



计 算 机 网 络

西北工业大学 软件学院

计算机网络

第4章 网络层：数据平面

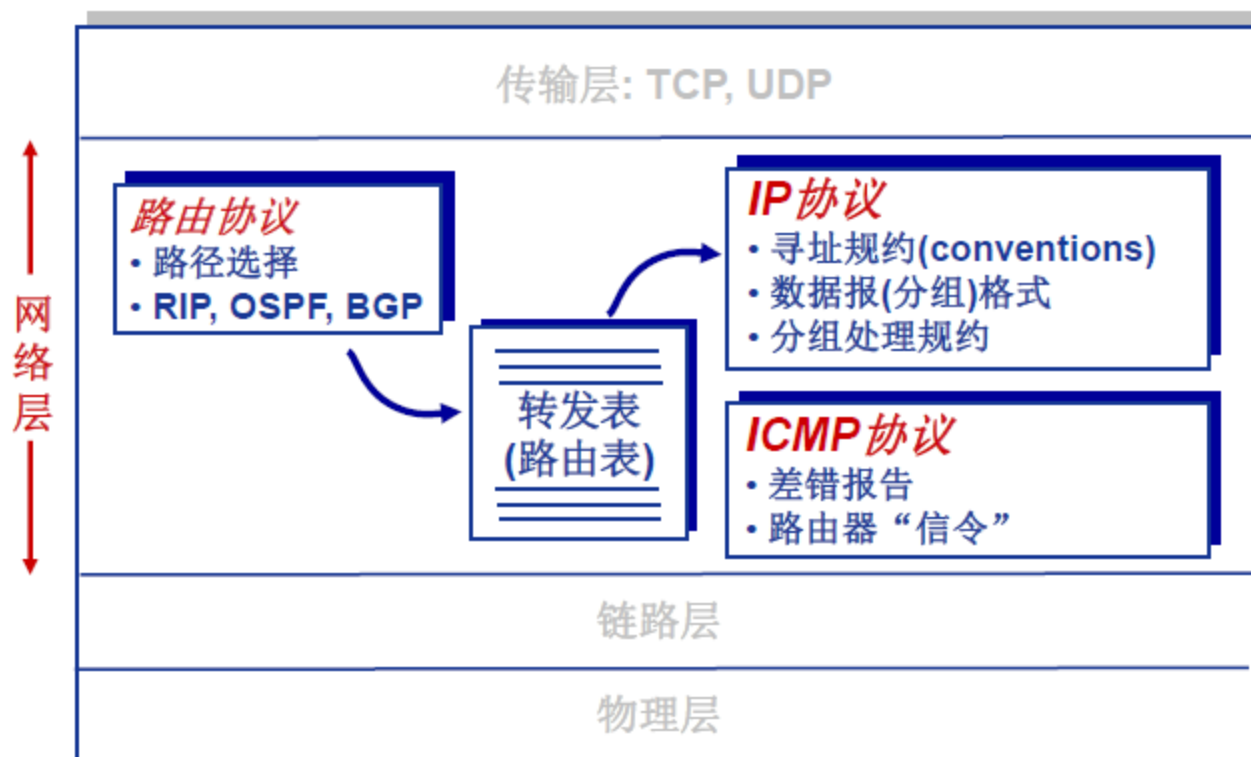
网络层


- 应用层:包含大量应用普遍需要的协议，支持网络应用
 - FTP, SMTP, HTTP
- 运输层: 主机到主机数据传输，负责从应用层接收消息，并传输应用层的message，到达目的后将消息上交应用。
 - TCP, UDP
- 网络层: 从源到目的地数据报的选路
 - IP, 选路协议
- 链路层: 在邻近网元之间传输数据
 - PPP, 以太网
- 物理层: 物理层负责将链路层帧中每一位(bit)从链路的一端传输到另一端。



网络层

主机、路由器网络层主要功能：





What is SDN? The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices

网络层服务模型

□ 无连接服务(connection-less service):

1. 不事先为系列分组的传输确定传输路径
2. 每个分组独立确定传输路径
3. 不同分组可能传输路径不同
4. 数据报网络(datagram network)

□ 连接服务(connection service):

1. 首先为系列分组的传输确定从源到目的经过的路径(建立连接)
2. 然后沿该路径 (连接) 传输系列分组
3. 系列分组传输路径相同
4. 传输结束后拆除连接
5. 虚电路网络(virtual-circuit network)

4.1 网络层提供的服务： 虚电路与数据报网络

4.1 两种服务

- 在计算机网络领域，网络层应该向运输层提供怎样的服务（“面向连接”还是“无连接”）曾引起了长期的争论。
- 争论焦点的实质就是：在计算机通信中，可靠交付应当由谁来负责？是网络还是端系统？

4.1 两种服务

1. 面向连接的通信方式

- 建立虚电路(Virtual Circuit)，以保证双方通信所需的一切网络资源。
- 如果再使用可靠传输的网络协议，就可使所发送的分组无差错按序到达终点。

4.1 两种服务

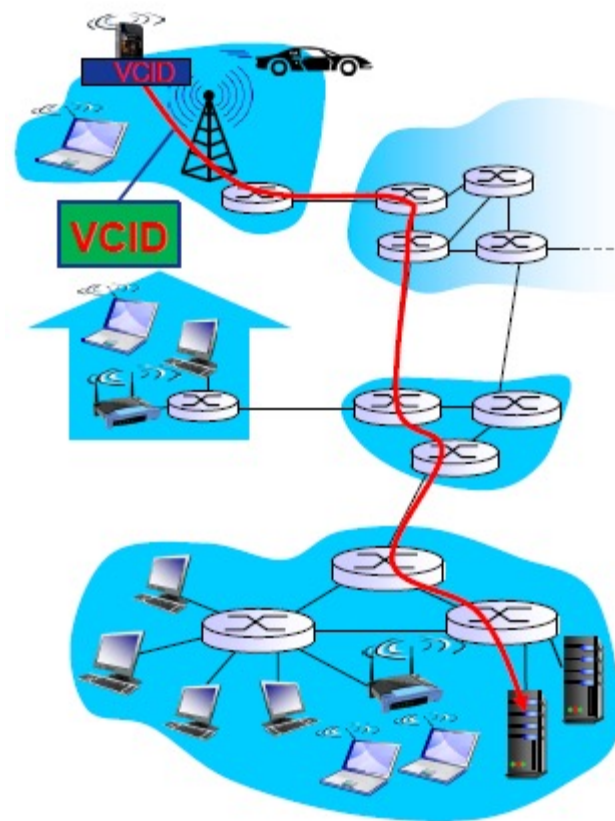
通信过程：

□ 呼叫建立(call setup)→数据传输→拆除呼叫

➤ 每个分组携带虚电路标识(VC ID)，而不是目的主机地址

➤ 虚电路经过的每个网络设备（如路由器），维护每条经过它的虚电路连接状态

➤ 链路、网络设备资源(如带宽、缓存等)可以面向VC进行预分配



4.1 两种服务

➤ 每条虚电路包括:

- 1.从源主机到目的主机的一条路径
- 2.虚电路号 (VCID) , 沿路每段链路一个编号
- 3.沿路每个网络层设备 (如路由器) , 利用转发表记录经过的每条虚电路

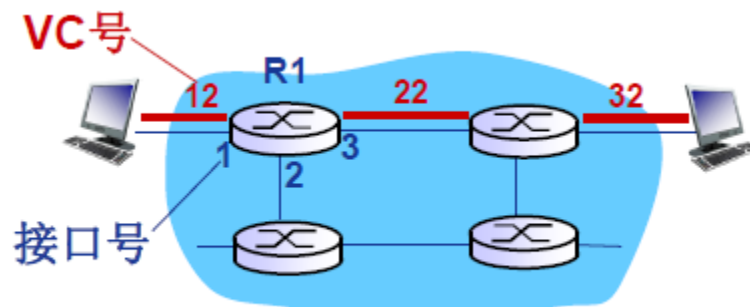
➤ 沿某条虚电路传输的分组, 携带对应虚电路的VCID, 而不是目的地址

➤ 同一条VC, 在每段链路上的VCID通常不同

➤ □ 路由器转发分组时依据转发表改写/替换虚电路号

4.1 两种服务

虚电路转发表



路由器R1的VC转发表:

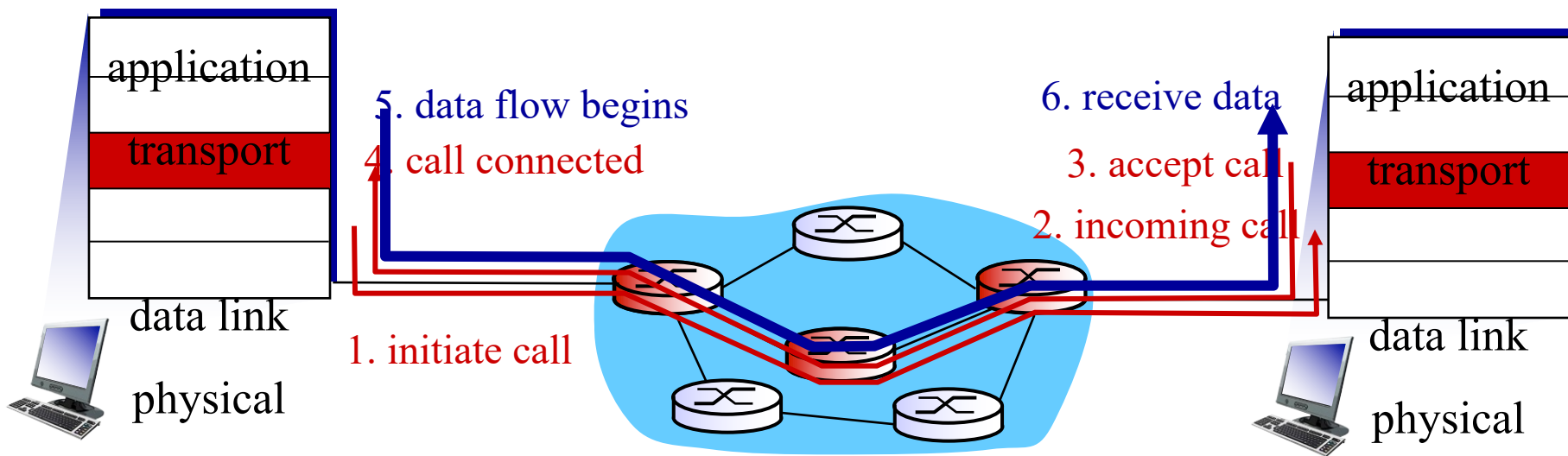
| 输入接口 | 输入VC # | 输出接口 | 输出VC # |
|------|--------|------|--------|
| 1 | 12 | 3 | 22 |
| 2 | 63 | 1 | 18 |
| 3 | 7 | 2 | 17 |
| 1 | 97 | 3 | 87 |
| ... | ... | ... | ... |

VC路径上每个路由器都需要维护VC连接的状态信息！

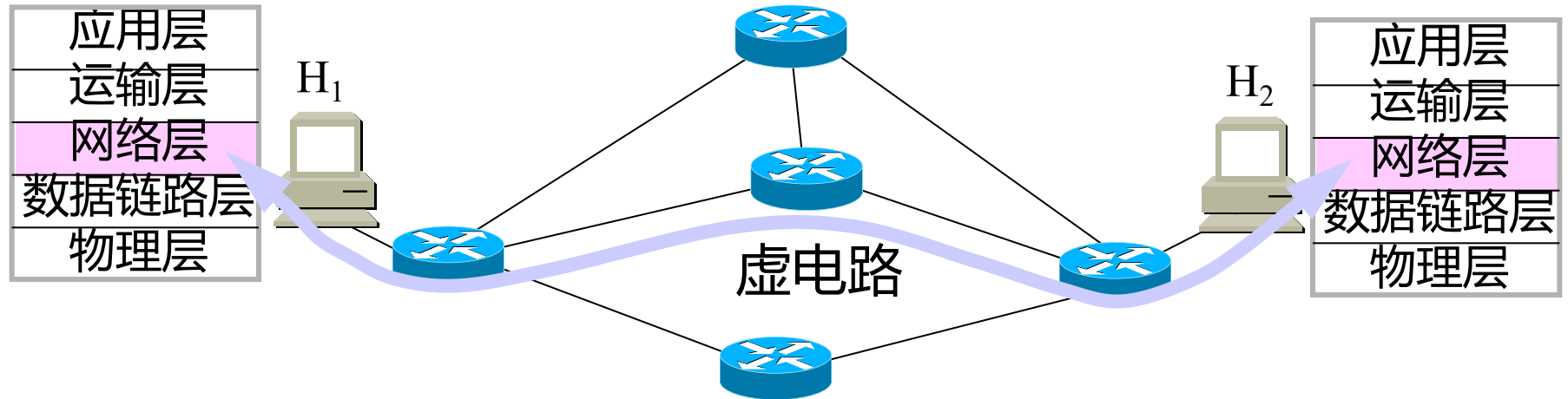
4.1 两种服务

虚电路转发表

- 用于VC的建立、维护与拆除
- 如ATM、帧中继(frame-relay)网络等
- 目前的Internet不采用



4.1 两种服务



H₁ 发送给 H₂ 的所有分组都沿着同一条虚电路传送

4.1 两种服务

- 虚电路表示这只是一条**逻辑上的连接**，分组都沿着这条逻辑连接按照存储转发方式传送，而并不是真正建立了一条物理连接。
- 请注意，电路交换的电话通信是先建立了一条**真正的连接**。因此分组交换的虚连接和电路交换的连接只是类似，但并不完全一样。

4.1 两种服务

2. 网络层向上只提供简单灵活的、无连接的、尽最大努力交付的数据报服务。

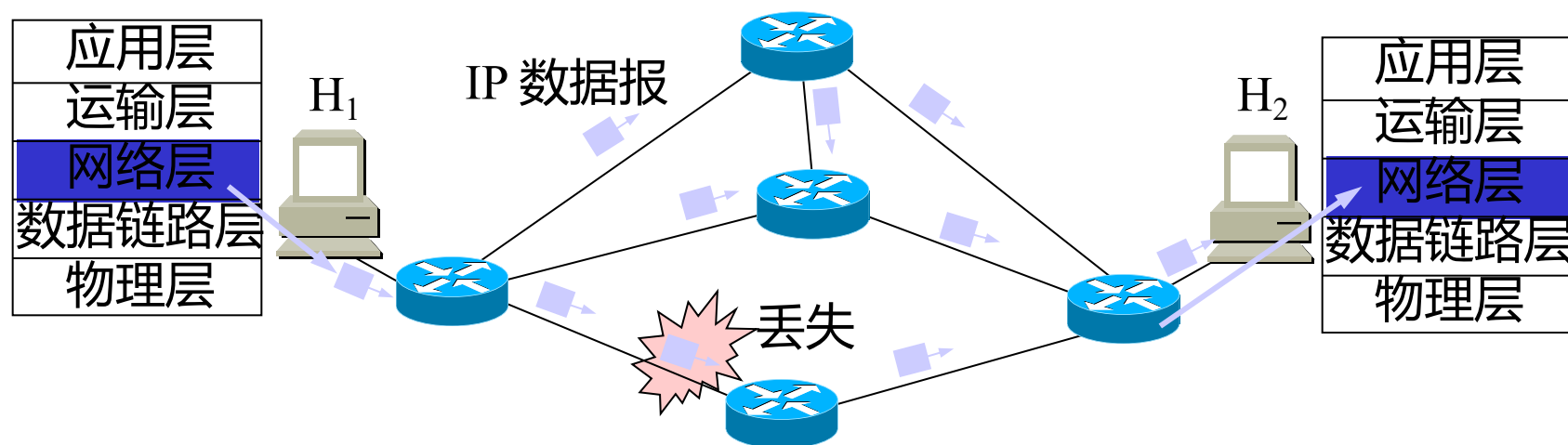
➤网络在发送分组时不需要先建立连接。每一个分组（即 IP 数据报）独立发送，与其前后的分组无关（不进行编号）。

➤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。

4.1 两种服务

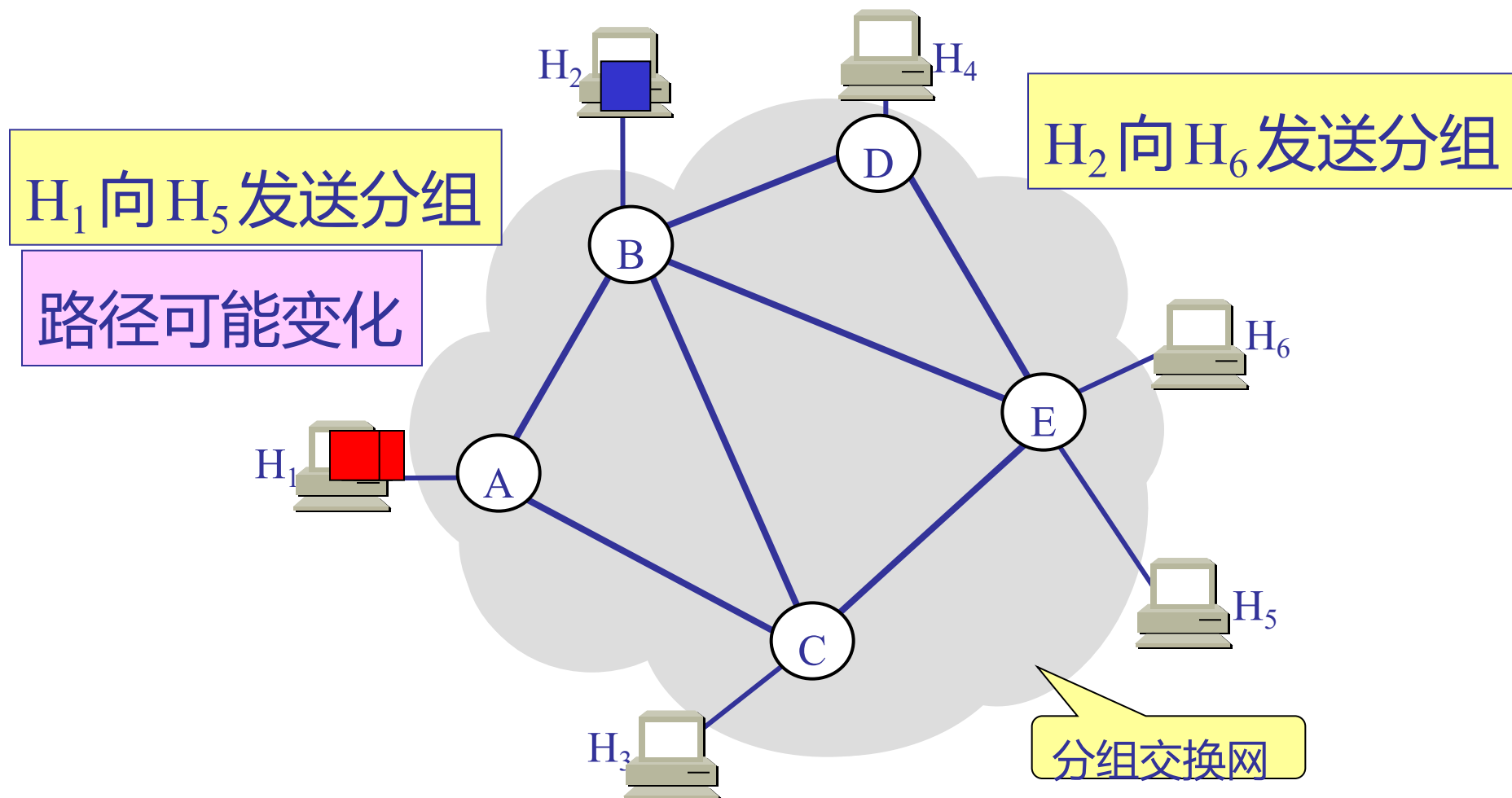
- 由于传输网络不提供端到端的可靠传输服务，这就使网络中的路由器可以做得比较简单，而且价格低廉（与电信网的交换机相比较）。
- 如果主机（即端系统）中的进程之间的通信要是可靠的，那么就由网络的主机中的运输层负责（包括差错处理、流量控制等）。
- 采用这种设计思路的好处是：网络的造价大大降低，运行方式灵活，能够适应多种应用。
- 因特网能够发展到今日的规模，充分证明了当初采用这种设计思路的正确性。

4.1 两种服务



H_1 发送给 H_2 的分组可能沿着不同路径传送

4.1 两种服务



4.1 两种服务

| 对比的方面 | 虚电路服务 | 数据报服务 |
|---------------|-------------------------|---------------------------|
| 思路 | 可靠通信应当由网络来保证 | 可靠通信应当由用户主机来保证 |
| 连接的建立 | 必须有 | 不需要 |
| 终点地址 | 仅在连接建立阶段使用，每个分组使用短的虚电路号 | 每个分组都有终点的完整地址 |
| 分组的转发 | 属于同一条虚电路的分组均按照同一路由进行转发 | 每个分组独立选择路由进行转发 |
| 当结点出故障时 | 所有通过出故障的结点的虚电路均不能工作 | 出故障的结点可能会丢失分组，一些路由可能会发生变化 |
| 分组的顺序 | 总是按发送顺序到达终点 | 到达终点时不一定按发送顺序 |
| 端到端的差错处理和流量控制 | 可以由网络负责，也可以由用户主机负责 | 由用户主机负责 |

需要解决的问题

VC是通过VCID来建立转发表，那么数据报网络如何建立转发表？

- 1.路由的工作原理。
- 2.IP地址的产生、获取与演进。
- 3.数据报网络中的消息传输原理。

4.2 路由器原理

网络互相连接起来要使用一些中间设备
中间设备又称为中间系统或中继(relay)系统。

- ◆物理层中继系统：转发器(repeater)。
- ◆数据链路层中继系统：网桥或桥接器(bridge)。
- ◆网络层中继系统：路由器(router)。
- ◆网桥和路由器的混合物：桥路器(brouter)。
- ◆网络层以上的中继系统：网关(gateway)。

4.2 路由器原理

- **路由器**是一种具有多个输入端口和多个输出端口的专用计算机，其任务是转发分组。也就是说，将路由器某个输入端口收到的分组，按照分组要去的目的地（即目的网络），将该分组从路由器的某个合适的输出端口转发给下一跳路由器。
- 下一跳路由器也按照这种方法处理分组，直到该分组到达终点为止。

4.2 路由器原理



思科(Cisco)公司创始人

列昂纳德·波萨克 和 桑德拉·勒纳

4.2 路由器原理

CASE HISTORY

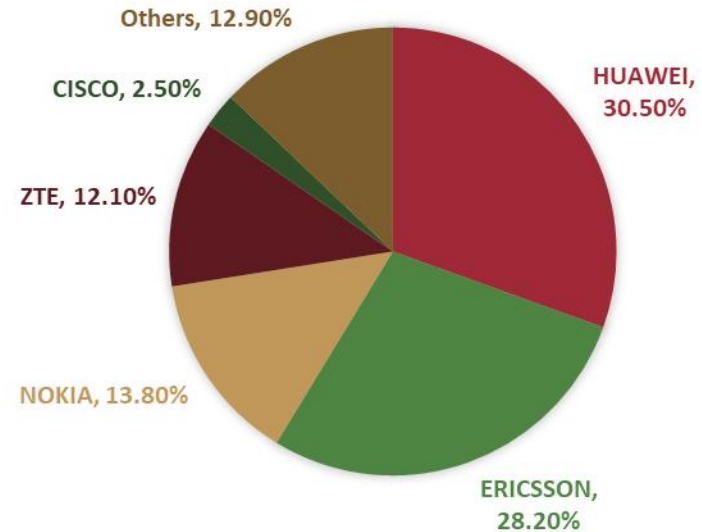
CISCO SYSTEMS **DOMINATING** THE NETWORK CORE

As of this writing 2012, Cisco employs more than 65,000 people. How did this gorilla of a networking company come to be? It all started in 1984 in the living room of a Silicon Valley apartment.

Len Bosak and his wife Sandy Lerner were working at Stanford University when they had the idea to build and sell Internet routers to research and academic institutions, the primary adopters of the Internet at that time. Sandy Lerner came up with the name Cisco (an abbreviation for San Francisco), and she also designed the company's bridge logo. Corporate headquarters was their living room, and they financed the project with credit cards and moonlighting consulting jobs. At the end of 1986, Cisco's revenues reached \$250,000 a month. At the end of 1987, Cisco succeeded in attracting venture capital—\$2 million from Sequoia Capital in exchange for one-third of the company. Over the next few years, Cisco continued to grow and grab more and more market share. At the same time, relations between Bosak/Lerner and Cisco management became strained. Cisco went public in 1990; in the same year Lerner and Bosak left the company.

Over the years, Cisco has expanded well beyond the router market, selling security, wireless caching, Ethernet switch, datacenter infrastructure, video conferencing, and voice-over IP products and services. However, Cisco is facing increased international competition, including from Huawei, a rapidly growing Chinese network-gear company. Other sources of competition for Cisco in the router and switched Ethernet space include Alcatel-Lucent and Juniper.

全球移动核心网出货量市场份额

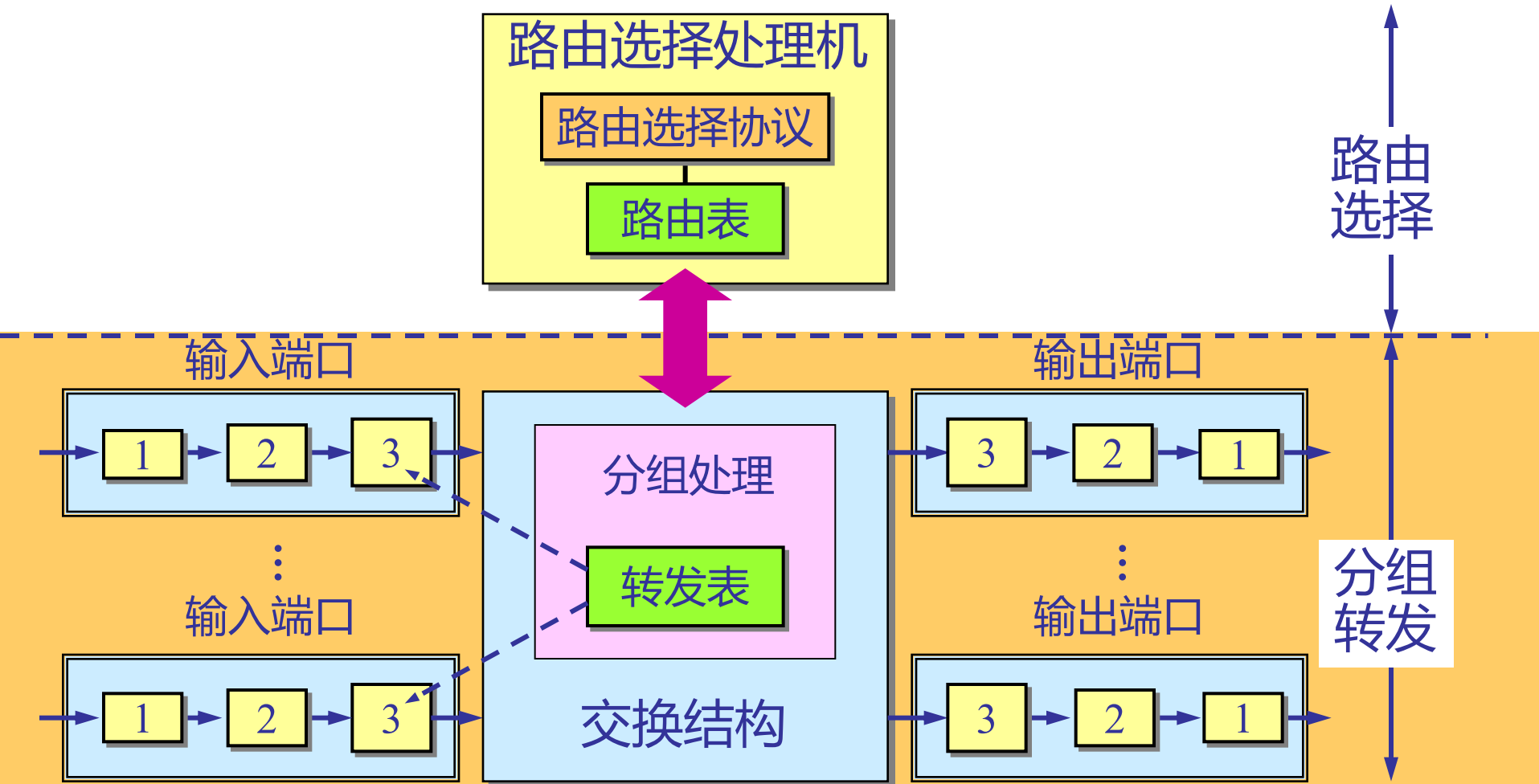


本书第七版（英文）已经删除了Case History

4.2 路由器原理



4.2 路由器原理



4.2 路由器原理

转发与路由

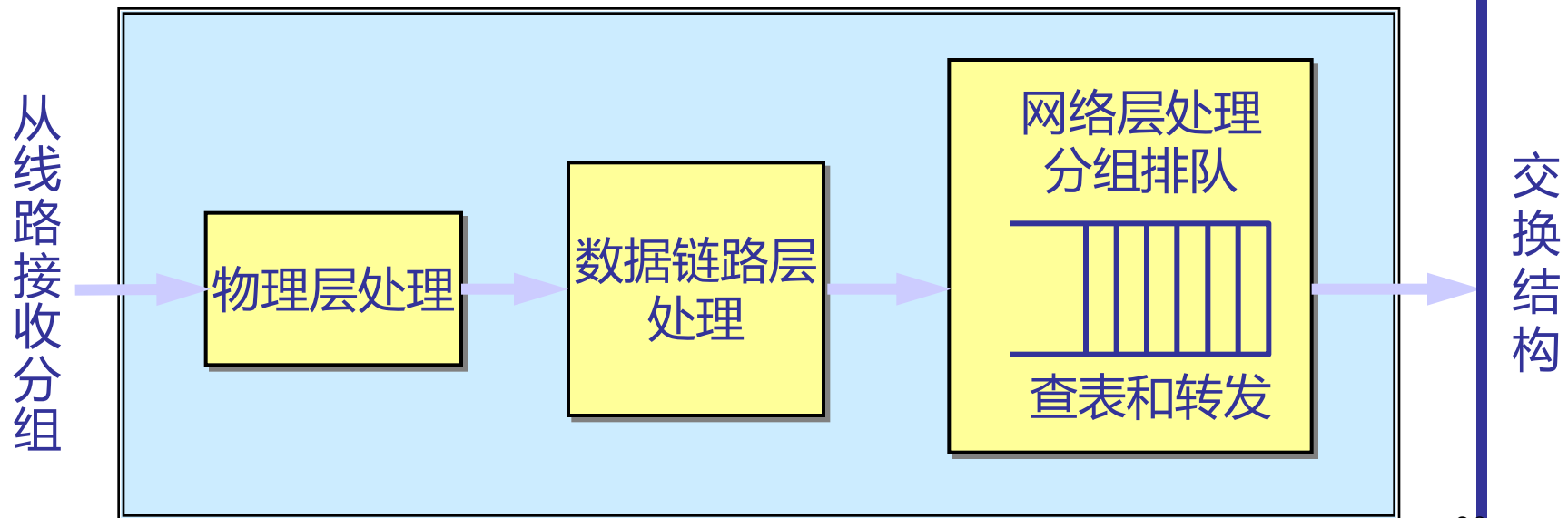
- “**转发**” (forwarding)就是路由器根据转发表将用户的 IP 数据报从合适的端口转发出去。
- “**路由选择**” (routing)则是按照分布式算法，根据从各相邻路由器得到的关于网络拓扑的变化情况，动态地改变所选择的路由。
- 路由表是根据路由选择算法得出的。而转发表是从路由表得出的。
- 在讨论路由选择的原理时，往往不去区分转发表和路由表的区别，

4.2 路由器原理

输入端口对线路上收到的分组的处理

- 数据链路层剥去帧首部和尾部后，将分组送到网络层的队列中排队等待处理。这会产生一定的时延。

输入端口的处理

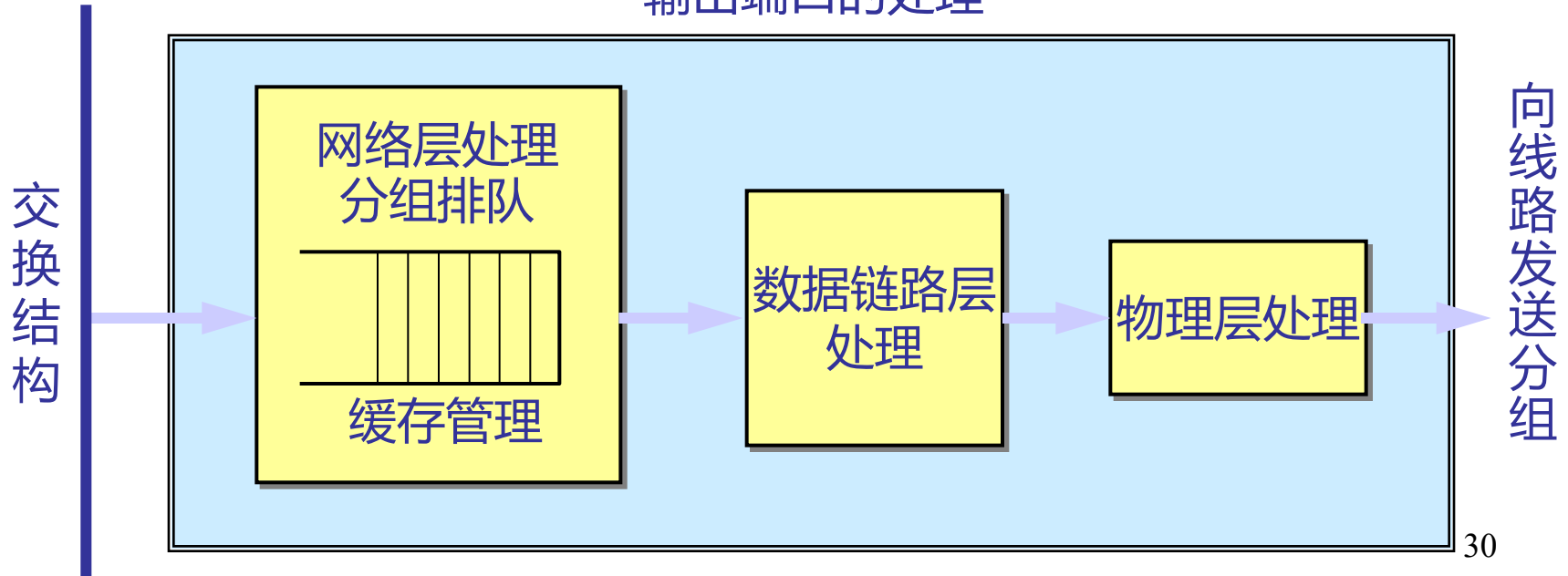


4.2 路由器原理

输出端口将交换结构传送来的分组发送到线路

- 当交换结构传送过来的分组先进行缓存。数据链路层处理模块将分组加上链路层的首部和尾部，交给物理层后发送到外部线路。

输出端口的处理

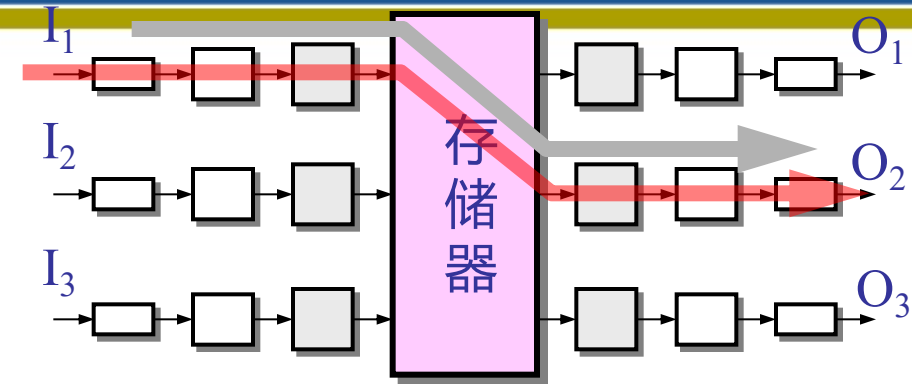


4.2 路由器原理

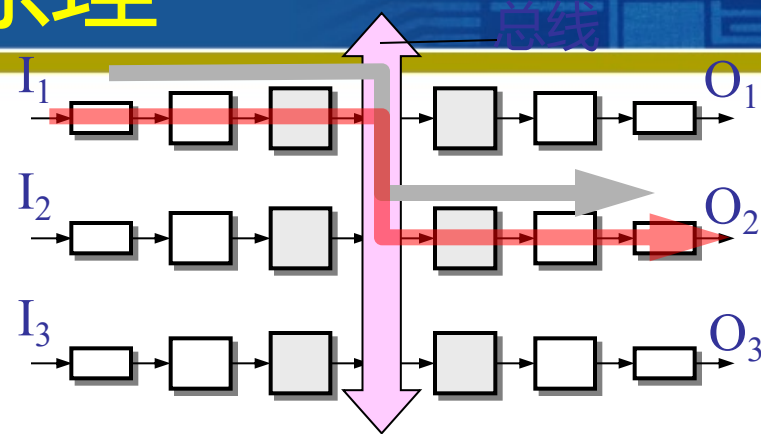
分组丢弃

- 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。
- 路由器中的输入或输出队列产生溢出是造成分组丢失的重要原因。

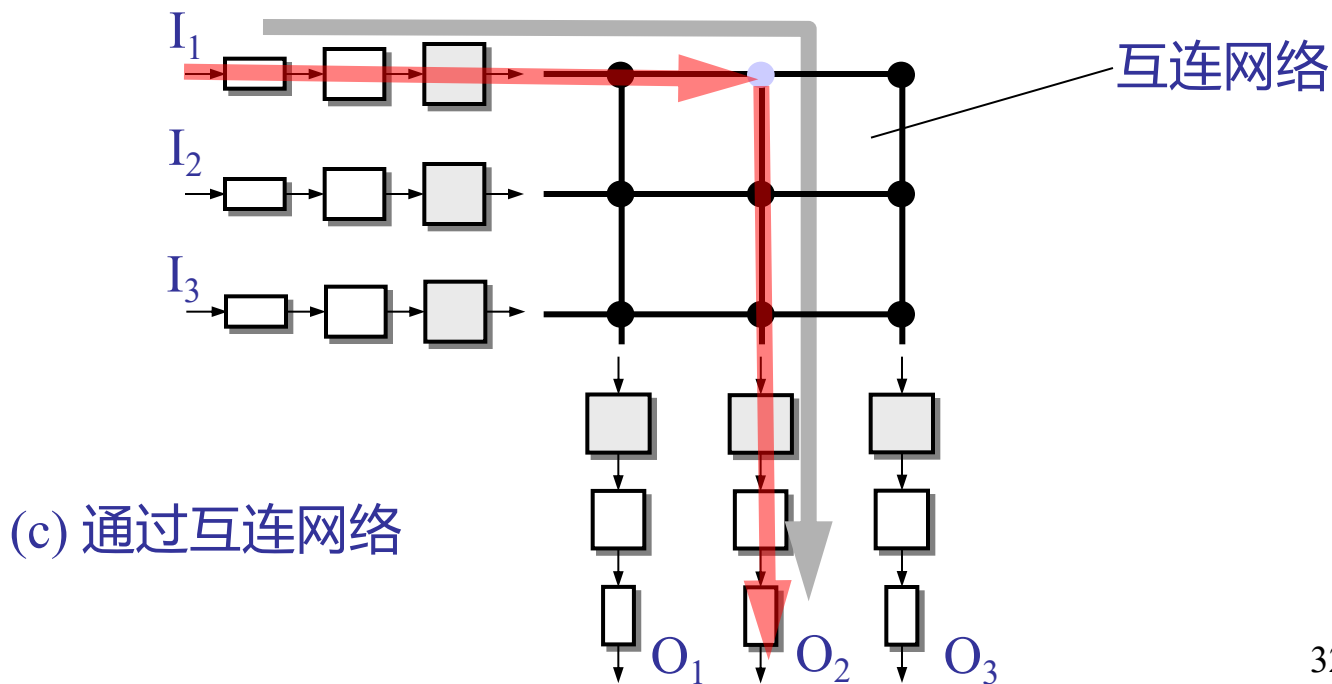
4.2 路由器原理



(a) 通过存储器



(b) 通过总线

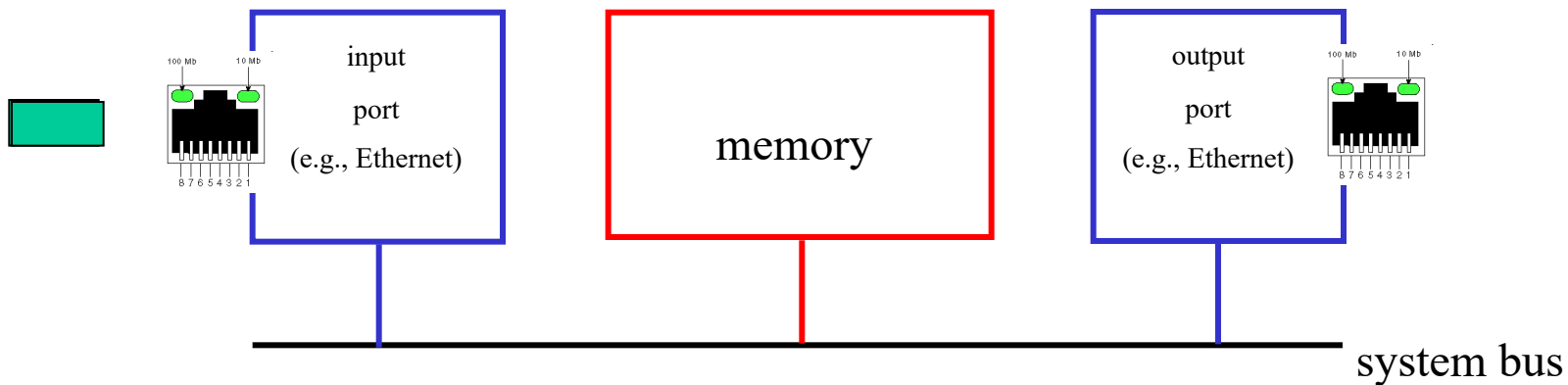


(c) 通过互连网络

4.2 路由器原理

通过存储:

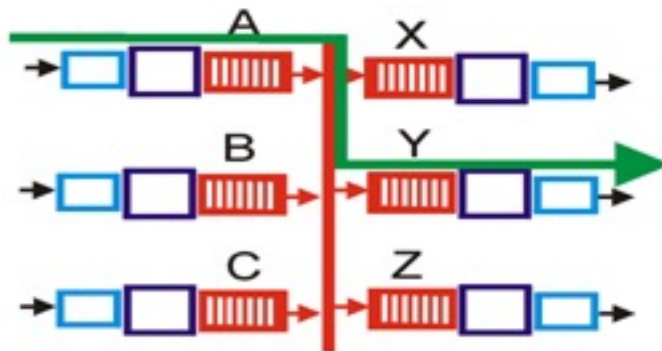
- 具有交换功能的传统计算机，在CPU的直接控制下
- 分组拷贝到系统的内存
- 速率受内存带宽限制(每数据报跨越两次总线)



4.2 路由器原理

通过总线

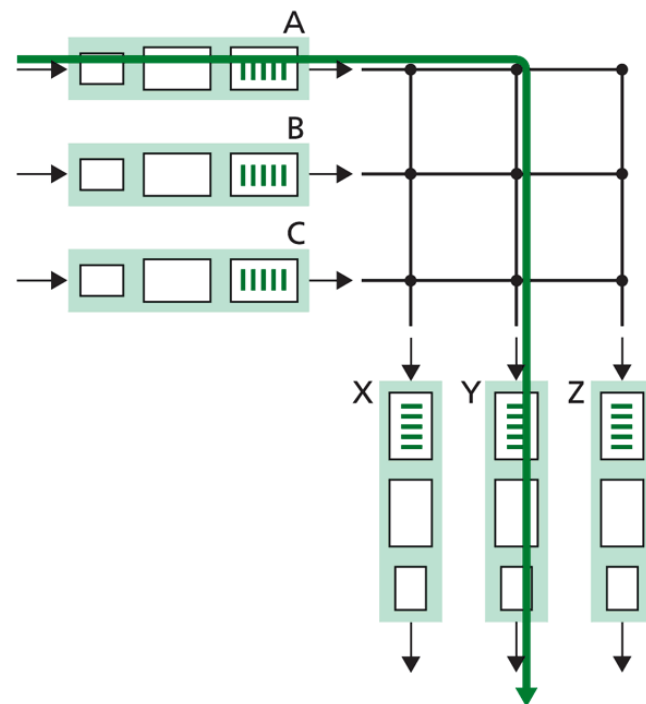
- 数据报从输入端口到输出端口内存经一个共享的总线（总线芯片），总线速度快于内存读取速度
 - 总线竞争: 任何时刻，总线仅能连通1个输入和1个输出
- 。 Cisco5600：数据32Gbps总线，对于小型接入网和企业网其交换速度通常是足够的



4.2 路由器原理

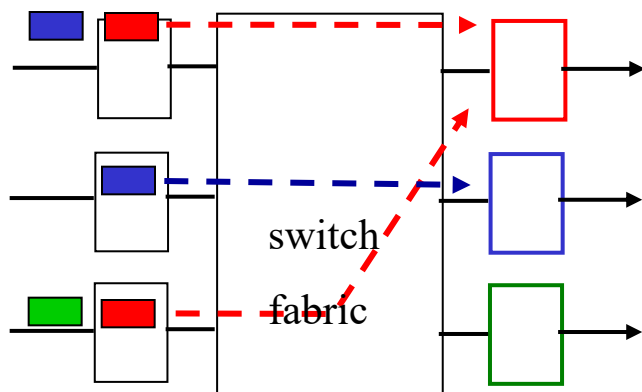
通过互联的网络

- 克服了总线带宽限制
- Crossbar一般同时满足多个输入和输出连通
- 一般是路由交换机
- Cisco 12000: 通过互联网络交换提供60Gbps

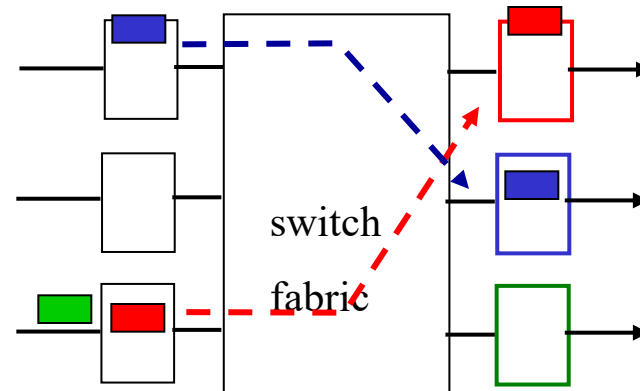


crossbar
switching fabric

4.2 路由器原理



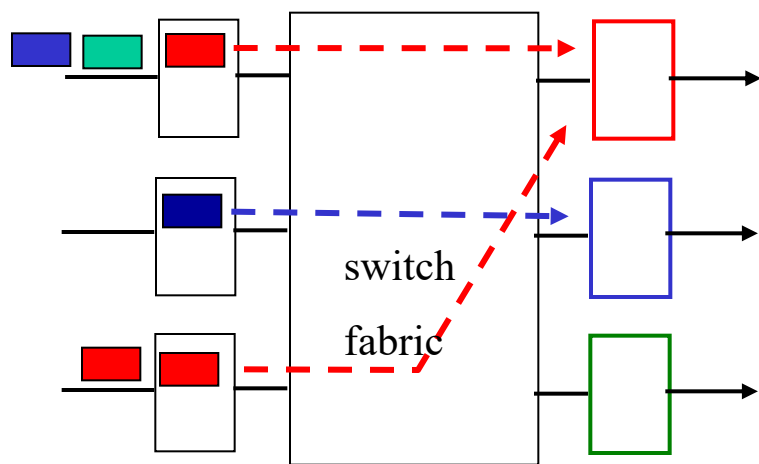
output port contention:
only one red datagram can be
transferred.
lower red packet is blocked



one packet time later:
green packet
experiences HOL
blocking

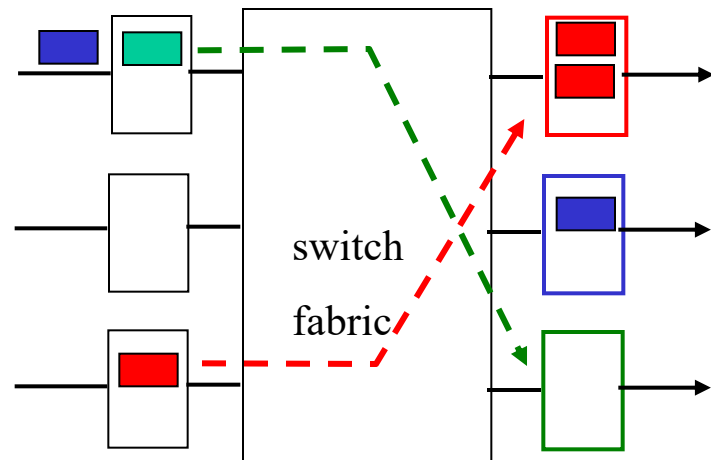
- 当交换结构的处理速度小于输入端口数据流入速度时，在输入端口就会出现排队等待当输入端口缓存溢出时，就会出现丢包
- 队首阻塞（HOL）：处在队首的分组阻碍了队列其他分组的转发

4.2 路由器原理



packets move

from input to output



one packet time later

输出端口排队

- 当交换速率大于输出端口链路速率时就需要缓存
- 当输出端口缓存溢出时，就会出现丢包

4.2 路由器原理

需要多大的缓存？

- RFC 3439规则：缓存大小=平均RTT*链路容量C（这个结果是基于相对少量的TCP流的排队动态分析得到的）
- 目前推荐的规则：当有大量的TCP流（N）流过一条链路时，缓存大小

$$= \frac{RTT \times C}{\sqrt{N}}$$

4.2 路由器原理

产生排队进而导致丢包产生的原因

- 交换速率和端口接收/发送数据的速率不匹配；
- 端口产生了竞争

解决问题的思路之一是调度

4.2 路由器原理

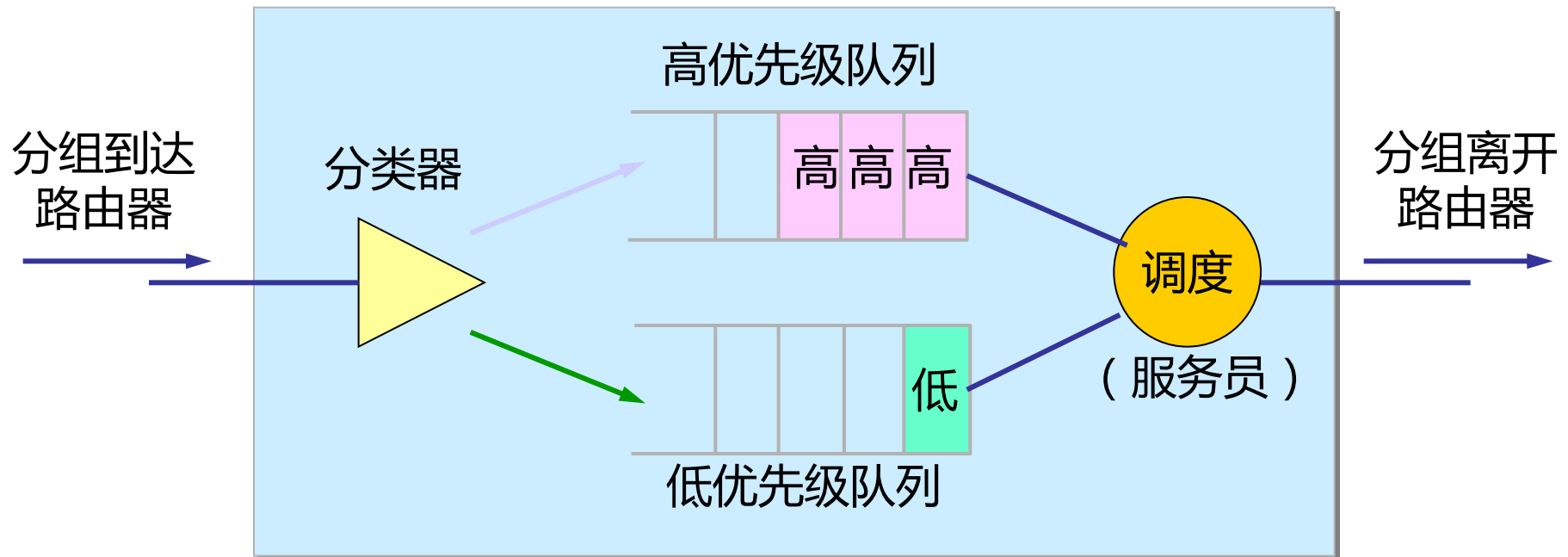
“调度”就是指排队的规则。

- 如不采用专门的调度机制，则默认排队规则就是**先进先出** FIFO (First In First Out)。当队列已满时，后到达的分组就被丢弃。
- 先进先出的最大缺点就是不能区分时间敏感分组和一般数据分组，并且也不公平。
- 在先进先出的基础上增加按**优先级排队**，就能使优先级高的分组优先得到服务。

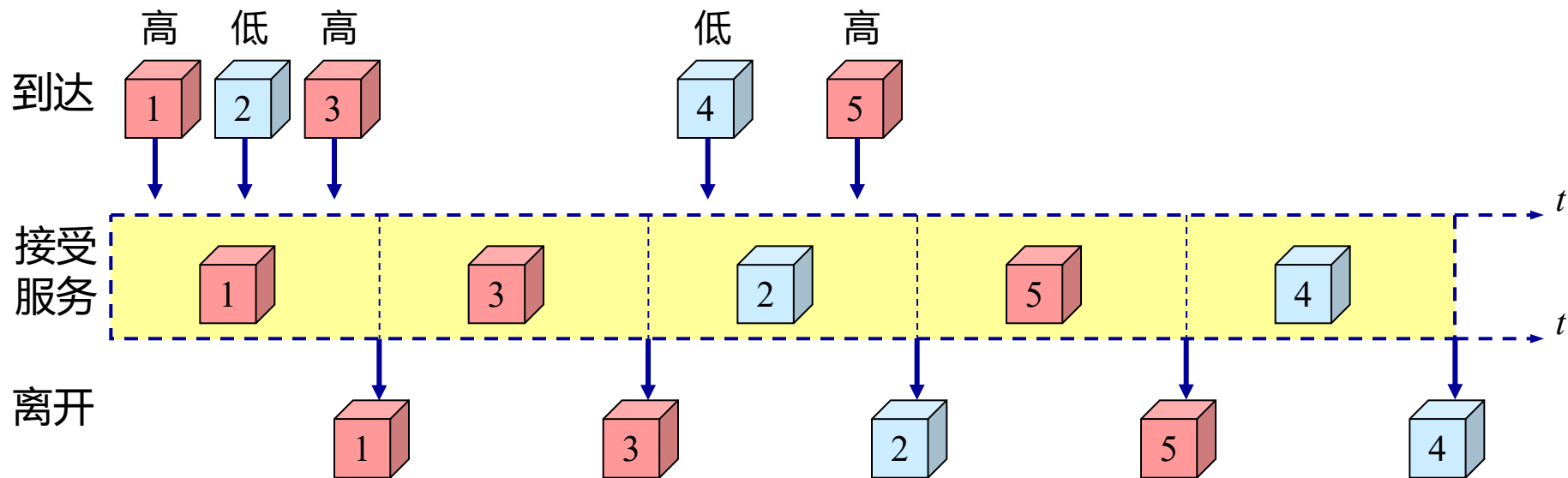
4.2 路由器原理

优先权排队

路由器



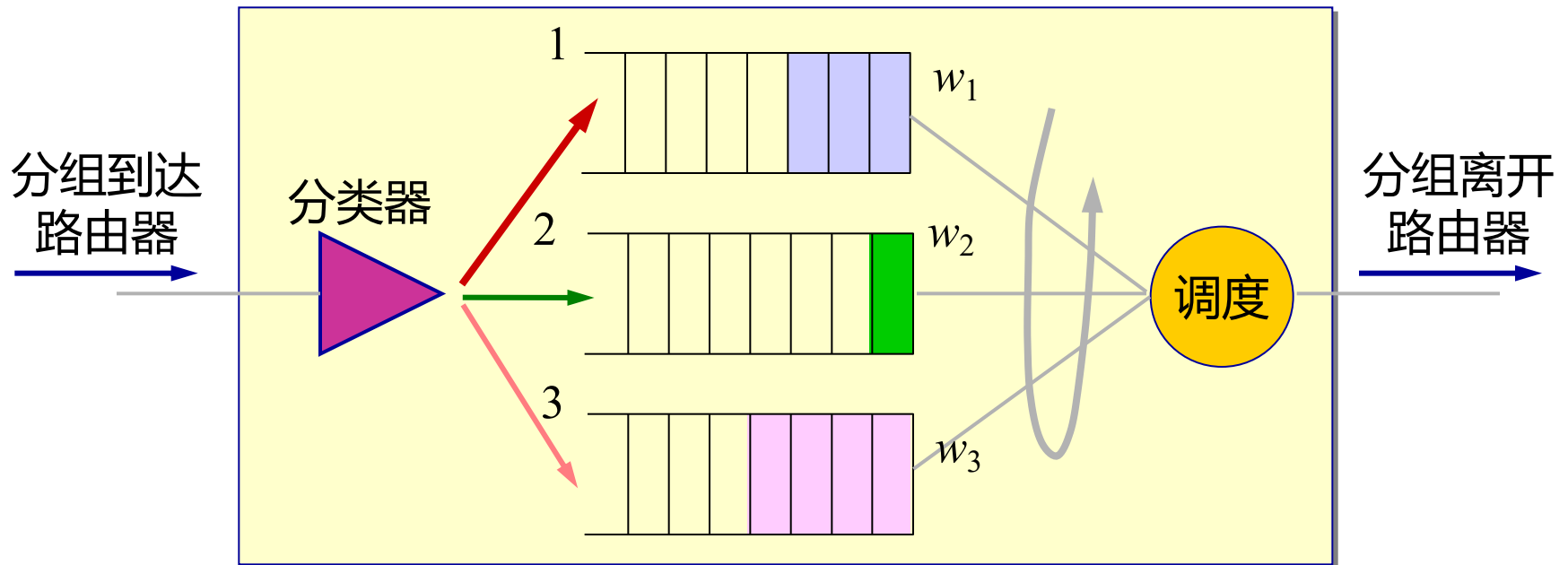
4.2 路由器原理



4.2 路由器原理

加权公平排队 (WFQ)

路由器



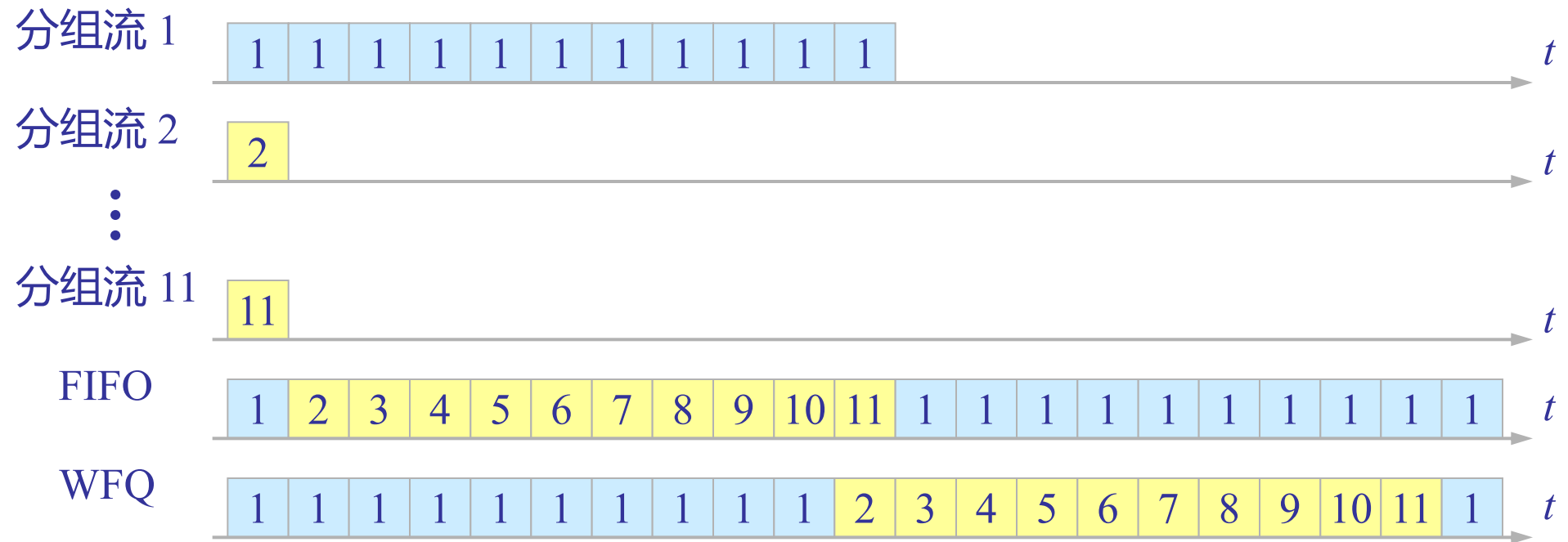
4.2 路由器原理

- 分组到达后就将分组进行分类，然后送交与其类别对应的队列。队列按顺序依次将队首的分组发送到链路。遇到队列空就跳过去。
- 给队列 i 指派一个权重 w_i 。队列 i 得到的平均服务时间为 $w_i / (\sum w_j)$ ，这里 $\sum w_j$ 是对所有的非空队列的权重求和。
- 队列 i 将得到的有保证的带宽 R_i 应为

$$R_i = \frac{R \times w_i}{\sum w_j}$$

4.2 路由器原理

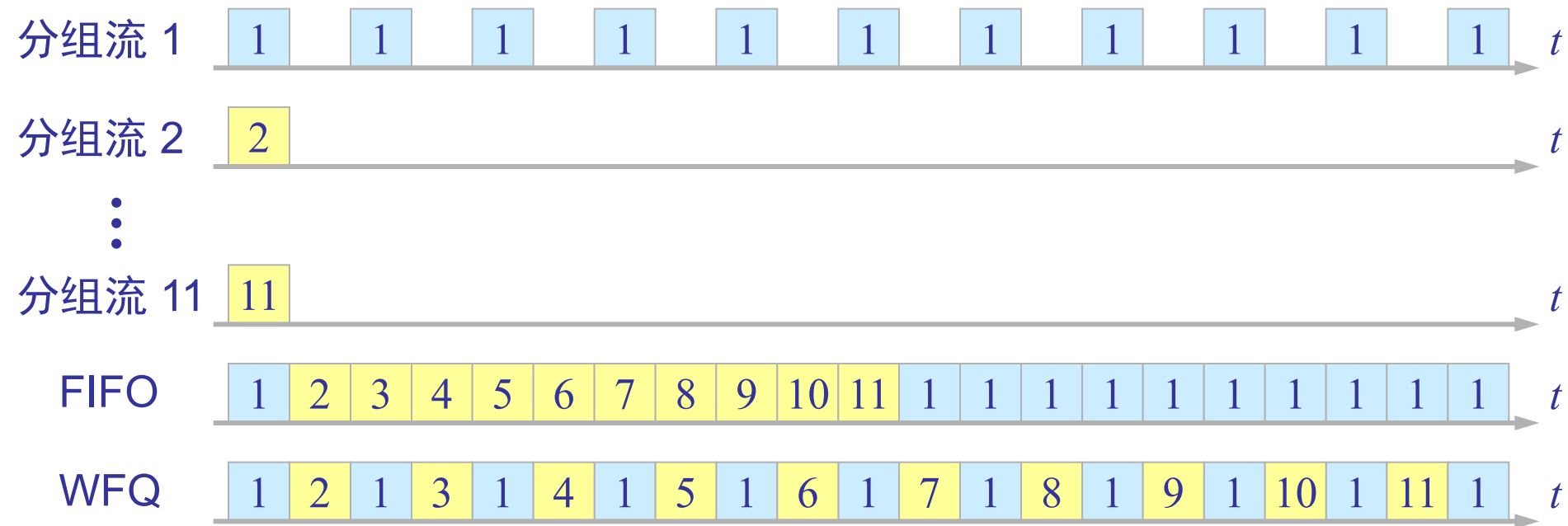
WFQ 与 FIFO 的比较 (a) 分组流 1 的分组连续输入



4.2 路由器原理

WFQ 与 FIFO 的比较

(b) 分组流 1 的分组断续输入



4.3 网际协议IP协议

4.3 IP协议

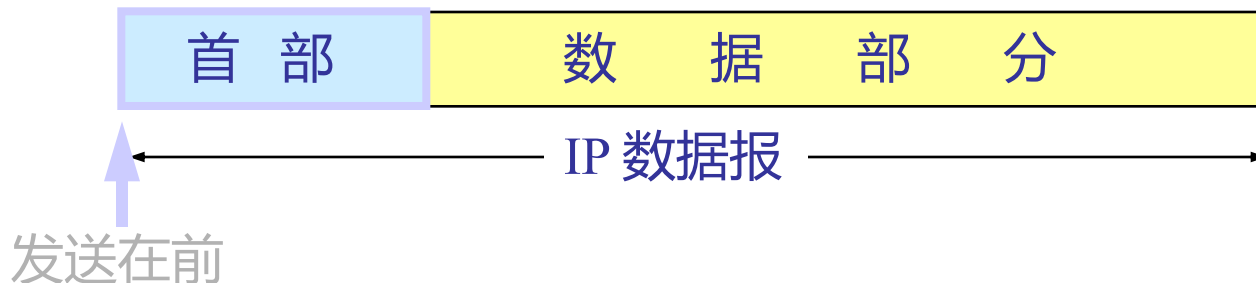
为什么以IP 地址做为端系统的标识

- 由于全世界存在着各式各样的网络，它们使用不同的硬件地址。要使这些异构网络能够互相通信就必须进行非常复杂的硬件地址转换工作，因此几乎是不可能的事。
- 连接到因特网的主机都拥有统一的 IP 地址，它们之间的通信就像连接在同一个网络上那样简单方便，因为调用 ARP 来寻找某个路由器或主机的硬件地址都是由计算机软件自动进行的，对用户来说是看不见这种调用过程的。

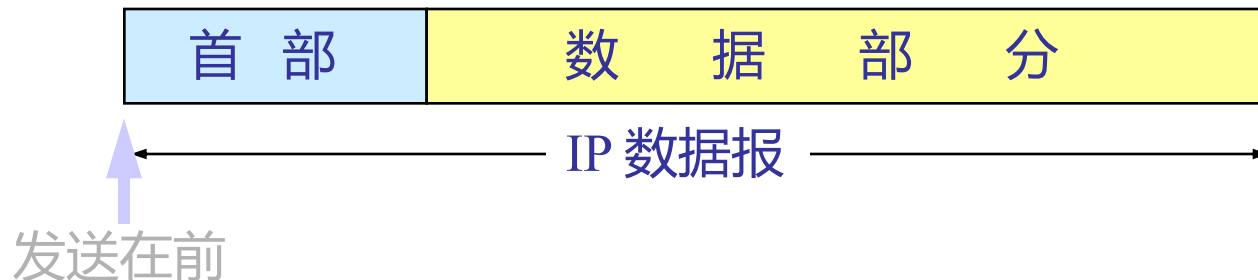
4.3 IP协议

- 一个 IP 数据报由首部和数据两部分组成。
- 首部的前一部分是固定长度，共 20 字节，是所有 IP 数据报必须具有的。
- 在首部的固定部分的后面是一些可选字段，其长度是可变的。

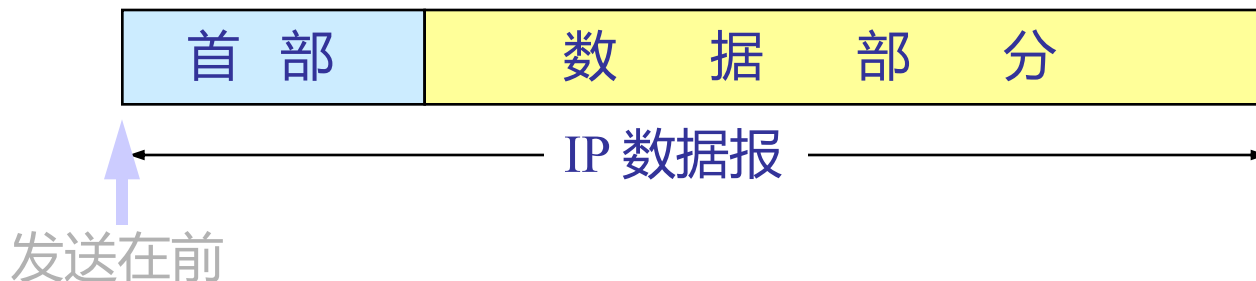
4.3.1 IP数据报结构



4.3.1 IP数据报结构



4.3.1 IP数据报结构



4.3.1 IP数据报结构

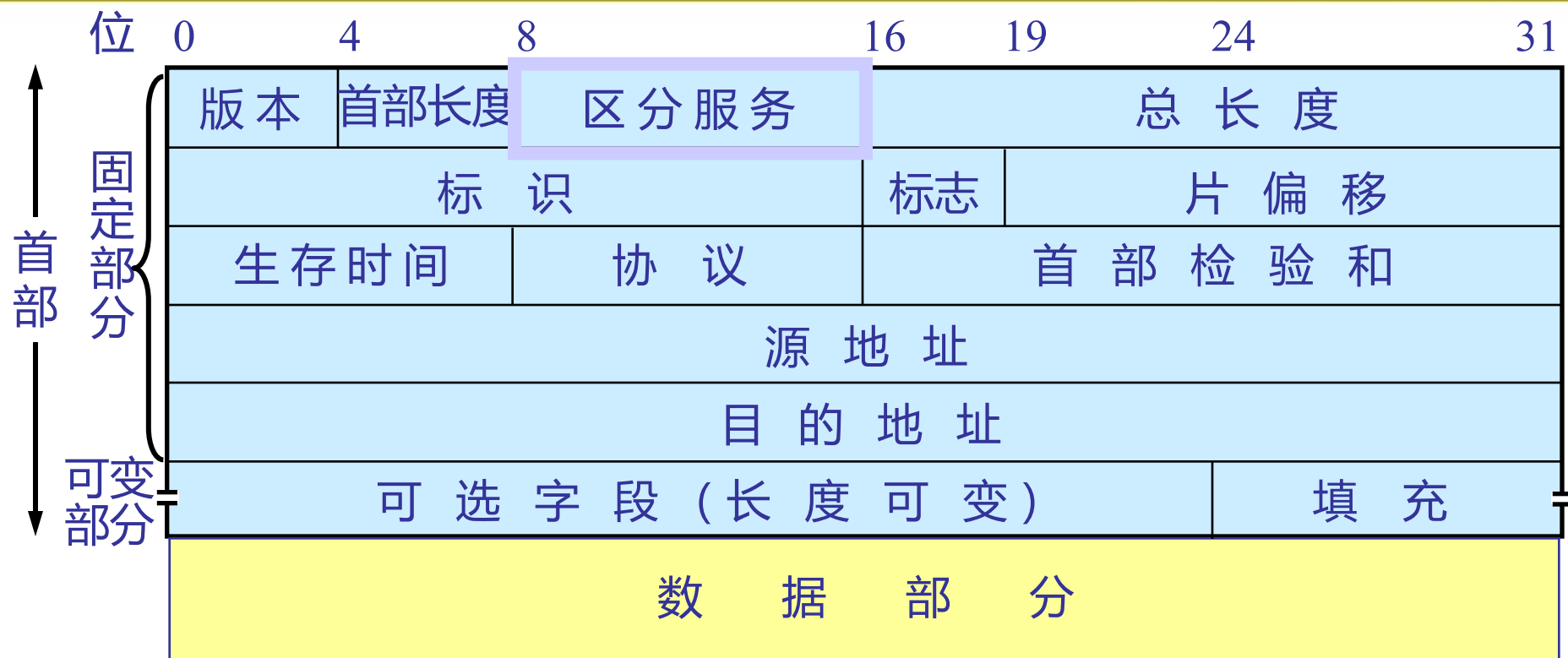


版本——占 4 位，指 IP 协议的版本
目前的 IP 协议版本号为 4 (即 IPv4)

4.3.1 IP数据报结构

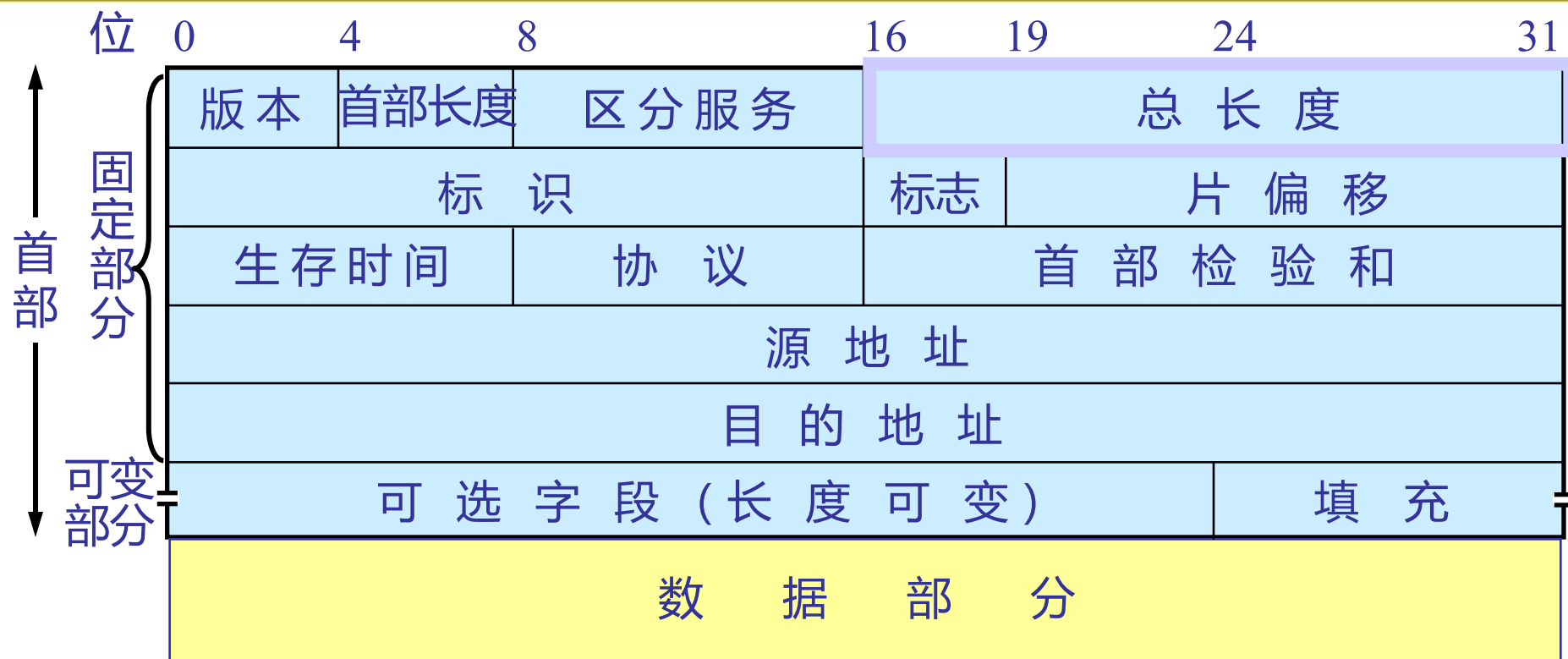


首部长度——占 4 位，可表示的最大数值是 15 个单位(一个单位为 4 字节)
因此 IP 的首部长度的最大值是 60 字节。



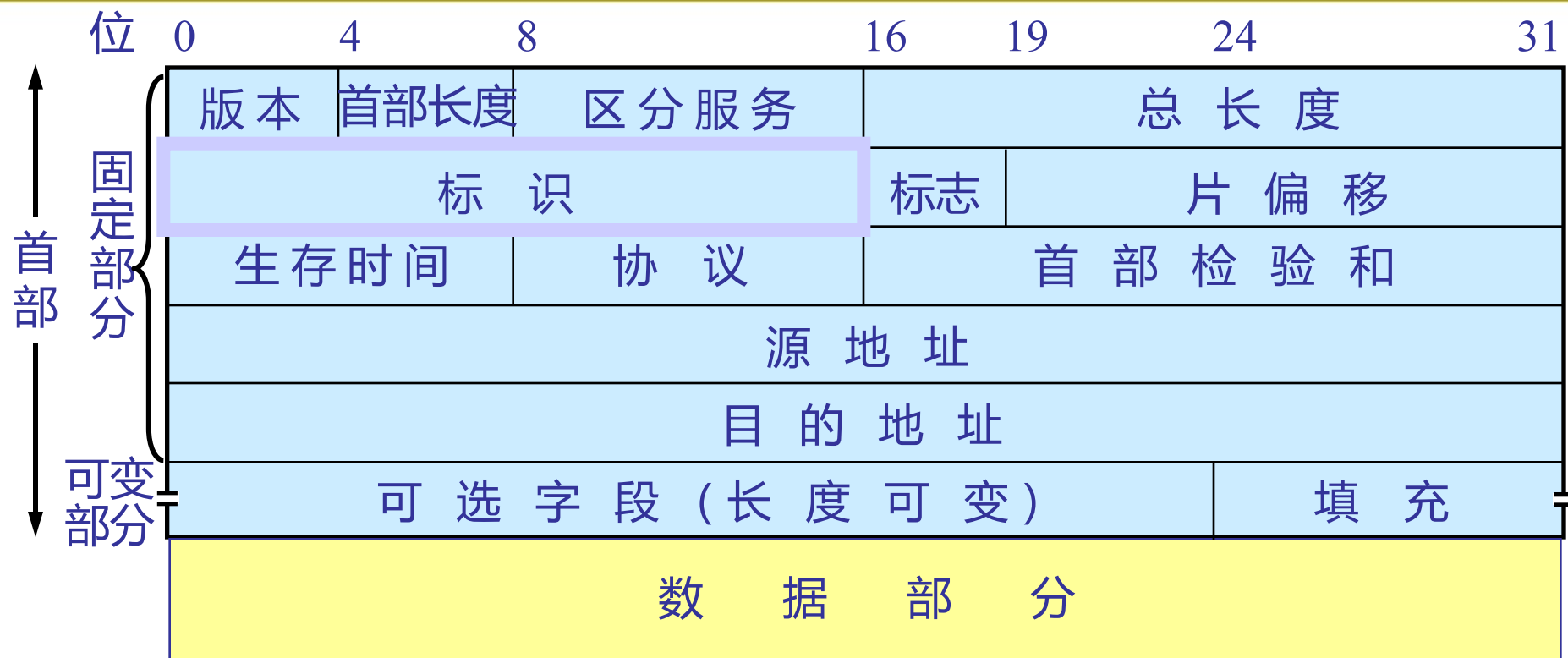
区分服务——占 8 位，用来获得更好的服务
 在旧标准中叫做服务类型，但实际上一直未被使用过。
 1998 年这个字段改名为区分服务。
 只有在使用区分服务 (DiffServ) 时，这个字段才起作用。
 在一般的情况下都不使用这个字段

4.3.1 IP数据报结构



总长度——占 16 位，指首部和数据之和的长度，单位为字节，因此数据报的最大长度为 65535 字节。
总长度必须不超过最大传送单元 MTU。

4.3.1 IP数据报结构



标识(identification) 占 16 位，
它是一个计数器，用来产生数据报的标识。



标志(flag) 占 3 位，目前只有前两位有意义。

标志字段的最低位是 **MF** (More Fragment)。

MF = 1 表示后面“还有分片”。MF = 0 表示最后一个分片。

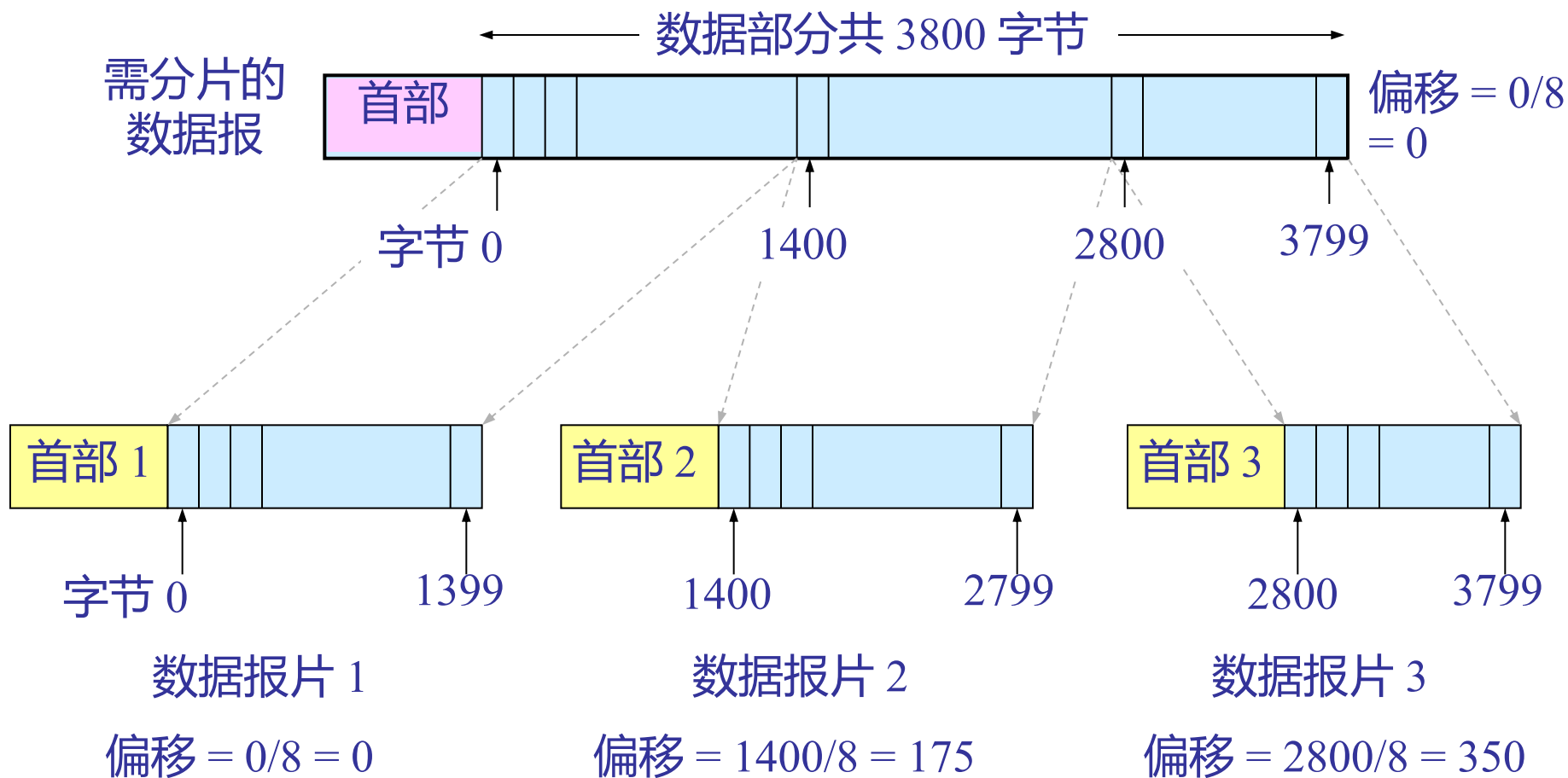
标志字段中间的一位是 **DF** (Don't Fragment)。

只有当 DF = 0 时才允许分片。

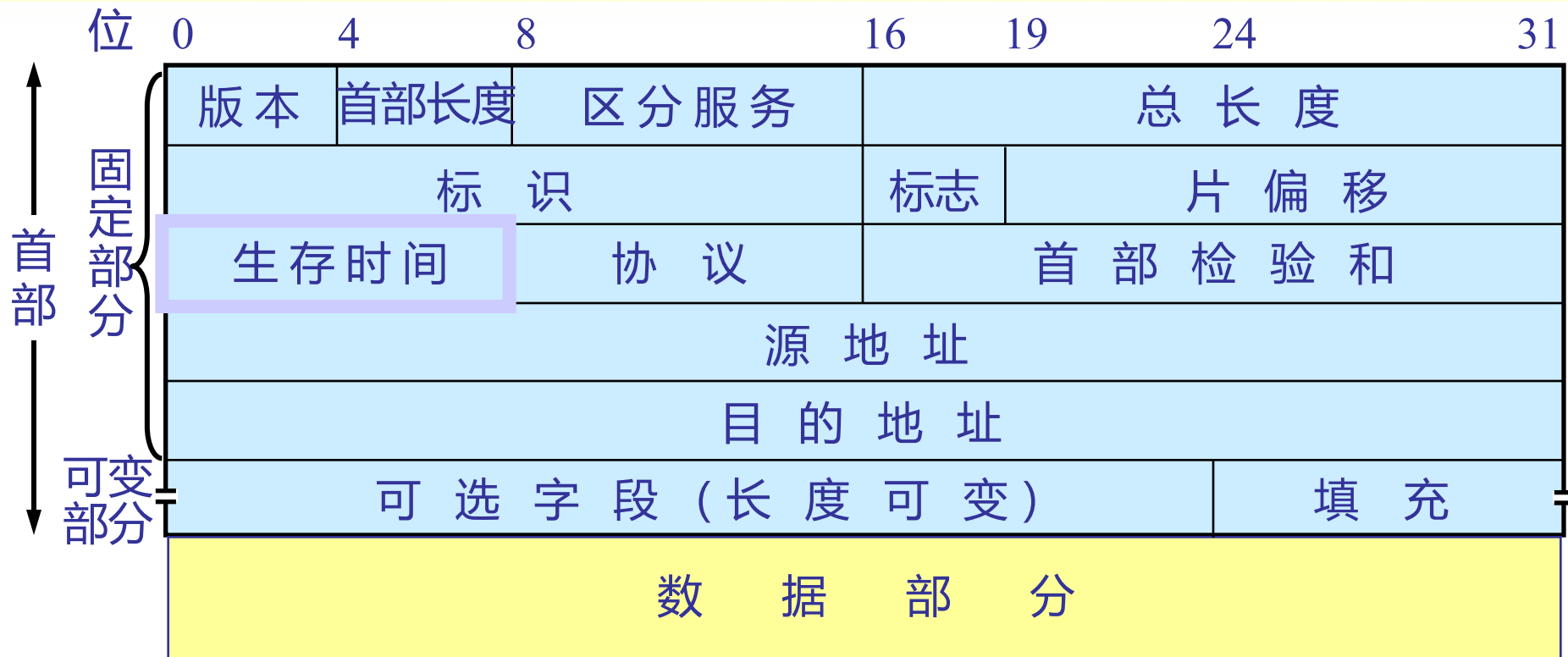


片偏移(13 位)指出：较长的分组在分片后
某片在原分组中的相对位置。
片偏移以 8 个字节为偏移单位。

4.3.1 IP数据报结构

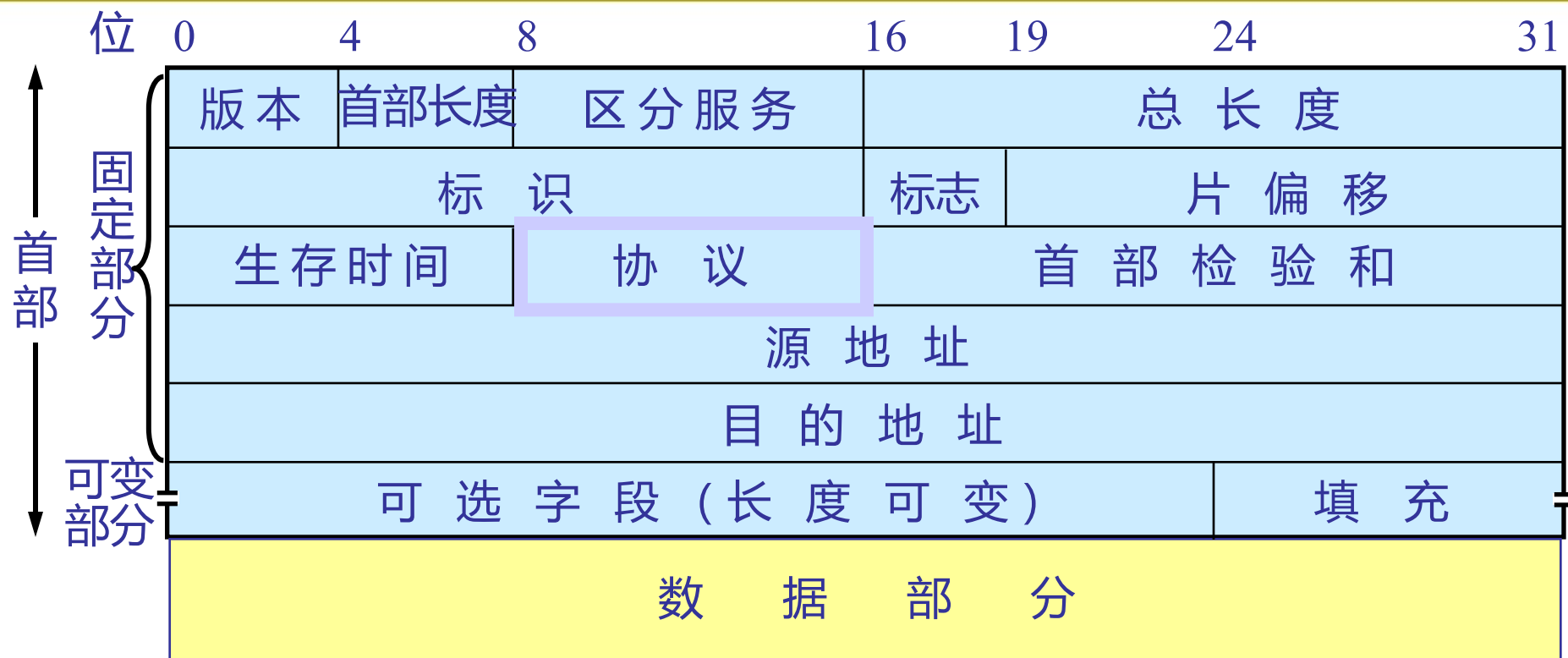


4.3.1 IP数据报结构



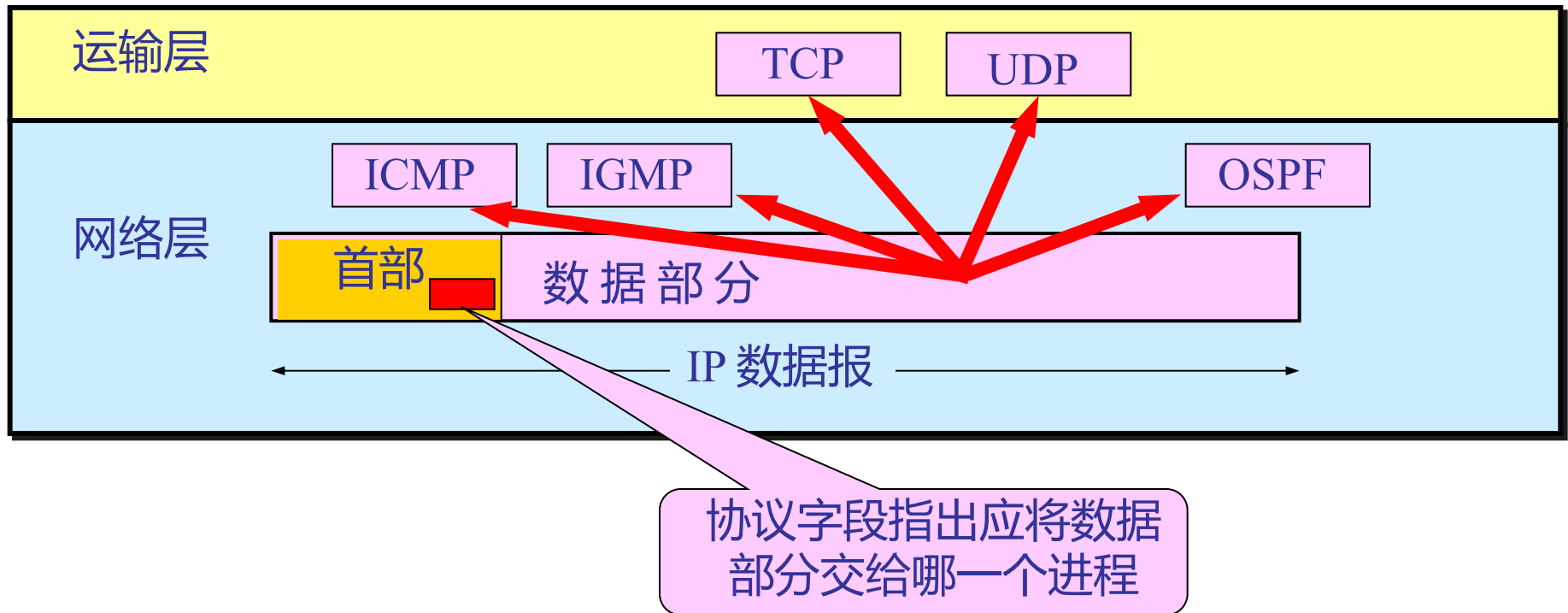
生存时间(8 位)记为 TTL (Time To Live)
数据报在网络中可通过的路由器数的最大值。

4.3.1 IP数据报结构

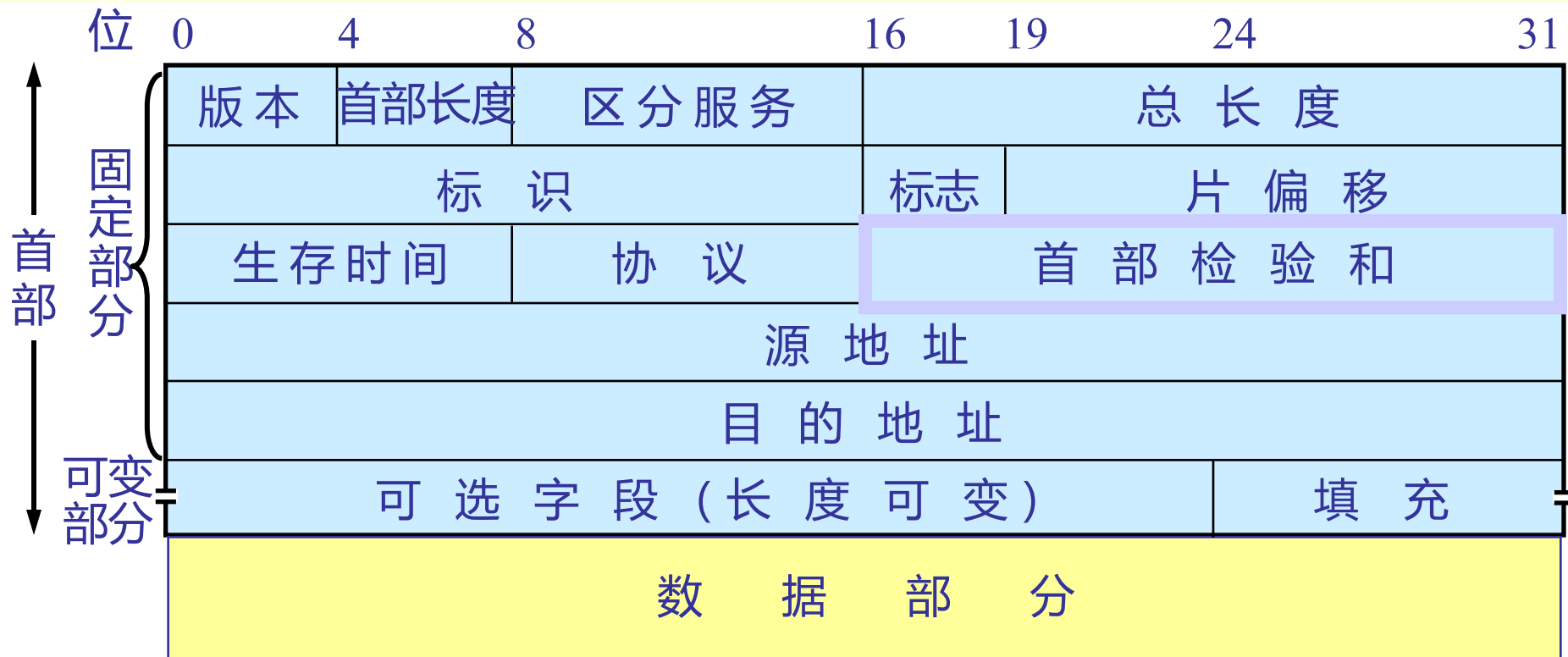


协议(8 位)字段指出此数据报携带的数据使用何种协议以便目的主机的 IP 层将数据部分上交给哪个处理过程

4.3.1 IP数据报结构



4.3.1 IP数据报结构



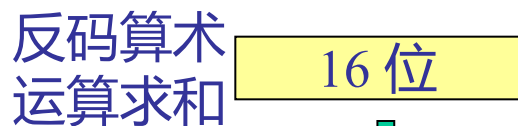
首部检验和(16 位)字段只检验数据报的首部
不检验数据部分。

这里不采用 CRC 检验码而采用简单的计算方法。

发送端

接收端

数据报首部

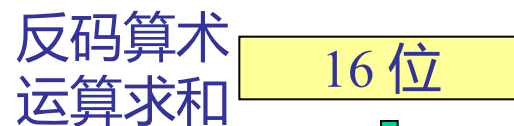
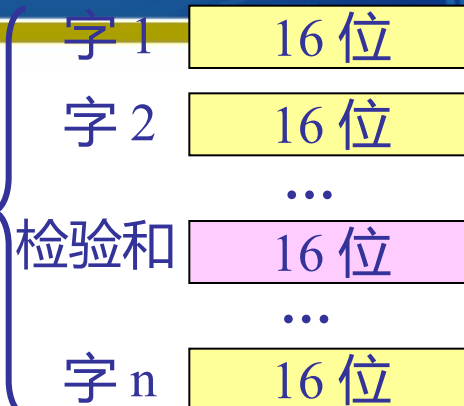
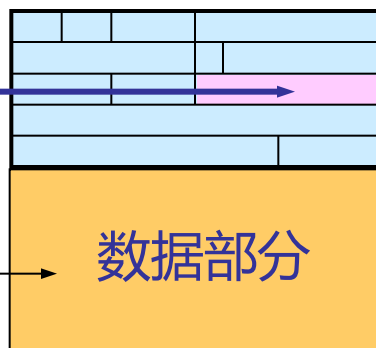


取反码

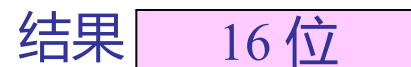


数据部分
不参与检验和的计算

IP 数据报

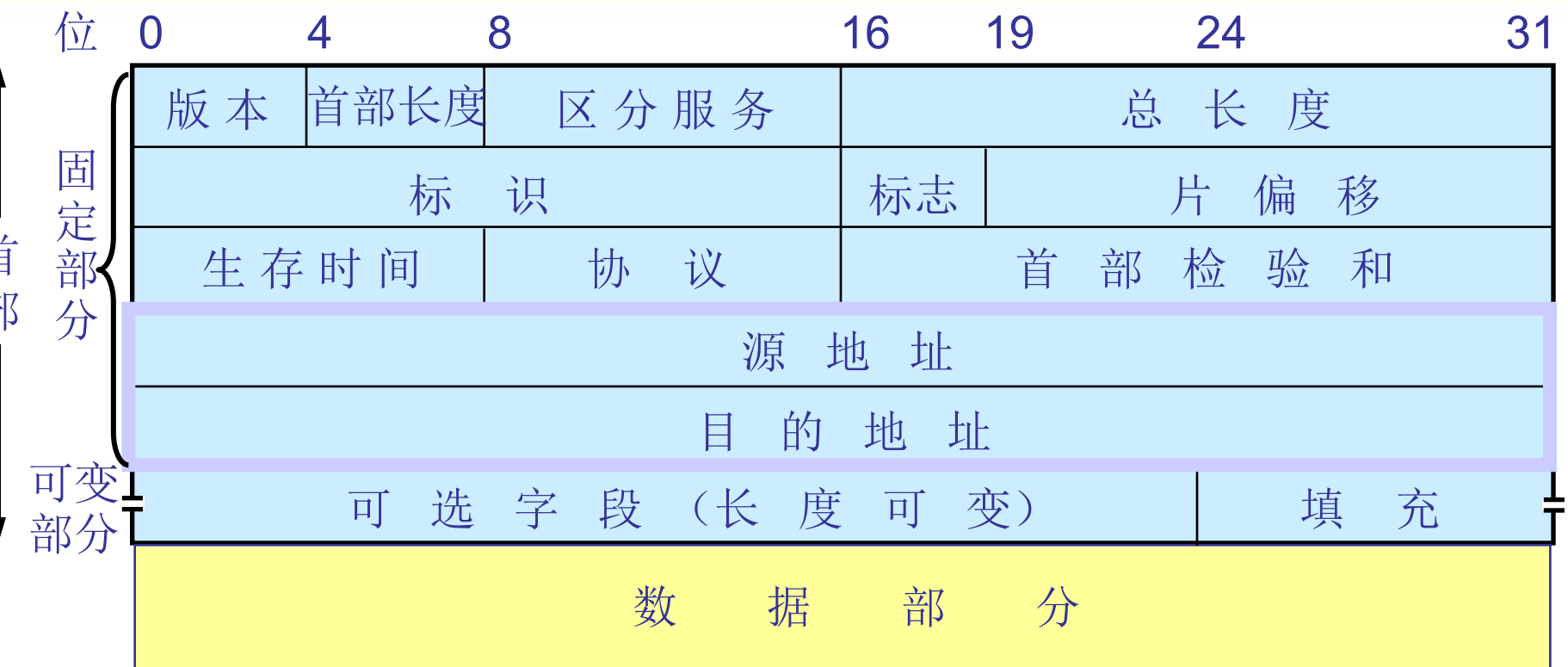


取反码



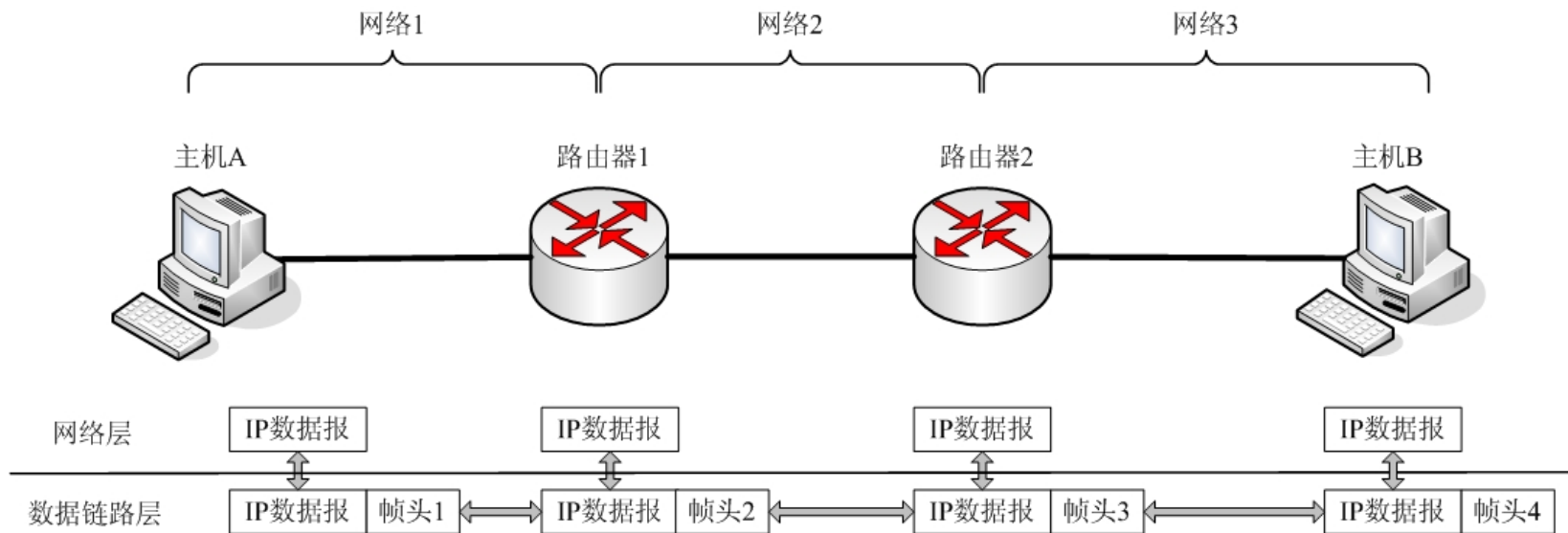
若结果为 0, 则保留;
否则, 丢弃该数据报

4.3.1 IP数据报结构



源地址和目的地址都各占 4 字节

4.3.2 IP数据报分片



4.3.2 IP数据报分片

- IP数据报在互联网上传输的时候，它可能要跨越多个不同种类的异构网络。每个网络的数据链路层都有其自己的特定帧格式且其大小有限制。
- 每种网络都规定最大传送单元MTU(Maximum Transfer Unit)。

4.3.2 IP数据报分片

| | | | | |
|-------------------|------|---------|-----------|-------|
| 版 本 | 首部长度 | 区 分 服 务 | 总 长 度 | |
| 标 识 | | | 标志 | 片 偏 移 |
| 生 存 时 间 | 协 议 | | 首 部 检 验 和 | |
| 源 地 址 | | | | |
| 目 的 地 址 | | | | |
| 可 选 字 段 (长 度 可 变) | | | 补 充 | |
| 数 据 部 分 | | | | |
| 首 部 | | 数 据 部 分 | | |

ID：每个数据报唯一
解决了：标识同一数据报的各个分片

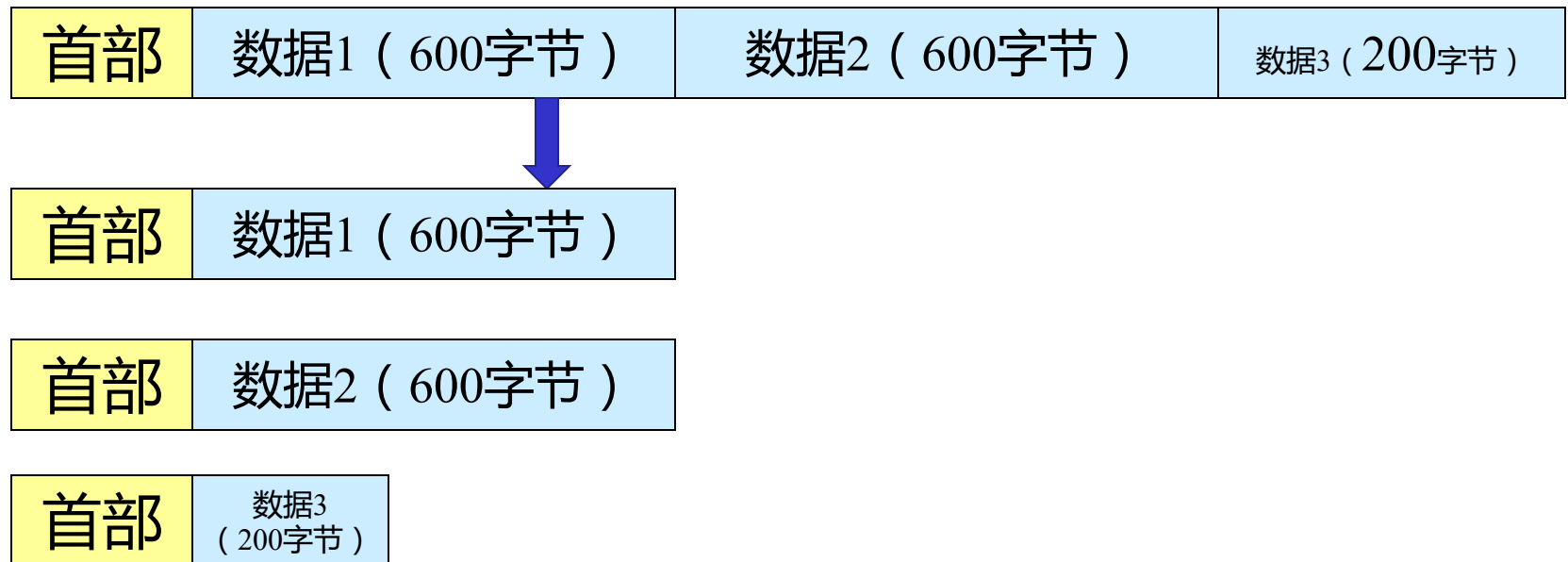
DF=0 可以分片
DF=1 不得分片

段标识：
MF=0，最后一片
MF=1，非最后一片

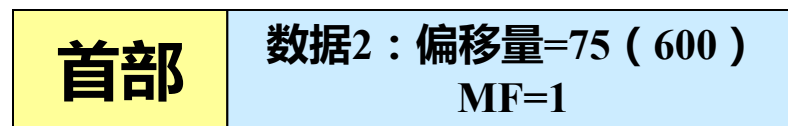
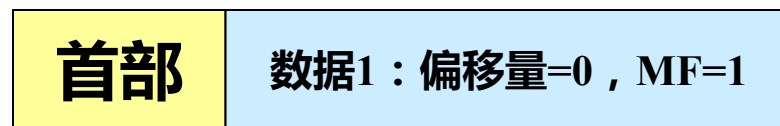
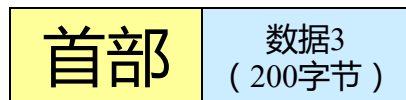
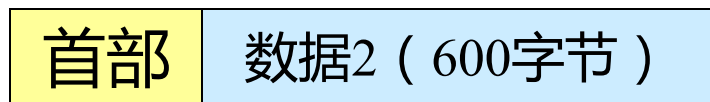
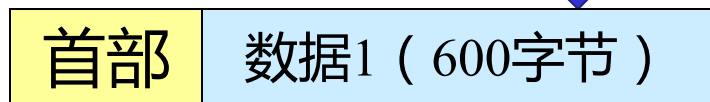
标识分片在原来数据报文中的位置，以8字节为单位

4.3.2 IP数据报分片

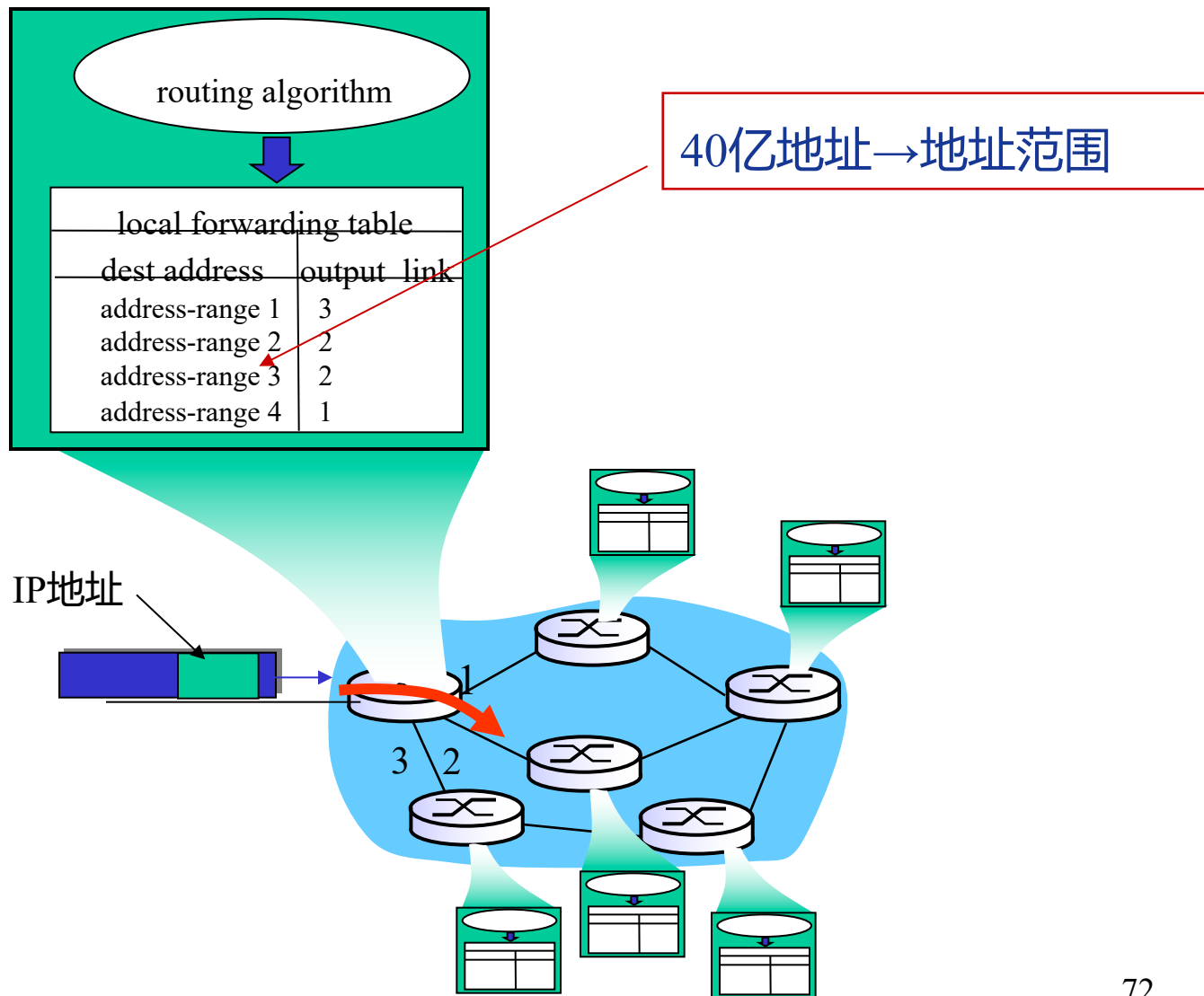
- 例：IP数据报长度1420字节（假如首部无选项），网络MTU620字节，如何分片？



4.3.2 IP数据报分片



4.3.2 IP编址



4.3.2 IP编址

| 目的地址范围 | 链路接口 |
|---|------|
| 11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111 | 0 |
| 11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111 | 1 |
| 11001000 00010111 00011000 00000000 through 11001000 00010111 00011111 11111111 | 2 |
| 其他 | 3 |

4.3.2 IP编址

最长前缀匹配优先：

| 目的地址范围 | 链路接口 |
|----------------------------------|------|
| 11001000 00010111 00010*** ***** | 0 |
| 11001000 00010111 00011000 ***** | 1 |
| 11001000 00010111 00011*** ***** | 2 |
| 其他 | 3 |

例子：

DA: 11001000 00010111 00010110 10100001

那个接口？ 0

DA: 11001000 00010111 00011000 10101010

那个接口？ 1

在检索转发表时，优先选择与分组目的地址匹配前缀最长的入口（entry）。

4.3.2 IP编址

| 目的地址范围 | 链路接口 |
|----------------------------------|------|
| 11001000 00010111 00010*** ***** | 0 |
| 11001000 00010111 00011000 ***** | 1 |
| 11001000 00010111 00011*** ***** | 2 |
| 其他 | 3 |

范围是什么？
网络地址/net-id

4.3.2 IP编址

- 分类的 IP 地址。这是最基本的编址方法，在 1981 年就通过了相应的标准协议。
- 子网的划分。这是对最基本的编址方法的改进，其标准[RFC 950]在 1985 年通过。
- CIDR。这是比较新的无分类编址方法。1993 年提出后很快就得到推广应用。

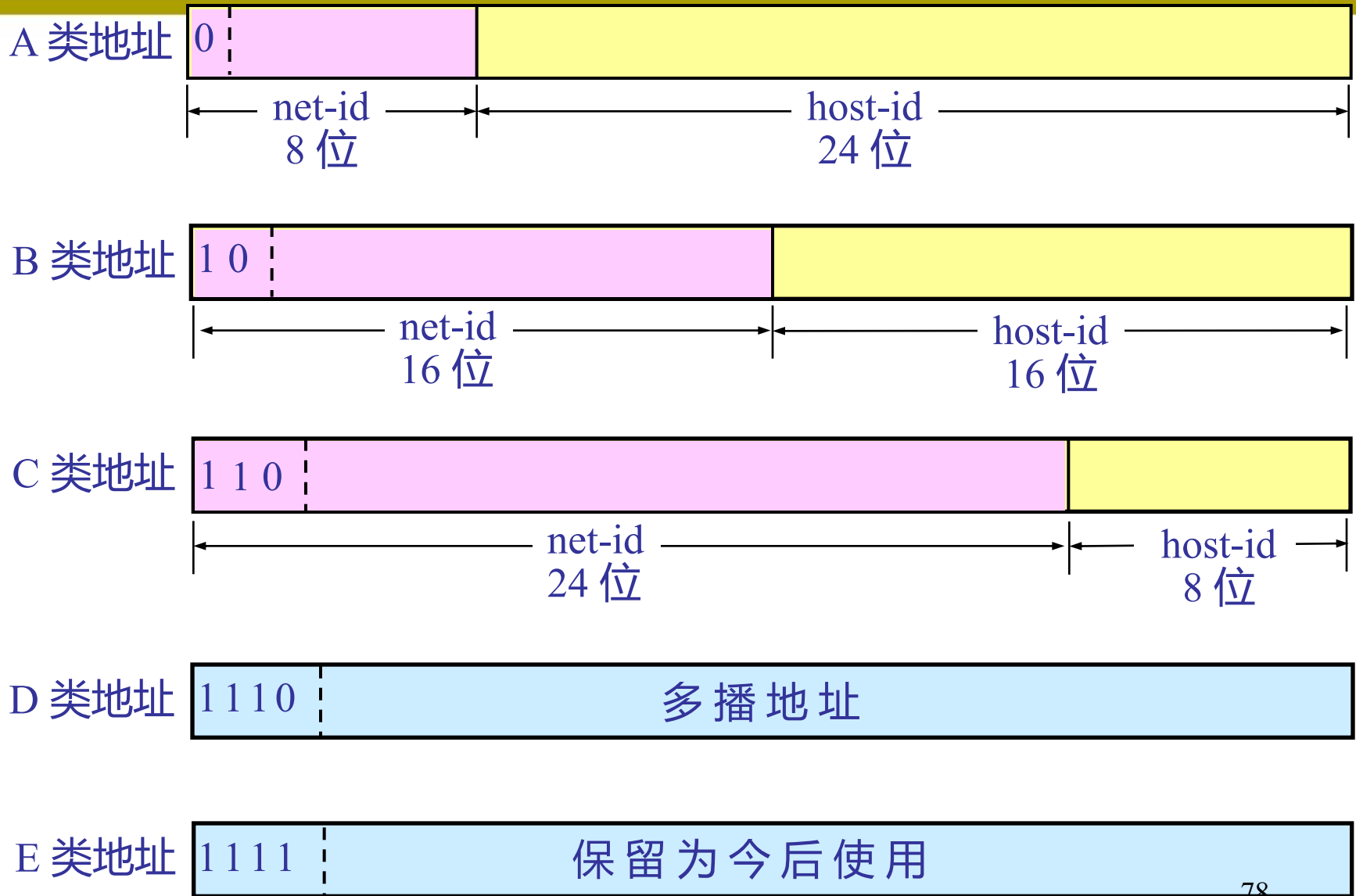
4.3.2 IP编址

- 每一类地址都由两个固定长度的字段组成，其中一个字段是网络号 net-id，它标志主机（或路由器）所连接到的网络，而另一个字段则是主机号 host-id，它标志该主机（或路由器）。
- 两级的 IP 地址可以记为：

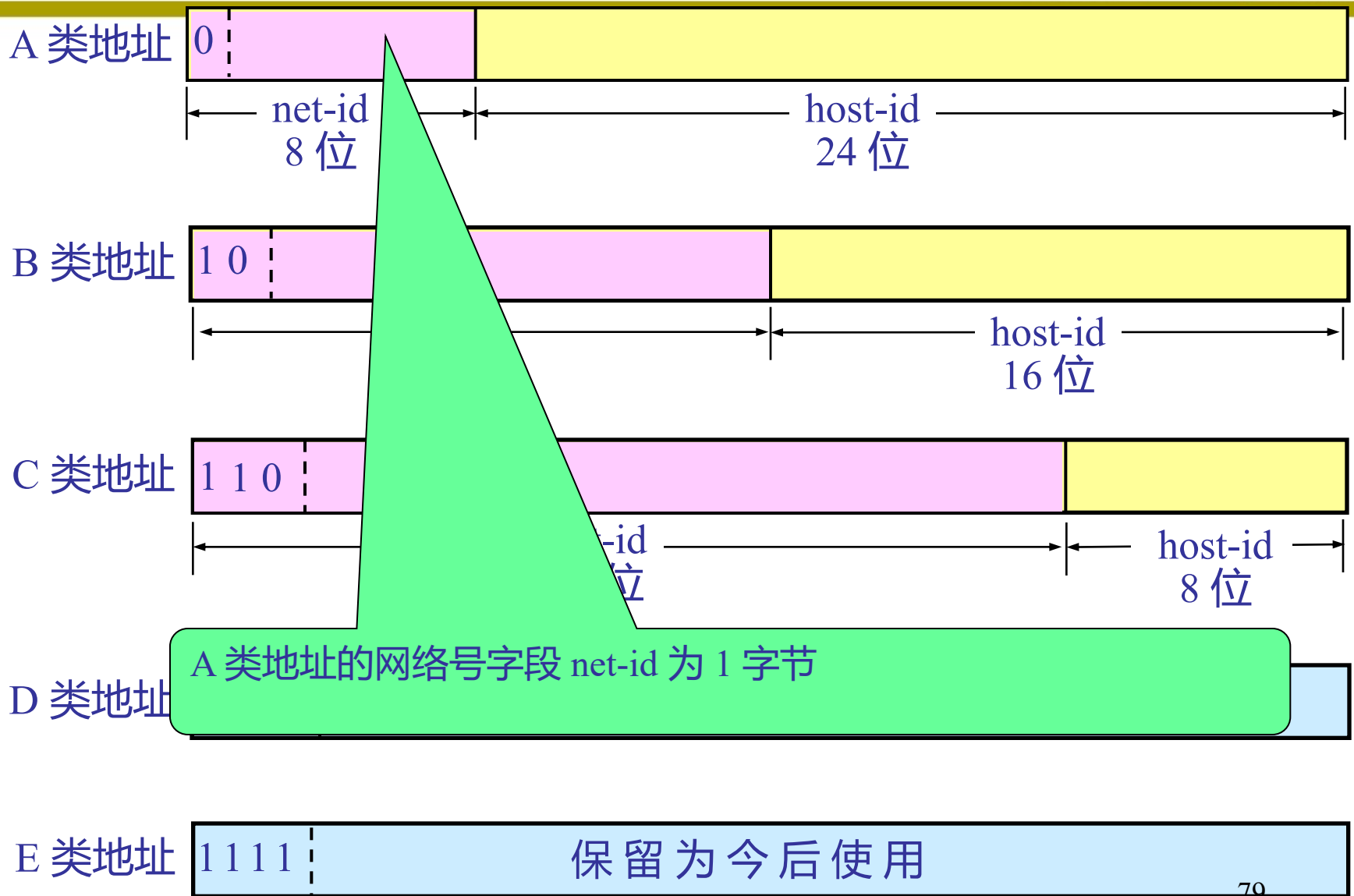
IP 地址 ::= { <网络号>, <主机号> }

::= 代表 “定义为”

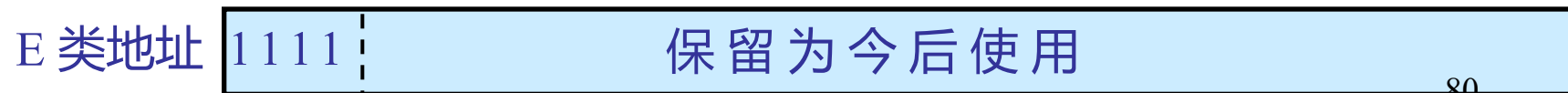
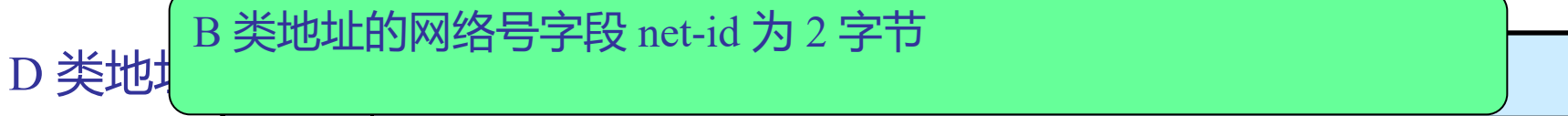
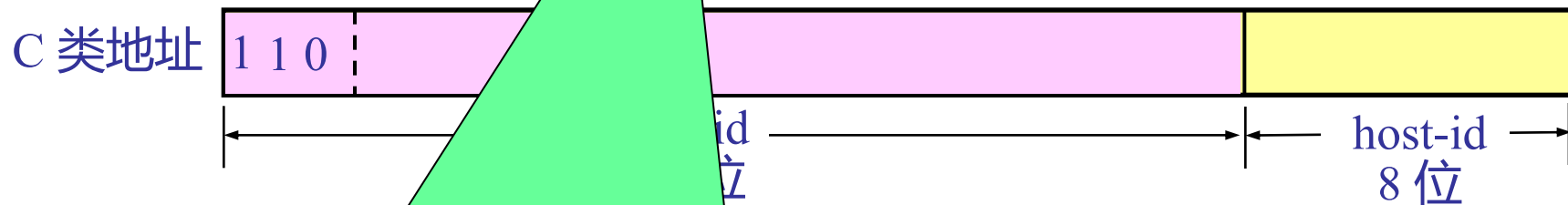
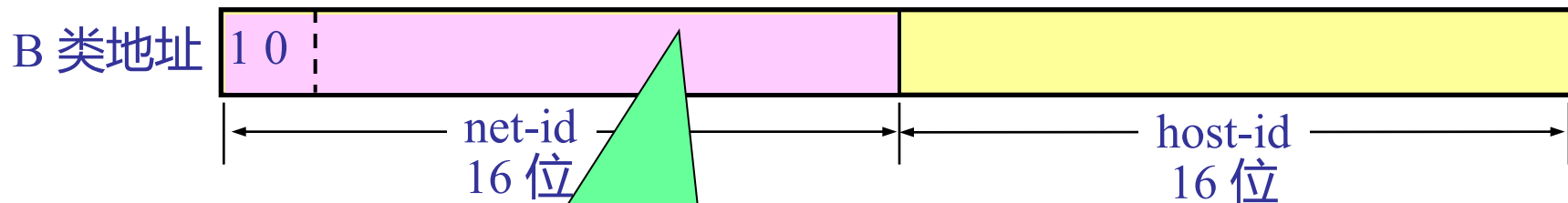
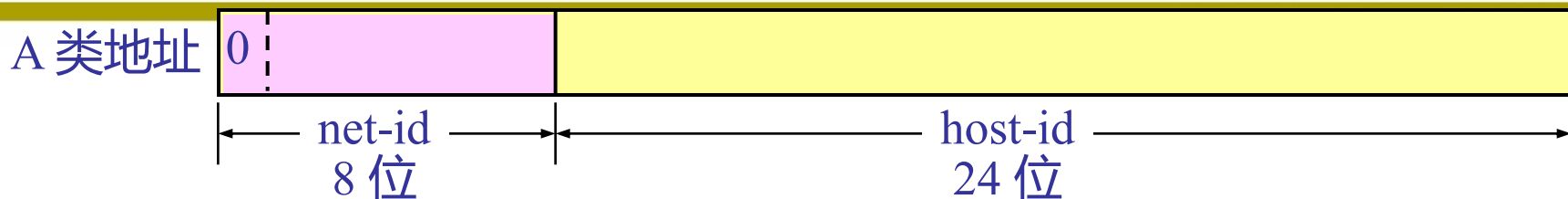
2级IP地址



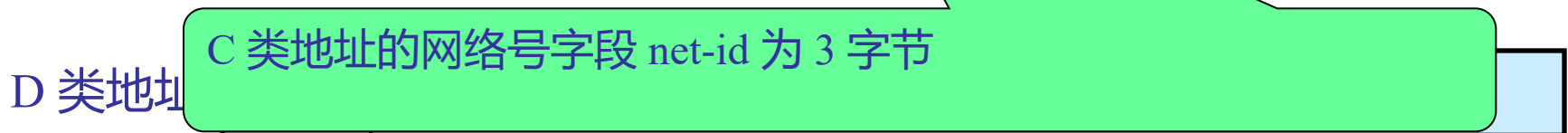
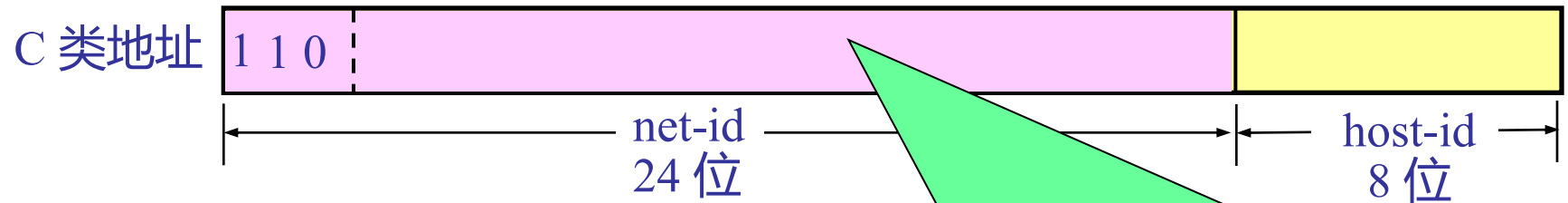
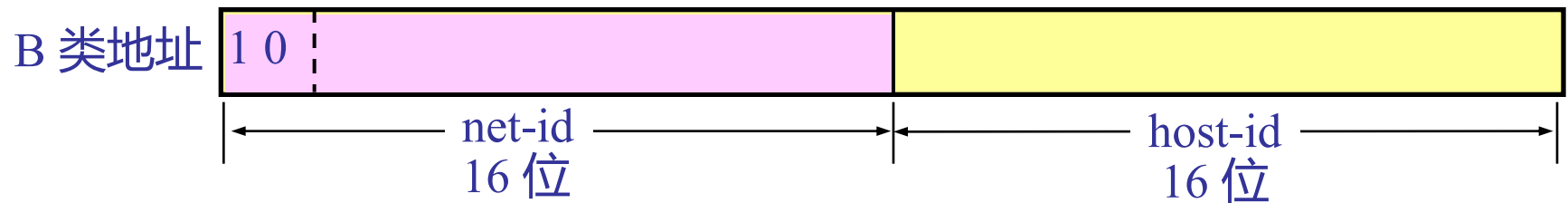
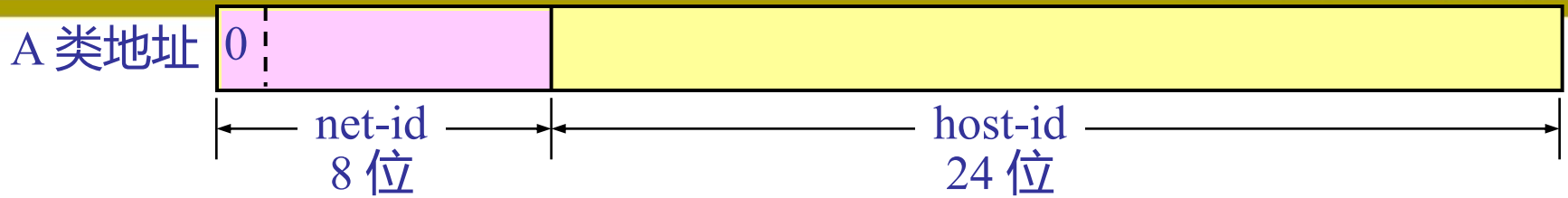
2级IP地址



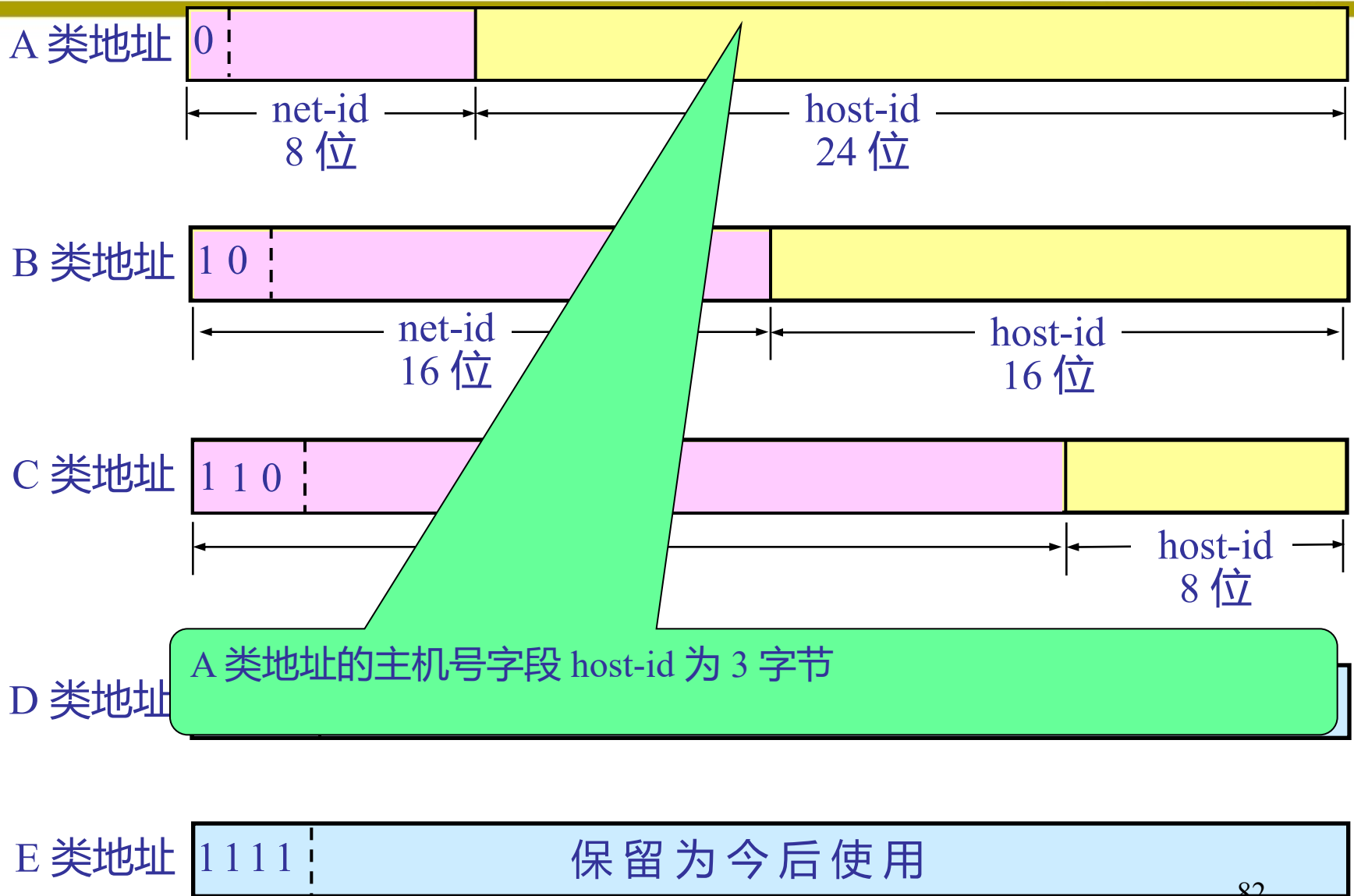
IP 地址中的网络号字段和主机号字段



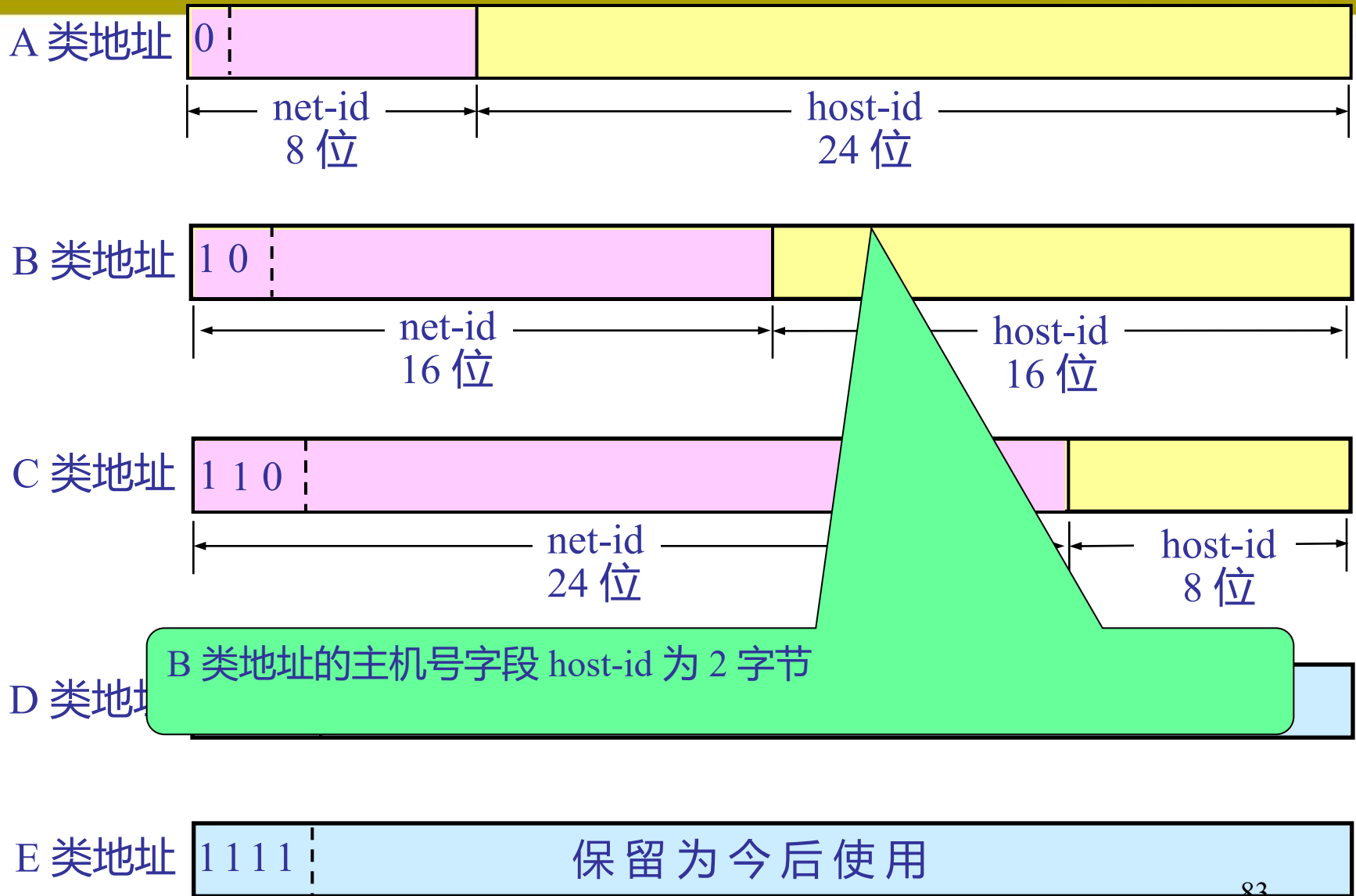
2级IP地址



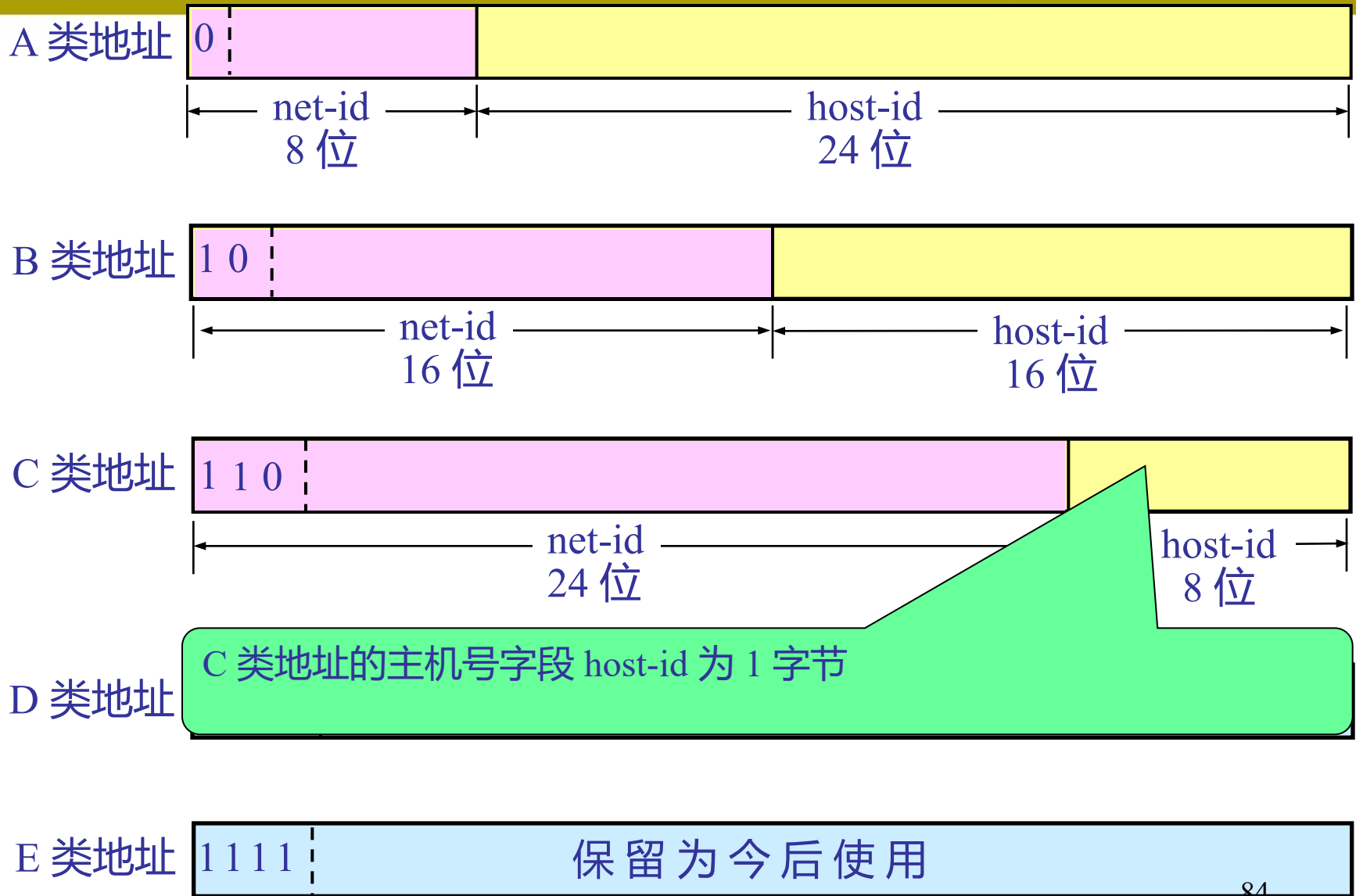
2级IP地址



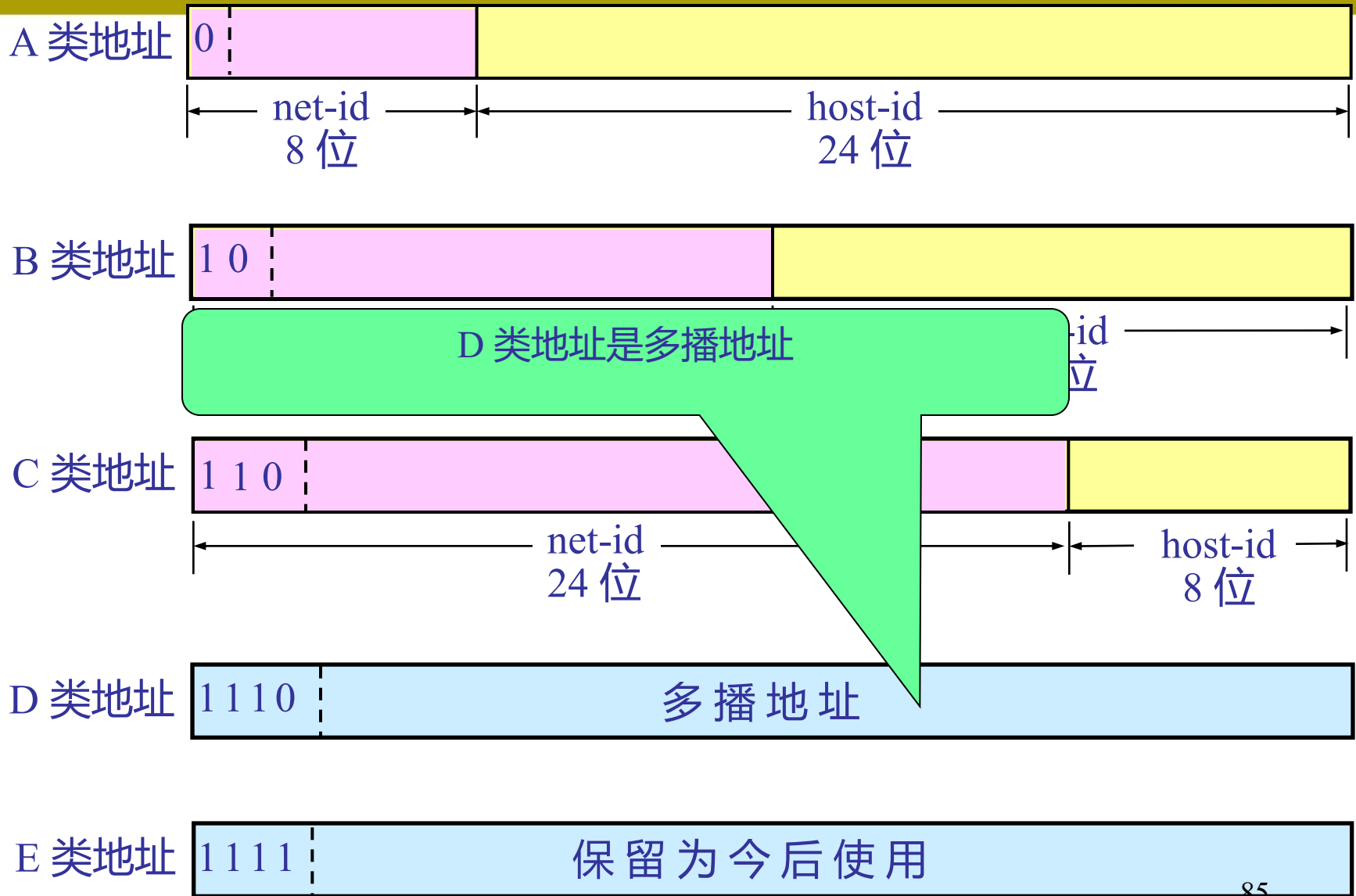
2级IP地址



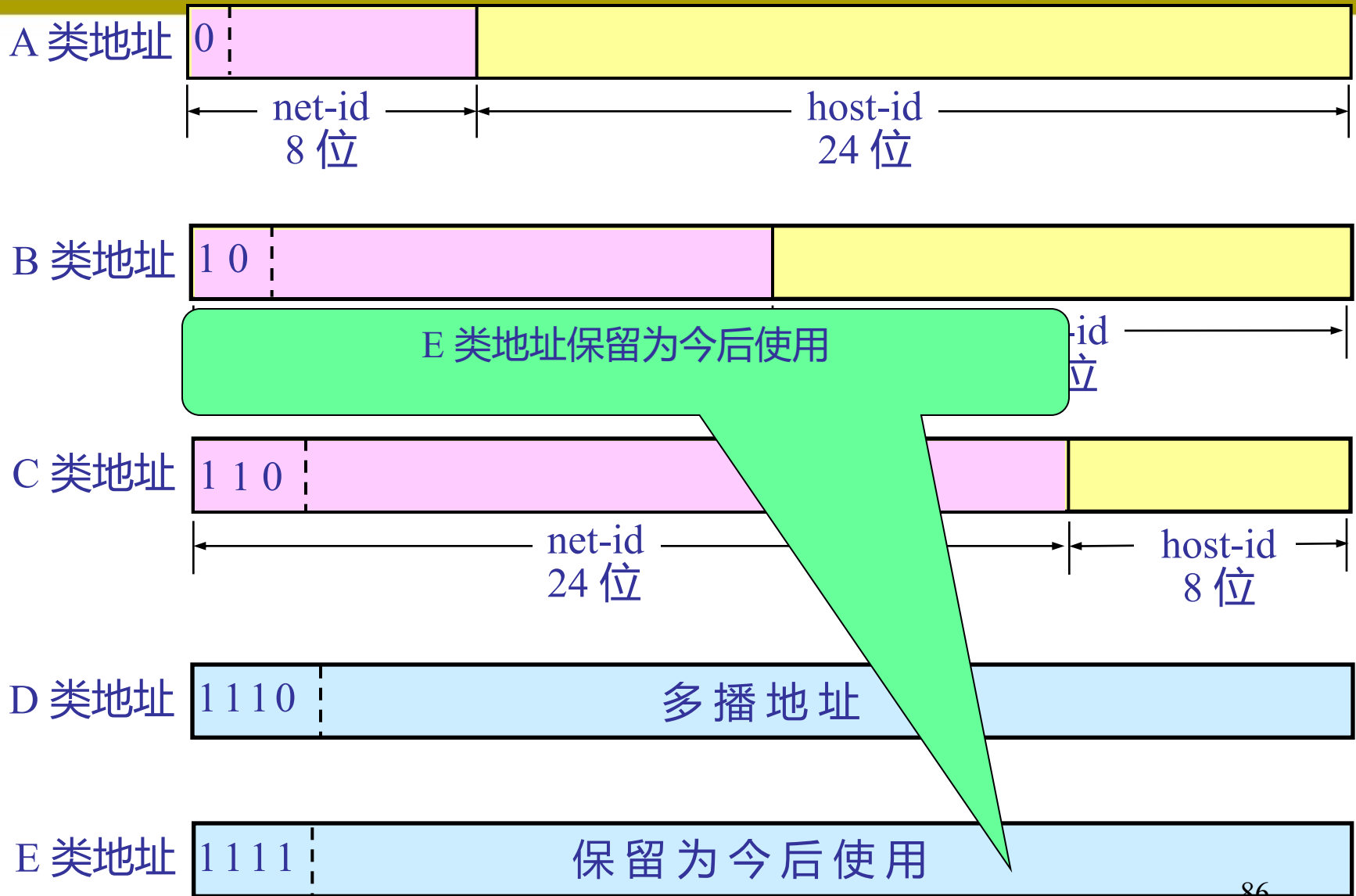
2级IP地址



2级IP地址

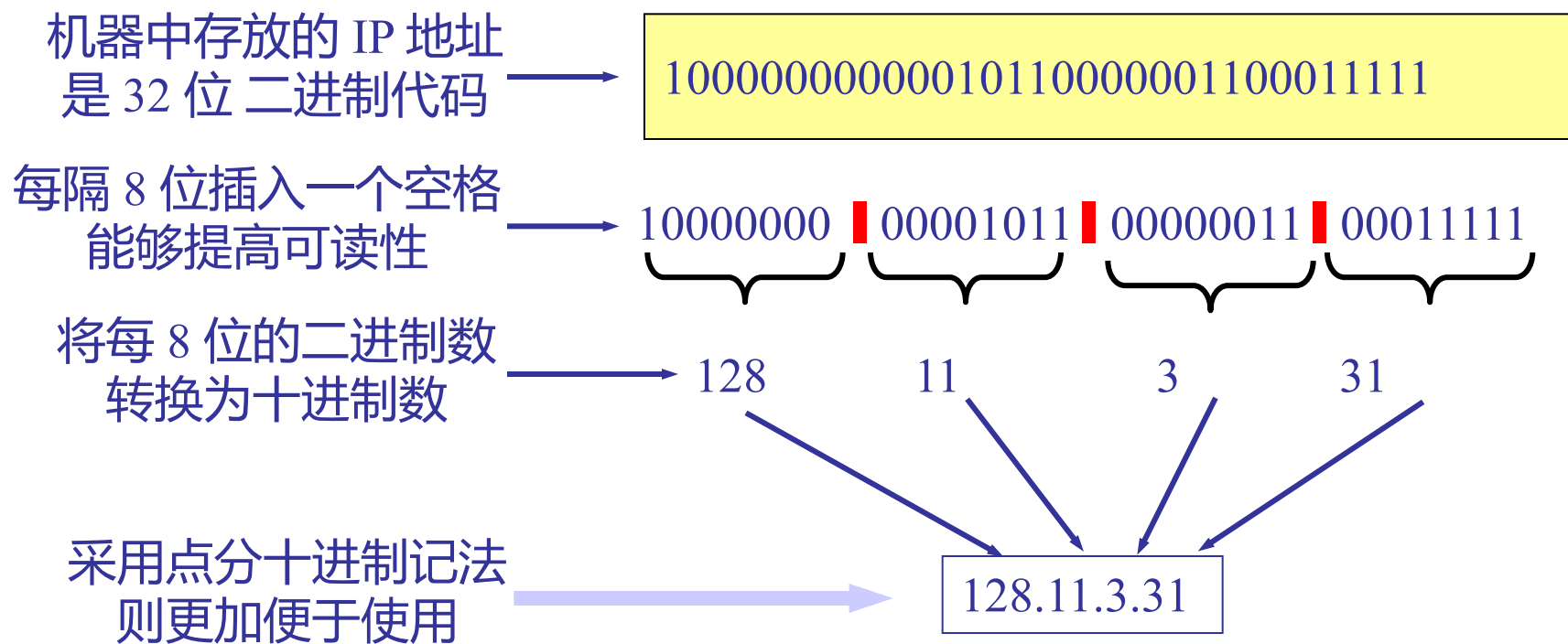


2级IP地址



4.3.2 IP编址

点分十进制记法



4.3.2 IP编址

IP 地址的一些重要特点

IP 地址是一种分等级的地址结构。分两个等级的好处是：

- 第一，IP 地址管理机构在分配 IP 地址时只分配网络号，而剩下的主机号则由得到该网络号的单位自行分配。这样就方便了 IP 地址的管理。
- 第二，路由器仅根据目的主机所连接的网络号来转发分组（而不考虑目的主机号），这样就可以使路由表中的项目数大幅度减少，从而减小了路由表所占的存储空间。

4.3.2 IP编址

IP 地址的一些重要特点

实际上 IP 地址是标志一个主机（或路由器）和一条链路的接口。

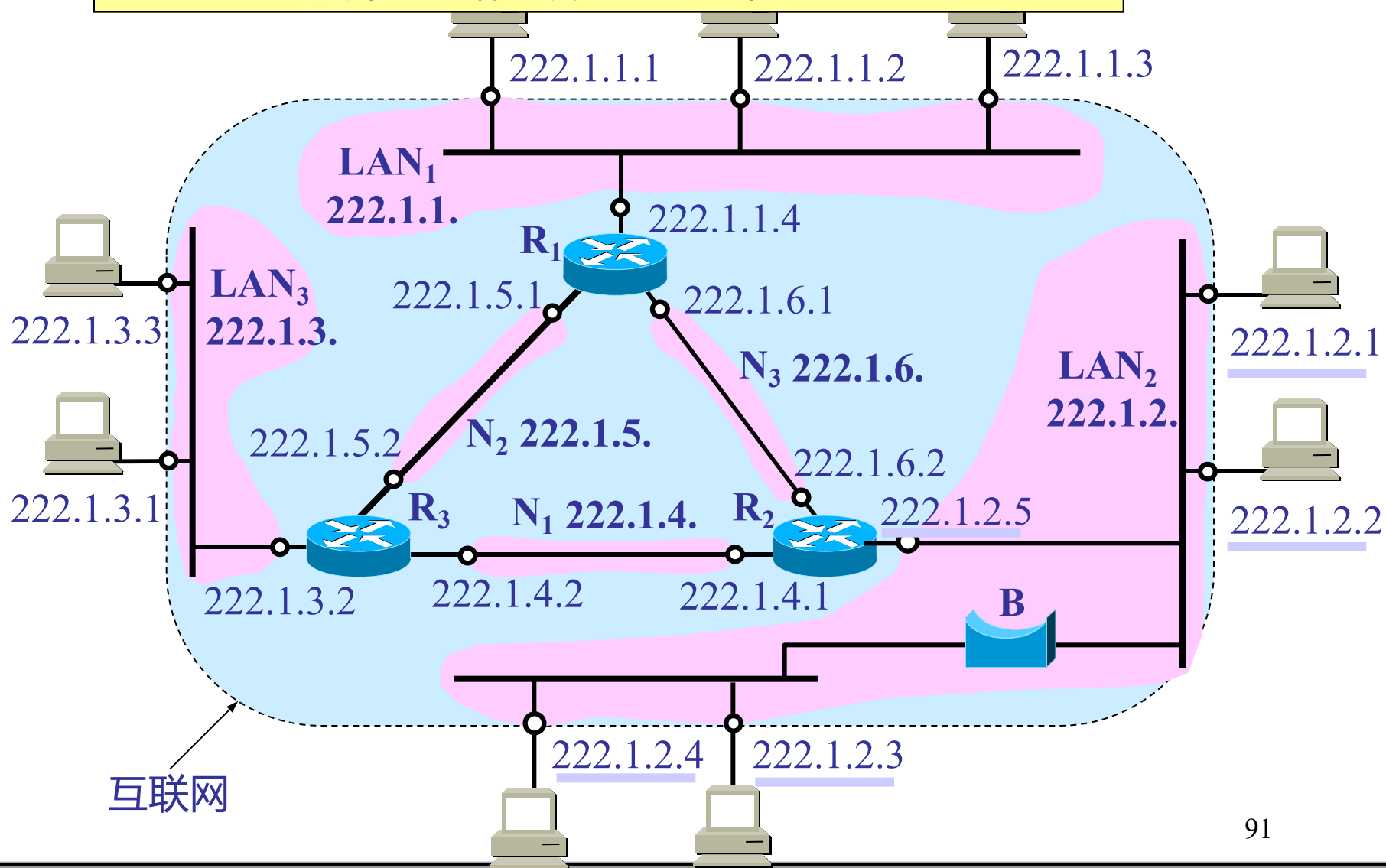
- 当一个主机同时连接到两个网络上时，该主机就必须同时具有两个相应的 IP 地址，其网络号 net-id 必须是不同的。这种主机称为**多归属主机** (multihomed host)。
- 由于一个路由器至少应当连接到两个网络（这样它才能将 IP 数据报从一个网络转发到另一个网络），因此一个路由器至少应当有两个不同的 IP 地址。

4.3.2 IP编址

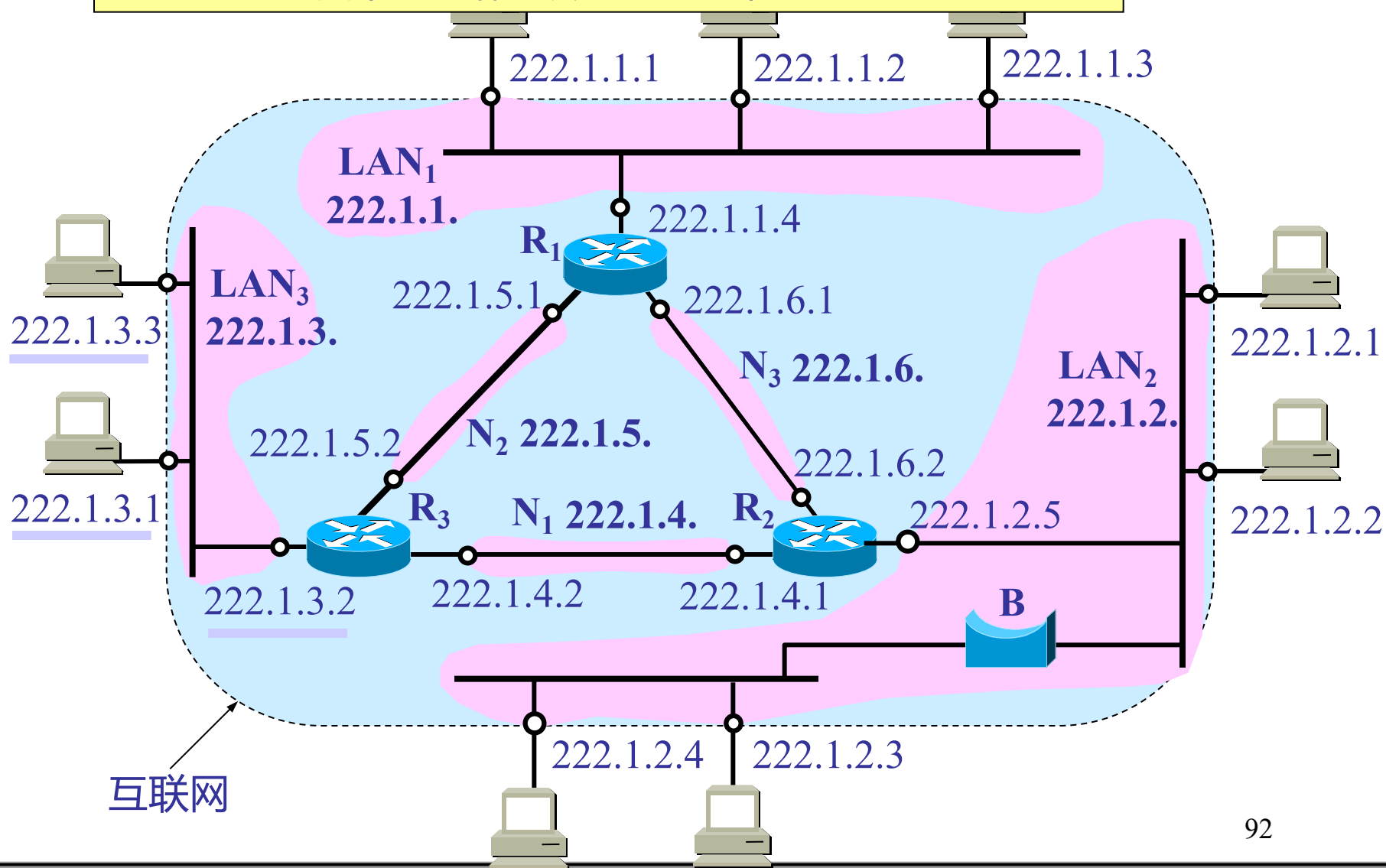
IP 地址的一些重要特点

- (3) 用转发器或网桥连接起来的若干个局域网仍为一个网络，因此这些局域网都具有同样的网络号 net-id。
- (4) 所有分配到网络号 net-id 的网络，范围很小的局域网，还是可能覆盖很大地理范围的广域网，都是平等的。

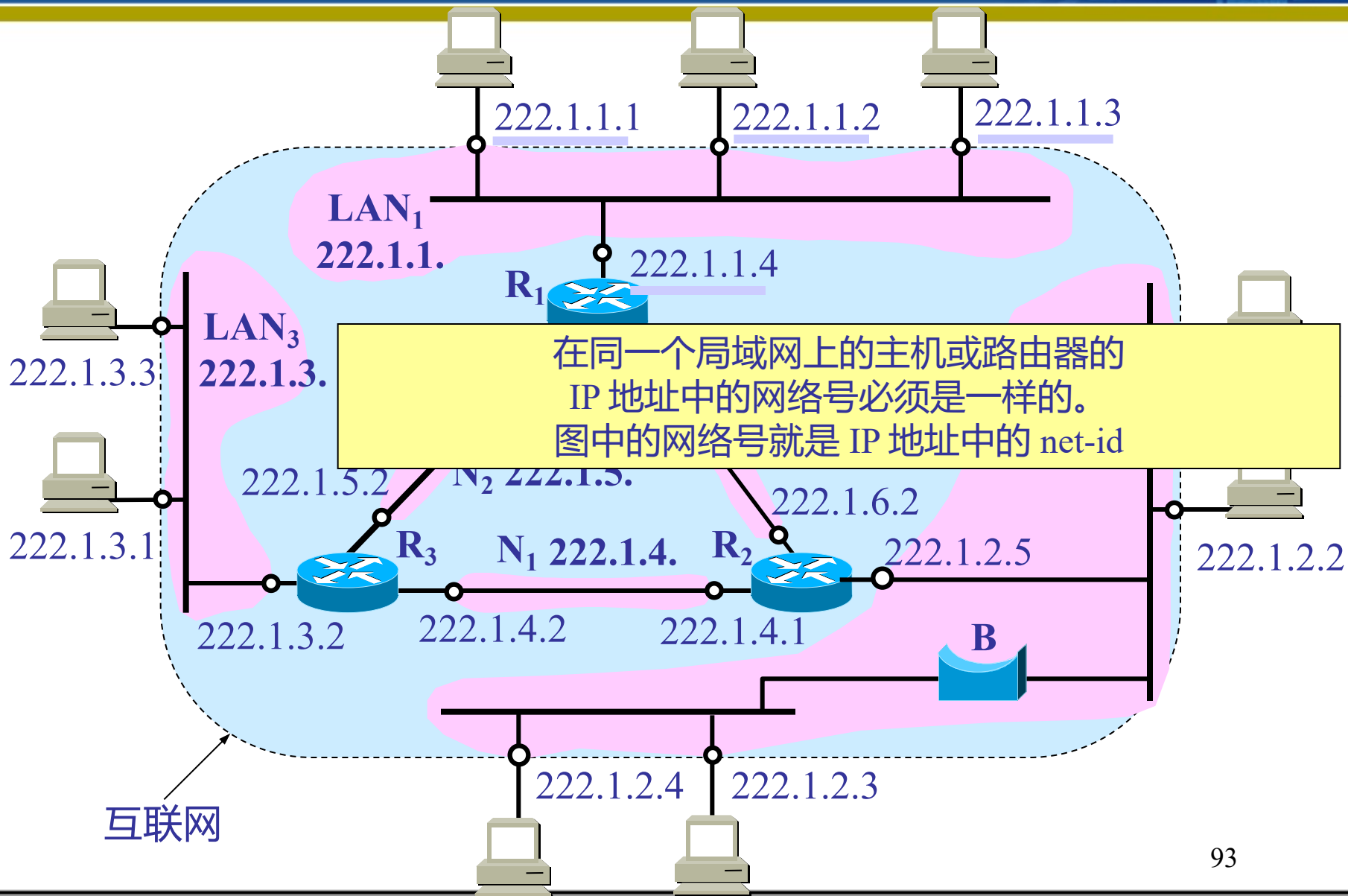
在同一个局域网上的主机或路由器的
IP 地址中的网络号必须是一样的。
图中的网络号就是 IP 地址中的 net-id



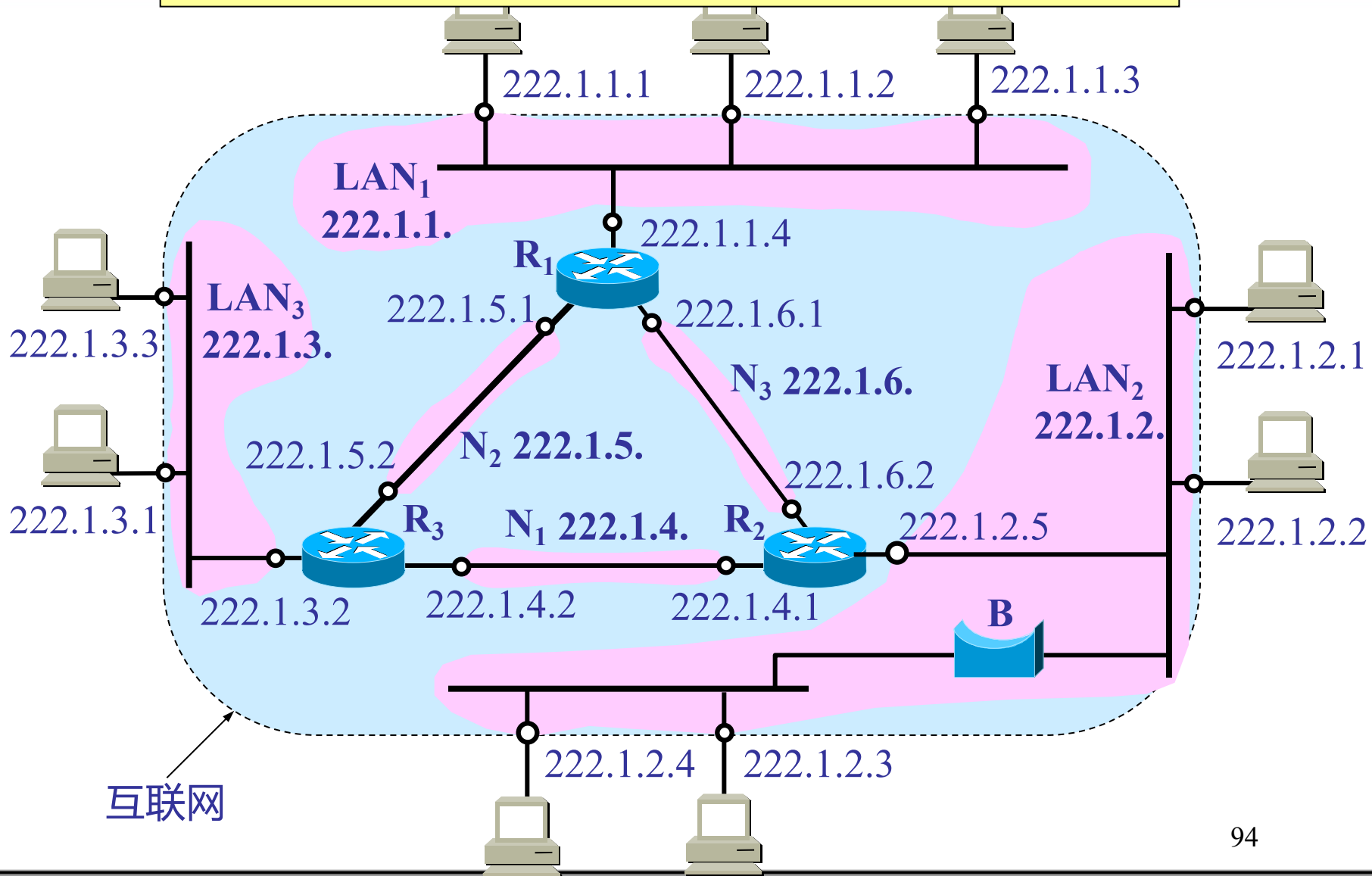
在同一个局域网上的主机或路由器的
IP 地址中的网络号必须是一样的。
图中的网络号就是 IP 地址中的 net-id



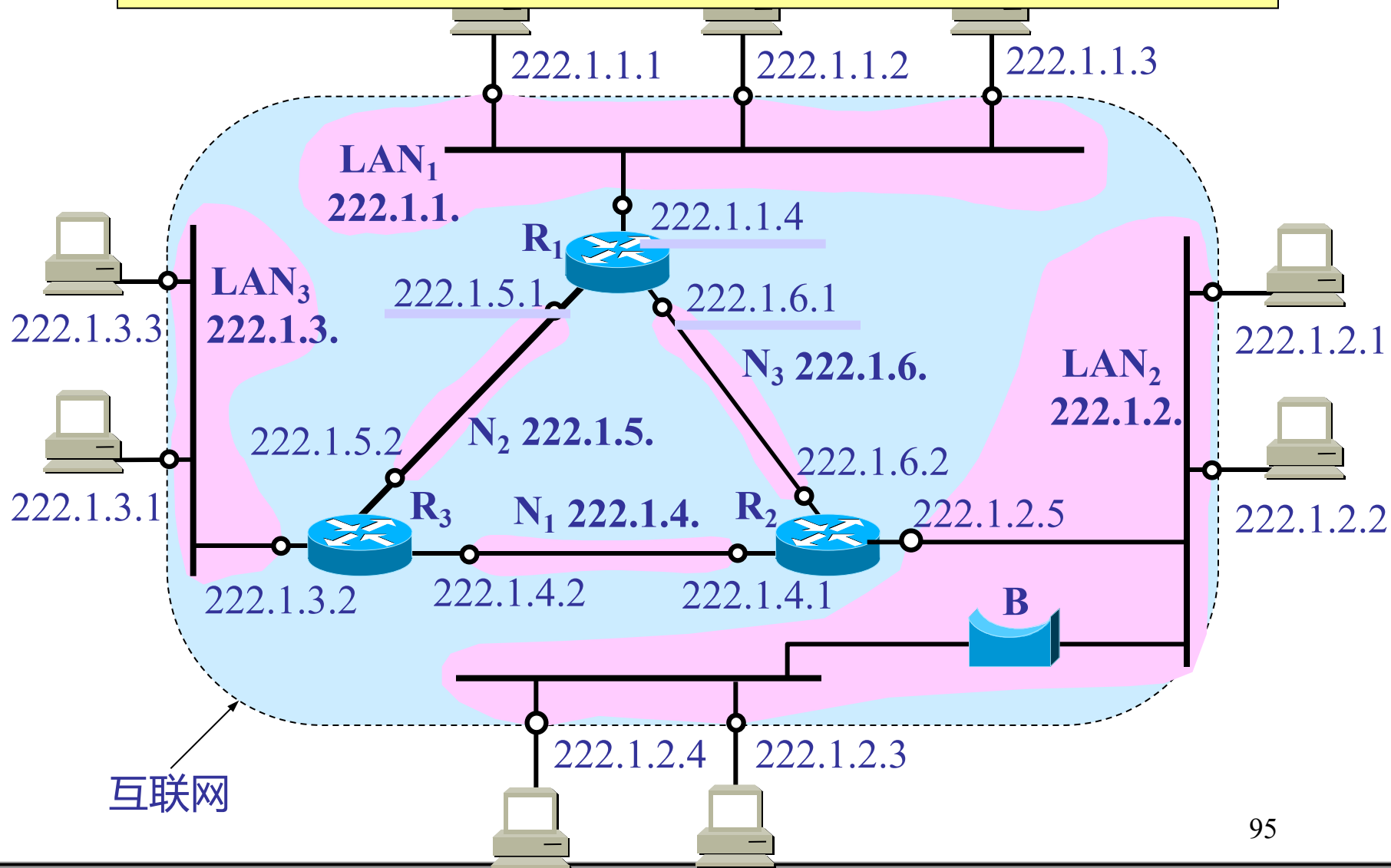
互联网中的 IP 地址



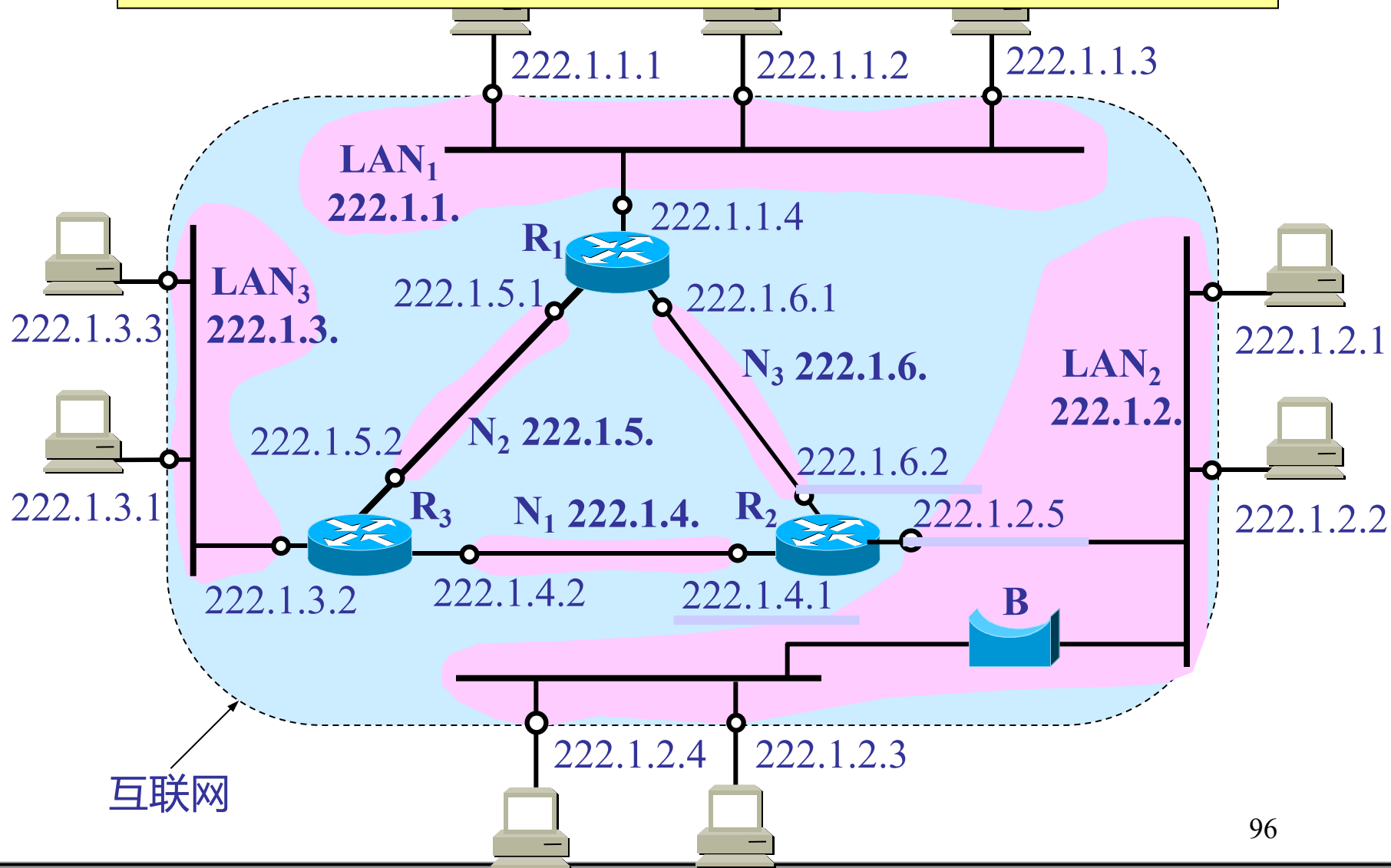
在同一个局域网上的主机或路由器的
IP 地址中的网络号必须是一样的。
图中的网络号就是 IP 地址中的 net-id



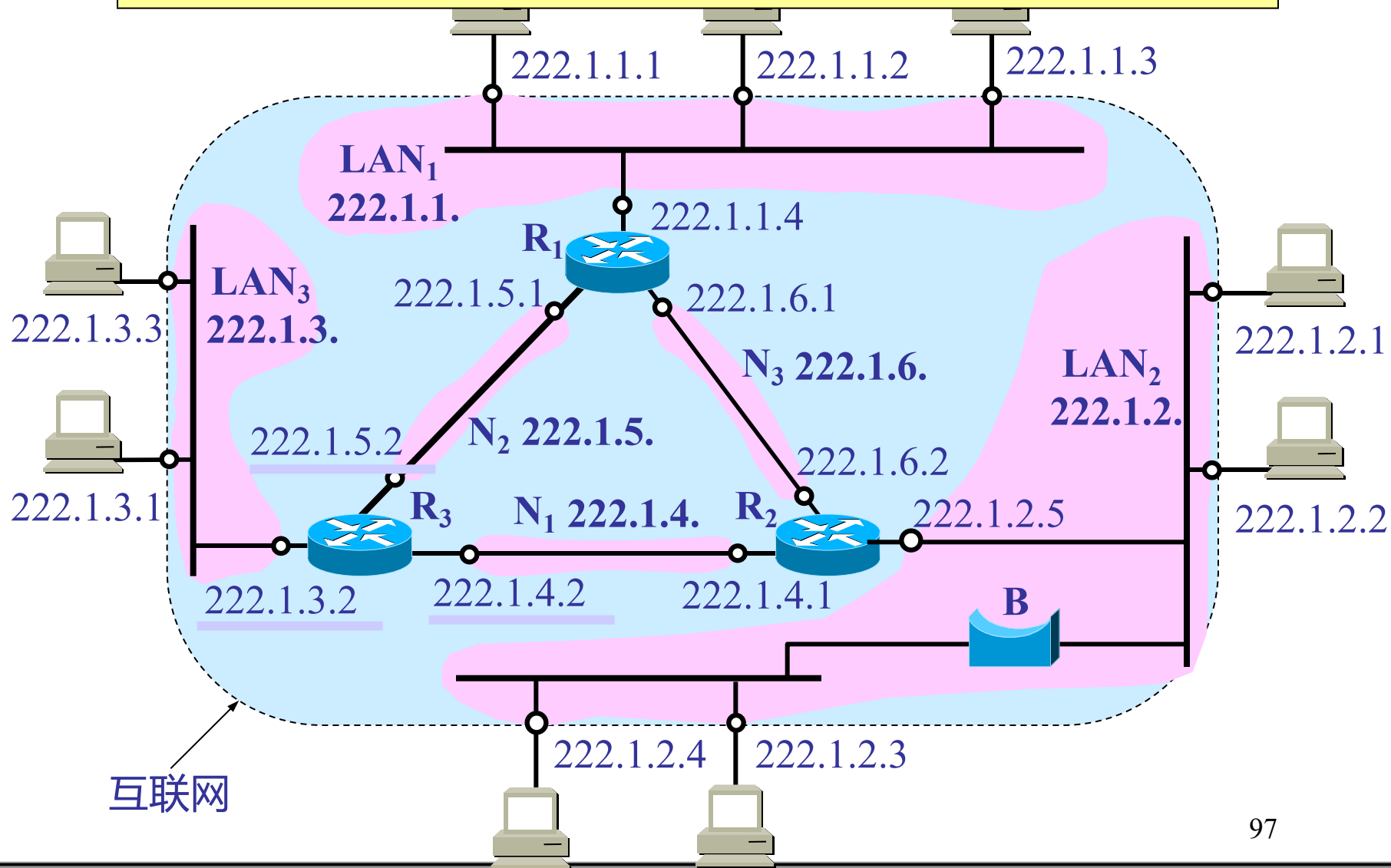
路由器总是具有两个或两个以上的 IP 地址。
路由器的每一个接口都有一个
不同网络号的 IP 地址。



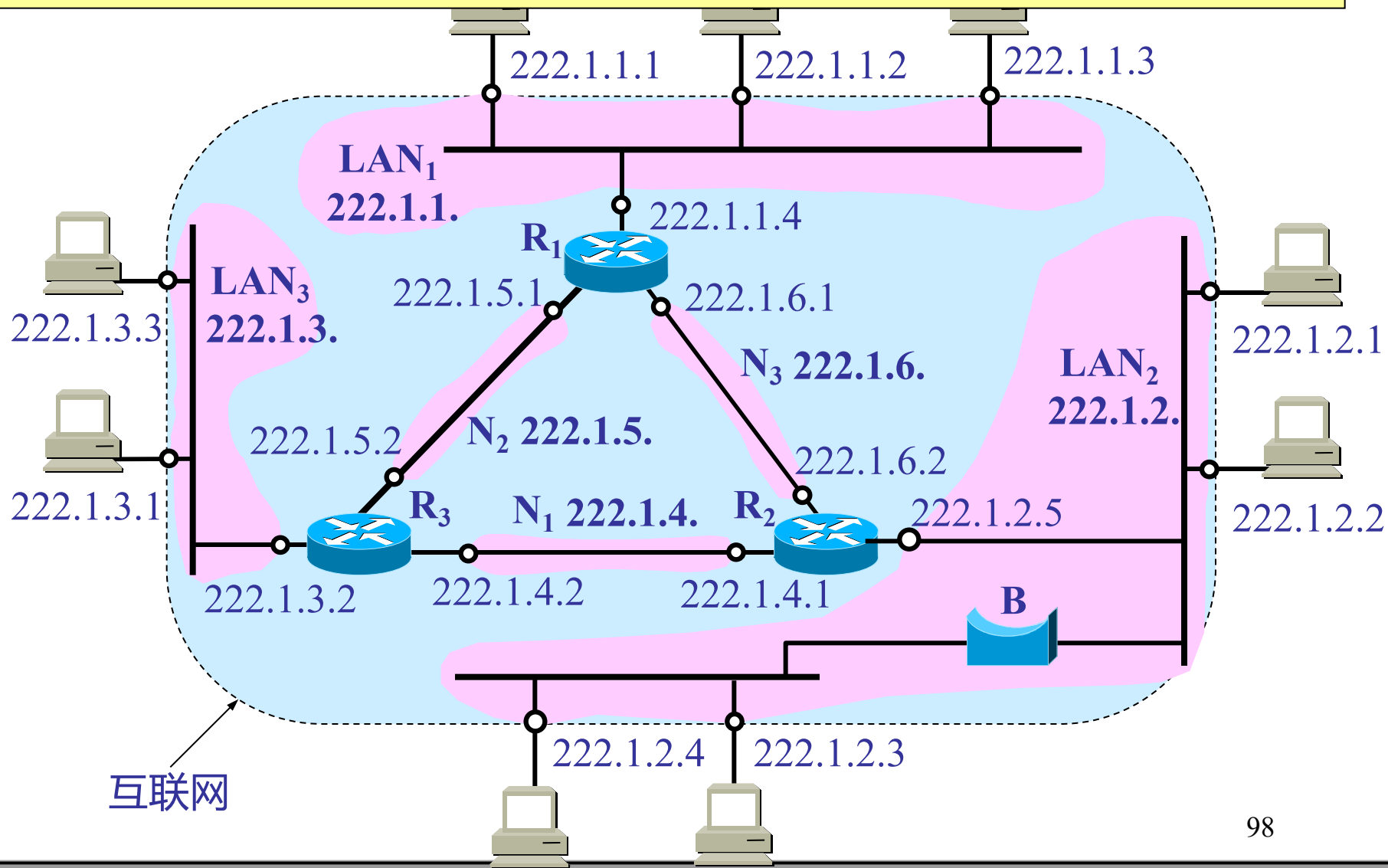
路由器总是具有两个或两个以上的 IP 地址。
路由器的每一个接口都有一个
不同网络号的 IP 地址。



路由器总是具有两个或两个以上的 IP 地址。
路由器的每一个接口都有一个
不同网络号的 IP 地址。



两个路由器直接相连的接口处，可指明也可不指明 IP 地址。如指明 IP 地址，则这一段连线就构成了一种只包含一段线路的特殊“网络”。现在常不指明 IP 地址。



4.3.2 IP编址

从两级 IP 地址到三级 IP 地址

在 ARPANET 的早期，IP 地址的设计确实不够合理。

- ◆ IP 地址空间的利用率有时很低。
- ◆ 给每一个物理网络分配一个网络号会使路由表变得太大因而使网络性能变坏。
- ◆ 两级的 IP 地址不够灵活。

4.3.2 IP编址

三级的 IP 地址

- 从 1985 年起在 IP 地址中又增加了一个“子网号字段”，使两级的 IP 地址变成为三级的 IP 地址。
- 这种做法叫作划分子网(subnetting)。划分子网已成为因特网的正式标准协议。

4.3.2 IP编址

划分子网的基本思路

- 划分子网纯属一个单位内部的事情。单位对外仍然表现为没有划分子网的网络。
- 从主机号借用若干个位作为子网号 subnet-id , 而主机号 host-id 也就相应减少了若干个位。

IP地址 ::= {<网络号>, <子网号>, <主机号>}

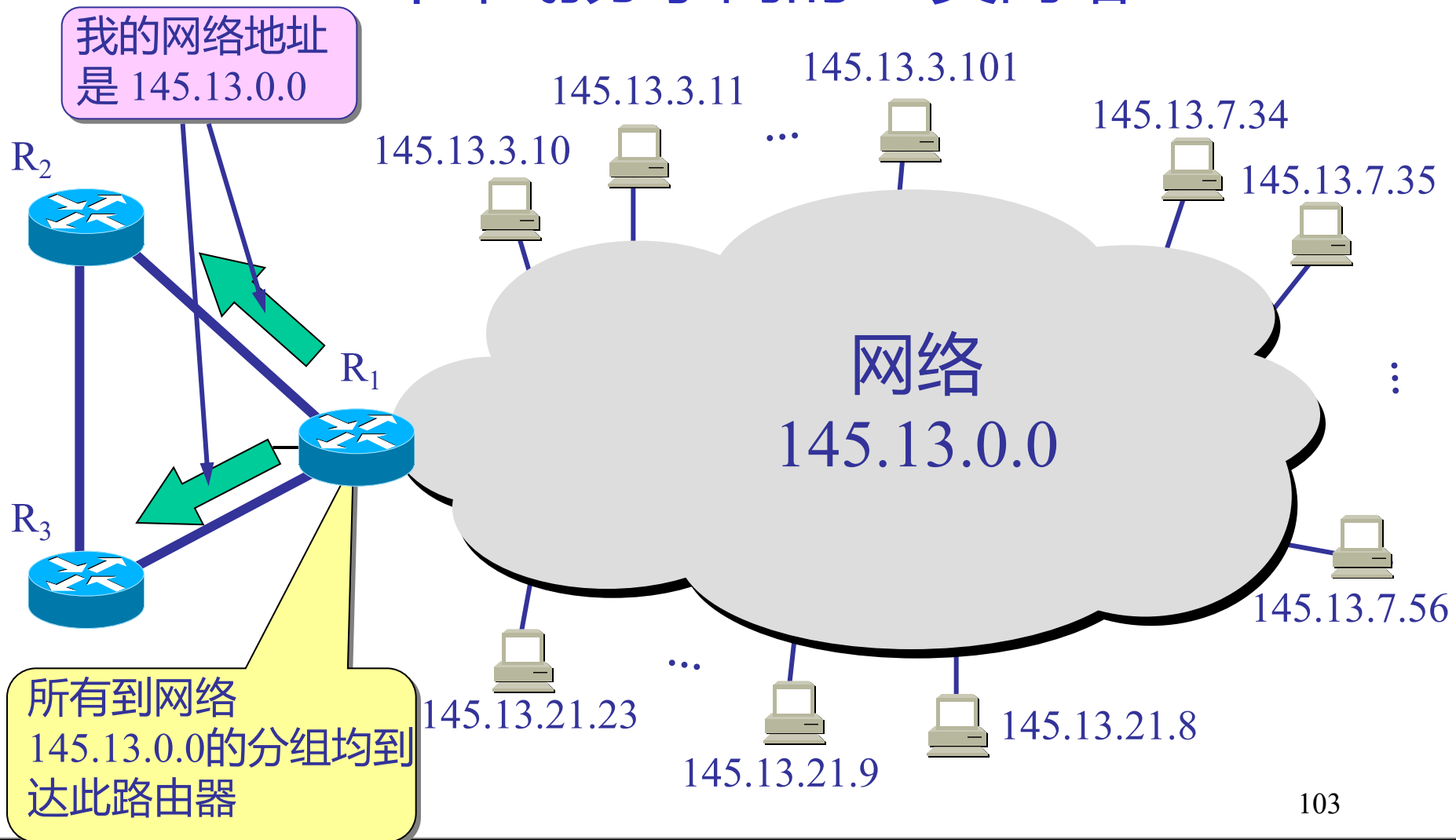
4.3.2 IP编址

划分子网的基本思路（续）

- 凡是从其他网络发送给本单位某个主机的 IP 数据报，仍然是根据 IP 数据报的**目的网络号** net-id，先找到连接在**本单位网络上的路由器**。
- 然后**此路由器**在收到 IP 数据报后，再按目的网络号 net-id 和子网号 subnet-id 找到目的子网。
- 最后就将 IP 数据报直接交付目的主机。

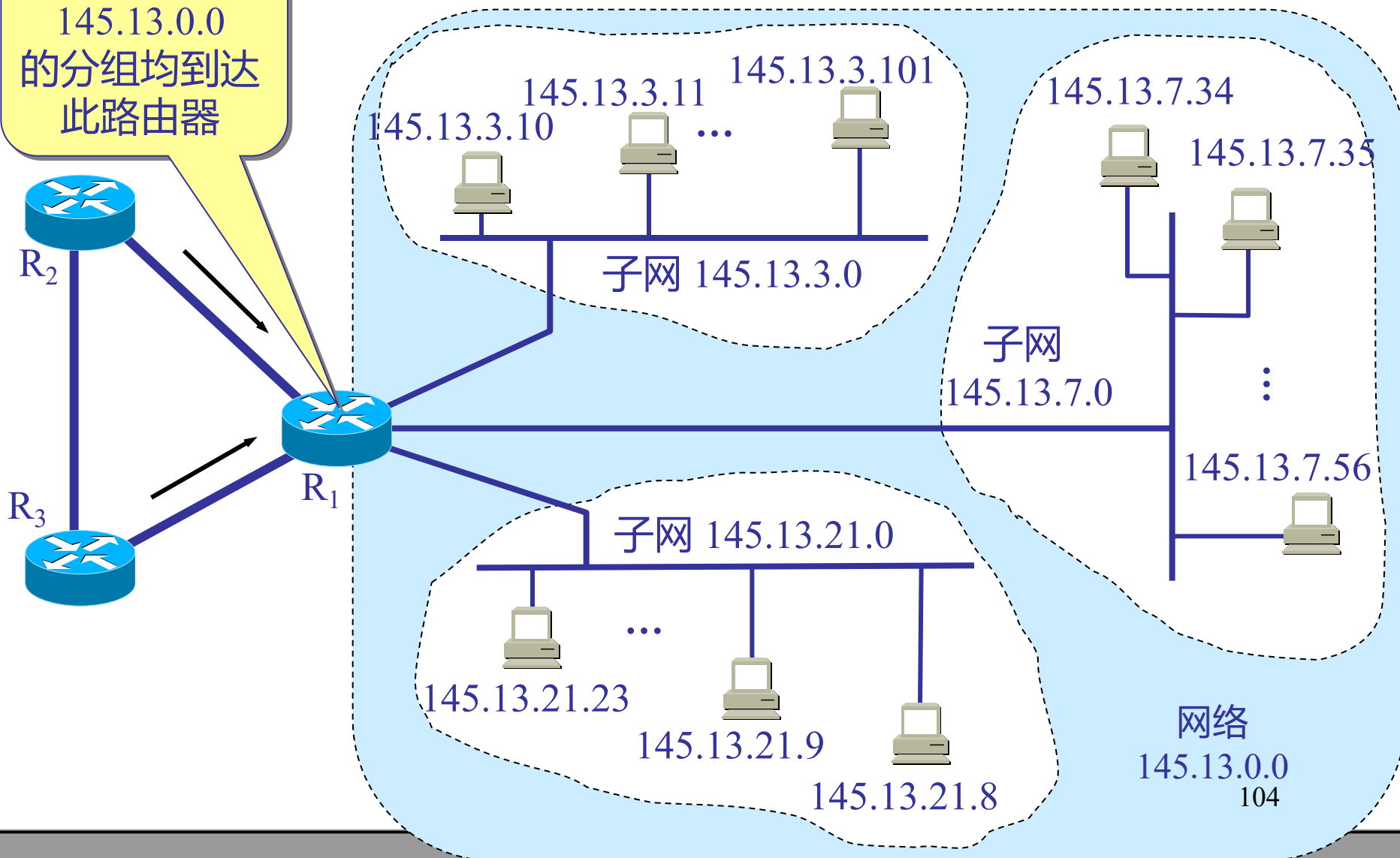
4.3.2 IP编址

一个未划分子网的 B 类网络145.13.0.0



4.3.2 IP编址

所有到达网络
145.13.0.0
的分组均到达
此路由器



4.3.2 IP编址

划分子网后变成了三级结构

- 当没有划分子网时，IP 地址是两级结构。
- 划分子网后 IP 地址就变成了三级结构。
- 划分子网只是把 IP 地址的主机号 host-id 这部分进行再划分，而不改变 IP 地址原来的网络号 net-id。

4.3.2 IP编址

子网掩码

- 从一个 IP 数据报的首部并**无法判断**源主机或目的主机所连接的网络是否进行了子网划分。
- 使用**子网掩码**(subnet mask)可以找出 IP 地址中的子网部分。

4.3.2 IP编址

IP 地址的各字段和子网掩码

| | | | | |
|---------------|---------------------------------|--|---------|-----------------|
| | ← 网络号 → | | ← 主机号 → | |
| 两级 IP 地址 | 145 . 13 . | | 3 . 10 | |
| | ← 网络号 → | | ← 子网号 → | ← 主机号 → |
| 三级 IP 地址 | 145 . 13 . | | 3 . | 10 |
| | ← 子网号为 3 的网络的网络号 → | | | ← 主机号 → |
| 三级 IP 地址的子网掩码 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | | | 0 0 0 0 0 0 0 0 |
| 子网的网络地址 | 145 . 13 . | | 3 | 0 |

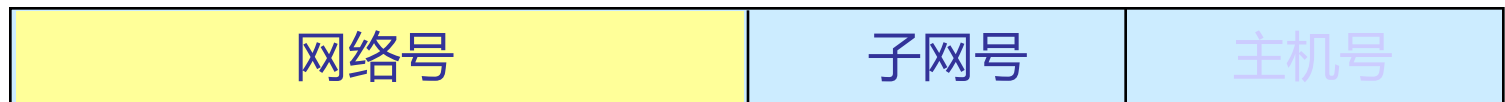
4.3.2 IP编址

(IP 地址) AND (子网掩码) = 网络地址

两级 IP 地址

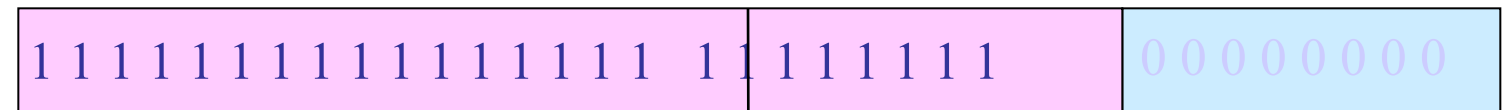


三级 IP 地址

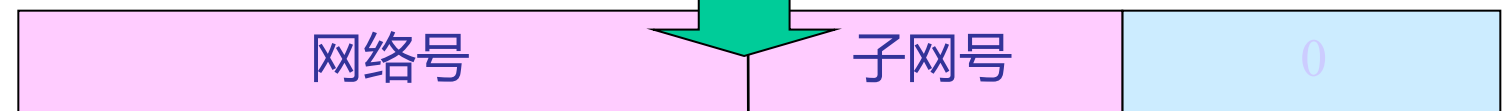


逐位进行 AND 运算

三级 IP 地址
的子网掩码



子网的
网络地址



4.3.2 IP编址

| 默认子网掩码 | | | |
|--------|-------------------------|---|---------|
| A类地址 | 网络地址 | 网络号 | 主机号为全 0 |
| | 默认子网掩码 255.0.0.0 | 1 1 1 1 1 1 1 1 0 | |
| B类地址 | 网络地址 | 网络号 | 主机号为全 0 |
| | 默认子网掩码 255.255.0.0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | |
| C类地址 | 网络地址 | 网络号 | 主机号为全 0 |
| | 默认子网掩码 255.255.255.0 | 1 0 0 0 0 0 0 0 0 | |

4.3.2 IP编址

子网掩码是一个重要属性

- 路由器在和相邻路由器交换路由信息时，必须把自己所在网络（或子网）的子网掩码告诉相邻路由器。
- 路由器的路由表中的每一个项目，除了要给出目的网络地址外，还必须同时给出该网络的子网掩码。

【例】已知 IP 地址是 141.14.72.24，子网掩码是 255.255.192.0。试求网络地址。

(a) 点分十进制表示的 IP 地址

| | | | | | | |
|-----|---|----|---|----|---|----|
| 141 | . | 14 | . | 72 | . | 24 |
|-----|---|----|---|----|---|----|

(b) IP 地址的第 3 字节是二进制

| | | | | | | |
|-----|---|----|---|----------|---|----|
| 141 | . | 14 | . | 01001000 | . | 24 |
|-----|---|----|---|----------|---|----|

(c) 子网掩码是 255.255.192.0

| | | | |
|----------|----------|----------|----------|
| 11111111 | 11111111 | 11000000 | 00000000 |
|----------|----------|----------|----------|

(d) IP 地址与子网掩码逐位相与

| | | | | | | |
|-----|---|----|---|----------|---|---|
| 141 | . | 14 | . | 01000000 | . | 0 |
|-----|---|----|---|----------|---|---|

(e) 网络地址（点分十进制表示）

| | | | | | | |
|-----|---|----|---|----|---|---|
| 141 | . | 14 | . | 64 | . | 0 |
|-----|---|----|---|----|---|---|

4.3.2 IP编址

使用子网掩码的分组转发过程

- 在不划分子网的两级 IP 地址下，从 IP 地址得出网络地址是个很简单的事。
- 但在划分子网的情况下，从 IP 地址却不能唯一地得出网络地址来，这是因为网络地址取决于那个网络所采用的子网掩码，但数据报的首部并没有提供子网掩码的信息。
- 因此分组转发的算法也必须做相应的改动。

4.3.2 IP编址

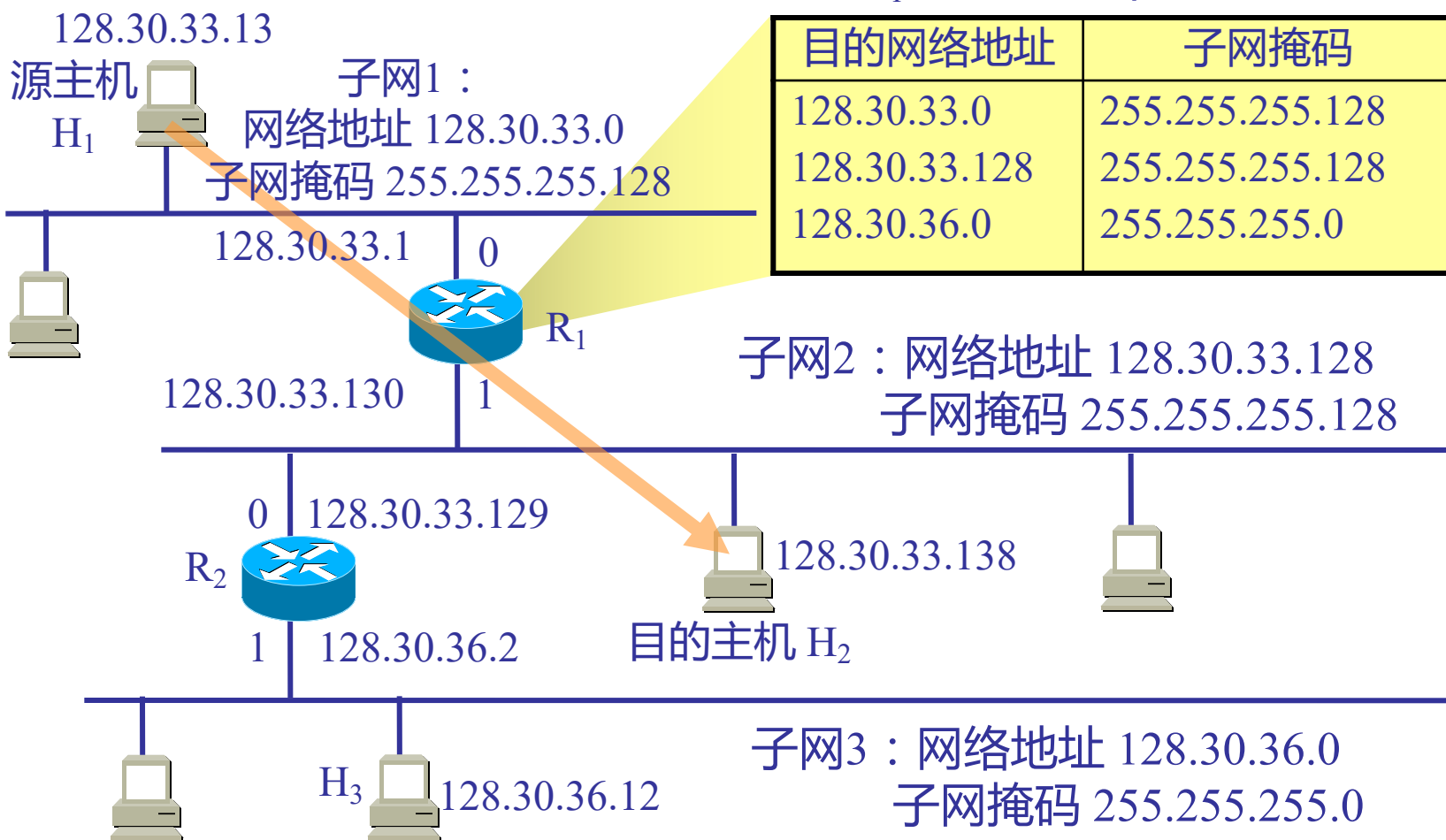
在划分子网的情况下路由器转发分组的算法

- (1) 从收到的分组的首部提取目的 IP 地址 D 。
- (2) 先用各网络的子网掩码和 D 逐位相“与”，找到匹配的
网络地址。

【例】已知互联网和路由器 R_1 中的路由表。主机 H_1 向 H_2 发送分组。试讨论 R_1 收到 H_1 向 H_2 发送的分组后查找路由表的过程。

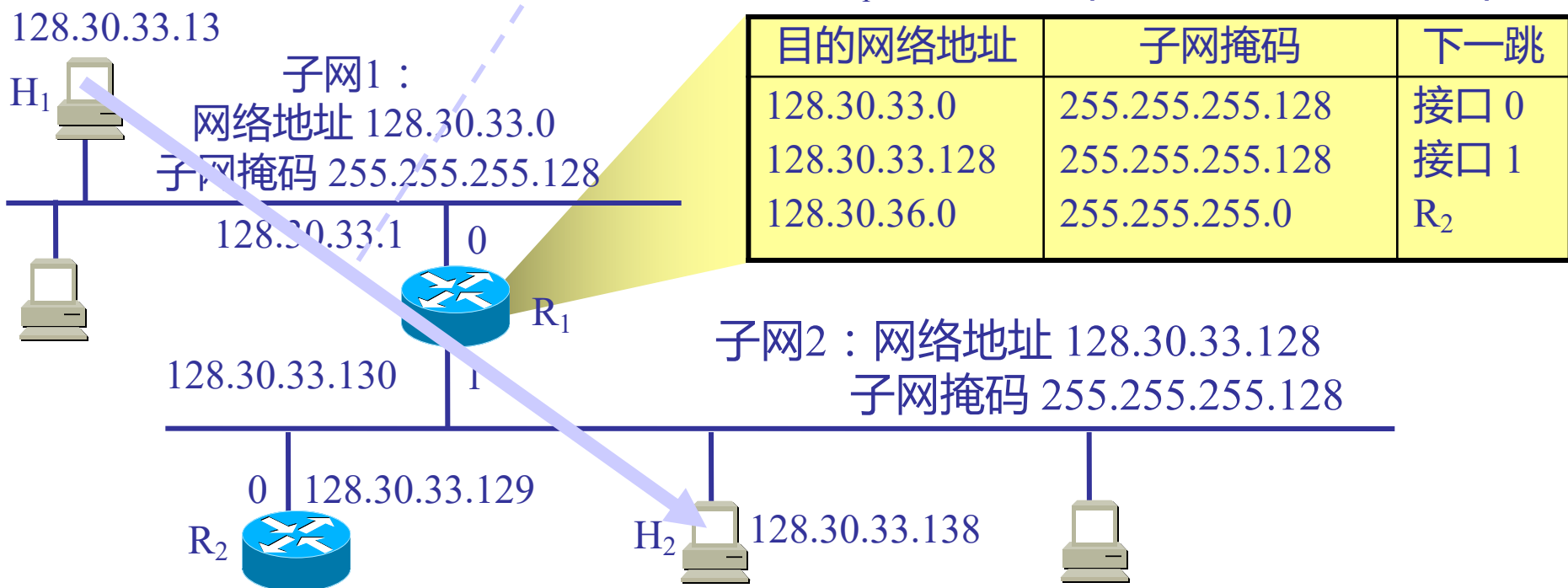
R_1 的路由表（未给出默认路由器）

| 目的网络地址 | 子网掩码 | 下一跳 |
|---------------|-----------------|-------|
| 128.30.33.0 | 255.255.255.128 | 接口 0 |
| 128.30.33.128 | 255.255.255.128 | 接口 1 |
| 128.30.36.0 | 255.255.255.0 | R_2 |



主机 H_1 要发送分组给 H_2

要发送的分组的目的地 IP 地址：128.30.33.138

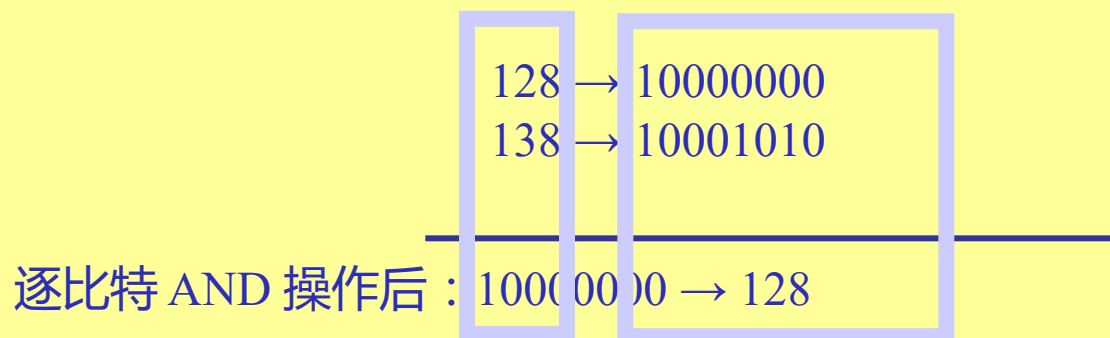


因此 H_1 首先检查主机 128.30.33.138 是否连接在本网络上
如果是，则直接交付；
否则，就送交路由器 R_1 ，并逐项查找路由表。

主机 H_1 首先将
本子网的子网掩码 255.255.255.128
与分组的 IP 地址 128.30.33.138 逐比特相 “与” (AND 操作)

255.255.255.128 AND 128.30.33.138 的计算

255 就是二进制的全 1，因此 $255 \text{ AND } xyz = xyz$ ，
这里只需计算最后的 128 AND 138 即可。

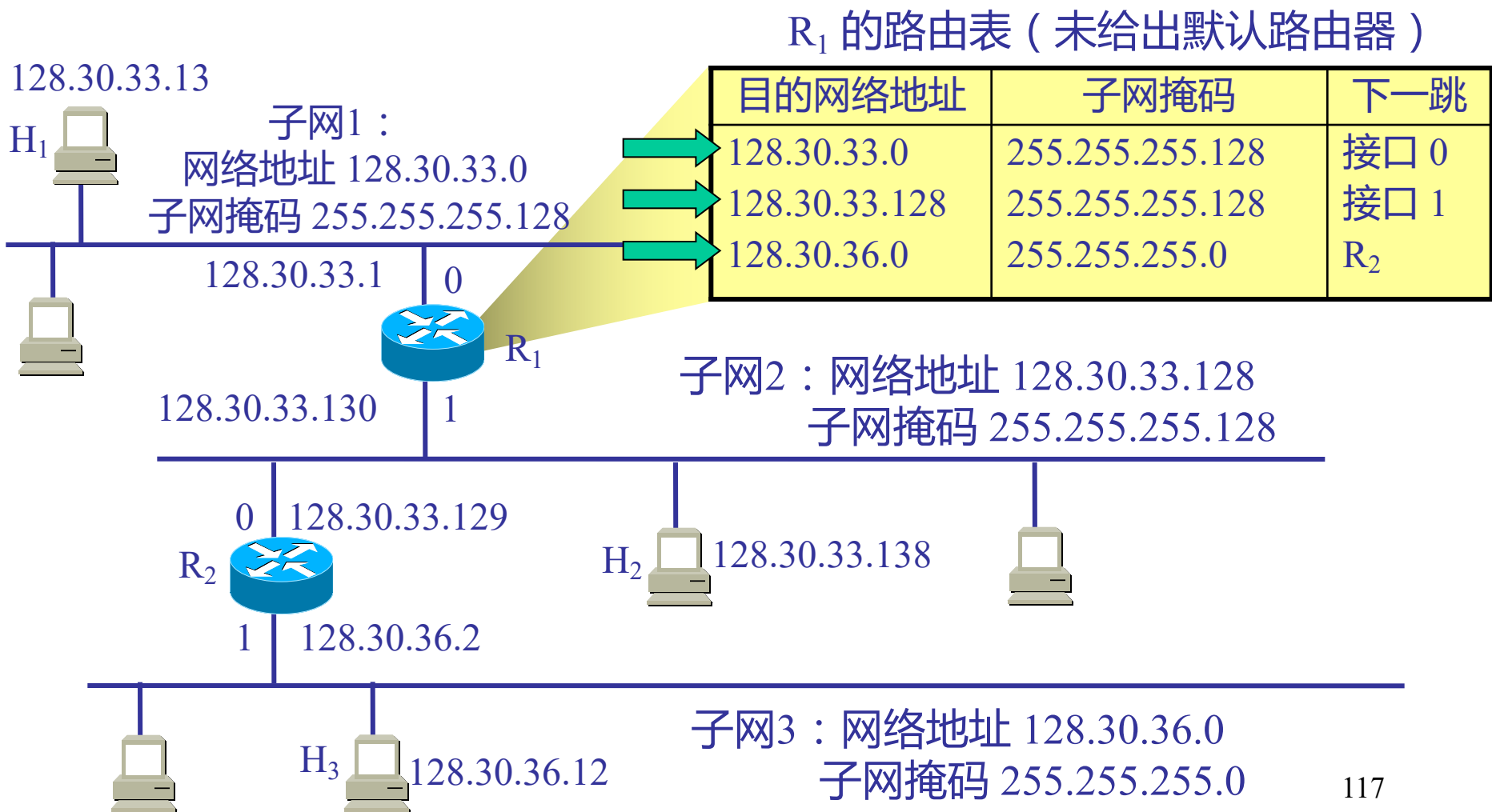


逐比特 AND 操作

| |
|-----------------|
| 255.255.255.128 |
| 128. 30. 33.138 |
| 128. 30. 33.128 |

H_1 的网络地址：
128.30.33.0

因此 H_1 必须把分组传送到路由器 R_1
然后逐项查找路由表

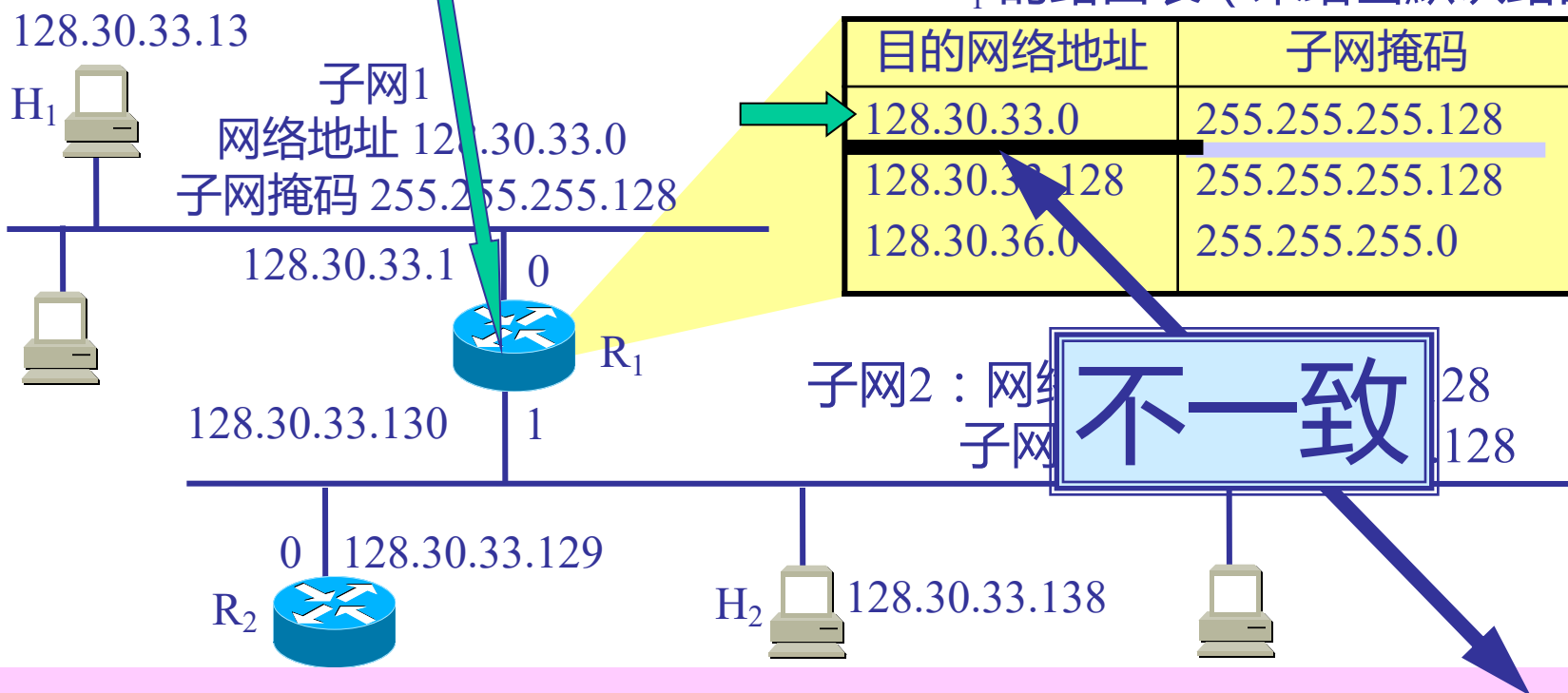


路由器 R₁ 收到分组后就用路由表中第 1 个项目的子网掩码和 128.30.33.138 逐比特 AND 操作

R₁ 收到的分组的目的 IP 地址：128.30.33.138

R₁ 的路由表（未给出默认路由器）

| 目的网络地址 | 子网掩码 | 下一跳 |
|---------------|-----------------|----------------|
| 128.30.33.0 | 255.255.255.128 | 接口 0 |
| 128.30.33.128 | 255.255.255.128 | 接口 1 |
| 128.30.36.0 | 255.255.255.0 | R ₂ |



$255.255.255.128 \text{ AND } 128.30.33.138 = 128.30.33.128$

不匹配!

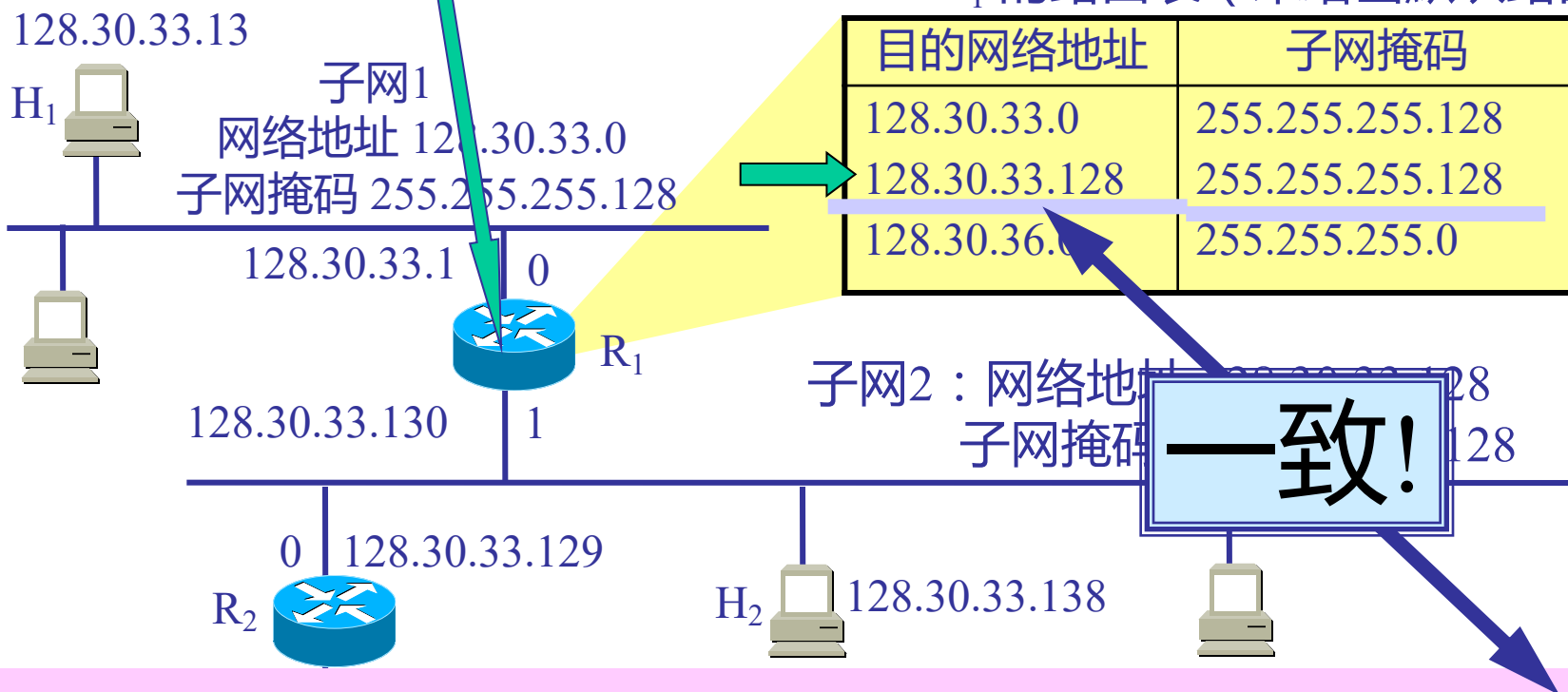
(因为 128.30.33.128 与路由表中的 128.30.33.0 不一致)

路由器 R₁ 再用路由表中第 2 个项目的子网掩码和 128.30.33.138 逐比特 AND 操作

R₁ 收到的分组的目的 IP 地址：128.30.33.138

R₁ 的路由表（未给出默认路由器）

| 目的网络地址 | 子网掩码 | 下一跳 |
|---------------|-----------------|----------------|
| 128.30.33.0 | 255.255.255.128 | 接口 0 |
| 128.30.33.128 | 255.255.255.128 | 接口 1 |
| 128.30.36.0 | 255.255.255.0 | R ₂ |



255.255.255.128 AND 128.30.33.138 = 128.30.33.128

匹配!

这表明子网 2 就是收到的分组所要寻找的目的网络

4.3.2 IP编址

无分类编址 CIDR

划分子网在一定程度上缓解了因特网在发展中遇到的困难。然而在 1992 年因特网仍然面临三个必须尽早解决的问题，这就是：

- B 类地址在 1992 年已分配了近一半，眼看就要在 1994 年 3 月全部分配完毕！
- 因特网主干网上的路由表中的项目数急剧增长（从几千个增长到几万个）。
- 整个 IPv4 的地址空间最终将全部耗尽。

4.3.2 IP编址

- 1987 年，RFC 1009 就指明了在一个划分子网的网络中可同时使用几个不同的子网掩码。使用变长子网掩码 VLSM (Variable Length Subnet Mask)可进一步提高 IP 地址资源的利用率。
- 在 VLSM 的基础上又进一步研究出无分类编址方法，它的正式名字是无分类域间路由选择 CIDR (Classless Inter-Domain Routing)。

4.3.2 IP编址

CIDR 最主要的特点

- 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。

4.3.2 IP编址

无分类的两级编址

- 无分类的两级编址的记法是：

IP地址 ::= {<网络前缀>, <主机号>}

- CIDR 还使用“斜线记法” (slash notation)，它又称为CIDR记法，即在 IP 地址面加上一个斜线 “/”，然后写上网络前缀所占的位数（这个数值对应于三级编址中子网掩码中 1 的个数）。
- CIDR 把网络前缀都相同的连续的 IP 地址组成“CIDR 地址块”。

4.3.2 IP编址

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 是网络前缀的位数，所以这个地址的主机号是 12 位）。
- 这个地址块的起始地址是 128.14.32.0。
- 在不需要指出地址块的起始地址时，也可将这样的地址块简称为 “/20 地址块”。
- 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。

4.3.2 IP编址

CIDR 记法的其他形式

- 10.0.0.0/10 可简写为 10/10，也就是把点分十进制中低位连续的 0 省略。
- 10.0.0.0/10 隐含地指出 IP 地址 10.0.0.0 的掩码是 255.192.0.0。此掩码可表示为

| | | | |
|--------------|----------|--------------|----------|
| 11111111 | 11000000 | 00000000 | 00000000 |
| └──────────┘ | | └──────────┘ | |
| 255 | | 192 | |
| | | 0 | |
| | | 0 | |

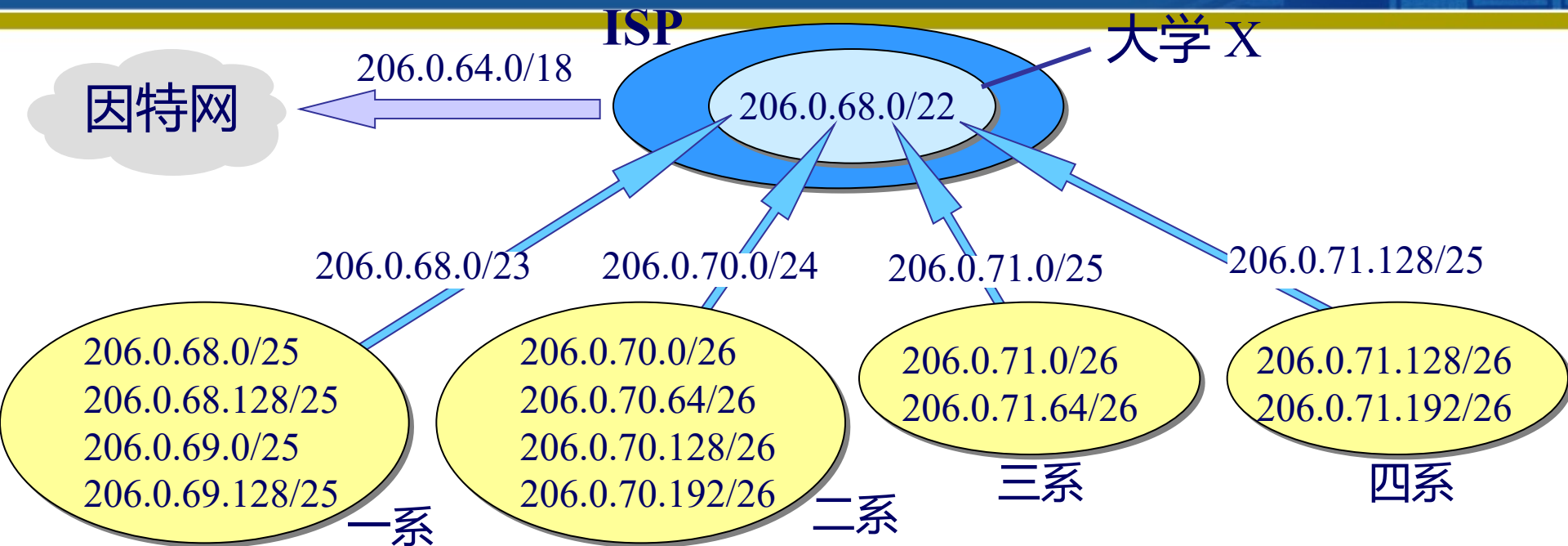
掩码中有 10 个连续的 1

4.3.2 IP编址

CIDR 记法的其他形式

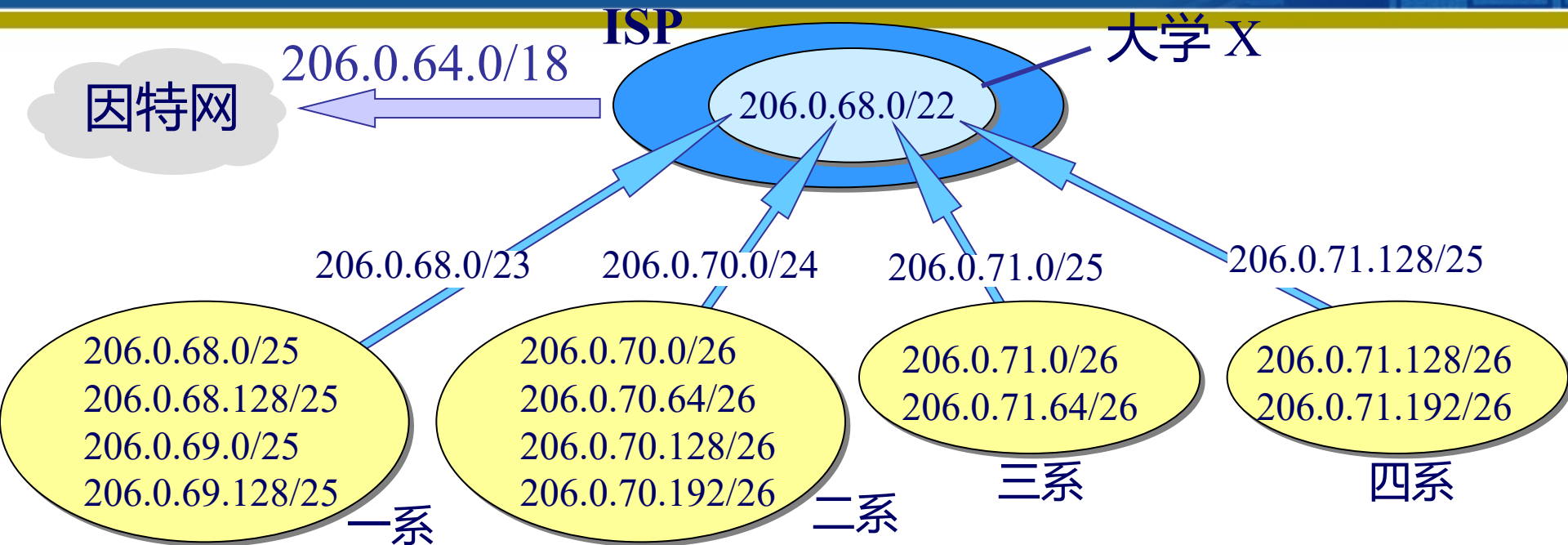
- 10.0.0.0/10 可简写为 10/10，也就是将点分十进制中低位连续的 0 省略。
- 10.0.0.0/10 相当于指出 IP 地址 10.0.0.0 的掩码是 255.192.0.0，即
11111111 11000000 00000000 00000000
- 网络前缀的后面加一个星号 * 的表示方法
如 00001010 00*，在星号 * 之前是网络前缀，而星号 * 表示 IP 地址中的主机号，可以是任意值。

CIDR 地址块划分举例



| 单位 | 地址块 | 二进制表示 | 地址数 |
|-----|-----------------|-------------------------------|-------|
| ISP | 206.0.64.0/18 | 11001110.00000000.01* | 16384 |
| 大学 | 206.0.68.0/22 | 11001110.00000000.010001* | 1024 |
| 一系 | 206.0.68.0/23 | 11001110.00000000.0100010* | 512 |
| 二系 | 206.0.70.0/24 | 11001110.00000000.01000110.* | 256 |
| 三系 | 206.0.71.0/25 | 11001110.00000000.01000111.0* | 128 |
| 四系 | 206.0.71.128/25 | 11001110.00000000.01000111.1* | 128 |

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。

4.3.2 IP编址

最长前缀匹配

- 使用 CIDR 时，路由表中的每个项目由“网络前缀”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网络前缀的路由：**最长前缀匹配**(longest-prefix matching)。
- 网络前缀越长，其地址块就越小，因而路由就越具体(more specific)。
- 最长前缀匹配又称为**最长匹配**或**最佳匹配**。

4.3.2 IP编址

收到的分组的目的地地址 $D = 206.0.71.128$

路由表中的项目： $206.0.68.0/22$ (ISP)
 $206.0.71.128/25$ (四系)

$D \text{ AND } (11111111 \ 11111111 \ 11111100 \ 00000000)$
 $= 206.0.68.0/22$ 匹配

$D \text{ AND } (11111111 \ 11111111 \ 11111111 \ 10000000)$
 $= 206.0.71.128/25$ 匹配

- 选择两个匹配的地址中更具体的一个，即选择最长前缀的地址。

4.3.2 IP编址：例子

- 假定某ISP拥有地址区间20.11.0.0/16，现需平均分配给8个公司，请选择合适的子网掩码，进行子网划分，写出前四个子网的网络号，子网掩码及IP地址起始范围。

| 网络号 | 子网掩码 | IP地址起始 |
|------------|----------------------------------|------------|
| 20.11.0.0 | 20.11.0.0/19 or 255.255.224.0 | 20.11.0.1 |
| 20.11.32.0 | 20.11.32.0/19 | 20.11.32.1 |
| 20.11.64.0 | 20.11.64.0/19 | 20.11.64.1 |
| 20.11.96.0 | 20.11.96.0/19 | 20.11.96.1 |

4.3.2 IP编址：例子

- Consider a router that interconnects three subnets: Subnet 1, Subnet 2, and Subnet 3. Suppose all of the interfaces in each of these three subnets are required to have the prefix 223.1.17/24. Also suppose that Subnet 1 is required to support up to 60 interfaces, Subnet 2 is to support up to 90 interfaces, and Subnet 3 is to support up to 12 interfaces. Provide three network addresses (of the form a.b.c.d/x) that satisfy these constraints.

4.3.2 IP编址

使用二叉线索查找路由表

- 当路由表的项目数很大时，怎样设法减小路由表的查找时间就成为一个非常重要的问题。
- 为了进行更加有效的查找，通常是将无分类编址的路由表存放在一种层次的数据结构中，然后自上而下地按层次进行查找。这里最常用的就是**二叉线索**(binary trie)。
- IP 地址中从左到右的比特值决定了从根结点逐层向下层延伸的路径，而二叉线索中的各个路径就代表路由表中存放的各个地址。
- 为了提高二叉线索的查找速度，广泛使用了各种压缩技术。

4.3.2 IP编址

用 5 个前缀构成的二叉线索

32 位的 IP 地址

唯一前缀

01000110 00000000 00000000 00000000

0100

01010110 00000000 00000000 00000000

0101

01100001 00000000 00000000 00000000

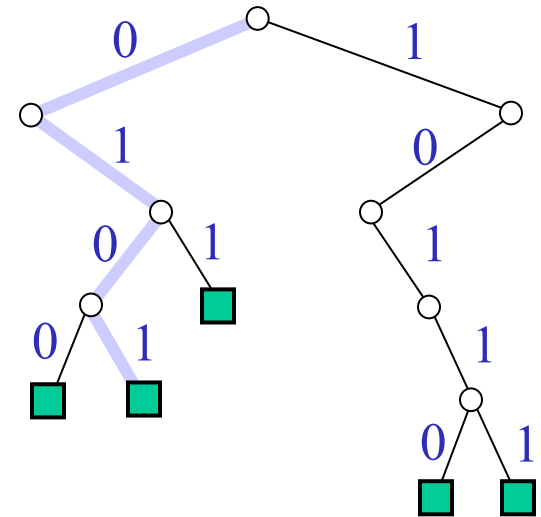
011

10110000 00000010 00000000 00000000

10110

10111011 00001010 00000000 00000000

10111



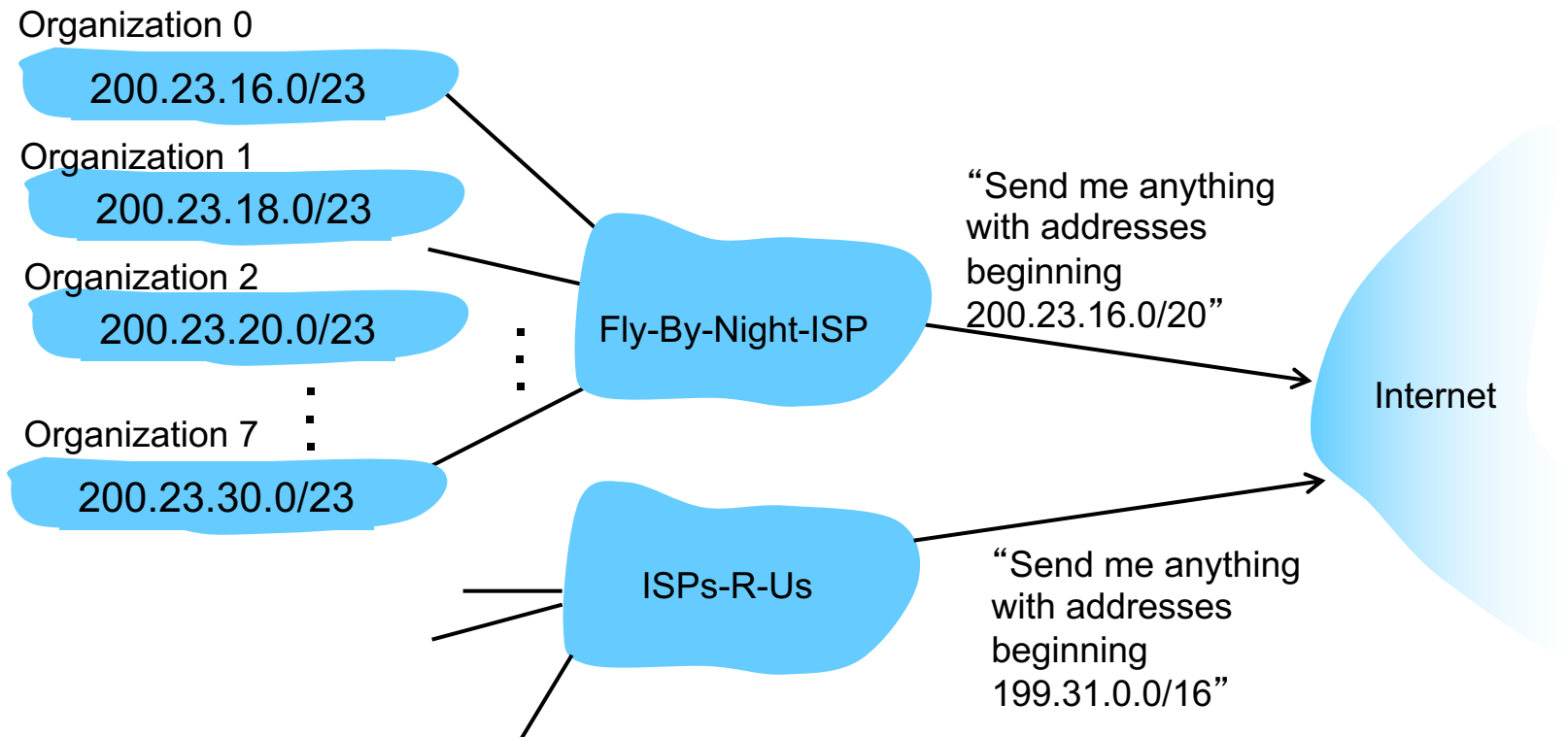
4.3.2 IP编址

路由聚合(route aggregation)

- 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（但不叫子网掩码）。
- 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。

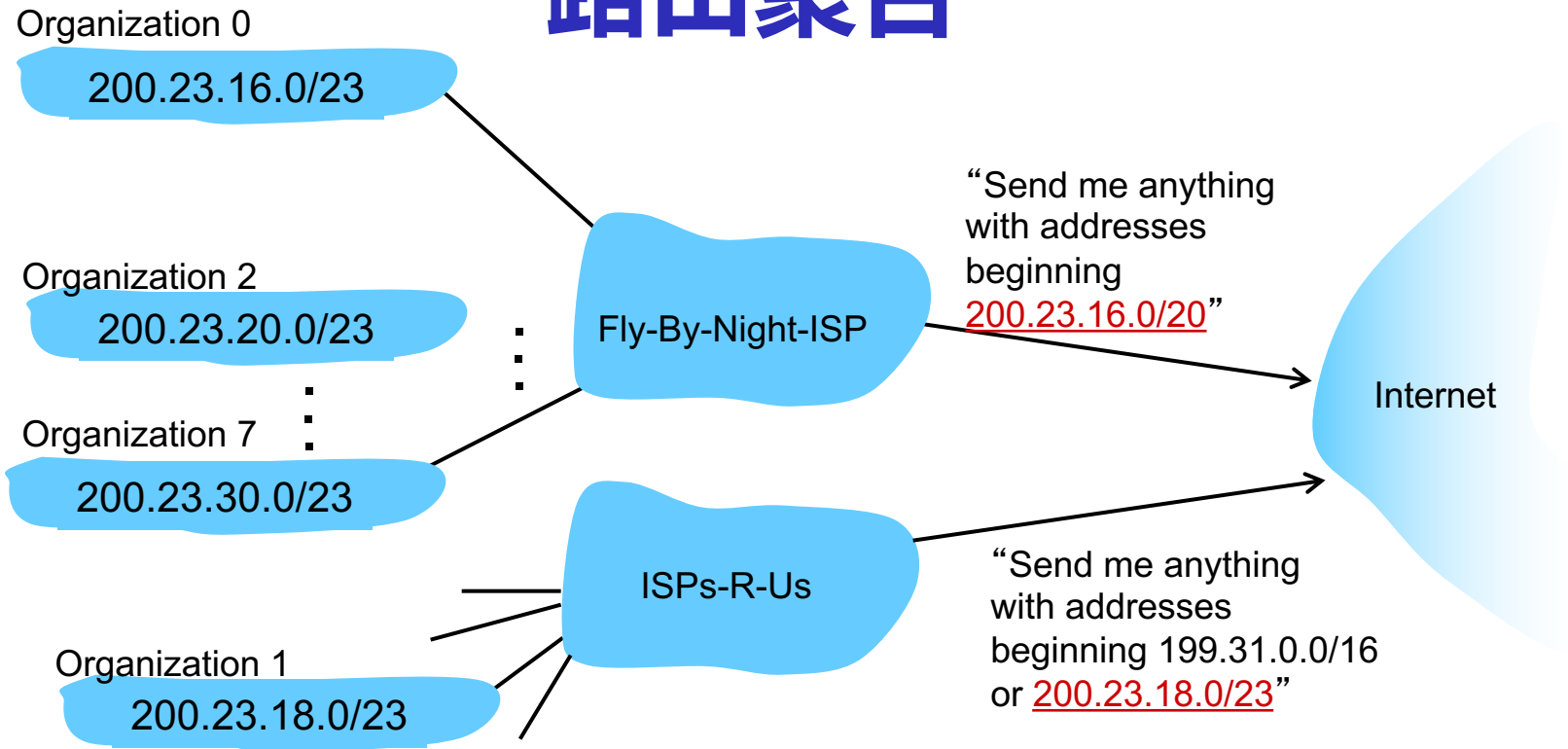
4.3.2 IP编址

路由聚合



4.3.2 IP编址

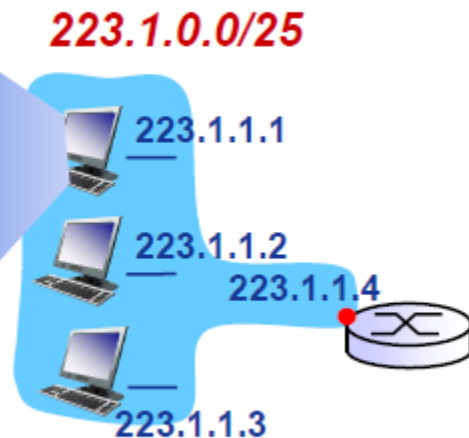
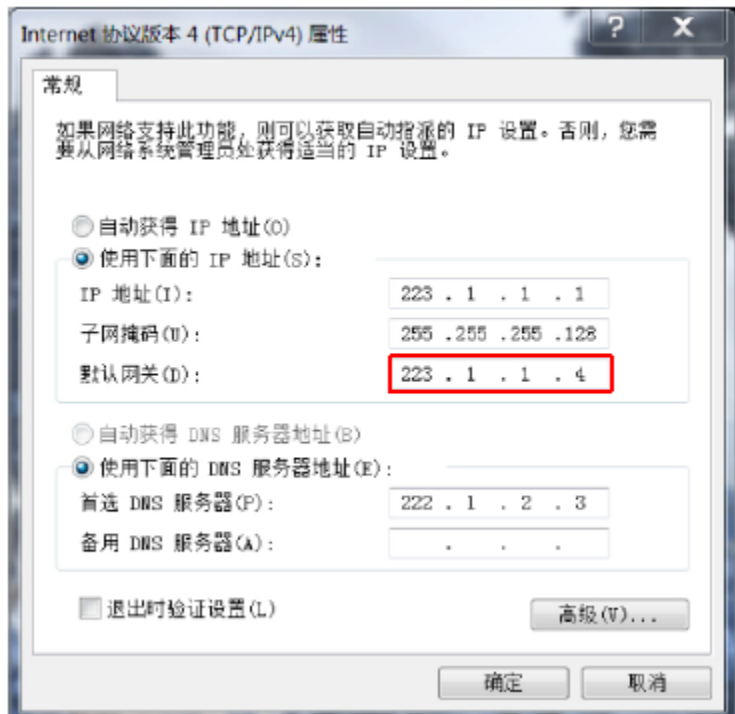
路由聚合



4.3.3 获取IP地址

Q: 一个主机如何获得IP地址？

• 静态配置



4.3.3 获取IP地址

动态主机配置协议 DHCP

(Dynamic Host Configuration Protocol)

- 动态主机配置协议 DHCP 提供了即插即用连网(plug-and-play networking)的机制。
- 这种机制允许一台计算机加入新的网络和获取IP地址而不用手工参与。

4.3.3 获取IP地址

DHCP 使用客户-服务器方式。

- 需要 IP 地址的主机在启动时就向 DHCP 服务器广播发送发现报文（DHCPDISCOVER），这时该主机就成为 DHCP 客户。
- 本地网络上所有主机都能收到此广播报文，但只有 DHCP 服务器才回答此广播报文。
- DHCP 服务器先在其数据库中查找该计算机的配置信息。若找到，则返回找到的信息。若找不到，则从服务器的 IP 地址池(address pool)中取一个地址分配给该计算机。DHCP 服务器的回答报文叫做提供报文（DHCPOFFER）。

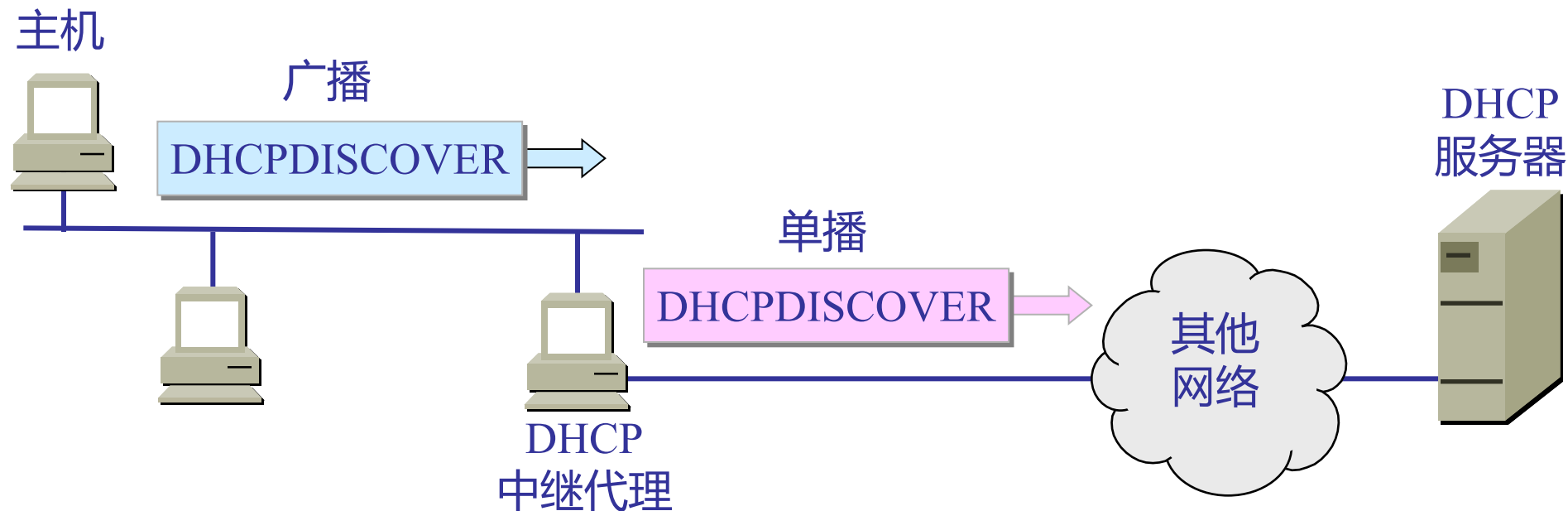
4.3.3 获取IP地址

DHCP 中继代理(relay agent)

- 并不是每个网络上都有 DHCP 服务器，这样会使 DHCP 服务器的数量太多。现在是每一个网络至少有一个 DHCP 中继代理，它配置了 DHCP 服务器的 IP 地址信息。
- 当 DHCP 中继代理收到主机发送的发现报文后，就以单播方式向 DHCP 服务器转发此报文，并等待其回答。收到 DHCP 服务器回答的提供报文后，DHCP 中继代理再将此提供报文发回给主机。

4.3.3 获取IP地址

DHCP 中继代理 以单播方式转发发现报文

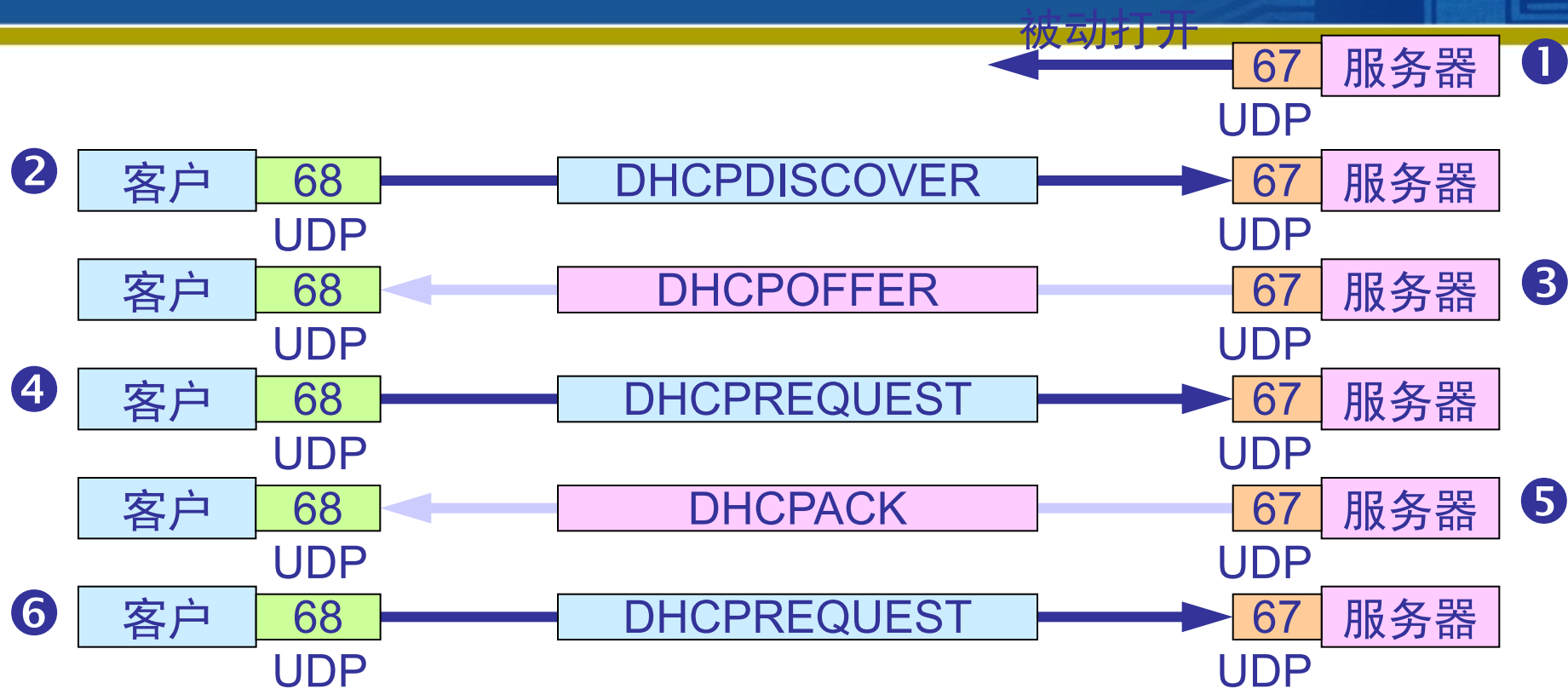


注意：DHCP 报文只是 UDP 用户数据报中的数据。

租用期(lease period)

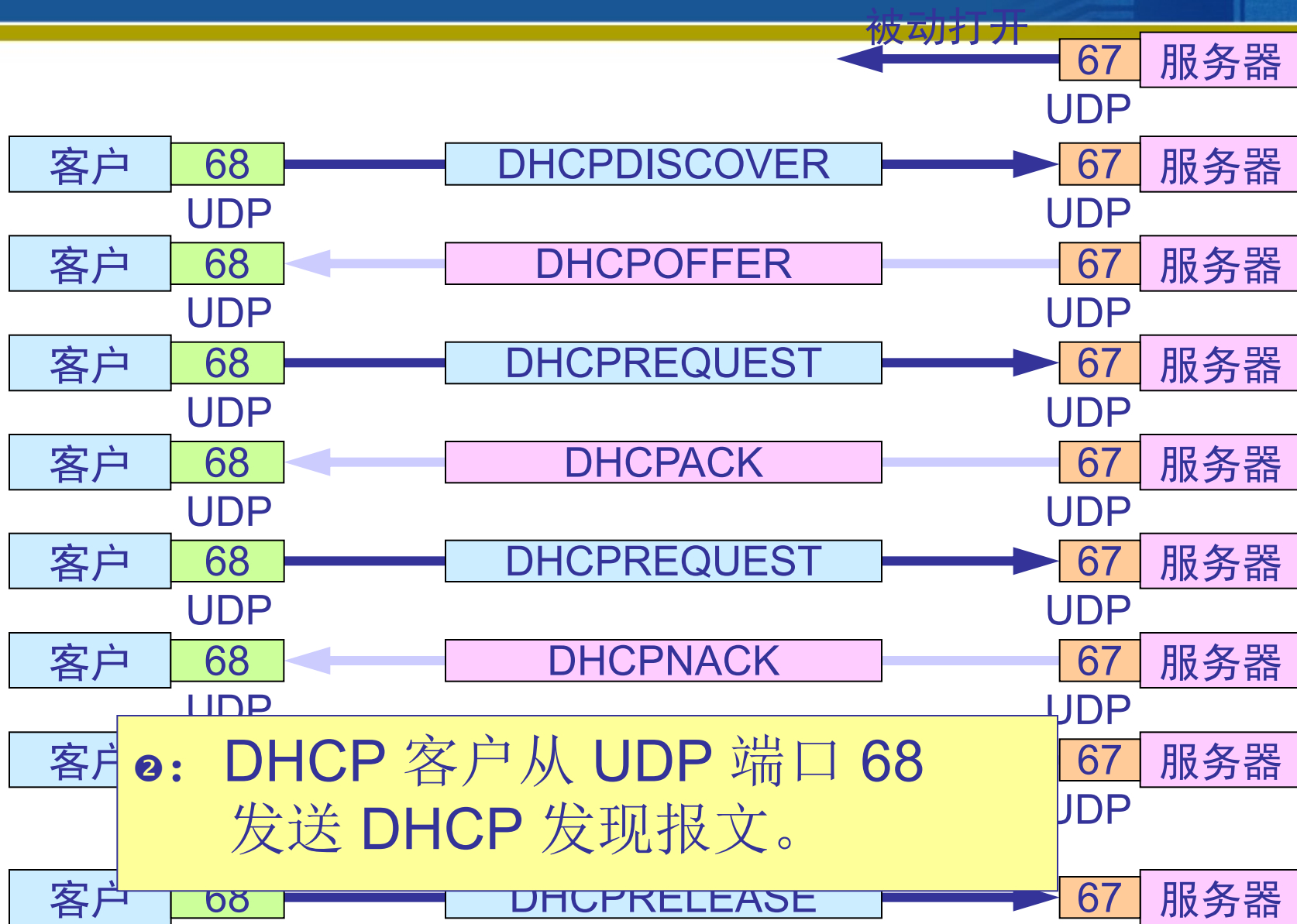
- DHCP 服务器分配给 DHCP 客户的 IP 地址的临时的，因此 DHCP 客户只能在一段有限的时间内使用这个分配到的 IP 地址。DHCP 协议称这段时间为租用期。
- 租用期的数值应由 DHCP 服务器自己决定。
- DHCP 客户也可在自己发送的报文中（例如，发现报文）提出对租用期的要求。

DHCP 协议的工作过程

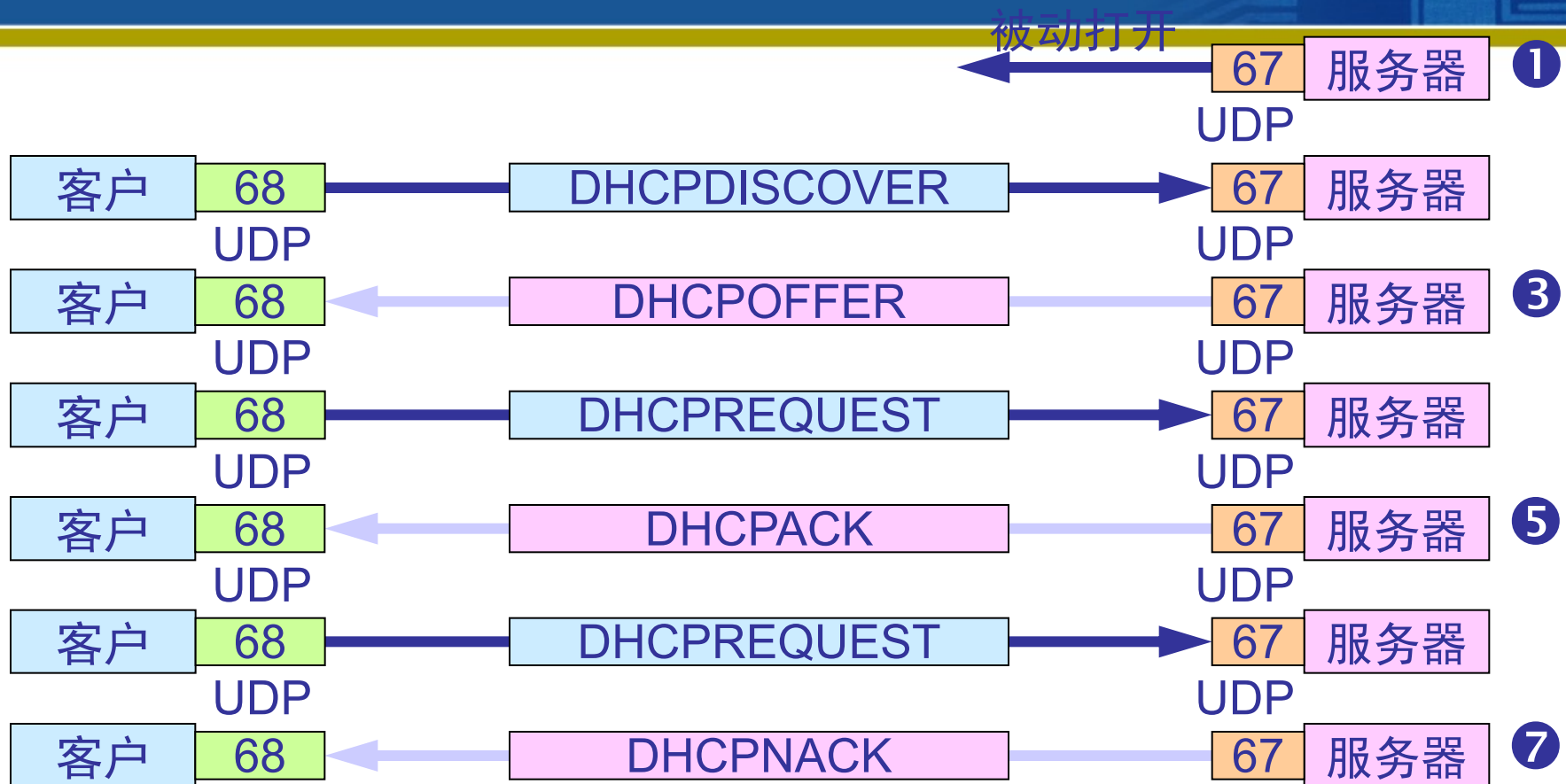


①: DHCP 服务器被动打开 UDP 端口 67, 等待客户端发来的报文。

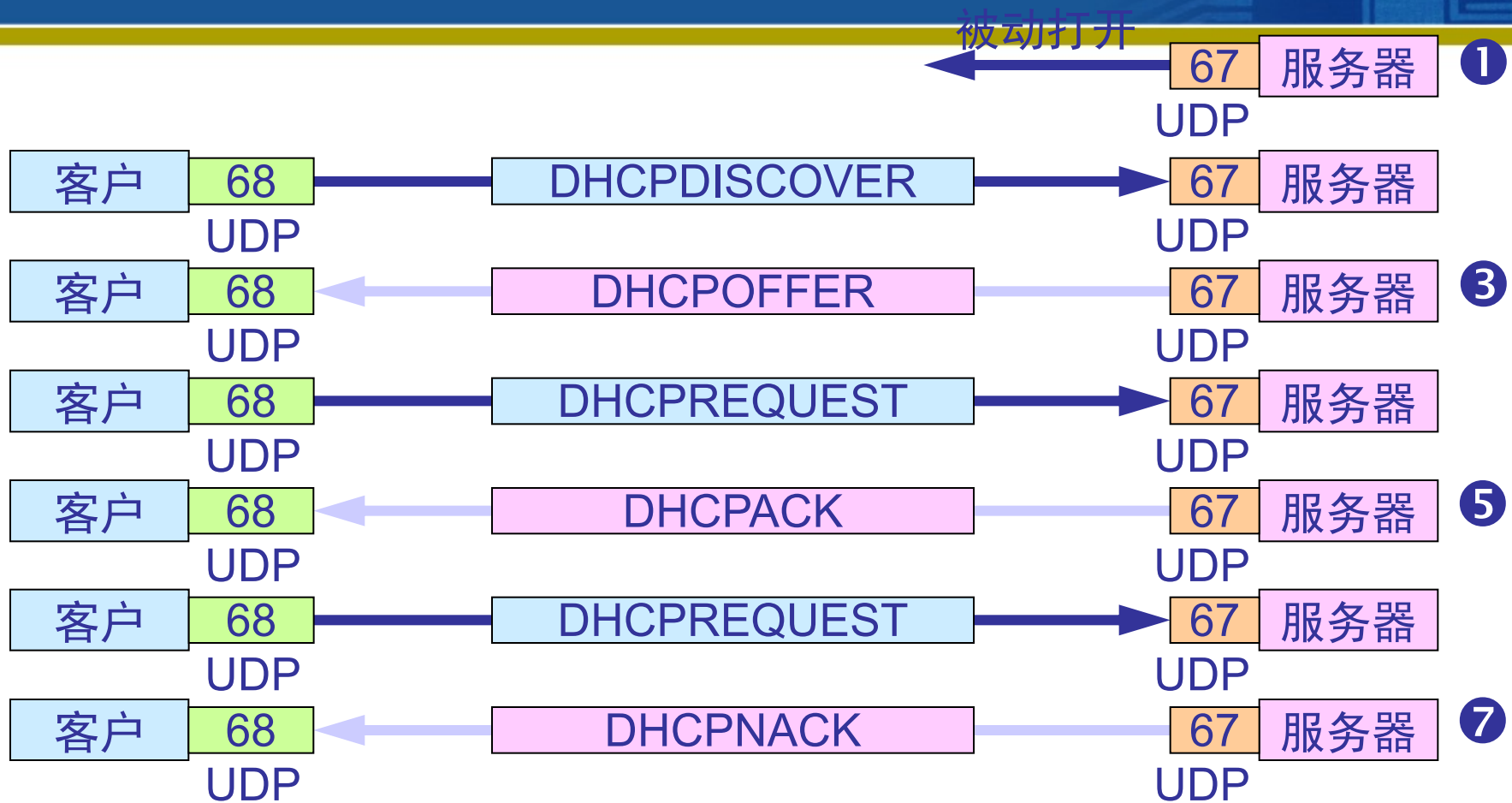
DHCP 协议的工作过程



DHCP 协议的工作过程

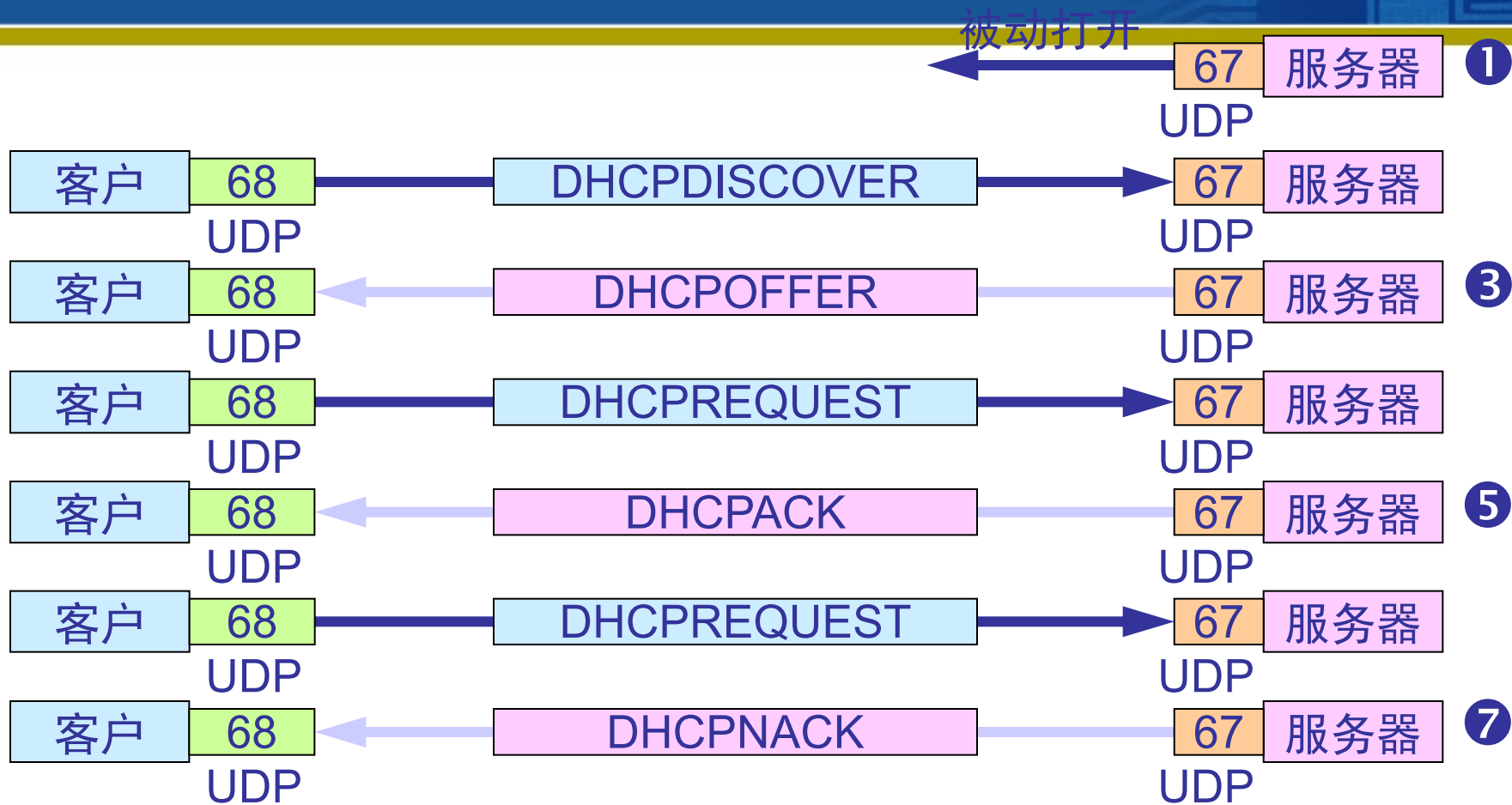


DHCP 协议的工作过程



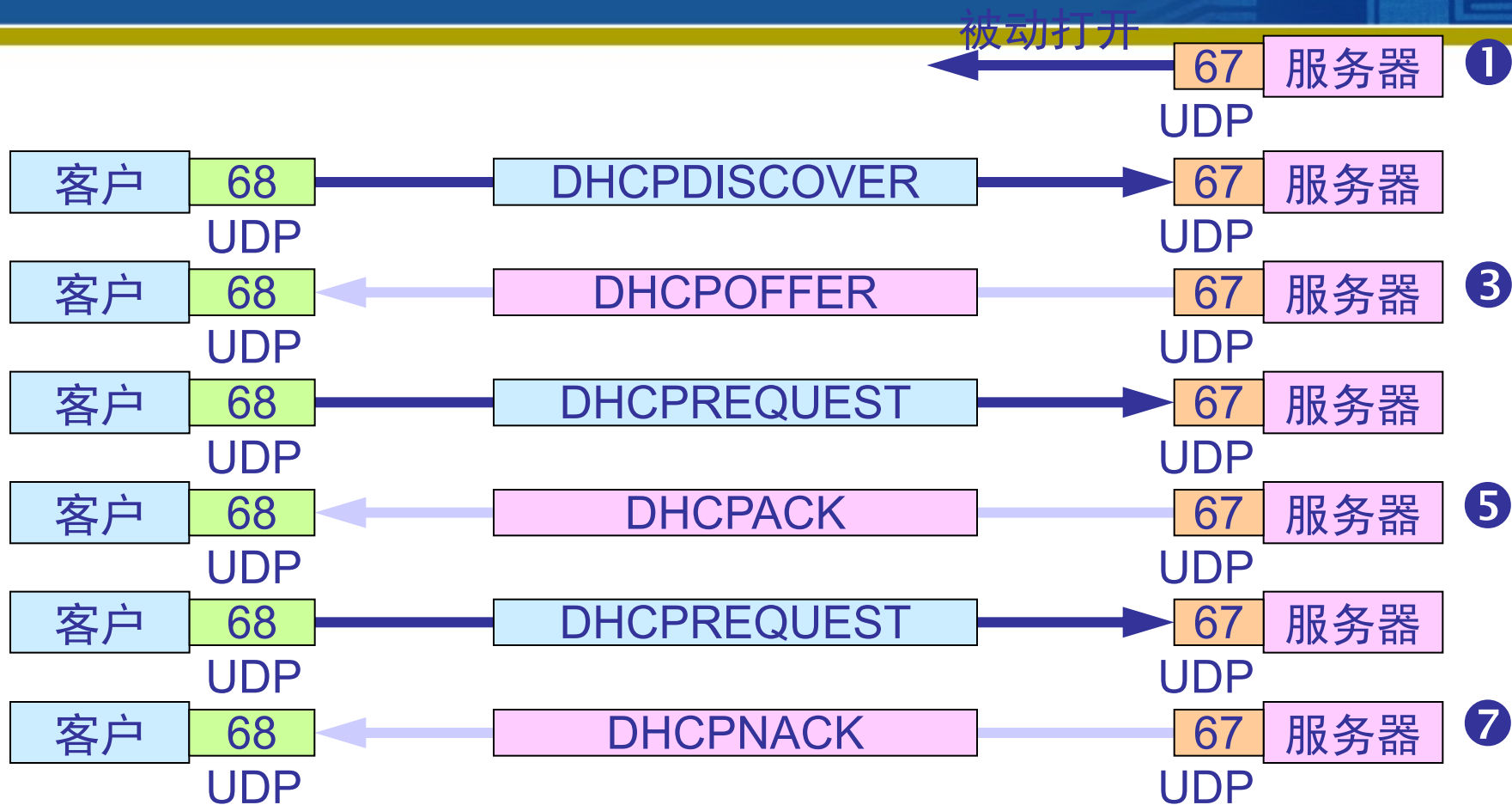
④: DHCP 客户从几个 DHCP 服务器中选择其中的一个，并向所选择的 DHCP 服务器发送 DHCP 请求报文。

DHCP 协议的工作过程



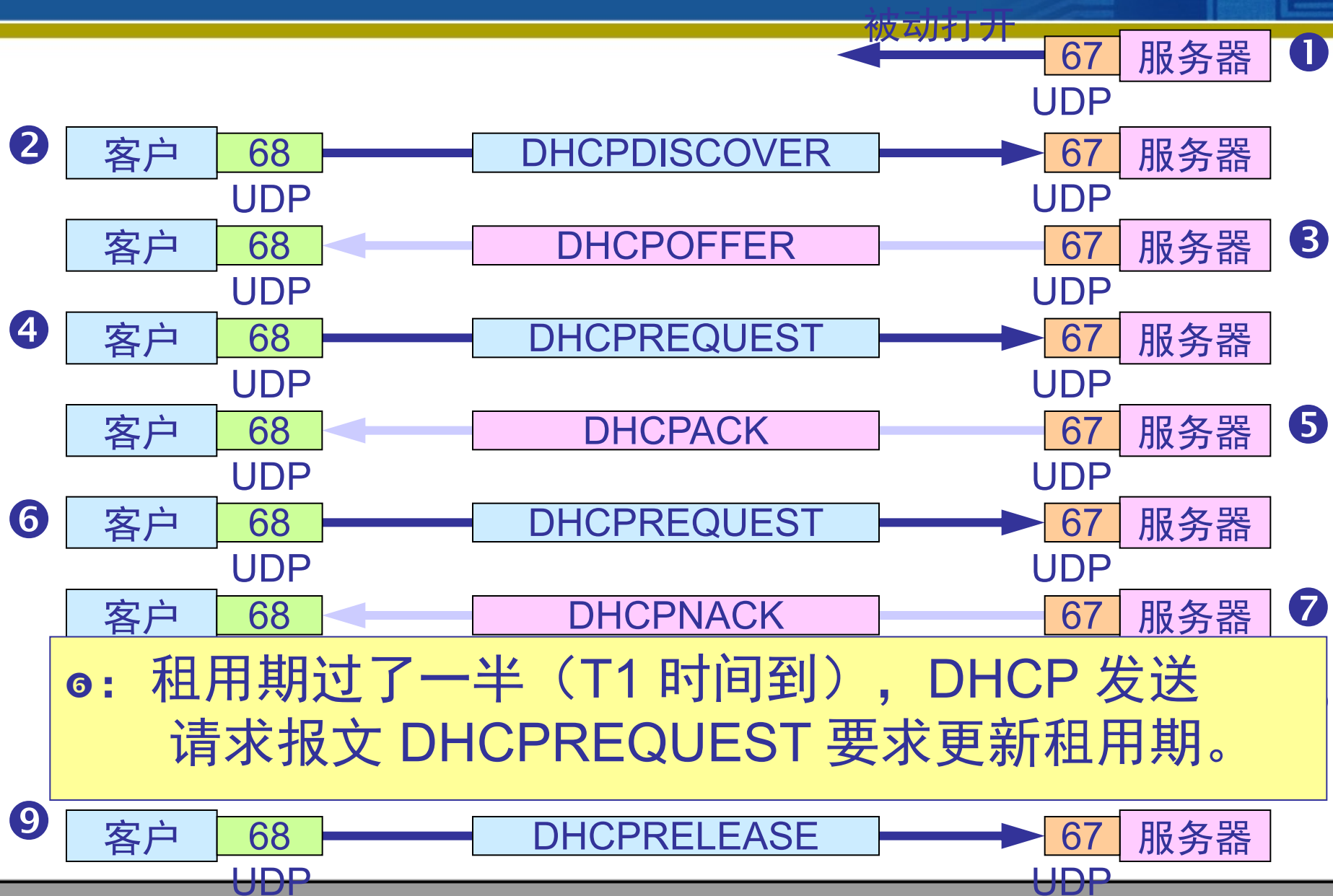
⑤：被选择的 DHCP 服务器发送确认报文 DHCPACK，进入已绑定状态，并可开始使用得到的临时 IP 地址了。

DHCP 协议的工作过程



DHCP 客户现在要根据服务器提供的租用期 T 设置两个计时器 T_1 和 T_2 ，它们的超时时间分别是 $0.5T$ 和 $0.875T$ 。当超时时间到就要请求更新租用期。

DHCP 协议的工作过程



DHCP 协议的工作过程

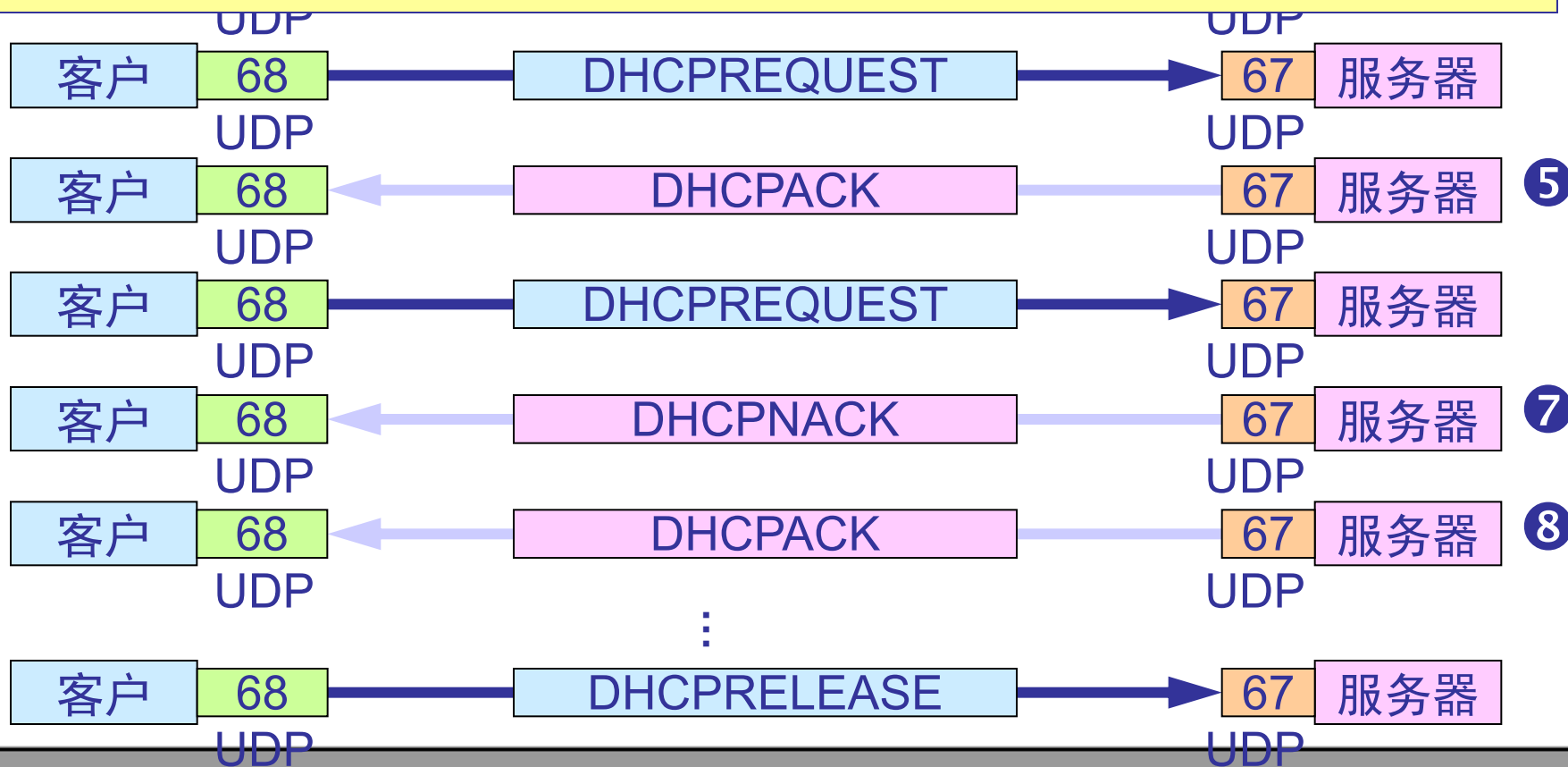
被动打开

67

服务器

1

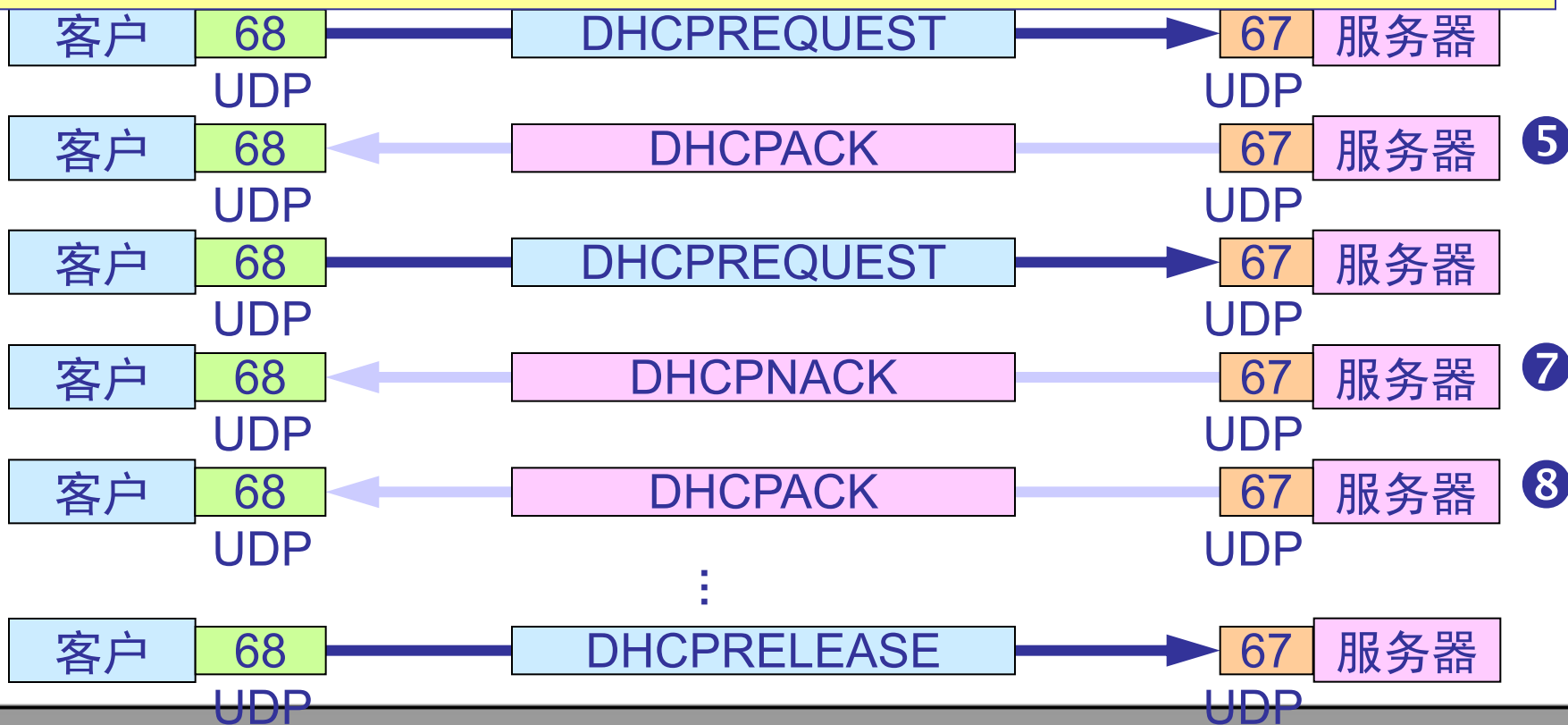
- ⑦: DHCP 服务器若同意, 则发回确认报文 DHCPACK。DHCP 客户得到了新的租用期, 重新设置计时器。



DHCP 协议的工作过程

续前图

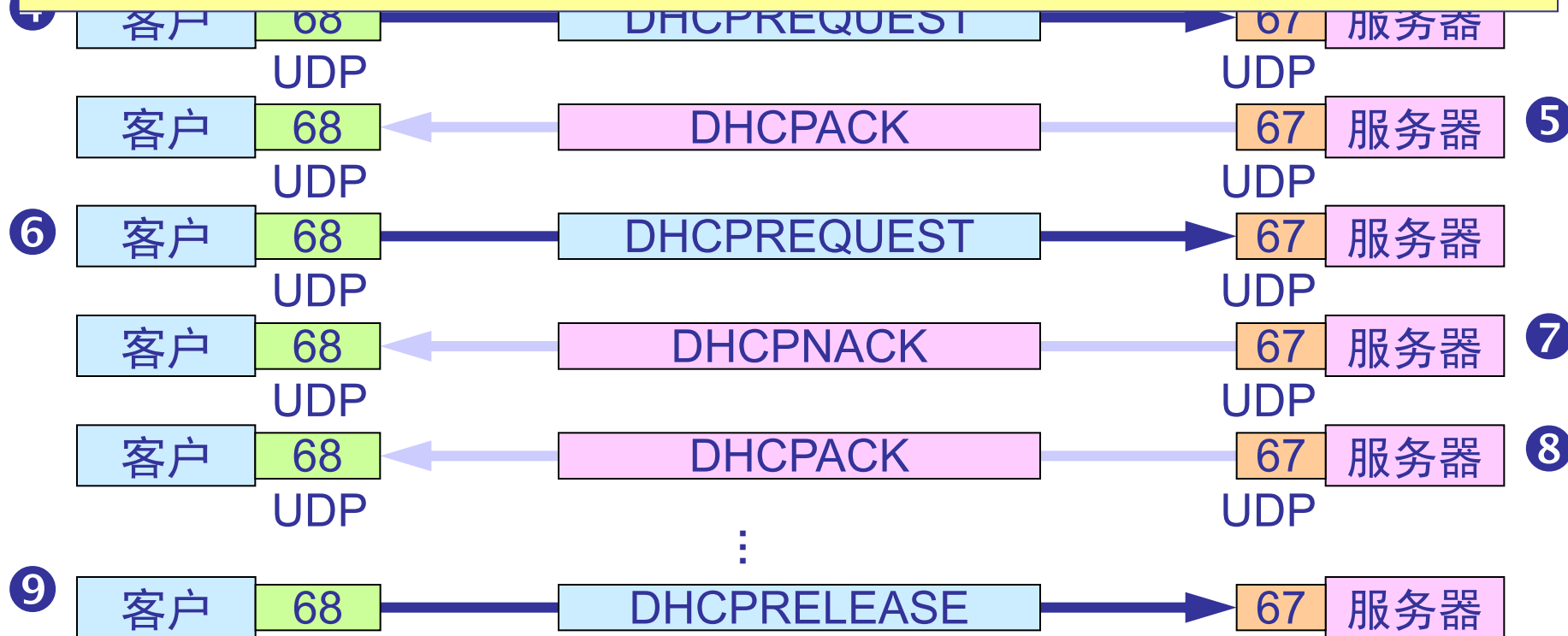
⑧：DHCP 服务器若不同意，则发回否认报文 DHCPNACK。这时 DHCP 客户必须立即停止使用原来的 IP 地址，而必须重新申请 IP 地址（回到步骤②）。



DHCP 协议的工作过程

被动打开

若 DHCP 服务器不响应步骤 ⑥ 的请求报文 DHCPREQUEST，则在租用期过了 87.5% 时，DHCP 客户必须重新发送请求报文 DHCPREQUEST（重复步骤 ⑥），然后又继续后面的步骤。



DHCP 协议的工作过程

被动打开

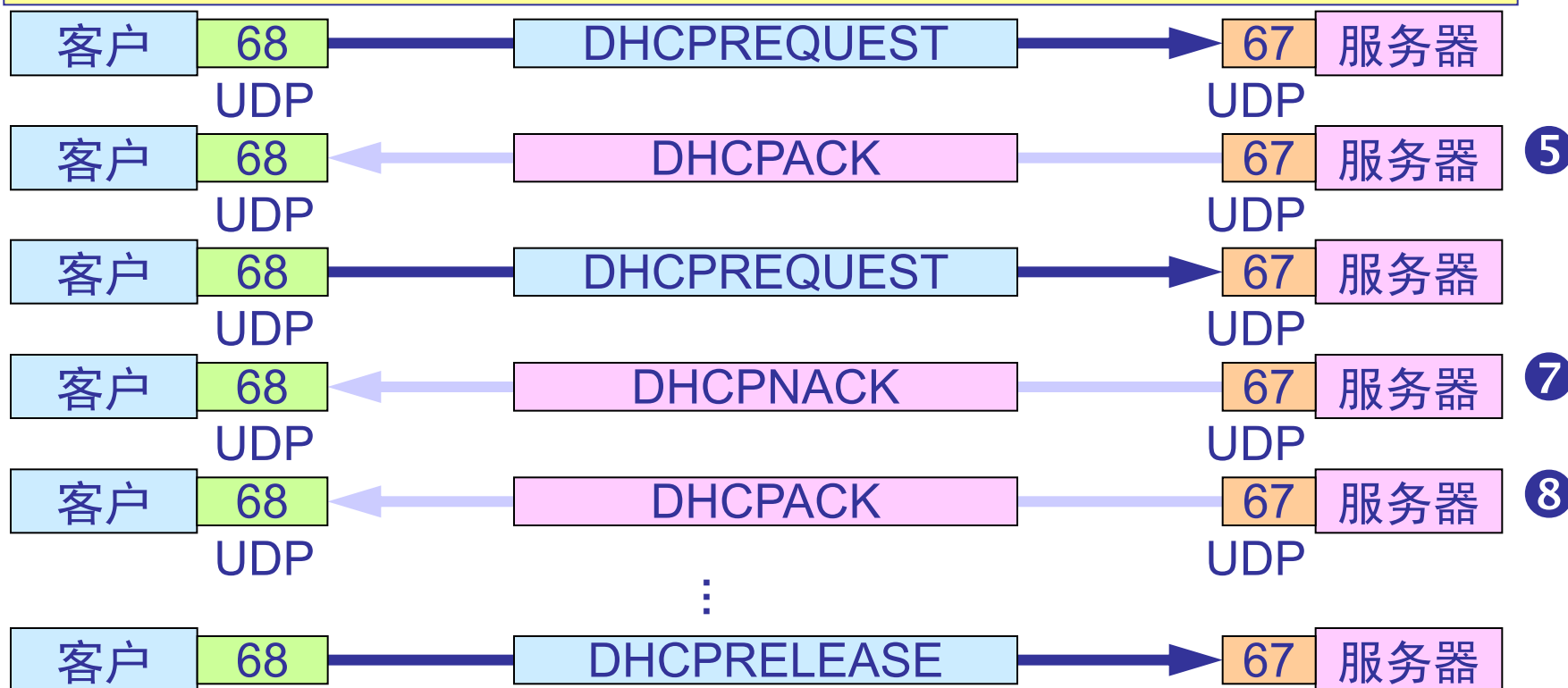
67

服务器

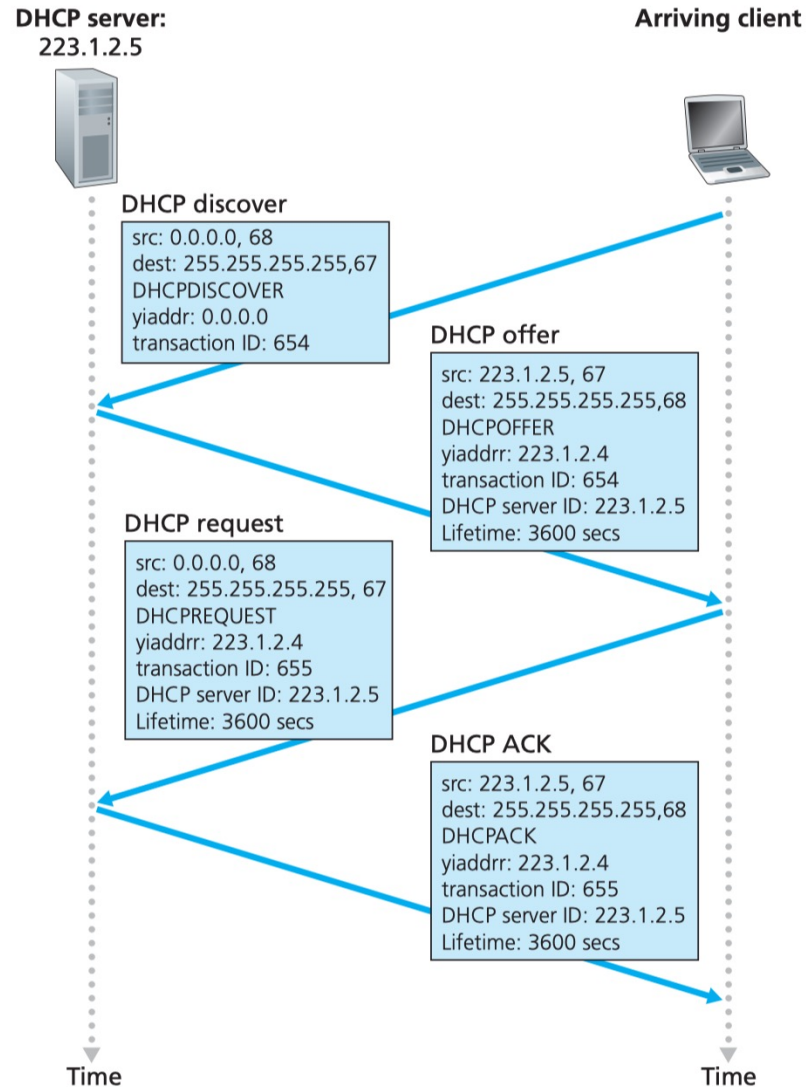
1

⑨: DHCP 客户可随时提前终止服务器所提供的租用期, 这时只需向 DHCP 服务器发送释放报文 DHCPRELEASE 即可。

3



DHCP 协议的工作过程



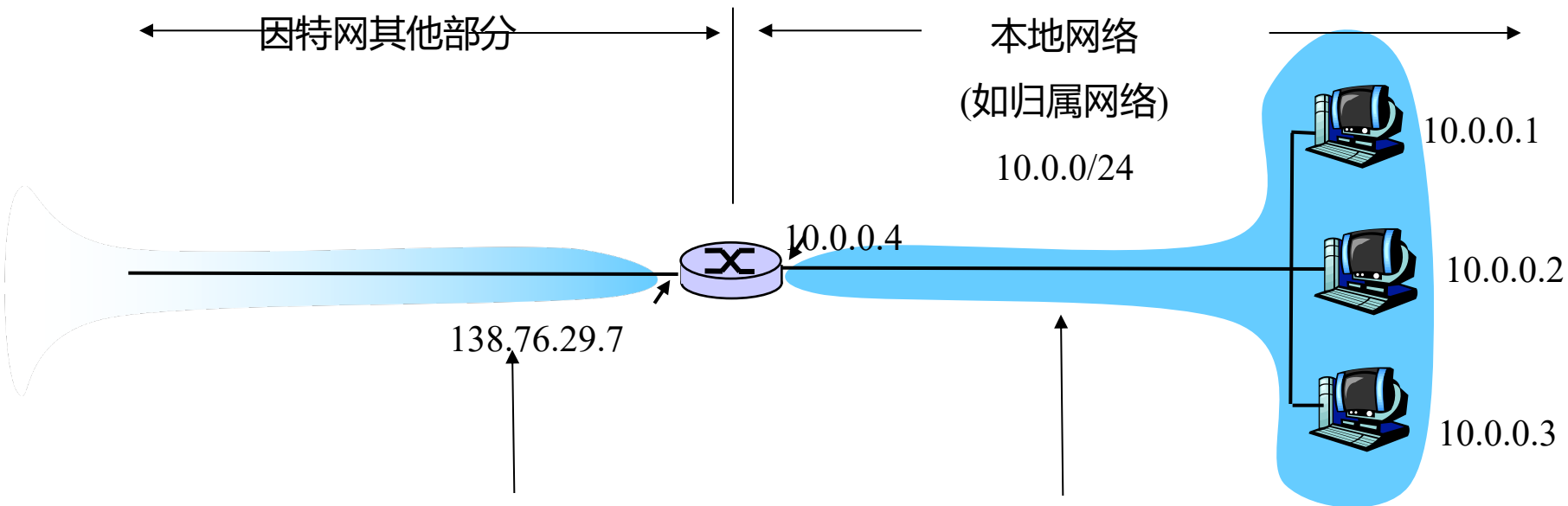
4.3.2 网络地址转换

问题：IPv4的IP地址共有多少个？够用吗？

解决办法：

- 下一代的IPv6，增加IP地址位数（彻底解决）
- NAT技术（地址代理技术），提供内部私有地址与共有地址的转换，支持内网与公网的通信

4.3.2 网络地址转换



所有数据报本地离开本地网络具有相同的单一源NAT IP地址: 138.76.29.7, 不同的源端口号

具有该网源或目的的数据报都有10.0.0/24的地址(照常)

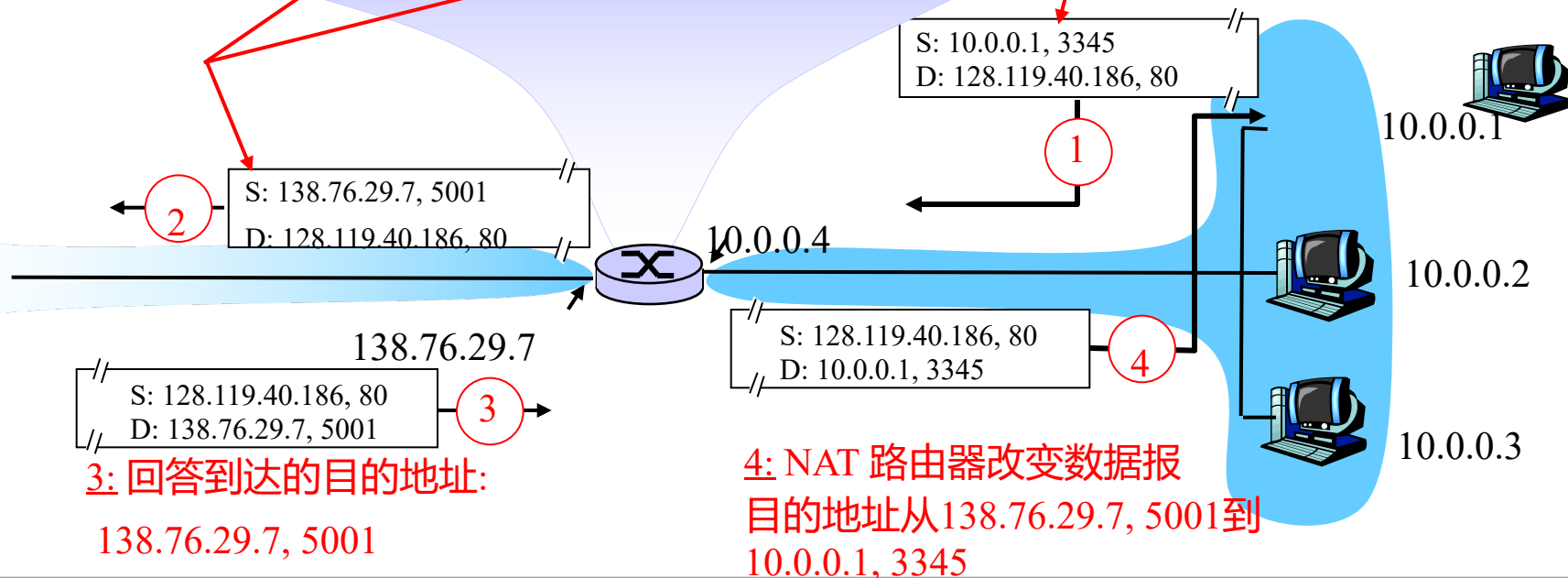
4.3.2 网络地址转换

外网IP地址与端口号 ↔ 内网IP地址与端口号

| NAT 转换表 | |
|-------------------|----------------|
| WAN 侧地址 | LAN 侧地址 |
| 138.76.29.7, 5001 | 10.0.0.1, 3345 |
| | |

2: NAT路由器改变数据报源地址从10.0.0.1, 3345 到138.76.29.7, 5001, 更新表

1: 主机10.0.0.1发送数据报到128.119.40, 80



4.3.2 网络地址转换

➤ 16-bit 端口号字段:

- ◆ 用一个LAN侧地址支持60,000 并行连接!

➤ NAT 引起争议:

- ◆ 路由器的处理上升为第三层

- ◆ 违反了端到端原则

- ◆ 应用设计者必须要考虑 NAT可能性, 如 P2P应用程序

- ◆ 地址短缺应当由IPv6来解决

总结

- 网络层提供的服务；
- 路由器工作原理：
 - ◆ 转发机制；
 - ◆ 调度机制
- IP编址：
 - ◆ IP数据报的格式和分片
 - ◆ IP编址的演化；
 - ◆ 子网划分，子网掩码；
 - ◆ IP地址获取，NAT机制

作业

P4, P5, P7, P8