



Data Communications and Networking

Fourth Edition

Forouzan

第 22 章

传递、转发和路由选择

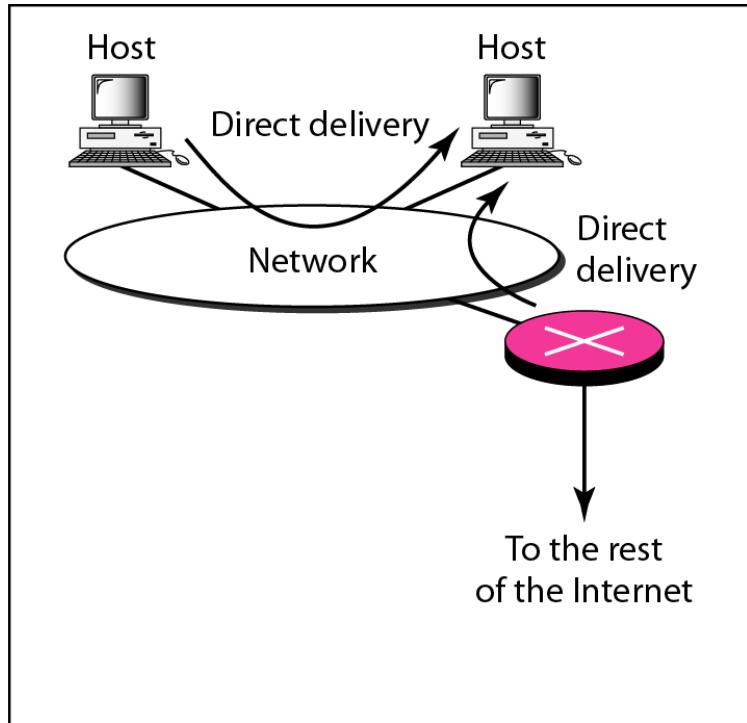
22-1 传递

网络层负责用底层物理网络处理分组，定义这种处理为分组的传递。

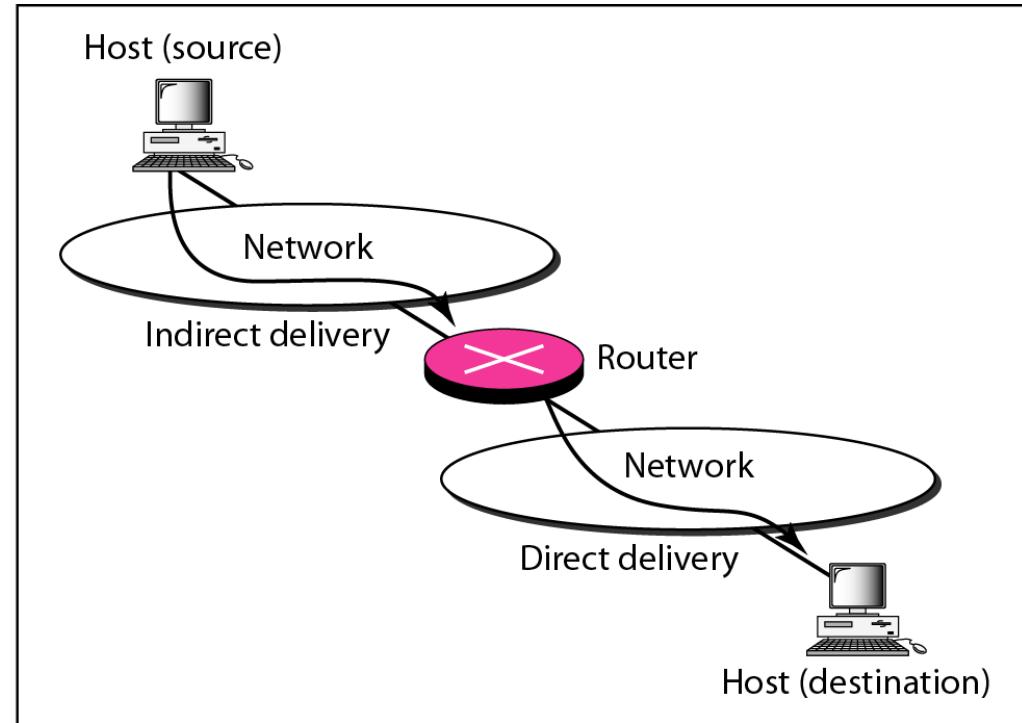
本节讨论：

直接传递和间接传递

图 22.1 直接传递和间接传递



a. Direct delivery



b. Indirect and direct delivery

转发是指将分组路由到它的目的端。转发要求主机或路由器有一个路由表。当主机有分组要发送时，或是路由器已收到一个分组要转发时，就要查找路由表以便求得到达最终目的端的路由。

本节讨论：

转发技术
转发过程
路由表

图 22.2 路由方法与下一跳方法

a. Routing tables based on route

Destination	Route
Host B	R1, R2, host B

Routing table
for host A

Destination	Route
Host B	R2, host B

Routing table
for R1

Destination	Route
Host B	Host B

Routing table
for R2

b. Routing tables based on next hop

Destination	Next hop
Host B	R1

Destination	Next hop
Host B	R2

Destination	Next hop
Host B	---

Host A



Network

R1

Network

R2

Network

Host B



图 22.3 特定主机方法与特定网络方法

Routing table for host S based
on host-specific method

Destination	Next hop
A	R1
B	R1
C	R1
D	R1

Routing table for host S based
on network-specific method

Destination	Next hop
N2	R1

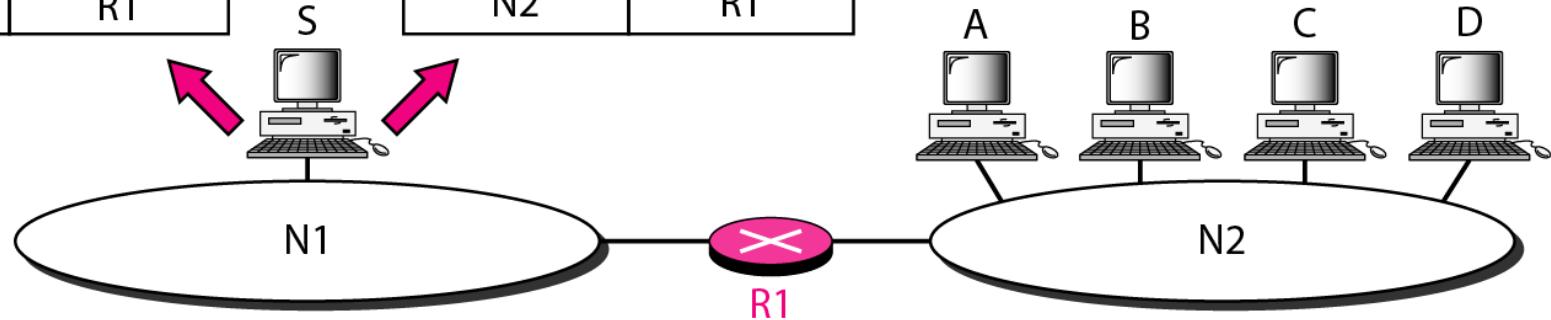


图 22.4 默认方法

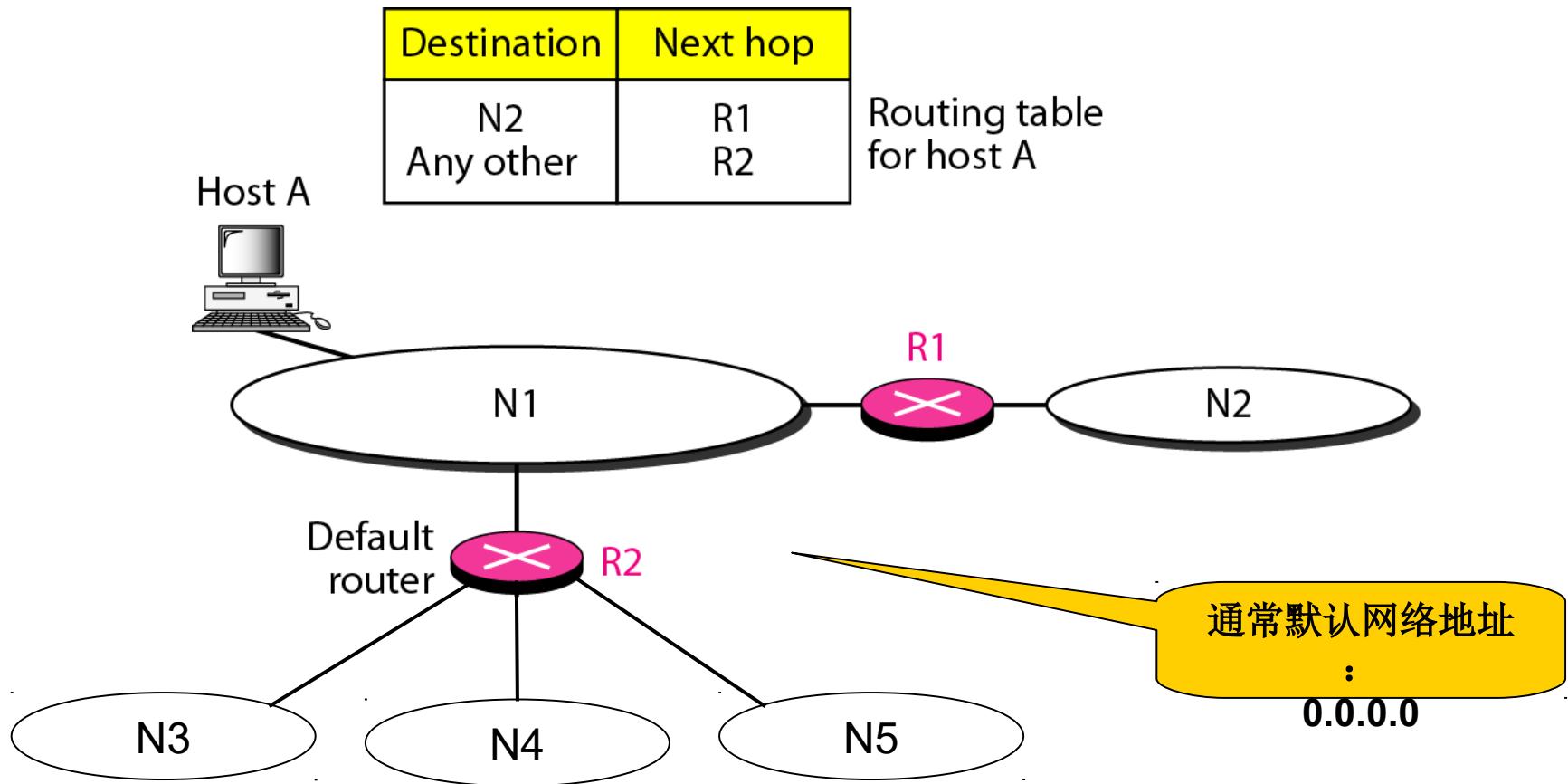
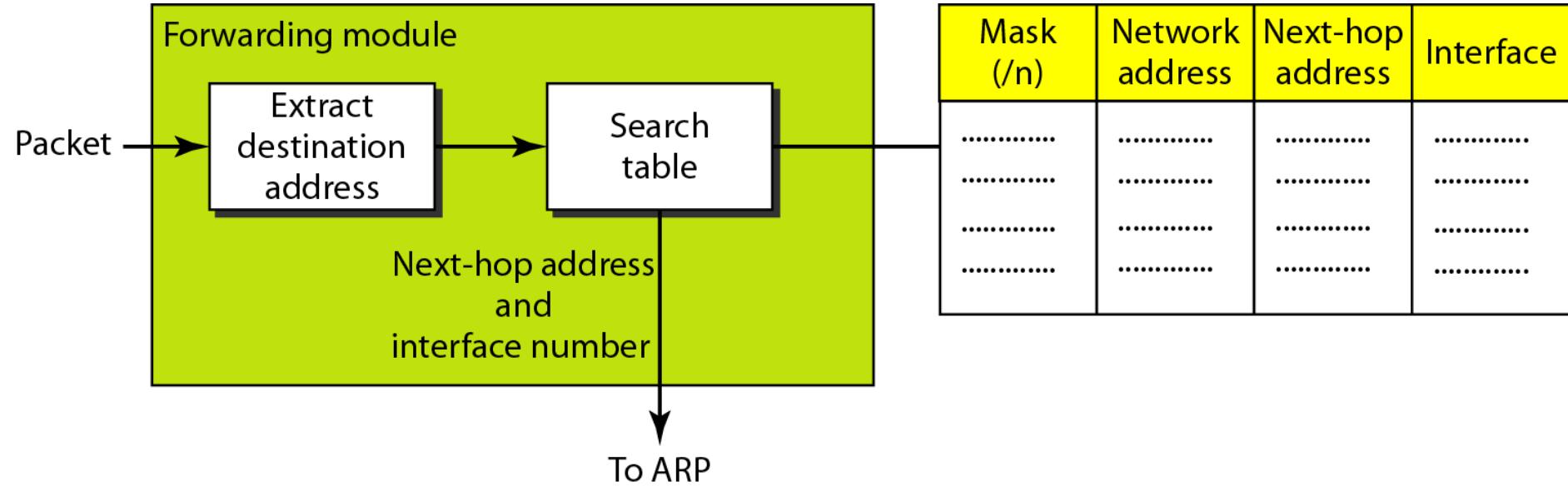


图 22.5 无类地址简化的转发模块



在无类寻址中，一个路由表至少要有 4 列。

Figure 22.6 制作 R1 的路由表

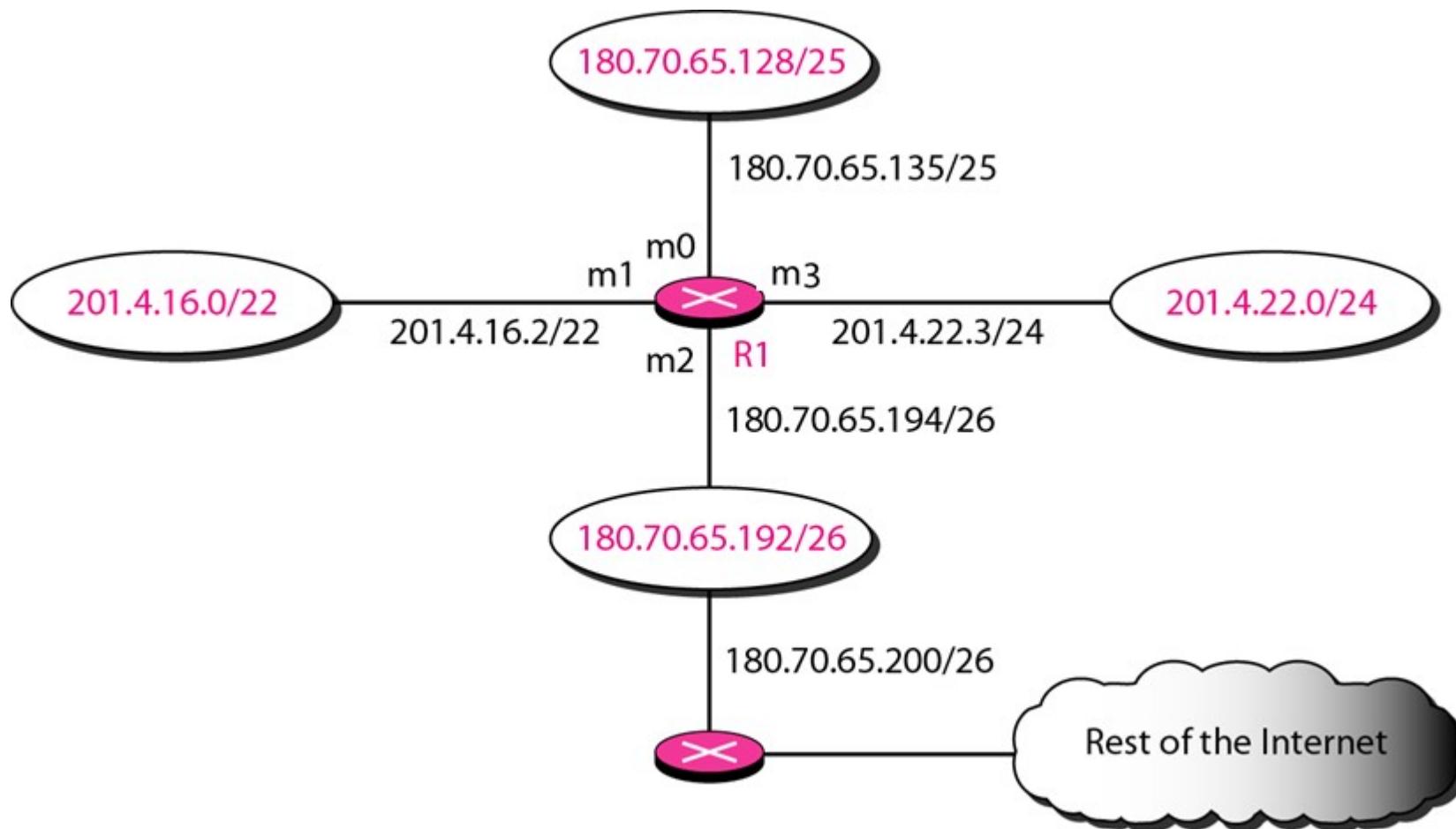
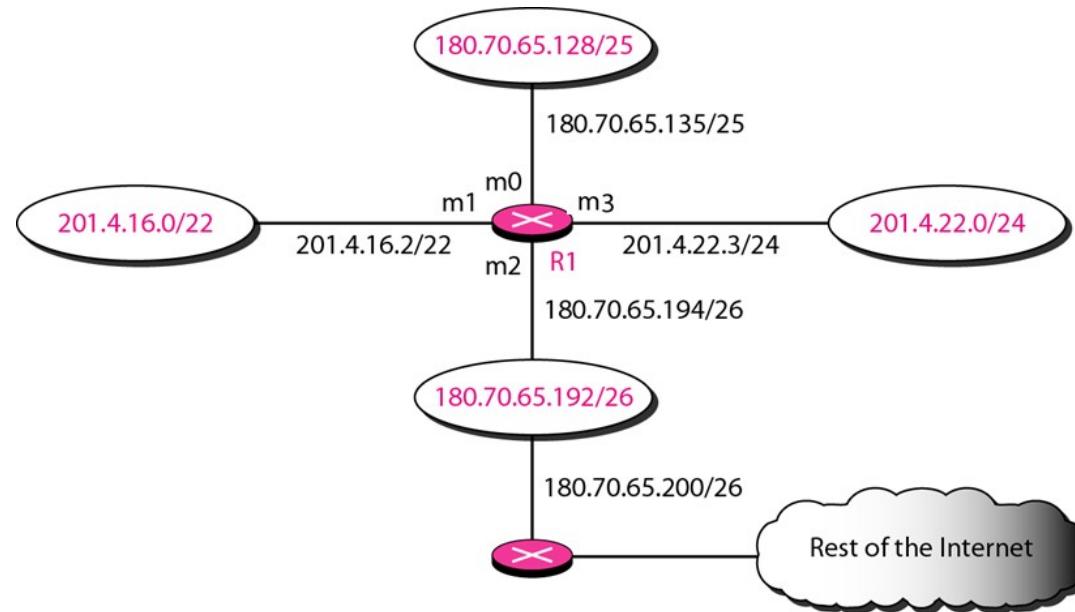
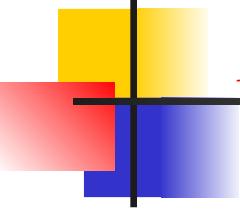


Table 22.1 图 22.6 中路由器 R1 的路由表



<i>Mask</i>	<i>Network Address</i>	<i>Next Hop</i>	<i>Interface</i>
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	m1
Any	Any	180.70.65.200	m2



Example 22.2

如果图 22.6 中的一个目的地址为 180.70.65.140 的分组到达路由器 R1，说明其转发过程。

Solution

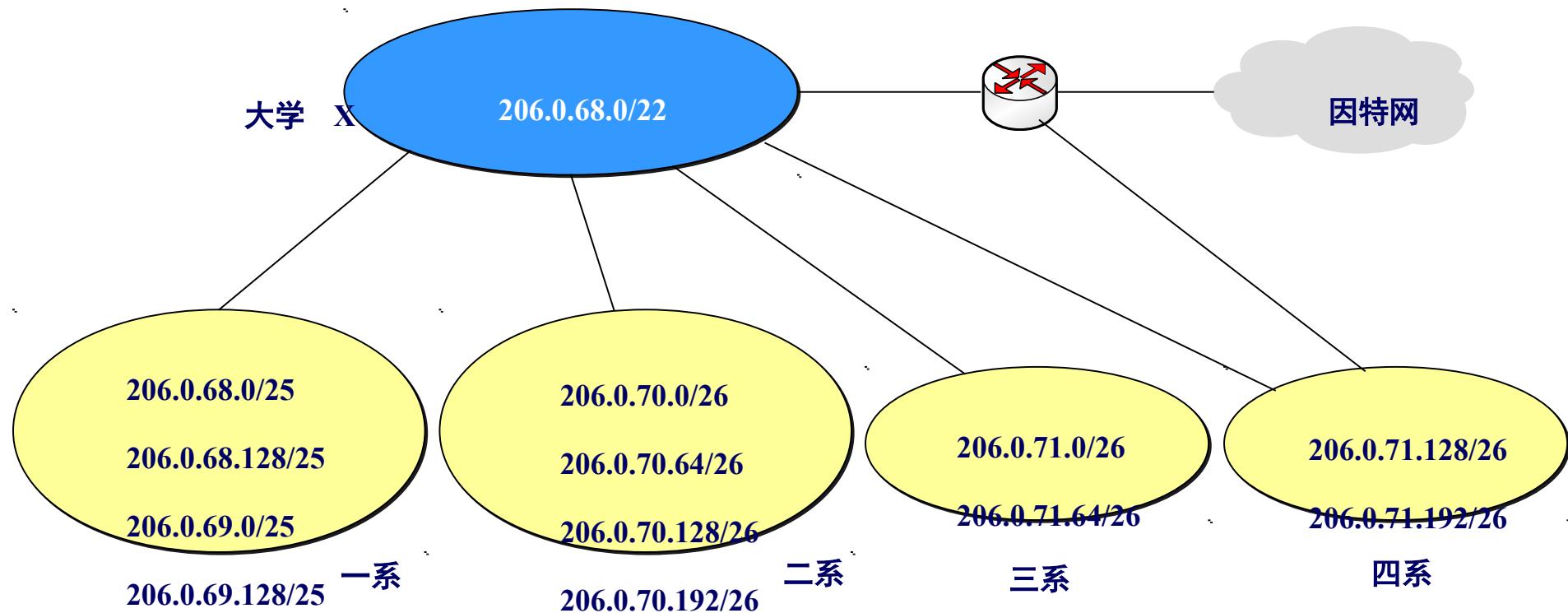
路由器执行以下步骤：

1. 第一个掩码 (/26) 作用于这个目的地址，其结果是 180.70.65.128，它与对应的网络地址不匹配；
2. 第二个掩码 (/25) 作用于这个目的地址，其结果是 180.70.65.128，它与对应的网络地址匹配，将下一跳地址和接口号传送到 ARP 做进一步处理。

最长掩码匹配

- 使用 CIDR 时，路由表中的每个项目由“掩码”、“网络地址”和“下一跳地址”组成。在查找路由表时可能会得到不止一个匹配结果。
- 应当从匹配结果中选择具有最长掩码的路由。之所以称为最长掩码匹配，是因为这个表项也是路由表中，与目的地址的高位匹配得最多的表项。
- 掩码越长，其地址块就越小，因而路由就越具体。
- 路由表中常常包含一个默认路由，这个路由在所有表项都不匹配的时候有着最短的掩码匹配。

最长前缀匹配举例



路由器收到的分组目的地址 $D = 206.0.71.130$

路由表中的项目: 206.0.68.0/22 (大学)

206.0.71.128/25 (四系)

路由表中的第 1 个项目 206.0.68.0/22 和 D 与掩码相与的结果匹配。

路由表中的第 2 个项目 206.0.71.128/25 和 D 与掩码相与的结果匹配。

图 22.8 最长掩码匹配

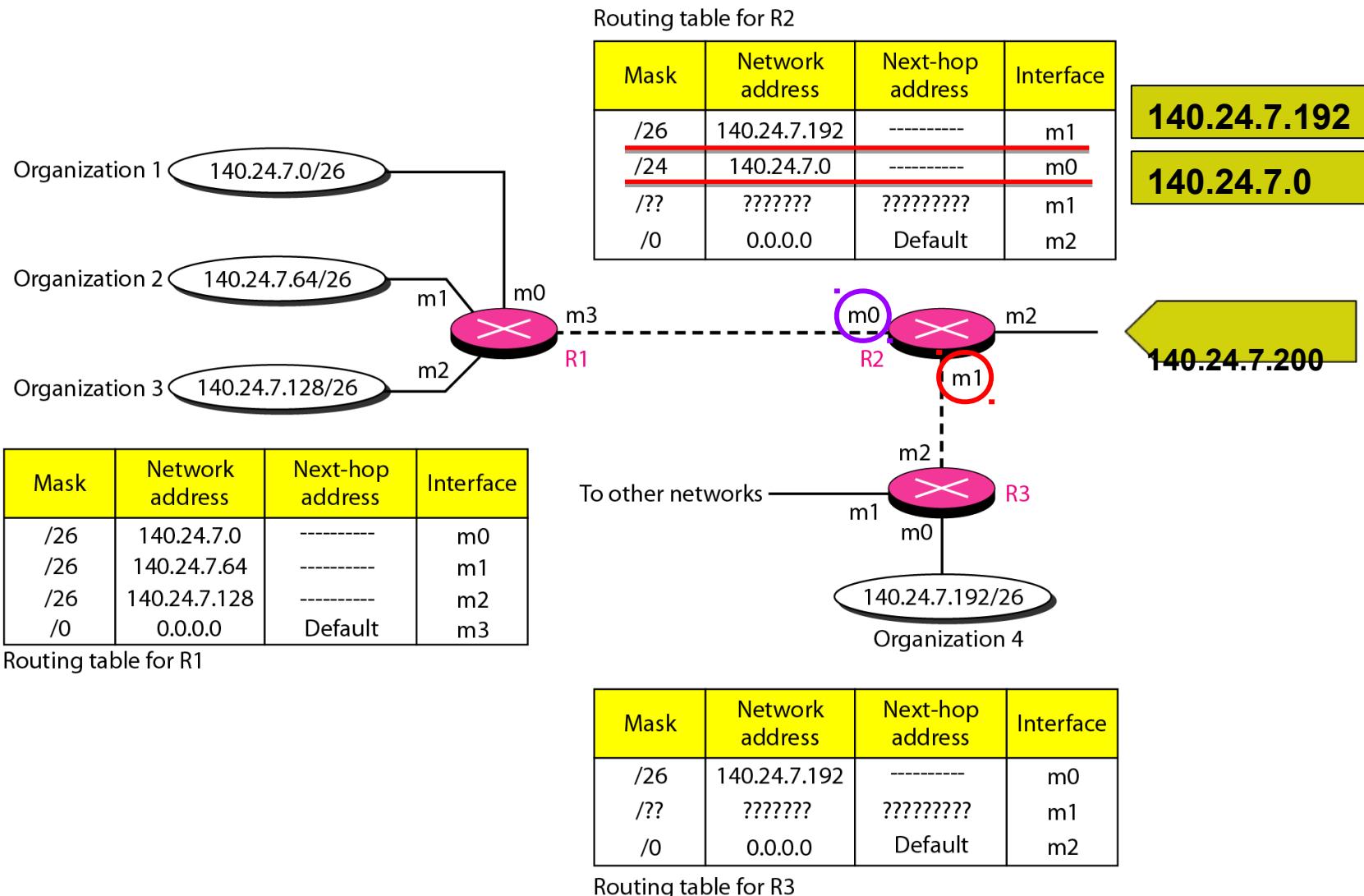
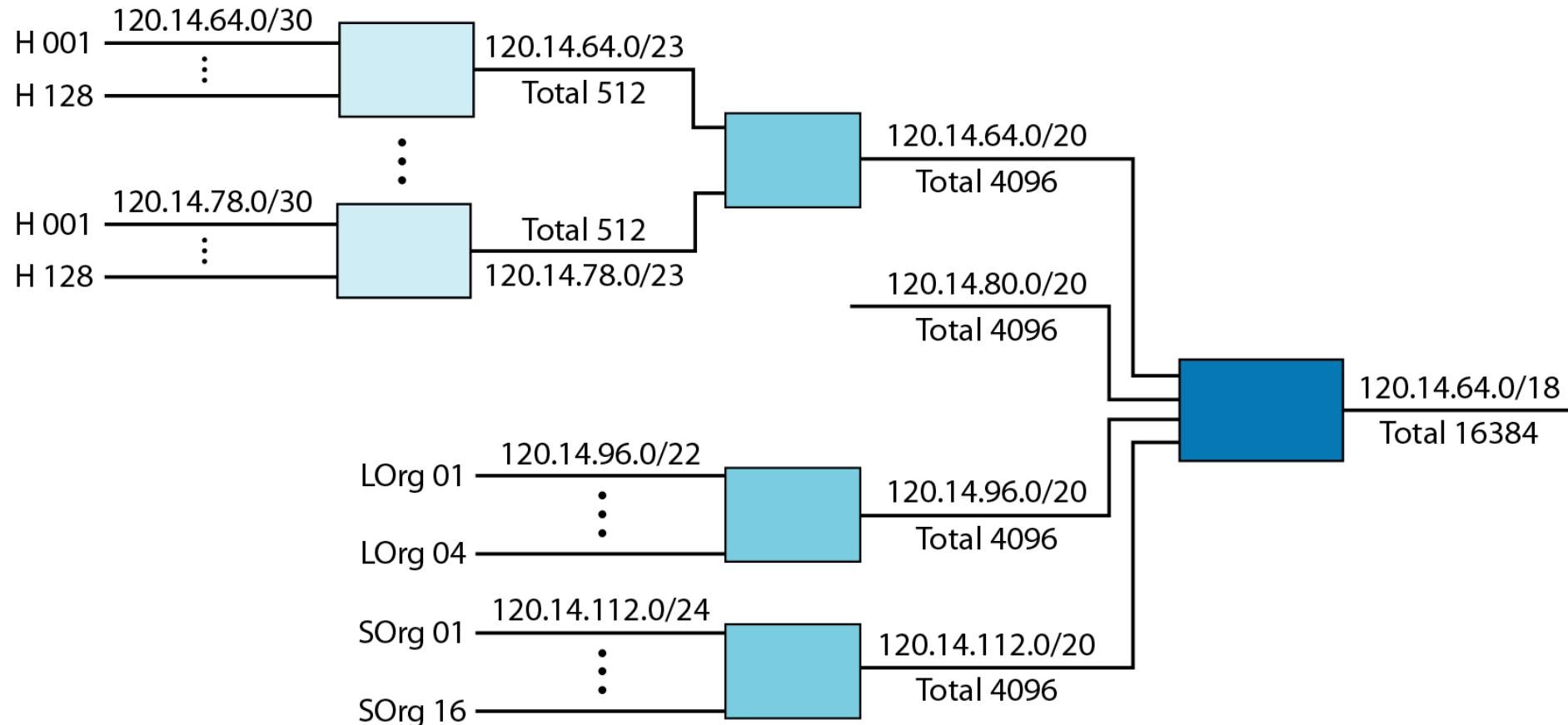


Figure 22.9 ISP 的层次结构路由选择



路由

路由：从某一网络设备出发去往某个目的地所经过的路径。路由器通过查询路由表为数据报选择转发路径。路由表只存在于终端计算机、路由器及三层交换机中，二层交换机中不存在路由表。

- ◆直连路由：设备自动发现的路由信息，路由器可自动发现与自己接口直接相连的网络的路由。
- ◆静态路由：人工输入，无法自动更新。用于小型互联网或试验网络。
- ◆动态路由：可周期性更新，适合大型网络。

Mask	Network address	Next-hop address	Interface	Flags	Reference count	Use
.....

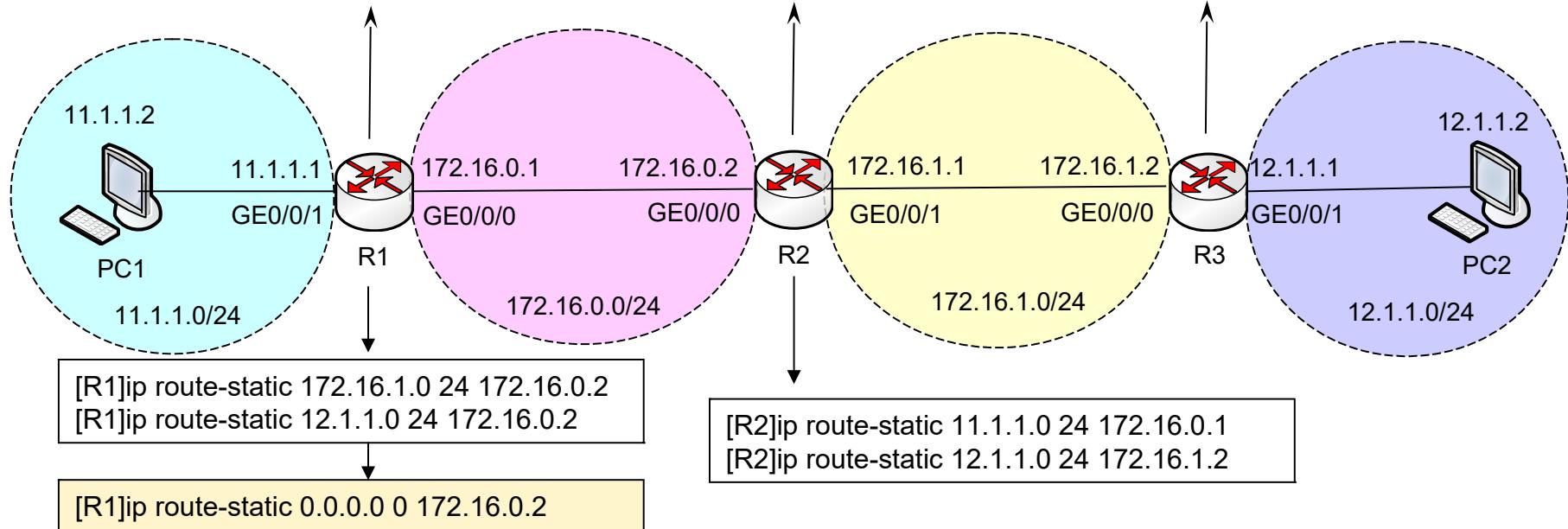
图 22.10 路由表中常用的字段

直连路由和静态路由

路由表		
Destination/Mask	Proto	NextHop
11.1.1.0/24	Direct	--
172.16.0.0/24	Direct	--
172.16.1.0/24	Static	172.16.0.2
12.1.1.0/24	Static	172.16.0.2
.....

路由表		
Destination/Mask	Proto	NextHop
172.16.0.0/24	Direct	--
172.16.1.0/24	Direct	--
11.1.1.0/24	Static	172.16.0.1
12.1.1.0/24	Static	172.16.1.2
.....

路由表		
Destination/Mask	Proto	NextHop
12.1.1.0/24	Direct	--
172.16.1.0/24	Direct	--
11.1.1.0/24	Static	172.16.1.1
172.16.0.0/24	Static	172.16.1.1
.....

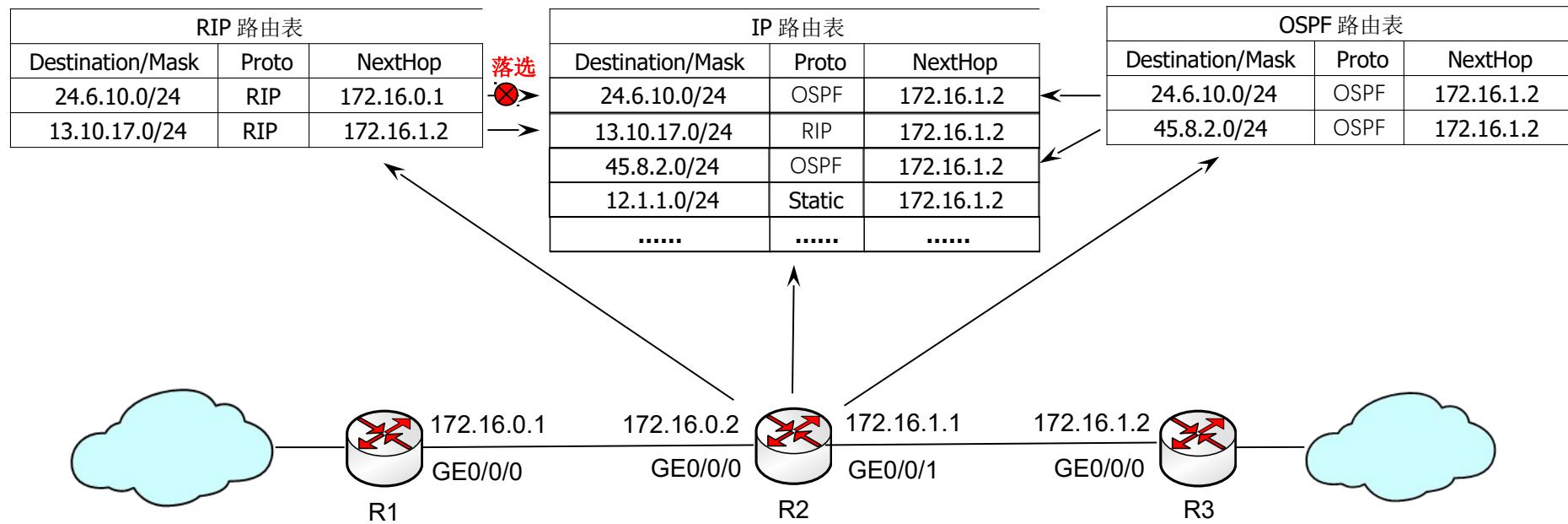


华为常用的路由协议优先级

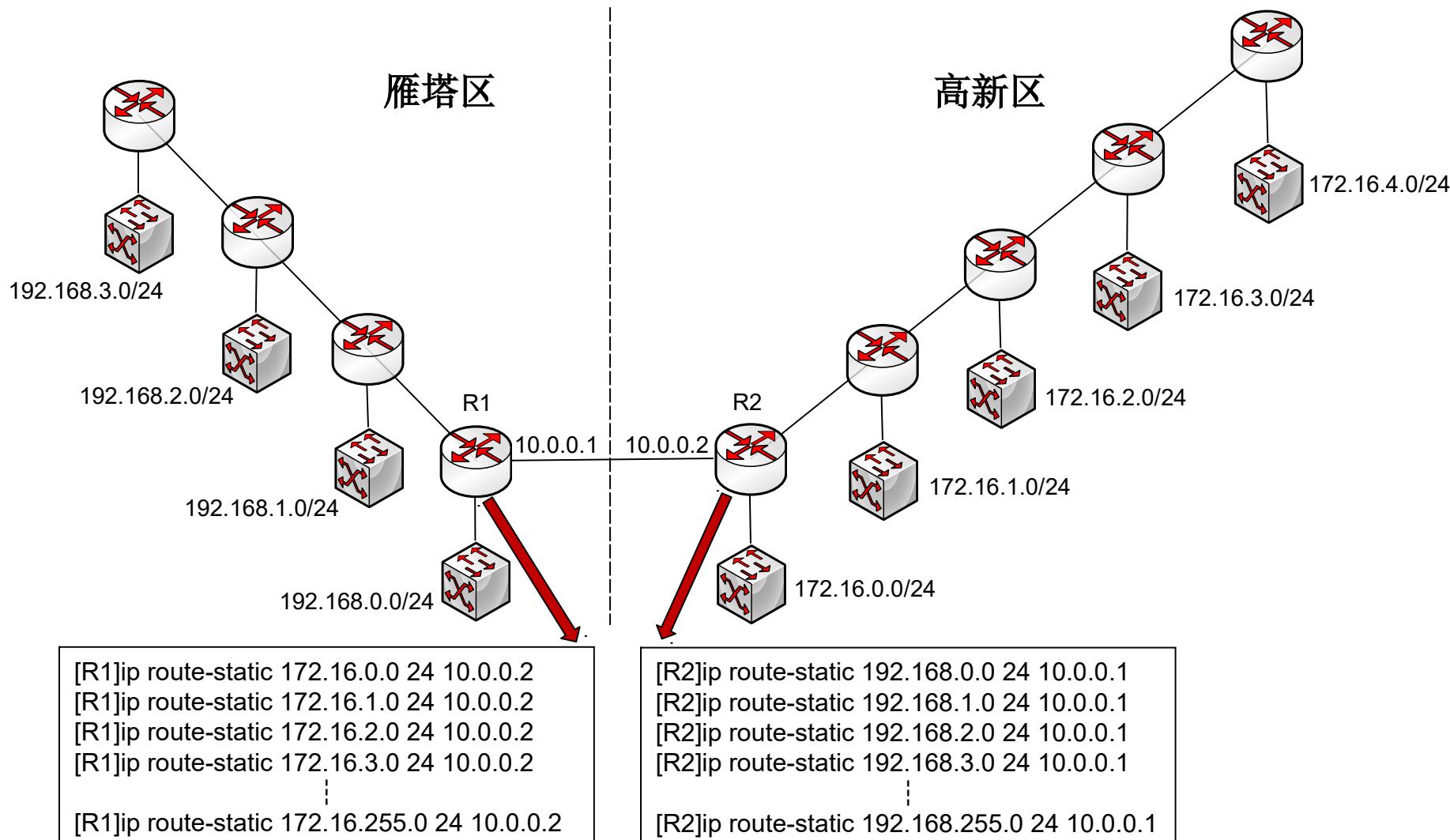
- ◆ 通常路由器会运行多种路由协议 (BGP， OSPF， ISIS 等)，同一条路由可以从不同的路由协议学习到，此时可人为定义协议优先级进行路由优选。**协议优先级越小，越被路由优选。**
- ◆ 管理路由优先级只具有本地意义，只可控制本地路由表相关路由的选择。
- ◆ OSPF-IN 表示 AS 内部传递路由； OSPF-OUT 表示引入的外部路由； iBGP 表示从 iBGP 邻居学习到路由， eBGP 表示从 eBGP 邻居学习到路由， local 表示 aggress 聚合路由。

路由协议	协议优先级
直连路由	0
OSPF-IN	10
ISIS-level1	15
ISIS-level2	18
静态 Static	60
RIP	100
OSPF-OUT	150
iBGP	255
eBGP	255

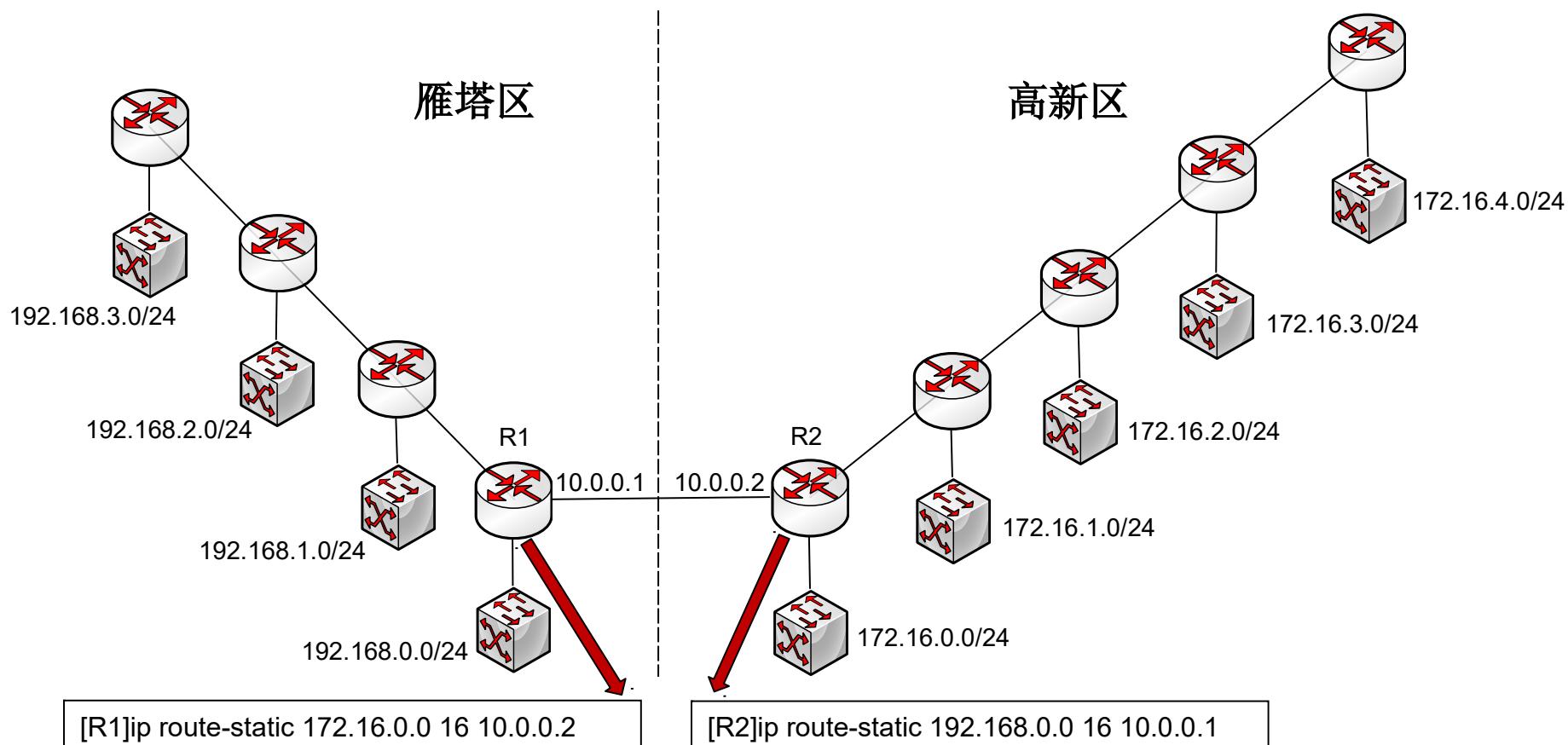
动态路由优先级



路由汇总



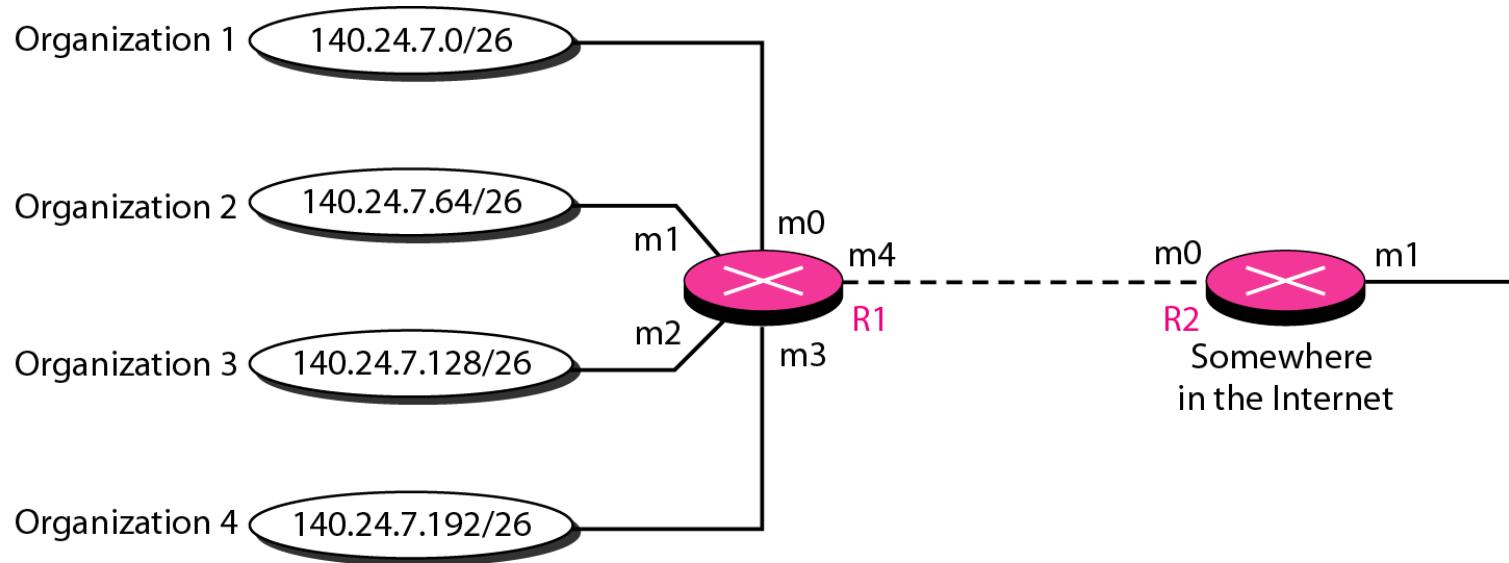
通过路由汇总简化路由表



到高新区的网络汇总成一条路由，将全部以 172.16 开头的网络进行了合并，汇总成一条路由。

到雁塔区的网络汇总成一条路由，将全部以 192.168 开头的网络进行了合并，汇总成一条路由。

Figure 22.7 地址聚合



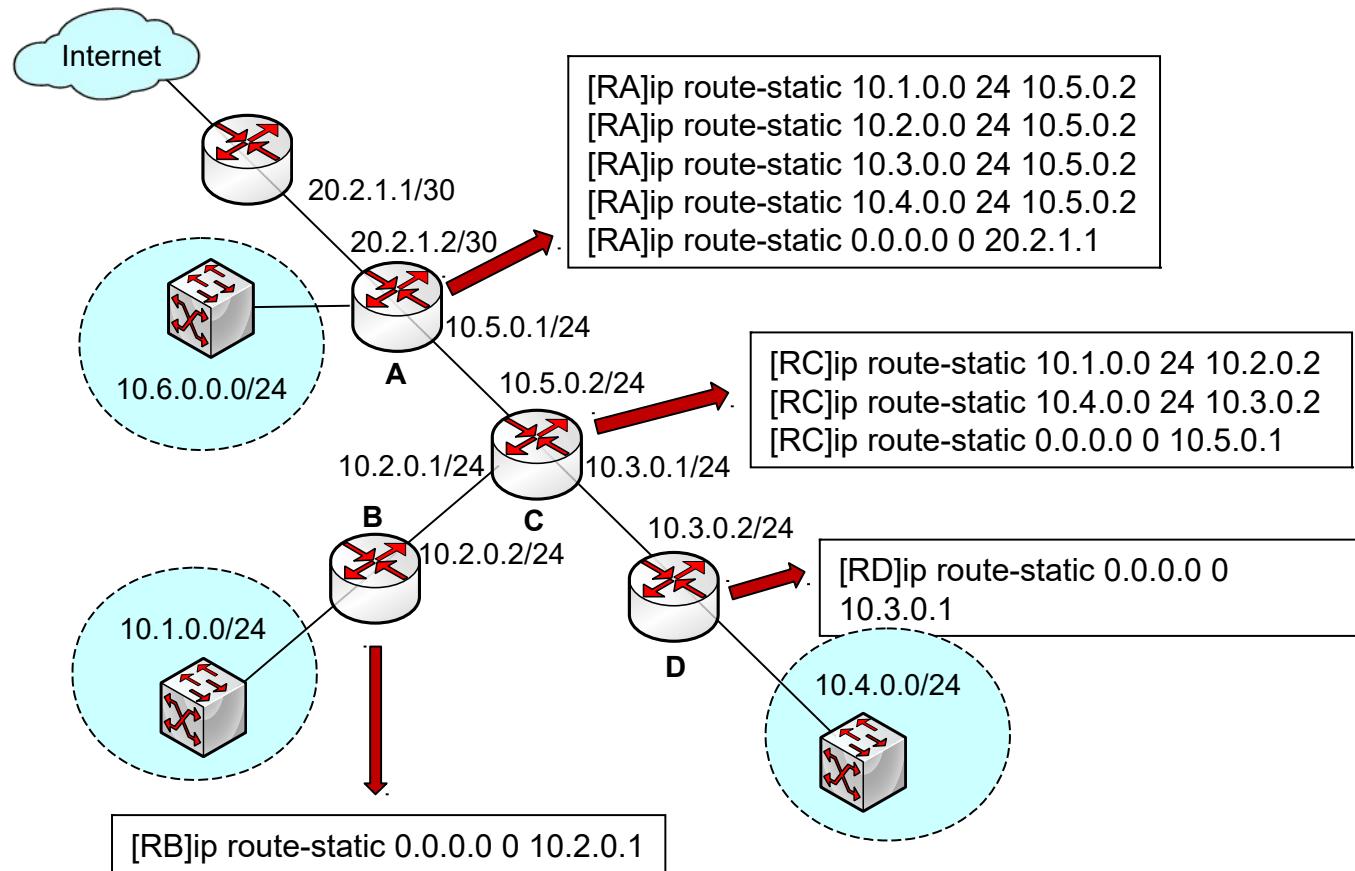
Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	-----	m0
/26	140.24.7.64	-----	m1
/26	140.24.7.128	-----	m2
/26	140.24.7.192	-----	m3
/0	0.0.0.0	Default	m4

Routing table for R1

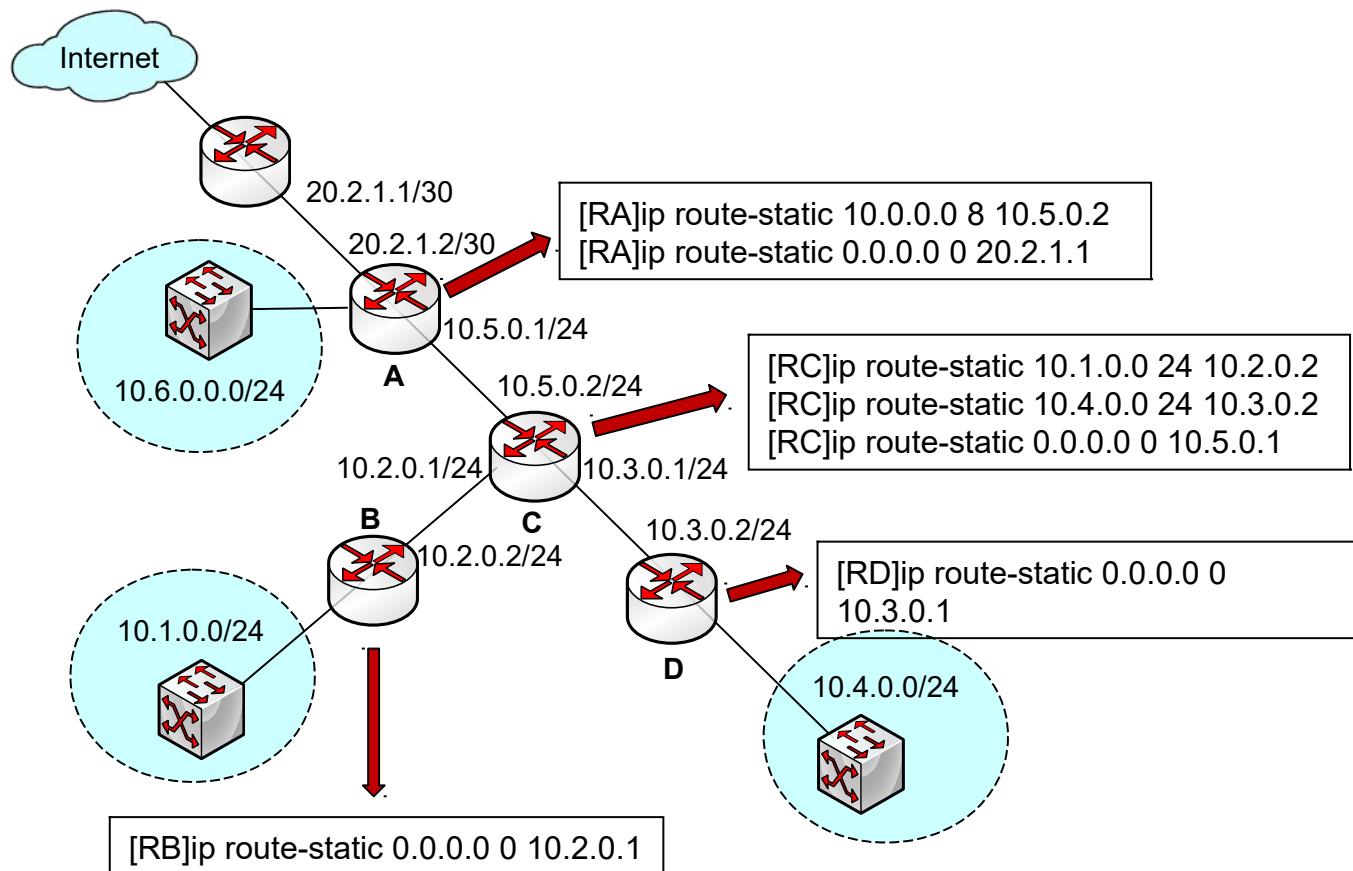
Mask	Network address	Next-hop address	Interface
/24	140.24.7.0	-----	m0
/0	0.0.0.0	Default	m1

Routing table for R2

默认路由



通过路由汇总进一步简化路由



路由算法（补充）

- 网络设计中最复杂也最关键的内容

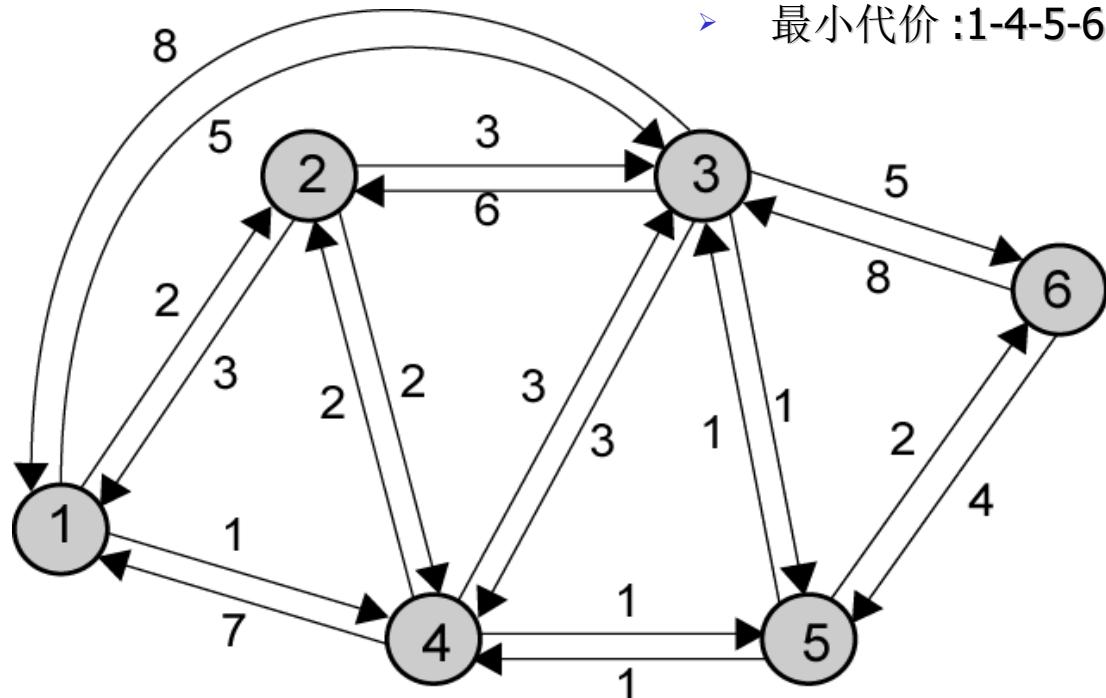
- 特性

- 正确性
- 简洁性
- 稳健性
- 稳定性
- 公平性
- 最优性
- 高效性

- 性能评估标准

- 最小跳数
- 最小代价

- Node1->node6
- 最小跳数 :1-3-6
- 最小代价 :1-4-5-6



Dijkstra 算法

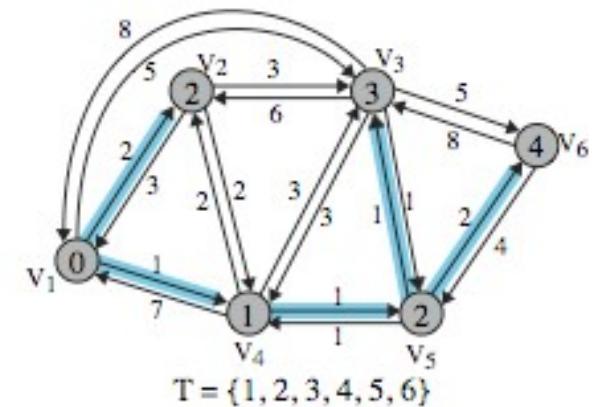
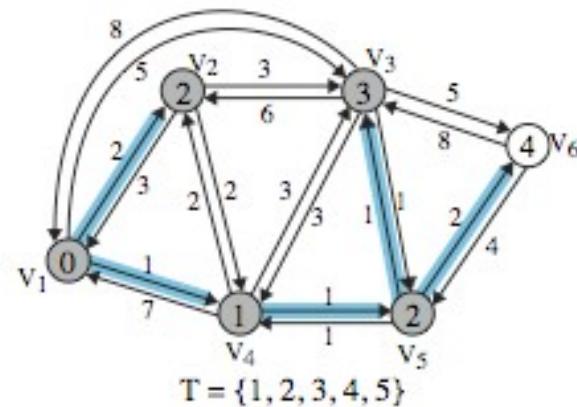
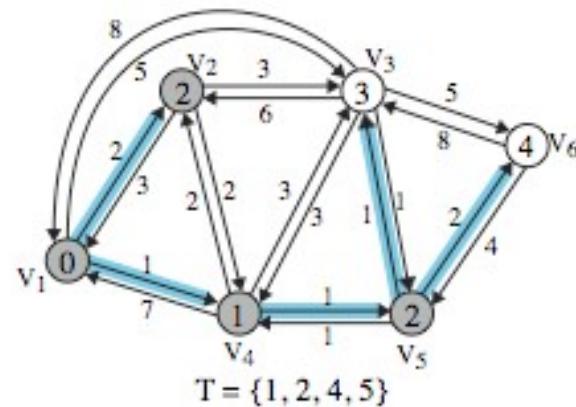
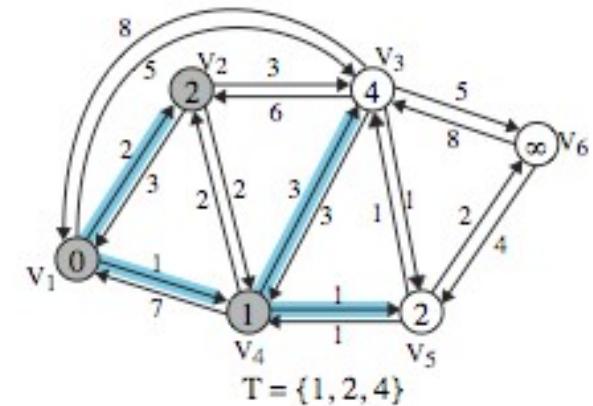
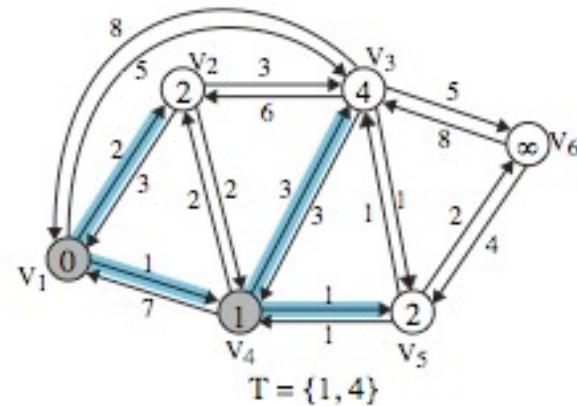
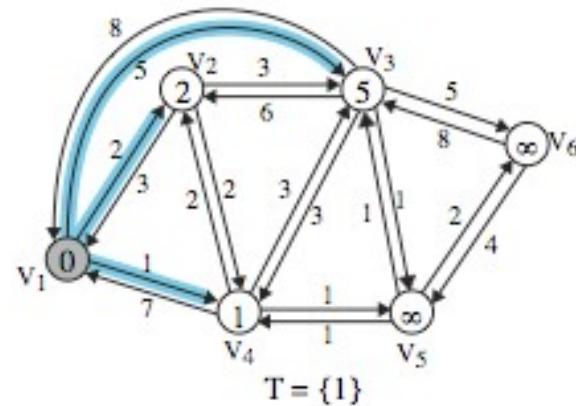
- 通过拓展路径以不断增加这条路径的长度，从而寻找源结点到所有其它结点之间的路径。
- N = 网络中结点集合
- s = 源结点
- T = 当前由算法合并的结点集合
- $w(i, j)$ = 结点 i 到结点 j 之间的链路代价
 - $w(i, i) = 0$
 - $w(i, j) = \infty$ 当结点之间不是直接连接时
 - $w(i, j) \geq 0$ 当两个结点直接连接时
- $L(n)$ = 算法目前所知从结点 s 到结点 n 之间的最小代价路径的代价
 - 算法结束时， $L(n)$ 是从 s 到目的结点 n 的最小代价路径的代价。

- Step 1 [初始化]
 - $T = \{s\}$ 目前结点集合只含有源结点
 - $L(n) = w(s, n)$ 当 $n \neq s$ ， 到相邻结点的初始路径代价就是链路代价
- Step 2 [找到下一结点]
 - 找出不在 T 中的某个相邻结点，这个相邻结点与 s 之间有最小代价路径；
 - 将这个结点合并到 T 中，同时也将该结点与 T 中某一结点形成的一条有用边加入到 T 中；
- Step 3 [更新最小代价路径]
 - $L(n) = \min[L(n), L(x) + w(x, n)]$ ，对所有 $n \notin T$
 - 如果后一个表达式的值最小，那么从 s 到 n 的路径变成了从 s 到 x 的路径以及从 x 到 n 的链路衔接。
- 当所有结点都已加入 T 后，算法结束。

Dijkstra 的例子

Iter	T	$L(2)$	Path	$L(3)$	Path	$L(4)$	Path	$L(5)$	Path	$L(6)$	Path
1	{1}	2	1-2	5	1-3	1	1-4	∞	-	∞	-
2	{1,4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	-
3	{1, 2, 4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	-
4	{1, 2, 4, 5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
5	{1, 2, 3, 4, 5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
6	{1, 2, 3, 4, 5, 6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

Dijkstra 生成树



Dijkstra 的结点

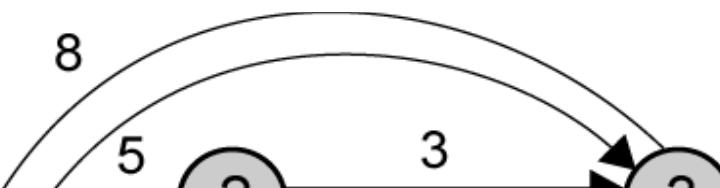
- 第 2 步和第 3 步每循环一次，就向 T 中增加一个新结点，并且定义了从 s 到该结点之间的最小代价路径。
- 算法结束时，与各结点 x 相关的 $L(x)$ 就是从 s 到 x 的最小代价路径的代价。

Bellman-Ford 算法

- 从给定的源结点找出一条最短路径，该最短路径是从所有最多只含 1 条链路的路径中选择出来的；
- 再找出条件为所有路径最多只含 2 条链路的最短路径；
- 依此类推 ...
- $s =$ 源点
- $w(i, j) =$ 结点 i 到结点 j 之间的链路代价
 - $w(i, i) = 0$
 - $w(i, j) = \infty$ 当两个结点不是直接连接时
 - $w(i, j) \geq 0$ 当两个结点直接连接时
- $h =$ 在算法目前阶段中的路径具有的最大链路数
- $L_h(n) =$ 在不多于 h 条链路的条件下，从结点 s 到结点 n 的最小代价路径的代价

- Step 1 [初始化]
 - $L_0(n) = \infty$, 对所有 $n \neq s$
 - $L_h(s) = 0$, 对所有 h
- Step 2 [更新]
 - 对每个后继的 $h \geq 0$
 - 对每个 $n \neq s$, 计算
$$L_{h+1}(n) = \min\{L_h(n), \min_j [L_h(j) + w(j,n)]\}$$
 - 将 n 与前一次处理的结点 j 连接, 以获取最小值;
 - 删除在以前循环时形成的 n 与任何其它前次处理结点之间的连接;
 - 从 s 到 n 的路径以从 j 到 n 的链路结束。

Bellman-Ford 的例子

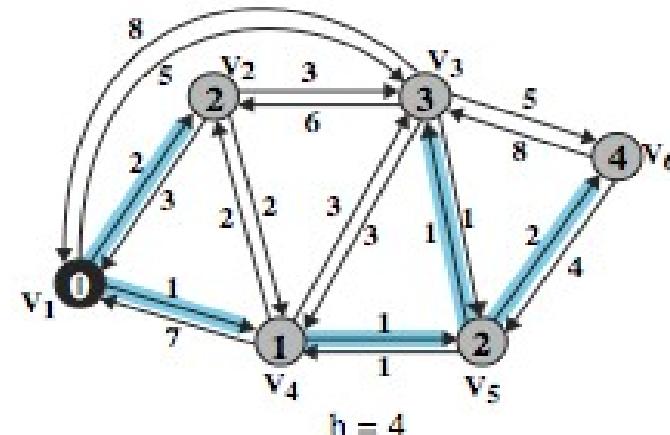
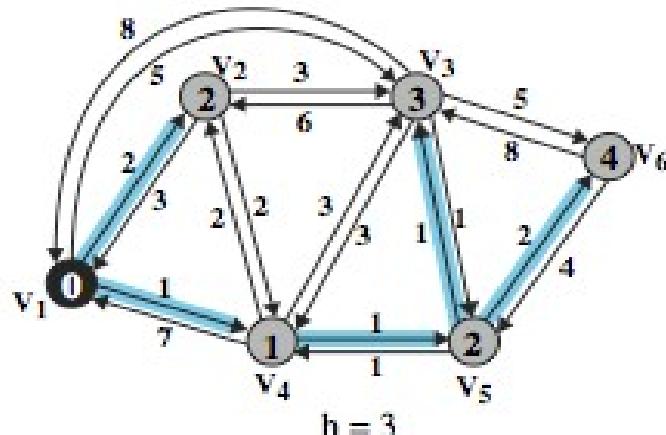
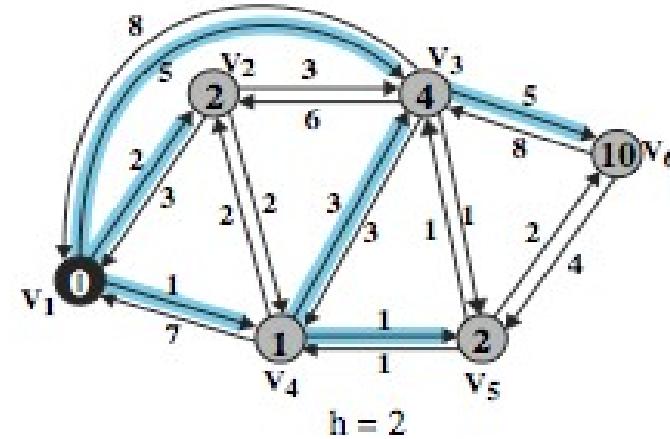
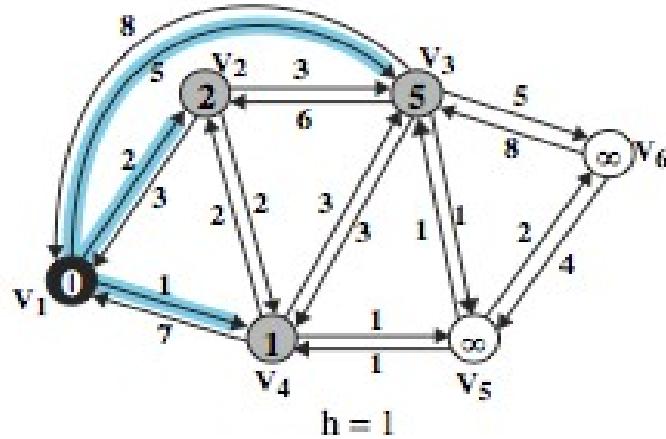


h	$L_h(2)$	Path	$L_h(3)$	Path	$L_h(4)$	Path	$L_h(5)$	Path	$L_h(6)$	Path
0	∞	-	∞	-	∞	-	∞	-	∞	-
1	2	1-2	5	1-3	1	1-4	∞	-	∞	-
2	2	1-2	4	1-4-3	1	1-4	2	1-4-5	10	1-3-6
3	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
4	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

Bellman-Ford 的结点

- 第 2 步不断重复，当 $h = K$ 时，对于每个目的结点 n ，算法将从 s 到 n 的长度为 $K+1$ 的可能路径与前一次循环结束时得到的路径相比较；
- 如果前次更短的路径具有较小的代价，仍然保持前次的路径；否则从 s 到 n 之间定义一条长度为 $K+1$ 的新路径；
- 这条新路径含有长度为 K 的从 s 到某个结点 j 的路径，再加上从结点 j 到结点 n 的直接 1 跳；
- 其中用到的从 s 到 j 的路径就是在前一次循环时为 j 定义的 K 跳路径。

Bellman-Ford 生成树



路由表可以是静态的也可以是动态的。静态路由表是由人工输入项目，而动态路由表在互联网中某处有变化时就会自动地更新。路由选择协议是一些规则和过程的组合，使得在互联网中的各路由器能够彼此互相通知这些变化。

本节主要讨论：

优化原则

域内部和域间路由选择

距离向量路由选择和 **RIP**

链路状态路由选择和 **OSPF**

路径向量路由选择和 **BGP**

动态路由协议的功能

- 知道有哪些邻居路由器；
- 能够学习到网络中有哪些网段；
- 能够学习到至某个网段的所有路径；
- 能够从众多的路径中选择最佳的路径；
- 能够维护和更新路由信息。

优化原则

- 路由器将分组转发到哪个与其相连的网络，取决于哪一个可用路径是最佳路径。
- 度量：给网络指定代价
 - 路由信息选择协议 RIP
 - 开放最短路径优先协议 OSPF
- 可达性
 - 边界网关协议

图 22.12 自治系统

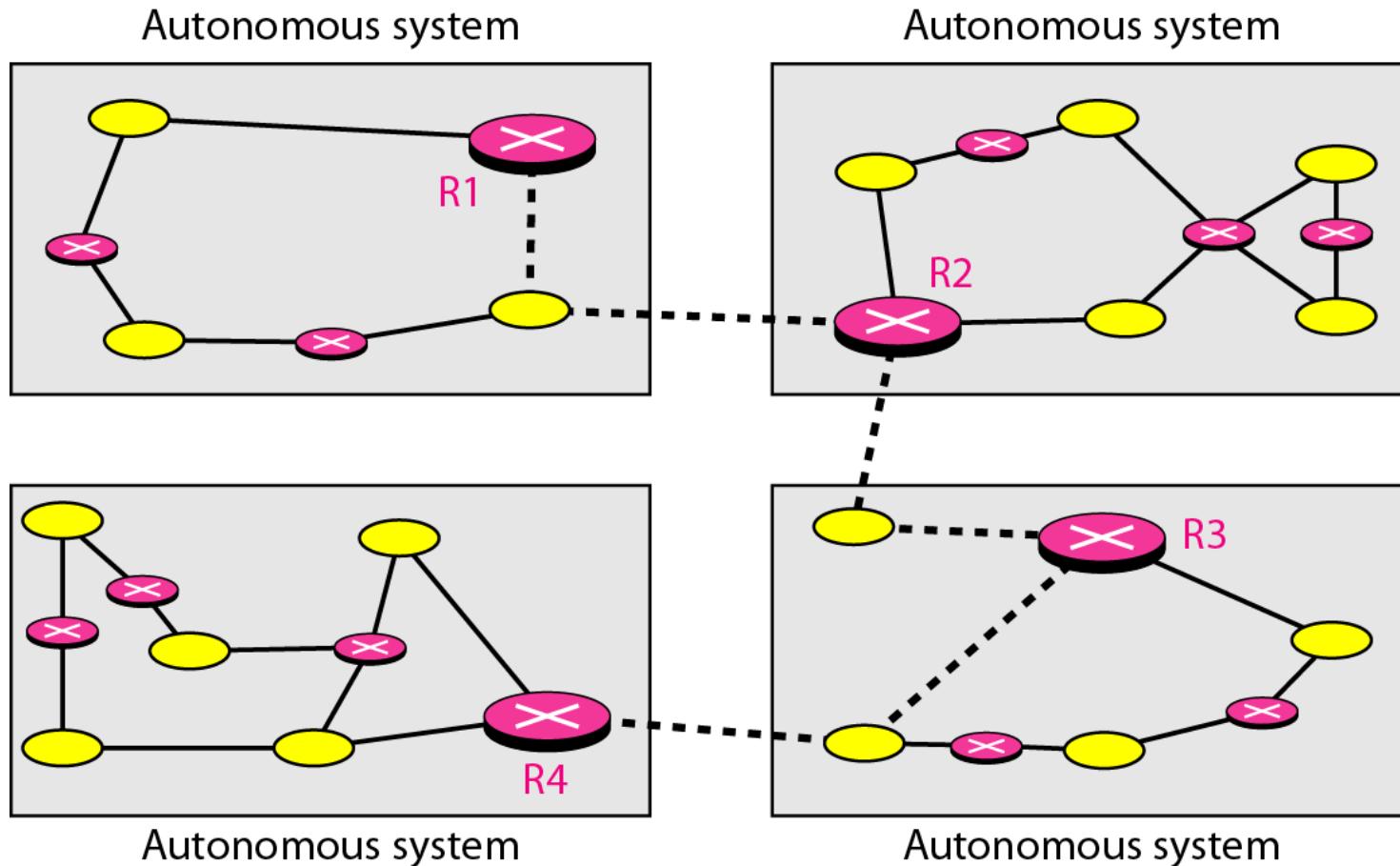


图 22.13 流行的路由选择协议

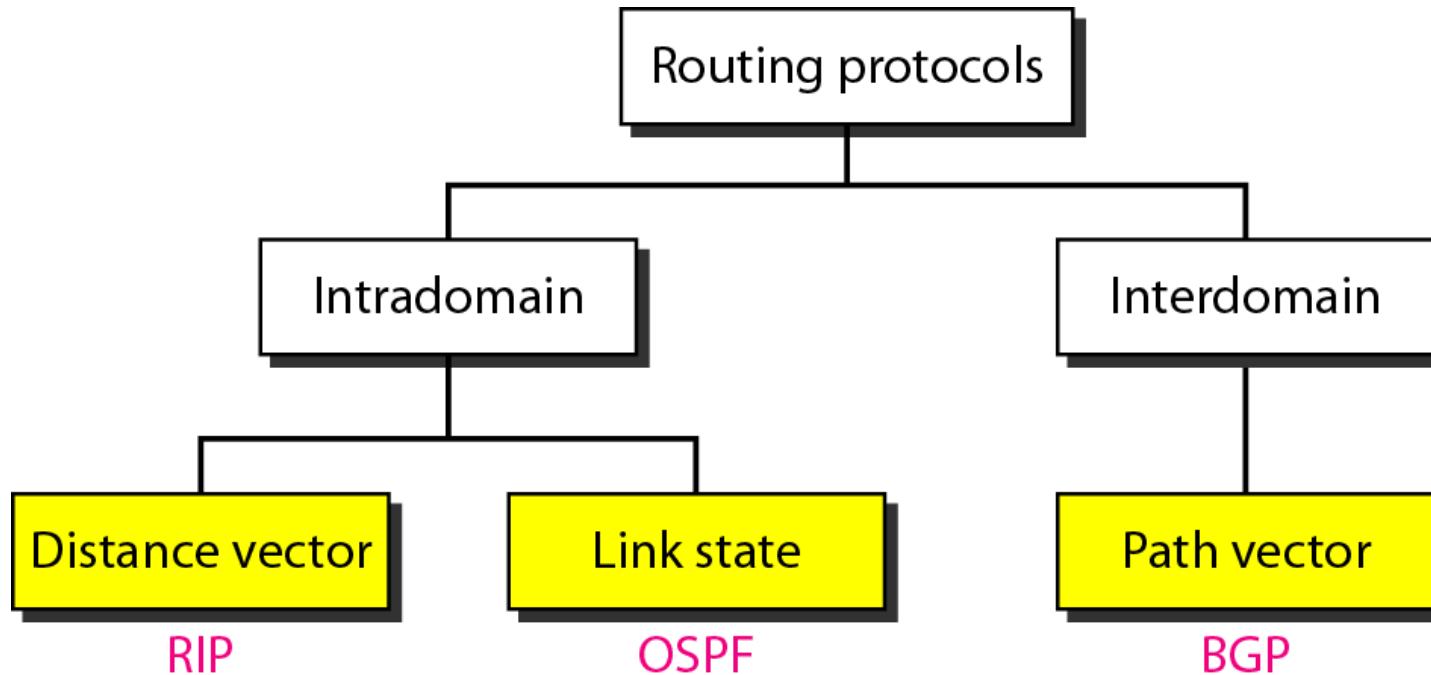
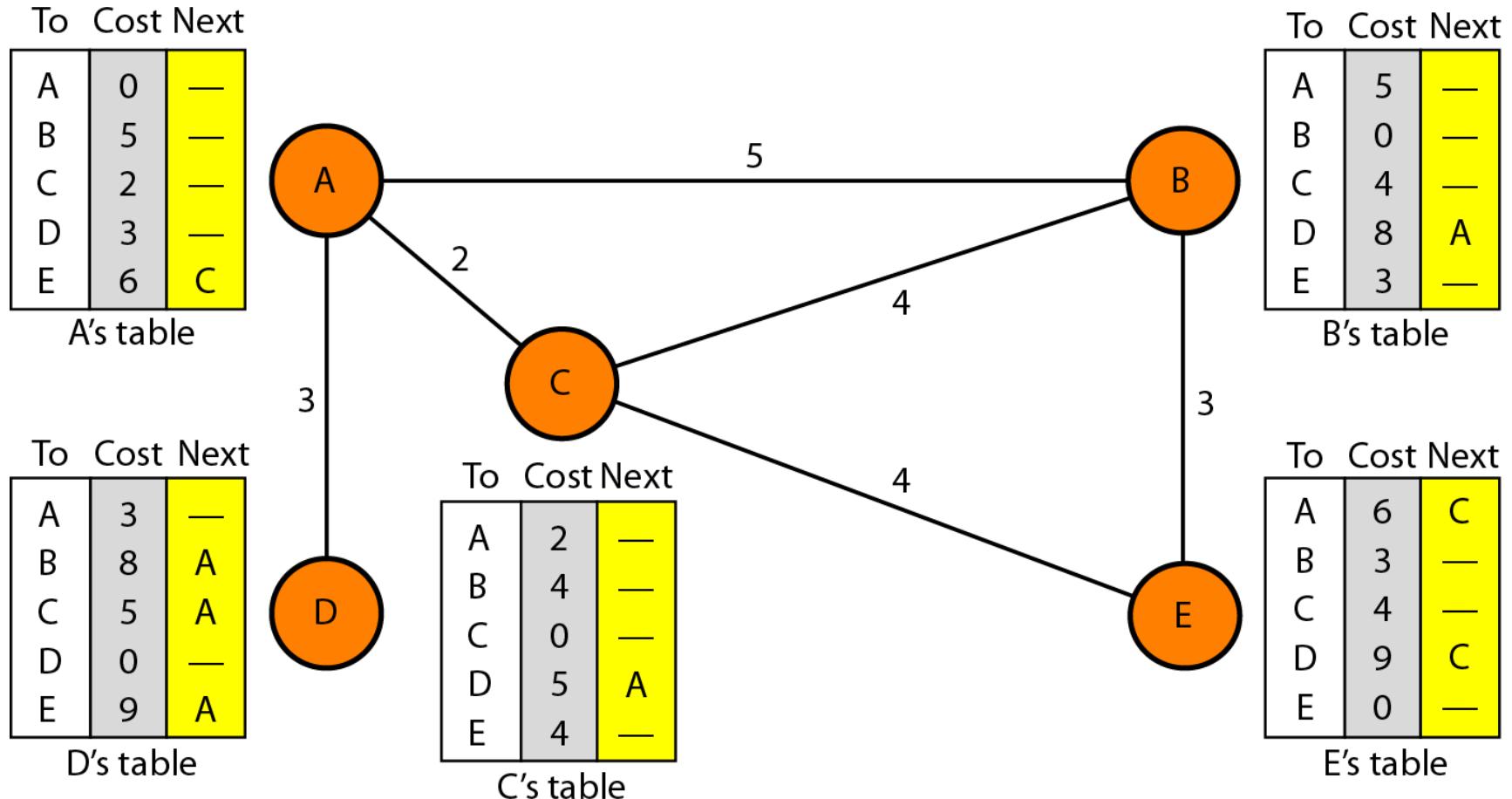
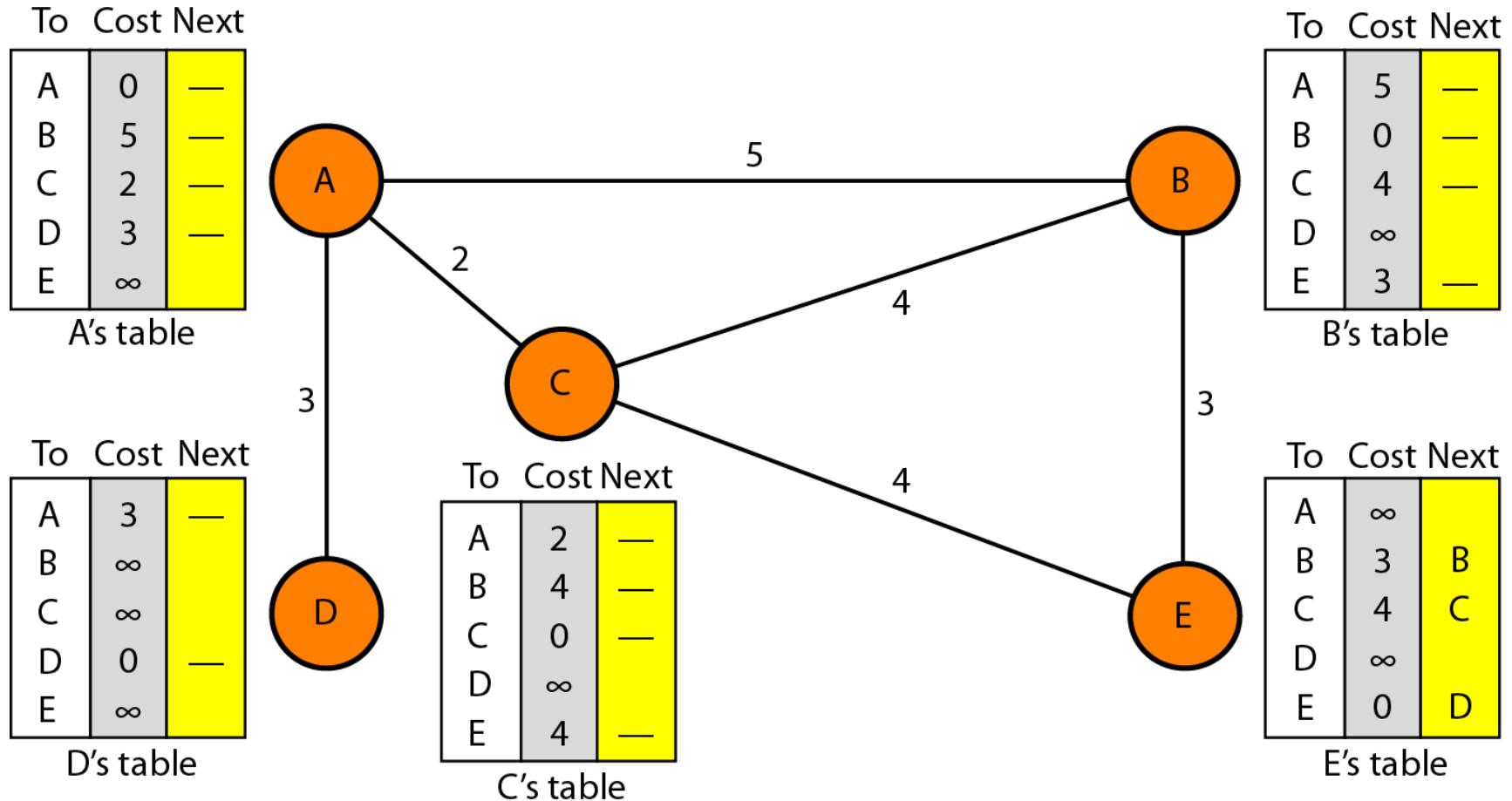


图 22.14 距离向量路由选择表



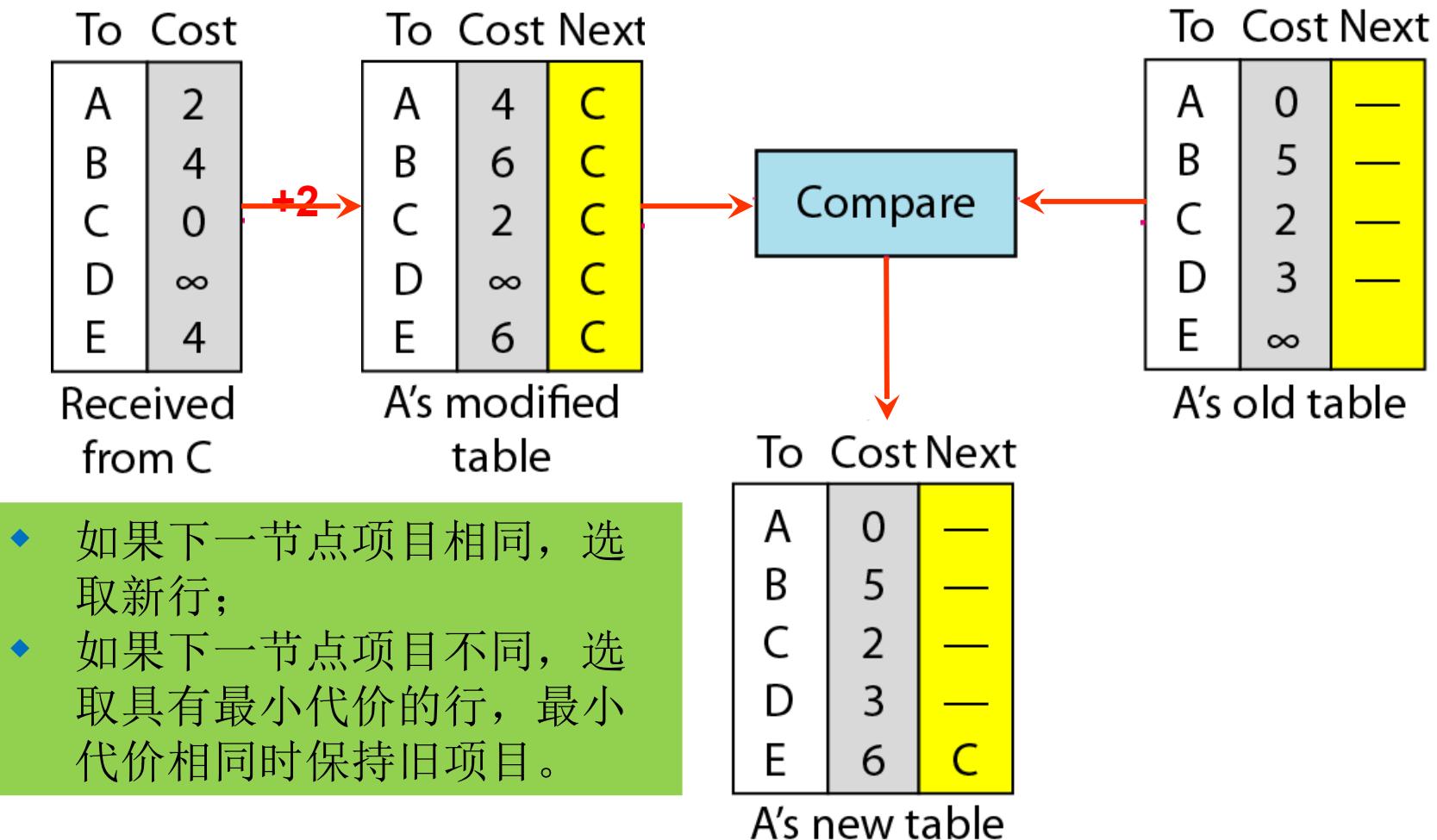
Step1: 初始化



Step2: 共享路由信息

- 在距离向量路由选择中，每个节点与它的邻站周期性地或有变化时共享其路由表。
- 每个节点向邻居节点共享它的完整路由表。
- 但是，表中的第三列对接收节点来说是没有用的，当它收到一个表的时候，把第三列都用发送方的节点名替代。

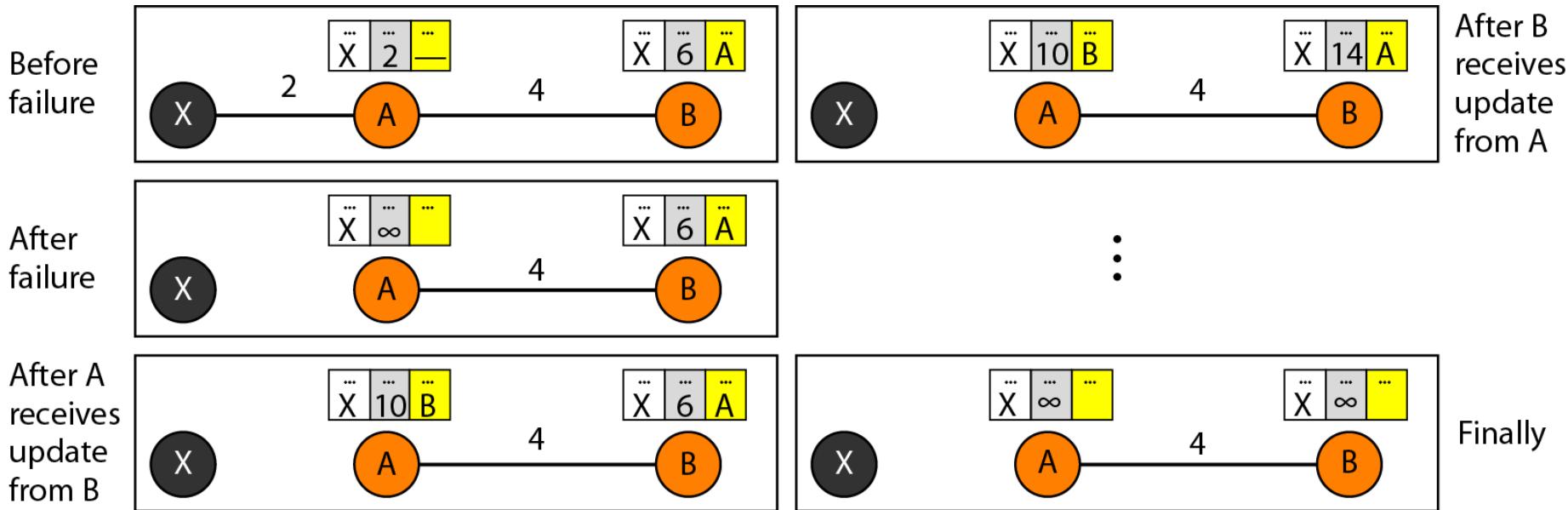
Step3: 距离向量更新



何时共享

- 周期性更新：通常每隔 30 秒。
- 触发更新：路由表有变化时。
 - 节点接收到邻站的表，引起自己表的更新；
 - 节点检测到邻站链路有故障。

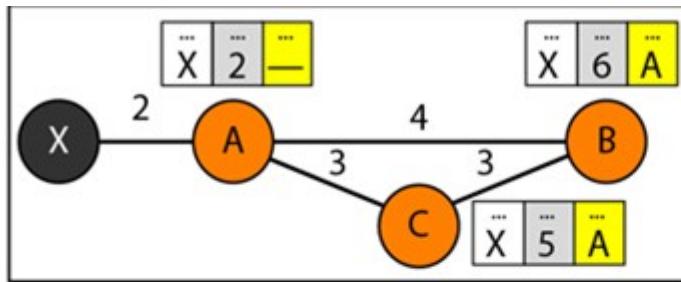
图 22.17 两个节点不稳定性



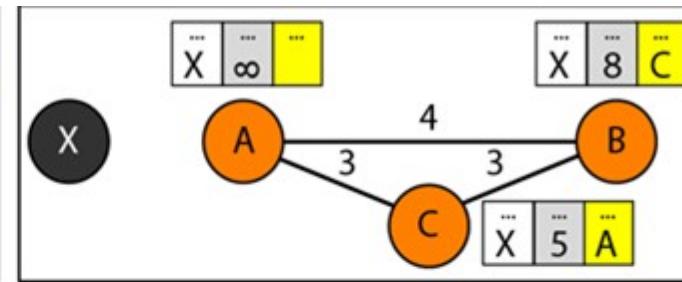
- ◆ 定义无穷大：将一个较小的数值定义为无穷大，RIP 协议为 16；
- ◆ 分割范围（split horizon）：发送表的一部分。如果节点 B 认为通过 A 到达 X 是最佳路径，则 B 不向 A 通知此消息；
- ◆ 毒性逆转（poison reverse）：如果节点 B 到达 X 的最佳路径是通过 A，它将告诉 A 自己到 X 的距离是无穷大。这样，B 向 A 撒了一个善意的谎言，使得只要 B 经过 A 选路到 X，它就会一直持续这个谎言，A 也就永远不会尝试从 B 选路到 X 了，因而避免了环路问题。

图 22.18 三个节点不稳定性

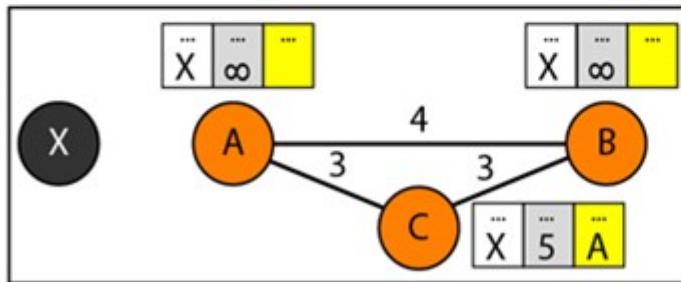
Before failure



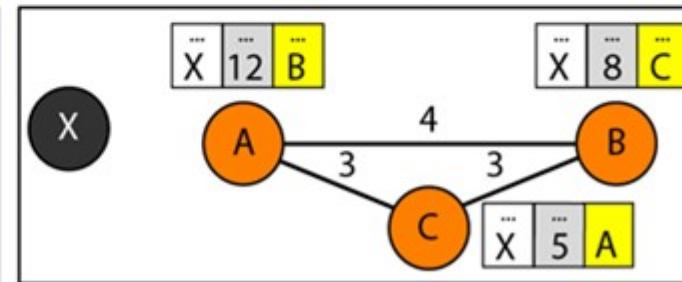
After C sends
the route to B



After A sends
the route to B
and C, but the
packet to C
is lost



After B sends
the route to A



路由信息协议 RIP (Routing Information Protocol)

- RIP 是基于距离矢量算法的路由协议，利用跳数来作为计量标准，每经过一个路由器，跳数就加 1。
- RIP 采用 Bellman-Ford 算法，是基于 UDP 的协议。
- RIP 协议分为 RIPv1 和 RIPv2，两者的区别如下：
 - RIPv1 仅适用于有类网络路由，其协议报文无法携带掩码信息，只能识别 A、B、C 类自然网段的路由，因此 RIPv1 不支持非连续子网。当网络中划分子网时，RIPv1 仅通告其主类网络地址，形成路由表。RIPv1 采用广播更新（255.255.255.255），十分占用网络资源。
 - RIPv2 报文中携带掩码信息，支持可变长子网掩码 VLSM 和无分类域间路由选择 CIDR，支持对协议报文进行验证，并提供明文验证和 MD5 验证两种方式，增强安全性。RIPv2 采用增量更新、组播更新（224.0.0.9），减少资源消耗。

RIP 协议的优缺点

缺点：

- 由于 15 跳为最大值， RIP 只能应用于小规模网络；
- 收敛速度慢（ 240s ）；
- 根据跳数选择的路由， 不一定是最优路由；
- 带宽占用率大（ RIPv1 广播更新， RIPv2 组播更新， 但都 30s 一次）；
- 网络可见度只有一跳。

优点：实现简单，开销较小。

RIP 协议算法

收到相邻路由器（其地址为 X）的一个 RIP 报文：

(1) 先修改此 RIP 报文中的所有项目：把“下一跳”字段中的地址都改为 X，并把所有的“距离”字段的值加 1。

(2) 对修改后的 RIP 报文中的每一个项目，重复以下步骤：

若原来路由表中没有目的网络 N，则把该项目加到路由表中；

否则（即路由表中有目的网络 N），此时查看下一跳地址。

若下一跳地址是 X，则把收到的项目替换原路由表中的项目；

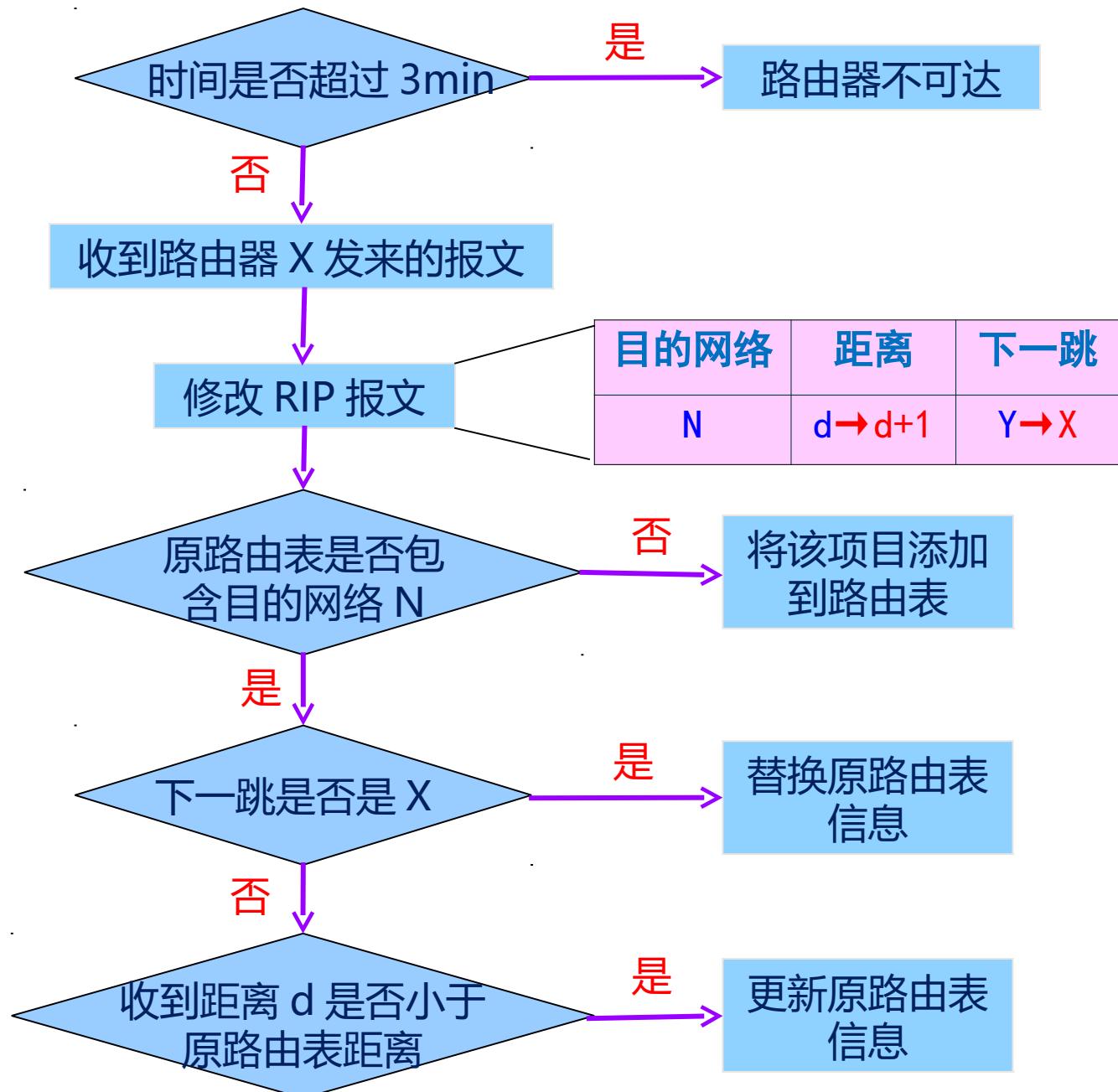
否则（即路由表中有目的网络 N，但下一跳不是 X）

若收到项目中的距离小于路由表中的距离，则进行更新；

否则什么也不做。

(3) 若 3 分钟还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达路由器，即将距离置为 16。

(4) 返回。



【例】已知路由器 R6 有表 (a) 所示的路由表。现在收到相邻路由器 R4 发来的路由更新信息，如表 (b) 所示。试更新路由器 R6 的路由表。

目的网络	距离	下一跳
net2	3	R4
net3	4	R5
...

(a) 路由器 R6 的路由表

目的网络	距离	下一跳
net1	3	R1
net2	4	R2
net3	1	直接交付

(b) 路由器 R4 发来的路由更新信息

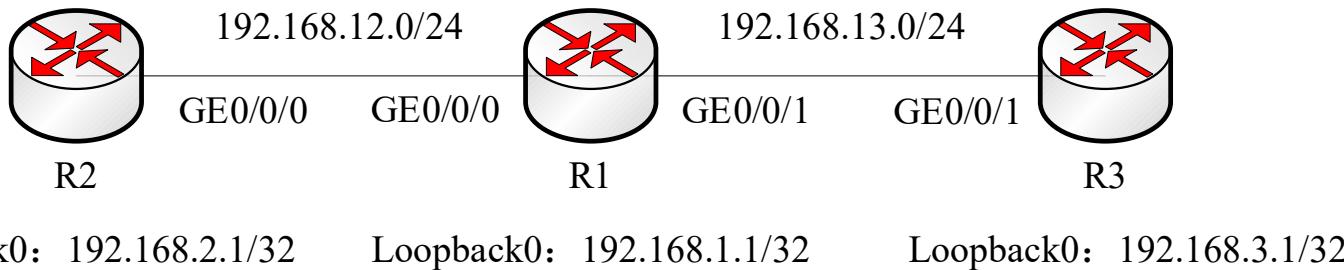
目的网络	距离	下一跳
net1	4	R4
net2	5	R4
net3	2	R4

(c) 将表 (b) 修改后得到的路由表

目的网络	距离	下一跳
net1	4	R4
net2	5	R4
net3	2	R4
...

(d) 路由器 R6 更新后的路由表

RIP 协议实现



- 配置环回接口地址

```
[R1]interface LoopBack 0
```

```
[R1-LoopBack0]ip address 192.168.1.1 32
```

- 在路由器上启用 RIPv1 并宣告直连网络

```
[R1]rip
```

```
[R1-rip-1]network 192.168.1.0 // 环回接口
```

```
[R1-rip-1]network 192.168.12.0 // 直连网段
```

```
[R1-rip-1]network 192.168.13.0 // 直连网段
```

- 查看路由表

```
[R1]display ip routing-table
```

链路状态路由选择

区域中的每个节点拥有该区域的全部拓扑结构，包括所有节点和链路的列表，它们如何连接包含类型、代价即度量和链路的接通或断开的情况。

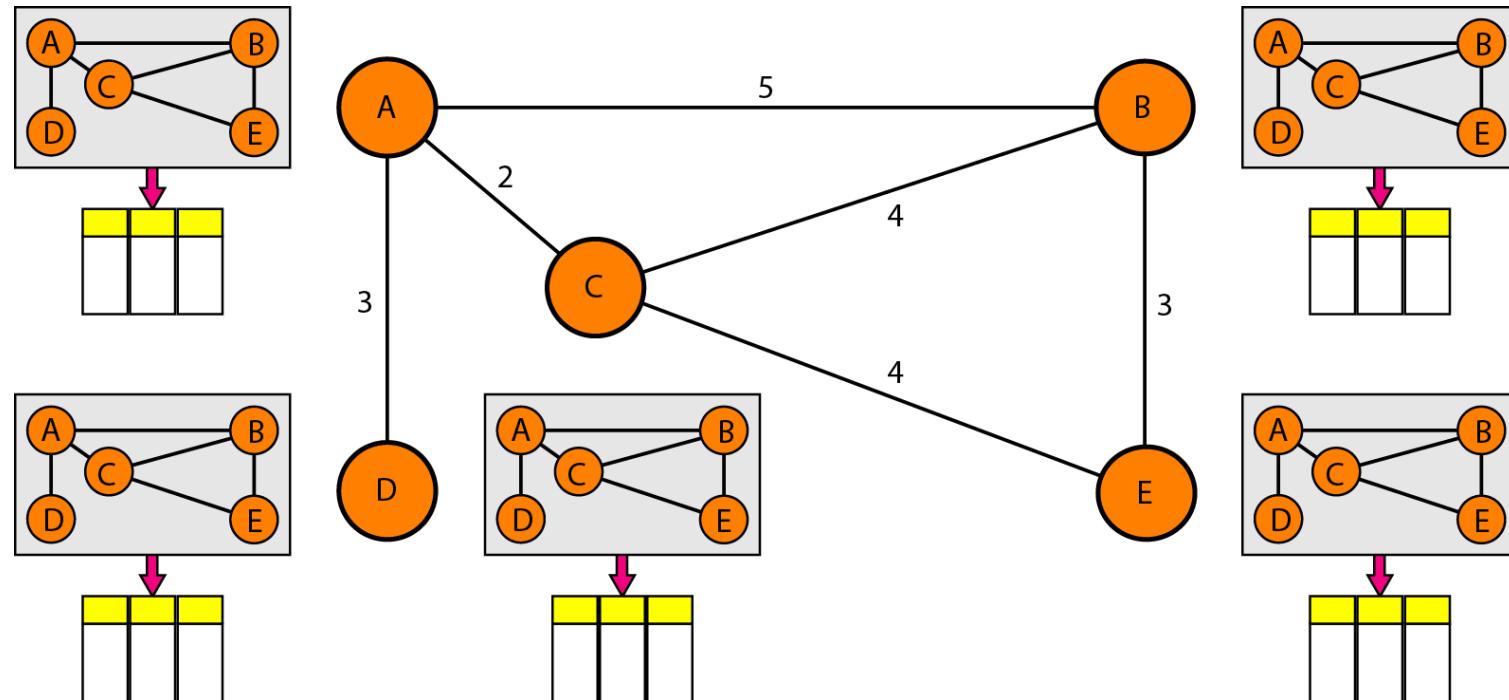
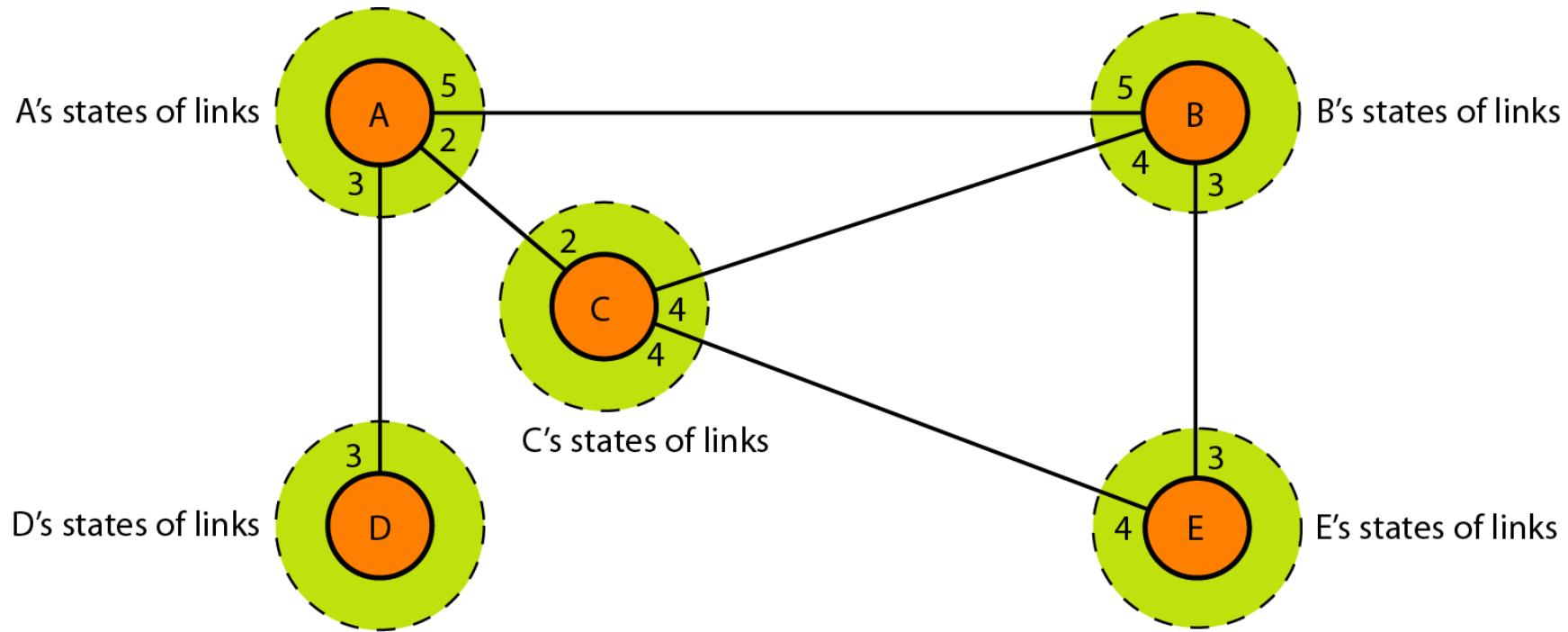


图 22.20 链路状态路由选择的概念

图 22.21 链路状态知识



每个节点知道自己这一部分的链路状态（如类型、状态和代价），整个拓扑可由每一个节点的部分知识复合而成。

链路状态路由选择的例子

目的地	总费用(元)	下一站
A 楼	0	本站
海棠公寓	0.5	海棠公寓
竹园公寓	1.3	海棠公寓
G 楼	3.2	丁香公寓
远望谷	2.2	丁香公寓
行政楼	1.9	丁香公寓
图书馆	1.1	丁香公寓
丁香公寓	0.6	丁香公寓



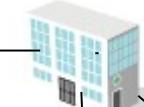
A 楼

海棠公寓



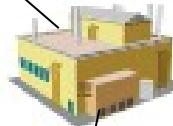
0.8 元

竹园公寓



2 元

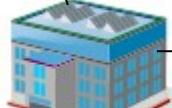
G 楼



A 楼

0.5 元

1 元



图书馆



0.8 元

行政楼

0.3 元



远望谷

0.5 元



丁香公寓

2 元

建立路由表

- 按每个节点建立链路状态分组 LSP 的链路状态；
 - 用洪泛法（flooding）向其它路由器扩散 LSP；
 - 为每个节点构成一个最短路径树；
-
- 22.57 ■ 基于最短路径树计算路由表。

生成链路状态分组 LSP

- 链路状态分组携带大量信息，如节点标识、链路清单、序列号和寿命。
- 生成 LSP 的情况
 - 区域的拓扑发生变化时
 - 周期性产生：60 分钟 ~2 小时

洪泛 flooding

- 创建节点的 LSP，并从每个接口发送 LSP 副本。
- 收到 LSP 的每个节点与已有的副本比较：
 - 丢弃旧的，保留新的；
 - 节点通过每个接口（接收副本的接口除外），再次转发副本。

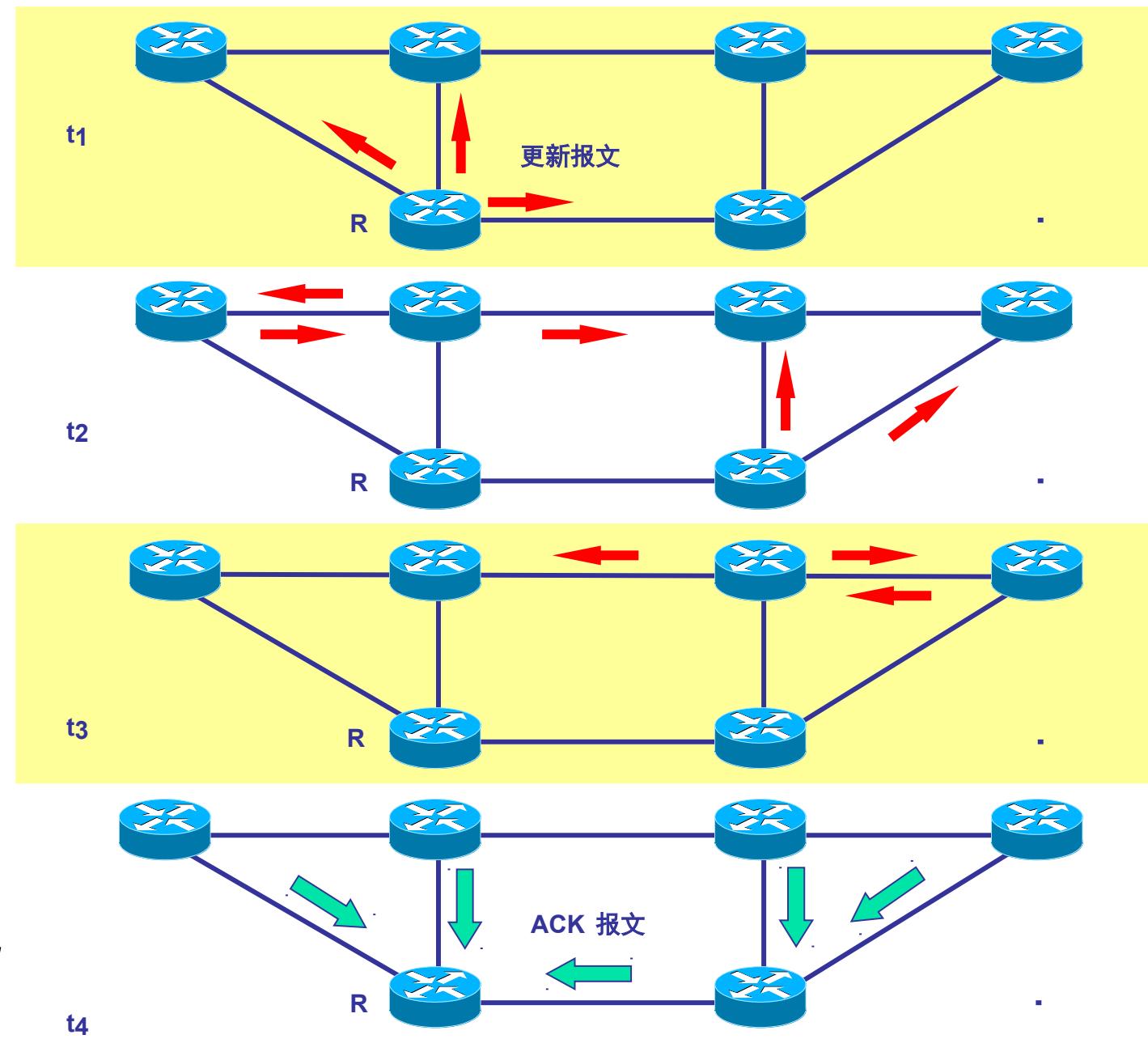
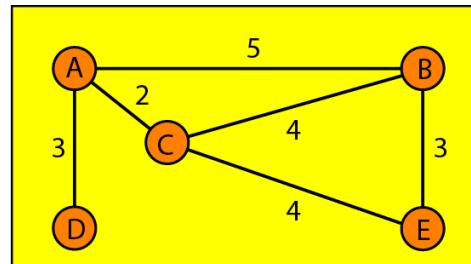


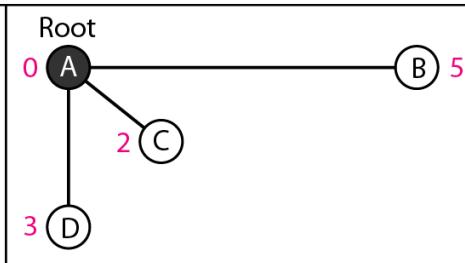
表 22.2 节点 A 的路由表

图 22.23 最短路径树构成的范例



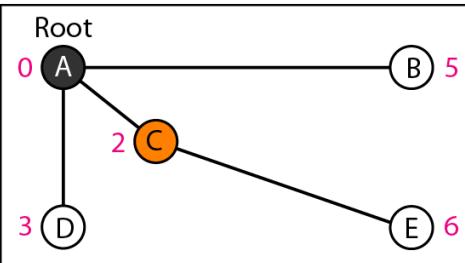
Topology

Node	Cost	Next Router
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

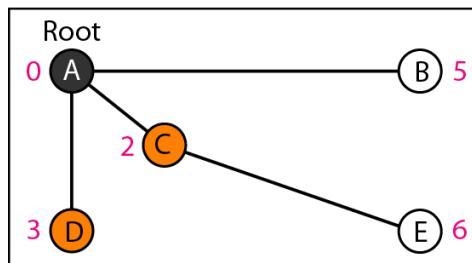


1. Set root to A and move A to tentative list.

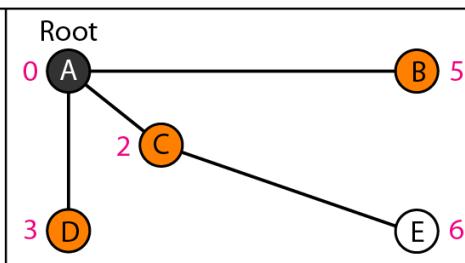
2. Move A to permanent list and add B, C, and D to tentative list.



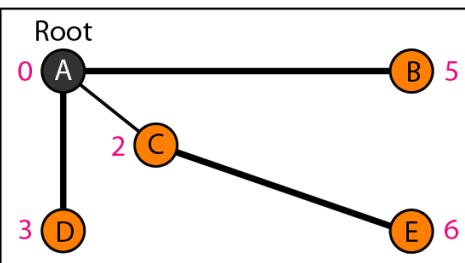
3. Move C to permanent and add E to tentative list.



4. Move D to permanent list.



5. Move B to permanent list.

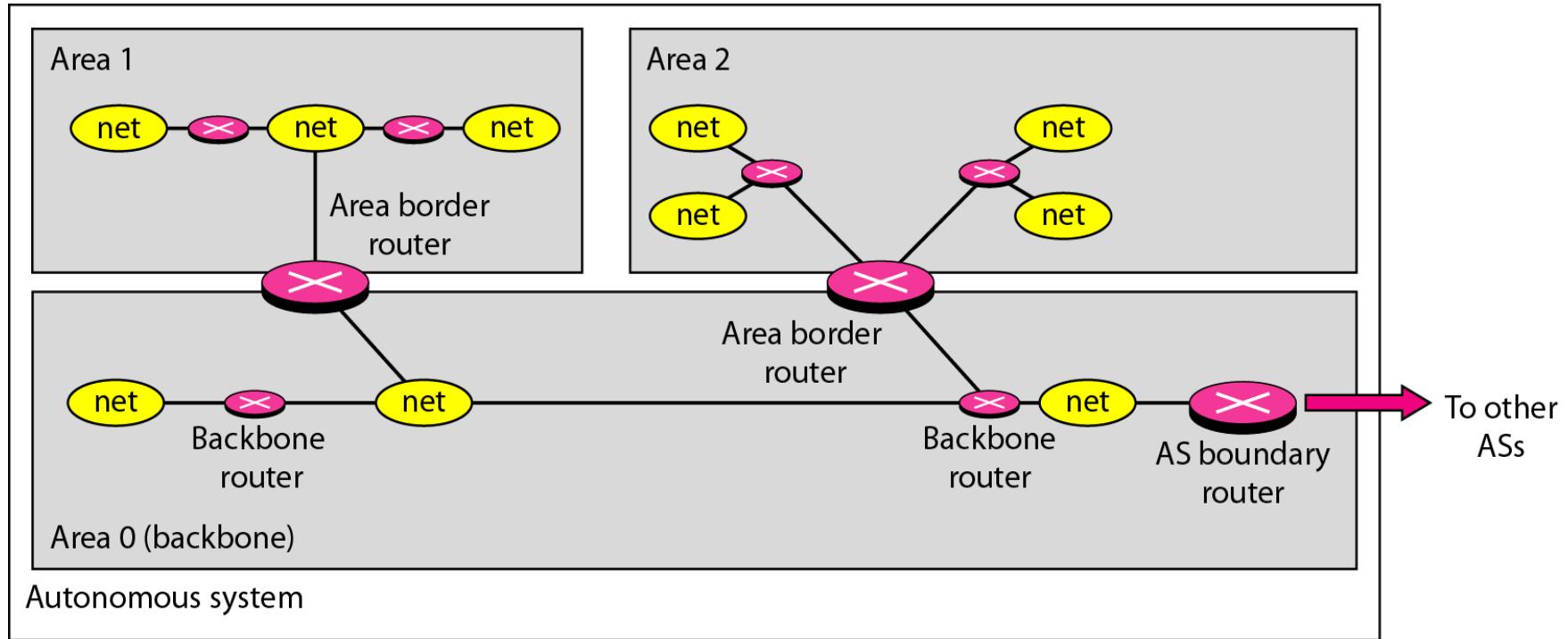


6. Move E to permanent list
(tentative list is empty).

最短路径优先协议 OSPF(Open Shortest Path First)

- Open 表示 OSPF 协议不受某一家厂商控制，是公开发表的。SPF 表示使用 Dijkstra 最短路径优先算法（但并不表示其它路由选择协议不是“最短路径优先”）。
- OSPF 是分布式的**链路状态**协议，其工作过程如下：
 - 了解自身链路：每台路由器了解其自身的链路，即与其直连的网络。
 - 寻找邻居：不同于 RIP，OSPF 协议运行后，并不立即向网络广播路由信息，而是先寻找网络中可与自己交换链路状态信息的周边路由器。可以交互链路状态信息的路由器互为邻居。
 - 创建链路状态数据包：建立了邻居关系后就可以创建链路状态数据包。
 - 链路状态信息传递：路由器将描述链路状态的 LSA 洪泛到邻居，最终形成包含网络完整链路状态信息的链路状态数据库。
 - 计算路由：路由区域内的每台路由器都可使用 SPF 算法来独立计算路由。

图 22.24 自治系统中的区域

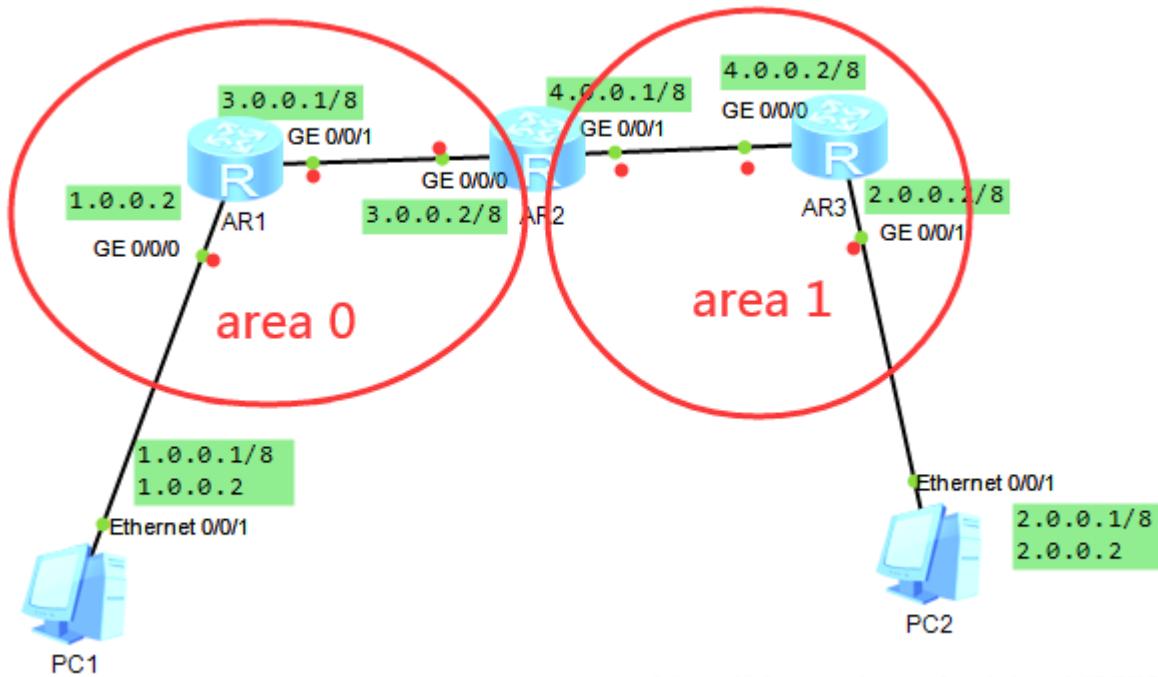


- 区域是 AS 中一些网络、主机和路由器的集合，区域之间是互联的；
- 每个区域都有标识，用 32 bit 的区域标识符表示（点分十进制）；
- OSPF 使用层次结构的区域划分，主干区域 0.0.0.0 为必需的，用来连通其它非主干区域；
- 区域不能太大，一个区域内的路由器最好不超过 200 个。

划分区域的优点

- 缩小数据库规模：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，减少了整个网络上的通信量。
- 方便路由控制：在一个区域内部的路由器只知道本区域的完整网络拓扑，而不知道其它区域的网络拓扑的情况。
- 加快收敛，增强稳定性，扩展性强。

OSPF 多区域配置



https://blog.csdn.net/weixin_44657888

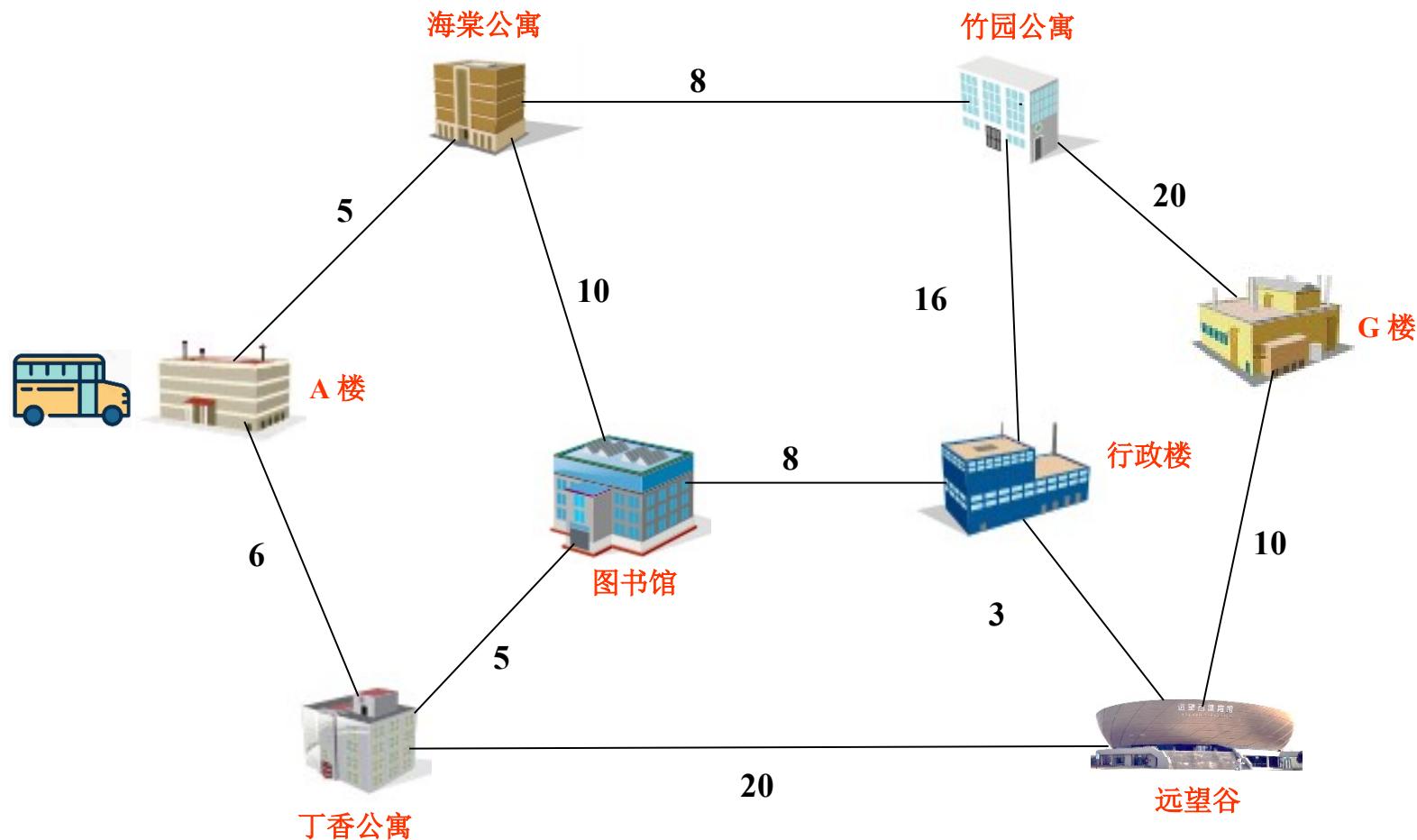
```
[AR2]ospf 1 router-id 5.5.5.5
[AR2-ospf-1]area 0
[AR2-ospf-1-area-0.0.0.0]network 3.0.0.0 0.255.255.255
[AR2-ospf-1-area-0.0.0.0]network 5.5.5.5 0.0.0.255

[AR2-ospf-1-area-0.0.0.0]area 1
[AR2-ospf-1-area-0.0.0.1]network 4.0.0.0 0.255.255.255
```

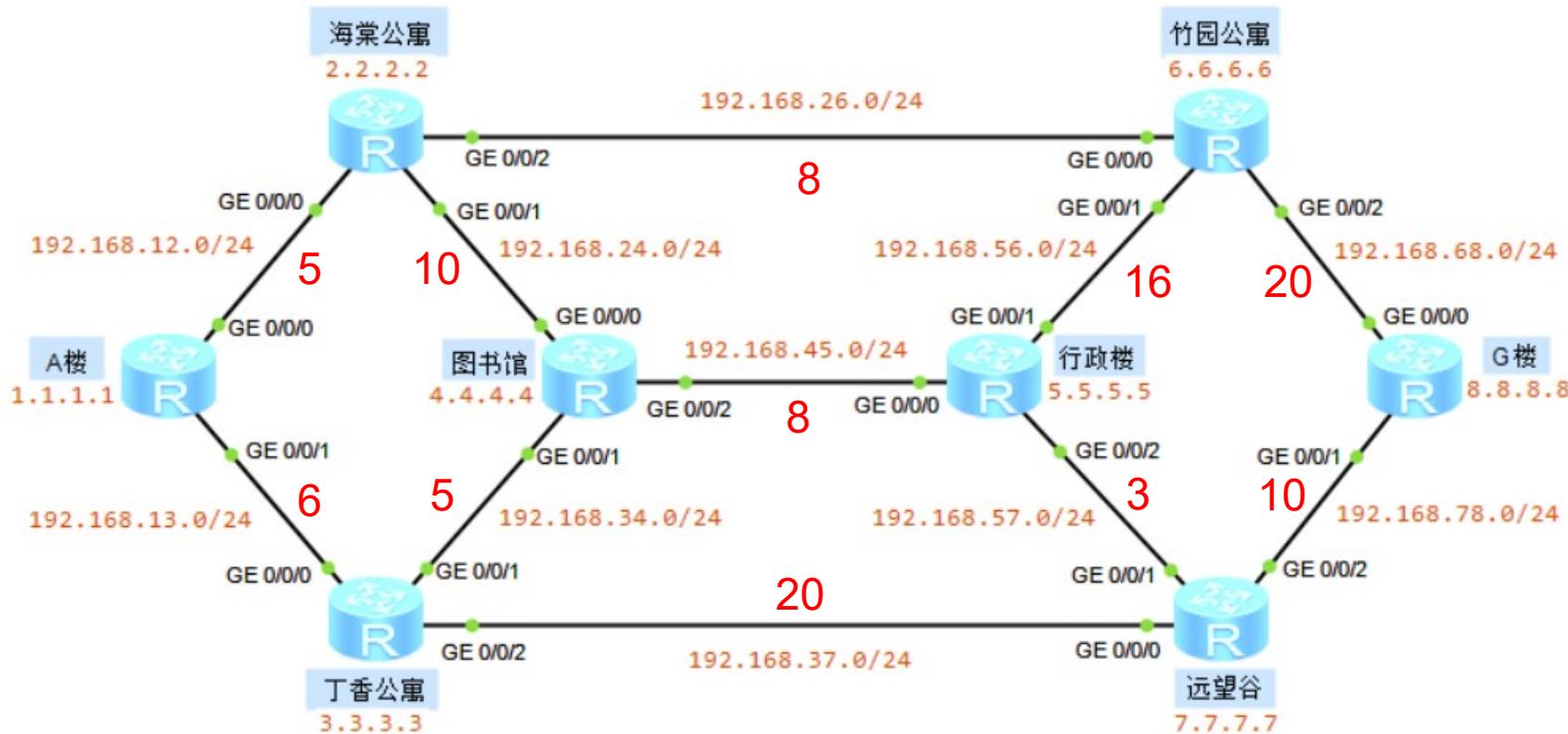
OSPF 协议的特点

- 适合大范围的网络： OSPF 对路由的跳数没有限制，所以可用于多种场合，也支持大的网络规模。
- 组播触发式更新： OSPF 在收敛完成后，会以触发方式发送拓扑变化的信息给其他路由器，可减少网络宽带的利用率；同时，可以减小对其他设备的干扰。
- 收敛速度快： 网络结构出现改变时 OSPF 系统会以最快的速度发出新的报文，从而使新的拓扑情况很快扩散到整个网络。
- 以开销作为度量值： OSPF 协议在设计时，就考虑到了链路带宽对路由度量值的影响。 OSPF 是以开销值作为标准，带宽越高开销就越小，因而 OSPF 选路主要基于带宽因素。
- 避免路由环路： 使用最短路径算法不会产生环路。
- 应用广泛： 应用最广泛的 IGP 之一。

OSPF 协议的实现



网络拓扑



路由器的配置

● 配置环回地址

```
[Huawei]int loopback0
```

```
[Huawei-LoopBack0]ip address 1.1.1.1 32
```

● 配置端口地址

```
[Huawei]int g0/0/0
```

```
[Huawei-GigabitEthernet0/0/0] ip address 192.168.12.1 24
```

```
[Huawei]int g0/0/1
```

```
[Huawei-GigabitEthernet0/0/0] ip address 192.168.13.1 24
```

● 配置端口代价

```
[Huawei]int g0/0/0
```

```
[Huawei-GigabitEthernet0/0/0]ospf cost 5
```

```
[Huawei]int g0/0/1
```

```
[Huawei-GigabitEthernet0/0/0]ospf cost 6
```

● 配置 OSPF 协议

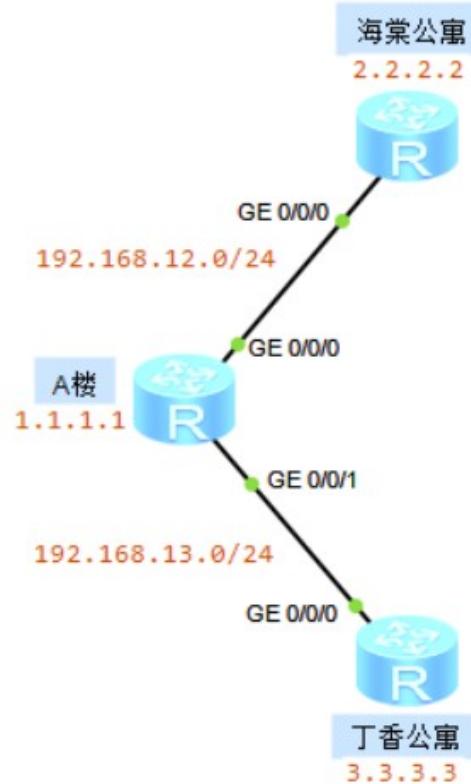
```
[Huawei]ospf 100
```

```
[Huawei-ospf-100]area 0
```

```
[Huawei-ospf-100-area-0.0.0.0] network 1.1.1.1 0.0.0.0
```

```
[Huawei-ospf-100-area-0.0.0.0] network 192.168.12.0 0.0.0.255
```

```
[Huawei-ospf-100-area-0.0.0.0] network 192.168.13.0 0.0.0.255
```



查看 OSPF 邻居的相关信息

```
<Huawei>display ospf peer

Area 0.0.0.0 interface 192.168.12.1(GigabitEthernet0/0/0)'s neighbors
Router ID: 192.168.12.2      Address: 192.168.12.2
    State: Full  Mode:Nbr is Master  Priority: 1
    DR: 192.168.12.2  BDR: 192.168.12.1  MTU: 0
    Dead timer due in 32  sec
    Retrans timer interval: 0
    Neighbor is up for 00:25:05
    Authentication Sequence: [ 0 ]

Area 0.0.0.0 interface 192.168.13.1(GigabitEthernet0/0/1)'s neighbors
Router ID: 192.168.13.2      Address: 192.168.13.2
    State: Full  Mode:Nbr is Master  Priority: 1
    DR: 192.168.13.2  BDR: 192.168.13.1  MTU: 0
    Dead timer due in 30  sec
    Retrans timer interval: 4
    Neighbor is up for 00:25:02
    Authentication Sequence: [ 0 ]
```

查看 OSPF 协议路由表

```
<Huawei>display ip routing-table protocol ospf  
Route Flags: R - relay, D - download to fib
```

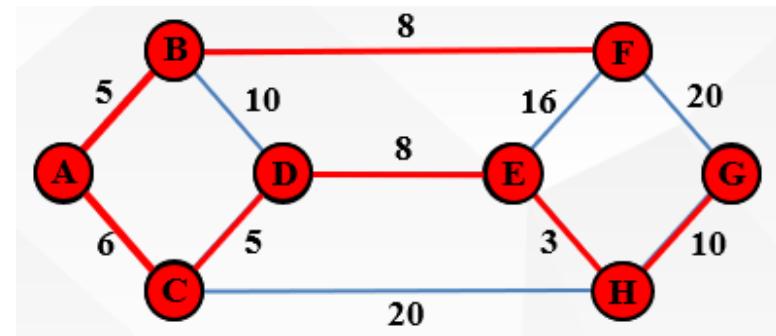
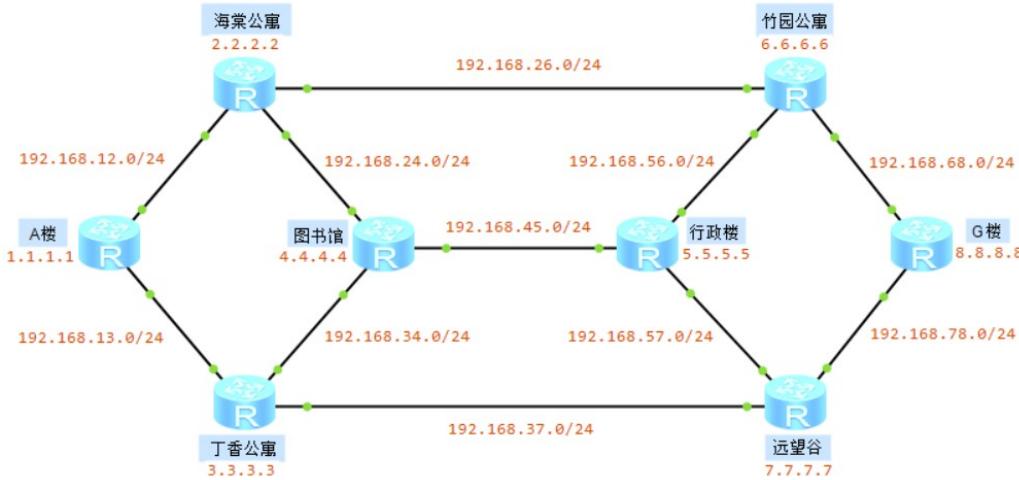
查看 OSPF 协议路由表

```
-----  
Public routing table : OSPF  
Destinations : 16 Routes : 16
```

```
OSPF routing table status : <Active>  
Destinations : 16 Routes : 16
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
2.2.2.2/32	OSPF	10	5	D	192.168.12.2	GigabitEthernet0/0/0
3.3.3.3/32	OSPF	10	6	D	192.168.13.2	GigabitEthernet0/0/1
4.4.4.4/32	OSPF	10	11	D	192.168.13.2	GigabitEthernet0/0/1
5.5.5.5/32	OSPF	10	19	D	192.168.13.2	GigabitEthernet0/0/1
6.6.6.6/32	OSPF	10	13	D	192.168.12.2	GigabitEthernet0/0/0
7.7.7.7/32	OSPF	10	22	D	192.168.13.2	GigabitEthernet0/0/1
8.8.8.8/32	OSPF	10	32	D	192.168.13.2	GigabitEthernet0/0/1
192.168.24.0/24	OSPF	10	15	D	192.168.12.2	GigabitEthernet0/0/0
192.168.26.0/24	OSPF	10	13	D	192.168.12.2	GigabitEthernet0/0/0
192.168.34.0/24	OSPF	10	11	D	192.168.13.2	GigabitEthernet0/0/1
192.168.37.0/24	OSPF	10	26	D	192.168.13.2	GigabitEthernet0/0/1
192.168.45.0/24	OSPF	10	19	D	192.168.13.2	GigabitEthernet0/0/1
192.168.56.0/24	OSPF	10	29	D	192.168.12.2	GigabitEthernet0/0/0
192.168.57.0/24	OSPF	10	22	D	192.168.13.2	GigabitEthernet0/0/1
192.168.68.0/24	OSPF	10	33	D	192.168.12.2	GigabitEthernet0/0/0
192.168.78.0/24	OSPF	10	32	D	192.168.13.2	GigabitEthernet0/0/1

追踪路由



```
<Huawei>tracert 8.8.8.8
```

```
traceroute to 8.8.8.8(8.8.8.8),
max hops: 30 , packet length: 40

1 192.168.13.2 30 ms 20 ms 20 ms
2 192.168.34.2 40 ms 30 ms 20 ms
3 192.168.45.2 50 ms 20 ms 30 ms
4 192.168.57.2 30 ms 40 ms 40 ms
5 192.168.78.2 50 ms 40 ms 40 ms
```

节点	代价	路径
B	5	A-B
C	6	A-C
D	11	A-C-D
E	19	A-C-D-E
F	13	A-B-F
G	32	A-C-D-E-H-G
H	22	A-C-D-E-H

路径向量路由选择协议

- 因特网的规模太大，使得自治系统之间路由选择非常困难。对于自治系统之间的路由选择，要寻找最佳路由是很不现实的。
 - 路由表的项目数太多。
 - 当一条路径通过几个不同自治系统时，要想对这样的路径计算出有意义的代价是不太可能的。
 - 比较合理的做法是在自治系统之间交换“可达性”信息。
- 自治系统之间的路由选择必须考虑有关策略。
- 因此，路径向量路由选择协议只能是力求寻找一条能够到达目的网络且**比较好的**路由（不能兜圈子），而并非要寻找一条**最佳**路由。

Step1: 初始化 & Step2: 共享

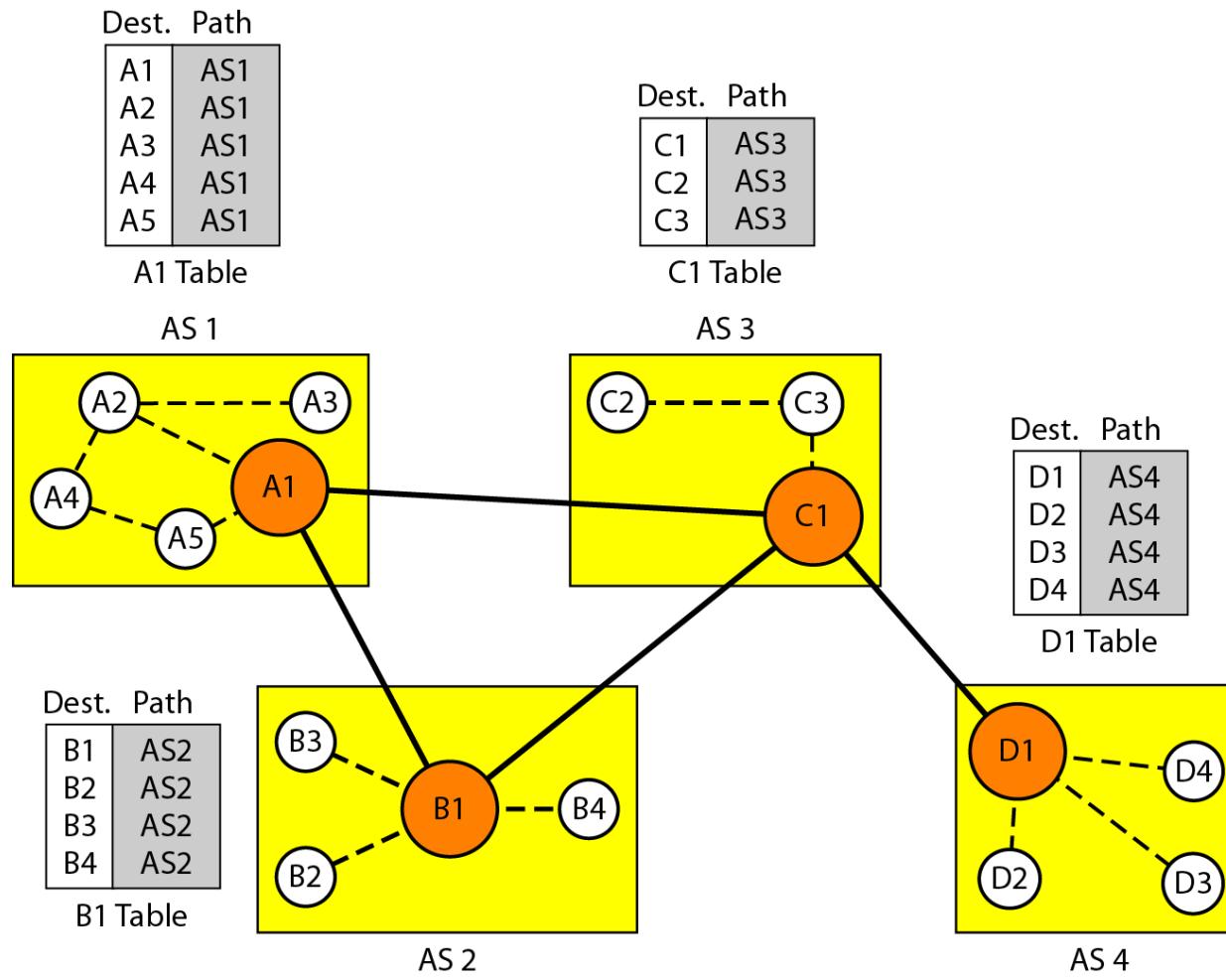


图 22.30 路径向量路由选择的初始路由表

Step3: 更新

Dest.	Path
A1 ... A5	AS1
B1 ... B4	AS1-AS2
C1 ... C3	AS1-AS3
D1 ... D4	AS1-AS3-AS4

A1 Table

Dest.	Path
A1 ... A5	AS2-AS1
B1 ... B4	AS2
C1 ... C3	AS2-AS3
D1 ... D4	AS2-AS3-AS4

B1 Table

Dest.	Path
A1 ... A5	AS3-AS1
B1 ... B4	AS3-AS2
C1 ... C3	AS3
D1 ... D4	AS3-AS4

C1 Table

Dest.	Path
A1 ... A5	AS4-AS3-AS1
B1 ... B4	AS4-AS3-AS2
C1 ... C3	AS4-AS3
D1 ... D4	AS4

D1 Table

图 22.31 三个独立系统的稳定表

路径向量路由选择协议的优点

- 预防回路：避免距离向量路由选择协议的不稳定性回路问题。
- 策略路由选择：路由器检查报文路径，如果路径中列出的某自治系统不符合策略，则忽略。
- 优化路径：符合组织机构标准的路径及保密、安全、可靠性等其它原则。

边界网关协议 BGP (Border Gateway Protocol)

- BGP 运行于 TCP 上，是唯一一个用来处理因特网大小网络的协议，也是唯一能够妥善处理好不相关路由域间的多路连接的协议。
- BGP 的主要功能是和其他 BGP 系统交换网络可达信息，包括列出的 AS 的信息，有效地构造 AS 互联的拓扑图并清除路由环路，同时在 AS 级别上可实施策略决策。
- BGP-4 在 1995 年发布，支持无类域间路由、路由聚合，每个自治系统管理员至少要选择一个路由器作为该自治系统的“BGP 发言人”。

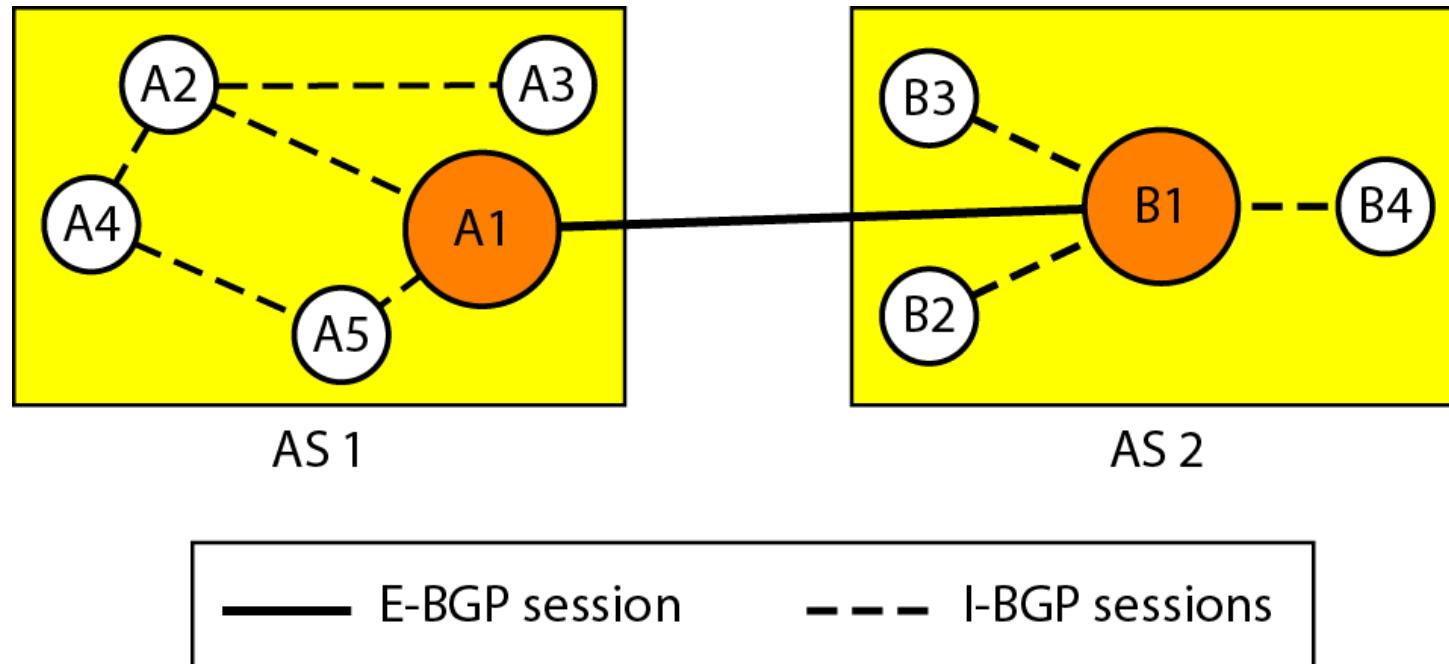
BGP 协议的特点

- 结点数量级是自治系统数的量级，比自治系统中的网络数少很多。
- 每一个自治系统中 BGP 代言节点的数目很少，使得自治系统之间的路由选择不致过分复杂。
- 支持 CIDR，因此 BGP 路由表包括目的网络前缀、下一跳路由器，以及到达该目的网络所要经过的各个自治系统序列。
- 在 BGP 刚刚运行时，BGP 的邻站是交换整个的 BGP 路由表。以后只在发生变化时更新有变化的部分，这样可节省网络带宽和减少路由器的处理开销。

BGP 会话

- 使用 BGP 的两个路由器之间交换路由信息即为会话。
- 一个 BGP 代言节点与其它自治系统中的 BGP 代言节点交换路由信息，需要先建立 TCP 连接，然后在此连接上交换 BGP 报文以建立 BGP 会话。
- 使用 TCP 连接能提供可靠的服务，也简化了路由选择协议。

图 22.32 内部和外部 BGP 会话



22-4 多播路由选择协议

在这一节讨论多播与多播路由选择协议。

本节讨论：

单播、多播与广播

应用

多播路由选择

多播路由选择协议

图 22.33 单播

在单播中，路由器将接收到的分组仅从其端口中的一个转发出去。

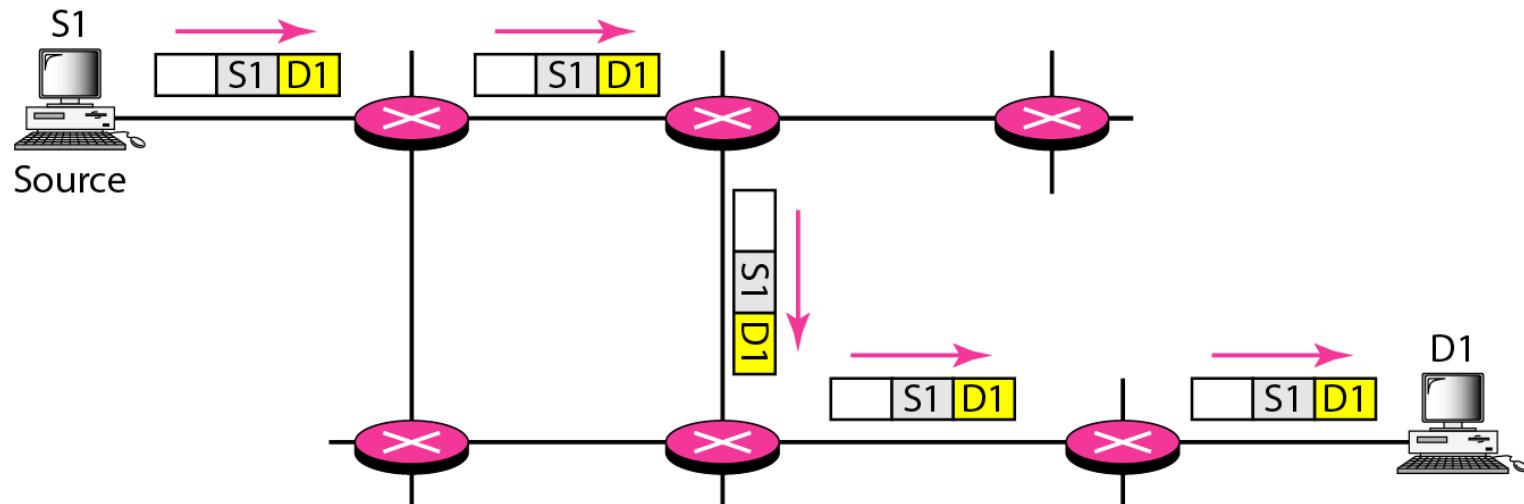


图 22.34 多播

在多播中，路由器可能通过它的多个端口将其所接收到的分组转发出去。

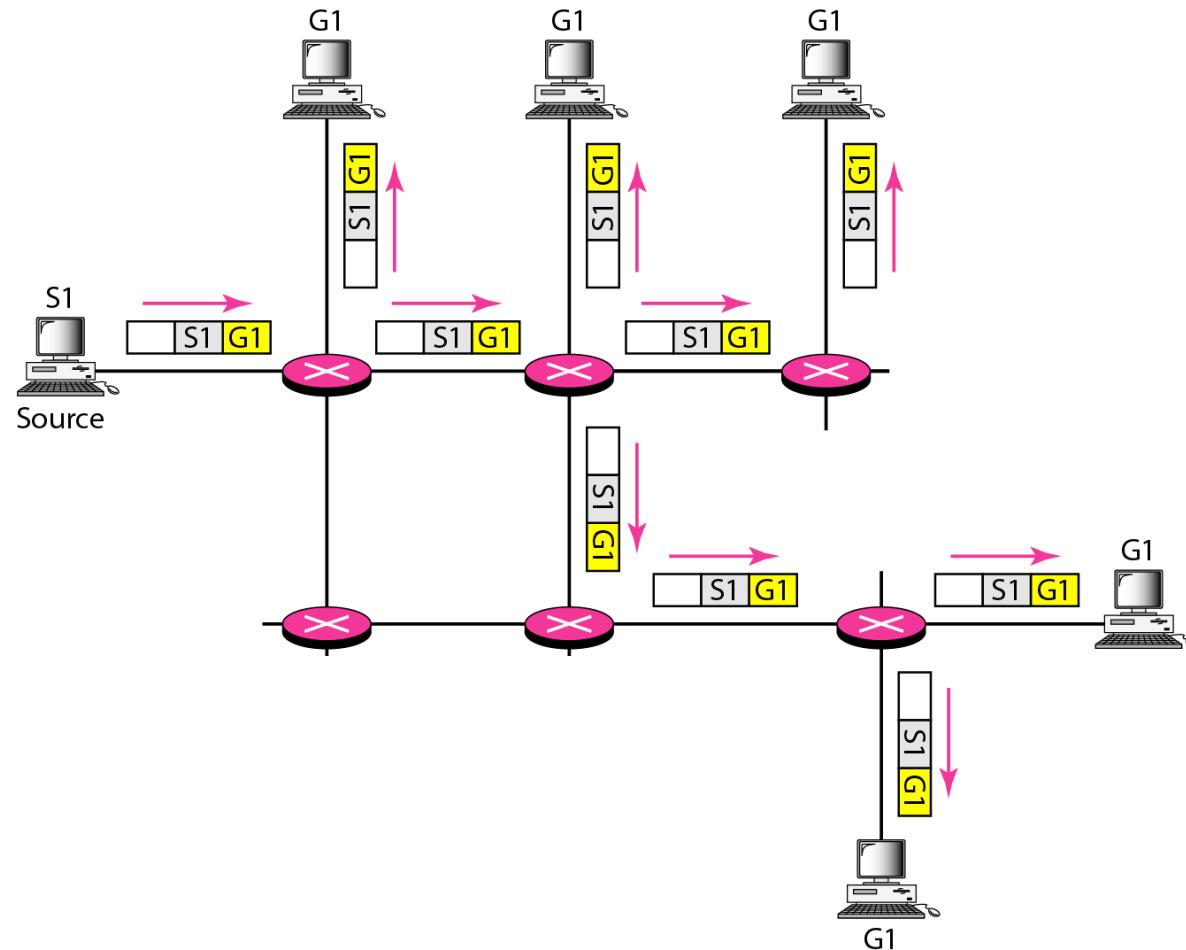
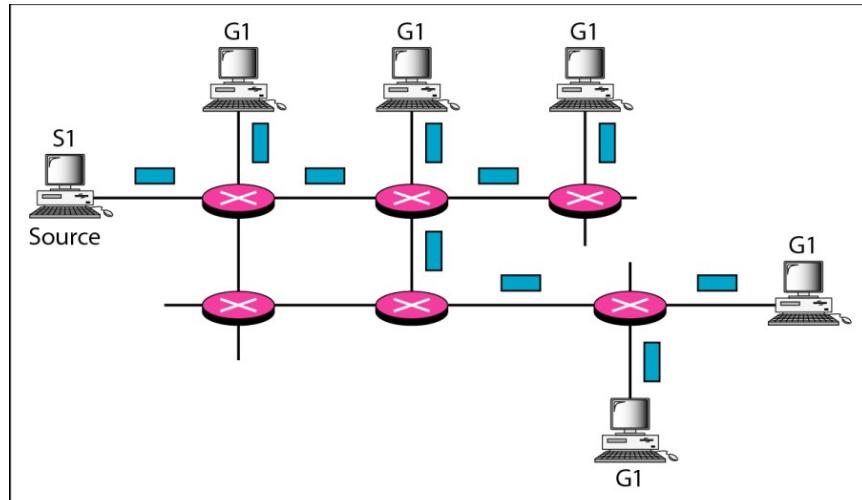
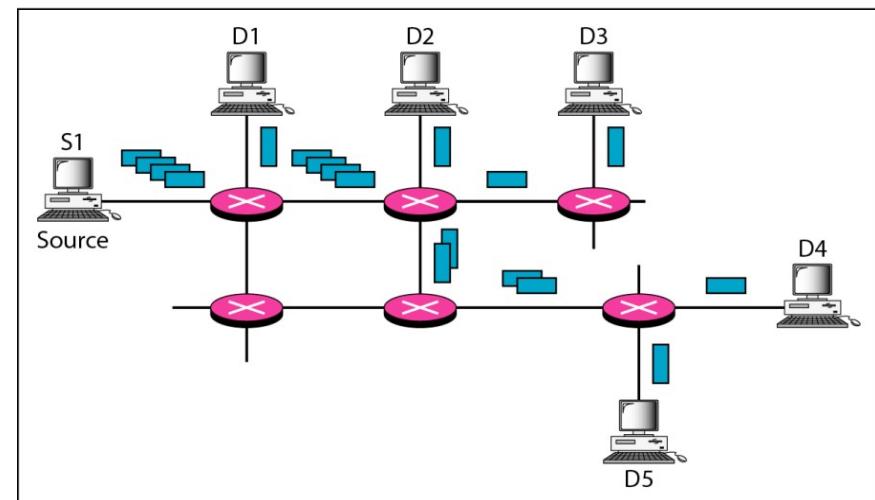


图 22.35 多播与多个单播



a. Multicasting



b. Multiple unicasting

用单播仿真多播效率不高并可能产生长的延迟，特别是对于大的分组。

图 22.36 单播路由选择中的最短路径树

在单播路由选择中，区域中的每一个路由器都有一张表，该表定义了到可能目的地址的一棵最短路径树。

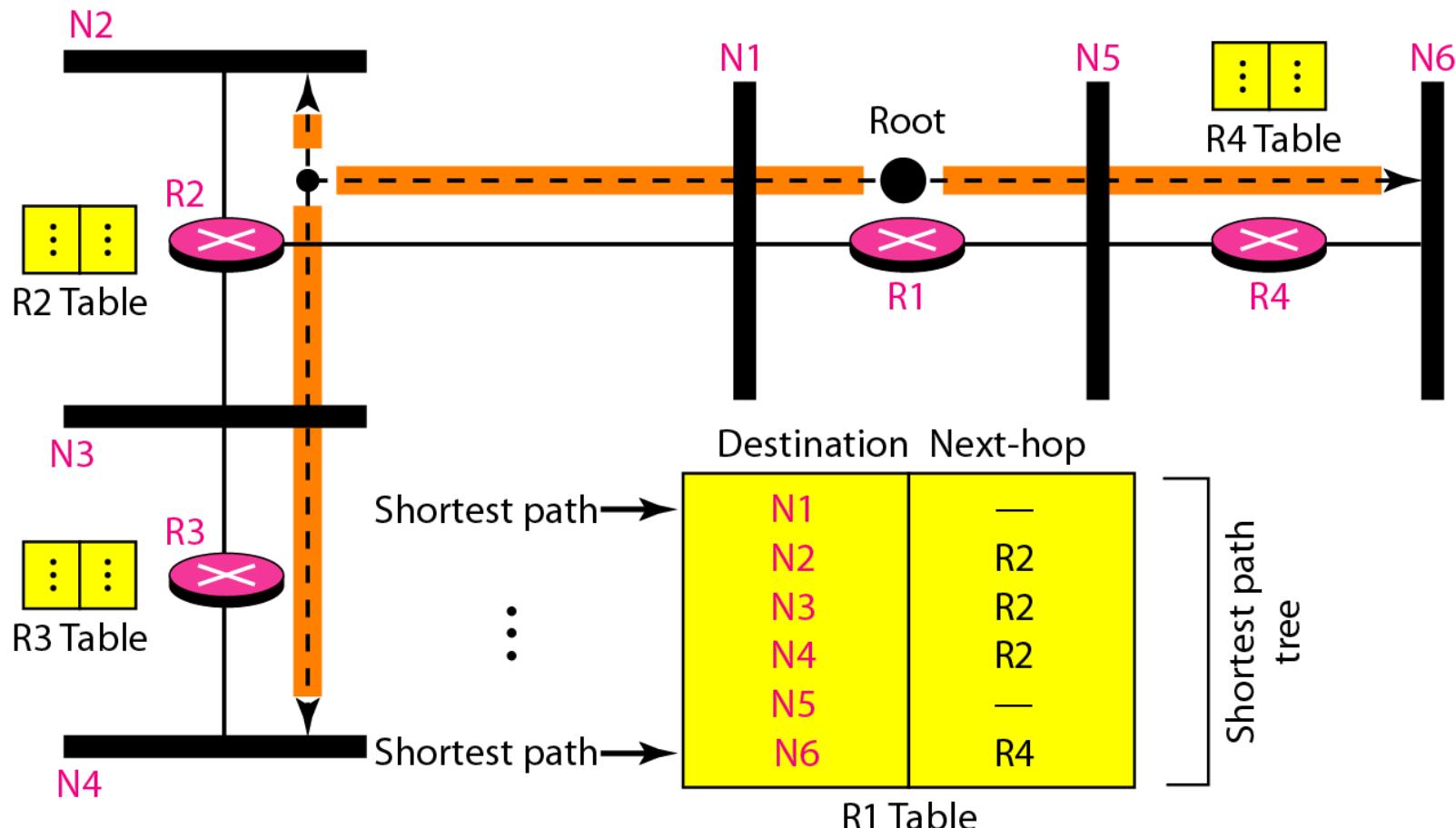


图 22.37 基于源树方法

在基于源树方法中，每个相关路由器都要为每个组构建一棵最短路径树。

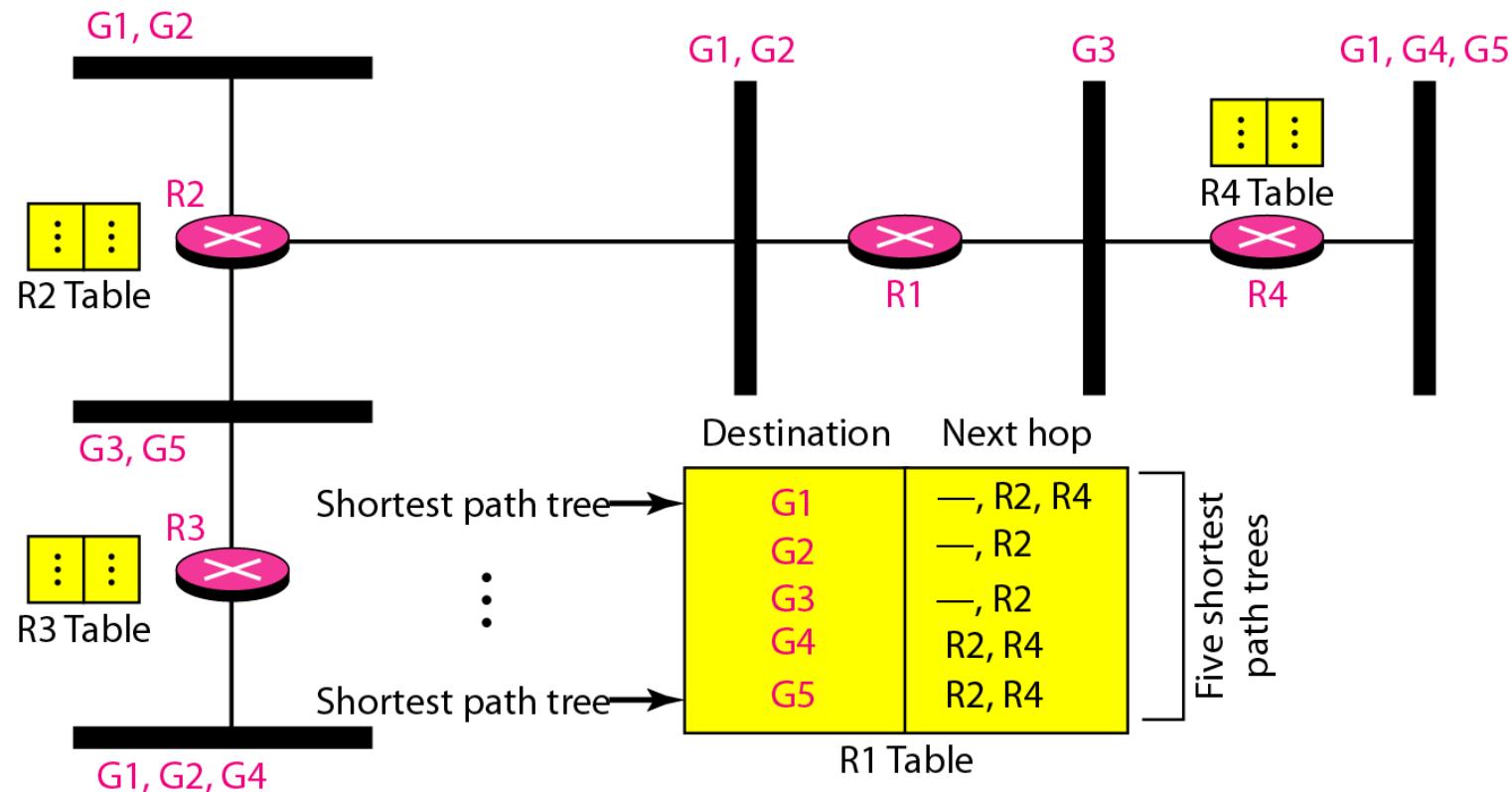


图 22.38 组共享树

在组共享树方法中，只有一个核心路由器，它对多播所涉及的每个组有一个最短路径树。

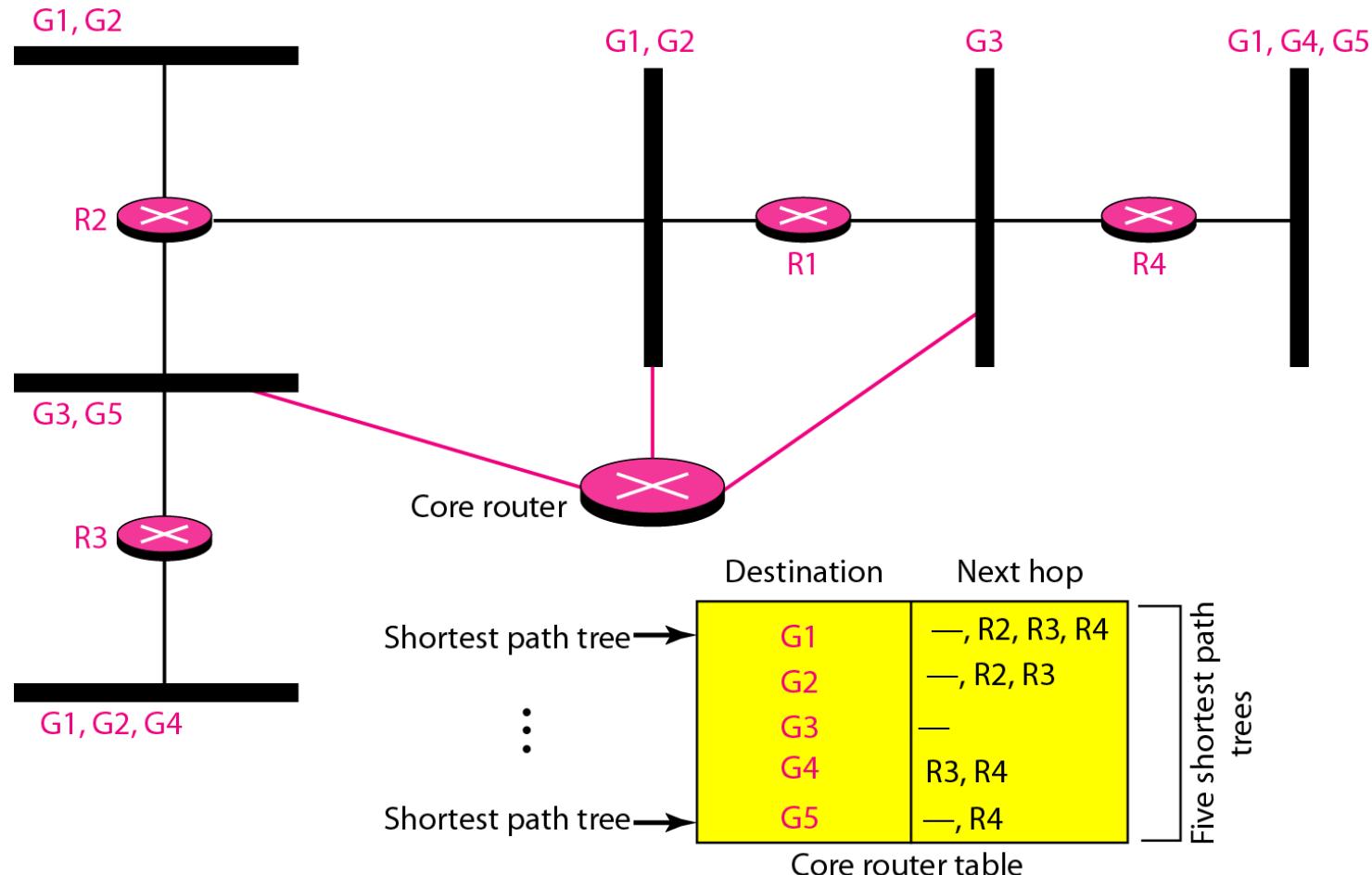
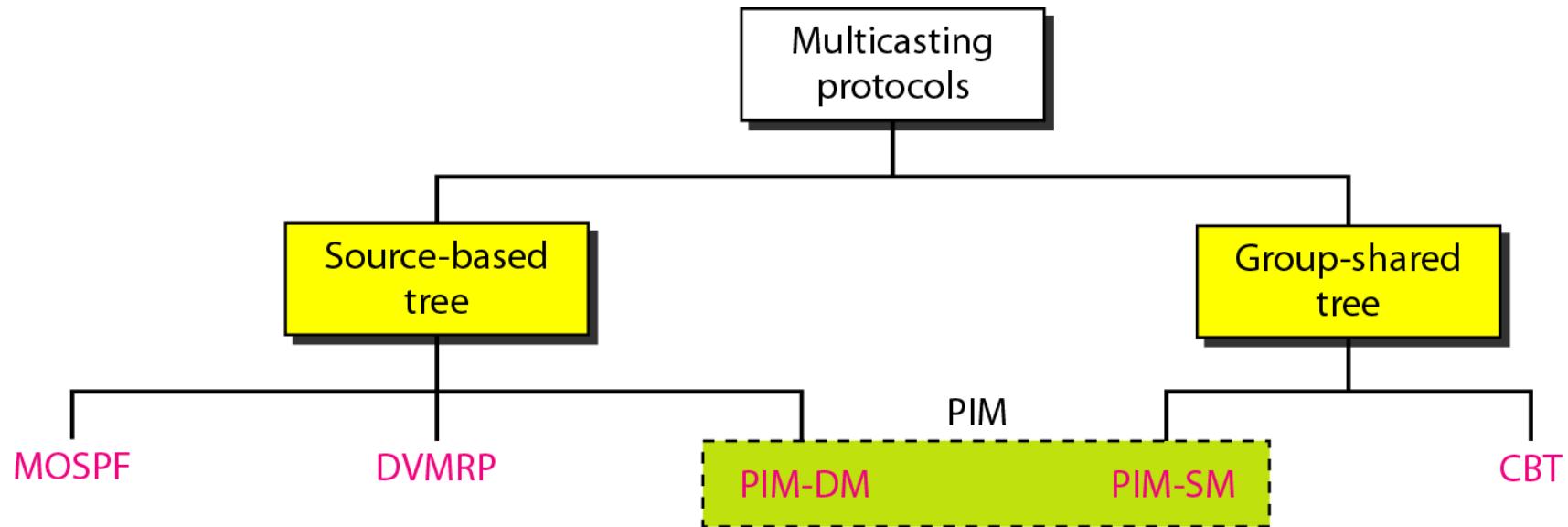


图 22.39 常用的多播路由选择协议的分类



作业

- P461 页 15 , 19
- 补充作业：使用 Dijkstra 路由选择算法，做出最小代价路由选择表和路由选择图。图中 C 为源点。

