

第19章

网络层：逻辑寻址

第四部分 网络层

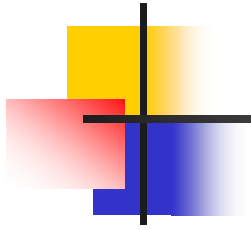
- 网络层负责在**计算机间**传输**分组**，可能经过多个网络（链路）；而数据链路层负责在**同一网络（链路）内**的两个系统之间传输**分组（或帧）**；
- 网络层保证每一分组从源到它的最后目的地；
- 网络层增加一个头部，它包含分组发送方与接收方的逻辑地址（**指主机地址**）；如果分组经过因特网传输，还需要寻址系统来识别源与目的地址；
- 当若干个独立的网络（链路）连接在一起组成一个互联网时，路由器或交换机（**指三层交换机**）对分组进行路由选择到达它的最终目的地，网络层的一个主要功能是提供路由机制。

19-1 IPv4地址

pIPv4地址是一个32位地址（4个字节），它唯一地、通用地定义了一个连接在因特网上的设备，例如主机或是路由器（**的一个网络接口**）；

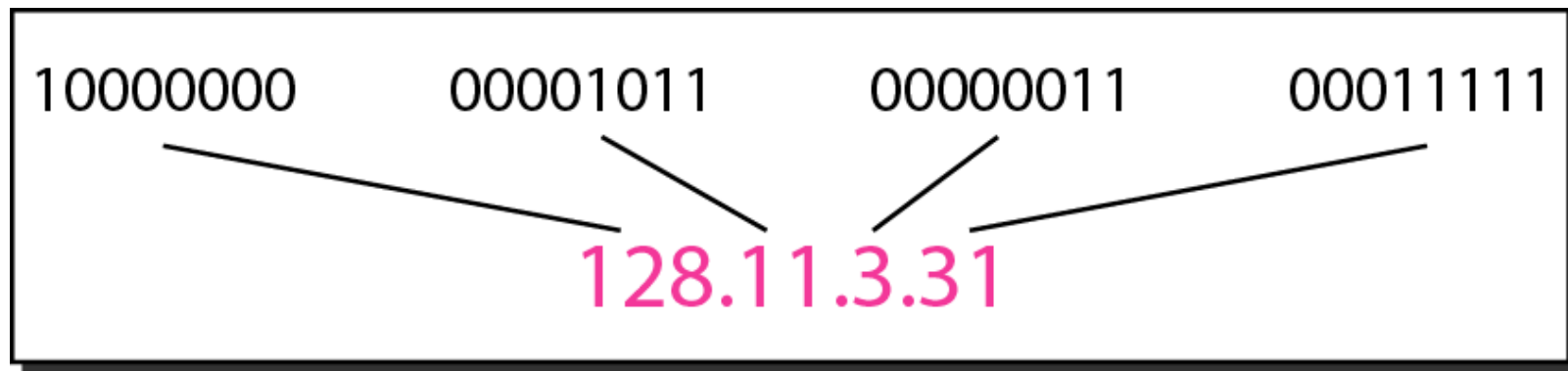
pIPv4地址是唯一的：每一个地址定义了因特网上的一个且仅有一个设备，在因特网上的两个设备永远不会具有同样的地址（**不包含内部地址情况**）；此外，如果因特网上一个设备连接 m 个物理网络，则它需要拥有 m 个地址（例如路由器）；

pIPv4地址是通用的：这个地址系统必须被任何一个愿意连接到因特网上的主机所接受（**是一个逻辑地址，与具体物理网络类型无关**）。



IPv4的地址空间是 2^{32} 或4,294,967,296。

图19.1 IPv4地址的二进制标记法和点分十进制标记法





例19.1

将下列IPv4地址由二进制标记法转为点分十进制标记法。

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

解：

我们可将8位组用一个等价的十进制数代替并添加点来分隔。

a. 129.11.11.239

b. 193.131.27.255



例19.2

将下列IPv4地址由点分十进制标记法转为二进制标记法。

a. 111.56.45.78

b. 221.34.7.82

解：

将十进制数用等价的二进制数代替。

a. 01101111 00111000 00101101 01001110

b. 11011101 00100010 00000111 01010010



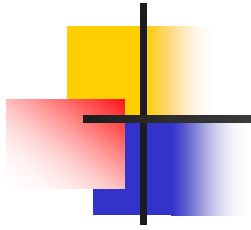
例19.3

寻找下列IPv4地址的标记法中的差错。

- a. 111.56.045.78
- b. 221.34.7.8.20
- c. 75.45.301.14
- d. 11100010.23.14.67

解：

- a. 首位必须非0（045）。
- b. IPv4地址不能多于四个数（字节）。
- c. 每个数必须小于等于 255（301在范围外）。
- d. 二进制标记法与点分十进制标记法不能混用。



**在分类寻址中，IPv4地址空间被划为5类：
A、B、C、D和E。**

图19.2 以二进制和点分十进制标记法求类（开头几位值确定地址所属的类）

| | First byte | Second byte | Third byte | Fourth byte |
|---------|------------|-------------|------------|-------------|
| Class A | 0 | | | |
| Class B | 10 | | | |
| Class C | 110 | | | |
| Class D | 1110 | | | |
| Class E | 1111 | | | |

a. Binary notation

| | First byte | Second byte | Third byte | Fourth byte |
|---------|------------|-------------|------------|-------------|
| Class A | 0-127 | | | |
| Class B | 128-191 | | | |
| Class C | 192-223 | | | |
| Class D | 224-239 | | | |
| Class E | 240-255 | | | |

b. Dotted-decimal notation



图19.4

求每个地址的类。

- a. 00000001 00001011 00001011 11101111
- b. 11000001 10000011 00011011 11111111
- c. 14.23.120.8
- d. 252.5.15.111

解：

- a. 第一位是 0，这是个A类地址。
- b. 前2位是11，第3位是0，这是个C类地址。
- c. 第一个字节是14（在0~127之间），这个地址是A类；
- d. 第一个字节是252（在240~255之间），这个地址是E类。

地址类与地址块

- 分类寻址的问题：每个地址类被分成一个固定数量的地址块，并且每个地址块都有固定的长度（指所含地址的数量）；
- 当一个组织机构申请一个地址块时，被允许A类、B类或C类中的一个块

表19.1 IPv4分类寻址块的个数与每块的长度

| <i>Class</i> | <i>Number of Blocks</i> | <i>Block Size</i> | <i>Application</i> |
|--------------|-------------------------|-------------------|--------------------|
| A | 128 | 16,777,216 | Unicast |
| B | 16,384 | 65,536 | Unicast |
| C | 2,097,152 | 256 | Unicast |
| D | 1 | 268,435,456 | Multicast |
| E | 1 | 268,435,456 | Reserved |

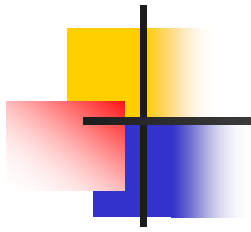
分类寻址带来的地址浪费

pA类地址为具有大量主机或路由器的大型组织机构设计，B类为可能具有数万台主机或路由器的中型组织机构设计，C类为具有少量主机或路由器的小型组织机构设计；

p缺点：A类地址块中的地址个数可能比几乎所有组织机构的需要都要大，许多地址被浪费；B类地址块中的地址个数比大多数中型组织机构的需要量也太大，许多地址也被浪费；而C类地址块中的地址个数比大多数组织机构的需要又太少；

pD类地址为多播设计，每一个地址定义连接因特网上的一组主机，因特网管理机构错误地预测需要268435356组，浪费了太多的地址；

pE类地址保留，仅有少数几个被使用，也引起了地址的浪费。



在分类寻址中，大部分可用的地址都被浪费了。

2011年2月4日， ICANN（互联网名称与数字地址分配机构）宣布，最后5个IPv4地址“大礼包”分配了出去。

网络号与主机号

• A类、B类和C类中IP地址被分成网络号（缺省情况下与地址块的数量相对应）与主机号两部分；

• 依照地址类的不同，它们具有不同的长度；

• 缺省情况下，A类中，一个字节定义网络号而三个字节定义主机号；在B类中，二个字节定义网络号，二个字节定义主机号；在C类中，三个字节定义网络号而一个字节定义主机号；

• 注意：D类和E类地址并没有分成网络号与主机号。

| | First byte | Second byte | Third byte | Fourth byte |
|---------|------------|-------------|------------|-------------|
| Class A | 0 | | | |
| Class B | 10 | | | |
| Class C | 110 | | | |
| Class D | 1110 | | | |
| Class E | 1111 | | | |

a. Binary notation

| | First byte | Second byte | Third byte | Fourth byte |
|---------|------------|-------------|------------|-------------|
| Class A | 0-127 | | | |
| Class B | 128-191 | | | |
| Class C | 192-223 | | | |
| Class D | 224-239 | | | |
| Class E | 240-255 | | | |

b. Dotted-decimal notation

掩码

- 可用掩码（或默认掩码）来区分网络号和主机号，32位，连续1后跟连续0的串组成；
- 连续1部分对应了网络号部分，连续0部分对应了主机号部分；
- 两种表示方式：点分十进制格式或 /n（斜杠标记，也称为无类域间路由选择**CIDR标记**），n代表了连续1的个数；
- 注意：D类和E类没有掩码的概念

| <i>Class</i> | <i>Binary</i> | <i>Dotted-Decimal</i> | <i>CIDR</i> |
|--------------|-------------------------------------|-----------------------|-------------|
| A | 11111111 00000000 00000000 00000000 | 255.0.0.0 | /8 |
| B | 11111111 11111111 00000000 00000000 | 255.255.0.0 | /16 |
| C | 11111111 11111111 11111111 00000000 | 255.255.255.0 | /24 |

子网化

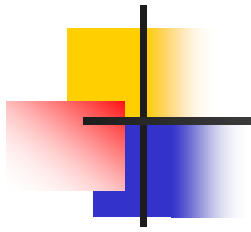
p 如果一个组织机构指派A类或B类中一大块地址，它可将这些地址划分为几个类组，并赋予每一组为较小的网络（称为子网，subnet）；

p 特殊情况下，一个子网与邻近子网间有共享地址的部分；

p 子网化是在掩码中增加1的个数实现（称为可变长子网掩码VLSM），将地址由二级层次变为三级层次结构（增加了子网号）

超网化

- ⌘ A类地址和B类地址已几乎用完，但仍有大量要求中等规模的地址块，而C类地址块只能容纳256个地址，无法满足多数组织机构的需要；
- ⌘ 一种解决问题的方法是超网化（supernetting）；
- ⌘ 在超网化中，一个组织机构能将几个C类块地址构成更大范围的地址空间；换言之，将几个网络联合起来构成一个超网（supernet）；
- ⌘ 一个组织机构要申请一组C类地址块而不是仅仅一个；
- ⌘ 例如，一个需要1000个地址的组织机构可以申请4个C类地址块（需要连续）；该组织机构就可以使用这些地址创建一个超网；
- ⌘ 超网化是在掩码中减少1的个数，例如，一个组织机构需要4个C类地址块，掩码从/24变成/22。



减轻地址耗尽问题的一个解决办法是无类寻址思想（子网化或超网化），这时的分类寻址被无类寻址所取代。

子网：将一个大的网络划分成几个较小的网络，且每一个网络都有自己的子网地址；

超网：将一个组织所属的几个网络合并成一个地址范围更大的逻辑网络。

无类寻址

p 为了克服地址耗尽并使更多组织机构接入因特网，设计并开发了无类寻址（**classless addressing**），在这个方案中，没有类但仍提供地址块；

p 在无类寻址中，当一个小的或大的实体需要连接因特网时，给它分配一个适合的地址块，块的大小（地址的个数）按实体大小规模与性质来决定；

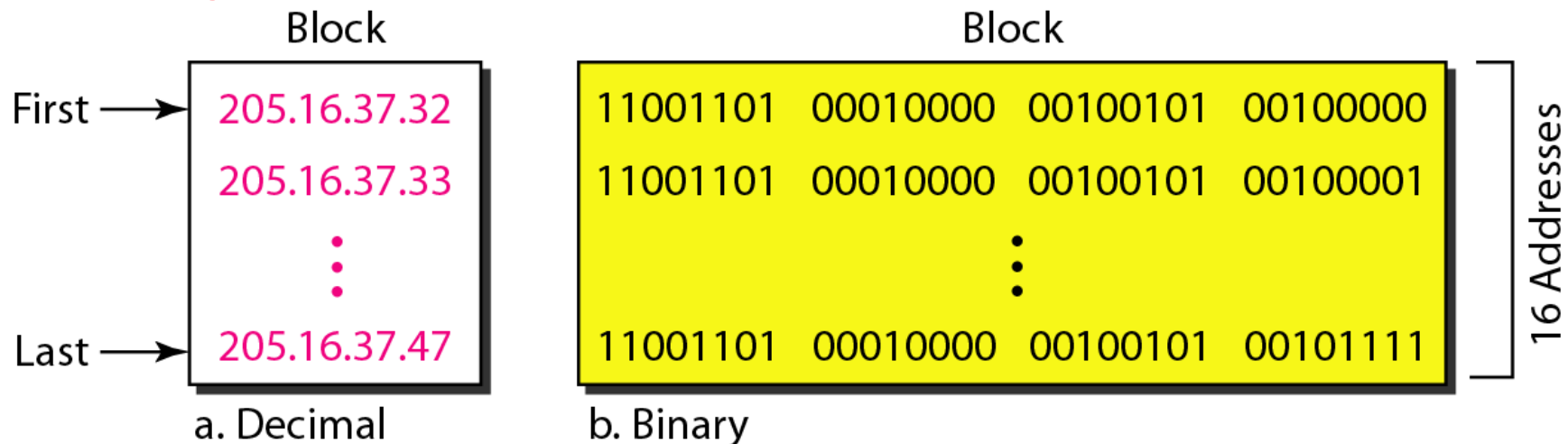
p 为了简化对地址处理，因特网管理机构对无类地址块强加了3个限制条件：

- Ø1. 块中的地址必须是一个接着一个连续的；
- Ø2. 一个块中地址的个数是2的整数次次幂（1，2，4，8，...）；
- Ø3. 块的起始地址必须能被块中地址的个数整除。

例19.5

图19.3表示了给一个小型商业公司分配一个由16个地址组成的地址块，用二进制标记法和点分十进制标记法表示。

可以看到对这个块实施了限制。地址都是连续的，地址的个数是2的次幂（ $16=2^4$ ），而起始地址可用16整除。（Q: 掩码？）



块地址

- 定义块地址的一个方法是选择块中任一地址和掩码；
- 在IPv4寻址中，块地址可用x.y.z.t/n来定义，其中x, y, z, t定义了一个地址，而/n定义掩码；
- 地址和/n可以完全定义整个地址块（起始地址、最后地址和地址个数）；
- 起始地址的二进制标记法可通过设置最右边32-n位都是0求得；
- 用二进制标记法的最后地址可设置最右边的32-n位都是1求得；
- 块中地址个数可简单用公式 2^{32-n} 求得。



例19.6

分配给某一小型组织机构一个地址块，我们已知块中一个地址是205.16.37.39/28，求该块的起始地址？

解：

已知地址的二进制表示是

11001101 00010000 00100101 00100111

如果置最右边32-28位都是0，则得到块的起始地址

11001101 00010000 00100101 00100000

或

205.16.37.32



例19.7

求例 19.6中的最后地址。

解：

已知地址的二进制表示是

11001101 00010000 00100101 00100111

如果置最右边32-28位都是1，则得到块的最后地址

11001101 00010000 00100101 00101111

或

205.16.37.47



例19.8

求例19.6中的地址个数。

解：

n的值是28，这就是说地址个数是 2^{32-28} 或16。



例19.9

求起始地址、最后地址和地址个数的另一种方法是用32位二进制（或8位十六进制）数的掩码表示。当我们编制一个程序去求这些信息时，这特别有用。在例19.5中，/28可表示为

11111111 11111111 11111111 11110000

（28个1和4个0）

求

- a. 起始地址
- b. 最后地址
- c. 地址个数。



例19.9 (cont.)

解:

a. 起始地址可用给定地址与掩码进行与运算求得，这里与运算是对应位逐位与操作，如果两位都是1，则与操作结果为1，否则为0。

| | | | | |
|----------------|----------|----------|----------|----------|
| Address: | 11001101 | 00010000 | 00100101 | 00100111 |
| Mask: | 11111111 | 11111111 | 11111111 | 11110000 |
| First address: | 11001101 | 00010000 | 00100101 | 00100000 |



例19.9 (cont.2)

b. 最后地址可用给定地址与掩码的反码进行或运算求得，这里或运算是对应位逐位或操作，如果两位都是0，则或操作结果为0，否则为1。

| | | | | |
|------------------|----------|----------|----------|----------|
| Address: | 11001101 | 00010000 | 00100101 | 00100111 |
| Mask complement: | 00000000 | 00000000 | 00000000 | 00001111 |
| Last address: | 11001101 | 00010000 | 00100101 | 00101111 |



例19.9 (cont.3)

c. 地址的个数可由掩码求反，它的十进制数加1求得。

Mask complement: 00000000 00000000 00000000 00001111

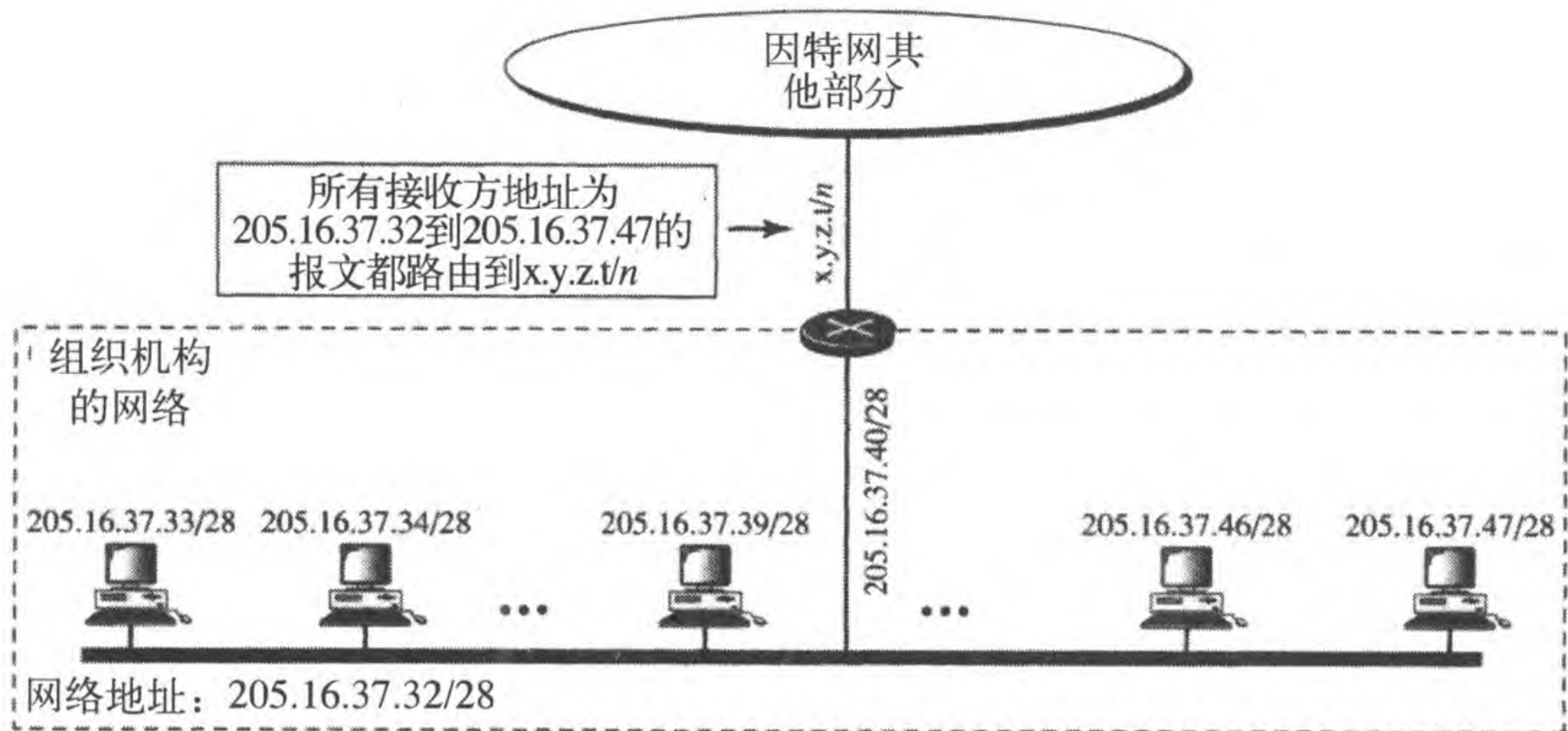
Number of addresses: $15 + 1 = 16$

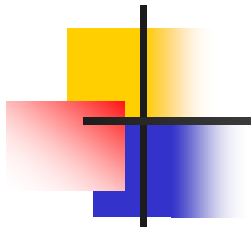
网络地址

- p IP寻址中一个很重要的概念是网络地址；
- p 当一个组织机构已有一个地址块时，它可自由地给连接到因特网的每一个设备分配地址，然而，该块的起始地址通常被用作一个特殊地址，它称为网络地址；
- p 网络地址定义该组织机构的网络，也就是对于全球其他部分来说它是该组织机构本身的网络；
- p 路由器本质上是根据网络地址来寻址的。

图19.4 有关地址块205.16.37.32/28的网络配置

- 路由器有两个地址，一个是属于获准的块，另一个属于路由器另一端所在的网络，记为x.y.z.t/n；
- 发给地址块中IP的所有指定报文都发送到x.y.z.t/n。





块中的起始地址通常不分配给任何设备；它用做向世界上其他部分表示该组织的网络地址。

注意：块中的最后地址通常也不分配给任何设备，它是该块的广播地址。

IP地址的层次结构

p 像其他地址一样，IP地址也有层次结构。

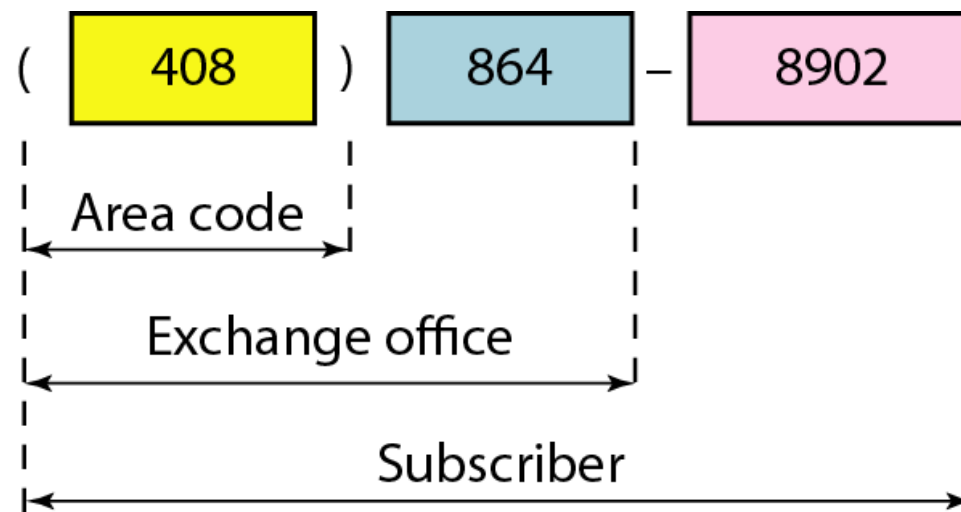
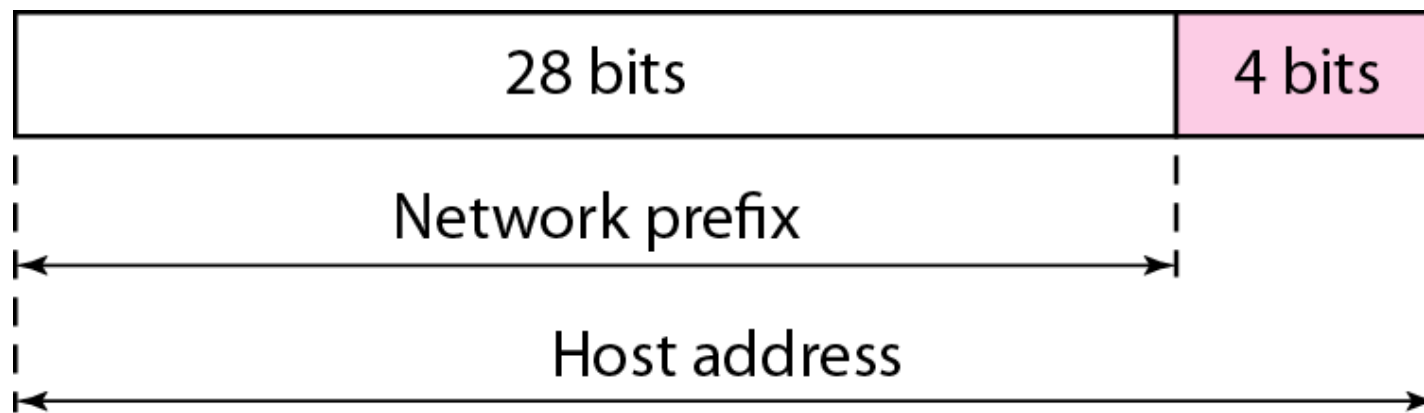
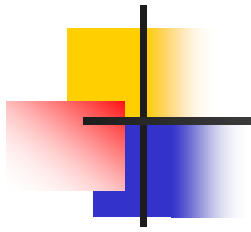


图19.5 北美电话网络层次结构

图19.6 IPv4地址中二级层次结构（没有划分子网时）





块中的每一个地址可看作二级层次结构：最左的 n 位（前缀）定义网络，最右的 $32-n$ 位（后缀）定义主机。

三级层次结构：子网化

- p 一个组织机构被指派一大块地址，它想要分成几个网络（称为子网），划分不同子网的地址；
- p 世界的其余部分仍将该组织机构看做是一个实体，但其内部有几个子网；
- p 将所有报文发送到将组织机构连接到因特网其余部分的路由器地址上，路由器将报文发送到合适的子网；
- p 然而，该组织机构需要创建多个较小地址子块，每个子块指向特殊的子网；
- p 组织机构有它自己的掩码，每个子网也必须有它自己的掩码。

图19.7 一个子网化网络中的配置和地址

假定已给一个组织机构分配的地址块为 **17.12.14.0/26**，它有64个地址；

该组织机构有3个部门，需要该块地址划分为32、16和16个地址的子块；

如何求每个子网的掩码？

如何求每个子网的网络地址？

还有其他划分的方法吗？

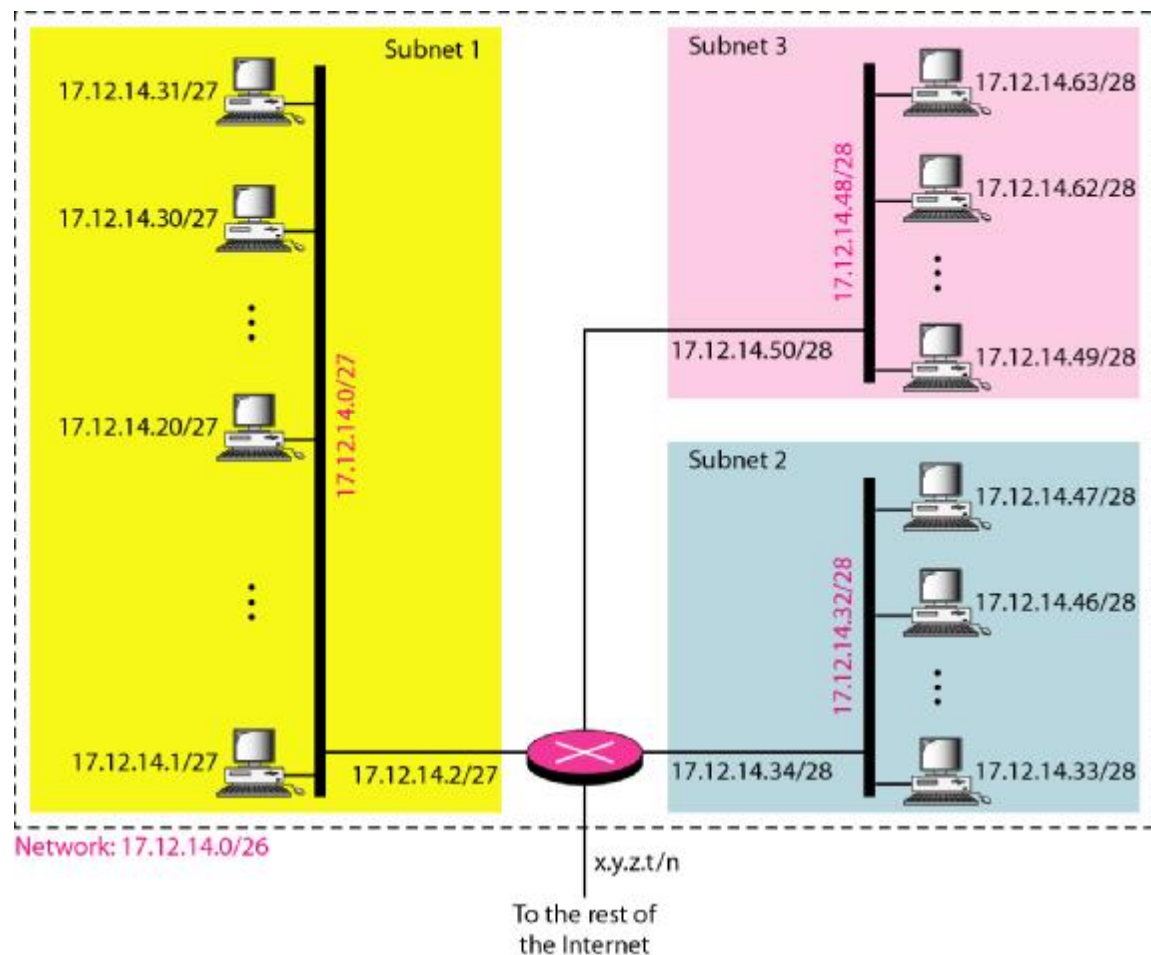
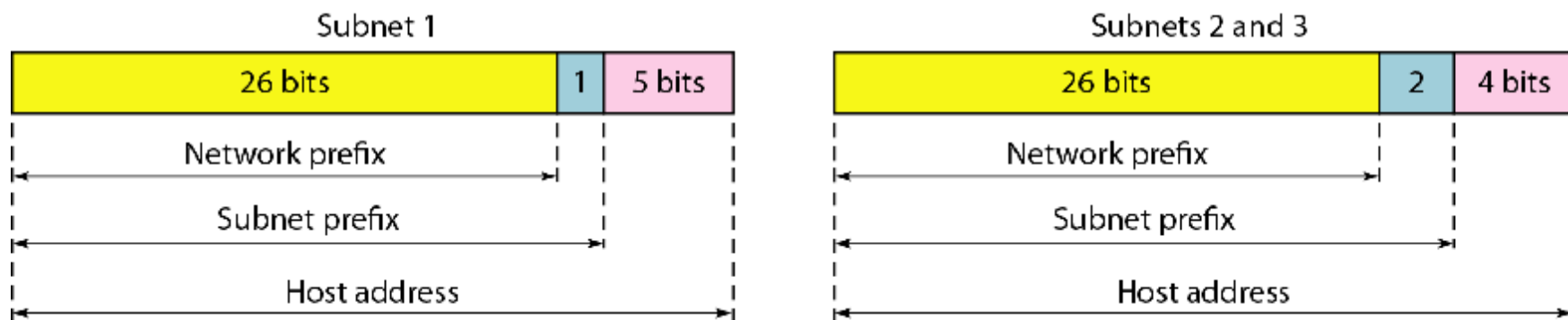


图19.8 IPv4中三级层次结构（针对图19.7）



多级层次结构

- p**分类寻址结构没有限制层次结构的级数，一个组织机构可以将已给予的地址块划分为子块，每个子块可依次再划分为更小的子块等；
- p**在ISP中可见到这样的例子：国家级的ISP可将已给予的大块地址划分为较小的子块，将它们分配给区域级的ISP；
- p**而每个区域级的ISP将从国家级的ISP接收到地址块再划分为更小子块，并将它们分配给本地级的ISP；
- p**每个本地级的ISP将从区域级的ISP接收到地址再划分为更小的子块，并将每一个子块分配给不同的组织机构；
- p**最后，组织机构划分接收到的块，并由它组成多个子网。

地址分配机构

- p 最终负责地址分配的是称为因特网名称和编号分配组织（ICANN）的全球权威机构；
- p 但是，ICANN通常不向个人组织机构分配地址；它分配一大块地址给ISP，每个ISP依次将已给予它的块划分为较小子块，并将各个子块分配给它的用户



例19.10

给一个ISP分配了起始地址为190.100.0.0/16（65536个地址）的地址块。ISP需要按如下给3组客户分发这些地址：

- a. 第一组有64个客户，每个需要256个地址；
- b. 第二组有128个客户，每个需要128个地址；
- c. 第三组有128个客户，每个需要64个地址。

设计这些子块，并求出分配后还有多少可用的地址？



例19.10 (cont.)

解:

图19.9 表示了该情况。

组1

对于这组，每个客户需要256个地址，就是说需要用8位 ($\log_2 256$) 定义每一台主机，那么前缀的长度是 $32 - 8 = 24$ 。地址是：

| | | |
|----------------------------------|-----------------|-------------------|
| 1st Customer: | 190.100.0.0/24 | 190.100.0.255/24 |
| 2nd Customer: | 190.100.1.0/24 | 190.100.1.255/24 |
| ... | | |
| 64th Customer: | 190.100.63.0/24 | 190.100.63.255/24 |
| Total = $64 \times 256 = 16,384$ | | |



例19.10 (cont.2)

组2

对于这组，每个客户需要128个地址，就是说需要用7位 ($\log_2 128$) 定义每一台主机，那么前缀的长度是 $32-7=25$ 。地址是：

| | | |
|-----------------------------------|--------------------|--------------------|
| 1st Customer: | 190.100.64.0/25 | 190.100.64.127/25 |
| 2nd Customer: | 190.100.64.128/25 | 190.100.64.255/25 |
| ... | | |
| 128th Customer: | 190.100.127.128/25 | 190.100.127.255/25 |
| Total = $128 \times 128 = 16,384$ | | |



例19.10 (cont.3)

组3

对于这组，每个客户需要64个地址，就是说需要用6位（ $\log_2 64$ ）定义每一台主机，那么前缀的长度是 $32-6=26$ 。地址是：

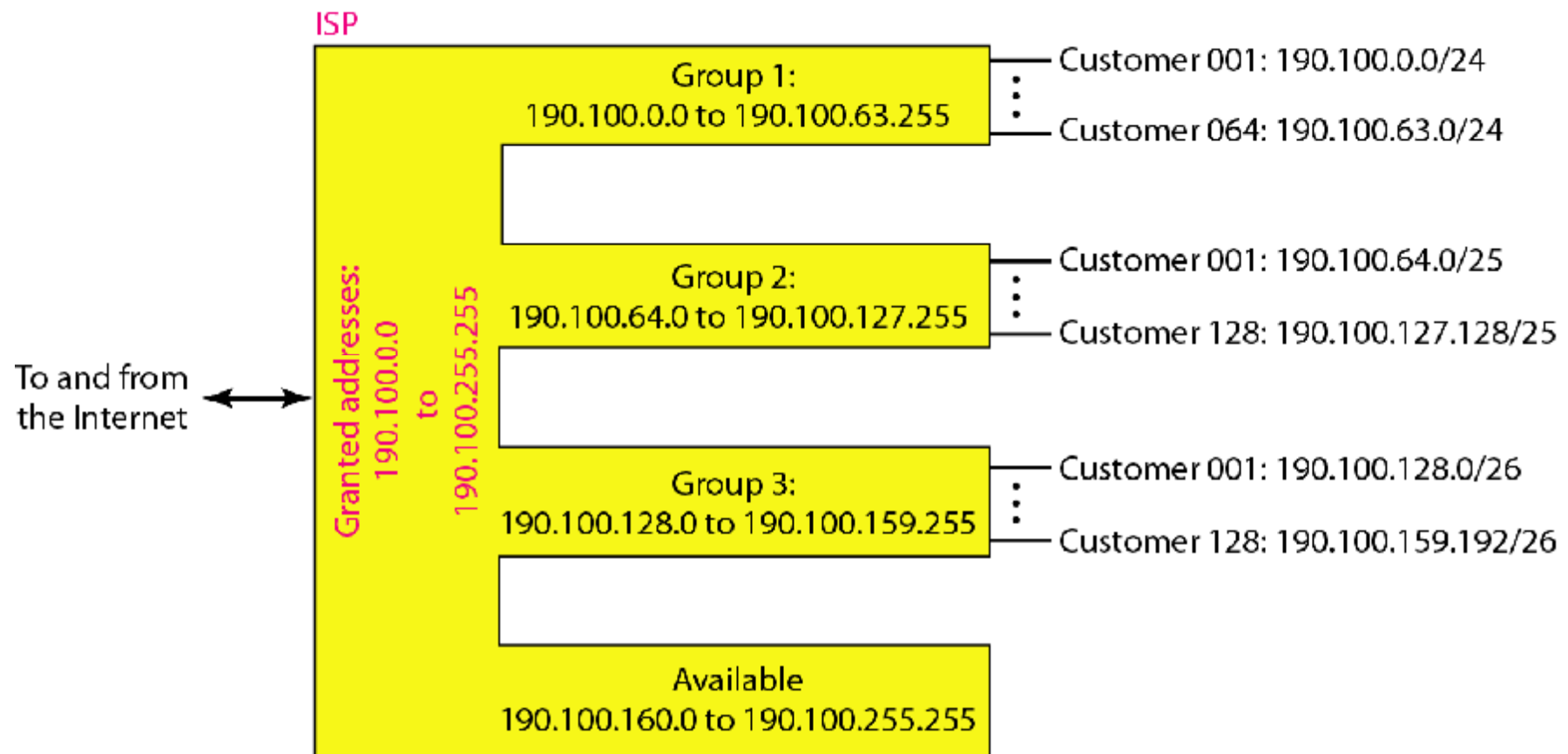
| | | |
|--------------------------------|--------------------|--------------------|
| 1st Customer: | 190.100.128.0/26 | 190.100.128.63/26 |
| 2nd Customer: | 190.100.128.64/26 | 190.100.128.127/26 |
| ... | | |
| 128th Customer: | 190.100.159.192/26 | 190.100.159.255/26 |
| Total = $128 \times 64 = 8192$ | | |

分配给该 ISP的地址个数：65536

由ISP分配的地址个数：40960

可用地址个数：24576

图19.9 由ISP分配和分发地址的例子



网络地址转换（NAT）

pNAT: network address translation, 是解决地址短缺的一个快速方案;

pNAT能使用户在内部拥有大量的地址, 而在外部只有少量的地址;

p而且, 内部通信能使用内部的地址, 而外部通信能使用外部地址;

p为了将家庭或商业公司内部使用的地址与因特网上使用的地址相分离, 因特网权威组织机构预留了三组地址作为专用地址;

p任何组织机构都能使用**来自这组地址**的地址而不必得到因特网权威组织机构的许可, 大家都知道这些预留地址是留给专用网络的;

p在组织机构内部它们是唯一的, 但在全世界范围内它们并不唯一, 路由器不能转发将这些地址作为目的地址的分组。

表19.3 专用网络地址

| <i>Range</i> | | | <i>Total</i> |
|--------------|----|-----------------|--------------|
| 10.0.0.0 | to | 10.255.255.255 | 2^{24} |
| 172.16.0.0 | to | 172.31.255.255 | 2^{20} |
| 192.168.0.0 | to | 192.168.255.255 | 2^{16} |

图19.10 NAT实现

p通过一台运行NAT软件的路由器，站点**有一条（或多条）**与全局因特网相连接的链路；

p专用网络使用专用地址，将一个网络与全球地址连接起来的路由器使用了一个专用地址和一个全球地址；

p专用网络相对于因特网的其他部分是透明的，并且因特网的其他部分只能看见一台地址为200.24.5.8的NAT路由器（**有什么好处？**）。

Site using private addresses

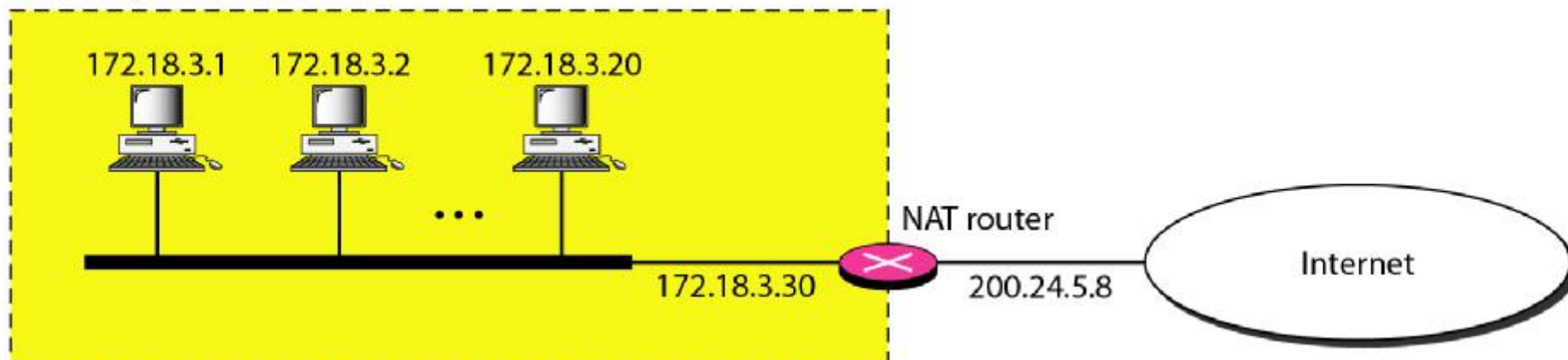


图19.11 NAT中的地址

所有外发的分组都通过NAT路由器发送出来，该路由器用全球NAT地址来替代分组中的源地址（需要修改IP头部中的源地址信息）；

所有输入的分组也要通过NAT路由器，该路由器用相应的专用地址来替代分组中的目的地址（NAT路由器的全球地址）；

地址转换需要转换表

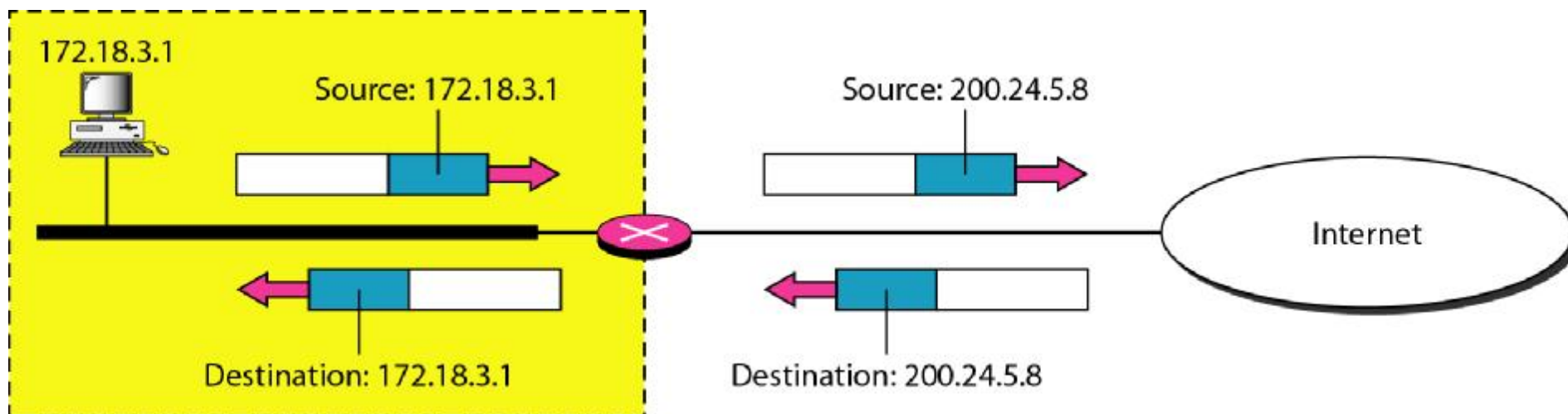
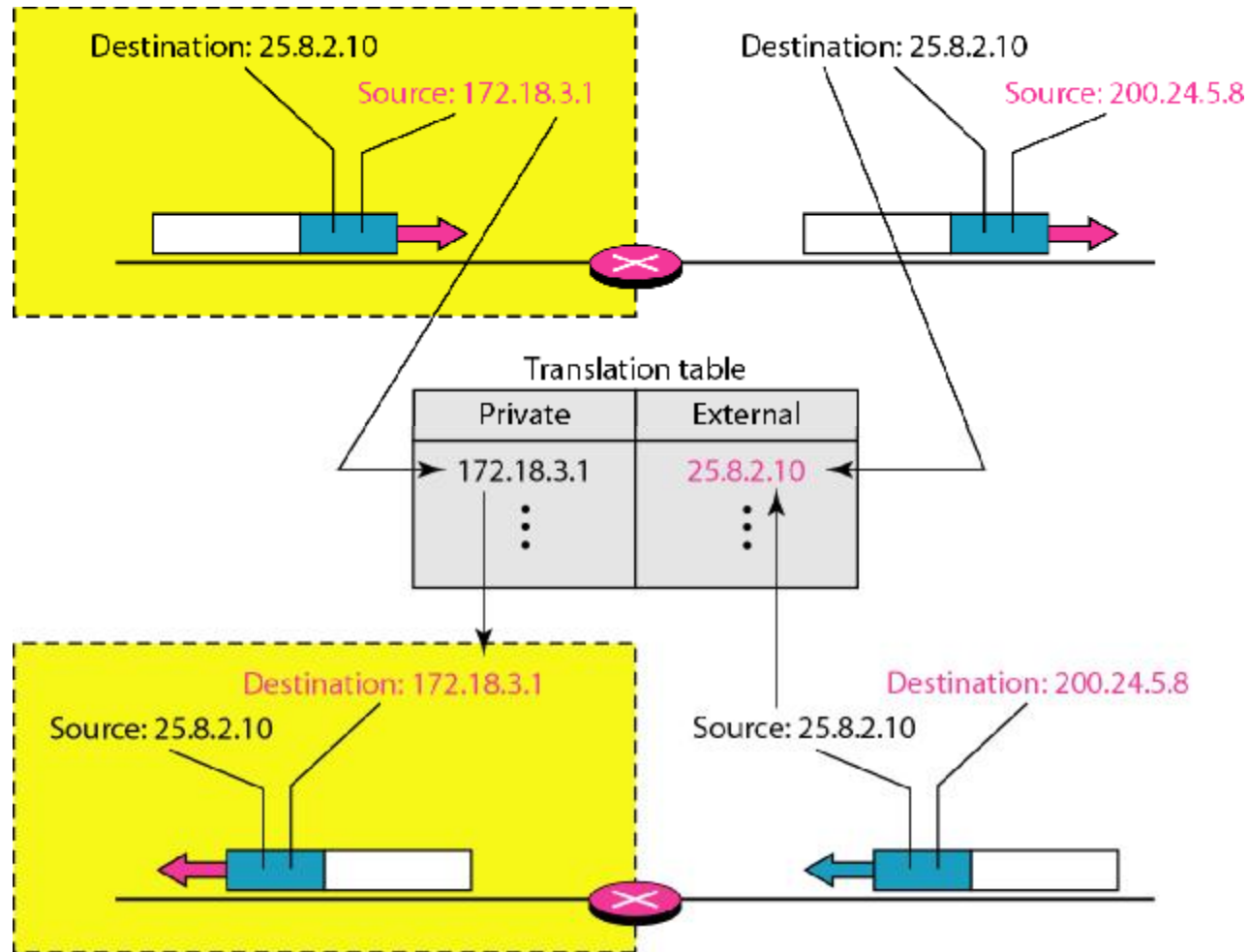


图19.12 NAT地址转换（使用一个IP地址，转换表只有两列）



运用IP地址池

- p 采用前面的方式，NAT路由器只有一个全球地址，因而只有一台专用网络主机能访问同样的外部主机；
- p 为了打破这种限制，NAT路由器使用了一种全球地址池技术，它不只使用一个全球地址，而是能使用多个地址；
- p 在这种情况下，有多台专用网络主机能够同时与同样的外部主机通信，因为每对地址都定义了一条连接；
- p 然而，还是存在一些不足：假定地址池中有四个可用的IP地址，则不能同时与外部建立四路以上的通信（即不能同时有四个以上的内部专用网络主机与外部通信）。（去掉“没有一台专用网络主机能同时访问两个外部服务器程序”）

同时运用IP地址和端口号

p为了能在专用网络主机和外部服务器程序之间建立多对多关系，需要获取转换表中更多的信息；

p例如，假定在一个专用网络中有两台主机，地址分别为172.18.3.1和172.18.3.2，需要访问一台地址为25.8.3.2的外部主机上的HTTP服务器，如果转换表有5列，而不是2列，其中包括传输层协议的源端和目的端的端口号，就可以消除二义性

| <i>Private Address</i> | <i>Private Port</i> | <i>External Address</i> | <i>External Port</i> | <i>Transport Protocol</i> |
|------------------------|---------------------|-------------------------|----------------------|---------------------------|
| 172.18.3.1 | 1400 | 25.8.3.2 | 80 | TCP |
| 172.18.3.2 | 1401 | 25.8.3.2 | 80 | TCP |
| ... | ... | ... | ... | ... |

表19.4 5列转换表

| <i>Private Address</i> | <i>Private Port</i> | <i>External Address</i> | <i>External Port</i> | <i>Transport Protocol</i> |
|------------------------|---------------------|-------------------------|----------------------|---------------------------|
| 172.18.3.1 | 1400 | 25.8.3.2 | 80 | TCP |
| 172.18.3.2 | 1401 | 25.8.3.2 | 80 | TCP |
| ... | ... | ... | ... | ... |

注意：当来自HTTP服务器的响应返回时，目的地址25.8.3.2和目的端口号1400的组合就确定了应该接收这一响应的内部网络主机。还须注意，要使这种转换能正常运作，临时端口号（1400和1401）必须是唯一的（实际系统中，“内部地址+内部临时端口号”会被NAT转换成“路由器外部接口地址+外部接口全局临时端口号”，5列->7列，从而保证其唯一性）。

NAT类型

pNAT需要修改IP头部中的IP地址信息（可能还需要修改传输层头部中的端口号字段）；

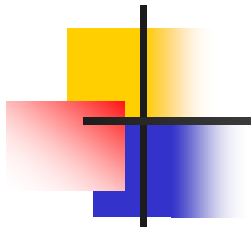
p源NAT（SNAT）：指对发起连接的IP头部中的源地址进行转换，可以实现内部用户访问外部网络的目的；通过将内部主机的私有地址转换为公有地址，使一个局域网中的多台主机使用少数的合法地址访问外部资源，有效的隐藏了内部局域网的主机IP地址，起到了安全保护的作用；

p目的NAT（DNAT）：对IP头部中的目的地址进行转换，通常用于隐藏一个对外提供服务的网络设备的真实IP地址，使客户端可以通过访问一个公网地址来访问这些服务器。

19-2 IPv6 地址

p 尽管有很多暂时解决问题的办法（比如无类寻址、NAT以及后面的动态主机配置协议DHCP），但是因特网地址耗尽依然是长期存在的问题；

p 这个问题与其他IP协议本身的问题（诸如安全性、实时音频和视频传输需求等）一起，促使IPv6的提出。



一个IPv6地址长度是128位（16个字节）。

Q：是IPv4地址空间的多少倍？

这使得地球上每平方米的面积就可以拥有 6×10^{23} 个数量级的唯一地址，即使地址分配低效也不用担心！

图19.14 IPv6地址用二进制与十六进制冒号标记法

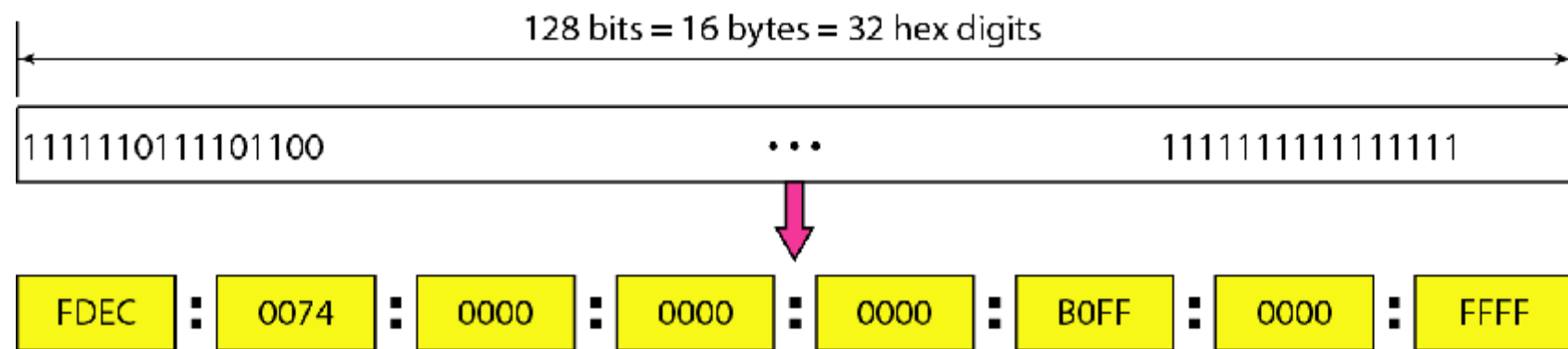
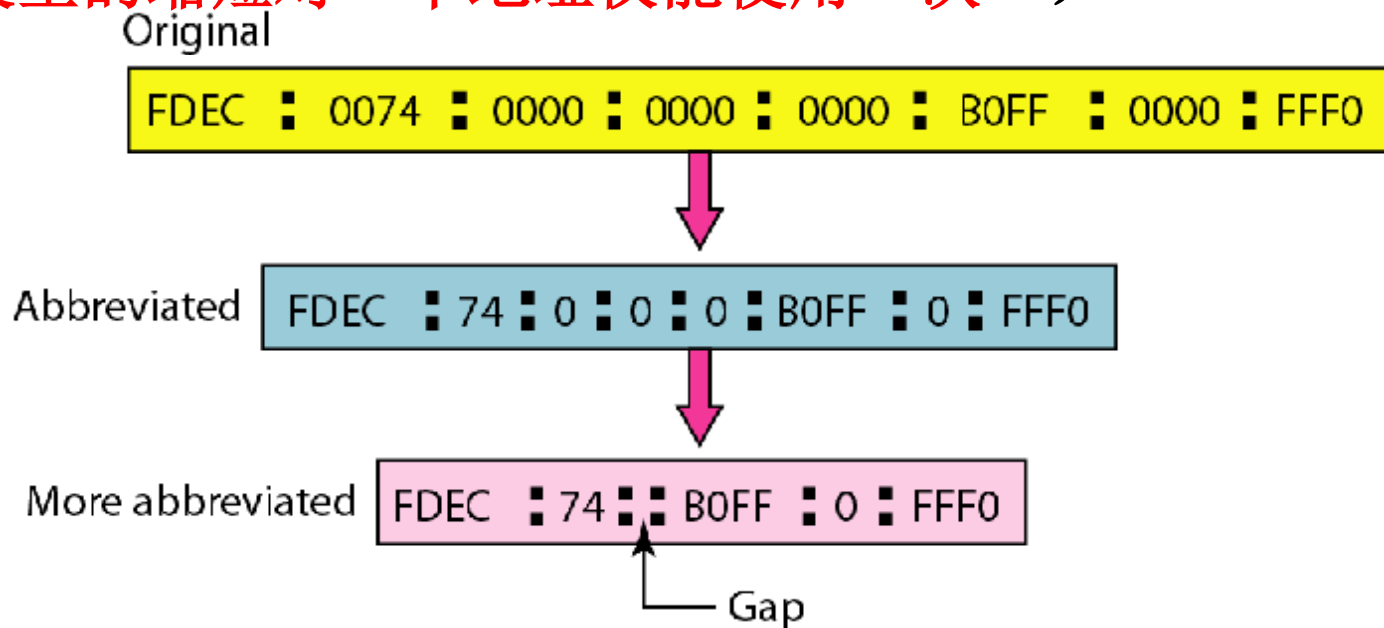


图19.15 缩短的IPv6地址

- 即使使用十六进制表示，IPv6地址也非常长，但许多数字是0；
- 可以将地址缩短：一个部分（即两个冒号之间的4个数字）中开始的一些0可以省略；
- 只有开始的一些0才可以省略，而在末尾的0不能被省略；
- 如果有几个连续的部分仅包含0，可用两个冒号来代替（注意：这种类型的缩短对一个地址仅能使用一次！）





例19.11

将地址 0:15::1:12:1213 扩展成原来的地址。

解：

首先需要将两倍冒号左边与原来模式左边对齐，然后将两倍冒号右边与原来模式右边插入我们所需要的多个0。

| |
|---|
| XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX |
| 0: 15: : 1: 12:1213 |

这就是说，原来的地址是：

| |
|---|
| 0000:0015:0000:0000:0000:0001:0012:1213 |
|---|

IPv6地址空间

- IPv6有非常大的地址空间，有 2^{128} 个可用的地址；
- IPv6地址设计者将地址空间划分成多个类，其中少数几个位称为类型前缀；
- 用类型前缀定义每个地址类；
- 这个类型前缀是可变长的，代码设计使得第一部分的代码都没有相同的，这样在给出一个地址时就不会有歧义，而类型前缀就能很容易确定

表19.5 IPv6地址的类型前缀

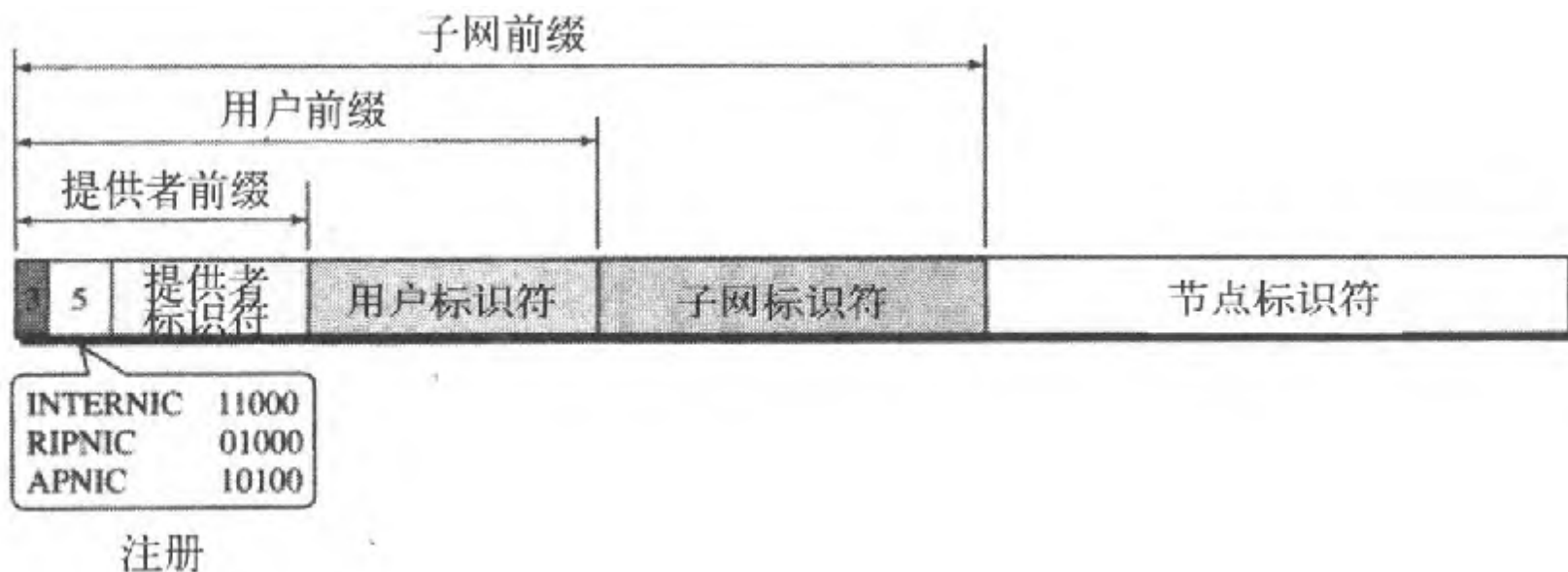
| <i>Type Prefix</i> | <i>Type</i> | <i>Fraction</i> |
|--------------------|---|-----------------|
| 0000 0000 | Reserved | 1/256 |
| 0000 0001 | Unassigned | 1/256 |
| 0000 001 | ISO network addresses | 1/128 |
| 0000 010 | IPX (Novell) network addresses | 1/128 |
| 0000 011 | Unassigned | 1/128 |
| 0000 1 | Unassigned | 1/32 |
| 0001 | Reserved | 1/16 |
| 001 | Reserved | 1/8 |
| 010 | Provider-based unicast addresses | 1/8 |

表19.5 IPv6地址的类型前缀（续）

| <i>Type Prefix</i> | <i>Type</i> | <i>Fraction</i> |
|--------------------|------------------------------------|-----------------|
| 011 | Unassigned | 1/8 |
| 100 | Geographic-based unicast addresses | 1/8 |
| 101 | Unassigned | 1/8 |
| 110 | Unassigned | 1/8 |
| 1110 | Unassigned | 1/16 |
| 1111 0 | Unassigned | 1/32 |
| 1111 10 | Unassigned | 1/64 |
| 1111 110 | Unassigned | 1/128 |
| 1111 1110 0 | Unassigned | 1/512 |
| 1111 1110 10 | Link local addresses | 1/1024 |
| 1111 1110 11 | Site local addresses | 1/1024 |
| 1111 1111 | Multicast addresses | 1/256 |

图19.16 基于提供者的单播地址

- 单播地址（unicast address）定义一个单独的计算机，发送到单播地址的分组必须传递给这个指定的计算机：
- IPv6地址协议定义了两类型的单播地址：基于地理的地址（留给将来定义）和基于提供者的地址；
- 基于提供者的地址通常由普通的主机作为一个单播地址使用



单播地址字段

p类型标识符：3位，010；

p注册标识符：5位，指出注册地址的机构，目前已经定义了3个注册中心，INTERNIC（11000）是北美的注册中心，RIPNIC（01000）是欧洲的注册中心，APNIC（10100）是亚洲和太平洋国家的注册中心；

p提供者标识符：可变长字段，标识因特网接入提供者，推荐长度16位；

p用户标识符：当一个组织机构通过一个提供者接入因特网时，就给它分配一个用户标识符，推荐长度是24位；

p子网标识符：每个用户可有多个子网，而每个子网可以有不同的标识符，它在用户的地区范围内定义一个特定的网络，推荐长度32位；

p节点标识符：定义连接到子网的节点的标识，推荐长度48位，使它和以太网使用的48位的链路（物理）地址相兼容。

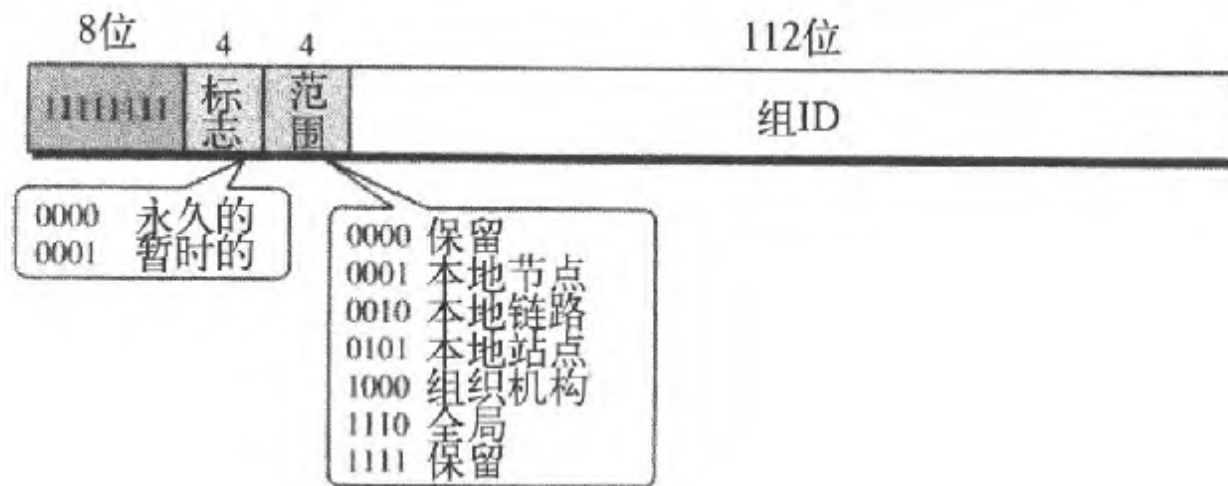
图19.17 IPv6中的多播地址

多播地址用于定义一组主机而不是一个主机（该地址不分配给特定的主机或设备），发送给多播（也叫组播，multicast）地址的分组必须传递到该组中的每一个成员；

第二个字段是定义组地址的一个标志：永久的或暂时的；

永久的组地址由因特网管理机构定义，并可在任何时刻对它进行访问；而暂时的组地址只是临时使用，例如，参加电话会议的系统就可使用暂时的组地址；

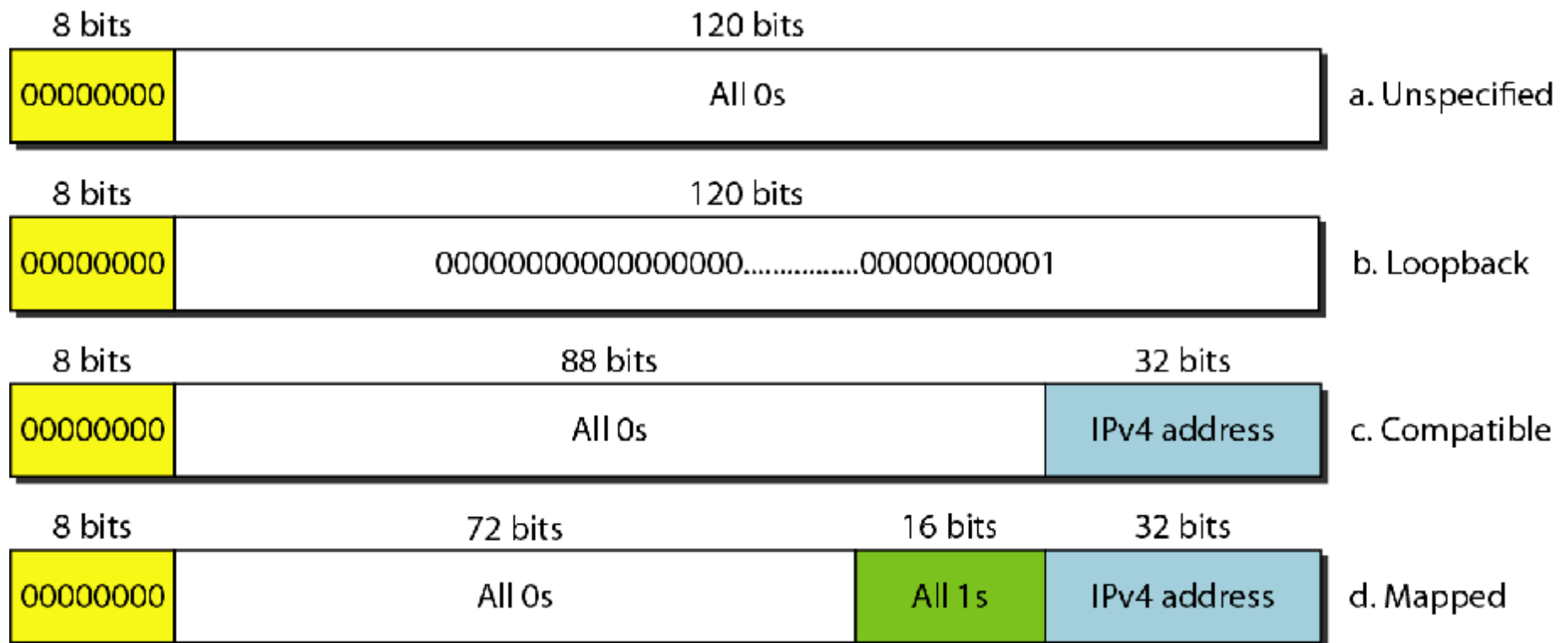
第三个字段定义组地址的范围



IPv6中的任播地址

- IPv6定义了任播地址的新概念；
- 任播地址（anycast address）与多播地址一样也定义一组计算机，然而，发送到任播地址的分组必须传递给该组成员中的一个且仅有一个，即最近的一个（具有最短路径的一个）；
- 虽然任播地址的定义还存在争议，但因特网中对网络服务商ISP在大范围逻辑区域内分布的所有路由器赋予一个任播地址是可以接受的，由此，ISP以外的路由器传递给ISP的分组发送到最近的ISP路由器；
- 没有给任播地址分配块。

图19.18 IPv6中保留地址



IPv6中保留地址

- p**未指明地址：当主机不知道自己的地址时，它发送查询以便找出其地址：
- p**环回地址：主机用来测试它自己，而不需要连接到网络上（**IPv4环回地址？**）；
- p**兼容地址：用于从IPv4到IPv6的转换（见第20章）；当一台使用IPv6协议的计算机要发送报文到另一台使用IPv6协议的计算机，但报文需要通过网络某一部分还仍旧使用IPv4协议操作时，就需要该地址；
- p**映射地址：也用于从IPv4到IPv6的转换，它用于一台计算机已安装了IPv6但要发送分组到一台仍使用IPv4协议的计算机。

图19.19 IPv6中本地地址

- 一个组织机构想要使用IPv6协议，但还没有连接到全球因特网，就要用这种地址，换言之，它们提供专用网络地址；
- 在该组织机构网络以外没有人能将报文发送给使用这些地址的节点；
- 本地链路地址（link local address）用于一个孤立的子网，而本地站点地址（site local address）用于具有几个子网的站点。

