路由策略与路由控制

前言

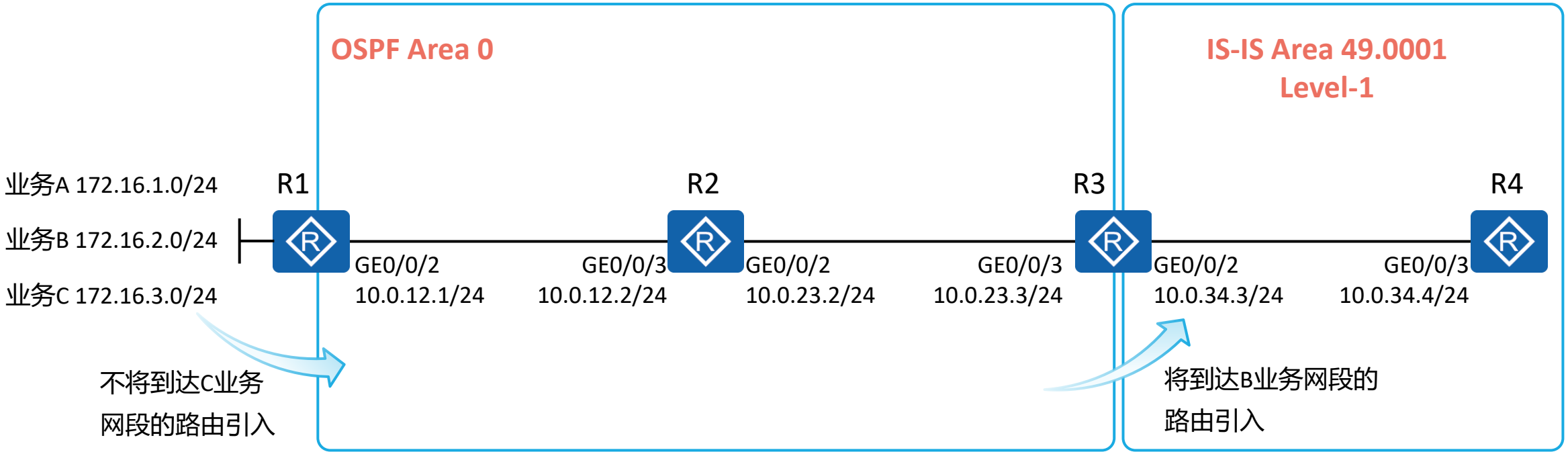
- 在复杂的数据通信网络中，根据实际组网需求，往往需要实施一些路由策略对路由信息进行过滤、属性设置等操作，通过对路由的控制，可以影响数据流量转发。
- 路由策略并非单一的技术或者协议，而是一个技术专题或方法论，里面包含了多种工具及方法。
- 本课程主要介绍网络中常用的路由选择工具以及路由策略的原理与配置。

目标

- 学完本课程后，您将能够：
  - 使用ACL匹配感兴趣路由
  - 使用IP-Prefix匹配感兴趣路由
  - 使用Filter-Policy进行路由过滤
  - 使用Route-Policy进行路由过滤及路由属性修改

目录

技术背景

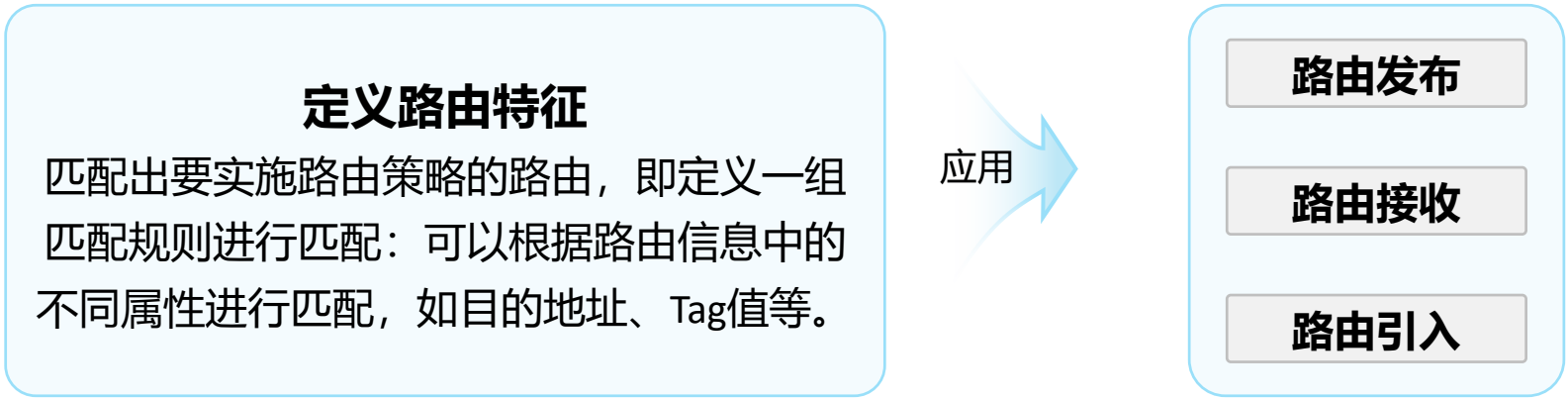


- R1上将业务A、B、C的网段路由引入时希望不引入业务C网段路由，同时R3上将OSPF路由引入到IS-IS中时，同样只希望引入B业务网段路由。
- 此时需要一个工具在引入路由时进行限制。

路由控制概述

路由控制可以通过路由策略（Route-Policy）实现，路由策略应用灵活而广泛，有以下几种常见方式：

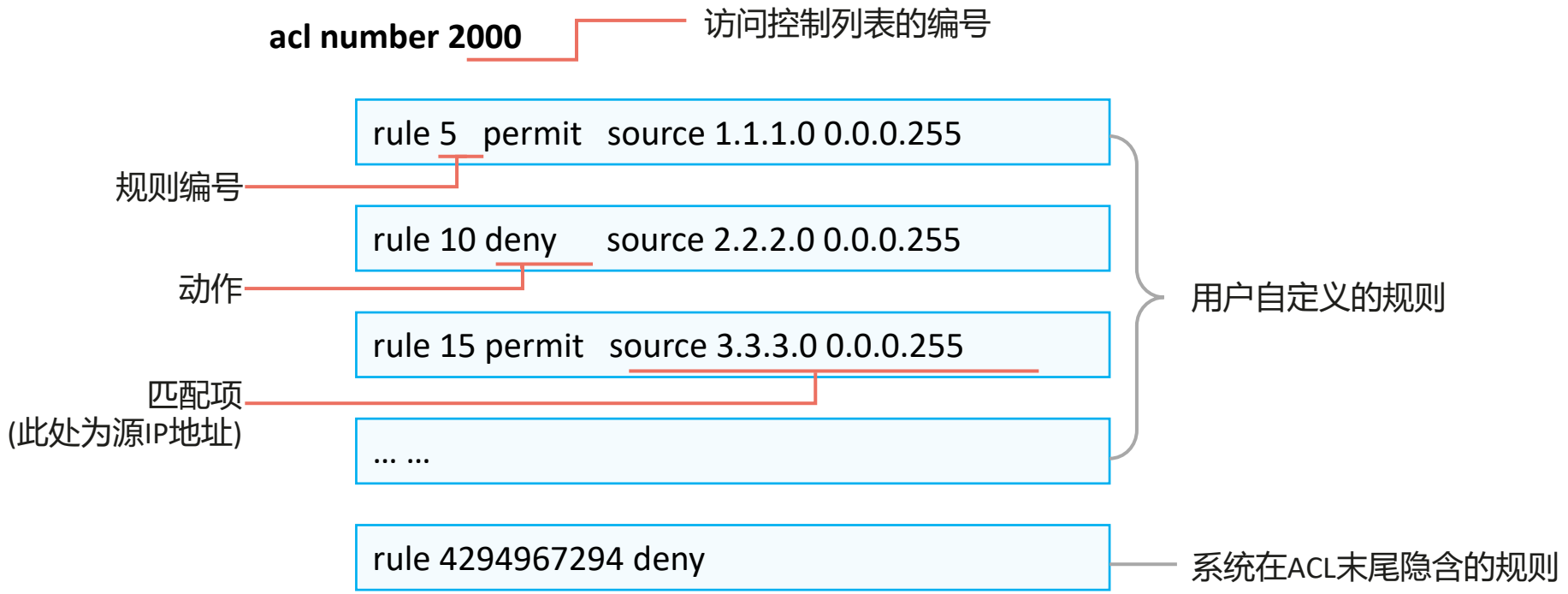
- 控制路由的发布：通过路由策略对发布的路由进行过滤，只发布满足条件的路由。
- 控制路由的接收：通过路由策略对接收的路由进行过滤，只接收满足条件的路由。
- 控制路由的引入：通过路由策略控制从其他路由协议引入的路由条目，只有满足条件的路由才会被引入。



目录

匹配工具1：访问控制列表

- 访问控制列表（Access Control List，ACL）是一个匹配工具，能够对报文及路由进行匹配和区分。
- ACL由若干条permit或deny语句组成。每条语句就是该ACL的一条规则，每条语句中的permit或deny就是与这条规则相对应的处理动作。



通配符

```
acl number 2000
                                                    通配符
 rule      5      deny      source  10.1.1.1  0
 rule      10     deny      source  10.1.1.2  0
 rule      15     permit    source  10.1.1.0  0.0.0.255
```

通配符 (Wildcard)

- 通配符是一个32比特长度的数值，用于指示IP地址中哪些比特位需要严格匹配，哪些比特位无需匹配。
- 通配符通常采用类似网络掩码的点分十进制形式表示，但是含义却与网络掩码完全不同。

- 匹配规则：

  “0”表示“匹配”；“1”表示“无需匹配”

如何匹配192.168.1.0/24网段内的IP地址?

192.168.1.1	11000000	10101000	00000000	00000001
0.0.0.255	00000000	00000000	00000000	11111111
	严格匹配			无需匹配

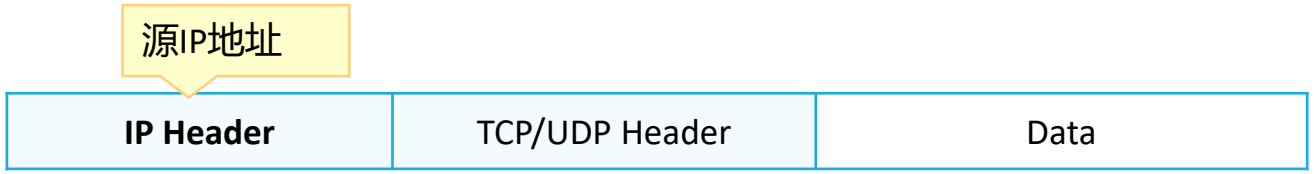
192.168.1.0/24网段

ACL的分类与基本ACL

- 基于ACL规则定义方式的划分

分类	编号范围	规则定义描述
基本ACL	2000~2999	仅使用报文的源IP地址、分片信息和生效时间段信息来定义规则。
高级ACL	3000~3999	可使用IPv4报文的源IP地址、目的IP地址、IP协议类型、ICMP类型、TCP源/目的端口、UDP源/目的端口号、生效时间段等来定义规则。
二层ACL	4000~4999	使用报文的以太网帧头信息来定义规则，如根据源MAC地址、目的MAC地址、二层协议类型等。
用户自定义ACL	5000~5999	使用报文头、偏移位置、字符串掩码和用户自定义字符串来定义规则。
用户ACL	6000~6999	既可使用IPv4报文的源IP地址或源UCL（User Control List）组，也可使用目的IP地址或目的UCL组、IP协议类型、ICMP类型、TCP源端口/目的端口、UDP源端口/目的端口号等来定义规则。

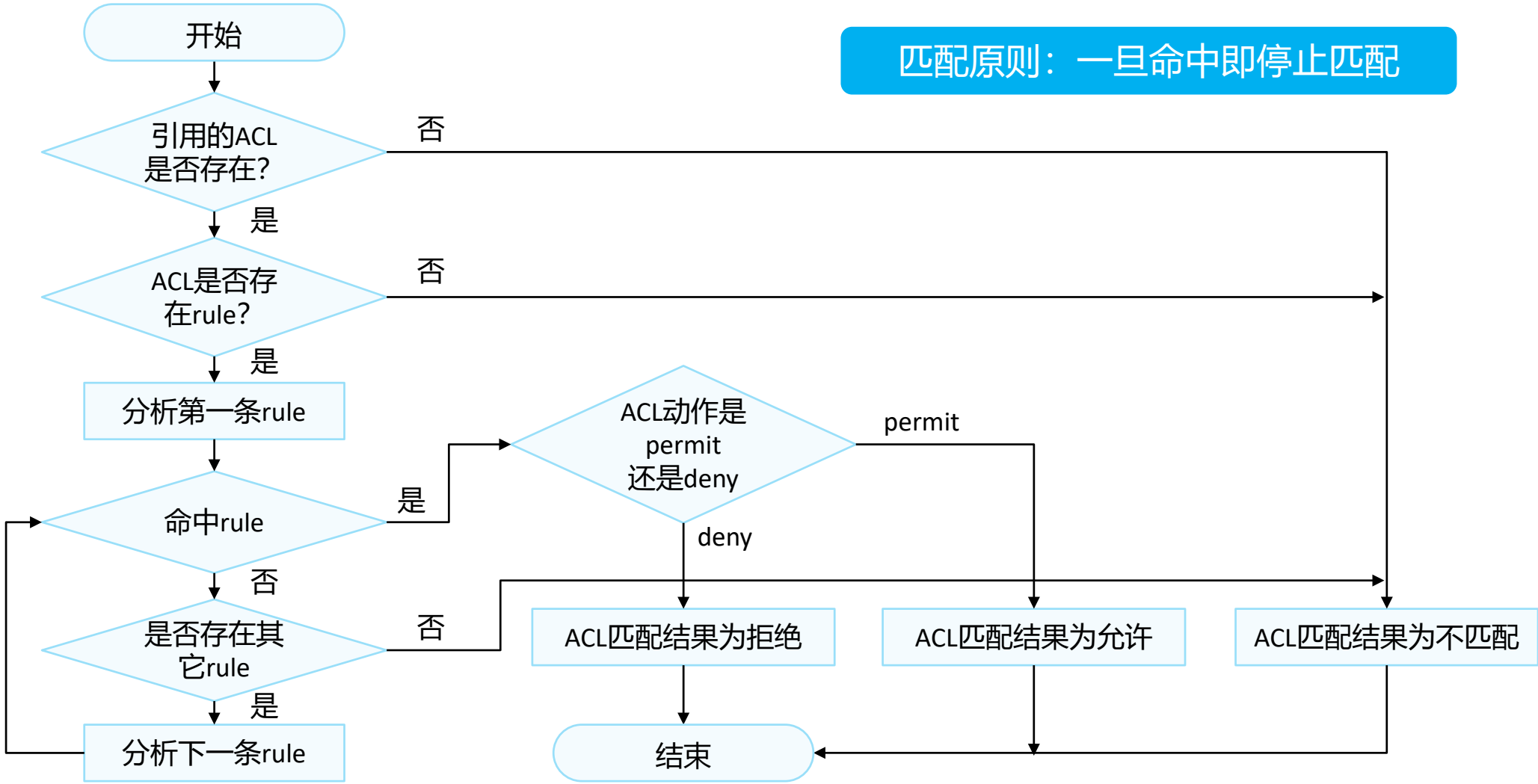
- 基本ACL




```
acl number 2000
 rule      5      deny      source  10.1.1.1  0
 rule      10     deny      source  10.1.1.2  0
 rule      15     permit    source  10.1.1.0  0.0.0.255
```

ACL的匹配机制

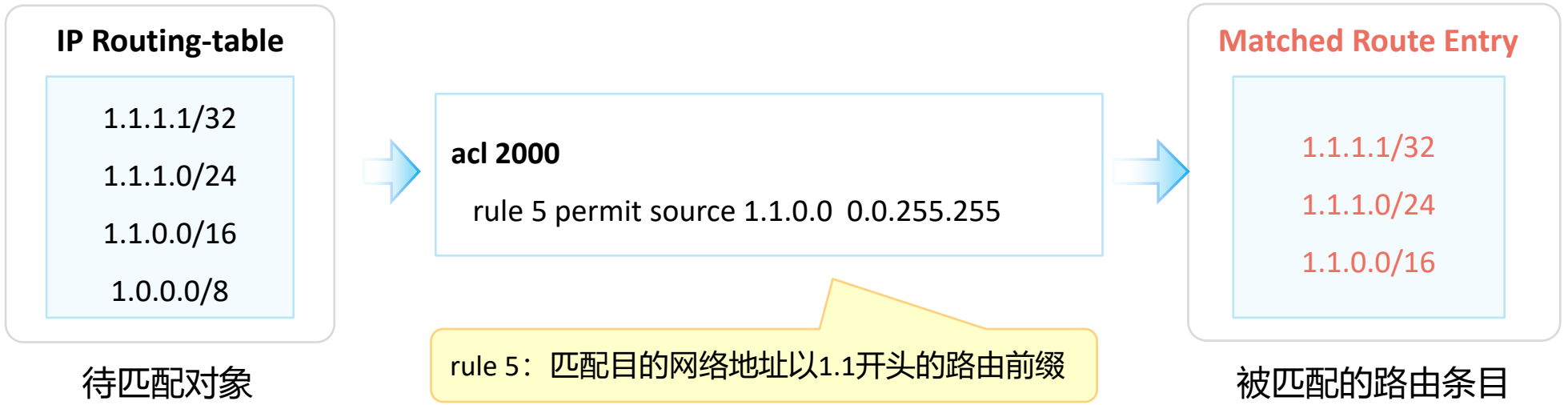
匹配原则：一旦命中即停止匹配



ACL的匹配顺序及匹配结果

配置顺序（config模式）

- 系统按照ACL规则编号从小到大的顺序进行报文匹配，规则编号越小越容易被匹配。



“允许”是指什么？

常用匹配举例

ACL只能匹配路由的前缀，无法匹配路由的网络掩码。

基本ACL的基础配置命令

1. 创建基本ACL

[Huawei] **acl** [**number**] *acl-number* [**match-order config**]

使用编号（2000 ~ 2999）创建一个数字型的基本ACL，并进入基本ACL视图。

[Huawei] **acl name** *acl-name* { **basic** | *acl-number* } [**match-order config**]

使用名称创建一个命名型的基本ACL，并进入基本ACL视图。

2. 配置基本ACL的规则

[Huawei-acl-basic-2000] **rule** [*rule-id*] { **deny** | **permit** } [**source** { *source-address source-wildcard* | **any** } | **time-range** *time-name*]

在基本ACL视图下，通过此命令来配置基本ACL的规则。

匹配工具2：IP前缀列表

- IP前缀列表（IP-Prefix List）是将路由条目的网络地址、掩码长度作为匹配条件的过滤器，可在各路由协议发布和接收路由时使用。
- 不同于ACL，IP-Prefix List能够同时匹配IP地址前缀长度以及掩码长度，增强了匹配的精确度。

[Huawei] **ip ip-prefix** test **index** 10 **permit** 192.168.1.0 22 **greater-equal** 24 **less-equal** 26

ip-prefix-name　序号　动作　IP网段与掩码　掩码范围

1、**ip-prefix-name**：地址前缀列表名称

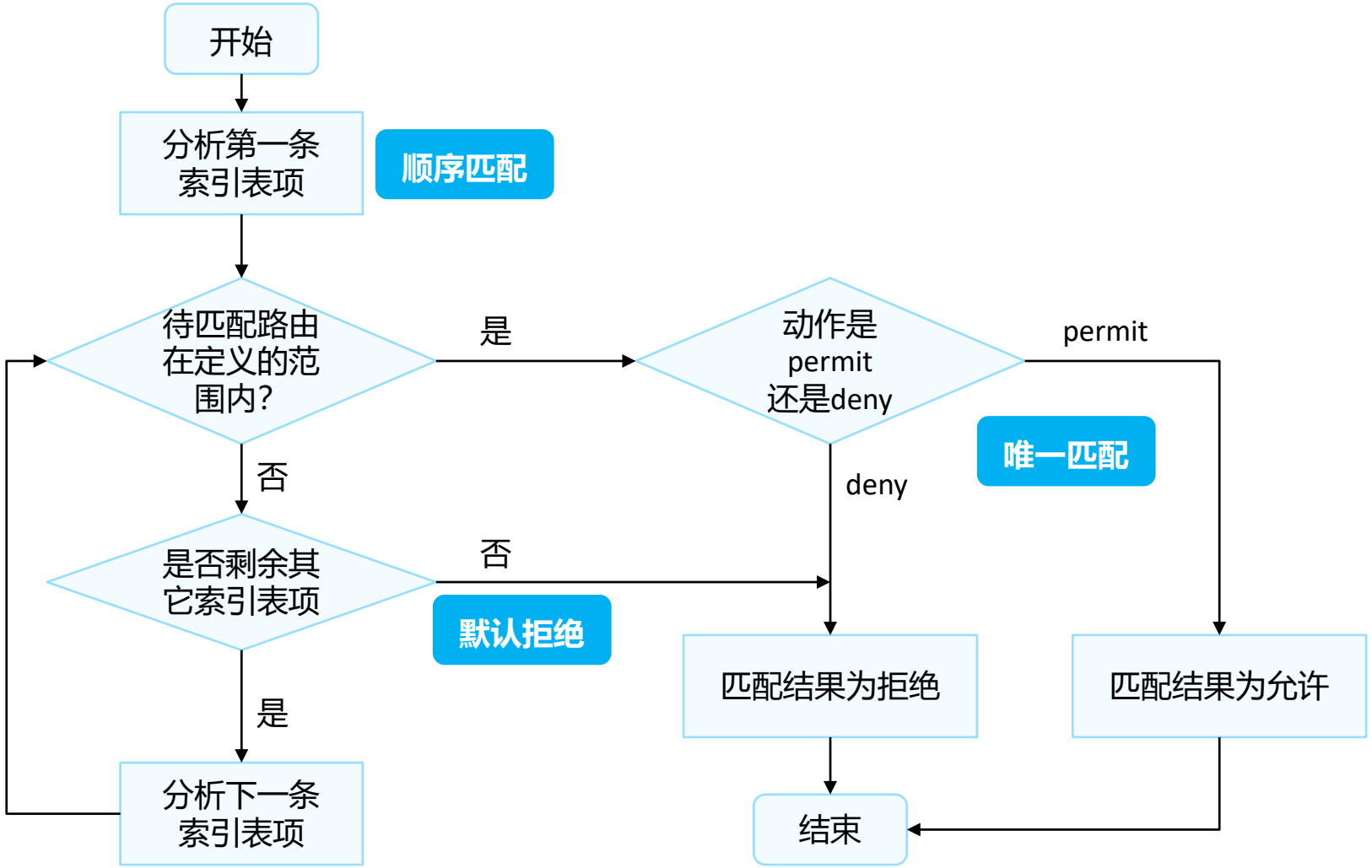
2、**序号**：本匹配项在地址前缀列表中的序号，匹配时根据序号从小到大进行顺序匹配

3、**动作**：permit/deny，地址前缀列表的匹配模式为允许/拒绝，表示匹配/不匹配

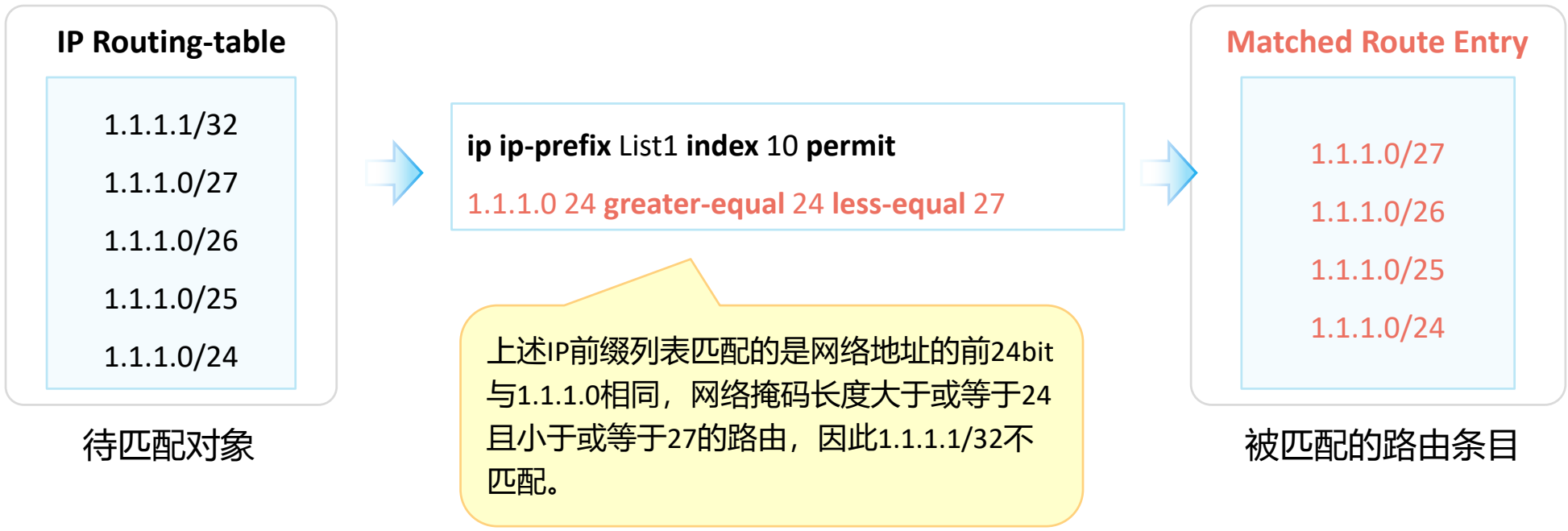
4、**IP网段与掩码**：匹配路由的网络地址，以及限定网络地址的前多少位需严格匹配

5、**掩码范围**：匹配路由前缀长度，掩码长度的匹配范围 mask-length<=greater-equal-value<=less-equal-value<=32

IP-Prefix的匹配机制



IP-Prefix的匹配示例



IP-Prefix的基础配置命令

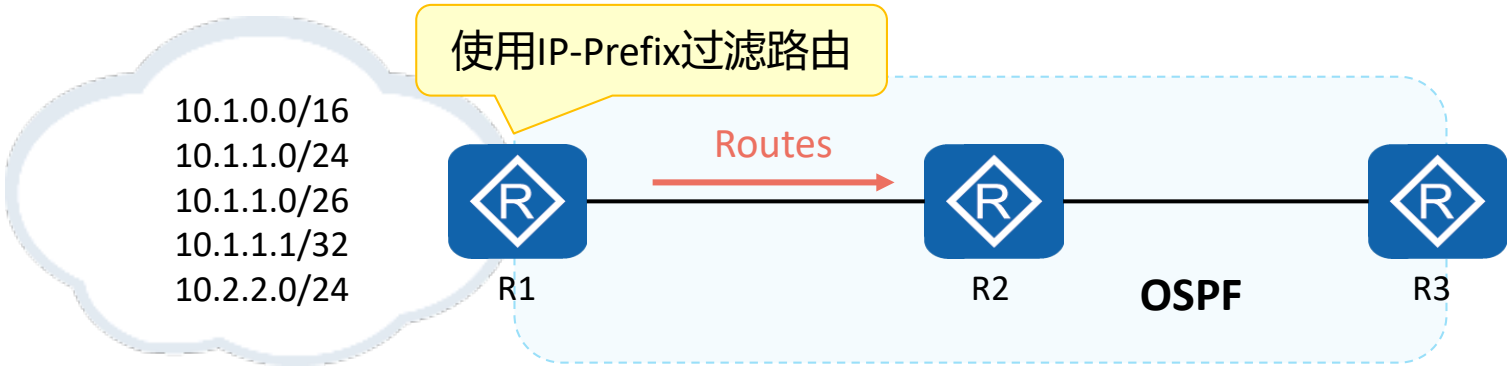
1. 创建IPv4地址前缀列表

[Huawei] **ip ip-prefix** *ip-prefix-name* [**index** *index-number*] { **permit** | **deny** } *ipv4-address mask-length* [**match-network**] [**greater-equal** *greater-equal-value*] [**less-equal** *less-equal-value*]

创建IPv4地址前缀列表或增加其中一个表项。

- *ip-prefix-name*：指定地址前缀列表的名称。
- **index** *index-number*：指定本匹配项在地址前缀列表中的序号。
- **permit**：指定地址前缀列表的匹配模式为允许。
- **deny**：指定地址前缀列表的匹配模式为拒绝。
- *ipv4-address mask-length*：指定IP地址和指定掩码长度。
- **greater-equal** *greater-equal-value*：指定掩码长度匹配范围的下限。
- **less-equal** *less-equal-value*：指定掩码长度匹配范围的上限。

IP-Prefix的配置举例 (1)



单语句匹配

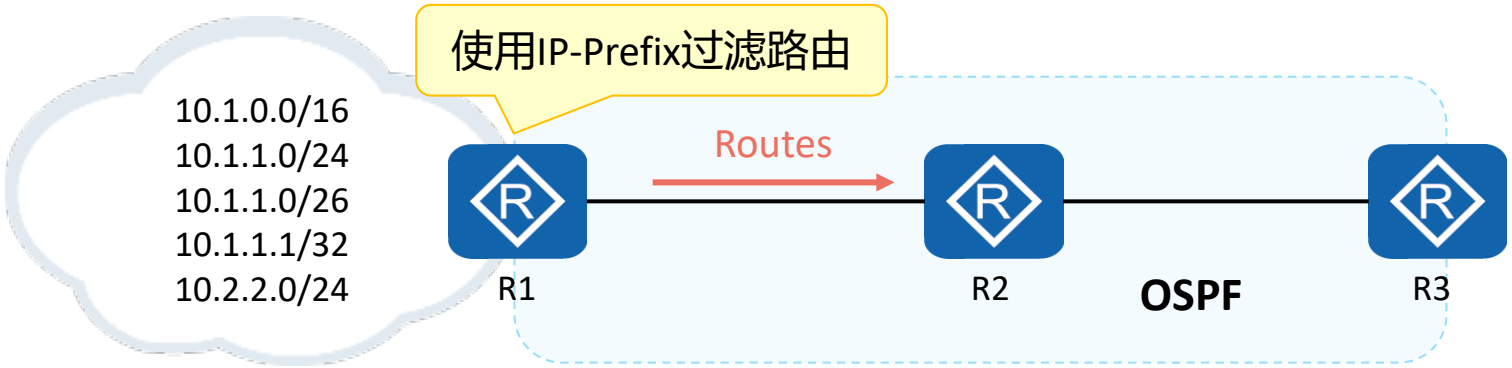
```
ip ip-prefix aa index 10 permit 10.1.1.0 24
```

- Case1：路由10.1.1.0/24被Permit，其他都被Deny。

```
ip ip-prefix bb index 10 deny 10.1.1.0 24
```

- Case2：路由全部被Deny。

IP-Prefix的配置举例 (2)



多语句匹配

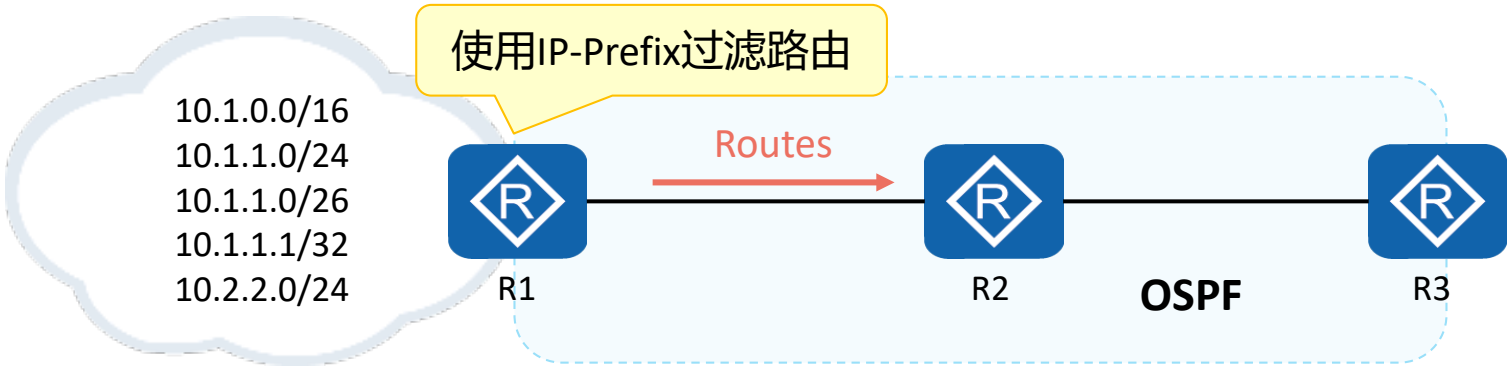
```
ip ip-prefix aa index 10 deny 10.1.1.0 24
ip ip-prefix aa index 20 permit 10.1.1.1 32
```

- Case1：路由10.1.1.0/24被Deny，路由10.1.1.1/32被Permit，其他路由都被Deny。

```
ip ip-prefix bb index 10 permit 10.1.1.0 24 greater-equal 26 less-equal 32
```

- Case2：路由10.1.1.0/26 ， 10.1.1.1/32被Permit，其他路由被Deny。

IP-Prefix的配置举例 (3)



通配地址匹配

```
ip ip-prefix aa index 10 permit 10.0.0.0 8 less-equal 32
```

- Case1：所有掩码长度在8到32的路由都被Permit。

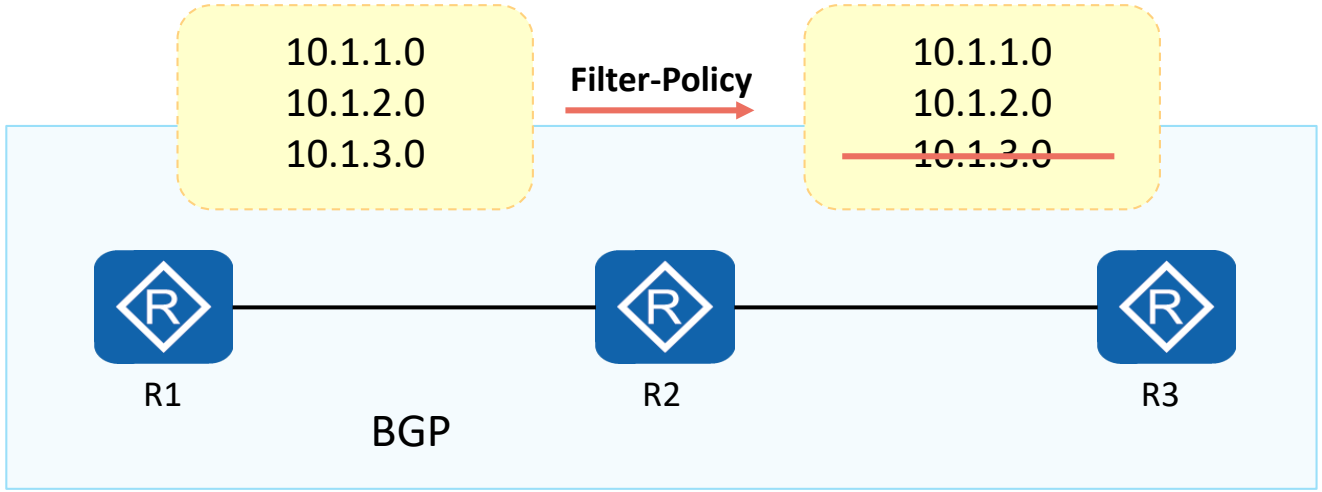
```
ip ip-prefix bb index 10 deny 10.1.1.0 24 less-equal 32
ip ip-prefix bb index 20 permit 10.1.0.0 16 less-equal 32
```

- Case2：路由10.1.0.0/16被Permit，其他路由被Deny。

目录

策略工具1： Filter-Policy

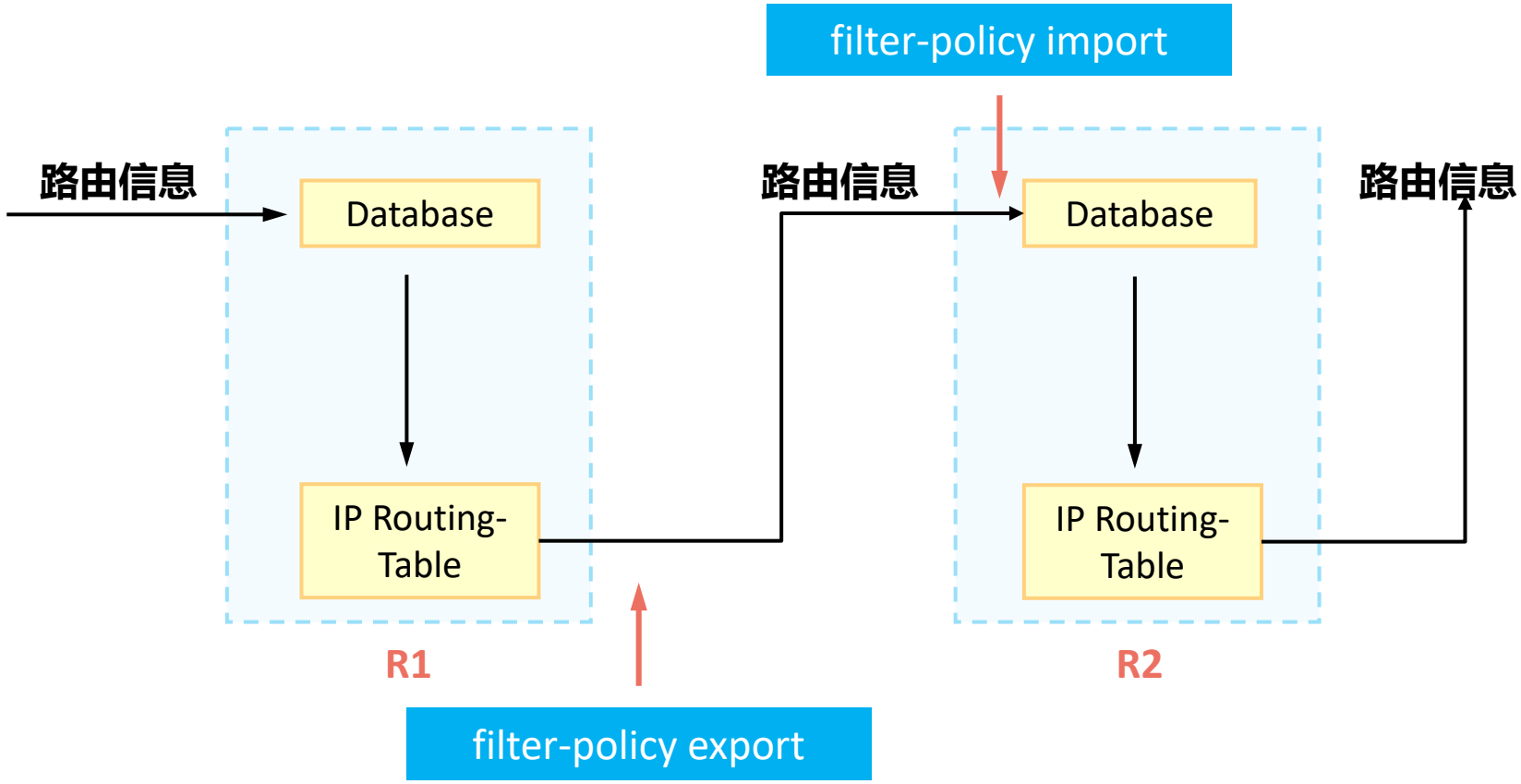
- Filter-Policy（过滤-策略）是一个很常用的路由信息过滤工具，能够对接收、发布、引入的路由进行过滤，可应用于IS-IS、OSPF、BGP等协议。



- 如图所示，R1、R2、R3之间运行BGP路由协议，路由在各个设备之间传递，当需要根据实际需求过滤某些路由信息的时候可以使用Filter-Policy实现。

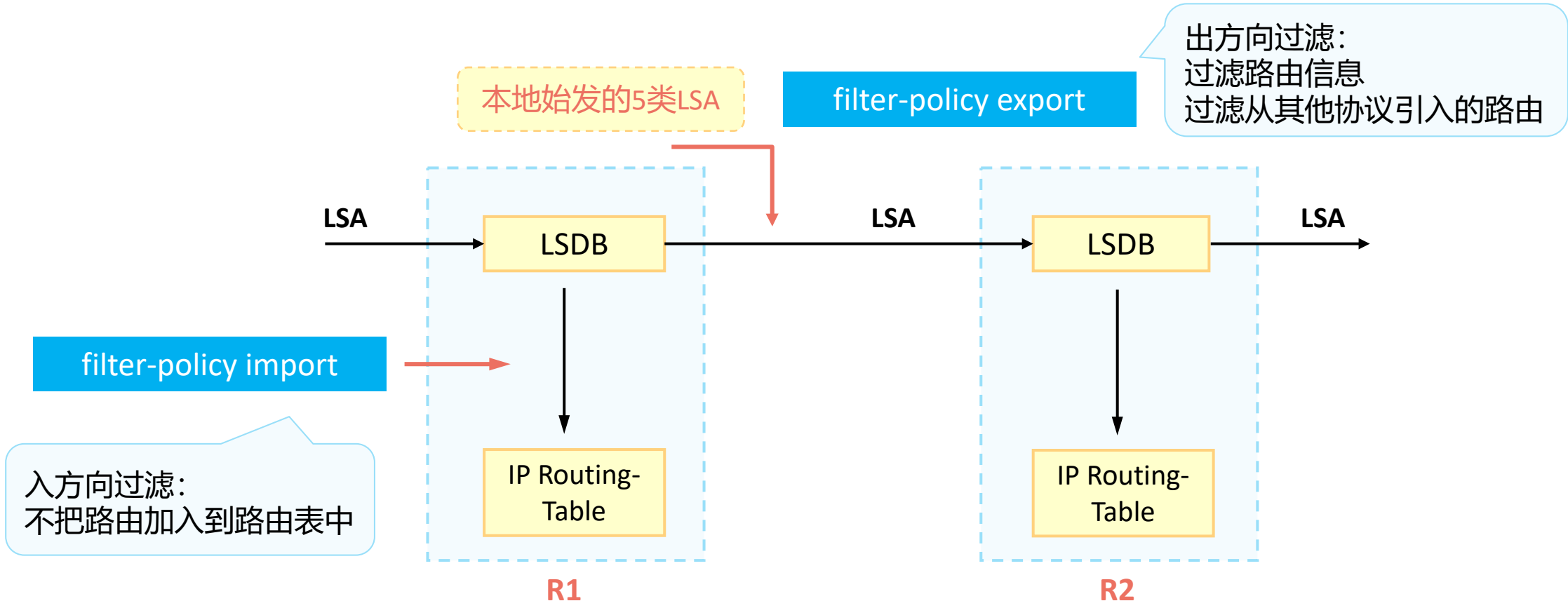
Filter-Policy在距离矢量路由协议中的应用

在距离矢量路由协议中，设备之间传递的是路由信息，如果需要对这种路由信息进行某种过滤，可以使用Filter-Policy实现，出方向和入方向的生效位置如图所示。



Filter-Policy在链路状态路由协议中的应用

在链路状态路由协议中，各路由设备之间传递的是LSA信息，然后设备根据LSA汇总成的LSDB信息计算出路由表。但是Filter-Policy只能过滤路由信息，无法过滤LSA。



Filter-Policy的基础配置命令 (1)

1. 在OSPF中的应用

[Huawei-ospf-100] **filter-policy** { *acl-number* | **acl-name** *acl-name* | **ip-prefix** *ip-prefix-name* | **route-policy** *route-policy-name* [**secondary**] } **import**

按照过滤策略，设置OSPF对接收的路由进行过滤。

[Huawei-ospf-100] **filter-policy** { *acl-number* | **acl-name** *acl-name* | **ip-prefix** *ip-prefix-name* | **route-policy** *route-policy-name* } **export** [*protocol* [*process-id*]]

按照过滤策略，设置对引入的路由在向外发布时进行过滤。

Filter-Policy的基础配置命令 (2)

2. 在IS-IS中的应用

[Huawei-isis-1] **filter-policy** { *acl-number* | **acl-name** *acl-name* | **ip-prefix** *ip-prefix-name* | **route-policy** *route-policy-name* } **import**

配置IS-IS路由加入IP路由表时的过滤策略。

[Huawei-isis-1] **filter-policy** { *acl-number* | **acl-name** *acl-name* | **ip-prefix** *ip-prefix-name* | **route-policy** *route-policy-name* } **export** [*protocol* [*process-id*]]

配置IS-IS对已引入的路由在向外发布时进行过滤的过滤策略。

Filter-Policy的基础配置命令 (3)

3. 在BGP中的应用

[Huawei-bgp-af-ipv4] **filter-policy** { *acl-number* | **acl-name** *acl-name* | **ip-prefix** *ip-prefix-name* } **import**

配置对接收的路由信息进行过滤。

[Huawei-bgp-af-ipv4] **filter-policy** { *acl-number* | **acl-name** *acl-name* | **ip-prefix** *ip-prefix-name* } **export** [*protocol* [*process-id*]]

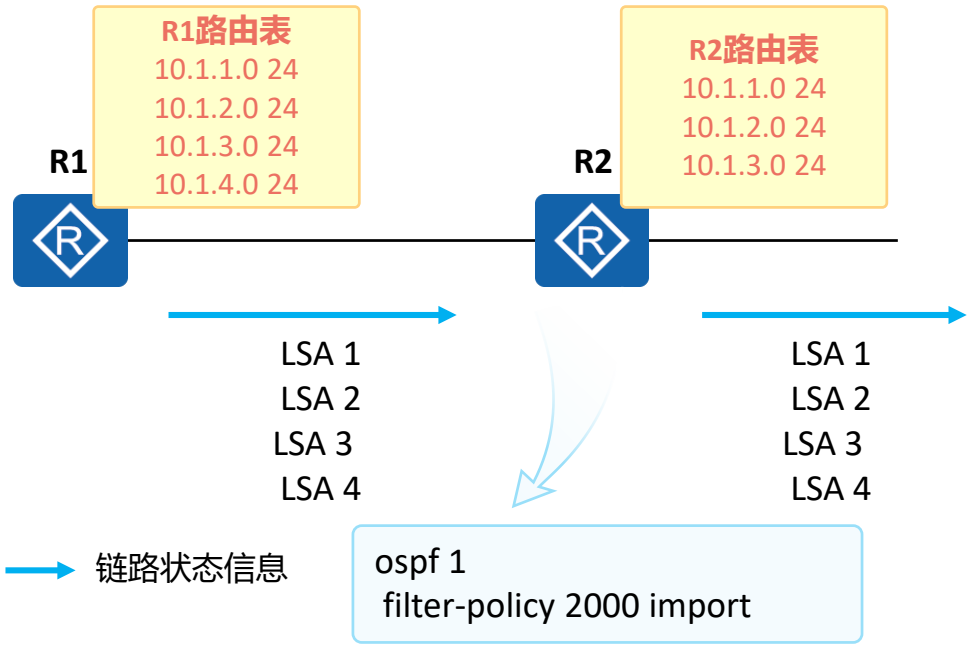
配置对发布的路由进行过滤，只有通过过滤的路由才被BGP发布。

[Huawei-bgp-af-ipv4] **peer** { *group-name* | *ipv4-address* } **filter-policy** { *acl-number* | **acl-name** *acl-name* } { **import** | **export** }

配置向对等体（组）发布或从对等体（组）接收路由时的过滤策略。

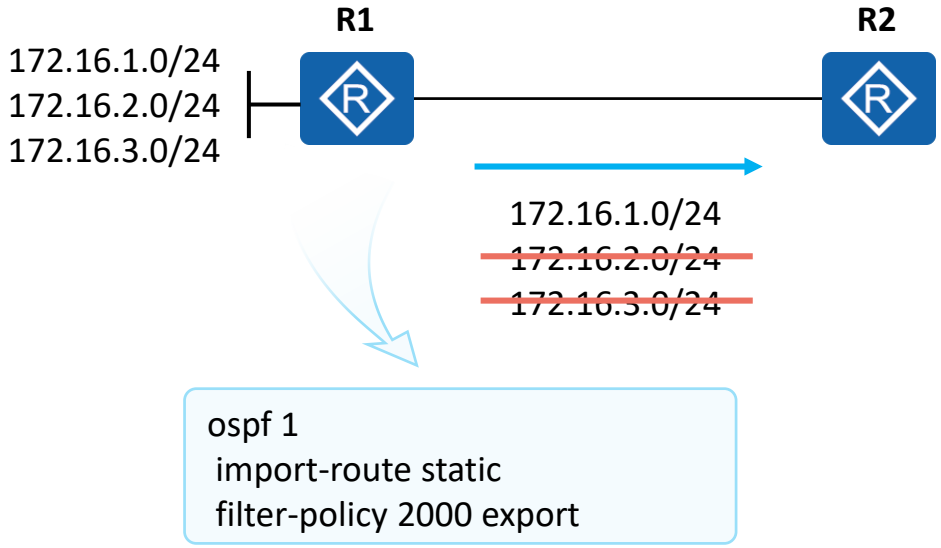
OSPF中使用Filter-Policy

filter-policy import



filter-policy import命令对接收的路由设置过滤策略，只有通过过滤策略的路由才被添加到路由表中，没有通过过滤策略的路由不会被添加进路由表，但不影响对外发布出去。

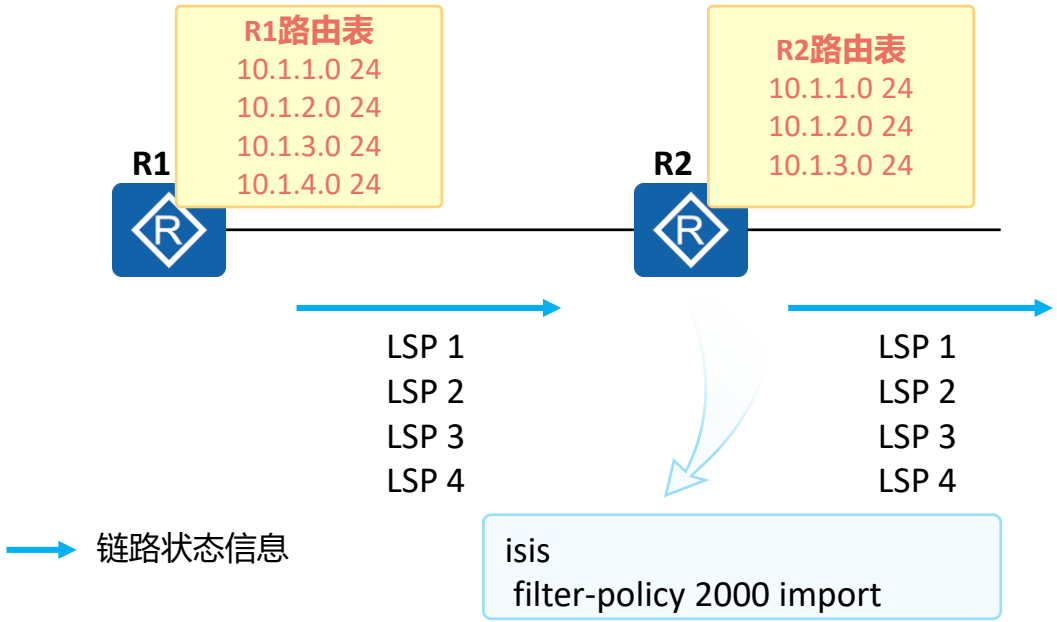
filter-policy export



OSPF通过命令**import-route**引入外部路由后，为了避免路由环路的产生，通过**filter-policy export**命令对引入的路由在发布时进行过滤，只将满足条件的外部路由转换为Type5 LSA（AS-external-LSA）并发布出去。

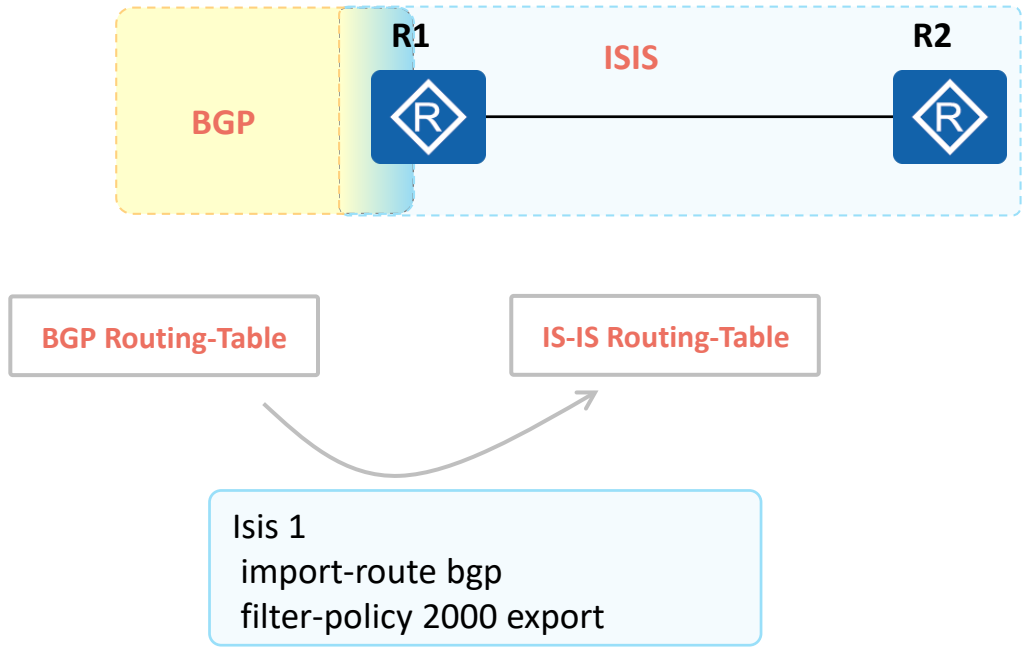
IS-IS中使用Filter-Policy

filter-policy import



与OSPF相似，**filter-policy import**命令只会对本地的路由表产生影响，不会将匹配的路由加入到路由表，不会影响本地设备的LSP的扩散和LSDB的同步。

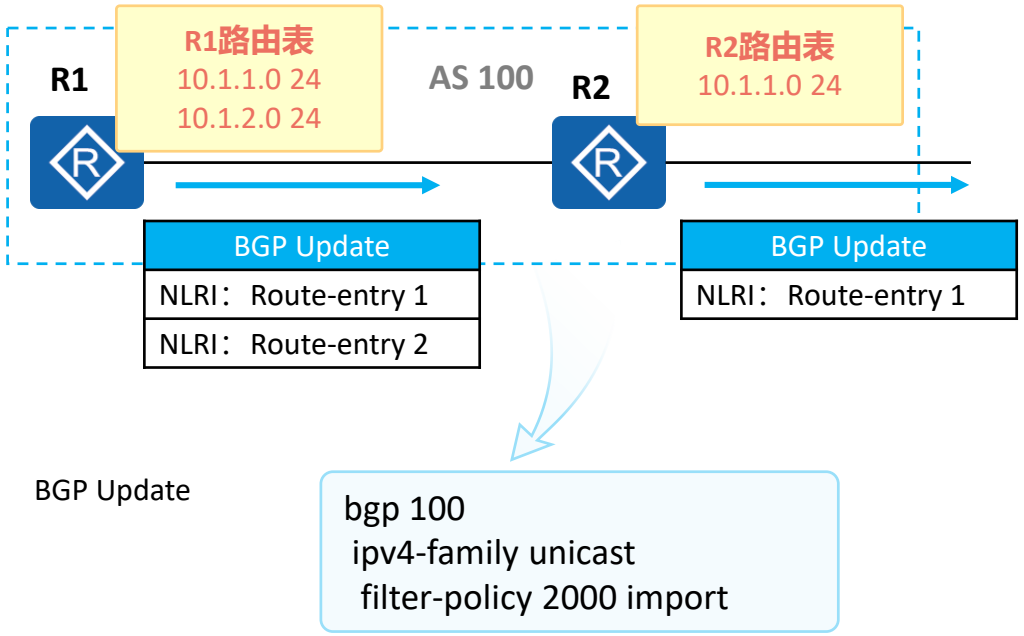
filter-policy export



当网络中同时部署了IS-IS和其他路由协议时，如果已经在边界设备上引入其他路由协议的路由，缺省情况下，该设备将把引入的全部外部路由发布给IS-IS邻居。如果只希望将引入的部分外部路由发布给邻居，可以使用**filter-policy export**命令实现。

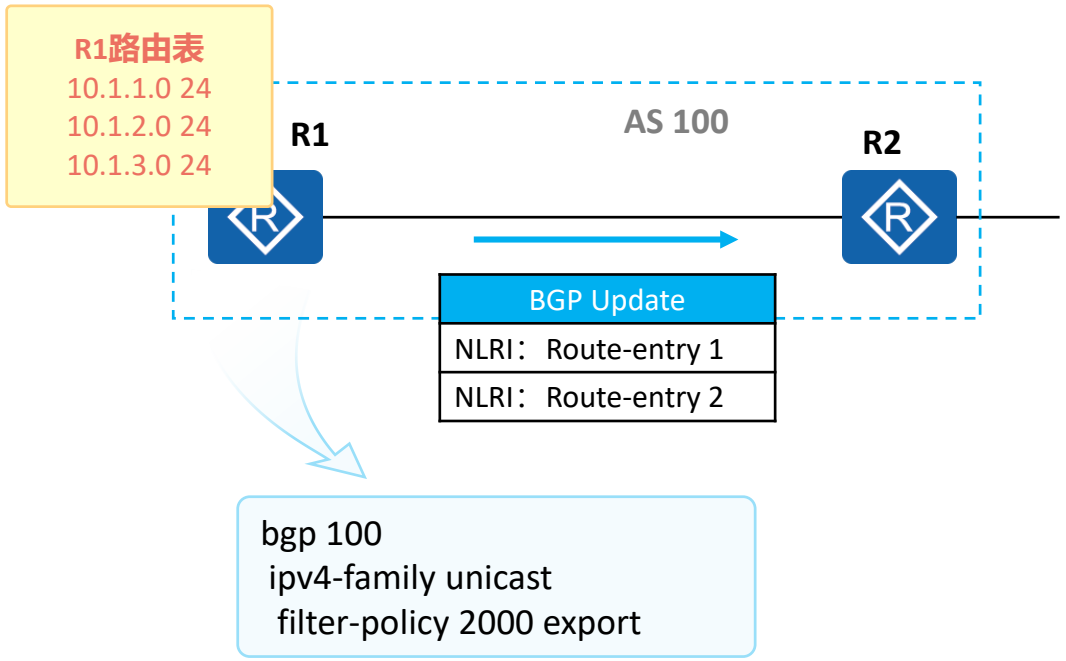
BGP中使用Filter-Policy

filter-policy import



使用**filter-policy import**命令可以对BGP设备全局接收的路由进行过滤，决定是否将路由添加到BGP路由表中。

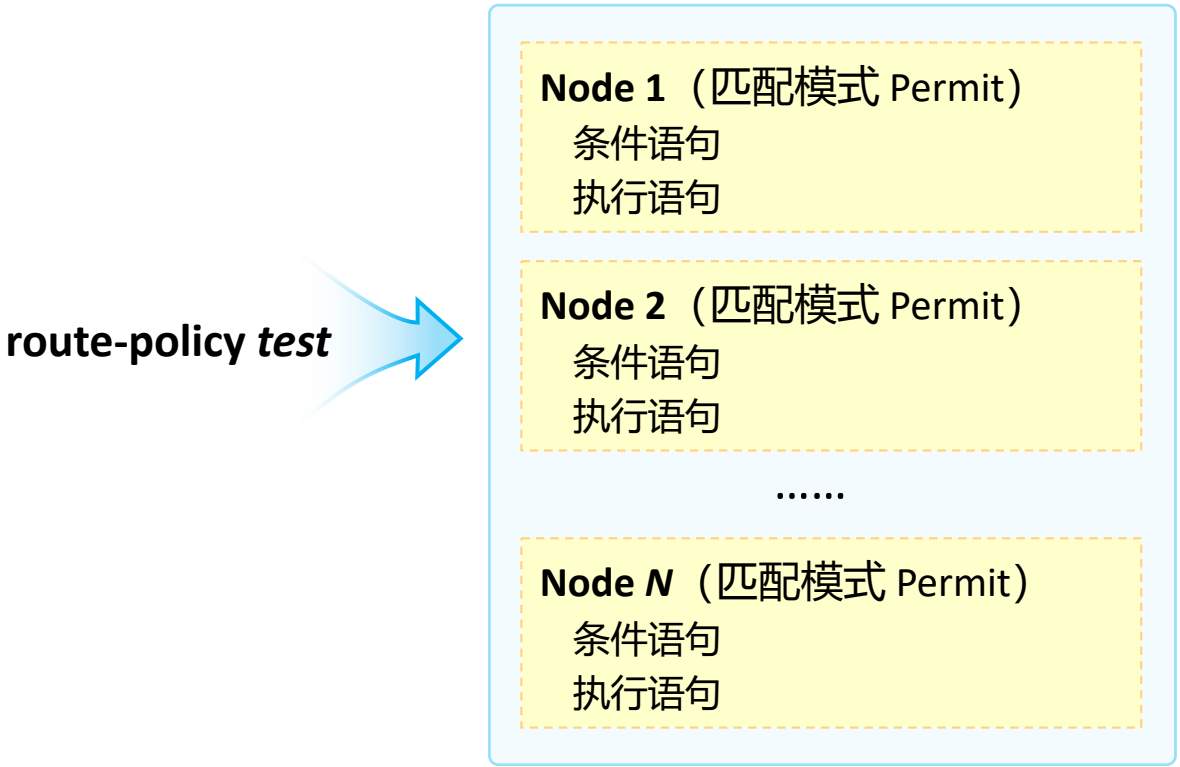
filter-policy export



使用**filter-policy export**命令可以将对外发布的路由进行过滤，只有通过过滤的路由才能加入BGP本地路由表，并被BGP发布。

策略工具2： Route-Policy

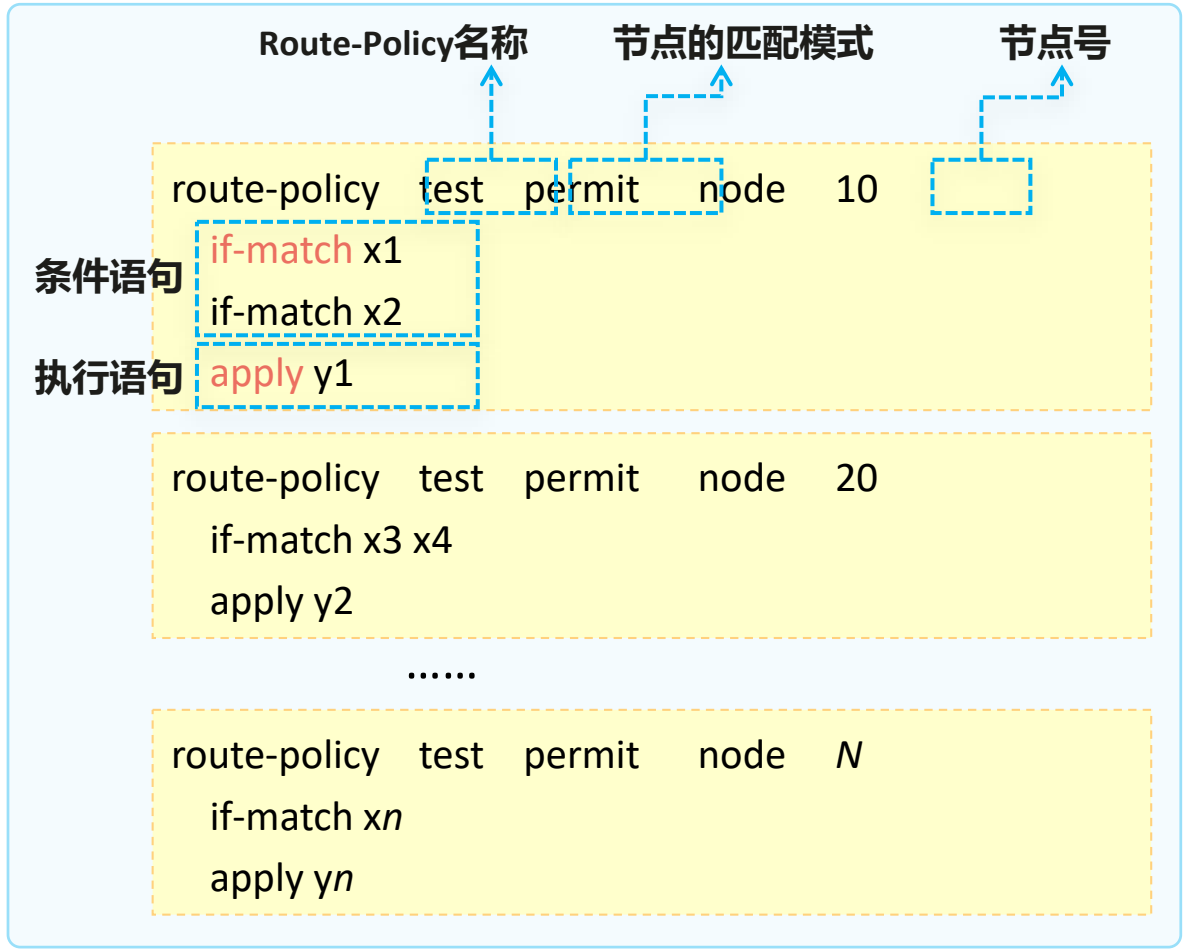
- Route-Policy是一个策略工具，用于过滤路由信息，以及为过滤后的路由信息设置路由属性。
- 一个Route-Policy由一个或多个节点（Node）构成，每个节点都可以是一系列条件语句（匹配条件）以及执行语句（执行动作）的集合，这些集合按照编号从小到大的顺序排列。



- 每个节点内可包含多个条件语句。节点内的多个条件语句之间的关系为“与”，即匹配所有条件语句才会执行本节点内的动作。
- 节点之间的关系为“或”，route-policy根据节点编号大小从小到大顺序执行，匹配中一个节点将不会继续向下匹配。

Route-Policy的组成

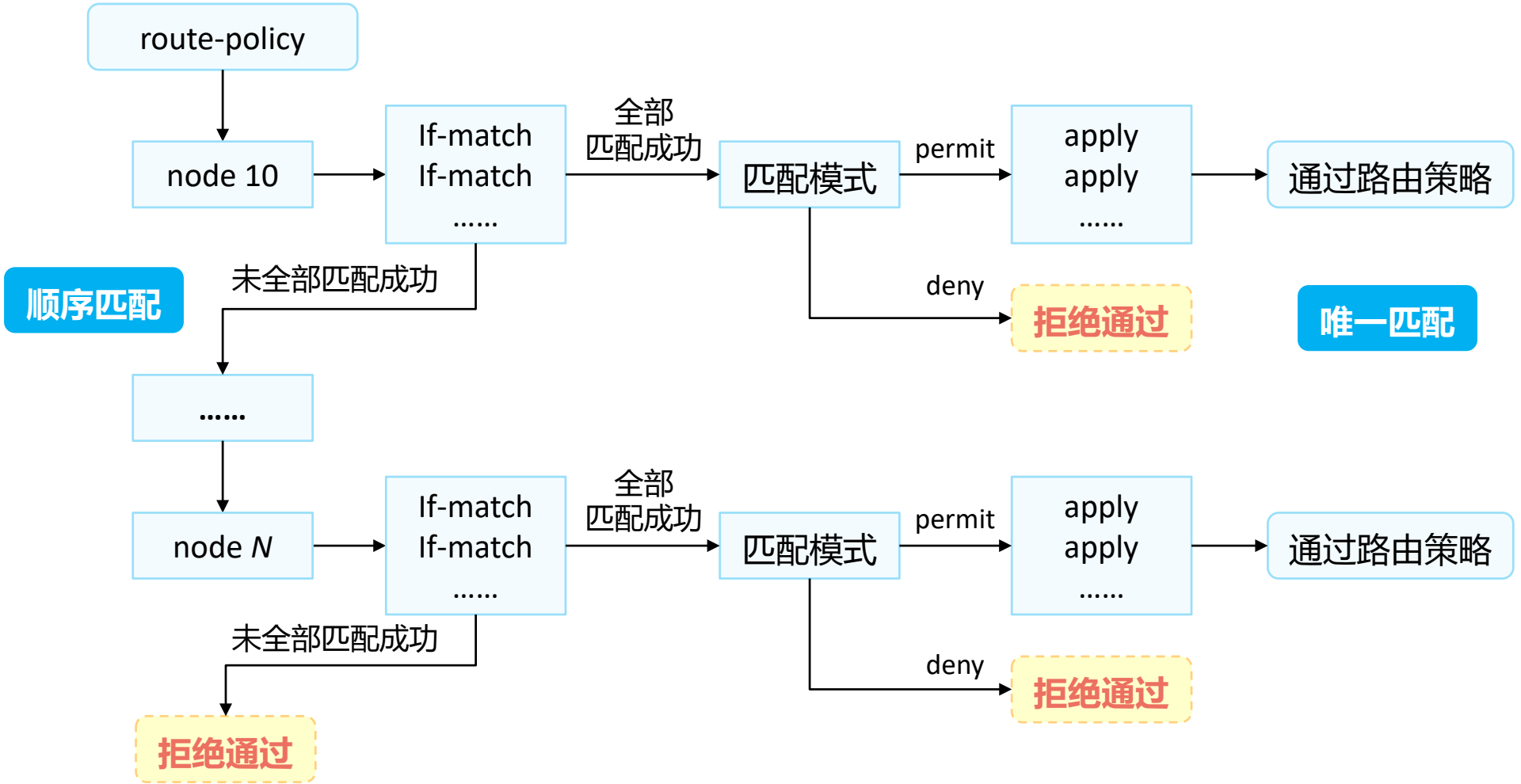
一个Route-Policy由一个或多个节点构成，每个节点包括多个if-match和apply子句。



- permit或deny：指定Route-Policy节点的匹配模式为允许或拒绝。
- node：指定Route-Policy的节点号。整数形式，取值范围是0～65535。
- if-match子句：定义该节点的匹配条件。
- apply子句：定义针对被匹配路由执行的操作。

Route-Policy的匹配顺序

路由策略使用不同的匹配条件和匹配模式选择路由和改变路由属性。



Route-Policy的基础配置命令 (1)

1. 创建Route-Policy

[Huawei] **route-policy** *route-policy-name* { **permit** | **deny** } **node** *node*

创建路由策略并进入Route-Policy视图。

2. （可选）配置if-match子句

```
[Huawei-route-policy] if-match ?
  acl                匹配基本ACL
  cost               匹配路由信息的cost
  interface          匹配路由信息的出接口
  ip-prefix          匹配前缀列表
  ......
```

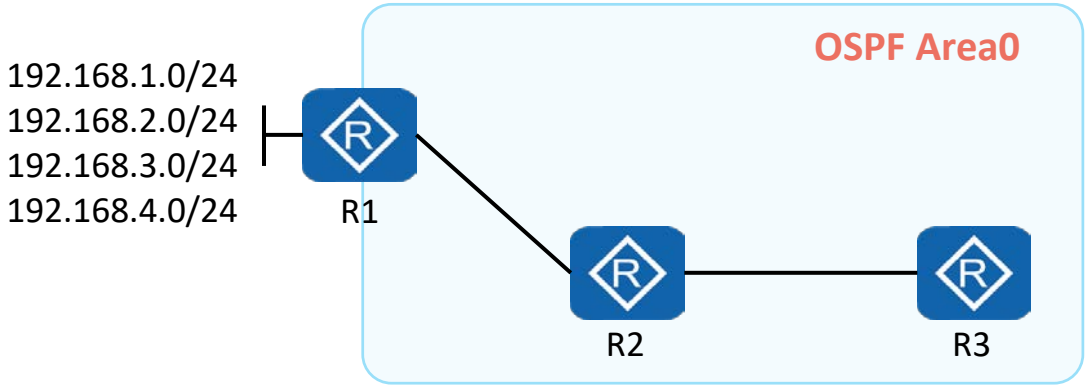
Route-Policy的基础配置命令 (2)

3. （可选）配置apply子句

```
[Huawei-route-policy] apply ?
  cost                          设置路由的cost
  cost-type {type-1 | type-2}   设置OSPF的开销类型
  ip-address next-hop           设置IPv4路由信息的下一跳地址
  preference                    设置路由协议的优先级
  tag                           设置路由信息的标记域
  ......
```

目录

对接收的路由进行过滤



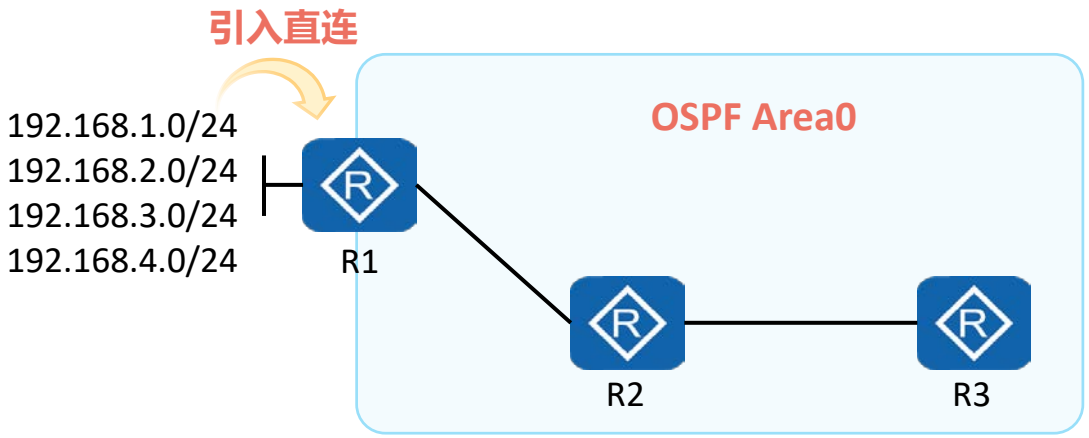
- R1 、 R2 、 R3 运 行 OSPF ， R1 将 192.168.1.0/24 、 192.168.2.0/24、 192.168.3.0/24和192.168.4.0/24宣告进OSPF。
- 现在要求R2不能访问R1上192.168.1.0/24网段，但是R3可以正常访问。
- 为实现该需求，可以在R2上对接收的路由使用Filter-Policy进行过滤。

R2的配置如下：

```
[R2] ip ip-prefix in index 10 permit 192.168.2.0 24
[R2] ip ip-prefix in index 20 permit 192.168.3.0 24
[R2] ip ip-prefix in index 30 permit 192.168.4.0 24

[R2] ospf
[R2-ospf-1] filter-policy ip-prefix in import
```

注意：网络基础配置略。

对发布的路由进行过滤



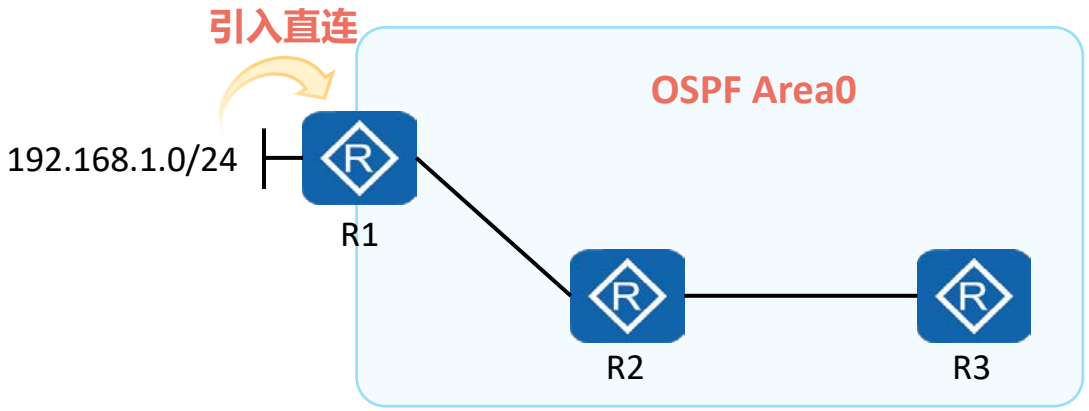
- R1、R2、R3运行OSPF，R1将直连网段192.168.1.0/24、192.168.2.0/24 、 192.168.3.0/24 和 192.168.4.0/24 引 入 OSPF。
- 现在要求R2、R3只能学习到192.168.1.0/24网段的路由，学习不到其他三个网段的路由。
- 为实现该需求，可以在R1上使用Filter-Policy对引入的路由在发布时进行过滤。

R1的配置如下：

```
[R1] ip ip-prefix out index 10 permit 192.168.1.0 24

[R1] ospf
[R1-ospf-1] import-route direct
[R1-ospf-1] filter-policy ip-prefix out export
```

注意：网络基础配置略。

修改路由属性



- R1、R2、R3运行OSPF，R1将直连网段192.168.1.0/24引入OSPF。
- 现在要求R2、R3学到的OSPF路由192.168.1.0/24为external-type 1路由（默认为external-type 2路由）。
- 为实现该需求，可以在R1上使用Route-Policy在引入路由时修改外部路由的类型为external-type 1。

R1的配置如下：

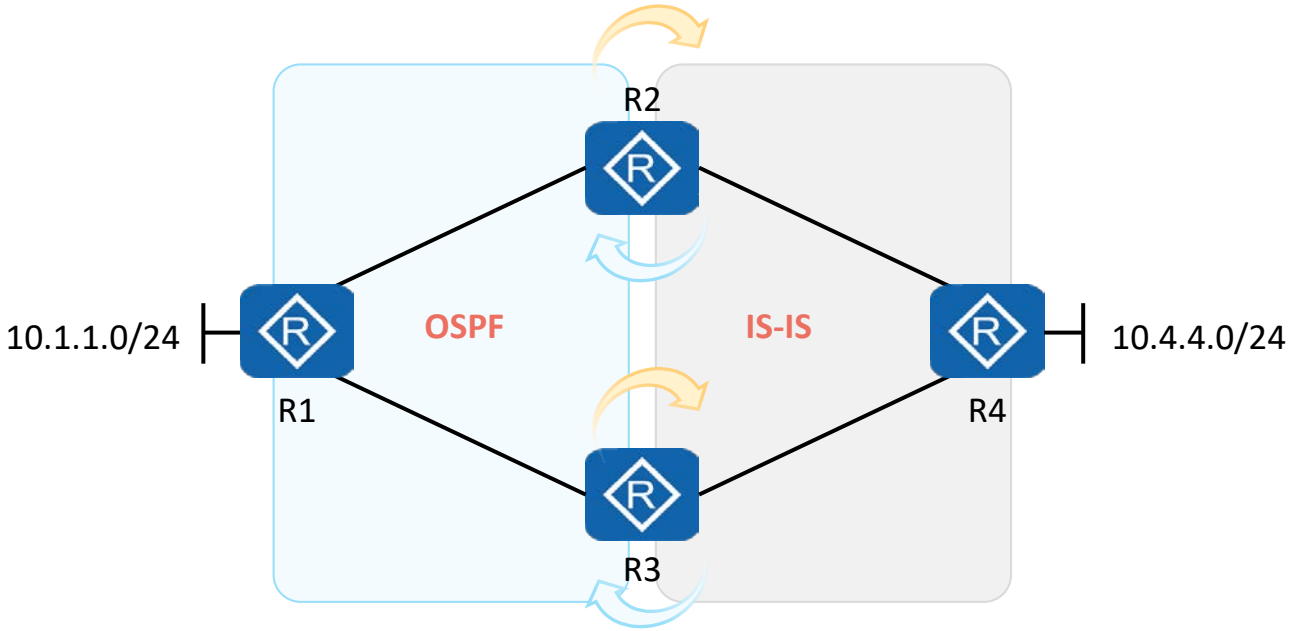
```
[R1] ip ip-prefix external index 10 permit 192.168.1.0 24

[R1] route-policy RP permit node 10
[R1-route-policy] if-match ip-prefix external
[R1-route-policy] apply cost-type type-1
[R1-route-policy] quit

[R1] ospf
[R1-ospf-1] import-route direct route-policy RP
```

注意：网络基础配置略。

双点双向路由重发布

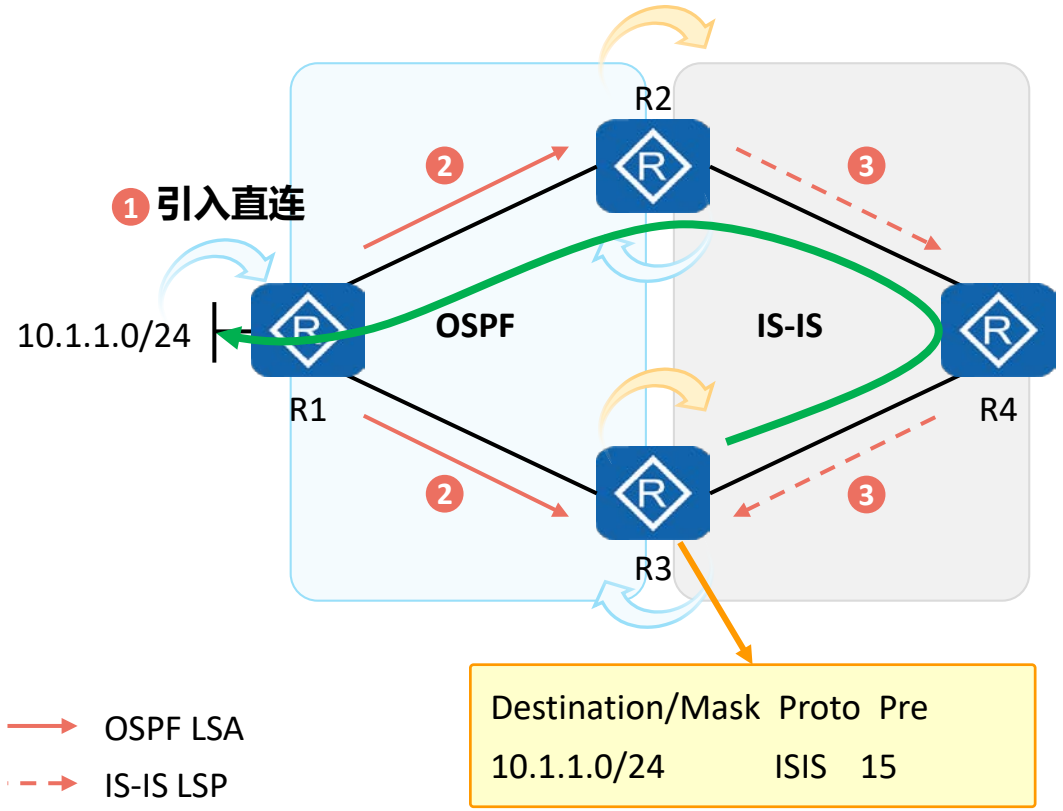
IS-IS中引入OSPF

OSPF中引入IS-IS



- 在边界路由器上把两个路由域的路由相互引入，称之为双向路由重发布。
- 两个路由域存在两个边界路由器，并且都执行双向路由重分发，此时称为双点双向路由重发布。
- 双点双向路由重发布是一种经典的路由模型，因单点的双向路由重发布缺乏冗余性，一旦单点的边界路由器故障，那么两个路由域之间的通信可能就会出现问题，因此在大型网络部署中一般采用双点双向路由重发布。
- 双点双向重路由发布虽然增强了网络的可靠性，但是容易引发：次优路径、路由环路等问题。

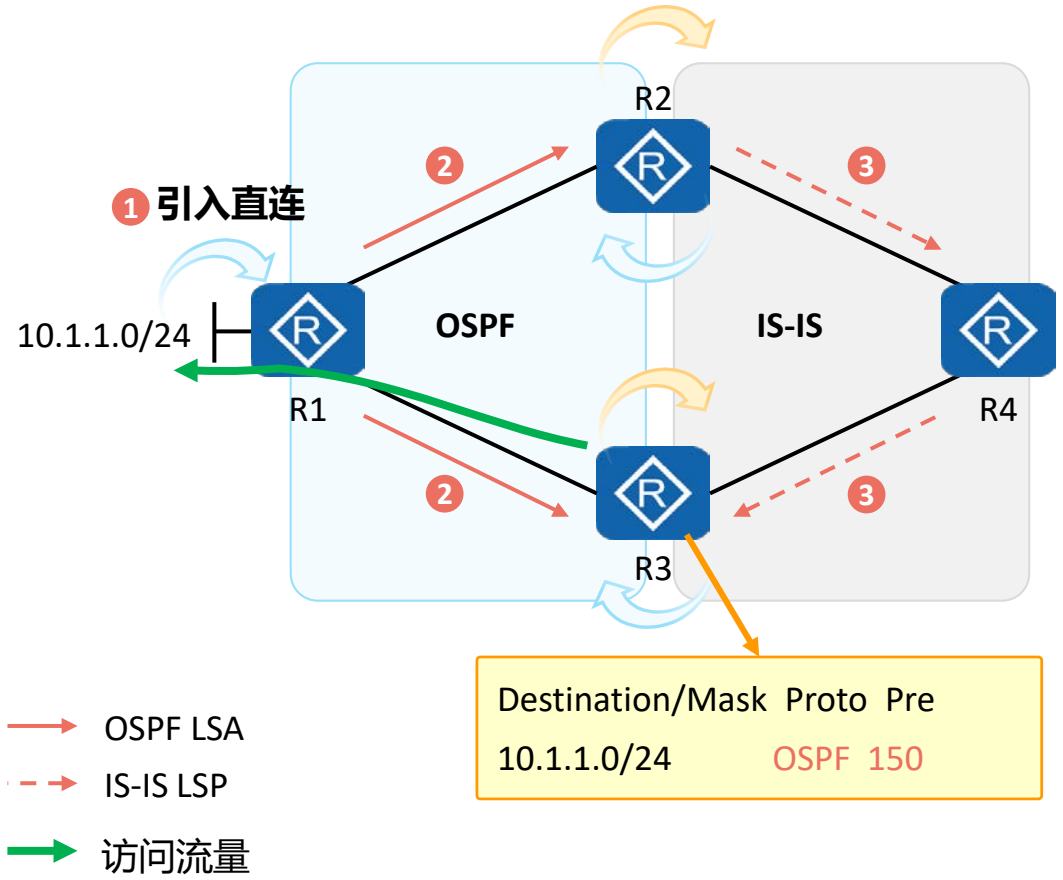
次优路径问题



以10.1.1.0/24为例：

- R1将直连路由10.1.1.0/24引入到OSPF中。
- R2、R3执行双向路由重发布，R2先将10.1.1.0/24重发布到IS-IS中，R3将会学习到来自R4的IS-IS路由。
- 对R3而言，IS-IS路由（优先级15）优于OSPF外部路由（优先级150），因此优选来自R4的IS-IS路由。后续R3访问10.1.1.0/24网段的路径为：R3->R4->R2->R1，这是次优路径。

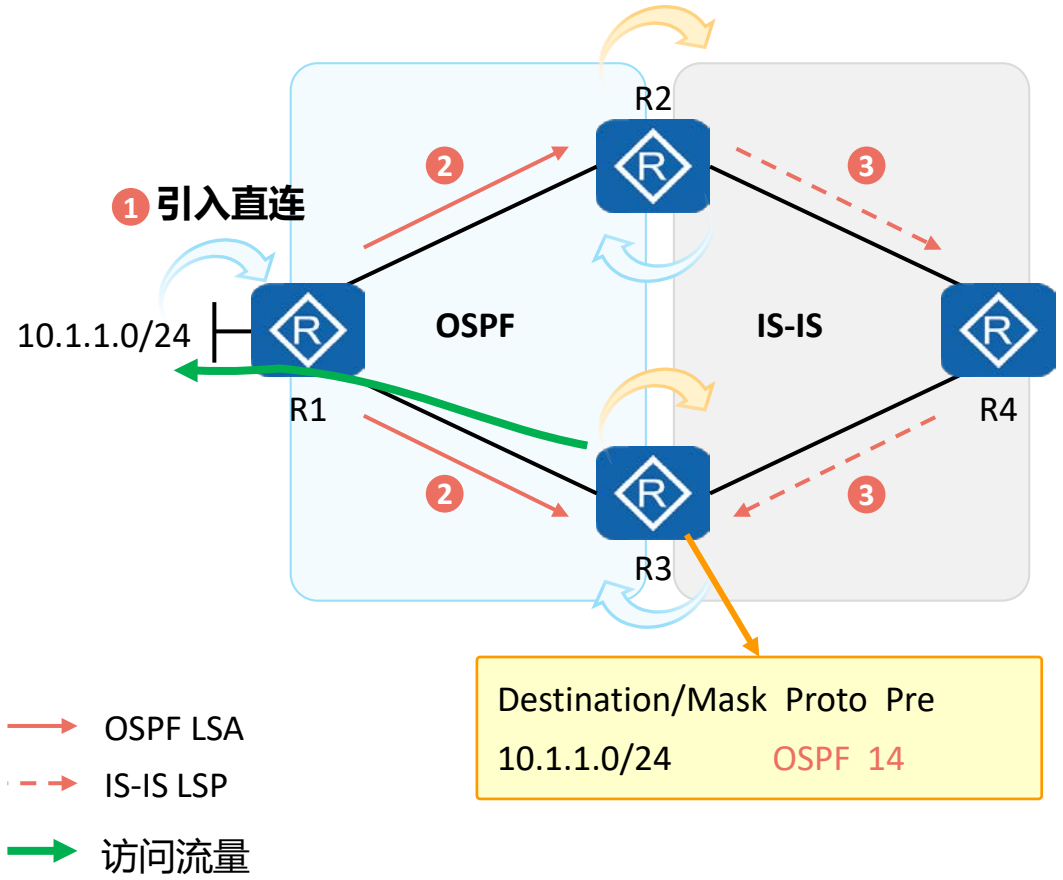
解决次优路径问题 (1)



- 解决方案一：在R3的IS-IS进程内，通过Filter-Policy禁止来自R4的10.1.1.0/24路由加入本地路由表。
- 在R3上执行以下操作：

```
[R3] acl 2001
[R3-acl-basic-2001] rule 5 deny source 10.1.1.0 0
[R3-acl-basic-2001] rule 10 permit

[R3] isis
[R3-isis-1] filter-policy 2001 import
```

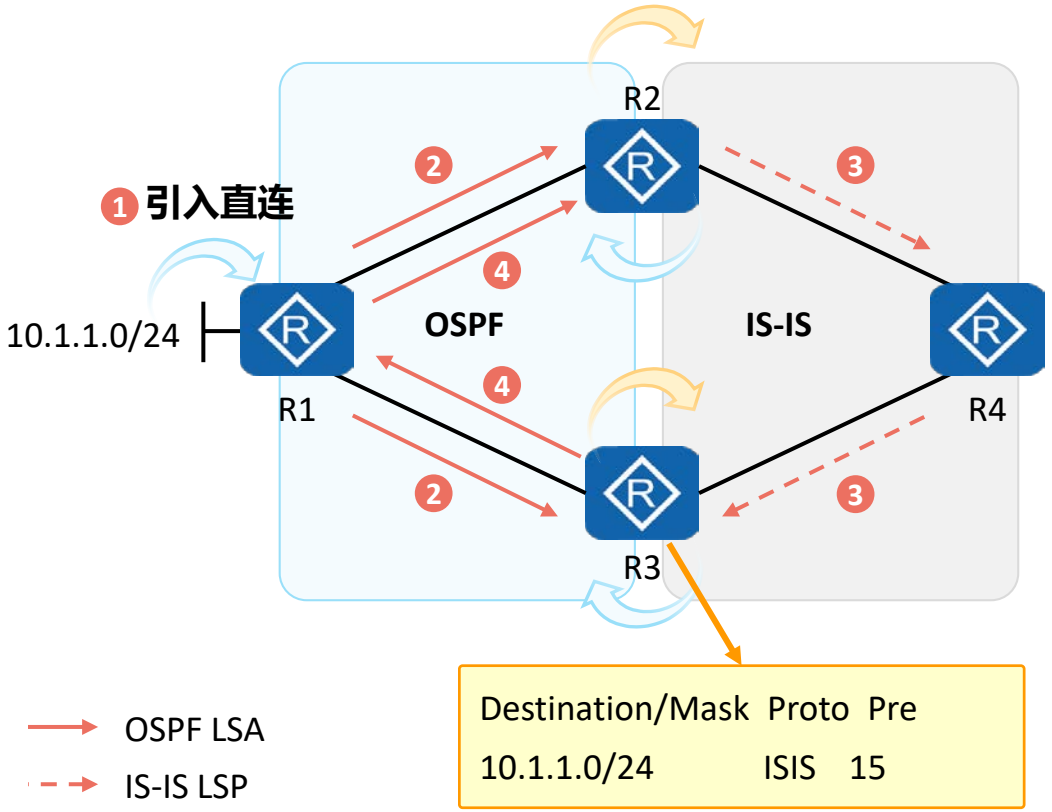
解决次优路径问题 (2)



- 解决方案二：R3通过ACL匹配10.1.1.0/24路由，在Route-Policy中调用该条ACL，将匹配这条ACL的路由的优先级设置为14（优于IS-IS）。在OSPF视图下使用**preference ase**命令调用Route-Policy修改外部路由的优先级。
- 在R3上执行以下操作：

```
[R3]acl 2000
[R3-acl-basic-2000] rule permit source 10.1.1.0 0
[R3-acl-basic-2000] quit
[R3]route-policy hcip permit node 10
[R3-route-policy] if-match acl 2000
[R3-route-policy] apply preference 14
[R3-route-policy] quit
[R3]ospf 1
[R3-ospf-1] preference ase route-policy hcip
```

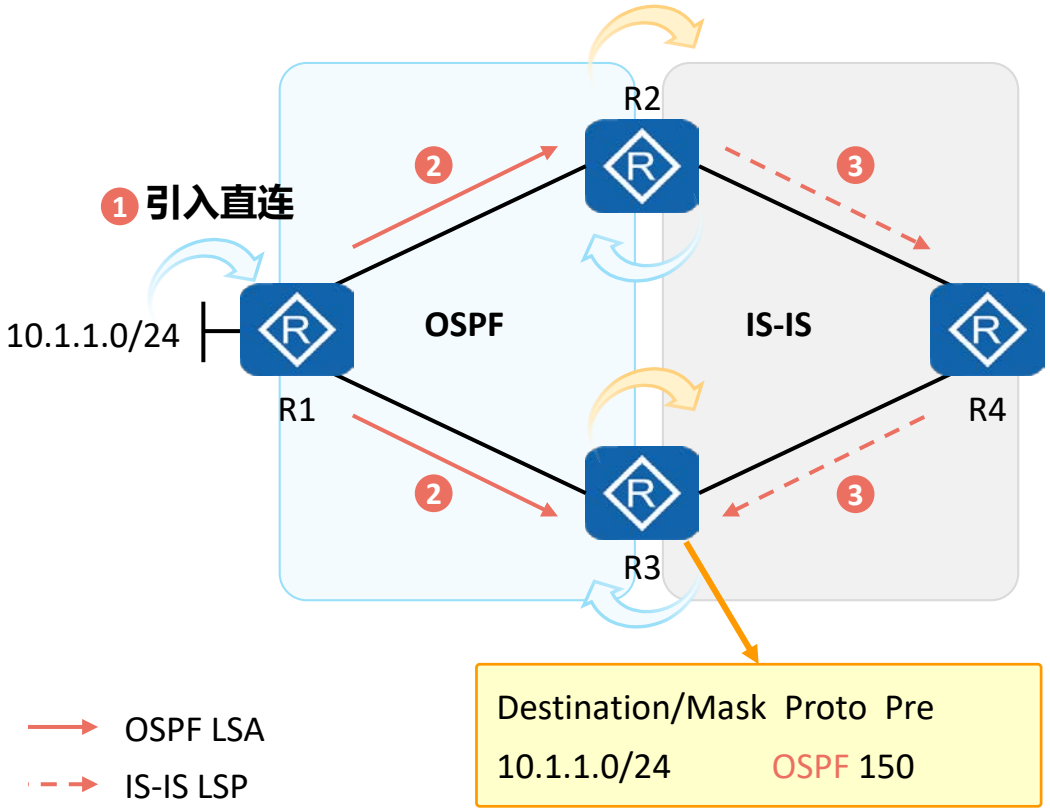
路由环路问题



场景描述：

1. R1将直连路由10.1.1.0/24引入到OSPF中。
2. R1、R2、R3运行OSPF协议，10.1.1.0/24网段路由在全OSPF域内通告。
3. R2执行了双向路由重发布。
4. R2、R3、R4运行IS-IS协议， 10.1.1.0/24网段路由在全IS-IS域内通告。
5. R3执行了双向路由重发布。
6. 10.1.1.0/24网段路由再次被通告进OSPF域内，形成路由环路。

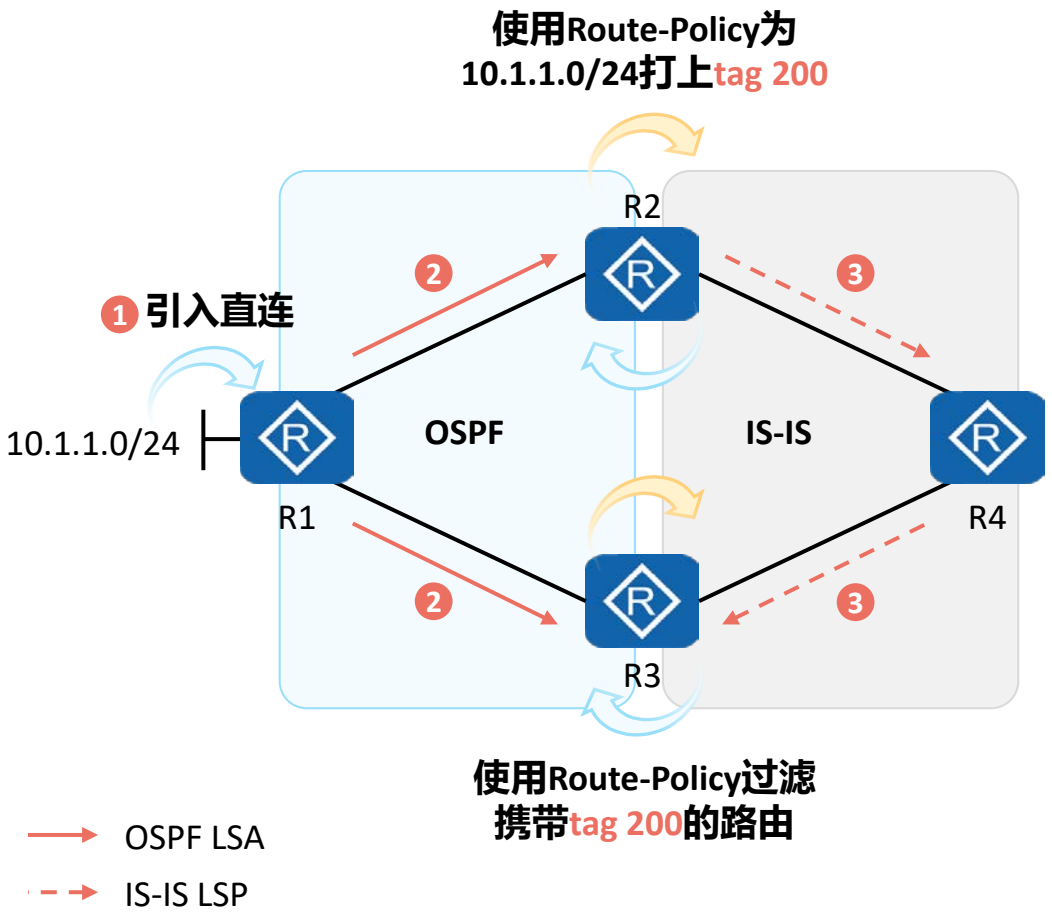
解决路由环路问题 (1)



- 解决方案一：在R3的OSPF中引入IS-IS路由时，通过Route-Policy过滤掉10.1.1.0/24路由。
- 在R3上执行以下操作：

```
[R3] acl 2001
[R3-acl-basic-2001] rule 5 deny source 10.1.1.0 0
[R3-acl-basic-2001] rule 10 permit

[R3] route-policy RP permit node 10
[R3-route-policy] if-match 2001
[R3-route-policy] quit

[R3] ospf
[R3-ospf-1] import-route isis 1 route-policy RP
```

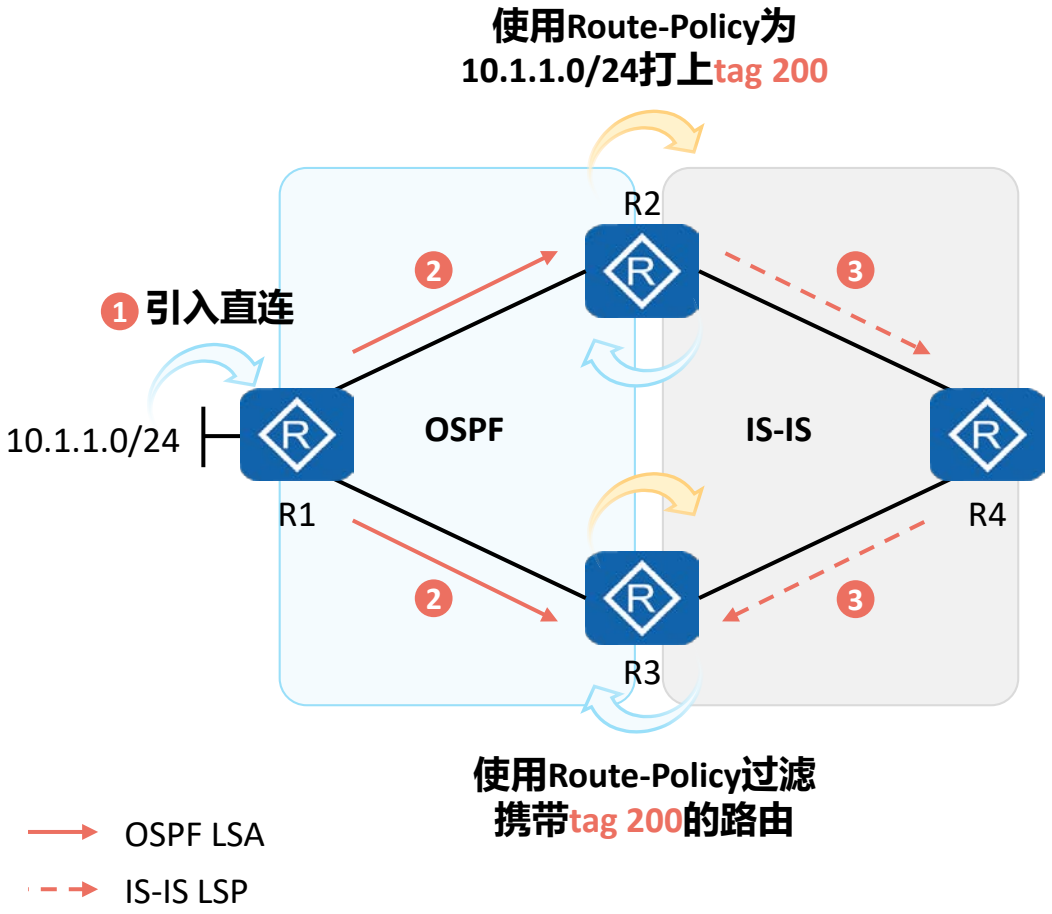
解决路由环路问题 (2)



- 解决方案二：使用Tag实现有选择性地路由引入，在R2上将路由10.1.1.0/24从OSPF引入到IS-IS中时打上Tag 200，在R3上将IS-IS引入到OSPF中时，过滤携带Tag 200的路由。
- 在R2上执行如下操作：

```
[R2]acl 2000
[R2-acl-basic-2000]rule permit source 10.1.1.0 0
[R2-acl-basic-2000]quit
[R2]route-policy hcip permit node 10
[R2-route-policy]if-match acl 2000
[R2-route-policy]apply tag 200
[R2-route-policy]quit
[R2]isis 1
[R2-isis-1]import-route ospf route-policy hcip
```

解决路由环路问题 (3)

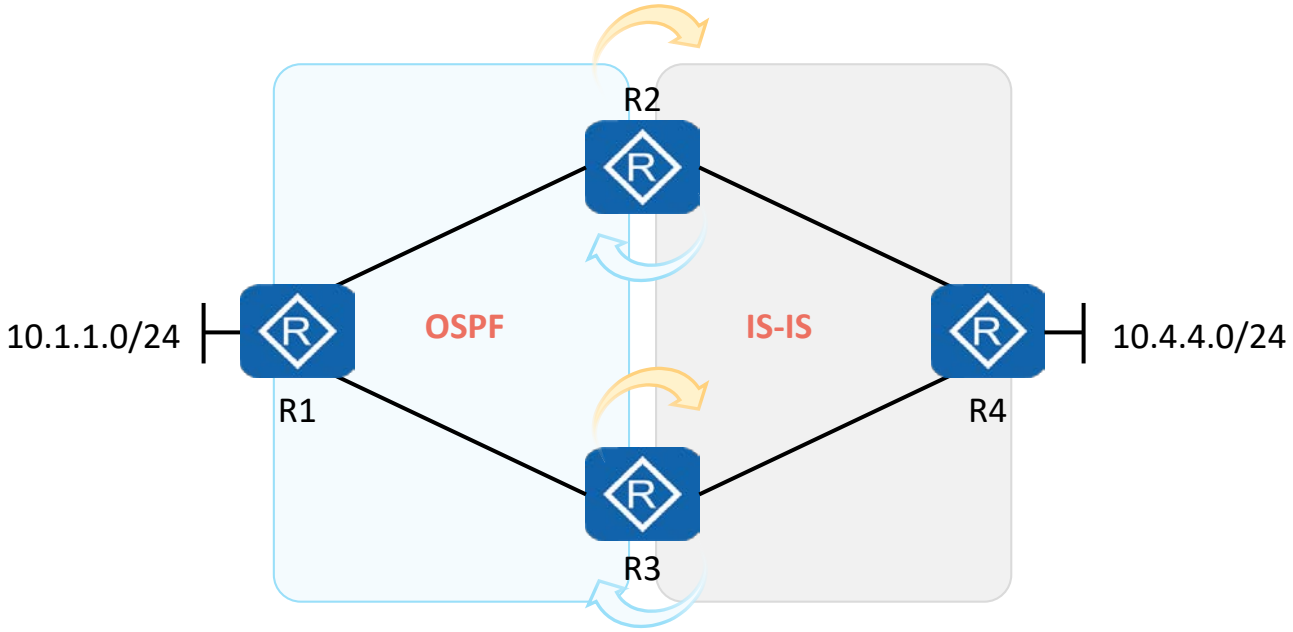


- 在R3上执行如下操作：

```
[R3]route-policy hcip deny node 10
[R3-route-policy]if-match tag 200
[R3-route-policy]quit
[R3]route-policy hcip permit node 20

[R3]ospf 1
[R3-ospf-1]import-route isis route-policy hcip
```

在路由重发布的实际应用中，通过IP前缀进行路由匹配固然可行，但当网络规模较大时，配置工作量较大；通过Tag进行路由匹配可以极大简化配置工作量。

场景思考



在R1上将10.1.1.0/24引入到OSPF，在R4上将10.4.4.0/24引入到IS-IS，R2、R3上执行路由双向路由重发布，该场景下如何使用匹配Tag的方式防止路由环路?

思考题

1. （简答题）Filter-Policy export在OSPF、BGP中的作用分别是？
2. （简答题）Route-Policy多个节点之间的逻辑关系为？一个节点内多个条件语句的逻辑关系为？

本章总结

- 控制路由的发布、接收时需要先将相应的路由使用匹配器进行抓取，最常见的匹配器有ACL、 IP-Prefix List 。
- Filter-Policy、 Route-Policy都可用来在发布、接收路由时进行过滤，但需要注意在链路状态路由协议中使用Filter-Policy并不能正常的过滤链路状态信息，只是影响了本地的路由表。
- Route-Policy在发布、接收路由时可以对路由的属性进行灵活地修改，以实现XXX。


谢 谢
www.huawei.com