval <seconds>]
```

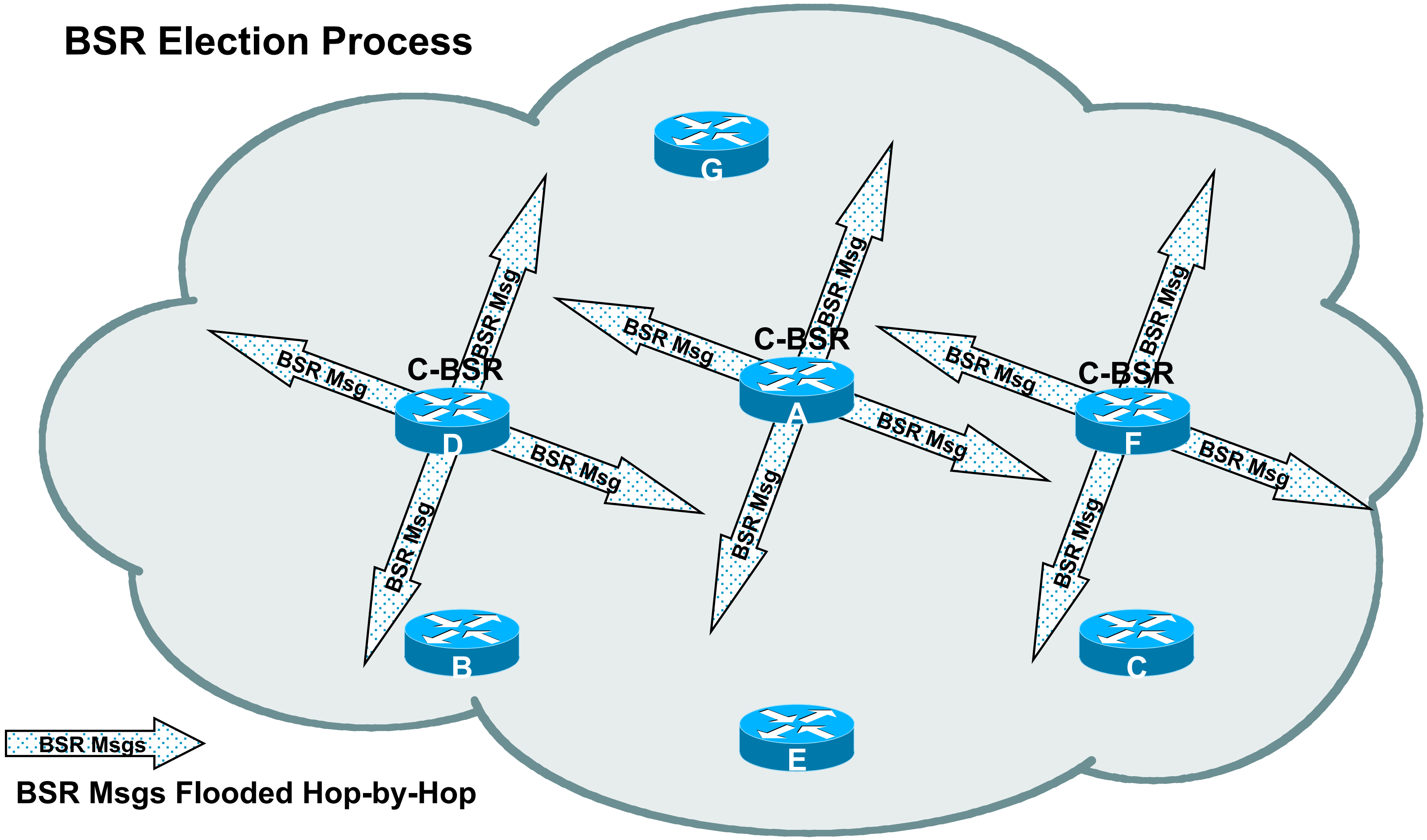
- Allows rp-announce-interval to be adjusted
- Smaller intervals = Faster RP failover
- Smaller intervals increase amount Auto-RP traffic
  - Increase is usually insignificant
- Total RP failover time reduced
  - Min. failover ~ 3 seconds
- **Consider using Anycast RP for faster failover**

BSR

BSR Overview

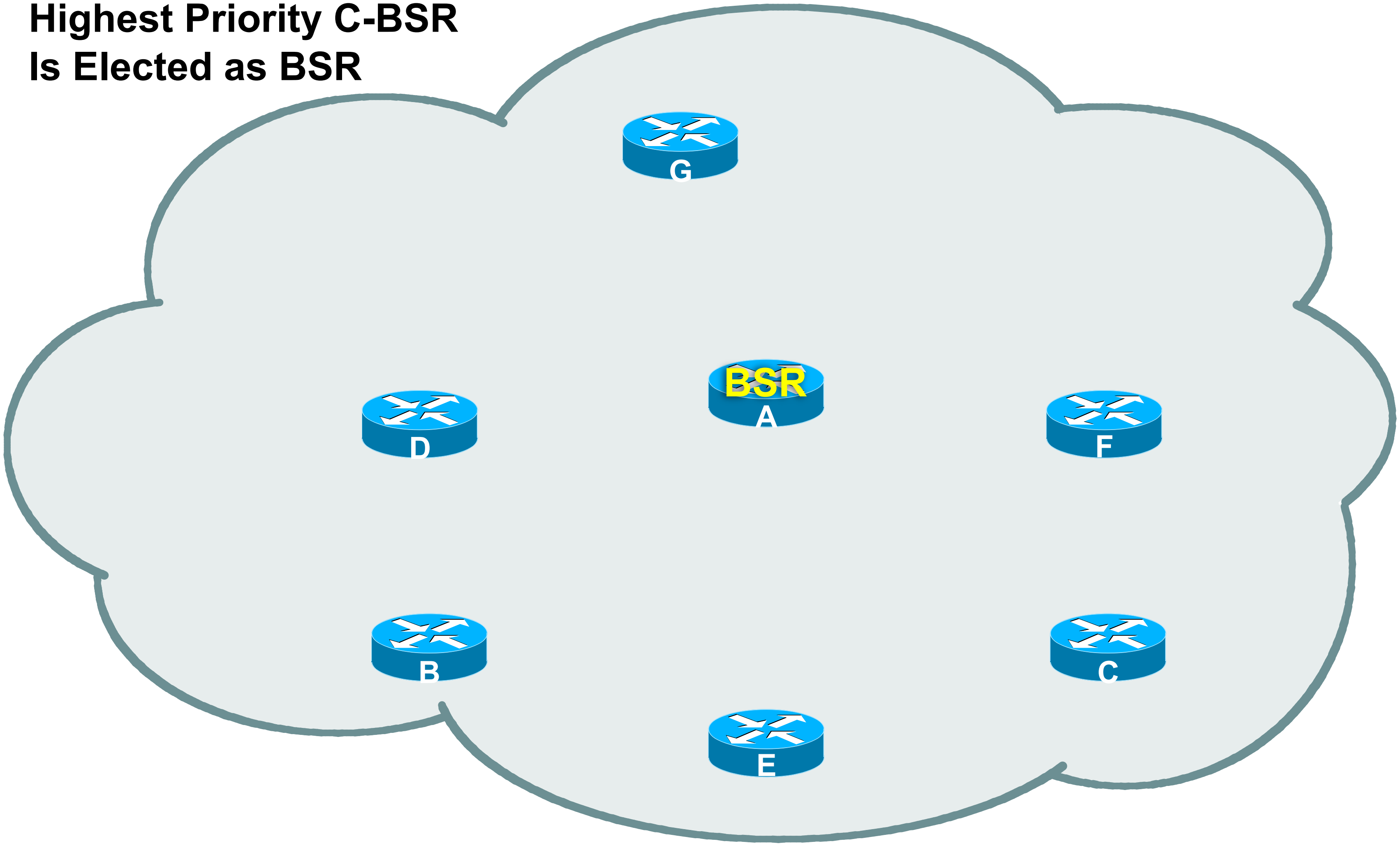
- A single Bootstrap Router (BSR) is elected
  - Multiple Candidate BSRs (C-BSR) can be configured
    - Provides backup in case currently elected BSR fails
  - C-RPs send C-RP announcements to the BSR
    - C-RP announcements are sent via unicast
    - BSR stores ALL C-RP announcements in the “RP-set”
  - BSR periodically sends BSR messages to all routers
    - BSR Messages contain entire RP-set and IP address of BSR
    - Messages are flooded hop-by-hop throughout the network away from the BSR
  - All routers select the RP from the RP-set
    - All routers use the same selection algorithm; select same RP
- BSR cannot be used with Admin-Scoping

BSR—From 10,000 Feet

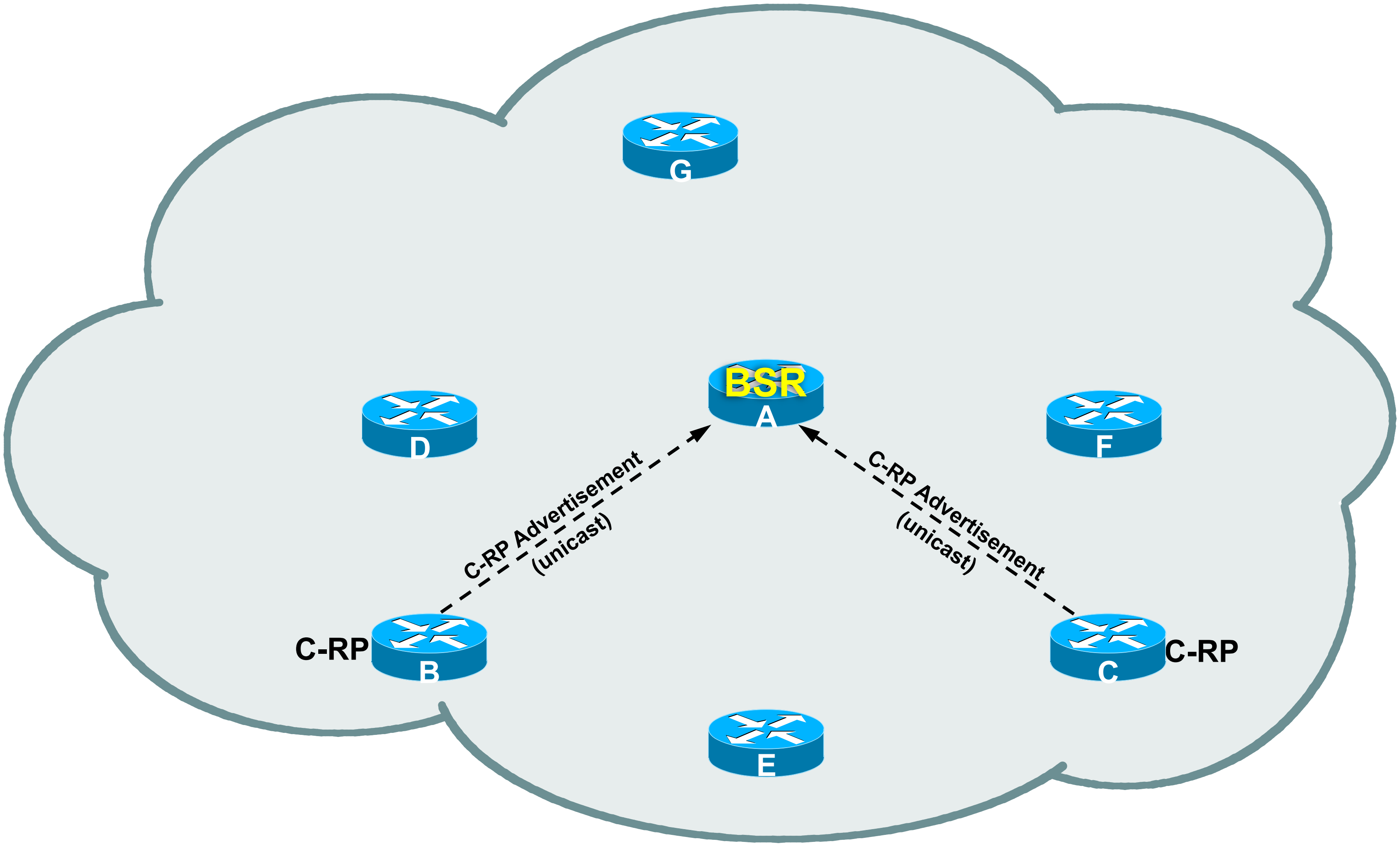


BSR—From 10,000 Feet

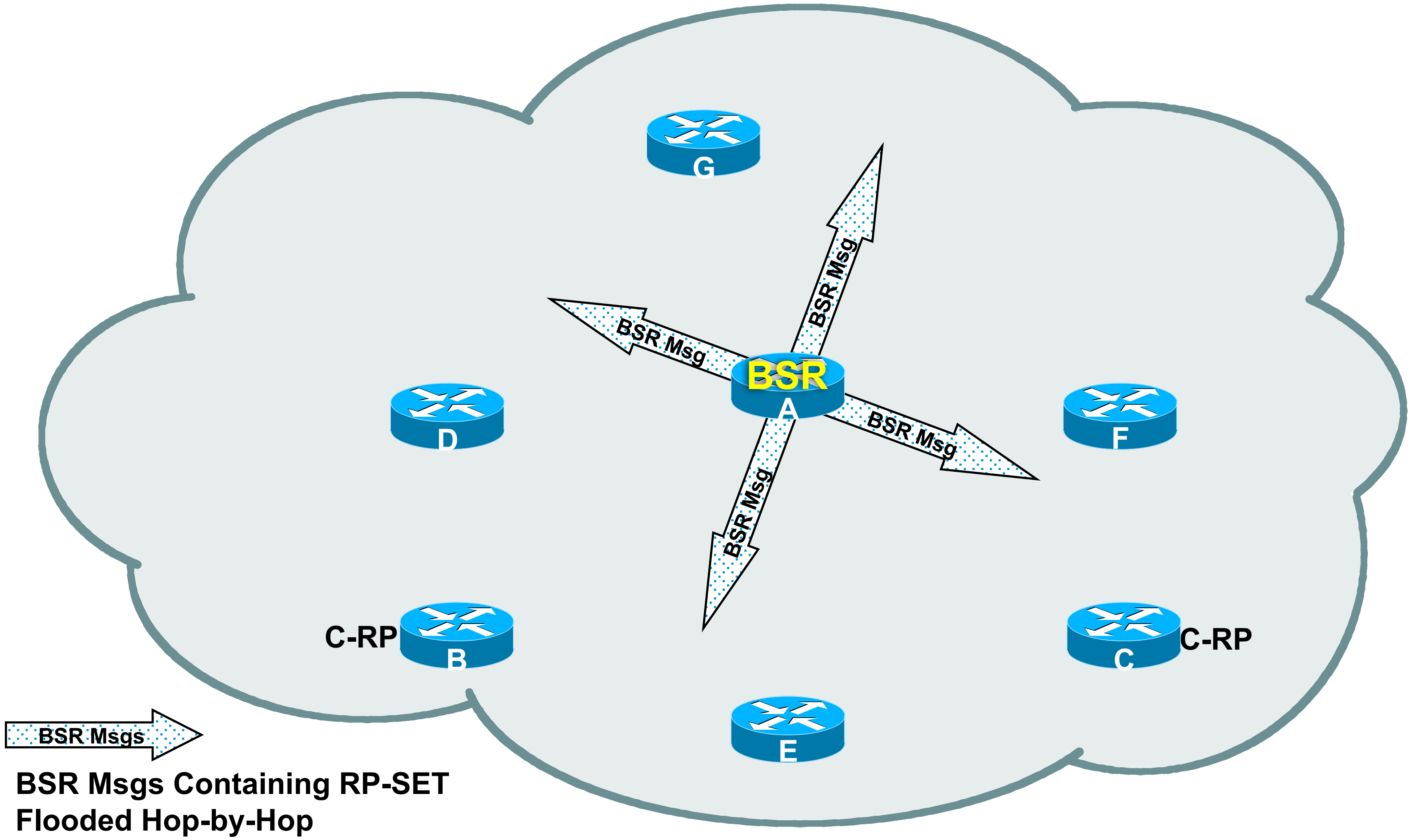
Highest Priority C-BSR
Is Elected as BSR



BSR—From 10,000 Feet



BSR—From 10,000 Feet



BSR Msgs Containing RP-SET Flooded Hop-by-Hop

BSR Fundamentals

- Candidate RPs
  - Unicast PIMv2 C-RP messages to BSR
    - Learns IP address of BSR from BSR messages
    - Sent every rp-announce-interval (default: 60 sec)
  - C-RP messages contain:
    - Group Range (default = 224.0.0.0/4)
    - Candidate's RP address
    - Holdtime = 3 x <rp-announce-interval>
  - Configured via global config command

```
ip pim rp-candidate <intfc> [group-list acl]
```


BSR Fundamentals

- Bootstrap router (BSR)
  - Receive C-RP messages
    - Accepts and stores ALL C-RP messages
    - Stored in Group-to-RP Mapping Cache w/holdtimes
  - Originates BSR messages
    - Multicast to All-PIM-Routers (224.0.0.13) group
      - (Sent with a TTL = 1)
    - Sent out all interfaces. Propagate hop-by-hop
    - Sent every 60 seconds or when changes detected
  - BSR messages contain:
    - Contents of BSR's Group-to-RP Mapping Cache
    - IP Address of active BSR

BSR Fundamentals

- Candidate bootstrap router (C-BSR)
  - C-BSR with highest priority elected BSR
    - C-BSR IP address used as tie-breaker
      - (Highest IP address wins)
    - The active BSR may be preempted
      - New router w/higher BSR priority forces new election
  - Configured via global config command
    - **ip pim bsr-candidate <intfc> <hash-length> [priority <pri>]**
      - **<intfc>**
        - Determines IP address
      - **<hash-length>**
        - Sets RP selection hash mask length
      - **<pri>**
        - Sets the C-BSR priority (default = 0)

BSR Fundamentals

- All PIMv2 routers
  - Receive BSR messages
    - Stored in local Group-to-RP Mapping Cache
    - Information used to determine active BSR address
  - Selects RP using Hash algorithm
    - Selected from local Group-to-RP Mapping Cache
    - All routers select same RP using same algorithm
    - Permits RP-load balancing across group range

Anycast RPs

Anycast RPs

- RFC 3446 “Anycast RP Mechanism … ”
- Basic concepts

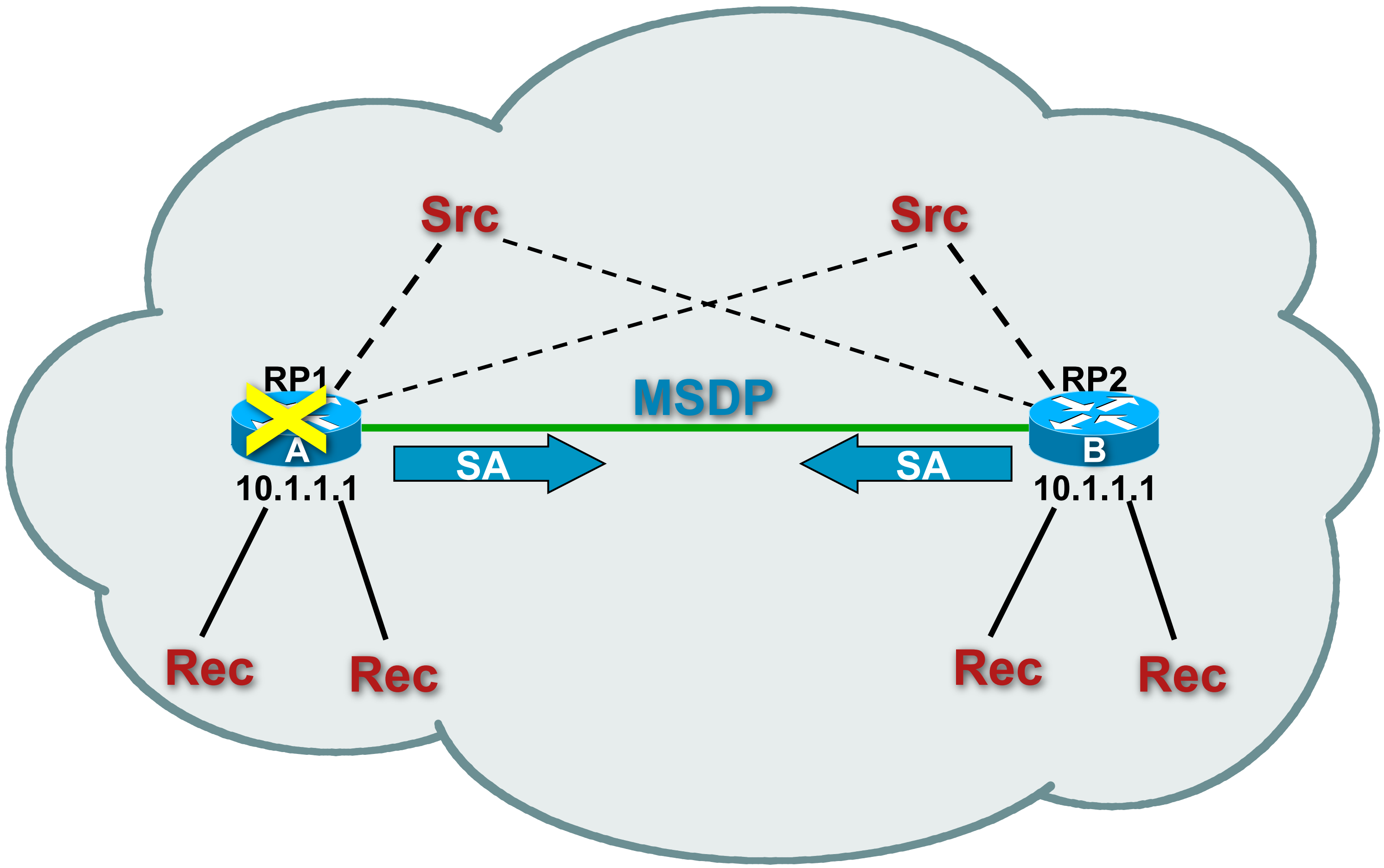
  Within a domain, deploy more than one RP for the same group range

  Give each RP the same IP address assignment

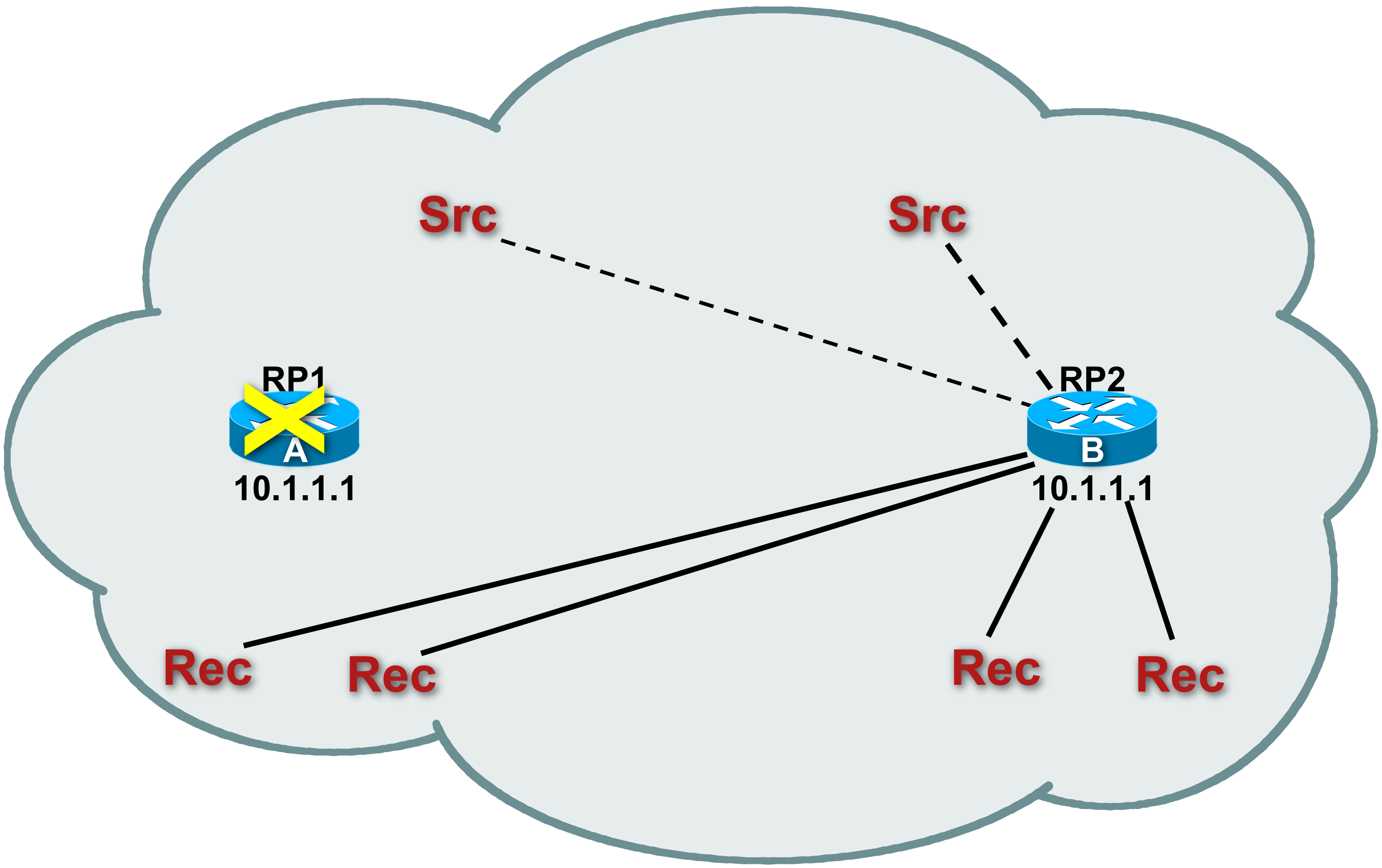
  Sources and receivers use closest RP

  Use MSDP (Multicast Source Discovery Protocol) to communicate existence of Sources between RPs

Anycast RP—Overview



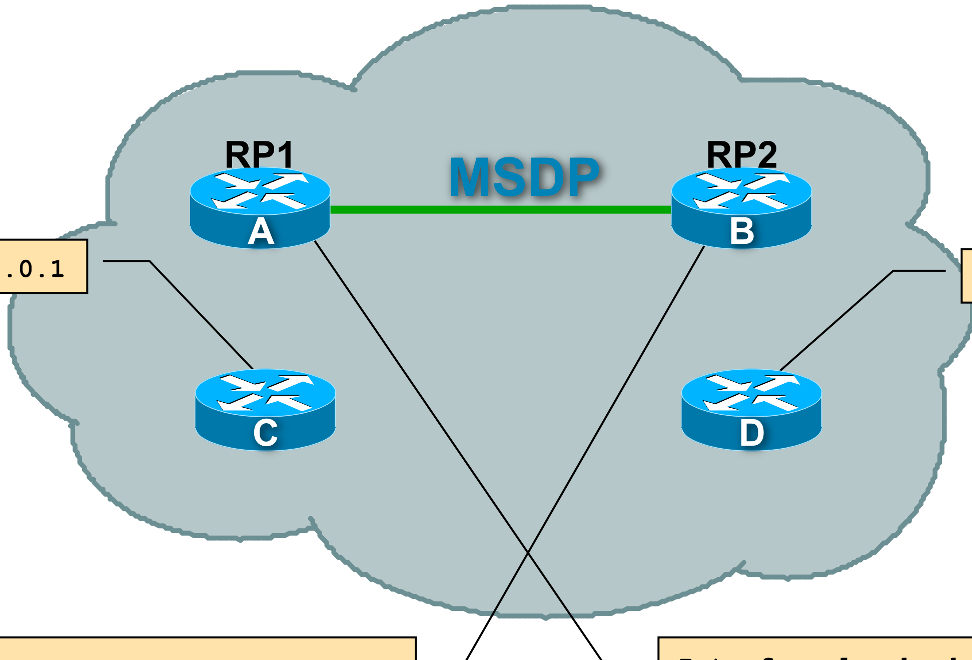
Anycast RP—Overview



Anycast RPs

- Advantages
  - Rapid RP Failover
    - Converges within seconds of unicast
  - No Dense Mode Fallback
    - Because RP address is statically defined
- Disadvantages
  - More configuration
    - Static RP definition on every router
  - Requires MSDP
    - Only necessary on RP routers

Anycast RP Configuration




```
ip pim rp-address 10.0.0.1
```

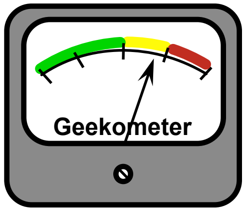
```
ip pim rp-address 10.0.0.1
```

```
Interface loopback 0
   ip address 10.0.0.1 255.255.255.255

Interface loopback 1
 ip address 10.0.0.2 255.255.255.255
!
ip msdp peer 10.0.0.3 connect-source loopback 1
ip msdp originator-id loopback 1
```

```
Interface loopback 0
 ip address 10.0.0.1 255.255.255.255

Interface loopback 1
 ip address 10.0.0.3 255.255.255.255
!
ip msdp peer 10.0.0.2 connect-source loopback 1
ip msdp originator-id loopback 1
```


Deploying IP Multicast

Agenda



- Basic Multicast Engineering
- Advanced Multicast Engineering

Basic Multicast Engineering

PIM Configuration Steps

PIM Configuration Steps

- Enable Multicast Routing on **every** router
- Configure **every** interface for PIM
- Configure the RP
  - Using Auto-RP or BSR
    - Configure certain routers as Candidate RP(s)
    - All other routers automatically learn elected RP
  - Anycast/Static RP addressing
    - RP address must be configured on every router
    - Note: Anycast RP requires MSDP

Group Mode vs. Interface Mode

- Group and Interface mode are independent
  - Interface mode
    - Determines how the **interface** operates when sending/ receiving multicast traffic
  - Group mode
    - Determines whether the group is Sparse or Dense

Configuring Interface

- Interface mode configuration commands
  - Enables multicast forwarding on the interface
  - Controls the **interface's** mode of operation
  - **`ip pim dense-mode`**
    - Not Recommended
  - **`ip pim sparse-mode`**
    - Interface mode is set to Sparse mode operation
  - **`ip pim sparse-dense-mode`**
    - Not Recommended

Group Mode

- Group mode is controlled by local RP info
  - Local RP Information
    - Stored in the Group-to-RP Mapping Cache
    - May be statically configured or learned via Auto-RP or BSR
  - If RP info exists, Group = Sparse
  - If RP info does not exist, Group = Dense
  - Mode Changes are automatic
    - i.e. if RP info is lost, Group falls back to Dense

RP Engineering

General RP Recommendations

General RP Recommendations

- Use Anycast RPs:
  - When network must connect to Internet or
  - When rapid RP failover is critical
- Pros
  - Fastest RP Convergence method
  - Required when connecting to Internet
- Cons
  - Requires more configuration
  - Requires use of MSDP between RPs

General RP Recommendations

- Use Auto-RP
  - When minimum configuration is desired and/or
  - When maximum flexibility is desired
- Pros
  - Most flexible method
  - Easiest to maintain
- Cons
  - Increased RP Failover times vs. Anycast
  - Special care needed to avoid DM Fallback
    - Some methods greatly increase configuration

General RP Recommendations

- Use BSR:
  - When Static/Anycast RPs cannot be used and
  - Uh..??
- Pros
  - Interoperates with all vendors
  - ..though so does AutoRP and static mapping.
- Cons
  - Increased RP Failover times vs. Anycast
  - Special care needed to avoid DM Fallback
    - Some methods greatly increase configuration
  - Not as “field-proven” as other methods

RP Engineering

Combining Anycast RP With Auto-RP

Combining Auto-RP and Anycast-RP

- Anycast-RP and Auto-RP may be combined
  - Provides advantages of both methods
    - Rapid RP failover of Anycast RP
    - No DM Fallback
    - Configuration flexibility of Auto-RP
    - Ability to effectively disable undesired groups

Combining Auto-RP and Anycast-RP

Configuration Steps

1. Enable Auto-RP

   Newer IOS images

   Use ip pim autorp listener global command and configure ip pim sparse-mode on all interfaces

   Older IOS images

   Configure ip pim sparse-dense-mode on all interfaces

2. Configure Auto-RP Mapping Agents

```
ip pim send-rp-discovery interface Loopback0 scope 32
```

Combining Auto-RP and Anycast-RP

Configuration Steps

3. Block DM Fallback
   - Newer IOS images
     - Use no ip pim dm-fallback
   - Older IOS images
     - Configure RP-of-last-Resort

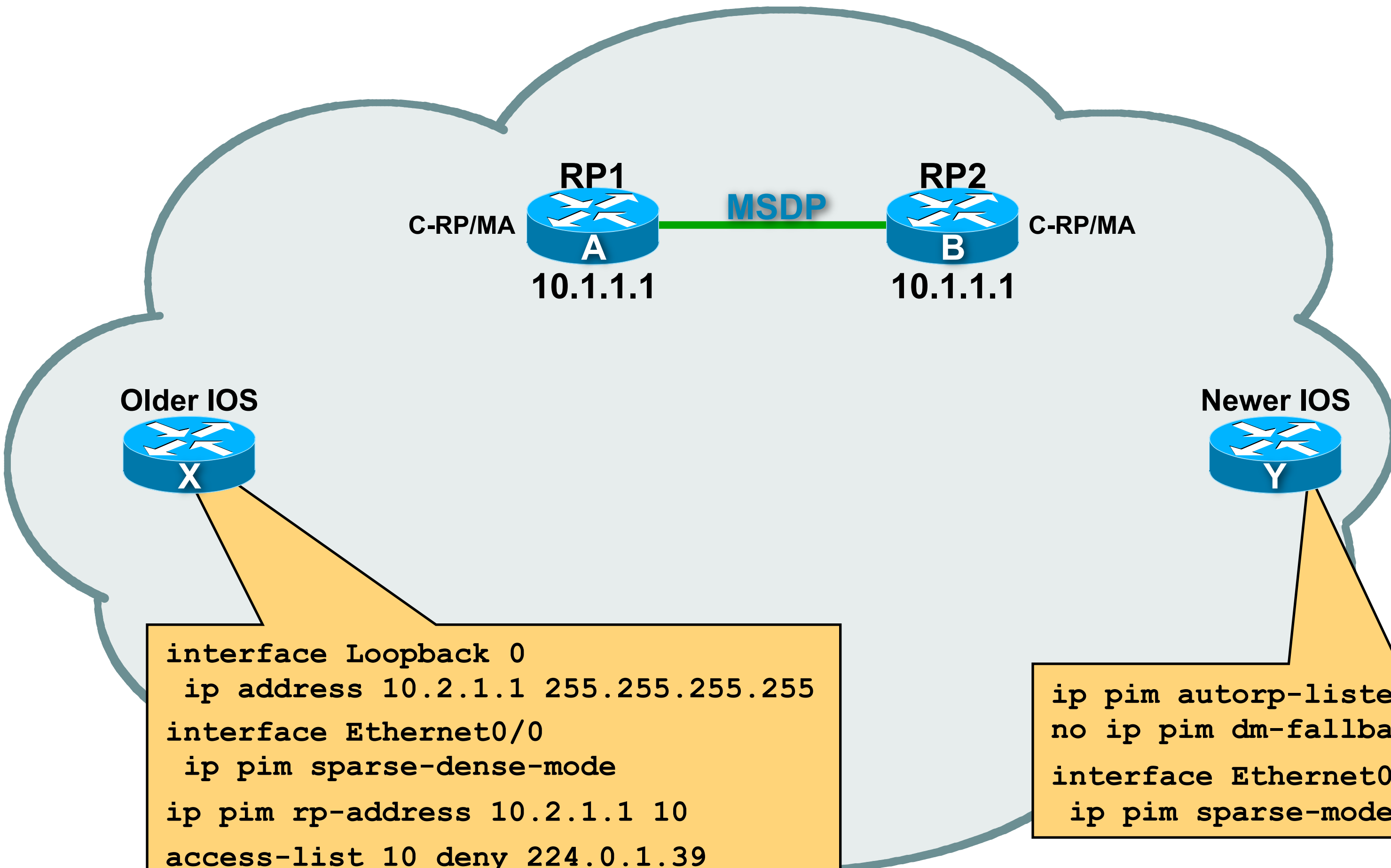
```
ip pim rp-address <local_loopback> 10
access-list 10 deny 224.0.1.39
access-list 10 deny 224.0.1.40
access-list 10 permit any
```

5. Configure Anycast RPs for desired group range
6. Configure Anycast RPs as Auto-RP C-RPs

```
ip pim send-rp-announce Loopback0 scope 32 group-list 10
```

   - Loopback0 = Anycast RP Address
     - Anycast-RPs will announce Anycast-RP address via Auto-RP

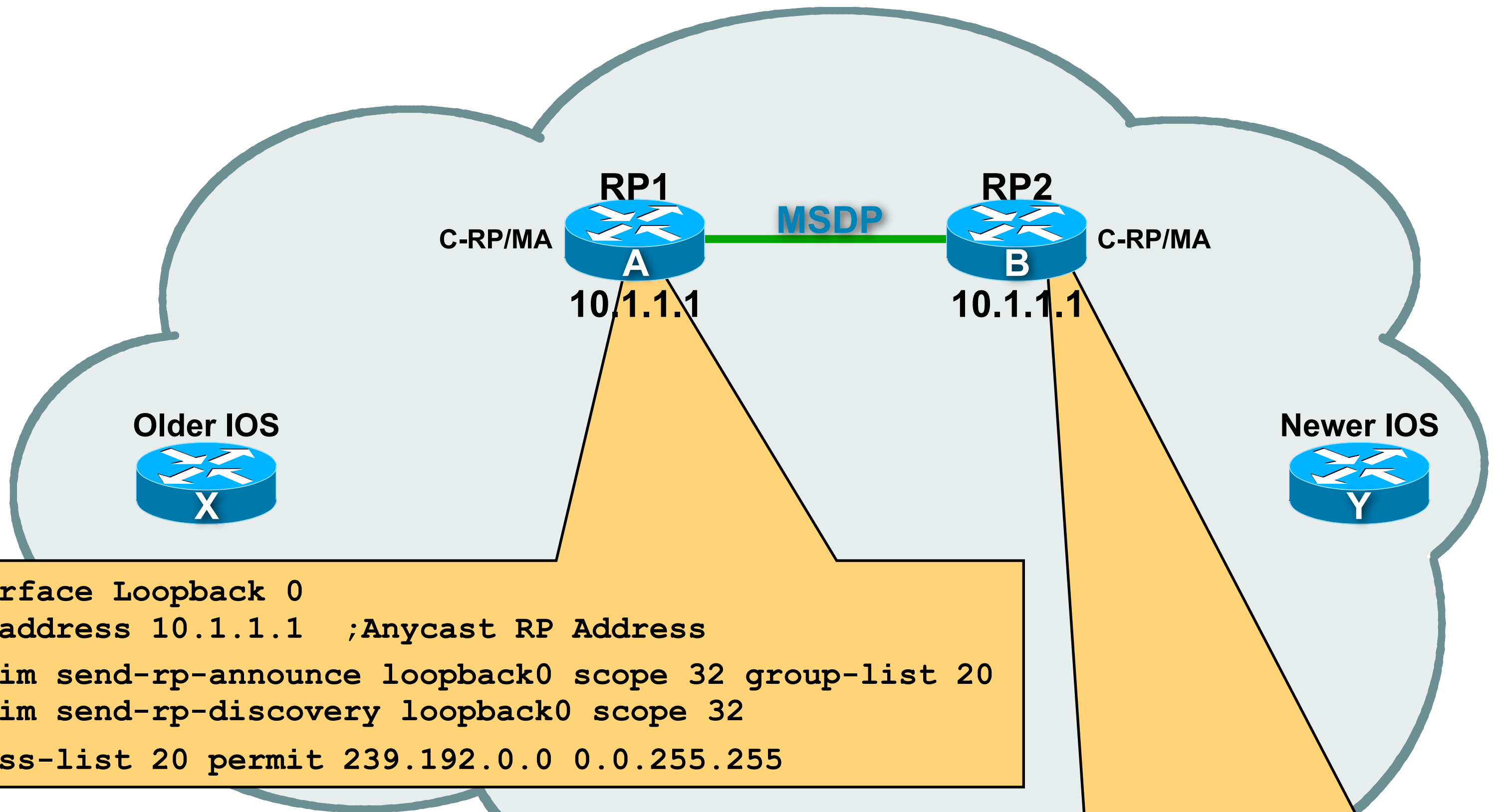
Example Auto-RP and Anycast-RP




```
interface Loopback 0
 ip address 10.2.1.1 255.255.255.255
interface Ethernet0/0
 ip pim sparse-dense-mode
ip pim rp-address 10.2.1.1 10
access-list 10 deny 224.0.1.39
access-list 10 deny 224.0.1.40
access-list 10 permit any
```

```
ip pim autorp-listener
no ip pim dm-fallback
interface Ethernet0/0
 ip pim sparse-mode
```


Example Auto-RP and Anycast-RP




```
interface Loopback 0
 ip address 10.1.1.1   ;Anycast RP Address
ip pim send-rp-announce loopback0 scope 32 group-list 20
ip pim send-rp-discovery loopback0 scope 32
access-list 20 permit 239.192.0.0 0.0.255.255
```

```
interface Loopback 0
 ip address 10.1.1.1   ; Anycast RP Address
ip pim send-rp-announce loopback0 scope 32 group-list 20
ip pim send-rp-discovery loopback0 scope 32
access-list 20 permit 239.192.0.0 0.0.255.255
```


RP Engineering

Avoid Dense Mode Fallback

Dense Mode Fallback

- Caused by loss of local RP information in older Cisco IOS releases
  - Entry in Group-to-RP mapping cache times out
- Can happen when:
  - All C-RPs fail
  - Auto-RP/BSR mechanism fails
    - Generally a result of network congestion
- Group is switched over to Dense mode
  - Dense mode state is created in the network
  - Dense mode flooding begins if interfaces configured as ip pim sparse-dense-mode

Avoiding DM Flooding

- Use global command
  - ip pim autorp listener → Recommended
  - Added support for Auto-RP Environments
    - Modifies interface behavior
      - Forces interfaces to **always** use DM for Auto-RP groups
      - Only needed if Auto-RP is to be used
  - Available 12.3(4)T, 12.2(28)S
- Use with interface command
  - ip pim sparse-mode → Recommended
    - Prevents DM Flooding
    - Does not prevent DM Fallback!

Avoiding DM Flooding

- Prior to IOS 12.3(4)T, 12.2(28)S
- Use RP-of-last-resort

  Assign local Loopback as RP-of-last-resort on each router

  Example

```
ip pim rp-address <local_loopback> 10
access-list 10 deny 224.0.1.39
access-list 10 deny 224.0.1.40
access-list 10 permit any
```

  Must also use ip pim sparse-dense mode interface command to support Auto-RP

Avoiding DM Fallback

- New IOS global command

  no ip pim dm-fallback → Recommended

- Totally prevents DM Fallback!!

  No DM Flooding since all state remains in SM

- Default RP Address = 0.0.0.0 [nonexistent]

  Used if all RPs fail

    Results in loss of Shared Tree

    All SPTs remain active

- Available 12.3(4)T, 12.2(28)S

Avoiding DM Fallback

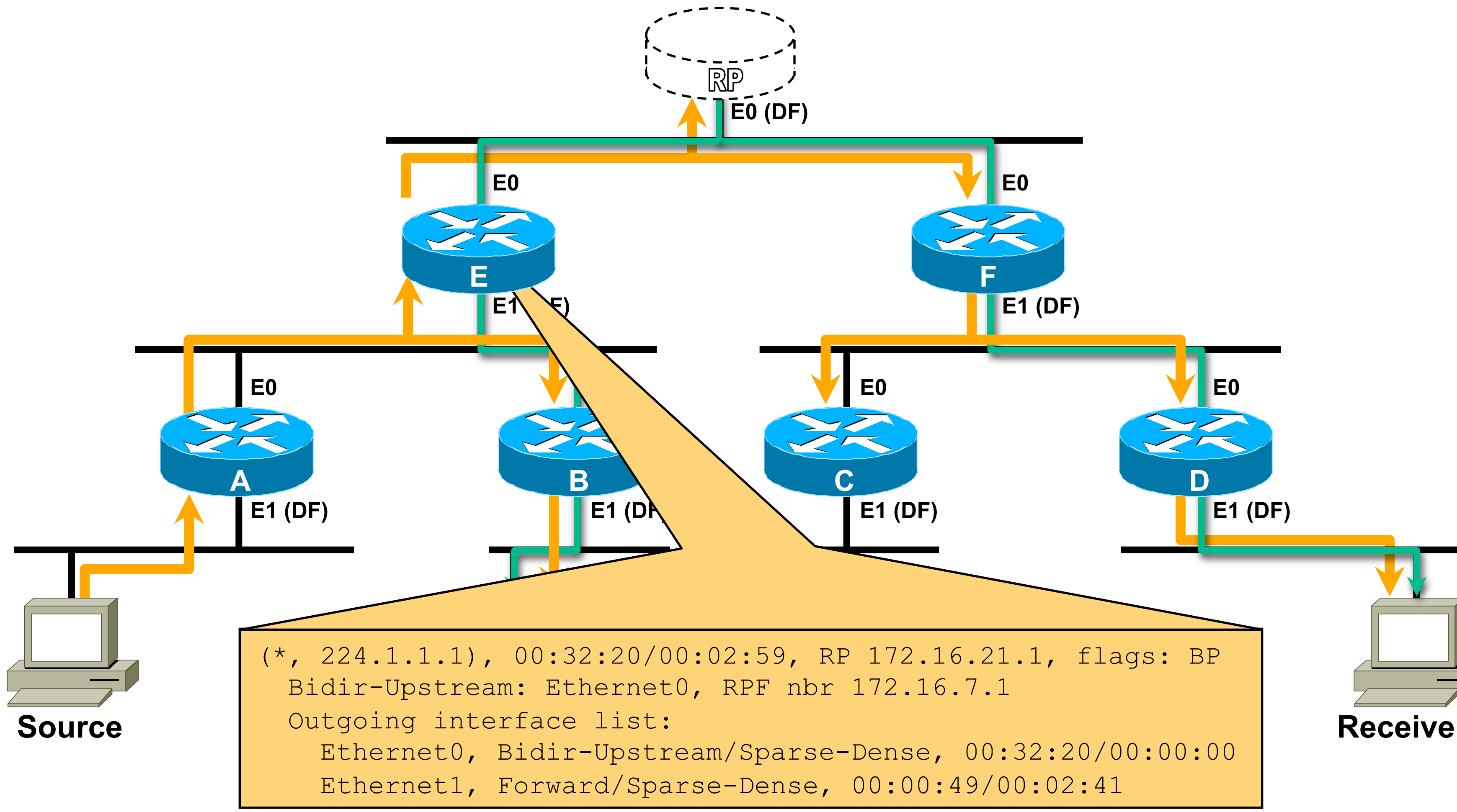
- Prior to IOS 12.3(4)T, 12.2(28)S
- Define an "RP-of-last-resort"
  - Configure as a Static RP on every router
    - Will only be used if all Candidate-RPs fail
    - Can be a dummy address or local Loopback
      - Recommendation: Use local Loopback on each router
  - **Must** use ACL to avoid breaking Auto-RP!

```
ip pim rp-address <RP-of-last-resort> 10
access-list 10 deny 224.0.1.39
access-list 10 deny 224.0.1.40
access-list 10 permit any
```


RP Engineering

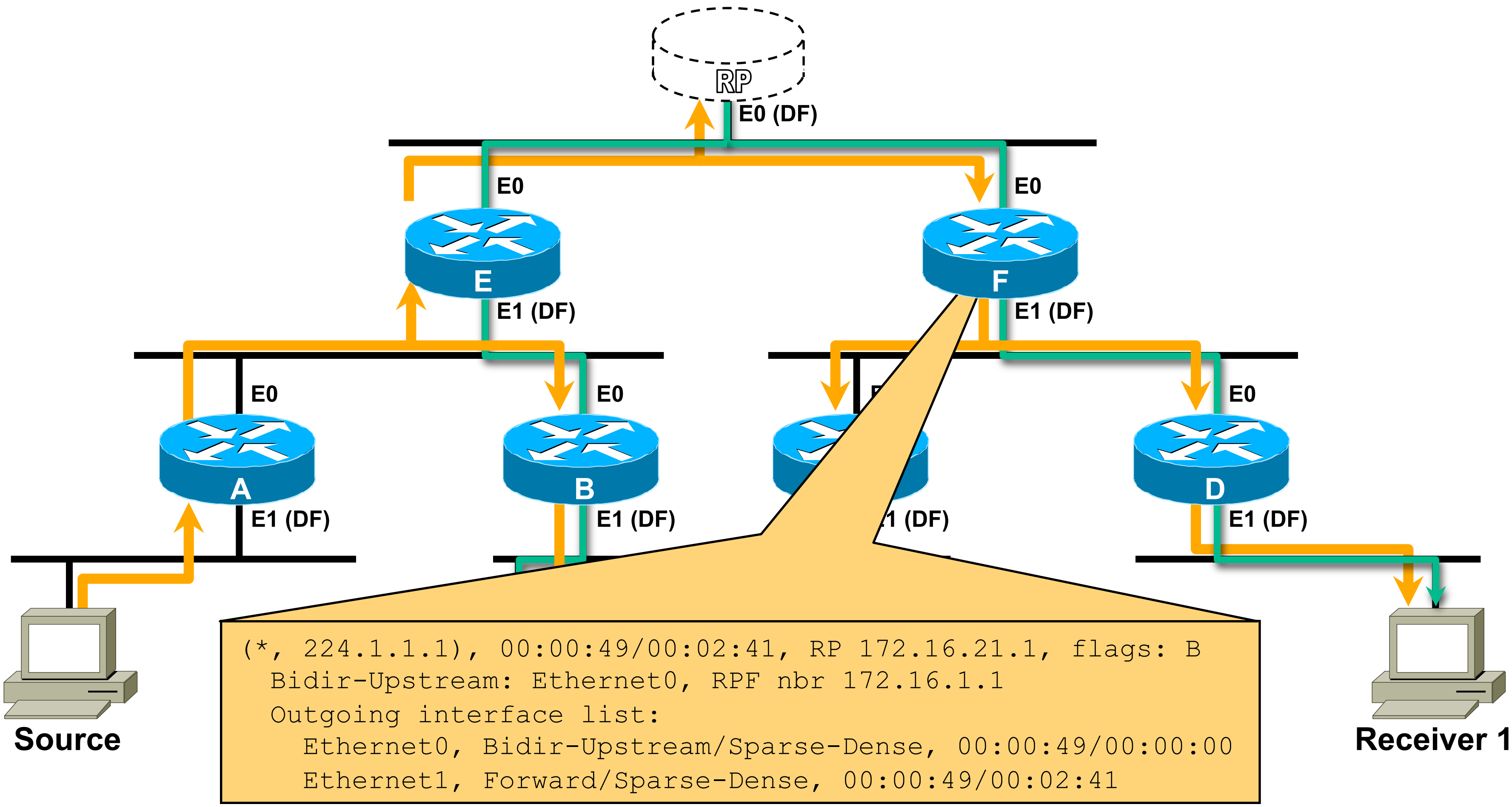
Phantom BiDir RPs

BiDir PIM—Phantom RP



Router “E” Forwards Traffic onto Core LAN Segment

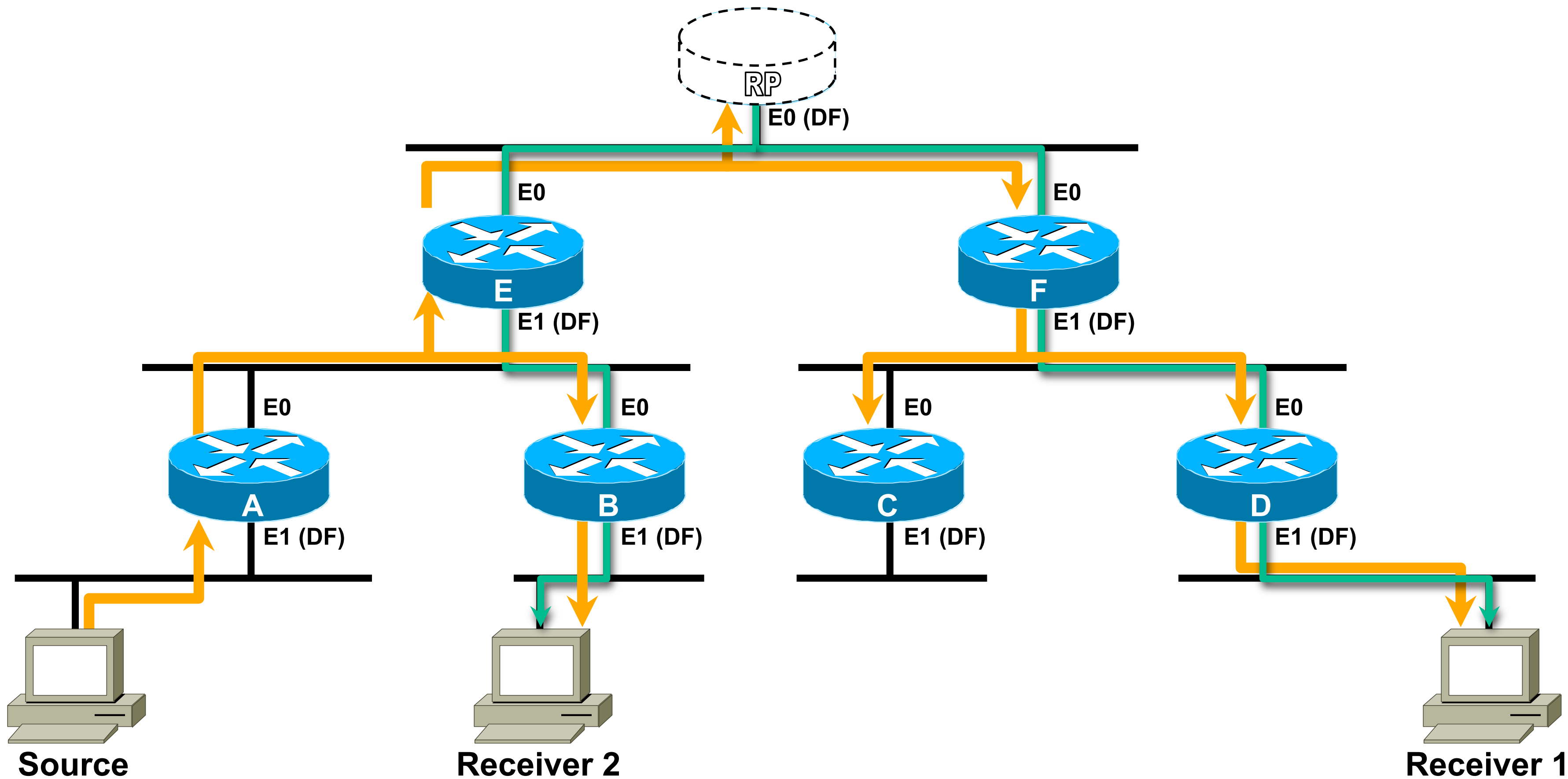
BiDir PIM—Phantom RP



Router "F" Forwards Traffic on Down the Shared Tree ala Normal PIM-SM

RP Doesn't Even Have to Physically Exist

BiDir PIM—Phantom RP

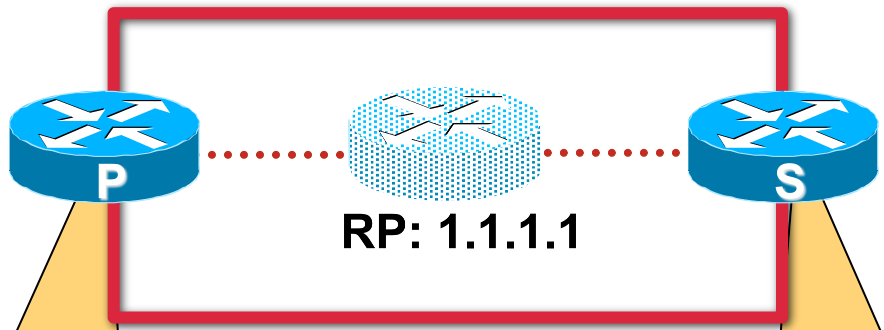


Question: Does a Bidir RP even have to physically exist?

Answer: No. It can just be a phantom address.

Phantom RP on Point-to-Point Core

Static Route Method

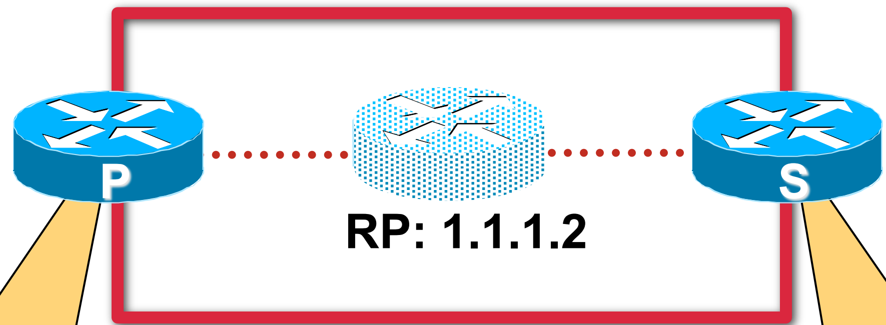



```
ip multicast-routing

interface Loopback0
 ip address 11.0.0.1 255.255.255.255
 ip pim sparse-mode

router ospf 11
 redistribute static subnets

ip route 1.1.1.1 255.255.255.255 Loopback0

ip pim bidir-enable
ip pim rp-address 1.1.1.1 bidir
```

```
ip multicast-routing

interface Loopback0
 ip address 11.0.0.2 255.255.255.255
 ip pim sparse-mode

router ospf 11
 redistribute static subnets

ip route 1.1.1.0 255.255.255.254 Loopback0

ip pim bidir-enable
ip pim rp-address 1.1.1.1 bidir
```


Phantom RP on Point-to-Point Core

Netmask Method



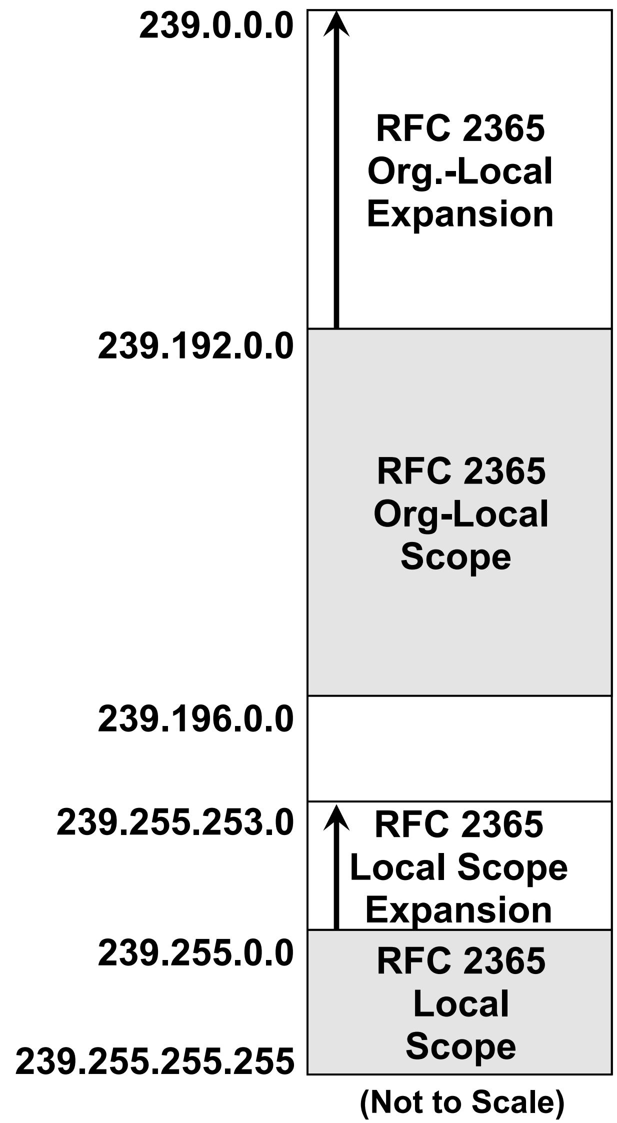

```
ip multicast-routing
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.252
 ip pim sparse-mode
 ip ospf network point-to-point
!
router ospf 11
 network 1.1.1.0 0.0.0.3 area 0
 network 10.1.1.0 0.0.0.255 area 0
 network 10.1.2.0 0.0.0.255 area 0
!
ip pim bidir-enable
ip pim rp-address 1.1.1.1
ip pim rp-address 1.1.1.2 bidir
```

```
ip multicast-routing
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.248
 ip pim sparse-mode
 ip ospf network point-to-point
!
router ospf 11
 network 1.1.1.0 0.0.0.7 area 0
 network 10.1.1.0 0.0.0.255 area 0
 network 10.1.2.0 0.0.0.255 area 0
!
ip pim bidir-enable
ip pim rp-address 1.1.1.1
ip pim rp-address 1.1.1.2 bidir
```

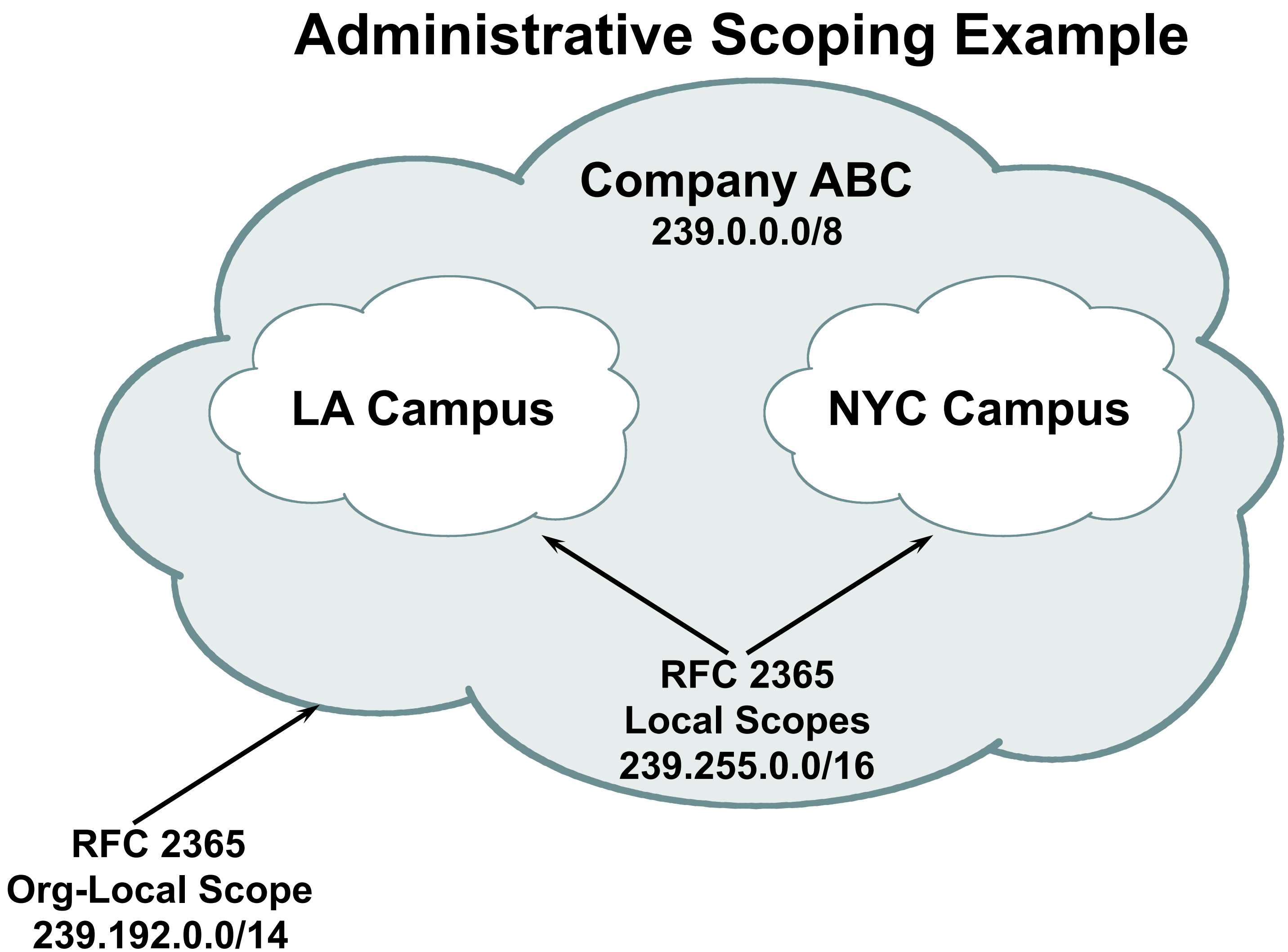

Deploying Administratively Scoped Zones

Administratively Scoped Address Range

- Address Range: 239.0.0.0/8
  - Private multicast address space
  - Similar to RFC1918 private unicast address space
- RFC 2365 Administratively Scoped Zones
  - Organization-Local Scope (239.192/14)
    - Largest scope within the Enterprise network (i.e. Enterprise-wide)
    - Expands downward in address range
  - Local Scope (239.255/16)
    - Smallest possible scope within the Enterprise network
    - Expands downward in address range
    - Other scopes may be equal but not smaller



Administratively Scoped Address Range

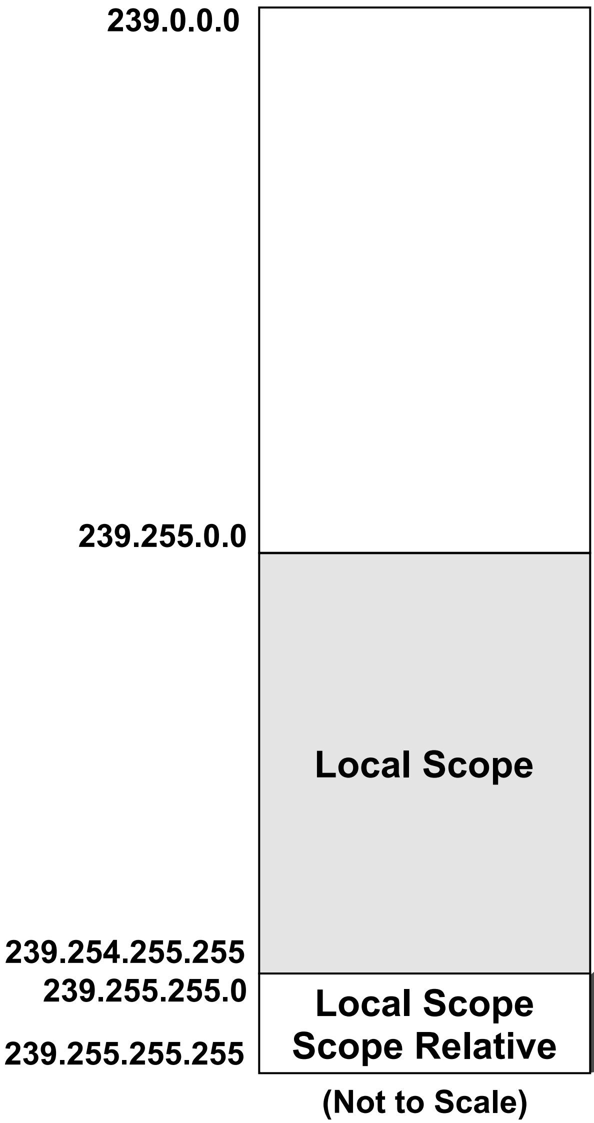


Scope Relative Addresses—RFC 2365

Top 256 Addresses of Every Admin. Scope Range

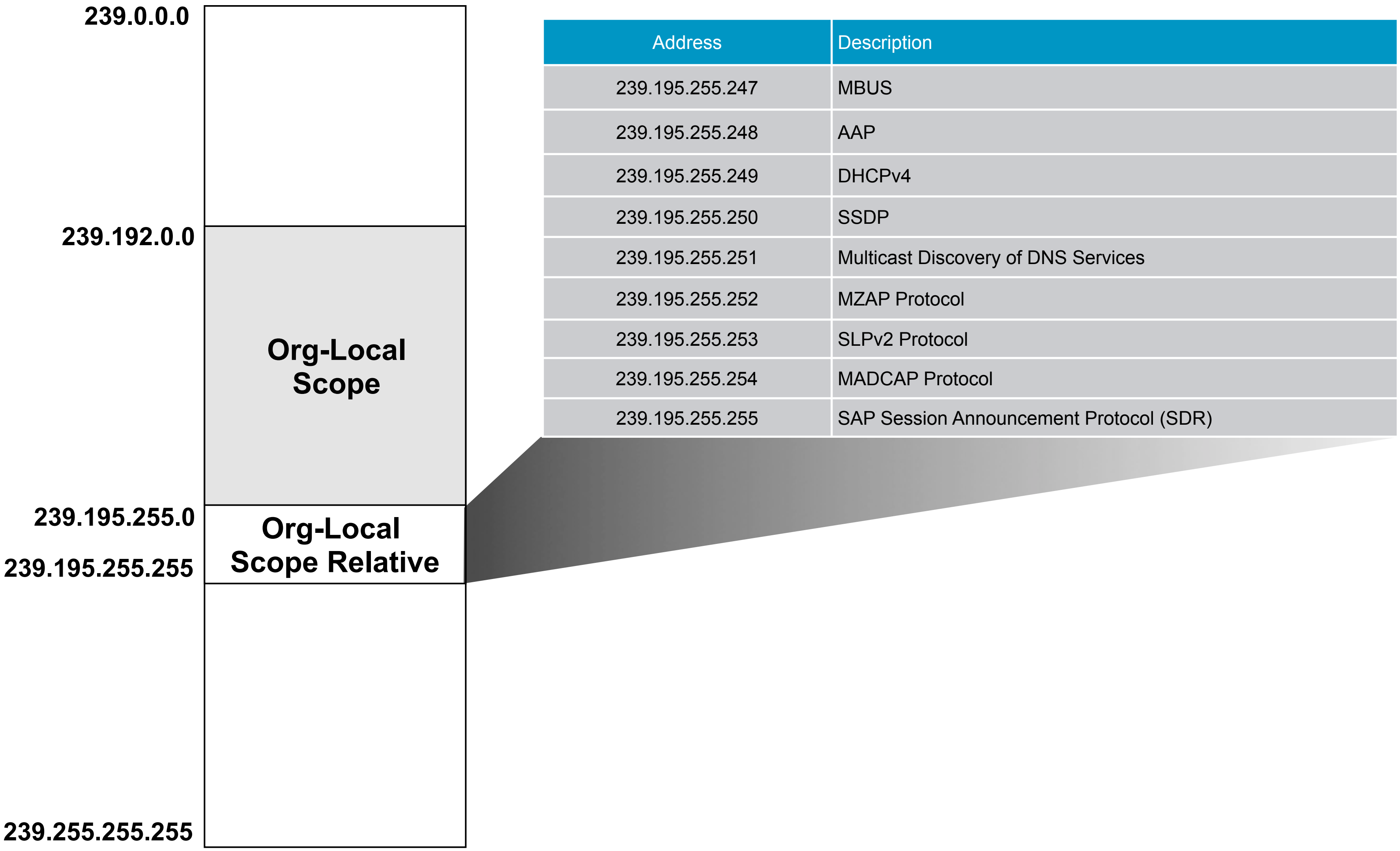
Last Octet	Offset	Description
.255	0	SAP Session Announcement Protocol (SDR)
.254	1	MADCAP Protocol
.253	2	SLPv2 Protocol
.252	3	MZAP Protocol
.251	4	Multicast Discovery of DNS Services
.250	5	SSDP
.249	6	DHCPv4
.248	7	AAP
.247	8	MBUS
	9 - 255	Unassigned

Scope Relative Example—Local Scope



Address	Description
239.255.255.247	MBUS
239.255.255.248	AAP
239.255.255.249	DHCPv4
239.255.255.250	SSDP
239.255.255.251	Multicast Discovery of DNS Services
239.255.255.252	MZAP Protocol
239.255.255.253	SLPv2 Protocol
239.255.255.254	MADCAP Protocol
239.255.255.255	SAP Session Announcement Protocol (SDR)

Scope Relative Example—Org-Local Scope



(Not to Scale)

Address	Description
239.195.255.247	MBUS
239.195.255.248	AAP
239.195.255.249	DHCPv4
239.195.255.250	SSDP
239.195.255.251	Multicast Discovery of DNS Services
239.195.255.252	MZAP Protocol
239.195.255.253	SLPv2 Protocol
239.195.255.254	MADCAP Protocol
239.195.255.255	SAP Session Announcement Protocol (SDR)

Administratively-Scoped Zones

- Used to limit:

  High-BW sources to local site

  Control sensitive multicast traffic

- Simple scoped zone example:

  239.193.0.0/16 = Campus Scope

  239.194.0.0/16 = Region Scope

  239.195.0.0/16 = Organization-Local (Enterprise) Scope

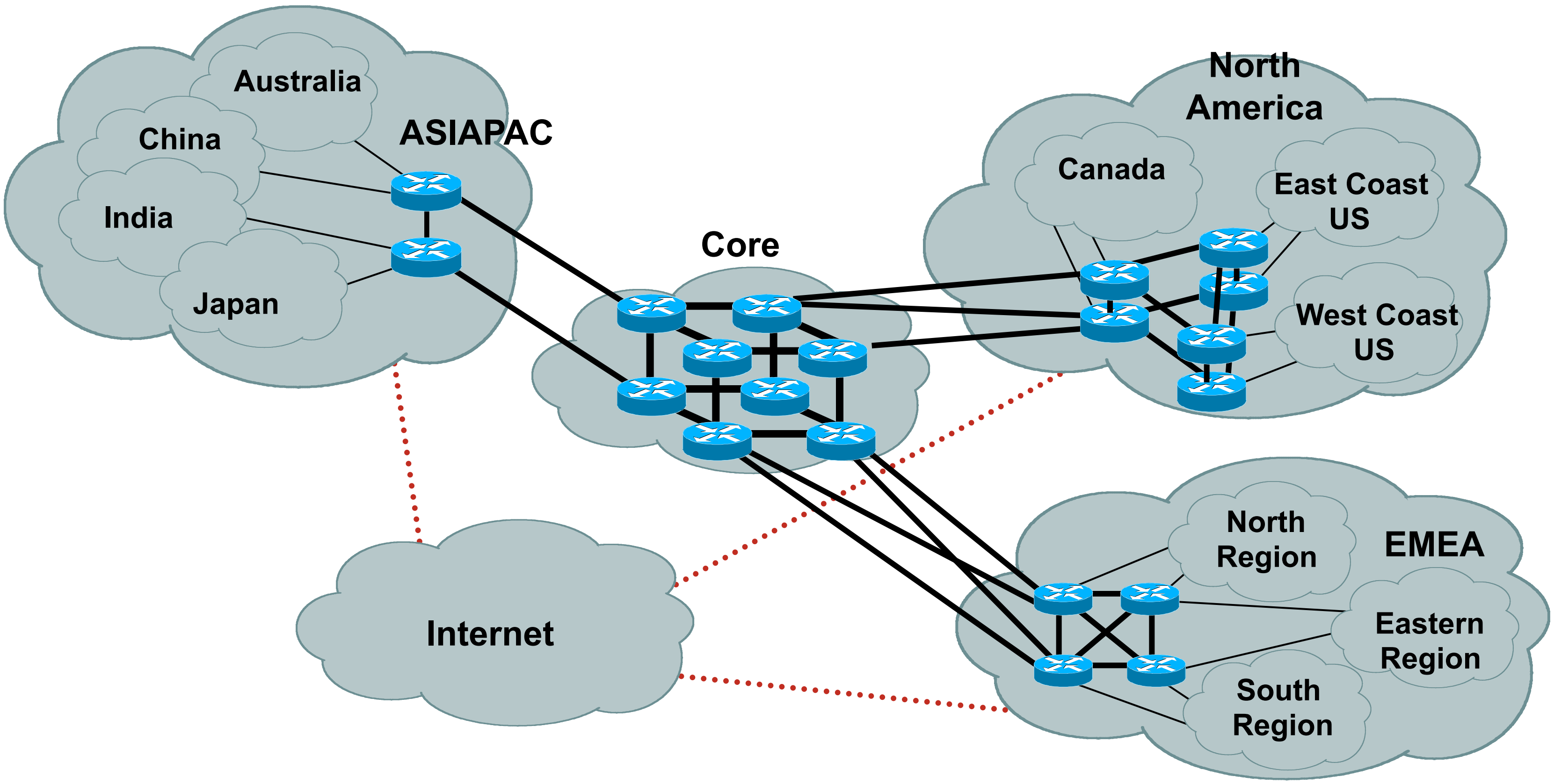
  224.1.0.0 - 238.255.255.255 = Global scope (Internet) zone

    High-BW sources use Site-Local scope

    Low-Med. BW sources use Org.-Local scope

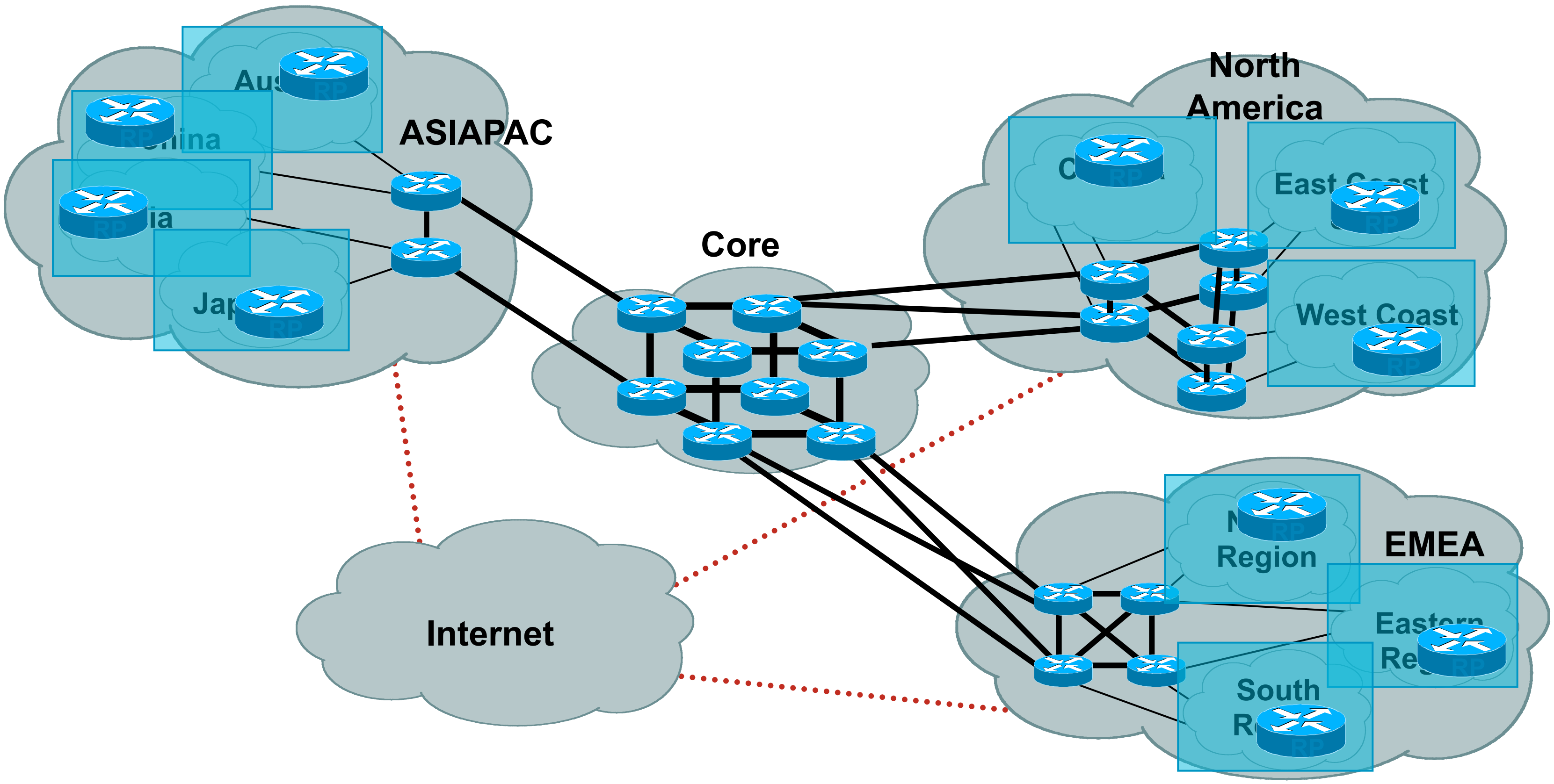
    Internet-wide sources use Global scope

Administratively-Scoped Zones Example



Administratively-Scoped Zones Example

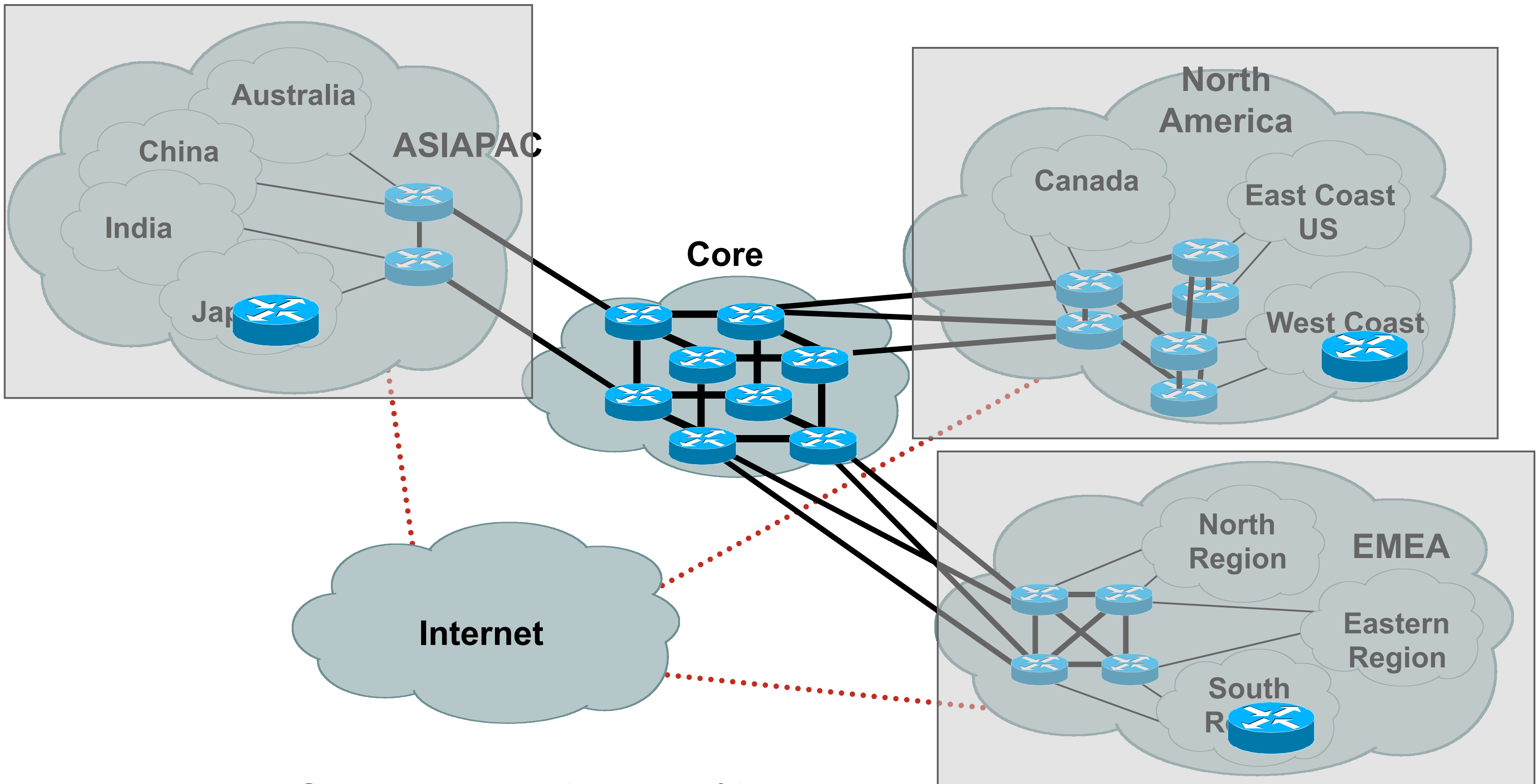
Level 1: Campus Scope



- Campus scope: 239.193.x.x/16
- RP per campus

Administratively-Scoped Zones Example

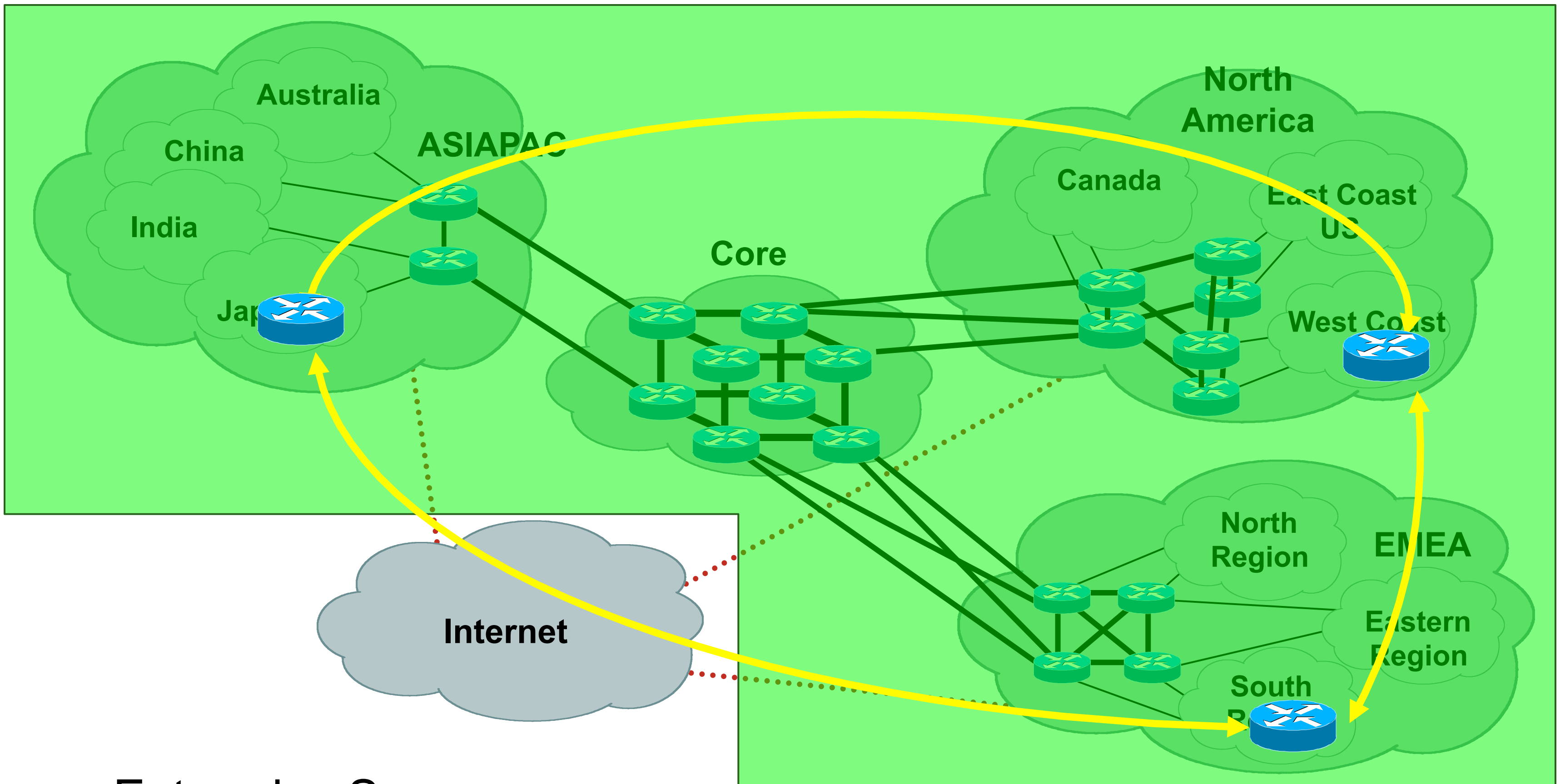
Level 2: Regional Scope



- Regional Scope: 239.194.x.x/16
- RP per Region

Administratively-Scoped Zones Example

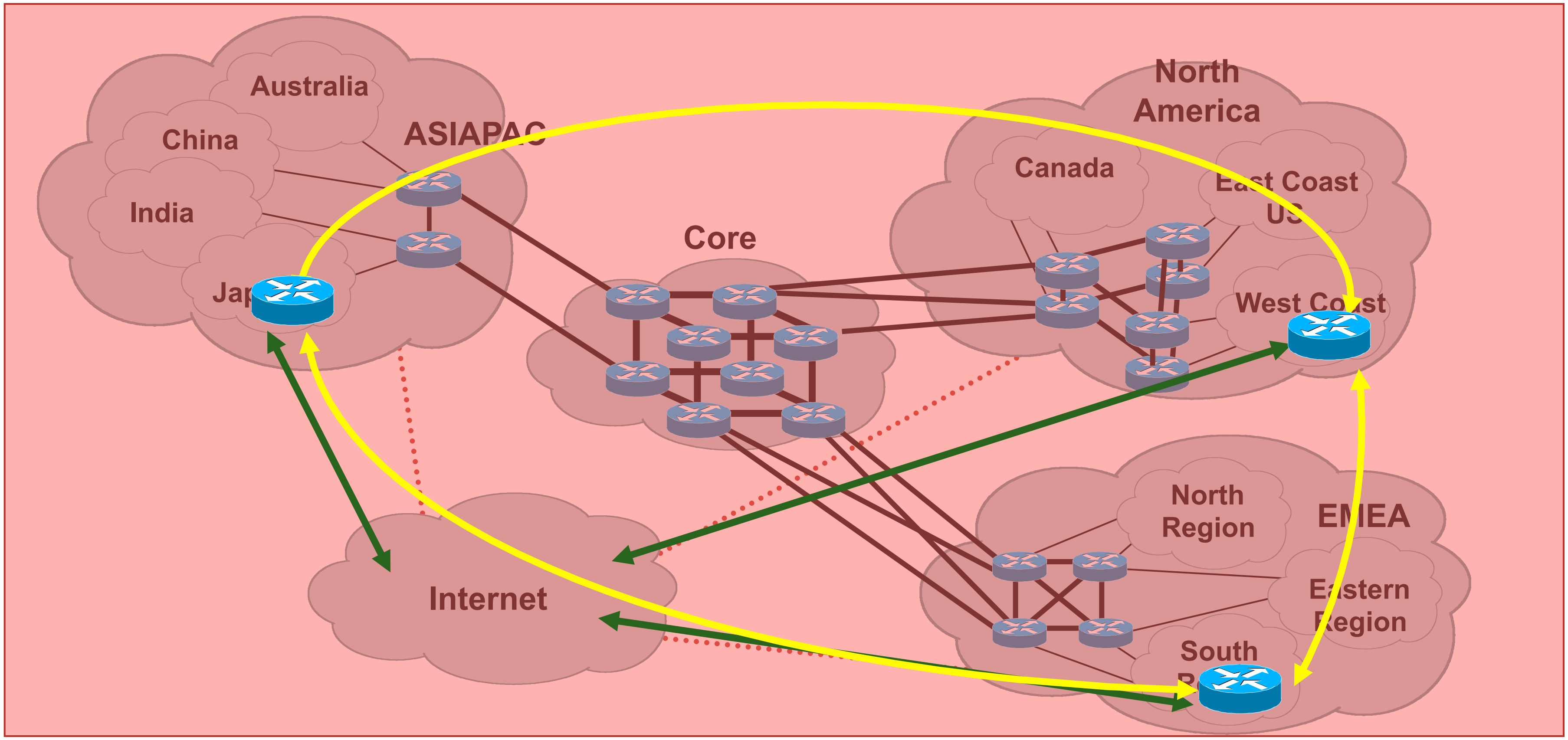
Level 3: Enterprise Scope



- Enterprise Scope:
  239.195.x.x/16
- Multiple Enterprise RPs (via MSDP full mesh)

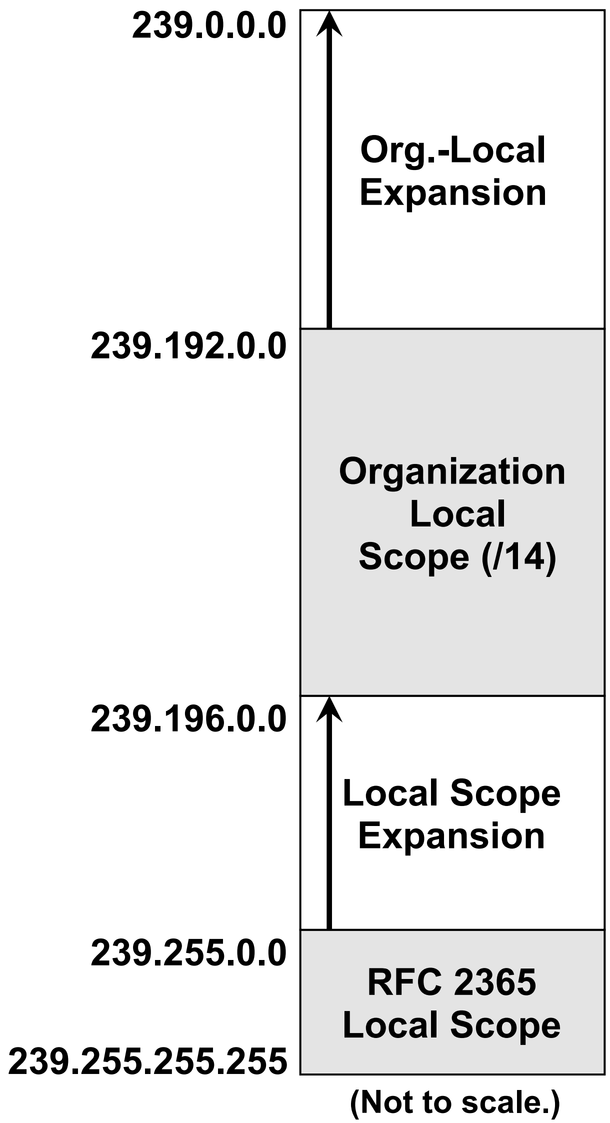
Administratively-Scoped Zones Example

Level 4: Internet Global Scope



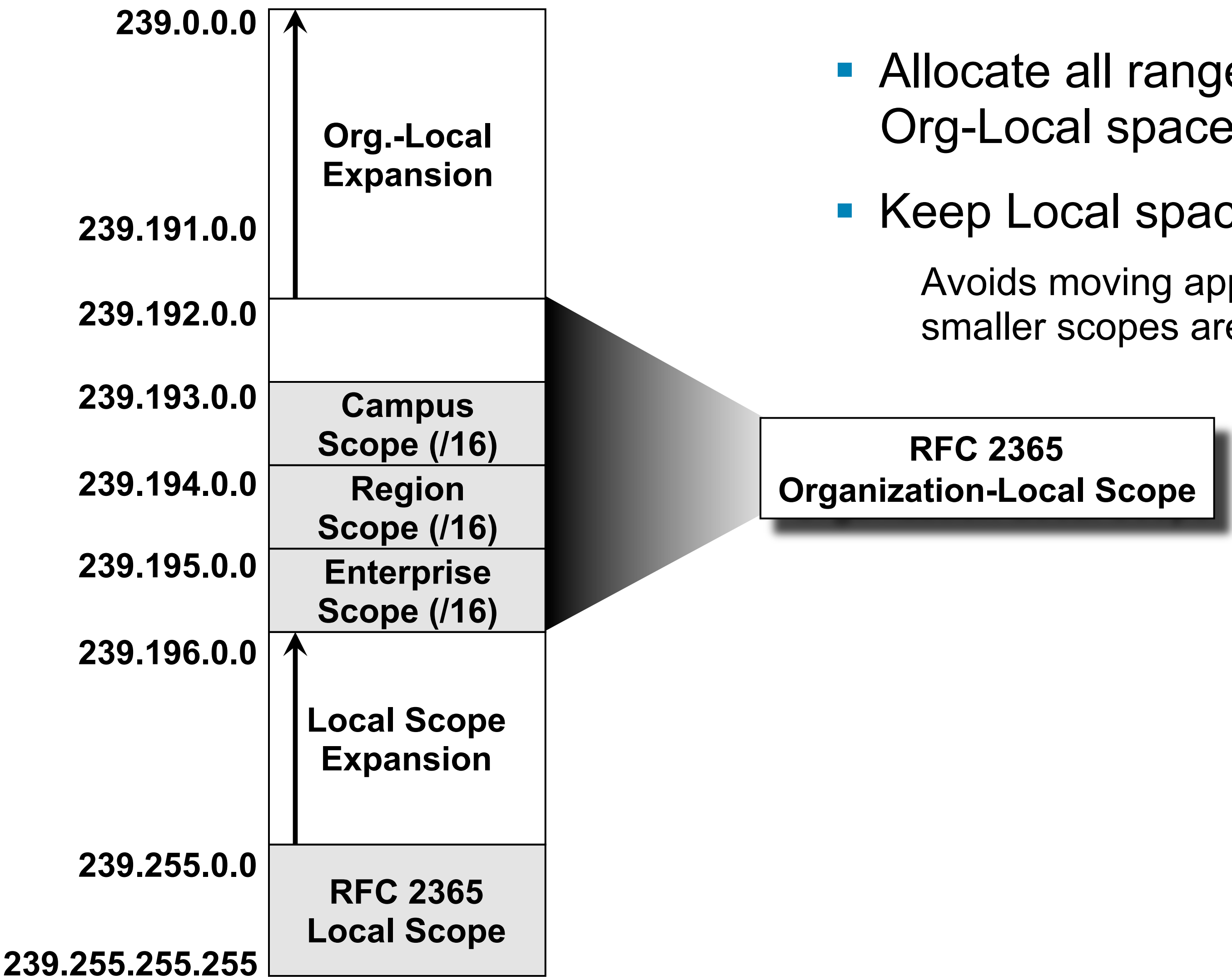
- Global Scope: 224.0.[2-255].x–238.255.255.255
- Multiple Global RPs (via MSDP full mesh)
- MSDP connectivity to SP network

Administratively Scoped Address Range



- RFC 2365 Administratively Scoped Zones
  - Organization-Local Scope (239.192/14)
  - Expands downward in address range
  - Local Scope (239.255/16)
  - Expands downward in address range
  - Smallest possible scope
  - Other scopes may be equal but not smaller

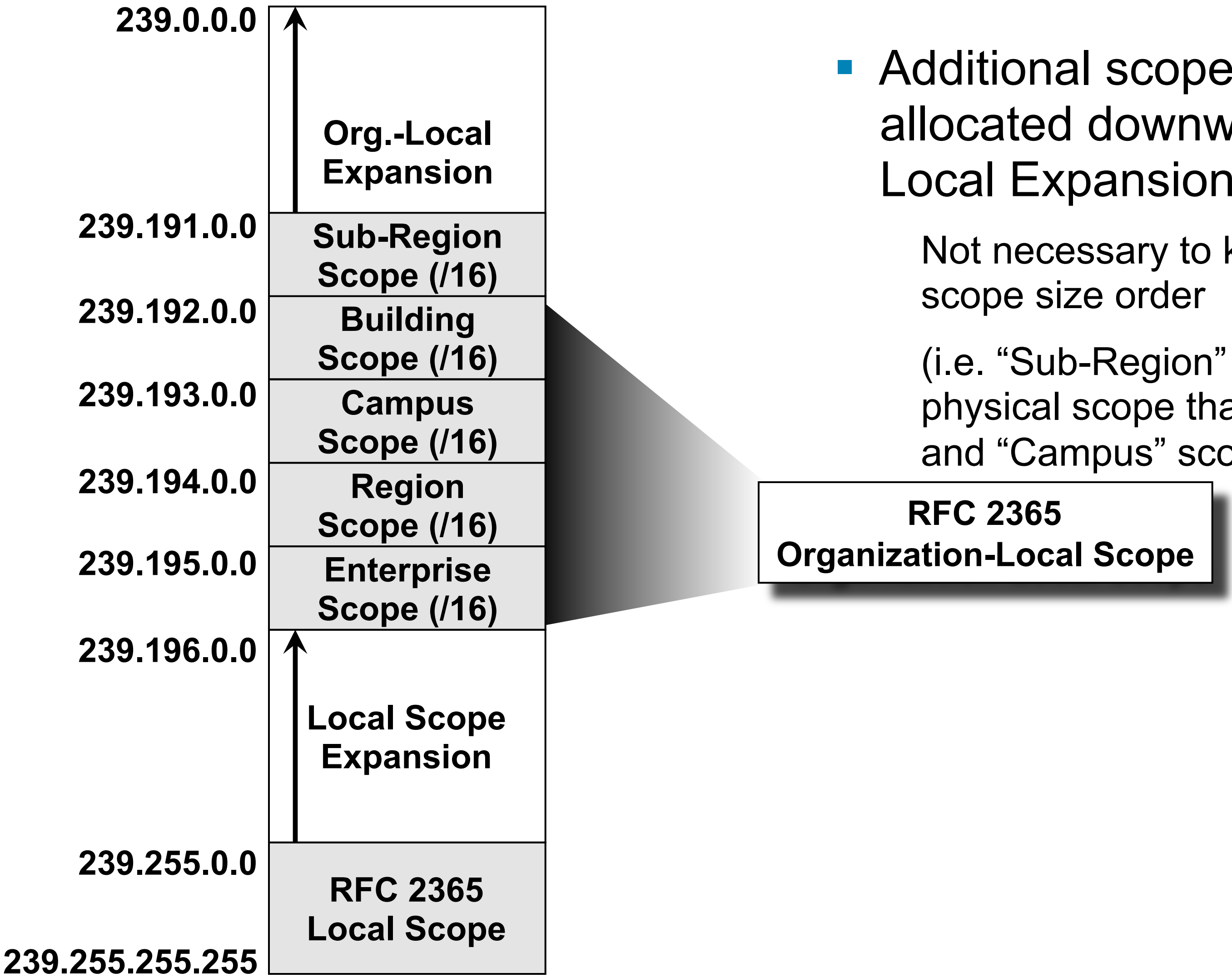
Example Scope Address Assignments



- Allocate all ranges from the Org-Local space
- Keep Local space separate

  Avoids moving applications when smaller scopes are added later

Adding Additional Scopes

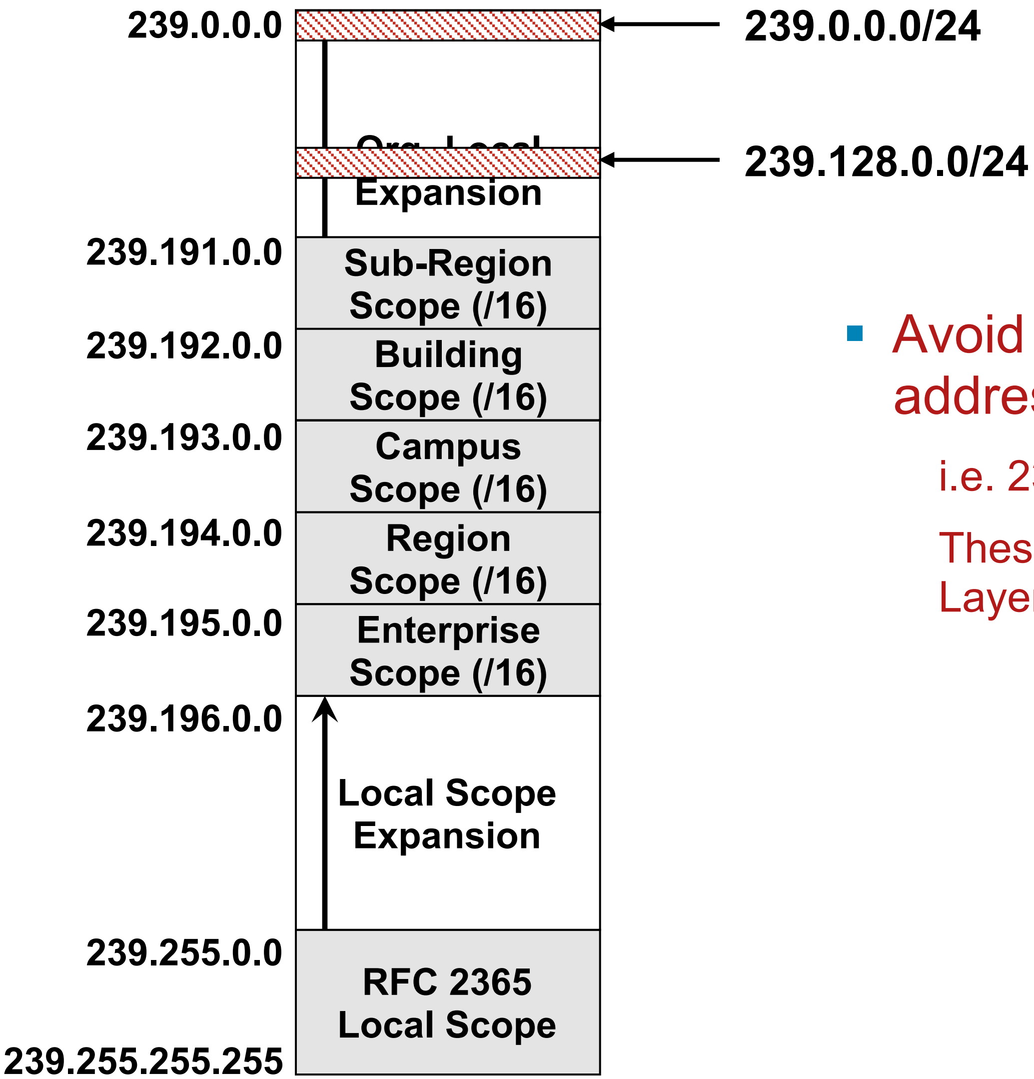


- Additional scope ranges are allocated downward into Org-Local Expansion

  Not necessary to keep ranges in scope size order

  (i.e. "Sub-Region" scope is a larger physical scope than the "Building" and "Campus" scopes)

Address Ranges to Avoid

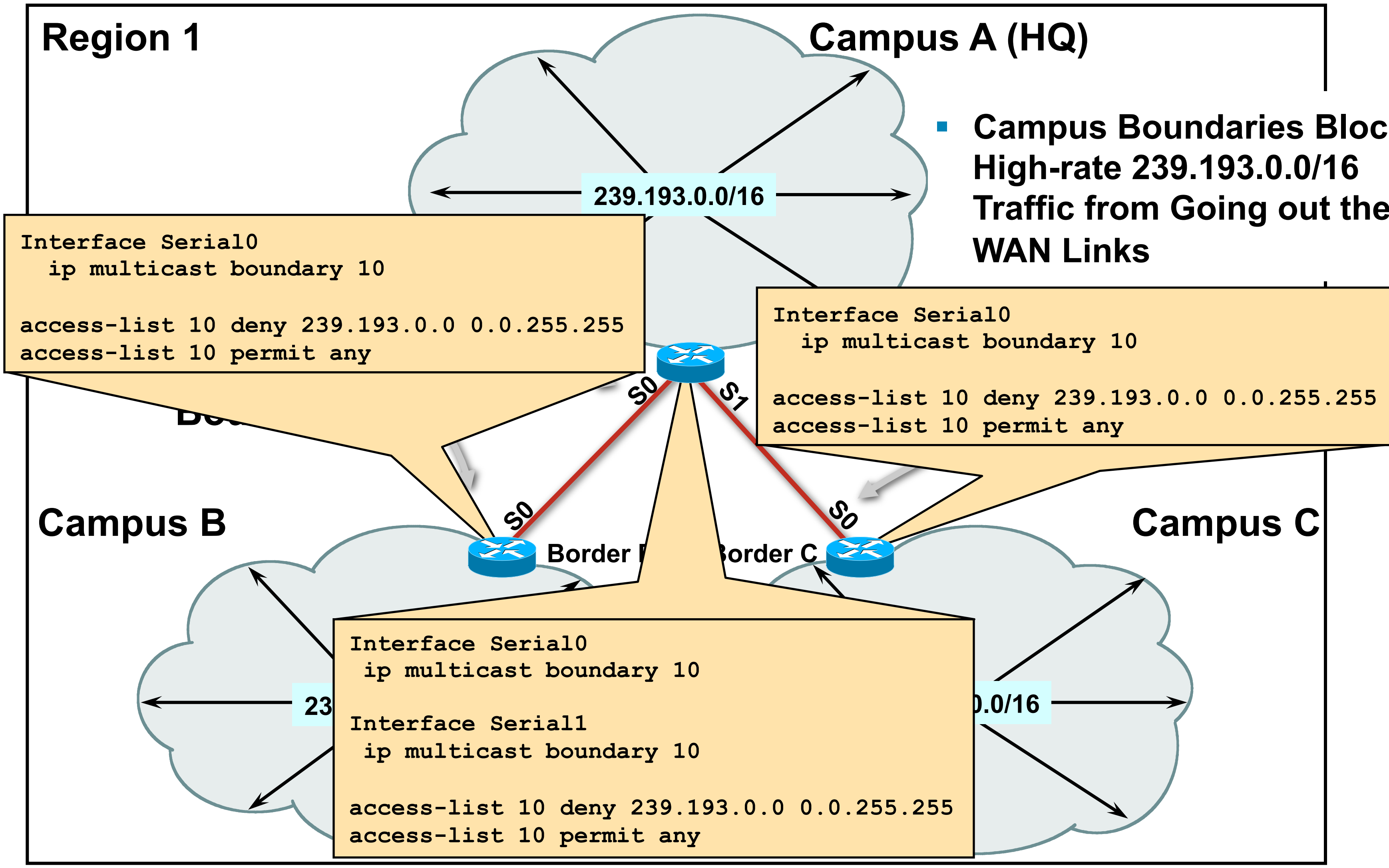


- Avoid ranges that map to a MAC address of 0x0100-5E00-00xx!

  i.e. 239.128.0/24 and 239.0.0/24

  These addresses are always flooded by Layer 2 switches!

Deploying Administratively-Scoped Zones



- Campus Boundaries Block High-rate 239.193.0.0/16 Traffic from Going out the WAN Links

Deploying Administratively-Scoped Zones

Preventing Auto-RP Info Leakage

- Multicast Boundary Command

  ```
  ip multicast boundary <acl> [filter-autorp]
  ```

  New ‘filter-autorp’ option

  - Filters contents of Auto-RP packets
    - Filters both Announcement and Discovery messages
    - C-RP entries that fail <acl> are removed from packet
  - Prevents C-RP information from leaking in/out of scoped zone
  - Greatly simplifies Admin. Scoped Zone support in Auto-RP
  - Available in 12.0(22)S, 12.2(12)

Deploying Administratively-Scoped Zones

Preventing Auto-RP Info Leakage

- How 'filter-autorp' option works:

  For each RP Entry in Auto-RP packet:

    If group-range in RP-Entry **'intersects'** any 'denied' group-range in the Multicast Boundary ACL, delete RP Entry from Auto-RP packet

  If resulting Auto-RP packet is non-empty, forward across multicast boundary

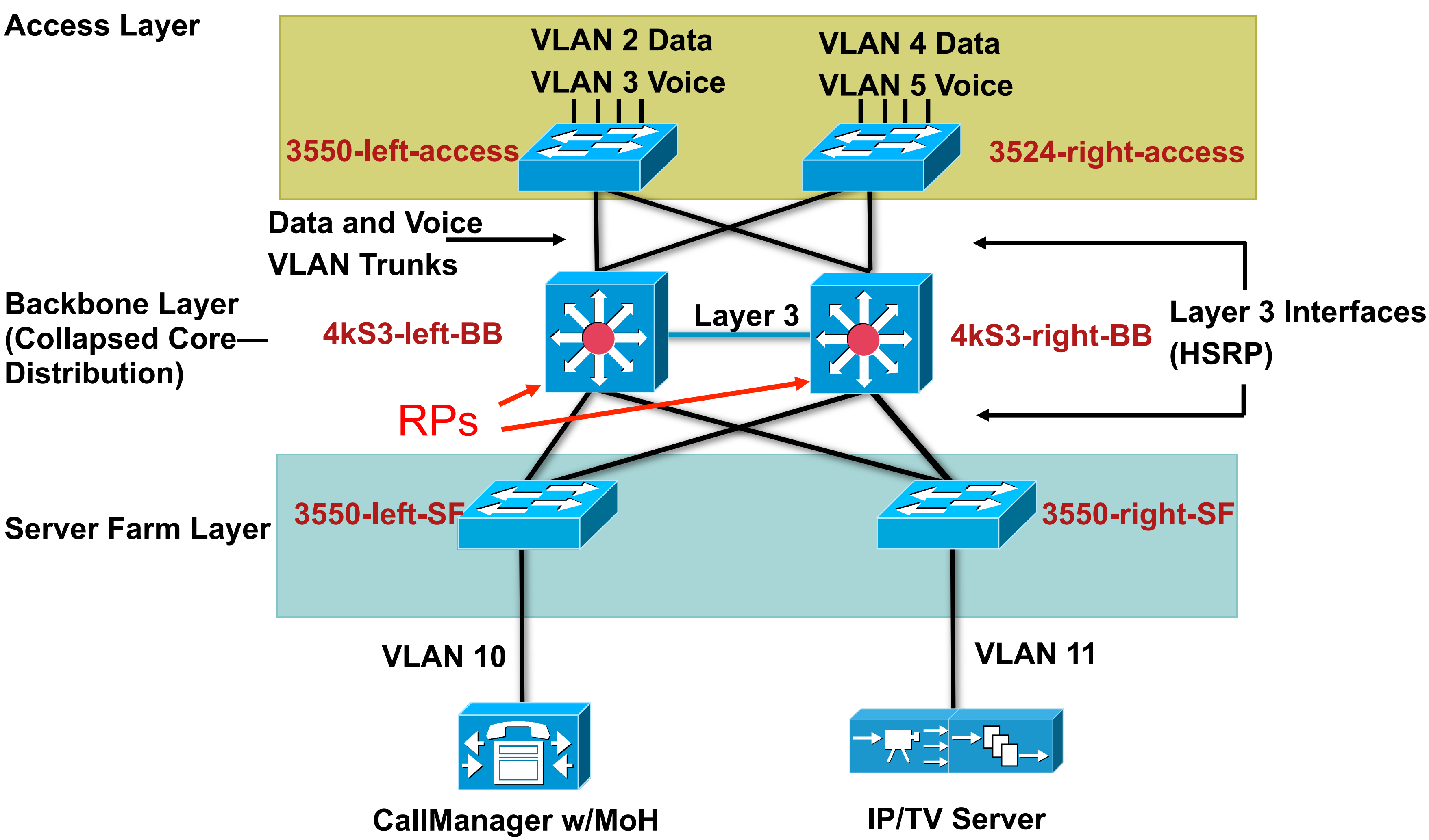
Deploying Administratively-Scoped Zones

Preventing Auto-RP Info Leakage

- Using Multicast Boundary 'filter-autorp'
  - Avoid Auto-RP Group-Range Overlaps
    - Overlapping ranges can "intersect" denied ranges at multicast boundaries
      - Can cause unexpected Auto-RP info filtering at multicast boundaries
      - Results in loss of Auto-RP info to other parts of network
  - Rule of Thumb:
    - Make sure Auto-RP Group-Ranges match exactly any Multicast Boundary Ranges!
      - (i.e. don't use overlapping Auto-RP group ranges)

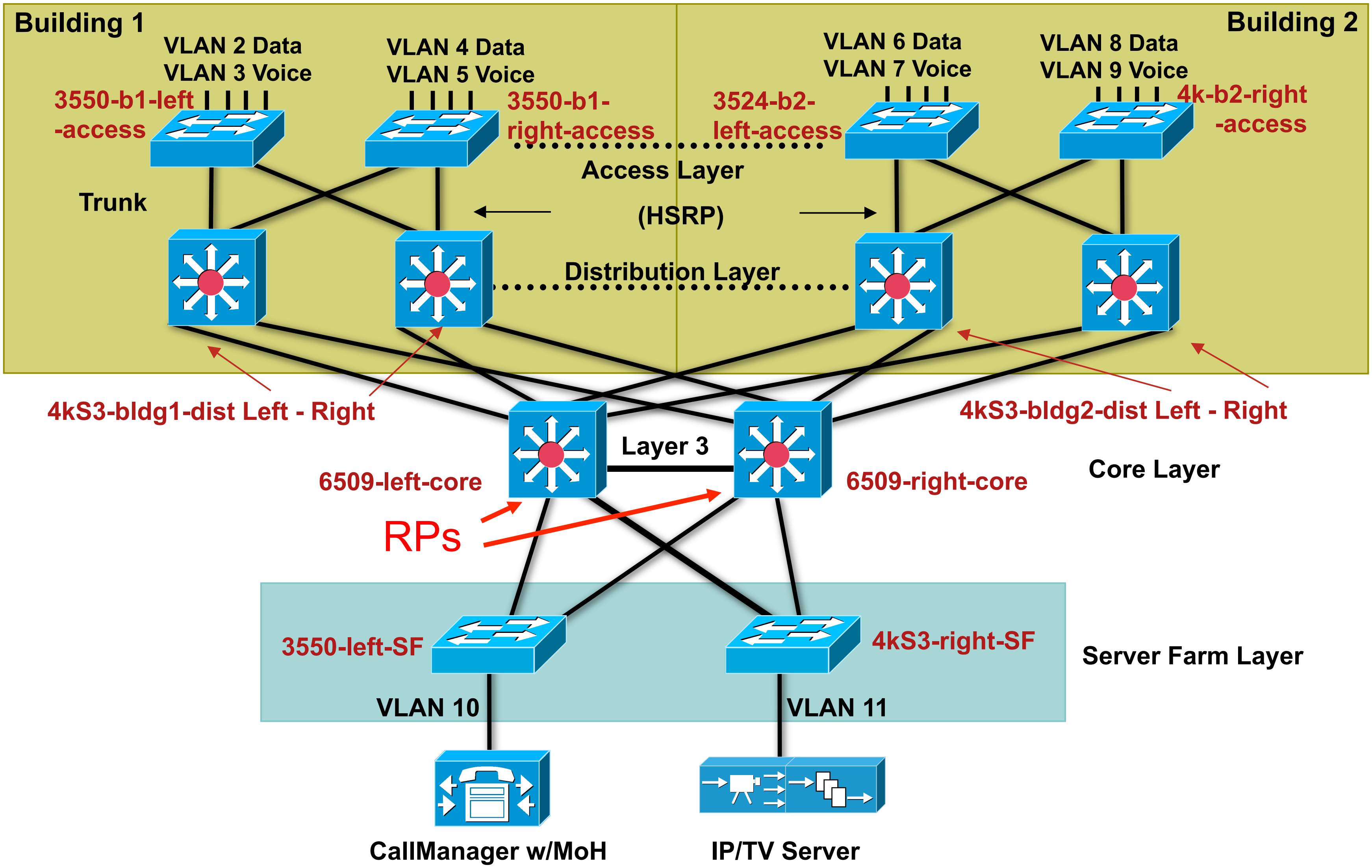
Basic Campus Designs

Small Campus Design

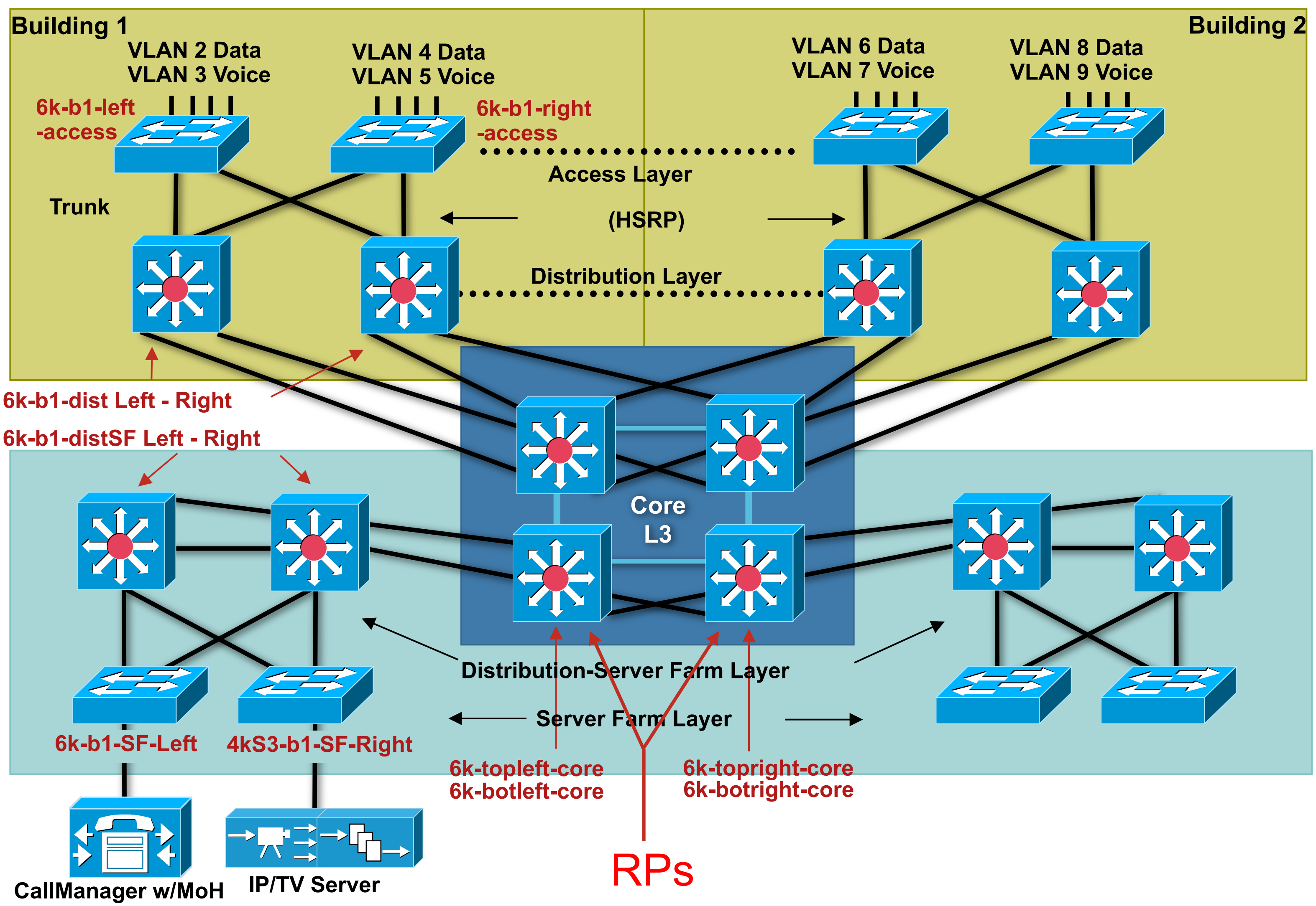


Note: Typically <1000 mroutes

Medium Campus Design



Large Campus Design



Putting It All Together: Full Any-to-Any Connectivity

- Multicast Protocols: Integration of SSM, Bidir PIM and Sparse Mode PIM based on application
- Group enabling/disabling: Different group ranges bound to application needs
- Scope group ranges by geography and bandwidth requirements

Summary of the Day

- When, why and how of Multicast
  - SSM
    - one-to-many
  - BiDir
    - many-to-many applications
- Know your application requirements
- Start with an Addressing Plan
- Understand the Multicast Capabilities of your L2
- Use your address plan to scope content
- Have fun!

Thank You!

Recommended Reading

- Continue your Cisco Live learning experience with further reading from Cisco Press
- Check the Recommended Reading flyer for suggested books

Available Onsite at the Cisco Company Store

More Information

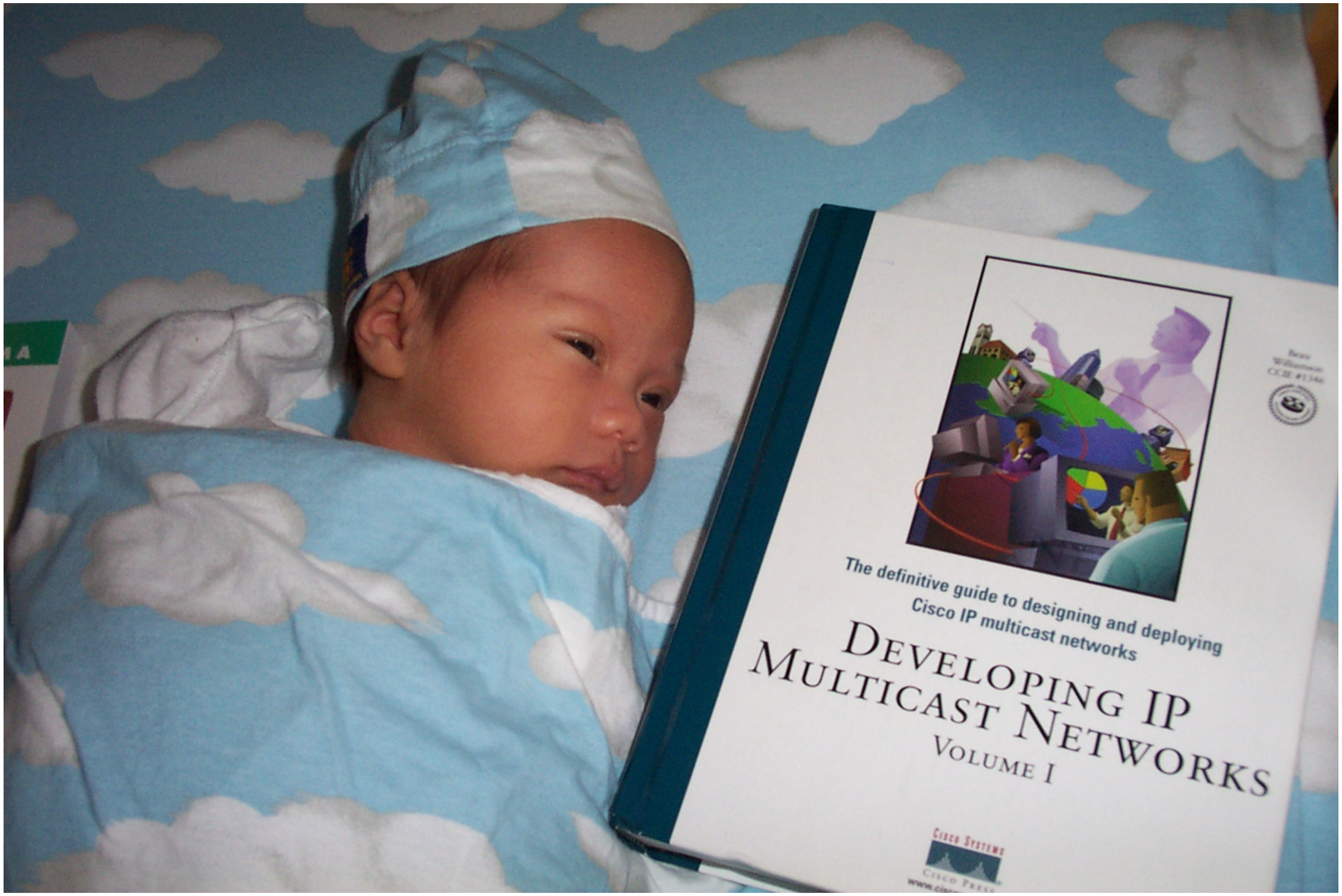
- White Papers
- Web and Mailers
- Cisco Press

RTFB

- CCO Multicast page:
  - http://www.cisco.com/go/ipmulticast
- Questions:
  - cs-ipmulticast@cisco.com
- Customer Support Mailing List:
  - tac@cisco.com

RTFB = “Read the Fine Book”

Multicast Bedtime Stories



Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Receive 20 Passport points for each session evaluation you complete.
- Complete your session evaluation online now (open a browser through our wireless network to access our portal) or visit one of the Internet stations throughout the Convention Center.

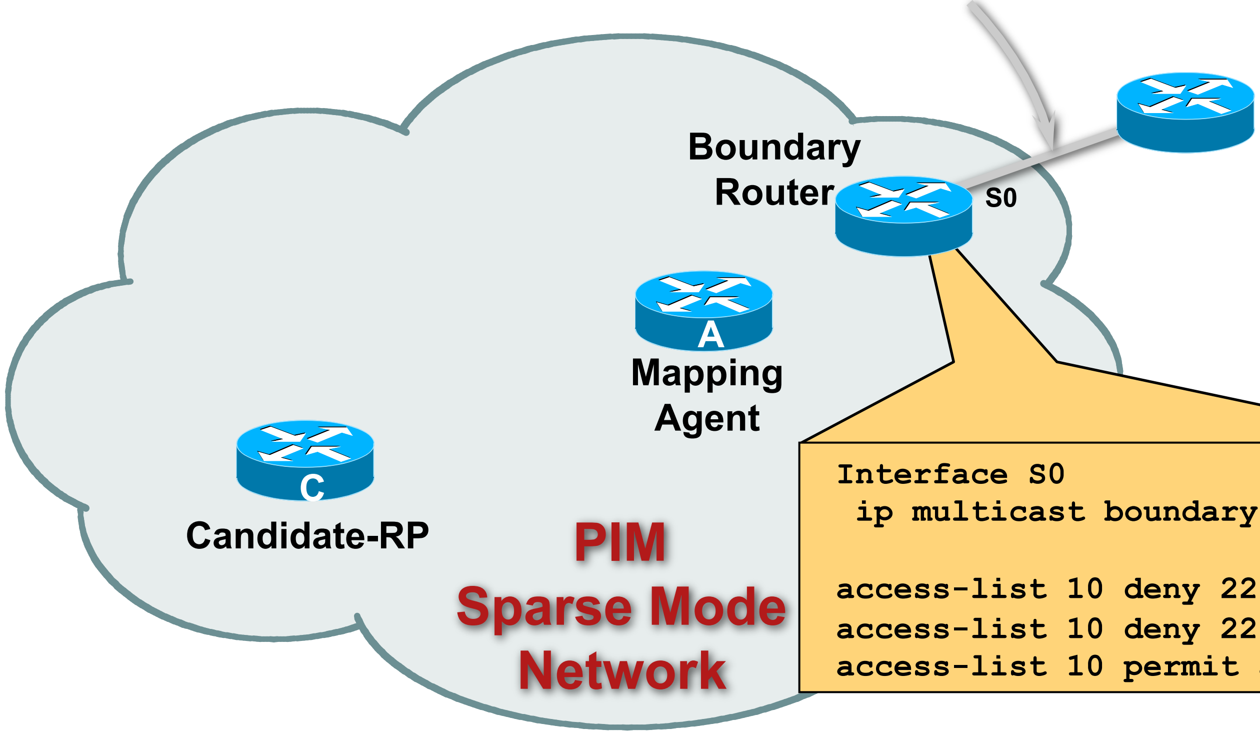
Don't forget to activate your Cisco Live Virtual account for access to all session material, communities, and on-demand and live activities throughout the year. Activate your account at the Cisco booth in the World of Solutions or visit www.ciscolive.com.

Back-Up Slides

Tuning RP Operations

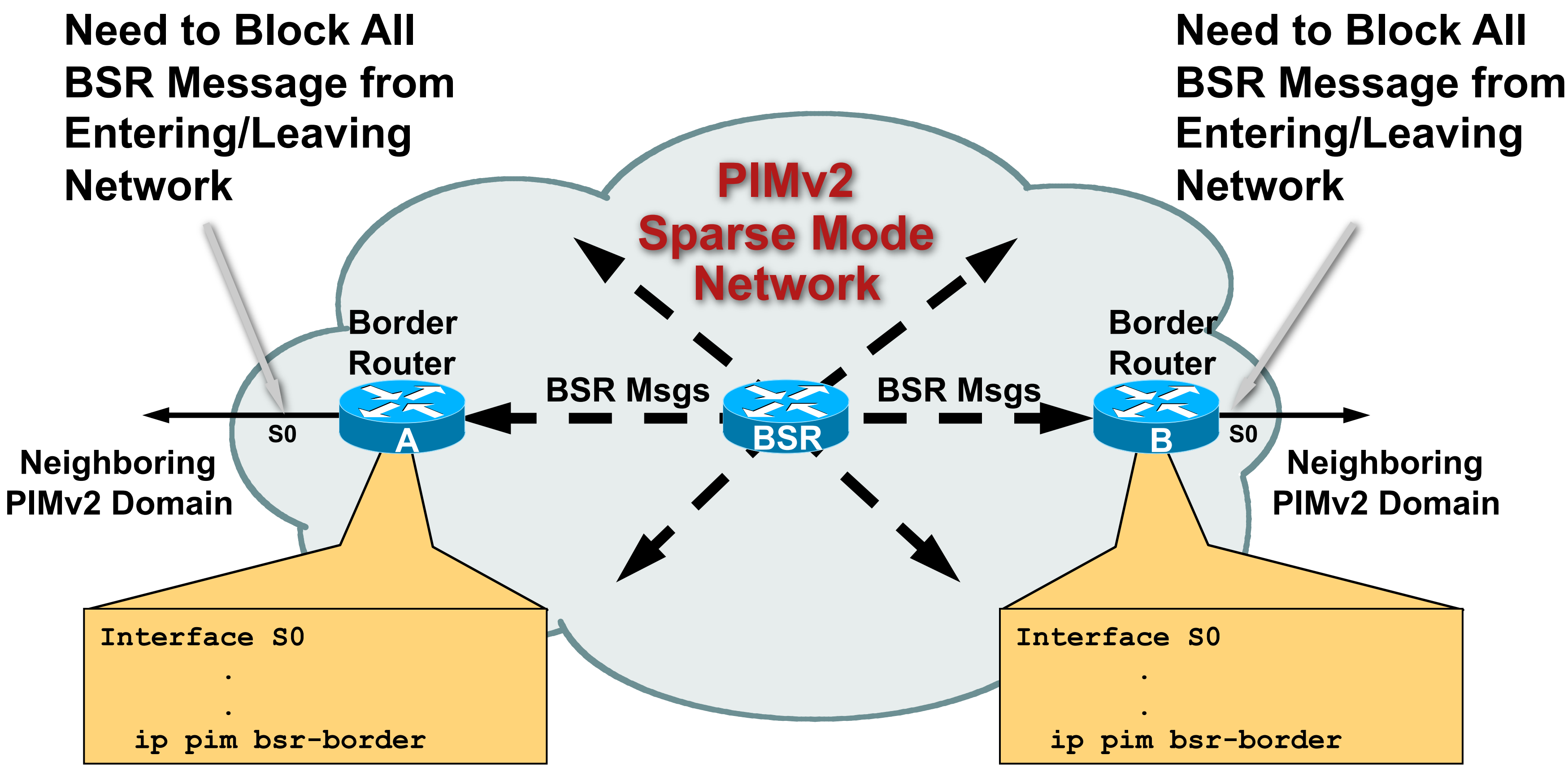
Constraining Auto-RP Messages

Need to Block Auto-RP Discovery (224.0.1.40) and Announcement (224.0.1.39) Messages from Entering/Leaving the Network




```
Interface S0
 ip multicast boundary 10

access-list 10 deny 224.0.1.39
access-list 10 deny 224.0.1.40
access-list 10 permit any
```

Constraining BSR Messages



Filtering C-RP Announcements

- Use on Mapping Agents to filter out bogus C-RPs

  Some protection from RP-Spoofing denial-of-service attacks

  Multiple commands may be configured as needed

- Global command

  ```
  ip pim rp-announce-filter rp-list <acl> [group-list <acl>]
  ```

  rp-list <acl>

  Specifies from which routers C-RP Announcements are accepted

  group-list <acl>

  Specifies which groups in the C-RP Announcement are accepted

  If not specified, defaults to deny all groups

Controlling Source Registration

- Global command

  `ip pim accept-register [list <acl>] | [route-map <map>]`

  Used on RP to filter incoming Register messages

  Filter on Source address alone (Simple ACL)

  Filter on (S, G) pair (Extended ACL)

  May use route-map to specify what to filter

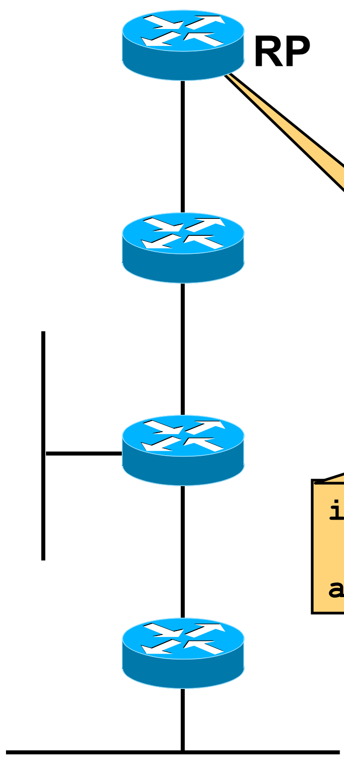
    Filter by AS-PATH if (m)BGP is in use

- Helps prevents unwanted sources from sending

  First hop router blocks traffic from reaching net

  Note: Traffic can still flow under certain situations

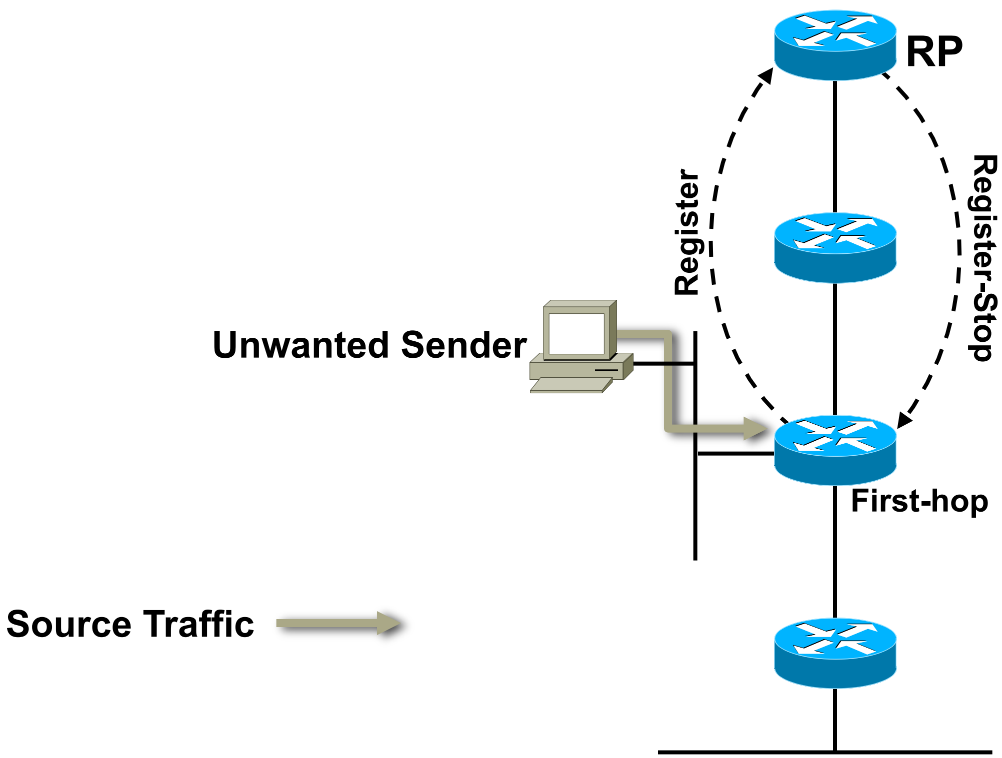
Controlling Source Registration



- RP configured to only accept Registers from specific source

```
ip pim accept-register list 10

access-list 10 permit 192.16.1.1
```

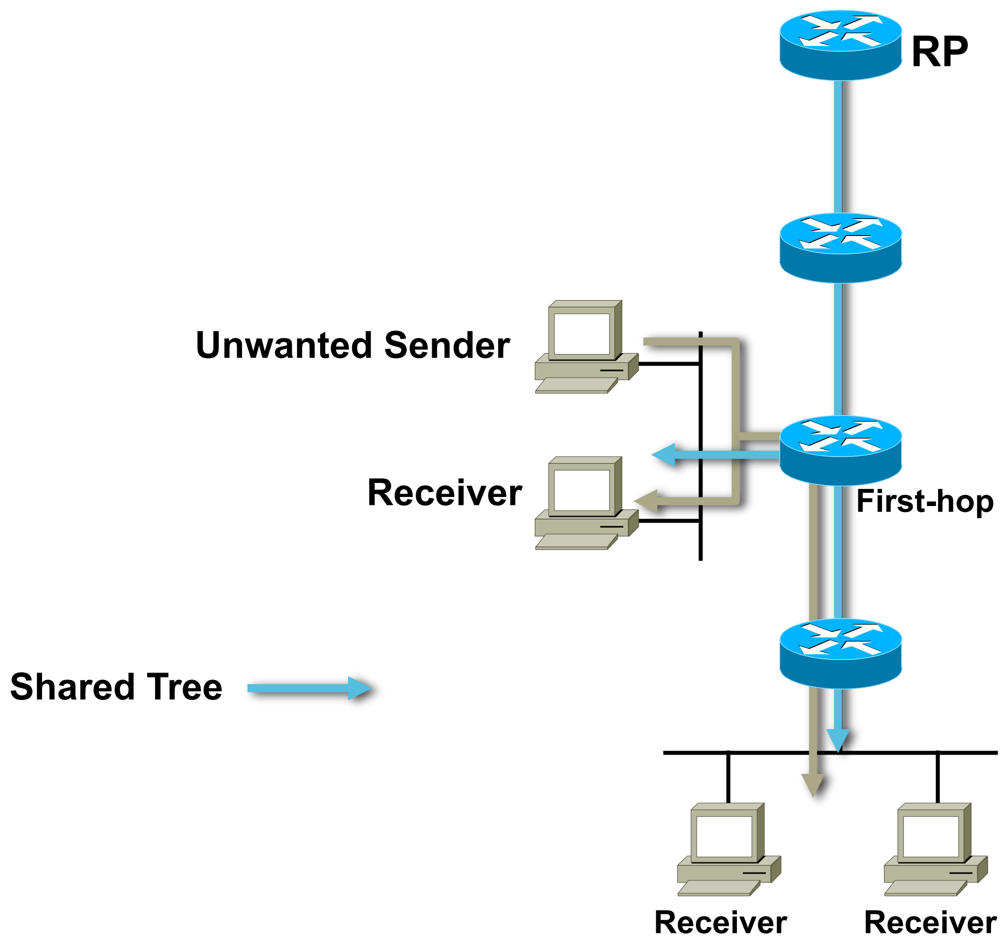
Controlling Source Registration



- Unwanted source traffic hits first-hop router
- First-hop router creates (S,G) state and sends Register
- RP rejects Register, sends back a Register-Stop

Controlling Source Registration

Weaknesses in 'accept-register' Usage



- Traffic will flow on local subnet where source resides
- Traffic will flow from first-hop router down any branches of the Shared Tree

  Results when (*,G) OIL is copied to (S,G) OIL at first-hop router

  Causes (S,G) traffic to flow down all interfaces in (*,G) OIL of first-hop router

Case Study: ACME Financials

ACME's Primary Multicast Applications

- IP/TV
- Hoot-n-Holler
- VoIP Music-on-Hold
- TIBCO Data Distribution
- Internet Multicast Access

IP/TV

- One-to-many video multicast

  Live or rebroadcast content

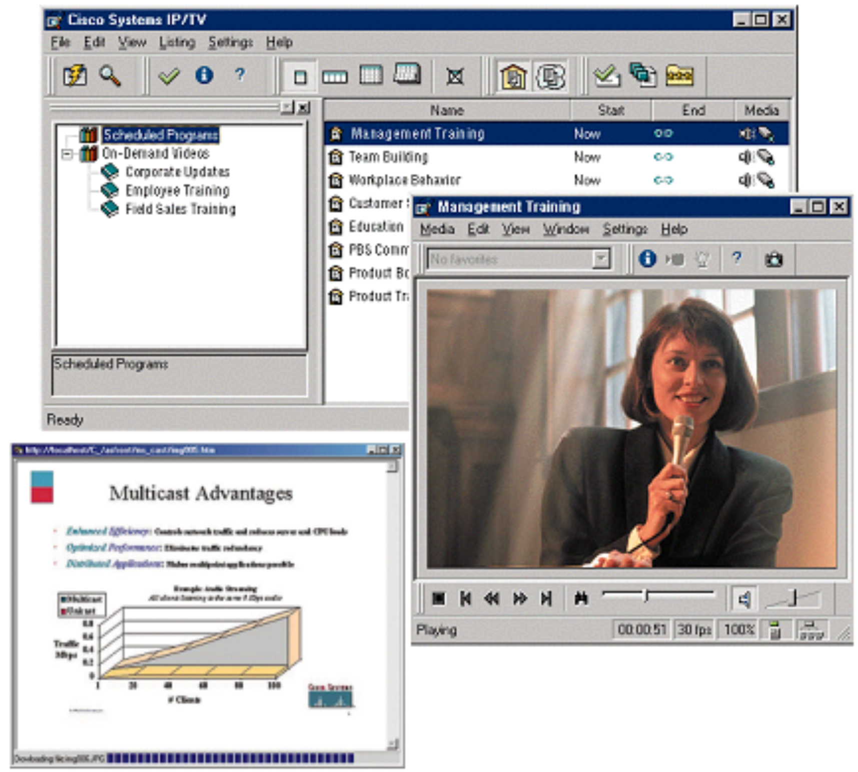
  Synchronized presentations

  Integrated
  “Question Manager”

  Supports “Source Specific Multicast” (SSM)

  Video-on-Demand (VoD)

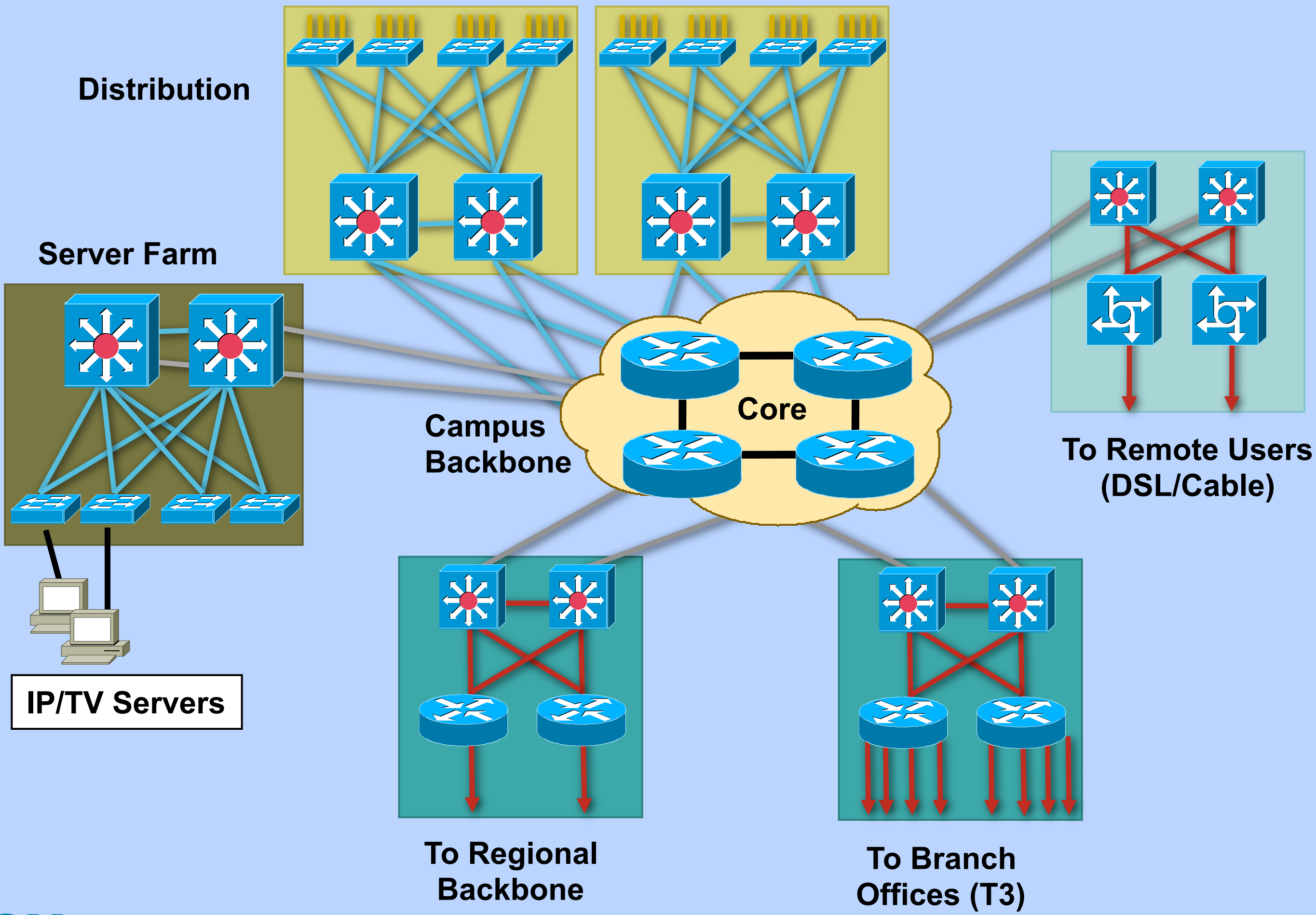
  (Unicast only)



Corporate Broadcasts (IP/TV)

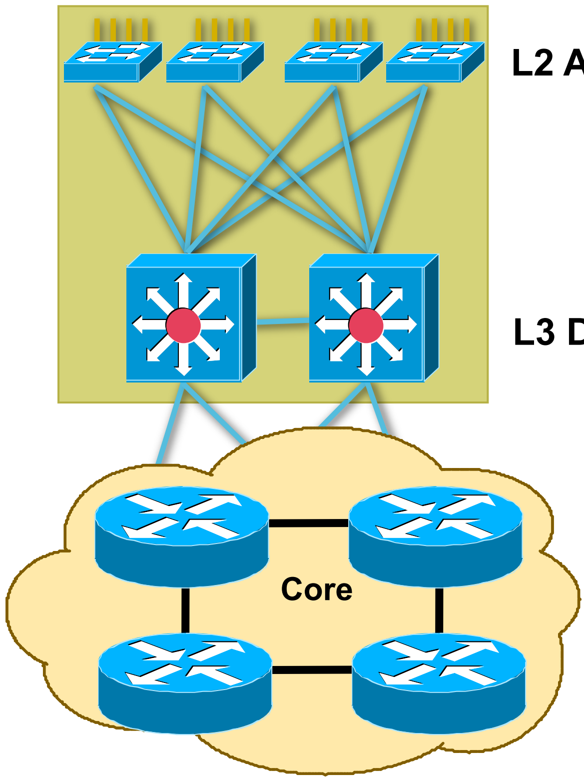
- Multicast Protocol: SSM
- IP/TV assigned to an SSM group range
  - **No** RPs, minimal configuration
  - Avoids “Capt. Midnight” problem
- Additional options:
  - Bandwidth-based group scoping

IP/TV Broadcast With SSM



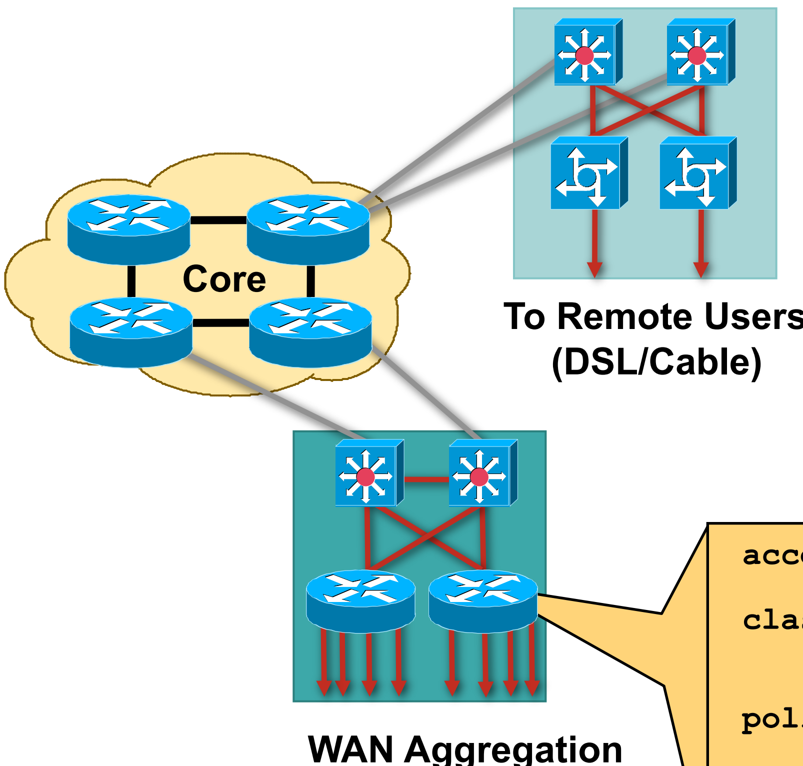
SSM (239.232.0.0/16 'Internal' SSM Address Range)

IP/TV—SSM on Campus



- IGMPv3 Snooping on Access where available on clients and switches
- 'static ssm mapping' when IGMPv3 is not available (L3 distribution)
- Use bandwidth scoping for SSM groups with different rates (239.232.QOS.x)

IP/TV—SSM Over WAN



- QoS protection necessary for remote users and WAN aggregation
- Use bandwidth scoping to deny high rate streams
- Use MQC to guarantee bandwidth to IP/TV:

```
access-list 101 permit ip any 239.232.224.0 0.0.31.255

class-map match-all iptv-qos-lowbandwidth
    match access-group 101

policy-map IPTV-over-T1
    class iptv-qos-lowbandwidth
       bandwidth 512
    class default
       fair-queue
```

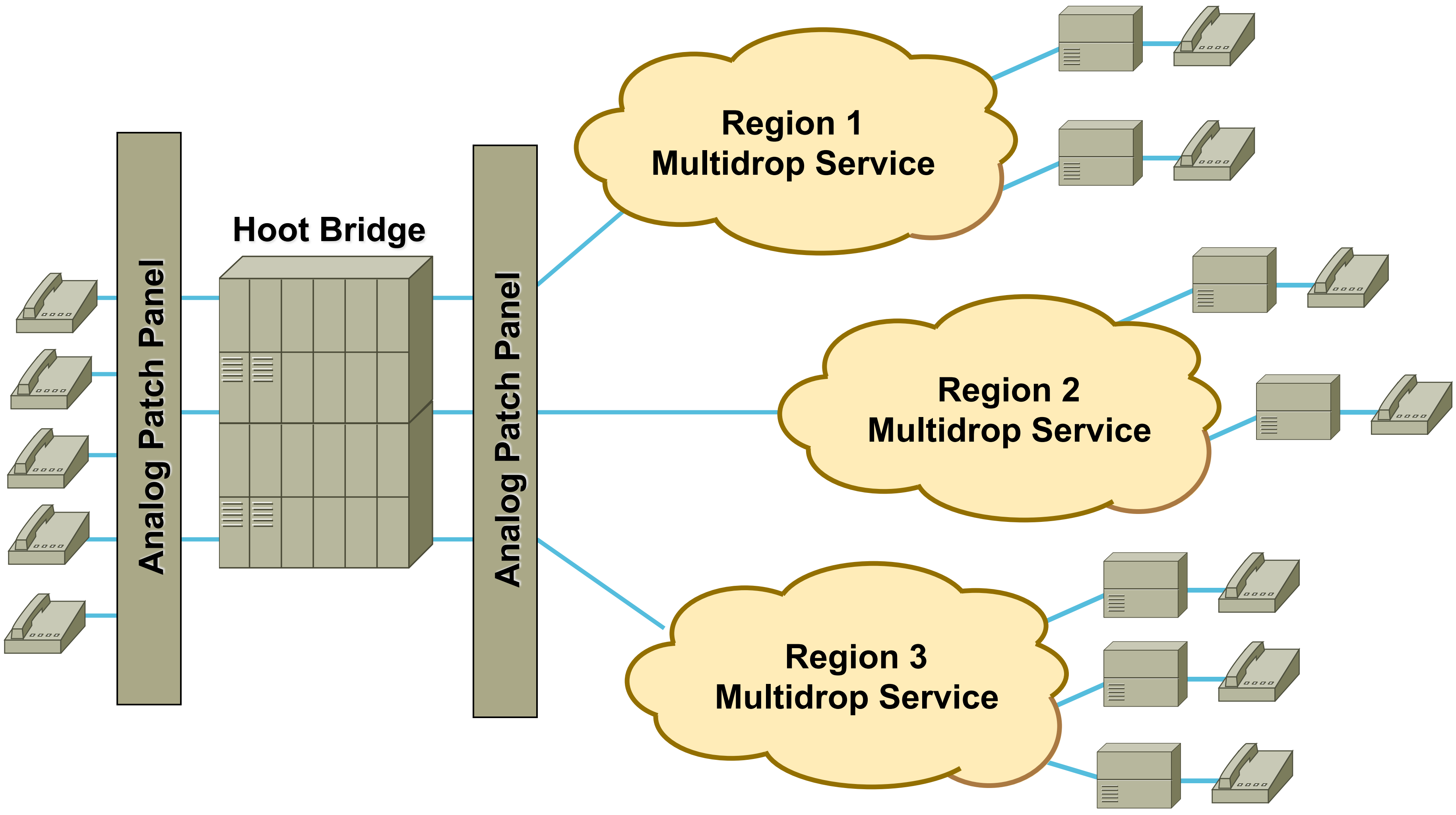

Hoot-n-Holler

Hoot-n-Holler Turret

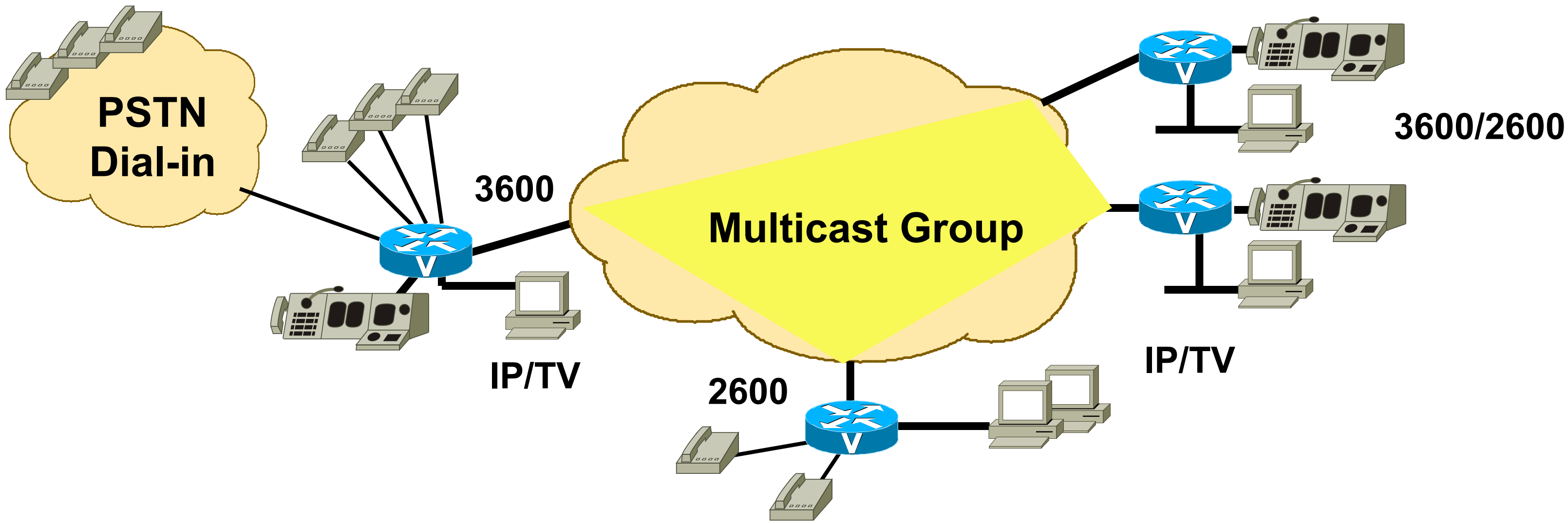
Hoot-n-Holler

- Broadcast audio network
- Typically point to multipoint
- Uses specialized analog 4-wire phones (hoot phones) and digital turrets
- Brokerages, utilities, media companies, mass transit, publishing, etc.

Traditional Hoot-n-Holler Network Design



Hoot-n-Holler Over IP Multicast



- Leverages VoIP, IP Multicast (IPmc) and QoS
- Bridging and audio mixing occurs in router voice DSPs
- Dynamic bandwidth sharing and cost savings
- Existing analog end systems and procedures retained

Hoot-n-Holler Over IP Multicast—
Considerations

- Multicast Protocol: Bidir PIM

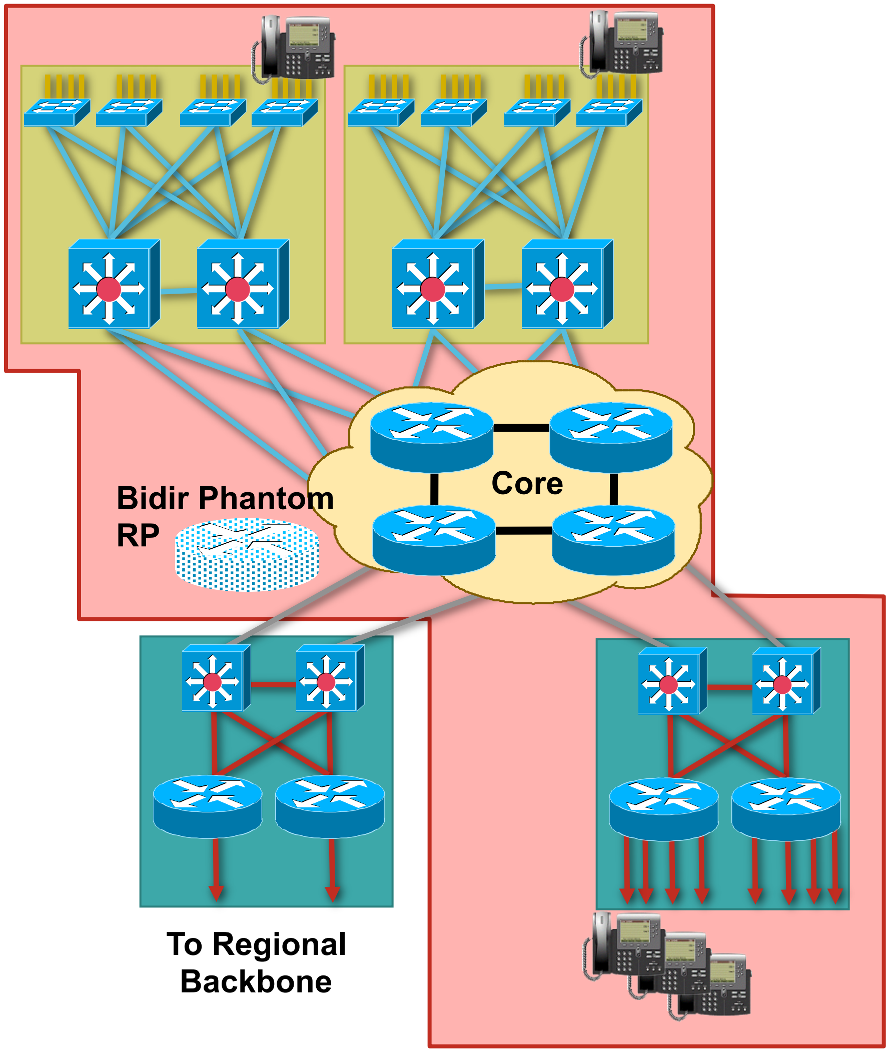
  Scales well for many-to-many applications

- Additional options:

  LLQ (Low latency queue) for Hoot-n-Holler traffic (voice traffic)

  CRTP Header Compression for low-speed links

Hoot and Holler on Campus and Over WAN

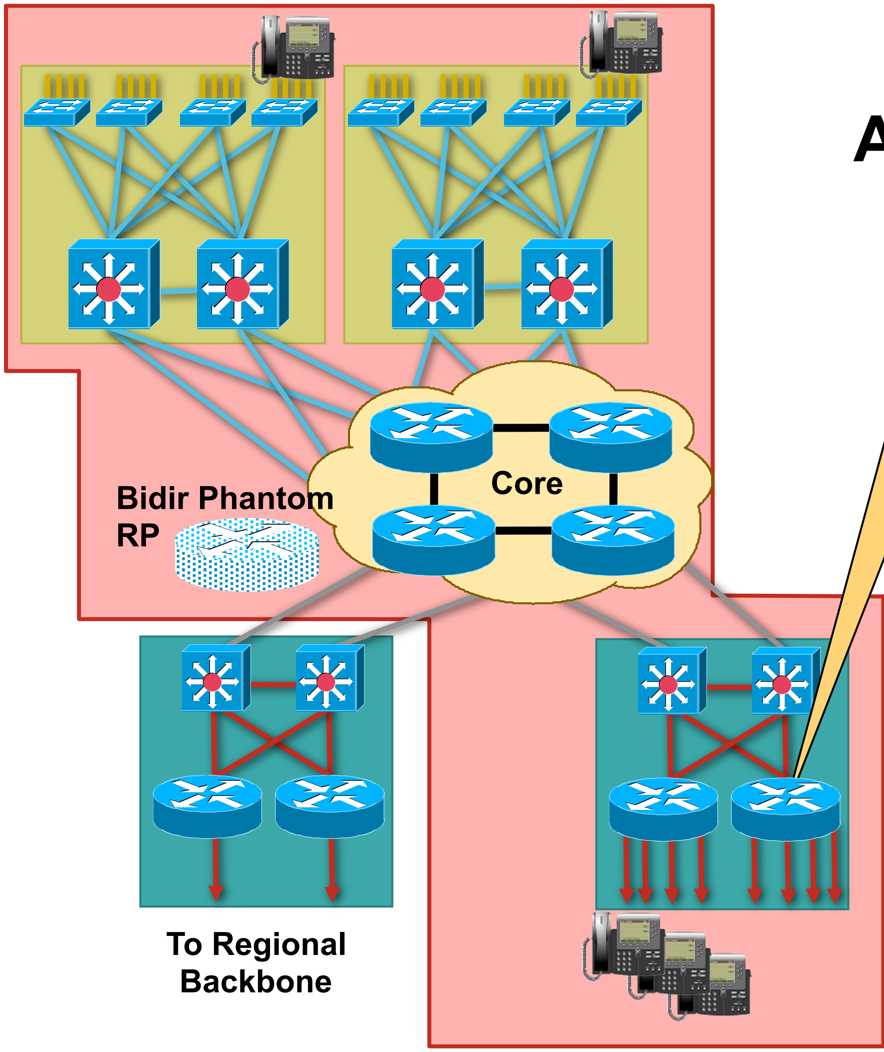


- Choose groups from Campus range for H&H

  Example 239.193.1.x

- Bidir-enabled on all (red zone) routers
- H&H is voice traffic, so treat it accordingly with Low Latency Queues (LLQ)

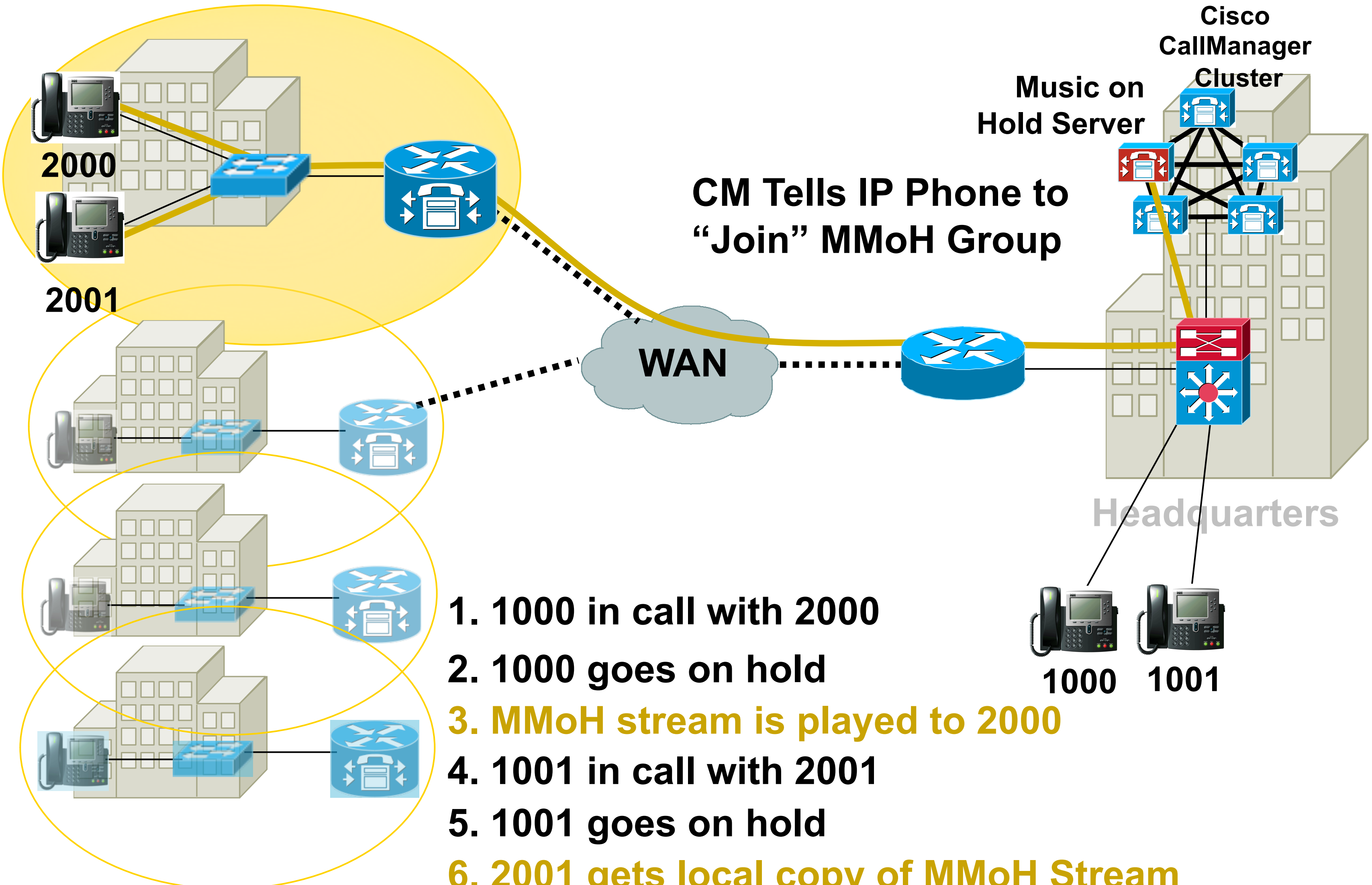
Hoot and Holler on Campus and Over WAN



Adding QoS for Hoot and Holler

```
access-list 101 permit ip any 239.232.224.0 0.0.31.255
access-list 102 permit ip any 239.193.1.0 0.0.0.255

class-map match-all iptv-qos-lowbandwidth
     match access-group 101

class-map match-all hootie
     match access-group 102

policy-map Mcast-over-T1
     class hootie
          priority 495
     class iptv-qos-lowbandwidth
          bandwidth 512
     class default
          fair-queue
```

Multicast Music-on-Hold (MMoH)



1. 1000 in call with 2000
2. 1000 goes on hold
3. MMoH stream is played to 2000
4. 1001 in call with 2001
5. 1001 goes on hold
6. 2001 gets local copy of MMoH Stream

Multicast Music-on-Hold (MMoH)

- Increment on IP addresses as opposed to port numbers
- Modify the “Max Hops” (TTL) default value of 2
  - Adjust according to the network topology and hop count to the receiver
- Use administratively scoped addressing for the MMoH address range
  - **Note: CSCdv01308—Default Multicast MoH IP address should not be 239.0.0.0—Fixed in 3.1(3)**
- Use G.729 for low-bandwidth sites—**warning—low quality**
- Know how many audio sources have been configured for IP Multicast

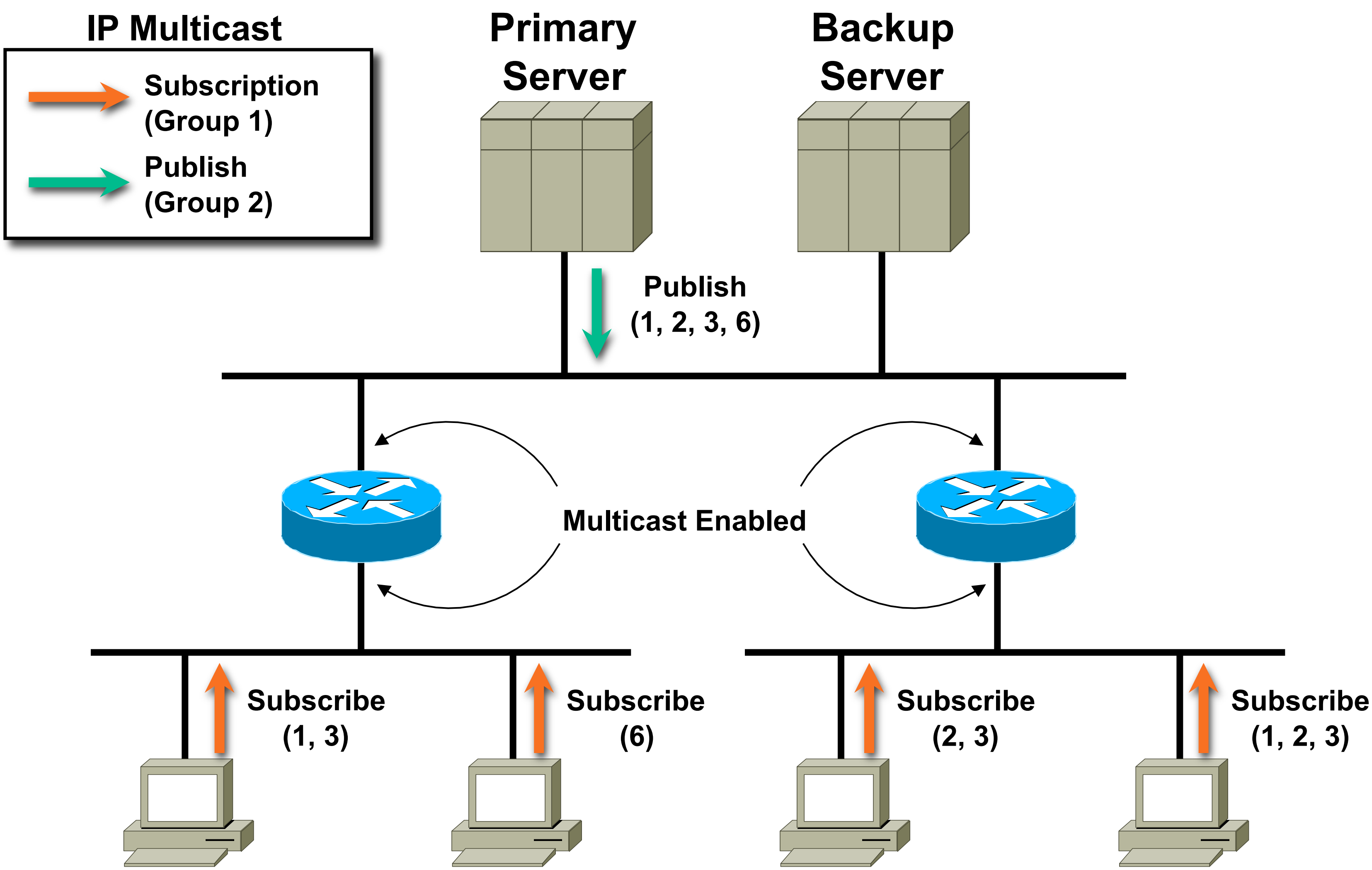
CODEC	Multicast Address
G.711ulaw	**239.192.240.1**
G.711alaw	**239.192.240.2**
G.729	**239.192.240.3**
Wideband	**239.192.240.4**

51 Possible Audio Sources
X
4 Multicast Addresses per Source
=
204 Multicast Addresses Consumed

TIBCO Data Distribution

- Popular with financial institutions
  - Used to send stock market data to traders
- Uses subscribe/publish model
- Clients multicast subscriptions messages
  - Specifying data flow(s) they wish to receive
- Servers receive subscriptions
  - Build list of all requested data flows
  - Primary server multicast requested flows
  - Backup server takes over if primary fails

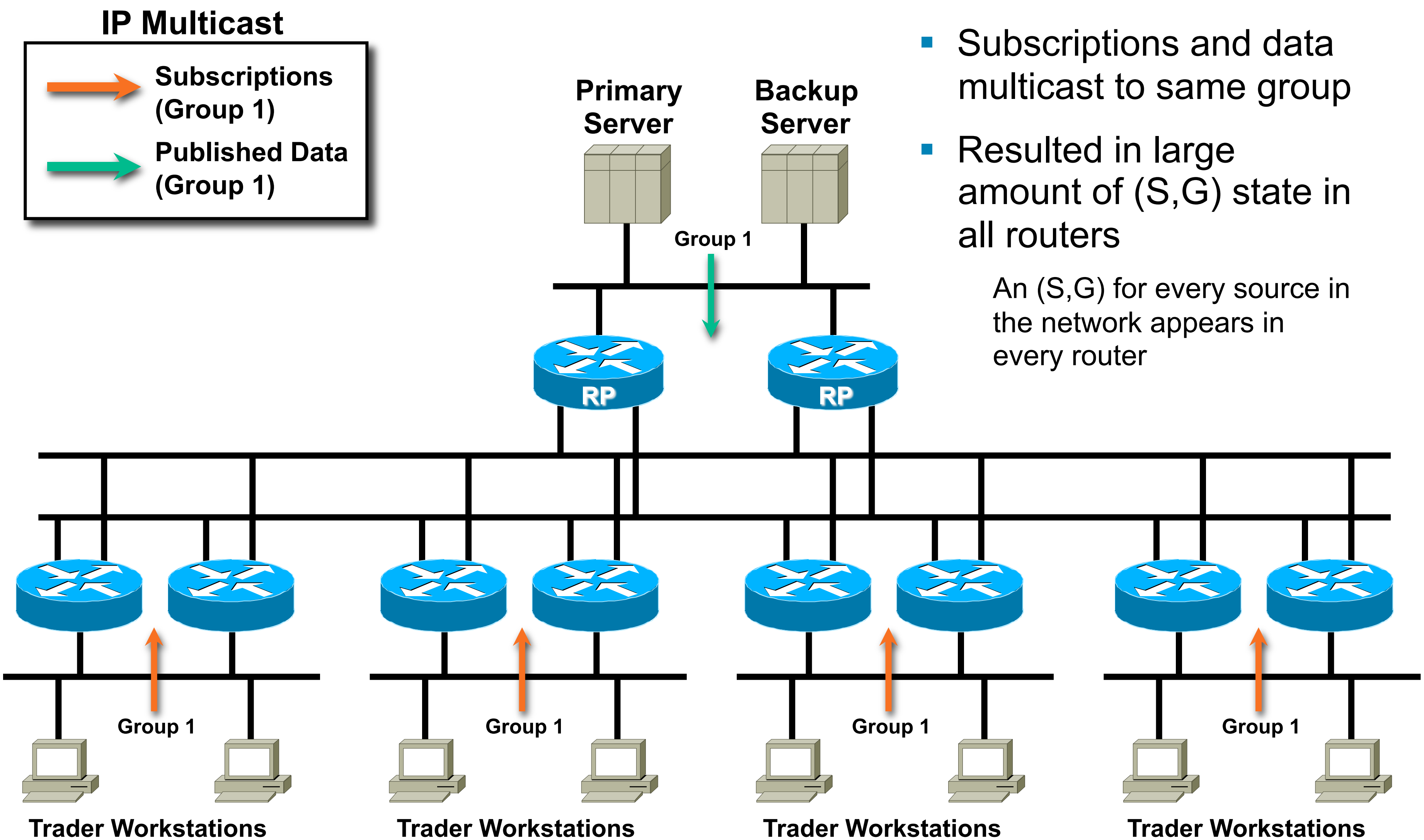
TIBCO Data Distribution



TIBCO Trading Floor Network
(ACME's Initial Deployment)

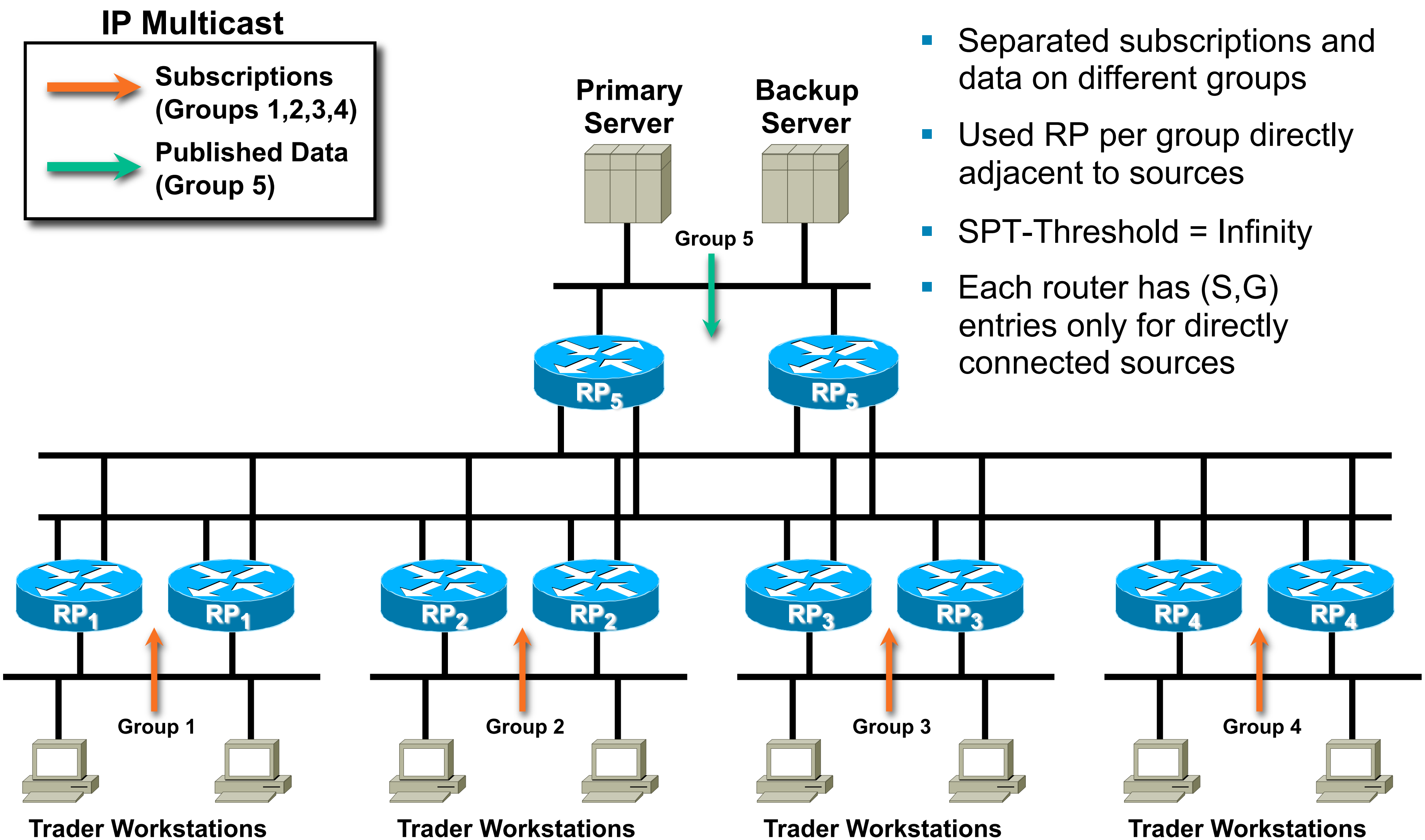
- Subscriptions and data multicast to same group
- Resulted in large amount of (S,G) state in all routers

  An (S,G) for every source in the network appears in every router



TIBCO Trading Floor Network

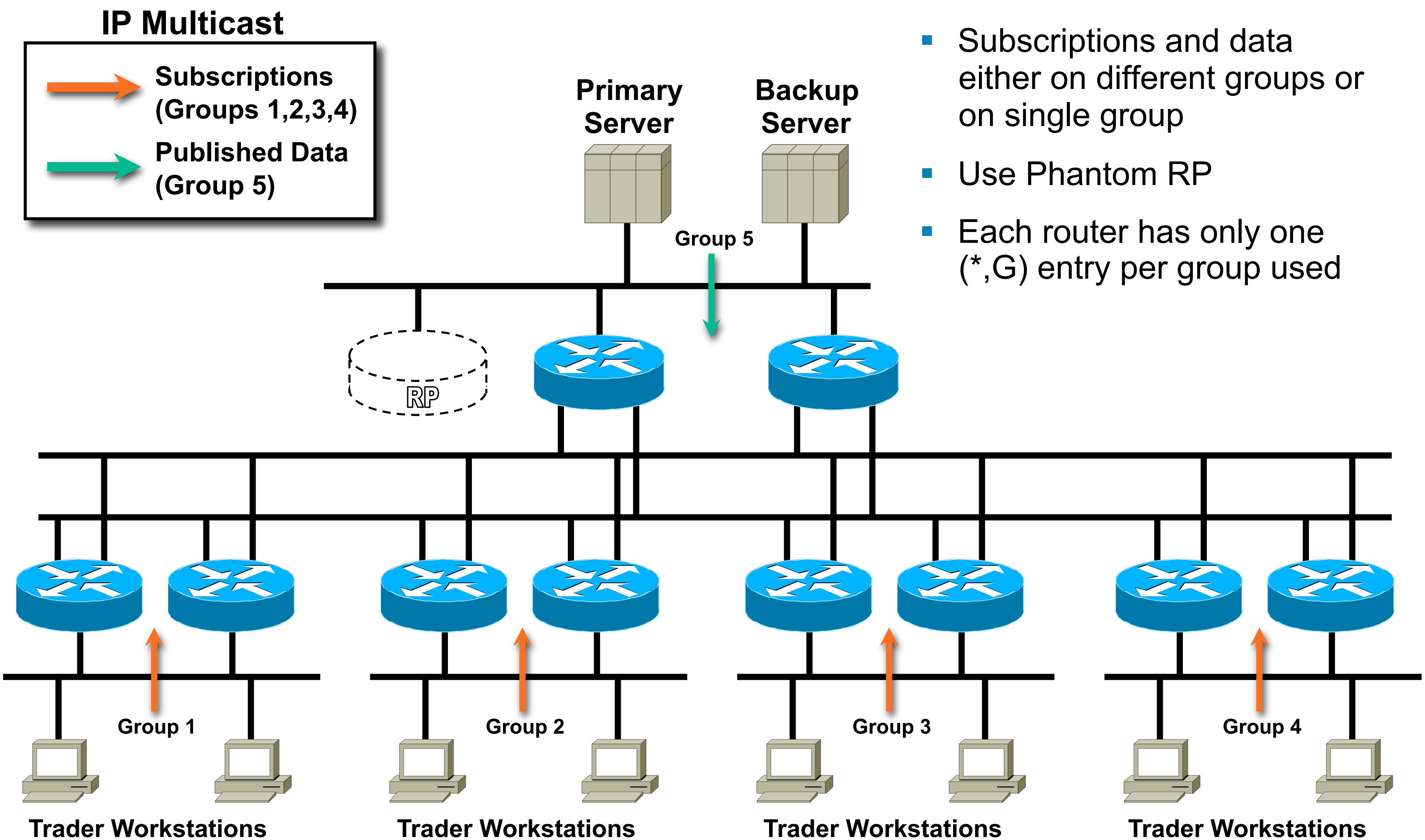
(Minimizing (S,G) State)



- Separated subscriptions and data on different groups
- Used RP per group directly adjacent to sources
- SPT-Threshold = Infinity
- Each router has (S,G) entries only for directly connected sources

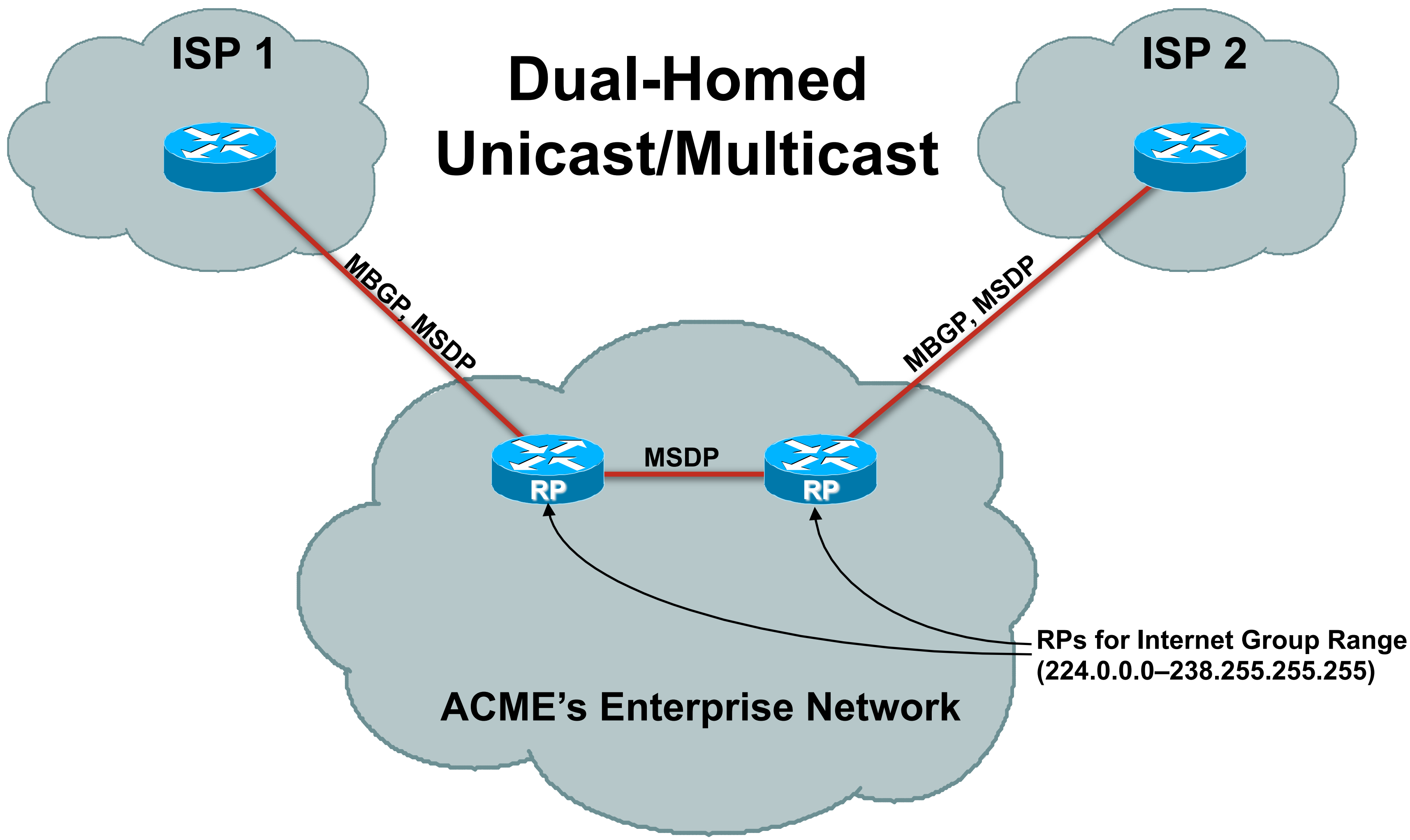
TIBCO Trading Floor Network

Optimum Solution—Bidir PIM

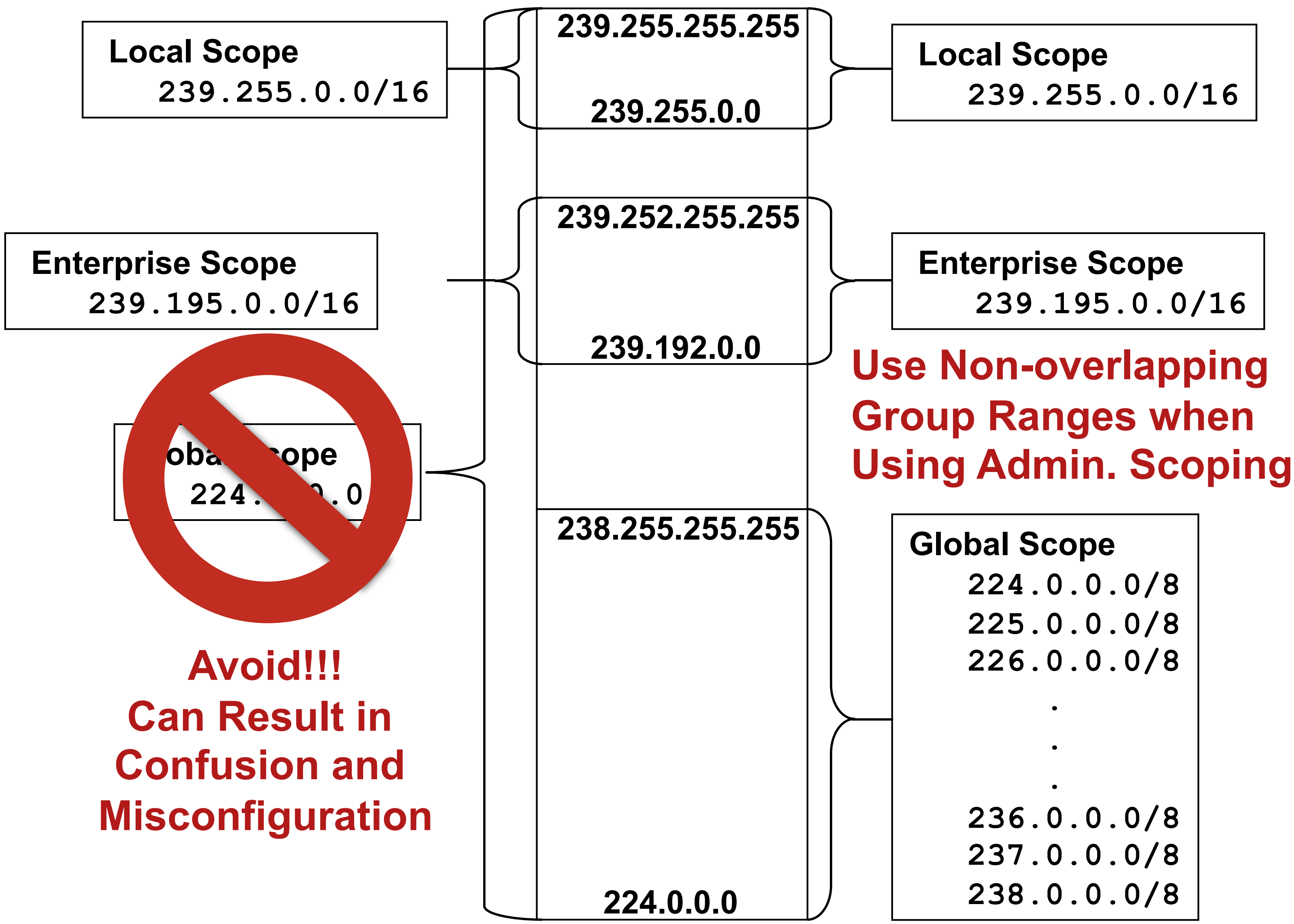


- Subscriptions and data either on different groups or on single group
- Use Phantom RP
- Each router has only one (*,G) entry per group used

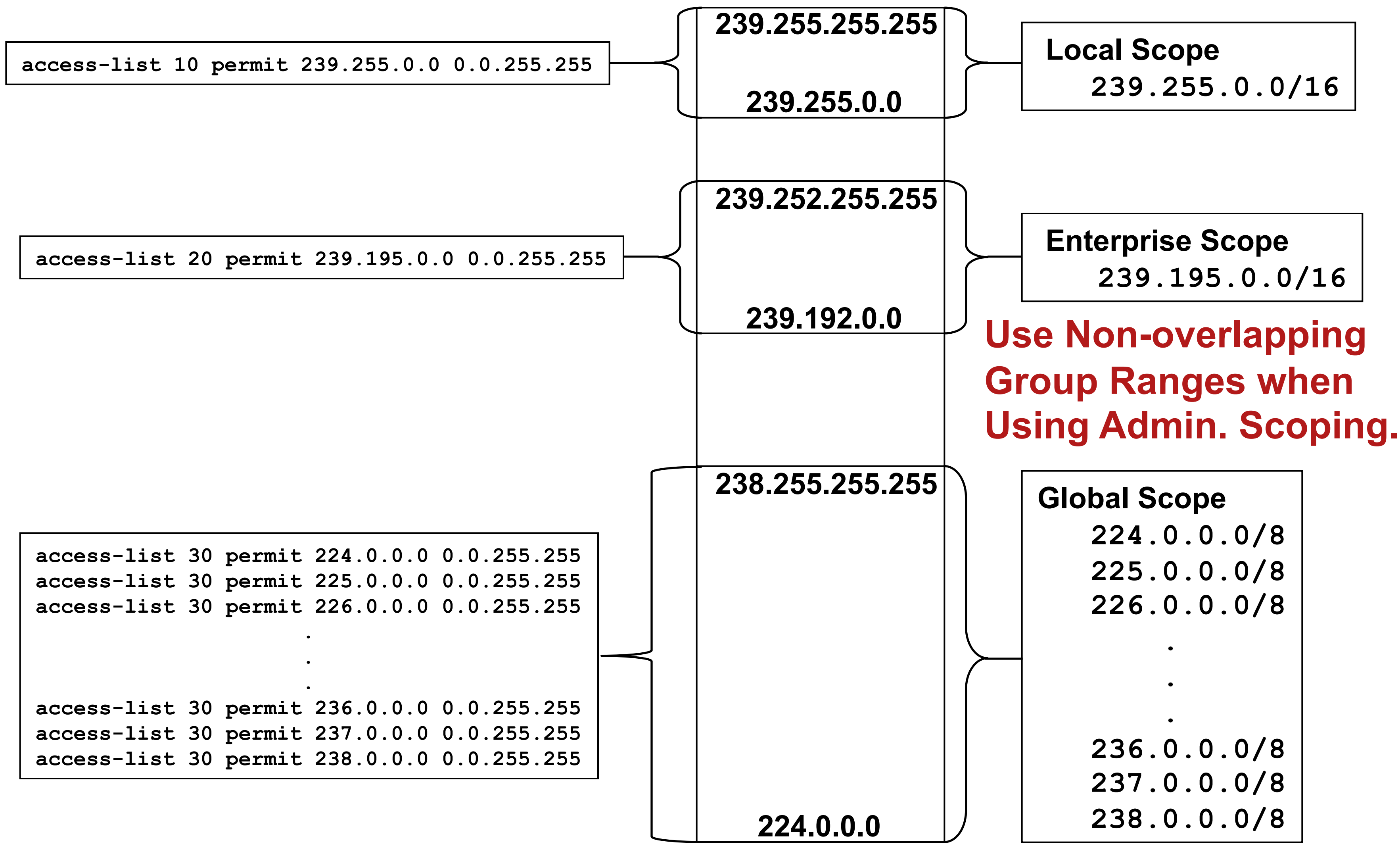
Internet Multicast Access



Avoid Overlapping Group Ranges



Avoid Overlapping Group Ranges



Avoiding Overlapping Group Ranges

- Avoiding Overlapping Group Ranges
  - Can't use "deny" clause in C-RP ACLs
    - Implies "Dense-mode Override"

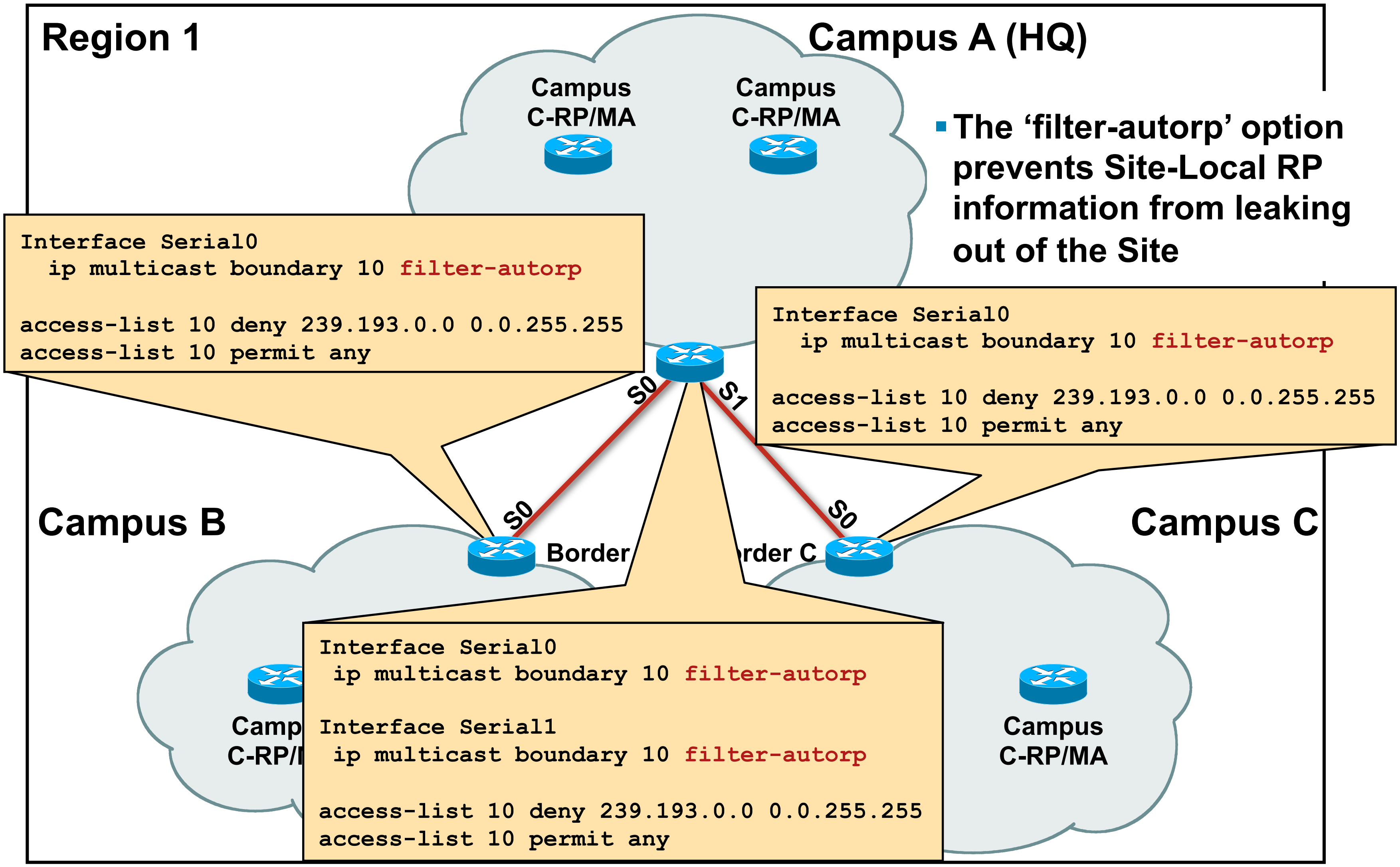
```
ip pim send-rp-announce loopback0 scope 16 group-list 10
access-list 10 deny 239.0.0.0 0.255.255.255
access-list 10 permit 224.0.0.0 15.255.255.255
```

  - Must only use "permit" clauses

```
ip pim send-rp-announce loopback0 scope 16 group-list 10
access-list 10 permit 224.0.0.0 0.255.255.255
access-list 10 permit 225.0.0.0 0.255.255.255
                    .
                    .
                    .
access-list 10 permit 238.0.0.0 0.255.255.255
```

Deploying Administratively-Scoped Zones

Auto-RP Example With 'filter-autorp' Boundaries



- The 'filter-autorp' option prevents Site-Local RP information from leaking out of the Site

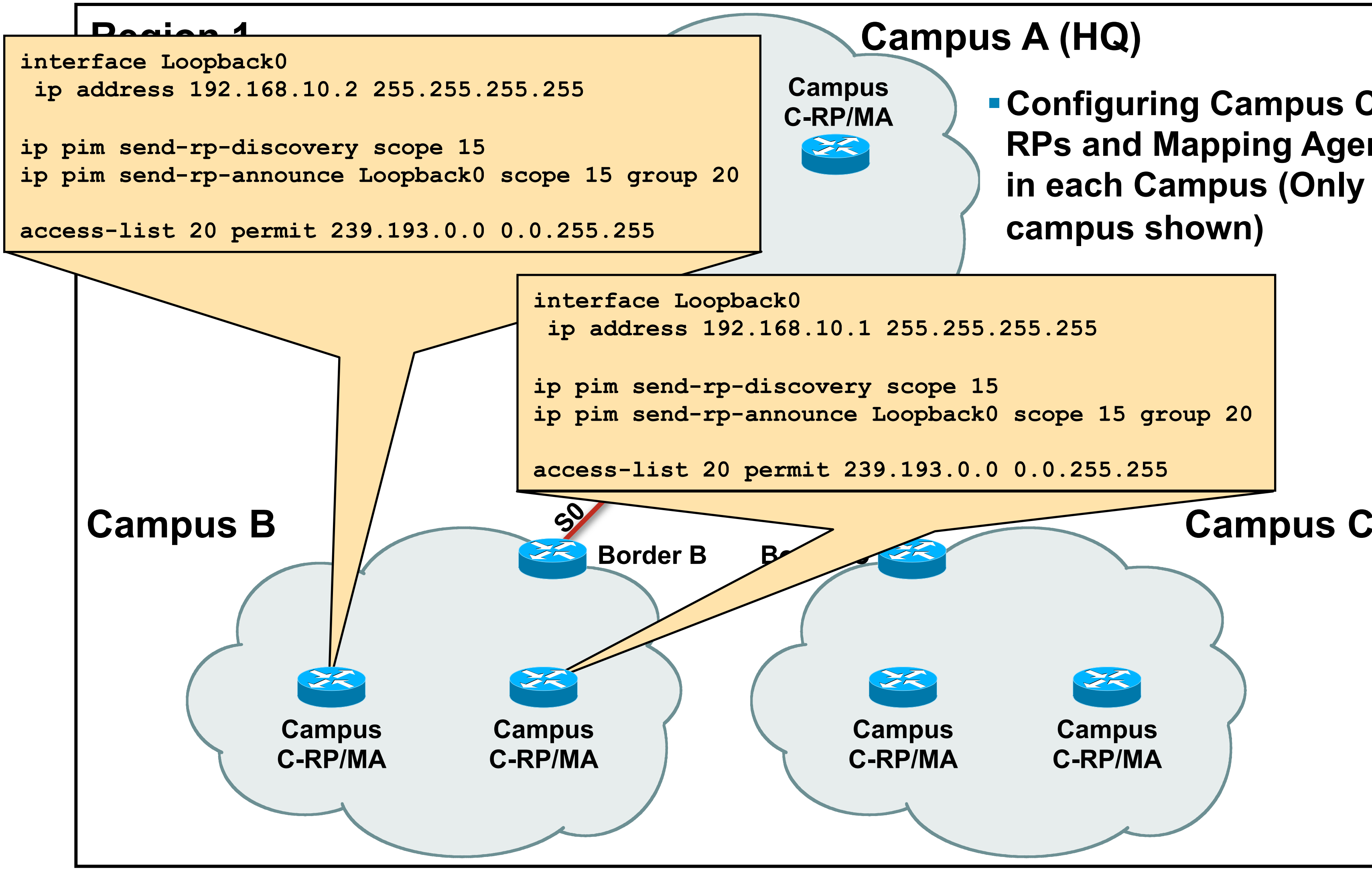
Deploying Administratively-Scoped Zones

Auto-RP Example With ‘filter-autorp’ Boundaries

- Configuring Campus C-RPs and Mapping Agents in each Campus (Only one campus shown)

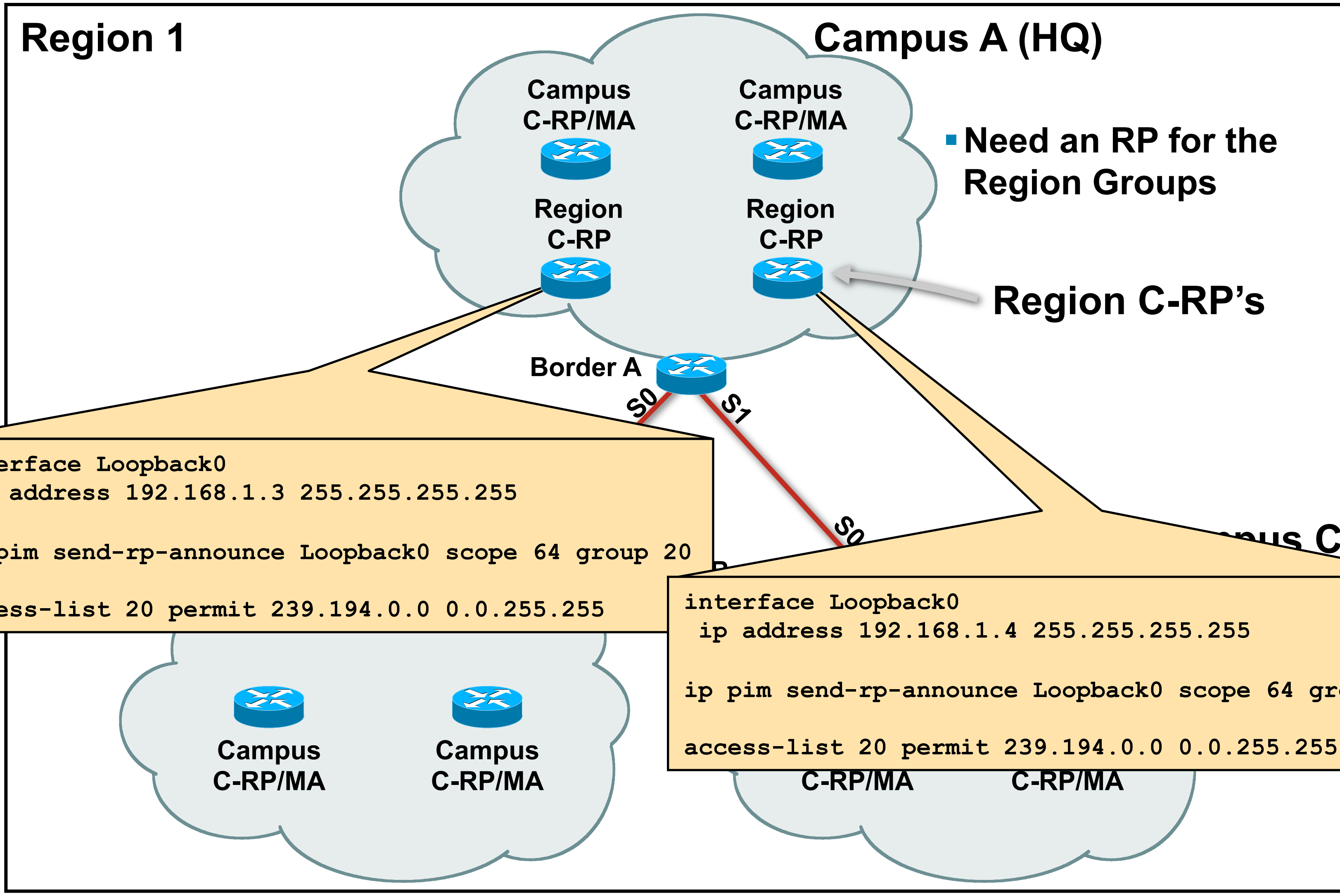
```
interface Loopback0
 ip address 192.168.10.2 255.255.255.255

ip pim send-rp-discovery scope 15
ip pim send-rp-announce Loopback0 scope 15 group 20

access-list 20 permit 239.193.0.0 0.0.255.255
```

```
interface Loopback0
 ip address 192.168.10.1 255.255.255.255

ip pim send-rp-discovery scope 15
ip pim send-rp-announce Loopback0 scope 15 group 20

access-list 20 permit 239.193.0.0 0.0.255.255
```



Deploying Administratively-Scoped Zones

Auto-RP Example With 'filter-autorp' Boundaries



- Need an RP for the Region Groups

```
interface Loopback0
 ip address 192.168.1.3 255.255.255.255

ip pim send-rp-announce Loopback0 scope 64 group 20

access-list 20 permit 239.194.0.0 0.0.255.255
```

```
interface Loopback0
 ip address 192.168.1.4 255.255.255.255

ip pim send-rp-announce Loopback0 scope 64 group 20

access-list 20 permit 239.194.0.0 0.0.255.255
```

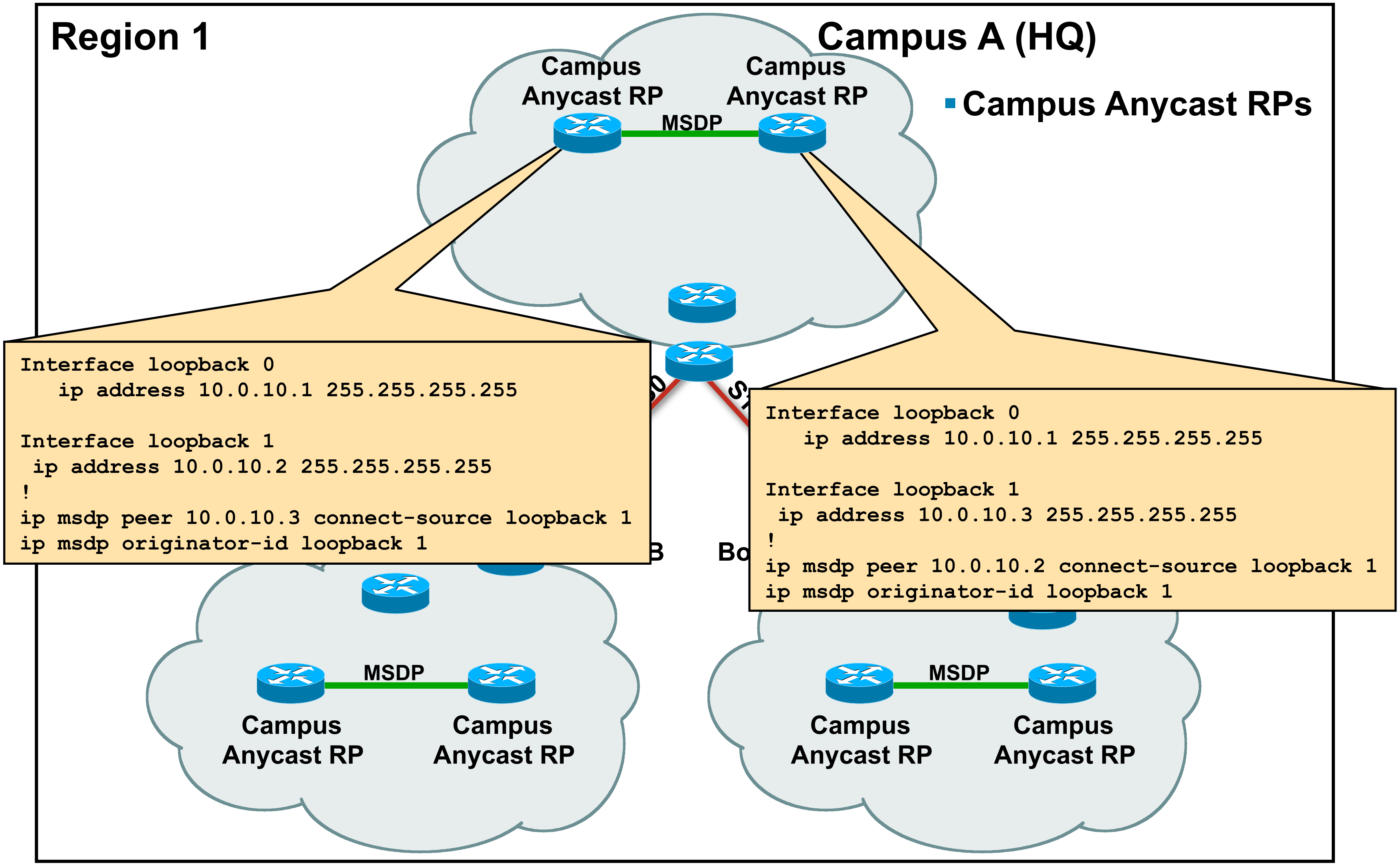
Administratively-Scoped Zones

Anycast-RP

- Admin. Scoping using Anycast RPs
  - Concept:
    - One set of Anycast RPs per physical zone
    - MSDP peer only between zone RPs
  - Advantages:
    - No filter-autorp needed at scope boundaries
  - Disadvantages:
    - Anycast RP address selection
      - Each physical zone must use its own unique Anycast RP address
    - Different static RP addresses within each zone

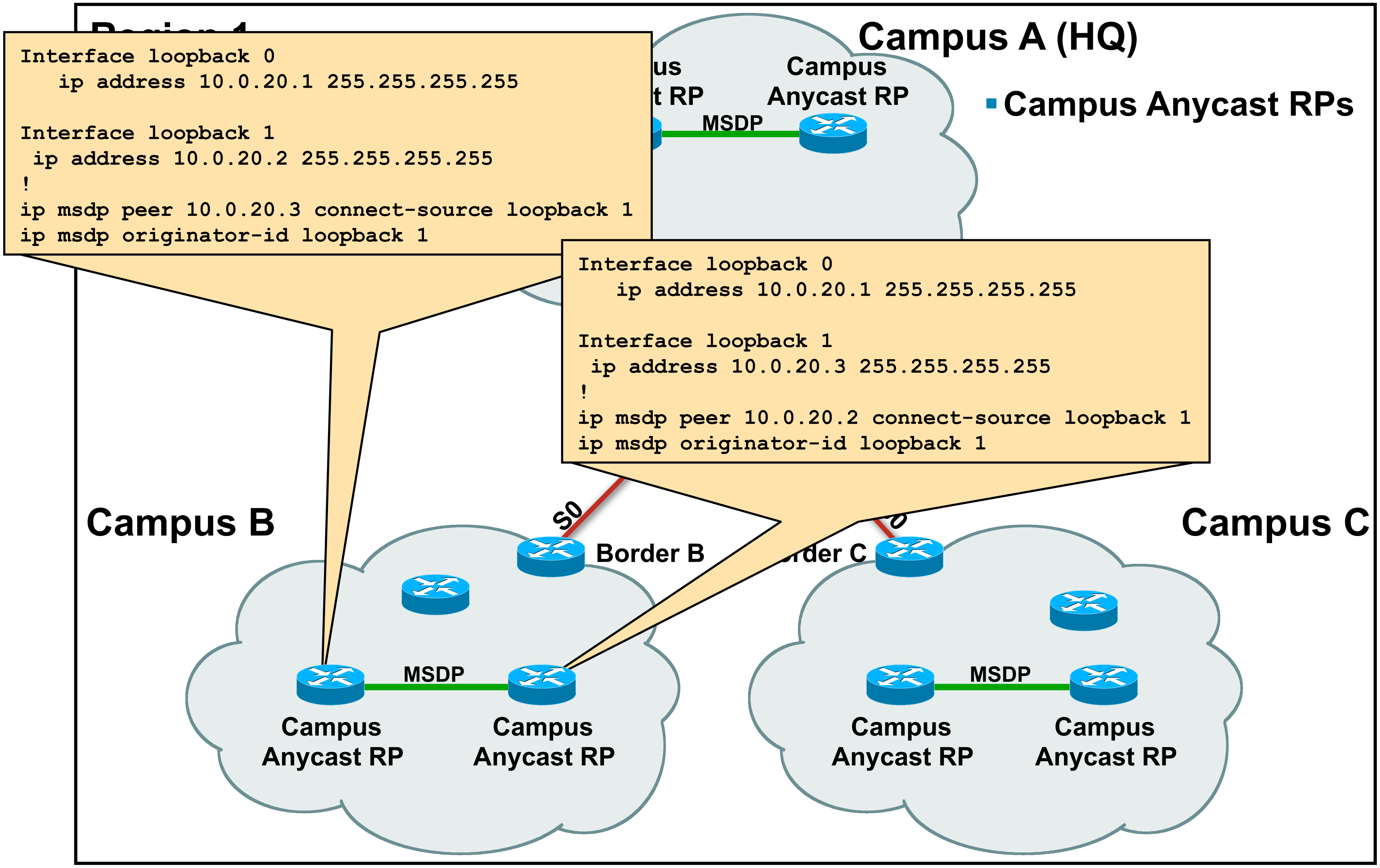
Administratively-Scoped Zones

Anycast RP Example



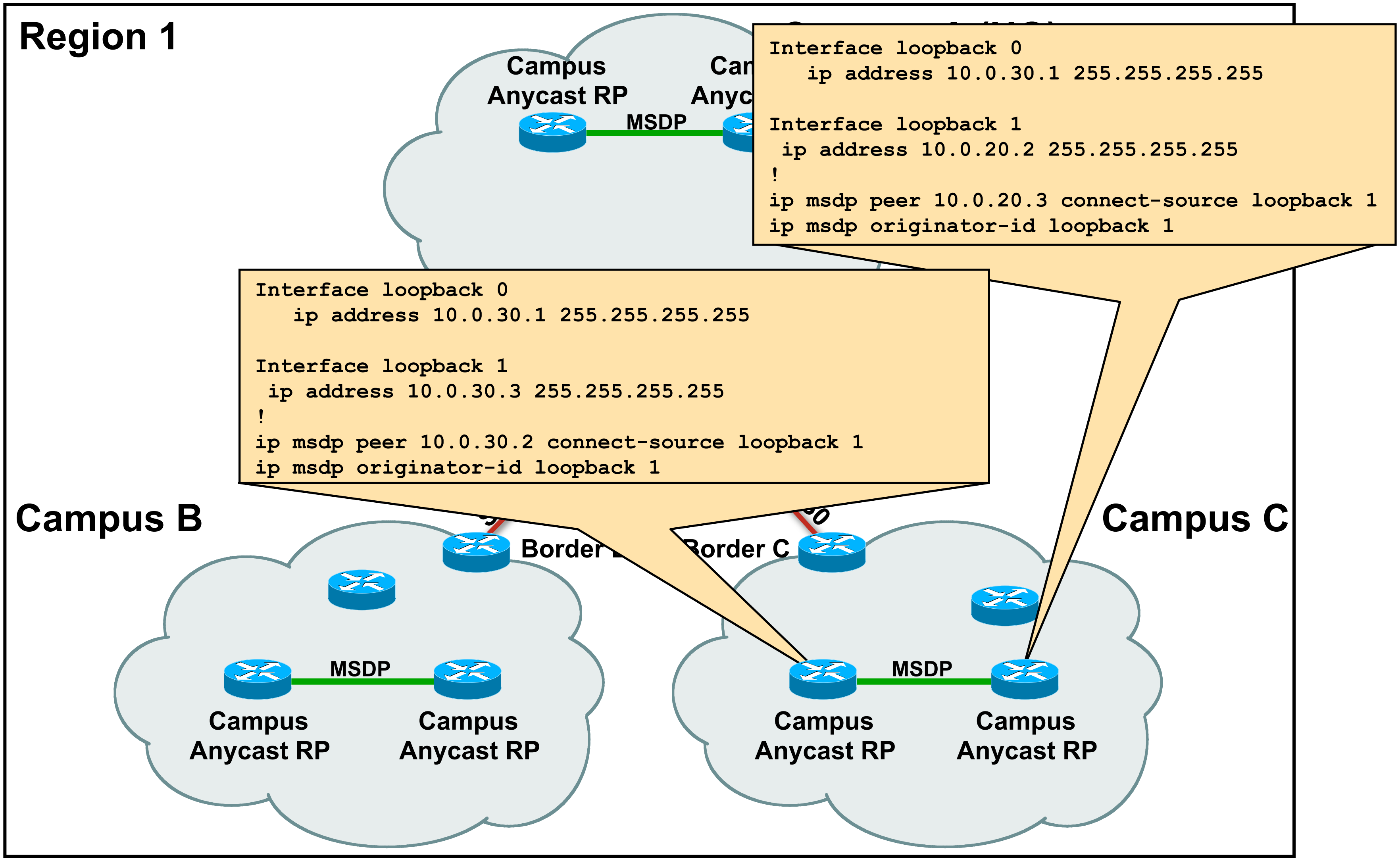
Administratively-Scoped Zones

Anycast RP Example



Administratively-Scoped Zones

Anycast RP Example

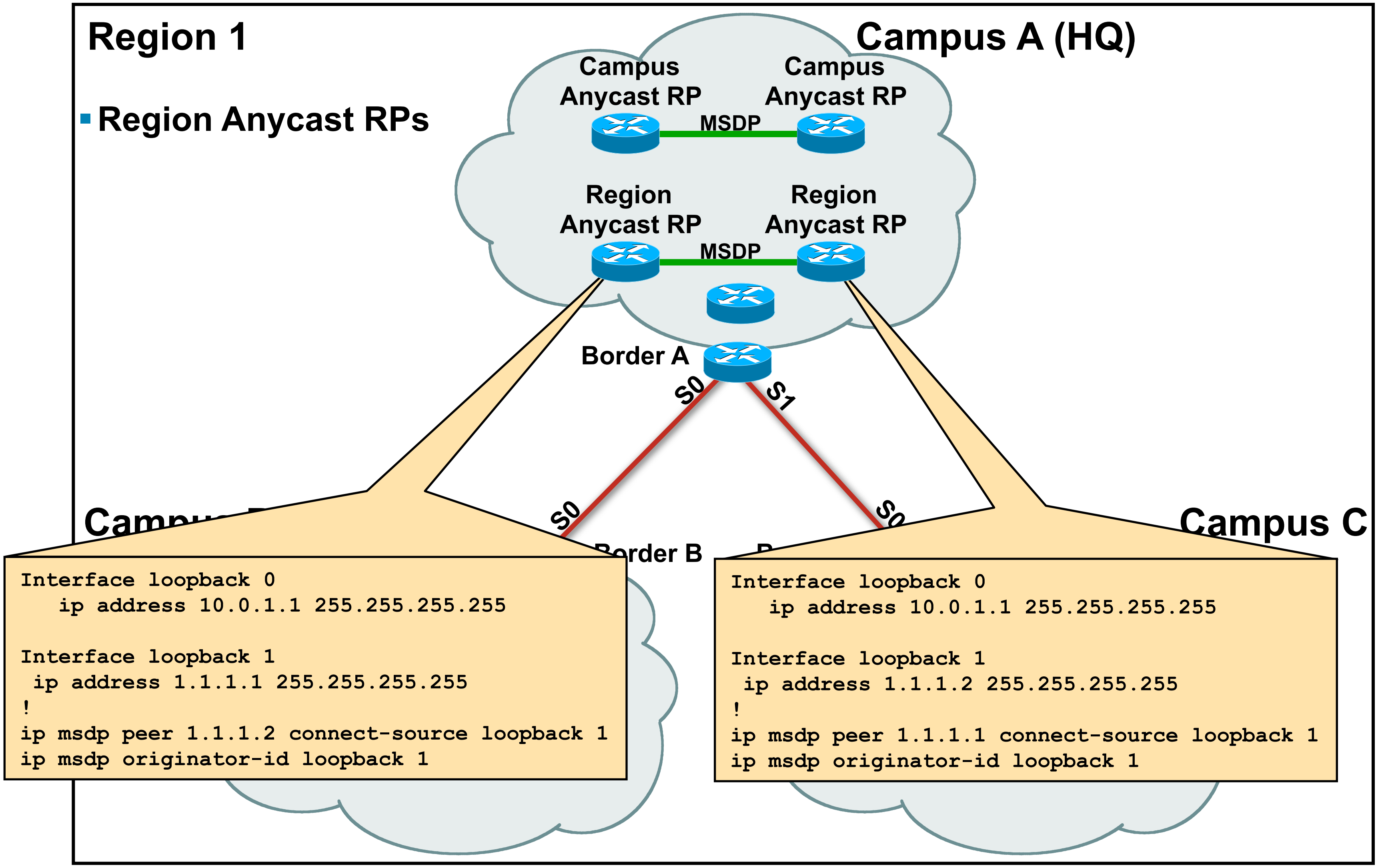



```
Interface loopback 0
   ip address 10.0.30.1 255.255.255.255

Interface loopback 1
 ip address 10.0.20.2 255.255.255.255
!
ip msdp peer 10.0.20.3 connect-source loopback 1
ip msdp originator-id loopback 1
```

```
Interface loopback 0
   ip address 10.0.30.1 255.255.255.255

Interface loopback 1
 ip address 10.0.30.3 255.255.255.255
!
ip msdp peer 10.0.30.2 connect-source loopback 1
ip msdp originator-id loopback 1
```

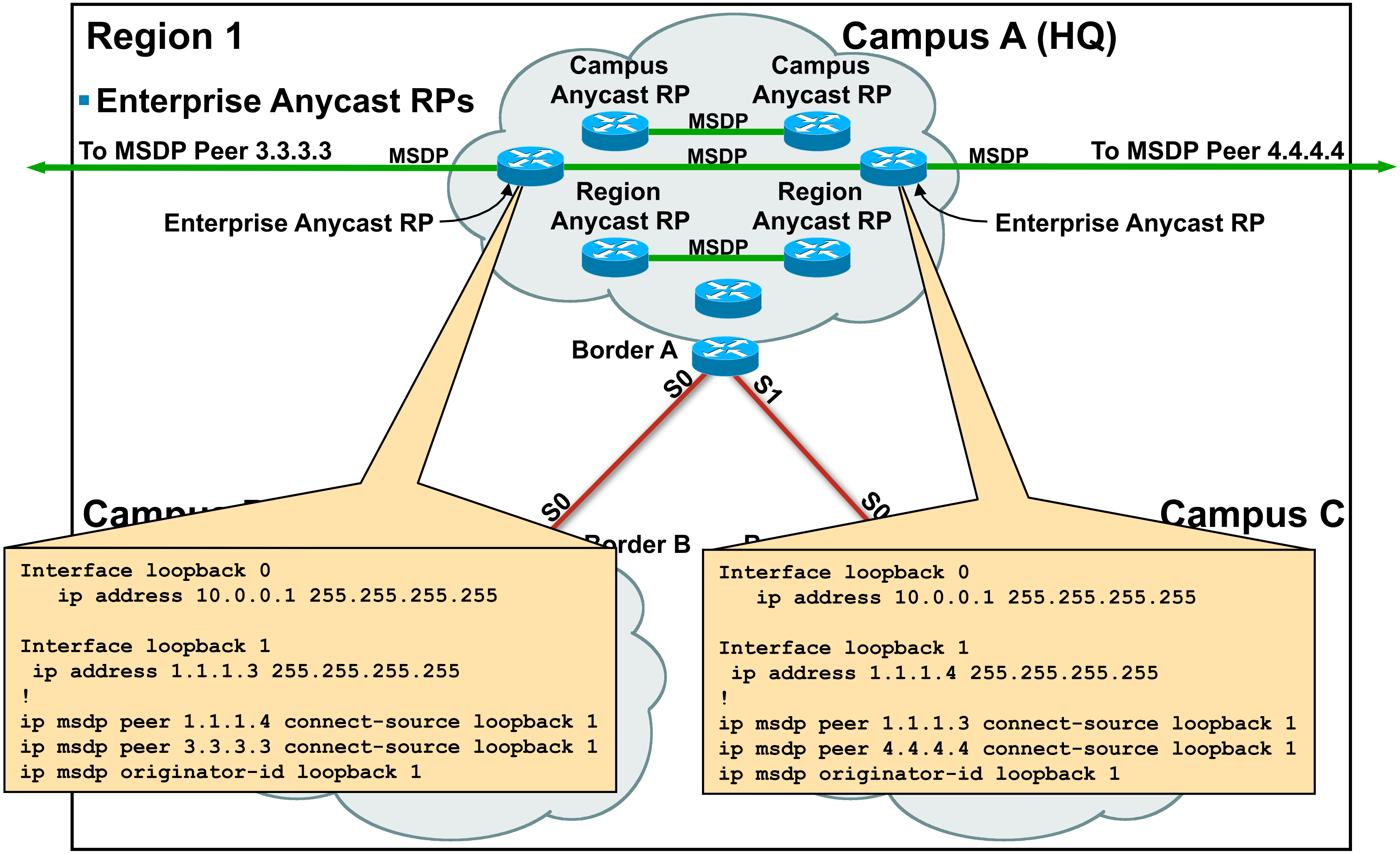
Administratively-Scoped Zones

Anycast RP Example



Administratively-Scoped Zones

Anycast RP Example



Administratively-Scoped Zones

Anycast RP Example