

目 录

1 RIPng	1-1
1.1 RIPng简介	1-1
1.1.1 RIPng的路由度量值.....	1-1
1.1.2 RIPng的路由数据库.....	1-1
1.1.3 RIPng报文及路由发布过程.....	1-1
1.1.4 协议规范.....	1-2
1.2 RIPng配置任务简介	1-2
1.3 配置RIPng的基本功能	1-2
1.4 配置RIPng的路由信息控制	1-3
1.4.1 配置接口附加度量值.....	1-3
1.4.2 配置RIPng路由聚合.....	1-3
1.4.3 配置RIPng发布缺省路由.....	1-4
1.4.4 配置RIPng对接收/发布的路由进行过滤.....	1-4
1.4.5 配置RIPng协议优先级.....	1-5
1.4.6 配置RIPng引入外部路由.....	1-5
1.5 调整和优化RIPng网络	1-6
1.5.1 配置RIPng定时器.....	1-6
1.5.2 配置水平分割和毒性逆转	1-6
1.5.3 配置最大等价路由条数	1-7
1.5.4 配置RIPng报文的发送速率.....	1-7
1.5.5 配置RIPng触发更新的时间间隔.....	1-8
1.6 配置RIPng GR.....	1-8
1.7 配置RIPng NSR	1-9
1.8 配置RIPng快速重路由	1-11
1.8.1 功能简介	1-11
1.8.2 配置限制和指导	1-12
1.8.3 配置RIPng快速重路由功能.....	1-12
1.8.4 配置RIPng快速重路由支持BFD检测功能.....	1-12
1.9 提高RIPng的安全性	1-13
1.9.1 配置RIPng报文的零域检查.....	1-13
1.9.2 配置IPsec保护RIPng报文	1-13
1.10 RIPng显示和维护.....	1-14
1.11 RIPng典型配置举例.....	1-14

1.11.1 RIPng基本功能配置举例	1-14
1.11.2 RIPng引入外部路由配置举例	1-17
1.11.3 RIPng GR配置举例	1-20
1.11.4 RIPng NSR配置举例	1-21
1.11.5 RIPng快速重路由配置举例	1-23
1.11.6 RIPng IPsec安全框架配置举例	1-25

1 RIPng

1.1 RIPng简介

RIPng (RIP next generation, 下一代 RIP 协议) 是基于距离矢量 (Distance-Vector) 算法的协议。它通过 UDP 报文交换路由信息, 使用的端口号为 521。RIPng 是对原来的 IPv4 网络中 RIP-2 协议的扩展, 大多数 RIP 的概念都可以用于 RIPng。

1.1.1 RIPng的路由度量值

RIPng 使用跳数来衡量到达目的地址的距离 (也称为度量值或开销)。在 RIPng 中, 从一个路由器到其直连网络的跳数为 0, 通过与其相连的路由器到达另一个网络的跳数为 1, 其余以此类推。当跳数大于或等于 16 时, 目的网络或主机就被定义为不可达。

1.1.2 RIPng的路由数据库

每个运行 RIPng 的路由器都管理一个路由数据库, 该路由数据库包含了到所有可达目的地的路由项, 这些路由项包含下列信息:

- 目的地址: 主机或网络的 IPv6 地址。
- 下一跳地址: 为到达目的地, 需要经过的相邻路由器的接口 IPv6 地址。
- 出接口: 转发 IPv6 报文通过的出接口。
- 度量值: 本路由器到达目的地的开销。
- 路由时间: 从路由项最后一次被更新到现在所经过的时间, 路由项每次被更新时, 路由时间重置为 0。
- 路由标记 (Route Tag): 用于标识外部路由, 以便在路由策略中根据 Tag 对路由进行灵活的控制。关于路由策略的详细信息, 请参见“三层技术-IP 路由配置指导”中的“路由策略”。

1.1.3 RIPng报文及路由发布过程

RIPng 有两种报文: Request 报文和 Response 报文, 并采用组播方式发送报文, 使用链路本地地址 FF02::9 作为 RIPng 路由更新的目的地址; 使用链路本地地址 FE80::/10 作为 RIPng 路由更新的源地址。

当 RIPng 路由器启动后或者需要更新部分路由表项时, 便会发出 Request 报文, 向邻居请求需要的路由信息。

Response 报文包含本地路由表的信息, 一般在下列情况下产生:

- 对某个 Request 报文进行响应
- 作为更新报文周期性地发出
- 在路由发生变化时触发更新

收到 Request 报文的 RIPng 路由器会以 Response 报文形式发回给请求路由器。

收到 Response 报文的路由器会更新自己的 RIPng 路由表。为了保证路由的准确性，RIPng 路由器会对收到的 Response 报文进行有效性检查，比如源 IPv6 地址是否是链路本地地址，端口号是否正确等，没有通过检查的报文会被忽略。

1.1.4 协议规范

与 RIPng 相关的规范有：

- RFC 2080: RIPng for IPv6
- RFC 2081: RIPng Protocol Applicability Statement

1.2 RIPng配置任务简介

RIPng 配置任务如下：

- (1) [配置RIPng的基本功能](#)
- (2) (可选) [配置RIPng的路由信息控制](#)
 - [配置接口附加度量值](#)
 - [配置RIPng路由聚合](#)
 - [配置RIPng发布缺省路由](#)
 - [配置RIPng对接收/发布的路由进行过滤](#)
 - [配置RIPng协议优先级](#)
 - [配置RIPng引入外部路由](#)
- (3) (可选) [调整和优化RIPng网络](#)
 - [配置RIPng定时器](#)
 - [配置水平分割和毒性逆转](#)
 - [配置最大等价路由条数](#)
 - [配置RIPng报文的发送速率](#)
 - [配置RIPng触发更新的时间间隔](#)
- (4) (可选) 提高 RIPng 的可靠性
 - [配置RIPng GR](#)
 - [配置RIPng NSR](#)
 - [配置RIPng快速重路由](#)
- (5) (可选) [提高RIPng的安全性](#)
 - [配置RIPng报文的零域检查](#)
 - [配置IPsec](#)

1.3 配置RIPng的基本功能

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

缺省情况下，系统没有运行 RIPng。

- (3) 退回系统视图。

```
quit
```

- (4) 进入接口视图。

```
interface interface-type interface-number
```

- (5) 在接口上使能 RIPng 路由协议。

```
ripng process-id enable
```

缺省情况下，接口上的 RIPng 功能处于关闭状态。

如果接口没有使能 RIPng，那么 RIPng 进程在该接口上既不发送也不接收 RIPng 路由。

1.4 配置RIPng的路由信息控制

1.4.1 配置接口附加度量值

1. 功能简介

附加度量值是在 RIPng 路由原来度量值的基础上所增加的度量值（跳数），包括发送附加度量值和接收附加度量值。

- 发送附加度量值：不会改变路由表中的路由度量值，仅当接口发送 RIPng 路由信息时才会添加到发送路由上。
- 接收附加度量值：会影响接收到的路由度量值，接口接收到一条合法的 RIPng 路由时，在将其加入路由表前会把附加度量值加到该路由上。

2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 设置接口接收 RIPng 路由时的附加度量值。

```
ripng metricin value
```

缺省情况下，接口接收 RIPng 路由时的附加度量值为 0。

- (4) 设置接口发送 RIPng 路由时的附加度量值。

```
ripng metricout value
```

缺省情况下，接口发送 RIPng 路由时的附加度量值为 1。

1.4.2 配置RIPng路由聚合

1. 功能简介

RIPng 的路由聚合是在接口上实现的，在接口上配置路由聚合，此时可以将 RIPng 要在这个接口上发布出去的路由按最长匹配原则聚合后发布出去。

RIPng 路由聚合可提高网络的可扩展性和效率，缩减路由表。

RIPng 将多条路由聚合成一条路由时，聚合路由的 Metric 值将取所有路由 Metric 的最小值。

例如，RIPng 从接口发布出去的路由有两条：11:11:11::24 Metric=2 和 11:11:12::34 Metric=3，在此接口上配置的聚合路由为 11::0/16，则最终发布出去的路由为 11::0/16 Metric=2。

2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 配置 RIPng 在接口发布聚合的 IPv6 地址，并指定被聚合的路由的 IPv6 前缀。

```
ripng summary-address ipv6-address prefix-length
```

缺省情况下，未配置 RIPng 在接口发布聚合的 IPv6 地址。

1.4.3 配置RIPng发布缺省路由

1. 功能简介

用户可以配置 RIP 以指定度量值向邻居发布一条缺省路由。

2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 配置 RIPng 发布缺省路由。

```
ripng default-route { only | originate } [ cost cost-value | route-policy  
route-policy-name ] *
```

缺省情况下，RIPng 进程不发布缺省路由。

缺省路由将被强制通过指定接口的路由更新报文发布出去，该路由的发布不考虑其是否已经存在于本设备的 IPv6 路由表中。

1.4.4 配置RIPng对接收/发布的路由进行过滤

1. 功能简介

用户可通过使用 IPv6 ACL 和 IPv6 前缀列表对接收到的路由信息进行过滤，只有通过过滤的路由才能被加入到 RIPng 路由表；此外，还可对本机所有要发布的路由进行过滤，包括从其它路由协议引入的路由和从邻居学到的 RIPng 路由，只有通过过滤的路由才能被发布给 RIPng 邻居。

2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

- (3) 对接收的路由信息进行过滤。

```
filter-policy { ipv6-acl-number | prefix-list prefix-list-name } import
```

缺省情况下，RIPng 不对接收的路由信息进行过滤。

- (4) 对发布的路由信息进行过滤。

```
filter-policy { ipv6-acl-number | prefix-list prefix-list-name } export  
[ protocol [ process-id ] ]
```

缺省情况下，RIPng 不对发布的路由信息进行过滤。

1.4.5 配置RIPng协议优先级

1. 功能简介

任何路由协议都具备特有的协议优先级，在设备进行路由选择时能够在不同的协议中选择最佳路由。可以手工设置 RIPng 协议的优先级，设置的值越小，其优先级越高。

2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

- (3) 配置 RIPng 路由的优先级。

```
preference { preference | route-policy route-policy-name } *
```

缺省情况下，RIPng 路由的优先级为 100。

1.4.6 配置RIPng引入外部路由

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

- (3) 引入外部路由。

- 配置 RIPng 引入 BGP4+ 协议的路由。

```
import-route bgp4+ [ as-number ] [ allow-ibgp ] [ cost cost-value |  
route-policy route-policy-name ] *
```

- 配置 RIPng 引入直连或静态路由。

```
import-route { direct | static } [ cost cost-value | route-policy  
route-policy-name ] *
```

- 配置 RIPng 引入 isisv6、ospfv3 协议或其他 ripng 进程的路由。

```
import-route { isisv6 | ospfv3 | ripng } [ process-id ] [ allow-direct |  
cost cost-value | route-policy route-policy-name ] *
```

缺省情况下，RIPng 不引入其它路由。

- (4) (可选) 配置引入路由的缺省度量值。

default cost *cost-value*

缺省情况下，引入路由的缺省度量值为 0。

1.5 调整和优化RIPng网络

1.5.1 配置RIPng定时器

1. 功能简介

用户可通过调节 RIPng 定时器来调整 RIPng 路由协议的性能，以满足网络需要。

2. 配置限制和指导

在配置 RIPng 定时器时需要注意，定时器值的调整应考虑网络的性能，并在所有运行 RIPng 的路由器上进行统一配置，避免增加不必要的网络流量。

3. 配置步骤

(1) 进入系统视图。

```
system-view
```

(2) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

(3) 配置 RIPng 定时器的值。

```
timers { garbage-collect garbage-collect-value | suppress  
suppress-value | timeout timeout-value | update update-value } *
```

缺省情况下，Update 定时器的值为 30 秒，Timeout 定时器的值为 180 秒，Suppress 定时器的值为 120 秒，Garbage-collect 定时器的值为 120 秒。

1.5.2 配置水平分割和毒性逆转

1. 配置限制和指导

- 如果同时配置了水平分割和毒性逆转，则只有毒性逆转功能生效。
- 配置水平分割可以使得从一个接口学到的路由不能通过此接口向外发布，用于避免相邻路由器间的路由环路。因此，建议不要关闭水平分割。
- 配置毒性逆转可以使得从一个接口学到的路由还可以从这个接口向外发布，但此时这些路由的度量值已设置为 16，即不可达。

2. 配置水平分割

(1) 进入系统视图。

```
system-view
```

(2) 进入接口视图。

```
interface interface-type interface-number
```

(3) 使能水平分割功能。

```
ripng split-horizon
```

缺省情况下，水平分割功能处于使能状态。

3. 配置毒性逆转

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 使能毒性逆转功能。

```
ripng poison-reverse
```

缺省情况下，毒性逆转功能处于关闭状态。

1.5.3 配置最大等价路由条数

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

- (3) 配置 RIPng 最大等价路由条数。

```
maximum load-balancing number
```

1.5.4 配置RIPng报文的发送速率

1. 功能简介

RIPng 周期性地将路由信息放在 RIPng 报文中向邻居发送。

如果路由表里的路由条目数量很多，同时发送大量 RIPng 协议报文有可能会对当前设备和网络带宽带来冲击；因此，路由器将 RIPng 协议报文分为多个批次进行发送，并且对 RIPng 接口每次允许发送的 RIPng 协议报文最大个数做出限制。

用户可根据需要配置接口发送 RIPng 报文的时间间隔以及接口一次发送 RIPng 报文的最大个数。

2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 配置 RIPng 报文的发送速率。

- 请依次执行以下命令在 RIP 视图下配置 RIPng 报文发送速率。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

```
output-delay time count count
```

缺省情况下，接口发送 RIPng 报文的时间间隔为 20 毫秒，一次最多发送 3 个 RIPng 报文。

- 请依次执行以下命令在接口视图下配置 RIPng 报文发送速率。

```
interface interface-type interface-number
```

```
ripng output-delay time count count
```

缺省情况下，接口发送 RIPng 报文的速率以 RIPng 进程配置的为准。

1.5.5 配置RIPng触发更新的时间间隔

1. 功能简介

RIPng 路由信息变化后将以触发更新的方式通知邻居设备，加速邻居设备的路由收敛。如果路由信息频繁变化，且每次变化都立即发送触发更新，将会占用大量系统资源，并影响路由器的效率。通过调节触发更新的时间间隔，可以抑制由于路由信息频繁变化带来的影响。本命令在路由信息变化不频繁的情况下将连续触发更新的时间间隔缩小到 *minimum-interval*，而在路由信息变化频繁的情况下可以进行相应惩罚，增加 $incremental-interval \times 2^{n-2}$ （n 为连续触发更新的次数），将等待时间按照配置的惩罚增量延长，最大不超过 *maximum-interval*。

2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

- (3) 配置 RIPng 触发更新的时间间隔。

```
timer triggered maximum-interval [ minimum-interval  
[ incremental-interval ] ]
```

缺省情况下，发送触发更新的最大时间间隔为 5 秒，最小间隔为 50 毫秒，增量惩罚间隔为 200 毫秒。

1.6 配置RIPng GR

1. 功能简介

GR（Graceful Restart，平滑重启）是一种在协议重启或主备倒换时 RIPng 进行平滑重启，保证转发业务不中断的机制。

GR 有两个角色：

- GR Restarter：发生协议重启或主备倒换事件且具有 GR 能力的设备。
- GR Helper：和 GR Restarter 具有邻居关系，协助完成 GR 流程的设备。

在普通的路由协议重启的情况下，路由器需要重新学习 RIPng 路由，并更新 FIB 表，此时会引起网络暂时的中断，基于 RIPng 的 GR 可以解决这个问题。

应用了 GR 特性的设备向外发送 RIPng 全部路由表请求报文，重新从邻居处学习 RIPng 路由，在此期间 FIB 表不变化。在路由协议重启完毕后，设备将重新学到的 RIPng 路由下刷给 FIB 表，使该设备的路由信息恢复到重启前的状态。

本配置在 GR Restarter 上进行。启动了 RIPng 的设备缺省就是 GR Helper。

2. 配置限制和指导

设备充当 GR Restarter 后不能再配置 RIPng NSR 功能。

3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

(2) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

(3) 使能 RIPng 协议的 GR 能力。

```
graceful-restart
```

缺省情况下，RIPng 协议的 GR 能力处于关闭状态。

(4) （可选）配置 RIPng 协议的 GR 重启间隔时间。

```
graceful-restart interval interval
```

缺省情况下，RIPng 协议的 GR 重启间隔时间为 60 秒。

1.7 配置RIPng NSR

1. 功能简介

NSR（Nonstop Routing，不间断路由）通过将 RIPng 路由信息从主进程备份到备进程，使设备在发生主备倒换时新主进程可以无缝完成路由的重新生成、下刷，邻接关系不会发生中断，从而避免了主备倒换对转发业务的影响。

GR 特性需要周边设备配合才能完成路由信息的恢复，在网络应用中有一定的限制。NSR 特性不需要周边设备的配合，网络应用更加广泛。

2. RIPng NSR与硬件适配关系

本命令的支持情况与设备型号有关，请以设备的实际情况为准。

型号	说明
MSR810、MSR810-W、MSR810-W-DB、MSR810-LM、MSR810-W-LM、MSR810-10-PoE、MSR810-LM-HK、MSR810-W-LM-HK、MSR810-LM-CNDE-SJK	不支持
MSR810-LMS、MSR810-LUS	不支持
MSR810-LMS-EA、MSR810-LME	不支持
MSR2600-6-X1、MSR2600-10-X1	不支持
MSR 2630	<ul style="list-style-type: none">当为独立运行模式设备时，不支持当为 IRF 模式时，支持
MSR3600-28、MSR3600-51	<ul style="list-style-type: none">当为独立运行模式设备时，不支持当为 IRF 模式时，支持
MSR3600-28-SI、MSR3600-51-SI	<ul style="list-style-type: none">当为独立运行模式设备时，不支持当为 IRF 模式时，支持
MSR3600-28-X1、MSR3600-28-X1-DP、MSR3600-51-X1、MSR3600-51-X1-DP	<ul style="list-style-type: none">当为独立运行模式设备时，不支持当为 IRF 模式时，支持
MSR3610-I-DP、MSR3610-IE-DP、MSR3610-IE-ES	不支持

MSR2600-6-X1、MSR2600-10-X1	不支持
MSR3610-X1、MSR3610-X1-DP、MSR3610-X1-DC、MSR3610-X1-DP-DC	<ul style="list-style-type: none"> • 当为独立运行模式设备时，不支持 • 当为 IRF 模式时，支持
MSR 3610、MSR 3620、MSR 3620-DP、MSR 3640、MSR 3660	<ul style="list-style-type: none"> • 当为独立运行模式设备时，不支持 • 当为 IRF 模式时，支持
MSR3610-G、MSR3620-G	<ul style="list-style-type: none"> • 当为独立运行模式设备时，不支持 • 当为 IRF 模式时，支持

型号	描述
MSR810-W-WiNet、MSR810-LM-WiNet	不支持
MSR830-4LM-WiNet	不支持
MSR830-5BEI-WiNet、MSR830-6EI-WiNet、MSR830-10BEI-WiNet	不支持
MSR830-6BHI-WiNet、MSR830-10BHI-WiNet	不支持
MSR2600-6-WiNet 、MSR2600-10-X1-WiNet	不支持
MSR2630-WiNet	<ul style="list-style-type: none"> • 当为独立运行模式设备时，不支持 • 当为 IRF 模式时，支持
MSR3600-28-WiNet	<ul style="list-style-type: none"> • 当为独立运行模式设备时，不支持 • 当为 IRF 模式时，支持
MSR3610-X1-WiNet	<ul style="list-style-type: none"> • 当为独立运行模式设备时，不支持 • 当为 IRF 模式时，支持
MSR3610-WiNet、MSR3620-10-WiNet、MSR3620-DP-WiNet、MSR3620-WiNet、MSR3660-WiNet	<ul style="list-style-type: none"> • 当为独立运行模式设备时，不支持 • 当为 IRF 模式时，支持

型号	说明
MSR2630-XS	不支持
MSR3600-28-XS	<ul style="list-style-type: none"> • 当为独立运行模式设备时，不支持 • 当为 IRF 模式时，支持
MSR3610-XS	<ul style="list-style-type: none"> • 当为独立运行模式设备时，不支持 • 当为 IRF 模式时，支持

型号	说明
MSR3620-XS	<ul style="list-style-type: none"> 当为独立运行模式设备时，不支持 当为 IRF 模式时，支持
MSR3610-I-XS	不支持
MSR3610-IE-XS	不支持

3. 配置限制和指导

各个进程的 NSR 功能是相互独立的，只对本进程生效。如果存在多个 RIPng 进程，建议在各个进程下使能 RIPng NSR 功能。

设备配置了 RIPng NSR 功能后不能在充当 GR Restarter。

4. 配置步骤

(1) 进入系统视图。

```
system-view
```

(2) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

(3) 使能 RIPng NSR 功能。

```
non-stop-routing
```

缺省情况下，RIPng NSR 功能处于关闭状态。

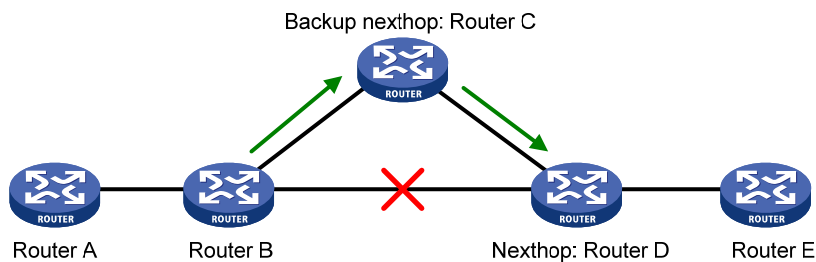
1.8 配置RIPng快速重路由

1.8.1 功能简介

在部署了备份链路的 RIPng 网络中，当主用链路发生故障时，RIPng 会对路由进行重新计算，在路由收敛完成后，流量可以通过备份链路进行传输。在路由收敛期间，数据流量将会被中断。

为了尽可能缩短网络故障导致的流量中断时间，网络管理员可以根据需要配置 RIPng 快速重路由功能。

图1-1 RIPng 快速重路由功能示意图



如 [图 1-1](#) 所示，通过在 Router B 上配置快速重路由功能，RIPng 可以为路由指定备份下一跳，当 Router B 检测到主用下一跳地址无法到达时，会直接使用备份下一跳地址来指导报文的转发，从而

大大缩短了流量路径切换的时间。在快速切换流量传输路径的同时，RIPng会根据变化后的网络拓扑重新计算路由，在路由收敛完毕后，使用新计算出来的最优路由来指导报文转发。

1.8.2 配置限制和指导

- 本功能只适合在主链路三层接口 **up**，主链路由双通变为单通或者不通的情况下使用。在主链路三层接口 **down** 的情况下，本功能不可用。单通现象，即一条链路上的两端，有且只有一端可以收到另一端发来的报文，此链路称为单向链路。
- RIPng 快速重路由功能仅对非迭代 RIPng 路由（即从直连邻居学到 RIPng 路由）有效。
- 等价路由不支持快速重路由功能。

1.8.3 配置RIPng快速重路由功能

- (1) 进入系统视图。

```
system-view
```

- (2) 配置路由策略。

在路由策略中通过 **apply ipv6 fast-reroute backup-interface** 命令在路由策略中指定备份下一跳。详细配置请参见“三层技术-IP 路由配置指导”中的“路由策略”。

- (3) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

- (4) 开启 RIPng 快速重路由功能。

```
fast-reroute route-policy route-policy-name
```

缺省情况下，RIPng 快速重路由功能处于关闭状态。

1.8.4 配置RIPng快速重路由支持BFD检测功能

1. 功能简介

RIPng 协议的快速重路由特性中，主用链路缺省不使用 BFD 进行链路故障检测。配置本功能后，将使用 BFD（Echo 方式）进行检测，可以加快 RIPng 协议的收敛速度。

2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 配置 BFD Echo 报文源地址。

```
bfd echo-source-ipv6 ipv6-address
```

缺省情况下，未配置 BFD Echo 报文源地址。

echo 报文的源 IPv6 地址用户可以任意指定。建议配置 echo 报文的源 IPv6 地址不属于该设备任何一个接口所在网段。

本命令的详细介绍请参见“可靠性命令参考”中的“BFD”。

- (3) 进入接口视图。

```
interface interface-type interface-number
```

- (4) 使能 RIPng 协议中主用链路的 BFD（Echo 方式）检测功能。

```
ripng primary-path-detect bfd echo
```

缺省情况下，RIPng 协议中主用链路的 BFD（Echo 方式）检测功能处于关闭状态。

1.9 提高RIPng的安全性

1.9.1 配置RIPng报文的零域检查

1. 功能简介

RIPng 报文头部中的一些字段必须配置为 0，也称为零域。使能 RIPng 报文的零域检查功能后，如果报文头部零域中的值不为零，这些报文将被丢弃，不做处理。如果能确保所有报文都是可信任的，则不需要进行该项检查，以节省 CPU 处理时间。

2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

- (3) 使能对 RIPng 报文头部的零域检查功能。

```
checkzero
```

缺省情况下，RIPng 报文的零域检查功能处于使能状态。

1.9.2 配置IPsec保护RIPng报文

1. 功能简介

在安全性要求较高的网络环境中，可以通过配置基于 IPsec 安全框架的认证方式来对 RIPng 报文进行有效性检查和验证。IPsec 安全框架的具体情况请参见“安全配置指导”中的“IPsec”。

设备在发送的报文中会携带配置好的 IPsec 安全框架的 SPI（Security Parameter Index，安全参数索引）值，接收报文时通过 SPI 值进行 IPsec 安全框架匹配：只有安全框架匹配的报文才能接收；否则将不会接收报文，从而不能正常建立邻居和学习路由。

2. 配置限制和指导

RIPng 支持在进程和接口下配置 IPsec 安全框架。进程下配置的 IPsec 安全框架对该进程下的所有报文有效，接口下的 IPsec 安全框架只对接口的报文有效。当接口和接口所在进程均配置了 IPsec 安全框架时，接口下的配置生效。

3. RIPng进程上应用IPsec安全框架

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 RIPng 视图。

```
ripng [ process-id ] [ vpn-instance vpn-instance-name ]
```

- (3) 配置 RIPng 进程应用 IPsec 安全框架。

```
enable ipsec-profile profile-name
```

缺省情况下，RIPng 进程没有应用 IPsec 安全框架。

4. 接口上应用IPsec安全框架

(1) 进入系统视图。

```
system-view
```

(2) 进入接口视图。

```
interface interface-type interface-number
```

(3) 配置使能了 RIPng 的接口上应用 IPsec 安全框架。

```
ripng ipsec-profile profile-name
```

缺省情况下, RIPng 接口没有应用 IPsec 安全框架。

1.10 RIPng显示和维护

在完成上述配置后, 在任意视图下执行 **display** 命令可以显示配置后 RIPng 的运行情况, 通过查看显示信息验证配置的效果。

在用户视图下执行 **reset** 命令可以重启 RIPng 进程或清除指定 RIPng 进程的统计信息。

表1-1 RIPng 显示和维护

操作	命令
显示RIPng进程的GR状态信息	display ripng [<i>process-id</i>] graceful-restart
显示RIPng进程的NSR状态信息	display ripng [<i>process-id</i>] non-stop-routing
显示RIPng进程的配置信息	display ripng [<i>process-id</i>]
显示RIPng发布数据库中的路由	display ripng process-id database [<i>ipv6-address</i> <i>prefix-length</i>]
显示指定RIPng进程的接口信息	display ripng process-id interface [<i>interface-type</i> <i>interface-number</i>]
显示RIPng进程的邻居信息	display ripng process-id neighbor [<i>interface-type</i> <i>interface-number</i>]
显示指定RIPng进程的路由信息	display ripng process-id route [<i>ipv6-address</i> <i>prefix-length</i> [verbose] peer <i>ipv6-address</i> statistics]
重启指定RIPng进程	reset ripng process-id process
清除RIPng进程的统计信息	reset ripng process-id statistics

1.11 RIPng典型配置举例

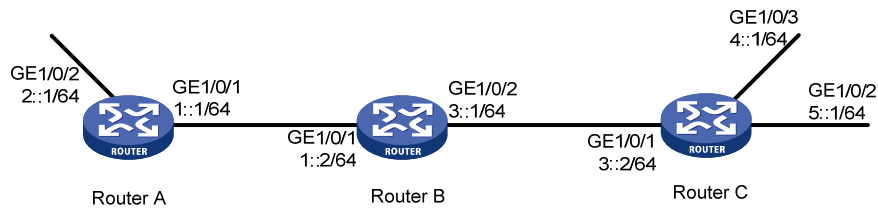
1.11.1 RIPng基本功能配置举例

1. 组网需求

- Router A、Router B 和 Router C 相连并通过 RIPng 来学习网络中的 IPv6 路由信息。
- 在 Router B 上对接收的 Router A 的路由(2::/64)进行过滤, 使其不加入到 Router B 的 RIPng 进程的路由表中, 发布给 Router A 的路由只有 (4::/64)。

2. 组网图

图1-2 RIPng 基本功能配置组网图



3. 配置步骤

- (1) 配置各接口的 IPv6 地址（略）
- (2) 配置 RIPng 的基本功能

配置 Router A。

```
<RouterA> system-view
[RouterA] ripng 1
[RouterA-ripng-1] quit
[RouterA] interface gigabitethernet 1/0/1
[RouterA-GigabitEthernet1/0/1] ripng 1 enable
[RouterA-GigabitEthernet1/0/1] quit
[RouterA] interface gigabitethernet 1/0/2
[RouterA-GigabitEthernet1/0/2] ripng 1 enable
[RouterA-GigabitEthernet1/0/2] quit
```

配置 Router B。

```
<RouterB> system-view
[RouterB] ripng 1
[RouterB-ripng-1] quit
[RouterB] interface gigabitethernet 1/0/1
[RouterB-GigabitEthernet1/0/1] ripng 1 enable
[RouterB-GigabitEthernet1/0/1] quit
[RouterB] interface gigabitethernet 1/0/2
[RouterB-GigabitEthernet1/0/2] ripng 1 enable
[RouterB-GigabitEthernet1/0/2] quit
```

配置 Router C。

```
<RouterC> system-view
[RouterC] ripng 1
[RouterC-ripng-1] quit
[RouterC] interface gigabitethernet 1/0/1
[RouterC-GigabitEthernet1/0/1] ripng 1 enable
[RouterC-GigabitEthernet1/0/1] quit
[RouterC] interface gigabitethernet 1/0/2
[RouterC-GigabitEthernet1/0/2] ripng 1 enable
[RouterC-GigabitEthernet1/0/2] quit
[RouterC] interface gigabitethernet 1/0/3
[RouterC-GigabitEthernet1/0/3] ripng 1 enable
```

```
[RouterC-GigabitEthernet1/0/3] quit
# 查看 Router B 的 RIPng 路由表。
[RouterB] display ripng 1 route
    Route Flags: A - Aging, S - Suppressed, G - Garbage-collect, D - Direct
                  O - Optimal, F - Flush to RIB
-----

Peer FE80::20F:E2FF:FE23:82F5 on GigabitEthernet1/0/1
Destination 2::/64,
    via FE80::20F:E2FF:FE23:82F5, cost 1, tag 0, AOF, 6 secs
Peer FE80::20F:E2FF:FE00:100 on GigabitEthernet1/0/2
Destination 4::/64,
    via FE80::20F:E2FF:FE00:100, cost 1, tag 0, AOF, 11 secs
Destination 5::/64,
    via FE80::20F:E2FF:FE00:100, cost 1, tag 0, AOF, 11 secs
Local route
Destination 1::/64,
    via ::, cost 0, tag 0, DOF
Destination 3::/64,
    via ::, cost 0, tag 0, DOF
```

查看 Router A 的 RIPng 路由表。

```
[RouterA] display ripng 1 route
    Route Flags: A - Aging, S - Suppressed, G - Garbage-collect, D - Direct
                  O - Optimal, F - Flush to RIB
-----

Peer FE80::200:2FF:FE64:8904 on GigabitEthernet1/0/1
Destination 3::/64,
    via FE80::200:2FF:FE64:8904, cost 1, tag 0, AOF, 31 secs
Destination 4::/64,
    via FE80::200:2FF:FE64:8904, cost 2, tag 0, AOF, 31 secs
Destination 5::/64,
    via FE80::200:2FF:FE64:8904, cost 2, tag 0, AOF, 31 secs
Local route
Destination 2::/64,
    via ::, cost 0, tag 0, DOF
Destination 1::/64,
    via ::, cost 0, tag 0, DOF
```

(3) 配置 Router B 对接收和发布的路由进行过滤

```
[RouterB] ipv6 prefix-list aaa permit 4:: 64
[RouterB] ipv6 prefix-list bbb deny 2:: 64
[RouterB] ipv6 prefix-list bbb permit :: 0 less-equal 128
[RouterB] ripng 1
[RouterB-ripng-1] filter-policy prefix-list aaa export
[RouterB-ripng-1] filter-policy prefix-list bbb import
[RouterB-ripng-1] quit
```

查看 Router B 和 Router A 的 RIPng 路由表。

```

[RouterB] display ripng 1 route
    Route Flags: A - Aging, S - Suppressed, G - Garbage-collect, D - Direct
                O - Optimal, F - Flush to RIB
-----

Peer FE80::1:1 on GigabitEthernet1/0/1

Peer FE80::3:1 on GigabitEthernet1/0/2
Destination 4::/64,
    via FE80::2:2, cost 1, tag 0, AOF, 11 secs
Destination 5::/64,
    via FE80::2:2, cost 1, tag 0, AOF, 11 secs
Local route
Destination 1::/64,
    via ::, cost 0, tag 0, DOF
Destination 3::/64,
    via ::, cost 0, tag 0, DOF
[RouterA] display ripng 1 route
    Route Flags: A - Aging, S - Suppressed, G - Garbage-collect, D - Direct
                O - Optimal, F - Flush to RIB
-----

Peer FE80::2:1 on GigabitEthernet1/0/1
Destination 4::/64,
    via FE80::1:1, cost 2, tag 0, AOF, 2 secs

```

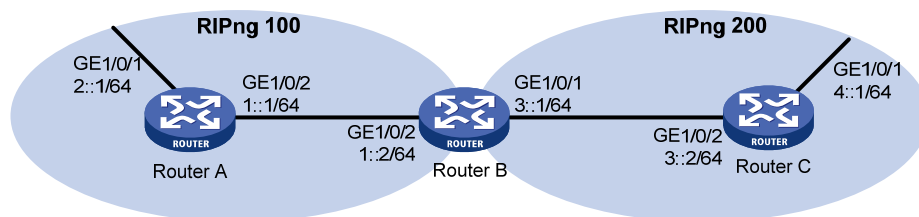
1.11.2 RIPng引入外部路由配置举例

1. 组网需求

- Router B 上运行两个 RIPng 进程：RIPng100 和 RIPng200。Router B 通过 RIPng100 和 Router A 交换路由信息，通过 RIPng200 和 Router C 交换路由信息。
- 要求在 Router B 上配置路由引入，将两个不同进程的 RIPng 路由相互引入到对方的 RIPng 进程中。

2. 组网图

图1-3 RIPng 引入外部路由配置组网图



3. 配置步骤

- (1) 配置各接口的 IPv6 地址（略）

(2) 配置 RIPng

在 Router A 上启动 RIPng 进程 100。

```
<RouterA> system-view
[RouterA] ripng 100
[RouterA-ripng-100] quit
[RouterA] interface gigabitethernet 1/0/1
[RouterA-GigabitEthernet1/0/1] ripng 100 enable
[RouterA-GigabitEthernet1/0/1] quit
[RouterA] interface gigabitethernet 1/0/2
[RouterA-GigabitEthernet1/0/2] ripng 100 enable
```

在 Router B 上启动两个 RIPng 进程，进程号分别为 100 和 200。

```
<RouterB> system-view
[RouterB] ripng 100
[RouterB-ripng-100] quit
[RouterB] interface gigabitethernet 1/0/2
[RouterB-GigabitEthernet1/0/2] ripng 100 enable
[RouterB-GigabitEthernet1/0/2] quit
[RouterB] ripng 200
[RouterB-ripng-200] quit
[RouterB] interface gigabitethernet 1/0/1
[RouterB-GigabitEthernet1/0/1] ripng 200 enable
```

在 Router C 上启动 RIPng 进程 200。

```
<RouterC> system-view
[RouterC] ripng 200
[RouterC] interface gigabitethernet 1/0/1
[RouterC-GigabitEthernet1/0/1] ripng 200 enable
[RouterC-GigabitEthernet1/0/1] quit
[RouterC] interface gigabitethernet 1/0/2
[RouterC-GigabitEthernet1/0/2] ripng 200 enable
[RouterC-GigabitEthernet1/0/2] quit
```

查看 Router A 的路由表信息。

```
[RouterA] display ipv6 routing-table
```

```
Destinations : 7 Routes : 7
```

```
Destination: ::1/128                Protocol : Direct
NextHop      : ::1                  Preference: 0
Interface    : InLoop0              Cost      : 0
```

```
Destination: 1::/64                 Protocol : Direct
NextHop      : ::                  Preference: 0
Interface    : GE1/0/2              Cost      : 0
```

```
Destination: 1::1/128               Protocol : Direct
NextHop      : ::1                  Preference: 0
Interface    : InLoop0              Cost      : 0
```

```
Destination: 2::/64                Protocol : Direct
NextHop      : ::                  Preference: 0
Interface    : GE1/0/1             Cost      : 0
```

```
Destination: 2::1/128              Protocol : Direct
NextHop      : ::1                 Preference: 0
Interface    : InLoop0             Cost      : 0
```

```
Destination: FE80::/10             Protocol : Direct
NextHop      : ::                  Preference: 0
Interface    : NULL0               Cost      : 0
```

```
Destination: FF00::/8              Protocol : Direct
NextHop      : ::                  Preference: 0
Interface    : NULL0               Cost      : 0
```

(3) 配置 RIPng 引入外部路由

在 Router B 上将两个不同 RIPng 进程的路由相互引入到对方的路由表中。

```
[RouterB] ripng 100
[RouterB-ripng-100] import-route ripng 200
[RouterB-ripng-100] quit
[RouterB] ripng 200
[RouterB-ripng-200] import-route ripng 100
[RouterB-ripng-200] quit
```

查看路由引入后 Router A 的路由表信息。

```
[RouterA] display ipv6 routing-table
```

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

```
Destination: ::1/128              Protocol : Direct
NextHop      : ::1                 Preference: 0
Interface    : InLoop0             Cost      : 0
```

```
Destination: 1::/64               Protocol : Direct
NextHop      : ::                  Preference: 0
Interface    : GE1/0/2             Cost      : 0
```

```
Destination: 1::1/128             Protocol : Direct
NextHop      : ::1                 Preference: 0
Interface    : InLoop0             Cost      : 0
```

```
Destination: 2::/64                Protocol : Direct
NextHop      : ::                  Preference: 0
Interface    : GE1/0/1             Cost      : 0
```

```
Destination: 2::1/128              Protocol : Direct
NextHop      : ::1                 Preference: 0
Interface    : InLoop0             Cost      : 0
```

Destination: 4::/64	Protocol : RIPng
NextHop : FE80::200:BFF:FE01:1C02	Preference: 100
Interface : GE1/0/2	Cost : 1

Destination: FE80::/10	Protocol : Direct
NextHop : ::	Preference: 0
Interface : NULL0	Cost : 0

Destination: FF00::/8	Protocol : Direct
NextHop : ::	Preference: 0
Interface : NULL0	Cost : 0

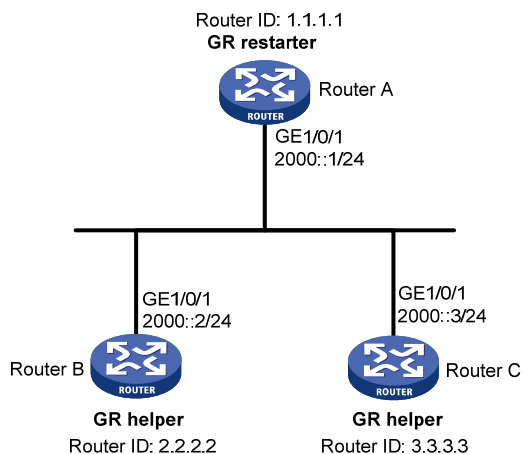
1.11.3 RIPng GR配置举例

1. 组网需求

- Router A、Router B 和 Router C 通过 RIPng 协议实现网络互连。
- Router A 作为 GR Restarter，Router B 和 Router C 作为 GR Helper 并且通过 GR 机制与 Router A 保持同步。

2. 组网图

图1-4 RIPng GR 配置组网图



3. 配置步骤

(1) 配置各路由器接口的 IPv6 地址和 RIPng 协议

请按照上面组网图配置各接口的 IPv6 地址，具体配置过程略。

配置各路由器之间采用 RIPng 协议进行互连，确保 Router A、Router B 和 Router C 之间能够在网络层互通，并且各路由器之间能够借助 RIPng 协议实现动态路由更新。

(2) 配置 RIPng GR

使能 Router A 的 RIPng GR 功能。

```

<RouterA> system-view
[RouterA] ripng 1
[RouterA-ripng-1] graceful-restart
  
```

4. 验证配置

在 Router A 上触发协议重启或主备倒换后，查看 RIPng 的 GR 状态。

```
<RouterA> display ripng 1 graceful-restart
RIPng process: 1
  Graceful Restart capability      : Enabled
  Current GR state                 : Normal
  Graceful Restart period         : 60 seconds
  Graceful Restart remaining time : 0 seconds
```

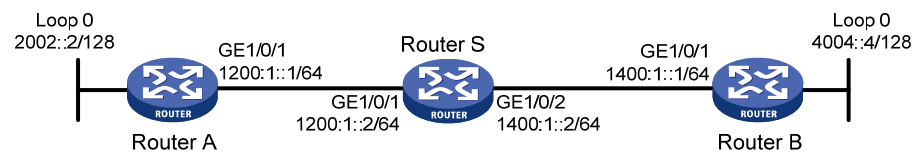
1.11.4 RIPng NSR配置举例

1. 组网需求

Router S、Router A、Router B 通过 RIPng 协议实现网络互连。要求对 Router S 进行主备倒换时，Router A 和 Router B 到 Router S 的邻居没有中断，Router A 到 Router B 的流量没有中断。

2. 组网图

图1-5 RIPng NSR 配置组网图



3. 配置步骤

(1) 配置各路由器接口的 IPv6 地址和 RIPng 协议

请按照上面组网图配置各接口的 IPv6 地址，具体配置过程略。

配置各路由器之间采用 RIPng 协议进行互连，确保 Router S、Router A 和 Router D 之间能够在网络层互通，并且各路由器之间能够借助 RIPng 协议实现动态路由更新。

(2) 配置 RIPng NSR

使能 Router S 的 RIPng NSR 功能。

```
<RouterS> system-view
[RouterS] ripng 1
[RouterS-ripng-1] non-stop-routing
[RouterS-ripng-1] quit
```

4. 验证配置

Router S 进行主备倒换。

```
[RouterS] placement reoptimize
Predicted changes to the placement
Program          Current location  New location
-----
lb               0/0              0/0
lsm              0/0              0/0
slsp            0/0              0/0
rib6             0/0              0/0
```

```

routepolicy          0/0          0/0
rib                  0/0          0/0
staticroute6        0/0          0/0
staticroute          0/0          0/0
eviisis              0/0          0/0
ospf                  0/0          1/0

```

Continue? [y/n]:y

Re-optimization of the placement start. You will be notified on completion

Re-optimization of the placement complete. Use 'display placement' to view the new placement

查看 Router A 上 RIPng 协议的邻居和路由。

```
[RouterA] display ripng 1 neighbor
```

```

Neighbor Address: FE80::AE45:5CE7:422E:2867
  Interface   : GigabitEthernet1/0/1
  Version     : RIPng version 1      Last update: 00h00m23s
  Bad packets: 0                    Bad routes : 0

```

```
[RouterA] display ripng 1 route
```

```

Route Flags: A - Aging, S - Suppressed, G - Garbage-collect, D - Direct
              O - Optimal, F - Flush to RIB

```

```

-----
Peer FE80::AE45:5CE7:422E:2867 on GigabitEthernet1/0/1
Destination 1400:1::/64,
  via FE80::AE45:5CE7:422E:2867, cost 1, tag 0, AOF, 1 secs
Destination 4004::4/128,
  via FE80::AE45:5CE7:422E:2867, cost 2, tag 0, AOF, 1 secs
Local route
Destination 2002::2/128,
  via ::, cost 0, tag 0, DOF
Destination 1200:1::/64,
  via ::, cost 0, tag 0, DOF

```

查看 Router B 上 RIPng 协议的邻居和路由。

```
[RouterB] display ripng 1 neighbor
```

```

Neighbor Address: FE80::20C:29FF:FECE:6277
  Interface   : GigabitEthernet1/0/1
  Version     : RIPng version 1      Last update: 00h00m18s
  Bad packets: 0                    Bad routes : 0

```

```
[RouterB] display ripng 1 route
```

```

Route Flags: A - Aging, S - Suppressed, G - Garbage-collect, D - Direct
              O - Optimal, F - Flush to RIB

```

```

-----
Peer FE80::20C:29FF:FECE:6277 on GigabitEthernet1/0/1
Destination 2002::2/128,
  via FE80::20C:29FF:FECE:6277, cost 2, tag 0, AOF, 24 secs
Destination 1200:1::/64,
  via FE80::20C:29FF:FECE:6277, cost 1, tag 0, AOF, 24 secs
Local route
Destination 4004::4/128,
  via ::, cost 0, tag 0, DOF
Destination 1400:1::/64,

```


via ::, cost 0, tag 0, DOF

通过上面信息可以看出在 Router S 发生主备倒换的时候，Router A 和 Router B 的邻居和路由信息保持不变，从 Router A 到 Router B 的流量转发没有受到主备倒换的影响。

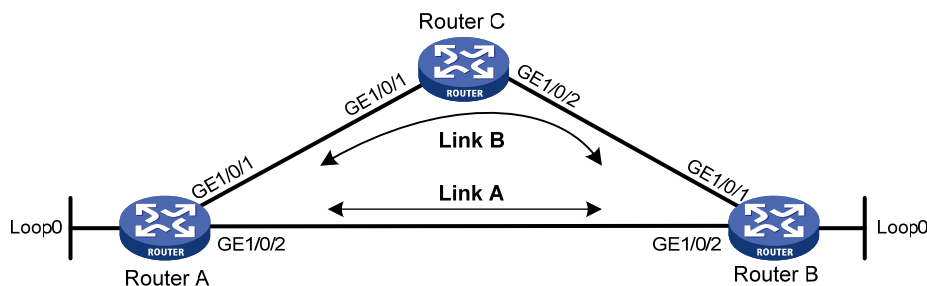
1.11.5 RIPng快速重路由配置举例

1. 组网需求

Router A、Router B 和 Router C 通过 RIPng 协议实现网络互连。要求当 Router A 和 Router B 之间的链路出现单通故障时，业务可以快速切换到链路 B 上。

2. 组网图

图1-6 RIPng 快速重路由配置组网图



设备	接口	IP地址	设备	接口	IP地址
Router A	GE1/0/1	1::1/64	Router B	GE1/0/1	3::1/64
	GE1/0/2	2::1/64		GE1/0/2	2::2/64
	Loop0	10::1/128		Loop0	20::1/128
Router C	GE1/0/1	1::2/64			
	GE1/0/2	3::2/64			

3. 配置步骤

(1) 配置各路由器接口的 IPv6 地址和 RIPng 协议

请按照上面组网图配置各接口的 IPv6 地址，具体配置过程略。

配置各路由器之间采用 RIPng 协议进行互连，确保 Router A、Router B 和 Router C 之间能够在网络层互通，并且各路由器之间能够借助 RIPng 协议实现动态路由更新。

具体配置过程略。

(2) 配置 RIPng 快速重路由

配置 Router A。

```
<RouterA> system-view
[RouterA] ipv6 prefix-list abc index 10 permit 20::1 128
[RouterA] route-policy frr permit node 10
[RouterA-route-policy-frr-10] if-match ipv6 address prefix-list abc
[RouterA-route-policy-frr-10] apply ipv6 fast-reroute backup-interface
gigabitethernet 1/0/1 backup-nextthop 1::2
[RouterA-route-policy-frr-10] quit
[RouterA] ripng 1
```

```

[RouterA-ripng-1] fast-reroute route-policy frr
[RouterA-ripng-1] quit
# 配置 Router B。
<RouterB> system-view
[RouterB] ipv6 prefix-list abc index 10 permit 10::1 128
[RouterB] route-policy frr permit node 10
[RouterB-route-policy-frr-10] if-match ipv6 address prefix-list abc
[RouterB-route-policy-frr-10] apply ipv6 fast-reroute backup-interface
gigabitethernet 1/0/1 backup-nexthop 3::2
[RouterB-route-policy-frr-10] quit
[RouterB] ripng 1
[RouterB-ripng-1] fast-reroute route-policy frr
[RouterB-ripng-1] quit

```

4. 验证配置

在 Router A 上查看 20::1/128 的路由信息，可以看到备份下一跳信息。

```
[RouterA] display ipv6 routing-table 20::1 128 verbose
```

```
Summary count : 1
```

```
Destination: 20::1/128
```

```
Protocol: RIPng
```

```
Process ID: 1
```

```
SubProtID: 0x0 Age: 00h17m42s
```

```
Cost: 1 Preference: 100
```

```
IpPre: N/A QoSLocalID: N/A
```

```
Tag: 0 State: Inactive Adv
```

```
OrigTblID: 0x0 OrigVrf: default-vrf
```

```
TableID: 0xa OrigAs: 0
```

```
NibID: 0x22000003 LastAs: 0
```

```
AttrID: 0xffffffff Neighbor: FE80::34CD:9FF:FE2F:D02
```

```
Flags: 0x41 OrigNextHop: FE80::34CD:9FF:FE2F:D02
```

```
Label: NULL RealNextHop: FE80::34CD:9FF:FE2F:D02
```

```
BkLabel: NULL BkNextHop: FE80::7685:45FF:FEAD:102
```

```
SRLLabel: NULL BkSRLLabel: NULL
```

```
Tunnel ID: Invalid Interface: GigabitEthernet1/0/2
```

```
BkTunnel ID: Invalid BkInterface: GigabitEthernet1/0/1
```

```
FtnIndex: 0x0 TrafficIndex: N/A
```

```
Connector: N/A VpnPeerId: N/A
```

```
Dscp: N/A
```

在 Router B 上查看 10::1/128 的路由信息，可以看到备份下一跳信息。

```
[RouterB] display ipv6 routing-table 10::1 128 verbose
```

```
Summary count : 1
```

```
Destination: 10::/128
```

```
Protocol: RIPng
```

```
Process ID: 1
```

```

SubProtID: 0x0                               Age: 00h22m34s
  Cost: 1                                     Preference: 100
  IpPre: N/A                                 QosLocalID: N/A
  Tag: 0                                     State: Inactive Adv
OrigTblID: 0x0                               OrigVrf: default-vrf
  TableID: 0xa                               OrigAs: 0
  NibID: 0x22000001                          LastAs: 0
  AttrID: 0xffffffff                          Neighbor: FE80::34CC:E8FF:FE5B:C02
  Flags: 0x41                                 OrigNextHop: FE80::34CC:E8FF:FE5B:C02
  Label: NULL                                 RealNextHop: FE80::34CC:E8FF:FE5B:C02
  BkLabel: NULL                               BkNextHop: FE80::7685:45FF:FEAD:102
  SRLLabel: NULL                             BkSRLLabel: NULL
  Tunnel ID: Invalid                          Interface: GigabitEthernet1/0/2
BkTunnel ID: Invalid                         BkInterface: GigabitEthernet1/0/1
  FtnIndex: 0x0                              TrafficIndex: N/A
  Connector: N/A                             VpnPeerId: N/A
  Dscp: N/A

```

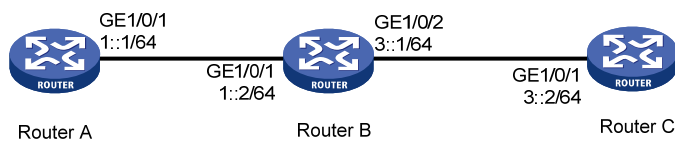
1.11.6 RIPng IPsec安全框架配置举例

1. 组网需求

- Router A、Router B 和 Router C 相连并通过 RIPng 来学习网络中的 IPv6 路由信息。
- 要求配置 IPsec 安全框架对 Router A、Router B 和 Router C 之间的 RIPng 报文进行有效性检查和验证。

2. 组网图

图1-7 RIPng IPsec 安全框架配置组网图



3. 配置步骤

- (1) 配置各接口的 IPv6 地址（略）
- (2) 配置 RIPng 基本功能

配置 Router A。

```

<RouterA> system-view
[RouterA] ripng 1
[RouterA-ripng-1] quit
[RouterA] interface gigabitethernet 1/0/1
[RouterA-GigabitEthernet1/0/1] ripng 1 enable
[RouterA-GigabitEthernet1/0/1] quit

```

配置 Router B。

```

<RouterB> system-view
[RouterB] ripng 1

```

```
[RouterB-ripng-1] quit
[RouterB] interface gigabitethernet 1/0/1
[RouterB-GigabitEthernet1/0/1] ripng 1 enable
[RouterB-GigabitEthernet1/0/1] quit
[RouterB] interface gigabitethernet 1/0/2
[RouterB-GigabitEthernet1/0/2] ripng 1 enable
[RouterB-GigabitEthernet1/0/2] quit
```

配置 Router C。

```
<RouterC> system-view
[RouterC] ripng 1
[RouterC-ripng-1] quit
[RouterC] interface gigabitethernet 1/0/1
[RouterC-GigabitEthernet1/0/1] ripng 1 enable
[RouterC-GigabitEthernet1/0/1] quit
```

(3) 配置 RIPng IPsec 安全框架

配置 Router A。创建名为 **protrf1** 的安全提议，报文封装形式采用传输模式，安全协议采用 **ESP** 协议。创建一条安全框架 **profile001**，协商方式为 **manual**，配置 **SPI** 和密钥。

```
[RouterA] ipsec transform-set protrf1
[RouterA-ipsec-transform-set-protrf1] esp encryption-algorithm 3des-cbc
[RouterA-ipsec-transform-set-protrf1] esp authentication-algorithm md5
[RouterA-ipsec-transform-set-protrf1] encapsulation-mode transport
[RouterA-ipsec-transform-set-protrf1] quit
[RouterA] ipsec profile profile001 manual
[RouterA-ipsec-profile-profile001-manual] transform-set protrf1
[RouterA-ipsec-profile-profile001-manual] sa spi inbound esp 256
[RouterA-ipsec-profile-profile001-manual] sa spi outbound esp 256
[RouterA-ipsec-profile-profile001-manual] sa string-key inbound esp simple abc
[RouterA-ipsec-profile-profile001-manual] sa string-key outbound esp simple abc
[RouterA-ipsec-profile-profile001-manual] quit
```

配置 Router B。创建名为 **protrf1** 的安全提议，报文封装形式采用传输模式，安全协议采用 **ESP** 协议。创建一条安全框架 **profile001**，协商方式为 **manual**，配置 **SPI** 和密钥。

```
[RouterB] ipsec transform-set protrf1
[RouterB-ipsec-transform-set-protrf1] esp encryption-algorithm 3des-cbc
[RouterB-ipsec-transform-set-protrf1] esp authentication-algorithm md5
[RouterB-ipsec-transform-set-protrf1] encapsulation-mode transport
[RouterB-ipsec-transform-set-protrf1] quit
[RouterB] ipsec profile profile001 manual
[RouterB-ipsec-profile-profile001-manual] transform-set protrf1
[RouterB-ipsec-profile-profile001-manual] sa spi inbound esp 256
[RouterB-ipsec-profile-profile001-manual] sa spi outbound esp 256
[RouterB-ipsec-profile-profile001-manual] sa string-key inbound esp simple abc
[RouterB-ipsec-profile-profile001-manual] sa string-key outbound esp simple abc
[RouterB-ipsec-profile-profile001-manual] quit
```

配置 Router C。创建名为 **protrf1** 的安全提议，报文封装形式采用传输模式，安全协议采用 **ESP** 协议。创建一条安全框架 **profile001**，协商方式为 **manual**，配置 **SPI** 和密钥。

```
[RouterC] ipsec transform-set protrf1
[RouterC-ipsec-transform-set-protrf1] esp encryption-algorithm 3des-cbc
```

```
[RouterC-ipsec-transform-set-protrf1] esp authentication-algorithm md5
[RouterC-ipsec-transform-set-protrf1] encapsulation-mode transport
[RouterC-ipsec-transform-set-protrf1] quit
[RouterC] ipsec profile profile001 manual
[RouterC-ipsec-profile-profile001-manual] transform-set protrf1
[RouterC-ipsec-profile-profile001-manual] sa spi inbound esp 256
[RouterC-ipsec-profile-profile001-manual] sa spi outbound esp 256
[RouterC-ipsec-profile-profile001-manual] sa string-key inbound esp simple abc
[RouterC-ipsec-profile-profile001-manual] sa string-key outbound esp simple abc
[RouterC-ipsec-profile-profile001-manual] quit
```

(4) RIPng 进程上应用 IPsec 安全框架

配置 Router A。

```
[RouterA] ripng 1
[RouterA-ripng-1] enable ipsec-profile profile001
[RouterA-ripng-1] quit
```

配置 Router B。

```
[RouterB] ripng 1
[RouterB-ripng-1] enable ipsec-profile profile001
[RouterB-ripng-1] quit
```

配置 Router C。

```
[RouterC] ripng 1
[RouterC-ripng-1] enable ipsec-profile profile001
[RouterC-ripng-1] quit
```

4. 验证配置

以上配置完成后，Router A、Router B 和 Router C 之间的 RIPng 报文将被加密传输。