

网络技术与信息安全



第2讲：构建综合园区网

河南中医药大学信息技术学院
许成刚

本章教学目标

- 掌握路由器的工作原理（互联、转发）
- 熟悉路由表各字段含义
- 掌握直连路由、静态路由、缺省路由的配置与应用
- 理解路由器与交换机的不同



一、认识路由器

认识路由器

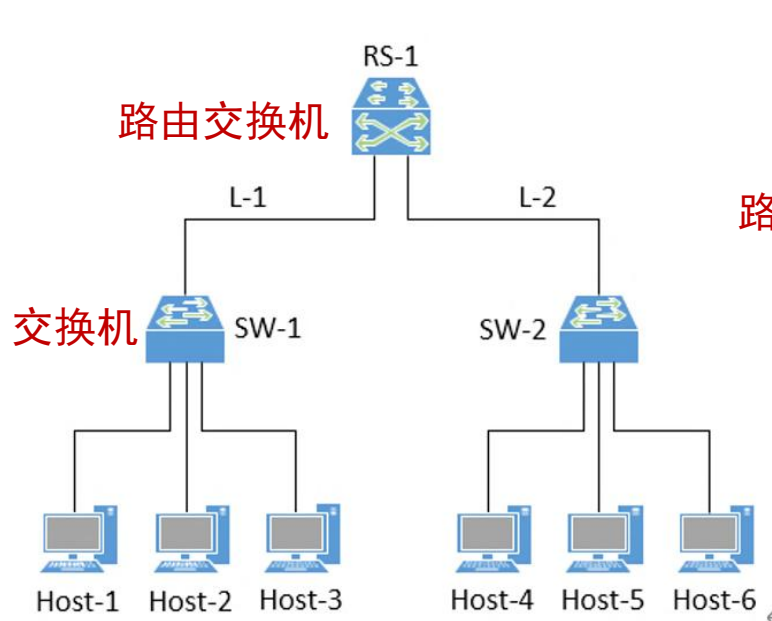
□ 有了路由交换机，还需要使用路由器吗？

■ 先理清概念：路由交换机（三层交换机）= 交换机 + 内置路由功能，很多人以为三层交换能干全部活，但它和路由器**核心定位不一样**，二者硬件、工作场景、擅长处理的业务完全不同。

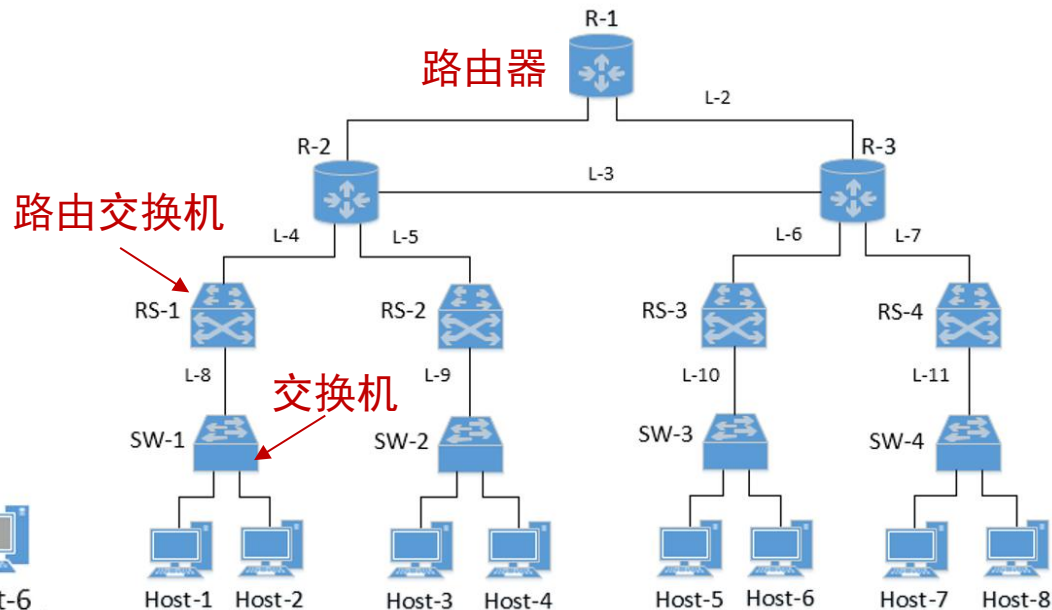
■ 核心本质区别

- **三层交换机**：主要做内网高速转发。路由功能是附加的，靠硬件芯片做 IP 转发，速度极快；但它的强项是局域网内部**不同 VLAN 之间互通**。
- **路由器**：主要做**跨网络连接（即不同网络间通信）**，连接不同种类网络（内网↔互联网、不同广域网线路），处理广域网协议、NAT、拨号、安全策略。

➤ 路由交换机 vs 路由器



路由交换机，实现不同VLAN之间的互联



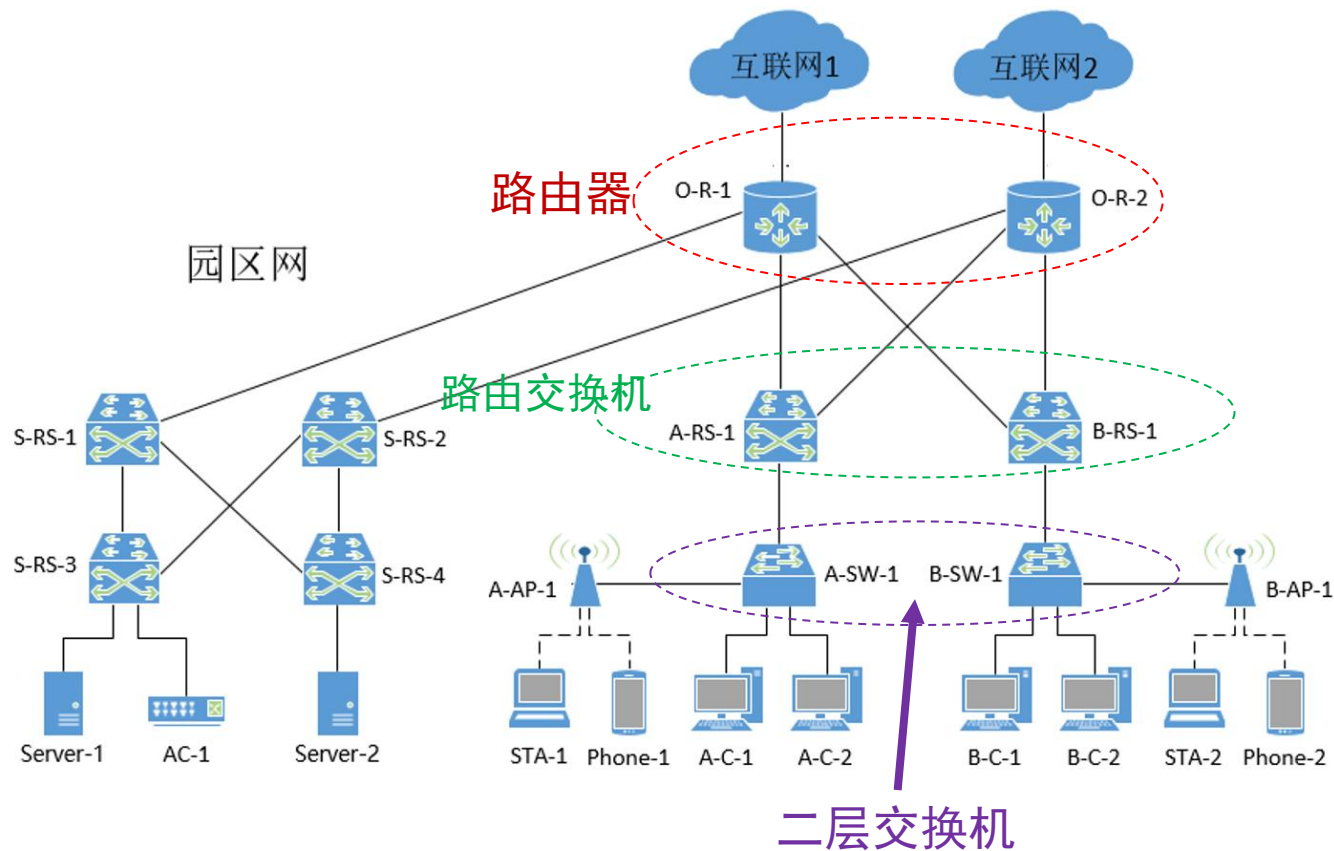
路由器，实现不同网络之间的互联，通常连接其他路由设备

► 常见的三层架构园区网

核心层：路由器，
实现接入互联网，
也可实现园区内不同网络之间的互联；

汇聚层：路由交换机，
实现其下方不同VLAN间通信；

接入层：交换机，
配置VLAN接口，实现用户主机的接入



认识路由器

□ 路由器的基本功能

■ 路由器(Router)，是实现网络互联的核心设备。

- 路由器也是一个**多接口**的网络设备，可将不同（不同网段或不同类型）网络互联起来，实现不同网络间的报文转发。也被称作**网际互连设备**。
- 核心功能：**互联**（不同网络）、**转发**（在不同网络间转发报文）



思科路由器



华为AR6300-S万兆路由器

认识路由器

先从路由器接口的角度，谈谈如何把不同网络互联起来

认识路由器

□ 先认识路由器的接口（实现网络互联）

■ 接口类型

- 路由器具有非常强大的网络连接和路由功能，它可以与各种各样的不同网络进行物理连接，这就决定了路由器的接口技术非常复杂；
- 路由器的接口主要包括
 - 管理配置接口
 - 局域网接口
 - 广域网接口

认识路由器

□ 先认识路由器的接口（实现网络互联）

■ 接口类型 —— 管理配置接口

- Console口：管理员可通过Console口对路由器进行配置管理；
- 通过console配置线连接配置电脑和路由器设备，采用串口协议方式配置。



认识路由器

□ 先认识路由器的接口（实现网络互联）

■ 接口类型 —— 与局域网的接口

➤ 以太网接口

- RJ-45接口（FE、GE）

➤ 光纤接口

- SC口等，用于光纤介质连接，常用于高速局域网



GE接口



FE接口

认识路由器

□ 先认识路由器的接口（实现网络互联）

■ 接口类型 —— 与广域网的接口

- **Ethernet接口**：以太网接口，例如GE接口
- **serial接口（高速同步串口）**：传输方式、支持的速度、使用的协议都与Ethernet接口不相同。Serial口多封装 PPP协议或Frame-Relay协议，用于广域网连接。
- **POS接口**： POS(Packet Over SONET/SDH) 技术支持光纤介质，它是一种高速、先进的广域网连接技术。在路由器上插入一块POS模块，路由器就可以提供POS接口。
-

认识路由器

□ 先认识路由器的接口（实现网络互联）

- 说明：本课程中，关于路由器接口的配置，都是针对以太网接口。

认识路由器

□ 先认识路由器的接口（实现网络互联）

■ 路由器接口的MAC地址与IP地址

- 路由器上的每个以太网接口，都具有独立的MAC地址。（对比交换机？）
- 不仅如此，在使用过程中，管理员还要根据联网需要，给路由器的每个以太网接口配置独立的IP地址。（对比交换机）
- 由于路由器的每个网络接口要连接不同的网络，因此路由器每个网络接口的IP地址不仅不能一样，而且必须具有不同的网络号。
- 因此，路由器的每个接口，就是一个独立的广播域。（对比交换机）
- 路由器必须经过配置才能正常工作，其中一项重要的配置操作，就是给路由器接口配置IP地址。

认识路由器

□ 先认识路由器的接口（实现网络互联）

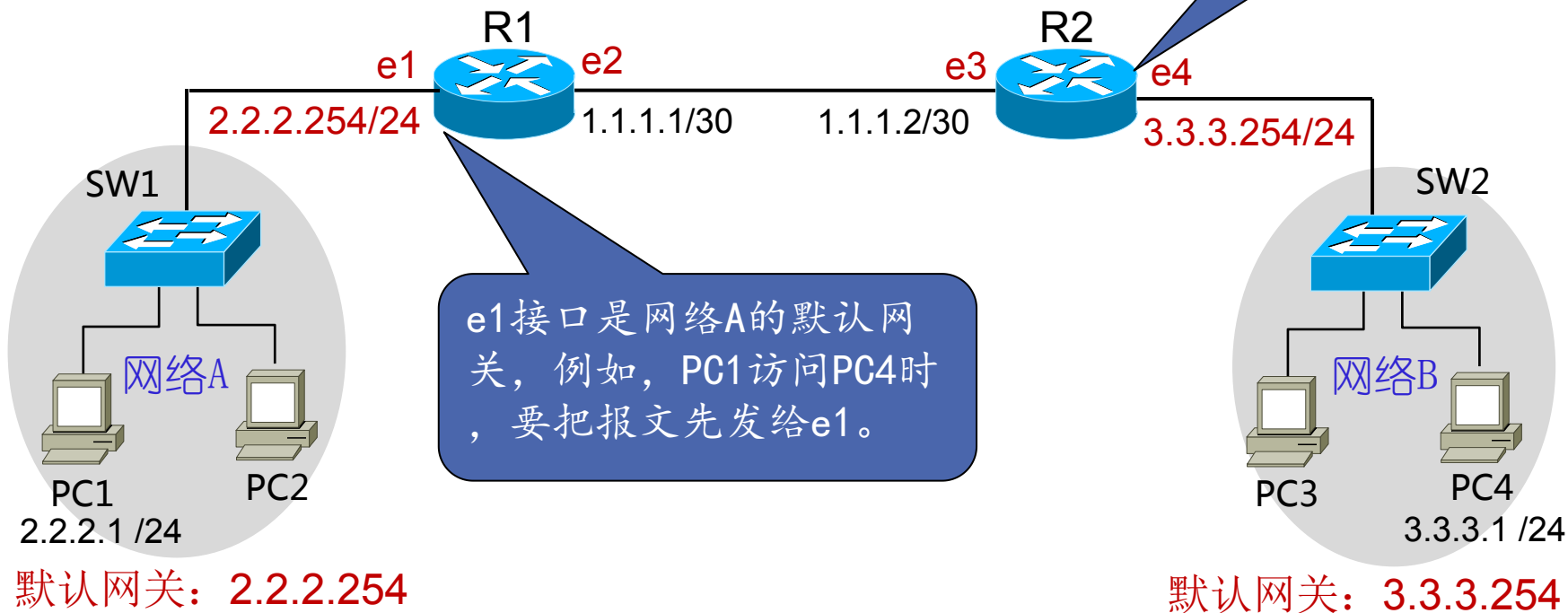
■ 应用1：作为所接入网络的默认网关

- 【复习】当源主机与目的主机不在同一网络时（IP地址的网络号不同），源主机无法将数据包直接发送给目的主机，而是先把数据包发给自己的默认网关。然后，由默认网关进行后续转发。
- 路由器接口可连接不同的网络，并且具有独立的MAC地址和IP地址，因此，可以将路由器接口作为其直连网络的默认网关。即直连网络中的各主机要将自己的默认网关地址，设置成路由器接口的IP地址。
- 注意：主机的IP地址一定要与自己默认网关的IP地址在同一网段，否则无法将数据包直接发给默认网关（提示：ARP的作用）
- 举例

认识路由器

□ 先认识路由器的接口（实现网络互联）

■ 作为所接入网络的默认网关



□ 配置主机的默认网关

- 在Windows操作系统中配置主机IP地址和默认网关。



□ 配置主机的默认网关

- 在Linux操作系统中，使用vi命令编辑网卡配置文件（此处为ifcfg-enp0s3），从而配置主机的IP地址和默认网关

```
#vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
```

```
TYPE=Ethernet
```

```
PROXY_METHOD=none
```

```
BROSWSET_ONLY=no
```

```
BOOTPROTO=static
```

```
DEFROUTE=yes
```

```
.....
```

```
DEVICE=enp0s3
```

```
ONBOOT=yes
```

```
IPADDR=192.168.100.200      本机IP地址
```

```
NETMASK=255.255.255.0
```

```
GATEWAY=192.168.100.254      默认网关地址
```

```
ONBOOT=yes
```

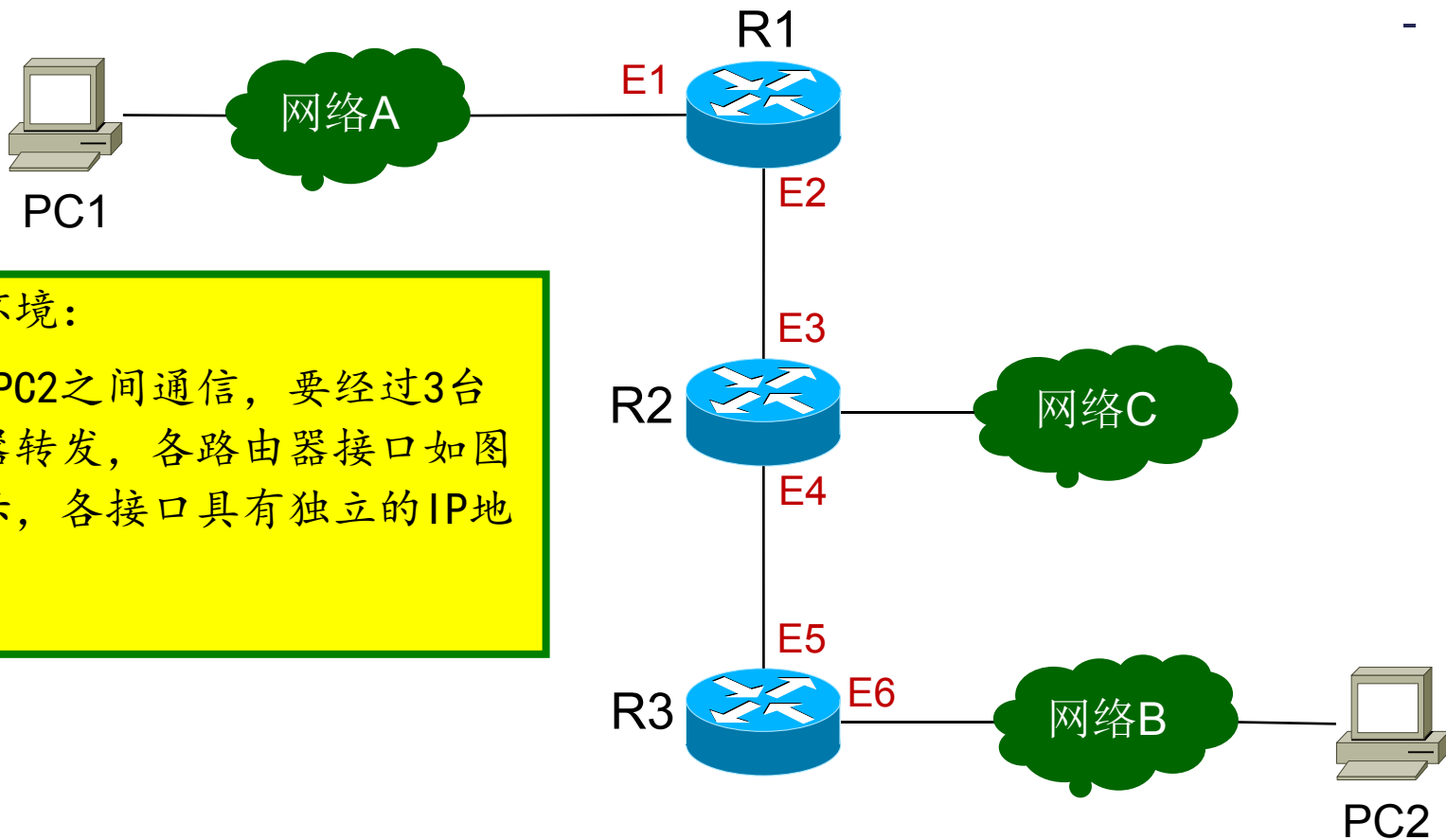
认识路由器

□ 先认识路由器的接口（实现网络互联）

■ 应用2：作为通信过程中，数据包的“下一跳”

- 当通信过程要经过多个路由器，即默认网关无法将收到的数据包直接转发到目的网络时，路由器（依据路由表）可以将数据包转发至与自己相邻的、另一台路由器的接口，然后由该路由器完成后续的转发工作。
- 这里的“相邻的、另一个路由器的接口”，就是该数据包下一步要到达的地方，即“下一跳”。
- 举例

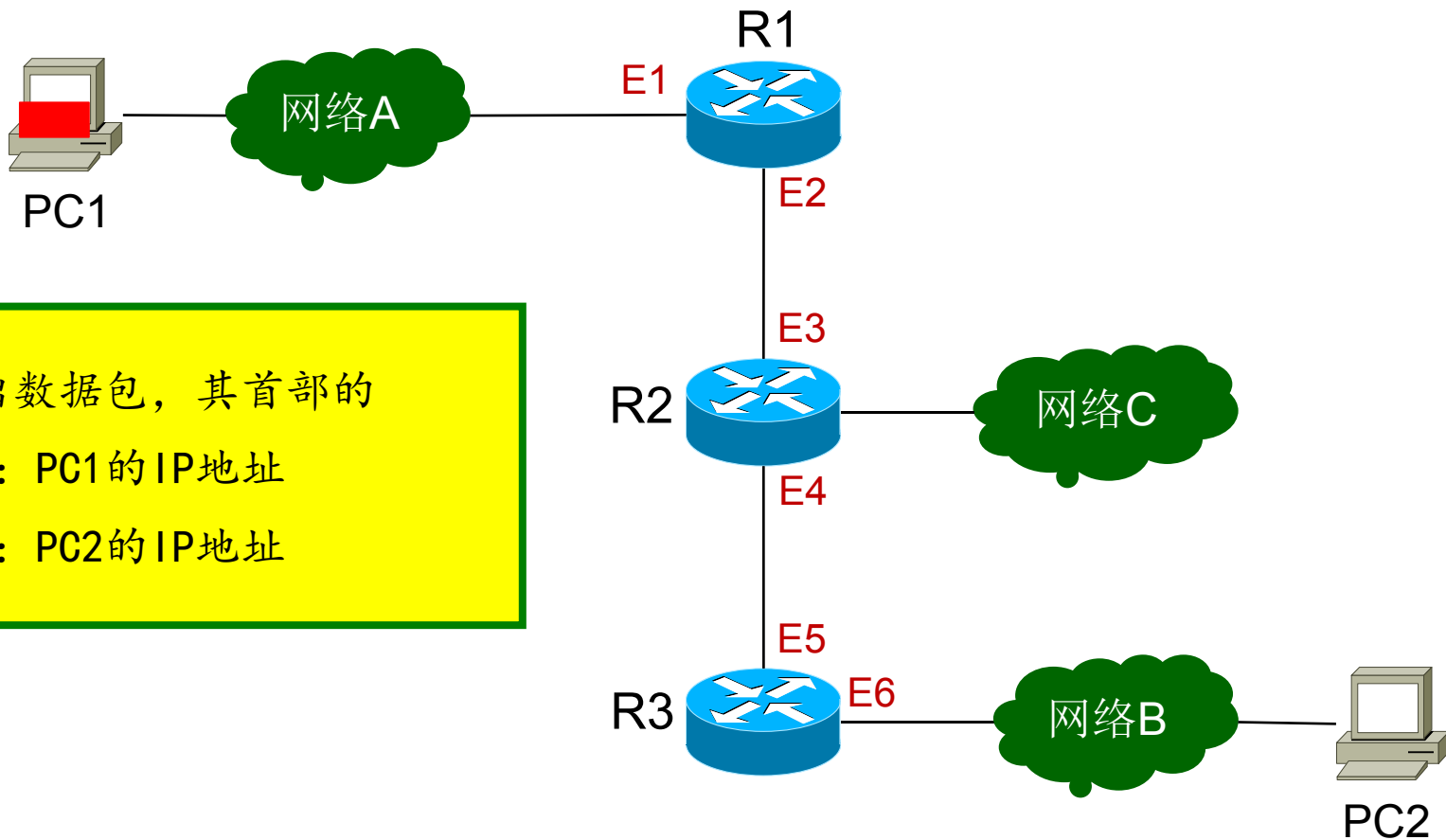
◆ 动画演示：路由器接口作为“下一跳”



联网环境：

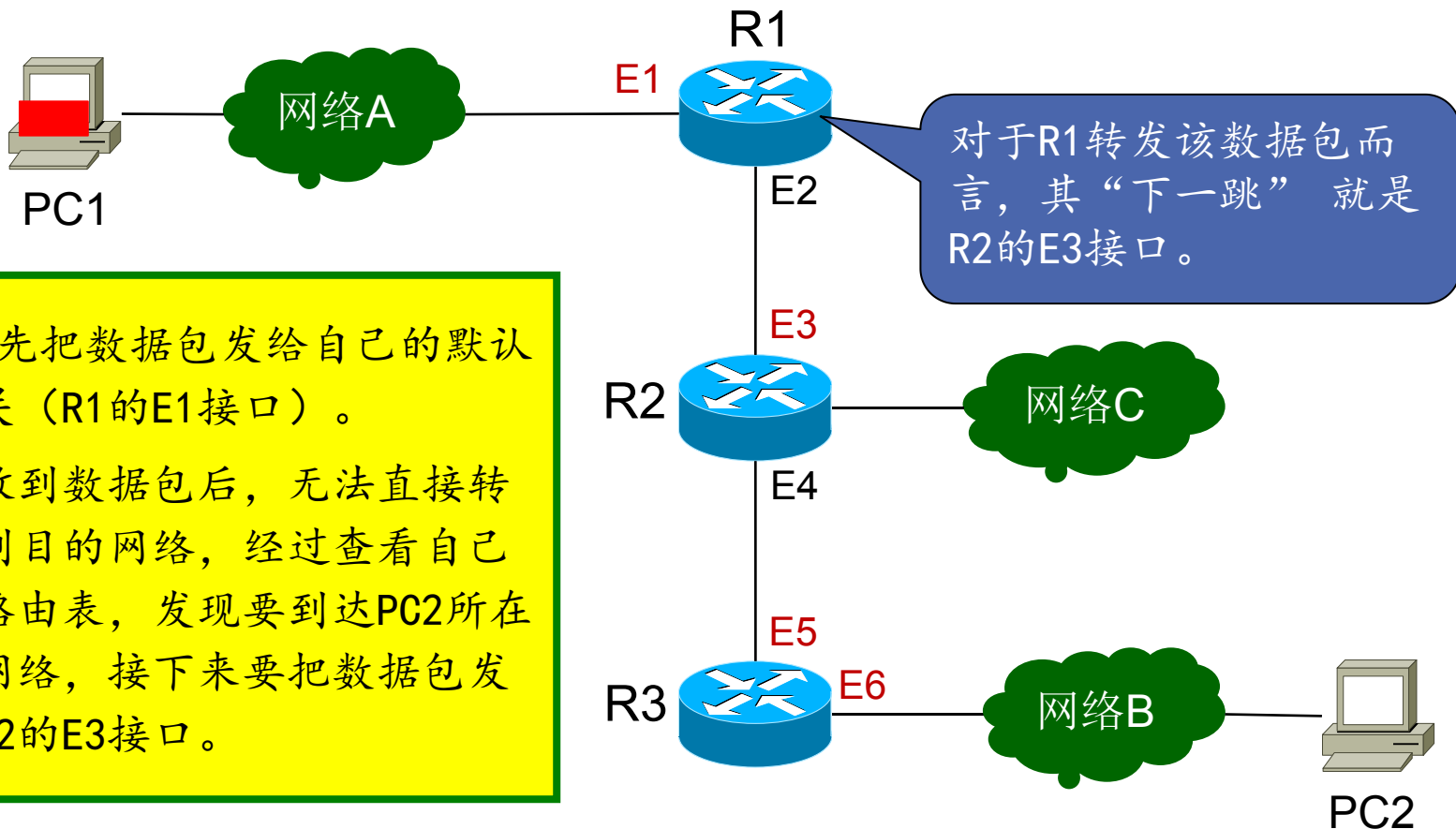
PC1和PC2之间通信，要经过3台路由器转发，各路由器接口如图中所示，各接口具有独立的IP地址。

◆ 动画演示：路由器接口作为“下一跳”



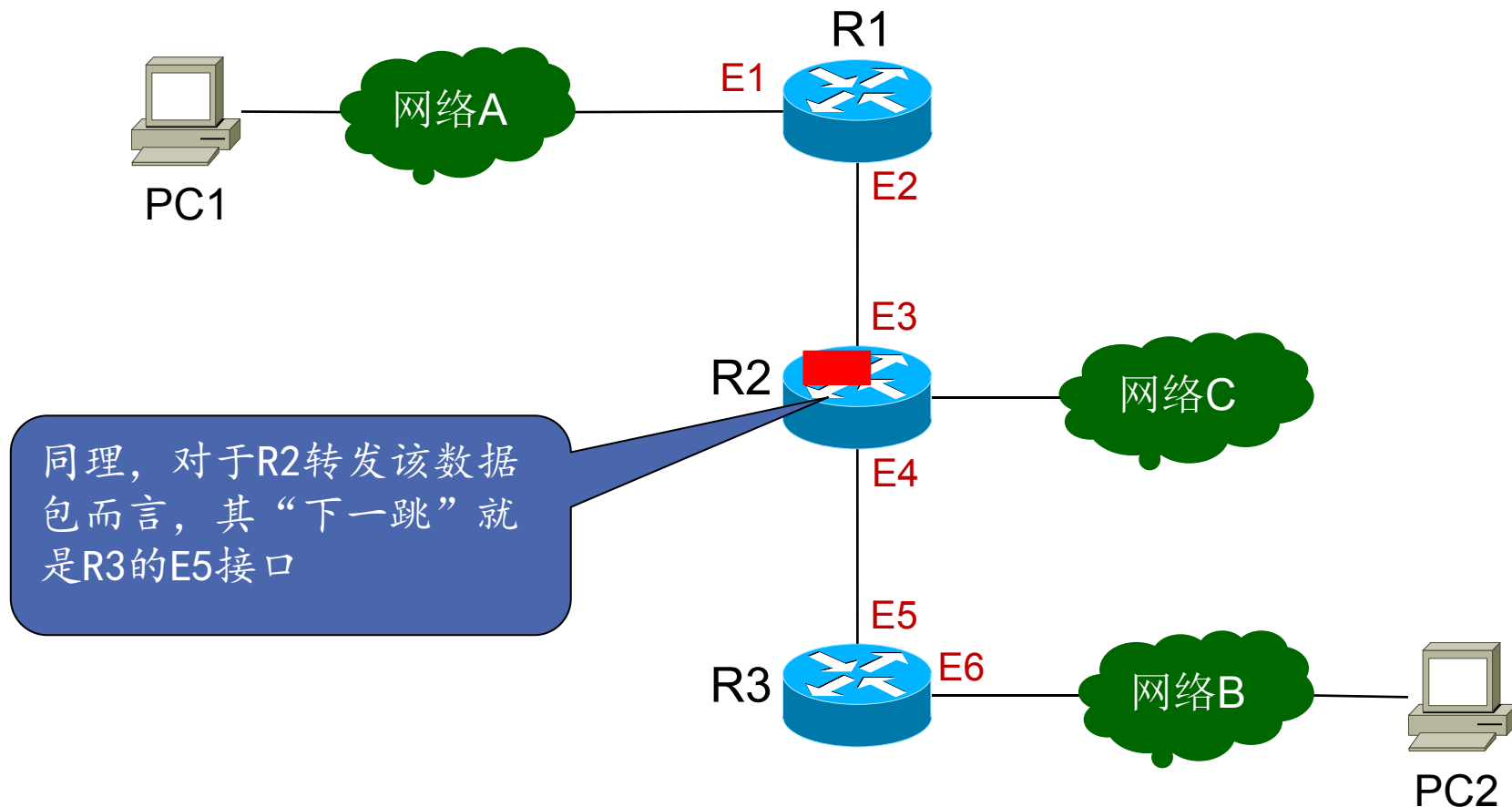
PC1发出数据包，其首部的
源IP：PC1的IP地址
目的IP：PC2的IP地址

◆ 动画演示：路由器接口作为“下一跳”

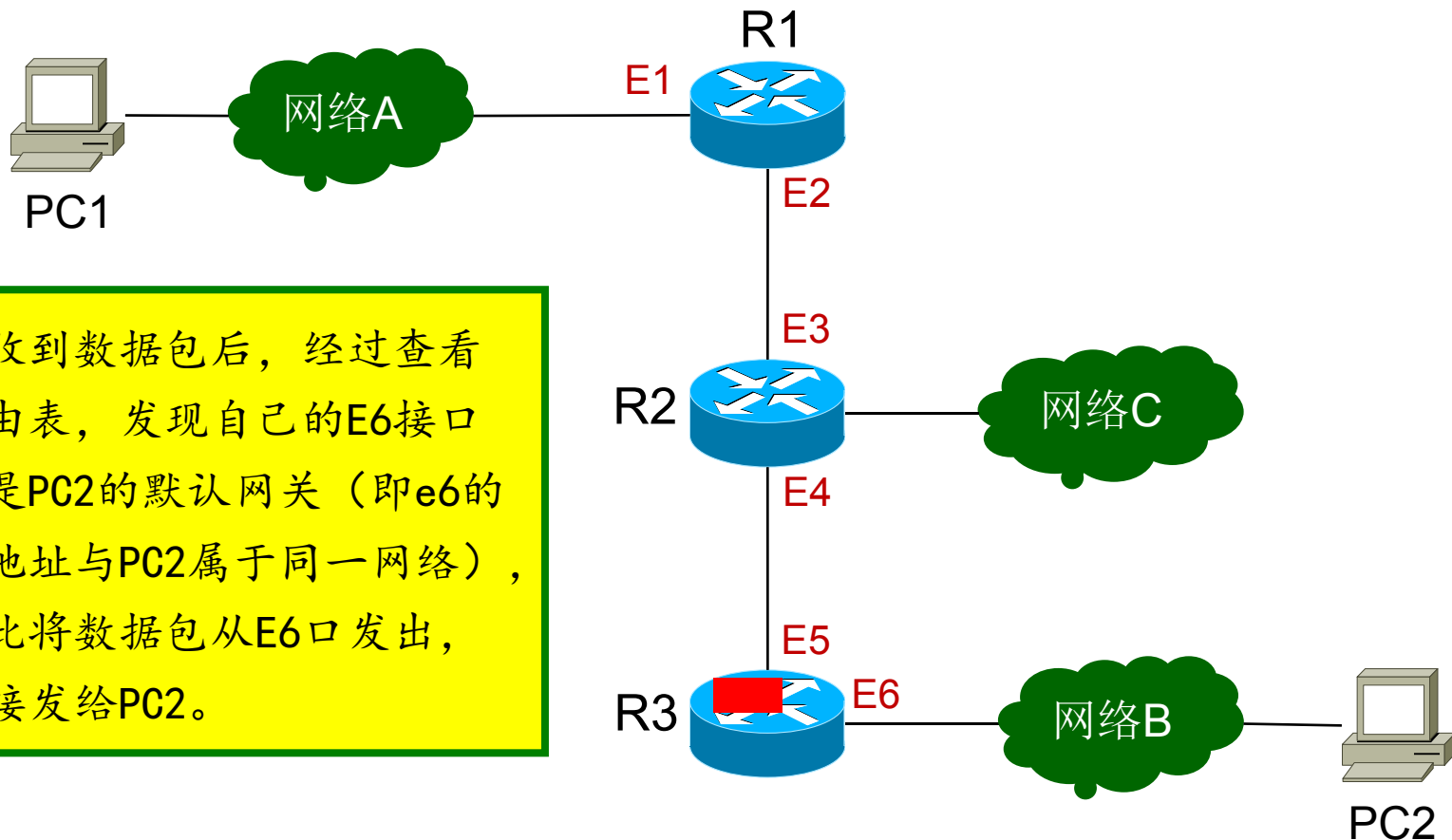


1. PC1先把数据包发给自己的默认网关（R1的E1接口）。
2. R1收到数据包后，无法直接转发到目的网络，经过查看自己的路由表，发现要到达PC2所在的网络，接下来要把数据包发给R2的E3接口。

◆ 动画演示：路由器接口作为“下一跳”



◆ 动画演示：路由器接口作为“下一跳”



3. R3收到数据包后，经过查看路由表，发现自己的E6接口就是PC2的默认网关（即e6的IP地址与PC2属于同一网络），因此将数据包从E6口发出，直接发给PC2。

认识路由器

□ 先认识路由器的接口（实现网络互联）

■ 应用2：作为通信过程中，数据包的“下一跳”

➤ 总结：（一台）路由器**通常**并不负责把报文“送到家”，而是只负责把报文转发到正确的“下一跳”，简单为“跳跃式前进”！

➤ 问：为何是“通常”？

- 若路由器的某个接口正是目的主机的默认网关，则路由器接口直接把报文发给目的主机。



认识路由器

□ 先认识路由器的接口（实现网络互联）

■ 应用3: 对报文进行“解封装”和“封装”

- 路由器属于“三层设备”，包含下三层功能。所以路由器接口通常也被称作“三层接口”，它既可以识别报文首部的MAC地址，也可以识别报文首部的IP地址。
- 当路由器收到一个报文后，接口的数据链路层和网络层会分别读取报文首部的目的MAC地址和目的IP地址，然后进行相应的处理。

认识路由器

□ 先认识路由器的接口（实现网络互联）

■ 应用3: 对报文进行“解封装”和“封装”

➤ 解封装（接收报文）

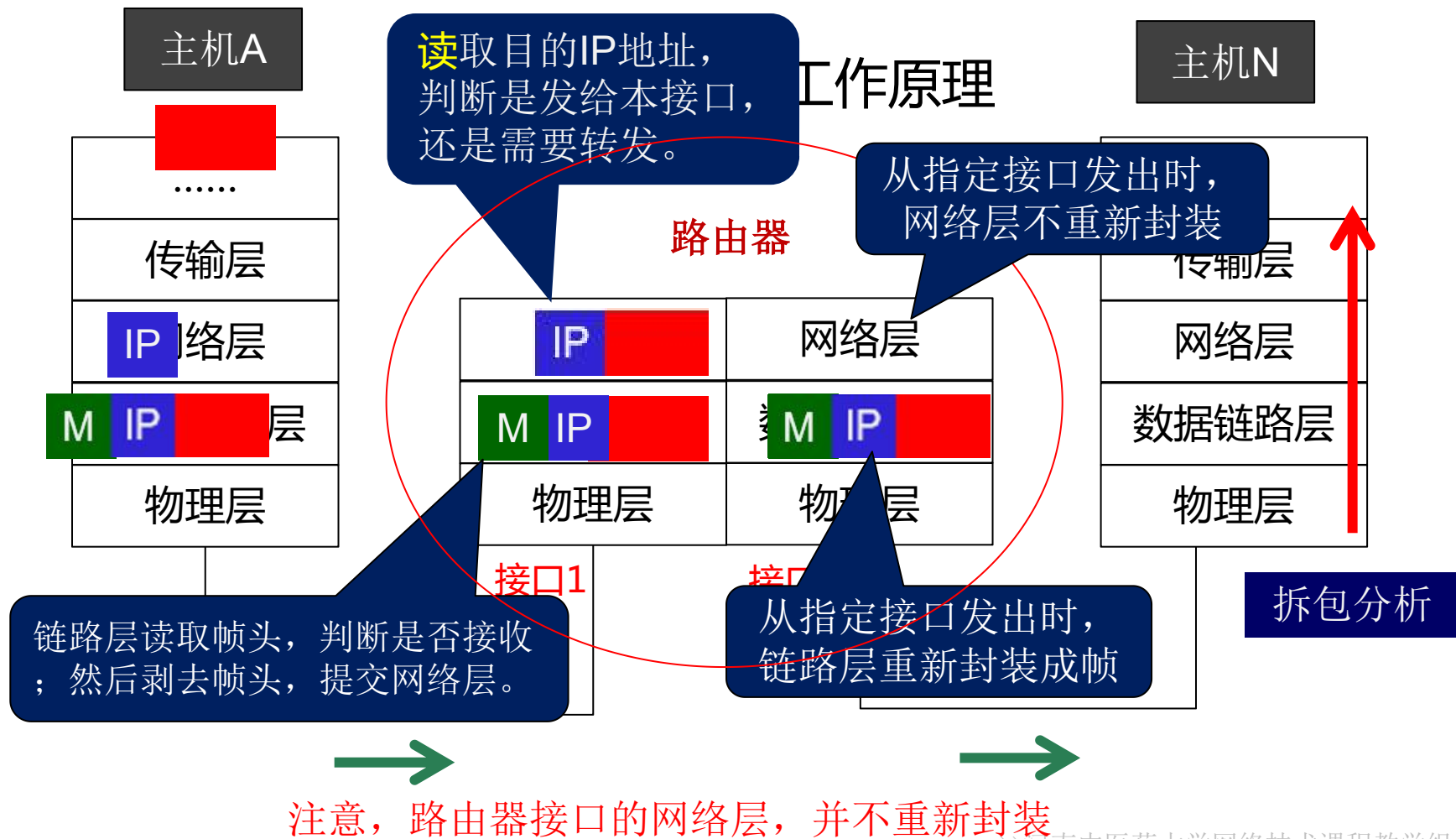
- 接收数据帧：从物理接口(如以太网口)接收数据链路层帧(如以太网帧)
- 解封装至网络层：剥离数据链路层头部(MAC地址)，提取网络层数据包(IP包)
- 检查目的IP地址：查看IP包首部的目的IP地址，确定是本接口IP或需转发；

➤ 封装（转发报文）

- 重新封装数据帧：根据出接口类型(如以太网、PPP)，添加新的数据链路层首部
- 转发数据包：将帧发送到下一跳设备或目的主机

➤ 动画演示

◆ 动画演示：路由器接口的解封装和封装



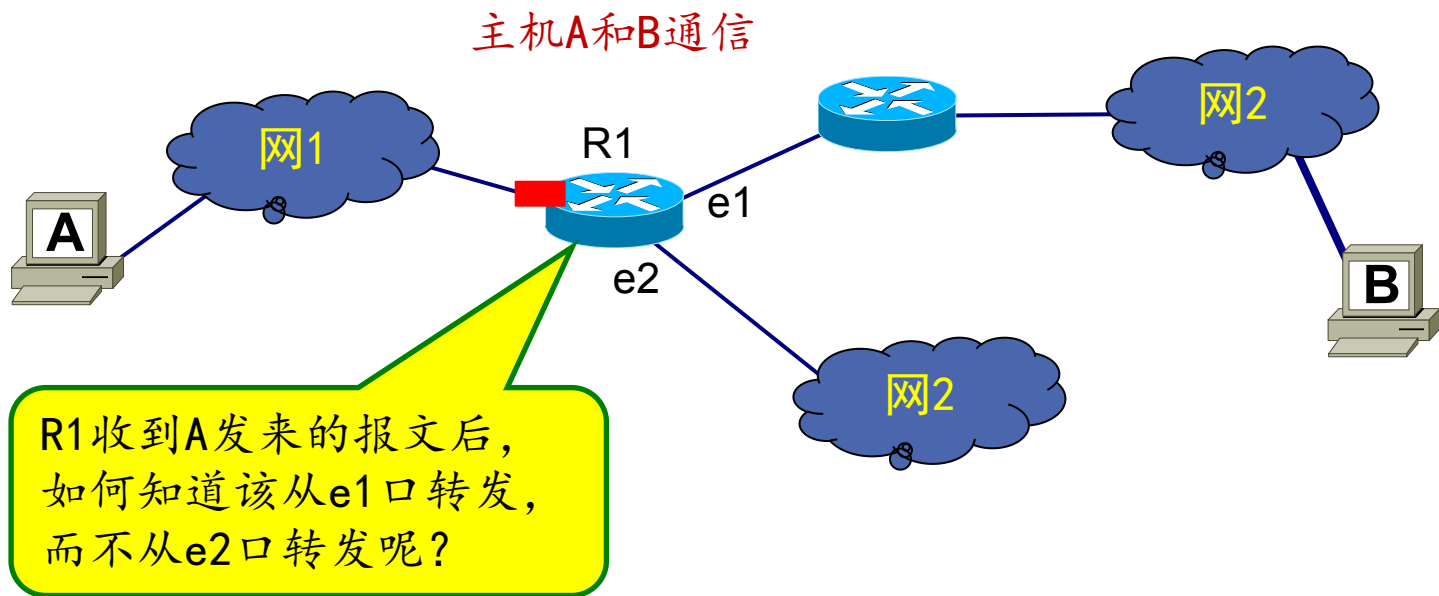
二、路由表的应用

—— 路由器收到数据包后，是如何依据路由表进行转发的？

认识路由表

□ 路由器为什么能够在不同网络间转发数据包？

■ 因为路由器中有路由表！路由表中有前往各个目的网络的路由信息！



认识路由表

□ 路由表里面有什么？

[R-2]display ip routing-table

华为路由器 **display ip routing-table**

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 16

Routes : 16

<u>Destination/Mask</u>	<u>Proto</u>	<u>Pre</u>	<u>Cost</u>	<u>Flags</u>	<u>NextHop</u>	<u>Interface</u>
10.0.0.4/30	Direct	0	0	D	10.0.0.5	GigabitEthernet0/0/1
10.0.5.0/30	Direct	0	0	D	10.0.5.2	GigabitEthernet0/0/3
172.16.1.0/24	Static	60	0	RD	10.0.0.6	GigabitEthernet0/0/1
192.168.64.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0
192.168.65.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0

认识路由表

□ 路由表里面有什么？

■ Destination/Mask（目的网络）

► 表示“目的网络及其地址掩码”

Destination/Mask	Proto	Pre				
10.0.0.4/30	Direct	0				Interface0/0/1
10.0.5.0/30	Direct	0	0	D	10.0.5.2	GigabitEthernet0/0/3
172.16.1.0/24	Static	60	0	RD	10.0.0.6	GigabitEthernet0/0/1
192.168.64.0/24	RIP					GigabitEthernet0/0/0
192.168.65.0/24	RIP					GigabitEthernet0/0/0

“Mask”指地址掩码（即子网掩码），其值表示掩码中“1”的个数，即网络前缀的长度。例如“24”表示前24位是网络位，对应的子网掩码是255.255.255.0

“Destination”通常用目的网络地址表示（例如172.16.1.0），而不是目的主机地址

认识路由表

□ 路由表里面有什么？

■ Proto（路由来源协议）

- 表示该路由的协议类型，即路由器获取该条路由记录的方式。
- 常见的协议类型包括3种：直连路由、静态路由、动态路由（例 RIP）

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.0.0.4/30	<u>Direct</u>	0	0	D	10.0.0.5	GigabitEthernet0/0/1
10.0.5.0/30	Direct	0	0	D	10.0.5.2	GigabitEthernet0/0/3
172.16.1.0/24	<u>Static</u>	60	0	RD	10.0.0.6	GigabitEthernet0/0/1
192.168.64.0/24	<u>RIP</u>	100	2	D	10.0.0.1	GigabitEthernet0/0/0
192.168.65.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0

认识路由表

□ 路由表里面有什么？

■ Proto（路由来源协议）

- **Direct**：直连路由，通过物理接口直接连接的网络产生。管理员给路由器接口配置IP地址后，即可自动生成直连路由。
- **Static**：静态路由。说明该路由记录是管理员手工配置的。
- **RIP**：路由信息协议（动态路由）。说明该路由记录是通过RIP协议获取到的路由信息。
- **OSPF**：开放最短路径优先协议（动态路由）。说明该路由记录是通过OSPF协议获取到的路由信息。

认识路由表

□ 路由表里面有什么？

■ Pre（优先级）

- 优先级用来标识不同路由的可靠性，数值越小越优：当存在多条到达同一目的地的路由时，设备优先选择优先级数值较小的路由条目
- 华为设备Pre默认值：直连路由(0)、静态路由(60)，RIP(100)，OSPF(10)。

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.0.0.4/30	Direct	0	0	D	10.0.0.5	GigabitEthernet0/0/1
10.0.5.0/30	Direct	0	0	D	10.0.5.2	GigabitEthernet0/0/3
172.16.1.0/24	Static	60	0	RD	10.0.0.6	GigabitEthernet0/0/1
192.168.64.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0
192.168.65.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0

认识路由表

□ 路由表里面有什么？

■ Cost（开销值）

- ▶ 当路由器通过同一种方式获取到多条去往同一网络的路由时，路由器根据Cost值来判断哪条路径更优。不同协议对于计算Cost值会使用不同的参数、按照不同的标准来计算，但总的来说，开销值越小，优先级越高。

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.0.0.4/30	Direct	0	0	D	10.0.0.5	GigabitEthernet0/0/1
10.0.5.0/30	Direct	0	0	D	10.0.5.2	GigabitEthernet0/0/3
172.16.1.0/24	Static	60	0	RD	10.0.0.6	GigabitEthernet0/0/1
192.168.64.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0
192.168.65.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0

认识路由表

□ 路由表里面有什么？

■ Flags（路由标记）（有的路由器不包含此项）

- Flags即路由标记。R，是relay的首字母，说明是迭代路由，会根据路由下一跳的IP地址获取出接口。配置静态路由时如果只指定下一跳IP地址，而不指定出接口，那么就是迭代路由，需要根据下一跳IP地址的路由获取出接口。D，是download的首字母，表示该路由下发到FIB表中。FIB的全称是转发信息库（Forwarding Information Base）。

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.0.0.4/30	Direct	0	0	D	10.0.0.5	GigabitEthernet0/0/1
10.0.5.0/30	Direct	0	0	D	10.0.5.2	GigabitEthernet0/0/3
172.16.1.0/24	Static	60	0	RD	10.0.0.6	GigabitEthernet0/0/1
192.168.64.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0
192.168.65.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0

认识路由表

□ 路由表里面有什么？

■ NextHop（下一跳）

- ▶ 下一跳。每个路由表条目都会指明转发数据包的下一跳地址，这个地址通常是相邻路由设备某个接口的IP地址。

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.0.0.4/30	Direct	0	0	D	10.0.0.5	GigabitEthernet0/0/1
10.0.5.0/30	Direct	0	0	D	10.0.5.2	GigabitEthernet0/0/3
172.16.1.0/24	Static	60	0	RD	10.0.0.6	GigabitEthernet0/0/1
192.168.64.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0
192.168.65.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0

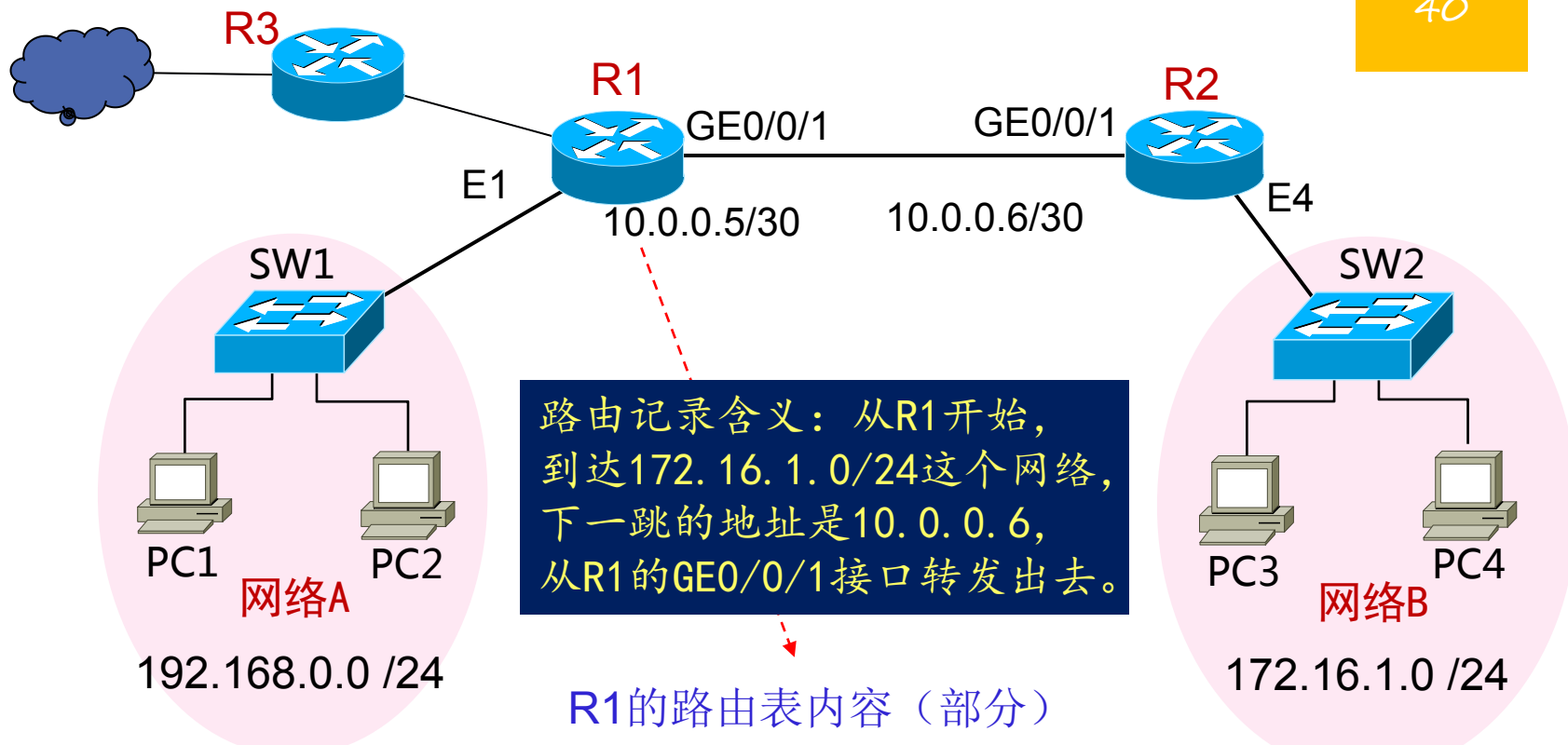
认识路由表

□ 路由表里面有什么？

■ Interface（出接口）

- 指明将分组从本路由器的哪个接口转发出去

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.0.0.4/30	Direct	0	0	D	10.0.0.5	GigabitEthernet0/0/1
10.0.5.0/30	Direct	0	0	D	10.0.5.2	GigabitEthernet0/0/3
172.16.1.0/24	Static	60	0	RD	10.0.0.6	GigabitEthernet0/0/1
192.168.64.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0
192.168.65.0/24	RIP	100	2	D	10.0.0.1	GigabitEthernet0/0/0



Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.0.5.0/30	Direct	0	0	D	10.0.5.2	GigabitEthernet0/0/3
172.16.1.0/24	Static	60	0	RD	10.0.0.6	GigabitEthernet0/0/1

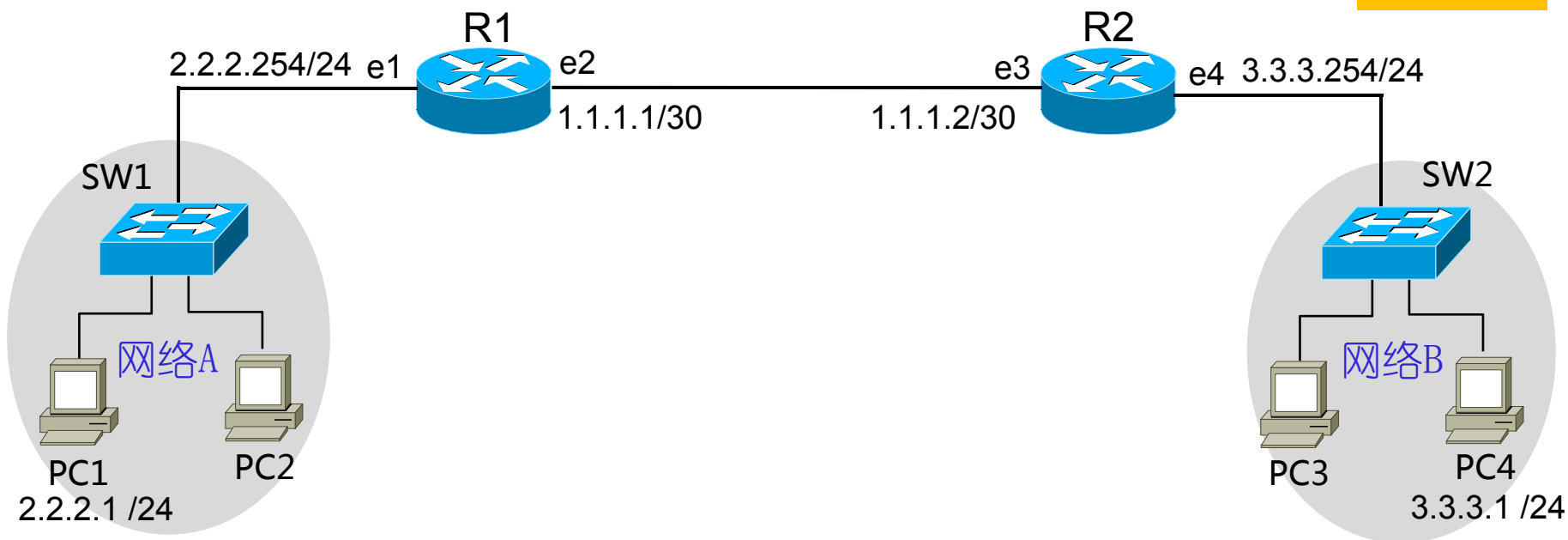
认识路由表

□ 路由表的应用！（生成路由表以后，按照路由表进行报文转发）

- 当路由器接口收到一个报文后，会分析报文首部的目的IP地址，然后通过**查找算法**（例如使用二叉线索查找转发表），查找该目的IP地址与路由表中哪条记录匹配。
- 若匹配上，就根据该记录所提供的路径，将报文从指定的路由器接口发送出去；
- 若没有匹配上，则丢弃该报文。（若交换机呢？）

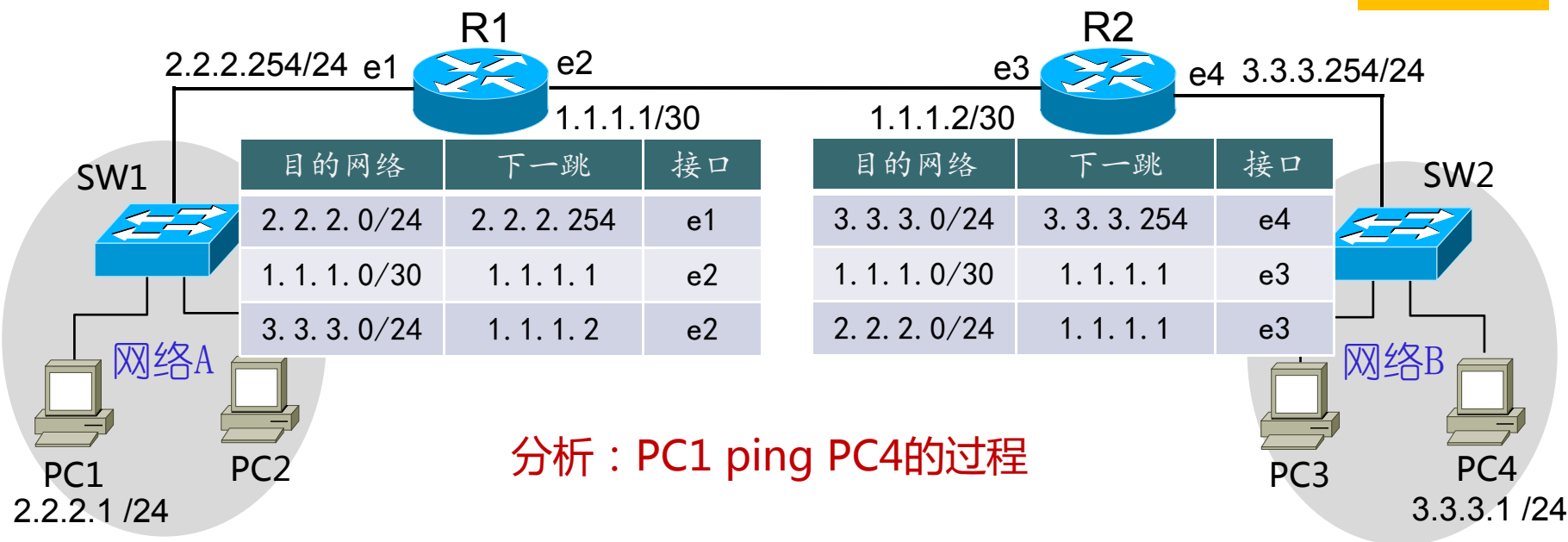
案例分析：路由器转发报文的过程

—— 路由表的应用



说明1:

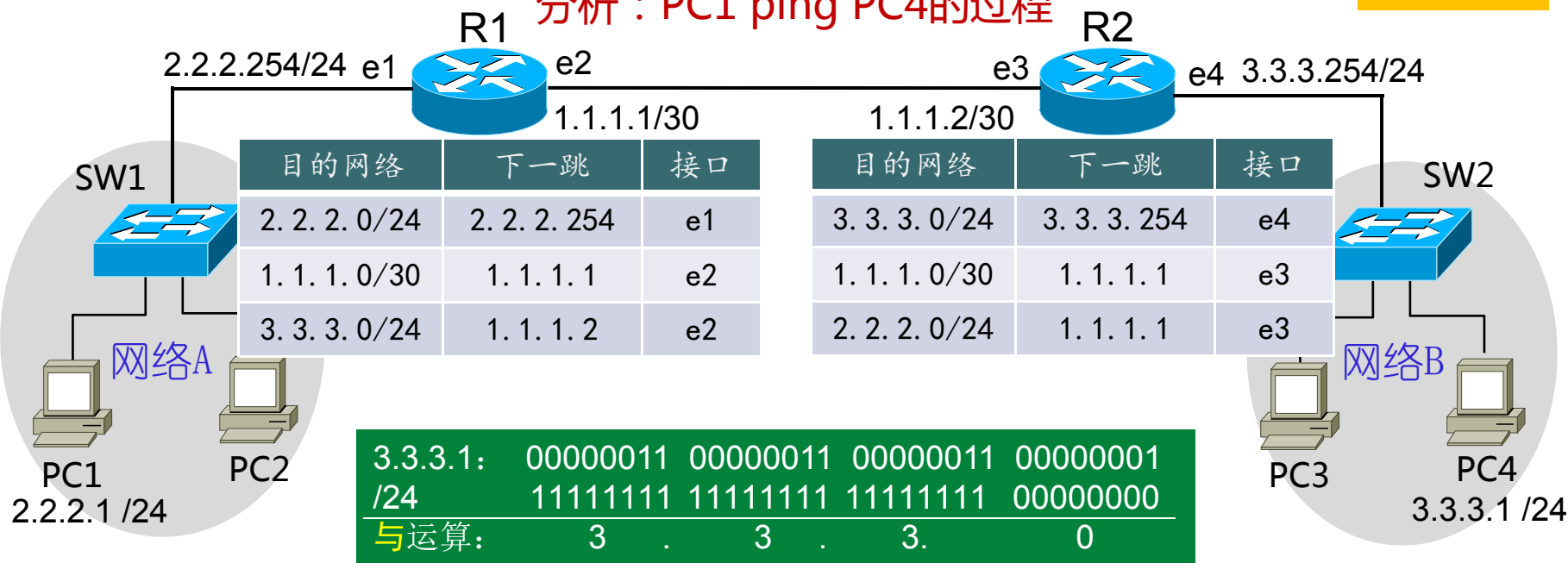
1. 如图，R1、R2是路由器，SW1和SW2是二层交换机；
2. 网络A的网络地址是2.2.2.0/24，其默认网关配置在R1的e1接口，其IP地址是2.2.2.254/24；
3. 网络B的网络地址是3.3.3.0/24，其默认网关配置在R2的e4接口，其IP地址是3.3.3.254/24；



说明2:

1. 路由器各接口已经配置IP地址;
2. SW1和SW2的MAC地址表已经生成（后续不讨论MAC地址表的生成过程）;
3. 如图，R1和R2的路由表已经生成（此处不讨论路由表的配置操作）;

分析：PC1 ping PC4的过程



步骤1: PC1首先判断目的主机与自己是否在同一网络。

- 将PC4的IP地址 (3.3.3.1) 和PC1的子网掩码 (/24, 255.255.255.0), 以二进制形式按位相与, 若结果等于PC1所在网络的网络地址 (即2.2.2.0), 表明PC4和PC1在同一网络
- 结论: 不在同一网络, PC1必须把报文先发给自己的默认网关(R1的e1口)

分析：PC1 ping PC4的过程



步骤2：PC1与本网络默认网关之间的通信（1）

- 由于是ping命令，所以PC1要发出ICMP回送请求报文。
- 在进行封装时，首部中的目的IP是IP_PC4，表明该报文的终点是PC4。
- 根据步骤1的分析，PC1要先把报文发给默认网关，所以在封装成帧时，帧首部中的目的MAC是MAC_e1（即R1的e1接口的MAC地址）。

分析：PC1 ping PC4的过程



步骤2：PC1与本网络默认网关之间的通信（2）

- 若PC1此时不知道默认网关的MAC地址，怎么办？
- PC1可通过ARP协议，获取到e1接口的MAC地址（过程分析略）
 - PC1发出ARP请求报文（源MAC？目的MAC？）
 - e1接口发回ARP响应报文（源MAC？目的MAC？）

分析：PC1 ping PC4的过程



步骤3：SW1转发PC1发来的ICMP请求报文（注意，转发其他报文同理）

- SW1分析所收到帧的首部中的目的MAC地址，并依据自身MAC地址表进行转发。
- 注意，转发过程中，SW1没有剥掉帧首部，也没有重新封装帧。只是读取目的MAC地址，然后依据MAC地址表进行转发而已。这一点与路由器不同！

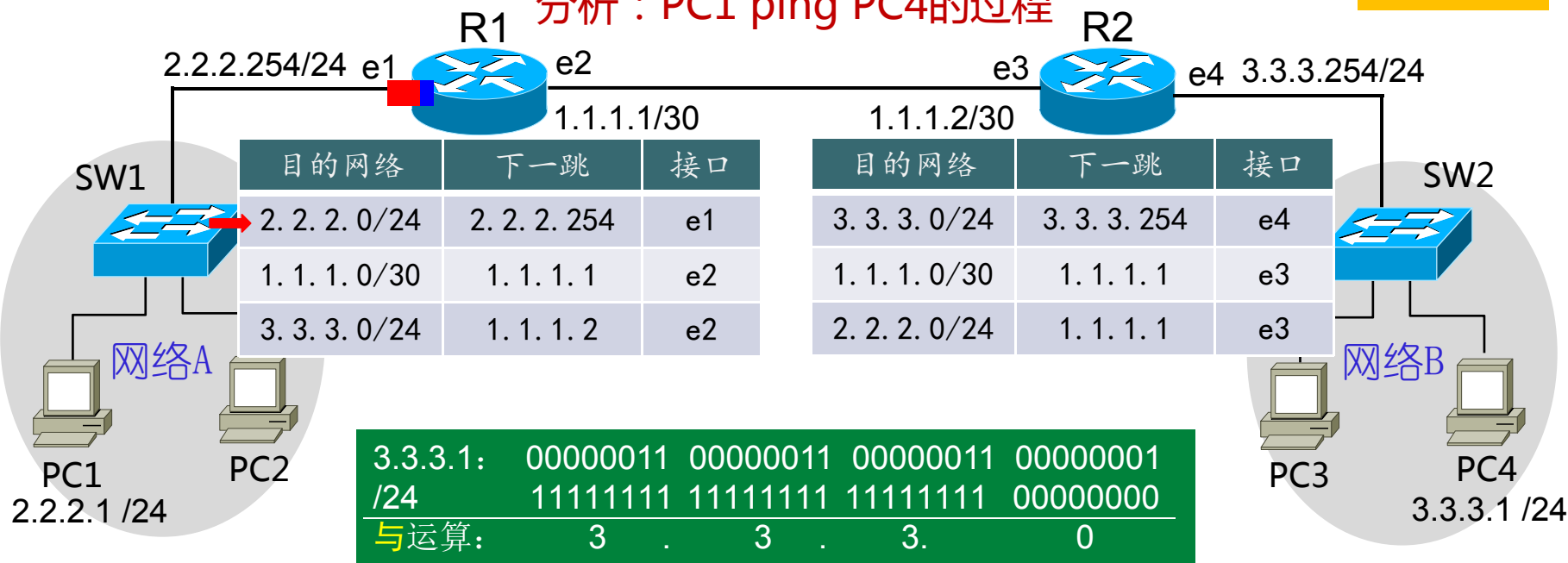
分析：PC1 ping PC4的过程



步骤4：R1接收PC1发来的ICMP请求报文

- R1收到PC1发来的报文后，是否有解封装操作？
 - 路由器是三层设备，其e1接口的数据链路层会分析帧首部的目的MAC地址，确认是发给自己的。然后去掉帧首、帧尾，将里面的分组发给自己的网络层，进行进一步处理。
 - 提示：路由器接口的数据链路层处理帧首部，网络层处理分组首部。

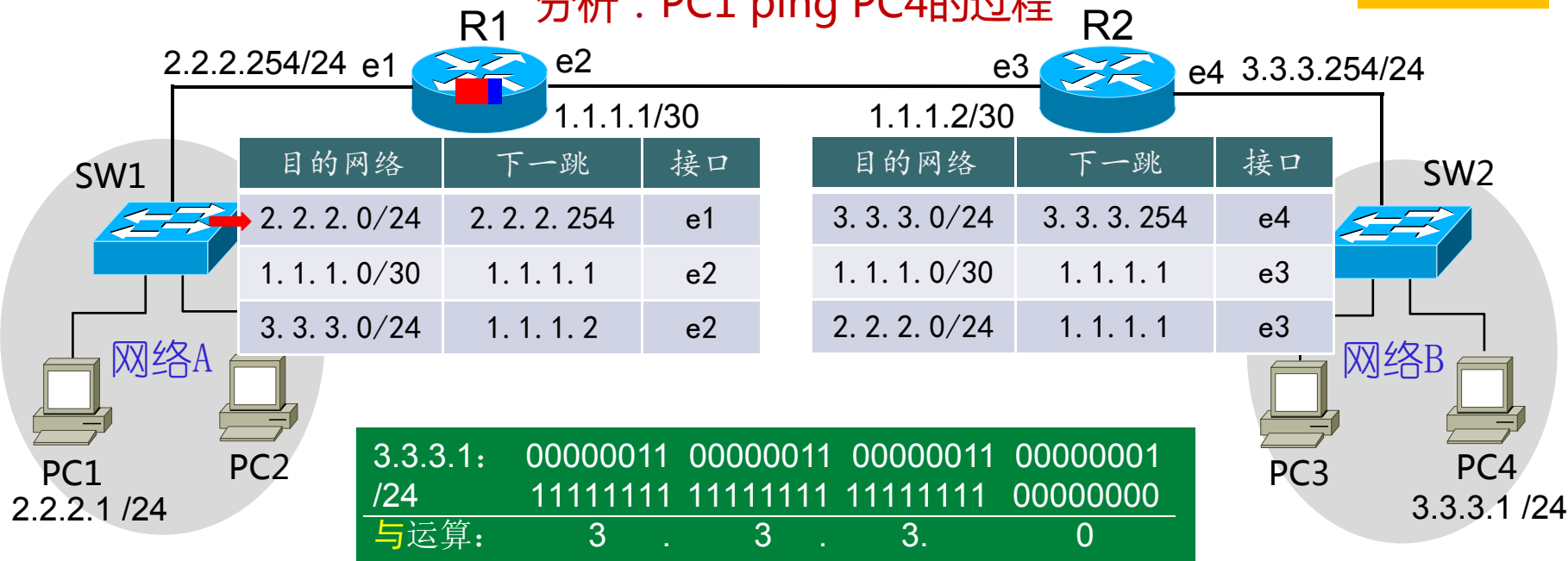
分析：PC1 ping PC4的过程



步骤5: R1查找转发表：寻找前缀匹配的过程 (1)

- 读取分组首部中的目的IP地址 (3.3.3.1)，与转发表第1条记录中的子网掩码 (/24, 255.255.255.0) 按二进制位进行与 (and) 运算。
- 运算结果 (3.3.3.0) 不等于该记录的目的网络地址 (2.2.2.0)，说明未匹配，继续查找。

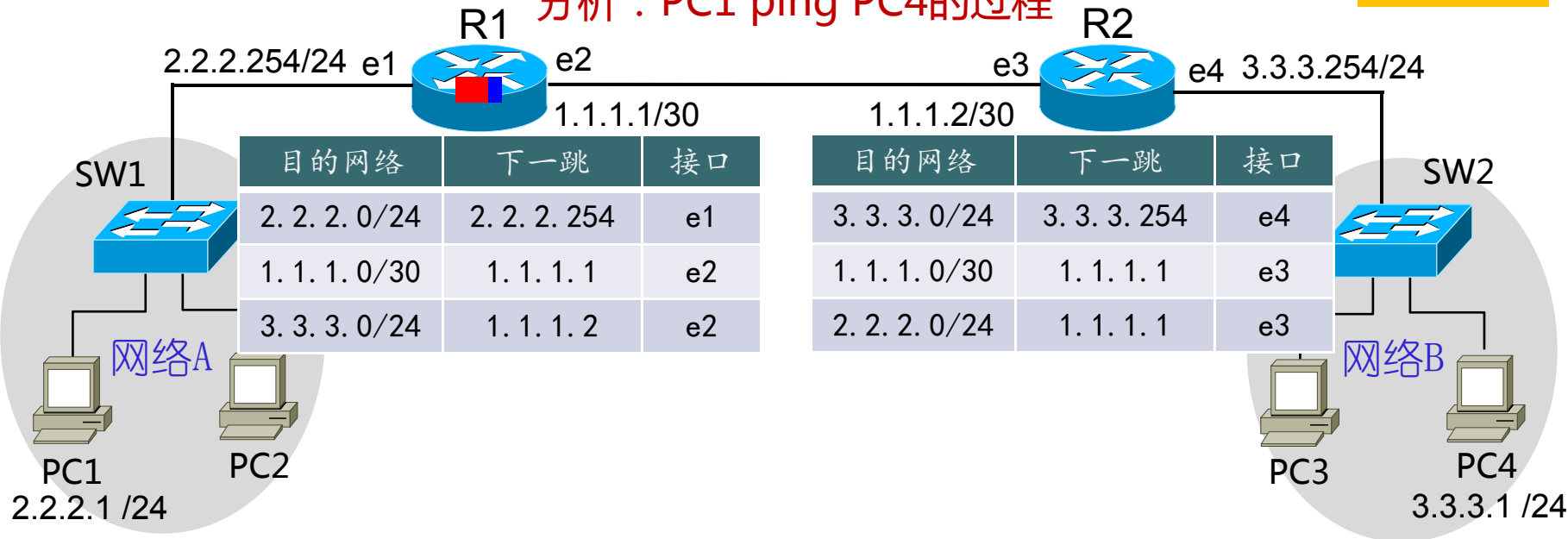
分析：PC1 ping PC4的过程



步骤5: R1查找转发表：寻找前缀匹配的过程 (2)

- 继续查找，发现第3条记录匹配，则按照该记录提供的路径进行转发。
- 含义：(从R1开始)要到达3.3.3.0/24这个网络，下一跳的IP地址是1.1.1.2 (注意，下一跳是e3, 不是e2)，把分组从本路由器的e2接口发出。

分析：PC1 ping PC4的过程



步骤6: R1 (e2接口) 重新封装并转发所收到的ICMP请求报文

- R1的e2接口在转发报文时，需要重新封装报文，具体如下：
 - 报文首部中源IP和目的IP保持不变，即报文的起点信息和终点信息不变；
 - e2接口的数据链路层重新封装成帧，帧首部的源MAC变为MAC_e2，目的MAC变为MAC_e3。
 - 提示，e2和e3的IP地址一定属于同一网段，e2可通过ARP获取e3的MAC。

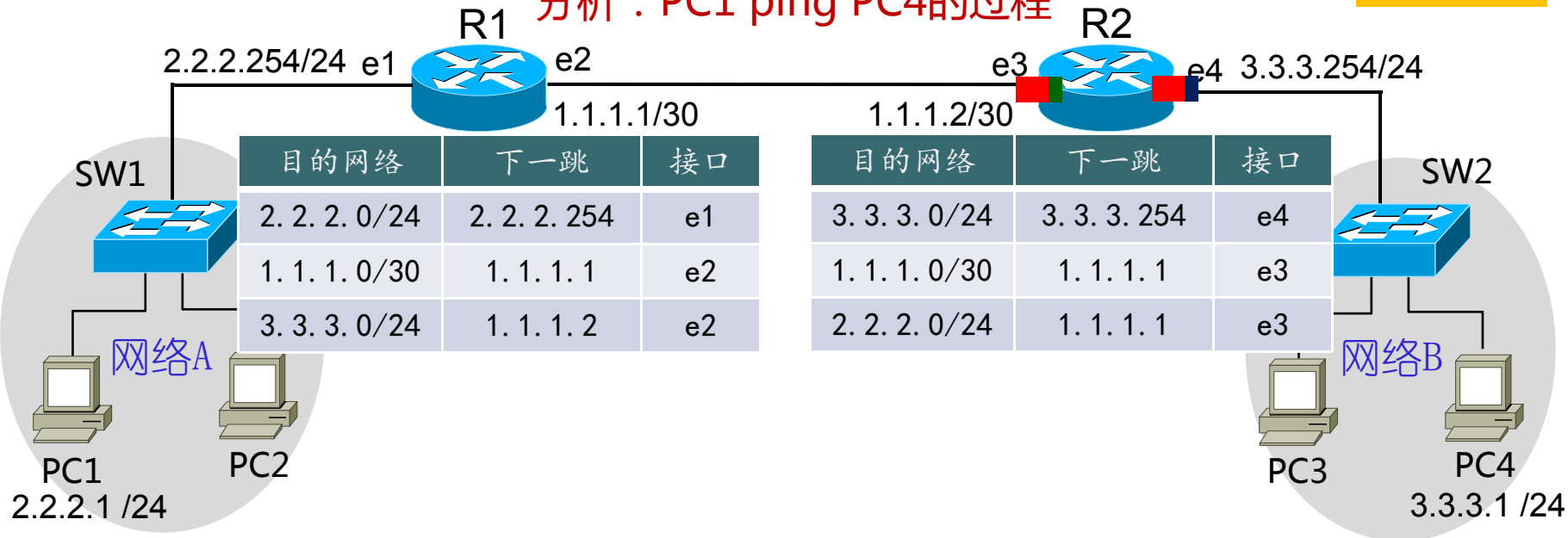
分析：PC1 ping PC4的过程



步骤7：R2转发所收到的ICMP请求报文

- R2的转发过程，与R1基本相同；
- **区别1**：在匹配转发表记录时，由于网络B是R2的直连网络（其默认网关就在R2上），所以其下一跳地址就是网络B的默认网关（即e4口）的IP地址；
- **区别2**：e4接口的数据链路层重新封装成帧，帧首部的源MAC变为MAC_e4，目的MAC变为MAC_PC4。（因为PC4和e4口属于同一网段）

分析：PC1 ping PC4的过程



步骤8：SW2转发所收到的ICMP请求报文（同SW1）

- SW2分析所收到帧的首部中的目的MAC地址，并依据自身MAC地址表进行转发。
- **注意**，转发过程中，SW1没有剥掉帧首部，也没有重新封装帧。只是读取目的MAC地址，然后依据MAC地址表进行转发而已。这一点与路由器不同！

分析：PC1 ping PC4的过程



步骤9：PC4接收ICMP请求报文

- PC4收到ICMP请求报文，解封装，分析目的MAC和目的IP地址，发现是给自己的，收下。

分析：PC1 ping PC4的过程



步骤10：PC4发回ICMP响应报文

- PC4发现该报文是ICMP的回送请求报文，根据ICMP协议，PC4会返回ICMP回送响应报文。
- 具体过程略

路由器转发报文的过程

□ 分析两个问题

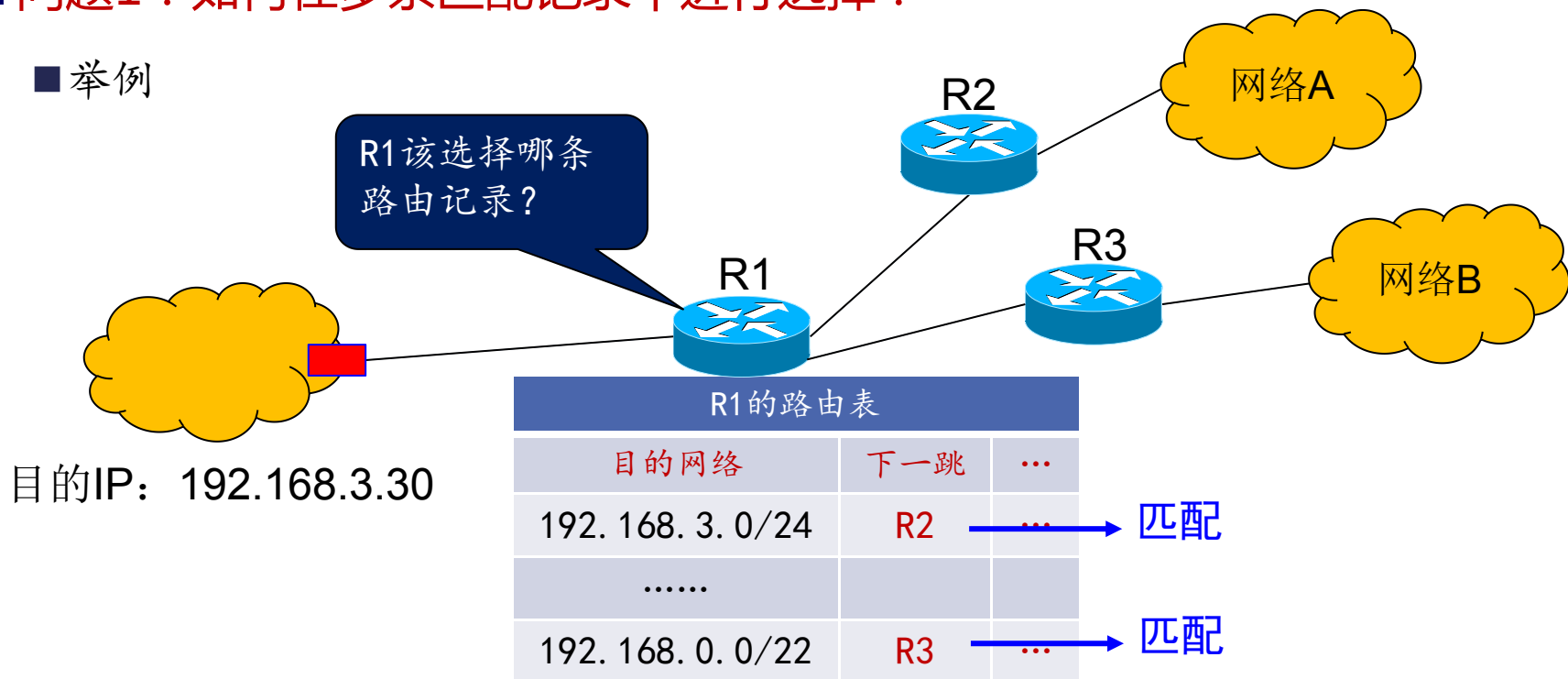
1. 路由器根据分组首部中的目的IP地址，在匹配转发表中的记录时，发现能匹配上的**不止一条**路由记录，按照哪一条进行转发？
2. 如何提高路由器查找路由表（即转发表）的速度？

【注】大家结合《计算机网络原理》课程的内容，自主学习

路由器转发报文的过程

□ 问题1：如何在多条匹配记录中进行选择？

■ 举例



路由器转发报文的过程

□ 问题1：如何在多条匹配记录中进行选择？

■ 再谈“最长前缀匹配”原则

- 路由器遵循**最长匹配原则**对收到的报文（分组）进行路由匹配！
- “在采用CIDR编址时，如果一个分组在转发表中可以找到多个匹配的前缀，那么就应当选择前缀最长的一个作为匹配的前缀。这个原则称为最长前缀匹配。网络前缀越长，其地址块就越小，因而路由就越具体。”

—— 《计算机网络

（第8版）》 P143

路由器转发报文的过程

□ 问题1：如何在多条匹配记录中进行选择？

■ 再谈“最长前缀匹配”原则

- 第1条匹配记录的网络前缀是24位（/24）
- 第2条匹配记录的网络前缀是22位（/22）

■ 根据“最长前缀匹配”原则，路由器选择第1条匹配记录。

R1选择该条路由记录！

R1的路由表		
目的网络	下一跳	...
192. 168. 3. 0/24	R2	...
.....		
192. 168. 0. 0/22	R3	...

路由器转发报文的过程

□ 问题2：如何提高路由器查找转发表的速度？

■ 路由器转发表的项目数很大时，**顺序查找**显然是很烂的算法！怎样设法缩短转发表的查找时间就成为一个非常重要的问题！

■ 二叉线索查找转发表（**好的数据结构 + 快速查找算法**）

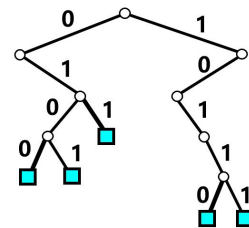
➤ 为了进行更加有效的查找，通常是把无分类编址的转发表存放在一种层次数据结构中，然后自上而下地按层次进行查找。最常用的就是二叉线索索。《计算机网络（第8版）》P144，4.3.3 使用二叉线索查找转发表

32 位的 IP 地址

```
01000110 00000000 00000000 00000000
01010110 00000000 00000000 00000000
01100001 00000000 00000000 00000000
10110000 00000010 00000000 00000000
10111011 00001010 00000000 00000000
```

唯一前缀

```
0100
0101
011
10110
10111
```



路由器转发报文的过程

□ 问题3：路由表如何生成？

- **回忆**：交换机开机启动后，会通过“MAC地址学习”功能，自动生成MAC地址表，从而实现数据帧在不同接口间转发；
- 路由器出厂配置中，路由表是空白的，即在出厂的缺省配置下，路由器是“**阻断**”一切通信的。
- **管理员必须对路由器进行一定的配置操作**，例如配置静态路由记录，或者启用某种动态路由协议（例如RIP）等，方可使路由表中有相应的路由记录。
- 路由表中的记录生成（即获取）方式通常有三种：
 - **直连路由** **静态路由** **动态路由**

三、路由表的生成

—— 直连路由

路由表的生成 —— 直连路由

□ 直连路由是如何生成的？

■ 给路由器接口配置IP地址

- 配置路由器时，进行的第1个操作通常就是给路由器接口配置IP地址；
- 当管理员给路由器**活动接口**配置了IP地址/子网掩码，路由器就会根据该地址，自动计算出该接口所属网络（即直连网络）的网络地址信息，并将该信息以直连路由的形式，写入路由表，即路由表中就会自动生成对应的直连路由。

路由表的生成 —— 直连路由

□ 直连路由是如何生成的？

■ 路由器接口配置IP地址举例

```
[R1] interface Ethernet 0/0/0
```

//进入接口视图

```
[R1-Ethernet0/0/0] ip address 192.168.1.1 255.255.255.0
```

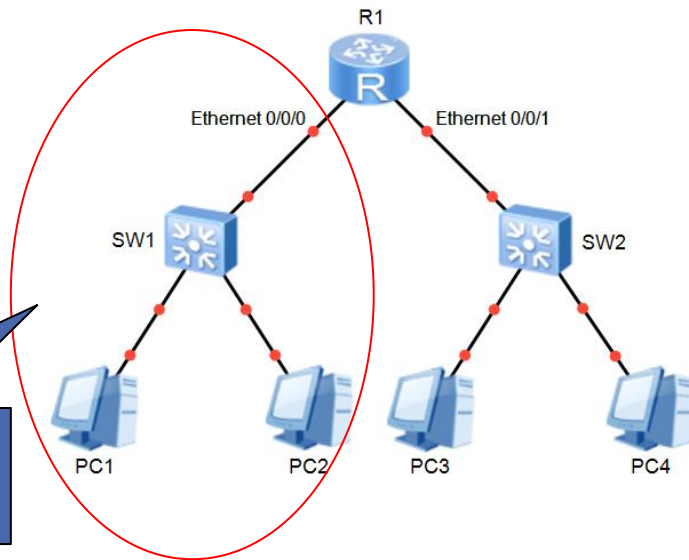
//配置当前接口的IP

给R1的Ethernet0/0/0接口配置IP地址

192.168.1.1，子网掩码为255.255.255.0

R1会计算出Ethernet0/0/0接口所直连的网络的网络地址为192.168.1.0/24，并自动在路由表中记下相应的路由记录。

该网络的网络地址是
192.168.1.0/24



路由表的生成 —— 直连路由

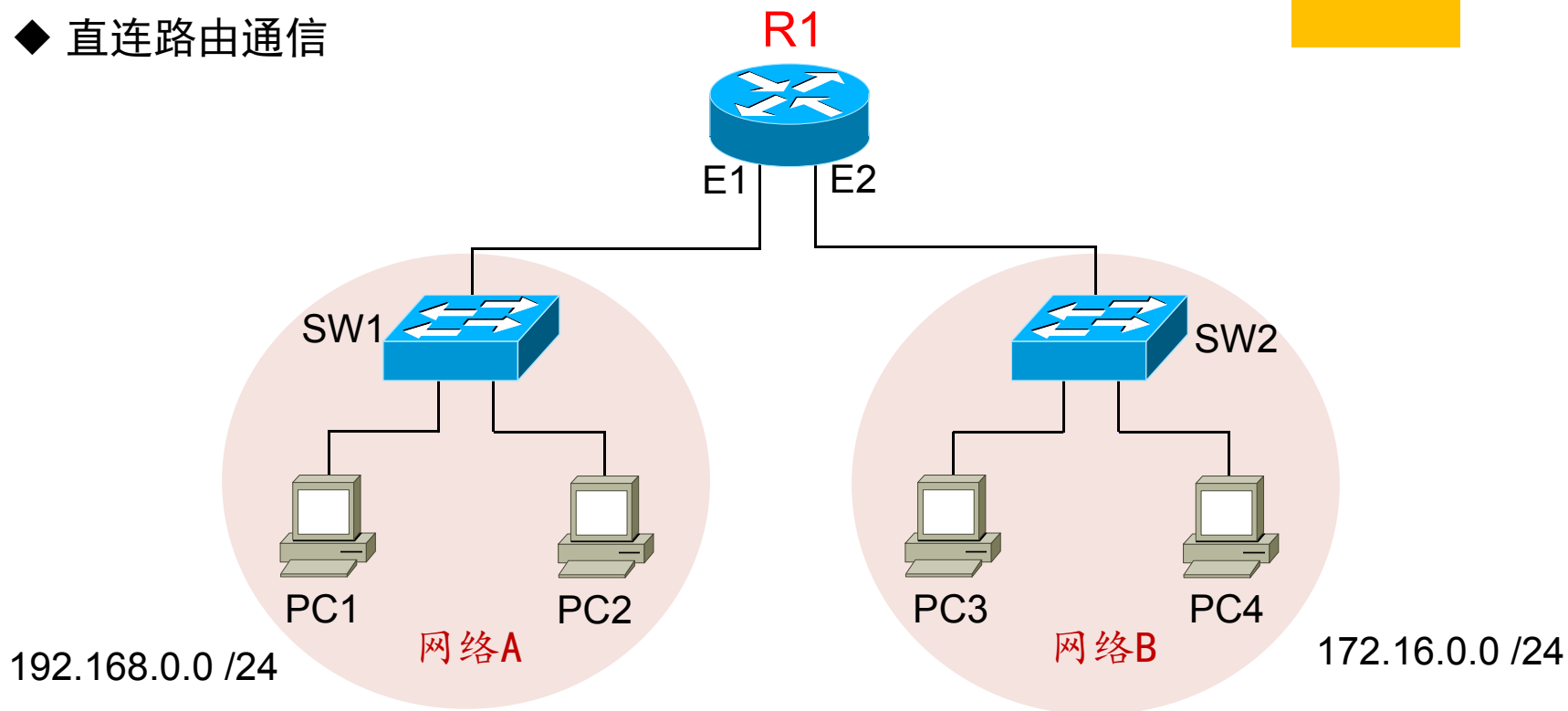
□ 直连路由的应用

■ 实现直连网络之间的互联互通。

- ▶ 路由器各接口直接连接的网络，称为直连网络，直连网络之间，通过路由器的直连路由即可实现通信。

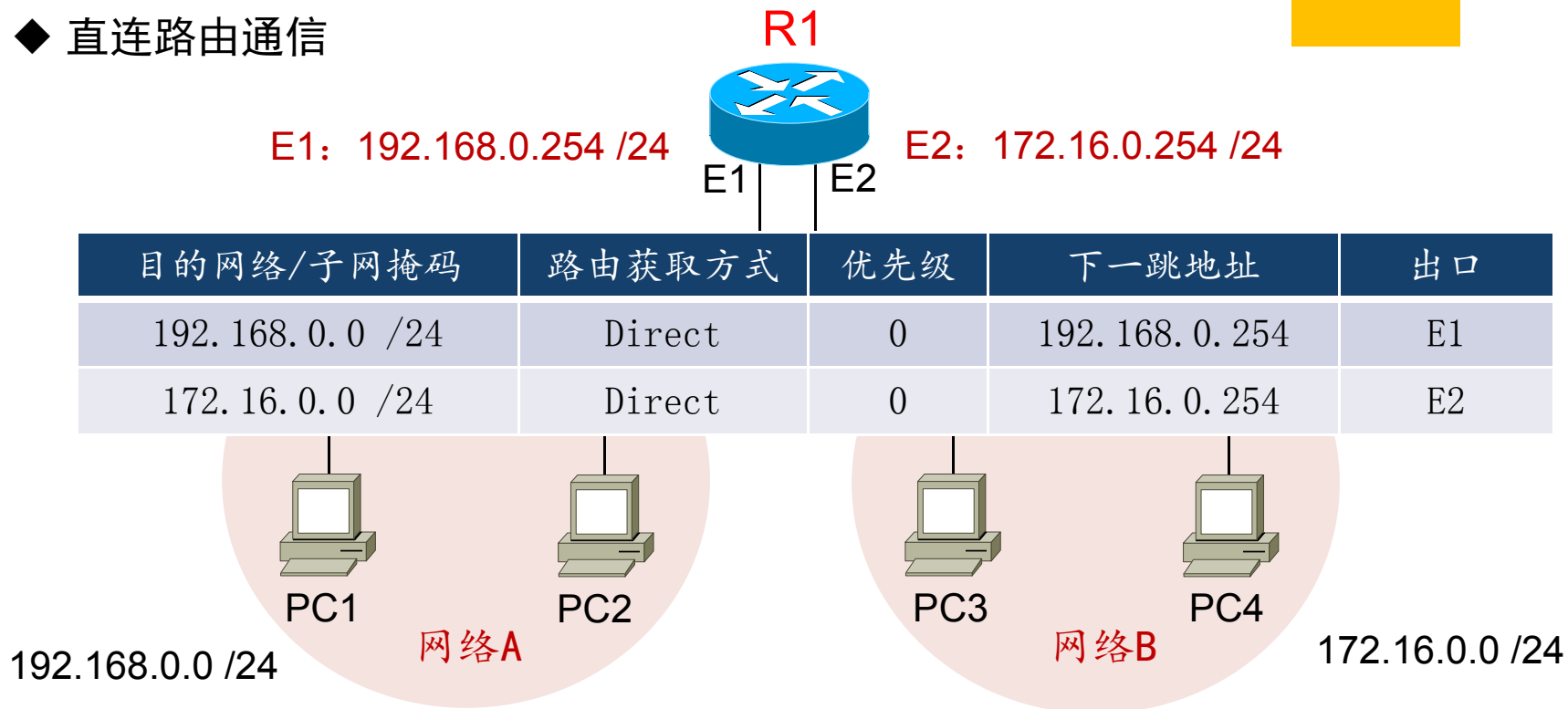
■ 举例

◆ 直连路由通信



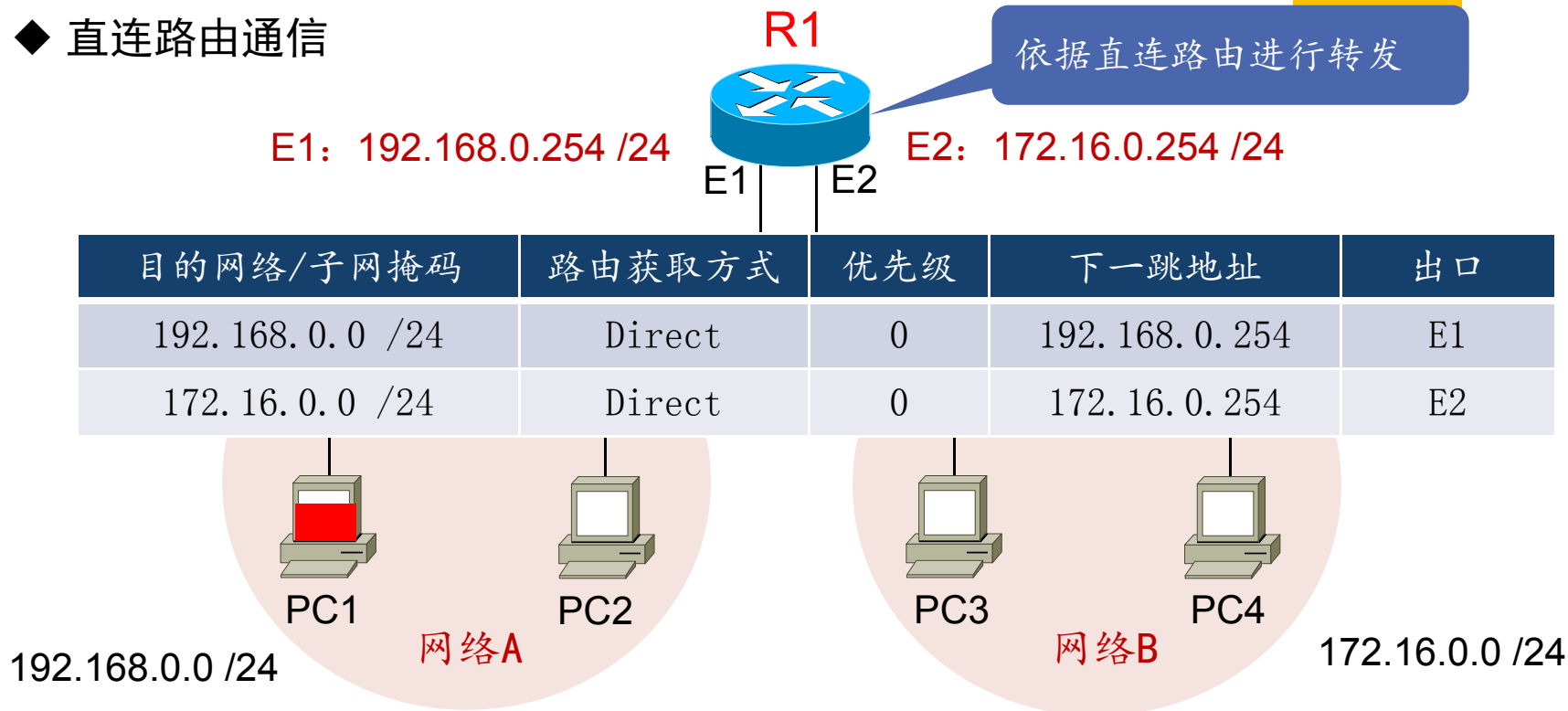
1. **网络规划**: R1为路由器, SW1和SW2为交换机。网络A直接连接R1的E1接口, 网络B直接连接R1的E2接口。网络A和网络B的IP地址如图所示。

◆ 直连路由通信



2. 分别给R1的E1接口和E2接口配置IP地址。注意，此处E1是网络A的默认网关，E2是网络B的默认网关。
配置完成后，R1中自动生成两条直连路由。

◆ 直连路由通信



3. 将网络A中各主机的默认网关地址设置为E1接口的IP地址，网络B中各主机的默认网关地址设置为E2接口的IP地址。则网络A和网络B可正常通信。（例如PC1 访问 PC4）

四、路由表的生成

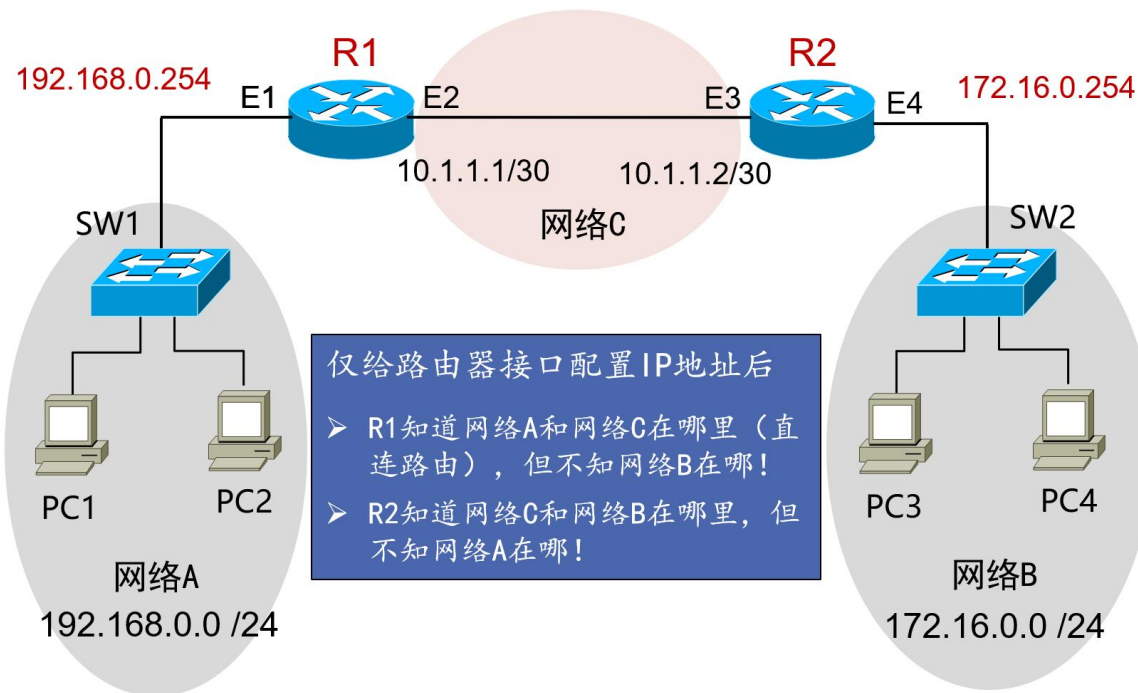
—— 静态路由选择策略

路由表的生成 —— 静态路由

□ 直连路由存在的问题

■ 仅通过直连路由，路由器无法获取远端网络的位置信息。

■ 举例



路由表的生成 —— 静态路由

□ 直连路由存在的问题

- 此时，就有必要为路由器添加对远端网络位置的认知信息，即配置路由选择策略。
- 根据路由选择策略能否随网络拓扑结构变化而自适应的调整，分为静态路由选择策略和动态路由选择策略。

路由表的生成 —— 静态路由

□ 静态路由的生成

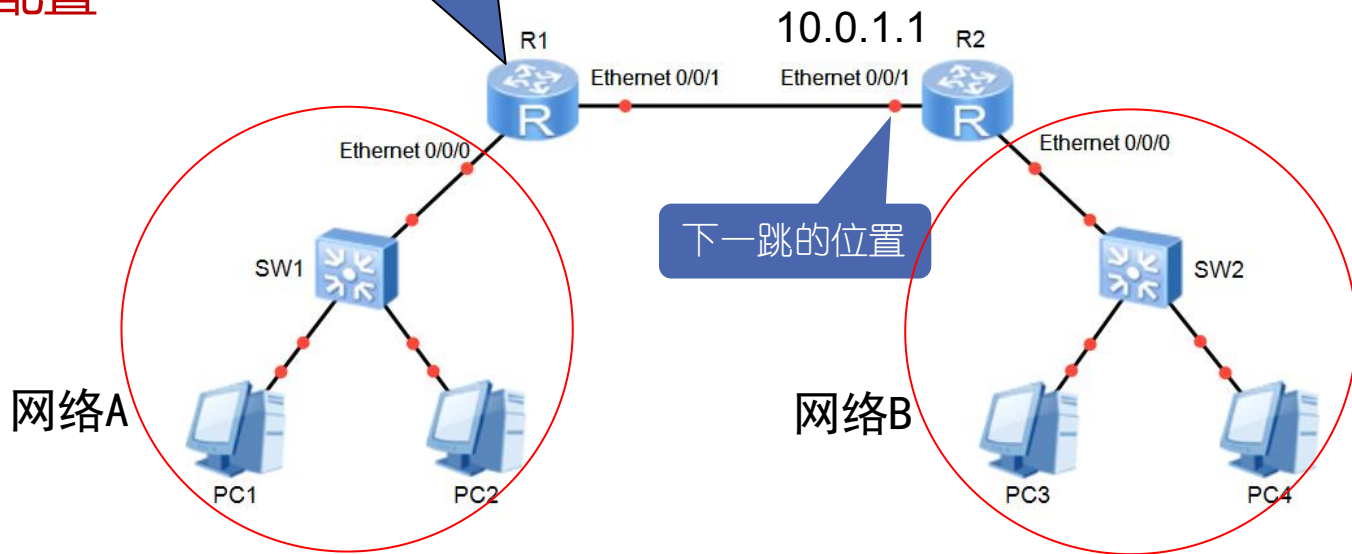
■ 手工配置

- 路由器不会自动生成所需的静态路由记录，需要网络管理员根据网络互联互通的实际需要，在路由器上一条条手工配置。
- 静态路由记录中主要包括：目的网络地址、下一跳地址、出端口。
- 配置命令举例

路由表的生成

在R1上配置到达网络B的静态路由

静态路由的配置



```
[R1]ip route-static 192.168.0.0 255.255.255.0 10.0.1.1
```



说明：在路由器R1上配置一条静态路由，其含义为：**从路由器R1开始**，到达192.168.0.0/24这个网络，其下一跳地址为10.0.1.1（即路由器R2的Ethernet0/0/1接口）

路由表的生成 —— 静态路由

□ 静态路由的特点

■ 固定不变

- 由网络管理员预先设置好路由表，除非网络管理员干预，否则静态路由不会随着网络拓扑结构的变化而自动发生变化。

■ 不可通告性

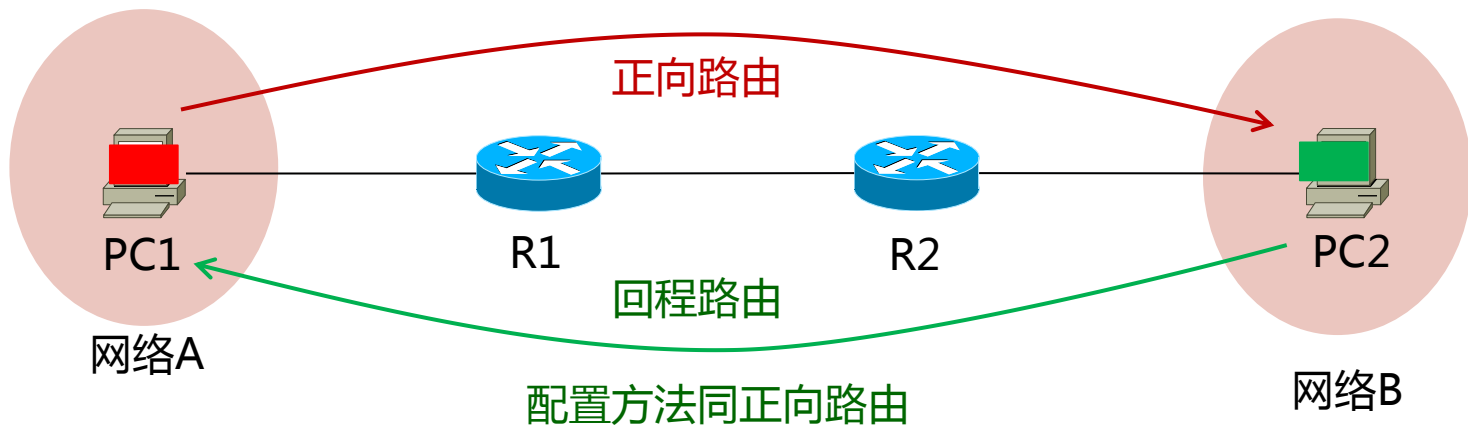
- 缺省情况下，一台路由器上配置的静态路由信息，不会被通告到网络中相连的其他路由器。（我所具有的，不会传给别人）

路由表的生成 —— 静态路由

□ 静态路由的特点

■ 单向性

- ▶ 静态路由是单向性的，也就是它仅为数据包提供沿着下一跳的方向进行路由，不提供反向路由。所以要使源主机与目的主机进行双向通信，就必须同时配置回程路由。（能到达网络B，但未必能返回！）



路由表的生成 —— 静态路由

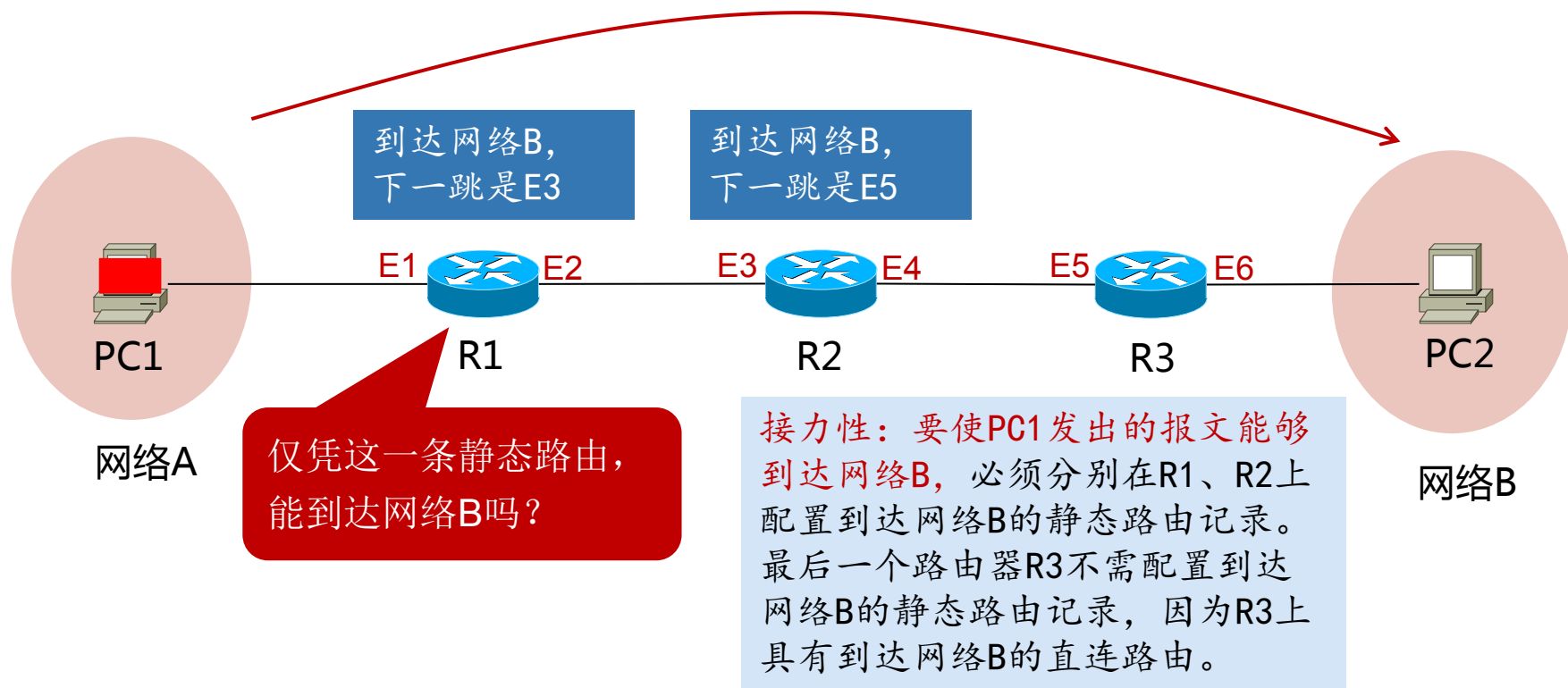
□ 静态路由的特点

■ 接力性

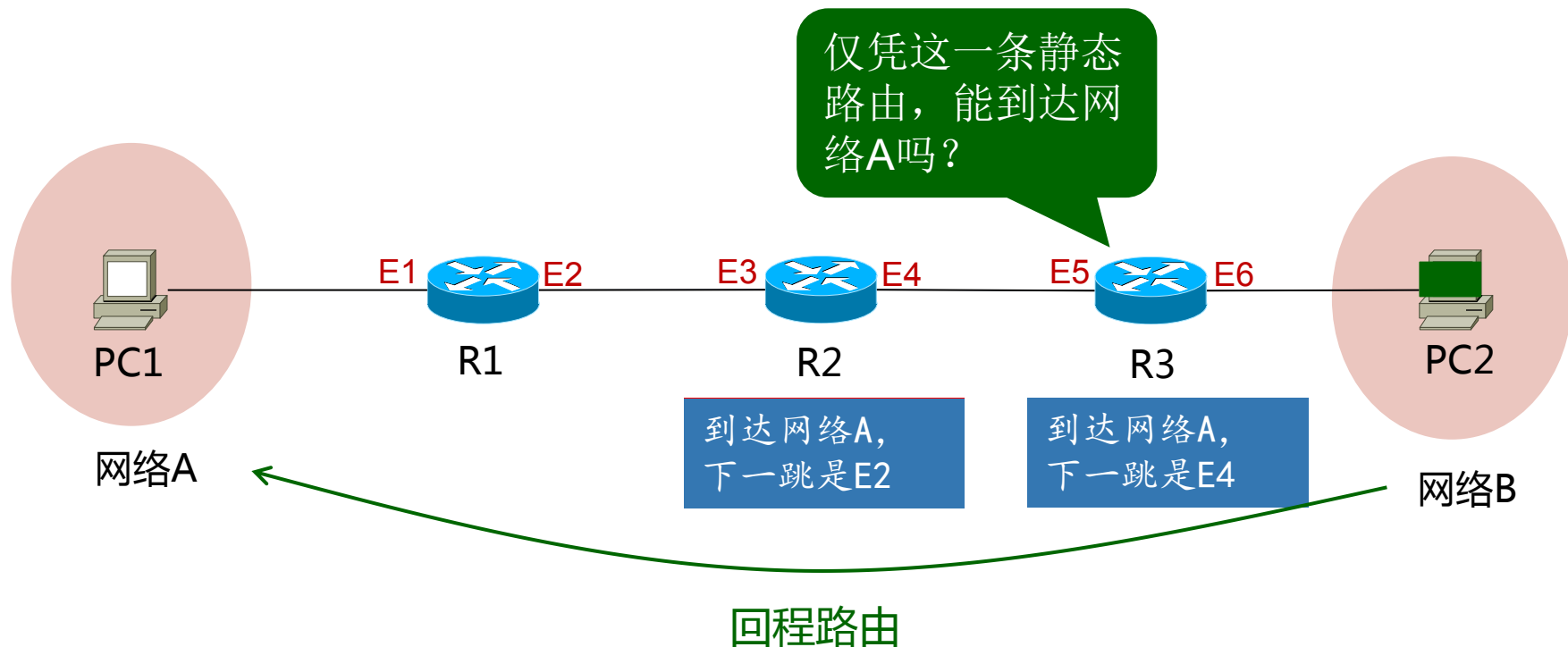
- 若某条静态路由的路径中间包含多台路由器，则必须在除最后一个路由器外的其他路由器上，依次配置到达相同目的网络的静态路由，这就是静态路由的接力性。
- 举例

◆ 演示：静态路由的接力性

正向路由



◆ 演示：静态路由的接力性



路由表的生成 —— 静态路由

□ 静态路由的特点

■ 适用于小型网络

- ▶ 根据静态路由的基本概念和工作特点，静态路由一般适用于较为简单的网络环境，在这样的环境中，管理员易于清楚地了解整个网络的拓扑结构，便于配置正确的路由信息。
- ▶ 如果网络规模较大，拓扑结构比较复杂，则适宜采用动态路由协议。

路由表的生成 —— 静态路由

特殊的静态路由：缺省路由及应用 (自主学习)

路由表的生成 —— 静态路由

□ 认识缺省路由（也叫默认路由）

- **作用：**缺省路由又称默认路由。当报文的路由与路由表中的所有路由都匹配失败时，为使报文有一个最终的发送地，路由器或网络主机将根据缺省路由转发报文。
- **若没有**配置缺省路由，则在路由表中没有匹配成功的数据包将被丢弃。
- **特殊之处：**路由记录中的“目的网络/掩码”用0.0.0.0 0.0.0.0表示。

路由表的生成 —— 静态路由

□ 缺省路由的配置

■ 缺省路由既能由网络管理者设定，也能由某些动态路由协议生成

【方式1】管理员设定

```
[R1]ip route-static 0.0.0.0 0.0.0.0 10.0.1.1
```

说明：在路由器R1上配置一条缺省路由，其含义为：从路由器R1开始，不论到达任何网络，其下一跳地址为10.0.1.1

路由表的生成 —— 静态路由

▣ 缺省路由的配置

【方式2】使用动态路由协议广播缺省路由（例如OSPF）

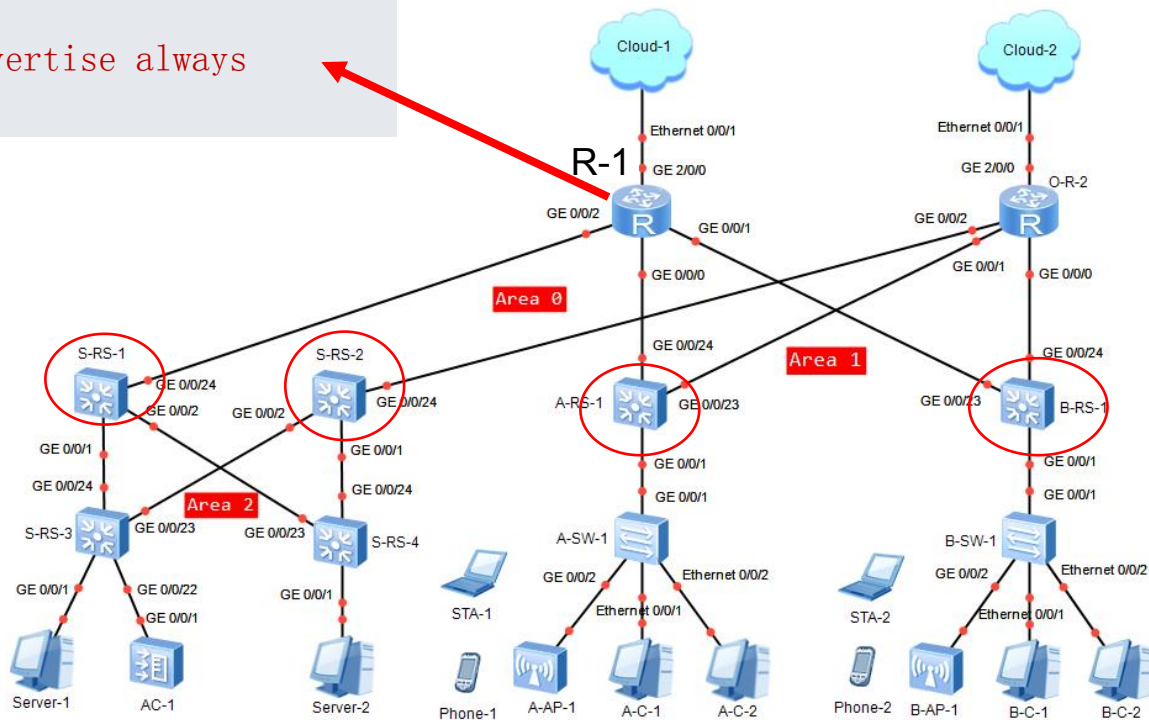
```
[R-1] ip route-static 0.0.0.0 0.0.0.0 11.11.11.2      //配置缺省路由
[R-1]ospf 1
//在ospf中引入缺省路由，并将其发布到整个OSPF 1区域
[R-1-ospf-1]default-route-advertise always
[R-1-ospf-1]area 0
[R-1-ospf-1-area-0.0.0.0]network 10.0.0.0 0.0.0.3
```

说明：某园区网内部各路由器使用OSPF动态路由协议，在边界路由器上配置了一条缺省路由，用以转发访问互联网的报文。为使内部各路由器也能转发发往互联网的报文，可在边界路由器上做如上配置，配置中用到的命令default-route-advertise将缺省路由通告到OSPF区域中，这样内部网中的其他路由器也会学习到该缺省路由，并自动修改下一跳地址。

【方式2】使用动态路由协议广播缺省路由（例如OSPF）

```
[R1] ip route-static 0.0.0.0 0.0.0.0 11.11.11.2
[R1] ospf 1
[R1-ospf-1] default-route-advertise always
[R1-ospf-1] area 0
```

通过在路由器R-1上配置上述命令，可将R-1上的缺省路由发到园区网内部的其他路由器，即其他路由器上也会自动产生一条缺省路由。



路由表的生成 —— 静态路由

□ 缺省路由应用案例

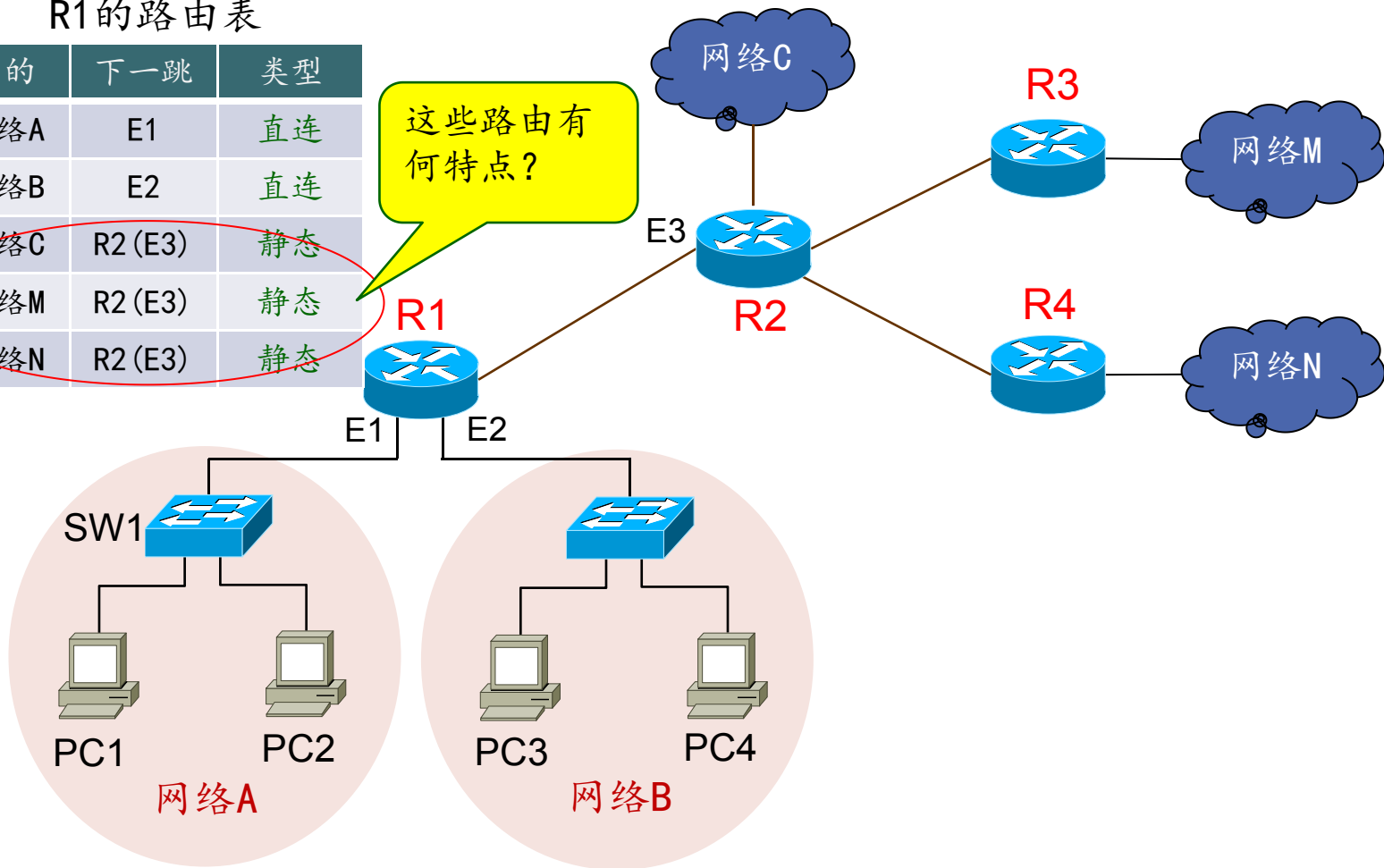
- **简化路由配置：**缺省路由在某些时候非常有效，例如，存在末梢网络时，缺省路由会大大简化路由器的配置，减轻管理员的工作负担，提高网络性能。
- 案例

◆ 演示：缺省路由的应用案例

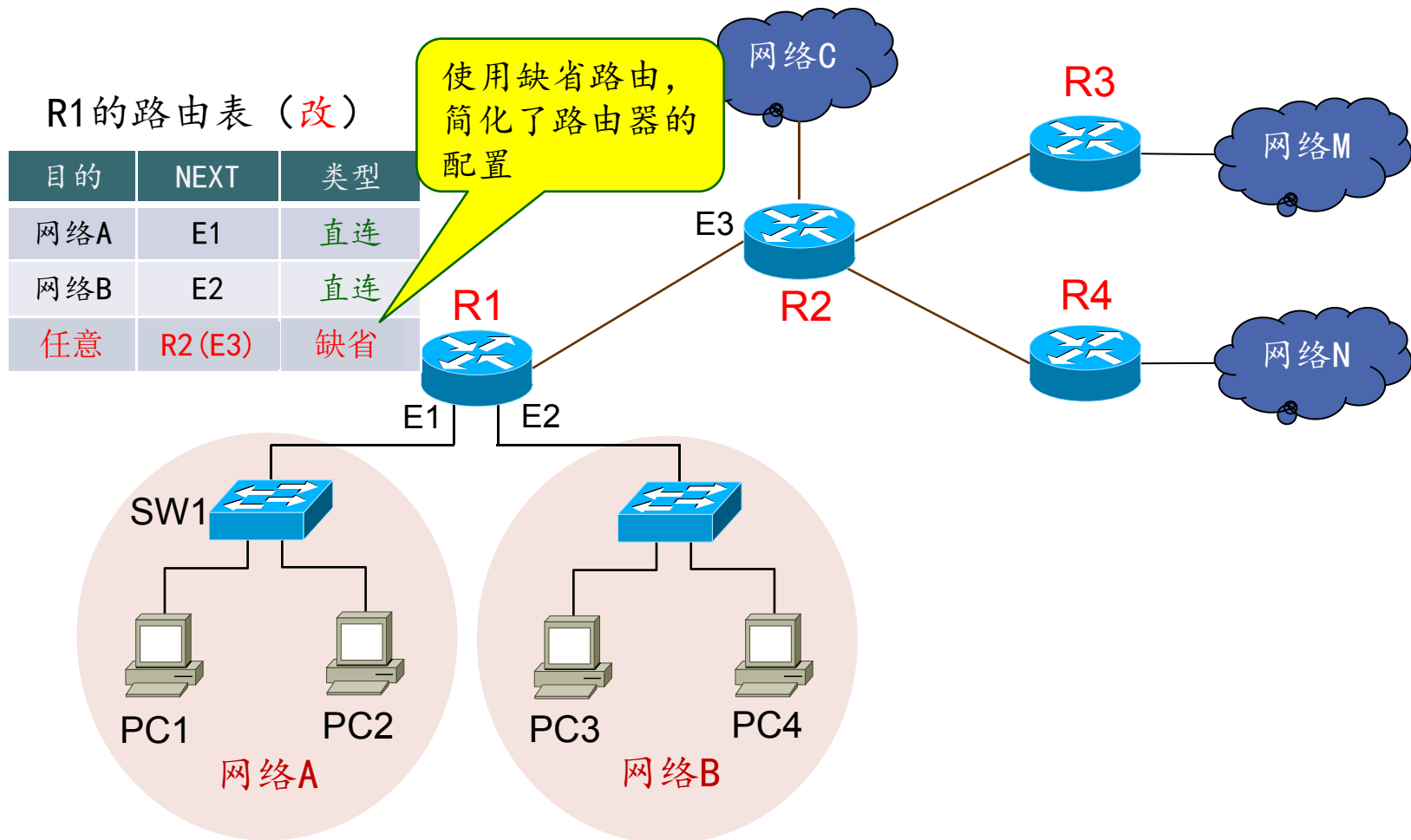
R1的路由表

目的	下一跳	类型
网络A	E1	直连
网络B	E2	直连
网络C	R2 (E3)	静态
网络M	R2 (E3)	静态
网络N	R2 (E3)	静态

这些路由有何特点？



◆ 演示：缺省路由的应用案例

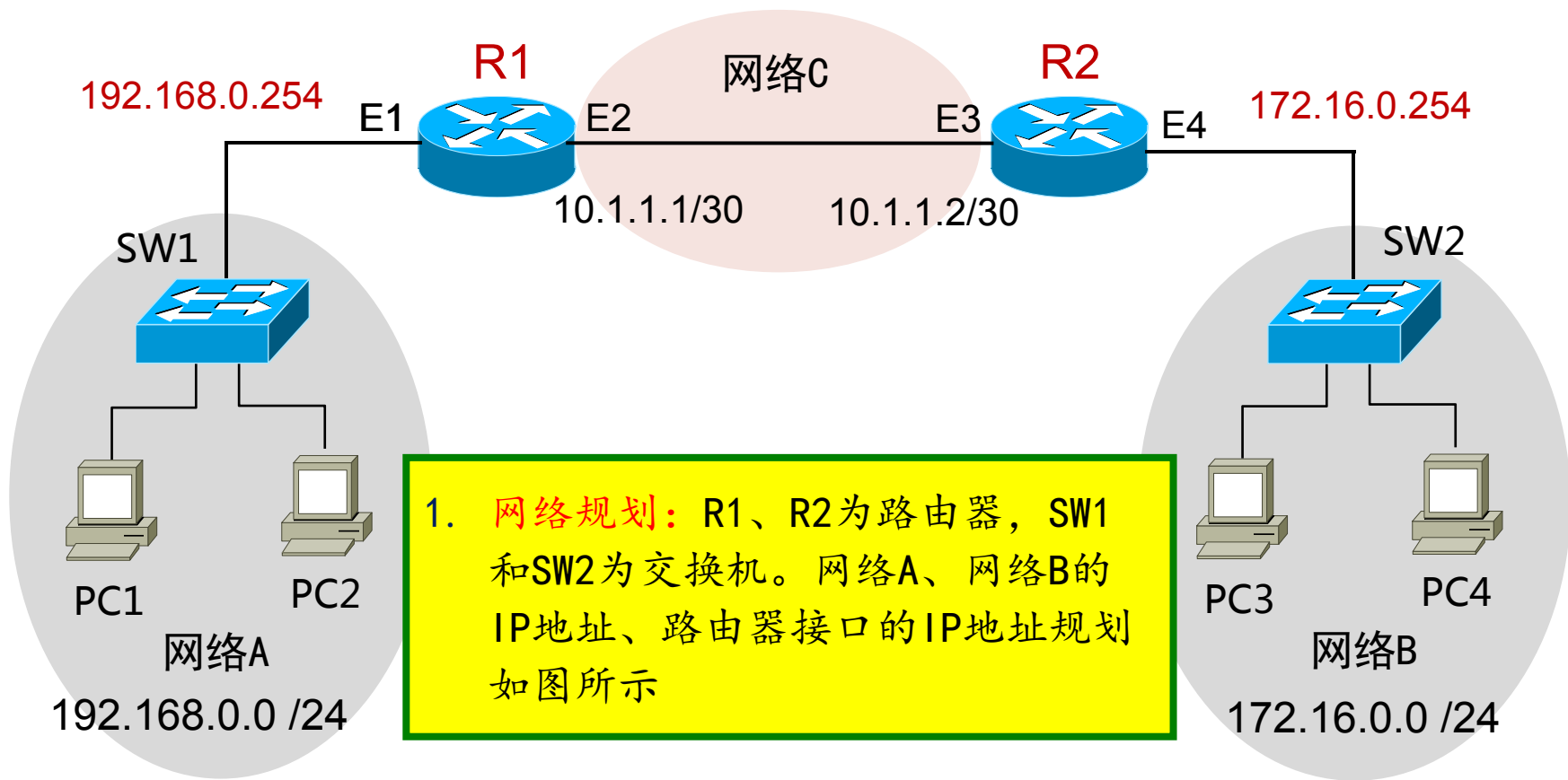


案例研讨

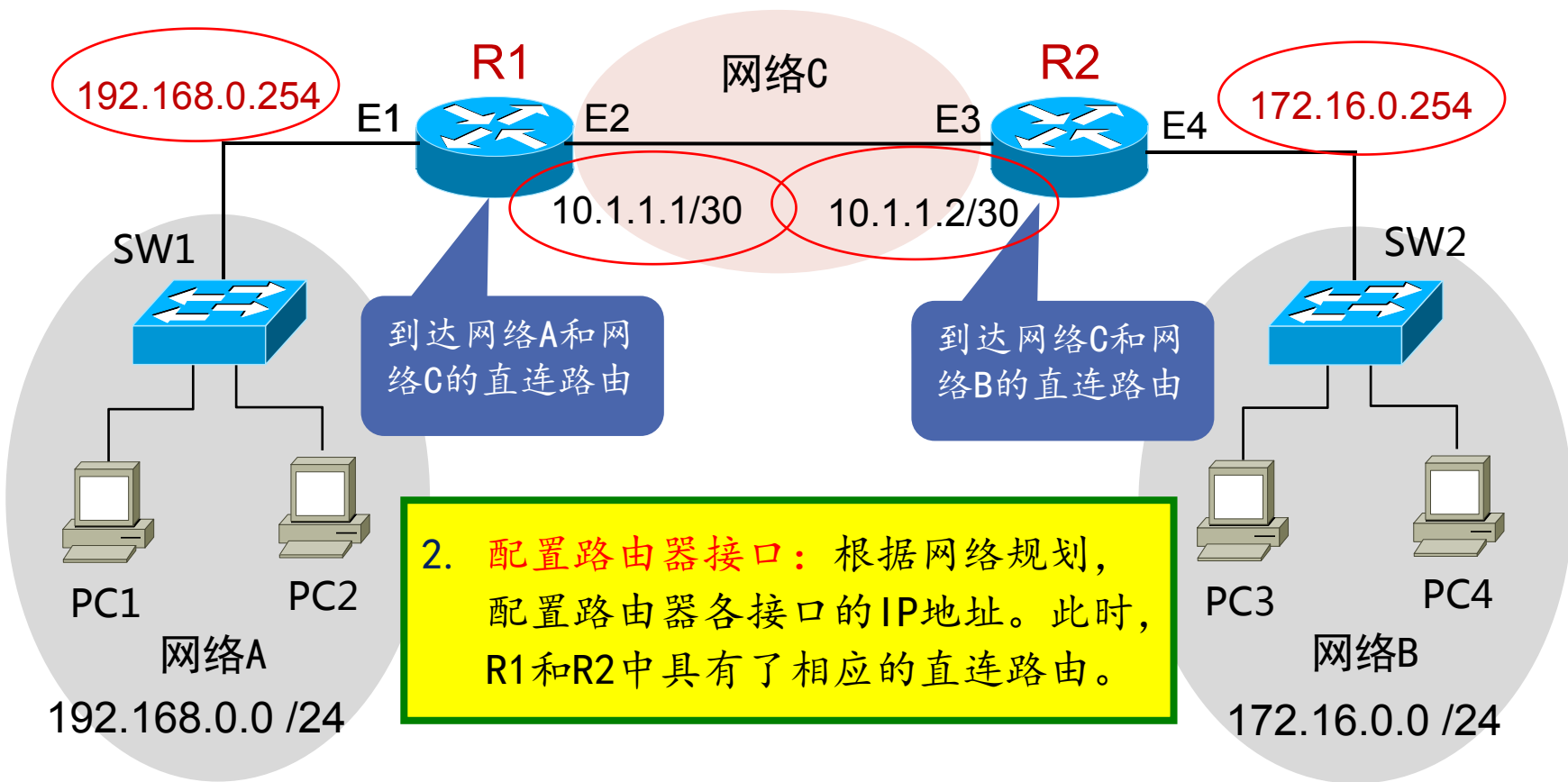
静态路由的配置与应用

(自主学习)

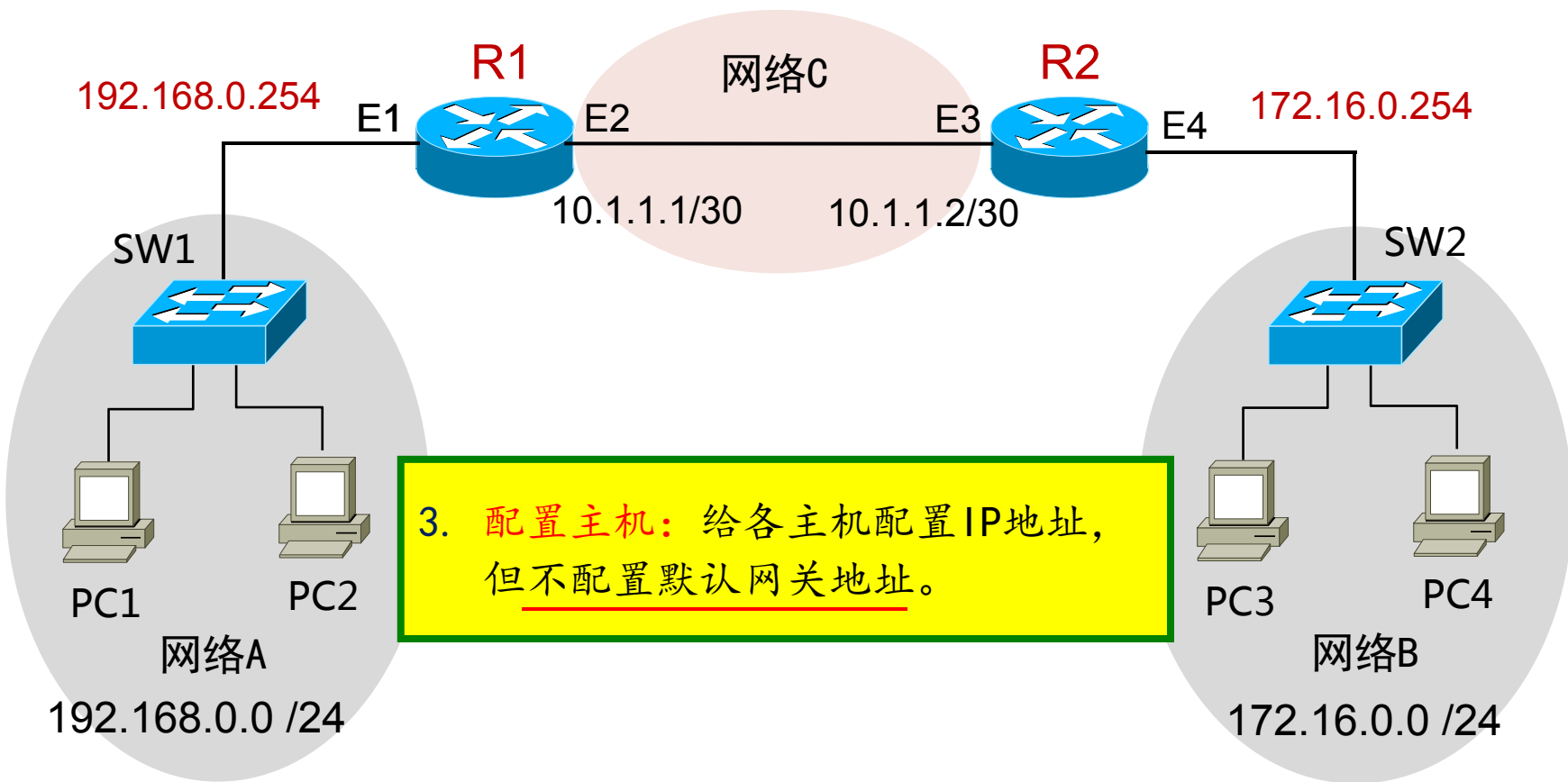
➤ 研讨：路由器组网通信分析



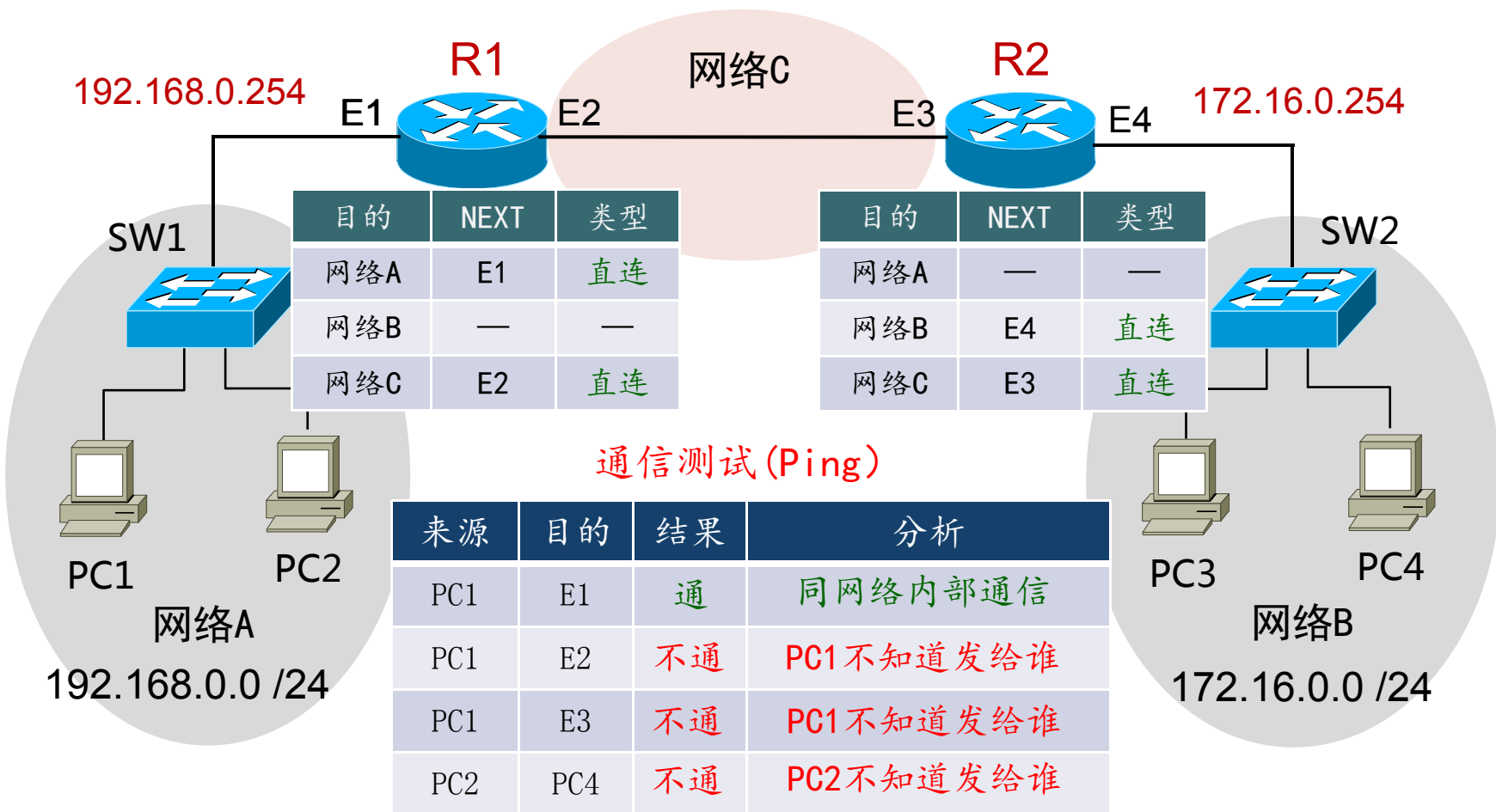
➤ 研讨：路由器组网通信分析



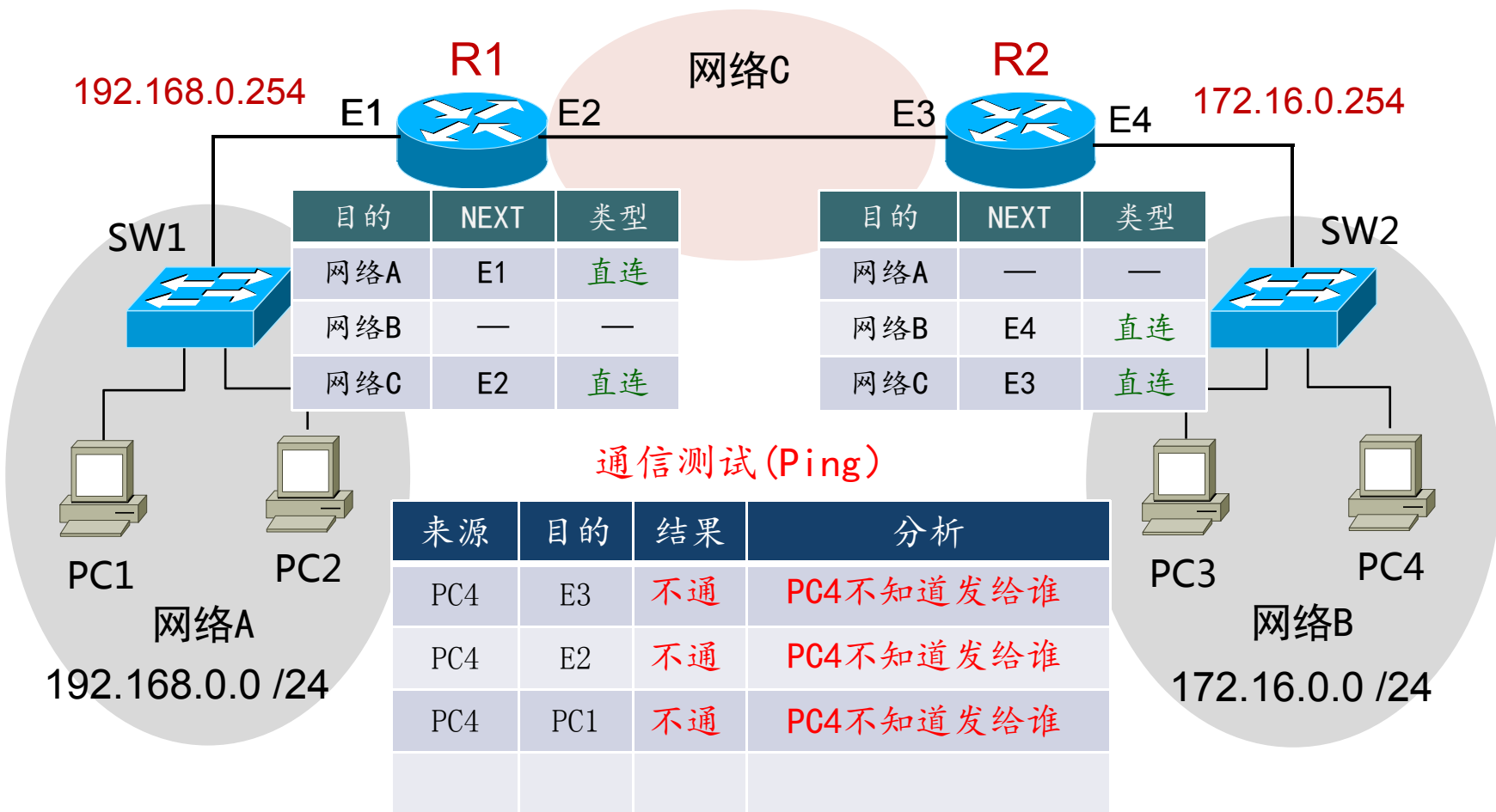
➤ 研讨：路由器组网通信分析



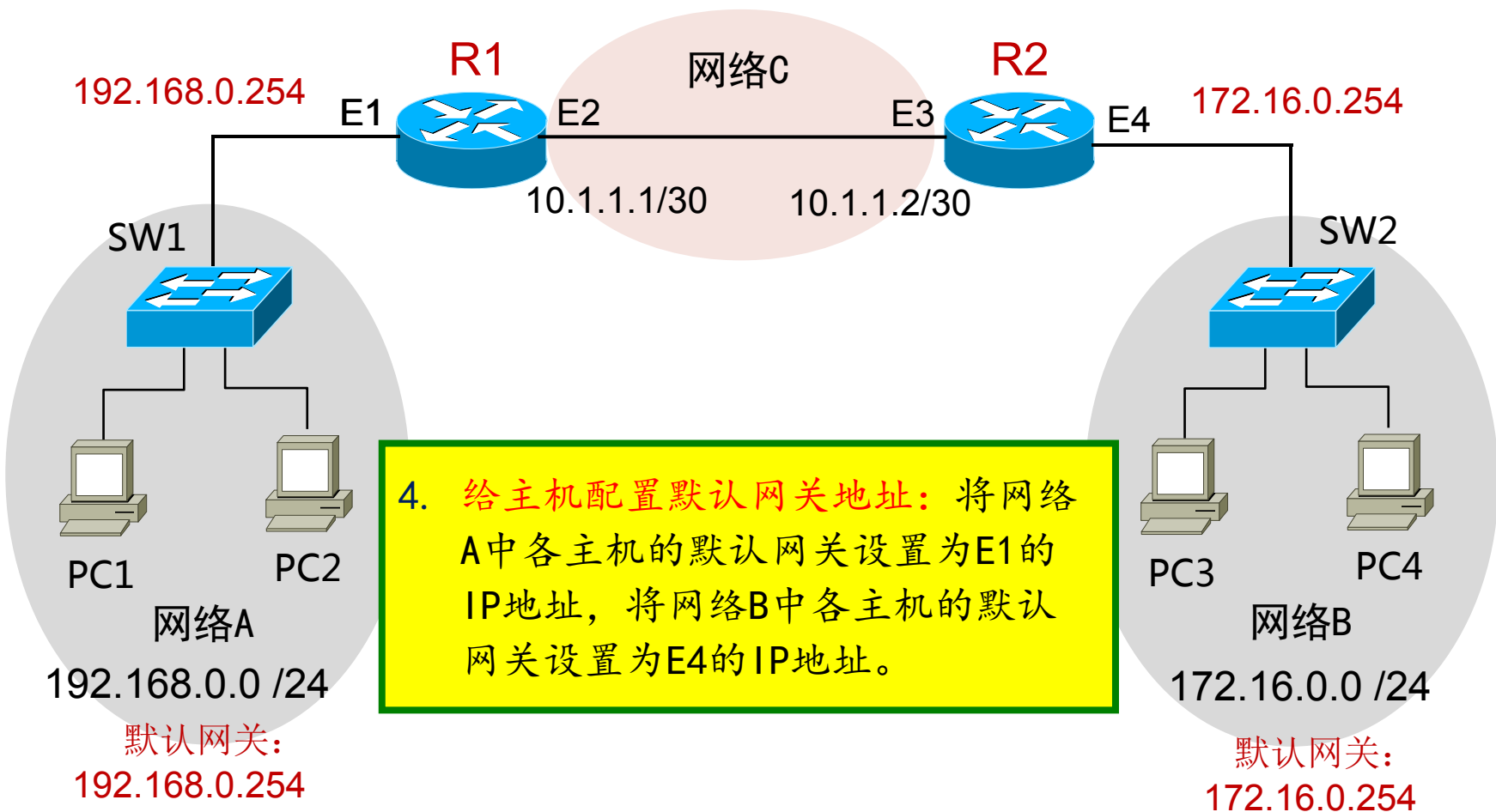
➤ 研讨：路由器组网通信分析



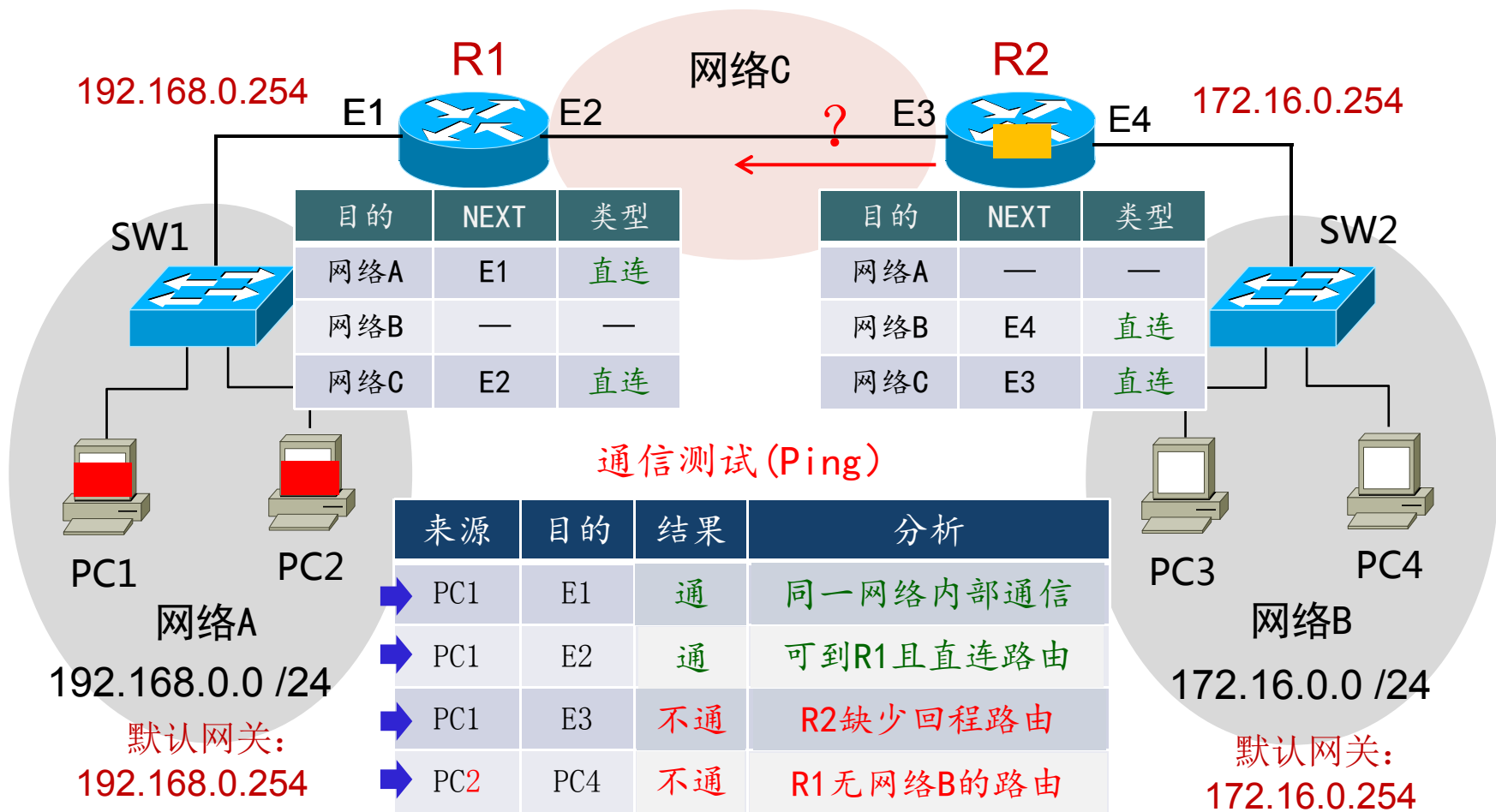
➤ 研讨：路由器组网通信分析



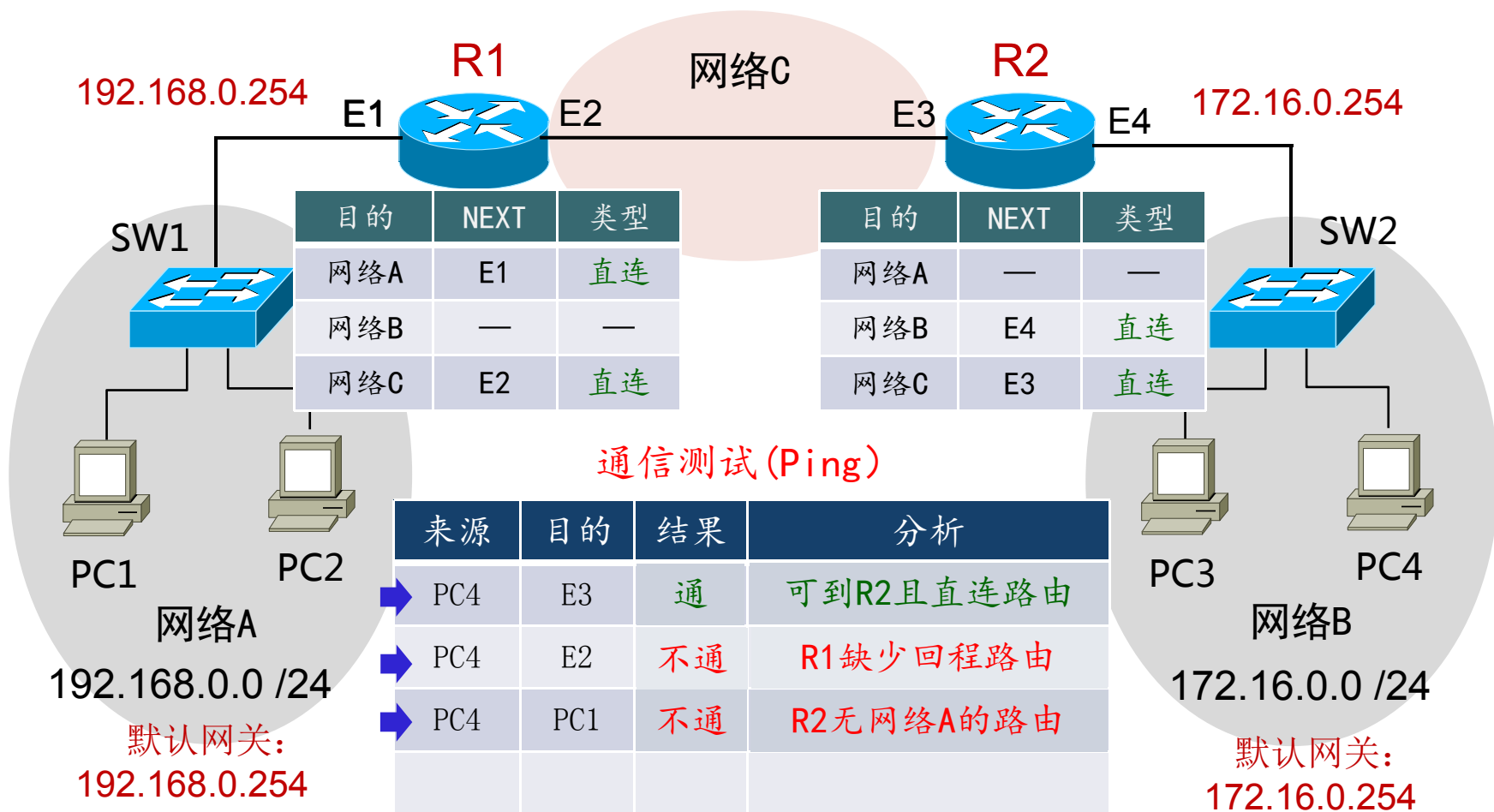
➤ 研讨：路由器组网通信分析



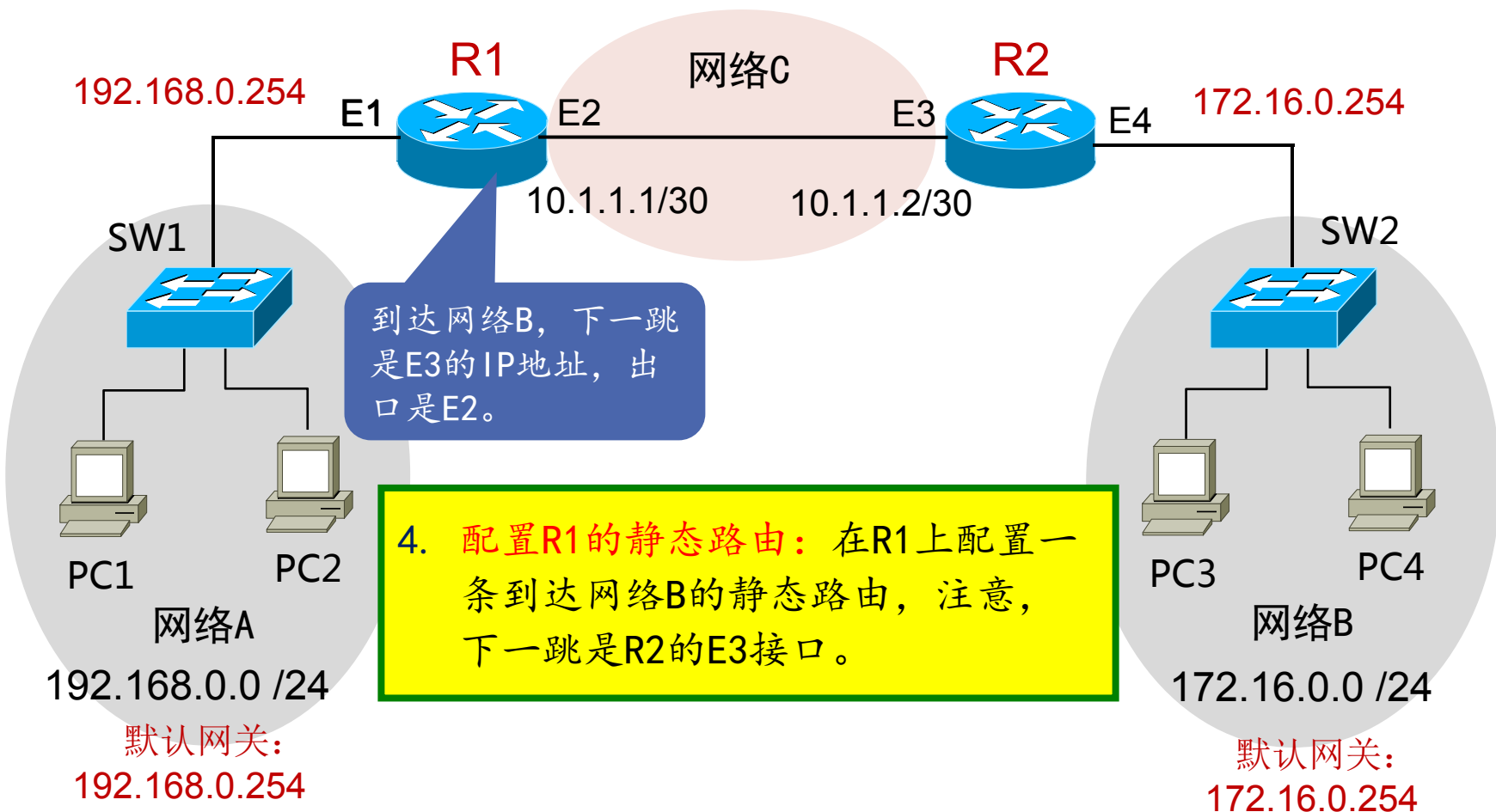
➤ 研讨：路由器组网通信分析



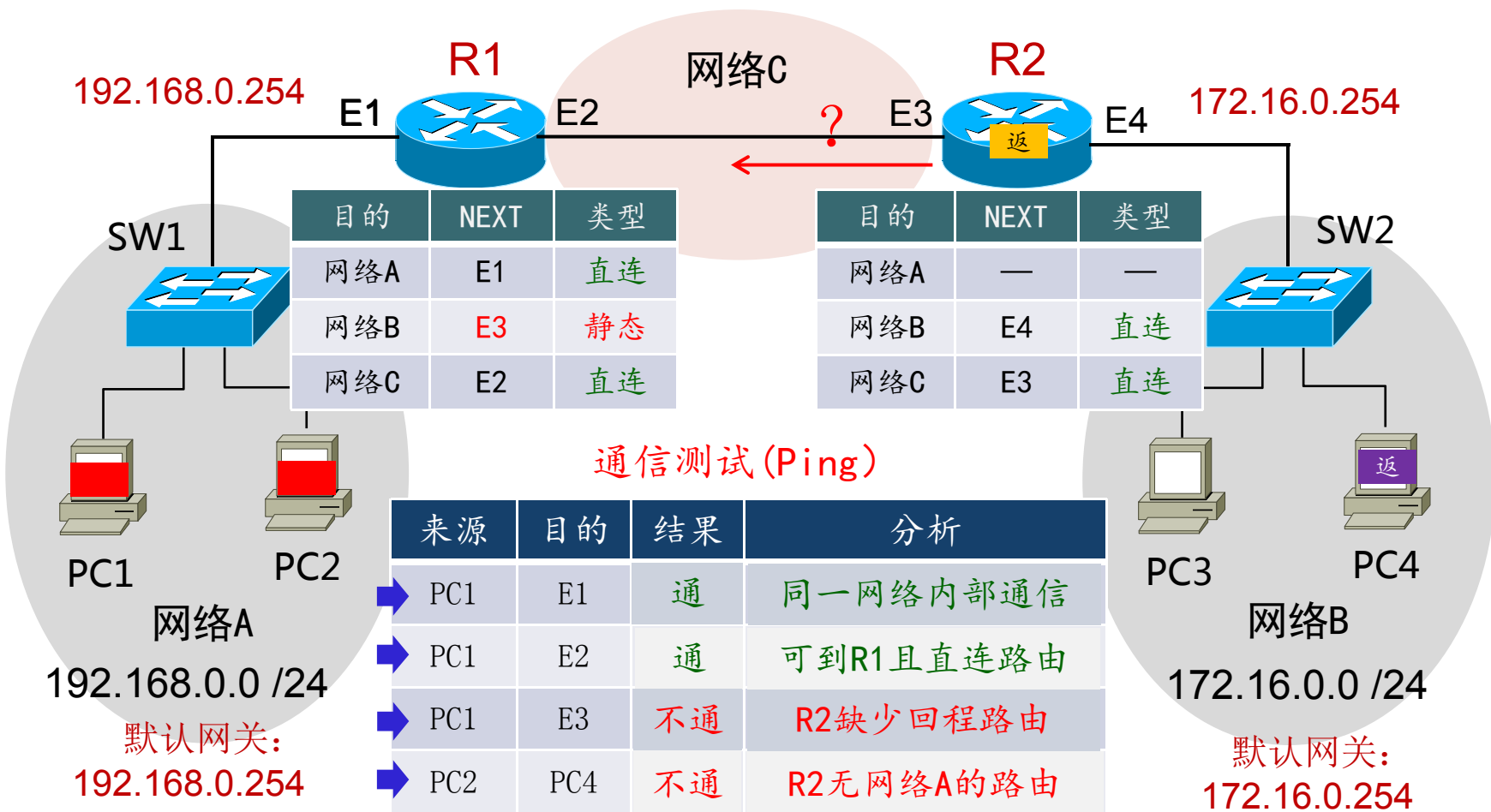
► 研讨：路由器组网通信分析



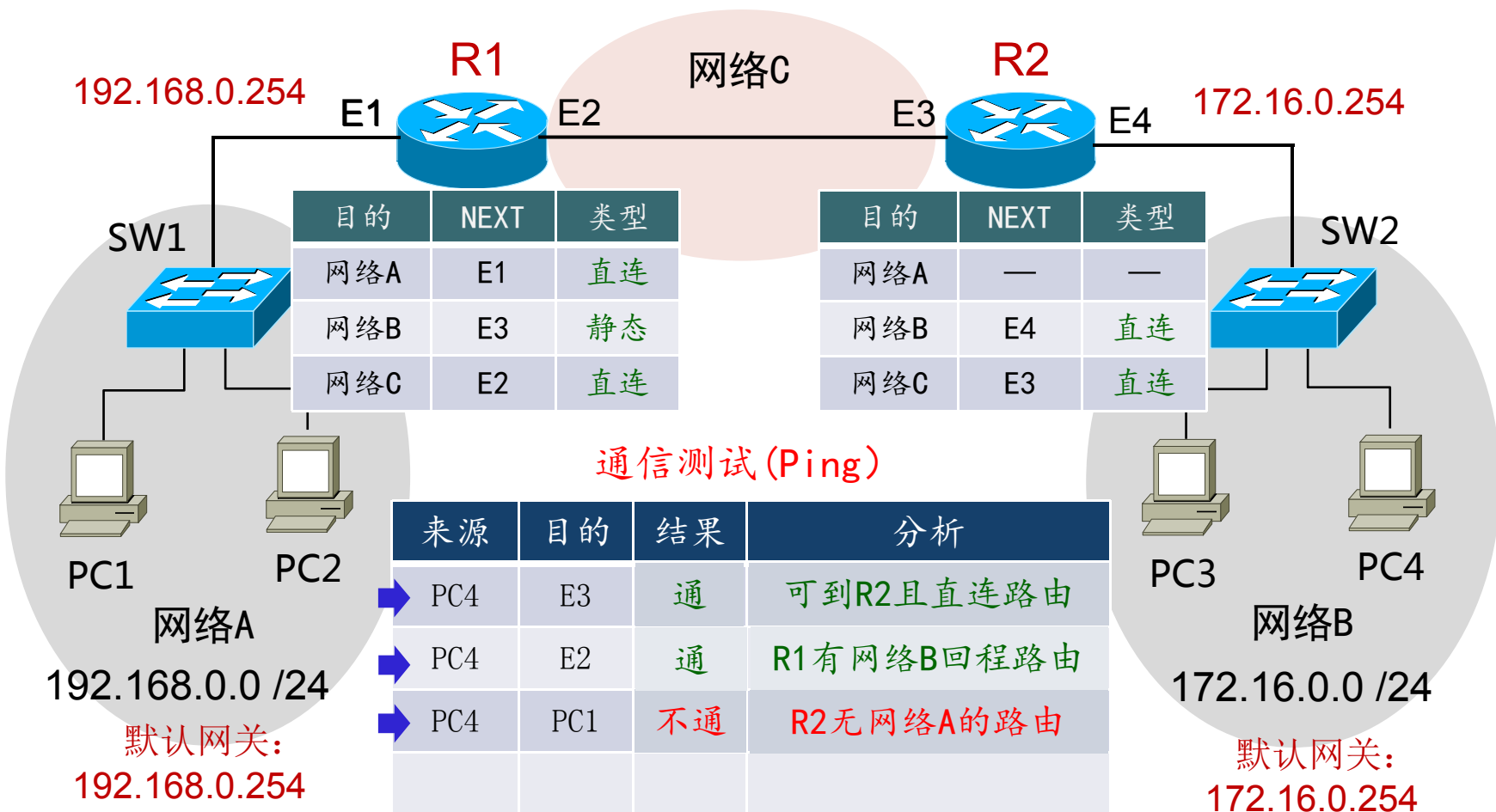
➤ 研讨：路由器组网通信分析



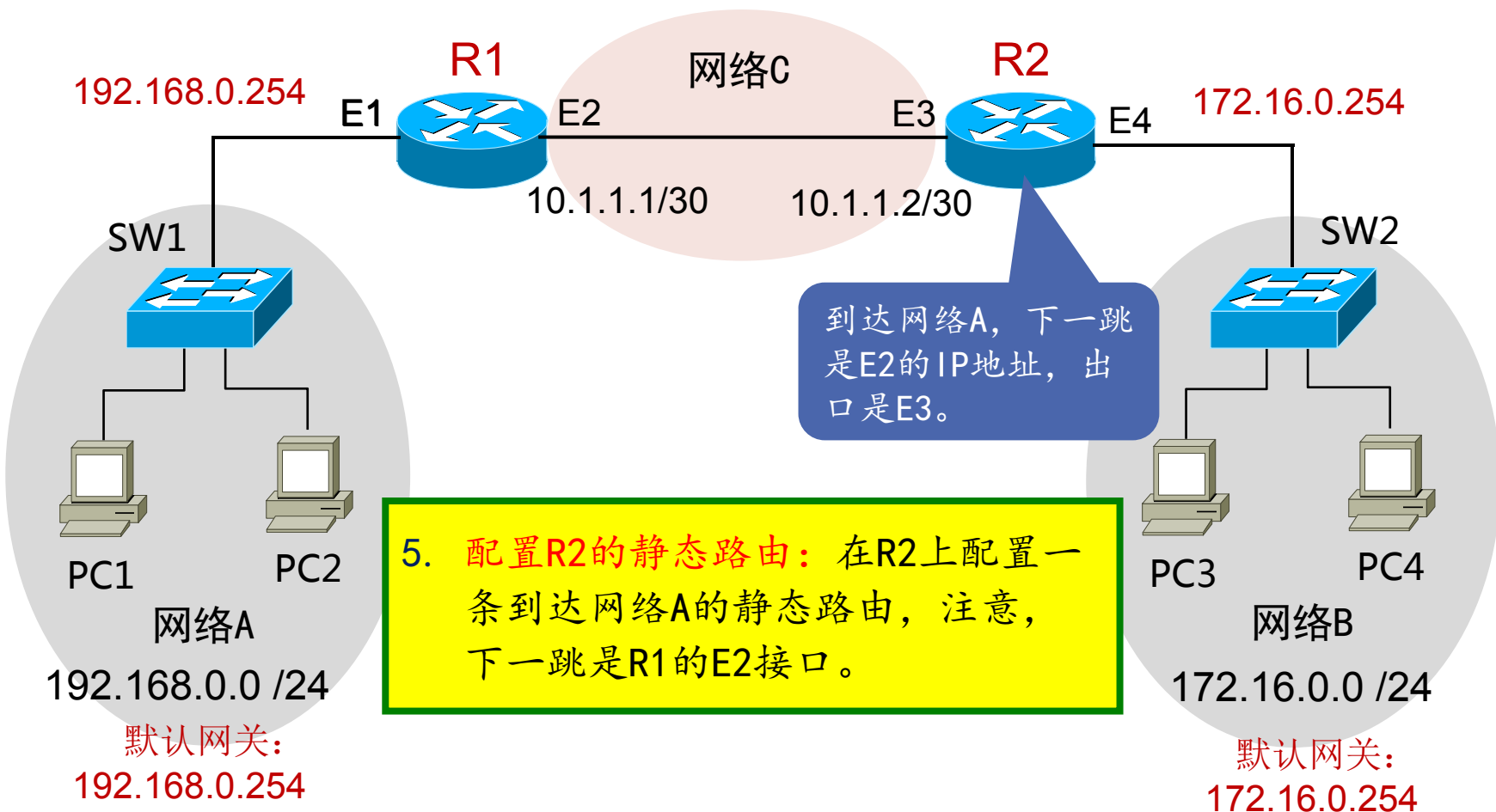
➤ 研讨：路由器组网通信分析



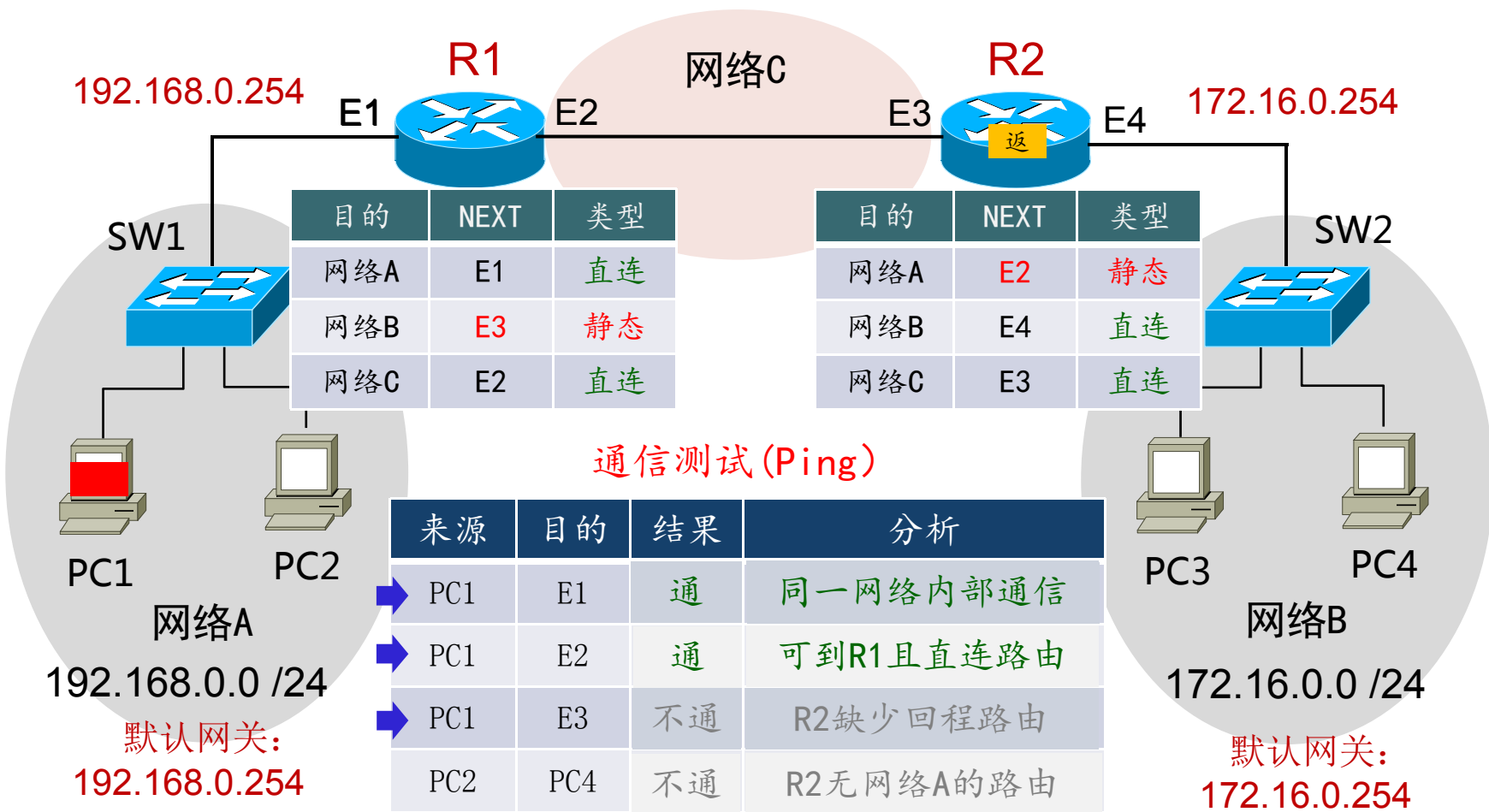
➤ 研讨：路由器组网通信分析



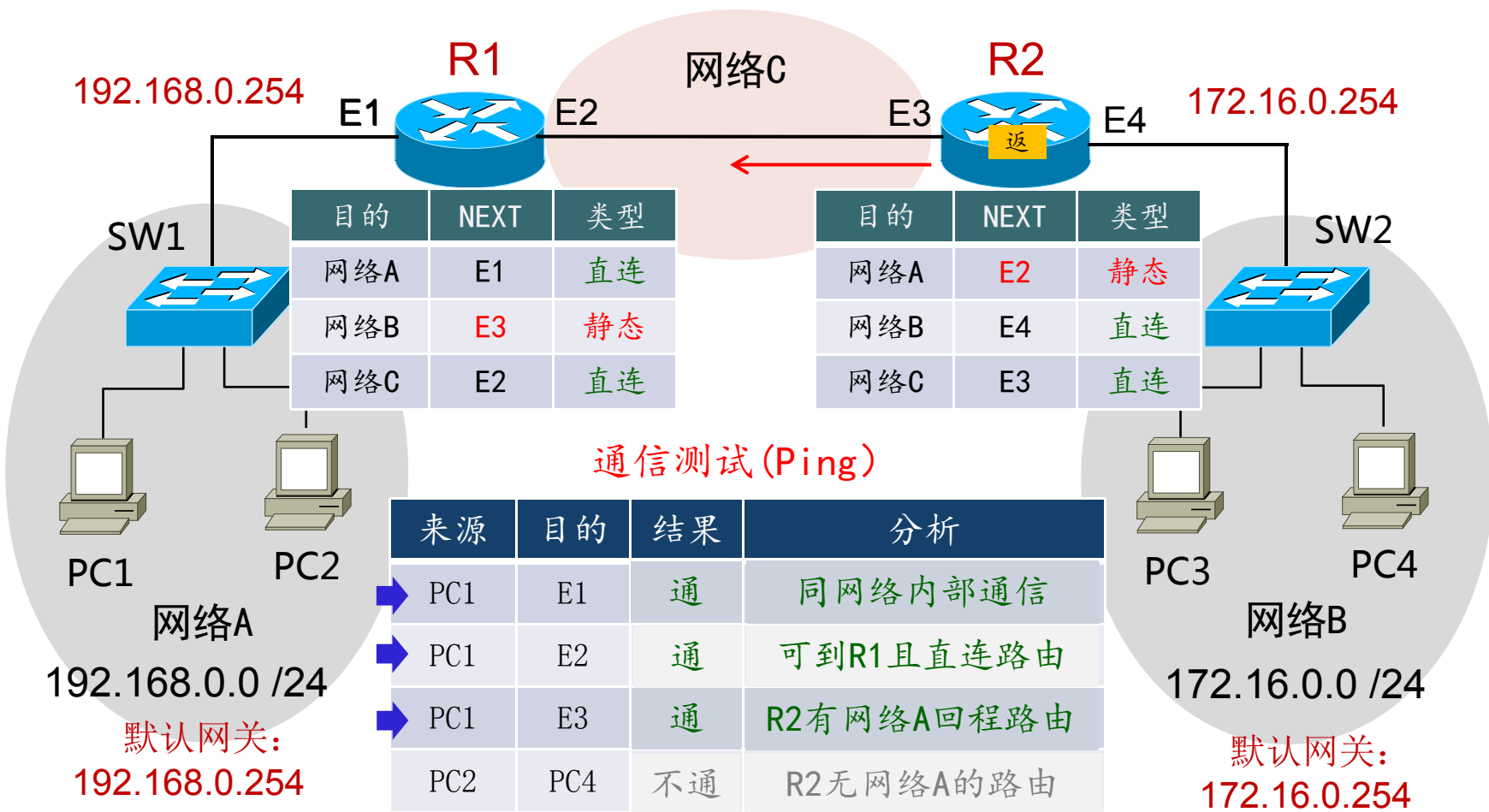
➤ 研讨：路由器组网通信分析



➤ 研讨：路由器组网通信分析



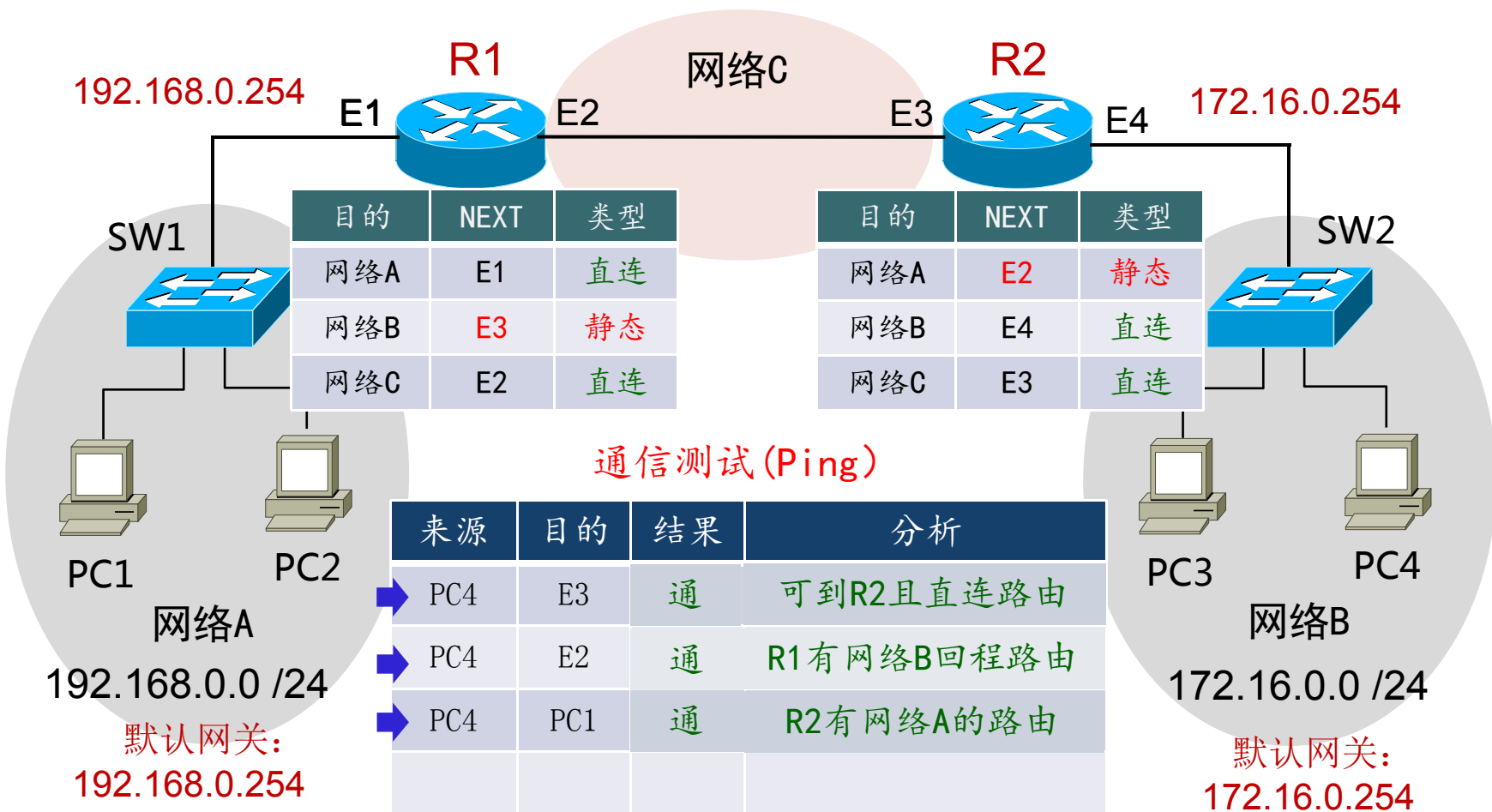
➤ 研讨：路由器组网通信分析



➤ 研讨：路由器组网通信分析



➤ 研讨：路由器组网通信分析



五、路由表的生成

—— 动态路由选择策略

动态路由选择策略

□ 认识动态路由协议

■ 动态路由选择策略，是指路由器通过使用动态路由协议，生成路由表的过程。

■ 动态路由协议

➤ 是用于路由器之间**交换路由信息**的协议。它基于某种路由算法（例如距离矢量算法），使网络中的路由器能够动态共享各自所具有的网络路由信息，并根据这些信息计算出从本路由器到达各个网络的最佳路径，然后将这些路径添加到各自的路由表中，即生成路由表。

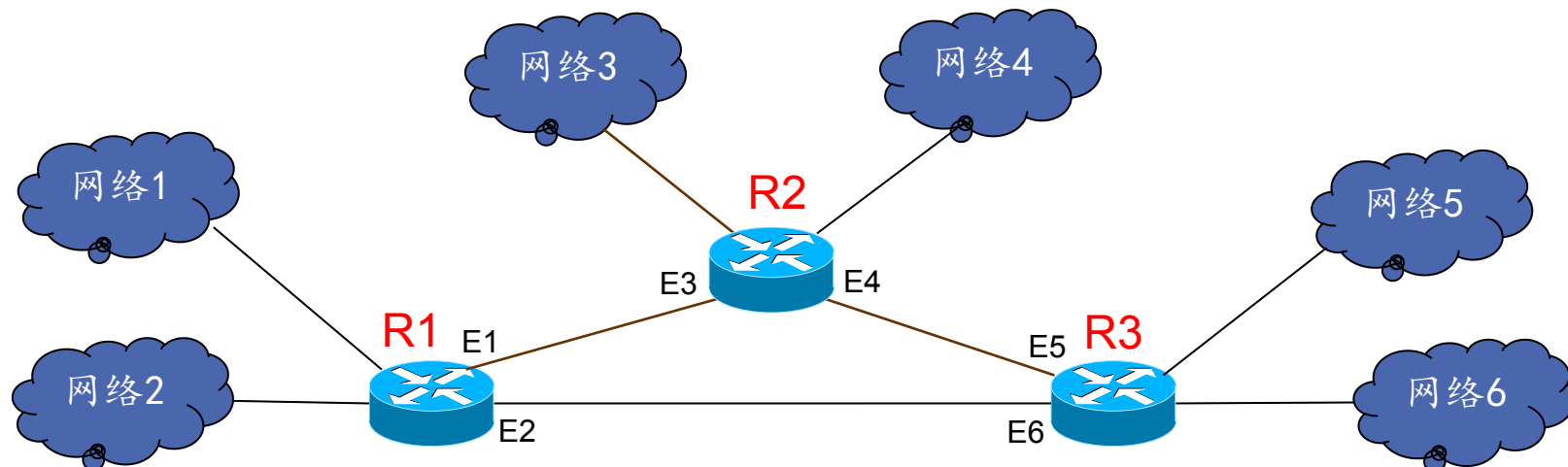
■ **简单描述**：当某园区网使用动态路由选择策略时，管理员需要首先在网络内的每台路由器上启用相同的动态路由协议（例如RIP或OSPF等），并进行一些基础配置，接下来，各台路由器上的动态路由协议就可以相互通信，最终生成路由器中的路由表！

动态路由选择策略

□ 动态路由协议的基本工作

- **路由发现与学习**：路由器通过动态路由协议向网络中的其他路由器发送和接收路由信息。每个路由器都会维护一个路由表，记录到达各个网络的路径信息。当网络拓扑发生变化时，如新增或删除网络设备、链路故障等，路由器会相互交换更新后的路由信息，以学习新的路由路径。
- **路由选择**：路由器根据收到的路由信息，结合自身的路由算法（如最短路径优先算法等），计算出到达每个目的网络的最优路径，完善自身路由表。最优路径的选择通常基于多种度量值，如跳数、带宽、延迟等。
- **路由维护**：路由器会定期发送路由更新消息，以确保路由表的准确性。当检测到网络拓扑变化时，路由器会立即发送更新消息，通知其他路由器进行相应的路由调整。
- **举例**

► 举例：动态路由的应用



R1路由表

目的	NEXT	类型
网络1	—	直连
网络2	—	直连

R2路由表

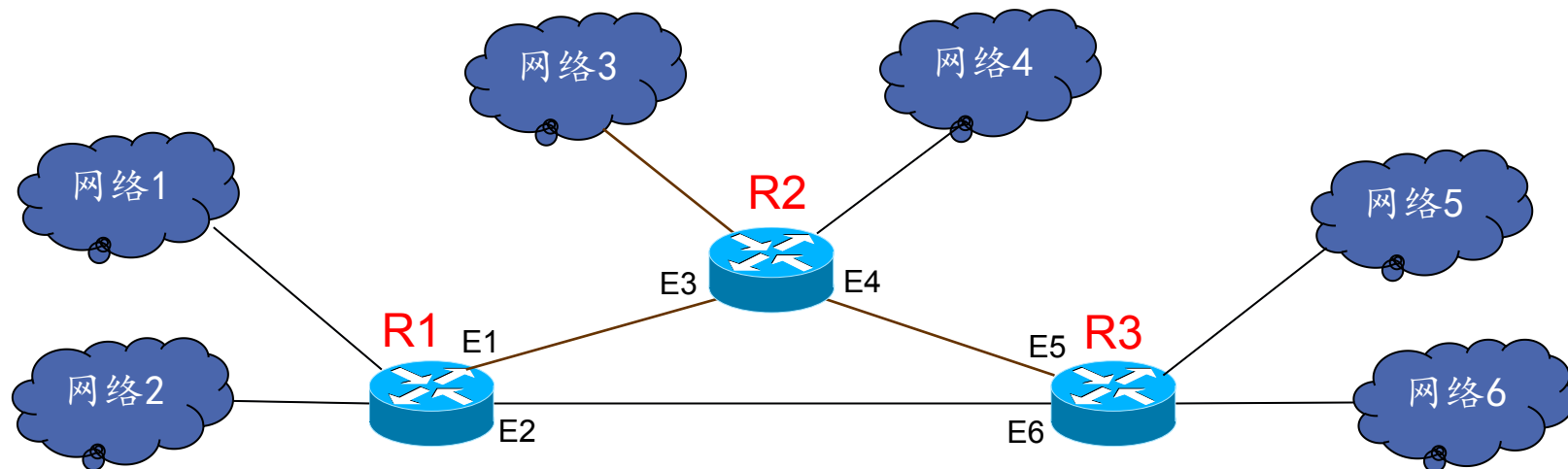
目的	NEXT	类型
网络3	—	直连
网络4	—	直连

R3路由表

目的	NEXT	类型
网络5	—	直连
网络6	—	直连

为节省篇幅，此处路由表中，省略了路由器之间的网络的直连路由

► 举例：动态路由的应用



R1路由表

目的	NEXT	类型
网络1	—	直连
网络2	—	直连

R2路由表

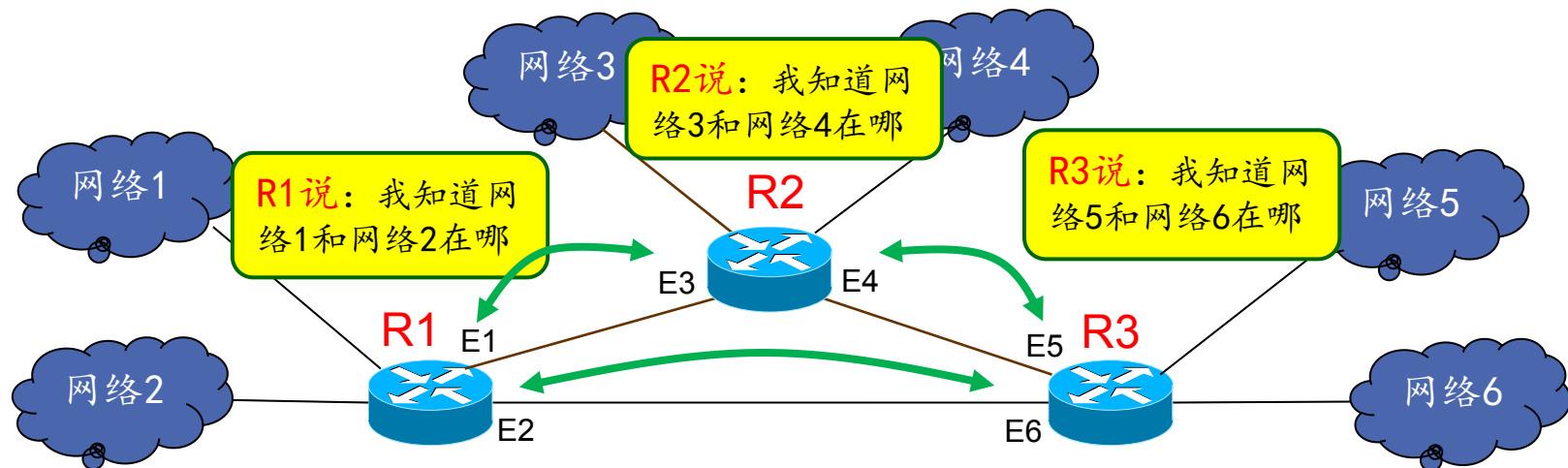
目的	NEXT	类型
网络3	—	直连
网络4	—	直连

R3路由表

目的	NEXT	类型
网络5	—	直连
网络6	—	直连

给路由器配置动态路由协议，使各路由器相互之间可以交换路由信息

► 举例：动态路由的应用



R1路由表

目的	下一跳	类型
网络1	—	直连
网络2	—	直连

R2路由表

目的	下一跳	类型
网络3	—	直连
网络4	—	直连

R3路由表

目的	下一跳	类型
网络5	—	直连
网络6	—	直连

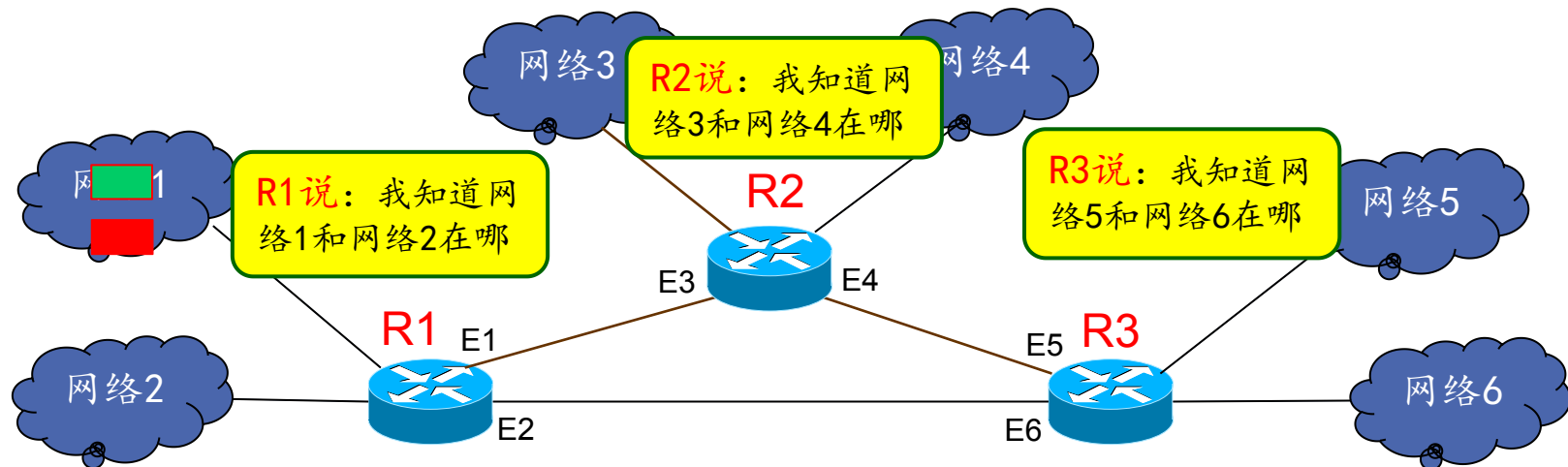
通过动态路由协议，各路由器相互交换自身的路由信息，使各路由都知道了到达所有目的网络的路由信息

网络6 E6 动态

网络6 E5 动态

网络2 E2 动态

➤ 举例：动态路由的应用

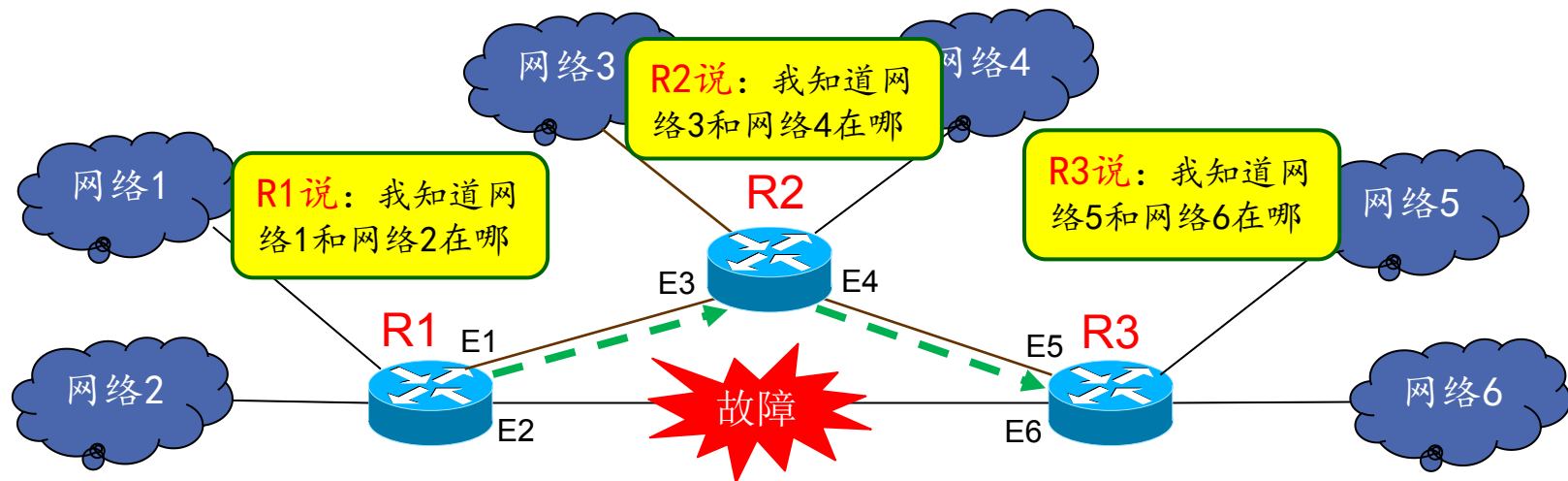


□ 问题：从网络1到达网络6，该如何走？

- 方案1：网络1→R1 →R2 →R3 →网络6（思考：方案1如何形成的？）
- 方案2：网络1→R1 →R3 →网络6（思考：方案2如何形成的？）
- 动态路由协议选择了方案2，认为这是最佳路径，并将相关信息写入R1路由表。
- 因此，当R1收到目的地是网络6的数据包时，就会转发至R3，而不是转发给R2。

通过动态路由协议的路由选择策略，可以确定“最佳”路径

► 举例：动态路由的应用



□ 问题：若R1和R3之间的链路出现故障，网络1访问网络6，该如何走？

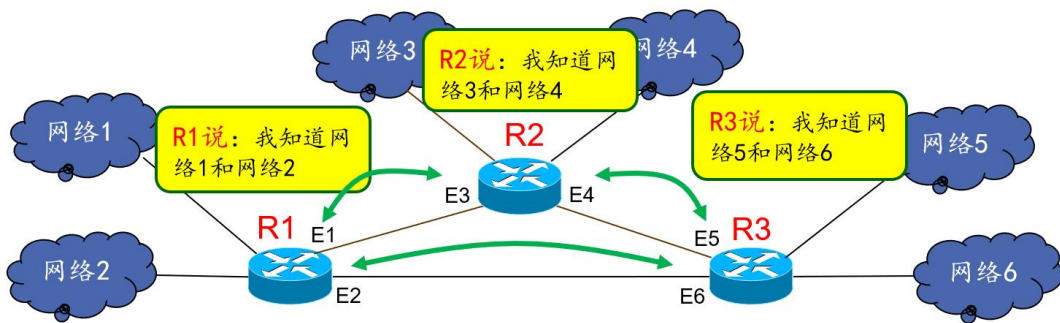
- 由于动态路由协议的作用，网络中的路由器会自动发现这一故障（即网络结构的变化），重新计算路由，并且自动更新路由器中的路由表。
- 此处，路由器R1中的动态路由协议，会根据网络结构变化，更新自己的路由表，将访问网络6的数据包，先转发至R2，即选择了方案1。这一过程不需要人工干预。
- 因此，当R1再转发目的地是网络6的数据包时，下一跳就是R2，而不是R3。

动态路由选择策略

□ 动态路由的特点

■ 自动通告

- 与静态路由不同，静态路由表项是一个路由器私有的，别的路由器并不知道。但是配置动态路由协议的路由器，可以把自己所知道的路由信息，以某种方式通告给其他相邻路由器。正是通过这种交互方式，使得路由器最终获取到整个网络内的路由信息。



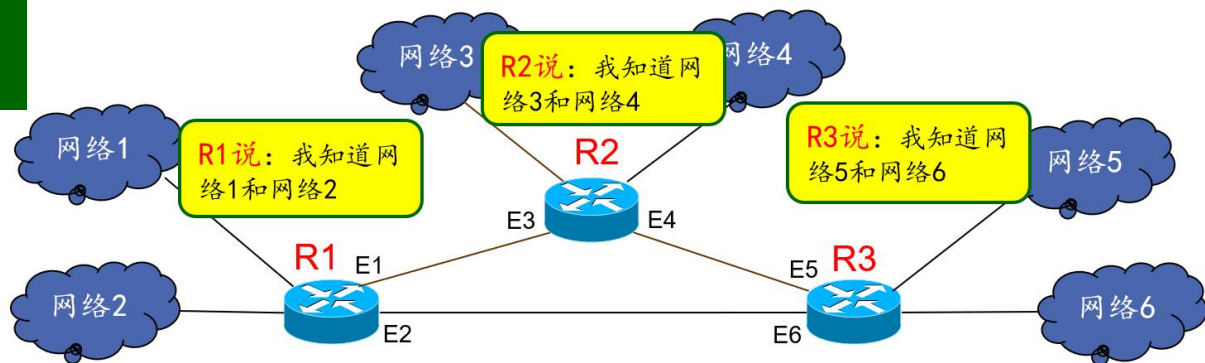
动态路由选择策略

□ 动态路由的特点

■ 自动生成

- 在“自动通告”的基础上，网络中各路由器利用从其他路由器那里收到的路由信息，生成自己的路由表，管理员无需一一手动创建。

R1中会自动生成到达网络3、网络4、网络5、网络6的动态路由记录。

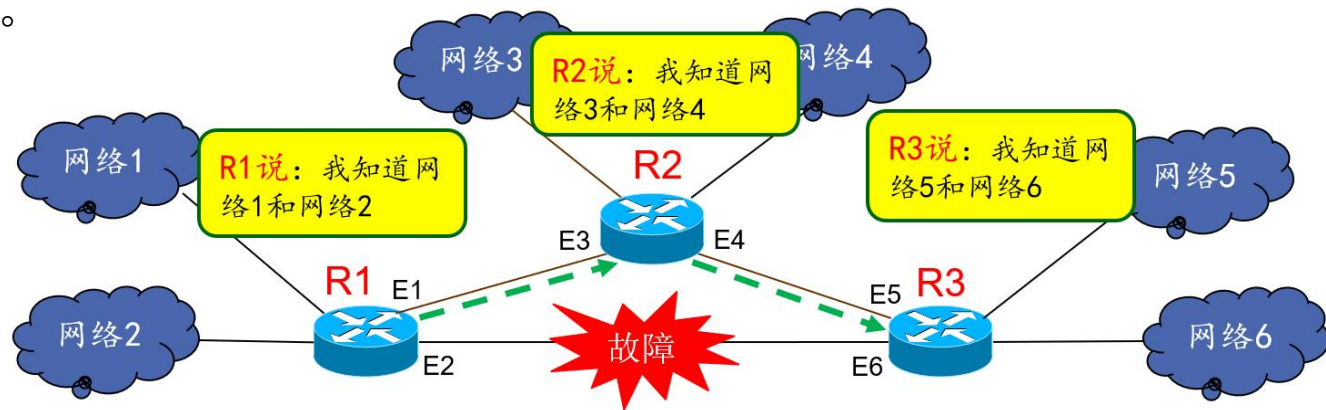


动态路由选择策略

□ 动态路由的特点

■ 自动更新

- 当网络结构发生改变，静态路由是无法自动跟着改变的。但动态路由协议可以根据网络拓扑结构的变化，自动重新计算网络内的路由信息，从而生成新的路由表项，同时还会自动删除无效的动态路由表项，更加方便路由管理。

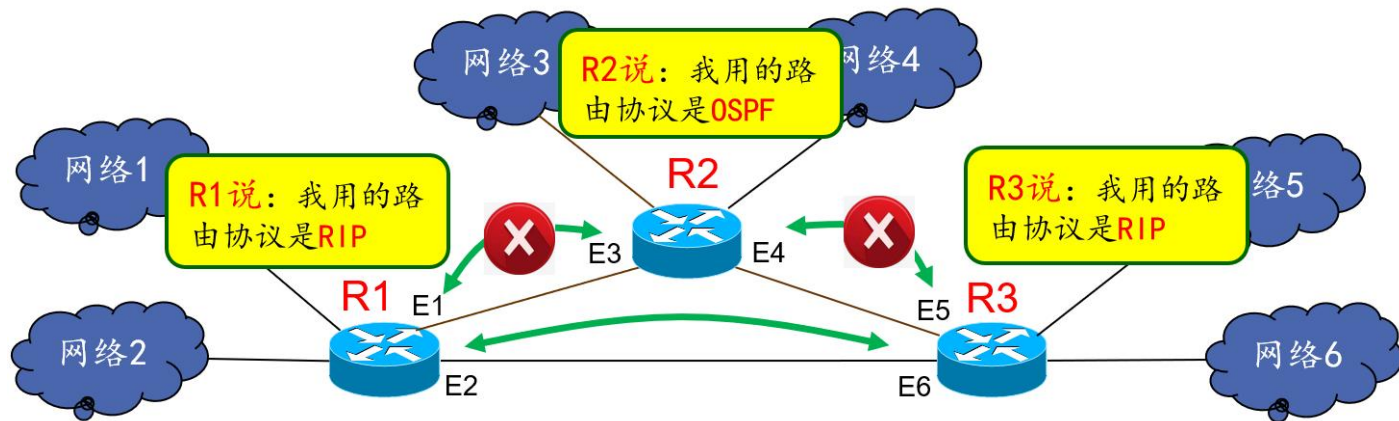


动态路由选择策略

□ 动态路由的特点

■ 不同动态路由协议不兼容

- 不同的动态路由协议适用的网络环境不一样，也是不兼容的。



动态路由协议案例

RIP的应用

动态路由选择策略 —— RIP的应用

□ RIP协议的概念

- 英文全称：Routing Information Protocol，路由信息协议
- RIP 是内部网关协议，用于AS（自治系统）内部通信。
- RIP采用距离矢量路由算法，所谓的“距离”也称为“跳数”，每经过一个路由器，跳数就加1。RIP认为“最佳路由”就是跳数最小的路由。
- RIP允许一条路径最多只能包含15个路由器，当“距离”最大值为16时，即相当于不可达。
- RIP协议有两个版本，RIPv1和RIPv2。和RIPv1相比，RIPv2支持路由汇总（聚合）、无类别域间路由（CIDR）和可变长子网掩码（VLSM）

动态路由选择策略 —— RIP的应用

▣ RIP的配置（以华为路由器为例）【结合实验学习】

1. 配置路由器接口IP地址

- 首先，需要在路由器上配置各接口的IP地址，这会生成最基本的直连路由。

2. 启动RIP进程

- 进入系统视图。
- 执行rip [process-id]命令启动RIP进程，并进入RIP视图。如果不指定process-id，则默认取值为1。process-id在同一网络中的不同路由器上必须相同，但在不同网络中可以不同。

动态路由选择策略 —— RIP的应用

▣ RIP的配置（以华为路由器为例）

3. 宣告网段信息

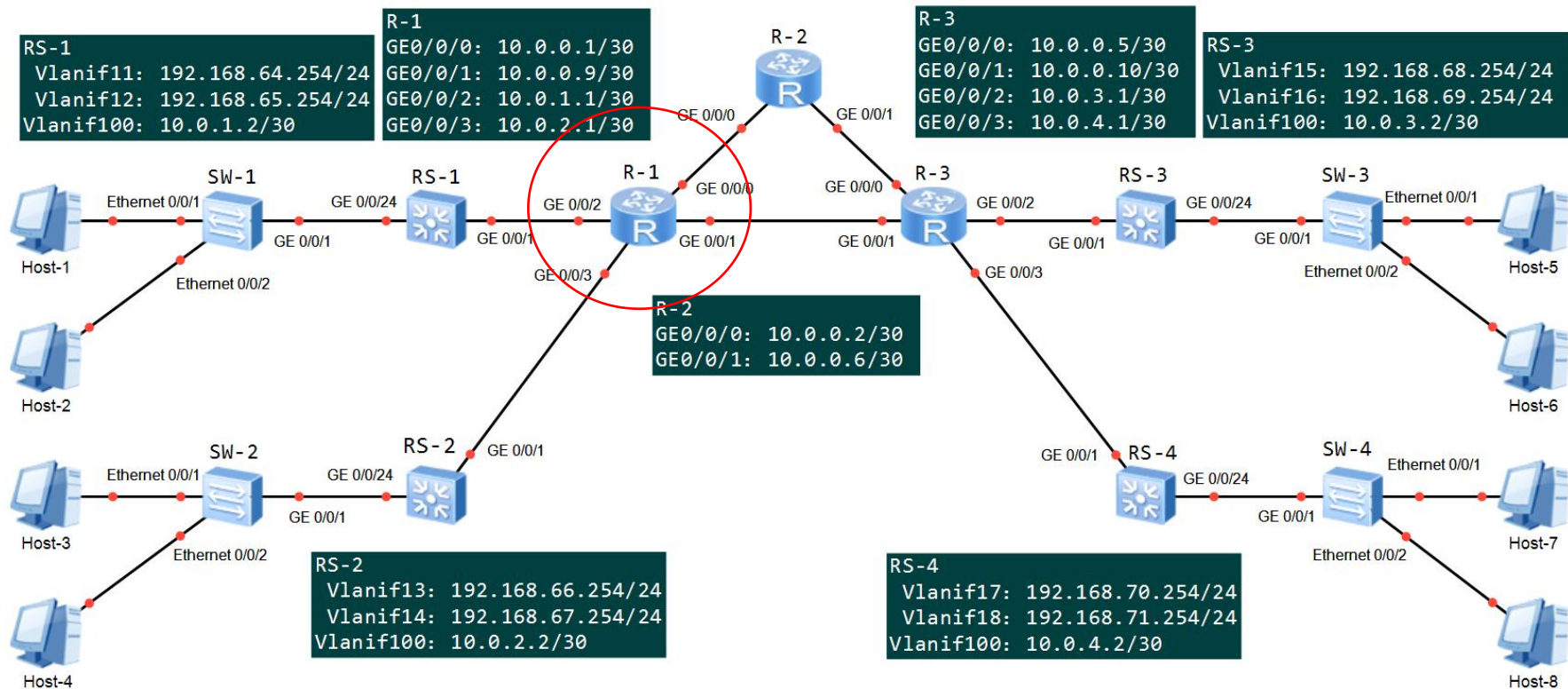
- 在RIP视图中，通过`network network-address`命令宣告路由器直连的网段。
- 这里的`network-address`必须是自然网段的网络地址（按照分类IP地址区分网段，例如10.0.1.0/24和10.1.0.0/24会被认为是同一个自然网段，即10.0.0.0/8），且路由器上至少有一个接口属于该网段。RIP路由器只将宣告的网段信息发送给邻居路由器。

4. 配置RIP版本（可选）

- 通常使用RIPv2版本，以便支持无类别域间路由（CIDR）、认证机制等。如果需要使用RIPv2，可以在RIP视图中通过`version 2`命令来指定。

■ 举例

◆ 举例：RIP协议的配置



◆ 举例：RIP协议的配置

[R-1]配置接口IP地址（见图）

//创建RIP进程1

[R-1]rip 1

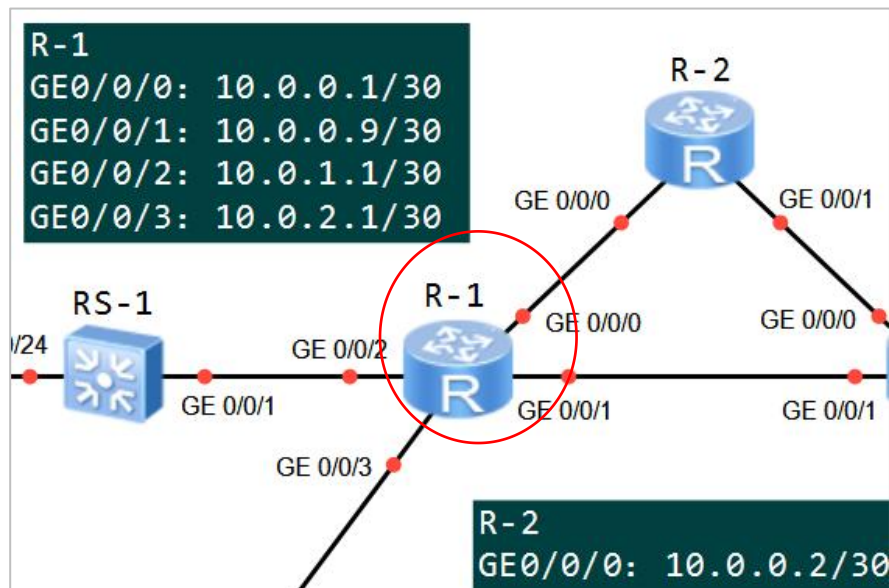
//启用RIP版本2

[R-1-rip-1]version 2

//宣告直连网络

[R-1-rip-1]network 10.0.0.0

[R-1-rip-1]quit



- 注意，此处R-1有四个直连网段，分别是10.0.0.1/30、10.0.0.8/30、10.0.1.0/30、10.0.2.0/30，但是，对于RIP协议来讲，路由器只能宣告自然网段的网络地址，即只宣告10.0.0.0。
- 但是，由于启用了RIPv2版本，支持CIDR，并且每个接口都配置了IP地址，所以会生成4条直连路由，可通过display命令查看。

动态路由协议案例

OSPF的应用

动态路由协议——OSPF

□ 开放最短路径优先 (Open Shortest Path First , OSPF)

- 由于RIP协议主要用于小型网络，在20世纪80年代中期就已不能适应大规模异构网络的互连，一种互连功能更强大的路由协议——OSPF，就随之产生了。
- OSPF只是协议名字，并不表示其他路由选择协议不是“最短路径优先”。实际上，所有的在自治系统内部使用的路由选择协议都是要寻找最短路径的。
- OSPF协议最主要的特征就是使用链路状态路由算法，而不是像RIP那样的距离矢量路由算法。

动态路由协议——OSPF

□ OSPF的特点 (和RIP对比)

- **和谁交换路由信息**：采用洪泛法 (flooding)，路由器向所有相邻路由器发送信息，而每一个相邻路由器再将此信息发往其所有的相邻路由器，最终本自治系统中所有路由器都得到了这个信息的一个副本。（RIP仅仅向自己相邻的几个路由器发送信息）
- **交换什么信息**：发送的是与本路由器相邻的所有路由器的**链路状态**。链路状态用来说明本路由器都和哪些路由器相邻，以及该链路的度量 (metric)。
 - RIP协议中，本路由器向相邻的路由器发送的是自己的路由表（到各网络的距离和下一跳路由）
- **何时交换**：当链路状态发生变化或每隔一段时间（如30分钟），路由器才用洪泛法向所有路由器发送此信息。（与RIP相似）

动态路由协议——OSPF

□ OSPF的特点 (和RIP对比)

- 最终结果如何：OSPF会构建起链路状态数据库（Link-state Database），由于各路由器之间频繁地交换链路状态信息，因此所有的路由器最终都能建立一个链路状态数据库，这个数据库实际上就是全网的拓扑结构图，它在全网范围内是一致的（这称为链路状态数据库的同步）。
 - 所以，配置OSPF的每个路由器都知道全网共有多少个路由器，以及哪些路由器是相连的，其代价是多少。即各路由器的链路状态数据库是相同的。每个路由器使用链路状态数据库中的数据构造自己的路由表
 - 再看RIP，RIP中的每一个路由器虽然知道到达所有的网络的距离以及下一跳路由器，但却不知道全网的拓扑结构（只有到了下一跳，才能知道接下来怎样走）。

动态路由协议——OSPF

□ OSPF的区域化设计

- 这也是OSPF和RIP的一个重要区别！
- OSPF支持层次化设计，可以通过区域（Area）的概念来组织大型网络。这有助于减少路由器的内存消耗和计算负担，因为路由器只需要了解它们所在区域内部的详细信息，而不需要知道整个网络的信息。
- 接下来谈谈OSPF区域

动态路由协议——OSPF

▣ OSPF区域 (area)

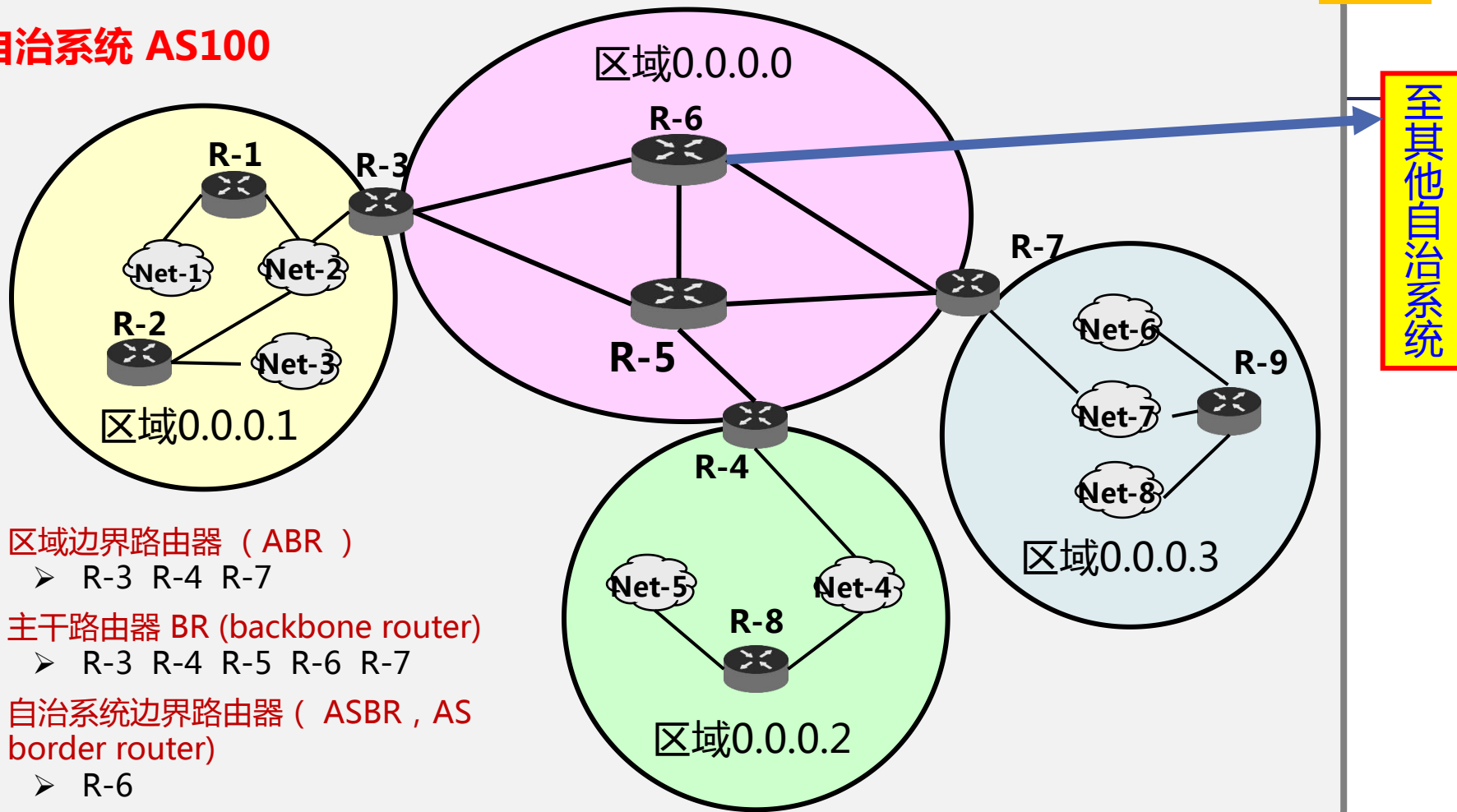
- 为了使OSPF能够用于规模很大的网络，OSPF 将一个自治系统再划分为若干个更小的范围，叫作区域。
- 划分区域的好处就是**将**利用洪泛法交换链路状态信息的**范围**局限于每一个区域而不是整个的自治系统，这就减少了整个网络上的通信量。
- 所以，在一个区域内部的路由器只需要知道本区域的完整网络拓扑，而不需知道其他区域的网络拓扑的情况。
- 那么，如何实现不同区域之间的路由通信呢？

动态路由协议——OSPF

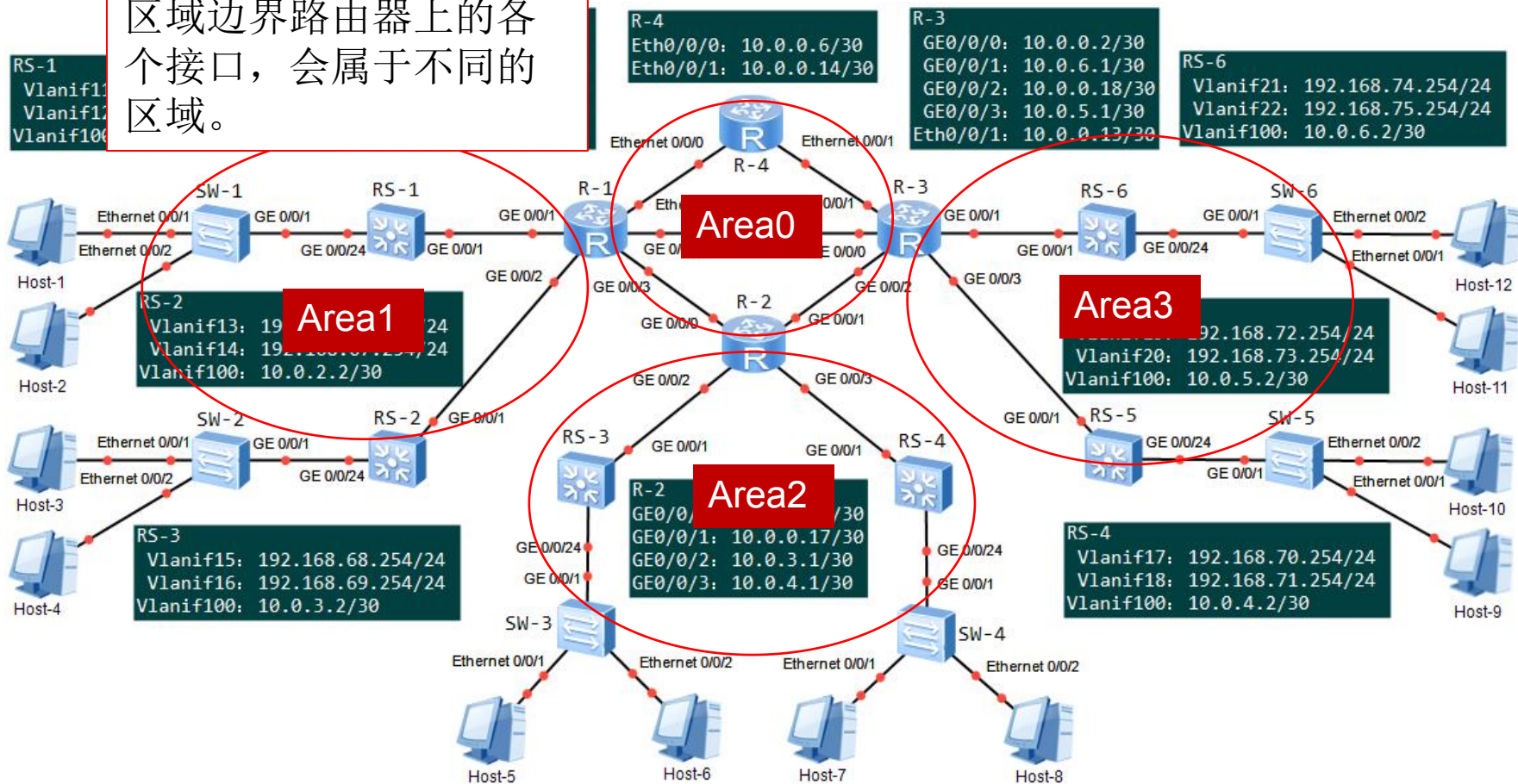
□ OSPF区域 (area)

- 为了使每个区域能够和本区域以外的区域进行通信，OSPF 使用了**主干区域** (backbone area) 和**普通区域**的概念。
- 在规划一个AS中的区域时，OSPF要设置一个主干区域和若干普通区域，并且，所有普通区域必须与主干区域直接相连。
- 每个区域有一个或若干个边界路由器，边界路由器将本区域中的路由信息进行概括后，只将基本信息发给其他区域（例如**普通2→主干→普通3**），从而既实现了不同区域之间路由信息交换，又减少了区域之间的通信量。
- 每一个区域都有一个 32 位的区域标识符，用点分十进制表示。例如主干区域的标识符规定为0.0.0.0。
- **举例**

自治系统 AS100



区域边界路由器上的各个接口，会属于不同的区域。



➤ 例1：在RS-1上配置OSPF

[RS-1]ospf 1

//创建并进入**OSPF**区域，此处是区域**1**（因为**RS-1**的接口都属于**area 1**）

[RS-1-ospf-1]area 1

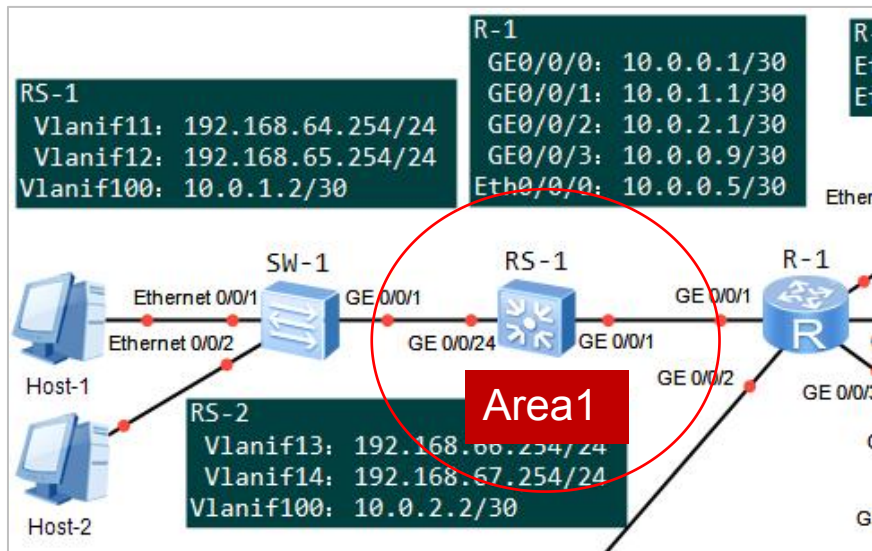
//宣告当前区域中的直连网络，注意需要配置子网掩码（与**RIP**不同）

[RS-1-ospf-1-area-0.0.0.1]network 192.168.64.0 0.0.0.255

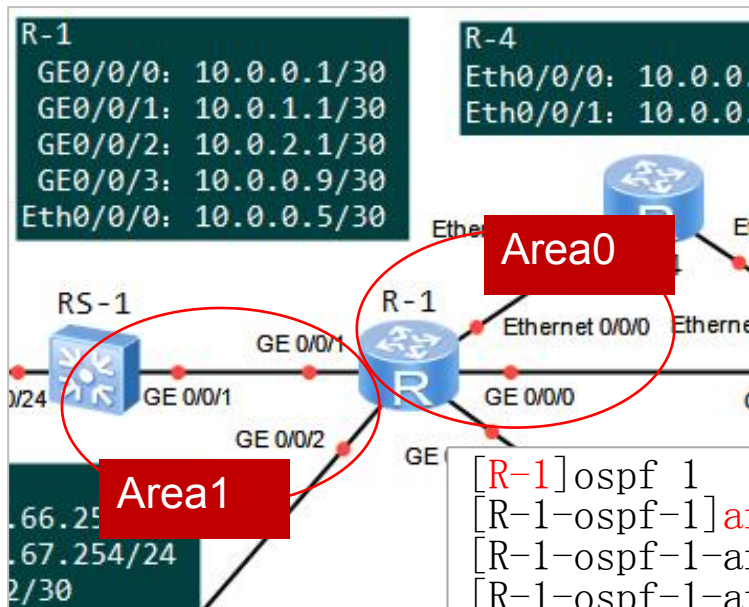
[RS-1-ospf-1-area-0.0.0.1]network 192.168.65.0 0.0.0.255

[RS-1-ospf-1-area-0.0.0.1]network 10.0.1.0 0.0.0.3

注意，ospf在宣告直连网络时，需要加上子网掩码，且以反码形式



► 例2：在R-1上配置OSPF



注意，R-1是区域边界路由器，有5个直连网络，其中3个属于area0, 2个属于area1，需要分别宣告。

```

[R-1]ospf 1
[R-1-ospf-1]area 0
[R-1-ospf-1-area-0.0.0.0]network 10.0.0.0 0.0.0.3
[R-1-ospf-1-area-0.0.0.0]network 10.0.0.4 0.0.0.3
[R-1-ospf-1-area-0.0.0.0]network 10.0.0.8 0.0.0.3
[R-1-ospf-1-area-0.0.0.0]quit
[R-1-ospf-1]area 1
[R-1-ospf-1-area-0.0.0.1]network 10.0.1.0 0.0.0.3
[R-1-ospf-1-area-0.0.0.1]network 10.0.2.0 0.0.0.3
[R-1-ospf-1-area-0.0.0.1]quit
  
```

第二讲 完