

COMP 3721数据通信导论

10b - 第10周 - 第2部分

学习成果

- 到本讲结束时，您将能够
 - 解释什么是 IPv4 地址。
 - 解释有类和无类的寻址方案。
 - 解释基于目的地的转发如何工作。
 - 解释 DHCP 的作用。
 - 解释什么是 NAT。

IPv4 地址

- An **IPv4** (Internet Protocol version 4) address is a **32-bit address** that **uniquely** and **universally** defines the connection of a **host** or a **router** to the Internet.
 - **唯一性**：每一个地址都定义了与互联网的一个且仅有一个连接。
 - **普适性**：任何想要连接到互联网的主机都必须接受该寻址系统。
- 该**IP 地址** 是 **连接** 的地址，而不是主机或路由器的地址。
 - 如果设备移至另一个网络，IP 地址可能会更改
 - 示例：如果一台设备通过两个网络有两个 Internet 连接，那么它就有两个 IPv4 地址

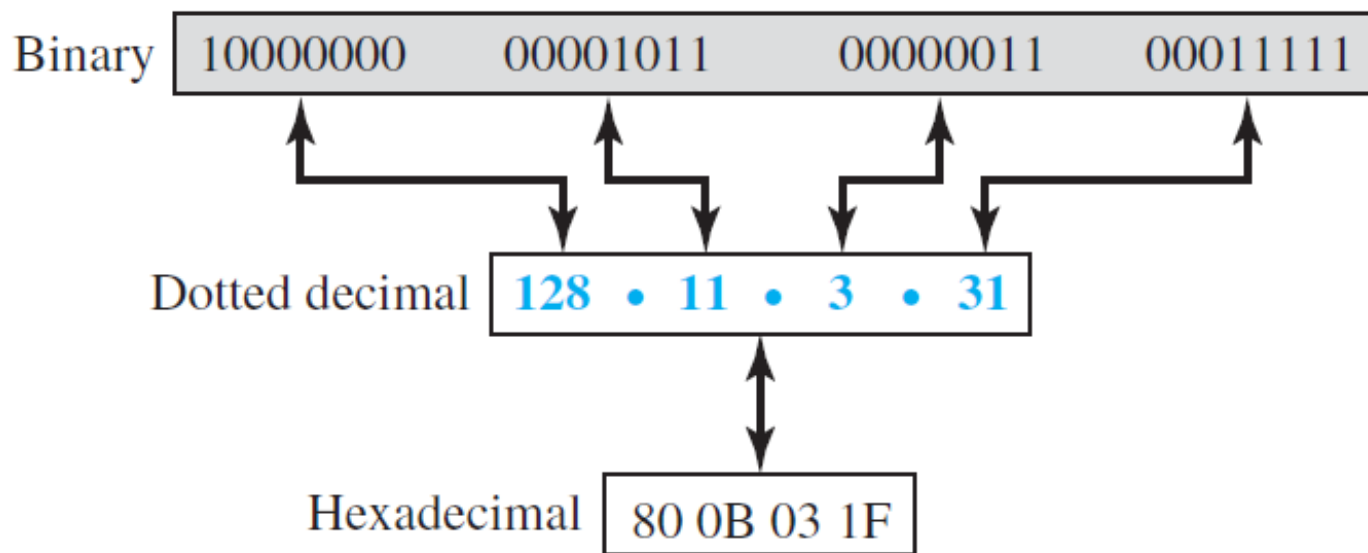
IPv4 地址空间

- 地址空间

- 协议使用的地址总数。
 - 如果一个协议使用 b 位来定义地址，地址空间为 2^b 。
- IPv4 使用 32 位地址，这意味着地址空间为： $2^{32} = 4,294,967,296$

三种常见表示 IPv4 的记法地址

1. 二进制表示法
2. 点分十进制表示法 → 更紧凑且更易阅读
3. 十六进制表示法 → 在网络编程中使用



地址层级- 两个类比



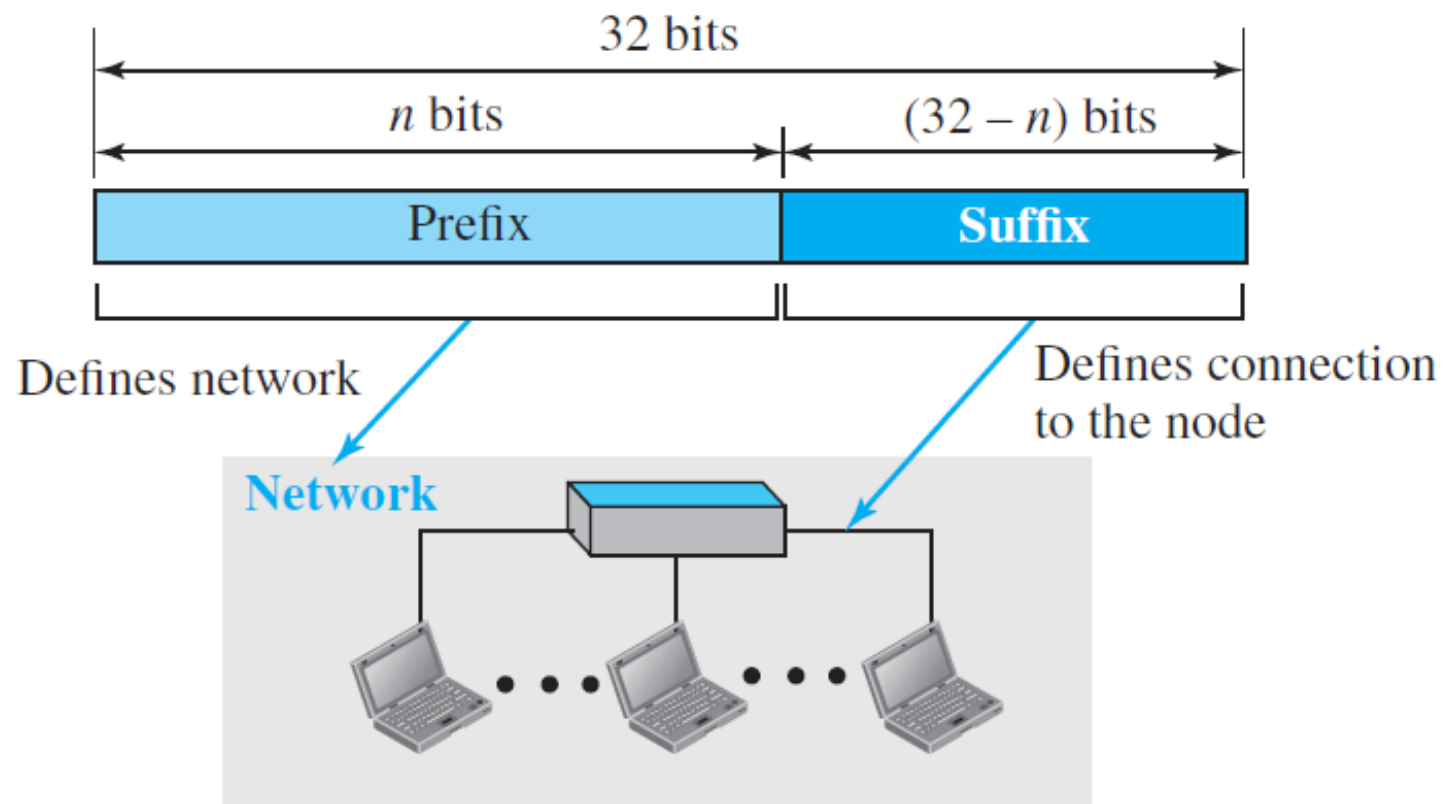
- 邮政网络
 - 邮寄地址（邮政地址）
包含国家、省/州、城市、街道、
门牌号以及收件人姓名



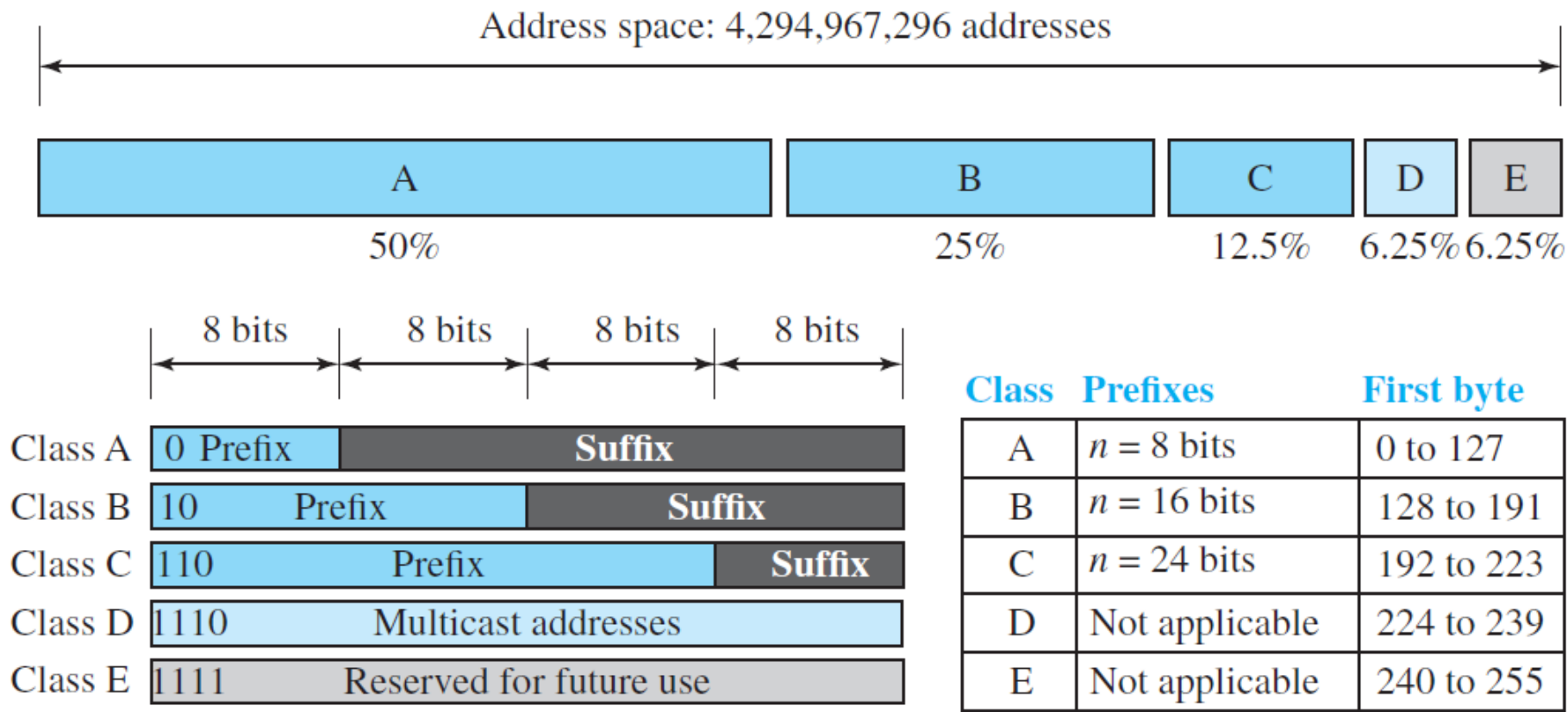
- 电话网络
 - 电话号码可分为国家代码、区号、
本地交换号和分机/接入号

地址中的层级结构

- 一个 32 位 IPv4 地址具有层次结构，分为两部分：



类式寻址



- 该方案现在已过时。
 - 由于地址耗尽（由类中地址分配不当导致）。

无类寻址

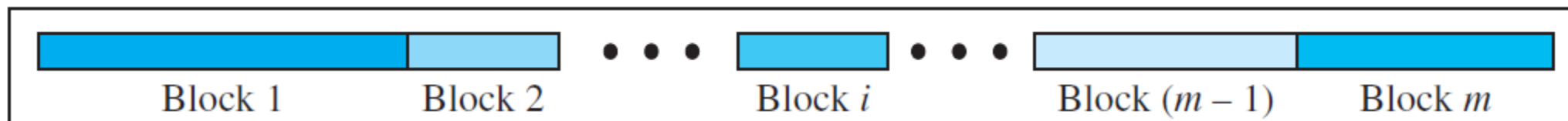
- 随着互联网的发展，显然需要更大的地址空间（地址枯竭/短缺问题）。
- 短期解决方案：
 - 无类寻址（1996年引入）
 - 继续使用相同的 IPv4 地址空间
 - 改变地址分配方式，以为每个组织提供公平份额。
 - NAT（网络地址转换）
- 长期解决方案：IPv6
 - 使用更大的地址空间。
 - 需要增加IP地址的长度，这意味着IP数据包的格式需要更改。
- $2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456$

无类寻址-动机

- IPv4 地址耗尽。
- ISP（互联网服务提供商）的出现。
 - ISP 被授予一大段地址范围，然后将其细分为地址并将它们分配给客户。

无类寻址

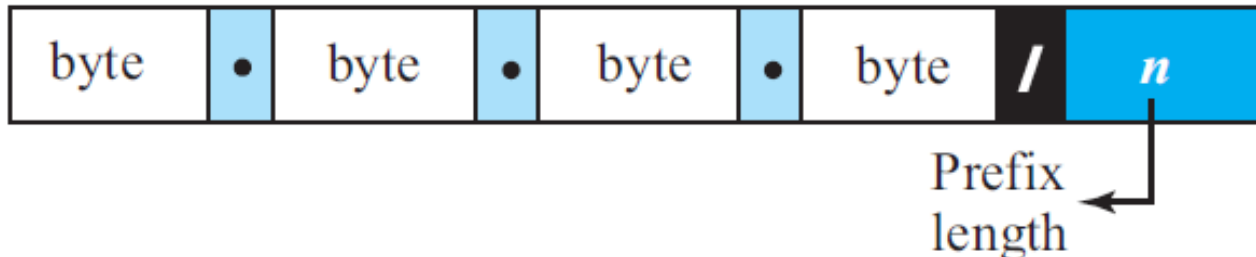
- 整个地址空间被划分为 **可变长度的块**。
- 地址中的 **前缀** 定义了 **块（网络）**；**后缀** 定义了 **节点（设备）**。
- 与有类寻址不同，前缀的长度是 **可变的**（0 – 32）。
- **限制**: 块内的地址数量必须是一个 **2 的幂**（我们可以有 2_0 、 2_1 、 2_2 、...、 2_{32} 个地址的块）。
- 网络的大小与前缀长度成反比。
 - 较小的前缀意味着更大的网络；较大的前缀意味着更小的网络。
- 一个组织可以被分配一块地址。



Address space

Classless Interdomain Routing (CIDR)

- 前缀长度（也称为掩码）， n ，添加到地址后，用斜杠分隔。
- 这种表示法称为CIDR（发音为 cider）或斜杠表示法。



Examples:

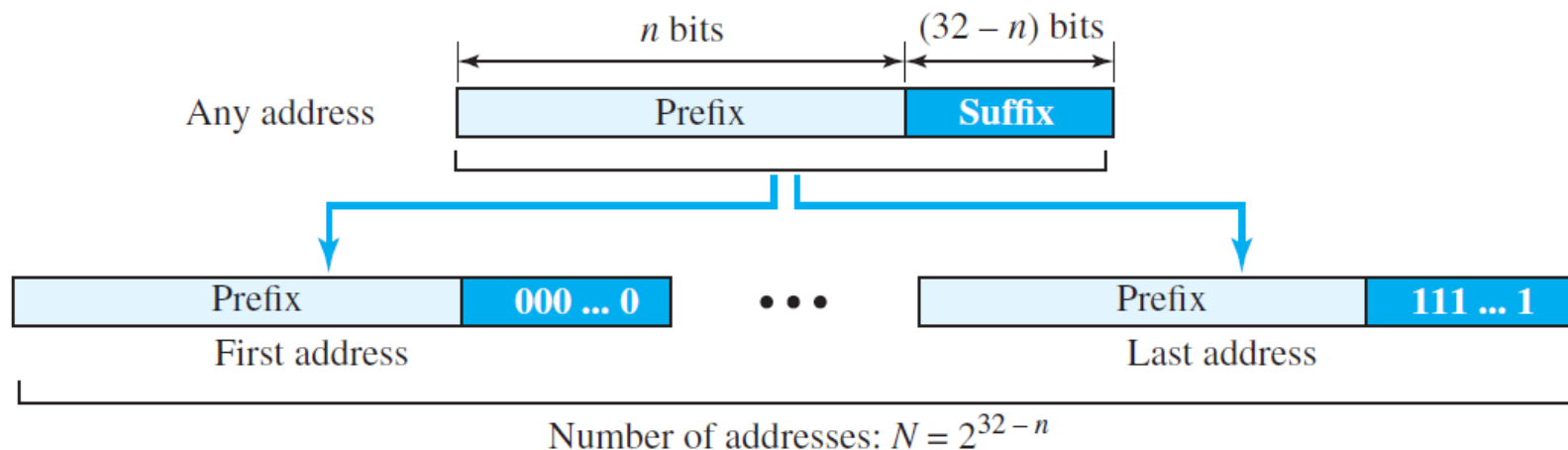
12.24.76.8/8

23.14.67.92/12

220.8.24.255/25

从地址中提取信息

- 给定区块中的任意地址，应提取三条信息。
 1. 地址的数量 ($N = 2^{32-n}$)。
 2. 区块中的第一个地址（保留 n 个最左边的位 并将 $(32-n)$ 个最右边的位全部置为 0）。
 3. 区块中的最后一个地址（保留 n 个最左边的位 并将 $(32-n)$ 个最右边的位全部置为 1）。



网络地址

- 该区块中的第一个地址称为网络地址。
- 网络地址很重要，因其被用于将数据包路由到其目的地。

无类寻址 – 示例 1

- 一个无类地址表示为 167.199.170.82/27。求出地址以及第一个和最后一个地址。

无类寻址 – 示例 1

- 一个无类地址表示为 167.199.170.82/27。求地址数以及首尾地址。

1. 网络中的**地址数**为 $2^{32-n} = 2^{32-27} = 2^5 = \mathbf{32 \text{ 个地址}}$ 。
2. 可通过**将前 27 位保留**来找到**首个地址**（27个最左侧位）并将其**余位改为 0**。

Address: 167.199.170.82/27	10100111	11000111	10101010	01010010
First address: 167.199.170.64/27	10100111	11000111	10101010	01000000

3. 最后一个地址可通过 b 找到 y **保留前 27 位** 并将其**余位改为 1**。

Last address: 167.199.170.95/27	10100111	11000111	10101010	01011111
---------------------------------	----------	----------	----------	----------

无类寻址 – 示例 2

- 在无类寻址中，单个地址本身不能定义该地址所属的块。
- 例如，地址 230.8.24.56 可以属于多个地址块。下面显示了其中一些块以及与该块关联的前缀值。

Prefix length:16	→	Block:	230.8.0.0	to	230.8.255.255
Prefix length:20	→	Block:	230.8.16.0	to	230.8.31.255
Prefix length:26	→	Block:	230.8.24.0	to	230.8.24.63
Prefix length:27	→	Block:	230.8.24.32	to	230.8.24.63
Prefix length:29	→	Block:	230.8.24.56	to	230.8.24.63
Prefix length:31	→	Block:	230.8.24.56	to	230.8.24.57

无类地址划分 – 示例 2

- 230.8.24.56/20

Prefix length:16	→	Block:	230.8.0.0	to	230.8.255.255
Prefix length:20	→	Block:	230.8.16.0	to	230.8.31.255
Prefix length:26	→	Block:	230.8.24.0	to	230.8.24.63
Prefix length:27	→	Block:	230.8.24.32	to	230.8.24.63
Prefix length:29	→	Block:	230.8.24.56	to	230.8.24.63
Prefix length:31	→	Block:	230.8.24.56	to	230.8.24.57

- 每个十进制数表示 8 位。为保留前 20 位，固定前 2 十进制（16 位），然后再固定 4 位。
- 24 = 00011000 → 230.8.00010000.0 → 230.8.16.0
- 00011111 = 31 → 230.8.00011111.255 → 230.8.31.255

地址掩码（子网掩码）

- 地址掩码是一个 32 位数字，其中最左边的 n 位设置为 1，其余位（ $32-n$ ）设置为 0。
- 例如，167.199.170.82/**27** 的子网掩码是
 - **/27** \rightarrow **11111111.11111111.11111111.111**00000 = 255.255.255.224

块分配

- 块分配由一个**全球权威机构**负责，称为**互联网名称与数字地址分配机构 (ICANN)**。
- ICANN **将一大块地址分配给 ISP**（或在本情形下被视为 ISP 的更大组织）。

块分配

- 为了 CIDR 的正常运行，**对分配的块施加** 两项限制：1. **所请求的**

(如果不是 2 的幂，则会考虑比请求数更大的、为 2 的幂的数字)if ，例
如： 1000 → 请

1024
求的地址 已授予

- **原因**： $N = 2^{32-n}$ 或 $n = 32 - \log_2 N$ (如果 N 不是 2 的幂，我们无法得到一个整数的 n)

2. **第一个地址需要可被该块中地址的数量整除。**

- **原因**：第一个地址需要是前缀，后面跟着 $(32-n)$ 个 0。第一个地址的十进制值为

$$\text{first address} = (\text{prefix in decimal}) \times 2^{32-n} = (\text{prefix in decimal}) \times N.$$

地址块分配示例

- 某 ISP 请求了一个包含 1000 个地址的地址块。说明将分配多少地址会被分配给该 ISP，并验证该分配。

块分配示例

- 一位 ISP 请求了一个包含 1000 个地址的地址块。展示将授予该 ISP 的地址数量并验证该分配。

- **答案：**

- 由于 1000 不是 2 的幂，**1024** 个地址被授予。前缀长度计算为 $n = 32 - \log_2 1024 = 22$ 。
- 一个可用的地址块，**18.14.12.0/22**，被分配给该 ISP。
- 可以看到第一个地址的十进制值为 302,910,464，能被 1024 整除。
- $00010010\ 00001110\ 00001100\ 00000000 = 302,910,464$
- $302,910,464 \bmod 1024 = 0$

子网划分

- 子网划分是**创建子网络的过程**。
- 创建更多层次的层级结构
- 被授予一段地址范围的组织（或互联网服务提供商）可以将该范围划分为多个子范围，并将每个子范围分配给一个子网络（或子网）。
 - 一个子网络可以被划分为多个子子网络。
 - 一个子子网络可以被划分为多个子子子网络，依此类推。

设计子网

- 我们的假设
 - N : 分配给组织的地址总数
 - n : 前缀长度
 - N_{sub} : 分配给每个子网的地址数量
 - n_{sub} : 每个子网的前缀长度
- 为保证子网的正常运行:
 1. 每个子网的地址数量应为 2 的幂。
 2. 每个子网的前缀长度应使用下列方法确定
公式: $n_{\text{sub}} = 32 - \log_2 N_{\text{sub}}$
 3. 每个子网的起始地址应能被该子网的地址数量整除。若我们先分配地址给较大的子网, 则可以实现这一点。

设计子网

- 网络中的子网络应被精心设计，以便能够对数据包进行路由。
- 在设计子网络之后，可以使用我们在查找 Internet 中每个网络信息时所描述的流程来确定每个子网络的信息，例如首地址和末地址。

设计子网示例

- 一个组织被授予了起始地址块 **14.24.74.0/24**。该组织需要有 **3 个子块** 用于其三个子网：一个子块包含 10 个地址，一个子块包含 **60 个地址**，以及一个子块包含 **120 个地址**。请设计这些子块。

设计子网示例

- 有 $32-24 = 256$ 个地址在此块中。
 - 第一个地址是 14.24.74.0/24;
 - 最后一个地址是 14.24.74.255/24.
- 为了满足子网划分要求，我们将地址分配给子块，从最大的开始，最后到最小的：

设计子网示例

a) 最大子块所需的地址数为**120 addresses**，这不是2的幂。我们分配了**128 addresses**。此子网的子网掩码可表示为 $n_1 = 32 - \log_2 128 = 25$ 。

- 该块的第一个地址是 **14.24.74.0/25**。
- 最后一个地址是 **14.24.74.127/25**。

设计子网示例

b) 第二大子块中所需的地址数 **60 个地址** 也不是 2 的幂。我们分配 **64 个地址**。该子网的子网掩码可按以下方式确定

$$n_2 = 32 - \log_2 64 = 26。$$

- 该块中的第一个地址为 **14.24.74.128/26**。
- 最后一个地址是 **14.24.74.191/26**。

子网设计示例

- c) 第二大子块中的地址数量，也就是需要 **10 个地址**，也不是 2 的幂。我们分配了 **16 个地址**。
- 该子网的子网掩码可按如下方式确定 $n_2 = 32 - \log_2 16 = 28$ 。
- 此块中的第一个地址是 **14.24.74.192/28**。
 - 最后一个地址是 **14.24.74.207/28**。
-
- 如果将之前子块中的所有地址相加，结果为 208 个地址，这意味着剩余 48 个地址。该范围内的第一个地址是 14.24.74.208。最后一个地址是 14.24.74.255。我们目前还不知道前缀长度。

特殊地址

- 用于特殊用途的五个特殊地址：

1. **本主机地址**：该网段中唯一的地址是 **0.0.0.0/32**

- 当主机需要发送 IP 数据报但不知道自己应使用哪个地址作为源地址时使用。

2. **广播地址**：该网段中唯一的地址是 **255.255.255.255/32**

- 当路由器或主机需要向网络中的所有设备发送数据报时使用。

- 然而，网络中的路由器会将具有此地址作为目标的分组阻止，该分组无法离开网络。

3. **回环地址**：该地址块 **127.0.0.0/8**

- 以该地址块中的某个地址作为目标地址的分组永远不会离开主机；它将留在主机内。

- 用于在本机测试软件、运行一些服务等。

特殊地址

4. 私有地址

- 有四个区段被分配为私有地址： **10.0.0.0**, **192.168.0.0**, **172.16.0.0/12**, **169.254.0.0/16...**和

5. 组播地址

- 区段 **224.0.0.0/4** 保留用于组播地址

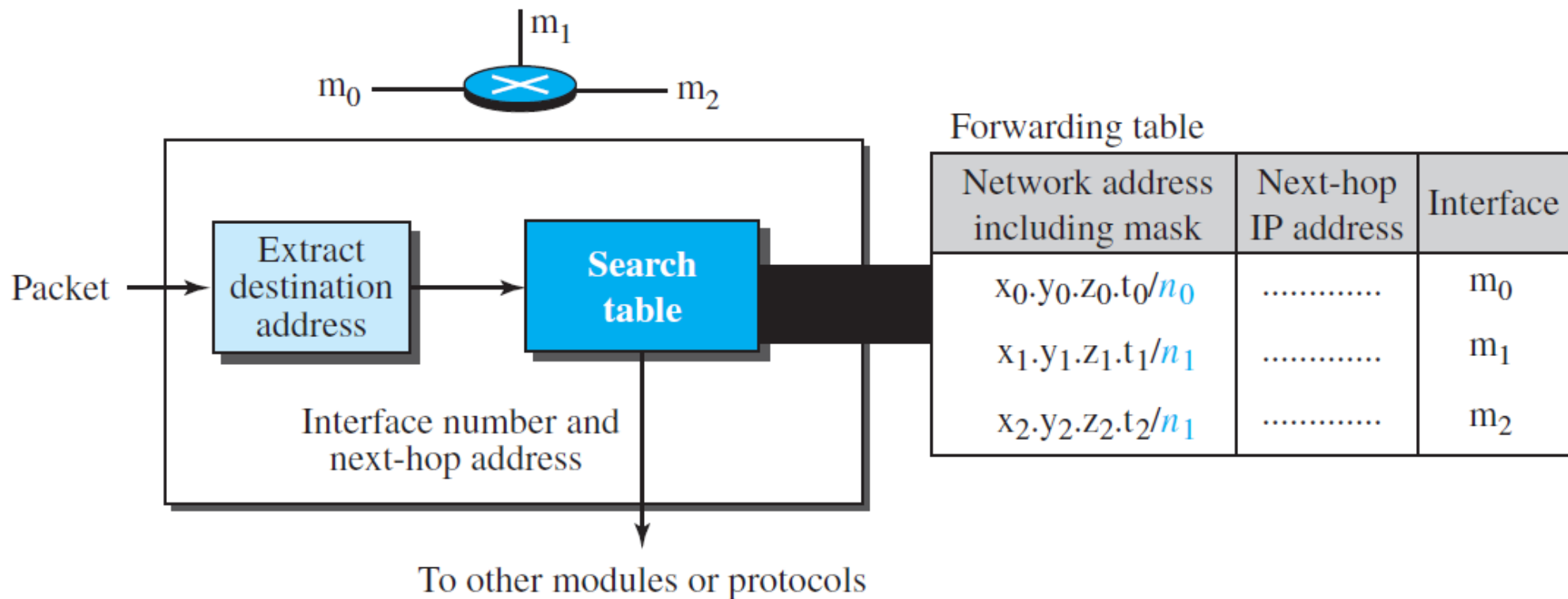
IP 数据包的转发

- 今天的互联网由若干链路（网络）的组合构成。
- **转发**：将数据包交付到下一跳（可以是最终目的地或中间的连接设备）。
- IP 协议最初被设计为**无连接**协议。
- **转发**基于**IP 数据报的目的地址**。
 - 传统方法且目前仍普遍使用

基于目的地址的转发

- 基于目的地的转发要求主机或路由器具有一个**转发表(路由)**。
- 在无类地址编址中，会为每个涉及的地址块在转发表中增加一行信息。
 - 四项信息：**掩码**，**网络地址**，**接口编号**，和**IP 下一跳路由器的地址**。
 - 前两项通常合并在一起。
- 转发表需要基于**搜索**，基于**网络地址**(即地址块中的第一个地址)。

基于目的地址的转发

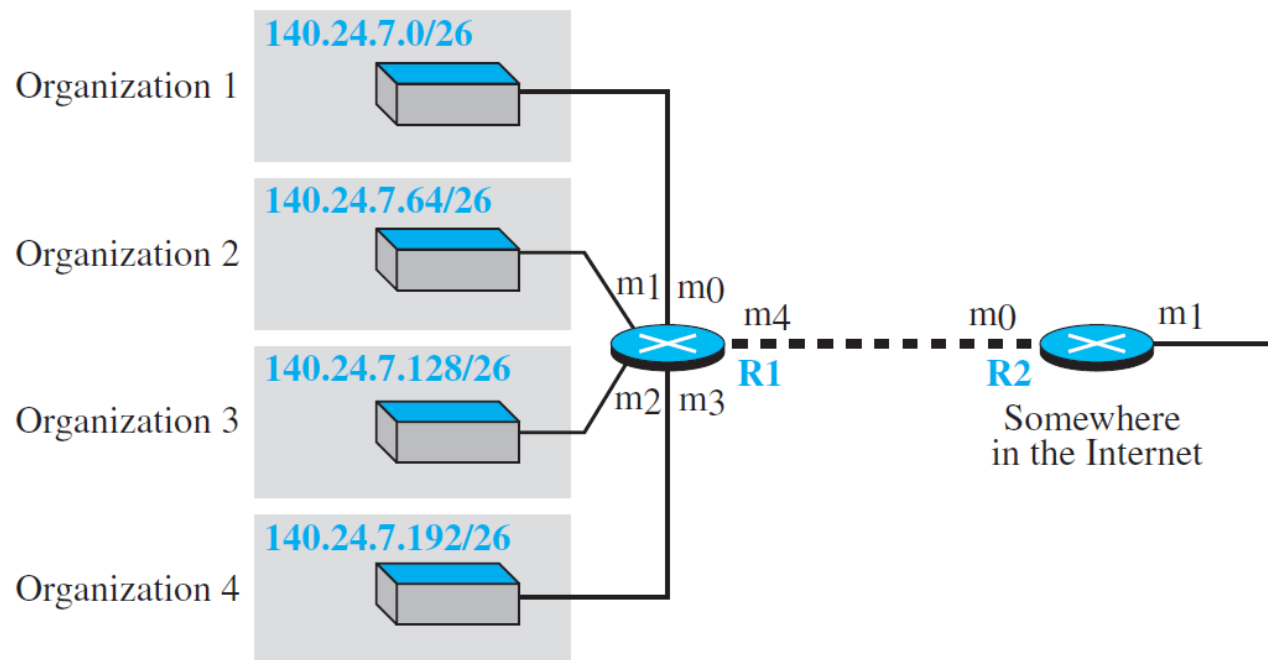


- 搜索：路由器逐行查看。保留目的地址的最左侧 n 位（前缀），其余位（后缀）设为0。如果得到的地址（网络地址）与第一列中的地址匹配，则提取接下来两列的信息；否则继续搜索。

地址聚合 (地址汇总/路由汇总)

- 当转发表中的条目数量增加时 → ，查找表的时间也会增加
- **地址聚合** → 以缓解上述问题
 - 当地址块被 **合并** 以创建更大的地址块时，**路由** 可以基于**前缀**来对该**更大地址块**进行
 - 这是CIDR策略的 **优点** 之一。
- ICANN 将一大块地址分配给一个 ISP
 - 每个 ISP 依次将其分配到的地址块划分为更小的子块，并将这些子块分配给其客户。

地址聚合示例



Forwarding table for R1

Network address/mask	Next-hop address	Interface
140.24.7.0/26	-----	m0
140.24.7.64/26	-----	m1
140.24.7.128/26	-----	m2
140.24.7.192/26	-----	m3
0.0.0.0/0	address of R2	m4

Forwarding table for R2

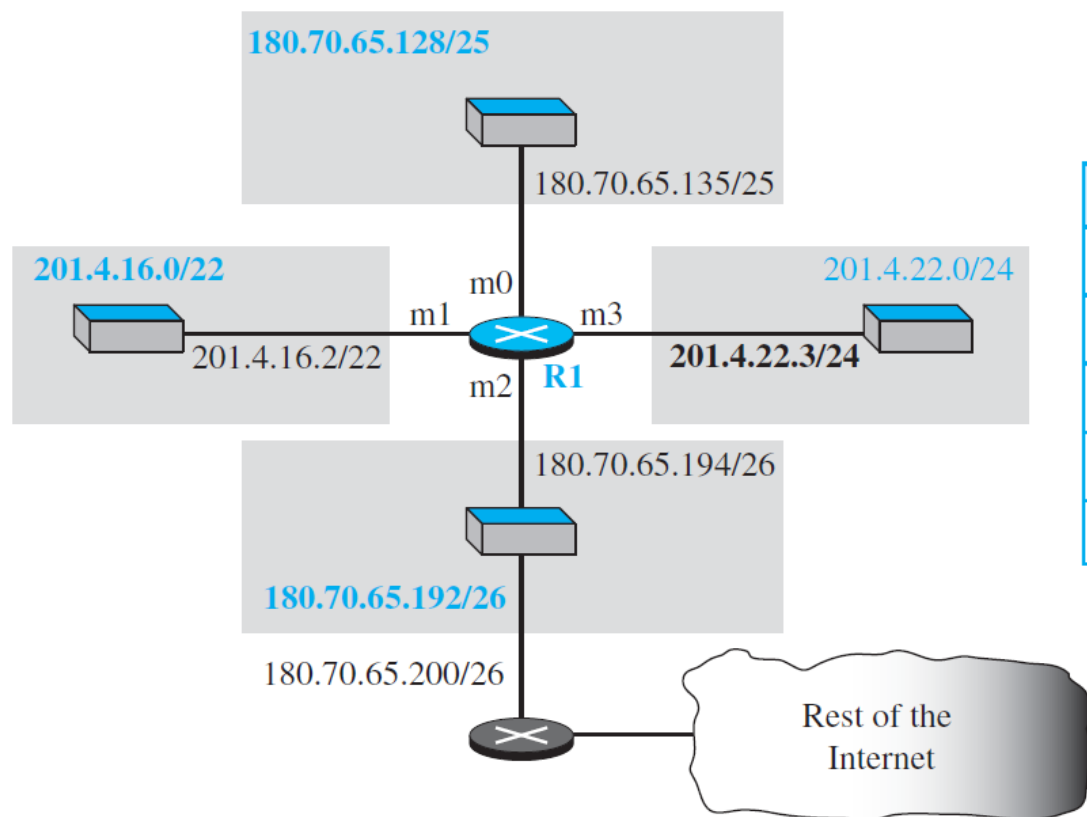
Network address/mask	Next-hop address	Interface
140.24.7.0/24	-----	m0
0.0.0.0/0	default router	m1

最长掩码匹配

- 最长掩码匹配
 - 在无类寻址中的 **路由原则**。
 - 指出转发表是从**最长掩码**到**最短掩码**排序的。
- 当路由器接收到IP数据包时，会将**目的IP**地址**逐位**与**路由表中的前缀**进行比较。具有**最多匹配位**的前缀是路由器将使用的前缀。

最长掩码匹配 – 示例

- 使用下图中的配置为路由器 R1 制作转发表。只填写网络地址/掩码和接口两列。

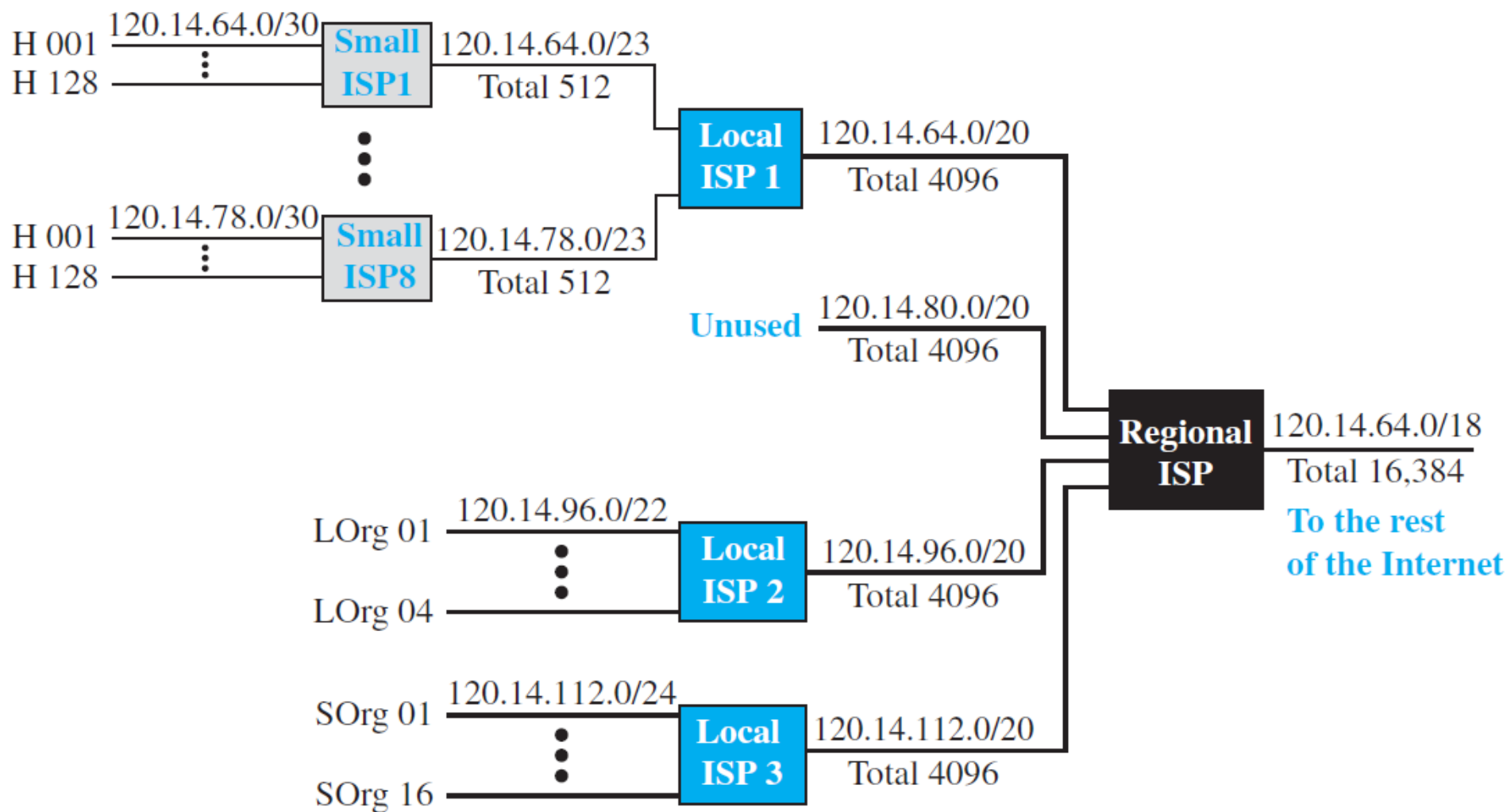


Network address/mask	Next hop	Interface
180.70.65.192/ 26	—	m2
180.70.65.128/ 25	—	m0
201.4.22.0/ 24	—	m3
201.4.16.0/ 22	—	m1
Default	180.70.65.200	m2

分层路由

- 为了解决巨大转发表的问题，我们可以在转发表中创建分层结构：
- 互联网
 - 骨干运营商
 - 国家级运营商
 - 区域性运营商
 - 本地 ISP
- 只要遵循寻址规则，我们在**无级层次**的**无类寻址**中可以有无限层级。

分层路由- 示例



动态主机配置协议（DHCP）

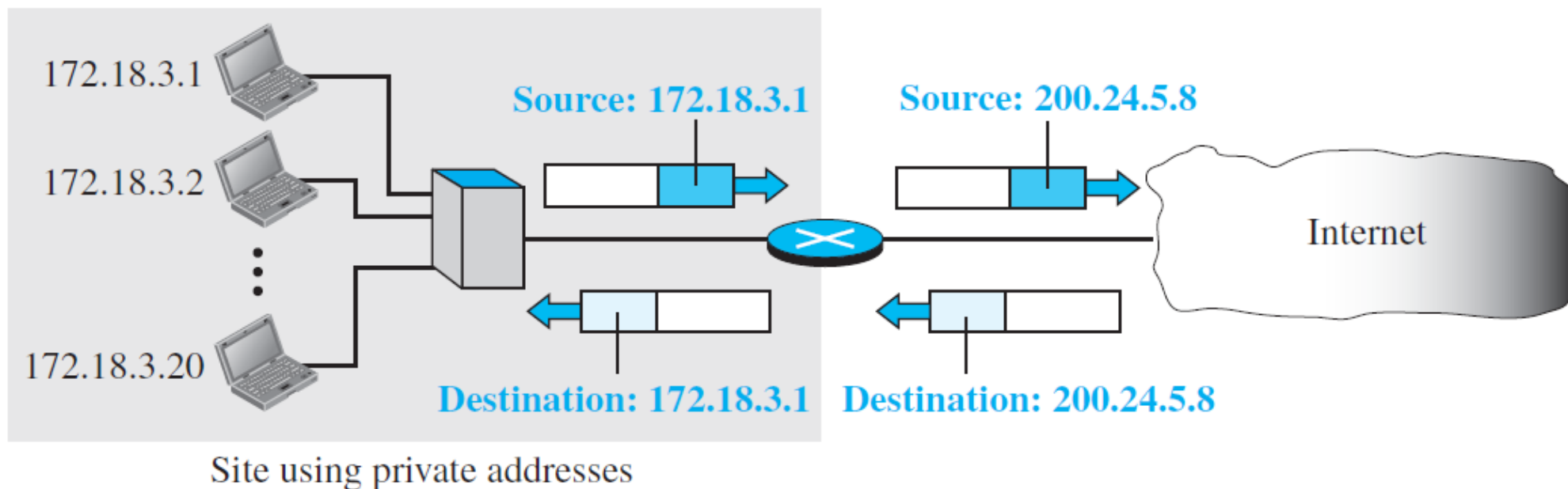
- 一个**应用层程序（协议）**，采用客户端-服务器范式，实际在网络层上为TCP/IP提供帮助。
- 当一段**地址块**分配给一个组织后，IP地址必须分配给**各个主机或路由器**。
 - **手动分配**：由网络管理员完成。
 - **自动/动态分配**：使用DHCP（由网络管理员配置）。
- DHCP可以配置为分配
 - **永久**IP 地址分配给主机和路由器。
 - **临时**，按需分配给主机的IP 地址（例如，旅客在酒店连接其笔记本电脑上网）。

网络地址转换 (NAT)

- 在大多数情况下，只有**小型网络中的一部分计算机**需要同时访问互联网。
 - 已分配地址的数量**不必与**网络中**计算机的数量相匹配**。
 - 公司通常使用 **私有 IP 地址** 来进行 **内部通信**。
 - 一组 **通用的** (公网/全局) **IP 地址** 由 ISP 分配用于 **通用通信**。
- **NAT** 是一项在 **私有** 和 **共有 IP 地址**之间提供映射的技术。

NAT 示例

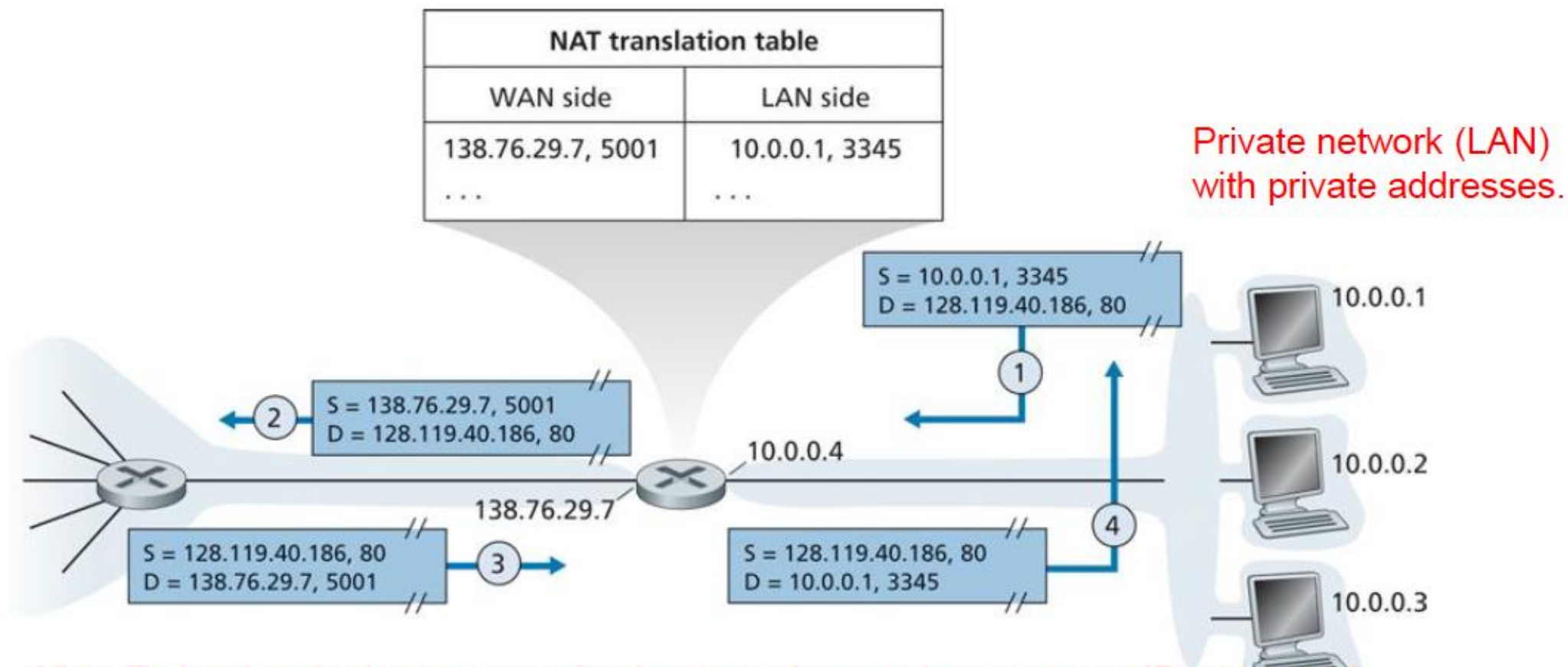
- 所有出站数据包都会通过 NAT 路由器，路由器将源地址替换为全局 NAT 地址。所有入站数据包也会通过 NAT 路由器，路由器将目的地址(数据包中的 NAT 路由器全局地址)替换为相应的私有地址。



NAT 转换表

- 将出站数据包的 **源地址** 翻译是很直接的。
- 来自互联网的数据包，其 **目的地址** 如何被 NAT 路由器识别？
 - NAT 路由器有一个 **转换表**。

NAT 翻译表



All traffic leaving the home router for the larger Internet has a source IP address of 138.76.29.7, and all traffic entering the home router must have a destination address of 138.76.29.7. In essence, the **NAT-enabled router** is hiding the details of the home network from the outside world.

NAT 的优点

- 只需 **一个** ISP 提供的 IP 地址即可供所有设备使用。
- 本地网络中主机的 **地址** 可以在 **不通知外部世界的情况下 更改**。
- 可以选择 **不同的 ISP**（以接收服务），而无需更改本地网络中设备的地址。
- **安全性**：本地网络内的设备无法被外部直接寻址或可见。
- NAT 在家庭和机构网络以及 4G/5G 蜂窝网络中被广泛使用。

概要

- 网络层提供的主要服务是对数据包进行分片/封装和将数据包从源头路由到目的地。
- IPv4 中的寻址
 - 两种地址分配机制：有类寻址和无类寻址。
- 基于目的地的路由由路由器执行。
- DHCP 用于为主机和路由器动态分配 IP 地址。
- 通过 NAT 暂时缓解 IPv4 地址短缺问题。

参考文献

[1] Behrouz A.Forouzan, 《Data Communications & Networking with TCP/IP Protocol Suite》, 第6版, 2022年, McGraw-Hill 公司。

阅读

- 教科书第7章，第7.4.1节。
- 教科书第7章，第7.8节（练习测试）