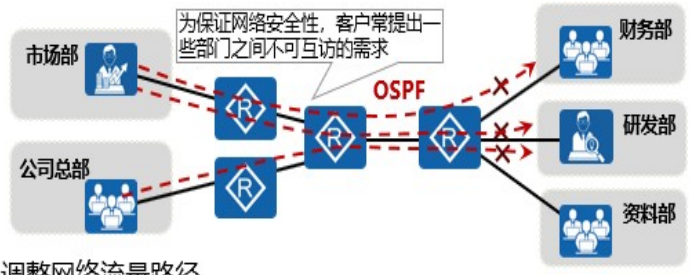
路由控制

前言

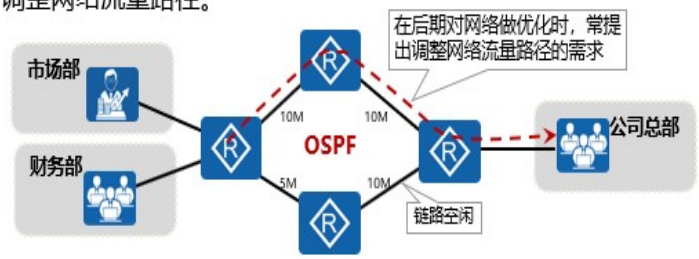
- 在企业网络的设备通信中，常面临一些非法流量访问的安全性及流量路径不优等问题，故为保证数据访问的安全性、提高链路带宽利用率，就需要对网络中的流量行为进行控制，如控制网络流量可达性、调整网络流量路径等。
- 而当面对更加复杂、精细的流量控制需求时，就需要灵活地使用一些工具来实现，本课程将主要介绍一些有关流量控制的常用工具及其使用场景。

对流量行为的控制需求分析

1. 控制网络流量可达性。



2. 调整网络流量路径。



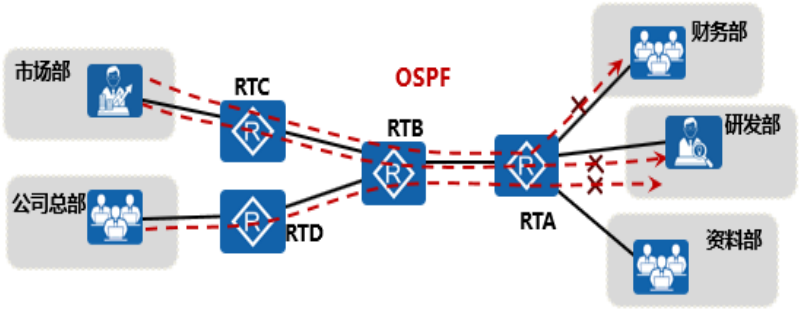
- 控制网络流量可达性：如图，为满足业务需求和保证数

据访问安全性，要求市场部不能访问财务部、研发部，公司总部不能访问研发部。

- 调整网络流量路径：如图，根据 OSPF 协议计算生成的路由，市场部和财务部访问公司总部都选择通过一条开销最小的路径，即使该路径发生拥塞也如此，而另外一条路径的链路带宽则一直处于空闲状态，这样就造成了带宽浪费的问题。

控制网络流量可达性

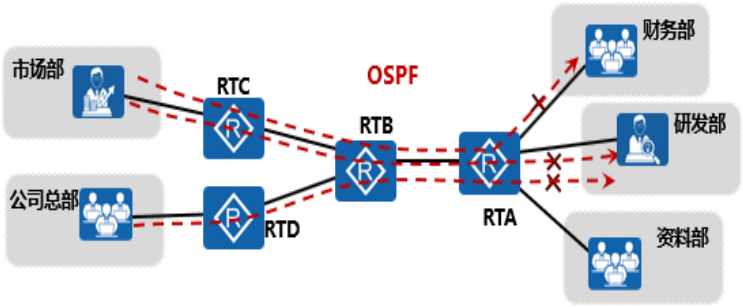
- 思考：如何控制网络流量可达性?



- 解决方案一：可通过修改路由条目（即对接收和发布的路由进行过滤）来控制流量可达性，这种方式称为**路由策略**。
- 解决方案二：可使用Traffic-Filter工具对数据进行过滤，这种方式称为**流量过滤**。

- 路由策略（Routing Policy）的作用是当路由器在发布、接收和引入路由信息时，可根据实际组网需要实施一些策略，以便对路由信息进行过滤或改变路由信息的属性，如：
- 控制路由的发布：只发布满足条件的路由信息。
- 控制路由的接收：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。
- 过滤和控制引入的路由：一种路由协议在引入其它路由协议时，只引入一部分满足条件的路由信息，并对所引入的路由信息的某些属性进行设置，以使其满足本协议的要求。
- 设置特定路由的属性：为通过路由策略过滤的路由设置相应的属性。

解决方案一：采用路由策略方式

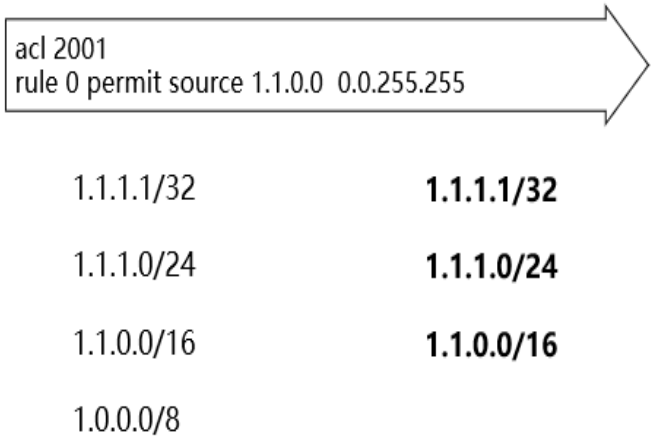


- 可利用**Filter-Policy**工具对RTA向OSPF引入的路由和RTC写入路由表的路由进行过滤：
  - 首先使用ACL或IP-Prefix List工具来匹配目标流量；
  - 然后在协议视图下，利用Filter-Policy向目标流量发布策略。
- 可利用**Route-Policy**工具，在RTA引入直连路由时对路由进行过滤：
  - 首先使用ACL或IP-Prefix List工具来匹配目标流量；
  - 然后在协议视图下，利用Route-Policy对引入的路由条目进行控制。

- 路由策略的实现分为两个步骤：
- 定义规则：首先要定义将要实施路由策略的路由信息的特征，即定义一组匹配规则，可以以路由信息中的不同属性作为匹配依据进行设置，如目的地址、AS 号等；
- 应用规则：根据设置的匹配规则，再将它们应用于路由的发布、接收和引入等过程中。
- 目前提供了如下几种过滤器供路由协议引用：
- 访问控制列表；
- 地址前缀列表；
- AS 路径过滤器；
- 团体属性过滤器；
- 扩展团体属性过滤器；
- 路由标识属性过滤器。

ACL应用示例 (1)

- ACL可通过匹配报文的信息实现对报文的分类。

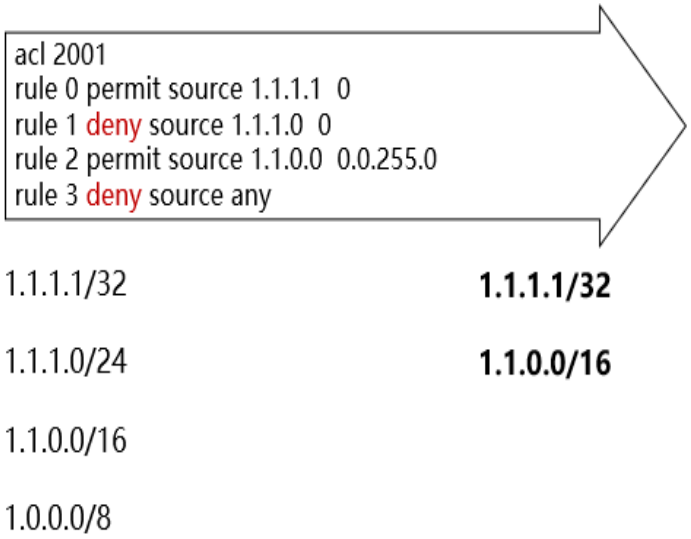


- 访问控制列表 ACL（Access Control List）是由 permit 或 deny 语句组成的一系列有顺序规则的集合，它通过匹配报文的信息实现对报文的分类。
- ACL 的分类：
- 基本 ACL：主要基于源地址、分片标记和时间段信息对数据包进行分类定义，编号范围为 2000-2999。
- 高级 ACL：可以基于源地址、目的地址、源端口号、目的端口号、协议类型、优先级、时间段等信息对数据包进行更为细致的分类定义，编号范围为 3000-3999。
- 二层 ACL：主要基于源 MAC 地址、目的 MAC 地址和报文类型等信息对数据包进行分类定义，编号范围为 4000-4999。
- 用户自定义 ACL：主要根据用户自定义的规则对数据报文做出相应的处理，编号范围为 5000-5999。
- 一个 ACL 可以由多条“deny | permit”语句组成，每一条语句描述了一条规则。设备收到数据流量后，会逐条匹配 ACL 规则，看其是否匹配。如果不匹配，则继续匹配下一条。一旦找到一条匹配的规则，就会执行规则中定义的动作，且不再

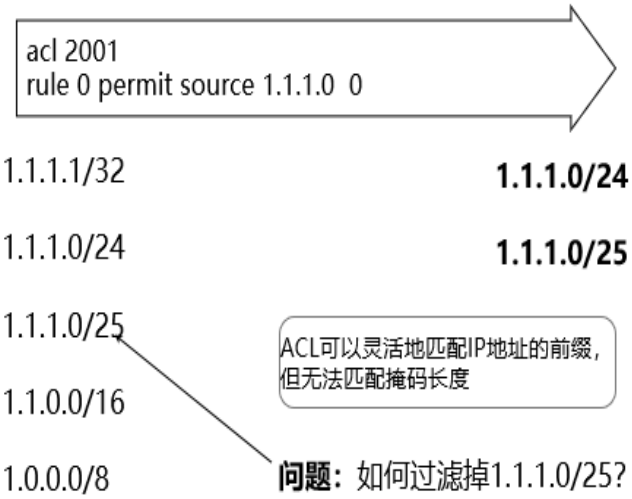
继续与后续规则进行匹配；如果找不到匹配的规则，则设备会对报文直接进行转发。

- 需要注意的是，ACL 中定义的这些规则可能存在重复或矛盾的地方。规则的匹配顺序决定了规则的优先级，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。

ACL应用示例 (2)



ACL应用示例 (3)



IP-Prefix List应用示例 (1)

IP-Prefix List能够同时匹配IP地址前缀及掩码长度。

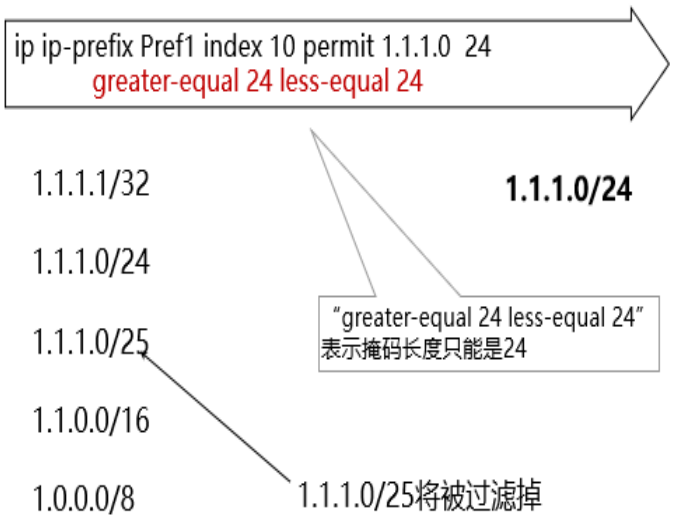
IP-Prefix List不能用于IP报文的过滤，只能用于路由信息的过滤。

ip ip-prefix test index 10 permit 10.0.0.0 16 greater-equal 24 less-equal 28
IP地址范围：10.0.0.0 – 10.0.x.x
24<= 掩码长度 <=28
例：10.0.1.0/24, 10.0.2.0/25, 10.0.2.192/26

- 地址前缀列表即 IP-Prefix List。可以通过地址前缀列表，将与所定义的前缀过滤列表相匹配的路由，根据定义的匹配模式进行过滤，以满足使用者的需要。

- 前缀列表的组成及匹配规则：
- 前缀过滤列表由 IP 地址和掩码组成，IP 地址可以是网段地址或者主机地址，掩码长度的配置范围为 0～32。
- IP-Prefix List 中的每一条 IP-Prefix 都有一个序列号 index，匹配的时候将根据序列号从小到大进行匹配。
- 如果不配置 IP-Prefix 的 index，那么对应的 index 在上次配置的同名 IP-Prefix 的 index 的基础上，以步长为 10 进行增长。如果配置的 IP-Prefix 的名字与 index 都和已经配置了的一项 IP-Prefix List 的相同，仅仅只是匹配的内容不同，则该 IP-Prefix List 将覆盖原有的 IP-Prefix List。
- 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，则默认匹配模式为 permit。
- 前缀掩码长度范围：
- 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，并通过配置关键字 greater-equal 和 less-equal 来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，前缀过滤列表会进行精确匹配，即只匹配掩码长度为与前缀过滤列表掩码长度相同的 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 位的长度；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 的指定值。

IP-Prefix List应用示例 (2)



Filter-Policy工具介绍

- Filter-Policy能够对接收或发布的路由进行过滤，可应用于ISIS、OSPF、BGP等协议。

对协议接收的路由进行过滤：
filter-policy { acl-number | ip-prefix ip-prefix-name } import

对协议发布的路由进行过滤：
filter-policy { acl-number | ip-prefix ip-prefix-name } export

- 应用各协议中的 Filter-Policy 工具可通过引用 ACL 或地址前缀列表，对接收、发布和引入的路由进行过滤。
- 对于距离矢量协议和链路状态协议，Filter-Policy 工具的

操作过程是不同的：

- 距离矢量协议是基于路由表生成路由的，因此过滤器会影响从邻居接收的路由和向邻居发布的路由。
- 链路状态路由协议是基于链路状态数据库来生成路由的，且路由信息隐藏在链路状态 LSA 中，但 Filter-Policy 不能对发布和接收的 LSA 进行过滤，故 Filter-Policy 不影响链路状态通告或链路状态数据库的完整性以及协议路由表，而只会影响本地路由表，且只有通过过滤的路由才被添加到路由表中，没有通过过滤的路由不会被添加进路由表。
- 不同协议应用 filter-policy export 命令对待发布路由的影响范围不同：
- 对于距离矢量协议，会对引入的路由信息、本协议发现的路由信息进行过滤。
- 对于链路状态协议，只对引入的路由信息进行过滤。

Route-Policy工具介绍

- Route-Policy是一种功能非常强大的路由策略工具，它可以灵活地与ACL、IP-Prefix List、As-Path-Filter等其它工具配合使用

```
Route-Policy:
  route-policy route-policy-name { permit | deny } node node
    if-match {acl/cost/interface/ip next-hop/ip-prefix}
    apply {cost/ip-address next-hop/tag}
```

- Route-Policy由若干个node构成，node之间是“或”的关系。且每个node下可以有若干个if-mach和apply子句，if-match之间是“与”的关系

- Route-Policy 的每个 node 都有相应的 permit 模式或 deny 模式。如果是 permit 模式，则当路由项满足该 node 的所有 if-match 子句时，就被允许通过该 node 的过滤并执行该 node

的 apply 子句，且不再进入下一个 node；如果路由项没有满足该 node 的所有 if-match 子句，则会进入下一个 node 继续进行过滤。如果是 deny 模式，则当路由项满足该 node 的所有 if-match 子句时，就被拒绝通过该 node 的过滤，这时 apply 子句不会被执行，并且不进入下一个 node；否则就进入下一个 node 继续进行过滤。

Route-Policy应用示例

Table-1

Network	Cost	NextHop
1.1.2.0/24	4687	34.34.34.2
	4687	13.13.13.1
1.1.3.0/24	4687	34.34.34.2
	4687	13.13.13.1
1.1.3.0/25	1	34.34.34.2
	1	13.13.13.1
5.5.5.5/32	4687	34.34.34.2
	4687	13.13.13.1
6.6.6.6/32	4687	34.34.34.2
	4687	13.13.13.1

Table-2

Network	Cost	NextHop
1.1.3.0/24	4687	34.34.34.2
	21	13.13.13.1
1.1.3.0/25	11	34.34.34.2
	21	13.13.13.1

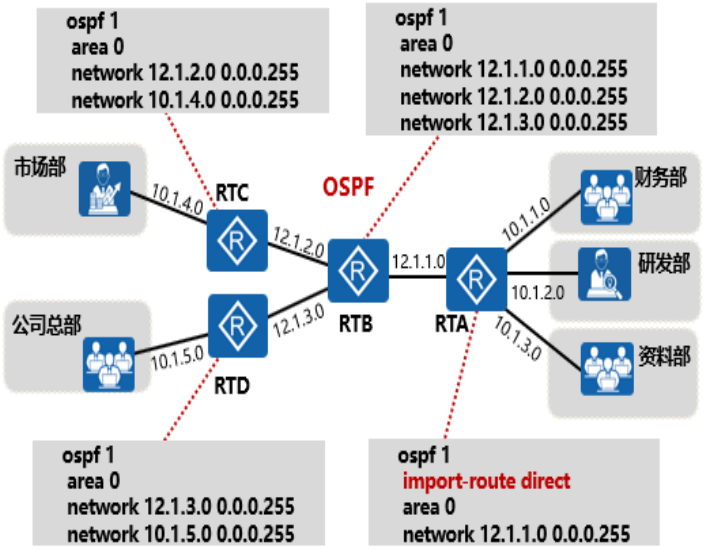
```
acl 2001
 rule 0 permit source 1.1.3.0 0.0.0.255
acl 2002
 rule 0 permit source 13.13.13.1 0

route-policy RP deny node 10
 if-match ip-prefix Pref1
route-policy RP permit node 20
 if-match ip-prefix Pref2
route-policy RP permit node 30
 if-match acl 2001
 if-match ip next-hop acl 2002
 apply cost 21
route-policy RP permit node 40
 if-match ip-prefix Pref3
 apply cost 11
route-policy RP permit node 50
#
 ip ip-prefix Pref1 index 10 permit 5.5.5.5 32
 ip ip-prefix Pref1 index 20 permit 1.1.2.0 24
 ip ip-prefix Pref2 index 10 deny 6.6.6.6 32
 ip ip-prefix Pref3 index 10 permit 1.1.3.0 24
  greater-equal 25 less-equal 25
```

- Pref1 用来匹配 5.5.5.5/32 或 1.1.2.0/24，它们将被 route-policy RP 的 node 10 过滤掉（deny），所以 Table-2 中见不到 5.5.5.5/32 和 1.1.2.0/24。
- Pref2 用来过滤 6.6.6.6/32（deny），所以尽管 route-policy RP 的 node 20 是 permit，6.6.6.6/32 仍然会被过滤掉。因此，Table-2 中见不到 6.6.6.6/32。
- route-policy RP 的 node 30 定义了两个 if-match 语句，分别针对 ACL 2001 和 ACL 2002。匹配 ACL 2001 的路由有 1.1.3.0/24（下一跳为 34.34.34.2）、1.1.3.0/24（下一跳为 13.13.13.1）、1.1.3.0/25（下一跳为 34.34.34.2）、1.1.3.0/25（下一跳为 13.13.13.1），同时又匹配 ACL 2002 的路由有 1.

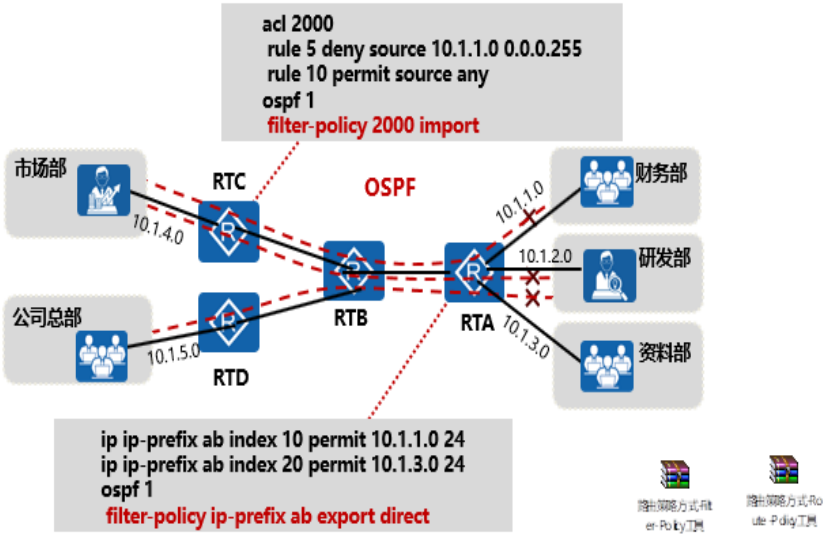
1.3.0/24（下一跳为 13.13.13.1）和 1.1.3.0/25（下一跳为 13.13.13.1）。于是，1.1.3.0/24（下一跳为 13.13.13.1）和 1.1.3.0/25（下一跳为 13.13.13.1）的 cost 被修改为 21。

- 1.1.3.0/24（下一跳为 34.34.34.2）和 1.1.3.0/25（下一跳为 34.34.34.2）继续尝试通过 route-policy RP 的 node 40。由于 1.1.3.0/25 满足 Pref3，所以 1.1.3.0/25（下一跳为 34.34.34.2）的 cost 被修改为 11。
- 最后，1.1.3.0/24（下一跳为 34.34.34.2）通过了 route-policy RP 的 node 50。

路由策略方式配置实现 (1)



路由策略方式配置实现 (2)



- RTA 还可以使用 Route-Policy 工具进行路由控制：
- acl 2000
- rule 0 permit source 10.1.1.0 0.0.0.255
- rule 5 permit source 10.1.3.0 0.0.0.255
- route-policy huawei-control permit node 10
- if-match acl 2000
- ospf 1
- import-route direct route-policy huawei-control

路由策略方式配置实现 (3)

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

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

  10.1.3.0/24       O_ASE   150  1           D   12.1.2.1        GigabitEthernet 0/0/0
  10.1.4.0/24       Direct  0    0           D   10.1.4.2        GigabitEthernet 0/0/1
  10.1.5.0/24       OSPF    10   3           D   12.1.2.1        GigabitEthernet 0/0/0
```

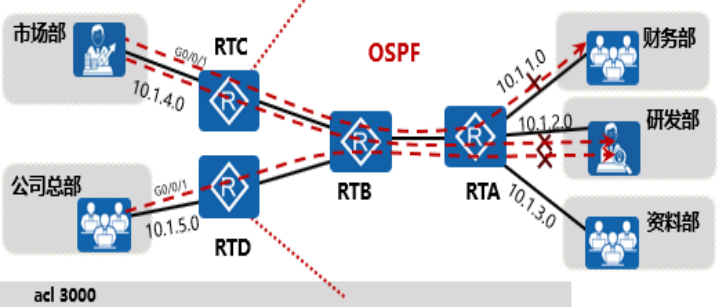
```
<RTD>dis ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 15        Routes : 15

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

  10.1.1.0/24       O_ASE   150  1           D   12.1.3.1        GigabitEthernet 0/0/0
  10.1.3.0/24       O_ASE   150  1           D   12.1.3.1        GigabitEthernet 0/0/0
  10.1.4.0/24       OSPF    10   3           D   12.1.3.1        GigabitEthernet 0/0/0
  10.1.5.0/24       Direct  0    0           D   10.1.5.2        GigabitEthernet 0/0/1
```

- 注明：上面显示的是部分关键信息，并非全部。

解决方案二：采用流量过滤方式 (1)

- 基于自定义策略实现：使用Traffic-Filter工具对数据进行过滤。

```
acl number 3000
 rule 0 deny ip source 10.1.4.0 0.0.0.255 dest 10.1.1.0 0.0.0.255
 rule 5 deny ip source 10.1.4.0 0.0.0.255 dest 10.1.2.0 0.0.0.255
 rule 10 permit ip source  any
int g0/0/1
 traffic-filter inbound acl 3000
```




```
acl 3000
 rule 0 deny ip source 10.1.5.0 0.0.0.255 dest 10.1.2.0 0.0.0.255
int g0/0/1
 traffic-filter inbound acl 3000
```

解决方案二：采用流量过滤方式 (2)

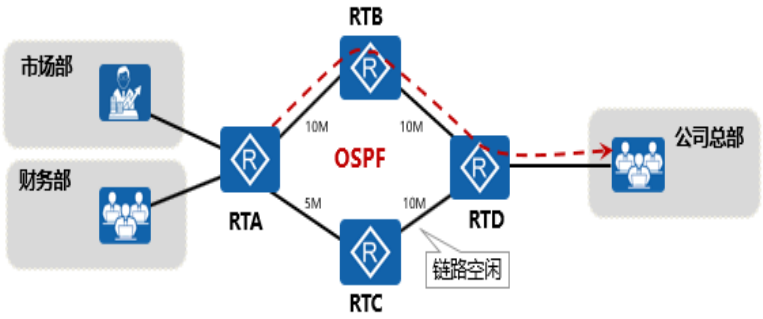
```
[RTC]dis ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 16        Routes : 16

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.1.1.0/24  O_ASE   150  1           D   12.1.2.1        GigabitEthernet 0/0/0
        10.1.2.0/24  O_ASE   150  1           D   12.1.2.1        GigabitEthernet 0/0/0
        10.1.3.0/24  O_ASE   150  1           D   12.1.2.1        GigabitEthernet 0/0/0
        10.1.4.0/24  Direct  0    0           D   10.1.4.2        GigabitEthernet 0/0/1
        10.1.5.0/24  OSPF    10   3           D   12.1.2.1        GigabitEthernet 0/0/0
```

```
PC-市场部>ping 10.1.1.1

Ping 10.1.1.1: 32 data bytes, Press Ctrl_C to break
Request timeout!
Request timeout!
Request timeout!
Request timeout!

--- 10.1.1.1 ping statistics ---
  4 packet(s) transmitted
  0 packet(s) received
  100.00% packet loss
```

- 经测试，在设置完流量过滤后，RTC 的路由表仍然有全网的路由，且市场部无法访问财务部和研发部，对于其他部门仍可正常访问。同样，RTD 的路由表也有全网的路由，且公司总部无法访问研发部，其他仍可正常访问。

调整网络流量路径 - 单协议简单场景

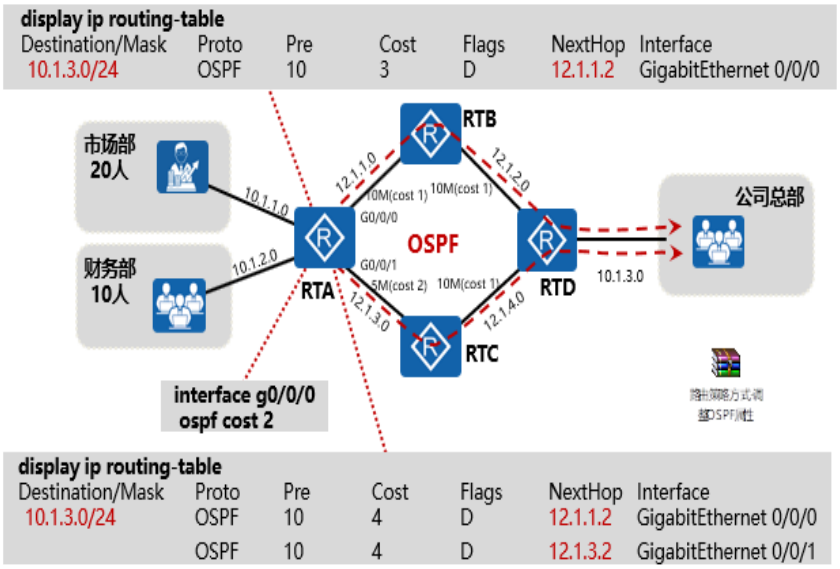
- 在后期对网络进行优化时，常出现调整网络流量路径的需求。



- 解决方案一：可通过**路由策略方式**修改协议属性来控制路由表条目，从而调整流量路径。
- 解决方案二：可采用**策略路由方式**在查找路由表之前控制流量行为。

- 可通过修改协议本身的一些属性来控制路由条目，从而影响流量转发路径：
- 若运行 OSPF 或 ISIS 协议，可通过调整接口 Cost 属性值来实现；
- 若运行 RIP 协议，可通过调整 Metric 或下一跳属性来实现；
- 若运行 BGP 协议，则可通过调整 AS-Path、Local_Pref、MED、Community 等属性来实现。

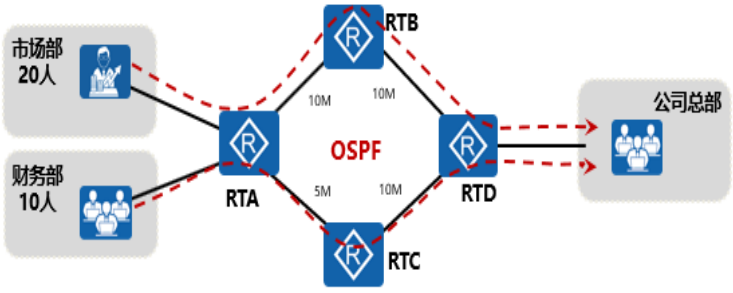
解决方案一：采用路由策略方式



- 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，思考该方式有何缺点？是否可以满足更复杂、更精确的控制需求？

解决方案一的局限性

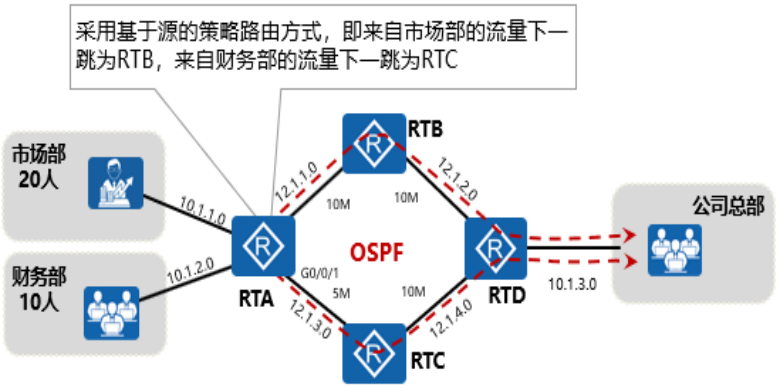
- 为充分利用链路带宽，现要求市场部访问总部流量路径为RTA-RTB-RTD，财务部访问总部流量路径为RTA-RTC-RTD。



- 如图，若采用解决方案一来实现以上需求，由于其只能依据数据包的目的地址做转发策略，所以无法满足需求；故当出现基于源地址、目的地址或基于应用层等一些复杂的控制需求时，就体现出其局限性。

- 正是由于路由策略实现方式的缺陷，促使了目前越来越多的用户希望能够在传统路由转发的基础上，根据自己定义的策略进行报文转发和选路。策略路由使网络管理者不仅能够根据报文的目的地址来制定策略，而且还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。

解决方案二：采用策略路由方式 (1)



- 策略路由方式常采用Traffic-Policy工具来实现：
  - 首先使用ACL工具匹配目标流量；
  - 然后对目标流量定义行为，如修改下一跳。

- 策略路由 PBR（Policy Based Routing）与单纯依照 IP 报文的目的地址查找路由表进行转发有所不同，它是一种依据用户制定的策略而进行流量转发的机制。
- 策略路由的查找优先级比路由策略高，当路由器接收到数据包并进行转发时，会优先根据策略路由的规则进行匹配，如果能匹配上，则根据策略路由进行转发，否则按照路由表中的路由条目来进行转发。其中策略路由不改变路由表中的任何内容，它可以通过预先设置的规则来影响数据报文的转发。
- 策略路由 PBR 分为：
- 本地策略路由：对本设备发送的报文实现策略路由，比如本机下发的 ICMP、BGP 等协议报文。

- 当用户需要实现不同源地址的报文或者不同长度的报文通过不同的方式进行发送时，可以配置本地策略路由。
- 常用 Policy-Based-Route 工具来实现。
- 接口策略路由：对本设备转发的报文生效，对本机下发的报文不生效。
- 当用户需要将收到的某些报文通过特定的下一跳地址进行转发时，需要配置接口策略路由。使匹配重定向规则的报文通过特定的下一跳出口进行转发，不匹配重定向规则的报文则根据路由表直接转发。接口策略路由多应用于负载分担和安全监控。
- 常用 Traffic-Policy 工具来实现。
- 智能策略路由：基于链路质量信息为业务数据流选择最佳链路。
- 当用户需要为不同业务选择不同质量的链路时，可以配置智能策略路由。
- 常用 Smart-Policy-Route 工具来实现，在本课程中不做重点介绍。

解决方案二：采用策略路由方式 (2)

```
[RTA]acl 3000
  rule 5 permit ip source 10.1.1.0 0.0.0.255 dest 10.1.3.0 0.0.0.255
traffic classifier huawei-control1
 if-match acl 3000
traffic behavior huawei-control1
 redirect ip-nexthop 12.1.1.2
traffic policy huawei-control1
 classifier huawei-control1 behavior huawei-control1
int g0/0/2
 traffic-policy huawei-control1 inbound
```

```
[RTA]acl 3001
 rule 5 permit ip source 10.1.2.0 0.0.0.255 dest 10.1.3.0 0.0.0.255
traffic classifier huawei-control2
 if-match acl 3001
traffic behavior huawei-control2
 redirect ip-nexthop 12.1.3.2
traffic policy huawei-control2
 classifier huawei-control2 behavior huawei-control2
int g4/0/0
 traffic-policy huawei-control2 inbound
```

- 本示例采用的是 MQC 的配置方式。

解决方案二：采用策略路由方式 (3)

```
<RTA>dis ip routing-table
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 19        Routes : 20

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.1.1.0/24 Direct  0    0           D   10.1.1.2        GigabitEthernet 0/0/2
        10.1.2.0/24 Direct  0    0           D   10.1.2.2        GigabitEthernet 4/0/0
        10.1.3.0/24 OSPF    10   3           D   12.1.1.2        GigabitEthernet 0/0/0
```

```
PC-市场部>tracert 10.1.3.1

traceroute to 10.1.3.1, 8 hops max
(ICMP), press Ctrl+C to stop
 1  10.1.1.2   47 ms  31 ms  15 ms
 2  12.1.1.2   47 ms  31 ms  32 ms
 3  12.1.2.2   93 ms  63 ms  46 ms
 4   *10.1.3.1   62 ms  31 ms
```

```
PC-财务部>tracert 10.1.3.1

traceroute to 10.1.3.1, 8 hops max
(ICMP), press Ctrl+C to stop
 1  10.1.2.2   16 ms  31 ms  16 ms
 2  12.1.3.2   62 ms  47 ms  31 ms
 3  12.1.4.2   47 ms  47 ms  31 ms
 4  10.1.3.1   32 ms  46 ms  32 ms
```

路由策略与策略路由的区别

路由策略	策略路由
基于控制平面，会影响路由表表项。	基于转发平面，不会影响路由表表项，且设备收到报文后，会先查找策略路由进行匹配转发，若匹配失败，则再查找路由表进行转发。
只能基于目的地址进行策略制定。	可基于源地址、目的地址、协议类型、报文大小等进行策略制定。
与路由协议结合使用。	需手工逐跳配置，以保证报文按策略进行转发。
常用工具：Route-Policy、Filter-Policy等。	常用工具：Traffic-Filter、Traffic-Policy、Policy-Based-Route等。

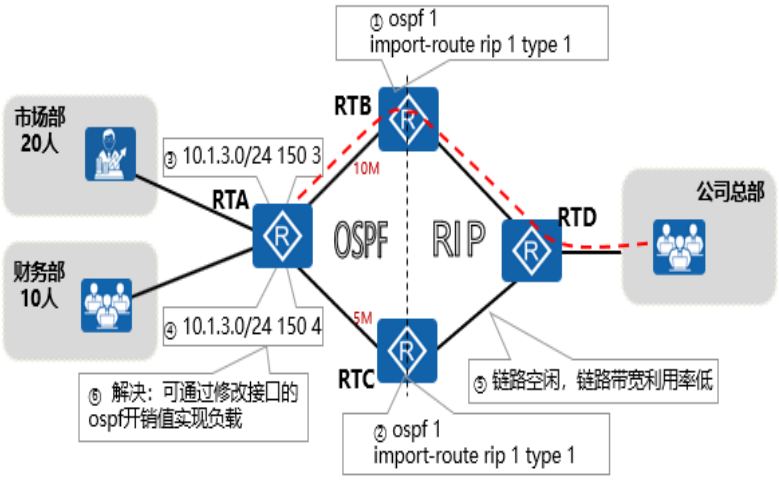
- 路由器存在两种类型的表：一个是路由表（routing-table），另一个是转发表（forwarding-table），转发表是由路由

表映射过来的，策略路由直接作用于转发表，路由策略直接作用于路由表。由于转发在底层，路由在高层，所以直接作用在转发表的转发优先级比查找路由表转发的优先级高。

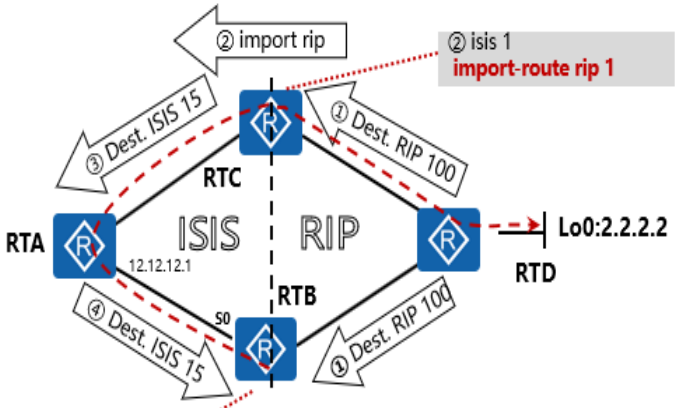
- 路由策略是在路由发现的时候产生作用，并根据一些规则，使用某种策略来影响路由发布、接收或路由选择的参数，从而改变路由发现的结果，从而最终改变路由表内容；策略路由是在数据包转发的时候发生作用，不改变路由表中的任何内容，它可以通过设置的规则影响数据报文的转发。

调整网络流量路径 - 多协议复杂场景

- 前文示例中描述的四台路由器都运行同一种协议，分析若运行不同协议会出现什么问题?



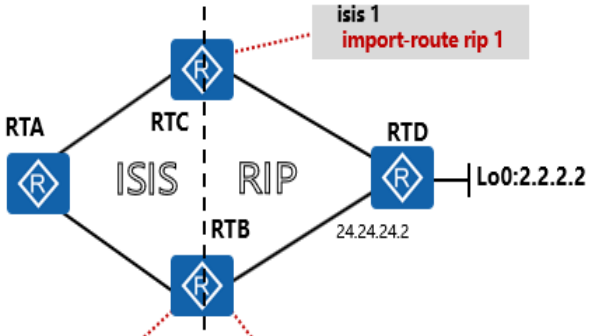
多协议复杂场景带来的其他问题 - 次优路由



⑤ [RTB]display ip routing-table

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
2.0.0.0/8	ISIS	15	84	12.12.12.1	Serial0

解决方案一：利用路由过滤避免次优路由

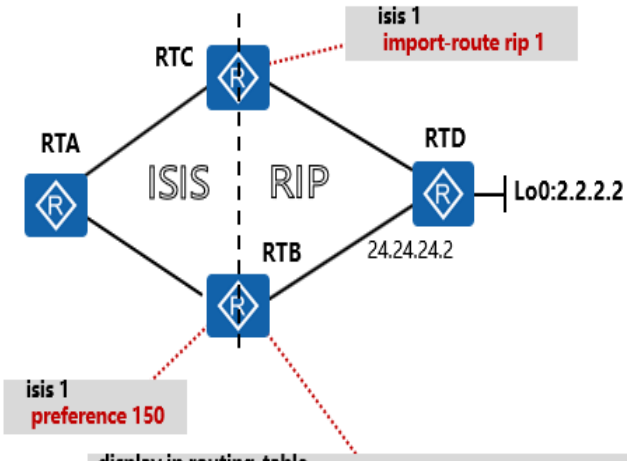



```
acl 2001
 rule 5 deny source 2.0.0.0 0.255.255.255
 rule 10 permit
isis 1
 filter-policy 2001 import
```

display ip routing-table

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
2.0.0.0/8	RIP	100	1	24.24.24.2	Serial1

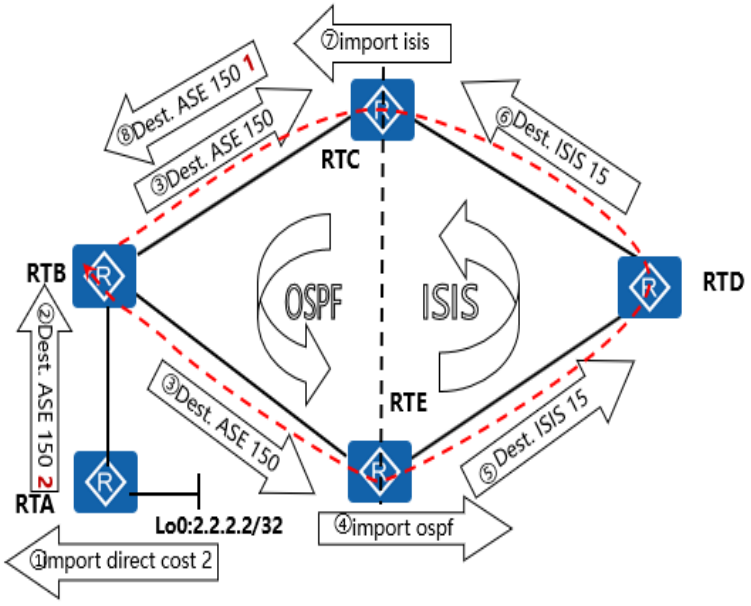
解决方案二：调整协议优先级避免次优路由



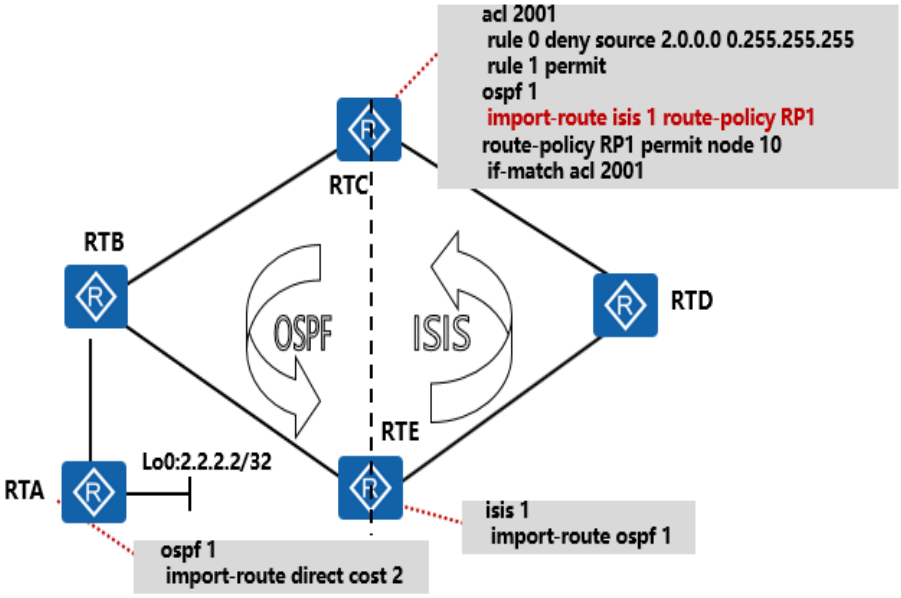
display ip routing-table

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
2.0.0.0/8	RIP	100	1	24.24.24.2	Serial1

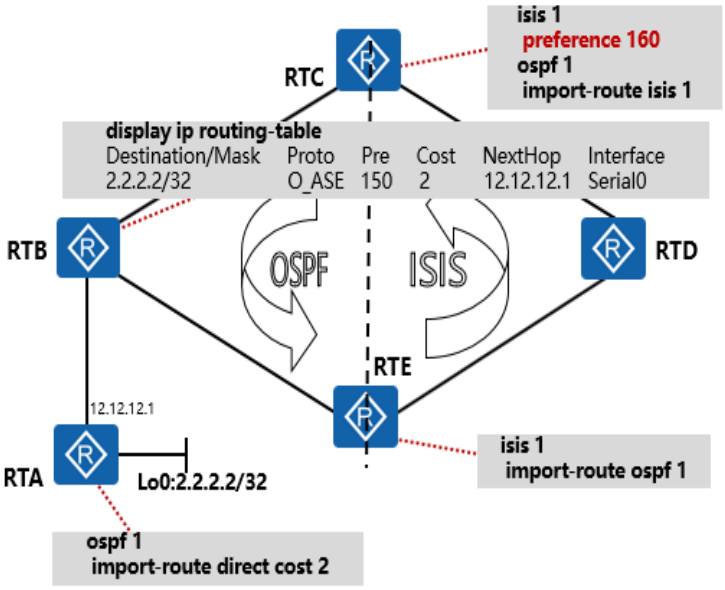
多协议复杂场景带来的其他问题 - 路由环路



解决方案一：利用路由过滤避免路由环路



解决方案二：调整协议优先级避免路由环路



思考题

1. IP-Prefix List可以用来过滤IP报文吗?
2. 常用调整网络流量路径的方式都包括哪些?
3. 路由引入可能会带来哪些问题? 常用的解决办法包括哪些?

- 答案：IP-Prefix List 可以用来过滤路由信息，但不能过滤 IP 报文。
- 答案：常用调整网络流量路径的方式包括：路由策略和策略路由方式。
- 答案：路由引入可能会带来次优路径、路由环路等问题，常采用路由过滤、调整协议优先级方式来解决。
-