

§. 路由器的基本功能及使用

1. 路由器的作用

路由器的作用是在两个不同的网络 (Netid不同) 间转发数据 (因此至少两个接口, 配置分属于不同网段的IP地址, 分别连接不同网段), 转发时仅根据目的主机所连接的网络号来转发而不考虑目的主机号, 这样可以使路由表中的项目数大幅度减少 (减少路由表所占空间, 加快查找速度)

★ 再次说明, 路由器转发时不带MAC地址

2. 路由表的基本工作原理

从某个接口接收IP数据包

=> 查找IP数据包的目的地址, 若是特殊地址/路由器自身地址, 则做处理

=> 如果ttl=0, 则丢弃该包

=> 查找路由表, 找到转发接口, 将该数据包ttl-1, 重新计算校验码后从该接口发出

3. 常见环境的路由设置

★ 静态路由: 人工在每个路由器中根据实际需要设置

★ 动态路由: 由动态路由算法根据邻接路由器发来的数据包进行动态计算

4. 路由器基本配置实验

实验准备: 两台PC, 禁用除有线网卡外的全部网卡, 记录有线网卡的MAC地址(ipconfig /all)

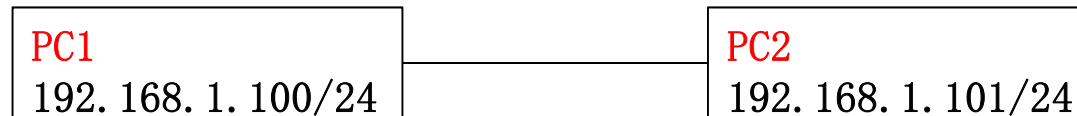
实验1：两台PC，连接在同一交换机上，

PC1 : 192.168.1.100/24，不设网关

PC2 : 192.168.1.101/24，不设网关

同网段

- (1) 两台PC相互是否能ping通？分别提示什么？
- (2) ping包的ttl值是多少？
- (3) 用 `arp -a` 观察对方MAC地址与IP地址的绑定关系



自行实验并观察

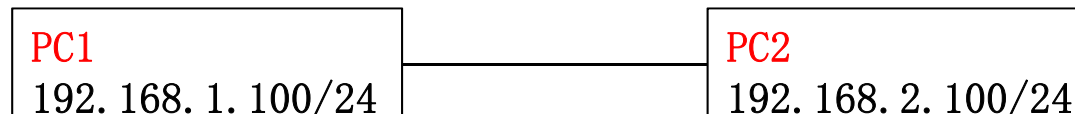
实验2：两台PC，连接在同一交换机上，

PC1 : 192.168.1.100/24，不设网关

PC2 : 192.168.2.100/24，不设网关

不同网段

- (1) 两台PC相互是否能ping通？分别提示什么？
- (2) ping包的ttl值是多少？
- (3) 用 `arp -a` 观察对方MAC地址与IP地址的绑定关系



自行实验并观察

实验3: 两台PC+一台路由器, 连接在同一交换机上,

PC1 : 192.168.1.100/24, 不设网关

PC2 : 192.168.2.100/24, 不设网关

R1-Ethernet0 : 192.168.1.1/24

Ethernet1 : 192.168.2.1/24

(1) PC之间, PC与R1的两个端口是否能通? 分别提示什么?

192.168.1.100 ping 192.168.2.100 : 不通

192.168.1.100 ping 192.168.1.1 : 通

192.168.1.100 ping 192.168.2.1 : 不通

192.168.2.100 ping 192.168.1.1 : 不通

192.168.2.100 ping 192.168.2.1 : 通

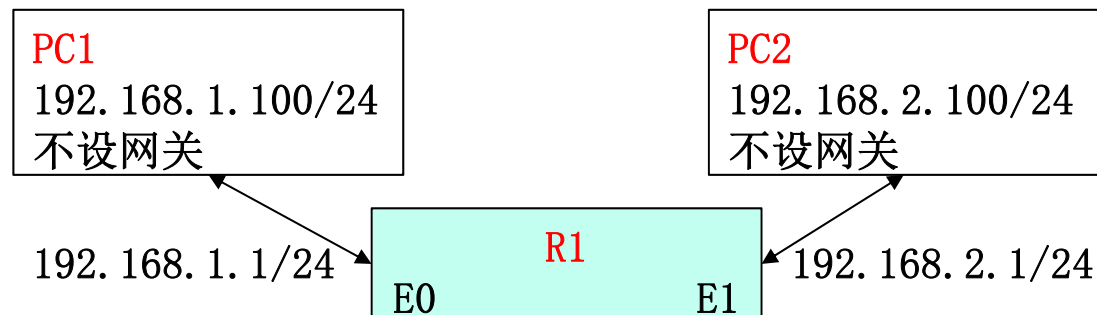
(2) ping包的ttl值是多少?

128

(3) 用 arp -a 观察对方MAC地址与IP地址的绑定关系

192.168.1.100 中只有 192.168.1.1 的MAC

192.168.2.100 中只有 192.168.2.1 的MAC



实验4: 两台PC+一台路由器, 连接在同一交换机上,

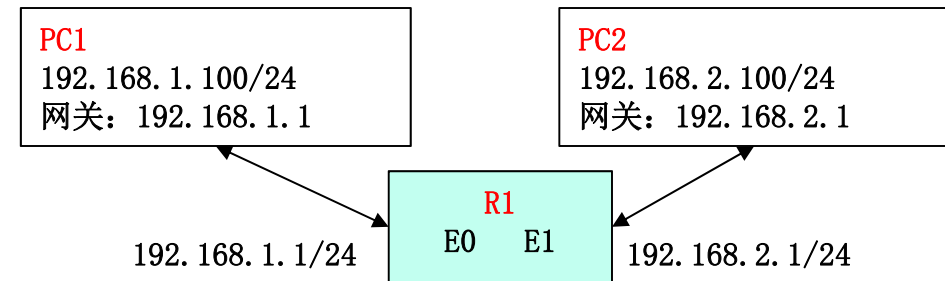
PC1 : 192.168.1.100/24, 网关 192.168.1.1

PC2 : 192.168.2.100/24, 网关 192.168.2.1

R1-Ethernet0 : 192.168.1.1/24

Ethernet1 : 192.168.2.1/24

- (1) PC之间, PC与R1的两个端口是否能通? 分别提示什么?
- (2) ping包的ttl值是多少?
- (3) 用 arp -a 观察对方MAC地址与IP地址的绑定关系
- (4) 记录路由器的配置以及路由器中路由表的显示



```
C:\Windows\system32>ping 192.168.2.100
正在 Ping 192.168.2.100 具有 32 字节的数据:
来自 192.168.2.100 的回复: 字节=32 时间=1ms TTL=127
来自 192.168.2.100 的回复: 字节=32 时间=1ms TTL=127
来自 192.168.2.100 的回复: 字节=32 时间=1ms TTL=127
来自 192.168.2.100 的回复: 字节=32 时间=1ms TTL=127

192.168.2.100 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 1ms, 最长 = 1ms, 平均 = 1ms
```

以太网适配器 本地连接:

PC1的配置

```
连接特定的 DNS 后缀 . . . . . :
描述. . . . . : Intel(R) Ethernet Connection I218-V
物理地址. . . . . : 28-D2-44-B8-CB-DC
DHCP 已启用 . . . . . : 否
自动配置已启用. . . . . : 是
IPv4 地址 . . . . . : 192.168.1.100(首选)
子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . : 192.168.1.1
TCP/IP 上的 NetBIOS . . . . . : 已启用
```

以太网适配器 本地连接:

PC2的配置

```
连接特定的 DNS 后缀 . . . . . :
描述. . . . . : Realtek PCIe GBE Family Controller
物理地址. . . . . : 38-2C-4A-B2-58-5B
DHCP 已启用 . . . . . : 否
自动配置已启用. . . . . : 是
IPv4 地址 . . . . . : 192.168.2.100(首选)
子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . : 192.168.2.1
TCP/IP 上的 NetBIOS . . . . . : 已启用
```

```
ts-6352#show running
```

R1的配置

```
使用了 8192 字节中的 118 个字节, 软件版本 12.06H
!
interface ethernet 0
 ip address 192.168.1.1 255.255.255.0
!
interface ethernet 1
 ip address 192.168.2.1 255.255.255.0
!
end
```

```
ts-6352#show ip route
```

R1的路由表

```
Codes: C - connected, S - static, R - RIP, O - OSPF,
        I - ISIS, B - BGP, > - selected route, * - FIB route

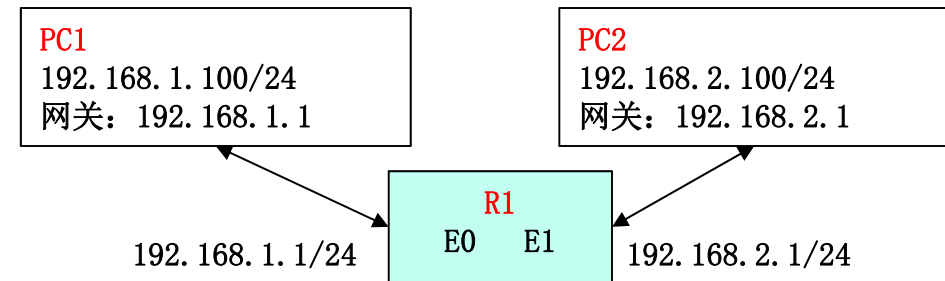
C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
C>* 192.168.1.0/24 is directly connected, Ethernet0
C>* 192.168.2.0/24 is directly connected, Ethernet1
```

实验4: 两台PC+一台路由器, 连接在同一交换机上,

变化1: PC1设置网关、PC2不设网关

变化2: PC1不设网关、PC2设置网关

- (1) PC之间, PC与R1的两个端口是否能通? 分别提示什么?
- (2) ping包的ttl值是多少?
- (3) 用 `arp -a` 观察对方MAC地址与IP地址的绑定关系
- (4) 记录路由器的配置以及路由器中路由表的显示



```
C:\Users\Larry>ping 192.168.2.100
```

```
正在 Ping 192.168.2.100 具有 32 字节的数据: 请求超时。
请求超时。
```

变化1:

PC2 ping PC1
的结果

```
C:\Windows\system32>ping 192.168.2.100
```

```
正在 Ping 192.168.2.100 具有 32 字节的数据:
```

```
PING: 传输失败。General failure.
```

```
PING: 传输失败。General failure.
```

```
PING: 传输失败。General failure.
```

```
PING: 传输失败。General failure.
```

变化2:

PC1 ping PC2的
结果

```
192.168.2.100 的 Ping 统计信息:
```

```
数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

以太网适配器 本地连接:

PC1的配置

```
连接特定的 DNS 后缀 . . . . . :
描述. . . . . : Intel(R) Ethernet Connection I218-V
物理地址. . . . . : 28-D2-44-B8-CB-DC
DHCP 已启用 . . . . . : 否
自动配置已启用. . . . . : 是
IPv4 地址 . . . . . : 192.168.1.100(首选)
子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . :
TCP/IP 上的 NetBIOS . . . . . : 已启用
```

以太网适配器 本地连接:

PC2的配置

```
连接特定的 DNS 后缀 . . . . . :
描述. . . . . : Realtek PCIe GBE Family Controller
物理地址. . . . . : 38-2C-4A-B2-58-5B
DHCP 已启用 . . . . . : 否
自动配置已启用. . . . . : 是
IPv4 地址 . . . . . : 192.168.2.100(首选)
子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . : 192.168.2.1
TCP/IP 上的 NetBIOS . . . . . : 已启用
```

```
ts-6352#show running
```

R1的配置

```
使用了 8192 字节中的 118 个字节, 软件版本 12.06H
!
interface ethernet 0
 ip address 192.168.1.1 255.255.255.0
!
interface ethernet 1
 ip address 192.168.2.1 255.255.255.0
!
end
```

```
ts-6352#show ip route
```

R1的路由表

```
Codes: C - connected, S - static, R - RIP, O - OSPF,
        I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
C>* 192.168.1.0/24 is directly connected, Ethernet0
C>* 192.168.2.0/24 is directly connected, Ethernet1
```

实验5-环境：两台PC+两台路由器，连接在同一交换机上

PC1 : 192.168.1.100/24, 网关 192.168.1.1

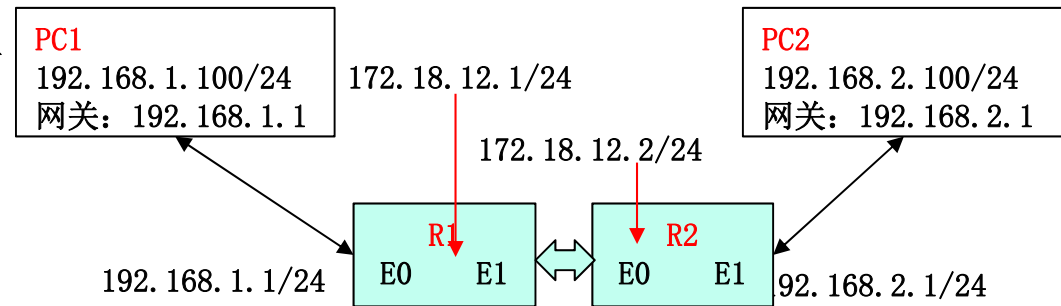
PC2 : 192.168.2.100/24, 网关 192.168.2.1

R1-Ethernet0 : 192.168.1.1/24

Ethernet1 : 172.18.12.1/24

R2-Ethernet0 : 172.18.12.2/24

Ethernet1 : 192.168.2.1/24



实验5-1：R1/R2中均不设置静态路由

- (1) PC之间，PC与R1/R2的4个端口是否能通？分别提示什么？
- (2) ping包的ttl值是多少？
- (3) 记录路由器的配置以及路由器中路由表的显示

```
C:\Windows\system32>ping 192.168.2.100
```

PC1 ping PC2 的结果

正在 Ping 192.168.2.100 具有 32 字节的数据:
 来自 192.168.1.1 的回复: 无法访问目标网。
 来自 192.168.1.1 的回复: 无法访问目标网。
 来自 192.168.1.1 的回复: 无法访问目标网。
 来自 192.168.1.1 的回复: 无法访问目标网。

192.168.2.100 的 Ping 统计信息:
 数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),

ts-6352#show running

R1的配置

```
使用了 8192 字节中的 118 个字节, 软件版本 12.06H
!
interface ethernet 0
 ip address 192.168.1.1 255.255.255.0
!
interface ethernet 1
 ip address 172.18.12.1 255.255.255.0
!
end
```

ts-8#show running

R2的配置

```
使用了 8192 字节中的 118 个字节, 软件版本 12.06H
!
interface ethernet 0
 ip address 172.18.12.2 255.255.255.0
!
interface ethernet 1
 ip address 192.168.2.1 255.255.255.0
!
end
```

ts-6352#show ip route

R1的路由表

```
Codes: C - connected, S - static, R - RIP, O - OSPF,
        I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
C>* 172.18.12.0/24 is directly connected, Ethernet1
C>* 192.168.1.0/24 is directly connected, Ethernet0
```

ts-8#show ip route

R2的路由表

```
Codes: C - connected, S - static, R - RIP, O - OSPF,
        I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
C>* 172.18.12.0/24 is directly connected, Ethernet0
C>* 192.168.2.0/24 is directly connected, Ethernet1
```


实验5-环境：两台PC+两台路由器，连接在同一交换机上

PC1 : 192.168.1.100/24, 网关 192.168.1.1

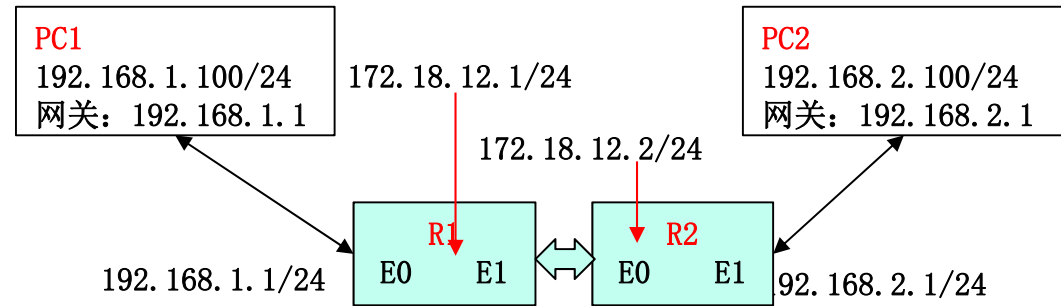
PC2 : 192.168.2.100/24, 网关 192.168.2.1

R1-Ethernet0 : 192.168.1.1/24

Ethernet1 : 172.18.12.1/24

R2-Ethernet0 : 172.18.12.2/24

Ethernet1 : 192.168.2.1/24



实验5-2：两台路由器均设置静态路由

在R1中设置静态路由：

```
ip router 192.168.2.0 255.255.255.0 172.18.12.2
```

在R2中设置静态路由：

```
ip router 192.168.1.0 255.255.255.0 172.18.12.1
```

- (1) PC之间，PC与R1/R2的4个端口是否能通？分别提示什么？
- (2) ping包的ttl值是多少？
- (3) 记录路由器的配置以及路由器中路由表的显示

```
C:\Windows\system32>ping 192.168.2.100
正在 Ping 192.168.2.100 具有 32 字节的数据:
来自 192.168.2.100 的回复: 字节=32 时间=13ms TTL=126
来自 192.168.2.100 的回复: 字节=32 时间=2ms TTL=126
来自 192.168.2.100 的回复: 字节=32 时间=2ms TTL=126
来自 192.168.2.100 的回复: 字节=32 时间=2ms TTL=126

192.168.2.100 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 2ms, 最长 = 13ms, 平均 = 4ms
```

```
ts-6352#show running
```

R1的配置

```
使用了 8192 字节中的 173 个字节, 软件版本 12.06H
!
ip route 192.168.2.0 255.255.255.0 172.18.12.2
!
interface ethernet 0
ip address 192.168.1.1 255.255.255.0
!
interface ethernet 1
ip address 172.18.12.1 255.255.255.0
!
end
```

```
ts-6352#show ip route
```

R1的路由表

```
Codes: C - connected, S - static, R - RIP, O - OSPF,
        I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
C>* 172.18.12.0/24 is directly connected, Ethernet1
C>* 192.168.1.0/24 is directly connected, Ethernet0
S>* 192.168.2.0/24 [1/0] via 172.18.12.2, Ethernet1
```

```
ts-8#show running
```

R2的配置

```
使用了 8192 字节中的 173 个字节, 软件版本 12.06H
!
ip route 192.168.1.0 255.255.255.0 172.18.12.1
!
interface ethernet 0
ip address 172.18.12.2 255.255.255.0
!
interface ethernet 1
ip address 192.168.2.1 255.255.255.0
!
end
```

```
ts-8#show ip route
```

R2的路由表

```
Codes: C - connected, S - static, R - RIP, O - OSPF,
        I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
C>* 172.18.12.0/24 is directly connected, Ethernet0
S>* 192.168.1.0/24 [1/0] via 172.18.12.1, Ethernet0
C>* 192.168.2.0/24 is directly connected, Ethernet1
```

实验5-环境：两台PC+两台路由器，连接在同一交换机上

PC1 : 192.168.1.100/24, 网关 192.168.1.1

PC2 : 192.168.2.100/24, 网关 192.168.2.1

R1-Ethernet0 : 192.168.1.1/24

Ethernet1 : 172.18.12.1/24

R2-Ethernet0 : 172.18.12.2/24

Ethernet1 : 192.168.2.1/24

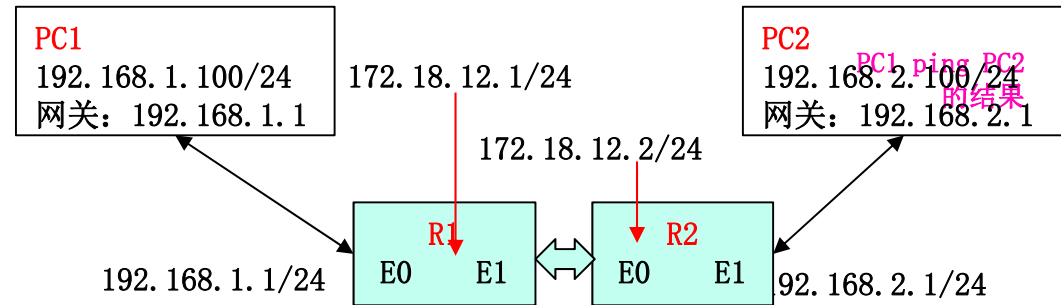
实验5-3：只有一台路由器设置静态路由

在R1中设置静态路由：

ip router 192.168.2.0 255.255.255.0 172.18.12.2

R2中不设置静态路由：

- (1) PC之间，PC与R1/R2的4个端口是否能通？分别提示什么？
- (2) ping包的ttl值是多少？
- (3) 记录路由器的配置以及路由器中路由表的显示



C:\Windows\system32>ping 192.168.2.100

正在 Ping 192.168.2.100 具有 32 字节的数据:

请求超时。

请求超时。

请求超时。

请求超时。

取消R2的静态路由后，为什么ping不通？为什么提示信息是“请求超时”？

192.168.2.100 的 Ping 统计信息:

数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

ts-6352#show running

R1的配置

使用了 8192 字节中的 173 个字节, 软件版本 12.06H

!

ip route 192.168.2.0 255.255.255.0 172.18.12.2

!

interface ethernet 0

ip address 192.168.1.1 255.255.255.0

!

interface ethernet 1

ip address 172.18.12.1 255.255.255.0

!

end

ts-8#show running

R2的配置

使用了 8192 字节中的 173 个字节, 软件版本 12.06H

!

~~ip route 192.168.1.0 255.255.255.0 172.18.12.1~~

!

interface ethernet 0

ip address 172.18.12.2 255.255.255.0

!

interface ethernet 1

ip address 192.168.2.1 255.255.255.0

!

end

ts-6352#show ip route

R1的路由表

Codes: C - connected, S - static, R - RIP, O - OSPF,

I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo

C>* 127.0.0.0/24 is directly connected, lo

C>* 172.18.12.0/24 is directly connected, Ethernet1

C>* 192.168.1.0/24 is directly connected, Ethernet0

S>* 192.168.2.0/24 [1/0] via 172.18.12.2, Ethernet1

ts-8#show ip route

R2的路由表

Codes: C - connected, S - static, R - RIP, O - OSPF,

I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo

C>* 127.0.0.0/24 is directly connected, lo

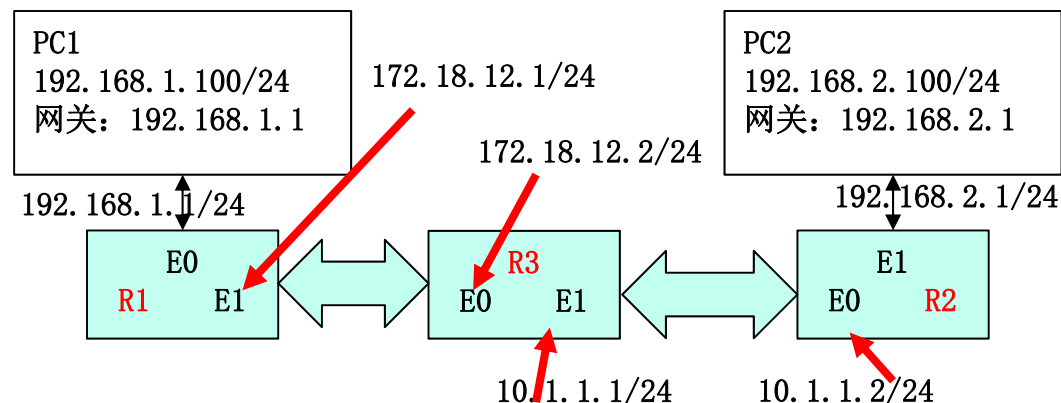
C>* 172.18.12.0/24 is directly connected, Ethernet0

~~S>* 192.168.1.0/24 [1/0] via 172.18.12.1, Ethernet0~~

C>* 192.168.2.0/24 is directly connected, Ethernet1

实验6-环境：两台PC+三台路由器，连接在同一交换机上

PC1 : 192.168.1.100/24, 网关 192.168.1.1
 PC2 : 192.168.2.100/24, 网关 192.168.2.1
 R1-Ethernet0 : 192.168.1.1/24
 Ethernet1 : 172.18.12.1/24
 R2-Ethernet0 : 10.1.1.2/24
 Ethernet1 : 192.168.2.1/24
 R3-Ethernet0 : 172.18.12.2/24
 Ethernet1 : 10.1.1.1/24



- 问：(1) 三台路由器的静态路由应该如何设置才能使PC1/PC2通
 (2) 怎样才能ping通三台路由器的所有端口？
 (3) 分别观察各ping包的ttl值
 (4) 记录路由器的配置以及路由器中路由表的显示

```
C:\Windows\system32>ping 192.168.2.100
```

正在 Ping 192.168.2.100 具有 32 字节的数据：
 来自 192.168.2.100 的回复: 字节=32 时间=2ms TTL=125
 来自 192.168.2.100 的回复: 字节=32 时间=2ms TTL=125
 来自 192.168.2.100 的回复: 字节=32 时间=3ms TTL=125
 来自 192.168.2.100 的回复: 字节=32 时间=2ms TTL=125

192.168.2.100 的 Ping 统计信息:
 数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
 往返行程的估计时间(以毫秒为单位):
 最短 = 2ms, 最长 = 3ms, 平均 = 2ms

PC1 ping PC2 的结果

ts-6352#show running **R1的配置**

```
使用了 8192 字节中的 173 个字节, 软件版本 12.06H
!
ip route 192.168.2.0 255.255.255.0 172.18.12.2
!
interface ethernet 0
ip address 192.168.1.1 255.255.255.0
!
interface ethernet 1
ip address 172.18.12.1 255.255.255.0
!
end
```

ts-22340#show running **R3的配置**

```
使用了 8192 字节中的 214 个字节, 软件版本 12.06H
!
ip route 192.168.1.0 255.255.255.0 172.18.12.1
ip route 192.168.2.0 255.255.255.0 10.1.1.2
!
interface ethernet 0
ip address 172.18.12.2 255.255.255.0
!
interface ethernet 1
ip address 10.1.1.1 255.255.255.0
!
end
```

ts-8#sh running **R2的配置**

```
使用了 8192 字节中的 167 个字节, 软件版本 12.06H
!
ip route 192.168.1.0 255.255.255.0 10.1.1.1
!
interface ethernet 0
ip address 10.1.1.2 255.255.255.0
!
interface ethernet 1
ip address 192.168.2.1 255.255.255.0
!
end
```

ts-6352#show ip route **R1的路由表**

Codes: C - connected, S - static, R - RIP, O - OSPF,
 I - ISIS, B - BGP, > - selected route, * - FIB route

```
C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
C>* 172.18.12.0/24 is directly connected, Ethernet1
C>* 192.168.1.0/24 is directly connected, Ethernet0
S>* 192.168.2.0/24 [1/0] via 172.18.12.2, Ethernet1
```

ts-22340#show ip route **R3的路由表**

Codes: C - connected, S - static, R - RIP, O - OSPF,
 I - ISIS, B - BGP, > - selected route, * - FIB route

```
C>* 10.1.1.0/24 is directly connected, Ethernet1
C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
C>* 172.18.12.0/24 is directly connected, Ethernet0
S>* 192.168.1.0/24 [1/0] via 172.18.12.1, Ethernet0
S>* 192.168.2.0/24 [1/0] via 10.1.1.2, Ethernet1
```

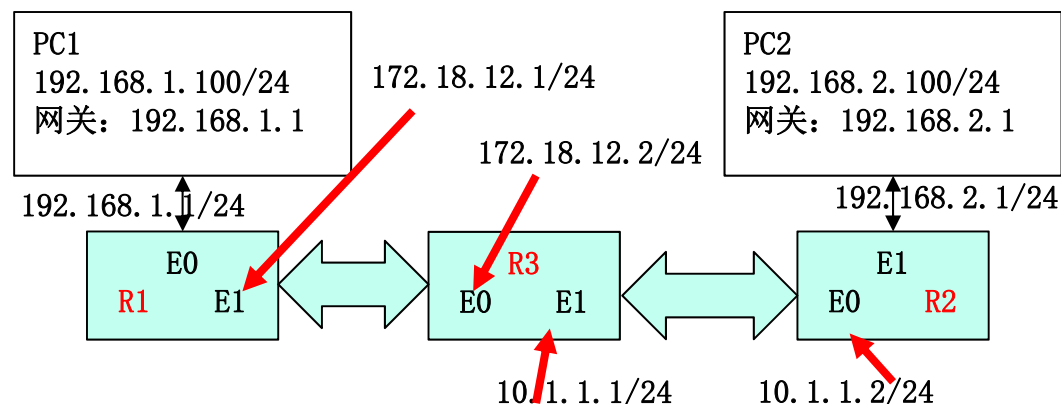
ts-8#sh ip rou **R2的路由表**

Codes: C - connected, S - static, R - RIP, O - OSPF,
 I - ISIS, B - BGP, > - selected route, * - FIB route

```
C>* 10.1.1.0/24 is directly connected, Ethernet0
C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
S>* 192.168.1.0/24 [1/0] via 10.1.1.1, Ethernet0
C>* 192.168.2.0/24 is directly connected, Ethernet1
```

实验6-环境：两台PC+三台路由器，连接在同一交换机上

PC1 : 192.168.1.100/24, 网关 192.168.1.1
 PC2 : 192.168.2.100/24, 网关 192.168.2.1
 R1-Ethernet0 : 192.168.1.1/24
 Ethernet1 : 172.18.12.1/24
 R2-Ethernet0 : 10.1.1.2/24
 Ethernet1 : 192.168.2.1/24
 R3-Ethernet0 : 172.18.12.2/24
 Ethernet1 : 10.1.1.1/24



问:R3中删除一条静态路由的配置后,为什么不通了?
 为什么是172.18.12.2的回复

```
C:\Windows\system32>ping 192.168.2.100
```

正在 Ping 192.168.2.100 具有 32 字节的数据:
 来自 172.18.12.2 的回复: 无法访问目标网。
 来自 172.18.12.2 的回复: 无法访问目标网。
 来自 172.18.12.2 的回复: 无法访问目标网。
 来自 172.18.12.2 的回复: 无法访问目标网。

192.168.2.100 的 Ping 统计信息:
 数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),

PC1 ping PC2 的结果

ts-6352#show running

R1的配置

```
使用了 8192 字节中的 173 个字节, 软件版本 12.06H
!
ip route 192.168.2.0 255.255.255.0 172.18.12.2
!
interface ethernet 0
ip address 192.168.1.1 255.255.255.0
!
interface ethernet 1
ip address 172.18.12.1 255.255.255.0
!
end
```

ts-22340#show running

R3的配置

```
使用了 8192 字节中的 214 个字节, 软件版本 12.06H
!
ip route 192.168.1.0 255.255.255.0 172.18.12.1
ip route 192.168.2.0 255.255.255.0 10.1.1.2
!
interface ethernet 0
ip address 172.18.12.2 255.255.255.0
!
interface ethernet 1
ip address 10.1.1.1 255.255.255.0
!
end
```

ts-8#sh running

R2的配置

```
使用了 8192 字节中的 167 个字节, 软件版本 12.06H
!
ip route 192.168.1.0 255.255.255.0 10.1.1.1
!
interface ethernet 0
ip address 10.1.1.2 255.255.255.0
!
interface ethernet 1
ip address 192.168.2.1 255.255.255.0
!
end
```

ts-6352#show ip route

R1的路由表

Codes: C - connected, S - static, R - RIP, O - OSPF,
 I - ISIS, B - BGP, > - selected route, * - FIB route

```
C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
C>* 172.18.12.0/24 is directly connected, Ethernet1
C>* 192.168.1.0/24 is directly connected, Ethernet0
S>* 192.168.2.0/24 [1/0] via 172.18.12.2, Ethernet1
```

ts-22340#show ip route

R3的路由表

Codes: C - connected, S - static, R - RIP, O - OSPF,
 I - ISIS, B - BGP, > - selected route, * - FIB route

```
C>* 10.1.1.0/24 is directly connected, Ethernet1
C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
C>* 172.18.12.0/24 is directly connected, Ethernet0
S>* 192.168.1.0/24 [1/0] via 172.18.12.1, Ethernet0
S>* 192.168.2.0/24 [1/0] via 10.1.1.2, Ethernet1
```

ts-8#sh ip rou

R2的路由表

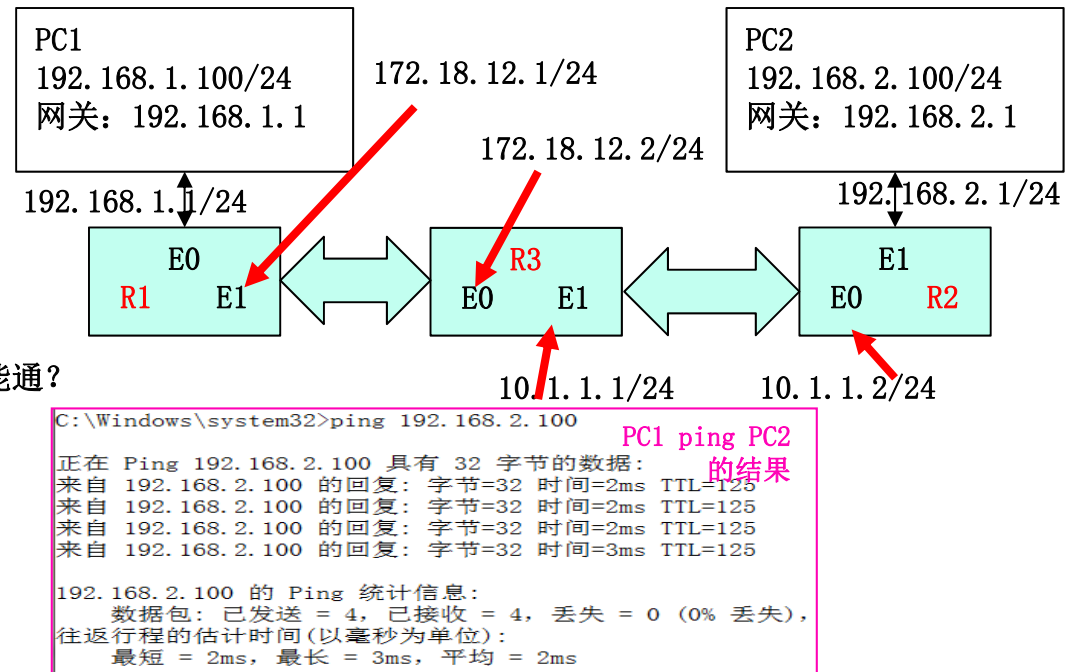
Codes: C - connected, S - static, R - RIP, O - OSPF,
 I - ISIS, B - BGP, > - selected route, * - FIB route

```
C>* 10.1.1.0/24 is directly connected, Ethernet0
C>* 127.0.0.0/8 is directly connected, lo
C>* 127.0.0.0/24 is directly connected, lo
S>* 192.168.1.0/24 [1/0] via 10.1.1.1, Ethernet0
C>* 192.168.2.0/24 is directly connected, Ethernet1
```

实验7: 所有接口IP地址配置同实验6, 删除所有静态路由, 改为RIP动态路由

R1: network 192.168.1.0 255.255.255.0
network 172.18.12.0 255.255.255.0
R2: network 10.1.1.0 255.255.255.0
network 192.168.2.0 255.255.255.0
R3: network 172.18.12.0 255.255.255.0
network 10.1.1.0 255.255.255.0

问: (1) PC1/PC2之间, PC与3台路由器6个端口之间是否能通?
(2) 分别观察各ping包的ttl值
(3) 记录路由器的配置以及路由器中路由表的显示



ts-6352#show running

R1的配置

使用了 8192 字节中的 198 个字节, 软件版本 12.06H

!

interface ethernet 0

ip address 192.168.1.1 255.255.255.0

!

interface ethernet 1

ip address 172.18.12.1 255.255.255.0

!

router rip

network 172.18.12.0 255.255.255.0

network 192.168.1.0 255.255.255.0

!

end

ts-22340#show running

R3的配置

使用了 8192 字节中的 192 个字节, 软件版本 12.06H

!

interface ethernet 0

ip address 172.18.12.2 255.255.255.0

!

interface ethernet 1

ip address 10.1.1.1 255.255.255.0

!

router rip

network 10.1.1.0 255.255.255.0

network 172.18.12.0 255.255.255.0

!

end

ts-8#show running

R2的配置

使用了 8192 字节中的 192 个字节, 软件版本 12.06H

!

interface ethernet 0

ip address 10.1.1.2 255.255.255.0

!

interface ethernet 1

ip address 192.168.2.1 255.255.255.0

!

router rip

network 10.1.1.0 255.255.255.0

network 192.168.2.0 255.255.255.0

!

end

ts-6352#show ip route

R1的路由表

Codes: C - connected, S - static, R - RIP, O - OSPF,

I - ISIS, B - BGP, > - selected route, * - FIB route

R>* 10.1.1.0/24 [120/2] via 172.18.12.2, Ethernet1, 00:00:49

C>* 127.0.0.0/8 is directly connected, lo

C>* 127.0.0.0/24 is directly connected, lo

C>* 172.18.12.0/24 is directly connected, Ethernet1

C>* 192.168.1.0/24 is directly connected, Ethernet0

R>* 192.168.2.0/24 [120/3] via 172.18.12.2, Ethernet1, 00:00:49

ts-22340#show ip route

R3的路由表

Codes: C - connected, S - static, R - RIP, O - OSPF,

I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 10.1.1.0/24 is directly connected, Ethernet1

C>* 127.0.0.0/8 is directly connected, lo

C>* 127.0.0.0/24 is directly connected, lo

C>* 172.18.12.0/24 is directly connected, Ethernet0

R>* 192.168.1.0/24 [120/2] via 172.18.12.1, Ethernet0, 00:02:25

R>* 192.168.2.0/24 [120/2] via 10.1.1.2, Ethernet1, 00:03:13

ts-8#show ip route

R2的路由表

Codes: C - connected, S - static, R - RIP, O - OSPF,

I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 10.1.1.0/24 is directly connected, Ethernet0

C>* 127.0.0.0/8 is directly connected, lo

C>* 127.0.0.0/24 is directly connected, lo

R>* 172.18.12.0/24 [120/2] via 10.1.1.1, Ethernet0, 00:02:25

R>* 192.168.1.0/24 [120/3] via 10.1.1.1, Ethernet0, 00:01:37

C>* 192.168.2.0/24 is directly connected, Ethernet1

§. IP地址的基本概念

6. 划分子网

6.1. 现有分类IP地址存在的问题

红色为(来自于18.0.0.0/8) 请求访问145.13.*.*
蓝色为(来自于221.12.76/24)请求访问145.13.*.*
R2-E0: 10.1.1.1/24 R1-E3:10.1.1.2/24
R3-E0: 20.1.1.1/24 R1-E4:20.1.1.2/24

问: R1/R2/R3中如何设置静态路由?

R2的路由表中应该有以下3项:

```
ip route 145.13.3.0 255.255.255.0 10.1.1.2
```

```
ip route 145.13.7.0 255.255.255.0 10.1.1.2
```

```
ip route 145.13.21.0 255.255.255.0 10.1.1.2
```

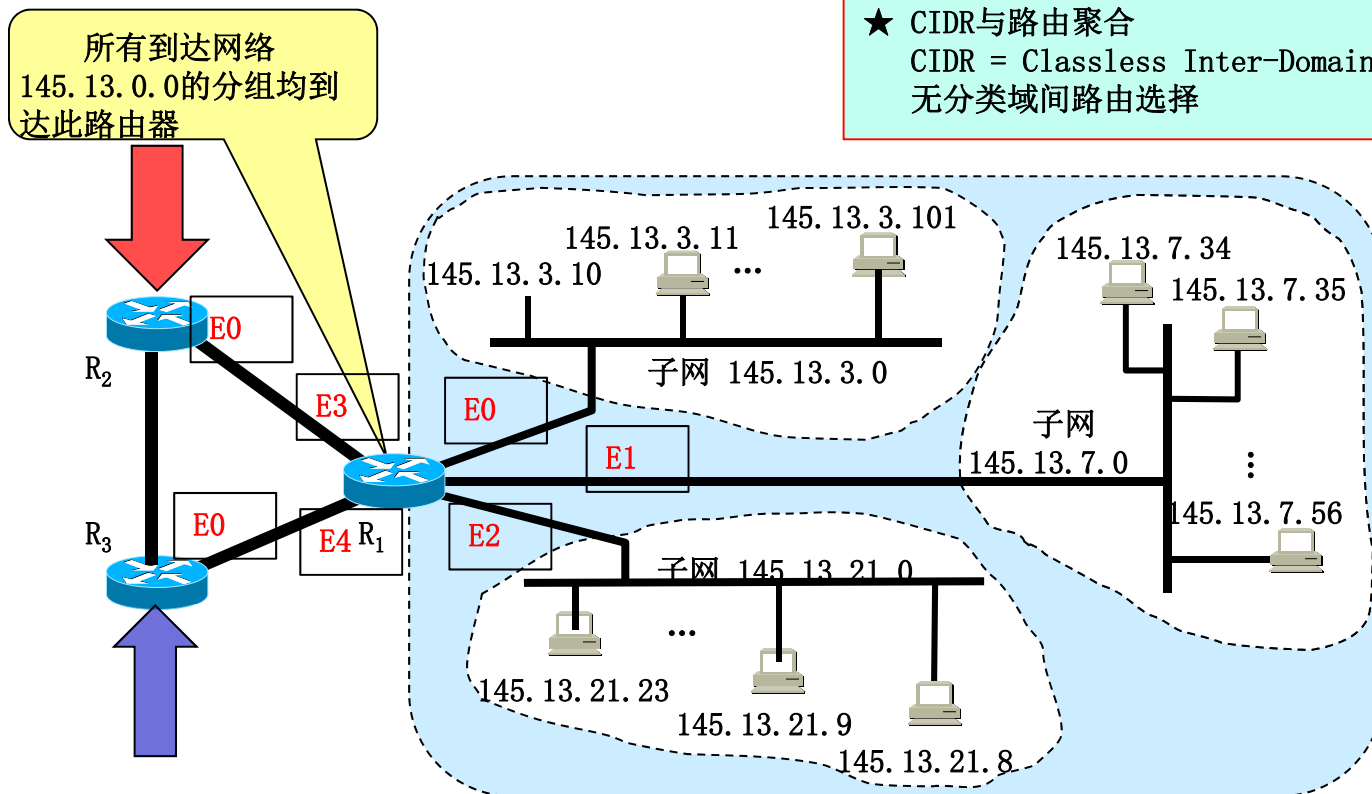
问: 什么情况下能合并为一项?

```
ip route 145.13.0.0 255.255.0.0 10.1.1.2
```

★ CIDR与路由聚合

CIDR = Classless Inter-Domain Routing

无分类域间路由选择



§ . 路由器的基本功能及使用

5. Windows及Linux下路由表的查看与修改

查看:

Windows: `route print`

Linux: `route -n`

修改:

自行查阅资料

§ . 路由器的基本功能及使用

6. 网络协议中各层的数据交换

回顾：协议层次结构

面向连接服务与无连接服务

具体见课件：“010001. 第01章 引言.PDF” P. 15 – P. 22

6. 1. 中继器/集线器/网桥/交换机/路由器和网关

第四章中的内容提前：

P. 263 – P. 265

4. 8. 4. 中继器/集线器/网桥/交换机/路由器和网关

§. 路由器的基本功能及使用

6. 网络协议中各层的数据交换

6.1. 中继器/集线器/网桥/交换机/路由器和网关

★ 名词解释

中继器 (repeater)：工作于OSI参考模型的最底层(物理层)的连接设备，通过对传输过程中出现衰减的数据信号进行放大和整形(再生和还原)后的重新发送或者转发，来扩大网络传输距离，一般适用于完全相同的两类网络的互连

集线器 (hub)：与中继器一样，工作于OSI参考模型的物理层，主要功能是对接收到的信号进行放大和整形，以扩大网络的传输距离，可以理解为一个多端口的中继器，一个端口收到数据后，会广播式发送到其余的所有端

- 多端口同时收发，则会出现碰撞冲突，解决方法在第四章，称为CSMA/CD(带冲突检测的载波监听多路访问技术)

- 只能工作于半双工(Half Duplex)模式，即数据可以双向传输，但是不能同时传输

网桥 (bridge)：工作于OSI参考模型的第二层(数据链路层)的连接设备，具有过滤帧的功能，根据接收到的MAC帧的目的地址对收到的帧进行转发(非广播式发送，而是先检查此帧的目的MAC地址，然后再确定将该帧转发到哪个接口)

- 多端口同时收发，不会出现碰撞冲突

- 可以工作于全双工(Full Duplex)模式，即数据可以同时双向传输

交换机 (switch)：网桥的一种，常用的是以太网交换机，采用交换技术分隔冲突域，交换技术的作用是根据所传递帧的目的MAC地址，将每帧独立从发送端口送至目的端口，避免与其它端口发生碰撞冲突，提高了网络的实际吞吐量

§. 路由器的基本功能及使用

6. 网络协议中各层的数据交换

6.1. 中继器/集线器/网桥/交换机/路由器和网关

★ 名词解释

集线器 (hub)：与中继器一样，工作于OSI参考模型的物理层，主要功能是对接收到的信号进行放大和整形，以扩大网络的传输距离，可以理解为一个多端口的中继器，一个端口收到数据后，会广播式发送到其余的所有端

- 多端口同时收发，则会出现碰撞冲突，解决方法在第四章，称为CSMA/CD(带冲突检测的载波监听多路访问技术)

- 只能工作于半双工(Half Duplex)模式，即数据可以双向传输，但是不能同时传输

交换机 (switch)：网桥的一种，常用的是以太网交换机，采用交换技术分隔冲突域，交换技术的作用是根据所传递帧的目的MAC地址，将每帧独立从发送端口送至目的端口，避免与其它端口发生碰撞冲突，提高了网络的实际吞吐量

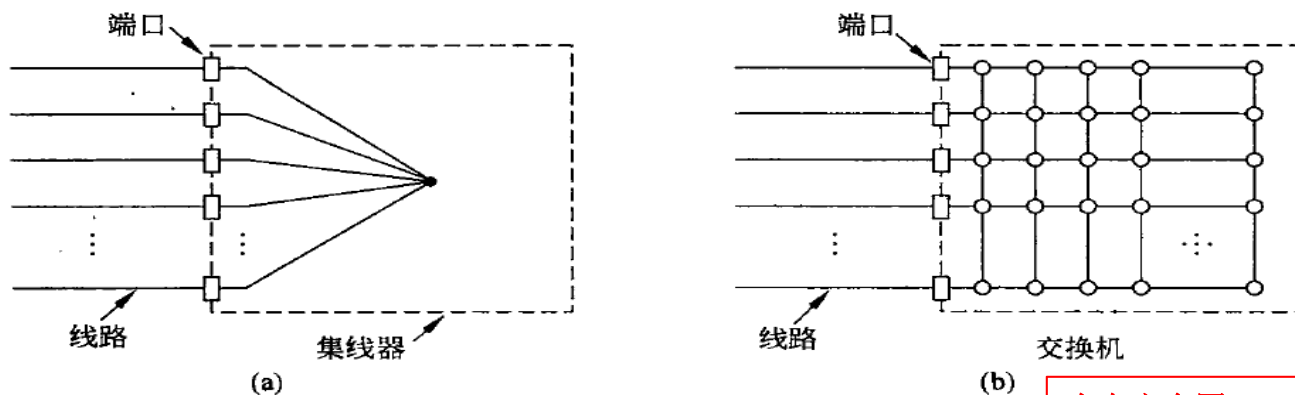


图 4-17
(a) 集线器; (b) 交换机

有向完全图
更容易理解

§. 路由器的基本功能及使用

6. 网络协议中各层的数据交换

6.1. 中继器/集线器/网桥/交换机/路由器和网关

★ 名词解释

路由器 (router)：工作于OSI参考模型的第三层(网络层)的连接设备，用于连接多个逻辑上分开的网络(局域网、广域网)，根据当前拓扑结构自动选择和设定路由，以最佳路径转发数据包

- 交换机使用第二层的MAC地址判断，路由器使用第三层的IP地址判断
- 路由器通过路由决定数据的转发。转发策略称为路由选择(routing)
- 路由器连接多个网段，只有指向特定IP地址的数据包才可以通过路由器。对于每一个接收到的数据包，路由器都会重新计算其校验值并写入新的MAC地址(路由器转发数据比交换机慢，但是对于那些结构复杂的网络，使用路由器可以提高网络的整体效率)

网关 (gateway)：工作于OSI参考模型第三层(网络层)及以上的连接设备，又称网间连接器、协议转换器。网关在网络层上以实现网络互连，是最复杂的网络互连设备，仅用于两个高层协议不同的网络互连

- 由于历史的原因，许多TCP/IP的文献把路由器称为网关，因此通常所指的网关就是路由器的IP
- 传输网关：连接运行不同传输协议的网关
- 应用网关：在不同应用间进行格式转换

§. 路由器的基本功能及使用

6. 网络协议中各层的数据交换

6.1. 中继器/集线器/网桥/交换机/路由器和网关

★ 名词解释

★ 工作在不同层次

应用层	应用网关
传输层	传输网关
网络层	路由器
数据链路层	网桥, 交换机
物理层	中继器, 集线器

(a)

6.2. 网桥的原理及使用

第四章中的内容提前:

P. 256 – P. 262

4.8.1. 网桥的使用

4.8.2. 学习网桥

4.8.4. 生成树网桥

§ 4. 介质访问控制子层

4.8. 数据链路层交换

4.8.1. 网桥的使用

★ 理想网桥的特点

- 即插即用，只需将网桥与各个欲通过它互联的局域网连接起来，整个系统就能够正常工作，不需要做任何硬件或软件上的设置，增删节点非常容易
- 网桥的插入不会中断现有网络的运行
- 网桥的存在对网络用户是透明的（透明网桥）
- “透明”是指局域网上的站点不知道所发送的帧将经过哪几个网桥，即网桥对各站来说是看不见的
- 透明网桥的标准是 IEEE 802.1D

★ 透明网桥的学习算法

- 向后学习算法(backward learning)
- 生成树算法(spanning tree)

§ 4. 介质访问控制子层

4.8. 数据链路层交换

4.8.1. 网桥的使用

★ 网桥常用的交换模式

- 直通转发模式 (Cut-through)：收到帧的前6字节（目的MAC地址）就直接转发，不进行缓存和校验
 - ◆ 转发时延短
 - ◆ 适用于相同速率的转发，否则会堵塞
 - ◆ 可能转发残帧、错误帧
- 存储转发模式 (Store-forward)：在缓冲区中存储整个接收到的封装数据包，然后检验数据包是否正确，正确则转发，否则丢弃
 - ◆ 转发时延长
 - ◆ 缓存空间大（要有流控算法）
 - ◆ 可以适应不同速率的转发
 - ◆ 屏蔽残帧和错误帧
- 准直通转发模式 (Interim cut-through switch)：对直通转发模式的一种简单改进（采用一种特殊的缓存：FIFO），只转发长度 ≥ 64 Bytes的帧，从而避免了残帧的转发
- 智能交换模式 (Intelligent)：能够根据所监控网络中错误包传输的数量，自动智能改变转发模式。如果堆栈发觉每秒错误少于限定值，将自动采用直通式转发模式；如果堆栈发觉每秒错误大于限定值或更多，将自动采用存储转发模式，直到返回的错误数量为0时，再切换回直通式转发模式

问题：直通转发模式时如何识别错误？

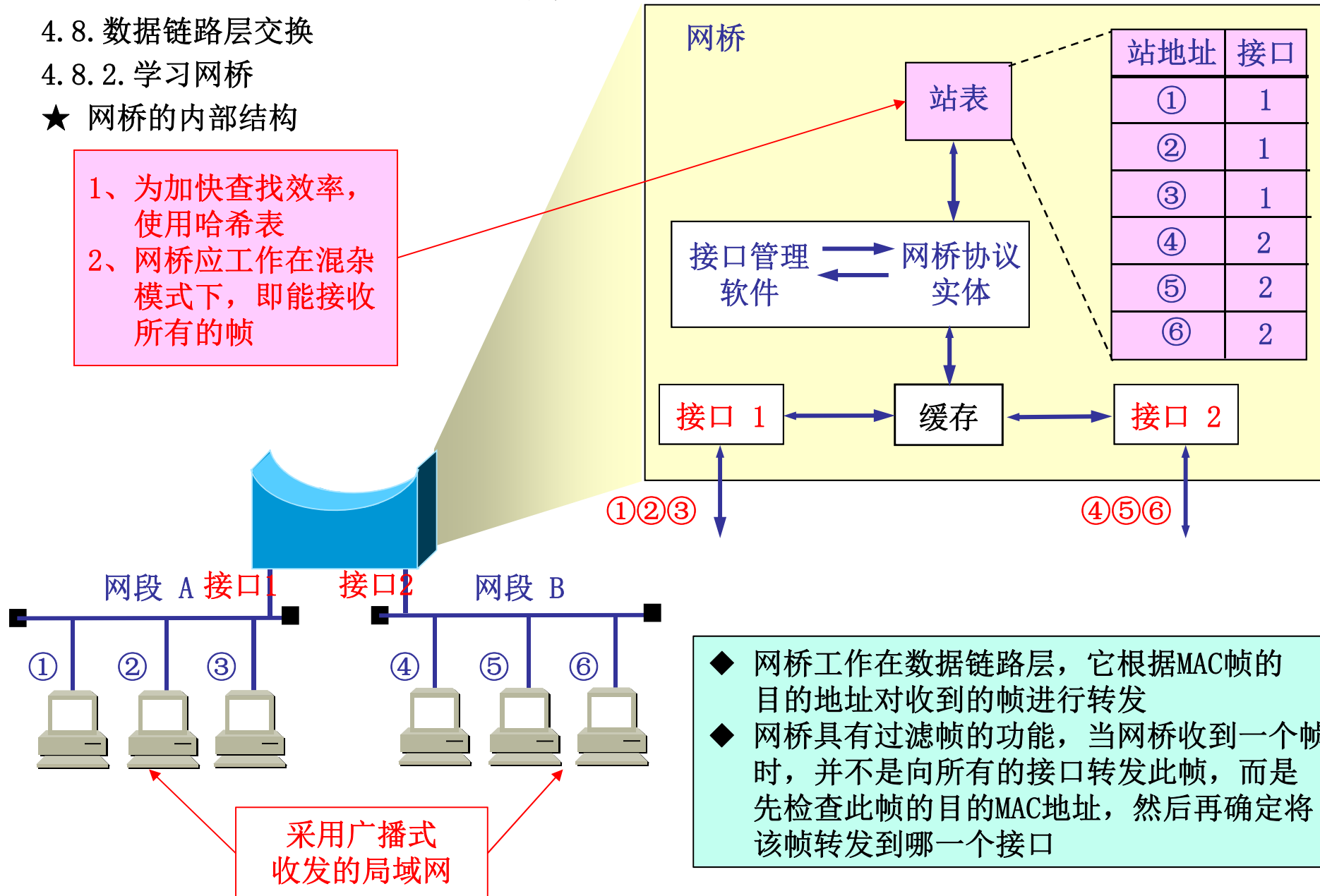
§ 4. 介质访问控制子层

4.8. 数据链路层交换

4.8.2. 学习网桥

★ 网桥的内部结构

- 1、为加快查找效率，使用哈希表
- 2、网桥应工作在混杂模式下，即能接收所有的帧



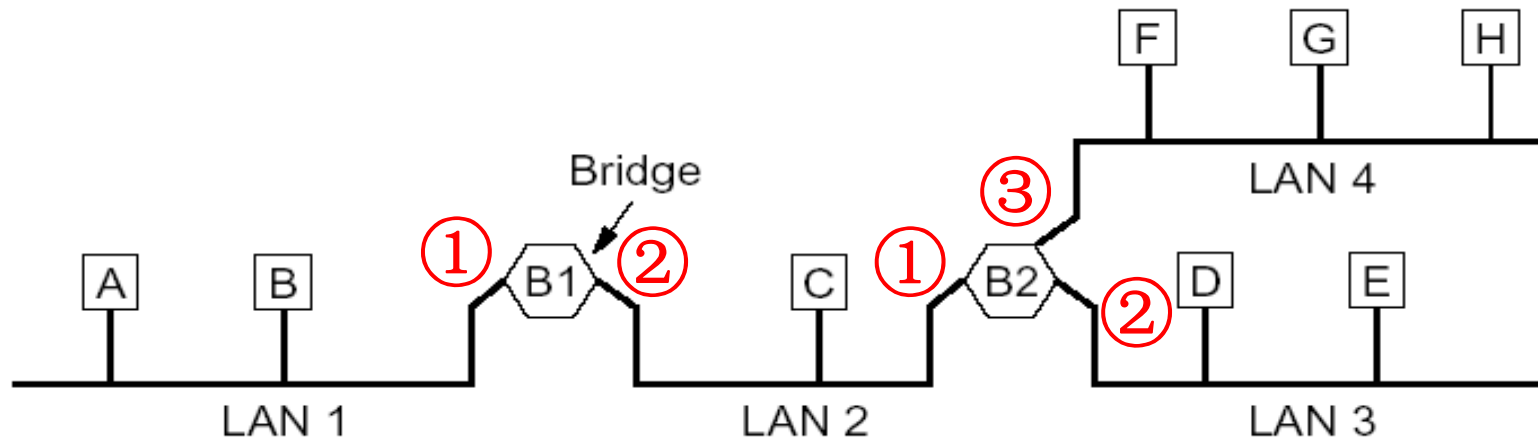
- ◆ 网桥工作在数据链路层，它根据MAC帧的目的地址对收到的帧进行转发
- ◆ 网桥具有过滤帧的功能，当网桥收到一个帧时，并不是向所有的接口转发此帧，而是先检查此帧的目的MAC地址，然后再确定将该帧转发到哪一个接口

§ 4. 介质访问控制子层

4.8. 数据链路层交换

4.8.2. 学习网桥

★ 网桥的内部结构



网桥B1中的转发表

MAC地址	端口
A	1
B	1
C	2
D	2
E	2
F	2
G	2
H	2

网桥B2中的转发表

MAC地址	端口
A	1
B	1
C	1
D	2
E	2
F	3
G	3
H	3

§ 4. 介质访问控制子层

4.8. 数据链路层交换

4.8.2. 学习网桥

★ 透明网桥的工作原理

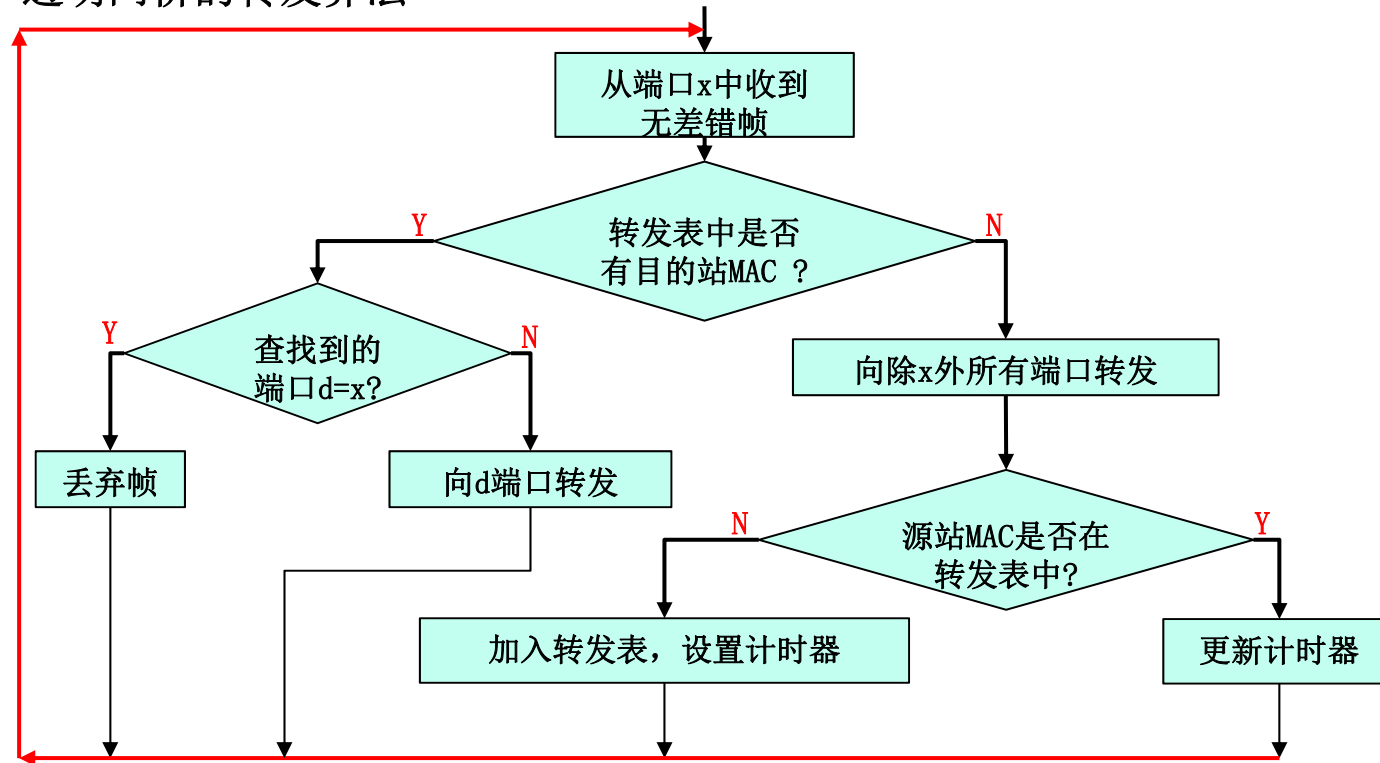
- 问题1：网桥如何知道是否应当转发帧，以及应往哪个端口转发？
 - ◆ 网桥内部使用一张转发表，表中记录各个目的地址在网桥的哪个端口上
 - ◆ 网桥用帧的目的地址查找转发表，若目的地址所在端口与帧的输入端口相同，丢弃帧；否则从目的地址所在端口转发帧
 - 问题2：如何生成网桥中的转发表？
 - ◆ 转发表初始为空，网桥通过**逆向学习**法获知各个目的地址所在端口，逐步建立转发表
 - ◆ **逆向学习**：网桥通过检查帧的源地址及输入端口来发现网络中的节点及所在的端口
 - 问题3：若转发表尚未完全建立或出现了新节点，即帧的目的地址不在转发表中，网桥该如何转发？
 - ◆ 对于每个发向未知目的地址对应端口的帧，使用**泛洪算法**(flooding algorithm)转发到除源端口外的所有端口，学习成功后则不再泛洪而只转发到正确端口
 - 问题4：网桥学习到的知识过时了怎么办？
 - ◆ 给转发表的每个表项添加一个时间项，称为生存期，有转发则更新生存期为当前时间
 - ◆ 网桥定期扫描转发表，发现当前时间与生存期的差值超过设定值，则删去该表项
 - 问题5：知识未过时，但某主机接入网桥的端口已经发生了改变怎么办？
 - ◆ 如果其它主机给它发包，不通
 - ◆ 如果它主动给其它主机发包，则更新转发表中所登记的端口
- => 改进：每次切换端口时，主机的网卡会感知并UP/DOWN，因此，每次UP时主动发送包含自己的MAC地址的广播包即可

§ 4. 介质访问控制子层

4. 8. 数据链路层交换

4. 8. 2. 学习网桥

★ 透明网桥的转发算法



★ 另一个进程周期性扫描转发表并删除当前时间与计时器差值超过设定值的表项

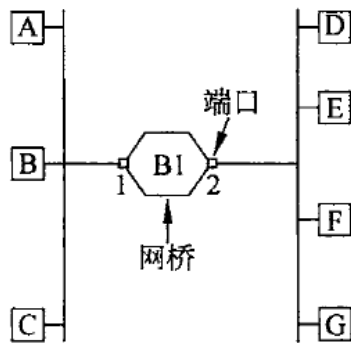
§ 4. 介质访问控制子层

4.8. 数据链路层交换

4.8.2. 学习网桥

★ 透明网桥的转发算法

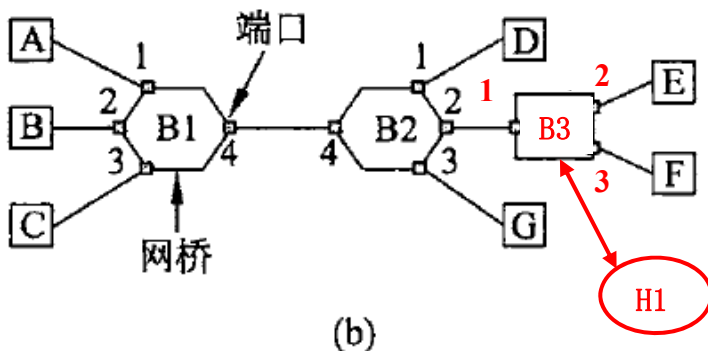
- 为提高转发效率，某些网桥（例：交换机）只要收到目的MAC地址就可以开始转发（此时帧还未接收完成），称为直通式交换/虫孔路由（WR = Wormhole Routing）
- 网桥转发时，帧的格式可能重写，也可能不变，只要网络层协议相同即可
- 多个网桥级联时，每个网桥只维护自己的转发表，不会造成冲突



(a)

假设以太网，ABC通过HUB互联，
DEFG通过HUB互联，B1为交换机

- ◆ A/B/C ⇔ A/B/C
B1的1#口收到帧，丢弃
- ◆ D/E/F/G ⇔ D/E/F/G
B1的2#口收到帧，丢弃
- ◆ A/B/C ⇔ D/E/F/G
B1的1#/2#相互转发



(b)

假设以太网，B1/B2/B3为交换机

- ◆ A/B/C ⇔ A/B/C B1内转发
- ◆ D/G ⇔ D/G B2内转发
- ◆ E/F ⇔ E/F B3内转发
- B1/B2/B3间无转发

将B3替换为HUB H1

- ◆ E/F ⇔ E/F
H1广播式发送，因此B2的2#
收到E/F发来的信息，但丢弃

假设以太网，B1/B2/B3为交换机

- ◆ A/B/C ⇔ D/G
- A/B/C ⇔ E/F
- D/G ⇔ E/F
- B1的4#：有DGEF信息
- B2的2#：有EF信息
- 4#：有ABC信息
- B3的1#：有ABCDG信息
- B1/B2/B3相互转发

§ 4. 介质访问控制子层

4.8. 数据链路层交换

4.8.3. 生成树网桥

★ 引入

在透明网桥中，网桥间只有一条链路，导致可靠性不高，为了提高可靠性，可以采用冗余备份的方式

- 产生的问题：(第一次)目的地址未知的帧会循环冗余转发，目的地址已知的帧会重复发送

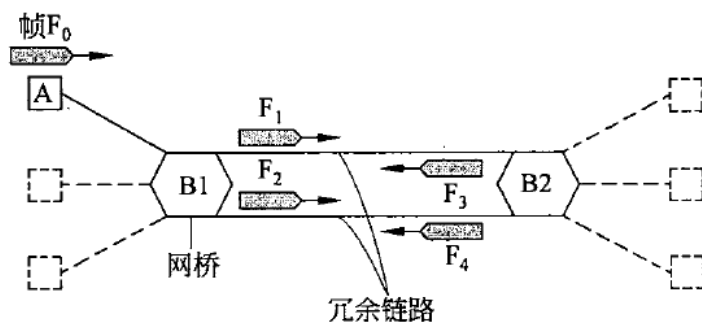


图 4-43 具有两条平行链路的网桥

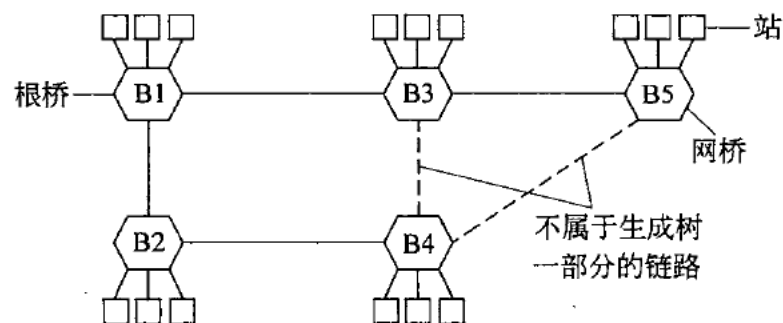


图 4-44 一棵连接 5 座网桥的生成树。虚线表示不属于生成树的链路

★ 解决循环冗余转发的方法

- 在有环的网络中，仅在网桥发生故障时才启用冗余链路，平时只使用没有环的拓扑转发帧
- 将网桥抽象成边，局域网抽象成顶点，形成网络拓扑图。求该图的一棵生成树 (spanning tree)，使得覆盖图中所有的顶点但没有环路
- 生成树构造完成后，只有位于生成树中的网桥能够在属于生成树的边上转发帧

§ 4. 介质访问控制子层

4.8. 数据链路层交换

4.8.3. 生成树网桥

★ 生成树的构造算法

- 选举根网桥：各个网桥广播自己的序列号，序列号最小的网桥成为生成树的根。根网桥可在所有的端口上转发帧
- 每个网桥计算自己到根的最短路径（跳数），记录自己的哪个端口在最短路径上，这个端口即为该网桥到根的优先路径
- 所有连接到同一个LAN上的网桥选一个指派网桥（designated bridge），指派网桥是这些网桥中离根最近的；若距离相等，则选序列号最小的
- 生成树的节点由根网桥和指派网桥组成，指派网桥到根网桥最短路径上的端口成为生成树上的边
- 生成树算法始终在每个网桥(包括非指派网桥)上运行，当某个网桥或局域网出现故障时，重新计算生成树

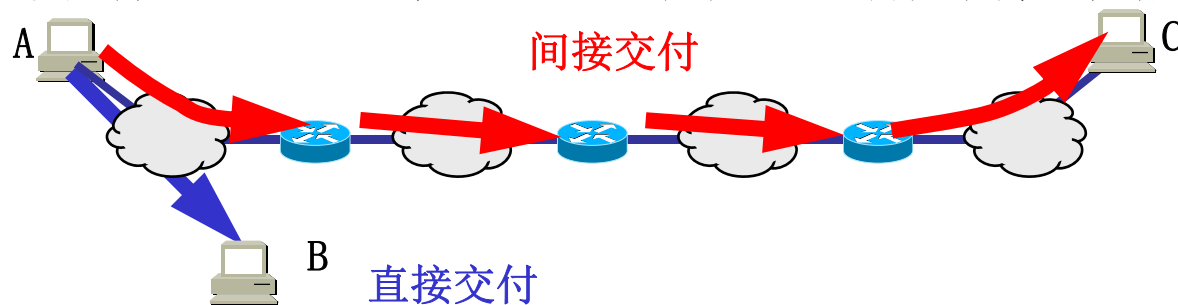
§. 路由器的基本功能及使用

6. 网络协议中各层的数据交换

6.3. 路由器的原理及使用

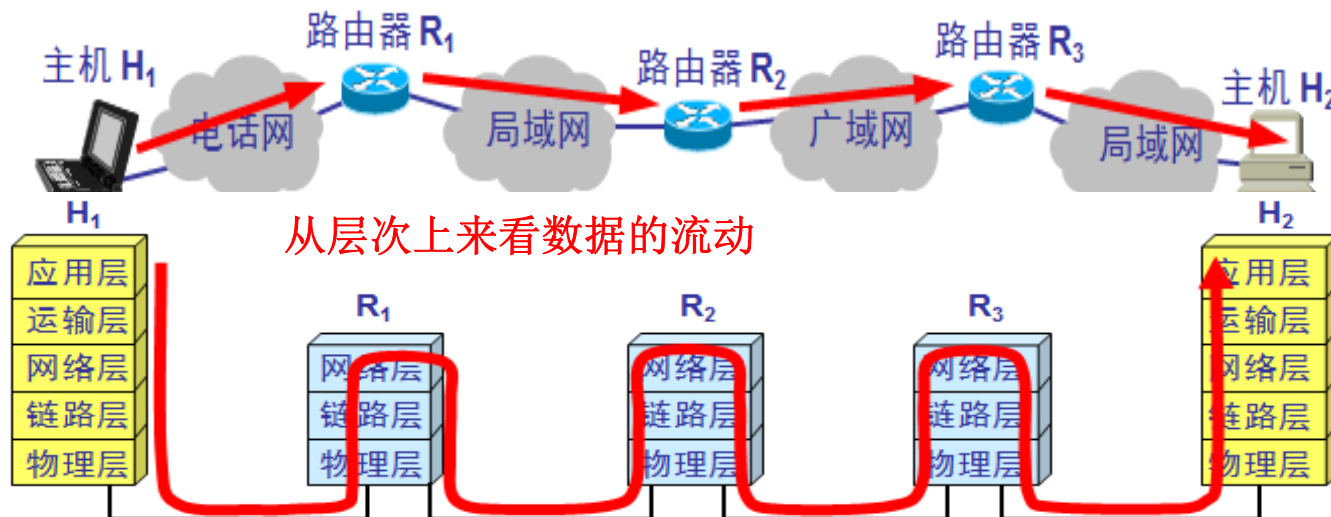
★ 直接交付与间接交付

- 直接交付：源和目的主机间不经过路由器而直接传递数据包（会经过网桥/HUB等）
- 间接交付：源和目的主机间通过1-n个路由器的转发来传递数据包（也有网桥/HUB等）



★ 虚拟通信与实际通信

主机 H_1 向 H_2 发送数据



§. 路由器的基本功能及使用

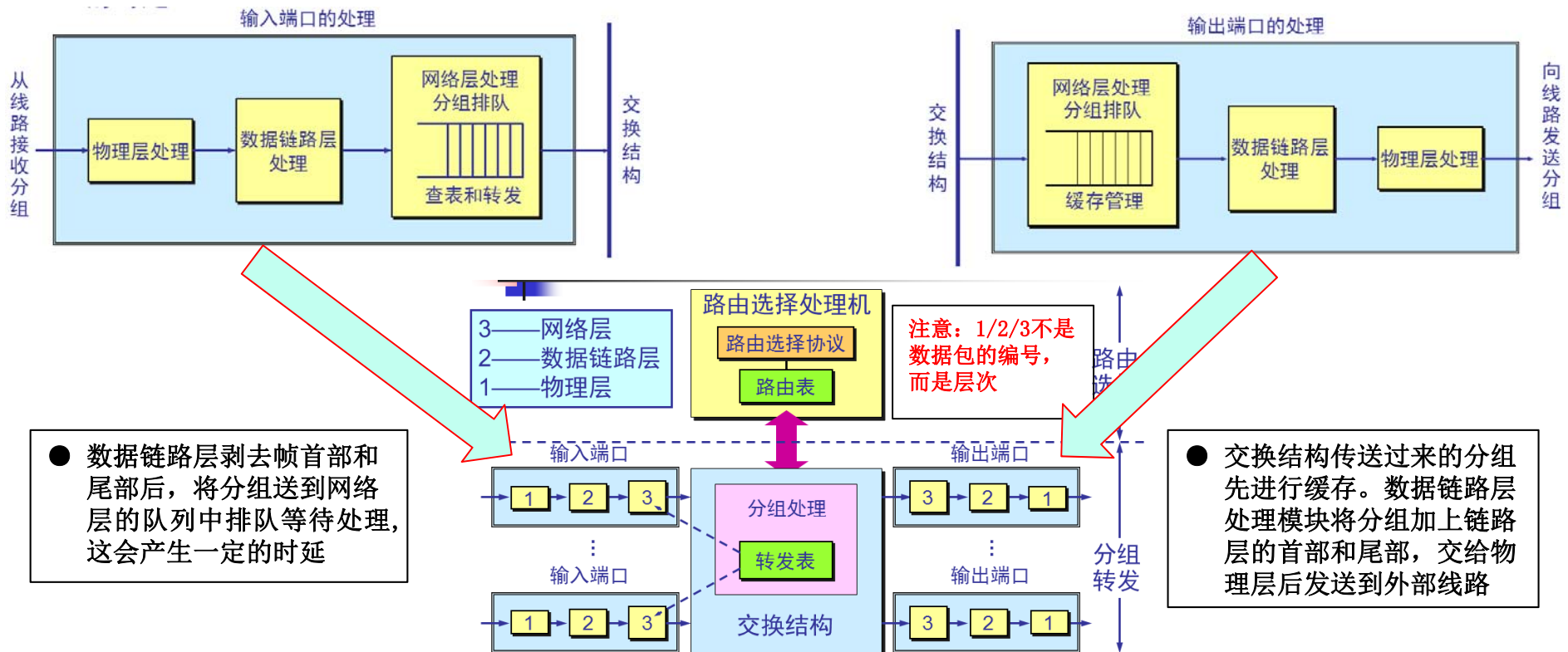
6. 网络协议中各层的数据交换

6.3. 路由器的原理及使用

★ 路由器的工作原理

- 当主机 A 要向另一个主机 B 发送数据报时，先要检查目的主机 B 是否与源主机 A 连接在同一个网络上(对IP地址，就是是否同一网段)
- 如果是，就将数据报直接交付给目的主机 B 而不需要通过路由器(可能会经过网桥/HUB等)
- 如果不是，则应将数据报发送给本网络上的某个路由器，由该路由器按照转发表指出的路由将数据报转发给下一个路由器，多次转发直到到达目的主机B为止(也可能会丢弃)

=> 路由器至少两个接口，连接两个网段



§. 路由器的基本功能及使用

6. 网络协议中各层的数据交换

6.3. 路由器的原理及使用

★ 路由器的工作原理

- “转发”(forwarding)就是路由器根据转发表将用户的 IP 数据报从合适的端口转发出去
- “路由选择”(routing)则是按照分布式算法, 根据从各相邻路由器得到的关于网络拓扑的变化情况, 动态地改变所选择的路由
- 路由表是根据路由选择算法得出的。而转发表是从路由表得出的
- 在讨论路由选择的原理时, 往往不区分转发表和路由表的区别

★ 分组丢弃

- 若路由器处理分组的速率赶不上分组进入队列的速率, 则队列的存储空间最终必定减少到零, 这就使后面再进入队列的分组由于没有存储空间而只能被丢弃, 称为分组丢弃
- 路由器中的输入或输出队列产生溢出是造成分组丢弃的重要原因

★ 虚拟互连网络

- 所谓虚拟互连网络也就是逻辑互连网络, 含义是互连起来的各种物理网络的异构性本身是客观存在的, 但可以利用跨不同物理链路层的网络层协议使这些性能各异的网络从角度用户看起来好像是一个统一的网络
- 使用虚拟互连网络的好处是, 当互联网上的主机进行通信时, 就好像在一个网络上通信一样, 而看不见互连的各具体的网络异构细节
- 使用 IP 协议的虚拟互连网络可简称为 IP 网

§ . 路由器的基本功能及使用

6. 网络协议中各层的数据交换

6. 1. 中继器/集线器/网桥/交换机/路由器和网关

6. 2. 网桥的原理及使用

6. 3. 路由器的原理及使用

第五章中的内容提前：

P. 274 – P. 277

5. 1. 1. 存储-转发数据包交换

5. 1. 2. 提供给传输层的服务

5. 1. 3. 无连接服务的实现

§ 5. 网络层

5.1. 网络层的设计问题

5.1.1. 存储转发数据包交换

★ 网络层的数据包传输采用存储-转发机制

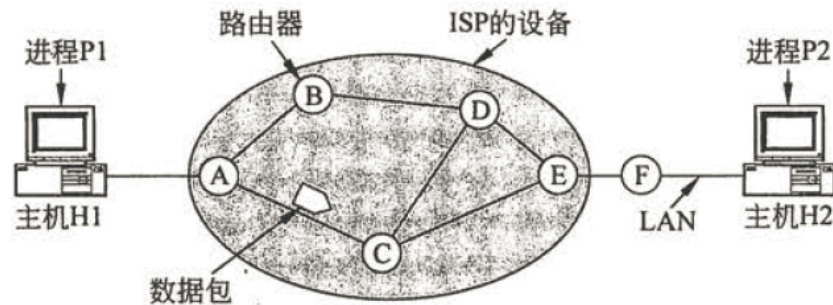


图 5-1 网络层协议的环境

- 所提供的服务与数据链路层类型无关
- 网络层的数量、类型和拓扑结构对传输层透明
- 网络层的编址方式应当跨越多个LAN及WAN

5.1.2. 提供给传输层的服务

★ 网络层为连接在网络上的主机所提供的服务有两类：

- 无连接的网络服务（数据报服务）
- 面向连接的网络服务（虚电路服务）

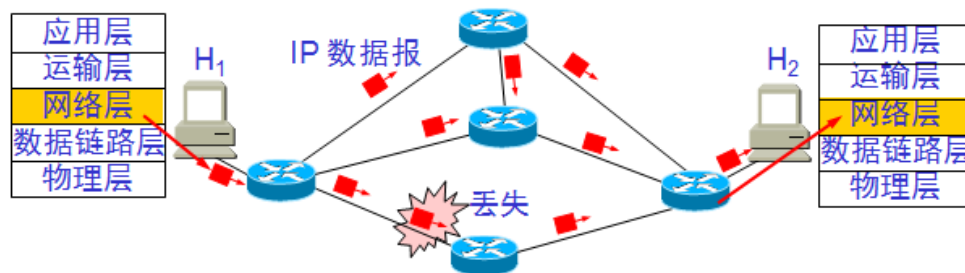
§ 5. 网络层

5.1. 网络层的设计问题

5.1.3. 无连接服务的实现

★ 基本思路

- 传输层传输的数据可以被分为若干数据报 (datagram)
- 每个数据报在网络层传输时是独立的
- 对应的网络层称为数据报网络 (datagram network)



H₁ 发送给 H₂ 的分组可能沿着不同路径传送

★ 无连接服务的特点

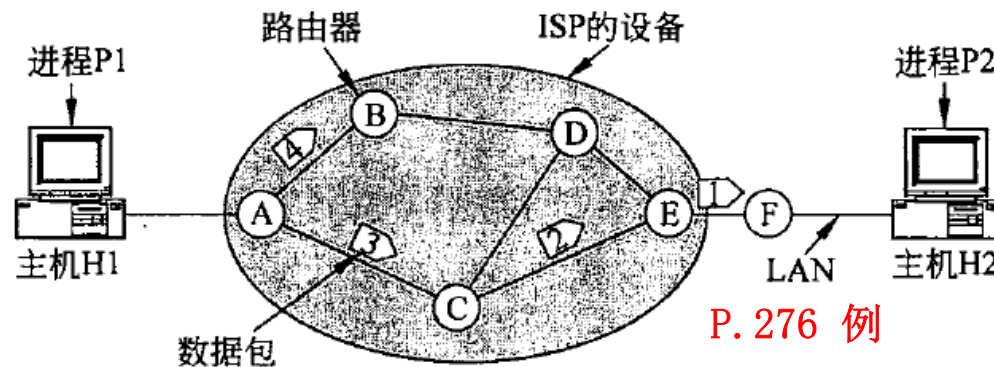
- 网络层向上提供简单灵活的、无连接的、不可靠的数据报服务（尽最大努力交付）
- 网络层在发送分组时不需要先建立连接。每一个分组独立发送，与其前后的分组无关（不编号）
- 网络层不提供服务质量的承诺。即所传送的分组可能出错、丢失、重复和失序（不按序到达终点），当然也不保证分组传送的时限
- Internet即使此方式，数据报为IP数据报

§ 5. 网络层

5.1. 网络层的设计问题

5.1.3. 无连接服务的实现

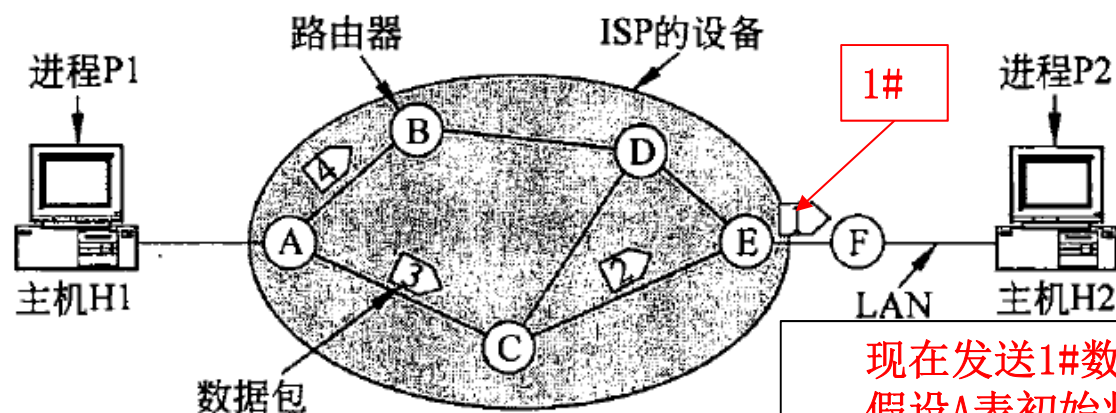
★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器



隐含
的
信息

- 1、ABCDEF至少有3/2/3/3/3/2个接口
(每条线代表一对同网段地址，不同线不同)
- 2、主机H1和H2不在同一网段(需要间接交付)
- 3、主机H1与A的一个接口在同一网段，主机H1的缺省网关设置为A的同网段地址
- 4、主机H2与F的一个接口在同一网段，主机H2的缺省网关设置为F的同网段地址

★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器



现在发送1#数据包：
假设A表初始状态如左图

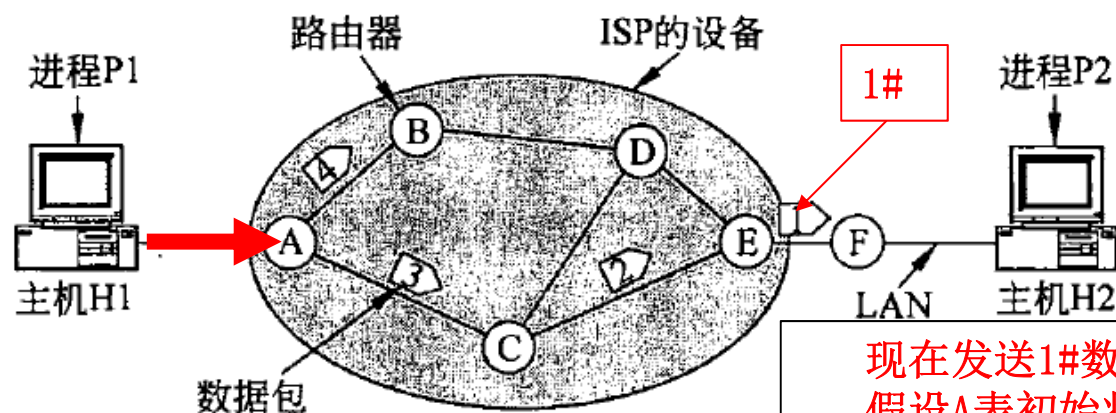
A的表(初始化)

A	—
B	B
C	C
D	B
E	C
F	C

} 直接
转发
}
间接
转发

图 5-2 数据报网络中的路由过程

★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器



现在发送1#数据包：
假设A表初始状态如左图
1、A收到H1的1#包，存储

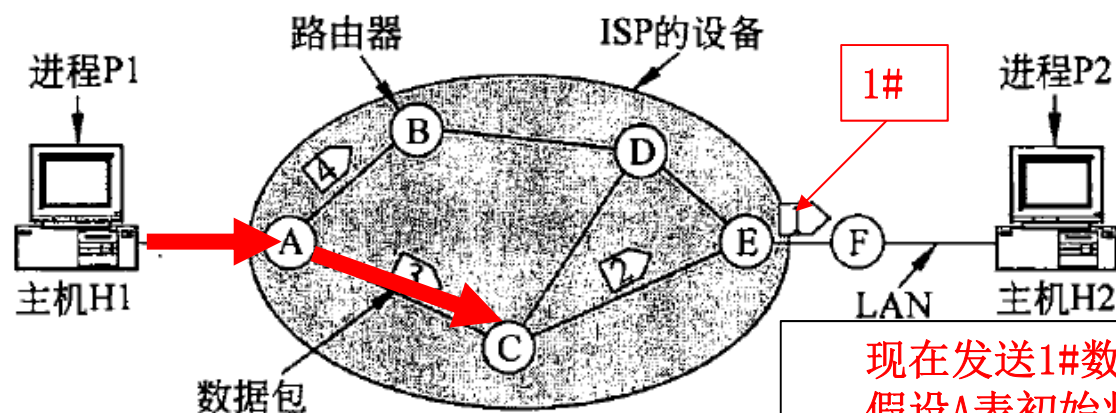
A的表(初始化)

A	—
B	B
C	C
D	B
E	C
F	C

} 直接
转发
} 间接
转发

图 5-2 数据报网络中的路由过程

★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器



现在发送1#数据包：

假设A表初始状态如左图

- 1、A收到H1的1#包，存储
- 2、A查询自己的路由表，得知F表有一个接口与1#包的目的地址(H2)在同一网段，因此向C转发

A的表(初始化)

A	—
B	B
C	C
D	B
E	C
F	C

} 直接
转发

} 间接
转发

图 5-2 数据报网络中的路由过程

★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器

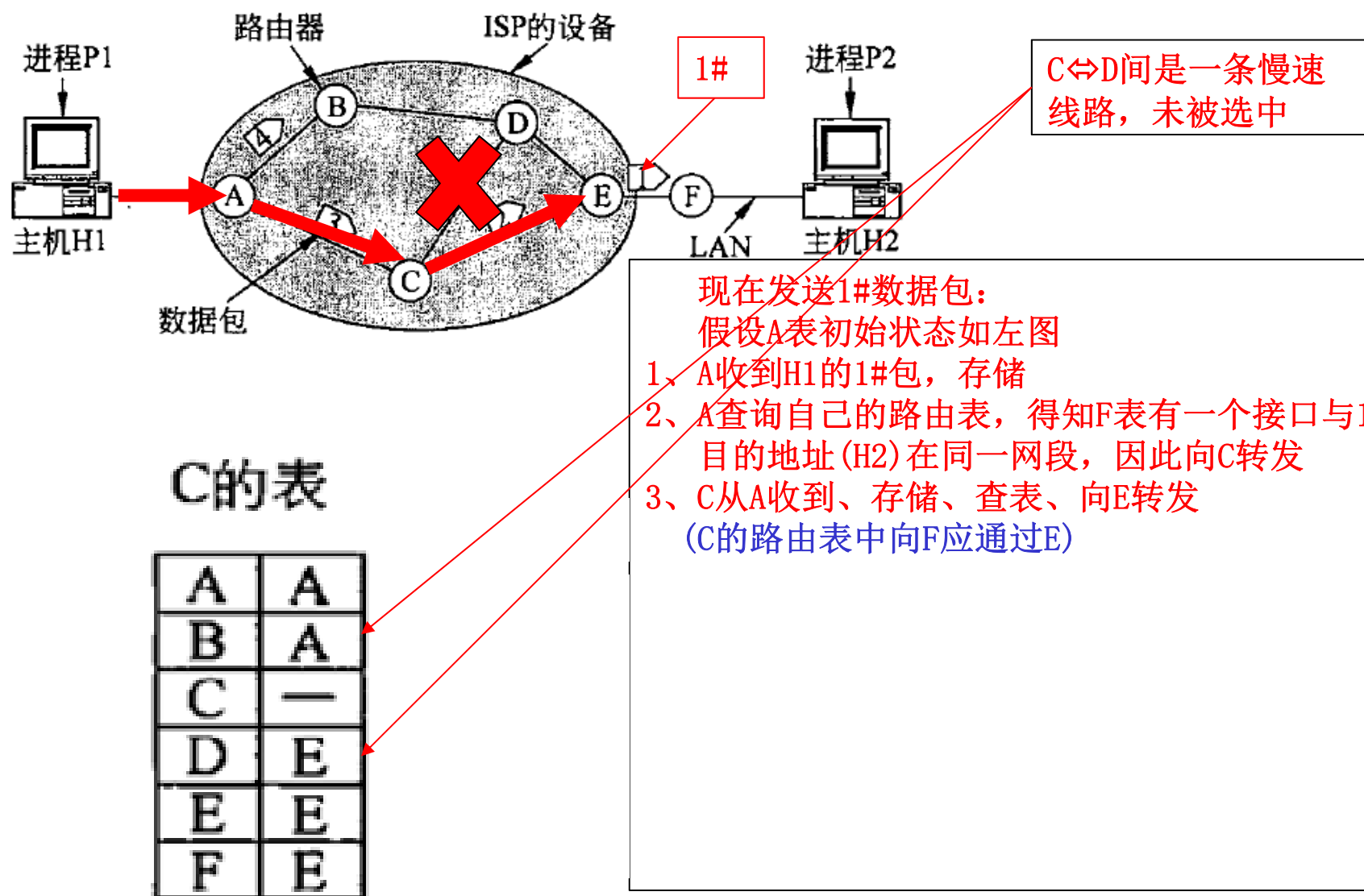
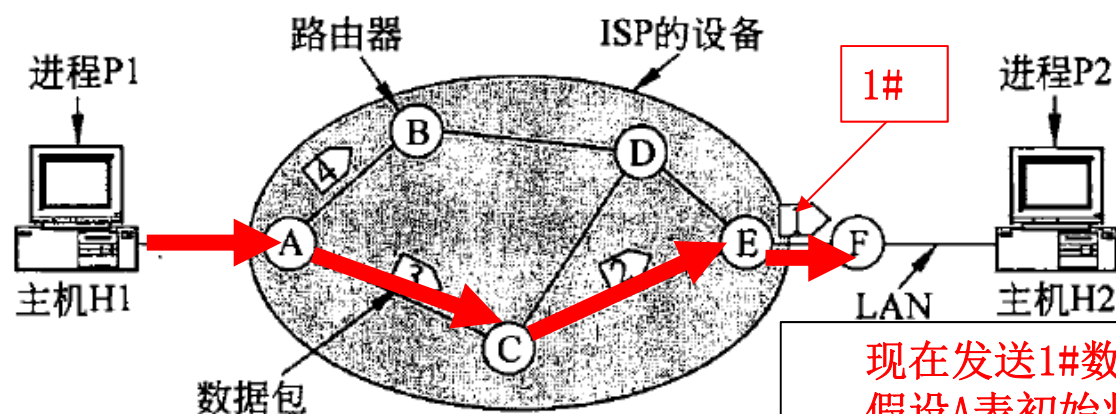


图 5-2 数据报网络中的路由过程

★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器



E的表

A	C
B	D
C	C
D	D
E	—
F	F

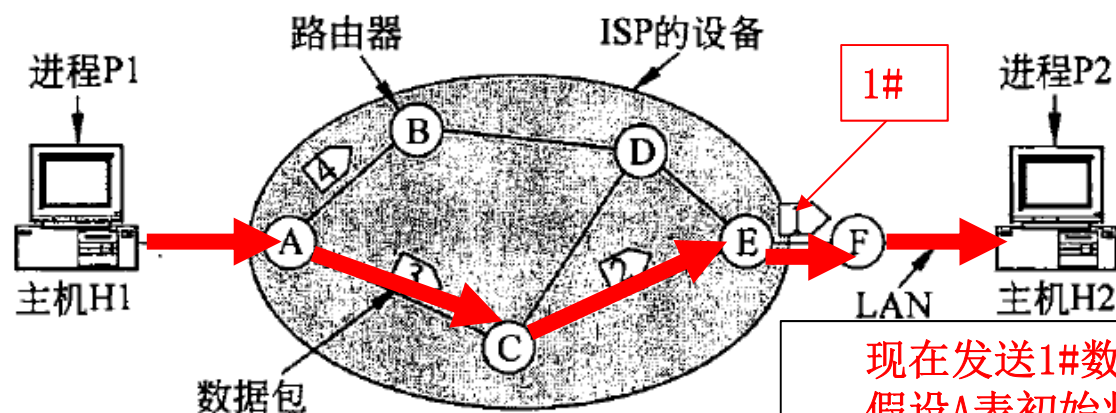
现在发送1#数据包：

假设A表初始状态如左图

- 1、A收到H1的1#包，存储
- 2、A查询自己的路由表，得知F表有一个接口与1#包的
目的地址(H2)在同一网段，因此向C转发
- 3、C从A收到、存储、查表、向E转发
- 4、E从C收到、存储、查表、向F转发
(E的路由表中向F应直接转发)

图 5-2 数据报网络中的路由过程

★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器



F的表

A	E
B	E
C	E
D	E
E	E
F	—

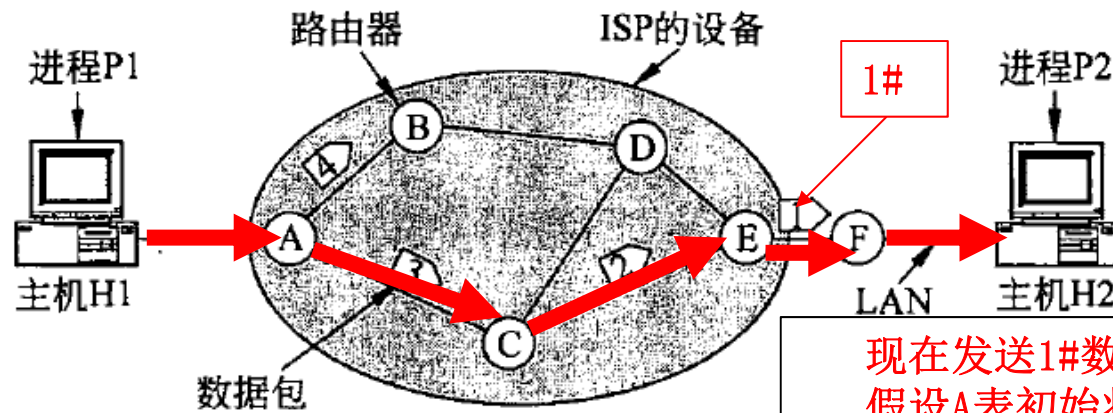
现在发送1#数据包：

假设A表初始状态如左图

- 1、A收到H1的1#包，存储
 - 2、A查询自己的路由表，得知F表有一个接口与1#包的目的地址(H2)在同一网段，因此向C转发
 - 3、C从A收到、存储、查表、向E转发
 - 4、E从C收到、存储、查表、向F转发
 - 5、F从E收到、存储、查表、向H2转发
- (F的路由表中向H2应直接转发)

图 5-2 数据报网络中的路由过程

★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器

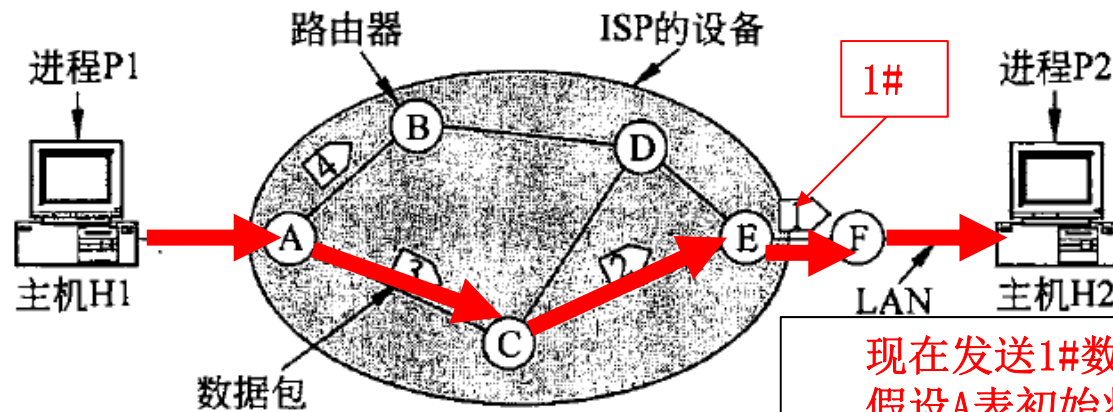


现在发送1#数据包：
假设A表初始状态如左图

1#包：

H1 → A → C → E → F → H2

★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器



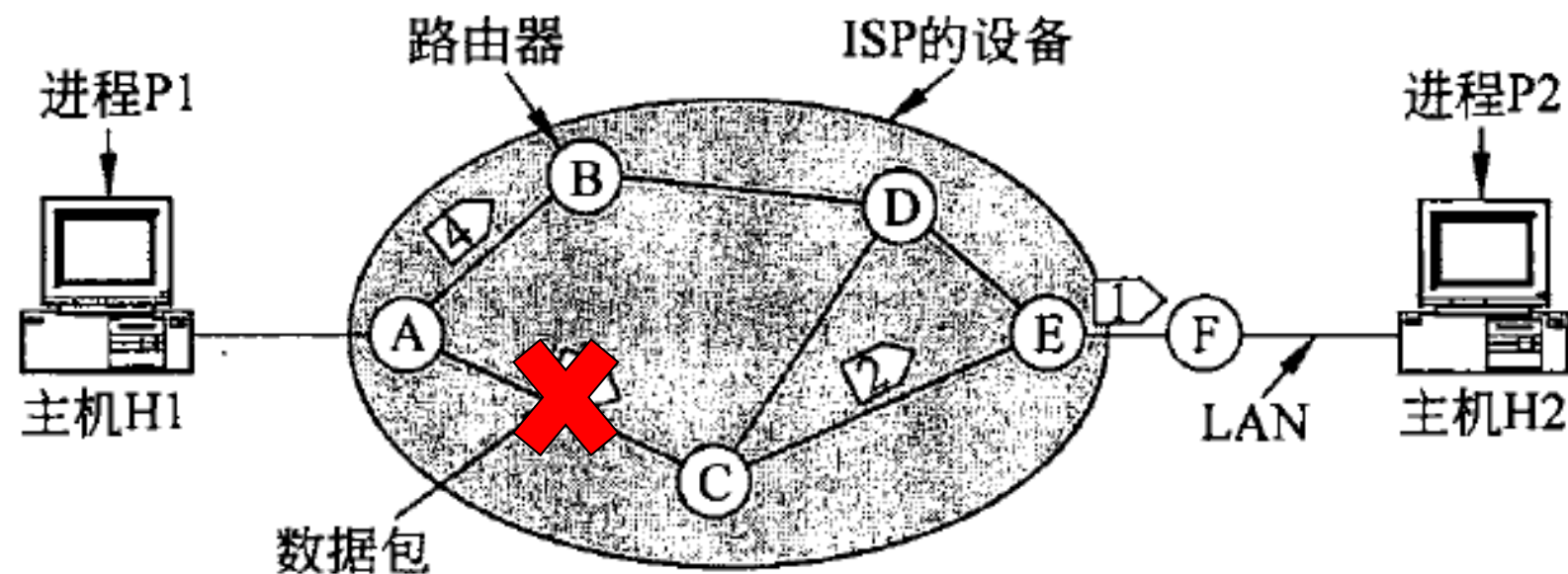
现在发送1#数据包：
假设A表初始状态如左图

1#包：

H1 → A → C → E → F → H2

假设2#/3#相同路径转发

★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器



A的表(初始化)

A	—
B	B
C	C
D	B
E	C
F	C

A的表(稍后)

A	—
B	B
C	C
D	B
E	B
F	B

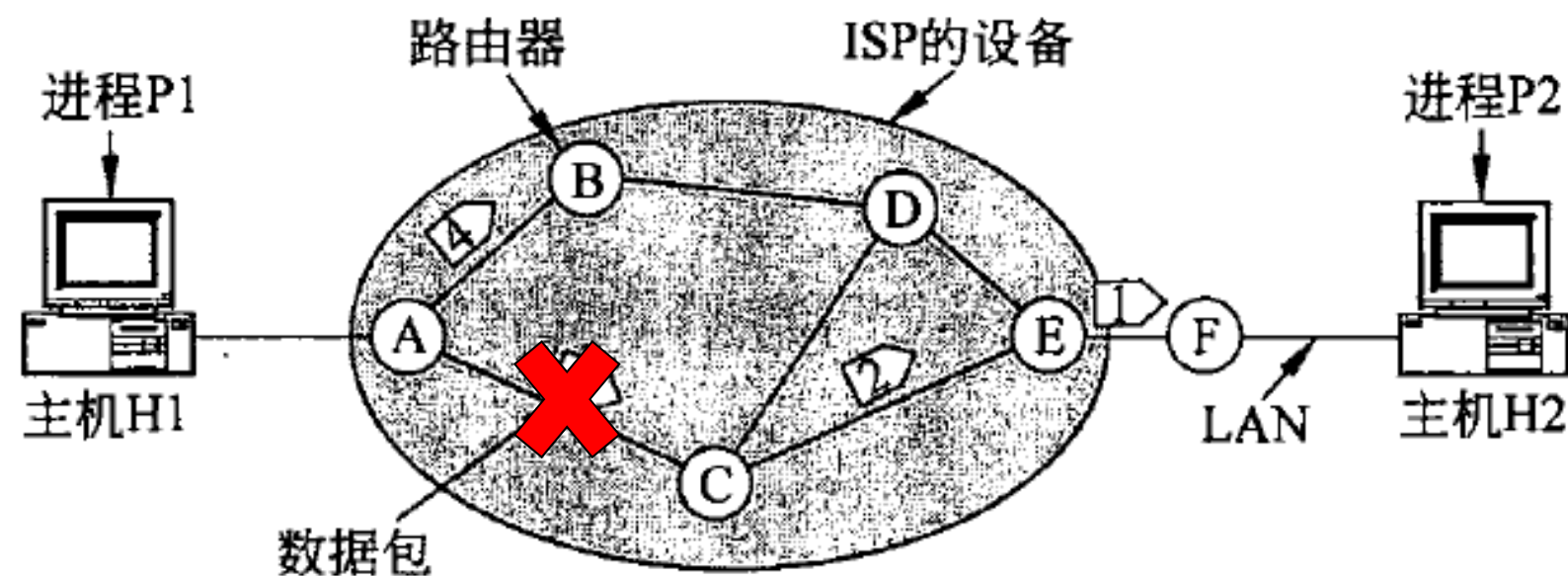
} 直接
转发

} 间接
转发

假设A⇔C间线路断开
或 A⇔C⇔E间拥塞

图 5-2 数据报网络中的路由过程

★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器



A的表(初始化)

A	—
B	B
C	C
D	B
E	C
F	C

A的表(稍后)

A	—
B	B
C	C
D	B
E	B
F	B

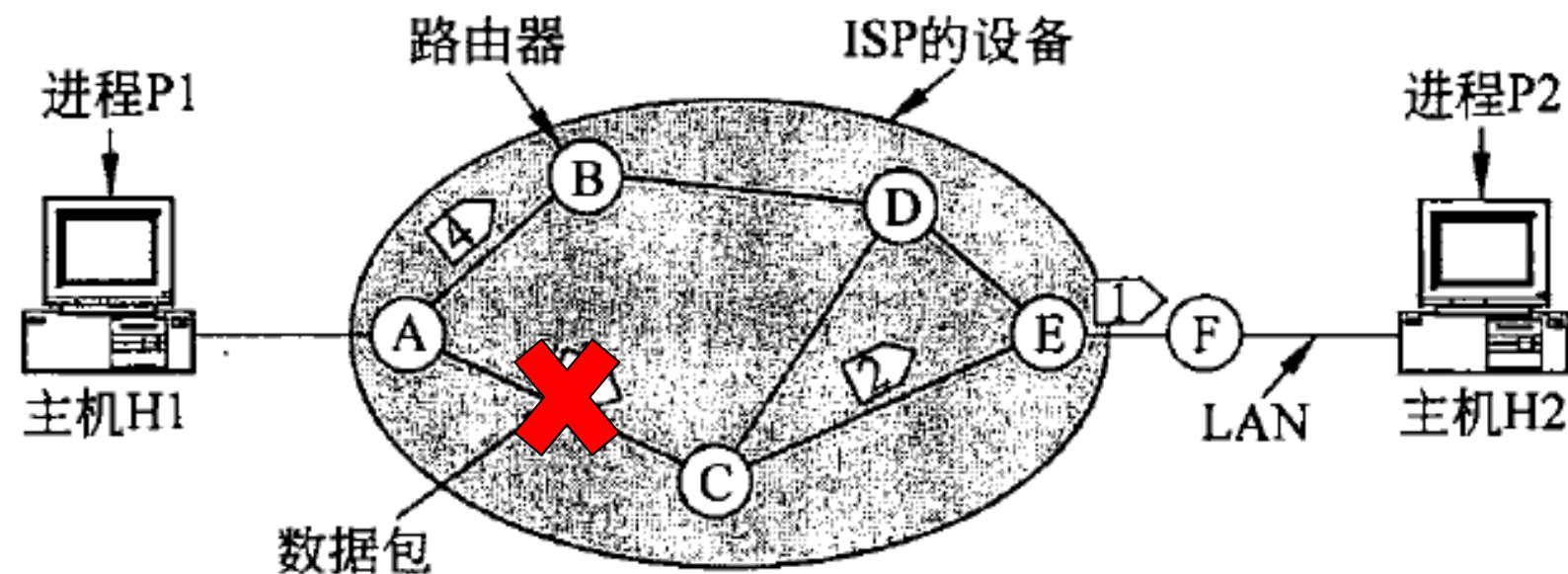
直接
转发

间接
转发

假设A↔C间线路断开
或 A↔C↔E间拥塞
A的路由表被更新

图 5-2 数据报网络中的路由过程

★ 例：设P1/P2为传输层进程，P1准备向P2发送的数据被分为4个网络层数据报发送，ABCDEF为路由器



A的表(初始化)

A	—
B	B
C	C
D	B
E	C
F	C

A的表(稍后)

A	—
B	B
C	C
D	B
E	B
F	B

直接
转发

间接
转发

假设A↔C间线路断开
或 A↔C↔E间拥塞
A的路由表被更新
此时发送4#包，则路径为：
H1→A→B→D→E→F→H2

图 5-2 数据报网络中的路由过程