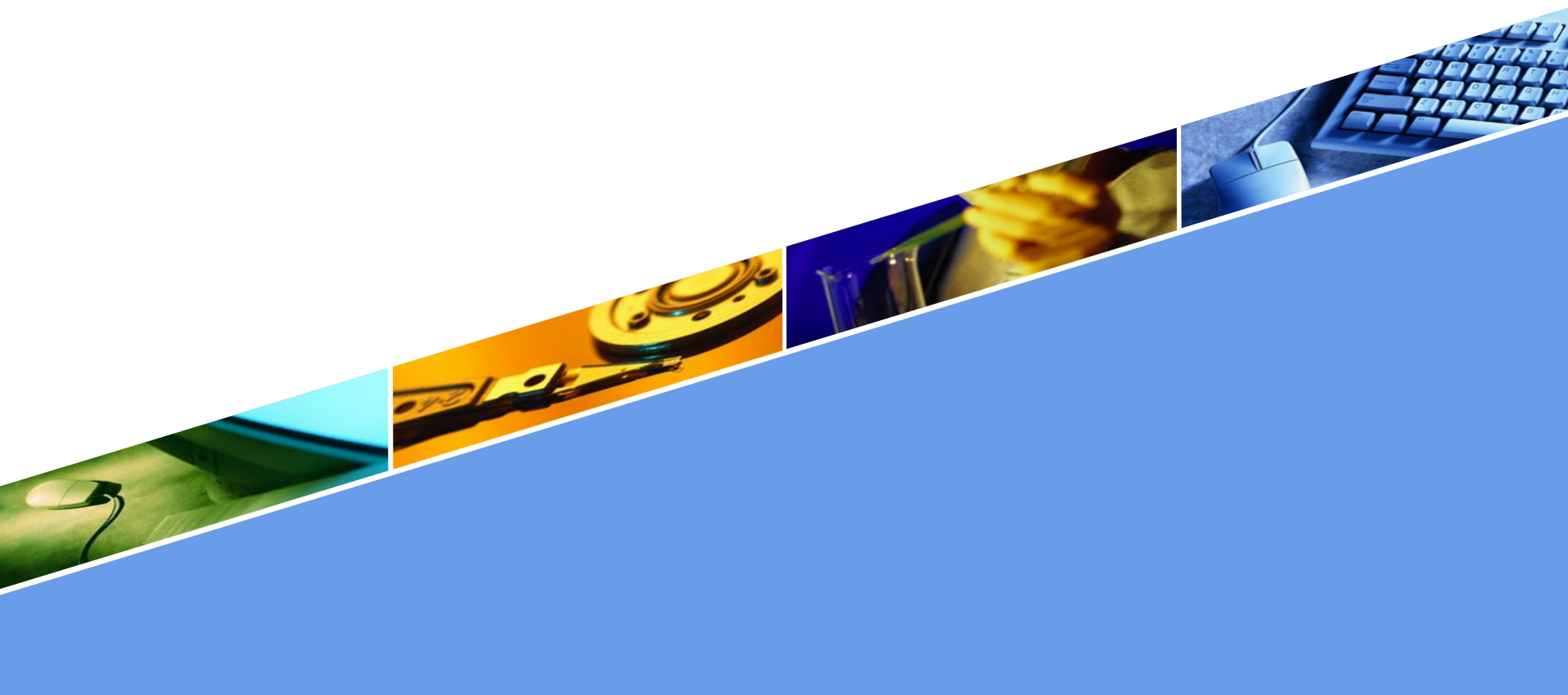


IP地址规划



主题 1



1 子网及子网掩码

2 划分子网

1 超网及地址前缀

3 构造超网

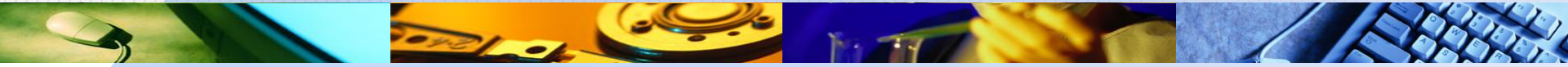
IPv4地址背景

- ❖ IPv4有类别编址：IP地址分为五类：A，B，C，D和E类，每一个地址类满足一个不同最大尺寸的网络。

| Class | Set Bits (First Octet) | Natural Mask | Number of Addresses |
|--------------|-----------------------------------|---------------------|----------------------------|
| A | 0xxxxxxx | 255.0.0.0 | $2^{24}-2$ |
| B | 10xxxxxx | 255.255.0.0 | $2^{16}-2$ |
| C | 110xxxxx | 255.255.255.0 | 2^8-2 |

- 自然掩码:区分网络号和主机号
- 类别决定网络规模
- 主机地址中减去2个地址：
 - 全1：广播地址
 - 全0：网络自身地址

划分子网的背景



在 ARPANET 的早期，IP 地址的设计不合理。

- IP 地址空间的利用率有时很低。
- 两级的 IP 地址不够灵活。

❖ 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。

划分子网的基本思路

- ❖ 划分子网属于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>}

子网划分举例

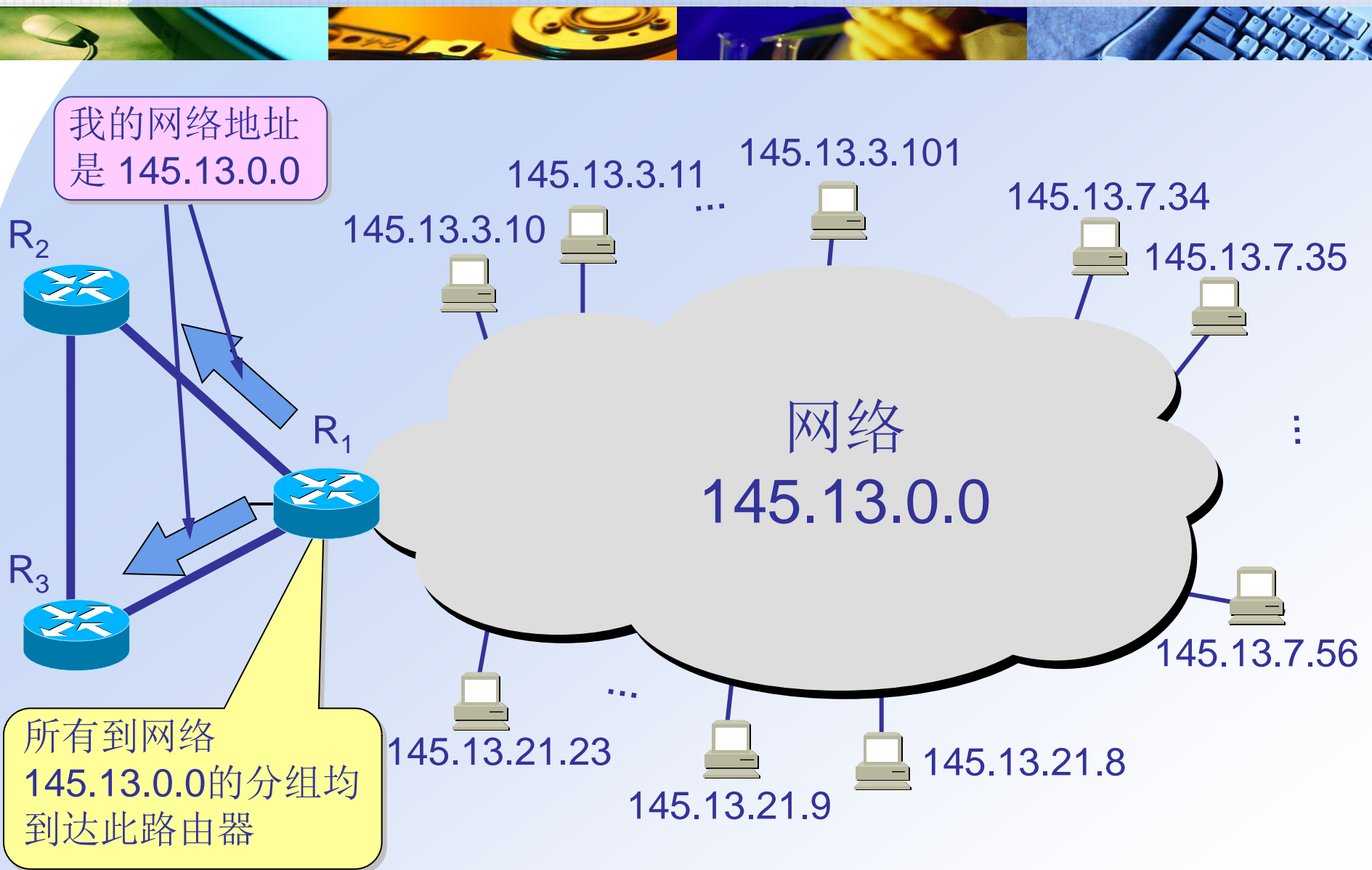
- 将B类网168.95.0.0划分为8个子网（需3位），每个子网中至多8192个主机（ 2^{13} ）。
- 子网号全“0”和全“1”的不可用，实际可用6个子网。

| | | | | |
|----------|----------|-----|-------|----------|
| 10101000 | 01011111 | 000 | 00000 | 00000000 |
| 10101000 | 01011111 | 001 | 00000 | 00000000 |
| 10101000 | 01011111 | 010 | 00000 | 00000000 |
| 10101000 | 01011111 | 011 | 00000 | 00000000 |
| 10101000 | 01011111 | 100 | 00000 | 00000000 |
| 10101000 | 01011111 | 101 | 00000 | 00000000 |
| 10101000 | 01011111 | 110 | 00000 | 00000000 |
| 10101000 | 01011111 | 111 | 00000 | 00000000 |

↑
网络号

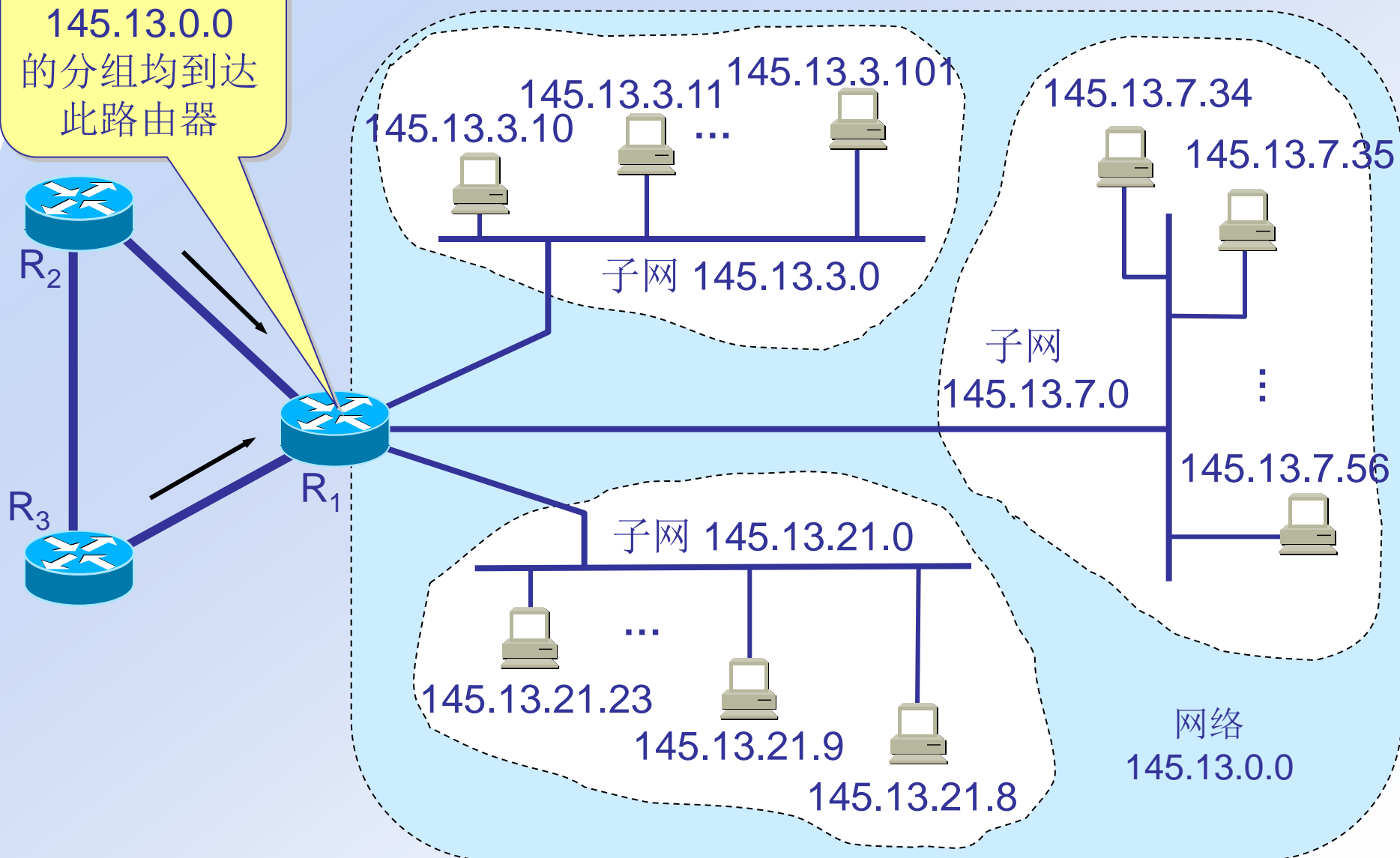
↑
子网号

一个未划分子网的 B 类网络 145.13.0.0



划分为三个子网后对外仍是一个网络

所有到达网络
145.13.0.0
的分组均到达
此路由器



划分子网后变成了三级结构

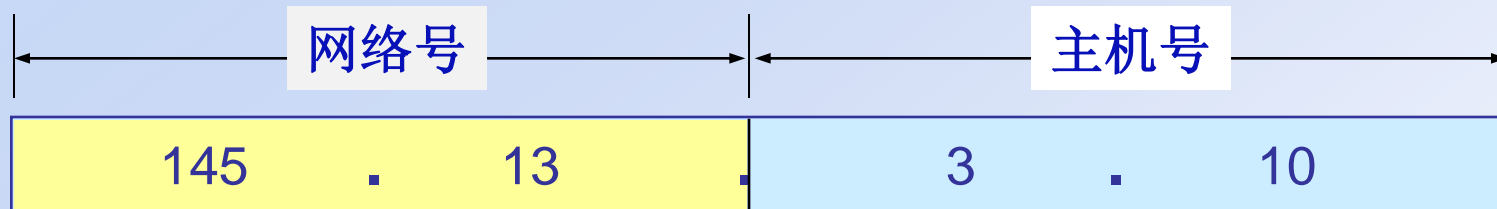
- ❖ 当没有划分子网时，IP 地址是两级结构。
- ❖ 划分子网后 IP 地址就变成了三级结构。
- ❖ 划分子网只是把 IP 地址的主机号 host-id 这部分进行再划分，而不改变 IP 地址原来的网络号 net-id。

子网掩码

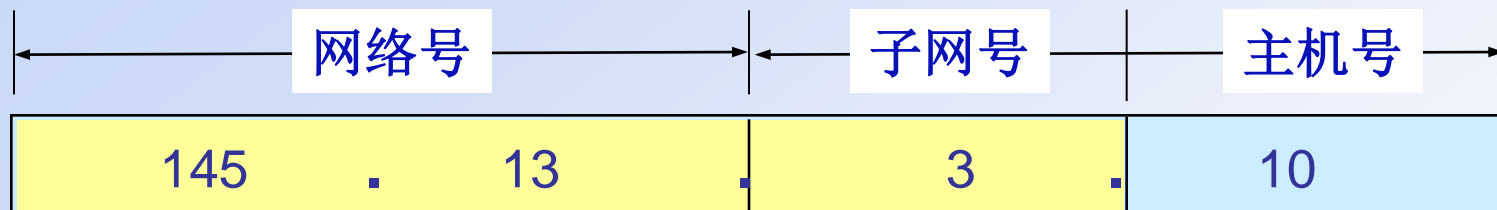
- ❖ 从一个 IP 数据报的首部并无法判断源主机或目的主机所连接的网络是否进行了子网划分。
- ❖ 使用子网掩码(subnet mask)可以找出 IP 地址中的子网部分。

IP 地址的各字段和子网掩码

两级 IP 地址



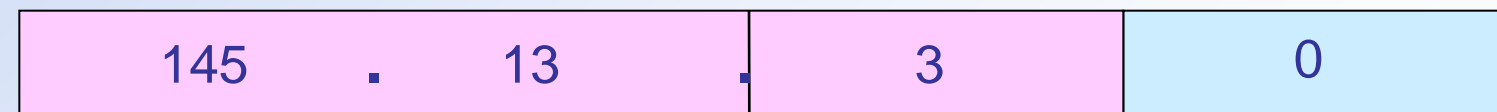
三级 IP 地址



三级 IP 地址的子网掩码



子网的网络地址



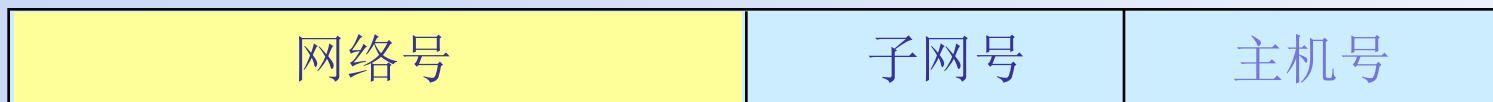
(IP 地址) AND (子网掩码) = 网络地址



两级 IP 地址

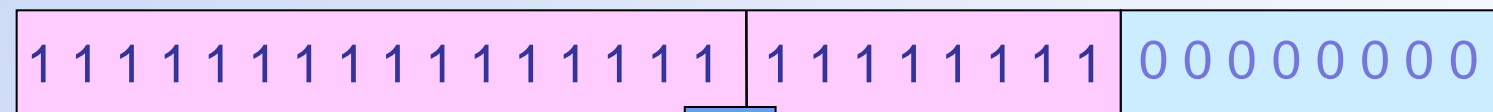


三级 IP 地址

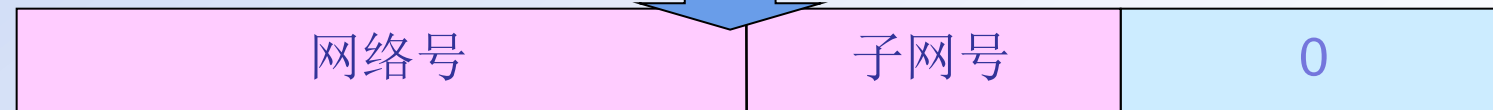


逐位进行 AND 运算

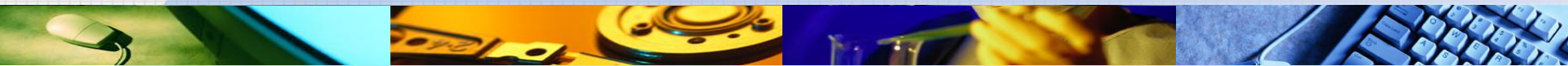
三级 IP 地址
的子网掩码



子网的
网络地址



子网掩码是子网的重要属性



- ❖ 子网掩码是一个子网的重要属性。
- ❖ 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。
- ❖ 若一个路由器连接在两个子网上就拥有两个网络地址和两个子网掩码。

【例子】 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 |
|-----|---|----|---|----|---|---|

【例4-3】 在上例中，若子网掩码改为 255.255.224.0。试求网络地址，讨论所得结果。

(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 255.255.224.0

| | | | |
|----------|----------|----------|----------|
| 11111111 | 11111111 | 11100000 | 00000000 |
|----------|----------|----------|----------|

(d) IP 地址与子网掩码逐位相与

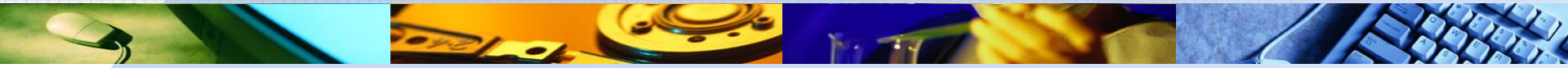
| | | | | | | |
|-----|---|----|---|----------|---|---|
| 141 | . | 14 | . | 01000000 | . | 0 |
|-----|---|----|---|----------|---|---|

(e) 网络地址（点分十进制表示）

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

不同的子网掩码得出相同的网络地址。
但不同的掩码的效果是不同的。

使用子网掩码的分组转发过程



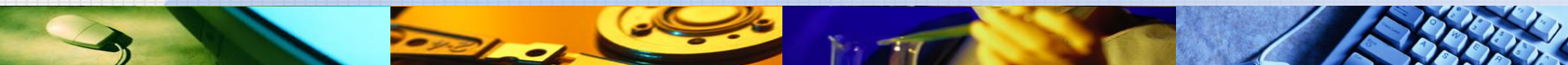
- ❖ 在不划分子网的两级 IP 地址下，从 IP 地址得出网络地址是个很简单的事。
- ❖ 但在划分子网的情况下，从 IP 地址却不能唯一地得出网络地址来，这是因为网络地址取决于那个网络所采用的子网掩码，但数据报的首部并没有提供子网掩码的信息。
- ❖ 因此分组转发的算法也必须做相应的改动。

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) 先用各直连网络的子网掩码和 D 逐位相“与”，看是否和相应的网络地址匹配。若匹配，则将分组直接交付。否则就是间接交付，执行(3)。
- (3) 若路由表中有目的地址为 D 的特定主机路由，则将分组传送给指明的下一跳路由器；否则，执行(4)。
- (4) 对路由表中的每一行的子网掩码和 D 逐位相“与”，若其结果与该行的目的网络地址匹配，则将分组传送给该行指明的下一跳路由器；否则，执行(5)。
- (5) 若路由表中有一个默认路由，则将分组传送给路由表中所指明的默认路由器；否则，执行(6)。
- (6) 报告转发分组出错。

通信示例一掩码



R₁ 的路由表

| 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 |

128.30.33.13

H₁
子网1:
网络地址 128.30.33.0
子网掩码 255.255.255.128

128.30.33.1 0

128.30.33.130 1

R₁

子网2: 网络地址 128.30.33.128
子网掩码 255.255.255.128

0 128.30.33.129
R₂
1 128.30.36.2

H₂ 128.30.33.138

H₃ 128.30.36.12

子网3: 网络地址 128.30.36.0
子网掩码 255.255.255.0

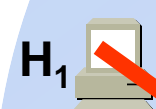
通信示例一掩码

要发送的分组的目的 IP 地址：128.30.33.138

R_1 的路由表

| 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 |

128.30.33.13



子网1:

网络地址 128.30.33.0

子网掩码 255.255.255.128

128.30.33.1 0



R_1

128.30.33.130 1

子网2: 网络地址 128.30.33.128

子网掩码 255.255.255.128

0 128.30.33.129



1 128.30.36.2

H_2

128.30.33.138



H_3

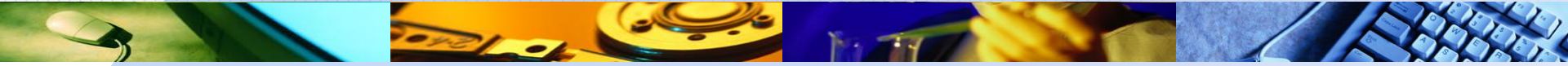
128.30.36.12



子网3: 网络地址 128.30.36.0

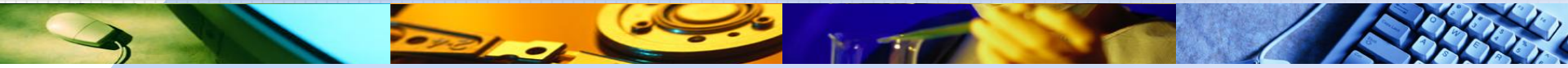
子网掩码 255.255.255.0

通信示例－掩码



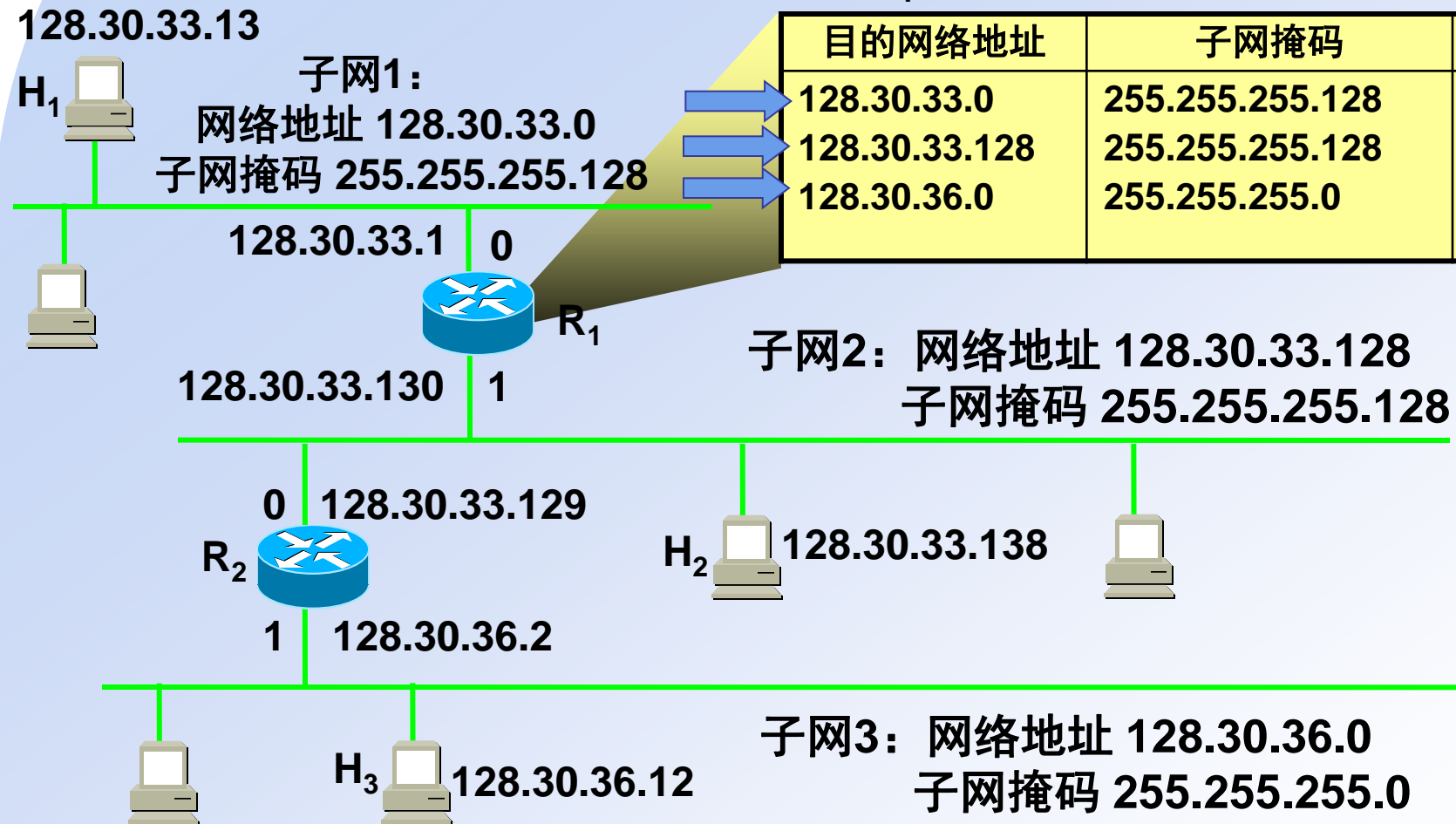
假设 H_1 要发送的分组是给 H_2 ，主机 H_1 首先将本子网的子网掩码 255.255.255.128，与分组的 IP 地址 128.30.33.138 逐比特相“与” (AND 操作)，得出128.30.33.128，它不等于 H_1 的网络地址（ 128.30.33.0），这说明 H_2 和 H_1 不在同一子网上，因此 H_1 知道不能将分组直接交付给 H_2 ，而是将分组交给路由器 R_1 ，然后逐项查找路由表，由 R_1 转发。

通信示例 - 掩码



R₁ 的路由表

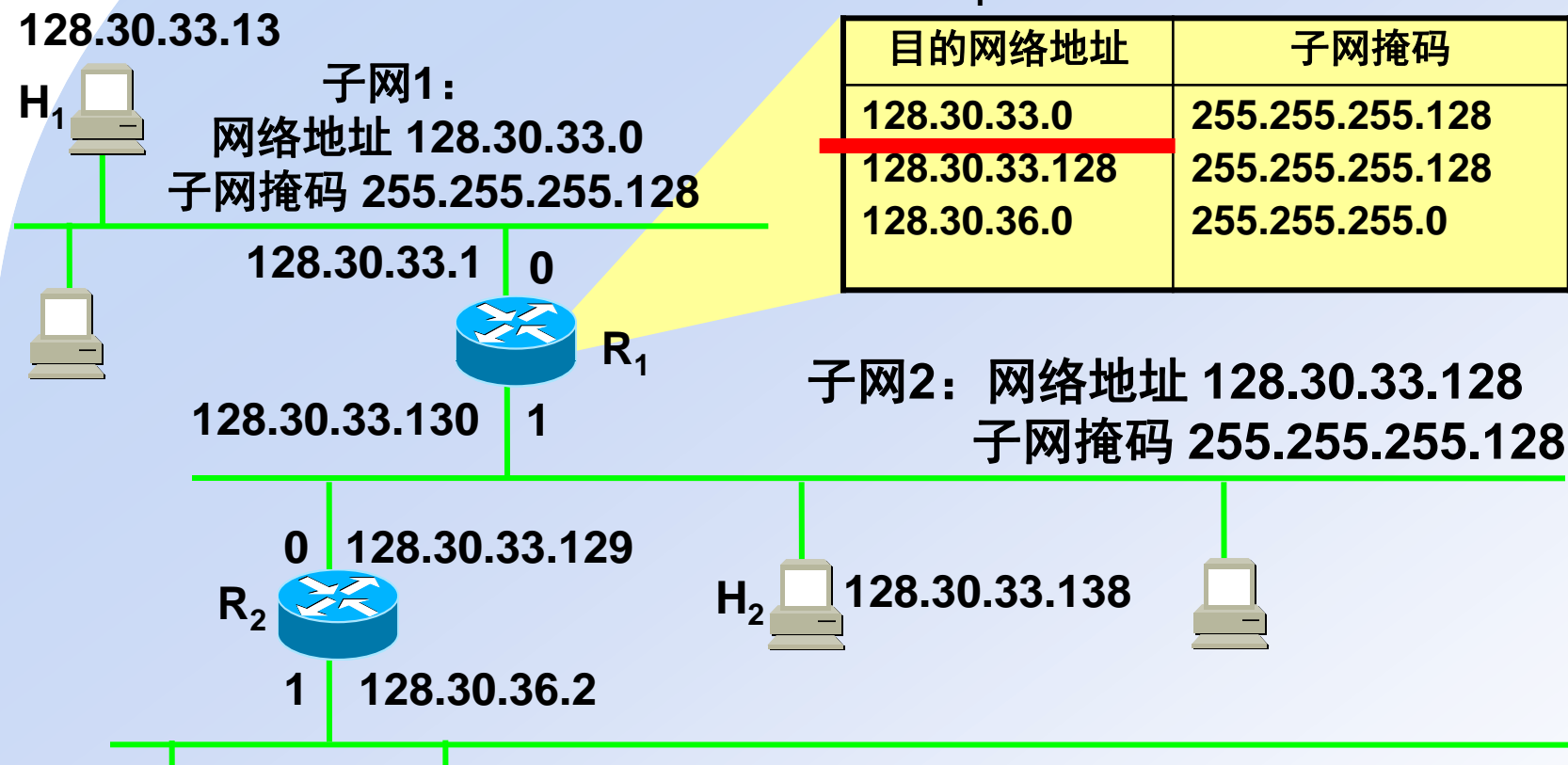
| 目的网络地址 | 子网掩码 | 下一跳 |
|---------------|-----------------|----------------|
| 128.30.33.0 | 255.255.255.128 | 接口 0 |
| 128.30.33.128 | 255.255.255.128 | 接口 1 |
| 128.30.36.0 | 255.255.255.0 | R ₂ |



通信示例 - 掩码

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 |



❖ R₁ 收到的分组的目的 IP 地址为 128.30.33.138，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，得网络地址为 128.30.33.128。和第 1 项的目的网络地址不一致。

通信示例 - 掩码

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 |

128.30.33.13

子网1:
网络地址 128.30.33.0
子网掩码 255.255.255.128

128.30.33.1 0



R₁

128.30.33.130 1

子网2: 网络地址 128.30.33.128
子网掩码 255.255.255.128

0 128.30.33.129



1 128.30.36.2

H₂ 128.30.33.138



H₃ 128.30.36.12

子网3: 网络地址 128.30.36.0
子网掩码 255.255.255.0

❖ 接着拿第 2 个项目的子网掩码和 128.30.33.138 逐比特 AND 操作，得网络地址为 128.30.33.128。和第 2 项的目的网络地址一致。这表明子网 2 就是收到的分组所要寻找的目的网络。

主题 2



1 子网及子网掩码

2 划分子网

1 超网及地址前缀

3 构造超网

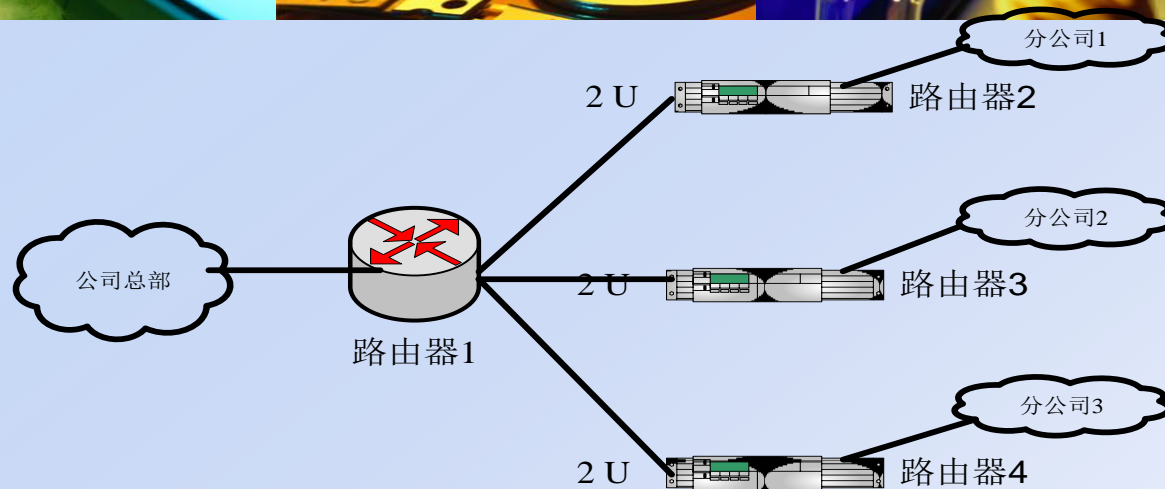
2种类型的子网划分方法

- ❖ 定长掩码子网划分：单一网络中所有子网使用相同的子网掩码。
 - 简单高效、易于配置，但所有子网大小相同。
- ❖ 可变长掩码子网划分：单一网络中不同子网使用不同的子网掩码。
 - 灵活，更有效利用主机地址，但实现复杂，所有路由器都要支持可变子网。

定长掩码子网划分

- ❖ 许多组织部门工作组规模差别不大，一般使用定长掩码进行地址管理。
- ❖ 定长掩码管理很容易理解，并且容易实现。
- ❖ 定长掩码子网划分是将一个网络分段为很多相同大小的子网。
 - 其中子网掩码的长度确定了子网的数目及每一子网的大小。
 - 这种子网划分的方法简单高效，每个设备使用相同的掩码，所有子网中都有相同的地址数量。

定长掩码子网划分实例



- ❖ 假设公司已经获得了一个C类网络地址192.168.153.0
- ❖ 采用定长掩码子网划分方法为该公司分配IP地址

| 子网 | IP地址数量 |
|------|--------|
| 总部 | 25 |
| 分公司1 | 14 |
| 分公司2 | 13 |
| 分公司3 | 7 |
| WAN1 | 2 |
| WAN2 | 2 |
| WAN3 | 2 |

确定子网掩码

❖ 网络应支持7个子网，在一个最大的子网中需要25个地址。

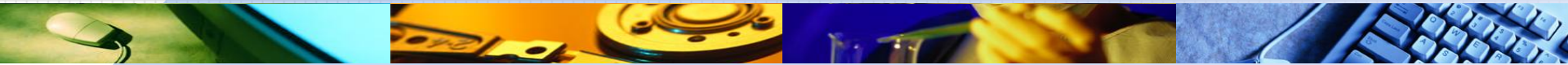
- **至少5个主机位。这也就是说，应在掩码中至少有5个0，为什么？：**
 $2^4-2=14$ （不够用）， $2^5-2=30$ （足够用），因此主机位数应为5。

❖ C类地址子网划分表：

| 子网位数 | 子网数量 | 主机位数 | 主机数量 | 掩码 |
|------|------|------|------|--------------------|
| 2 | 2 | 6 | 62 | 255. 255. 255. 192 |
| 3 | 6 | 5 | 30 | 255. 255. 255. 224 |
| 4 | 14 | 4 | 14 | 255. 255. 255. 240 |
| 5 | 30 | 3 | 6 | 255. 255. 255. 248 |
| 6 | 62 | 2 | 2 | 255. 255. 255. 252 |

是否能够找到一个掩码支持7个子网，
并且每个子网中有25台主机？

方法1—使用无编址的网络接口



- ❖ 也叫做无编址的IP，许多常用的路由器都拥有这个特性。
 - 缺点是不能直接对这些网络接口进行测试和管理。需要在管理上和地址节约上做出选择。

❖ 子网数将缩减成4个：

| 子网 | IP地址数量 |
|------|--------|
| 总部 | 25 |
| 分公司1 | 14 |
| 分公司2 | 13 |
| 分公司3 | 7 |

❖ 掩码可确定为：255.255.255.224

方法1的子网划分结果

❖ 使用此定长掩码可划分如下6个子网：

192.168.153.32 192.168.153.128

192.168.153.64 192.168.153.160

192.168.153.96 192.168.153.192

❖ 将四个子网分配给总部和3个子公司，具体分配如下：

总公司： 192.168.153.32/ 255.255.255.224

分公司1： 192.168.153.64/ 255.255.255.224

分公司2： 192.168.153.96/ 255.255.255.224

分公司3： 192.168.153.128/ 255.255.255.224

❖ 其它二个子网留待将来扩展使用。

方法2—请求更大的地址块

❖ 申请两个C类地址：

192.168.152.0和192.168.153.0

- 在总部LAN使用192.168.152.0地址，掩码是255.255.255.0。
- 剩余的6个LAN和WAN连接使用192.168.153.0地址。
 - 利用掩码255.255.255.224进行子网划分时会有6个子网。
 - 而且每个子网中可以有30个主机地址。
 - 完全满足这6个子网的需求。

方法2的子网划分结果

具体子网划分如下：

- ❖ 总公司： 192.168.152.0/255.255.255.0
- ❖ 分公司1： 192.168.153.32/ 255.255.255.224
- ❖ 分公司2： 192.168.153.64/ 255.255.255.224
- ❖ 分公司3： 192.168.153.96/ 255.255.255.224
- ❖ WAN1： 192.168.153.128/ 255.255.255.224
- ❖ WAN2： 192.168.153.160/ 255.255.255.224
- ❖ WAN3： 192.168.153.192 / 255.255.255.224

方法3—使用“子网零”



- ❖ 在早期的子网划分标准（RFC 950）中，不能使用全0或全1做为二进制子网标识。在RFC 1812中，这个限制已被取消。
- ❖ 示例中就子网0（全零子网）。
 - 选择了255.255.255.224做为掩码，如果使用了子网0，则可以有7个可用的子网来满足要求，此时会有足够的主机地址。

方法3的子网划分结果

具体子网划分如下：

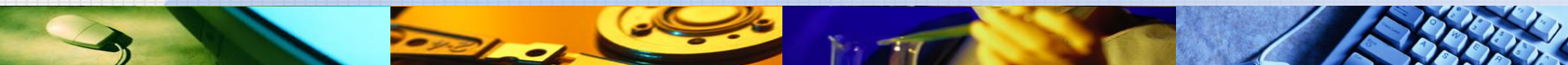
- ❖ 总公司： 192.168.153.0/ 255.255.255.224
- ❖ 分公司1： 192.168.153.32/ 255.255.255.224
- ❖ 分公司2： 192.168.153.64/ 255.255.255.224
- ❖ 分公司3： 192.168.153.96/ 255.255.255.224
- ❖ WAN1： 192.168.153.128/ 255.255.255.224
- ❖ WAN2： 192.168.153.160/ 255.255.255.224
- ❖ WAN3： 192.168.153.192 / 255.255.255.224

可变长掩码子网划分



- ❖ 为了充分利用地址空间，有必要将网络划分不同大小。
- ❖ 划分不同大小的子网是通过将子网掩码设置成可变长的机制实现的。
- ❖ 这种子网的划分更能有效的进行地址的分配，并能将一个工作组更好地映射到子网上。
- ❖ 需要在同一网络上识别多个子网掩码

可变长掩码子网划分实例



| Workgroup | Groups | Size/Group(Hosts) |
|----------------|--------|--------------------|
| Engineering | 3 | 400 => /23 (510) |
| Marketing | 1 | 1950 => /21 (2046) |
| Administration | 1 | 200 => /24 (254) |
| Sales | 15 | 35-90 => /25 (126) |
| R&D | 1 | 150 => /24 (254) |
| Surpport | 22 | 10-40 => /26 (62) |

某企业申请了一B类网络地址：136.178.0.0

可变长子网划分结果

R&D and Admin are both /24 so we can assign them:

136.178.ssss ssss.hhhh hhhh/24 so

136.178.0000 0000.xxxx xxxx/24 (136.178. 0.0/24) Admin

136.178.0000 0001.xxxx xxxx/24 (136.178. 1.0/24)R&D

Marketing is /21, we can assign it:

136.178.ssss shhh.hhhh hhhh/21 so

136.178.0000 1xxx.xxxx xxxx/21(136.178.8.0/21)

Engineering is /23, we can assign :

136.178.ssss sshh.hhhh hhhh/23 so

136.178.0000 001x.xxxx xxxx/23(136.178.2.0/23)Engineering 1

136.178.0000 010x.xxxx xxxx/23(136.178.4.0/23) Engineering 2

136.178.0000 011x.xxxx xxxx/23 (136.178.6.0/23) Engineering 3

.....



子网划分方法不是唯一的!

主题 3



1 子网及子网掩码

2 划分子网

1 超网及地址前缀

3 构造超网

提出“超网”的背景

- ❖ A类和B类地址: 数量少, 地址空间大;
- ❖ C类地址: 数量多 (上百万), 但地址空间小。
- ❖ 满足组织机构对大地址空间的需求的解决方法:
 - 在需要A类和B类地址的地方分配几个C类地址。
 - 以C类地址代替A类地址或者B类地址会发生什么情况呢?
- ❖ C类网络数目的激增使得路由器中发布的路由信息激增, 因特网主干网上的路由表中的项目数急剧增长, 路由器中的内存和处理能力需求也以指数级增长。
- ❖ IETF提出无分类编址方法, 它的正式名字是无分类域间路由选择 CIDR (Classless Inter-Domain Routing)。

CIDR本质上是在分类的网络的基础上划分超网, 即基于超网聚合进行层次化编址。

CIDR 最主要的特点

- ❖ CIDR 消除了传统的 A 类、B 类和 C 类地址，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。

无分类的两级编址

❖ 无分类的两级编址的记法是：

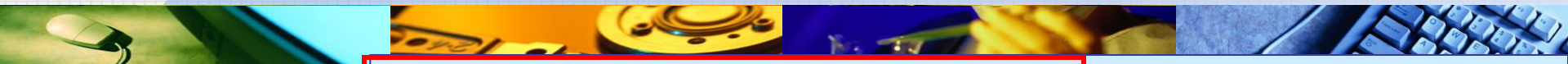
$$\text{IP地址} ::= \{ \langle \text{网络前缀} \rangle, \langle \text{主机号} \rangle \}$$

- ❖ CIDR 还使用“斜线记法”(slash notation)，它又称为CIDR记法，即在IP地址上面加上一个斜线“/”，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”。

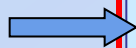
CIDR 地址块

- ❖ 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。

128.14.32.0/20 表示的地址 (2^{12} 个地址)



最小地址

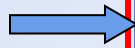


```
10000000 00001110 00100000 00000000
10000000 00001110 00100000 00000001
10000000 00001110 00100000 00000010
10000000 00001110 00100000 00000011
10000000 00001110 00100000 00000100
10000000 00001110 00100000 00000101
```

...

...

```
10000000 00001110 00101111 11111011
10000000 00001110 00101111 11111100
10000000 00001110 00101111 11111101
10000000 00001110 00101111 11111110
10000000 00001110 00101111 11111111
```



最大地址

所有地址
的 **20** 位
前缀都是
一样的

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 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。

CIDR 记法的其它形式

- ❖ 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

1 1 1 1 1 1 1 1 0

255 192 0 0

掩码中有 10 个连续的 1

- ❖ 另一种简化表示法：在网络前缀后面加一个 “*”
例如：00001010 00*

主题 4



1 子网及子网掩码

2 划分子网

1 超网及地址前缀

3 构造超网

构成超网

- ❖ 前缀长度不超过 23 位的 CIDR 地址块都包含了多个 C 类地址。
- ❖ 这些 C 类地址合起来就构成了超网。
- ❖ 网络前缀越短，其地址块所包含的地址数就越多。

划分超网应遵循的规则

- ❖ 1、CIDR块中地址的数目是2的幂；
- ❖ 2、地址块的端地址必须落在位边界内；
- ❖ 3、地址块中的地址必须是连续的（不能出现地址空洞）

违背规则2的例子

Original Address Mask 255.255.255.0

192.92.239.0 to
192.92.255.0

192.92.239.0/20?

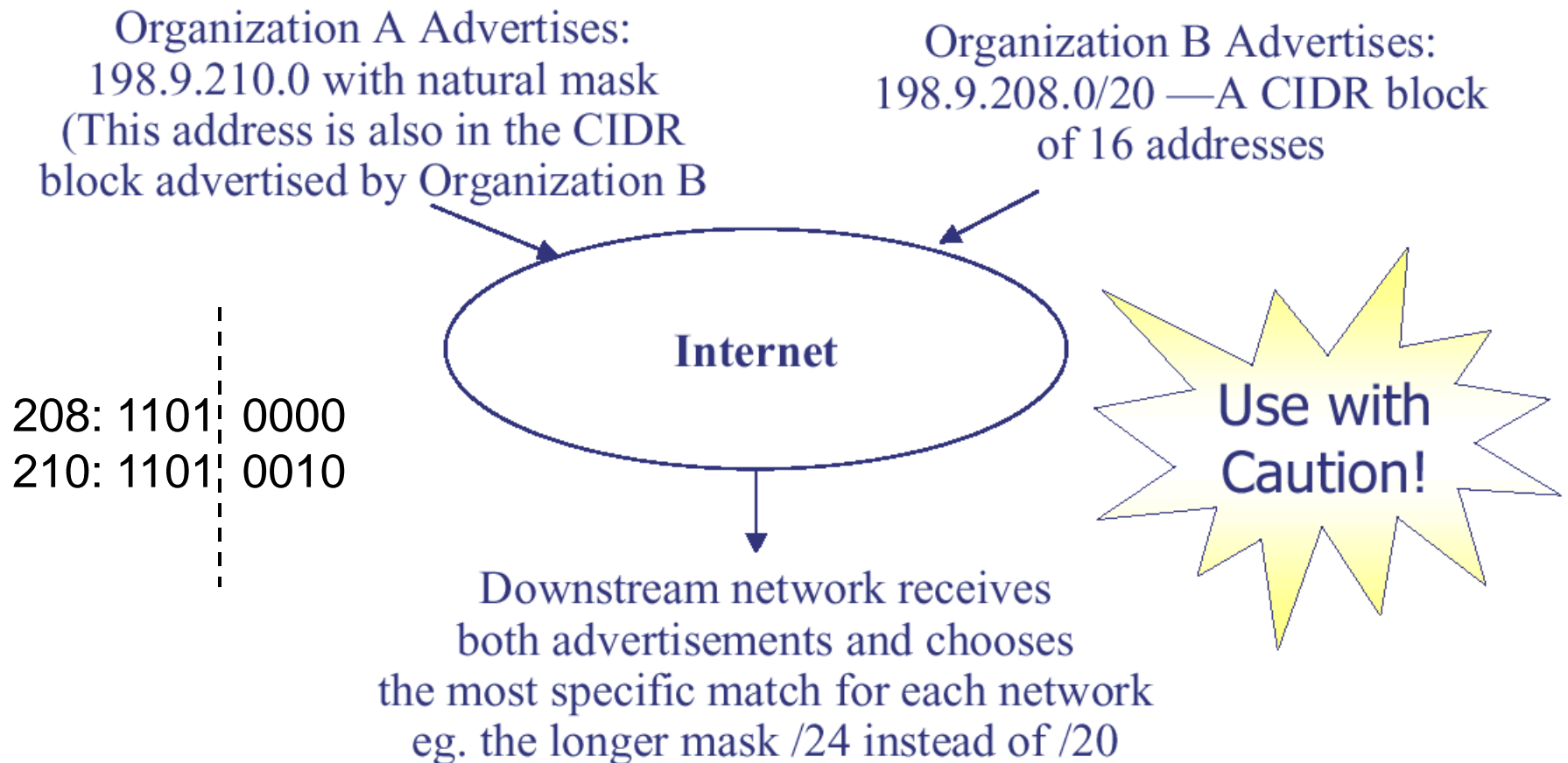
Invalid CIDR block!

| | .XXXX | XXXX. |
|--------------|-------|-------|
| 192.92.239.0 | 1110 | 1111 |
| 192.92.240.0 | 1111 | 0000 |
| 192.92.241.0 | 1111 | 0001 |
| ... | | |
| 192.92.253.0 | 1111 | 1101 |
| 192.92.254.0 | 1111 | 1110 |

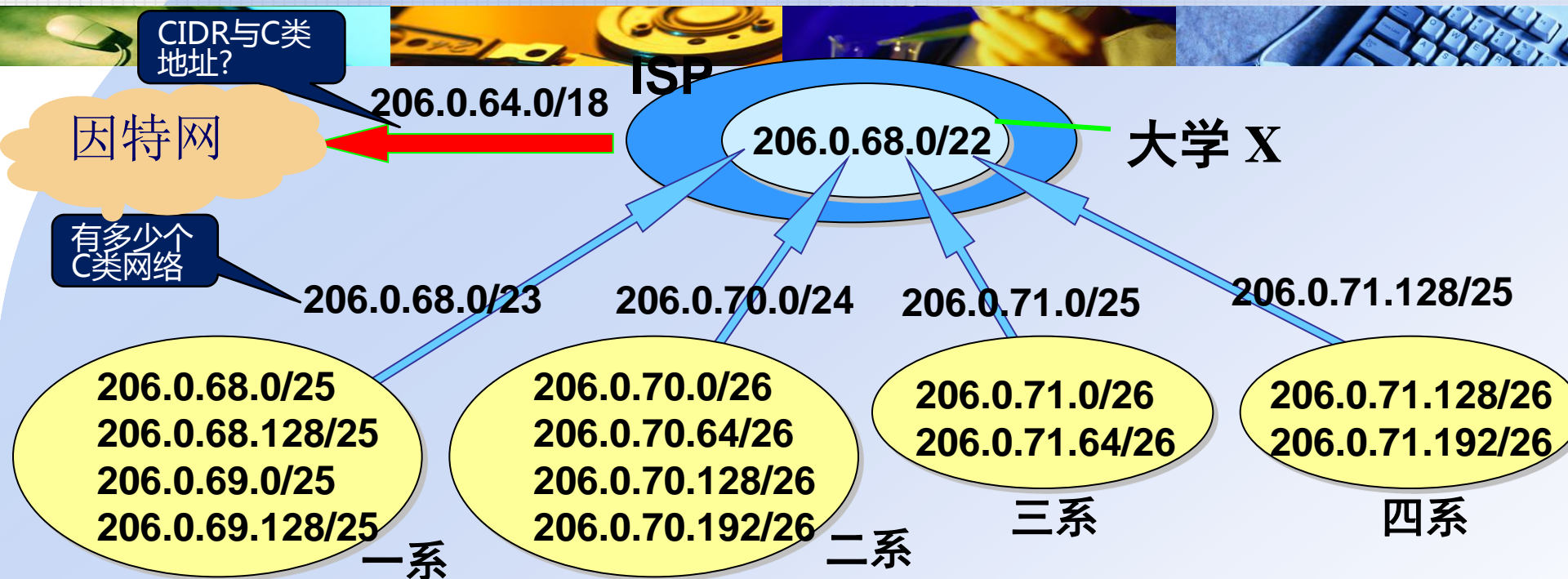
Bits now change in the fourth bit position.

CIDR 地址块中出现地址孔洞的例子

- ❖ 当下游网络接收到多个冲突的地址时使用最长地址匹配原则，路由器将选择与数据包的目标地址最好的匹配



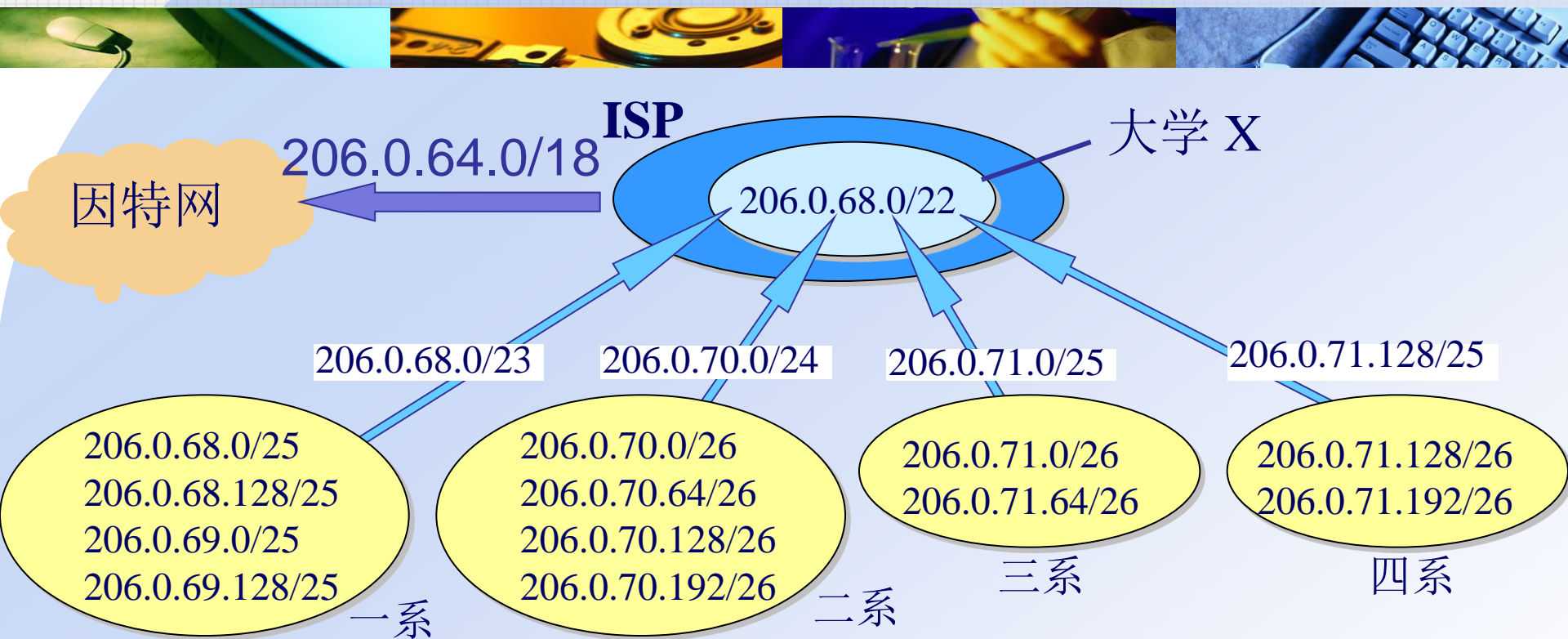
CIDR 地址块划分举例



| 单位 | 地址块 | 二进制表示 | 地址数 |
|-----|-----------------|-------------------------------|-------|
| ISP | 206.0.64.0/18 | 11001110.00000000.01* | 16384 |
| | 206.0.68.0/22 | 11001110.00000000.010001* | 1024 |
| | 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 |

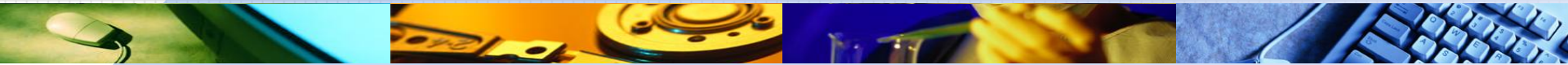
这是网络地址吗?

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。

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。

最长前缀匹配举例

假设收到的分组的目的地址 $D = 206.0.71.130$

路由表中的项目：206.0.68.0/22 (大学)

206.0.71.128/25 (四系)

查找路由表中的第 1 个项目，第 1 个项目 206.0.68.0/22 的掩码 M 有 22 个连续的 1。

$M = 11111111\ 11111111\ 11111100\ 00000000$

| | | | | | |
|-----|-----|----------|----------|-----------|----------|
| | M = | 11111111 | 11111111 | 11111100 | 00000000 |
| AND | D = | 206. | 0. | 01000111. | 10000010 |
| | | 206. | 0. | 01000100. | 0 |

与 206.0.68.0/22 匹配

最长前缀匹配举例

收到的分组的地址 $D = 206.0.71.130$

路由表中的项目: $206.0.68.0/22$ (大学)

$206.0.71.128/25$ (四系)

查找路由表中的第 2 个项目, 第 2 个项目 $206.0.71.128/25$ 的掩码 M 有 25 个连续的 1。

$M = 11111111\ 11111111\ 11111111\ 10000000$

因此只需把 D 的第 4 个字节转换成二进制。

| | | | | |
|-----------|----------|----------|----------|----------|
| $M =$ | 11111111 | 11111111 | 11111111 | 10000000 |
| AND $D =$ | 206. | 0. | 71. | 10000010 |
| | 206. | 0. | 71. | 10000000 |

与 $206.0.71.128/25$ 匹配

最长前缀匹配举例

$D \text{ AND } (11111111 \ 11111111 \ 11111100 \ 00000000)$

$= 206.0.68.0/22$ 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。

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。

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
01010110 00000000 00000000 00000000
01100001 00000000 00000000 00000000
10110000 00000010 00000000 00000000
10111011 00001010 00000000 00000000

0100
0101
011
10110
10111

