



IP Multicast

Phạm Huy Hoàng - SoICT/HUST
hoangph@soict.hust.edu.vn

ONE LOVE. ONE FUTURE.



Tổng quan về IP multicast

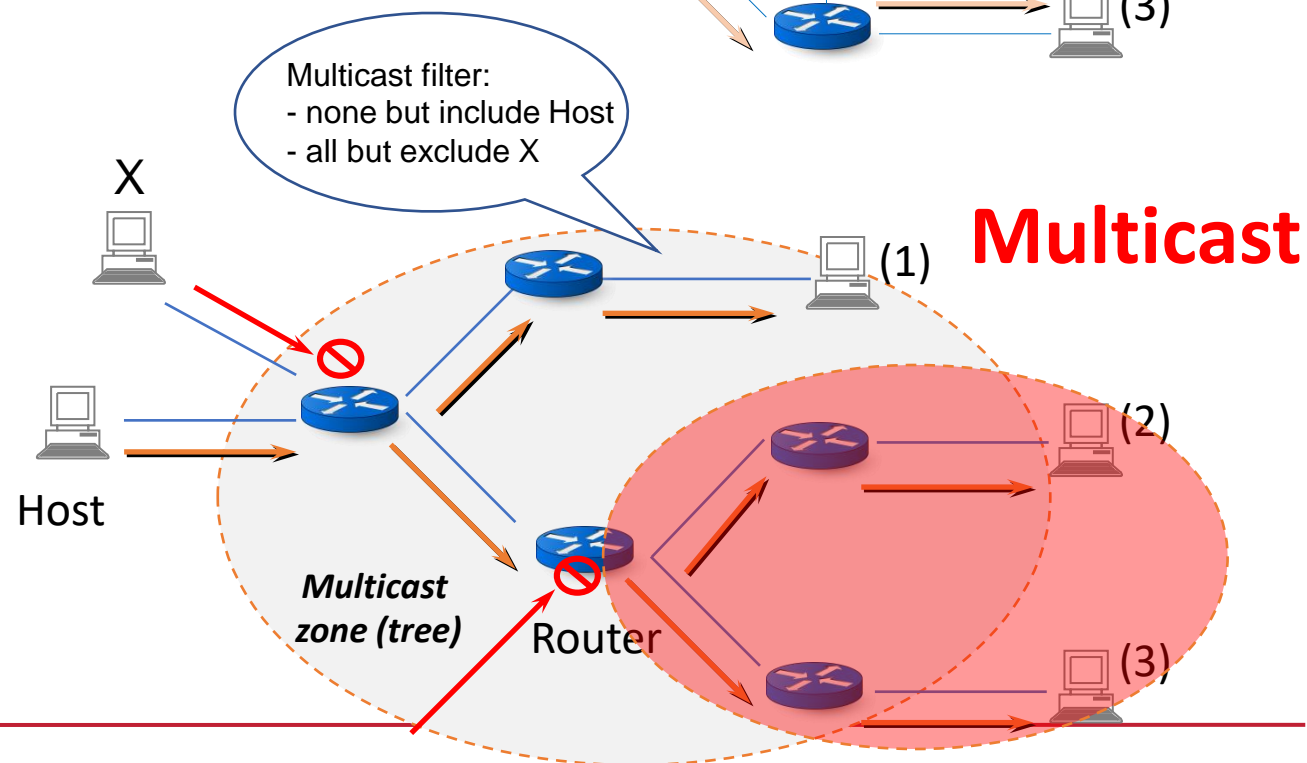
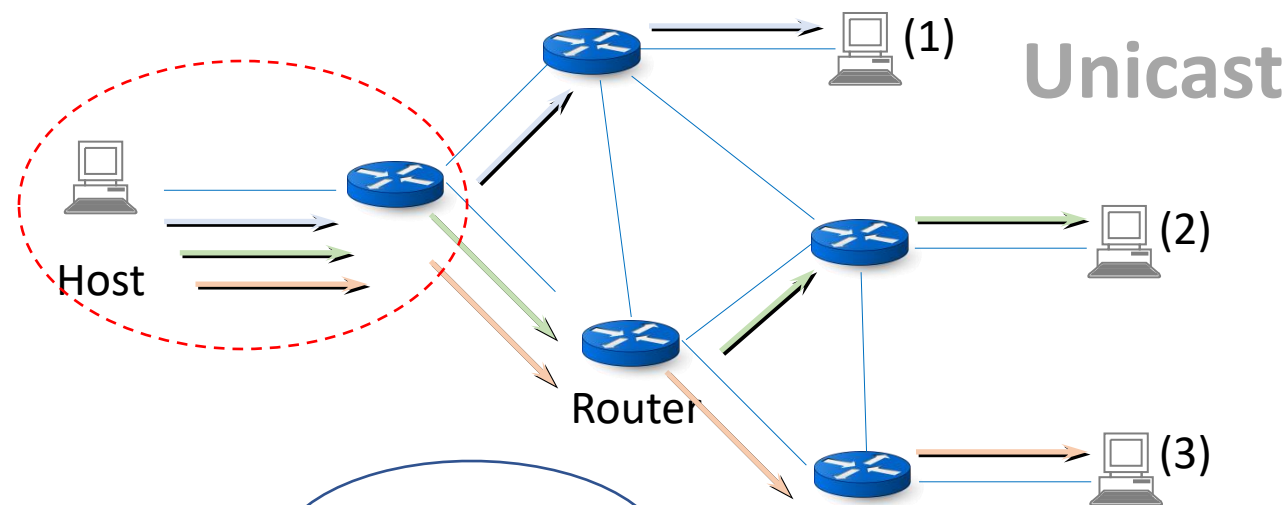
- Unicast & Multicast
- Địa chỉ multicast & giao thức 2 pha
- Demo
- Multicast tree & join host



Unicast & Multicast

- Tại sao cần multicast?
 - Gửi cùng dữ liệu đến nhiều người nhận
 - Tối ưu hóa đường truyền
- Ví dụ ứng dụng
 - Video/audio conferencing
 - Internet TV
 - Quảng cáo trực tuyến
 - Thông tin chứng khoán
 - V.v..
- Vấn đề an toàn thông tin
 - Lộ/mất kiểm soát cây multicast
 - Mọi router cần tham gia

→ **Vai trò của router
(multicast routing)**

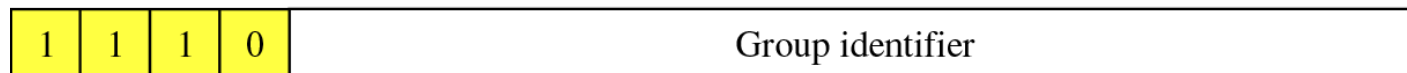


Địa chỉ IP Multicast

- IP Class:

- A: 0xxxxxxx.a.b.c
- B: 10xxxxxx.a.b.c
- C: 110xxxxx.a.b.c
- D: 1110xxxx.a.b.c

Class D address
Range: 224.0.0.0 to 239.255.255.255



- Các địa chỉ “224.0.0”:

- Link Local Addresses. IANA giữ các địa chỉ “224.0.0” để phục vụ routing cho mạng local. Các địa chỉ “link local” không bao giờ được route liên mạng, chỉ để các trạm liên lạc trực tiếp qua liên kết vật lý với gói tin có TTL = 1
- Mapping đến các địa chỉ multicast của mạng vật lý (ethernet)

- Một số địa chỉ link local:

- 224.0.0.1: các trạm trong mạng mà có hỗ trợ multicast
- 224.0.0.2: các router trong mạng
- 224.0.0.5: các OSPF routers
- 224.0.0.12: các DHCP Server
- 224.0.0.13: các router hỗ trợ PIM trong mạng (PIM: multicast routing)
- 224.0.0.22: router hỗ trợ IGMP
- 224.0.0.251: multicast DNS [RFC6762]
- Các địa chỉ multicast khác: <https://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml>

Đăng ký địa chỉ IP unicast vào “nhóm”

- Nguyên tắc hoạt động:
 - Một địa chỉ lớp D tương ứng với một nhóm trạm nhận (group identifier)
 - Gói tin gửi đến địa chỉ lớp D sẽ được các router xử lý “nhân” tại vị trí phù hợp để gửi đến từng trạm nhận (thay vì “nhân” tại gốc ~ n x unicast)
- Giao thức 2 pha:
 - Join: quá trình các host đăng ký với các router để join vào một địa chỉ lớp D
 - *Join tự động*: tùy theo chức năng của trạm mà tự động join vào các địa chỉ “224”
 - *Join chủ động*: đăng ký địa chỉ IP của mình với một địa chỉ IP multicast
 - Broadcast (multicast): quá trình lan truyền gói tin theo địa chỉ multicast đến các trạm đã join vào địa chỉ này trong pha “join”.

```
$ netstat -4 -g -n
IPv6/IPv4 Group Memberships
Interface      RefCnt Group
-----
lo              1      224.0.0.1
enp0s9          1      224.0.0.1
enp0s10         1      224.0.0.1

$ sudo systemctl start pimd
$ netstat -4 -g -n
IPv6/IPv4 Group Memberships
Interface      RefCnt Group
-----
lo              1      224.0.0.1
enp0s9          1      224.0.0.13
enp0s9          1      224.0.0.22
enp0s9          2      224.0.0.1
enp0s10         1      224.0.0.13
enp0s10         1      224.0.0.1
```

Linux

```
> netsh
netsh> int ip
netsh interface ipv4> show joins
```

Interface 25: Wi-Fi

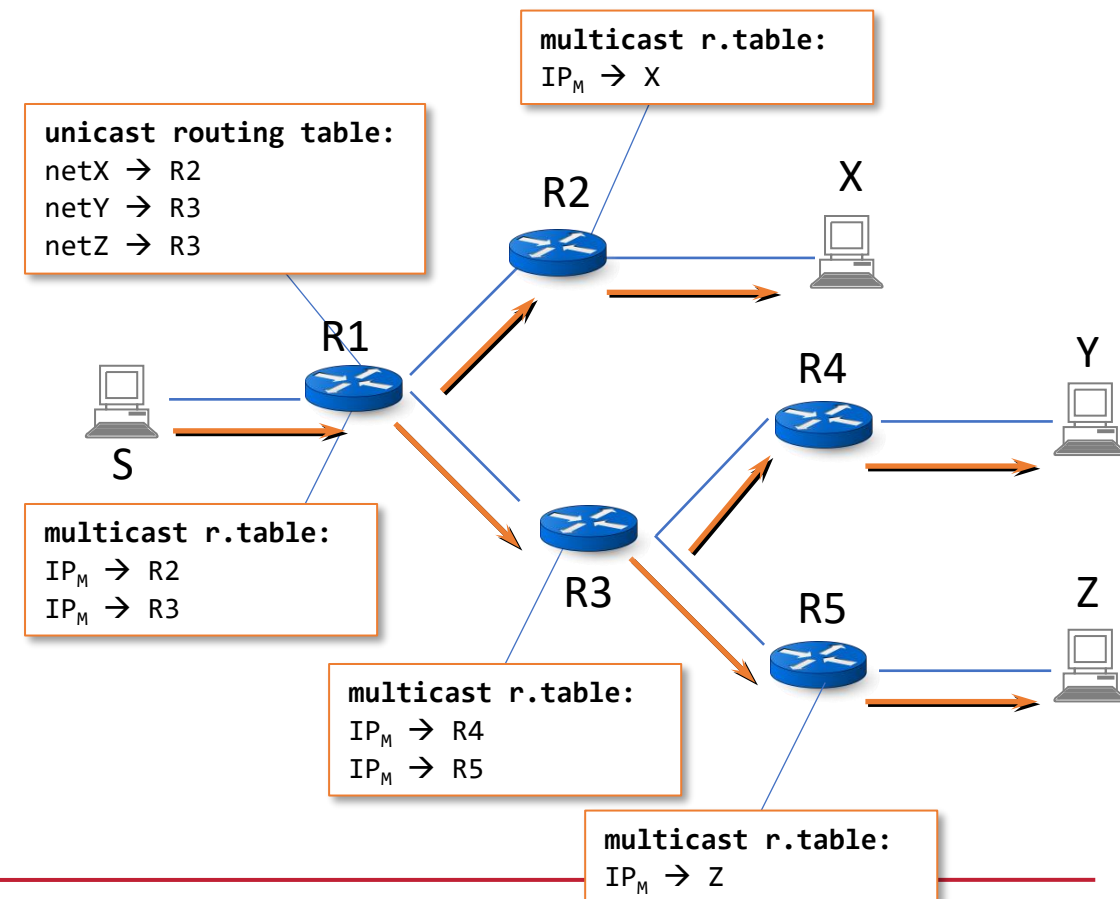
Scope	References	Last	Address
0	0	Yes	224.0.0.1
0	3	Yes	224.0.0.251
0	1	Yes	224.0.0.252
0	2	Yes	239.255.255.250

Windows

Xử lý địa chỉ nhóm trong router

- $IP_M = [IP_X, IP_Y, IP_Z]$
- IP_M phải được cấu hình tại tất cả các router trong hệ thống để router nhận biết và xử lý gói tin IP multicast
- IP_M được xử lý tại các router như một “destination network”
- Gói tin multicast: $IP_S \rightarrow IP_M$
- Router nhận gói tin IP, đối chiếu các địa chỉ IP trong gói tin với các “dòng multicast” trong bảng routing \rightarrow nhân bản gói tin IP theo từng “dòng” phù hợp & chuyển tiếp gói tin theo “next hop”
- Cú pháp cây source specific (S,G) & (*,G)
 - (192.168.1.120, 226.96.1.1)
 - (*, 226.96.1.1)

	Bits 0–7	Bits 8–15	Bits 16–23	Bits 24–31
Header (20 bytes)	Version/IHL	Type of service	Length	
	Identification		flags and offset	
	Time To Live (TTL)	Protocol	Header Checksum	
	Source IP address			IP_S
	Destination IP address			IP_M
ICMP Header (8 bytes)	Type of message	Code	Checksum	
	Header Data			
ICMP Payload (optional)	Payload Data			



Multicast Demo

- Host A phát tin theo nhóm 226.96.1.1
→ cây source specific (192.168.1.120, 226.96.1.1)

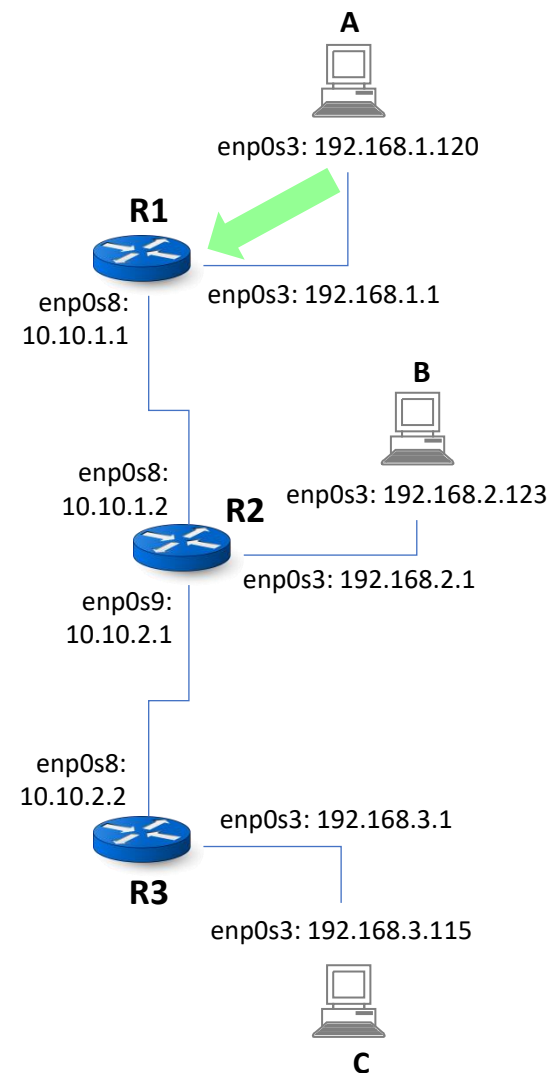
```
A:~$ iperf -c 226.96.1.1 -u
```

```
-----  
Client connecting to 226.96.1.1, UDP port 5001  
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)  
Setting multicast TTL to 1  
UDP buffer size: 208 KByte (default)  
-----
```

- Router R1 nhận được gói tin trên liên kết mạng với A

```
R1:~$ sudo tcpdump -i enp0s3 | grep 226.96.1.1
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes  
10:25:11.625716 IP 192.168.1.120.51134 > 226.96.1.1.5001: UDP, length 1470  
10:25:11.637009 IP 192.168.1.120.51134 > 226.96.1.1.5001: UDP, length 1470  
10:25:11.648147 IP 192.168.1.120.51134 > 226.96.1.1.5001: UDP, length 1470
```



Multicast Demo (2)

- Router R1 không chuyển tiếp gói tin từ *enp0s3* sang *enp0s8*

```
R1:~$ sudo tcpdump -i enp0s8 | grep 226.96.1.1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), capture size 262144 bytes
```

- Tạo luật forward gói tin multicast trong R1:

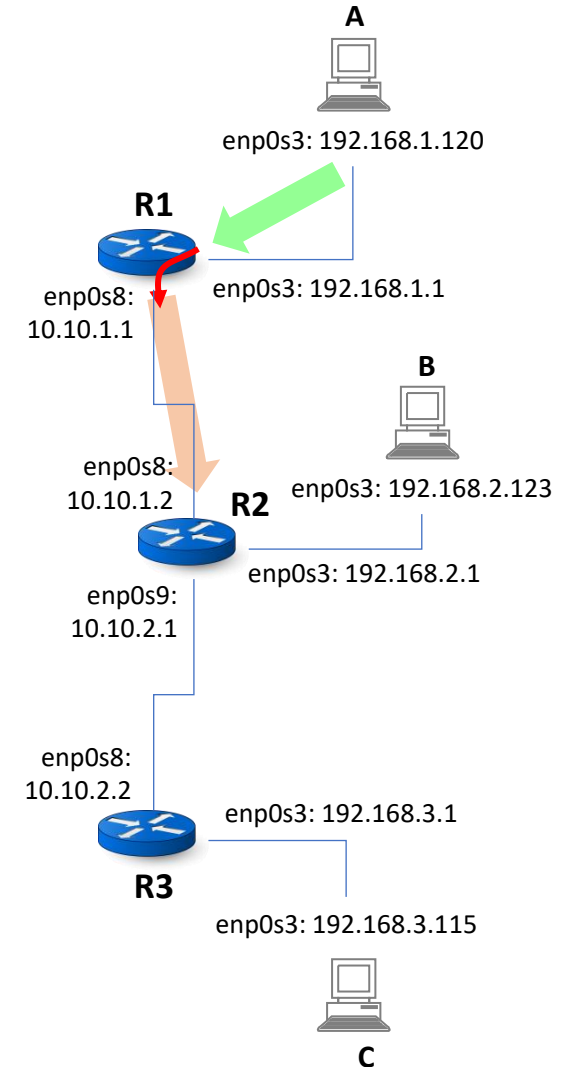
```
R1(config)# interface enp0s3
R1(config-if)# ip mroute enp0s8 226.96.1.1 192.168.1.120
R1(config-if)# end

R1# show ip mroute
Proto: I=IGMP P=PIM S=STATIC
```

Source	Group	Proto	Input	iVifI	Output	oVifI	TTL	Uptime
192.168.1.120	226.96.1.1	S	enp0s3	5	enp0s8	4	1	00:00:42

- Router R1 chuyển tiếp gói tin sang kết nối *enp0s8*

```
R1:~$ sudo tcpdump -i enp0s8 | grep 226.96.1.1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), capture size 262144 bytes
10:25:11.625716 IP 192.168.1.120.51134 > 226.96.1.1.5001: UDP, length 1470
10:25:11.637009 IP 192.168.1.120.51134 > 226.96.1.1.5001: UDP, length 1470
```



Multicast Demo (3)

- Router R2 nhận gói multicast trên *enp0s8* nhưng không chuyển tiếp

```
R2:~$ sudo tcpdump -i enp0s8 | grep 226.96.1.1
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on enp0s8, link-type EN10MB (Ethernet), capture size 262144 bytes  
10:25:11.625716 IP 192.168.1.120.51134 > 226.96.1.1.5001: UDP, length 1470
```

```
R2:~$ sudo tcpdump -i enp0s3 | grep 226.96.1.1
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on enp0s10, link-type EN10MB (Ethernet), capture size 262144 bytes
```

- Tạo luật forward gói tin multicast trong R2:

```
R2(config-if)# mroute enp0s3 226.96.1.1
```

```
R2(config-if)# mroute enp0s9 226.96.1.1
```

```
R2# show ip mroute
```

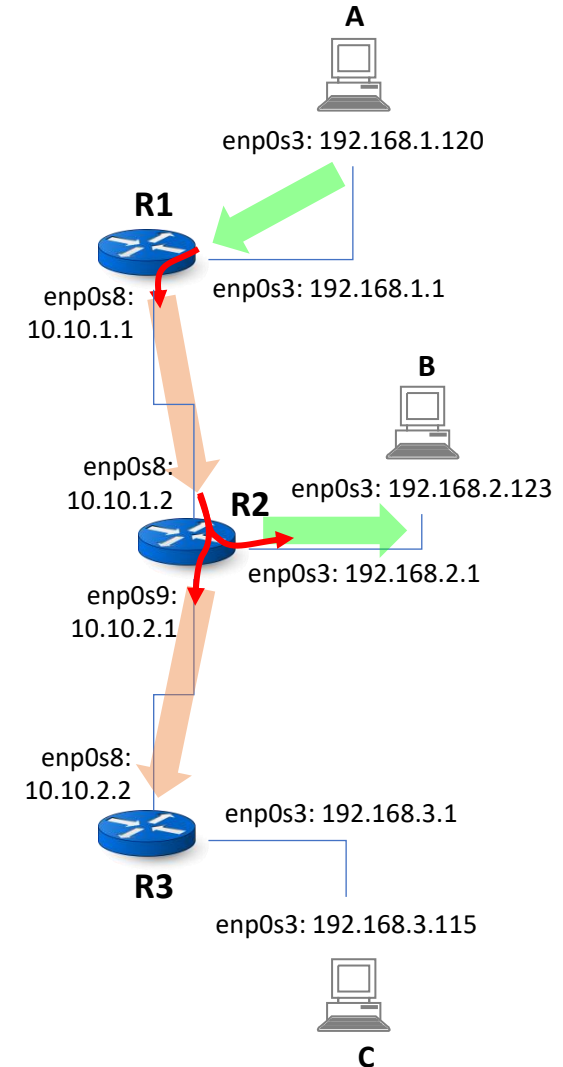
```
Proto: I=IGMP P=PIM S=STATIC
```

Source	Group	Proto	Input	iVifI	Output	oVifI	TTL	Uptime
*	226.96.1.1	S	enp0s8	5	enp0s3	4	1	00:00:42
*	226.96.1.1	S	enp0s8	5	enp0s9	4	1	00:00:42

- Router R2 chuyển tiếp gói tin sang 2 kết nối *enp0s9* và *enp0s3*

```
R2:~$ sudo tcpdump -i enp0s9 | grep 226.96.1.1
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on enp0s9, link-type EN10MB (Ethernet), capture size 262144 bytes  
10:25:11.637009 IP 192.168.1.120.51134 > 226.96.1.1.5001: UDP, length 1470
```



Multicast Demo (4)

- Host B nhận gói multicast phát từ A qua cây R1 → R2

```
B:~$ iperf -s -u -B 226.96.1.1 -i 1
```

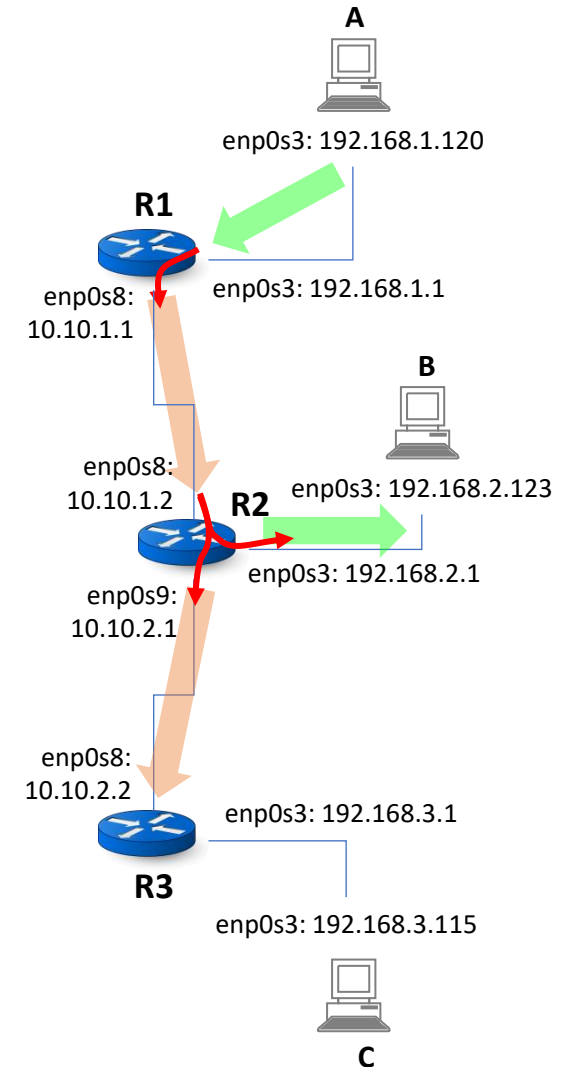
```
-----  
Server listening on UDP port 5001  
Binding to local address 226.96.1.1  
Joining multicast group 226.96.1.1  
Receiving 1470 byte datagrams  
UDP buffer size: 208 KByte (default)  
-----
```

```
[ 3] local 226.96.1.1 port 5001 connected with 192.168.1.120 port 34116  
[ ID] Interval      Transfer    Bandwidth    Jitter    Lost/Total Datagrams  
[ 3] 0.0- 1.0 sec   129 KBytes  1.06 Mbits/sec  0.045 ms    0/ 90 (0%)  
[ 3] 1.0- 2.0 sec   128 KBytes  1.05 Mbits/sec  0.061 ms    0/ 89 (0%)  
[ 3] 2.0- 3.0 sec   128 KBytes  1.05 Mbits/sec  0.053 ms    0/ 89 (0%)
```

- Router R3 nhận được gói tin multicast phát từ A qua cây R1 → R2 → R3

```
R3:~$ sudo tcpdump -i enp0s8 | grep 226.96.1.1
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on enp0s9, link-type EN10MB (Ethernet), capture size 262144 bytes  
10:25:11.637009 IP 192.168.1.120.51134 > 226.96.1.1.5001: UDP, length 1470  
10:25:11.637009 IP 192.168.1.120.51134 > 226.96.1.1.5001: UDP, length 1470
```



Multicast Demo (5)

- Tạo luật forward gói tin multicast trong R3:

```
R3# show ip mroute
Proto: I=IGMP P=PIM S=STATIC

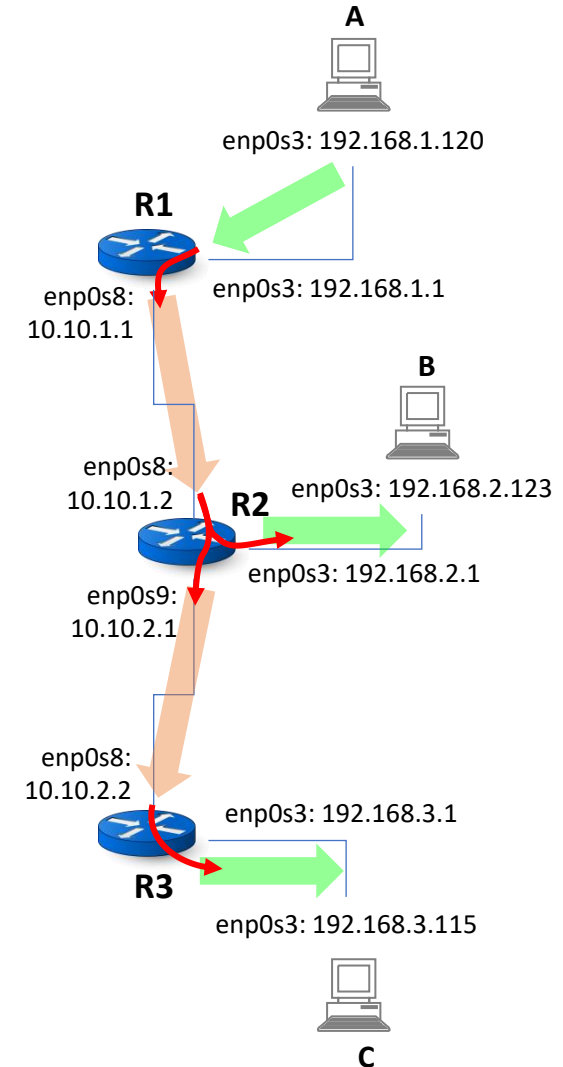
Source          Group          Proto Input   iVifI Output  oVifI TTL Uptime
192.168.1.120    226.96.1.1     S    enp0s8    5    enp0s3    4    1    00:00:42
```

- Router R3 chuyển tiếp gói tin sang *enp0s3*:

```
R3:~$ sudo tcpdump -i enp0s3 | grep 226.96.1.1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
10:25:11.637009 IP 192.168.1.120.51134 > 226.96.1.1.5001: UDP, length 1470
10:25:11.637009 IP 192.168.1.120.51134 > 226.96.1.1.5001: UDP, length 1470
```

- Host C nhận gói multicast phát từ A qua cây R1 → R2 → R3

```
C:~$ iperf -s -u -B 226.96.1.1 -i 1
. . . . .
[ 3] local 226.96.1.1 port 5001 connected with 192.168.1.120 port 34116
[ ID] Interval      Transfer      Bandwidth      Jitter    Lost/Total Datagrams
[ 3] 0.0- 1.0 sec    129 KBytes    1.06 Mbits/sec  0.045 ms   0/ 90 (0%)
[ 3] 1.0- 2.0 sec    128 KBytes    1.05 Mbits/sec  0.061 ms   0/ 89 (0%)
```



Tổng kết demo: Dựng cây multicast thủ công

R2\$ route -n

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
10.10.1.0	0.0.0.0	255.255.255.0	U	0	0	0	enp0s9
10.10.2.0	0.0.0.0	255.255.255.0	U	0	0	0	enp0s10
192.168.1.0	10.10.1.1	255.255.255.0	UG	0	0	0	enp0s9
192.168.2.0	10.10.2.2	255.255.255.0	UG	0	0	0	enp0s10

Khi A multicast đến B:

R2> show ip mroute

Proto: I=IGMP P=PIM S=STATIC

Source	Group	Proto	Input	iVifI	Output	oVifI	TTL	Uptime
192.168.1.120	226.96.1.1	S	enp0s9	4	enp0s3	2	1	00:08:06

Thông tin để đối chiếu với các trường địa chỉ trong gói tin IP

Thêm C nhận multicast:

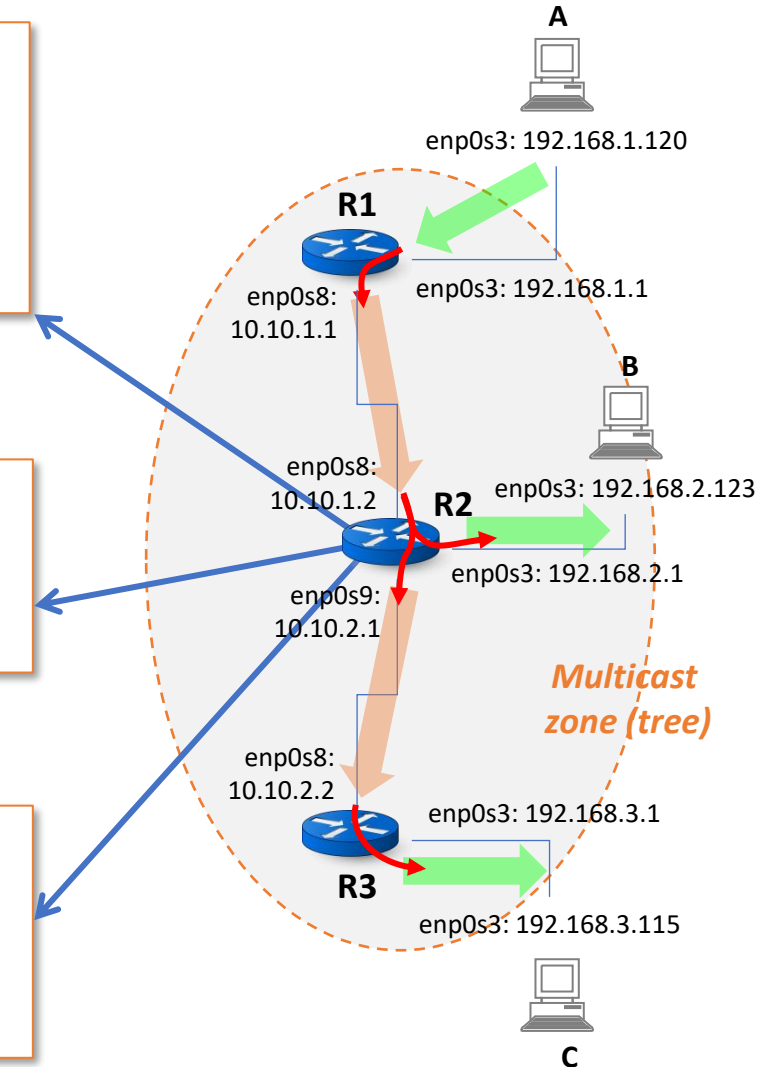
R2> show ip mroute

Proto: I=IGMP P=PIM S=STATIC

Source	Group	Proto	Input	iVifI	Output	oVifI	TTL	Uptime
192.168.1.120	226.96.1.1	S	enp0s9	4	enp0s3	2	1	00:09:31
192.168.1.120	226.96.1.1	S	enp0s9	4	enp0s10	5	1	00:00:09

Multicast protocol: tạo đường định tuyến static hoặc bằng m.protocol

Input & Output interface trong tuyến multicast



Tổng kết demo(2): Gắn nút lá vào cây multicast

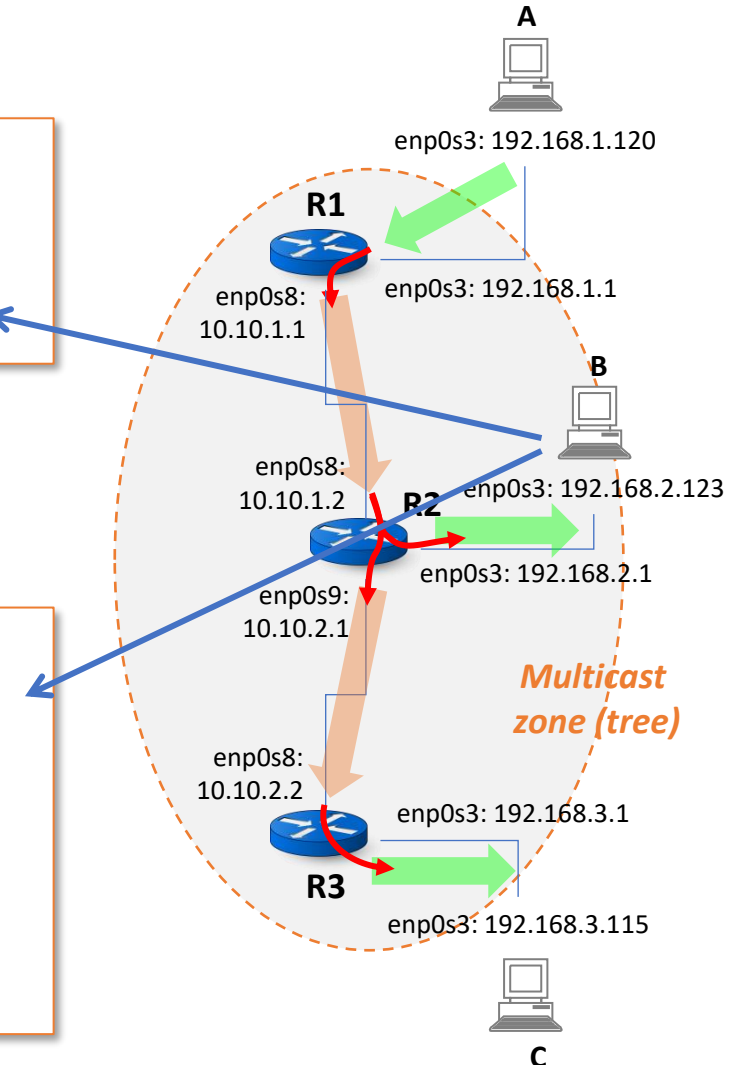
- Giao diện kết nối mạng của mỗi host có thể được join với địa chỉ multicast (giống như được gán nhiều địa chỉ IP unicast)
- Khi gói tin multicast IP được router forward đến giao diện kết nối mạng của host mà đã join vào địa chỉ multicast (nút lá) → host xử lý nhận gói tin như nhận gói tin IP unicast thông thường
- Trường hợp nguồn phát được xác định trước (source specific multicast), các giao thức liên quan đến pha “join” của multicast chịu trách nhiệm cập nhật bảng routing của các router trong multicast zone (tree) và join host vào địa chỉ multicast tương ứng
- Trường hợp nguồn phát không xác định khi join, router chỉ có thể lưu các thông tin đăng ký này và đợi khi nguồn phát bắt đầu phát tin để thiết lập cây multicast

```
B:~$ ip -4 maddress
1:      lo
        inet  224.0.0.1
2:      enp0s10
        inet  224.0.0.1
```

B đăng ký nhận multicast:

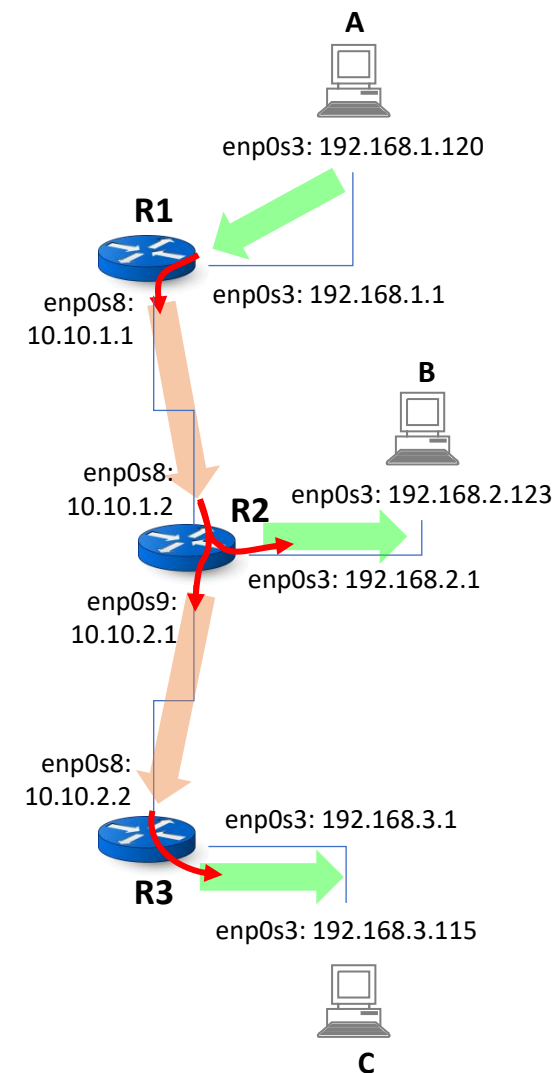
```
B:~$ iperf -s -u -B
226.96.1.1 -i 1

H2:~$ ip -4 maddress
1:      lo
        inet  224.0.0.1
2:      enp0s10
        inet  226.96.1.1
        inet  224.0.0.1
```



Thực hành: Static Multicast

- Tạo cây multicast R1 → R2 bằng phương pháp static
- Tạo nguồn phát trên A với *iperf -c*
- Tạo trạm nghe trên B với *iperf -s*
- Theo dõi các gói tin multicast gửi từ A vào nhóm và gói tin được chuyển tiếp qua R1, R2 đến B
- Thêm R3 vào cây multicast cũng bằng phương pháp static để hoàn thành cây R1 → R2 → R3
- Tạo trạm nghe trên C với *iperf -s*
- Theo dõi các gói tin multicast gửi từ A vào nhóm và được chuyển tiếp qua R1, R2, R3 đến H3



A decorative background consisting of a large number of red dots of varying sizes. These dots are arranged in a circular pattern, with the density of the dots increasing towards the right side of the image, creating a sense of depth and movement.

Internet Group Management Protocol (IGMP)

ONE LOVE. ONE FUTURE.

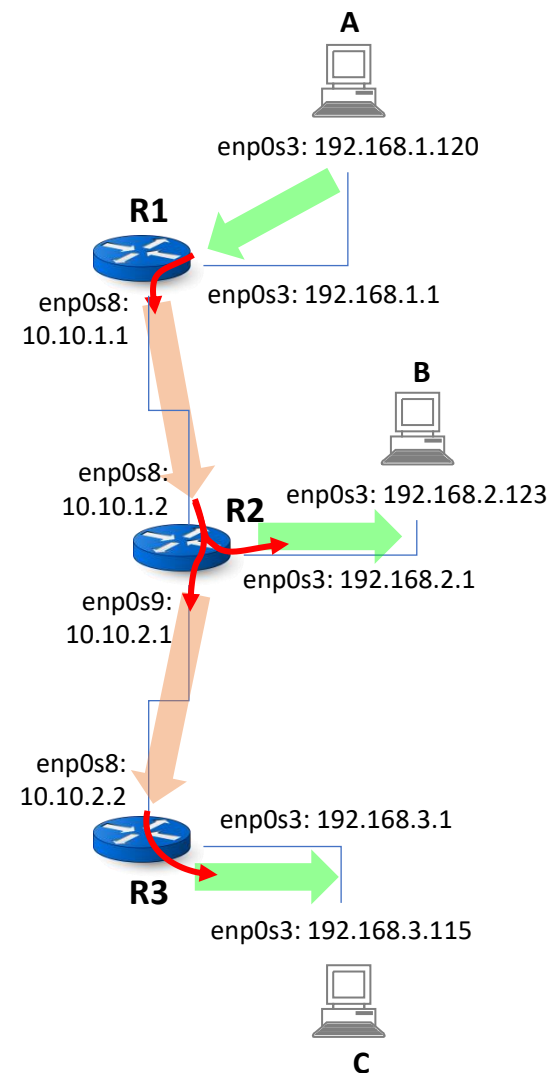
Static (demo) & yêu cầu thực tế

- Demo static multicast:

- Cây (S,G) được qui hoạch sẵn trên backbone network
- Dựng cây bằng các luật multicast trên từng router
- Nguồn phát vào “gốc” cây & join nút thu vào “ngọn” khi cần
- Khi (tất cả) nút thu rời khỏi một cành cây
 - ➔ cây không cập nhật (dòng multimedia vẫn phát trên cành không lá)

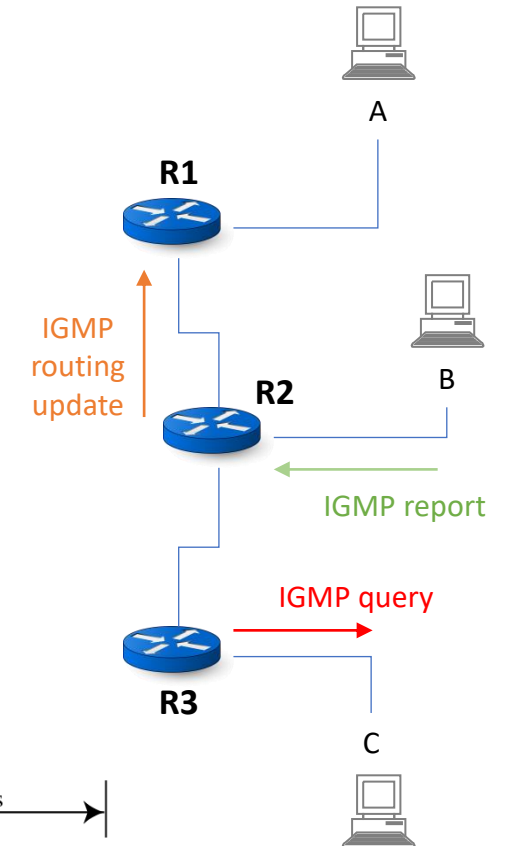
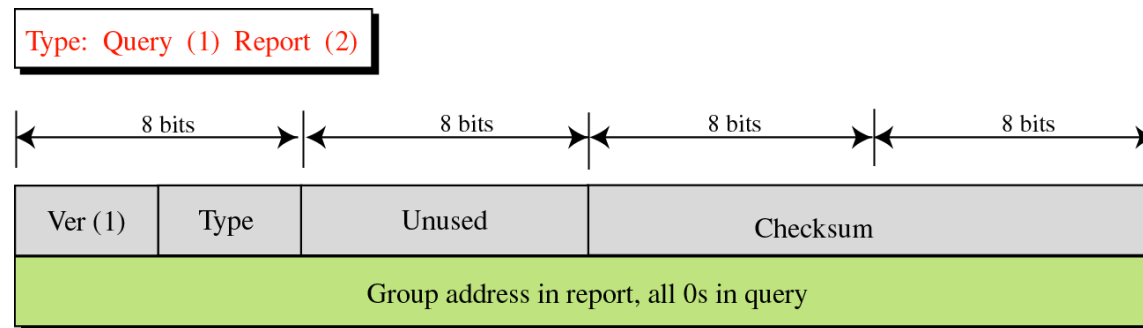
- Yêu cầu thực tế:

- Mật độ trạm nhận có thể dày đặc (dense) hoặc thưa thớt (sparse)
- Nguồn phát xác định trước (source specific)
- Nguồn phát không xác định (*, G) ➔ giải pháp Rendezvous Point
- Nhiều nguồn phát ([a, b, c], G)



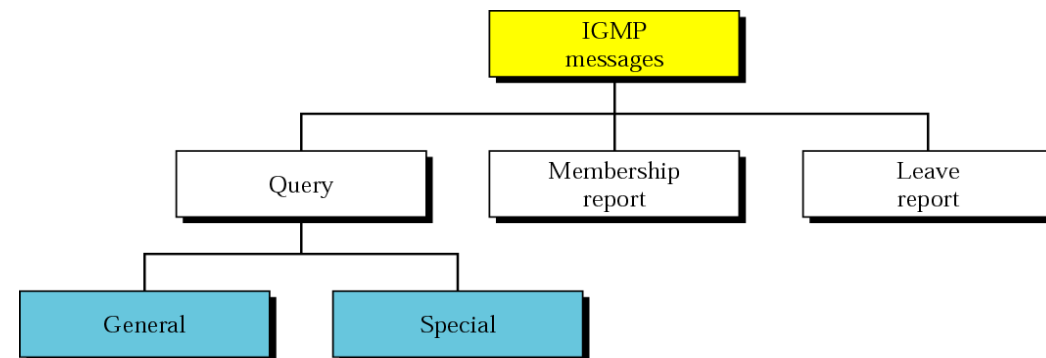
Internet Group Management Protocol (IGMP)

- Dùng để join/leave nút (router/host) trong nhóm (multicast IP)
- Cũng có thể cập nhật thông tin multicast routing giữa các router
- Chạy trực tiếp trên tầng IP với protocol number=2 (giống ICMP, p.num=1)
- IGMPv1 (RFC1112), v2 (RFC2236), v3 (RFC3376)
- IPv6:
 - ICMPv6 cài đặt chức năng Multicast Listener Discovery (MDL)
 - MLDv1 tương tự IGMPv2
 - MLDv2 tương tự IGMPv3
- IGMP query:
 - Gửi đến 224.0.0.1 hoặc group address
 - Router truy vấn host
- IGMP report:
 - Gửi đến 224.0.0.22
 - Host yêu cầu router
 - Thay đổi thông tin nhóm



IGMP Message: Query

- General Query:
 - Group Address = 0
 - Router gửi well known multicast (224.0.0.1) để thu thập thông tin routing mạng
 - Router gửi well known multicast (224.0.0.22) để thu thập thông tin liên quan đến IGMP
- Group-Specific Query
 - Group Address = group multicast IP address
 - Router gửi query đến địa chỉ multicast group để yêu cầu thu thập thông tin trong group
- Source Addresses:
 - Các địa chỉ nguồn phát (unicast) của nhóm

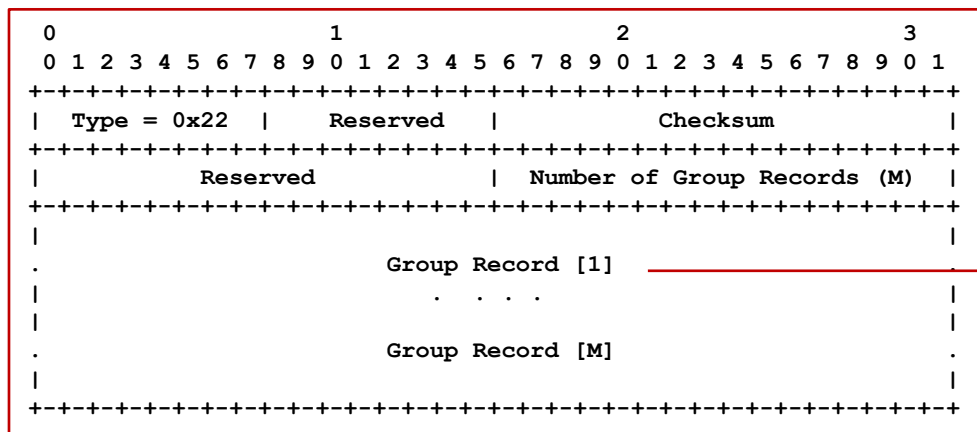


IGMP Query

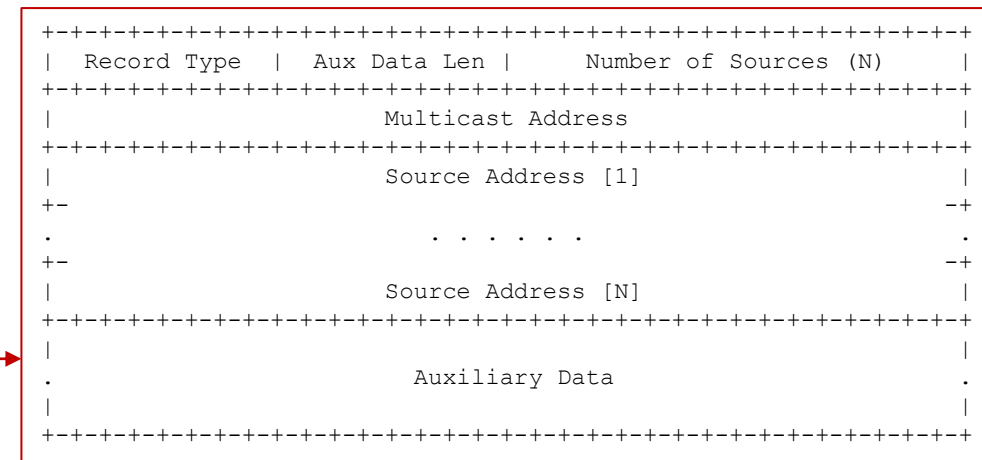
```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Type = 0x11 | Max Resp Code | Checksum |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Group Address |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Resv | S | QRV | QQIC | Number of Sources (N) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Source Address [1] |
+-+-----+-----+-----+-----+-----+-----+-----+-----+
| Source Address [2] |
+-+-----+-----+-----+-----+-----+-----+-----+-----+
| . |
| . |
+-+-----+-----+-----+-----+-----+-----+-----+-----+
| Source Address [N] |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

IGMP Message: Report

IGMP Report



Group Record



Tôi (sender) muốn làm gì với nhóm
(multicast address) cùng với nguồn phát
(source address)

Current-State Record: trả lời khi nhận được query

- MODE_IS_INCLUDE: nhận nguồn phát
- MODE_IS_EXCLUDE: không nhận nguồn phát

Filter-Mode-Change Record: thay đổi chế độ làm việc (filter mode)

- CHANGE_TO_INCLUDE_MODE
- CHANGE_TO_EXCLUDE_MODE

Source-List-Change Record: thêm hoặc xóa nguồn phát trong nhóm

- ALLOW_NEW_SOURCES
- BLOCK_OLD_SOURCES

State-Change record

- Các host khi tham gia vào nhóm và muốn thay đổi nguồn phát (hoặc muốn kết thúc tham gia nhóm) sẽ sử dụng `report to_in()` hoặc `to_ex()` với nội dung nguồn phát phù hợp
- Bảng chuyển trạng thái trước & sau của kết nối tùy theo report gửi đi:

Old State	New State	State-Change Record Sent
INCLUDE (A)	INCLUDE (B)	ALLOW (B-A), BLOCK (A-B)
EXCLUDE (A)	EXCLUDE (B)	ALLOW (A-B), BLOCK (B-A)
INCLUDE (A)	EXCLUDE (B)	TO_EX (B)
EXCLUDE (A)	INCLUDE (B)	TO_IN (B)

```
R1:~$ sudo tcpdump -n -i enp0s10 -vv
10.10.1.2 > 224.0.0.22: \
  igmp v3 report, 1 group record(s) \
  [gaddr 226.96.1.1 \
    allow { 192.168.1.120 }]
10.10.1.2 > 224.0.0.22: \
  igmp v3 report, 2 group record(s) \
  [gaddr 224.0.0.22 to_in { }] \
  [gaddr 226.96.1.1 \
    block { 192.168.1.120 }]
```

(S=192.168.1.120, G=226.96.1.1)

IGMP Message: Leave

- IGMPv3 không thiết kế thông điệp để yêu cầu leave như trường hợp join bằng IGMP Report, thay vào đó có thể dùng thông điệp chuyển trạng thái (state-change report) kết hợp với nguồn phát.
- **CHANGE_TO_INCLUDE_MODE & SOURCE = []**
 - Host gửi report thông báo hoạt động của mình trong nhóm 226.96.1.1 cần chuyển sang chế độ INCLUDE
 - Nhưng không nhận nguồn phát nào
 - ➔ tương đương yêu cầu rời khỏi nhóm này (leave)
- **CHANGE_TO_EXCLUDE_MODE & SOURCE = []**
 - Host gửi report thông báo hoạt động của mình trong nhóm 226.96.1.1 cần chuyển sang chế độ EXCLUDE
 - Nhưng không có nguồn phát nào
 - ➔ tương đương yêu cầu tham gia nhóm với nguồn phát chưa xác định
- **CHANGE_TO_INCLUDE_MODE & SOURCE <> []**
 - Yêu cầu join vào nhóm với nguồn phát xác định

```
R3:~$ sudo tcpdump -n -i enp0s10 -vv
192.168.3.115 > 224.0.0.22: \
  igmp v3 report, 1 group record(s) \
  [gaddr 226.96.1.1 to_in { }]
```

```
R3:~$ sudo tcpdump -n -i enp0s10 -vv
192.168.3.115 > 224.0.0.22: \
  igmp v3 report, 1 group record(s) \
  [gaddr 226.96.1.1 to_ex { }]
```

IGMP demo

- Join C vào nhóm multicast với nguồn phát A. Thông điệp IGMP Report được gửi từ C cho R3:

```
C:~$ iperf -s -u -B 226.96.1.1 -H 192.168.1.120 -i 1
```

```
-----  
Server listening on UDP port 5001  
Binding to local address 226.96.1.1  
Joining multicast (S,G)=192.168.1.120,226.96.1.1
```

- R3 được tự động join vào nhóm multicast cùng với luật chuyển tiếp gói tin từ enp0s9 sang enp0s10, được thiết lập theo protocol IGMP:

```
R3> show ip igmp groups
```

Interface	Address	Group	Mode	Timer	Srcs	V	Uptime
enp0s9	10.10.2.2	224.0.0.22	EXCL	00:04:03	0	3	00:01:02
enp0s10	192.168.3.1	224.0.0.22	EXCL	00:03:55	0	3	00:01:02
enp0s10	192.168.3.1	226.96.1.1	INCL	--:--:--	1	3	00:00:35

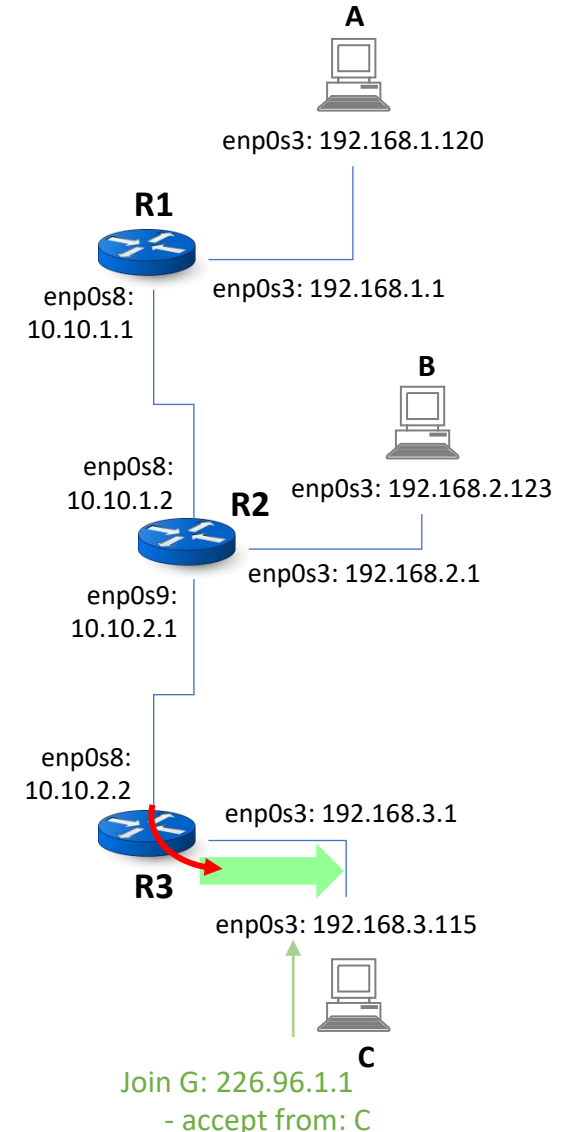
```
R3> show ip igmp sources
```

Interface	Address	Group	Source	Timer	Fwd	Uptime
enp0s10	192.168.3.1	226.96.1.1	192.168.1.120	04:12	Y	00:00:46

```
R3> show ip mroute
```

```
Proto: I=IGMP P=PIM S=STATIC
```

Source	Group	Proto	Input	iVifI	Output	oVifI	TTL	Uptime
192.168.1.120	226.96.1.1	I	enp0s9	4	enp0s10	5	1	00:00:13



IGMP demo (2)

- R3 kích hoạt giải thuật tạo cây multicast. Các router sử dụng PIM tự động dựng các luật trong bảng multicast routing để tạo cây

```
R3# show ip mroute
```

Source	Group	Proto	Input	Output	TTL	Uptime
192.168.1.120	226.96.1.1	IGMP	enp0s8	enp0s3	1	00:00:07

```
R2# show ip mroute
```

Source	Group	Proto	Input	Output	TTL	Uptime
192.168.1.120	226.96.1.1	PIM	enp0s8	enp0s9	1	00:00:22

```
R1# show ip mroute
```

Source	Group	Proto	Input	Output	TTL	Uptime
192.168.1.120	226.96.1.1	PIM	enp0s3	enp0s8	1	00:00:12

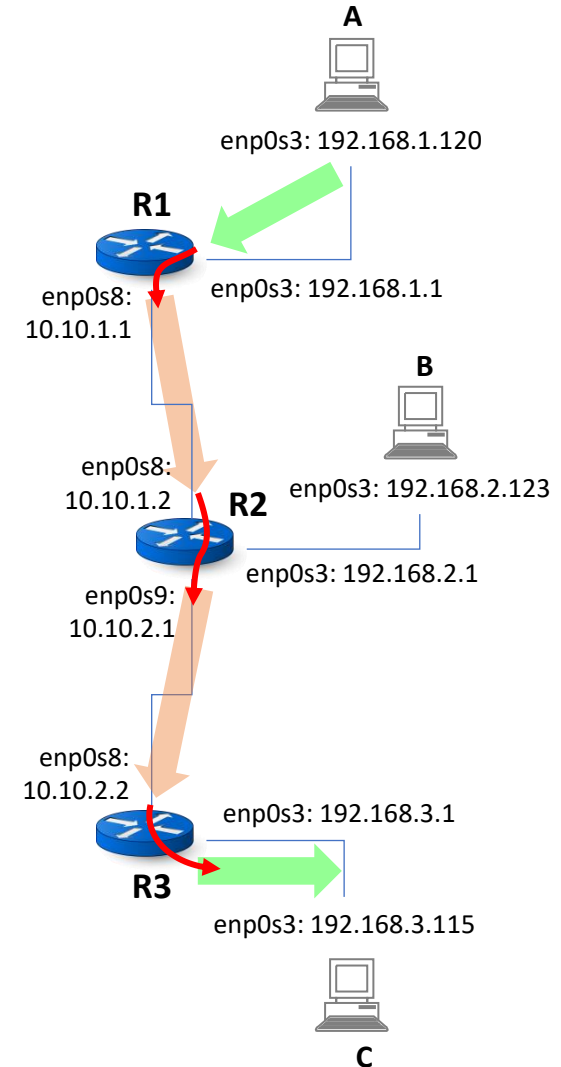
- A phát multicast, C nhận được:

```
A:~$ iperf -c 226.96.1.1 -u -T 10 -i 1
```

[3]	0.0- 1.0 sec	131 KBytes	1.07 Mbits/sec
[3]	1.0- 2.0 sec	128 KBytes	1.05 Mbits/sec
[3]	2.0- 3.0 sec	128 KBytes	1.05 Mbits/sec

```
C:~$ iperf -s -u -B 226.96.1.1 -H 192.168.1.120 -i 1
```

[3]	0.0- 1.0 sec	129 KBytes	1.06 Mbits/sec	0.871 ms	0/	90 (0%)
[3]	1.0- 2.0 sec	128 KBytes	1.05 Mbits/sec	0.584 ms	0/	89 (0%)
[3]	2.0- 3.0 sec	128 KBytes	1.05 Mbits/sec	0.738 ms	0/	89 (0%)



IGMP demo (3)

- B tham gia vào nhóm multicast vào nhóm, IGMP được gửi đến R2

```
B:~$ iperf -s -u -B 226.96.1.1 -H 192.168.1.120 -i 1
```

```
-----  
Server listening on UDP port 5001  
Binding to local address 226.96.1.1  
Joining multicast (S,G)=192.168.1.120,226.96.1.1
```

- R2 xử lý IGMP, tạo luật multicast để nối thêm cành vào cây:

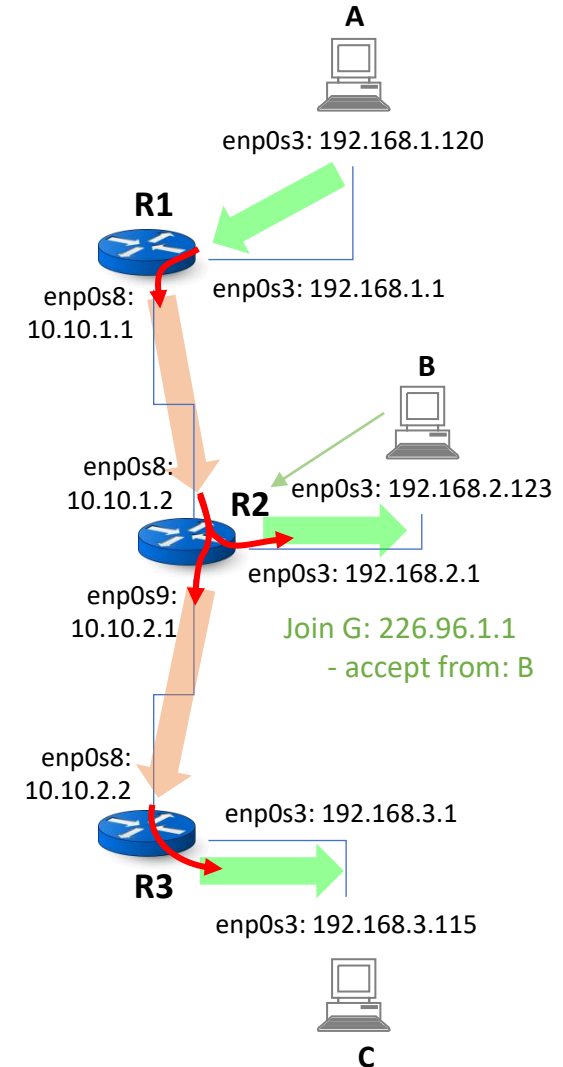
```
R2# show ip mroute
```

Source	Group	Proto	Input	Output	TTL	Uptime
192.168.1.120	226.96.1.1	IGMP	enp0s8	enp0s3	1	00:00:03
		IGMP		enp0s9	1	00:00:03

- B nhận được dòng multicast từ A

```
B:~$ iperf -s -u -B 226.96.1.1 -H 192.168.1.120 -i 1
```

```
[ 3] 0.0- 1.0 sec 129 KBytes 1.06 Mbits/sec 0.871 ms 0/ 90 (0%)  
[ 3] 1.0- 2.0 sec 128 KBytes 1.05 Mbits/sec 0.584 ms 0/ 89 (0%)  
[ 3] 2.0- 3.0 sec 128 KBytes 1.05 Mbits/sec 0.738 ms 0/ 89 (0%)
```



IGMP demo (4)

- C kết thúc *iperf*, R3 nhận được IGMP từ C yêu cầu rời khỏi cây

```
192.168.3.115 > 224.0.0.22: igmp v3 report, 1 group record(s)
[gaddr 226.96.1.1 block { 192.168.1.120 }]
192.168.3.1 > 226.96.1.1: igmp query v3 [max resp time 1.0s]
[gaddr 226.96.1.1 { 192.168.1.120 }]
```

- R2 xử lý IGMP, tạo luật multicast để nối thêm cành vào cây:

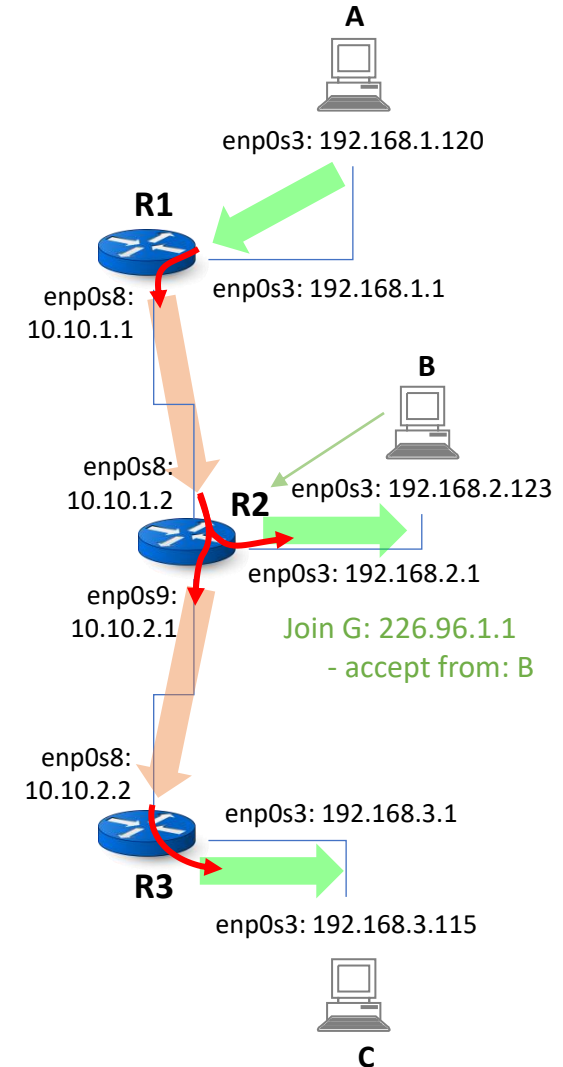
```
R2# show ip mroute
```

Source	Group	Proto	Input	Output	TTL	Uptime
192.168.1.120	226.96.1.1	IGMP	enp0s8	enp0s3	1	00:00:03
		IGMP		enp0s9	1	00:00:03

- B nhận được dòng multicast từ A

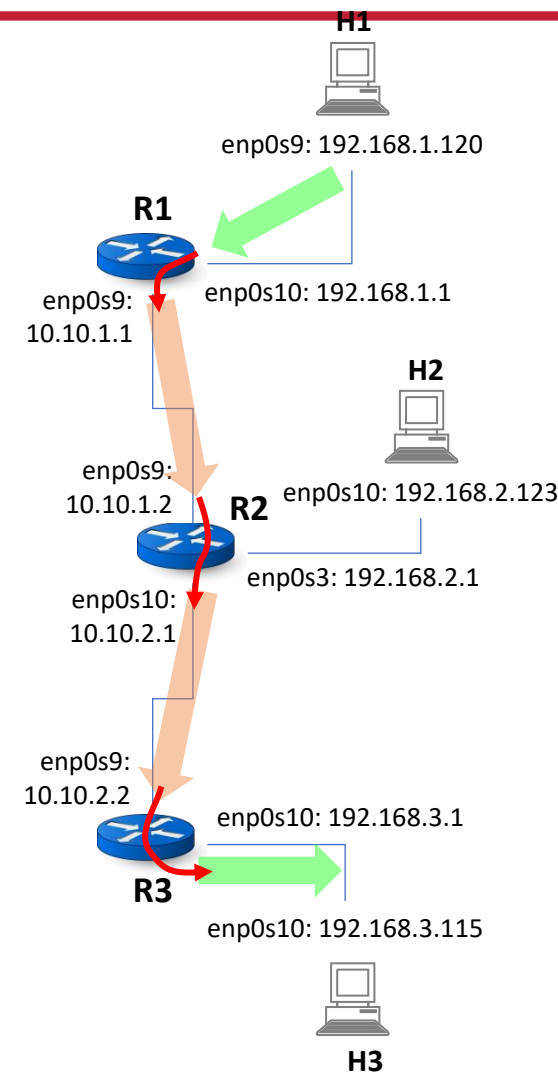
```
B:~$ iperf -s -u -B 226.96.1.1 -H 192.168.1.120 -i 1
```

[3]	0.0- 1.0 sec	129 KBytes	1.06 Mbits/sec	0.871 ms	0/	90 (0%)
[3]	1.0- 2.0 sec	128 KBytes	1.05 Mbits/sec	0.584 ms	0/	89 (0%)
[3]	2.0- 3.0 sec	128 KBytes	1.05 Mbits/sec	0.738 ms	0/	89 (0%)



Thực hành IGMP

- Thiết lập hệ thống mạng với 3 router và 3 host
- Chạy iperf trên H3 để join vào nhóm multicast với nguồn phát được xác định là H1
- Kiểm tra cấu hình multicast trên router R3 được tự động thiết lập theo gói tin IGMP Report gửi đến từ H3
- Join R3 (interface enp0s9) vào cây multicast, kiểm tra cấu hình multicast trên router R2 được tự động thiết lập theo gói tin IGMP Report gửi đến từ R3
- Join R2 (interface enp0s9) vào cây multicast, kiểm tra cấu hình multicast trên router R1 được tự động thiết lập theo gói tin IGMP Report gửi đến từ R2
- Chạy iperf trên H1 để phát gói tin vào nhóm, kiểm tra H3 nhận được gói tin phát trong nhóm
- Bắt các gói tin IGMP giữa H3-R3 và R2-R3 để phân tích





Multicast zone (tree) Theory

ONE LOVE. ONE FUTURE.

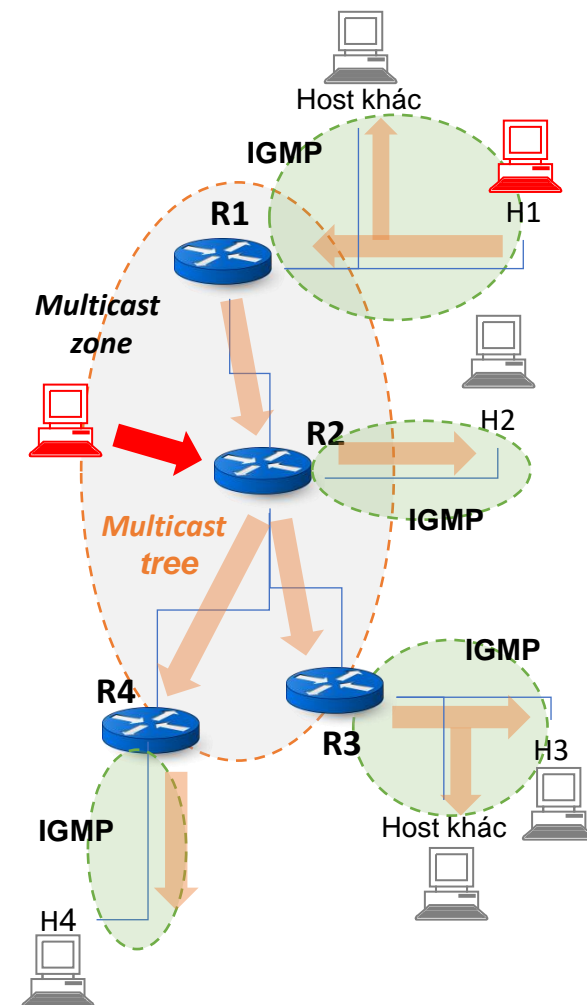
Multicast zone & Multicast tree

- Multicast zone
 - Gắn với một địa chỉ group (multicast)
 - Vùng lan tỏa của gói tin IP gửi theo địa chỉ group, gồm có các host & các router
- Multicast tree
 - Cây!!!
 - Host: nút gốc (source) & nút lá (receiver)
 - Router: nút bên trong
- Xây dựng cây
 - Dựng cây thủ công (manual) → static
 - IGMP: local link, thêm/bớt nút con vào cha
 - Dựng cây tự động (dynamic) → multicast routing protocol (nhớ lại routing protocol!!!)
- Bài toán nút gốc (source)
 - Xác định nút gốc thế nào? (source chạy *iperf -c* & gửi multicast)
 - Khi nào? (khi H3 chạy *iperf -H*, hoặc khi H1 chạy *iperf -c*)
 - Nhiều nút gốc? (các host receiver chạy *iperf -H* để thiết lập source khác nhau)
- Bài toán xác định cây tối ưu từ đồ thị kết nối mạng

```
R3> show ip mroute
```

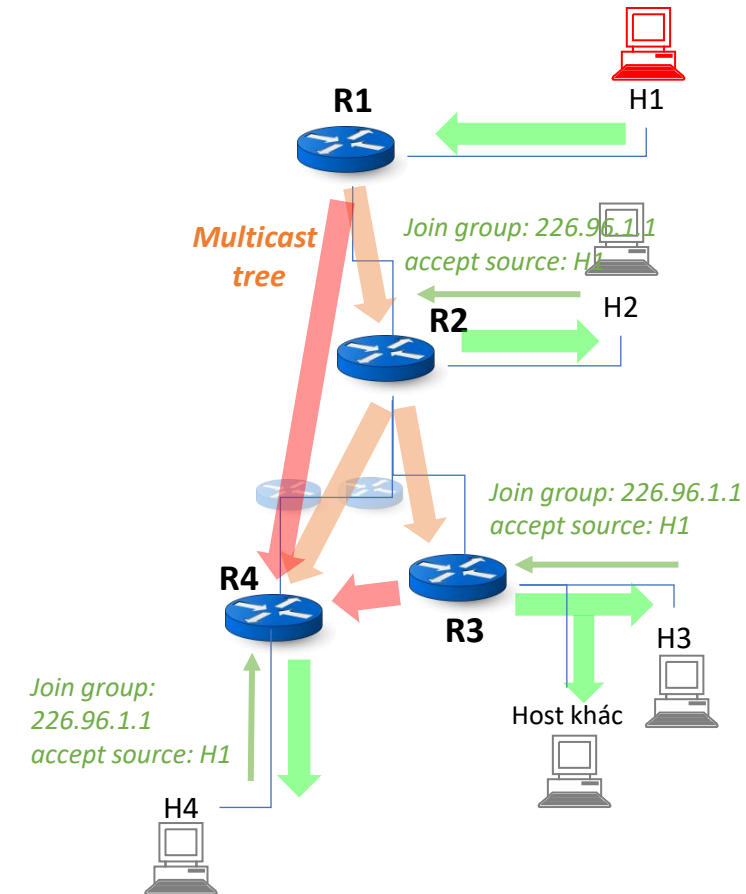
```
Proto: I=IGMP P=PIM S=STATIC
```

Source	Group	Proto	Input ...
192.168.1.120	226.96.1.1	S	enp0s9 ...



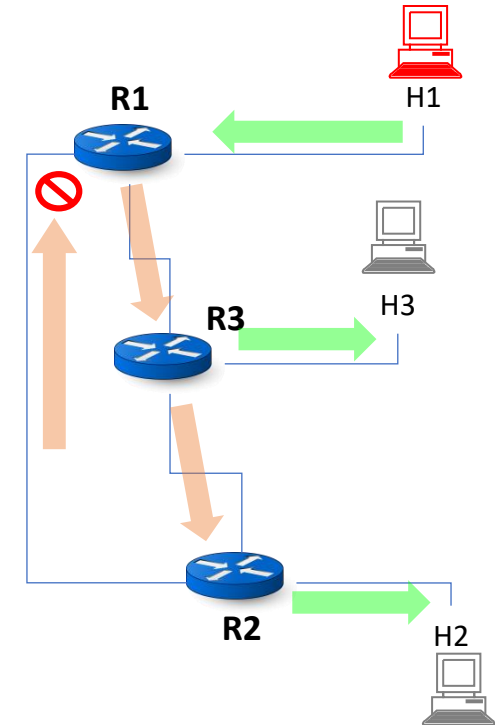
Xây dựng cây multicast

- Dựa vào nút gốc, tạo cây trong pha “join” (source specific mode) hoặc tạo cây trong pha truyền multicast (lúc đó source mới xuất hiện)
- Hình thành dần dần, khi có yêu cầu thêm (các) trạm nghe
- Local router nhận IGMP join report từ trạm nghe → tìm đến cây multicast đang có, hoặc khởi tạo cây nếu chưa có
- Local router thiết lập cành cây mới (có thể gồm nhiều router trung gian) để nối vào nút “phù hợp” trên cây multicast đã được thiết lập
- **Có nhiều lựa chọn để gắn cành mới, chọn phù hợp nhất?**
- Shortest Path Tree (SPT): cây con (cành) có gốc tại source hoặc một nút trên cây đã có, và lan tỏa đến (các) trạm nghe theo đường đi “tối ưu”
- Reverse-Path Forwarding (RPF): quá trình xuất phát từ một nút, qua các router trung gian để tìm ngược về source → xác định nút cha & tự gắn vào cây multicast (quá trình chuyển tiếp ngược - RPF)



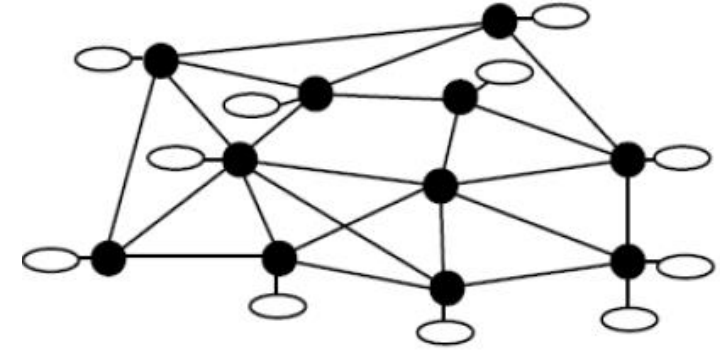
Thuật toán tràn (flooding)

- Kỹ thuật đơn giản nhất, chuyển tiếp gói tin multicast đến tất cả các router, trừ router mà nó nhận được gói tin này
- Nếu nhận được gói tin multicast đã xử lý thì không được chuyển tiếp nữa (lặp chu trình)
- Tối ưu multicast zone bằng các địa chỉ well known local link: 224.0.0.1
- Ưu điểm:
 - Đơn giản
 - Không đòi hỏi cơ chế routing phức tạp tại các router
- Nhược điểm:
 - Router cần duy trì cơ chế kiểm soát tránh lặp gói tin theo chu trình → tốn tài nguyên trong router
 - Chỉ phù hợp trong phạm vi liên mạng nhỏ (AS), không thể áp dụng cho mạng Internet

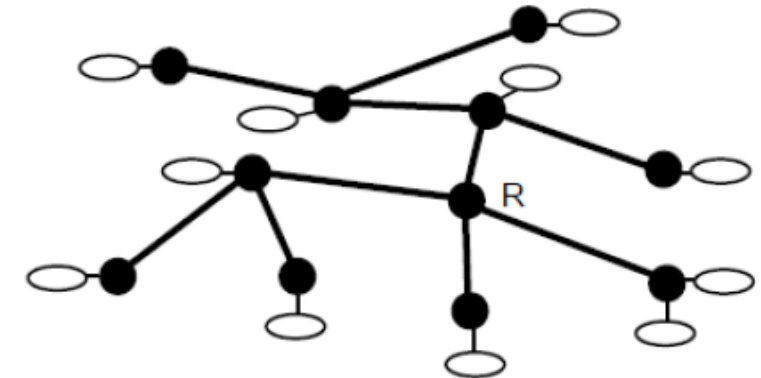


Thuật toán cây khung (Spanning Tree)

- Cải tiến từ thuật toán flooding, tối ưu tài nguyên router khi xử lý tránh lặp chu trình
- Chọn một tập con trong đồ thị kết nối mạng đầy đủ mà phủ hết các nút nhưng không có chu trình
→ cây khung (Spanning Tree)
- Áp dụng thuật toán flooding trên cây khung thay vì trên toàn bộ kết nối mạng: router flood gói tin multicast theo các giao diện kết nối mạng **mà có tham gia vào cây khung**
- Hạn chế: lưu lượng mạng bị dồn vào các liên kết trên cây khung, gây tắc nghẽn trong khi các liên kết khác không được sử dụng

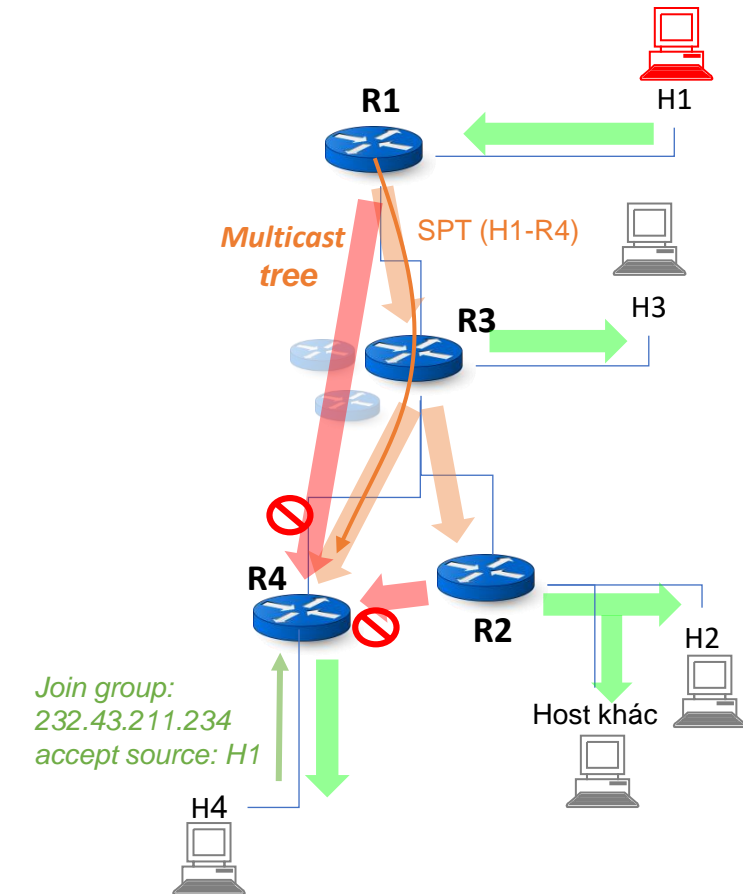


Xây dựng cây khung từ đồ thị kết nối mạng đầy đủ



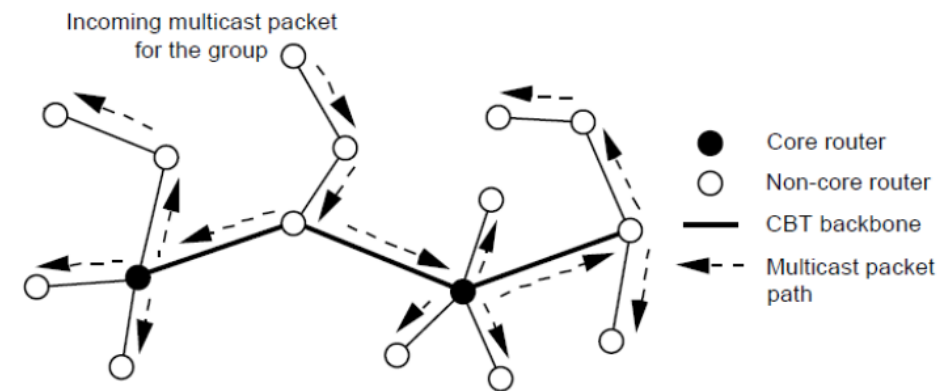
Thuật toán quảng bá ngược (RPB)

- Cải tiến hạn chế lưu lượng dồn vào các nhánh của cây khung trong thuật toán spanning tree → xây dựng cây khung theo nhóm (source, multicast group)
- Thuật toán Reverse Path Broadcasting (RPB):
 - Với mỗi cặp (source, multicast group), router xác định liên kết mạng nhận gói tin (với router kề trước trong “forward path”) và xác định đường đi “tối ưu” từ nó quay trở lại nguồn phát (SPT)
 - Nếu liên kết mạng nhận tin thuộc SPT → gắn “cành” vào cây khung
 - Nếu liên kết mạng nhận tin không thuộc SPT → bỏ qua
- Triển khai RPB:
 - Duy trì thông tin (các) cây khung trên tất cả router → Multicast Extensions to OSPF (RFC1584)
 - Không trì thông tin (các) cây khung trên tất cả router:
 - Multicast router chỉ biết nút cha và các nút con trực tiếp của nó trên cây khung
 - R4 lần ngược lại SPT để thiết lập các liên kết cha-con cho đến khi gặp router đầu tiên thuộc cây khung (reverse path!!!) là R3 thì gắn “cành” vào R3
 - Distance Vector Multicast Routing Protocol (FRC1075)



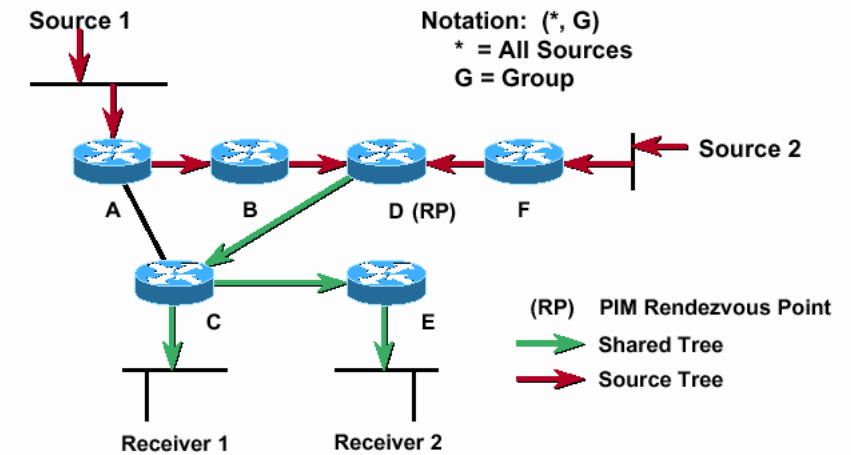
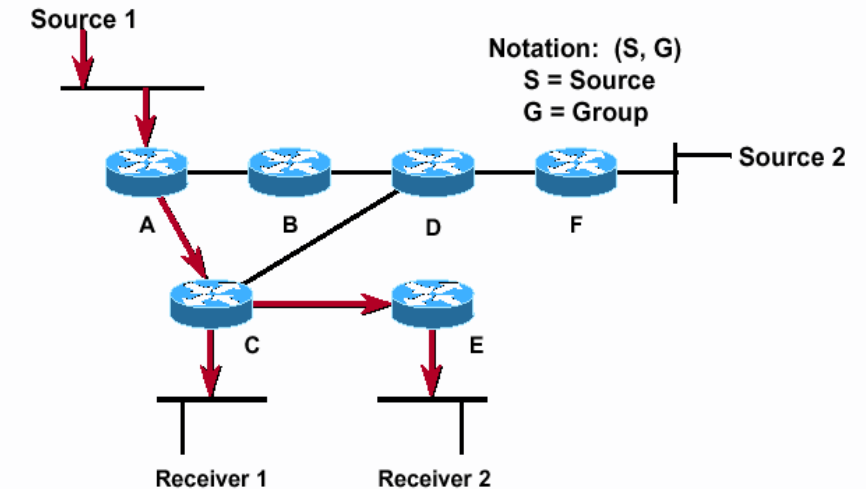
Thuật toán Core-Based Trees (CBT)

- RPB hoạt động dựa trên yếu tố chính là SPT đến nguồn phát. Áp dụng vào bài toán mạng có kích thước lớn cộng với việc mỗi router không lưu trữ hình ảnh đầy đủ của kết nối mạng → vấn đề hiệu năng khi xác định SPT
- Giải pháp của CBT (RFC2210):
 - Giải quyết hiệu năng SPT bằng cách thiết lập sẵn một cây lõi (core tree) là một tập con các router của toàn bộ hệ thống
 - Điều kiện tiên quyết là các cây khung cho mỗi nhóm multicast phải kết nối vào cây lõi này (các gói tin multicast của tất cả các nhóm sẽ được chuyển trên cây lõi – giống như mạng xương sống của một hệ thống mạng)
 - Tạo cây multicast: khi một host muốn tham gia vào một nhóm, thay vì tìm SPT đến source thì chỉ cần tìm SPT đến một trong các nút thuộc cây lõi, sau đó nối SPT này vào cây lõi
 - Gửi multicast từ trong nhóm: source là thành viên của cây multicast → chuyển gói tin multicast theo địa chỉ nhóm, qua các router trên cây (chứa cây lõi)
 - Gửi multicast từ ngoài nhóm: source gửi unicast đến một thành viên của cây (nếu không biết thì gửi đến cây lõi) và phiên multicast sẽ được bắt đầu từ thành viên này
- Việc xác lập các core router để xây dựng cây lõi có thể được thực hiện thủ công, dựa trên thông số vật lý của các kết nối mạng cũng như các thông số thống kê của việc sử dụng mạng



Tổng kết các thuật toán multicast

- Thuật toán multicast dựa trên nguồn phát (source)
 - Dựa trên xác định SPT đến nguồn phát → tối ưu đường truyền từ trạm truyền đến trạm nhận
 - Router phải xác định SPT và duy trì các đường định tuyến (của multicast group) theo từng nguồn phát
→ Bài toán hiệu năng: $O(S * G)$
- Thuật toán multicast dựa trên cây chung (shared tree)
 - Một thông tin gốc được cấu hình trước và chia sẻ với tất cả nhóm multicast (CBT core tree là một ví dụ).
 - Thông tin gốc dùng chung này thường được gọi là Rendezvous Point (RP) – điểm hẹn gặp.
 - Ưu điểm là giảm tính phức tạp → cải thiện hiệu năng: $O(G)$
 - Nhược điểm là không tối ưu đường truyền giữa trạm truyền và trạm nhận.
 - Vấn đề xác định RP dựa trên nhiều yếu tố thủ công, không tự động và phụ thuộc vào tính toán của người quản trị.
 - → dựa vào các cài đặt cụ thể



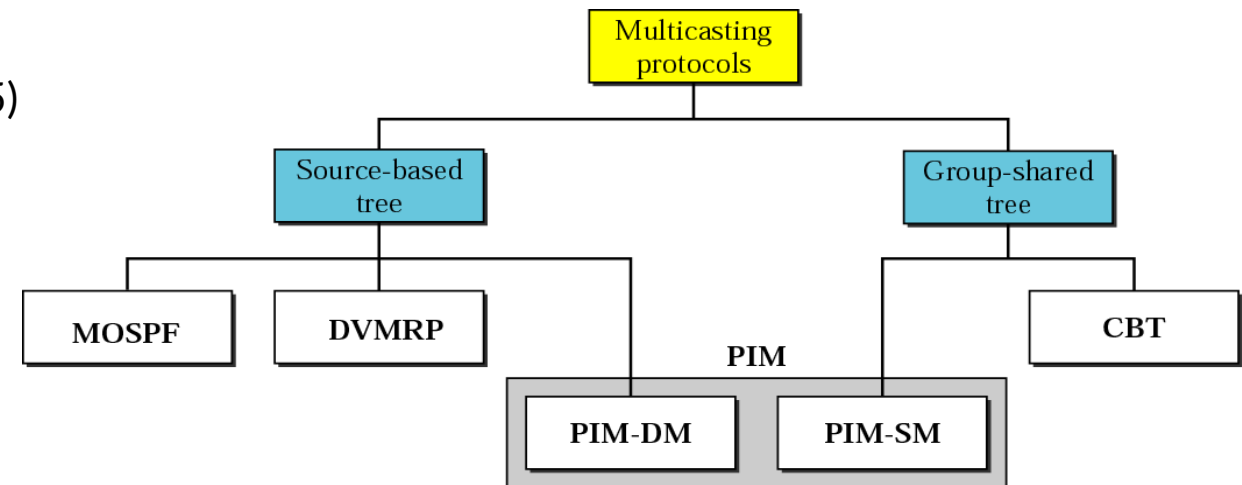


Multicast routing protocol

ONE LOVE. ONE FUTURE.

Tổng quan về multicast routing protocol

- Giống routing protocol, multicast routing protocol dùng để kết nối các router khi xây dựng cây multicast (pha 1 trong giao thức 2 pha multicast). Pha 2 (gửi gói tin) được thực hiện theo cơ chế forwarding dựa trên bảng routing (có cải tiến về nguồn phát & duplicate gói tin khi cần)
- Multicast routing protocol có thể được triển khai bằng cách cài đặt một thuật toán multicast vào một (unicast) routing protocol:
 - OSPF: Multicast Extensions to OSPF (RFC1584)
 - RIP: Distance Vector Multicast Routing (RFC1075)
- Multicast routing protocol không dựa trên unicast routing protocol:
 - Protocol-Independent Multicast (PIM)
 - Xử lý bài toán scalability:
 - Phân biệt 2 chế độ hoạt động dựa trên mức độ phân bố các trạm tham gia vào nhóm
 - PIM Dense Mode (PIM-DM): dày đặc
 - PIM Sparse Mode (PIM-SM): thưa thớt

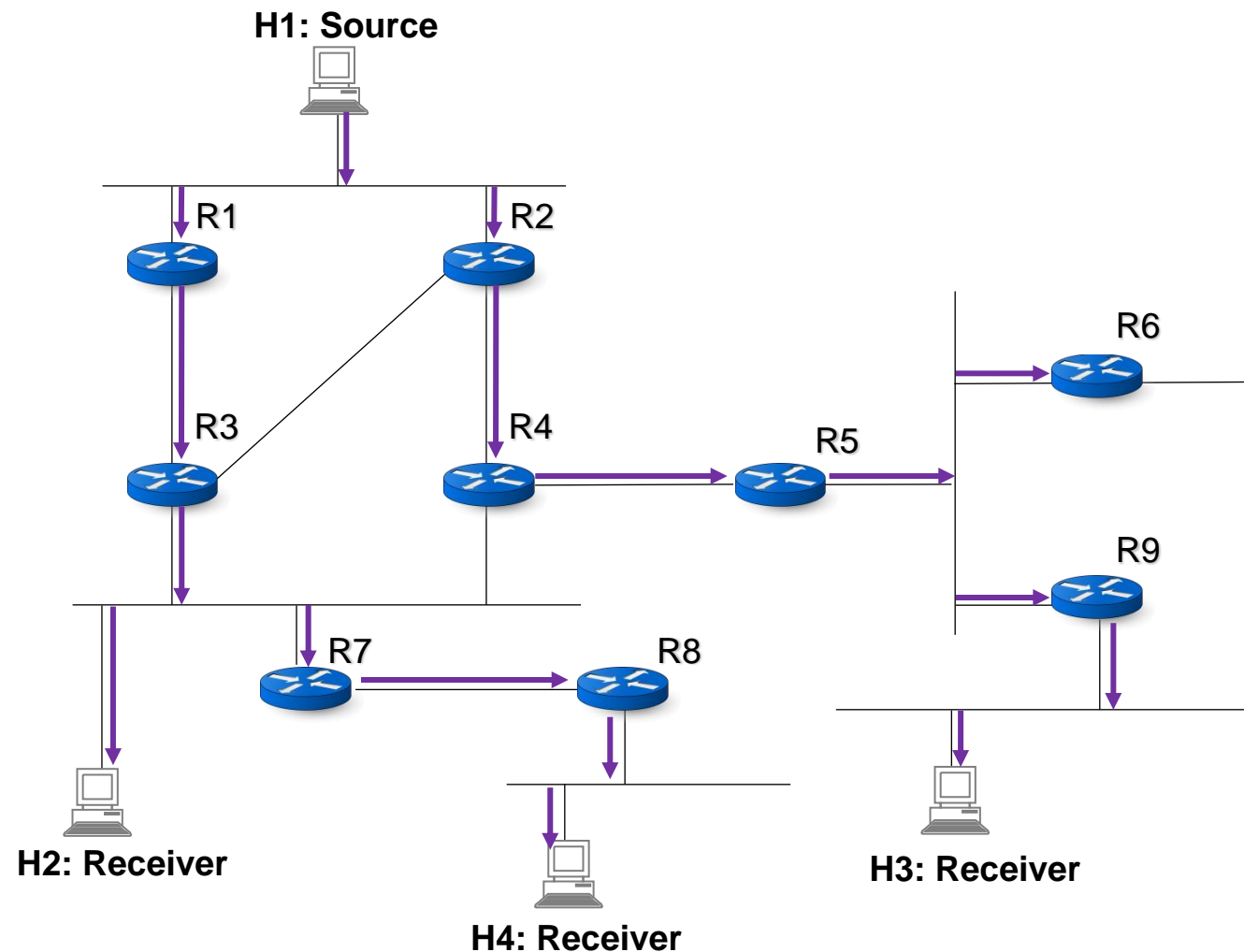


Protocol Independent Multicast (PIM)

- Overview
 - Independent of unicast routing protocols
 - Extract required routing information from any unicast routing protocol
 - Work across multiple AS with different unicast routing protocols
- Strategy
 - Flooding is inefficient over large sparse internet
 - Little opportunity for shared spanning trees
 - Focus on providing multiple shortest path unicast routes
- Two operation modes
 - Dense Mode (PIM DM): for intra-AS
 - Sparse Mode (PIM SM): Inter-AS multicast routing
- PIM SSM (source-specific) is a subset of PIM SM

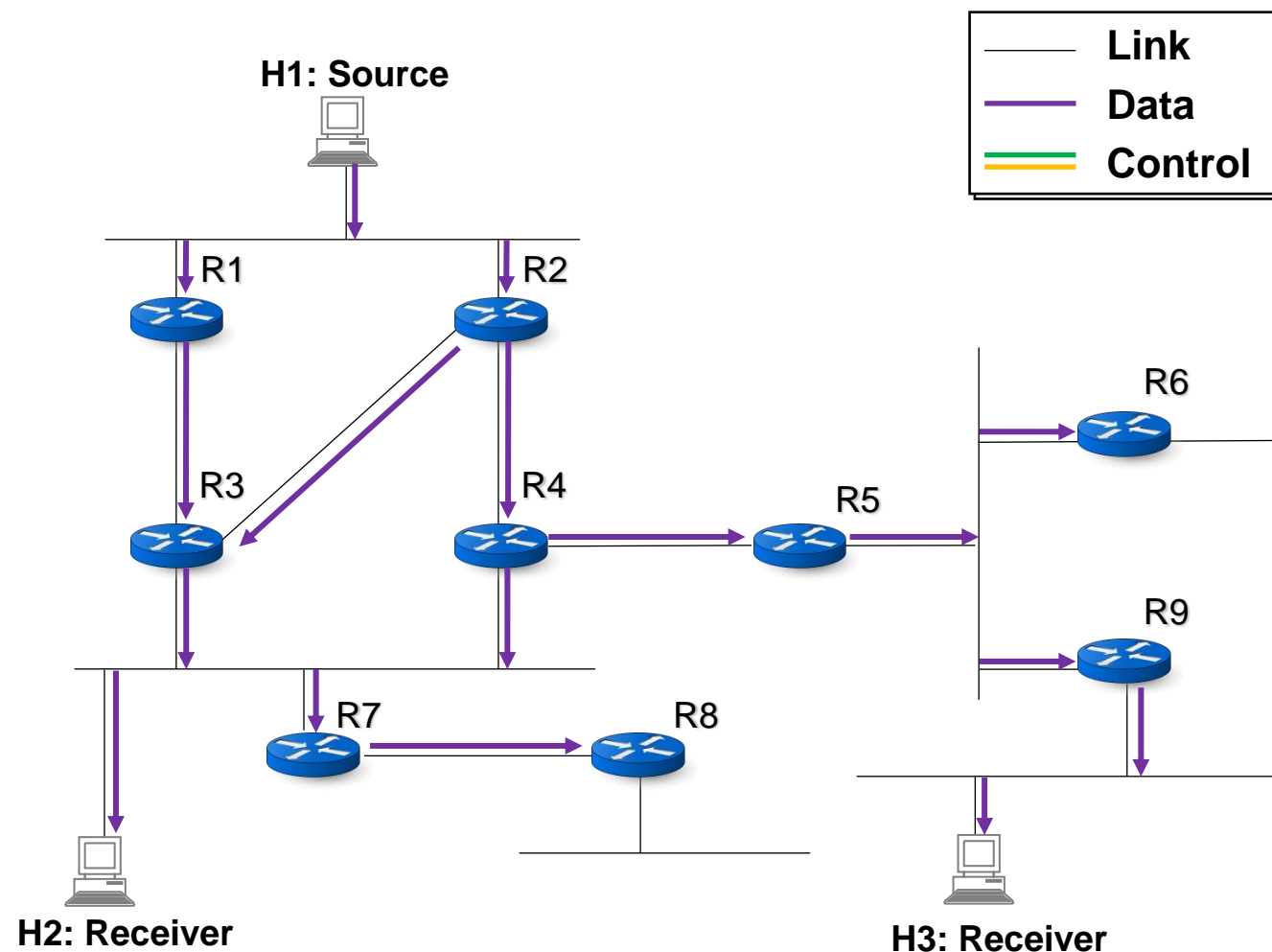
PIM Dense Mode

- PIM DM hoạt động hiệu quả trong môi trường dày đặc các trạm nghe & router
- Sử dụng thuật toán tràn (flooding) kết hợp tỉa cành (prun)
- Dựng cây multicast:
 - Áp dụng thuật toán tràn (flooding), cây multicast ban đầu là toàn bộ kết nối mạng
 - Tỉa cành upstream: router áp dụng thuật toán RPF và tỉa đi những cành không thuộc RPF
 - Tỉa cành downstream: cây con (cành) không có trạm nghe (host và router) thì được tỉa đi
 - Tỉa nút cha thừa: nhiều router cùng chuyển tiếp nhóm đến cùng một local link downstream → lựa chọn 1 router bằng thủ tục Assert và tỉa đi các cành downstream của router khác
 - Thêm cành mới: áp dụng lần lượt RPF, trạm nghe mới (host & router) join vào nút cha giống xử lý trong IGMP



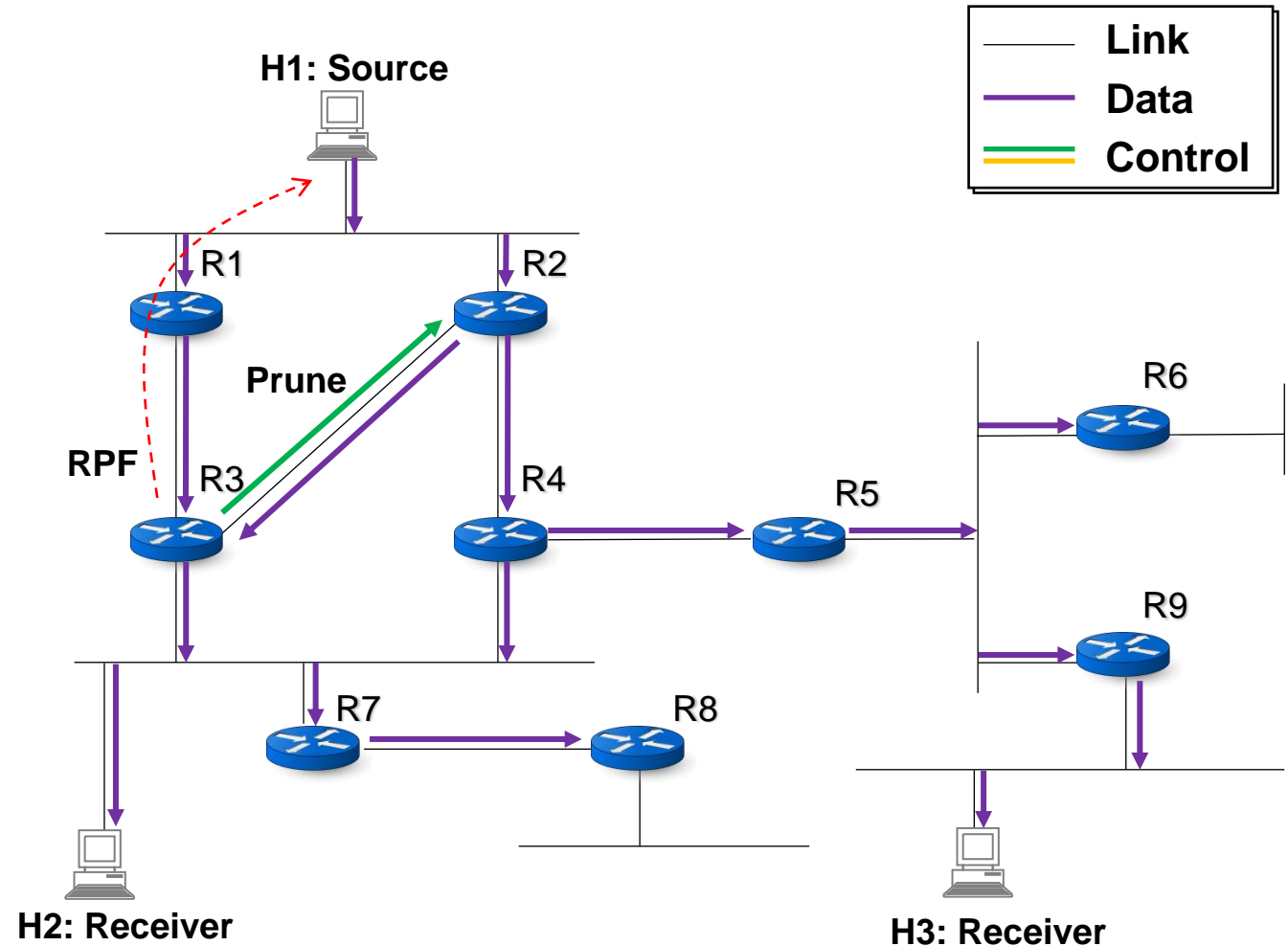
PIM DM Demo

- Nguồn phát hoạt động
- Data được chuyển tiếp theo thuật toán tràn (flooding)
- Mỗi router nhận data liền chuyển tiếp đến tất cả các kết nối với router láng giềng, trừ kết nối đang nhận data
- Router kiểm tra các trạm nghe đã đăng ký trên các kết nối LAN để quyết định có flood data hay không



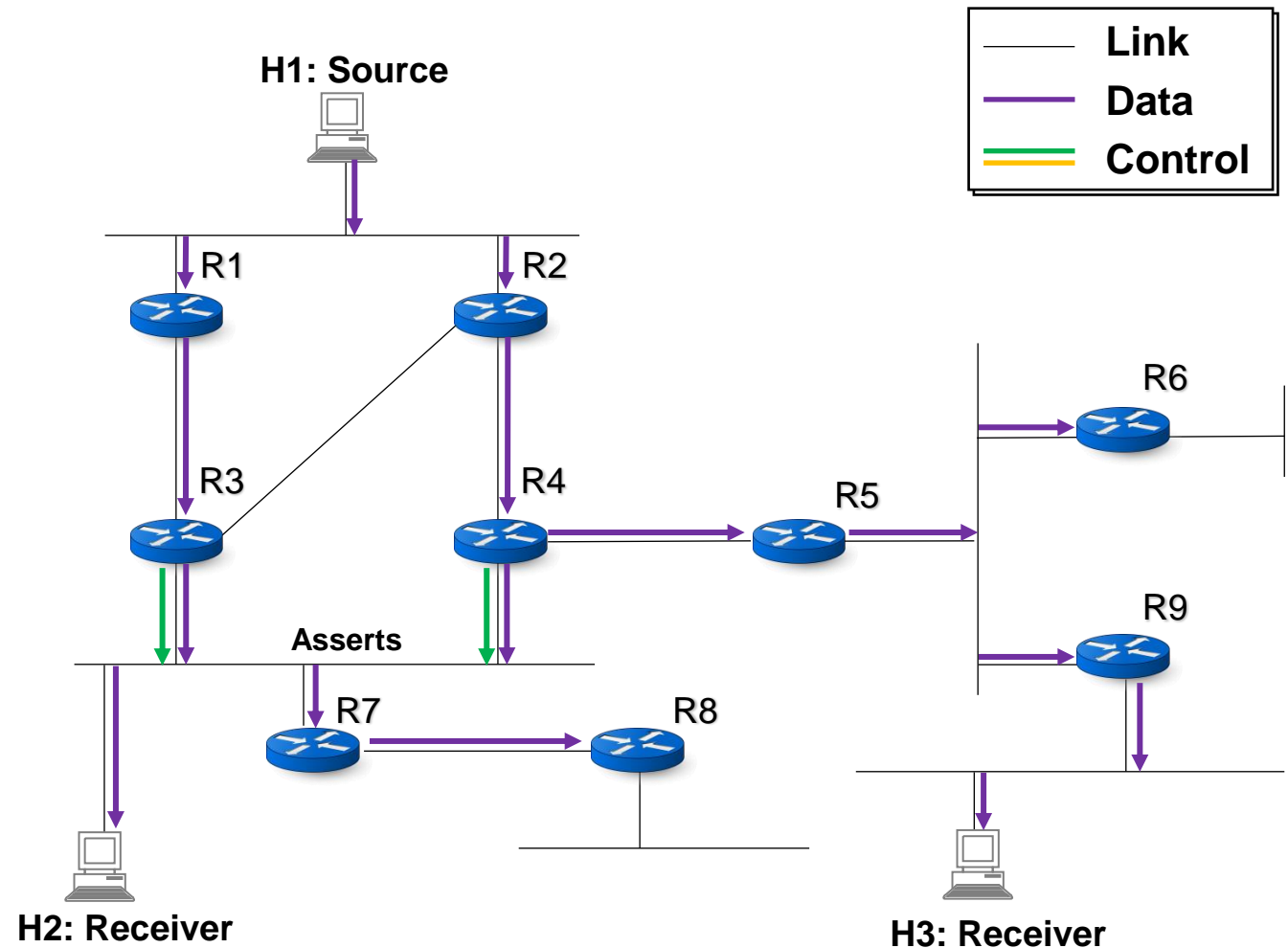
PIM DM Demo (2)

- Tỉa cành. Router R3 nhận data từ 2 nguồn (R1 & R2) → kiểm tra RPF về source
- R3 xác định R2 không nằm trong RPF → yêu cầu R2 dừng chuyển tiếp gói tin cho mình (prun)
- R2 nhận yêu cầu prun từ R3 và dừng phát tin trên link kết nối với R3



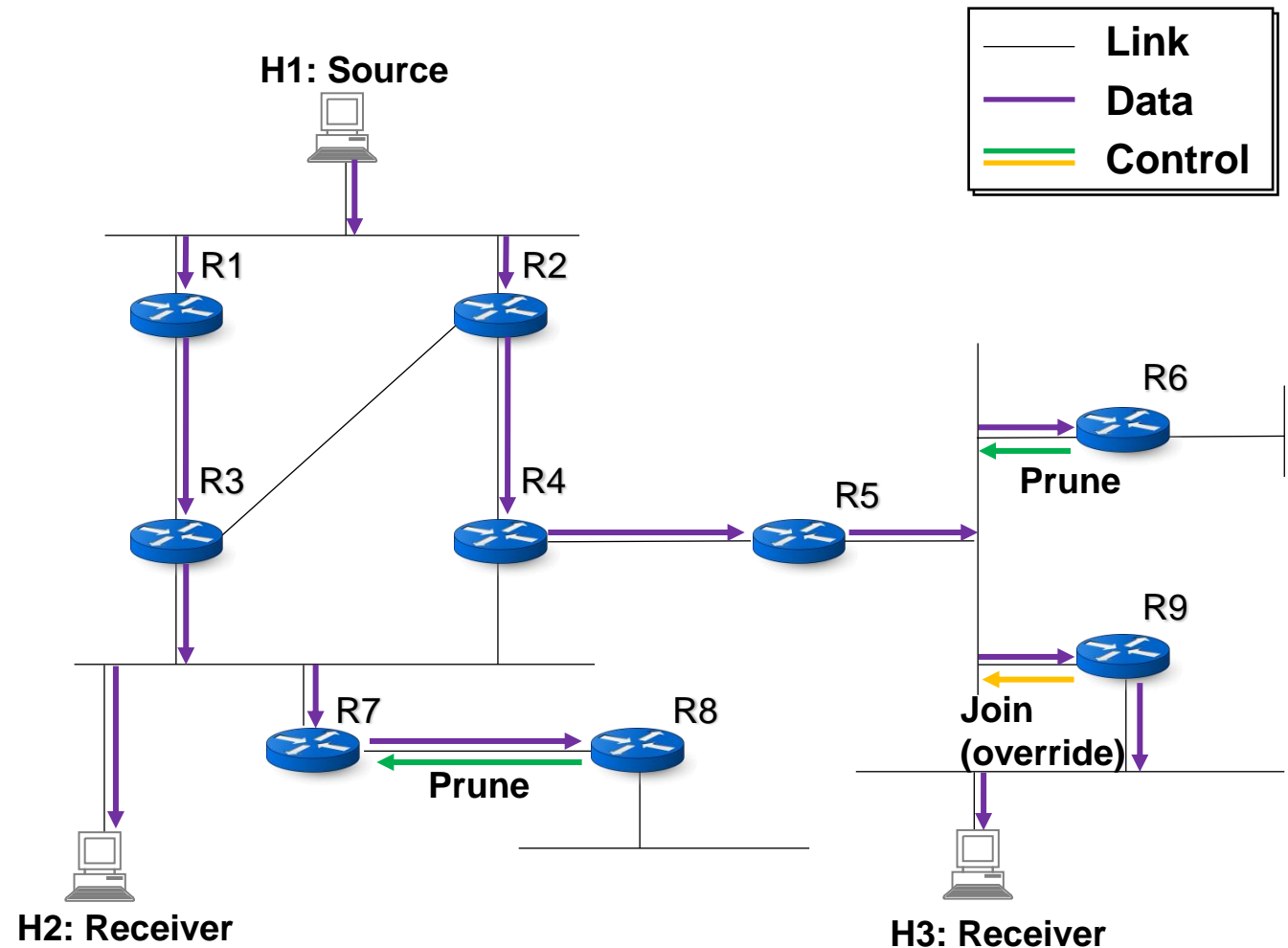
PIM DM Demo (3)

- Tỉa cành bằng Assert. Cả router R3 và R4 đều chuyển tiếp gói tin vào local link (LAN) → cần chọn một.
- Sử dụng Assert để so sánh → R3 được chọn
- R4 ngừng chuyển multicast data trên link đã được ủy quyền cho R3



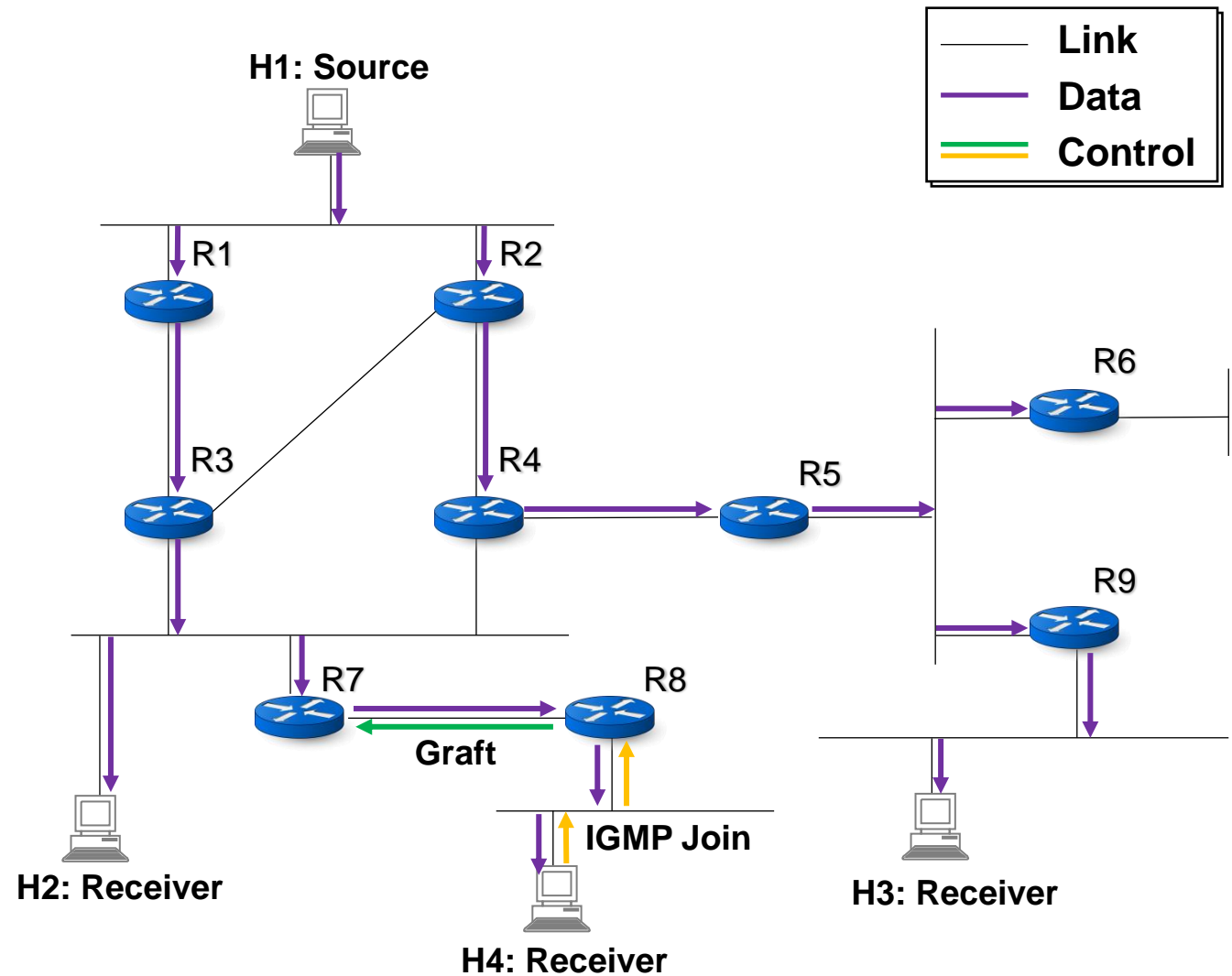
PIM DM Demo (4)

- Router R6 & R8 xác định không có trạm nhận (downstream) để forward gói tin → yêu cầu dừng chuyển tiếp gói tin cho mình
- R7 nhận được yêu cầu prun từ R8, nhưng xác định có trạm nghe (H2) đang nghe trên cùng local link với mình (do R3 phát) → tiếp tục giữ link nhận data từ R3
- R5 nhận yêu cầu prun từ R6 nhưng cũng nhận yêu cầu join từ H3 → bỏ qua



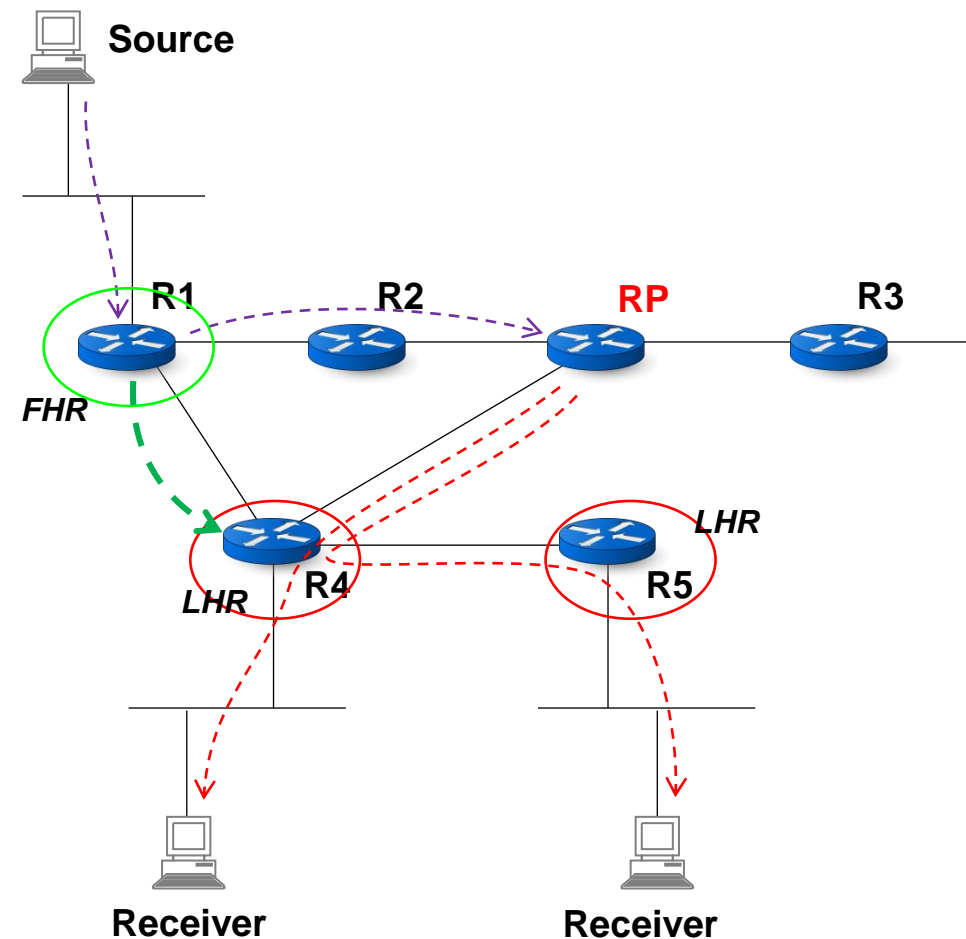
PIM DM Demo (5)

- Xuất hiện trạm nghe mới (H4).
- IGMP Join được gửi từ trạm nghe mới đến R8, R8 yêu cầu join vào cây bằng thông điệp Graft đến R7
- R7 xác định vẫn đang nhận được multicast data từ R3 → xử lý Graft từ R8 và hoàn thành
- R8 nhận multicast data từ R7 và chuyển tiếp sang link kết nối với trạm nghe mới H4



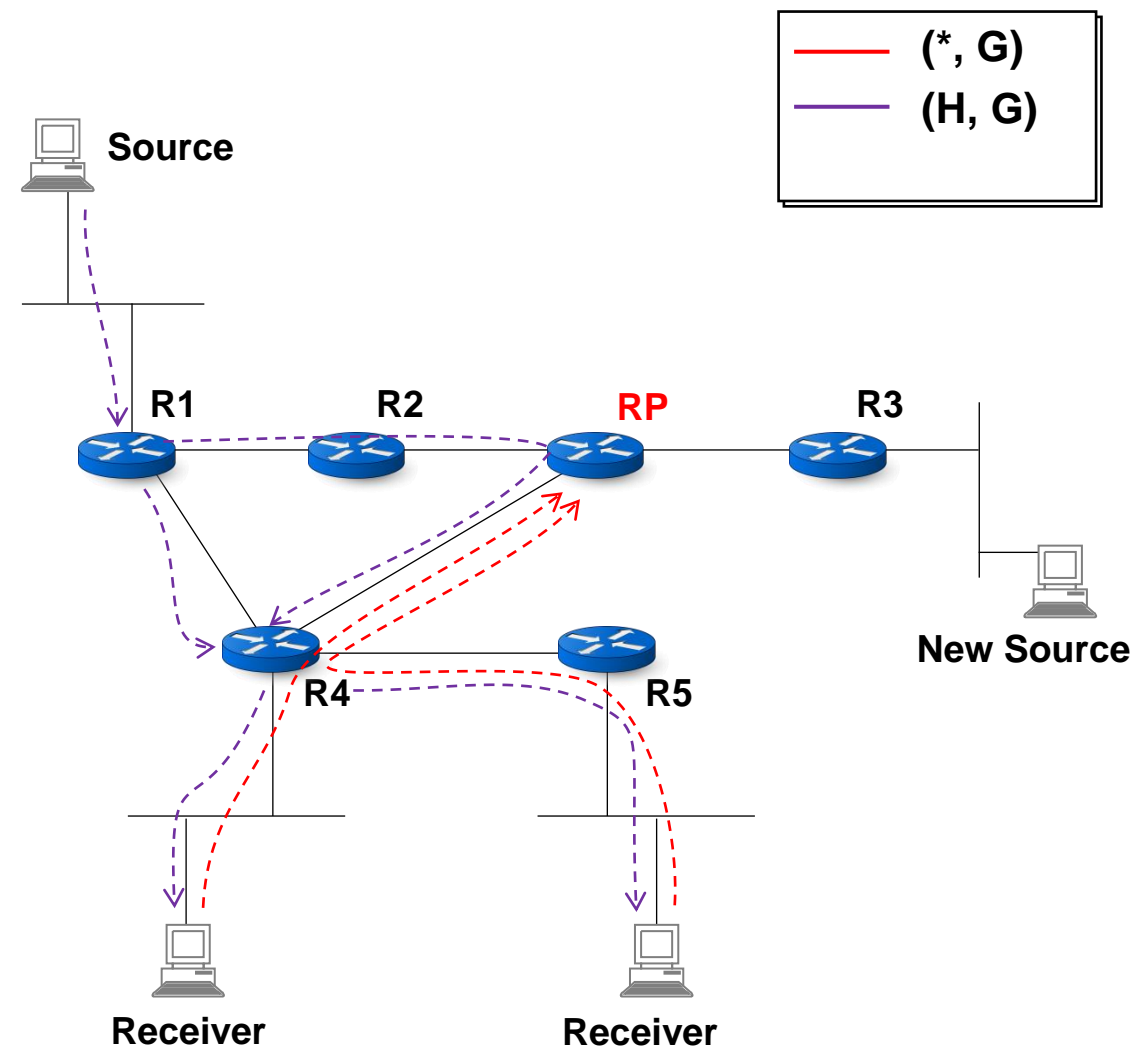
PIM Sparse Mode

- Hoạt động theo giải thuật CBT (RP)
- Hỗ trợ source specific như một trường hợp đặc biệt của RP
- Phù hợp với multicast trên diện rộng (multi AS), với các trạm nhận bố trí thưa thớt
- Phương pháp sử dụng Rendezvous Point (RP)
 - Trạm truyền và các trạm nhận sử dụng “rendezvous” như một điểm hẹn để tìm thấy nhau
 - Cây multicast được xây dựng bằng cách nối 2 cây: shared tree (gốc là RP) và source tree (gốc là nguồn phát)
- Bổ sung khâu tối ưu cây
 - Nối 2 cây (shared tree và source tree) tạo được cây multicast (hoạt động được) nhưng không hiệu quả
 - Tối ưu cây dựa vào vai trò các router FHR và LHR



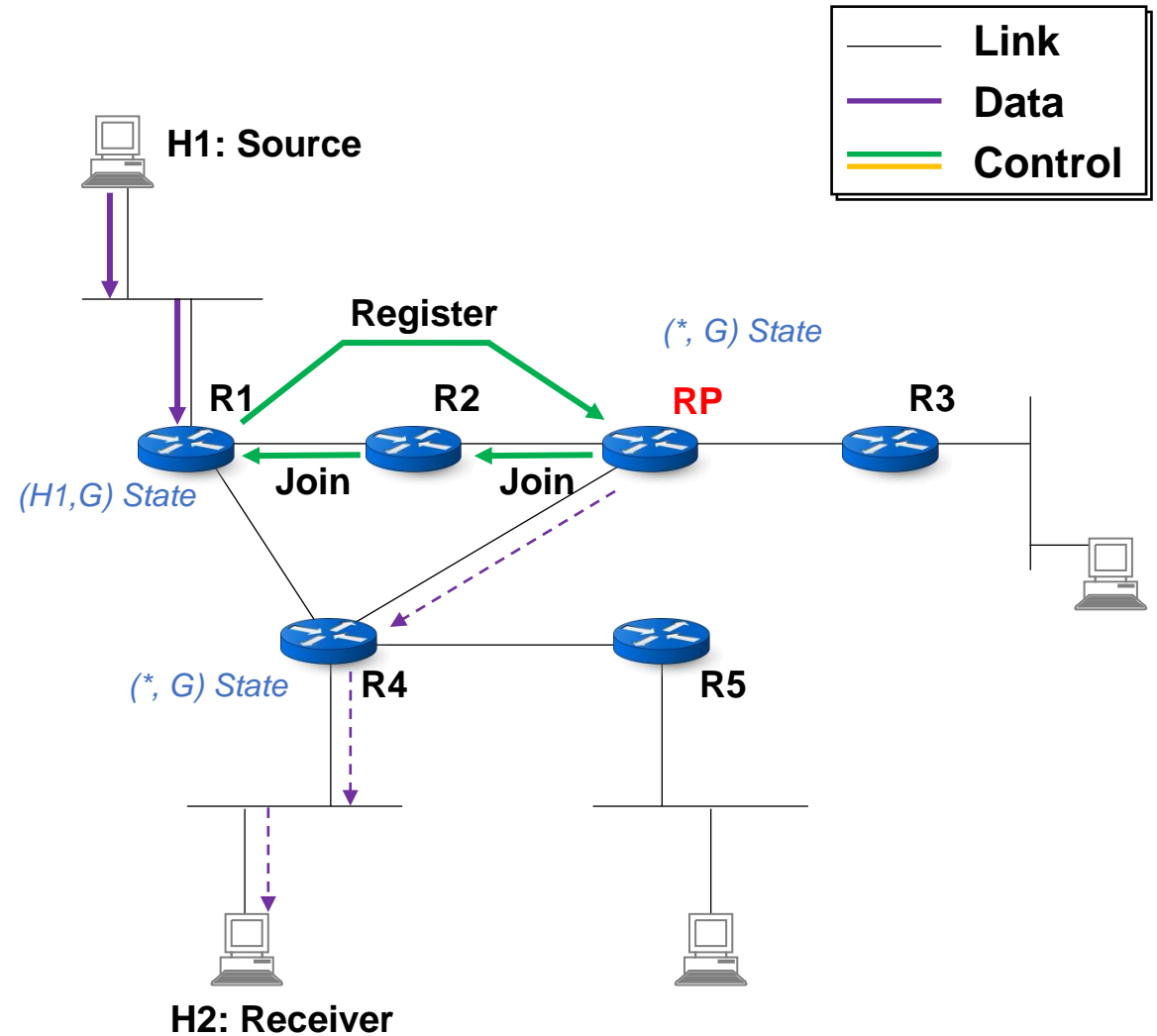
Vai trò của Rendezvous Point (RP)

- Hệ thống mạng kết nối bằng các router trong đó xác định trước router có vai trò RP
- Các trạm phát & nhận tham gia vào phiên multicast ngẫu nhiên (không phải source specific)
- Khi nguồn phát chưa xác định, các router sử dụng RP để dựng cây multicast
- Khi nguồn phát xuất hiện, router đăng ký (register) với RP và nhận cây từ RP (bằng cách nối cây gốc RP vào nguồn phát)
- Tối ưu cây multicast bằng RPF (do lúc này nguồn phát đã được xác định)
- Thêm nguồn phát bổ sung?



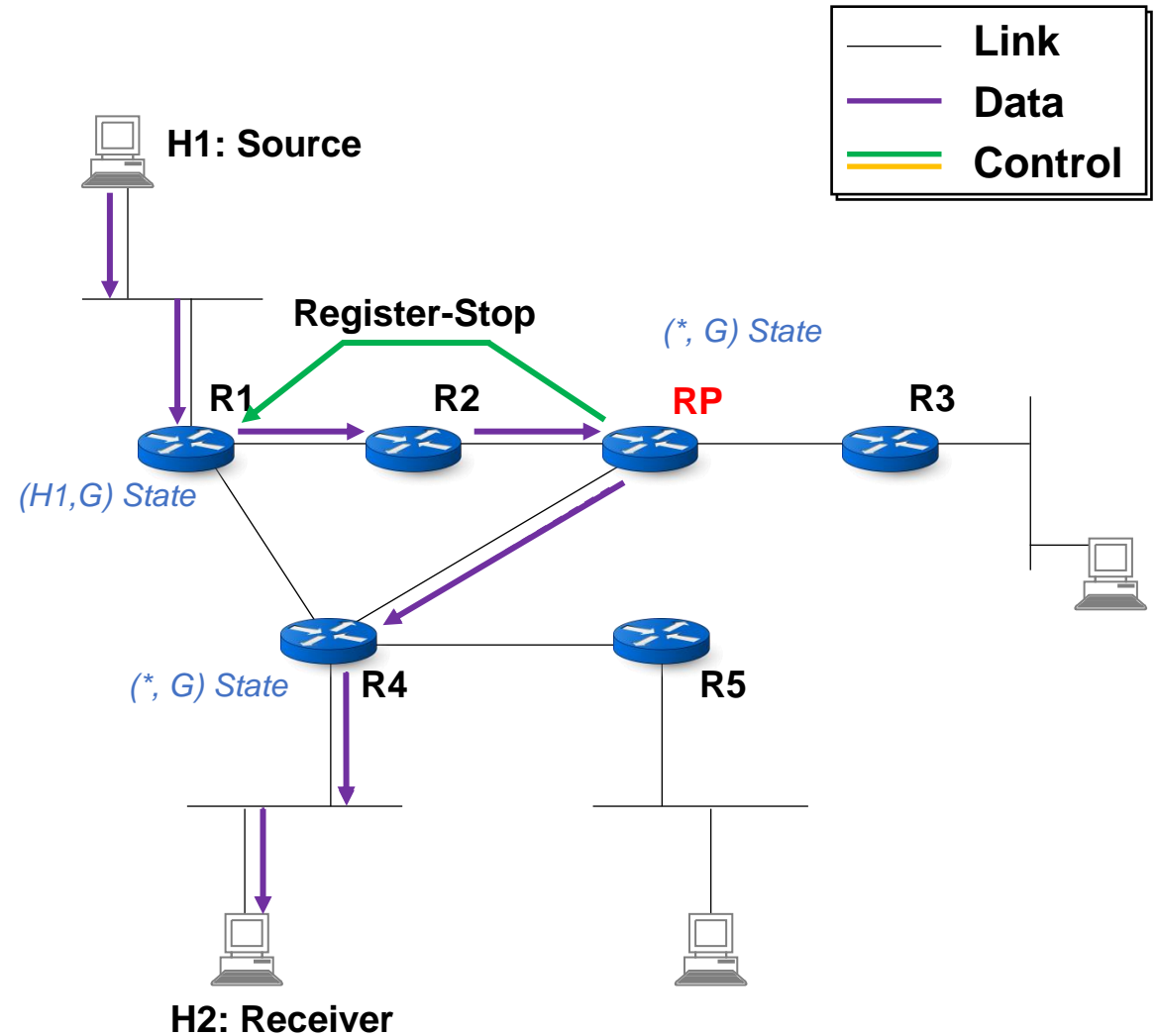
PIM SM Demo (2)

- H1 bắt đầu phát tin vào nhóm G
- FHR (R1) nhận được multicast data phát từ H1, nó xác định chưa có nhóm multicast này → cần dựng cây source tree đến RP
- FHR khởi tạo thông điệp PIM-Register đến RP
- RP áp dụng RPF về R1 và lần lượt gửi PIM Join từ down stream router đến up stream router trong RPF
- Cành được lần lượt gắn vào source tree từ RP đến FHR (R1) qua router trung gian (là R2)



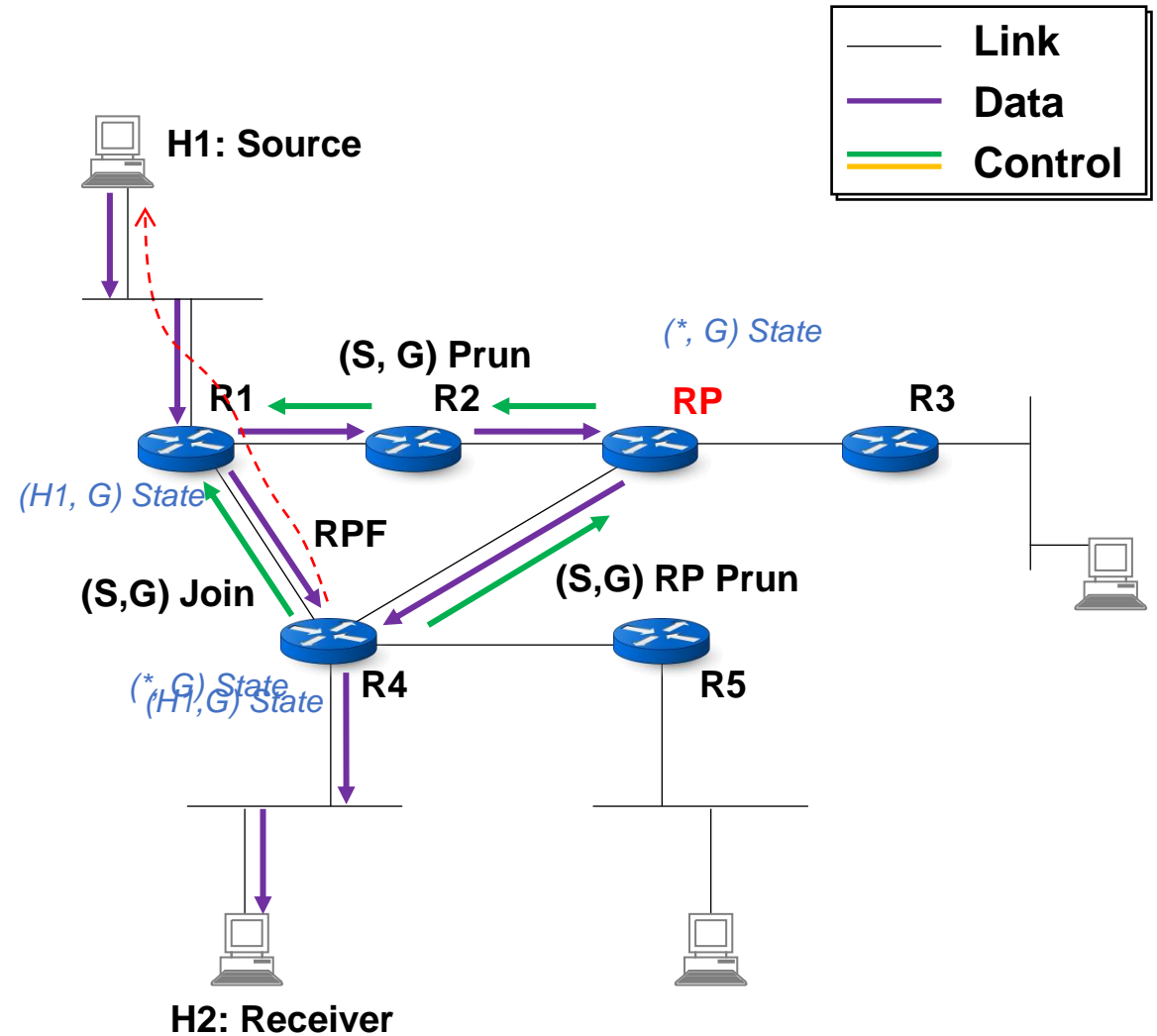
PIM SM Demo (3)

- RP gửi PIM Register-Stop về R1 để thông báo quá trình register đã hoàn thành
- Cây multicast nguồn phát xác định (H1) được xây dựng bằng cách nối source tree (từ H1 đến RP) với shared tree (từ RP đến các trạm thu)
- Phiên truyền thông multicast từ nguồn phát đến các nguồn thu đã có thể được tiến hành
- Cây multicast chưa được tối ưu
- RP phải hỗ trợ tất cả các phiên multicast → quá tải



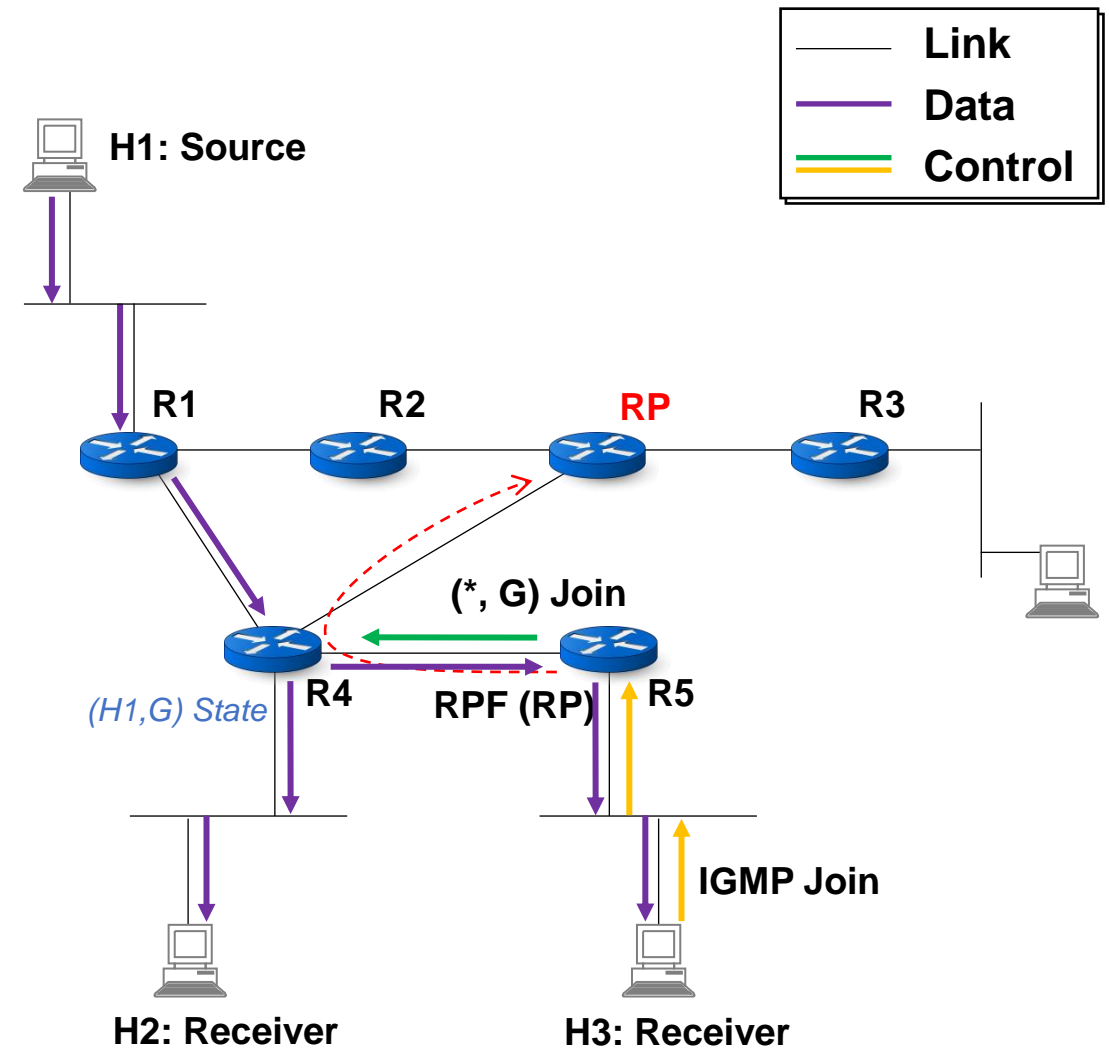
PIM SM Demo (4)

- Thủ tục PIM-Register kết thúc, các LHR router sử dụng RPF về nguồn phát xác định (H1) để “chỉnh hình” cây multicast bằng cách lần lượt join vào “up stream router”. Quá trình kết thúc khi gặp “up stream router” thuộc cây multicast
- R4 gửi PIM Join (S,G) đến R1
- R1 xử lý PIM Join từ R4, thêm luật chuyển tiếp multicast cho nhóm (S, G) sang downstream mới là R4
- Ngay khi nhận được multicast data từ R1, R4 gửi PIM RP Prun đến RP để hủy kết nối này trong nhóm
- RP xóa trạng thái (S,G) và gửi Prun lên upstream
- Các router cập nhật luật chuyển tiếp multicast cho nhóm (S,G) → cây multicast được thiết lập với vai trò của RP được hủy bỏ



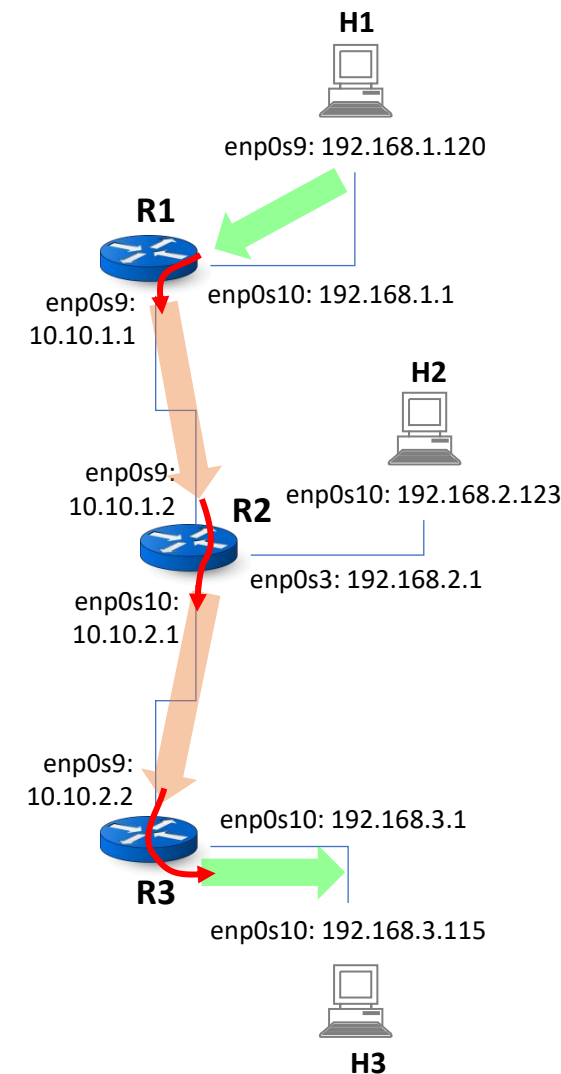
PIM SM Demo (5)

- H3 xin tham gia vào cây (*, G)
- IGMP Join được chuyển từ H3 đến R5
- R5 xác định RPF về RP (R5 chưa biết nguồn phát nên dùng RP để thực hiện RPF) và gửi PIM Join đến nút cha trong RPF của mình.
- Quá trình Join sẽ kết thúc khi đến được RP hoặc một nút trong cây (*,G), trong trường hợp này là R4
- R4 tạo luật multicast để forward gói tin từ upstream (R1) sang downstream
- R5 nhận được multicast data trên upstream (R4) và chuyển tiếp đến downstream cho H3
- R5 xác định nguồn phát là H1, thực hiện RPF để tối ưu cây multicast (trường hợp này cây đã tối ưu)



Thực hành PIM SM

- Thiết lập hệ thống mạng với 3 router và 3 host
- Thiết lập RP của cả hệ thống là R2 (192.168.2.1)
- Chạy iperf trên H3 để join vào nhóm multicast với nguồn phát không xác định
- Kiểm tra cấu hình multicast trên router R3 & R2:
 - Tự động thiết lập theo gói tin IGMP Report gửi đến từ H3
 - Tự động join R3 (enp0s9) và R2 (enp0s10) vào cây multicast với gốc là RP
- Chạy iperf trên H1 để phát gói tin vào nhóm, kiểm tra H3 nhận được gói tin phát trong nhóm
- Kiểm tra PIM state trên các router thấy có sự cập nhật phù hợp với các thuật toán PIM



Một số khái niệm về “Sparse”

- Sparse group:
 - Number of networks/domains with group members present significantly small than number of networks/domains in internet
 - Internet spanned by group not sufficiently resource rich to ignore overhead of current multicast schemes
- Group Destination Router
 - Has local group members
 - Router becomes destination router for given group when at least one host joins group
 - Using IGMP or similar
- Group source router
 - Attaches to network with at least one host transmitting on multicast address via that router

PIM Sparse Mode

- For a group, one router designated rendezvous point (RP)
- Group destination router sends join message towards RP requesting its members be added to group
 - Use unicast shortest path route to send
 - Reverse path becomes part of distribution tree for this RP to listeners in this group
- Node sending to group sends towards RP using shortest path unicast route
- Destination router may replace group-shared tree with shortest path tree to any source
 - By sending a join back to source router along unicast shortest path
- Selection of RP dynamic
 - Not critical

PIM Sparse Mode (2)

- Explicit join model
 - Receivers join to the Rendezvous Point (RP)
 - Senders register with the RP
 - Data flows down the shared tree and goes only to places that need the data from the sources
 - Last hop routers can join source tree if the data rate warrants by sending joins to the source
- RPF check for the shared tree uses the RP
- RPF check for the source tree uses the source
- Only one RP is chosen for a particular group
- RP statically configured or dynamically learned (Auto-RP, PIM v2 candidate RP advertisements)
- Data forwarded based on the source state (S, G) if it exists, otherwise use the shared state (*, G)

Vai trò của FHR và LHR

- First Hop Router (FHR)
 - R1
- Last Hop Router (LHR)
 - R4 & R5 (cho các trạm nghe khác nhau)

