

第 8 章 RIP 路由配置与管理

8.1 RIP 路由协议基础

RIP (Routing Information Protocol, 路由信息协议) 是应用较早、使用较普遍的内部网关协议 (Interior Gateway Protocol, IGP), 适用于小型同类网络的一个自治系统 (AS) 内的路由信息的传递。RIP 协议采用“距离矢量”(也就紧“距离”作为路由变量) 路由算法, 使用“跳数”, 即 Hop 来衡量到达目标地址的路由距离, 且最大有效跳数值仅为 15。

RIP 协议现在有两个版本, 即 RIPv1 和 RIPv2, 新版本 RIPv2 支持纯文本和 MD5 身份认证、路由汇总、无类别域间路由 (Classless Interdomain Routing, CIDR) 和可变长度子网掩码 (Variable-Length Subnet Masks, VLSM)。

8.1.1 动态路由分类

动态路由与静态路由最大的不同就是, 动态路由无需手动一跳跳地配置路由, 且可在网络通信流量或者网络拓扑结构发生变化时能自动调整路由路径, 实现持续路由服务, 无需管理员手动调整。而且动态路由信息的改变会通过路由更新报文向它们的邻居路由器依次通告, 最终实现整个网络中各路由器上维持稳定的全网路由表项。有关动态路由的基本特点在第 7.1.2 节已有详细介绍, 在此不再赘述。

这也是为什么动态路由配置 (无论是本章介绍的 RIP 动态路由, 还是后面各章将要介绍的 OSPF、EIGRP、BGP、IS-IS 等动态路由) 最基本的配置就是创建一个对应动态路由协议的进程, 然后把各路由器所连接的网络通过 `network` 命令进行宣告, 然后再由邻居路由器向它的其他邻居路由进行宣告, 一级级地传递下去。对于可以划分区域 (Area) 的动态路由协议 (如 OSPF、EIGRP、IS-IS 等) 还需要创建所需的区域。基本的配置就这么简单, 根本无需为到达每个外部网络或者外部网络中的主机配置专门的路由。

但要注意的是, 动态路由协议不是仅本章将要介绍的 RIP 一种, 本书后面介绍的 OSPF、EIGRP、IS-IS 和 BGP 等都是动态路由协议。总体来说, 如果从应用环境来分的话, 这些动态路由协议可分为“内部网关协议” (Interior Gateway Protocol, IGP) 和“外部网关协议” (Exterior Gateway Protocol, EGP) 两大类, 如图 8-1 所示。

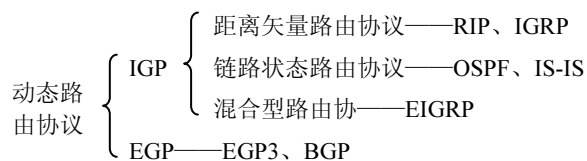


图 8-1 动态路由协议分类

IGP 是指用于在一个自治系统 (AS) 内部所使用的一种路由协议, 用于一个 AS 内部网关间 (也就是主机与路由器之间) 交换路由信息。而一个 AS 是指一组主机和使用相同路由选择协议的路由器集合, 并由单一机构管理, 通常是以一个组织的网络为边界。EGP 是为两个相邻的位于各自 AS 边界上的路由器提供一种路由信息交换的路由协议, 各 AS 内部的路由信息交换仍由 IGP 类型的路由协议来解决。

在整个动态路由协议中, 应用于 AS 内部的 IGP 类型路由协议更多。如果从所采用的路由算法类型来分的话, 这些 IGP 动态路由协议又可分为“距离矢量路由算法” (Distance Vector, DV)、

“链路状态路由算法”（Link State, LS），以及同时采用 DV 和 LS 两种算法的混合型路由协议。DV 类型的协议主要包括 RIP（路由信息协议）、IGRP（内部网关路由协议，是思科私有协议）这两种，而 LS 类型的动态路由协议又主要包括 OSPF（开放最短路径优先）、IS-IS（中间系统到中间系统）这两种；混合型的动态路由协议目前主要是思科私有的 EIGRP（增强型内部网关路由协议，也是思科私有协议）协议。

至于 EGP 类型的协议，目前主要是 BGP（边界网关协议），以前还有一种 EGP3（外部网关协议版本 3）协议，但不能把 EGP3 与我们这里所分类的 EGP 等同起来，EGP3 是一种独立的外部网关协议。**如果从路由算法上来分类的话，BGP 也可以说是一种混合型路由协议，因为它既有距离矢量路由协议的特点，又具有链路状态路由协议的特点，具体将在第 12 章介绍。**

8.1.2 RIP 路由的度量机制

每一种路由协议，都是想通过某一个参考变量来实现最佳路由的计算，也就是每一种路由协议都有它自己的路由度量机制。不同的路由协议所采取的路由度量机制是不一样的，这取决于各自所采用的路由算法。RIP 采用的是“跳数”（Hop Count）作为路由的度量。RIP 是一种简单的动态路由协议，它的设计思想就是按照不同路由路径的“距离”长短来决定不同路由的优先级的。而这“距离”就是 RIP 路由协议中所说的 hop（跳数），也即所经过的路由器或其他三层设备的个数（本地路由器不算）。“距离”越短的 RIP 路由优先级越高，最长为 15。

1. 理解“跳数”

RIP 协议采用的是“距离矢量”路由算法，是仅以“距离”作为度量标准，也就是路由的“距离”是 RIP 路由的度量。但要注意的是，这里“距离”不是指实际的通信线路长度，也不是两个网络或者两台主机相距的物理距离，而是指除源网络所连接的路由器外，到达目的网络的整条路由路径中所经过的路由器数，就是前面所说的“跳数”。

假设在图 8-2 的网络中，四个路由器都运行了 RIP 协议。现在要配置从 PC1 所在网络到达 PC2 所在网络的 RIP 路由，则跳数就是 3（PC1 直接连接和 R1 不算在内，其他每个路由器算一跳）。但是我们看一下，这四个路由器所连接的网络数达到了 5 个（从 1.1.1.0/24 到 5.1.1.0/24），当然这还只是最简单的网络了，因为每个路由器还可以连接多个网络。另外要注意的是，在同一个路由器上直接连接的多个网络，彼此间的度量值为 0，因为它们是直连路由。

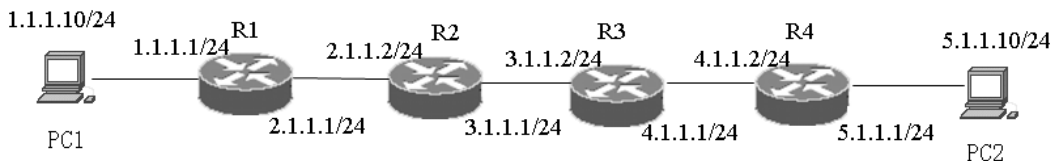


图 8-2 RIP 路由跳数计算示例

2. 度量更新规则

RIP 路由器采用完整路由表更新方式，也就是每个 RIP 路由器会把自己的路由表发给相邻的 RIP 路由器，以此来进行彼此的路由表更新。总体来说，它遵循以下几个基本原则：

- 路由表项每经过一次邻居路由器之间的传递，其度量值加 1（最大值为 15，下同）。

在如图 8-1 所示的网络中，R1 路由器把它所连接的 1.1.1.0/24 网络向 R2 通告，这时在 R2 上到达 1.1.1.0/24 网络的度量为 1，当 R2 再把它获取的到达 1.1.1.0/24 网络的路由发给 R3 时，R3 上到达 1.1.1.0/24 网络的度量就变为 2，而当 R3 把获取的到达 1.1.1.0/24 网络的路由发给 R4 时，R4 上

到达 1.1.1.0/24 网络的度量就变为 3。

- 路由器在收到新的路由更新表项时，会在其路由表中添加新的路由表项，其度量是在接收的路由表项度量基础上加 1，同时在新添加的路由表项中标注其下一跳地址就是发送路由更新的邻居路由器的接口。

如图 8-2 所示的示例中，假设 R4 在收到来自 R3 的路由表更新时发现包含了一条到达 6.1.1.0/24 网络的新路由表项，并且其度量为 2，这时 R3 就会在自己的路由表中添加这条新的路由表项，并且设置度量为 3 (=2+1)，“下一跳地址”为 R3 与 R4 相连的那个接口 IP 地址。

- 收到原有路由表项的路由更新时，先对有更新的路由表项的度量加 1，然后与对应的路由表项中原度量进行比较，**仅接收度量值更小或相等的更新，忽略度量值比原来的值更大的路由更新。**

假设在图 8-2 中，R2 中原有的路由表项如图 8-3 所示，现又收到来自 R3 的如图 8-4 所示的路由更新（启用了水平分割功能后就不会出现这样的更新了，本章后面将具体介绍）。

目的网络	下一跳	距离
2.1.1.0	—	0
3.1.1.0	—	0
4.1.1.0	3.1.1.2	1
5.1.1.0	3.1.1.2	2

图 8-3 R2 路由器上原来的路由表

目的网络	下一跳	距离
2.1.1.0	3.1.1.1	1
3.1.1.0	—	0
4.1.1.0	—	0
5.1.1.0	4.1.1.2	1

图 8-4 R2 收到来自 R3 的路由更新

对于 R2 来说，它在收到来自 R3 的路由更新后，首先对图 8-4 所示的各路由表项的度量加 1，结果得到的路由表项如图 8-5 所示。对比图 8-5 和图 8-3 可以看出，到达 2.1.1.0 网络和 3.1.1.0 网络的度量值比原来的还大，所以忽略更新，而到达 4.1.1.0 网络和 5.1.1.0 网络的度量值是相等的，进行路由表项更新，所以最终 R2 上的路由表还是如图 8-3 所示。

目的网络	下一跳	距离
2.1.1.0	3.1.1.2	2
3.1.1.0	3.1.1.2	1
4.1.1.0	3.1.1.2	1
5.1.1.0	3.1.1.2	2

图 8-5 R2 对收到的路由更新表项度量值加 1 后的路由表

如果一个接口连接网络的网络没有指定（也就是没有宣告所直接连接的网络），则它不会在任何 RIP 更新中被通告。如图 8-2 中在 R1 的配置中没有宣告它所连接的 1.1.1.0/24 网络，则其他路由器上也就没有到达这个网络的 RIP 路由表项，如果这些路由器仅启用了 RIP 协议，且不配置静态路由，则其他路由器也就不能到达 1.1.1.0/24 这个网络了。

当然，到达同一目的网络，也有可能距离是一样的，这时就可能存在同一目的网络的两条 RIP 路由表项。就像多条默认路由一样，这样两条相同目的网络的 RIP 路由也可以实现负载均衡。

8.1.3 RIP 路由更新机制

RIP 协议有两种更新机制：一是定期更新，二是触发更新。“定期更新”是根据设置的更新计

时器定期发送 RIP 路由通告。该通告报文中携带了除“水平分割”机制抑制的 RIP 路由之外本地路由器中的所有 RIP 路由信息。而“触发更新”则是 RIP 路由器仅在路由表项发生变化时发送的 RIP 路由通告，仅携带本地路由表中有变化的路由信息。RIP 路由器一旦察觉到网络变化，就尽快甚至是立即发送更新报文，而不等待更新周期结束。只要触发更新的速度足够快，就可以大大地防止“计数到无穷大”的发生，但是这一现象还是有可能发生的。

无论是定期更新，还是触发更新，RIP 路由的更新规则如下：

- 如果更新的某路由表项在路由表中没有，则直接在路由表中添加该路由表项；
- 如果路由表中已有相同目的网络的路由表项，且来源端口相同，那么无条件根据最新的路由信息更新其路由表；
- 如果路由表中已有相同目的网络的路由表项，但来源端口不同，则要比它们的度量值，将度量值较小的一个作为自己的路由表项；
- 如果路由表中已有相同目的网络的路由表项，且度量值相等，保留原来的路由表项。

下面主要介绍 RIP 路由的定期更新机制。

1. RIP 路由定期更新机制

RIP 路由器总是会每隔 30 秒（这是默认值，可以修改，而且也可能与设置值有些偏差）通过 UDP 520 端口以 RIP 广播应答方式向邻居路由器发送的一个路由更新包，包中包括了本路由器上的完整的路由表（除了被“水平分割”机制抑制的路由表项），用来向邻居路由器提供路由更新，同时用来向邻居路由器证明自己的存在。RIP 的路由表中主要包括“目的网络”、“下一跳地址”和“距离”这三个字段，参见图 8-3。

如果一个路由器在 180 秒（这也是默认值，可以修改）内没有收到某个邻居路由器发来的路由更新，则这个路由器就会标记该邻居路由器为不可达路由器，使这个邻居路由器处于抑制周期。当路由器处于抑制周期内，它仍然用于向前转发数据包，但网络中的其他路由器不学习到达该路由器所连网络的路由信息，除非是一条更好的到达该路由器所连网络的路由信息，如本来是 3 跳，在抑制周期内学到了一条 2 跳的路由信息。但抑制周期过后，即使是差的路由信息也接受。

如果在连续的 240 秒（这也是默认值，可以修改）内还没收到这个路由器的路由更新，则本地路由器会在路由表中删除与该邻居路由器相关的路由表项。

由此可见，这个路由更新不仅影响着整个 RIP 网络中的路由器上路由表的更新和所有需要到达，或者经过该路由器的数据包路由，还影响着其他邻居路由器是否当它存在。试想一下，如果有一个数据包是要发送到连接某个某个 RIP 路由器的网络的一台主机上，但这台 RIP 路由器当时恰好出现了故障，没有这个路由器更新机制的话，其他路由器也就不知道它当前出现了故障，仍按原来的路由路径传输数据包，结果当然是数据包总是无法到达目的主机了，尽管可能经过多次尝试。

2. RIP 路由定期更新机制解析示例

为了更好地理解 RIP 协议路由表的更新机制，下面以图 8-6 所示的简单的互连网络为例来讨论图中各个路由器中的路由表是怎样建立的。

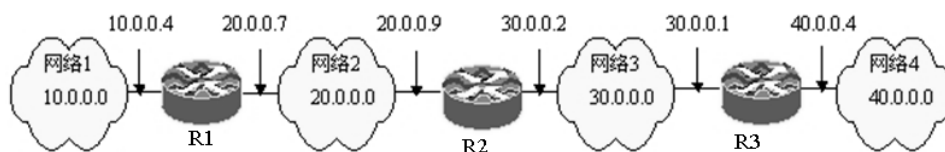


图 8-6 RIP 路由表建立网络示例

(1) 在一开始，所有路由器中的路由表只有自己所直接连接的网络的路由表项信息。但不是 RIP 路由表项，是直连路由表项，无需下一跳（用“--”表示），度量“距离”也均为 0，各路由

器的初始路由表如图 8-7 所示，均只有两条直连网络的路由表项。

目的网络	下一跳	距离
10.0.0.0	—	0
20.0.0.0	—	0

目的网络	下一跳	距离
20.0.0.0	—	0
30.0.0.0	—	0

目的网络	下一跳	距离
30.0.0.0	—	0
40.0.0.0	—	0

图 8-7 R1、R2 和 R3 的初始路由表

(2) 接下来，各路由器就会按设置的周期（默认为 30 秒）向邻居路由器发送路由更新了。具体哪个路由器会先发送路由更新，取决于哪个路由器先开了。现假设路由器 R2 先收到来自路由器 R1 和 R3 的路由更新，然后就更新自己的路由表，如图 8-8 所示。从中可以看出，它新添加了分别通过 R1 和 R3 到达 10.0.0.0 网络和 30.0.0.0 网络的路由表项，度量值均为 1，因为它只经过了一跳。

(3) R2 更新自己的路由表后，会把完整的路由表发给邻居路由器 R1 和 R3。路由器 R1 和 R3 分别再进行更新。根据前面介绍的 RIP 路由表更新的规则可以知道，R1 首先是把从 R2 上接收到的如果图 8-8 的路由表中，对每项度量进行加 1，得到的路由表如图 8-9 所示。

目的网络	下一跳	距离
20.0.0.0	—	0
30.0.0.0	—	0
10.0.0.0	20.0.0.7	1
40.0.0.0	30.1.1.1	1

图 8-8 R2 在路由更新后的路由表

目的网络	下一跳	距离
20.0.0.0	20.0.0.9	1
30.0.0.0	20.0.0.9	1
10.0.0.0	20.0.0.9	2
40.0.0.0	20.0.0.9	2

图 8-9 R1 对收到的来自 R2 路由表进行度量加 1 后形成的路由表

(4) R1 再把图 8-9 所示的路由表与自己原来的路由表（图 8-6 中的左图所示）进行比较，凡是新添加的，和度量值小于等于原来的路由表项均将更新，度量值更大的路由表项将忽略更新。经过行比较发现有两条新的路由表项，其目的网络分别为 30.0.0.0 和 40.0.0.0，直接在路由表中添加。而原来已有的两条 10.0.0.0 和 20.0.0.0 表项，发现路由度量（“距离”）值 1 比原来的 0 还大，忽略更新，结果就得到 R1 更新后的路由表，如图 8-10 所示。

用同样的方法可以得出 R3 在收到 R2 路由更新后的路由表如图 8-11 所示。但 RIP 路由协议存在一个问题，那就是网络收敛比较慢，当网络出现故障时，要经过比较长的时间才能将此信息传送到所有的路由器，而且中间有许多是无效路由更新。

目的网络	下一跳	距离
10.0.0.0	—	0
20.0.0.0	—	0
30.0.0.0	20.0.0.9	1
40.0.0.0	20.0.0.9	2

图 8-10 R1 在收到 R2 路由更新后的路由表

目的网络	下一跳	距离
30.0.0.0	—	0
40.0.0.0	—	0
10.0.0.0	30.0.0.2	2
20.0.0.0	30.0.0.2	1

图 8-11 R3 在收到 R2 路由更新后的路由表

仍以图 8-6 为例，现在三个路由器都已经建立了各自的稳定路由表，假设 R1 路由器和网 1（10.0.0.0）的连接线路断开了。此时 R1 可以立即发现，并更新自己的路由表，将到 10.0.0.0 的路

由表项距离改为 16（即不可达），并在 30 秒后将此路由由更新信息发给 R2。但是，R2 从 R3 得到的路由更新是“经过 R2 到达 10.0.0.0 网络的距离为 2”，明显度量值更小，于是 R2 将此路由表项更新为“经过 R3 到达 10.0.0.0 的距离为 3”，然后再通过路由更新发给 R3，此时 R3 的路由表中更新为“经过 R2 到达 10.0.0.0 网络的距离为 4”。R3 再通过路由更新发给 R2 信息，结果是“经过 R3 到达 10.0.0.0 网络的距离为 5”，一直如此反复，直到该路由表项的距离达到 16，R2 和 R3 才知道 10.0.0.0 网络是不可达的。

为了解决这一不足，产生了水平分割技术，就是同一路由表项更新不再从接收该路由表项的接口发送出去，具体将在本章后面介绍。下面再具体介绍 RIP 路由的收敛机制。

8.1.4 RIP 路由收敛机制

任何距离矢量类路由选择协议（RIP 也是这类路由协议）都有一个问题，路由器不知道网络的全局情况，路由器必须依靠相邻路由器来获取网络的可达信息。由于路由更新信息在网络上传播慢，所以所有距离矢量路由算法都有一个收敛慢的问题，这个问题将导致网络中各路由器路由信息不一致的现象产生。RIP 协议使用以下机制可以减少因网络上的不一致性带来的路由选择环路的可能性。

1. 记数到无穷大机制

RIP 协议允许最大跳数值为 15。大于 15 的目的地被认为是不可达的。这个数字在限制了网络大小的同时也防止了一个叫做“记数到无穷大”的问题。记数到无穷大机制的工作原理如图 8-12 所示。

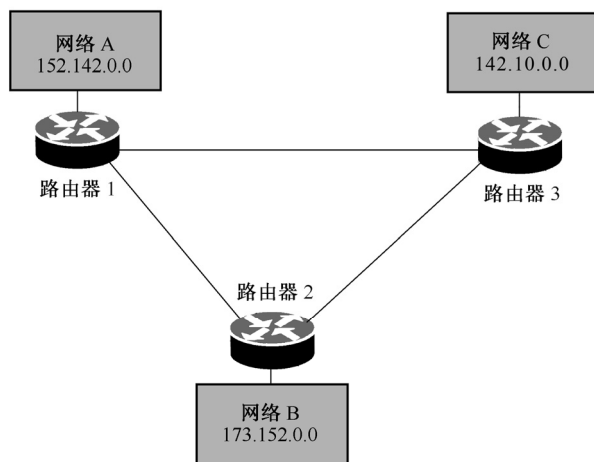


图 8-12 路由器收敛机制示例

(1) 现假设路由器 1 断开了与网络 A 相连，此时路由器 1 会立即产生一个路由更新向其邻居路由器 2 和路由器 3 通告，告诉它们，路由器 1 不再有到达网络 A 的路径。假设这个更新信息传输到路由器 2 被推迟了（CPU 忙、链路拥塞等），但到达了路由器 3，所以路由器 3 会立即从路由表中去掉到网络 A 的路径。

(2) 但由于路由器 2 没有收到路由器 1 的这个路由更新信息，于是它仍会定期向它的邻居（包括路由器 1 和路由器 3）发送路由更新信息，通告网络 A 是以 2 跳的距离可达。路由器 3 收到这个更新信息后，认为出现了一条通过路由器 2 到达网络 A 的新路径，于是路由器 3 告诉路由器 1，它能以 3 跳的距离到达网络 A。

(3) 路由器 1 在收到路由器 3 的路由更新后，就把这个信息加上一跳后向路由器 2 和路由器

3 同时发出更新信息，告诉它们路由器 1 可以以 3 跳的距离到达网络 A。

(4) 此时路由器 2 在收到路由器 1 的消息后，比较发现与原来到达网络 A 的路径不符，更新成可以以 4 跳的距离到达网络 A。这个消息再次会发往路由器 3，以此循环，直到跳数达到超过 RIP 协议允许的最大值（在 RIP 中定义为 16）。一旦一个路由器达到这个值，它将声明这条路径不可用，并从路由表中删除此路径。

由于记数到无穷大问题，路由选择信息将从一个路由器传到另一个路由器，每次跳数加 1。路由选择环路问题将无限制地进行下去，除非达到某个限制。这个限制就是 RIP 的最大跳数。当路径的跳数超过 15，这条路径才从路由表中删除。

2. 水平分割法

“水平分割”就是使路由器不向对应路由更新表项输入的方向回传此条路由表信息，使它只沿一个方向通告。通俗地讲就是，如果一条路由信息是从某个端口学习到的，那么从该端口发出的路由更新中将不再包含该条路由信息，其目的就是为了避免出现路由更新环路。

在如图 8-13 所示的网络中，路由器 2 通过帧中继连接路由器 1 和路由器 3，两个 PVC（永久虚电路）都在路由器 2 的同一个物理接口（S0）中止。如果在路由器 2 启用了水平分割功能，那么路由器 3 将收不到路由器 1 的路由选择信息，反之亦然。因为路由器 1，或路由器 3 向路由器 2 通告的路由更新信息不会再从 S0 接口向外通告给路由器 3，或路由器 1。

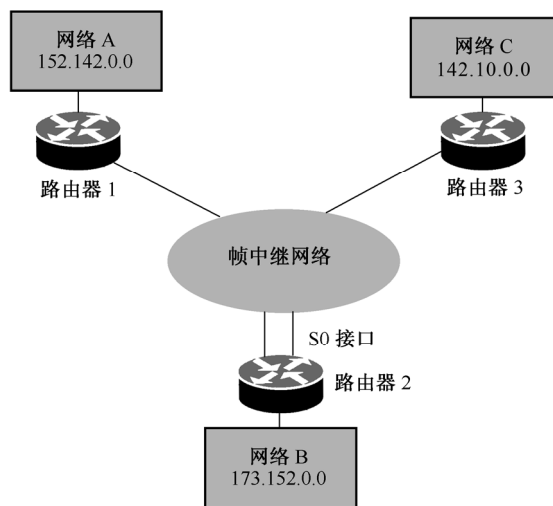


图 8-13 水平分割示例

在如图 8-6 所示的网络中，如果 R2 启用了水平分割功能，则 R2 在收到 R1 发来的包括 10.0.0.0 网络的路由更新后，在向 R1 发送的路由更新中就不会再包括 10.0.0.0 对应的路由表项了，这样就可以避免路由表发送的死循环出现。

有关 RIP 水平分割机制在本章后面还有具体的介绍。

3. 毒性逆转水平分割法

“水平分割”功能是路由器用来防止把一个接口得来的路由又从此接口回传，导致路由更新环路的出现。“毒性逆转水平分割”方法是在更新信息中包括这些回传路由，但会把这些回传路由的跳数直接设为 16（无穷），直接使该路由为不可达路由（这可能是取“毒性逆转”这么看似很严重的名称的原因吧）。通俗地讲就是，如果一条路由信息是从某个端口学习到的，那么从该端口发出的路由更新分组中将继续包含该条路由信息，但将这条信息的 metric 置为 16。通过把跳数设为无穷，并把这条路径告诉源路由器，能够更快的消除路由信息的环路。但它增加了路由更新的负担。

4. 保持定时器法

“保持定时器法”是设置路由信息被抑制的时间，默认为 180 秒。当路由器接收到一个不可达的路由更新时，路由器将会把这条路由更新置于无效抑制状态，不再接收对应路由的更新信息，也不向外发送这条路由更新信息，一直持续到接收到一个带有更好度量的对应路由更新分组，或者这个保持计时器到期为止。

在如图 8-12 所示的网络中，由于线路原因，从路由器 1 发往路由器 2 的路由更新被延迟到达，致使路由器 2 不能及时更新，所以路由器 2 仍会以更新新的错误路由信息向路由器 3 发送。但使用了“保持计数器法”后，这种情况将不会发生，因为路由器 3 在收到来自路由器 1 的网络 A 不可达的路由更新后，将在 180 秒内不接受通向网络 A 的新的路由信息，而经过这段时间后，路由器 2 也已正确进行了更新，将不会再发送错误的路由信息给路由器 3。

8.1.5 RIP 报文格式

前面说了，RIP 使用 UDP 报文来在邻居路由器间交换路由表，所使用的 UDP 端口号为 520。通常情况下 RIPv1 报文为广播报文；而 RIPv2 报文为组播报文，组播地址为 224.0.0.9，默认每隔 30 秒向邻居路由器发送一次路由更新报文。如果设备经过 180 秒没有收到来自对端的路由更新报文则将所有来自此设备的路由信息标志为不可达，若在 240 秒内仍未收到更新报文就将这些路由从路由表中删除。如图 8-14 所示为 RIP 路由更新报文格式，注意其中 v1 版本和 v2 版本的不同。

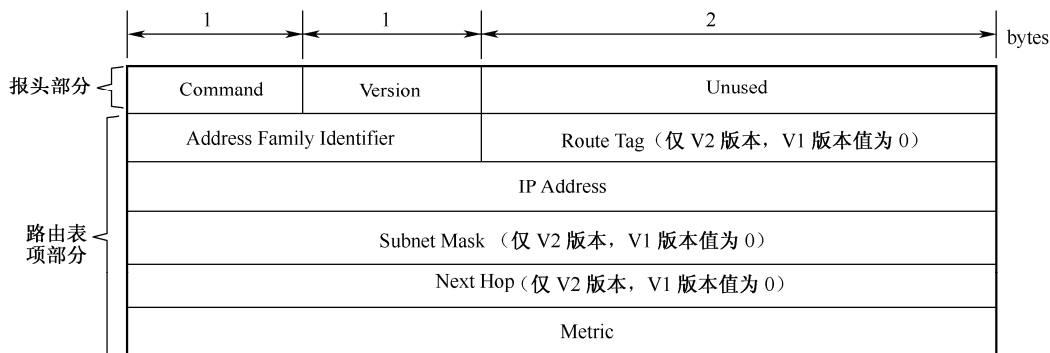


图 8-14 RIP 协议报文格式

Command

命令字段，占 1 个字节，用来指定数据报分组是请求分组，还是响应分组。在 v1 版本中主要有以下四种分组类型：Request（请求，对应值为 1）、Response（响应，对应值为 2）、Traceon（启用跟踪标记，对应值为 3，自 v2 版本后已经淘汰）、Traceoff（关闭跟踪标记，对应值为 4，自 v2 版本后已经淘汰）。请求分组是请求邻居路由器发送全部或部分路由表信息的分组，响应分组可以是路由器主动提供的周期性路由更新分组或者对请求分组的响应。

Version

版本字段，占 1 个字节，v1 版本值为 1，v2 版本值为 2。

Unused

未使用的字段，占 2 个字节，值固定为 0。

Address Family Identifier (AFI)

地址族标识符字段，占 2 个字节，指出所使用的地址族。RIP 设计用于携带多种不同协议的路由信息，每个项都有地址族标志来表明使用的地址类型，IP 地址的 AFI 是 2。

Route Tag

路由标记字段，占 2 个字节，提供区分内部路由（由 RIP 学得）和外部路由（由其他协议学得）的方法。它携带着一个 EGP 和 BGP 的自治系统号。该字段仅在 v2 版本使用，使用对应外部路由的标记值，v1 版本不用（值固定为 0），因为 v1 版本不支持由其他类型路由重发布为 RIP 路由。

IP Address

IP 地址字段，占 4 个字节，指定路由的目的网络地址，可以是标准网段地址、子网地址。

Subnet Mask

子网掩码字段，占 4 个字节，指定目的网络的子网掩码，仅在 v2 版本中使用，v1 版本中该字段值固定为 0。因为 v1 版本不支持无类别网络，也就是不支持子网，仅支持标准的有类网络，此时具体网段的子网掩码是固定的。

Next Hop

下一跳字段，占 4 个字节，指出 RIP 路由的下一跳的 IP 地址。如果为 0.0.0.0，则表示发布此条路由信息的路由器地址就是最优下一跳地址。该字段也仅在 v2 版本中使用，v1 版本中该字段值固定为 0，因为 v1 版本采用的是广播方式发送，无具体的下一跳。

Metric

RIP 路由的 Metric（度量）值字段，也就是“距离”值，占 4 个字节，最大有效值为 15，16 时表示该路由不可达了。

【说明】在以上整个 RIP 报文各字段中可分为“头部”（Header）和“路由表项”（Route Entries）两大部分。“头部”包括 Command、Version 和 Unused 这三个字段，其余字段都属于“路由表项”。在一个 RIP 报文中，最多可以有 25 个路由表项，也就是在一个 RIP 分组中最多可含有 25 个地址项，即一个分组中最多可一次性通告 25 条 RIP 路由表项。

8.2 Cisco 路由器 RIP 路由配置与管理

从本节开始就要正式介绍 RIP 动态路由的各方面配置。毕竟 RIP 主要用来满足中小型企业网络内部的路由需求，所以总体上 RIP 路由的配置还是非常简单的。最基本的配置就启用 RIP 协议，然后配置一些必要的参数，其他配置，像 RIP 认证、RIP 水平分割、源 IP 地址确认等特性都是可选的配置任务，只有在有这些方面需求时才进行配置。

Cisco 路由器中 RIP 路由可以进行的主要配置任务如下（除第一项为必选配置任务外，其他的均为可选配置任务）：

- 启用 RIP 路由进程，配置 RIP 参数（必选）
- 指定 RIP 版本，启用 RIP 认证（可选）
- 启用或禁用水平分割（可选）
- RIP 路由汇总（可选）
- 禁止源 IP 地址确认（可选）
- RIP 路由更新包间延时配置（可选）
- IP-RIP 延时开始配置（可选）

下面分别予以介绍。

8.2.1 RIP 进程启用和参数配置

本节要介绍如何在 Cisco 路由器上全局启用 RIP 路由进程，发布本地所连接的网络，配置偏移

列表、计时器等这些基本参数。其中以下三步是必需要进行的：

- 通过 **router rip** 全局配置模式命令启用 RIP 协议；
- 通过 **network ip-address** 路由器配置模式命令宣告本地路由器所连接的网络；
- 通过 **neighbor ip-address** 路由器配置模式命令指定与本地路由器直接连接的邻居路由器信息。

另外还可选配置 RIP 路由度量偏移和 RIP 路由的四种计时器值。在具体介绍 RIP 路由进程启用和参数配置方法前，先来了解以上提到的两个相对比较复杂的 RIP 技术——RIP 度量偏移和四种 RIP 计时器，以便可以正确理解将在本节后面要介绍的具体配置步骤。

1. RIP 度量偏移

在较大的 RIP 网络中，可能某台 RIP 路由器会从不同邻居路由器接收到了到达同一目的网络，且度量一样的路由更新，这时为了不出现路由冲突，就需要人为地改变这些不同路由表项的度量值，以实现一定程度上的优先级定义。

在 RIP 路由更新消息交换中有一个偏移（offset）机制，可用来在通过 RIP 协议学习到的路由中增加流入和流出路由的度量值，使得原来具有相同目的网络、相同度量的多条路由具有不同的度量值。因为从同一路由器上到达同一目的网络的路由，度量值越小，优先级越高，所以这样一样就可以使得这些原来相同度量、相同目的网络的多条路由具有了不同的优先级，或者使某些路由无效（因为 RIP 路由的最大值为 15，当改为 16 或以上时就无效了）。

你可以使用以下 **offset-list** 路由器配置模式命令，通过标准 ACL 列表（限制路由的源 IP 地址）来限定要应用偏移设置的路由，也可以通过指定具体的接口来限制要应用偏移设置的路由器接口：

```
offset-list {access-list-number | access-list-name} {in | out} offset {interface-type interface-number}
```

- **access-list-number | access-list-name**：二选一参数，指定用来限制应用度量偏移设置的 RIP 路由源 IP 地址的标准 ACL 列表号或列表名称。此 ACL 必须在路由器上同时配置。
- **in | out**：二选一选项，指定度量偏移设置是应用于接收到上的 RIP 路由，还是发出的 RIP 路由。
- **offset**：具体设置的 RIP 路由度量偏移值。
- **interface-type interface-number**：二选一参数，指定度量偏移设置要应用在本地 RIP 路由器的哪个接口上。

要取消原来所设置的偏移，则可用对应的 **no offset-list** {access-list-number | access-list-name} {in | out} offset {interface-type | interface-number} 命令。

如下的配置是把 serial 1/0 接口上接收到的所有源 IP 地址在 ACL 1 所指定的网段的 RIP 路由度量增加 16，这样如果 RIP 路由器没有其他接口所连接网络启用 RIP 进程时，则该路由器上所有 RIP 路由都是不可达的了，因为 RIP 路由的最大度量值就只是 15。可以在用户模式或者特权模式下通过 **show ip route** 命令查看所有路由得到验证，其中已没有 RIP 路由了。

```
R1(config)#router rip
R1(config-router)#offset-list 1 in 16 s1/0

R1#show ip router
  1.0.0/24 is subnetted, 1 subnets
C    1.1.1.0 is directly connected, Loopback0
  12.0.0/24 is subnetted, 1 subnets
C    12.1.1.0 is directly connected, Serial1/0
```

2. RIP 路由协议的四个主要计时器

RIP 路由协议主要使用 Update Timer（更新计时器）、Invalid Timer（失效计时器）、Holddown

Timer（抑制计时器）和 Flush Timer（刷新计时器）这四个计时器。它们可以通过 `timers basic update invalid holddown flush` 路由器配置模式命令分别进行配置。它们具体的作用分别如下：

- Update Timer（更新计时器）

“更新计时器”（对应命令中的 `update` 参数）是指运行 RIP 协议的路由器向自己的邻居路由器发布路由更新的时间间隔，默认是 30 秒。但对于具体接口来讲，这个时间也不是绝对的，包含了一个随机变量，以防止所有路由器都同时发送路由更新造成太大流量，给实际的数据通信带来太大压力。Cisco IOS 中的更新计时器允许有一个向内 15% 的偏移，因此，默认情况下在 Cisco 路由器中的实际 RIP 路由更新时间是在 25.5~30 秒之间。当然，这个计时器的默认值也是可以由管理员自己修改，具体的方法就是通过前面说到的 `timers basic` 命令。

- Invalid Timer（失效计时器）

“失效计时器”（对应命令中的 `invalid` 参数）是一个针对路由表中的特定路由表项的计时器，用来确定本地路由器认为自己由表中某条 RIP 路由无效的时间间隔，也就是说在路由表中的 RIP 路由表项不是永久有效的，这也是动态路由中“动态”特性的一个表现。

RIP 路由默认的失效时间为 180 秒，也就是默认情况下，从路由器接收到这个路由表项更新开始，在没有后续更新的情况下，最多保持 180 秒。如果要修改这个失效计时器，同样可以用到前面说到的 `timers basic` 命令，但该计时器的值至少应为“更新计时器”（也就是 `update` 参数值）的 3 倍。

路由器是依据什么来确定某条 RIP 路由表项到了失效时间呢？那就是在这个 `invalid` 参数指定的时间内（默认就是 180 秒）没有收到包含该条路由的更新消息。但这里要明确的是，不一定非要某条路由表项本身有所变化，只要能在这个时间内从邻居路由器上再次收到包含该路由表项的路由更新包，让本地 RIP 路由器知道这条路由还存在就行（可以是没有变化的，当然也可以是有变化的）。就像学生离校在外要定期向学校报告，让学校的老师知道你一切正常就可以了一样。



经验之谈

前面我们已说到，RIP 有定期更新和触发更新两种。在 RIP 定期路由更新中，不会仅发送包括某一条或几条路由表项的路由更新包，而是把本地路由器上整个路由表（除了被“水平分割”机制所抑制的路由表项外）向邻居路由器发送路由更新包的；而“触发更新”是仅发送有变化的路由表项到邻居路由器上。

当有一条新的 RIP 路由被建立，则该路由的失效计时器就会被初始化为 180 秒。RIP 路由器每收到一次包含该路由表项的更新就把该计时器置 0。也就是说每条 RIP 路由必须在每隔该计时器时间内收到包含该条 RIP 路由的更新消息，否则路由器就标记该 RIP 路由无效，并向所有其邻居路由器通告对应 RIP 路由不可达，邻居路由器在收到这样的广播消息后仍标记该 RIP 路由无效。这时该条 RIP 路由将进入到后面的“抑制”（Holddown）阶段，但是**此时 RIP 路由仍然用来转发数据包**。

- Holddown Timer（抑制计时器）

“抑制计时器”（对应命令中的 `holddown` 参数）是指路由器如果在相同的接口上收到某个 RIP 路由表项的度量比原先收到的相同目的网络的 RIP 路由表项的度量更小（也就是比原来同目的网络的路由更优），那么路由器将启动一个“抑制计时器”，默认为 180 秒。同样可以通过 `timers basic` 命令修改抑制计时器值，但也应至少等于“更新计时器”（也就是 `update` 参数值）的 3 倍。

“抑制计时器”主要用来防止 RIP 路由环路，该计时器的原理是引用一个怀疑量，不管是真的还是假的路由消息，路由器先认为是假消息来避免路由环路。如果在抑制计时器超时后还接受到这样的度量更小的路由消息，那么这时路由器就认为该路由消息是真的。在抑制计时器的时间内该路由表项被标记为不可到达，直到“抑制计时器”超时，路由器才对该路由表项进行更新。

“抑制计时器”还应用于当 RIP 路由器从邻居路由器收到一条指示某条 RIP 路由表项不可达的路由更新时。也就是在该条 RIP 路由表项到了“失效计时器”，由邻居路由器发送该条路由不可达的路由更新后，这条 RIP 路由就进入了抑制期，这在前面介绍“抑制计时器”时就有说到。进入抑制期的 RIP 路由将被标记为不可访问，并且会继续向邻居路由器发送该路由不可达的路由更新。但是**此时 RIP 路由仍然用来转发数据包**，直到“抑制计时器”到期后，路由器才会接受到达相同目的网络的新的 RIP 路由。

- Flush Timer（刷新计时器）

“刷新计时器”（对应命令中的 *flush* 参数）其实是“失效计时器”的延续。“失效计时器”是把指定的时间内没有收到更新的路由视为无效，而这里的“刷新计时器”则是把在这个指定时间内没有收到路由更新的路由从路由表中彻底删除，刷新路由拓扑。“刷新计时器”的默认时间为 240 秒，同样你可以通过 **timers basic** 命令修改刷新计时器值，但应当大于所设置的“失效计时器”的值（也就是 *invalid* 参数值）。因为我们建议“失效计时器”值应至少等于 3 倍“更新计时器”值，所以“刷新计时器”值应大于 3 倍“更新计时器”值。

可以这么理解“失效计时器”和“刷新计时器”之间的关系：“失效计时器”看成是记过处分，还有改过自新的机会，也就是该路由还可以在路由表中存在，并且可以重新被激活（在到达“刷新计时器”之前重新收到该路由表项的更新时）；而“刷新计时器”则可看成是开除处分，没有改过自新的机会了，到期后就会从路由表中删除。删除了的路由是不能再在路由表中存在，更不能恢复了，只能重新建立。

但是“刷新计时器”与“失效计时器”是同时启动的，也就是当一个路由表项默认在 180 秒内没有收到路由更新时，“失效计时器”超时。此时该路由表项将被标志为“x.x.x.x is possibly down”，而默认到了 240 秒时，“刷新计时器”也超时了，直接从路由器的路由表中删除该条路由表项。

3. RIP 启动和参数配置步骤

本节所涉及的主要技术在前面都有介绍了，下面介绍在使用 RIP 路由时必须进行的配置（有些配置步骤是可选的），具体配置步骤如表 8-1 所示。

表 8-1 RIP 路由进程启动及参数配置步骤

步骤	命令	说明
1	enable 例如： Router> enable	进入特权模式。如果配置了特权模式密码，则要按提示输入正确的特权模式密码才能进入
2	configure terminal 例如： Router# configure terminal	进入全局配置模式
3	router rip 例如： Router (config)# router rip	在本地路由器上启用 RIP 路由进程，进入路由器配置模式
4	network ip-address 例如： Router (config-router)# network 10.1.1.0	宣告本地路由器上直接连接，需要启用 RIP 路由进程的网络，相当于一要在对应网络中启用 RIP 路由进程。凡是需要启用 RIP 路由进程的网络都要单独用一条该命令进行宣告。参数 <i>ip-address</i> 用来指定要宣告的网络的网络地址 示例中定义的一个直连网络为 10.1.1.0
5	neighbor ip-address 例如： Router(config-router)# neighbor 1.1.1.2	定义一个需要与本地路由器交换路由信息的邻居路由器。可定义多个参数 <i>ip-address</i> 用来指定对应邻居路由器的 IP 地址（实际是邻居路由器的某个接口 IP 地址） 示例中指定的其中一个邻居路由器接口 IP 地址为 1.1.1.2
6	offset-list access-list-number access-list-name] {in out} offset [interface-type interface-number] 例如： Router(config-router)# offset-list 98 in 1 Ethernet 1/0	（可选）向通过 RIP 学习到的流入或流出路由度量添加一个偏移，具体参见本节前面介绍 示例中设置的是将在 ACL 98 中所定义的 RIP 路由中，在 Ethernet 1/0 接口流入方向上的路由原度量基础上加 1

续表

步骤	命令	说明
7	timers basic update invalid holddown flush [sleeptime] 例如： Router(config-router)# timers basic 50 150 150 200	(可选) 调整 RIP 路由协议各计时器设置，各参数的具体说明参见本节前面介绍 示例中配置的更新计时器、失效计时器、抑制计时器和刷新计时器的值分别为 50、150、160、200 秒
8	end 例如： Router(config-router)# end	退出路由器配置模式，返回特权模式。

8.2.2 RIP 版本指定和 RIP 认证启用配置

本节所介绍的配置任务是可选的，仅当需要在 RIP 路由器上启用 RIPv2 版本，并使用 RIPv2 版本中的路由更新认证时才需要进行。

前面说了，RIP 协议有两个版本，而且两个版本所支持的功能不完全一样，默认启用的是 RIPv1 版本。如果路由器支持 RIPv2 版本，而且又想使用 v2 版本的一些功能（如 RIP 路由更新认证），则需要完成本节所要介绍的配置步骤。下面先来介绍与本节配置密切相关的 RIP 路由更新认证。

1. RIP 路由更新认证概述

RIPv2 支持认证、密钥管理、路由汇总、CIDR 和 VLSM。默认情况下，Cisco IOS 软件可以同时接收 RIPv1 和 RIPv2 版本消息包，但是仅发送 RIPv1 版本数据包。你可以通过指 RIPv2 版本，使 IOS 软件仅接收和发送 RIPv1 或者 RIPv2 版本消息包。考虑到默认行为，你可以配置一个接口仅发送哪个版本的 RIP 消息包，你也可以控制一个接口仅接收哪个版本的 RIP 消息包。但 RIPv1 版本不支持身份认证。如果你发送和接收 RIPv2 版本消息包，则你可以在接口上启用 RIP 身份认证。

通过配置 RIP 邻居路由认证功能来预防 RIP 路由器接收到欺骗性的路由更新。在配置了 RIP 路由更新认证功能后，在两个邻居路由器之间交换路由信息就会对邻居进行认证，只有密钥一致的路由更新才会接收。这样就可以确保 RIP 路由器仅从信任源接收到可靠的路由信息。

如果没有启用 RIP 邻居路由认证功能，非授权或者恶意 RIP 路由更新可能威胁到网络通信安全。例如，一个路由器可能会发送一个虚假的路由更新，使得你的路由器发送通信到错误的目的站点。这种转移通信可能使恶意人员分析你公司组织机密信息，或者中断你公司正常的网络通信。而有了 RIP 路由更新认证功能后，就可以阻止你的 RIP 路由器接收任何欺骗性（密钥不正确的都认为是带有欺骗性的）的路由更新。

RIP 有两种类型的邻居认证：纯文本认证和 MD5 消息摘要认证。它们的认证方式是一样的，只是 MD5 认证中发送的是经过加密的消息摘要，而不是发送自己的认证密钥，更加安全。消息摘要使用认证密钥和发送的消息共同创建的，但是不发送自己的认证密钥，以防在传输过程中被截取。纯文本认证会通过线路发送自己的认证密钥，显然安全性不够高。在纯文本认证中，每个参与通信的邻居路由器必须共享一个认证密钥。这个密钥需要在每个路由器上配置。可以为不同协议通信配置不同的密钥，但是每个密钥必须用一个密钥号进行标识。

2. RIP 路由更新认证原理

通常，在 RIP 路由更新包发送后，会按以下流程进行纯文本认证：

(1) 一路由器发送一个带有它自己认证密钥和相应的密钥号码的路由更新到邻居路由器。在只有一个协议的情况下，密钥号总为 0。邻居路由器接收到路由更新后，用它自己的认证密钥与更新包中的密钥进行比对。

(2) 如果发现两个密钥是一样的，邻居 RIP 路由器就接收这个更新包，否则邻居 RIP 路由拒

绝接收这个更新包。

MD5 认证方式与纯文本认证方式的工作原理类似，只是它从不会在线路中直接发送自己的密钥，而是使用 MD5 算法产生一个哈希摘要消息。然后用这个摘要随着路由更新包一起发送，以确保不能通过在线路上偷听而获取认证密钥。

另外，RIP 路由更新认证是使用密钥链来配置、管理密钥的。当你配置了一个密钥链后，你可以指定一系列带有有效期的密钥（相当于创建了一个密钥库），IOS 软件可以轮流使用这些密钥对邻居 RIP 路由器的路由更新进行认证，这样可以减少破解认证密钥的可能性，更加安全。

3. RIP 版本指定和 RIP 认证启用配置步骤

指定 RIP 版本的命令是 **version**，而且还可以针对具体接口指定发送或者可以接收的 RIP 消息包版本，所使用的命令分别是：**ip rip send version** 和 **ip rip receive version**。启用 RIPv2 认证的配置步骤包括两个命令，一是用于配置认证密钥链的 **ip rip authentication key-chain** 命令，另一个是指定认证方式的 **ip rip authentication mode {text | md5}** 命令。具体配置步骤如表 8-2 所示。

表 8-2 RIP 版本指定和 RIP 认证启用的配置步骤

步骤	命令	说明
1	enable 例如： Router> enable	进入特权模式。如果配置了特权模式密码，则应按提示输入正确的特权模式密码才能进入
2	configure terminal 例如： Router# configure terminal	进入全局配置模式
3	router rip 例如： Router(config)# router rip	启用 RIP 路由进程，进入路由器配置模式
4	version {1 2} 例如： Router(config-router)# version 1	(可选) 全局配置 RIP 路由器使用的 RIP 协议版本，可以不全局指定，而在下面基于接口指定 可用 no version 命令删除全局配置，恢复默认的版本支持，即可以同时接收 v1 和 v2 版本的 RIP 路由更新包，但仅可以发送 v1 版本的路由更新包
5	exit 例如： Router(config-router)# exit	退出路由器配置模式，返回全局配置模式
6	key chain name-of-chain 例如： Router(config)# key chain rip-md5	指定 RIP 路由更新认证所用的密钥链（仅在启用 RIPv2 时有效）。参数 <i>name-of-chain</i> 用来指定密钥名称。一个密钥链中可以最多有 2147483647 个密钥
7	key key-id 例如： Route (config-keychain)# key 1	为密钥链中的每个密钥指定一个密钥 ID，有多少个密钥，就需要配置多少个密钥 ID。参数 <i>key-id</i> 的取值范围为 0~2147483647，不一定是连续的。可用 no key key-id 命令删除原来所配置的密钥 ID
8	key-string text 例如： Router (config-keychain)# key-string winda	指定用于 RIP 路由更新认证的真正密钥字符串。可用 no key-string text 命令删除原来所配置的密钥字符串
9	accept-lifetime start-time {infinite end-time duration seconds} 例如： Router (config-keychain)# accept-lifetime 13:30:00 Jan 25 2013 duration 7200	指定密钥链中的密钥可以被有效接收的时间。命令的各参数和选项说明如下： <ul style="list-style-type: none"> ● start-time: 指定密钥可以被有效接收的起始时间，格式可以是如下两种： <ul style="list-style-type: none"> ➢ <i>hh:mm:ss Month date year</i> ➢ <i>hh:mm:ss date Month year</i> ● infinite: 多选一选项，指定密码可以被不受限制地被接收 ● end-time: 多选一参数，指定密钥可以被有效接收的终止时间 ● duration seconds: 多选一参数，指定密钥可以被有效接收的持续时长度（以秒为单位），取值范围为 1~2147483646 秒

续表

步骤	命令	说明
10	send-lifetime <i>start-time</i> {infinite <i>end-time</i> <i>duration seconds</i> } 例如： Router (config-keychain)# send-lifetime 15:00:00 Jan 25 2013 duration 3600	指定密钥中的密钥可以被有效发送的时间，各参数和选项说明参见上步对应介绍，不同的这里是指定密钥可以有效发送的时间
11	exit 例如： Router(config-keychain-key)# exit	退出密钥链密钥配置模式，返回密钥链配置模式
12	exit 例如： Router(config-keychain)# exit	退出密钥链配置模式，返回全局配置模式
13	interface <i>type number</i> 例如： Router(config)# interface Ethernet 3/0	(可选) 指定要配置 RIP 路由更新认证的接口 (可以是子接口)，进入接口配置模式。该接口 (或子接口) 要事先配置好 IP 地址等基本参数
14	ip rip send version [1] [2] 例如： Router(config-if)# ip rip send version 1	(可选) 配置上述接口可以接收的路由更新包的 RIP 版本。它将覆盖通过 version {1 2} 全局配置模式命令所设置的 RIP 版本
15	ip rip receive version [1] [2] 例如： Router(config-if)# ip rip receive version 1	(可选) 配置上述接口可以发送的路由包的 RIP 版本。它将覆盖通过 version {1 2} 全局配置模式命令所设置的 RIP 版本
16	ip rip authentication key-chain <i>name-of-chain</i> 例如： Router(config-if)# ip rip authentication key-chain chainname	(可选) 当接口启用 RIPv2 时启用 RIP 路由更新认证，并通过 <i>name-of-chain</i> 参数指定使用第 6 步所创建的认证密钥链
17	ip rip authentication mode {text md5} 例如： Router(config-if)# ip rip authentication mode md5	(可选) 当接口启用 RIPv2 时指定使用 text (纯文本认证，也称明文认证) 或者 MD5 认证方式
18	end 例如： Router(config-if)# end	退出接口配置模式，返回到特权模式

以下示例是配置一个名为 keychain1 的密钥链。密钥字符串的有效接收时间范围是 1:30 p.m~3:30 p.m，有效被发送的时间范围为 2:00 p.m~3:00 p.m；密钥字符串的有效接收时间范围是 2:30 p.m~4:30 p.m，有效被发送的时间范围为 3:00 p.m~4:00 p.m。

```

Router(config)#router rip
Router(config-router)# network 172.19.0.0  !---宣告所连接的网络 172.19.0.0
Router(config-router)#version 2  !---全局指定启用 RIPv2 版本，也就是可以同时接收和发送 v2 版本的 RIP 路由更新包
Router(config-router)#exit
Router(config)#key chain keychain1  !---定义一个名为 keychain1 的密钥链
Router(config-keychain)#key 1  !---创建一个 ID 号为 1 的密钥
Router(config-keychain)#key-string string1  !---定义 1 号密钥的密钥字符串
Router(config-keychain-key)#accept-lifetime 13:30:00 Jan 25 2013 duration 7200  !---指定包含 1 号密钥的路由更新包可以有效接收的时间
Router(config-keychain-key)#send-lifetime 14:00:00 Jan 25 2013 duration 3600  !---指定包含 1 号密钥的路由更新包可以有效发送的时间
Router(config-keychain-key)#exit
Router(config-keychain)#key 2
Router(config-keychain-key)#key-string string2
Router(config-keychain-key)#accept-lifetime 14:30:00 Jan 25 2013 duration 7200
Router(config-keychain-key)#send-lifetime 15:00:00 Jan 25 2013 duration 3600
Router(config-keychain-key)#exit
Router(config-keychain)# exit
Router(config)#interface ethernet 0  !---进入连接 172.19.0.0 网络的路由器接口
Router(config-if)#ip rip authentication mode md5  !---指定以上接口采用 MD5 RIP 认证模式
Router(config-if)# ip rip authentication key-chain keychain1  !---指定以上接口 RIP 路由更新认证所用的密钥链为 keychain1

```

8.2.3 启用或者禁止水平分割

RIP 水平分割机制默认是启用的，一般不用修改配置。本节属于选配任务，一般仅当在一些非

广播类型的路由器接口上 RIP 的水平分割功能。

通常，在连接到广播类型 IP 网络情形中，使用距离矢量路由协议（包括 RIP 和 IGRP）的路由器会使用水平分割（Split horizon）机制来减少可能出现的路由环路。水平分割功能可以阻止从一个接口上接收或学习到的路由信息再从该接口上被通告出去（它将把这些路由表项从 RIP 的定期路由更新包清除），主要用于广播类型的网络（如以太网局域网）中。其实可以从其名称的“水平分割”加以理解，水平分割就是要求路由消息一直向前发送，不要回头。这样要求发送的路由消息中目的地址就不能是路由源接口。

1. 水平分割原理剖析

在 8.1.4 节简单介绍了 RIP 的水平分割技术，本节再举一个示例进行具体的分析。示例的网络结构如图 8-15 所示。

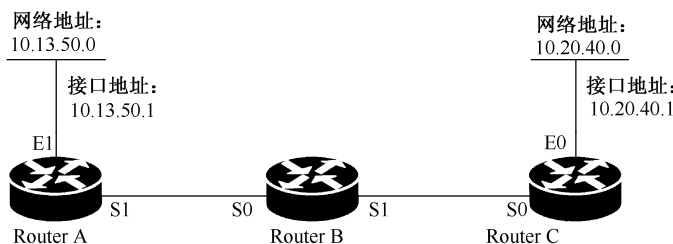


图 8-15 水平分割示例拓扑结构

如果 Router B 的 S1 接口从 Router C 的 S0 接口上接收到了一个到达 10.20.40.0 网络的路由通告信息（假设为路由①）。原路由的路径为：Router C 的 S0 接口→Router C 的 E0 接口→目的网络就是 10.20.40.0，度量为 0，因为 10.20.40.0 网络是直接连接在 Router C 上的。Router B 学习到后会在自己的路由表中创建一条新路由（假设为路由②）。该路由的路径为：Router B 的 S1 接口→Router C 的 S0 接口→目的网络 10.20.40.0，度量为 1。

如果此时允许 Router B 再通过其 S1 接口（原来学习到达网络 10.20.40.0 的①路由的接口）向 Router C 的 S0 接口发送包含这个路由的更新通告，则 Router C 就又会创建一条新路由（假设为路由③）。该路由的路径为：Router C 的 S0 接口→Router B 的 S1 接口→目的网络 10.20.40.0，度量为 2。同样，Router C 又会再通过其 S0 接口向 Router B 通告其新创建的路由，结果一直这么循环下去，直到所创建的路由度量大于 15（每次新建的 RIP 路由度量都是在学习到的源 RIP 路由度量基础上加 1），被宣告为不可达才结束。

这种现象显然不是我们所希望发生的，因为一则在同一台路由器上会为到达相同目的网络建立许多条度量不同的 RIP 路由，极大地增加了路由表的容量，使得路由更复杂；再则还可能形成路由环路，路由效率大大降低，甚至可以使数据包无法到达真正的目的地。如 Router C 中如果采用了③号路由，则访问网络 10.20.40.0 的数据包将不会直接从本地路由器上到达，而是再迂回到 Router B 的 S1 接口，然后再进入 Router C 的 S0 接口，最后才到目的网络 10.20.40.0。

为了解决这些问题，就推出了“水平分割”机制，禁止从学习某条 RIP 路由的接口上再通告该条路由的更新，这样就可以避免在同一路由器上创建到达同一网络的多条 RIP 路由。但是对于非广播类型网络，如帧中继和 SMDS（交换多兆位数字系统），因为出现路由环路的机会相当小，所以你可以禁止 RIP 的水平分割功能。除了采用了帧中继或者 SMDS 封装的其他所有接口，默认是启用了水平分割的。如果在接口配置中包括了 **encapsulation frame-relay** 或者 **encapsulation smds** 命令，则默认是禁用水平分割功能的。使用任何 X.25 协议封装的接口默认都不禁用水平分割功能的。

【注意】如果在一个接口配置了辅助 IP 地址，并且启用了水平分割功能，则通过辅助 IP 地址

进行路由更新时也可能将不会成功，因为水平分割是基于接收路由的源接口的。另外，RIP 路由更新是基于每个标准网络进行的（也就是基于整个网络进行更新的），而不是基于子网进行的，除非你禁止了水平分割功能。也就是说，**启用水平分割功能后，就可以基于子网进行路由更新了。**

在有些情况下又应禁止水平分割机制，最常见的一种环境是帧中继的 Hub-Spoke 结构的网络。Hub 路由器接收到 Spoke 端的路由更新，必须再从这个接口发出去，以便其他 Spoke 路由器来接收这些路由。如果启用了水平分割，从一个 Spoke 传递到 Hub 的路由将无法到达其他 Spoke 路由器，导致 Spoke 路由器之间相互没有路由。

2. RIP 水平分割的启用与禁用

RIP 水平分割的功能启用与禁用是基于接口配置的，所以需要在对应的 RIP 路由器接口配置模式下进行。启用 RIP 水平分割技术的命令很简单，就是 **ip split-horizon** 命令，如果要禁止某接口的水平分割技术，则可用 **no ip split-horizon** 命令。

除了帧中继，或者 X.25、SMDS 封装的接口外，其他所有接口默认都是启用水平分割功能的。

以下示例显示了如何在采用 x.25 协议封装的串行接口上禁止水平分割功能。

```
Router(config)#interface serial 0
Router(config-if)#encapsulation x25
Router(config-if)#no ip split-horizon
```

8.2.4 RIP 路由汇总配置

“路由汇总”（Route Summarization）就是把多个小网络或子网的路由汇总成一个大网络或大子网的路由，各种动态路由协议都有此功能，不仅本章介绍的 RIP。通过路由汇总功能就可以把位于同一标准网络下的多个子网路由汇总成一条汇总路由，可以提高 RIP 路由在大型网络中的可伸缩性和路由效率。有了路由汇总功能后，在 RIP 路由表中无需为子路由（也就是为一个汇总地址内部的任何单一子网所创建的路由）配置专门的路由项，减小路由表的大小，使路由器可以处理更多的路由。

Cisco IOS 系统默认启用自动路由汇总功能，**仅当需要关闭全局自动路由汇总功能，或者需要基路由器接口进行路由汇总才需要进行本节的配置任务。**

1. 路由汇总简介

RIP 有两种路由汇总方式：一是全局自动汇总，二是基于接口的手动汇总。RIPv1 仅支持自动汇总方式，RIPv2 同时支持自动汇总和手动汇总这两种方式。

汇总路由比多个个别通告 IP 路由具有更高的效率，主要原因如下：①在 RIP 路由数据库中，汇总路由将被优先处理；②任何包括在一个汇总路由中的子路由在路由数据库查找中是直接跳过的，减少了所需的处理时间。但**如果同时在对应的 RIP 路由器接口上启用了水平分割功能，则子网路由和汇总路由都不会在该接口上被通告。**

下面分别介绍这两种 RIP 路由汇总方式：

● 自动汇总

RIPv1 和 RIPv2 默认都启用自动路由汇总（auto-summary）功能，且 RIPv1 只支持自动汇总功能。自动汇总是把会汇总各子网路由的子网前缀到一个对应类型的标准网络边界，即自动汇总路由的子网掩码必须是 255.0.0.0、255.255.0.0、255.255.255.0 之类的。自动汇总**总是进行最大限度的汇总**，也就是把对应子网路由直接汇总到对应的标准网络，也就是到了可以汇总的边界。

如 172.16.1.0/24、172.16.2.0/24、172.16.3.0/24 和 172.16.4.0/24 这四个子网路由可以汇总成 172.16.0.0/16 这样一个标准的 B 类网络路由；如 192.168.1.0/26、192.168.64/26、192.168.128/26、

192.168.192/26 这四个子网路由就可以汇总成 192.168.1.0/24 这样一个标准的 C 类网络路由。

- 手动汇总

手动汇总仅 RIPv2 支持，因为它支持 CIDR（无类别域间路由）。手动汇总是基于路由器接口配置的，可以把对应接口上接收到的多个子网路由表项汇总成一个大的标准网络或者大的子网路由。它与自动汇总最明显的区别就是，它的汇总后的路由地址前缀可以不是标准网络的，也就是汇总路由的子网掩码不一定非要是 255.0.0.0、255.255.0.0、255.255.255.0。

【注意】这里有一个非常重要的地方要注意：在路由器全局配置的自动汇总功能，每个子网只能汇总到它所对应类别的标准网络上，如 A 类子网 IP 地址，则汇总后的汇总地址掩码只能是 255.0.0.0；B 类子网 IP 地址，则汇总后的汇总地址掩码只能是 255.255.0.0；C 类子网 IP 地址，则汇总后的汇总地址掩码只能是 255.255.255.0。但是如果是在接口上通过 **ip summary-address rip ip-address ip-network-mask** 命令配置的路由汇总，则汇总地址掩码只要是标准网络的都可以，而不一定是子网路由 IP 地址所对应类别的标准网络。如果是 A 类子网 IP 地址，则汇总后的汇总地址掩码可以是 A 类地址掩码，也可以是 B 类地址掩码，还可以是 C 类地址掩码；如果是 B 类子网 IP 地址，则汇总后的汇总地址掩码可以是 B 类地址掩码，也可以是 C 类地址掩码；如果是 C 类子网 IP 地址，则汇总后的汇总地址掩码只能是 C 类地址掩码。

当 RIP 协议确定在 RIP 路由数据库中需要一个汇总地址时，就会在 RIP 路由数据库中创建一个汇总条目。只要在汇总地址中有子路由，则这个汇总地址就会一直保留在路由数据库中。当最后一个子路由从数据库中删除时，则这个汇总条目也将从数据库中删除。这种处理数据库条目的方法可以减少在数据库中的条目数量，因为每个子路由不会在实际使用的路表条目中列出，而当不再有相关的子路由存在时，这个汇总的路由条目也将删除。

RIPv2 路由汇总需要代表最佳路由的最低度量值，或者直接采用汇总路由所代表的所有子路由中的最小的度量值。汇总路由度量值在初始化时或者路由更新通告时对指定路由进行度量编辑时计算的，不是在汇总路由通告时计算的。

2. RIP 路由汇总配置步骤

路由汇总功能可以基于全局启动自动汇总功能，还可以基于路由器接口启用自动汇总功能，并指定要汇总的子网路由。但无论是全局，还是接口，它们默认都是启用了自动汇总功能的。如果同时配置，则对于接口来说，最终的路由汇总功能是使用在对应接口上的汇总配置，否则直接采用全局配置。具体的配置方法如表 8-3 所示。

表 8-3 RIP 路由汇总配置步骤

步骤	命令	说明
1	enable 例如： Router> enable	进入特权模式。如果配置了特权模式密码，则要按提示输入正确的特权模式密码才能进入
2	configure terminal 例如： Router# configure terminal	进入全局配置模式
3	interface type number 例如： Router(config)# interface Ethernet 3/0	指定要启用路由汇总的接口，进入接口配置模式
4	ip summary-address rip ip-address ip-network-mask 例如： Router(config-if)# ip summary-address rip 10.2.0.0 255.255.0.0	（可选）在以上接口下启用手动汇总功能，并指定手动汇总规则。命令中的参数说明如下： <ul style="list-style-type: none"> ● ip-address：被汇总的汇总路由地址 ● ip-network-mask：汇总路由子网掩码。这个子网掩码必须大于或等于对应标准网络的掩码（也就是前缀长度一定要大于或等于对应的标准网络的前缀长度），而不能是超网掩码 可用 no ip summary-address rip ip-address ip-network-mask 命令取消对对应子网路由的汇总 示例中是指定要把 10.2.0.0 网络汇总成一个标准的 B 类网络路由

续表

步骤	命令	说明
5	exit 例如： Router(config-if)# exit	退出接口配置模式
6	router rip 例如： Router(config)# router rip	启用 RIP 路由协议，进入路由器配置
7	no auto-summary 例如： Router(config-router)# no auto-summary	(可选) 禁止自动汇总，默认是开启的。禁止后可用 auto-summary 命令恢复
8	end 例如： Router(config-router)# end	退出路由器配置模式，返回特权模式

可使用 **show ip protocols** 特权模式命令来查看接口上哪个路由被汇总了，也可以通过 **show ip rip database** 特权模式命令在 RIP 路由数据库中检测汇总地址。当用于汇总路由的最后一个子路由失效后，这个汇总路由也将从路由表中删除。

以下示例主网络为 10.0.0.0（从 **network** 命令宣告可以得出），如果采用以下配置，在 Ethernet1 接口上配置的汇总路由地址 10.2.0.0/16 将覆盖这个 10.0.0.0 自动汇总地址（注意，全局的自动汇总功能是默认开启的，所以一般不用另外通过 **auto-summary** 路由器配置模式命令开启），所以基于接口的汇总地址 10.2.0.0/16 将由 Ethernet1 接口被通告出去，而 10.0.0.0/8 自动汇总地址将不被通告。但是，如果启用了水平分割，则无论是自动汇总地址，还是以 **ip summary-address rip** 命令配置的接口汇总地址 **都不会被通告**。

```
Router(config)#interface Ethernet1
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# ip summary-address rip 10.2.0.0 255.255.0
Router(config-if)# exit
Router(config)#router rip
Router(config-router)# network 10.0.0.0
```

3. RIP 路由汇总配置注意事项

在配置 RIP 路由汇总时，要注意几个重要方面，否则最终可能并不能达到预期目的。

- 不汇总没有连接的子网

如果在汇总的子网中，有没有连接的子网，则建议在路由自动汇总中不要通告这个子网。当禁止了路由汇总功能，IOS 软件将通过有类别网络边界发送子网和主机路由信息。

- 汇总路由的地址前缀只能大于或等于对应标准网络地址前缀

这一点要特别注意，也是仅在 RIP 的汇总路由中所特有的。那就是 RIP 的手动汇总路由（自动汇总路由的地址前缀自动等于对应标准网络的地址前缀）的地址前缀只能大于或等于对应标准网络的地址前缀，也就是不能汇总成超网（Supernet）。通俗地讲是，A 类标准网络划分的子网路由最多只能汇总成 A 类标准网络路由，而不能汇总成 B 类，或者 A 类标准网络或子网路由；B 类标准网络划分的子网最多只能汇总成 B 类网络路由，而不能汇总成 A 类标准网络或子网路由；A 类网络划分的子网最多只能汇总成 A 类网络路由。

如你想要把 192.168.0.0/26、192.168.0.4/26、192.168.0.8/24、192.168.0.12/26 汇总成 192.168.0.0/22 的汇总路由，在 RIP 中是不行的，因为它的汇总路由地址前缀（22）小于对应的 C 类标准网络的地址前缀（24），否则会出现这样的错误提示，是说“汇总路由掩码必须大于或等于主网络”（也就是标准网络）。但你可以把以上这些子网汇总成地址前缀为 25，最多为 24 的标准 C 类网络。

```
Summary mask must be greater or equal to major net
```

同样你不能把 172.16.0.0/18、172.16.4.0/18、172.16.8.0/18、172.16.12.0/18 这四个子网汇总成地址前缀为 14 的路由，但你可以汇总成地址前缀为 16 或 17 的路由。同理，下面的 RIP 路由汇总配置是无效的，因为子网掩码 252.0.0.0 小于标准的 A 类网络掩码 255.0.0.0。

```
Router(config)#interface Ethernet 1
Router(config-if)#ip summary-address rip 10.0.0.0 252.0.0.0
```

- 接口上的每个路由汇总必须有唯一的主网络

每个接口上为一类子网进行的路由汇总，各汇总路由的主网络必须唯一，即使多个汇总路由的子网掩码是一样的。例如，下面的汇总是不允许的，因为在这里的 Ethernet 1 接口上配置的 10.2.2.0/16 汇总路由的主网络应该是 10.2.0.0，而不是 10.2.2.0。如果把第二条路由汇总配置中的主网络改成 10.2.0.0 255.255.0.0 就可以了。

```
Router(config)#interface Ethernet 1
Router(config-if)#ip summary-address rip 10.1.0.0 255.255.0.0
Router(config-if)#ip summary-address rip 10.2.2.0 255.255.0.0
```

8.2.5 源 IP 地址确认配置

默认情况下，IOS 软件会确认流入 RIP 路由更新信息的源 IP 地址。如果源 IP 地址无效，IOS 软件就会放弃这个路由更新。所以本节也属于可选配置任务，仅当要禁止路由器接口上流入的路由更新包中的源 IP 地址确认功能才需要进行。

确认源 IP 地址的目的是确保流入路由更新的源 IP 地址是与接收这个路由更新的路由器接口上的一个 IP 地址一致，也就是确认这个路由更新是从真实的路由器接口上接收来的，而不是伪装的。在接收流入路由更新的接口上禁止水平分割功能也会使 IOS 系统执行这个源地址确认。

但是如果有一个路由器处于离线状态，而你又希望在本地球器上接收这台路由器的路由更新，则你可能要在本地球器上禁止源 IP 地址确认功能。另外，对于无编号 IP 的接口（也就是此接口不配置自己的 IP 地址，但会借用其他接口的 IP 地址），也不执行源地址确认。

表 8-4 所示的是在接口上启用，以及全局禁用源 IP 地址确认的配置步骤（IOS 系统默认全局启用源 IP 地址确认功能）。

表 8-4 在接口上启用或全局禁用源 IP 地址确认功能的配置步骤

步骤	命令	说明
以下是在具体路由器接口上启用源 IP 地址确认功能的配置步骤，因为系统默认是全局启用该功能的，所以只需要在对应接口上启用水平分割功能即可。		
1	enable 例如： Router> enable	进入特权模式。如果配置了特权模式密码，则按提示输入正确的特权模式密码才能进入
2	configure terminal 例如： Router# configure terminal	进入全局配置模式
3	interface type number 例如： Router(config)# interface Ethernet 3/0	指定要禁止源 IP 地址确认的接口，进入接口配置模式
4	ip split-horizon 例如： Router(config-if)# ip split-horizon	在上述接口上启用水平分割功能，相当于同时在以上接口上启用了源 IP 地址确认功能
下面是全局禁用源 IP 地址确认功能的配置步骤		
1	configure terminal 例如： Router# configure terminal	进入全局配置模式
2	router rip 例如： Router(config)# router rip	进入路由器配置模式

续表

步骤	命令	说明
3	no validate-update-source 例如： Router(config-router)# no validate-update-source	全局禁止对流入的 RIP 路由更新包的源 IP 地址确认
4	end 例如： Router(config-router)# end	退出路由器配置模式，返回特权模式

以下示例是在 RIP 路由器上全局禁用对流入的 RIP 路由更新的源 IP 地址确认功能。

```
Router# configure terminal
Router(config)#router rip
Router(config-router)#network 10.105.0.0
Router(config-router)#no validate-update-source
```

8.2.6 更新包间延时配置

本节也属选配任务，仅在当高性能路由器与低性能路由器连接时，在高性能路由器上进行配置。

默认情况下，在多个 RIP 更新包的发送过程中各个包之间是不插入延时的。如果更新包是从高性能路由器发送到低性能路由器上，你可能要高性能 RIP 路由器上，在 RIP 多个更新包发送过程中要插入包间延时。具体的配置步骤如表 8-5 所示。

表 8-5 RIP 路由更新包发送延迟配置步骤

步骤	命令	说明
1	enable 例如： Router> enable	进入特权模式。如果配置了特权模式密码，则应按提示输入正确的特权模式密码才能进入
2	configure terminal 例如： Router# configure terminal	进入全局配置模式
3	router rip 例如： Router(config)# router rip	进入路由器配置模式
4	output-delay milliseconds 例如： Router(config-router)# output-delay 8	为输出 RIP 更新包配置包间延时。参数 <i>milliseconds</i> 的取值范围为 8~50 毫秒，默认值为 0，即不插入延时 要删除原来所做的延时设置，则可用 no output-delay 路由器配置模式命令
5	end 例如： Router(config-router)# end	退出路由器配置模式，返回特权模式

以下示例是设置包间发送间隔为 10 这秒。

```
Router(config)#router rip
Router(config-router)#output-delay 10
```

8.2.7 通过帧中继网络连接的路由器的 IP-RIP 延时开始配置

本节也是选配任务，仅在 Cisco 路由器通过帧中继网络与非 Cisco 路由器进行 RIPv2 MD5 认证时才需要配置。但由于目前帧中继技术已基本被淘汰了，所以本节可以不学习。

“IP-RIP 延时开始”（IP-RIP Delay Start）功能用于 Cisco 路由延时启动 RIPv2 邻居会话，直到与邻居路由器之间的网络连通性完全正常，从而确保发送到非 Cisco 邻居路由器的 MD5 包序列号

为 0。默认情况下，只要物理接口启用了，路由器就可以与使用 MD5 认证的邻居路由器建立 RIPv2 会话，发送 MD5 包。

当一台 Cisco 路由器与一台非 Cisco 路由器通过帧中继网络，使用 MD5 认证建立 RIPv2 邻居关系时，要经常用到 IP-RIP 延时启动功能的。当 RIPv2 邻居路由器通过帧中继网络连接时，可能出现串行接口在帧中继电路已准备好传输和接收数据前就已激活。当串行接口已激活了，但帧中继电路还没有时，则任何想通过这个串行接口传输的 MD5 认证路由更新包都将被丢弃。这样当帧中继电路已准备好后再传输 MD5 包，包的序列号就不再是初始的 0 了，而问题是有些非 Cisco 路由器不允许第一个接收的 MD5 包的序列号不是 0，而中断 RIPv2 认证，所以在 Cisco 路由器上才有本节所介绍的“IP-RIP 延时开始”功能，只有等帧中继电路已准备好才发送第一个 MD5 包，这样就可以保证对方非 Cisco 路由器接收到的第一个 MD5 包的序列号为 0。

【注意】 Cisco 路由器就允许从其他路由器接收到的第一个 MD5 认证包的序号大于 0。如果你的网络中仅使用了 Cisco 路由器，不需要使用 IP-RIP 延时启动功能。

另外，延时启动功能在其他类型接口，如快速以太网和千兆以太网接口。如果你的路由器不能使用 MD5 认证与非 Cisco 设备建立 RIPv2 会话，则这个延时启动功能可能可以解决这个问题。但你的 Cisco 路由器必须使用 12.4 (12) 以上版本的 IOS 系统。而且这个延时启动功能仅当你的 Cisco 路由器配置用来建立一个与非 Cisco 设备 RIPv2 邻居关系，并且使用 MD5 进行邻居认证时才需要使用。

配置延时启动功能需要完成以下三方面的配置：

- 配置 RIPv2 协议
- 在串行子接口上配置帧中继
- 配置 RIPv2 的 MD5 认证和在串行子接口上的 IP-RIP 延时启动功能

1. 配置 RIPv2 协议

因为 MD5 认证只有 RIPv2 版本才支持，所以要配置 IP-RIP 延时启动功能必须先在对应的 Cisco 路由器上启用 RIPv2 版本协议。RIPv2 的具体配置步骤参见 8.2.2 节。

2. 在串行子接口上配置帧中继

本节其实介绍的是如何为帧中继网络配置一个串行子接口。但此处介绍的方法仅是许多种在一个子接口上配置帧中继方案中的一种，具体步骤如表 8-6 所示。

表 8-6 为帧中继网络配置串行子接口的步骤

步骤	命令	说明
1	enable 例如： Router> enable	进入特权模式。如果配置了特权模式密码，则要按照提示输入正确的特权模式密码才能进入
2	configure terminal 例如： Router# configure terminal	进入全局配置模式
3	interface type number 例如： Router (config)# interface serial3/0	指定要配置帧中继的接口，进入接口配置模式
4	no ip address 例如： Router (config-if)# no ip address	删除接口上当前配置的 IP 地址
5	encapsulation frame-relay mfr number [name] 例如： Router(config-if)# encapsulation frame-relay ietf	指定上述接口的帧中继封装类型，具体参见本节下面介绍
6	frame-relay lmi-type {cisco ansi q933a} 例如： Router(config-if)# frame-relay lmi-type ansi	指定上述接口帧中继本地管理接口（local management interface, LMI）类型，具体参见本节下面介绍

续表

步骤	命令	说明
7	exit 例如： Router(config-if)# exit	退出接口配置模式
8	interface type number/subinterface-number {point-to-point multipoint} 例如： Router(config)# interface serial3/0.1 point-to-point	指定要配置帧中继的子接口和连接类型（ point-to-point 为点对点类型， multipoint 为点对多点类型），进入子接口配置模式
9	frame-relay interface-dlci dlci [ietf cisco] 例如： Router(config-subif)# frame-relay interface-dlci 100 ietf	为上子接口分配一个 DLCI（Data-Link Connection Identifier，数据链路连接标识符），具体参见本节下面介绍

1) encapsulation frame-relay 命令。

encapsulation frame-relay mfr number [name]接口配置模式命令用来创建一条多链路帧中继（Multilink Frame Relay, MFR）捆绑链路（bundle link），并指定一个捆绑（bundle）名。可用最前面带 **no** 关键字选项的该命令从捆绑中删除捆绑定的链路。

捆绑与捆绑链路是多链路帧中继的两个基本概念。一个 MFR 接口对应一个捆绑，一个捆绑链路可以包含多个被捆绑的链路，一个被捆绑的链路对应着一个物理接口。捆绑对它的被捆绑链路进行管理。MFR 接口是逻辑接口，多个物理接口可以捆绑成一个 MFR 接口。对捆绑定和捆绑链路的配置实际上是对 MFR 接口和物理接口的配置。MFR 接口的功能和配置与普通的 FR（帧中继）接口相同，也支持 DTE、DCE 接口类型。

命令中的参数说明如下：

- **number**: 指定对应物理接口被捆绑的 MFR 接口号。
- **name**: 可选参数，指定对应物理接口的捆绑链路标识（Bundle link identification, LID）名称，可以是最多 49 个字符长度。默认是物理接口名称。这个名称将用于对端设备标识捆绑链路，使得对端设备可以确定哪个捆绑链路与哪个捆绑相关联。LID 名称可以在被捆绑的物理接口上通过 **frame-relay multilink lid** 命令进行分配或者修改。

以下示例显示 s0 接口作为名为 mfr 0 接口与名为 BL1 的捆绑链路的关联接口。

```
Router(config)#interface serial 0
Router(config-if)#encapsulation frame-relay mfr 0 BL1
```

2) frame-relay lmi-type 命令。

frame-relay lmi-type {ansi | cisco | q933a}接口配置模式命令用来选择本地管理接口（Local Management Interface, LMI）类型。要返回到默认的 LMI 类型，则要用最前面带 **no** 关键字选项的该命令。命令中的可选项说明如下：

- **ansi**: 多选一选项，指定采用由 ANSI 标准定义的 LMI 类型。
- **cisco**: 多选一选项，指定采用由 Cisco 和其他三个公司共同定义的 LMI 类型。
- **q933a**: 多选一选项，指定采用由 ITU-T Q.933 标准中的 LMI 类型。

默认情况下，LMI 自动感知功能是被激活的，可通过与交换机的通话检测 LMI 类型。

以下示例配置的是定义 s1 接口的 LMI 类型为 ANSI 标准定义的 LMI 类型。

```
Router(config)#interface Serial1
Router(config-if)#encapsulation frame-relay
Router(config-if)#frame-relay lmi-type ansi
Router(config-if)#keepalive 15
```

3) frame-relay interface-dlci 命令。

frame-relay interface-dlci dlci [ietf | cisco]接口配置模式命令用来分配一个数据链路控制标识符（DLCI）到一个指定的路由器或者接入服务器上的帧中继子接口，并指定所采用的帧中继封装类型。要删除这个配置，可用最前面带 **no** 关键字选项的该命令。

命令中的可选项说明如下：

- **dldci**: 分配一个在指定子接口上使用的 DLCI 号。
- **ietf**: 二选一可选项, 指定采用 IETF 定义的帧中继封装类型。
- **cisco**: 二选一可选项, 指定采用 Cisco 定义的帧中继封装类型。

以下示例显示分配 DLCI 100 到 S5.17 子接口上。

```
Router(config)# interface serial 5
Router(config-if)# interface serial 5.17
Router(config-subif)# frame-relay interface-dlci 100
```

以下示例是分配 DLCI 100 到一个串行接口。

```
Router(config)# interface serial 1/1
Router(config-if)# frame-relay interface-dlci 100
```

3. 配置 RIPv2 的 MD5 认证和帧中继子接口的 IP-RIP 延时开始功能

在 Cisco 路由器上配置 MD5 RIPv2 认证和帧中继子接口上的 IP-RIP 延时开始功能的具体配置步骤如表 8-7 所示。

表 8-7 配置 RIPv2 的 MD5 认证和帧中继子接口的 IP-RIP 延时开始功能的步骤

步骤	命令	说明
1	enable 例如: Router> enable	进入特权模式。如果配置了特权模式密码, 则按提示输入正确的特权模式密码才能进入
2	configure terminal 例如: Router# configure terminal	进入全局配置模式
3	key chain name-of-chain 例如: Router(config)# key chain rip-md5	创建一个认证密钥链, 进入密钥链配置模式。一个密钥链可以有 2,147,483,647 个密钥
4	key number 例如: Router(config-keychain)# key 123456	指定密钥标识符, 进入密钥链配置模式。密钥链标识符的取值范围为 0~2147483647
5	key-string string 例如: Router(config-keychain-key)# key-string abcde	配置密钥字符串。可以最多为 80 个 ASCII 字符, 但首位不能是数字。它与上面的密 ID 是一一对应的关系, 即一个密 ID 要有一个密钥字符串
6	exit 例如: Router(config-keychain-key)# exit	退出密钥链密钥配置模式
7	exit 例如: Router(config-keychain)# exit	退出密钥链配置模式
8	interface type number/subinterface-number 例如: Router(config)# interface serial3/0.1	指定要配置 RIP 认证的帧中继子接口, 进入接口配置模式
9	no cdp enable 例如: Router(config-subif)# no cdp enable	在上述子接口上禁止 CDP (Cisco Discovery Protocol, 思科发现协议)。CDP 协议在非 Cisco 设备中是不支持, 在你要连接非 Cisco 路由器时, 是需要 IP-RIP 延时启动功能的。所以你应当在所有要配置 IP-RIP 延时启动功能的接口上禁用 CDP
10	ip address ip-address subnet-mask 例如: Router (config-subif)# ip address 172.16.10.1 255.255.255.0	为帧中继子接口配置 IP 地址
11	ip rip authentication mode {text md5} 例如: Router(config-subif)# ip rip authentication mode md5	指定 RIPv2 认证模式
12	ip rip authentication key-chain name-of-chain 例如: Router (config-subif)# ip rip authentication key-chain rip-md5	指定先前配置的, 用于 RIPv2 MD5 认证的密钥链
13	ip rip initial-delay delay 例如: Router(config-subif)# ip rip initial-delay 45	在子接口上配置 IP-RIP 延时开始功能, 指定初始延迟时间 (取值范围为 0~1800 秒)。路由器将按本命令中所配置的延迟值, 延迟发送第一个 MD5 认证包到 RIPv2 邻居路由器
14	end 例如: Router(config-subif)# end	退出子接口配置模式, 返回到特权模式

以上配置步骤中所涉及的主要的命令就是——**ip rip initial-delay delay** 接口或者子接口配置模式命令。它是用来延时与采用 MD5 认证的 RIPv2 邻居的会话，直到邻居路由完全正常工作，并具有好的连通性。命令中的 *delay* 参数用来设置在发送第一个 MD5 认证包到 RIPv2 邻居前要等待的时间。要删除这个延时配置，可用 **no ip rip initial-delay** 命令。

以下示例显示了配置一个路由器在发送第一个 MD5 认证包到一个非思科设备前要等待 45 秒。

```
Router(config)# interface POS 0/1/0
Router(config-subif)# ip rip initial-delay 45
```

以下示例包含了配置 IP-RIP 延时启动功能配置的基本命令。

```
Router#configure terminal
Router(config)#key chain rip-md5  !---创建用于 MD5 认证的密钥链
Router(config-keychain)#key 123456  !---定义一个 ID 号为 123456 的密钥
Router(config-keychain-key)#key-string abcde  !---指定 ID 号为 123456 的密钥的密钥字符串为 abcde
Router(config-keychain-key)#exit
Router(config-keychain)#exit
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#network 172.16.0.0
Router(config-router)#no auto-summary  !---禁止自动路由汇总
Router(config-router)#exit
Router(config)#interface Serial3/0
Router(config-if)#no ip address
Router(config-if)#encapsulation frame-relay ietf  !---指定采用 IETF 定义的帧中继封装方式
Router(config-if)#frame-relay lmi-type ansi  !---指定采用 ANSI 定义的 LMI 类型
Router(config-if)#exit
Router(config)#interface Serial3/0.1 point-to-point  !---为 Serial3/0.1 子接口封装 PPP 协议
Router(config-if)#ip address 172.16.10.1 255.255.255.0
Router(config-if)#ip rip initial-delay 45  !---指定以上子接口发起第一个 MD5 认证包的延时时间为 45 秒
Router(config-if)#ip rip authentication mode md5  !---指定 RIP 邻居间采用 MD5 认证模式
Router(config-if)#ip rip authentication key-chain rip-md5  !---指定 MD5 认证中所采用的密钥为前面创建的 rip-md5
Router(config-if)#frame-relay interface-dlci 100  !---为以上子接口分配一个数据链路控制标识符（DLCI）——100
Router(config-if)#end
```

8.2.8 RIP 路由管理

RIP 也是一种相对比较简单动态路由协议，不仅它的功能配置相对也比较简单，管理方法也比较简单。除了我们在前面，以及在第 7 章介绍静态路由管理时经常用到的 **show ip route** 管理命令外（要单独查看 RIP 路由信息，可用 **show ip route rip** 命令），在 RIP 路由中，还有 **show ip rip neighbors**（查看 RIP 邻居路由器）和 **show ip rip database**（查看 RIP 路由数据库）这两个主要的管理命令。另外还有一条也可用于 RIP 管理的 **show** 命令，那就是 **show ip protocols**。它们都可以在特权模式下执行，下面分别予以具体介绍。

1. show ip rip neighbors 命令

show ip rip neighbors 命令是没有任何参数和选项的命令，用于查看已建立双向转发检测（Bidirectional Forwarding Detection, BFD）的 RIP 邻居路由器。

下面是一个执行该命令后的输出示例，从输出信息中可以见到有两个已与本地路由器建立了 BFD 的邻居路由器（Neighbor，本示例中这两个邻居路由器与本地路由器连接的接口 IP 地址分别为 10.10.10.2 和 10.10.20.2），还可从中了解本地路由器与这两个邻居路由器间是通过哪个接口（Interface）连接的，以及与这两个邻居路由器间各自建立的会话号（SessionHandle）。

```
Router# show ip rip neighbors
BFD sessions created for the RIP neighbors
Neighbor      Interface      SessionHandle
```

```
10.10.10.2   Ethernet0/0   1
10.10.20.2  Ethernet1/0   2
```

2. show ip rip database 命令

show ip rip database [*ip-address mask*]命令用来查看 RIP 路由数据库中包括汇总路由在内的所有，或者指定路由条目。两个可选参数（必须同时指出）说明如下：

- *ip-address*: 指定要查看的路由条目的 IP 地址；
- *mask*: 指定要查看的路由条目 IP 地址所对应的子网掩码。

【注意】 汇总地址仅当相关的子路由被汇总时才会出现在 **show ip rip database** 命令的输出信息中。如果汇总地址中的最后一条子路由无效，则汇总地址也将从路由表中被清除。

另外，仅当在 RIP 路由器上使用 **ip rip triggered** 接口配置模式命令启用了触发扩展才会有 RIP 私有的路由数据库。

以下示例是显示当前路由器 RIP 路由数据库的路由条目。从中可以看出它包括了三条汇总路由：10.0.0.0/8 和 10.1.0.0/8 为自动汇总条目，10.11.0.0/16 为接口汇总条目。另外还包括四条直连路由。

```
Router# show ip rip database
10.0.0.0/8   auto-summary
10.11.11.0/24  directly connected, Ethernet2
10.1.0.0/8   auto-summary
10.11.0.0/16  int-summary
10.11.10.0/24  directly connected, Ethernet3
10.11.11.0/24  directly connected, Ethernet4
10.11.12.0/24  directly connected, Ethernet5
```

下面的示例是指定了路由地址和子掩码的 **show ip rip database** 命令输出。从输出信息中可以得知目的地址为 172.19.86.0/24 的这条 RIP 路由通过两个接口学习到了，一个是通过 Serial0 接口从 IP 地址为 172.19.67.38 的邻居路由器上学习得到，另一个是通过 Serial1 接口从 IP 地址为 172.19.70.36 的邻居路由器上学习得到。

```
Router# show ip rip database 172.19.86.0 255.255.255.0
172.19.86.0/24
[1] via 172.19.67.38, 00:00:25, Serial0
[2] via 172.19.70.36, 00:00:14, Serial1
```

3. show ip protocols 命令

可以使用命令校验路由器接口上的路由汇总（但要注意的是，该命令显示的不仅指 RIP 路由汇总）。如以下示例是查看到当前路由器上运行的动态路由协议，各路由器接口上运行的协议版本，以及路由汇总信息。从输出中可以看出当前路由器上运行的是 RIP 路由协议，并且显示自动汇总是关闭的，而在 Ethernet2 接口上配置了一条汇总路由 12.11.0.0/16（均以粗体字显示）。

```
Router# show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 8 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface        Send  Recv  Triggered RIP  Key-chain
  Ethernet2         2     2
  Ethernet3         2     2
  Ethernet4         2     2
  Ethernet5         2     2
Automatic network summarization is not in effect
Address Summarization:
  12.11.0.0/16 for Ethernet2
```

你可以继续利用前面介绍的 `show ip rip database` 命令查看具体的汇总路由条目。

8.3 RIP 路由配置示例

RIP 协议虽然比较简单，但有些功能技术原理和配置仍还是比较难以理解。为了加深大家对以上 RIP 功能和应用配置方法的理解，本节给出几个对应的综合配置示例。

8.3.1 RIP 路由基本配置示例

本示例拓扑结构如图 8-16 所示，各路由器接口分配的 IP 地址也在图中作了标注。现要求通过 RIP 动态路由协议实现各路由器的互通。

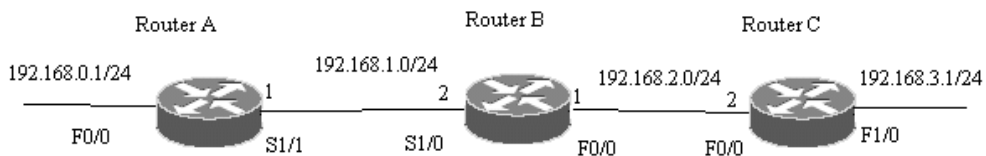


图 8-16 RIP 路由基本配置示例拓扑结构

下面是具体的配置步骤：

(1) 按照图中所标注的接口 IP 地址，配置各路由器的接口 IP 地址（略），配置方法可参见第 3 章相关内容。

(2) 先使用 `show ip route` 命令查看一下各路由器当前的路由表。结果证实各路由器当前仅有各自直接连接的路由。下面是 Router A 上当前的路由表项（**注意输出信息中的粗体字部分**）。

```
RouterA#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

C    192.168.0.0/24    is directly connected,FastEthernet0/0
C    192.168.1.0/24    is directly connected,Serial1/1
```

(3) 在 Router A 上 ping Router C，结果显示不通，输出信息如下所示。

```
RouterA#ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) :56 data bytes
.....
----192.168.2.2 ping statistics----
5 packets transmitted,0 packets received,100% packet loss
```

(4) 在 Router A 上启用 RIP 路由协议，并宣告其直接连接的网络。

```
RouterA>enable
RouterA#configure terminal
RouterA(config)#router rip
RouterA(config-router)#network 192.168.0.0
RouterA(config-router)#network 192.168.1.0
RouterA(config-router)#end
```

现在再来查看 Router A 上的路由表，结果显示仍是前面第 2 步中所显示的两条直连路由。因为其他路由器上没有启用 RIP 协议，所以它根本无法学习到来自其他路由器的路由。



经验之谈

在 RIP 路由器上通过 **network** 命令宣告自己所连接的网络并不是为了本地路由自身，因为这些网络本来就是与本地路由器直接连接的，在路由表中已有优先级别最高的对应直连路由，不会在本地路由器上以 RIP 路由表项显示本地路由器上直连的这些网络。通过 **network** 命令宣告这些直连网络的目的就是告诉本地路由器可以把这些网络的路由信息通过 RIP 协议向邻居 RIP 路由器进行通告，使邻居路由器通过 RIP 协议学习到这些网络。

(5) 在 Router B 上启用 RIP 路由协议，并宣告其直接连接的网络。

```
RouterB>enable
RouterB#configure terminal
RouterB(config)#router rip
RouterB(config-router)#network 192.168.1.0
RouterB(config-router)#network 192.168.2.0
RouterB(config-router)#end
```

现在来查看 Router B 上的路由表，此时可以看到它已学习到了 Router A 路由器上所连接的网络，有一个 RIP 路由。具体如下（**注意输出信息中的粗体字部分**）：

```
RouterB#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

R   192.168.0.0/24      [120/1] via 192.168.1.1, 00:00:25, Serial1/0
C   192.168.1.0/24      is directly connected, Serial1/0
C   192.168.2.0/24      is directly connected, FastEthernet0/0
```

【说明】“R 192.168.0.0/24 [120/1] via 192.168.1.1, 00:00:25, Serial1/0” RIP 路由项说明如下（后面的 RIP 路由按照同样的方法来理解）：

- R: 说明该路由条目是通过 RIP 路由协议学习来的；
- 192.168.0.0/24: 这是路由条目的目的网络地址；
- 120: 这是 RIP 路由协议的默认管理距离；
- 1: 指从路由器 Router B 到达目的网络 192.168.0.0/24 的度量值为 1 跳，因为只经过一个路由器；
- Via 192.168.1.1: 这是该 RIP 路由的下一跳 IP 地址；
- 00:00:025: 这是指明距离下一次更新还有 5（30-25）秒；
- Serial1/0: 指出接收该 RIP 路由条目的本地路由器接口为 s1/0。



经验之谈

细心的读者可能会发现，Router A 路由器中明明宣告了 192.168.0.0/24 和 192.168.1.0/24 这两个网络，为什么 Router B 却只学习到了 192.168.0.0/24 这一条 RIP 路由呢？那是因为 192.168.1.0/24 网络也是与 Router B 直接连接的，在 Router B 中也当成了直连路由。同时，因为直连路由的优先级是最高的，比 RIP 路由的优先级就更高了，所以在路由表中只显示了 192.168.1.0/24 网络的直连路由。如以上输出信息中的“C 192.168.1.0/24 is directly connected, Serial1/0”就是指出了这条直连路由。

另外，在路由器上通过 **router rip** 命令启用 RIP 进程的用途是为了使本地路由器可以通过 RIP

动态路由协议学习没有直接连接本地路由器上，位于其他已启用 RIP 进程的路由器上，并且已使用 **network** 命令宣告的网络。

(6) 在 Router C 上启用 RIP 协议，并宣告其直接连接的网络。

```
RouterC>enable
RouterC#configure termina
RouterC(config)#router rip
RouterC(config-router)#network 192.168.2.0
RouterC(config-router)#network 192.168.3.0
RouterC(config-router)#end
```

此时再来查看 Router C 上的路由表，可以看到它从 Router A 和 Router B 上学习到两条 RIP 路由。具体如下（**注意输出信息中的粗体字部分**）。至于为什么原来在 Router B 中宣告的 192.168.2.0/24 的这条路由没有以 RIP 路由显示，原因同上，因为它同样是 Router C 上的直连路由，已以“C 192.168.2.0/24 is directly connected, FastEthernet0/0”直连路由显示了。

```
RouterC#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

R   192.168.0.0/24    [120/2] via 192.168.2.1, 00:00:25, FastEthernet0/0
R   192.168.1.0/24    [120/1] via 192.168.2.1, 00:00:25, FastEthernet0/0
C   192.168.2.0/24    is directly connected, FastEthernet0/0
C   192.168.3.0/24    is directly connected, FastEthernet1/0
```

(7) 现在再次查看 Router A 和 Router B 上的路由表，可以得到如下结果（**注意输出信息中的粗体字部分**），显示它们相互从其他两个路由器上学习到了另外两路由上所连接的网段路由。

```
RouterA#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

C   192.168.0.0/24    is directly connected, FastEthernet0/0
C   192.168.1.0/24    is directly connected, Serial1/1
R   192.168.2.0/24    [120/1] via 192.168.1.2, 00:00:25, Serial1/1
R   192.168.3.0/24    [120/2] via 192.168.1.2, 00:00:25, Serial1/1

RouterB#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

R   192.168.0.0/24    [120/1] via 192.168.1.1, 00:00:25, Serial1/0
C   192.168.1.0/24    is directly connected, Serial1/0
C   192.168.2.0/24    is directly connected, FastEthernet0/0
R   192.168.3.0/24    [120/1] via 192.168.2.2, 00:00:25, FastEthernet0/0
```

(8) 最后同样进行 ping 操作来验证路由器之间是否已连通。结果显示都是通的，证明 RIP 路

由配置是正确的。在此就不具体举例了。

8.3.2 RIP 水平分割功能配置示例

在如图 8-17 所示的示例中，描述了连接帧中继网络的 Router C 通过一个串行接口（S0）连接两个 IP 子网，其中一个是通过该接口的辅助 IP 地址连接的。

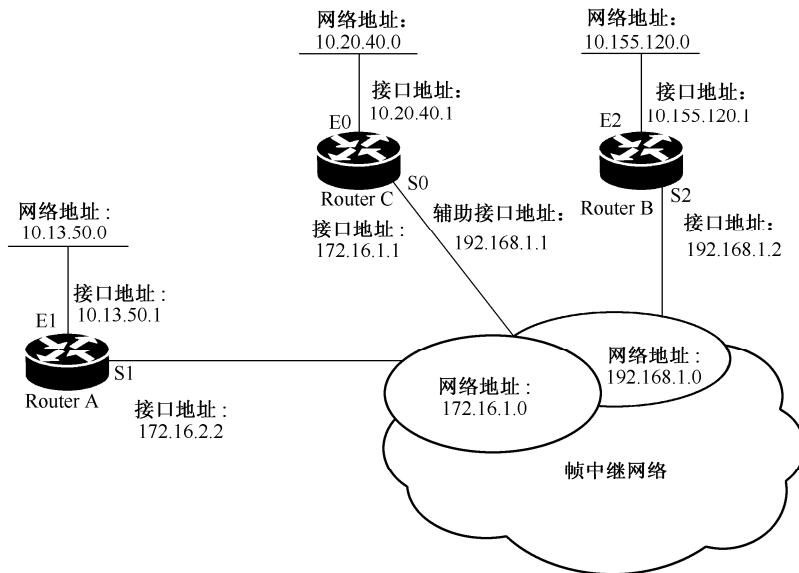


图 8-17 RIP 水平分割配置示例拓扑结构

示例中的 Router A、Router B 和 Router C 的以太网接口（分别连接的网路是 10.13.50.0、10.155.120.0 和 10.20.40.0）都默认启用水平分割功能，而 Router C 上连接到 172.16.1.0 和 192.168.1.0 两个子网的串行接口 S0 用 **no ip split-horizon** 命令禁止了水平分割功能，这样在这个接口上就可以基于子网路由更新了。

在本示例中，水平分割功能在所有串行接口上是禁止的，而且在 Router C 上必须禁止水平分割功能，以便 172.16.0.0 网络可以向 192.16.0.0 网络进行通告，反之亦然。这两个子网是在路由器的 s0 接口交迭的。如果在路由器 C 的 S0 接口上启用了水平分割功能，则它不能向 172.16.0.0 和 192.16.0.0 这两个帧中继网络通告路由。

路由器 A 上的配置：

```
RouterA>enable
RouterA# configure terminal
RouterA(config)#interface ethernet 1
RouterA(config-if)#ip address 10.13.50.1
RouterA(config-if)#exit
RouterA(config)#interface serial 1
RouterA(config-if)#ip address 172.16.2.2
RouterA(config-if)#encapsulation frame-relay !---在 s1 串口上封装帧中继协议
RouterA(config-if)#no ip split-horizon !---在 s1 串口上禁止水平分割功能
```

路由器 B 上的配置：

```
RouterB>enable
RouterB# configure terminal
RouterB(config)#interface ethernet 2
RouterB(config-if)#ip address 10.155.120.1
RouterB(config-if)#exit
RouterB(config)#interface serial 2
```

```
RouterB(config-if)#ip address 192.168.1.2
RouterB(config-if)#encapsulation frame-relay
RouterB(config-if)#no ip split-horizon
```

路由器 C 上的配置：

```
RouterC>enable
RouterC# configure terminal
RouterC(config)#interface ethernet 0
RouterC(config-if)#ip address 10.20.40.1
RouterC(config-if)#exit
RouterC(config)#interface serial 0
RouterC(config-if)#ip address 172.16.1.1
RouterC(config-if)#ip address 192.168.1.1 secondary
RouterC(config-if)#encapsulation frame-relay
RouterC(config-if)#no ip split-horizon
```



经验之谈

通常，不建议改变默认状态，除非确认你的应用需求必须要改变默认设置，以便正确地通告路由。记住，如果在一个串行接口（或者连接到一个包交换网络上接口）上禁止通了水平分割功能，你必须在网络中所有路由器的任何多播组中禁止水平分割功能。

8.3.3 RIP 路由更新认证配置示例

本节将通过一个具体的示例介绍 Cisco 路由器上的 RIP 路由更新认证的具体配置和应用方法，但要注意的是仅配置支持 RIPv2 的 Cisco 路由器才可以配置 RIP 路由更新认证。

Cisco RIPv2 路由器支持两种 RIP 路由更新认证：纯文本认证和 MD5 认证。纯文认证是默认的认证方式，但在存在安全隐患时建议不要采用纯文本认证，因为纯文本认证方式中的密钥将在每个发送的 RIPv2 路由更新包中以明文显示，很容易被截取。

本示例拓扑结构如图 8-18 所示。很简单，就是两个支持 RIPv2 的路由器（RA 和 RB）相接相连，现要在这两个 RIP 路由器之间配置 RIP 认证。

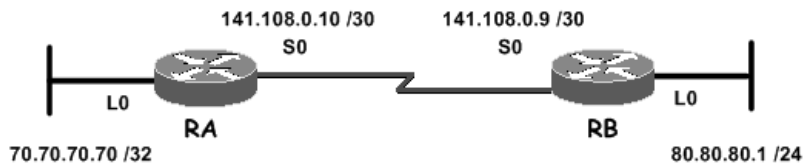


图 8-18 RIP 认证示例拓扑结构

RIP 认证的基本配置流程如下：

- 定义一个 RIP 认证密钥链；
- 为上面定义的密钥链配置一个或多个密钥（其实就是密钥 ID）；
- 为以上每个密钥 ID 配置具体的密钥字符串，它们是最终用于认证的密钥；
- 为各个需要 RIP 认证的路由器接口上启用 RIP 认证，并指定要使用的认证密钥链；
- 指定各个接口所采用的 RIP 认证模式，是纯文本认证，还是 MD5 认证。因为在启用了 RIP 认证后默认是采用纯文本认证，所以如果要采用纯文本认证，则本步可以不用配置。
- （可选）配置密钥管理。密钥管理是一种控制认证密钥的方法，用于从一个认证密钥迁移到另一个认证密钥。

下面分别介绍纯文本 RIP 认证和 MD5 RIP 认证的具体配置方法。

1. 纯文本 RIP 认证配置

RA 路由器上的配置（本端所配置的密钥 ID 和密钥字符串必须要与对端路由器上所配置的密钥 ID 和密钥字符串对应一致，但密钥链名称可以不一致）：

```
RA>enable
RA# configure terminal
RA(config)#key chain kal !---定义 RA 上的密钥链名称 kal，一个密钥链可以包括多个密钥，可以与对 RB 端路由器上配置的密钥链名称不一样
RA (config-keychain)#key 1 !---定义 kal 密钥链中的 1 号密钥，可以与对端 RB 路由器上配置的密钥号不一致（这一点与后面将要介绍的 MD5 认证要求是不同的）
RA (config-keychain-key)#key-string 234 !---定义 1 号密钥的密钥字符串为 234，必须与对端 RB 路由器上配置的密钥字符串一致
RA (config-keychain-key)#exit
RA (config-keychain)#exit
RA (config)#interface Loopback0
RA (config-if)#ip address 70.70.70.70 255.255.255.255 !---为 loopback0 环路接口指定 IP 地址，每个路由器通常必须配置一个环路接口 IP 地址，以此作为本路由器 ID，但它不是 RIP 认证所必须配置的
RA (config-if)#exit
RA (config)#interface Serial0
RA (config-if)#ip address 141.108.0.10 255.255.255.252
RA (config-if)#ip rip authentication key-chain kal !---在 Serial0 接口上启用 RIP 认证，并指定使用前面已创建名为 kal 的密钥链
RA (config-if)#exit
RA (config)#router rip
RA (config-router)#version 2
RA (config-router)#network 141.108.0.0
RA (config-router)#network 70.0.0.0
```

RB 路由器上的配置（本端所配置的密钥 ID 和密钥字符串必须要与对端路由器上所配置的密钥 ID 和密钥字符串对应一致，但密钥链名称可以不一致）：

```
RB>enable
RB# configure terminal
RB(config)#key chain kal !---定义 RB 上的密钥链名称 kal，一个密钥链可以包括多个密钥，可以与对 RA 端路由器上配置的密钥链名称不一样
RB (config-keychain)#key 1 !---定义 kal 密钥链中的 1 号密钥，可以与对端 RA 路由器上配置的密钥号不一致这一点与后面将要介绍的 MD5 认证要求是不同的
RB (config-keychain-key)#key-string 234 !---定义 1 号密钥的密钥字符串为 234，必须与对端 RA 路由器上配置的密钥字符串一致
RB (config-keychain-key)#exit
RB (config-keychain)#exit
RB (config)#interface Loopback0
RB (config-if)#ip address 80.80.80.1 255.255.255.0 !---为 loopback0 环路接口指定 IP 地址，每个路由器通常必须配置一个环路接口 IP 地址，以此作为本路由器 ID，但它不是 RIP 认证所必须配置的
RB (config-if)#exit
RB (config)#interface Serial0
RB (config-if)#ip address 141.108.0.9 255.255.255.252
RB (config-if)#ip rip authentication key-chain kal !---在 Serial0 接口上启用 RIP 认证，并指定使用前面已创建名为 kal 的密钥链
RB (config-if)#exit
RB (config)#router rip
RB (config-router)#version 2
RB (config-router)#network 141.108.0.0
RB (config-router)#network 80.0.0.0
```

2. 验证纯文本 RIP 认证

下面通过观察 `debug ip rip` 调试命令和 `show ip route` 命令的输出信息校验以上的纯文本 RIP 认证工作正常。要正常工作，所有 RA 与 RB 之间交换的路由信息在接受前必须经过认证。

以下是在 RB 上执行 `debug ip rip` 特权模式命令的输出，从中可以看出在正式接受来自 RA 的路由更新包前经过了纯文本认证（输出信息中的执行流程是自上而下进行的）。

```
RB#debug ip rip

RIP protocol debugging is on
```

```
*Mar  3 02:11:39.207: RIP: received packet with text authentication 234

*Mar  3 02:11:39.211: RIP: received v2 update from 141.108.0.10 on Serial0

*Mar  3 02:11:39.211: RIP: 70.0.0.0/8 via 0.0.0.0 in 1 hops
```

以下是在 RB 上执行 **show ip route** 特权模式命令，从中可以看出 RB 已从 RA 上学习到了一条 RIP 路由（以**粗体字**显示）。

```
RB#show ip route

R    70.0.0.0/8 [120/1] via 141.108.0.10, 00:00:25, Serial0

      80.0.0.0/24 is subnetted, 1 subnets

C    80.80.80.0 is directly connected, Loopback0

      141.108.0.0/30 is subnetted, 1 subnets

C    141.108.0.8 is directly connected, Serial0
```

3. MD5 RIP 认证配置

RA 路由器上的配置（本端所配置的**密钥 ID、密钥字符串和 MD5 认证模式**必须要与对端路由器上所配置的**密钥 ID、密钥字符串和 MD5 认证模式**对应一致，但**密钥链名称**可以不一致）：

```
RA>enable
RA# configure terminal
RA(config)#key chain kal    !---定义 RA 上的密钥链名称 kal，一个密钥链可以包括多个密钥，可以与对 RB 端路由器上配置的密钥链名称不一样
RA (config-keychain)#key 1  !---定义 kal 密钥链中的 1 号密钥，必须与对端 RB 路由器上配置的密钥号一致（这一点与前面介绍的纯文本认证要求是不同的）
RA (config-keychain-key)#key-string 234  !---定义 1 号密钥的密钥字符串为 234，必须与对端 RB 路由器上配置的密钥字符串一致
RA (config-keychain-key)#exit
RA (config-keychain)#exit
RA (config)#interface Loopback0
RA (config-if)#ip address 70.70.70.255.255.255.255  !---为 loopback0 环路接口指定 IP 地址，每个路由器通常必须配置一个环路接口 IP 地址，以此作为本路由器 ID，但它不是 RIP 认证所必须配置的
RA (config-if)#exit
RA (config)#interface Serial0
RA (config-if)#ip address 141.108.0.10 255.255.255.252
RA (config-if)#ip rip authentication mode md5  !---指定在 Serial0 接口上采用 MD5 RIP 认证模式
RA (config-if)#ip rip authentication key-chain kal  !---在 Serial0 接口上启用 RIP 认证，并指定使用前面已创建名为 kal 的密钥链
RA (config-if)#exit
RA (config)#router rip
RA (config-router)#version 2
RA (config-router)#network 141.108.0.0
RA (config-router)#network 70.0.0.0
```

RB 上的配置（本端所配置的**密钥 ID、密钥字符串和 MD5 认证模式**必须要与对端路由器上所配置的**密钥 ID、密钥字符串和 MD5 认证模式**对应一致，但**密钥链名称**可以不一致）：

```
RB>enable
RB# configure terminal
RB(config)#key chain kal    !---定义 RB 上的密钥链名称 kal，一个密钥链可以包括多个密钥，可以与对 RA 端路由器上配置的密钥链名称不一样
RB (config-keychain)#key 1  !---定义 kal 密钥链中的 1 号密钥，必须与对端 RA 路由器上配置的密钥号一致（这一点与前面介绍的纯文本认证要求是不同的）
RB (config-keychain-key)#key-string 234  !---定义 1 号密钥的密钥字符串为 234，必须与对端 RA 路由器上配置的密钥字符串一致
RB (config-keychain-key)#exit
RB (config-keychain)#exit
RB (config)#interface Loopback0
RB (config-if)#ip address 80.80.80.1 255.255.255.0  !---为 loopback0 环路接口指定 IP 地址，每个路由器通常必须配置一个环路接口 IP 地址，以此作为本路由器 ID，但它不是 RIP 认证所必须配置的
RB (config-if)#exit
RB (config)#interface Serial0
```

```
RB (config-if)#ip address 141.108.0.9 255.255.255.252
RB (config-if)#ip rip authentication mode md5 !---指定在 Serial0 接口上采用 MD5 RIP 认证模式
RB (config-if)#ip rip authentication key-chain kal !---在 Serial0 接口上启用 RIP 认证, 并指定使用前面已创建名为 kal 的密钥链
RB (config-if)#exit
RB (config)#router rip
RB (config-router)#version 2
RB (config-router)#network 141.108.0.0
RB (config-router)#network 80.0.0.0
```

4. 验证 MD5 RIP 认证

同样可以通过调试命令和命令来验证以上配置的 MD5 RIP 认证是否正确, 两路由器是否进行了正常的路由信息交换。

以下是在 RB 上执行 **debug ip rip** 特权模式命令的输出信息, 从中也可以看出在正式接受来自 RA 的 RIP 路由更新前也是经过了 MD5 认证 (输出信息中的执行流程是自上而下进行的)。

```
RB#debug ip rip

RIP protocol debugging is on

*Mar  3 20:48:37.046: RIP: received packet with MD5 authentication

*Mar  3 20:48:37.046: RIP: received v2 update from 141.108.0.10 on Serial0

*Mar  3 20:48:37.050:  70.0.0.0/8 via 0.0.0.0 in 1 hops
```

以下是在 RB 上执行 **show ip route** 特权模式命令, 从中可以看出 RB 已从 RA 上学习到了一条 RIP 路由 (以粗体字显示)。

```
RB#show ip route

R    70.0.0.0/8 [120/1] via 141.108.0.10, 00:00:03, Serial0

      80.0.0.0/24 is subnetted, 1 subnets

C        80.80.80.0 is directly connected, Loopback0

      141.108.0.0/30 is subnetted, 1 subnets

C        141.108.0.8 is directly connected, Serial0
```

5. 故障排除

在 RIP 认证配置中最常见的故障就是通过调试命令查看时显示无效的认证 (**invalid authentication**), 如下所示。

```
RA#debug ip rip

RIP protocol debugging is on

*Mar 1 06:47:42.422: RIP: received packet with text authentication 234

*Mar 1 06:47:42.426: RIP: ignored v2 packet from 141.108.0.9 (invalid authentication)

RB#debug ip rip

RIP protocol debugging is on

*Mar 1 06:48:58.478: RIP: received packet with text authentication 235

*Mar 1 06:48:58.482: RIP: ignored v2 packet from 141.108.0.10 (invalid authentication)
```

同时通过 `show ip route` 命令查看路由时发现竟然没有学习到邻居路由器上的 RIP 路由。下面是在 RB 上操作的示例：

```
RB#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

80.0.0.0/24 is subnetted, 1 subnets

    C 80.80.80.0 is directly connected, Loopback0

    141.108.0.0/30 is subnetted, 1 subnets

    C 141.108.0.8 is directly connected, Serial0

RB#
```

原因很简单，就是配置不匹配造成的。当采用纯文本 RIP 认证时，一定要确保以下两个方面的配置与对端路由器的对应配置是完全一致的：一是所配置的密钥字符串，二是认证模式必须都是纯文本认证模式（默认为纯文本认证模式，无需配置）。

而当采用 MD5 RIP 认证时，一定要确保以下三个方面的配置与对端路由器上的对应配置完全一致：一是所配置的密钥字符串，二是所配置的密钥 ID（这一要求在纯文本认证模式中是没有的），三是认证模式必须都是 MD5 认证模式，必须专门指定。

8.3.4 RIP 路由汇总配置示例一

本示例的拓扑结构如图 8-19 所示，各路由器的各接口所连接的网段在图中均有标注。要做 RIP 路由汇总时要注意的一点是，并不是需要在每个路由器，也不是需要在一个路由器的每个接口都启用路由汇总。只是在有多个同一标准网络之下的子网 RIP 路由经过的路由器的对应接口上才可能需要进行 RIP 路由汇总配置。这句话包括两个必要条件：**一是必须有多个由同一个标准网络或子网划分的子网路由经过，二是这些子网路由必须是通过 RIP 学习到的。**下面分别对本示例中各路由器以及它们的各个接口进行路由汇总可行性和必要性的分析。

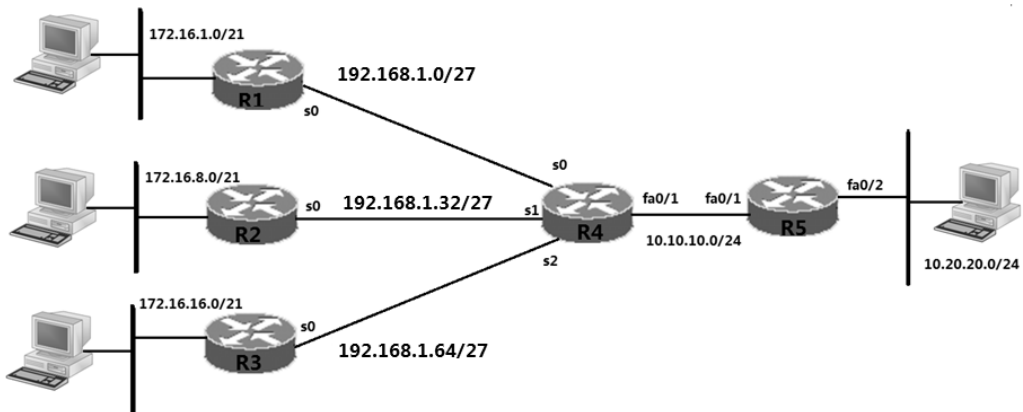


图 8-19 RIP 路由汇总配置示例拓扑结构

1. 路由汇总配置分析

在本示例中 R1、R2 和 R3 这三个路由器上虽然它们都连接了两个子网，但它们对于 R1、R2 或 R3 路由器来说都是直连路由，不需要通过 RIP 协议学习，所以它们不满足上面所说的条件二。况且对于每个路由器所连接的两个子网来说，它们都不是由同一标准网络了网划分，也不满足上面所说的条件一。如 R1 所连接的 172.16.1.0/21 和 192.168.1.0/27 这两个子网不是连续的，所以不能把这两个子网的路由进行汇总。R2 和 R3 上所连接的两个子网一样。

在图中的 R4 路由器有 4 个接口连接了不同的网段，而只有 fa0/1 接口上才可能满足以上所说的两个路由汇总的必要条件，因为它上面可能通过 RIP 学习到 R1、R2 和 R3 路由器所连接的 172.16.1.0/21、172.16.8.0/21 和 172.16.16.0/21 这三个位于同一标准网络 172.16.0.0/16 之下的三个子网 RIP 路由。但在这里要注意的是：R4 路由器上的 fa0/1 接口不能把 R4 路由器上的 s0、s1 和 s2 这三个接口所连接的两个虽然也位于同一标准网络 192.168.1.0/24 之下的子网路由进行汇总，原因就是这三个子网的路由在 R4 路由器上是直接连接的，不是通过 RIP 学习到的。而对于 R5 路由器上所连接的两个子网，虽然表面上也可以汇总成 10.0.0.0/8，或者 10.0.0.0/16 标准网络之下，但因为其中的 10.10.10.0/24 子网在 R4 路由器上也是直接连接的，所以也不能把这两个子网路由进行汇总。但如果 R5 还有其他接口连接了同样位于 10.0.0.0/8 标准网络之下的子网，则在 R4 路由器的 fa0/1 上就也可以把这些子网汇总成 10.0.0.0/8，或者 10.0.0.0/16。

最后分析一下 R5 路由器，虽然连接了同时位于 10.0.0.0/8 这个标准网络下的两个子网，但同样它们在 R5 路由器是直接连接的，不是通过 RIP 协议学习到的，所以这两个子网路由也不会再在 R5 上被汇总。但是在 R5 路由器上的 fa0/1 接口上也可以对 R1、R2 和 R3 路由器所连接的三个位于同一标准网络 172.16.0.0/16 之下的三个子网 RIP 路由进行汇总。但这里要注意的是，如果在 R4 路由器上的 fa0/1 接口已对这三个子网进行了汇总，在 R5 路由器的 fa0/1 接口上就不必再配置对这三个子网路由进行汇总了。因为这三个子网路由到达 R5 路由器的源接口，也就是 R4 路由器上的 fa0/1 接口已对它们进行了汇总，到达 R5 路由器的 fa0/1 接口时不会再有各子网路由，只有一条汇总路由了。至于 R5 路由器的 fa0/2 接口就更没有必要配置对 R1、R2 和 R3 路由器所连接的三个位于同一标准网络 172.16.0.0/16 之下的三个子网 RIP 路由进行汇总了，原因是一样的。当然如果 R4 路由器上的 fa0/1 接口没有配置以上这三个子网的 RIP 路由汇总，则可以在 R5 路由器上的 fa0/1 接口或者 fa0/2 接口上配置以上这三个子网的 RIP 路由汇总。

但是在 R5 路由器的 fa0/1 接口或者 fa0/2 接口上对 R4 路由器 s0、s1 和 s2 接口上所连接的两个子网路由进行汇总，因为它们同时满足以上所说的 RIP 路由汇总的两个条件。

通过以上分析，可以得出，在本示例中可以选择性地进行如下路由汇总配置：

- 仅在 R4 路由器上的 fa0/1 接口上配置对 R1、R2 和 R3 路由器所连接的三个位于同一标准网络 172.16.0.0/16 之下的三个子网 RIP 路由进行汇总，汇总路由掩码就是 255.255.0.0。对于本示例来说，**这是最佳选择**，因为这样可以减少在 R5 路由器上的路由条目。
- 如果 R4 路由器上的 fa0/1 接口没有配置以上这三个子网的 RIP 路由汇总，可在在 R5 路由器上的 fa0/1 接口或者 fa0/2 接口上配置以上这三个子网的 RIP 路由汇总，汇总路由掩码就是 255.255.0.0。
- 可在 R5 路由器的 fa0/1 接口或者 fa0/2 接口上对 192.168.1.0/27、192.168.1.32/27、192.168.1.64/27 这三个子网路由进行汇总。

2. RIP 路由汇总配置

下面仅介绍在 R4 路由器上的 fa0/1 接口上对 172.16.1.0/21、172.16.8.0/21 和 172.16.16.0/21 这三个子网的 RIP 路由汇总，在 R5 路由器 fa0/1 接口上对 192.168.1.0/27、192.168.32.0/27、192.168.64.0/27 这三个子网路由进行汇总。各路由器与 RIP 相关的配置如下（至于各路由器接口 IP

地址配置在此没有给出，具体参见第 3 章):

R1 路由器上的配置:

```
R1(config)#router rip
R1(config-router)#network 172.16.1.0
R1(config-router)#network 192.168.64.0
```

R2 路由器上的配置:

```
R2(config)#router rip
R2(config-router)#network 172.16.8.0
R2(config-router)#network 192.168.32.0
```

R3 路由器上的配置:

```
R3(config)#router rip
R3(config-router)#network 172.16.16.0
R3(config-router)#network 192.168.0.0
```

R4 路由器上的配置 (这是关键配置):

```
R4(config)#router rip
R4(config-router)#version 2 !---此处也是可选的，可以在具体接口上选择启用 RIPv2 的支持
R4(config-router)#network 10.10.10.0
R4(config-router)#network 192.168.0.0
R4(config-router)#network 192.168.32.0
R4(config-router)#network 192.168.64.0
R4(config-router)#no auto-summary !---禁止自动汇总功能
R4(config-router)#exit
R4(config)#interface fa0/1
R4(config-if)# no ip split-horizon !---这里必须禁止水平分割，否则各个子网路由和汇总路由都将不会向邻居路由器通告
R4(config-if)#ip summary-address rip 172.16.0.0 255.255.0.0 !---把 172.16.0.0 下的所有子网路由汇总成 172.16.0.0/16 路由
```

R5 路由器上的配置:

```
R5(config)#router rip
R5(config-router)#network 10.10.10.0
R5(config-router)#network 10.20.20.0
R5(config-router)#exit
R5(config)#interface fa0/1
R5(config-if)# no ip split-horizon
R5(config-if)#ip summary-address rip 192.168.1.0 255.255.255.0 !---把 192.168.1.0 下的所有子网路由汇总成 192.168.1.0/24 路由
```

最后大家分析一下，在本示例中还有哪些接口上可以配置对哪些子网的手动的 RIP 路由由汇总。

8.3.5 RIP 路由汇总配置示例二

为了加深对路由总的理解，再举一个简单的配置示例。本示例的 RIP 路由器 GigabitEthernet 0/2 接口会通过 RIP 学习到包括 10.1.3.0/25、10.1.3.128/25、10.2.1.0/24、10.2.2.0/24、10.1.2.0/24 和 10.1.1.0/24 这六个子网的 RIP 路由。

这时我们该如何配置汇总路由呢？有两种方式：一是把它们都汇总成 A 类路由。显然这种汇总的力度是最大的，也是最粗的汇总方式（如果启用自动汇总，则会这样汇总），有可能把一些本不想被汇总的子网路由给汇总了，如 10.4.1.0/24、10.5.1.0/25 等。另一种方式就是汇总成 B 类路由，这样不仅可以更加细分允许被汇总的子网路由，而且配置更简单，因为只要网络地址一样的多个子网可以通过一条汇总命令来配置。

如本示例中的六个子网中，如果都汇总成 B 类路由，则我们只需考虑各子网中的前 16 位了，这样这六个子网其实也就汇总成了两条汇总路由了，即 10.1.0.0/16 和 10.2.0.0/16，其中 10.1.0.0/16 包括 10.1.1.0/24、10.1.2.0/24、10.1.3.0/25、10.1.3.128/25 这四条子网路由，而 10.2.0.0/16 包括 10.2.1.0/24 和 10.2.2.0/24 这两条子网路由。

最终的配置如下:

```
Router(config)#interface GigabitEthernet 0/2
Router(config-if)# no ip split-horizon !---这是必须配置的，否则汇总路由不可被通告
Router(config-if)#ip summary-address rip 10.1.0.0 255.255.0.0
Router(config-if)#ip summary-address rip 10.2.0.0 255.255.0.0
```

8.4 RIP 路由重发布

我们知道，除了不能把动态路由重发布为静态路由外，包括静态路由在内的所有路由都可以通过 **redistribute** 路由器配置模式命令被重发布为其他动态路由，成为其他动态路由域中的路由，以便在实现整个网络中不同路由域的互联互通。

8.4.1 把其他协议路由重发布到 RIP 路由域

与其他动态路由协议一样，RIP 也一样既可以在本地 RIP 路由进程下把其他路由（包括静态路由）发布到 RIP 路由域中，也可以在其他动态路由进程下把 RIP 路由重发布到对应的动态路由域中。在这里我们仅介绍在 RIP 路由进程中如何把其他路由重发布到 RIP 路由域中。所对应的重发布命令格式如下，命令中的参数和可选项说明如表 8-8 所示：

```
redistribute protocol [process-id] {level-1 | level-1-2 | level-2} [as-number] [metric {metric-value | transparent}] [match {internal | external 1 | external 2}] [tag tag-value] [route-map map-tag]
```

表 8-8 redistribute 命令的参数和选项说明

参数选项	说明
<i>protocol</i>	要重发布的源路由协议，在 RIP 路由进程中本参数可选的协议类型包括： bgp （边界网关协议）、 connected （直接连接）、 eigrp （扩展内部网关协议）、 isis （中间系统到中间系统）、 ospf （开放最短路径优先）、 static （IP 静态路由）
<i>process-id</i>	可选参数，指定被重发布的路由所对应的路由协议进程 ID。当重发布的路由是 BGP 或者 EIGRP 协议时，该参数所代表的是一个 16 位十进制自治系统号码；当重发布的路由是 IS-IS 协议时，该参数所代表的是定义 IS-IS 路由进程名称的标记（tag）值；当重发布的路由是 OSPF 协议时，该参数所代表的是要重发布路由的 OSPF 进程 ID，是一个非零十进制；当重发布的路由是 RIP 协议时，无需配置此参数。默认是没有定义进程 ID 的
level-1	三选一可选项，仅用于重发布 IS-IS 路由时选用，表示仅重发布级别 1 的 IS-IS 路由到其他 IP 路由协议
level-1-2	三选一可选项，仅用于重发布 IS-IS 路由时选用，表示同时重发布级别 1 和级别 2 的 IS-IS 路由到其他 IP 路由协议
level-2	三选一可选项，仅用于重发布 IS-IS 路由时选用，表示仅重发布级别 2 的 IS-IS 路由到其他 IP 路由协议
<i>autonomous-system-number</i>	可选参数，指定路由重发布后的路由的 AS 号，取值范围为 1~65535 之间的整数。但要注意，在 IOS 12.0 (32) SY8、12.0 (33) S3、12.2 (33) SRE、12.2 (33) XNE、12.2 (33) SX11 和 Cisco IOS XE 2.4 及以后版本中，支持在 65536~4294967295 之间的 4 个字节的连续值，或者在 1.0~ 65535.65535 之间的点分计数值 在 IOS 12.0 (32) S12、12.4 (24) T 和 Cisco IOS XE 2.3 版本中，4 个字节的自治系统号仅支持 1.0 ~ 65535.65535 之间的点分计数值
metric metric-value	可选参数，指定路由重发布后的 RIP 路由度量
metric transparent	可选项，仅用于重发布路由到本地路由器 RIP 路由时，指定 RIP 使用被重发布路由的路由度量作为重发布后的 RIP 路由度量
match {internal external 1 external 2}	可选项，指定可以重发布到其他路由域的 OSPF 路由的条件，也就是哪些 OSPF 路由可以重发布到其他路由域。本可选项仅当重发布 OSPF 路由到其他路由域时选用： internal 表示只有在指定自治系统中的内部路由可以被重发布； external 1 表示只有自治系统的外部路由可以被重发布，但导入到 OSPF 网络域后是作为类型 1 外部路由的； external 2 表示只有自治系统的外部路由可以被重发布，但导入到 OSPF 网络域后是作为类型 2 外部路由的。默认为内部路由和类型 1 外部路由可以被重发布
tag tag-value	可选参数，为重发布后的路由设置标记值。如果没有指定，来自 BGP 和 EGP 的路由将使用远程自治系统号作为标记值；其他路由协议将使用 0 作为标记值
route-map	可选项，指定允许从源路由协议导入到当前路由协议的路由映射（与下面的 <i>map-tag</i> 参数一起使用）。如果没有指定，所有路由都可以重发布。如果选择了此关键字，但是没有配置下面的 <i>map-tag</i> 参数，则不会有任何路由被导入
<i>map-tag</i>	可选参数，指定所配置的路由映射标识符

以下示例是重发布 OSPF 9000 进程中的路由到 RIP 路由进程中。

```
Router(config)#router rip
Router(config-router)#redistribute ospf 9000
Router(config-router)# default metric 1    !---设置重发布后的 RIP 路由默认度量为 1
```

表面上看起来，路由重发布的配置还是很简单的，但是也涉及许多注意事项，否则重发布的路由不会在 RIP 路由域中被通告，因为 RIP 和 IGRP 路由协议中，在进行路由通告前会进行一系列的检查，只有符合通告原则的路由更新才会向邻居路由器通告，RIP 或者 IGRP 邻居路由器在接受路由通告前也会进行一系列的检查，也只有符合接受原则的路由更新才会接受。下面具体介绍。

8.4.2 RIP 路由更新通告发送原则

当 RIP 或者 IGRP 发送路由更新时，它们会在通告更新前执行以下一系列的检查。

(1) 所发送的子网路由更新所对应的主网络与源更新接口上对应的主网络相同吗？

- 如果主网络不同，则源路由器在所发送的子网路由所对应的主网边界上进行路由汇总，并通告这个汇总网络。
- 如果主网络相同，则继续进行下面的检查。

(2) 所通告的子网路由与源接口上 IP 地址具有相同的子网掩码吗？

- 如果具有相同的子网掩码，则源路由器通告这个子网。
- 如果子网掩码不同，再继续下面的检查。

(3) 所通告的子网路由掩码是不是 32 位的吗？

- 如果是 32 位的，且是 RIP 路由协议，则源路由器通告该网络；如果是 IGRP 协议，则 Router 1 丢弃该网络的更新。
- 如果不是 32 位的，则源路由器丢弃该网络的更新。

下面以一个具体的示例进行介绍。如图 8-20 所示，Router 1 上连接了两个网络：131.108.5.0/24 和 137.99.88.0/24。现在 Router 1 要向 Router 2 通告这两个网络。

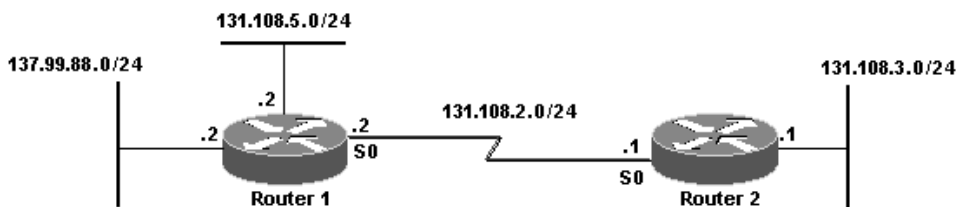


图 8-20 RIP 路由通告发送示例

当 Router 1 发送 131.108.5.0/24 路由更新给 Router 2，它将执行以下检查：

(1) 131.108.5.0/24 所对应的主网络与源更新接口所在的主网络 131.108.2.0/24 相同吗？经过检查得出结论是相同，因为它们的主网络都是 131.108.0.0/16，然后继续下面的检查。

(2) 131.108.5.0/24 与源接口所在网络 131.108.2.0/24 具有相同的子网掩码吗？

经过检查得出结论也是相同的，因为它们都是 255.255.255.0。

根据前面介绍的通告发送原则可知，此时 Router 1 会向 Router 2 通告 131.108.5.0/24 路由。

同理，当 Router 1 发送 137.9.88.0/24 路由更新给 Router 2，它将执行以下检查：

137.99.88.0/24 所对应的主网络与源更新接口所在的主网络 131.108.2.0/24 相同吗？

经过检查得出结论是不相同，因为 137.99.88.0/24 所对应的主网络为 137.99.0.0/16，而源更新接口所在的主网络 131.108.0.0/16。

此时根据以前介绍的通告发送原则得知，无需再进行后面的检查，Router 1 会对所通告的子

网路由在主网络边界进行汇总，通告汇总路由 137.99.0.0/16。

以下是在 Router 1 上执行 **debug ip rip** 命令查看发送 131.108.5.0 和 137.99.0.0 这两条路由更新时的进程。

```
Router 1#debug ip rip
*Mar 25 00:22:46.177: RIP: sending v1 update to 255.255.255.255 via Serial0 (131.108.2.2)
*Mar 25 00:22:46.178: RIP: build update entries
*Mar 25 00:22:46.182: subnet 131.108.5.0, metric 1
*Mar 25 00:22:46.185: network 137.99.0.0, metric 1
```

8.4.3 RIP 路由更新通告接受原则

当 RIP 或者 IGRP 路由器接收路由更新时，它们也会在接收路由更新前执行一系列的检查。下面是在 Router 2 接受来自 Router 1 的路由更新前所执行的一系列检查。

(1) 所接收的子网更新路由所对应的主网与接收该更新的源接口所对应的主网络相同吗？

- 如果主网络相同，则接收路由器会应用接收该子网路由更新的接口 IP 地址所对应的子网掩码。如果被通告的网络中有一位主机 ID，则接收路由器会应用 32 位主机掩码。此时，如果运行的 RIP 协议，会继续通告这个/32 位主机掩码的路由给邻居路由器，而如果运行的是 IGRP 协议，则不会继续通告。
- 如果主网络不同，则继续下面的检查。

(2) 在路由表中存在位于所接收的主网路由之中的明细路由表项吗？这时这个更新应该是主网络，除非在两路由器之间的链路是无编号（也就是没有实际 IP 地址，而是借用其他接口的 IP 地址）的链路，才可能是子网更新。

- 如果有，则接收路由器会忽略这个主网络路由的更新通告。
- 如果没有，则接收路由器会接受该主网络路由的更新，并应用这个主网络的子网掩码。

下面同样以上节的图 8-9 为例，看看 Router 2 是如何接受 Router 1 的路由更新通告的。

当 Router 2 接收到来自 Router 1 的 137.99.0.0/24 的路由更新时，将执行以下检查：

(1) 接收的主网络路由 137.99.0.0 与接收该更新的源接口所在网络 131.108.2.0 所对应的主网络相同吗？

经过检查是不相同的，因为 137.99.0.0 路由的主网络就是 137.99.0.0/16，而源接口所对应的主网络为 131.108.0.0/16，然后接着进行下面的检查。

(2) 在路由表中存在该主网路由所包含的子网表项吗？

经检查没有，则 Router 2 应用实际的子网掩码 16，因为 137.99.0.0 是一个 B 类地址。

同理，当 Router 2 接收到来自 Router 1 的 131.108.5.0/24 的路由更新时，将执行以下检查：

接收的子网 131.108.5.0 路由所对应的主网络与接收该更新的源接口所在网络 131.108.2.0 所对应的主网络相同吗？

经检查是相同的，因为它们所对应的主网络都是 131.108.0.0/16。此时 Router 2 不会再进行其他检查了，直接应用该主网络子网掩码/24，并接受这个 24 位掩码的路由更新。

以下是在 Router 2 上执行 **debug ip rip** 命令查看接收 131.108.5.0 和 137.99.0.0 这两条路由更新时的进程。

```
Router 2#debug ip rip
*Mar 25 00:22:46.201: RIP: received v1 update from 131.108.2.2 on Serial0
*Mar 25 00:22:46.203:131.108.5.0 in 1 hops
*Mar 25 00:22:46.205:137.99.0.0 in 1 hops
```

此时在 Router 2 上执行 **show ip route** 命令，会显示到刚才通过 RIP 所学习的这两条静态路由了（参见输出信息中的粗体字部分）。注意其中的 137.99.0.0/16 与 Router 1 上所连接的

137.99.88.0/24 是不同的，这是在 RIP 路由通告发送时就已做了更改的，具体参见上节相关内容。

```
Router 2#show ip route
R    137.99.0.0/16 [120/1] via 131.108.2.2, 00:00:07, Serial0
     131.108.0.0/24 is subnetted, 3 subnets
R    131.108.5.0 [120/1] via 131.108.2.2, 00:00:08, Serial0
C    131.108.2.0 is directly connected, Serial0
C    131.108.3.0 is directly connected, Ethernet0
```

8.4.4 到 RIP 重发布路由通告存在的问题

在把其他动态路由通告到 RIP 或者 IGRP 协议时要特别注意的一个问题，那就是 RIP 和 IGRP 不会通告与源接口所在网络对应的主网络相同，但子网掩码不同的路由。下面通过一些具体的示例来介绍不同情形下的解决方法。

1. OSPF 路由子网掩码比 RIP 路由子网掩码长

在如图 8-21 所示的示例中，Router GW-2 在 RIP 和 OSPF 之间进行路由重发布。OSPF 路由域中的路由掩码（28 位，对应的子网掩码为 255.255.255.240）比 RIP 路由域中的掩码（24 位，对应的子网掩码为 255.255.255.0）长，但是它们的主网络相同，都是 128.103.0.0/16。所以 RIP 将不会通告从 OSPF 路由域中学习并重发布到 RIP 路由域中的路由。

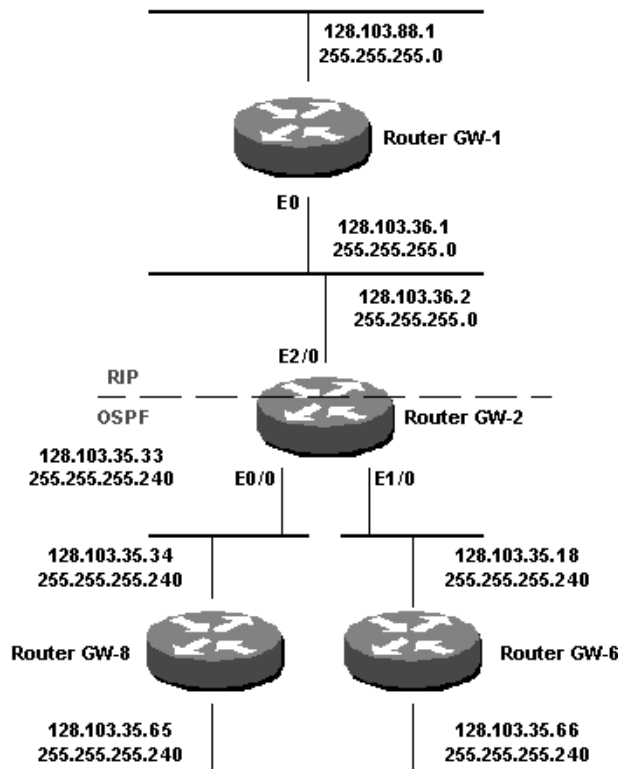


图 8-21 RIP 和 OSPF 路由重发布示例一

一般情况下，一个网络配置好后是不可能随便改变所在的网段的，所以本示例中 OSPF 路由域中的子网掩码是难以改变的。唯有找一个可替代方案，那就是在 Router GW-2 中添加一条指向，并且能包括 OSPF 路由域中路由的静态路由，子网掩码与 RIP 路由域中的一样为 255.255.255.0，但是下一跳为 Null0 接口（其用意就是表示这条静态路由仅用于本地路由表更新，不用于实际的路由转发）。然后重发布这条静态路由到 RIP 路由域，这样就相当于把 OSPF 路由域中的路由重发布到了

RIP 路由域中。下面是具体的配置。

```
RouterGW-2(config)#ip route 128.103.35.0 255.255.255.0 null0
RouterGW-2(config-router)#router rip
RouterGW-2(config-router)#redistribute static
RouterGW-2(config-router)#default metric 1
```

这样就允许 128.103.35.0 被运行 RIP 协议的 Router GW-2 E2/0 接口通告到了 RIP 路由域，而同时 Router GW-2 在路由表中仍然可以保存通过 OSPF 学习的更加明细的路由。

2. RIP 路由子网掩码比 OSPF 路由子网掩码长

前面介绍了当 OSPF 路由域中的子网掩码比 RIP 路由域中的子网掩码长时可以通过配置一条以 RIP 路由域中子网掩码，并且指向 OSPF 路由域中的静态路由，然后通过重发布这条静态路由，并向邻居路由器进行通告来间接地重发布并且向邻居通告了 OSPF 路由域中的路由。但这种方法对于 RIP 路由域中的子网掩码比 OSPF 路由域中的子网掩码长的情形是不适用的，因为这时配置的静态路由比 OSPF 路由域中的路由更明细，根本包括不了 OSPF 路由域中的路由。但是解决思路还是一样的，一条不行，可以多条啊，毕竟一个大的网络划分后也就包含有限数量的子网，只要通过多条静态路由把这些子网路由全包括了，对应大的网络路由也就全包括了。下面也通过一个具体的示例进行介绍。

在如图 8-22 所示的示例中，RIP 路由域中的子网掩码长度为 29（对应的子网掩码为 255.255.255.248），而 OSPF 路由域中的子网掩码长度为 28 位（对应的子网掩码为 255.255.255.240）。这时 RIP 不会通告通过 OSPF 学习，然后重发布到 RIP 路由域中的路由。

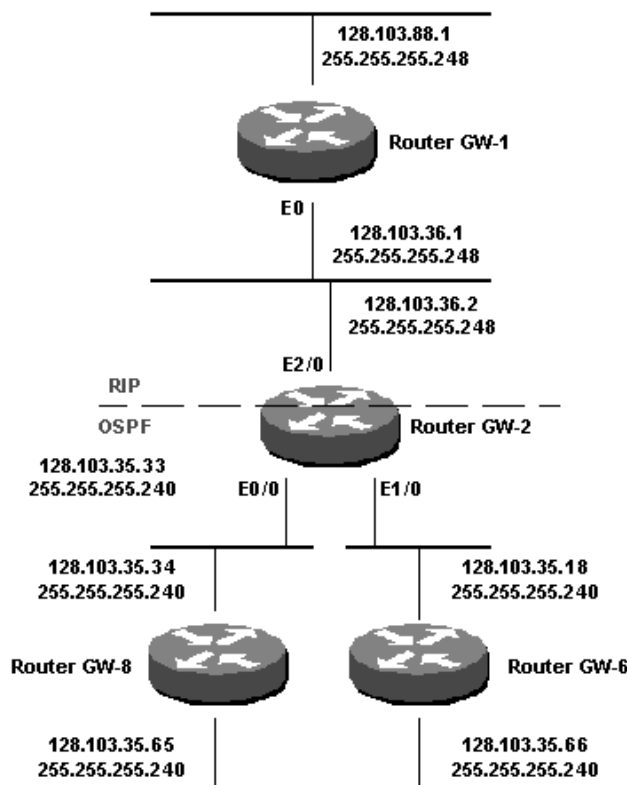


图 8-22 RIP 和 OSPF 路由重发布示例二

我们可以在 Router GW-2 上添加一条指向 OSPF 路由域的静态路由，子网掩码为 255.255.255.248。但是因为这条静态路由的子网掩码比原来 OSPF 路由域中的路由掩码长度更长，所以其下一跳必须为真实的下一跳或接口（不能像前面那样以 Null0 这样的虚拟接口作为下一

跳)。另外，仅这一条静态路由是不够的，还需要配置多条静态路由，以便可以全面覆盖 OSPF 路由域。

在以下配置中，前面两条静态路由覆盖了 OSPF 路由域中的 128.103.35.32/28 范围。而中间两条静态路由覆盖了 OSPF 路由域中的 128.103.35.16/28 范围，最后面的四条静态路由覆盖了 OSPF 路由域中的 128.103.35.64/28 范围。然后通过重发布这些静态路由，就相当于把 OSPF 路由域中的所有动态路由都发布到了 RIP 路由域中。

```
RouterGW-2(config)#ip route 128.103.35.32 255.255.255.248 E0/0
RouterGW-2(config)#ip route 128.103.35.40 255.255.255.248 E0/0

RouterGW-2(config)#ip route 128.103.35.16 255.255.255.248 E1/0
RouterGW-2(config)#ip route 128.103.35.24 255.255.255.248 E1/0

RouterGW-2(config)#ip route 128.103.35.64 255.255.255.248 128.103.35.34
RouterGW-2(config)#ip route 128.103.35.64 255.255.255.248 128.103.35.18
RouterGW-2(config)#ip route 128.103.35.72 255.255.255.248 128.103.35.34
RouterGW-2(config)#ip route 128.103.35.72 255.255.255.248 128.103.35.18
RouterGW-2(config)#router rip
RouterGW-2(config-router)#redistribute static
RouterGW-2(config-router)#default metric 1
```