红茶三杯
Vinsoney

学习 沉淀 成长 分享

IP路由选择原理

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2012-08-01

课程目标

IP路由选择原理

有类及无类路由协议

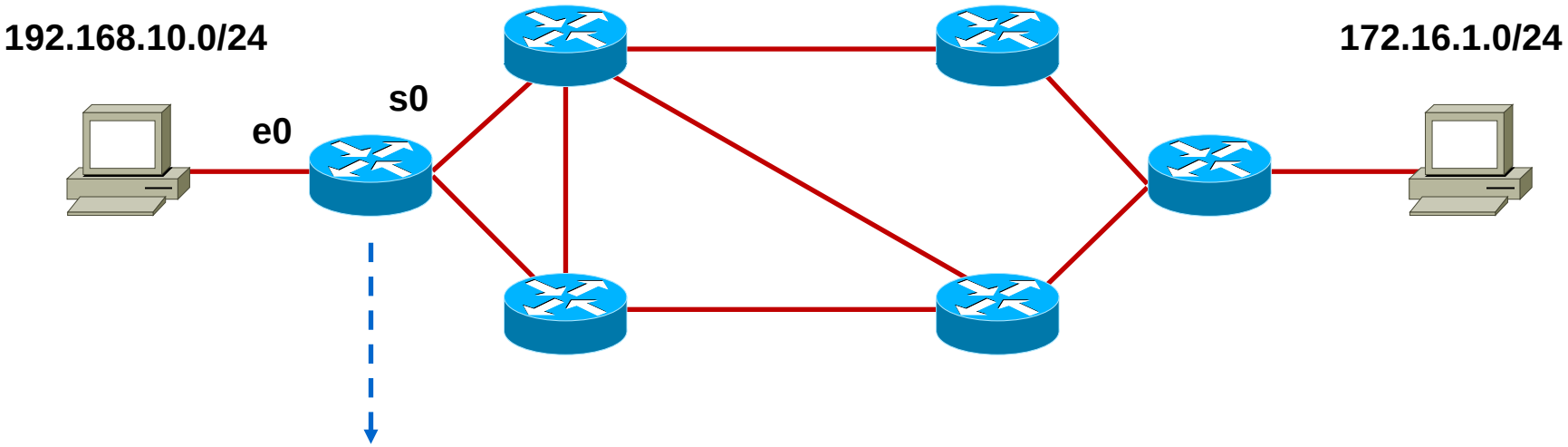
静态路由

红茶三杯
Vinsoney

IP路由选择原理

- IP路由的概念
- 路由信息来源
- 管理距离（AD值）
- 有类及无类路由查找
- 最长匹配原则
- 递归查

IP路由选择原理



Protocol	Destination Network	Exit Interface
Connected	192.168.10.0	e0
RIP	172.16.1.0	s0

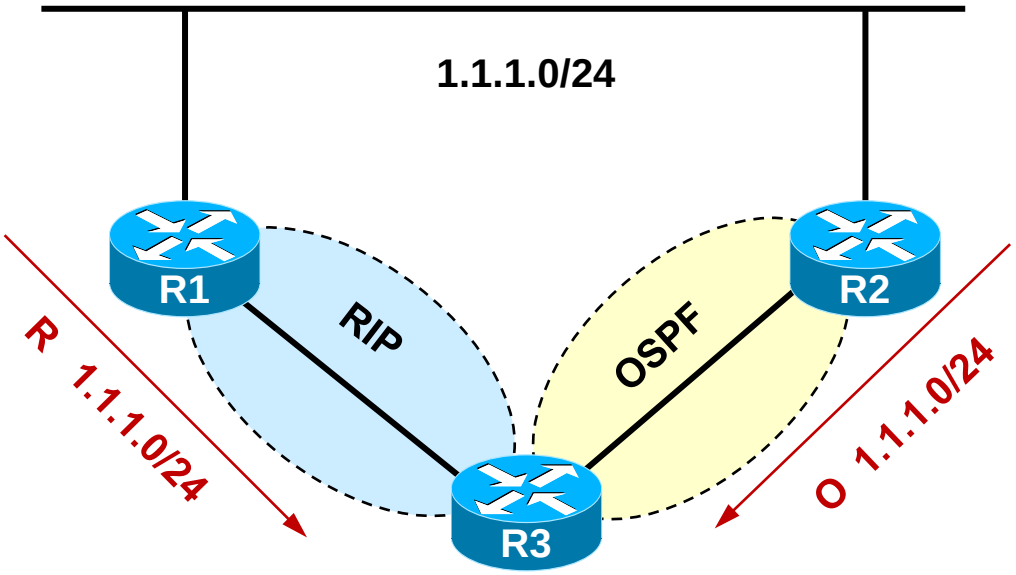
IP路由选择原理

- 路由信息来源
- 管理距离（AD值）
- 有类及无类路由查找
- 最长匹配原则
- 递归查询

路由信息的来源

- **直连路由**
  - 接口配置IP，该接口的物理层和数据链路层UP
  - 通过接口感知到的直连网络
- **静态路由**
  - 使用静态路由命令手工配置的路由
- **动态路由**
  - 通过动态路由协议学习
  - 常见路由协议：RIP、OSPF、IS-IS、Eigrp、BGP

管理距离（AD值）



Protocol	Destination Network	Exit Interface
O	192.168.1.0/24	e0

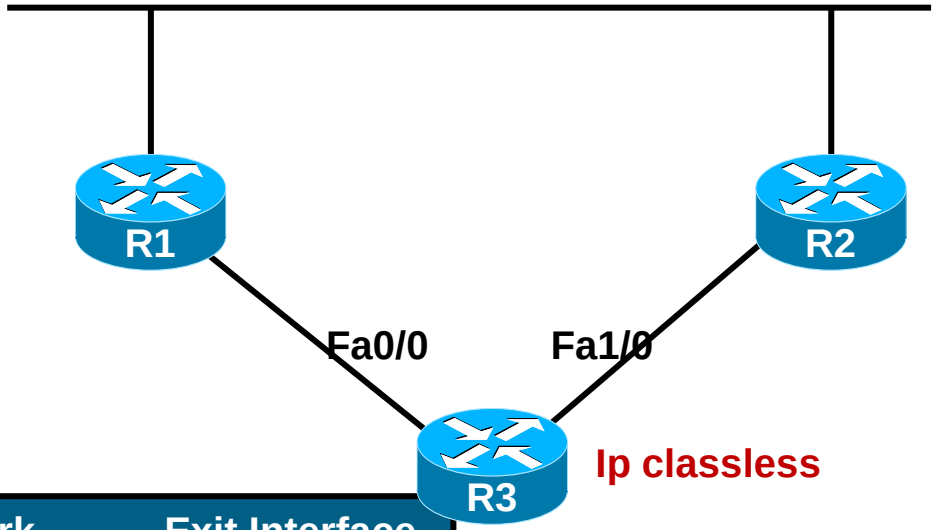
管理距离（AD值）

Routing Protocols	AD	备注
直连接口	0	
关联出接口的静态路由	1	Metric =0
关联下一跳的静态路由	1	Metric =0
EIGRP 汇总路由	5	
外部 BGP	20	
内部EIGRP	90	
IGRP	100	
OSPF	110	
RIPv1、v2	120	
外部EIGRP	170	
内部BGP	200	

有类及无类路由查找

- 无类路由查找：路由器不会注意目的地址的类别，它会在目的地址和所有已知的路由之间逐位(bit by bit)执行最长匹配
- 有类路由查找（no ip classless）：路由器收到一个数据包时，先查找目的地址所属主类，如果路由表中有主类路由，则再去找子网，如果有子网路由，则查询被限定在子网中，并进一步查找，如果最终查找失败，则丢弃数据包，即使有默认路由存在；如果本地没有该主类路由，则看是否有默认路由，如果有，则转发，如果无，则丢弃。

有类及无类路由查找

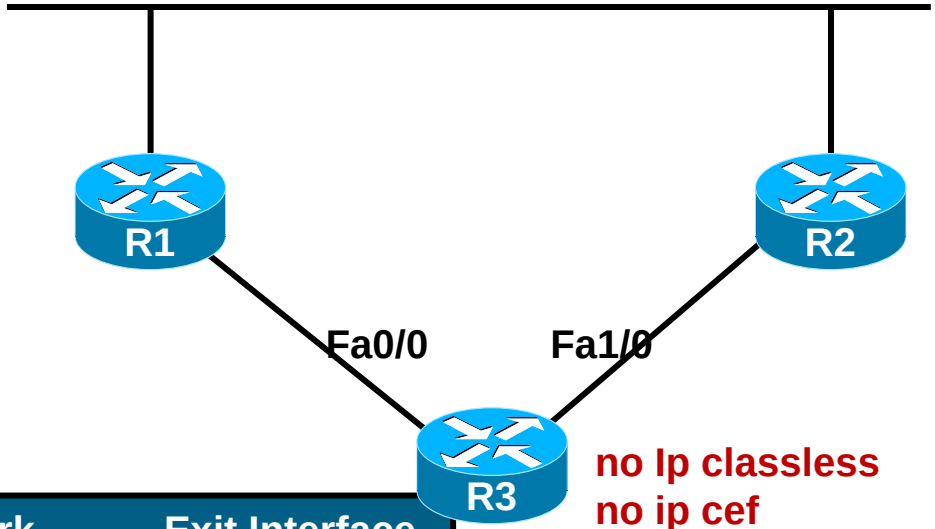


Destination Network	Exit Interface
192.168.1.0/24	Fa0/0
192.168.1.0/27	Fa1/0

DATA：192.168.1.1

DATA：192.168.1.33

有类及无类路由查找



Destination Network	Exit Interface
192.168.1.0/24	Fa0/0
192.168.1.0/27	Fa1/0

no Ip classless
no ip cef

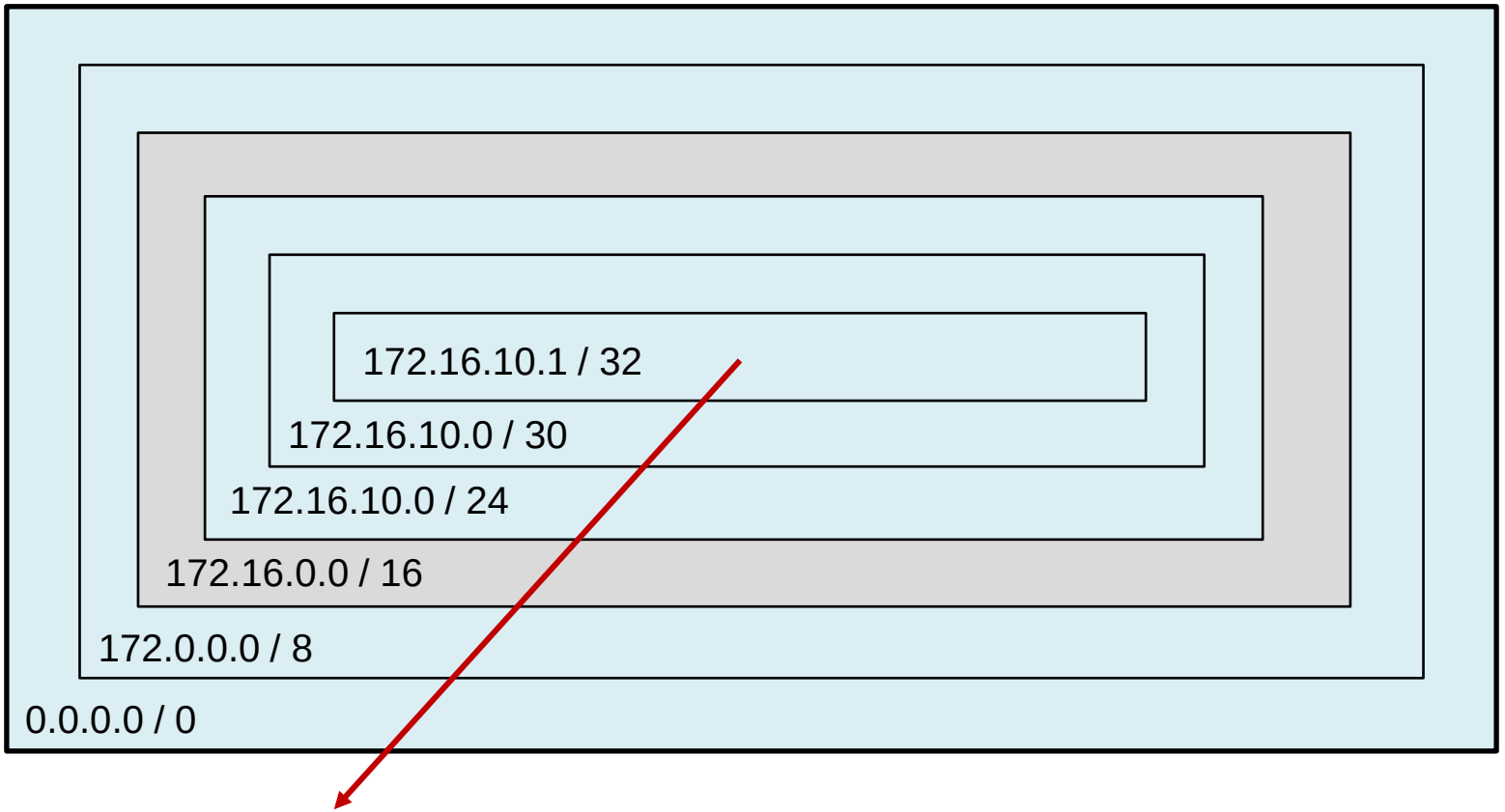
DATA：192.168.1.1

DATA：192.168.1.33

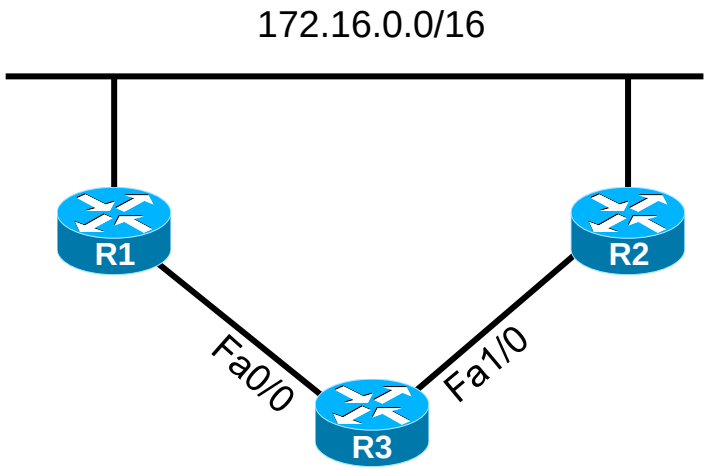
最长匹配原则

- 主机地址(主机路径)
- 子网
- 一组子网（汇总路由）
- 主网号
- 超网(CIDR)
- 缺省地址

最长匹配原则


172.16.10.1 / 32
172.16.10.0 / 30
172.16.10.0 / 24
172.16.0.0 / 16
172.0.0.0 / 8
0.0.0.0 / 0

最长匹配原则

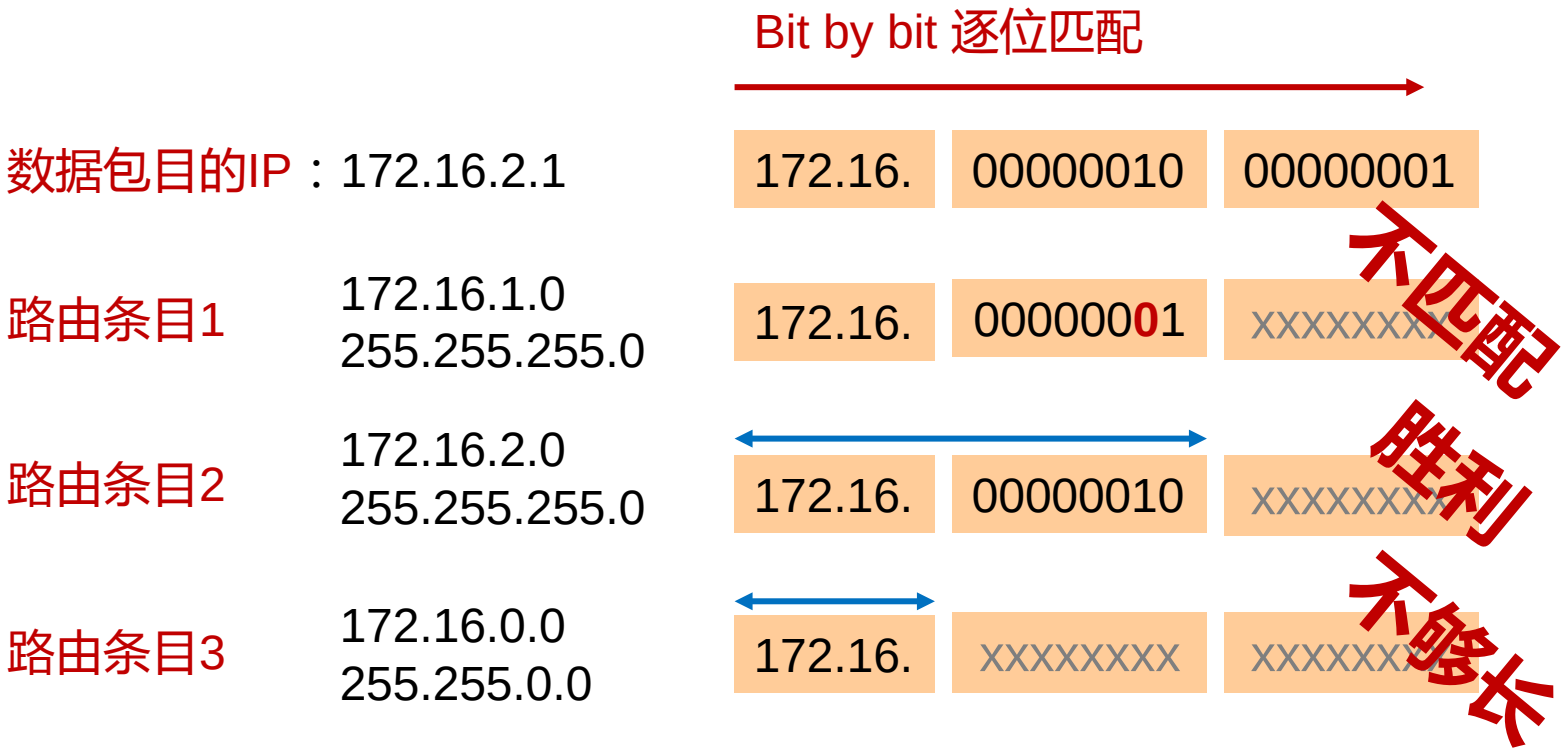


S 172.16.1.0/24 [1/0] via 192.168.13.1

S 172.16.2.0/24 [1/0] via 192.168.13.2

S 172.16.0.0/16 [1/0] via 192.168.23.2

最长匹配原则

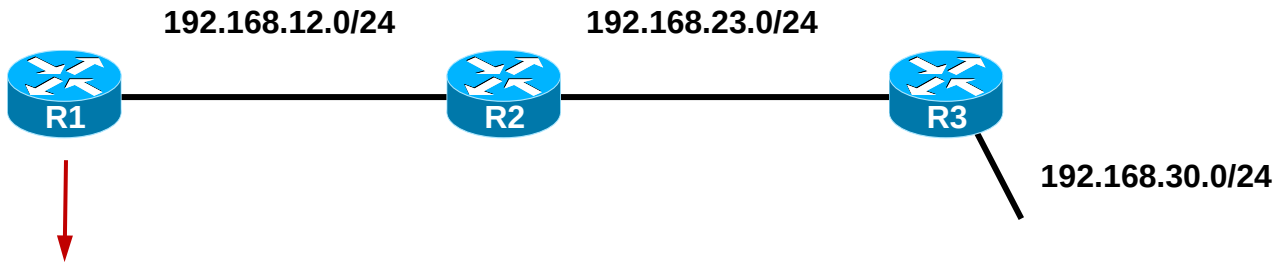



```
S  172.16.1.0/24 [1/0] via 192.168.13.1
S  172.16.2.0/24 [1/0] via 192.168.13.2
S  172.16.0.0/16 [1/0] via 192.168.23.2
```

路由选择原理

- **路由表的查找**
  - 不同的前缀，在路由表中属于不同的路由
  - 相同的前缀，通过不同的协议获取，先比AD，后比metric
  - 最长匹配，匹配，转发；不匹配，丢弃
  - 路由器的行为是逐跳的，到目标网络的沿路径每个路由器都必须有关于目标的路由
  - 数据是双向的，考虑流量的时候，要关注流量的往返。

路由表的递归查询



	Destination Network	Next-hop
S	192.168.30.0	192.168.23.3
S	192.168.23.0	192.168.12.2
C	192.168.12.0	Serial0/0

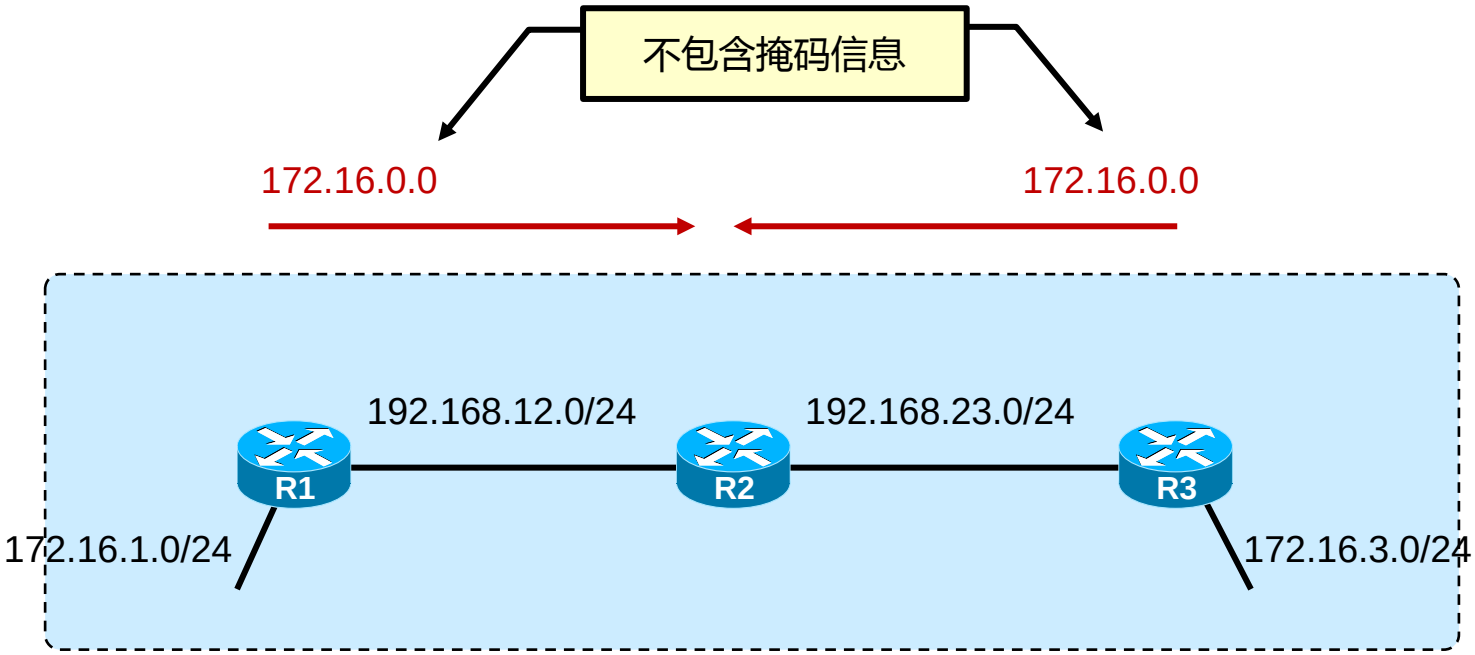
有类及无类路由协议

- 有类无类路由协议的概念
- 有类路由协议
- 无类路由协议

有类及无类路由选择协议

- **协议分类**
  - 有类路由选择协议：RIPv1、IGRP
  - 无类路由选择协议：OSPF、EIGRP、ISIS、BGP等
  - 根本区别在于：更新消息中是否包含网络掩码信息

有类路由选择协议

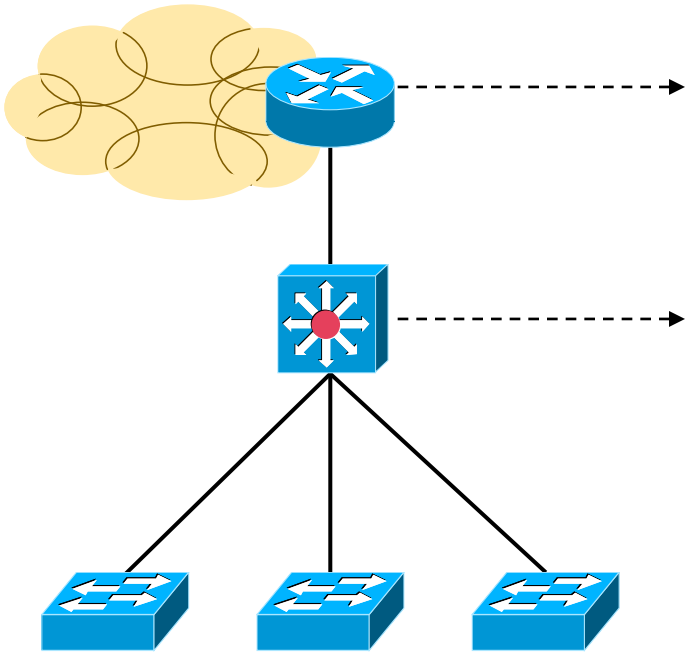

不包含掩码信息
172.16.0.0
172.16.0.0
192.168.12.0/24
192.168.23.0/24
R1
R2
R3
172.16.1.0/24
172.16.3.0/24

红茶三杯
Vinsoney

静态路由专题

- 静态路由
- 静态路由的配置
- 浮动静态路由
- 静态汇总路由

静态路由应用环境



内网网段
192.168.1.0/24
192.168.2.0/24
192.168.3.0/24

Ip route 192.168.1.0 255.255.0.0 192.168.254.1
Ip route 192.168.2.0 255.255.0.0 192.168.254.1
Ip route 192.168.3.0 255.255.0.0 192.168.254.1

Ip route 0.0.0.0 0.0.0.0 192.168.254.2

静态路由的配置

```
Ip route 192.168.10.0 255.255.255.0 192.168.1.1
```

使用指向下一跳的静态路由

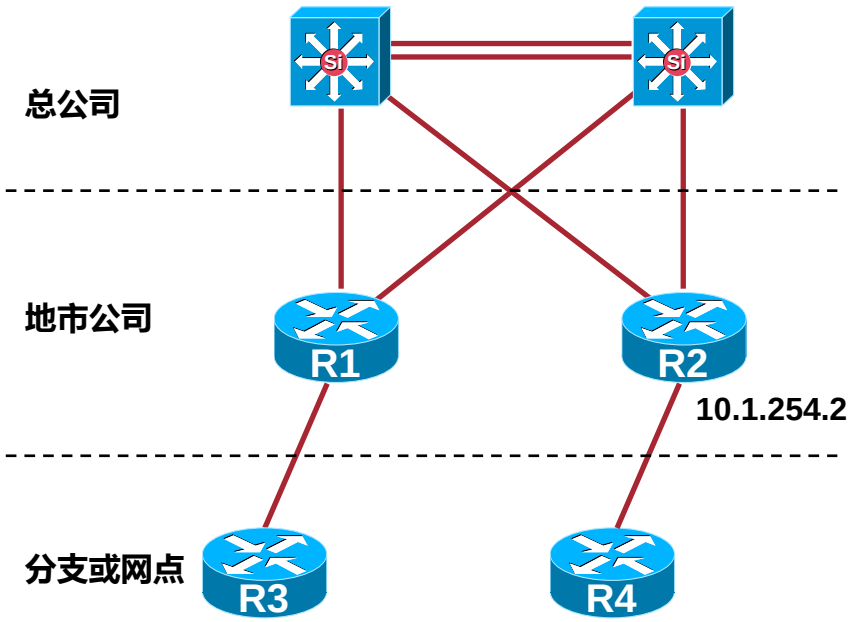
```
Ip route 192.168.10.0 255.255.255.0 fa 0/0
```

使用关联出接口的方式配置静态路由
该条目将作为直连网络输入到路由表中
如果出接口为广播型接口，可能会给接口下的节点造成额外的负担（ARP）

```
Router#show ip route
Gateway of last resort is not set
     9.0.0.0/24 is subnetted, 2 subnets
C       9.9.12.0 is directly connected, FastEthernet0/0
S       9.9.23.0 is directly connected, FastEthernet0/0
```

静态路由

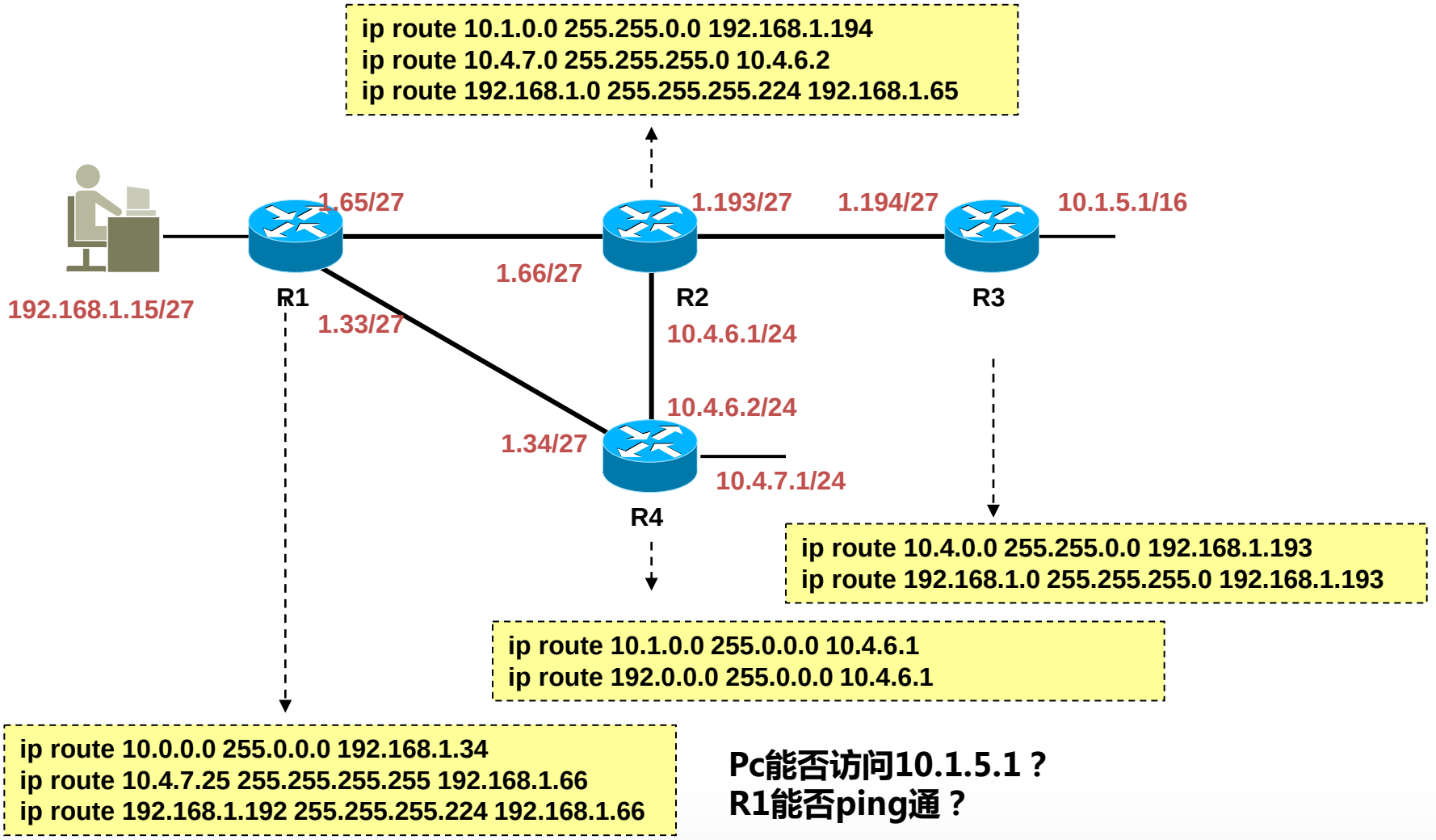
- 【案例】xx公司网点升级项目



R3、R4的配置：
ip route 10.1.0.0 255.255.0.0 gi 0/0

R1、R2为替换上的新设备(x厂商)
替换前后配置没变
但升级后分支公司及网点无法访问总公司服务器资源

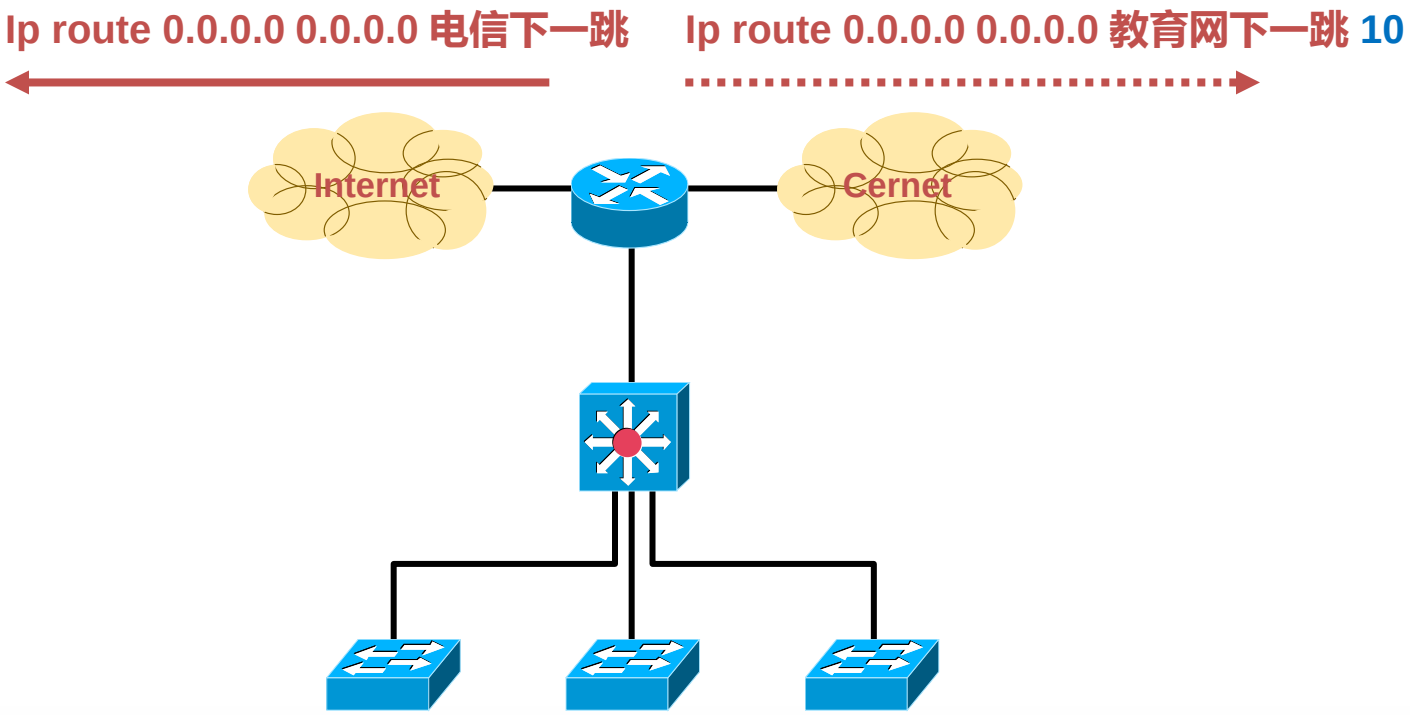
静态路由



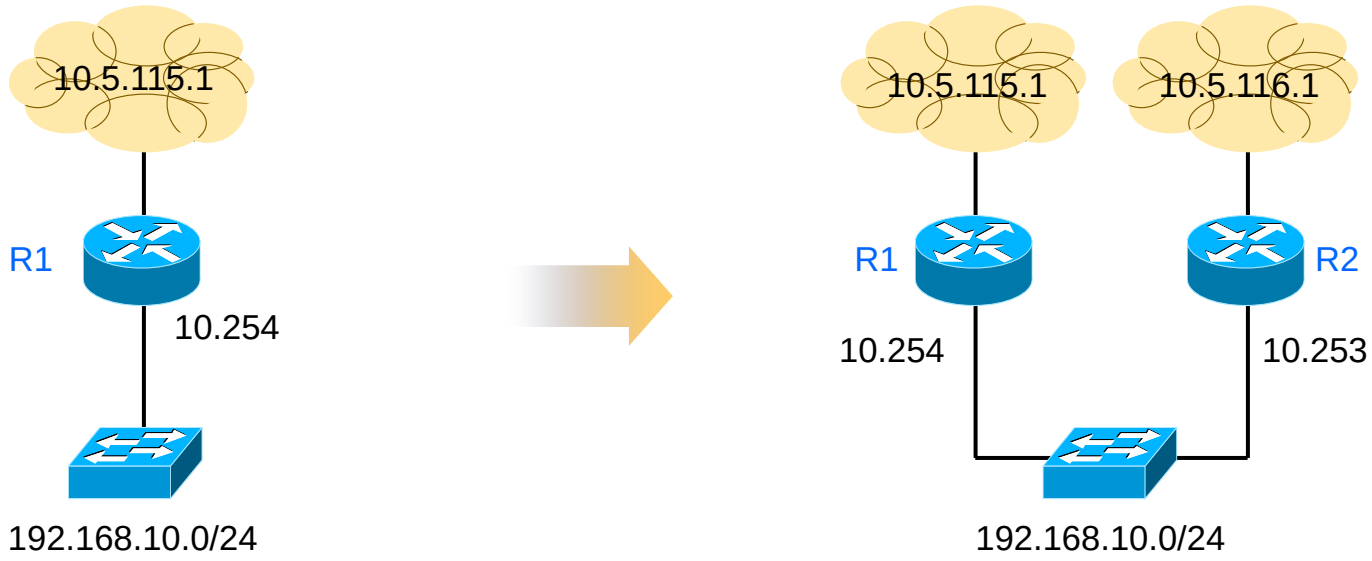
Pc能否访问10.1.5.1？
R1能否ping通？

浮动静态路由

将本条静态路由的AD值调整为10。如此一来当指向电信出口的默认路由正常时，这条指向教育网下一跳的默认路由就不会出现在路由表中，只有当前者失效时，该条路由才会“浮现”出来，故称为浮动路由。



静态路由 案例



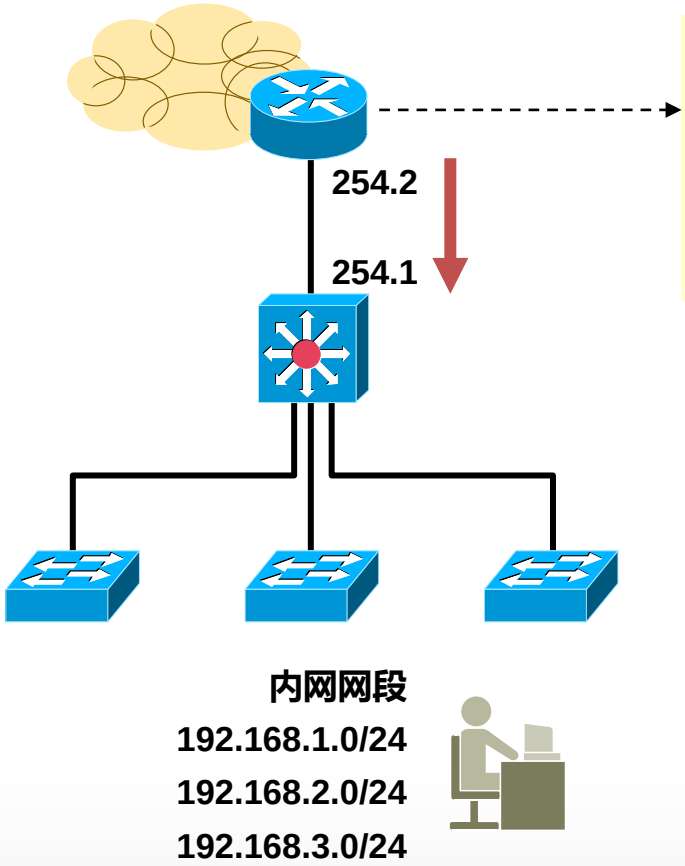
R1

Ip route 0.0.0.0 0.0.0.0 10.5.115.1
Ip route 0.0.0.0 0.0.0.0 192.168.10.253 10

R2

Ip route 0.0.0.0 0.0.0.0 10.5.116.1
Ip route 0.0.0.0 0.0.0.0 192.168.10.254 10

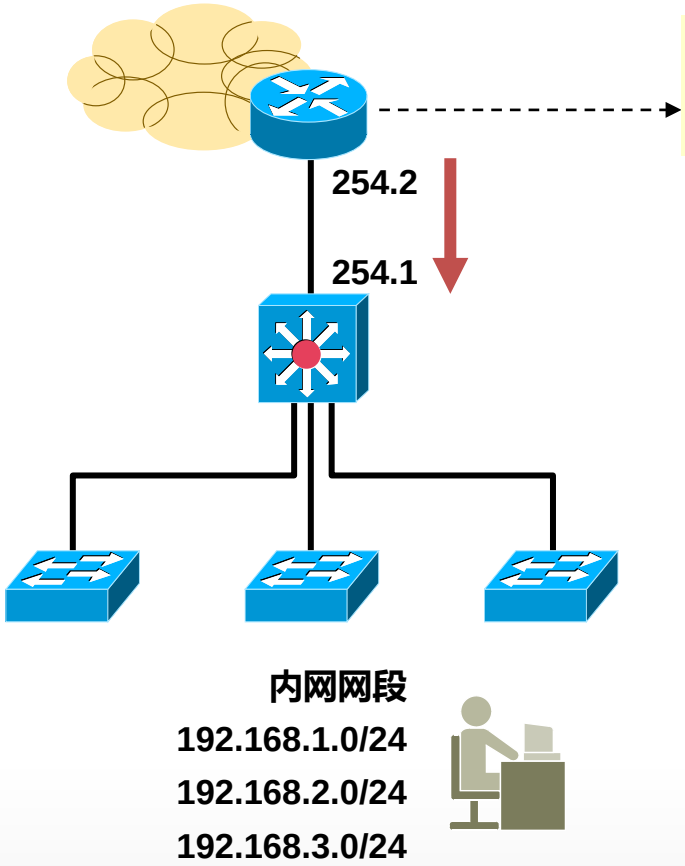
静态路由汇总




```
Ip route 0.0.0.0 0.0.0.0 202.101.100.2
Ip route 192.168.1.0 255.255.255.0 192.168.254.1
Ip route 192.168.2.0 255.255.255.0 192.168.254.1
Ip route 192.168.3.0 255.255.255.0 192.168.254.1
```

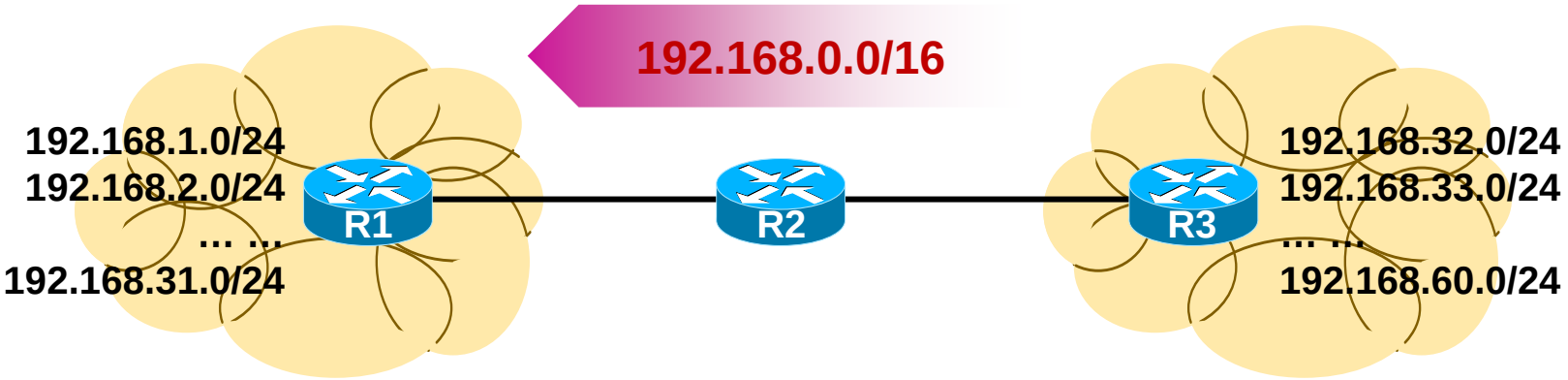
静态路由汇总

- 改变子网掩码，通过汇总路由匹配明细，从而简化路由表




```
Ip route 0.0.0.0 0.0.0.0 202.101.100.2
Ip route 192.168.0.0 255.255.0.0 192.168.254
```

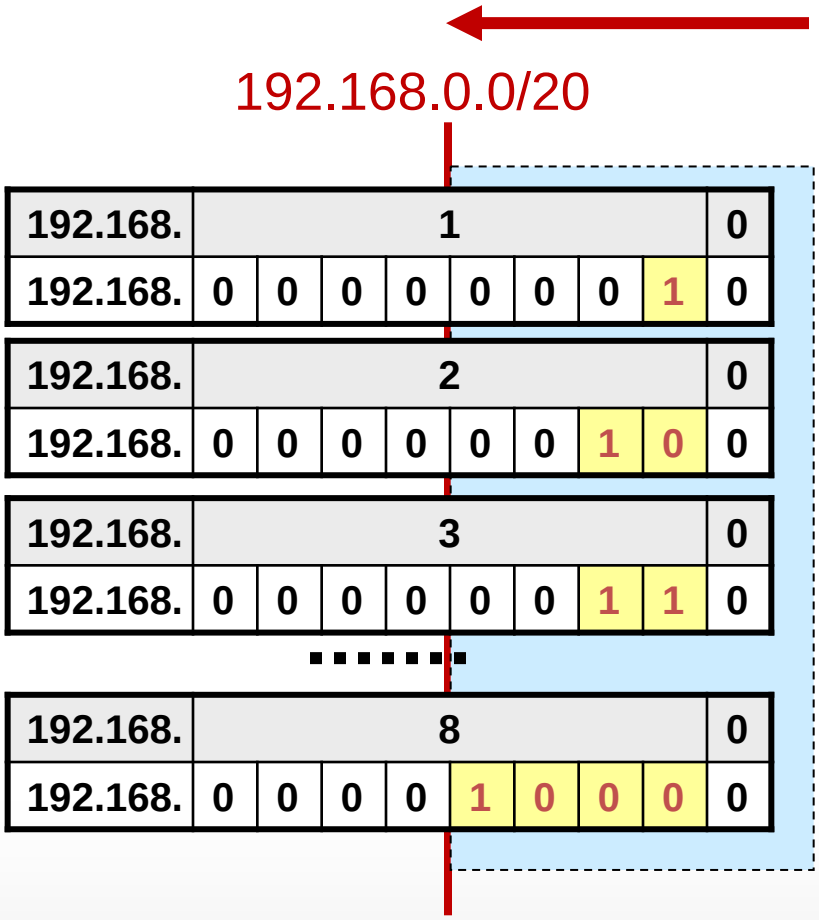
静态路由汇总



静态汇总路由

- 改变子网掩码，通过汇总路由匹配明细，从而简化路由表

192.168.1.0/24
192.168.2.0/24
……
192.168.8.0/24



静态汇总路由

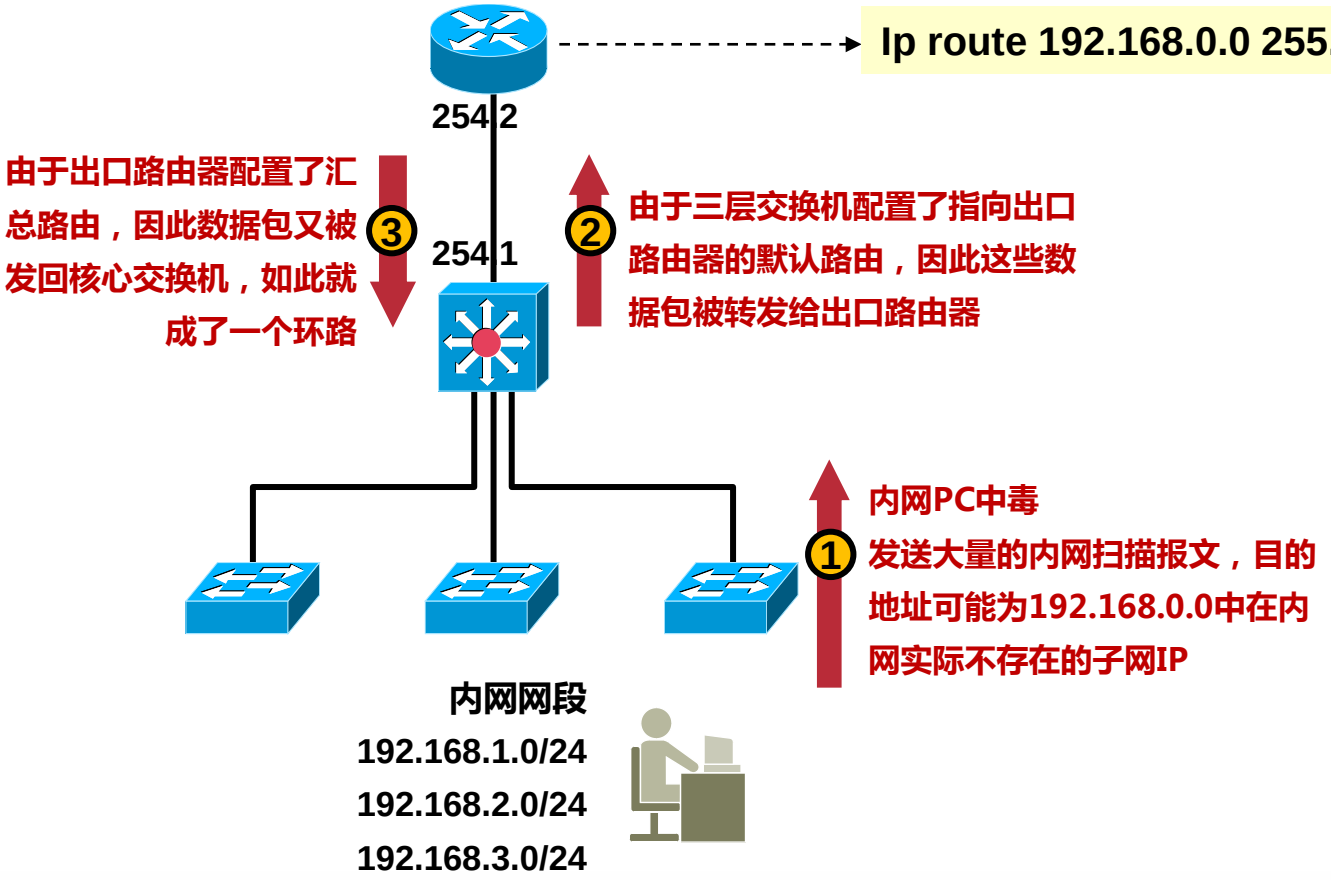
- 改变子网掩码，通过汇总路由匹配明细，从而简化路由表

172.16.32.0/24
172.16.33.0/24
172.16.36.0/24
……
172.16.47.0/24

汇总地址是？

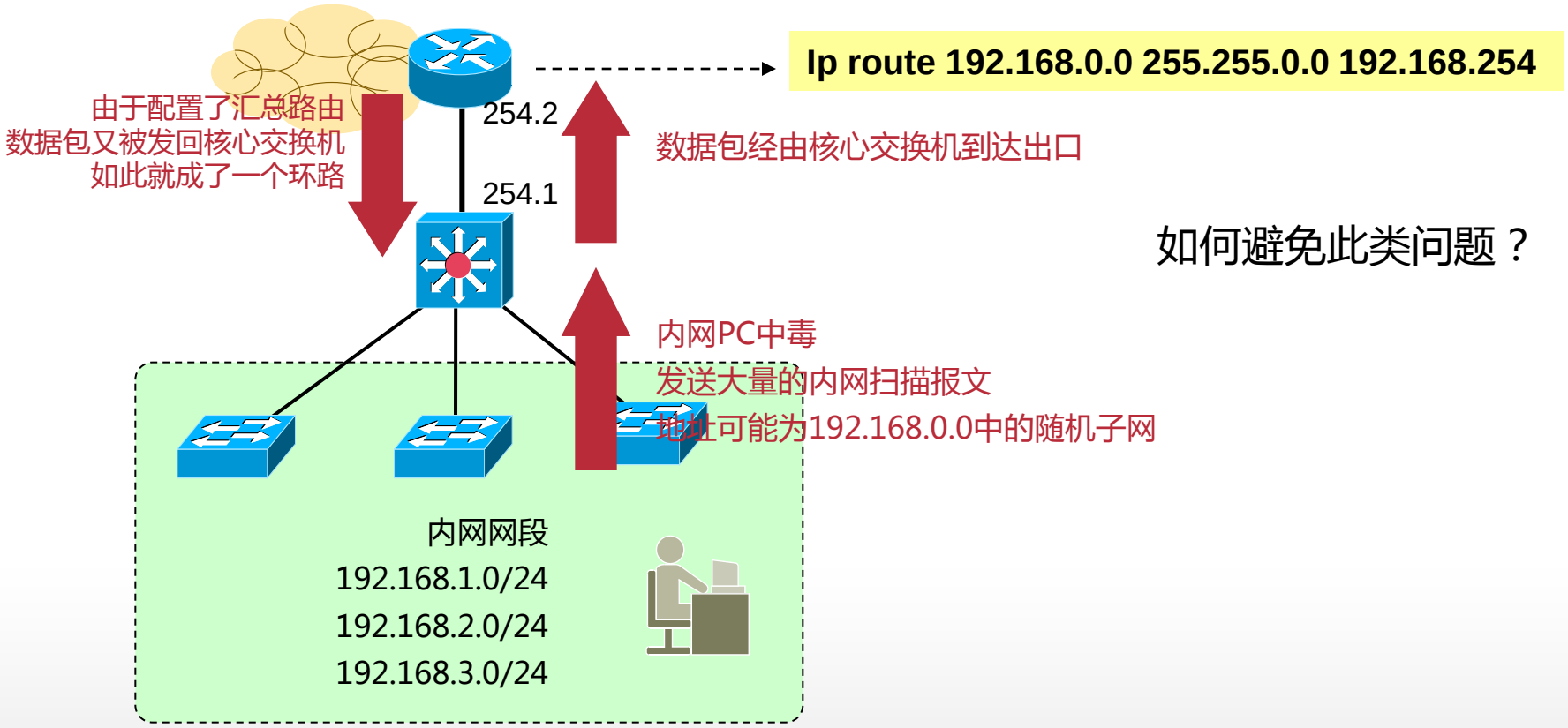
172.16.	32								0
172.16.	0	0	1	0	0	0	0	0	0
172.16.	33								0
172.16.	0	0	1	0	0	0	0	1	0
172.16.	36								0
172.16.	0	0	1	0	0	1	0	0	0
▪▪▪▪▪▪▪									
172.16.	47								0
172.16.	0	0	1	0	1	1	1	1	0

静态汇总路由存在的隐患



静态汇总路由

- **静态汇总路由存在的隐患**
  - 路由环路的产生



红茶三杯
Vinsoney

学习 沉淀 成长 分享

关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

EIGRP

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2012-08-01

课程目标

- EIGRP工作机制
- EIGRP的配置和验证
- EIGRP认证
- 优化EIGRP实施

EIGRP协议特点

- CISCO私有的增强型的距离矢量路由协议
- 快速汇聚：采用DUAL来实现快速汇聚
- 触发更新
- 按需更新：EIGRP发送部分更新，把更新的部分传递给需要的路由器
- 支持多种网络层协议：使用协议无关模块来支持
- 使用多播和单播：使用多播和单播而不是广播，多播地址224.0.0.10
- 支持VLSM：支持无类
- 精密的度量值：能实现不等价的负载均衡

EIGRP关键技术

- **邻居发现协议**

  使用Hello包发现邻居，并动态的获悉其直连的网络中的其他路由器

- **可靠传输协议（RTP）**

  确保EIGRP分组按顺序以可靠的方式传输给所有邻居

- **DUAL有限状态机**

  选择最低的度量值和无环的路径到达目的网段

- **协议无关模块**

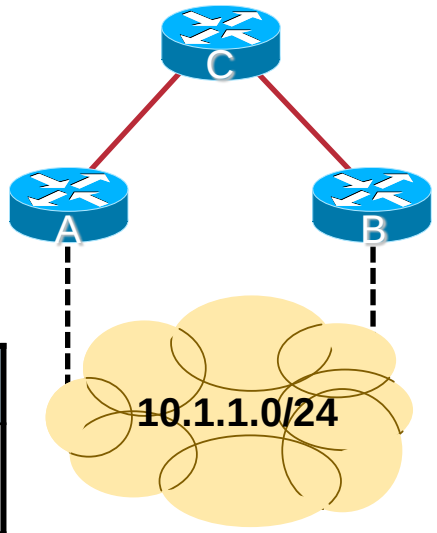
  EIGRP支持IP、IPv6、Apple talk和IPX，其都有独立EIGRP模块，负责处理网络层协议而异的需求。

EIGRP的三张表

IP EIGRP Neighbor table	
Next-hop Router	Interface
Router A Router B	Eth0 Eth1

IP EIGRP Topology table			
Network	Feasible Distance (EIGRP metric)	Advertised Distance	EIGRP Neighbor
10.1.1.0/24 10.1.1.0/24	2000 2500	1000 1500	Router A (E0) Router B (E1)

IP Routing table			
Network	Metric (Feasible Distance)	Outbound Interface	Next hop (EIGRP neighbor)
10.1.1.0/24	2000	E0	RouterA



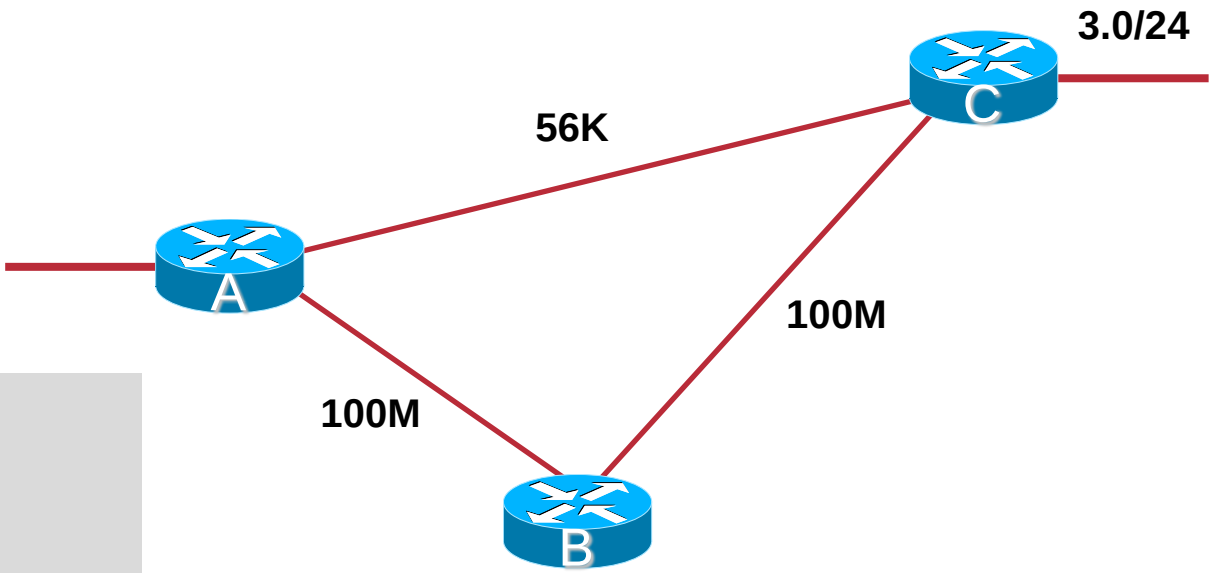
EIGRP数据包

HELLO分组	以224.0.0.10发送，无需确认Hello包 EIGRP依靠分组来发现， 验证和重新发现邻居router 以固定的时间发送hello包，该时间间隔与接口带宽有关 LAN上默认为5s
更新	EIGRP协议的这些更新数据包只在必要的时候传递必要的信息,而且仅仅传递给需要路由信息的路由器。当只有某一指定的路由器需要路由更新时,更新数据包就是单播发送的;当有多台路由器需要路由更新时,更新数据包就是组播发送的 以可靠的方式发送，需要确认
查询	当某条路由丢失，向邻居查询关于路由信息，通常靠组播方式发送，有时也用单播重传；可靠地发送
应答	响应查询分组，单播；可靠方式发送
确认（ACK）	以单播发送的HELLO包(不包含数据)，包含确认号。用来确认更新、查询和应答。ACK本身不需确认。

EIGRP Timers

Hello Time	LAN 上默认的Hello时间为5s；在点到点链路（PPP、HDLC、p2p的FR链路、ATM子接口等）及带宽高于T1的多点电路（如ISDN PRI、SMDS、ATM和FR），默认的hello时间也是5S；在低速链路（如带宽低于T1的多点电路，ISDN BRI、FR、SMDS、ATM和X.25），hello时间默认60S 接口上ip hello-interval eigrp xx
Hold Time	hold time指出在多长时间未收到邻居的hello包和其他eigrp分组时，将邻居视为down状态。 默认hold time= 3*Hello time 接口上 ip hold-time eigrp xxx
SRTT	平滑回程时间，eigrp计算时间，是指一各数据包发出去开始，直到对方给我确认的时间。该定时器用于确定重传间隔RTO。
RTO	在router上维护着一个邻居表，当一个数据包(update包)发给邻居后，在RTO(单位是毫秒)重传间隔时间过后邻居还没有确认，那么该router就会给这个邻居重传。重传的包是单播的.当重传的次数到达16次，该router会reset他们的邻居关系。
active-time	当routerA中一条路由上的后继路由器失效了，且没有fs，该路由会标记为active 这时router A会向除该出问题的邻居router之外的所有邻居查询， 邻居如果没有该信息，它也会向它的邻居查询..。在默认3分钟（180S）的活跃时间内 (可使用 timers active-time xxmini disable来更改)， 如果被查询的router没有回应，查询的路由就会被置于 stuck in ative 状态。
Multicast flow timer	组播流定时器指定了从组播切换到单播之前，等待邻居ACK分组的时间。

Metric的计算



Bandwidth

Delay

Reliability

Loading

MTU

Metric的计算

$$BW = \frac{10^7}{\text{接口最小带宽kbit/s}} \times 256 \ (\text{kbit/s})$$

带宽取值沿路所有数据沿路出接口（或路由入口）的接口带宽的最小值

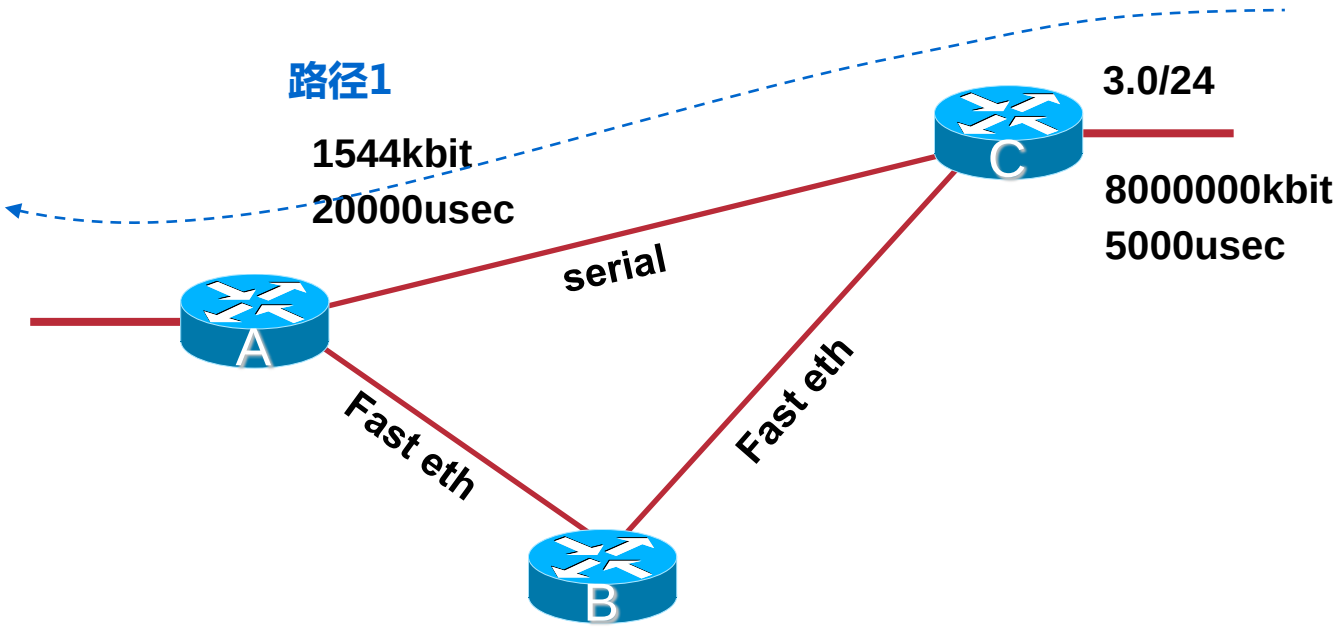
$$DLY = \frac{\text{延迟(us)}}{10} \times 256 \ (\text{us})$$

延迟取值沿路所有数据沿路出接口（或路由入口）的接口延迟的累加

$$Metric = \left[K1 \times BW + \frac{K2 \times BW}{256 - LOAD} + K3 \times DLY \right] \times \left[\frac{K5}{RELIA + K4} \right]$$

- 默认 K1 = 1, K2 = 0, K3 = 1, K4 = 0, K5 = 0
- EIGRP路由metric默认为 **延迟+带宽**

Metric的计算



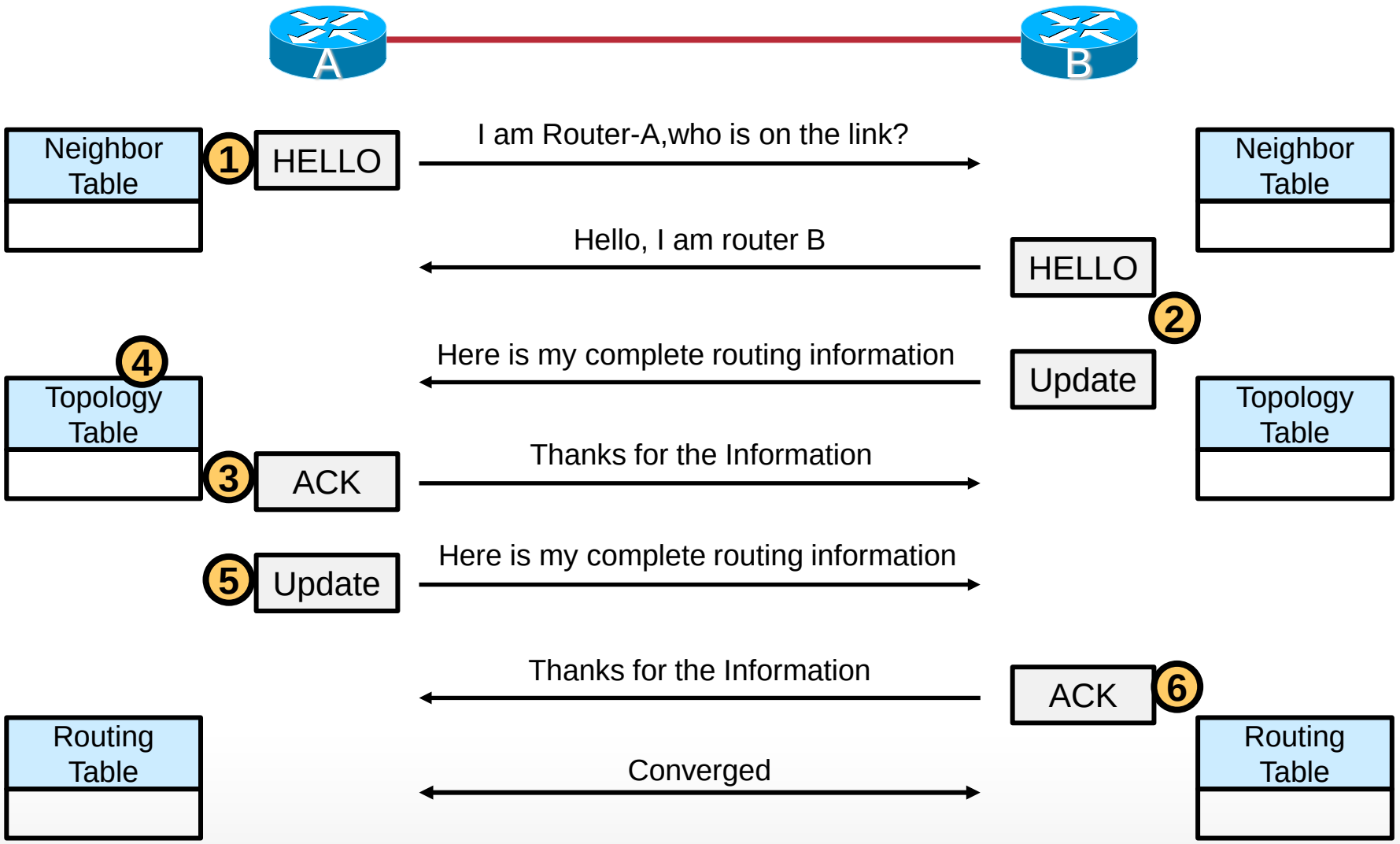
A路由器上，看到的3.3.3.0/24的路由metric（路径1）？

BW = 10^7 / 1544 * 256 = 6476(去掉小数) * 256 = 1657856

DLY = 20000/10 * 256 + 5000/10 * 256 = 640000

Metric = 640000 + 1657856 = 2297856

邻居关系的建立过程

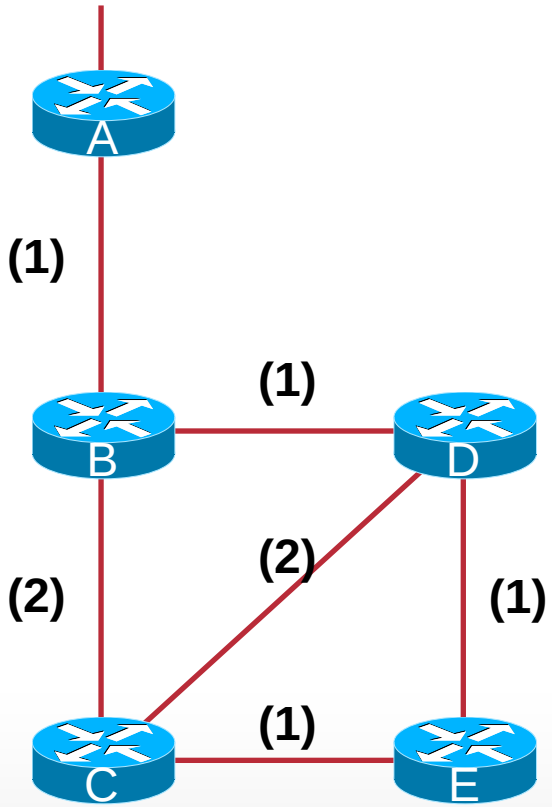


DUAL扩散更新算法

- **DUAL**
  - Diffusing Update Algorithm 用于计算最佳无环路径和备用路径
- **几个术语：**
  - 后继路由器
  - 可行距离 (FD)
  - 可行后继路由器 (FS)
  - 通告距离 (AD)
  - 可行条件，或称可行性条件 (FC)

DUAL算法示例-1

1.0/24网段



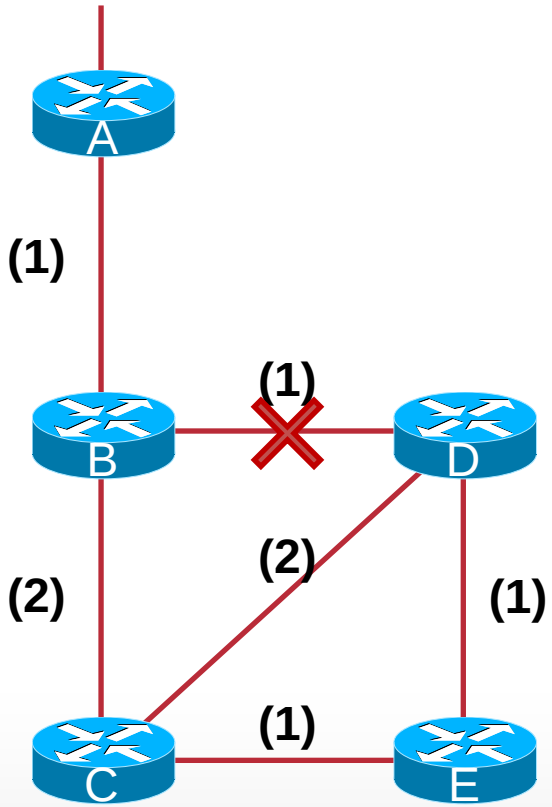
C (1.0)	EIGRP	FD	AD	Topology
		3		(FD)
	via B	3	1	(Successor)
	via D	4	2	(FS)
	via E	4	3	

D (1.0)	EIGRP	FD	AD	Topology
		2		(FD)
	via B	2	1	(Successor)
	via C	5	3	

E (1.0)	EIGRP	FD	AD	Topology
		3		(FD)
	via D	3	2	(Successor)
	via C	4	3	

DUAL算法示例-2

1.0/24网段

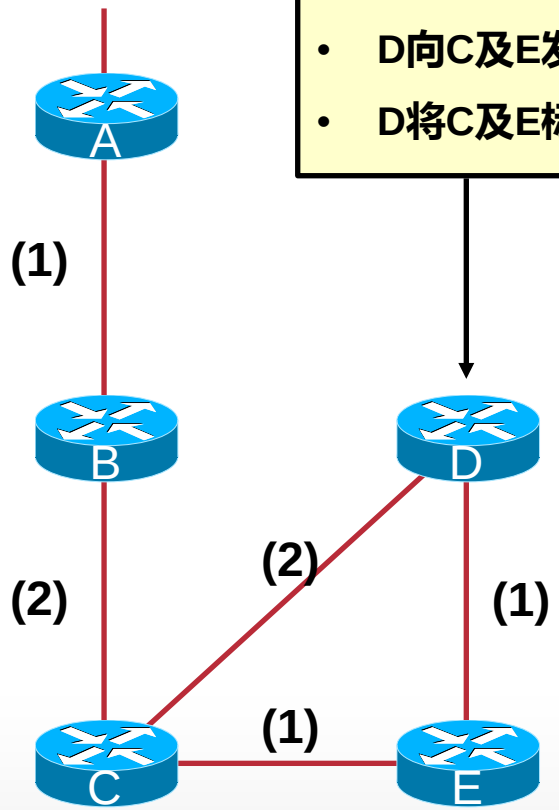


C (1.0)	EIGRP	FD	AD	Topology
		3		(FD)
	via B	3	1	(Successor)
	via D	4	2	(FS)
	via E	4	3	

D (1.0)	EIGRP	FD	AD	Topology
		2		(FD)
	~~via B~~	~~2~~	~~1~~	~~(Successor)~~
	via C	5	3	

E (1.0)	EIGRP	FD	AD	Topology
		3		(FD)
	via D	3	2	(Successor)
	via C	4	3	

DUAL算法示例-3



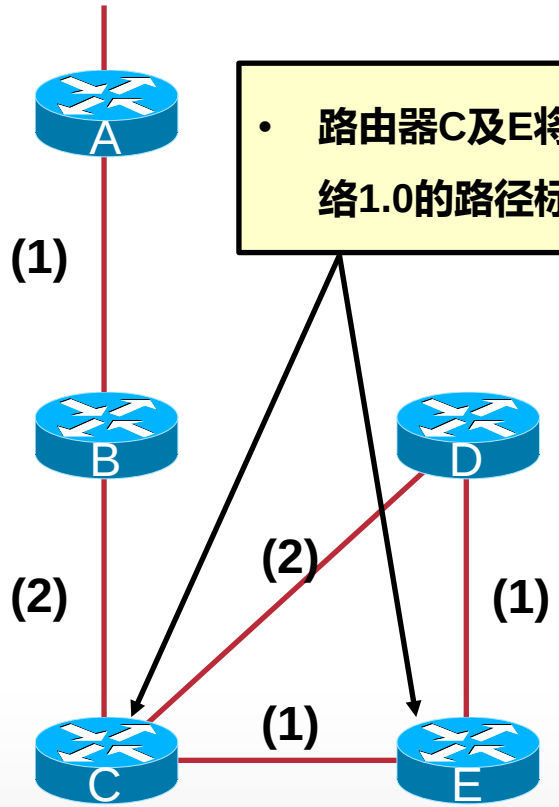
- 将1.0的metric设置为不可达（-1表示不可达）
- 由于没有FS，1.0被标记为Active状态
- D向C及E发送查询信息，询问去往1.0的替代路径
- D将C及E标记为未应答查询（q）

C	EIGRP	FD	AD	Topology
(1.0)				(FD)
				(Successor)
				(FS)

D	EIGRP	FD	AD	Topology
(1.0)		-1		active
	via E			(q)
	via C	5	3	(q)

E	EIGRP	FD	AD	Topology
(1.0)		3		(FD)
	via D	3	2	(Successor)
	via C	4	3	

DUAL算法示例-4



- 路由器C及E将经由D路由器前往网络1.0的路径标记为不可达

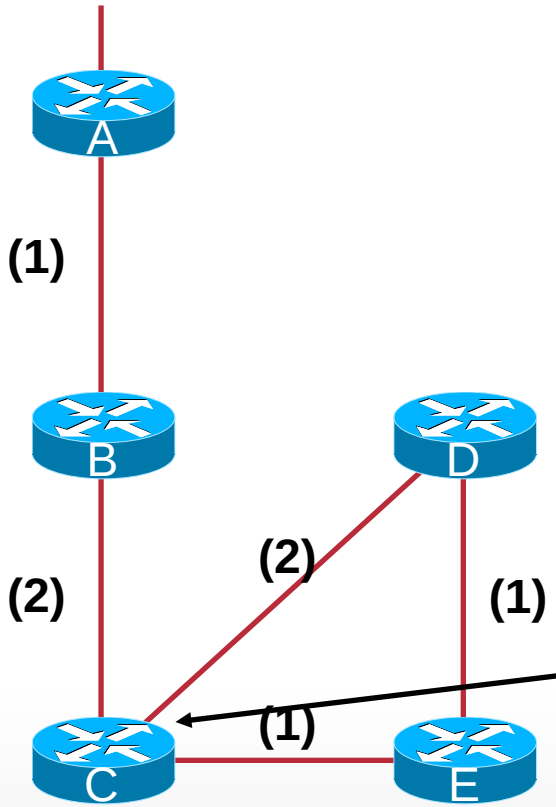
C (1.0)	EIGRP	FD	AD	Topology
		3		(FD)
	via B	3	1	(Successor)
	~~via D~~	~~4~~	~~2~~	~~(FS)~~
	via E	4	3	

RP (1.0)		FD	AD	Topology
		-1		active
	via E			(q)
	via C	5	3	(q)

E (1.0)	EIGRP	FD	AD	Topology
		3		(FD)
	~~via D~~	~~3~~	~~2~~	~~(Successor)~~
	via C	4	3	

DUAL算法示例-5

1.0/24网段



C (1.0)	EIGRP	FD	AD	Topology
		3		(FD)
	via B	3	1	(Successor)
	via D			
	via E			

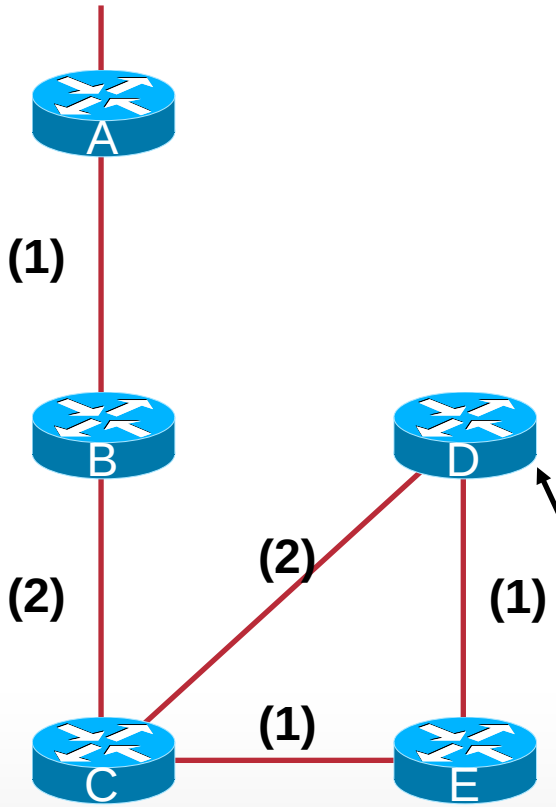
D (1.0)	EIGRP	FD	AD	Topology
		-1		active
	via E			(q)
	via C	5	3	(q)

E (1.0)	EIGRP	FD	AD	Topology
		3		(FD)

- **路由器C发送应答消息给D，指出到达1.0网段的路径没变**

DUAL算法示例-6

1.0/24网段



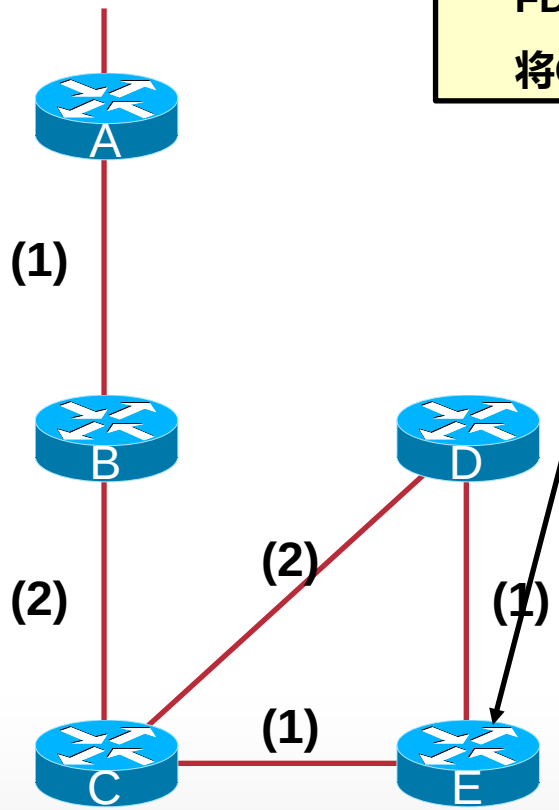
C (1.0)	EIGRP	FD	AD	Topology
		3		(FD)
	via B	3	1	(Successor)
	via D			
	via E			

D (1.0)	EIGRP	FD	AD	Topology
		-1		active
	via E			(q)
	via C	5	3	

E (1.0)	EIGRP	FD	AD	Topology
		3		(FD)

- **D收到C的应答，将C的查询未应答标记删除**
- **保持前往1.0路由的active状态，同时等待E的应答**

DUAL算法示例-7



- **E由于从C前往1.0网段的AD值=3，不小于原来的FD=3，所以E将路由标记为active，且向C查询，并将C标记为查询未应答**

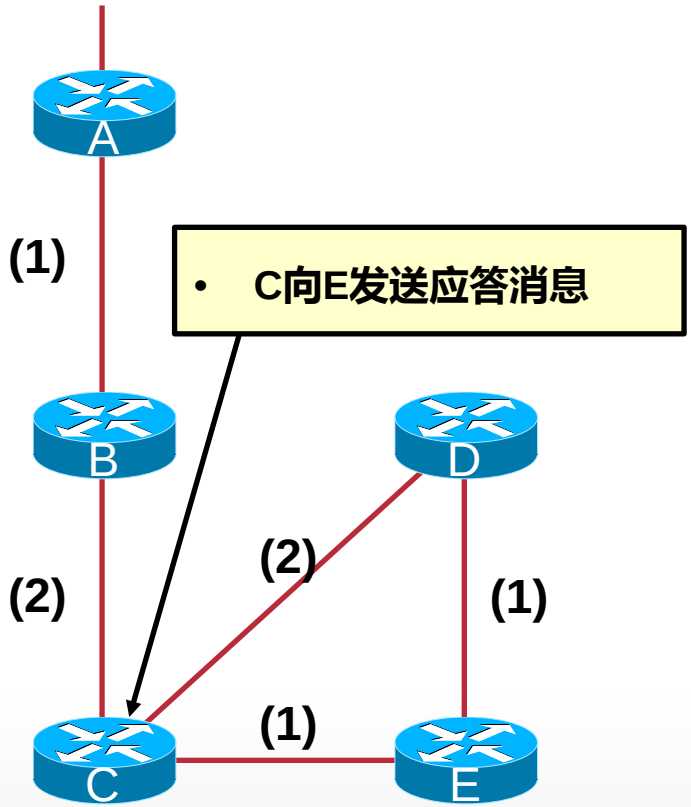
C	EIGRP	FD	AD	Topology
				ccessor)

D	EIGRP	FD	AD	Topology
(1.0)		-1		active
	via E			(q)
	via C	5	3	

E	EIGRP	FD	AD	Topology
(1.0)		-1		active
	via D			
	via C	4	3	(q)

DUAL算法示例-8

1.0/24网段



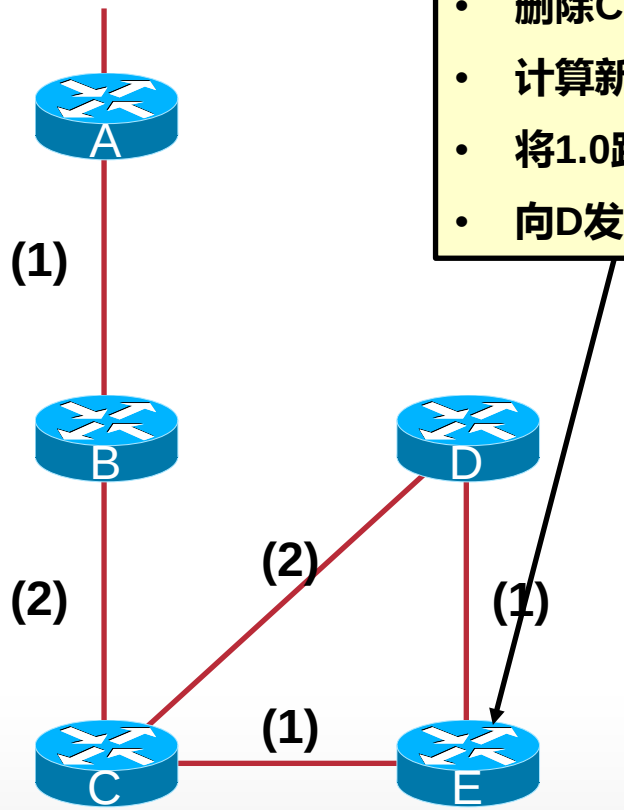
- C向E发送应答消息

C (1.0)	EIGRP	FD	AD	Topology
		3		(FD)
	via B	3	1	(Successor)
	via D			
	via E			

D (1.0)	EIGRP	FD	AD	Topology
		-1		active
	via E			(q)
	via C	5	3	

E (1.0)	EIGRP	FD	AD	Topology
		-1		active
	via D			
	via C	4	3	(q)

DUAL算法示例-9



- E收到C发送的应答消息
- 删除C的查询未应答标记
- 计算新的FD，把后继路由加入到拓扑表
- 将1.0路由切换到passvie状态
- 向D发送应答消息

C	EIGRP	FD	AD	Topology
				(FD)
			1	(Successor)

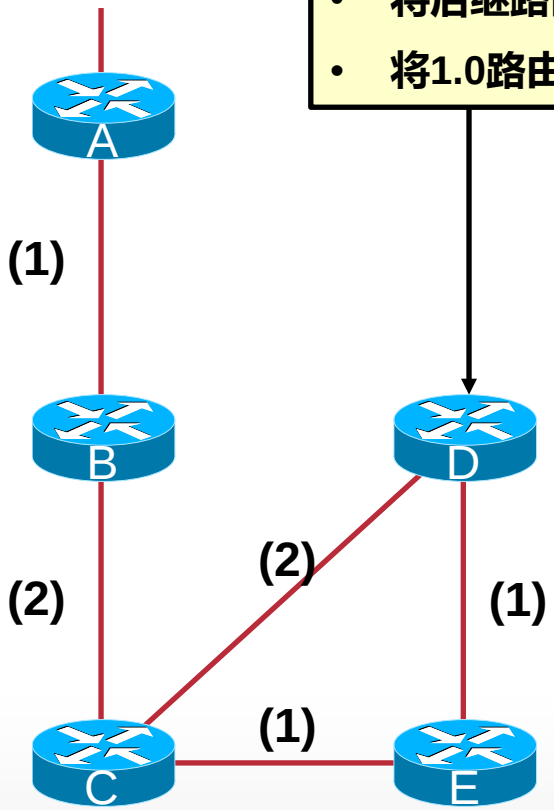
		AD	Topology
(1.0)	-1		active
via E			(q)
via C	5	3	

E	EIGRP	FD	AD	Topology
(1.0)		4		
	via C	4	3	(successor)
	via D			

DUAL算法示例-10

- D收到E的应答消息
- 删除E的查询未应答标记
- 计算新的FD
- 将后继路由加入到拓扑表
- 将1.0路由切换到 passive

1.0/24网段



	EIGRP	FD	AD	Topology
		3		(FD)
	via B	3	1	(Successor)
	via D			
	via E			

D (1.0)	EIGRP	FD	AD	Topology
		5		
	via E	5	4	(successor)
	via C	5	3	(successor)

E (1.0)	EIGRP	FD	AD	Topology
		4		
	via C	4	3	(successor)
	via D			

EIGRP基本配置

- 基本配置及验证
- Passive-interface
- 默认路由的传递
- 路由汇总
- 负载均衡
- SIA及限制EIGRP查询

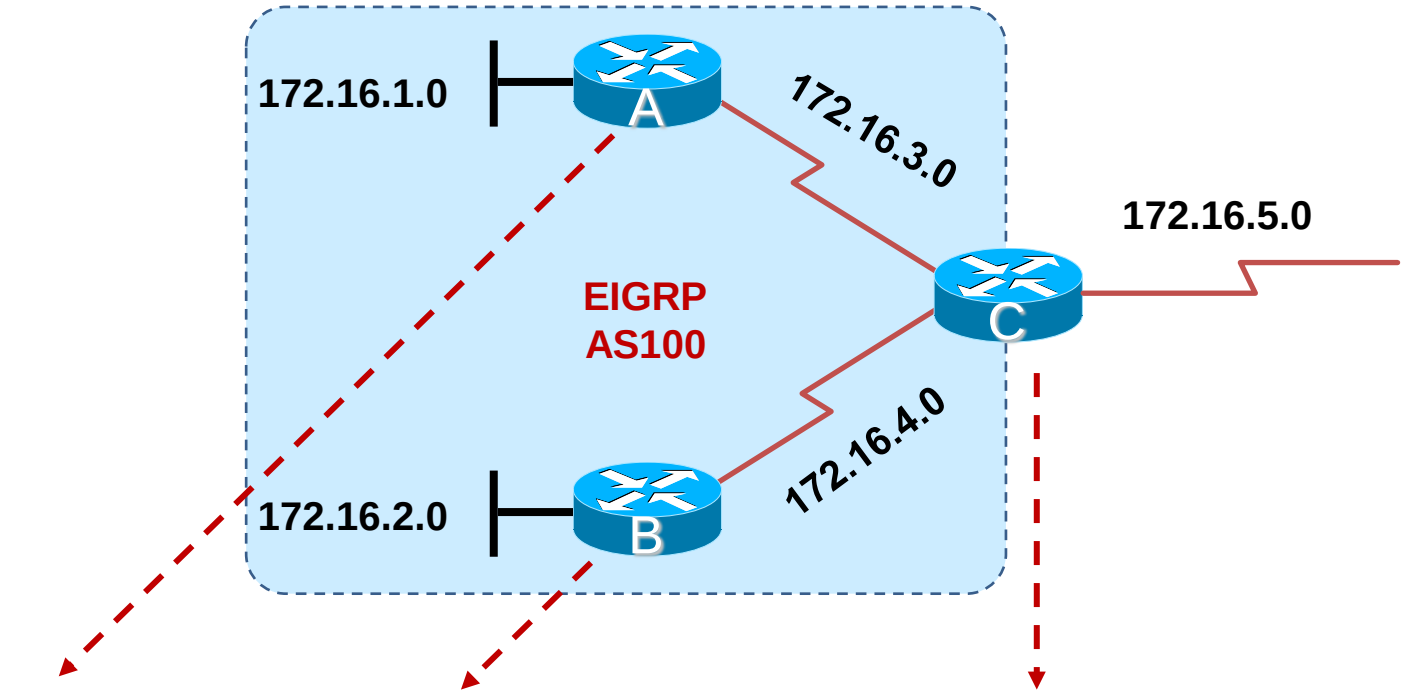
基本配置

Router(config)# router eigrp autonomous-system

EIGRP 将*autonomous-system*参数称为“自治系统”编号。

Router(config-router)# network network-number [wildcard-mask]

基本配置 示例




```
router eigrp 100
  network 172.16.0.0
```

```
router eigrp 100
  network 172.16.3.0 0.0.0.255
  network 172.16.4.0 0.0.0.255
```

查看EIGRP的运行情况

- show ip eigrp neighbors

```
R1#sh ip eigrp neighbor
IP-EIGRP neighbors for process 1
H   Address         Interface   Hold Uptime (sec)   SRTT(ms)   RTO   Q Cnt   Seq Num
0   192.168.12.2    Se0/0       11 00:00:03         1          3000  2       0
```

- show ip eigrp topology

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(192.168.12.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 2.0.0.0/8, 1 successors, FD is 2297856
        via 192.168.12.2 (2297856/128256), Serial0/0
P 192.168.12.0/24, 1 successors, FD is 2169856
        via Connected, Serial0/0
```

查看EIGRP的运行情况

- **show ip route eigrp**
- **show ip protocols**
- **show ip eigrp traffic**
- **show ip eigrp interface**

Passive-interface配置

- **被动接口的配置：**

```
Router(config-router)#passive-interface {type number} | default
```

  - 该命令用于将特定接口设置为被动状态；default关键字将所有路由器接口设置为被动状态

- **被动接口作用：**
  - 禁止通过被动接口建立邻接关系
  - 禁止通过被动接口接收或发送路由更新
  - 让EIGRP进程通告被动接口连接的子网

- **查看：**
  - show ip protocols
  - show ip eigrp neighbor

EIGRP默认路由

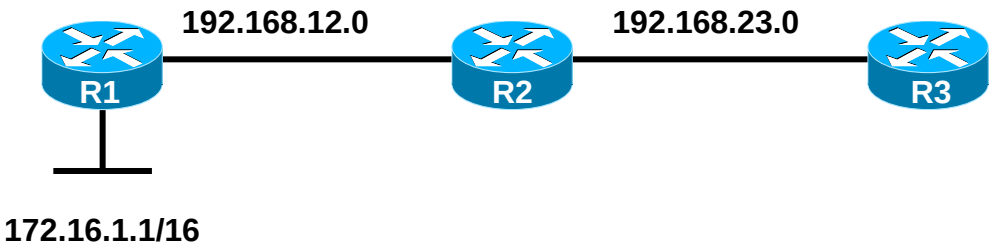
将路由表中某个网络宣告为缺省网络

```
Router(config)# Ip default-network network-number
```

将指定的网络号通告给其他的路由器

```
Router(config-router)# network network-number
```

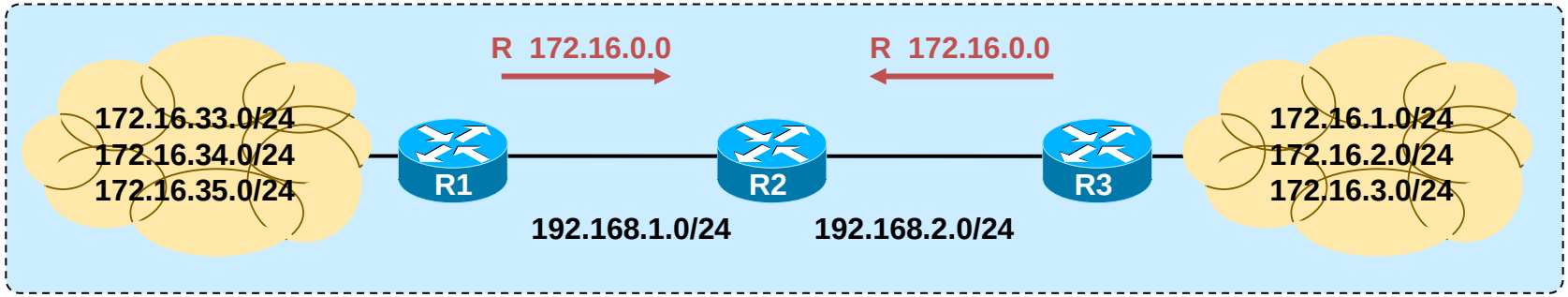
EIGRP默认路由 配置示例



R1(config)# Ip default-network network-number

EIGRP路由汇总

- **自动汇总**



EIGRP路由汇总

- **手工汇总**

关闭自动汇总

```
Router(config-router)# no auto-summary
```

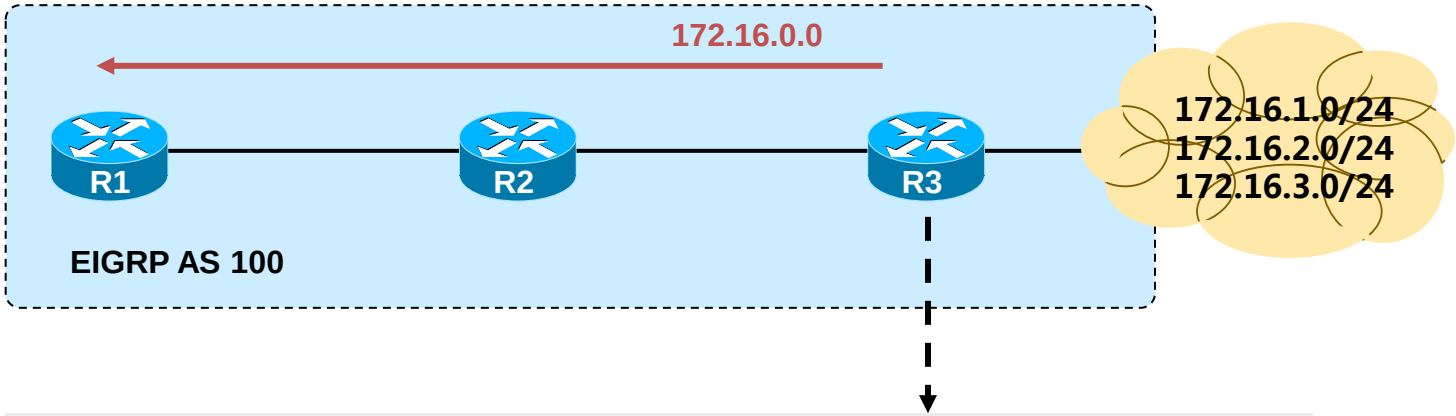
配置手工汇总

```
Router(config-if)# ip summary-address eigrp as-number address mask [admin-distance]
```

- 手工配置汇总时，仅当路由选择表中至少有一条该汇总路由的明细路由时，汇总路由才被通告出去。
- ip summary-address eigrp进行汇总的路由AD=5

EIGRP路由汇总

- **手工汇总**




```
Router eigrp 100
  no auto-summary
interface e0
 ip summary-address eigrp 100 172.16.0.0 255.255.0.0
```

路由表发生了什么变化？

EIGRP负载均衡

- **等价负载均衡**
  - EIGRP在度量值相同的所有路径之间分配数据流量
  - 默认为4条等价路径之间均衡IP负载，最大可为16条

```
Router(config-router)# Maximum-paths maximum-path
```

EIGRP负载均衡

- **非等价负载均衡**
  - EIGRP也能在度量值不同的多条路径之间负载均衡

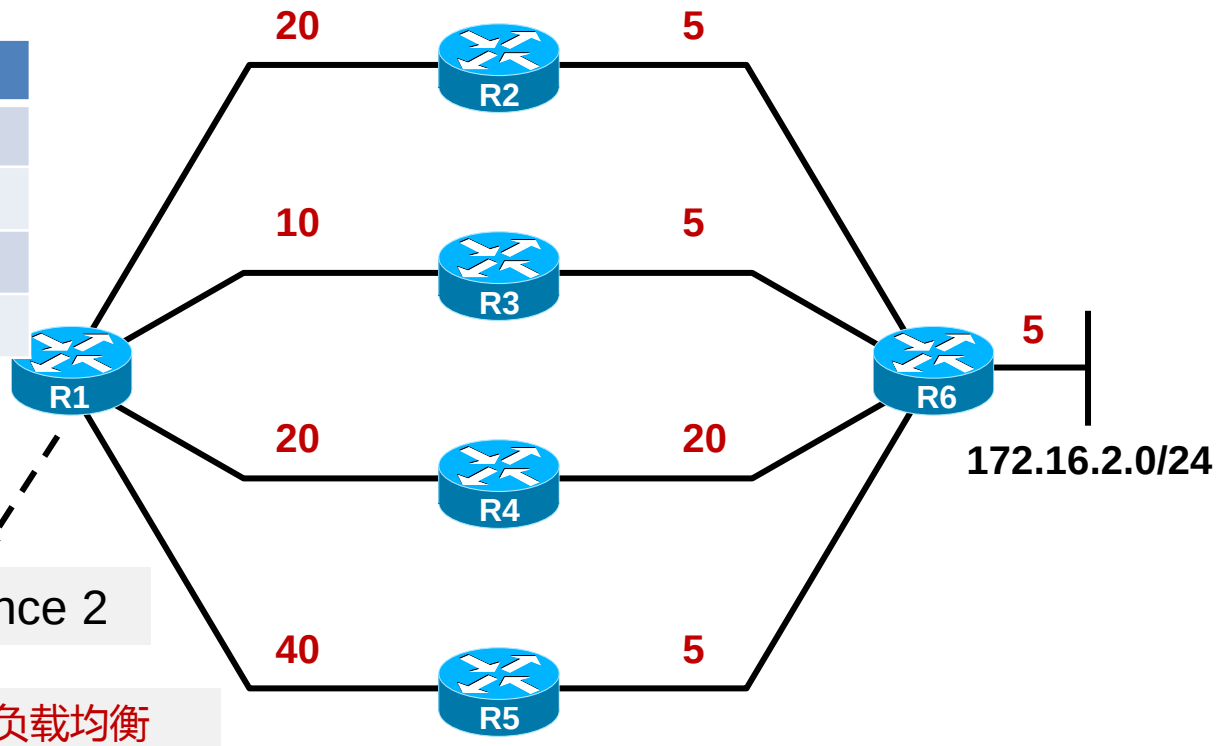
```
Router(config-router)# Variance multiplier
```

  - multiplier默认值为1，范围1~128
  - 只有可行路径才被用于负载均衡，可行条件为：
    - 路由必须是无环的。（即AD<FD min）
    - FD<= FD min x multiplier
  - 注：variance不指定最大路径，而指定了度量值得范围

EIGRP负载均衡

- **非等价负载均衡**

Network	Neighbor	FD	AD
2.0/24	R2	30	10
	R3	20	10
	R4	45	25
	R5	50	10



Router(config-router)# Variance 2

- 因此R1使用R2及R3进行不等价负载均衡
- 流量比例为：3/5：2/5

EIGRP认证

- **路由器使用两种身份验证方式**
  - 简单密码身份验证

    IS-IS

    OSPF

    RIPv2
  - MD5身份验证

    OSPF

    BGP

    EIGRP

    RIPv2

EIGRP认证

- **EIGRP MD5身份验证配置**

定义key chain (全局模式)

```
key chain name-of-chain
  key key-id
    key-string text
      accept-lifetime start-time {infinite | end-time | duration seconds}
      send-lifetime start-time {infinite | end-time | duration seconds}
```

EIGRP认证

- **EIGRP MD5身份验证配置(cont.)**

关联key chain（接口模式）

ip authentication key-chain eigrp *autonomous-system name-of-chain*

启用认证（接口模式）

ip authentication mode eigrp *autonomous-system* **md5**

EIGRP认证

- **EIGRP MD5身份验证实例**
  - R1配置

```
key chain R1chain
 key 1
  key-string firstkey
  accept-lifetime 04:00:00 Jan 1 2006 infinite
  send-lifetime 04:00:00 Jan 1 2006 04:01:00 Jan 1 2006
 key 2
  key-string secondkey
  accept-lifetime 04:00:00 Jan 1 2006 infinite
  send-lifetime 04:00:00 Jan 1 2006 infinite
!
interface Serial0/0/1
 ip authentication mode eigrp 100 md5
 ip authentication key-chain eigrp 100 R1chain
!
router eigrp 100
 network 192.168.12.0
 auto-summary
```

EIGRP认证

- **EIGRP MD5身份验证实例**
  - R2配置

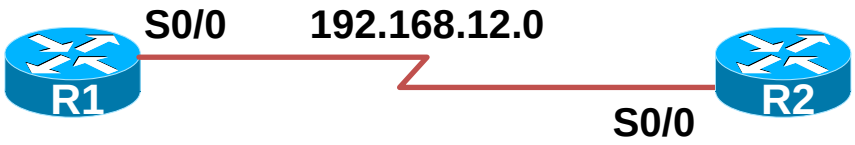
```
key chain R2chain
 key 1
  key-string firstkey
  accept-lifetime 04:00:00 Jan 1 2006 infinite
  send-lifetime 04:00:00 Jan 1 2006 infinite
 key 2
  key-string secondkey
  accept-lifetime 04:00:00 Jan 1 2006 infinite
  send-lifetime 04:00:00 Jan 1 2006 infinite
!
interface Serial0/0/1
 ip authentication mode eigrp 100 md5
 ip authentication key-chain eigrp 100 R2chain
!
router eigrp 100
 network 192.168.12.0
 auto-summary
```

EIGRP认证

```
R1#debug eigrp packets
EIGRP Packets debugging is on
    (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB, SIAQUERY, SIAREPLY)
*Jan 21 16:38:51.745: EIGRP: received packet with MD5 authentication, key id = 1
*Jan 21 16:38:51.745: EIGRP: Received HELLO on Serial0/0/1 nbr 192.168.1.102
*Jan 21 16:38:51.745:   AS 100, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 pe
erQ un/rely 0/0
R2#debug eigrp packets
EIGRP Packets debugging is on
    (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB, SIAQUERY,
SIAREPLY)
R2#
*Jan 21 16:38:38.321: EIGRP: received packet with MD5 authentication, key id = 2
*Jan 21 16:38:38.321: EIGRP: Received HELLO on Serial0/0/1 nbr 192.168.1.101
*Jan 21 16:38:38.321:   AS 100, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 pe
erQ un/rely 0/0
```

EIGRP认证

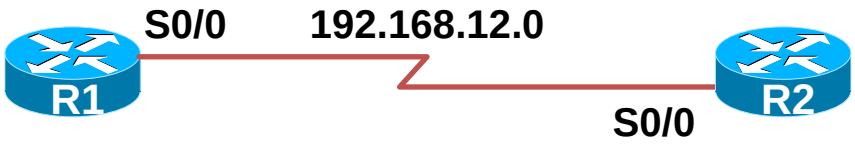
- **EIGRP MD5身份验证（测试）**



R1：key1 string spoto **R2**：key1 string spoto		成功
R1：key1 string spoto **R2**：key2 string spoto		Key ID不一致，即使密码一致也无法验证成功
R1：key1 string spoto **R2**：key1 string spoto	key2 string spoto1	成功
R1：key1 string spoto **R2**：key1 string spoto	key2 string spoto1 key2 string spoto	成功
R1：key1 string spoto **R2**：key1 string spoto1	key2 string spoto key2 string spoto	不成功，key1验证不通过，即使key2相同也不行
R1：Key2 string spoto **R2**：key1 string spoto1	key2 string spoto	无法正常建立邻居关系

EIGRP认证

- **EIGRP MD5身份验证（测试）**



因此eigrp默认只发送本地 第一个key，如果本地第一个key是key2（没有定义key1），那么就发送key2，如果对端有key1 key2，对端用第一个有效key即key1协商，而如果key1的密码不对，即使两端key2密钥一致，也无法建立邻居。

优化EIGRP实施

- **大型网络EIGRP的可扩展性**

  大型EIGRP网络通常会存在以下一些问题

  – 需要处理的路由表很大

  – 大量的邻居，要维护庞大的拓扑表

  – 需要交换大量的路由更新，发送大量的查询和应答

  这使得影响网络可扩展性的变量变多，如：

  – 邻居间交换的信息量

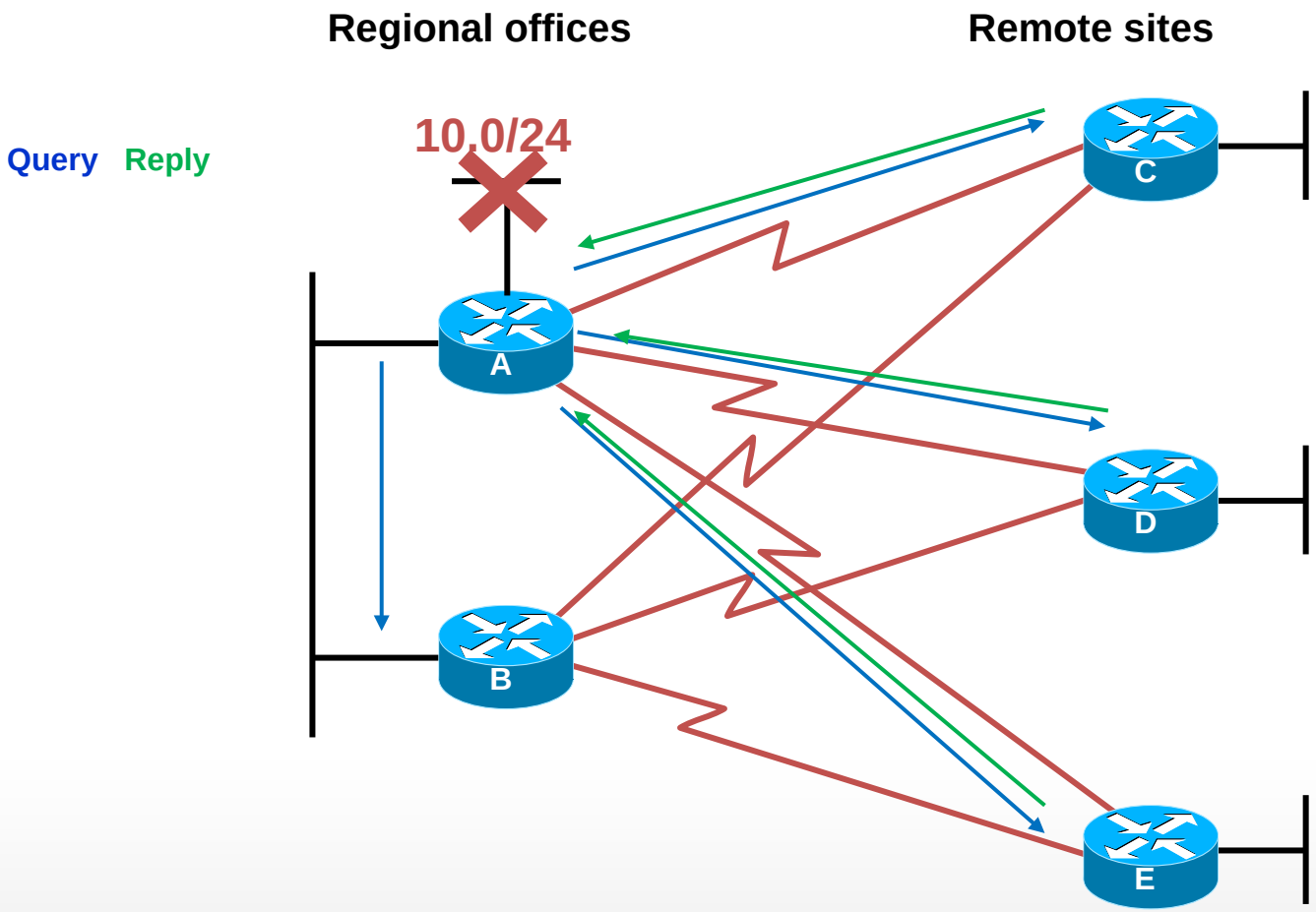
  – 路由器数量

  – 拓扑深度

  – 网络中的替代路径数

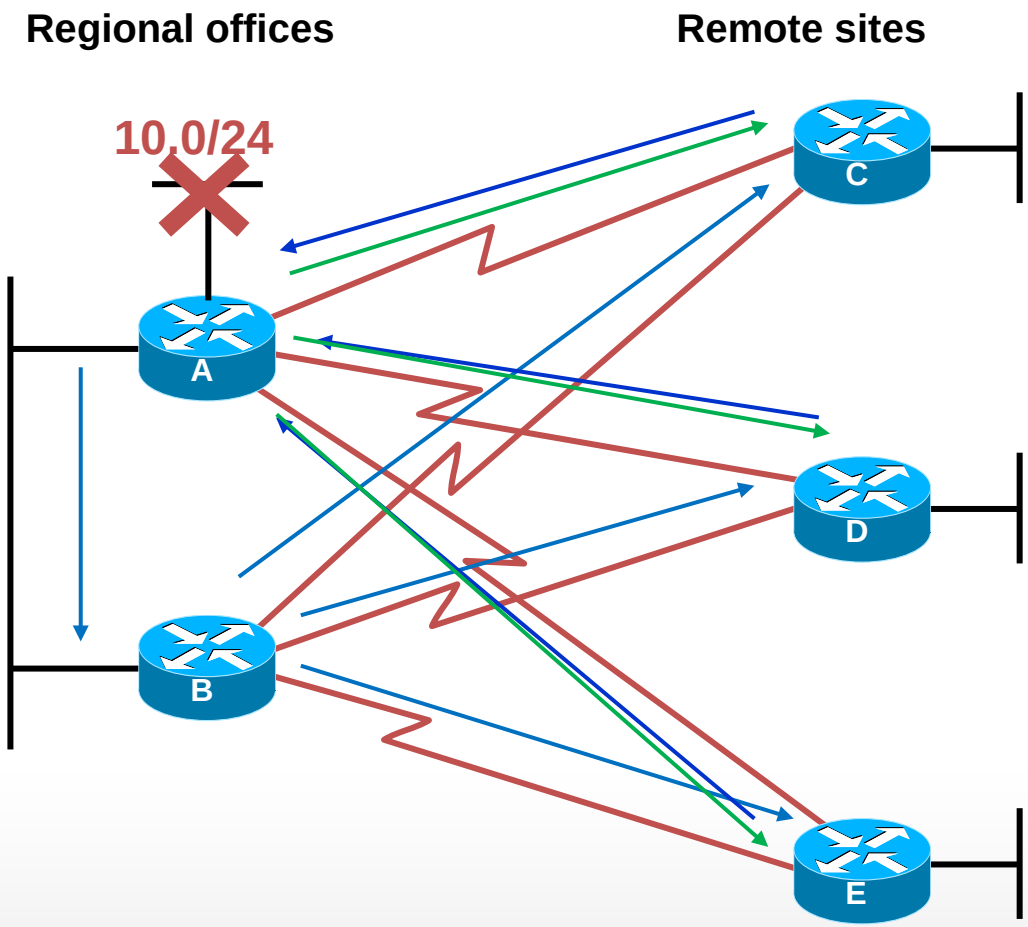
优化EIGRP实施

- EIGRP查询和主动状态



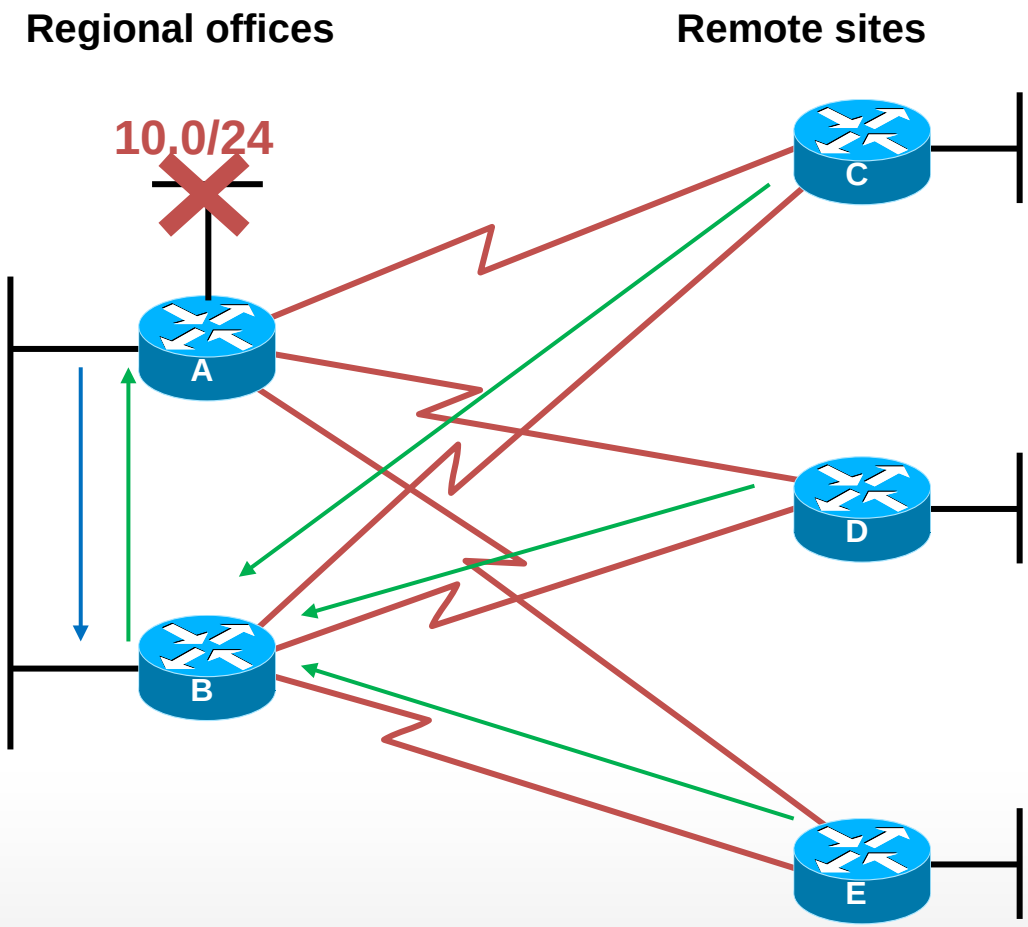
优化EIGRP实施

- **EIGRP查询和主动状态**



优化EIGRP实施

- EIGRP查询和主动状态



优化EIGRP实施

- **陷入主动状态**
  - 路由器陷入主动状态并因此发起查询，仅当收到每个查询的应答后，该路由器才会脱离主动状态进入被动状态
  - 如果路由器在3分钟内没有收到查询应答，路由将陷入主动状态（SIA），此时路由器将重置与未应答的邻居之间的邻接关系。

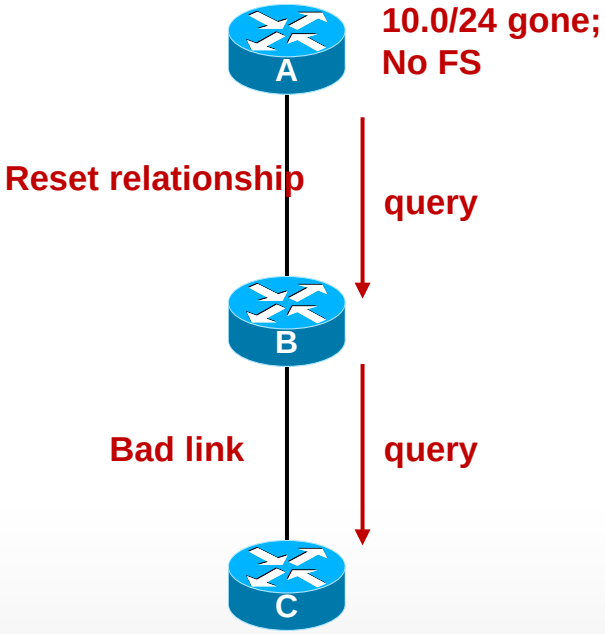
  导致路由进入SIA的常见原因

  - 路由器太忙无法回答查询
  - 路由器之间的链路质量低劣
  - 单向链路

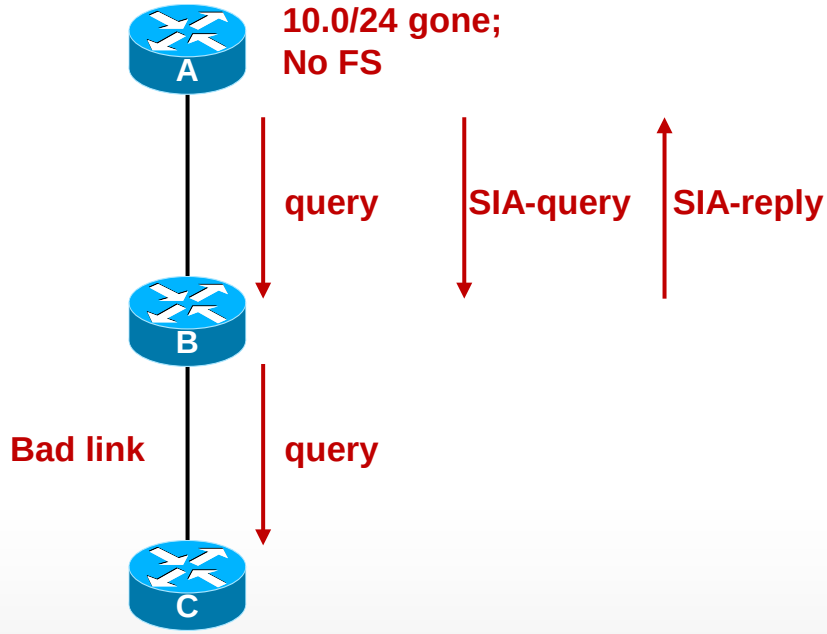
优化EIGRP实施

- **防范SIA**
  - EIGRP分组新增加了SIA-查询和SIA-应答，是由主动过程改进的

改进前：主动定时器到期后，A重置与B的邻接关系，但问题出在B和C之间的链路上



改进后：主动定时器过半后，A发送SIA-查询，而B确认查询，从而保持邻接关系

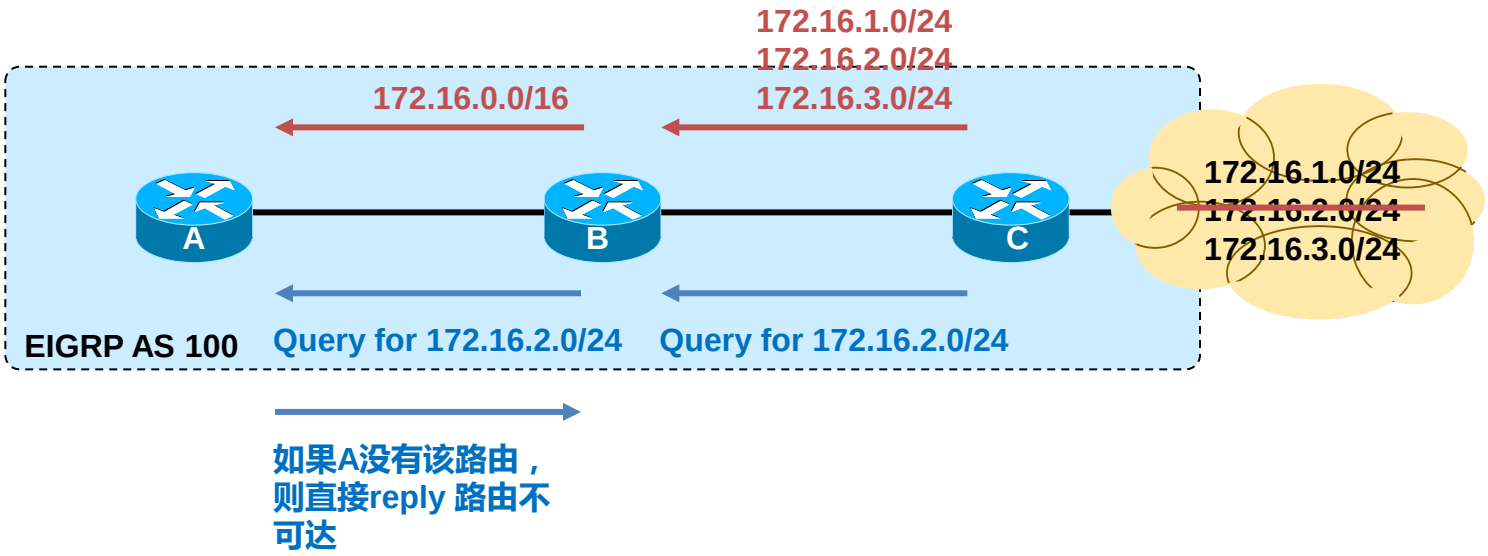


优化EIGRP实施

- **限制查询范围**
  - 确定好路由需求后，可提高EIGRP的可扩展性，使用一下两种方式
    - 在合适的路由器上使用路由汇总
    - 将远程路由器设置有末节EIGRP路由器

优化EIGRP实施

- **限制查询范围**
  - 使用路由汇总



- 查询在收到汇总路由的路由器结束
- 仅当路由表中有与被查询的网络完全匹配的路由时，远程路由器才会进一步传播查询

优化EIGRP实施

- **限制查询范围**
  - 将远程路由器设置有末节EIGRP路由器

    中央-分支网络拓扑中，stub路由器将所有非本地数据流转发给hub路由器，而无需保存完整的路由表。

    对于hub路由器来说，不应将stub路由器最为中转路由器，禁止stub路由器将hub路由器通告给其他的hub。

    stub路由器不会收到查询，与stub区域相连的hub路由器将代表末节路由器对查询做出应答。
  - 末节路由器时指：该路由器与网络核心相连，且不会被用来中转数据，末节路由器的EIGRP邻居全部都是中央路由器。

优化EIGRP实施

- **限制查询范围**
  - 将远程路由器设置有末节EIGRP路由器

配置

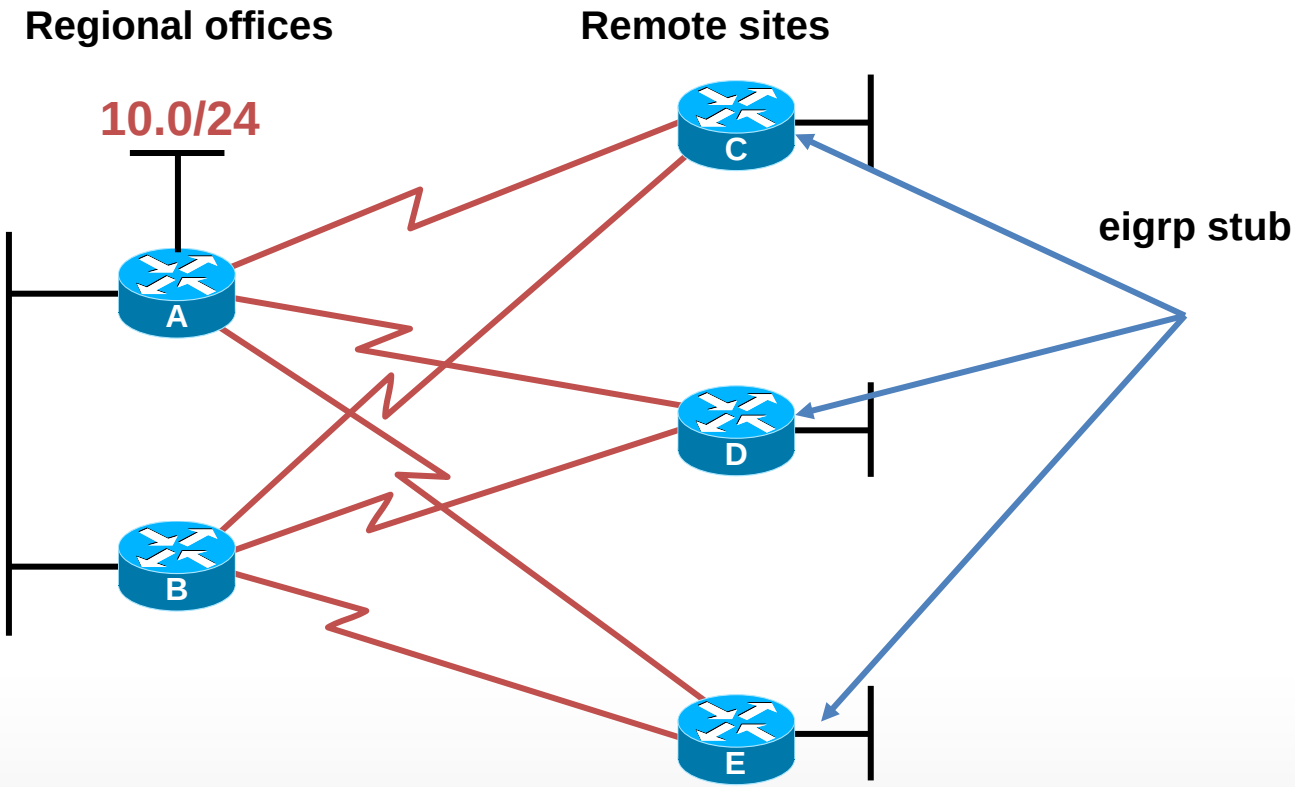
Router(config-router)#

Eigrp stub
[receive-only|connected|static|summary|redistributed]

- 默认通告直连路由和汇总路由
- Receive-only：禁止任何路由器共享其路由
- Connected：默认启用，允许发送直连路由
- Static：允许发送静态路由
- Summary：默认启用，允许发送汇总信息
- Redistribute：允许通告重分发而来的路由

优化EIGRP实施

- **限制查询范围**
  - 将远程路由器设置有末节EIGRP路由器



红茶三杯
Vinsoney

沉淀 提升 成长 分享
关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

多区域OSPF的概念及部署

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2012-08-01

课程目标

- OSPF基础知识回顾
- OSPF的报文类型
- OSPF邻居关系建立过程
- OSPF多区域的概念

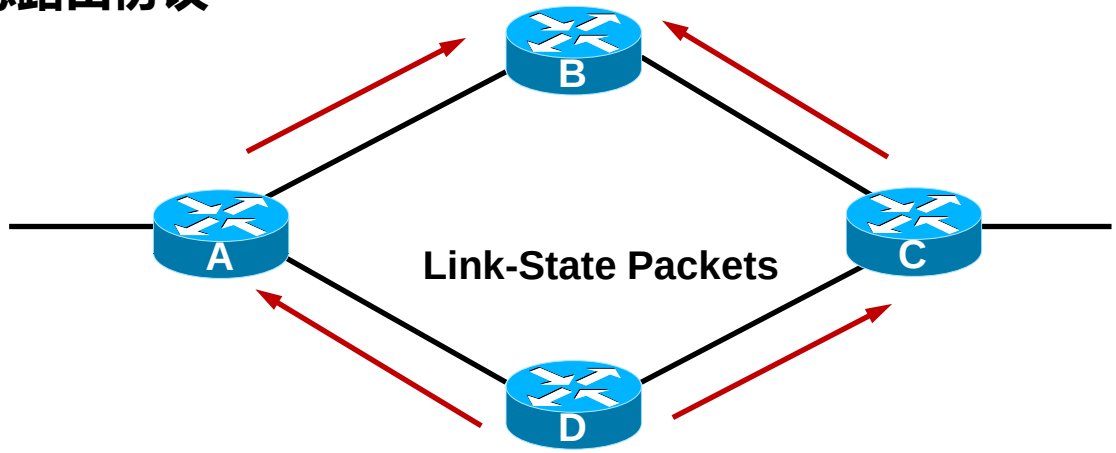
OSPF基础知识回顾

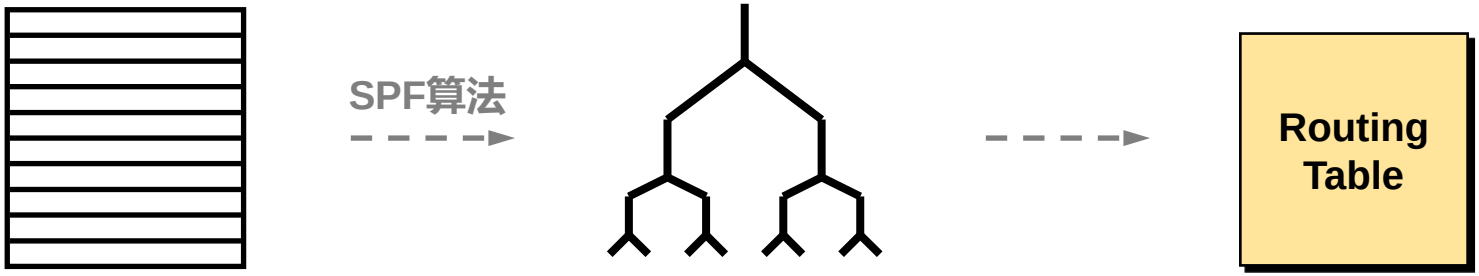
OSPF基础知识回顾

- **OSPF（Open Shortest Path First，开放最短路径优先）是一种链路状态路由协议，无路由循环（全局拓扑），属于IGP。RFC 2328，“开放”意味着非私有的，对公众开放的。**
- **OSPF的报文封装**
  - OSPF协议包直接封装于IP，协议号89。
- **OSPF协议使用的组播地址**
  - 所有OSPF路由器——224.0.0.5；DR BDR——224.0.0.6
- **OSPF路由协议的管理距离：110**

OSPF基础知识回顾

- 链路状态路由协议





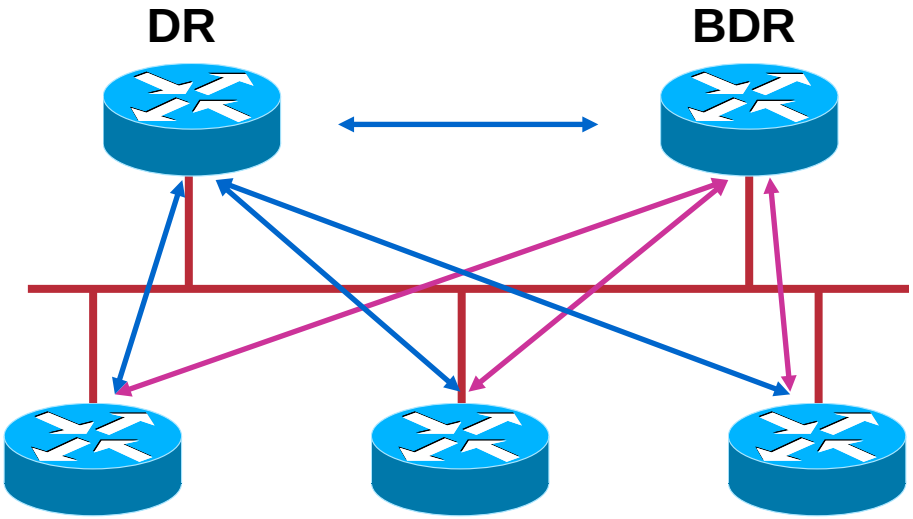
OSPF基础知识回顾

- **Router-ID**
  - 在一个OSPF域中，唯一地标识一台OSPF路由器
  - 32bits，表现为IPv4地址形式。在未有手工指定的情况下，如果本地有激活的Loopback接口，则取Loopback接口IP最大值；如果没有LP接口，则取激活的物理接口IP中的最大值
  - 为了提高路由器的RID的稳定性和网络的稳定性建议手动的设置路由器的Router-ID：在OSPF的进程下修改:router-id <x.x.x.x>
  - 项目实施中，一般是建立loopback口，并且手工指定loopback口地址为router-id

OSPF基础知识回顾

- **DR、BDR**
  - DR的作用：多路访问中为了减少邻接关系（N平方的问题）和LSA的洪泛，采用DR机制,BDR提供了备份
  - MA网络上的所有路由器均与DR、BDR建立邻居关系
- **DR选举比较顺序：**
  - 接口优先级数字越大越优先（优先级为0不能参与DR的选举）
  - Router ID越大越好
  - 稳定压倒一切（非抢占）
  - 通过控制接口优先级是控制DR选举的好办法
  - DR的选举是基于接口的，如果说某个路由器是DR，这种说法是错误的

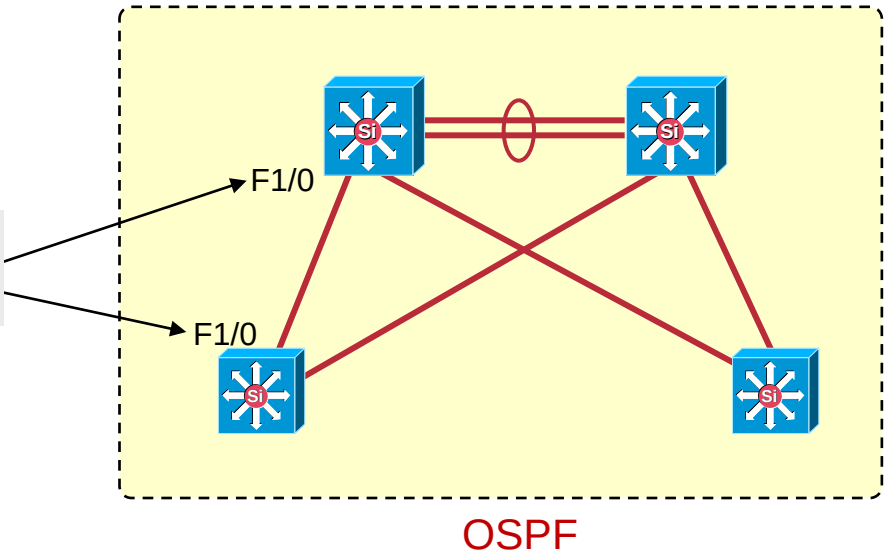
DR、BDR



DR、BDR

- 在实际的网络环境中，如若是三层以太网口运行OSPF且物理上为点对点连接，则OSPF仍然会在以太网上选举DR、BDR，因此一般将接口的OSPF网络类型进行更改，以跳过DR、BDR选举过程，加快OSPF邻居建立过程。

```
interface f1/0
  ip ospf network point-to-point
```



OSPF Cost

- **OSPF接口COST = 参考带宽（10的8次方）/ 接口带宽**

  接口带宽为接口逻辑带宽，可以使用bandwith命令调整，主要用于路由计算，而不是接口物理带宽，但一般情况：接口逻辑带宽 = 接口物理带宽。

- **手工修改接口Cost的方法**

  Router(config)# int serial 1

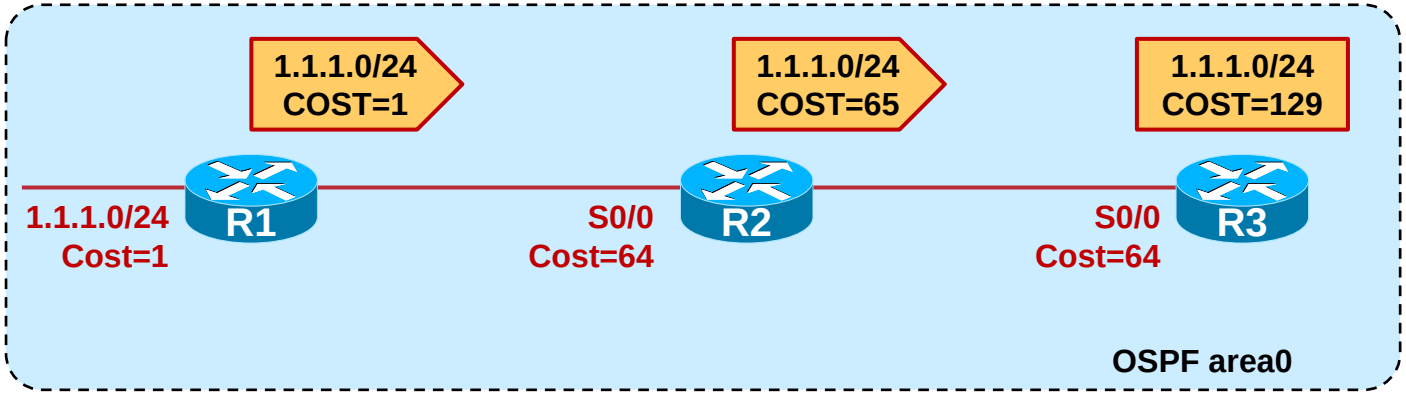
  Router(config-if)# ip ospf cost 100　　!! 该命令在接收路由的入口上配置

- **可修改“参考带宽”,来保障OSPF在现如今的网络中正常运转**

  auto-cost reference-bandwidth <参考带宽以Mbits为单位>

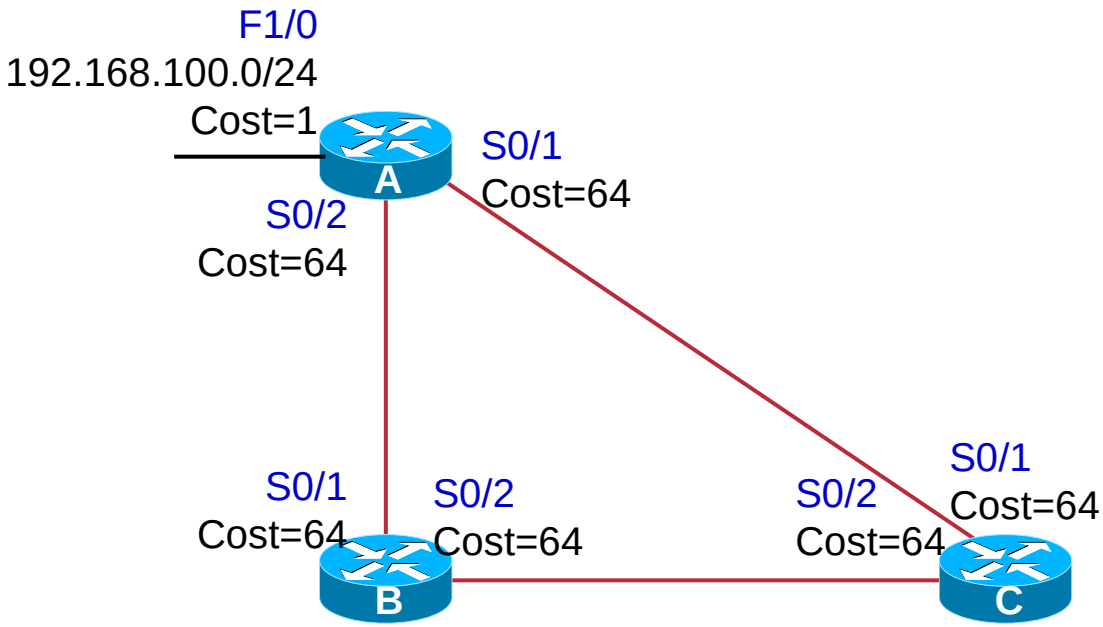
OSPF Cost

- **路由的COST**



OSPF Cost

- **Router C上路由表存放的192.168.100.0的路由，COST是多少？**



OSPF Cost

- **如何查看？**

  Router#sh ip os interface f 0/0

  FastEthernet0/0 is up, line protocol is up

  Internet Address 1.1.1.1/24, Area 0

  Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, **Cost: 1**

  Transmit Delay is 1 sec, State WAITING, Priority 1

- **如何更改？**

  Router(config-if)#ip ospf cost 100 // 接口模式下修改cost值

  Router(config-if)#bandwidth 100000 // 或者直接设置接口带宽，让OSPF自己计算

- **如何显示？**

```
Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
O       10.32.1.0 [110/74] via 192.168.1.1, 00:00:40, Serial1
C    192.168.1.0/24 is directly connected, Serial1
O    192.168.2.0/24 [110/128] via 192.168.1.1, 00:00:40, Serial1
                    [110/128] via 192.168.3.1, 00:00:40, Serial0
```

Show ip ospf interface x

show ip ospf interface ethernet1
Ethernet1 is up, line protoco1 1s up
Internet Address 192.168,32.4/24, Area 78
Process ID 1, Router ID 192,168.30,70, Network Type BROADCAST,Cost：10
Transmit Delay is 1 sec, state DROTHER, Priority 1
Designated Router (ID) 192.168.30.254, Interface address 192.168.32.2
Backup Oesignated router (ID) 192.168.30.80, Interface address 192.168.32.1
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
He11o due in 00:00:01
Neighbor count is 5, Adjacent neighbor count is 2
Adjacent with neighbor 192.168.30.80 (Backup Designated Rovter)
Adjacent with neighbor 192.168.3O.254 (Des1gnated Router)

OSPF的三张表

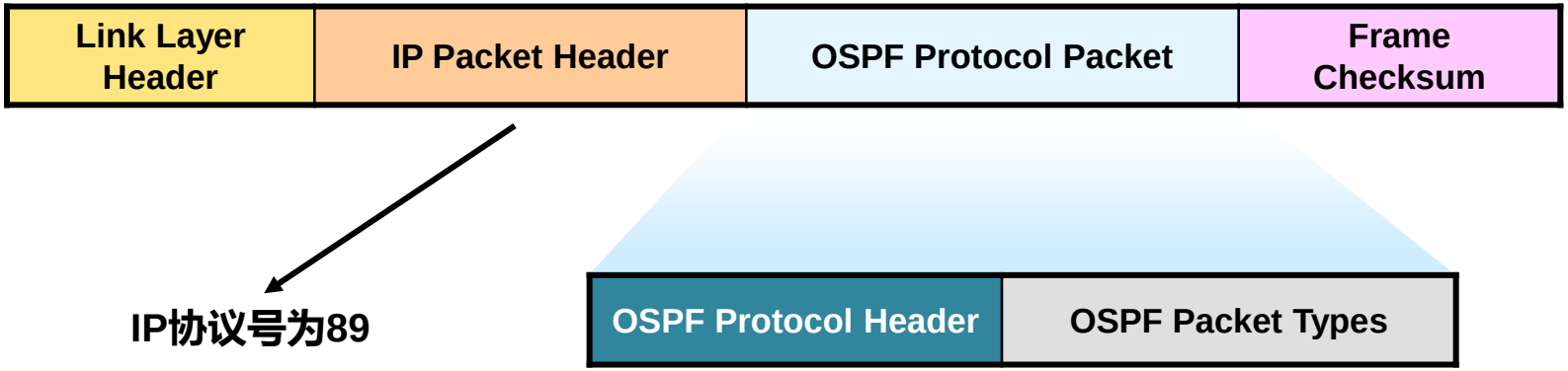
邻 居 表

链路状态数据库

路 由 表

- 相邻两台路由器运行OSPF协议
- 两台路由器直接连接
- 在同一自治系统
- Hello/Dead时间一致
- 区域ID一致
- 认证密码一致
- MTP值一致
- *网络类型一致
- *链路两端接口掩码一致

OSPF报文类型



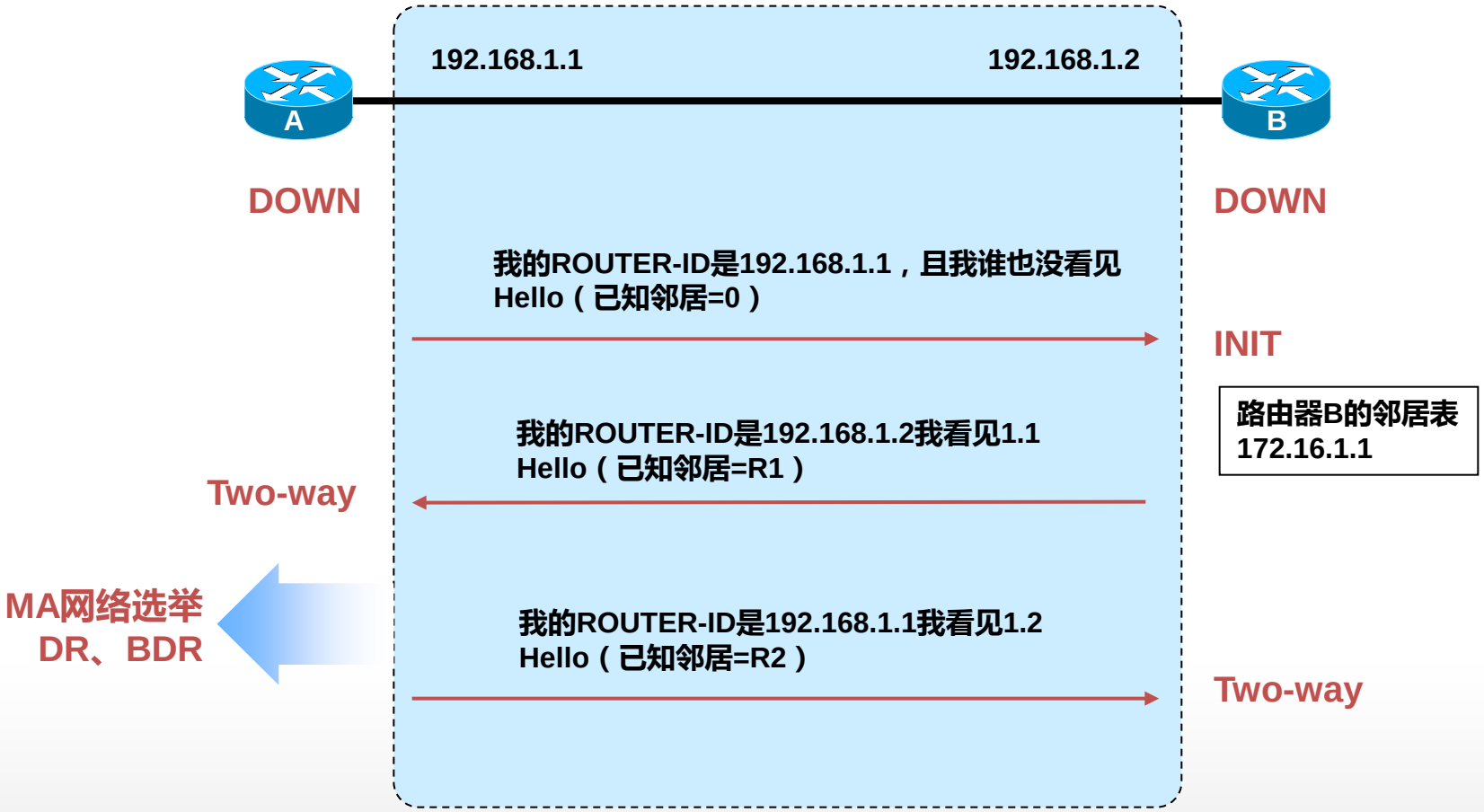
- **Hello** 建立和维护OSPF邻居关系。
- **DBD** 链路状态数据库描述信息（描述LSDB中LSA头部列表）
- **LSR** 链路状态请求,向OSPF邻居请求链路状态信息
- **LSU** 链路状态更新（包含一条或多条LSA）
- **LSAck** 确认报文

红茶三杯
Vinsoney

OSPF邻居关系建立过程

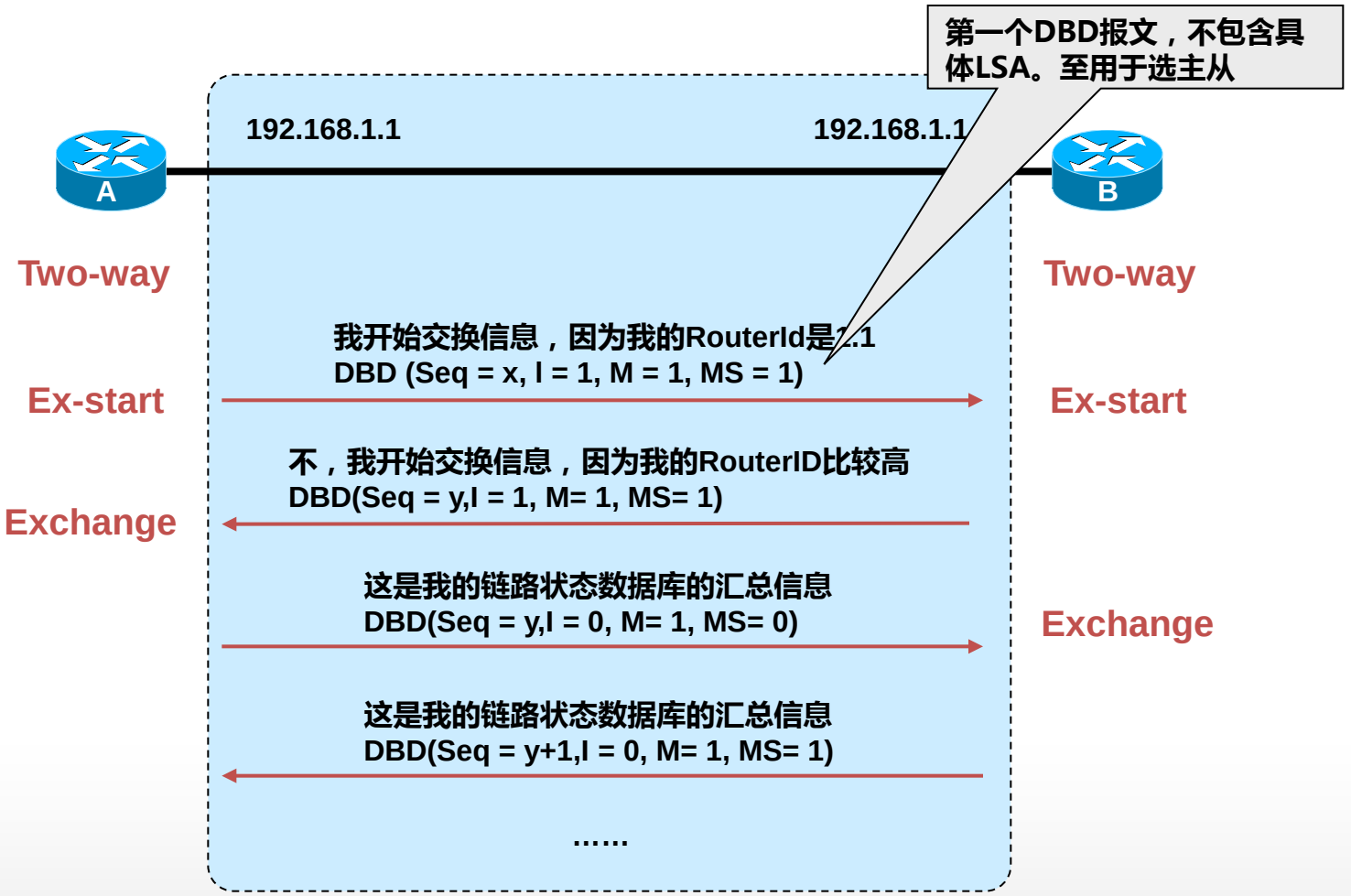
OSPF邻居建立过程

- **OSPF邻居建立过程 – 邻居发现**



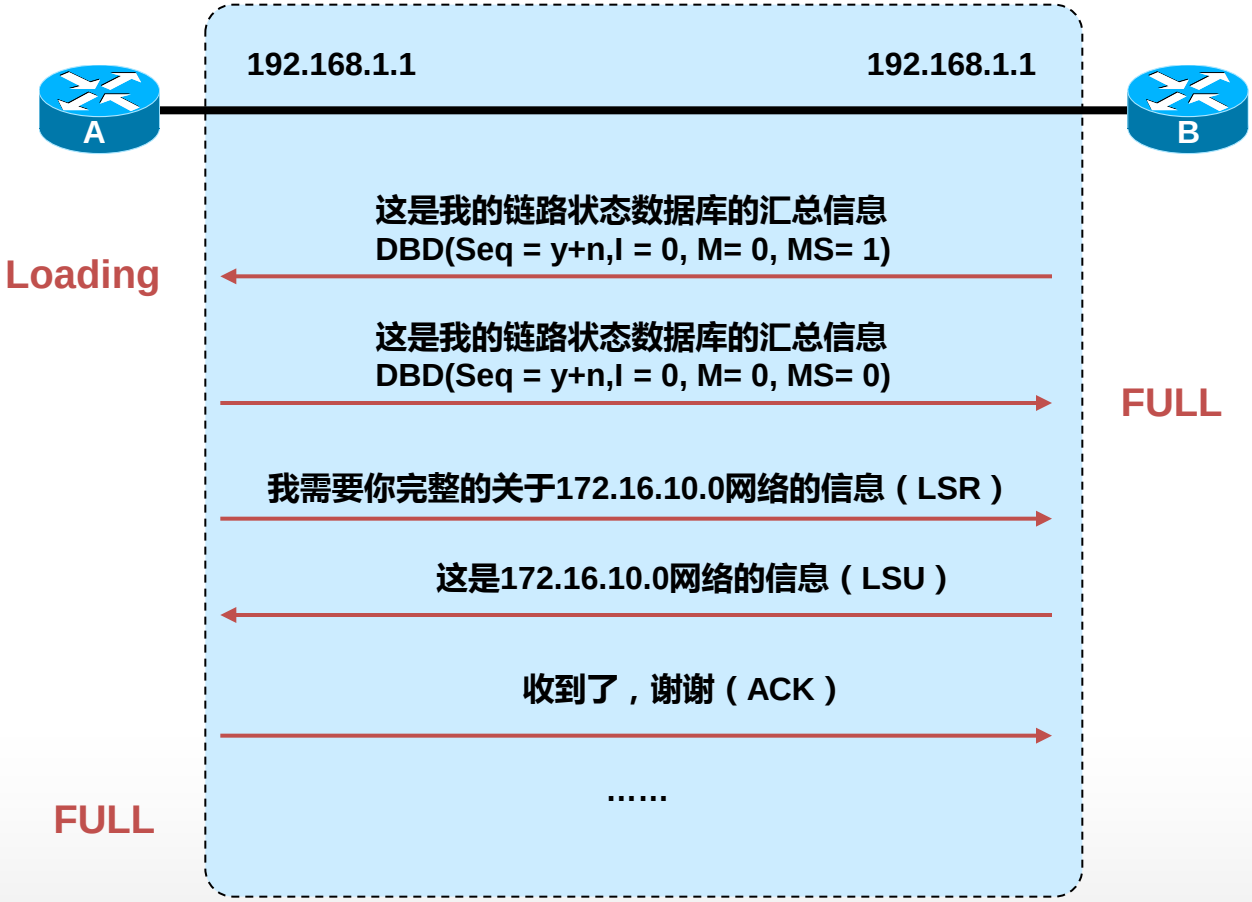
OSPF邻居建立过程 cont.

- **OSPF邻居建立过程 – 路由发现阶段**

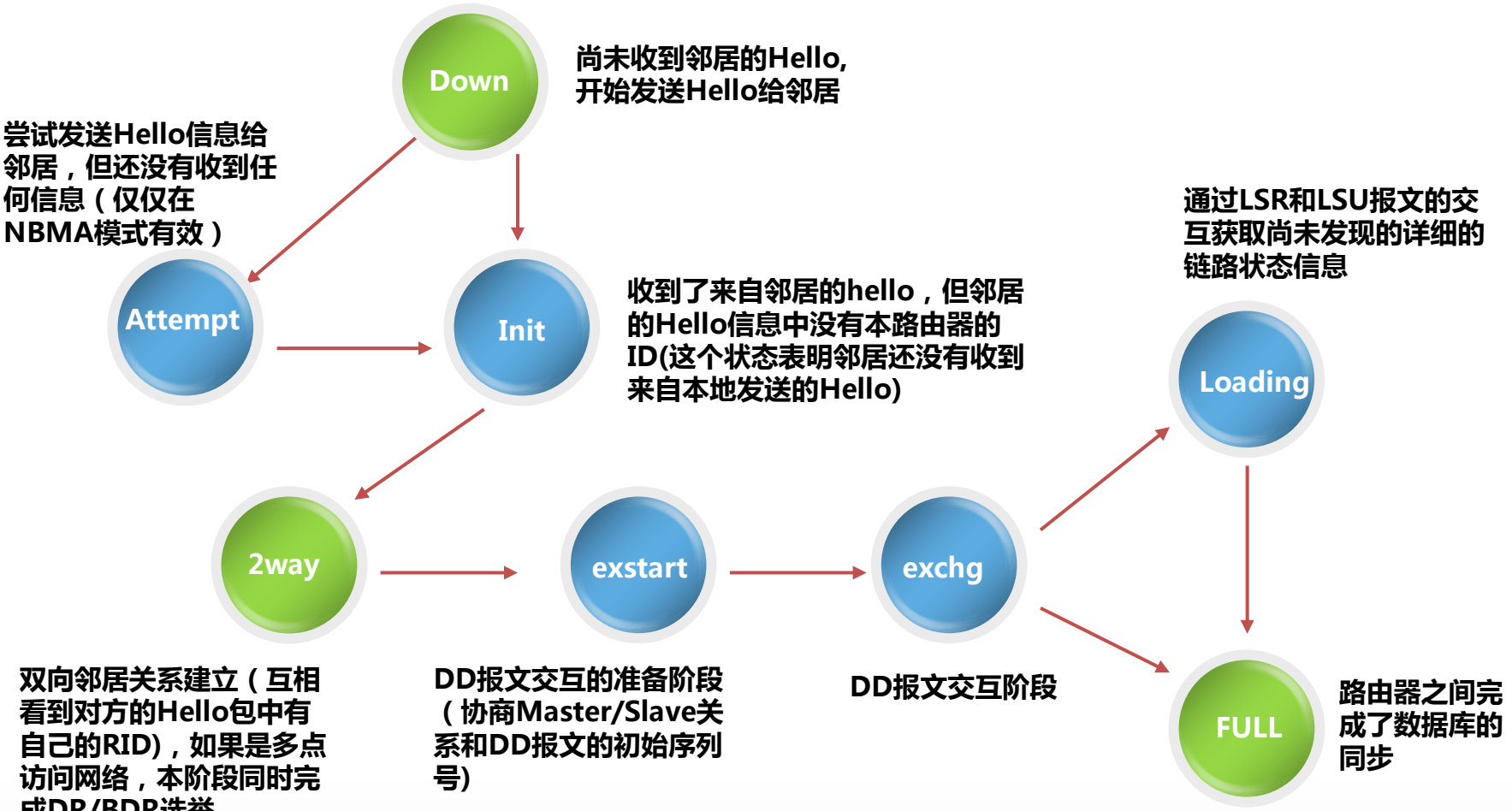


OSPF邻居建立过程 cont.

- **OSPF邻居建立过程 – 路由发现阶段**



OSPF邻居建立过程 小结



OSPF邻居建立过程

- **OSPF路由器建立邻接关系的过程详细描述**

1. **OSPF路由器接口up，发送Hello包，（NBMA模式时将进入Attempt状态）。**
2. **OSPF路由器接口收到Hello包，进入Init状态；并将该Hello包的发送者的Router ID，添加到Hello包（自己将要从该接口发送出去的Hello包）的邻居列表中。**
3. **OSPF路由器接口收到邻居列表中含有自己Router ID的Hello包，进入Two-way状态，形成OSPF邻居关系，并把该路由器的Router ID添加到自己的OSPF邻居表中。**
4. **在进入Two-way状态后，广播、非广播网络类型的链路，在DR选举等待时间内进行DR选举。点对点没有这个过程。**

OSPF邻居建立过程

- **OSPF路由器建立邻接关系的过程详细描述（Cont.）**

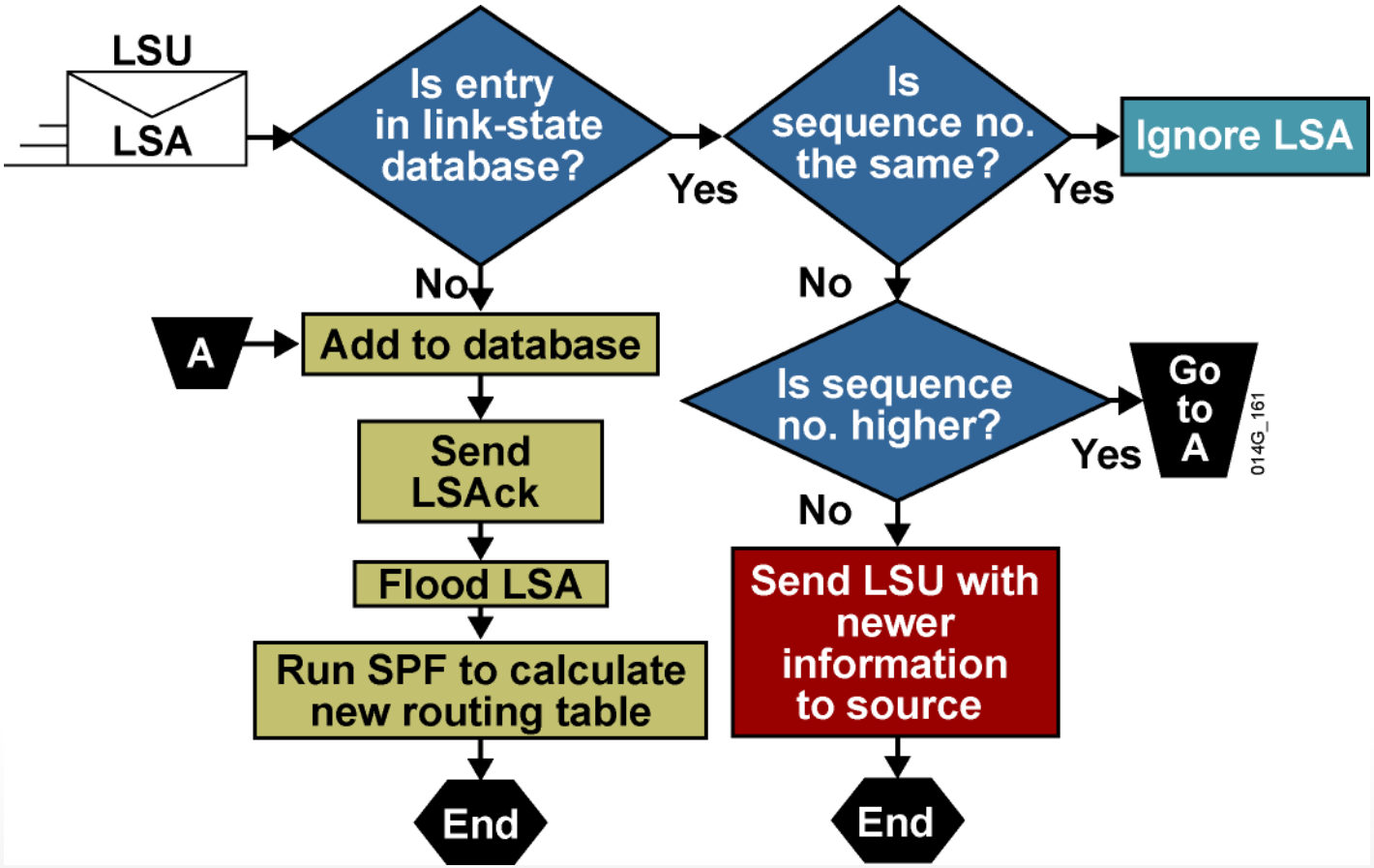
5. 在DR选举完成或跳过DR选举后，建立OSPF邻接关系，进入exstart（准启动）状态；并选举DBD交换主从路由器，以及由主路由器定义DBD序列号，Router ID大的为主路由器。目的是为了解决DBD自身的可靠性。
6. 主从路由器选举完成后，进入Exchange（交换）状态，交换DBD信息。
7. DBD交换完成后，进入Loading状态，对链路状态数据库和收到的DBD的LSA头部进行比较，发现自己数据库中没有的LSA就发送LSR，向邻居请求该LSA；邻居收到LSR后，回应LSU；收到邻居发来的LSU，存储这些LSA到自己的链路状态数据库，并发送LSAck确认。
8. LSA交换完成后，进入FULL状态，所有形成邻居的OSPF路由器都拥有相同链路状态数据库。
9. 定期发送Hello包，维护邻居关系。

OSPF邻居关系

- **OSPF邻接关系**
  - 路由器之间链路状态信息必须同步，LSA具有以下特征：
  - LSA（LSU）是可靠的传输，需要LSAck确认
  - LSA将被扩散到整个区域
  - LSA有序列号和寿命，以确保是最新的LSA
  - LSA被定期的刷新以确保拓扑信息的有效性

OSPF邻居关系

- 链路状态数据的结构



OSPF链路状态序列号

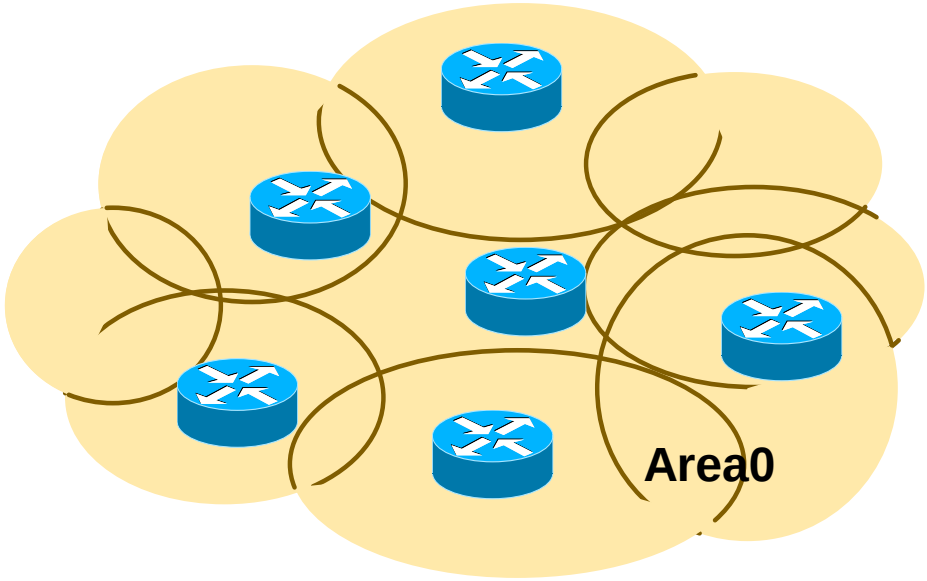
- 最大寿命定时器、刷新定时器和链路状态序列号确保数据库中只包含最新的链路状态记录
- 序列号大的为新的LSA（线性空间），第一个序列号为0x80000001，最后一个序列号为0x7FFFFFFF
- OSPF每隔30分钟对每条LSA记录扩散一次，此间隔称为LSA刷新时间，每当记录被扩散一次，序列号都加1

红茶三杯
Vinsoney

OSPF多区域的概念

OSPF多区域的概念

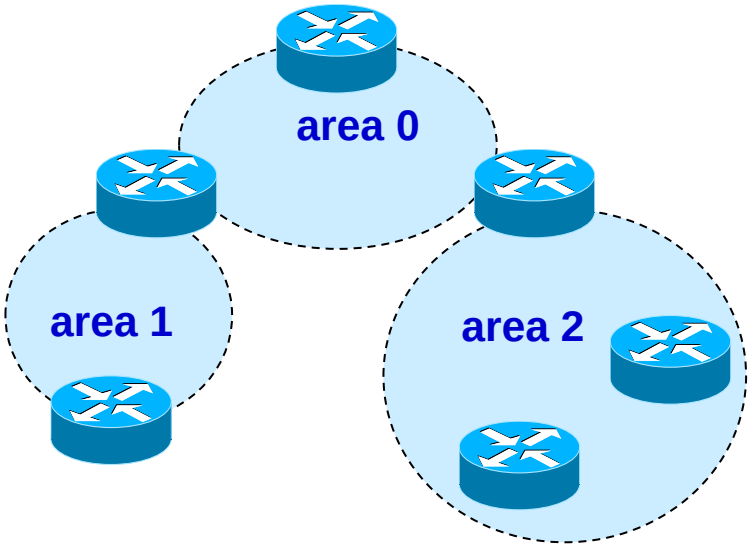
- **OSPF单区域**



- 收到的LSA通告太多了，OSPF路由器的负担很大
- 内部动荡会引起全网路由器的完全SPF计算
- 资源消耗过多，LSDB庞大，设备性能下降，影响数据转发
- 每台路由器都需要维护的路由表越来越大，单区域内路由无法汇总

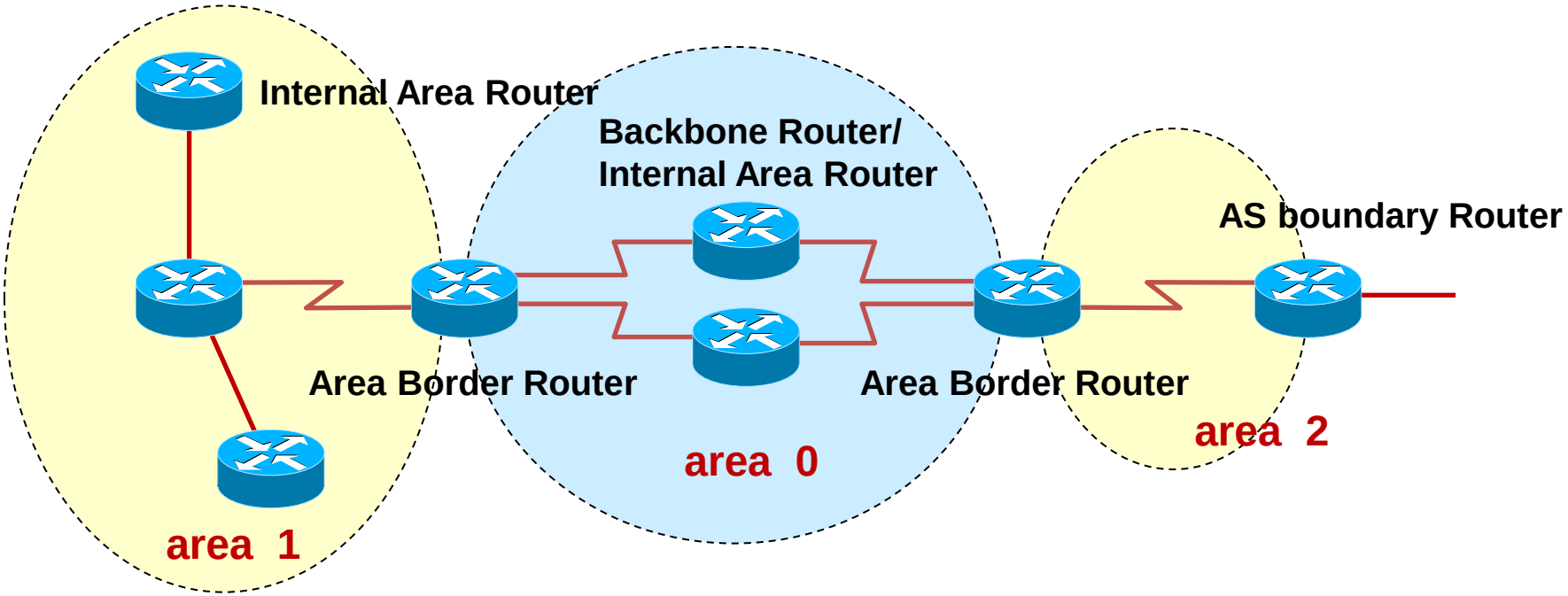
OSPF多区域的概念

- **解决方案:**
  - 把大型网络分隔为多个较小，可管理的单元 – 区域 area;



- 减少了LSA洪泛的范围，有效地把拓扑变化控制在区域内，达到网络优化的目的
- 在区域边界可以做路由汇总，减小了路由表
- 充分利用OSPF特殊区域的特性，进一步减少LSA泛洪，从而优化路由
- 多区域提高了网络的扩展性，有利于组建大规模的网络

OSPF路由器角色



OSPF多区域配置

- **OSPF进程及网络宣告**

Router(config)#

router ospf *process-id* **[vrf** *vpn-name***]**

Router(config-router)#

network *ip-address wildcard-mask* **area** *area-id*

OSPF多区域配置

- **Router-ID**

Router(config-router)#

router-id *ip-address*

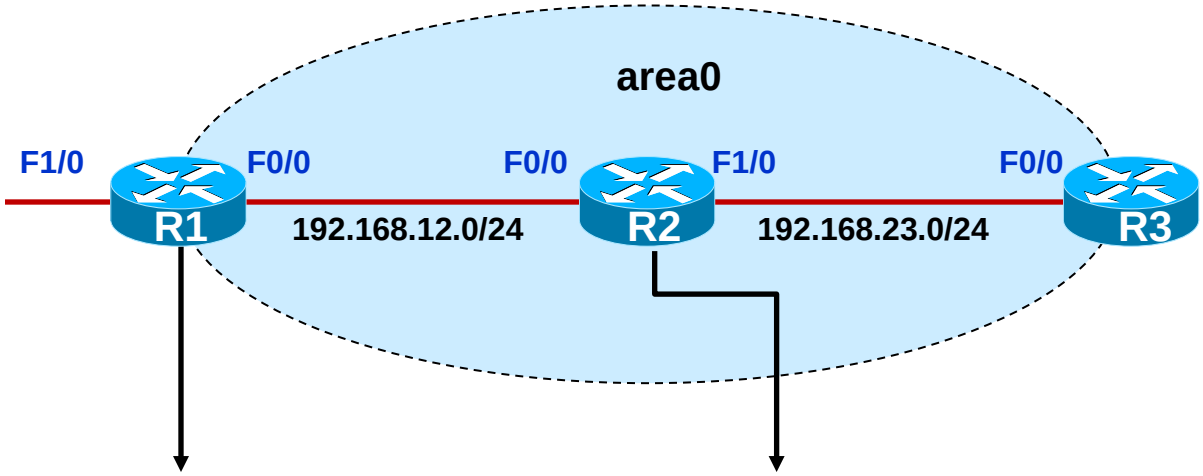
Router#

clear ip ospf process

TIPS:　　建议在工程环境中，对OSPF进程的Router-ID进行统一规划，并为每台路由器分配router-ID，开设loopback接口，同时在OSPF进程中手动指定。

OSPF多区域配置

- **例 单区域配置**

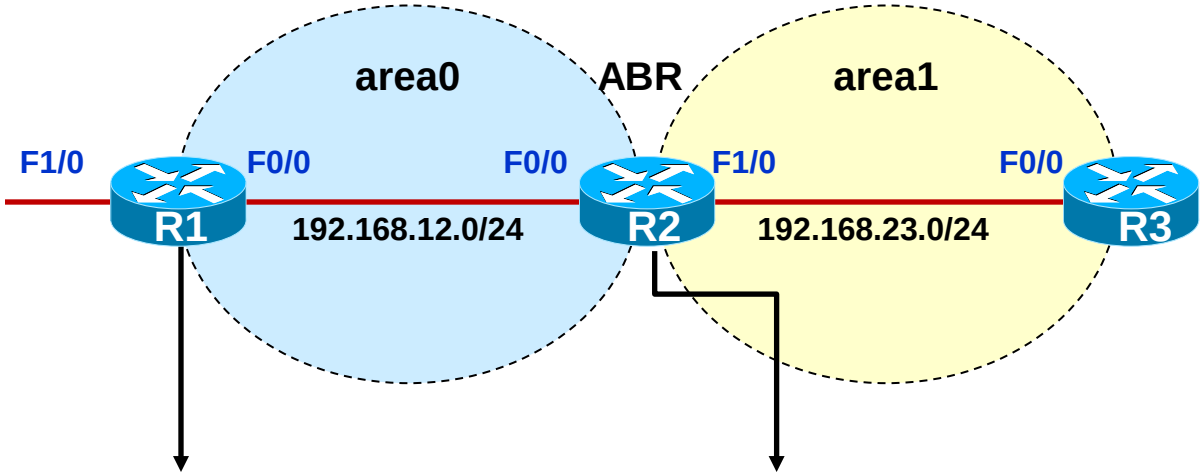


router ospf 1
router-id 192.168.255.1
network 192.168.12.0 0.0.0.255 area 0

router ospf 1
router-id 192.168.255.2
network 192.168.12.0 0.0.0.255 area 0
network 192.168.23.2 0.0.0.0 area 0

OSPF多区域配置

- **例 多区域配置**



router ospf 1
router-id 192.168.255.1
network 192.168.12.0 0.0.0.255 area 0

router ospf 1
router-id 192.168.255.2
network 192.168.12.0 0.0.0.255 area 0
network 192.168.23.2 0.0.0.0 area 1

OSPF配置验证

- **show ip ospf 显示OSPF路由器ID，OSPF定时器以及LSA信息**
- **show ip ospf interface *type* 显示各种定时器和邻接关系**
- **show ip route ospf 显示路由器学习到的OSPF路由**
- **show ip protocols 显示IP路由协议参数**
- **debug ip ospf events 显示OSPF相关事件**
- **debug ip ospf adj 跟踪邻接关系的建立和终止**
- **debug ip ospf packet 查看正在传输的OSPF分组**

OSPF多区域配置

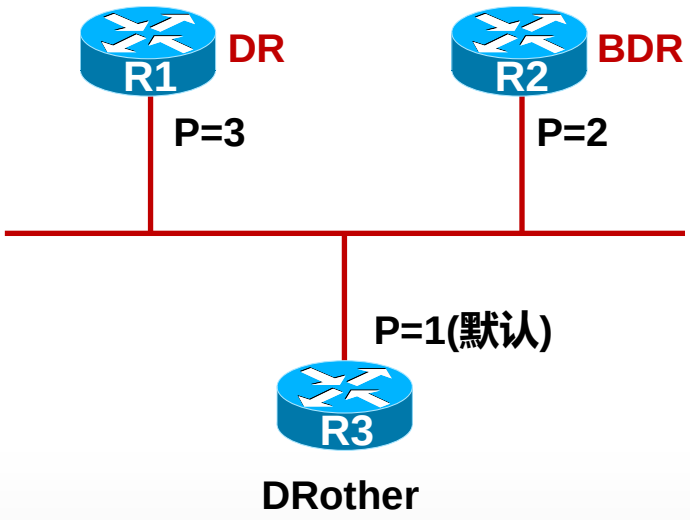
- **DR及BDR选举的控制**

Router(config-if)#

ip ospf priority *10*

比较次序

- 优先级
- 路由器ID
- 优先级为0的不能成为DR或BDR



OSPF网络类型

- **OSPF网络类型包括以下几种**
  - 点到点
  - 广播
  - 非广播
    - 非广播又包括了5种运行模式：

      NBMA (RFC)
      P2MP (RFC)
      P2MP nonbroadcast(CISCO)
      Broadcast(CISCO)
      P2P(CISCO)

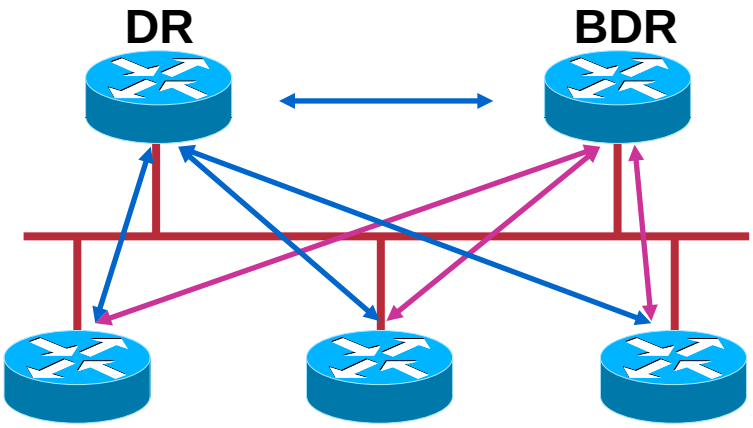
OSPF网络类型

- **点到点类型**

- 如果二层的协议为PPP、HDLC等，则OSPF网络类型为P2P
- 如果帧中继子接口类型为P2P的，则OSPF网络类型也为P2P
- 不选举DR、BDR
- 使用组播地址224.0.0.5
- OSPF能够根据二层封装自动检测到P2P网络类型

OSPF网络类型

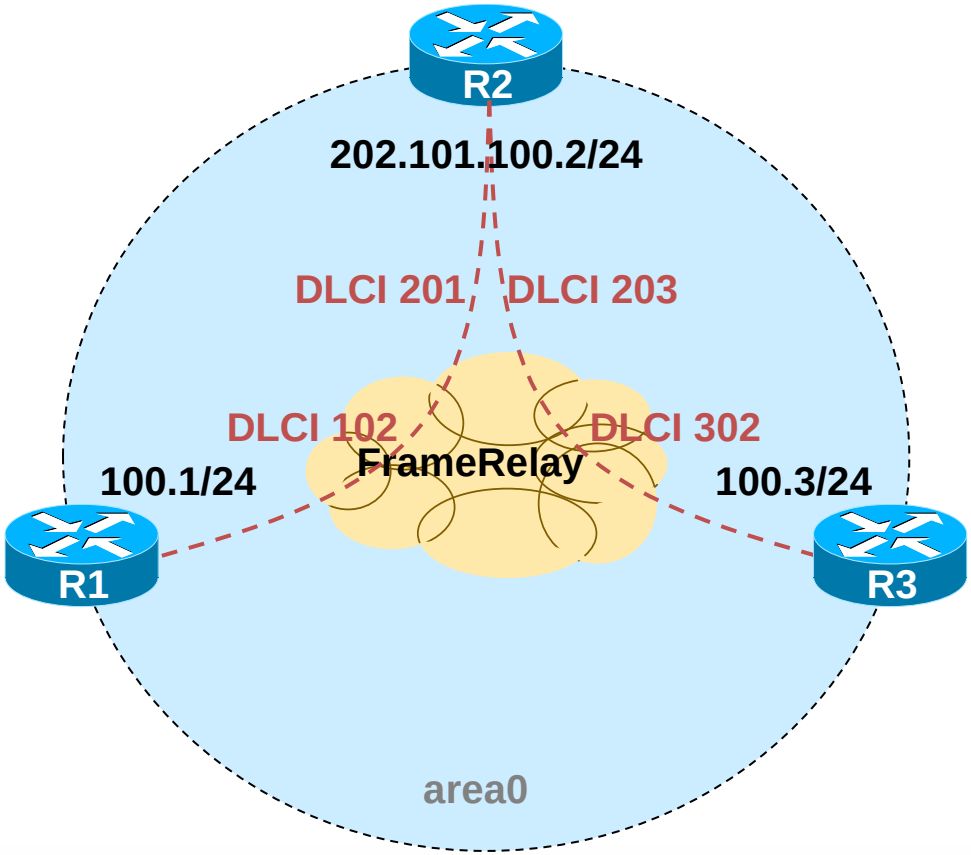
- **广播型多路访问**



- 通常出现在以太网
- 选举DR、BDR
- 所有路由器均与DR及BDR建立邻接关系
- 使用组播地址224.0.0.5及224.0.0.6

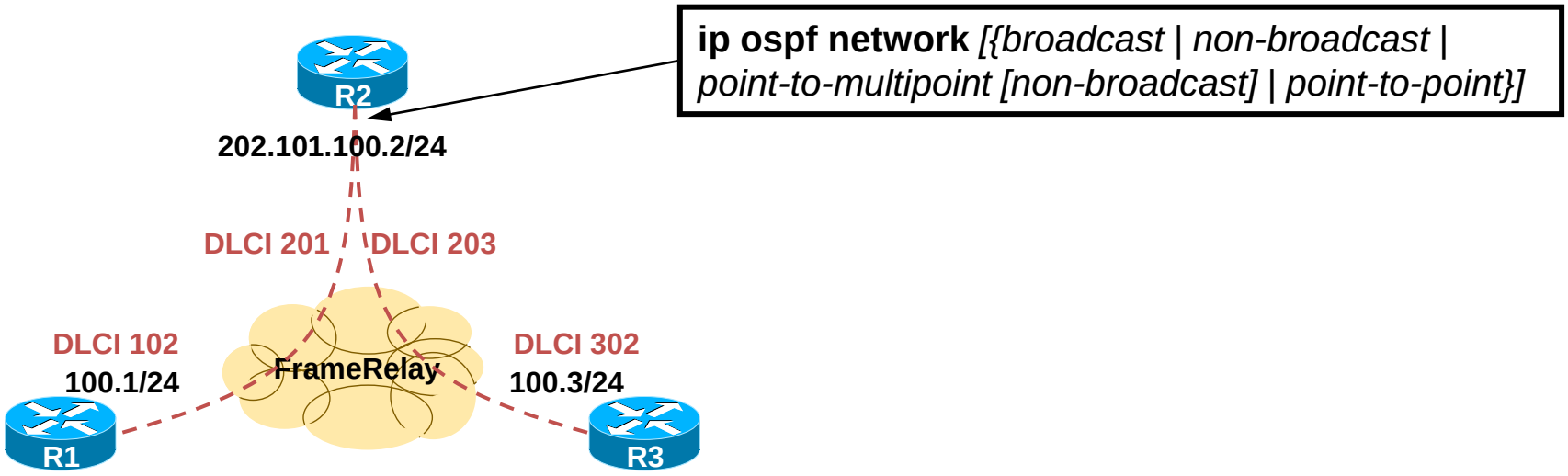
OSPF网络类型

- NBMA网络选择OSPF的模式



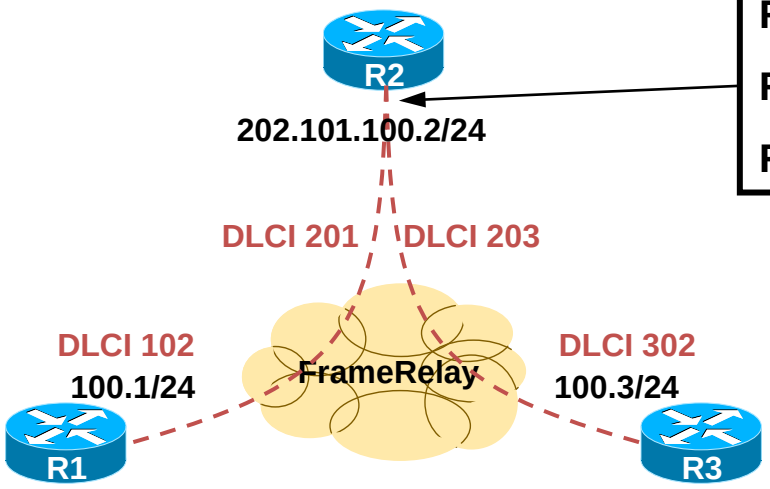
OSPF网络类型

- **NBMA网络选择OSPF的模式**
  - 在帧中继主接口上，默认的OSPF模式为非广播
  - 在点到点帧中继子接口上，默认的OSPF模式为点到点
  - 在帧中继多点子接口上，默认的OSPF模式为非广播



OSPF网络类型

- **NBMA网络下OSPF的运行**
  - 方式1

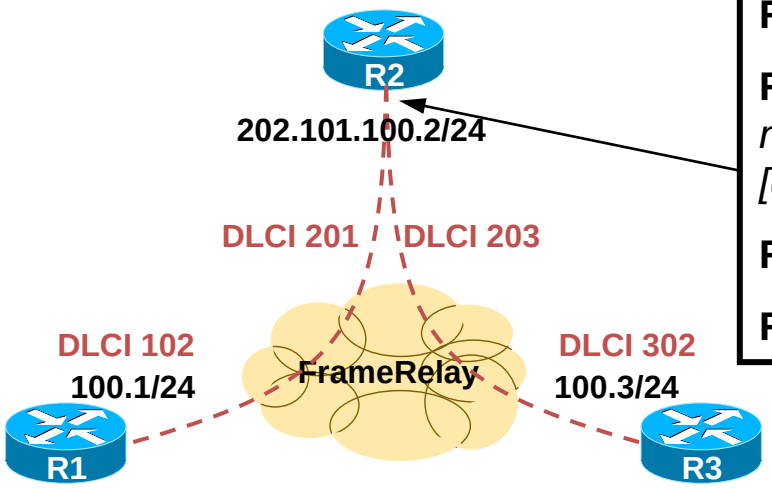



```
Router(config)#interface serial 0/0
Router(config-if)#encapsulation frame-relay
Router(config-if)#ip ospf network broadcast
```

- 指定OSPF网络类型为broadcast
- 使用OSPF多播Hello分组来自动发送邻居
- 选举DR和BDR
- DR和BDR必须与其他所有路由器直接相连

OSPF网络类型

- **NBMA网络下OSPF的运行**
  - 方式2



Router(config)#interface serial 0/0

Router(config)#neighbor *ip-address [priority number] [poll-interval number] [cost number] [database-filter all]*

Router(config-if)#encapsulation frame-relay

Router(config-if)#ip ospf network non-broadcast

- 网络类型为non-broadcast（默认）
- 手动指定邻居
- 选举DR和BDR
- DR和BDR必须与其他所有路由器直接相连

OSPF网络类型

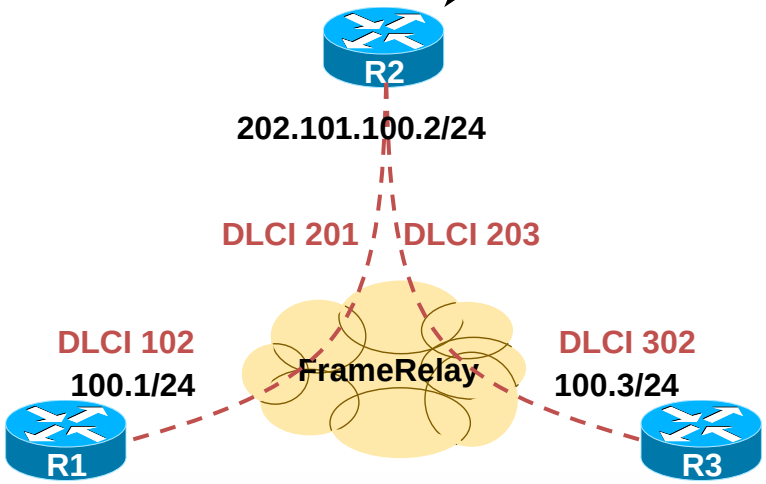
- **NBMA网络下OSPF的运行**
  - 方式2 示例

R2(config)# router ospf 100

R2(config-router)# network 202.101.100.0 0.0.0.255 area 0

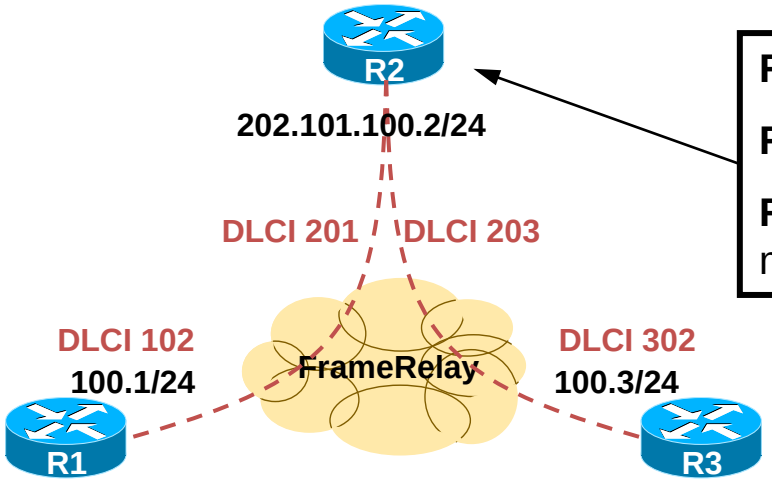
R2(config-router))# neighbor 202.101.100.1 priority 0

R2(config-router))# neighbor 202.101.100.3 priority 0



OSPF网络类型

- **NBMA网络下OSPF的运行**
  - 方式3



Router(config)#interface serial 0/0

Router(config-if)#encapsulation frame-relay

Router(config-if)#ip ospf network point-to-multipoint

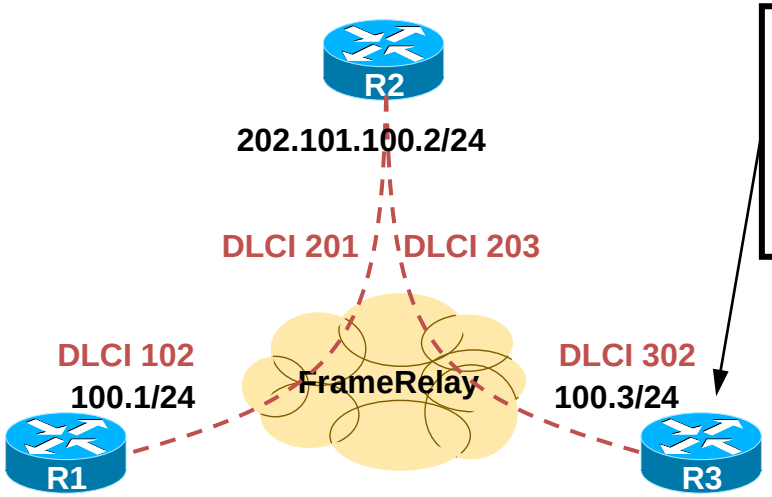
- 指定网络类型为point-to-multipoint
- 使用OSPF多播Hello分组来自动发现邻居
- 不选举DR和BDR

OSPF网络类型

- **NBMA网络下OSPF的运行**
  - 方式3 示例

```
interface Serial0/0
 encapsulation frame-relay
 ip ospf network point-to-multipoint

router ospf 100
network 202.101.100.0 0.0.0.255 area 0
```

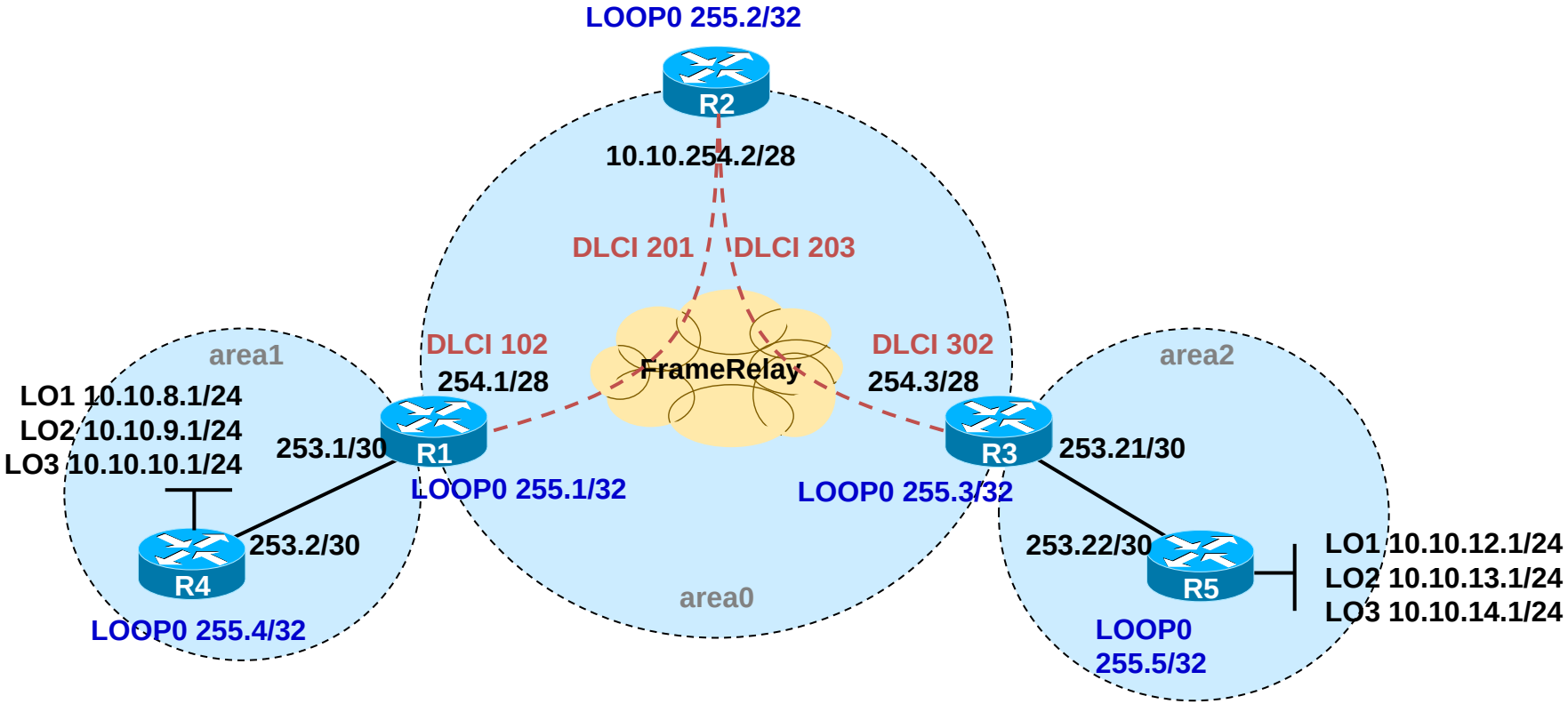
```
interface Serial0/0
 encapsulation frame-relay
 ip ospf network point-to-multipoint
 ip ospf priority 0
```



OSPF网络类型

- **NBMA网络选择OSPF的模式**

OSPF模式	NBMA拓扑类型	接口子网地址	Hello时间 dead时间	手工指定邻居	是否选举DR/BDR
广播 (CISCO模式)	部分互联 全互联	相同	10/40S	NO	YES
非广播 (RFC模式)	部分互联 全互联	相同	30/120S	YES	YES
点到多点 (RFC模式)	部分互联 星型	相同	30/120S	NO	NO
点到多点非广播 (CISCO模式)	部分互联 星型	相同	30/120S	YES	NO
点到点 (RFC模式)	部分互联 星型	相同	10/40S	NO	NO



10.10.0.0/16 Subnets

红茶三杯
Vinsoney

学习 沉淀 成长 分享

关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

OSPF LSA及特殊区域详解

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2012-08-01

课程目标

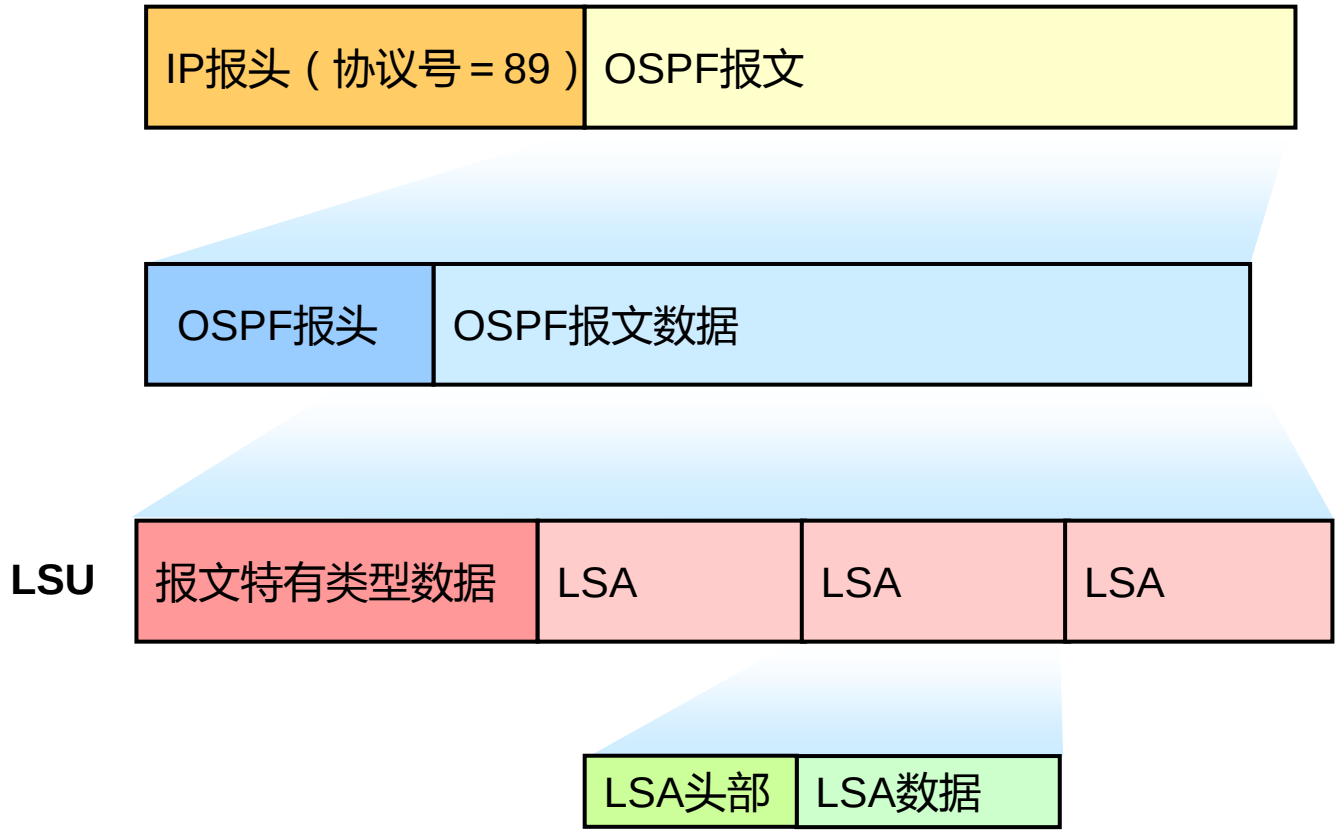
- LSA类型及详解
- OSPF特殊区域详解
- 配置OSPF特殊区域

红茶三杯
Vinsoney

OSPF LSA类型及详解

- LSA概述
- 各种类型LSA介绍
- LSA报文结构及字段含义

什么是LSA


IP报头（协议号=89）
OSPF报文
OSPF报头
OSPF报文数据
LSU
报文特有类型数据
LSA
LSA
LSA
LSA头部
LSA数据

LSA类型

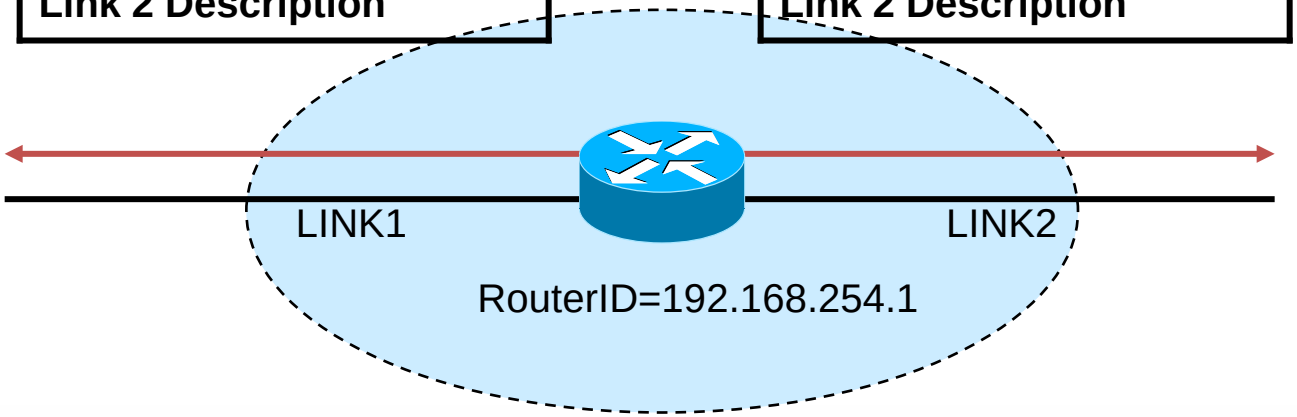
类别	名称	描述
1	路由器LSA	区域内Router产生，描述了路由器所有接口、链路和Cost值。只能在本区域内泛洪。
2	网络LSA	由DR产生，报文包括了其连接的所有Router的routerID，其中包含自己的routerID。
3	网络汇总LSA	可以通知本区域内的路由器通往区域外的路由信息。默认路由也被通告。 Link ID为目标网段的ID
4	ASBR汇总LSA（ASBR summary LSA）	也是由ABR产生，但是它是一条主机LSA，指向ASBR路由器
5	自治系统外部汇总LSA	由ASBR产生，告诉本自治区的路由器通往外部自治区的路径。
7	NSSA外部LSA	由ASBR产生，几乎和LSA5通告是相同的，但NSSA外部LSA通告仅仅在始发这个NSSA外部LSA通告的非纯末梢区域内部进行泛洪。

LSA类型

- **类型1：路由器LSA Router LSA**

TYPE=1
RouterID=192.168.254.1
Numbers of Links =2
Link 1 Description
Link 2 Description

TYPE=1
RouterID=192.168.254.1
Numbers of Links =2
Link 1 Description
Link 2 Description

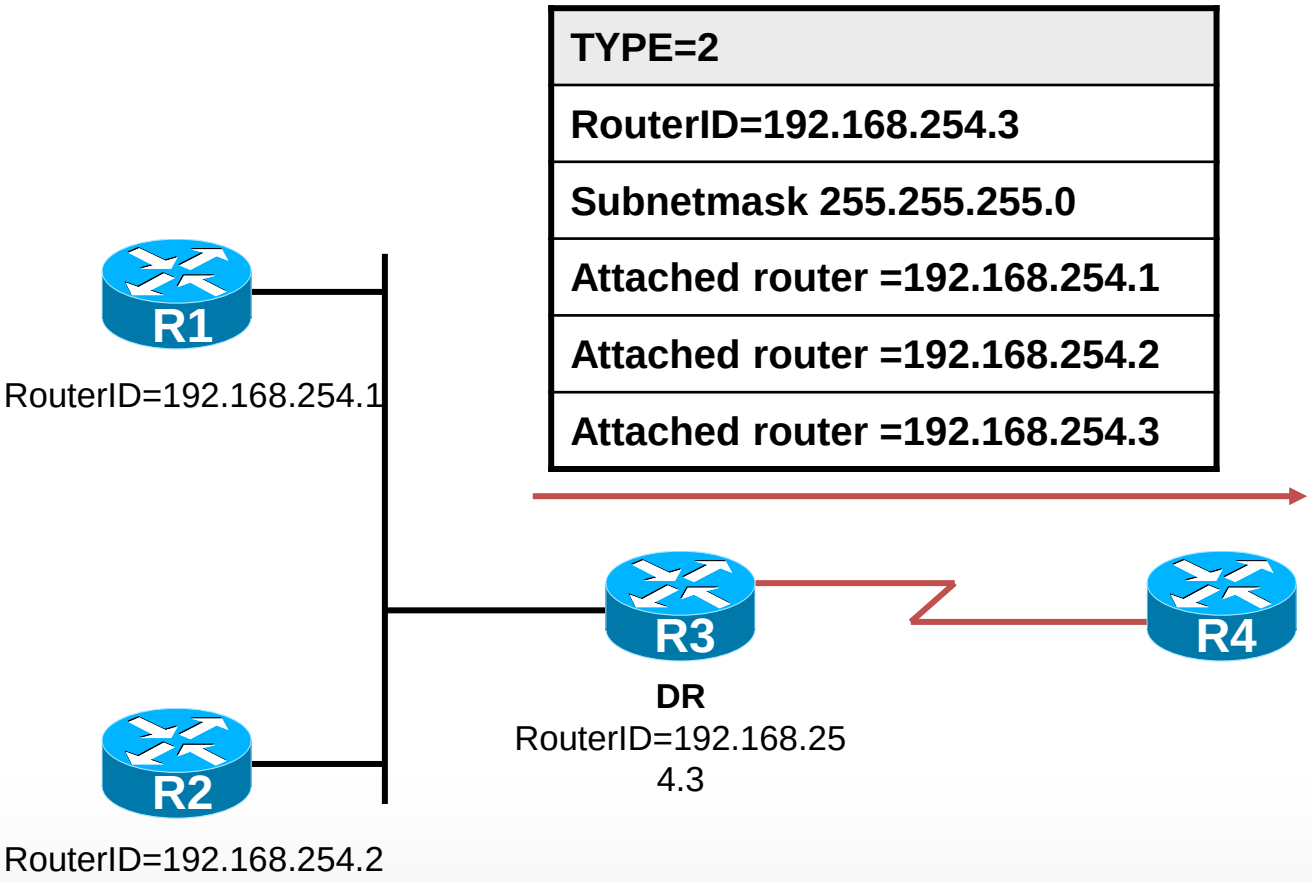


LSA类型

- **类型1：路由器LSA Router LSA**
  - 每个路由器针对它所在的区域产生LSA1，描述区域内部与路由器直连的链路的信息（包括链路类型，Cost等）；
  - LSA1只允许在本区域内洪泛，不允许跨越ABR；
  - LSA中会标识路由器是否是ABR(B比特置位),ASBR（E比特置位）或者是Virtual-link（V比特置位）的端点的身份信息；

LSA类型

- **类型2：网络LSA Network LSA**



LSA类型

- **类型2：网络LSA Network LSA**
  - 描述TransNet（包括Broadcast和NBMA网络）网络信息；
  - 由DR生成，描述其在该网络上连接的所有路由器以及网段掩码信息，以及这个MA所属的路由器；
  - LSA类型2只在本区域Area内洪泛，不允许跨越ABR；
  - Network LSA ID是DR进行宣告的那个接口的IP地址
  - Network LSA 中没有COST字段

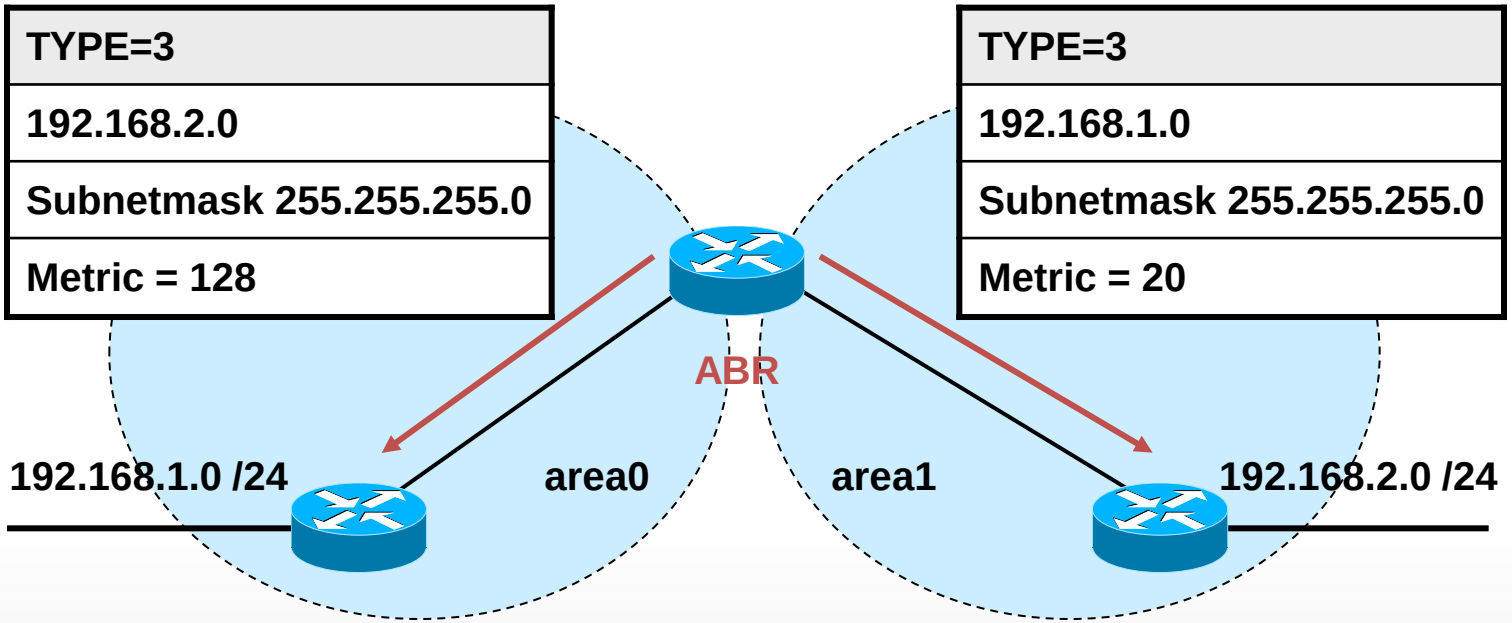
LSA类型

- **类型1、2 总结**
  - 通过LSA1，LSA2在区域内洪泛,使区域内每个路由器的LSDB达到同步，计算生成标识为 “O”的路由，解决区域内部的通信问题；

```
RT2#show ip route
Codes: C - connected. S - static,           R - RIP,              B - BGP
                                      O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default


Gateway of last resort is not set

     172.25.0.0/24 is subnetted, 1 subnets
O       172.25.1.0 [110/20] via 10.1.1.2, 03:32:41, Ethernet0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet0
RT2#
```

LSA类型

- **类型3： 网络汇总LSA Network Summary LSA**
  - 由ABR生成，实际上就是将区域内部的Type1 Type2的信息收集起来以路由子网的形式扩散出去， 这就是Summay LSA中Summay的含义（注意这里的summary与路由汇总没有关系）；



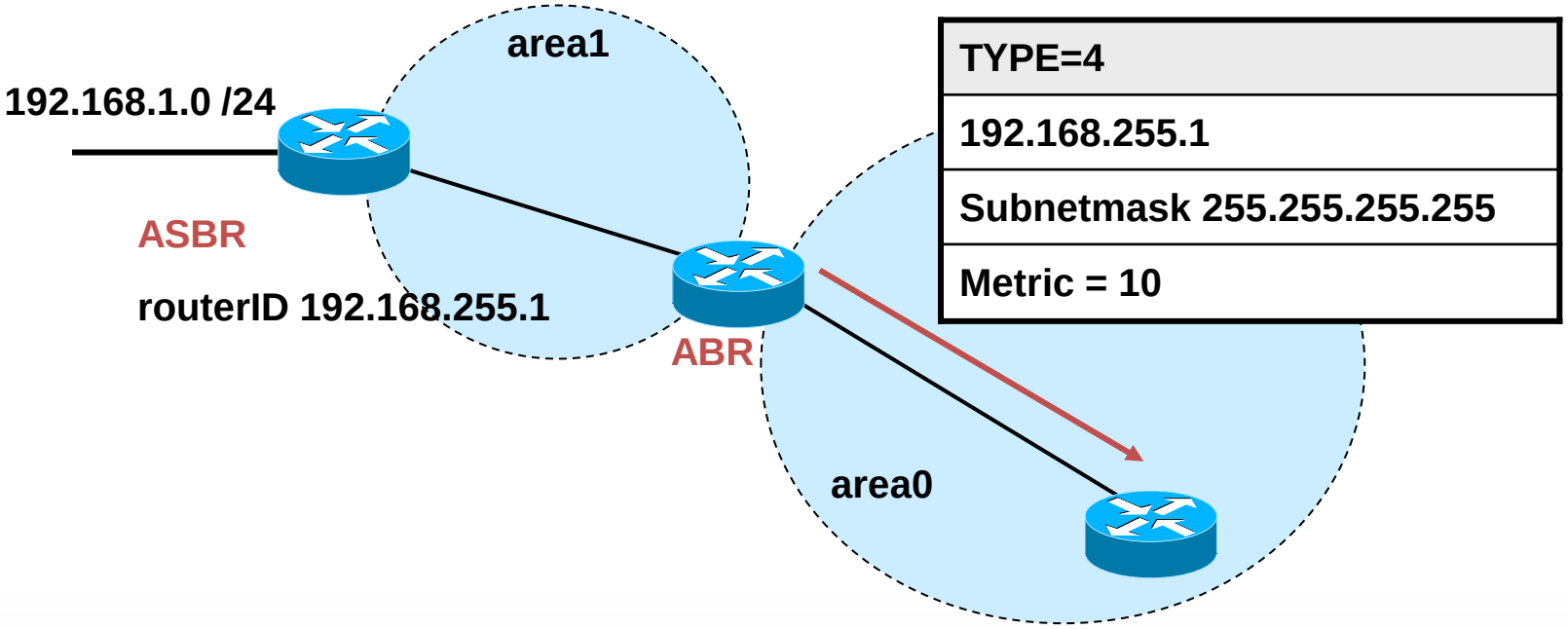
LSA类型

- **类型3： 网络汇总LSA Network Summary LSA**
  - Type 3 的链路状态ID是目的网络地址。
  - 如果一台ABR路由器在与它本身相连的区域内有多条路由可以到达目的地,那么它将只会始发单一的一条网络汇总LSA到骨干区域,而且这条网络汇总LSA是上述多条路由中代价最低的。
  - ABR收到来自同区域其它ABR传来的Type 3 LSA后重新生成新的Type3 LSA（Advertising Router改为自己）然后继续在整个OSPF系统内扩散

```
     172.25.0.0/24 is subnetted, 1 subnets
O IA    172.25.1.0 [110/20] via 10.1.1.2, 00:00:01, Ethernet0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet0
```

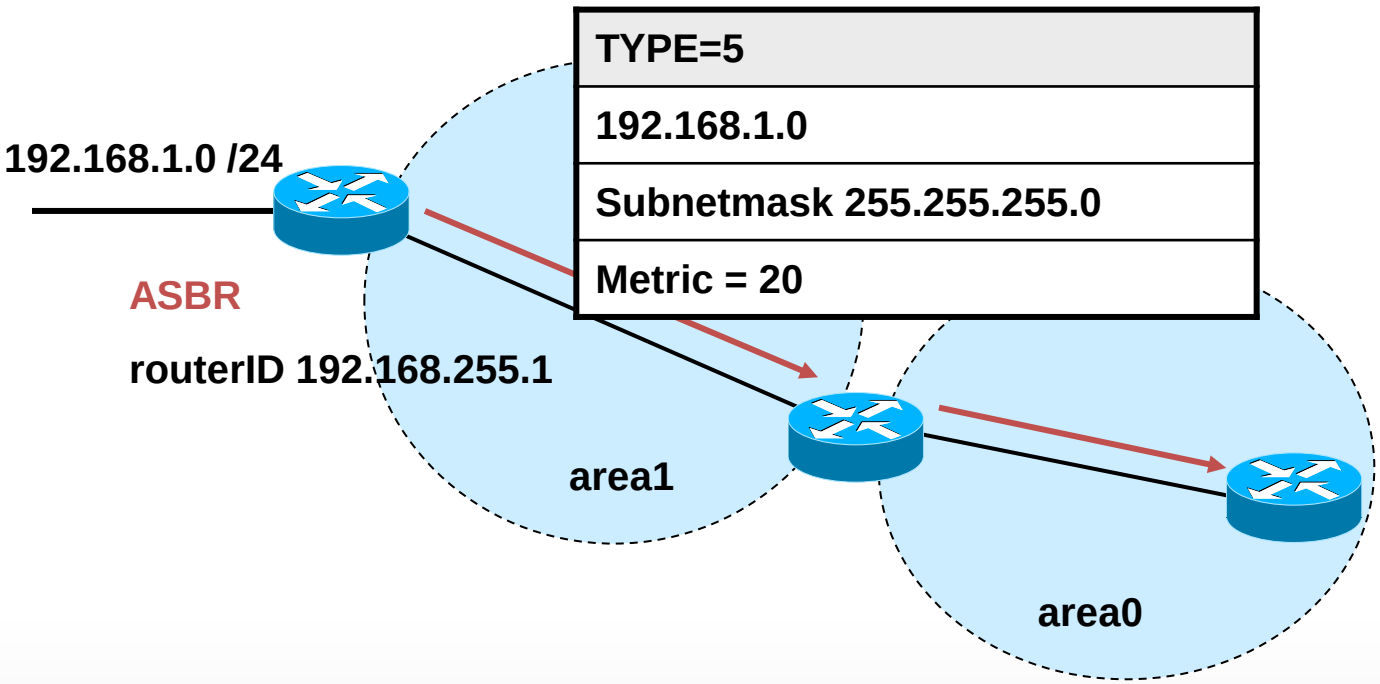
LSA类型

- **类型4： ASBR Summary LSA**
  - ASBR Summary LSA由ABR生成，用于描述ABR能够到达的ASBR它的链路状态ID为目的ASBR的RID。



LSA类型

- **类型5： 自治系统外LSA　AS External LSA**
  - Autonomous System External LSA由ASBR生成用于描述OSPF自治域系统外的目标网段信息链路状态ID是目的地址的IP网络号。

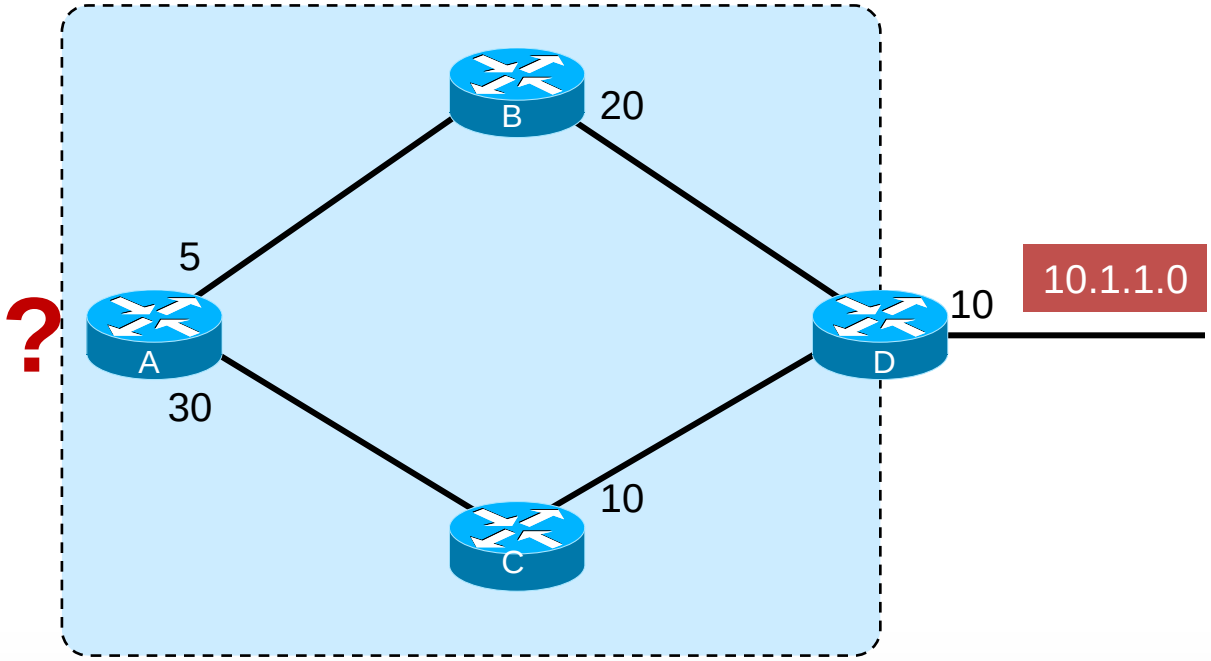


LSA类型

- **类型5： 自治系统外LSA　AS External LSA**
  - 外部路由通过重发布，引入OSPF路由域，相应信息(路由条目)由ASBR以LSA5的形式生成然后进入OSPF路由域；
  - 缺省情况下，LSA5生成路由用OE2表示，可强行指定为OE1；
    - OE2 开销 = 外部开销；
    - OE1 开销 = 外部开销 + 内部开销；
  - LSA5不允许进入特殊区域 —— stub存根区& NSSA区；

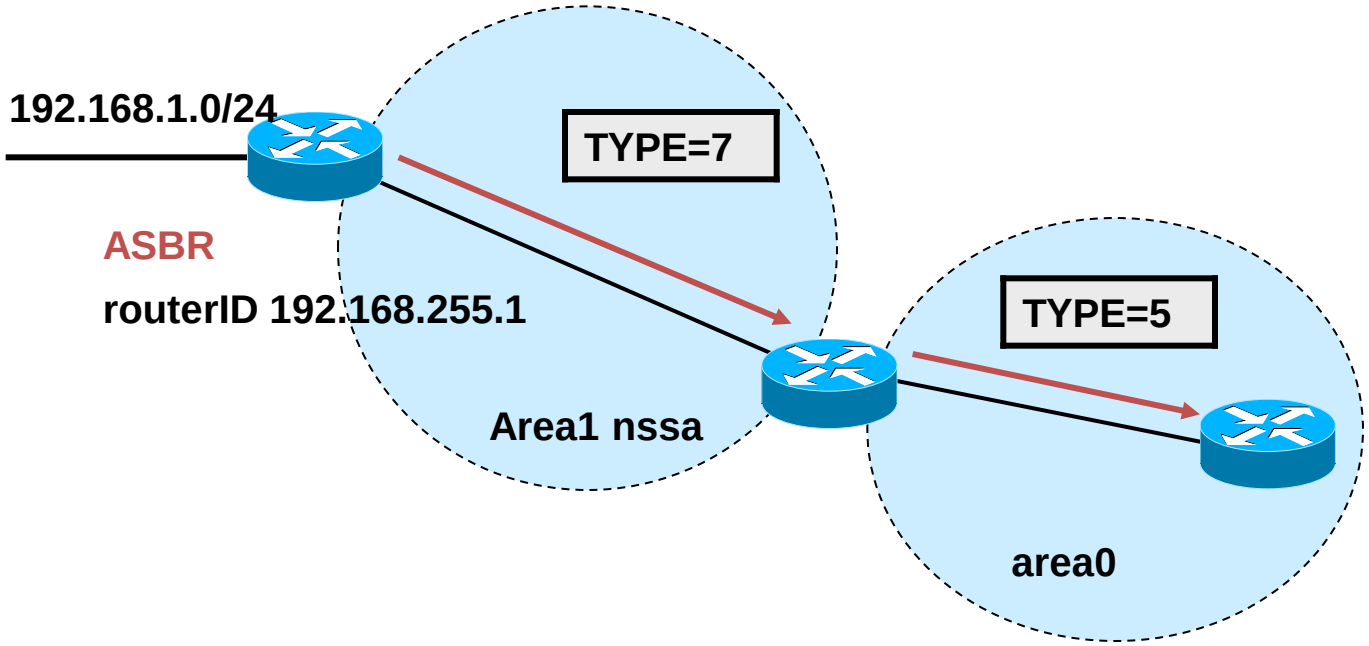
LSA类型

- **类型5： 自治系统外LSA　AS External LSA**
  - OE1、OE2的区别



LSA类型

- **类型7： NSSA中的外部LSA NSSA External LSA**



LSA类型

- **类型7： NSSA中的外部LSA NSSA External LSA**
  - 在NSSA(非完全存根区域)not-so-stubby area中ASBR针对外部网络产生类似于LSA5的LSA类型7,
  - LSA类型7只能在NSSA区域中洪泛，到达NSSA区域ABR后，NSSA ABR将其转换成LSA类型5外部路由，传播到Area 0，从而传播到整个OSPF路由域
  - 生成路由缺省用ON2表示，也可指定为ON1；

LSA类型

- **类型7： NSSA中的外部LSA NSSA External LSA**

```
RT4#show ip route
Codes: C - connected, S - static, I - IGRP,    - RIP,              B - BGP
                                        O -    PF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 -    PF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF    ternal type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 -     IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate    fault,


Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
O N2    172.16.1.0 [110/20] via 10.1.1.1, 0    1:17, Serial1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ser    1
C    192.168.1.0/24 is directly connected,     ial0
RT4#
```

LSA类型

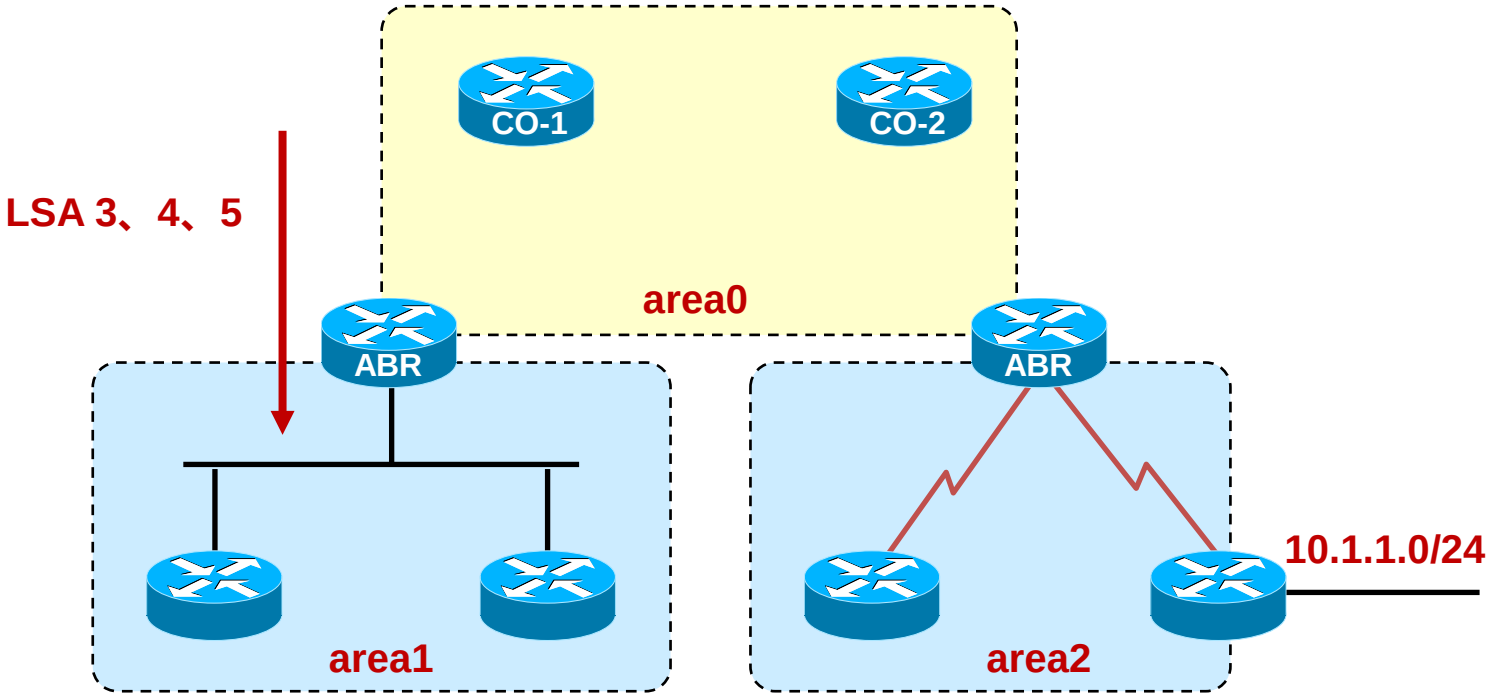
- **OSPF LSDB和路由表**
  - 查看LSDB：show ip ospf database
  - 查看路由表：show ip route ospf

LSA类型

- **OSPF LSDB和路由表**
  - 查看路由表：show ip route ospf
  - O > O IA > O E1 > O E2

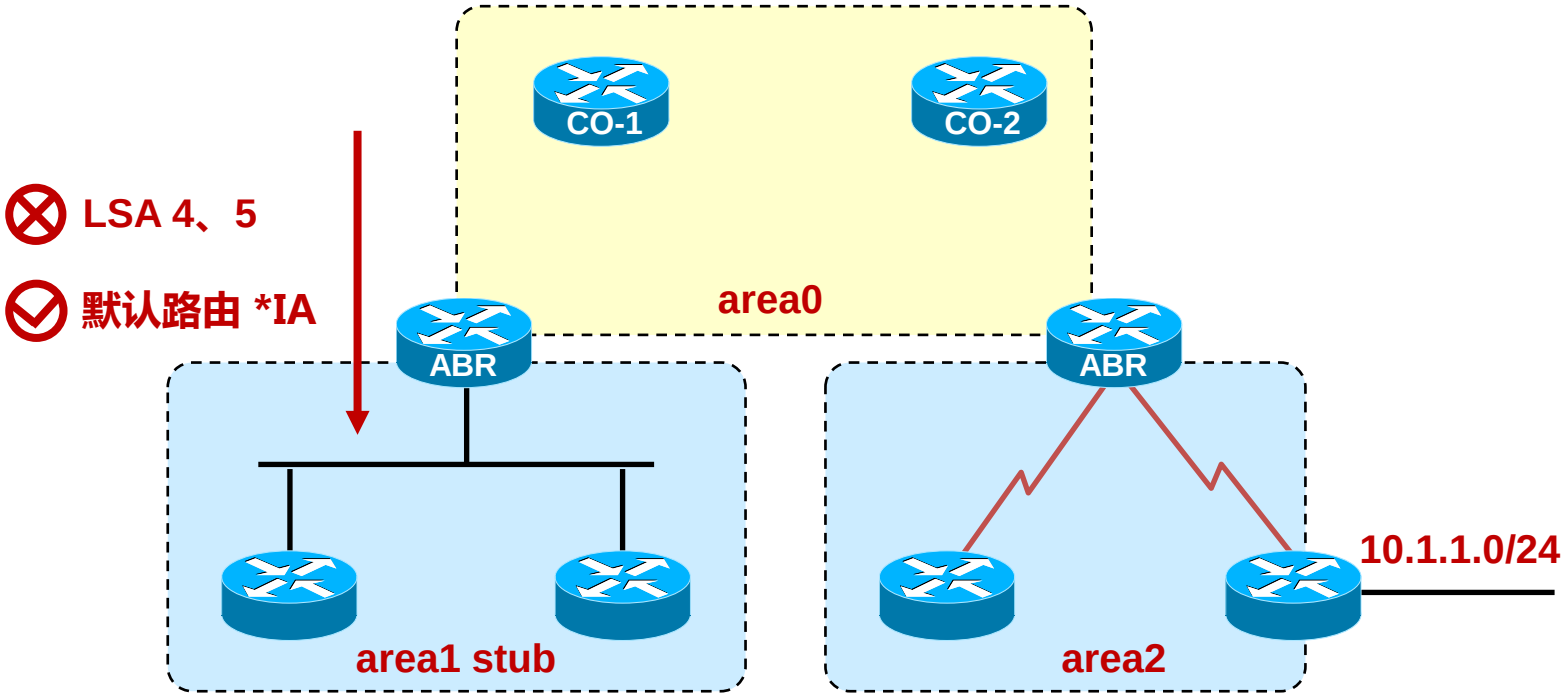
路由指示符	路由类型
O	OSPF 区域内路由
O IA	OSPF 区域间路由
O E1	1 类外部路由
O E2	2 类外部路由

OSPF特殊区域



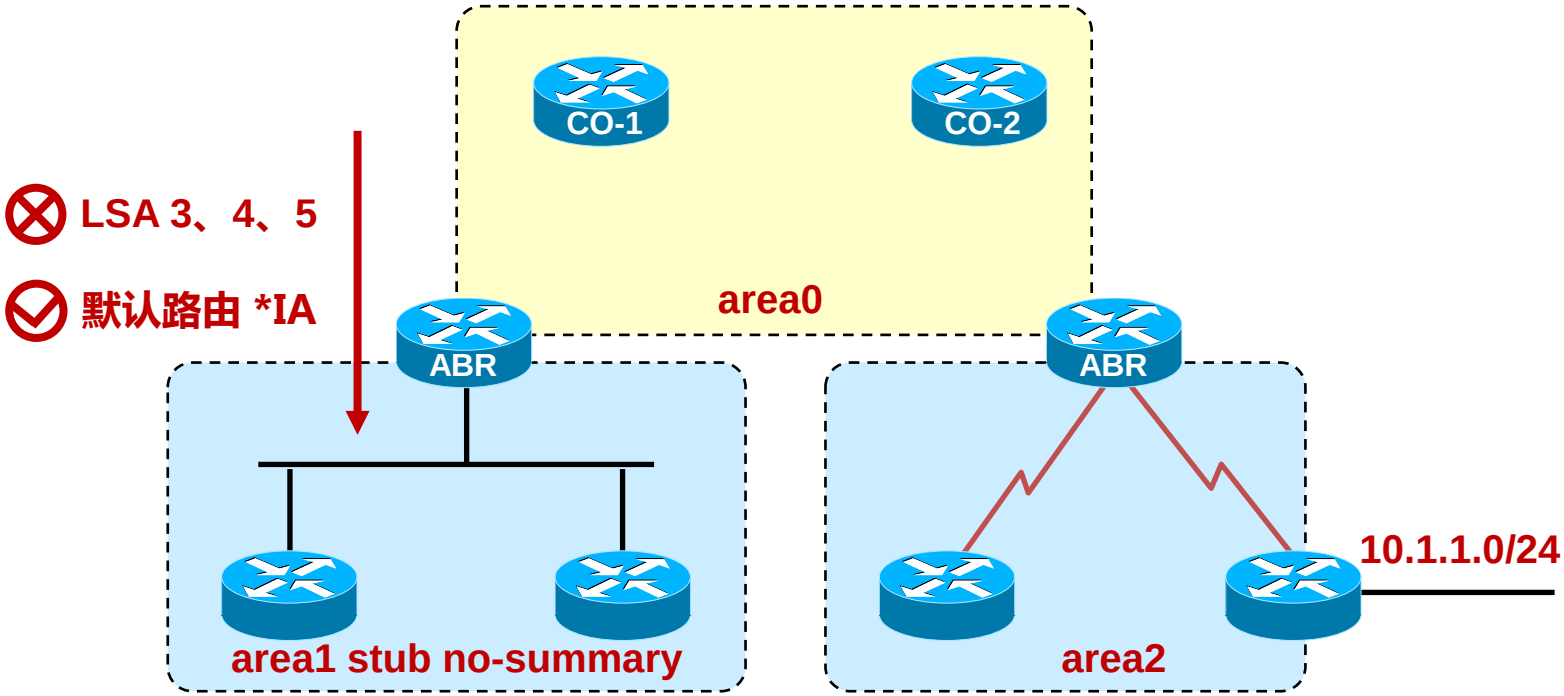
OSPF特殊区域

- **末梢区域 stub area**



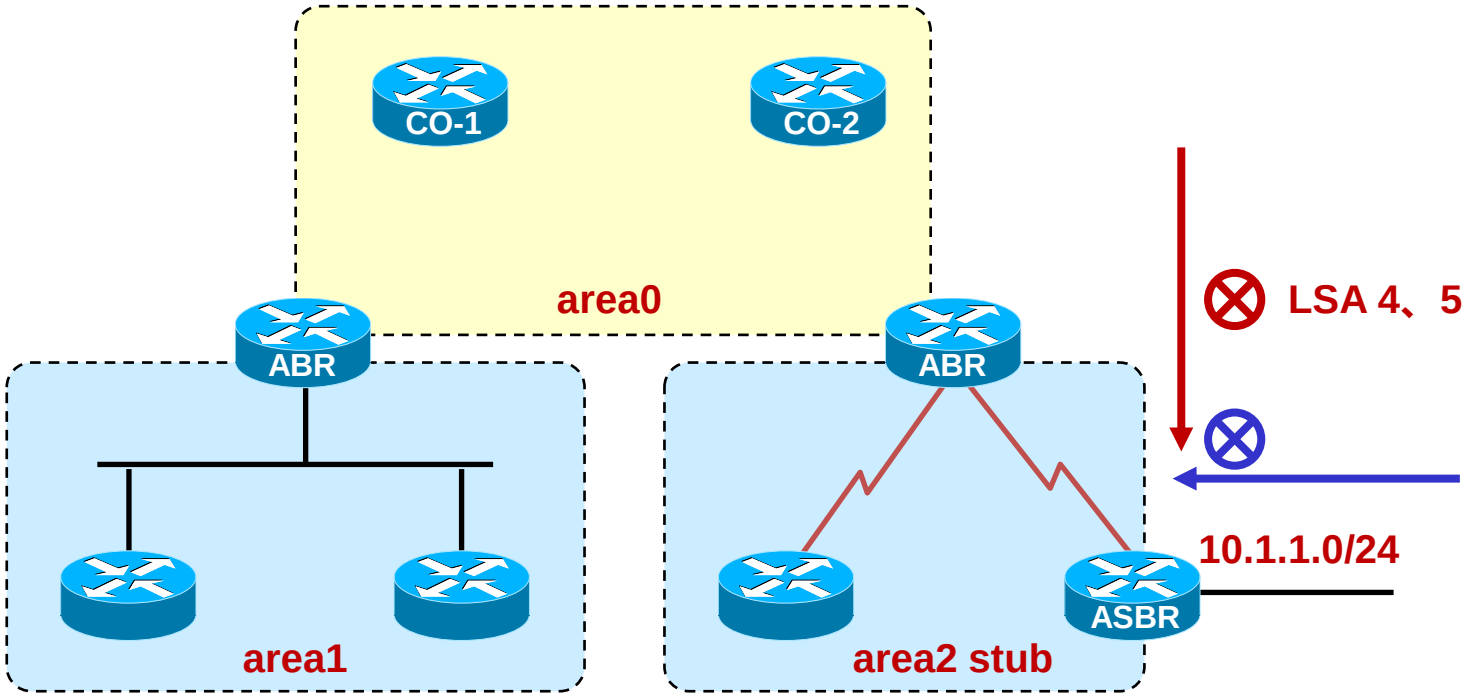
OSPF特殊区域

- **完全末梢区域 totally stub area**



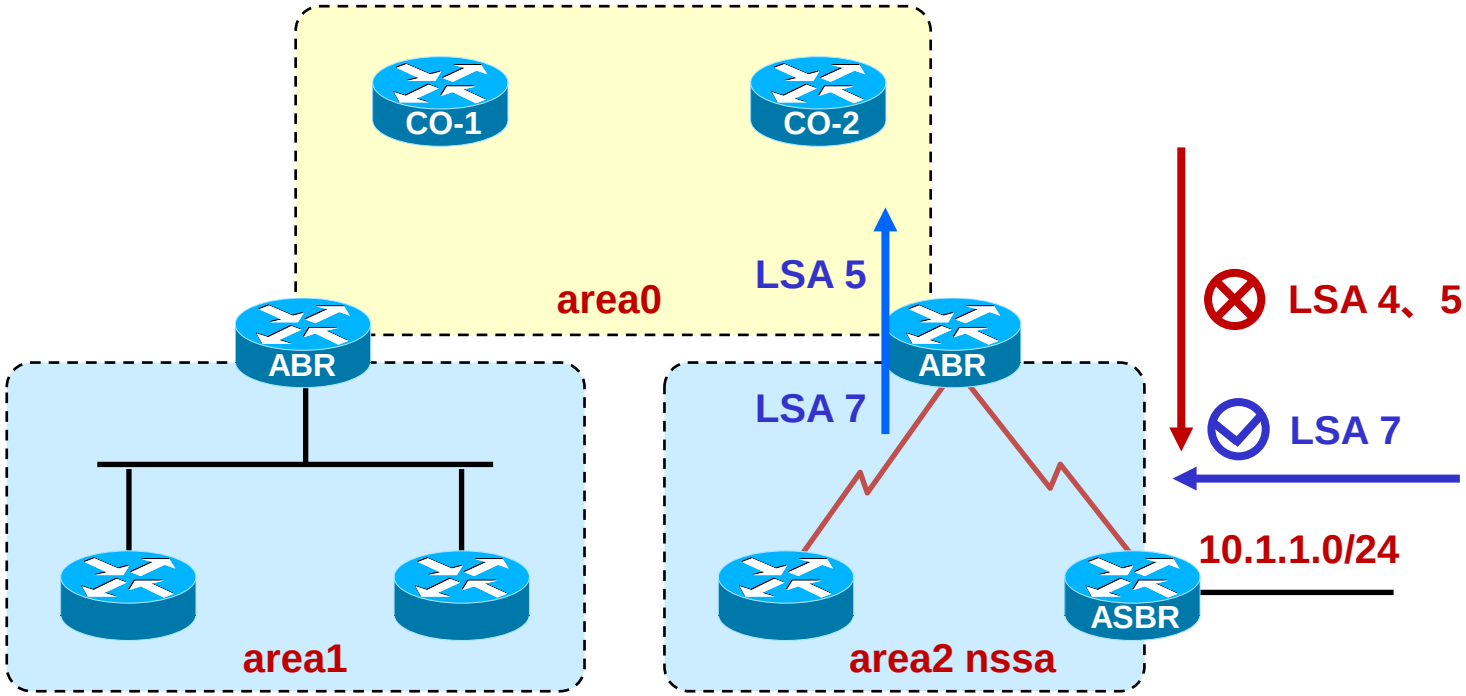
OSPF特殊区域

- **非完全末梢区域 not-so-stubby area**



OSPF特殊区域

- **非完全末梢区域 not-so-stubby area**



OSPF特殊区域

- **OSPF区域的类型与LSA的洪泛范围**

OSPF区域总结

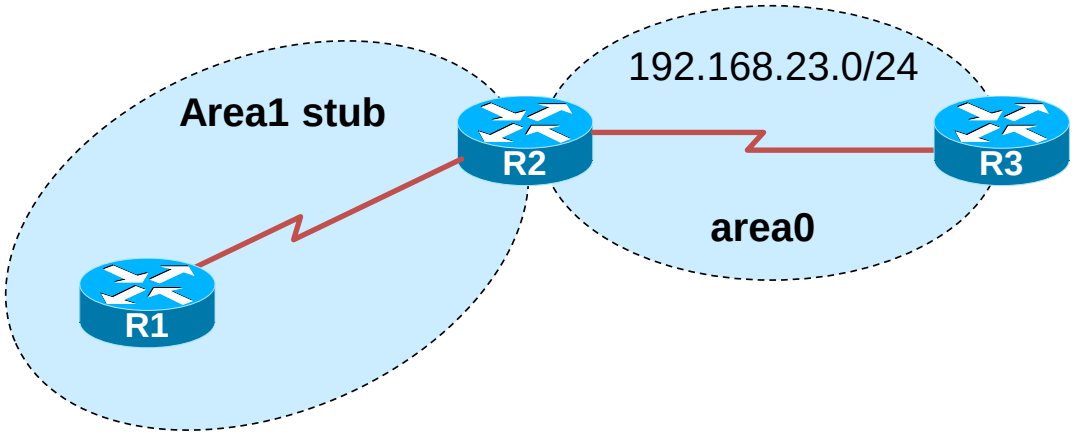
OSPF特殊区域

- **OSPF区域的类型与LSA的洪泛范围**
  - 一个区域所设置的特性决定着它能接收的路由,OSPF将整个OSPF路由域划分为不同的区域,目的是为减少不必要的路由信息的传递，精简路由表。

LSA类型 / Area Type	1&2	3	4	5	7
骨干区域(Area 0)	Yes	Yes	Yes	Yes	No
非骨干标准区域(Non-area 0)	Yes	Yes	Yes	Yes	No
存根区域 (Stub Area)	Yes	Yes	No	No	No
完全存根区域 (Totally Stub Area)	Yes	No*	No	No	No
NSSA区域 (Not-so-stubby Area)	Yes	YES	No	No	Yes

OSPF特殊区域

- **特殊区域的配置 stub**



R1的配置

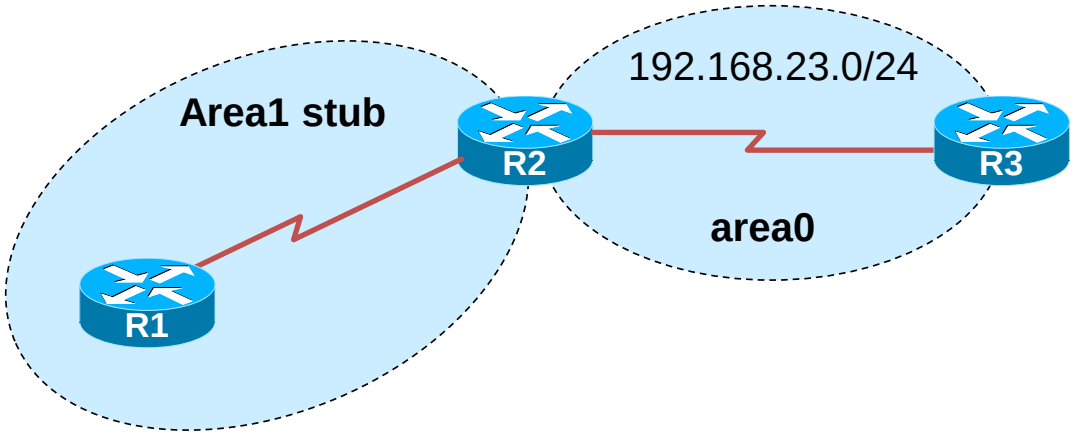
```
router ospf 1
 network 192.168.12.0 0.0.0.255 area 1
 area 1 stub
```

R2的配置

```
router ospf 1
 network 192.168.12.0 0.0.0.255 area 1
 network 192.168.23.0 0.0.0.255 area 0
 area 1 stub
```

OSPF特殊区域

- **特殊区域的配置 totally stub**



R1的配置

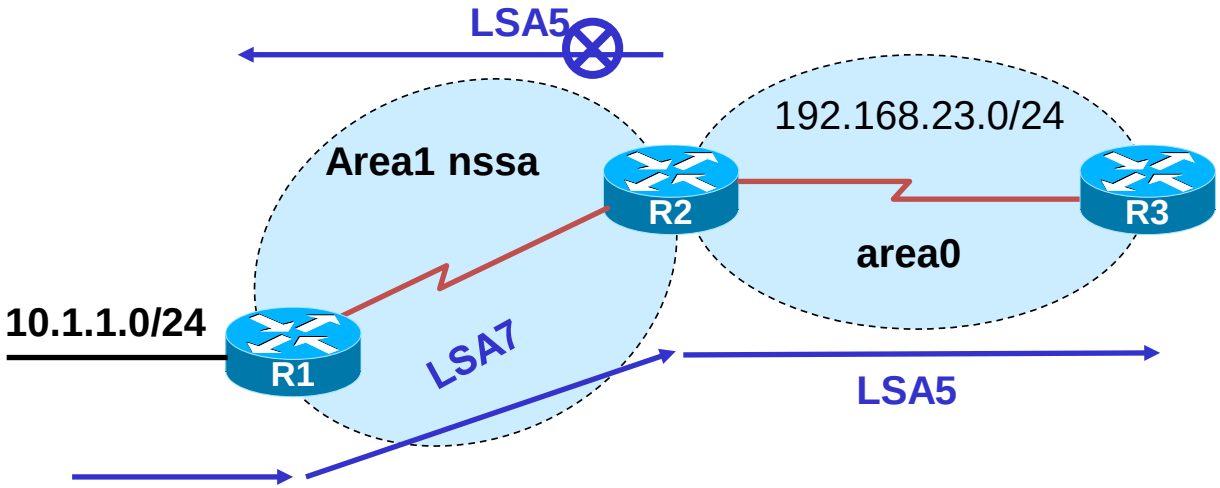
```
router ospf 1
 network 192.168.12.0 0.0.0.255 area 1
 area 1 stub
```

R2的配置

```
router ospf 1
 network 192.168.12.0 0.0.0.255 area 1
 network 192.168.23.0 0.0.0.255 area 0
 area 1 stub no-summary
 area 1 default-cost 10
```

OSPF特殊区域

- **特殊区域的配置 nssa**



R1的配置

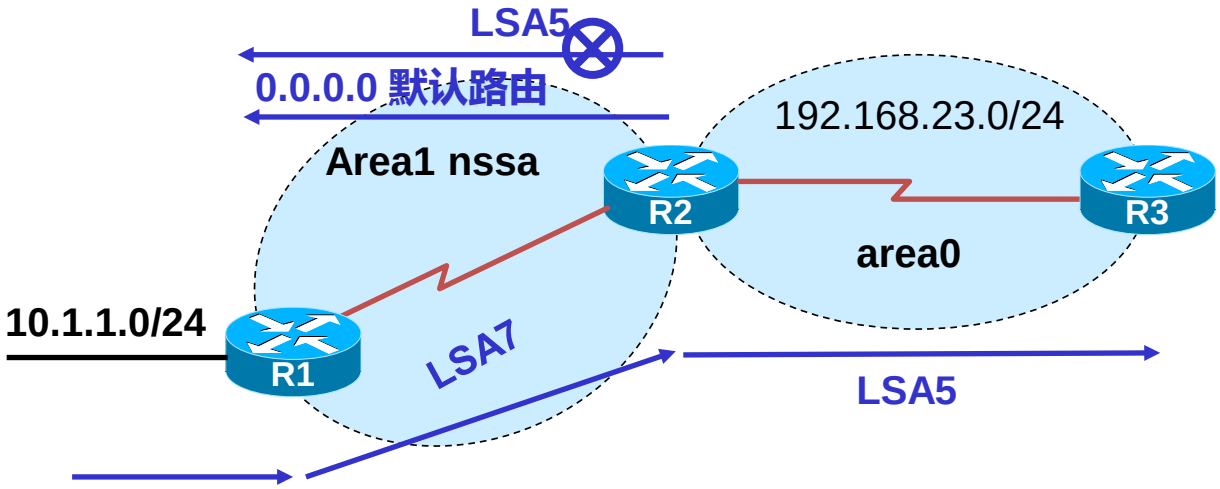
```
router ospf 1
 network 192.168.12.0 0.0.0.255 area 1
 area 1 nssa
```

R2的配置

```
router ospf 1
 network 192.168.12.0 0.0.0.255 area 1
 network 192.168.23.0 0.0.0.255 area 0
 area 1 nssa
```

OSPF特殊区域

- **特殊区域的配置 nssa**



R1的配置

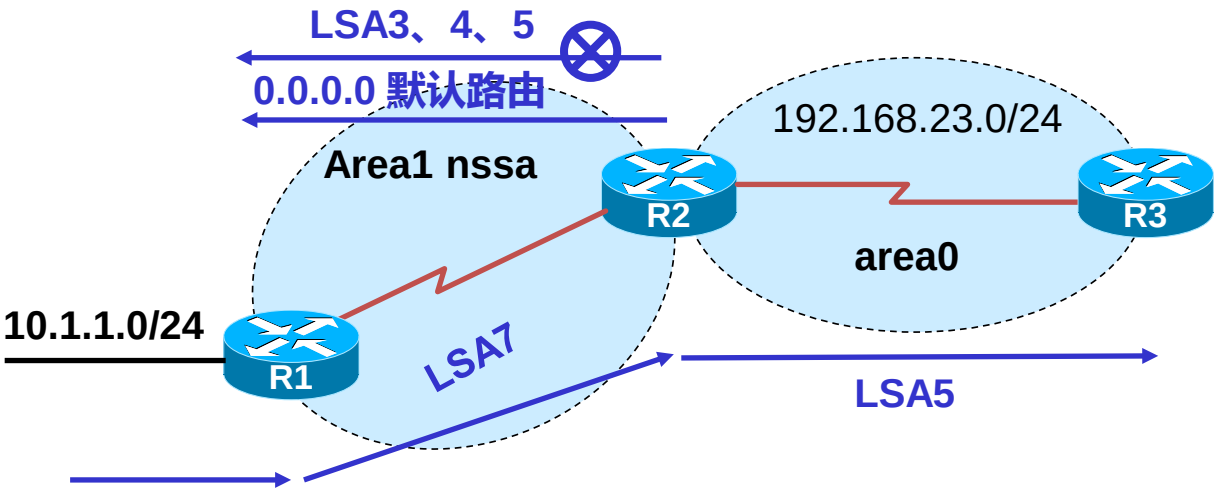
```
router ospf 1
 network 192.168.12.0 0.0.0.255 area 1
 area 1 nssa
```

R2的配置

```
router ospf 1
 network 192.168.12.0 0.0.0.255 area 1
 network 192.168.23.0 0.0.0.255 area 0
 area 1 nssa default-information-originate
```

OSPF特殊区域

- **特殊区域的配置 完全nssa**



R1的配置

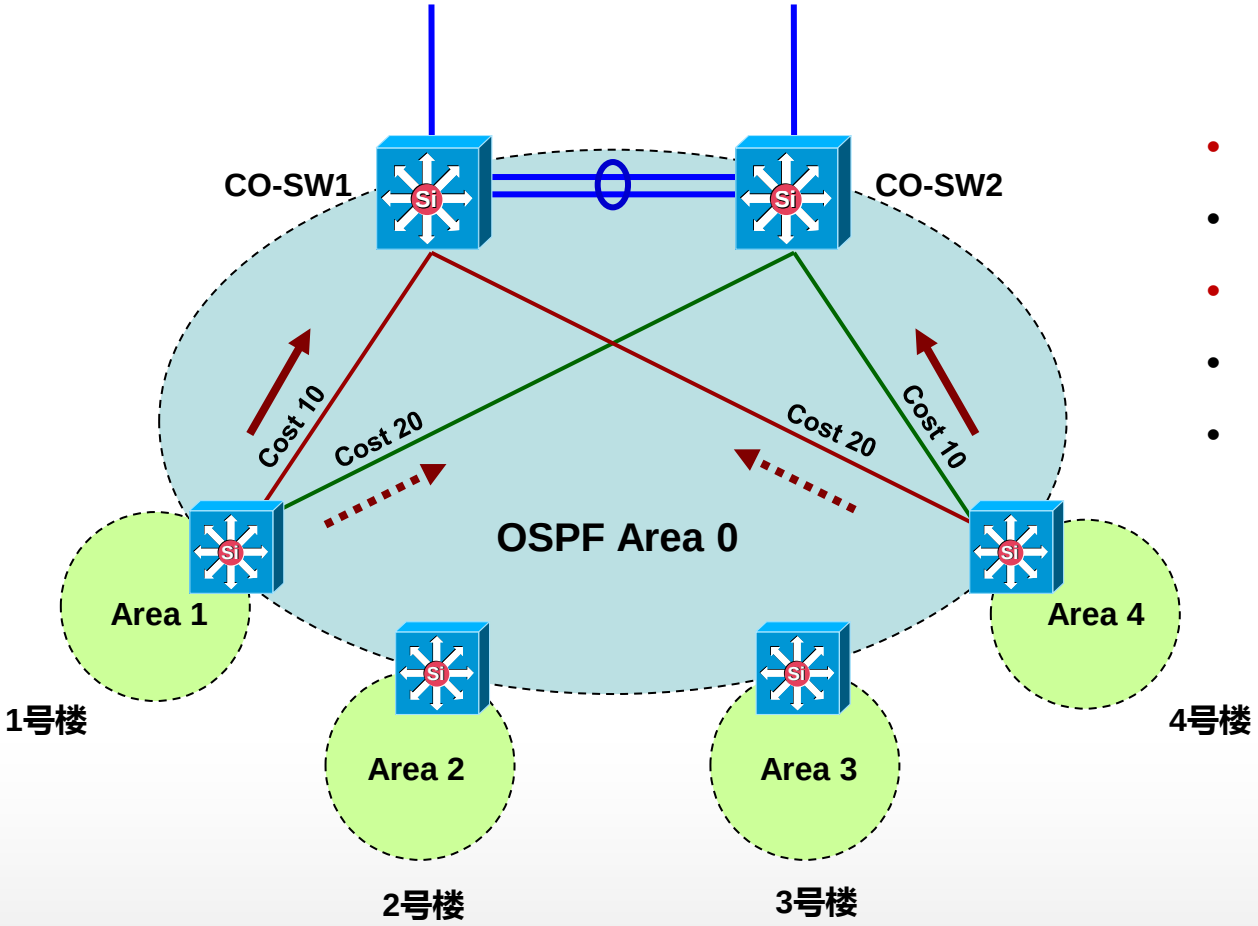
```
router ospf 1
 network 192.168.12.0 0.0.0.255 area 1
 area 1 nssa
```

R2的配置

```
router ospf 1
 network 192.168.12.0 0.0.0.255 area 1
 network 192.168.23.0 0.0.0.255 area 0
 area 1 nssa no-summary
```

OSPF特殊区域

- **特殊区域的配置 特殊区域在工程中的运用**



- **区域划分（含特殊区域）**
- **路由汇总**
- **主备线路(COST)**
- **默认路由传递**
- **Passive-interface**

OSPF特殊区域

- **OSPF特殊区域**

  查看

  – Show ip ospf

  – show ip ospf database

  – show ip ospf database ?

  – show ip route

红茶三杯
Vinsoney

沉淀 提升 成长 分享
关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

OSPF高级特性及配置

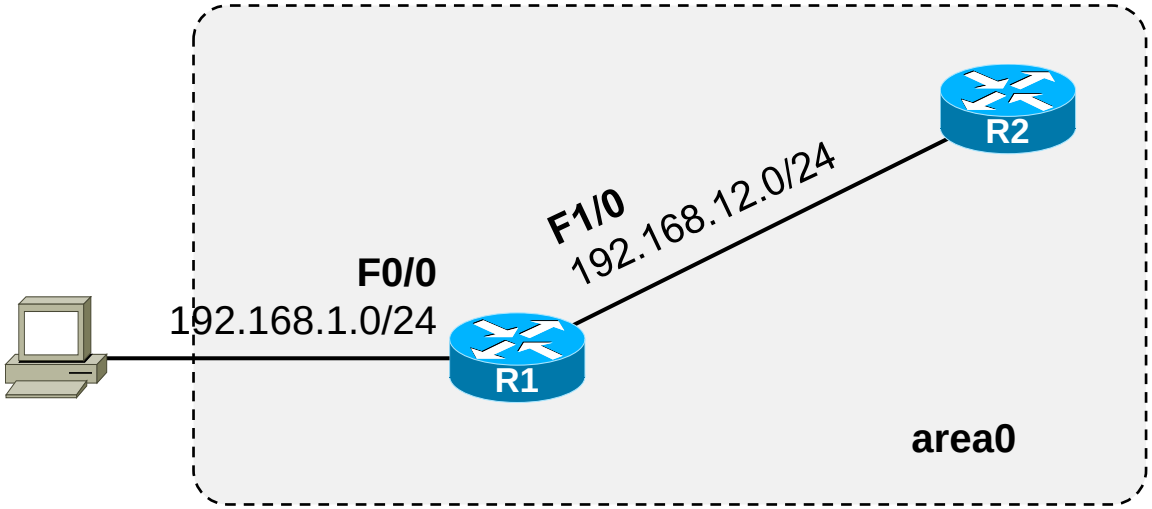
红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2012-08-01

课程目标

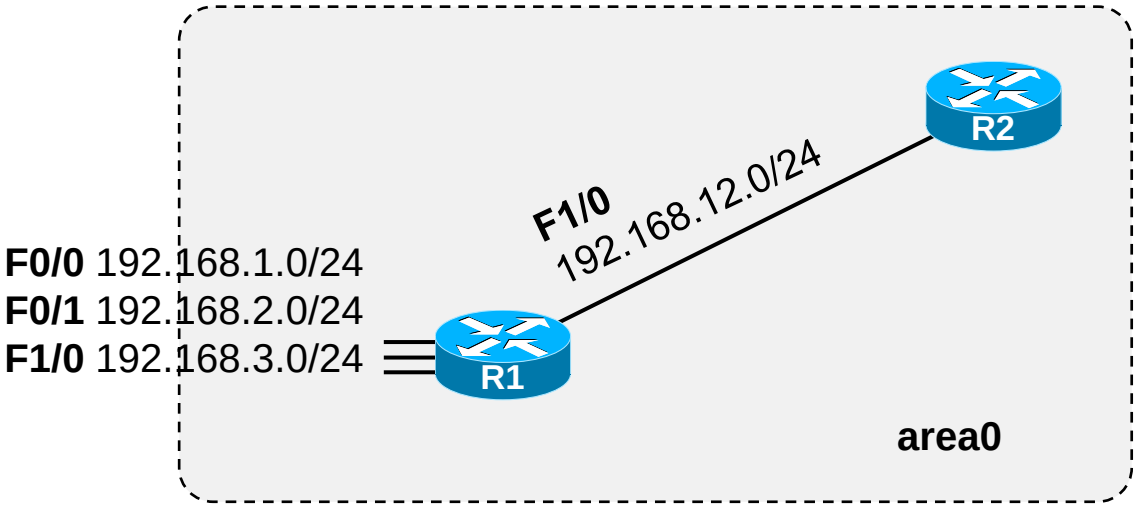
- Passive-interface
- OSPF默认路由的注入
- 路由汇总
- 虚链路配置
- OSPF验证

Passive-interface




```
router ospf 1
  network 192.168.1.0 0.0.0.255 area 0
  network 192.168.12.0 0.0.0.255 area 0
  passive-interface fa 0/0
```

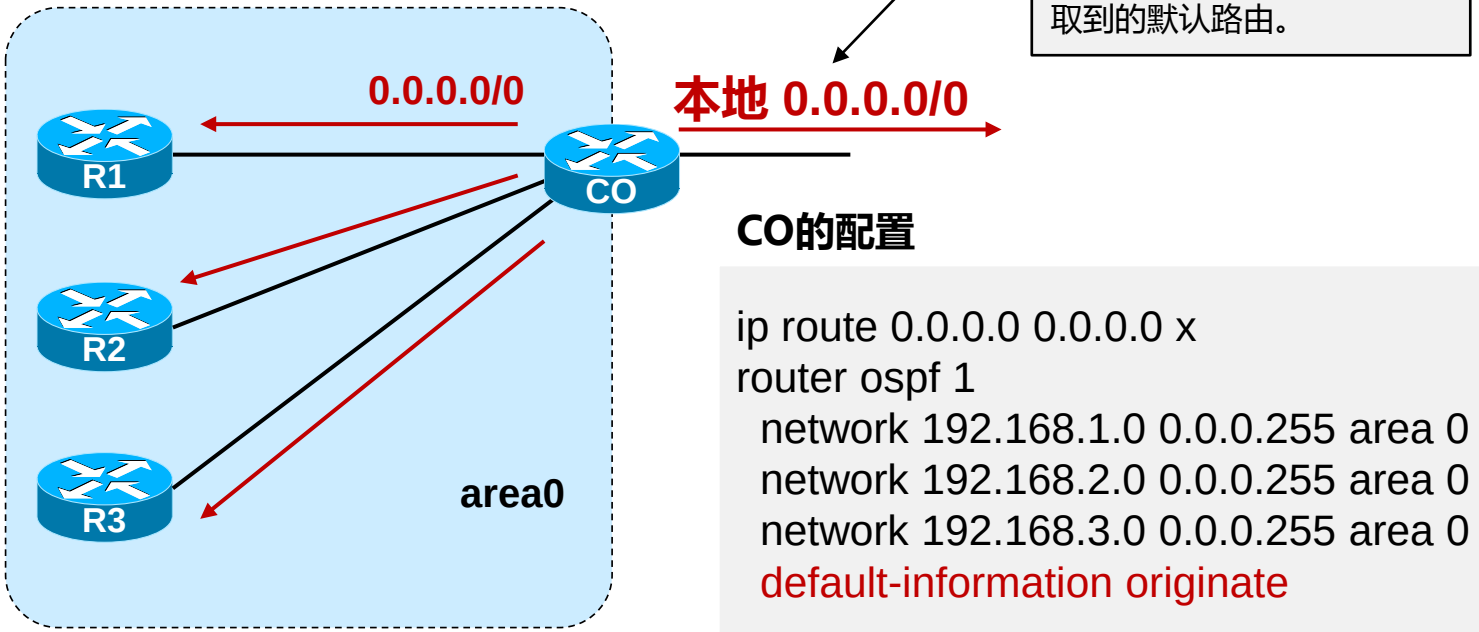
Passive-interface




```
router ospf 1
  network 192.168.1.0 0.0.0.255 area 0
  network 192.168.2.0 0.0.0.255 area 0
  network 192.168.3.0 0.0.0.255 area 0
  network 192.168.12.0 0.0.0.255 area 0
  passive-interface default
  no passive-interface fa 2/0
```

OSPF默认路由的注入

- **default-information originate**

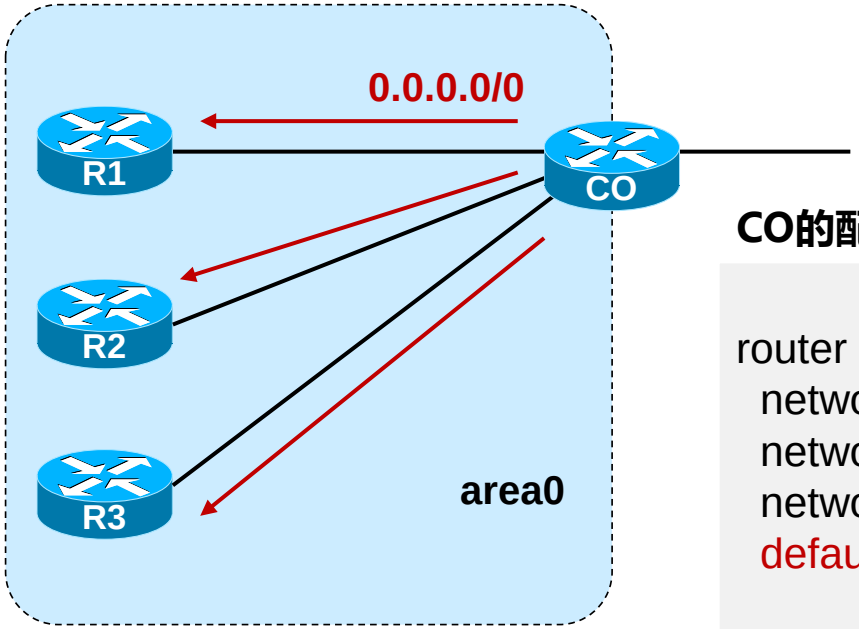


CO的配置

```
ip route 0.0.0.0 0.0.0.0 x
router ospf 1
  network 192.168.1.0 0.0.0.255 area 0
  network 192.168.2.0 0.0.0.255 area 0
  network 192.168.3.0 0.0.0.255 area 0
  default-information originate
```

OSPF默认路由的注入

- **default-information originate always**

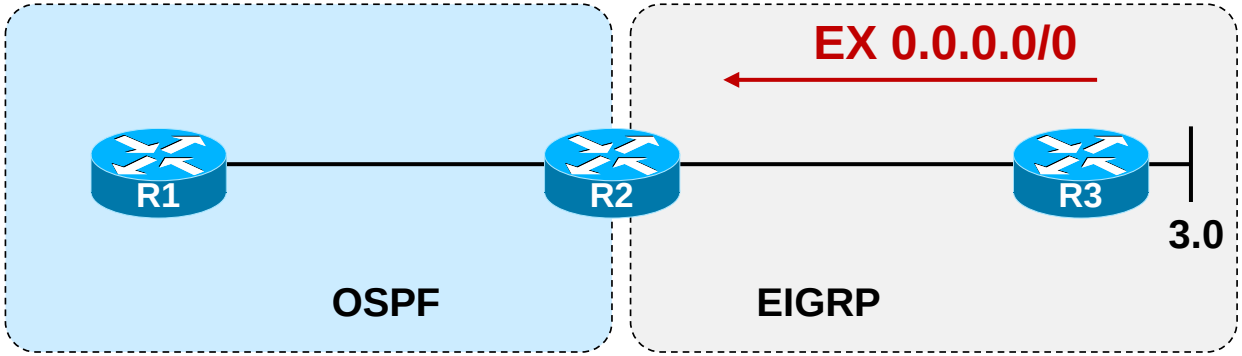


CO的配置

```
router ospf 1
  network 192.168.1.0 0.0.0.255 area 0
  network 192.168.2.0 0.0.0.255 area 0
  network 192.168.3.0 0.0.0.255 area 0
  default-information originate always
```

OSPF默认路由的注入

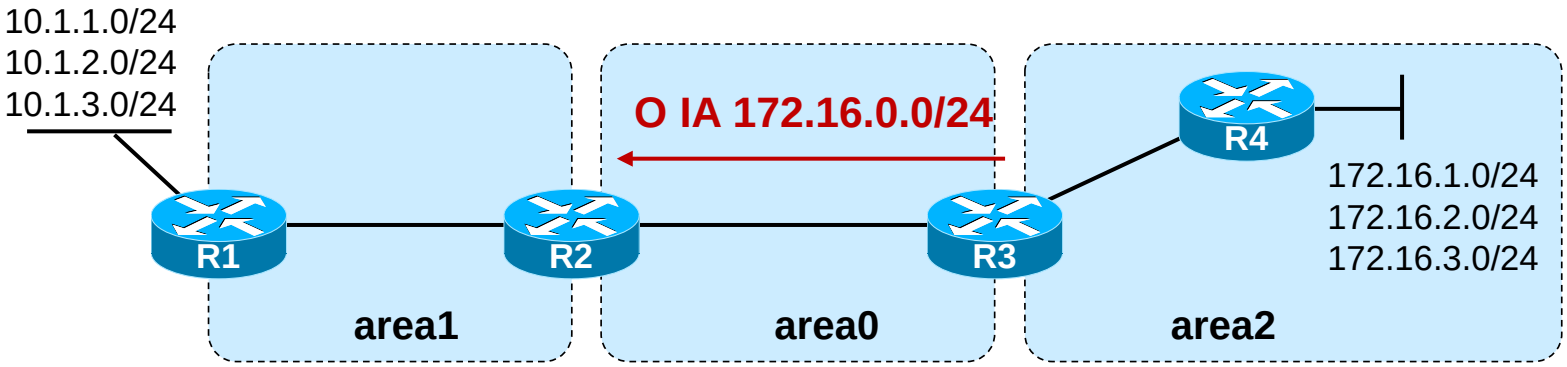
- **redistribute**



使用重发布的方式，无法将本地的静态默认路由或从其他协议学习到的默认路由注入OSPF

OSPF路由汇总

- **区域内路由汇总**

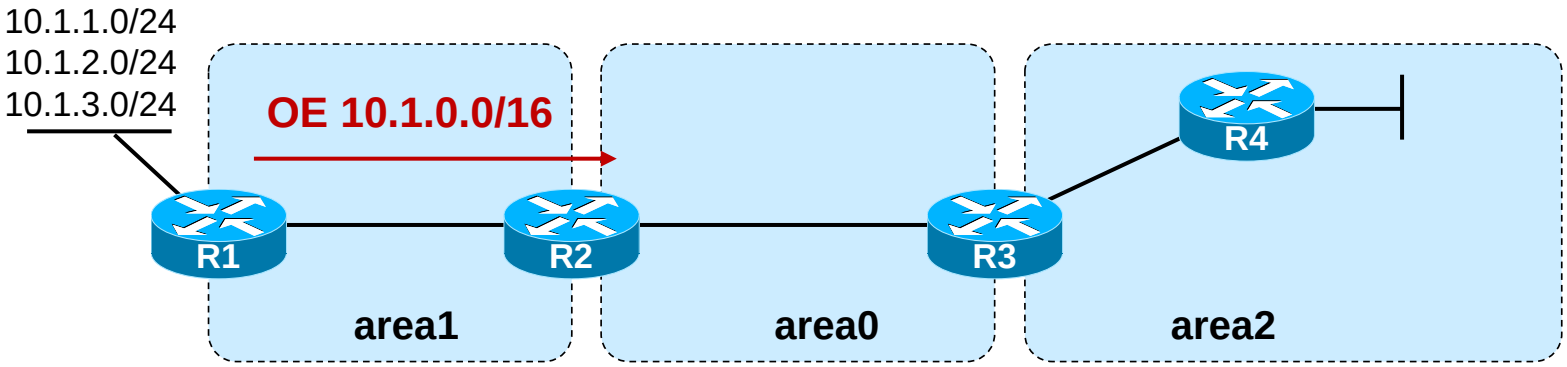


R3的配置（ABR）

```
router ospf 1
  area 2 range 172.16.0.0 255.255.0.0 cost ?
```

OSPF路由汇总

- **外部路由汇总**

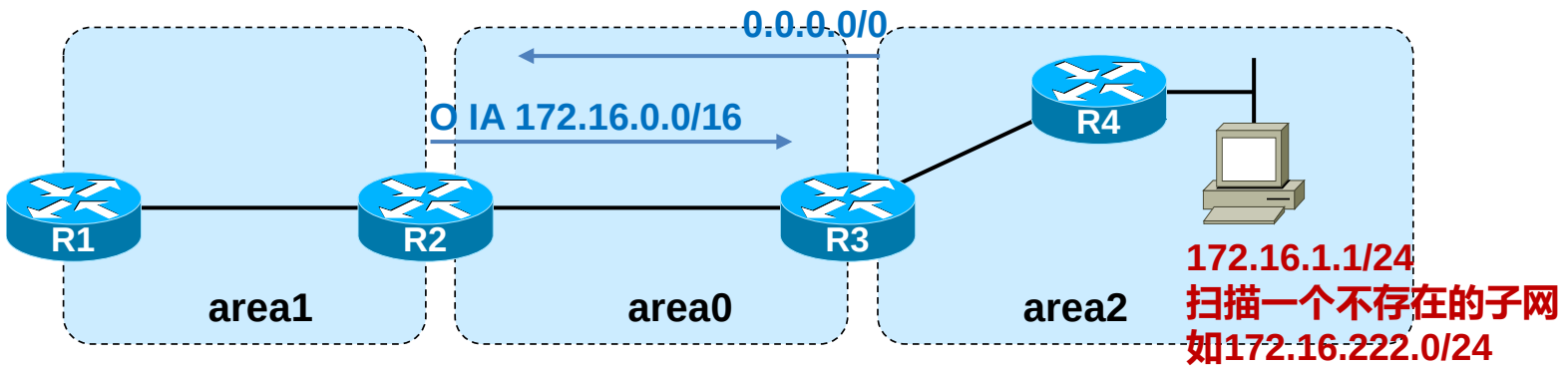


R1的配置（ASBR）

```
router ospf 1
  summary-address 10.1.0.0 255.255.0.0
```

OSPF路由汇总

- **OSPF汇总路由的防环**

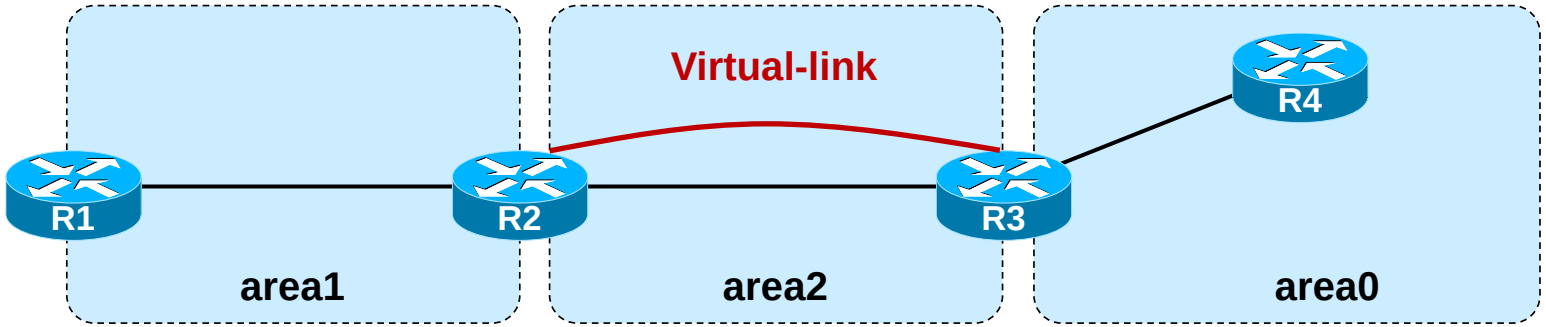


R3、R4都有默认路由出去，R3作为ABR，将172.16.0.0子网进行汇总，将汇总路由传递给了骨干区域。

此时R4下某个子网里，有PC在扫描一个不存在的子网内的IP，如172.16.222.0/24，这些数据包会被默认路由匹配一路传递到R2，而R2上由于收到R3传递过来的汇总路由，因此又把这些数据包丢回去给R3，R3又丢回R2，如此反复，直到报文TTL为0。

OSPF为了解决这个问题，在进行汇总时，OSPF会在R3上会在本地自动产生一条指向Null0的汇总路由，这样一来当再类似事件发生，数据包将在R3这就被丢弃。

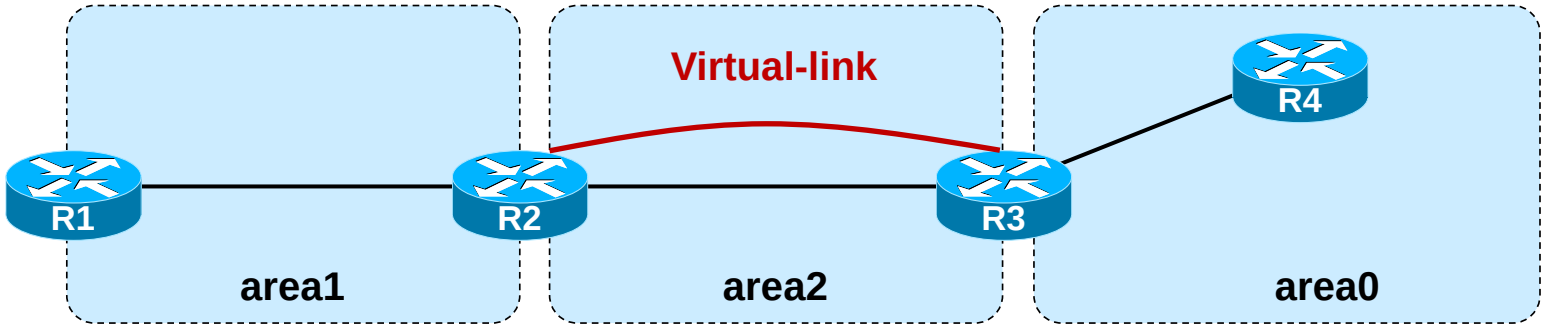
Virtual-link



router ospf 1
area area-id virtual-link router-id [authentication [message-digest | null]] [hello-interval seconds] [retransmit-interval seconds] [transmit-delay seconds] [dead-interval seconds] [[authentication-key key] | [message-digest-key key-id md5 key]]

Virtual-link

- **配置示例**



R2的配置

```
router ospf 1
 area 2 virtual-link 3.3.3.3
```

R3的配置

```
router ospf 1
 area 2 virtual-link 2.2.2.2
```

Virtual-link

- **验证**
  - show ip ospf virtual-link
  - show ip ospf neighbor
  - show ip ospf database
  - debug ip ospf adj

OSPF身份验证

- **OSPF身份验证**
  - Null
  - 简单密码身份验证
  - MD5身份验证
  - 接口认证
  - 区域认证

OSPF身份验证

- **OSPF身份验证（明文）**

接口认证

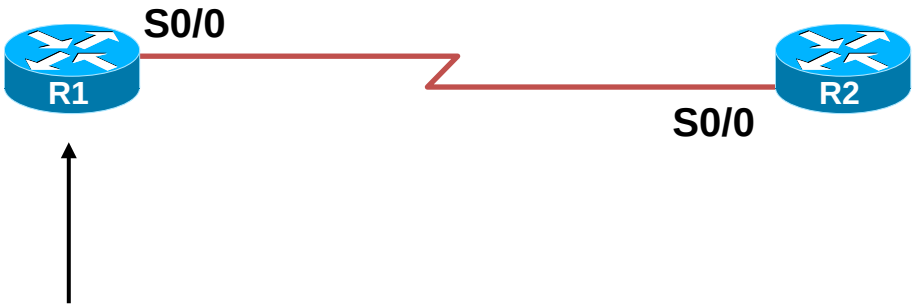
```
Router(config-if)# ip ospf authentication-key password
Router(config-if)# ip ospf authentication
```

区域认证

```
Router(config-if)# ip ospf authentication-key password
Router(config-router)# area area-id authentication
```

OSPF身份验证

- **OSPF身份验证（明文）配置示例**




```
interface s0/0
  ip ospf authentication-key SPOTO
  ip ospf authentication
router ospf 1
  router-id 1.1.1.1
  network 192.168.12.0 0.0.0.255 area 0
```

OSPF身份验证

- **OSPF身份验证（密文）**

接口认证

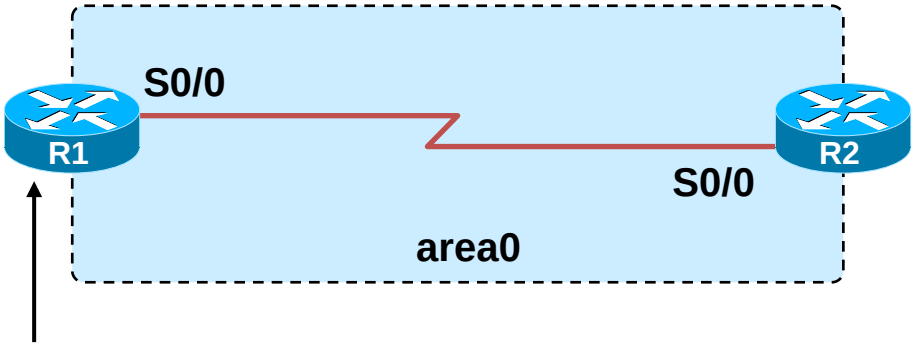
```
Router(config-if)# ip ospf message-digest-key key-id md5 key
Router(config-if)# Ip ospf authentication message-digest
```

区域认证

```
Router(config-if)# ip ospf message-digest-key key-id md5 key
Router(config-router)# area 0 authentication messae-digest
```

OSPF身份验证

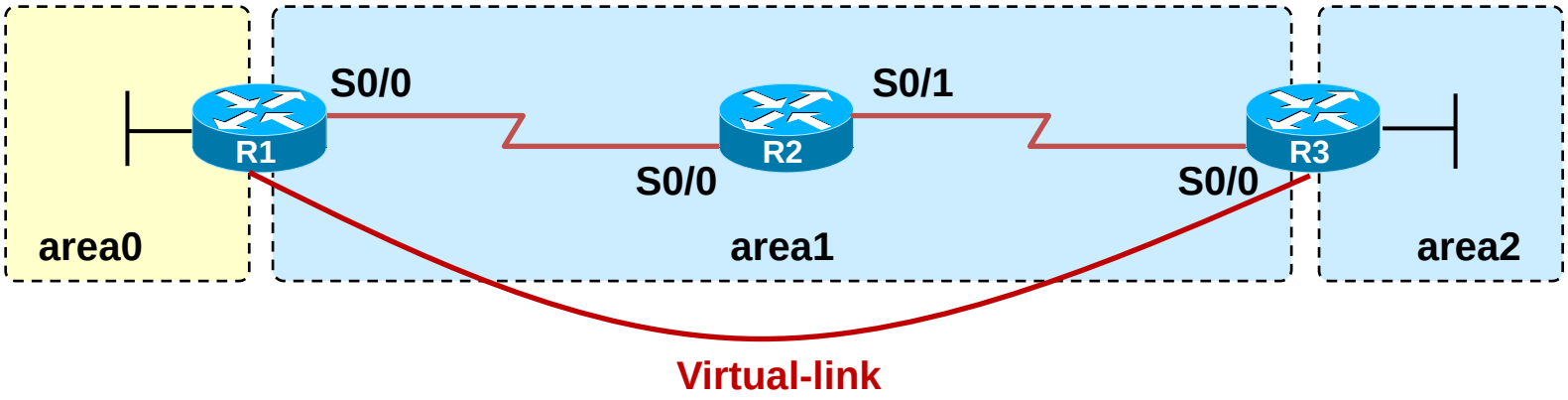
- **OSPF身份验证（密文）**




```
interface s0/0
  ip ospf message-digest–key 1 md5 spoto

router ospf 1
  router-id 1.1.1.1
  network 192.168.12.0 0.0.0.255 area 0
  area 0 authentication messae-digest
```

OSPF身份验证

- **在虚链路上配置OSPF简单密码身份验证**




```
Router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
 network 172.17.0.0 0.0.255.255 area 1
 area 0 authentication
!
 area 1 virtual-link 3.3.3.3 authentication-key SPOTO
```

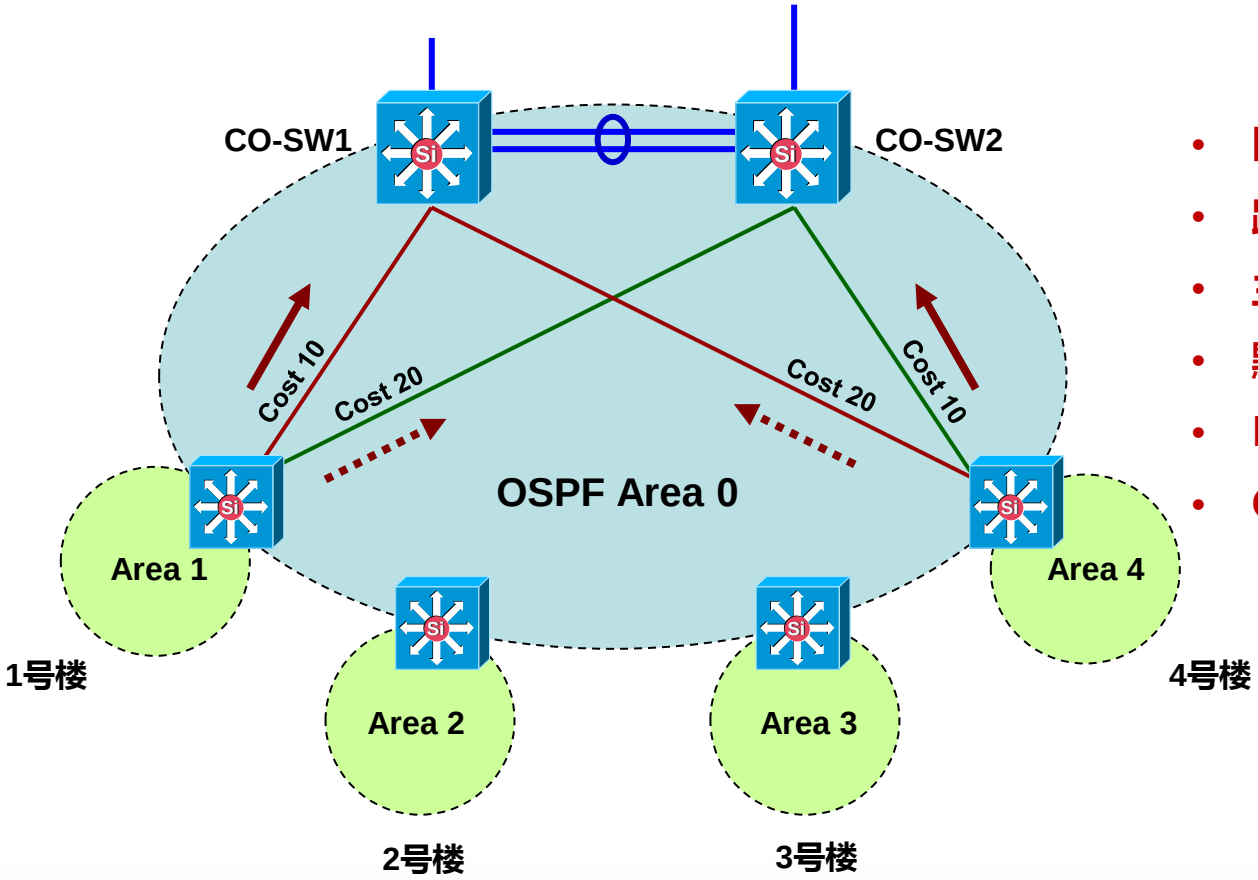
```
Router ospf 1
 network 172.18.0.0 0.0.255.255 area 0
 network 172.19.0.0 0.0.255.255 area 1
 area 0 authentication
!
 area 1 virtual-link 1.1.1.1 authentication-key SPOTO
```

OSPF身份验证

- **查看及验证**
  - show ip ospf neighbor
  - show ip ospf interface
  - debug ip ospf obj
  - debug ip ospf adj

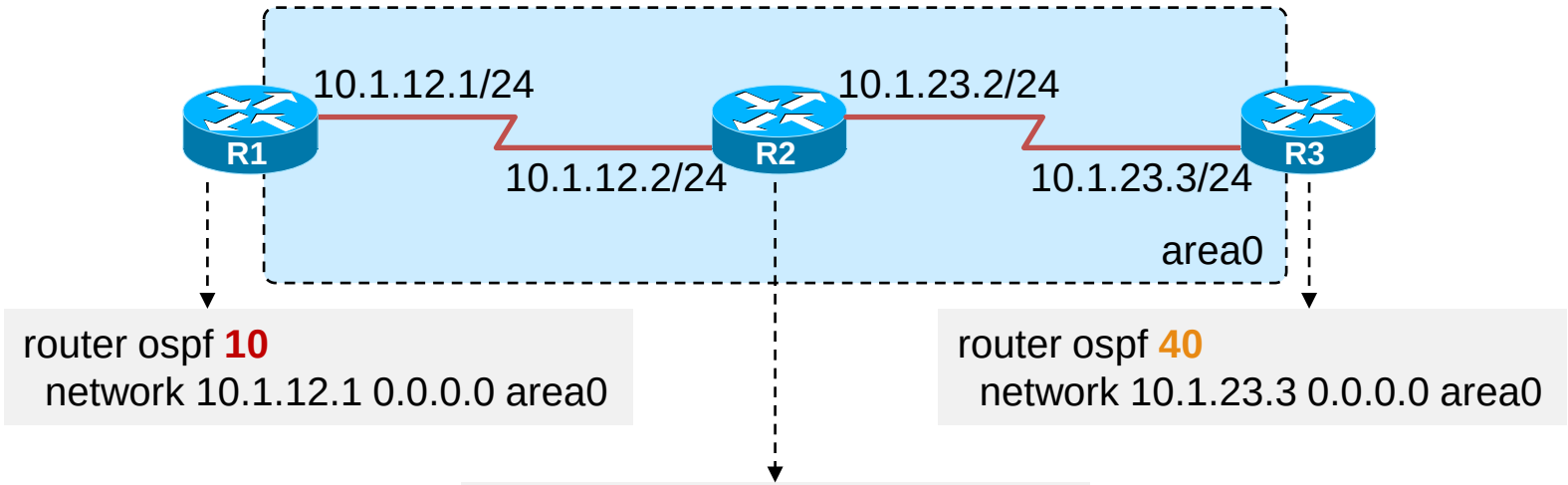
OSPF特殊区域

- OSPF在工程中的运用



- 区域划分（含特殊区域）
- 路由汇总
- 主备线路(COST)
- 默认路由传递
- Passive-interface
- OSPF认证

OSPF Process ID



红茶三杯
Vinsoney

沉淀 提升 成长 分享
关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

Redistribute Routing Protocols

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2012-08-01

课程目标

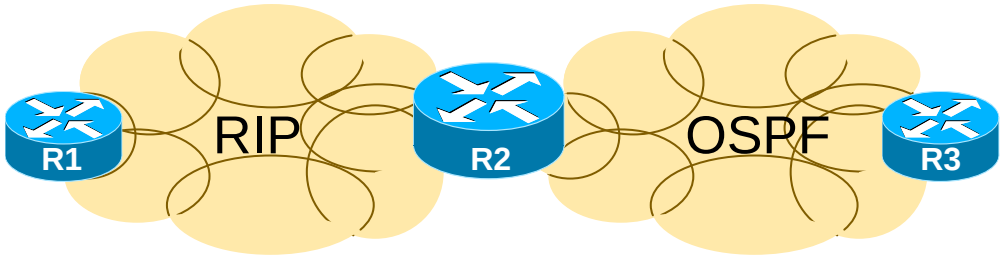
- 路由重发布的概念
- 路由重发布实施要点
- 路由重发布的实现

路由重发布的基本概念

路由重发布的概念

- **网络中使用多种IP路由协议**
  - 需要使用多种IP路由协议的原因
    - 多厂商的路由环境
    - 网络合并（同一协议或是不同协议）
    - 从旧的路由协议过渡到新的路由协议
    - 路由策略的需要（可靠性、冗余性、分流模型等）
  - 路由重分发（多个重分发点，双向重分发）

路由重发布的概念



R 192.168.1.0/24

R 192.168.2.0/24

O 192.168.3.0/24

O 192.168.4.0/24

R 192.168.1.0/24

R 192.168.2.0/24

O 192.168.3.0/24

O 192.168.4.0/24

Routing Table

路由重发布的概念

- **路由重发布**

  路由重分发是指连接到不同路由域（自治系统）的边界路由器在他们之间交换和通告路由选择信息的能力。

  – 从一种协议到另一种协议

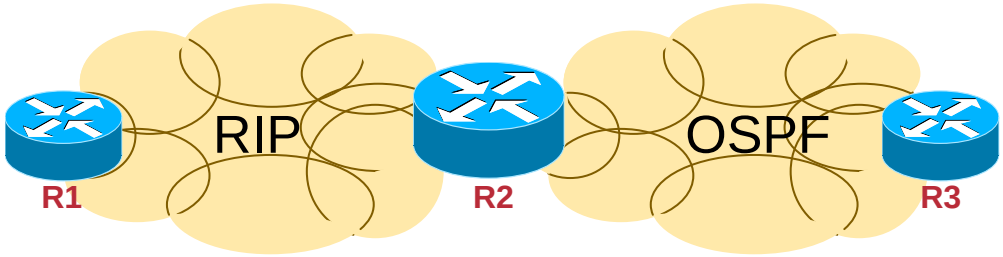
  – 同一种协议的多个实例

  注意

  – 重分发总是向外的，执行重分发的路由器不会修改其路由表

  – 路由必须要位于路由表中才能被重分发

路由重发布的概念



R 192.168.1.0/24

R 192.168.2.0/24

O 192.168.3.0/24

O 192.168.4.0/24

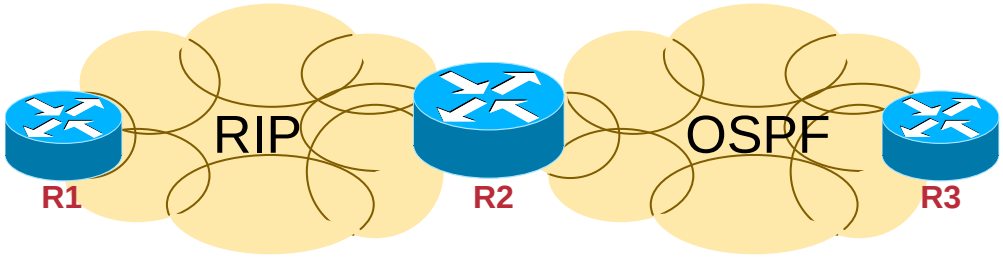
R 192.168.1.0/24
R 192.168.2.0/24

RIP进程

O 192.168.3.0/24
O 192.168.4.0/24

OSPF进程

路由重发布的概念

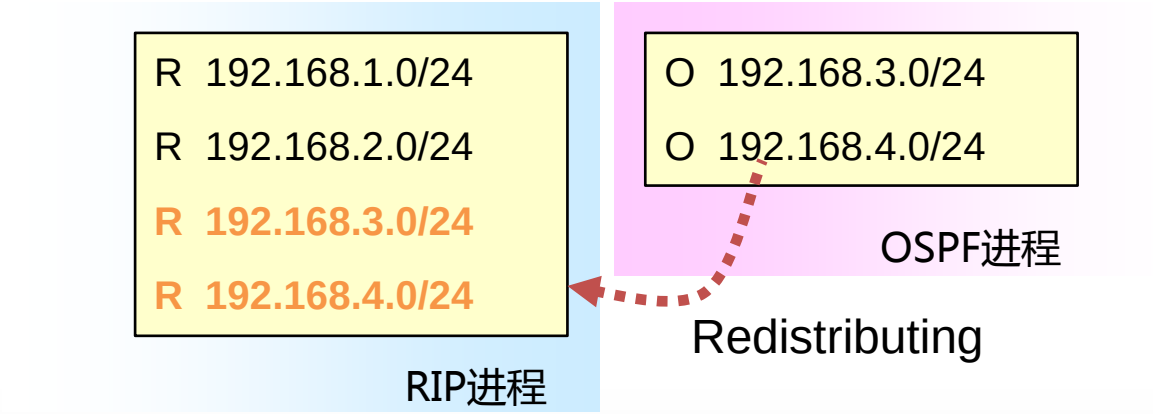


R 192.168.1.0/24

R 192.168.2.0/24

O 192.168.3.0/24

O 192.168.4.0/24

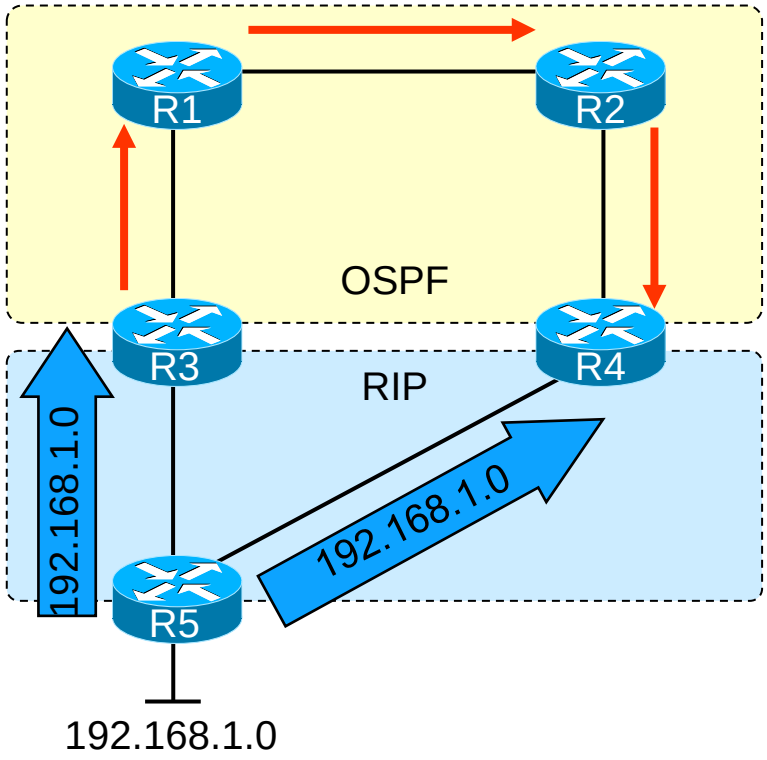


路由重发布实施要点

路由重发布的概念

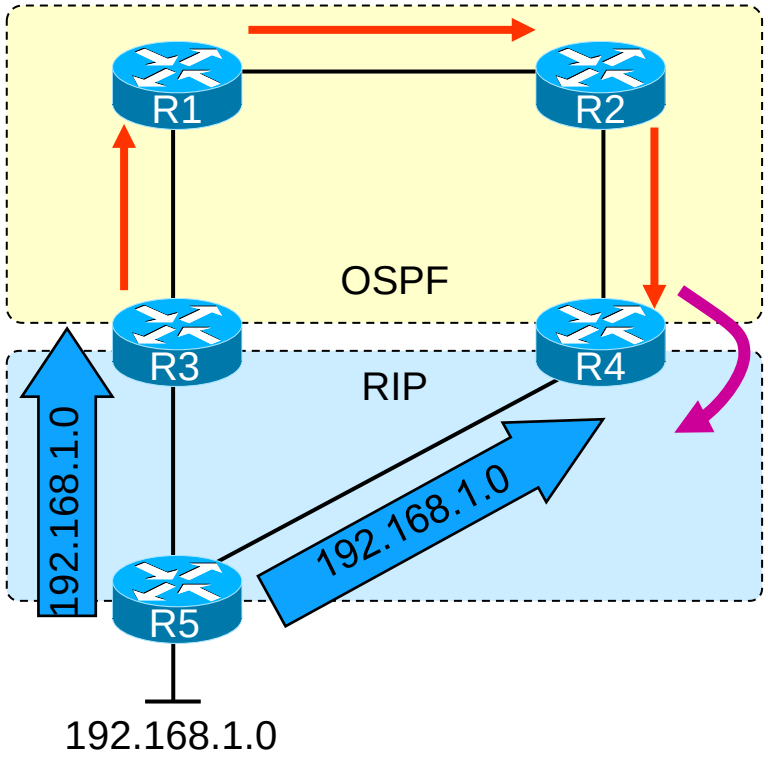
- **路由重发布**
  - 重分发需要考虑的因素
    - 路由回馈
    - 路由信息不兼容（度量值信息不一致）
    - 收敛时间不一致（不同路由协议的收敛速度不同）
  - 如何选择最佳路由
    - 管理距离
    - 度量值

路由feedback



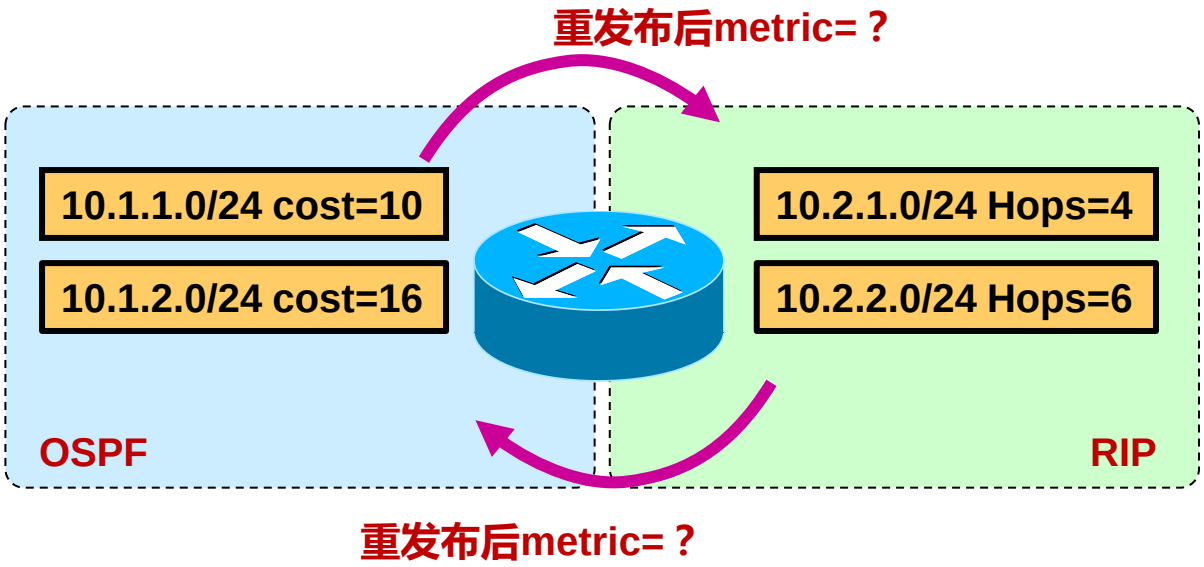
R4产生次优路径

路由feedback (cont.)



如果部署了双向重发布，则有可能会影响到RIP域中的路由选择

Metric的问题


重发布后metric= ?
10.1.1.0/24 cost=10
10.1.2.0/24 cost=16
10.2.1.0/24 Hops=4
10.2.2.0/24 Hops=6
OSPF
RIP
重发布后metric= ?

管理距离

路由来源	管理距离
直连接口	0
静态路由	1
EIGRP汇总路由	5
外部BGP	20
内部EIGRP	90
OSPF	110
IS-IS	115
RIPv1、RIPv2	120
外部EIGRP	170
内部BGP	200
未知	255

Metric的问题

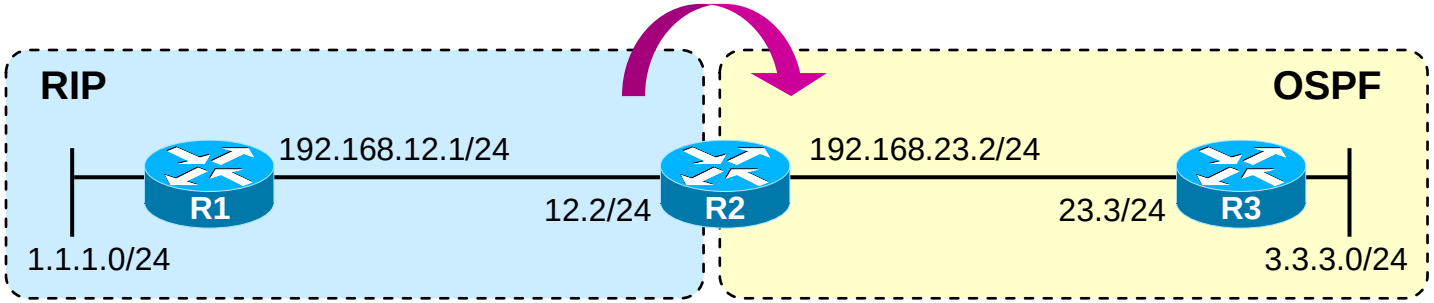
- **种子度量值**
  - 路由器通告与其接口直接相连的链路时，使用的初始度量值叫做种子度量值（也叫做默认度量值），是根据接口的特征得到的。
  - 种子度量值或默认度量值是在重分发配置期间定义的，并在自治系统内部正常递增，除了OSPF E2路由。
  - 可使用命令default-metric或是redistribute中使用metric来指定种子度量值

Metric的问题

- **种子度量值**

将路由重分发到该协议	默认种子度量值
RIP	0，视为无穷大
IGRP/EIGRP	0，视为无穷大
OSPF	BGP为1，其他路由为20，OSPF之间度量值保持不变
IS-IS	0
BGP	BGP度量值被设置为IGP度量值

Metric的问题

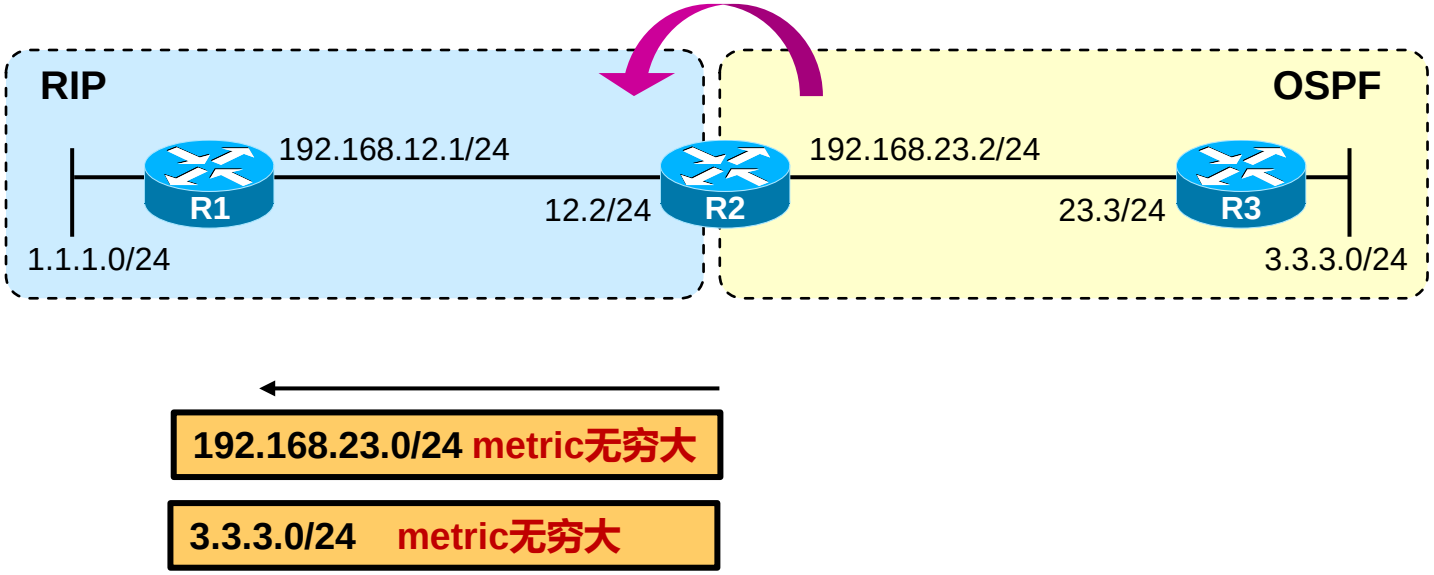


192.168.12.0/24 cost=20

1.1.1.0/24 cost=20

- default-metric 可以修改种子度量
- 或者在执行重发布时手工指定

Metric的问题



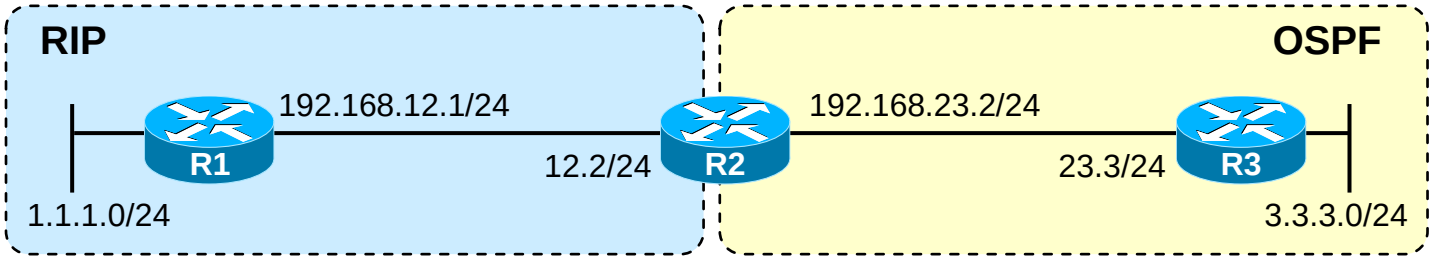
- default-metric 可以修改种子度量
- 或者在执行重发布时手工指定

路由重发布的实现

路由重发布的配置

```
RtrA(config)#router rip
RtrA(config-router)#redistribute ?
  bgp          Border Gateway Protocol (BGP)
  connected    Connected
  eigrp        Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis         ISO IS-IS
  iso-igrp     IGRP for OSI networks
  metric       Metric for redistributed routes
  mobile       Mobile routes
  odr          On Demand stub Routes
  ospf         Open Shortest Path First (OSPF)
  rip          Routing Information Protocol (RIP)
  route-map    Route map reference
  static       Static routes
```

实验1 OSPF到RIP的路由重发布



R3 routing table

C	192.168.12.0
C	1.1.1.0

R2 routing table

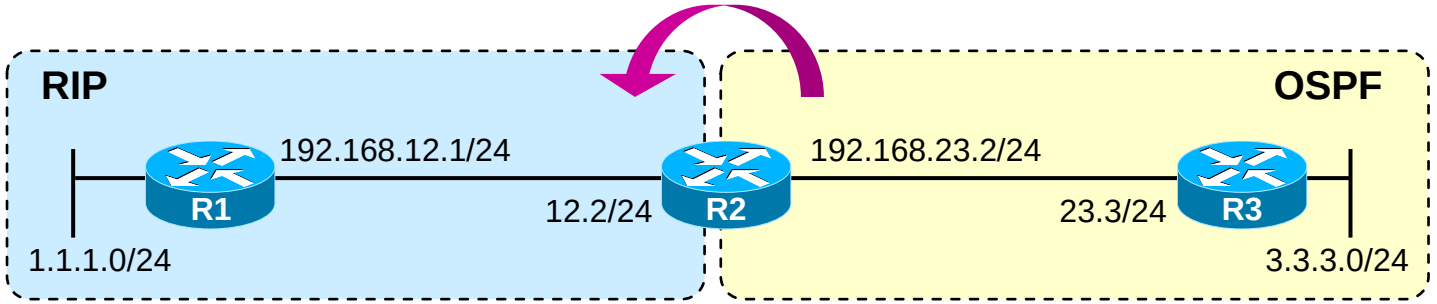
C	192.168.12.0
C	192.168.23.0
R	1.1.1.0
O	3.0.0.0

R3 routing table

C	192.168.23.0
C	3.3.3.0

```
R2(config)# router rip
R2(config-router)# redistribute ospf 1 metric 3
```

实验1 OSPF到RIP的路由重发布



R3 routing table

C	192.168.12.0
C	1.1.1.0
R	**192.168.23.0**
R	**3.3.3.0**

R2 routing table

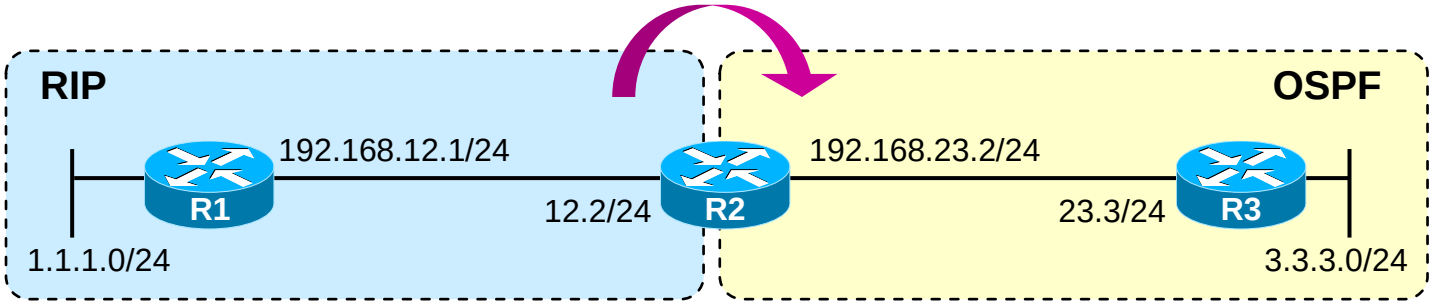
C	192.168.12.0
C	192.168.23.0
R	1.1.1.0
O	3.3.3.0

R3 routing table

C	192.168.23.0
C	3.3.3.0

```
R2(config)# router rip
R2(config-router)# redistribute ospf 1 metric 3
```

实验1 RIP到OSPF的路由重发布



R3 routing table

C	192.168.12.0
C	1.1.1.0
R	**192.168.23.0**
R	**3.3.3.0**

R2 routing table

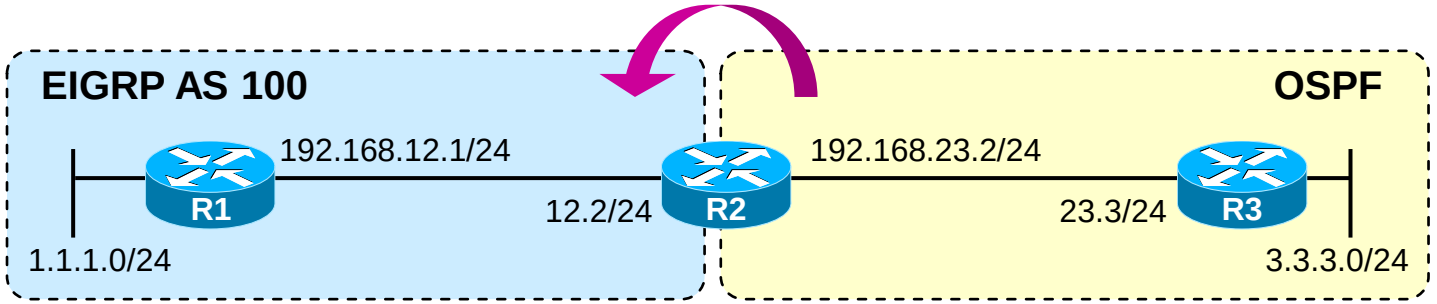
C	192.168.12.0
C	192.168.23.0
R	1.1.1.0
O	3.3.3.0

R3 routing table

C	192.168.23.0
C	3.3.3.0
OE	**192.168.12.0**
OE	**1.1.1.0**

```
R2(config)# router ospf 1
R2(config-router)# redistribute rip subnets
```

实验2 OSPF到EIGRP的重发布



R3 routing table

C 192.168.12.0
C 1.1.1.0
D EX 192.168.23.0
D EX 3.3.3.0

R2 routing table

C 192.168.12.0
C 192.168.23.0
D 1.1.1.0
O 3.3.3.0

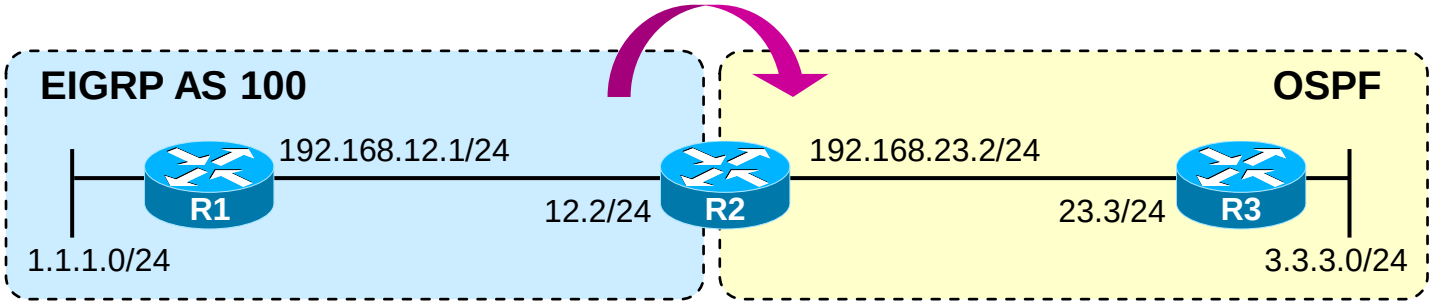
R3 routing table

C 192.168.23.0
C 3.3.3.0

R2(config)# router eigrp 100

R2(config-router)# redistribute os 1 **metric 100000 1000 255 1 1500**

实验2 EIGRP到OSPF的重发布



R3 routing table

C 192.168.12.0
C 1.1.1.0
D EX 192.168.23.0
D EX 3.3.3.0

R2 routing table

C 192.168.12.0
C 192.168.23.0
D 1.1.1.0
O 3.3.3.0

R3 routing table

C 192.168.23.0
C 3.3.3.0
OE 192.168.12.0
OE 1.1.1.0

R2(config)# router ospf 1

R2(config-router)# redistribute eigrp 100 **subnets**

红茶三杯
Vinsoney

学习 沉淀 成长 分享

关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

Routing Policy

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2012-08-01

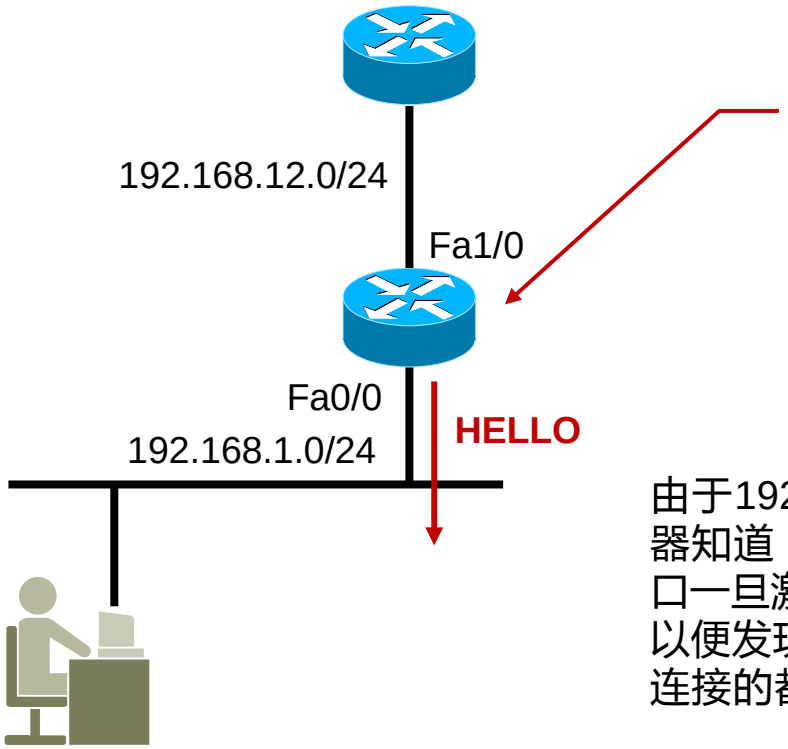
课程目标

- Passive-interface
- 控制管理距离
- Route-map
- Distribute-list
- Prefix-list

红茶三杯
Vinsoney

Passive-interface

背景




```
router ospf 1
  network 192.168.1.0 0.0.0.255 area0
  network 192.168.12.0 0.0.0.255 area0
```

由于192.168.1.0/24网段需要被OSPF域的路由器知道，因此被宣告进了OSPF，然而当Fa0/0口一旦激活OSPF，该接口就会尝试发送Hello包以便发现链路上的OSPF邻居，但是，该链路上连接的都是主机，这些Hello包实际上是多余的。

Passive-interface

- RIP/IGRP——在指定接口不向外发送路由更新，但是接收路由更新
- EIGRP——在指定接口不向外发送Hello消息，而且通过这个接口不与其他路由器建立邻接关系，不发送其他EIGRP的数据流
- OSPF——在指定接口不向外发送Hello消息，而且通过这个接口不与其他路由器建立邻接关系，不发送和接收路由信息。（有些IOS版本中，OSPF在被动接口上发送Hello和DBD分组，但是不发送LSU。）

Passive-interface的配置

将某个接口配置为被动接口：

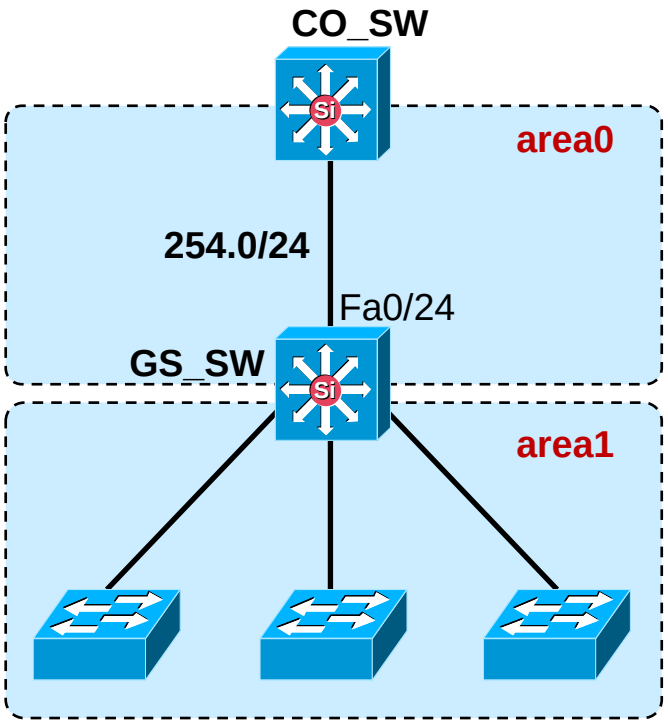
Router(config-router)# **passive-interface** int-*type int-num*

将所有接口配置为被动接口，并手动激活特定接口

Router(config-router)# **passive-interface** default

Router(config-router)# **no passive-interface** int-*type int-num*

Passive-interface应用示例

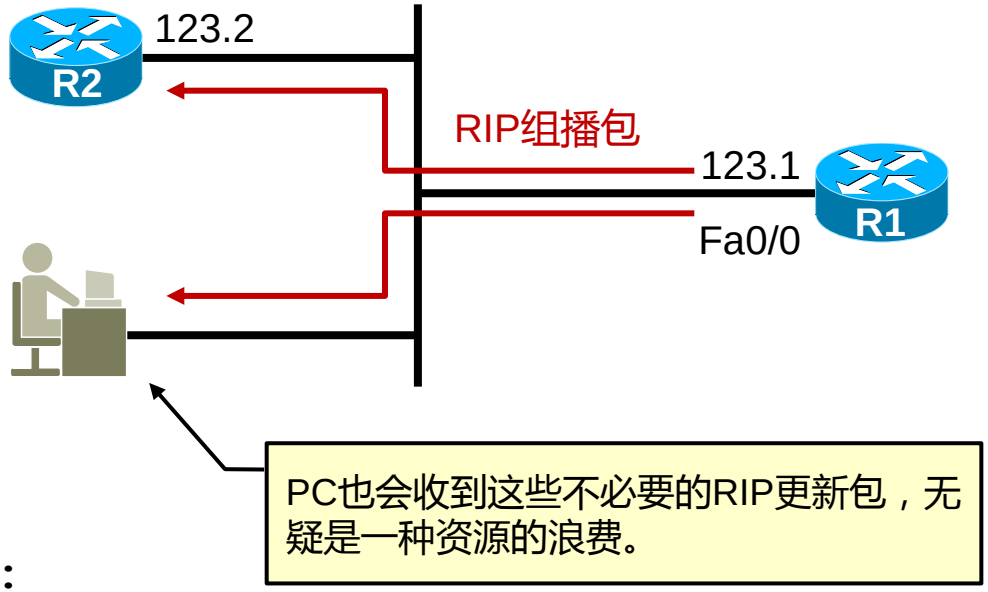



```
interface vlan 10
  ip address 192.168.10.254 255.255.255.0
interface vlan 20
  ip address 192.168.20.254 255.255.255.0

router ospf 1
  network 192.168.10.0 0.0.0.255 area 1
  network 192.168.20.0 0.0.0.255 area 1
  network 192.168.254.0 0.0.0.255 area 1
  passive-interface defualt
  no passive-interface fast 0/24
```

汇聚交换机上所有的三层接口都Network进相应的Area

汇聚交换机将向所有VLAN接口发送HELLO报文，尝试建立邻居关系，而底层的用户也会收到其并不需要的HELLO包。

单播更新（RIP环境）



配置RIP单播更新：

Router(config) **router rip**

Router(config-router)# **passive-interface** fast 0/0

Router(config-router)# **neighbor** 192.168.123.2

单播更新（EIGRP环境）

- 如果是EIGRP环境，需实现单播更新，那么路由更新接口不能被PASSIVE（这与RIP不一样），而是直接使用neighbor命令去指定邻居即可。
- 如果接口一旦被PASSIVE，则即使手工指定了neighbor，也是无法正常建立EIGRP邻居关系。

控制管理距离

常见路由协议管理距离

Routing Protocols	AD
直连接口	0
关联出接口的静态路由	1
关联下一跳的静态路由	1
EIGRP 汇总路由	5
外部 BGP	20
内部EIGRP	90
IGRP	100
OSPF	110
RIPv1、v2	120
外部EIGRP	170
内部BGP	200

调整路由协议的管理距离

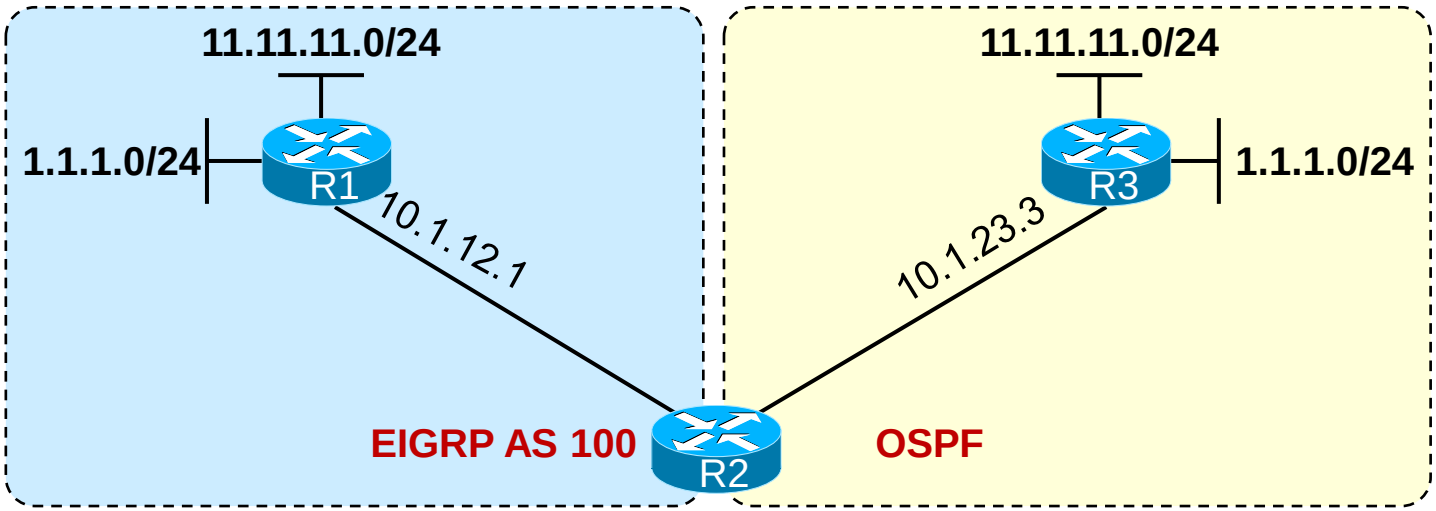
修改OSPF的AD值

```
Router(config)# router ospf 1
Router(config-router)# distance AD ip-src wildmask acls
Router(config-router)# distance ospf external ad1 inter-area ad2 intra-area ad3
```

修改EIGRP的AD值

```
Router(config)# router eigrp 100
Router(config-router)# distance AD ip-src wildmask acls
Router(config-router)# distance eigrp internal-distance external-distance
```

调整路由协议的管理距离 示例



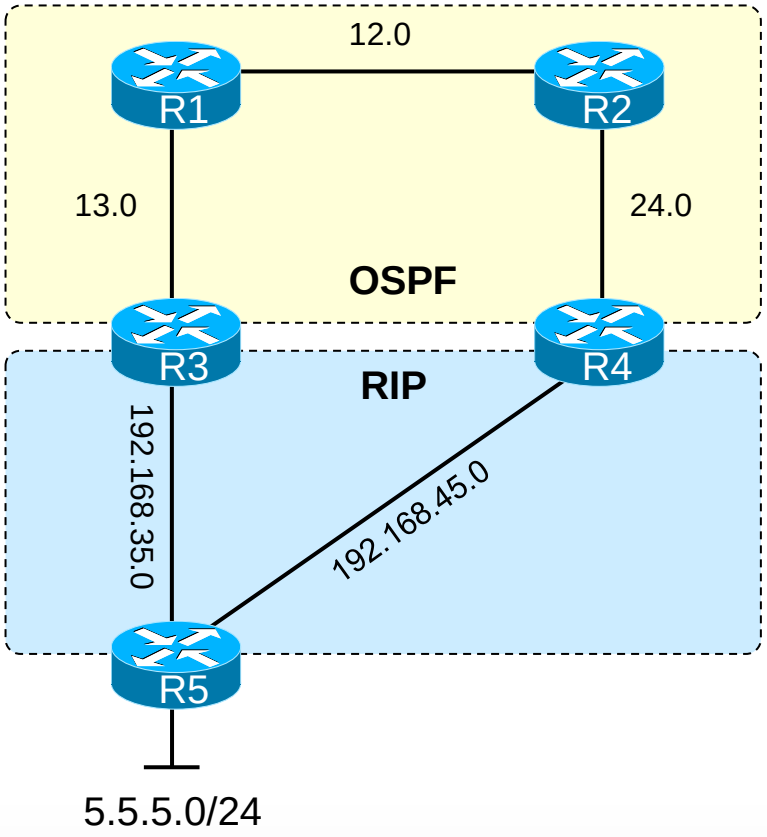
实现需求（R2的路由表）：

```
O 11.11.11.0
D 1.1.1.0
```

实现方式（R2的配置）：

```
access-list 1 per 11.11.11.0
!
router eigrp 100
distance 130  10.1.12.1 0.0.0.0 1
```

通过调整管理距离解决双点双向重分发存在的问题



通过调整管理距离解决双点双向重分发存在的问题 cont.

Hostname R3

router ospf 1

redistribute rip metric 100 metric-type 1 subnets

network 192.168.12.0 0.0.0.255 area 0

network 192.168.13.0 0.0.0.255 area 0

router rip

version 2

redistribute ospf 1 metric 5

network 192.168.35.0

no auto-summary

Hostname R4

router ospf 1

redistribute rip metric 100 metric-type 1 subnets

network 192.168.12.0 0.0.0.255 area 0

network 192.168.24.0 0.0.0.255 area 0

router rip

version 2

redistribute ospf 1 metric 5

network 192.168.45.0

no auto-summary

通过调整管理距离解决双点双向重分发存在的问题 cont.

Hostname R3

router ospf 1

distance 125 0.0.0.0 255.255.255.255 10

access-list 10 permit 192.168.35.0

access-list 10 permit 192.168.45.0

access-list 10 permit 5.5.5.0

Hostname R4

router ospf 1

distance 125 0.0.0.0 255.255.255.255 10

access-list 10 permit 192.168.35.0

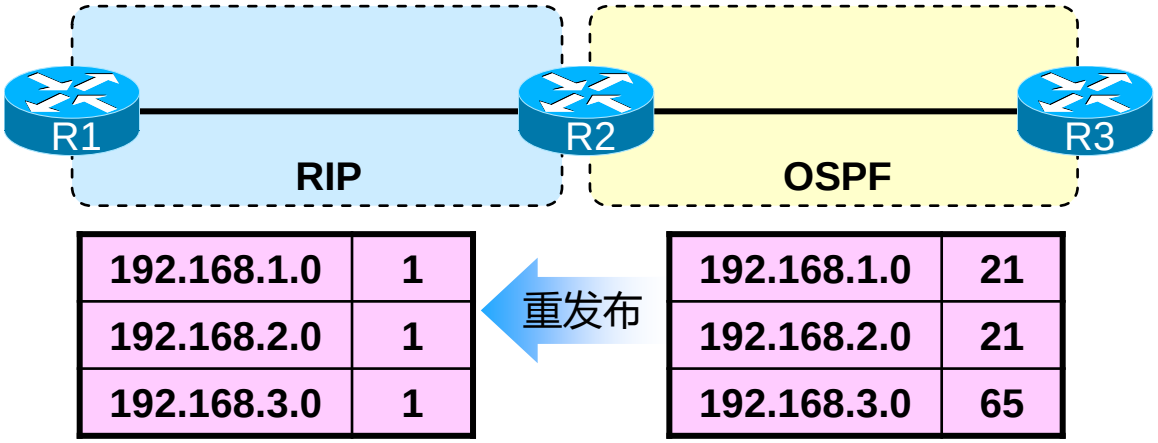
access-list 10 permit 192.168.45.0

access-list 10 permit 5.5.5.0

Route-map

小背景

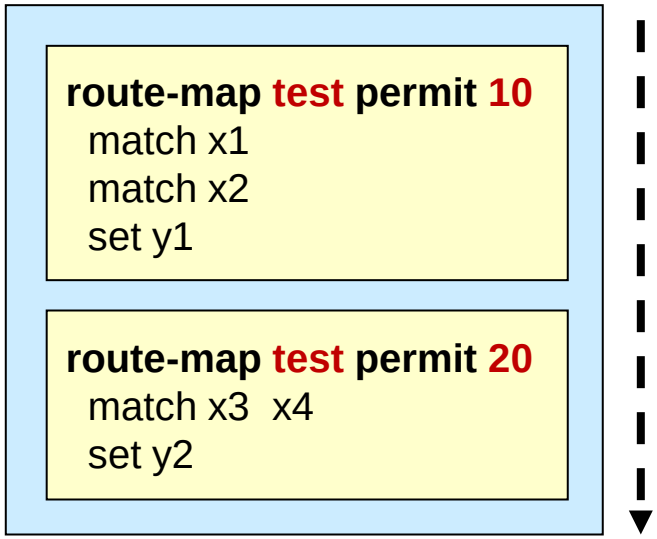
redistribute ospf 1 metric 1



Route-map的用途

- 重分发期间进行路由过滤或执行策略
- PBR（策略路由）
- NAT（网络地址转换）
- BGP中的策略部署
- 其他用途

Route-map初相识



逐级向下比对
匹配每一个match语句
如果匹配，则执行SET动作，
否则进入下一个条目

Route-map的特点

- 使用match命令匹配特定的分组或路由，set修改该分组或路由相关属性。
- Route-map中的每个序列号语句相当于于访问控制列表中的各行。
- Route-map默认为permit，默认序列号为10，序列号不会自动递增，需要指定序列号
- 末尾隐含deny any
- 单条match语句包括多个条件时，使用逻辑or运算；多条match语句时，使用逻辑and运算。

Route-map的配置

match ip address 匹配访问列表或前缀列表

match length 根据分组的第三层长度进行匹配

match interface 匹配下一跳出接口为指定接口之一的路由

match ip next-hop 匹配下一跳地址为特定访问列表中被允许的那些路由

match metric 匹配具有指定度量值的路由

match route-type 匹配指定类型的路由

match community 匹配BGP共同体

match tag 根据路由的标记进行匹配

Route-map的配置（cont.）

set metric 设置路由协议的度量值

set metric-type 设置目标路由协议的度量值类型

set default interface 指定如何发送这样的分组

set interface 指定如何发送这样的分组

set ip default next-hop指定转发的下一跳

set ip next-hop 指定转发的下一跳

set next-hop 指定下一跳的地址，指定BGP的下一跳

set as-path

set community

set local-preference

set weight

set origin

set tag

default 关键字优先级低于明细路由

Route-map的配置（cont.）

route-map

- 这个全局配置命令创建一个route-map，使用自定义的字符串来表示这个route-map，你可以在一个route-map下定义多个序列号。序列号在进行匹配动作时具有优先顺序。
- Permit/deny关键字在不同的部署场合中作用有所不同

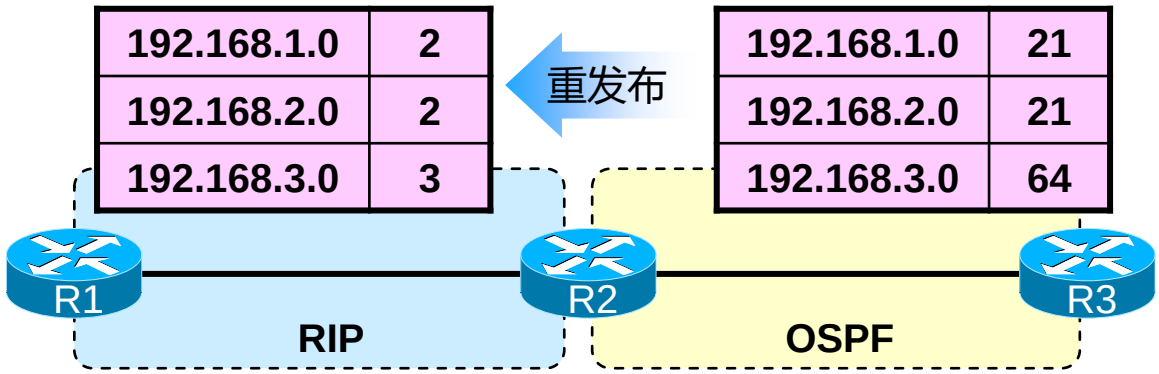
```
route-map test permit/deny 10
  match x1
  match x2
  set Y
route-map test permit/deny 20
  match x3,x4
  set Y
```

逻辑与

逻辑或

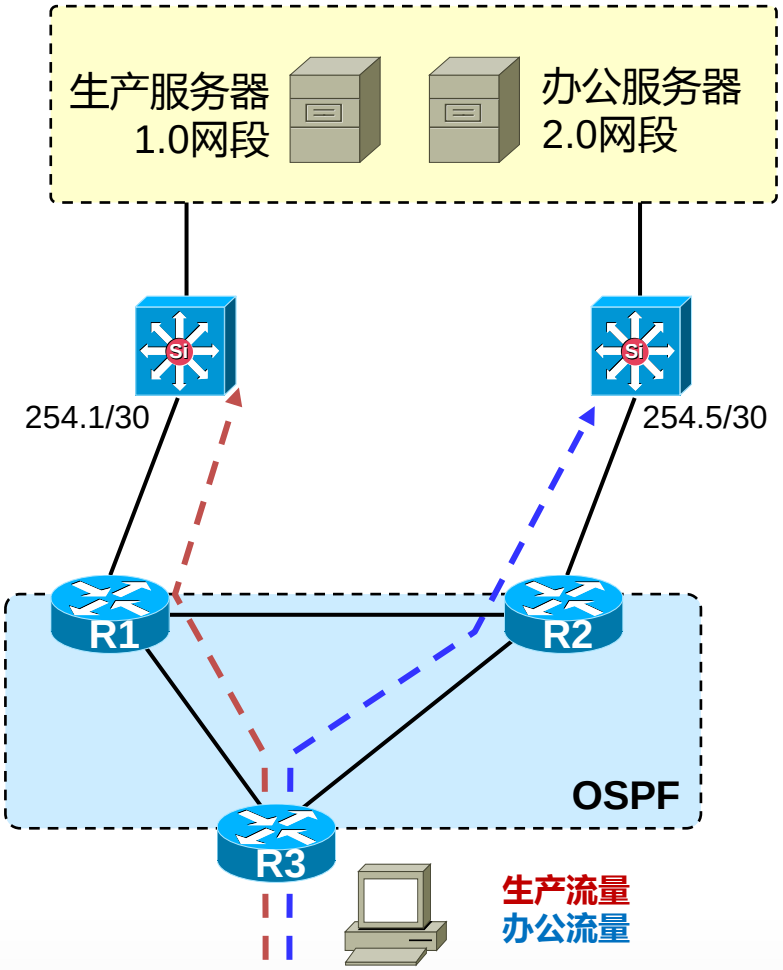
- 如果没有match语句，表示match any
- 每个route-map最后都会隐藏一个类似deny any的语句

Route-map配置示例1




```
access-list 1 permit 192.168.1.0
access-list 1 permit 192.168.2.0
access-list 2 permit 192.168.3.0
route-map test permit 10
  match ip address 1
  set metric 2
route-map test permit 20
match ip address 2
  set metric 3
router rip
  redistribute ospf 1 route-map test
```

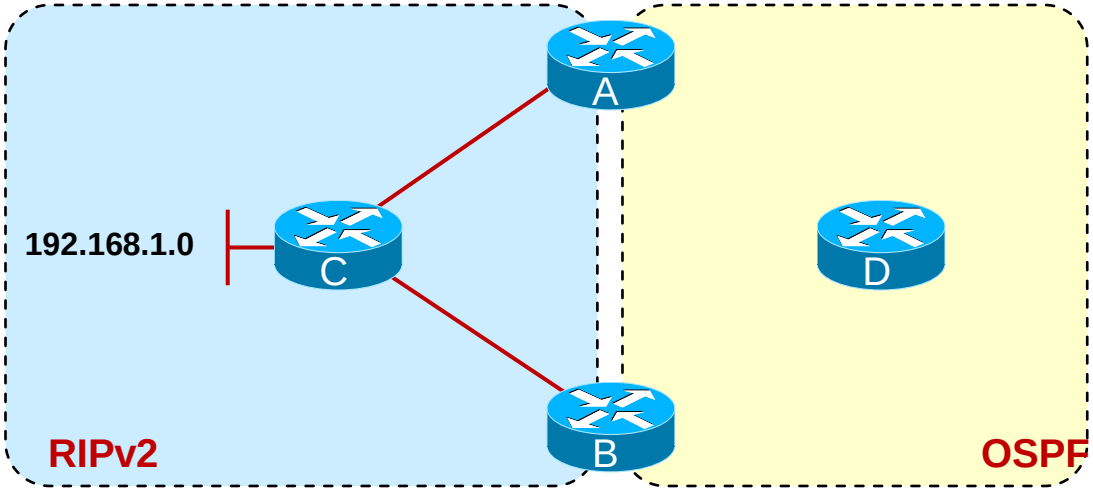
Route-map配置示例2



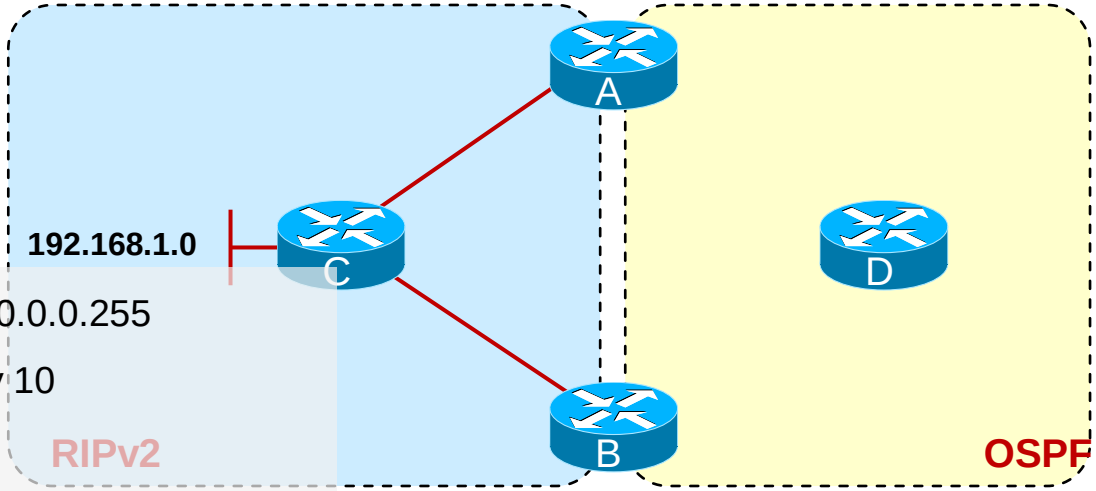
R1的配置如下：

```
Ip route 10.1.1.0 255.255.255.0 10.1.254.1
Ip route 10.1.2.0 255.255.255.0 10.1.254.1

access-list 1 permit 10.1.1.0
access-list 2 permit 10.1.2.0

route-map cisco permit 10
    match ip address 1
    set metric 10
route-map cisco permit 20
    match ip address 2
    set metric 20

router ospf 100
  redis static route-map cisco
```

Route-map配置示例3



Route-map配置示例3 (cont.)



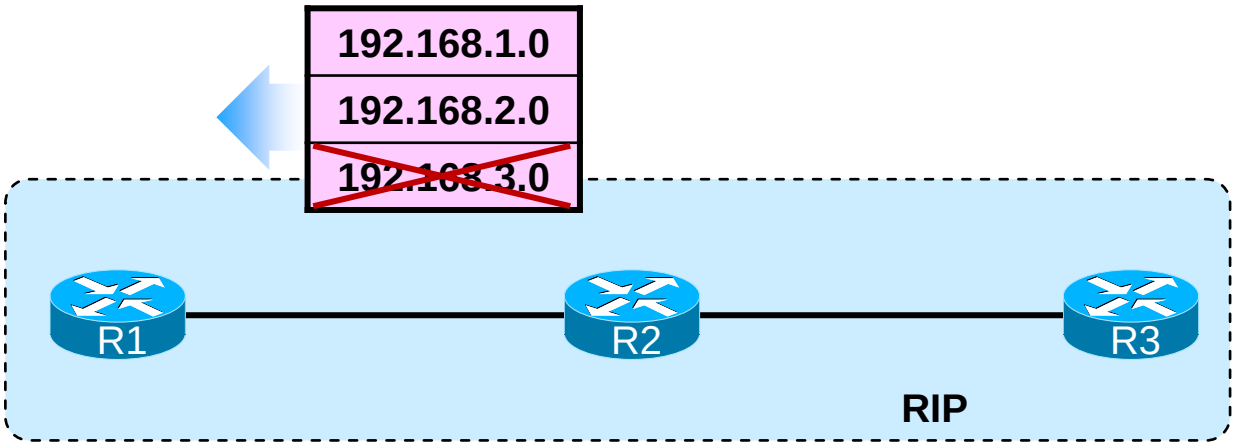

```
access-list 1 permit 192.168.1.0 0.0.0.255
route-map OSPF_into_RIP deny 10
  match ip address 1
route-map OSPF_into_RIP permit 20
router rip
  redistribute ospf 10 route-map OSPF_into_RIP
router ospf 10
  redistribute rip subnets
```

红茶三杯
Vinsoney

Distribute-list

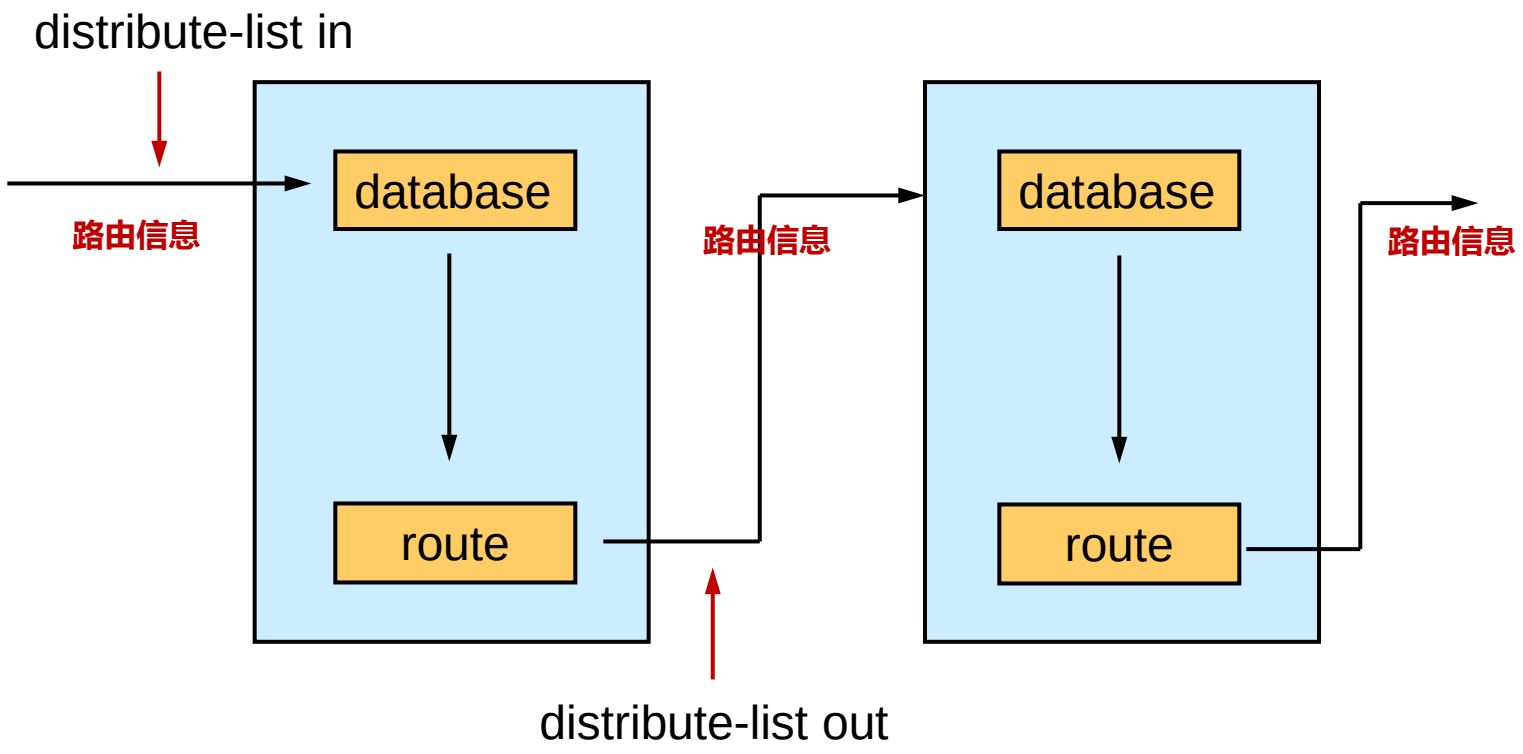
distribute-list

- 用于控制路由更新的一个工具
- 只能过滤路由信息，无法过滤LSA



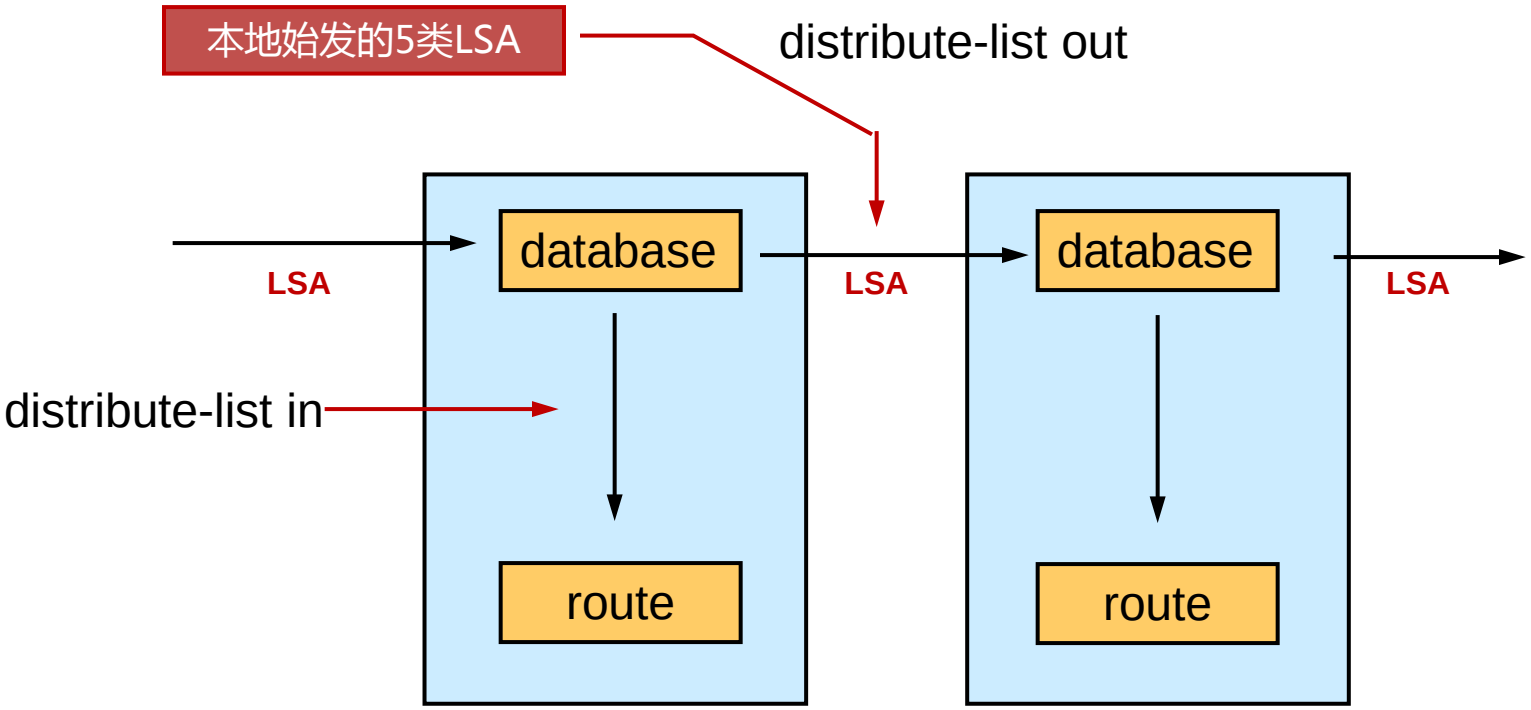
distribute-list

- **对于距离矢量路由协议**



distribute-list

- **对于链路状态路由协议**



distribute-list的配置

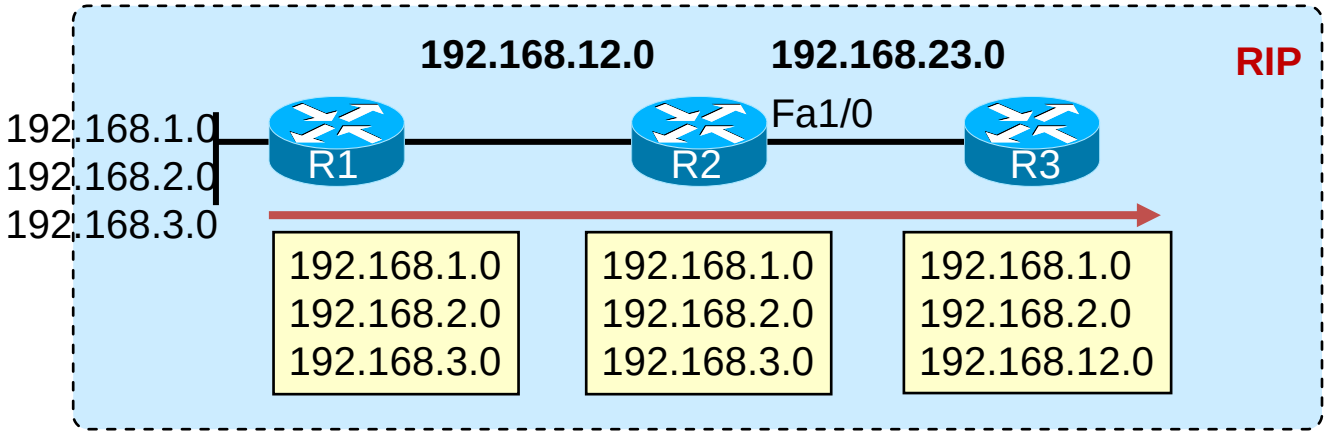
Out方向的分发列表

Router(config-router)# **distribute-list** *{access–list-number | name}* **out** *[interface–name | routing–process [routing-process parameter]]*

in方向的分发列表

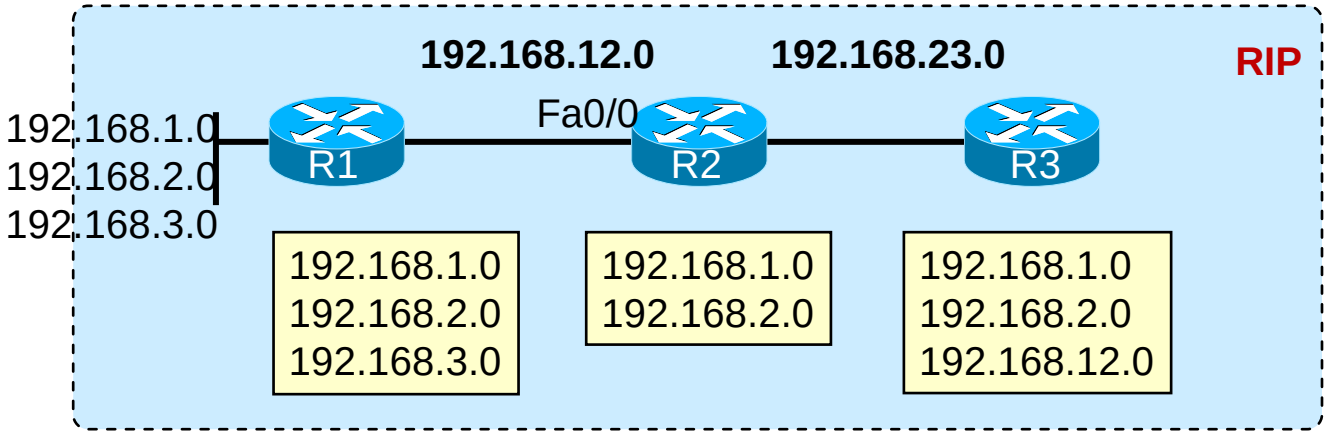
Router(config-router)# **distribute-list** *[access–list-number | name] | [route-map map-tag]* **in** *[interface-type interface-number]]*

配置示例1（单一路由协议环境下-RIP）




```
R2(config)# access-list 1 deny 192.168.3.0
R2(config)# access-list 1 permit any
R2(config)# router rip
R2(config-router)# distribute-list 1 out fa 1/0
```

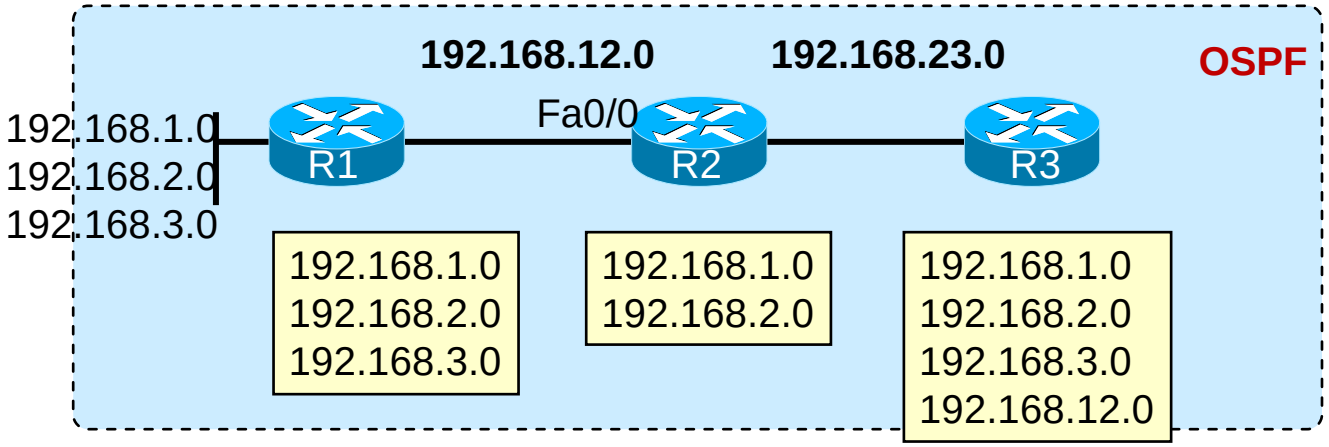
配置示例2（单一路由协议环境下-RIP）



R2如下配置，则R2、R3的路由表将如何？

```
R2(config)# access-list 1 deny 192.168.3.0
R2(config)# access-list 1 permit any
R2(config)# router rip
R2(config-router)# distribute-list 1 in fa0/0
```

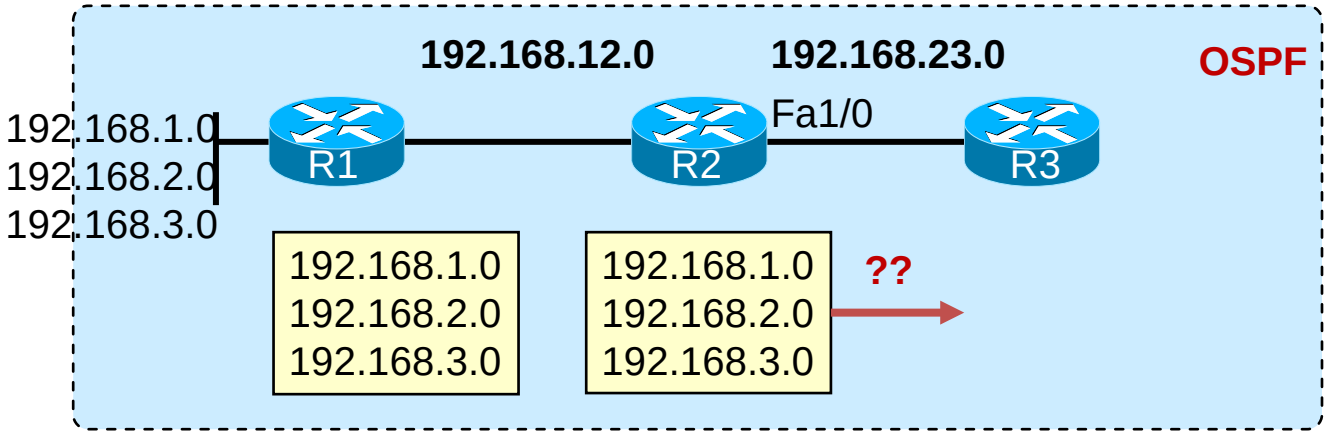
配置示例3（单一路由协议环境下-OSPF）



协议更换为OSPF。R2如下配置，则R2自己、R3的路由表将如何？

```
R2(config)# access-list 1 deny 192.168.3.0
R2(config)# access-list 1 permit any
R2(config)# router ospf 1
R2(config-router)# distribute-list 1 in fa0/0
```

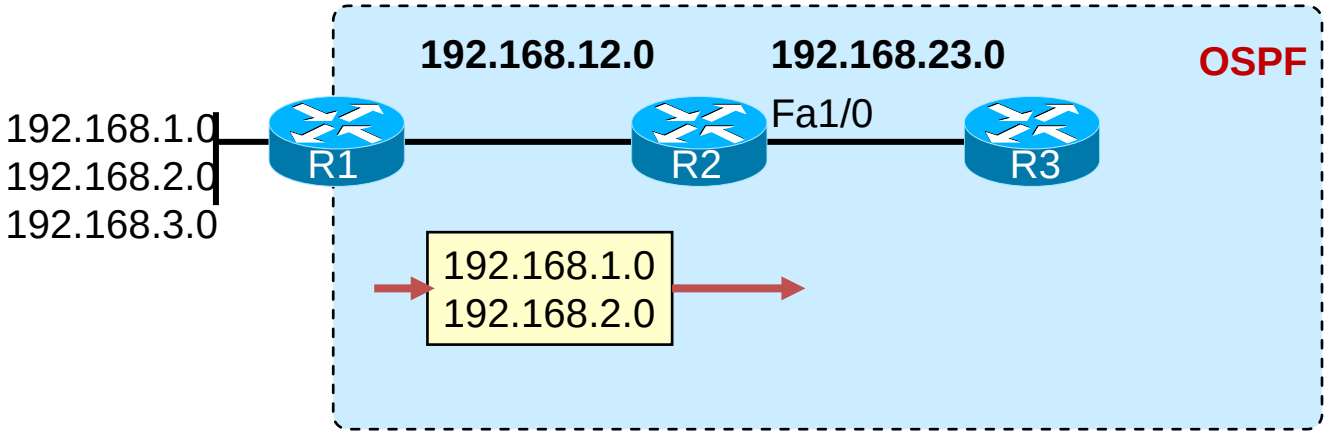
配置示例4（单一路由协议环境下-OSPF）



R2如下配置，则R3的路由表将如何？

```
R2(config)# access-list 1 deny 192.168.3.0
R2(config)# access-list 1 permit any
R2(config)# router ospf 1
R2(config-router)# distribute-list 1
```

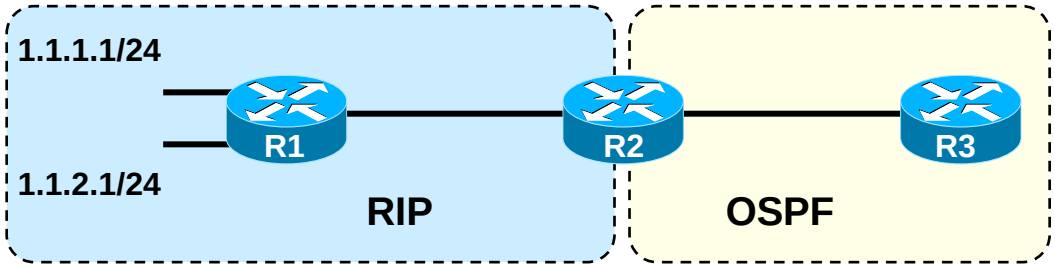
配置示例5（单一路由协议环境下-OSPF out方向分发列表）



在OSPF环境下，分发列表只能用于过滤本地始发的外部路由（因此只能在R1实施）

```
R1(config)# access-list 1 deny 192.168.3.0
R1(config)# access-list 1 permit any
R1(config)# router ospf 1
R1(config-router)# redistribute connected subnets
R1(config-router)# network 192.168.12.1 0.0.0.0 area 0
R1(config-router)# distribute-list 1 out
```

配置示例6 重发布时部署分发列表

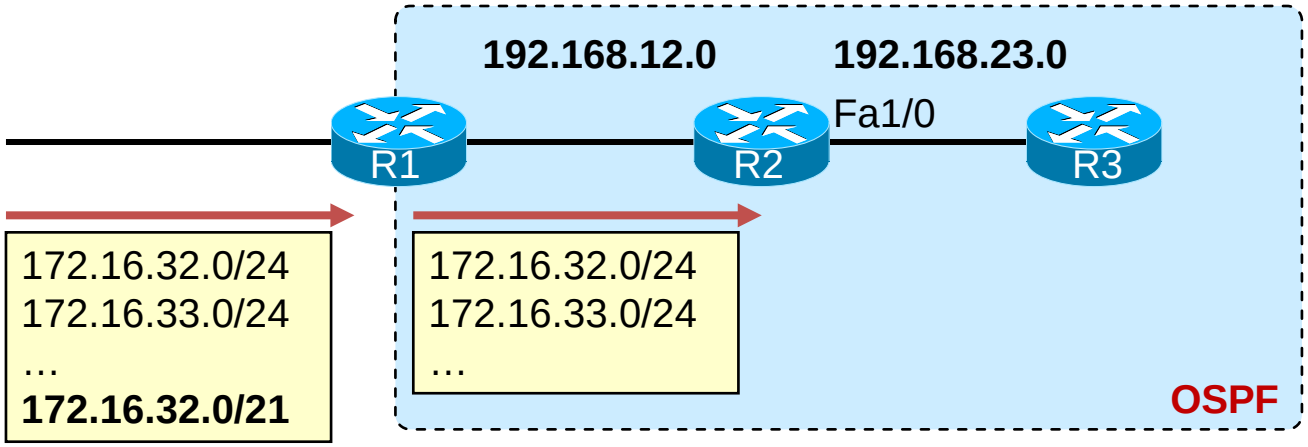


在R2上开设loopback接口2.2.2.0/24，R2既重发布RIP进OSPF，又重发布直连进OSPF，R2的配置如下：

```
access-list 1 permit 1.1.1.0
router ospf 1
  redistribute connected subnets
  redistribute rip metric 10 subnets
  distribute-list 1 out rip          // 注意加RIP与不加的区别
```

Prefix-list

prefix-list应用背景



外部路由172.16.32.0 – 39.0/24，以及汇总路由32.0/21被R1引入OSPF

现在需在注入过程中，仅将汇总路由32.0/21过滤，所有明细放行，使用标准ACL匹配路由，该如何写？

```
R1(config)# access-list 1 deny 172.16.32.0          //如果加上反掩码呢？
R1(config)# access-list 1 permit any
```

prefix-list应用背景

- **如何用最精简、最精确的标准ACL匹配下列路由？**

192.168.8.0/24
192.168.9.0/24
192.168.10.0/24
192.168.11.0/24

access-list 1 permit

192.168.8.0 0.0.3.0

缺陷：无法匹配掩码

prefix-list应用背景

- **如何使用扩展ACL匹配路由及掩码？**

192.168.8.0/24
192.168.9.0/24
192.168.10.0/24
192.168.11.0/24

access-list 100 permit

192.168.8.0 0.0.3.0 255.255.255.0 0.0.0.0

prefix-list前缀列表

- 前缀列表的可控性比访问列表高得多，支持增量修改，更为灵活
- 可匹配路由前缀中的网络号及前缀长度，增强了匹配的精确度
- 前缀列表包含序列号，从最小的开始匹配
- 如果前缀不与前缀列表中的任何条目匹配，将被拒绝

prefix-list的配置

router(config)# **ip prefix-list** {list-name [**seq** number] {**deny | permit**} *network/length* [**ge** *ge-value*] [**le** *le-value*]

参数	描述
ge *ge-value*	要匹配的前缀范围，范围为ge-value到32
le *le-value*	要匹配的前缀范围，范围为length到le-value

输入条件：length < ge-value < le-value <= 32

prefix-list配置示例

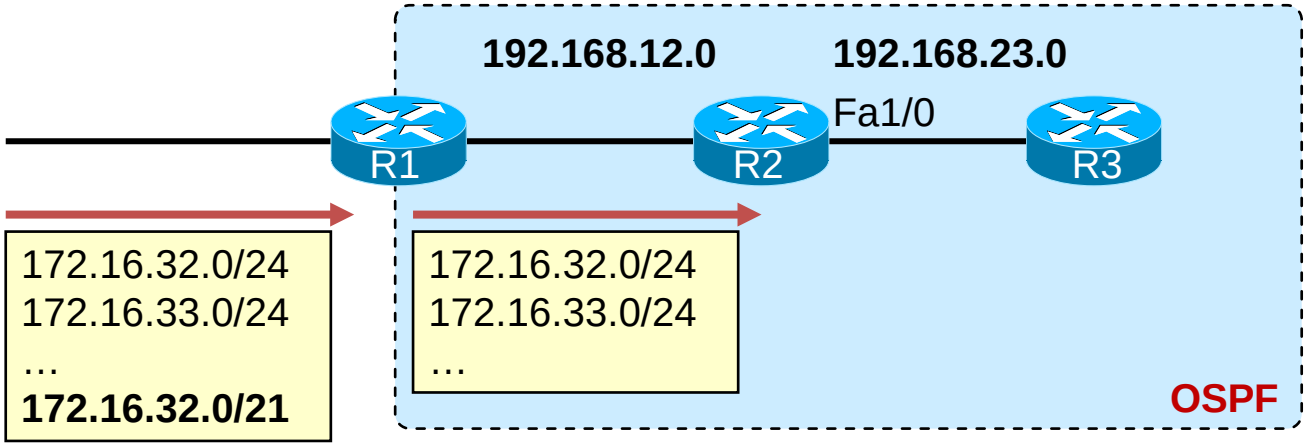
- **匹配某条特定路由：192.168.1.0/24**
  - ip prefix-list pxlist 192.168.1.0/24
- **匹配默认路由**
  - ip prefix-list pxlist permit 0.0.0.0/0
- **匹配所有主机路由**
  - ip prefix-list pxlist permit 0.0.0.0/0 ge 32
- **匹配所有路由(any)**
  - ip prefix-list list1 permit 0.0.0.0/0 le 32

prefix-list配置示例 cont.

- **匹配以下路由（用最精确最简洁的方式）：**
  - 192.168.4.0/24
  - 192.168.5.0/24
  - 192.168.6.0/24
  - 192.168.7.0/24

```
ip prefix-list test permit 192.168.4.0/22 ge 24 le 24
```

prefix-list配置示例 cont.



外部路由172.16.32.0 – 39.0/24，以及汇总路由32.0/21被R1引入OSPF

现在需在注入过程中，仅将汇总路由32.0/21过滤，所有明细放行。

```
R1(config)# ip prefix-list list1 deny 172.16.32.0/21
R1(config)# ip prefix-list list1 permit 0.0.0.0/0 le 32
R1(config)# route-map test permit 10
R1(config-route-map)# match ip address prefix-list list1
```

红茶三杯
Vinsoney

学习 沉淀 成长 分享

关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

Path Control

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2012-08-01

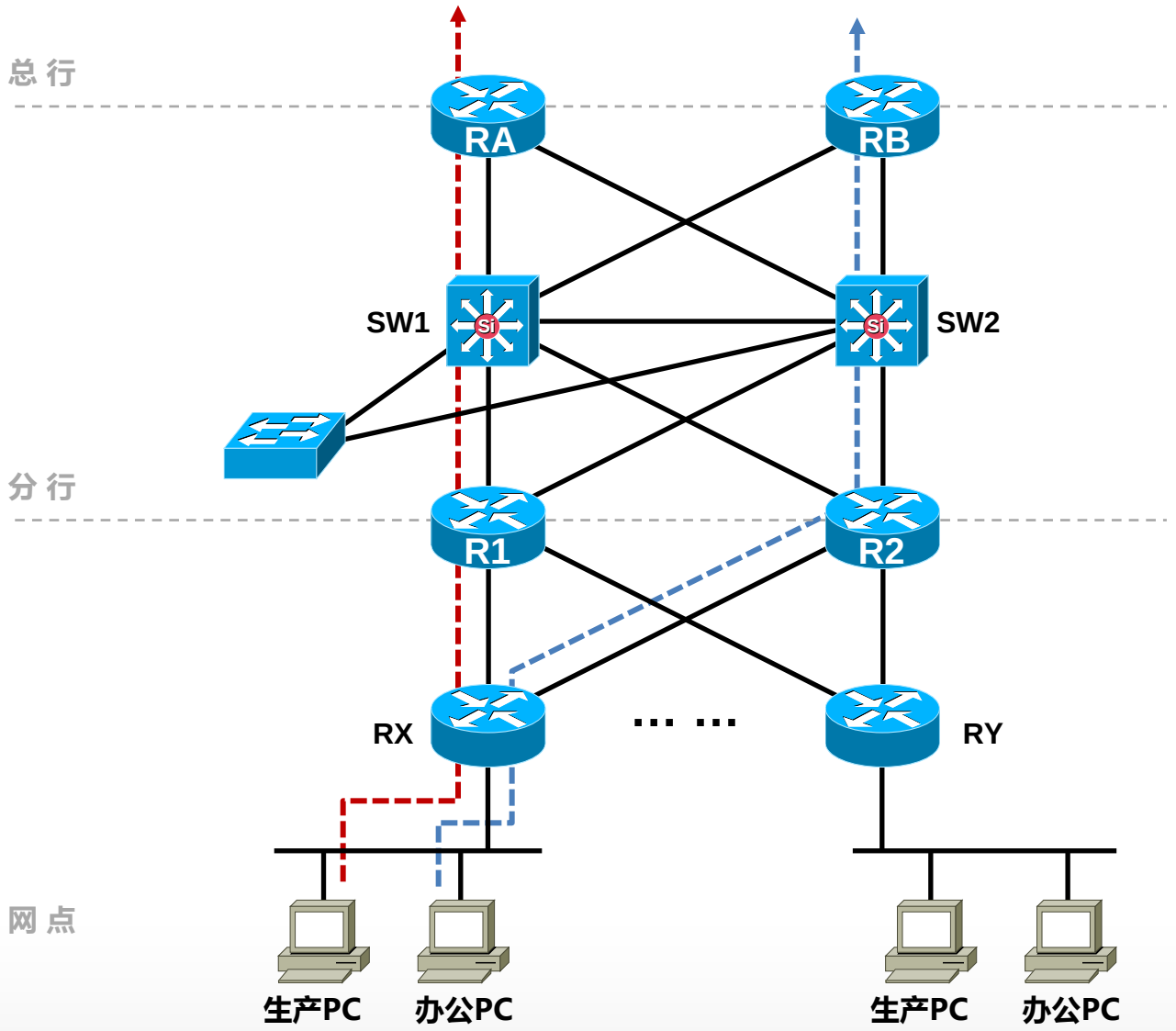
课程目标

- 路径控制概述
- Offset-list
- Policy-based routing
- 综合实验

红茶三杯
Vinsoney

路径控制概述

技术背景



路径控制工具

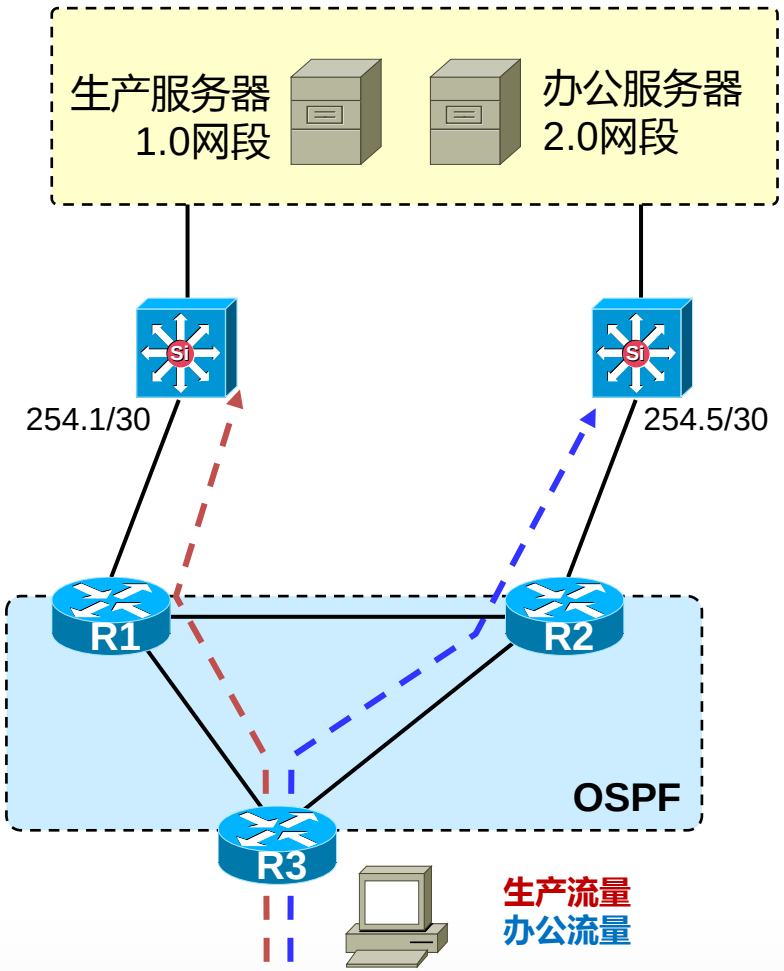
- 妥善的编址方案：VLSM和CIDR
- 重分发和路由协议的特征
- passive-interface
- distribute-list
- prefix-list
- AD的把控

- route-map
- 路由标记
- offset-list
- Cisco IOS IP SLAs
- PBR
- ……

路径控制

再次强调一下，这里只是简单的各种工具的罗列，在实际网络的部署中，需要根据实际的情况，灵活的挑选最经济、最科学、最可靠的工具和方法来部署。

路径控制 案例回顾



R1的配置如下：

```
Ip route 10.1.1.0 255.255.255.0 10.1.254.1
Ip route 10.1.2.0 255.255.255.0 10.1.254.1

access-list 1 permit 10.1.1.0
access-list 2 permit 10.1.2.0

route-map cisco permit 10
    match ip address 1
    set metric 10
route-map cisco permit 20
    match ip address 2
    set metric 20

router ospf 100
  redis static route-map cisco
```

Offset-list 偏移列表

Offset-list的作用

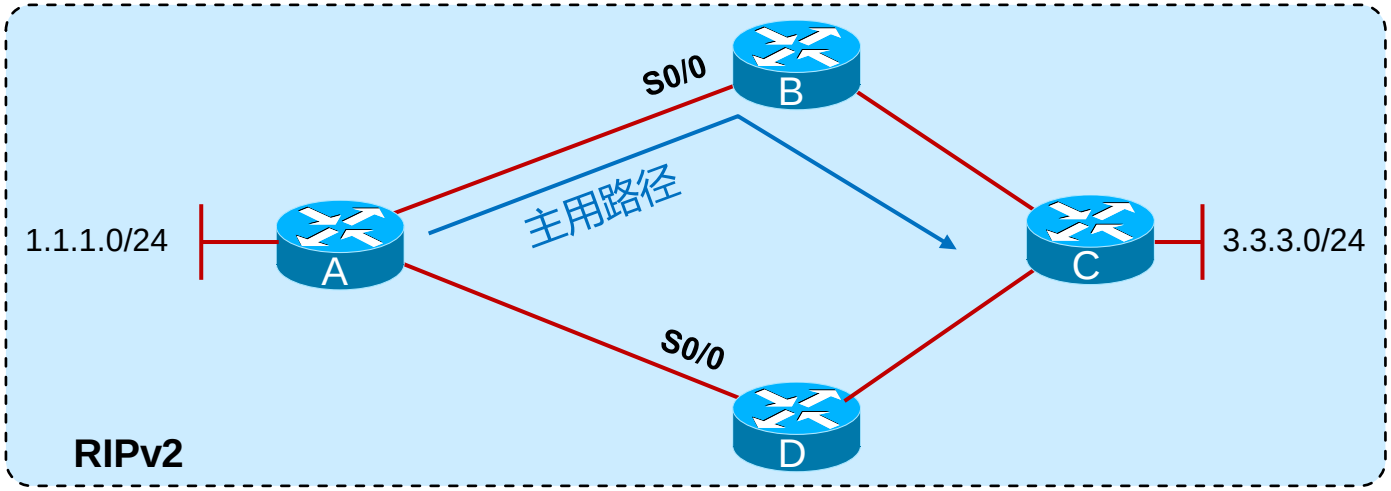
- **用于在入站或出站时增大通过EIGRP或RIP获悉的路由的度量值。**

router(config-router)#

offset-list {*access–list-number* | *name*} {in|out} *offset* [*interface-type interface-number*]

Offset-list

- **offset-list的配置（RIP）**

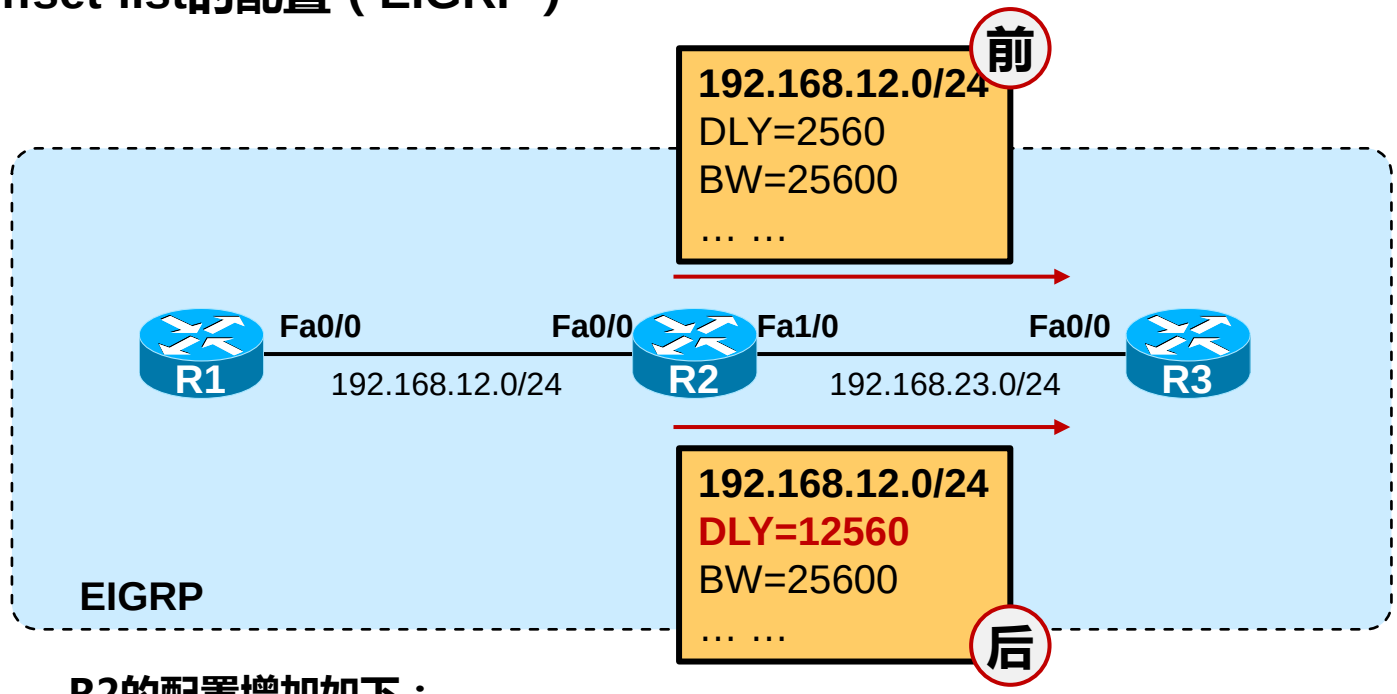


D路由器的配置

```
access-list 1 permit 3.3.3.0
router rip
  offset-list 1 out 2 serial 0/0
```

Offset-list

- **offset-list的配置（EIGRP）**

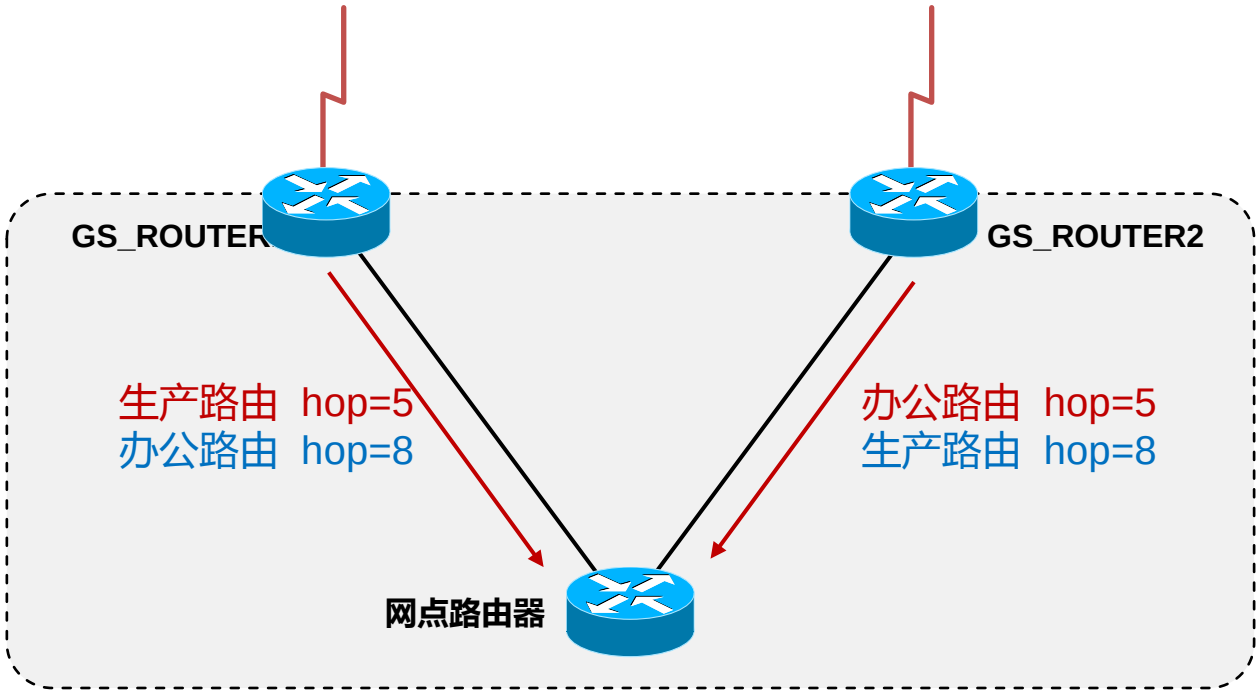


R2的配置增加如下：

```
access-list 1 permit 192.168.12.0
router rip
 offset-list 1 out 10000 fastEthernet 1/0
```

Offset-list

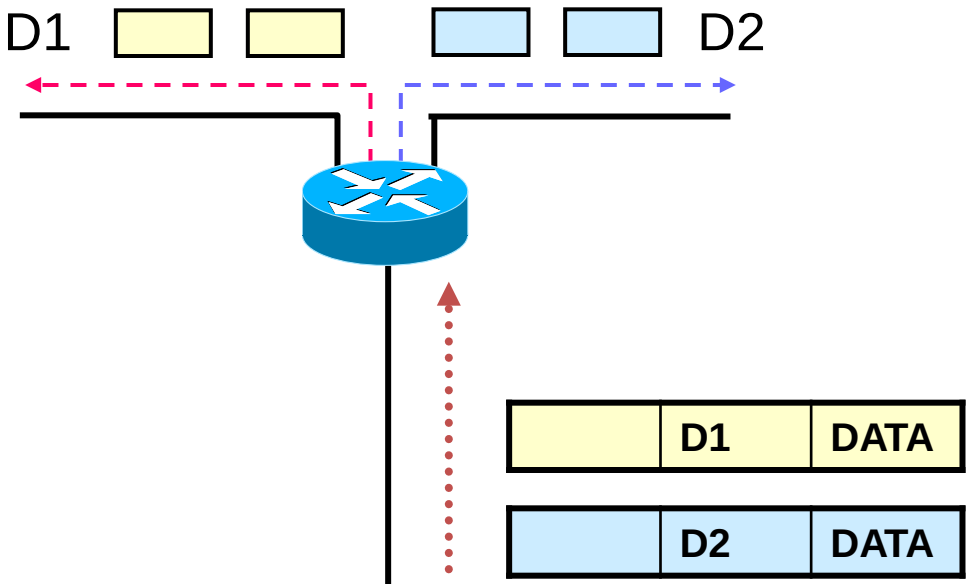
- **实施案例**



Policy-based routing

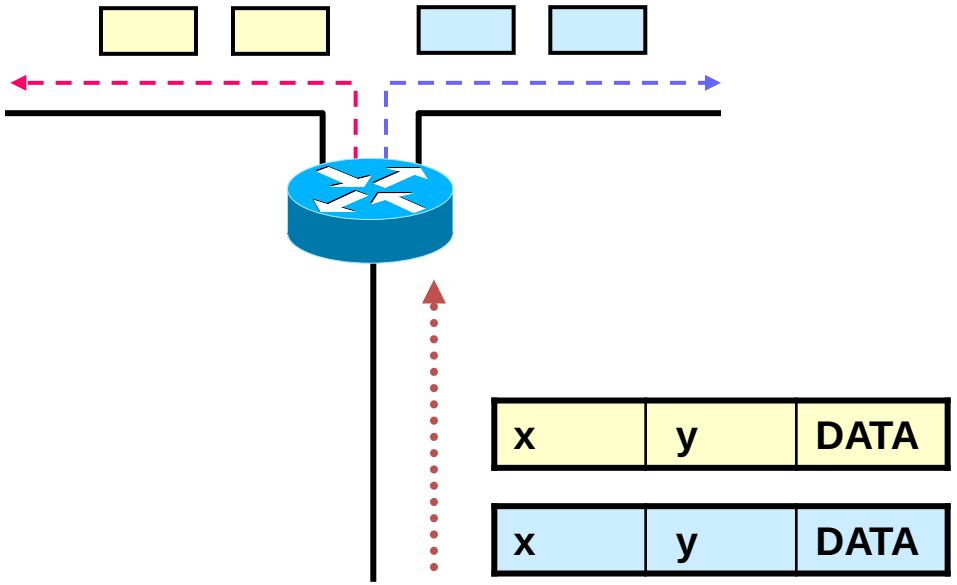
PBR策略路由

- **传统路由**



PBR策略路由

- **策略路由(Policy-Based Routing)**

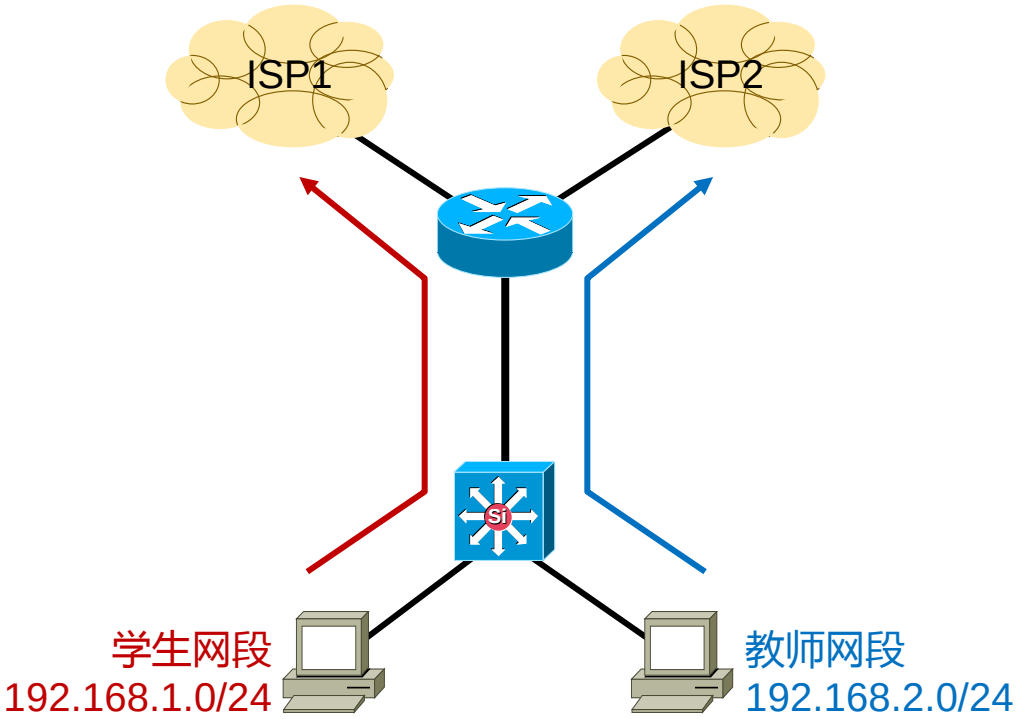


PBR策略路由

- 基于策略的路由比传统路由能力更强，使用更灵活，它使网络管理者不仅能够根据目的地址而且能够根据协议类型、报文大小、应用或IP源地址来选择转发路径.

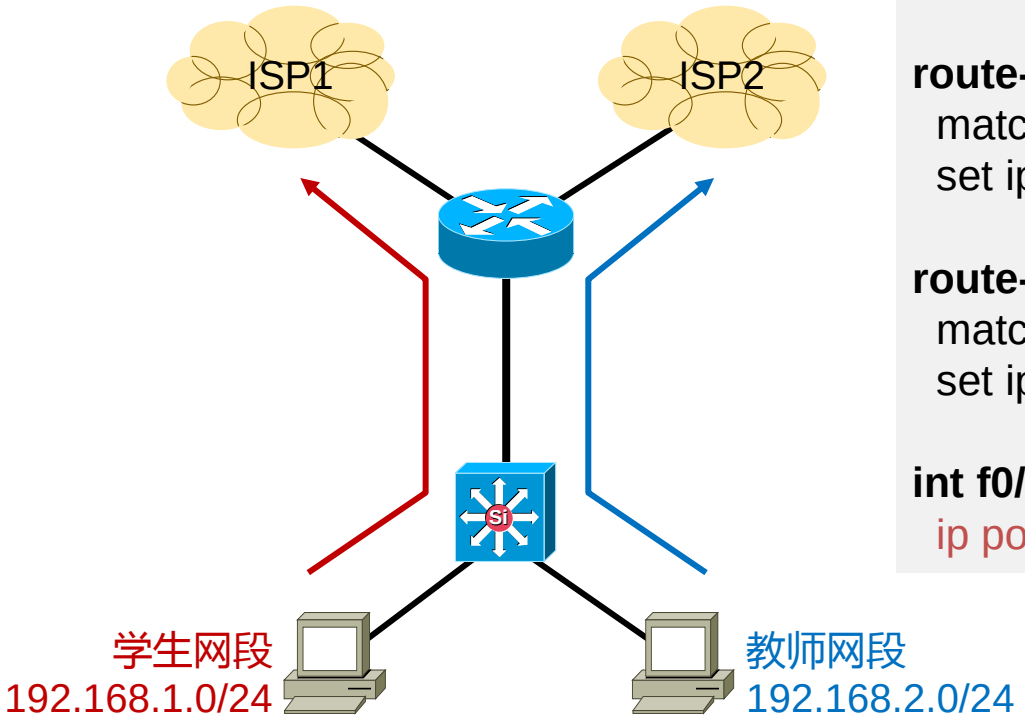
PBR策略路由

- **PBR应用示例**



PBR策略路由

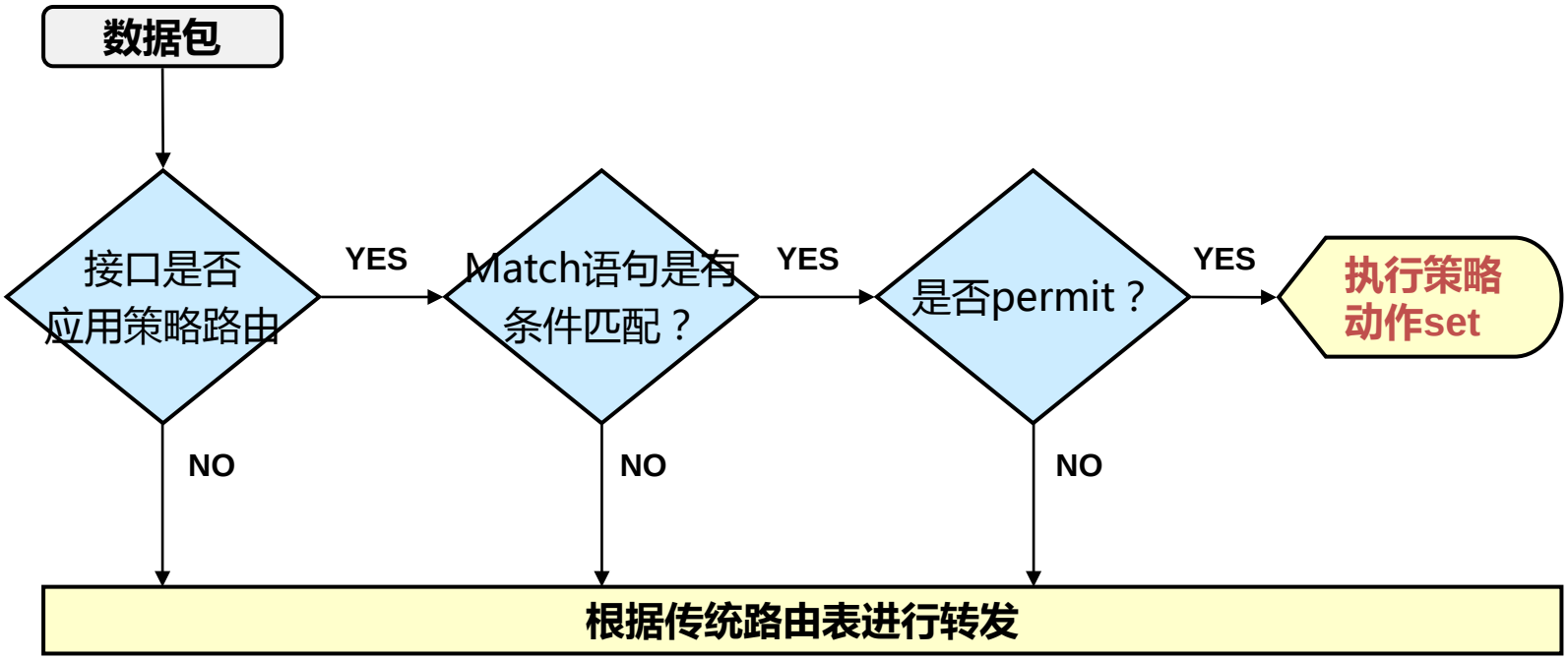
- **PBR应用示例**




```
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 2 permit 192.168.2.0 0.0.0.255

route-map test permit 10
  match ip address 1
  set ip next-hop ISP1的下一跳地址

route-map test permit 20
  match ip address 2
  set ip interface ISP2的下一跳地址

int f0/0
  ip policy route-map test
```

PBR策略路由

- **PBR对数据的处理**



PBR的配置

匹配数据包IP地址、前缀列表

Router(config)# **route-map** *rp-name*

Router(config-route-map)# **match** *ip address {access-list-number|name}*

[…access-list-number|name]|prefix-list prefix-list-name […prefix-list-name]

匹配数据包大小

Router(config-route-map)# **match** *length min max*

PBR的配置

设定分组的下一跳IP（必须为直连IP）

```
set  ip  next-hop  ip-address […ip-address]
```

设定分组的出接口

```
set  interface  type  number […type number]
```

PBR的配置

应用PBR（对进入接口的数据流量生效，本地始发的流量无效）

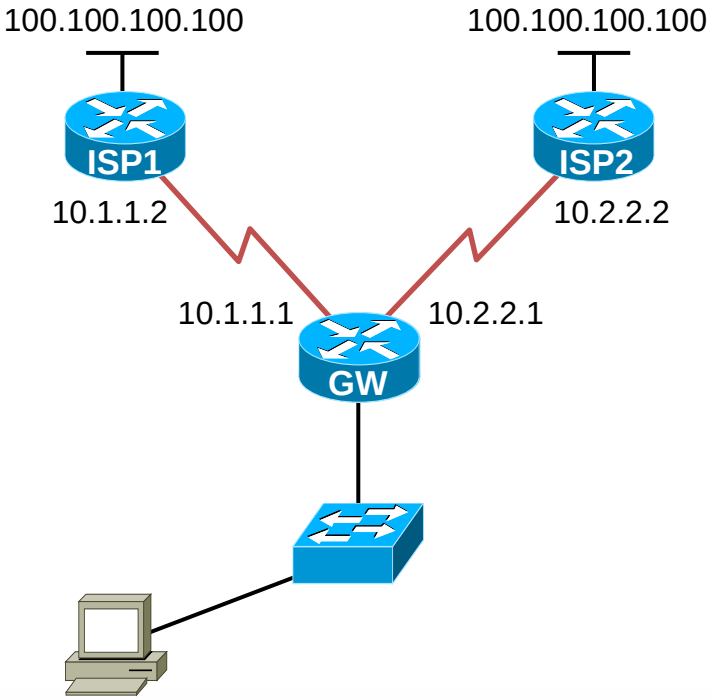
```
router(config-if)# ip policy  route-map map
```

应用PBR（针对本地始发的流量生效）

```
router(config)# ip local  policy  route-map map
```

PBR策略路由

- **PBR的配置 场景1**




```
access-list 1 permit any
route-map PBR permit 10
 match ip address 1
 set ip next-hop 10.1.1.2 10.2.2.2
```

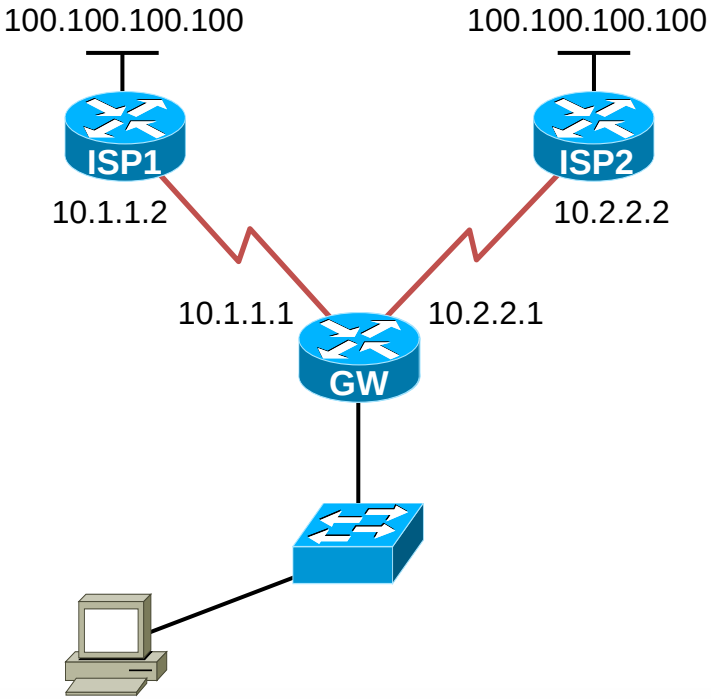
当网络正常时，PC访问外网数据走ISP1

当ISP1宕机时（GW感知到），数据切换至ISP2

当ISP1宕机时（GW无感知），数据仍然走ISP1

PBR策略路由

- **PBR的配置 场景2**




```
access-list 1 permit any
route-map PBR permit 10
 match ip address 1
 set ip next-hop 10.1.1.2 10.2.2.2
 set ip next-hop verify-availability
```

当网络正常时，PC访问外网数据走ISP1

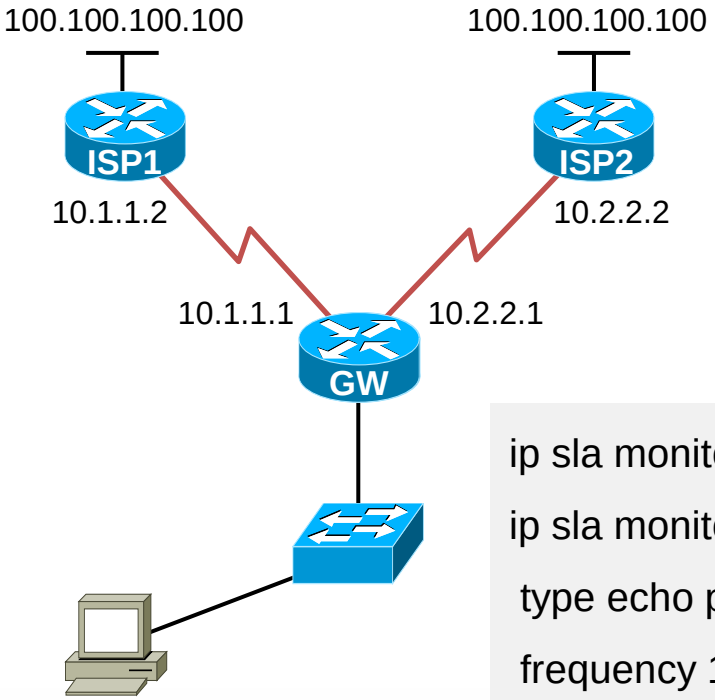
当ISP1宕机时（GW感知到），数据切换至ISP2

当ISP1宕机时（GW通过CDP感知到），数据切换至ISP2

set ip next-hop verify-availability 需借助CDP

PBR策略路由

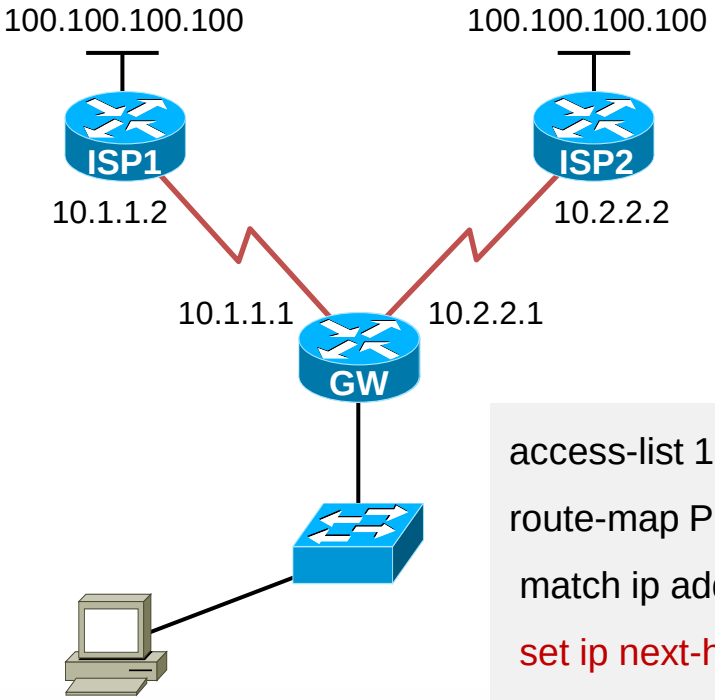
- **PBR的配置 场景3**




```
ip sla monitor responder
ip sla monitor 1
 type echo protocol ipIcmpEcho 10.1.1.2 source-ipaddr 10.1.1.1
 frequency 10
ip sla monitor schedule 1 life forever start-time now
track 1 rtr 1 reachability
```

PBR策略路由

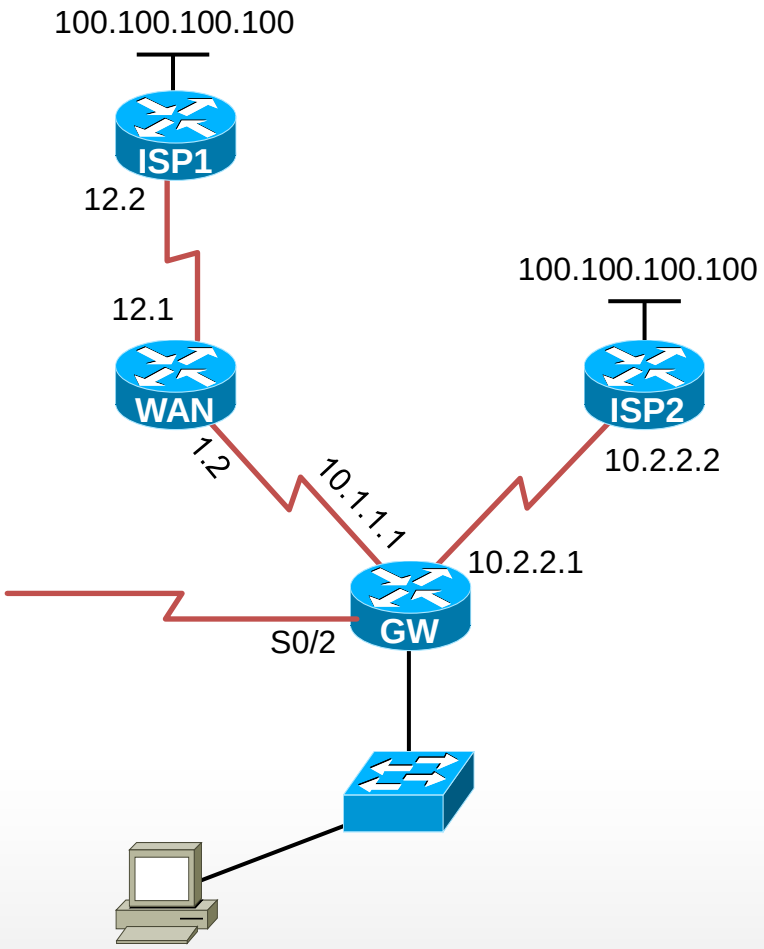
- **PBR的配置 场景3**




```
access-list 1 permit any
route-map PBR permit 10
 match ip address 1
 set ip next-hop verify-availability 10.1.1.2 10 track 1
 set ip next-hop verify-availability 10.2.2.2 20 track 2
```

PBR策略路由

- **PBR的配置 场景4**



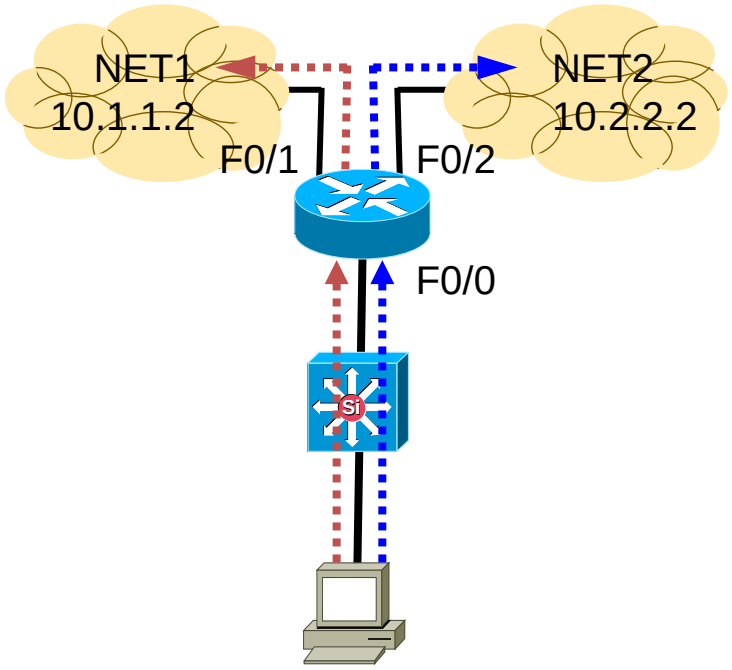

```
access-list 1 permit any
route-map PBR permit 10
 match ip address 1
 set ip next-hop 10.2.2.2
 set ip next-hop recursive 10.1.12.2

ip route 10.1.12.0 255.255.255.0 10.1.1.2
ip route 0.0.0.0 0.0.0.0 serial s0/2
```

PBR策略路由

- **PBR的验证**
  - show ip policy
  - show route-map *[map-name]*

PBR策略路由

- **PBR案例1**

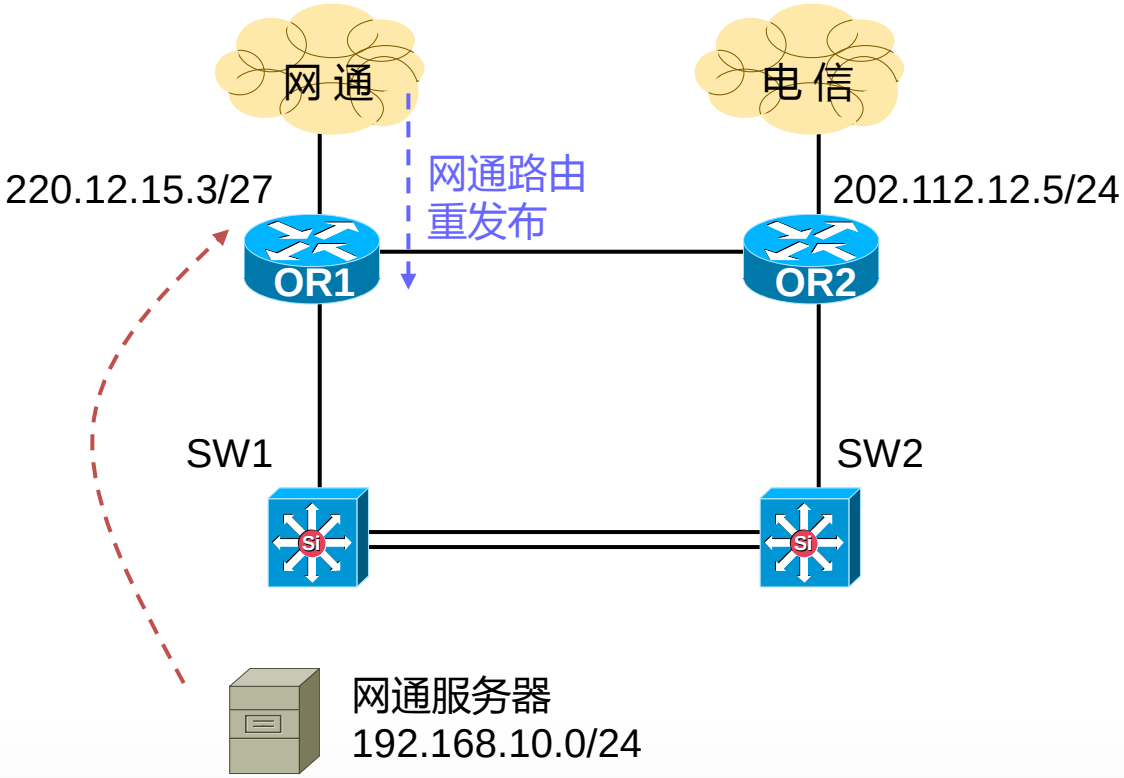


学生：192.168.1.0
老师：192.168.2.0

```
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 2 permit 192.168.2.0 0.0.0.255

route-map test permit 10
match ip address 1
set ip next-hop 10.1.1.2

route-map test permit 40
match ip address 2
set ip next-hop 10.2.2.2

int f0/0
 ip policy route-map test

ip route 0.0.0.0 0.0.0.0 10.1.1.2
ip route 0.0.0.0 0.0.0.0 10.2.2.2 10
```

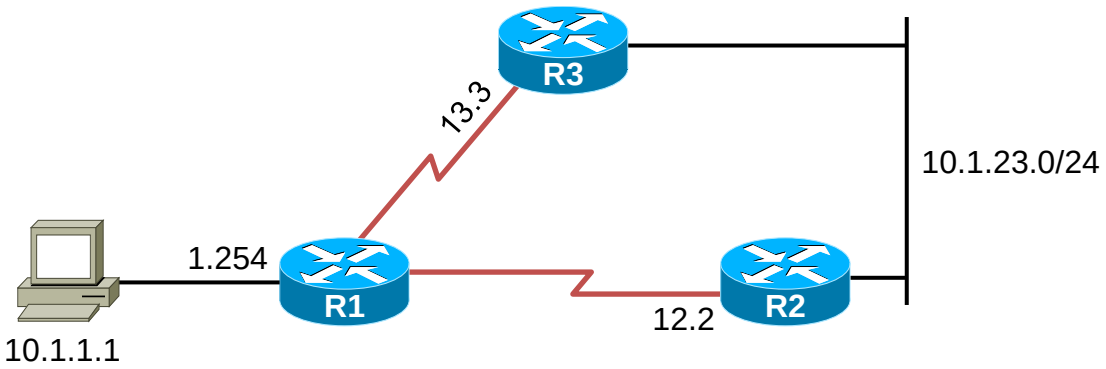
PBR策略路由

- **PBR案例2**
  - 电信用户通过网通服务器映射的外网地址访问该服务器，无法访问？



PBR策略路由

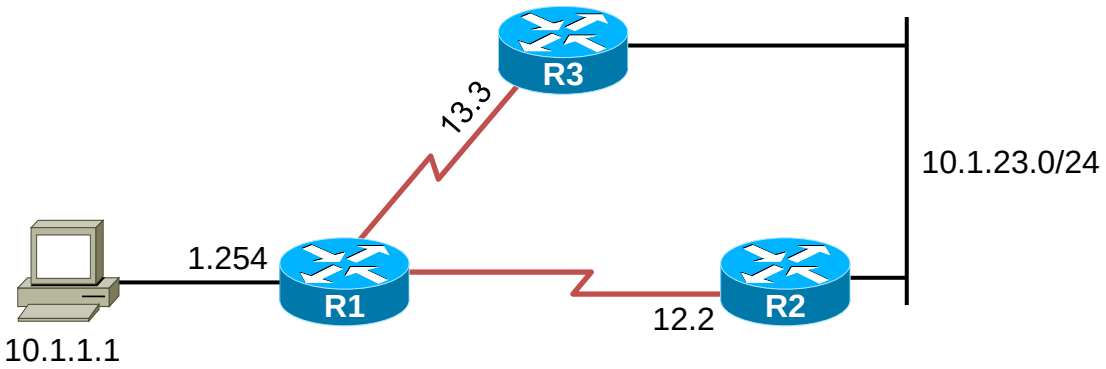
- **PBR案例3**




```
access-list 1 permit 10.1.1.0 0.0.0.255

route-map PBR permit 10
  match ip address 1
  set ip next-hop 10.1.13.3

int f0/0
  ip policy route-map PBR

ip route 0.0.0.0 0.0.0.0 10.1.13.2
```

PBR策略路由

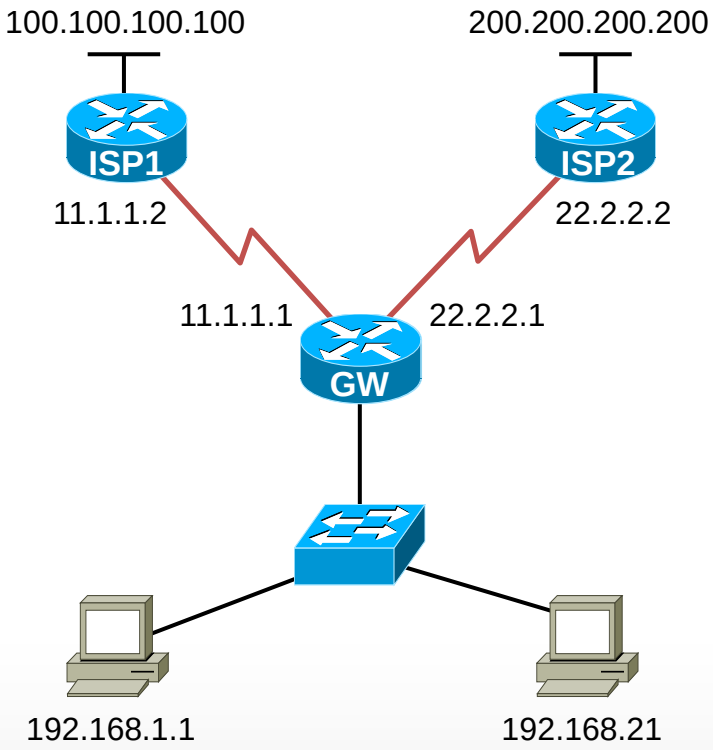
- **PBR案例3(cont.)**




```
access-list 1 permit 10.1.1.0 0.0.0.255

route-map PBR permit 10
  match ip address 1
  set ip default next-hop 10.1.13.3

int f0/0
  ip policy route-map PBR

Ip route 10.1.23.0 255.255.255.0 10.1.12.2
```

PBR策略路由

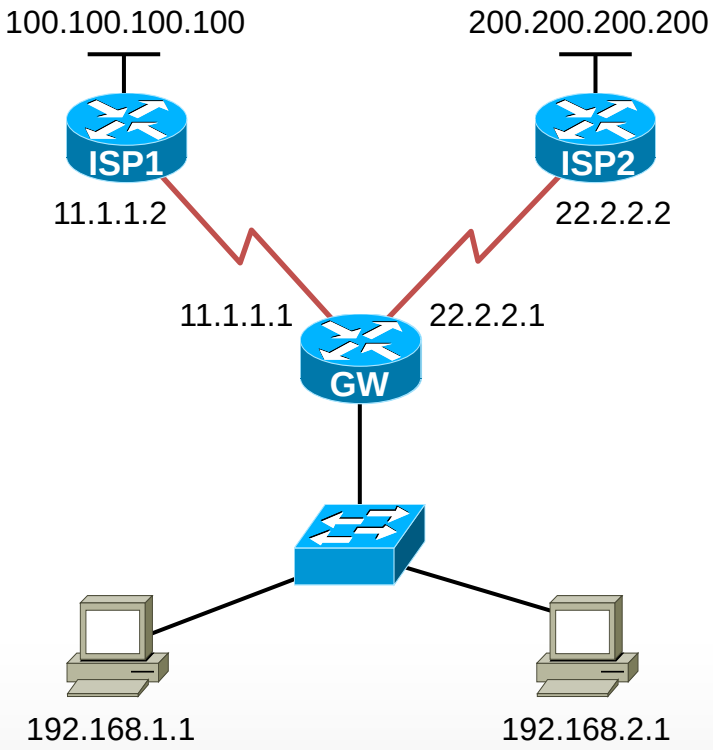
- **PBR案例4**



- 192.168.1.0用户访问外网强制走ISP1的线路，当ISP1线路DOWN掉，则自动切换至ISP2
- 192.168.2.0用户访问外网强制走ISP2的线路，当ISP2线路DOWN掉，则自动切换至ISP1
- 内网为私有IP，访问公网需使用公网地址

PBR策略路由

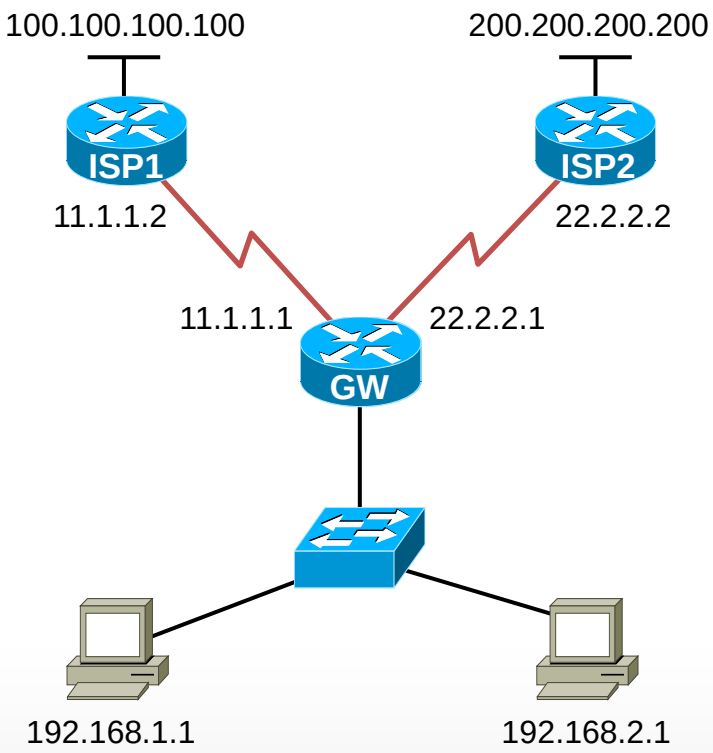
- **PBR案例4 配置**




```
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 2 permit 192.168.2.0 0.0.0.255
route-map PBR permit 10
  match ip address 1
  set ip next-hop 11.1.1.2
exit
route-map PBR permit 20
  match ip address 2
  set ip next-hop 22.2.2.2
exit
```

PBR策略路由

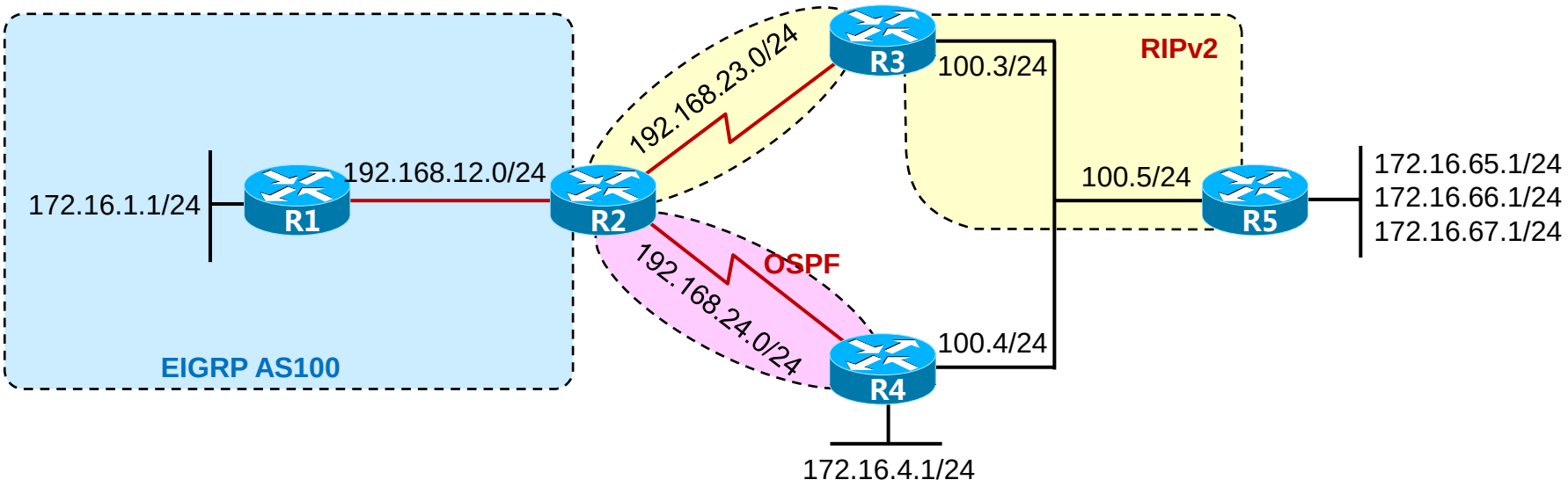
- **PBR案例4 配置cont.**




```
route-map nat1 permit 10
  match ip address 1
  match interface serial0/0          !! 匹配数据包的出口
route-map nat2 permit 10
  match ip address 1
route-map nat3 permit 10
  match ip address 2
  match interface serial0/1
route-map nat4 permit 10
  match ip address 2

ip nat inside source route-map nat1 interface serial0/0 overload
ip nat inside source route-map nat2 interface serial0/1 overload
ip nat inside source route-map nat3 interface serial0/1 overload
ip nat inside source route-map nat4 interface serial0/0 overload
```

红茶三杯
Vinsoney

综合实验



【实验需求】：

1、R1能访问23.0、24.0，同时能访问100.0以及R5的所有LOOPBACK接口；

2、R1访问R5下属LOOPBACK接口时，数据流走向为R1、R2、R3、R5，同时确保往返路径一致

3、当R2、R3之间的链路故障时，R1访问R5下属LOOPBACK的流量自动切换为经R2、R4、R5且往返路径一致

4、R1访问R4下属LOOPBACK，默认走R2、R4，且往返路径一致。

当R2至R4间的链路DOWN了，则自动切换至 R2、R3、R4，且往返路径一致

5、R3、R5均宣告各自的以太网接口进RIP进程，R4在100.0的接口上并不激活任何动态路由协议

6、注：除R4及R5外，其他路由器上不允许配置任何静态路由

红茶三杯
Vinsoney

学习 沉淀 成长 分享

关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

BGP概述及路径属性详解

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2012-08-01

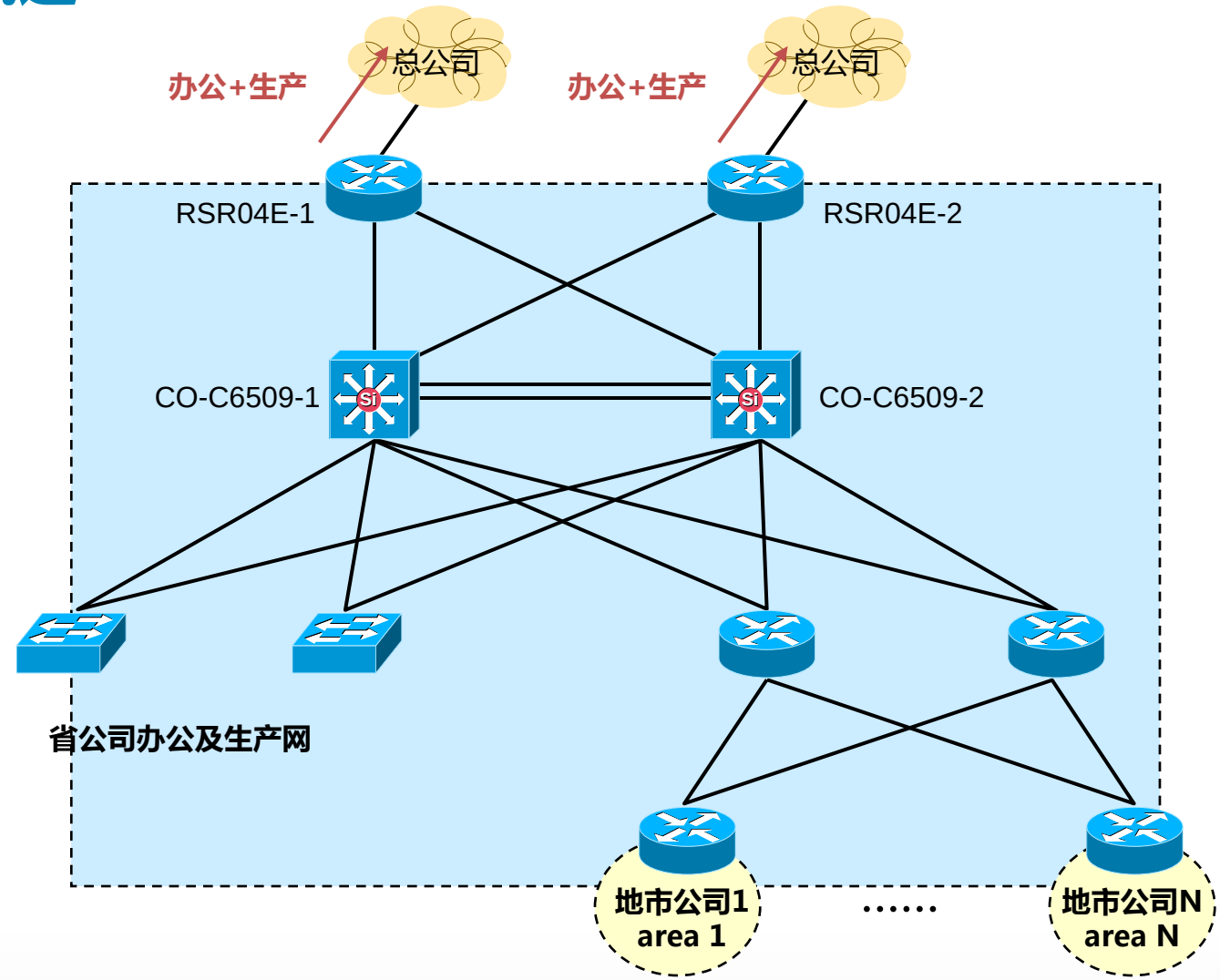
课程目标

- BGP概述
- BGP基本配置
- 路径属性详解

红茶三杯
Vinsoney

BGP概述

BGP概述

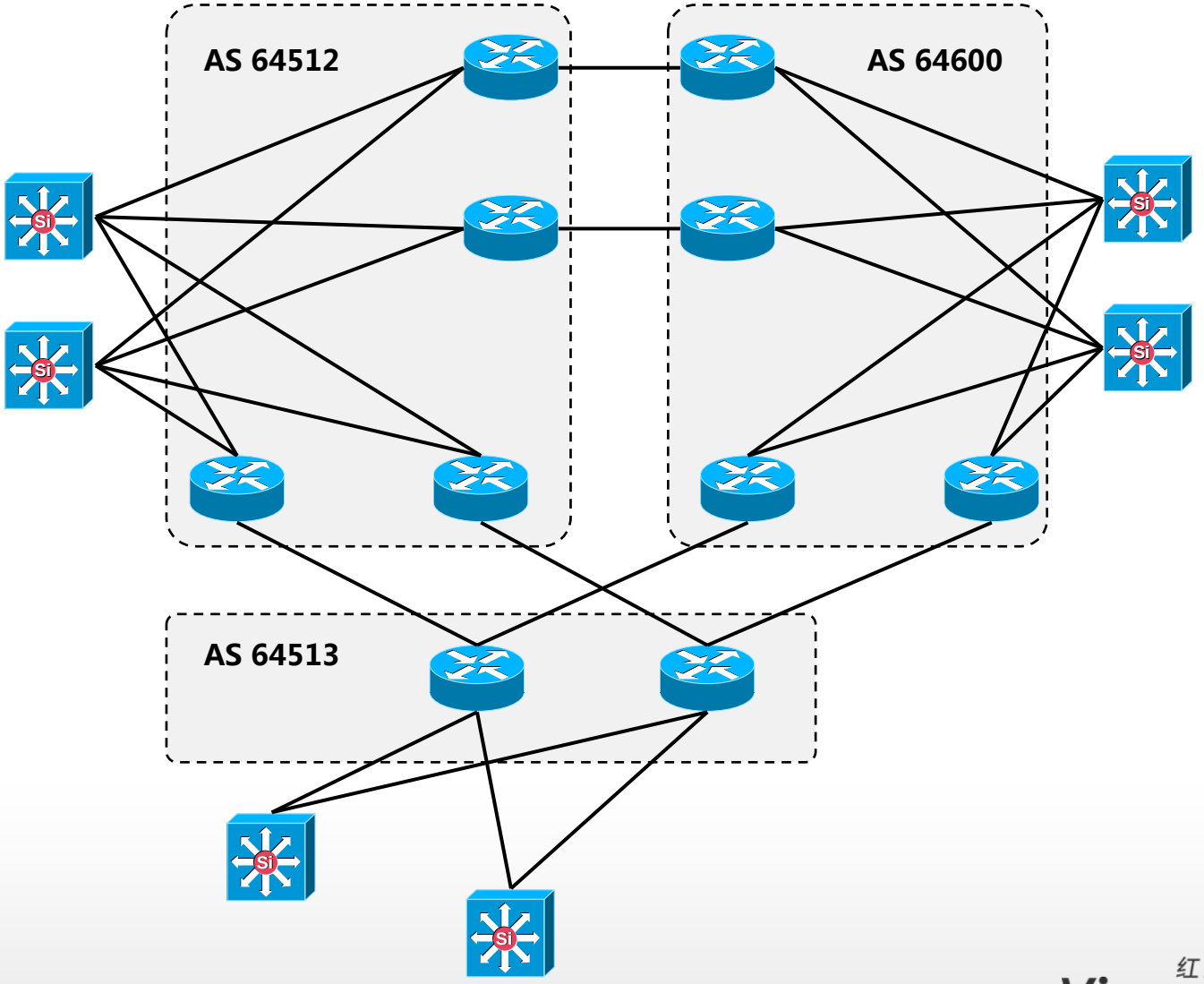


BGP概述

- **IGP具有以下某些特性或全部特性**
  - 执行拓扑发现
  - 尽力完成快速收敛
  - 需要周期性的更新来确保路由选择信息的精确性
  - 受同一个管理机构的控制
  - 采取了共同的路由选择策略
  - 提供了优先的策略控制能力

BGP概述

- 关于BGP



BGP概述

- **关于BGP**
  - AS：autonomous system 自治系统，指的是在同一个组织管理下使用相同策略的设备的集合。
  - 不同AS通过AS号区分，AS号取值范围1 - 65535，其中64512 - 65535是私有AS号。IANA负责AS号的分发。
  - 中国电信163 AS号：4134
  - 中国电信CN2 AS号：4809
  - 中国网通AS 号：9929

BGP概述

- **BGP概述**
  - BGP为Border Gateway Protocol 边界网关路由协议（路径矢量）
  - 主要作用是在AS之间传递路由信息
  - 目前BGP有4个版本：V1、V2、V4、V4＋（即MBGP）
- **使用BGP的三大理由：**
  - 大量路由需要承载，IGP只能容纳千条，而BGP可以容纳上万
  - 支撑MPLS/VPN的应用，传递客户VPN路由。
  - 策略能力强，可以很好的实现路由决策与数据控制。

BGP概述

- **BGP概述**
  - 企业连接到ISP

    连接到两家或是多家ISP，提供链路的可靠性，连接方式如下：

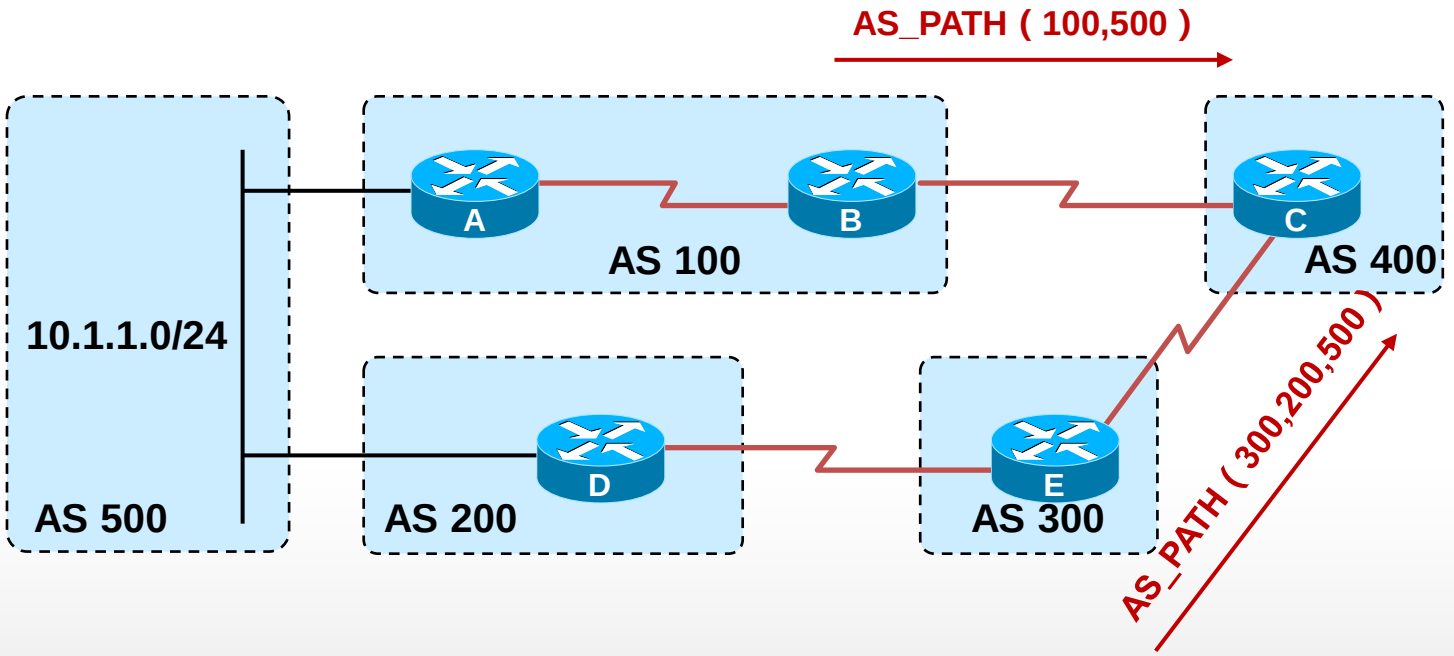
    1. Single homed 单宿：只连接到一家ISP且没有冗余链路
    2. Dual homed 双宿：只连接到一家ISP，使用两条链路来提供冗余
    3. Multihomed 多宿：连接到多家ISP
    4. Dual Multihomed 双多宿：连接到多家ISP，同时使用两条链路

BGP概述

- **BGP概述**
  - 企业连接到ISP
  - 采用多宿或是双多宿的原因：
  - 1、提高internet连接的可靠性：一条连接出现故障时，可使用另一条
  - 2、提高连接的性能：前往某些目的地时，可使用更佳的路径

BGP概述

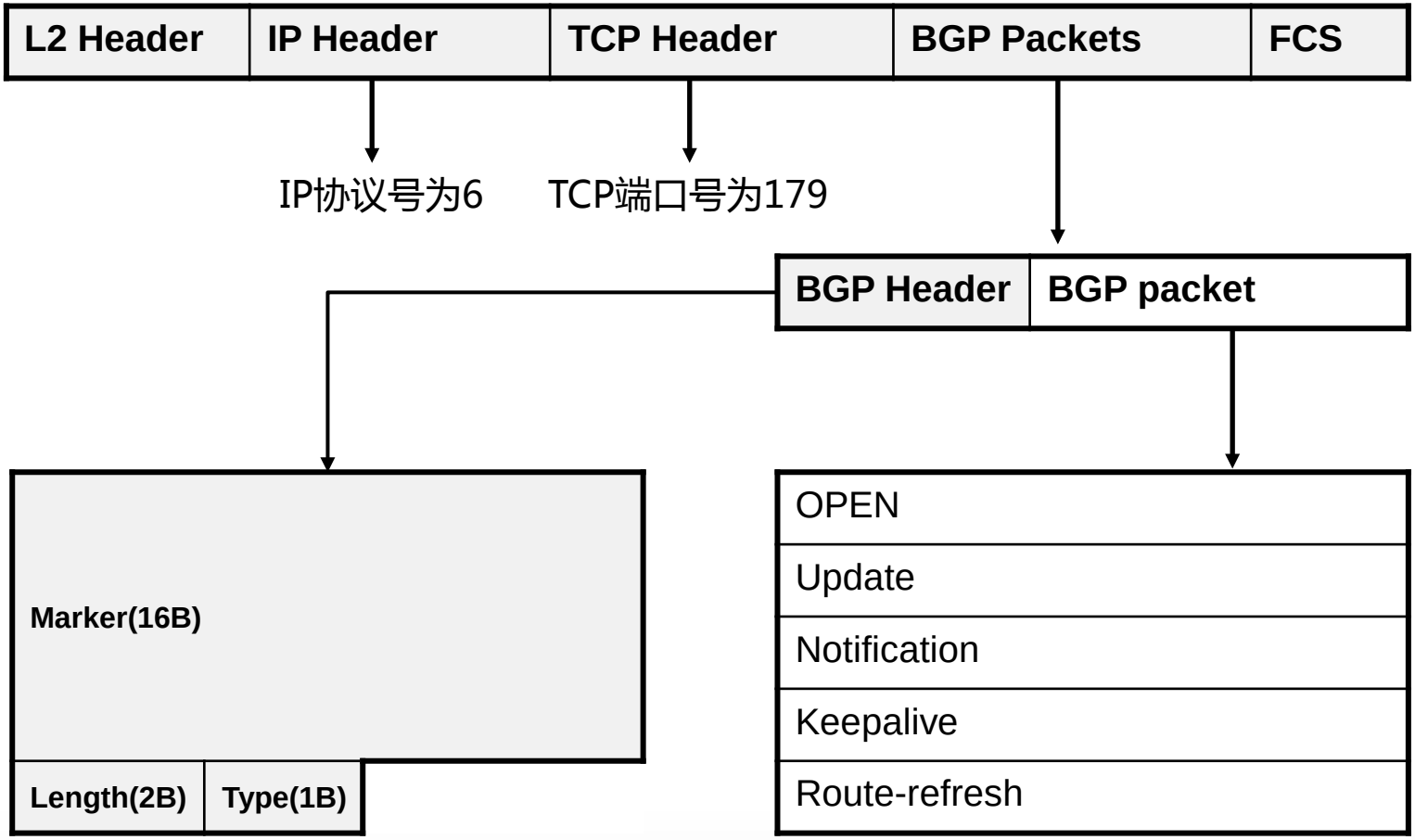
- **BGP的路径矢量特征**
  - 路径矢量信息中包含一个BGP自治系统号列表
  - BGP路由器不接受路径列表中包含其AS号的路由更新，是无环路的。
  - BGP支持对BGP自治系统路径应用路由策略
  - BGP路由器只能将其使用的路由通告给邻接自治系统中的对等体



BGP概述

- **BGP特征**
  - BGP使用TCP为传输层协议，TCP端口号179
  - BGP路由器之间建立TCP连接，这些路由器称为BGP对等体也叫BGP邻居：EBGP、IBGP
  - 对等体之间交换整个BGP路由表
  - BGP路由器只发送增量更新或触发更新（不会周期性更新）
  - 具有丰富的路径属性
  - BGP通告成千上万的路由，可采用TCP滑动窗口的机制，停止并等待确认前，可以发送65576个字节

BGP packets



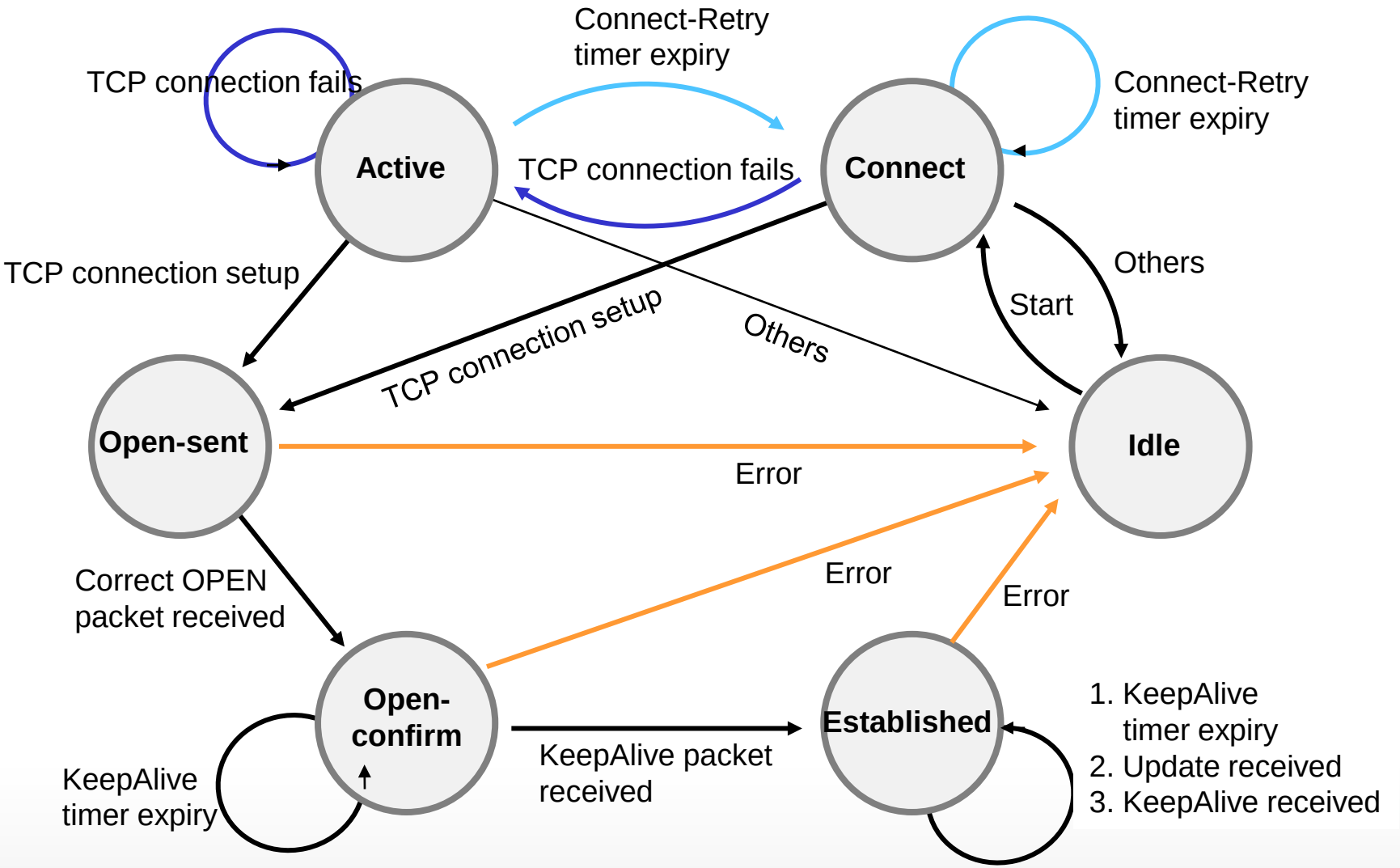
BGP packets

报文名称	作用是什么	什么时候发包
OPEN	协商BGP邻居的各项参数，建立邻居关系	通过TCP建立BGP连接，发送open报文
UPDATE	进行路由信息的交换	连接建立后，有路由需要发送或路由变化时，发送UPDATE通告对端路由信息
NOTIFICATION	报告错误，中止邻居关系	当BGP在运行中发现错误时，要发送NOTIFICATION报文通告BGP对端
KEEPALIVE	维持邻居关系	定时发送KEEPALIVE报文以保持BGP邻居关系的有效性
Route-refresh	为保证网络稳定，触发更新路由的机制	当路由策略发生变化时，触发请求邻居重新通告路由

BGP的有限状态机

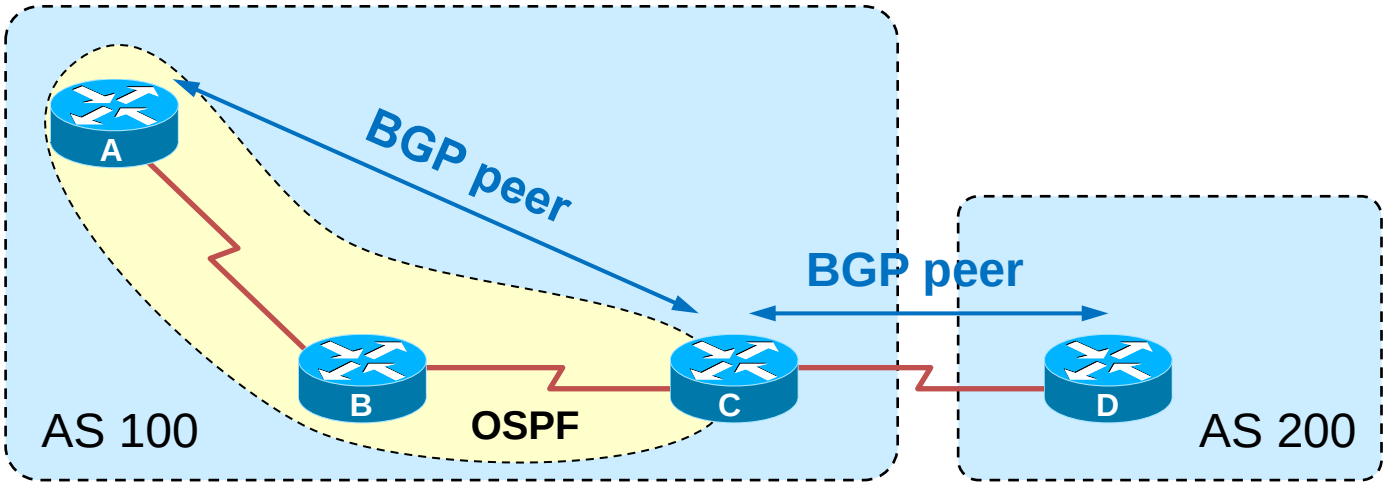
Peer状态名称	发什么包	在做什么
Idle	尝试建立TCP连接	开始准备TCP的连接并监视远程peer启动TCP连接，启用BGP时，要准备足够的资源
Connect	发TCP包	正在进行TCP连接，等待完成中，认证都是在TCP建立期间完成的。如果TCP连接不上则进入Active状态，反复尝试连接。
Active	发TCP包	TCP连接没建立成功，反复尝试TCP连接。
OpenSent	发Open包	TCP连接建立已经成功，开始发送Open包，Open包携带参数协商对等体的建立。
OpenConfirm	发Keepalive包	参数、能力特性协商成功，自己开始发送Keepalive包，等待对方的Keepalive包。
Established	发Update包	已经收到对方的Keepalive包，双方能力特性一致，开始使用Update通告路由信息。

BGP的有限状态机



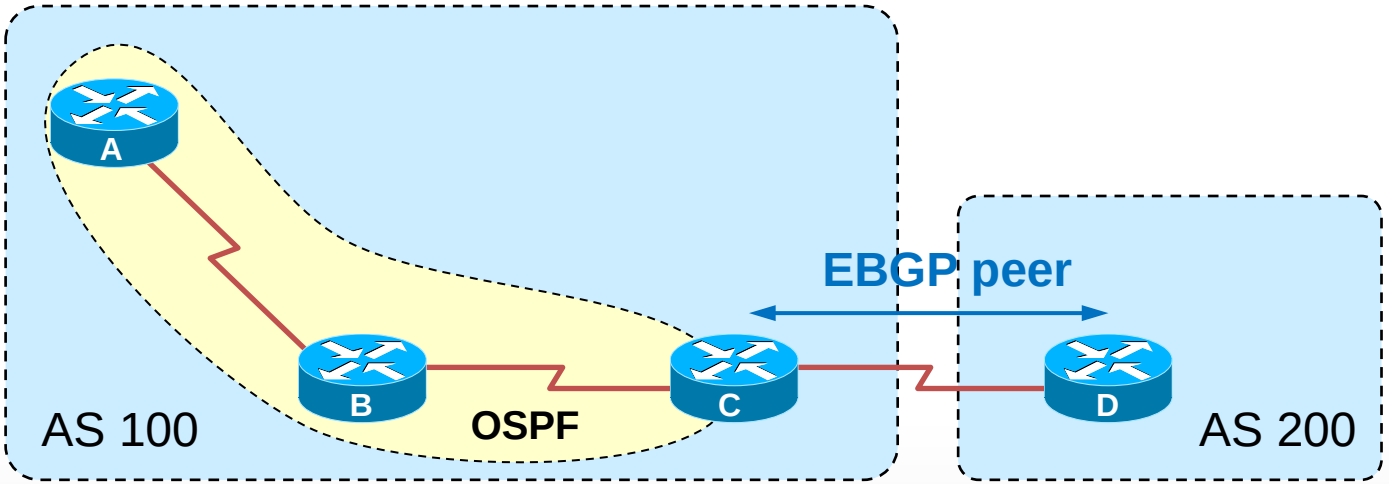
BGP Peer

- 运行BGP的路由器被称为BGP speaker
- BGP对等体也叫BGP邻居，建立基于TCP的关系



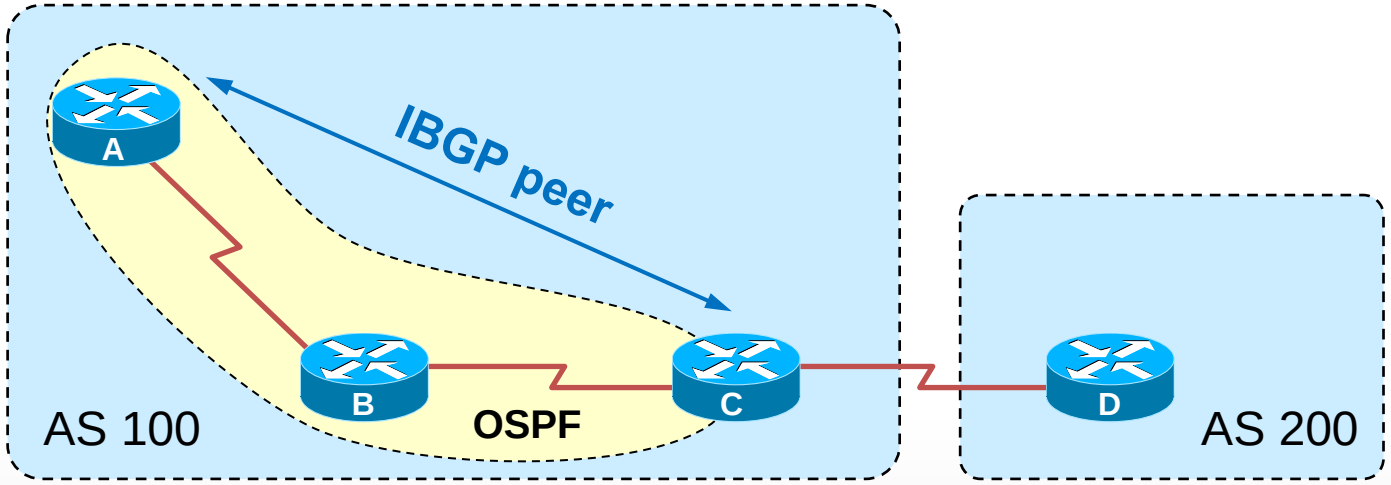
BGP Peer

- **EBGP：BGP位于不同自治系统的路由器之间的BGP邻接关系**
- **建立EBGP邻接关系，必须满足三个条件**
  - EBGP之间自制系统号不同
  - 定义邻居建立TCP会话
  - neighbor中指定的IP地址要可达

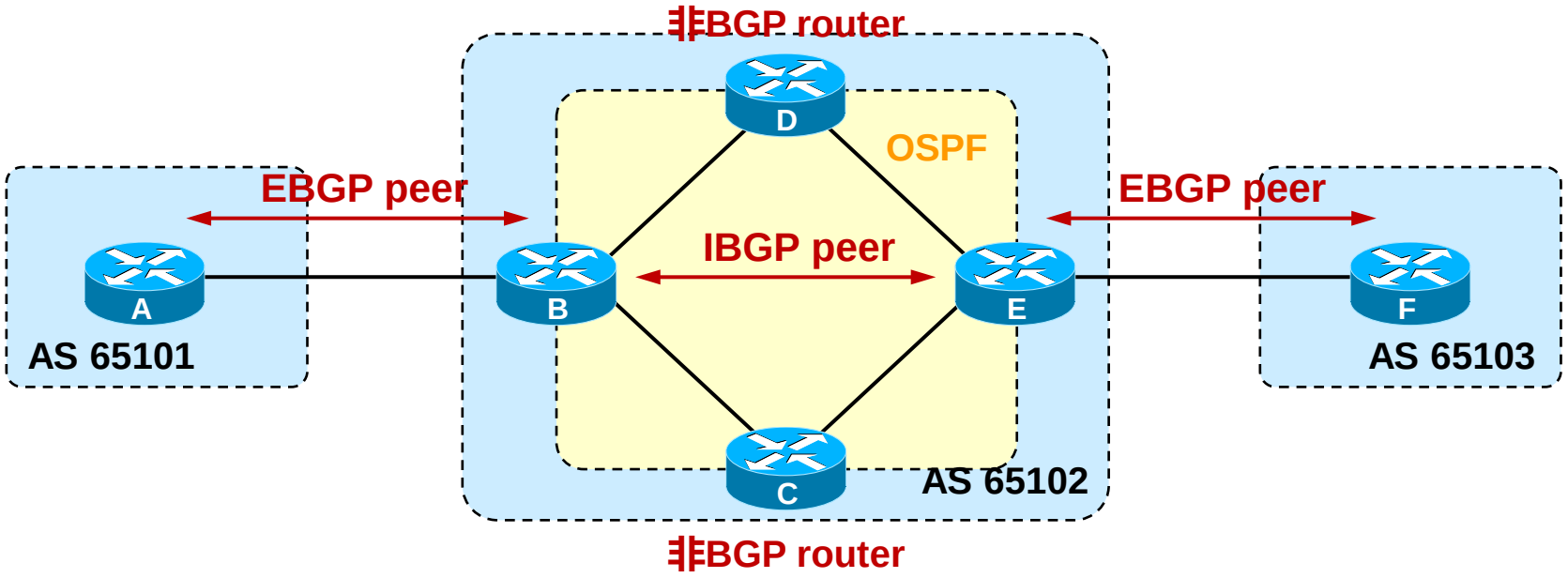


BGP Peer

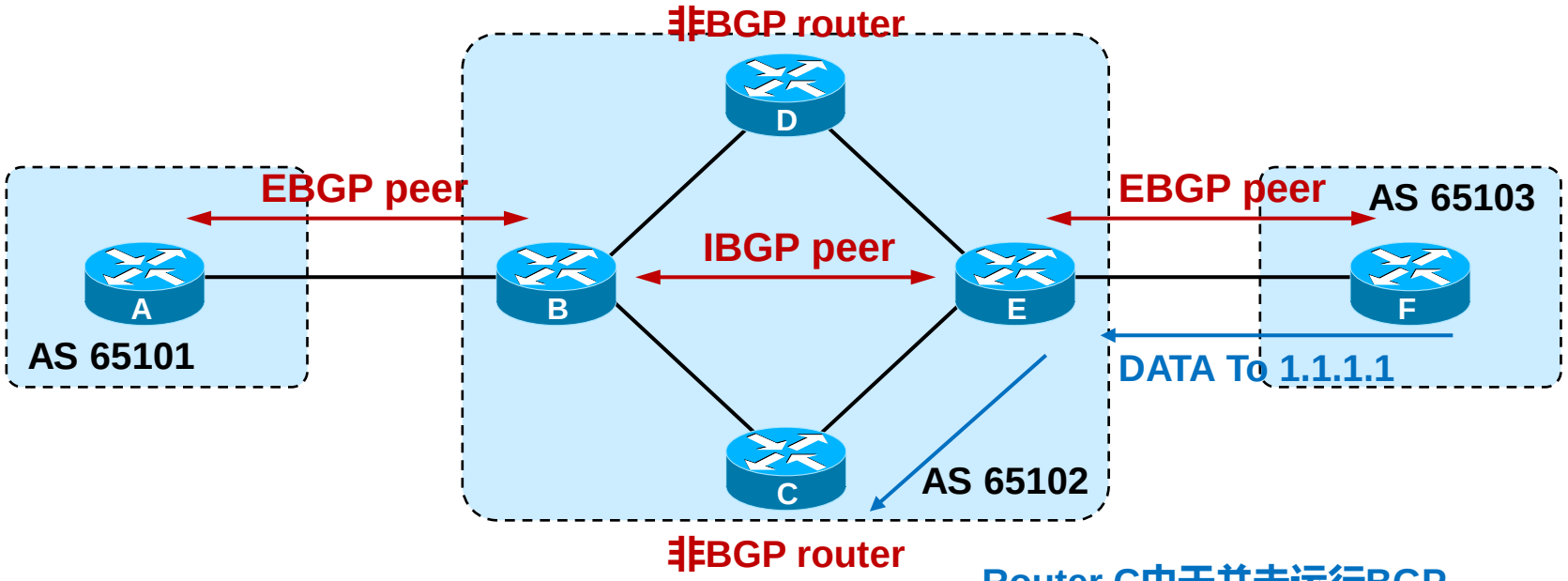
- **IBGP：位于同一个自治系统的路由器之间的BGP邻接关系**
- **建立IBGP邻接关系，满足的条件**
  - 自治系统号相同
  - 定义邻居建立TCP会话
  - IBGP邻居可达



中转AS中的IBGP路由传递



中转AS中的IBGP路由传递



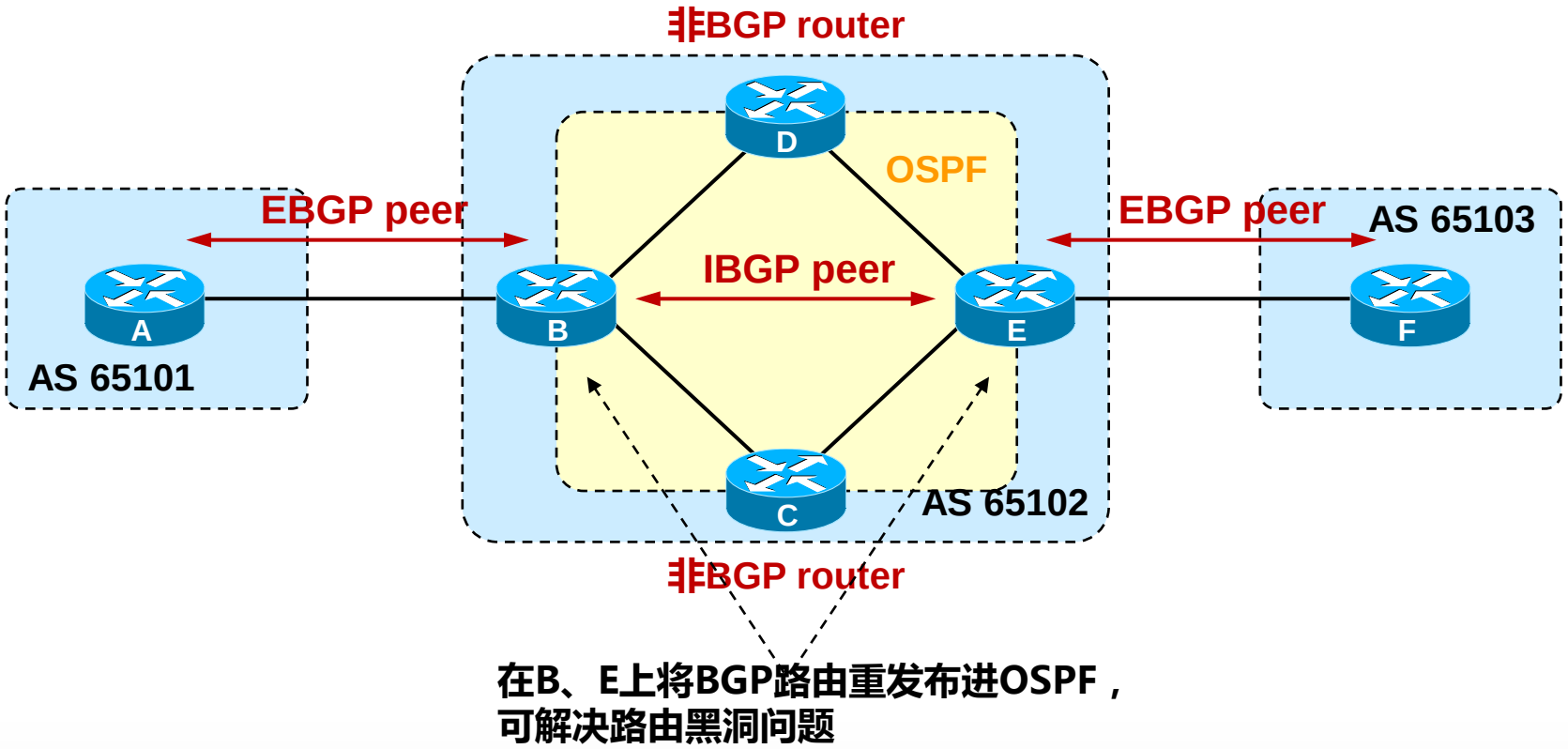
Router C由于并未运行BGP，因此不知道1.1.1.0的路由，遂丢弃数据包，这里就产生了路由黑洞

BGP同步规则

- BGP同步规则指出，BGP路由器不应使用通过IBGP获悉的路由或将其通告给外部邻居，除非该路由是本地的或通过IGP获悉的。
- Cisco IOS默认关闭同步
- 同步关闭，则BGP可以将使用这样的路由，并将其通告给外部BGP邻居：从IBGP邻居那里获悉的且没有出现在本地路由表中的路由。
- 同步开启，则路由器通过IBGP获悉路由后，将等待IGP将该路由传遍整个自治系统，然后再将其通告给外部对等体。

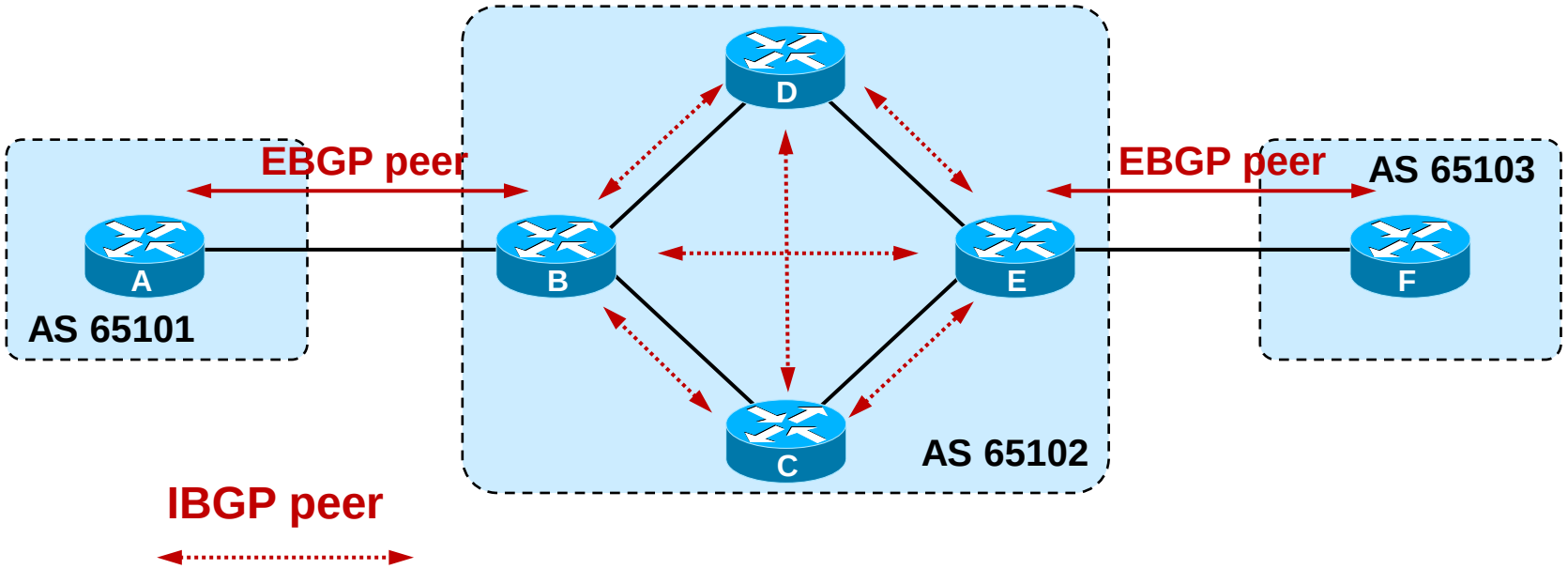
中转AS中的IBGP路由传递问题 解决办法1

- IGP-BGP路由重发布



中转AS中的IBGP路由传递问题 解决办法2

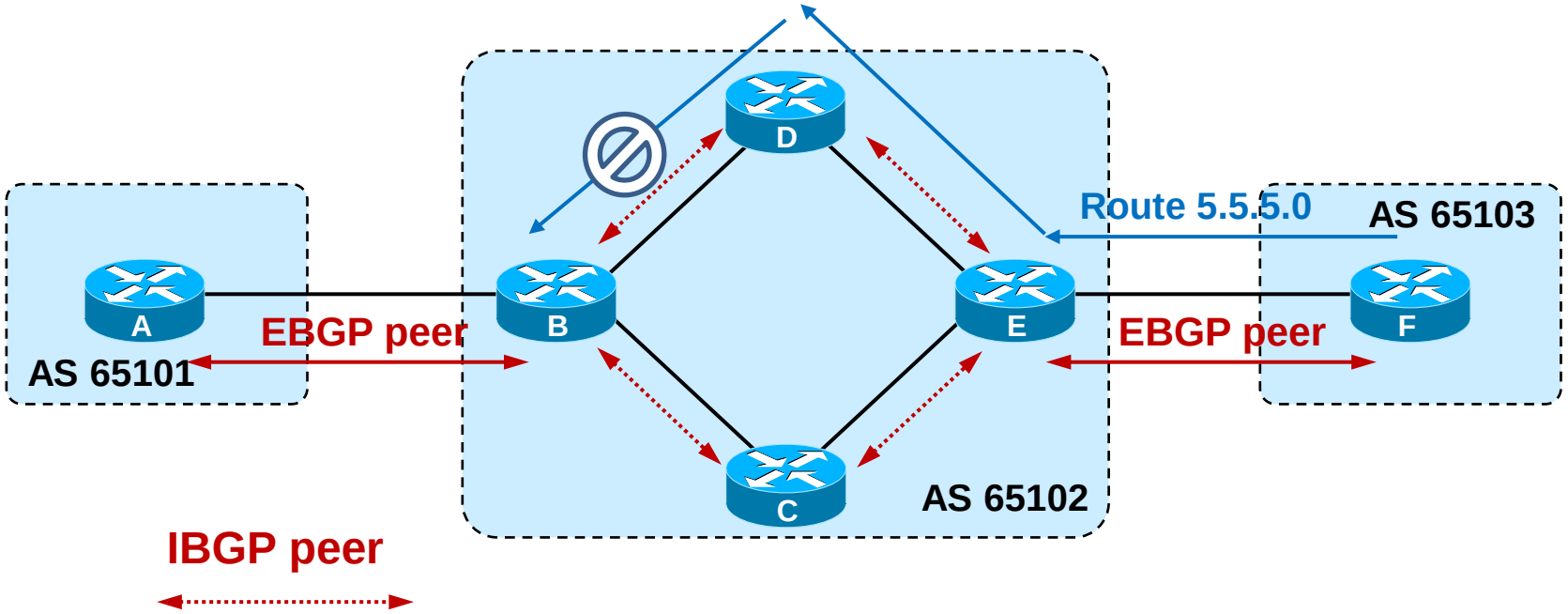
- IBGP邻居全互联



中转AS中的IBGP路由传递问题 解决办法2

- **IBGP邻居全互联**
  - IBGP全互联虽然能解决transit AS内的路由黑洞问题，但是却造成BGP路由器需耗费大量资源维护大量BGP连接的新问题。
    - **路由反射器**
    - **联邦**

IBGP水平分割原则



IBGP水平分割原则

BGP防环是通过AS_PATH实现的，而AS_PATH仅仅在路由离开AS才回被更改，因此在AS内，IBGP就没有EBGP的防环能力，为了防止环路的出现，BGP路由器不会将从IBGP邻居学习过来的路由再通告给自己其他IBGP邻居。--- BGP的水平分割原则。

由于水平分割原则存在，BGP要求AS内，须保证IBGP全互联（neighbor命令 指定）。（根本原因是在AS内部， AS-PATH不会改变，无法使用AS_PATH防环，因此很容易出现环路）

BGP路由通告规则

- 当存在多条路径时，BGP Router只选取最优的路由（BEST）来使用（没有负载均衡的情况下）
- BGP只把自己使用的路由，也就是自己认为Best的路由传递给BGP peer
- BGP Speaker从EBGP获得的路由会向它所有BGP相邻体通告（包括EBGP和IBGP）
- BGP Speaker从IBGP获得的路由不向它的IBGP相邻体通告（避免循环，水平分割；存在路由RR的情况除外）
- BGP Speaker从IBGP获得的路由是否通告给它的EBGPpeer要视IGP和BGP同步的情况来决定

Tables

- **BGP邻居表**：邻居列表
- **BGP表**：包含了从邻居学习所有路由，以及到达目的网段的多个路径和属性
- **路由表**：列出了到达目的网段的最佳路径，EBGP路由AD为20，IBGP路由AD为200

BGP表

- 运行BGP的路由器有一个独立的表
- 路由器将BGP表中最佳路由提供给IP路由表
- BGP显式的配置每个邻居，邻居之间建立TCP关系，默认每隔60S发送一次BGP/TCP存活消息，保持时间为180S

BGP的基本配置

BGP基本配置

- **创建BGP进程**

Router(config)#router bgp *autonomous-system*

- 仅仅执行命令router bgp并不能在路由器上激活BGP，必须至少执行一个子命令才能在路由器上激活BGP进程
- 在路由器上只能配置一个BGP实例

BGP基本配置

- **指定BGP邻居及激活BGP会话**

Router(config-router)# neighbor {ip-address | peer-group-name} **remote-as** autonomous-system

- 邻居指定的IP地址必须路由可达
- BGP邻居都需手工指定，不能像IGP那样通过协议自动发现
- AS决定了与邻居建立的是EBGP会话还是IBGP会话

BGP基本配置

- **指定BGP将通告的网络**

Router(config-router)#network *network-number* [mask *network-mask*] [route-map *map-tag*]

- network命令与IGP不同，BGP命令network为通告哪些IGP路由进BGP进程，而不是在接口上启用BGP
- network支持无类前缀，前缀必须与路由表中的条目完全匹配
- 如果不指定mask，只通告主类网络号，而且仅当主类网络中至少有一个子网出现在IP路由表中，BGP才会将该主类网络作为一条BGP路由通告
- 指定了mask，则仅当路由选择表中有与该网络完全匹配的条目时才被通告出去

BGP基本配置

- **BGP同步**

```
Router(config-router)#no synchronization
```

- 关闭同步（默认关闭）

```
Router(config-router)#synchronization
```

- 开启同步

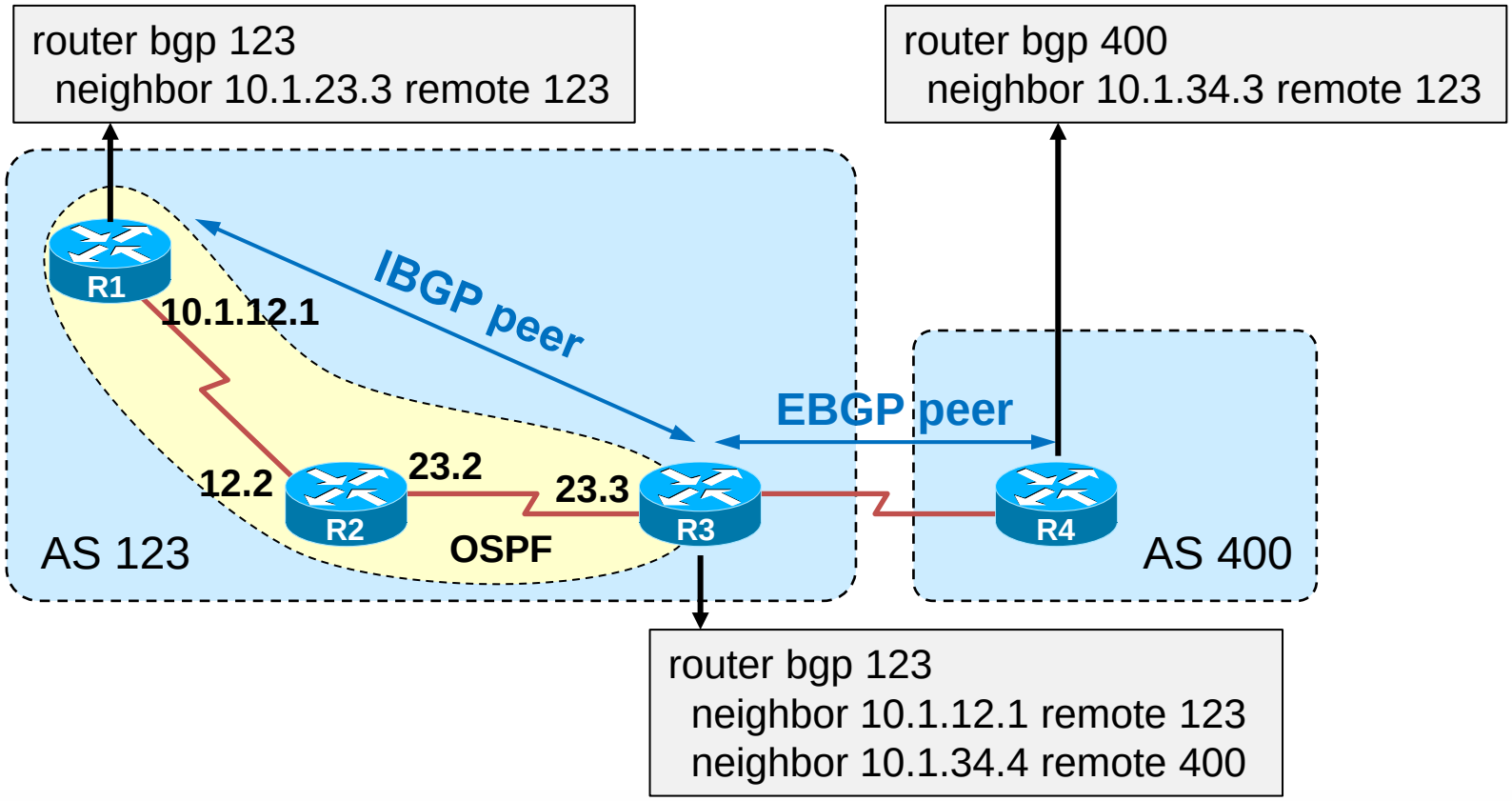
BGP基本配置

- **BGP router-id**

```
Router(config-router)# bgp router-id x.x.x.x
```

  - 手工设置BGP routerID

BGP基本配置示例

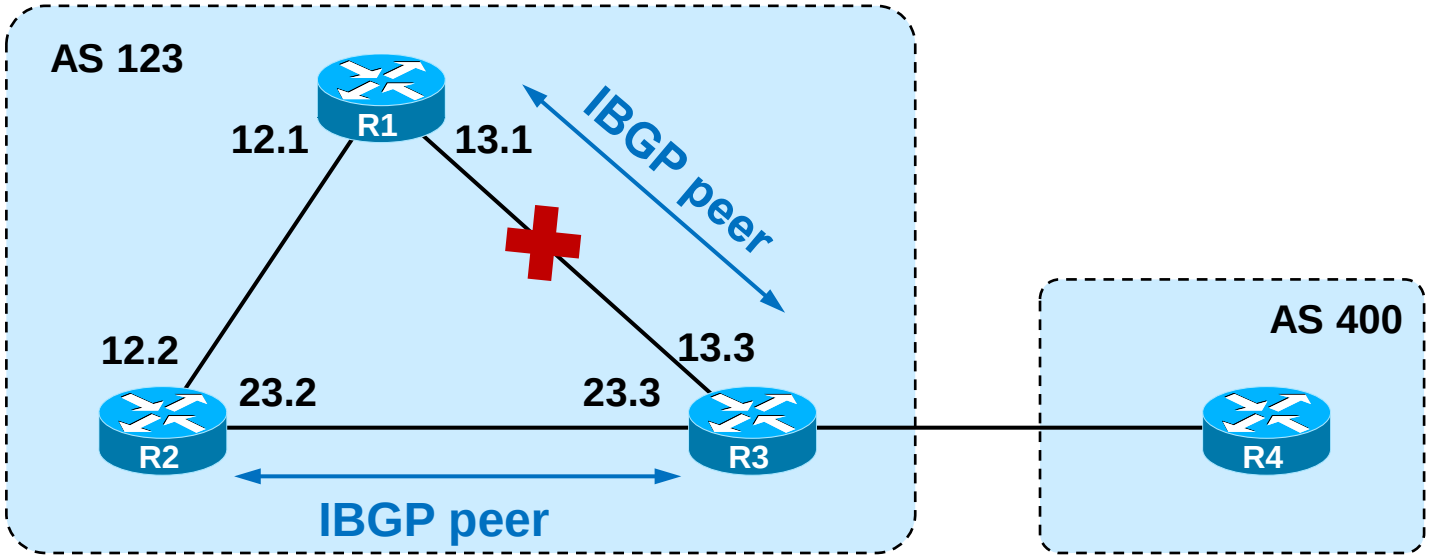


BGP基本配置

- **指定更新源**
  - 当创建BGP会话，邻居状态定义了目的IP地址和出接口定义了源IP地址
  - 对于BGP分组，源IP地址必须与另一台路由器上相应的neighbor命令指定的地址相同，不相同，则BGP分组将被忽略

BGP基本配置

- **指定更新源**




```
hostname R3
router bgp 123
  neighbor 10.1.13.1 remote 123
  neighbor 10.1.23.2 remote 123
```

BGP基本配置

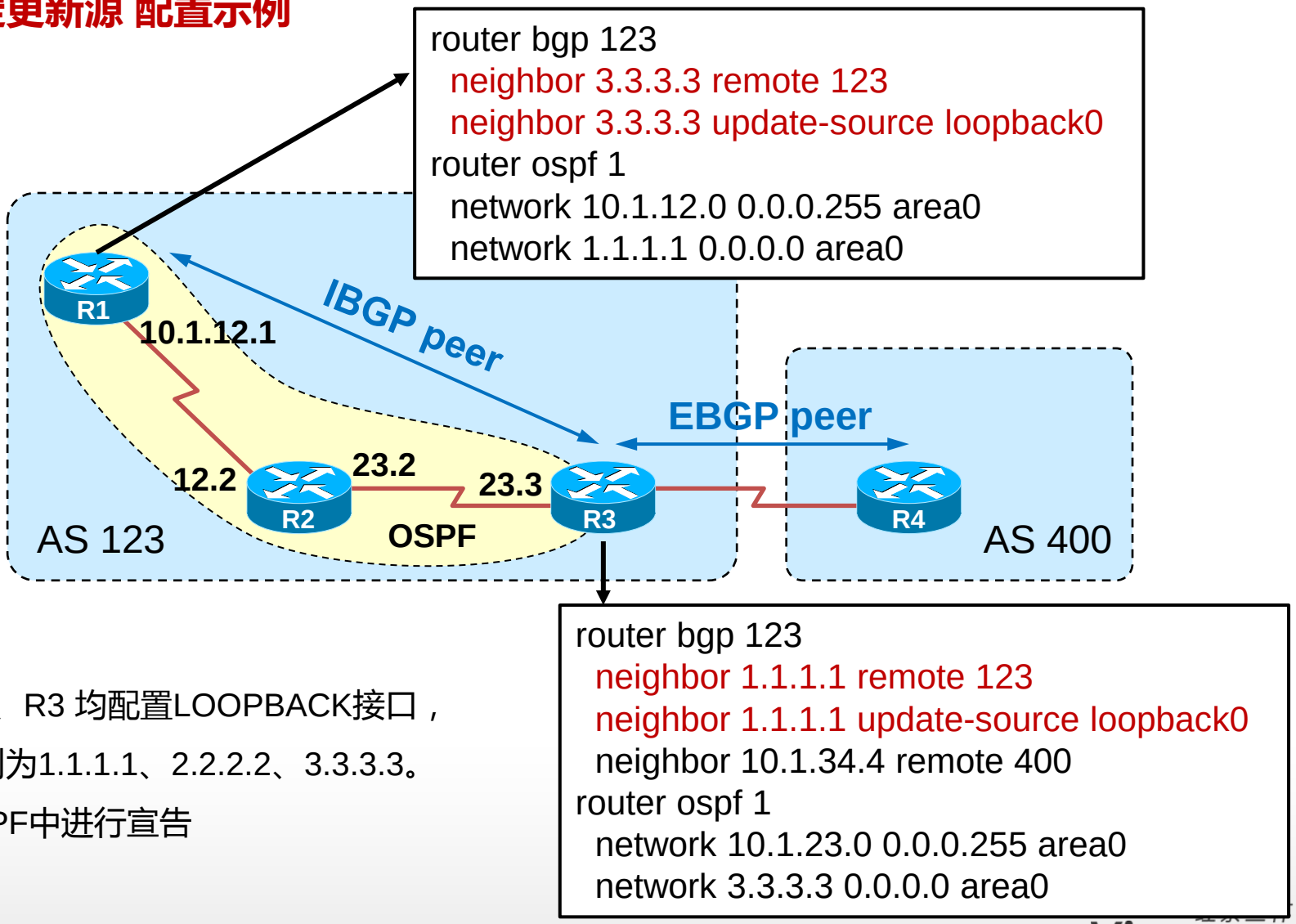
- **指定更新源**

Router(config-router)#neighbor *{ip-address | peer-group-name}* **update-source** *interface-type interface-number*

- 路由器指定环回接口地址用作到邻居的BGP连接的源地址，只要路由器在运行，loopback接口始终可用
- 当IBGP邻居之间有多条路径时，通常可使用环回接口地址建立IBGP会话
- EBGP邻居通常使用直连接口的IP地址作为源地址，如若使用环回接口建立EBGP则应注意EBGP邻居多跳问题。

BGP基本配置

- **指定更新源 配置示例**



R1、R2、R3 均配置LOOPBACK接口，
地址分别为1.1.1.1、2.2.2.2、3.3.3.3。
并在OSPF中进行宣告

BGP基本配置

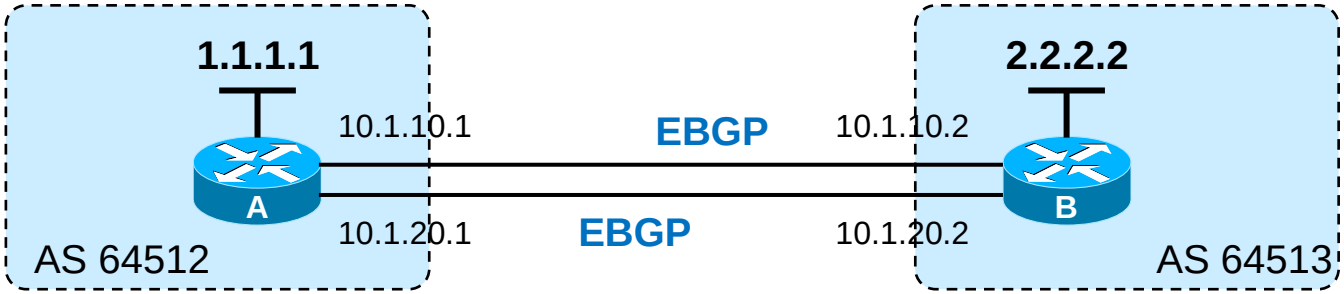
- **EBGP multihop**

Router(config-router)#neighbor {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]

- 建立对等关系时，如果不进行额外配置，EBGP路由器只能使用与外部EBGP路由器直接相连的接口地址
- 以上命令如若不显式指定跳数时，则为255跳

BGP基本配置

- **EBGP multihop 配置示例**




```
router bgp 64512
  neighbor 2.2.2.2 remote 64513
  neighbor 2.2.2.2 update-source loopback0
  neighbor 2.2.2.2 ebgp-multihop 2
ip route 2.2.2.2 255.255.255.255 10.1.10.2
ip route 2.2.2.2 255.255.255.255 10.1.20.2
```

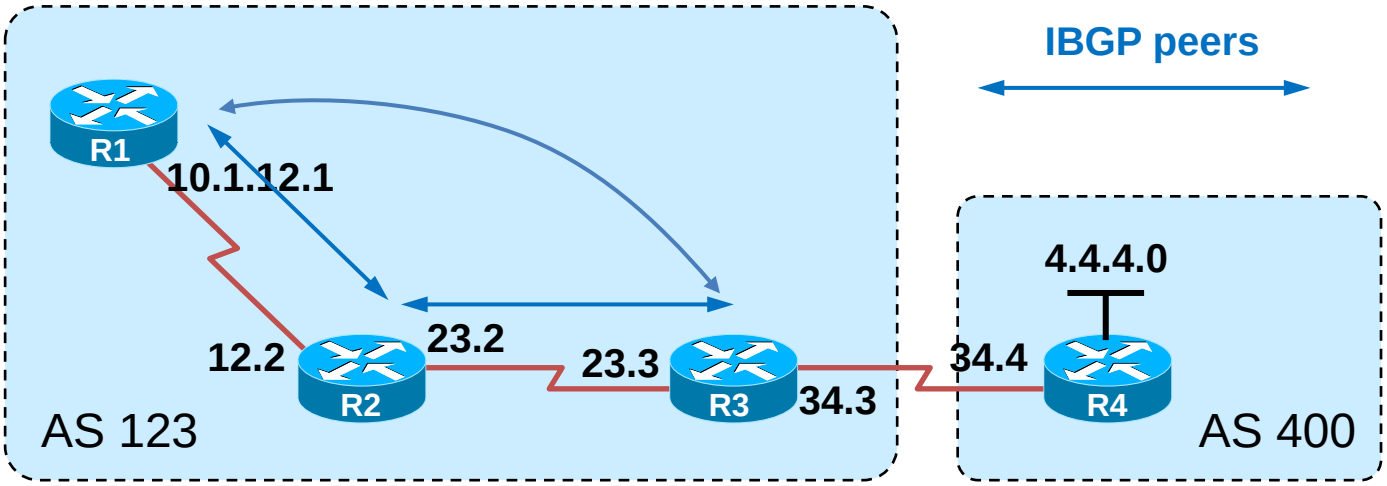
```
router bgp 64513
  neighbor 1.1.1.1 remote 64512
  neighbor 1.1.1.1 update-source loopback0
  neighbor 1.1.1.1 ebgp-multihop 2
ip route 1.1.1.1 255.255.255.255 10.1.10.1
ip route 1.1.1.1 255.255.255.255 10.1.20.1
```

BGP基本配置

- **修改next-hop**
  - BGP是AS-by-AS的路由协议，而不是router-by-router的路由协议
  - 在BGP中，next-hop并不意味着是下一台路由器，而是到达下一个AS的IP地址
  - EBGP中，默认next-hop为发送更新的邻居路由器的IP地址
  - IBGP中，从EBGP传来的next-hop属性在IBGP中保持不变的被传递

BGP基本配置

- **修改next-hop**



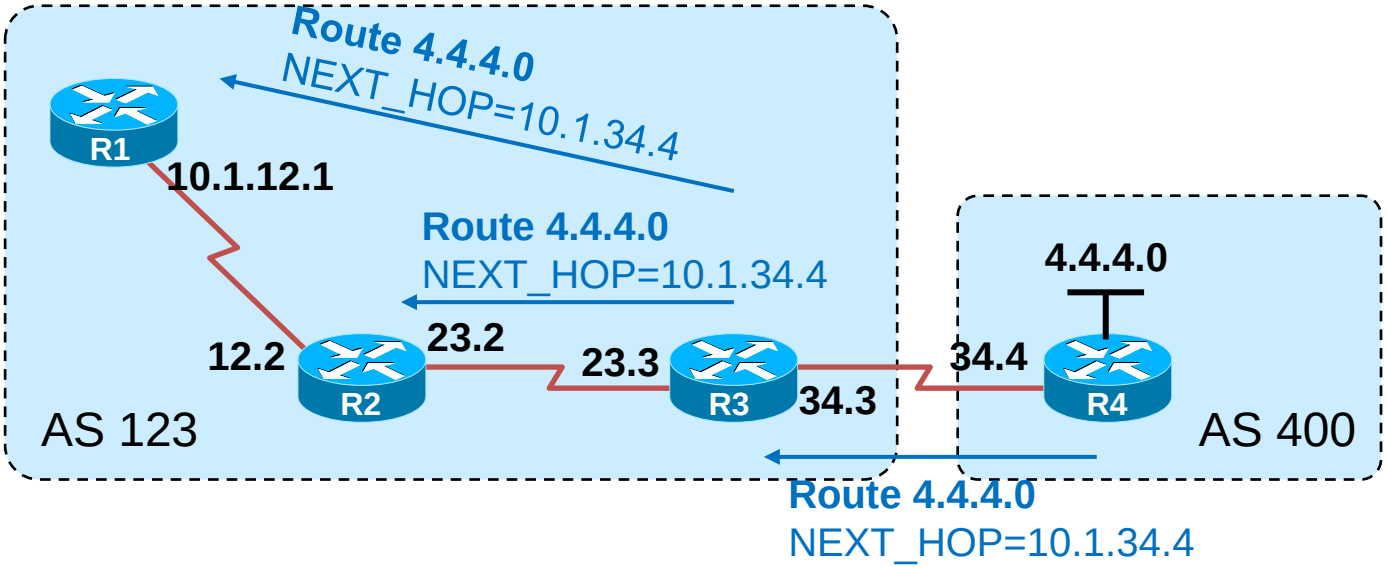
为了防止路由黑洞问题，R1、R2、R3建立IBGP全互联

且均用各自的LOOPBACK接口建立IBGP关系

此处OSPF部分省略不再赘述

BGP基本配置

- **修改next-hop**



R1、R2不知道如何去往10.1.34.4，因此路由不可达，也就不是best

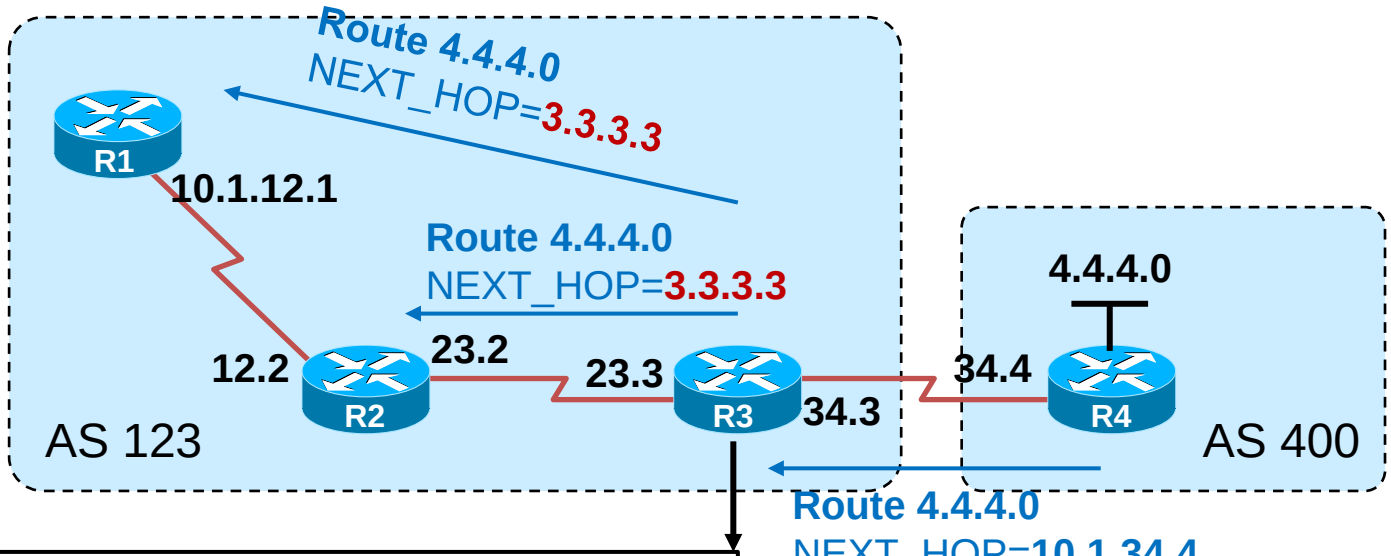
BGP基本配置

- **修改next-hop**

Router(config-router)#neighbor {*ip-address* | *peer-group-name*} **next-hop-self**

BGP基本配置

- **修改next-hop**




```
router bgp 123
  neighbor 1.1.1.1 remote 123
  neighbor 1.1.1.1 update-source loopback0
  neighbor 1.1.1.1 next-hop-self
  neighbor 2.2.2.2 remote 123
  neighbor 2.2.2.2.update-source loopback0
  neighbor 2.2.2.2 next-hop-self
```

BGP基本配置

- **BGP身份验证**

Router(config-router)#neighbor *{ip-address | peer-group-name}* **password** *string*

- BGP支持MD5邻居身份验证
- 启用身份验证后，将对通过对等体之间的TCP连接传输的所有数据等进行验证
- 认证都是在TCP建立连接的时候完成的!

BGP基本配置

- **维护BGP**
  - 重置BGP会话
  - 将新策略应用于所有路由，必须触发一个更新

  3种触发更新的方式
  - 硬重置
  - 软重置
  - 路由刷新

BGP基本配置

- **维护BGP**
  - 硬重置
    - 断开相应的TCP连接，通过这些会话收到的所有信息都将失效，并从BGP表中删除
    - clear ip bgp *{neighbor-address}*
    - clear ip bgp *

BGP基本配置

- **维护BGP**
  - 软重置（soft reconfiguration）
    - 不拆除并重建TCP或BGP连接，而是仅触发更新操作以便让新的路由策略生效
    - 软重置可以仅由于出站或入站策略，也可同时用于出入站策略

BGP基本配置

- **维护BGP**
  - 出站软重置
    - 不会拆除TCP连接，不会重置BGP会话，仅促发更新操作以便让新的路由策略生效（发送update消息）
    - 需要修改出站策略时，建议使用该命令
    - clear ip bgp soft out

BGP基本配置

- **维护BGP**
  - 入站软重置
    - 本地发送route-refresh给所有BGP邻居
    - clear ip bgp soft in

CISCO IOS 12.1开始全面支持入站路由的动态软重配置，但在之前的版本在使用入站软重配置之前必须首先在BGP进程中增加如下配置：

neighbor x.x.x.x soft-reconfiguration inbound

然后再使用clear ip bgp soft in命令

这条命令会将x.x.x.x邻居发送过来的BGP路由存储在内存中，当配置入站软重置后，路由器不再向邻居发送更新请求，而是直接在内存中存储的路由中执行新配置的入站策略，以此来防止触发大批量的路由更新而造成资源的浪费，但是这种操作仍会耗费内存，因此在使用的时候要非常慎重。

BGP基本配置

- **维护BGP**

发送给该邻居的最后一个BGP表版本号

每当BGP表发生变化就加1.

```
Router# show ip bgp summary
BGP router identifier 10.1.1.1, local AS number 65001
BGP table version is 124, main routing table version 124
9 network entries using 1053 bytes of memory
22 path entries using 1144 bytes of memory
12/5 BGP path/bestpath attribute entries using 1488 bytes of memory
6 BGP AS-PATH entries using 144 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3829 total bytes of memory
BGP activity 58/49 prefixes, 72/50 paths, scan interval 60 secs

Neighbor    V    AS    MsgRcvd    MsgSent    TblVer    InQ    OutQ    Up/Down    State/PfxRcd

10.1.1.0    4    64998    21    18    124    0    0    00:01:13    6
10.1.2.0    4    64999    11    10    124    0    0    00:01:11    6
```

state是bgp session的的状态，如果达到established就显示收到的路由的数目，否则就是active之类的状态。

BGP基本配置

- **维护BGP**

```
Router# show ip bgp
BGP table version is 2, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop            Metric     LocPrf     Weight     Path
*> 11.11.11.0/24 10.1.12.1           0          0                     64512 i
* i              10.1.23.3           0          0                     300 64512 i
```

BGP基本配置

- **维护BGP**

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 11.11.11.0/24	10.1.12.1	0	0		64512 i
* i	10.1.23.3	0	0		300 64512 i

第一栏的可能取值如下：

*　　可用的路由（但不一定是最优）

s　　被抑制的路由条目，例如做了路由汇总，抑制了明细

d　　被惩罚（dampening）的路由，路由受到了惩罚，虽该路由当前可能正常，但惩罚期结束前不会被通告

h　　被惩罚（dampening）的路由，路由可能出现了故障（down），有历史信息，但没有最佳路由

r　　路由没有被装载进RIB表，例如由于AD值等原因导致

S　　大写的S，stale，表示过期的路由

第二栏　　>　　BGP算法选出的最优路径

第三栏　　为空，或为i。为空表示该路由从EBGP邻居获取，为i表示这是从IBGP学习到的路由

BGP基本配置

- **维护BGP**

```
R5#sh ip b 30.30.30.0
BGP routing table entry for 30.30.30.0/24, version 2
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Flag: 0x820
  Not advertised to any peer
  Local
    3.3.3.3 (metric 65) from 4.4.4.4 (4.4.4.4)
      Origin IGP, metric 0, localpref 100, valid, internal
      Originator: 3.3.3.3, Cluster list: 4.4.4.4
  Local
    3.3.3.3 (metric 65) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

BGP基本配置

- **维护BGP**
  - 查看
    - show ip bgp neighbor {*address*} received-routes
    - show ip bgp neighbors {*address*} routes
    - show ip bgp neighbors {*address*} advertised-routes

红茶三杯
Vinsoney

BGP路径属性

BGP路径属性

- **属性分类**

  公认属性 Well-Known

  – 公认强制属性 Well-known mandatory

  – 公认自由决定属性 Well-known discretionary

  可选属性 Optional

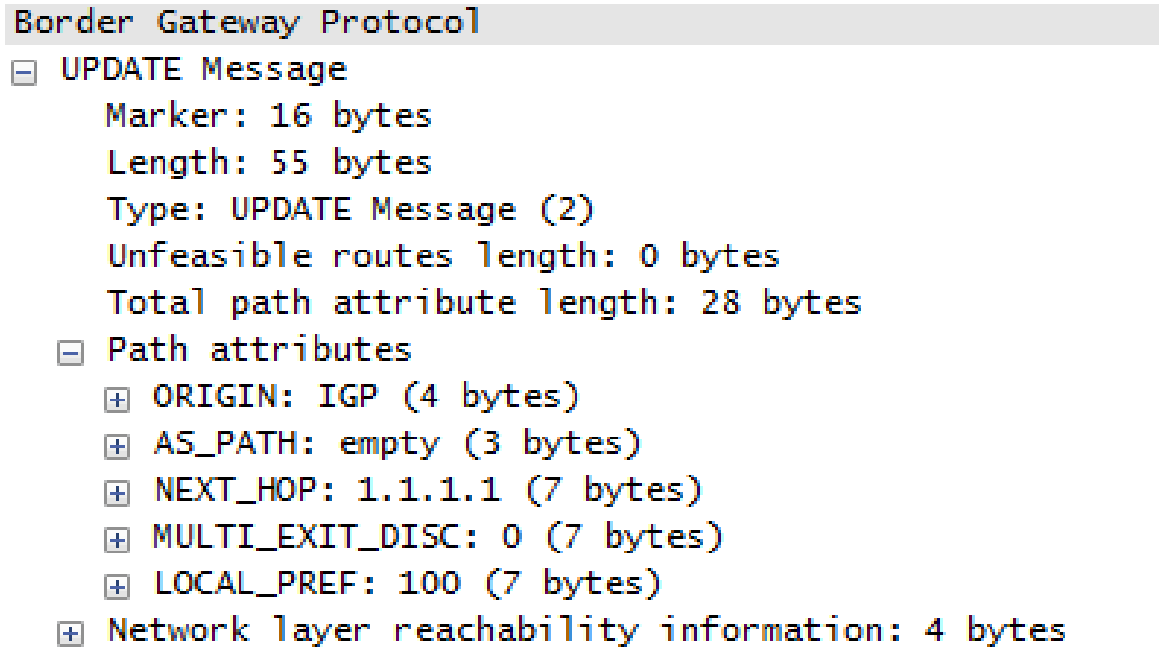
  – 可选传递的 Optional non-transitive

  – 可选非传递的 Optional non-transitive

BGP路径属性

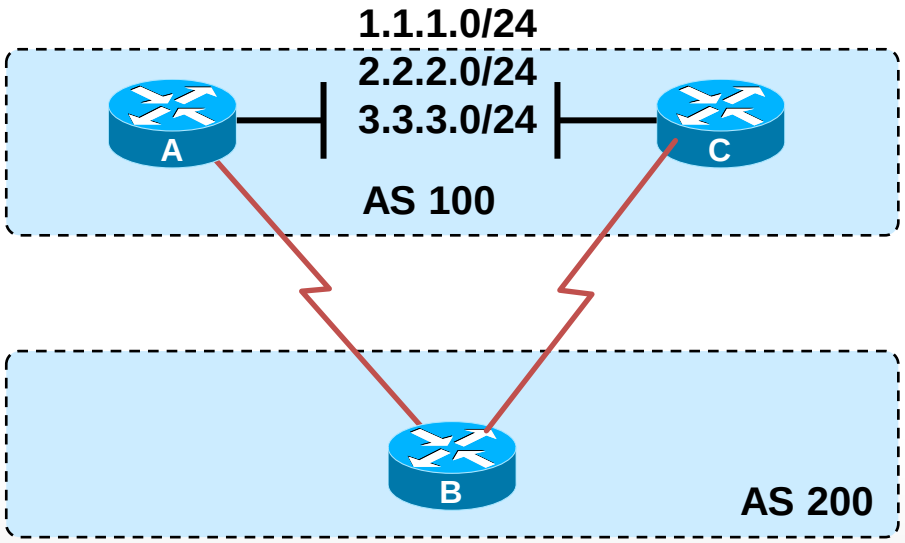
公认属性	**公认必遵**	BGP 必须都能识别，且在更新消息必须包含	Origin AS-Path Next hop
	公认自决	BGP 必须都能识别，更新消息可包含可不包含	Local-Preference ATOMIC_Aggregate
可选属性	**可选传递**	可以不支持该属性，但即使不支持也应当接受包含该属性的路由并传递给其他邻居	Community Aggregator
	可选非传递	可以不支持该属性，不识别的BGP进程忽略包含这个属性的更新消息并且不传递给其他BGP邻居	MED Originator_ID Cluster_list *Weight

BGP路径属性


Border Gateway Protocol
UPDATE Message
Marker: 16 bytes
Length: 55 bytes
Type: UPDATE Message (2)
Unfeasible routes length: 0 bytes
Total path attribute length: 28 bytes
Path attributes
ORIGIN: IGP (4 bytes)
AS_PATH: empty (3 bytes)
NEXT_HOP: 1.1.1.1 (7 bytes)
MULTI_EXIT_DISC: 0 (7 bytes)
LOCAL_PREF: 100 (7 bytes)
Network layer reachability information: 4 bytes

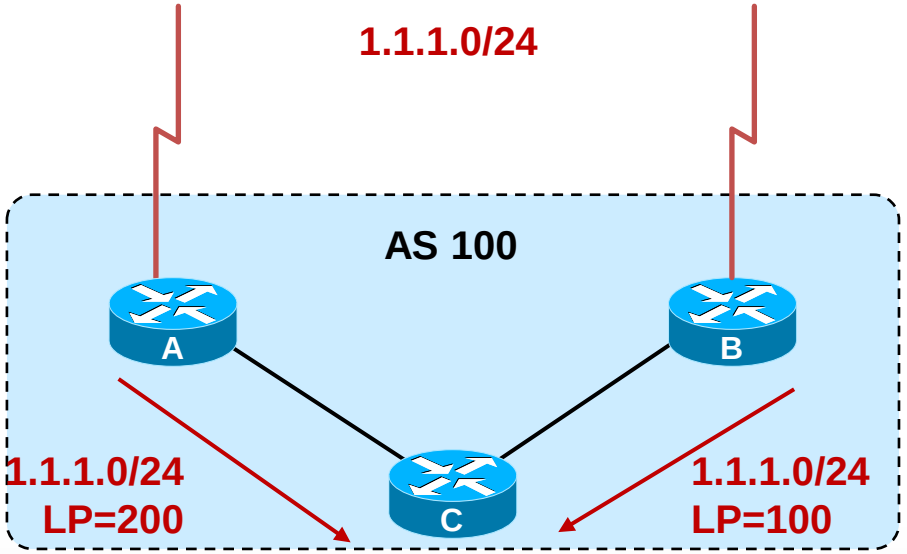
WEIGHT

- 在路由器本地配置，只提供本地路由策略，不会传播给任何BGP邻居
- 范围：0~65535；越大越优先
- 路由器本地始发的路径默认权重为32768，从其他BGP邻居学习到的为0



LOCAL PREFERENCE

- **公认自由决定属性**
- **告诉AS中的路由器，哪条路径是离开AS的首选路径**
- **LP越高路径越优**
- **只发送给IBGP邻居，而不能传递给EBGP邻居**
- **默认本地优先级为100**



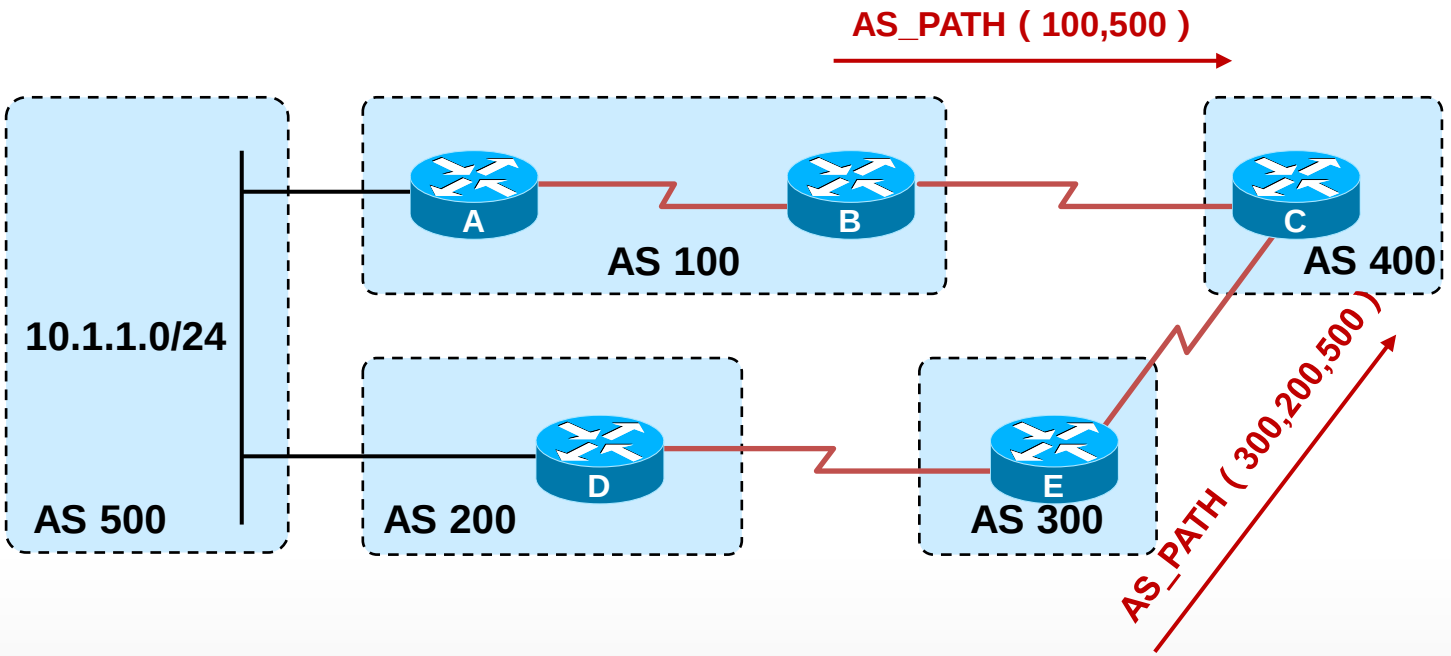
LOCAL PREFERENCE

- **注意事项**

1. 只能在IBGP Peer之间传递（除非做了策略否则LP值在AS内的IBGP邻居间传递不会丢失），不能在EBGP Peer之间传递，如果在EBGP Peer之间收到的路由的路径属性中携带了Local Preference，则会触发Notifacation报文,造成会话中断；但是可以再AS 边界路由器上使用IN方向的策略。
2. bgp default local-preference 500 //默认lp值
3. BGP路由器在向其EBGP邻居发送路由更新时，不能携带LP属性，对方收到该EBGP路由的LP值为空（连LP这个字段都木有），但是它会在本地为这条路由赋一个默认值，也就是100，然后再传递给自己的IBGP
4. 本地network及重发布的路由，LP默认100，并能在AS内向其他IBGP邻居传输，传输过程中除非部署策略，否则LP不变

AS_PATH

- **公认强制属性**
- **是前往目标网络的路由经过的自制系统号列表，通告该路由的自治系统号位于列表末尾**
- **作用：确保无环，通告给EBGP时会加上自己的AS号；通告给IBGP时不修改AS-path**

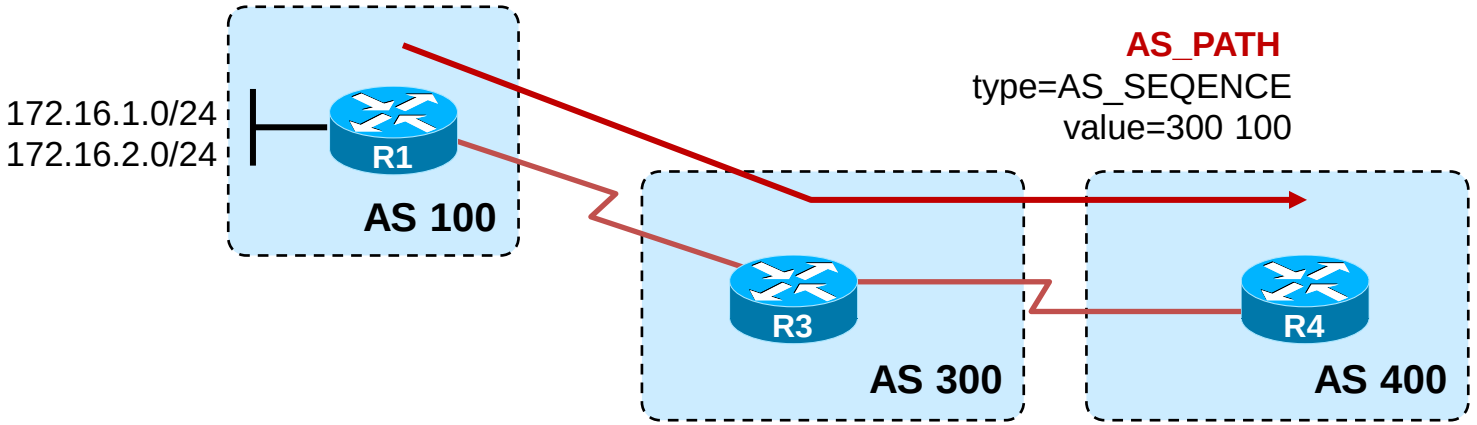


AS_PATH

- **四种类型**
  - AS_SET：一个去往特定目的地所经路径上的无序AS号列表
  - AS_SEQENCE：一个有序的AS号列表
  - AS_CONFED_SEQUENCE：一个去往特定目的地所经路径上的有序AS号列表，其用法与AS_SEQUENCE完全一样，区别在于该列表中的AS号属于本地联邦中的AS
  - AS_CONFED_SET：一个去往特定目的地所经路径上的无序AS号列表，去用方法与AS_SET完全一样，区别在于列表中的AS号属于本地联邦中的AS

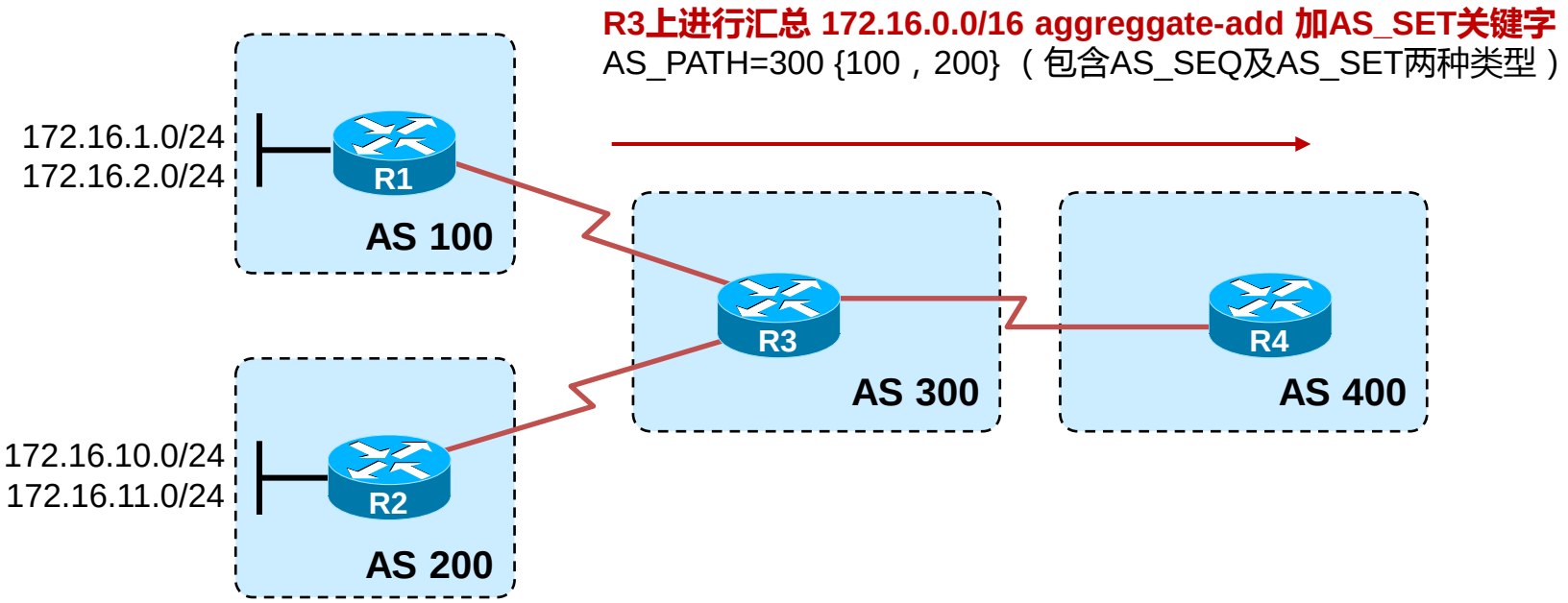
AS_PATH

- **四种类型**



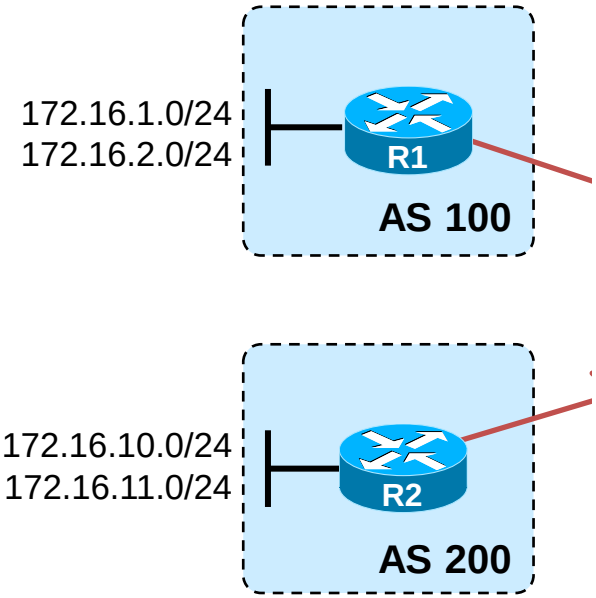
AS_PATH

- **四种类型**



AS_PATH

- **四种类型**




```
Border Gateway Protocol
⊟ UPDATE Message
    Marker: 16 bytes
    Length: 66 bytes
    Type: UPDATE Message (2)
    Unfeasible routes length: 0 bytes
    Total path attribute length: 40 bytes
  ⊟ Path attributes
    ⊞ ORIGIN: IGP (4 bytes)
    ⊟ AS_PATH: 300 {100, 200} (13 bytes)
      ⊞ Flags: 0x40 (Well-known, Transitive, Complete)
        Type code: AS_PATH (2)
        Length: 10 bytes
      ⊟ AS path: 300 {100, 200}
        ⊟ AS path segment: 300
            Path segment type: AS_SEQUENCE (2)
            Path segment length: 1 AS
            Path segment value: 300
        ⊟ AS path segment: {100, 200}
            Path segment type: AS_SET (1)
            Path segment length: 2 ASs
            Path segment value: 100 200
    ⊞ NEXT_HOP: 10.1.34.3 (7 bytes)
    ⊞ MULTI_EXIT_DISC: 0 (7 bytes)
    ⊞ AGGREGATOR: AS: 300 origin: 10.1.34.3 (9 bytes)
  ⊟ Network layer reachability information: 3 bytes
    ⊞ 172.16.0.0/16
```

AS_SEQ

AS_SET，是无序的

ORIGIN

- **公认强制属性**
- **标识路由的起源，有下列3种可能：**
  - i　　通过BGP network，也就是起源于IGP，因为BGP network必须保证该网络在路由表中
  - e　　是由 EGP 这种早期的协议重发布而来
  - ?　　Incomplete，从其他渠道学习到的，路由来源不完全（确认该路由来源的信息不完全）。 (重发布的路由)

  - 路由优选顺序： lowest origin code (IGP > EGP > Incomplete)

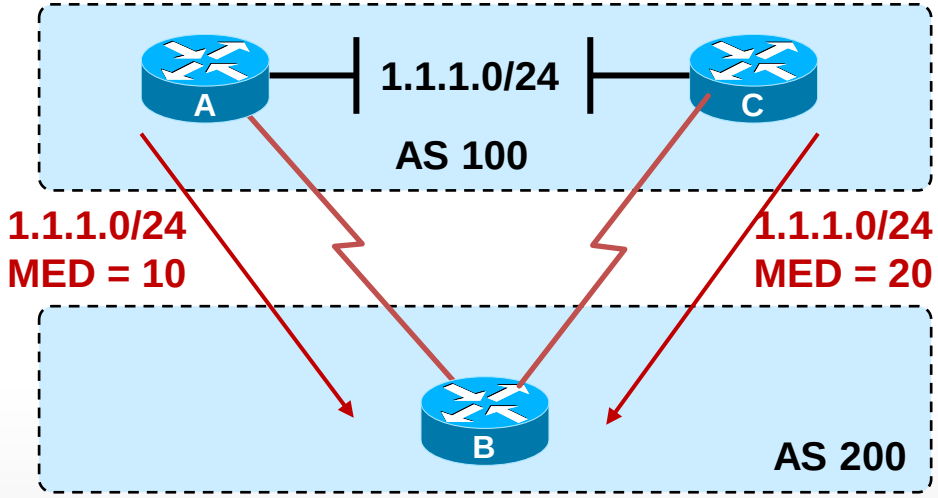
ORIGIN

- **Show ip bgp**

```
RouterA# show ip bgp
BGP table version is 14, local router ID is 172.31.11.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.0.0/24      0.0.0.0                  0         32768 i
* i                 10.1.0.2                 0    100      0 i
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
*>i10.1.2.0/24      10.1.0.2                 0    100      0 i
*> 10.97.97.0/24    172.31.1.3                             0 64998 64997 i
*                   172.31.11.4                            0 64999 64997 i
* i                 172.31.11.4              0    100      0 64999 64997 i
*> 10.254.0.0/24    172.31.1.3               0             0 64998 i
*                   172.31.11.4                            0 64999 64998 i
* i                 172.31.1.3               0    100      0 64998 i
r> 172.31.1.0/24    172.31.1.3               0             0 64998 i
r                   172.31.11.4                            0 64999 64998 i
r i                 172.31.1.3               0    100      0 64998 i
*> 172.31.2.0/24    172.31.1.3               0             0 64998 i
<output omitted>
```

MED

- **可选非传递属性**
- **是一种度量值，用于向外部邻居指出进入AS的首选路径，即当入口有多个时，自治系统可以使用MED动态的影响其他AS如何选择进入路径**
- **度量值越小路径越优**
- **MED是在AS之间交换，MED发送给EBGP对等体，这些路由器在AS内传播MED，不传递给下一个AS**



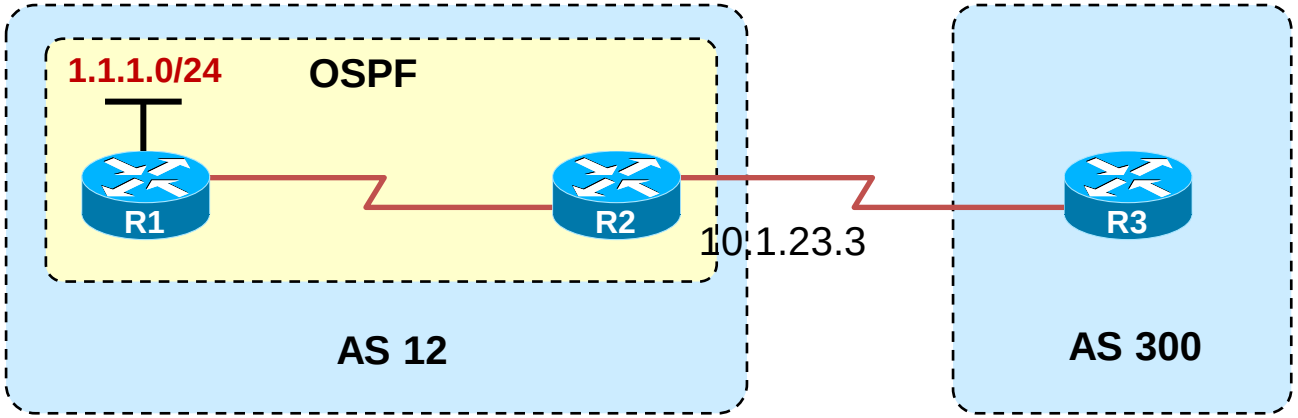
MED

- **默认情况下，仅当路径来自同一个AS中的不同EBGP邻居时，路由器才比较他们的MED属性**
- **MED影响进入AS的数据流；LP影响离开AS的数据流**

比较原则及配置注意事项

- 本地在将一条BGP路由通告给EBGP Peer时，是否携带MED值,需要根据以下条件进行判断(不对EBGP Peer使用Route-map的情况下)：
  - 如果该BGP路由是本地始发(network或redistribute)的,则携带MED值发送给EBGP Peer (如果MED为空,则设置为0)
  - 如果该BGP路由是从其他BGP Peer学习过来的，那么将该路由通告给EBGP Peer时不携带MED
- 本地在将一条BGP路由通告给IBGP Peer时，一定会携带MED值
  - 如果接收或产生的路由的MED为空,那么在向IBGP Peer通告时,将MED设置为0
- 总结1) 2) 两点就是MED在IBGP之间传递没有问题（不会丢失），但是在EBGP之间传递要看路由是否起源于自己。

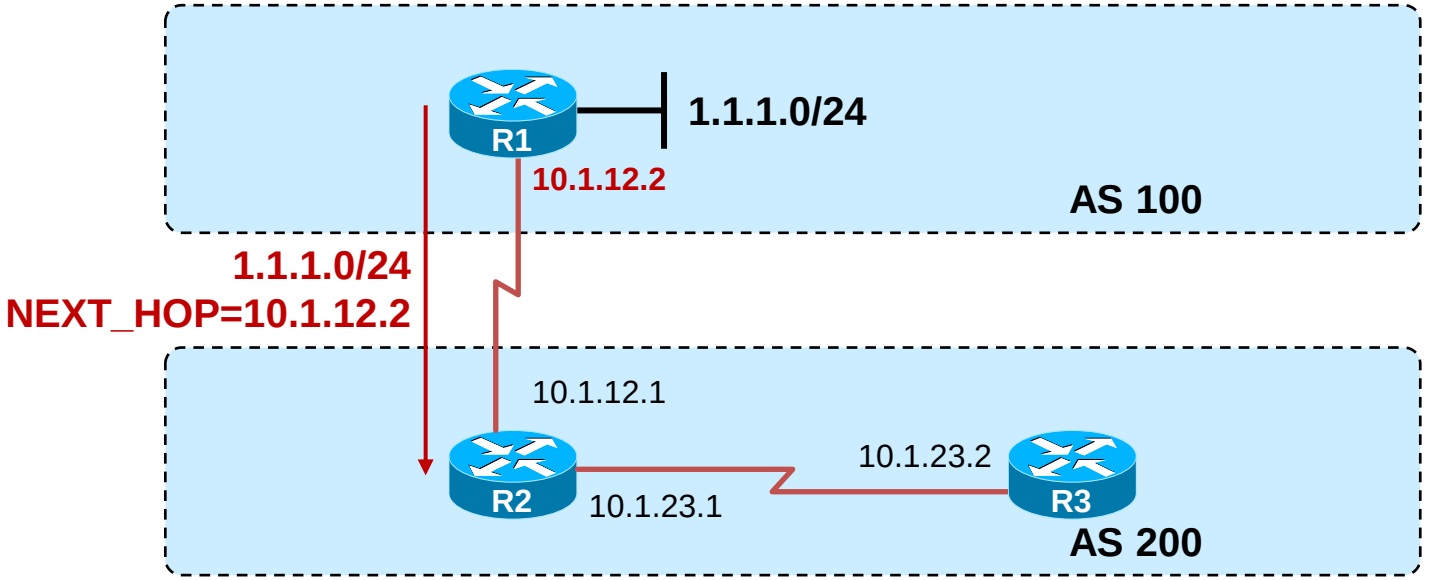
MED



- network本地从IGP路由协议学习到的路由进BGP，MED值继承IGP协议中的metric
- network本地直连接口的网段进BGP，MED值为0；
  network本地静态路由进BGP，MED值为0
- redistribute本地从IGP路由协议学习到的路由进BGP，MED值继承IGP协议中的metric
- redistribute本地直连接口网段进BGP，MED值为0
  redistribute本地静态路由进BGP，MED值为0

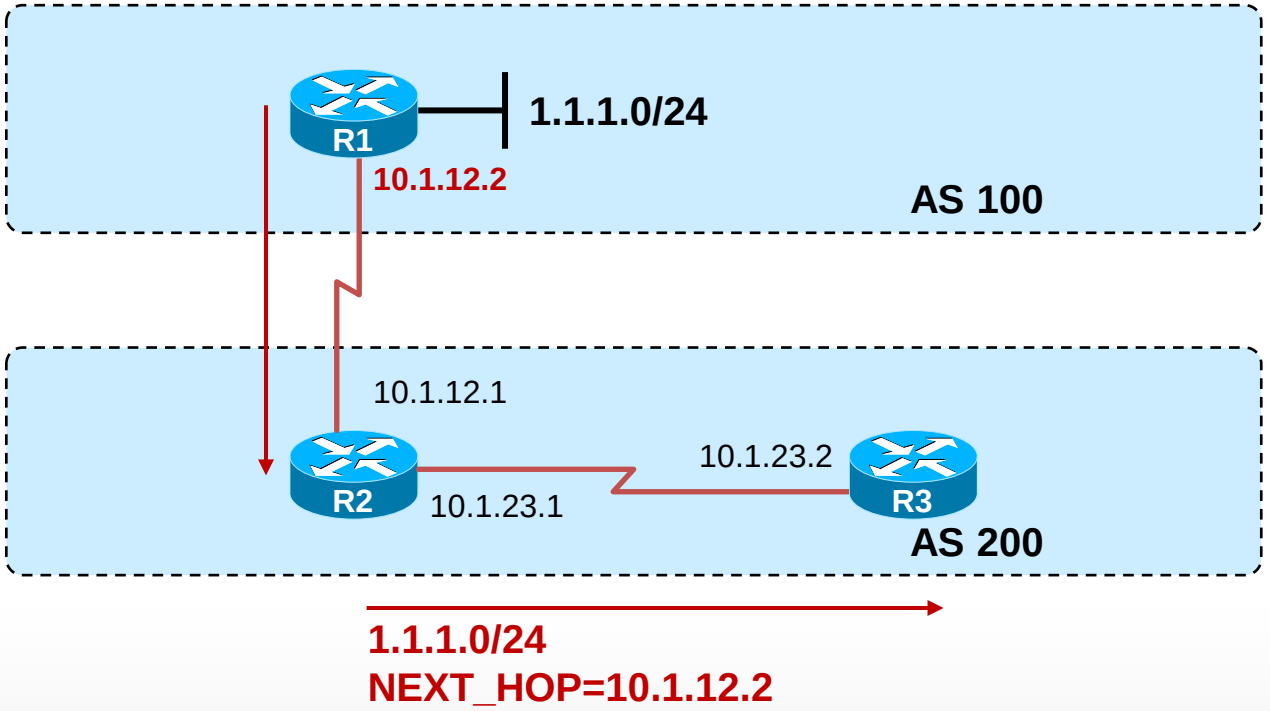
NEXT_HOP

- 如果路由传递自EBGP peer



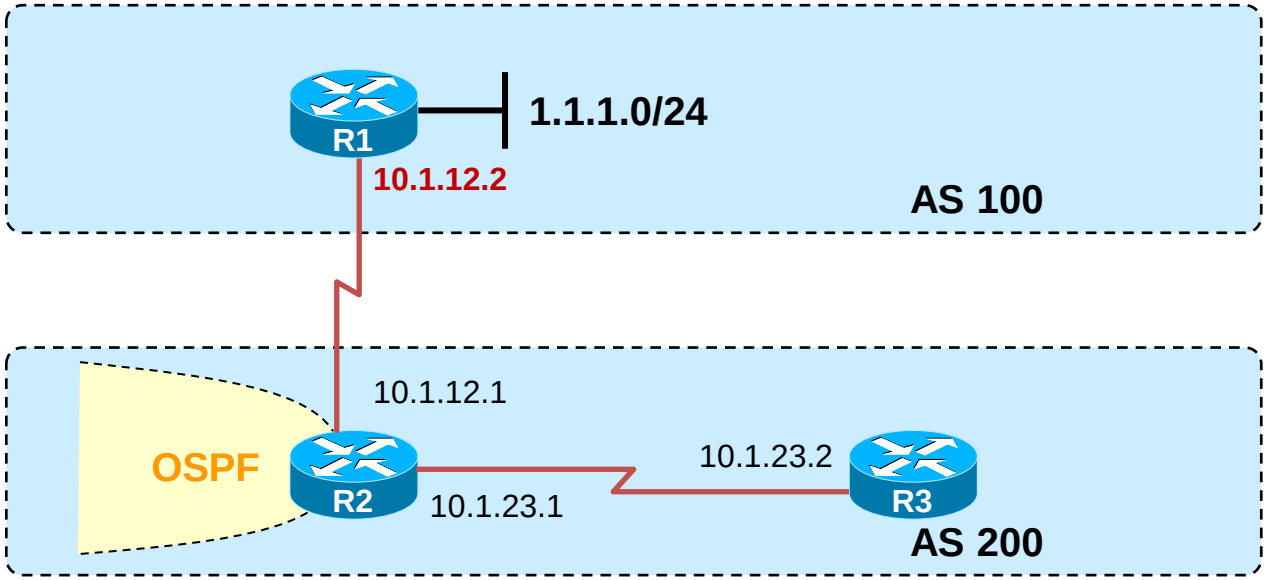
NEXT_HOP

- 如果路由传递自IBGP邻居，并描述的是AS外的目的地



NEXT_HOP

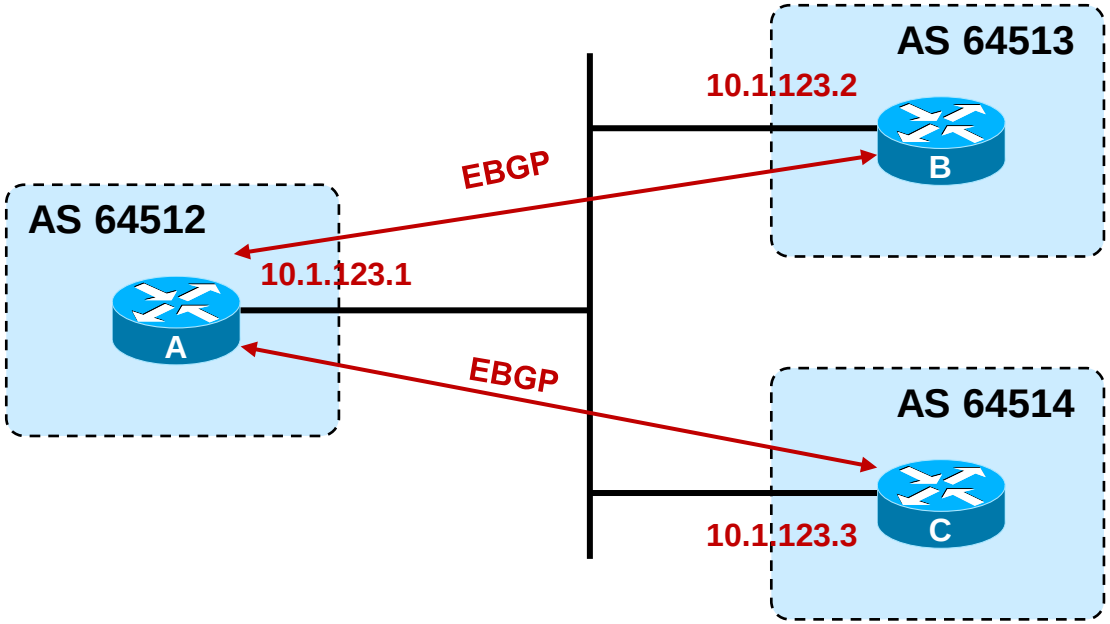
- 如果路由传递自IBGP邻居，并由AS内BGP路由器引入



如果IBGP peer使用network或重发布的方式引入IGP路由，那么通告者将使用这些路由的IGP下一跳作为NH；如果这些路由是该IBGP peer配置的BGP汇总路由，则NH为其更新源IP。

NEXT_HOP

- **NEXT_HOP on shared Media**



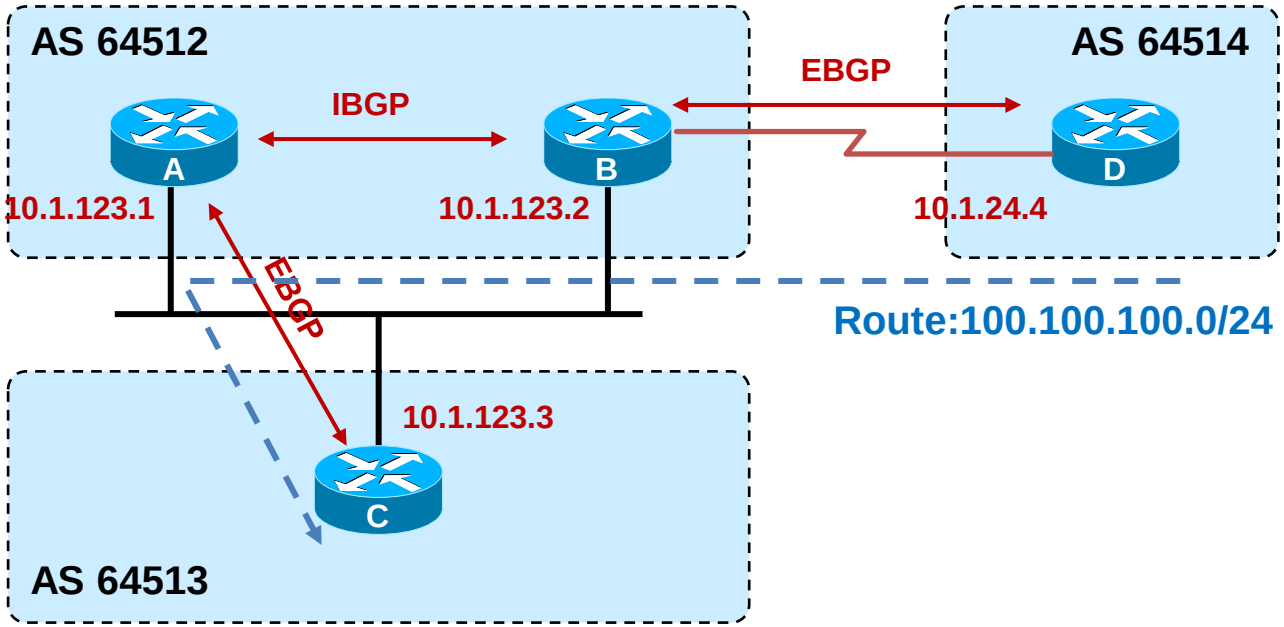
RouterB将路由100.100.100.0/24传递给A，NEXT_HOP为10.1.123.2；

RouterA将路由100.100.100.0/24传递给C，此时NEXT_HOP保持不变；

如果路由器收到某条BGP路由，该路由的NEXT_HOP地址值与EBGP邻居（更新对象）同属一个网段，那么该条路由的NEXT_HOP地址将保持不变并传递给它的BGP邻居

NEXT_HOP

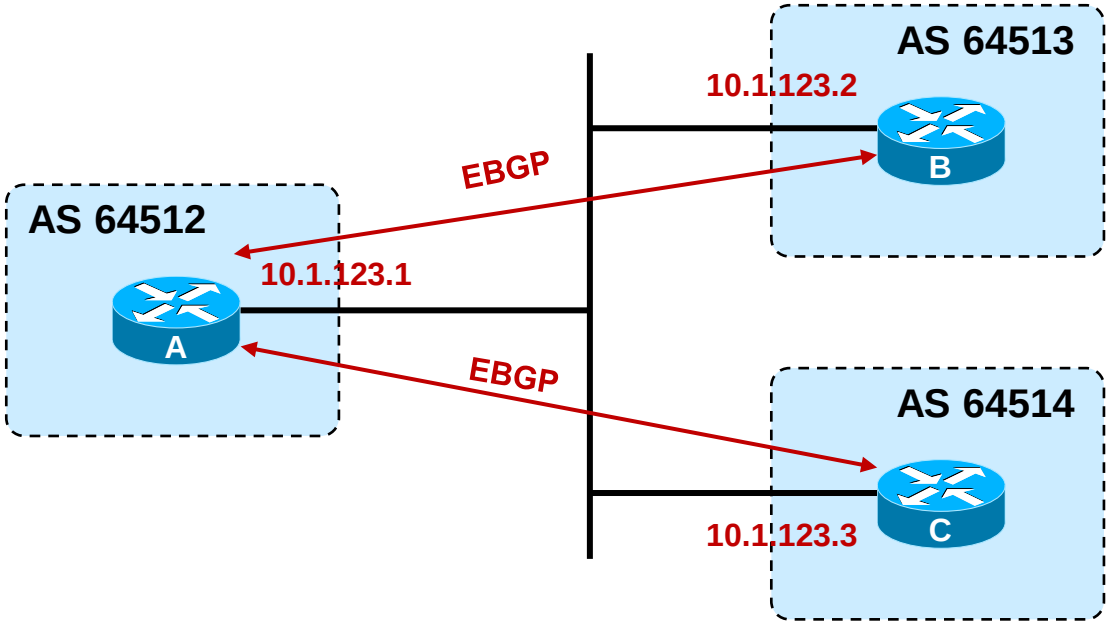
- NEXT_HOP 的另一个例子



Router C 上关于100.100.100.0/24的路由，NEXT_HOP属性为10.1.123.2

NEXT_HOP

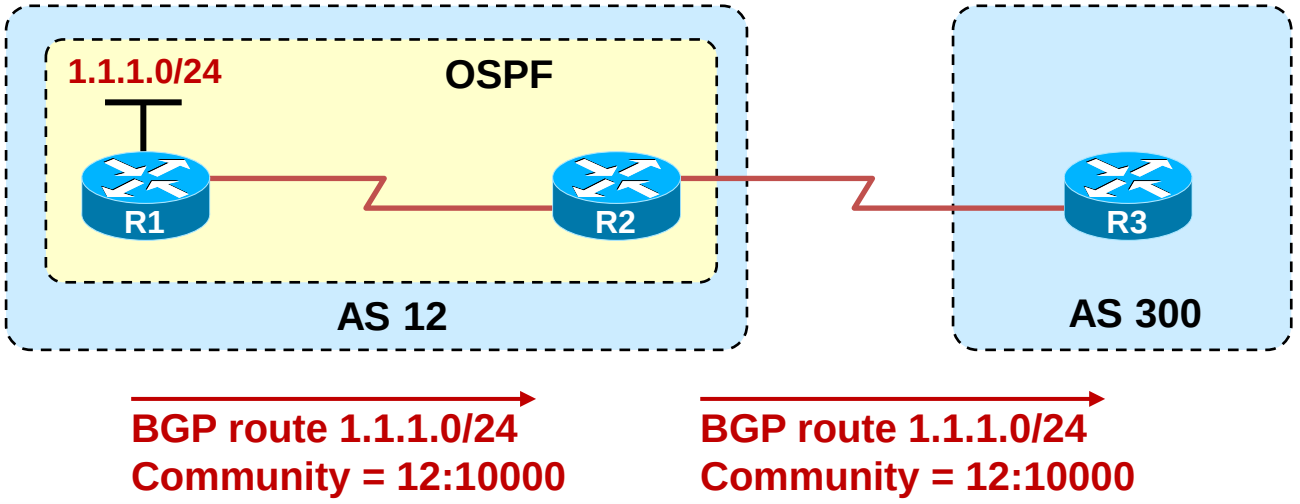
- **NEXT_HOP on NBMA**



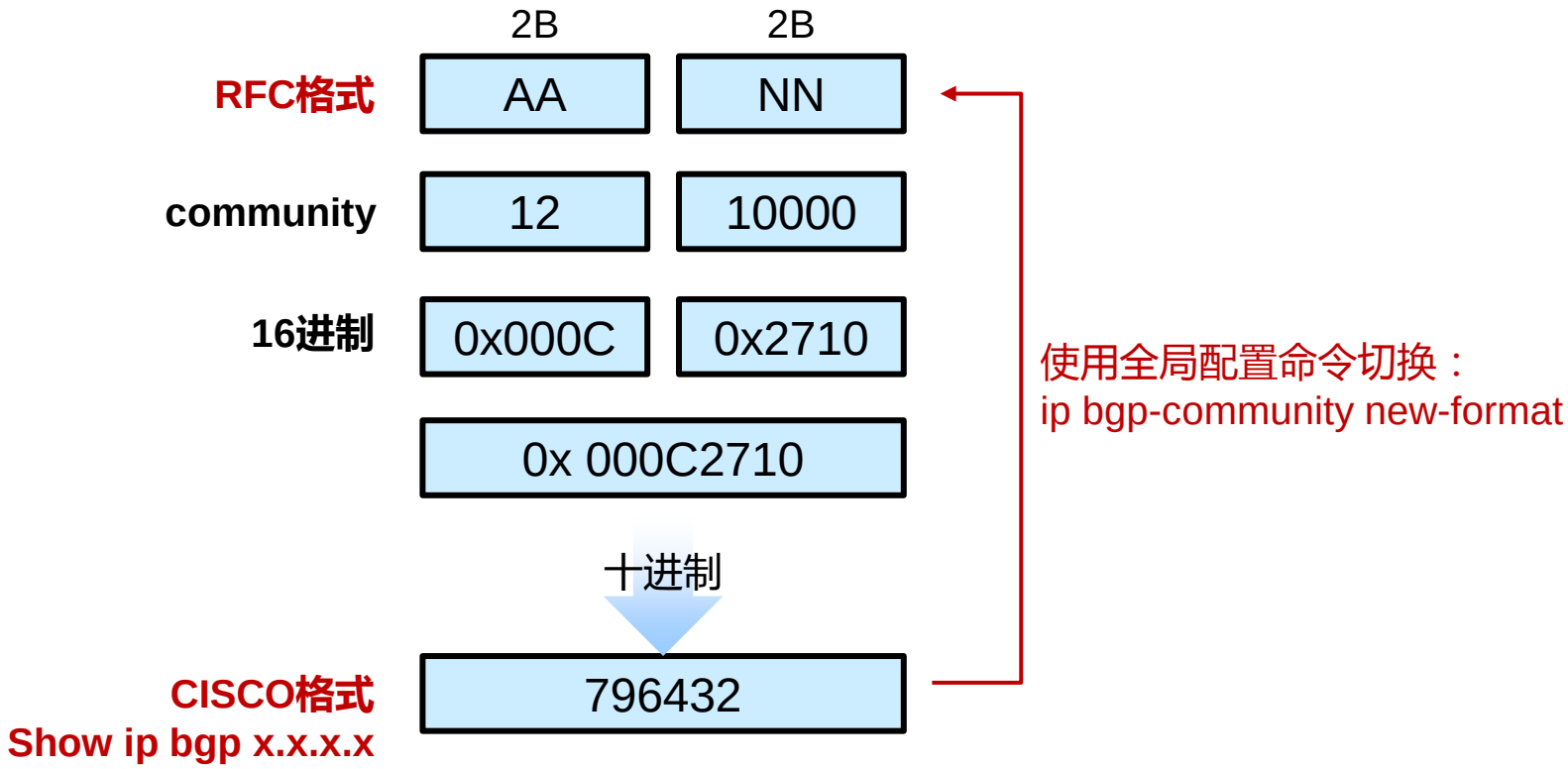
如果中间的多路访问网络不是Ethernet，而是帧中继（NBMA）呢？

COMMUNITY

- **可选传递属性**
- **一种标记，用于简化路由策略的执行**
- **可以将某些路由分配一个特定的COMMUNITY属性，之后就可以基于COMMUNITY值而不是每条路由进行BGP属性的设置了**



COMMUNITY



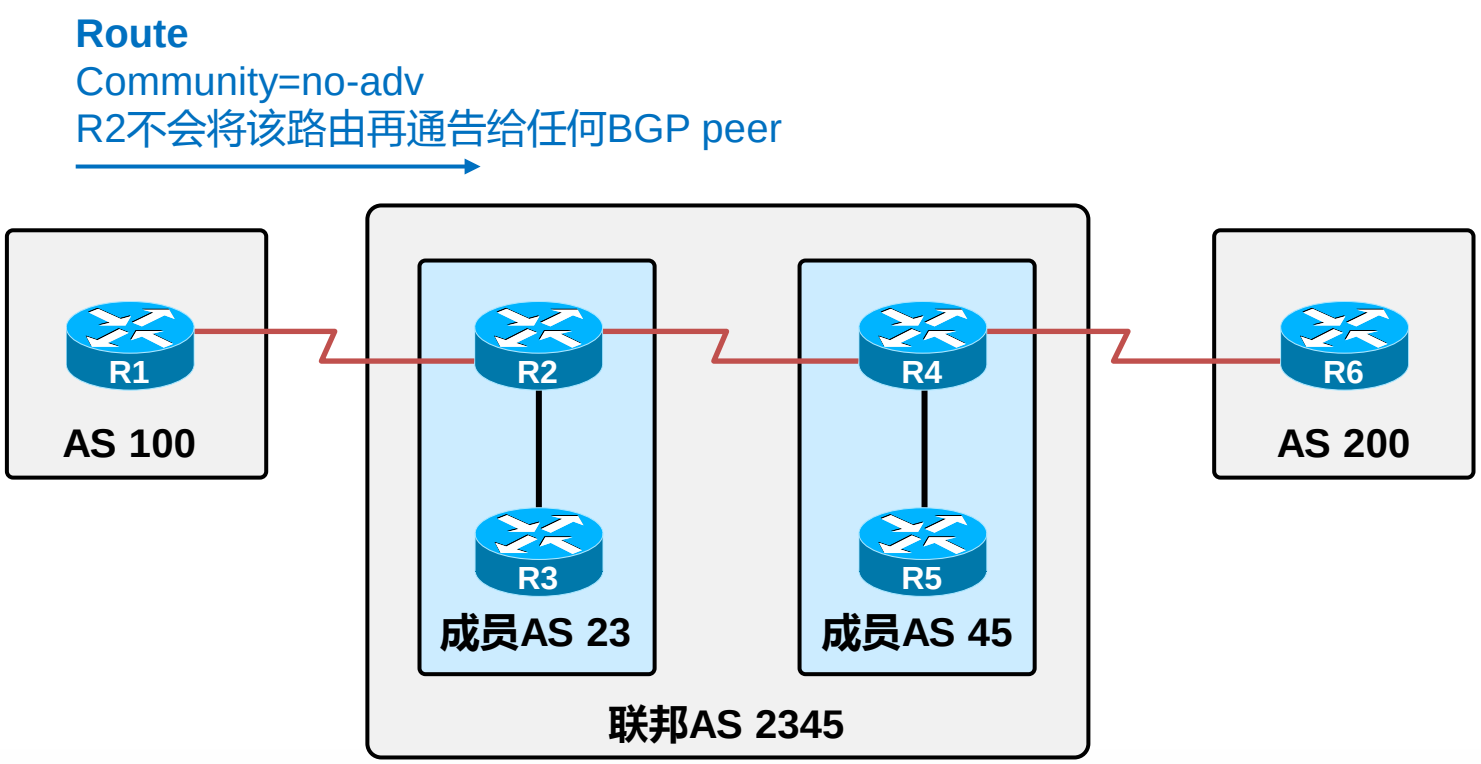
COMMUNITY

- **几个众所周知的值**

```
route-map test permit 10
set community ?
  <1-4294967295>    community number
  aa:nn             community number in aa:nn format
  additive          Add to the existing community
  internet          Internet (well-known community)
  local-AS          Do not send outside local AS (well-known community)
  no-advertise      Do not advertise to any peer (well-known community)
  no-export         Do not export to next AS (well-known community)
  none              No community attribute
```

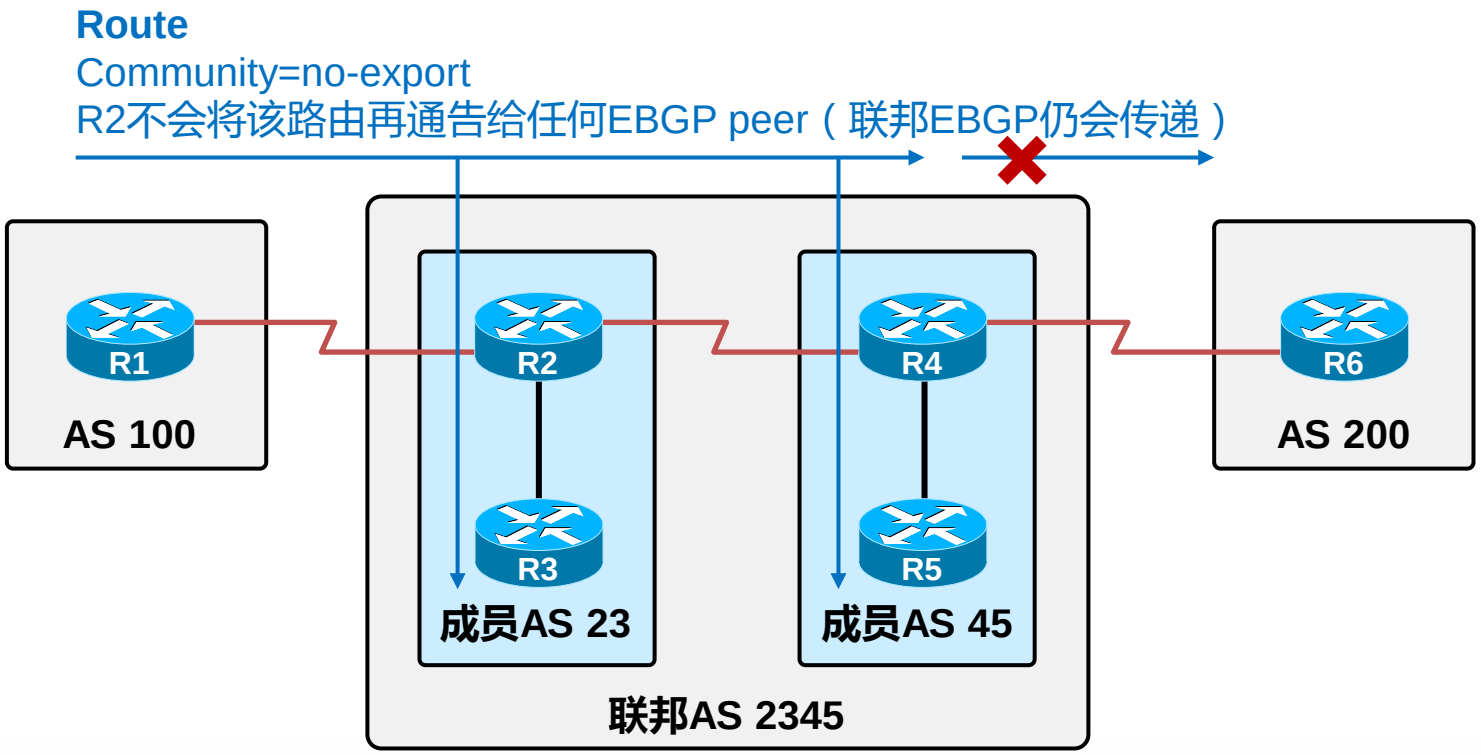
COMMUNITY

- **no-advertise**



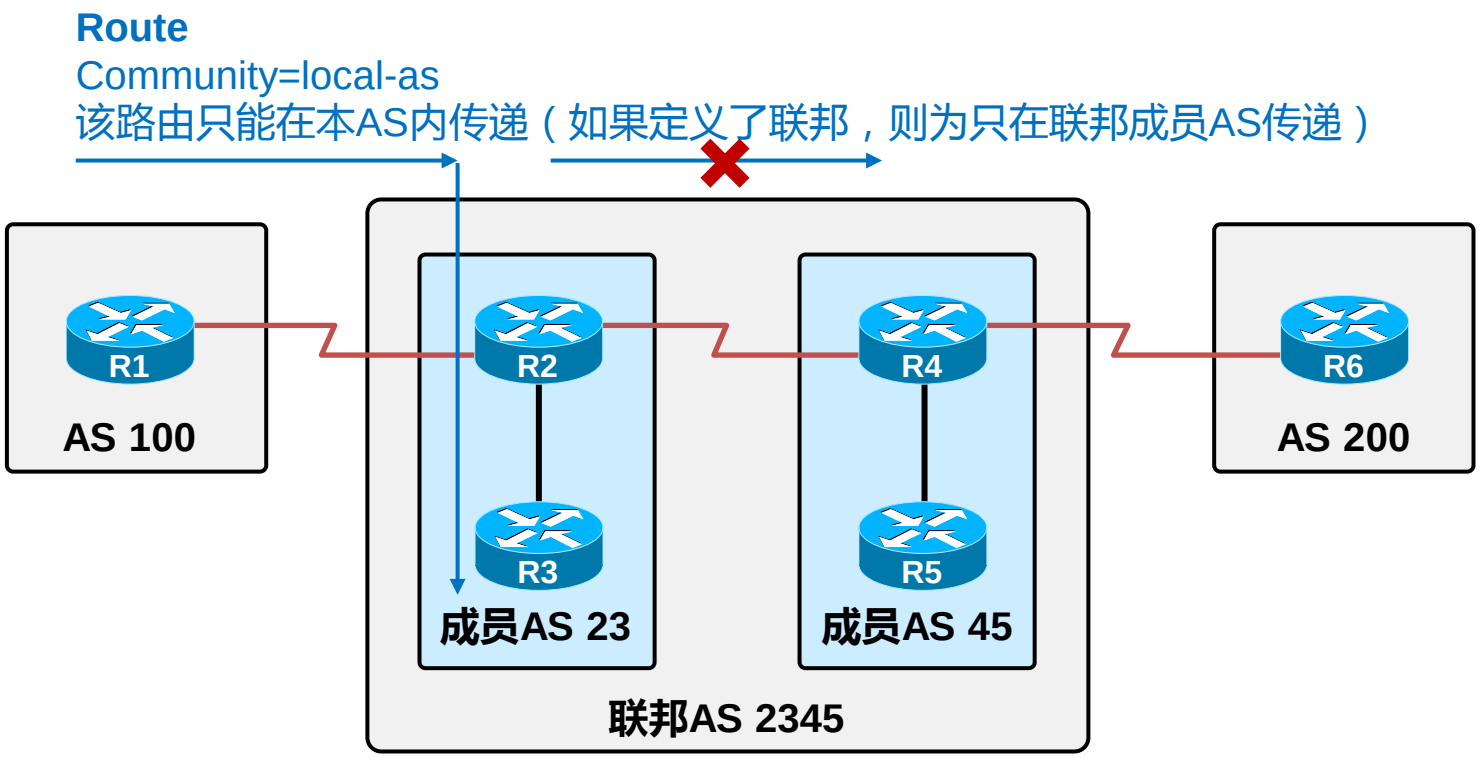
BGP属性

- **no-export**

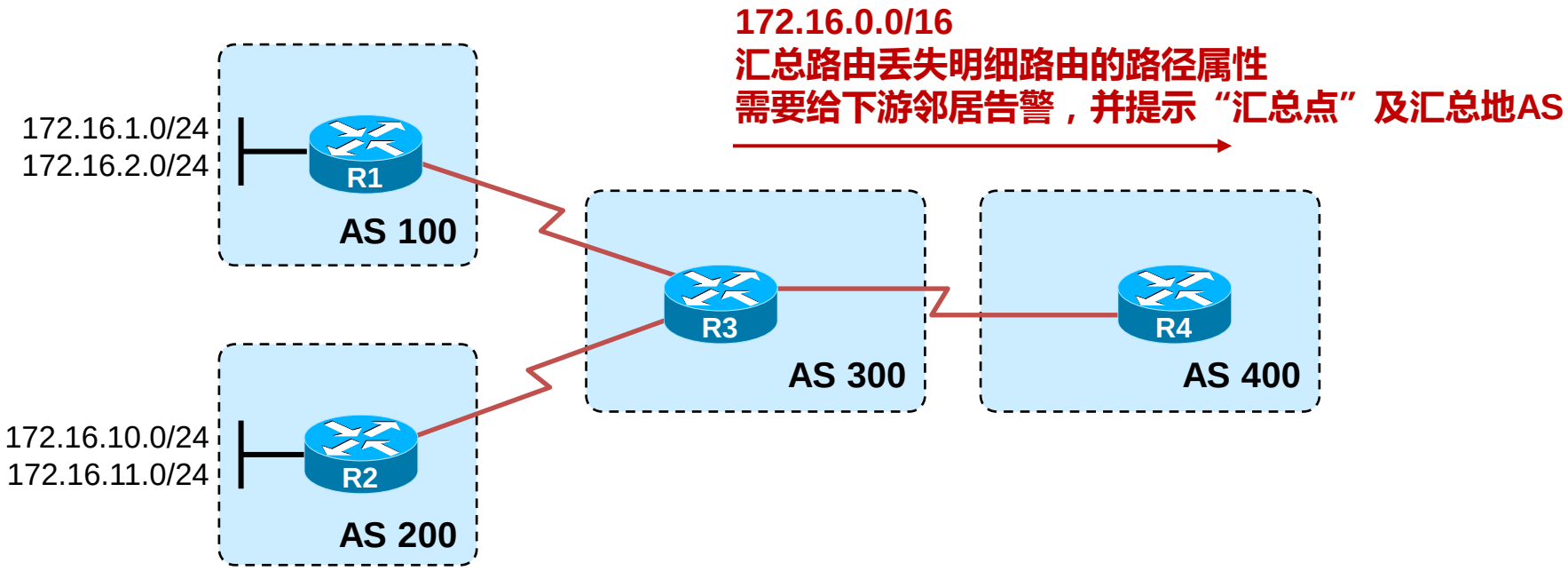


BGP属性

- **local-as**



Atomic_Aggregate及aggregator



aggregate-address 172.16.0.0 255.255.0.0 **summary-only**

Atomic_Aggregate及aggregator

R4#show ip bgp 172.16.0.0

BGP routing table entry for 172.16.0.0/16, version 4

Paths: (1 available, best #1, table Default-IP-Routing-Table)

Flag: 0x820

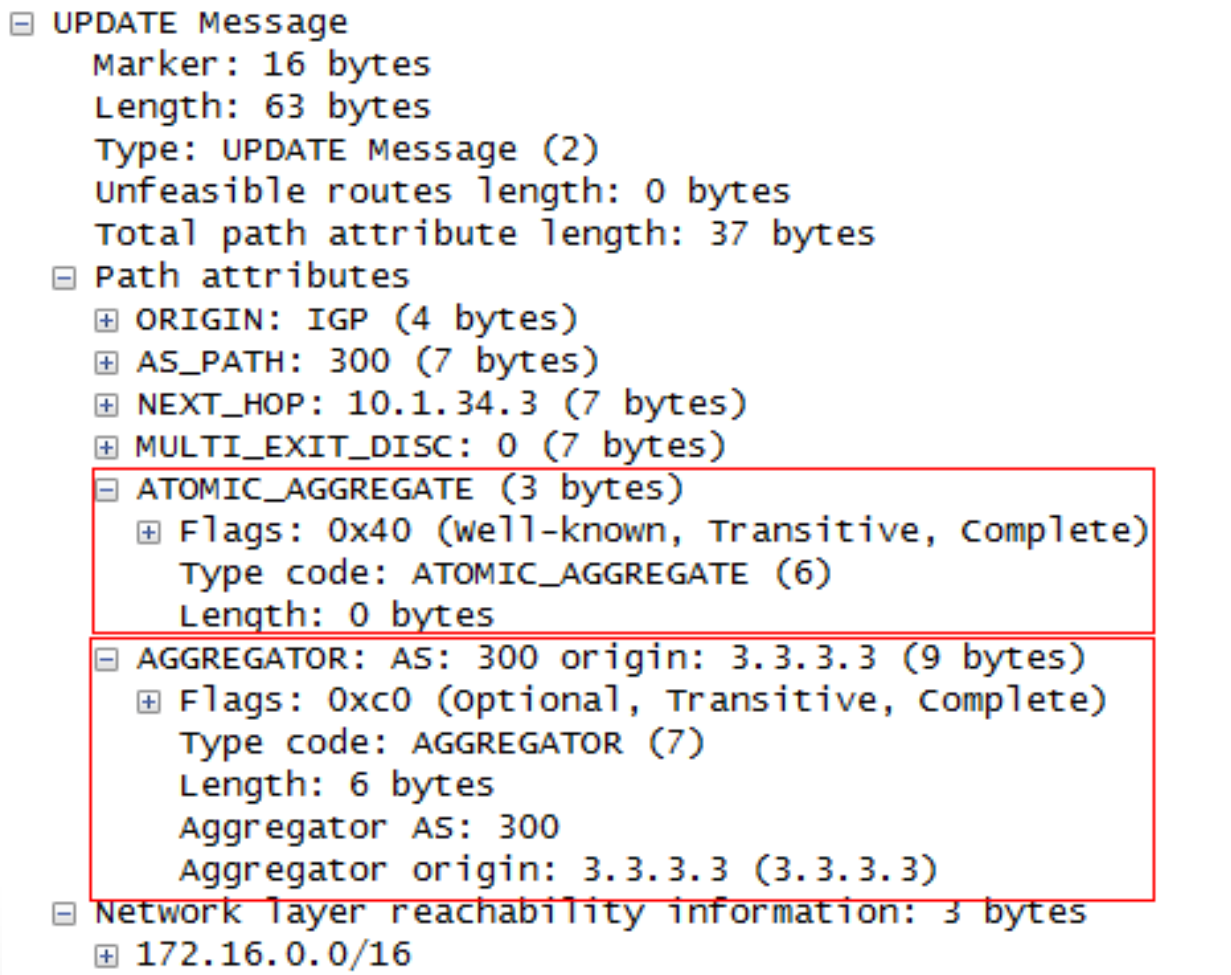
Not advertised to any peer

300, (**aggregated by 300 3.3.3.3**)

10.1.34.3 from 10.1.34.3 (3.3.3.3)

Origin IGP, metric 0, localpref 100, valid, external, **atomic-aggregate**, best

Atomic_Aggregate及aggregator



红茶三杯
Vinsoney

学习 沉淀 成长 分享

关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

BGP路由策略

红茶三杯（朱SIR） http://weibo.com/vinsoney

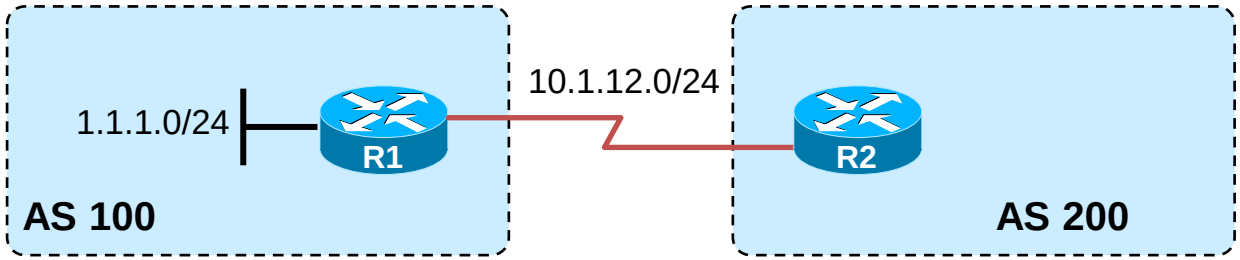
Latest update: 2012-08-01

课程目标

- BGP路由汇总
- distribute-list
- 正则表达式及as-path access-list
- advertise-map
- 使用community操控BGP路由
- ORF
- Prefix-list
- 路由拆分 BGP Deaggregation
- Route-map

BGP路由汇总

- **BGP自动汇总**

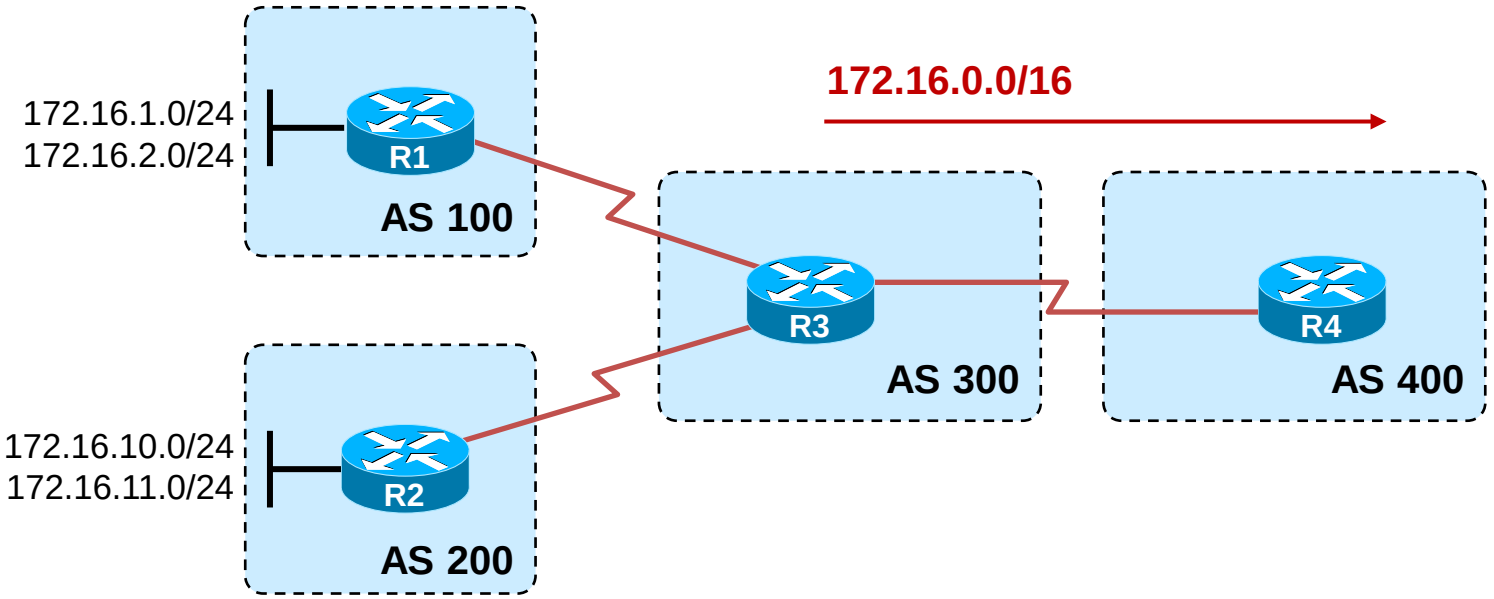


- 若R1开启auto-summary，并用重发布直连的方式引入1.1.1.0/24，则该子网会被汇总
- 若R1开启auto-summary，且network 1.1.1.0 mask 255.255.255.0，则仍以明细更新
- 若R1开启auto-summary，且network 1.0.0.0 mask 255.0.0.0，则该子网会被汇总
- 上面这条network等同于network 1.0.0.0（network的有类宣告）

因此BGP自动汇总（auto-summary）只汇总重发布引入的路由，以及使用network命令有类宣告方式引入的路由。目前CISCO IOS默认关闭自动汇总。

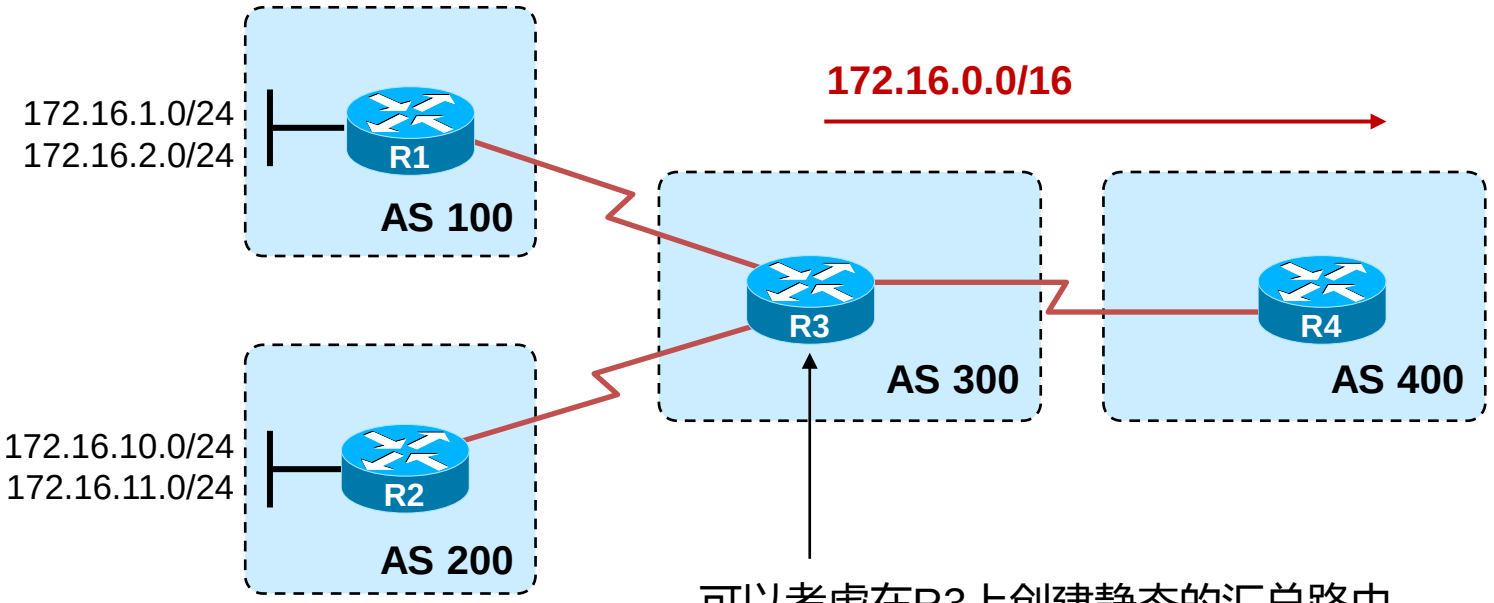
BGP路由汇总

- **BGP手工汇总**



BGP路由汇总

- **BGP手工汇总**



可以考虑在R3上创建静态的汇总路由

ip route 172.16.0.0 255.255.0.0 null0

然后再将汇总路由network进BGP

不过这个方法不是特别推荐

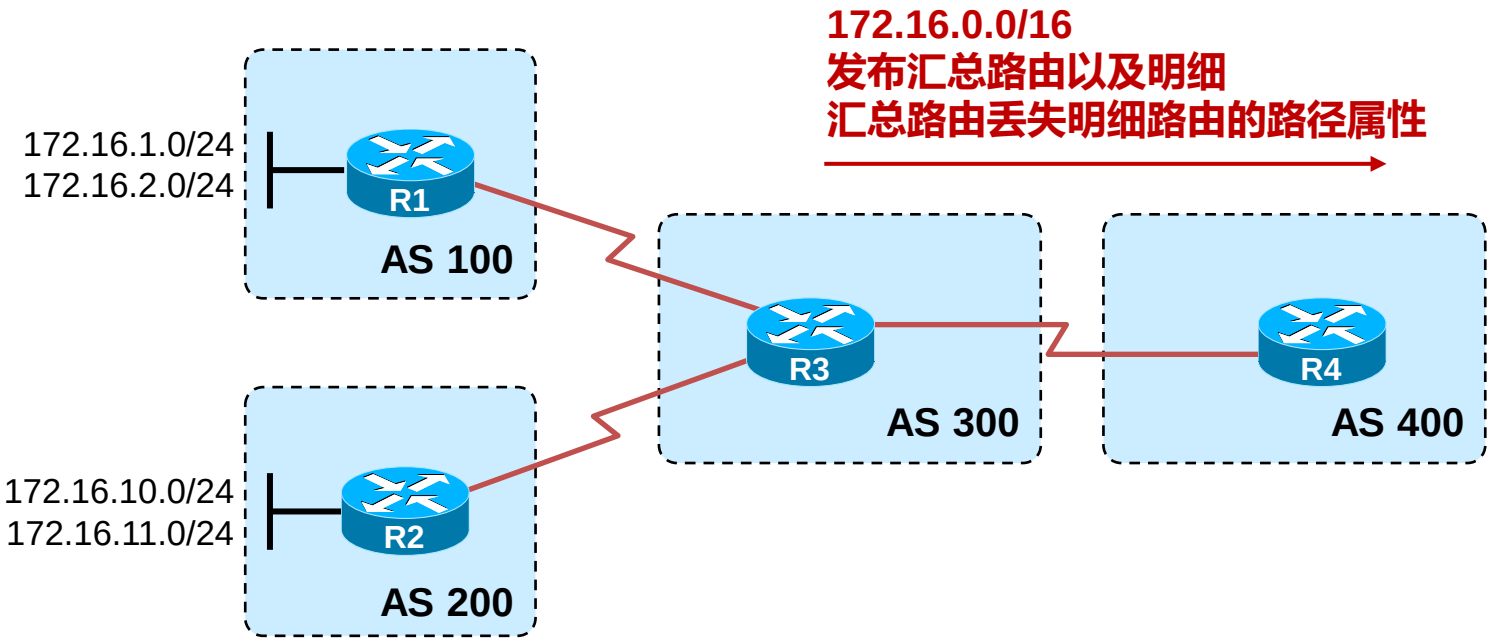
BGP路由汇总

- **BGP手工汇总**

```
R3(config-router)#aggregate-address 172.16.0.0 255.255.0.0 ?
  advertise-map     Set condition to advertise attribute
  as-set            Generate AS set path information
  attribute-map     Set attributes of aggregate
  nlri              Nlri aggregate applies to
  route-map         Set parameters of aggregate
  summary-only      Filter more specific routes from updates
  suppress-map      Conditionally filter more specific routes from updates
  <cr>
```

BGP路由汇总

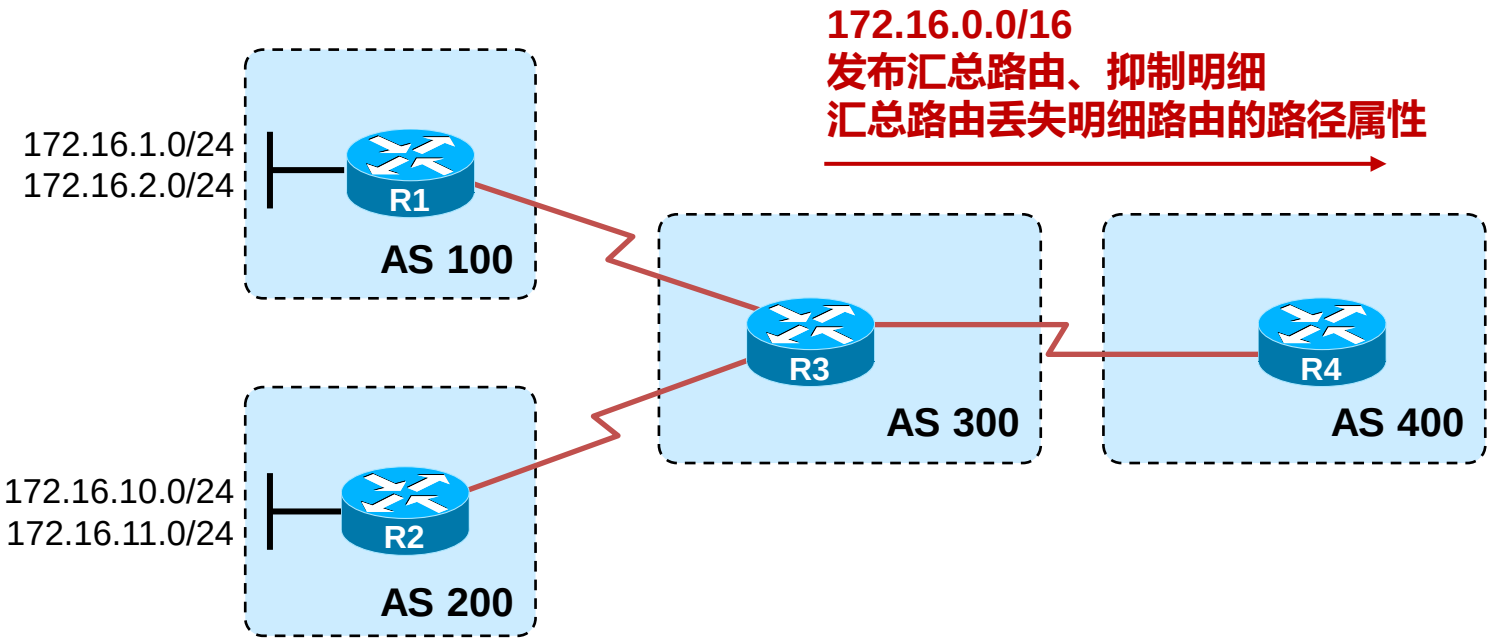
- **BGP手工汇总**




```
aggregate-address 172.16.0.0 255.255.0.0
```

BGP路由汇总

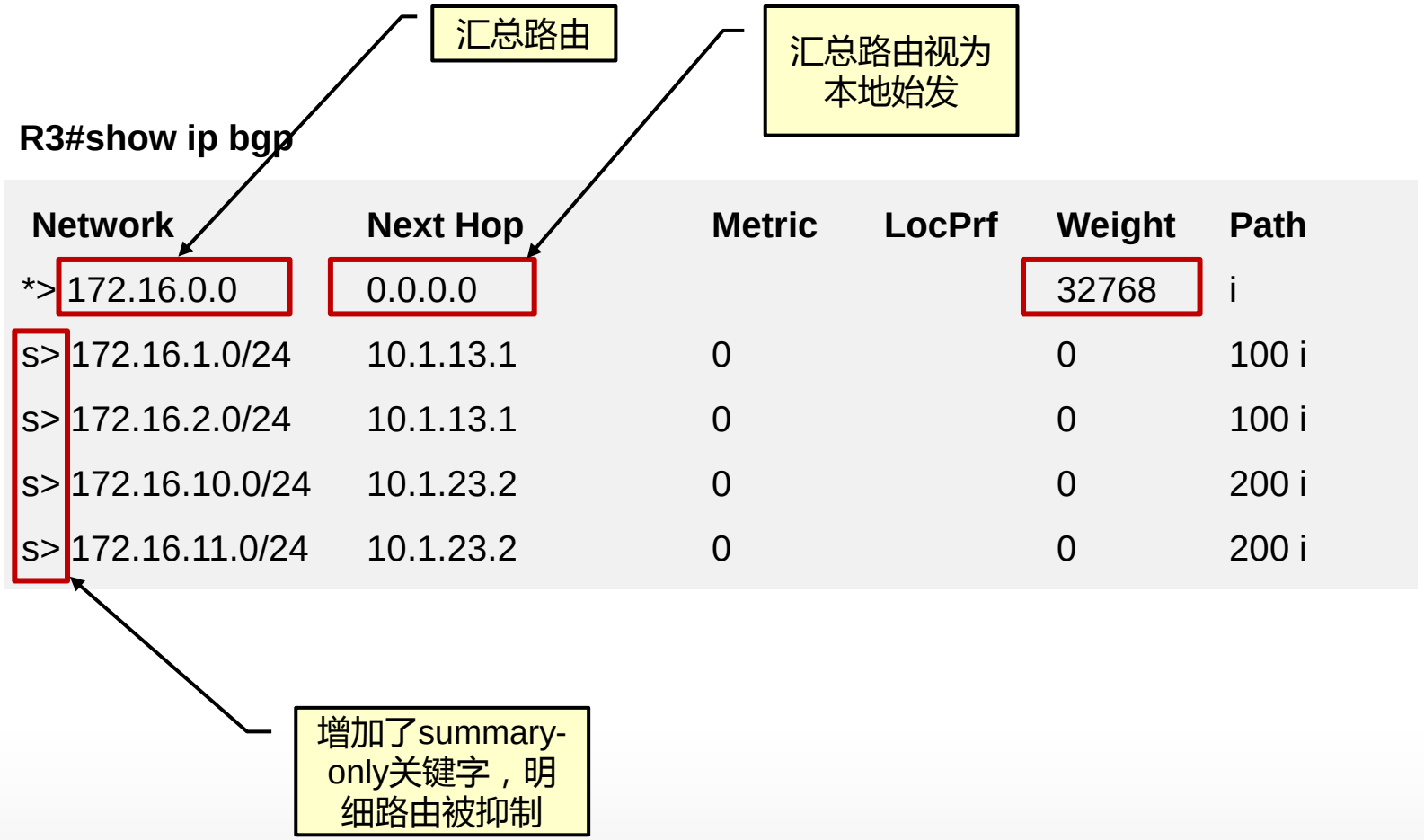
- **BGP手工汇总**



aggregate-address 172.16.0.0 255.255.0.0 **summary-only**

BGP路由汇总

- **BGP手工汇总**



BGP路由汇总

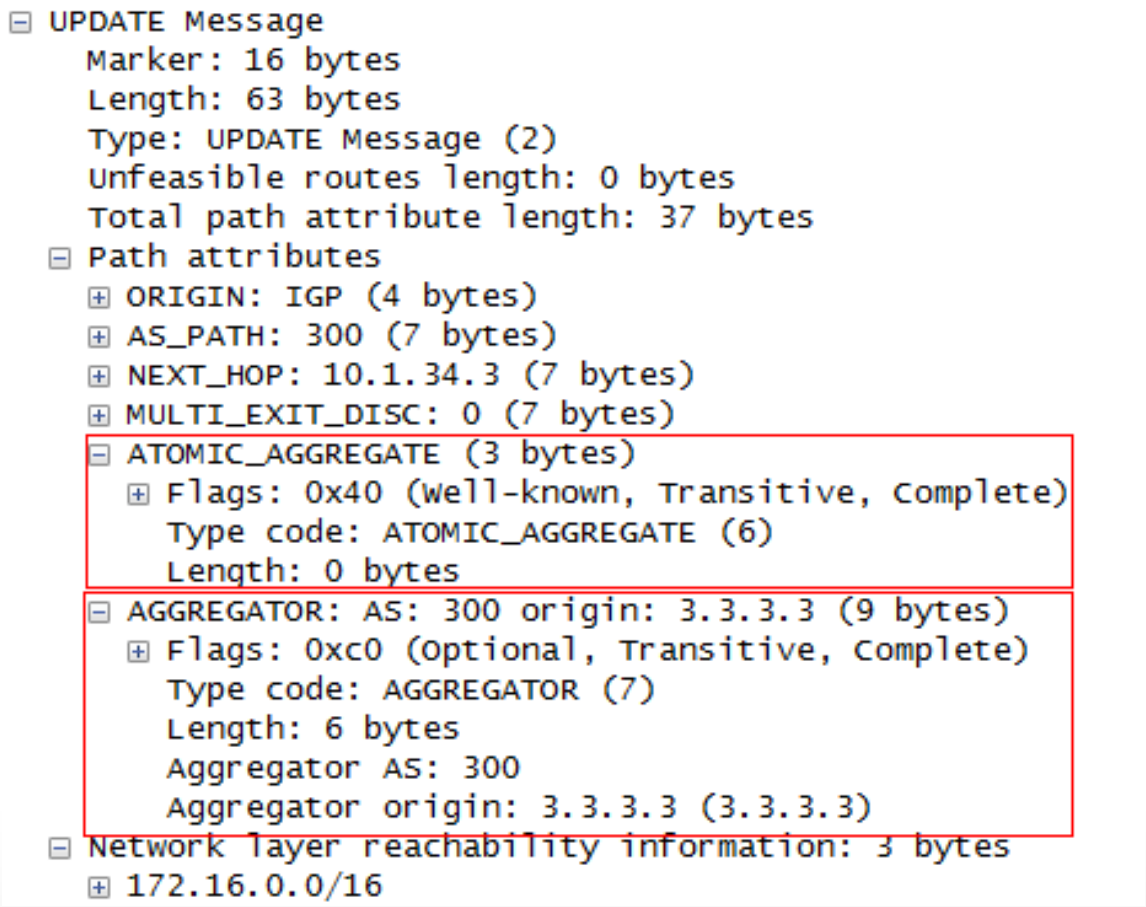
- **BGP手工汇总**

R4#show ip bgp 172.16.0.0

```
BGP routing table entry for 172.16.0.0/16, version 4
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
  Not advertised to any peer
  300, (aggregated by 300 3.3.3.3)
    10.1.34.3 from 10.1.34.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, external, atomic-aggregate, best
```

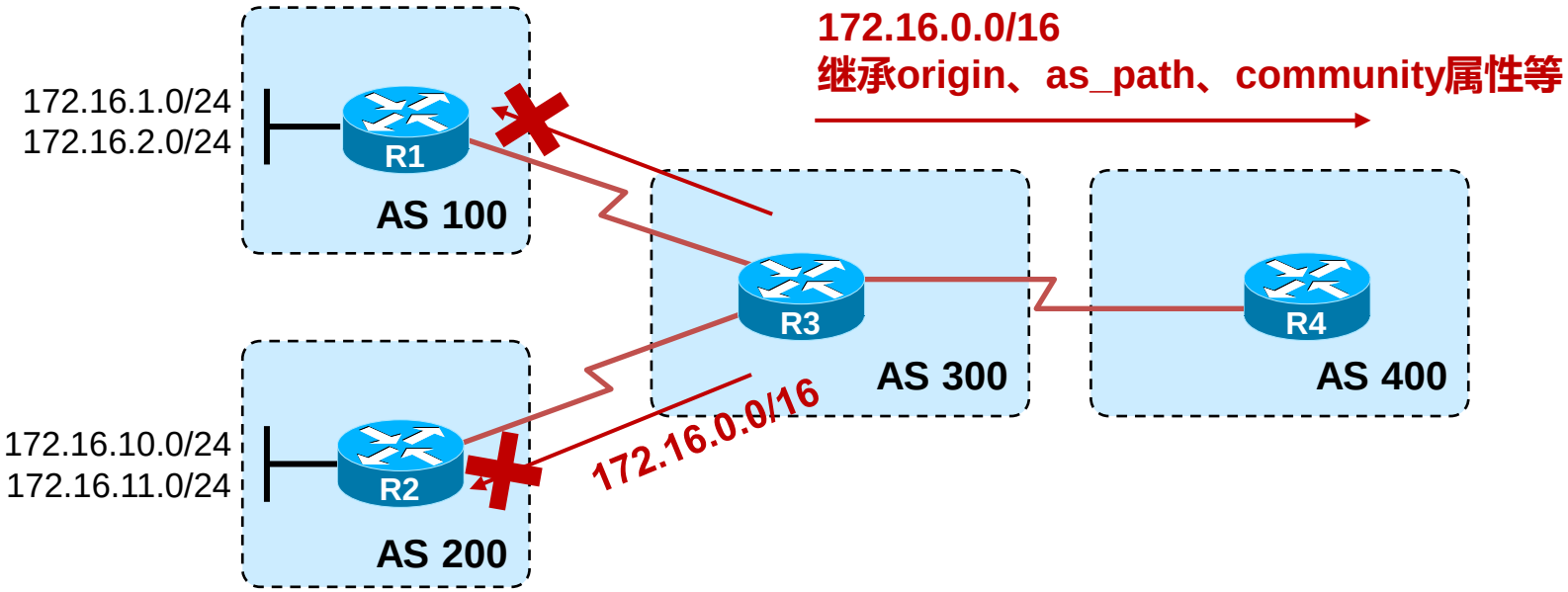
BGP路由汇总

- **BGP手工汇总**



BGP路由汇总

- **BGP手工汇总**



aggregate-address 172.16.0.0 255.255.0.0 summary-only **as-set**

BGP路由汇总

- **BGP手工汇总**

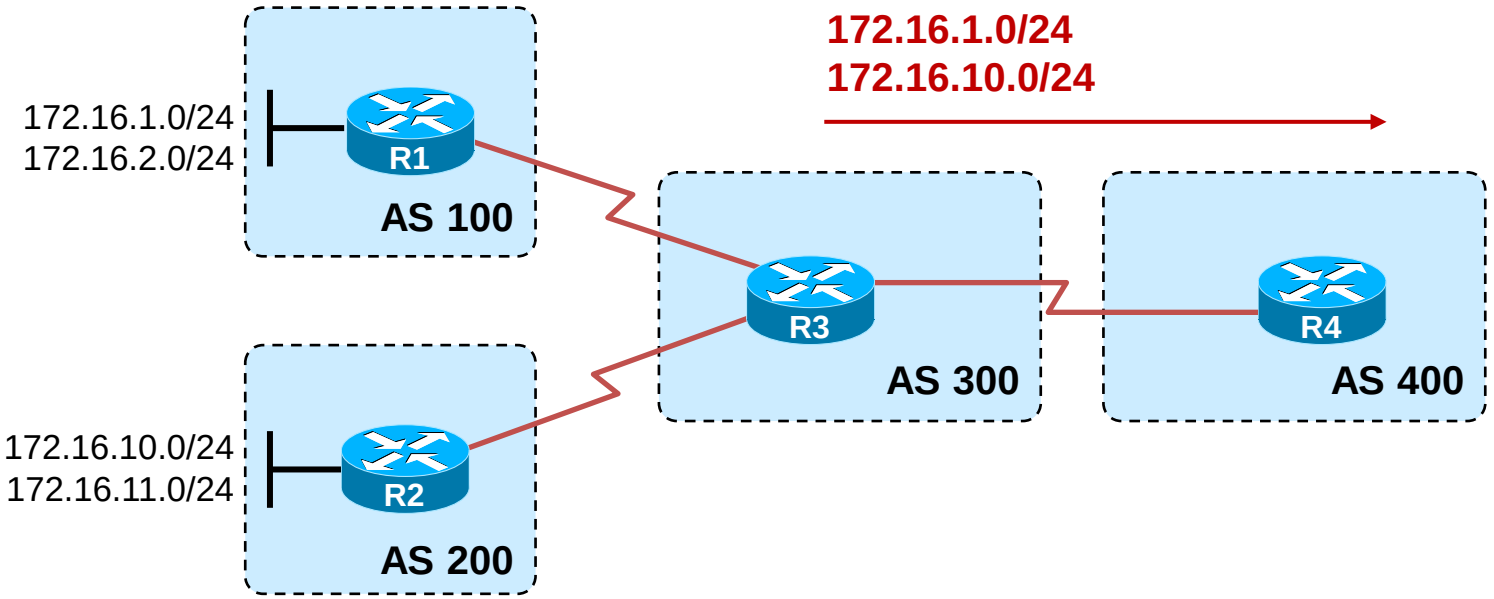
R4#show ip bgp 172.16.0.0

```
R4#show ip bgp 172.16.0.0
BGP routing table entry for 172.16.0.0/16, version 3
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  300 {100,200}, (aggregated by 300 3.3.3.3)
    10.1.34.3 from 10.1.34.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

由于汇总加了as-set关键字，明细的路径属性不至于丢失，因此无需atomic-aggregate这个属性。

BGP路由汇总

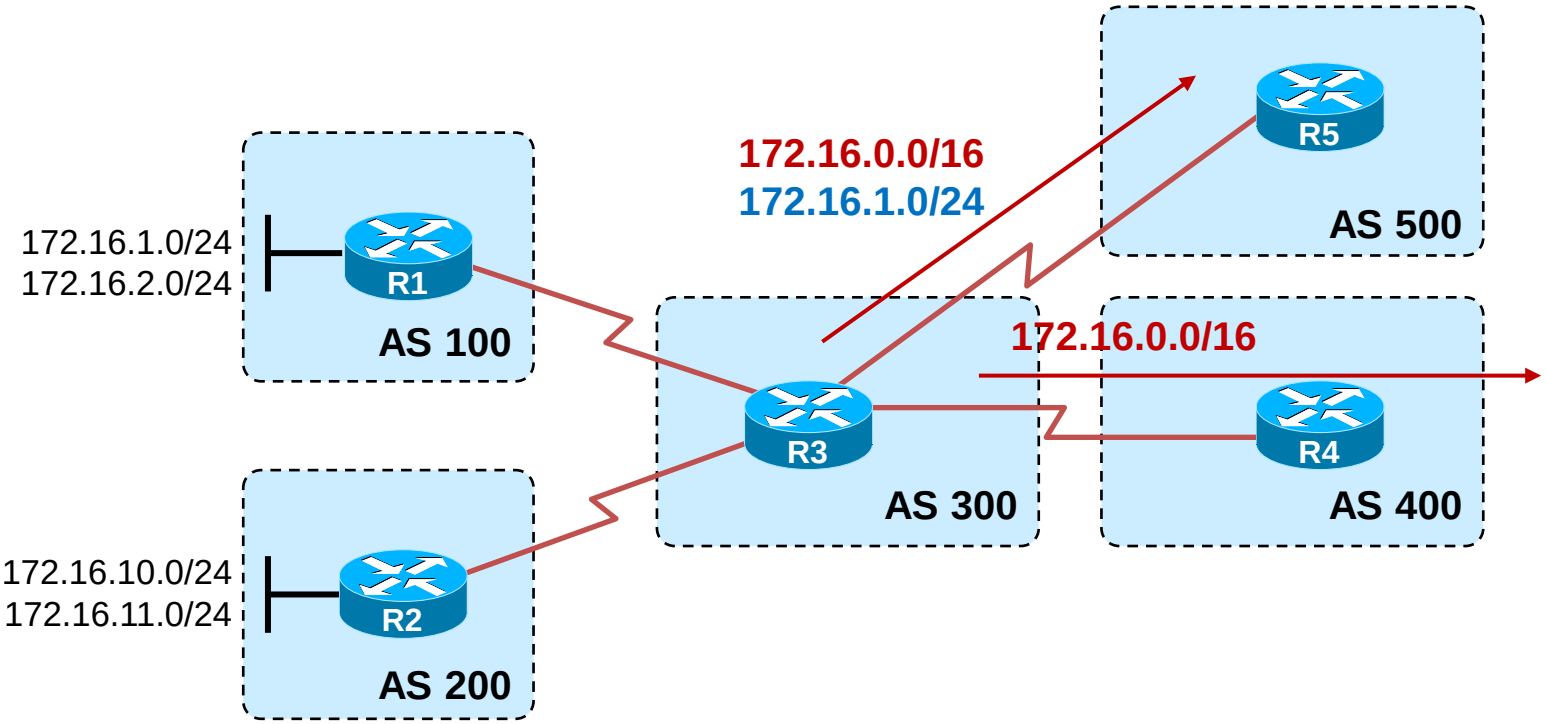
- **BGP手工汇总**




```
access-list 1 permit 172.16.2.0
access-list 1 permit 172.16.11.0
route-map suppmap permit 10
   match ip address 1
aggregate-address 172.16.0.0 255.255.0.0 suppress-map suppmap
```

BGP路由汇总

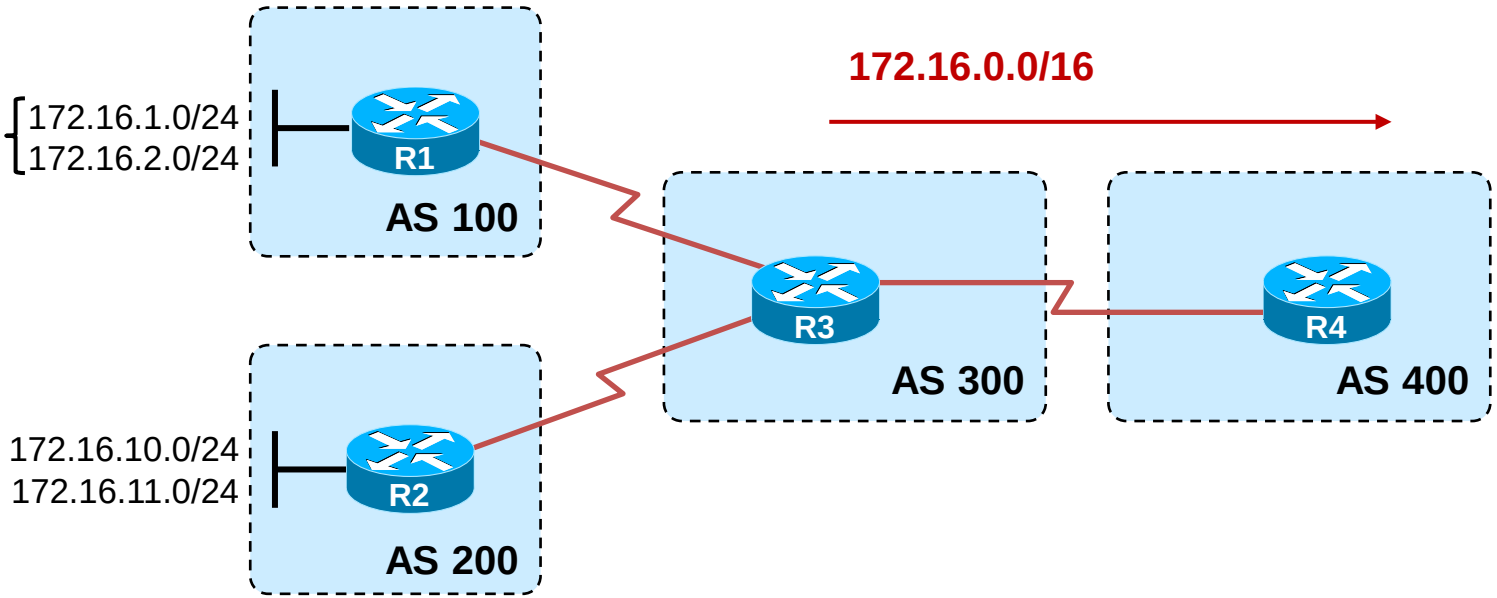
- **BGP手工汇总 针对特定邻居取消抑制**




```
access-list 1 permit 172.16.1.0
route-map unsupp permit 10
 match ip address 11
router bgp 300
   neighbor 10.1.35.5 unsuppress-map unsupp
   aggregate-address 172.16.0.0 255.255.0.0 as-set summary-only
```

BGP路由汇总

- **BGP手工汇总 advertise-map**




```
ip prefix-list list1 permit 172.16.10.0/24
ip prefix-list list1 permit 172.16.11.0/24
route-map adv permit 10
   match ip address prefix-list list1
aggregate-address 172.16.0.0 255.255.0.0 summary-only as-set advertise-map adv
```

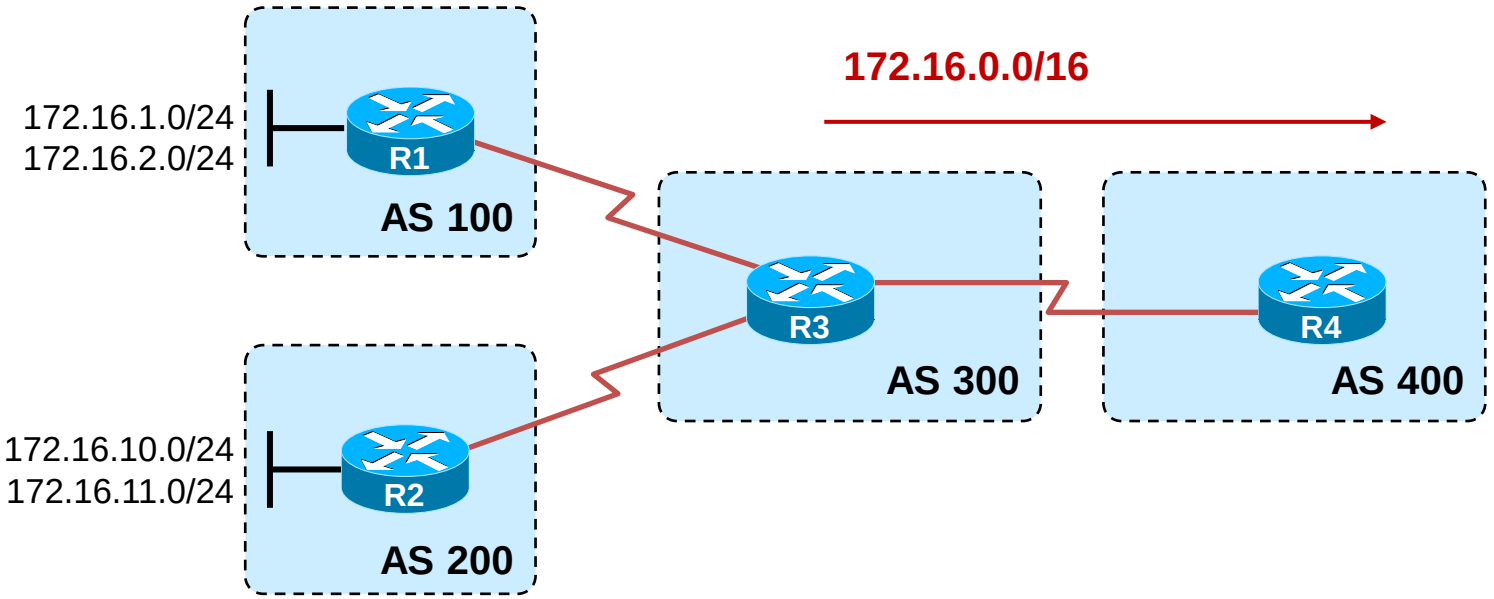
BGP路由汇总

- **BGP手工汇总 advertise-map**

advertise-map与summary-only合用时，aggregate-address的汇总地址下所有明细均被抑制（只发布汇总路由），同时advertise-map匹配的条目中明细如果全都挂了，则汇总路由也消失，（只要advertise-map匹配的明细有一条在，汇总就在）；并且汇总路由仅继承advertise-map匹配明细路由的BGP路径属性

BGP路由汇总

- **BGP手工汇总 attribute-map**



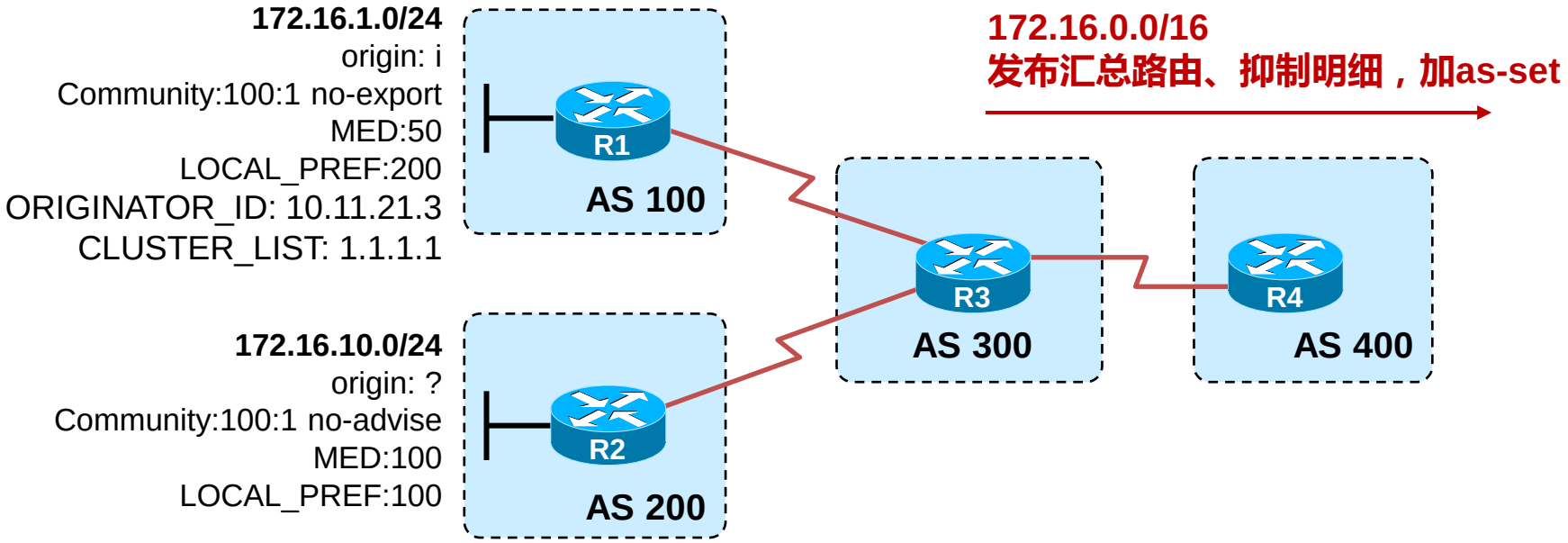
route-map **attr** permit 10

set ?

aggregate-address 172.16.0.0 255.255.0.0 summary-only as-set **attribute-map attr**

BGP路由汇总

- **BGP手工汇总 测试题**

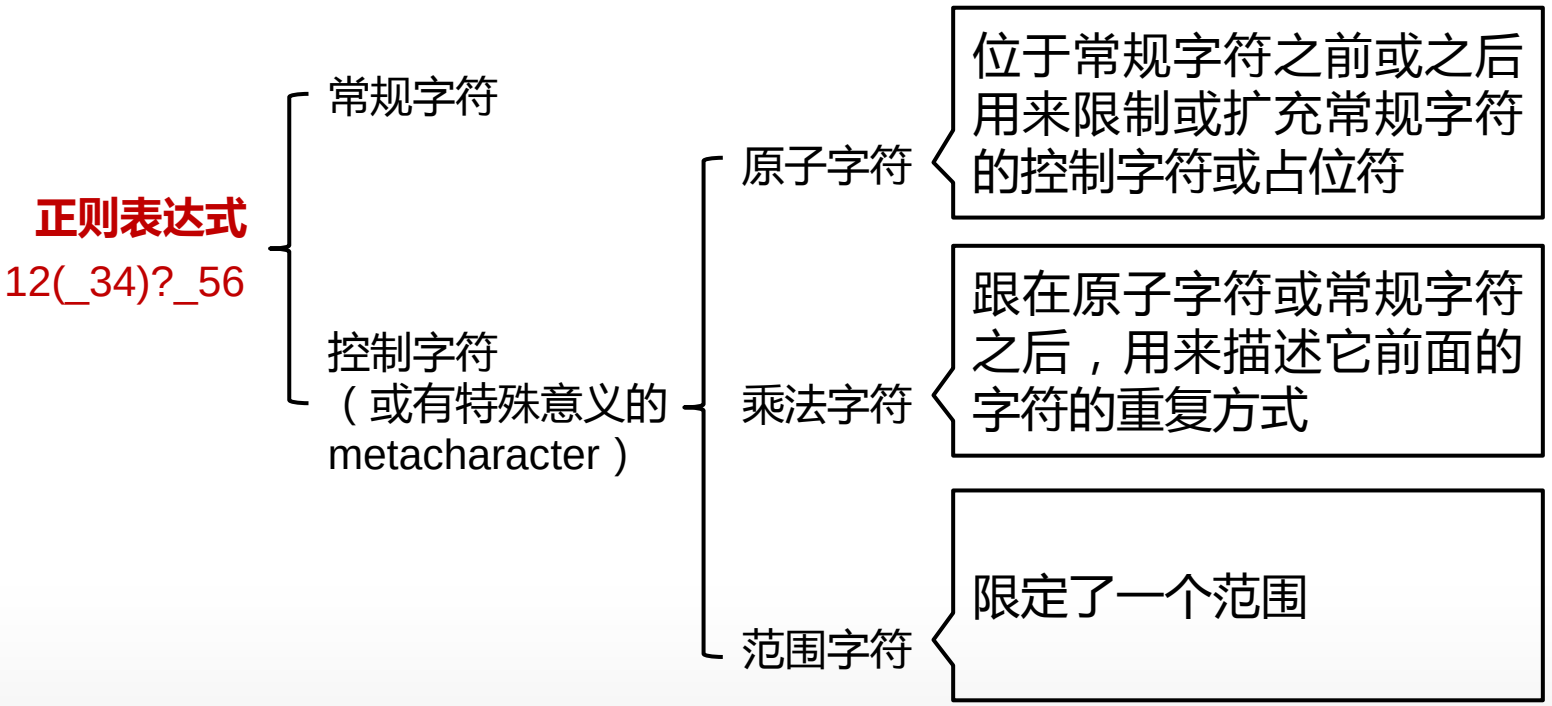


红茶三杯
Vinsoney

正则表达式及as-path access-list

正则表达式

- **正则表达式的构成**
  - 正则表达式(regular expression)是按照一定的模板来匹配字符串的公式。



正则表达式

- **原子字符**

.	匹配任何单个的字符，包括空格
^	一个字符串的开始
$	一个字符串的结束
_	下划线，匹配任意的一个分隔符如 ^、$、空格、tab、逗号、{、}
|	管道符，逻辑或
\	转义符，用来将紧跟其后的控制字符转变为普通字符

正则表达式

- **原子字符 示例**

^a.$	匹配一个以a开始，任意单一字符结束的字符串，如a0，a!等
^100_	匹配100、100 200、100 300 400等
^100$	匹配100
100$\|400$	匹配100、1400、300 400等
^\(65000\)$	仅仅匹配(65000)

正则表达式

- **乘法字符**

*	匹配前面字符0次或多次出现
+	匹配前面字符1次或多次出现
?	匹配前面字符的0次或1次出现

- **范围字符**

[ ]	表示一个范围。只匹配包含在范围内的字符之一。 可以在一个范围的开始使用 ^ 来排除范围内的所有字符，也可以使用下划线 _ 来指定一个区间。

正则表达式

- **乘法字符 示例**

abc*d	匹配abd、abcd、abccd、abcccd等
abc+d	匹配abcd、abccd、abcccd等
abc?d	匹配abd、abcd、abcdefg等
a(bc)?d	匹配ad、abcd、aaabcd等

- 一个乘法字符可以应用于一个单字符或多个字符，如果应用于多字符，需将字符串放入（ ）中。

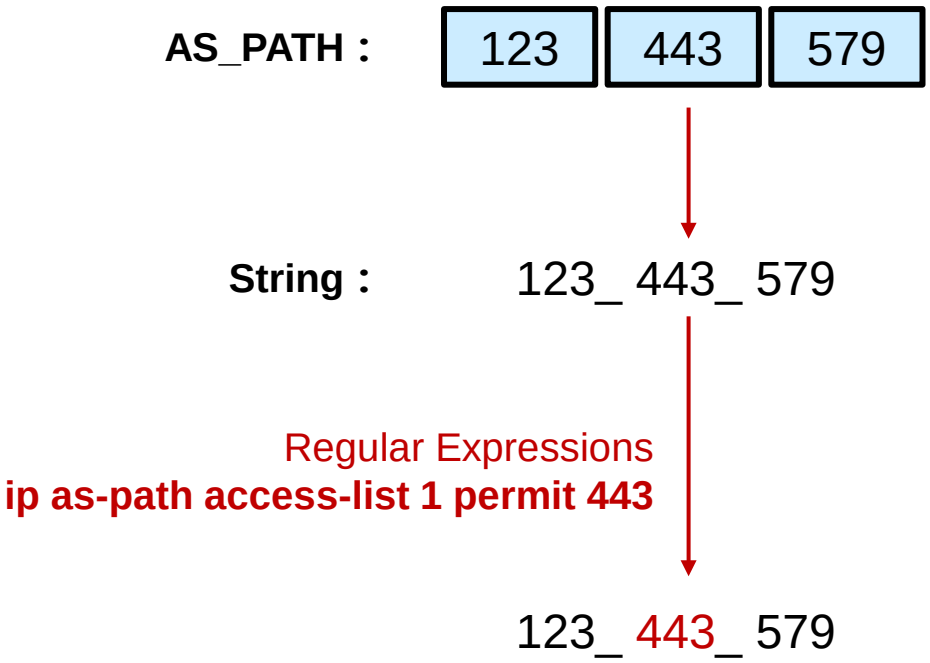
正则表达式

- **范围字符 示例**

[abcd]	匹配只要出现了a、b、c、d的内容
[a-c 1-2]$	匹配a、a1、62、1b、xv2等
[^act]$	匹配不以a或c或t结尾的内容
[123].[7-9]	159 220、91 70

正则表达式

- **使用正则表达式匹配AS_PATH**



AS_PATH可以当做字符串并使用正则表达式进行匹配

String中的“_”为空格，这也是一个字符，也有可能被匹配

正则表达式

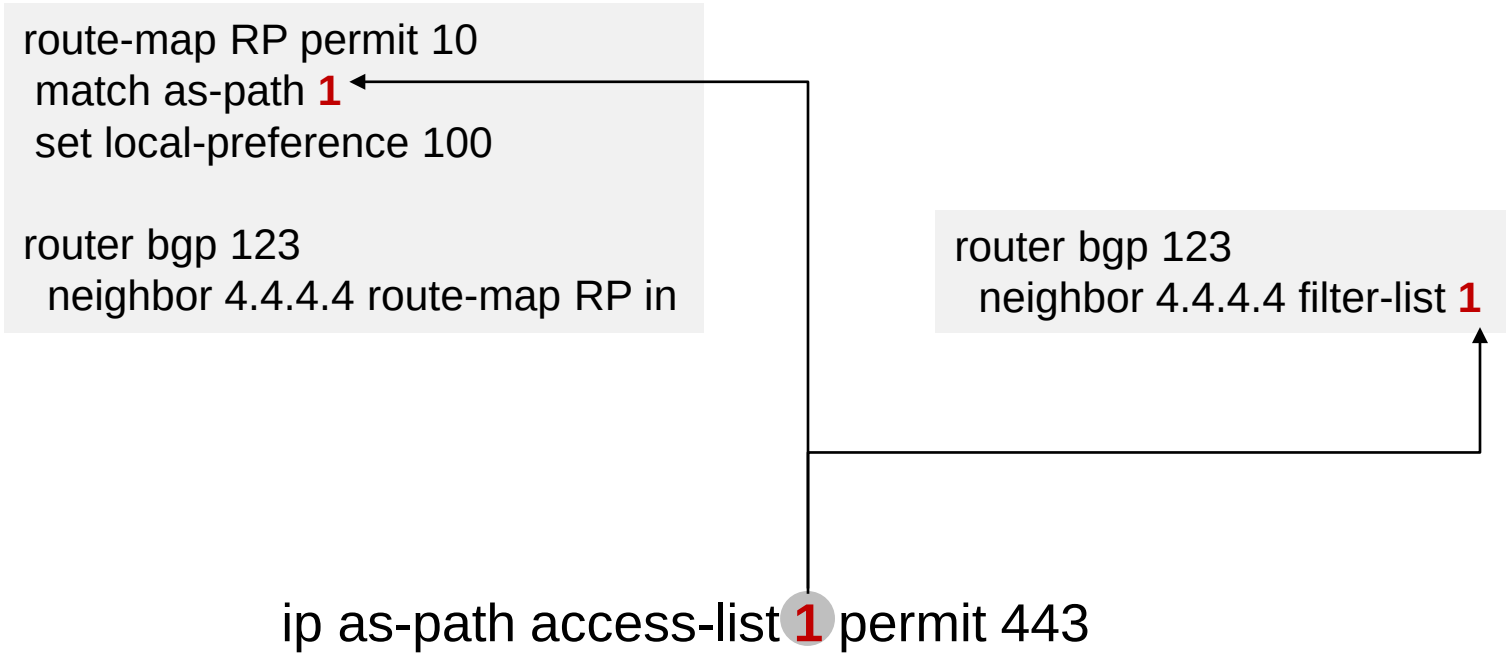
- **使用正则表达式匹配AS_PATH 示例**

^$	匹配不包含任何AS号的AS_PATH，也就是本AS内的路由
.*	一个点和一个星号，匹配所有，任何
^100$	就匹配100的这个AS_PATH
_100$	以100结束的AS_PATH，也就是路由起源于100AS的路由
^10[012349]$	匹配100、101、102、103、104、109这些AS_PATH
^10[^0-6]$	匹配除了100~106外的AS_PATH
^10.	匹配100~109，以及10，因为 “.” 也包含空格
^(100|200)$	匹配包含100及200的AS_PATH
12(_34)?_56	匹配12 56 及 12 34 56

- 注意as-path access-list也是默认隐含拒绝所有

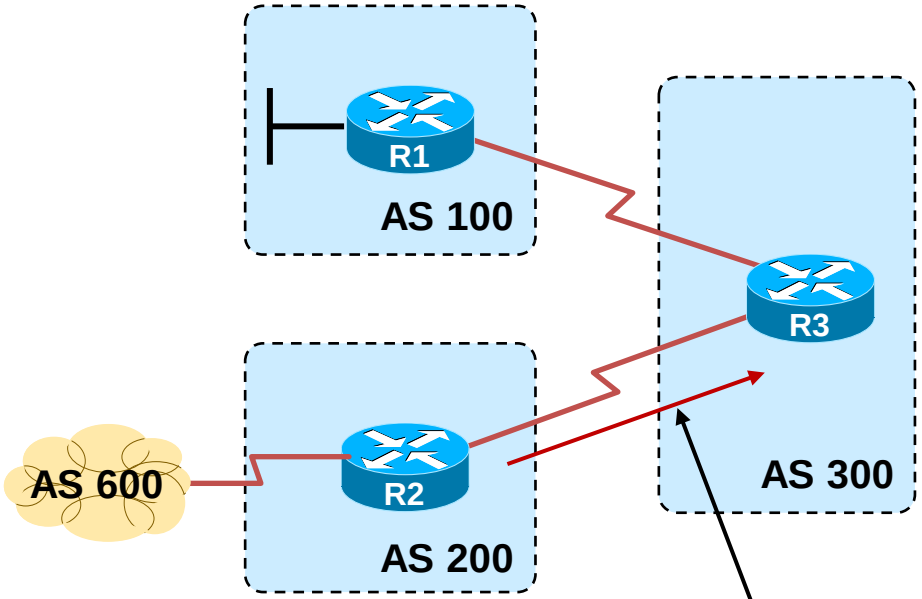
正则表达式

- **使用as-path access-list 匹配路由**



正则表达式

- **使用as-path access-list 匹配路由 示例1搭配filter-list**



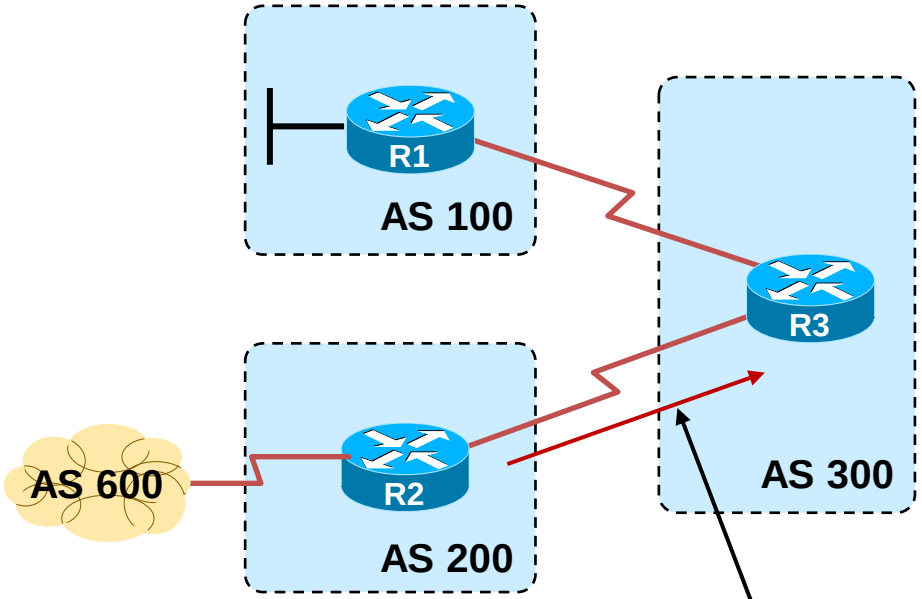
R2传递EBGP路由给R3，其中部分路由为R2本地始发，另一部分为AS600传递给R2再由R2更新给R3。现在要过滤掉AS600过来的路由，其他放行。

在R3上可做策略

```
ip as-path access-list 1 deny _600$
ip as-path access-list 1 permit .*
router bgp 300
  neighbor 10.1.23.2 filter-list 1 in
```

正则表达式

- **使用as-path access-list 匹配路由 示例2搭配route-map**



R2传递EBGP路由给R3，其中部分路由为R2本地始发，另一部分为AS600传递给R2再由R2更新给R3。现在要将来自AS600的路由限制在AS300内传递。

在R3上可做策略

```
ip as-path access-list 1 permit _600$
route-map setCommu permit 10
  match as-path 1
  set community local-as
route-map setCommu permit 10

router bgp 300
  neighbor 10.1.23.2 route-map setCommu in
```

正则表达式

- **配置命令**

Router(config)# ip as-path access-list *num* {permit|deny} *regexp*

配置as-path access-list

Router(config-router)# neighbor x.x.x.x filter-list *as-path-filter* {in|out}

关联as-path access-list到filter-list，起到路由过滤作用

正则表达式

- **验证及查看**

```
Router# show ip as-path-access-list
```

查看配置的 as-path access-list

```
Router# show ip bgp regexp xx
```

显示BGP表中所有被该正则表达式匹配上的路由，这是一个非常不错的工具

```
Router# show ip bgp filter-list access-list-num
```

显示BGP表中所有被该filter-list匹配的路由

红茶三杯
Vinsoney

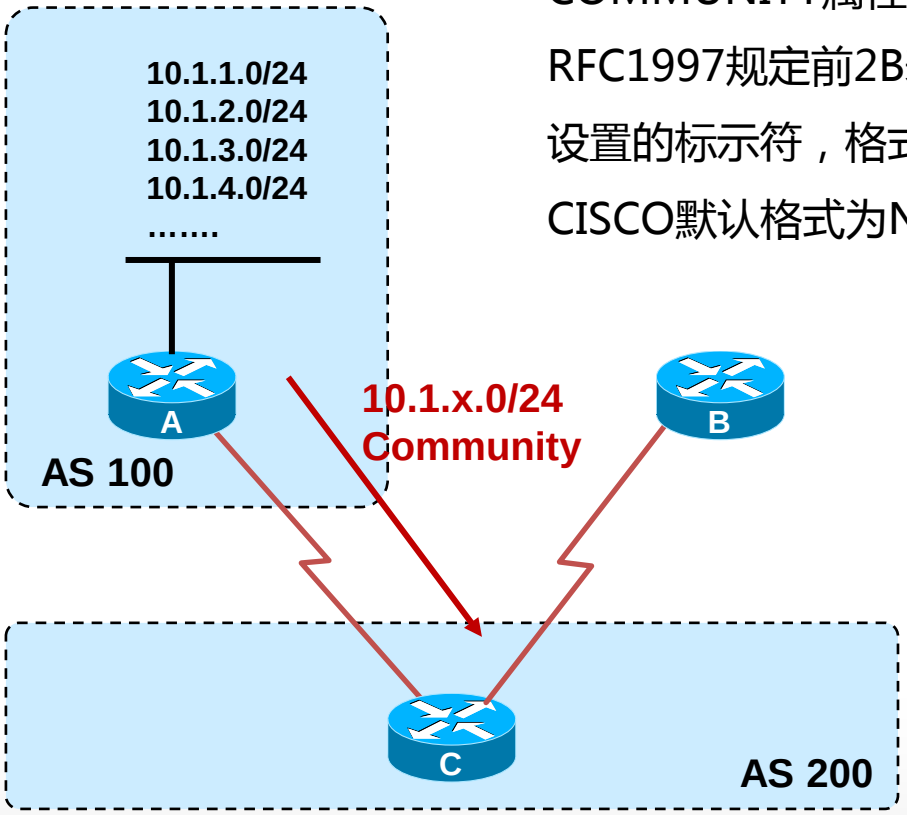
使用community操控BGP路由

使用community操控BGP路由

- **BGP Communities**
  - BGP communities是一种路由标记方法，用于确保路由过滤和选择的连续性
  - 可选传递属性，不支持该属性的BGP router原封不动的将community值传递给下游BGP邻居

使用community操控BGP路由

- **BGP路径操纵**
  - 设置Community



可选传递，用于简化路由策略的执行

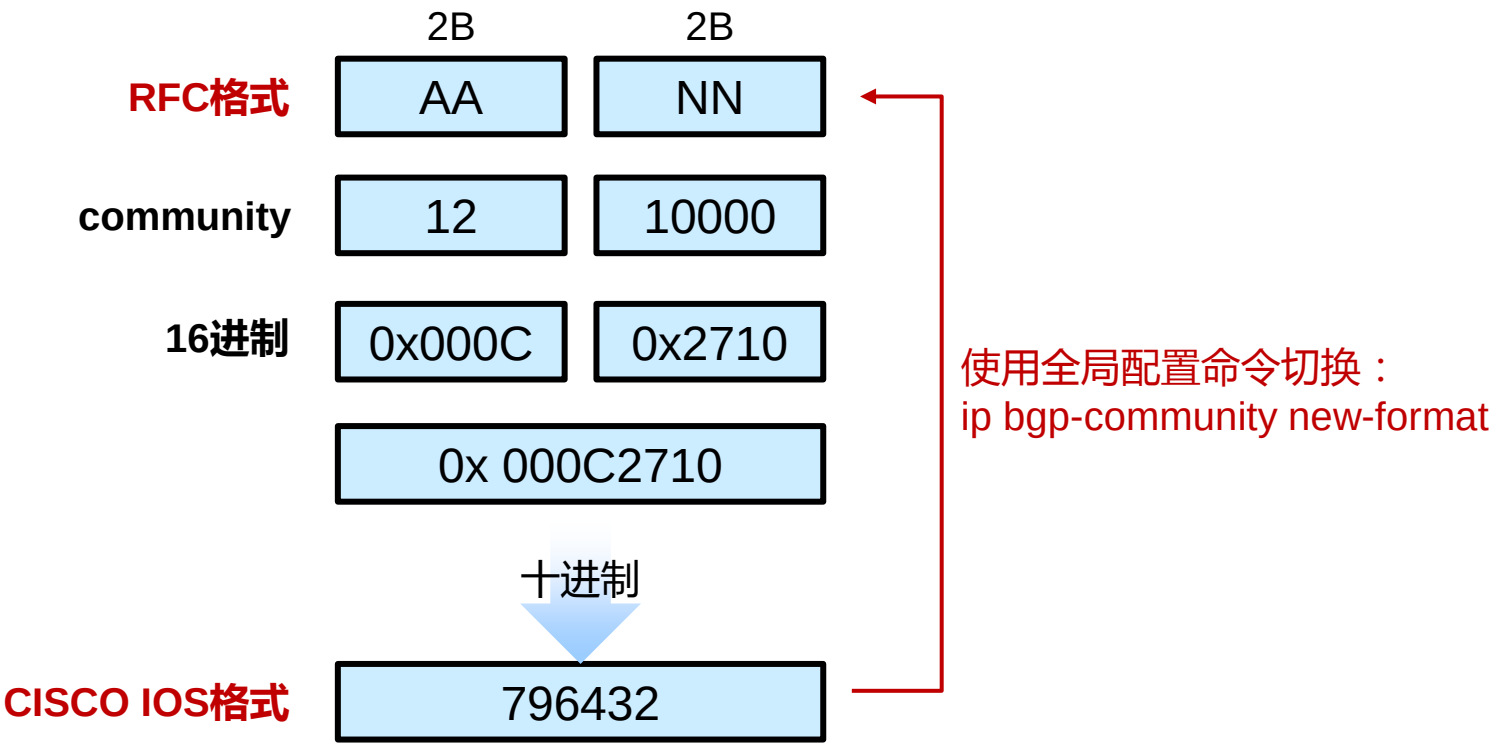
COMMUNITY属性是一组4个8位组的数值

RFC1997规定前2B表示AS号，后2B表示基于管理目的设置的标示符，格式为AA：NN

CISCO默认格式为NN：AA

使用community操控BGP路由

- **Community**



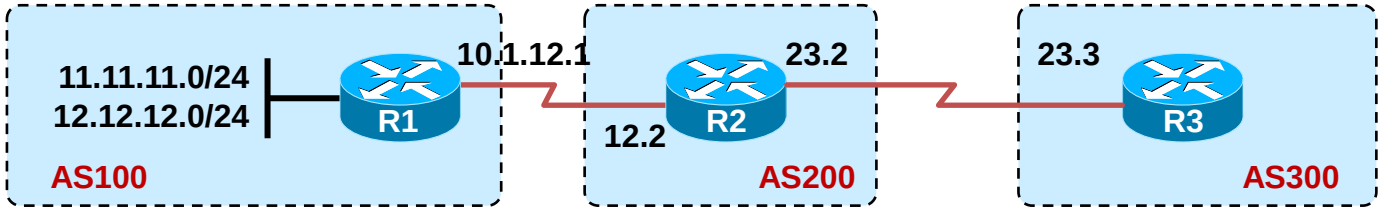
使用community操控BGP路由

- **设置Community**

```
route-map test permit 10
set community ?
  <1-4294967295>   community number
  aa:nn            community number in aa:nn format
  additive         Add to the existing community
  internet         Internet (well-known community)
  local-AS         Do not send outside local AS (well-known community)
  no-advertise     Do not advertise to any peer (well-known community)
  no-export        Do not export to next AS (well-known community)
  none             No community attribute
```

使用community操控BGP路由

- **为路由前缀分配Community**

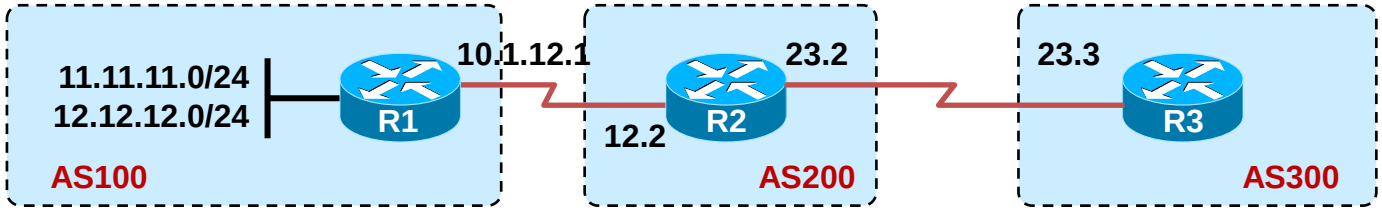


R1的配置如下：

```
ip prefix-list 11 permit 11.11.11.0/24
route-map test permit 10
 match ip address prefix-list 11
 set community 100:11

router bgp 100
 network 11.11.11.0 mask 255.255.255.0
 neighbor 10.1.12.2 remote-as 200
 neighbor 10.1.12.2 send-community
 neighbor 10.1.12.2 route-map test out
```

使用community操控BGP路由

- **为路由前缀分配多个Community**

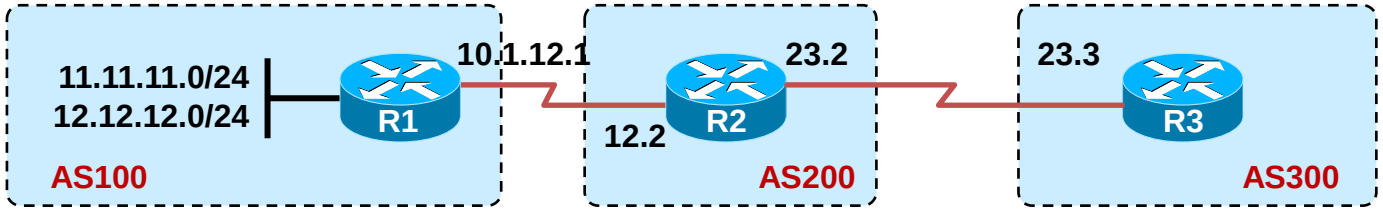


R2的配置如下：

```
ip community-list 11 permit 100:11
route-map test permit 10
 match community 11
 set community no-export additive

router bgp 200
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 neighbor 10.1.23.3 send-community
 neighbor 10.1.23.3 route-map test out
```

使用community操控BGP路由

- **为路由前缀分配多个Community(cont.)**

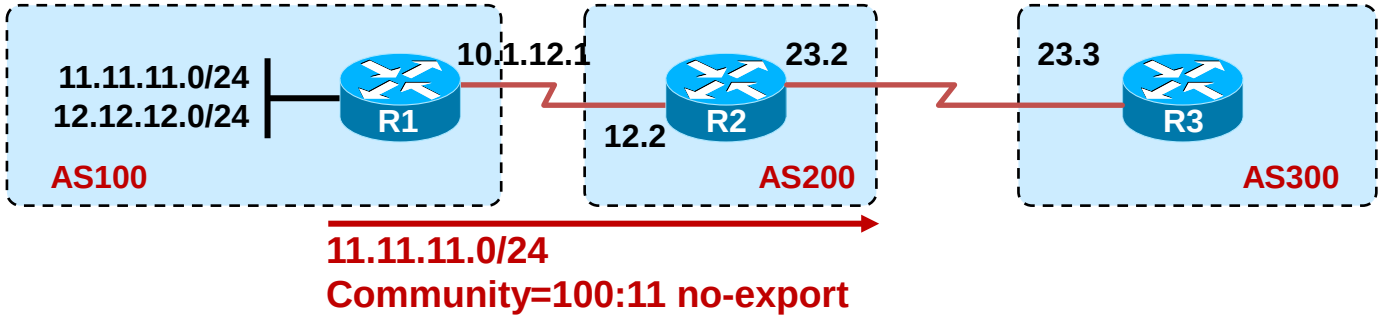


R3上查看11.11.11.0的BGP路由：

```
R3#sh ip b 11.11.11.0
BGP routing table entry for 11.11.11.0/24, version 5
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised
to EBGP peer)
Flag: 0x820
  Not advertised to any peer
  200 100
    10.1.23.2 from 10.1.23.2 (10.1.23.2)
      Origin IGP, localpref 100, valid, external, best
      Community: 100:11 no-export
```

使用community操控BGP路由

- **用community-list匹配团体属性**



Ip community-list 1 permit 100:11

匹配。匹配community中包含100:11的路由

Ip community-list 1 permit 100:11 no-adv

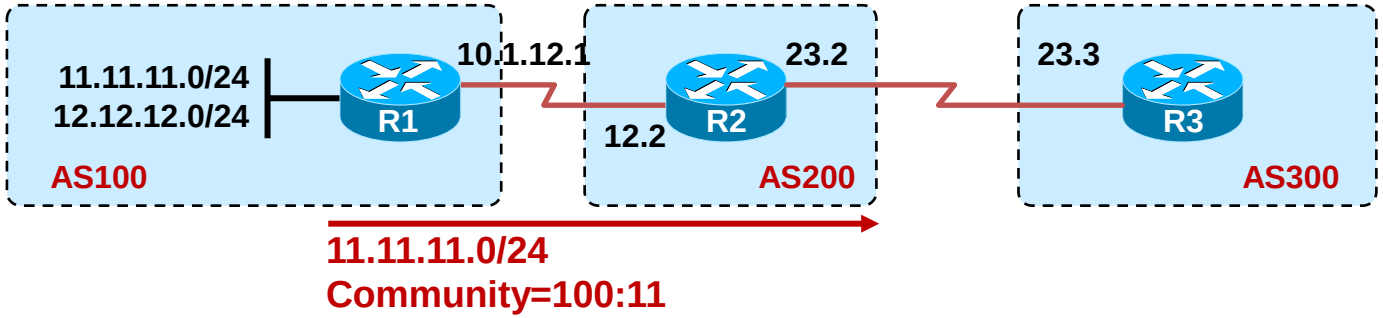
不匹配。要求100:11及no-adv两者都有才匹配成立

Ip community-list 1 permit 100:11
Ip community-list 1 permit no-export（或将no-export换成no-adv）

匹配。只要community中包含100:11或no-export

使用community操控BGP路由

- **用community-list匹配团体属性 cont.**




```
Ip community-list 1 permit 100:11
```

匹配。匹配community中包含100:11的路由

```
Ip community-list 1 permit 100:11 no-adv
```

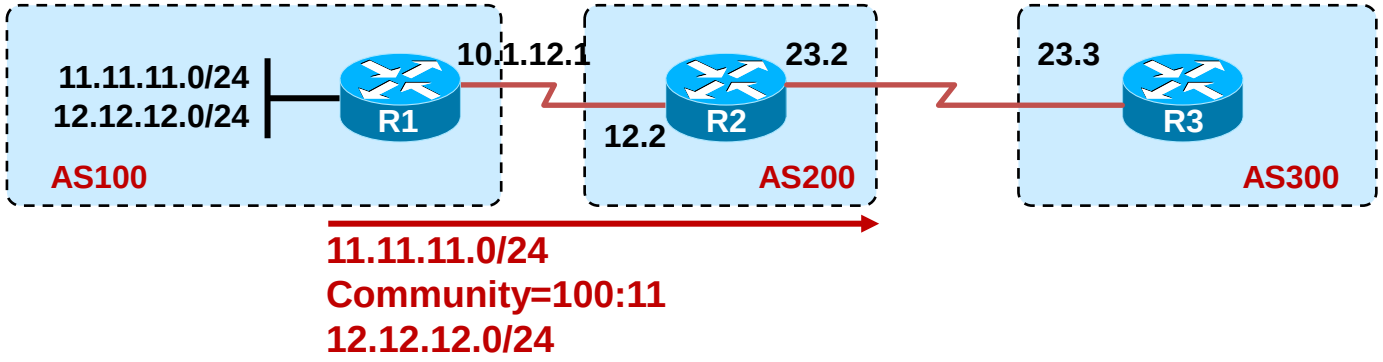
不匹配。要求100:11及no-adv两者都有才匹配成立

```
Ip community-list 1 permit 100:11
Ip community-list 1 permit no-export（或将no-export换成no-adv）
```

匹配。只要community中包含100:11或no-export

使用community操控BGP路由

- **用community-list匹配团体属性 cont.**

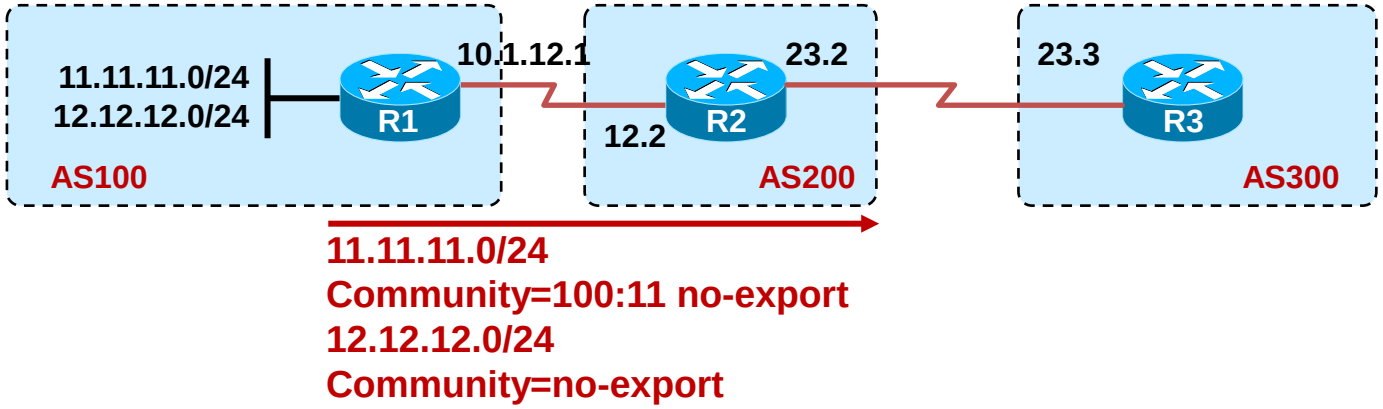



```
ip community-list 12 permit internet
```

默认所有路由都属于internet

使用community操控BGP路由

- **用community-list匹配团体属性 严格匹配**




```
Ip community-list 11 permit no-export
route-map test permit 10
  match community 11 exact-match            // 严格匹配
```

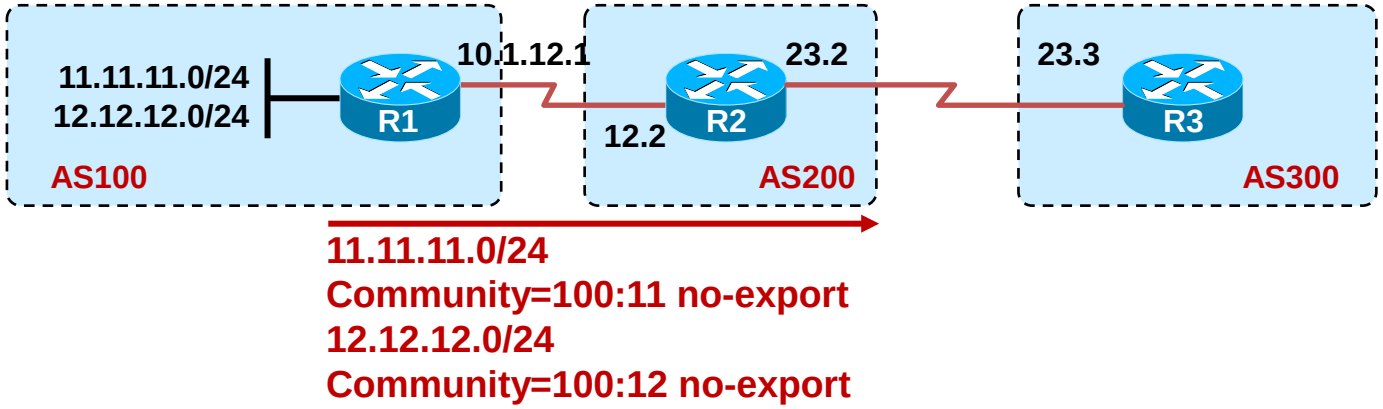
严格匹配community属性为no-export的路由，多一点，少一点都不行

```
Show Ip community-list 1 [exact-match]
```

查看community-list 1

使用community操控BGP路由

- **删除某个或多个community值**

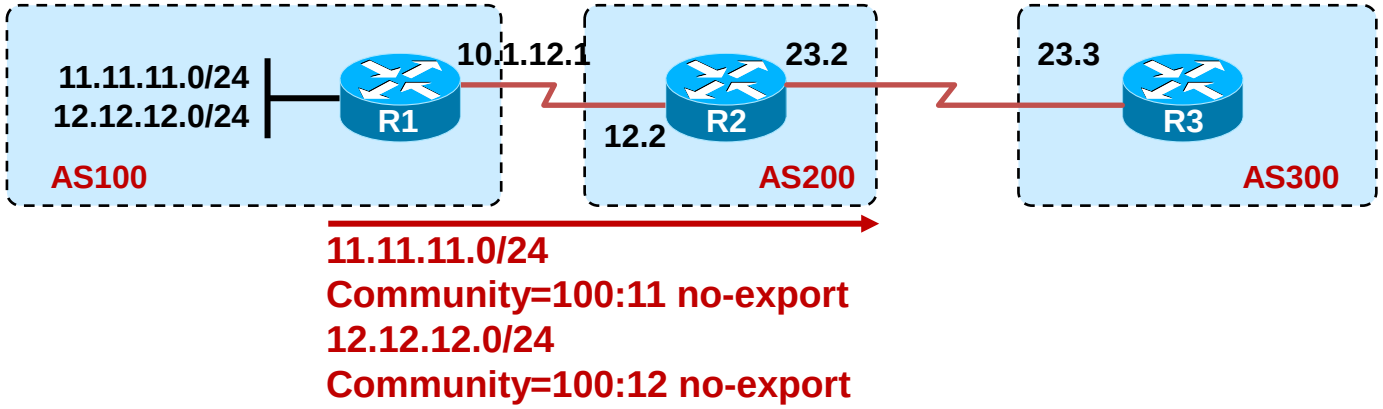


一条路由，允许携带多个community值，构成一个community列表，那么如何删除某个或者某几个community值？譬如只想删除11路由的no-export属性

```
ip community-list standard del permit no-export          // 匹配要删除的commu值
route-map test permit 10
 set comm-list del delete                                 // 用这条命令删除
```

使用community操控BGP路由

- **删除多个community值**




```
ip community-list standard del permit no-export
ip community-list standard del permit 100:11
route-map test permit 10
 set comm-list del delete
```

用多行community-list

使用community操控BGP路由

- **配置community-list**

```
Ip community-list 1-99 permit|deny value [value…]
```

定义标准的community-list，使用internet关键字匹配任何community

```
Ip community-list 100-199 permit|deny regexp
```

定义扩展的community-list，可使用正则表达式匹配community

```
show ip community-list
```

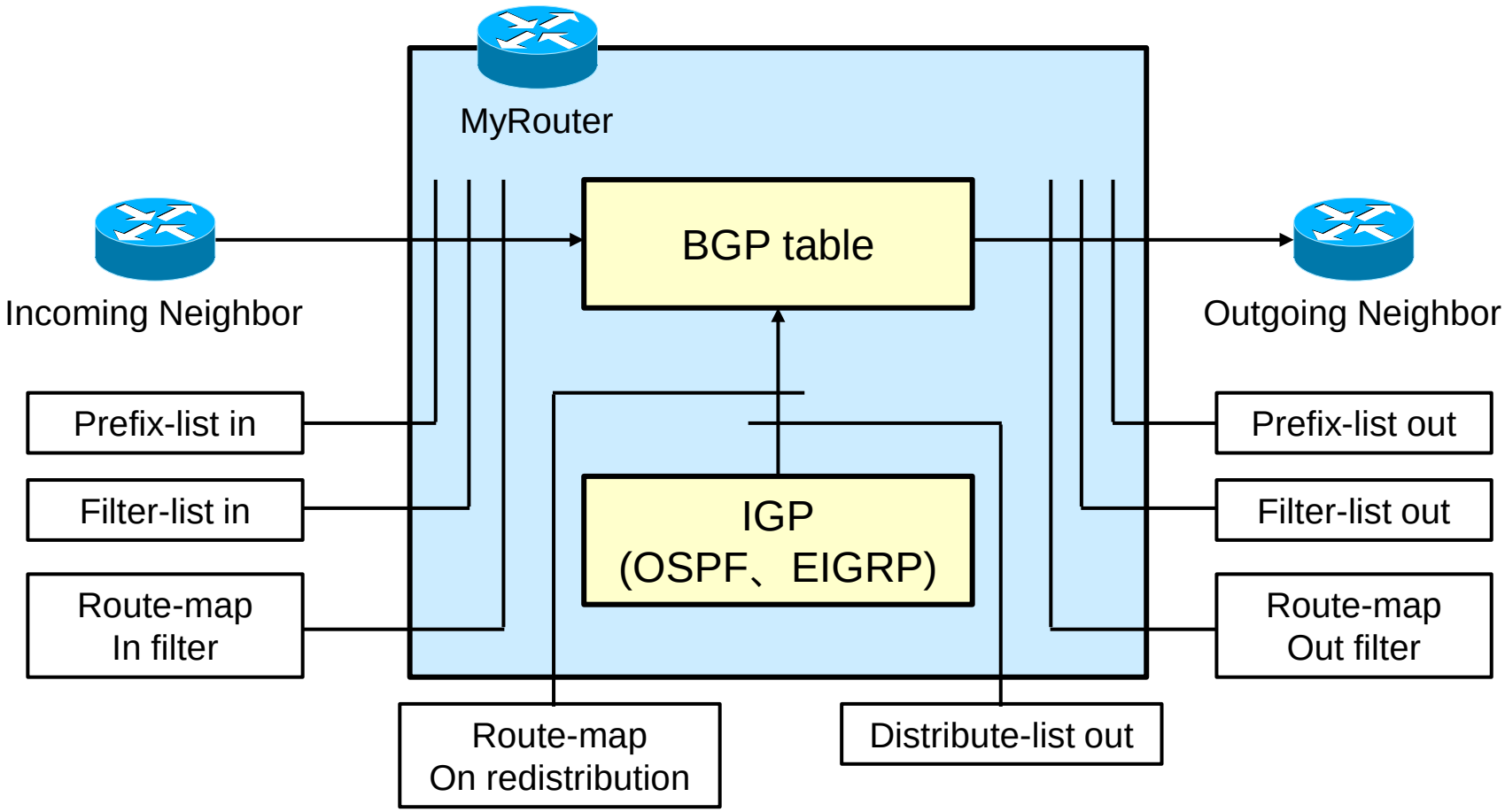
查看配置的community-list

```
show ip bgp x.x.x.x
```

查看BGP路由的详细信息，包括community

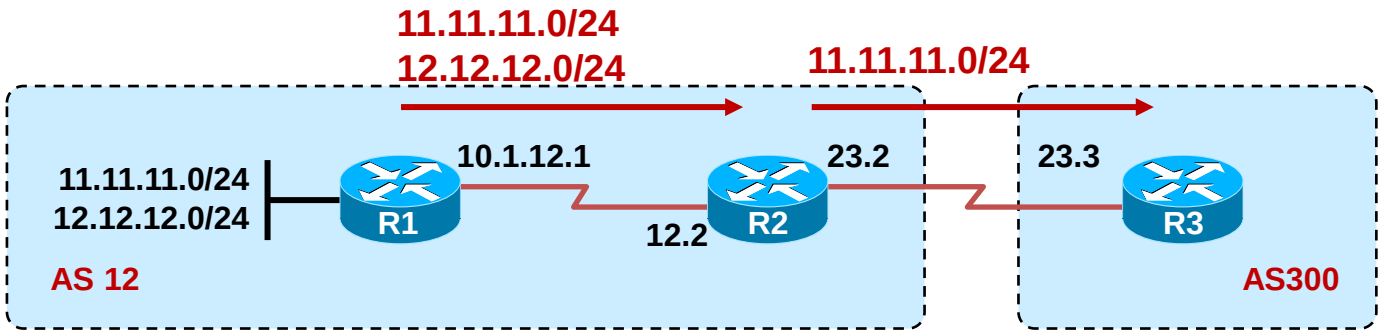
Prefix-list

BGP Filters



Prefix-list

- **配置示例**



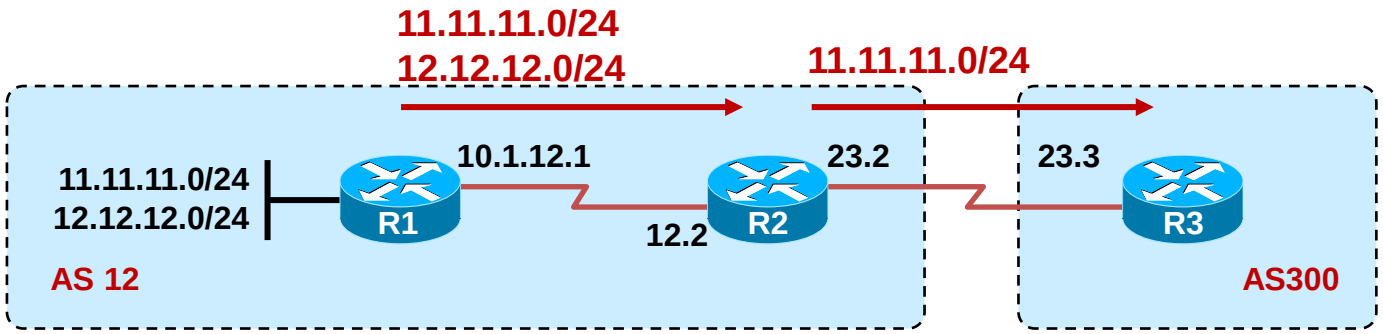
R2上，过滤掉12.12.12.0/24路由，其他放行

```
R2(config)# ip prefix-list 12 deny 12.12.12.0/24
R2(config)# ip prefix-list 12 permit 0.0.0.0/0 le 32
R2(config)# router bgp 12
R2(config-router)# neighbor 10.1.23.3 prefix-list 12 out
```

distribute-list

distribute-list

- **配置示例**

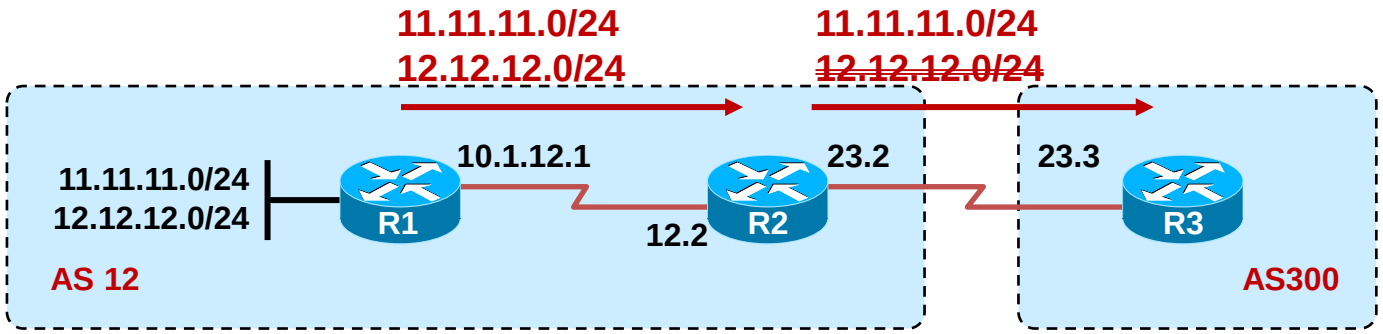


R2上，过滤掉12.12.12.0/24路由，其他放行

```
R2(config)# access-list 1 deny 12.12.12.0
R2(config)# access-list 1 permit any
R2(config)# router bgp 12
R2(config-router)# neighbor 10.1.23.3 distribute-list 1 out
```

distribute-list

- **配置示例2**

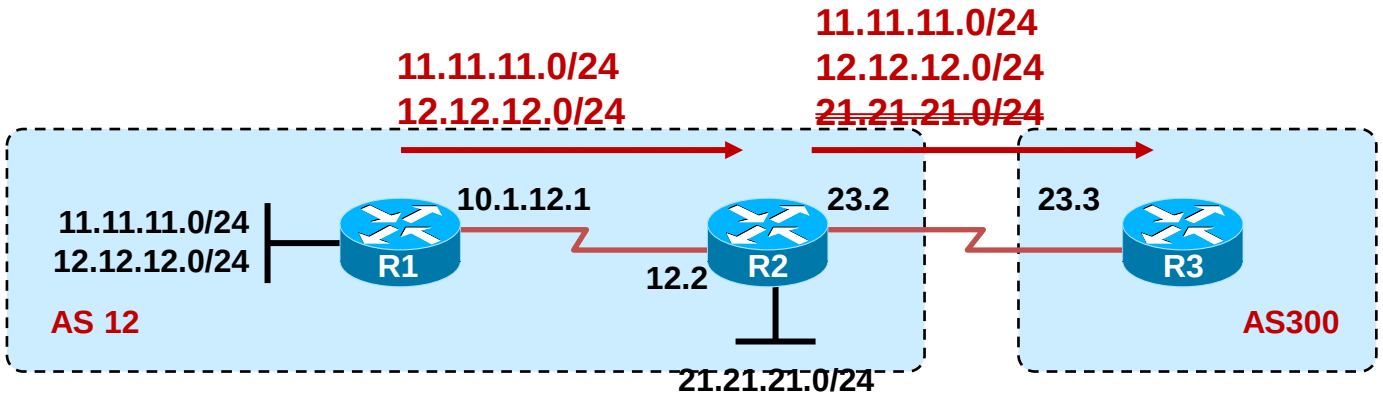


R2上，过滤掉12.12.12.0/24路由，其他放行

```
R2(config)# ip prefix-list 12 deny 12.12.12.0/24
R2(config)# ip prefix-list 12 permit 0.0.0.0/0 le 32
R2(config)# router bgp 12
R2(config-router)# distribute-list prefix 12 out          !! 12路由被过滤，如果out关键字后再加个接口，则无效
```

distribute-list

- **配置示例3**

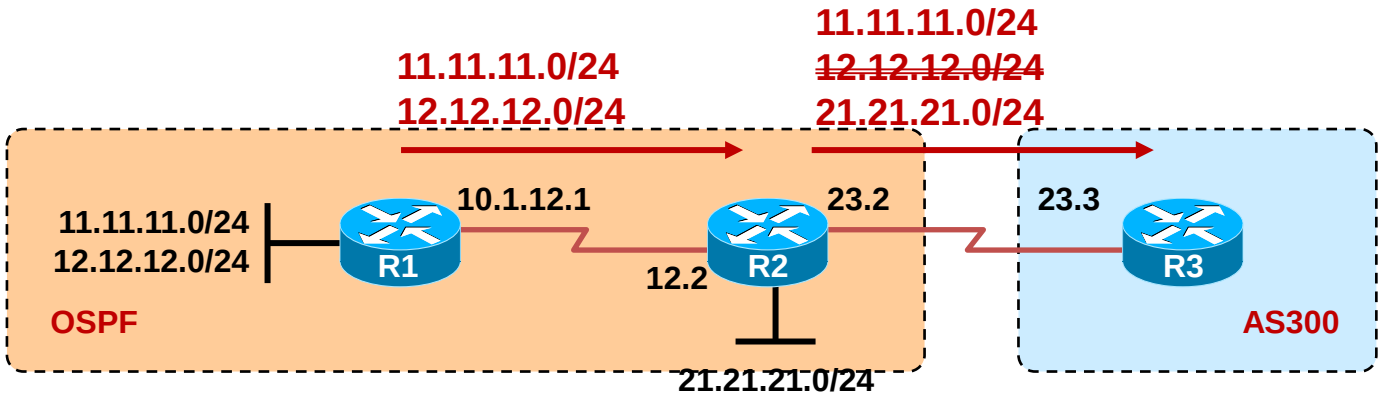


R2上，重发布直连接口，并用分发列表进行过滤

```
R2(config)# ip prefix-list 21 deny 21.21.21.0/24
R2(config)# ip prefix-list 21 permit 0.0.0.0/0 le 32
R2(config)# router bgp 12
R2(config-router)# redistribute connected
R2(config-router)# distribute-list prefix 21 out          !! 21路由被过滤
```

distribute-list

- **配置示例4**



R2上，重发布直连接口，并用分发列表进行过滤

```
R2(config)# ip prefix-list 12 deny 12.12.12.0/24
R2(config)# ip prefix-list 12 permit 0.0.0.0/0 le 32
R2(config)# router bgp 12
R2(config-router)# redistribute ospf 1 match internal external
R2(config-router)# redistribute connected
R2(config-router)# distribute-list prefix 12 out          !! 12.12.12.0/24被过滤，再关联接口则无效，关联协议例如加一个ospf 1则可以
```

Route-map

Route-map

- **可以在以下的BGP命令中使用route-map关键字**
  - neighbor
  - bgp dampening
  - network
  - redistribute

Route-map

- **可以为特定的目的在不同的命令中调用定义好的route-map**
  - suppress-map
  - unsuppress-map
  - advertise-map
  - inject-map
  - exist-map
  - non-exist-map
  - tabel-map

Route-map

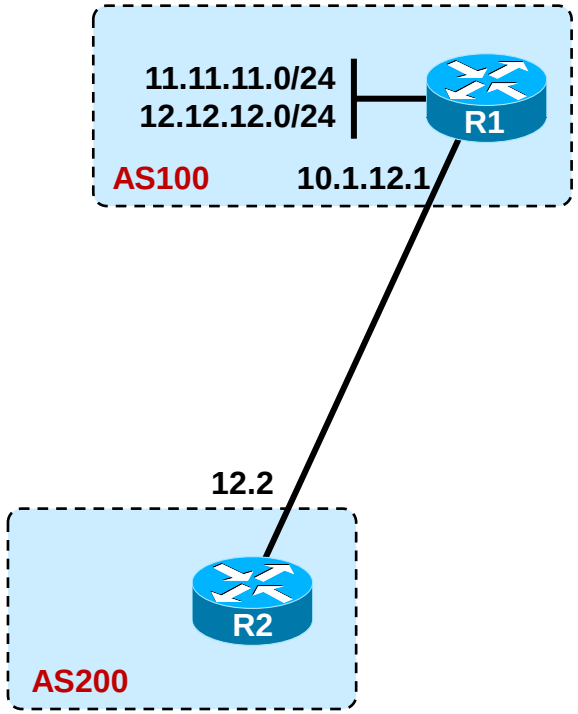
- **match语句能匹配**
  - Access-list
  - Ip prefix-list
  - Ip next-hop
  - local-preference
  - metric
  - Tag
  - AS_PATH
  - BGP community
  - IGP route-type(internal / external)
  - ……

Route-map

- **set语句能设置**
  - Origin
  - Weight
  - BGP community
  - LOCAL PREFERENCE
  - MED
  - ……

Route-map

- **配置示例 关联network 执行策略**



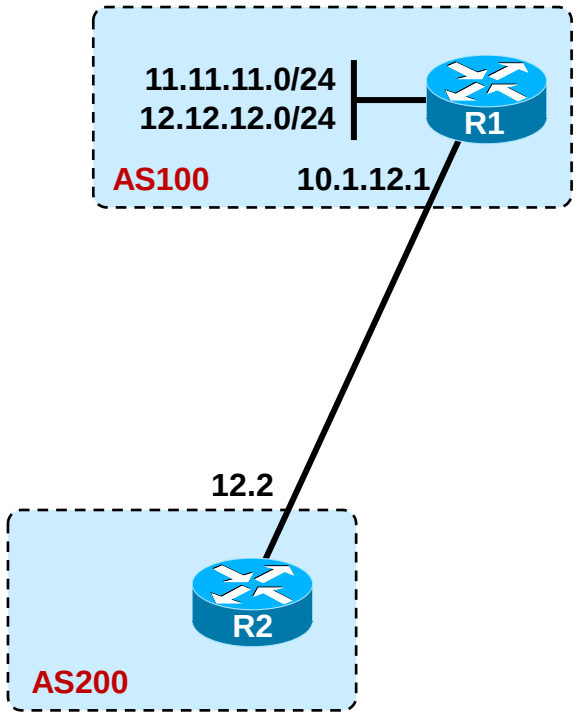
R1上，network引入路由时设置路径属性

```
ip prefix-list 11 permit 11.11.11.0/24
ip prefix-list 12 permit 12.12.12.0/24
route-map RP1 permit 10
  set community 100:11
route-map RP2 permit 20
  set community 100:12
router bgp 100
  network 11.11.11.0 mask 255.255.255.0 route-map RP1
  networl 12.12.12.0 mask 255.255.255.0 route-map RP2
  neighbor 10.1.12.2 send-community
```

该策略对所有BGP邻居有效

Route-map

- **配置示例 关联neighbor，针对特定邻居执行策略**

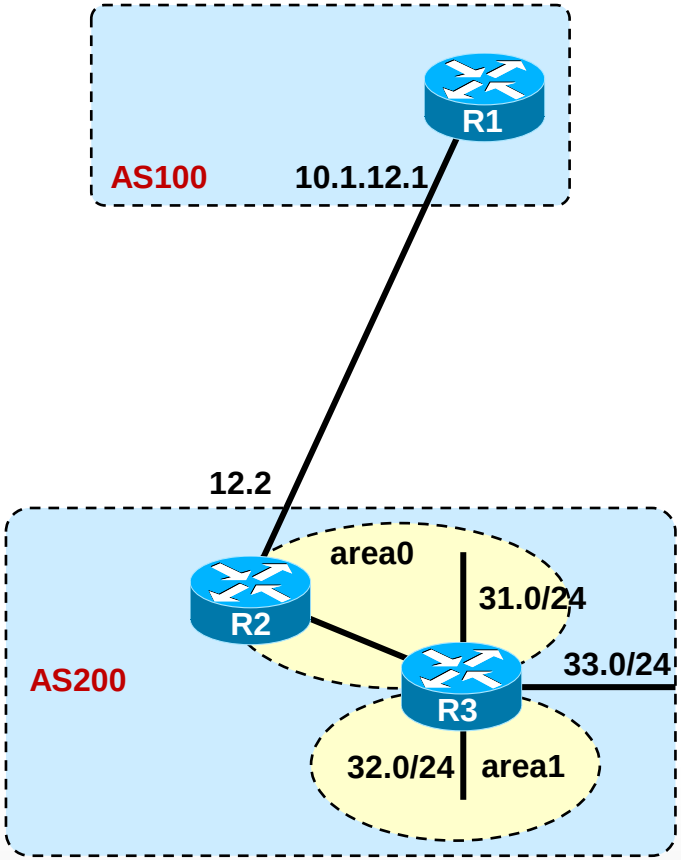


R1上，对R2传递路由时，设定MED属性值

```
ip prefix-list 11 permit 11.11.11.0/24
ip prefix-list 12 permit 12.12.12.0/24
route-map RP permit 10
  match ip address 11
  set metric 1000
route-map RP permit 20
  match ip address 12
  set metric 2000
router bgp 100
  neighbor 10.1.12.2 route-map RP out
```

Route-map

- **配置示例 重发布关联route-map**



R2上，将OSPF路由重发布进BGP

```
router bgp 200
 redistribute ospf 1 route-map RP match ?
 neighbor 10.1.12.1 remote-as 100
 no auto-summary
```

redistribute ospf 1

- 默认只重发布Intra-Area及Inter-Area路由
- match external只重发布E1及E2
- match external 1只重发布E1 ； external 2只重发布E2
- match nssa-external 只重发布NSSA外部路由

Route-map

- **policy-list**
  - 可预先将包含一组match语句的route-map定义成一个命令列表，这个列表称为policy-list
  - 这些policy-list可以在route-map中被调用
  - 一个policy-list就像个只包含match语句的route-map
  - 当route-map被执行，被其调用的policy-list中所包含的match语句将一并被遍历且执行match动作
  - 这是一种在大中型网络中运用、使得配置“模块化”的特性

Route-map

- **policy-list的特性**
  - Ipv6不支持
  - 12.0(22)S 和12.2(15)T之前的CISCO IOS版本不支持该特性，另外更老的IOS版本的路由器重启存在路由策略配置丢失的风险
  - Policy-list中不能包含set语句，但是它被route-map调用后，该route-map中可以包含set语句
  - Policy-list只在BGP中支持，其他的IP路由协议并不支持这个特性

Route-map

- **policy-list的配置**

```
ip policy-list as100 permit
  match as-path 1
  match community 1
```

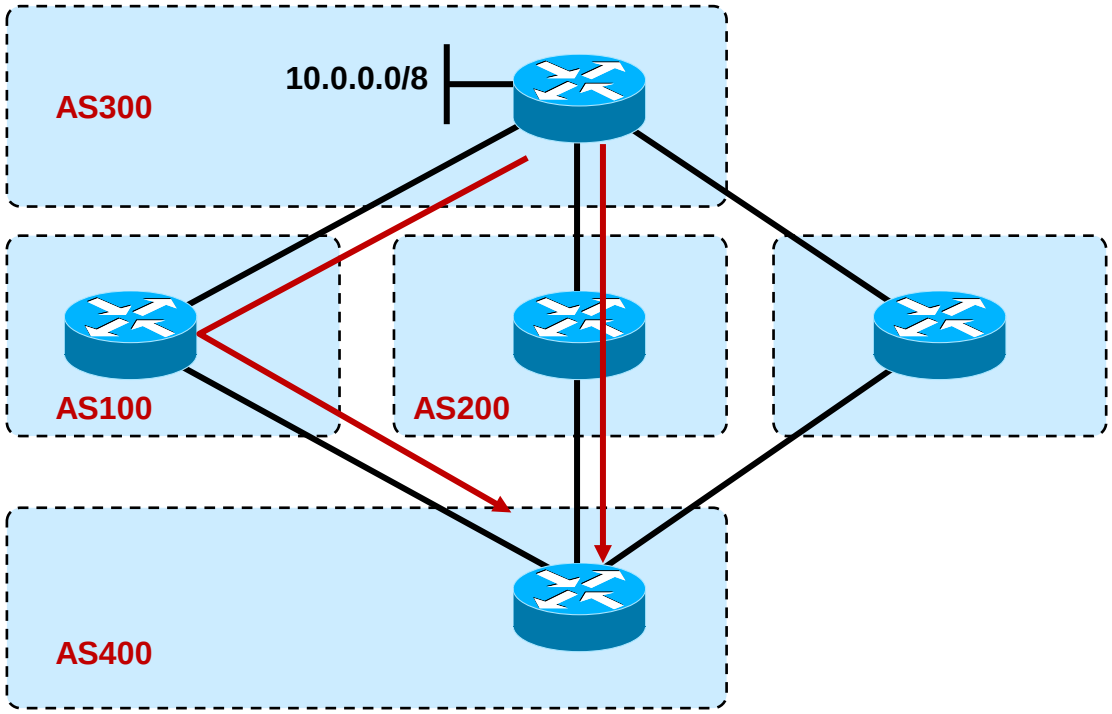
- 上述命令创建一个policy-list
- Policy-list只能包含match语句

```
route-map RP permit 10
  match policy-list as100
  match ip address prefix-list 100
  set local-preference 300
```

- 上述命令在route-map中调用定义好的policy-list
- 一个route-map中可调用多个policy-list

Route-map

- **policy-list配置示例**



Route-map

- **policy-list配置示例**

```
ip prefix-list 1 permit 10.0.0.0/8
ip as-path access-list 1 permit ^100_
ip as-path access-list 2 permit ^200_
ip community-list 1 permit 300:105
```

```
ip policy-list as100 permit
  match as-path 1
  match community 1
```

```
ip policy-list as200 permit
  match as-path 2
  match community 1
```

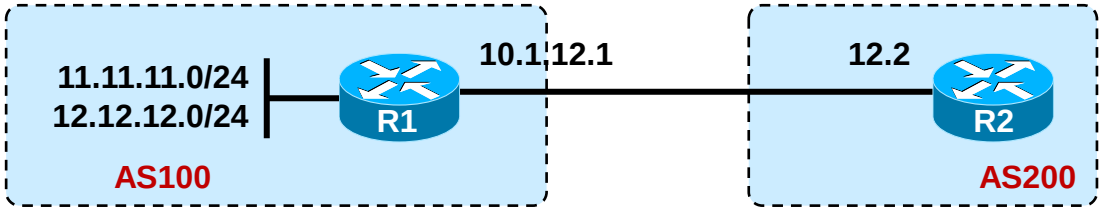
逻辑的或

```
route-map Test permit 10
  match ip address prefix-list 1
  match policy-list as100 as200
  set metric 1000
route-map Test permit 20
```

advertise-map

advertise-map

- **配置示例**

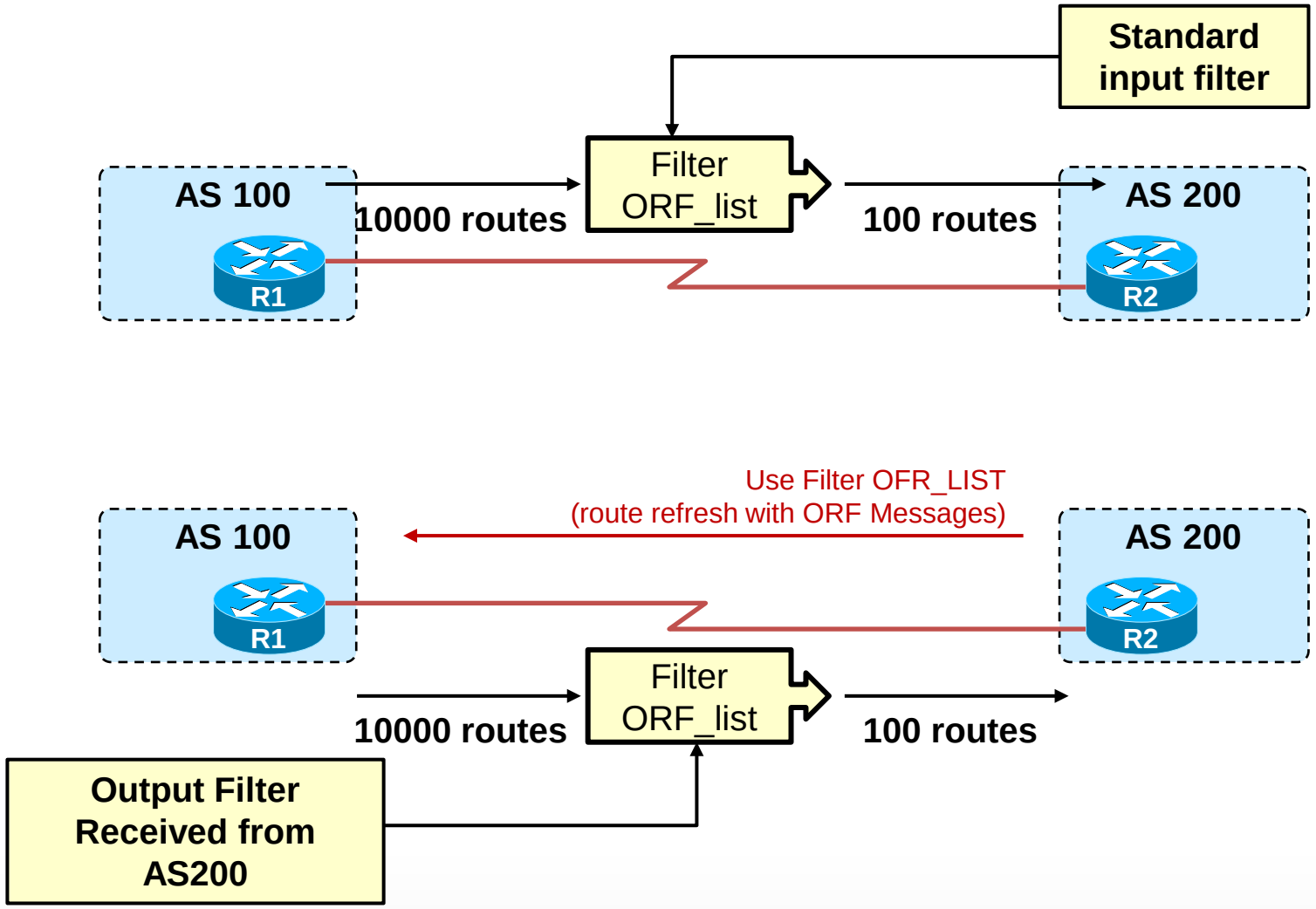



```
ip prefix-list 1 permit 11.11.11.0/24
ip prefix-list 2 permit 12.12.12.0/24
route-map RP1 permit 10
  match ip address prefix-list 1
route-map RP2 permit 10
  match ip address prefix-list 2
router bgp 100
  neighbor 10.1.12.2 advertise-map RP1 non-exist-map RP2
```

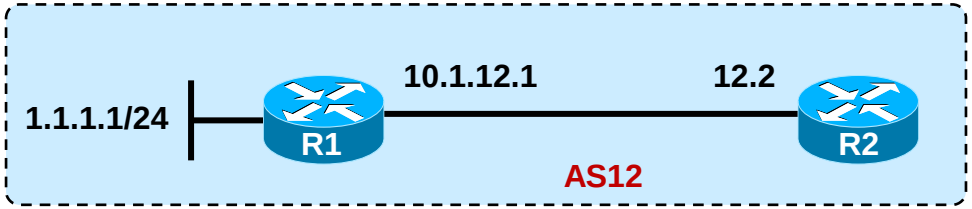
红茶三杯
Vinsoney

ORF

ORF



ORF的配置



R2的配置（sender）：

```
router bgp 12
  address-family ipv4 unicast
  neighbor 10.1.12.1 capability orf prefix-list send
  neighbor 10.1.12.1 prefix-list FILTER in

ip prefix-list FILTER deny 1.1.1.0/24
ip prefix-list FILTER permit
```

R1的配置（receiver）：

```
router bgp 12
  address-family ipv4 unicast
  neighbor 10.1.12.2 capability orf prefix-list receive
```

注意：省去基本配置，如手工指定BGP邻居

ORF

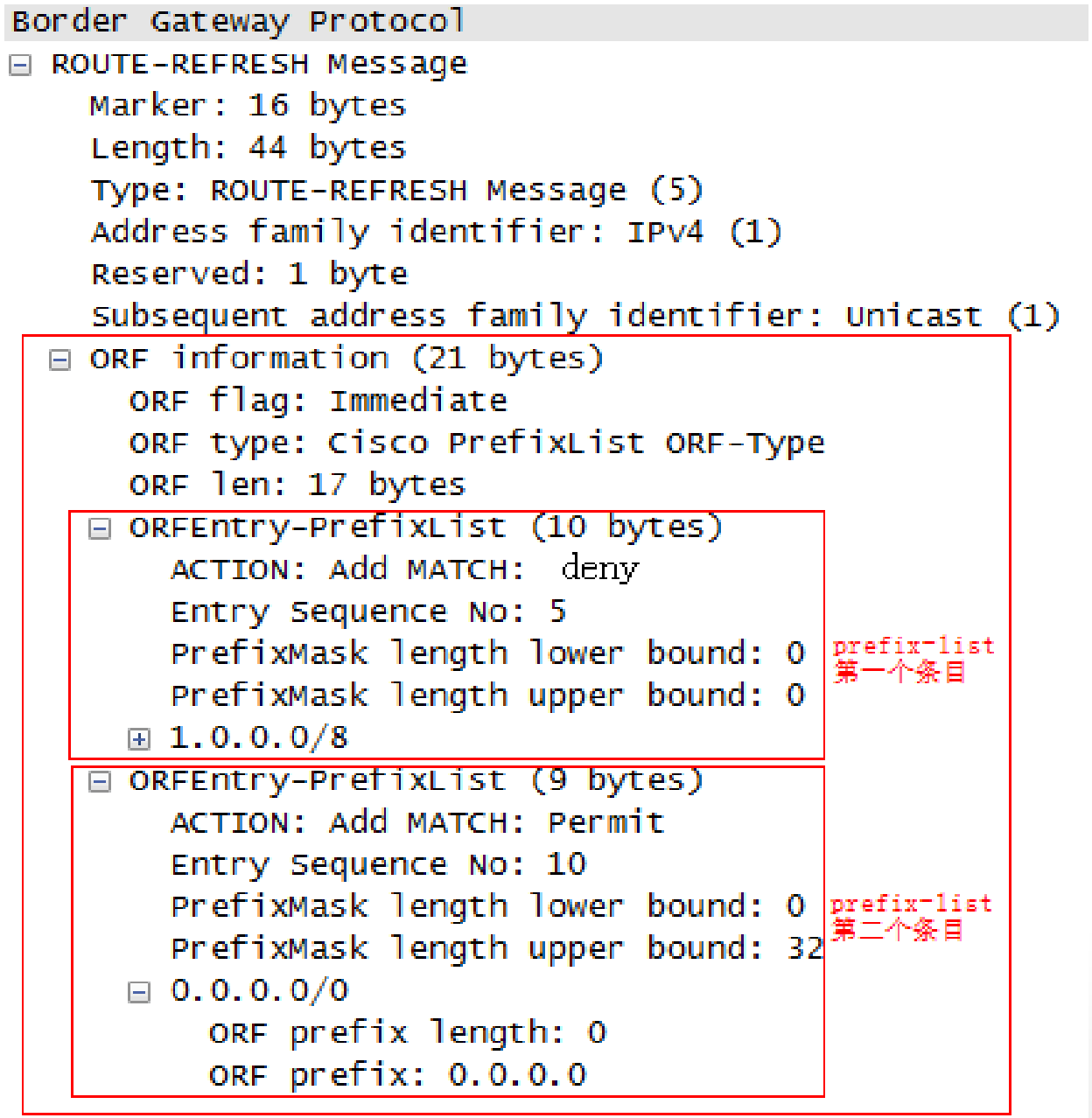
- **ORF消息包含以下内容：**
  - AFI / SAFI ipv4 unicast
  - ORF TYPE
  - When to refresh
  - List of ORF entries
- **ORF类型不同，ORF entries也不尽相同**
- **每种ORF类型都需要进行ORF能力的协商**

ORF

- **ORF type**
  - Type=1 NLRI
    - Filters based on the prefix
  - Type=2 Communities
    - Filters based on standard BGP community attributes
  - Type=3 Extended Communities
    - Filters based on extended BGP community attibutes
  - Type=128 Prefix-list
    - Filters based on Cisco implmentation of prefix filtering

ORF

- **ORF type**

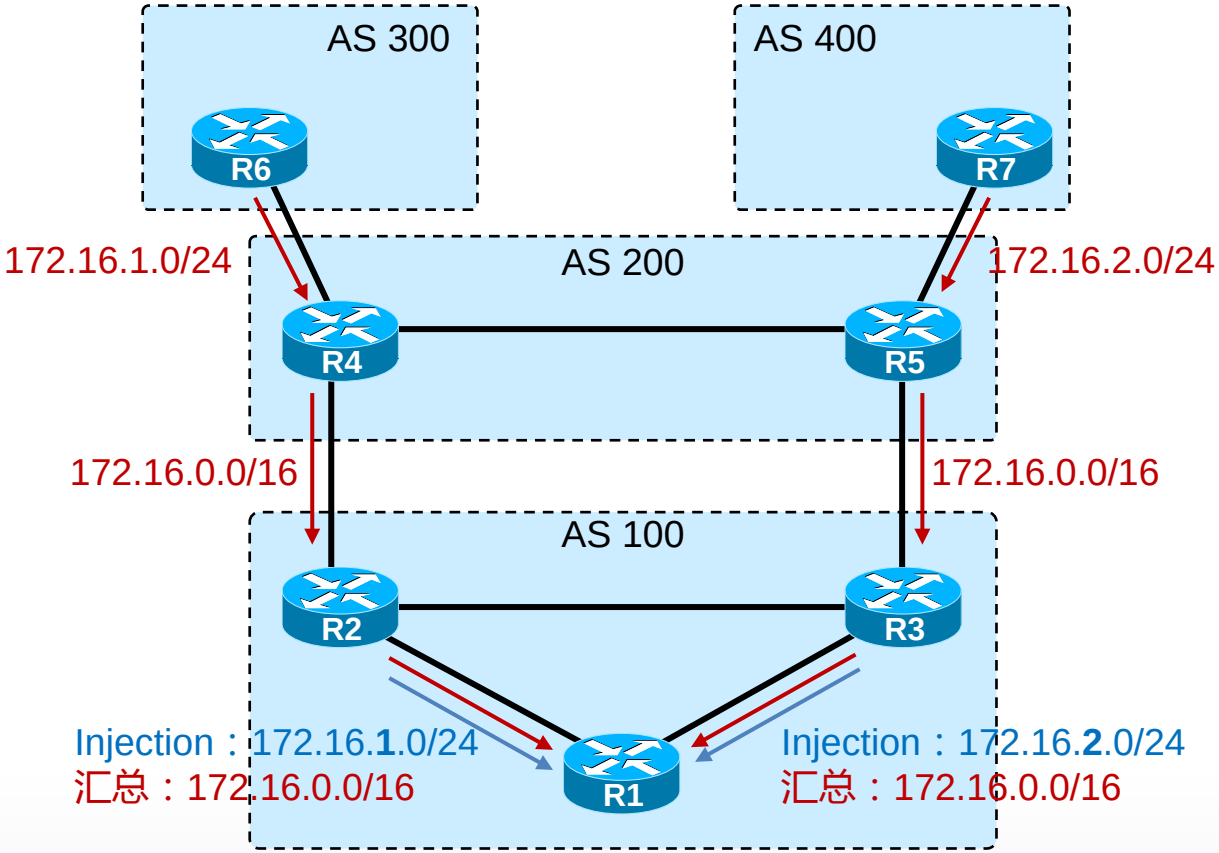


红茶三杯
Vinsoney

路由拆分 BGP Deaggregation

路由拆分 BGP Deaggregation

- **背景**



路由拆分 BGP Deaggregation

- **背景**

拆分可以通过使用条件注入（conditional injection）来完成，所谓的conditional injection指的就是，当特定的汇总路由存在时，我可以生成其下属的特定明细，这些明细路由将被注入到本地BGP RIB（本地路由表也会加载明细路由信息），以便在本地AS中提供比汇总路由更详细的路由选择信息（更长的前缀）。

路由拆分 BGP Deaggregation

- **配置**

Conditional inject的配置如下（BGP路由选择进程模式下）：

```
bgp inject-map map1 exist-map map2 [copy attributes]
```

上述命令的意思是当map2所匹配的汇总路由正常时，在本地BGP RIB中注入map1中定义的明细路由。当汇总路由挂掉，这条明细也就跟着消失，这就是所谓的条件注入—conditional injection。下面我们在看来一下这两个route-map的详细内容，这些是需要格外注意的。

路由拆分 BGP Deaggregation

- **配置 cont.**

  - exist-map使用的route-map最少具有以下两个match语句：

match ip address prefix-list

上面这条match语句用来匹配汇总路由

match ip route-source

上面这条match语句用来匹配发送该汇总路由的邻居IP。如果指定了copy attributes选项，那么被inject的明细路由会继承汇总路由的路径属性，否则明细将被当成本地生成的路由。

路由拆分 BGP Deaggregation

- **配置 cont.**

  - Inject-map使用的route-map中

Set ip address prefix-list

上面的这条set命令用来定义将被注入到本地BGP RIB的明细路由。被注入的前缀可以使用

Show ip bgp injected-path来显示

路由拆分 BGP Deaggregation

- **配置 cont.**

R2的配置如下：

```
ip prefix-list huizong permit 172.16.0.0/16   //用来匹配汇总路由
ip prefix-list mingxi permit 172.16.1.0/24          //用来定义准备注入的条件前缀
ip prefix-list xiayitiao permit 10.1.24.4/32   //用来匹配传递给我汇总路由的BGP邻居，这里是R4的IP
route-map RP_mingxi permit 10
  set community 100:200 no-export//100:200表示这是针对AS200的
  set ip address prefix-list mingxi
route-map RP_huizong permit 10
  match ip address prefix-list huizong
  match ip route-source xiayitiao
router bgp 300
  bgp inject-map RP_mingxi exist-map RP_huizong copy-attributes
  neighbor 10.1.23.2 remote-as 200
```

红茶三杯
Vinsoney

学习 沉淀 成长 分享

关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

路由反射器及联邦

红茶三杯（朱SIR）

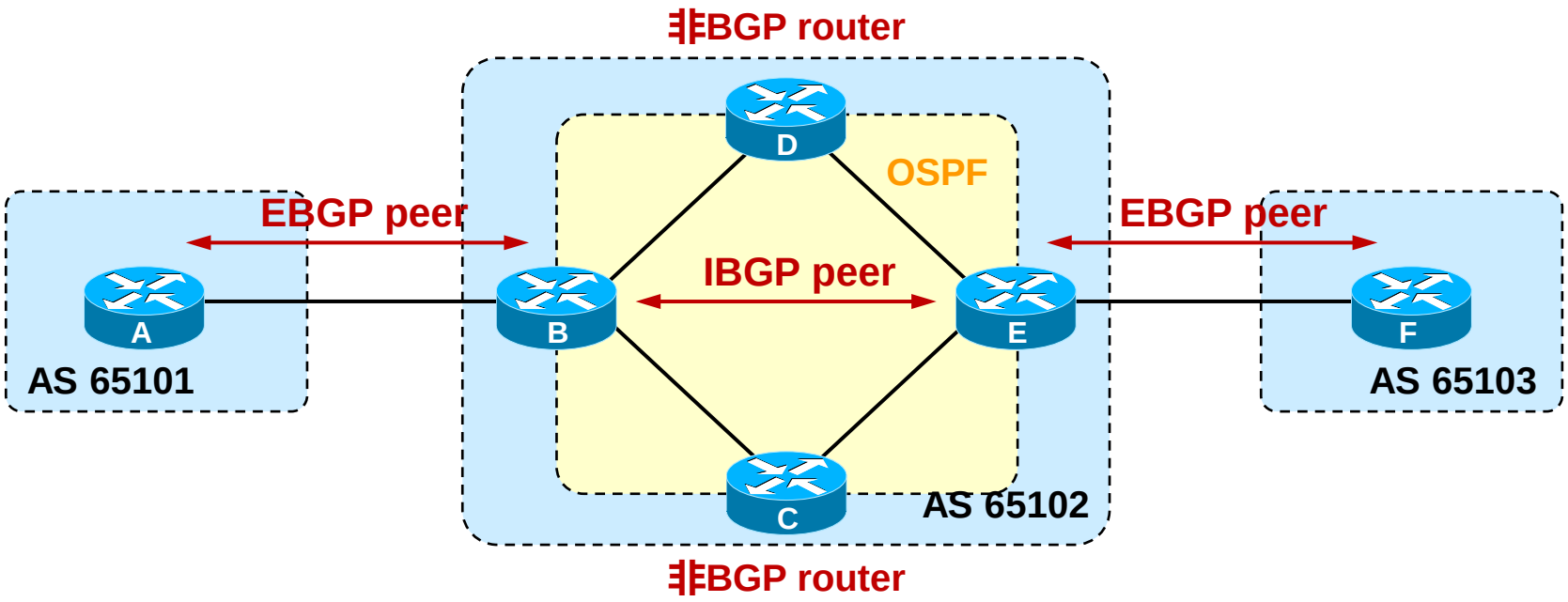
Latest update: 2012-08-01

课程目标

- 路由反射器
- BGP联邦

路由反射器

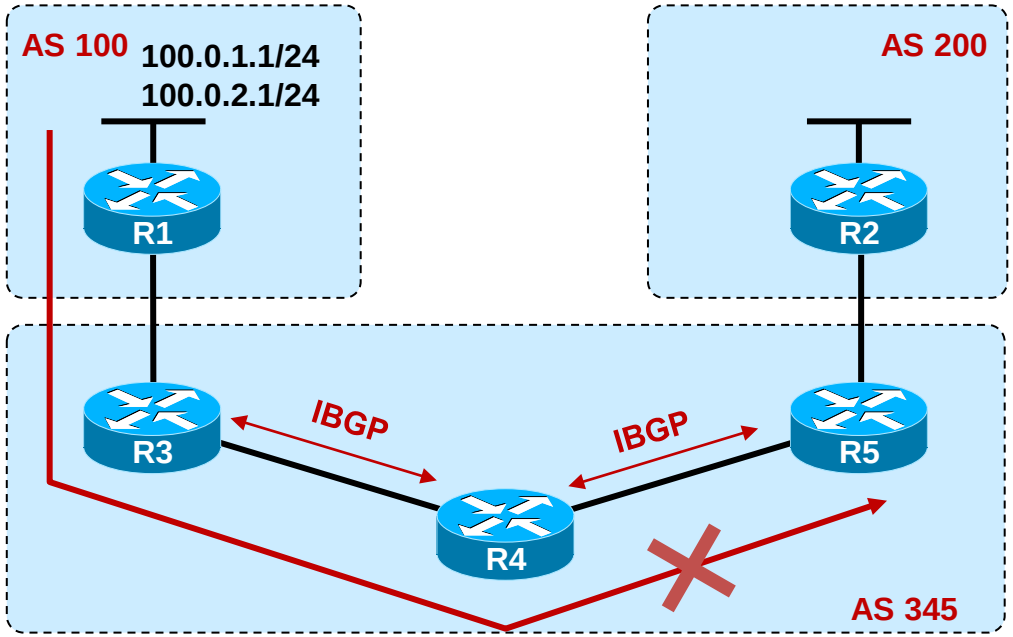
- 中转AS中的IBGP问题



路由反射器

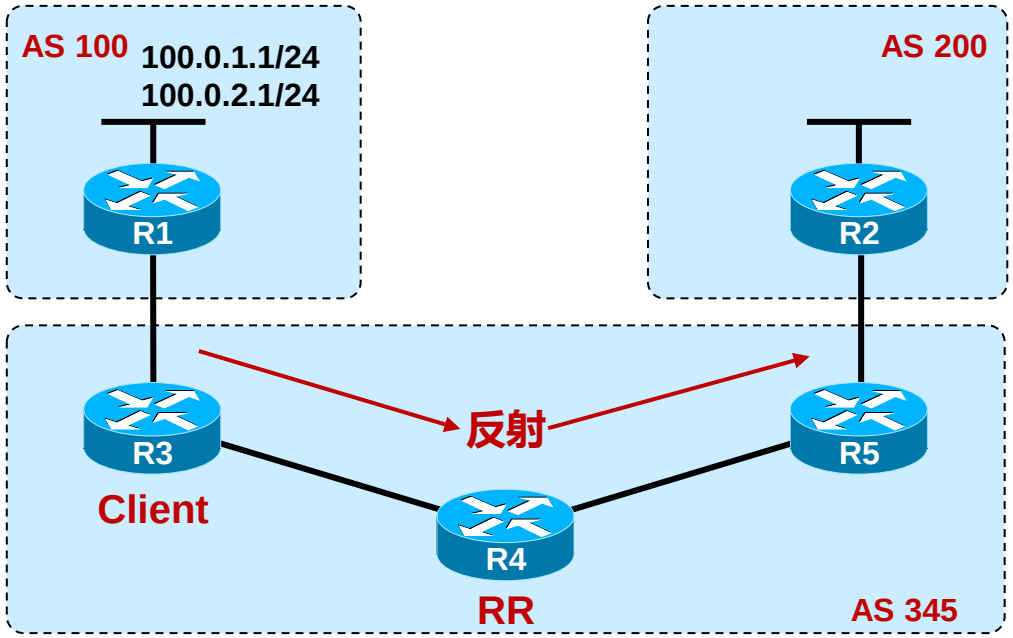
- **中转AS中的IBGP问题**
  - AS内要求IBGP全互联（IBGP水平分割）
  - BGP Routers
    - 需维护大量的TCP及BGP连接
    - 网络中充斥着BGP路由信息
  - 解决方案
    - 路由反射器
    - BGP联邦

路由反射器技术背景



- 因为IBGP水平分割原则，导致AS内部需要维护大量的BGP连接（要求IBGP全互联），从而影响网络性能，路由反射器可以“放宽”水平分割原则，缓解该问题。

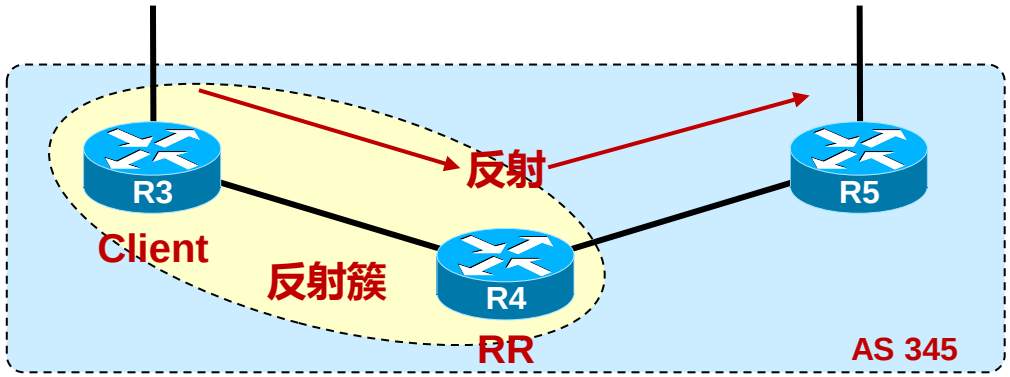
路由反射器技术背景



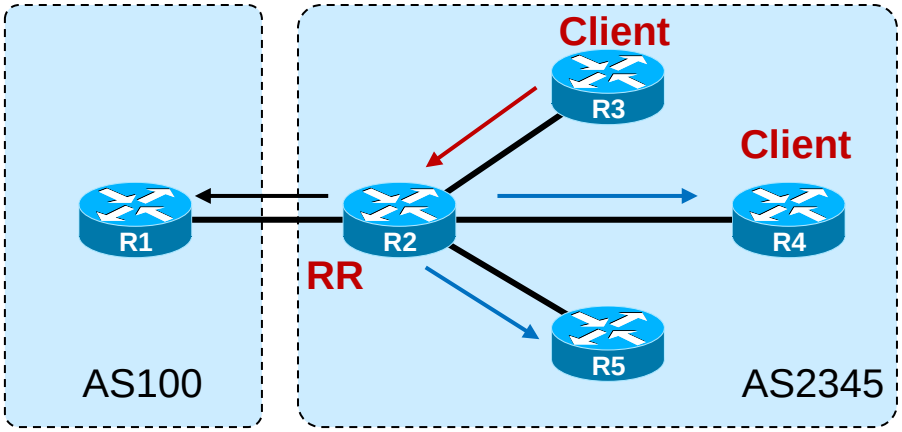
- Route Reflector
- Client

路由反射规则

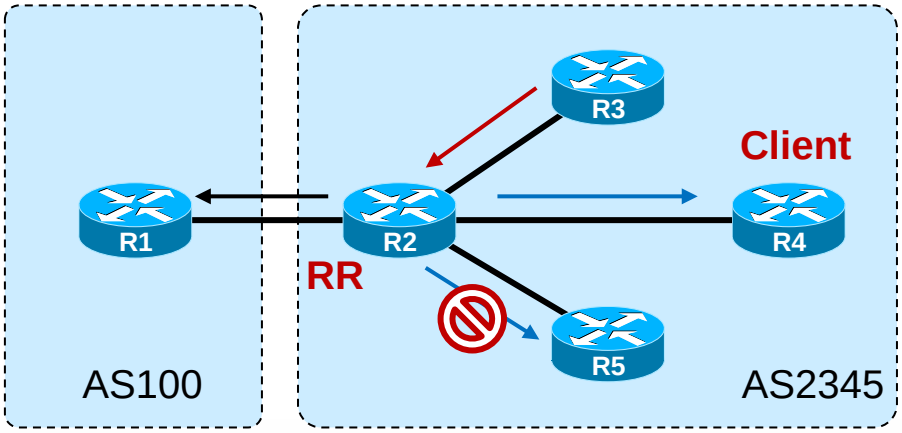
- 如果路由学习自非client IBGP对等体，则反射给所有client
- 如果路由学习自一client，则反射给所有非client IBGP邻居和除了该client以外的所有client
- 如果路由学习自EBGP邻居，则发送给所有client和非client IBGP邻居



路由反射器 规则示例1

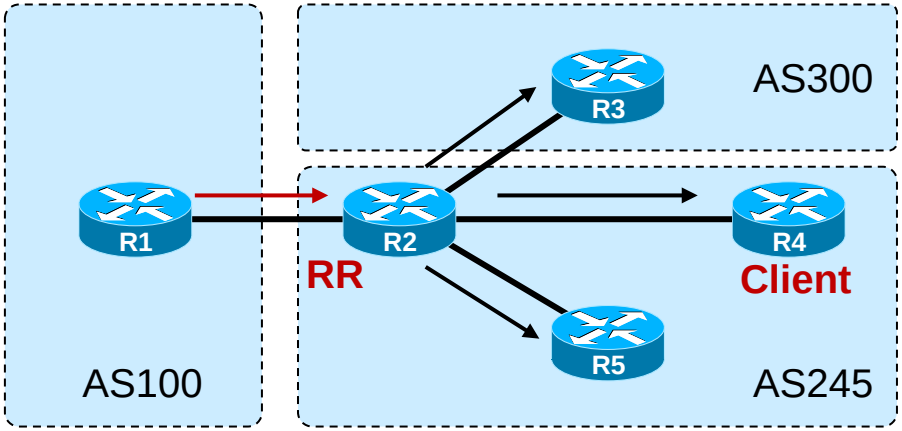


- 如果路由学习自一client，则反射给所有非client IBGP邻居和除了该client以外的所有client



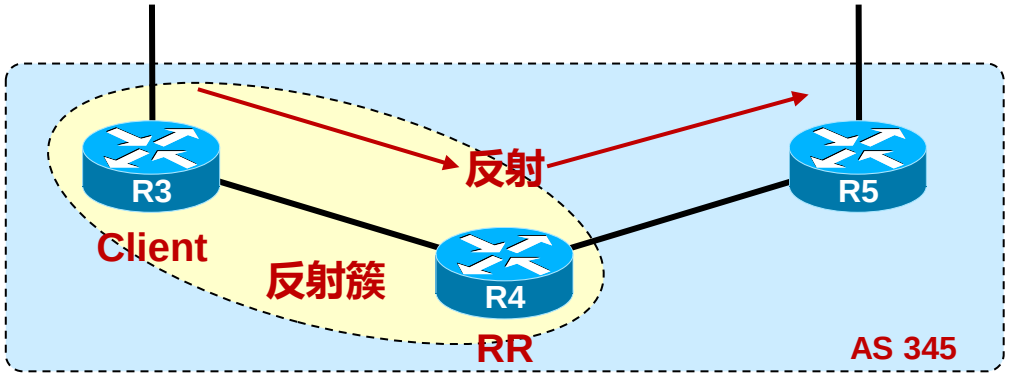
- 如果路由学习自非client IBGP对等体，则反射给所有client

路由反射器 规则示例2



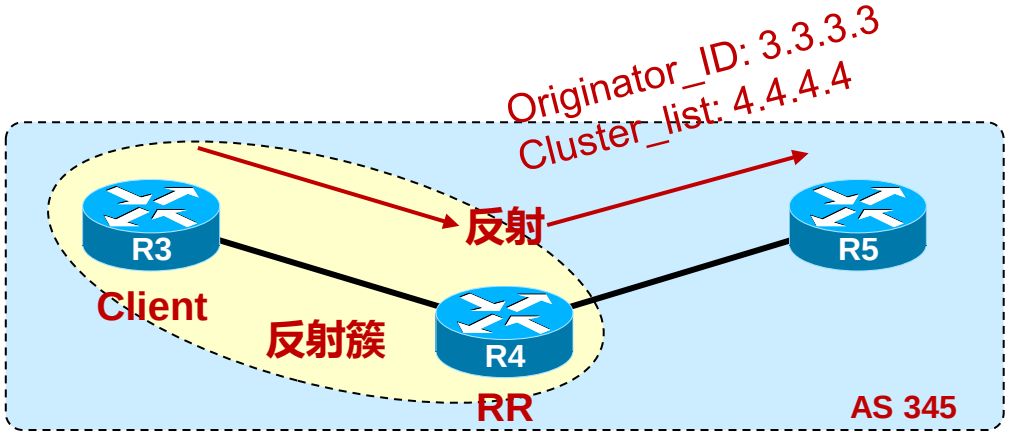
- 如果路由学习自己EBGP邻居，则发送给所有client和非client IBGP邻居

路由反射器环境下的防环



由于AS_PATH属性在AS内部不会发生变化（仅当路由离开本AS才会被更新），因此AS内才有水平分割的机制用于防止环路，而路由反射器实际上是放宽了水平分割原则，这个就会给环路带来一定的隐患，因此路由反射器需使用以下两个属性防止环路：**ORIGINATOR_ID**和**CLUSTER_LIST**是路由反射器使用的可选非传递属性，用来防止环路。

Originator_ID、Cluster_list

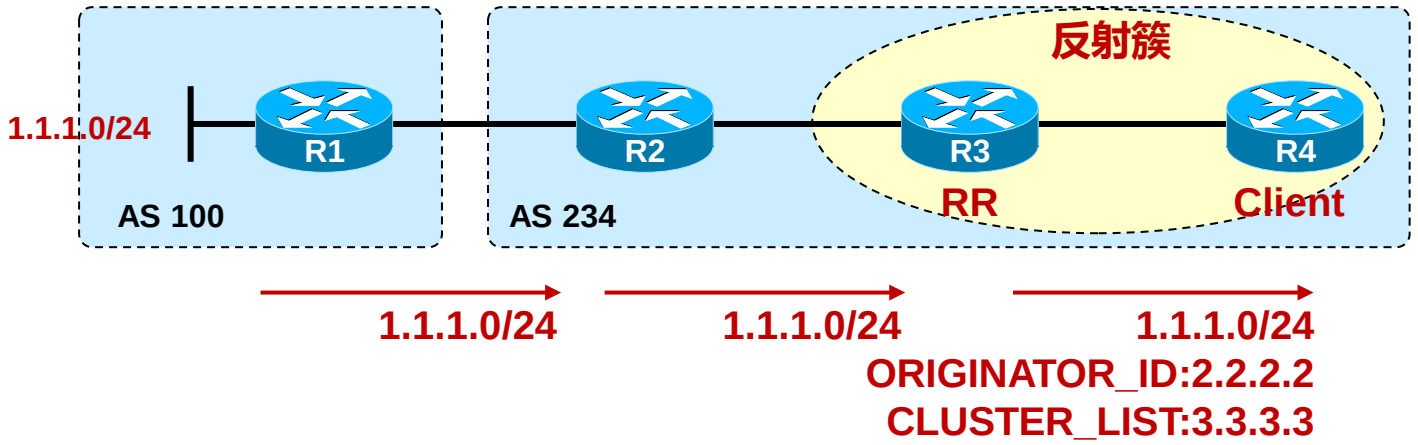


Originator_ID、Cluster_list

- **Originator_ID**
  - 每当一条路由被路由反射器反射时，该路由的始发IBGP路由器的Router-ID将会被存在路由的originator_ID属性中
  - 当一台路由器收到IBGP路由且其originator_ID与该路由器的Router_ID相同，则路由器忽略该条路由
  - Originator_ID及Cluster-list属性将会影响BGP路径决策

Originator_ID、Cluster_list

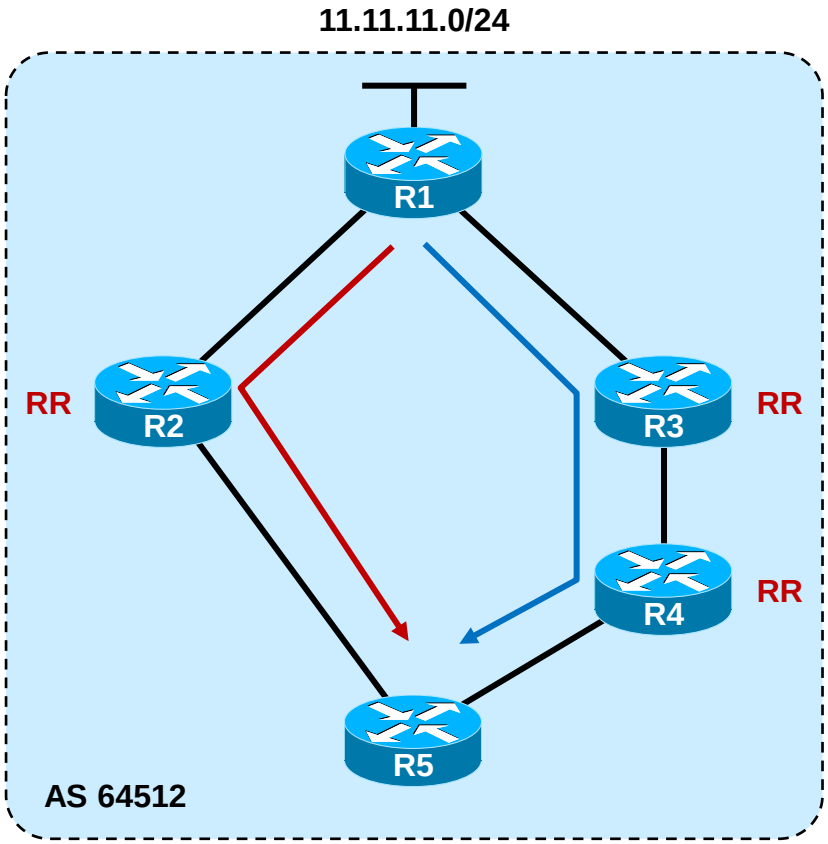
- **Originator_ID的值**



路由反射簇

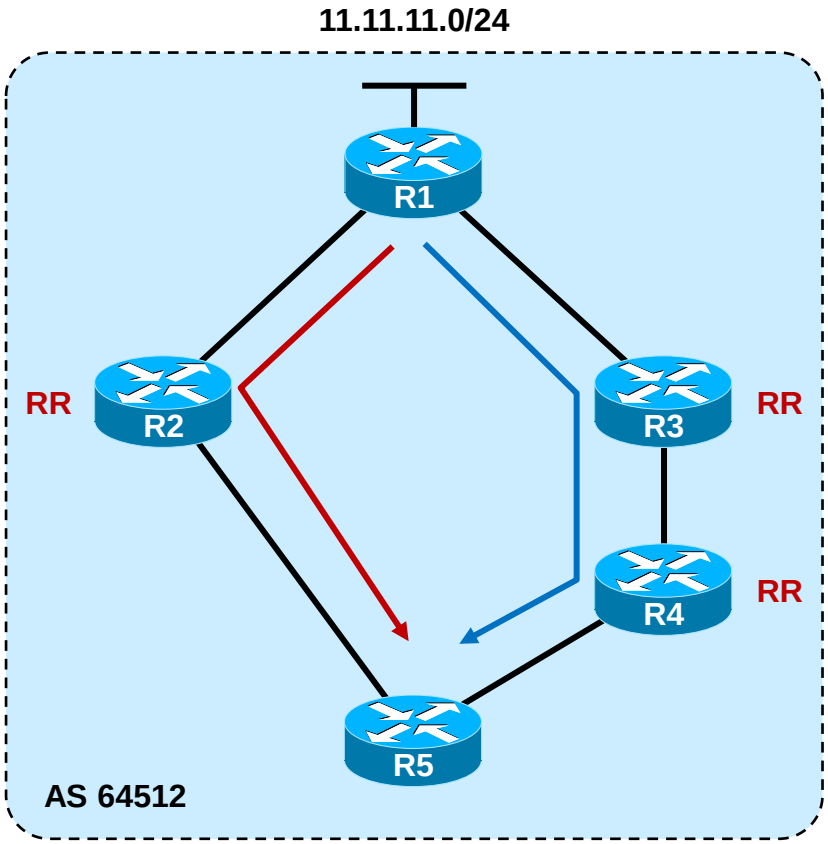
- 路由反射簇包括反射器及其Client
- 每一个簇都有唯一的簇ID
- 每当一条路由被反射器反射后，该反射器（该簇）的Cluster_ID就会被添加至路由的Cluster_list属性中
- 每当反射器收到一条Cluster_list属性已经包含该簇的Cluster_ID的路由时，该路由基于防环的目的将不被反射

Originator_ID、Cluster_list



环境描述：
R1是R3 RR的Client
R3是R4 RR的Client
所有的路由器Loopback口地址为x.x.x.x，x为设备编号，IBGP邻居关系基于该Loopback口建立

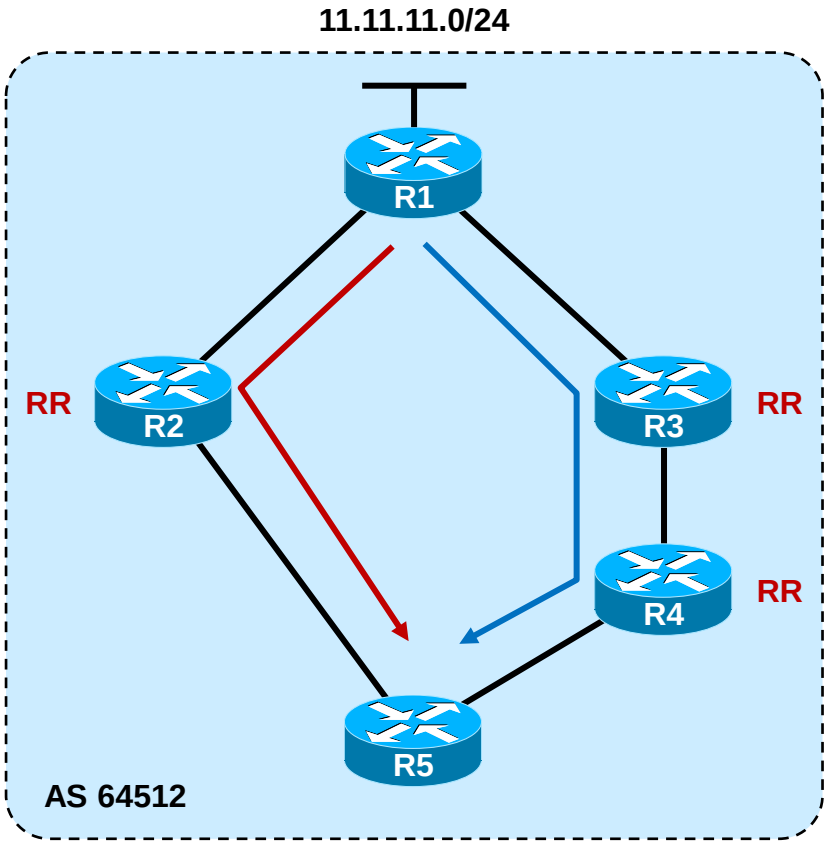
Originator_ID、Cluster_list



R4 show ip bgp 11.11.11.0

BGP routing table entry for 11.11.11.0/24, version 2
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
  Advertised to update-groups:
     1
  Local, (Received from a RR-client)
    1.1.1.1 (metric 129) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 1.1.1.1, Cluster list: 3.3.3.3

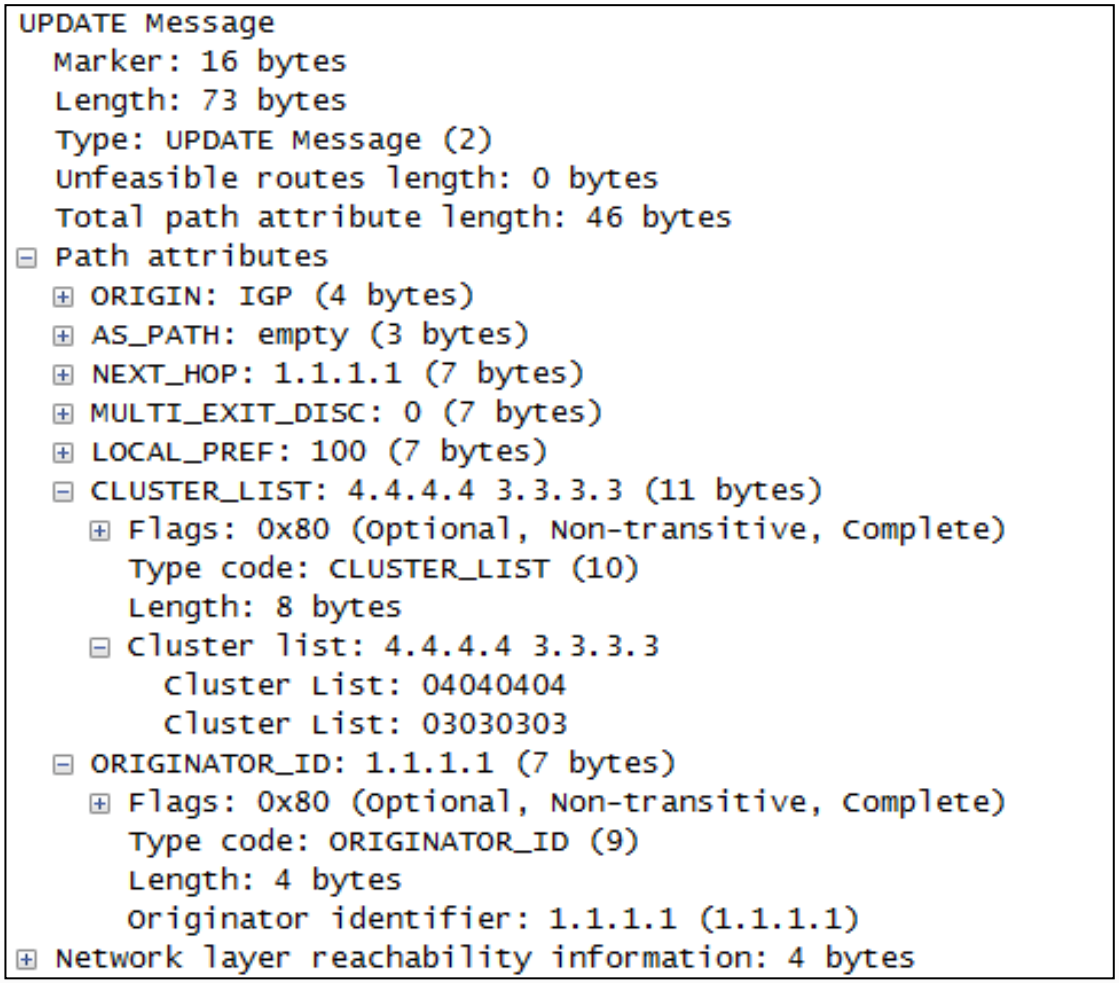
Originator_ID、Cluster_list



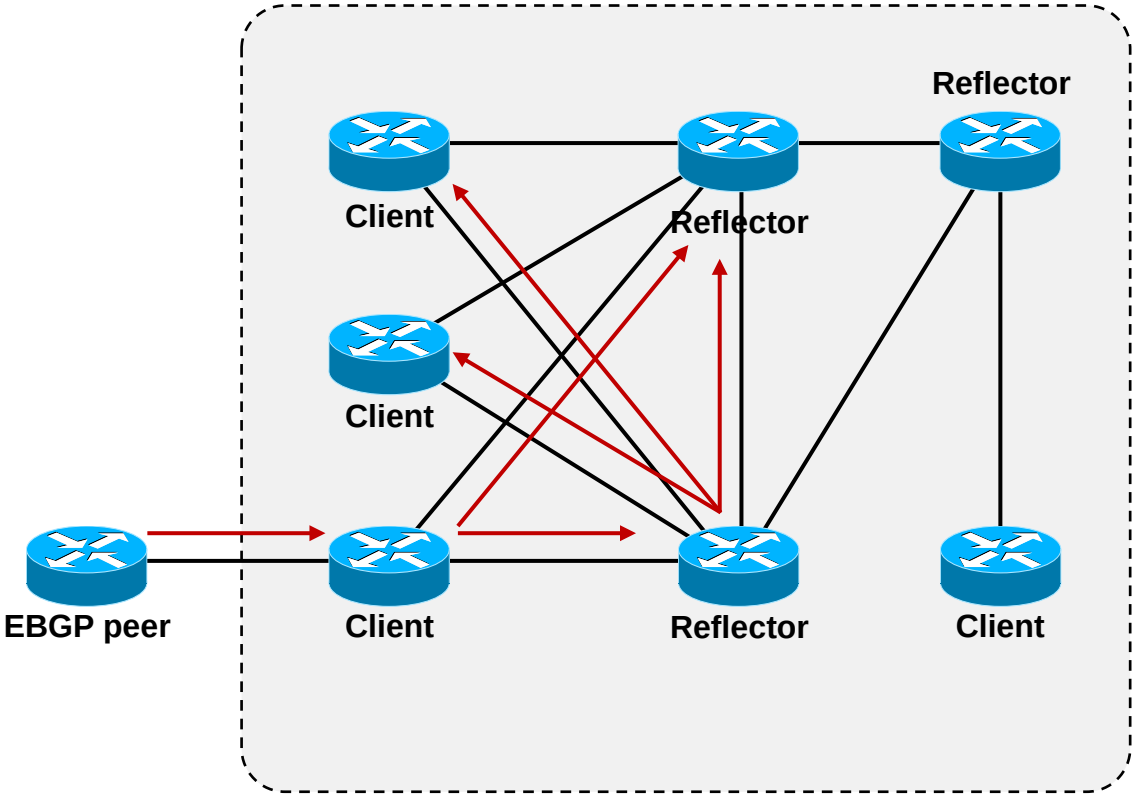
R5 show ip bgp 11.11.11.0

BGP routing table entry for 11.11.11.0/24, version 2
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Flag: 0x820
  Not advertised to any peer
  Local
    1.1.1.1 (metric 129) from 4.4.4.4 (4.4.4.4)
      Origin IGP, metric 0, localpref 100, valid, internal
      Originator: 1.1.1.1, Cluster list: 4.4.4.4, 3.3.3.3
  Local
    1.1.1.1 (metric 129) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, **best**
      Originator: 1.1.1.1, Cluster list: 2.2.2.2

Originator_ID、Cluster_list

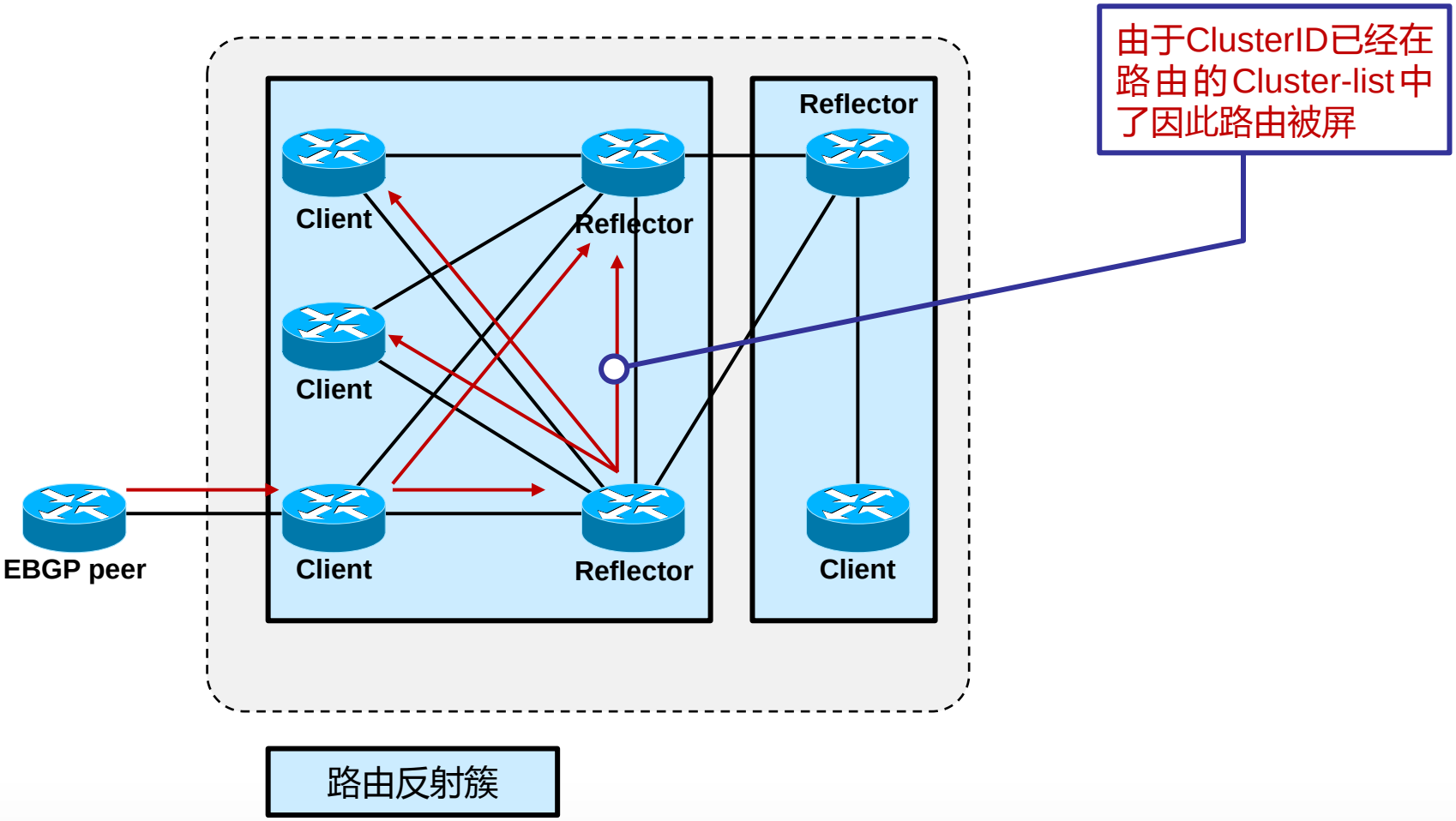


路由反射器的冗余

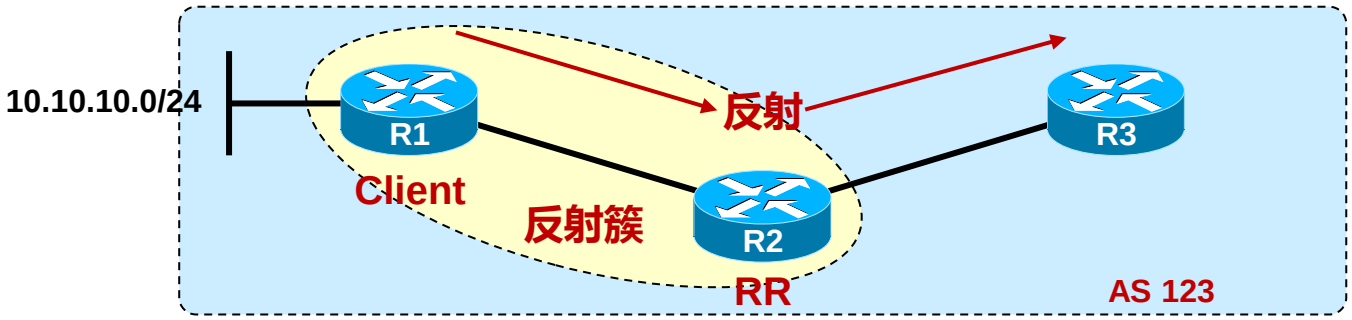


- **冗余RR增加了网络的健壮性**
- **使用Originator_ID、Cluster_list属性来在冗余RR环境中避免路由环路。**
  - 例如将两个RR的Cluster_ID配置为一样，那么可以起到进一步的防环作用
  - 所有的RR之间建议采用IBGP全互联
- **Client会收到来自两个RR反射的路由，如何决策？**

路由反射器的冗余


由于ClusterID已经在路由的Cluster-list中了因此路由被屏
Reflector
Client
Reflector
Client
EBGP peer
Client
Reflector
Client
路由反射簇

路由反射器的配置

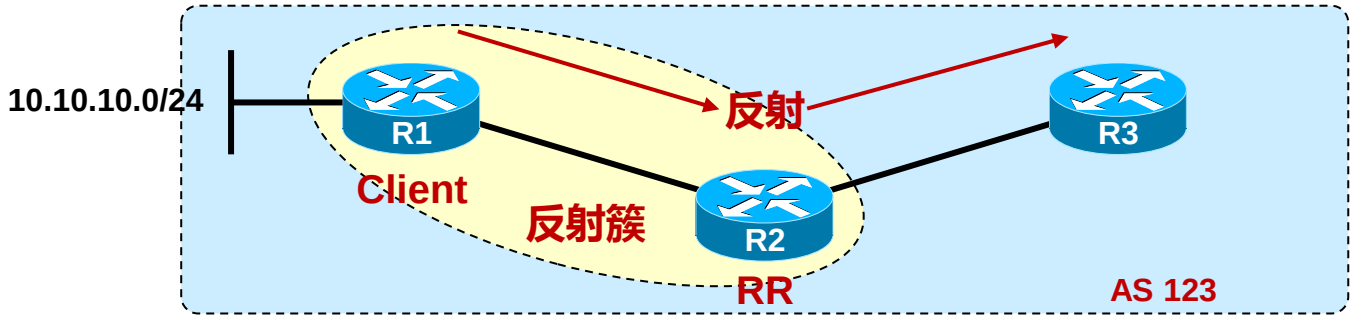


R2的BGP配置如下：

```
router bgp 123
 neighbor 1.1.1.1 remote-as 123
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.1.1.1 route-reflector-client
```

- Client并不知道自己属于反射簇，只有RR知道

路由反射器的配置



R3#show ip bgp 10.10.10.0

BGP routing table entry for 10.10.10.0/24, version 7

Path 1, ... IP-Routing-Table)

Flag: 0x820

Not advertised to any peer

Local

1.1.1.1 (metric 129) from 2.2.2.2 (2.2.2.2)

Origin IGP, metric 0, localpref 100, valid, internal, best

Originator: 1.1.1.1, Cluster list: 2.2.2.2

到达该NEXT_HOP的metric(IGP)

BGP邻居的更新源地址

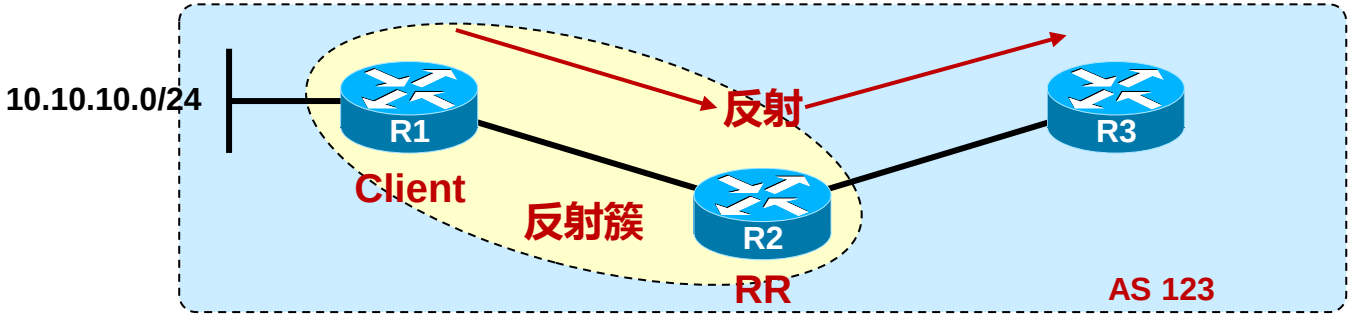
NEXT_HOP

邻居的RouterID

路由始发者的RouterID

默认是RR的RouterID

路由反射器的配置



RR可修改cluster-id

```
router bgp 123
  bgp cluster-id 222.222.222.222
```

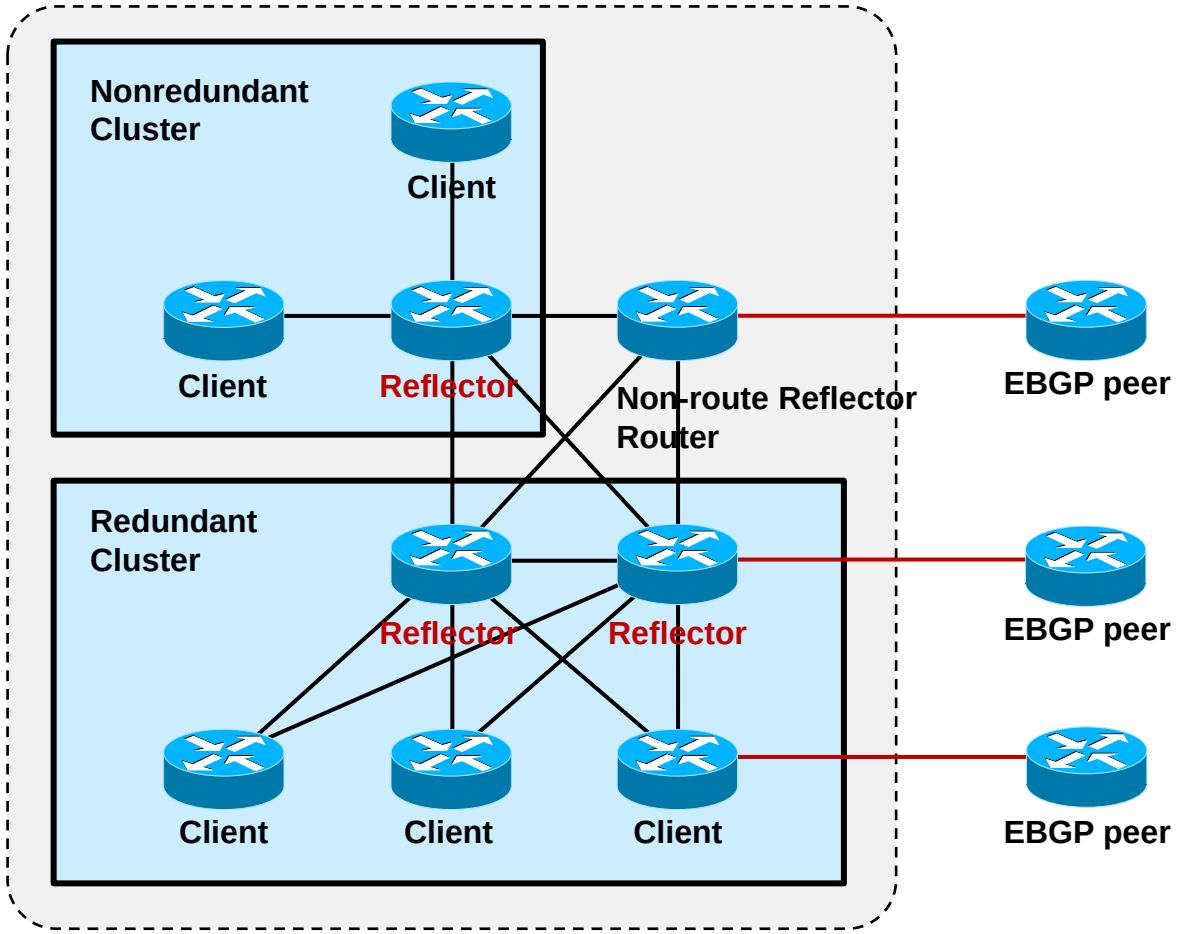
路由反射器的实施

- 路由反射器将传输AS（transit AS）分割成小单元，也就是反射簇
- 每个簇包含反射器及其client
- 不支持路由反射器功能的路由器（不支持RR特性）可以充当单路由器簇或充当client

路由反射器的实施

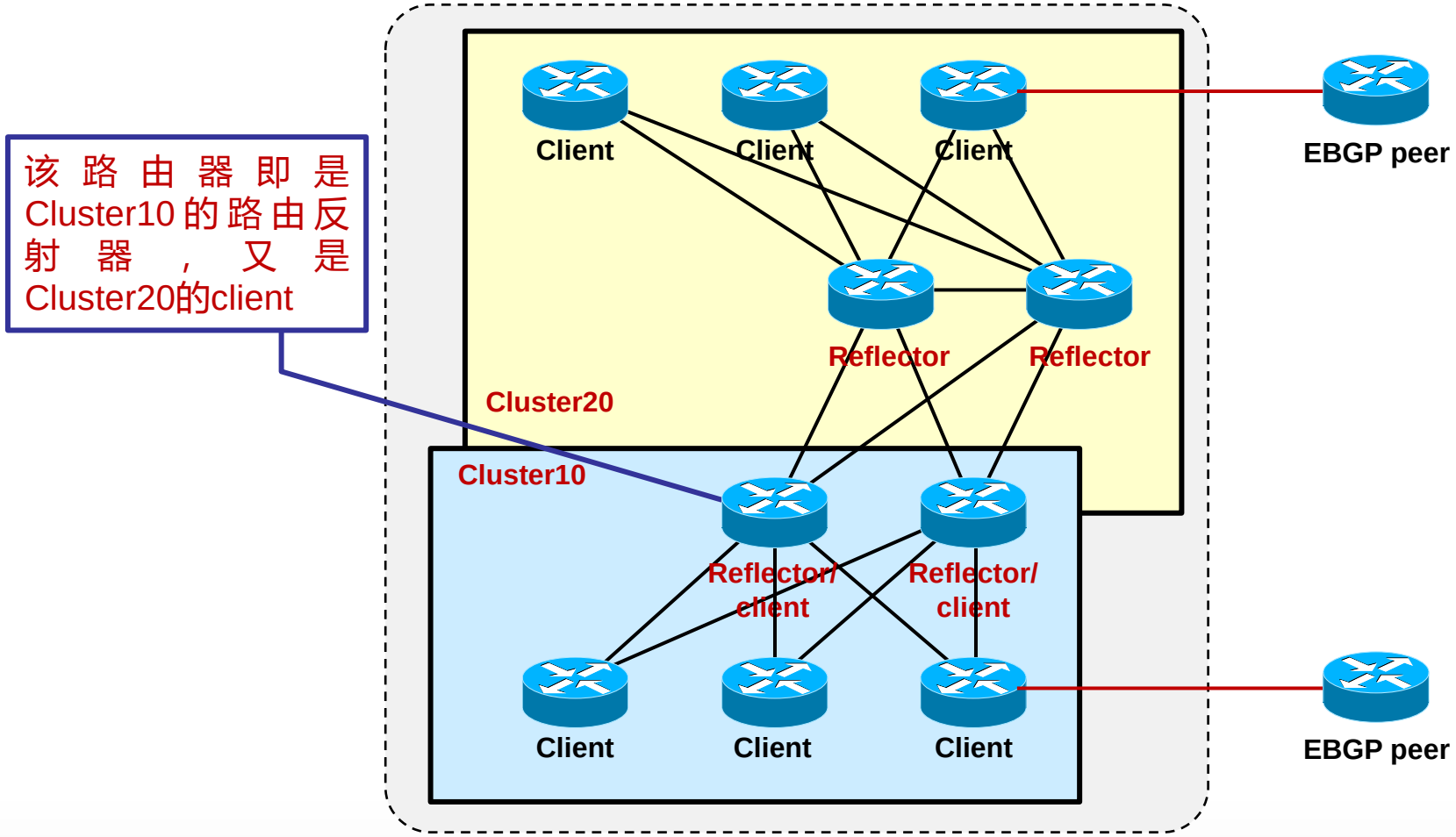
- **IBGP session原则**
  - 路由反射簇中的所有client都应该并且只与簇中所有的RR建立IBGP连接
  - AS内的路由反射器之间要求全IBGP互联
  - 非反射器的路由器既可参与IBGP全互联也可配置为反射器的client

路由反射器的实施



路由反射簇

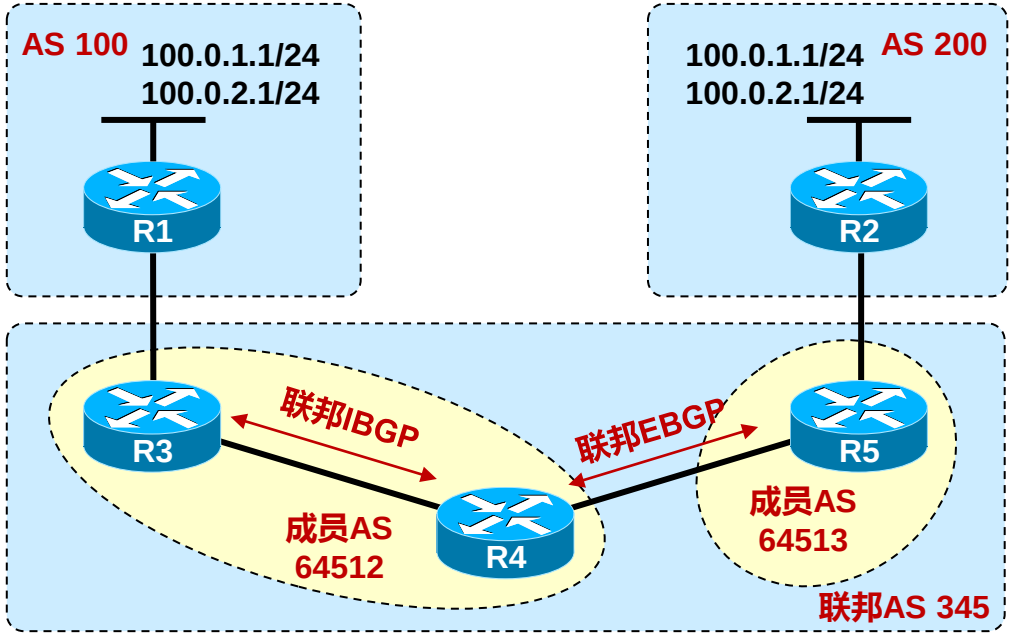
层次化的路由反射器



红茶三杯
Vinsoney

BGP联邦

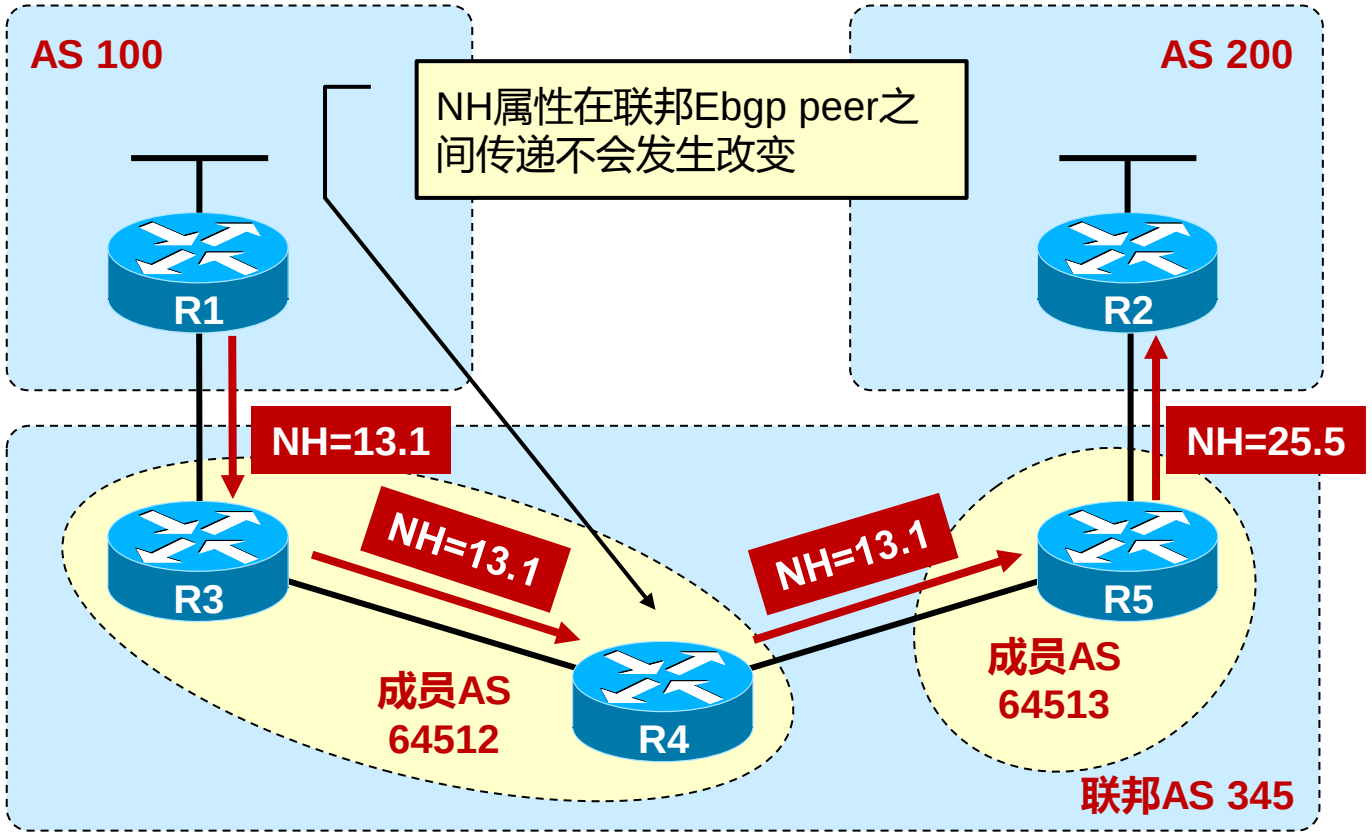
联邦BGP Confederation



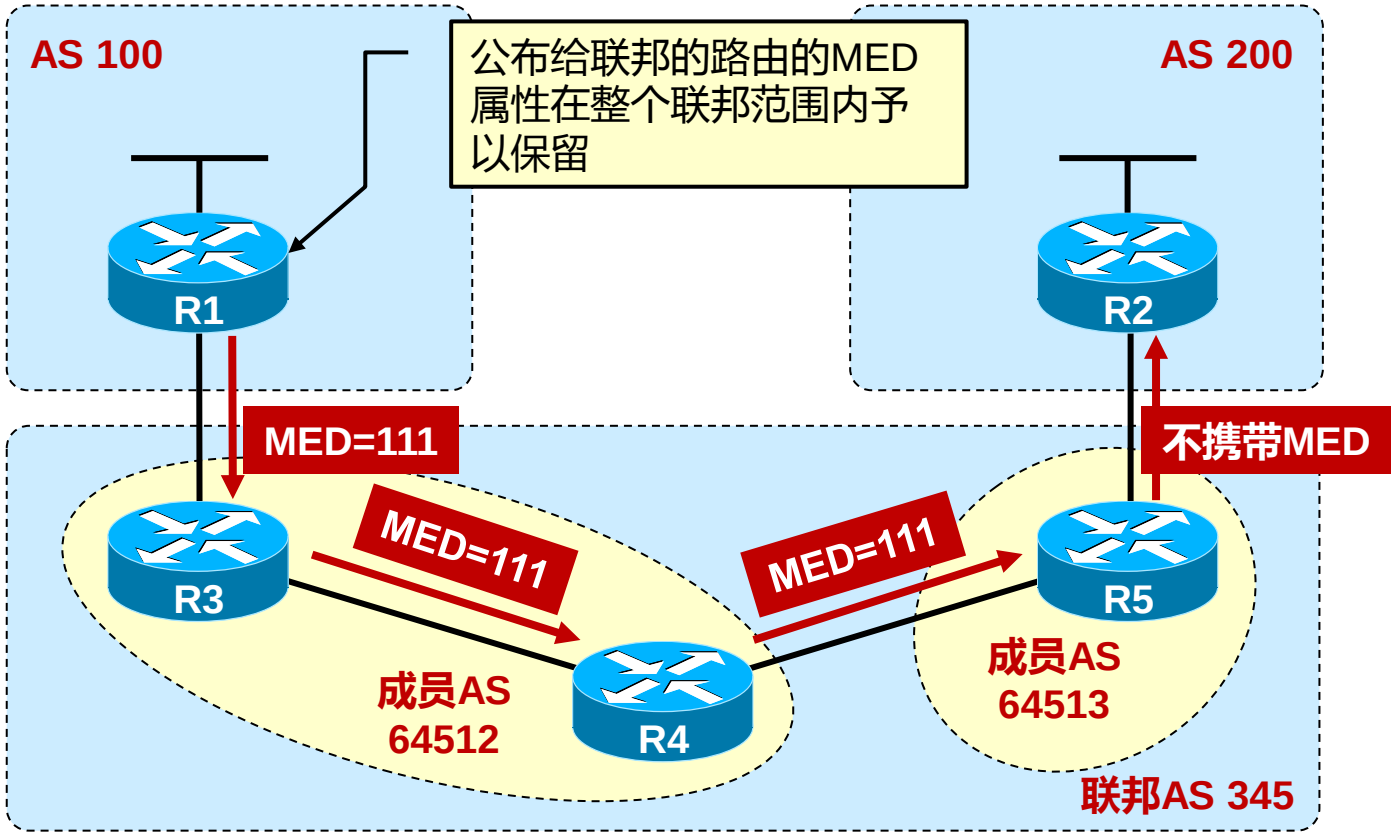
联邦内的BGP路由路径属性

- 在联邦内部保留联邦外部路由的NEXT_HOP属性
- 公布给联邦的路由的MED属性在整个联邦范围内予以保留
- 路由的LP属性在整个联邦范围内予以保留
- 在联邦范围内，将成员AS号压入AS_PATH，但不公布到联邦外，并且使用TYPE3、4的AS_PATH
- AS_PATH中的联邦成员AS号用于在联邦内部避免环路；联邦内成员AS号不参与AS_PATH长度计算

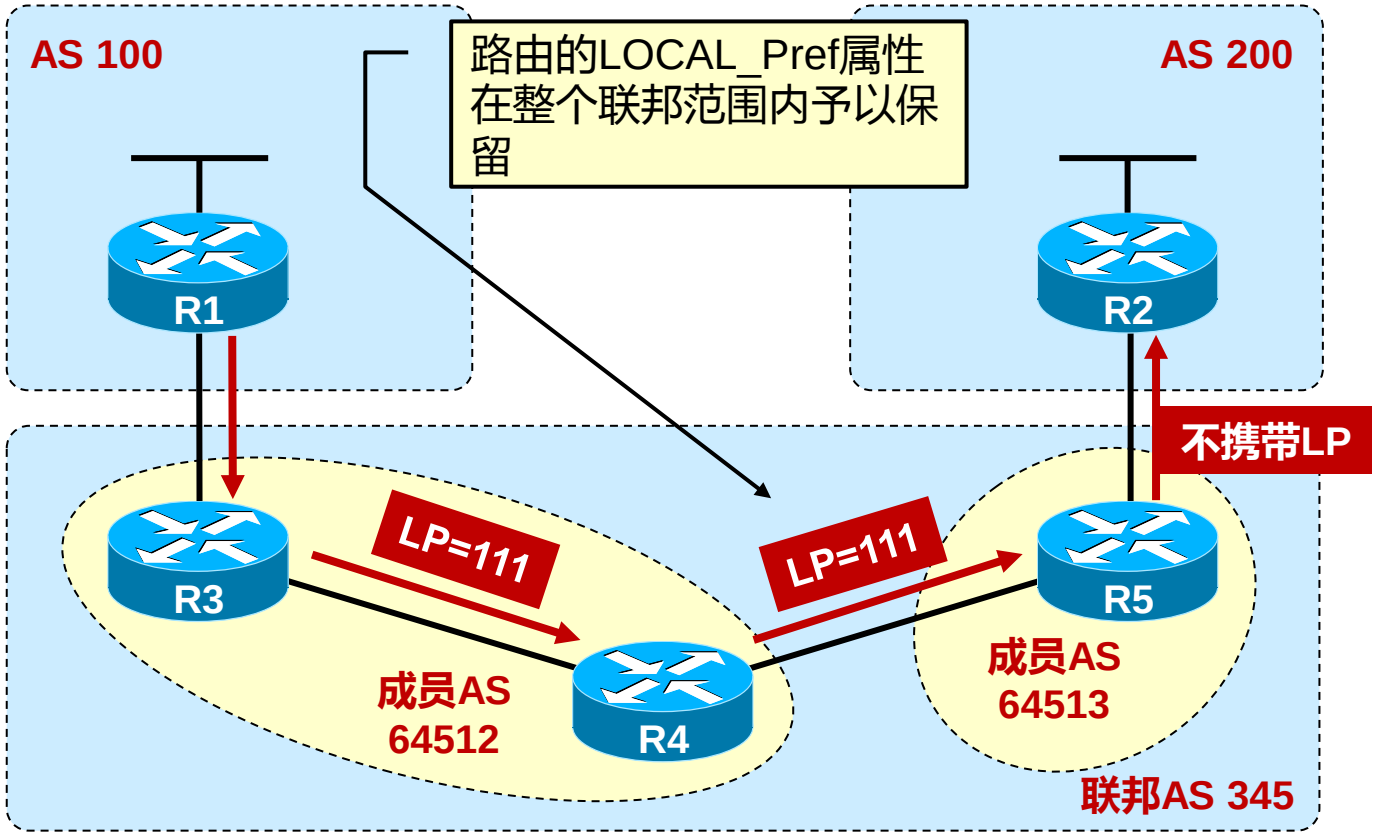
联邦内的BGP路由路径属性 示例



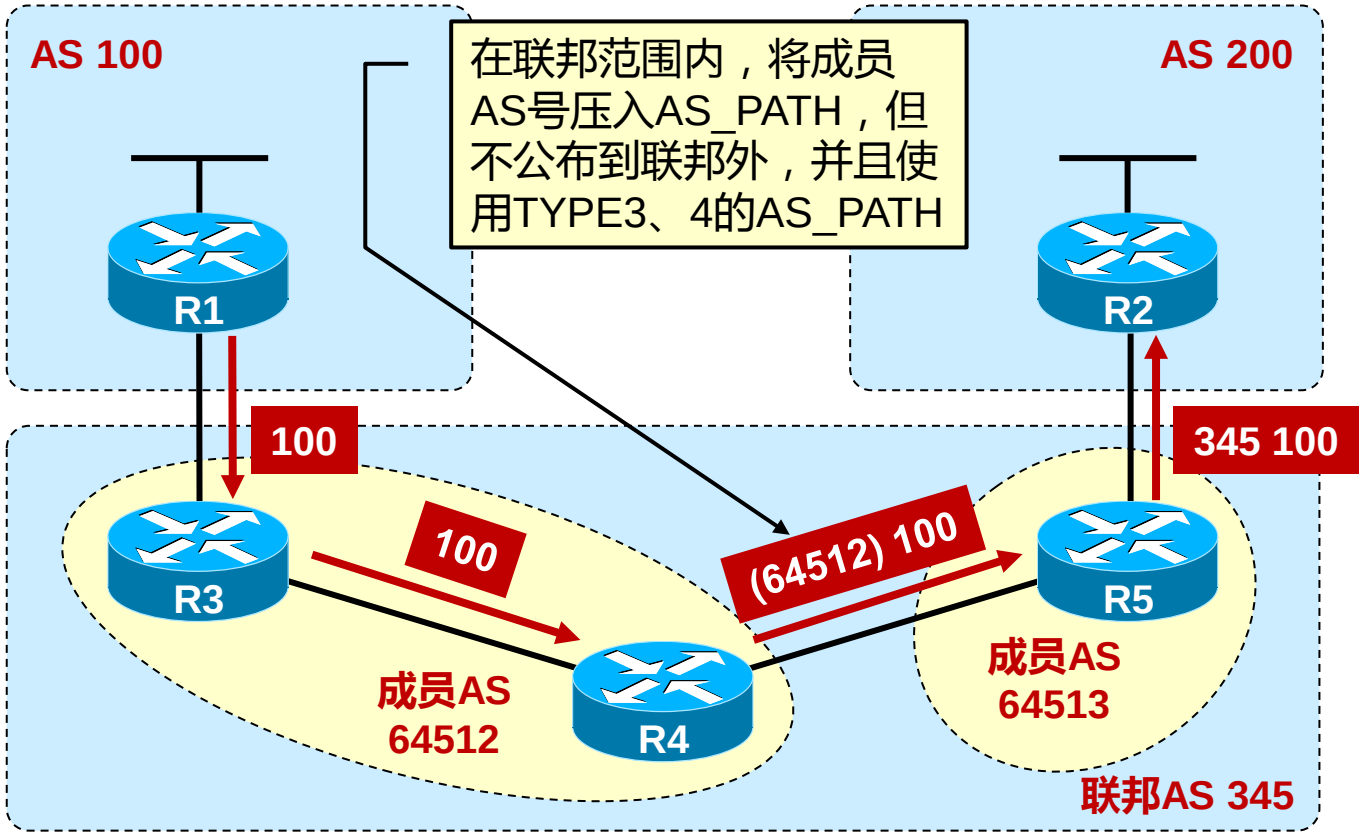
联邦内的BGP路由路径属性 示例(cont.)



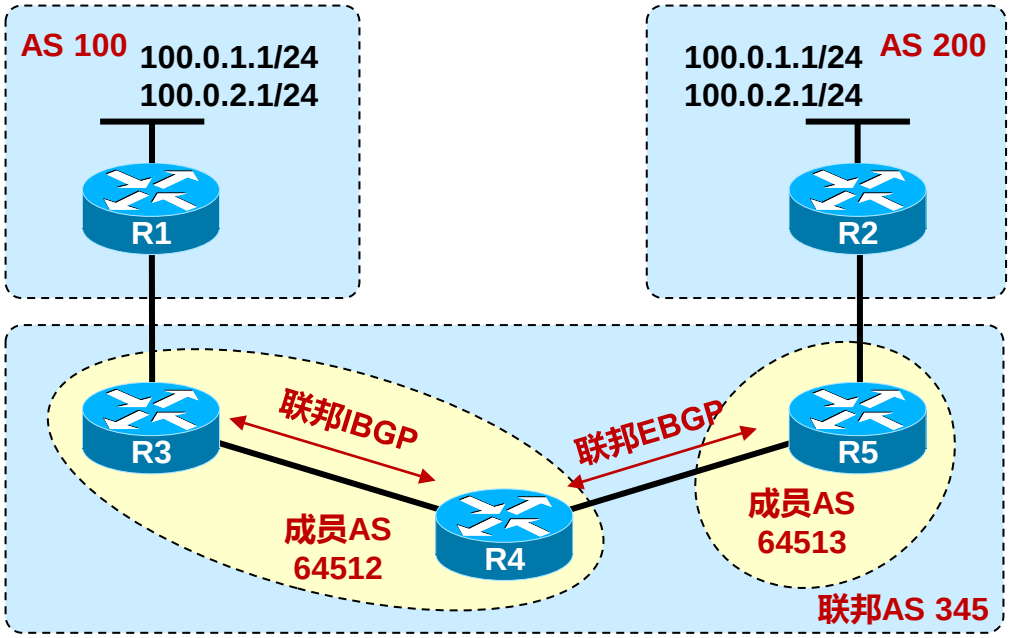
联邦内的BGP路由路径属性 示例(cont.)



联邦内的BGP路由路径属性 示例(cont.)



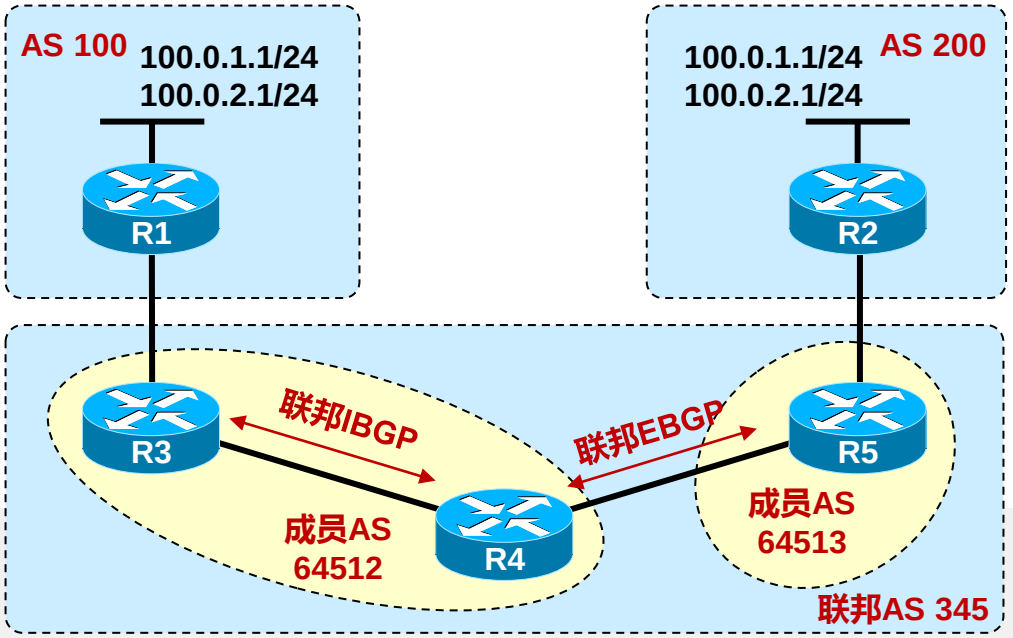
联邦的配置及实现



R3的配置如下

```
router bgp 64512                          // 使用联邦成员AS号建立BGP
 bgp confederation identifier 345         // 标识联邦AS号
 neighbor 4.4.4.4 remote-as 64512
 neighbor 4.4.4.4 update-source Loopback0
 neighbor 10.1.13.1 remote-as 100
```

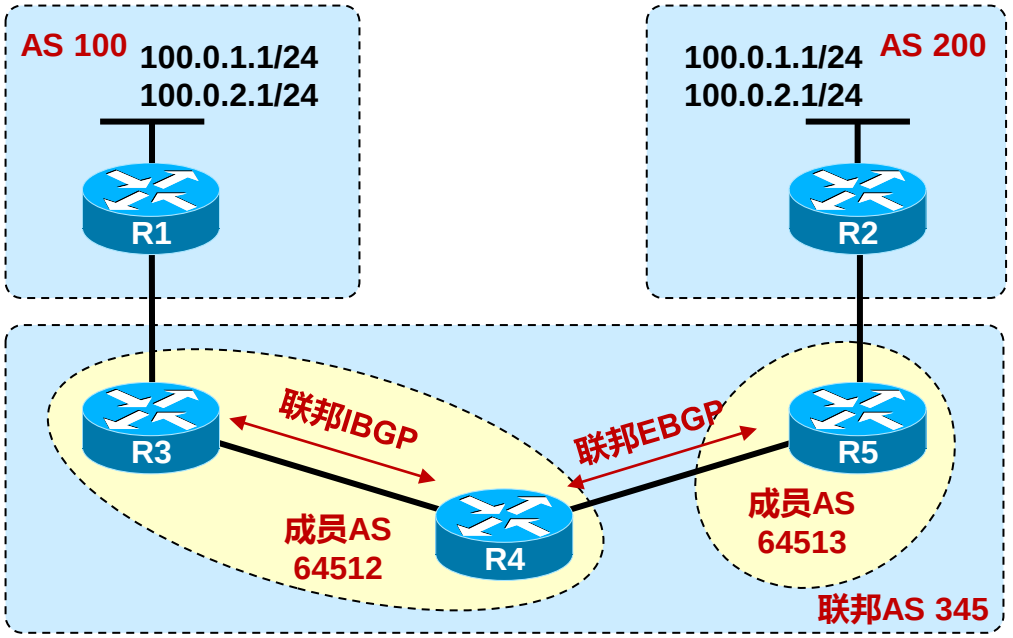
联邦的配置及实现 (cont.)



R4的配置如下

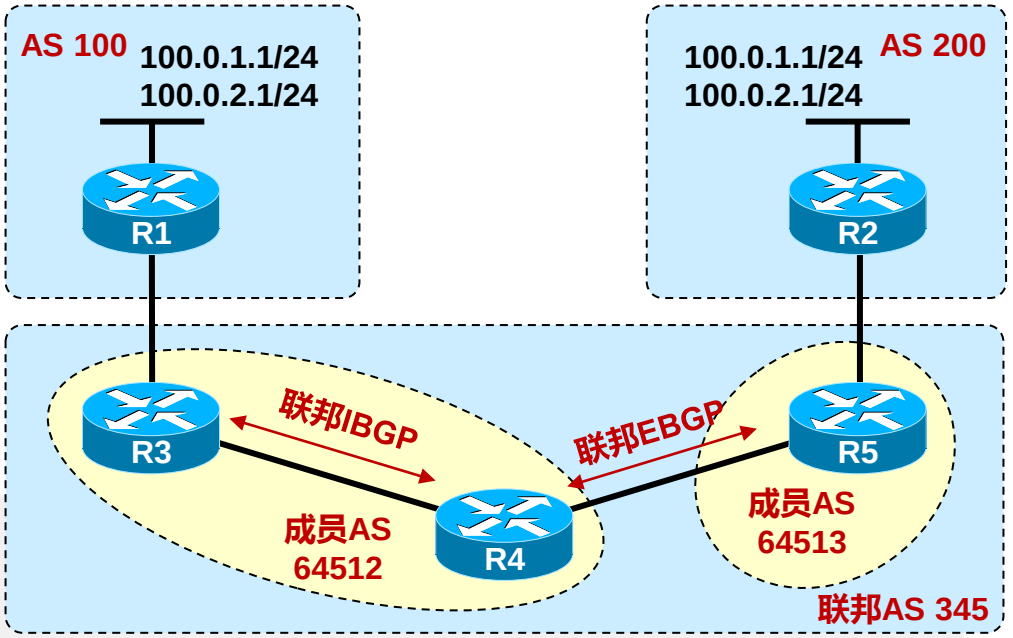
```
router bgp 64512
 bgp confederation identifier 345
 bgp confederation peers 64513          //联邦EBGP邻居的AS号
 neighbor 3.3.3.3 remote-as 64512
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 5.5.5.5 remote-as 64513
 neighbor 5.5.5.5 ebgp-multihop 3
 neighbor 5.5.5.5 update-source Loopback0
```

联邦的配置及实现 (cont.)



- 由于R4与R5是联邦的EBGP，同样有TTL的问题，因此他们使用LOOPBACK建立邻居关系的话，要注意设置ebgp-multihop
- 在R4上增加bgp confederation peers 64513 命令，那么R4将对AS64513视为它的联邦EBGP peer，而对除了AS64513外的AS视为普通的AS

联邦的配置及实现 (cont.)



R5的配置如下

```
router bgp 64513
  bgp confederation identifier 345
  bgp confederation peers 64512
  neighbor 4.4.4.4 remote-as 64512
  neighbor 4.4.4.4 ebgp-multihop 4
  neighbor 4.4.4.4 update-source Loopback0
  neighbor 10.1.25.2 remote-as 200
```

路由反射器与联邦

- 路由反射器(Route Reflector)相比于联邦，优势在于，联邦中所有路由器都需要支持并理解联邦机制，而路由反射器只需要RR理解反射器机制即可，另外，路由反射器的实现机制也相对简单一些。当然如果希望用各种EBGP机制来管理大规模AS，那么联邦将是一个更优的解决方案。

红茶三杯
Vinsoney

学习 沉淀 成长 分享

关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

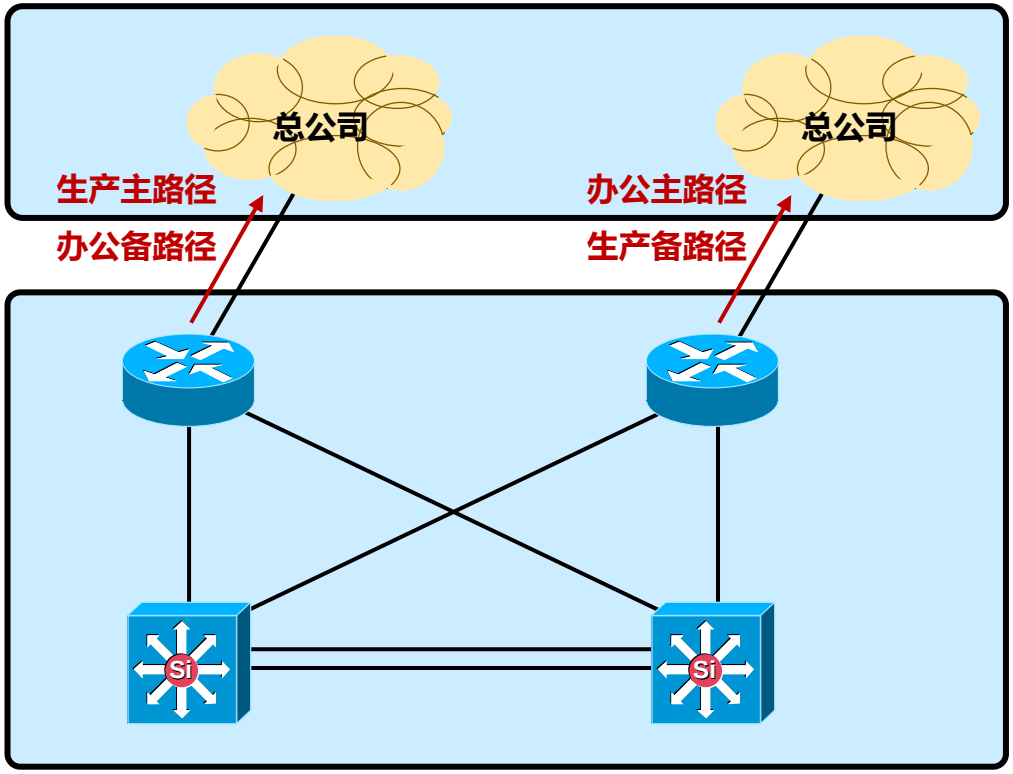
BGP选路规则详解

红茶三杯（朱SIR） http://weibo.com/vinsoney

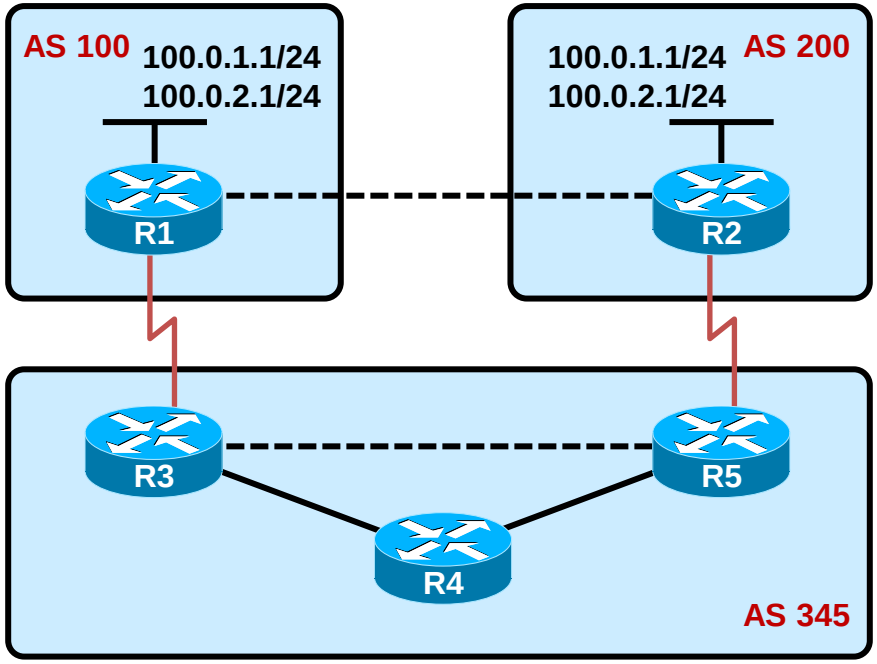
Latest update: 2012-08-01

BGP选路规则

- BGP的选路规则为我们提供了丰富的路由策略部署依据



演示拓扑



BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

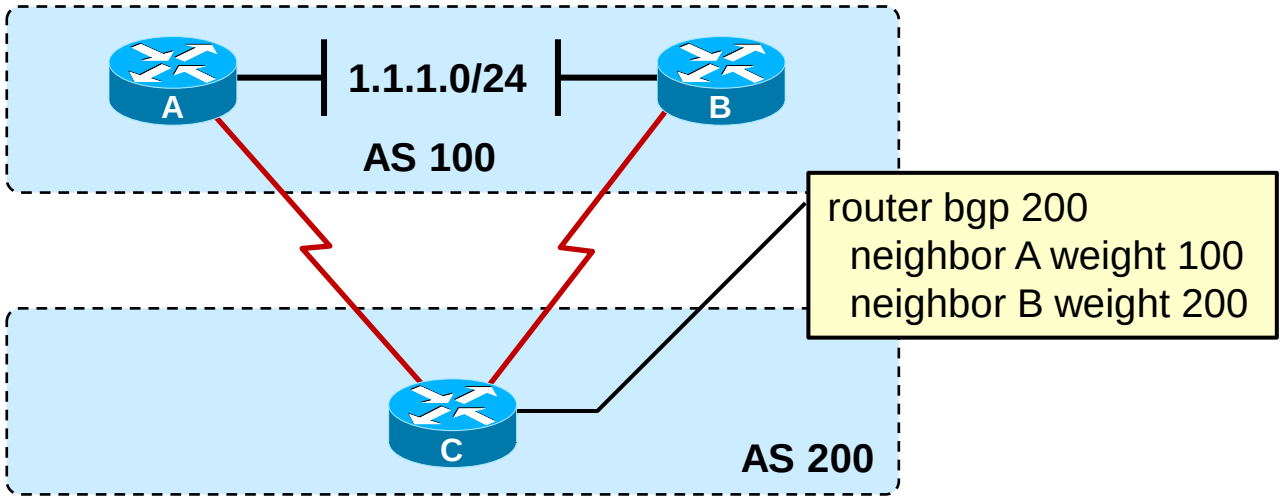
1.优选具有最大Weight值的路由

- **Weight属性回顾**
  - CISCO 私有，越大越优先
  - 作用范围是本路由器（不传递），该值不既不会被包含在update消息中，也不会传递给任何BGP邻居
  - 范围0-65535
  - 如果路由是从其他邻居学过来的，则（在本地该路由的WT）默认值是0
  - 本地network产生的路由weight是32768
  - 本地重发布的直连接口路由、静态路由的weight为32768
  - 本地汇总产生的BGP路由weight值也为32768

1.优选具有最大Weight值的路由

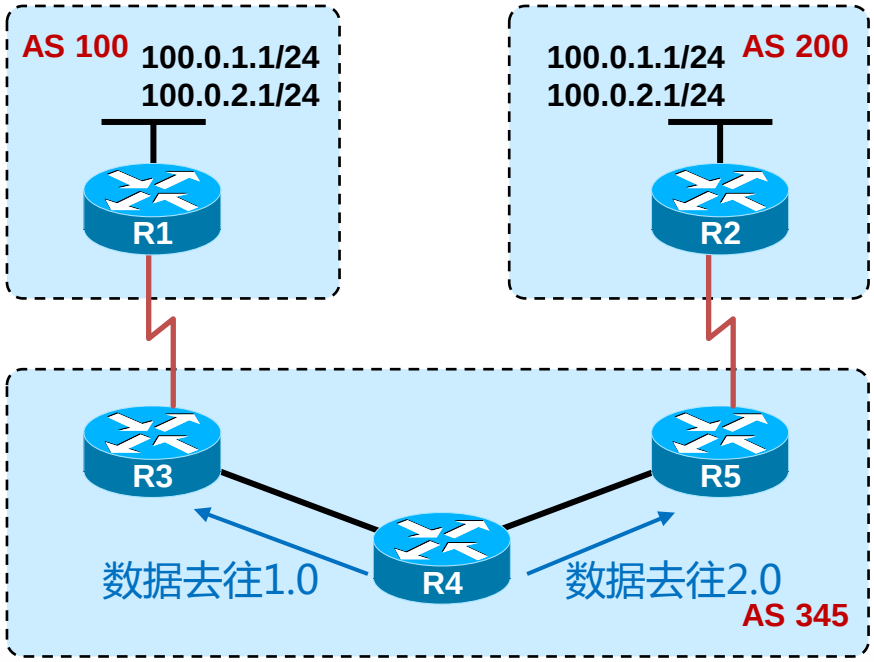
- **修改从特定邻居收到的所有路由的权重**

Router(config-router)#neighbor *{ip-address|peer-group_name}* **weight** *weight*



1.优选具有最大Weight值的路由

- **使用route-map修改权重**



Router-4

```
ip prefix-list 1 permit 100.0.1.0/24
ip prefix-list 2 permit 100.0.2.0/24

route-map WT1 permit 10
  match ip address prefix-list 1
  set weight 20
 route-map WT1 permit 20
match ip address prefix-list 2
  set weight 10

route-map WT2 permit 10
 match ip address prefix-list 1
  set weight 10
route-map WT2 permit 20
match ip address prefix-list 2
  set weight 20

router bgp 345
  neighbor R3 route-map WT1 in
  neighbor R5 route-map WT2 in
```

BGP选路规则

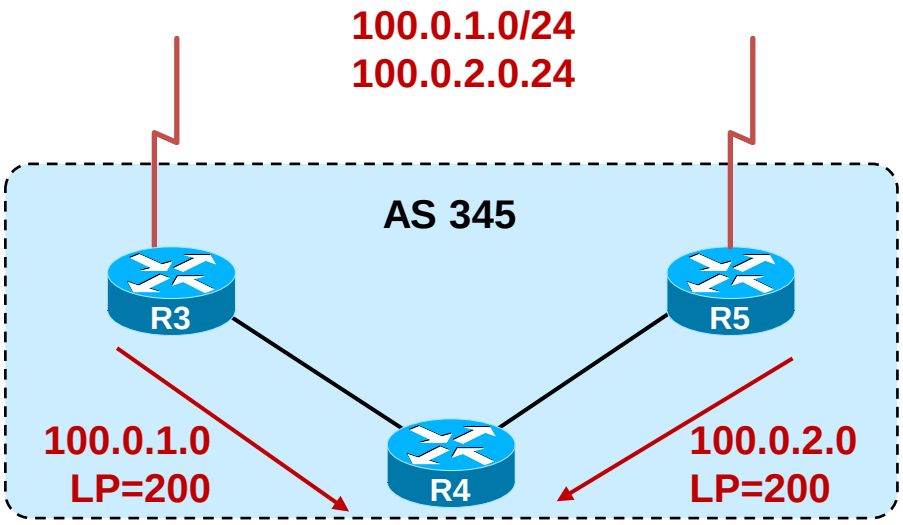
1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

2.优选具有最大Local_Pref值的路由

- **LOCAL_PREFERENCE**
  - 公认自决属性，值越大越优先
  - LOCAL_PREF只能在IBGP Peer之间传递,不能在EBGP Peer之间传递
  - 默认情况下，本地始发的路由的LP为100
    - 可用 bgp default local-preference ？修改默认值
  - BGP路由器在向其EBGP邻居发送路由更新时，不能携带LP属性，但是对方会在本地为这条路由赋一个默认值，也就是100，然后再传递给自己的IBGP邻居
  - 本地network及重发布的路由，LP默认100，并能在AS内向其他IBGP邻居传输，传输过程中除非部署策略，否则LP不变

2.优选具有最大Local_Pref值的路由

- **通过route-map修改本地优先级**



Router-4

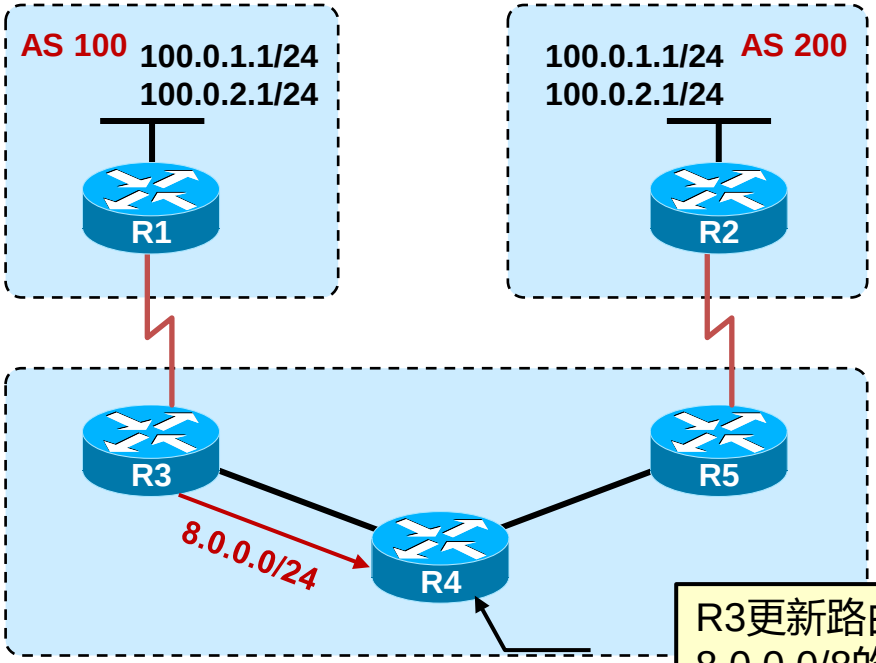
```
ip prefix-list 1 permit 100.0.1.0/24
ip prefix-list 2 permit 100.0.2.0/24

route-map LP1 permit 10
  match ip address prefix-list 1
  set local-preference 200
route-map LP1 permit 20
!
route-map LP2 permit 10
  match ip address prefix-list 2
  set local-preference 200
route-map LP2 permit 20
!
router bgp 345
  neighbor 3.3.3.3 route-map LP1 in
  neighbor 5.5.5.5 route-map LP2 in
```

- 当然，也能在R3、R5上对R4执行OUT方向的策略实现相同的效果

BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

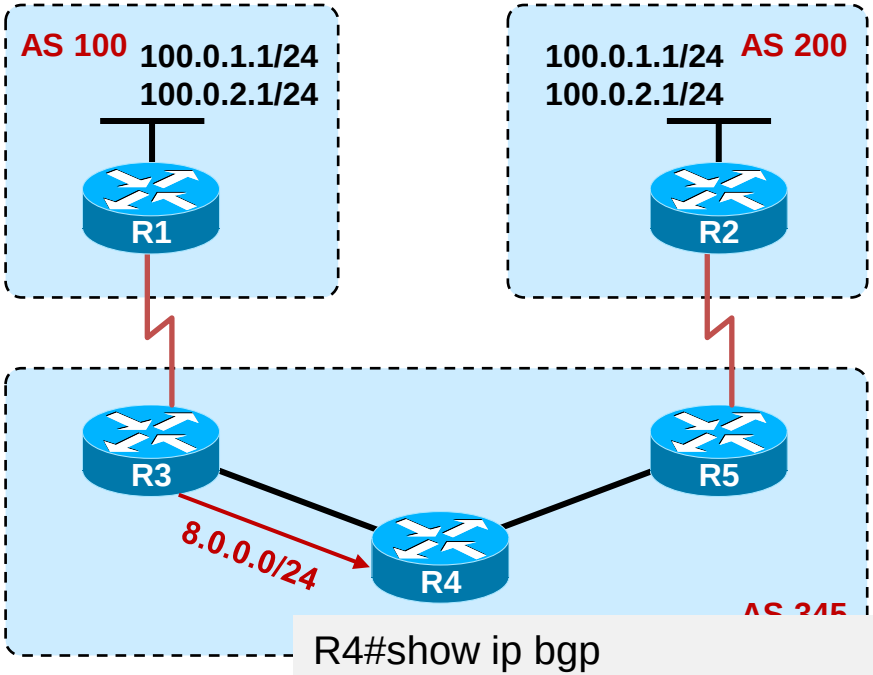
3.优选起源于本地的路由




```
route-map test permit 10
  set weight 0
!
router bgp 345
  network 8.0.0.0 mask 255.0.0.0 route-map test
```

R3更新路由8.0.0.0/8给R4，R4本地又network了一条8.0.0.0/8的路由，为跳过weight值对选路规则的影响，用route-map将R4宣告的这条路由在本地的weight值修改为0。
由于在R4上，来自R3及本地network的这条路由weight都为0，且LP都为100，最终根底第3条选路规则优选本地network的路由。

3.优选起源于本地的路由



R4#show ip bgp

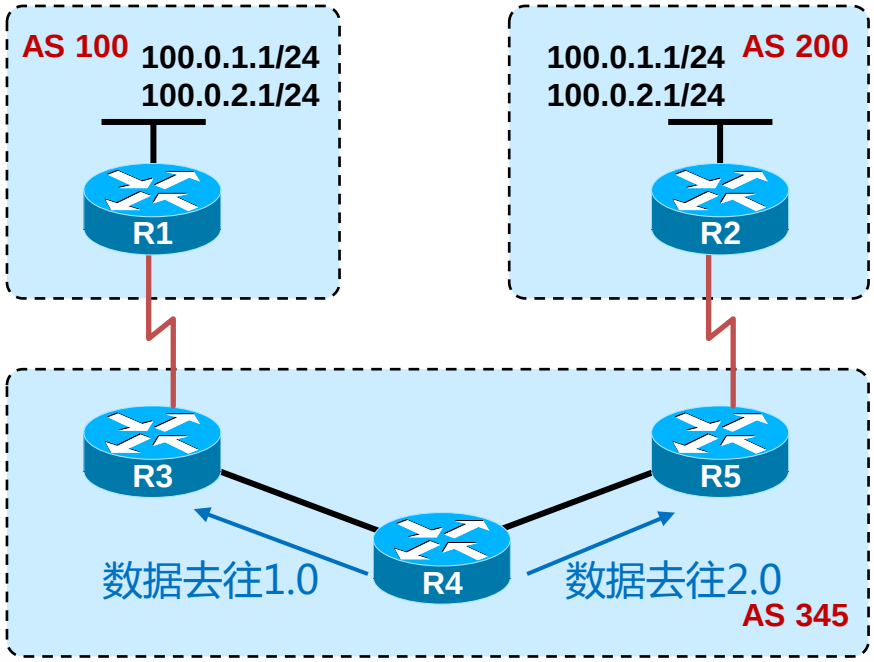
Network	Next Hop	Metric	LocPrf	Weight	Path
*> 8.0.0.0/8	0.0.0.0	0		0	i
* i	3.3.3.3	0	100	0	i

BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

4. 优选AS_PATH最短的路由

- **通过Route-map修改AS-PATH**



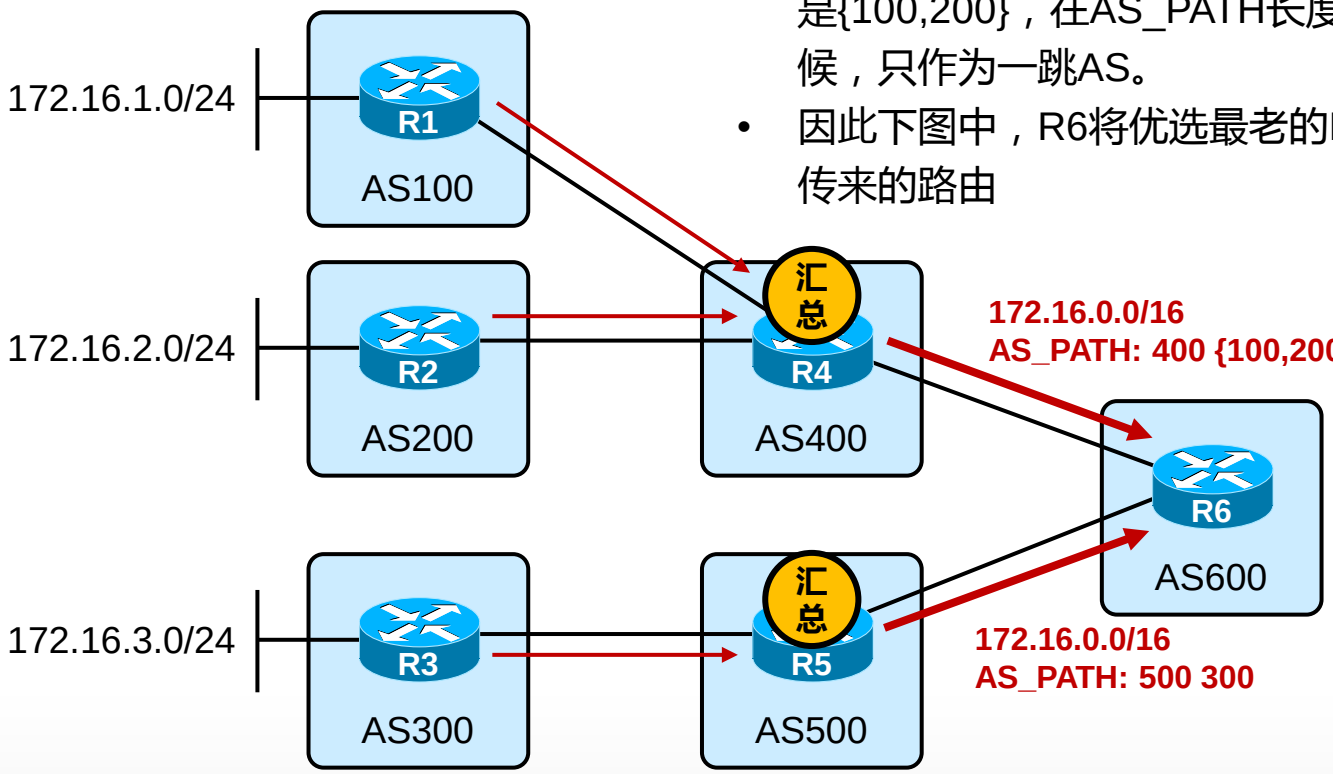
R1的配置如下：

```
ip prefix-list 2 permit 100.0.2.0/24

route-map ASPATH permit 10
  match ip address prefix-list 2
  set as-path prepend 100 100

router bgp 100
  neighbor 10.1.13.3 route-map ASPATH out
```

- as-path的策略，只能在AS之间执行，因为as-path只会在离开AS的时候发生改变。
- 注意，在R1上执行as-path插入，与在R3上用in方向执行相同的策略有何不同？

4. 优选AS_PATH最短的路由

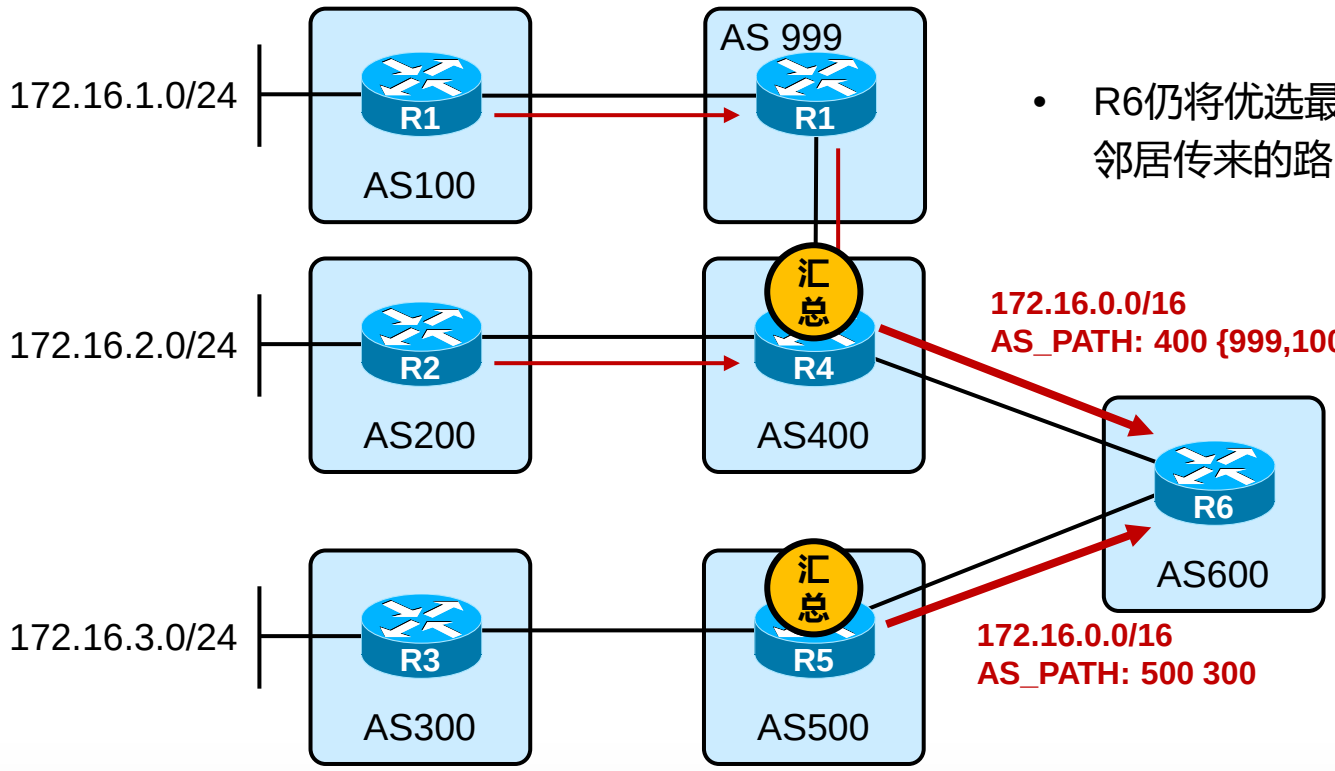
- **规则补充：**在做聚合路由时，使用as-set关键字后产生的AS-Path列表中的{}里的AS号长度只算一个AS号的长度



- AS_PATH中，as-set类型的AS号，也就是{100,200}，在AS_PATH长度计算的时候，只作为一跳AS。
- 因此下图中，R6将优选最老的EBGP邻居传来的路由

4. 优选AS_PATH最短的路由

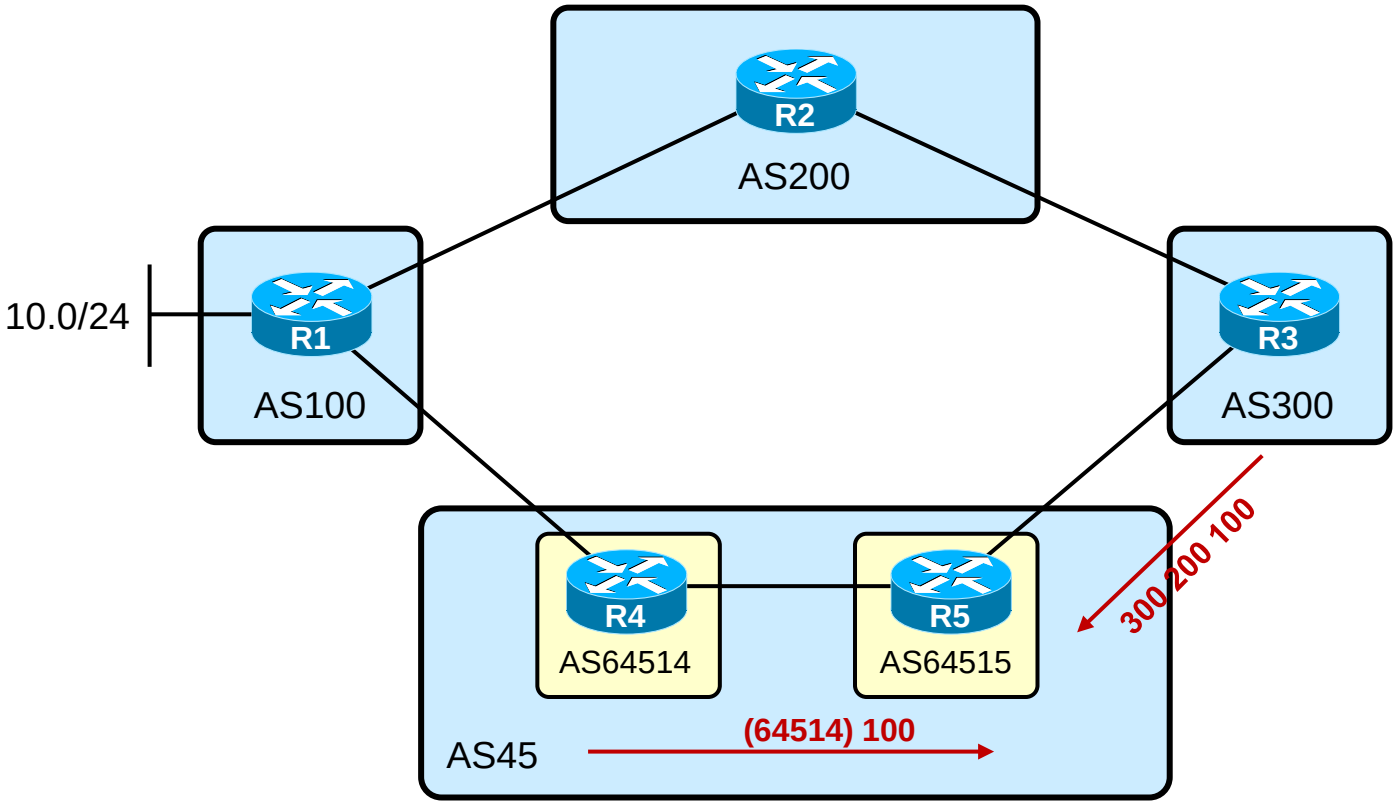
- **规则补充：**在做聚合路由时，使用as-set后产生的AS-Path列表中的{}里的AS号长度只算一个AS号的长度



- R6仍将优选最老的EBGP邻居传来的路由

4. 优选AS_PATH最短的路由

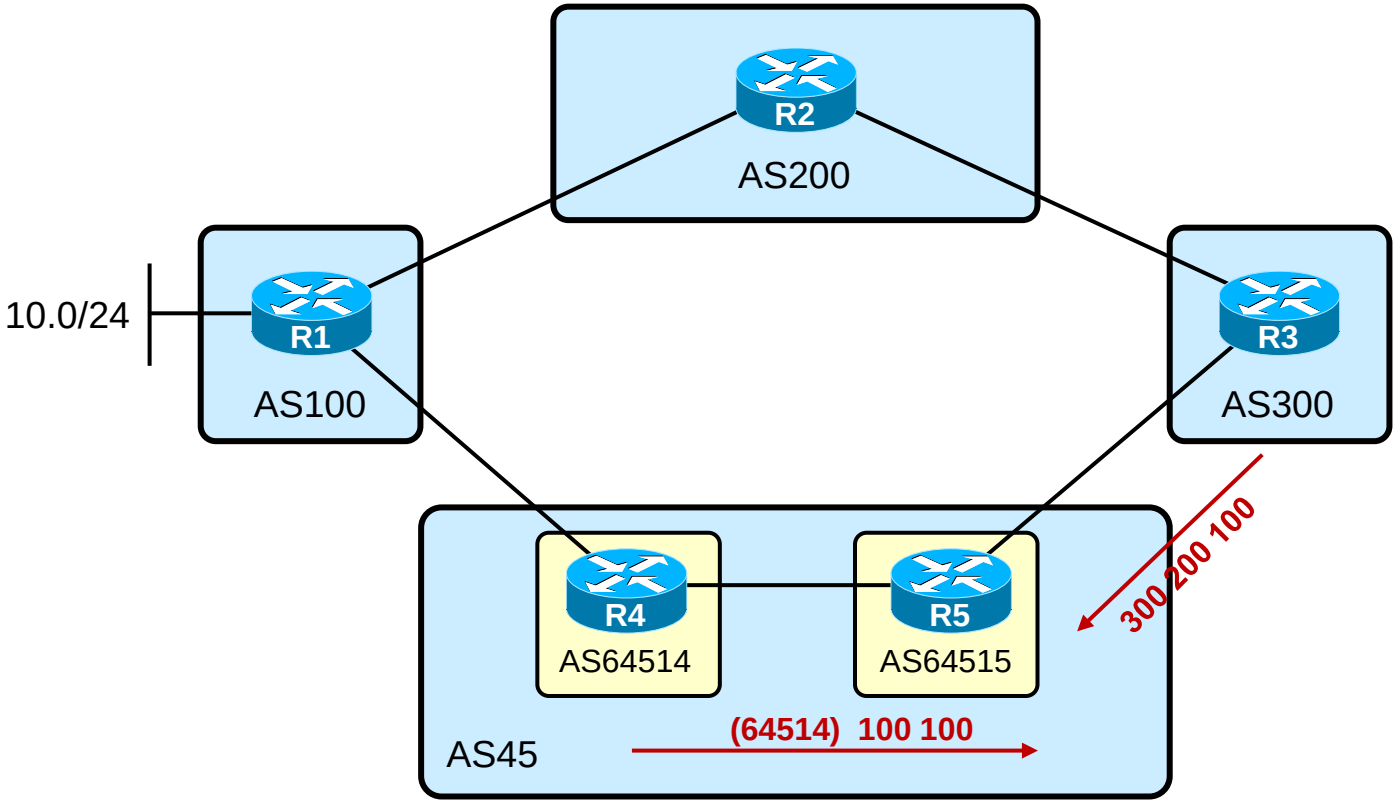
- **规则补充：**在联邦内的AS-Path列表中的()内的AS号长度不做计算依据



- 完成基本配置后R5优选R4作为到达10.0网络的下一跳，因为R4传递给R5的BGP路由，next-hop为R1，R5不可达因此不best。
- 在R4对R5使用next-hop-self后，R5优选R4，因为AS_PATH短

4. 优选AS_PATH最短的路由

- **规则补充：**在联邦内的AS-Path列表中的()内的AS号长度不做计算依据



- 在R1上对R4使用route-map，修改更新的BGP路由10.0的AS_PATH，插入**100**的AS号
- 则R5从R3及R4收到的BGP路由10.0的AS_PATH如上图，R5优选R4，因为AS_PATH更短，注意联邦里的成员AS号不参与AS_PATH计算
- 在R1上对R4使用route-map，修改更新的BGP路由10.0的AS_PATH，插入**100 100**的AS号
- 则R5优选R3

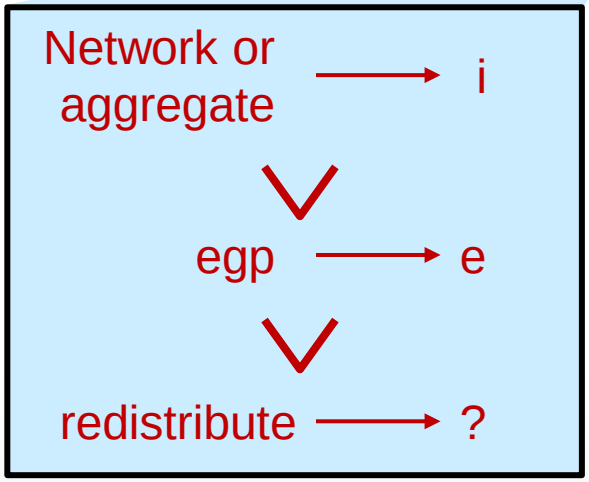
BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

5. ORIGIN

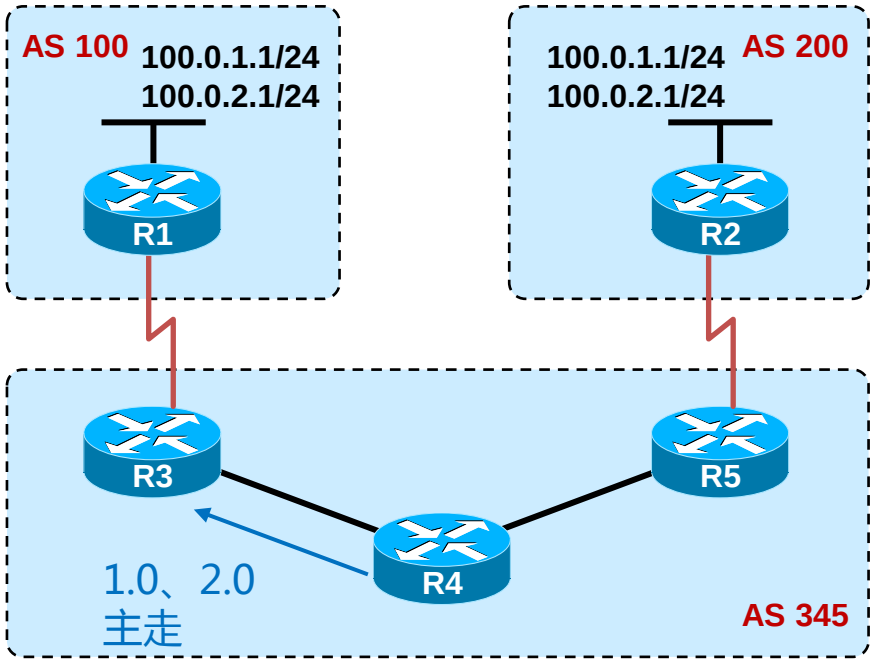
- **Origin**

```
Router# show ip bgp
<!– some output omitted->

Network          Next Hop        Metric  LocPrf  Weight  Path
*> 11.11.11.0/24 10.1.12.1       0       0               64512 i
* i              10.1.23.3       0       0               300 64512 i
```



5. ORIGIN

- **引入路由方式**



R1的配置如下：

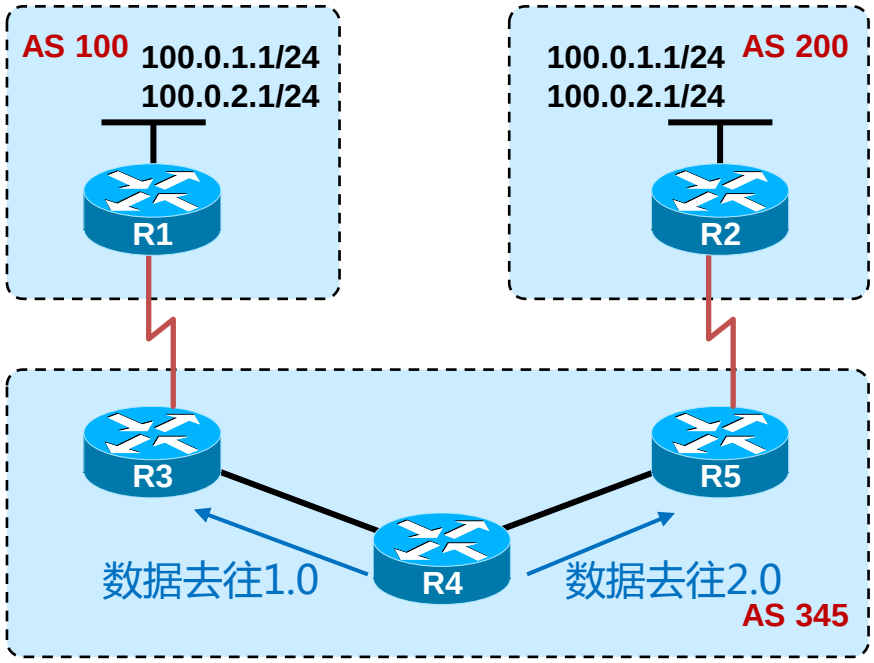
```
router bgp 100
  network 100.0.1.0 mask 255.255.255.0
  network 100.0.2.0 mask 255.255.255.0
```

R2的配置如下：

```
ip prefix-list 1 permit 100.0.1.0/24
ip prefix-list 2 permit 100.0.1.0/24
route-map Conn permit 10
  match ip address prefix-list 1 2

router bgp 200
  redistribute connected route-map Conn
```

5. ORIGIN

- **使用route-map改变origin属性**



R4的配置如下：

```
ip prefix-list 1 permit 100.0.1.0/24
ip prefix-list 2 permit 100.0.2.0/24

route-map setOri1 permit 10
  match ip address prefix-list 1
  set origin

route-map setOri2 permit 10
  match ip address prefix-list 2
  set origin


router bgp 345
  neighbor 3.3.3.3 route-map setOri1 in
  neighbor 5.5.5.5 route-map setOri2 in
```

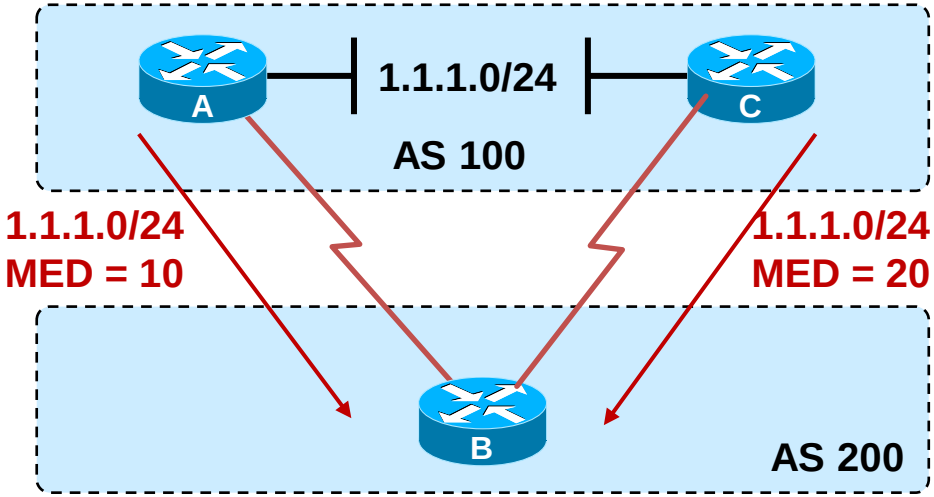
- 不推荐用此规则来影响BGP决策或数据走向

BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

6. 优选MED最小的路由

- **MED属性**
  - 可选非传递属性，值越小越优先，一般用于AS之间影响BGP路由决策

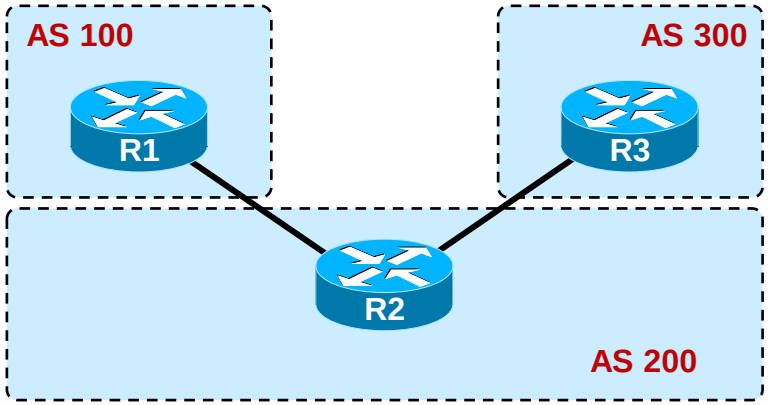


6. 优选MED最小的路由

- **MED属性设置方法**
  - 将IGP路由引入BGP时关联Route-map进行设置
  - 对BGP Peer应用IN/OUT方向的Route-map进行设置
  - 非Route-map(自动)方式:
    - 使用network或redistribute方式将IGP路由引入BGP时,MED将继承IGP路由的Metric(直联路由及静态路由的Metric为0)
    - 使用aggregate-address方式引入路由，则MED为空

6. 优选MED最小的路由

- **MED注意事项**
  - 默认情况下，只比较来自同一邻居AS的BGP路由的MED值，就是说如果同一个目的地的两条路由来自不同的AS，则不进行MED值的比较。如果仍然希望比较来自不同邻接AS的路由，可使用如下命令：
    - bgp always-compare-med
  - MED只是在直接相连的自治系统间影响业务量，而不会跨AS传递



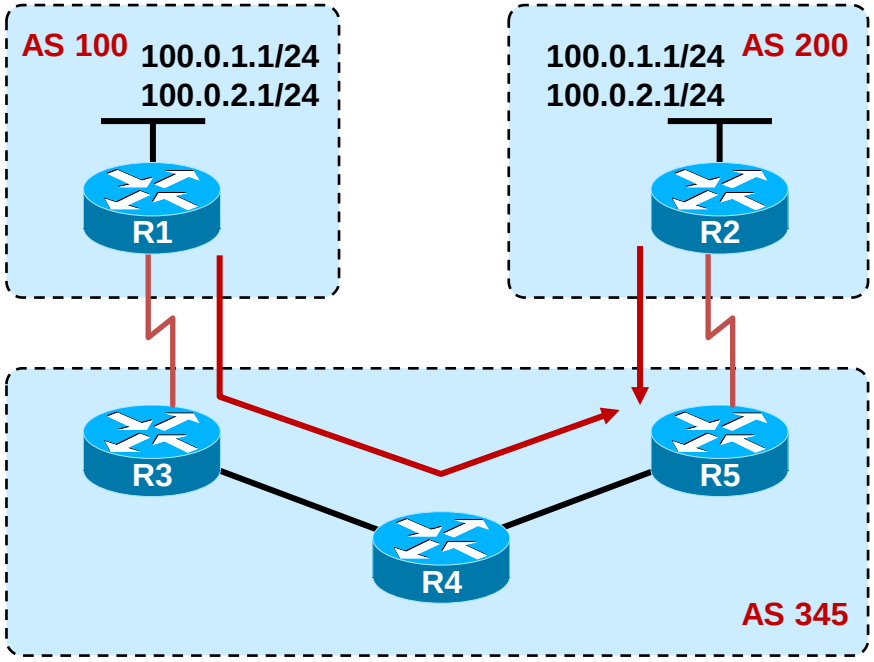
BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED

7

7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

7. 优选EBGP邻居的路由（相对于IBGP）



- 100.0.1.0 及 2.0 的路由，R5 同时从EBGP邻居R2及IBGP邻居R3收到，将优选EBGP邻居作为路由的下一跳。
- 注意IBGP邻居关系（R3-R5之间需要有iBGP邻居关系）

BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED

8

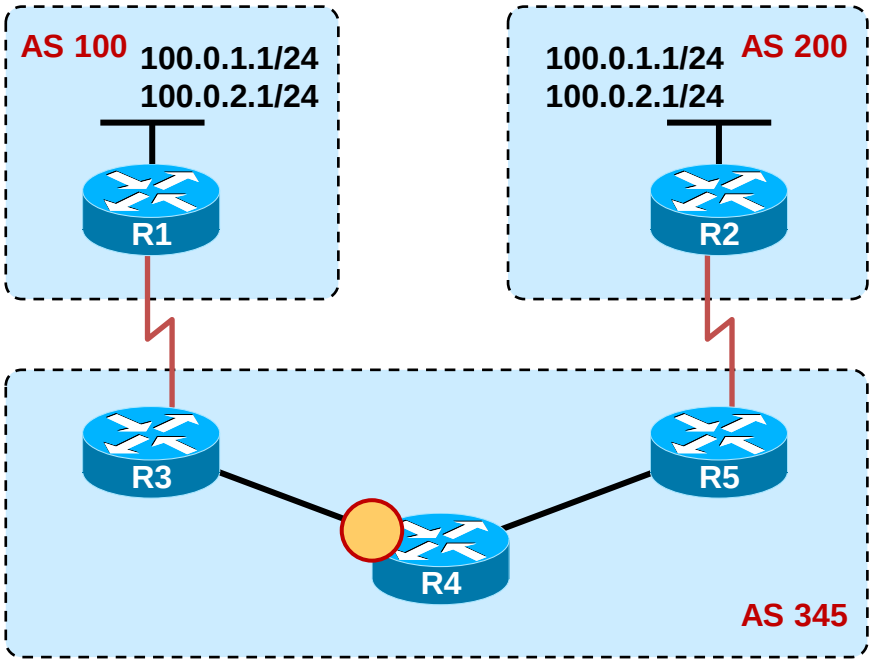
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

8. 优选到NEXT_HOP最近的路由

- 注意，这里严格的说应该这么表述：我从两个BGP邻居各收到一条路由，这两条BGP路由有相同的路由前缀，首先这两条BGP路由的NEXT_HOP是不相同的，否则不具有可比性，那么我比较本地到达这两个NEXT_HOP的IGP度量值，谁metric小，我就优选谁。

8. 优选到NEXT_HOP最近的路由

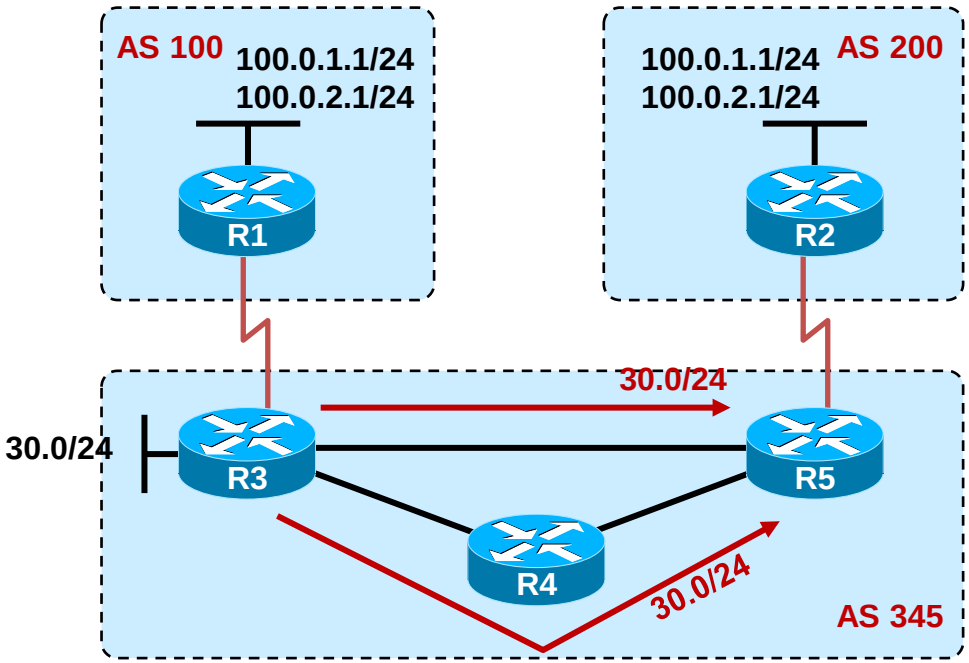
- **场景示例1**



- R4同时从R3、R5学习到100网段的路由，NEXT_HOP分别为3.3.3.3及5.5.5.5，此时，将R4连接R3的接口COST调大，那么R4到达NH 3.3.3.3的metric就变大了，则优选R5这条BGP路径，因为到达NH 5.5.5.5的metric较小

8. 优选到NEXT_HOP最近的路由

- **场景示例2（思考）**

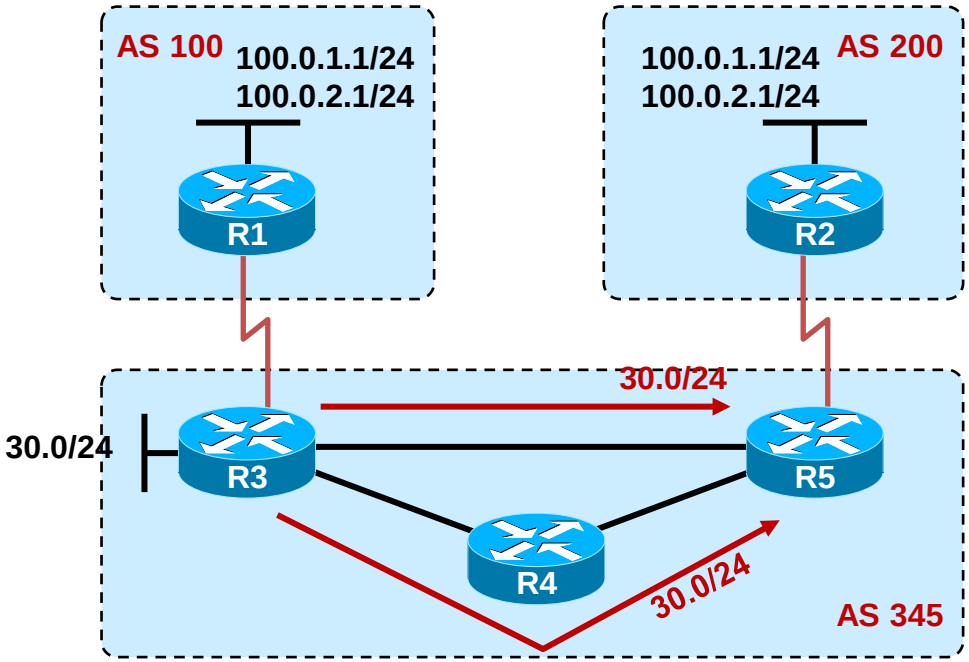


- R3-R4；R4-R5；R3-R5 维护IBGP邻居关系并且都是用各自的LOOPBACK口作为更新源及建立BGP邻居关系，地址分别为3.3.3.3、4.4.4.4、5.5.5.5。
- AS内运行OSPF协议，使得所有路由器都能获取到其他路由器的LOOPBACK网段。在R3上发布30.0网段的BGP路由，同时配置R4为RR，R3为RR client

- R5会同时从R3及R4收到30网段的路由更新，它将如何选路？那一条选路规则生效？

8. 优选到NEXT_HOP最近的路由

- **场景示例2（解答）**

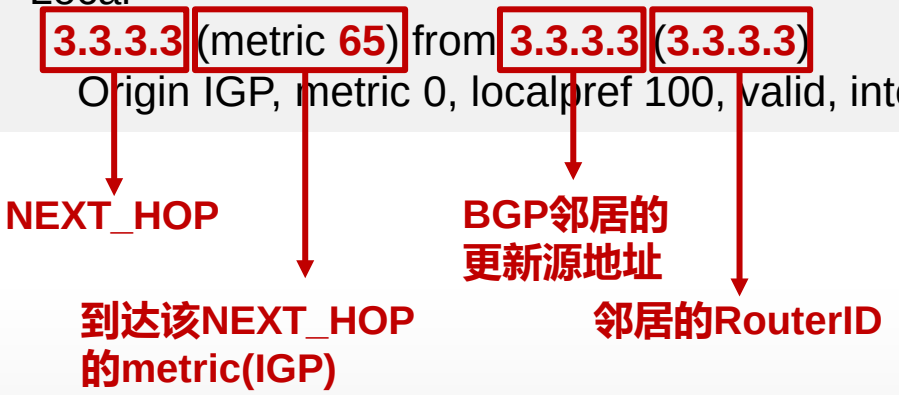


- **首先规则8不适用**，因为两条BGP路由NEXT_HOP相等，都是3.3.3.3，不具有可比性（到达3.3.3.3的metric就一个值）
- 其次规则9、10略过
- 再次规则11，似乎生效了，但是此时将R3的BGP routerID改的比R4大，发现R5仍然优选R4，这是因为“如果一条路径包含RR属性，产生者ID将在最优路径选择过程中代替RID”
- 因此到规则12：如果多条路径始发路由器ID或路由器ID相同，那么优选Cluster-List 最短的路径

8. 优选到NEXT_HOP最近的路由

- **场景示例2**

```
R5#sh ip b 30.30.30.0
BGP routing table entry for 30.30.30.0/24, version 2
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Flag: 0x820
  Not advertised to any peer
  Local
    3.3.3.3 (metric 65) from 4.4.4.4 (4.4.4.4)
      Origin IGP, metric 0, localpref 100, valid, internal
      Originator: 3.3.3.3, Cluster list: 4.4.4.4
  Local
    3.3.3.3 (metric 65) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```



BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED

9

7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由 BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

9. BGP负载均衡

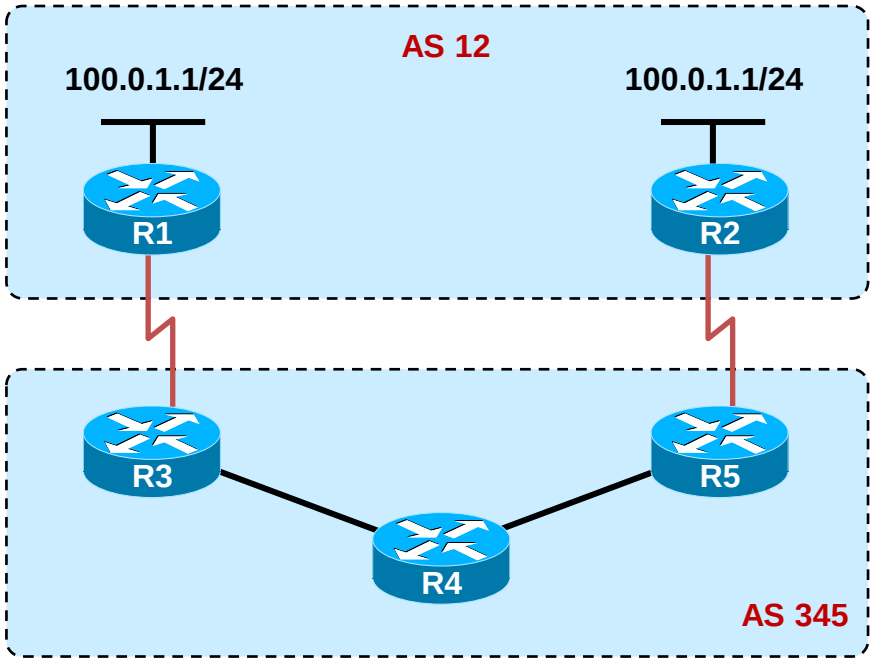
- **当前面的8条选路原则都无法优选出最优路由时，并且在BGP进程下面配置了maximum-paths [ibgp] n,n的取值为2-6,那么将执行等价负载均衡，也就是将这些等代价的BGP路径都放进IP路由表使用，但是要注意，虽然这些路径在本地都用了，最终却只有一条BGP路径是best最优的。**
- **具备等价负载均衡条件的候选路径需满足如下条件：**
  - 必须有相同的路径属性，如weight、LP、AS_PATH（不仅是长度，整个AS_PATH包括AS号都要相同）、origin code、MED及IGP的Distance值
  - 每一条路径的下一跳都不相同

9. BGP负载均衡

- **maximum-paths [ibgp] n**
  - 如果不关联ibgp关键字，那么只会对external路由执行等价负载均衡（默认仅对EBGP路由）
  - 如果要对internal路由做负载均衡，则需关联ibgp关键字
  - 如果不配置maximum-paths，那么将进行到下一条选路原则

9. BGP负载均衡

- **IBGP等价负载均衡**



- R4同时从IBGP邻居R3、R5收到100网段的路由，在不执行任何策略的情况下，这些路由通过BGP决策的规则1-8都无法抉择，并且所有的路径属性都相等，具备实施等价负载均衡的条件,命令如下 ：

- maximum-paths ibgp 2

9. BGP负载均衡

- **IBGP等价负载均衡**

show ip bgp

Network	Next Hop	Metric	LocPrf	Weight	Path
* i100.0.1.0/24	5.5.5.5	0	100	0	100 i
*>i	3.3.3.3	0	100	0	100 i

- 注意虽然配置了maximum-paths，路由表中关于100网段出现了负载均衡，但R4在BGP优选动作仍然只会优选一条BGP路由，并只将这条路由更新给BGP邻居

show ip bgp 100.0.1.0

```
BGP routing table entry for 100.0.1.0/24, version 9
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Multipath: iBGP
  Not advertised to any peer
  100
    5.5.5.5 (metric 65) from 5.5.5.5 (5.5.5.5)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
  100
    3.3.3.3 (metric 65) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath, best
```

9. BGP负载均衡

- **IBGP等价负载均衡**

Show ip route

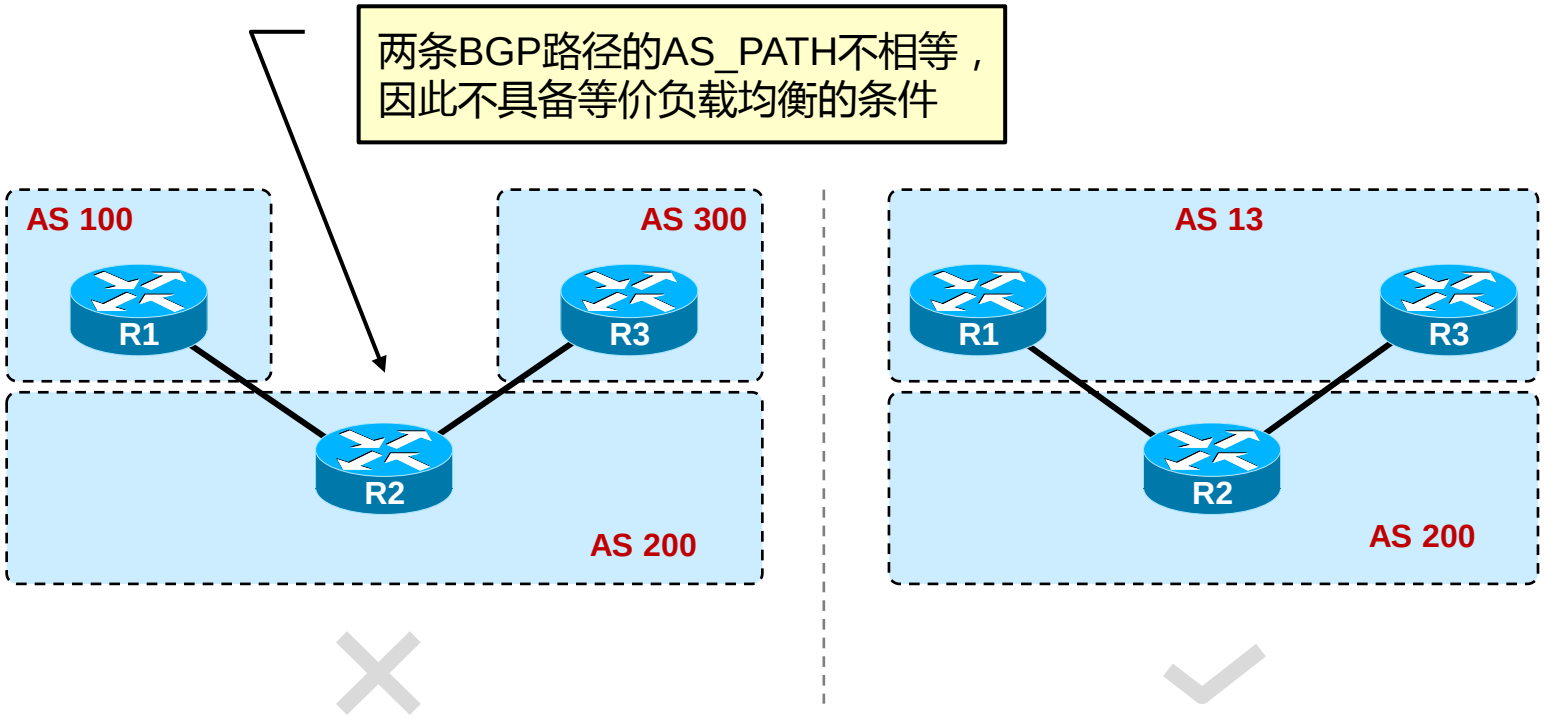
```
B   100.0.1.0 [200/0] via 5.5.5.5, 00:04:48
                  [200/0] via 3.3.3.3, 00:04:48
```

show ip route 100.0.1.0

```
Routing entry for 100.0.1.0/24
  Known via "bgp 345", distance 200, metric 0
  Tag 100, type internal
  Last update from 3.3.3.3 00:05:41 ago
  Routing Descriptor Blocks:
    5.5.5.5, from 5.5.5.5, 00:05:41 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 100
  * 3.3.3.3, from 3.3.3.3, 00:05:41 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 100
```

9. BGP负载均衡

- **EBGP等价负载均衡**

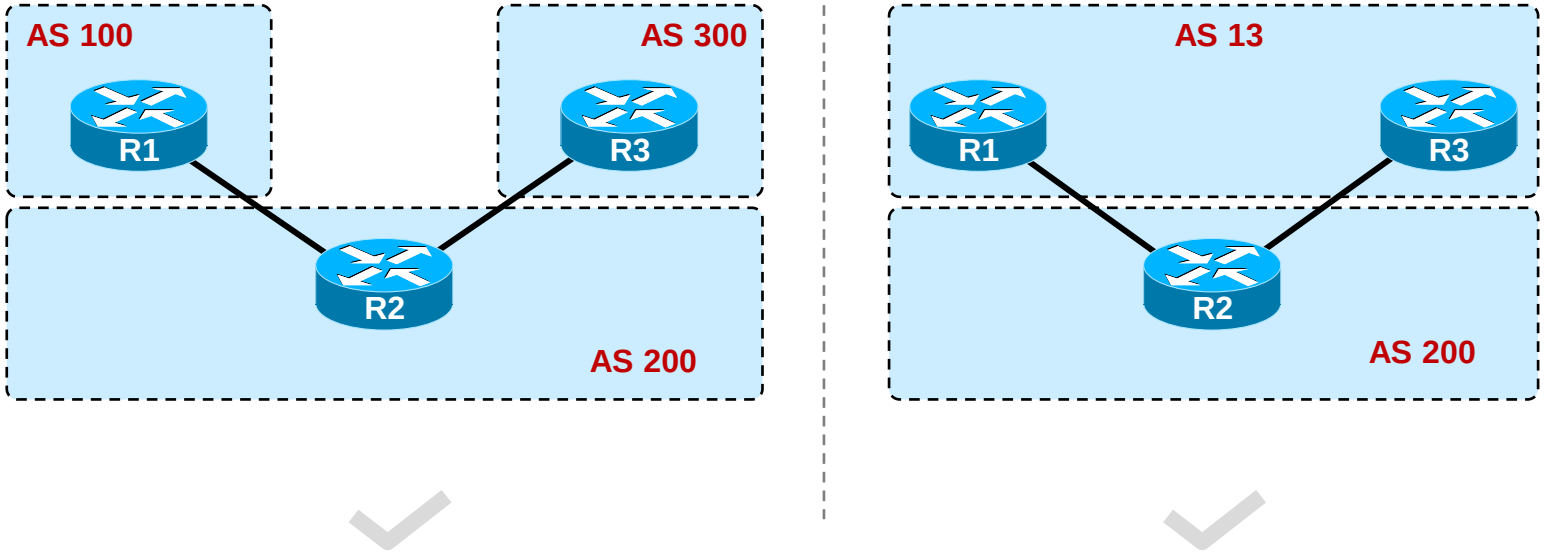


BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

10. 优选最老的EBGP邻居的路由

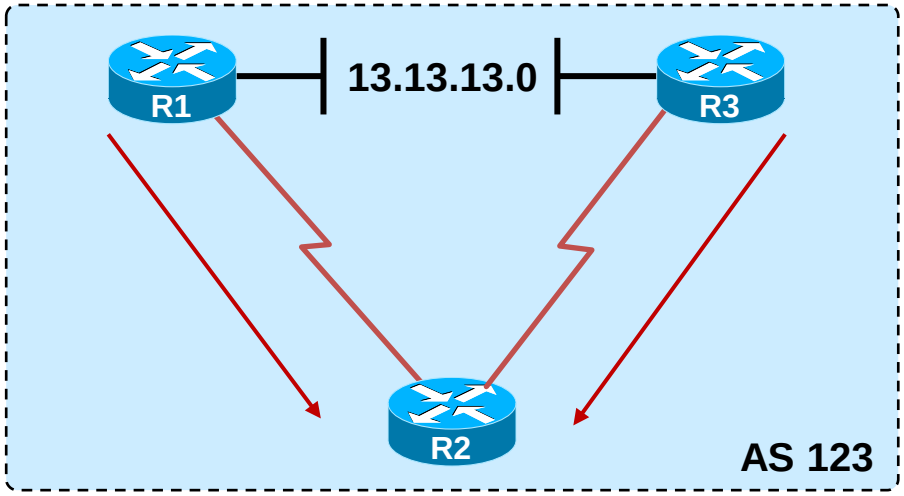
- **规则描述**
  - 如果路由都来自EBGP邻居，则优选最老的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太强）



CISCO IOS Version 12.4(25)

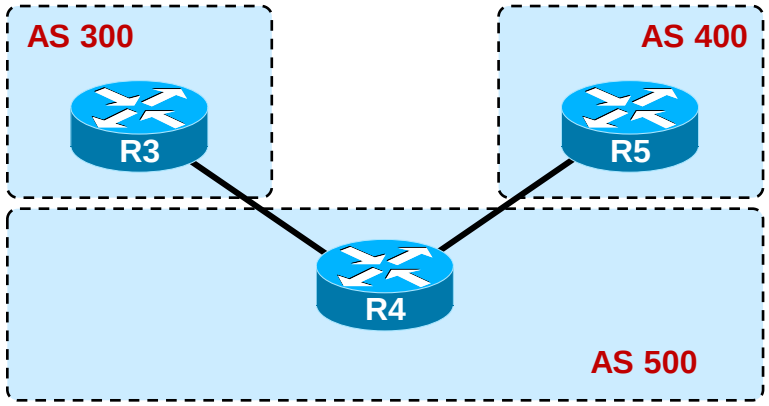
10. 优选最老的EBGP邻居的路由

- **规则描述**
  - 在IBGP环境内，该条规则不适用



10. 优选最老的EBGP邻居的路由

- **规则描述**
  - 在配置了bgp bestpath compare-routerid命令后，将跳过该条原则，拥有最小BGP RouterID的邻居被选为最优

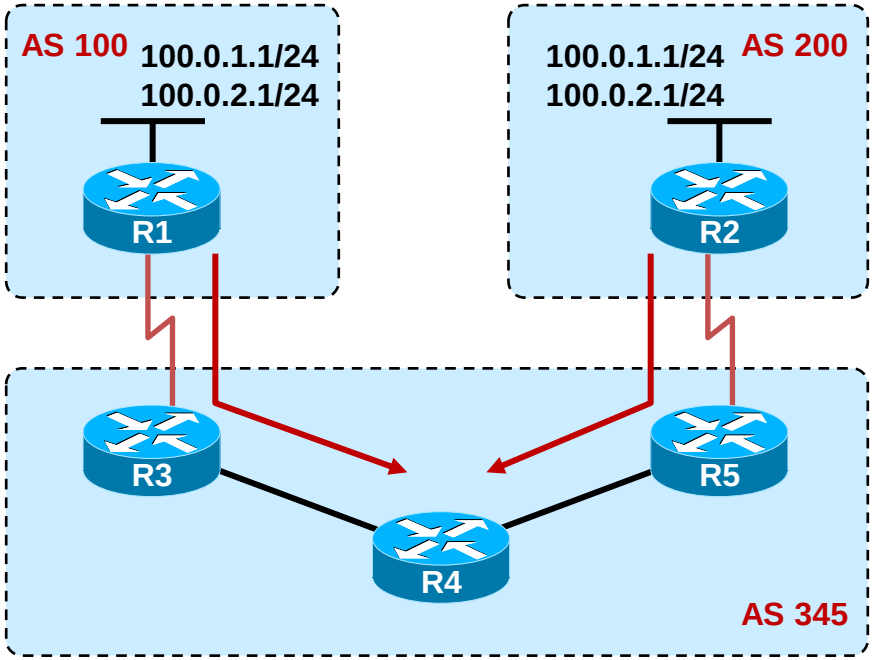


BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

11.优选RouterID最小的BGP邻居的路由

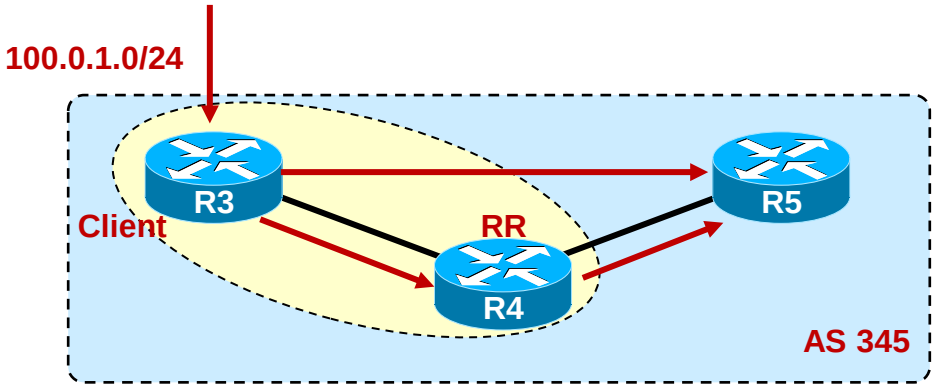
- **实验验证**



- 在不做任何策略的情况下，R4同时从R3及R5收到100.0.1.0及2.0的BGP路由，规则1-10无法决策，规则11优选RID小的BGP邻居，也就是R3（RID为3.3.3.3）

11.优选RouterID最小的BGP邻居的路由

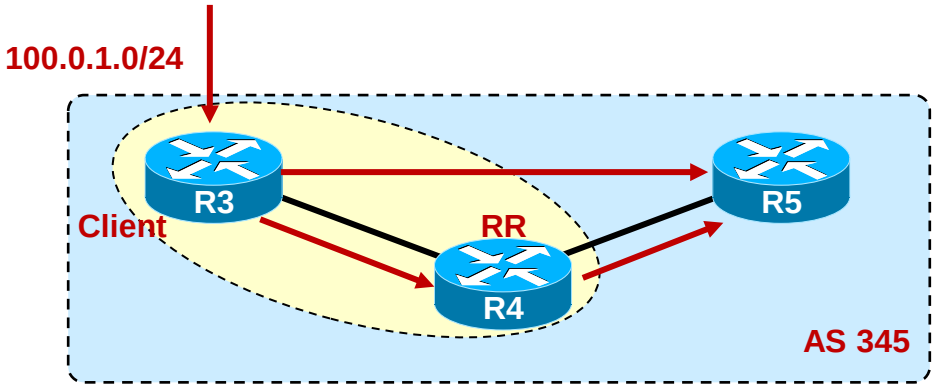
- **规则补充：**如果一条路径包含RR属性，originator属性值将在规则11的最优路径选择过程中代替RouterID



- R3-R4，R4-R5，R3-R4建立IBGP邻居关系（通过各自Loopback口建立）
- R4为RR，R3为R4的client
- R3为R4及R5配置next-hop-self
- AS100内的R1将路由100.0.1.0/24更新给R3
- R3将100.0.1.0/24的路由更新给R5；R4也将路由反射给R5
- 那么R5将如何优选?

11.优选RouterID最小的BGP邻居的路由

- **规则补充：**如果一条路径包含RR属性，originator属性值将在规则11的最优路径选择过程中代替RouterID



R5#show ip bgp 100.0.1.0

BGP routing table entry for 100.0.1.0/24, version 7

Paths: (2 available, best #1, table Default-IP-Routing-Table)

Flag:

Not advertised to any peer

2

100

3.3.3.3 (metric 129) from 4.4.4.4 (4.4.4.4)

Origin IGP, metric 0, localpref 100, valid, internal

Originator: 3.3.3.3, Cluster list: 4.4.4.4

到达该NEXT_HOP的metric(IGP)

BGP邻居的更新源地址

NEXT_HOP

邻居的RouterID

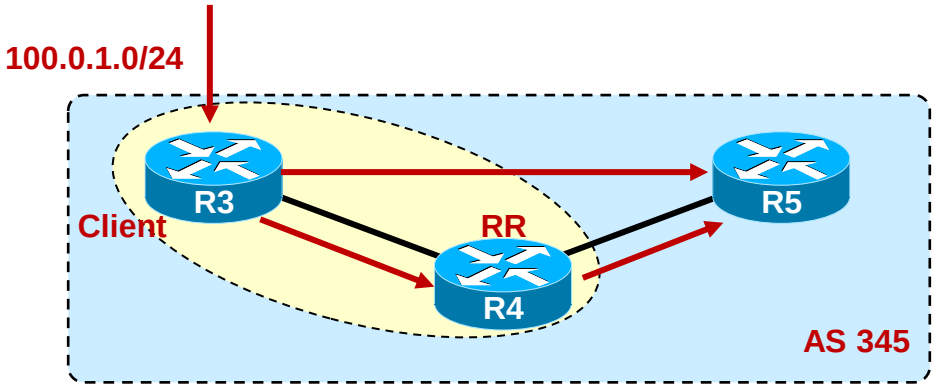
路由始发者的RouterID

默认是RR的RouterID

未完待续…

11.优选RouterID最小的BGP邻居的路由

- **规则补充：**如果一条路径包含RR属性，originator属性值将在规则11的最优路径选择过程中代替RouterID

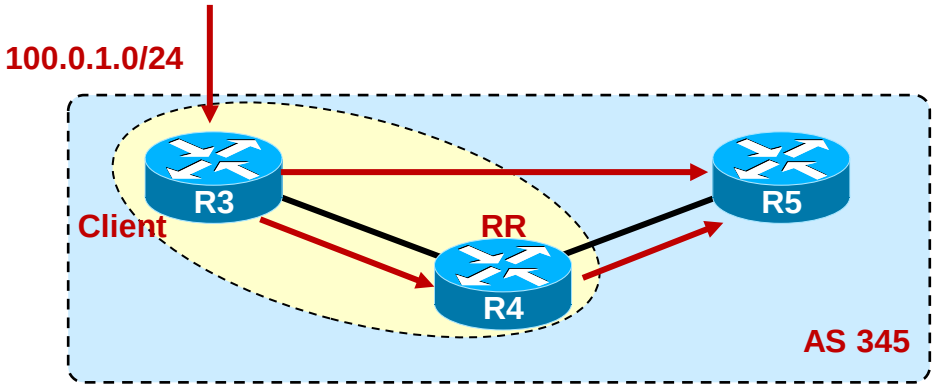


R5#show ip bgp 100.0.1.0

```
BGP routing table entry for 100.0.1.0/24, version 7
Paths: (2 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
  Not advertised to any peer
      2
  100
    3.3.3.3 (metric 129) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal,best
```

11.优选RouterID最小的BGP邻居的路由

- **规则补充：**如果一条路径包含RR属性，originator属性值将在规则11的最优路径选择过程中代替RouterID



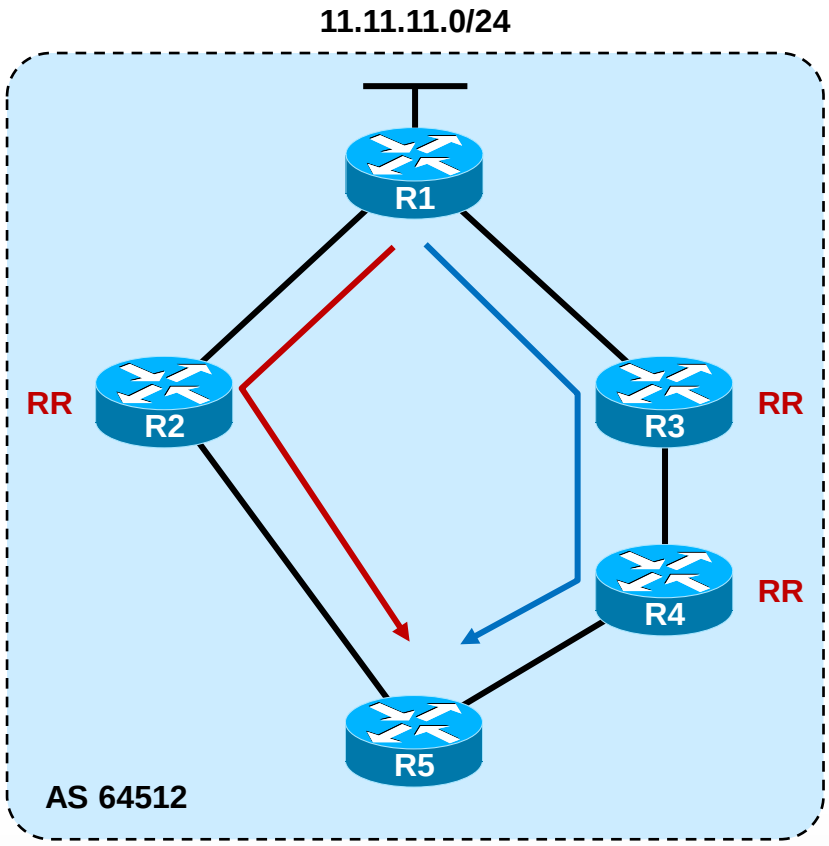
- 可以看出最终R5优选R3传来的路由。这里看似比较到规则11，R3的RouerID为3.3.3.3小于R4的RouterID4.4.4.4所以优选R3，其实不是。
- 由于R4反射给R5的路由携带了Originator属性，因此这个属性值3.3.3.3将在规则11的比较中取代R4的RouterID进行比较，那么这里比较结果是相等，因此规则11无法做决策。（测试的方法很简单，将R4的BGP RouterID改小即可）
- 最终在规则12中，由于从R3传来的路由Cluster_List长度为0，也就是没有，而从R4反射过来的路由Cluster_list长度为1，因此优选R3。

BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

12. 优选Cluster_List最短的路由

- **实验验证**



R5 show ip bgp 11.11.11.0

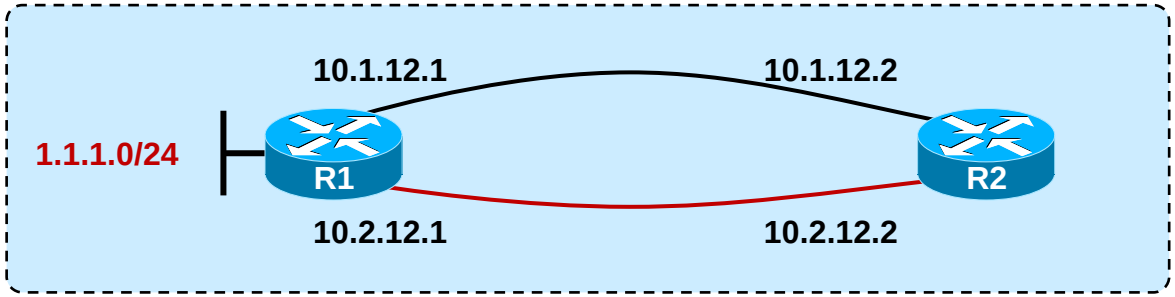
BGP routing table entry for 11.11.11.0/24, version 2
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Flag: 0x820
  Not advertised to any peer
  Local
    1.1.1.1 (metric 129) from 4.4.4.4 (4.4.4.4)
      Origin IGP, metric 0, localpref 100, valid, internal
      Originator: 1.1.1.1, Cluster list: 4.4.4.4, 3.3.3.3
  Local
    1.1.1.1 (metric 129) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, **best**
      Originator: 1.1.1.1, Cluster list: 2.2.2.2

BGP选路规则

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

选择邻居IP地址最小的路由

- **实验验证**



R1的配置如下：

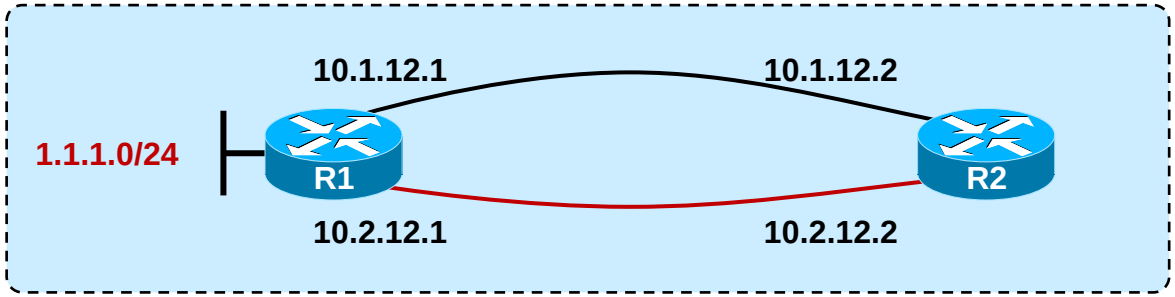
```
router bgp 12
 bgp router-id 1.1.1.1
 network 1.1.1.0 mask 255.255.255.0
 neighbor 10.1.12.2 remote-as 12
 neighbor 10.2.12.2 remote-as 12
 no synchronization
```

R2的配置如下：

```
router bgp 12
 bgp router-id 2.2.2.2
 neighbor 10.1.12.1 remote-as 12
 neighbor 10.2.12.1 remote-as 12
 no synchronization
```

选择邻居IP地址最小的路由

- **实验验证 cont.**



R2#show ip bgp 1.1.1.0

```
BGP routing table entry for 1.1.1.0/24, version 2
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  Local
    10.2.12.1 from 10.2.12.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, internal
  Local
    10.1.12.1 from 10.1.12.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

红茶三杯
Vinsoney

学习 沉淀 成长 分享

关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

BGP非等价负载均衡
Cost Community

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2012-08-01

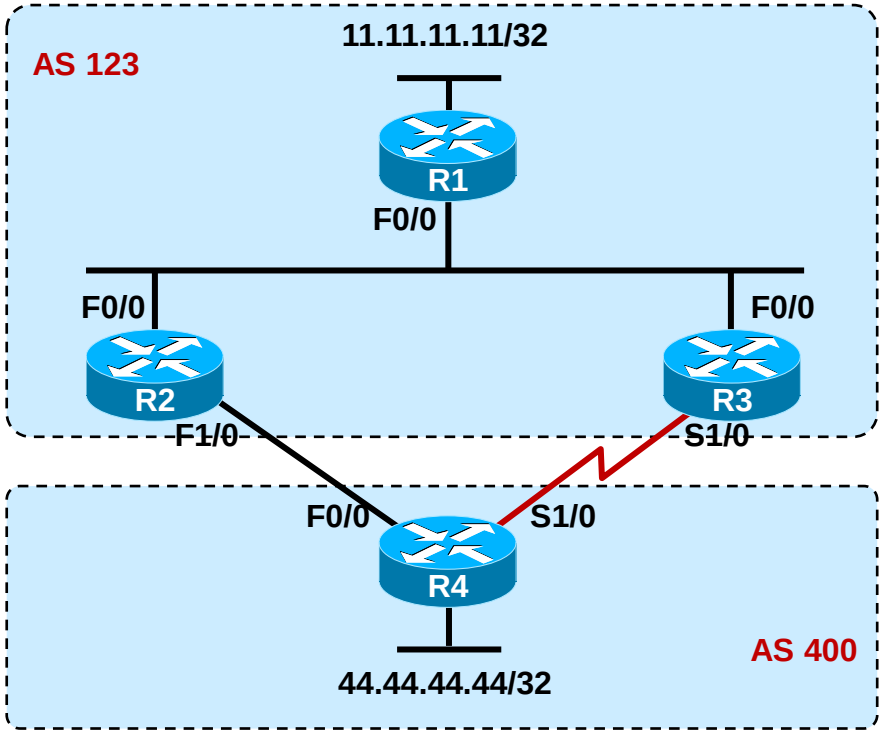
课程目标

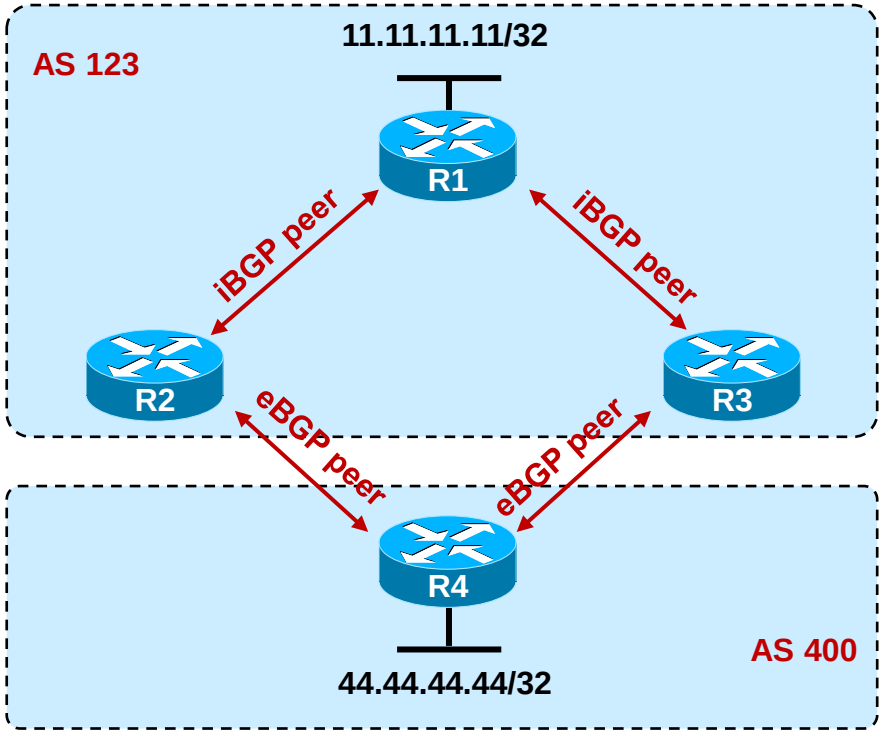
- BGP非等价负载均衡
- Cost Community

红茶三杯
Vinsoney

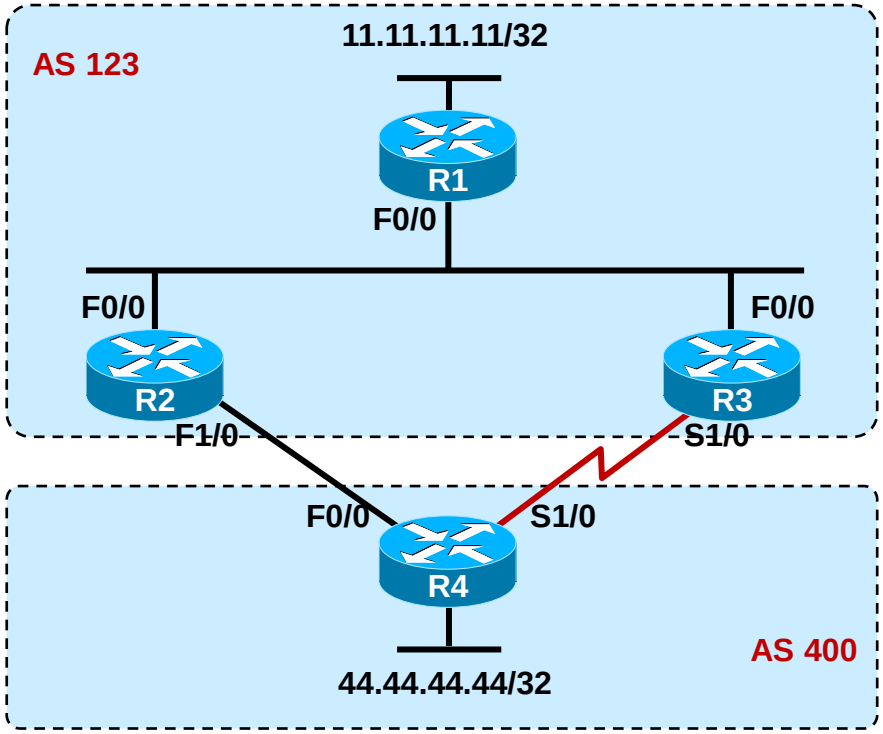
BGP非等价负载均衡

环境描述





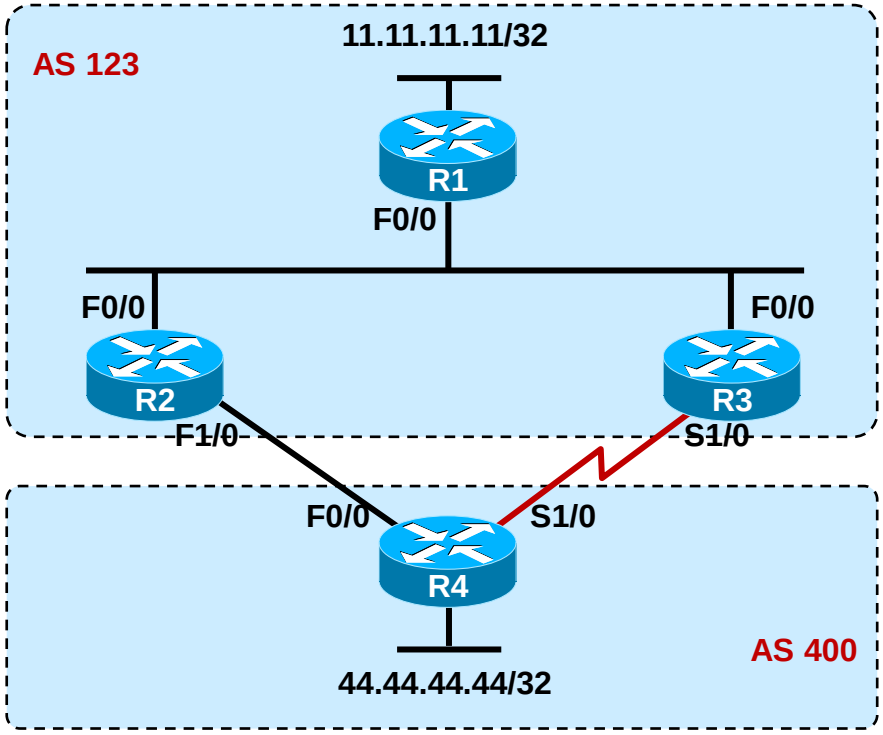
eBGP非等价负载均衡



R4的配置如下：

```
router bgp 400
 bgp router-id 4.4.4.4
 bgp dmzlink-bw
 network 44.44.44.44 mask 255.255.255.255
 neighbor 10.1.24.2 remote-as 123
 neighbor 10.1.24.2 dmzlink-bw
 neighbor 10.1.34.3 remote-as 123
 neighbor 10.1.34.3 dmzlink-bw
 maximum-paths 2
 no auto-summary
```

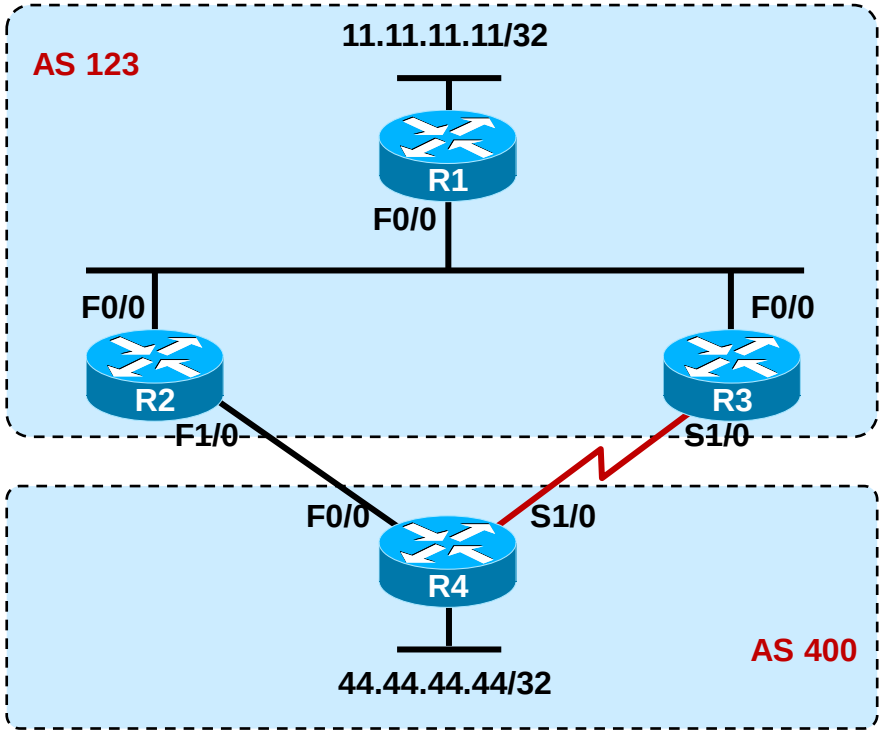
eBGP非等价负载均衡



R4的BGP表项

R4#show ip bgp 11.11.11.11
BGP routing table entry for 11.11.11.11/32, version 8
Paths: (2 available, best #1, table Default-IP-Routing-Table)
Multipath: eBGP
Flag: 0x840
  Advertised to update-groups:
     1
  123
    10.1.24.2 from 10.1.24.2 (2.2.2.2)
      Origin IGP, localpref 100, valid, external, multipath, best
      DMZ-Link Bw 12500 kbytes
  123
    10.1.34.3 from 10.1.34.3 (3.3.3.3)
      Origin IGP, localpref 100, valid, external, multipath
      DMZ-Link Bw 193 kbytes

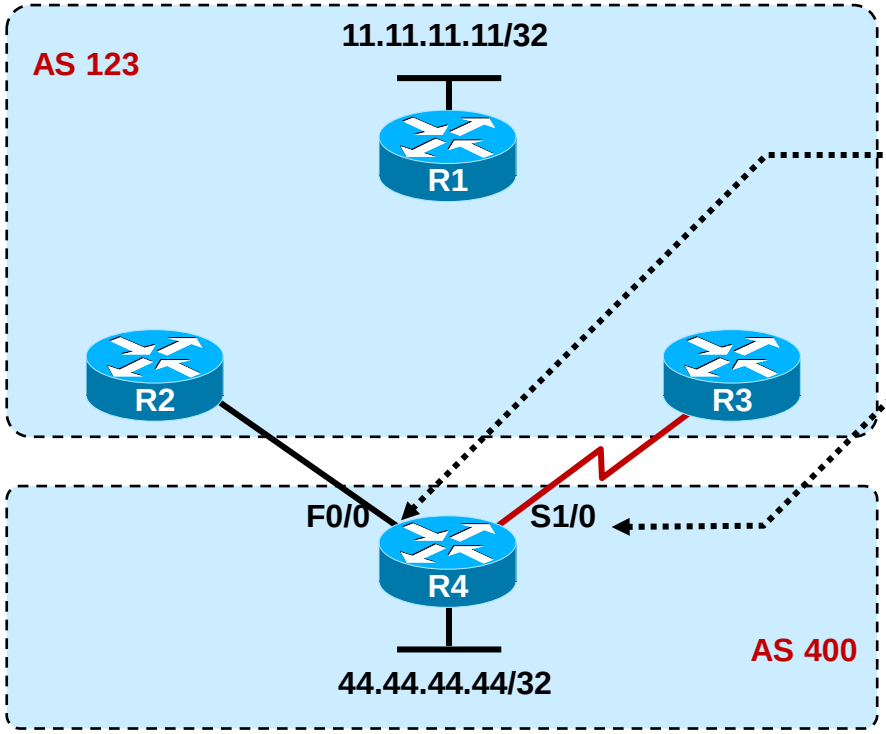
eBGP非等价负载均衡



R4的路由表

```
R4#show ip route 11.11.11.11
Routing entry for 11.11.11.11/32
  Known via "bgp 400", distance 20, metric 0
  Tag 123, type external
  Last update from 10.1.34.3 00:00:02 ago
  Routing Descriptor Blocks:
    10.1.34.3, from 10.1.34.3, 00:00:02 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 123
  * 10.1.24.2, from 10.1.24.2, 00:00:02 ago
      Route metric is 0, traffic share count is 60
      AS Hops 1
      Route tag 123
```

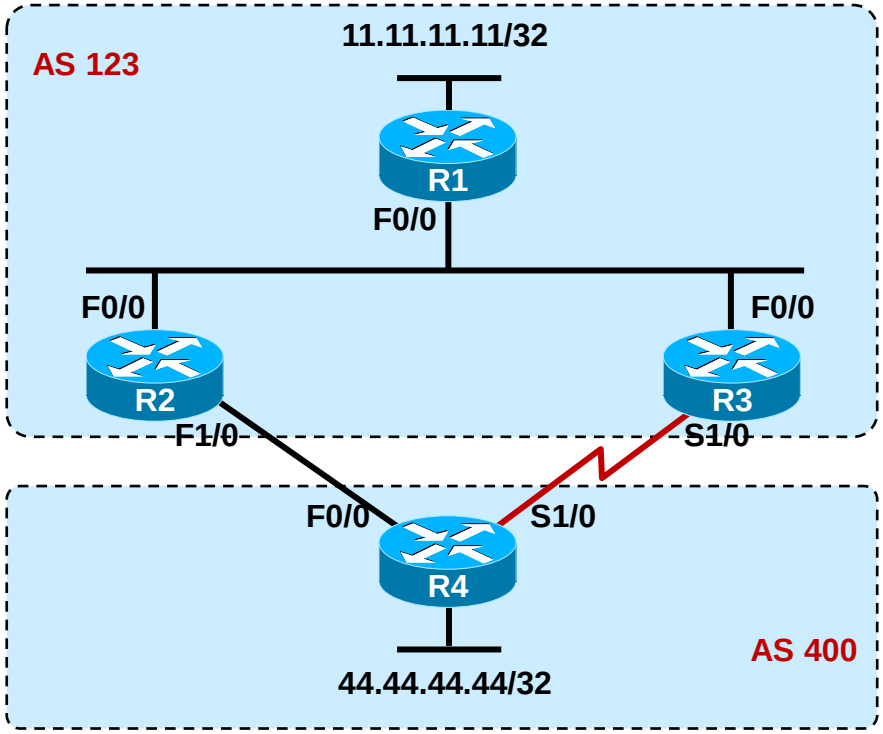
eBGP非等价负载均衡



ip load-sharing per-packet

注意此时虽然在R4上路由表中已经出现了非等价负载均衡的两条路由，但是由于该设备目前的交换机制是CEF，而CEF默认的负载均衡方式是per-desination的方式，实际上就是基于源、目地址对的负载均衡。因此为了真正实现非等价负载均衡，需要在R4的F0/0及S1/0接口上配置：ip load-sharing per-packet。这样一来当R4下挂的用户要访问11.11.11.11网络，数据就会被R4在F0/0及S1/0口上执行非等价负载均衡。流量的比例大致是1/60

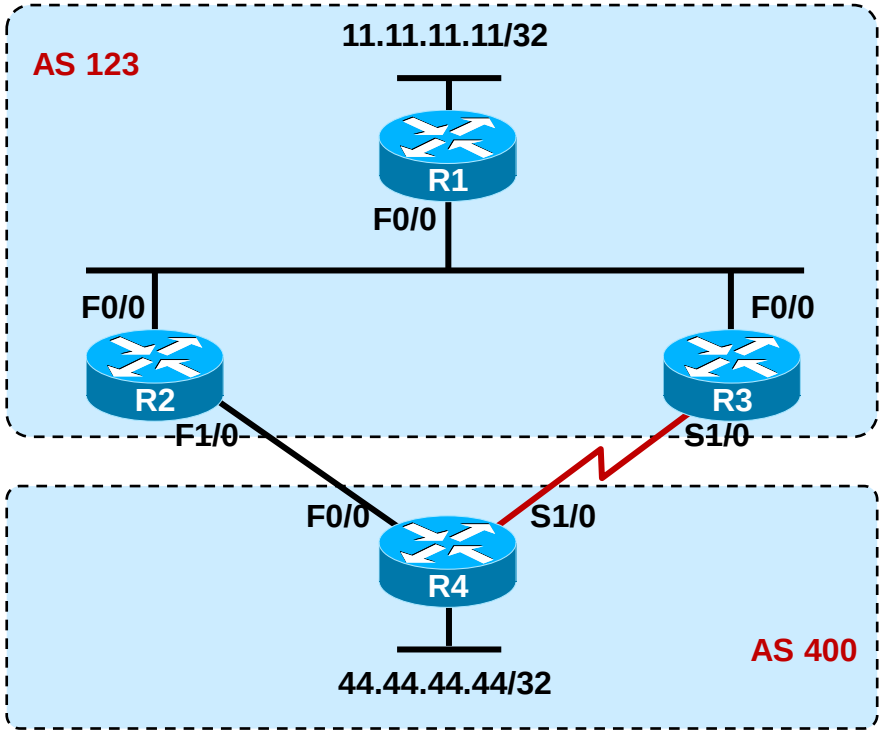
iBGP非等价负载均衡



R1的配置如下：

```
router bgp 123
 bgp router-id 1.1.1.1
 bgp dmzlink-bw
 neighbor 2.2.2.2 remote-as 123
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 3.3.3.3 remote-as 123
 neighbor 3.3.3.3 update-source Loopback0
 maximum-paths ibgp 2
```

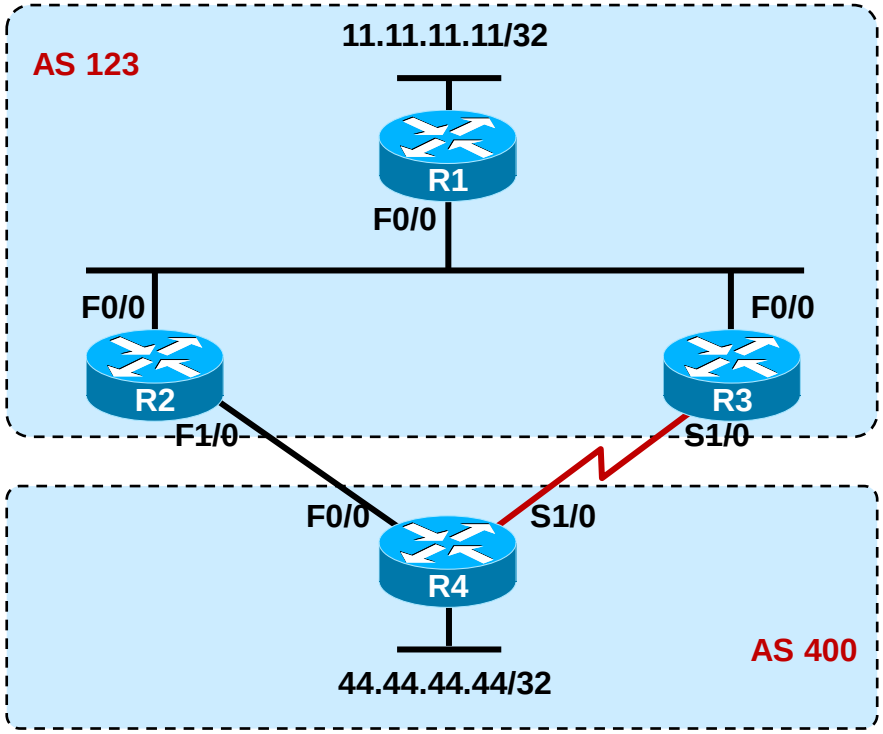
iBGP非等价负载均衡 cont.



R2的配置如下：

```
router bgp 123
 bgp router-id 2.2.2.2
 bgp dmzlink-bw
 neighbor 1.1.1.1 remote-as 123
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.1.1.1 next-hop-self
 neighbor 1.1.1.1 send-community extended
 neighbor 10.1.24.4 remote-as 400
 neighbor 10.1.24.4 dmzlink-bw
```

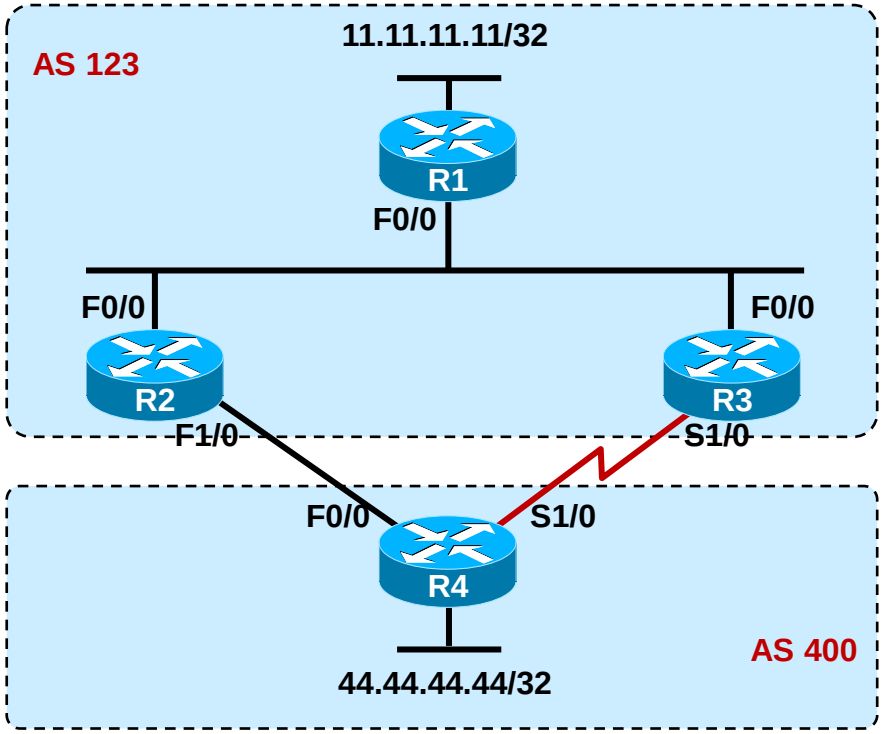
iBGP非等价负载均衡 cont.



R3的配置如下：

```
router bgp 123
 bgp router-id 3.3.3.3
 bgp dmzlink-bw
 neighbor 1.1.1.1 remote-as 123
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.1.1.1 next-hop-self
 neighbor 1.1.1.1 send-community extended
 neighbor 10.1.34.4 remote-as 400
 neighbor 10.1.34.4 dmzlink-bw
```

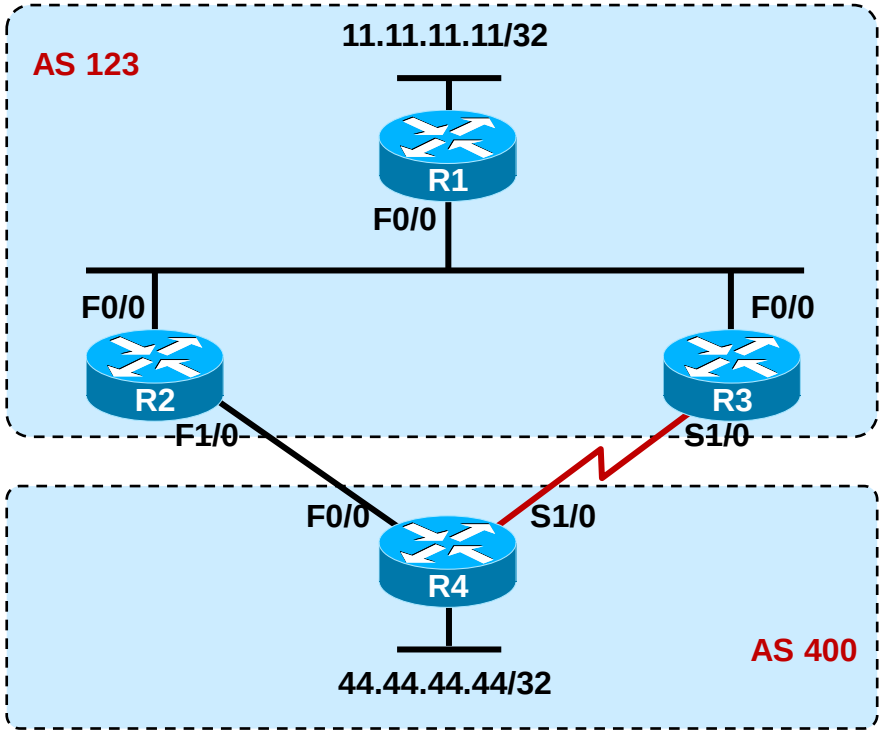
iBGP非等价负载均衡



R1的BGP表

R1#show ip bgp 44.44.44.44
BGP routing table entry for 44.44.44.44/32, version 4
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Multipath: iBGP
  Not advertised to any peer
  400
    3.3.3.3 (metric 2) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      DMZ-Link Bw 193 kbytes
  400
    2.2.2.2 (metric 2) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath, best
      DMZ-Link Bw 12500 kbytes

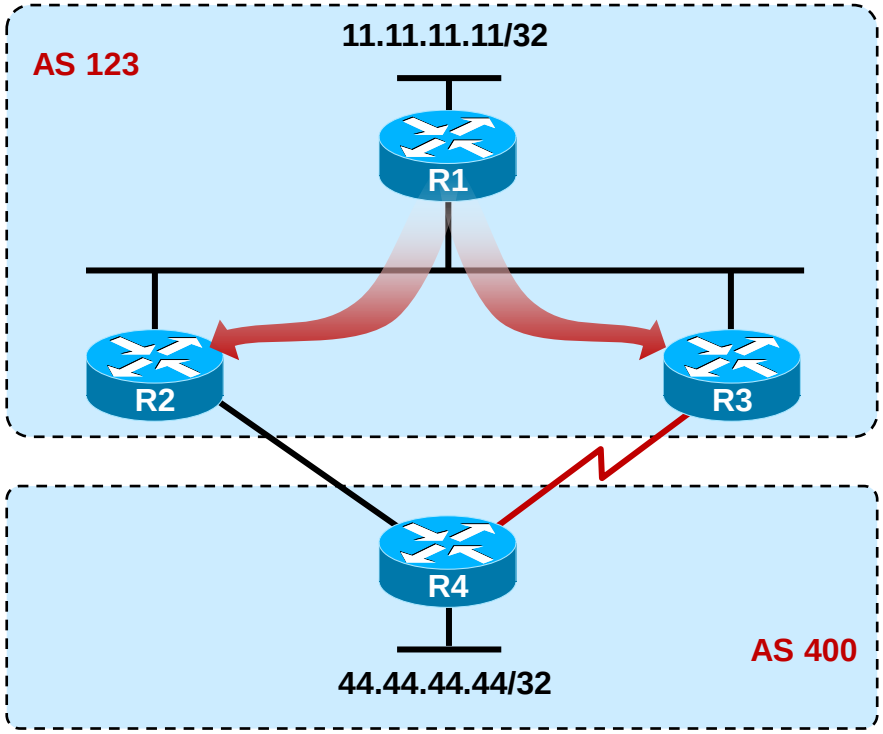
iBGP非等价负载均衡



R1的路由表

```
R1#show ip route 44.44.44.44
Routing entry for 44.44.44.44/32
  Known via "bgp 123", distance 200, metric 0
  Tag 400, type internal
  Last update from 2.2.2.2 00:10:40 ago
  Routing Descriptor Blocks:
  * 3.3.3.3, from 3.3.3.3, 00:10:40 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 400
    2.2.2.2, from 2.2.2.2, 00:10:40 ago
      Route metric is 0, traffic share count is 60
      AS Hops 1
      Route tag 400
```

iBGP非等价负载均衡



R1的路由表

```
R1#show ip route 44.44.44.44
Routing entry for 44.44.44.44/32
  Known via "bgp 123", distance 200, metric 0
  Tag 400, type internal
  Last update from 2.2.2.2 00:10:40 ago
  Routing Descriptor Blocks:
  * 3.3.3.3, from 3.3.3.3, 00:10:40 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 400
    2.2.2.2, from 2.2.2.2, 00:10:40 ago
      Route metric is 0, traffic share count is 60
      AS Hops 1
      Route tag 400
```

BGP Link Bandwidth

- 通过利用BGP的Link Bandwidth特性，我们能够在AS边界路由器外部链路带宽不等价的情况下，实现BGP路由的非等价负载均衡。该特性通过在BGP进程的IPv4或VPNv4地址族中使用bgp dmzlink-bw命令激活。这个特性搭配BGP multipath特性，即可实现非等价负载均衡。
- BGP Link Bandwidth特性用于在扩展Community属性中通告一条AS出口链路的带宽。
- 这个特性配置在一台AS边界路由器上，指向其eBGP邻居，那么此时该特性所描述的就是该路由器与其eBGP邻居之间的链路带宽。而且该链路带宽信息（使用扩展Community描述）除了AS边界路由器自己用（用于本地的非等代价负载均衡），还可以向AS内的iBGP邻居传递，当然，前提是得配置send-community extended.

BGP Link Bandwidth的预备条件

- 需先激活maximum-paths特性
- 当要向iBGP邻居通告Link Bandwidth特性时，需send-community extended
- CEF或dCEF必须在所有参与该特性的路由器上打开

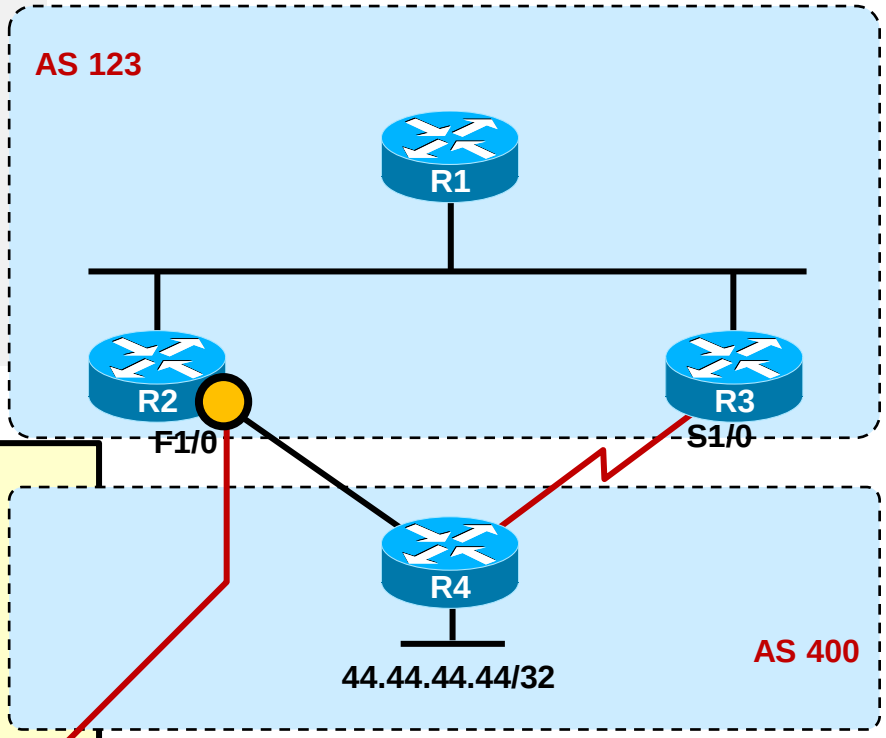
BGP Link Bandwidth特性的限制

- 该特性只能配置在BGP进程的IPv4或VPNv4地址族下
- BGP只能够在Link Bandwidth Community中通告与eBGP邻居直连的链路（接口）带宽
- 在IPv4及VPNv4地址族中，iBGP及eBGP负载均衡都是支持的；但是 eiBGP负载均衡却只能在VPNv4地址族中才能够支持，也就是iBGP与eBGP的负载均衡。

BGP Link Bandwidth Extended Community

R2的配置如下：

```
router bgp 123
 bgp router-id 2.2.2.2
 bgp dmzlink-bw
 neighbor 1.1.1.1 remote-as 123
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.1.1.1 next-hop-self
 neighbor 1.1.1.1 send-community extended
 neighbor 10.1.24.4 remote-as 400
 neighbor 10.1.24.4 dmzlink-bw
```




```
R2#sh ip b 44.44.44.44
… …
  400
    10.1.24.4 from 10.1.24.4 (4.4.4.4)
      Origin IGP, metric 0, localpref 100, valid,
external, best
      DMZ-Link Bw 12500 kbytes
```

红茶三杯
Vinsoney

COST COMMUNITY

Cost Community概述

- Cost Community是一个**扩展的Community属性，只能传递给iBGP邻居或联邦peer（含联邦iBGP及联邦eBGP邻居），不能传递给eBGP邻居；**
- 通过利用Cost Community，我们能够在一个AS或联邦内部自定义BGP的最优路径选择。Cost Community事实上是提供给我们除了“BGP13条选路规则”之外的又一**“插入点”（ point of insertion），相当于提供给我们另一个操控路由优选的手柄。**
- 针对internal路由，在route-map中使用set extcommunity cost命令去设置cost community值。在上述set命令后，配置**一个ID（0-255）以及cost number（0-4294967295）**。Cost number值可以影响路径的优选，越小越优先。如果两个路径cost number值相等，那么拥有小ID值的被优选。（这里不能死记，具体的PK方法，在下文有详细介绍）

Cost Community概述 cont.

- Cost community属性值是扩展community，在向邻居发送前需配置neighbor send-community extand。
- 下述命令可以跟route-map来设置Cost Community：
  - aggregate-address
  - neighbor default-originate route-map {in | out}
  - neighbor route-map
  - network route-map
  - redistribute route-map

POI（igp）

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED

POI=IGP

7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

POI（pre-bestpath）

POI=pre-bestpath

1. 优选具有最大Weight值的路由
2. 优选具有最大Local_Pref值的路由
3. 优选起源于本地的路由（如本地network、aggregate或redistribute的）即下一跳是0.0.0.0(在BGP 表中,当前路由器通告的路由的下一跳为0.0.0.0)
4. 优选AS-Path最短的路由
5. Origin（IGP > EGP > Incomplete）
6. 优选MED最小的路由，默认情况下仅有当所有的备选路由来自同一AS才会比较MED
7. 优选EBGP邻居发来的路由（相对于IBGP邻居学过来的），在联邦EBGP和IBGP中首选联邦EBGP 路由
8. 优选到BGP NEXT_HOP最近的路由
9. BGP负载均衡
10. 优选最老的EBGP邻居传来的路由，降低滚翻的影响（此条主要对EBGP路由起效，但是现在基本不用该条，因不确定性太大）
11. 优选RouterID最小的BGP邻居的路由
12. 优选Cluster-List 最短的路由
13. 优选邻居ip地址（BGP的neighbor配置中的那个地址）最小的路由

Cost Community的限制

- Cost Community特性只能部署在一个AS或联邦AS（大AS）内部。她是一个扩展Community属性并且只能传递给iBGP邻居或联邦邻居（联邦iBGP或联邦eBGP邻居），不能传递给eBGP邻居（不会报错，就是单纯的不携带）。
- 在部署Cost Community之前需确保AS或联邦内所有路由器都能识别她，并且要在AS或联邦内保证Cost Community配置的连续性，以防止潜在的环路问题。
- Multiple cost community set clauses may be configured with the set extcommunity cost command in a single route map block or sequence. However, each set clause must be configured with a different ID value (0-255) for each point of insertion (POI). The ID value determines preference when all other attributes are equal. The lowest ID value is preferred.

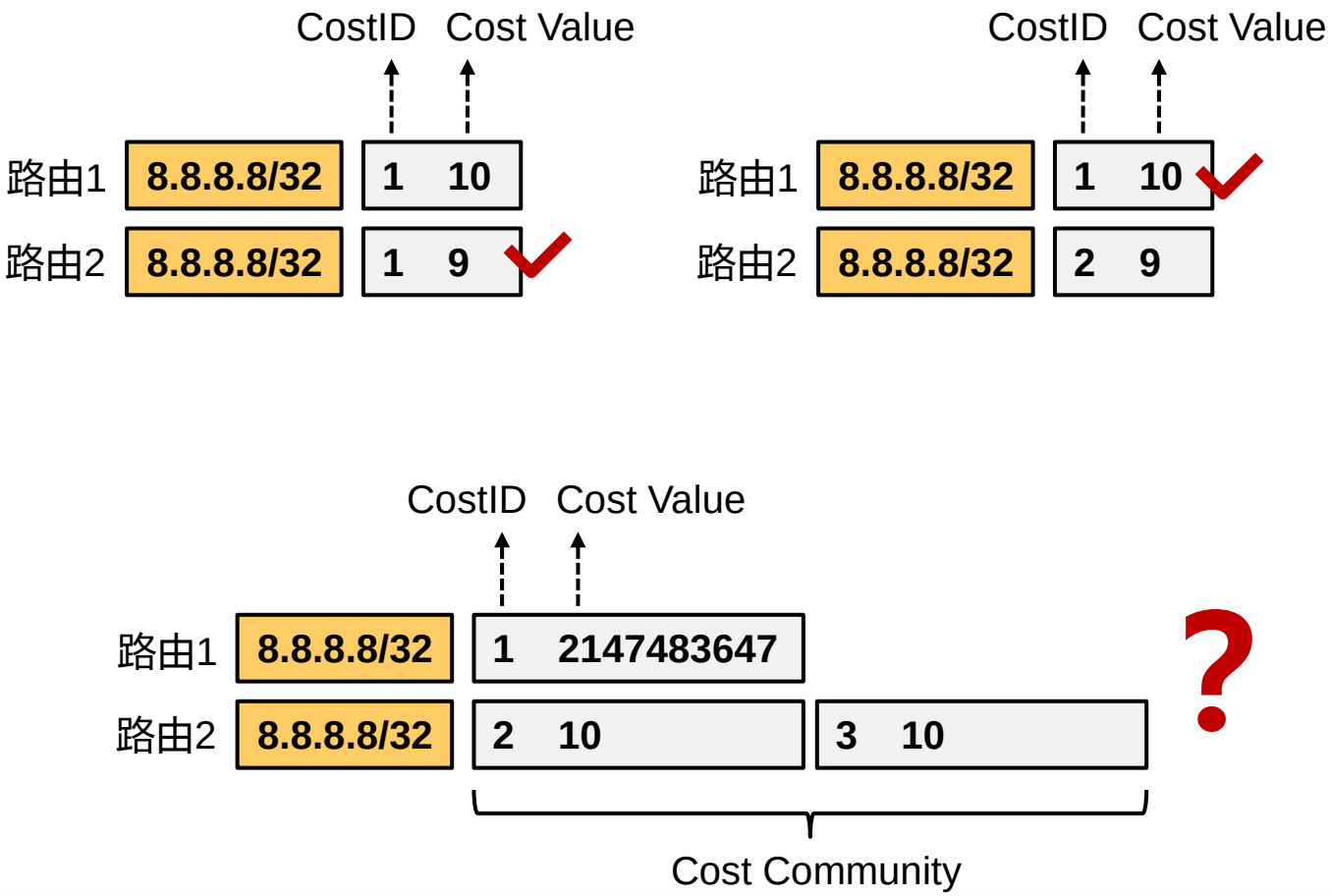
Cost Community如何影响BGP最佳路径选择

- Cost Community事实上是提供给我们除了“BGP13条选路规则”之外的又一“插入点”（ point of insertion ），相当于提供给我们另一个操控路由优选的手柄。
- 默认情况下，这个插入点在“BGP13条选路规则”的规则八之后，规则九之前，也就是在负载均衡规则的前面。当一个BGP路由器有多条路径可达同一个目的地，选路规则进程会决定哪条路径是best，这条best的路径最终被安装进路由表使用。Cost Community允许我们在此之前横刀进入，干预路由的优选。如果路由器在本地路由优选过程中不能识别cost community，那么就默默的忽略cost community。

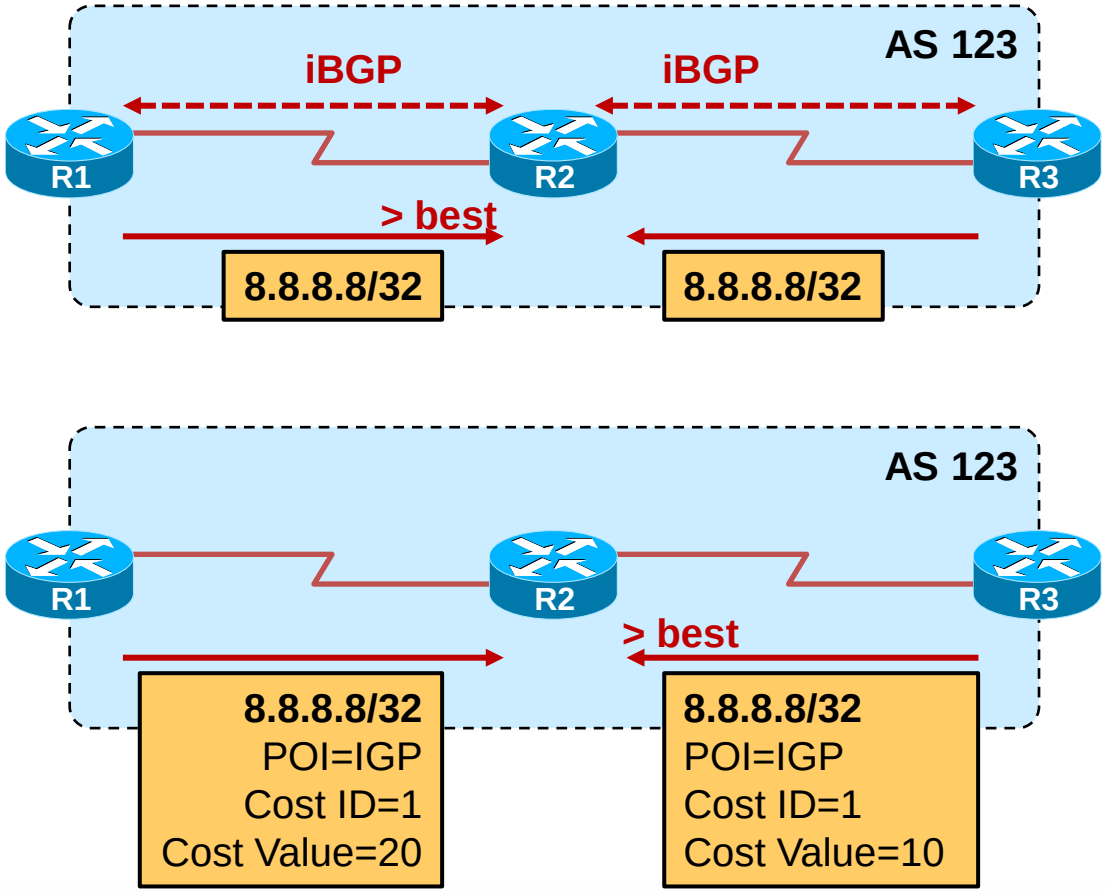
Cost Community如何影响BGP最佳路径选择 cont.

- 我们可以针对多条路径，在同一个POI设置不同的Cost Community。当这些路由进行PK时，cost value最低的优选，如果cost value相等，则优选拥有最小cost ID的路由。如果某条路由没有携带cost community，那么在cost community PK的这个环节，一个默认的cost value会被赋给这条路由， 这个值是2147483647，也就是cost community最大可选值4294967295的一半儿。
- 默认情况下，在CISCO IOS中如果BGP路由器收到的某条路由携带了Cost Community，那么这玩意儿就开始工作了（开始在POI影响选路规则并进行路由间的PK），如果本路由器想完全忽略Cost Community，那么可以在BGP进程中配置bgp bestpath cost-community ignore，即可无视Cost Community。

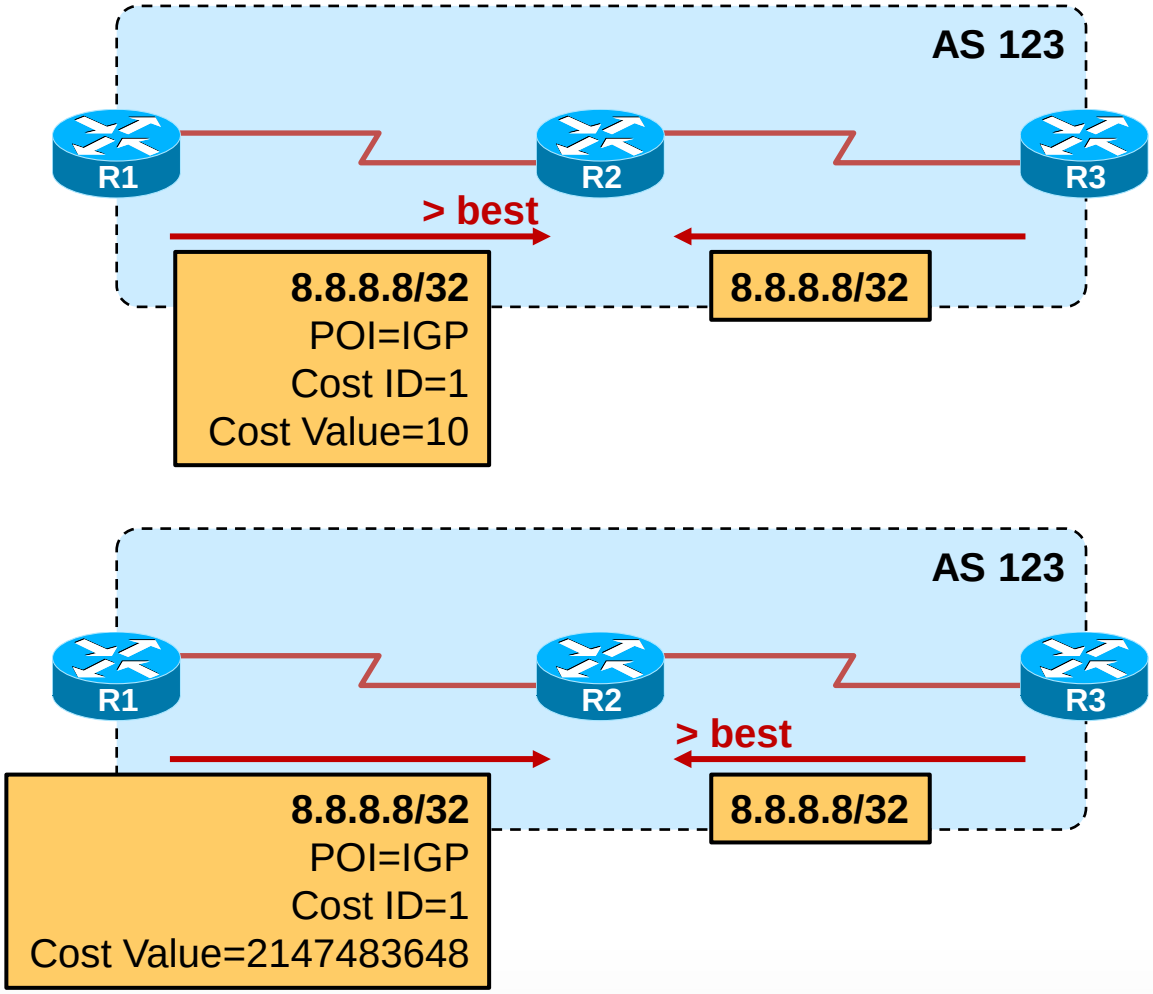
Cost Community



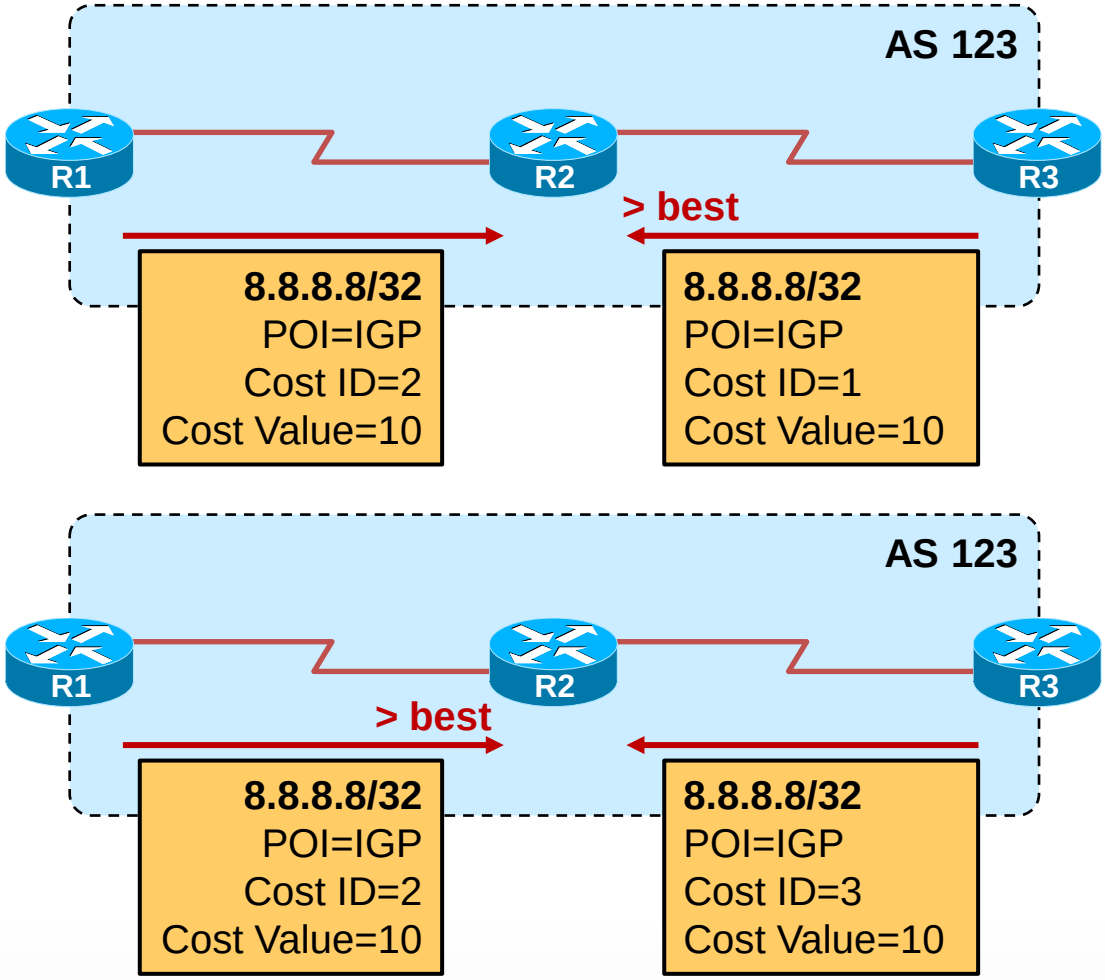
实验：Cost Community基础



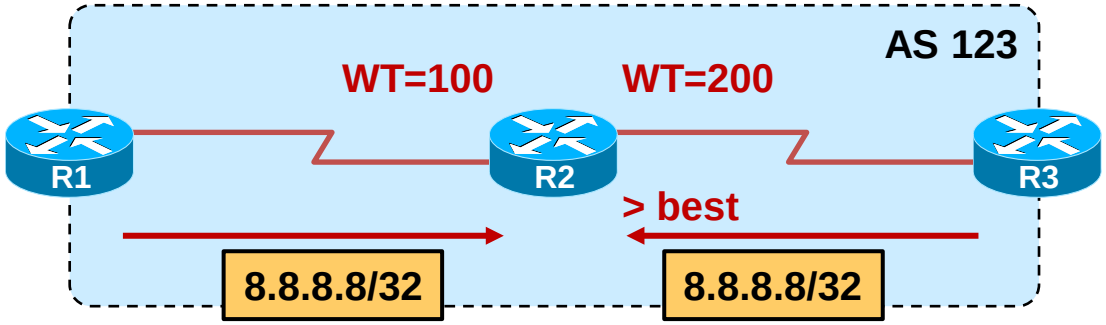
实验：测试默认Cost Community

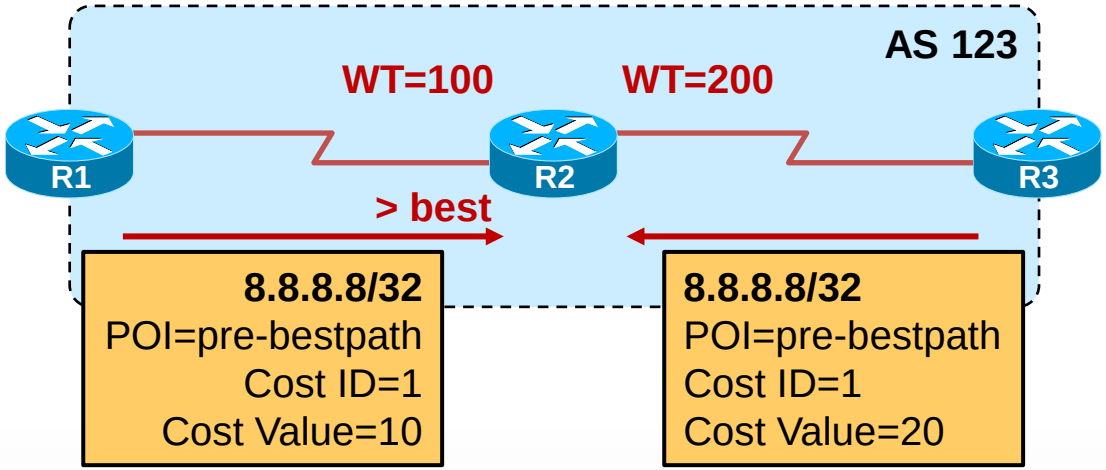


实验：Cost Value和Cost ID的PK

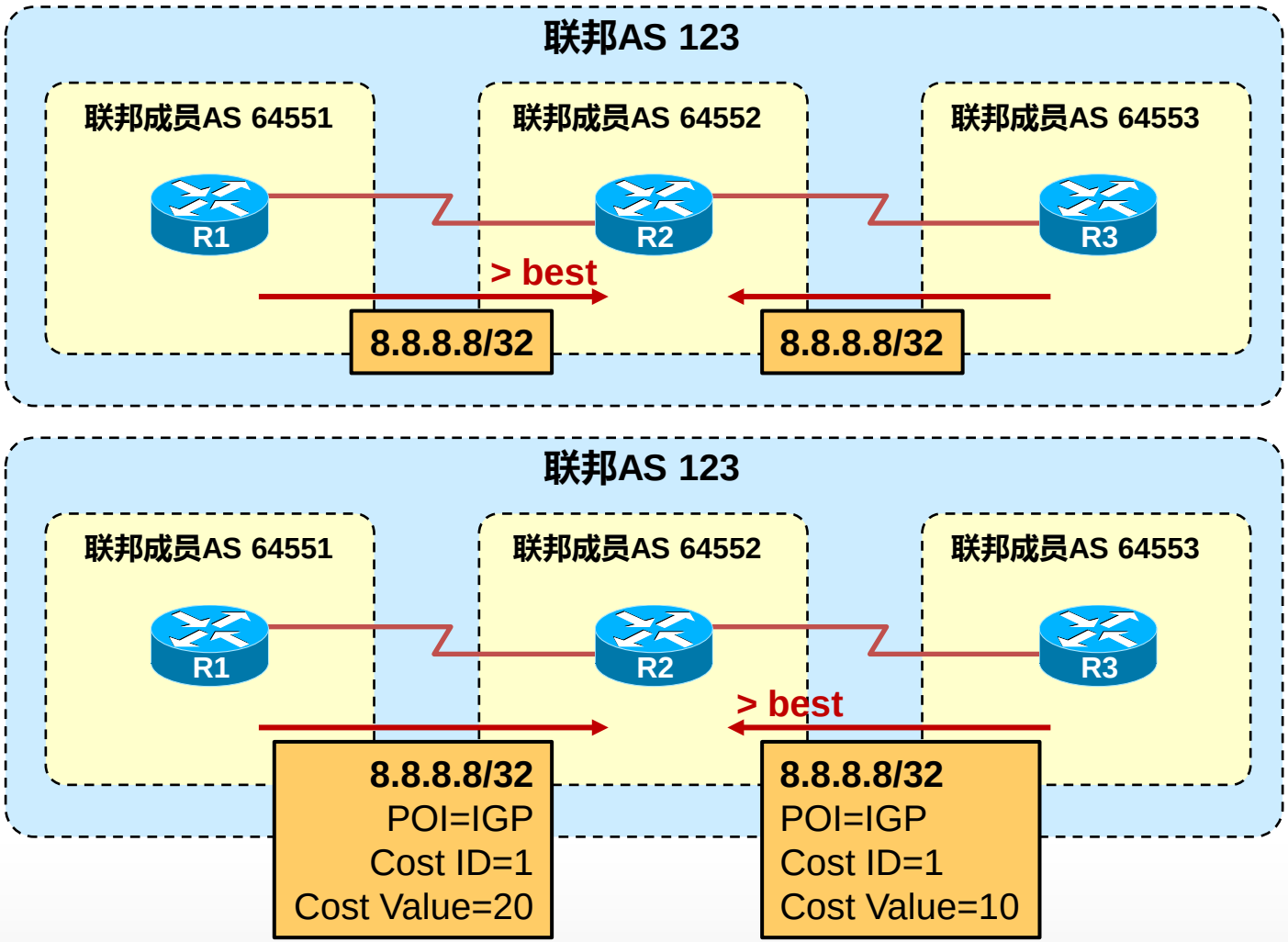


实验：测试pre-bestpath的POI





实验：cost community在联邦eBGP邻居之间的传递



红茶三杯
Vinsoney

学习 沉淀 成长 分享

关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

BGP在客户与运营商网络间的部署

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2012-08-01

课程目标

单宿主应用环境

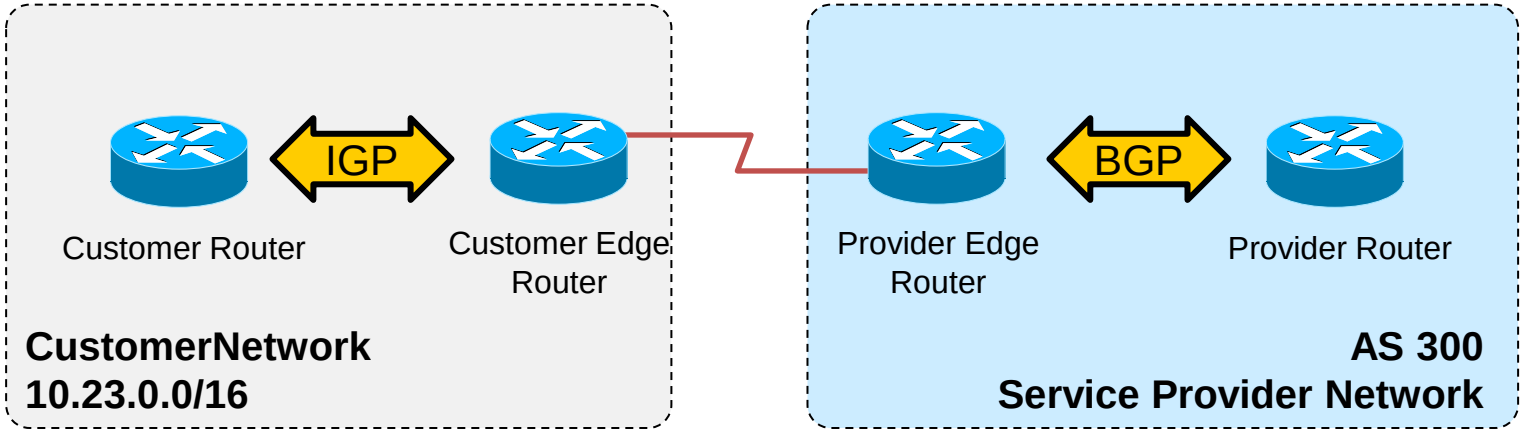
移除私有AS

DUAL AS的实现

红茶三杯
Vinsoney

单宿主应用环境

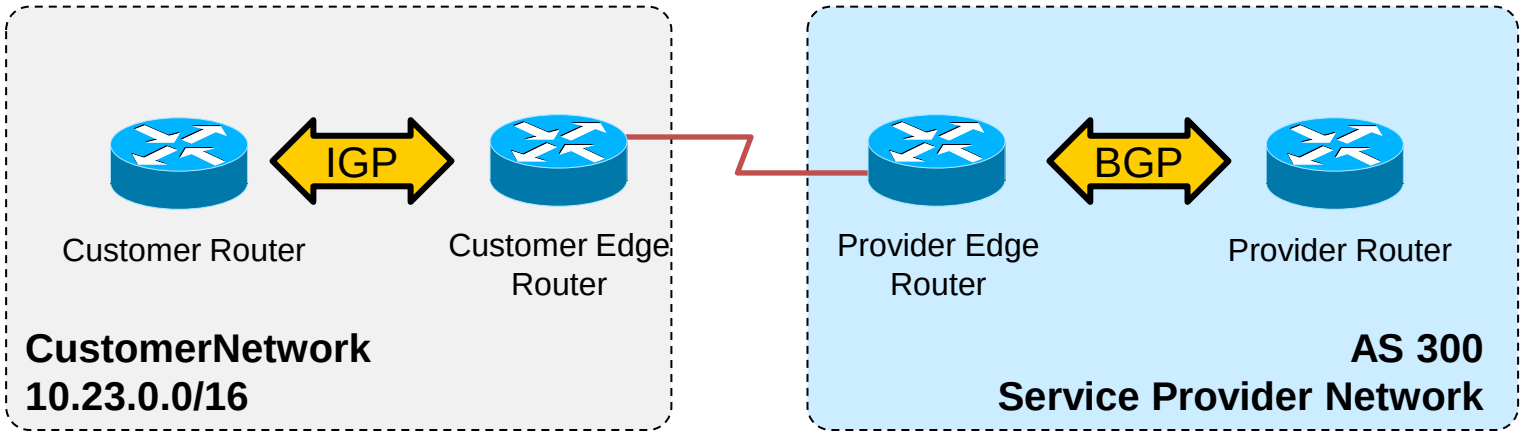
单线环境 静态路由实现




```
ip route 0.0.0.0 0.0.0.0 x.x.x.x
router ospf 1
  default-information originate
```

```
ip route 10.23.0.0 255.255.0.0 x.x.x.x
router bgp 300
  redistribute static [route-map x]
```

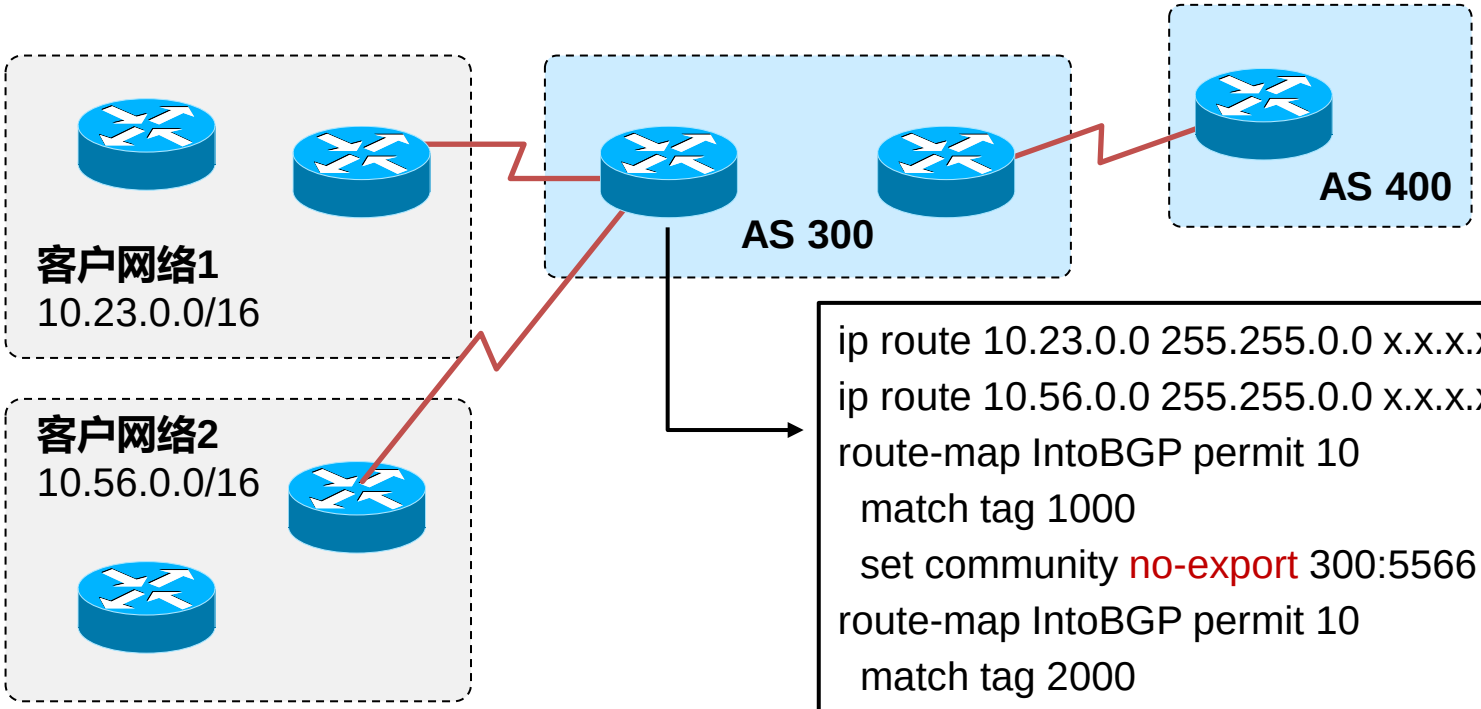
单线环境 静态路由实现



可以给路由设置tag，或者配置community值，以便后续基于这些标记做策略

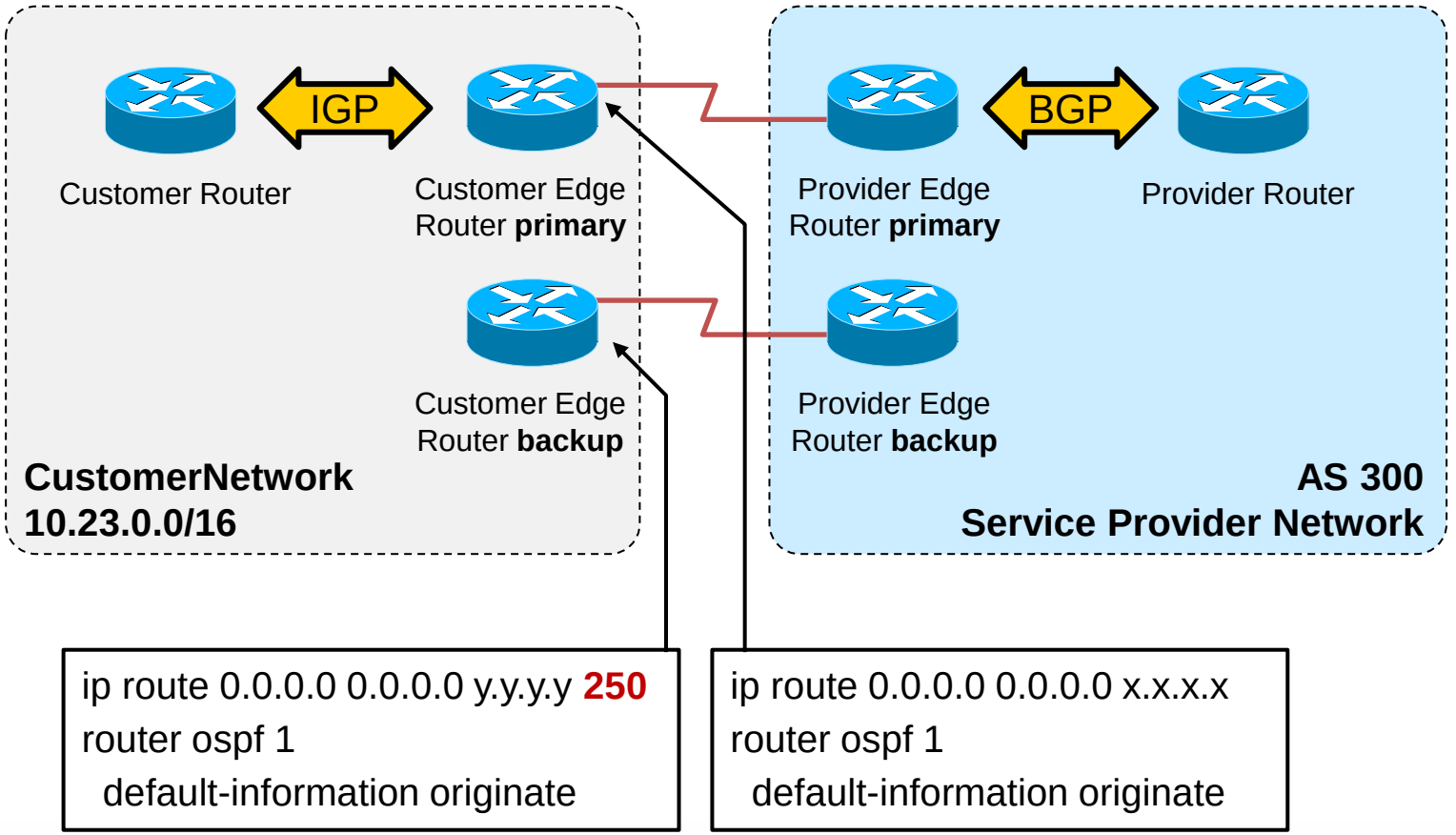
```
ip route 10.23.0.0 255.255.0.0 x.x.x.x
router bgp 300
 redistribute static [route-map x]
 no auto-summary
```

单线环境 静态路由实现 示例

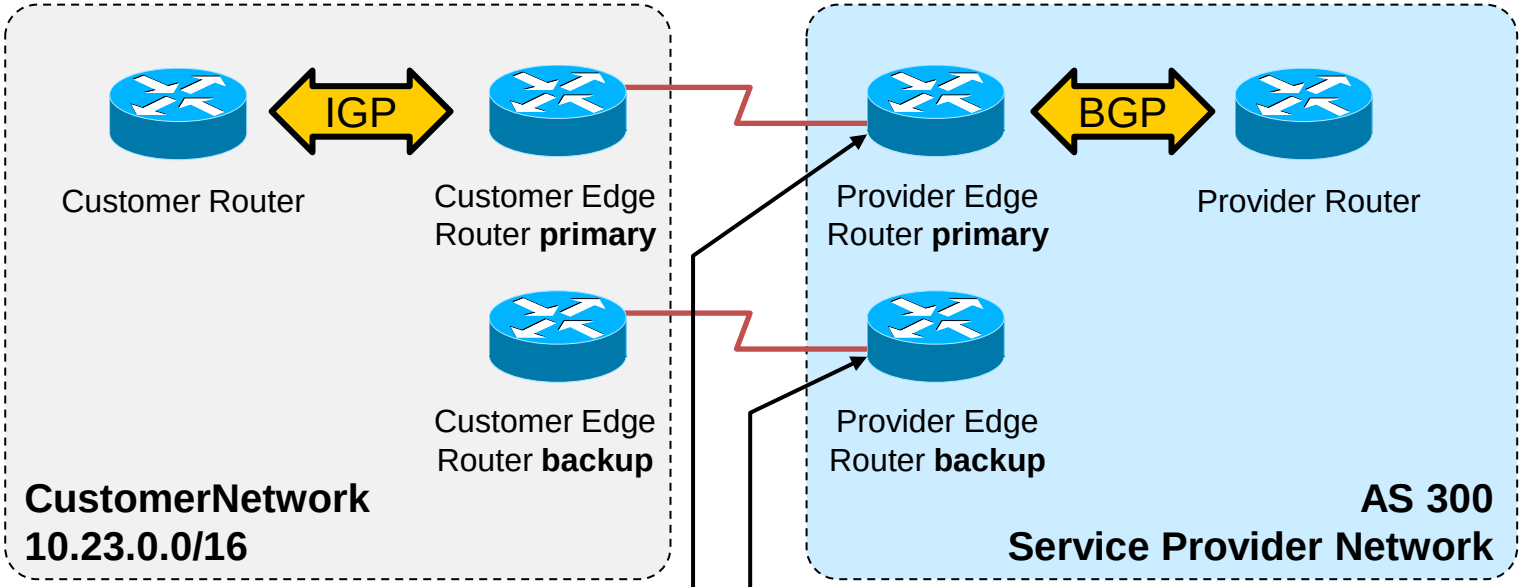



```
ip route 10.23.0.0 255.255.0.0 x.x.x.x tag 1000
ip route 10.56.0.0 255.255.0.0 x.x.x.x tag 2000
route-map IntoBGP permit 10
  match tag 1000
  set community no-export 300:5566
route-map IntoBGP permit 10
  match tag 2000
  set community 300:5566
router bgp 300
  redistribute static route-map IntoBGP
  neighbor x.x.x.x send-community
  no auto-summary
```

双线环境（主备）-客户网络配置



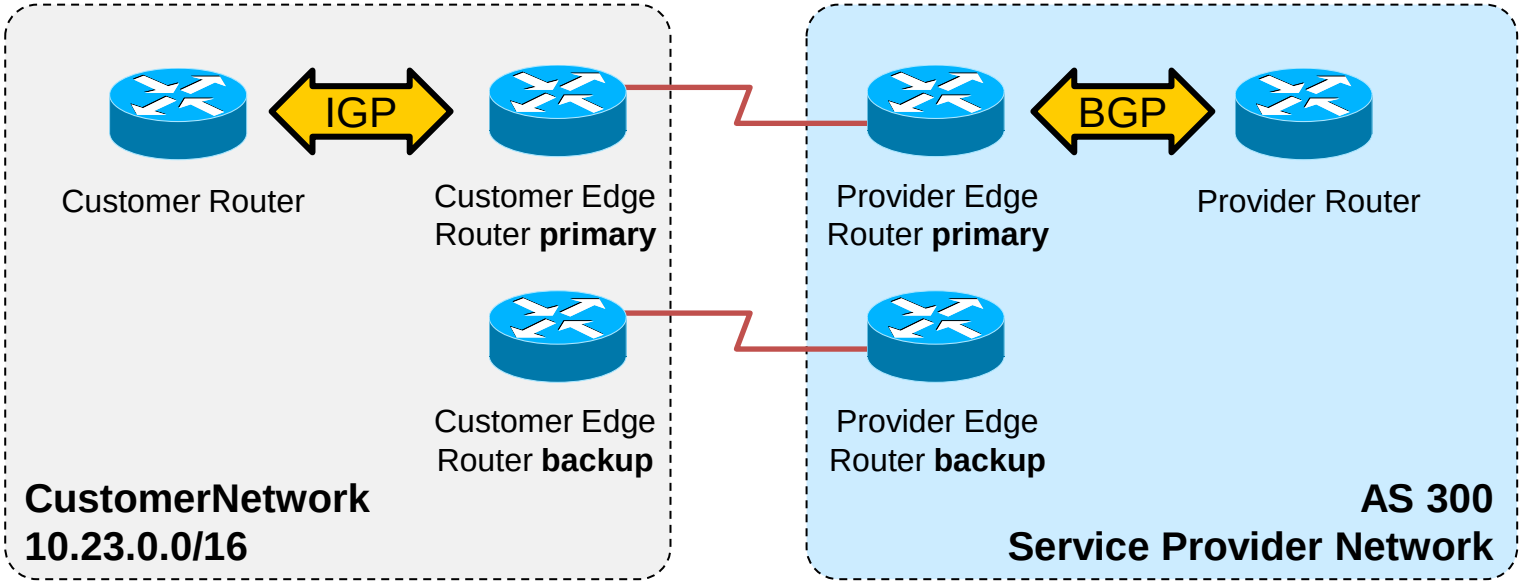
双线环境（主备）-运营商配置




```
ip route 10.23.0.0 255.255.0.0 x.x.x.x tag 1000
router bgp 300
  redistribute static route-map IntoBGP
```

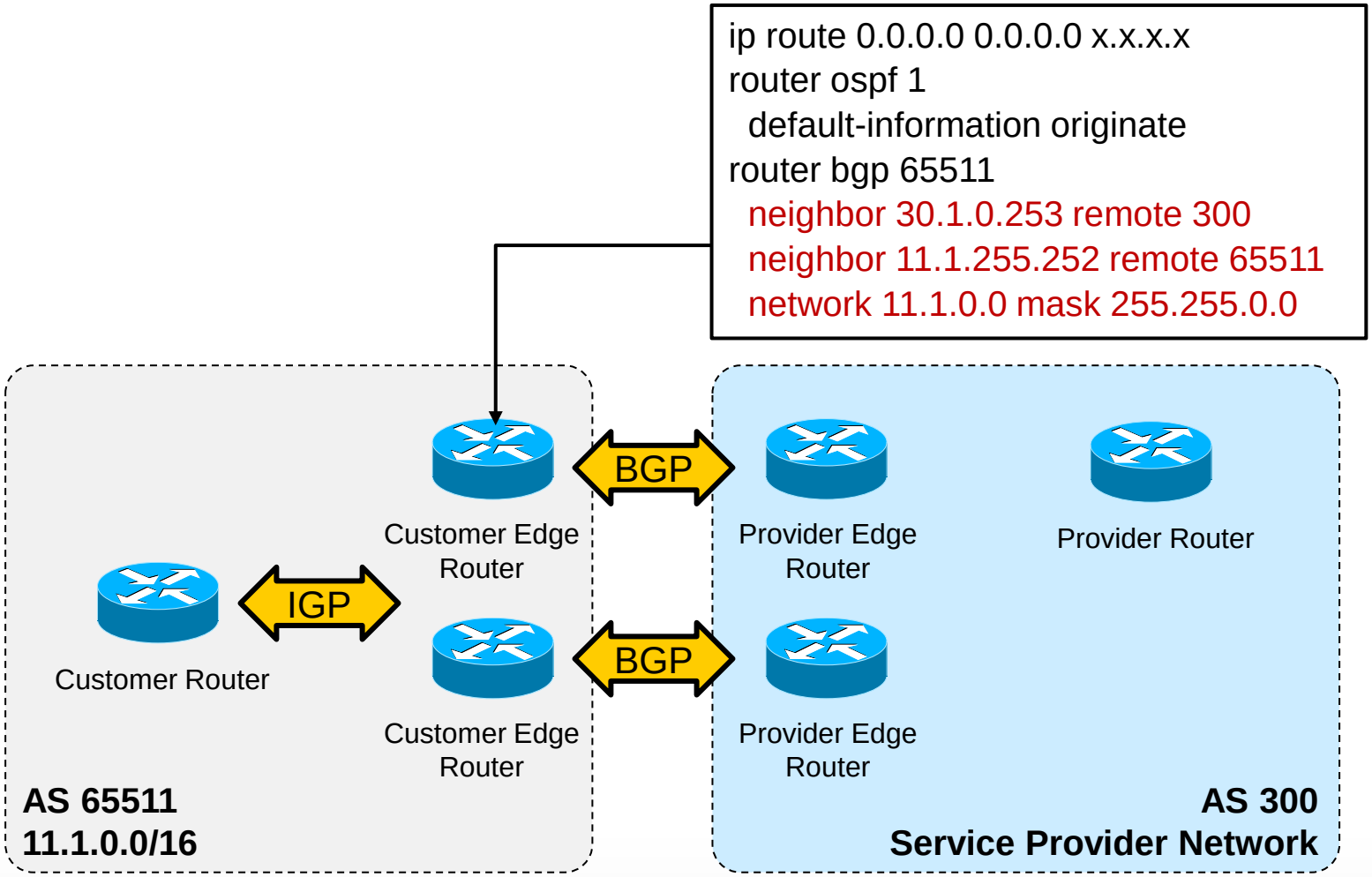
```
ip route 10.23.0.0 255.255.0.0 x.x.x.x tag 1000 250
router bgp 300
  redistribute static route-map IntoBGP
```

双线环境（主备）-运营商配置

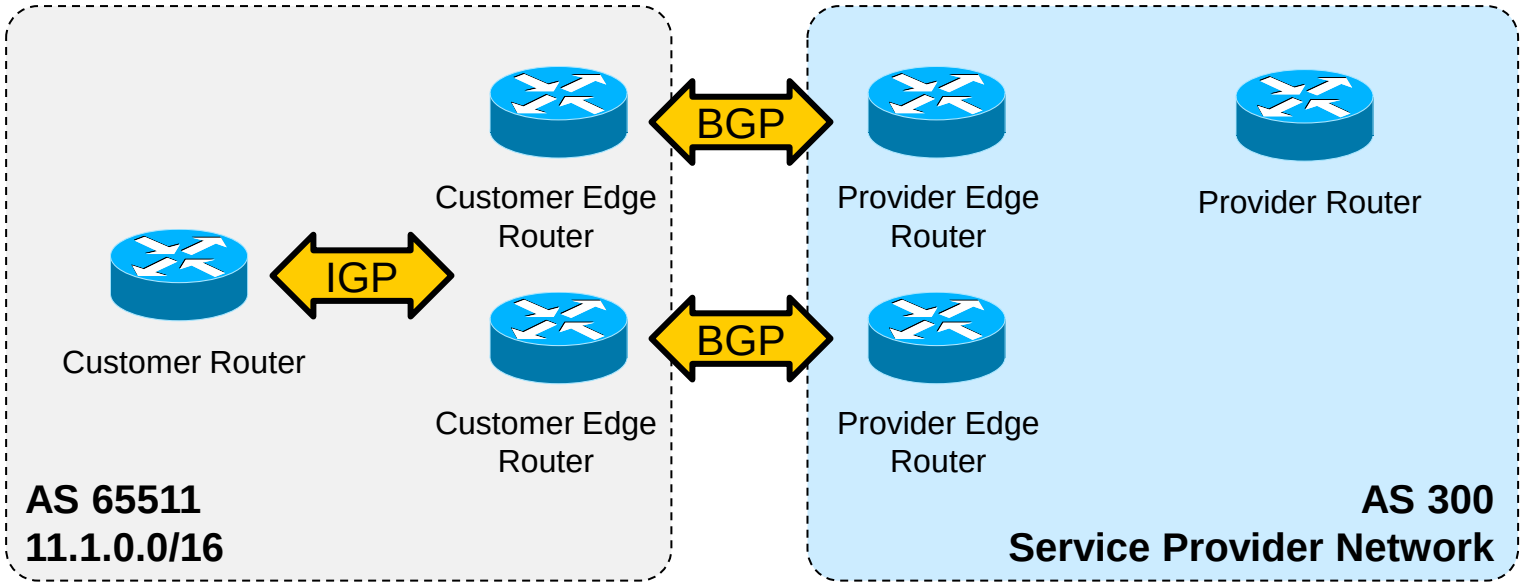


还有其他的解决方案么？
如使用weight，在backup路由器上，将路由的weight调低，并对Primary设备使用策略调高weight（in方向）
再如如LP等

双线环境 用BGP解决方案 客户的角度



双线环境 用BGP解决方案 运营商角度



- 可考虑向客户网络通告BGP的默认路由
- 只接受客户传递过来的被分配的合法前缀
- 只接受源于客户AS的路由
- 针对这些路由，在需要时，可选择no-export的community属性值，以防客户路由被传递给其他AS

双线环境 用BGP解决方案 运营商角度

- **BGP默认路由的产生**

方法一：静态默认路由network

Ip route 0.0.0.0 0.0.0.0 null0

然后在BGP进程中network 0.0.0.0即可将此默认路由注入BGP进程

使用该方法在BGP中引入的默认路由会被传递给所有BGP邻居

双线环境 用BGP解决方案 运营商角度

- **BGP默认路由的产生 cont.**

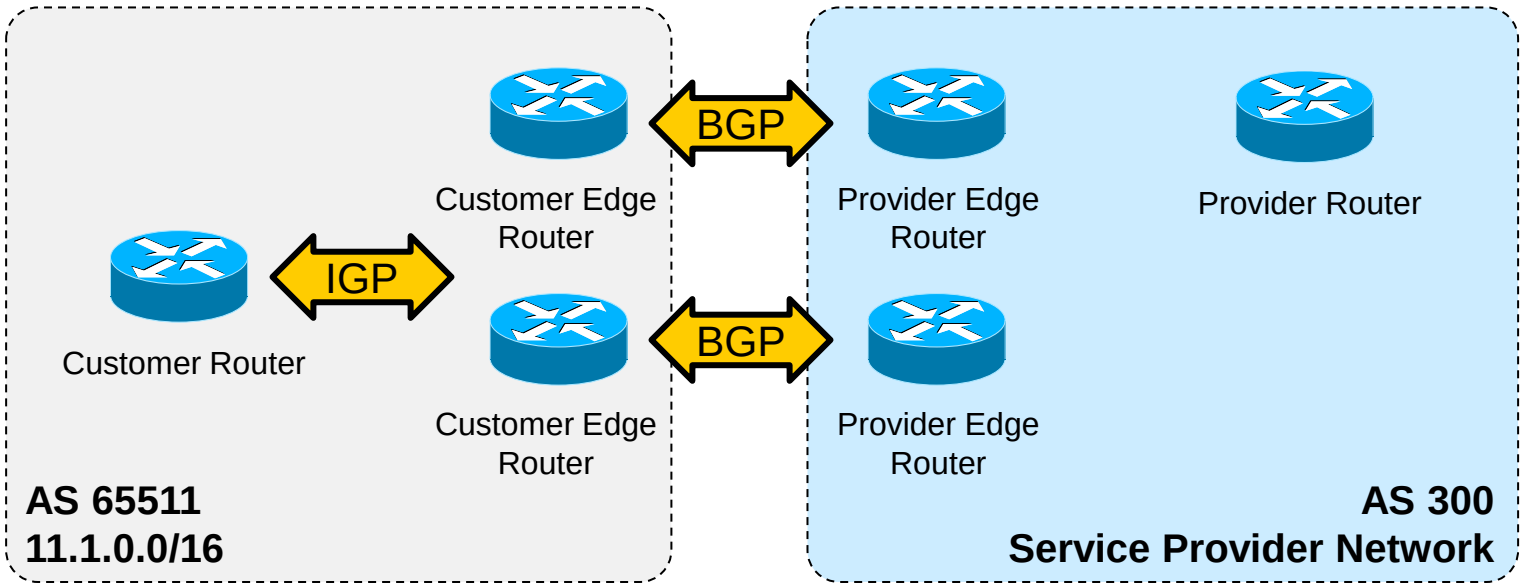
方法二：neighbor xxx default-originate

BGP进程中： neighbor xxx default-originate　向特定的邻居传递默认路由

此配置无需路由表中存在默认路由。有点类似OSPF的default-originate always

- 注意OUT方向的BGP Filters，是无法阻挡本地产生的默认路由传递给其他BGP邻居的，当然，在BGP邻居处，可使用IN方向的BGP Filters阻挡。

双线环境 用BGP解决方案 运营商角度

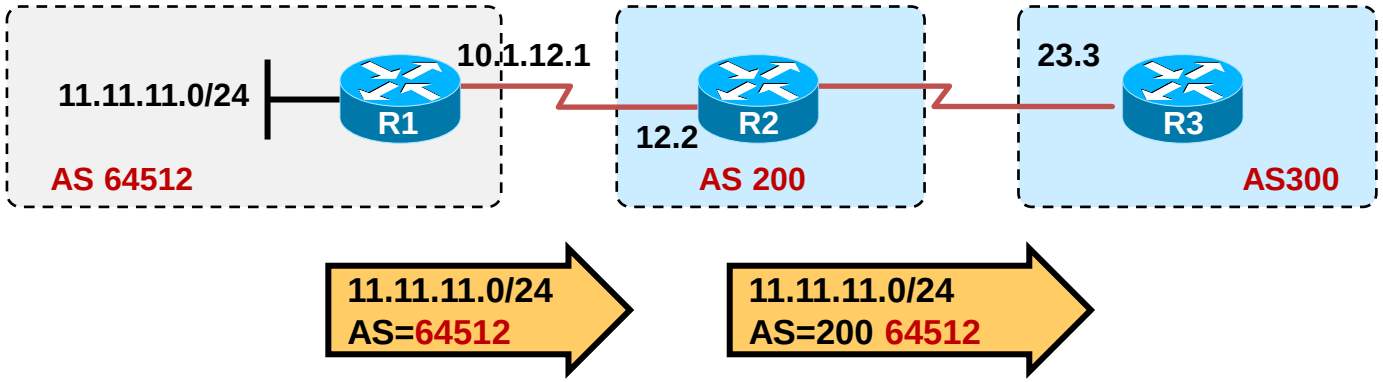



```
router bgp 300
  neighbor 30.1.0.252 remote 65511
  neighbor 30.1.0.252 default-originate
  neighbor 30.1.0.252 prefix-list DefaultOnly out
  neighbor 30.1.0.252 prefix-list Customer11 in
  neighbor 30.1.0.252 filter-list 11 in
  neighbor 30.1.0.252 route-map AllCustomerIn in
```

```
ip prefix-list DefaultOnly permit 0.0.0.0/0
ip prefix-list Customer11 permit 11.1.0.0/16 le 32
ip as-path access-list 11 permit ^65511(_65511)*$
route-map AllCustomerIn permit 10
  match ip address prefix-list Customer11
  set community no-export additive
route-map AllCustomerIn permit 9999
```

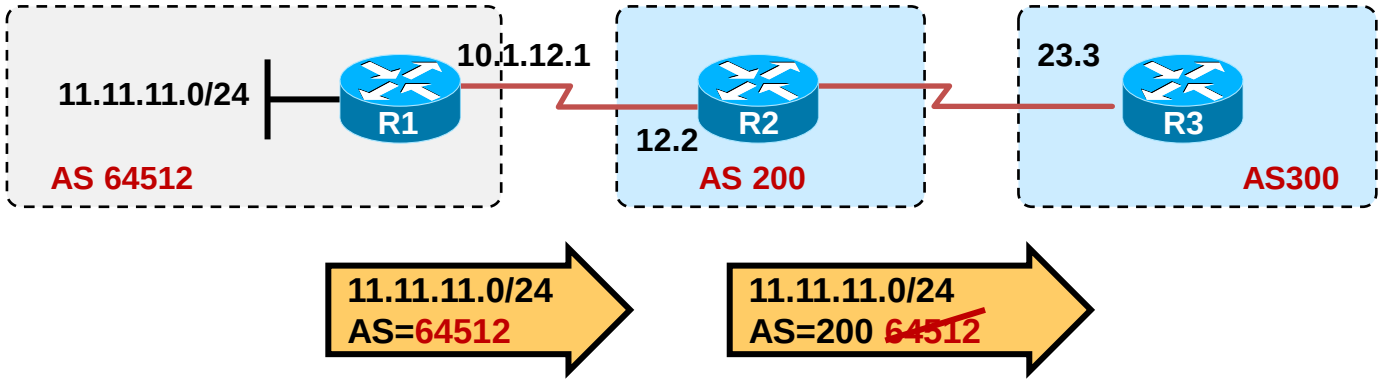
移除私有AS

私有AS



私有AS号被传递给公网AS

如何移除AS_PATH中的私有AS



为了防止私有AS号进入公网，需在边界设备上使用私有AS号的擦除特性：

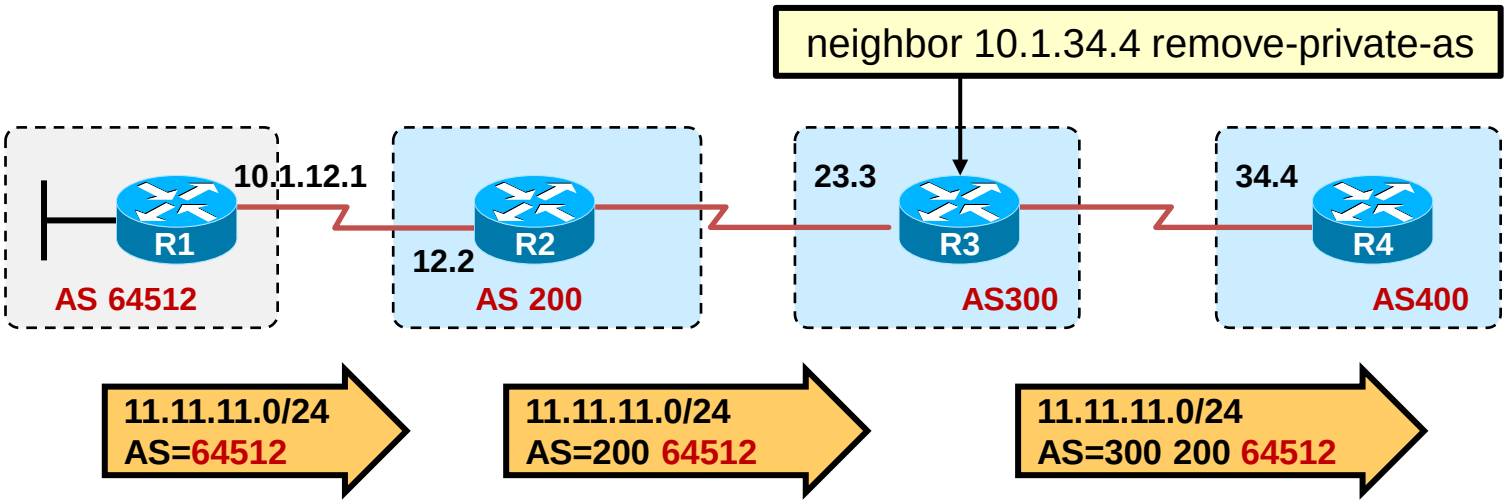
如，在R2上部署：

```
router bgp 200
  neighbor 10.1.23.3 remote 300
  neighbor 10.1.23.3 remove-private-as
```

neighbor remove-private-as命令注解

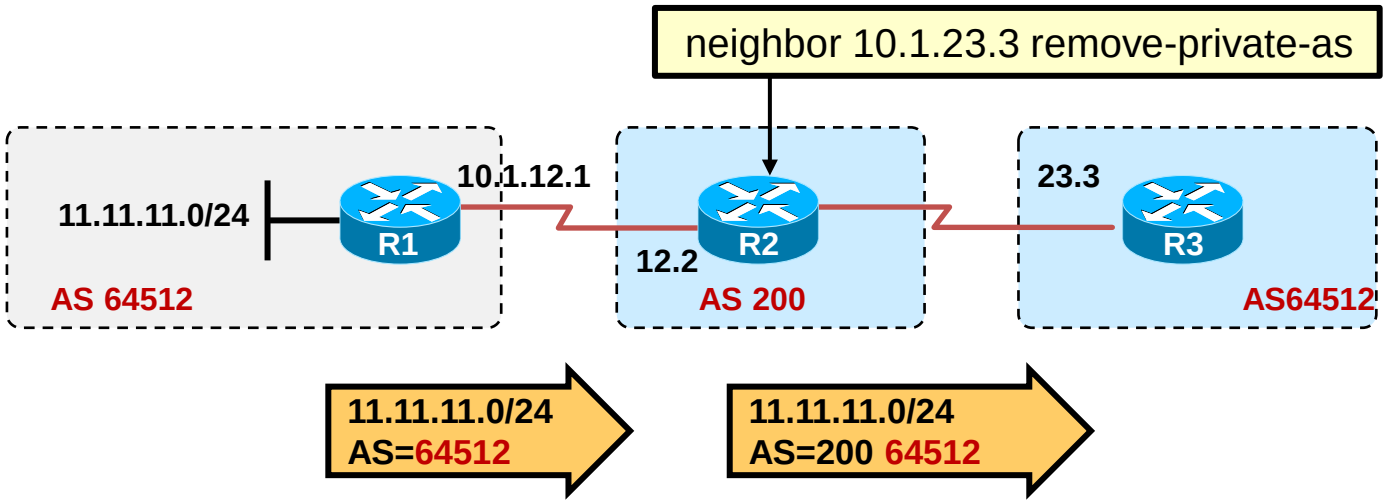
- neighbor remove-private-as只能用在eBGP邻居上
- 如果AS_PATH里只有私有AS号，则BGP移除这些AS号
- 如果AS_PATH里已经有了私有AS号，并且还有公有AS号，也就是两者并存的情况，那么BGP将不会移除私有AS号，这种情况视为一种配置错误
- If the AS_PATH contains the AS number of the eBGP neighbor, BGP does not remove the private AS number
- If the AS_PATH contains confederations, BGP removes the private AS numbers only if they come after the confederation portion of the AS_PATH.

neighbor remove-private-as命令注解



如果AS_PATH里已经有了私有AS号，并且还有公有AS号，也就是两者并存的情况，那么BGP将不会移除私有AS号，这种情况视为一种配置错误

neighbor remove-private-as命令注解



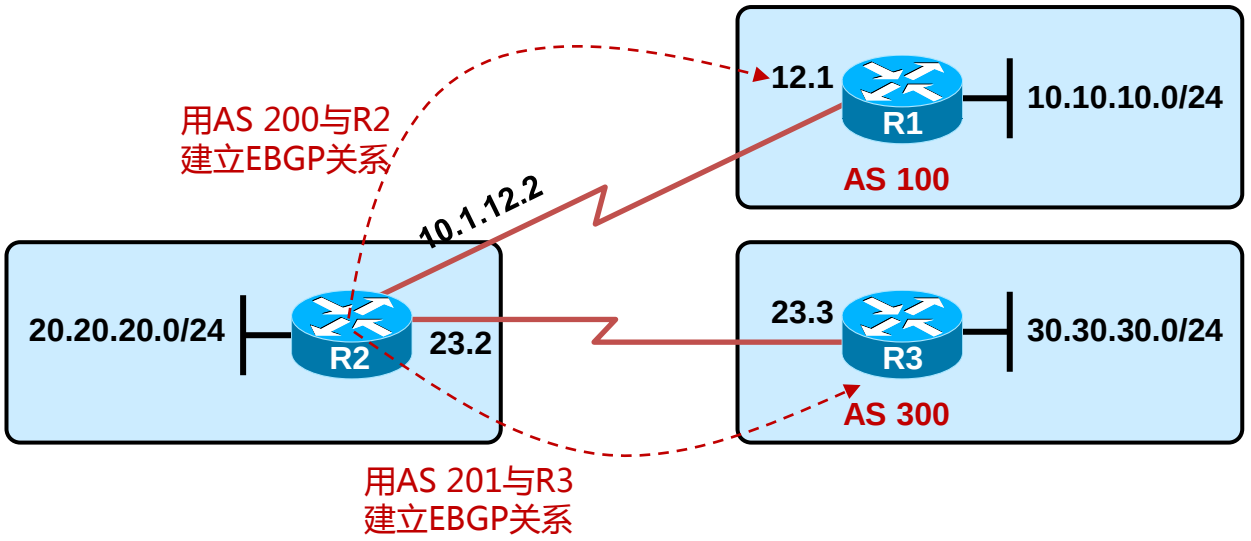
If the AS_PATH contains the AS number of the eBGP neighbor, BGP does not remove the private AS number

DUAL AS的实现

Dual AS

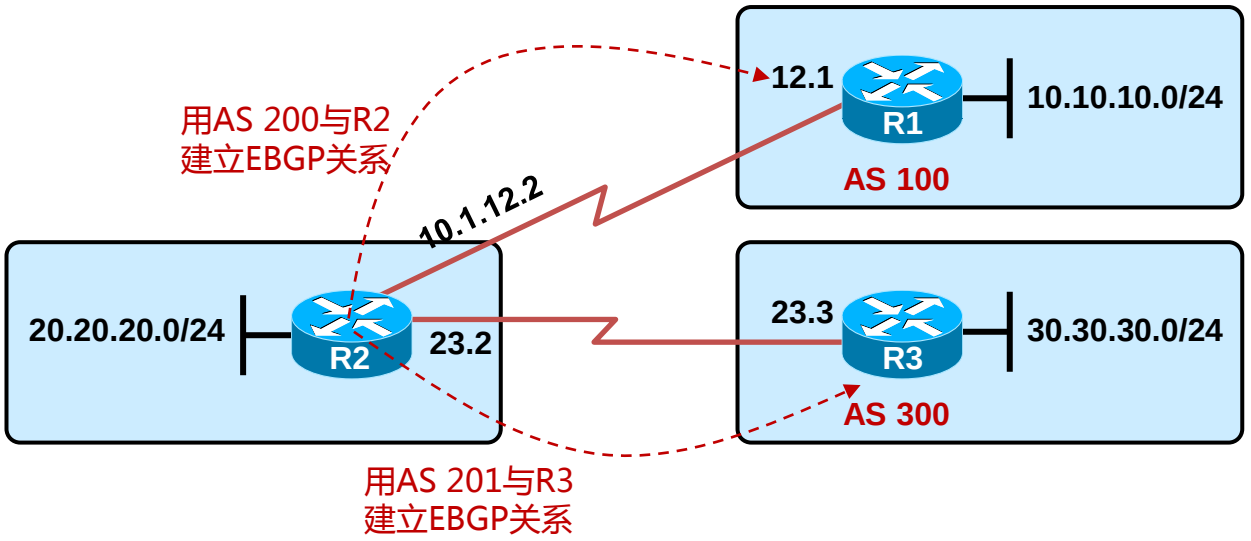
- 默认情况下，在单台Router上只能启动一个BGP进程，并且只能属于一个AS。DUAL AS允许我们在不中断现有BGP连接的情况下，在primary AS下同时运行一个secondary AS，从而提供一种网络迁移的机制。
- 在迁移期间，运行DUAL AS的路由器可以同时使用primary AS及secondary AS与外部AS建立EBGP连接，并且都能进行 BGP路由的更新和传递。
- 在不用断开现有连接的情况下，可缩短网络迁移的时间，且不用大批量的变更设备BGP配置

Dual AS的实现



AS100为R2（AS200）的主运营商，R1、R2之间的BGP也一直在运行着，此时，如果增加了一个运营商R3（AS300），并且该运营商分配给R2的AS号为201，那么传统的配置，R2肯定是要断开与R1的连接并且重新配置BGP，但 DUAL AS允许我们同时在R2上运行两个BGP AS，用AS200与R1形成邻居关系，AS201与R3邻接，从而实现快速的过渡。

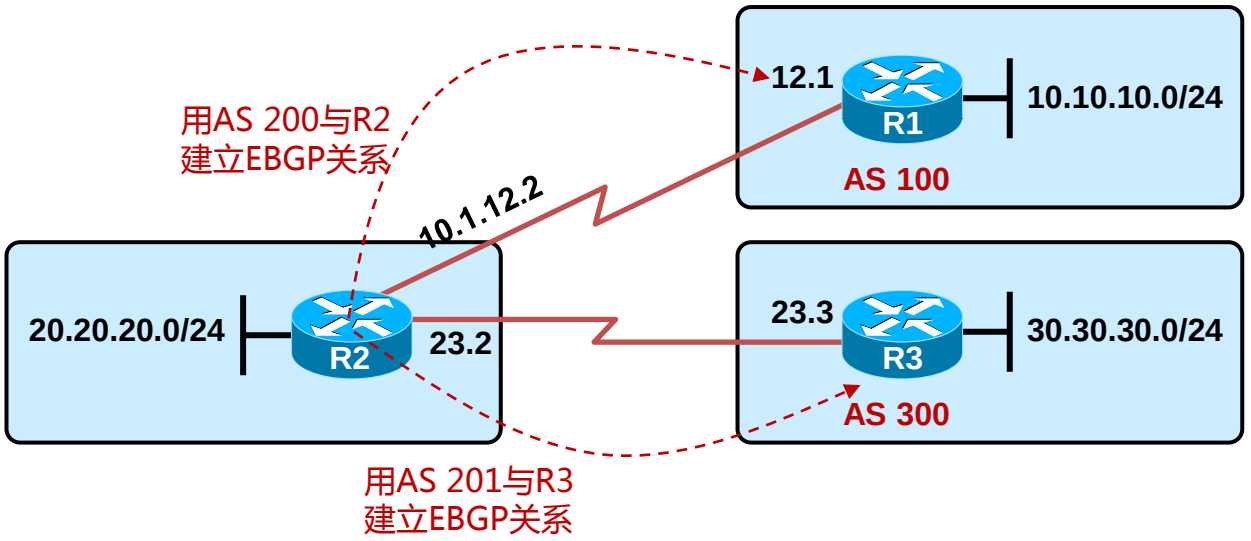
Dual AS的实现



R2上的配置如下：

```
router bgp 200
 no synchronization
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 neighbor 10.1.23.3 local-as 201 no-prepend replace-as dual-as
```

Dual AS的实现



注意事项：

- R2 router bgp 200，配置的BGP进程AS号为200，这是primary as，这个AS用于R1建立EBGP邻居关系，并可正常传递路由
- R2 的AS号201为secondaryAS，用于R3建立EBGP邻居关系，并可正常传递路由。R3上指neighbor时，R2的AS号为201

Dual AS命令详解

R1上的配置如下：

```
router bgp 200
neighbor 10.1.23.3 local-as 201 no-prepend replace-as dual-as
```

设置secondary AS

Do not prepend local-as to updates from ebgp peers
向Primary AS的EBGP邻居通告路由时，不附加secondary AS号

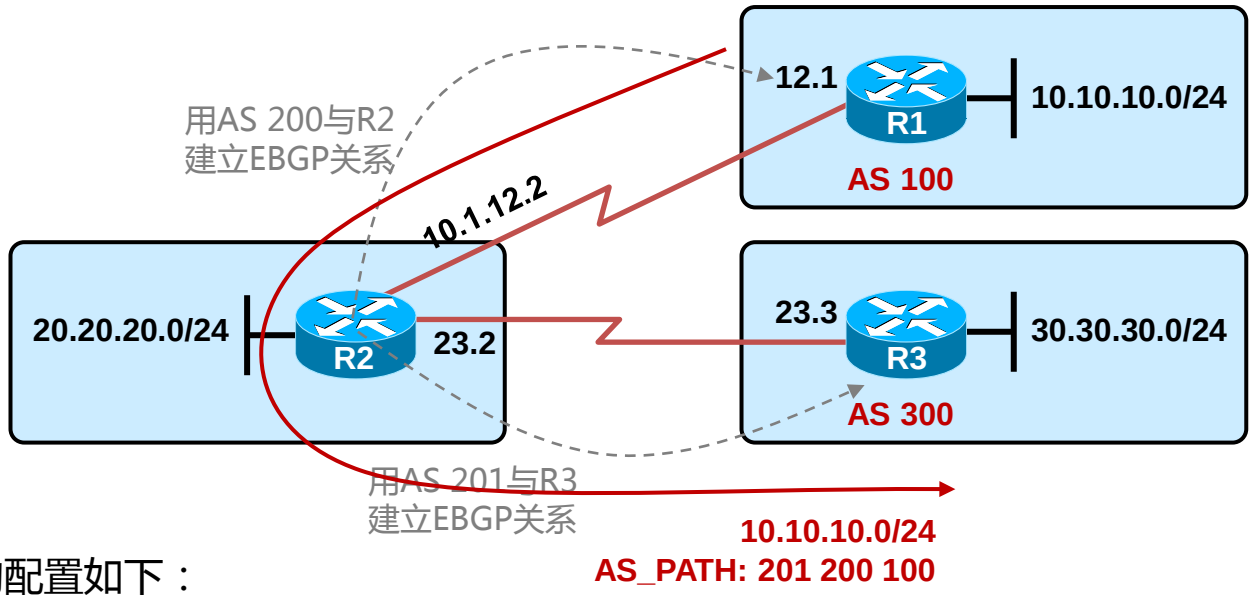
Replace real AS with local AS in the EBGP updates
当路由器向secondaryAS的EBGP邻居发送更新时，用secondaryAS号替代Pri AS号

Accept either real AS or local AS from the ebgp peer
EBGP对等体既可以使用PriAS也可以使用SecAS对本地指remote-as

Dual AS命令详解

- The replace-as keyword is used to prepend only the local autonomous-system number (as configured with the ip-address argument) to the AS_PATH attribute. The autonomous-system number from the local BGP routing process is not prepended.
- The dual-as keyword is used to configure the eBGP neighbor to establish a peering session using the real autonomous-system number (from the local BGP routing process) or by using the autonomous-system number configured with the ip-address argument (local-as).
- The example configures the peering session with the 10.0.0.1 neighbor to accept the real autonomous system number and the local-as number.

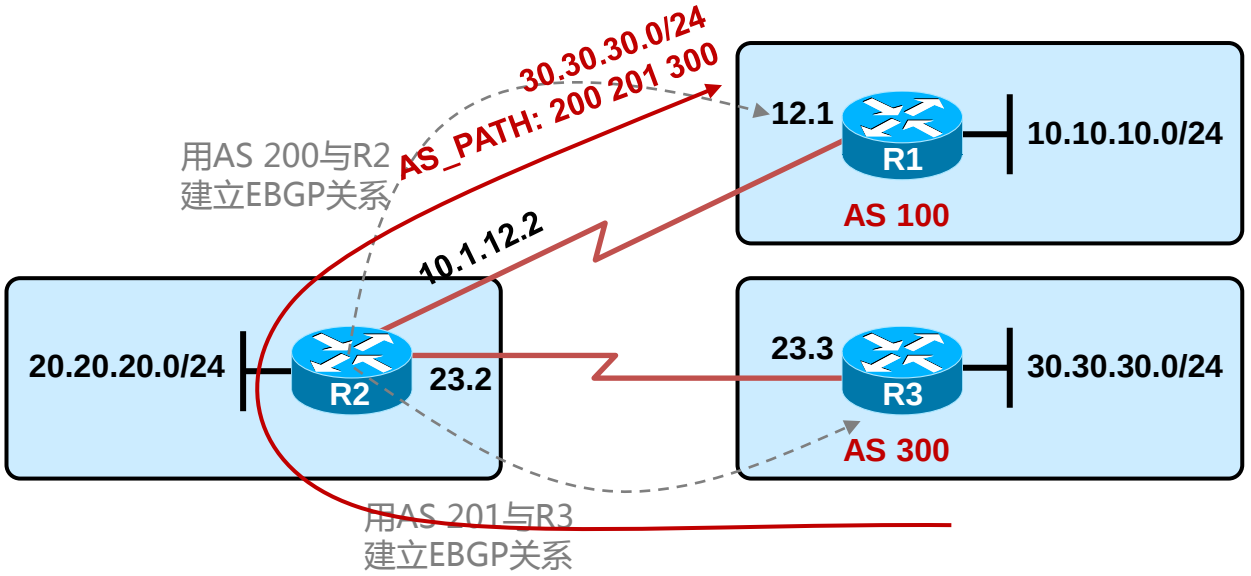
Dual AS配置示例 1



R1上的配置如下：

```
router bgp 200
 no synchronization
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 neighbor 10.1.23.3 local-as 201
```

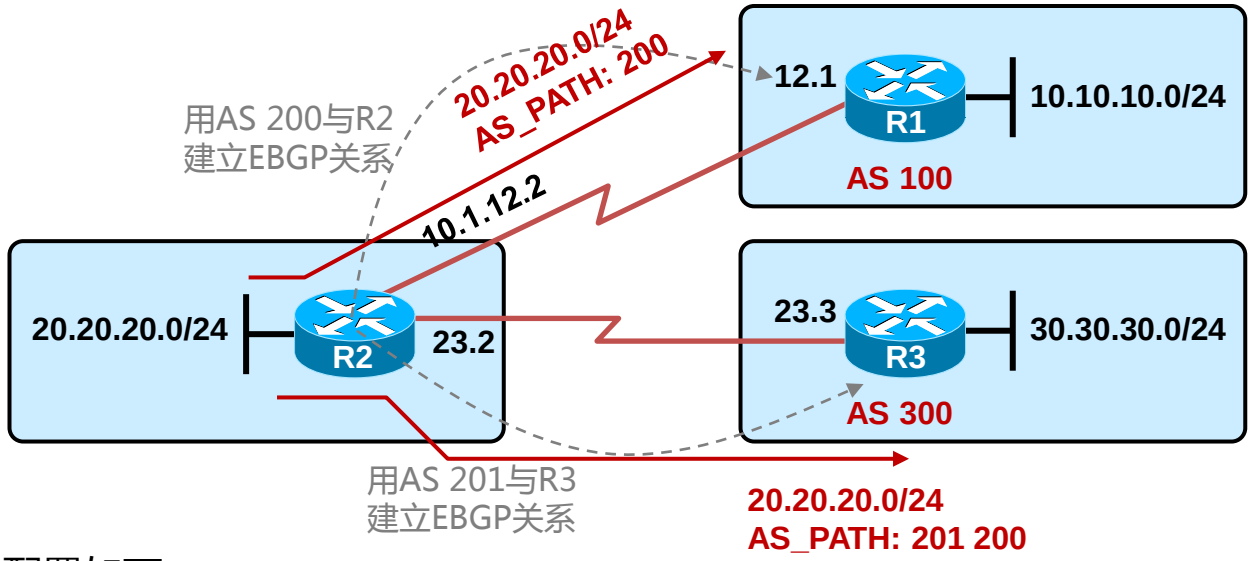
Dual AS配置示例 1 (cont.)



R1上的配置如下：

```
router bgp 200
 no synchronization
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 neighbor 10.1.23.3 local-as 201
```

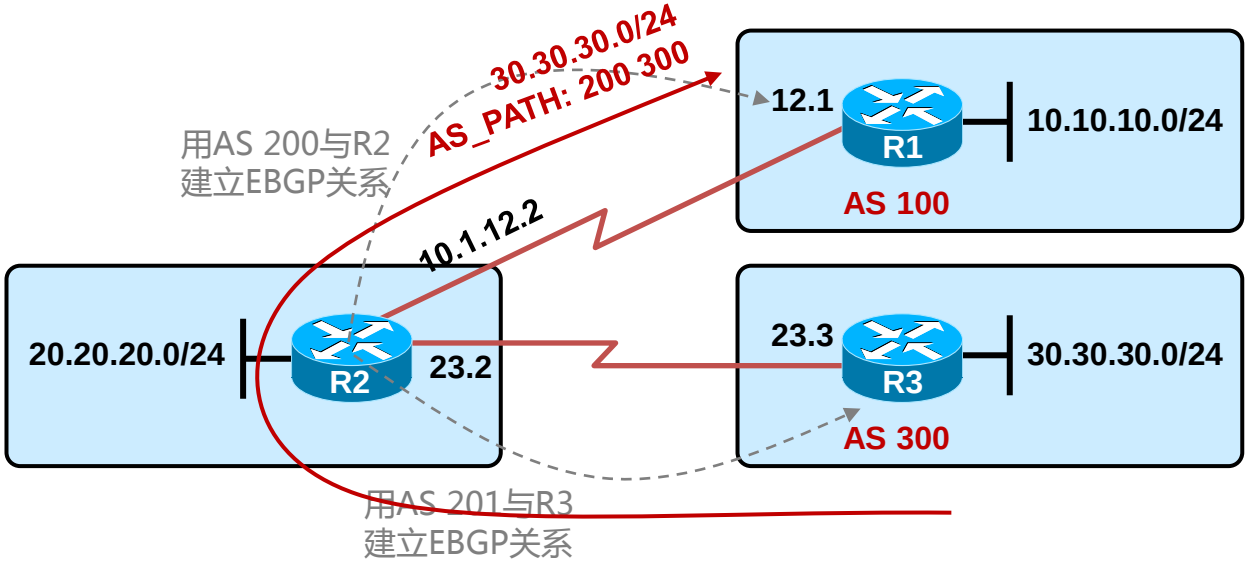
Dual AS配置示例 1 (cont.)



R1上的配置如下：

```
router bgp 200
 no synchronization
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 network 20.20.20.0 mask 255.255.255.0
 neighbor 10.1.23.3 local-as 201
```

Dual AS配置示例 2

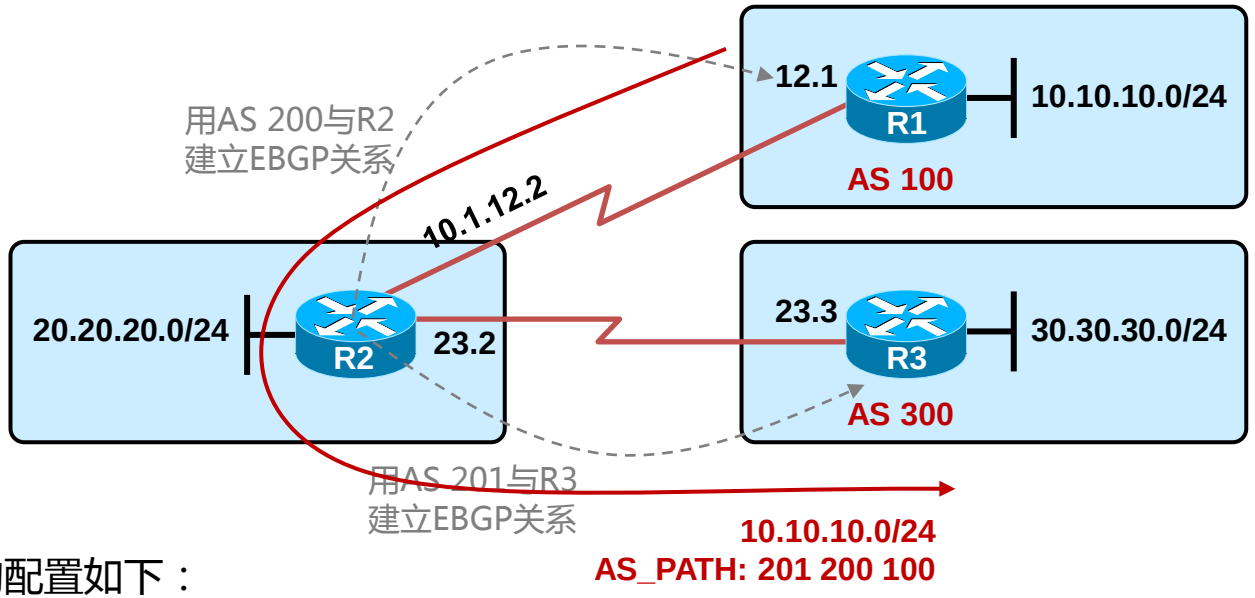


R1上的配置如下：

```
router bgp 200
 no synchronization
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 neighbor 10.1.23.3 local-as 201 no-prepend
```

注意这里的变化：R2将30.0的路由传递给R1时，不再插入seconday AS号

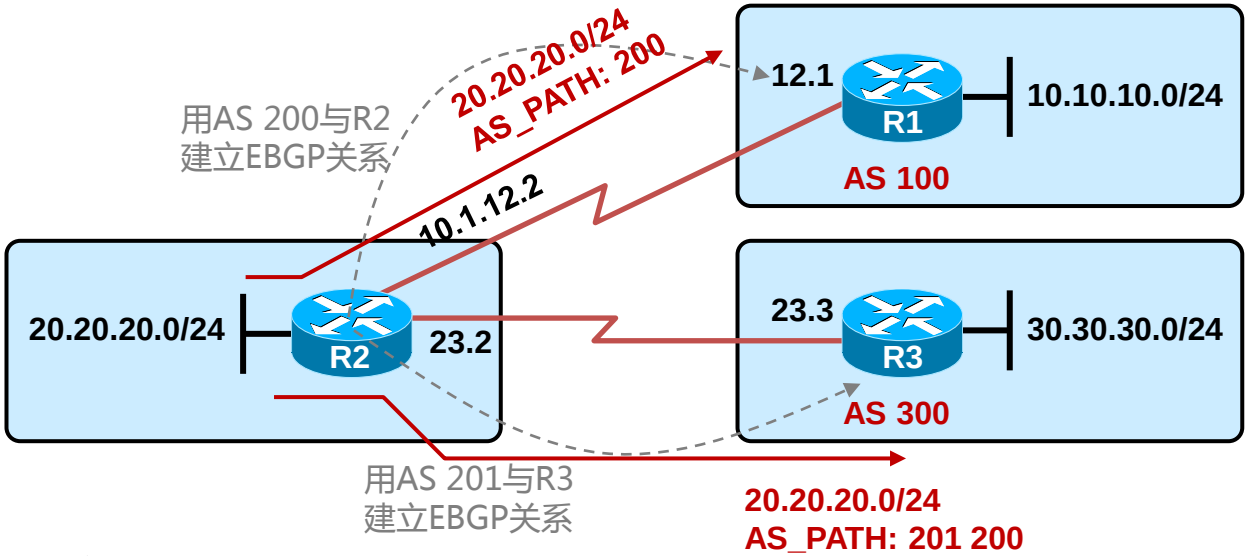
Dual AS配置示例 2 (cont.)



R1上的配置如下：

```
router bgp 200
 no synchronization
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 neighbor 10.1.23.3 local-as 201 no-prepend
```

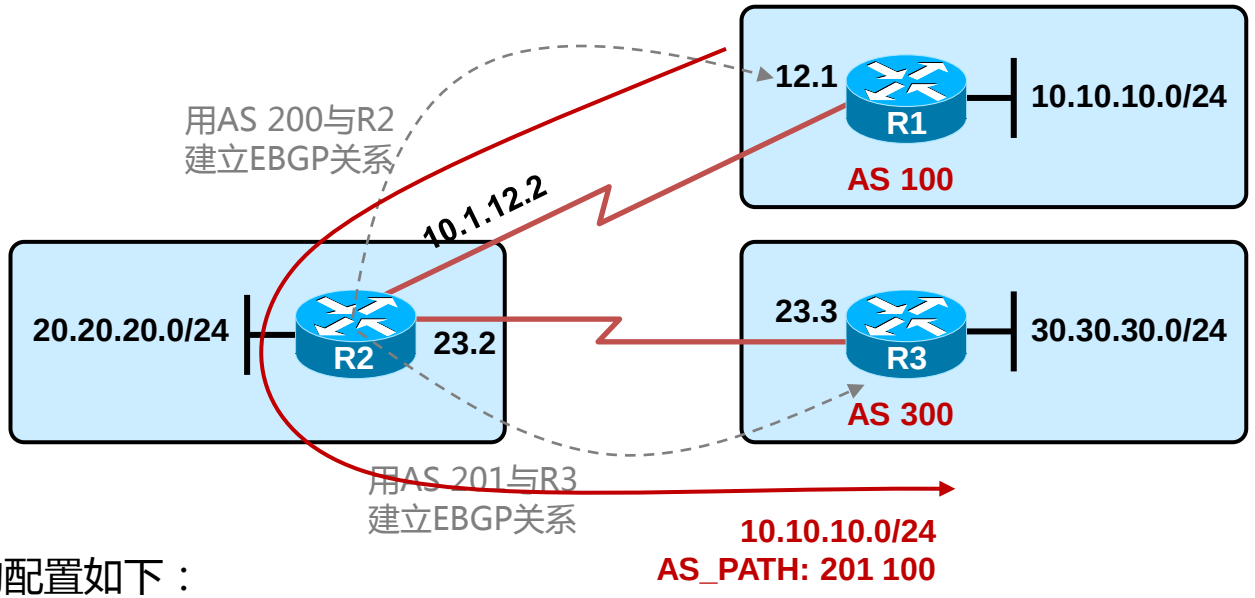
Dual AS配置示例 2 (cont.)



R1上的配置如下：

```
router bgp 200
 no synchronization
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 network 20.20.20.0 mask 255.255.255.0
 neighbor 10.1.23.3 local-as 201 no-prepend
```

Dual AS配置示例 3

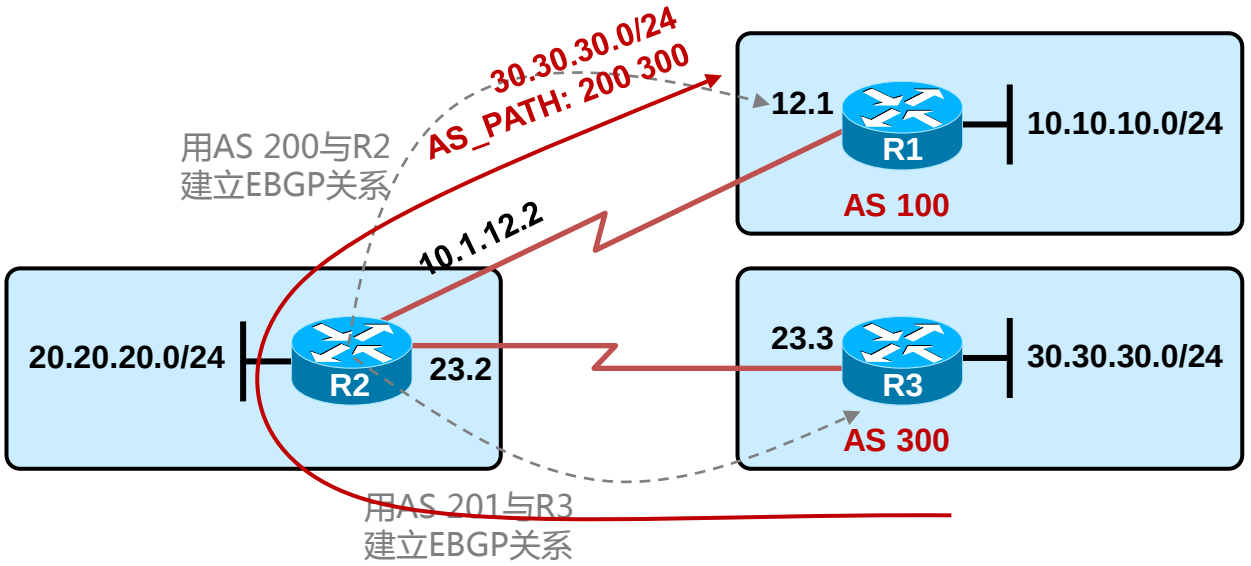


R1上的配置如下：

```
router bgp 200
 no synchronization
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 neighbor 10.1.23.3 local-as 201 no-prepend replace-as
```

注意变化，replace-as 关键字让R2向secAS的EBGP邻居发送路由时，用local-as201替代真实AS200

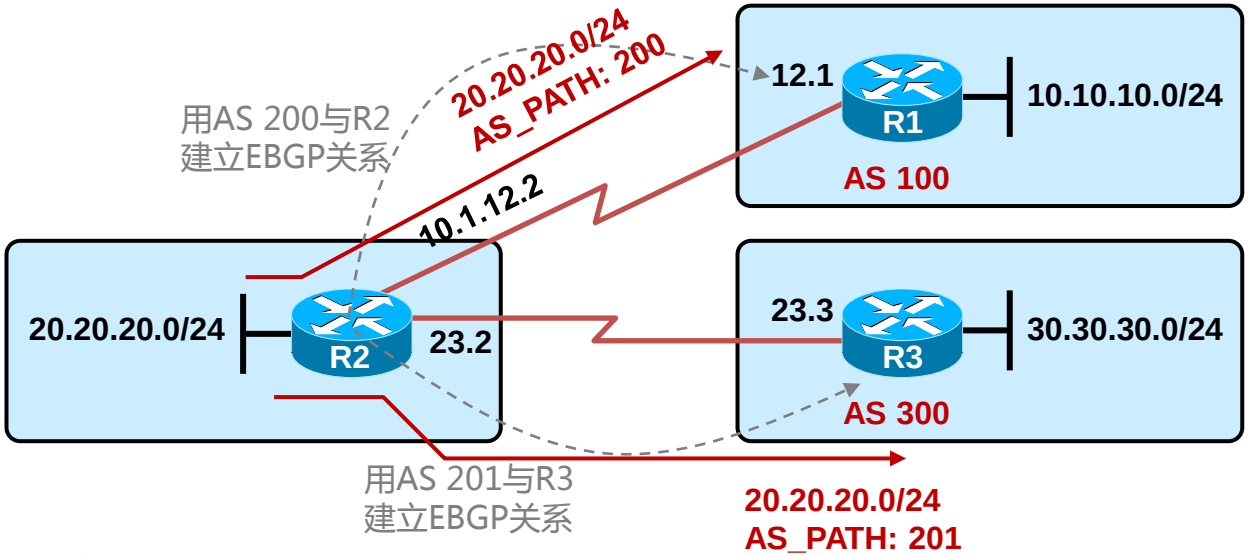
Dual AS配置示例 3 (cont.)



R1上的配置如下：

```
router bgp 200
 no synchronization
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 neighbor 10.1.23.3 local-as 201 no-prepend replace-as
```

Dual AS配置示例 3 (cont.)

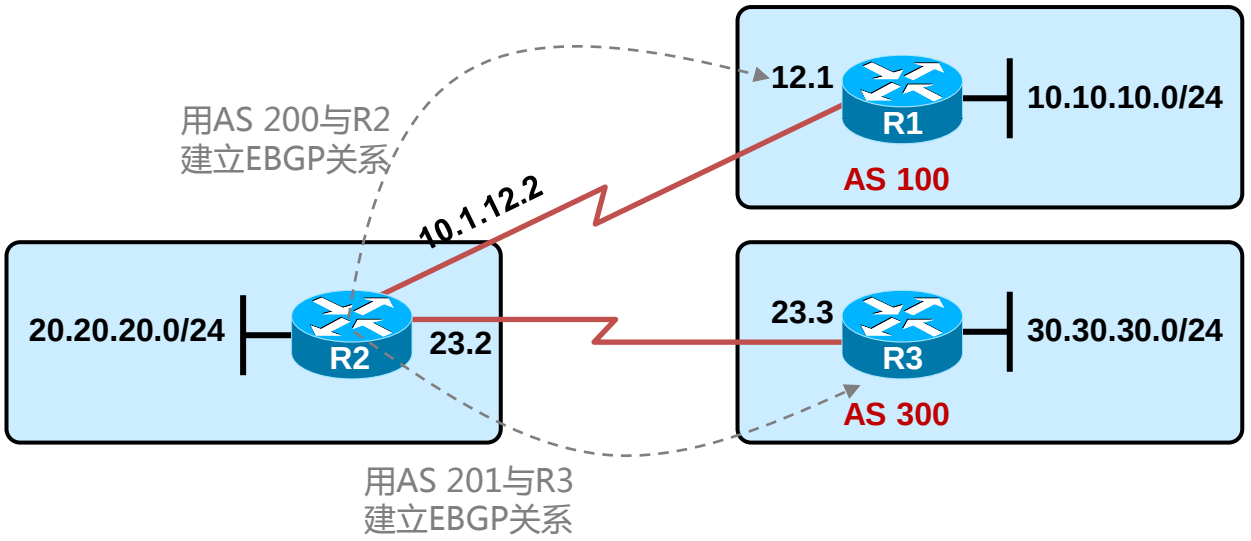


R1上的配置如下：

```
router bgp 200
 no synchronization
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 neighbor 10.1.23.3 local-as 201 no-prepend replace-as
 network 20.20.20.0 mask 255.255.255.0
```

注意变化，replace-as 关键字让R2向secAS的EBGP邻居发送路由时，用local-as201替代真实AS200

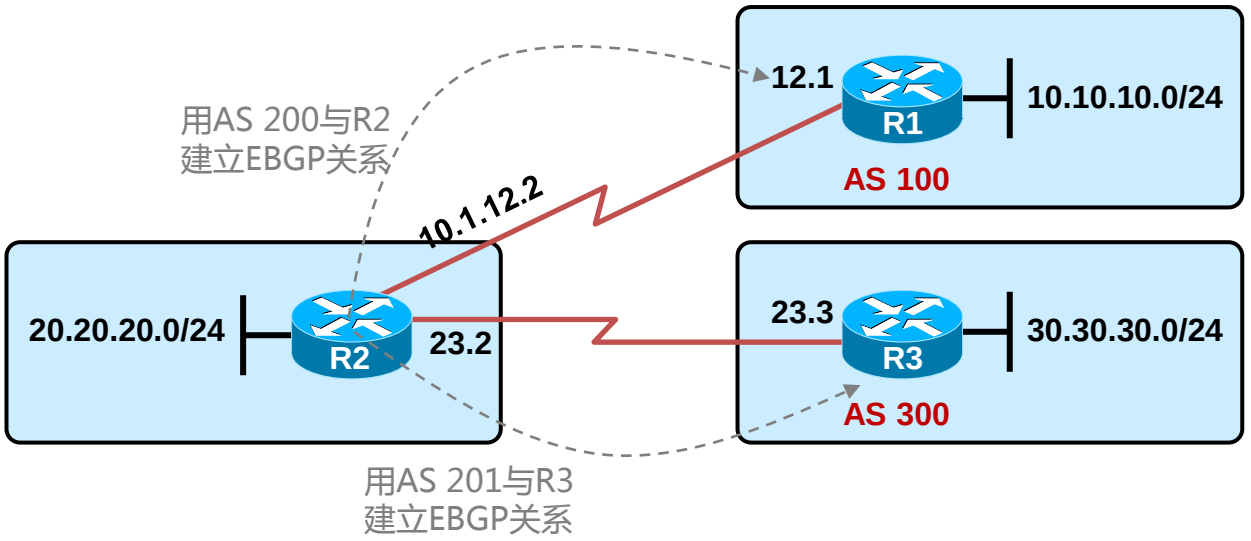
Dual AS配置示例 4



R1上的配置如下：

```
router bgp 200
 no synchronization
 neighbor 10.1.12.1 remote-as 100
 neighbor 10.1.23.3 remote-as 300
 neighbor 10.1.23.3 local-as 201 no-prepend replace-as dual-as
```

Dual AS配置示例



R3上的配置如下：

```
router bgp 300
 no synchronization
 neighbor 10.1.23.2 remote-as 200 或 neighbor 10.1.23.2 remote-as 201
 network 30.30.30.0 mask 255.255.255.0
```

在R2使用**dual-as** 关键字之前，R3只能将R2配置为remote-as 201，当R2配置了**dual-as**后，R2接受其EBGP邻居使用remote-as 200或201与其建立邻居关系

红茶三杯
Vinsoney

学习 沉淀 成长 分享

关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

IPv6基础

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2013-06-18

课程目标

IPv6概述

IPv6编址

IPv6基本配置

红茶三杯
Vinsoney

IPv6概述

IPv4存在的问题

- 地址耗尽
- Internet用户快速增长
- Internet路由表增大
- 缺乏真正的端到端模型
- 无法适应新技术的发展，如物联网等
- 所有行业都是IPv6的潜在用户
- ……

IPv6特点

- 128bits的地址方案，为未来数十年提供了巨大的IP地址空间
- 多等级层次有助于路由聚合，提高了因特网网络路由的效率及可扩展性
- 自动配置过程允许IPv6网络中的节点更加便捷的接入IPv6网络
- 重新编址机制使得IPv6提供商之间的转换对最终用户是透明的
- 无需NAT
- 不再有广播、不再有ARP
- IPv6的包头比IPv6更有效率，数据字段更少，去掉了包头校验和。更简单的报头提高了路由器的处理效率。新的扩展包头替代了IPv4的选项字段，并且提供了更多的灵活性
- 更有效的支持移动性和安全性
- v4v6过渡方式丰富多彩

IPv6分组包头

IPv4 Header

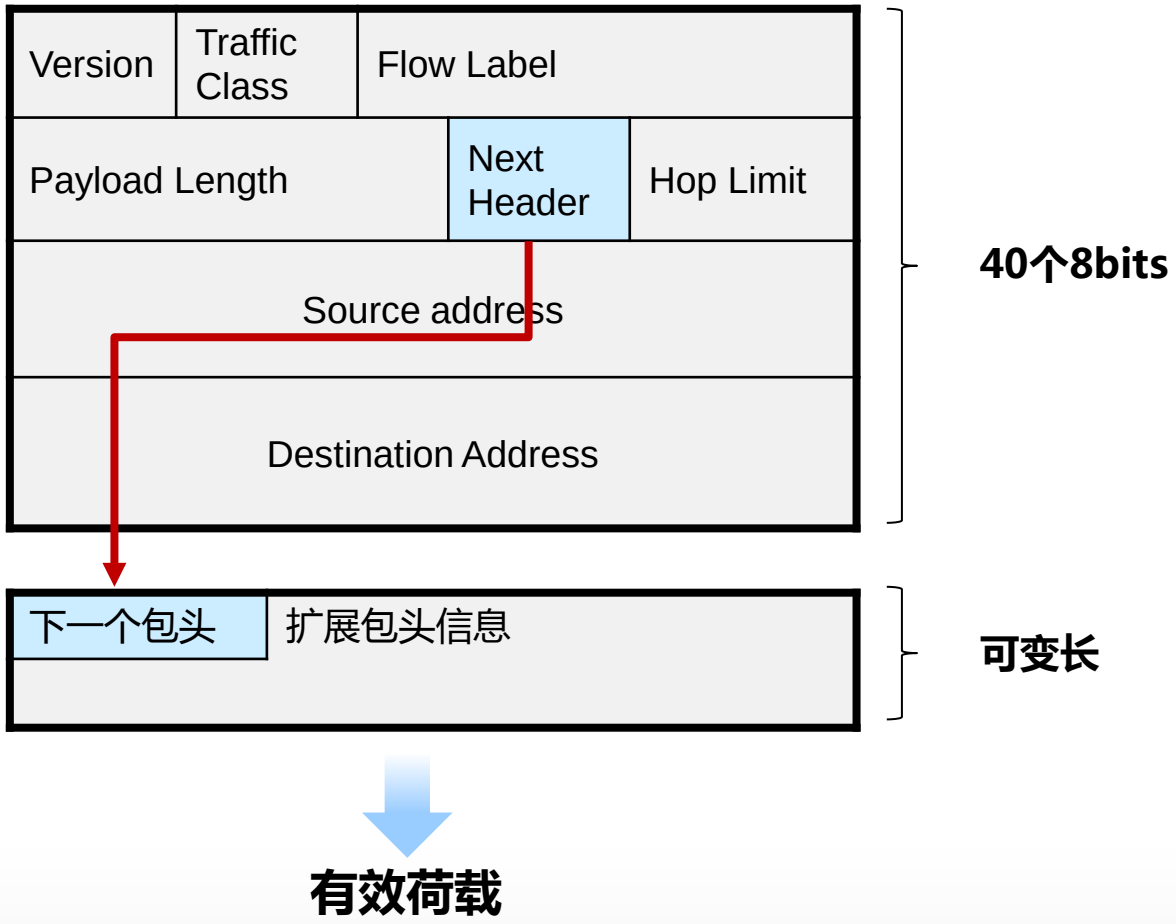
Version	IHL	ToS	Total Len
Identification		Flags	Fragment Offset
TTL	Protocol	Head Checksum	
Source address			
Destination Address			
Options			padding

IPv6 Header

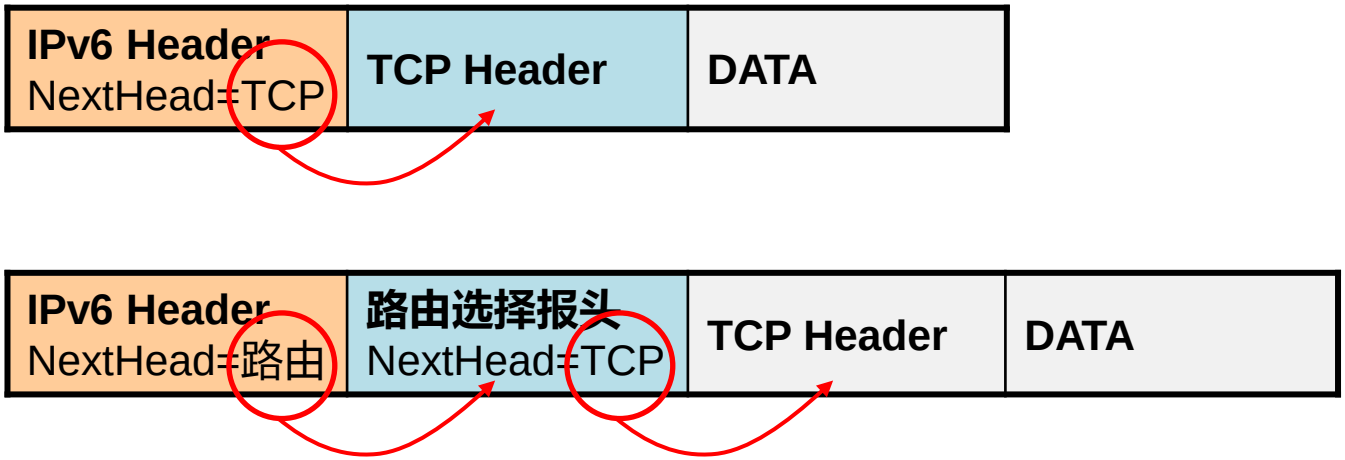
Version	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source address			
Destination Address			

保留的字段　取消的字段　名字位置变化　新增字段

IPv6分组包头 及 扩展包头



IPv6分组包头 及 扩展包头



- 扩展报头只有目标节点查看，其他节点不查看和处理大部分扩展报头
- 要按顺序查看扩展报头的内容

IPv6扩展包头

- **使用扩展报头时，报头顺序如下（RFC2460）：**
  1. IPv6报头
  2. 逐跳选项报头：所有路由器都要对其处理
  3. 目标选项报头：使用了路由选择报头
  4. 路由选择报头：列出了一个或多个中间点
  5. 分段报头
  6. 身份验证报头（AH）和封装安全有效负载（ESP）报头
  7. 上层报头：主要为TCP和UDP等

红茶三杯
Vinsoney

IPv6编址

IPv6编址

- IPv6地址为128位
- 可提供给PC、无线IP电话、机顶盒、视频设备、安保监控设备等等，海量地址空间
- 使用冒号分隔十六进制格式表示

  2001：0da8：0207：0000：0000：0000：0000：8207
- IPv6地址有多种呈现方式

IPv6地址简写方式

- IPv6地址在简写的时候，每组16bits的单元中多个前导0可以省略成一个0：
  - 如：2001：00a8：0207：0000：0000：0000：0000：8207
  - 可缩写成：2001：a8：207：0：0：0：0：8207
- 一个或多个连续的16比特字段为0时，可用：：表示，但整个缩写中只允许有一个：：
  - 如上面的例子，可以进一步缩写成：2001：a8：207：：8207

IPv6编址 简化书写示例

压缩前： 0000:0000:0000:0000:0000:0000:0000:0001

压缩后：::1

压缩前： 2001:0410:0000:0000:FB00:1400:5000:45FF

压缩后： 2001:410 :: FB00:1400:5000:45FF

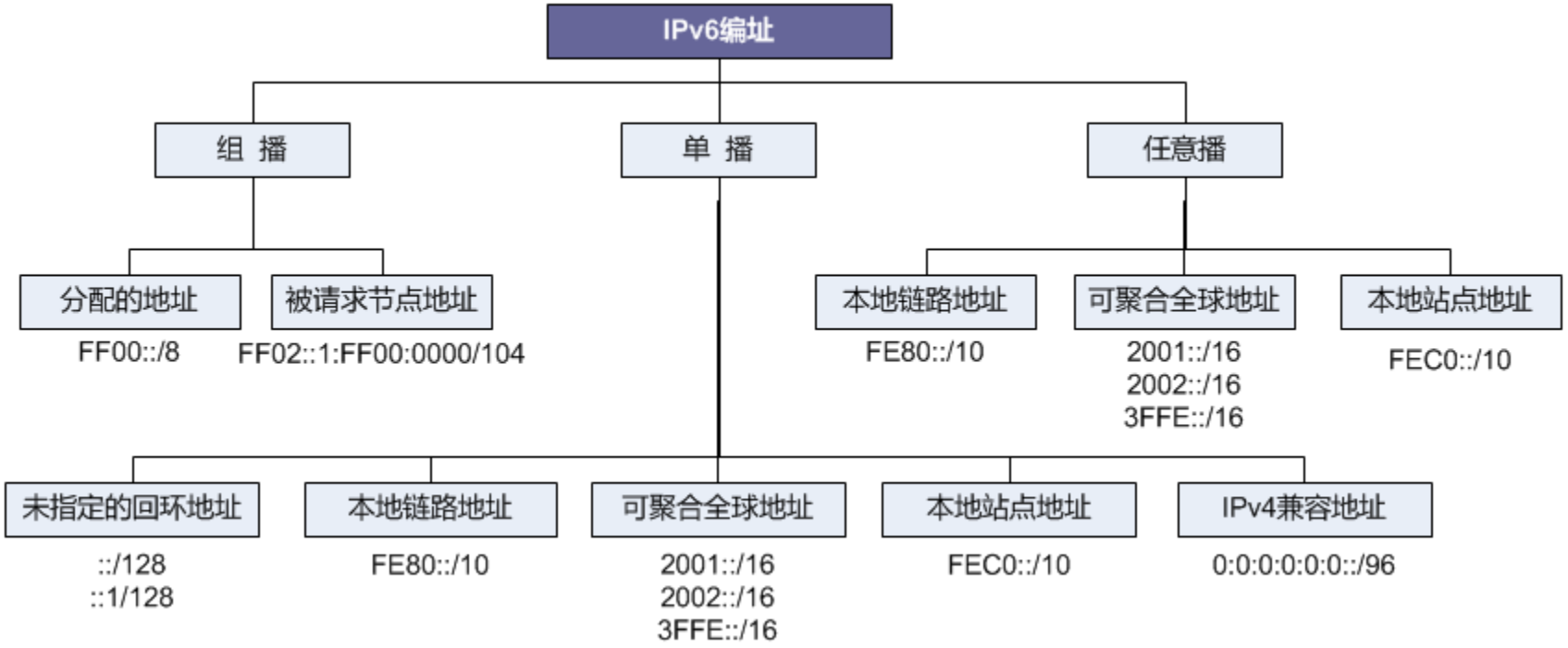
压缩前： 2001:0410:0000:1234:FB00:1400:5000:45FF

压缩后： 2001:410::1234:FB00:1400:5000:45FF

压缩前： 3ffe:0000:0000:0000:1010:2a2a:0000:0001

压缩后： 3ffe::1010:2a2a:0:1

IPv6地址空间



IPv6地址类型

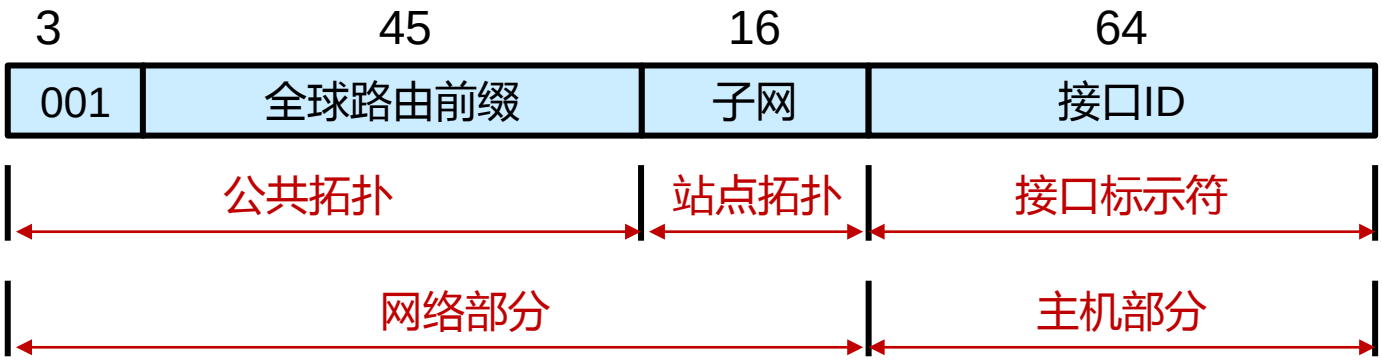
- **单播地址（Unicast Address）**
  - 标识一个接口，目的地址为单播地址的报文会被送到被标识的接口
- **组播地址（Multicast Address）**
  - 标识多个接口，目的地址为组播地址的报文会被送到被标识的所有接口
- **任播地址（Anycast Address）**
  - 标识多个接口，目的为任播地址的报文会被送到最近的一个被标识接口，最近节点是由路由协议来定义的
- **IPv6没有定义广播地址**

单播地址（Unicast）

- 可聚合全球单播地址
- 链路本地单播地址
- 站点本地单播地址
- ……

单播地址 - 可聚合全局单播地址

- **相当于IPv4全局单播地址**
- **由48位的全局路由选择前缀+16位的子网ID+64位的接口ID组成**



一般从运营商处申请到的IPv6地址空间为/48，再由自己根据需要进一步规划

单播地址 - 可聚合全局单播地址

- **可聚合全球单播地址的范围：**
  - 2000:0000:0000:0000:0000:0000:0000:0000 到
  - 3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
  - 由此看出，可聚合全球单播地址占IPv6总地址空间的8分之1。

单播地址 - 可聚合全局单播地址

- **可聚合全球单播地址的范围：**

Unicast Global [001]		
2001::/16	0010 0000 0000 0001	IPv6 InternetARIN, APNIC, RIPE NCC, LACNIC
2002::/16	0010 0000 0000 0010	6 to 4 transition mechanisms
2003::/16	0010 0000 0000 0011	IPv6 InternetRIPE NCC
2400:0000::/19 2400:2000::/19 2400:4000::/21	0010 0100 0000 0000	IPv6 InternetAPNIC
2600:0000::/22 2604:0000::/22 2608:0000::/22 260C:0000::/22	0010 0110 0000 0000 0010 0110 0000 0100 0010 0110 0000 1000 0010 0110 0000 1100	IPv6 InternetARIN
2A00:0000::/21 2A01:0000::/23	0010 1010 0000 0000 0010 1010 0000 0001	IPv6 InternetRIPE NCC
3FFE::/16		6bone

单播地址 - SiteLocal address

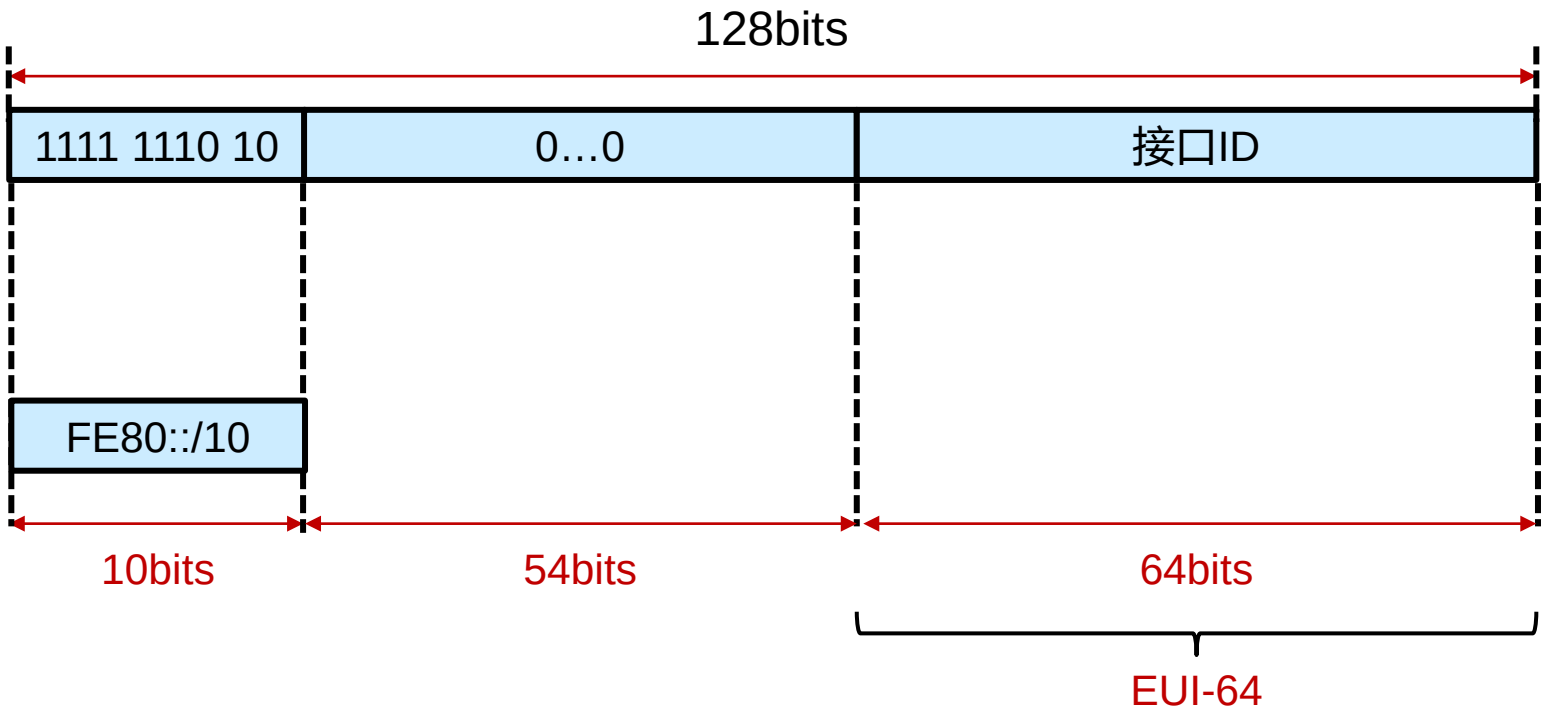
- 类似于IPv4私有地址
- 使用站点本地地址意味着需要NAT，地址不是端到端的
- 地址开头为 FEC0::/10，紧接着是连续的38bits的0
- 对于站点本地地址来说，前48bits 总是固定的。在接口ID和48bits 特定前缀之间有16bits子网ID字段，供机构在内部构建子网
- 本地站点地址永远不会用于与全球ipv6因特网通信，一般用于内网通信

单播地址 - LinkLocal address

- 有效范围为本地链路
- 以FE80::/10为前缀，11-64位为0 + 一个64位接口标识
- 用于自动地址配置、邻居发现、路由器发现等
- 在一条链路上，必须知道对方节点的链路本地地址，如果不知道，将是不能通信的，所以一条链路中的IPv6节点要通信，必须拥有链路本地地址，并且这个链路本地地址只在一条链路中有效，也不能被路由，而不同链路的链路本地地址是可以重复的。

单播地址 - LinkLocal address

- Linklocal地址的构成



接口标识符

- **关于接口ID**
  - 接口ID为64bits，用于标识链路上的接口，在每条链路上接口ID必须唯一
- **接口ID的设置**
  - 可以根据IEEE的EUI-64规范将48比特的MAC地址转化为64比特的接口ID。
  - 手工配置
  - 某些系统支持自动生成随机接口ID
- **接口ID的作用**
  - 可用于构成LinkLocal地址
  - 可在无状态配置环境中用于构成IPv6地址

EUI-64地址

MAC地址 0012-3400-ABCD

二进制表示

0000000000010010	0011010000000000	1010101111001101

插入FFFE

0000000000010010	0011010011111111	1111111000000000	1010101111001101

设置U/L位

0000001000010010	0011010011111111	1111111000000000	1010101111001101

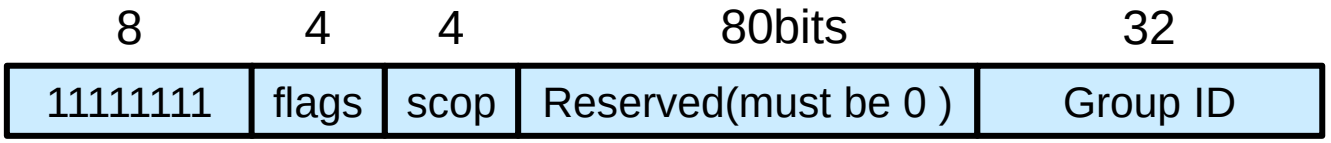
1 =全球唯一
0 =本地唯一

EUI-64地址 0212:34FF:FE00:ABCD

组播地址（Multicast）

- 用来标识一组接口，发送给多播地址的数据流同时传输到多个目的地
- 范围：FF00::/8
- 几个常见的IPv6组播地址：
  - FF02::1：表示链路上的所有节点
  - FF02::2：表示链路上的所有路由器
  - FF02::9：表示链路上的所有RIP路由器

组播地址



- **Flags**
  - 用来表示永久或临时组播组。
  - 0000表示 永久分配或众所周知 ；
  - 0001表示 临时的
- **Scope**
  - 表示组播组的范围
- **Group ID**
  - 组播组ID

0：预留；

1：节点本地范围；单个接口有效，仅用于Loopback通讯

2：链路本地范围； FF02::1

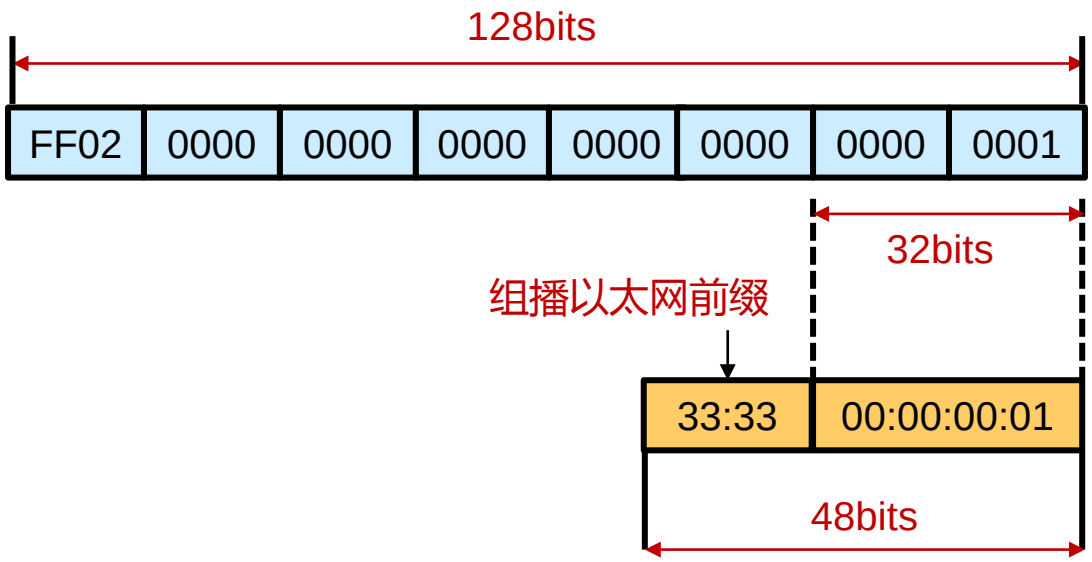
5：站点本地范围；

8：组织本地范围；

E：全球范围；

F：预留。

组播地址的MAC映射

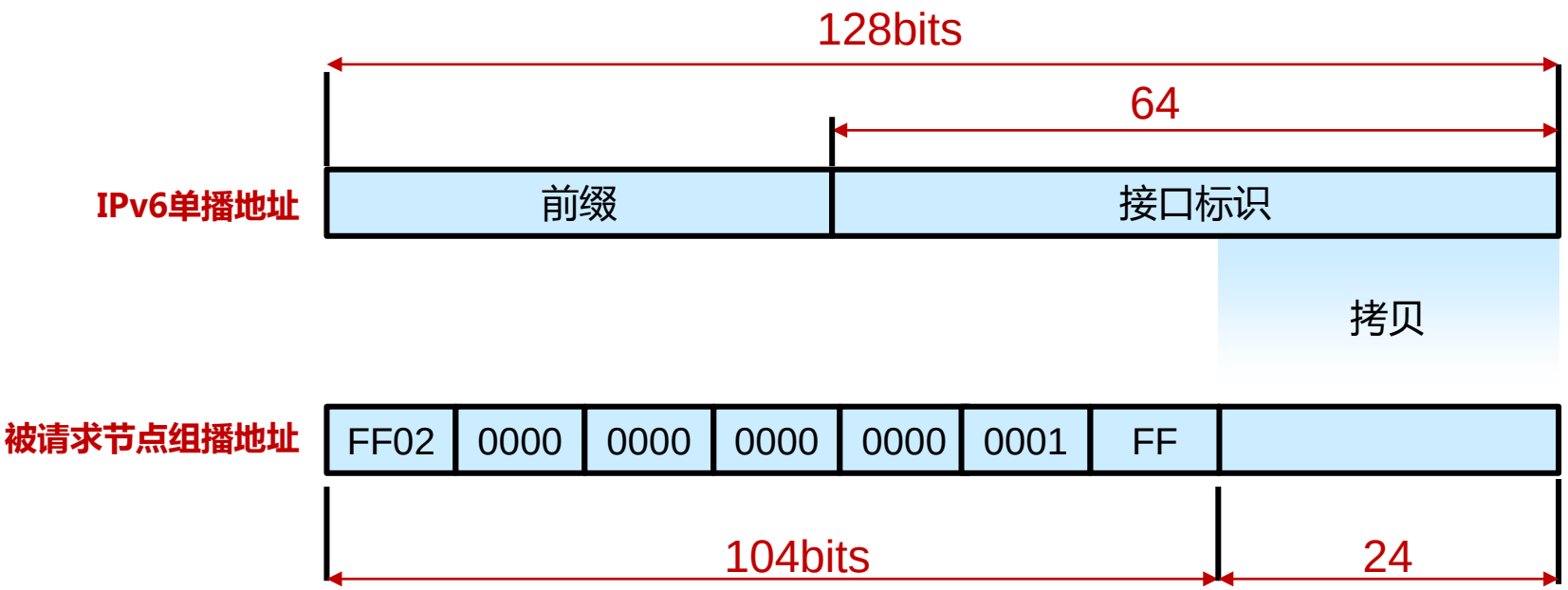


组播地址 - 被请求节点组播地址 Solicited-node

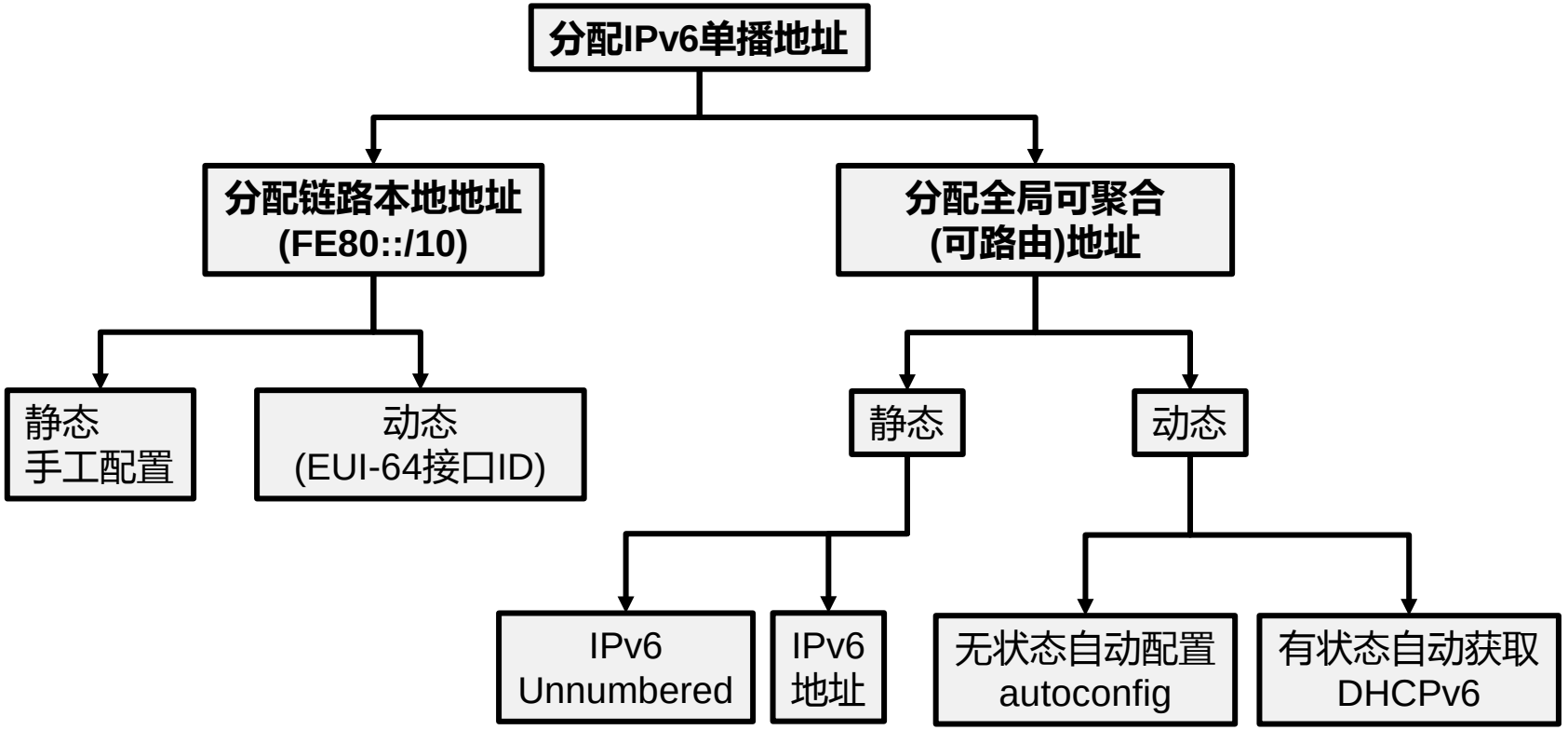
- 主要用于重复地址检测（DAD）和替代ipv4中的ARP
- 由前缀FF02::1:FF00:0 / l04和ipv6单播地址的最后24位组成
- 一个ipv6单播地址对应一个Solicited-node地址
- Solicited-node地址受限范围为本地链路范围

组播地址 - 被请求节点组播地址 Solicited-node

- IPv6地址对应的被请求节点组播地址



IPv6单播地址的分配方法

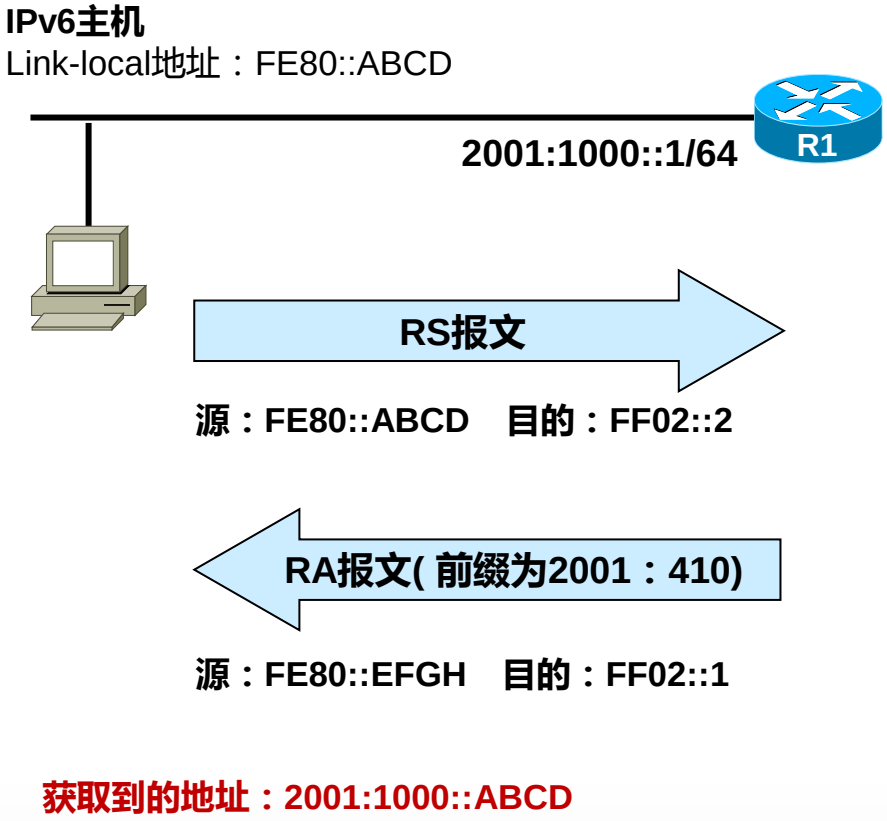


静态IPv6地址配置（使用EUI-64地址）

```
r1(config)# interface fast0/0
r1(config-if)# ipv6 address 2035:1:2bc5::87c:0:a/64 eui-64
```

动态IPv6地址分配（无状态）

1. 主机发送router Solicitation报文
2. 路由器回应Router Advertisement报文
3. 主机获得前缀及其它参数
4. 路由器周期性地向外发送RA报文



获取到的地址：2001:1000::ABCD

红茶三杯
Vinsoney

IPv6基本配置

IPv6基本配置

- **配置和验证IPv6单播地址**

```
ipv6 unicast-routing
```

- 启用转发IPv6单播流量转发功能

```
Interface fast0/0
  Ipv6 enable
  ipv6 address address/prefix-lengrh [ eui-64 | link-local ]
  no shutdown
```

- 为接口配置IPv6地址和前缀

IPv6基本配置

- **配置和验证IPv6单播地址**

```
show ipv6 interface
show ipv6 routers
show ipv6 neighbors
debug ipv6 packet
```

  - 验证IPv6配置

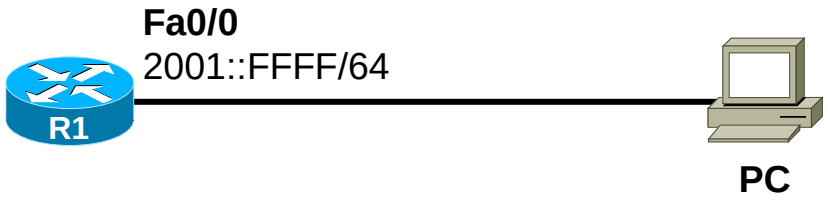
IPv6基本配置

- **配置和验证IPv6单播地址**

```
PC1#sh ipv6 int brief
FastEthernet0/0                        [up/up]
    FE80::CE03:18FF:FE68:0
    2001:1::1
```

  - 验证IPv6配置

实验1 无状态自动配置



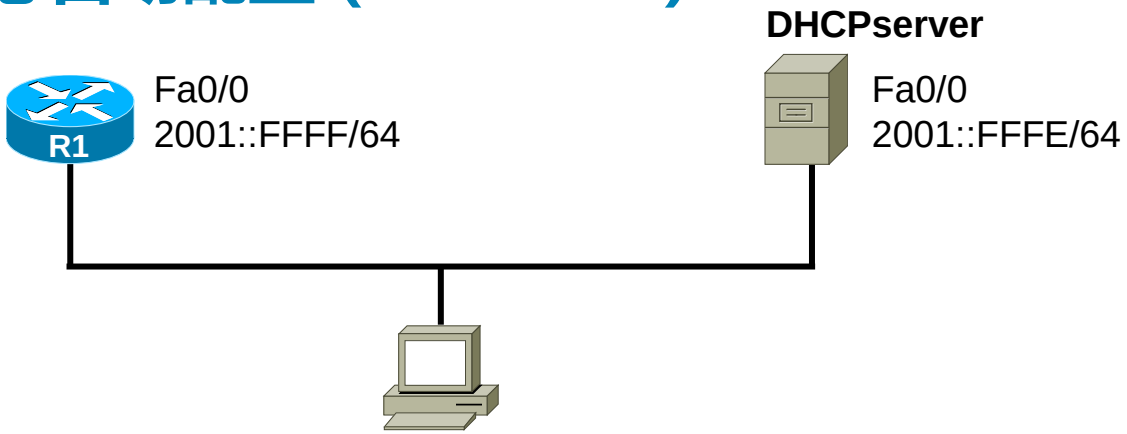
R1的配置如下：

```
ipv6 unicast-routing
interface fast0/0
  ipv6 address 2001::FFFF/64
  no ipv6 nd suppress-ra
  !! 接口地址2001::1/64， 同时开启路由器通告（默认关闭）
```

PC的配置（用路由器模拟）：

```
ipv6 address autoconfig [default]
!! 如果加default关键字，则会在获取到地址的接口上添加一个默认网关（默认路由）
```

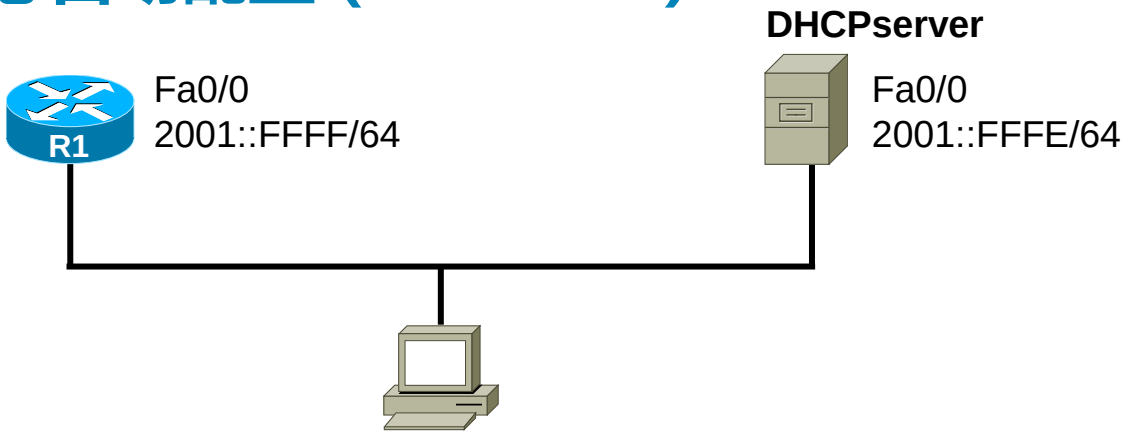
实验2 有状态自动配置（DHCPv6）



DHCPserver的配置：

```
ipv6 unicast-routing
ipv6 dhcp pool DHCP-pool
  prefix-delegation pool dhcppool lifetime 1800 600
  dns-server 2000::8
  domain-name HelloWord
ipv6 local pool dhcppool 2001::/64 64
!
interface FastEthernet0/0
  ipv6 enable
  ipv6 address 2001::FFFE/64
  ipv6 nd other-config-flag
  ipv6 nd managed-config-flag
  ipv6 dhcp server DHCP-pool                    !! 在接口上开启ipv6 DHCP，并调用池
```

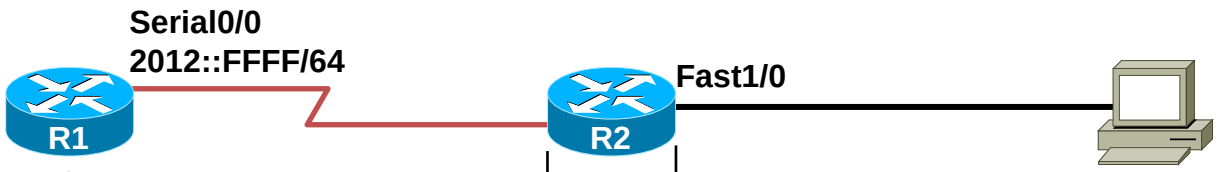
实验2 有状态自动配置（DHCPv6） cont.



DHCPclient的配置：

```
interface FastEthernet0/0
ipv6 enable                          !! 接口激活IPV6
ipv6 dhcp client pd test             !! 配置DHCP client，并且指定获取到的前缀名称为
test（本地有效）
ipv6 address test ::/64 eui-64       !! 使用获取到的前缀（test），加上本接口的EUI64，
构成接口全局ipv6地址
```

实验3 DHCP-PD




```
ipv6 dhcp pool DHCP-pool
  prefix-delegation pool dhcppool lifetime 1800 600
  dns-server 2000::8
  domain-name HelloWord
ipv6 local pool dhcppool 2001::/64 64

interface Serial0/0
  ipv6 address 2012::FFFF/64
  ipv6 dhcp server DHCP-pool
```

```
interface Serial0/0
  ipv6 address autoconfig default
  ipv6 dhcp client pd test
```

```
Interface fast1/0
  ipv6 address test  ::FFFF/64
```

Tea 红茶三杯
ccietea.com

沉淀 提升 成长 分享
关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

ICMPv6

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2013-06-18

课程目标

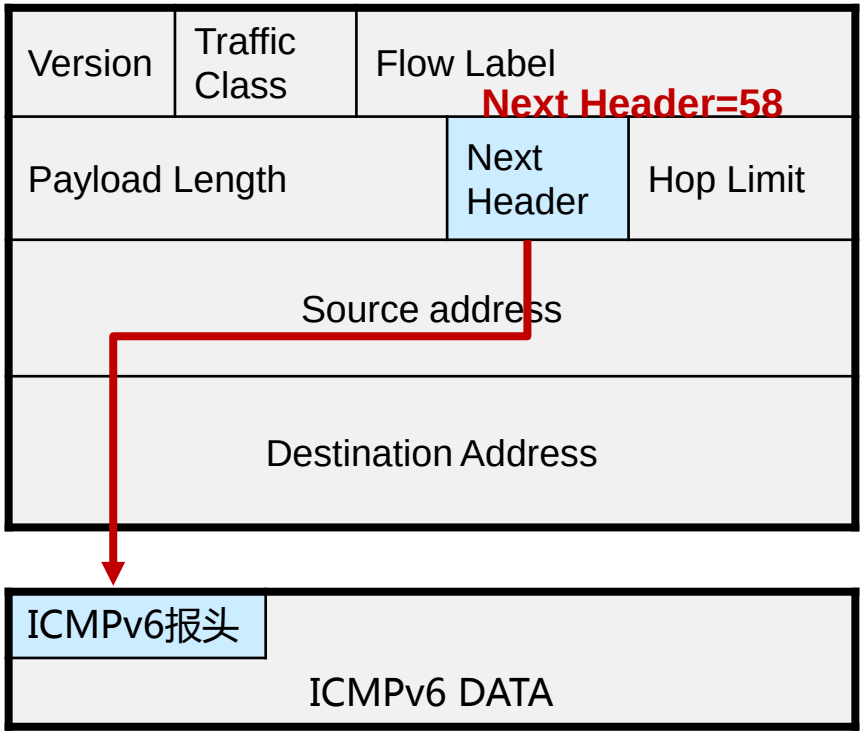
- ICMPv6概述
- PMTUD
- NDP

ICMPv6概述

ICMPv6概述

- ICMPv6 是IPv6的基础协议之一
- 协议号58，该协议号在IPv6报头的“下一个包头”字段中。
- ICMP报文有两种：差错消息及信息消息

ICMPv6



ICMPv6

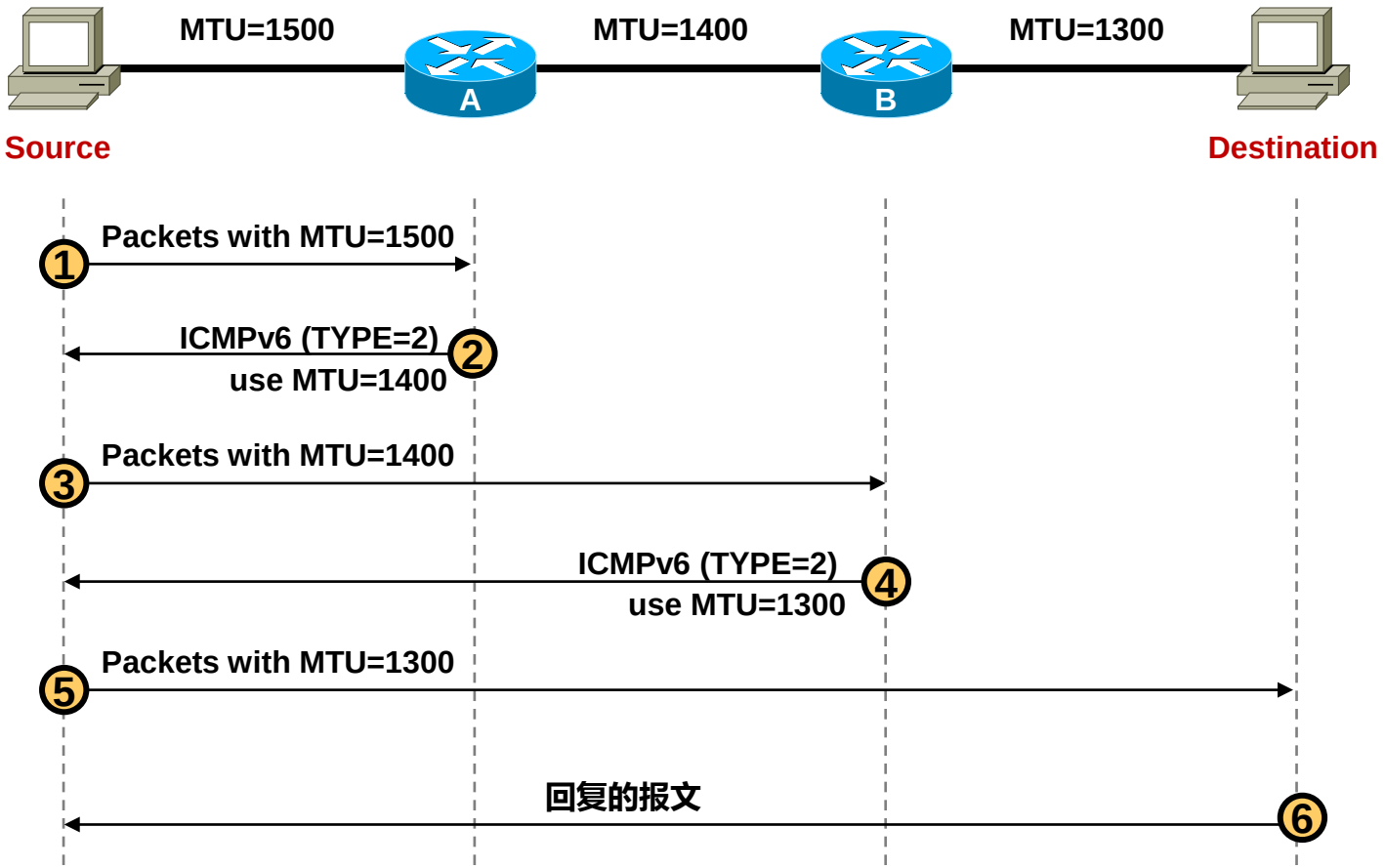
消息类型	TYPE	名称	CODE
差错消息	1	目的不可达	0 无路由
			1 因管理原因禁止访问
			2 未指定
			3 地址不可达
			4 端口不可达
	2	数据包过长	0
	3	超时	0 跳数到0
			1 分片重组超时
	4	参数错误	0 错误的包头字段
			1 无法识别的下一包头类型
			2 无法识别的ipv6选项
信息消息	128	Echo request	0
	129	Echo reply	0

还有一些其他报文，为NDP而定义，后续介绍

红茶三杯
Vinsoney

PMTUD

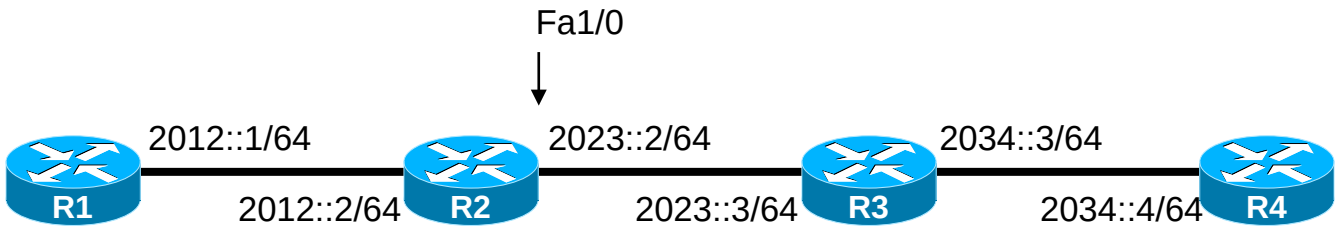
PMTUD过程



PMTUD过程

1. 首先Source用1500字节作为MTU向目标节点发送一个IPv6数据包
2. 中间路由器A意识到数据包过大，MTU为1400，于是回复一个ICMPv6 type=2消息向Source应答，在该ICMPv6消息中指定较小的MTU=1400
3. Source开始使用MTU=1400发送IPv6数据包，该数据包到了B
4. 然而B意识到本地接口MTU为1300，于是回复一个ICMPv6 type=2消息向Source应答
5. Source开始使用MTU=1300发送IPv6数据包，该数据包顺利到达了目的地。
6. Source和Destination之间的会话被建立起来。

测试

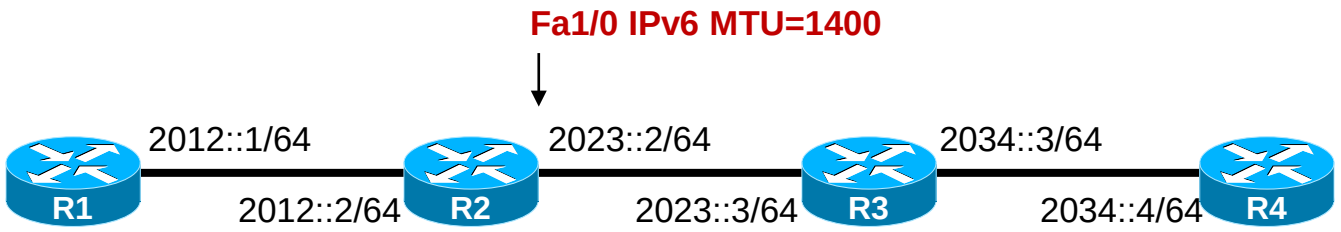


R2的配置修改如下：

```
Interface fast1/0
  Ipv6 mtu 1400
```

```
R2#show ipv6 interface f1/0
FastEthernet1/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::CE01:CFF:FEF0:10
  Global unicast address(es):
    2023::2, subnet is 2023::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::5
    FF02::6
    FF02::1:FF00:2
    FF02::1:FFF0:10
  MTU is 1400 bytes
……
```

测试 cont.



R1#ping 2034::4 repeat 1 size 1500

!! 让R1产生一个报文大小为1500的包，发给R4

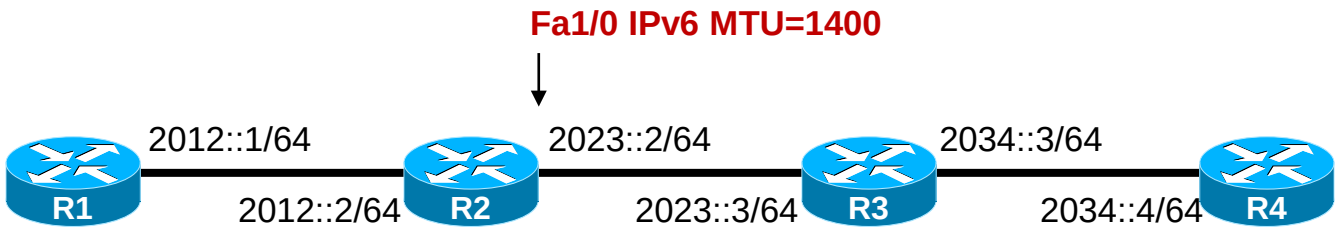
Type escape sequence to abort.

Sending 1, 1500-byte ICMP Echos to 2034::4, timeout is 2 seconds:

B

Time	Source	Destination	Protocol	Length	Info
62.942000	2012::1	2034::4	ICMPv6	1514	Echo (ping) request id=0x02
62.963000	2012::2	2012::1	ICMPv6	1294	Packet Too Big

测试 cont.

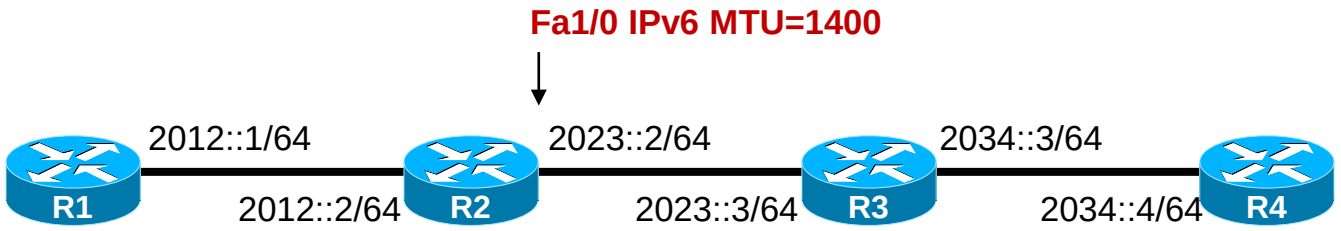


Time	Source	Destination	Protocol	Length	Info
62.942000	2012::1	2034::4	ICMPv6	1514	Echo (ping) request id=0x02
62.963000	2012::2	2012::1	ICMPv6	1294	Packet Too Big

R2回送给R1的ICMPv6差错消息如下

```
Ethernet II, Src: cc:01:0c:f0:00:00 (cc:01:0c:f0:00:00), Dst: cc:00:0c:f0:00:
Internet Protocol Version 6, Src: 2012::2 (2012::2), Dst: 2012::1 (2012::1)
Internet Control Message Protocol v6
  Type: Packet Too Big (2)
  Code: 0
  Checksum: 0xbb6f [correct]
  MTU: 1400
⊞ Internet Protocol Version 6, Src: 2012::1 (2012::1), Dst: 2034::4 (2034::4)
⊞ Internet Control Message Protocol v6
```

测试 cont.



此时，在R1上进一步查看一下：

R1#sh ipv6 mtu

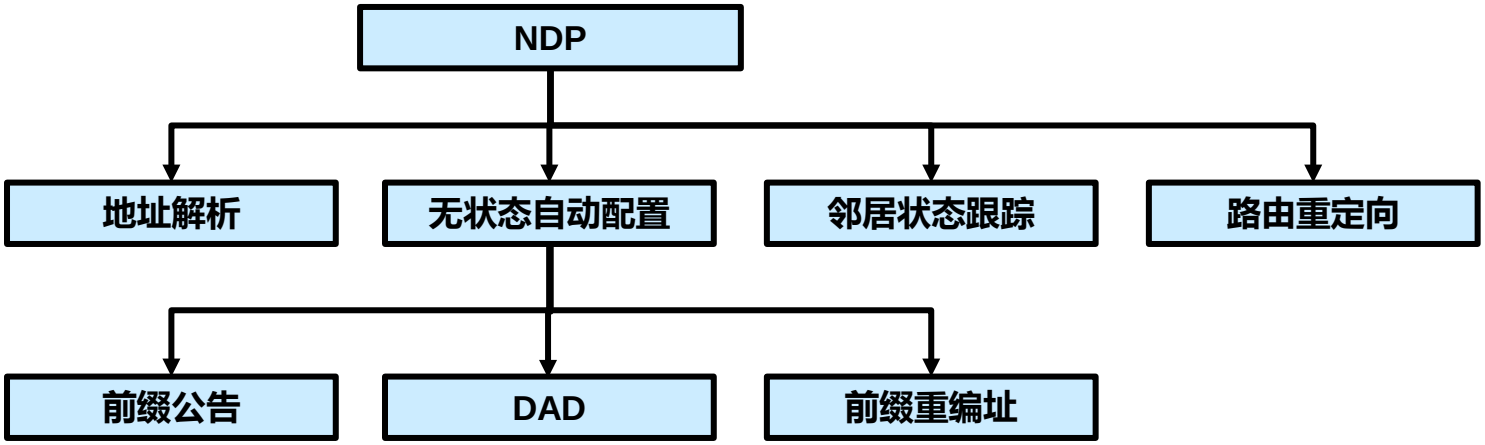
MTU	Since	Source Address	Destination Address
1400	00:00:27	2012::1	2034::4

此后R1如果要发送超过1400字节的报文到R4，则会在本地进行分片，可抓包查看。

红茶三杯
Vinsoney

IPv6邻居发现机制（NDP）

NDP能够实现这些功能



为NDP定义的icmpv6消息

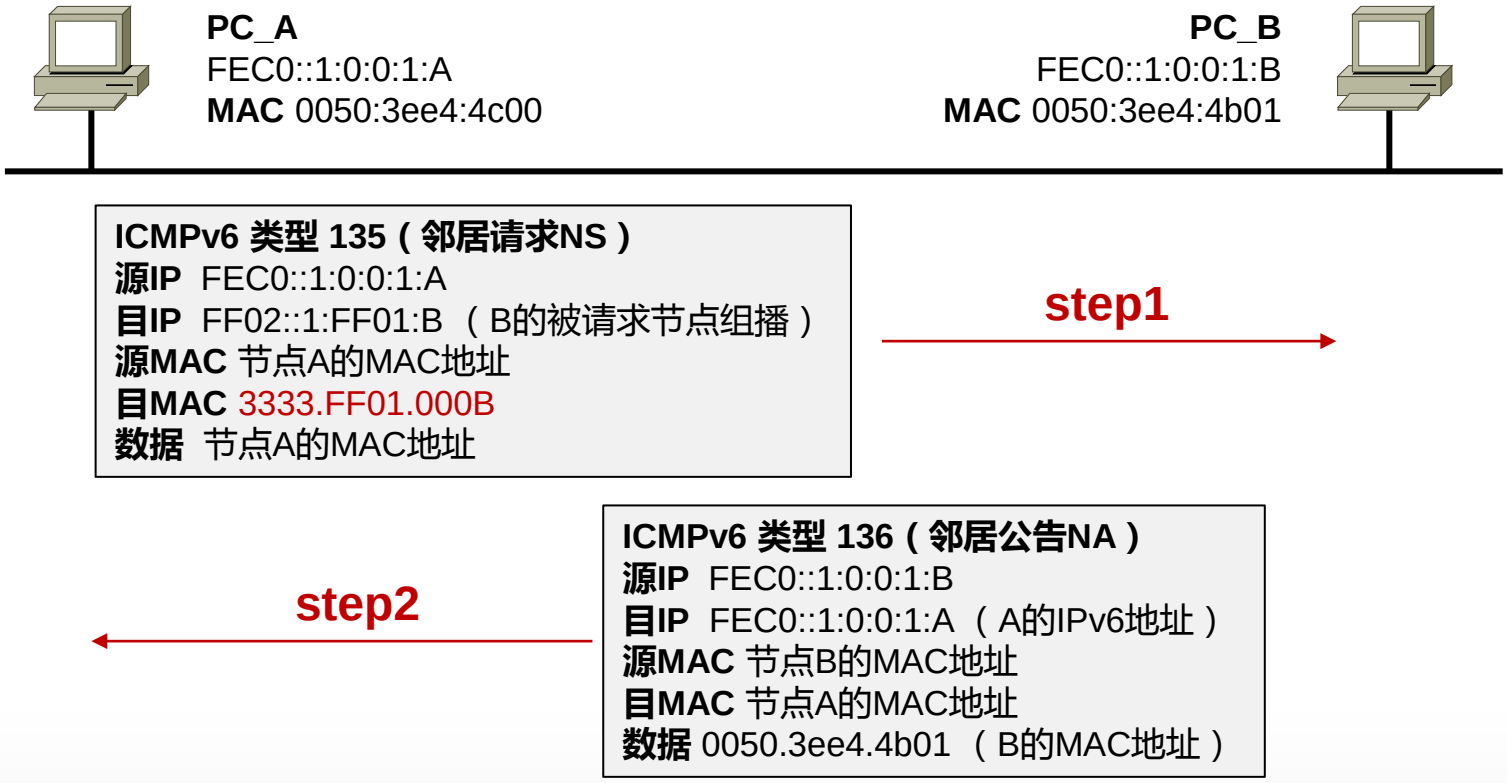
ICMPv6 TYPE	消息名称
133	路由器请求（RS）
134	路由器通告（RA）
135	邻居请求（NS）
136	邻居通告（NA）
137	重定向消息

NDP机制使用的ICMPv6消息

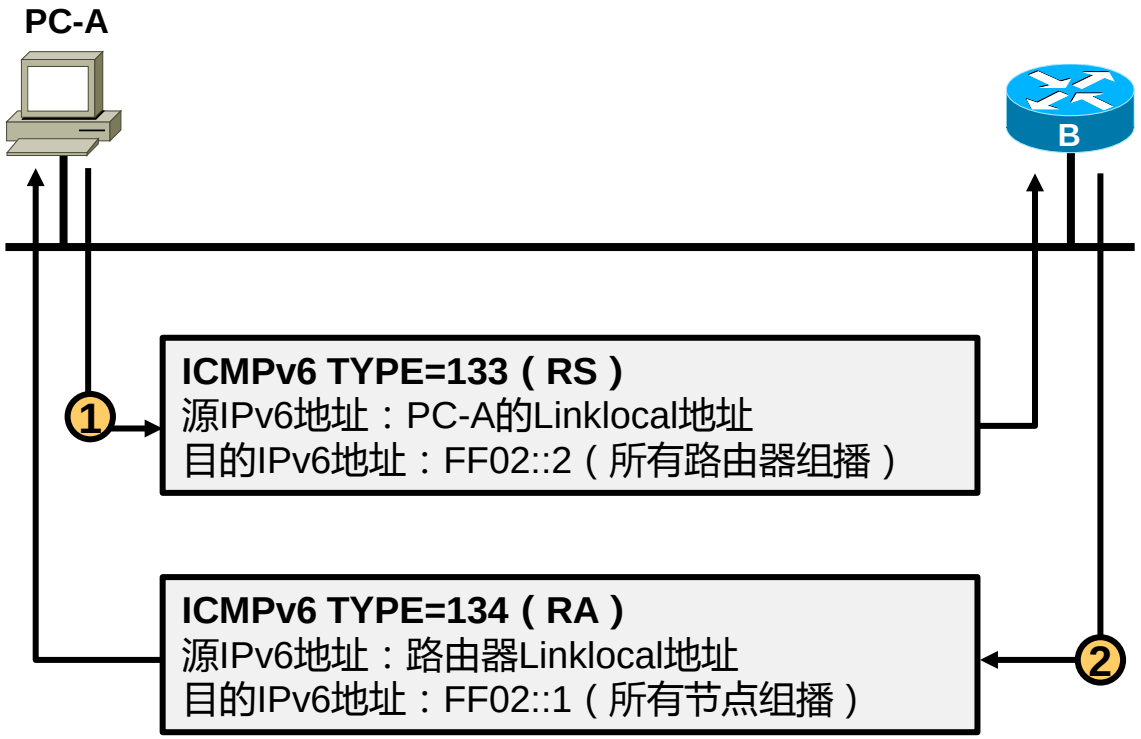
机制	RS 133	RA 134	NS 135	NA 136	重定向消息 137
报文介绍	主机可以发送RS要求路由器立即产生RA	包含MTU、前缀信息等	用来判断邻居的链路层地址也用于DAD等		
替代ARP			X	X	
前缀公告	X	X			
前缀重新编制	X	X			
DAD			X		
路由重定向					X

用ICMPv6实现地址解析

- 通过邻居请求（NS）和邻居通告（NA）报文来解析三层地址对应的链路层地址。

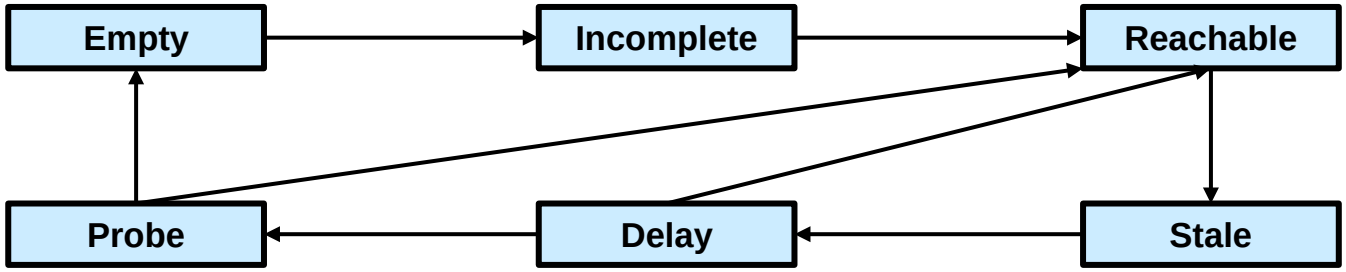


节点主动发送RS来请求RA



为了避免路由器请求消息在本地链路上泛滥，在启动时每个节点只能发送3个RS消息。

用ICMPv6跟踪邻居状态



1. A发送NS，并生成缓存条目，状态为Incomplete
2. 若B回复NA，则Incomplete ->Reachable。否则10S后Incomplete->Empty，即删除条目
3. 经过ReachableTime（默认30S），B的条目状态Reachable->stale
4. 或者在Reachable状态，收到B的非请求NA，且链路层地址不同，则马上->stale
5. 在Stale状态若A要向B发送数据，并从Stale->Delay，等待应用层的提示信息，提示邻居可达
6. 在Delay_First_Probe_Time（默认5S）内，若有NA应答或者应用层的提示信息，则Delay->Reachable，无应用层提示信息，Delay->Probe
7. 在Probe状态，每隔RetransTimer（默认1S）发送单播NS，发送MAX_Unicast_Solicit个后再等RetransTimer，有应答则Reachable，无则进入Empty，即删除条目

ICMPv6 DAD

- **机制概述**
  - 无状态配置和节点启动时的一个NDP机制，通过 NS（ ICMP 135）
  - 使用源地址（ ：： ）、目的地址为获取到的v6地址对应的被请求节点组播地址的NS报文
- **原理**
  - 一个地址在通过重复地址检测之前称为“tentative地址”，试验地址。接口还暂时不能使用这个试验地址进行正常单播通讯，但是会加入和tentative地址所对应的Solicited-Node组播组。
  - 重复地址检测：节点向一个自己将使用的tentative地址所在的组播组发送一个NS，如果收到某个其他站点回应的NA，就证明该地址已被网络上使用，节点将不能使用该tentative地址通讯。

ICMPv6 DAD

ICMPv6 类型 135（邻居请求NS）
源IP ：：
目IP **FF02::1:FF00:1**（2000::1对应的被请求节点组播）
ICMP内容：Target address：2000::1

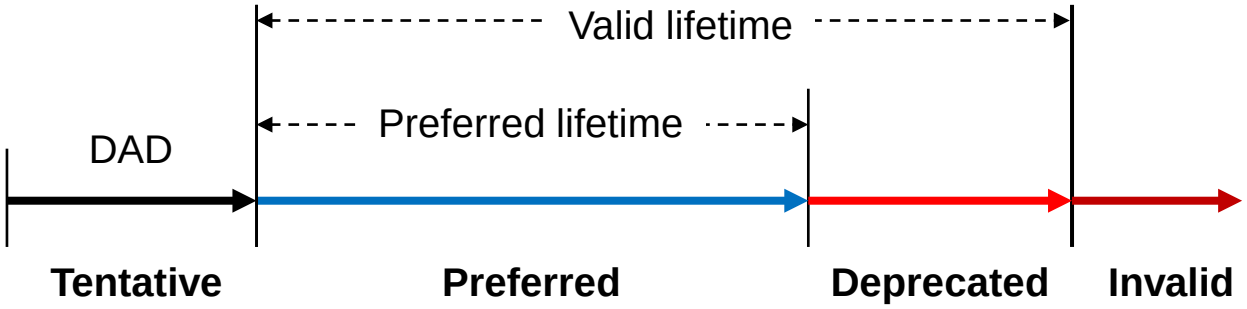
step1 →

← **step2**

ICMPv6 类型 136（邻居通告NA）
源IP **2000::1**
目IP **FF02::1**
ICMP内容：Target address：2000::1

如果1S后没有检测到冲突，A就会发送non-solicited advertisement（一个NA消息），宣告大家我将正式使用这个IPv6地址

ICMPv6 DAD

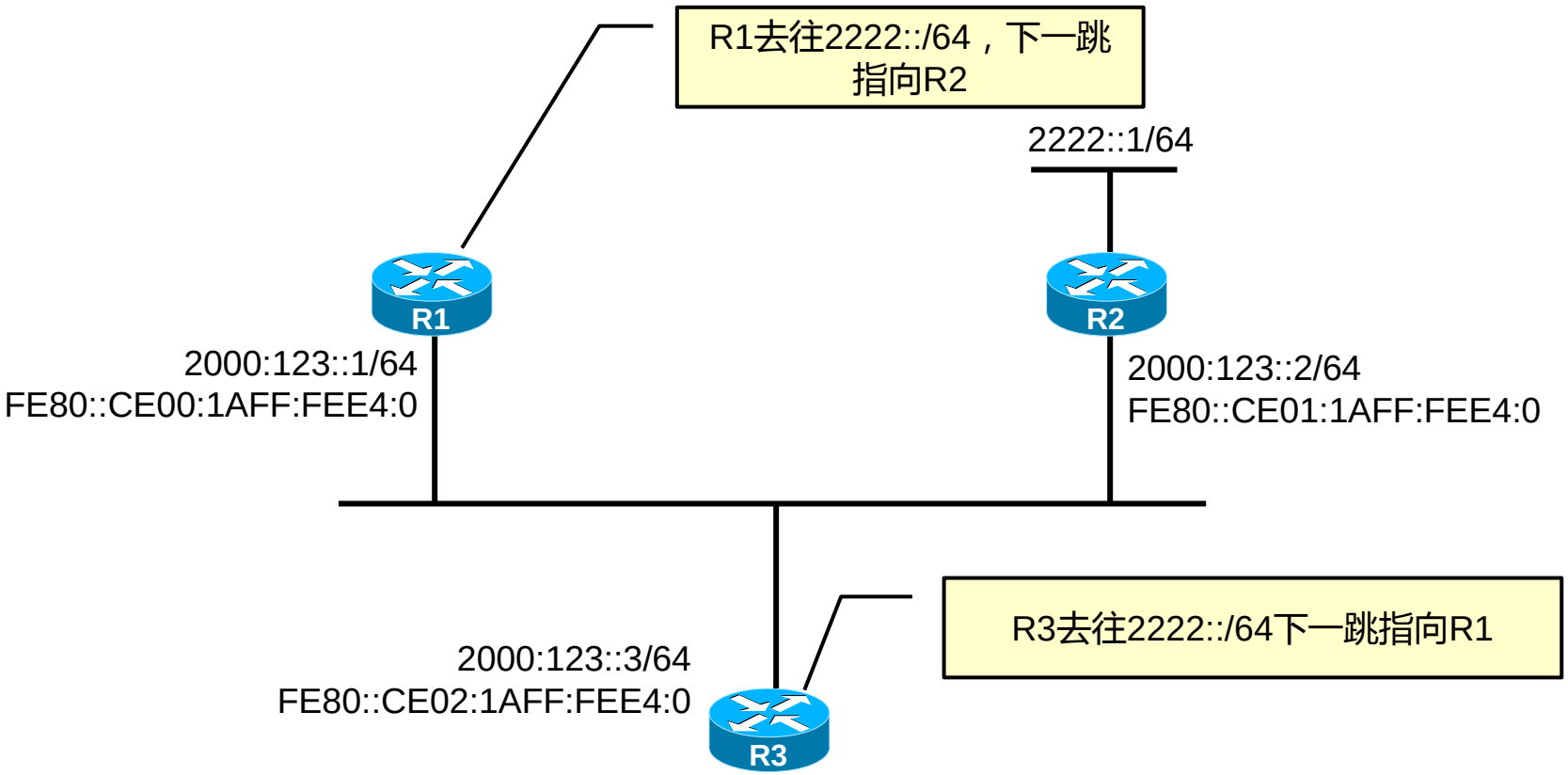


- 当地址处于Deprecated状态，地址不能主动的发起连接只能是被动的接受连接，这也是为了保证上层应用而设计的，但是过了valid lifetime时间地址就变为invalid，这时任何连接就会down掉

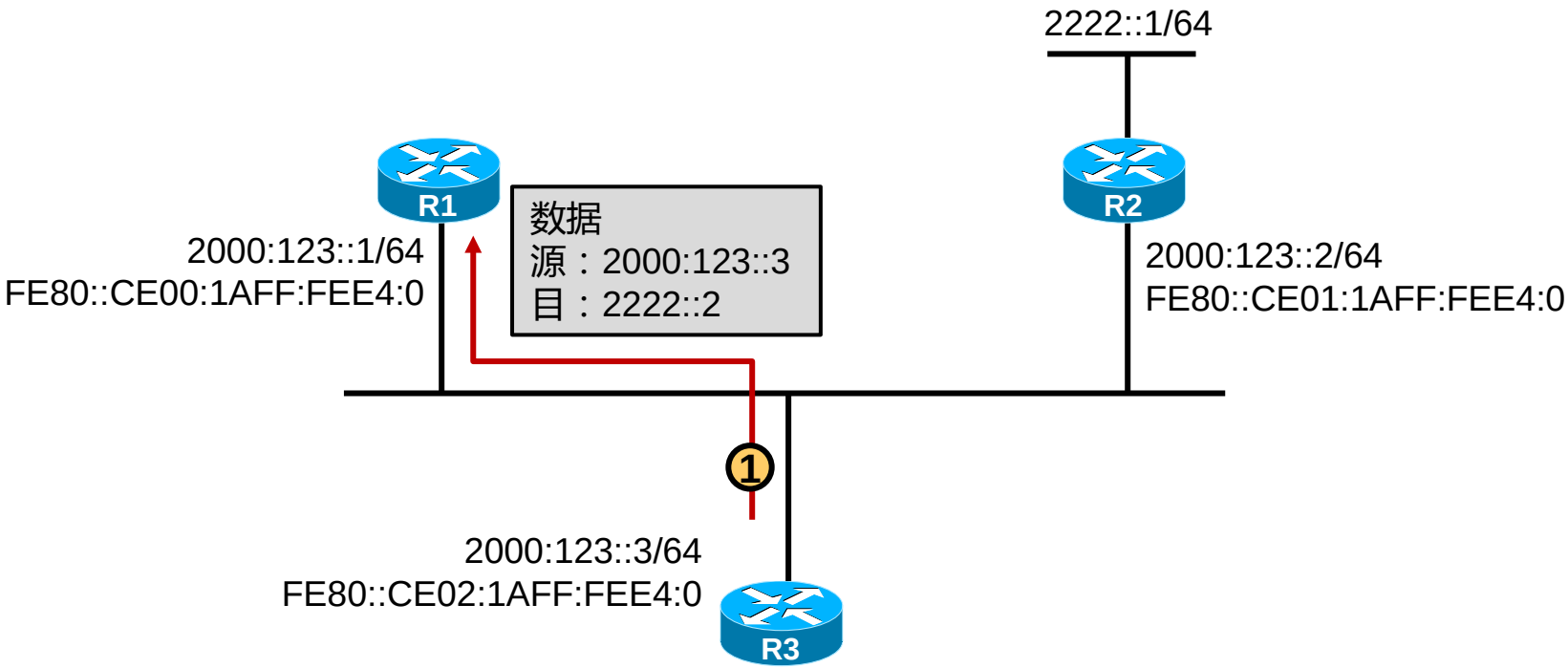
ICMPv6重定向

- 路由器使用ICMPv6重定向消息（ICMP TYPE 137）通知链路上的节点，在链路上存在一个更好的前转数据包的路由器。接收到这个ICMPv6重定向消息的节点可以根据重定向消息中新的路由器地址修改它的本地路由选择表。

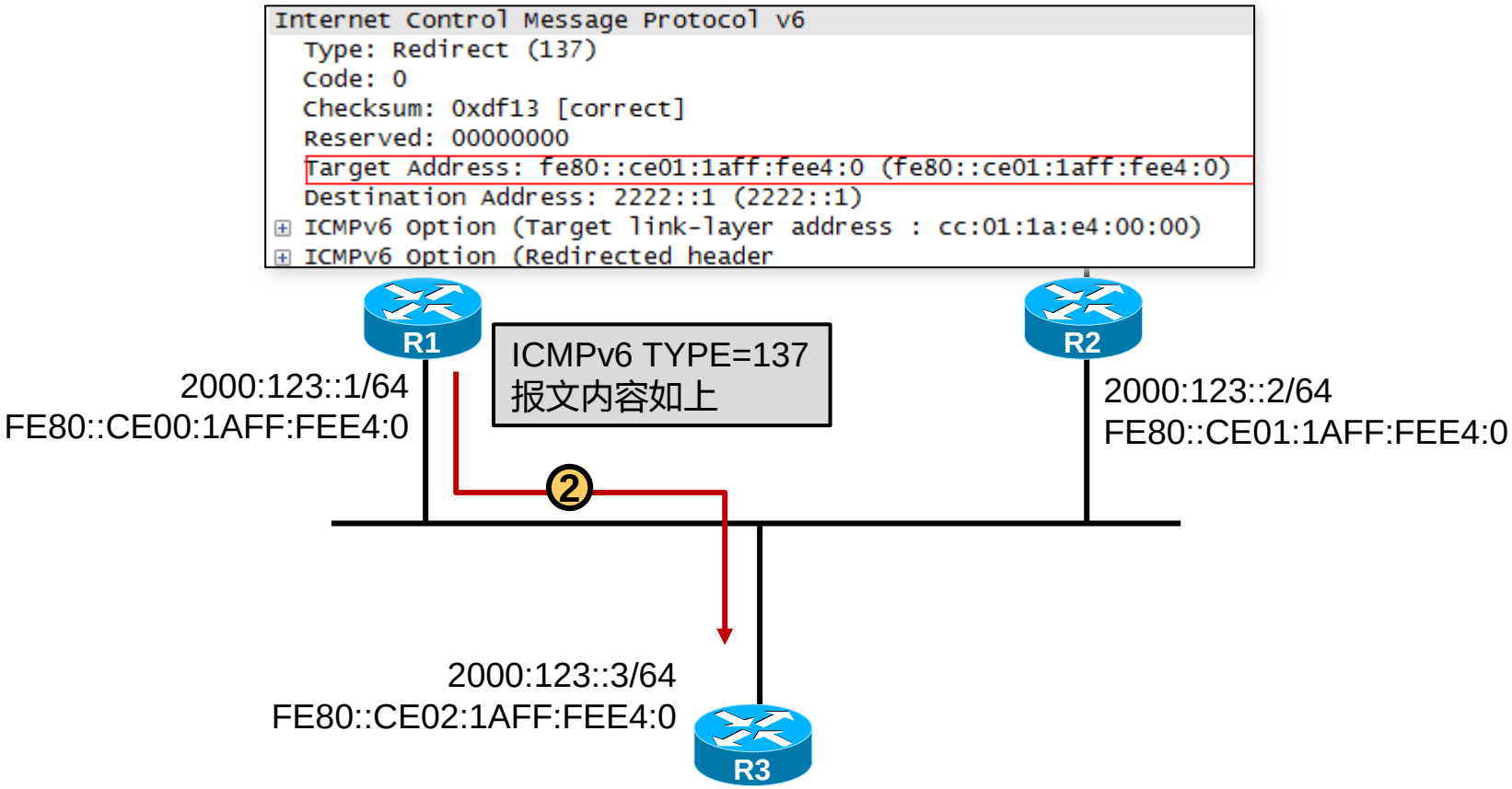
ICMPv6重定向



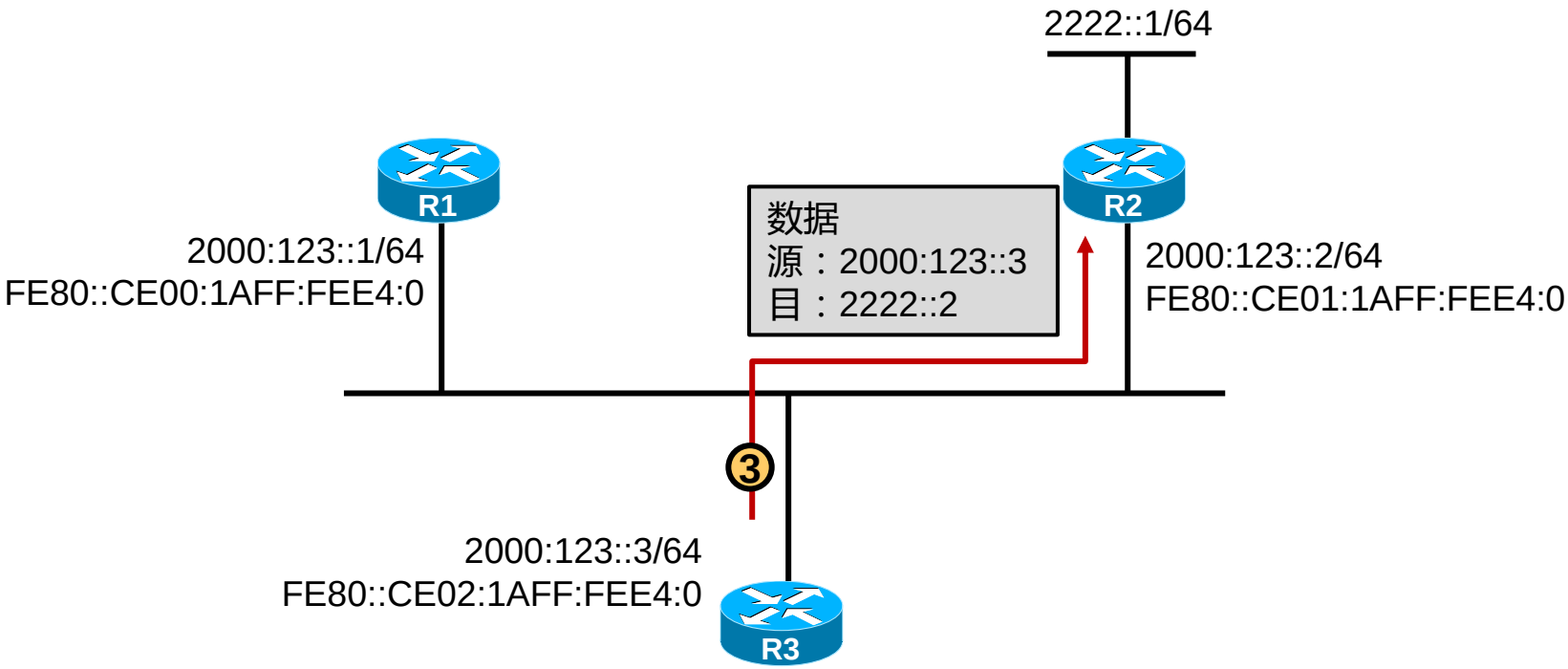
ICMPv6重定向



ICMPv6重定向



ICMPv6重定向



Tea 红茶三杯
ccietea.com

沉淀 提升 成长 分享
关注@红茶三杯：weibo.com/vinsoney

Thank You

红茶三杯
Vinsoney

学习 沉淀 成长 分享

IPv6过渡技术

红茶三杯（朱SIR） http://weibo.com/vinsoney

Latest update: 2013-06-18

课程目标

IPv6过渡技术概述

IPv4/IPv6双栈

Tunnel机制

NAT-PT

红茶三杯
Vinsoney

IPv6过渡技术概述

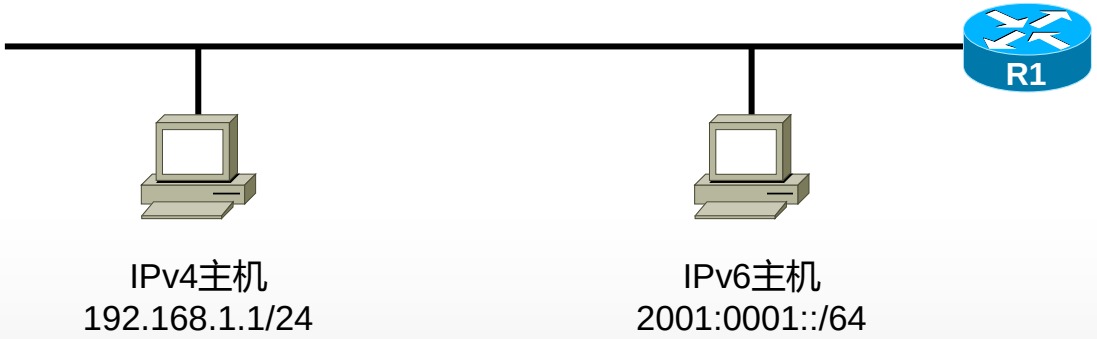
从IPv4过渡到IPv6

- **双协议栈技术**
  - 设备上同时使用IPv4和IPv6协议栈，设备同时支持V4及V6协议栈，并且灵活进行选择，该机制是其他过渡技术的基础
- **隧道技术(Tunnel)**
  - 把IPv6报文封装在IPv4报文中，IPv6网络之间穿越IPv4网络进行通信
- **协议转换技术**
  - 具备 IPv4和IPv6协议转换功能的转换设备，修改协议报文头，使IPv4网络与IPv6网络能够互通

IPv4/IPv6双栈

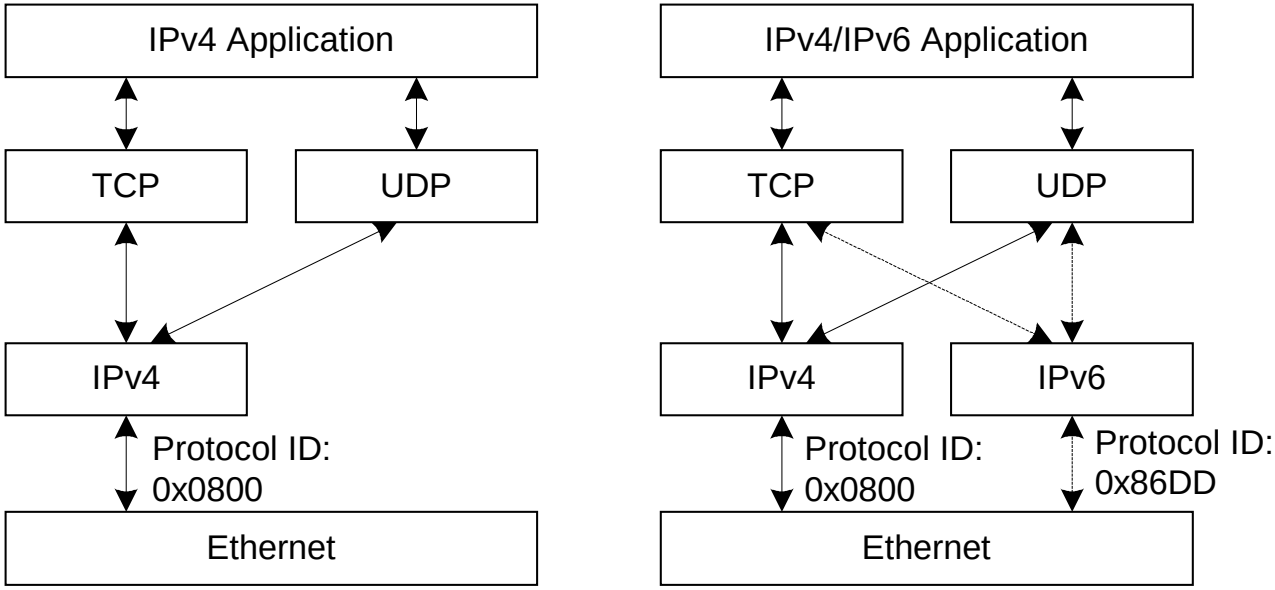
- 节点有IPv4及IPv6两个协议栈
- 缺点：每台设备都需要配置两种协议，需要占用资源，设备需要存储两个路由表（如果是路由器），需要独立处理每种协议。

```
router(config)# ipv6 unicast-routing
router(config)# interface fast0/0
router(config-if)# ipv6 enable
router(config-if)# ip address 192.168.1.254 255.255.255.0
router(config-if)# ipv6 address 2001::0001::FFFE/64
```

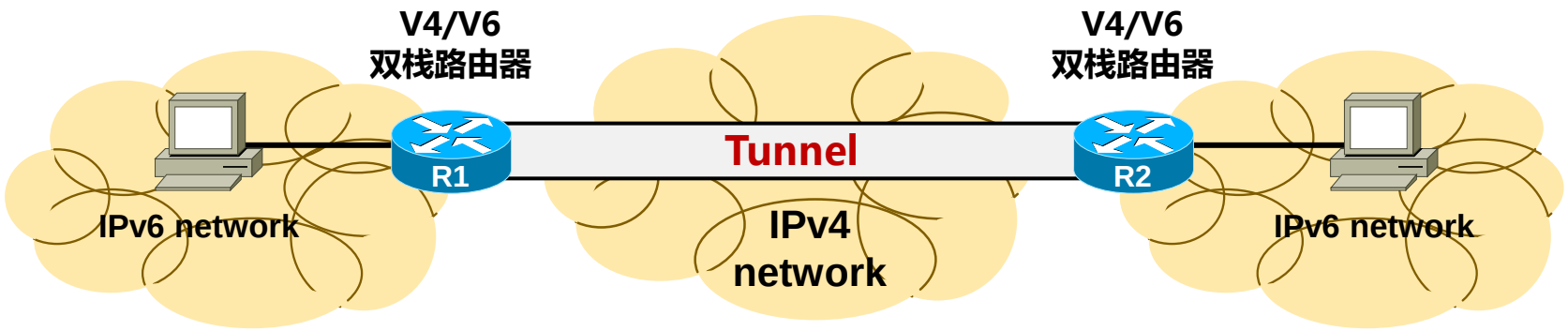


IPv4/IPv6双栈

- **节点有IPv4及IPv6两个协议栈**

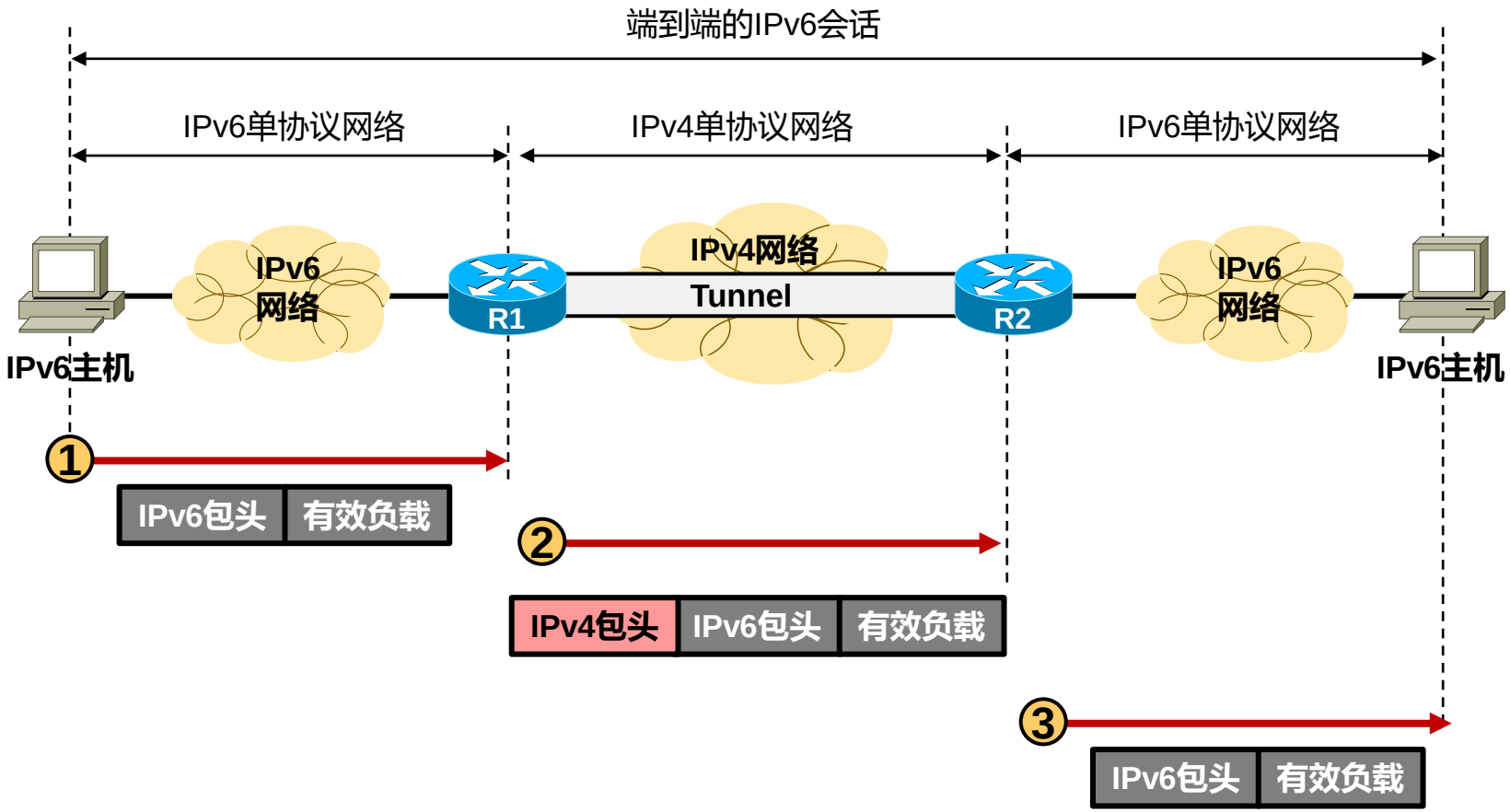


隧道机制概述

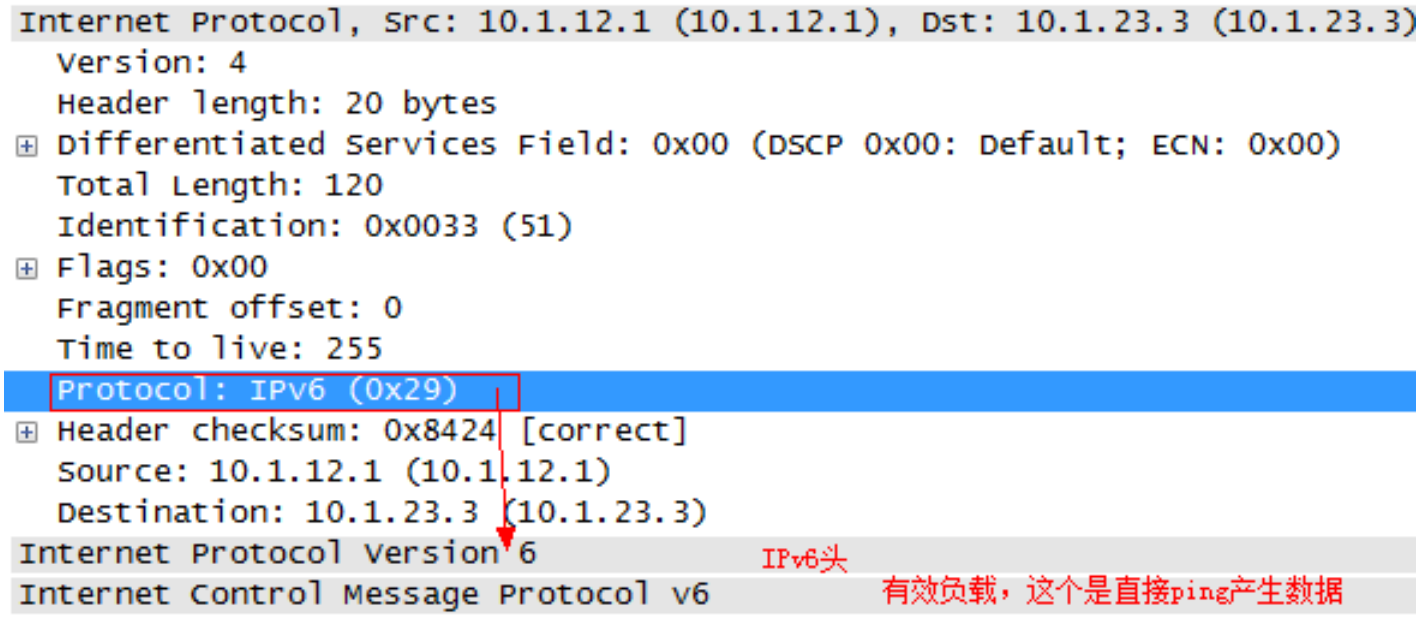


- **隧道技术**一般用于在现有网络中传输不兼容的协议或特殊的数据。在因特网上无处不在的现有IPv4基础设施中，使用隧道技术可以使得IPv6孤岛得以连接。
- 当然，相对于任何过渡和并存的策略（如隧道技术），我们还是首选由纯IPv6连接组成的网络、链路和基础设施。实际上，只有当在网络、连接和基础设施中不可能获得纯IPv6连接性时，IPv4基础设施上的IPv6隧道机制才被认为是一种可选择的方法。

隧道机制概述



隧道机制概述(cont.)



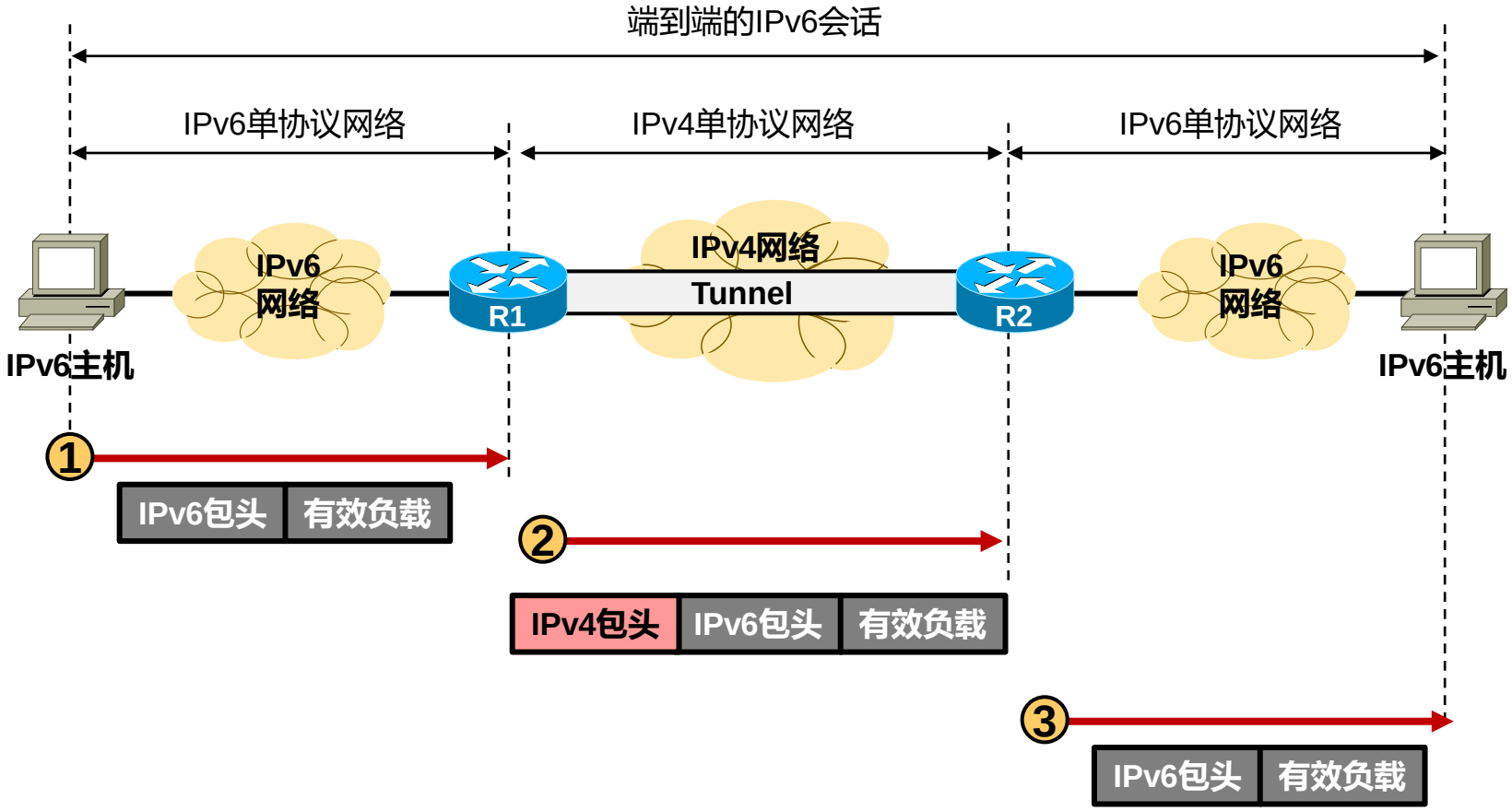
红茶三杯
Vinsoney

Tunnel机制

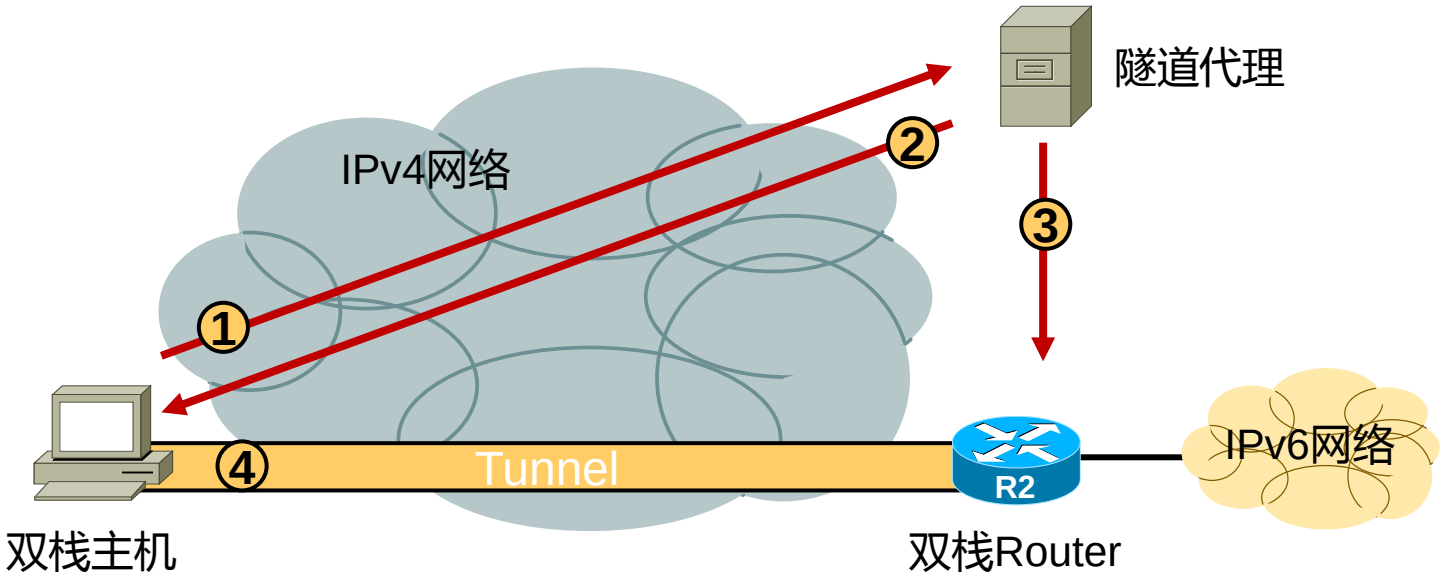
隧道技术种类

- 手工隧道技术：手工IPv6 over IP隧道、GRE隧道
- 自动隧道技术：6to4自动隧道、IPv4兼容IPv6自动隧道、ISATAP隧道
- 6PE：6PE技术依赖于BGP，BGP的Peer是需要手工指定的，可以算是一种半自动隧道技术。

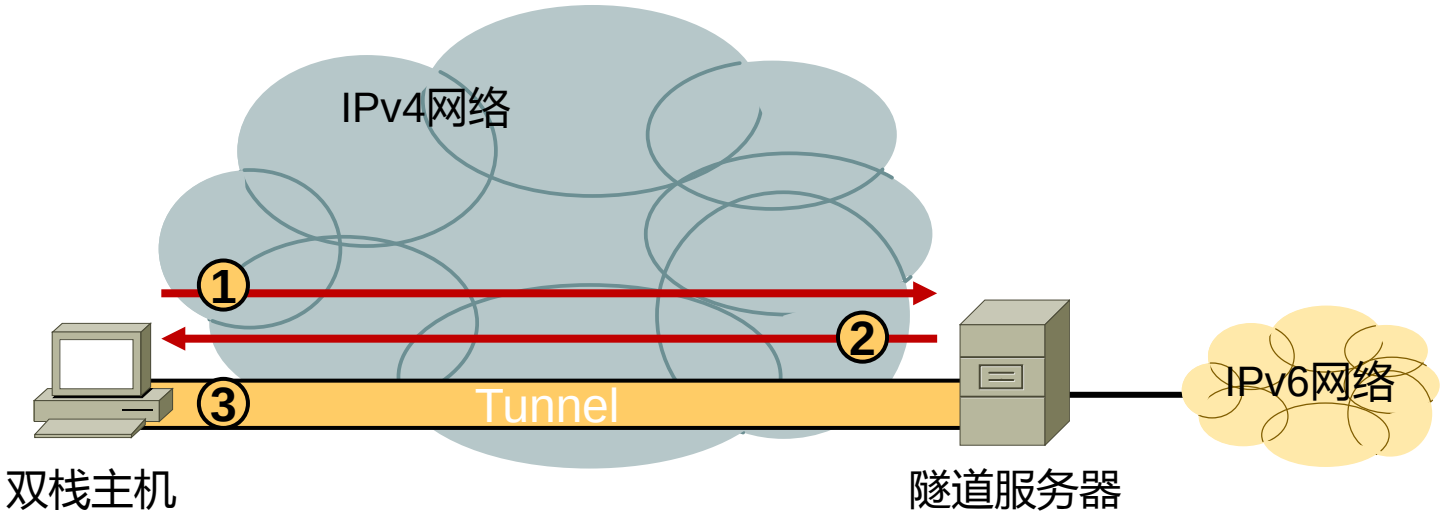
手工IPv6 over IP隧道



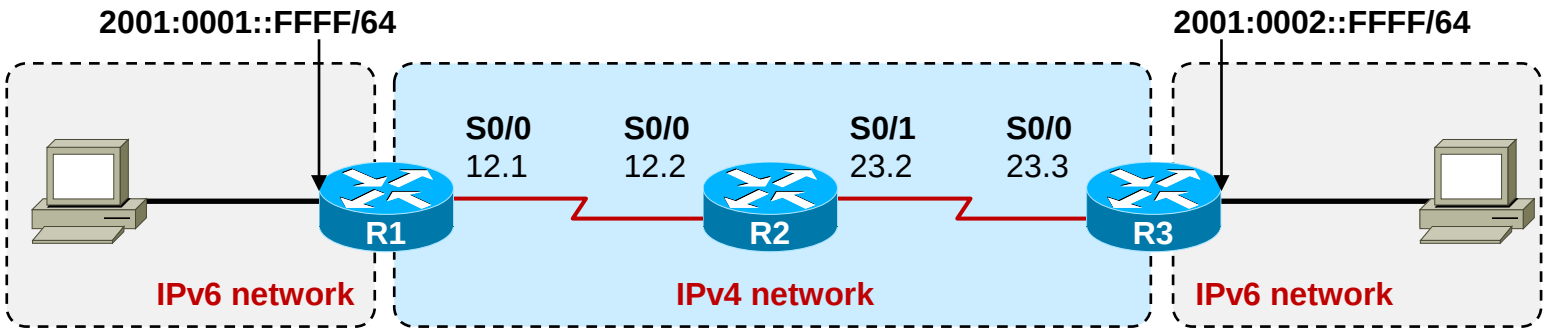
手工IPv6 over IP隧道 – 隧道代理



手工IPv6 over IP隧道 – 隧道服务器



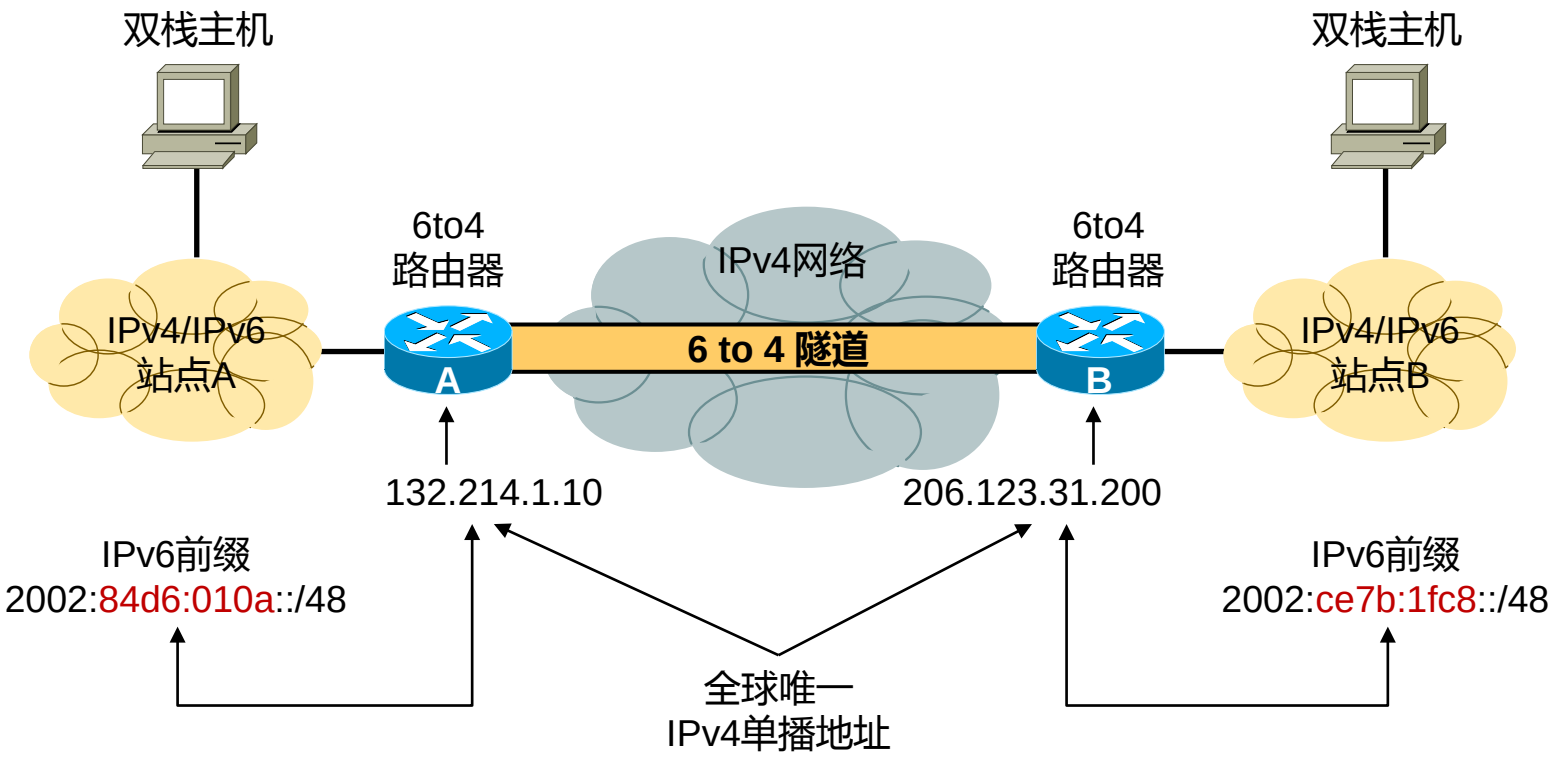
手工IPv6 over IP隧道配置



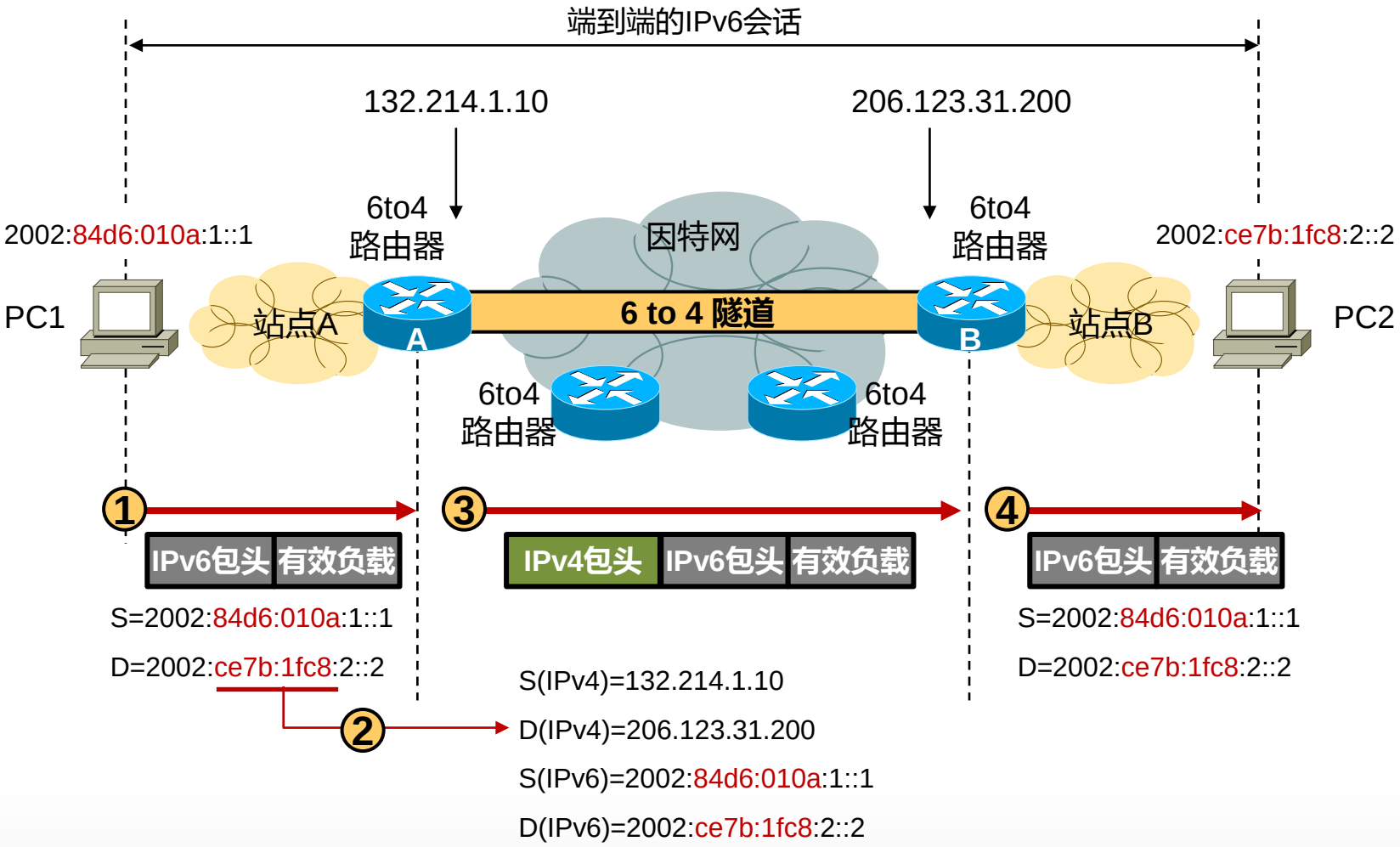

```
Ipv6 unicast-routing
Interface serial0/0
  ip address 10.1.12.1 255.255.255.0
Interface fast1/0
  Ipv6 enable
  Ipv6 address 2001:0001::FFFF/64
Interface tunnel 0
Ipv6 enable
tunnel mode ipv6ip
Tunnel source serial 0/0
Tunnel destination 10.1.23.3
ip route 0.0.0.0 0.0.0.0 10.1.12.2
Ipv6 route ::/0  tunnel 0
```

```
Ipv6 unicast-routing
Interface serial0/0
  ip address 10.1.23.3 255.255.255.0
Interface fast1/0
  Ipv6 enable
  Ipv6 address 2001:0002::FFFF/64
Interface tunnel 0
Ipv6 enable
tunnel mode ipv6ip
Tunnel source serial 0/0
Tunnel destination 10.1.12.1
ip route 0.0.0.0 0.0.0.0 10.1.23.2
Ipv6 route ::/0  tunnel 0
```

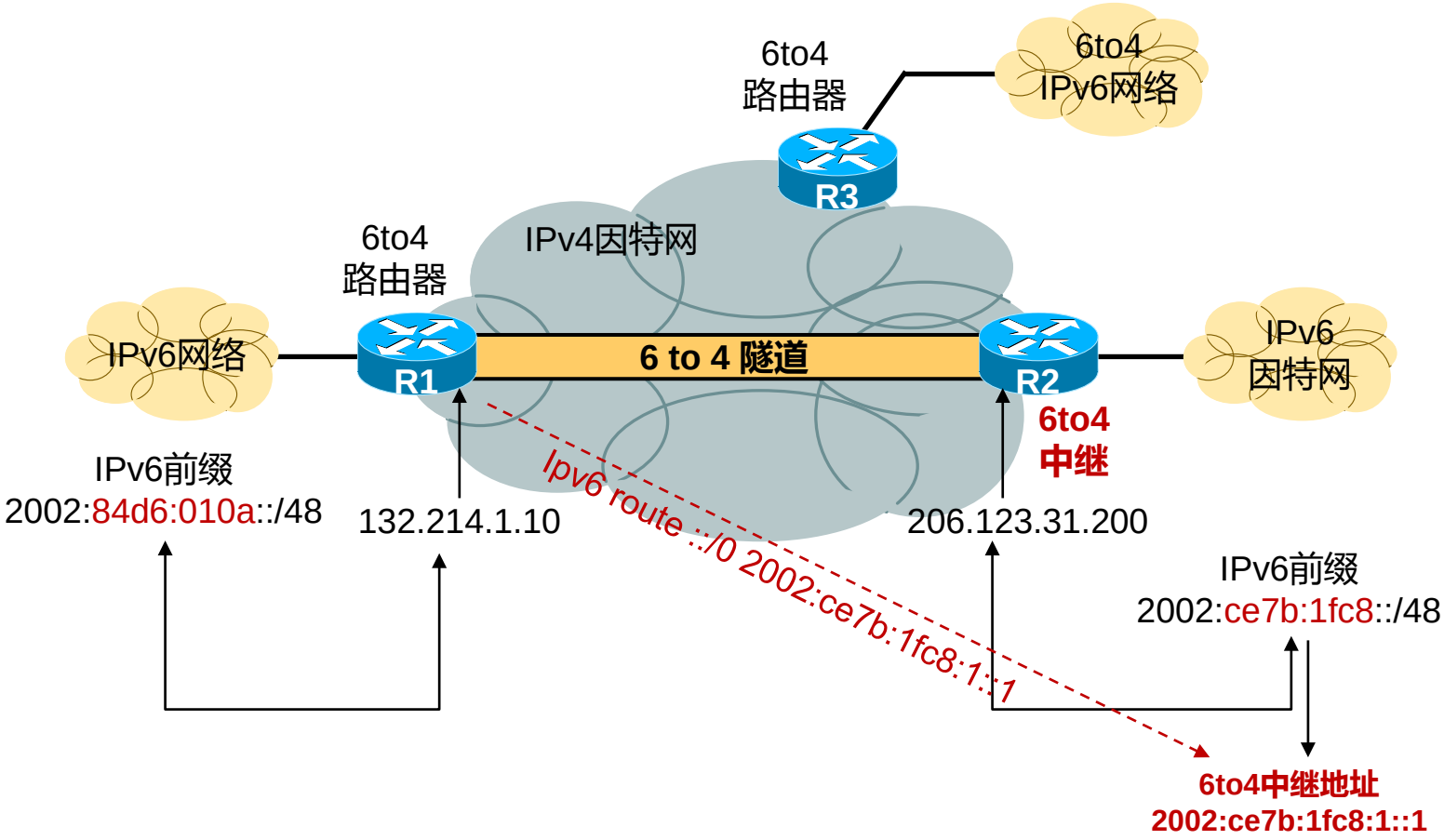
6to4自动隧道



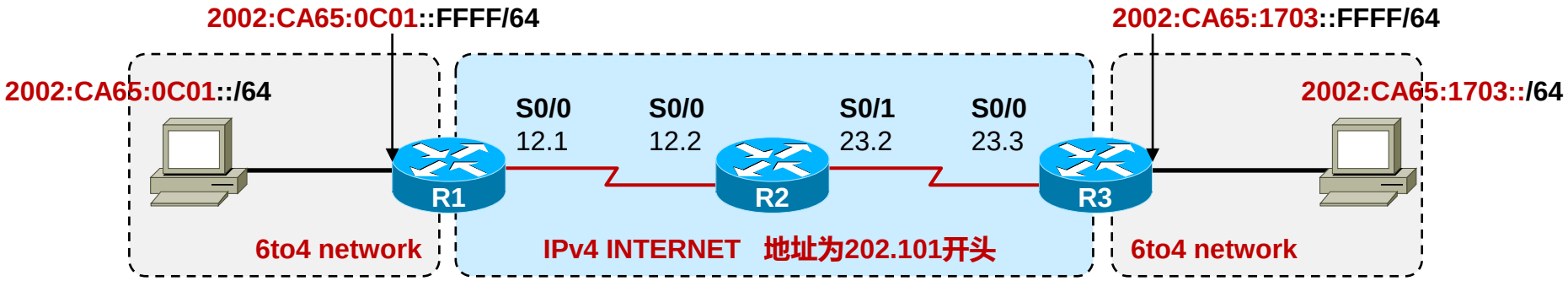
6to4自动隧道 – 详细报文交互过程



6to4中继



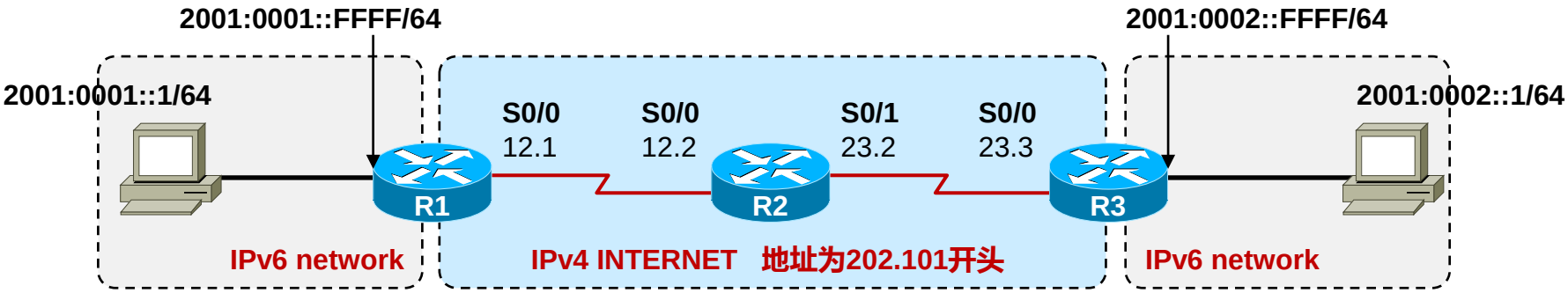
6to4自动隧道 配置（两边都是6to4网络）




```
ipv6 unicast-routing
interface Tunnel0
  ipv6 enable
  tunnel source Serial0/0
  tunnel mode ipv6ip 6to4
interface fast1/0
  ipv6 address 2002:CA65:0C01::FFFF/64
  ipv6 enable
  no ipv6 nd suppress-ra
interface Serial0/1
  ip address 202.101.12.1 255.255.255.0
ipv6 route 2002::/16 Tunnel0
```

```
ipv6 unicast-routing
interface Tunnel0
  ipv6 enable
  tunnel source Serial0/0
  tunnel mode ipv6ip 6to4
interface fast1/0
  ipv6 address 2002:CA65:1703::FFFF/64
  ipv6 enable
  no ipv6 nd suppress-ra
interface Serial0/1
  ip address 202.101.23.3 255.255.255.0
ipv6 route 2002::/16 Tunnel0
```

6to4自动隧道 配置（两边都是普通IPv6网络）



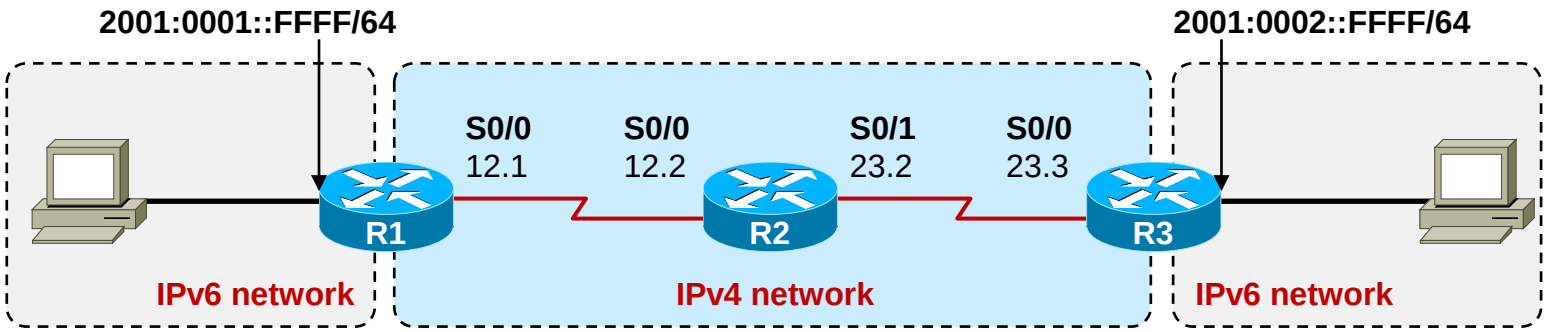

```
ipv6 unicast-routing
interface Tunnel0
  ipv6 address 2002:CA65:C01::FFFF/64
  ipv6 enable
  tunnel source Serial0/0
  tunnel mode ipv6ip 6to4
interface fast1/0
ipv6 address 2001:0001::FFFF/64
 ipv6 enable
interface Serial0/0
 ip address 202.101.12.1 255.255.255.0

ip route 0.0.0.0 0.0.0.0 202.101.12.2
ipv6 route 2001::/16  2002:CA65:1703::FFFF
ipv6 route 2002:CA65:1703::/48 Tunnel0
```

```
ipv6 unicast-routing
interface Tunnel0
  ipv6 address 2002:CA65:1703::FFFF/64
  ipv6 enable
  tunnel source Serial0/0
  tunnel mode ipv6ip 6to4
interface fast1/0
ipv6 address 2001:0002::FFFF/64
 ipv6 enable
interface Serial0/0
 ip address 202.101.23.3 255.255.255.0

ip route 0.0.0.0 0.0.0.0 202.101.23.2
ipv6 route 2001::/64  2002:CA65:C01::FFFF
ipv6 route 2002:CA65:C01::/48 Tunnel0
```

GRE隧道

- GRE隧道和手工隧道很相似
- GRE隧道是CISCO开发的，在Cisco路由器上，默认隧道使用GRE封装，可部署在多种传输协议之上，还可承载多种乘客协议
- 传输协议：IPv4或是IPv6
- 隧道协议：GRE
- 乘客协议：IPv6或是其他协议
- GRE协议号是：47

GRE隧道 配置



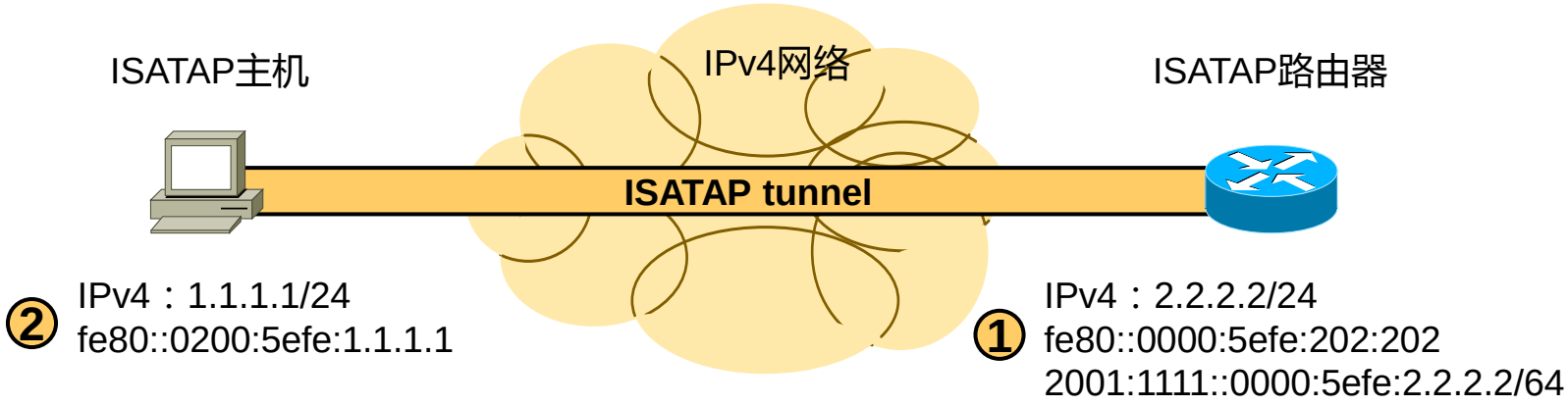

```
Ipv6 unicast-routing
!
Interface serial0/0
  ip address 10.1.12.1 255.255.255.0
  Interface fa1/0
  Ipv6 enable
  Ipv6 address 2001:0001::FFFF/64
!
Interface tunnel 0
  Ipv6 enable
  tunnel mode gre ip          !! 注意隧道模式
  tunnel source serial 0/0
  tunnel destination 10.1.23.3
!
ip route 0.0.0.0 0.0.0.0 10.1.12.2
Ipv6 route ::/0  tunnel 0
```

```
Ipv6 unicast-routing
!
Interface serial0/0
  ip address 10.1.23.3 255.255.255.0
Interface fa1/0
  Ipv6 enable
  Ipv6 address 2001:0002::FFFF/64
!
Interface tunnel 0
  Ipv6 enable
  tunnel mode gre ip          !! 注意隧道模式
  tunnel source serial 0/0
  tunnel destination 10.1.12.1
!
ip route 0.0.0.0 0.0.0.0 10.1.23.2
Ipv6 route ::/0  tunnel 0
```

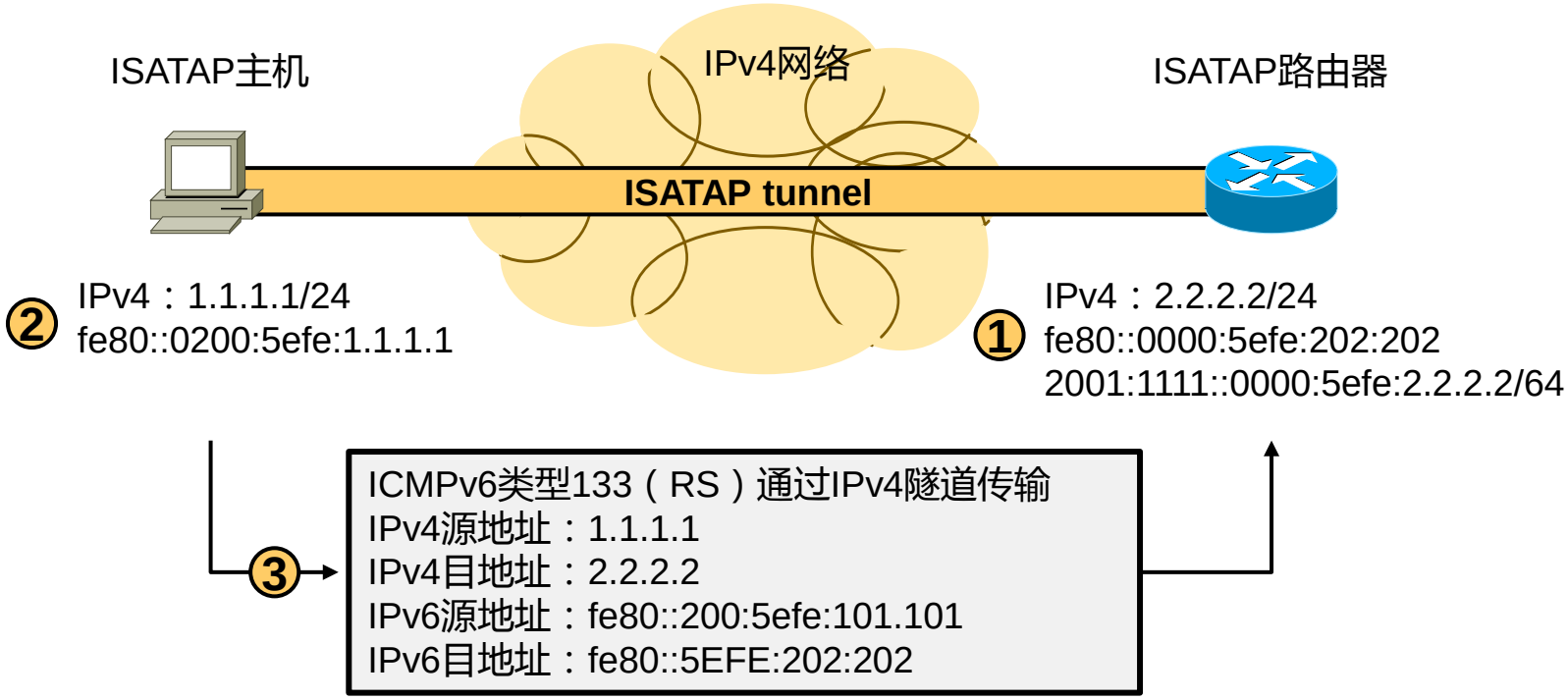
GRE隧道

```
⊞ Cisco HDLC
⊟ Internet Protocol, Src: 10.1.12.1 (10.1.12.1), Dst: 10.1.23.3 (10.1.23.3)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 124
    Identification: 0x0030 (48)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 255
    Protocol: GRE (0x2f)
  ⊞ Header checksum: 0x841d [correct]
    Source: 10.1.12.1 (10.1.12.1)
    Destination: 10.1.23.3 (10.1.23.3)
⊞ Generic Routing Encapsulation (IPv6)
⊟ Internet Protocol Version 6
  ⊞ 0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 60
    Next header: ICMPv6 (0x3a)
    Hop limit: 63
    Source: 2001:1::1 (2001:1::1)
    Destination: 2001:2::1 (2001:2::1)
⊞ Internet Control Message Protocol v6
```

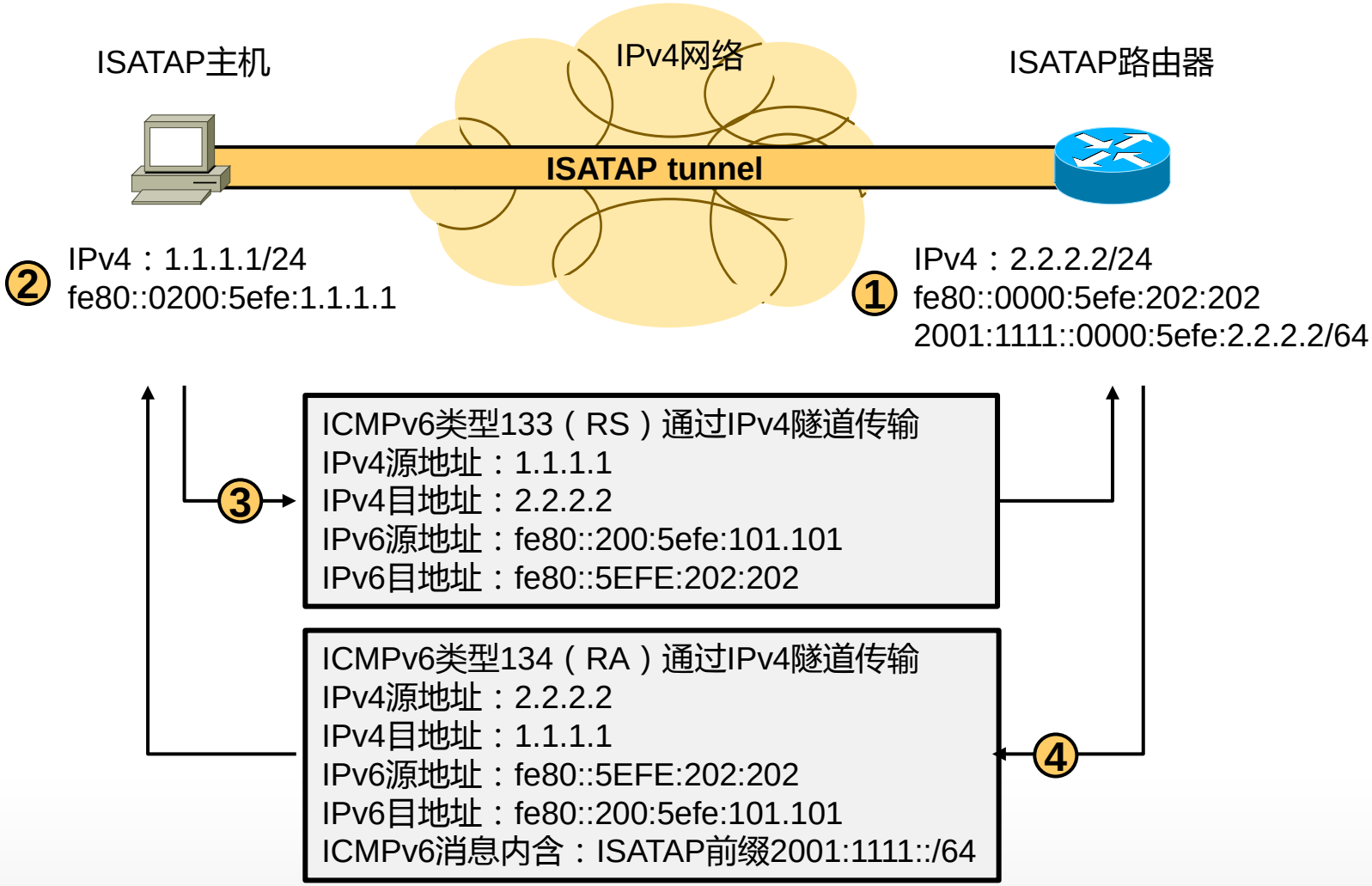
ISATAP隧道工作机制 -1



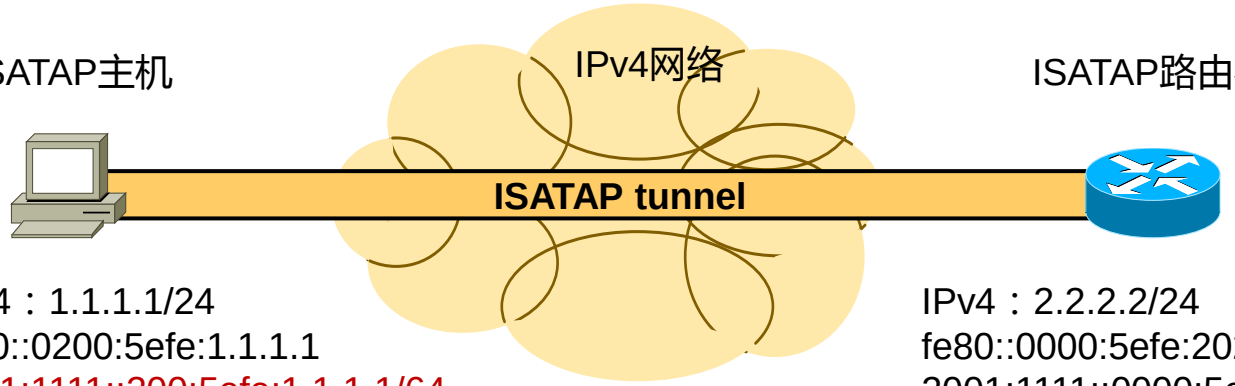
ISATAP隧道工作机制 -2

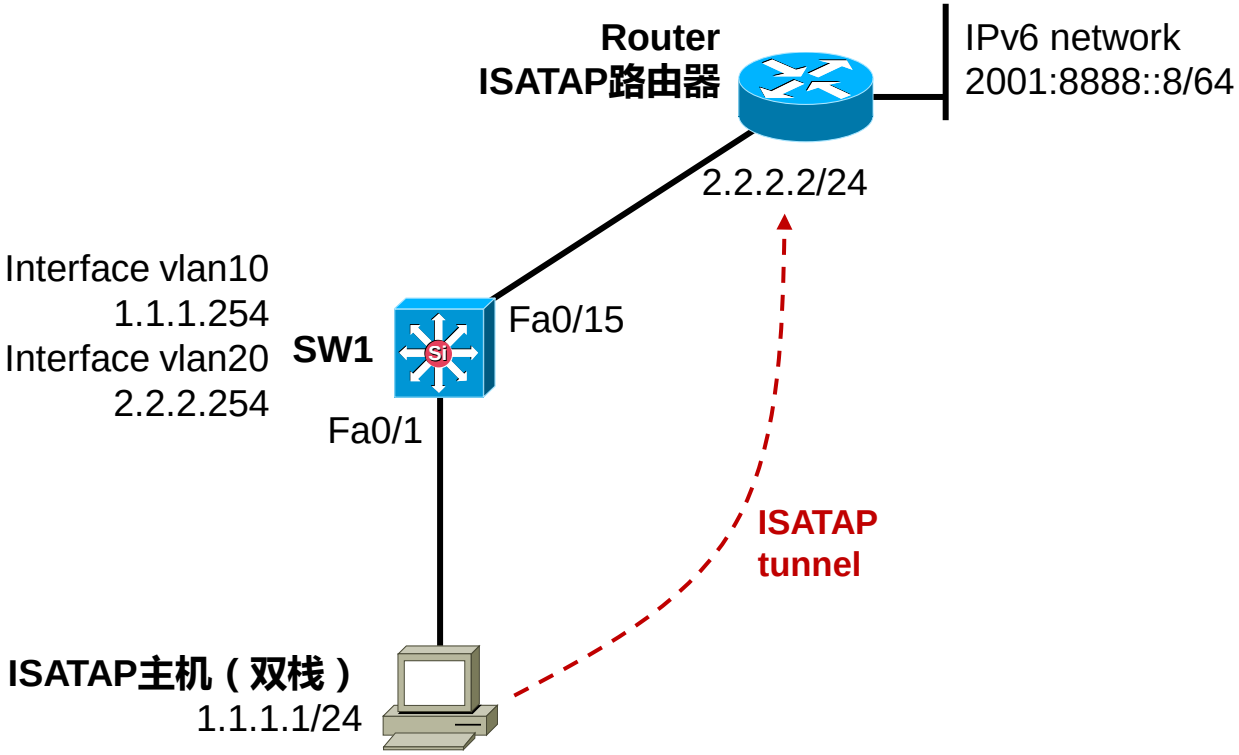


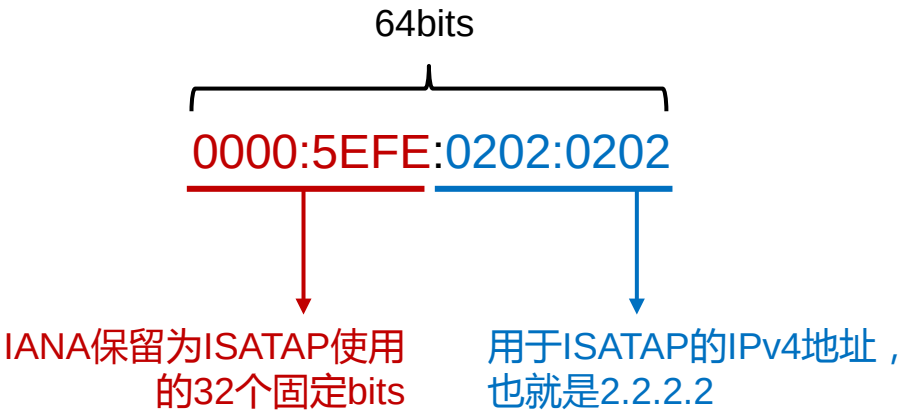
ISATAP隧道工作机制 -3

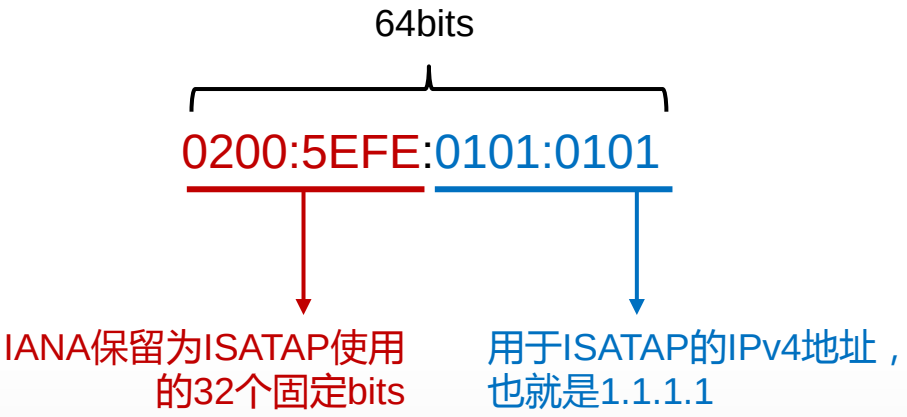


ISATAP隧道工作机制 -4





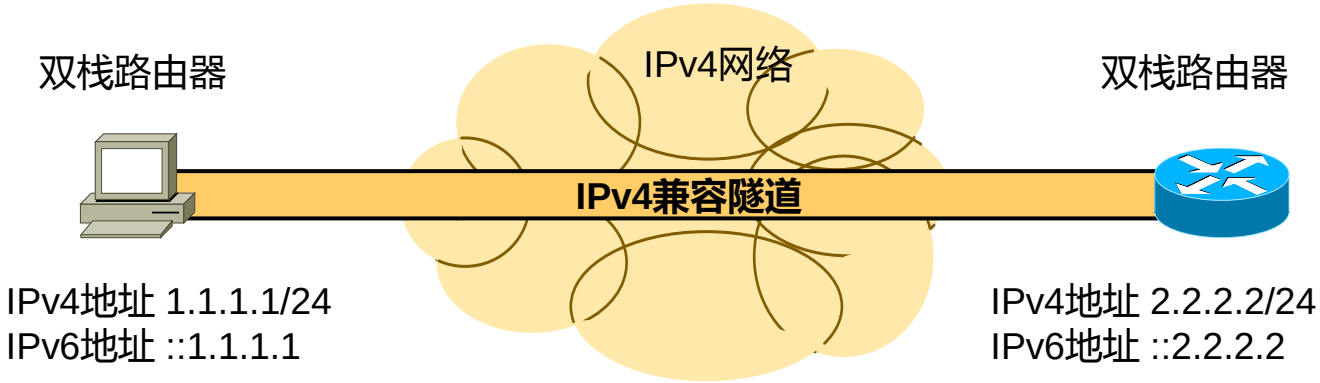

64bits
0000:5EFE:0202:0202
IANA保留为ISATAP使用
的32个固定bits
用于ISATAP的IPv4地址，
也就是2.2.2.2


64bits
0200:5EFE:0101:0101
IANA保留为ISATAP使用
的32个固定bits
用于ISATAP的IPv4地址，
也就是1.1.1.1

ISATAP隧道

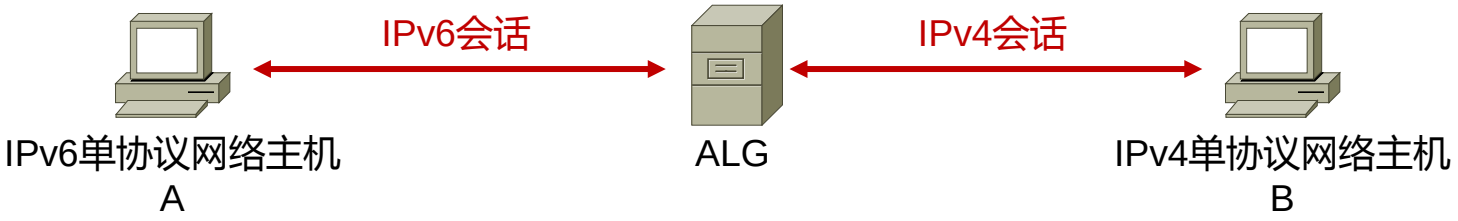
PC	netsh interface ipv6 isatap set router 2.2.2.2

Router	ipv6 unicast-routing
	interface FastEthernet0/0
	  ip address 2.2.2.2 255.255.255.0
	  no shutdown
	interface loopback0
	  ipv6 enable
	  ipv6 address 2001:8888::8/64
	!
	interface Tunnel1
	  ip unnumbered fastEthernet 0/0
	!! 这个IPv4地址就是ISATAP隧道的目的地址
	  ipv6 enable
	  ipv6 address 2001:1111::/64 eui-64
	!! 这个IPv6地址的前缀会被通告给ISATAP主机
	  no ipv6 nd suppress-ra
	  tunnel source fastEthernet 0/0
	  tunnel mode ipv6ip isatap
	!
	ip route 0.0.0.0 0.0.0.0 2.2.2.254

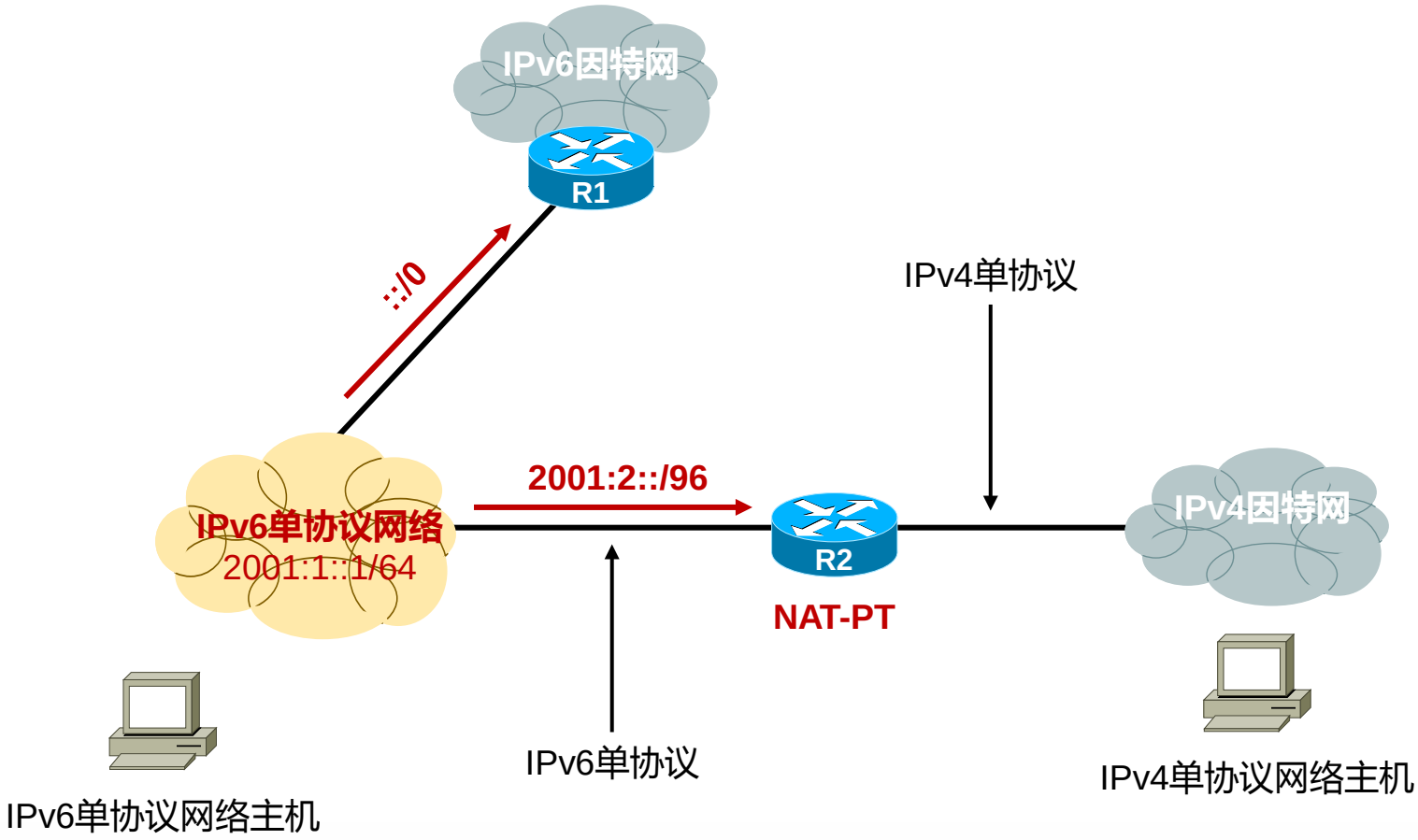

双栈路由器
IPv4网络
双栈路由器
IPv4兼容隧道
IPv4地址 1.1.1.1/24
IPv6地址 ::1.1.1.1
IPv4地址 2.2.2.2/24
IPv6地址 ::2.2.2.2

ALG

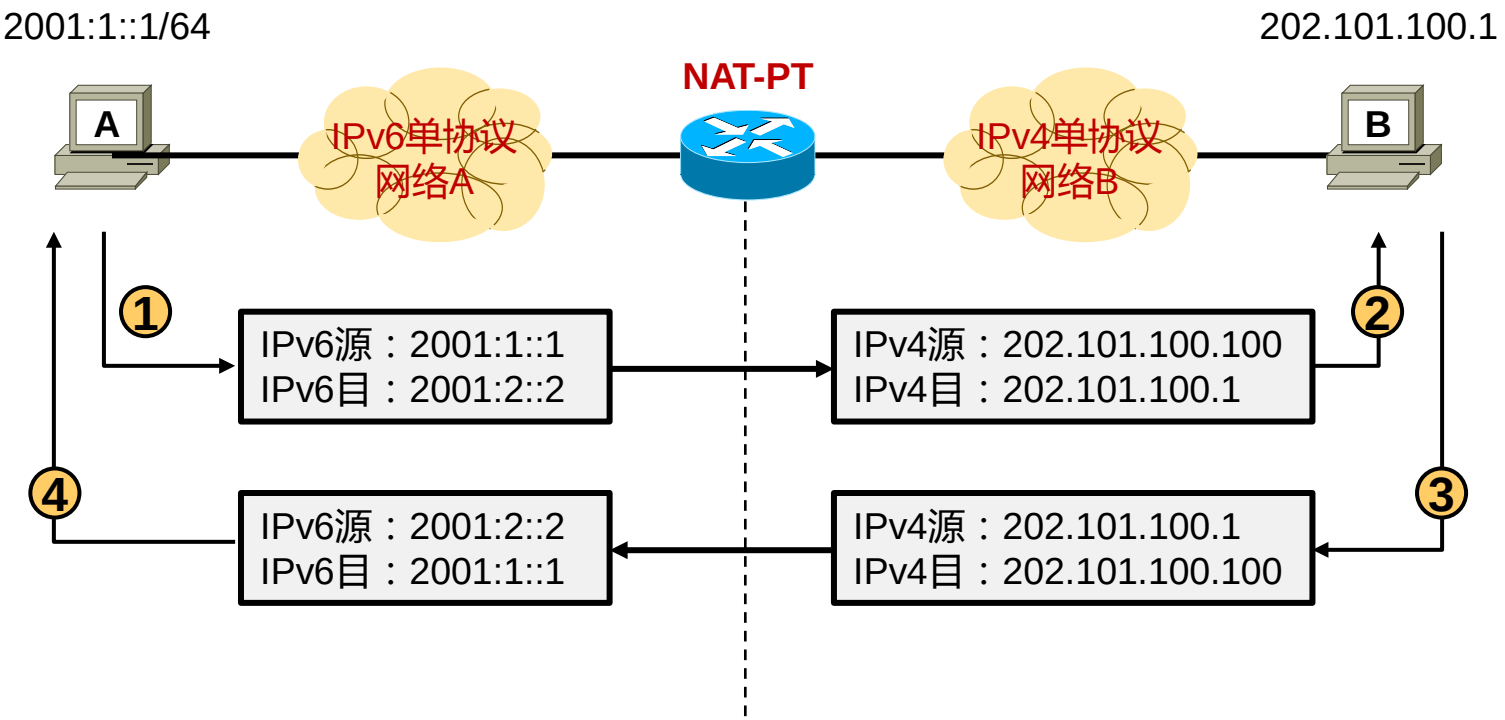
支持IPv4及V6的应用



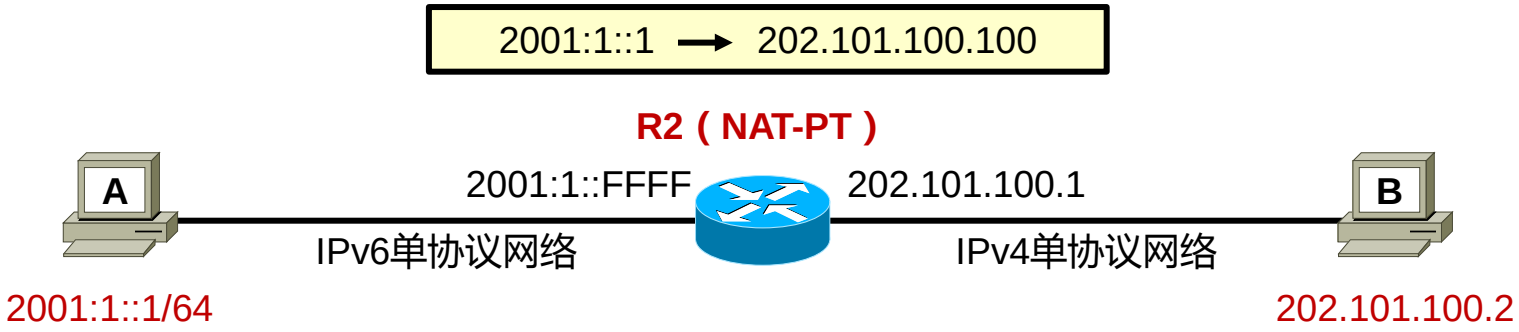
NAT-PT机制概述



NAT-PT操作

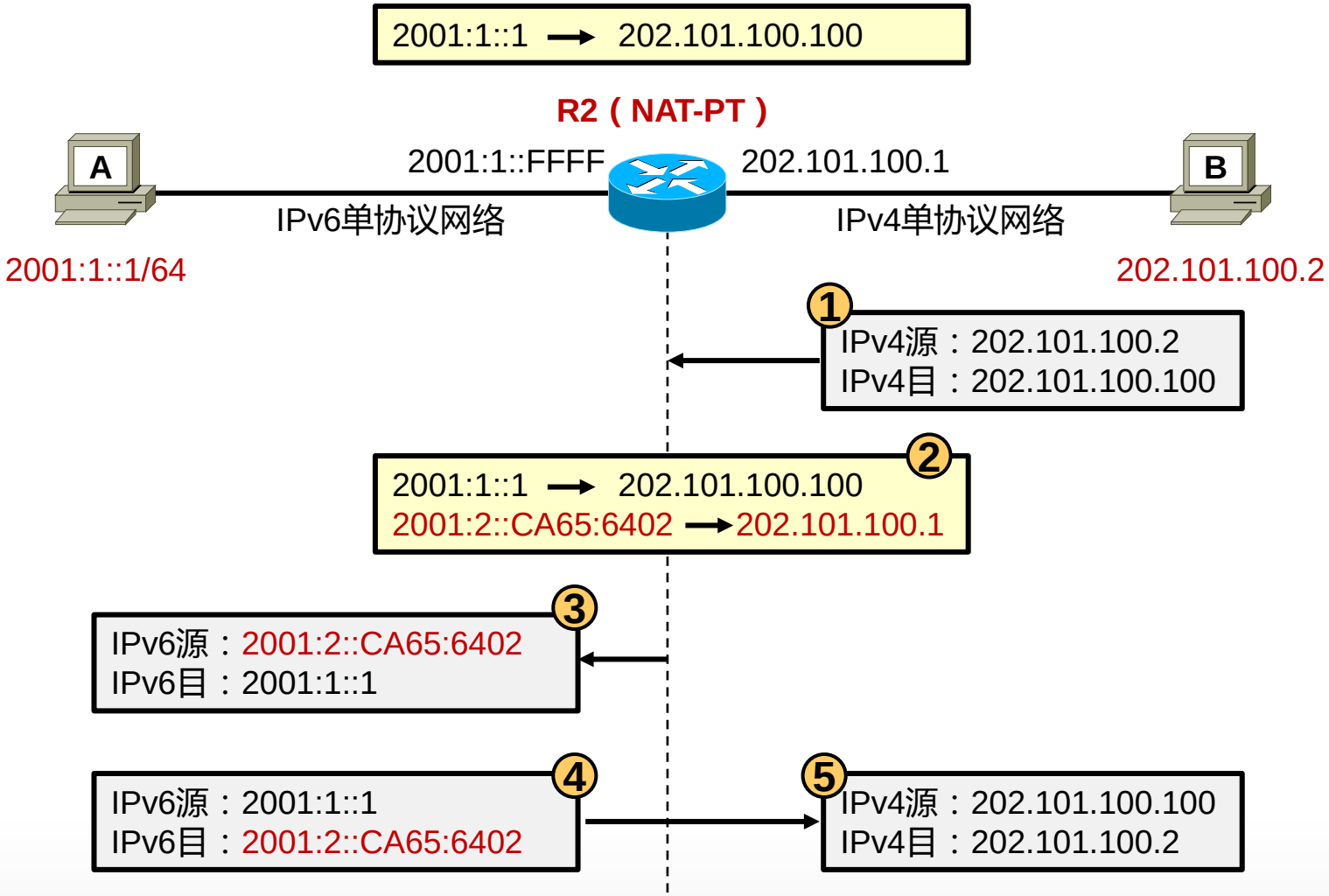


静态NAT-PT映射（单向）

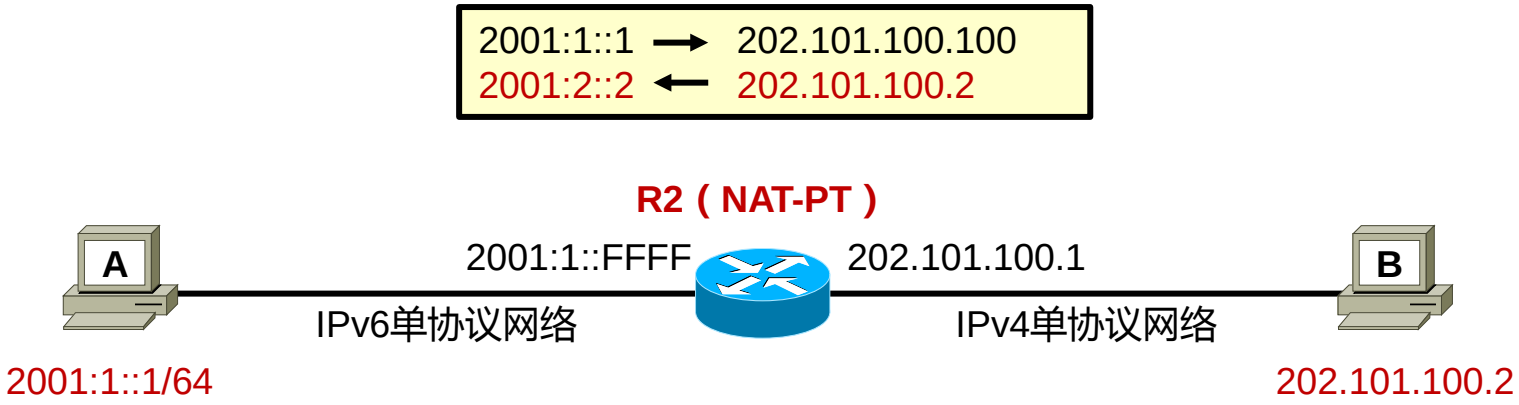



```
ipv6 unicast-routing
!
interface FastEthernet0/0
  ipv6 enable
  ipv6 address 2001:1::FFFF/64
  ipv6 nat
!
interface FastEthernet1/0
  ip address 202.101.100.1 255.255.255.0
  ipv6 nat
!
ipv6 nat prefix 2001:2::/96
ipv6 nat v6v4 source 2001:1::1 202.101.100.100
```

静态NAT-PT映射（单向 cont.）

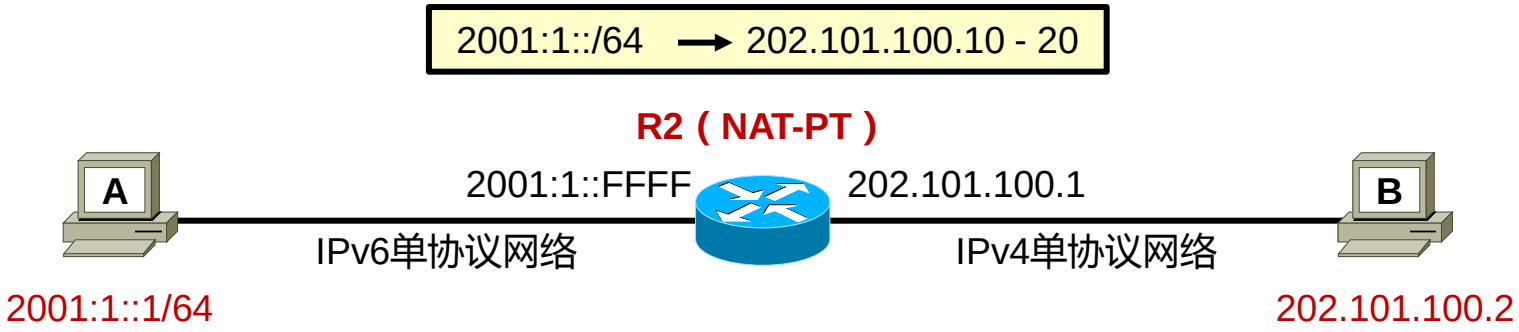


静态NAT-PT映射（双向）



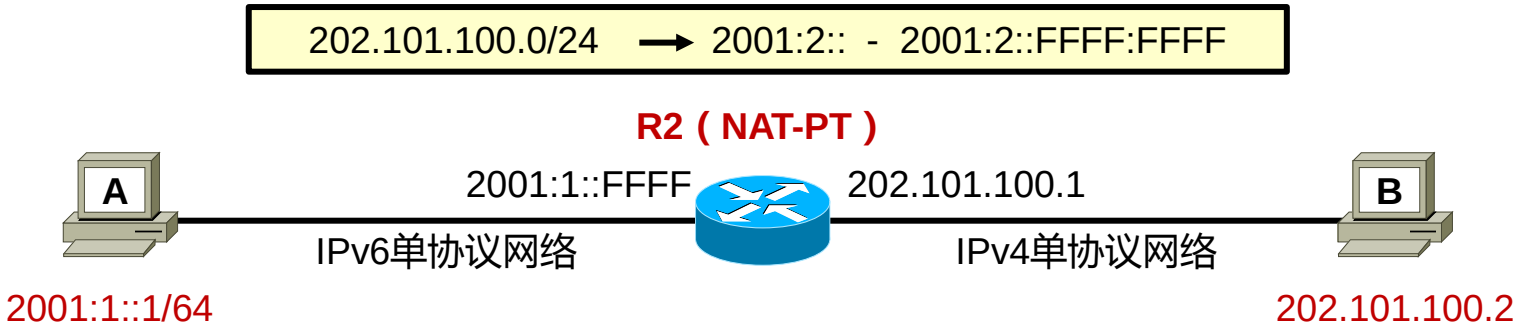

```
ipv6 unicast-routing
!
interface FastEthernet0/0
  ipv6 enable
  ipv6 address 2001:1::FFFF/64
  ipv6 nat
!
interface FastEthernet1/0
  ip address 202.101.100.1 255.255.255.0
  ipv6 nat
!
ipv6 nat prefix 2001:2::/96
ipv6 nat v4v6 source 202.101.100.2 2001:2::2
ipv6 nat v6v4 source 2001:1::1 202.101.100.100
```

动态NAT-PT映射（V6主动发起）



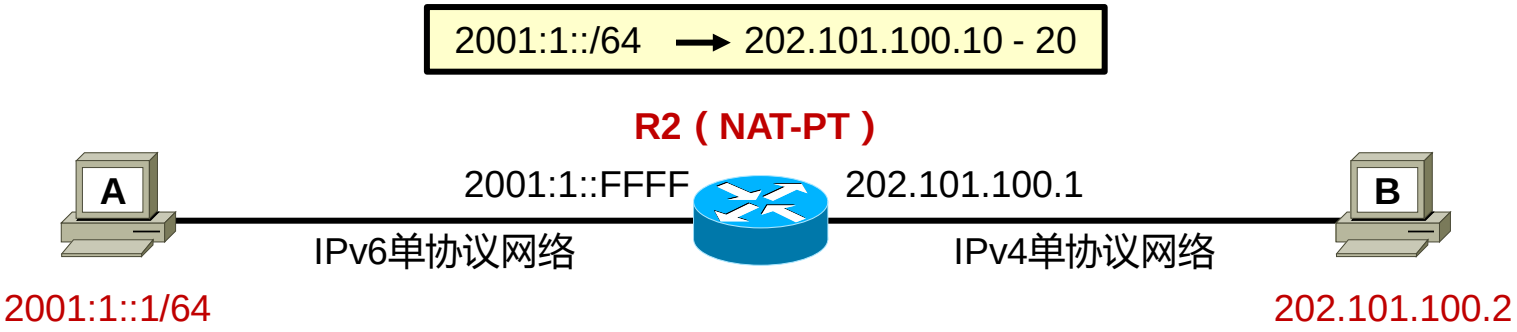

```
ipv6 unicast-routing
!
interface FastEthernet0/0
  ipv6 enable
  ipv6 address 2001:1::FFFF/64
  ipv6 nat
!
interface FastEthernet1/0
  ip address 202.101.100.1 255.255.255.0
  ipv6 nat
!
Ipv6 access-list ipv6only-network permit 2001:1::/64 any
ipv6 nat prefix 2001:2::/96
ipv6 nat v6v4 pool v6v4-pool 202.101.100.10 202.101.100.20 prefix-length 24
ipv6 nat v6v4 source list ipv6only-network pool v6v4-pool
ipv6 nat v4v6 source 202.101.100.2 2001:2::2
```

动态NAT-PT映射（V4主动发起）




```
ipv6 unicast-routing
!
interface FastEthernet0/0
  ipv6 enable
  ipv6 address 2001:1::FFFF/64
  ipv6 nat
!
interface FastEthernet1/0
  ip address 202.101.100.1 255.255.255.0
  ipv6 nat
!
access-list 1 permit 202.101.100.0 0.0.0.255
ipv6 nat v4v6 pool v4v6-pool 2001:2:: 2001:2::FFFF:FFFF prefix-length 96
ipv6 nat v4v6 source list 1 pool v4v6-pool
ipv6 nat prefix 2001:2::/96
ipv6 nat v6v4 source 2001:1::1 202.101.100.111
```

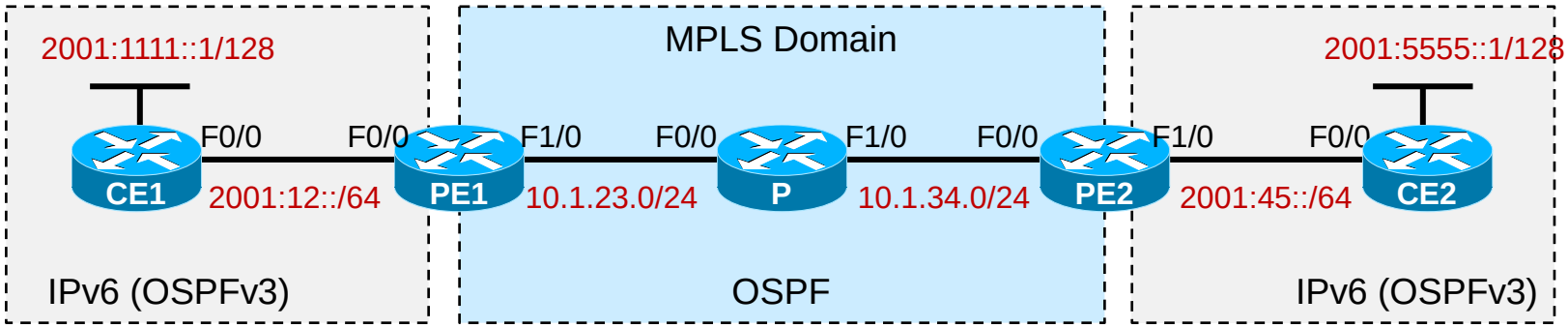
动态NAT-PT映射（V6主动发起 IPv4-Mapped NAT-PT）



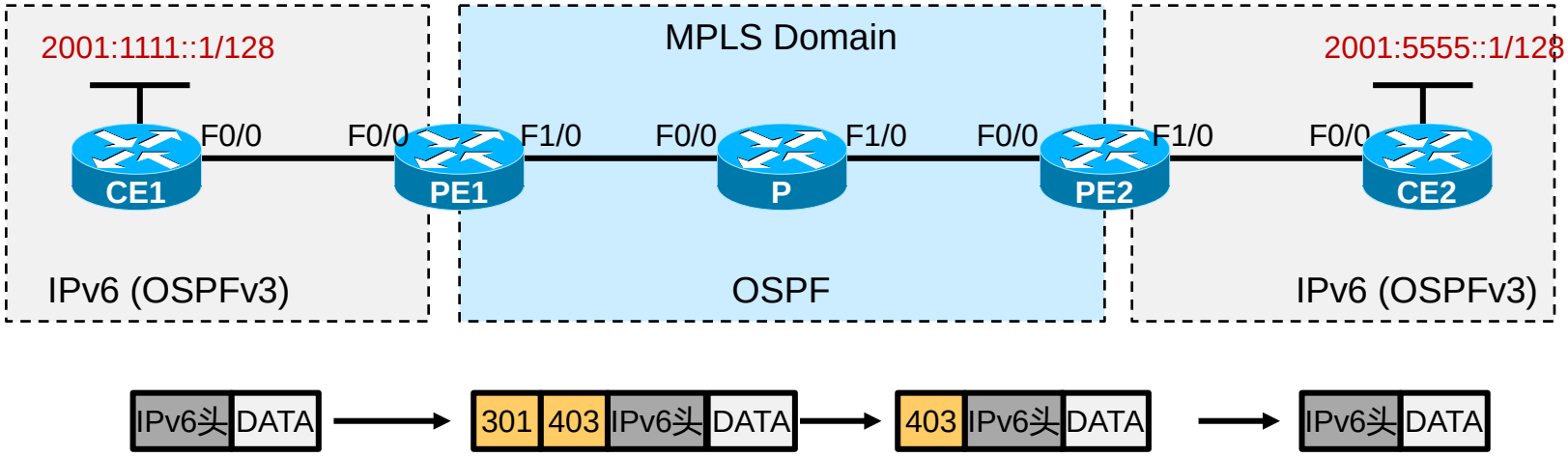

```
ipv6 unicast-routing
!
interface FastEthernet0/0
  ipv6 enable
  ipv6 address 2001:1::FFFF/64
  ipv6 nat
!
interface FastEthernet1/0
  ip address 202.101.100.1 255.255.255.0
  ipv6 nat
!
Ipv6 access-list ipv6only-network permit 2001:1::/64 any
ipv6 nat v6v4 pool v6v4-pool 202.101.100.10 202.101.100.20 prefix-length 24
ipv6 nat v6v4 source list ipv6only-network pool v6v4-pool
ipv6 access-list v4map permit 2001:1::/64 2001:2::/96
ipv6 nat prefix 2001:2::/96 v4-mapped v4map
```

6PE and 6VPE

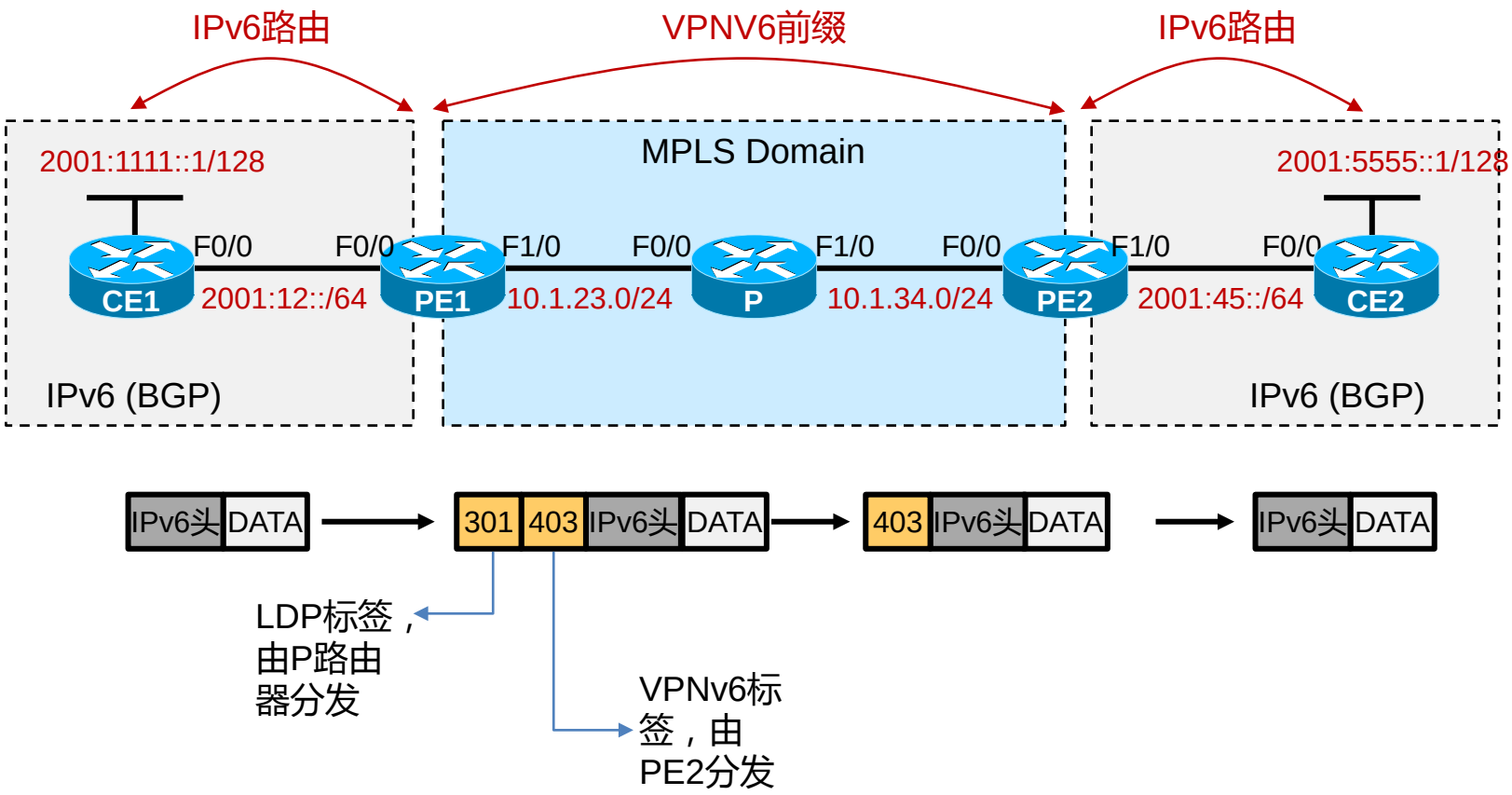
6PE



6PE



6VPE



Tea 红茶三杯
ccietea.com

沉淀 提升 成长 分享
关注@红茶三杯：weibo.com/vinsoney

Thank You