

# IPv6路由基础



# 前言

---

- 在一个典型的数据通信网络中往往存在多个不同的IP网段，数据在不同的IP网段之间交互是需要借助三层设备的，这些设备具备路由能力，能够实现数据的跨网段转发。
- 路由是数据通信网络中最基本的要素。
- 路由信息是指导报文转发的路径信息，路由过程就是报文转发的过程。
- 本课程将会向读者介绍路由的基本概念及技术原理。

# 课程目标

---

学完本课程后，您将能够：

- 描述路由器的基本工作原理
- 理解路由器选择最优路由的机制
- 描述路由表的具体内容
- 理解静态路由的基础功能及进阶特性

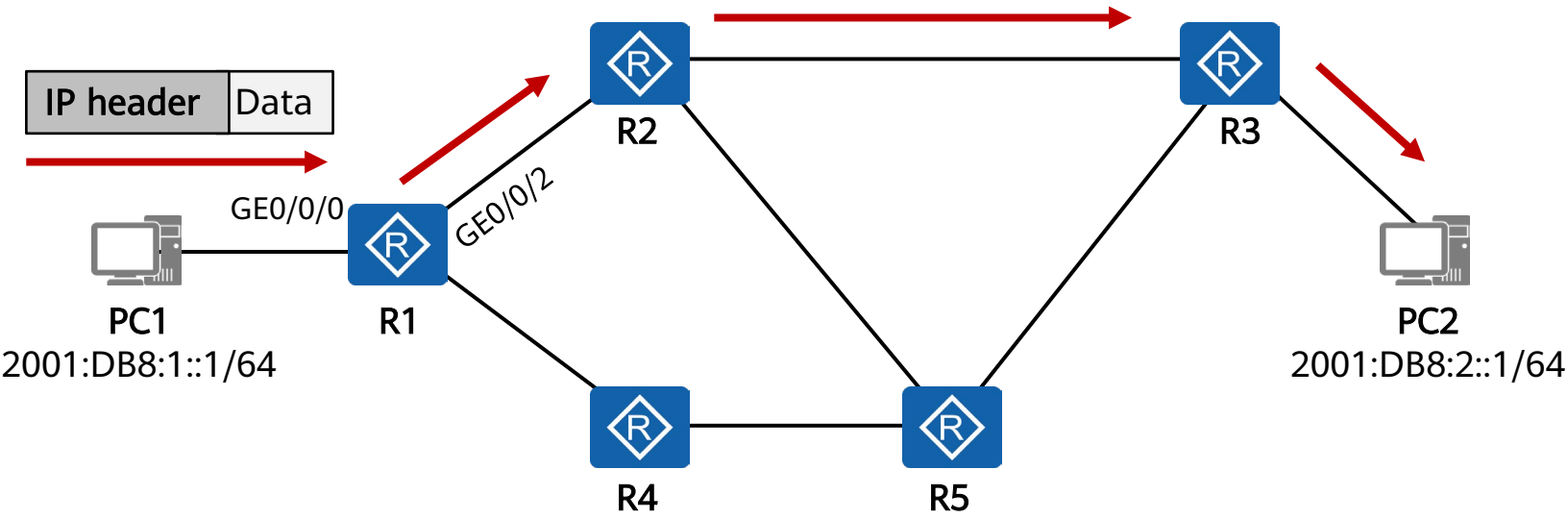
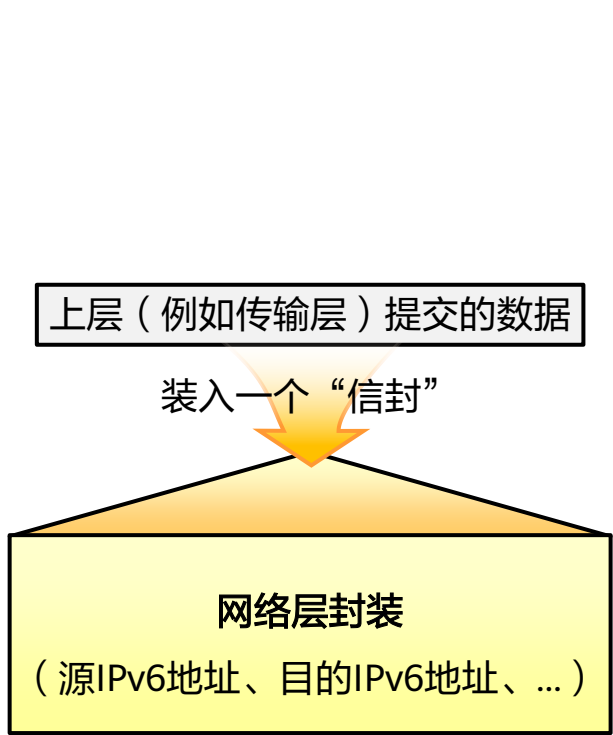
# 目录

---

1. 路由的基本概念
2. 静态路由基础
3. 静态路由进阶



# 什么是路由？



R1的IPv6路由表

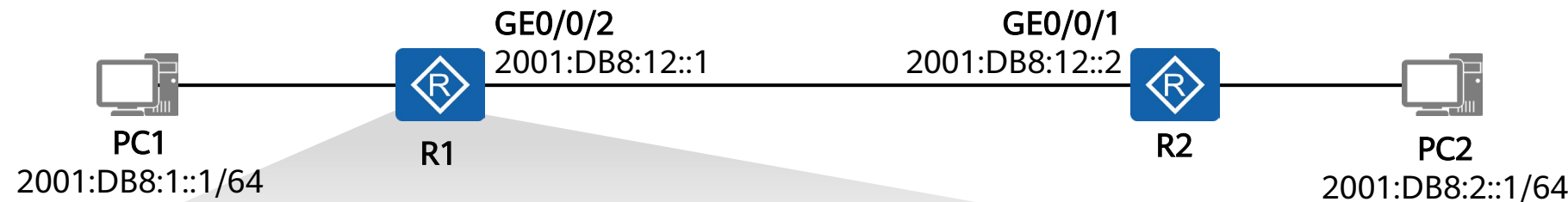
| 目的网络         | 前缀长度 | 下一跳              | 优先级 | 开销  | 协议     | 接口      |
|--------------|------|------------------|-----|-----|--------|---------|
| 2001:DB8:1:: | 64   | 2001:DB8:1::FFFF | 0   | 0   | Direct | GE0/0/0 |
| 2001:DB8:2:: | 64   | 2001:DB8:12::2   | 60  | 0   | Static | GE0/0/2 |
| ...          | ...  | ...              | ... | ... | ...    | ...     |



# 路由表

- 路由器通过各种方式发现路由表项。
- 路由器选择最优的路由条目放入路由表中。
- 路由表指导设备对IP报文的转发。
- 路由器通过对路由表的管理实现对路径信息的管理。

# 路由表中的路由信息



| 目的网络         | 前缀长度 | 下一跳            | 优先级 | 开销 | 协议     | 接口      |
|--------------|------|----------------|-----|----|--------|---------|
| 2001:DB8:2:: | 64   | 2001:DB8:12::2 | 60  | 0  | Static | GE0/0/2 |

- 目的网络（Destination）：此路由对应的目的网络/主机的地址。
- 前缀长度（Prefix Length）：此路由的前缀长度。
- 下一跳（Next Hop）：对于本路由器而言，到达该路由指向的目的网络的下一跳地址。
- 优先级（Preference）：路由优先级，优先级最高（数值最小）者将成为当前的最优路由。
- 开销（Cost）：路由开销值，当到达同一目的地的多条路由具有相同的路由优先级时，Cost值最小的将成为当前的最优路由。
- 协议（Protocol）：该路由的协议类型，也即路由器是通过什么协议获知该路由的。
- 接口（Interface）：路由的出接口，指明数据将从本路由器的哪个接口转发出去。

# 在网络设备上查看路由表

[R1] display ipv6 routing-table  
Routing Table : Public  
Destinations : 7      Routes : 7

|        |        |   |              |   |                      |              |   |        |   |      |
|--------|--------|---|--------------|---|----------------------|--------------|---|--------|---|------|
| 路由条目 1 | 目的网络地址 | ← | Destination  | : | FC00:23::            | PrefixLength | : | 64     | → | 前缀长度 |
|        | 下一跳地址  | ← | NextHop      | : | FC00:12::2           | Preference   | : | 60     | → | 优先级  |
|        | 开销     | ← | Cost         | : | 0                    | Protocol     | : | Static | → | 协议类型 |
|        |        |   | RelayNextHop | : | ::                   | TunnelID     | : | 0x0    |   |      |
|        | 出接口    | ← | Interface    | : | GigabitEthernet0/0/3 | Flags        | : | RD     |   |      |
| 路由条目 2 |        |   | Destination  | : | FC00:12::            | PrefixLength | : | 64     |   |      |
|        |        |   | NextHop      | : | FC00:12::1           | Preference   | : | 0      |   |      |
|        |        |   | Cost         | : | 0                    | Protocol     | : | Direct |   |      |
|        |        |   | RelayNextHop | : | ::                   | TunnelID     | : | 0x0    |   |      |
|        |        |   | Interface    | : | GigabitEthernet0/0/3 | Flags        | : | D      |   |      |

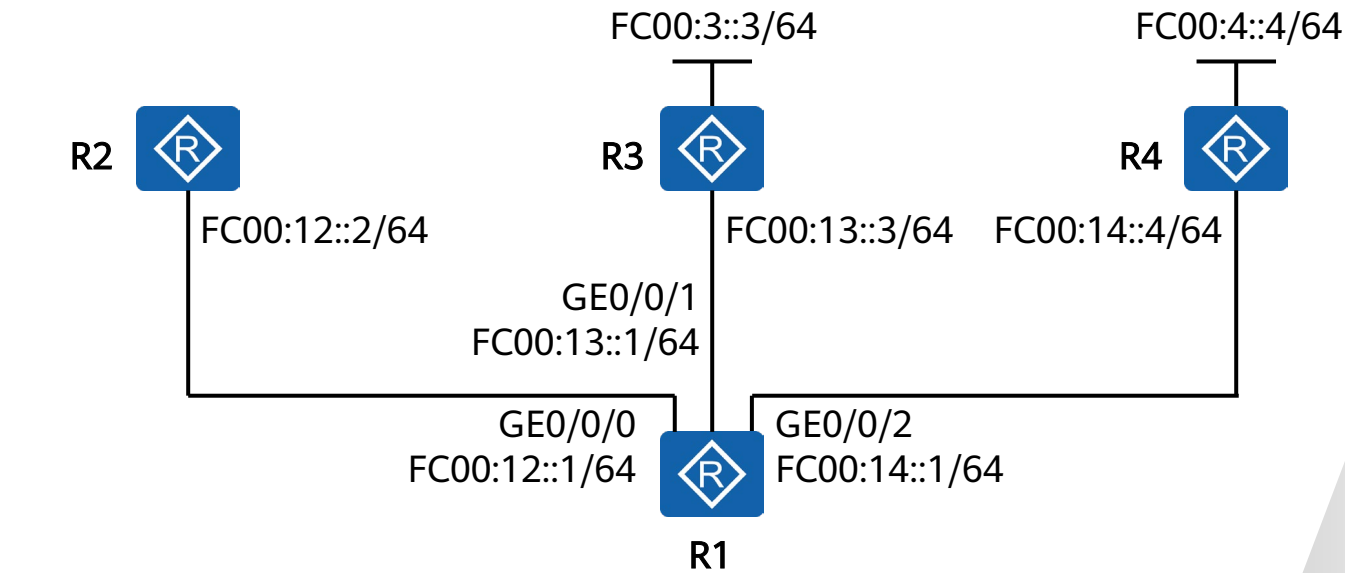
... ..





# 路由类型及发现方式

- 为实现数据转发，路由器需要发现路由并维护路由表。



| 目的网络 | 前缀长度 | 下一跳 | 协议  | 接口  |
|------|------|-----|-----|-----|
| ...  | ...  | ... | ... | ... |

R1的路由表

## 直连路由

到达本地直连网络的路由

## 静态路由

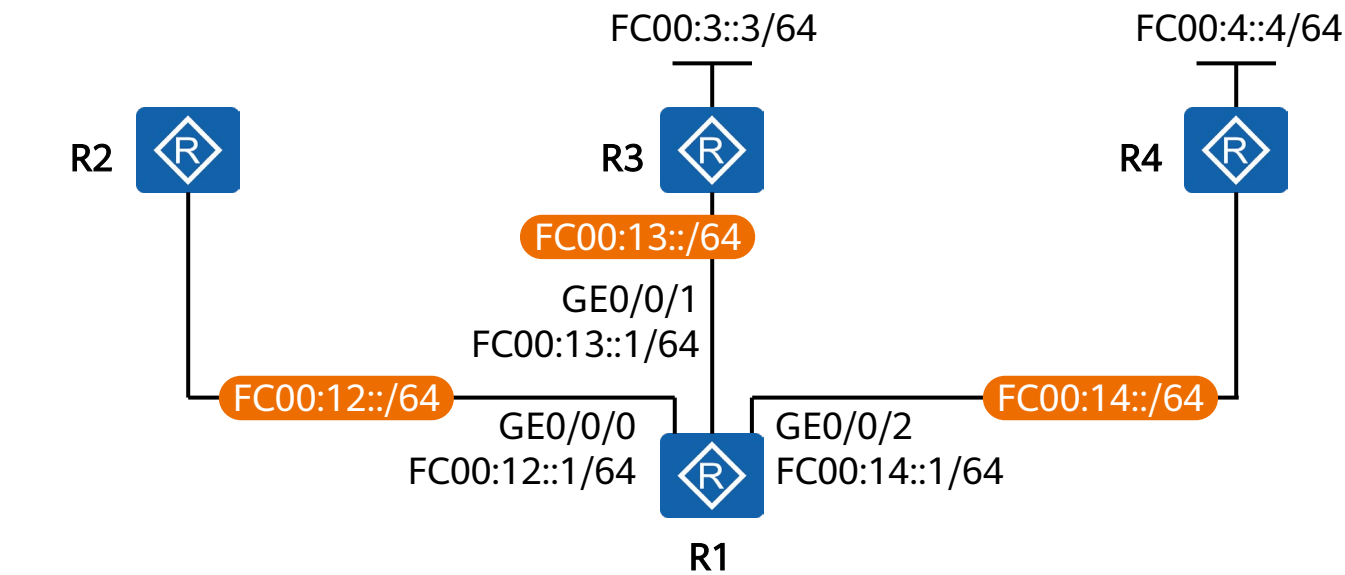
网络管理员手工为设备配置的路由

## 动态路由

设备通过动态路由协议所学习到的路由



# 直连路由

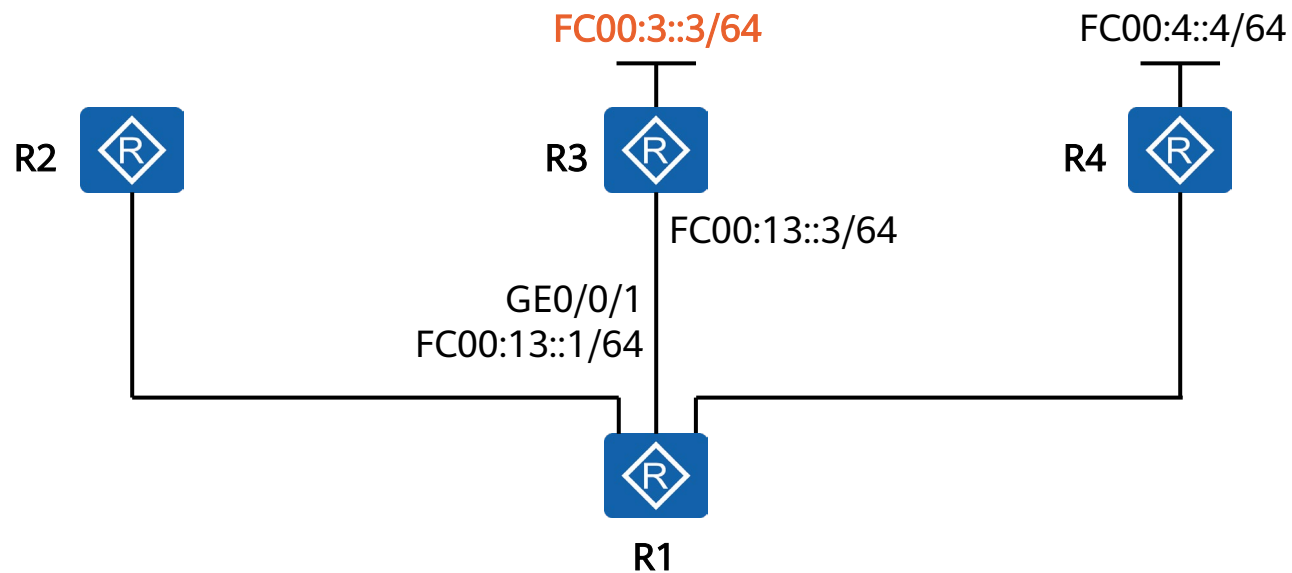


R1的路由表

| 目的网络      | 前缀长度 | 下一跳        | 协议     | 接口      |
|-----------|------|------------|--------|---------|
| FC00:12:: | 64   | FC00:12::1 | Direct | GE0/0/0 |
| FC00:13:: | 64   | FC00:13::1 | Direct | GE0/0/1 |
| FC00:14:: | 64   | FC00:14::1 | Direct | GE0/0/2 |
| ...       | ...  | ...        | ...    | ...     |

- 直连路由的目的网络是路由器自身某个接口所在的网络。
- 直连路由的发现是路由器自动完成的，无需人为干预。

# 静态路由

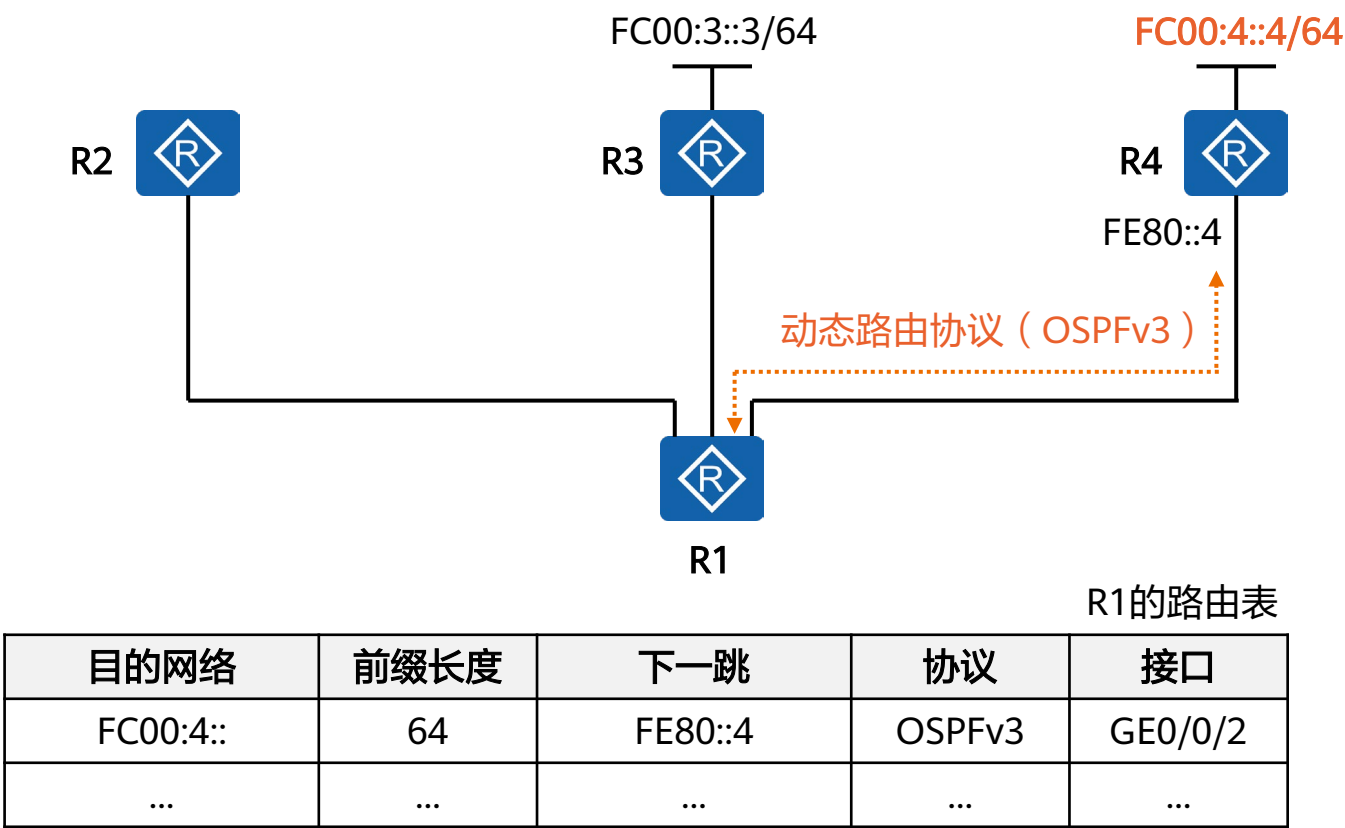


R1的路由表

| 目的网络     | 前缀长度 | 下一跳        | 协议     | 接口      |
|----------|------|------------|--------|---------|
| FC00:3:: | 64   | FC00:13::3 | Static | GE0/0/1 |
| ...      | ...  | ...        | ...    | ...     |

- 在设备上通过网络管理员手动配置的路由称为静态路由。
- 静态路由配置方便，对系统要求低，适用于拓扑结构简单并且稳定的小型网络。
- 缺点是不能自动适应网络拓扑的变化。

# 动态路由



- 通过动态路由协议发现的路由称为动态路由。
- 动态路由协议有自己的路由算法，能够自动适应网络拓扑的变化，适用于具有一定数量三层设备的网络。
- 缺点是管理维护对用户的要求较高，对设备的性能要求高于静态路由，并将占用一定的网络资源和性能资源。

# 动态路由协议的分类

## 按工作区域分类

### IGP ( Interior Gateway Protocol ) 内部网关协议

RIPng

OSPFv3

IS-IS

### EGP ( Exterior Gateway Protocol ) 外部网关协议

BGP

根据作用范围不同，路由协议可分为：

- IGP：在一个自治系统（ Autonomous System, AS ）内运行的路由协议。
- EGP：实现不同自治系统之间路由信息交互的路由协议。

## 按工作机制及算法分类

### ( Distance Vector Routing Protocol ) 距离矢量路由协议

RIPng

BGP

### ( Link-State Routing Protocol ) 链路状态路由协议

OSPFv3

IS-IS

根据使用算法不同，路由协议可分为：

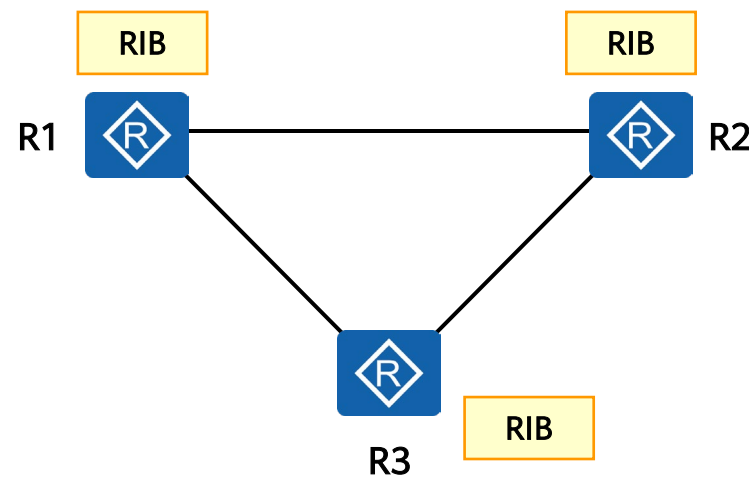
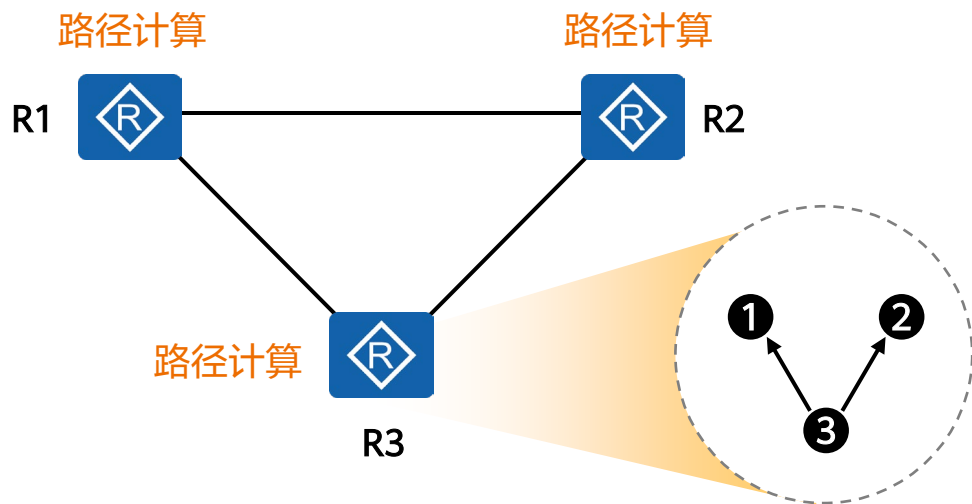
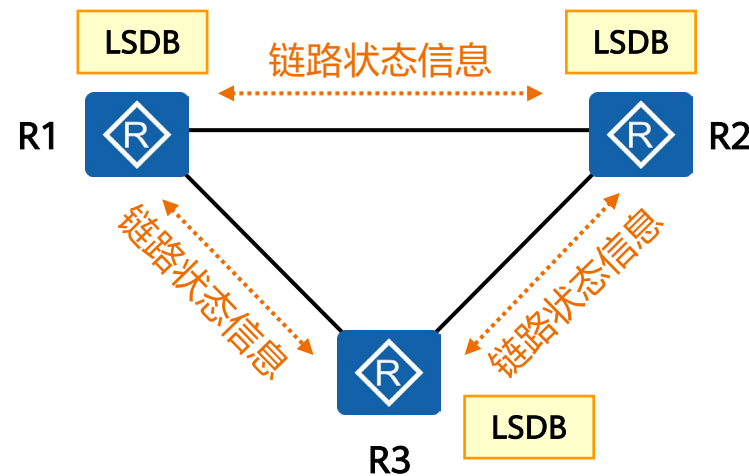
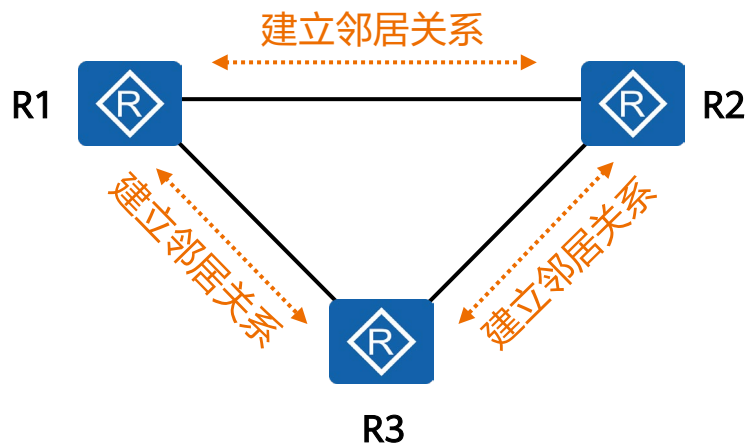
- 距离矢量路由协议：在设备之间交互路由信息，使用跳数（ Hop Count ）来衡量到达目的地址的距离。其中，BGP也被称为路径矢量协议（ Path-Vector ）。
- 链路状态路由协议：在设备之间交互链路状态信息，运行特定算法进行无环路由计算，目前被广泛应用。

# 距离矢量路由协议



- 距离矢量路由协议在设备之间交互路由信息。
- 使用跳数（Hop Count）来衡量到达目的地址的距离。

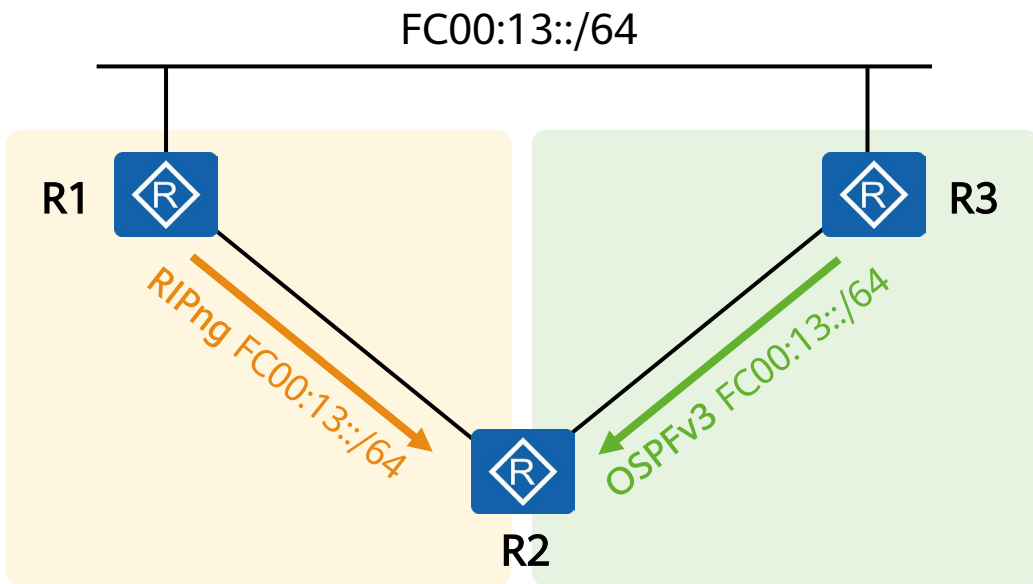
# 链路状态路由协议



1 2  
3 4

LSDB: LinkState DataBase  
RIB : Routing Information Base

# 路由优先级



R2同时从RIPng及OSPFv3学习到去往FC00:13::/64的  
路由，它将如何优选？

- 对于相同的目的地，不同的路由协议（包括静态路由）可能会发现不同的路由。
- 在某一时刻，到某一目的地的当前路由仅能由唯一的路由协议来决定。
- 为了判断最优路由，各路由协议（包括直连路由及静态路由）都被赋予了一个优先级，当存在多个路由信息源时，具有较高优先级（值更小）的路由协议发现的路由将成为最优路由，最优路由被放入设备路由表中。



# 路由协议缺省时的优先级

| 路由协议的类型       | 优先级 |
|---------------|-----|
| Direct        | 0   |
| OSPF          | 10  |
| IS-IS Level-1 | 15  |
| IS-IS Level-2 | 18  |
| Static        | 60  |
| RIP           | 100 |
| IBGP          | 200 |
| EBGP          | 20  |
| ...           | ... |

[R1] display ipv6 routing-table

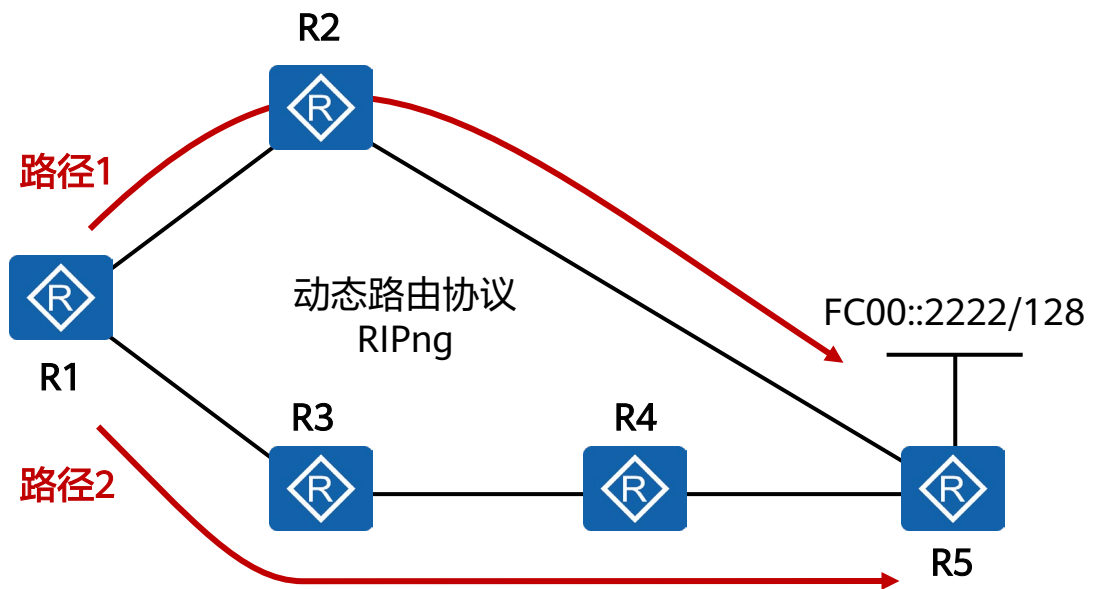
Routing Table : Public

Destinations : 7

Routes : 7

|              |                        |              |          |
|--------------|------------------------|--------------|----------|
| Destination  | : FC00::2222           | PrefixLength | : 64     |
| NextHop      | : FC00:12::            | Preference   | : 60     |
| Cost         | : 0                    | Protocol     | : Static |
| RelayNextHop | : ::                   | TunnelID     | : 0x0    |
| Interface    | : GigabitEthernet0/0/3 | Flags        | : RD     |
| ...          | ...                    |              |          |

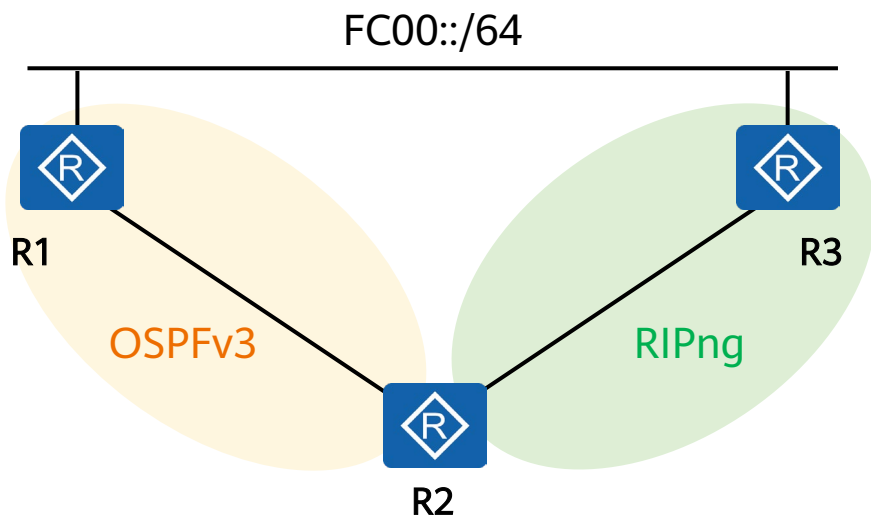
# 度量值



- 路由度量值（Metric）表示到达路由所指目的地址的代价。
- 度量值也被称为开销（Cost）。
- 当路由器通过某种路由协议发现了多条到达同一个目的网络的路由时度量值将作为路由优选的依据之一。
- 度量值示例：跳数、带宽、时延、代价、负载、可靠性等。

# 路由选择决策

路由优先级如何影响路由选择



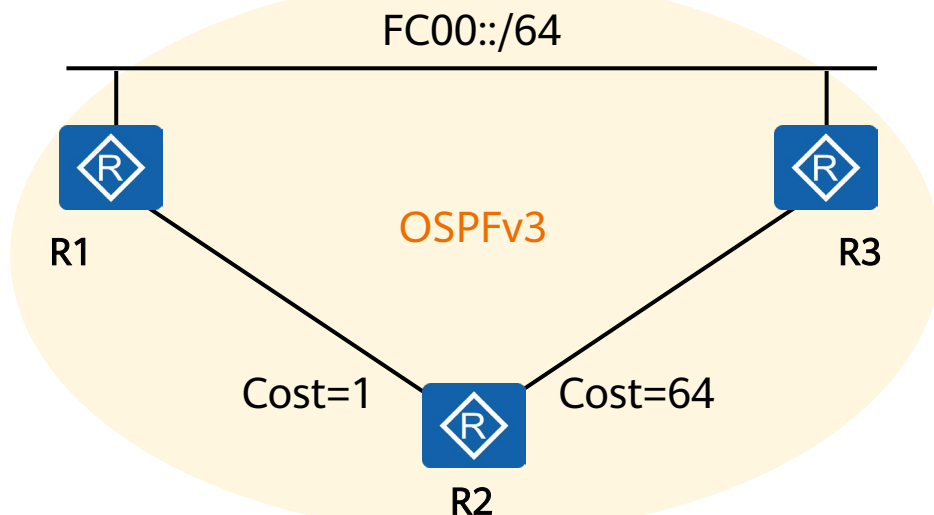
发现到达FC00::/64  
的OSPFv3路由

发现到达FC00::/64  
的RIPng路由

比较路由优先级

路由优先级值更小的胜出

路由度量值如何影响路由选择



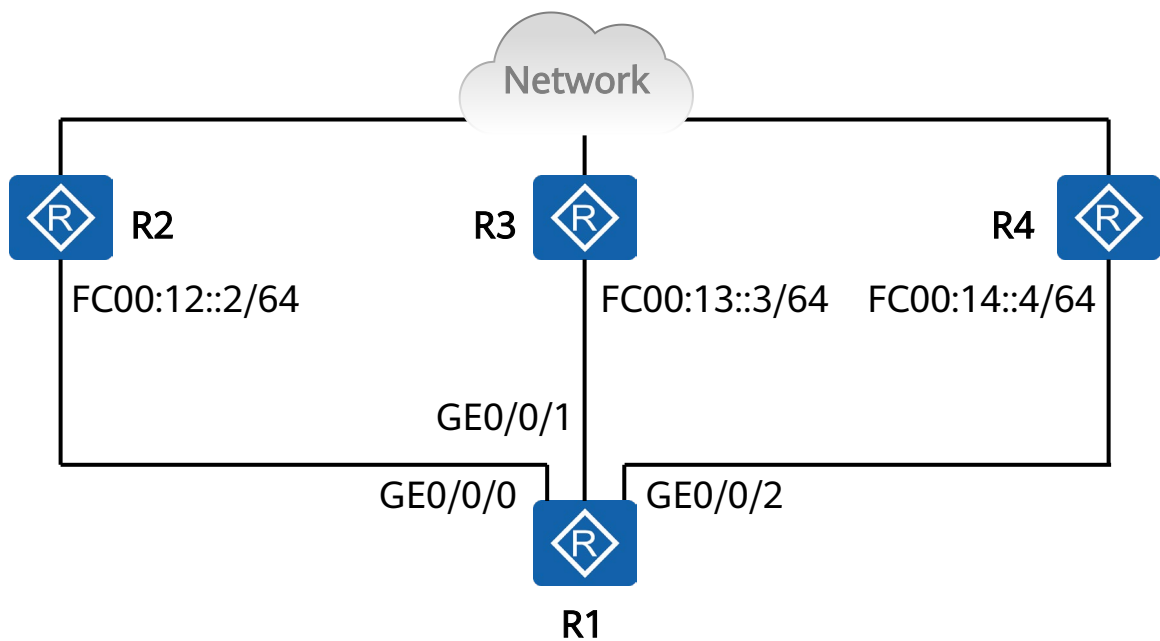
发现到达FC00::/64  
的OSPFv3路由

发现到达FC00::/64  
的OSPFv3路由

“视情况” 比较路由度量值

路由度量值更小的胜出

# 路由查询的最长前缀匹配原则：案例



R1的路由表

| 目的网络         | 前缀长度 | 下一跳        | 协议     | 接口      |
|--------------|------|------------|--------|---------|
| FC00:1:2::   | 48   | FC00:12::2 | Static | GE0/0/0 |
| FC00:1:2:3:: | 64   | FC00:13::3 | Static | GE0/0/1 |
| FC00:1:2:4:: | 64   | FC00:14::4 | Static | GE0/0/2 |

R1的路由表如图所示。

- 当其收到发往FC00:1:2:3:4::1的报文，会如何转发？
- 当其收到发往FC00:1:2:4:5::1的报文，会如何转发？
- 当其收到发往FC00:1:2:B::1的报文，会如何转发？

下文以发往FC00:1:2:3:4::1的报文为例解析。

# 路由查询的最长前缀匹配原则：解析

报文的目的IPv6地址 FC00 : 0001 : 0002 : 0003 : 0004 : 0000 : 0000 : 0001

目的地址与路由表项1、2、3的掩码分别作“逻辑与”运算，然后与路由前缀进行匹配



路由表项1  
FC00:1:2::/48

/48  
FC00: 0001 : 0002 : 0000 : 0000 : 0000 : 0000 : 0000

匹配长度48

路由表项2  
FC00:1:2:3::/64

/64  
FC00 : 0001 : 0002 : 0003 : 0000 : 0000 : 0000 : 0000

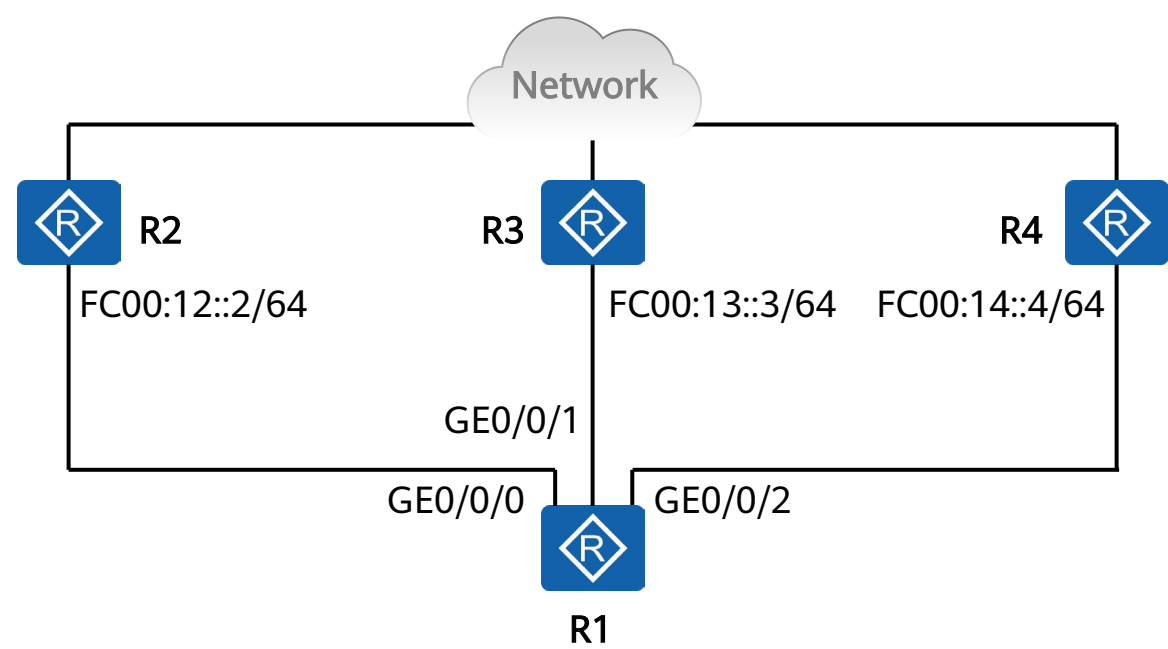
匹配长度64

路由表项3  
FC00:1:2:4::/64

/64  
FC00 : 0001 : 0002 : 0004 : 0000 : 0000 : 0000 : 0000

不匹配

# 路由查询的最长前缀匹配原则：结果



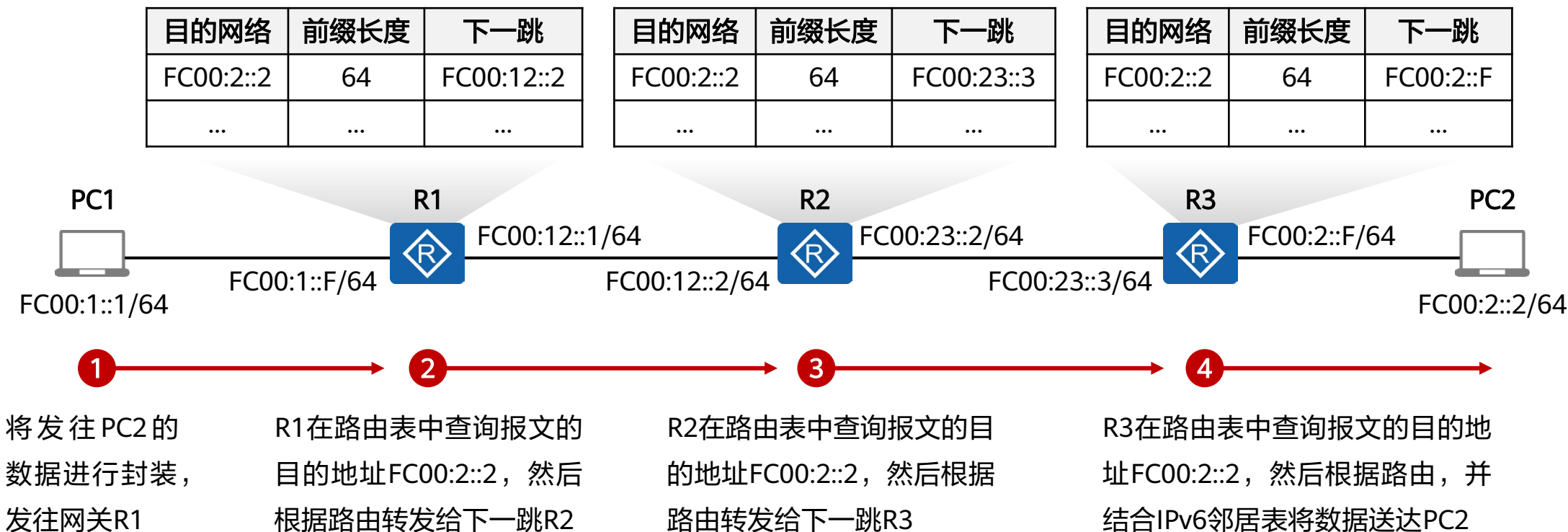
R1的路由表

| 目的网络         | 前缀长度 | 下一跳        | 协议     | 接口      |
|--------------|------|------------|--------|---------|
| FC00:1:2::   | 48   | FC00:12::2 | Static | GE0/0/0 |
| FC00:1:2:3:: | 64   | FC00:13::3 | Static | GE0/0/1 |
| FC00:1:2:4:: | 64   | FC00:14::4 | Static | GE0/0/2 |

R1的路由表如图所示。

- 当其收到发往FC00:1:2:3:4::1的报文，转发给R3。
- 当其收到发往FC00:1:2:4:5::1的报文，转发给R4。
- 当其收到发往FC00:1:2:B::1的报文，转发给R2。

# 数据包的路由过程



# IP路由表小结

- 当路由器从多种不同的途径获知到达同一个目的网段的路由（这些路由的目的网络地址及前缀长度均相同）时，会选择路由优先级值最小的路由；如果这些路由学习自相同的路由协议，则选择度量值最优的。最优的路由加入路由表。
- 当路由器收到一个数据包时，会在自己的路由表中查询数据包的目的IP地址。如果能够找到匹配的路由表项，则依据表项所指示的出接口及下一跳来转发数据；如果没有匹配的表项，则丢弃该数据包。
- 数据包的路由过程是逐跳执行的，数据包从源到目的地沿路径每个路由器都必须有关于目的网段的路由，否则就会造成丢包。
- 数据通信往往是双向的，因此要关注数据的往返路由。



# 目录

---

1. 路由的基本概念
2. **静态路由基础**
3. 静态路由进阶

# 静态路由

- 在设备上通过网络管理员手动配置的路由称为静态路由。
- 静态路由配置方便，对系统要求低，适用于拓扑结构简单并且稳定的小型网络。
- 静态路由也广泛应用于大型网络中的部分场景，例如大型园区网络出口等。
- 在大型网络中，网段及设备数量多、网络拓扑复杂，单纯使用静态路由无法满足需求，且配置维护工作量巨大。静态路由不能自动适应网络拓扑的变化。

# 静态路由配置命令

## 1. 关联下一跳地址的方式

```
[Huawei] ipv6 route-static dest-ipv6-address prefix-length nexthop-ipv6-address
```

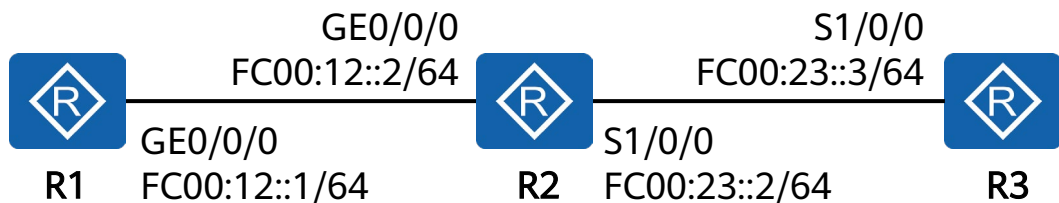
## 2. 关联出接口的方式

```
[Huawei] ipv6 route-static dest-ipv6-address prefix-length interface-type interface-number
```

## 3. 关联出接口和下一跳IP方式

```
[Huawei] ipv6 route-static dest-ipv6-address prefix-length interface-type interface-number nexthop-ipv6-address
```

# 静态路由配置举例



- 在路由器上完成静态路由配置，使得R1与R3能够通信。
- 因报文是逐跳转发的，所以每一跳路由设备上都需要配置到达目的网段的路由。
- 另外需要注意通信是双向的，针对通信过程中的往返流量，都需关注沿途设备上的路由配置。

R1的配置如下：

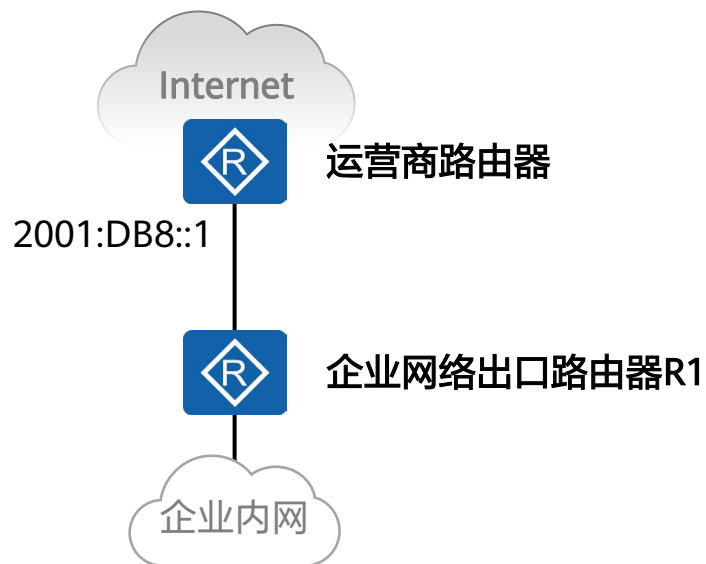
```
[R1] ipv6 route-static FC00:23:: 64 FC00:12::2
```

R3的配置如下：

```
[R3] ipv6 route-static FC00:12:: 64 Serial 1/0/0
```

# 缺省路由

- 缺省路由是一种特殊的路由，是当报文没有在路由表中找到匹配的具体路由时才使用的路由。
- 如果报文的目的地不能与路由表的任何目的地相匹配，那么该报文将选取缺省路由进行转发。
- IPv6缺省路由在路由表中的形式为::/0，缺省路由也称为默认路由。
- 缺省路由一般用于网络出口：通过配置缺省路由使出口设备能够转发到达Internet上任意地址的报文。

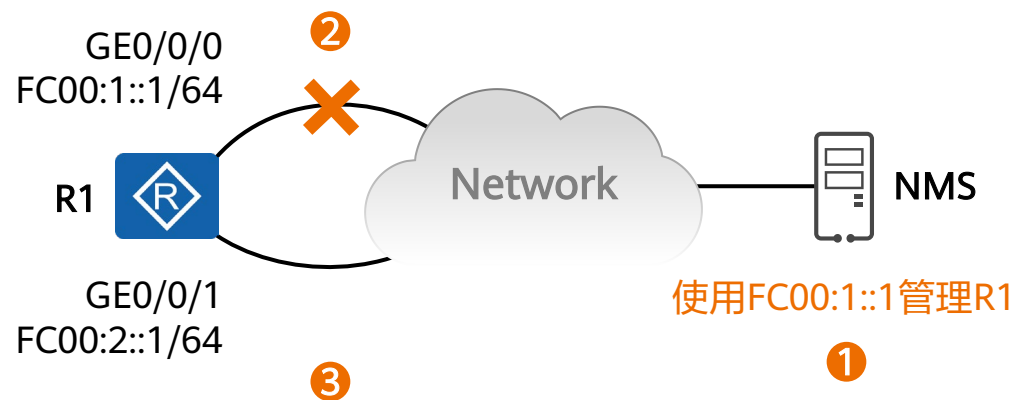


```
[R1] ipv6 route-static :: 0 2001:DB8::1
```

# 知识点：环回接口技术背景

## 未使用环回口时

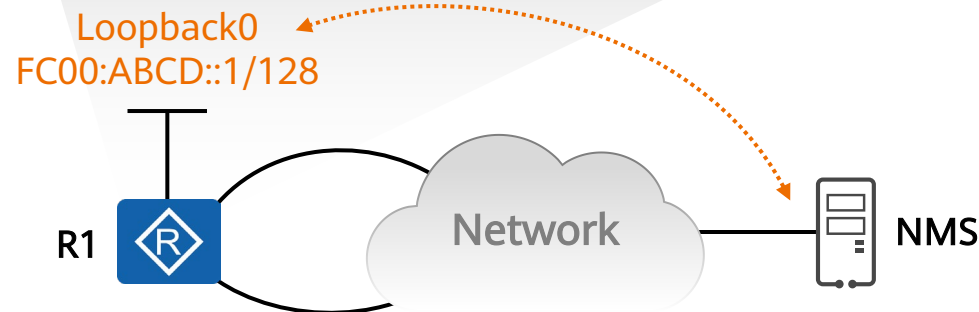
当GE0/0/0接口故障或不可达时，NMS丢失与R1的连接



然而，其实通过GE0/0/1接口依然可以到达R1，但对于NMS而言，此时需要重建会话

## 使用环回口

```
[R1] interface LoopBack 0
[R1-LoopBack0] ipv6 enable
[R1-LoopBack0] ipv6 address FC00:ABCD::1 128
```



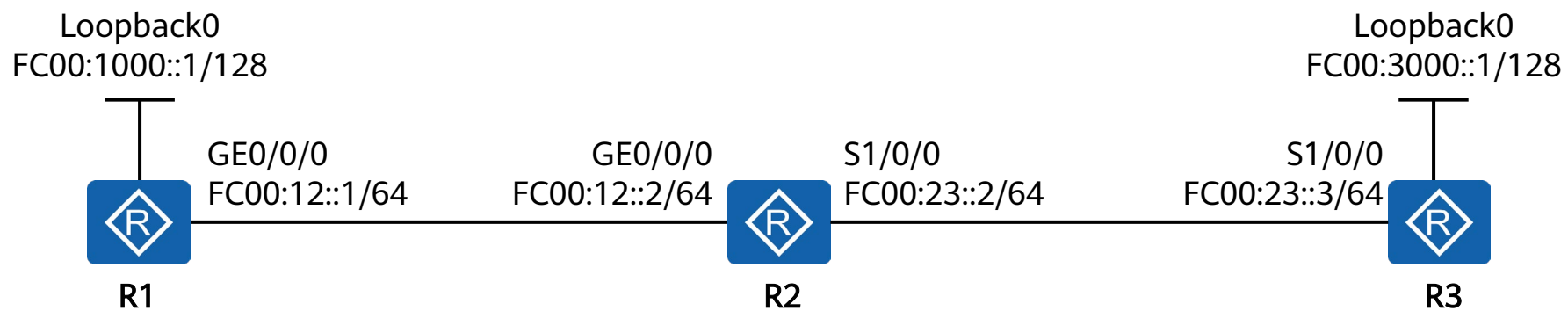
可以在R1上创建环回接口并为其分配IPv6地址，将到达该地址的路由通告到Network，如此一来，只要R1的至少一个物理接口依然活跃，NMS到R1的会话都可以保持。

# 知识点：环回接口（ Loopback ）

- Loopback接口，也叫环回接口，是一种逻辑接口。
- 使用**interface loopback**命令加上接口编号可创建Loopback接口。创建完成后即可为该接口配置网络地址。
- Loopback接口被创建后，是永远不会DOWN掉的（除非人为关闭），因此非常稳定。
- Loopback接口常用于：
  - 模拟设备的直连网段，可用于网络测试。
  - 可用于设备管理（ Loopback接口比较稳定 ）。
  - 供其他协议使用，例如OSPF、BGP、MPLS等的Router-ID。
  - 作为SNMP traps消息的源地址。
  - 其他用途（ Loopback接口的用途十分广泛 ）。

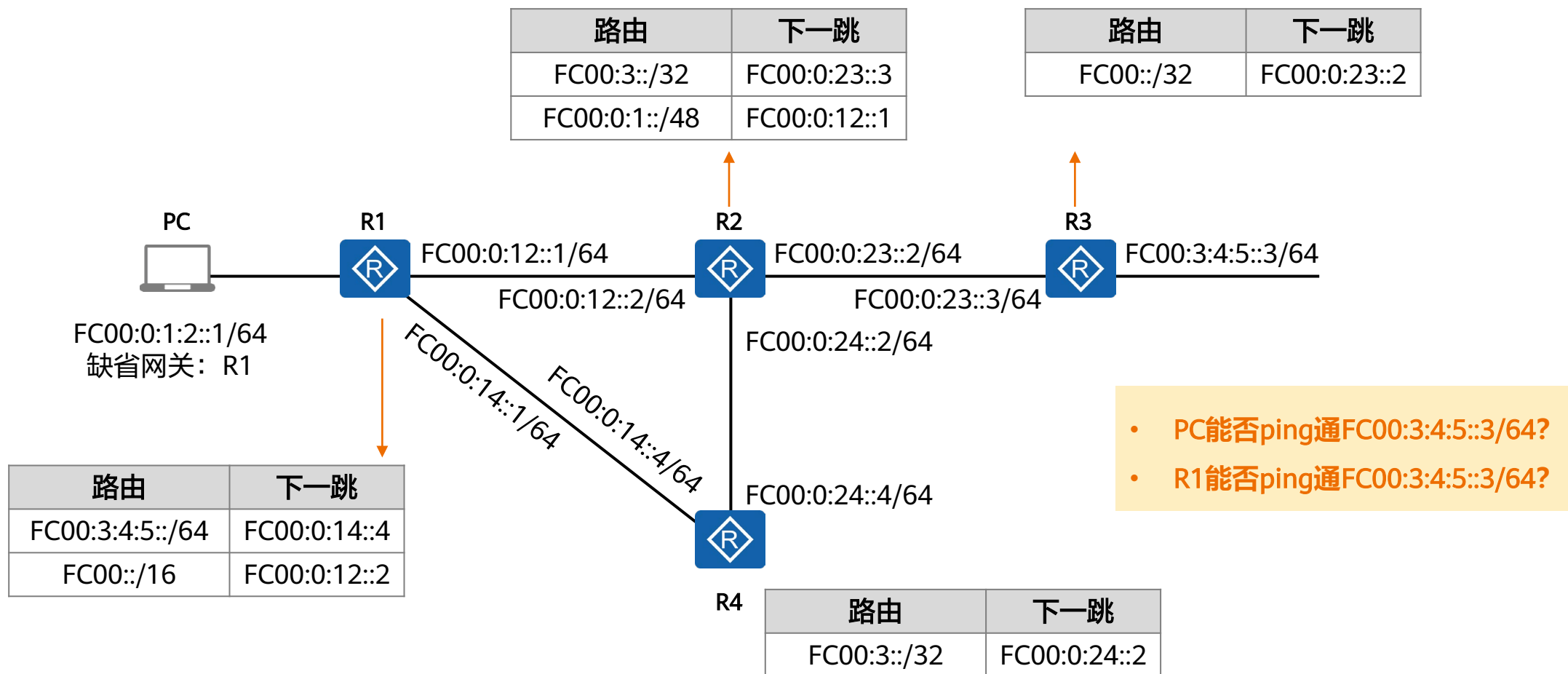
# 案例：IPv6静态路由部署

- 在如下场景中，如果要实现全网互通，需要分别在三台路由器上配置哪些IPv6静态路由？





# 案例：IPv6静态路由故障诊断



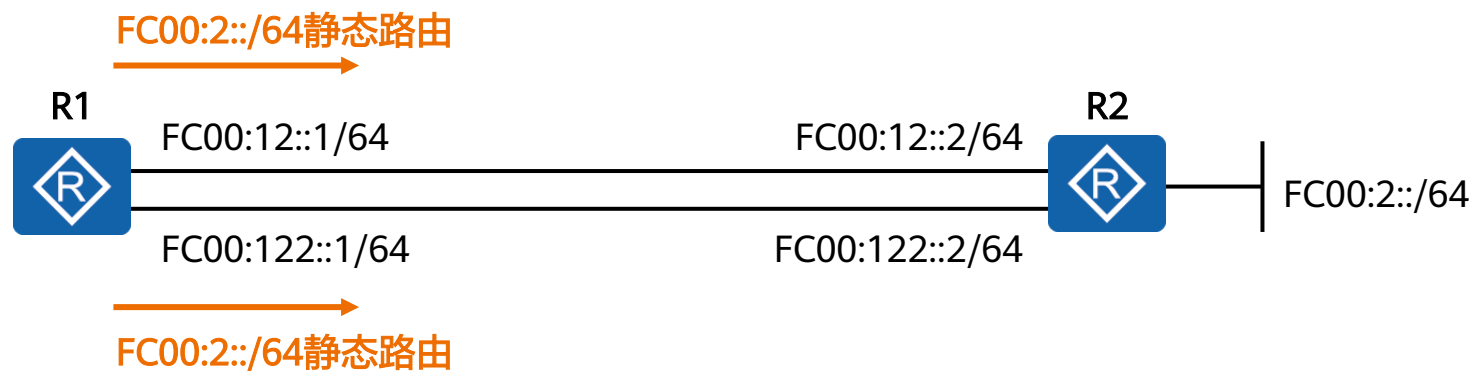
# 目录

---

1. 路由的基本概念
2. 静态路由基础
3. **静态路由进阶**
  - **浮动路由**
  - 路由汇总
  - 路由迭代
  - 静态路由与BFD
  - 静态路由与NQA

# 等价路由与负载分担

当多条路由的路由优先级和路由度量都相同时，这几条路由就称为等价路由，多条等价路由可以实现负载分担。  
当这几条路由为非等价路由时，就可以实现路由备份。

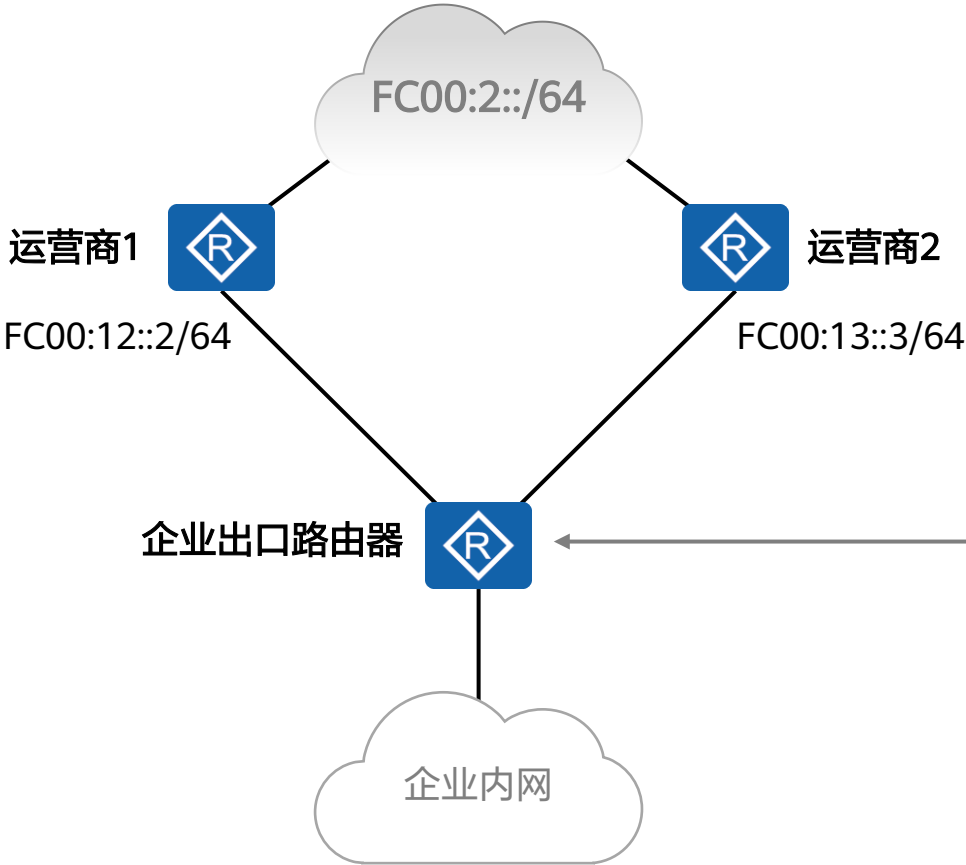


R1的路由表

| 目的网络     | 前缀长度 | 下一跳         | 协议     |
|----------|------|-------------|--------|
| FC00:2:: | 64   | FC00:12::2  | Static |
| FC00:2:: | 64   | FC00:122::2 | Static |
| ...      | ...  | ...         | ...    |

来源相同、Cost相同的路由都会被加入路由表，形成的路由为等价路由（路由指向的目的网段相同，但是具有不同的下一跳地址），设备会将流量分布到多条路径上。

# 浮动路由技术背景

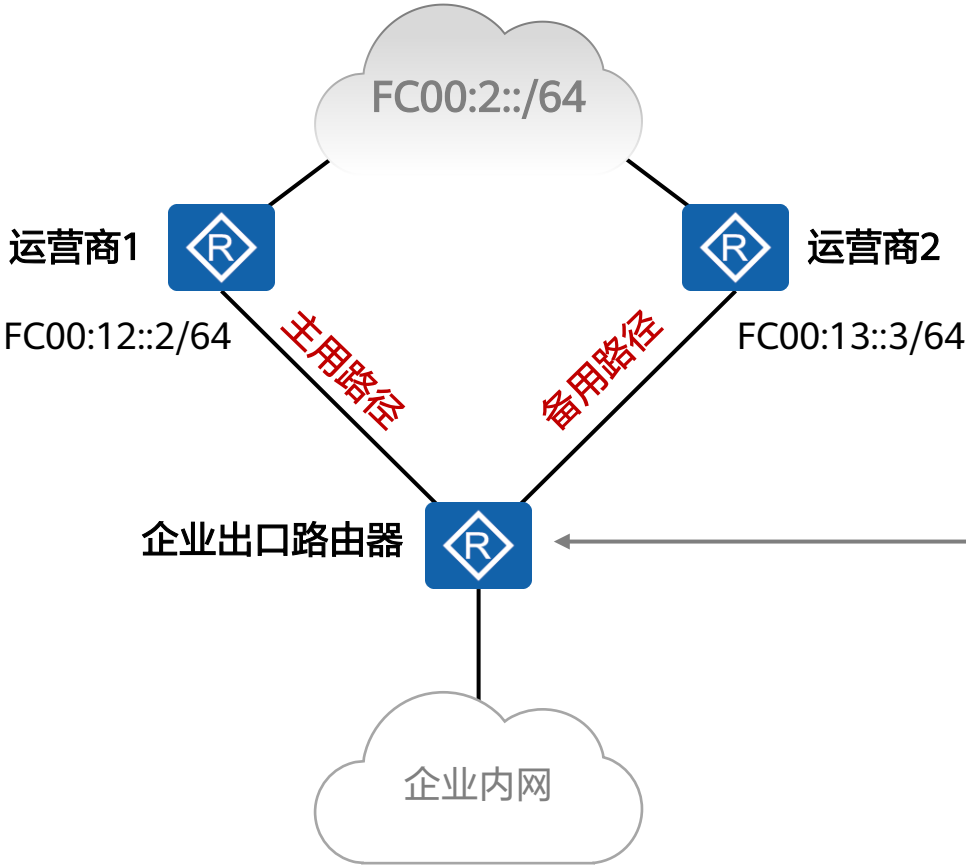


由于企业出口路由器上配置的两条静态路由的优先级一样，Cost也一样，路由同时被装载进了路由表进行等价负载分担。如果只是希望实现主备路径呢？

```
[Huawei] ipv6 route-static FC00:2:: 64 FC00:12::2  
[Huawei] ipv6 route-static FC00:2:: 64 FC00:13::3
```

| 目的网络     | 前缀长度 | 下一跳        | 协议     | 优先级 |
|----------|------|------------|--------|-----|
| FC00:2:: | 64   | FC00:12::2 | Static | 60  |
| FC00:2:: | 64   | FC00:13::3 | Static | 60  |

# 浮动路由



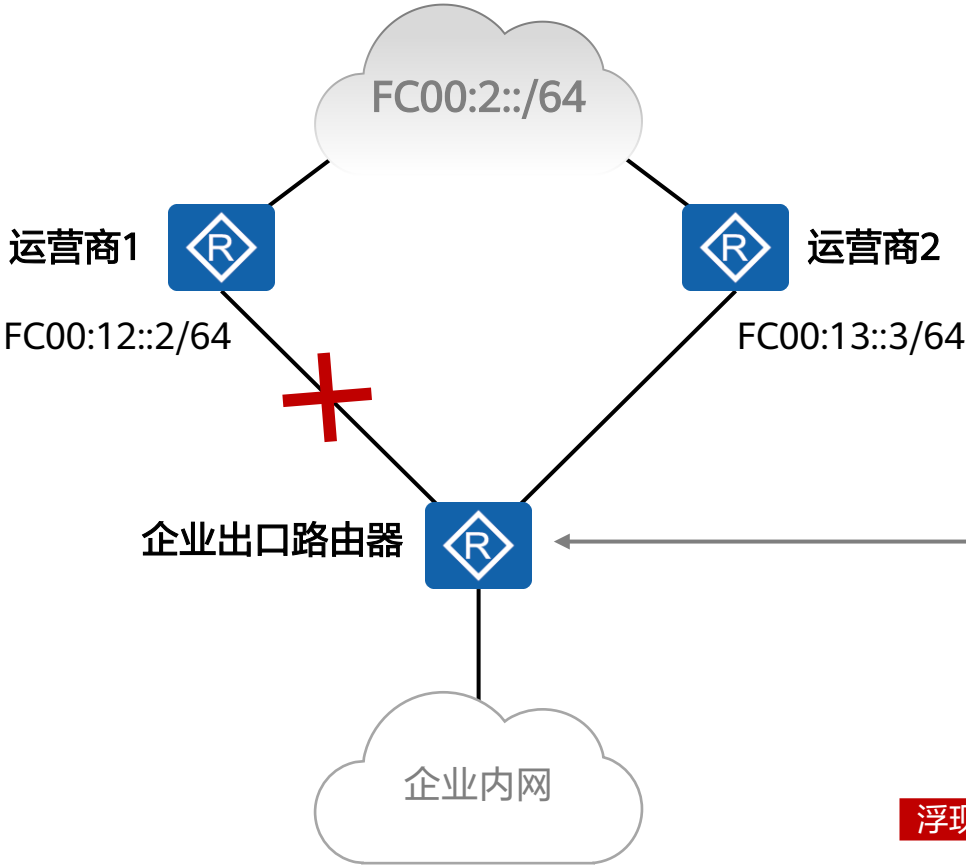
静态路由支持配置时手动指定优先级，可以通过配置目的地址/前缀长度相同、优先级不同、下一跳不同的静态路由，实现转发路径的备份。

浮动路由是主用路由的备份，保证链路故障时提供备份路由。主用路由可达时该备份路由不会出现在路由表。

```
[Huawei] ipv6 route-static FC00:2:: 64 FC00:12::2
[Huawei] ipv6 route-static FC00:2:: 64 FC00:13::3 preference 80
```

| 目的网络     | 前缀长度 | 下一跳        | 协议     | 优先级 |
|----------|------|------------|--------|-----|
| FC00:2:: | 64   | FC00:12::2 | Static | 60  |
| FC00:2:: | 64   | FC00:13::3 | Static | 80  |

# 浮动路由



当主用路径故障时，主用路由失效，此时ipv6 route-static FC00:2:: 64 FC00:13::3 preference 80路由在路由表中“浮现”出来，去往FC00:2::/64的流量从运营商2到达。

```
[Huawei] ipv6 route-static FC00:2:: 64 FC00:12::2
[Huawei] ipv6 route-static FC00:2:: 64 FC00:13::3 preference 80
```

| 目的网络               | 前缀长度      | 下一跳               | 协议            | 优先级       |
|--------------------|-----------|-------------------|---------------|-----------|
| FC00:2::           | 64        | FC00:12::2        | Static        | 60        |
| <b>浮现</b> FC00:2:: | <b>64</b> | <b>FC00:13::3</b> | <b>Static</b> | <b>80</b> |

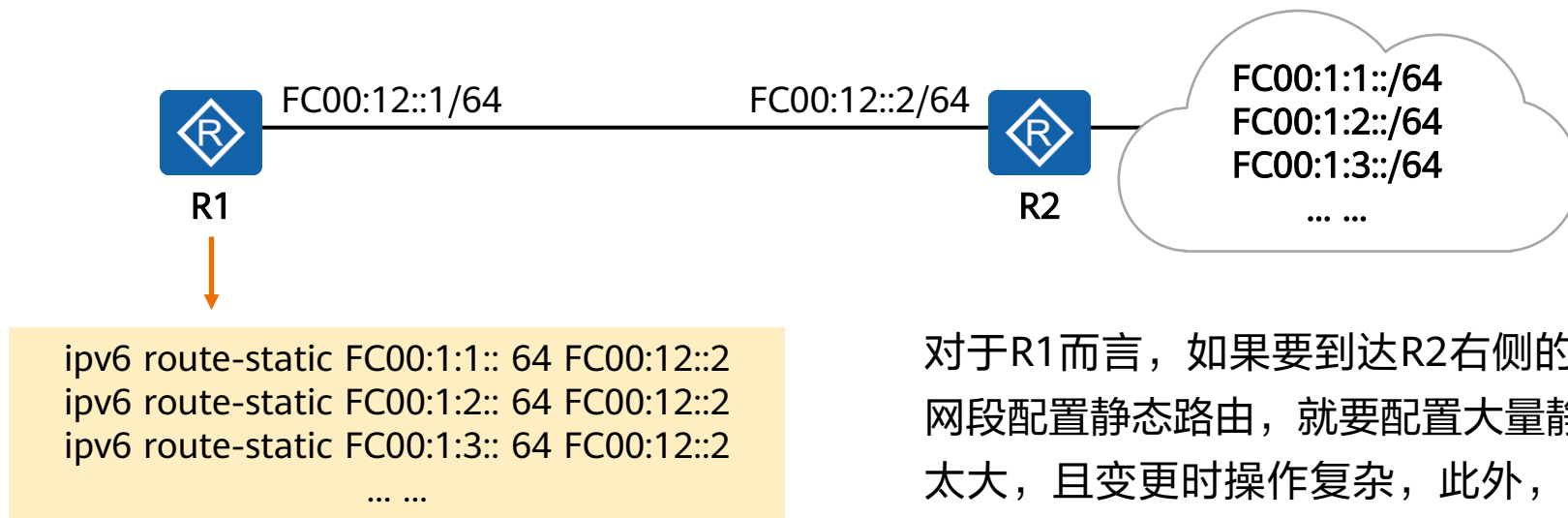
# 目录

---

1. 路由的基本概念
2. 静态路由基础
3. **静态路由进阶**
  - 浮动路由
  - **路由汇总**
  - 路由迭代
  - 静态路由与BFD
  - 静态路由与NQA

# 路由汇总：技术背景

- 一个网络如果具备科学的IP编址，并且进行合理的规划，是可以利用多种手段减小设备路由表规模的。
- 常见而又有效的办法就是使用路由汇总。

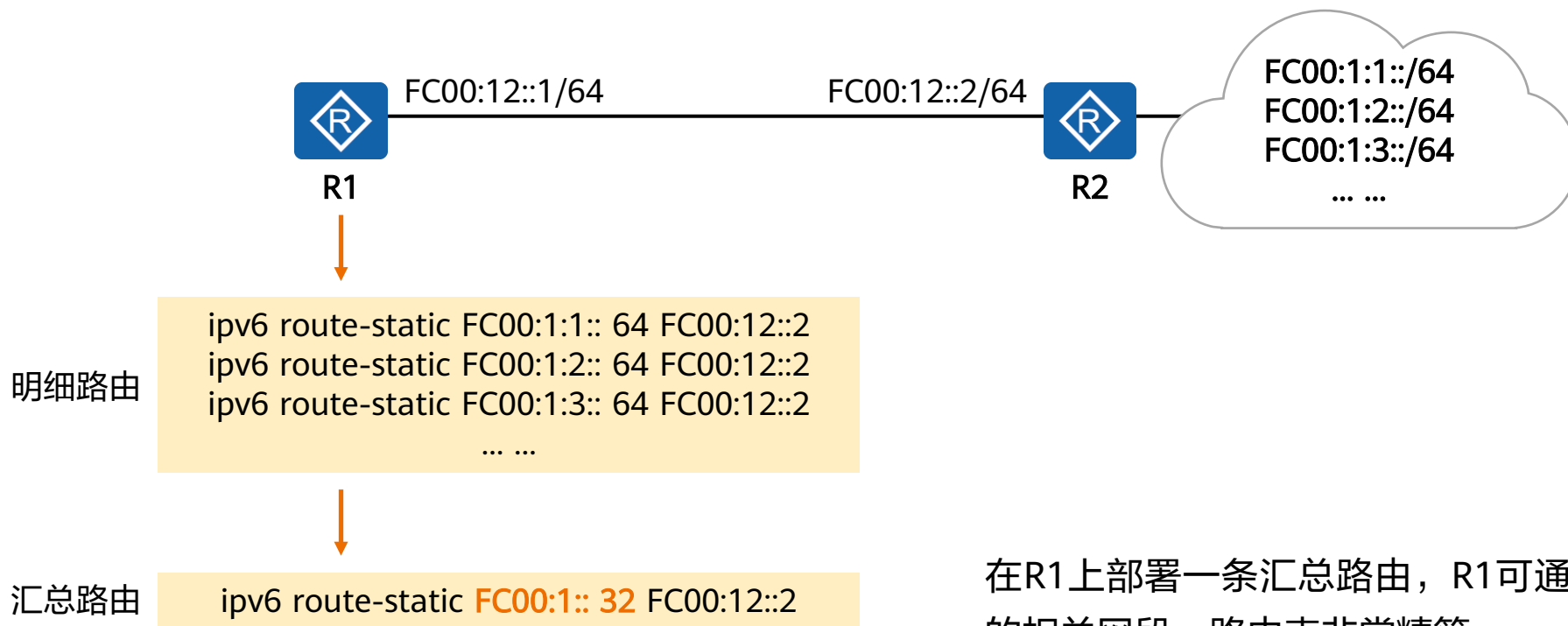


对于R1而言，如果要到达R2右侧的网段，若手工为每个网段配置静态路由，就要配置大量静态路由，配置工作量太大，且变更时操作复杂，此外，R1的路由表也将变得非常臃肿。



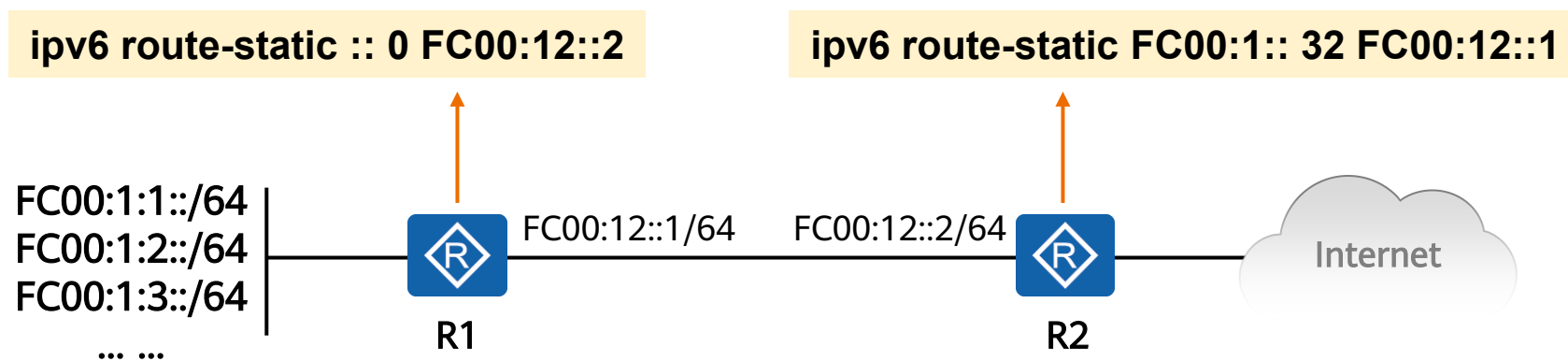
# 路由汇总的基本概念

- 路由汇总又被称为路由聚合，即将一组有规律的路由汇聚成一条路由，从而达到减小路由表规模以及优化设备资源利用率的目的。汇聚之前的路由是精细路由或明细路由，汇聚之后的路由是汇总路由或聚合路由。



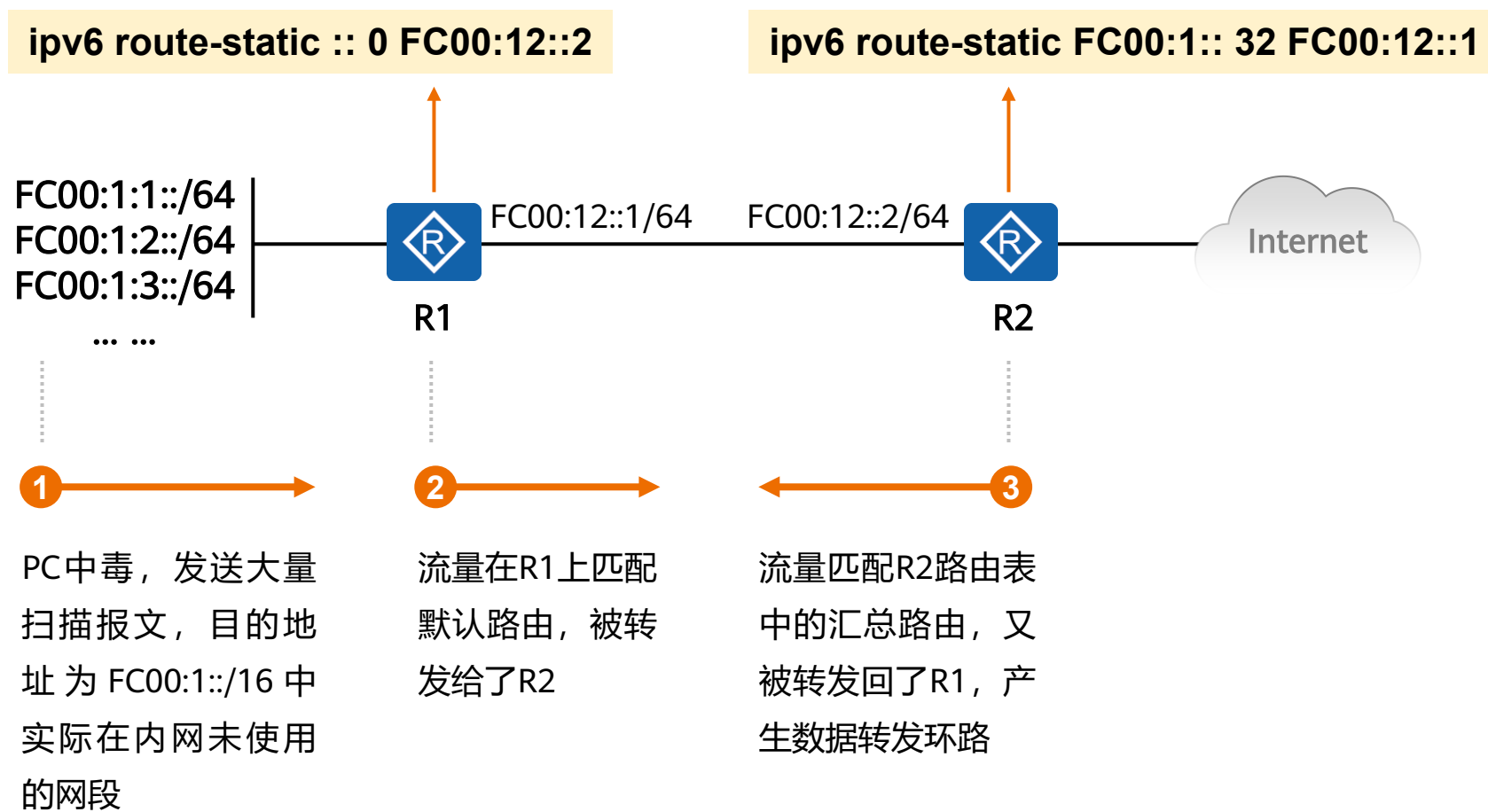
在R1上部署一条汇总路由，R1可通过该路由到达R2右侧的相关网段，路由表非常精简。

# 汇总路由存在的问题：技术背景

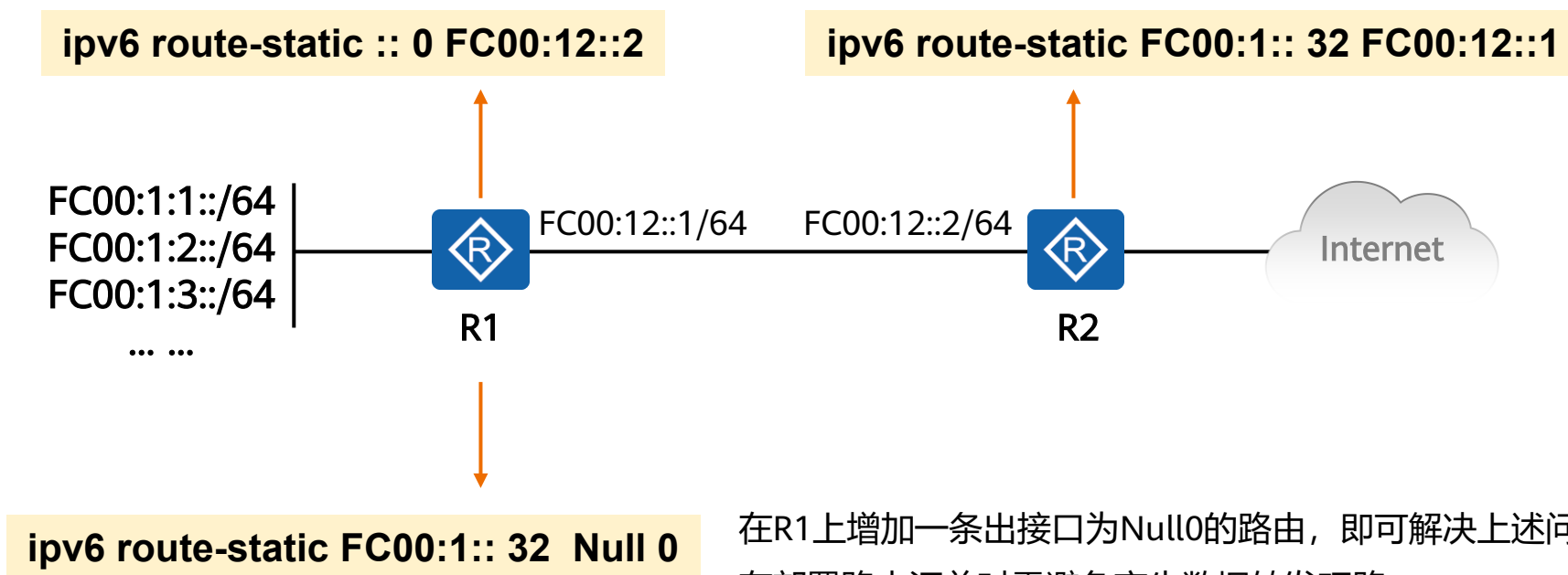


路由汇总是一个非常重要的网络优化思想，然而如果使用不当，也有可能带来问题，在上图中，R1左侧有一系列网络，为了让它们能够访问Internet，R1配置了指向R2的默认路由。而为了让回程数据能够顺利从R2到达R1，又为了精简路由表，于是配置了一条汇总路由并指向R1。这个网络看似没什么问题，但.....

# 汇总路由存在的问题：引发数据环路



# 汇总路由存在的问题：如何解决



在R1上增加一条出接口为Null0的路由，即可解决上述问题。  
在部署路由汇总时需避免产生数据转发环路。

# 关于NULL0接口与黑洞路由

- 系统会自动创建一个NULL0接口（可以简单理解为黑洞接口）。
- NULL0接口一直处于UP状态，但是不能转发数据包，任何发送到该接口的网络数据报文都会被丢弃。
- 如果在静态路由中指定到达某一网段的下一跳为NULL0接口，则任何发送到该网段的数据报文都会被丢弃，因此可以将需要过滤掉的报文直接发送到NULL0接口而不必配置访问控制列表。
- NULL接口只能有一个，固定为NULL0，该接口始终处于Up状态，不能被关闭或删除。
- 黑洞路由指的是以NULL0为出接口的路由。
- 在部署了路由汇总的网络中，可以使用黑洞路由来防止出现数据转发环路。
- 在部署了NAT的网络中，用于防止出现数据转发环路。
- 在BGP网络中，偶用于发布到达特定网段的路由。

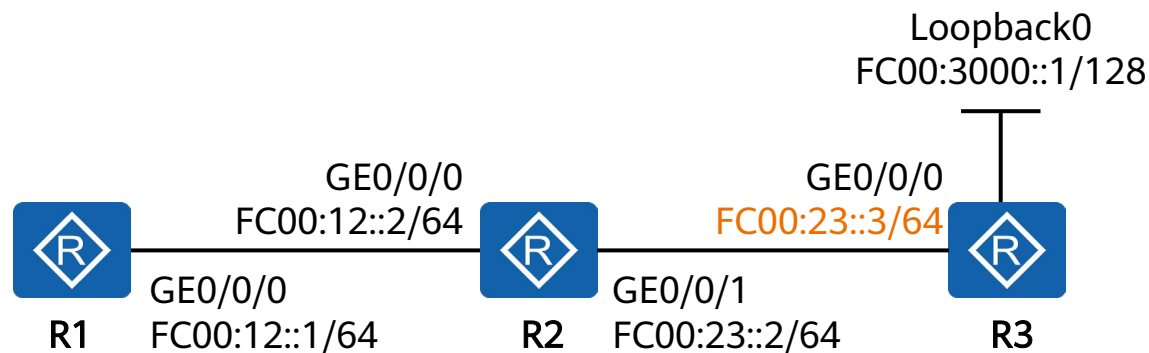
# 目录

---

1. 路由的基本概念
2. 静态路由基础
3. **静态路由进阶**
  - 浮动路由
  - 路由汇总
  - **路由迭代**
  - 静态路由与BFD
  - 静态路由与NQA

# 什么是路由迭代？

- 路由的下一跳“最终”必须关联到本地出接口，以及下一跳IP地址，而这个下一跳IP地址必须处于本地直连网络，否则该路由就视为无效。



R1的配置如下：

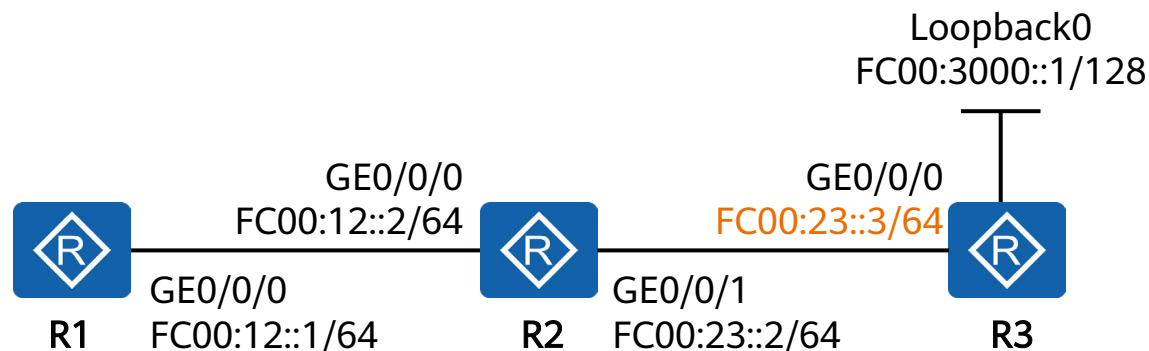
```
[R1] ipv6 route-static FC00:3000::1 128 FC00:23::3
```

对于R1而言，FC00:23::3并非处于本地直连网络，而且R1的路由表中又没有到达FC00:23::3的路由，因此使用FC00:23::3作为下一跳的FC00:3000::1/128路由被视为无效。



# 路由迭代概述

- 路由必须有直连的下一跳才能够指导转发，但是路由生成时下一跳可能不是直连的，因此需要计算出一个直连的下一跳和对应的出接口，这个过程就叫做路由迭代。



R1的配置如下：

```
[R1] ipv6 route-static FC00:3000::1 128 FC00:23::3
```

迭代

```
[R1] ipv6 route-static FC00:23:: 64 FC00:12::2
```

在本例中，为R1增加一条到达FC00:23::/64的路由，使得到达FC00:3000::1/128的路由能够完成递归。



# 路由迭代概述

```
[R1] display ipv6 routing-table
```

```
Routing Table : Public
```

```
Destinations : 8
```

```
Routes : 8
```

```
.....
```

|              |                        |              |          |
|--------------|------------------------|--------------|----------|
| Destination  | : FC00:23::            | PrefixLength | : 64     |
| NextHop      | : FC00:12::2           | Preference   | : 60     |
| Cost         | : 0                    | Protocol     | : Static |
| RelayNextHop | : ::                   | TunnelID     | : 0x0    |
| Interface    | : GigabitEthernet0/0/0 | Flags        | : RD     |

|              |                        |              |          |
|--------------|------------------------|--------------|----------|
| Destination  | : FC00:3000::1         | PrefixLength | : 128    |
| NextHop      | : FC00:23::3           | Preference   | : 60     |
| Cost         | : 0                    | Protocol     | : Static |
| RelayNextHop | : FC00:12::2           | TunnelID     | : 0x0    |
| Interface    | : GigabitEthernet0/0/0 | Flags        | : RD     |

```
.....
```

迭代后得到的直连下一跳

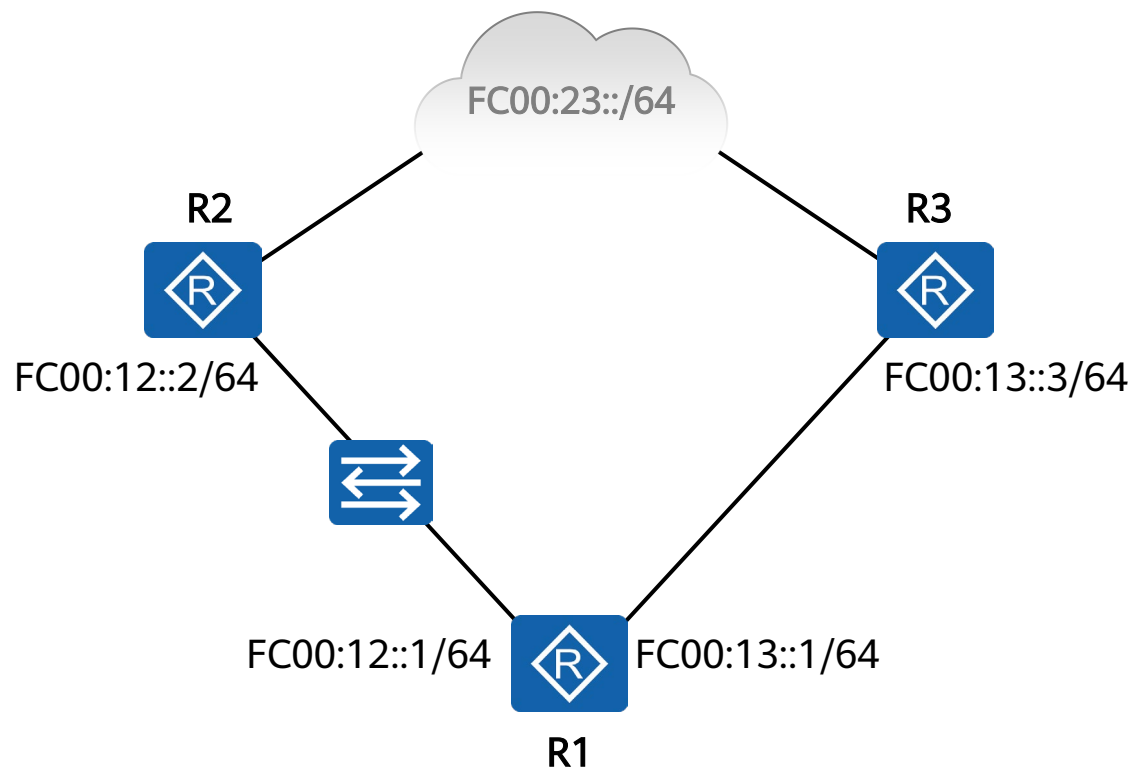
标志中的R表示Relay，即路由进行了迭代

# 目录

---

1. 路由的基本概念
2. 静态路由基础
3. **静态路由进阶**
  - 浮动路由
  - 路由汇总
  - 路由迭代
  - **静态路由与BFD**
  - 静态路由与NQA

# 静态路由与BFD联动的技术背景



## 项目需求:

- R1配置了如图所示的静态路由;
- 要求当网络正常时, R1收到发往FC00:23::/64的数据时, 将其转发给R2; 当R2本身或其直连链路发生故障时, R1将流量切换到R3。

## 问题描述:

- 当R2发生故障时, R1无法感知 (中间隔着一个交换机), 流量中断。

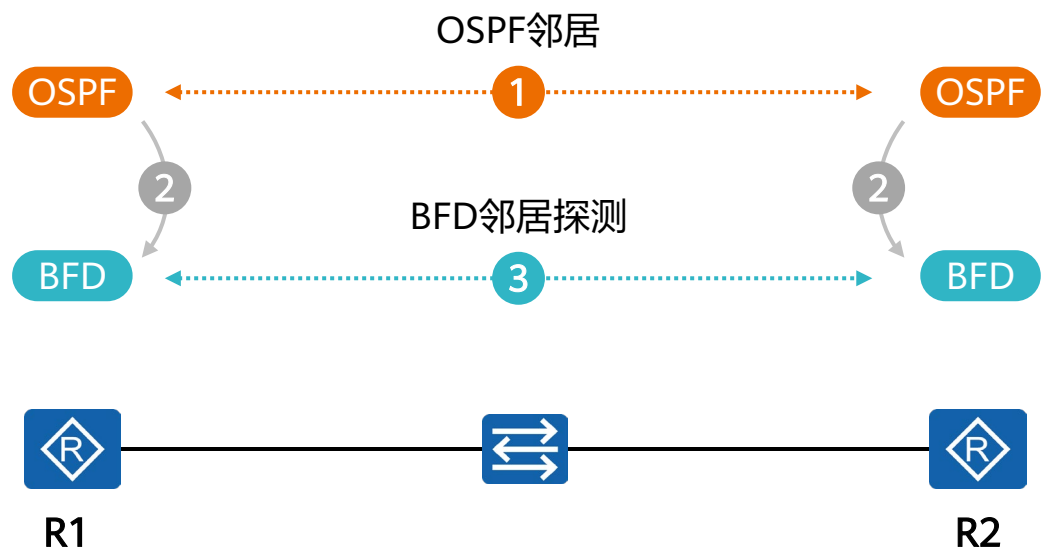
```
[R1] ipv6 route-static FC00:23:: 64 FC00:12::2  
[R1] ipv6 route-static FC00:23:: 64 FC00:13::3 preference 80
```

# BFD概述

- 双向转发检测BFD（ Bidirectional Forwarding Detection ）是一种检测机制，用于快速检测、监控网络中链路或者IP路由的转发连通状况。
- BFD可以实现快速检测并监控网络中链路或IP路由的转发连通状态，改善网络性能。
- 相邻系统之间通过快速检测发现通信故障，可以更快地帮助用户建立起备份通道以便恢复通信，保证网络可靠性。
- BFD在两台网络设备上建立会话，用来检测网络设备间的双向转发路径，为上层应用服务。BFD本身并没有邻居发现机制，而是靠被服务的上层应用通知其邻居信息以建立会话。会话建立后会周期性地快速发送BFD报文，如果在检测时间内没有收到BFD报文则认为该双向转发路径发生了故障，通知被服务的上层应用进行相应的处理。

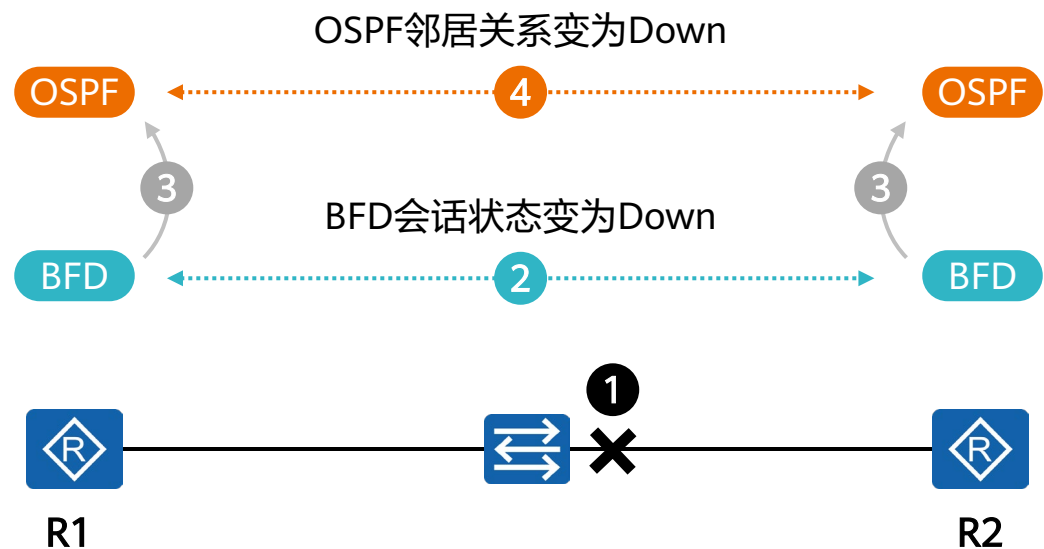
# BFD的基本原理

## BFD会话建立



1. OSPF发现邻居并建立连接，建立了新的邻居关系后，将邻居信息（包括目的地址和源地址等）通告给BFD。
2. BFD根据收到的邻居信息建立会话。
3. 会话建立以后，BFD开始检测链路故障，并做出快速反应。

## BFD故障发现



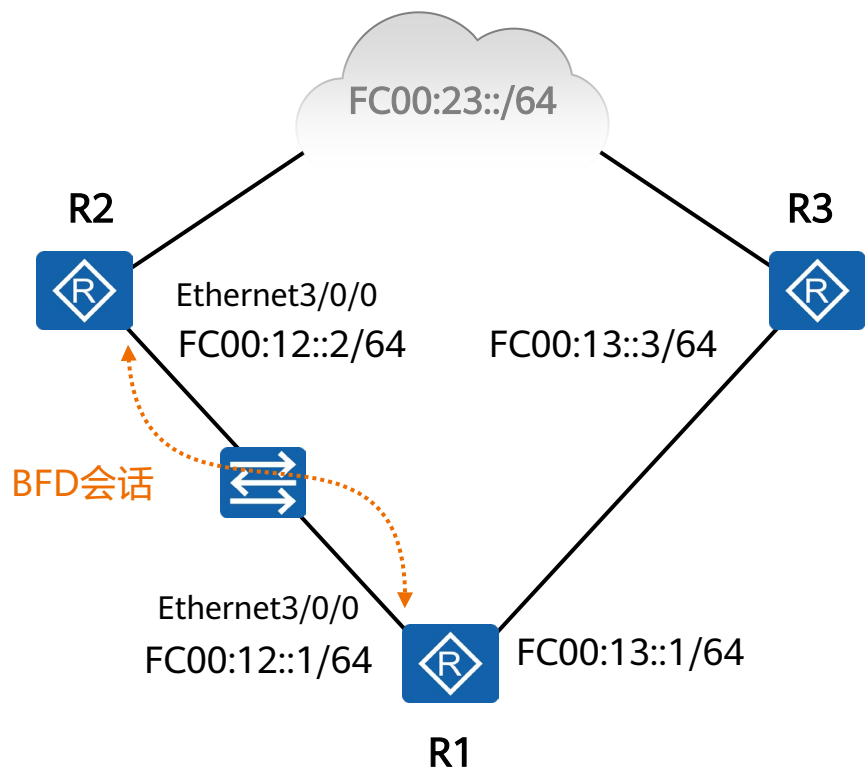
1. 被检测链路出现故障。
2. BFD快速检测到链路故障，BFD会话状态变为Down。
3. BFD通知本地OSPF进程BFD邻居不可达。
4. 本地OSPF进程中断OSPF邻居关系。



# BFD会话建立方式

- BFD会话的建立有两种方式，即静态建立BFD会话和动态建立BFD会话，主要区别在于本地标识符（ Local Discriminator ）和远端标识符（ Remote Discriminator ）的配置方式不同。
  - BFD通过控制报文中的Local Discriminator和Remote Discriminator区分不同的会话。
- **静态建立BFD会话：**静态建立BFD会话是指通过命令行手工配置BFD会话参数，包括配置本地标识符和远端标识符等，然后手工下发BFD会话建立请求。
- **动态建立BFD会话：**动态建立BFD会话时，系统对本地标识符和远端标识符的处理方式如下：
  - **动态分配本地标识符：**当应用程序触发动态创建BFD会话时，系统分配属于动态会话标识符区域的值作为BFD会话的本地标识符。然后向对端发送Remote Discriminator的值为0的BFD控制报文，进行会话协商。
  - **自学习远端标识符：**当BFD会话的一端收到Remote Discriminator的值为0的BFD控制报文时，判断该报文是否与本地BFD会话匹配，如果匹配，则学习接收到的BFD报文中Local Discriminator的值，获取远端标识符。

# BFD与静态路由联动



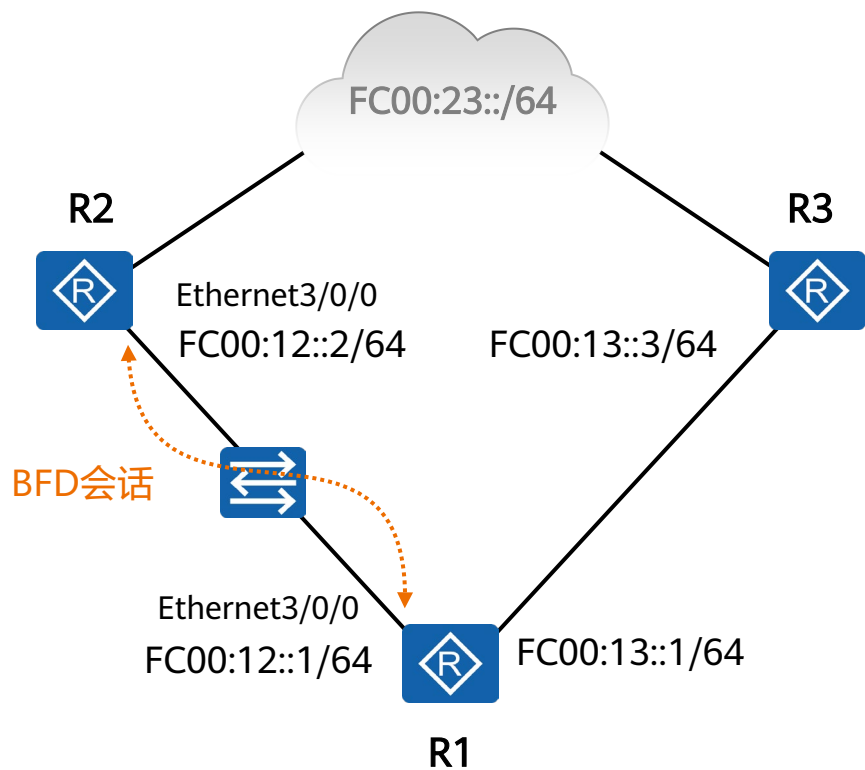
将R2指定为R1到达FC00:23::/64的主下一跳，当R2不可达时，切换至R3。

1. 完成路由器的接口IPv6配置。
2. 在R1上配置BFD，并将静态路由关联BFD。

```
[R1] bfd
[R1-bfd] quit
[R1] bfd R1-R2 bind peer-ipv6 FC00:12::2 interface Ethernet 3/0/0
[R1-bfd-session-R1-R2] discriminator local 1
[R1-bfd-session-R1-R2] discriminator remote 2
[R1-bfd-session-R1-R2] commit
[R1-bfd-session-R1-R2] quit

[R1] ipv6 route-static FC00:23:: 64 FC00:12::2 track bfd-session R1-R2
[R1] ipv6 route-static FC00:23:: 64 FC00:13::3 preference 80
```

# BFD与静态路由联动（续）



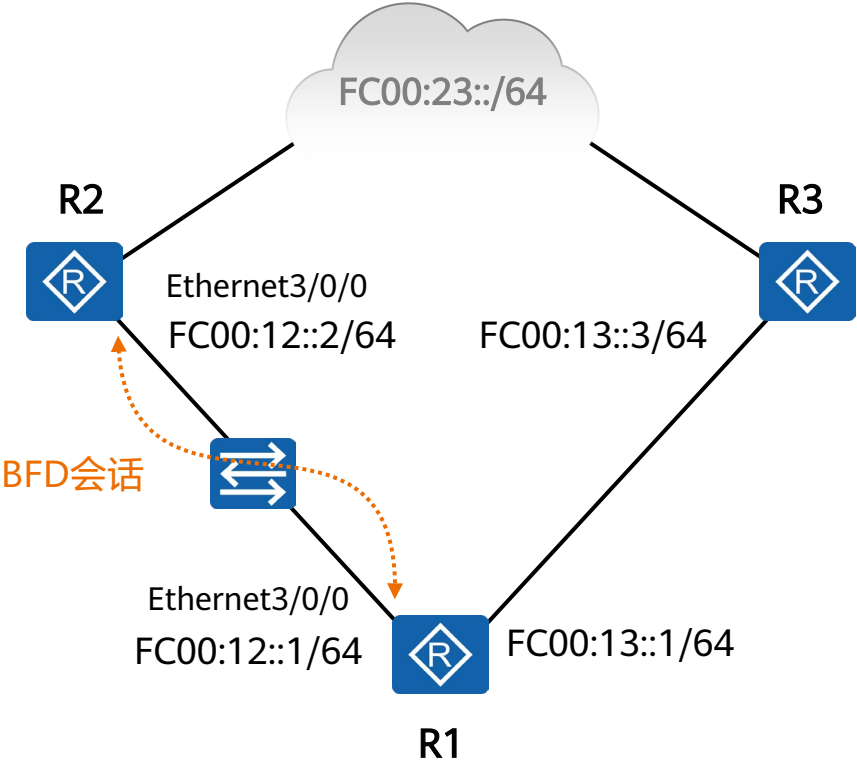
3. 在R2上配置BFD。

```
[R2] bfd
[R2-bfd] quit
[R2] bfd R2-R1 bind peer-ipv6 FC00:12::1 interface Ethernet 3/0/0
[R2-bfd-session-R2-R1] discriminator local 2
[R2-bfd-session-R2-R1] discriminator remote 1
[R2-bfd-session-R2-R1] commit
[R2-bfd-session-R2-R1] quit
```

将R2指定为R1到达FC00:23::/64的主下一跳，当R2不可达时，切换至R3。在R1与R2之间部署BFD会话，并将主路由与BFD联动。



# BFD与静态路由联动（续）



将R2指定为R1到达FC00:23::/64的主下一跳，当R2不可达时，切换至R3。在R1与R2之间部署BFD会话，并将主路由与BFD联动。

在R1上查看BFD：

[R1] display bfd session all

(w): State in WTR

(\*): State is invalid

| Local | Remote | PeerIpAddr | State | Type    | InterfaceName |
|-------|--------|------------|-------|---------|---------------|
| 1     | 2      | FC00:12::2 | Up    | S_IP_IF | Ethernet3/0/0 |

Total UP/DOWN Session Number : 1/0

当R1到R2不可达时，BFD会话状态变为Down，此时主路由失效，浮动路由显现在R1的路由表中。

# 目录

---

1. 路由的基本概念
2. 静态路由基础
- 3. 静态路由进阶**
  - 浮动路由
  - 路由汇总
  - 路由迭代
  - 静态路由与BFD
  - 静态路由与NQA**

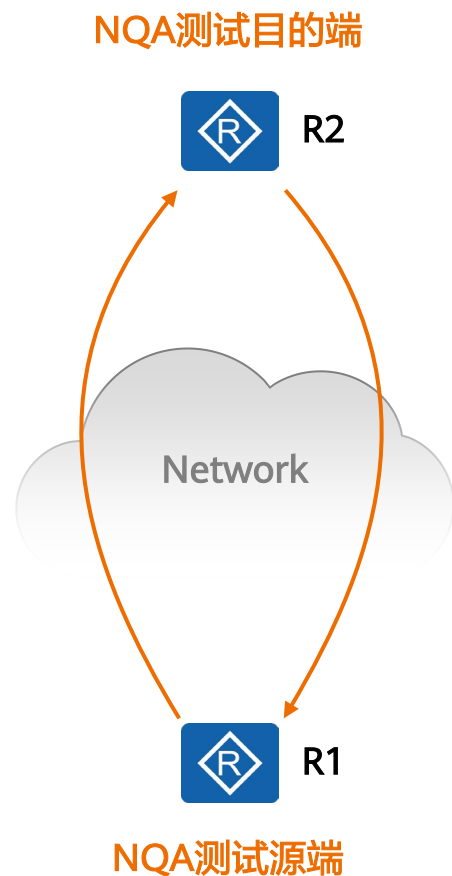
# NQA简介

- 网络质量分析NQA（ Network Quality Analysis ）是一种实时的网络性能探测和统计技术，可以对响应时间、网络抖动、丢包率等网络信息进行统计。
- NQA可以实现对网络运行状况的准确测试，输出统计信息。
- NQA监测网络上运行的多种协议的性能，使用户能够实时采集到各种网络运行指标，例如：
  - HTTP的总时延
  - TCP连接时延
  - DNS解析时延
  - 文件传输速率
  - FTP连接时延
  - DNS解析错误率等。

# NQA原理描述

- **构造测试例：**NQA测试中，NQA的测试是由客户端（源端）发起。在客户端通过命令行配置测试例或由网管端发送相应测试例操作后，NQA把相应的测试例放入到测试例队列中进行调度。
- **启动测试例：**启动NQA测试例，可以选择立即启动、延迟启动、定时启动。在定时器的时间到达后，则根据测试例的测试类型，构造符合相应协议的报文。但配置的测试报文的大小如果无法满足发送本协议报文的最小尺寸，则按照本协议规定的最小报文尺寸来构造报文发送。
- **测试例处理：**测试例启动后，根据返回的报文，可以对相关协议的运行状态提供数据信息。发送报文时的系统时间作为测试报文的发送时间，给报文打上时间戳，再发送给服务器端。服务器端接收报文后，返回给客户端相应的回应信息，客户端在接收到报文时，再一次读取系统时间，给报文打上时间戳。根据报文的发送和接收时间，计算出报文的往返时间。

# 基于ICMP的NQA



NQA的ICMP测试例用于检测源端到目的端的路由是否可达。ICMP测试提供类似于命令行下的Ping命令功能，但输出信息更为丰富：

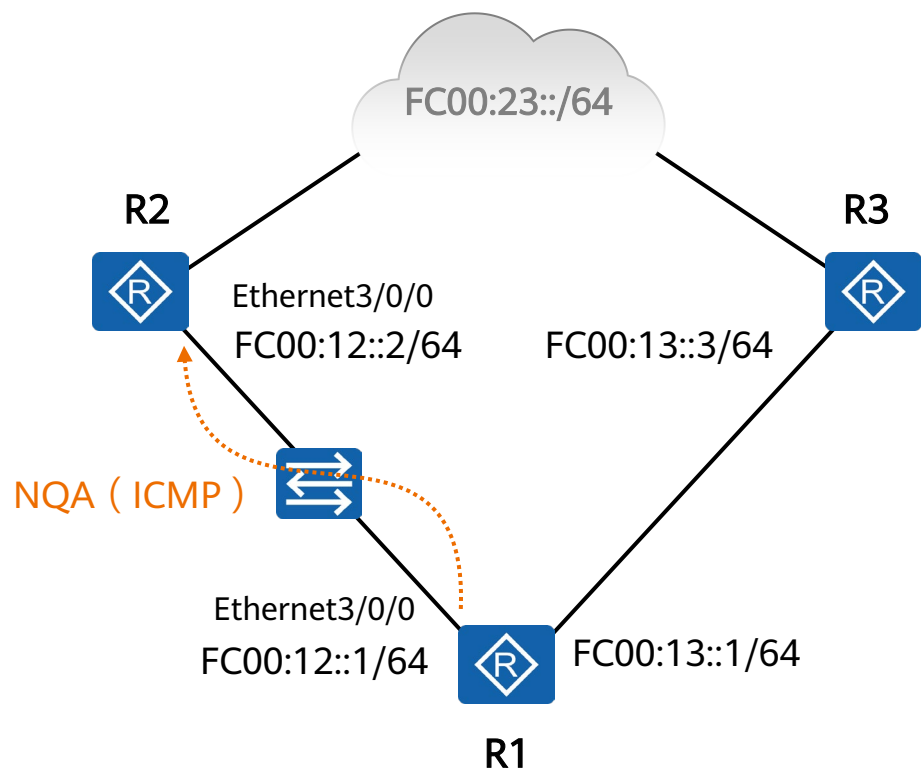
- 默认情况下能够保存最近5次的测试结果。
- 结果中能够显示平均时延、丢包率，最后一个报文正确接收的时间等信息。

ICMP测试的过程如下：

1. 源端向目的端发送构造的ICMP Echo Request报文。
2. 目的端收到报文后，直接回应ICMP Echo Reply报文给源端。

源端收到报文后，通过计算源端接收时间和源端发送时间之差，计算出源端到目的端的通信时间，从而清晰的反应出网络性能及网络畅通情况。

# NQA与静态路由联动



将R2指定为R1到达FC00:23::/64的主下一跳，当R2不可达时，切换至R3。在R1上部署到R2的NQA测试用例，并将NQA与主路由联动。

1. 完成路由器的接口IPv6配置。
2. 在R1上配置NQA，并将静态路由关联NQA测试用例。

```
[R1] nqa test-instance admin icmp
[R1-nqa-admin-icmp] test-type icmp
[R1-nqa-admin-icmp] destination-address ipv6 FC00:12::2
[R1-nqa-admin-icmp] frequency 6
[R1-nqa-admin-icmp] probe-count 2
[R1-nqa-admin-icmp] interval seconds 2
[R1-nqa-admin-icmp] timeout 2
[R1-nqa-admin-icmp] start now

[R1] ipv6 route-static FC00:23:: 64 FC00:12::2 track nqa admin icmp
[R1] ipv6 route-static FC00:23:: 64 FC00:13::3 preference 80
```

# 思考题

---

- 路由器如何优选路由条目？
- 如何配置实现浮动路由？

# 课程总结

---

- 在本课程中，我们学习了路由的基本概念，了解了路由如何指导路由器对IP报文进行转发，同时还了解了常见的路由属性、路由表中路由信息的来源。
- 路由查询的最长前缀匹配原则是路由器执行路由查询的基本规则。
- 静态路由适用于组网固定、网络规模较小的场景，而通过联动BFD或NQA测试用例，亦可在网络拓扑出现变化时使得静态路由做出及时响应。



# Thank you.

把数字世界带入每个人、每个家庭、  
每个组织，构建万物互联的智能世界。

Bring digital to every person, home and  
organization for a fully connected,  
intelligent world.

Copyright©2018 Huawei Technologies Co., Ltd.  
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

