

实验：RIP

HCIP 分解实验 - RIP

臧家林制作



- RIP 实验 1：RIP 基本配置，认证
- RIP 实验 2：RIP 路由聚合
- RIP 实验 3：RIP 版本兼容，定时器
- RIP 实验 4：RIP 制接口及单播更新
- RIP 实验 5：RIP 不连续子网
- RIP 实验 6：RIP 水平分割
- RIP 实验 7：RIP 附加度量值
- RIP 实验 8：RIP 路由引入
- RIP 实验 9：next hop 作用

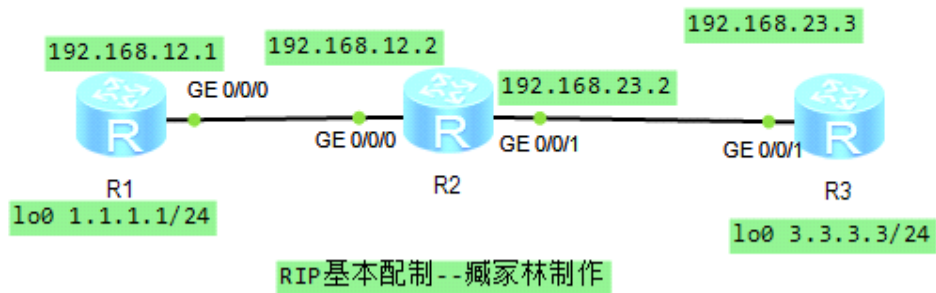
=====

RIP 实验 1：RIP 基本配置，认证

RIP 作为最早的距离矢量 IP 路由协议，也是最先得到广泛使用的一种路由协议，采用 Bellman-Ford 算法，其最大特点就是配置简单。

RIP 协议要求网络中每一台路由器都要维护从自身到每

一个目的网络的路由信息。RIP 协议使用跳数来称量网络间的“距离”，从一台路由器到其直连网络的跳数定义为 1，从一台路由器到其非直连网络的距离定义为每经过一个路由器则距离加 1。距离也称为跳数。RIP 允许路由的最大跳数为 15，因此，16 即为不可达。可见 RIP 协议只适用于小型网络。



1.基本接口配置

R1:

```
int lo0
```

```
ip add 1.1.1.1 24
```

```
int g0/0/0
```

```
ip add 192.168.12.1 24
```

R2:

```
int g0/0/0
```

```
ip add 192.168.12.2 24
```

```
int g0/0/1
```

```
ip add 192.168.23.2 24
```

R3:

```
int lo0
```

```
ip add 3.3.3.3 24
```

```
int g0/0/1
```

```
ip add 192.168.23.3 24
```

2.运行 RIPv1

```
[R1]rip ?  
  INTEGER<1-65535>  Process ID  
  mib-binding        Mib-Binding a process  
  vpn-instance       VPN instance  
  <cr>              Please press ENTER to
```

rip 直接回车，为进程 1

R1:

rip

net 1.0.0.0

net 192.168.12.0

R2:

rip

net 192.168.23.0

net 192.168.12.0

R3:

rip

net 3.0.0.0

net 192.168.23.0

R1 通过 RIP 协议，学习到了两条路由

```

[R1]display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
Destinations : 12      Routes : 12

Destination/Mask    Proto  Pre  Cost    Flags NextHop         Interface
1.1.1.0/24         Direct  0    0        D   1.1.1.1           LoopBack0
1.1.1.1/32         Direct  0    0        D   127.0.0.1          LoopBack0
1.1.1.255/32        Direct  0    0        D   127.0.0.1          LoopBack0
3.0.0.0/8           RIP     100  2        D   192.168.12.2       GigabitEthernet0/0/0
127.0.0.0/8         Direct  0    0        D   127.0.0.1          InLoopBack0
127.0.0.1/32        Direct  0    0        D   127.0.0.1          InLoopBack0
127.255.255.255/32  Direct  0    0        D   127.0.0.1          InLoopBack0
192.168.12.0/24     Direct  0    0        D   192.168.12.1       GigabitEthernet0/0/0
192.168.12.1/32     Direct  0    0        D   127.0.0.1          GigabitEthernet0/0/0
192.168.12.255/32   Direct  0    0        D   127.0.0.1          GigabitEthernet0/0/0
192.168.23.0/24     RIP     100  1        D   192.168.12.2       GigabitEthernet0/0/0
255.255.255.255/32  Direct  0    0        D   127.0.0.1          InLoopBack0

```

display rip 显示 RIP 协议的相关信息

```

[R1]display rip
Public VPN-instance
  RIP process : 1
    RIP version : 1
    Preference : 100
    Checkzero : Enabled
    Default-cost : 0
    Summary : Enabled
    Host-route : Enabled
    Maximum number of balanced paths : 8
    Update time : 30 sec
    Age time : 180 sec
    Garbage-collect time : 120 sec
    Graceful restart : Disabled
    BFD : Disabled
    Silent-interfaces : None
    Default-route : Disabled
    Verify-source : Enabled
    Networks :
      192.168.12.0 1.0.0.0

```

display rip 1 route

```

[R1]display rip 1 route
Route Flags : R - RIP
              A - Aging, G - Garbage-collect
-----
Peer 192.168.12.2 on GigabitEthernet0/0/0
  Destination/Mask    Nexthop    Cost    Tag    Flags    Sec
  192.168.23.0/24     192.168.12.2    1      0      RA       24
  3.0.0.0/8           192.168.12.2    2      0      RA       24
[R1]

```

使用 debugging 命令，查看 RIP 协议定期更新情况
 debug 命令需要在用户视图下才能使用，使用 terminal debugging 和 terminal monitor 命令开启 debug 信息在屏幕上显示的功能，才能在电脑屏幕上看到路由器之间的 RIP 协议交互的信息。

```

<R1>debugging rip 1
<R1>terminal debugging
<R1>terminal monitor
<R1>debugging rip 1 event

```

```

Nov 27 2015 08:17:59.498.3-08:00 R1 RIP/7/DBG: 6: 13476: Packet: Version 1,
Length 24
<R1>
Nov 27 2015 08:17:59.498.4-08:00 R1 RIP/7/DBG: 6: 13527: Dest 1.0.0.0, Cost
<R1>
Nov 27 2015 08:18:08.318.1-08:00 R1 RIP/7/DBG: 6: 13414: RIP 1: Receiving v
GigabitEthernet0/0/0 from 192.168.12.2 with 2 RTEs
<R1>
Nov 27 2015 08:18:08.318.2-08:00 R1 RIP/7/DBG: 6: 13465: RIP 1: Receive res
.168.12.2 on GigabitEthernet0/0/0
<R1>

```

```

<R1>undo debugging all  关闭

```

开启过多的 debug 功能会耗费大量路由资源，甚至会导致宕机，慎重使用开启批量 debug 功能的命令，如 debug all

RIPv1 抓包

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.12.2	255.255.255.255	RIPv1	Response
2	3.261000	192.168.12.1	255.255.255.255	RIPv1	Response
3	34.024000	192.168.12.2	255.255.255.255	RIPv1	Response
4	37.269000	192.168.12.1	255.255.255.255	RIPv1	Response

10 102.306000 192.168.12.1 255.255.255.255 RIPv1 Response					
Frame 10: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)					
Ethernet II, Src: HuaweiTe_24:04:08 (00:e0:fc:24:04:08), Dst: Broadcast (ff:ff:ff:ff:ff:ff)					
Internet Protocol, Src: 192.168.12.1 (192.168.12.1), Dst: 255.255.255.255 (255.255.255.255)					
User Datagram Protocol, Src Port: router (520), Dst Port: router (520)					
Routing Information Protocol					
Command: Response (2)					
Version: RIPv1 (1)					
IP Address: 1.0.0.0, Metric: 1					
Address Family: IP (2)					
IP Address: 1.0.0.0 (1.0.0.0)					
Metric: 1					

3.运行 RIPv2

R1:

rip

version 2

R2:

rip

version 2

R3:

rip

version 2

数据抓包，可以看到，RIPv2 使用组播 224.0.0.9，有子网掩码

Time	Source	Destination	Protocol	Info
1 0.000000	192.168.12.2	224.0.0.9	RIPv2	Response
2 6.084000	192.168.12.1	224.0.0.9	RIPv2	Response

2 6.084000 192.168.12.1 224.0.0.9 RIPv2 Response				
+ Frame 2: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)				
+ Ethernet II, Src: HuaweiTe_24:04:08 (00:e0:fc:24:04:08), Dst: IPv4mcast_00:00:09 (01:00:00:00:00:09)				
+ Internet Protocol, Src: 192.168.12.1 (192.168.12.1), Dst: 224.0.0.9 (224.0.0.9)				
+ User Datagram Protocol, Src Port: router (520), Dst Port: router (520)				
- Routing Information Protocol				
Command: Response (2)				
Version: RIPv2 (2)				
Routing Domain: 0				
- IP Address: 1.1.1.0, Metric: 1				
Address Family: IP (2)				
Route Tag: 0				
IP Address: 1.1.1.0 (1.1.1.0)				
Netmask: 255.255.255.0 (255.255.255.0)				
Next Hop: 0.0.0.0 (0.0.0.0)				
Metric: 1				

4.RIP 认证

配置协议的认证可以降低设备接受非法路由选择更新消息的可能性。

RIPv1 不支持验证报文。RIPv2 的验证报文主要有两种类型，简单验证和 MD5 密文验证。

(1) 简单验证：

R1:

```
int g0/0/0
```

```
rip authentication-mode simple huawei
```

R2:

```
int g0/0/0
```

```
rip authentication-mode simple huawei
```

抓包可以看到，R1 R2 之间发送的 RIP 报文中含 authentication 字段，密码是明文 huawei

```
⊕ Frame 1: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits)
⊕ Ethernet II, Src: HuaweiTe_0b:5e:d3 (00:e0:fc:0b:5e:d3), Dst: IPv4mcast_00:00:09 (01:00:5e
⊕ Internet Protocol, Src: 192.168.12.2 (192.168.12.2), Dst: 224.0.0.9 (224.0.0.9)
⊕ User Datagram Protocol, Src Port: router (520), Dst Port: router (520)
⊖ Routing Information Protocol
    Command: Response (2)
    Version: RIPv2 (2)
    Routing Domain: 0
    ⊖ Authentication: Simple Password
        Authentication type: Simple Password (2)
        Password: huawei
    ⊕ IP Address: 1.0.0.0, Metric: 16
    ⊕ IP Address: 3.3.3.0, Metric: 2
    ⊕ IP Address: 3.0.0.0, Metric: 16
    ⊕ IP Address: 192.168.12.0, Metric: 16
    ⊕ IP Address: 192.168.23.0, Metric: 1
```

(2) MD5 密文验证

R2:

```
int g0/0/1
```

```
rip authentication-mode md5 usual huawei
```

R3:

```
int g0/0/1
```

```
rip authentication-mode md5 usual huawei
```

配置时可以选择 MD5 密文验证方式的报文格式

usual 参示使用通用报文格式

nonstandard 参数表示使用非标准报文格式 (IETF 标准)

两端报文格式一致，这是选用通用标准格式

抓包


```

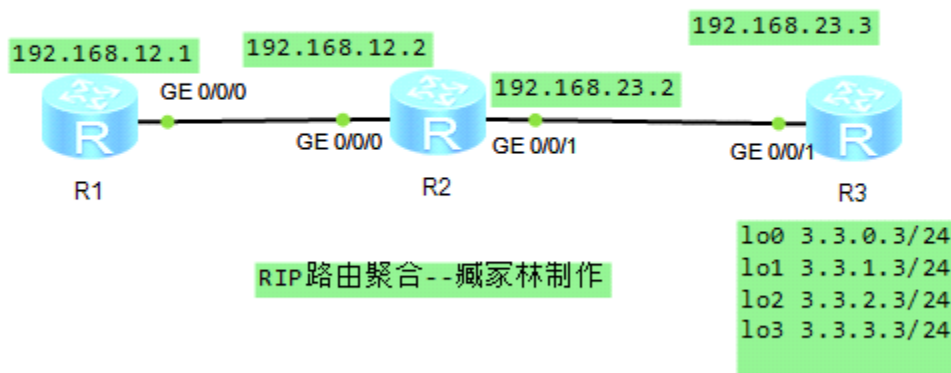
+ Frame 2: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
+ Ethernet II, Src: HuaweiTe_0b:5e:d4 (00:e0:fc:0b:5e:d4), Dst: IPv4mcast_00:00:09
+ Internet Protocol, Src: 192.168.23.2 (192.168.23.2), Dst: 224.0.0.9 (224.0.0.9)
+ User Datagram Protocol, Src Port: router (520), Dst Port: router (520)
+ Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  Routing Domain: 0
  Authentication: Keyed Message Digest
    Authentication type: Keyed Message Digest (3)
    Digest Offset: 60219
    Key ID: 168
    Auth Data Len: 164
    Seq num: 1965174231
    Zero Padding

```

=====

RIP 实验 2：RIP 路由聚合

当网络中路由器的路由条目非常多时，可以通过路由汇总（又称路由汇聚或路由聚合）来减少路由条目数，加快路由收敛时间和增强网络的稳定性。路由汇总的原理是，同一个自然网段内的不同子网的路由在向外（其他网段）发送时聚合成一个网段的路由发送。由于汇总后路由器将不会感知被汇总子网有关的变化，从而提高了网络稳定性，减少了不必要的路由器更新。



1.运行 RIP 版本 1

```
<R1>display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 9           Routes : 9

Destination/Mask    Proto   Pre  Cost   Flags NextHop
-----
 3.0.0.0/8          RIP     100   2       D  192.168.12.2
127.0.0.0/8          Direct   0     0       D  127.0.0.1
127.0.0.1/32         Direct   0     0       D  127.0.0.1
```

R1 上只能看到 3.0.0.0/8，没有任何明细路由条目

RIPv1 没的携带掩码信息，只发布汇总后的有类路由，默认开启自动汇总，且无法关闭，也不支持手工汇总。

display default-parameter rip 查看 RIP 默认配置信息，可以看到默认开启了自动汇总。

```
<R1>display default-parameter rip
-----
Protocol Level Default Configurations
-----
RIP version       : 1
Preference        : 100
Checkzero         : Enabled
Default-cost      : 0
Auto Summary      : Enabled
Host-route        : Enabled
Maximum Balanced Paths : 8
```

2.运行 RIP 版本 2

rip

version 2

配置完成后，R1 能看到 R3 的明细路由,而没有汇总路由，即此时版本 2 默认自动汇总关没有生效。

```
[R1]display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 12          Routes : 12
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
3.3.0.0/24	RIP	100	2	D	192.168.12.2
3.3.1.0/24	RIP	100	2	D	192.168.12.2
3.3.2.0/24	RIP	100	2	D	192.168.12.2
3.3.3.0/24	RIP	100	2	D	192.168.12.2
127.0.0.0/8	Direct	0	0	D	127.0.0.1
127.0.0.1/32	Direct	0	0	D	127.0.0.1
127.255.255.255/32	Direct	0	0	D	127.0.0.1
192.168.12.0/24	Direct	0	0	D	192.168.12.1
192.168.12.1/32	Direct	0	0	D	127.0.0.1
192.168.12.255/32	Direct	0	0	D	127.0.0.1
192.168.23.0/24	RIP	100	1	D	192.168.12.2
255.255.255.255/32	Direct	0	0	D	127.0.0.1

为了防止环路和不连续子网问题的产生，在启用了水平分割或毒性逆转的接口上，RIPv2 的默认自动汇总就会失效，所以 R3 通告过来的都是具体明细路由条目。

RIPv2 的默认自动汇总，有两种方法

(1)summary always

R3:

rip

summary always

(2)关闭接口下的水平分割:

将上面的命令删掉，查看 RIP 进程下命令是正常的，R1 看到的是明细路由

R3:

rip

undo summary

summary

```
#  
rip 1  
version 2  
network 192.168.23.0  
network 3.0.0.0
```

R3 接口上关闭水平分割 R3 :

int g0/0/1

undo rip split-horizon

```
[R1]disp ip rout  
Route Flags: R - relay, D - download to fib  
-----  
Routing Tables: Public  
Destinations : 9 Routes : 9  


| Destination/Mask   | Proto  | Pre | Cost | Flags | NextHop      |
|--------------------|--------|-----|------|-------|--------------|
| 3.0.0.0/8          | RIP    | 100 | 2    | D     | 192.168.12.2 |
| 127.0.0.0/8        | Direct | 0   | 0    | D     | 127.0.0.1    |
| 127.0.0.1/32       | Direct | 0   | 0    | D     | 127.0.0.1    |
| 127.255.255.255/32 | Direct | 0   | 0    | D     | 127.0.0.1    |
| 192.168.12.0/24    | Direct | 0   | 0    | D     | 192.168.12.1 |


```

R1 上只能看到自动汇总的一条路由

3.0.0.0 是 A 类 IP 地址，默认掩码为 8 位

3.配置 RIPv2 手工汇总

R3 打开水平分割，R1 看到的是明细路由，再进行手工汇总

int g0/0/1

rip split-horizon

R3 :

int g0/0/1

rip summary-address 3.3.0.0 255.255.252.0

```

[R1]dis ip rout
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 9           Routes : 9

Destination/Mask    Proto   Pre  Cost   Flags NextHop
-----
  3.3.0.0/22        RIP     100   2       D    192.168.12.2
  127.0.0.0/8        Direct   0     0       D    127.0.0.1
  127.0.0.1/32       Direct   0     0       D    127.0.0.1
  127.255.255.255/32 Direct   0     0       D    127.0.0.1
  192.168.12.0/24    Direct   0     0       D    192.168.12.1

```

R1 上有手工汇总的路由

=====

RIP 实验 3：.RIP 版本兼容，定时器

RIP 配置时如不指定版本，默认为 RIPv1 接口默认情况下能接收 v1 和 v2 的报文，但只能发送 v1 报文

指定为版本时，RIPv1 只能发送和接收 v1 的报文，RIPv2 只能发送和接收 v2 的报文

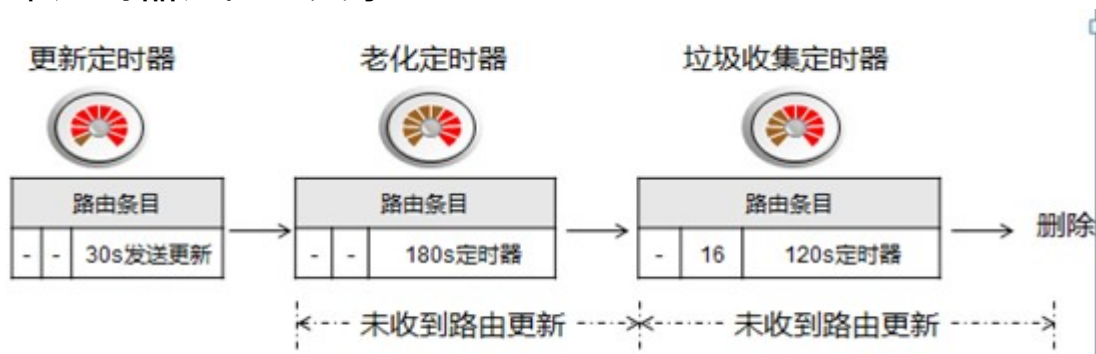
RIP 主要使用三个定时器

更新定时器

老化定时器

垃圾收集定时器

三个定时器之间的关系

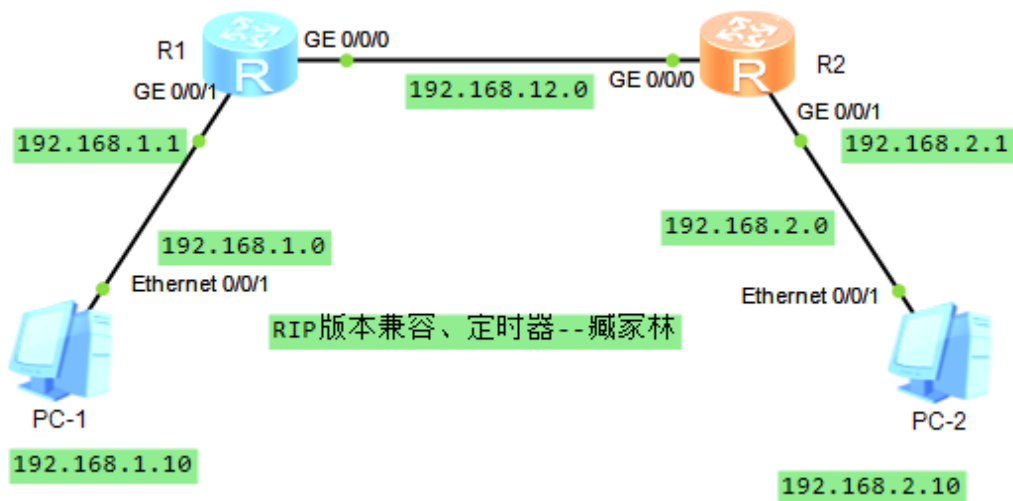


更新定时器：路由器每 30s 发送一次路由应答或者更新。

老化定时器：如果收到了对应的路由，老化计时器就会复位 180s。如果路由器 180s 没有收到路由更新或者是收到 metric 是 16 的路由更新时就会将路由的 metric 值置为 16，这条路由从路由表 (display ip routing-table) 中清除，但是不会从 RIP 的数据库 (display rip 1 database) 中清除，因为路由器还要告诉其他相邻的路由器这条路由是不可达的。，否则就会发生环路。

垃圾收集定时器：老化计时器记录的路由超时，在 rip 数据库中 metric 为 16 跳的路由，这种路由发给其它路由器只要是为了防止环路，由于 rip 是不可靠的 UDP 传输的协议，所以路由器并不是发送一次就把 16 跳的路由删除，而是会保持 120s (相当于发送 4 次)，120s 过后就会从 database 中删除这条 16 跳的路由。

RIP 的定时器有 3 种：[更新定时器](#)，默认每 30s 发送一次更新，[超时计时器](#)，默认时间 180s，如果在超时计时器内没有收到邻居发来的更新报文，则把该路由的度量值设置为 16，并启动垃圾收集定时器。[垃圾收集定时器](#)，默认时间为 120s，如果启动了该计时器，那么 120s 超时后，路由表会删除该路由表项。



1.版本兼容

基本配置

R1 不指定版本默认为 RIPv1, R2 指定为 RIPv2

R1 :

rip

net 192.168.1.0

net 192.168.12.0

R2 :

rip

version 2

net 192.168.2.0

net 192.168.12.0

查看 R1 R2 的路由表

```
<R1>display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
    Destinations : 11          Routes : 11

Destination/Mask    Proto    Pre  Cost      Flags NextHop
-----
      127.0.0.0/8    Direct  0     0          D   127.0.0.1
      127.0.0.1/32    Direct  0     0          D   127.0.0.1
127.255.255.255/32    Direct  0     0          D   127.0.0.1
      192.168.1.0/24    Direct  0     0          D   192.168.1.1
      192.168.1.1/32    Direct  0     0          D   127.0.0.1
      192.168.1.255/32    Direct  0     0          D   127.0.0.1
      192.168.2.0/24    RIP     100    1          D   192.168.12.2
      192.168.12.0/24    Direct  0     0          D   192.168.12.1
```

```
<R2>display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
    Destinations : 10          Routes : 10

Destination/Mask    Proto    Pre  Cost      Flags NextHop
-----
      127.0.0.0/8    Direct  0     0          D   127.0.0.1
      127.0.0.1/32    Direct  0     0          D   127.0.0.1
127.255.255.255/32    Direct  0     0          D   127.0.0.1
      192.168.2.0/24    Direct  0     0          D   192.168.2.1
      192.168.2.1/32    Direct  0     0          D   127.0.0.1
      192.168.2.255/32    Direct  0     0          D   127.0.0.1
      192.168.12.0/24    Direct  0     0          D   192.168.12.2
      192.168.12.2/32    Direct  0     0          D   127.0.0.1
      192.168.12.255/32    Direct  0     0          D   127.0.0.1
255.255.255.255/32    Direct  0     0          D   127.0.0.1
```

可以看到 R1 路由表中有 192.168.2.0 网段的路由，但在到 R2 路由表中却没有 192.168.1.0 网段的路由
这是因为 R1 为版本 1，默认可以处理接收版本 1 和版本 2 的

报文，但只发送版本 1 的报文
而 R2 为版本 2，仅接收和发送版本 2 的报文。

因此 R1 发送的版本 1 的报文，R2 不能正确接收，所以 R2 路由表中没有 192.168.1.0，而 R2 发送的版本 2 的报文，能够被 R1 处理接收，所以 R1 的路由表中有 192.168.2.0

R2 能接收 192.168.1.0 路由，需要让 R1 以广播的方式发送版本 2

R1 :
int g0/0/0
rip version 2 broadcast
配置完成后，R2 路由表

```
<R2>display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 11          Routes : 11

Destination/Mask    Proto    Pre  Cost           Flags NextHop
-----
      127.0.0.0/8     Direct   0     0             D    127.0.0.1
      127.0.0.1/32    Direct   0     0             D    127.0.0.1
127.255.255.255/32  Direct   0     0             D    127.0.0.1
      192.168.1.0/24  RIP      100    1             D    192.168.12.1
      192.168.2.0/24  Direct   0     0             D    192.168.2.1
      192.168.2.1/32  Direct   0     0             D    127.0.0.1
      192.168.2.255/32 Direct   0     0             D    127.0.0.1
      192.168.12.0/24 Direct   0     0             D    192.168.12.2
```

2.查看 RIP 定时器
更新计时器

在 R1 的接口下抓包，查看路由更新，默认 RIP 协议每隔 30s 左右发送一次路由更新。

不一定是 30s 那么准

2	25.688000	192.168.12.1
3	28.016000	192.168.12.2
4	53.047000	192.168.12.2
5	60.719000	192.168.12.1
6	84.063000	192.168.12.2
7	91.766000	192.168.12.1
8	114.078000	192.168.12.2

超时计时器

为了验证效果，在 R1 的 g0/0/0 接口上配置停止发送 RIP 路由更新

R1 :

```
int g0/0/0
```

```
undo rip output
```

配置完成后，R1 已经无法发送任何 RIP 路由更新，此时查看 R2 路由表，R1 的路由条目依然存在，原因是 RIP 超时定时器没有到期。

```
<R2>display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
                Destinations : 11          Routes : 11

Destination/Mask    Proto    Pre  Cost           Flags NextHop
-----
        127.0.0.0/8  Direct   0    0             D    127.0.0.1
        127.0.0.1/32 Direct   0    0             D    127.0.0.1
127.255.255.255/32 Direct   0    0             D    127.0.0.1
        192.168.1.0/24 RIP      100   1             D    192.168.12.1
        192.168.2.0/24 Direct   0    0             D    192.168.2.1
        192.168.2.1/32 Direct   0    0             D    127.0.0.1
        192.168.2.255/32 Direct   0    0             D    127.0.0.1
```

<R2>display rip 1 database 查看 R2 的 RIP 发布数据库中所有激活路由

路由条目也没有发生变化，状态仍然为 A，即仍然被通告。

```
<R2>display rip 1 database
-----
Advertisement State : [A] - Advertised
                     [I] - Not Advertised/withdraw
-----
192.168.1.0/24, cost 1, ClassfulSumm
192.168.1.0/24, cost 1, [A], nexthop 192.168.12.1
192.168.2.0/24, cost 0, ClassfulSumm
192.168.2.0/24, cost 0, [A], Rip-interface
192.168.12.0/24, cost 0, ClassfulSumm
192.168.12.0/24, cost 0, [A], Rip-interface
```

在等待超时计时器到 180s 以后再查看 R2 的路由表，就没有 R1 的路由条目，原因是超时定时器已经到期，该路由条目被定义为失效，已经从路由表中清除了

```

<R2>display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 10          Routes : 10

Destination/Mask    Proto   Pre  Cost   Flags NextHop
-----
      127.0.0.0/8    Direct  0     0        D   127.0.0.1
      127.0.0.1/32    Direct  0     0        D   127.0.0.1
127.255.255.255/32    Direct  0     0        D   127.0.0.1
      192.168.2.0/24   Direct  0     0        D   192.168.2.1
      192.168.2.1/32   Direct  0     0        D   127.0.0.1
      192.168.2.255/32 Direct  0     0        D   127.0.0.1
      192.168.12.0/24  Direct  0     0        D   192.168.12.2
      192.168.12.2/32  Direct  0     0        D   127.0.0.1
      192.168.12.255/32 Direct  0     0        D   127.0.0.1
255.255.255.255/32    Direct  0     0        D   127.0.0.1

```

垃圾收集定时器

同时再次检查 R2 的数据库表，该路由条目被标记为 16 路由，即不可达，状态为 I，该路由将不能被通告出去。虽然已经失效，但仍然存在于发布数据库中的原因是 RIP 垃圾收集定时器启动，且还没有到期，暂时不能从数据库不清除。

```

<R2>display rip 1 database
-----
Advertisement State : [A] - Advertised
                     [I] - Not Advertised/withdraw
-----
192.168.1.0/24, cost 16, classfulsumm
192.168.1.0/24, cost 16, [I], nexthop 192.168.12.1
192.168.2.0/24, cost 0, classfulsumm
192.168.2.0/24, cost 0, [A], Rip-interface
192.168.12.0/24, cost 0, classfulsumm
192.168.12.0/24, cost 0, [A], Rip-interface

```

如果 120s 仍然没有收到更新报文，垃圾收集定时器超时后将删除该表项。经过 120s 后再查看 R2 上的数据库

```
<R2>display rip 1 database
-----
Advertisement State : [A] - Advertised
                     [I] - Not Advertised/Withdraw
-----
 192.168.2.0/24, cost 0, ClassfulSumm
 192.168.2.0/24, cost 0, [A], Rip-interface
 192.168.12.0/24, cost 0, ClassfulSumm
 192.168.12.0/24, cost 0, [A], Rip-interface
<R2>
```

3.修改 RIP 定时器

可以通过 timer rip 命令来修改几个定时器的默认值来影响 R IP 的收敛速度。

将 R1 的接口恢复

R1:

int g0/0/0

rip output

R1:

rip

timers rip 5 10 15

R2:

rip

timers rip 5 10 15

display rip 修改之后，查看一下，是否生效

```

[R2]display rip
Public VPN-instance
  RIP process : 1
    RIP version      : 2
    Preference       : 100
    Checkzero        : Enabled
    Default-cost     : 0
    Summary          : Enabled
    Host-route       : Enabled
    Maximum number of balanced paths : 8
    Update time      : 5 sec
    Age time         : 10 sec
    Garbage-collect time : 15 sec
    Graceful restart : Disabled

```

抓包查看更新时间，5s 左右

1	0.000000	192.168.12.1
2	6.000000	192.168.12.1
3	11.296000	192.168.12.2
4	16.015000	192.168.12.1
5	19.296000	192.168.12.2
6	25.296000	192.168.12.2
7	29.031000	192.168.12.1

R1 接口不发送路由更新

R1:

int g0/0/0

undo rip output

display ip routing-table

display rip 1 database

查看超时定时器和垃圾收集定时器，时间明显变短很多

=====

RIP 实验 4：RIP 抑制接口及单播更新

RIP 支持抑制接口的配置，即配置后禁止接口发送更新报文，

但此接口所在网段的路由可以发布出去。

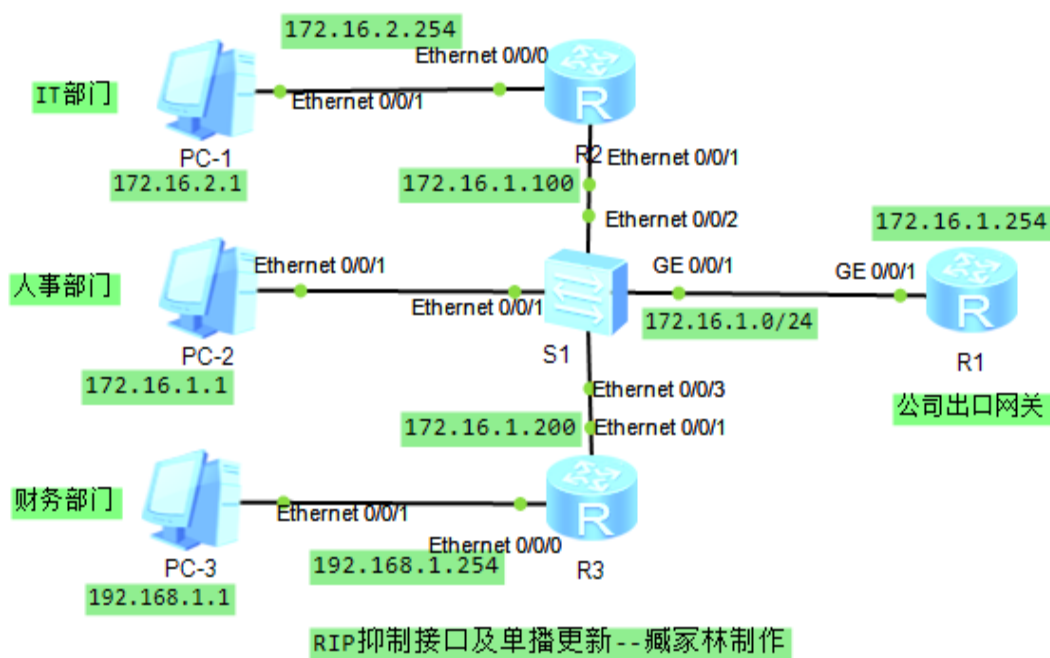
两种方法

进程下 `silent-interface`

接口下 `undo rip output`

`silent-interface` 的优先级大于在接口下配置的 `undo rip output`, 禁止接口接收 RIP 报文，这也是预防路由环路的一种方式。

单播更新是指 RIP 使用单播发送 RIP 报文。在默认情况下，RIP 每隔 30s 以广播或组播方式交换整个路由表的信息，这将耗费大量网络带宽，特别在广域网上，可以出现严重性能问题。当使用 `silent-interface` 命令配置抑制接口后，再指定单播更新的目的地址后，单播更新有效。如果在接口下使用 `undo rip output` 命令来配置抑制接口，即使再指定单播更新的目的地址也是无法更新。



1. 基础配置

R1:

undo ter mo

sy

sys R1

int g0/0/1

ip add 172.16.1.254 24

q

rip

version 2

network 172.16.0.0

R2:

undo ter mo

sy

sys R2

int e0/0/1

ip add 172.16.1.100 24

int e0/0/0

ip add 172.16.2.254 24

q

rip

version 2

network 172.16.0.0

R3:

undo ter mo

sy

sys R3

int e0/0/1

ip add 172.16.1.200 24


```

int e0/0/0
ip add 192.168.1.254 24
q

rip
version 2
network 172.16.0.0
network 192.168.1.0

```

正常配置完成后，R1 上能看到 R2 R3 上的路由

```

[R1]display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
                Destinations : 9                Routes : 9

Destination/Mask    Proto    Pre  Cost    Flags
-----
127.0.0.0/8         Direct   0    0        D
127.0.0.1/32        Direct   0    0        D
127.255.255.255/32  Direct   0    0        D
172.16.1.0/24        Direct   0    0        D
172.16.1.254/32     Direct   0    0        D
172.16.1.255/32     Direct   0    0        D
172.16.2.0/24       RIP      100   1        D
192.168.1.0/24      RIP      100   1        D
255.255.255.255/32  Direct   0    0        D

```

在 PC-2 的接口上抓包，可以看到收到很多无用的 RIP 路由更新

10	19.640000	HuaweiTe_a0:01:6b	Spanning-tree-(for-STP	MST. Root =
11	21.859000	HuaweiTe_a0:01:6b	Spanning-tree-(for-STP	MST. Root =
12	24.031000	HuaweiTe_a0:01:6b	Spanning-tree-(for-STP	MST. Root =
13	25.218000	172.16.1.200	255.255.255.255	RIPv1 Response
14	25.218000	255.255.255.255	172.16.1.200	ICMP Destination
15	26.250000	HuaweiTe_a0:01:6b	Spanning-tree-(for-STP	MST. Root =
16	27.828000	172.16.1.100	255.255.255.255	RIPv1 Response
17	27.828000	255.255.255.255	172.16.1.100	ICMP Destination
18	28.484000	HuaweiTe_a0:01:6b	Spanning-tree-(for-STP	MST. Root =
19	30.640000	HuaweiTe_a0:01:6b	Spanning-tree-(for-STP	MST. Root =

PC-1 和 PC-3 也一样，会收到 RIP 路由信息

2.配置 RIP 抑制接口，优化网络

为了减少对带宽和资源的浪费，不让 PC 收到大量无关的 RIP

报文，可以受用抑制接口的方法来实现。使用路由器的接口只接收 RIP 更新报文，而不发送更新报文。

2.配置抑制接口

R1:

rip

silent-interface GigabitEthernet0/0/1

R2:

rip

silent-interface Ethernet1/0/0

silent-interface Ethernet1/0/1

R3:

rip

silent-interface Ethernet1/0/0

silent-interface Ethernet1/0/1

display rip 1 查看路由器上，哪些接口为抑制接口

```

[R2]display rip 1
Public VPN-instance
  RIP process : 1
    RIP version      : 1
    Preference       : 100
    Checkzero        : Enabled
    Default-cost     : 0
    Summary          : Enabled
    Host-route       : Enabled
    Maximum number of balanced paths : 8
    Update time      : 30 sec
    Age time         : 180 sec
    Garbage-collect time : 120 sec
    Graceful restart : Disabled
    BFD              : Disabled
    Silent-interfaces :
      Ethernet1/0/0
      Ethernet1/0/1
    Default-route    : Disabled
    Verify-source     : Enabled

```

在 PC 上抓包，已经接收不到 RIP 的路由更新

12	24.188000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
13	26.375000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
14	28.547000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
15	30.688000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
16	32.860000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
17	35.125000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
18	37.297000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
19	39.469000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
20	41.688000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
21	43.828000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
22	45.985000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
23	48.156000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3
24	50.297000	HuaweiTe_a0:01:6b	Spanning-tree-(for- STP	MST. Root = 3

reset ip routing-table statistics protocol rip

R1 查看路由表，没有通过 RIP 学习的路由

3.配置单播更新

上面完成了对网络的优化，使 PC 收不到无用的 RIP 路由更新，但各路由器之间也无法相互访问，路由表中没有其它网络的路由信息。

将路由器接口配置成抑制接口后，接口将无法以广播或组播的方式发送 RIP 更新报文。

现在为了让 RIP 网络能够正常通信，可以增加 RIP 的单播更新配置来实现，RIP 更新会以单播的形式发送。

要双向指邻居，单向指，只会单向发路由

R2 指向 R1，但 R1 没有指 R2，这们配置 R1 上会有 R2 的路由，但 R2 上没有 R1 的路由

R2 指向 R1,意味着 R2 把自己的路由发送给 R1

R2 :

```
rip
```

```
peer 172.16.1.254
```

双向配置

R1 :

```
rip
```

```
peer 172.16.1.100
```

```
peer 172.16.1.200
```

R2 :

```
rip
```

```
peer 172.16.1.254
```

```
peer 172.16.1.200
```

R3 :

```
rip
```

```
peer 172.16.1.254
```

```
peer 172.16.1.100
```

display rip 1 查看 RIP 配置信息

```
Graceful restart : Disabled
BFD : Disabled
Silent-interfaces :
GigabitEthernet0/0/1
Default-route : Disabled
Verify-source : Enabled
Networks :
172.16.0.0
Configured peers :
172.16.1.200 172.16.1.100
Number of routes in database : 3
```

查看路由表

```
[R1]display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
                Destinations : 9                Routes : 9

Destination/Mask    Proto    Pre  Cost           Flags NextHop
-----
    127.0.0.0/8      Direct   0    0             D    127.0.0.1
    127.0.0.1/32     Direct   0    0             D    127.0.0.1
127.255.255.255/32   Direct   0    0             D    127.0.0.1
    172.16.1.0/24     Direct   0    0             D    172.16.1.254
    172.16.1.254/32   Direct   0    0             D    127.0.0.1
    172.16.1.255/32   Direct   0    0             D    127.0.0.1
    172.16.2.0/24     RIP      100    1             D    172.16.1.100
    192.168.1.0/24    RIP      100    1             D    172.16.1.200
255.255.255.255/32   Direct   0    0             D    127.0.0.1
```

现在每台路由器都已经拥有了所有网段的路由信息。现在网络通信恢复正常，所有 PC 也不会接收不 RIP 报文。

4.undo rip output

RIP 协议中还可以通过 undo rip output 命令来配置抑制接口，禁止接口发送 RIP 报文。

只以 R3 为例做个演示，一般不使用 undo rip output

删除 R3 在上面的配置

R3:

rip

undo silent-interface e1/0/1

undo silent-interface e1/0/0

undo peer 172.16.1.100

undo peer 172.16.1.254

配置 undo rip output

R3 :

interface Ethernet1/0/1

undo rip output

配置完成后，等一会，RIP 有一个超时计时器 180s

```
<R1>display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 8           Routes : 8

Destination/Mask    Proto   Pre  Cost   Flags NextHop
-----
      127.0.0.0/8     Direct   0    0        D  127.0.0.1
      127.0.0.1/32     Direct   0    0        D  127.0.0.1
127.255.255.255/32   Direct   0    0        D  127.0.0.1
      172.16.1.0/24    Direct   0    0        D  172.16.1.254
      172.16.1.254/32  Direct   0    0        D  127.0.0.1
      172.16.1.255/32  Direct   0    0        D  127.0.0.1
      172.16.2.0/24    RIP      100    1        D  172.16.1.100
255.255.255.255/32   Direct   0    0        D  127.0.0.1
```

可以看到没有 R3 上的 192.168.1.0 路由条目，说明 undo rip output 生效

在 R3 上配置与 R1 的单播更新

R3 :

rip

peer 172.16.1.254

配置完成后，等一会，路由表收敛后再在 R1 上查看路由表

```
<R1>display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 8           Routes : 8

Destination/Mask    Proto   Pre  Cost   Flags NextHop
-----
      127.0.0.0/8     Direct   0    0       D   127.0.0.1
      127.0.0.1/32     Direct   0    0       D   127.0.0.1
127.255.255.255/32   Direct   0    0       D   127.0.0.1
      172.16.1.0/24    Direct   0    0       D   172.16.1.254
      172.16.1.254/32  Direct   0    0       D   127.0.0.1
      172.16.1.255/32  Direct   0    0       D   127.0.0.1
      172.16.2.0/24    RIP      100   1       D   172.16.1.100
255.255.255.255/32   Direct   0    0       D   127.0.0.1
```

可以看到 R1 上仍然没有 192.168.1.0 路由。由此可以证明使用 undo rip output 命令来抑制接口，即使配置了单播更新，路由也是无法以单播形式发送路由更新的。

silent-interface 优先级大于 undo rip output

允许接口接收 RIP 更新报文 rip input

禁止接口接收 RIP 更新报文 undo rip input

允许接口发送 RIP 更新报文 rip output

禁止接口发送 RIP 更新报文 undo rip output

打开 RIP 的报文调试信息开关 debugging rip packets [interface type number]

打开 RIP 的接收报文情况调试开关 debugging rip receive

打开 RIP 的发送报文情况调试开关 debugging rip send

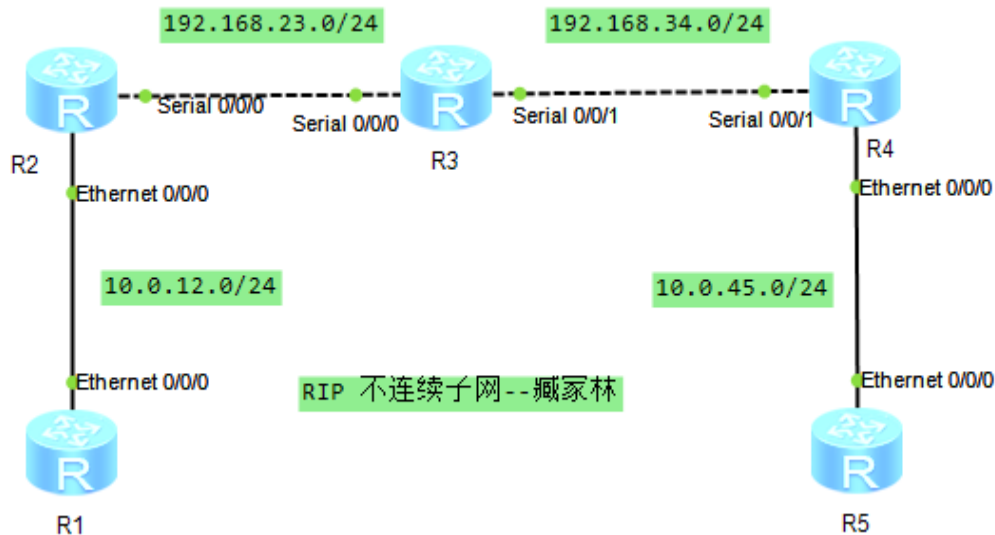
=====

RIP 实验 5：RIP 不连续子网

RIP 会在主网络边界自动汇总，当汇总发生时，汇总的子网路由在边界处被抑制掉，而仅通告主网路由。如果主网不连续，被其他主网所分隔，主网边界的自动汇总就会存在问题。

连续子网是指所相连的子网属于同一主网

不连续子网是指相同主网下的子网被另一主网分隔



1. 基本配置

```
R1 :  
undo terminal monitor  
sys  
sysname R1  
int e0/0/0  
ip add 10.0.12.1 24  
q
```


R2 :
undo terminal monitor
sys
sysname R2
int e0/0/0
ip add 10.0.12.2 24
int s0/0/0
ip add 192.168.23.2 24
q

R3 :
undo terminal monitor
sys
sysname R3
int s0/0/1
ip add 192.168.34.3 24
int s0/0/0
ip add 192.168.23.3 24
q

R4 :
undo terminal monitor
sys
sysname R4
int s0/0/1
ip add 192.168.34.4 24
int e0/0/0
ip add 10.0.45.4 24
q

R5 :
undo terminal monitor
sys
sysname R5

```
int e0/0/0
ip add 10.0.45.5 24
q
```

=====

运行 RIPv1

```
R1 :
rip
net 10.0.0.0
```

```
R2:
rip
net 10.0.0.0
net 192.168.23.0
```

```
R3:
rip
net 192.168.23.0
net 192.168.34.0
```

```
R4:
rip
net 10.0.0.0
net 192.168.34.0
```

```
R5:
rip
net 10.0.0.0
```

在 R1 上查看路由表

R1 : 没有路由 10.0.45.0

```
[R1]dis ip rout
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 9           Routes : 9

Destination/Mask    Proto   Pre  Cost           Flags NextHop
-----
      10.0.12.0/24   Direct  0    0             D    10.0.12.1
      10.0.12.1/32   Direct  0    0             D    127.0.0.1
      10.0.12.255/32 Direct  0    0             D    127.0.0.1
      127.0.0.0/8    Direct  0    0             D    127.0.0.1
      127.0.0.1/32   Direct  0    0             D    127.0.0.1
127.255.255.255/32   Direct  0    0             D    127.0.0.1
      192.168.23.0/24 RIP     100   1             D    10.0.12.2
      192.168.34.0/24 RIP     100   2             D    10.0.12.2
255.255.255.255/32   Direct  0    0             D    127.0.0.1
```

R3 查看路由表

```
[R3]display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 13          Routes : 14

Destination/Mask    Proto   Pre  Cost           Flags NextHop
-----
      10.0.0.0/8     RIP     100   1             D    192.168.23.2
      10.0.0.0/8     RIP     100   1             D    192.168.34.4
      127.0.0.0/8    Direct  0     0             D    127.0.0.1
      127.0.0.1/32   Direct  0     0             D    127.0.0.1
127.255.255.255/32   Direct  0     0             D    127.0.0.1
      192.168.23.0/24 Direct  0     0             D    192.168.23.3
      192.168.23.2/32 Direct  0     0             D    192.168.23.2
      192.168.23.3/32 Direct  0     0             D    127.0.0.1
```

R3 从 R2 R4 分别收到了 10.0.12.0 和 10.0.45.0 的路由条目，默认打开了自动有类汇总功能，所以在主网边界向外发送路由信息的时候都汇总成了 10.0.0.0/8，发送给 R3，最终在 R3 上收到了两条目的网段相同，cost 值相同的路由条目。

R3 测试与 R1 和 R5 的连通性

```
[R3]ping 10.0.45.5
PING 10.0.45.5: 56 data bytes, press CTRL_C to break
Request time out
Reply from 10.0.45.5: bytes=56 Sequence=2 ttl=254 time=40 ms
Reply from 10.0.45.5: bytes=56 Sequence=3 ttl=254 time=20 ms
Reply from 10.0.45.5: bytes=56 Sequence=4 ttl=254 time=20 ms
Reply from 10.0.45.5: bytes=56 Sequence=5 ttl=254 time=20 ms

[R3]ping 10.0.12.1
PING 10.0.12.1: 56 data bytes, press CTRL_C to break
Request time out
Request time out
Request time out
Request time out
Request time out
```

发现呈现出有一端无法通信的现象。

现在每台设备上的路由表中没有清晰地反馈出拓扑中的真实子网信息，这是由于在 RIPv1 默认自动汇总开启的情况下，设计网络时没有遵循主网的子网应连接这一要求。

2.RIPv1 中解决不连接子网问题

如果把不连续子网转变成连续的子网，问题就可以解决，办法是给接口配置第二个 IP 地址，IP 地址取 10.0.0.0/8 主网的子网。

在路由器上配置从 IP 地址，只要在常规配置 IP 地址的命令之后加上 sub 参数即可

R2 :

```
int s0/0/0
```

```
ip add 10.0.23.2 24 sub
```

R3 :

```
int s0/0/0
```

```
ip add 10.0.23.3 24 sub
```

```
int s0/0/1
```

```
ip add 10.0.34.3 24 sub
```

```
R4 :  
int s0/0/1  
ip add 10.0.34.4 24 sub
```

R3 RIP 中再加入 10.0.0.0 网段

```
R3 :  
rip  
net 10.0.0.0
```

这样配置完成后，相当于原先被孤立的两个不连续子网，被新添加的 10.0.23.0 和 10.0.34.0 连接起来了，即现在已经构成了一个连续的子网。

```
<R3>display ip routing-table  
Route Flags: R - relay, D - download to fib  
-----  
Routing Tables: Public  
      Destinations : 21          Routes : 22  
  
Destination/Mask    Proto    Pre  Cost    Flags NextHop  
-----  
      10.0.0.0/8     RIP      100   1        D   192.168.23.2  
                   RIP      100   1        D   192.168.34.4  
      10.0.12.0/24   RIP      100   1        D   10.0.23.2  
      10.0.23.0/24   Direct    0     0        D   10.0.23.3  
      10.0.23.3/32   Direct    0     0        D   127.0.0.1  
      10.0.23.255/32 Direct    0     0        D   127.0.0.1  
      10.0.34.0/24   Direct    0     0        D   10.0.34.3  
      10.0.34.3/32   Direct    0     0        D   127.0.0.1  
      10.0.34.255/32 Direct    0     0        D   127.0.0.1  
      10.0.45.0/24   RIP      100   1        D   10.0.34.4
```

R3 测试与 R1 和 R5 的连通性，现在都是可以通的

```
<R3>ping 10.0.12.1
```

```
<R3>ping 10.0.45.5
```

上面是通过在不连续子网之间的链路上配置相同主网的子网 IP 地址，即采用配置从 IP 地址的方式来实现子网的连续性，此种做法的优点是 RIPv1 在不做大的拓扑调整的前提下仅靠第二个 IP 地址就解决了不连续子网问题，不足之处是需要配置第二个 IP 地址，要消耗掉多个子网网段。

3.RIPv2 解决不连续子网

如果路由器运行的是 RIPv2，则可以直接关闭自动汇总，子网是否连续就不重要了，因为 RIPv2 会直接通告相应的子网路由。

删除上面的从 IP 地址配置

R2：

```
int s0/0/0  
undo ip add 10.0.23.2 24 sub
```

R3：

```
int s0/0/0  
undo ip add 10.0.23.3 24 sub  
int s0/0/1  
undo ip add 10.0.34.3 24 sub
```

R4：

```
int s0/0/1  
undo ip add 10.0.34.4 24 sub
```

查看 R3 的路由表，只能识别 10.0.0.0/8

```

<R3>dis ip rout
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
                Destinations : 13          Routes : 14

Destination/Mask    Proto    Pre  Cost    Flags NextHop
-----
 10.0.0.0/8         RIP      100   1       D    192.168.23.2
                    RIP      100   1       D    192.168.34.4
 127.0.0.0/8        Direct    0     0       D    127.0.0.1
 127.0.0.1/32       Direct    0     0       D    127.0.0.1
127.255.255.255/32  Direct    0     0       D    127.0.0.1

```

运行 RIPv2，关闭自动汇总

R1 :
 rip
 version 2
 undo summary

R2 :
 rip
 version 2
 undo summary

R3 :
 rip
 version 2
 undo summary

R4 :
 rip
 version 2
 undo summary

配置完成后，查看 R3 的路由表，有明细路由，也是可以 ping 通的

```

<R3>display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 14          Routes : 14

Destination/Mask    Proto   Pre  Cost   Flags NextHop
-----
10.0.12.0/24       RIP     100   1       D    192.168.23.2
10.0.45.0/24       RIP     100   1       D    192.168.34.4
127.0.0.0/8        Direct   0     0       D    127.0.0.1
127.0.0.1/32       Direct   0     0       D    127.0.0.1

```

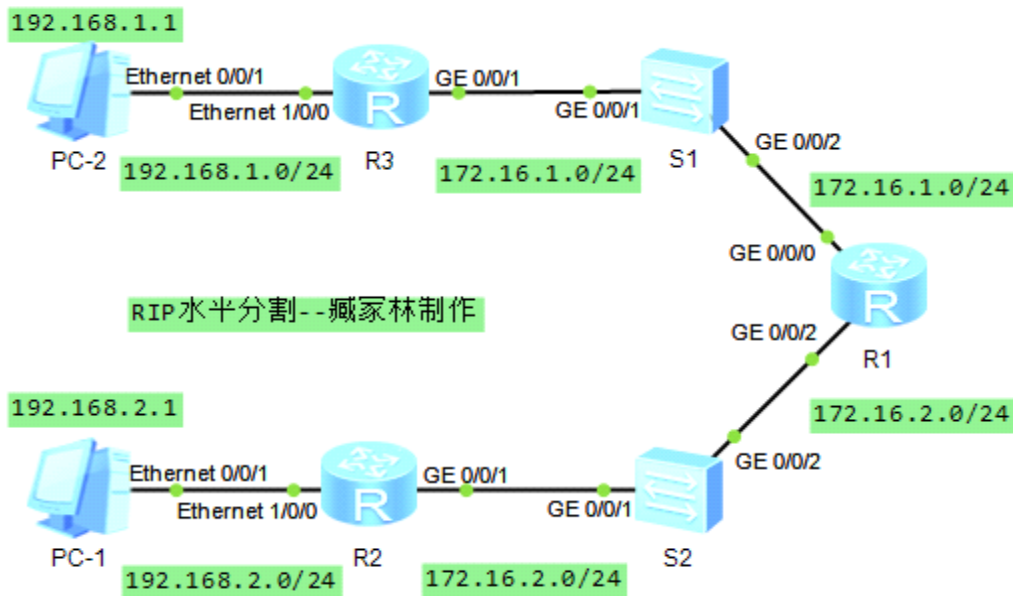
=====

RIP 实验 6 : RIP 水平分割

水平分割 (Split Horizon) 指的是 RIP 从某个接口收到的路由信息，不会从该接口再发回给邻居设备。这样不但减少了带宽消耗，还可以防止路由环路。在华为设备上，水平分割功能默认情况下是开启的。帧中继网络中，水平分割默认是禁止的。

触发更新 (Triggered Updates) 的原理是，当路由信息发生变化时，运行 RIP 的设备会立即向邻居设备发送更新报文，而不必等待定时更新，从而缩短了网络收敛时间。在华为设备上，没有相关命令能关闭触发更新。

毒性逆转 (Poison Reverse) 指的是 RIP 从某个接口接收到路由信息后，将该路由的 cost 值设置为 16，并从原接口发回邻居设备。利用这种方式，可以消除对方路由表中无用的路由，而不必等待老化时间。如果同时配置了毒性逆转和水平分割，毒性逆转生效。在华为设备上毒性逆转功能默认是关闭的，需要手动打开此功能。



1.基本配置，运行 RIPv2

R1 :

un ter mo

sy

sys R1

int g0/0/0

ip add 172.16.1.1 24

int g0/0/2

ip add 172.16.2.1 24

rip

version 2

net 172.16.0.0

q

R2 :

un ter mo

sy

sys R2

```
int g0/0/1
ip add 172.16.2.2 24
int e0/0/0
ip add 192.168.2.2 24
rip
version 2
net 172.16.0.0
net 192.168.2.0
q
```

```
R3 :
un ter mo
sy
sys R3
int g0/0/1
ip add 172.16.1.3 24
int e0/0/0
ip add 192.168.1.3 24
rip
version 2
net 172.16.0.0
net 192.168.1.0
q
```

配置完成后，查看 R1 的路由表，所有网段都已经学习到

2. 验证触发更新

断掉 R3 与 S1 之间的链路，模拟器直接删除线，然后查看 R1 的路由表，192.168.1.0 仍然存在，这是因为断掉的不是 R1 的直连接口，R1 此时无法直接感知故障的发生，路由条目等 180s 的老化计时器超时后，才会删除 R1 R2 的路由，还有 192.168.1.0

```
[R1]dis ip routing-table
Route Flags: R - relay, D - download to fib
```

```
-----
Routing Tables: Public
      Destinations : 8          Routes : 8
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
127.0.0.0/8	Direct	0	0	D	127.0.0.1
127.0.0.1/32	Direct	0	0	D	127.0.0.1
172.16.1.0/24	Direct	0	0	D	172.16.1.1
172.16.1.1/32	Direct	0	0	D	127.0.0.1
172.16.2.0/24	Direct	0	0	D	172.16.2.1
172.16.2.1/32	Direct	0	0	D	127.0.0.1
192.168.1.0/24	RIP	100	1	D	172.16.1.3
192.168.2.0/24	RIP	100	1	D	172.16.2.2

等 180 之后，在路由表中没有，在数据库表中还是有的，cost 值为 16

```
[R1]dis rip 1 database
```

```
-----
Advertisement State : [A] - Advertised
                     [I] - Not Advertised/withdraw
-----
```

```
172.16.0.0/16, cost 0, ClassfulSumm
  172.16.1.0/24, cost 0, [A], Rip-interface
  172.16.2.0/24, cost 0, [A], Rip-interface
192.168.1.0/24, cost 16, ClassfulSumm
192.168.1.0/24, cost 16, [I], nexthop 172.16.1.3
192.168.2.0/24, cost 1, ClassfulSumm
192.168.2.0/24, cost 1, [A], nexthop 172.16.2.2
```

接口打开,并在 R2 的路由表正常后，断掉 R1 与 S1 之间的链路，查看 R2 的路由表

```
[R2]dis ip routing-table
Route Flags: R - relay, D - download to fib
```

```
-----
Routing Tables: Public
      Destinations : 6           Routes : 6
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
127.0.0.0/8	Direct	0	0	D	127.0.0.1
127.0.0.1/32	Direct	0	0	D	127.0.0.1
172.16.2.0/24	Direct	0	0	D	172.16.2.2
172.16.2.2/32	Direct	0	0	D	127.0.0.1
192.168.2.0/24	Direct	0	0	D	192.168.2.2
192.168.2.2/32	Direct	0	0	D	127.0.0.1

可以看到 192.168.1.0 已经没有，因为断掉的是 R1 的直连接口，所以 R1 能够直接感知到链路发生故障，在路由表中删除，并会触发更新，使得 R2 上的路由表为最新状态。

3.验证水平分割

路由器运行 RIP 默认开启水平分割

在 R2 上打开 debug 功能，查看 R2 发送给 R1 的路由条目

```
<R2>debugging rip 1 send g0/0/1
```

```
<R2>terminal monitor
```

```
<R2>terminal debugging
```

```
Jun  1 2018 15:34:55.480.1-08:00 R2 RIP/7/DBG: 6: 12227: RIP 1: Sending response
on interface GigabitEthernet0/0/1 from 172.16.2.2 to 224.0.0.9
Jun  1 2018 15:34:55.480.2-08:00 R2 RIP/7/DBG: 6: 12247: Packet: Version 2, Cmd
response, Length 24
Jun  1 2018 15:34:55.480.3-08:00 R2 RIP/7/DBG: 6: 12315: Dest 192.168.2.0/24, Ne
xthop 0.0.0.0, Cost 1, Tag 0
<R2>
```

可以看到 R2 发送给 R1 的路由条目中没有包含 192.168.1.0 ,
因为该路由条目是从 R1 始发过来的

下面关闭 debug,在 R2 和 R1 上关闭水平分割

```
<R2>undo debugging all
```

R2 :

```
int g0/0/1
```

```
undo rip split-horizon
```

R1 :

```
int g0/0/2
```

```
undo rip split-horizon
```

配置完成后 , 查看 debug 信息 <R2>debugging rip 1
send g0/0/1

```
Jun  1 2018 15:36:23.840.3-08:00 R2 RIP/7/DBG: 6: 12315: Dest 172.16.0.0/16, Nex
thop 0.0.0.0, Cost 1, Tag 0
Jun  1 2018 15:36:23.840.4-08:00 R2 RIP/7/DBG: 6: 12315: Dest 192.168.1.0/24, Ne
xthop 0.0.0.0, Cost 3, Tag 0
Jun  1 2018 15:36:23.840.5-08:00 R2 RIP/7/DBG: 6: 12315: Dest 192.168.2.0/24, Ne
xthop 0.0.0.0, Cost 1, Tag 0
<R2>
```

从 debug 信息中可以看到 R2 发给 R1 的路由条目中包含了 192.168.1.0 网段，此时接口上的水平分割功能不生效。

4.验证毒性逆转

恢复 R2 g0/0/1 接口下的水平分割功能。

```
R2 :  
int g0/0/1  
rip split-horizon
```

在 R2 接口下开启毒性逆转功能

```
R2 :  
int g0/0/1  
rip poison-reverse
```

查看都是 enabled 开启的

```
[R2]dis rip 1 int g0/0/1 verbose  
GigabitEthernet0/0/1(172.16.2.2)  
State           : UP           MTU       : 500  
Metricin        : 0  
Metricout       : 1  
Input           : Enabled      Output    : Enabled  
Protocol        : RIPv2 Multicast  
Send version    : RIPv2 Multicast Packets  
Receive version : RIPv2 Multicast and Broadcast Packets  
Poison-reverse  : Enabled  
Split-Horizon   : Enabled  
Authentication type : None  
Replay Protection : Disabled
```

配置完成后，查看 debug 信息 <R2>debugging rip 1

send g0/0/1

```
Jun  1 2018 15:38:57.510.3-08:00 R2 RIP/7/DBG: 6: 12315: Dest 172.16.1.0/24, Nex  
thop 172.16.2.1, Cost 16, Tag 0  
Jun  1 2018 15:38:57.510.4-08:00 R2 RIP/7/DBG: 6: 12315: Dest 192.168.1.0/24, Ne  
xthop 172.16.2.1, Cost 16, Tag 0  
Jun  1 2018 15:38:57.510.5-08:00 R2 RIP/7/DBG: 6: 12315: Dest 192.168.2.0/24, Ne  
xthop 0.0.0.0, Cost 1, Tag 0  
<R2>
```

可以看到 R2 发给 R1 的路由条目中包含了 192.168.1.0 网段，但 cost 值为 16，说明在毒性逆转和水平分割同时开启的情况下，毒性逆转起作用。

=====

RIP 实验 7：RIP 附加度量值

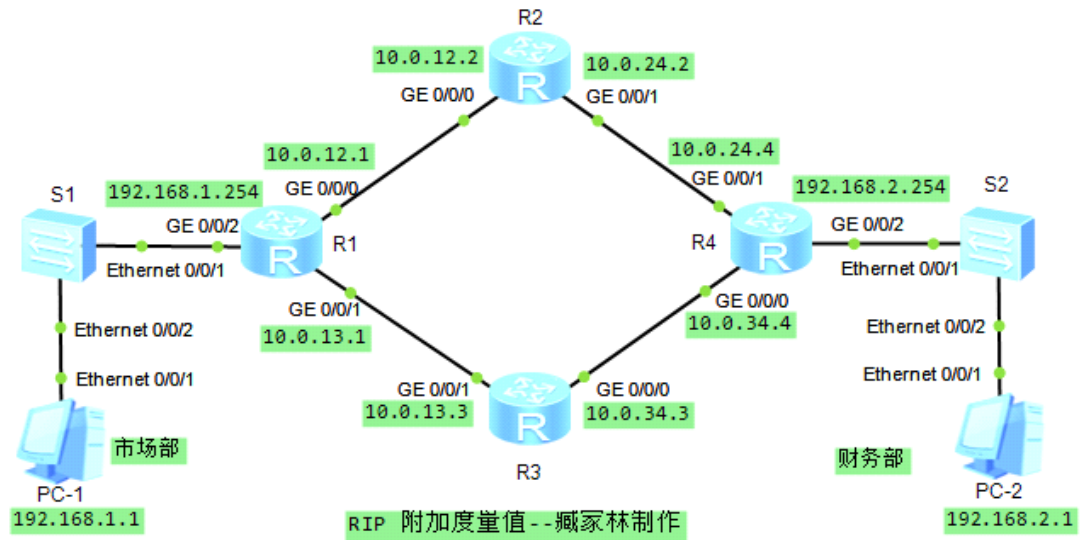
路由附加度量值是在 RIP 路由原来度量值的基础上所增加或减少的度量值（跳数）。对于 RIP 接收和发布路由，可通过不同的命令配置附加度量值更加灵活地控制 RIP 的路由选择。

rip metricin

用于在接收到路由后，给其增加一个附加度量值，再加入路由表中，使得路由表中的度量值发生变化。运行该命令会影响本地设备和其他设备的路由选择。

rip metricout

用于自身路由的发布，发布时增加一个附加的度量值，但本地路由表中的度量值不会发生变化。运行该命令不会影响本地设备的路由选择，但是会影响其他设备的路由选择。



1.基本配置

R1 :

un ter mo

sy

sys R1

int g0/0/0

ip add 10.0.12.1 24

int g0/0/1

ip add 10.0.13.1 24

int g0/0/2

ip add 192.168.1.254 24

q

rip

version 2

net 10.0.0.0

net 192.168.1.0

q

R2 :

un ter mo


```
sy
sys R2
int g0/0/0
ip add 10.0.12.2 24
int g0/0/1
ip add 10.0.24.2 24
q
```

```
rip
version 2
net 10.0.0.0
q
```

```
R3 :
un ter mo
sy
sys R3
int g0/0/0
ip add 10.0.34.3 24
int g0/0/1
ip add 10.0.13.3 24
q
```

```
rip
version 2
net 10.0.0.0
q
```

```
R4 :
un ter mo
sy
sys R4
int g0/0/0
ip add 10.0.34.4 24
```

```

int g0/0/1
ip add 10.0.24.4 24
int g0/0/2
ip add 192.168.2.254 24
q

```

```

rip
version 2
net 10.0.0.0
net 192.168.2.0
q

```

R1 到 192.168.2.0 有两条路径

192.168.1.0/24	Direct	0	0	D	192.168.1.254
192.168.1.254/32	Direct	0	0	D	127.0.0.1
192.168.1.255/32	Direct	0	0	D	127.0.0.1
192.168.2.0/24	RIP	100	2	D	10.0.13.3
	RIP	100	2	D	10.0.12.2

现在是负载均衡的

2.配置 RIP metricin

修改附加度量值,设置 R1 在接收 R3 发送来的路由条目时增加度量值 2 , 这样 R1 会优选 R2 发来的 RIP 路由条目 , 并加入到路由表中。

```

R1:
int g0/0/1
rip metricin 2

```

display ip routing R1 查看一下路由表 , 现在选择的是 R2 过来的路由

```

192.168.1.254/32 Direct 0 0 D 127.0.0.1
192.168.1.255/32 Direct 0 0 D 127.0.0.1
192.168.2.0/24 RIP 100 2 D 10.0.12.2

```

display rip 1 database 查看 R1 的数据库，下一跳为 R3 的路由，cost 值被设置为 16，即不可达

```

192.168.1.0/24, cost 0, ClassfulSumm
192.168.1.0/24, cost 0, [A], Rip-interface
192.168.2.0/24, cost 2, ClassfulSumm
192.168.2.0/24, cost 16, [I], nexthop 10.0.13.3
192.168.2.0/24, cost 2, [A], nexthop 10.0.12.2
[R1]

```

3.配置 RIP metricout

为了减轻 R2 的负担，所有 PC-2 去往 PC-1 的流量都由 R3 来转发

在 R4 上查看，默认是负载分担的

```

10.0.34.255/32 Direct 0 0 D 127.0.0.1
127.0.0.0/8 Direct 0 0 D 127.0.0.1
127.0.0.1/32 Direct 0 0 D 127.0.0.1
.255.255.255/32 Direct 0 0 D 127.0.0.1
192.168.1.0/24 RIP 100 2 D 10.0.34.3
RIP 100 2 D 10.0.24.2
192.168.2.0/24 Direct 0 0 D 192.168.2.254

```

在 R2 的 g0/0/1 接口下使用 rip metricout 3,设置 R2 在向 R4 发送路由条目时增加度量值 3，这样 R4 收到来自 R2 的路由度量值就会大于来自 R3 的路由，R4 会选择从 R3 走

R2:

```
int g0/0/1
```

rip metricout 3

R4 查看路由表，选择从 R3 去往 192.168.1.0

10.0.34.255/32	Direct	0	0	D	127.0.0.1
127.0.0.0/8	Direct	0	0	D	127.0.0.1
127.0.0.1/32	Direct	0	0	D	127.0.0.1
.255.255.255/32	Direct	0	0	D	127.0.0.1
192.168.1.0/24	RIP	100	2	D	10.0.34.3
192.168.2.0/24	Direct	0	0	D	192.168.2.254

PC-2 做路由追踪

PC>tracert 192.168.1.1

```
PC>tracert 192.168.1.1

tracert to 192.168.1.1, 8 hops max
(ICMP), press Ctrl+C to stop
 1  192.168.2.254    62 ms  16 ms  31 ms
 2  *10.0.34.3      47 ms  31 ms
 3  * *10.0.13.1    16 ms
 4  *192.168.1.1    94 ms  62 ms

PC>
```

可以看到，数据包是经过 R3 转发到 PC-1

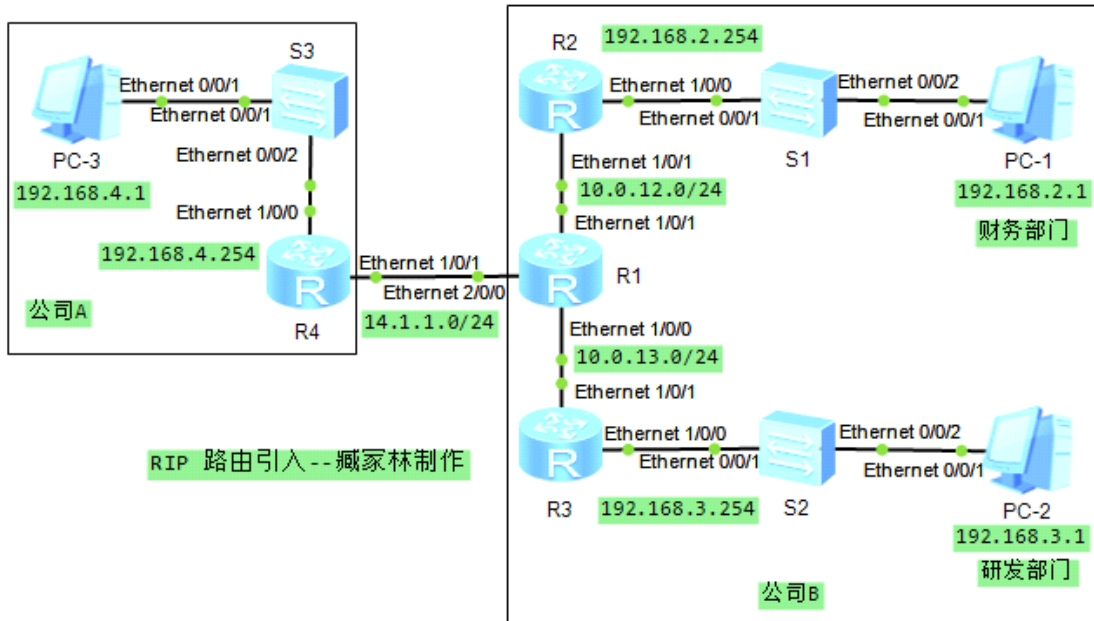
=====

RIP 实验 8：RIP 路由引入

在规划网络时，一般都设计成仅运行一种路由协议，以降低网络复杂性，易于维护。但是如果在网络升级、扩展或合并时，就可以造成在网络中同时运行几种不同的路由协议，这时就需要部署路由协议间的引入，使路由信息能够在不同协议间传递。

RIP 支持不同路由协议的引入，包括直连路由、静态路由或其他动态路由协议。由于 RIP 的度量值是跳数且最大值不能超

过 15，所以在将其他路由协议引入至 RIP 时需要注意设置度量值，避免引入的路由度量值超过 15。



1. 基本配置，运行 RIPv2

R1:

```
un ter mo
```

```
sy
```

```
sys R1
```

```
int e1/0/0
```

```
ip add 10.0.13.1 24
```

```
int e1/0/1
```

```
ip add 10.0.12.1 24
```

```
int e2/0/0
```

```
ip add 14.1.1.1 24
```

R2:

```
un ter mo
```

```
sy
```

```
sys R2
```

```
int e1/0/0
```

```
ip add 192.168.2.254 24
```

```
int e1/0/1
ip add 10.0.12.2 24
```

```
R3:
un ter mo
sy
sys R3
int e1/0/0
ip add 192.168.3.254 24
int e1/0/1
ip add 10.0.13.3 24
```

```
R4:
un ter mo
sy
sys R4
int e1/0/0
ip add 192.168.4.254 24
int e1/0/1
ip add 14.1.1.4 24
```

配置 RIP 协议

```
R1:
rip
version 2
undo summary
net 10.0.0.0
```

```
R2:
rip
version 2
undo summary
net 10.0.0.0
net 192.168.2.0
```

```
R3:
rip
```

```
version 2
undo summary
net 10.0.0.0
net 192.168.3.0
```

配置完成后 R1 学习到了 192.168.2.0 和 192.168.3.0

2. 优化 RIP 网络，引入直连路由

在 R2 的 e1/0/0 抓取数据包，接口连接的是 PC

3	4.390000	HuaweiTe_7f:07:04	Spanning-tree-(for-STP	MST. Root =
4	6.593000	HuaweiTe_7f:07:04	Spanning-tree-(for-STP	MST. Root =
5	8.797000	HuaweiTe_7f:07:04	Spanning-tree-(for-STP	MST. Root =
6	9.625000	192.168.2.254	224.0.0.9	RIPv2 Response
7	11.062000	HuaweiTe_7f:07:04	Spanning-tree-(for-STP	MST. Root =
8	13.265000	HuaweiTe_7f:07:04	Spanning-tree-(for-STP	MST. Root =

可以看到，此时 R2 连接的是 PC，而这些 RIP 报文对 PC 来说是毫无用处的。原因是使用 network 命令通告 192.168.2.0 网段后，R2 的 e1/0/0 接口就会收到 RIP 协议报文，不管对端是否利用。

为了让 PC 不接收这些无用的 RIP 报文，可以采用引入直连路由的方式来代替通告路由。使用 import-route 配置路由引入，指定引入的源路由协议为直连路由。

在一台设备上配置路由引入，需要保证被引入的路由条目已经存在于当前路由表中。

```
R2 :
rip
undo net 192.168.2.0
import-route direct
```

配置完成后，R1 的路由表，同样可以学习到 192.168.2.0 路由，同时 PC 不再收到与其无关的 RIP 更新报文。

同理 R3 也可以进行相应的修改

3.引入静态路由

公司 A 要与公司 B 互相访问。

在 R1 上配置静态路由到达 R4，并在 RIP 进程中引入该条静态路由。

R1:

```
ip route-static 192.168.4.0 24 14.1.1.4
rip
import-route static
```

在 R4 上配置一条默认路由，下一跳为 R1

R4:

```
ip route-static 0.0.0.0 0.0.0.0 14.1.1.1
```

R2 R3 路由表中，有 192.168.4.0 路由

```
127.255.255.255/32 Direct 0 0 D 127.0.0.1
192.168.2.0/24 Direct 0 0 D 192.168.2.254
192.168.2.254/32 Direct 0 0 D 127.0.0.1
192.168.2.255/32 Direct 0 0 D 127.0.0.1
192.168.3.0/24 RIP 100 2 D 10.0.12.1
192.168.4.0/24 RIP 100 1 D 10.0.12.1
255.255.255.255/32 Direct 0 0 D 127.0.0.1
<R2>
```

3 台 PC 之间也可以相互 ping 通

PC1 ping PC3

PC>ping 192.168.4.1

```
PC>ping 192.168.4.1

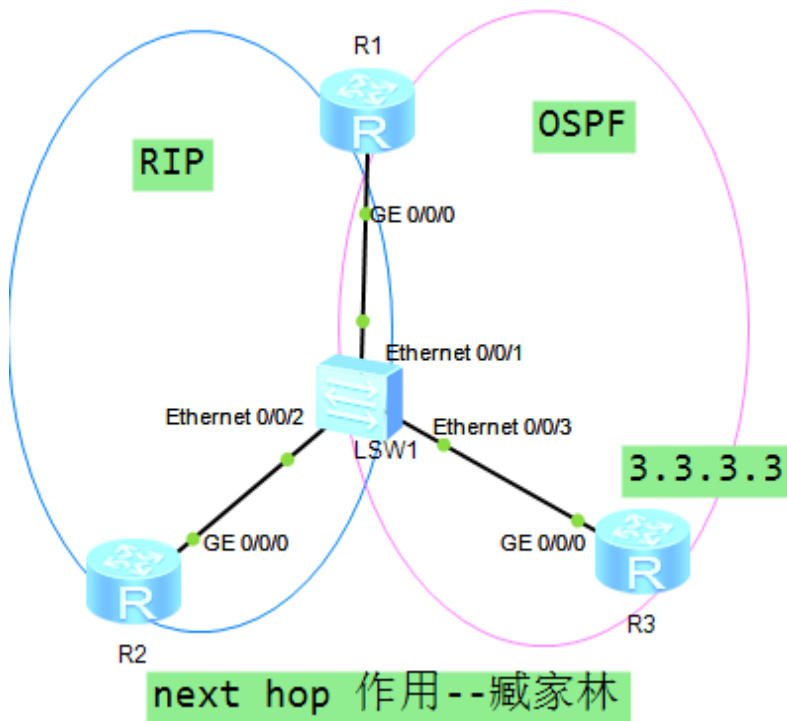
Ping 192.168.4.1: 32 data bytes, Press Ctrl_C to break
From 192.168.4.1: bytes=32 seq=1 ttl=125 time=63 ms
From 192.168.4.1: bytes=32 seq=2 ttl=125 time=93 ms
From 192.168.4.1: bytes=32 seq=3 ttl=125 time=79 ms
From 192.168.4.1: bytes=32 seq=4 ttl=125 time=93 ms
From 192.168.4.1: bytes=32 seq=5 ttl=125 time=47 ms
```

公司 A 和公司 B 之间可以正常通信。

=====

next hop 作用

RIP v1 没有 next hop 字段，RIP v2 有 next hop，那么 next hop 有什么作用？



基本 IP 地址配置

```
R1 :
undo ter mo
sy
sys R1
int loo0
ip add 1.1.1.1 24
int g0/0/0
ip add 192.168.123.1 24
q
```

```
R2 :
undo ter mo
sy
sys R2
int loo0
ip add 2.2.2.2 24
int g0/0/0
ip add 192.168.123.2 24
q
```

```
R3 :
undo ter mo
sy
sys R3
int loo0
ip add 3.3.3.3 24
int g0/0/0
ip add 192.168.123.3 24
q
```

=====

配置路由协议

```
R1 :
rip
net 1.0.0.0
net 192.168.123.0
q

ospf router-id 1.1.1.1
area 0
net 192.168.123.1 0.0.0.0
```

q

```
rip
import-route ospf 1
ospf
import-route rip 1
```

R2 :

```
rip
net 2.0.0.0
net 192.168.123.0
q
```

R3 :

```
ospf router-id 3.3.3.3
area 0
net 3.3.3.3 0.0.0.0
net 192.168.123.3 0.0.0.0
q
```

[在 R2 上查看](#)

```
display ip routing-table
display rip 1 route
```

R1 R2 如果运行的 RIP v1, 那么 R2 学习到 R3 的环回口 3.3.3.3 的下一跳是多少
是 R1 的地址 192.168.123.1

```
[R2]dis ip routing-table
Route Flags: R - relay, D - download to fib
```

```
-----
Routing Tables: Public
```

```
Destinations : 8          Routes : 8
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
1.0.0.0/8	RIP	100	1	D	192.168.123.1
2.2.2.0/24	Direct	0	0	D	2.2.2.2
2.2.2.2/32	Direct	0	0	D	127.0.0.1
3.0.0.0/8	RIP	100	1	D	192.168.123.1
127.0.0.0/8	Direct	0	0	D	127.0.0.1
127.0.0.1/32	Direct	0	0	D	127.0.0.1
192.168.123.0/24	Direct	0	0	D	192.168.123.2
192.168.123.2/32	Direct	0	0	D	127.0.0.1

```
=====
```

运行 RIP v2

如果运行的 RIP v2 呢，学习到 R3 的环回口 3.3.3.3 的下一跳是多少

是 R3 的地址 192.168.123.3

```
[R2]dis ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 8          Routes : 8

Destination/Mask    Proto    Pre  Cost           Flags NextHop
-----
      1.1.1.0/24    RIP      100   1             D    192.168.123.1
      2.2.2.0/24    Direct    0     0             D    2.2.2.2
      2.2.2.2/32    Direct    0     0             D    127.0.0.1
      3.3.3.3/32    RIP      100   1             D    192.168.123.3
      127.0.0.0/8    Direct    0     0             D    127.0.0.1
      127.0.0.1/32   Direct    0     0             D    127.0.0.1
      192.168.123.0/24 Direct    0     0             D    192.168.123.2
      192.168.123.2/32 Direct    0     0             D    127.0.0.1
```

在 R2 的 g0/0/0 上抓取数据包，可以看到 next hop 字段

```
Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  Routing Domain: 0
  + IP Address: 1.1.1.0, Metric: 1
  - IP Address: 3.3.3.3 Metric: 1
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 3.3.3.3 (3.3.3.3)
    Netmask: 255.255.255.255 (255.255.255.255)
    Next Hop: 192.168.123.3 (192.168.123.3)
    Metric: 1
```

总结：

很明显，R2 学习 R3 的环回口，直接从 R3 学习到的最科学，如果从 R1 学习的，需要从 R1 中转一次，不科学。这是因为 RIP v1 中没有 next hop 字段，无法记录路由是怎么来的，而 RIP v2 则有 next hop 字段，可以明确的记录路由的由来。

=====

RIP v2 的 tag 标记

可以在 R1 上对 R3 的路由在引入 RIP 协议是打 tag

```
R1 :
acl 2000
rule permit source 3.3.3.3 0
route-policy 10 permit node 10
if-match acl 2000
apply tag 99

rip
import-route ospf 1 route-policy 10
```

在 R2 上查看 tag, dis rip 1 route

```
[R2]dis rip 1 route
Route Flags : R - RIP
               A - Aging, G - Garbage-collect
-----
Peer 192.168.123.1 on GigabitEthernet0/0/0
  Destination/Mask    Nexthop    Cost    Tag
      3.3.3.3/32      192.168.123.3    1    99
      1.1.1.0/24      192.168.123.1    1     0
[R2]
```

tag 打好之后，可以依据 tag 标识，做相应的过滤处理

=====

[avoid-feedback](#)

向外宣告一条聚合路由，并且禁止从该接口学习到相同的聚合路由，以防止产生路由环路

在 R2 上起多个环回口，模拟多个网段，进行自动聚合，手工聚合

```
int loo0
ip add 2.2.0.2 24
int loo1
ip add 2.2.1.2 24
int loo2
ip add 2.2.2.2 24
```

RIP v2 自动聚合

1.

```
rip
summary alway
```

2.

```
int g0/0/0
undo rip split-horizon
```

RIP v2 手工聚合

```
int g0/0/0
rip summary-address 2.2.0.0 255.255.252.0 avoid-
feedback
```

avoid-feedback：禁止 R2 再从 g0/0/0 学习到 2.2.0.0 的聚合路由

=====

filter-policy 做过滤

R1 不希望收到 R2 的路由

上一步骤做了聚合，deny 的路由为 2.2.0.0 255.255.252.0

R1 :

```
acl 2000
```

```
rule deny source 2.2.0.0 0.0.3.255
```

```
rule permit
```

```
rip
```

```
filter-policy 2000 import
```

在 R1 上查看，就不会有 R2 的路由

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
1.1.1.0/24	Direct	0	0	D	1.1.1.1
1.1.1.1/32	Direct	0	0	D	127.0.0.1
3.3.3.0/24	RIP	100	5	D	192.168.123.3
0/0/0					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
127.0.0.1/32	Direct	0	0	D	127.0.0.1
192.168.123.0/24	Direct	0	0	D	192.168.123.1
0/0/0					
192.168.123.1/32	Direct	0	0	D	127.0.0.1
0/0/0					

也可以这样操作,效果是一样的

R1 :

```
ip ip-prefix 1 deny 192.168.123.2 32 greater-equal  
32 less-equal 32
```

```
rip
```

```
filter-policy ip-prefix 1 import
```


=====

replay-protect

通过使能 Replay-protect (重放保护) 功能，可以得到接口 Down 之前所发送 RIP 报文的 Identification (标符)，避免双方的 RIP 路由信息不同步、丢失。其中 Identification 是 IP 数据报中的标识字段。

假设运行 RIP 的接口状态变为 Down 之前发送的最后的 RIP 报文的 Identification 为 X，该接口状态变为 Up 后，再次发送 RIP 报文的 Identification 会变为 0。

如果对方没有收到这个 Identification 为 0 的 RIP 报文，那么后续的 RIP 报文都将被丢弃，直到收到 Identification 为 X + 1 的 RIP 报文。这样就会导致双方的 RIP 路由信息不同步、丢失。

通过使能 Replay-protect 功能，当接口从 Down 变为 Up 之后，再次发送 RIP 报文的 Identification 会顺次加一，从而避免了上述情况的发生。

R1 R2 之间启用 RIP 认证，格式为 nonstandard，huawei 1 后面的数字 1，代表 key id 为 1

配置时可以选择 MD5 密文验证方式的报文格式

usual 参示使用通用报文格式

nonstandard 参数表示使用非标准报文格式 (IETF 标准)

R1:

```
int g0/0/0
rip authentication-mode md5 nonstandard huawei 1
rip replay-protect 50
```

指定面向连接的收发缓冲区大小，缺省 window-range 为 50 整数形式，取值范围是 50 ~ 50000，单位是个。

```
[R1]display rip 1 int g0/0/0 verbose
GigabitEthernet0/0/0(192.168.123.1)
  State           : UP           MTU       : 500
  Metricin        : 4
  Metricout       : 1
  Input           : Enabled      Output    : Enabled
  Protocol        : RIPv2 Multicast
  Send version    : RIPv2 Multicast Packets
  Receive version : RIPv2 Multicast and Broadcast Packets
  Poison-reverse  : Disabled
  Split-Horizon   : Enabled
  Authentication type : MD5 (Non-standard - Password)
    Last Sequence Number Sent : 0xc
  Replay Protection : Enabled
    Packets Dropped          : 0
    Window Size              : 50
```

=====

RIPv1 的主机路由处理方式

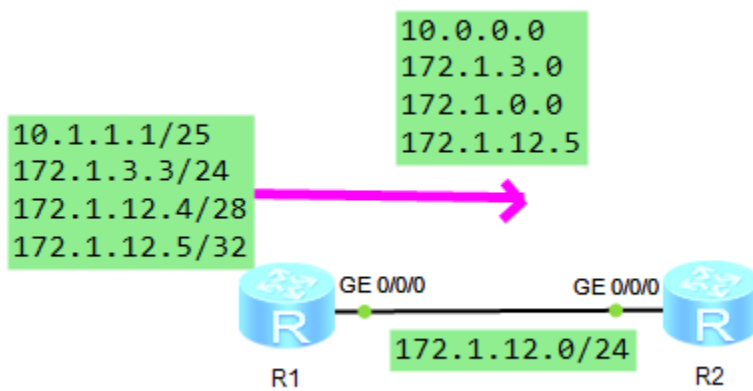
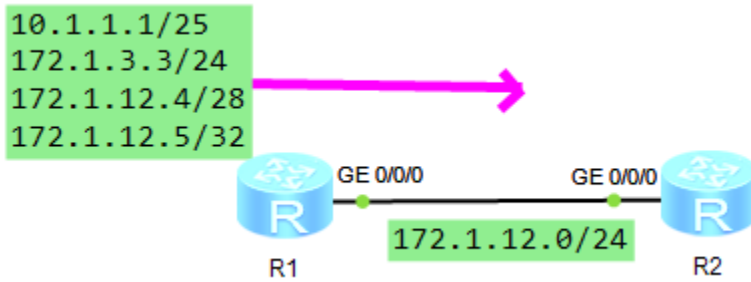
发送规则：

发送的路由与接口的 IP 不在同一主网，自动汇总发送。

发送的路由与接口的 IP 在同一主网，

掩码一致主机位变 0 发送，

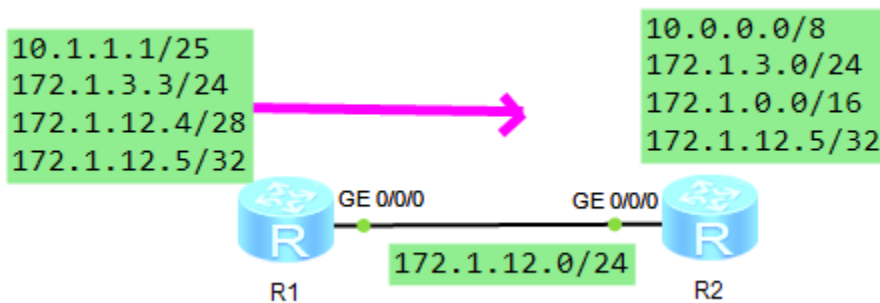
掩码不一致按照主网汇总发送（32 位掩码除外）



接收规则

接收到的路由与接口 IP 不在同一主网，匹配主网掩码。

接收到的路由与接口 IP 在同一主网，依次匹配（直到主机位为 0）：主网掩码——接口掩码——32 位。



```
<R2>display ip routing-table protocol rip
Route Flags: R - relay, D - download to fib
```

```
-----
Public routing table : RIP
```

```
Destinations : 4          Routes : 4
```

```
RIP routing table status : <Active>
```

```
Destinations : 4          Routes : 4
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
10.0.0.0/8	RIP	100	1	D	172.1.12.1
172.1.0.0/16	RIP	100	1	D	172.1.12.1
172.1.3.0/24	RIP	100	1	D	172.1.12.1
172.1.12.5/32	RIP	100	1	D	172.1.12.1

```
RIP routing table status : <Inactive>
```

```
Destinations : 0          Routes : 0
```

如果把两台设备修改为 RIPv2 版本，掩码不会变化

R1: R2

rip

version 2

```
[R2]dis ip routing-table protocol rip
Route Flags: R - relay, D - download to fib
```

```
-----
Public routing table : RIP
```

```
Destinations : 4          Routes : 4
```

```
RIP routing table status : <Active>
```

```
Destinations : 4          Routes : 4
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
10.1.1.0/25	RIP	100	1	D	172.1.12.1
172.1.3.0/24	RIP	100	1	D	172.1.12.1
172.1.12.0/28	RIP	100	1	D	172.1.12.1
172.1.12.5/32	RIP	100	1	D	172.1.12.1