HCRSE108-路由 Import&Control

路由选择工具

ACL 访问控制类表

ACL 是由 permit 或 deny 语句组成的一系列有顺序规则的集合，它通过匹配报文的信息实现对报文的分类。路由器根据 ACL 定义的规则判断哪些报文可以接收，哪些报文需要拒绝，从而实现对报文的过滤。ACL 通过配置的一系列匹配规则对特定的数据包进行过滤，从而识别需要过滤的对象。然后，根据预先设定的策略允许或禁止相应的数据包通过。同时，ACL 可以作为基础配置被其他功能模块引用。

ip-prefix 前缀类表

ip-prefix 将路由条目与前缀过滤列表的每一项进行匹配，然后根据定义的匹配模式进行过滤，达成路由筛选的目的。 ip-prefix 不能用来过滤数据包，只能过滤路由信息。

as-path-filter AS 路径过滤器

BGP 的路由信息中，包含一个自治系统路径域。As-path-filter 就是针对自治系统路径域指定匹配条件。AS 路径过滤器仅应用于 BGP 协议。

community-filter 团体属性过滤器

BGP 的路由信息中，可以携带一个或多个团体属性，团体属性过滤器就针对团体属性域指定匹配条件。

ACL

ACL 编号：用于标识 ACL，表明该 ACL 是数字型 ACL。
根据 ACL 规则功能的不同，ACL 被划分为基本 ACL、高级 ACL、二层 ACL 和用户 ACL 这几种类型，每类 ACL 编号的取值范围不同。
基本 ACL（编号范围 2000-2999）
高级 ACL（编号范围 3000-3999）

二层 ACL（编号范围 4000-4999）
用户 ACL（编号范围 6000-6031）

除了可以通过 ACL 编号标识 ACL，设备还支持通过名称来标识 ACL，就像用域名代替 IP 地址一样，更加方便记忆。这种 ACL，称为命名型 ACL。
命名型 ACL 实际上是“名字+数字”的形式，可以在定义命名型 ACL 时同时指定 ACL 编号。如果不指定编号，则由系统自动分配。
规则：即描述报文匹配条件的判断语句。
规则编号：用于标识 ACL 规则。可以自行配置规则编号，也可以由系统自动分配。ACL 规则的编号范围是 0～4294967294，所有规则均按照规则编号从小到大进行排序。系统按照规则编号从小到大的顺序，将规则依次与报文匹配，一旦匹配上一条规则即停止匹配。
动作：包括 permit/deny 两种动作，表示允许/拒绝。
匹配项：ACL 定义了极其丰富的匹配项。除了源地址和生效时间段，ACL 还支持很多其他规则匹配项。例如，二层以太网帧头信息（如源 MAC、目的 MAC、以太帧协议类型）、三层报文信息（如目的地址、协议类型）以及四层报文信息（如 TCP/UDP 端口号）等。
如果规则存在，则系统会从 ACL 中编号最小的规则开始查找。如果匹配上了 permit 规则，则停止查找规则，并返回 ACL 匹配结果为：匹配（允许）。如果匹配上了 deny 规则，则停止查找规则，并返回 ACL 匹配结果为：匹配（拒绝）。如果未匹配上规则，则继续查找下一条规则，以此循环。如果一直查到最后一条规则，报文仍未匹配上，则返回 ACL 匹配结果为：不匹配。

基本 ACL：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：既可使用 IPv4 报文的源 IP 地址，也可使用目的 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：使用报文的以太网帧头信息来定义规则，如根据源 MAC（Media Access Control）地址、目的 MAC 地址、二层协议类型等。
用户 ACL：既可使用 IPv4 报文的源 IP 地址，也可使用目的 IP 地址、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。

除此之外，还有基于 IPv6 的 ACL，基本 ACL6 和高级 ACL6。
基本 ACL6：可使用 IPv6 报文的源 IPv6 地址、分片信息和生效时间段来定义规则。
高级 ACL6：可以使用 IPv6 报文的源 IPv6 地址、目的 IPv6 地址、IPv6 协议类型、ICMPv6 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。

IP-Prefix

用于过滤 IP 前缀，能同时匹配前缀号和掩码长度 ，不能用于数据包过滤
缺省情况下，存在最后一条默认匹配模式为 deny
当引用的前缀过滤列表不存在时，默认匹配模式为 permit

每个地址前缀列表可以包含多个 IP-Prefix 条目，每个 IP-Prefix 条目对应一个索引号（index）。路由将按照索引号从小到大依次检查 IP-Prefix 列表，任意一个 index 匹配成功，将不再检查其余项。若所有 index 都匹配失败，路由信息将被过滤。
根据匹配的前缀不同，前缀过滤列表可以进行精确匹配，也可以进行在一定掩码长度范围内匹配。

前缀过滤列表可以进行精确匹配或者在一定掩码长度范围内匹配，可以通过配置关键字 greater-equal 和 less-equal 指定待匹配的前缀掩码长度范围。如果没有配置关键字 greater-equal 或 less-equal，前缀过滤列表进行精确匹配，即只匹配掩码长度为前缀过滤列表掩码长度的相同 IP 地址路由；如果只配置了关键字 greater-equal，则待匹配的掩码长度范围为从 greater-equal 指定值到 32 位长度；如果只匹配了关键字 less-equal，则待匹配的掩码长度范围为从指定的掩码到关键字 less-equal 指定值。
greater-equal-value 与 less-equal-value 的取值限制：mask-length<=greater-equal-value<=less-equal-value<=32。

IPv6-Prefix

IPv6-Prefix 根据配置的规则对 IPv6 报文进行分类，其实现原理和 IP-Prefix 基本相同。
IPv6 地址前缀列表用于过滤 IPv6 地址。同一个地址前缀列表可包含多个表项，每个表项指定一个地址前缀范围。此时，各表项之间是“或”的关系，即只要通过其中一个表项就认为已通过该地址前缀列表的过滤，所有表项都没有通过则意味着没有通过该地址前缀列表的过滤。
默认所有未匹配的路由将被拒绝通过过滤列表。如果所有表项都配置成 deny 模式，则任何路由都不能通过该过滤列表。因此，需要在多条 deny 模式的表项后定义一条 permit :: 0 less-equal 128 的表项，以允许其它所有 IPv6 路由信息通过。

AS-Path-Filter

以 BGP 中的 AS_Path 属性为匹配条件 ，使用正则表达式进行定义

AS 路径过滤器是将 BGP 中的 AS_Path 属性作为匹配条件的过滤器，只有 BGP 在收发路由的时候才能使用。由于 AS_Path 属性记录 AS 号是将最后经历的 AS 号放在 AS_Path 记录中的最左侧，所以在配置 as-path-filter 时需要格外注意。
如果一条路由起源于 AS100，然后依次经过 AS300, AS200, AS500，最后到达 AS600。那么在 AS600 里，路由的 AS-PATH 属性表示为（500 200 300 100）。

表达式	含义
^$	表示本地AS始发
.*	表示所有
_10_	表示必须通过AS10
^10_	表示只接受来自AS10的路由
_10$	从AS10始发的所有路由

Community-Filter

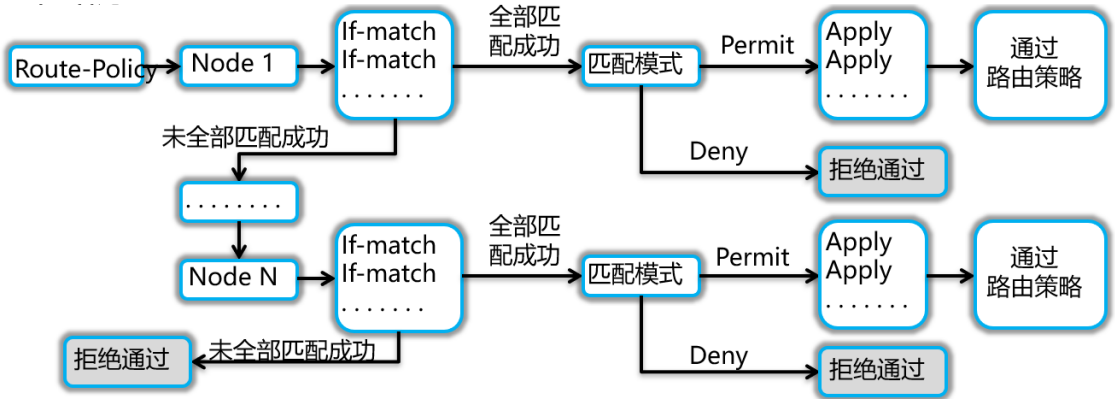
以 BGP 中的 community 属性为匹配条件 。团体属性过滤器是将 BGP 中的团体属性作为匹配条件的过滤器，只有 BGP 在收发路由的时候才能使用。

团体属性包括基本 basic 团体属性和 extended 团体属性
自定义团体属性和公认团体属性均属于 basic 团体属性。
MPLS VPN 中的 RT 和 SOO 均属于 extended 团体属性。 SOO (site of origin)

路由策略

主要实现了路由过滤和路由属性设置等功能，它通过改变路由

属性（包括可达性）来改变网络流量所经过的路径。主要通过 route-policy 实现



路由策略常用于如下场景：
控制路由的引入：在对路由相互引入，为防止次优路径或者环路，可以使用路由策略加以解决。
控制路由的接收和发布： 根据网络需求，接收或者发布特定的路由。
设置特定路由的属性：可以通过路由策略修改路由的属性，以对网络进行优化、调整。

路由策略原理
一个 Route-Policy 由多个节点构成，路由进入路由策略后，按节点序号从小到大依次检查各个节点是否匹配。一个节点包括多个 if-match 和 apply 子句。if-match 子句用来定义该节点的匹配条件，apply 子句用来定义通过过滤的路由行为。
if-match 子句的过滤规则关系是“与”，即该节点的所有 if-match 子句都必须匹配。
Route-Policy 节点间的过滤关系是“或”，即只要通过了一个节点的过滤，就可通过该 Route-Policy。如果没有通过任何一个节点的过滤，路由信息将无法通过该 Route-Policy。

对于同一个 Route-Policy 节点，在匹配的过程中，各个 if-match 子句间是“与”的关系，即路由信息必须同时满足所有匹配条件，才可以执行 apply 子句的动作。但命令 if-match route-type 和 if-match interface 除外，这两个命令的各自 if-match 子句间是“或”的关系，与其它命令的 if-match 子句间仍是“与”的关系。

Filter-policy Import (OSPF)

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。
OSPF 的路由信息记录在 LSDB 中，filter-policy import 命令实际上是对 OSPF 计算出来的路由进行过滤，不是对发布和接收的 LSA 进行过滤。

Filter-policy Export

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、7 LSA 并发布出去。

Filter-policy Import (IS-IS)

IS-IS 的路由表项需要被成功下发到 IP 路由表中，才能用来指导 IP 报文转发。如果 IS-IS 路由表中有到达某个目的网段的路由，但是并不希望将该路由下发到 IP 路由表中，可以使用该命令结合基本 ACL、IP-Prefix、路由策略等方式，只将部分 IS-IS 路由下发到 IP 路由表中。
Filter-policy import 命令用来配置 IS-IS 路由加入 IP 路由表时的过滤策略。配置该命令后，不会影响本地设备的 LSP 的扩散和 LSDB 的同步，只会影响本地的 IP 路由表。

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了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。
Filter-policy export 命令用来配置 IS-IS 对已引入的路由在向外发布时进行过滤的过滤策略。配置该命令后，不会影响本地设备的路由，只会将引入的部分外部路由发布给 IS-IS 邻居。

Filter-policy Import (BGP)

该命令可以对 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 命令用于对设备全局发布的路由信息进行过滤。
对于通过 import-route (BGP)命令引入的路由，配置了这个命令后，BGP 会在引入这些路由之前就进行过滤，只有通过过滤的路由才能加入 BGP 本地路由表，并被 BGP 发布。

==========

策略路由

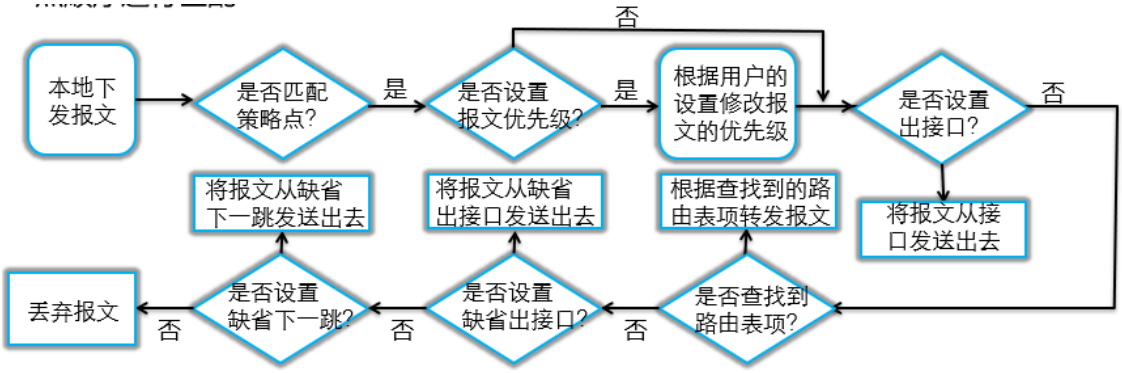
传统的路由转发原理是首先根据报文的目的地址查找路由表，然后进行报文转发。而策略路由使网络管理者不仅能够根据报文的目的地址，而且能够根据报文的源地址、报文大小和链路质量等属性来制定策略路由，以改变数据包转发路径，满足用户需求。
策略路由具有如下优点：

可以根据用户实际需求制定策略进行路由选择，增强路由选择的灵活性和可控性。
可以使不同的数据流通过不同的链路进行发送，提高链路的利用效率。
在满足业务服务质量的前提下，选择费用较低的链路传输业务数据，从而降低企业数据服务的成本。

策略路由与路由策略存在以下不同：
策略路由的操作对象是数据包，在路由表已经产生的情况下，不按照路由表进行转发，而是根据需要，依照某种策略改变数据包转发路径。
路由策略的操作对象是路由信息。路由策略主要实现了路由过滤和路由属性设置等功能，它通过改变路由属性（包括可达性）来改变网络流量所经过的路径。

策略路由可以分为本地策略路由、接口策略路由和智能策略路由。
本地策略路由:本地策略路由仅对本机下发的报文进行处理，对转发的报文不起作用。
接口策略路由:只对转发的报文起作用，对本地下发的报文（比如本地的 Ping 报文）不起作用。
智能策略路由:基于业务需求策略路由，匹配链路质量和网络业务对链路质量需求，实现智能选路。

本地策略路由
本地策略路由是仅对本机下发的报文进行处理的策略路由，对转发的报文不起作用,可配置多个本地策略路由，每个本地策略路由称为一个节点，报文按照本地策略路由节点顺序进行匹配



匹配顺序：

如果找到了匹配的本地策略路由节点，则按照以下步骤发送报文：

查看用户是否设置了报文的优先级: 如果用户设置了报文的优先级，首先根据用户设置的优先级设置报文的优先级，然后继续向下执行;如果用户未设置报文的优先级，则继续向下执行

查看用户是否设置了本地策略路由的出接口: 如果用户设置了出接口，则将报文从出接口发送出去，不再继续执行下面的步骤;如果用户未设置出接口，则继续向下执行

查看用户是否设置了本地策略路由的下一跳（用户可以设置两个下一跳以达到负载分担的目的）如果用户设置了策略路由的下一跳，则将报文发往下一跳，不再继续执行下面的步骤; 如果用户未设置下一跳，则按照正常流程根据报文的目的地址查找路由。如果没有查找到路由，则继续向下执行

查看用户是否设置了本地策略路由的缺省出接口: 如果用户设置了缺省出接口，则将报文从缺省出接口发送出去，不再继续执行下面的步骤;如果用户未设置缺省出接口，则继续执行

查看用户是否设置了本地策略路由的缺省下一跳:如果用户设置了缺省下一跳，则将报文发往缺省下一跳，不再继续执行下面的步骤;如果用户未设置缺省下一跳，则继续执行

丢弃报文，产生 ICMP_UNREACH 消息

如果没有找到匹配的本地策略路由节点，按照发送 IP 报文的一般流程，根据目的地址查找路由

路由策略和策略路由比较

特性	特点
路由策略	1.基于目的地址按路由表转发 2.基于控制平面，为路由协议和路由表服务 3.与路由协议结合完成策略 4.应用命令route-policy
策略路由	1.基于策略的转发，失败后再查找路由表转发 2.基于转发平面，为转发策略服务 3.需要手工逐跳配置，以保证报文按策略转发 4.应用命令policy-based-route

前言

- 在复杂的IP网络中，根据实际组网需求，往往需要实施一些策略，通过改变路由属性（包括可达性）来改变网络流量所经过的路径，实现路由过滤和路由属性设置等功能，如控制路由的接收和发布、控制路由的引入、设置特定路由的属性等等。
- 本文主要介绍网络中常用的路由选择工具、路由策略、策略路由的原理与配置。

路由选择工具

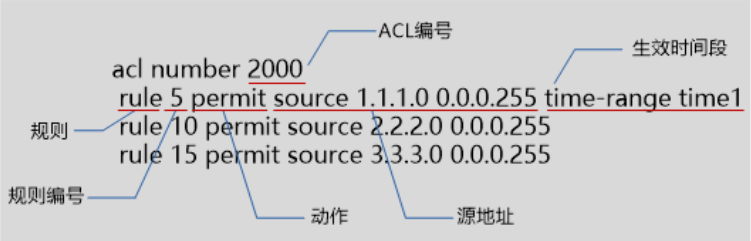
- ACL
  - 匹配报文信息
- IP-Prefix
  - 匹配路由信息
- As-Path-Filter
  - 针对AS路径指定匹配条件
- Community-Filter
  - 针对团体属性指定匹配条件

- ACL 访问控制类表
- ACL 是由 permit 或 deny 语句组成的一系列有顺序规则的集合，它通过匹配报文的信息实现对报文的分类。路由器根据 ACL 定义的规则判断哪些报文可以接收，哪些报文需要拒绝，从而实现对报文的过滤。
- ACL 通过配置的一系列匹配规则对特定的数据包进行过滤，从而识别需要过滤的对象。然后，根据预先设定的策略允许或禁止相应的数据包通过。同时，ACL 可以作为基础配置被其他功能模块引用。
- ip-prefix 前缀类表
- ip-prefix 将路由条目与前缀过滤列表的每一项进行匹配，然后根据定义的匹配模式进行过滤，达成路由筛选的目的。
- ip-prefix 不能用来过滤数据包，只能过滤路由信息。
- as-path-filter AS 路径过滤器
- BGP 的路由信息中，包含一个自治系统路径域。As-path-filter 就是针对自治系统路径域指定匹配条件。AS 路径过滤器仅应用于 BGP 协议。
- community-filter 团体属性过滤器

- BGP 的路由信息中，可以携带一个或多个团体属性，团体属性过滤器就针对团体属性域指定匹配条件。

路由选择工具 - ACL基本原理

- 访问控制列表ACL（Access Control List）是由一条或多条规则组成的集合。所谓规则，是指描述报文匹配条件的判断语句，这些条件可以是报文的源地址、目的地址、端口号等。
- ACL本质上是一种报文过滤器，规则是过滤器的滤芯。设备基于这些规则进行报文匹配，可以过滤出特定的报文，并根据应用ACL的业务模块的处理策略来允许或阻止该报文通过。



- ACL 编号：用于标识 ACL，表明该 ACL 是数字型 ACL。
- 根据 ACL 规则功能的不同，ACL 被划分为基本 ACL、高级 ACL、二层 ACL 和用户 ACL 这几种类型，每类 ACL 编号的取值范围不同。
- 除了可以通过 ACL 编号标识 ACL，设备还支持通过名称来标识 ACL，就像用域名代替 IP 地址一样，更加方便记忆。这种 ACL，称为命名型 ACL。
- 命名型 ACL 实际上是“名字+数字”的形式，可以在定义命名型 ACL 时同时指定 ACL 编号。如果不指定编号，则由系统自动分配。
- 规则：即描述报文匹配条件的判断语句。
- 规则编号：用于标识 ACL 规则。可以自行配置规则编号，也可以由系统自动分配。ACL 规则的编号范围是 0～4294967294，所有规则均按照规则编号从小到大进行排序。所以，图 1 中的 rule 5 排在首位，而规则编号最大的 rule15 排在末位。系统按照规则编号从小到大的顺序，将规则依次与报文匹配，一旦匹配上一条规则即停止匹配。

- 动作：包括 permit/deny 两种动作，表示允许/拒绝。
- 匹配项：ACL 定义了极其丰富的匹配项。除了图中的源地址和生效时间段，ACL 还支持很多其他规则匹配项。例如，二层以太网帧头信息（如源 MAC、目的 MAC、以太帧协议类型）、三层报文信息（如目的地址、协议类型）以及四层报文信息（如 TCP/UDP 端口号）等。
- 如果规则存在，则系统会从 ACL 中编号最小的规则开始查找。如果匹配上了 permit 规则，则停止查找规则，并返回 ACL 匹配结果为：匹配（允许）。如果匹配上了 deny 规则，则停止查找规则，并返回 ACL 匹配结果为：匹配（拒绝）。如果未匹配上规则，则继续查找下一条规则，以此循环。如果一直查到最后一条规则，报文仍未匹配上，则返回 ACL 匹配结果为：不匹配。

路由选择工具 - ACL分类

- ACL类型根据不同的划分规则可以有不同的分类
  - 基于ACL标识方法的划分
    - 数字型ACL
    - 命名型ACL
  - 基于ACL规则定义方式的划分
    - 基本ACL（编号范围2000-2999）
    - 高级ACL（编号范围3000-3999）
    - 二层ACL（编号范围4000-4999）
    - 用户ACL（编号范围6000-6031）

- 基本 ACL
- 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
- 既可使用 IPv4 报文的源 IP 地址，也可使用目的 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
- 使用报文的以太网帧头信息来定义规则，如根据源 MAC（Media Access Control）地址、目的 MAC 地址、二层协议类型等。
- 用户 ACL
- 既可使用 IPv4 报文的源 IP 地址，也可使用目的 IP 地址、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。

- 除此之外，还有基于 IPv6 的 ACL，基本 ACL6 和高级 ACL6。
- 基本 ACL6：可使用 IPv6 报文的源 IPv6 地址、分片信息和生效时间段来定义规则。
- 高级 ACL6：可以使用 IPv6 报文的源 IPv6 地址、目的 IPv6 地址、IPv6 协议类型、ICMPv6 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。

路由选择工具 - ACL匹配顺序

- 配置顺序（config）
  - 按ACL规则的编号（rule-id）按照从小到大的顺序进行匹配
- 自动排序（auto）
  - 使用“深度优先”的原则进行匹配，常用ACL的匹配顺序如下：

ACL类型	匹配原则
基本ACL	1.带VPN实例的规则优先 2.源IP地址范围小的优先 3.rule ID小的优先
高级ACL	1.带VPN实例的规则优先 2.指定了IP 协议承载的协议类型的规则优先 3.源IP地址范围小的优先 4.目的IP地址范围小的优先 5.四层端口号范围小的规则优先 6. rule-id 小的优

- ACL 的匹配顺序
- 一个 ACL 可以由多条“deny | permit”语句组成，每一条语句描述一条规则，这些规则可能存在重复或矛盾的地方（一条规则可以包含另一条规则，但两条规则不可能完全相同）。
- 设备支持两种匹配顺序，即配置顺序（config）和自动排序（auto）。当将一个数据包和访问控制列表的规则进行匹配的时候，由规则的匹配顺序决定规则的优先级，ACL 通过设置规则的优先级来处理规则之间重复或矛盾的情形。缺省的 ACL 匹配顺序是 config 模式。
- 配置顺序
- 如果配置规则时指定了规则编号，则规则编号越小，规则插入位置越靠前，该规则越先被匹配。
- 如果配置规则时未指定规则编号，则由系统自动为其分配一个编号。该编号是一个大于当前 ACL 内最大规则编号且是步长整数倍的最小整数，因此该规则会被最后匹配。
- 自动排序（auto）
- 自动排序（auto）使用“深度优先”的原则进行匹配。
- “深度优先”即根据规则的精确度排序，匹配条件（如协议

类型、源和目的 IP 地址范围等）限制越严格越精确。例如可以比较地址的反掩码，反掩码越小，则指定的主机的范围就越小，限制就越严格。

- 若“深度优先”的顺序相同，则匹配该规则时按 rule-id 从小到大排列。

IPv6 ACL

- IPv6 ACL对根据配置的规则对IPv6报文进行分类，其实现原理和ACL基本相同。
- IPv6 ACL简称ACL6。

- ACL6的分类

分类	对应编号范围	应用场景
基本ACL6	编号范围为2000～2999。	可以使用报文的源IPv6地址、VPN（Virtual Private Network）实例、分片标记和时间段信息来定义规则。
高级ACL6	编号范围为3000～3999。	可以使用报文的源IPv6地址、目的IPv6地址、IPv6承载的协议类型、针对协议的特性（例如TCP的源端口、目的端口和ICMPv6协议的类型、ICMPv6 Code）等内容定义规则。

- ACL6 和 ACL 命令行不同，而对应的编号可以相同，二者互不影响。
- 例如：
- [RouterA] acl ipv6 number 3001
- [RouterA-acl6-adv-3001] rule deny ipv6 source 3001::2/64
- [RouterA] acl 3001
- [Router-acl-adv-3001] rule permit ip source 202.169.10.5 0.0.0.0

路由选择工具 - IP-Prefix

- IP-Prefix
  - 用于过滤IP前缀，能同时匹配前缀号和掩码长度
  - 不能用于数据包过滤
  - 缺省情况下，存在最后一条默认匹配模式为deny
  - 当引用的前缀过滤列表不存在时，默认匹配模式为permit

- ip-prefix
- 每个地址前缀列表可以包含多个 IP-Prefix 条目，每个 IP-Prefix 条目对应一个索引号（index）。路由将按照索引号从小到大依次检查 IP-Prefix 列表，任意一个 index 匹配成功，将不再检查其余项。若所有 index 都匹配失败，路由信息将被过滤。
- 根据匹配的前缀不同，前缀过滤列表可以进行精确匹配，也可以进行在一定掩码长度范围内匹配。
- 前缀过滤列表可以进行精确匹配或者在一定掩码长度范围内匹配，可以通过配置关键字 greater-equal 和 less-equal 指定待匹配的前缀掩码长度范围。如果没有配置关键字 greater-equal 或 less-equal，前缀过滤列表进行精确匹配，即只匹配掩码长度为前缀过滤列表掩码长度的相同 IP 地址路由；如果只配置了关键字 greater-equal，则待匹配的掩码长度范围为从 greater-equal 指定值到 32 位长度；如果只匹配了关键字 less-equal，则待匹配的掩码长度范围为从指定的掩码到关键字 less-equal 指定值。
- greater-equal-value 与 less-equal-value 的取值限制：mask-length<=greater-equal-value<=less-equal-value<=32。

- ip-prefix 特点
- 当所有前缀过滤列表均未匹配时，缺省情况下，存在最后一条默认匹配模式为 deny。
- 当引用的前缀过滤列表不存在时，默认匹配模式为 permit。

路由选择工具 - IP-Prefix示例

- ip ip-prefix FILTER index 10 permit 1.1.1.0 24
  - 该ip-prefix为精确匹配，只有1.1.1.0/24才能permit
- ip ip-prefix FILTER index 10 permit 1.1.1.0 24 less-equal 32
  - 掩码范围在24-32之间的网络1.1.1.0才能permit
- ip ip-prefix FILTER index 10 permit 1.1.1.0 24 greater-equal 26
  - 掩码范围在26-32之间的网络1.1.1.0才能permit
- ip ip-prefix FILTER index 10 permit 1.1.1.0 24 greater-equal 26 less-equal 32
  - 掩码范围在26-32之间的网络1.1.1.0才能permit
- ip ip-prefix FILTER index 10 permit 0.0.0.0 0 greater-equal 8 less-equal 32
  - 所有掩码长度在8到32的路由都被permit
- ip ip-prefix FILTER index 20 permit 0.0.0.0 0 less-equal 32
  - 所有路由均被permit

IPv6-Prefix

- IPv6-Prefix根据配置的规则对IPv6报文进行分类，其实现原理和IP-Prefix基本相同。
- IPv6地址前缀列表用于过滤IPv6地址。同一个地址前缀列表可包含多个表项，每个表项指定一个地址前缀范围。此时，各表项之间是“或”的关系，即只要通过其中一个表项就认为已通过该地址前缀列表的过滤，所有表项都没有通过则意味着没有通过该地址前缀列表的过滤。
- 默认所有未匹配的路由将被拒绝通过过滤列表。如果所有表项都配置成deny模式，则任何路由都不能通过该过滤列表。因此，需要在多条deny模式的表项后定义一条permit :: 0 less-equal 128的表项，以允许其它所有IPv6路由信息通过。

IPv6-Prefix示例

- 允许掩码长度在32位到64位之间的地址通过。
  - [Huawei] ip ipv6-prefix abc permit :: 0 greater-equal 32 less-equal 64
- 拒绝前缀为3FFE:D00::/32，且前缀长度大于32位的地址通过，允许其他的IPv6路由通过。
  - [Huawei] ip ipv6-prefix abc deny 3FFE:D00:: 32 less-equal 128
  - [Huawei] ip ipv6-prefix abc permit :: 0 less-equal 128
- 配置名为p3的地址前缀列表，拒绝::1～::FFFF:FFFF范围内的所有路由通过，允许其他路由通过。
  - [Huawei] ip ipv6-prefix p3 index 10 deny :: 96 match-network
  - [Huawei] ip ipv6-prefix p3 index 20 permit :: 0 less-equal 128

路由选择工具 - AS-Path-Filter

- AS-Path-Filter
  - 以BGP中的AS_Path属性为匹配条件
  - 使用正则表达式进行定义
- 示例
  - ip as-path-filter 10 permit .*
    - 匹配所有AS-PATH属性
  - ip as-path-filter 10 permit _100$
    - 匹配从AS100发起的路由
  - ip as-path-filter 10 permit ^100_
    - 匹配从AS100接收的路
  - ip as-path-filter 10 permit _100|200$
    - 匹配从AS100或200发起的路由

- AS 路径过滤器是将 BGP 中的 AS_Path 属性作为匹配条件的过滤器，只有 BGP 在收发路由的时候才能使用。

- 由于 AS_Path 属性记录 AS 号是将最后经历的 AS 号放在 AS_Path 记录中的最左侧，所以在配置 as-path-filter 时需要格外注意。
- 如果一条路由起源于 AS100，然后依次经过 AS300, AS200, AS500，最后到达 AS600。那么在 AS600 里，路由的 A

S-PATH 属性表示为（500 200 300 100）。

路由选择工具 - 常用的正则表达式

表达式	含义
^$	表示本地AS始发
.*	表示所有
_10_	表示必须通过AS10
^10_	表示只接受来自AS10的路由
_10$	从AS10始发的所有路由

路由选择工具 - Community-Filter

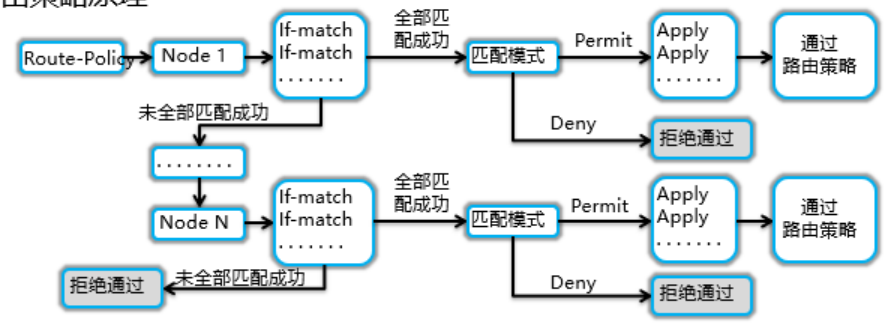
- Community-Filter
  - 以BGP中的community属性为匹配条件
- 示例
  - ip community-filter 1 permit 100:1
    - 匹配community属性为100:1
  - ip community-filter 1 permit no-export
    - 匹配community属性为no-export

- 团体属性过滤器是将 BGP 中的团体属性作为匹配条件的过滤器，只有 BGP 在收发路由的时候才能使用。

- 团体属性包括基本 basic 团体属性和 extended 团体属性
- 自定义团体属性和公认团体属性均属于 basic 团体属性。
- MPLS VPN 中的 RT 和 SOO 均属于 extended 团体属性。

路由策略

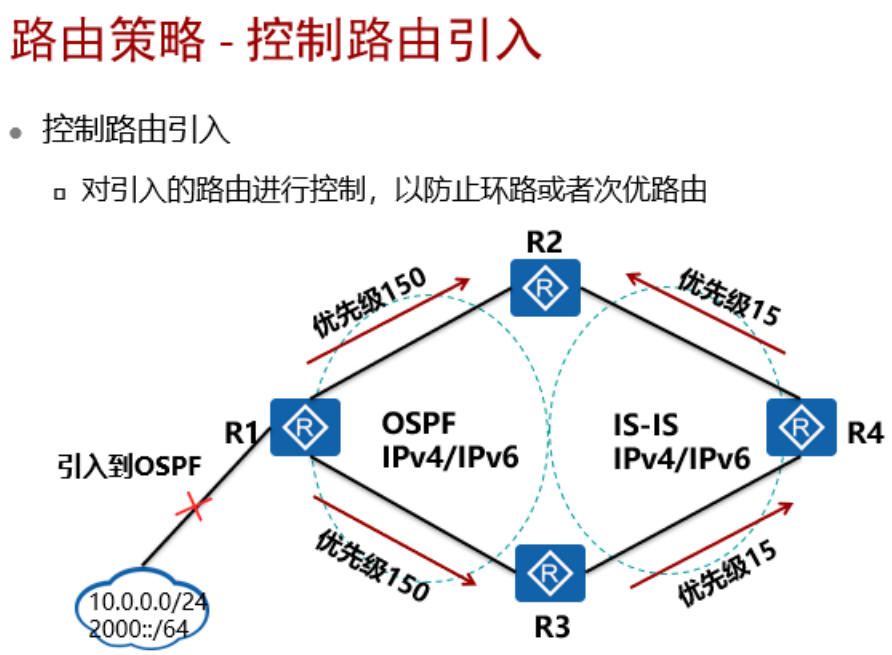
- 路由策略
  - 主要用于路由过滤和路由属性设置等，从而影响流量所经过的路径。
  - 主要通过route-policy实现
- 路由策略原理



- 路由策略主要实现了路由过滤和路由属性设置等功能，它通过改变路由属性（包括可达性）来改变网络流量所经过的路径。
- 路由策略常用于如下场景：
- 控制路由的引入：
- 在对路由做相互引入时，为了防止次优路径或者环路，可以使用路由策略加以解决。
- 控制路由的接收和发布：
- 根据网络需求，接收或者发布特定的路由。
- 设置特定路由的属性：
- 可以通过路由策略修改路由的属性，以对网络进行优化、调整。
- 路由策略原理
- 一个 Route-Policy 由多个节点构成，路由进入路由策略后，按节点序号从小到大依次检查各个节点是否匹配。一个节点包括多个 if-match 和 apply 子句。if-match 子句用来定义该节点的匹配条件，apply 子句用来定义通过过滤的路由行为。if-match 子句的过滤规则关系是“与”，即该节点的所有 if-match 子句都必须匹配。Route-Policy 节点间的过滤关系是“或”，即

只要通过了一个节点的过滤，就可通过该 Route-Policy。如果没有通过任何一个节点的过滤，路由信息将无法通过该 Route-Policy。

- 对于同一个 Route-Policy 节点，在匹配的过程中，各个 if-match 子句间是“与”的关系，即路由信息必须同时满足所有匹配条件，才可以执行 apply 子句的动作。但命令 if-match route-type 和 if-match interface 除外，这两个命令的各自 if-match 子句间是“或”的关系，与其它命令的 if-match 子句间仍是“与”的关系。



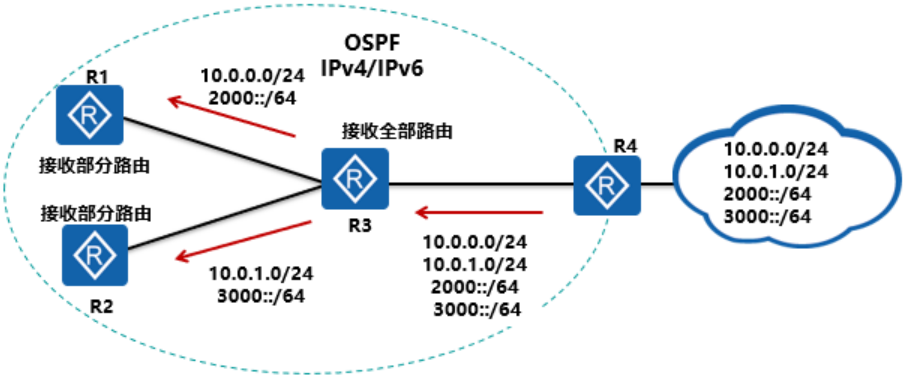
- 上面拓扑为双点双向重发布的示意图，如果不加控制，将会发生次优路由和环路等故障，产生故障的过程具体分析如下：
- 在拓扑中，R1 将网段 10.0.0.0/24 及 2000::/64 引入到 OSPF。R2/R3 分别将该路由引入 ISIS，正常情况下，R2/R3 将路由引入 ISIS 会有先后，假设 R3 先于 R2 将该路由引入到 IS-IS。那么 R2 就会同时从 OSPF 和 IS-IS 学到 10.0.0.0/24 及 2000::/64 的路由。于是 R2 会根据路由协议的优先级优选

通过 IS-IS 学到的路由（OSPF 的外部路由优先级为 150，IS-IS 的路由优先级为 15）。于是，当 R2 访问 10.0.0.0/24 及 2000::/64 网段时，会使用 R4-R3-R1 这条次优路径。为了避免出现这种情况，可以在路由器 R2 上使用 route-policy 修改 OSPF ASE 路由的优先级，使得 OSPF ASE 的路由优先于从 IS-IS 学到的路由，使得 R2 选择正确的路由。

- 当 R1 连接网络 10.0.0.0/24 及 2000::/64 的接口断掉之后，虽然该条外部 LSA 会在 OSPF 区域内老化，但是由于 R2 通过 IS-IS 已经学习到了 10.0.0.0/24 及 2000::/64 网络，所以 R2 会将该网络引入到 OSPF，进而使 R1 和 R3 会通过 R2 学习到了网络 10.0.0.0/24 及 2000::/64。当 R2 访问访问网络 10.0.0.0/24 及 2000::/64 时，流量会沿 R4-R3-R1-R2 的路径发送，从而引发环路。为避免这种情况，我们可以通过给路由条目添加标记 tag 标签，然后对特定标签进行过滤的方法来避免环路的发生。

路由策略 - 控制路由的接收和发布

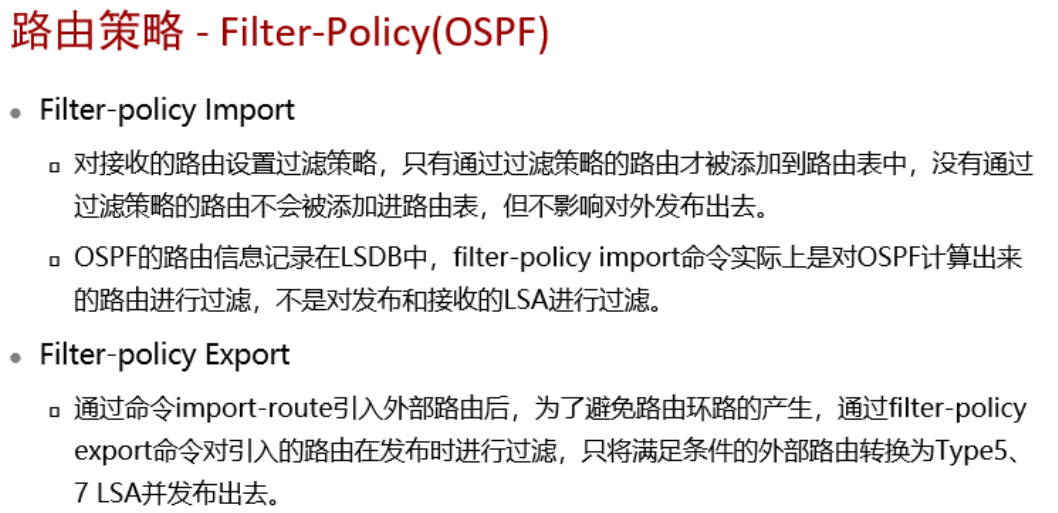
- 控制路由的接收和发布
  - 可以接收指定的路由，从而优化路由表项
  - 一般通过filter-policy实现



- 控制路由的接收和发布
- 只接收必要、合法的路由信息，以控制路由表的容量，

提高网络的安全性。

- 拓扑解释
- R4 将网络 10.0.X.0/24 及 2000::/64&3000::/64 引入到 OSPF 中。根据业务需要，R1 只能接收网络 10.0.0.0/24 和 2000::/64；而 R2 只能接收网络 10.0.1.0/24 和 3000::/64。对于这种需求，可以通过 filter-policy 加以实现。



- Filter-policy import 命令用来按照过滤策略，设置 OSPF 对接收的路由进行过滤。
- Filter-policy export 命令用来按照过滤策略，设置对引入的路由在向外发布时进行过滤。
- 通过指定 protocol 或 process-id 对特定的某一种协议或某一进程的路由进行过滤。如果没有指定 protocol 和 process-id，则 OSPF 将对所有引入的路由信息进行过滤。
- 由于 Type5、7 LSA 是有 ABSR 产生的，因此，本命令仅在 ASBR 上配置。

路由策略 - Filter-Policy(IS-IS)

- Filter-policy Import
  - IS-IS的路由表项需要被成功下发到IP路由表中，才能用来指导IP报文转发。如果IS-IS路由表中有到达某个目的网段的路由，但是并不希望将该路由下发到IP路由表中，可以使用该命令结合基本ACL、IP-Prefix、路由策略等方式，只将部分IS-IS路由下发到IP路由表中。
- 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了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。

- Filter-policy import 命令用来配置 IS-IS 路由加入 IP 路由表时的过滤策略。
- 配置该命令后，不会影响本地设备的 LSP 的扩散和 LSDB 的同步，只会影响本地的 IP 路由表。
- Filter-policy export 命令用来配置 IS-IS 对已引入的路由在向外发布时进行过滤的过滤策略。
- 配置该命令后，不会影响本地设备的路由，只会将引入的部分外部路由发布给 IS-IS 邻居。

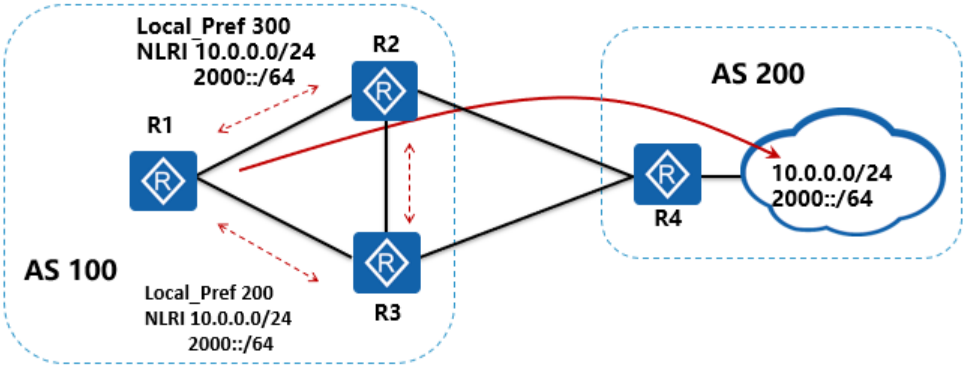
路由策略 - Filter-Policy(BGP)

- Filter-policy Import
  - 该命令可以对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
  - filter-policy export命令用于对设备全局发布的路由信息进行过滤。
  - 对于通过import-route (BGP)命令引入的路由，配置了这个命令后，BGP会在引入这些路由之前就进行过滤，只有通过过滤的路由才能加入BGP本地路由表，并被BGP发布。

- filter-policy import 命令用来配置对接收的路由信息进行过滤。
- filter-policy export 命令用来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。
- 如果指定 protocol 参数，将只对引入的这种协议产生的路由进行过滤，对引入的其他协议产生的路由不受影响。如果没有指定 protocol 参数，对引入的任何一个协议产生的路由都要进行过滤。

路由策略 - 设置特定路由的属性

- 设置特定路由的属性
  - 使用路由策略为特定的路由设置相应的属性



- 拓扑描述
- 通过 route-policy 修改 BGP 中的 Local_Pref 属性，进而影响流量走向。R2 将从 EBGP 学到的 10.0.0.0/24 和 2000::/64 路由的 Local_Pref 设为 300，R3 将从 EBGP 学到的该路由的 Local_Pref 设置为 200，R1/R2/R3 相互之间通过 IBGP 交互各自的路由，最终将选择以 R2 作为本 AS 去往网络 10.0.0.0/24 和 2000::/64 的出口。

策略路由

- 传统的路由转发原理是首先根据报文的目的地址查找路由表，然后进行报文转发。而策略路由使网络管理者不仅能够根据报文的目的地址，而且能够根据报文的源地址、报文大小和链路质量等属性来制定策略路由，以改变数据包转发路径，满足用户需求。
- 策略路由具有如下优点：
  - 可以根据用户实际需求制定策略进行路由选择，增强路由选择的灵活性和可控性。
  - 可以使不同的数据流通过不同的链路进行发送，提高链路的利用效率。
  - 在满足业务服务质量的前提下，选择费用较低的链路传输业务数据，从而降低企业数据服务的成本。

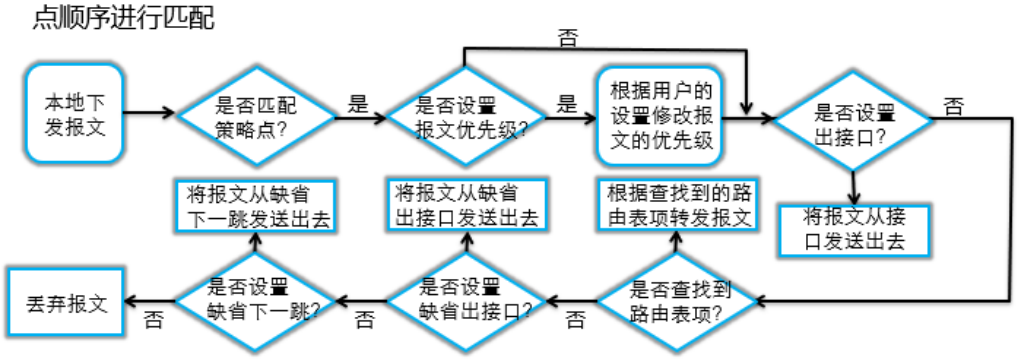
- 策略路由与路由策略存在以下不同：
- 策略路由的操作对象是数据包，在路由表已经产生的情况下，不按照路由表进行转发，而是根据需要，依照某种策略改变数据包转发路径。
- 路由策略的操作对象是路由信息。路由策略主要实现了路由过滤和路由属性设置等功能，它通过改变路由属性（包括可达性）来改变网络流量所经过的路径。

策略路由的分类

- 策略路由可以分为本地策略路由、接口策略路由和智能策略路由。
- 本地策略路由
  - 本地策略路由仅对本机下发的报文进行处理，对转发的报文不起作用。
- 接口策略路由
  - 接口策略路由只对转发的报文起作用，对本地下发的报文（比如本地的Ping报文）不起作用。
- 智能策略路由
  - 智能策略路由是基于业务需求的策略路由，通过匹配链路质量和网络业务对链路质量的需求，实现智能选路。

策略路由 - 本地策略路由

- 本地策略路由
  - 本地策略路由是仅对本机下发的报文进行处理的策略路由，对转发的报文不起作用
  - 可配置多个本地策略路由，每个本地策略路由称为一个节点，报文按照本地策略路由节点顺序进行匹配



- 匹配顺序：
- 如果找到了匹配的本地策略路由节点，则按照以下步骤发送报文：
- 查看用户是否设置了报文的优先级: 如果用户设置了报文的优先级，首先根据用户设置的优先级设置报文的优先级，然后继续向下执行;如果用户未设置报文的优先级，则继续向下执行
- 查看用户是否设置了本地策略路由的出接口: 如果用户设置了出接口，则将报文从出接口发送出去，不再继续执行下面的步骤;如果用户未设置出接口，则继续向下执行
- 查看用户是否设置了本地策略路由的下一跳（用户可以设置两个下一跳以达到负载分担的目的）如果用户设置了策略路由的下一跳，则将报文发往下一跳，不再继续执行下面的步骤; 如果用户未设置下一跳，则按照正常流程根据报文的目的地址查找路由。如果没有查找到路由，则继续向下执行
- 查看用户是否设置了本地策略路由的缺省出接口: 如果用户设置了缺省出接口，则将报文从缺省出接口发送出去，不再继续执行下面的步骤;如果用户未设置缺省出接口，则继续执行

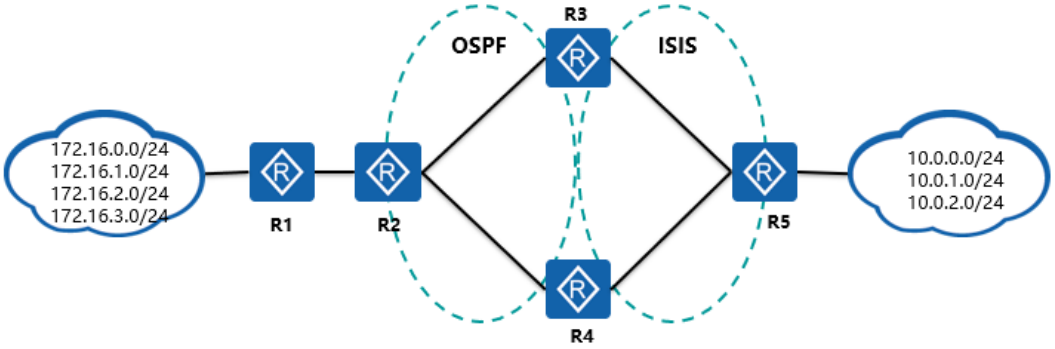
- 查看用户是否设置了本地策略路由的缺省下一跳:如果用户设置了缺省下一跳，则将报文发往缺省下一跳，不再继续执行下面的步骤;如果用户未设置缺省下一跳，则继续执行
- 丢弃报文，产生 ICMP_UNREACH 消息
- 如果没有找到匹配的本地策略路由节点，按照发送 IP 报文的一般流程，根据目的地址查找路由

路由策略和策略路由比较

特性	特点
路由策略	1.基于目的地址按路由表转发 2.基于控制平面，为路由协议和路由表服务 3.与路由协议结合完成策略 4.应用命令route-policy
策略路由	1.基于策略的转发，失败后再查找路由表转发 2.基于转发平面，为转发策略服务 3.需要手工逐跳配置，以保证报文按策略转发 4.应用命令policy-based-route

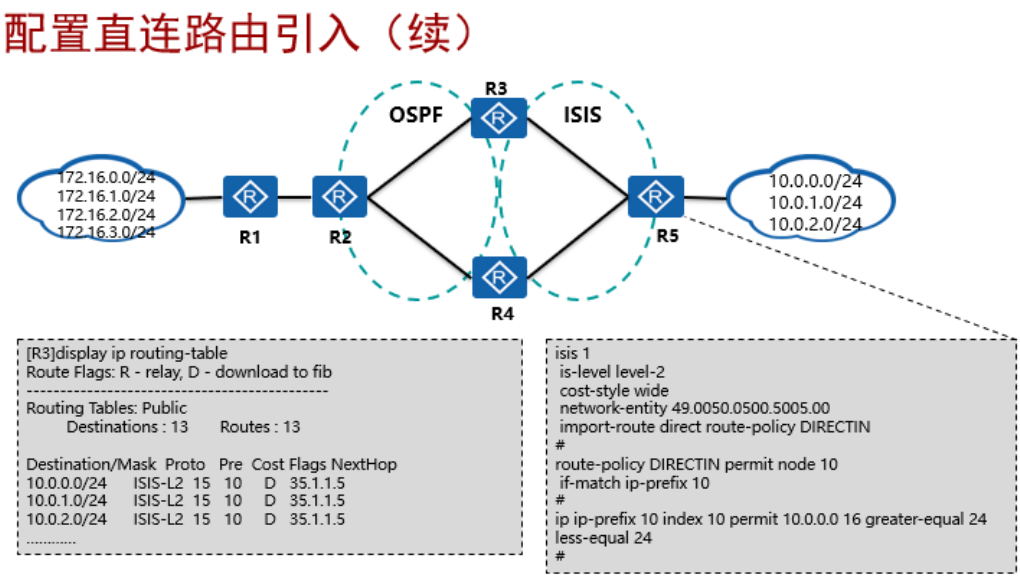
配置直连路由引入

- 公司A网络拓扑如下所示，现根据需求完成如下配置：
  - R2、R3和R4运行OSPF，R3、R4和R5运行ISIS；
  - 将与R5直连的网段10.0.X.0/24引入到ISIS。



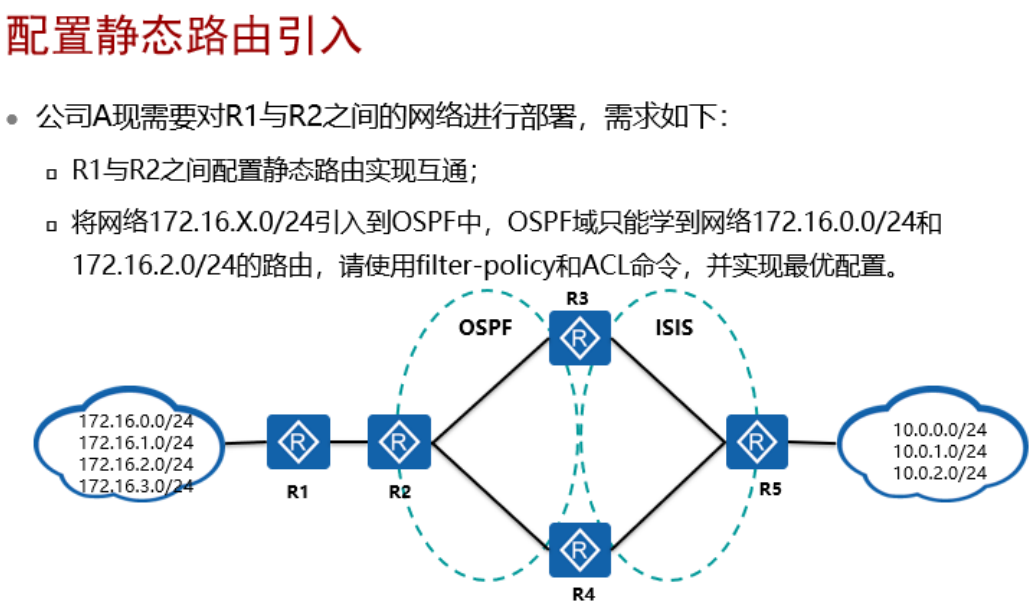
- 本案例中设备互联地址规则如下：
- 如 RTX 与 RTY 互联，则互联地址为 XY.1.1.X 与 XY.1.1.

Y，掩码长度为 24 位。

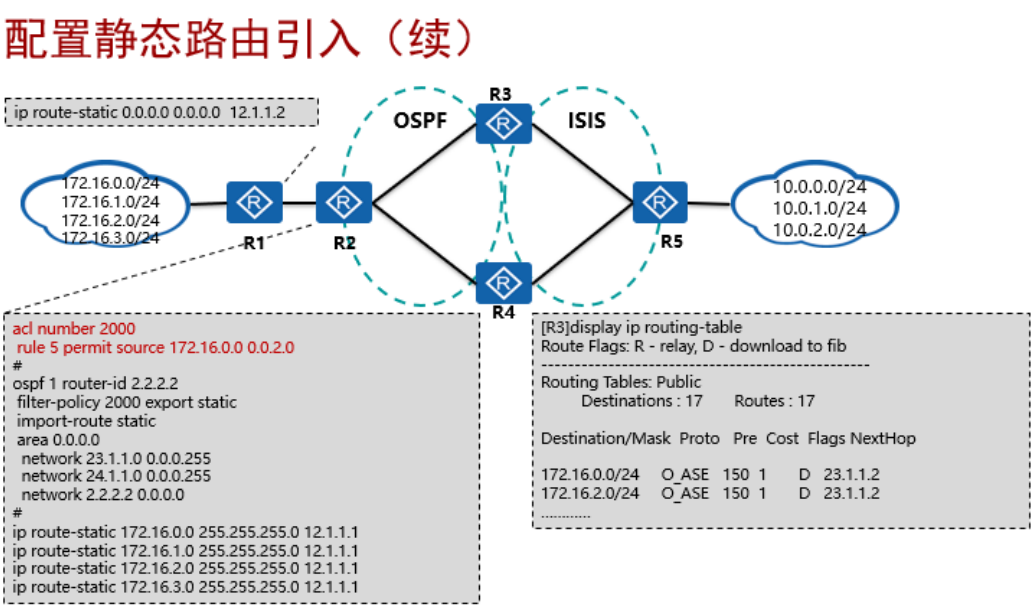


- 命令含义
- route-policy 命令用来创建 Route-Policy 并进入该 Route-Policy 视图。
- 参数意义
- route-policy route-policy-name { permit | deny } node node
- route-policy-name：指定 Route-Policy 名称。
- permit：指定 Route-Policy 节点的匹配模式为允许。如果路由匹配所有的 if-match 子句，该路由可通过过滤并执行此节点 apply 命令中规定的一系列动作；否则，必须进行下一节点的测试。
- deny：指定 Route-Policy 节点的匹配模式为拒绝。如果路由匹配所有的 if-match 子句，该路由不能通过过滤从而不能进入下一节点的测试。
- node node：Route-Policy 的节点索引。
- 注意事项
- Route-Policy 用于过滤路由信息以及为通过过滤的路由

信息设置路由属性。一个 Route-Policy 由多个节点构成。一个节点包括多个 if-match 和 apply 子句。if-match 子句用来定义该节点的匹配条件，apply 子句用来定义通过过滤的路由行为。if-match 子句的过滤规则关系是“与”，即该节点的所有 if-match 子句都必须匹配。Route-Policy 节点间的过滤关系是“或”，即只要通过了一个节点的过滤，就可通过该 Route-Policy。如果没有通过任何一个节点的过滤，路由信息将无法通过该 Route-Policy。



- 此处的需求是对之前案例的扩展，在原案例的基础上进行配置。
- 对于该需求，主要考察对 filter-policy 和 ACL 的理解。所谓最优配置，实际上就是使用最少的命令达到要求的效果。

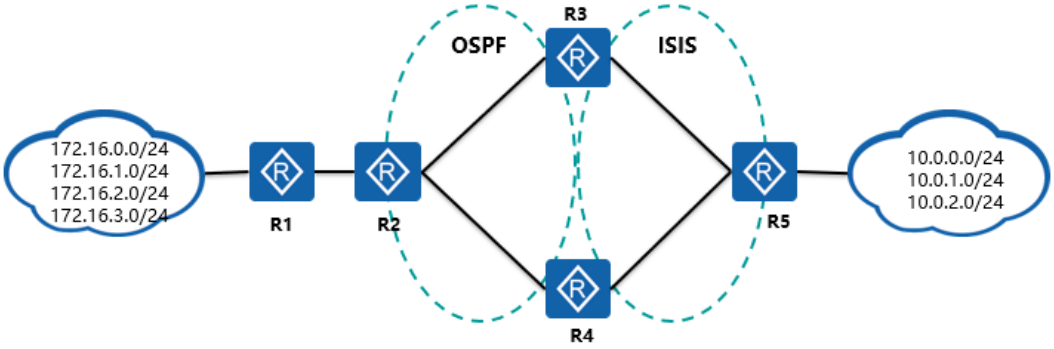


- 命令含义
- filter-policy export 命令用来按照过滤策略，设置对引入的路由在向外发布时进行过滤。
- 参数意义
- filter-policy { acl-number | acl-name acl-name | ip-prefix ip-prefix-name } export [protocol [process-id]]
- acl-number：指定基本访问控制列表号。
- acl-name acl-name：指定访问控制列表名称。
- ip-prefix ip-prefix-name：指定地址前缀列表名称。
- protocol：指定发布路由信息的协议。
- process-id：当发布的路由协议为 rip、isis、ospf 时，可以指定进程号。
- 注意事项
- 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 Type-5 LSA（AS-external-LSA）并发布出去。
- 通过指定 protocol 或 process-id 对特定的某一种协议或

某一进程的路由进行过滤。如果没有指定 protocol 和 process-id，则 OSPF 将对所有引入的路由信息进行过滤。

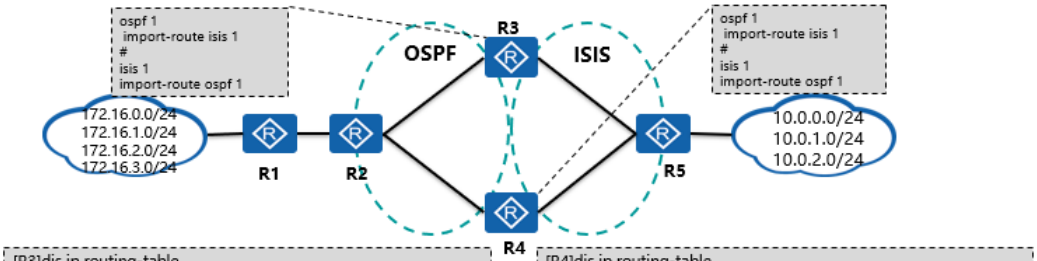
配置双点双向路由引入

- 公司A需要实现网络的互联互通，现需求如下：
  - 在R3和R4上进行双向路由引入



- 该案例拓扑和之前的拓扑一致。我们在完成需求后，要充分考虑到是否存在次优路由，是否发生了环路。

配置双点双向路由引入（续）



[R3]dis ip routing-table
Route Flags: R - relay, D - download to fib

Routing Tables: Public
Destinations : 17 Routes : 17

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
10.0.0.0/24	ISIS-L2	15	10	D	35.1.1.5
10.0.1.0/24	ISIS-L2	15	10	D	35.1.1.5
10.0.2.0/24	ISIS-L2	15	10	D	35.1.1.5
24.1.1.0/24	OSPF	10	2	D	23.1.1.2
45.1.1.0/24	ISIS-L2	15	20	D	35.1.1.5
172.16.0.0/24	O_ASE	150	1	D	23.1.1.2
172.16.2.0/24	O_ASE	150	1	D	23.1.1.2

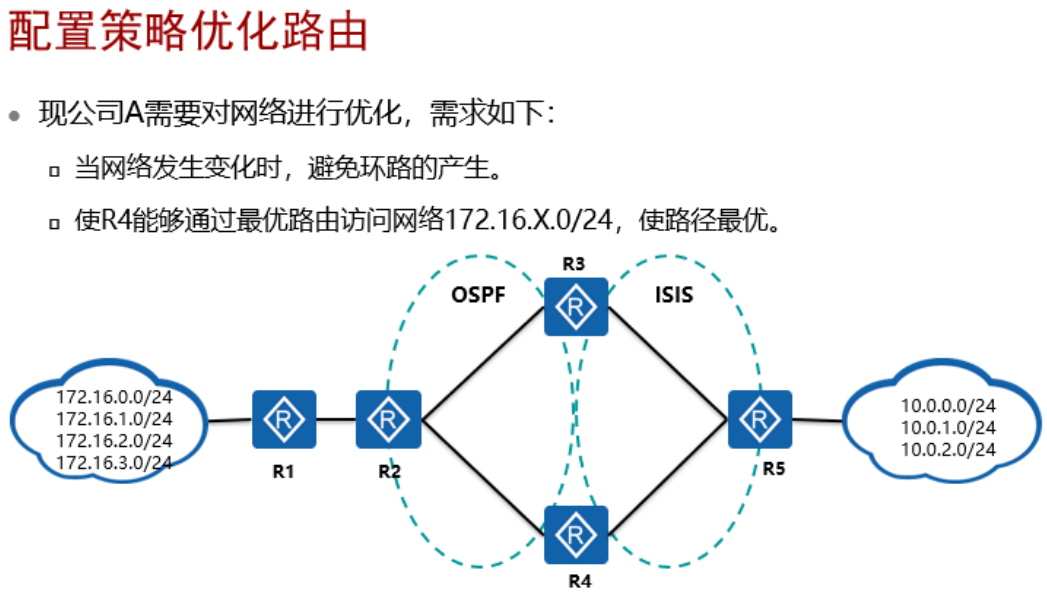
[R4]dis ip routing-table
Route Flags: R - relay, D - download to fib

Routing Tables: Public
Destinations : 17 Routes : 17

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
10.0.0.0/24	ISIS-L2	15	10	D	45.1.1.5
10.0.1.0/24	ISIS-L2	15	10	D	45.1.1.5
10.0.2.0/24	ISIS-L2	15	10	D	45.1.1.5
23.1.1.0/24	OSPF	10	2	D	24.1.1.2
35.1.1.0/24	ISIS-L2	15	20	D	45.1.1.5
172.16.0.0/24	ISIS-L2	15	20	D	45.1.1.5
172.16.2.0/24	ISIS-L2	15	20	D	45.1.1.5

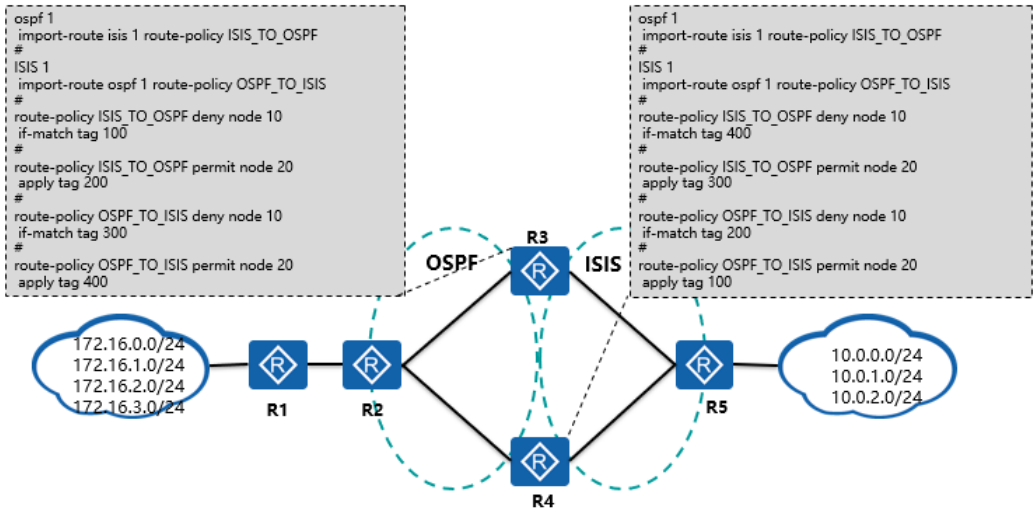
- 通过路由协议的相互引入之后，R4 到达网络 172.16.X.0/24 出现次优路径。产生次优路径主要是因为 R3 将 OSPF 路由先行分发入 ISIS 协议域，于是 R4 从 OSPF 和 ISIS 同时学

到了 172.16.X.0/24 的路由，因为 OSPF 的外部路由的 preference 值为 150，而 ISIS 的 preference 值为 15，所以 R4 选择了 ISIS 路由前往 172.16.X.0/24 网段，于是产生次优路由。



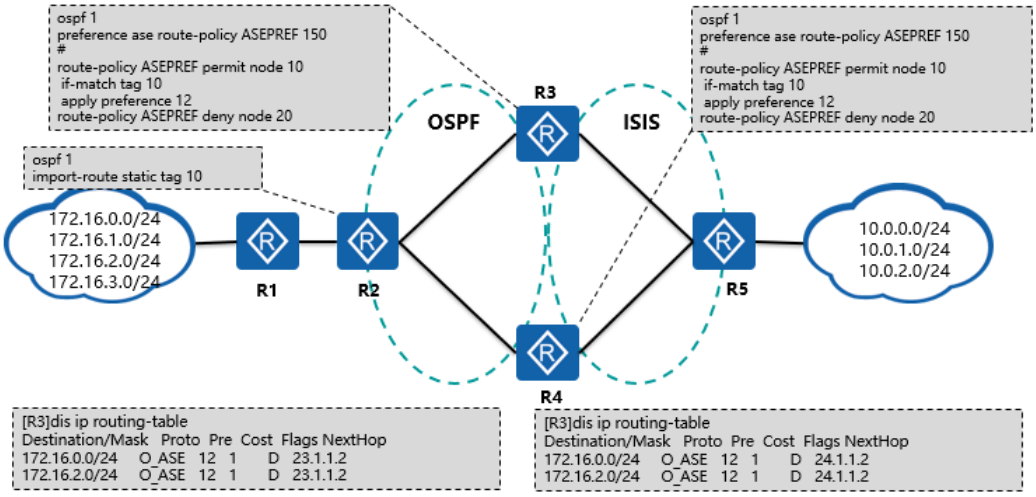
- 本案例中的需求是对之前案例的扩展，在原案例的基础上进行配置。
- 需要修正 R4 访问 172.16.X.0/24 网段的路由，避免出现经过 ISIS 区域的次优路由。
- 可以通过 Tag 实现对双点双向路由引入的控制，从而避免环路。

配置策略优化路由（续）



- 如果在做路由互相引入的时候不做过滤，那么，在网络发生变化的时候，可能会在网络中导致环路。为避免环路就要保证路由协议间相互引入的时候只引入各路由域自身的路由。在上面的配置场景中，使用了 tag 来实现路由相互引入时的限制。使用 tag 的优势是不需要指定具体的路由条目，当路由域内具体的路由项有增减的时候，引入的路由条目和限制会随之变化，不需要手工干预，具有很好的扩展性。
- 虽然上面的配置场景的路由策略能够很好地避免环路，但是并不能解决次优路由的问题。

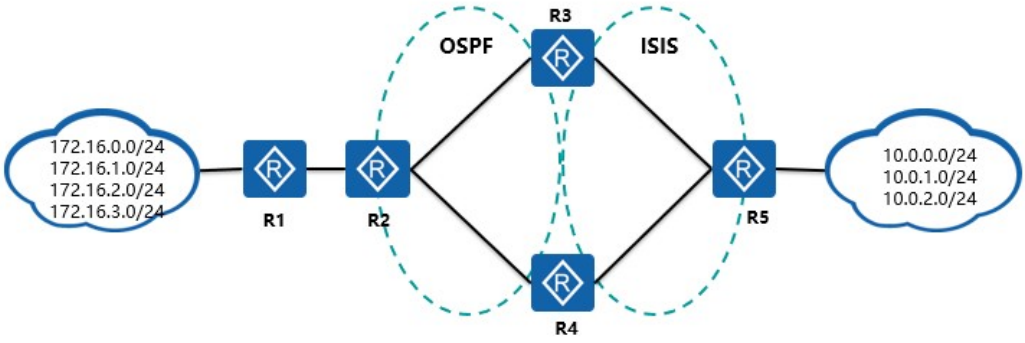
配置策略优化路由（续）



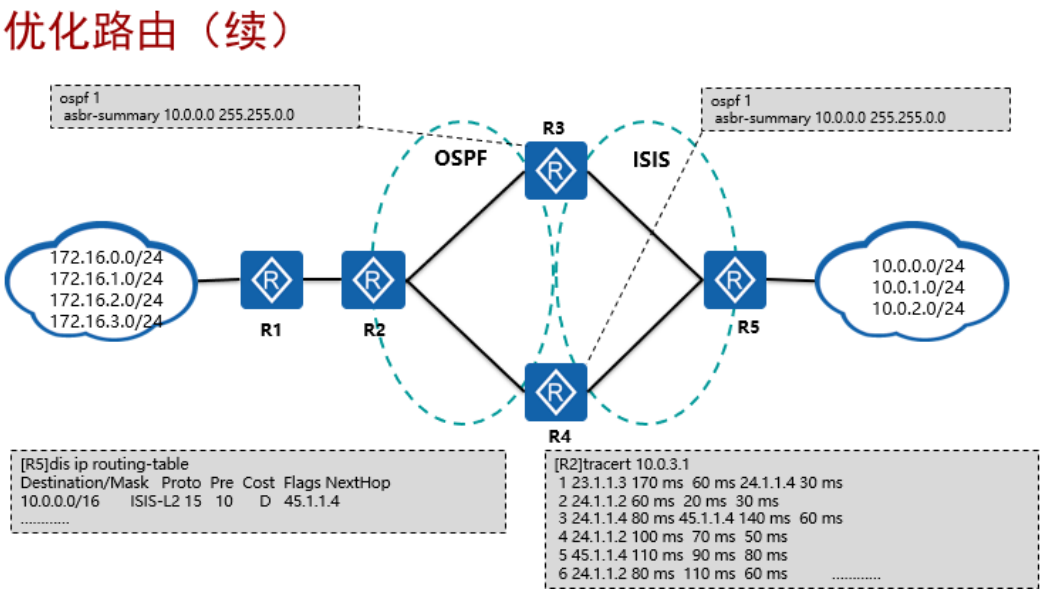
- 次优路由主要是因为双点双向导入时，R3/R4 中的某一台路由器会同时从两侧得到 172.16.X.0/24 路由，而因为 OSPF 的外部路由 preference 值大于 ISIS 的 preference 值（preference 值越小越优先），导致 R3/R4（其中一台）选择了次优路由。要解决此问题，需要修改 OSPF 外部路由条目的 preference 值，只要使 OSPF_ASE 路由的 preference 值小于 ISIS 路由的 preference 值就可以解决此问题。
- 考虑到合理性问题，不建议将 OSPF_ASE 路由的 preference 值设置成比 OSPF 内部路由的 preference 值（10）还小。

优化路由

- 公司A为了优化网络，现需求如下：
  - R3和R4在OSPF域中，将网络10.0.X.0/24汇总为10.0.0.0/16，请充分避免环路。



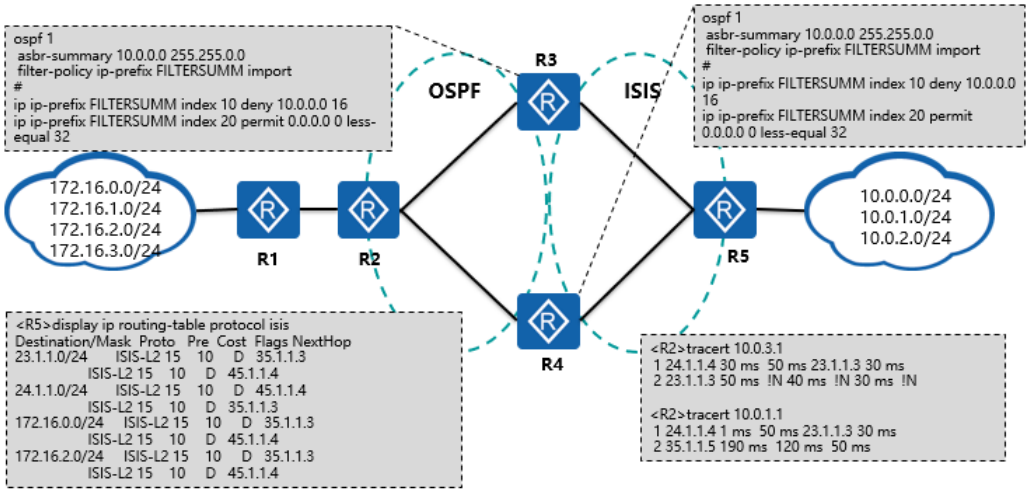
- 本案例中，是对原案例进行的扩展，在原案例的基础上进行配置。



- 当仅仅进行路由汇总，发现存在两个问题。第一个问题，R5 学习到了该汇总路由；第二个问题，在 R2 ping 一个不存在的地址产生了环路。
- 第一个问题产生的原因，主要是由于 R3 和 R4 学习到各自产生的汇总路由后，再引入到 ISIS 域中产生的。此处，R3 首先进行了 OSPF 汇聚配置，传递到 R4，在 ISIS 引入了该路由后，传递到 R3，因此在 R3 可以收到从 R4 过来的 OSPF 外部路由，也可以收到从 R5 过来的 ISIS 路由。因为 ISIS 优先级优于 OSPF 外部路由，所以在 R3 的路由表中可以看到 10.0.0.0/16 路由的下一跳是 35.1.1.5，也就是 R5。然后在 R4 做路由汇聚，路由通过 OSPF 传递到 R3，还是由于优先级的关系，R3 的 ISIS 引入 OSPF 路由时，由于路由表中这条路由是 ISIS 协议产生的，所以引入不到 ISIS 中，这就是 R5 上看到该路由下一跳是 R4 的原因。
- 第二个问题产生的原因，是因为 R2 将 ping 包发给 R4，R4 查路由表又转发给 R2，R2 又给 R4，行成环路。

- 为了解决上面两个问题，我们需要保证 R3 和 R4 既不能学习到对方产生的汇总路由，又不能将该路由引入到 ISIS 路由域。所以，我们只需在 R3 和 R4 上将他们学习到的对方的汇总路由过滤即可。

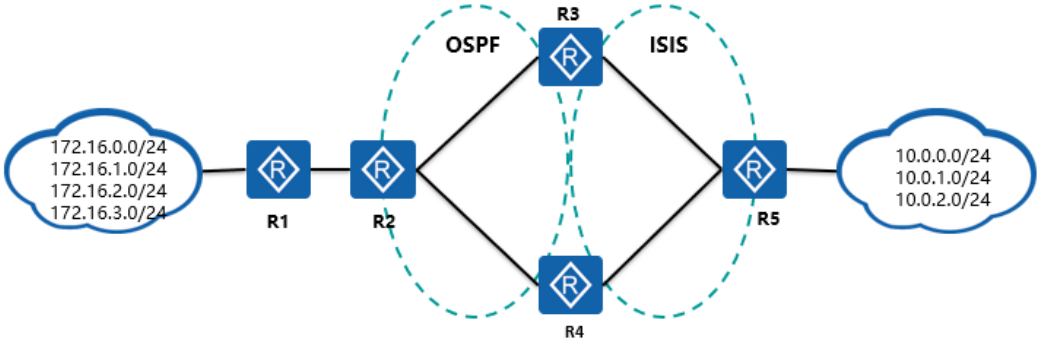
优化路由（续）



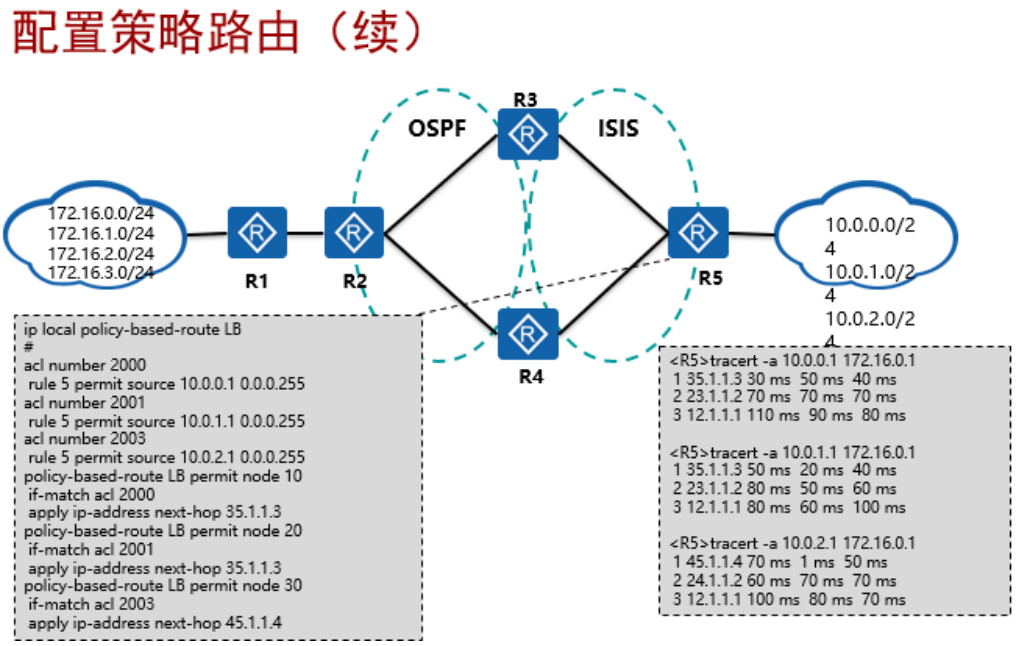
- 在 R3/R4 上增加过滤策略，不从 OSPF 接收特定的汇聚路由。保证该汇聚路由不再重新导入到 ISIS 路由域。避免了环路。

配置策略路由

- 现公司A需要进一步优化网络，现需求如下：
  - 来自10.0.0.0/24和10.0.1.0/24的流量访问网络172.16.X.0/24时经由R3；
  - 来自10.0.2.0/24的流量访问网络172.16.X.0/24时，经由R4。



- 案例中的需求是对之前案例的扩展，在原案例的基础上进行配置。

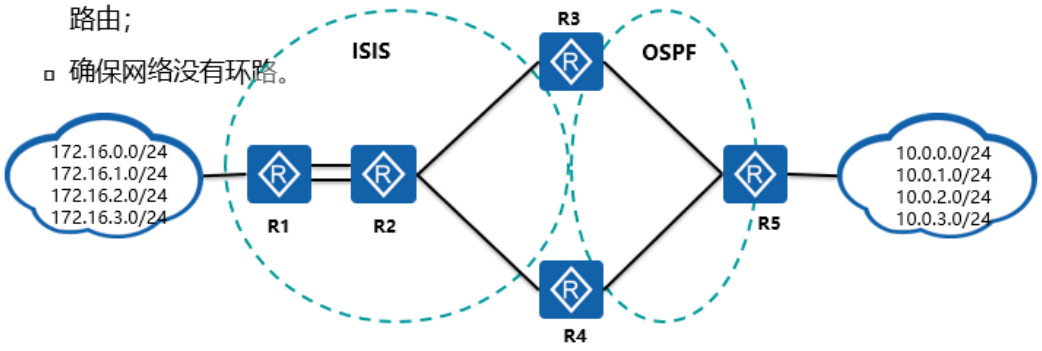


- 命令含义：
- policy-based-route 命令用来创建或修改策略路由和策略点。
- ip local policy-based-route 命令用来使能本地策略路由。
- 参数意义
- policy-based-route policy-name { permit | deny } node node-id
- policy-name：指定策略名称。
- permit：策略点的模式，表示对满足匹配条件的报文进行策略路由。
- deny：策略点的模式，表示对满足匹配条件的报文不进行策略路由。
- node-id：指定策略点的顺序号。
- ip local policy-based-route policy-name
- policy-name：指定策略路由的名称。
- 注意事项

- 在部署策略路由时，如果需要配置报文的出接口，则报文的出接口不能为以太接口等广播型接口。
- 实验结果
- 在 R5 上指定不同的源地址对到达相同目的地的数据包进行跟踪，可以发现数据包选用了不同的路径。注意，ip local policy-based-route 命令应用的策略只对路由器本地发起的数据包起作用。

案例1

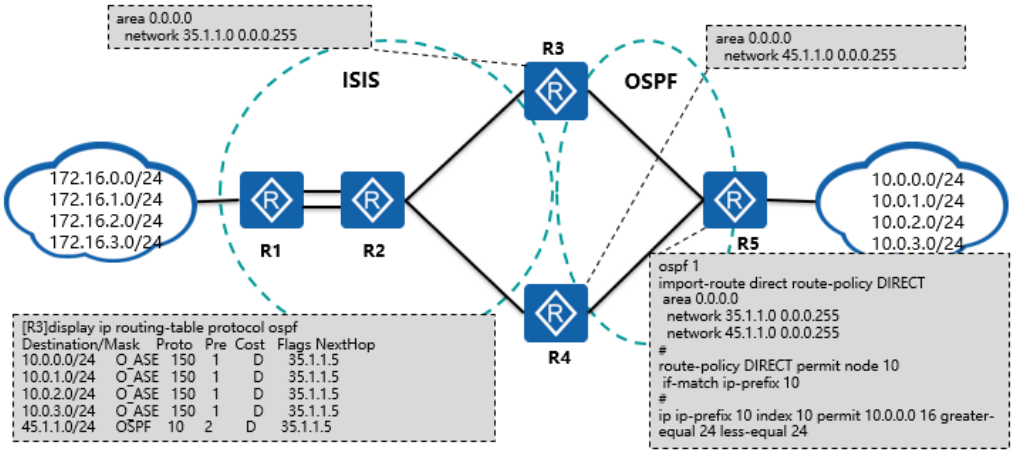
- 公司B网络部分拓扑如下图所示，现公司B要求如下：
  - R3、R4和R5属于OSPF区域0，仅将10.0.X.0/24引入到OSPF区域；
  - 在R5上对10.0.X.0/24进行汇总，汇总为10.0.0.0/16；在R3和R4向OSPF区域下发缺省路由；
  - 确保网络没有环路。



- 本案例中设备互联地址规则如下：
- 如 RTX 与 RTY 互联，则互联地址为 XY.1.1.X 与 XY.1.1.Y，掩码长度为 24 位。

案例1 - 需求1

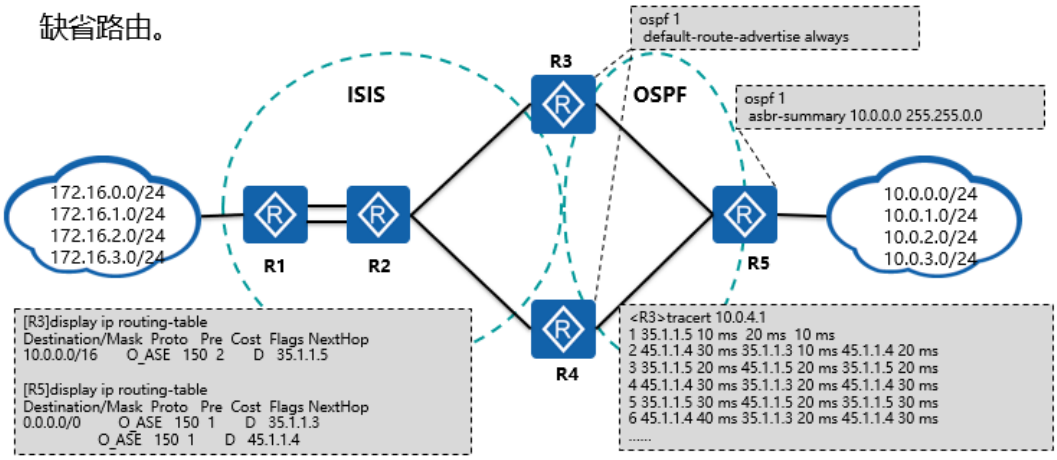
- R3、R4和R5属于OSPF区域0，仅将10.0.X.0/24引入到OSPF区域。



- 该部分只需要注意 R5 在做路由引入时，需要精细匹配。

案例1 - 需求2

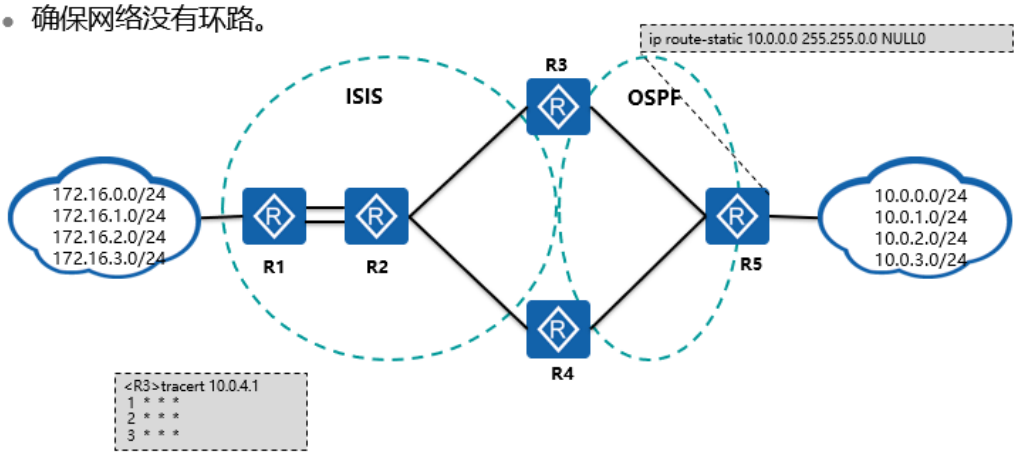
- 在R5上对10.0.X.0/24进行汇总，汇总为10.0.0.0/16；在R3和R4向OSPF区域下发缺省路由。



- 当 tracert 一个不存在的、但是在网络 10.0.0.0/16 内的地址时，会发生环路。该环路的产生主要是由于 OSPF 产生汇总路由时，不自动生成指向 null0 的路由所致。

案例1 - 需求3

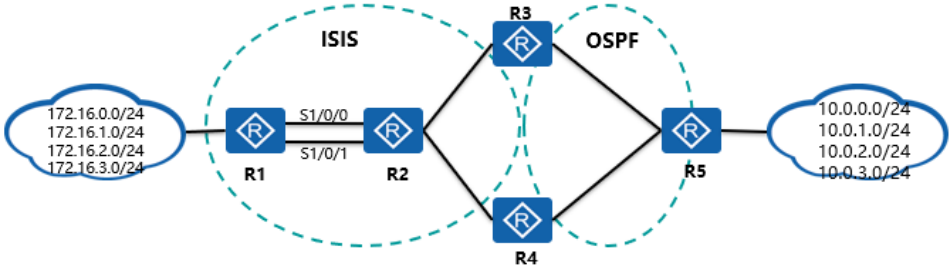
- 确保网络没有环路。



- 在 R5 上通过命令配置指向 null0 的静态路由，即可破除环路。

案例2

- 公司B网络在原有配置的基础上对网络进行了扩充，需求如下：
  - IS-IS基本配置已完成；
  - 结合filter-policy，将172.16.0.0/24和172.16.2.0/24引入到IS-IS中；
  - 在R3和R4上进行路由相互引入，充分避免环路，并消除次优路由；
  - 172.16.0.0/24网络访问OSPF网络经由S1/0/0， 172.16.2.0/24网络访问OSPF网络经由S1/0/1。

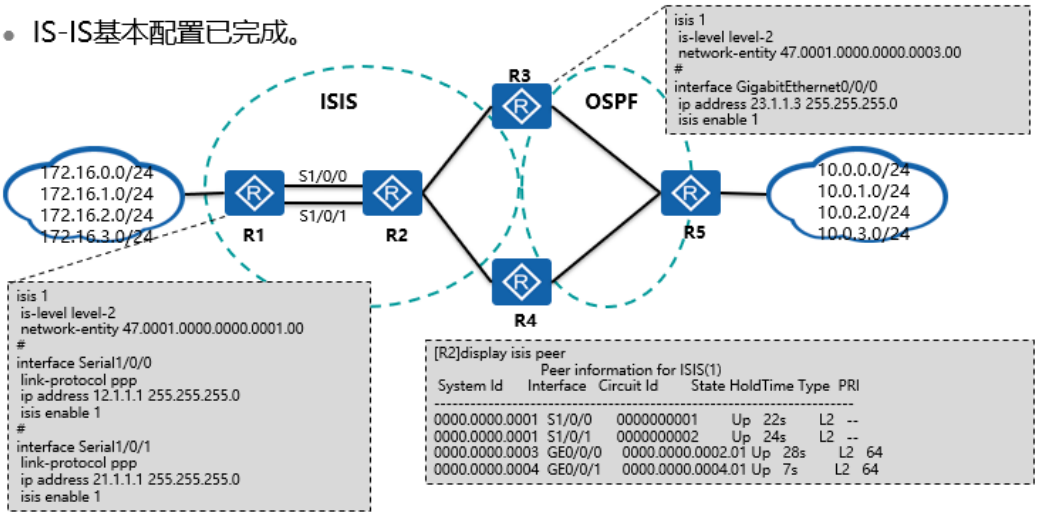


- 本案例中的需求是对之前案例的扩展，在原案例的基础上进行配置。
- 本案例中设备互联地址规则如下：
- 如 RTX 与 RTY 互联，则互联地址为 XY.1.1.X 与 XY.1.1.Y,掩码长度为 24 位。
- R1 接口 S1/0/0 地址为 12.1.1.1/24，R2 接口 S1/0/0 地址

为 12.1.1.2/24；R1 接口 S1/0/1 地址为 21.1.1.1/24，R2 接口 S1/0/1 地址为 21.1.1.2/24。

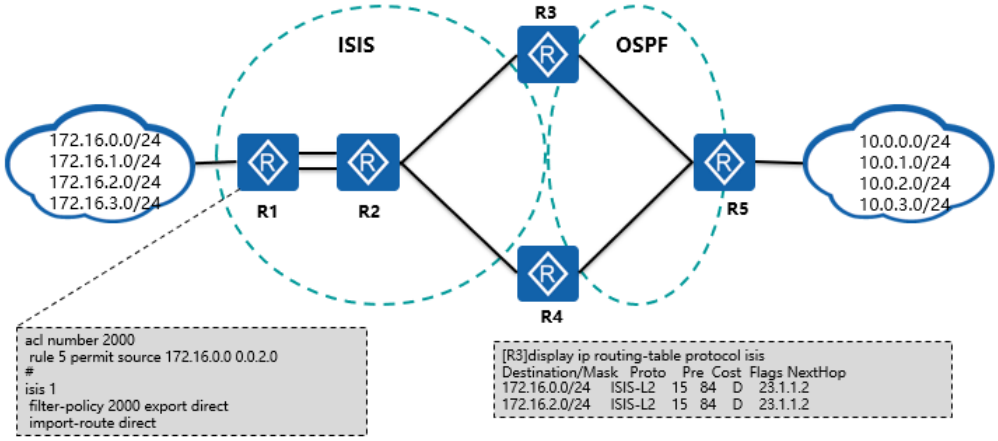
案例2 - 预配

- IS-IS基本配置已完成。



案例2 - 需求1

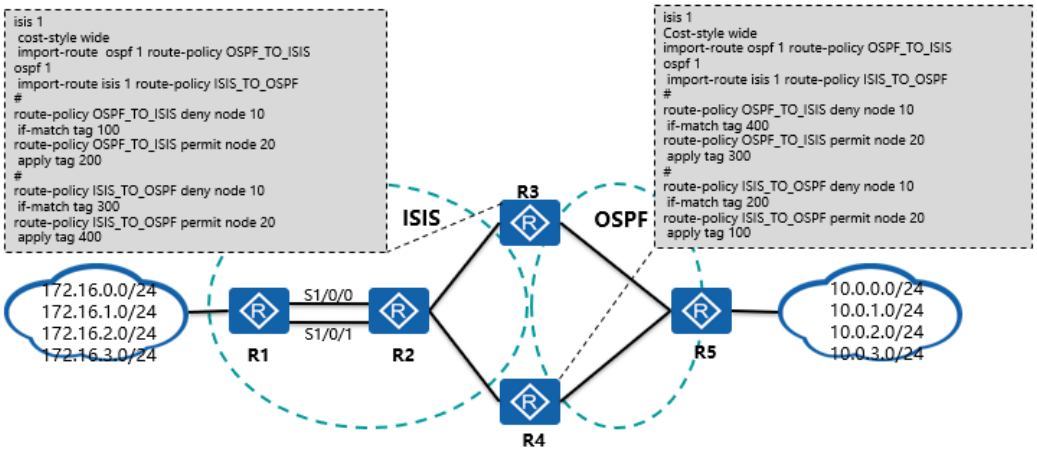
- 结合filter-policy，将172.16.0.0/24和172.16.2.0/24引入到IS-IS中。



- 通过 filter-policy 结合 ACL，将需求中指定的两个网段导入到 ISIS 协议中，注意，当使用 filter-policy 过滤路由协议的导入的时候，使用 export 方向。

案例2 - 需求2

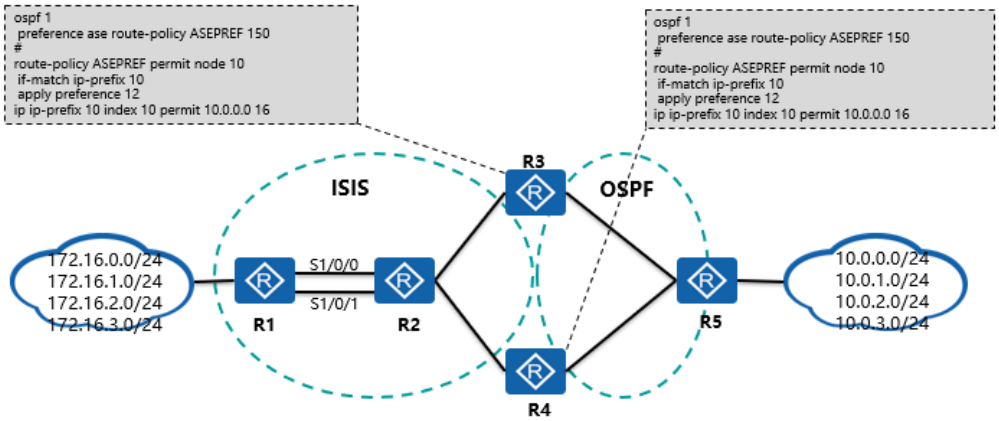
- 在R3和R4上进行路由相互引入，充分避免环路，并消除次优路由。



- 本案例中通过在导入路由的时候给路由条目加 Tag 的方法防止出现路由环路。ISIS 路由协议如果需要支持 Tag，必须要使用 wide 类型的开销，否则 ISIS 路由不能携带 Tag 标记。
- 使用 Tag 标签防止了路由回环，但是不能防止次优路由的产生。如果要避免次优路由，需要修改相应路由的 preference 值。

案例2 - 需求2（续）

- 在R3和R4上进行路由相互引入，充分避免环路，并消除次优路由。

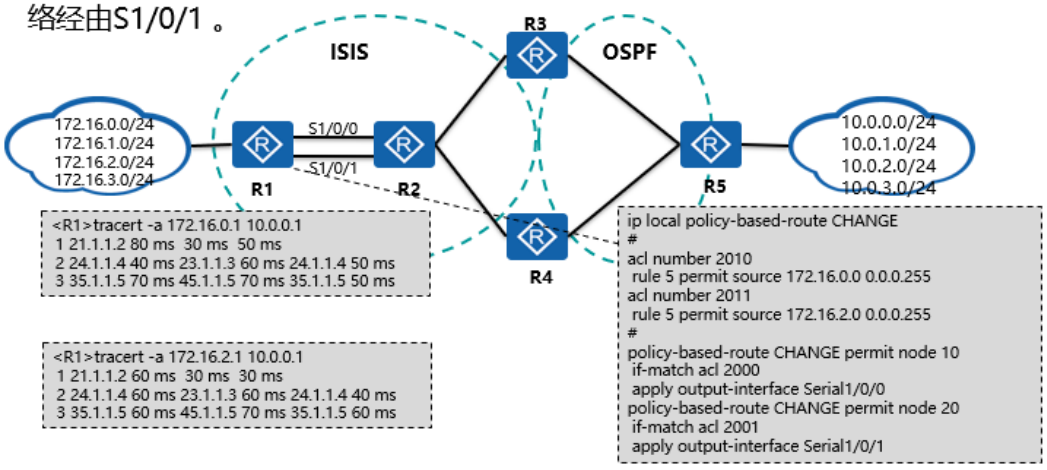


- 本例中的配置将避免路由 10.0.0.0/16 在 R3/R4 上出现次优路由。因为导入速度的不一致，总会导致 R3/R4 两台路由

器中的一台会从 ISIS 和 OSPF 同时学到 10.0.0.0/16，如果 R3 先行导入，那么 R4 就会同时从 ISIS 和 OSPF 学到 10.0.0.0/16 路由，而 R4 选择路由时会比较他们的 preference，因 OSPF 外部路由的 preference 值为 150，而 isis 的 preference 值为 15，所以 R4 会选择到达经由 IS-IS 域到达网络 10.0.0.0/16，该路径是次优路径。所以，通过在 R4 上，修改 OSPF 外部路由 10.0.0.0/16 的 preference 值，使其小于 IS-IS 的 preference 值，从而消除次优路径，考虑合理性，建议 OSPF 的外部路由 preference 值要大于 OSPF 的内部 preference 值（10）。

案例2 - 需求3

- 172.16.0.0/24网络访问OSPF网络经由S1/0/0， 172.16.2.0/24网络访问OSPF网络经由S1/0/1 。



思考题

1. 路由控制都包括哪些内容（　　）

A. 路由的发布

B. 路由的接收

C. 过滤和控制引入的路由

D. 设置特定路由的属性

2. 前缀列表：ip ip-prefix Prefix1 permit 160.0.0.0 8 的含义是（　　）

A. 前缀的前三个比特必须为"101",掩码长度必须在8和32之间

B. 前缀的前三个比特必须为"101",掩码长度必须为8

C. 前缀号必须为"160",掩码长度必须在8和32之间

D. 前缀号必须为“160”,掩码长度必须为8

- ABCD
- D