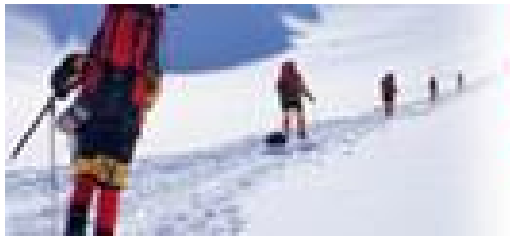


Data Communications and Networking Fourth Edition

Forouzan

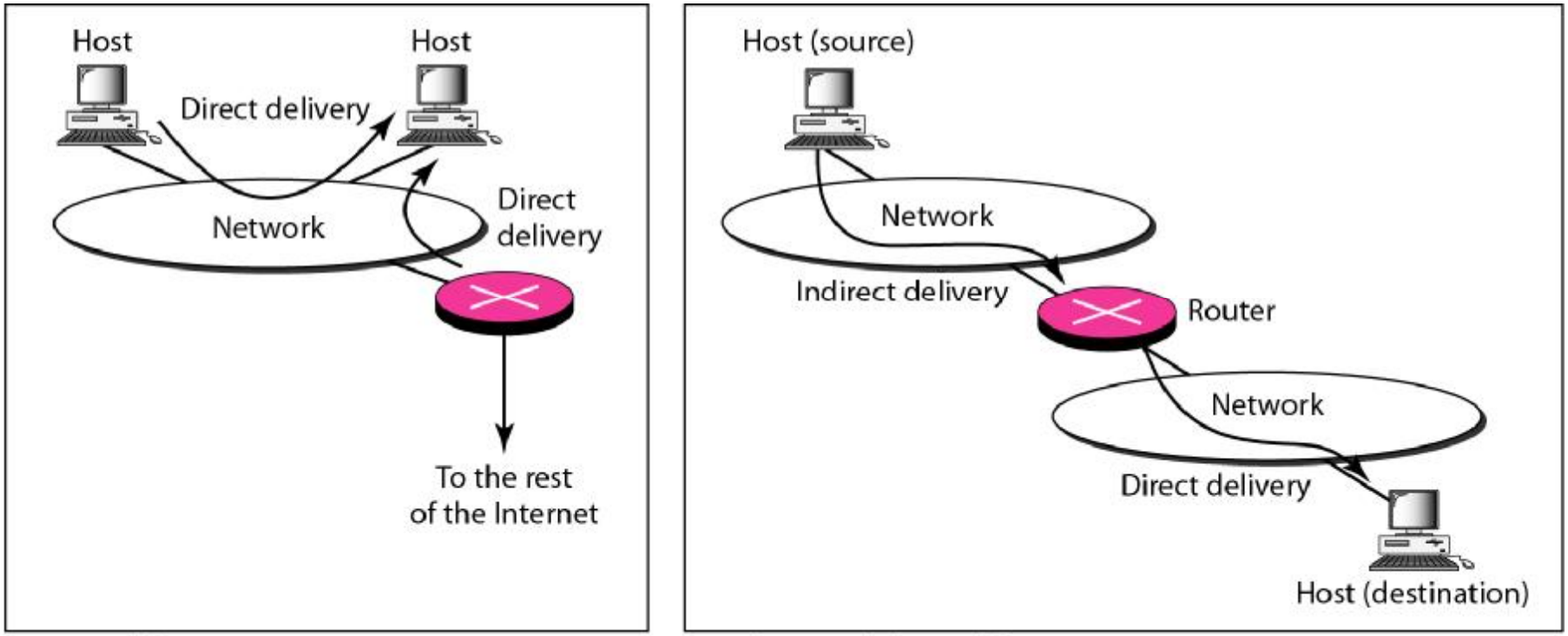
第22章

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，这种处理称为分组的传递；
- 分组传递可使用两种方法：直接传递和间接传递；
- 直接传递指分组的最终目的端的主机与发送方都连接在同一个物理网络上；发送方只要提取出分组目的端的网络地址（用掩码），并与它所连接的网络地址进行比较，如果相同，则传递就是直接的；
- 如果目的主机与发送方不在同一个网络上，分组就是间接传递；此时，分组从一个路由器传送到另一个路由器，直到它到达与最终目的端连接在同一个物理网络上的路由器为止；
- 注意：最后的传递总是直接传递。

图22.1 直接传递和间接传递



a. Direct delivery

b. Indirect and direct delivery

22-2 转发

- 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；
- 路由表中的项目数太多，使得路由表的查找效率很低，应设法简化路由表中的内容
- 简化路由表的技术：
  - 下一跳方法与路由方法
  - 特定网络方法与特定主机方法
  - 默认方法

图22.2 路由方法与下一跳方法

p路由（route）方法：在路由表中保留完整路由信息的技术；
p下一跳（next-hop）方法：在路由表中只保留下一跳地址，而不保留完整的路由信息。

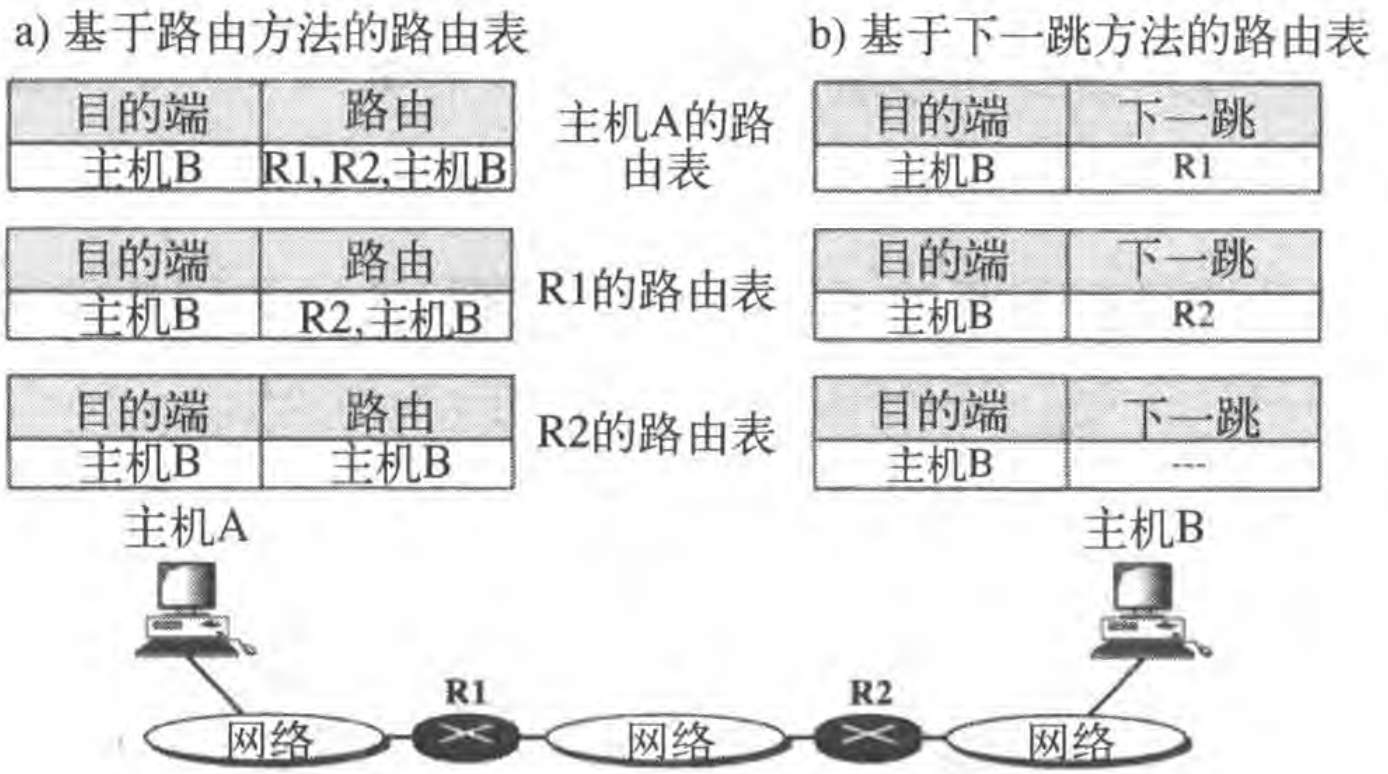


图22.3 特定主机方法与特定网络方法

p特定网络方法：仅用一个项目来定义这个目的网络本身的地址；

p特定主机方法：对连接在同一个物理网络上的每台主机都有一个项目

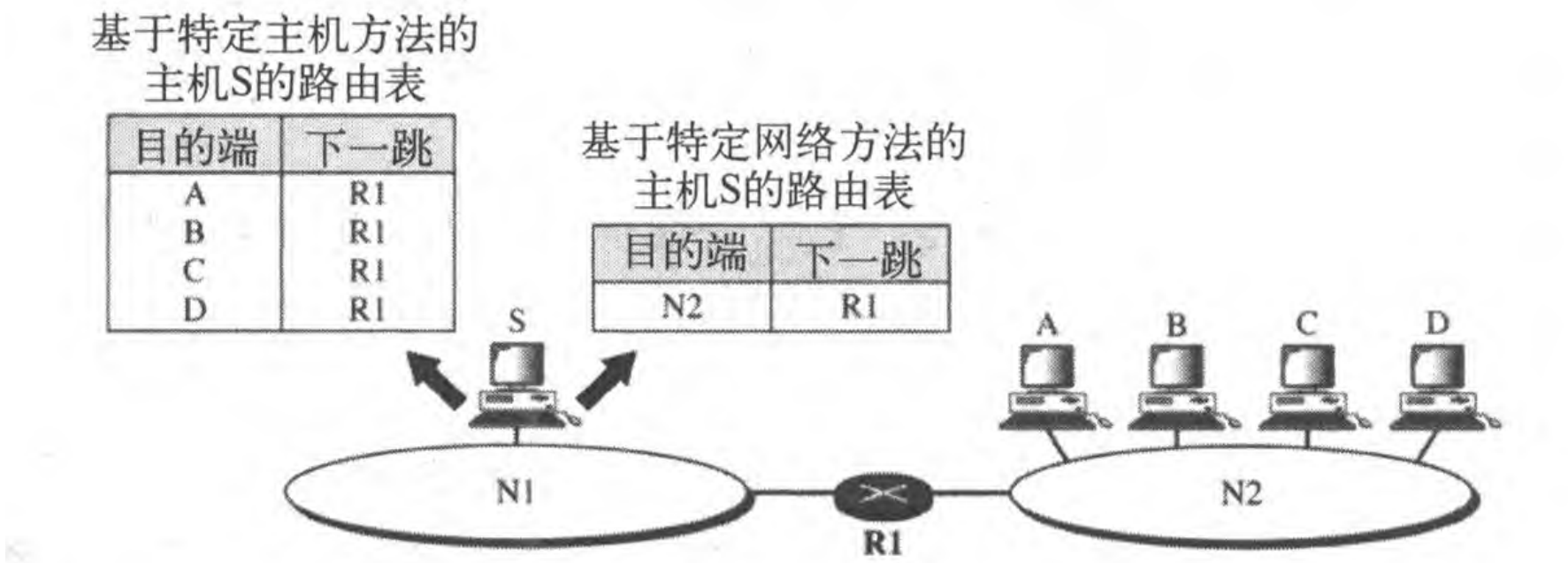
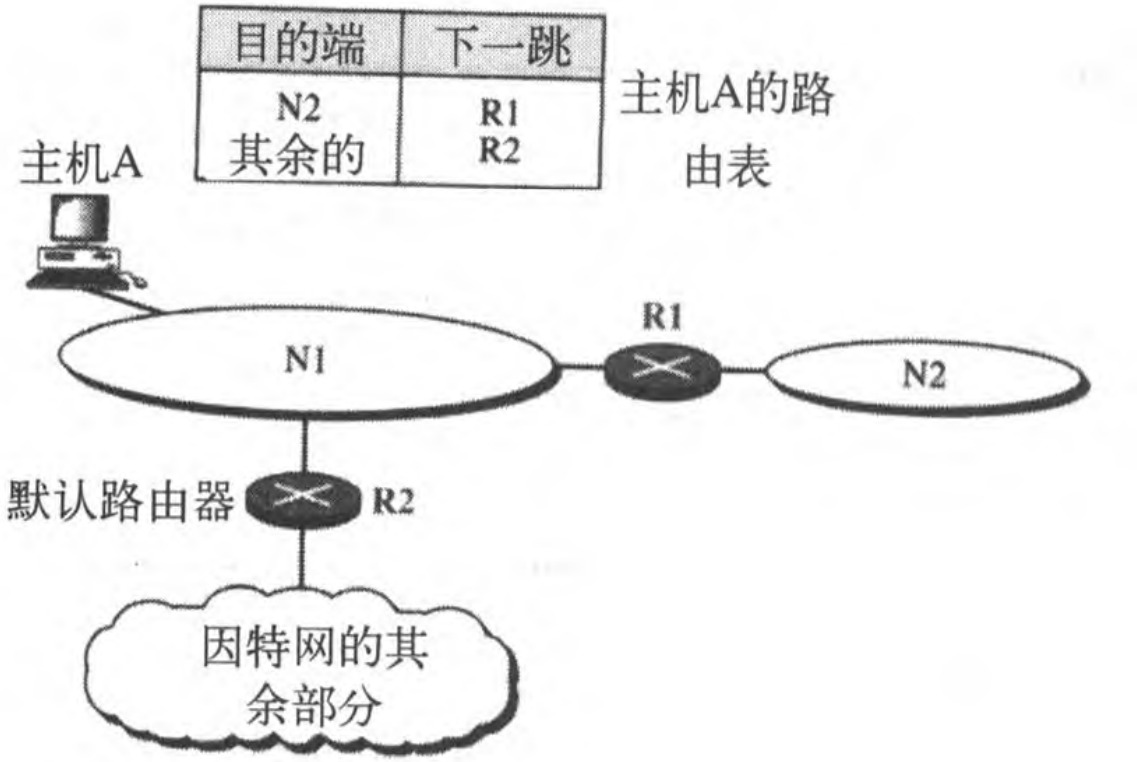


图22.4 默认方法

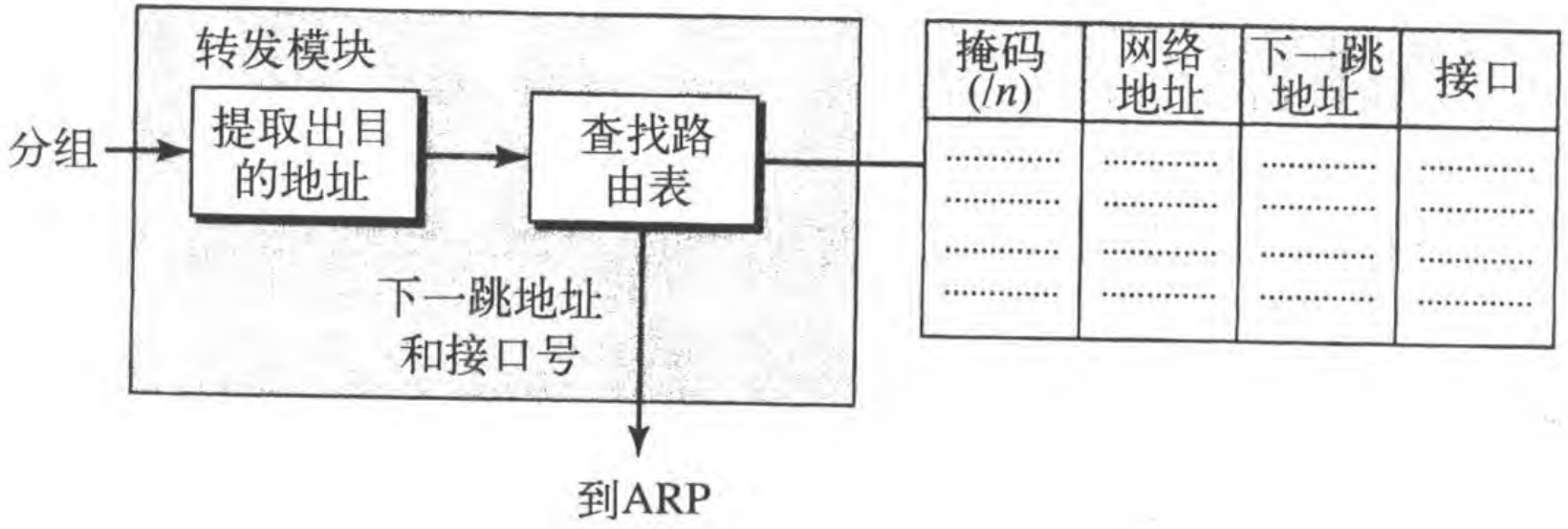
p路由器R1用来将分组转发到连接网络N2的主机，但是，对因特网的其余部分，则使用路由器R2；

p可不必将整个因特网中的所有网络都列出，主机A可以仅使用一个称为默认的项目（通常定义网络地址为0.0.0.0）。



转发过程

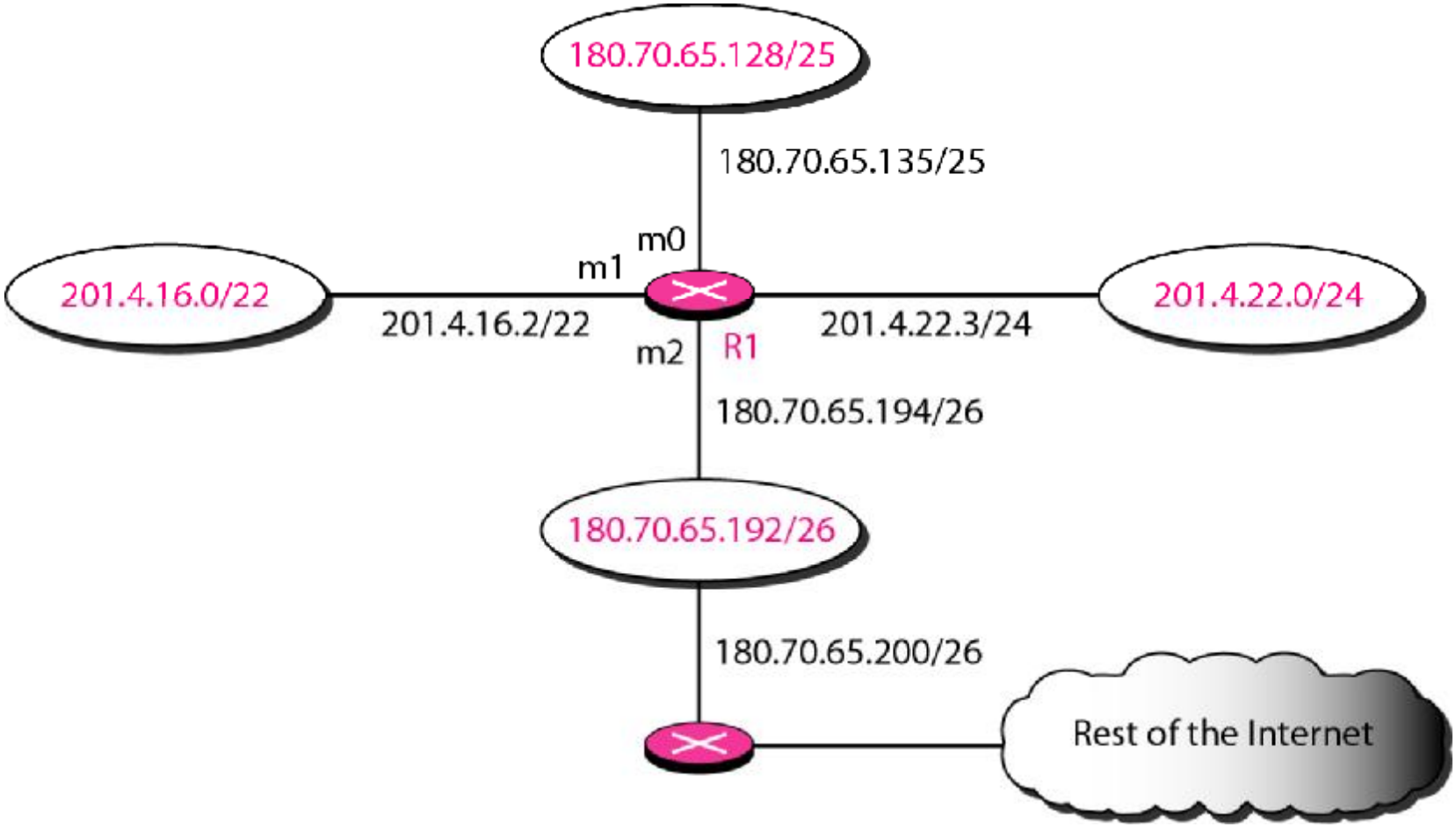
- 假定主机和路由器使用无类寻址（分类寻址可看作特例），路由表对涉及到的每一个地址块都需要有一行信息；
- 路由表需要根据网络地址（地址块的第一个地址）进行查询，但分组中只有目的地址而没有网络地址；
- 为此，在路由表中需要包含掩码（/n）；对相应的地址块，需要有包含该掩码的附加列



在无类寻址中，一个路由表至少要有4列

例22.1

利用图22.6的网络配置，做出R1的路由表。



解：

表22.1是相应的路由表。

表22.1 图 22.6中路由器R1的路由表

Mask	*Network Address*	*Next Hop*	*Interface*
/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

注意：最后一行默认路由可以表示为：

/0 0.0.0.0 180.70.65.200 m2

例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。

解：

路由器执行下列步骤：

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（nex-hop address）和接口号m0传送到ARP做进一步处理。

例22.3

如果图22.6中的一个目的地址为210.4.22.35的分组到达路由器R1，说明其转发过程。

解：

路由器执行下列步骤：

1. 第一个掩码（/26）作用于这个目的地址，其结果210.4.22.0，它与对应的网络地址不匹配（第一行）
2. 第二个掩码（/25）作用于这个目的地址，其结果是210.4.22.0，它与对应的网络地址不匹配（第二行）
3. 第三个掩码（/24）作用于这个目的地址，其结果是201.4.22.0，它与对应的网络地址匹配。分组的目的地址和接口号m3传送到ARP。

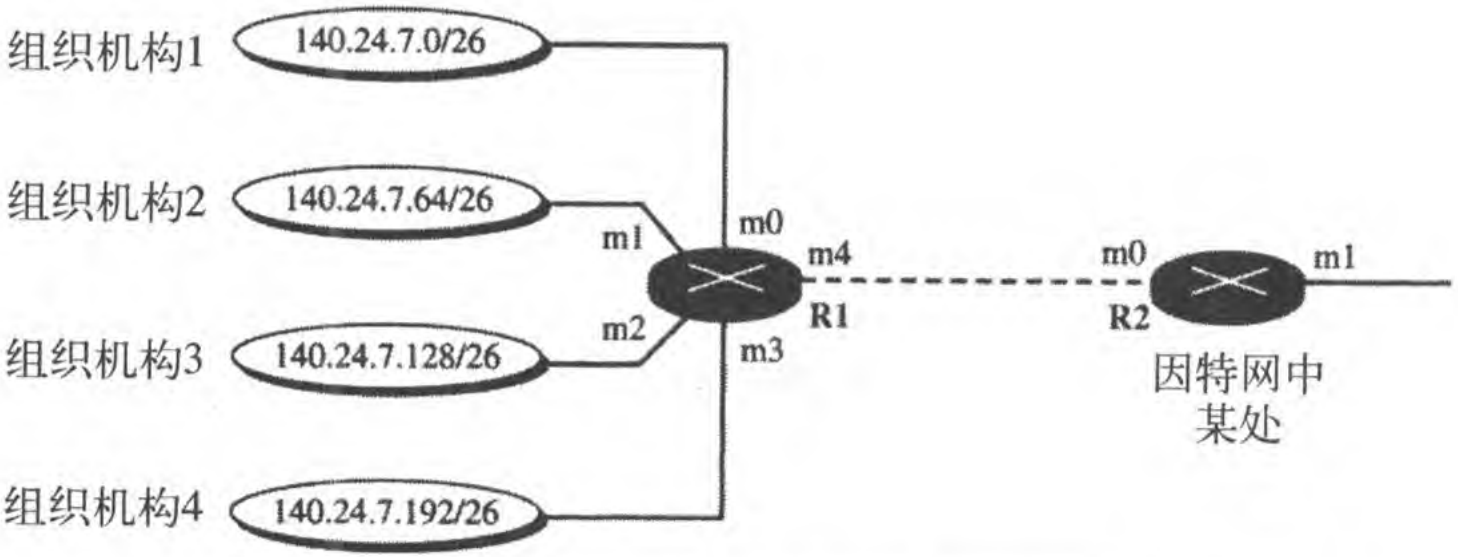
例22.4

如果图22.6中的一个目的地址为18.24.32.78的分组到达路由器R1，说明其转发过程。

解：

这时所有掩码作用于这个目的地址都与网络地址不匹配。当它到达表的尾部时，模块将下一跳地址180.70.65.200和接口号m2传送到ARP。这就是需要通过路由器转发到因特网某处的输出包。

图22.7 地址聚合（缓解路由表增大问题）



掩码	网络地址	下一跳地址	接口
/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	默认	m4

R1路由表

掩码	网络地址	下一跳地址	接口
/24	140.24.7.0	----------	m0
/0	0.0.0.0	默认	m1

R2路由表

无类别域间路由选择CIDR

- CIDR（Classless InterDomain Routing）技术有效地解决了路由缩放问题
  - 其一，大多数中等规模的企业一般拥有几千台主机，没有适合的地址空间：C类网络太小，只有254个地址，B类网络太大，有65000多个地址，A类网络就更不适用了；
  - 其二，路由表增长太快，如果C类网络号都在路由表中占一行，这样路由表就会太大，其查找速度就会降低
- CIDR优点：
  - 掩码长度更灵活；
  - 可以更加有效的分配IPv4的地址空间

无类别域间路由选择CIDR（cont.）

- 无类域间路由的基本思想是以可变长分块方式分配剩下的200万个C类地址（也可以是其他类别地址）；
- CIDR技术可以把若干个C类网络分配给一个用户，并且在路由表中只占一行，是一种将大块的地址空间合并为少量路由信息的策略；
- 称为路由表聚合/路由聚合 (routing table aggregation)；
- 假设某个机构需要1000个IP地址，就给它一个1024地址块（4个连续的C类地址），而不是一个B类地址；
- 用几个C类地址来代替一个B类地址可以解决B类地址耗尽的问题，同时不会带来路由表爆炸

无类别域间路由选择CIDR（cont.2）

- CIDR消除了传统的A类、B类和C类地址以及划分子网的概念，因而可以更加有效地分配IPv4的地址空间；
- CIDR使用各种长度的“网络前缀”(network-prefix)来代替分类地址中的网络号和子网号；
- IP地址从三级编址（使用子网掩码）又回到了两级编址

无类别域间路由选择CIDR（cont.3）

- CIDR使用“斜线记法”(slash notation)，它又称为CIDR记法，即在IP地址后面加上一个斜线“/”，然后写上网络前缀所占的比特数；
- CIDR将网络前缀都相同的连续的IP地址组成“CIDR地址块”；
- 128.14.32.0/20表示的地址块共有2^{12}个地址（因为斜线后面的20是网络前缀的比特数，所以主机号的比特数是12）；
- 这个地址块的起始地址是128.14.32.0；
- 128.14.32.0/20地址块的最小地址：128.14.32.0；
- 128.14.32.0/20地址块的最大地址：128.14.47.255；
- 全0和全1的主机号地址一般不使用

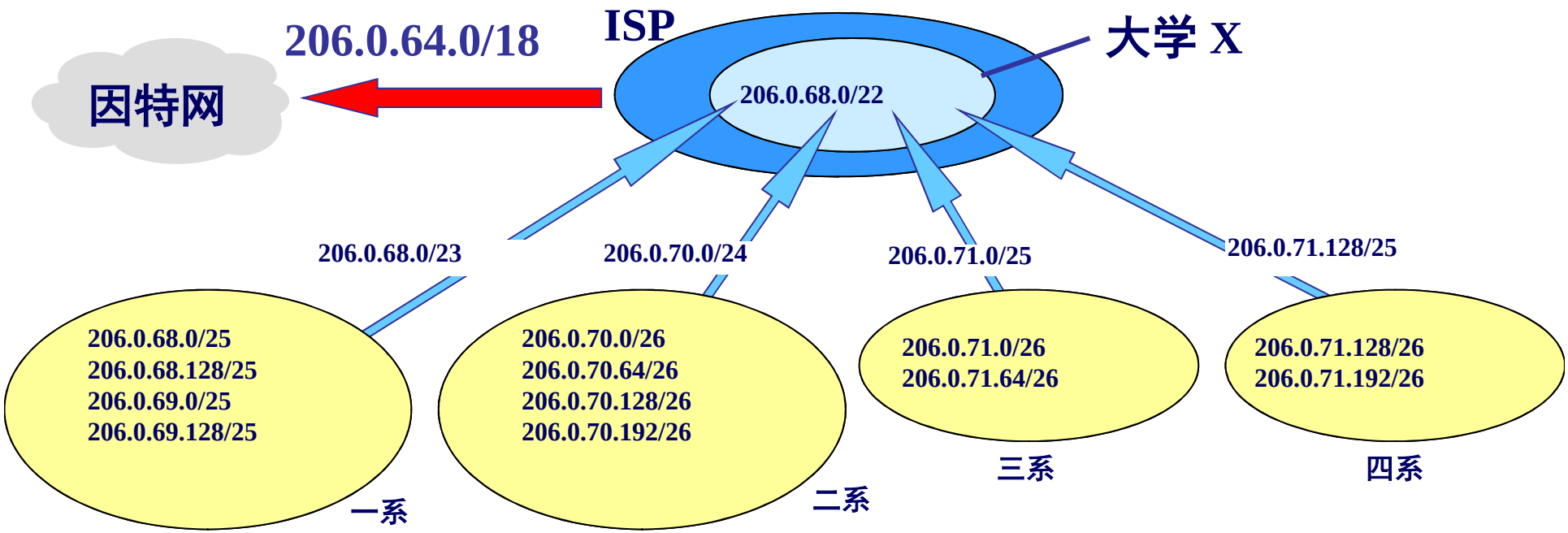
无类别域间路由选择CIDR（cont.4）

- 一个CIDR地址块可以表示很多地址，这种地址的聚合常称为路由聚合，它使得路由表中的一个项目可以表示很多个（例如上千个）原来传统分类地址的路由；
- 路由聚合也称为构成超网（supernetting）；
- CIDR虽然不使用子网了，但仍然使用“掩码”这一名词；
- 对于/20地址块，它的掩码是20个连续的1，斜线记法中的数字就是掩码中1的个数；
- CIDR地址块中的地址数一定是2的整数次幂，网络前缀越短，地址块所包含的地址数越多

练习题

- CIDR技术的作用是_____。
  - A. 把小的网络汇聚成大的超网
  - B. 把大的网络划分成小的子网
  - C. 解决地址资源不足的问题
  - D. 由多个主机共享同一个网络地址

CIDR地址块划分举例



这个ISP共有64个C类网络。如果不采用CIDR技术，则在与该ISP的路由器交换路由信息的每一个路由器的路由表中，就需要有64个项目。但采用地址聚合后，只需用路由聚合后的1个项目206.0.64.0/18就能找到该ISP。

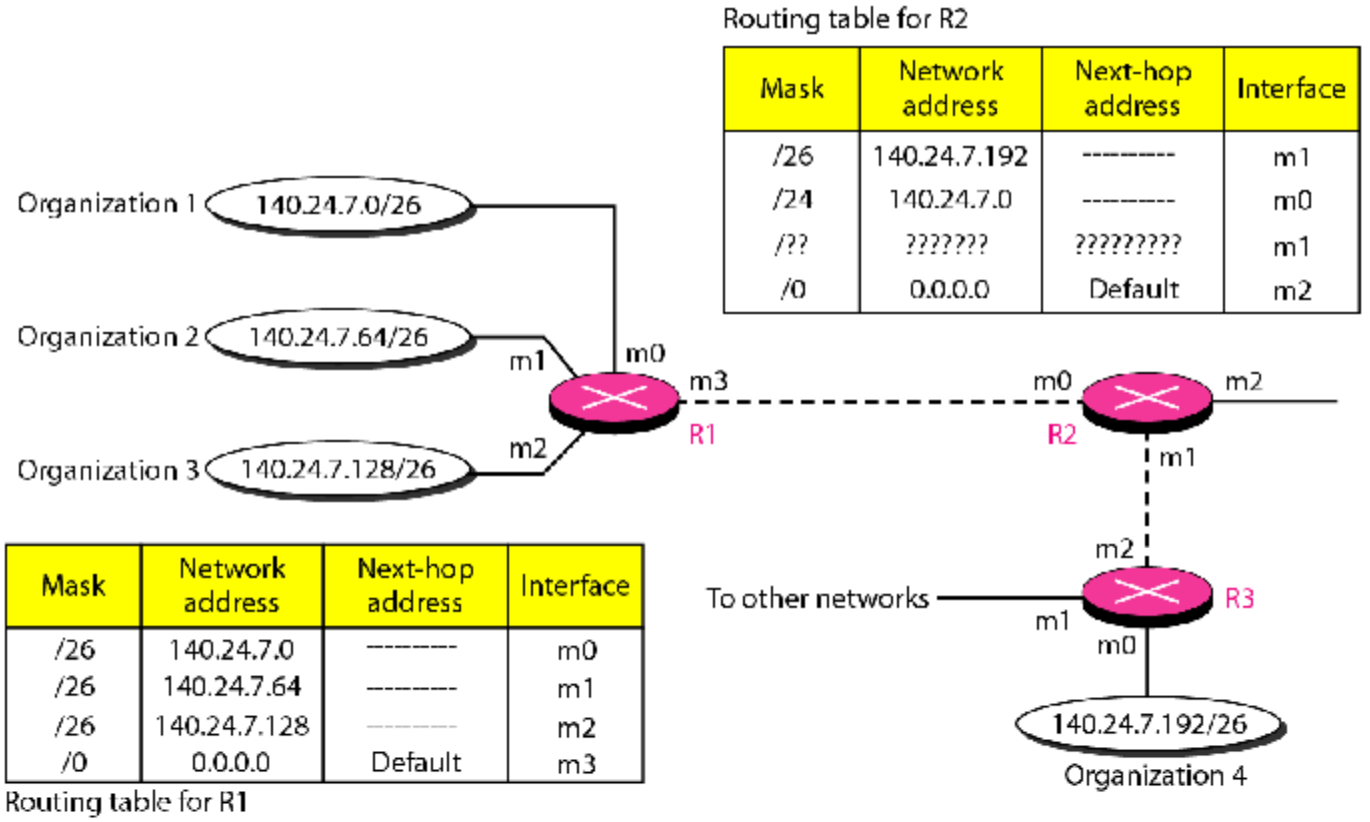
CIDR路由表查找问题

- 使用CIDR时，路由表中的每个条目由“网络前缀”和“下一跳地址”（和/或接口）组成，在查找路由表时可能会得到不止一个匹配结果，这时应该选择哪一条路由呢？
- 选择具有最长网络前缀（即最长掩码）的路由：因为网络前缀越长，其地址块就越小，因而路由就越具体；
- 最长前缀匹配（longest-prefix matching）又称为最长匹配或最佳匹配

最长掩码匹配（书上内容）

- 这个原则指明在路由表中掩码存放是按最长的到最短的次序；
- 言之，如果有三个掩码/27、/26和/24，则/27必定是第一个项目，而/24必定是最后项目；
- 用这一原则，如果组织机构4与另外三个组织机构分离，情形会是怎样？如图22.8所示；
- 假定一个分组到达目的地址是140.24.7.200的组织机构4，将路由器R2中第一个掩码作用于它，得到网络地址140.24.7.192. 该分组被正确通过接口m1而到达组织机构4；但是，如果路由表不按最长的前缀存储，作用掩码/24将给出分组到路由器R1的不正确路由选择。

图22.8 最长掩码匹配


Routing table for R2
Mask
Network address
Next-hop address
Interface
/26
140.24.7.192
----
m1
/24
140.24.7.0
----
m0
/??
???????
?????????
m1
/0
0.0.0.0
Default
m2
Organization 1
140.24.7.0/26
Organization 2
140.24.7.64/26
Organization 3
140.24.7.128/26
m0
m1
m2
m3
R1
m0
m2
R2
m1
m2
To other networks
R3
m1
m0
140.24.7.192/26
Organization 4
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
/0
0.0.0.0
Default
m3
Routing table for R1
Mask
Network address
Next-hop address
Interface
/26
140.24.7.192
----
m0
/??
???????
?????????
m1
/0
0.0.0.0
Default
m2
Routing table for R3

最长前缀匹配举例（更通用情况，不考虑掩码存放顺序）

例如：收到的分组的目的地址D = 206.0.68.1
路由表中的条目：206.0.68.0/24　下一跳IP
206.0.0.0/16　下一跳IP

查找路由表中的第1个条目，第1个条目206.0.68.0/24的掩码M有24个连续的1：

	M =	11111111	11111111	11111111	00000000
AND	*D* =	206.	0.	01000100.	1
		206.	0.	01000100.	0

结果等于206.0.68.0，故与路由表第1个条目匹配

例如：收到的分组的目的地址D = 206.0.68.1
路由表中的条目：206.0.68.0/24　下一跳IP
206.0.0.0/16　下一跳IP

查找路由表中的第2个条目，第2个条目206.0.0.0/16的掩码M有16个连续的1：

	M =	11111111	11111111	00000000	00000000
AND	*D* =	206.	0.	01000100.	1
		206.	0.	0.	0

结果等于206.0.0.0，故与路由表第2个条目也匹配

- 两个路由条目都匹配：
  - 206.0.68.0/24　下一跳IP
  - 206.0.0.0/16　下一跳IP
- 选择两个匹配的条目中地址更具体的一个，即选择最长前缀的条目（第一条）

分层路由选择

- 因特网的其余部分不必知道ISP内部的子网划分，对ISP内部客户来说，在世界上的每个路由器中都只有一项，它们都属于同一组；
- 当然，在本地ISP内部，路由器必须识别出子块并路由到目的客户；
- 如果某一客户是一较大的组织机构，那么它也可以通过子网化和划分它的子块为更小的子块（子块的子块）建立另一个层次；
- 在无类路由选择中，只要遵循无类寻址规则，层次的级数是没有限制的。

例22.5

作为分层路由选择，考虑图22.9。区域ISP被授予以地址120.14.64.0开始的16384个地址。区域ISP决定将这个块划分为4个子块，每块4096个地址。其中的3个子块分别指派给3个本地 ISP，而第2子块保留作将来使用。注意：由于原始块的掩码是/18，因此每块的掩码是/20.

第一个本地ISP将已指派给它的子块划分成8个较小的子块，每一个指派给一个小的ISP。每个小的ISP对128个家庭（H001到H128）提供服务，每个家庭使用4个地址。

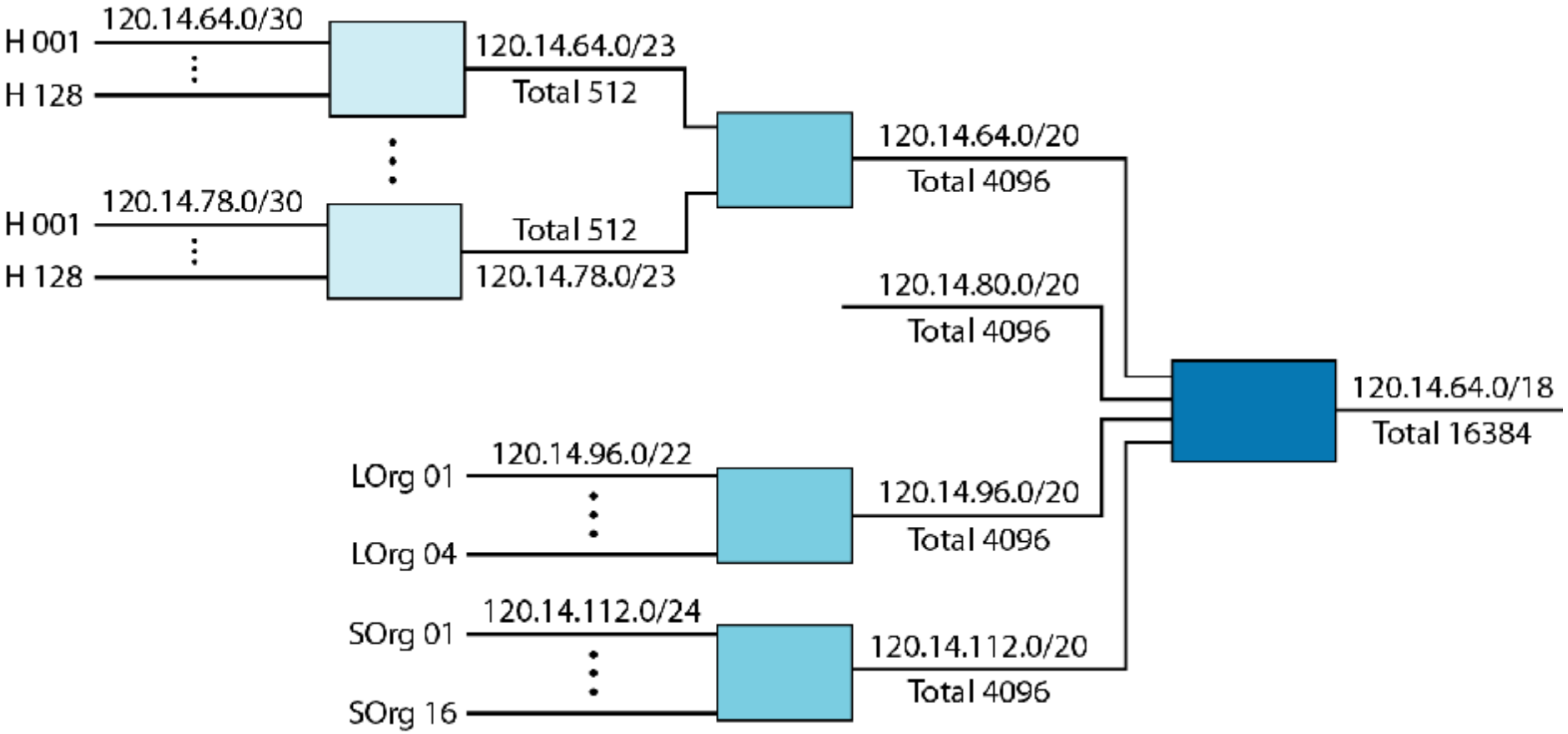
例22.5（cont.）

第二个本地ISP将它的块划分成4个子块，并将地址指派给4个大型的组织机构（LOrg01到LOrg04）。

第三个本地ISP将它的块划分成16个子块，并将每个子块指派给一个小型的组织机构（SOrg01到SOrg16），每一个小型的组织机构有256个地址，掩码是/24。

在这种配置中，存在层次结构。在因特网上的所有路由器将带有目的地址从120.13.64.0到120.14.127.255的分组发送到区域ISP。

图 22.9 ISP层次结构路由选择



路由表

- 主机或路由器要转发IP分组就要有一个路由表，并给每一个目的端设置一个项目；
- 路由表可以静态的，也可以是动态的；
- 静态路由表包含有人工输入的信息，网管人员将每一个目的地址的路由输入到路由表中；路由表生成后，因特网中的变化无法自动在路由表中进行自动更新；静态路由表用在不会经常改动的小型直联网中，或用于故障查找的试验互联网中；
- 动态路由表使用一个动态路由选择协议，如RIP，OSPF或BGP，因而可以周期性地进行更新；当因特网中发生变化时，例如当某个路由器关闭或某条链路中断，动态路由选择协议就自动更新所有路由器的路由表；
- 为了有效地传递IP分组，一个大的互联网如因特网需要动态地更新其路由表

图22.10 路由表中常用的字段

- 掩码、网络地址、下一跳地址、接口；
- 标记：一个通/断开关，它表示或者存在或者不存在；5个标记是：U（工作）、G（网关，目的端是另一个网络，分组必须传递到下一跳路由器以便传递-间接传递）、H（特定主机，指出在地址字段的项目是一个特定主机地址）、D（由于ICMP重定向报文而增加的）和M（由于ICMP重定向报文而修改的）；
- 引用计数：给出在任何时候使用本路由的用户个数；
- 使用：指出经过本路由器发送到相应的目的端的分组数。

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

例22.6

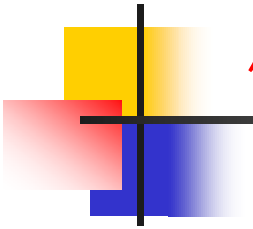
在UNIX或者LINUX操作系统下，可用于查找路由信息和路由表内容的一个命令是netstat。下面显示了一个默认服务器的内容列表。我们在命令中使用了两个选项r和n，其中选项r表示我们对路由表感兴趣，而选项n是查找地址的数字形式。注意：这是一个主机的路由表，而不是路由器的。尽管我们整章都在讨论路由器的路由表，实际上主机也是需要路由表的。

例22.6（cont.）

```
$ netstat -rn
Kernel IP routing table
```

Destination	Gateway	Mask	Flags	Iface
153.18.16.0	0.0.0.0	255.255.240.0	U	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	lo
0.0.0.0	153.18.31.254	0.0.0.0	UG	eth0

此处目的端列定义网络地址。UNIX中的网关与路由器同义，该列定义下一跳地址。值0.0.0.0表示传递是直接的。最后一行中的G表示了通过一个路由器（默认路由器）到达目的端。Iface列定义接口。

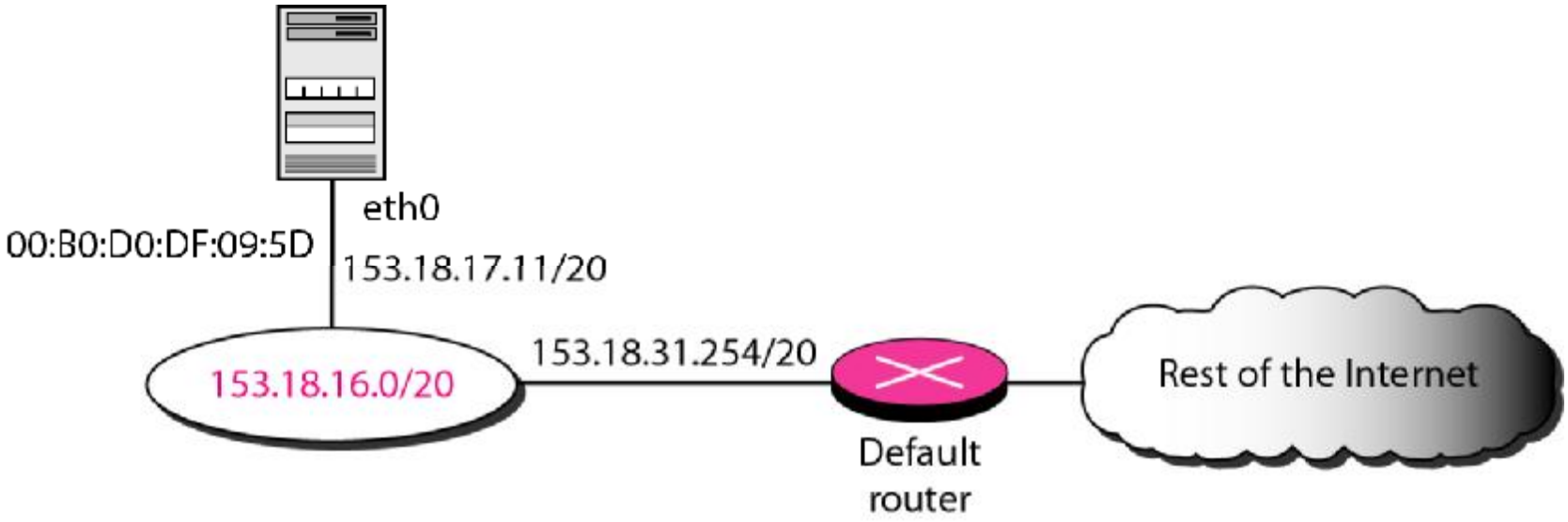


例22.6（cont.2）

通过在给定接口（**eth0**）上使用**ifconfig**命令可以查找到有关服务器**IP**地址和物理地址更多的信息。

```
$ ifconfig eth0
eth0   Link encap:Ethernet  HWaddr 00:B0:D0:DF:09:5D
inet addr:153.18.17.11  Bcast:153.18.31.255  Mask:255.255.240.0
. . .
```

图22.11 例22.6的服务器的配置



22-3 单播路由选择协议

- 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；
- 由于需要有动态路由表，因此产生了多种路由选择协议；
- 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。

优化原则

- 一个路由器通常连接到多个网络，当它接收到分组肘，它应当将分组转发到哪一个网络呢?这个决定是基于最优化原则而做出；
- 给通过网络指定代价（度量，metric），而给每一个网络指定的度量取决于协议的类型；
- RIP简单以跳数作为度量（每一个/跳网络都是1），OSPF允许网络管理员基于所需服务类型（比如吞吐量、延迟等）指定通过网络的代价，即通过一个网络的路由可以有不同的代价；
- 还有一些协议以完全不同的方式定义度量，比如在BGP中，准则就是可由网络管理员设置的策略，策略定义应当选择什么路径。

域内和域间路由选择

- 互联网非常大，仅使用一个路由选择协议无法处理更新所有路由器路由表的任务，为此，需要将互联网划分为多自治系统；
- 自治系统（autonomous system）是一个单一的管理机构管辖下的一组网络和路由器；
- 自治系统内部的路由选择称为域内路由选择（intradomain routing），自治系统之间的路由选择称为域间路由选择（interdomain routing）；
- 每个自治系统可选择一种或多种域内路由选择协议处理自治系统内部的路由选择；但处理自治系统之间的路由选择，通常只能使用一种域间路由选择协议

图22.12 自治系统

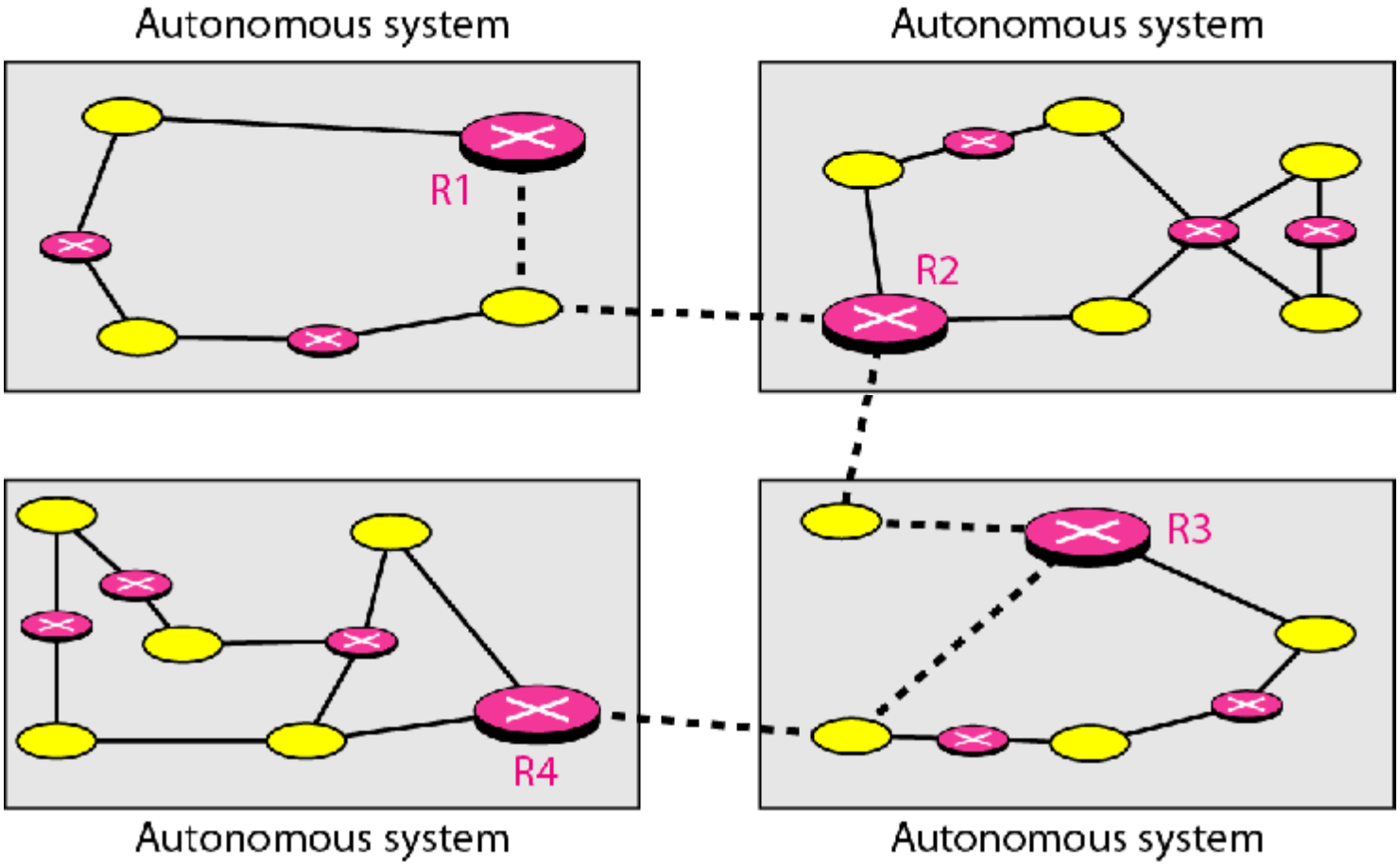
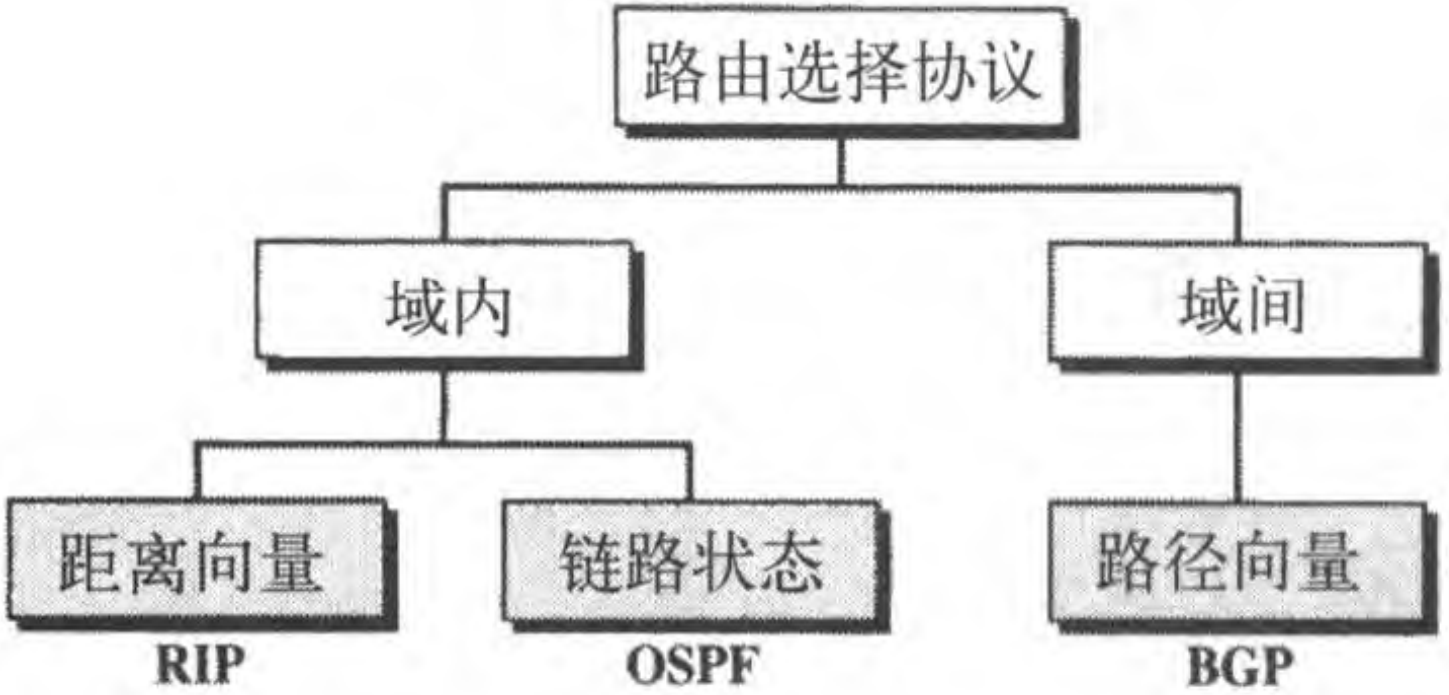

Autonomous system
Autonomous system
R1
R2
R3
R4
Autonomous system
Autonomous system

图22.13 流行的路由选择协议



距离向量路由选择

- 任何两个节点之间最低代价的路由是最小距离的路径；
- 每个节点都保留一张到其他每个节点的最小向量距离（表）；
- 每个节点用这张表中所表示的路由中的下一个节点（下一跳路由选择）指导分组流向目的节点。

图22.14 距离向量路由选择表

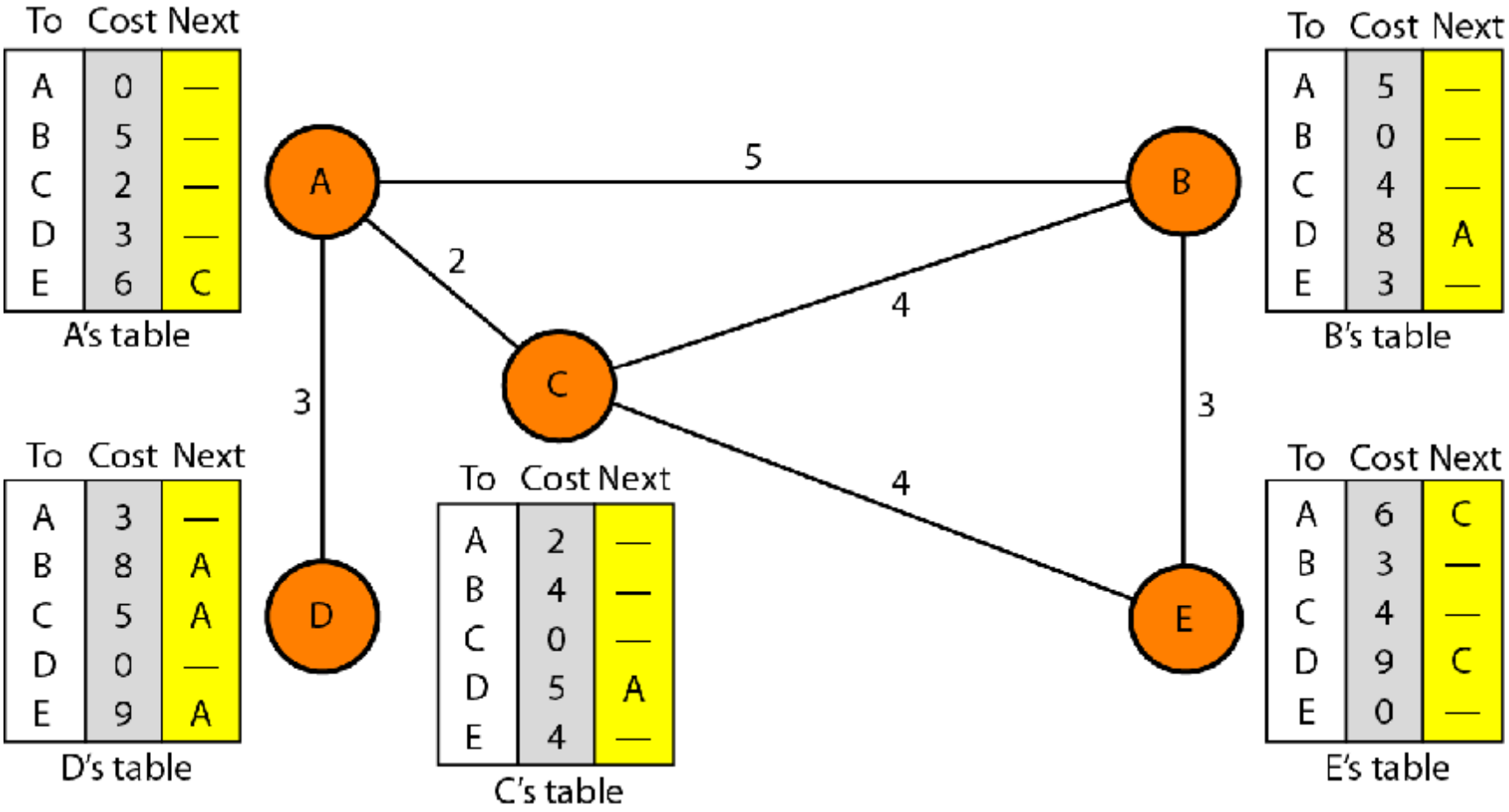
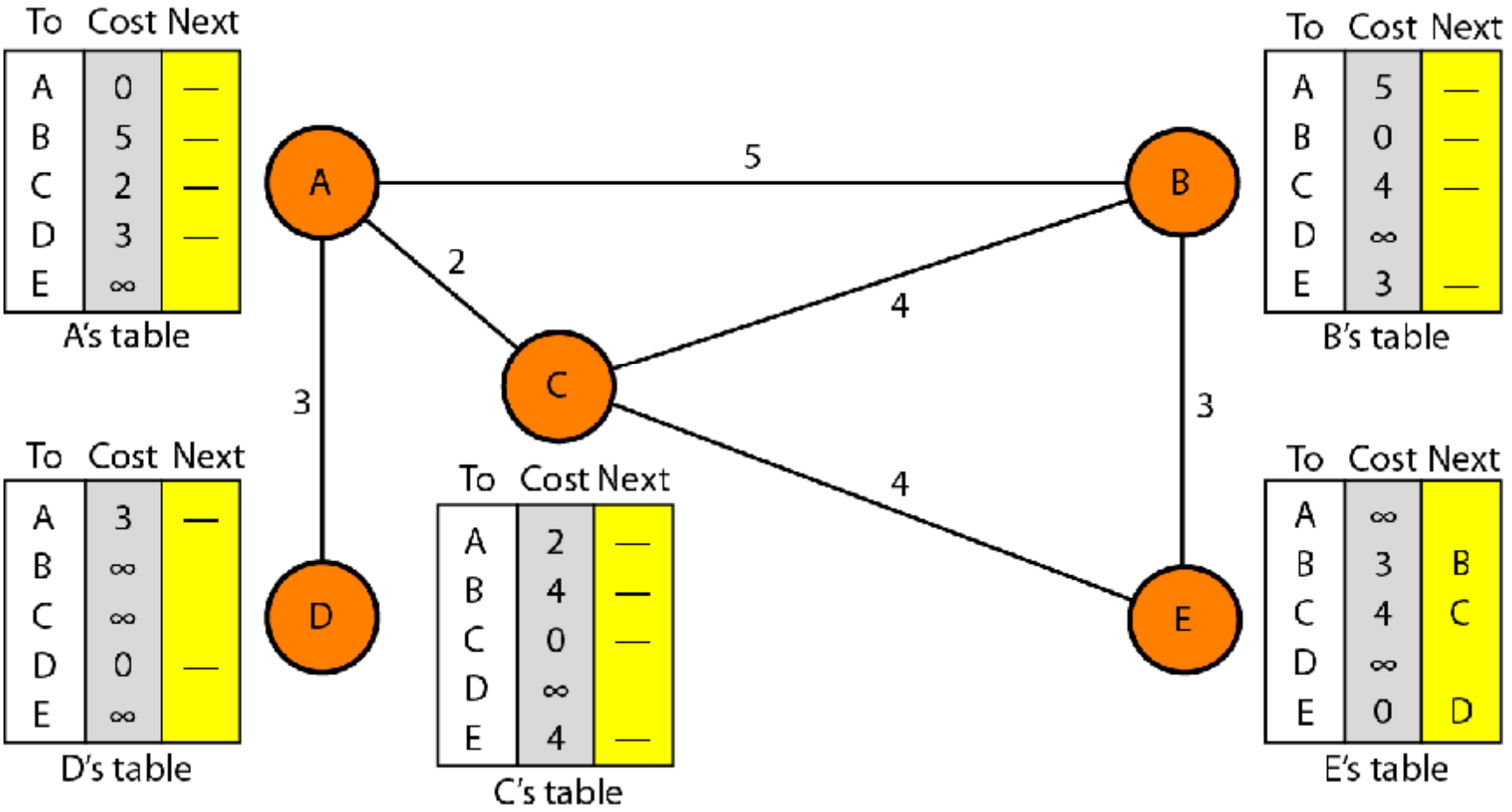


图22.15 距离向量路由选择的初始表（只有邻站的距离）



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它的路由表（只需要共享前两列信息）。

更新算法

p 当一个节点从它的邻站接收到二列的表时，它需要更新它的路由表，三个步骤：

Ø1.接收节点把表的第二列中的每一个值加上它与发送节点之间代价的值；

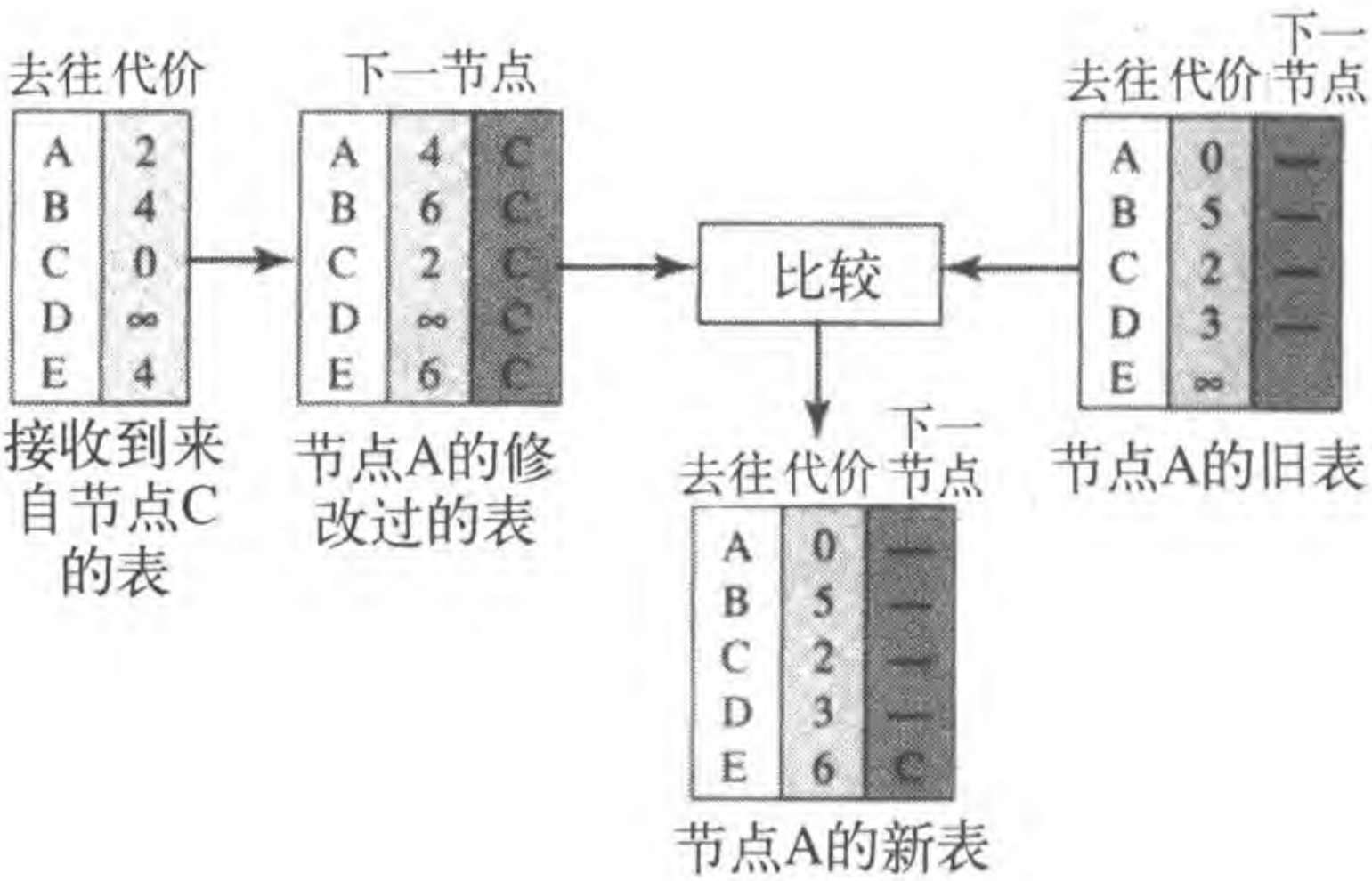
Ø2.如果接收节点使用来自任一行的信息，接收节点需要把发送节点名加入作为第三列，也就是发送节点作为路由的下一个节点；

Ø3. 接收节点将修改过的接收到的表与它的旧表的相应行进行逐行比较：

u a . 如果下一个节点项目不同，则接收节点选取具有最小代价的行；如果最小代价相同，则保持旧的；

u b . 如果下一个节点项目相同，则接收节点选取新行（距离信息更新了）。

图22.16 距离向量更新



何时共享

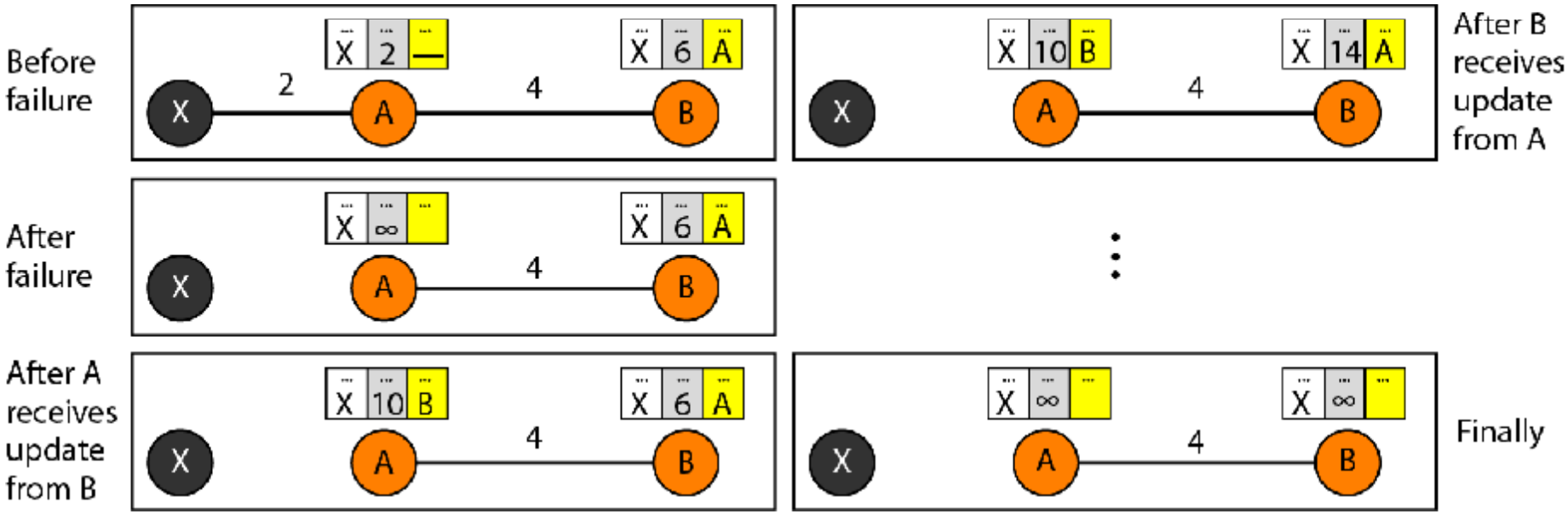
- 周期更新：通常每隔30秒，节点发送它的路由表一次，定期更新，其更新时间依赖于所用的距离向量路由选择的协议；
- 触发更新：节点在它的路由表有变化时，向它的邻站节点发送它的二列路由表，变化是由下列的原因引起：
  - 1.节点接收到邻站的表，引起它自己表的更新；
  - 2. 节点检测到邻站链路有故障。

图22.17 两个节点不稳定性

p距离向量路由选择的一个问题是不稳定性；

p开始时，节点A与B都知道如何到达X，但突然A与X断开，A改变路由表，若A立即发送它的表给B，一切正常；但若在B接收A的路由表前，B已发送它的路由表，该系统变成不稳定的；

pA认为通过B有路径到达X，而B认为通过A有路径到达X，分组往返循环

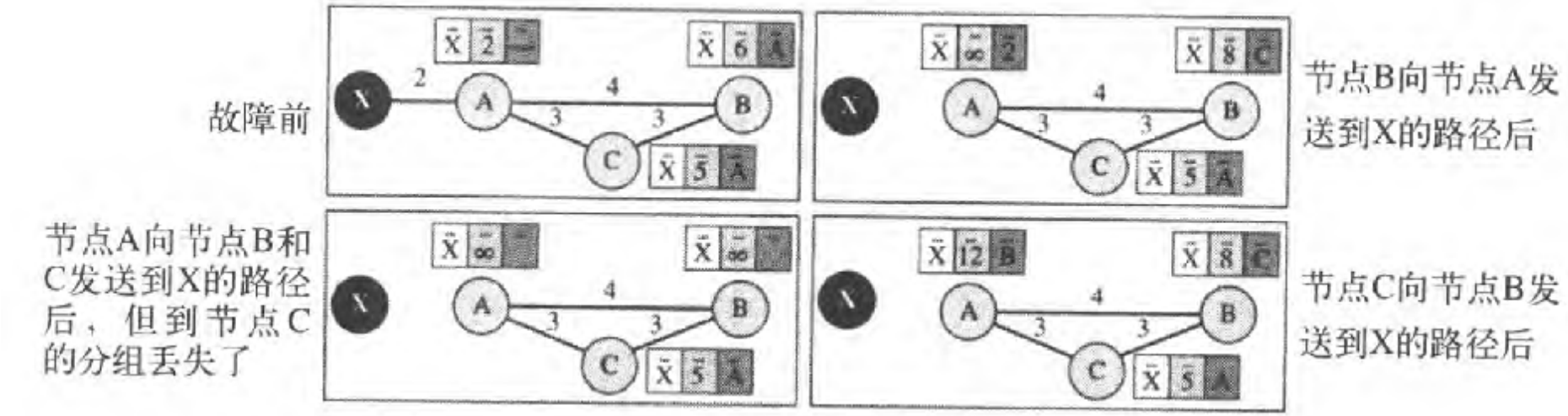


解决两个节点不稳定性的办法

- 办法一是将一个较小的数值定义为无穷大，例如100；事实上，大多数距离向量路由协议的实现都定义两个相邻的节点距离为1，并认为16为无穷大，即距离向量路由协议不能用于大的系统，网络的规模在每个方向不能超过15次跳；
- 另一办法是分割范围：不是通过每一个接口发送整个表，而是每一个节点通过每一个接口仅发送它的表的一部分，如果节点B认为通过A到达X是最佳路径，则节点B不需要向A通知这一部分信息，因为这一信息来自A（A已知道）；
- 距离向量路由协议经常使用定时器，如果没有关于一条路径的新信息，那么节点就在它的表中删除该路径；分割范围策略可与毒性逆转（poison reverse）策略结合起来，节点B依旧可（向A）通知X的值，但如果信息源是节点A，则用无穷大表示距离作为警告“不要使用这个值，我知道的这个路径来自你”

图22.18 三个节点不稳定性

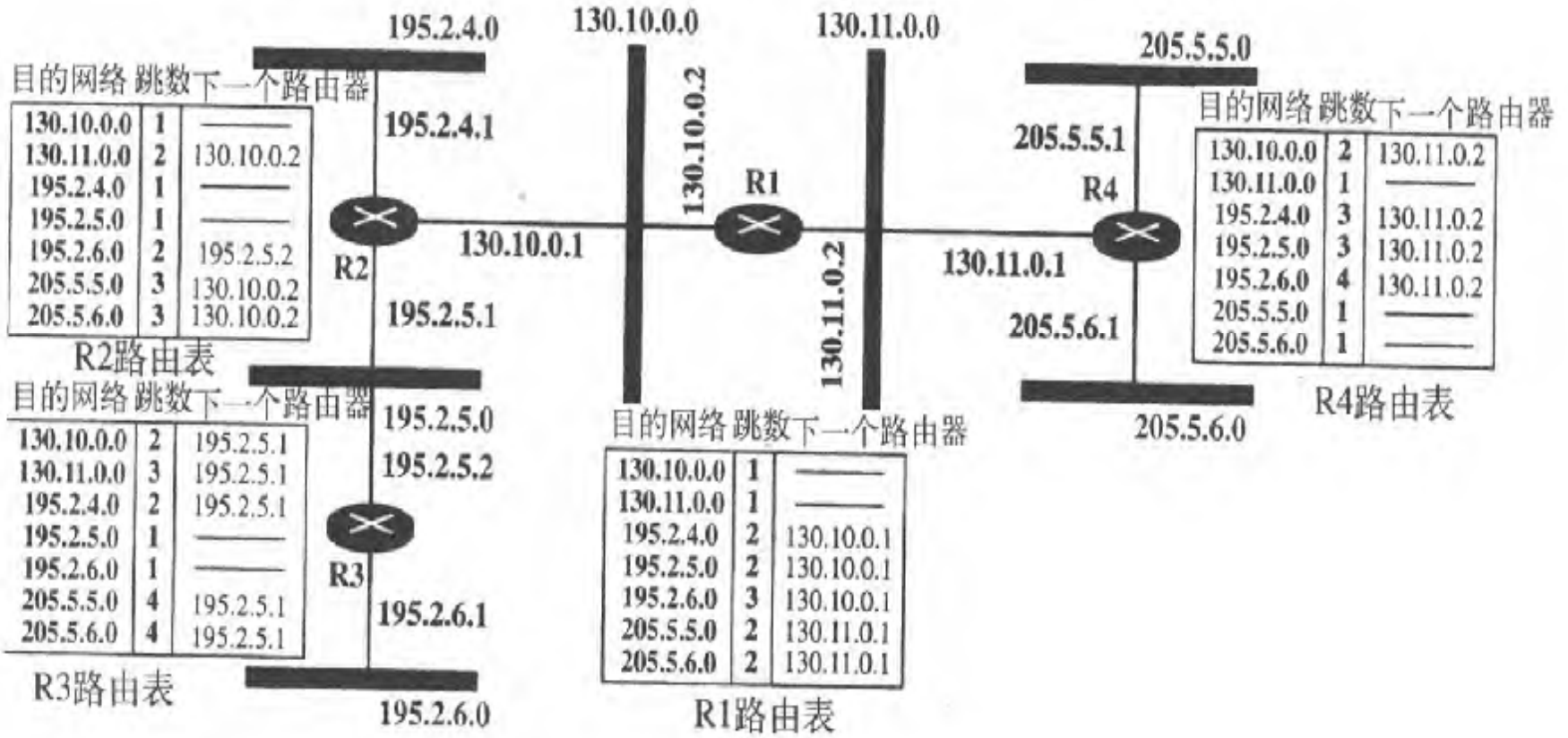
- 分割范围与毒性逆转策略解决不了三个节点不稳定性问题；
- 发现X不可达后，A发送一个分组给节点B和C，B立即更新它的表，可是到C的分组在网络中丢失，因此永远到达不了C；
- C仍认为有一条通过A到达X的距离为5的路径，它向B发送它的路由表，其中包含有到X路径，B更新它的路由表并向节点A通知这条路径，A糊涂了并更新它的路由表，指出A通过B可达到X，其代价为12；当每个节点的代价到达无穷大时，循环停止。



RIP（Routing Information Protocol，路由选择信息协议）

- RIP是域内路由选择协议，是基于距离向量路由选择的一个非常简单的协议；
- RIP基于下列考虑直接实现距离向量路由选择：
  - Ø1. 在一个自治系统中，包括了路由器和网络（链路），路由器有路由表，而网络没有路由表；
  - Ø2. 路由表中的目的端这一列是网络，这表示它的第一列定义了目的网络地址；
  - Ø3. RIP所用的度量很简单，距离定义为到达目的端的链路（网络）个数，因此 RIP的度量称为跳数（hop count）；
  - Ø4. 16就定义为无穷大，就是说在使用RIP 的任何自治系统中，任何路径不能大于15跳；
  - Ø5. 下一个节点这一列定义为被发送分组所要到达的目的路由器的地址。

图22.19 使用RIP的区域例子



RIP协议的层次

- RIP协议使用传输层的用户数据报协议UDP进行传送，它使用UDP的520端口；
- 因此RIP协议位于应用层。

链路状态路由选择

- 在链路状态路由选择中，区域中的每一个节点拥有该区域的全部拓扑结构（所有节点和链路的列表，它们如何连接包含类型、代价也就是度量和链路的接通或断开的情形）；
- 根据拓扑结构，一个节点可用Dijkstra算法建立路由表

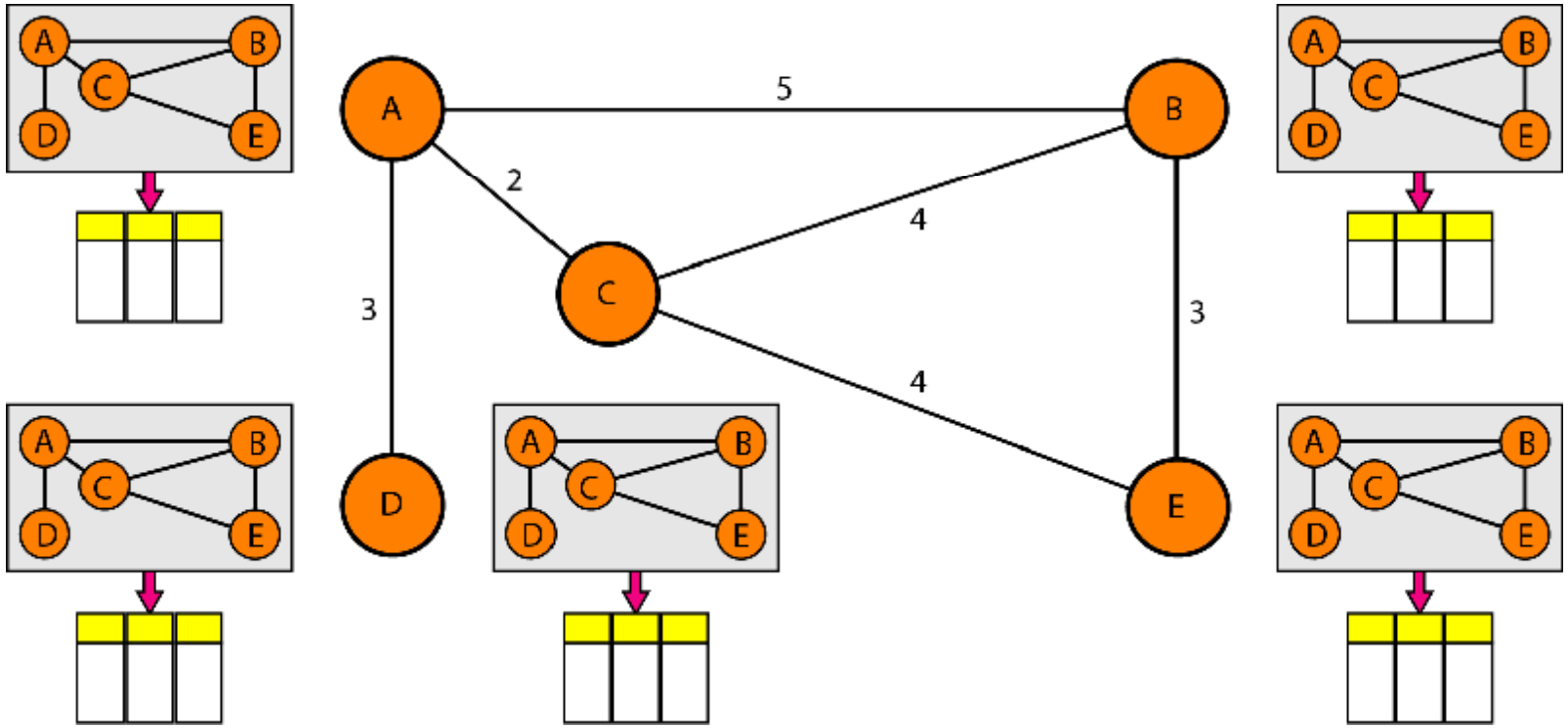
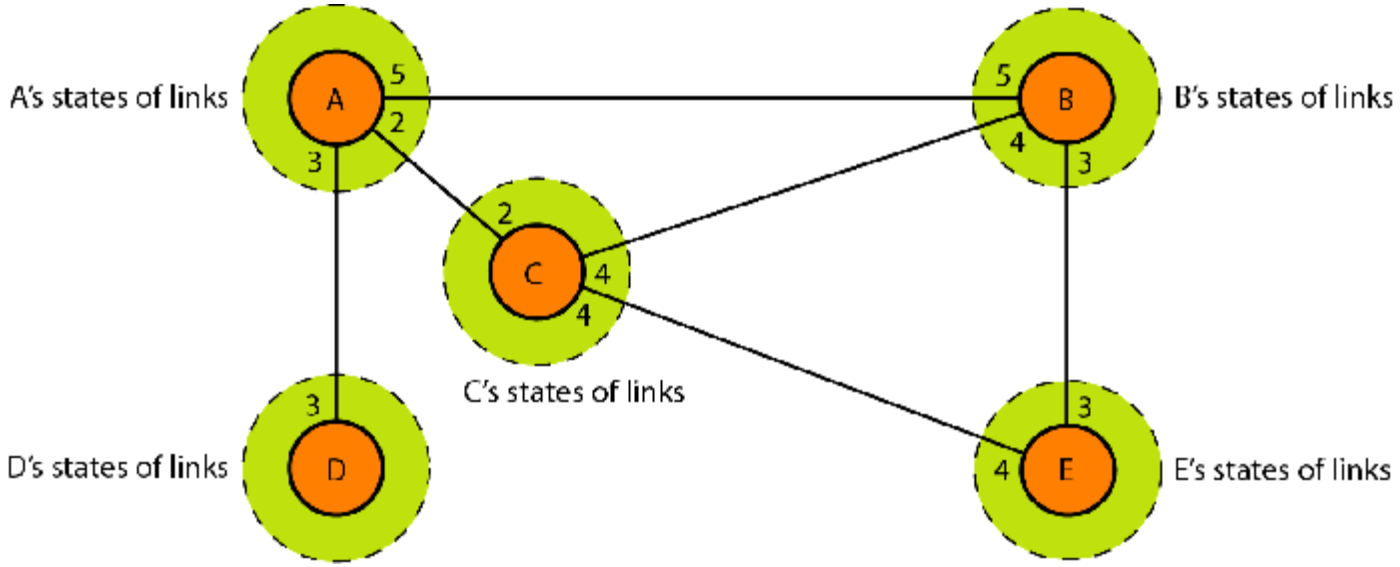


图22.21 链路状态知识

p拓扑必须动态表示每一节点最近的状态和链路，若网络中任一点改变（例如某一链路断开连接），则更新每个节点的拓扑；

p没有一个节点开始时或网络上的某处发生变化后就能知道该拓扑，但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），换言之，整个拓扑可由每一个节点的部分知识复合而成。



建立路由表

p在链路状态路由选择中，为了保证每个节点到达其余节点都具有最小成本的路由表，需要做4件事：

- Ø1.为每个节点建立称为链路状态分组（LSP）的链路状态；
- Ø2.用一种有效而可靠的方法向其他每个路由器扩散LSP，这称为洪泛；
- Ø3.为每个节点构成一个最短路径树，
- Ø4.基于最短路径树计算路由表。

链路状态分组（LSP）的生成

- 链路状态分组可携带大量信息，但假定它携带最小的数据量：节点的标识、链路的清单、序列号和寿命；
- 前两个是生成拓扑所需要的，序列号用于洪泛和区别新与旧的LSP，寿命防止旧的LSP在区域中长期保留；
- 在两种情况下产生LSP：
  - 1.当区域的拓扑有变化时：触发LSP扩散是快速通知区域中任一节点更新它的拓扑的主要方法；
  - 2.基于周期性产生：周期比距离向量路由选择的周期长，它保证旧的信息从区域中除去，定时器设定的范围通常是60分或2小时，以防止洪泛在网络上产生太多的通信量。

- LSP洪泛法：一个节点准备好LSP后，必须向其余所有的节点扩散，不仅只是邻站；
- 接收到所有的LSP后，每个节点就有了完整的拓扑副本，但是对到其余每个节点找出最短距离还是没有完成，需要有最短路径树；
- 最短路径树是根（源节点）与其他节点之间路径最短的一个树，对每个节点，需要有该节点为根的最短路径树；
- Dijkstra算法能从图创建一个最短路径树。

Dijkstra算法

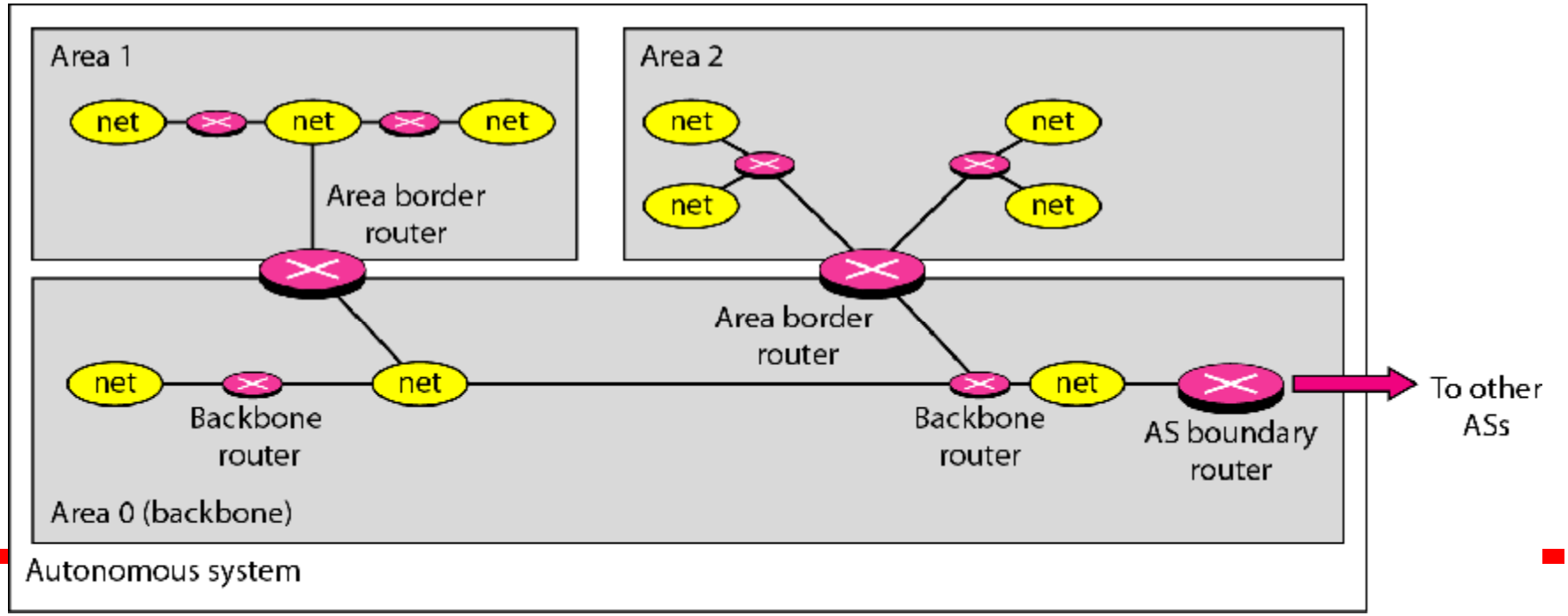
- 参见“补充内容-ch08-2-交换网络中的路由选择”。

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

- OSPF是基于链路状态路由选择的一个域内路由选择协议；
- 为了有效和及时地处理路由选择，OSPF将自治系统划分为一些区域；
- 一个区域（area）是包含在自治系统中的一些网络、主机和路由器的集合，自治系统可划分为多个不同的区域，在一个区域里所有网络必须是互相连接的；
- 一个区域内的路由器使用洪泛法传送路由选择信息，在一个区域的边界，区域边界路由器（area border router）将本区域的信息概括起来发送给其他区域；
- 自治系统中有一个特殊区域称为主干，自治系统中的所有区域必须连接到主干上；换言之，主干相当于主区域，其他区域相当于从区域，但并不表示在各区域内的路由器不能相互连接；
- 在主干中的路由器称为主干路由器（backbone router）；

OSPF（cont.）

- 一个主干路由器也可以同时是一个区域的边界路由器；
- 如果由于某些问题，主干和区域间的连通性被破坏了，那么网络管理员就必须在路由器之间建立一条虚链路（**virtual link**）（使用一条更长的路径），以保持作为主区域的主干的各种功能的连续性；
- 每一个区域有一个区域标识，主干的区域标识是**0**



OSPF（cont.2）

- OSPF协议允许网络管理员给每一条路由指定一个代价，称为度量（metric）；
- 度量可基于服务类型（最小延迟、最大吞吐量等），一个路由器可以有多个路由表，而每一个路由表基于不同服务类型；
- 在OSPF术语中，一个连接称为链路，已定义了4种类型的链路：点对点链路、过渡链路、残桩链路和虚链路

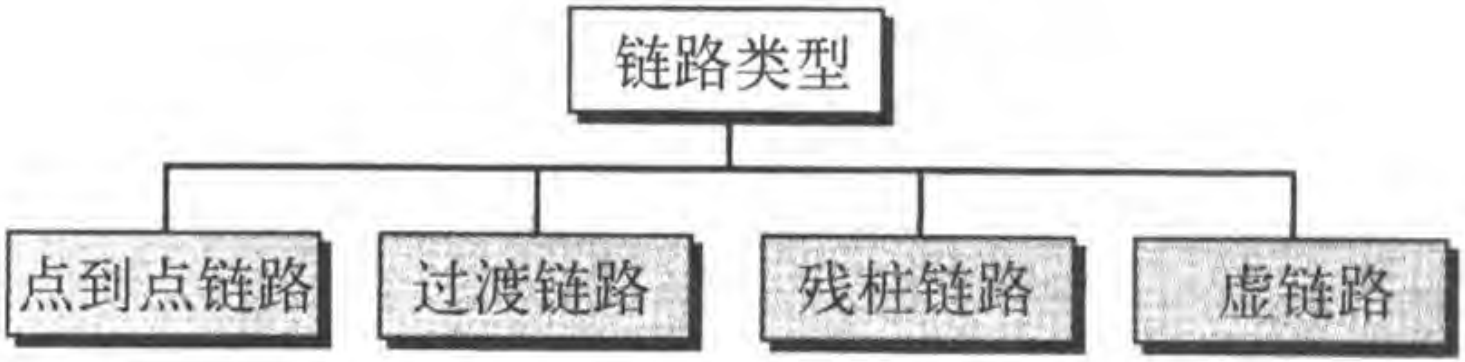


图22.26 点到点链路

p连接两个路由器，而中间没有任何其他的主机或路由器；
p例子：两个路由器用一条电话线（或一条T线）连接起来；
p用图表示时，路由器用节点表示，而链路用一条连接两个节点的双向边来表示；度量表示在两端，各表示每一个方向的度量

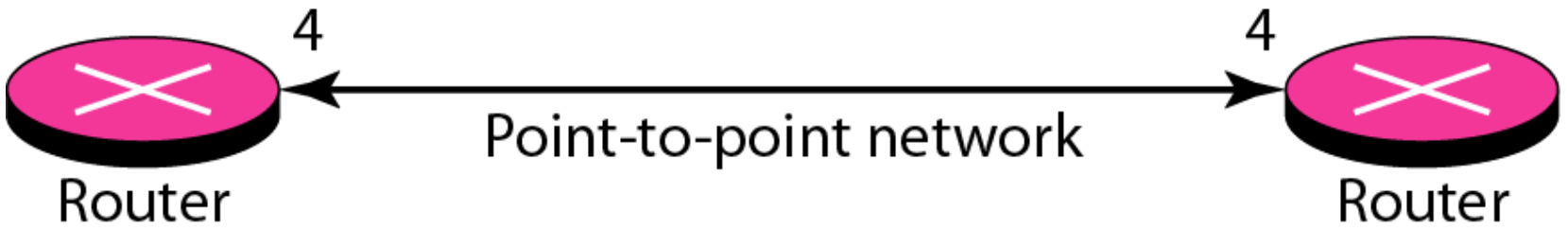


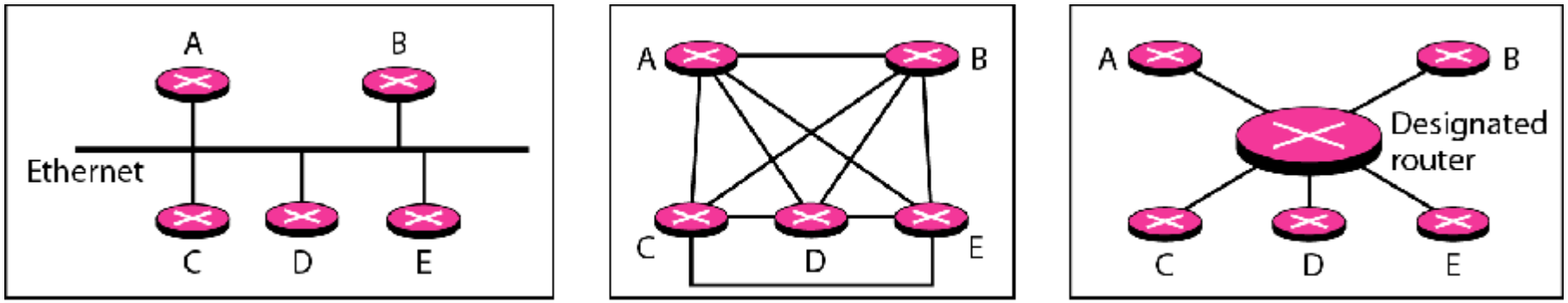
图22.27 过渡链路

p是一种连接多个路由器的网络，数据可以从任何一个路由器进入网络，并从任何一个路由器离开网络；

p所有局域网和某些具有两个或更多的路由器的广域网都属于这种类型；每一个路由器都有好几个邻站；

p这种链路既低效又不实际，低效是因为每一个路由器需要通知其他4个邻站的路由器，不实际是因为在每一对路由器之间没有一个单独的网络（链路），只有一个网络用做交叉路口；

p将该网络用一个节点表示，用指定路由器（双重作用）代表该网络，每一个路由器都只有一个邻站，即指定路由器



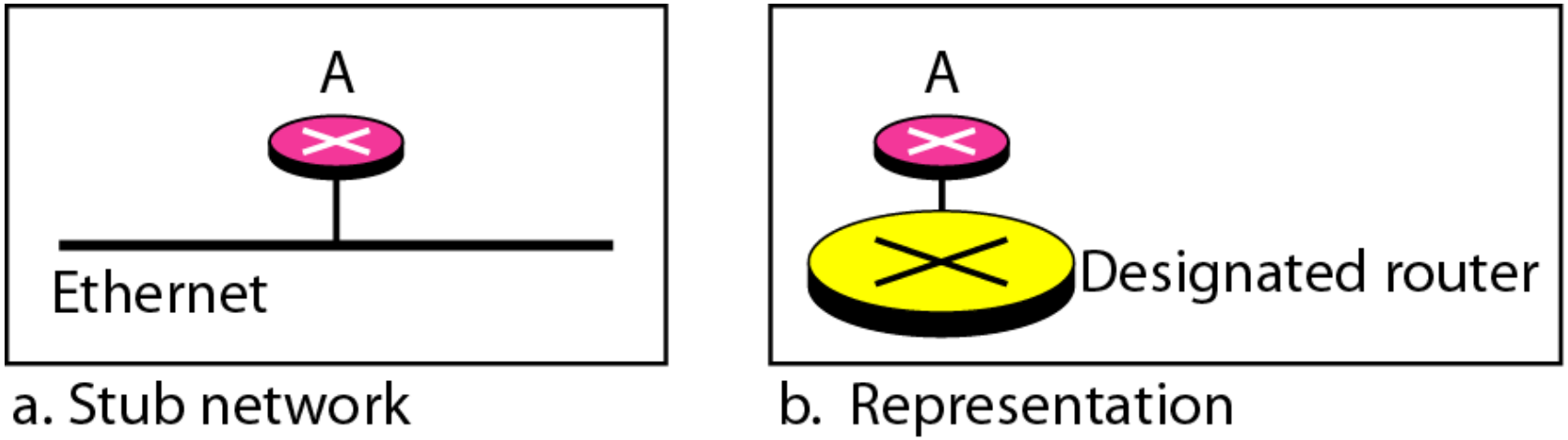
a. Transient network　b. Unrealistic representation　c. Realistic representation

残桩链路和虚链路

p残桩链路是只连接到一个路由器的网络，数据分组通过这个单一的路由器进入网络，而离开网络也是通过这个路由器；

p残桩链路是过渡网络的一个特例，可以将路由器表示为一个节点，而用指定路由器表示这个网络；

p当两个路由器间的链路断开时，网络管理员就在它们间使用一条更长的路径，可能经过好几个路由器，创建一条虚链路

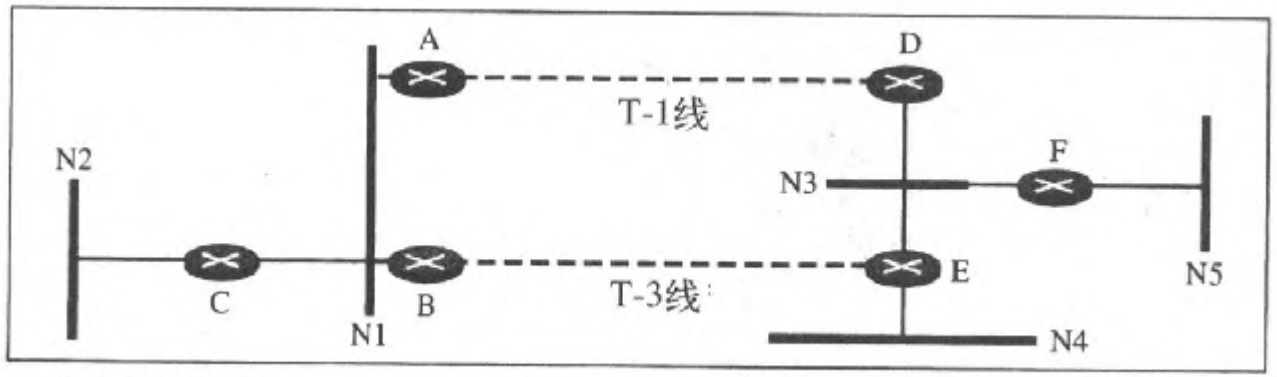


a. Stub network

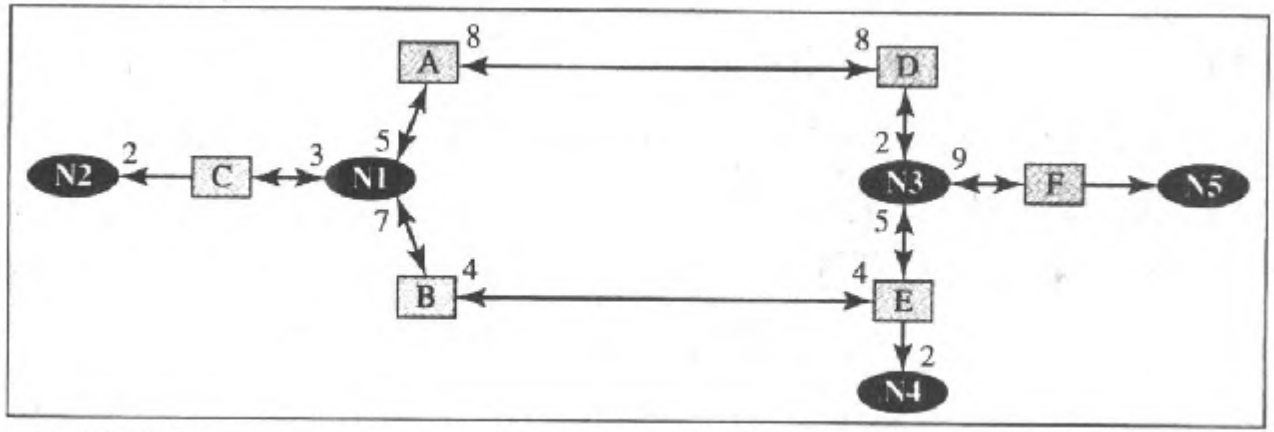
b. Representation

图22.29 一个自治系统的例子及其OSPF 图形表示

- p表示具有7个网络和6个路由器的一个小型自治系统；
- p用正方形节点表示路由器，用椭圆节点表示网络（用指定路由器表示的网络），但OSPF将两者都看做节点；
- p共有3个残桩网络。



a) 自治系统例子



b) 图形表示

OSPF直接用IP数据报传送

- OSPF位于网络层（协议号：89），OSPF不用UDP而是直接用IP数据报传送；
- OSPF构成的数据报很短，这样做可减少路由信息的通信量；
- 数据报很短的另一好处是不会发生分片，而分片传送的数据报只要丢失一个，就无法组装成原来的数据报，整个数据报就必须重传。

路径向量路由选择

- 距离向量路由选择和链路状态路由选择都是域内路由选择协议，用于自治系统内部，而不是自治系统之间；
- 当操作区域变大时，这两个路由选择协议变得很难处理；
- 如果在操作区域中存在多个跳数，距离向量路由选择协议不稳定，而链路状态路由选择协议需要巨大的资源来计算路由表，洪泛也会产生严重的通信拥塞；
- 因此，需要第三个路由选择协议，称为路径向量路由选择（**path vector routing**）；

路径向量路由选择（cont.）

- 在路径向量路由选择中，假定每个自治系统有一个节点的行为代表了整个自治系统（也可以多个），该节点称为代言节点（speaker node）；
- 一个自治系统的代言节点生成一个路由表并通知相邻自治系统中的代言节点；
- 除了每个自治系统仅有一个代言节点可以彼此通信外，它与距离向量路由选择的思想相同；但通知的内容不同，一个代言节点通知的是在它的自治系统或其他自治系统中的路径，而不是节点跳数的度量。

图22.30 路径向量路由选择的初始路由表

- 在开始时，每个代言节点仅能知道它的自治系统内部节点的可达性；
- 自治系统的一个代言节点与它邻站的代言节点共享它的路由表

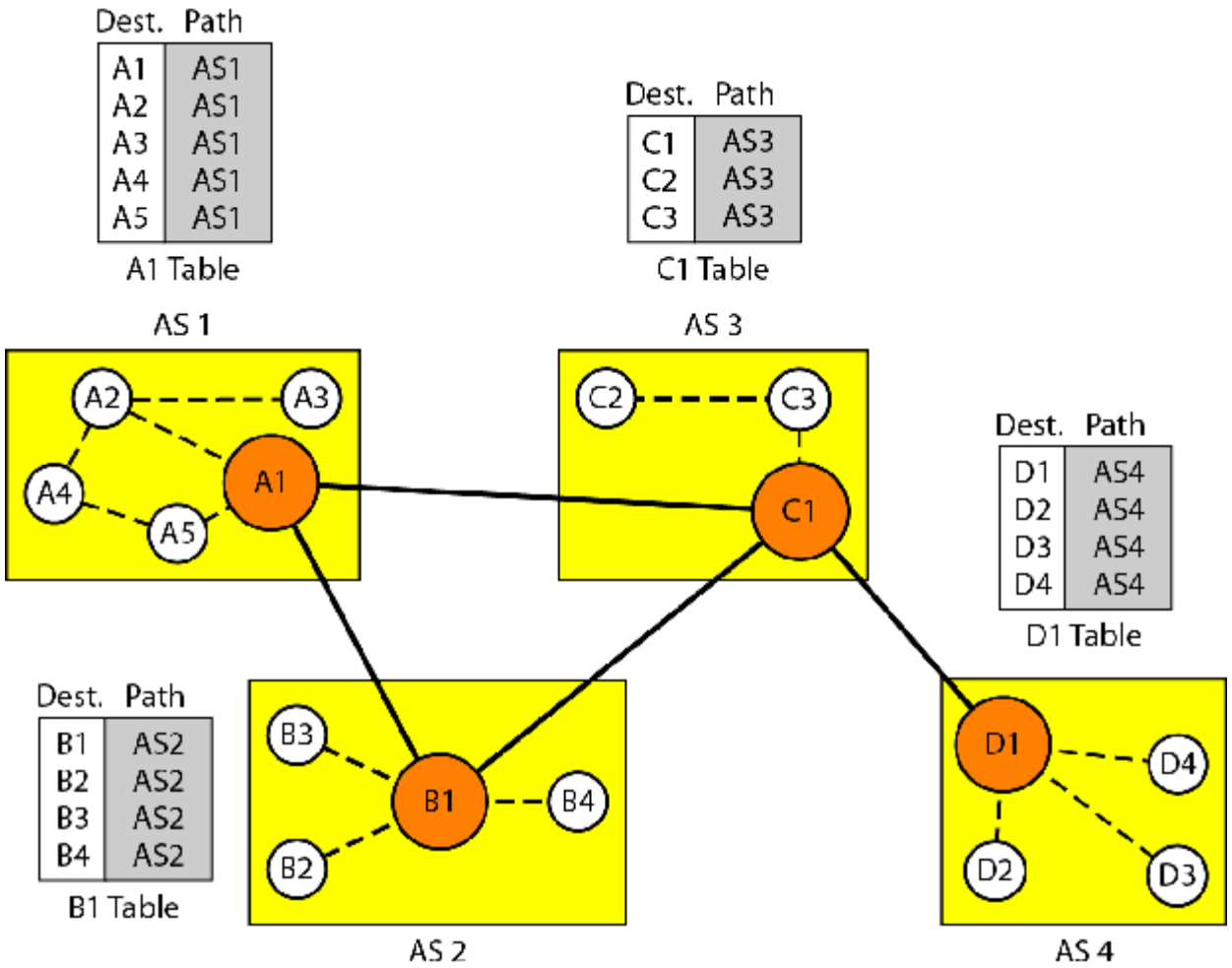
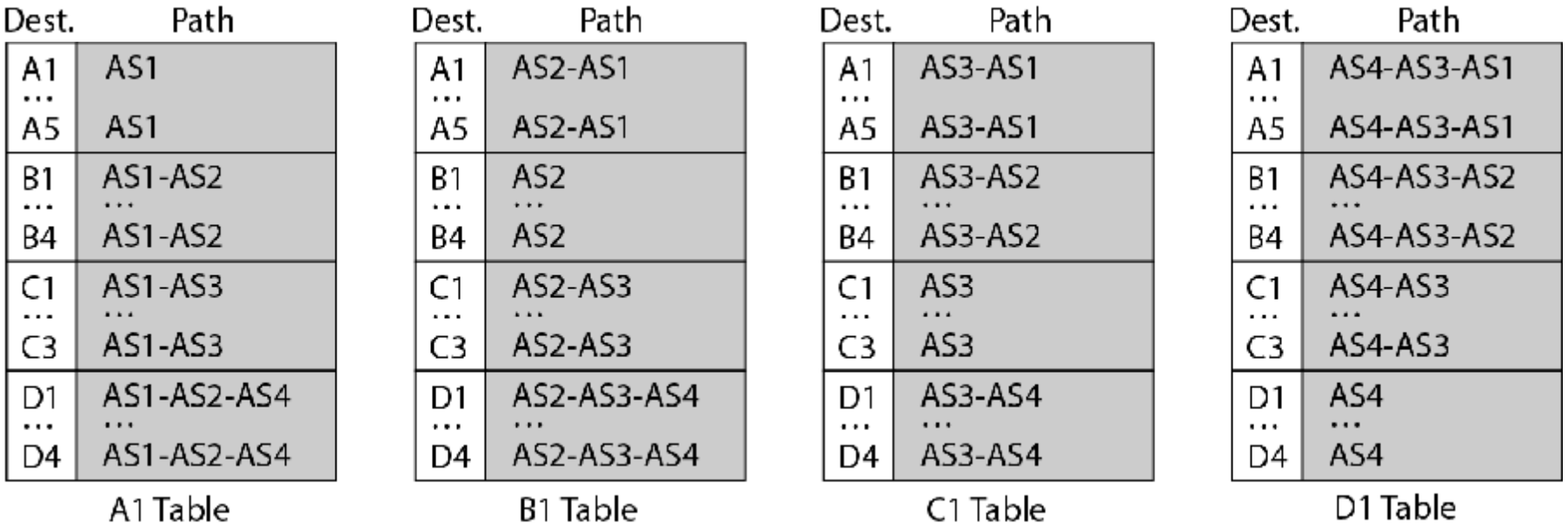


图22.31 三个独立系统的稳定表

- 当一个代言节点从一个邻站接收到一个两列的路由表时，它更新它自己的路由表，更新内容包括增加不在表中的节点以及其自治系统与发送方的自治系统之间的路径；
- 此后，每一个代言节点都有一个如何到达其他自治系统各个节点的路由表。



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

BGP（Border Gate Protocol，边界网关协议）

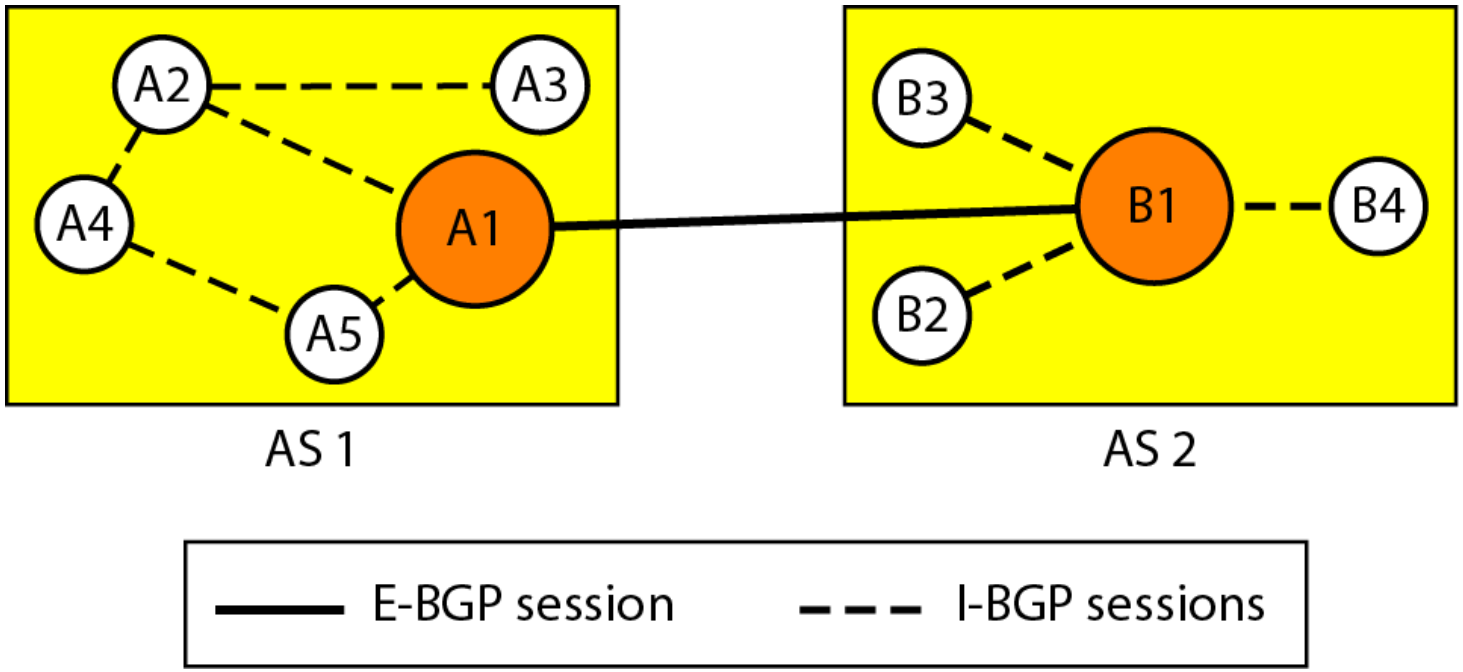
- 使用路径向量路由选择的域间路由选择协议，已有4个版本；
- 三种自治系统类型：残桩的（与其他自治系统只有一个连接）、多接口的（与其他自治系统有多个连接，但它仍旧还是数据流量的源端或接收器）和转送的（多接口自治系统，但它允许过渡数据流量）；
- 路径属性：熟知的和可选的；熟知属性是每一个BGP路由器必须知道的，而可选属性则不需要被每一个路由器都知道；
- 熟知属性分两类：强制的和自选的；熟知强制属性是在一条路由的描述中必须出现的属性，熟知自选属性是每一个路由器必须知道的，但不一定需要包括在每一个更新报文中；例如熟知强制属性ORIGN定义路由选择信息的源端（RIP 、OSPF等）；
- 可选属性分两类：传递的（若没有实现则传递给下一个路由器）和非传递的（若没有实现就丢弃）

BGP会话

- 使用BGP的两个路由器之间的路由信息的交换产生一次会话；
- 会话是两个BGP系统仅为交换路由选择信息而建立的一次连接；
- 为了可靠性，BGP使用TCP（端口号179）作为其传输层协议；换言之，作为一个应用程序，BGP级的会话是TCP级的一条连接（Q：BGP位于哪一层？）；
- 但是，对于BGP和其他应用程序所做的TCP连接略有不同，对于BGP建立的TCP连接可持续一段较长的时间直到某一不寻常的事件发生，因此 BGP会话有时称为半永久连接。

图22.32 内部和外部 BGP会话

- 两种类型的BGP会话：外部E-BGP会话和内部I-BGP会话；
- E-BGP会话用于属于两个不同自治系统的两个代言节点之间交换信息，而I-BGP会话用于一个自治系统内部的两个路由器之间交换路由信息（从自治系统中的其他路由器收集信息）。



BGP使用的环境

- 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；
- 自治系统之间的路由选择必须考虑有关策略；
- 因此，边界网关协议BGP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。

22-4 多播路由选择协议（了解多播概念）

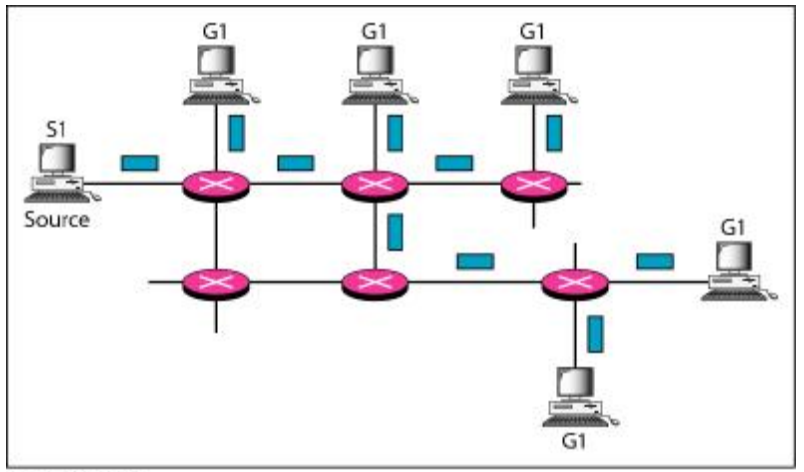
p单播：只有一个源端和一个目的端，IP数据报中的源地址和目的主机（确切的说是网络接口）的单播地址；在单播中，路由器将接收到的分组仅从其端口中的一个转发出去；

p多播：一个源端和一组目的端；在多播中，路由器可能通过它的多个端口将其所接收的分组转发出去；

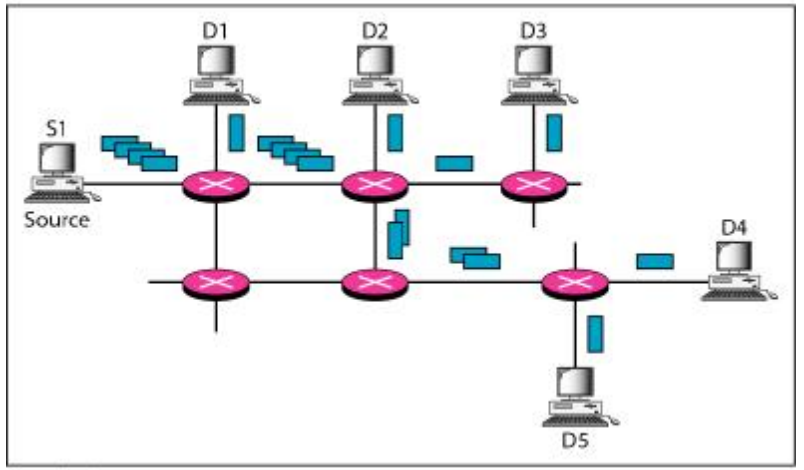
p广播：源端只有一个，而目的端是其余所有的主机，会产生大量的通信量；路由器一般不转发广播报文

图22.35 多播与多个单播

- 多播是从源地址以一个单个分组发出，它被路由器复制；每个分组中的目的地址对所有的副本是相同的，在任何两个路由器之间只有分组的一个副本；
- 在多个单播中，从源端发出多个分组；例如，如果有5个目的地址，源端发送5个分组，每个分组具有不同的单播目的地址。



a. Multicasting



b. Multiple unicasting

多播典型应用

- 分布式数据库；
- 信息发布；
- 新闻传播；
- 电话会议；
- 远程学习等。