

第 23 章 组播

随着网络中的视频等应用越来越多，组播也越来越时髦。路由器转发组播流的方式和转发单播流有很大的差别，发送数据的源不知道接收者在何处。组播也有很多路由协议，它们为组播流建立一棵无环路、无重复流量的转发树。本章将介绍常用的、简单的两个协议：PIM dense 和 PIM sparse。

23.1 组播简介

当有 10000 个用户通过网络看电视时，如果分别为每个用户传输一路流量，不仅服务器受不了，网络也承载不了。组播（也称多播）则像电视一样，传输一份数据，需要接收数据的计算机加入到这个组就行了。虽然组播配置并不复杂，但理论知识则相当复杂。组播采用 224.0.0.0—239.255.255 的地址，不同地址就是不同的组，一个组可能有多个源，而需要接收数据的设备是这个组的成员。

路由器转发组播流的方式和转发单播流有很大的差别，发送数据的组播源不知道接收者在何处。保证接收者能收到数据，并且数据不会在不必要的网络上存在是很重要的事情。路由器必须为组播确定出一条转发路径。路由器采用反向路径转发（RPF），即对每个接收到的组播进行源地地址测试，如果数据是从到达源的接口上接收到的，就往下游路由器转发。

为了确定是否应该往某个网络转发组播流，路由器使用 IGMP（Internet Group Manage Protocol）和主机之间通信，确定这个网络是否有某个组的成员。IGMP 有 V1、V2、V3，目前 IOS 采用 V2。IGMP 有各种消息，例如：Membership Query 消息、Membership Report 消息、Leave Group 消息、General Query 消息、Group-Specific Query 消息等。

PIM（Protocol Independent Multicast）是一个组播路由协议，独立于协议的意思是该组播协议不关心单播路由是通过 RIP 还是 OSPF 或者其他方式学习到的。PIM 有两种模式：PIM Dense 和 PIM Sparse，后者通常和 Dense 结合使用，成为 PIM Sparse-Dense。

Dense 通常用于组成员比较密集的网络中。在 Dense 模式中，当有组播源出现时，路由器假设所有的网络都有组成员，构建了一棵从源开始的转发树，全部网络就都有了组播流量。然而各个路由器会紧接着查询自己的接口上是否有这个组的成员存在，如果没有成员，将停止往这个接口转发组播流。如果路由器上一个成员都没有，它将向上游路由器发送消息，把它从转发树上修剪掉。一级一级地往上发送消息，最终多播路由协议将构建一棵以源为根、不会有多余组播流量存在的转发树。如果有新的成员加入，路由器将一级一级往上发送消息，建立转发路径。在 Dense 模式中，会为不同的源建立不同的树，这样树的数量可能会很多。

Sparse 则通常用于组成员比较稀疏的网络中。在 Sparse 模式中，路由器假设所有的网络都没有组成员，除非有主机明确表示加入该组。转发树的建立从终端的叶节点组成员开始，向后扩展到中心的根节点上。和 Dense 模式不同，Sparse 是基于共享树的。也就是说某个组的流量是先发送到中心节点上（称为 RP），然后在从 RP 转发到各个组成员上的。组成员加入到这个组时，本地路由器向 RP 发送成员报告，沿途的路由器将树枝加入到共享树中。组成员从组中退出时，才执行修剪。这样好处是树的数量少，然而可能造成一些组播数据绕了一圈才到达主机。因此，实际上默认时，当路由器发现不是从到达源的最佳路径的接口上收到组播流，会自动切换到基于源的树。Sparse 模式中，我们可以手工为某个组指定 RP，也可以让路由器自动选举。要注意的是，路由器自动选举 RP 时发送的是组播流量，由于 RP 还没出现，所以只能使用 Dense 模式传输这些组播流量，所以 Sparse 通常和 Dense 结合使用。

而对于交换机，也不能说从一个接口收到组播，就洪到全部接口。然而交换机并没有 IGMP 协议和主机通信，交换机采用 2 种方案，一种是 IGMP Snopping，另一种是 CGMP。IGMP Snopping 中，交换机监听主机和路由器之间的 IGMP 消息，从而确定出哪个接口上有什么组的成员存在，组播流则从这些特定的接口发送出去。在 2 层交换机或者低端的三层交换机上，IGMP Snopping 基本是默认采用的。而 CGMP 协议则是交换机用来和路由器进行通信，从路由器获得组的成员名单，从而确定哪些接口应该转发哪些组的流量。

23.2 实验 1: PIM Dense

1. 实验目的

通过本实验，读者可以掌握如下技能：

- (1) 理解 PIM Dense 的工作原理
- (2) 掌握 PIM Dense 的配置

2. 实验拓扑

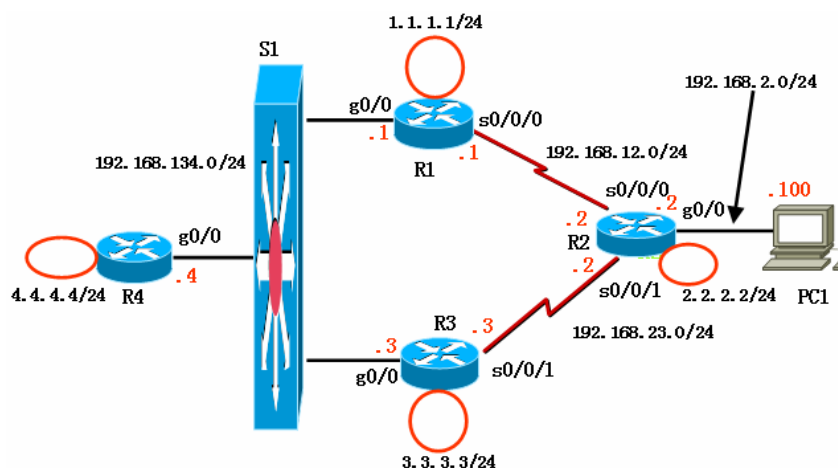


图 23-1 实验 1、实验 2 拓扑图

拓扑中 R2 的 g0/0 接口以及 PC1 也是接在交换机 S1 上的，我们把这两个接口单独划分在一个 VLAN 中。图 23-1 没有画出它们和 S1 的连接情况。

3. 实验步骤

- (1) 步骤 1: 配置 IP 地址、把 R2 的 g0/0 和 PC1 所接的交换机接口单独划在一个 VLAN

各路由器的 IP 如图 23-1 所示，在每个路由器上创建环回口 Loopback0，配置 IP 地址。把 R2 的 g0/0 所接的 f0/2 和 PC1 所接的 f0/5 接口单独划分一个 VLAN，如下：

```
S1(config)#vlan 2
S1(config)#int f0/2
S1(config-if)#switch mode access
S1(config-if)#switch access vlan 2
S1(config)#int f0/5
S1(config-if)#switch mode access
S1(config-if)#switch access vlan 2
```

(2) 步骤 2: 配置路由协议

```
R1(config)#router rip
R1(config-router)#network 1.0.0.0
R1(config-router)#network 192.168.12.0
R1(config-router)#network 192.168.134.0
```

```
R2(config)#router rip
R2(config-router)#network 192.168.12.0
R2(config-router)#network 192.168.23.0
R2(config-router)#network 192.168.2.0
R2(config-router)#network 2.0.0.0
```

```
R3(config)#router rip
R3(config-router)#network 192.168.23.0
R3(config-router)#network 192.168.134.0
R3(config-router)#network 3.0.0.0
```

```
R4(config)#router rip
R4(config-router)#network 192.168.134.0
R4(config-router)#network 4.0.0.0
```

(3) 步骤 3: 配置 PIM Dense

```
R1(config)#ip multicast-routing
//以上是启用组播路由功能
R1(config)#int loopback0
R1(config-if)#ip pim dense
//组播的配置相当简单，在接口上运行 pim dense 协议即可
R1(config)#int s0/0/0
R1(config-if)#ip pim dense
R1(config)#int g0/0
R1(config-if)#ip pim dense
```

```
R2(config)#ip multicast-routing
R2(config)#int loopback0
R2(config-if)#ip pim dense
R2(config)#int s0/0/0
R2(config-if)#ip pim dense
R2(config)#int s0/0/1
R2(config-if)#ip pim dense
R2(config)#int g0/0
R2(config-if)#ip pim dense
```

```
R3(config)#ip multicast-routing
R3(config)#int loopback0
R3(config-if)#ip pim dense
```

```
R3(config)#int s0/0/1
R3(config-if)#ip pim dense
R3(config)#int g0/0
R3(config-if)#ip pim dense
```

```
R4(config)#ip multicast-routing
R4(config)#int loopback0
R4(config-if)#ip pim dense
R4(config-if)#ip igmp join-group 237.0.0.1
//该接口加入到 237.0.0.1 组中，我们要利用 237.0.0.1 组做测试
R4(config)#int g0/0
R4(config-if)#ip pim dense
```

(4) 步骤 4: 检查 pim 邻居、测试组播路由

```
R1#show ip pim neighbor
```

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
S - State Refresh Capable

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
192.168.12.2	Serial0/0/0	06:57:59/00:01:27	v2	1 / S
192.168.134.3	GigabitEthernet0/0	06:25:46/00:01:29	v2	1 / S
192.168.134.4	GigabitEthernet0/0	06:25:46/00:01:24	v2	1 / DR S

//以上显示了 R1 上 pim 邻居

在 PC1 上配置 IP 地址，执行 “**ping -t 237.0.0.1**”，这时 PC1 实际上就是组 237.0.0.1 的组播源了。在各个路由器上检查多播路由表：

```
R2#show ip mroute
```

(此处省略)

```
(*, 237.0.0.1), 00:03:07/stopped, RP 0.0.0.0, flags: D
```

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

Serial0/0/1, Forward/Dense, 00:03:07/00:00:00

Serial0/0/0, Forward/Dense, 00:03:07/00:00:00

```
(192.168.2.100, 237.0.0.1), 00:03:07/00:02:52, flags: T
```

Incoming interface: GigabitEthernet0/0, RPF nbr 0.0.0.0

//以上表明 R2 是从 g0/0 接口接收到多播流的

Outgoing interface list:

Serial0/0/1, Forward/Dense, 00:03:08/00:00:00

Serial0/0/0, Prune/Dense, 00:01:09/00:01:50

//可以看到 R2 上只往 s0/0/1 接口转发多播流量；s0/0/0 接口不转发，处于被修剪状态。

```
(*, 224.0.1.40), 06:49:33/stopped, RP 0.0.0.0, flags: DCL
```

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

Serial0/0/1, Forward/Dense, 06:49:33/00:00:00

Serial0/0/0, Forward/Dense, 06:49:33/00:00:00

【技术要点】需要仔细观察组播树的情况。在各个路由器上使用“**show ip mroute**”命令，应该可以看到：组播数据从 R2 的 g0/0 接口接收到，从 s0/0/1 接口转发出去到达 R3 的 s0/0/1，s0/0/0 接口则不转发；R3 从 s0/0/1 接口收到组播，从 g0/0 接口转发出去到达 R4 的 g0/0；R4 从 g0/0 接口收到组播，从 loopback0 接口转发出去。

把 R1 上的 loopback0 接口也加入到组 237.0.0.1 中，重新在 R2 上检查多播路由表。如下：

```
R1(config)#int loopback0
```

```
R1(config-if)#ip igmp join-group 237.0.0.1
```

```
R2#show ip mroute
```

(此处省略)

```
(192.168.2.100, 237.0.0.1), 00:10:21/00:02:58, flags: T
```

```
Incoming interface: GigabitEthernet0/0, RPF nbr 0.0.0.0
```

Outgoing interface list:

```
Serial0/0/1, Forward/Dense, 00:10:22/00:00:00
```

```
Serial0/0/0, Forward/Dense, 00:00:03/00:00:00
```

//可以看到 R2 上的 s0/0/0 接口开始转发数据了，原因在于 R1 上有组成员了。可见 PIM Dense 会根据组成员的加入或者退出，动态地维护转发树。

(此处省略)

```
R1#show ip mroute
```

(此处省略)

```
(192.168.2.100, 237.0.0.1), 00:10:44/00:02:59, flags: LT
```

```
Incoming interface: Serial0/0/0, RPF nbr 192.168.12.2
```

Outgoing interface list:

```
Loopback0, Forward/Dense, 00:02:25/00:00:00
```

```
GigabitEthernet0/0, Prune/Dense, 00:01:43/00:01:16
```

//可以看到 R1 上的 g0/0 处于修剪状态，因为 R3 已经往 192.168.134.0/24 网段转发数据了，这样该网段才不会有 2 份组播流量。

(此处省略)

```
R3#show ip mroute
```

(此处省略)

```
(192.168.2.100, 237.0.0.1), 06:42:14/00:02:58, flags: T
```

```
Incoming interface: Serial0/0/1, RPF nbr 192.168.23.2
```

Outgoing interface list:

```
GigabitEthernet0/0, Forward/Dense, 00:13:21/00:00:00, A
```

//可以看到 R3 在往 192.168.134.0/24 网段转发数据。

(此处省略)

在 PC1 上停止 ping，在各个路由器上查看多播路由表，各个路由器的多播路由表在大约 3 分钟后会消失。这是因为没有了源，路由器无需再维护这些路由表了。

(5) 步骤 5: 查看 IGMP 组成员

```
R1#show ip igmp groups
```

IGMP Connected Group Membership

Group Address	Interface	Uptime	Expires	Last Reporter	Group Accounted
237.0.0.1	Loopback0	00:15:14	00:02:48	1.1.1.1	
224.0.1.40	GigabitEthernet0/0	00:28:07	00:02:50	192.168.134.4	
224.0.1.40	Serial0/0/0	07:30:07	stopped	192.168.12.1	

//这是我们用命令加入的成员

//以上是查看 R1 上各个接口上各组有什么成员存在。

(6) 步骤 6: 配置、检查 IGMP Snooping

IGMP Snooping 是在交换机上工作的。默认时就已经启用，也可以用以下命令打开：

```
S1(config)#ip igmp snooping
```

```
S1#show ip igmp snooping
```

Global IGMP Snooping configuration:

```
-----
IGMP snooping           : Enabled
IGMPv3 snooping (minimal) : Enabled
Report suppression      : Enabled
TCN solicit query       : Disabled
TCN flood query count    : 2
Last Member Query Interval : 1000
```

Vlan 1:

```
-----
IGMP snooping           : Enabled
IGMPv2 immediate leave   : Disabled
Explicit host tracking    : Enabled
Multicast router learning mode : pim-dvmrp
Last Member Query Interval : 1000
CGMP interoperability mode : IGMP_ONLY
```

//以上是显示交换机上的 IGMP Snooping 运行情况

```
S1#show ip igmp snooping groups
```

Vlan	Group	Type	Version	Port List
1	224.0.1.40	igmp	v2	Fa0/1, Fa0/3, Fa0/4

// f0/1、f0/3、f0/4 接口上有 224.0.1.40 组的成员存在，则应该往这些接口转发 224.0.1.40 的多播流，其他接口不转发，节约了带宽。

23.2 实验 2: PIM Sparse-Dense

1. 实验目的

通过本实验，读者可以掌握如下技能：

- (1) 理解 PIM Sparse-Dense 的工作原理
- (2) 掌握 PIM Sparse-Dense 的配置

2. 实验拓扑

如图 23-1。

3. 实验步骤

在实验 1 基础上继续本实验。

- (1) 步骤 1：配置 PIM Sparse，采用静态 RP

```
R1(config)#int loopback0
R1(config-if)#ip pim sparse-dense
//在接口上运行 pim sparse-dense 协议即可
R1(config)#int s0/0/0
R1(config-if)#ip pim sparse-dense
R1(config)#int g0/0
R1(config-if)#ip pim sparse-dense
R1(config)#ip pim rp-address 3.3.3.3
//以上静态配置 R3 为 RP。
R1(config)#ip pim spt-threshold infinity
//以上是防止从基于 RP 的树切换到基于源的树，默认是切换的，稍后介绍树的切换问题。
```

```
R2(config)#int loopback0
R2(config-if)#ip pim sparse-dense
R2(config)#int s0/0/0
R2(config-if)#ip pim sparse-dense
R2(config)#int s0/0/1
R2(config-if)#ip pim sparse-dense
R2(config)#int g0/0
R2(config-if)#ip pim sparse-dense
R2(config)#ip pim rp-address 3.3.3.3
```

```
R3(config)#int loopback0
R3(config-if)#ip pim sparse-dense
R3(config)#int s0/0/1
R3(config-if)#ip pim sparse-dense
R3(config)#int g0/0
R3(config-if)#ip pim sparse-dense
R3(config)#ip pim rp-address 3.3.3.3
```

```
R4(config)#int loopback0
R4(config-if)#ip pim sparse-dense
```

```
R4(config)#int g0/0
```

```
R4(config-if)#ip pim sparse-dense
```

```
R4(config)#ip pim rp-address 3.3.3.3
```

(2) 步骤 2: 测试组播路由

在 PC1 上执行 “`ping -t 237.0.0.1`”，在各个路由器上检查多播路由表：

```
R1#show ip mroute
```

(此处省略)

```
(*, 237.0.0.1), 07:31:41/00:02:59, RP 3.3.3.3, flags: SCL
```

```
  Incoming interface: GigabitEthernet0/0, RPF nbr 192.168.134.3
```

//可以看到 R1 是从 g0/0 接口接收到组播流的，原因在于 RP 是 R3，而从 R1 到达 R3，从 g0/0 是最远的。这时，组播流实际上是从 PC1 到 R2，再到 R3 (RP) 上，然后从 R3 的 g0/0 再到 R1 的 g0/0 接口，很显然绕了一圈，这是不合理的。

【技术要点】在基于 RP 的树中，组播数据是先从源到达 RP 的，然后从 RP 到达各个组成员。从 RP 到达各个组成员的数据转发是根据共享树的构建情况来转发；而从组播源到达 RP 的组播流的转发实际上也需要一棵树，这时采用是基于源的树。

(3) 步骤 3: 从基于 RP 的树切换到基于源的树

```
R1(config)#no ip pim spt-threshold infinity
```

//以上是允许从基于 RP 的树切换到基于源的树，实际上这是默认值，在步骤 1 中被我们修改了。

```
R1#show ip mroute
```

(此处省略)

```
(*, 237.0.0.1), 07:36:48/00:02:58, RP 3.3.3.3, flags: SJCL
```

```
  Incoming interface: GigabitEthernet0/0, RPF nbr 192.168.134.3
```

```
  Outgoing interface list:
```

```
    Loopback0, Forward/Sparse-Dense, 00:57:35/00:02:26
```

```
(192.168.2.100, 237.0.0.1), 00:00:01/00:02:59, flags: LJTL
```

```
  Incoming interface: Serial0/0/0, RPF nbr 192.168.12.2
```

```
  Outgoing interface list:
```

```
    Loopback0, Forward/Sparse-Dense, 00:00:01/00:02:58
```

//可以看到这时多了一条多播路由，这是基于源的路由，这时 R1 实际上是从 s0/0/0 接口接收到数据，显然这才是合理的。

【技术要点】树的切换是很重要、很复杂的事情。树的切换是避免组播流绕了一圈才到达接受者。默认时是会切换的，因此在 Sparse 模式中，也是存在基于源的树的。

(4) 步骤 4: 配置 Auto RP

在各个路由器上，去掉配置静态 RP 的命令，如下：

```
R1(config)#no ip pim rp-address 3.3.3.3
```

```
R2(config)#no ip pim rp-address 3.3.3.3
```

```
R3(config)#no ip pim rp-address 3.3.3.3
```

```
R4(config)#no ip pim rp-address 3.3.3.3
```

```
R2(config)# ip pim send-rp-discovery loopback 0 scope 255
```


//把 R2 配置为映射代理，Loopback0 为代理地址。
R3(config)# ip pim send-rp-announce Loopback0 scope 255
//把 R3 配置为候选 RP，Loopback0 为 RP 地址。

在各个路由器上检查 RP 的地址：
R1#show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
 RP 3.3.3.3 (?), v2v1
 Info source: 2.2.2.2 (?), elected via Auto-RP
 Uptime: 00:00:36, expires: 00:02:19

//在各个路由器上都可以看到，所有的组播地址的 RP 都已经映射到 R3 了。
再按照步骤 2 测试组播路由。

23.4 本章小结

本章介绍了组播的目的和基本工作原理，组播的原理甚是复杂。组播路由协议目的是建立一棵合理的转发树，并能动态地根据组成员的加入、退出调整转发树。本章介绍了 PIM Dense 和 PIM Sparse 这两个组播路由协议的配置。在 Dense 中，是先向所有网络转发数据，然后修剪不必要的树枝；而 Sparse 中，组播流先从源发到 RP 上，当有组成员加入时，才沿途打开接口转发组播流。在交换机上是采用 IGMP Snooping 和 CGMP 来决定应该往什么接口转发什么样的组流量。表 23-1 是本章出现的命令。

表 23-1 本章命令汇总

命令	作用
ip multicast-routing	启用组播路由功能
ip pim dense	在接口上运行 pim dense 协议
ip igmp join-group 237.0.0.1	该接口加入到 237.0.0.1 组中
show ip pim neighbor	显示了 pim 邻居
show ip mroute	显示组播路由表
show ip igmp groups	显示组成员
ip igmp snooping	在交换机上配置 IGMP Snooping 功能
show ip igmp snooping	显示交换机上的 IGMP Snooping 运行情况
show ip igmp snooping groups	交换机上显示各组的成员在什么接口上
ip pim sparse-dense	在接口上运行 pim sparse-dense 协议
ip pim spt-threshold infinity	静止从基于 RP 的树切换到基于源的树
ip pim send-rp-discovery loopback 0 scope 255	把路由器设为映射代理，其 Loopback0 为代理地址
ip pim send-rp-announce Loopback0 scope 255	把路由器设为候选 RP，其 Loopback0 为 RP 地址