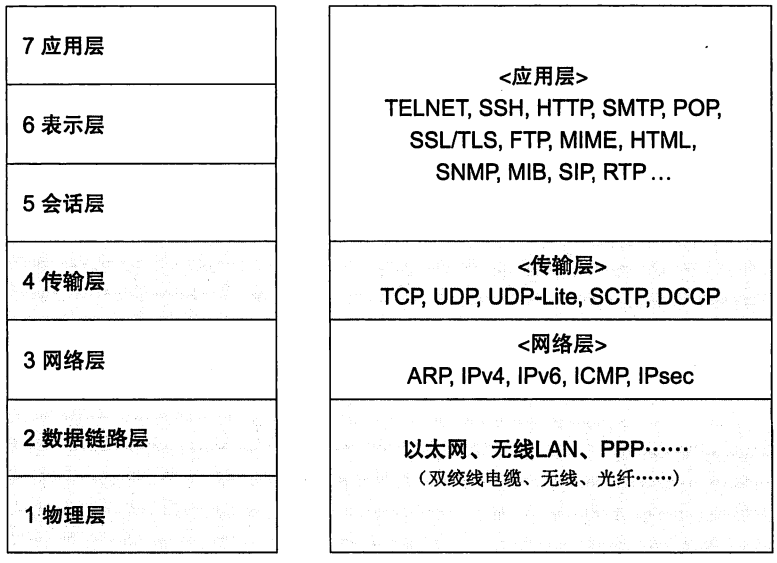
# 第4章

# IP协议

本章我们来学习IP（Internet Protocol，网际协议）。IP作为整个TCP/IP中至关重要的协议，主要负责将数据包发送给最终的目标计算机。因此，IP能够让世界上任何两台计算机之间进行通信。本章旨在详细介绍IP协议的主要功能及其规范。



# 4.1 IP 即网际协议

TCP/IP 的心脏是互联网层。这一层主要由 IP（Internet Protocol）和 ICMP（Internet Control Message Protocol）两个协议组成。本章仅对 IP 协议进行详细说明。关于 DNS、ARP、ICMP 等 IP 相关的其他协议将在第 5 章做详细介绍。

此外，鉴于目前的 IP 已无法应对互联网的需求，于是出现了更高版本的 IP 协议（称作 IPv6）。本章将按照 IPv4、IPv6 的顺序逐一介绍。

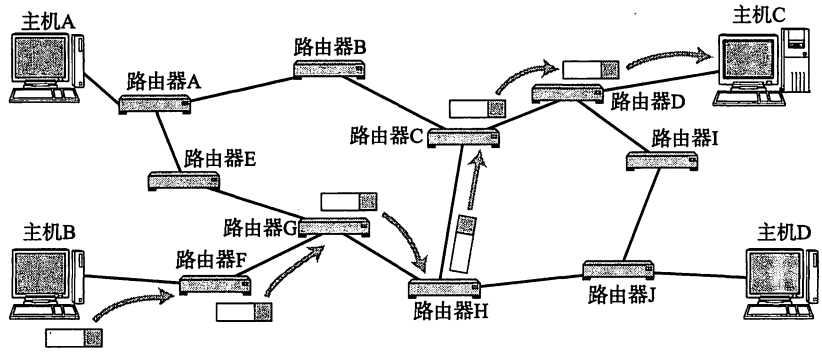
## 4.1.1 IP 相当于 OSI 参考模型的第 3 层

IP（IPv4、IPv6）相当于 OSI 参考模型中的第 3 层——网络层。

网络层的主要作用是“实现终端节点之间的通信”。这种终端节点之间的通信也叫“点对点（end-to-end）通信”。

从前面的章节可知，网络层的下一层——数据链路层的主要作用是在互连同一种数据链路的节点之间进行包传递。而一旦跨越多种数据链路，就需要借助网络层。网络层可以跨越不同的数据链路，即使是在不同的数据链路上也能实现两端节点之间的数据包传输。

图 4.1
IP 的作用




IP的主要作用就是在复杂的网络环境中将数据包发给最终的目标地址。

**主机与节点**

在互联网世界中，将那些配有 IP 地址的设备叫做“主机”。这里的主机如同在 1.1 节中所介绍的那样，可以是超大型计算机，也可以是小型计算机。这是因为互联网在当初刚发明的时候，只能连接这类大型的设备，因此习惯上就将配有 IP 地址的设备称为“主机”。

然而，准确地说，主机的定义应该是指“配置有 IP 地址，但是不进行路由控制▼的设备”。既配有 IP 地址又具有路由控制能力的设备叫做“路由器”，跟主机有所区别。而节点则是主机和路由器的统称▼。

▼路由控制
英文叫做 Routing。是指中转分组数据包。更多细节请参考 4.2.2 节和第 7 章。

▼这些都是 IPv6 的规范 RFC2460 中所使用的名词术语。在 IPv4 的规范 RFC791 中，将具有路由控制功能的设备叫做“网关”，然而现在都普遍叫做路由器（或 3 层交换机）。

## 4.1.2 网络层与数据链路层的关系

数据链路层提供直连两个设备之间的通信功能。与之相比，作为网络层的 IP

则负责在没有直连的两个网络之间进行通信传输。那么为什么一定需要这样的两个层次呢？它们之间的区别又是什么呢？

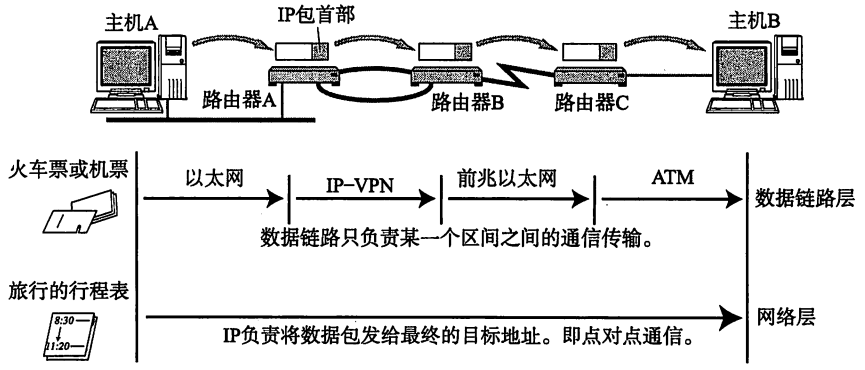
在此，我们以旅行为例说明这个问题。有个人要去一个很远的地方旅行，并且计划先后乘坐飞机、火车、公交车到达目的地。为此，他决定先去旅行社购买机票和火车票。

旅行社不仅为他预订好了旅途过程中所需要的机票和火车票，甚至为他制定了一个详细行程表，详细到几点几分需要乘坐飞机或火车都一目了然。

当然，机票和火车票只有特定区间▼内有效，当你换乘不同公司的飞机或火车时，还需要重新购票。

▼这里的“区间”与“段”(3.1节)同义。

图 4.2 IP 的作用与数据链路的作用




仔细分析一下机票和火车票，不难发现，每张票只能够在某一限定区间内移动。此处的“区间内”就如同通信网络上的数据链路。而这个区间内的出发地点和目的地点就如同某一个数据链路的源地址和目标地址等首部信息▼。整个全程的行程表的作用就相当于网络层。

▼出发地点好比源 MAC 地址，目标地点好比目的 MAC 地址。

如果我们只有行程表而没有车票，就无法搭乘交通工具到达目的地。反之，如果除了车票其他什么都没有，恐怕也很难到达目的地。因为你不知道该坐什么车，也不知道该在哪里换乘。因此，只有两者兼备，既有某个区间的车票又有整个旅行的行程表，才能保证到达目的地。与之类似，计算机网络中也需要数据链路层和网络层这个分层才能实现向最终目标地址的通信。

# 4.2 IP 基础知识

IP 大致分为三大作用模块，它们是 IP 寻址、路由（最终节点为止的转发）以及 IP 分包与组包。以下就这三个要点逐一介绍。

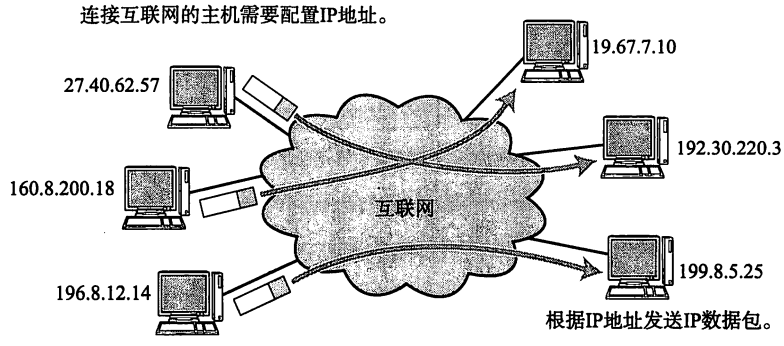
## 4.2.1 IP 地址属于网络层地址

在计算机通信中，为了识别通信对端，必须要有一个类似于地址的识别码进行标识。第 3 章中，我们介绍过数据链路的 MAC 地址。MAC 地址正是用来标识同一个链路中不同计算机的一种识别码。

作为网络层的 IP，也有这种地址信息。一般叫做 IP 地址。IP 地址用于在“连接到网络中的所有主机中识别出进行通信的目标地址”。因此，在 TCP/IP 通信中所有主机或路由器必须设定自己的 IP 地址▼。

▼严格来说，要针对每块网卡至少配置一个或一个以上的 IP 地址。

图 4.3
IP 地址




不论一台主机与哪种数据链路连接，其 IP 地址的形式都保持不变。以太网、无线局域网、PPP 等，都不会改变 IP 地址的形式▼。更多细节请参考 4.2.3 节。网络层对数据链路层的某些特性进行了抽象。数据链路的类型对 IP 地址形式透明，这本身就是其中抽象化中的一点。

▼数据链路的 MAC 地址的形式不一定必须一致。

另外，在网桥或交换集线器等物理层或数据链路层数据包转发设备中，不需要设置 IP 地址▼。因为这些设备只负责将 IP 包转化为 0、1 比特流转发或对数据链路帧的数据部分进行转发，而不需要应对 IP 协议▼。

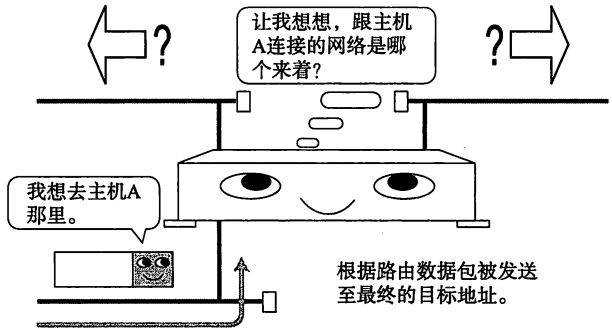
▼在用 SNMP 进行网路管理时有必要设置 IP 地址。不指定 IP 则无法利用 IP 进行网路管理。

▼反之，这些设备既可以在 IPv4 环境中使用，也可以在 IPv6 环境中使用。

## 4.2.2 路由控制

路由控制（Routing）是指将分组数据发送到最终目标地址的功能。即使网络非常复杂，也可以通过路由控制确定到达目标地址的通路。一旦这个路由控制的运行出现异常，分组数据极有可能“迷失”，无法到达目标地址。因此，一个数据包之所以能够成功地到达最终的目标地址，全靠路由控制。

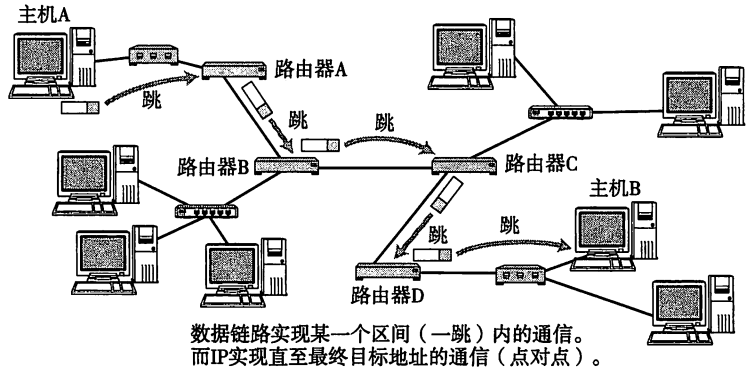
图 4.4
路由控制




## 发送数据至最终目标地址

Hop 译为中文叫“跳”。它是指网络中的一个区间。IP 包正是在网络中一个个跳间被转发。因此 IP 路由也叫做多跳路由。在每一个区间内决定着包在下一跳被转发的路径。

图 4.5
多跳路由




### 一跳的范围

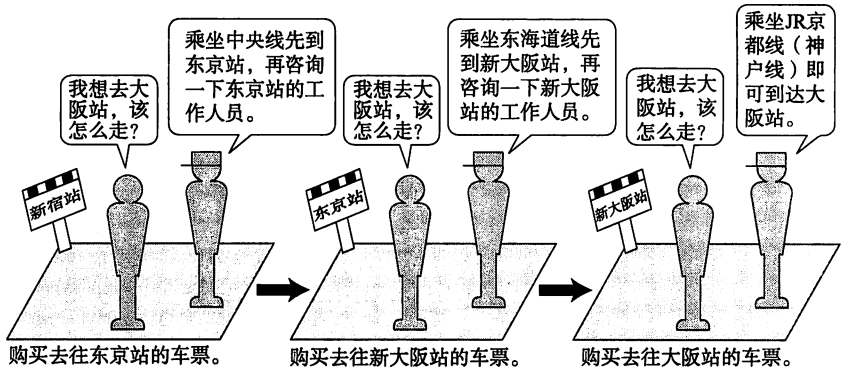
一跳（1 Hop）是指利用数据链路层以下分层的功能传输数据帧的一个区间。

以太网等数据链路中使用 MAC 地址传输数据帧。此时的一跳是指从源 MAC 地址到目标 MAC 地址之间传输帧的区间。也就是说它是主机或路由器网卡不经其他路由器而能直接到达的相邻主机或路由器网卡之间的一个区间。在一跳的这个区间内，电缆可以通过网桥或交换集线器相连，不会通过路由器或网关相连。

多跳路由是指路由器或主机在转发 IP 数据包时只指定下一个路由器或主机，而不是将到最终目标地址为止的所有通路全都指定出来。因为每一个区间（跳）在转发 IP 数据包时会分别指定下一跳的操作，直至包达到最终的目标地址。

如图 4.6，以乘坐火车旅游为例具体说明。

图4.6
每到一站再打听接下来该坐什么车




（IP就相当于在每次换乘时购买了标有下一个MAC地址的车票）

在前面的例子中，虽然已经确定了最终的目标车站，但是一开始还是不知道如何换乘才能到达这个终极目标地址。因此，工作人员给出的方法是首先去往最近的一个车站，再咨询这一车站的工作人员。而到了这个车站以后再询问工作人员如何才能达到最终的目标地址时，仍然得到同样的建议：乘坐某某线列车到某某车站以后再询问那里的工作人员。

于是，该乘客就按照每一个车站工作人员的指示，到达下一车站以后再继续询问车站的工作人员，得到类似的建议。

因此，即使乘客不知道其最终目的地的方向也没有关系。可以通过每到一个车站咨询工作人员的这种极其偶然▼的方法继续前进，也可以到达最终的目标地址。

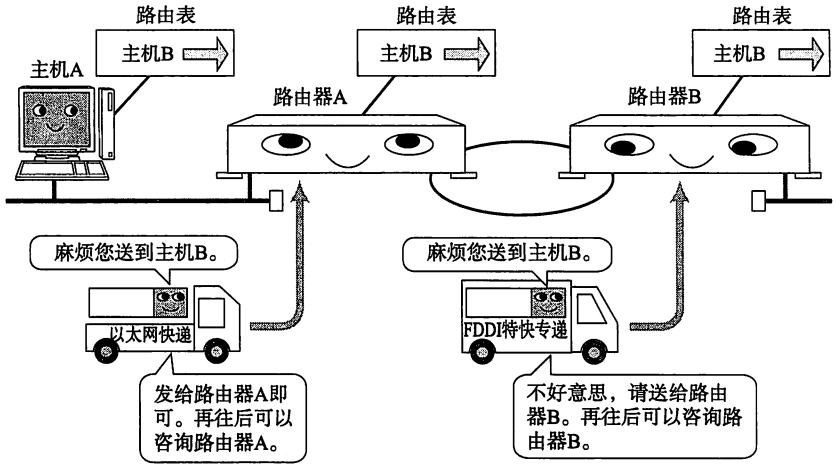
▼英文叫做“Ad Hoc”，是指具有偶然性的、在各跳之间无计划传输的意思。尤其在谈到IP时经常会用到该词。

IP数据包的传输亦是如此。可以将旅行者看做IP数据包，将车站和工作人员看做路由器。当某个IP包到达路由器时，路由器首先查找其目标地址▼，从而再决定下一步应该将这个包发往哪个路由器，然后将包发送过去。当这个IP包到达那个路由器以后，会再次经历查找下一目标地址的过程，并由该路由器转发给下一个被找到的路由器。这个过程可能会反复多次，直到找到最终的目标地址将数据包发送给这个节点。

▼IP包被转发到途中的某个路由器时，实际上是装入数据链路层的数据帧以后再被送出。以以太网为例，目标MAC地址就是下一个路由器的MAC地址。关于IP地址与MAC地址相关的细节请参考5.3.3节。

这里还可以用快递的送货方式来打比方。IP数据包犹如包裹，而送货车犹如数据链路。包裹不可能自己移动，必须有送货车承载转运。而一辆送货车只能将包裹送到某个区间范围内。每个不同区间的包裹将由对应的送货车承载、运输。IP的工作原理也是如此。

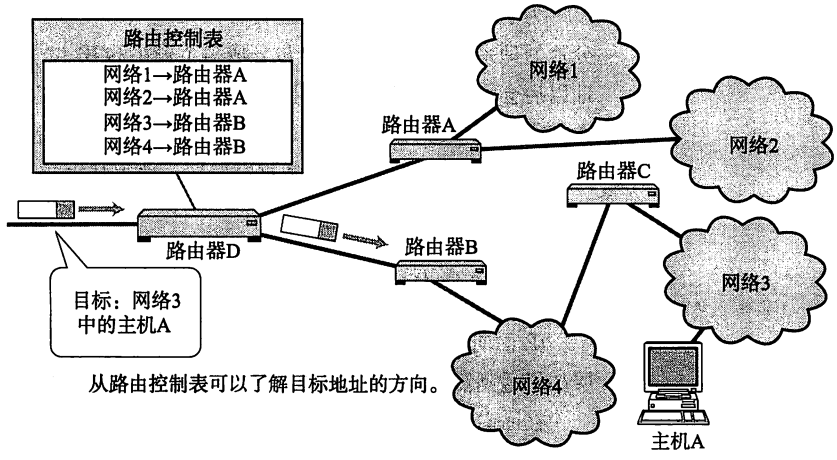
图 4.7
IP 包的发送




### 路由控制表

为了将数据包发给目标主机，所有主机都维护着一张路由控制表（Routing Table）。该表记录 IP 数据在下一步应该发给哪个路由器。IP 包将根据这个路由表在各个数据链路上传输。

图 4.8
路由控制表



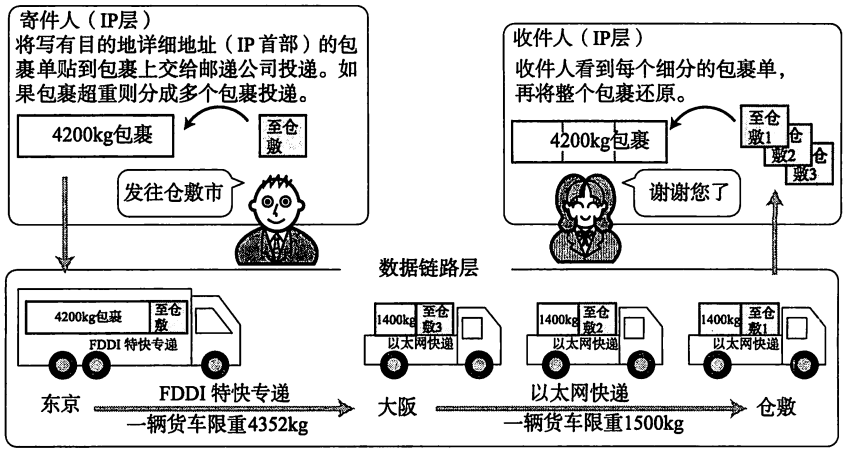

## 4.2.3 数据链路的抽象化

IP 是实现多个数据链路之间通信的协议。数据链路根据种类的不同各有特点。对这些不同数据链路的相异特性进行抽象化也是 IP 的重要作用之一。在 4.2.1 节也曾提到过，数据链路的地址可以被抽象化为 IP 地址。因此，对 IP 的上一层来说，不论底层数据链路使用以太网还是无线 LAN 亦或是 PPP，都将被一视同仁。

不同数据链路有个最大的区别，就是它们各自的最大传输单位（MTU：Maximum Transmission Unit）不同。就好像人们在邮寄包裹或行李时有各自的大小限制一样。

图 4.9 中展示了很多运输公司在运送包裹时所限定的包裹大小。

图4.9
不同数据链路的最大传输单位




到了大阪，将包裹卸车转为以太网快递邮送。由于以太网快递限重，就需要对原包裹进行拆分，并在每一个分包上贴上相应序号的包裹单。到了仓敷可以根据包裹单的序号再将整个包裹合并复原。

MTU 的值在以太网中是 1500 字节，在 FDDI 中是 4352 字节，而 ATM 则为 9180 字节▼。IP 的上一层可能会要求传送比这些 MTU 更多字节的数据，因此必须在线路上传送比包长还要小的 MTU。

▼关于 MTU 的更多取值，请参考 4.3 节。

为了解决这个问题，IP 进行分片处理（IP Fragmentation）。顾名思义，所谓分片处理是指，将较大的 IP 包分成多个较小的 IP 包▼。分片的包到了对端目标地址以后会再被组合起来传给上一层。即从 IP 的上次层看，它完全可以忽略数据包在途中的各个数据链路上的 MTU，而只需要按照源地址发送的长度接收数据包。IP 就是以这种方式抽象化了数据链路层，使得从上层更不容易看到底层网络构造的细节。

▼关于分片处理的更多细节，请参考 4.5 节。

## 4.2.4　IP 属于面向无连接型

IP 面向无连接。即在发包之前，不需要建立与对端目标地址之间的连接。上层如果遇到需要发送给 IP 的数据，该数据会立即被压缩成 IP 包发送出去。

在面向有连接的情况下，需要事先建立连接。如果对端主机关机或不存在，也就不可能建立连接。反之，一个没有建立连接的主机也不可能发送数据过来。

而面向无连接的情况则不同。即使对端主机关机或不存在，数据包还是会被发送出去。反之，对于一台主机来说，它会何时从哪里收到数据也是不得而知的。通常应该进行网络监控，让主机只接收发给自己的数据包。若没有做好准备很有可能会错过一些该收的包。因此，在面向无连接的方式下可能会有很多冗余的通信。

那么，为什么 IP 要采用面向无连接呢?

主要有两点原因：一是为了简化，二是为了提速。面向连接比起面向无连接处理相对复杂。甚至管理每个连接本身就是一个相当繁琐的事情。此外，每次通信之前都要事先建立连接，又会降低处理速度。需要有连接时，可以委托上一层提供此项服务。因此，IP 为了实现简单化与高速化采用面向无连接的方式。

**为了提高可靠性，上一层的 TCP 采用面向有连接型**

IP 提供尽力服务（Best Effort），意指“为了把数据包发送到最终目标地址，尽最大努力。”然而，它并不做“最终收到与否的验证”。IP 数据包在途中可能会发生丢包、错位以及数据量翻倍等问题。如果发送端的数据未能真正发送到对端目标主机会造成严重的问题。例如，发送一封电子邮件，如果邮件内容中很重要的一部分丢失，会让收件方无法及时获取信息。

因此提高通信的可靠性很重要。TCP 就提供这种功能。如果说 IP 只负责将数据发给目标主机，那么 TCP 则负责保证对端主机确实接收到数据。

那么，有人可能会提出疑问：为什么不让 IP 具有可靠传输的功能，从而把这两种协议合并到一起呢？

这其中的缘由就在于，如果要一种协议规定所有的功能和作用，那么该协议的具体实施和编程就会变得非常复杂，无法轻易实现。相比之下，按照网络分层，明确定义每层协议的作用和责任以后，针对每层具体的协议进行编程会更加有利于该协议的实现。

网络通信中如果能进行有效分层，就可以明确 TCP 与 IP 各自协议的最终目的，也有利于后续对这些协议进行扩展和性能上的优化。分层也简化了每个协议的具体实现。互联网能够发展到今天，与网络通信的分层密不可分。

# 4.3 IP地址的基础知识

在用TCP/IP通信时，用IP地址识别主机和路由器。为了保证正常通信，有必要为每个设备配置正确的IP地址。在互联网通信中，全世界都必须设定正确的IP地址。否则，根本无法实现正常的通信。

因此，IP地址就像是TCP/IP通信的一块基石。

## 4.3.1 IP地址的定义

IP地址（IPv4地址）由32位正整数来表示。TCP/IP通信要求将这样的IP地址分配给每一个参与通信的主机。IP地址在计算机内部以二进制▼方式被处理。然而，由于人类社会并不习惯于采用二进制方式，需要采用一种特殊的标记方式。那就是将32位的IP地址以每8位为一组，分成4组，每组以“.”隔开，再将每组数转换为十进制数▼。下面举例说明这一方法。

▼二进制是指用0、1表示数字的方法。

▼这种方法也叫做“十进制点符号”（Dot-decimal notation）。

| | | | | | |
|---|---|---|---|---|---|
| 例） | $2^8$ | $2^8$ | $2^8$ | $2^8$ | |
| | 10101100 | 00010100 | 00000001 | 00000001 | （2进制） |
| | 10101100. | 00010100. | 00000001. | 00000001 | （2进制） |
| | 172. | 20. | 1. | 1 | （10进制） |

将表示成IP地址的数字整体计算，会得出如下数值。

$$2^{32} = 4\ 294\ 967\ 296$$

从这个计算结果可知，最多可以允许43亿台计算机连接到网络▼。

▼虽然43亿这个数字听起来还算比较大，但是还不到地球上现有人口的总数。

实际上，IP地址并非是根据主机台数来配置的，而是每一台主机上的每一块网卡（NIC）都得设置IP地址▼。通常一块网卡只设置一个IP地址，其实一块网卡也可以配置多个IP地址。此外，一台路由器通常都会配置两个以上的网卡，因此可以设置两个以上的IP地址。

▼Windows或Unix中设置IP地址的命令分别为ipconfig/all和ifconfig-a。

因此，让43亿台计算机全部连网其实是不可能的。后面将要详细介绍IP地址的两个组成部分（网络标识和主机标识），了解了这两个组成部分后你会发现实际能够连接到网络的计算机个数更是少了很多▼。

▼根据一种可以更换IP地址的技术NAT，可连接计算机数超过43亿台。关于NAT的更多细节请参考5.6节。

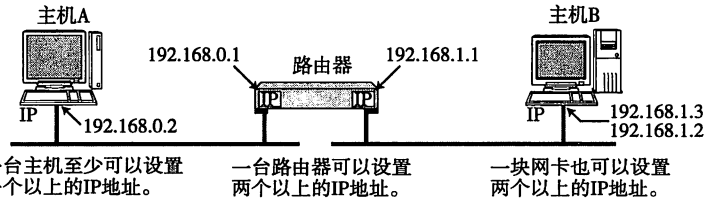



图4.10 每块网卡可以分配一个以上的IP地址

## 4.3.2 IP地址由网络和主机两部分标识组成

IP地址由“网络标识（网络地址）”和“主机标识（主机地址）”两部分组成▼。

▼192.168.128.10/24中的“/24”表示从第1位开始到多少位属于网络标识。在这个例子中，192.168.128之前的都是该IP的网络地址。更多细节请参考4.3.6节。

如图4.11所示，网络标识在数据链路的每个段配置不同的值。网络标识必须保证相互连接的每个段的地址不相重复。而相同段内相连的主机必须有相同的网

络地址。IP 地址的“主机标识”则不允许在同一个网段内重复出现。

由此，可以通过设置网络地址和主机地址，在相互连接的整个网络中保证每台主机的 IP 地址都不会相互重叠。即 IP 地址具有了唯一性▼。

▼唯一性是指在整个网络中，不会跟其他主机的 IP 地址冲突。关于唯一性的解释还可以参考 1.8.1 节。

如图 4.12 所示，IP 包被转发到途中某个路由器时，正是利用目标 IP 地址的网络标识进行路由。因为即使不看主机标识，只要一见到网络标识就能判断出是否为该网段内的主机。

那么，究竟从第几位开始到第几位算是网络标识，又从第几位开始到第几位算是主机标识呢？关于这点，有约定俗成的两种类型。最初二者以分类进行区别。而现在基本以子网掩码（网络前缀）区分。不过，请读者注意，在有些情况下依据部分功能、系统和协议的需求，前一种的方法依然存在。

图 4.11
IP 地址的主机标识

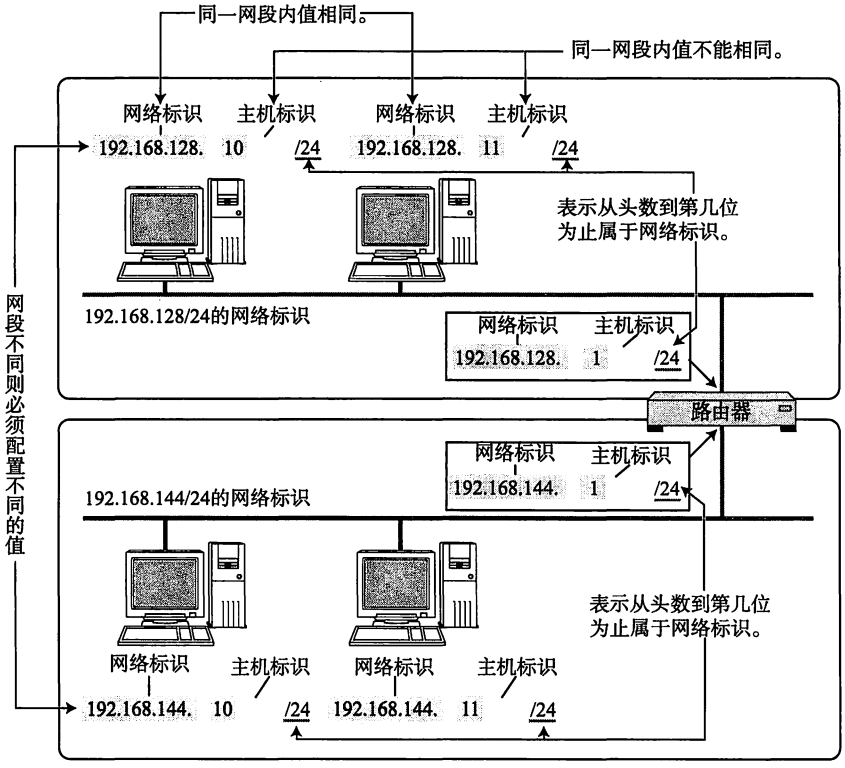
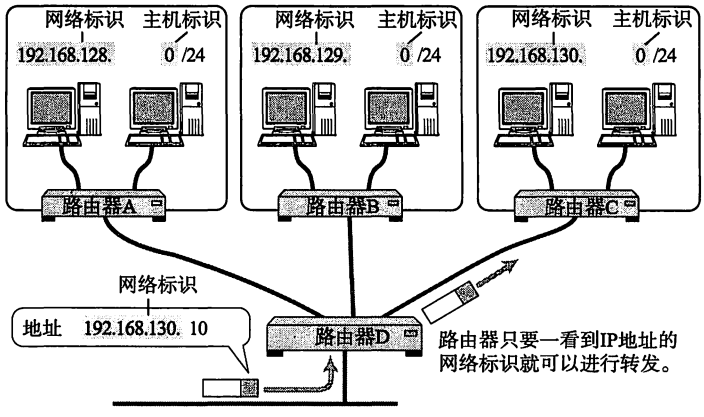



图 4.12
IP 地址的网络标识



## 4.3.3　IP地址的分类

▼还有一个一直未使用的E类。

IP地址分为四个级别，分别为A类、B类、C类、D类▼。它根据IP地址中从第1位到第4位的比特列对其网络标识和主机标识进行区分。

### A类地址

▼去掉分类位剩下7位。

A类IP地址是首位以“0”开头的地址。从第1位到第8位▼是它的网络标识。用十进制表示的话，0.0.0.0~127.0.0.0是A类的网络地址。A类地址的后24位相当于主机标识。因此，一个网段内可容纳的主机地址上限为16，777，214个▼。

▼关于A类地址总数的计算请参考附录2.1节。

### B类地址

▼去掉分类位剩下14位。

B类IP地址是前两位为“10”的地址。从第1位到第16位▼是它的网络标识。用十进制表示的话，128.0.0.1~191.255.0.0是B类的网络地址。B类地址的后16位相当于主机标识。因此，一个网段内可容纳的主机地址上限为65，534个▼。

▼关于B类地址总数的计算请参考附录2.2节。

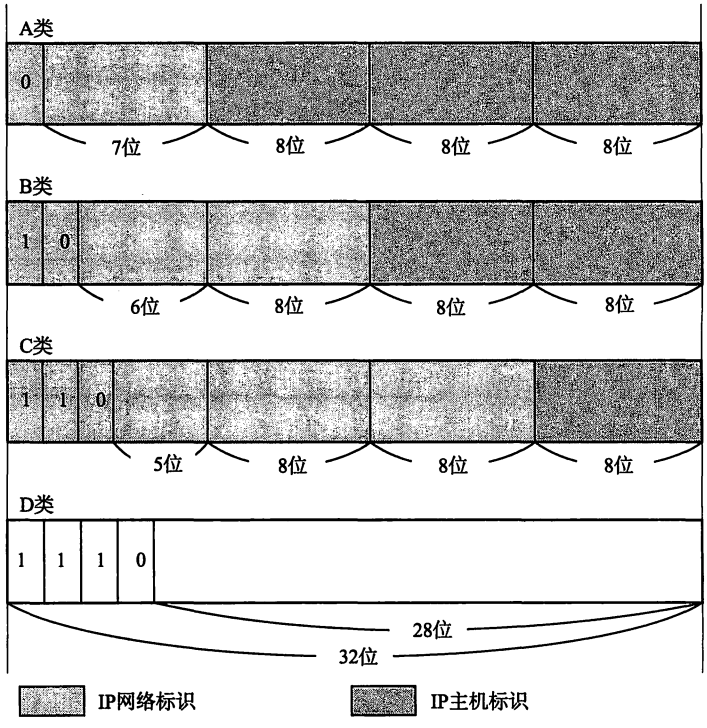



图4.13
IP地址的分类

### C 类地址

▼去掉分类位剩下 21 位。

C 类 IP 地址是前三位为“110”的地址。从第 1 位到第 24 位▼是它的网络标识。用十进制表示的话，192.168.0.0～239.255.255.0 是 C 类的网络地址。C 类地址的后 8 位相当于主机标识。因此，一个网段内可容纳的主机地址上限为 254 个▼。

▼关于 C 类地址总数的计算请参考附录 2.3 节。

### D 类地址

▼去掉分类位剩下 28 位。

D 类 IP 地址是前四位为“1110”的地址。从第 1 位到第 32 位▼是它的网络标识。用十进制表示的话，224.0.0.0～239.255.255.255 是 D 类的网络地址。D 类地址没有主机标识，常被用于多播。关于多播的更多细节请参考 4.3.5 节。

### 关于分配 IP 主机地址的注意事项

在分配 IP 地址时关于主机标识有一点需要注意。即要用比特位表示主机地址时，不可以全部为 0 或全部为 1。因为全部为只有 0 在表示对应的网络地址或 IP 地址不可获知的情况下才使用。而全部为 1 的主机地址通常作为广播地址。

因此，在分配过程中，应该去掉这两种情况。这也是为什么 C 类地址每个网段最多只能有 254（$2^8 - 2 = 254$）个主机地址的原因。

## 4.3.4 广播地址

广播地址用于在同一个链路中相互连接的主机之间发送数据。IP 地址中的主机地址部分全部设置为 1，就成为了广播地址▼。例如把 172.0.0/16 用二进制表示如下：

▼以太网中如果将 MAC 地址的所有位都改为 1，则形成 FF：FF：FF：FF：FF：FF 的广播地址。因此，广播的 IP 包以数据链路的帧的形式发送时，得通过 MAC 地址为全 1 比特的 FF：FF：FF：FF：FF：FF 转发。

10101100.00010100.00000000.00000000 （二进

将这个地址的主机部分全部改为 1，则形成广播地址：

10101100.00010100.11111111.11111111 （二进）

再将这个地址用十进制表示，则为 172.20.255.255。

### 两种广播

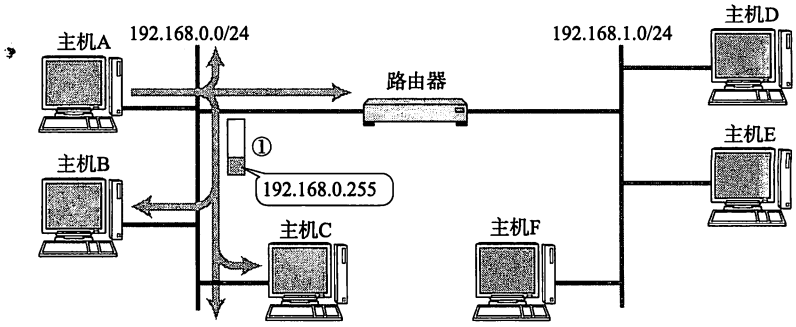
广播分为本地广播和直接广播两种。

在本网络内的广播叫做本地广播。例如网络地址为 192.168.0.0/24 的情况下，广播地址是 192.168.0.255。因为这个广播地址的 IP 包会被路由器屏蔽，所以不会到达 192.168.0.0/24 以外的其他链路上。

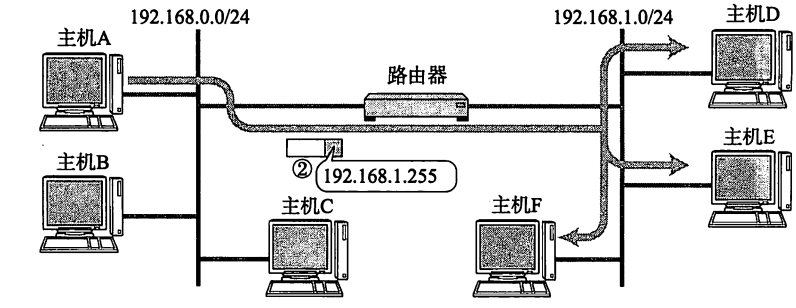
在不同网络之间的广播叫做直接广播。例如网络地址为 192.168.0.0/24 的主机向 192.168.1.255/24 的目标地址发送 IP 包。收到这个包的路由器，将数据转发给 192.168.1.0/24，从而使得所有 192.168.1.1～192.168.1.254 的主机都能收到这个包▼。

▼由于直接广播有一定的安全问题，多数情况下会在路由器上设置为不转发。

图4.14
本地广播与直接广播




① 的包不会到达192.168.1.0/24的网络。（本地广播）




② 是指向192.168.1.0/24的广播包。（直接广播）

## 4.3.5　IP多播

### 同时发送提高效率

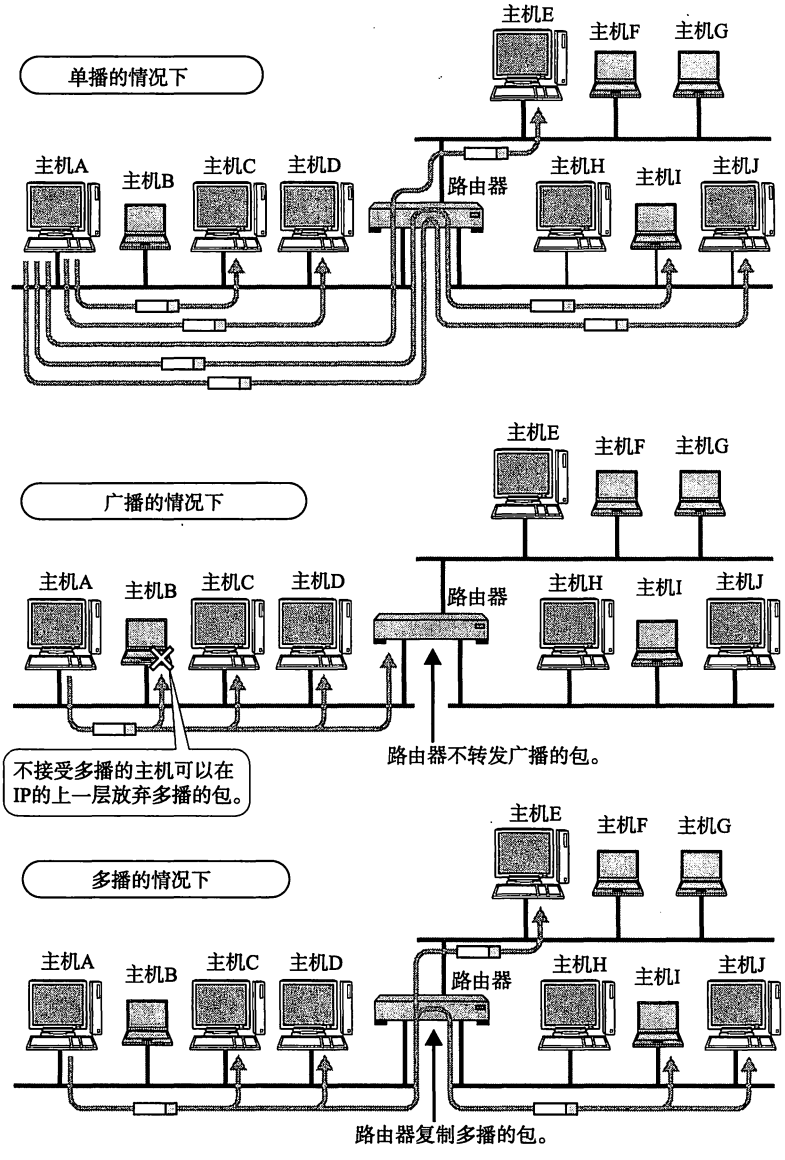
多播用于将包发送给特定组内的所有主机。由于其直接使用IP协议，因此也不存在可靠传输。

而随着多媒体应用的发展，对于向多台主机同时发送数据包，在效率上的要求也日益提高。在电视会议系统中对于1对N、N对N通信的需求明显上升。而具体实现上往往采用复制1对1通信的数据，将其同时发送给多个主机的方式。

在人们使用多播功能之前，一直采用广播的方式。那时广播将数据发给所有终端主机，再由这些主机IP之上的一层去判断是否有必要接收数据。是则接收，否则丢弃。

然而这种方式会给那些毫无关系的网络或主机带来影响，造成网络上很多不必要的流量。况且由于广播无法穿透路由，若想给其他网段发送同样的包，就不得不采取另一种机制。此，多播这种既可以穿透路由器，又可以实现只给那些必要的组发送数据包的，不就成为必选之路了。

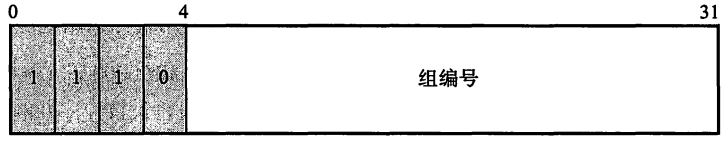
图 4.15
单播、广播、多播通信




## IP 多播与地址

多播使用 D 类地址。因此，如果从首位开始到第 4 位是“1110”，就可以认为是多播地址。而剩下的 28 位可以成为多播的组编号。

图 4.16
多播地址




从 224.0.0.0 到 239.255.255.255 都是多播地址的可用范围。其中从 224.0.0.0 到 224.0.0.255 的范围不需要路由控制，在同一个链路内也能实现多

播。而在这个范围之外设置多播地址会给全网所有组内成员发送多播的包▼。

▼可以利用生存时间（TTL，Time To Live）限制包的到达范围。

此外，对于多播，所有的主机（路由器以外的主机和终端主机）必须属于224.0.0.1的组，所有的路由器必须属于224.0.0.2的组。类似地，多播地址中有众多已知的地址，它们中具有代表性的部分已在表4.1中列出。

利用IP多播实现通信，除了地址外还需要IGMP▼等协议的支持。关于它的更多细节请参考5.8.1节。

▼ Internet Group Management Protocol

表4.1 既定已知的多播地址

| 地址 | 内　容 |
|---|---|
| 224.0.0.0 | （预定） |
| 224.0.0.1 | 子网内所有的系统 |
| 224.0.0.2 | 子网内所有的路由器 |
| 224.0.0.5 | OSPF路由器 |
| 224.0.0.6 | OSPF指定路由器 |
| 224.0.0.9 | RIP2路由器 |
| 224.0.0.10 | IGRP路由器 |
| 224.0.0.11 | Mobile-Agents |
| 224.0.0.12 | DHCP服务器/中继器代理 |
| 224.0.0.14 | RSVP-ENCAPSULATION |
| 224.0.1.1 | NTP Network Time Protocol |
| 224.0.1.8 | SUN NIS+ Information Service |
| 224.0.1.22 | Service Location（SVRLOC） |
| 224.0.1.33 | RSVP-encap-1 |
| 224.0.1.34 | RSVP-encap-2 |
| 224.0.1.35 | Directory Agent Discovery（SVRLOC-DA） |
| 224.0.2.2 | SUN RPC PMAPPROC CALLIT |

## 4.3.6　子网掩码

### 分类造成浪费？

一个IP地址只要确定了其分类，也就确定了它的网络标识和主机标识。例如A类地址前8位（除首位“0”还有7位）、B类地址前16位（除首位“10”还有14位）、C类地址前24位（除首位“110”还有21位）分别是它们各自的网络标识部分。

由此，按照每个分类所表示的网络标识的范围如下所示。

例）　A类　11111111.　00000000.　00000000.　00000000
　　B类　11111111.　11111111.　00000000.　00000000
　　C类　11111111.　11111111.　11111111.　00000000

用“1”表示IP网络地址的比特范围，用“0”表示IP主机地址范围。将它们以十进制表示，如下所示。其中“1”的部分是网络地址部分，“0”的部分是

主机地址部分。

例）

| | | | | |
|---|---|---|---|---|
| A 类 | 255. | 0. | 0. | 0 |
| B 类 | 255. | 255. | 0. | 0 |
| C 类 | 255. | 255. | 255. | 0 |

网络标识相同的计算机必须同属于同一个链路。例如，架构 B 类 IP 网络时，理论上一个链路内允许 6 万 5 千多台计算机连接。然而，在实际网络架构当中，一般不会有在同一个链路上连接 6 万 5 千多台计算机的情况。因此，这种网络结构实际上是不存在的。

因此，直接使用 A 类或 B 类地址，确实有些浪费。随着互联网的覆盖范围逐渐增大，网络地址会越来越不足以应对需求，直接使用 A 类、B 类、C 类地址就更加显得浪费资源。为此，人们已经开始一种新的组合方式以减少这种浪费。

### 子网与子网掩码

现在，一个 IP 地址的网络标识和主机标识已不再受限于该地址的类别，而是由一个叫做“子网掩码”的识别码通过子网网络地址细分出比 A 类、B 类、C 类更小粒度的网络。这种方式实际上就是将原来 A 类、B 类 C 类等分类中的主机地址部分用作子网地址，可以将原网络分为多个物理网络的一种机制。

自从引入了子网以后，一个 IP 地址就有了两种识别码。一是 IP 地址本身，另一个是表示网络部的子网掩码。子网掩码用二进制方式表示的话，也是一个 32 位的数字。它对应 IP 地址网络标识部分的位全部为“1”，对应 IP 地址主机标识的部分则全部为“0”。由此，一个 IP 地址可以不再受限于自己的类别，而是可以用这样的子网掩码自由地定位自己的网络标识长度。当然，子网掩码必须是 IP 地址的首位开始连续的“1”▼。

▼最初提出子网掩码时曾允许出现不连续的子网掩码，但现在基本不允许出现这种情况。

对于子网掩码，目前有两种表示方式。以 172. 20. 100. 52 的前 26 位是网络地址的情况为例，以下是其中一种表示方法，它将 IP 地址与子网掩码的地址分别用两行来表示。

| | | | | |
|---|---|---|---|---|
| IP 地址 | 172. | 20. | 100. | 52 |
| 子网掩码 | 255. | 255. | 255. | 192 |
| 网络地址 | 172. | 20. | 100. | 0 |
| 子网掩码 | 255. | 255. | 255. | 192 |
| 广播地址 | 172. | 20. | 100. | 63 |
| 子网掩码 | 255. | 255. | 255. | 192 |

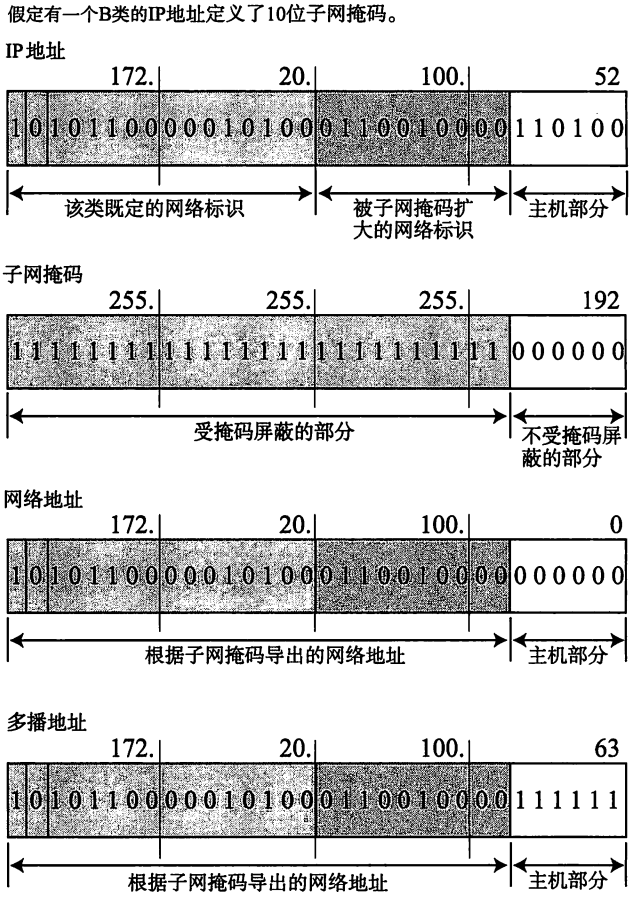
另一种表示方式如下所示。它在每个 IP 地址后面追加网络地址的位数▼用“/”隔开。

▼这种方式也叫“后缀”表示法。

| | | | | | |
|---|---|---|---|---|---|
| IP 地址 | 172. | 20. | 100. | 52 | /26 |
| 网络地址 | 172. | 20. | 100. | 0 | /26 |
| 广播地址 | 172. | 20. | 100. | 63 | /26 |

不难看出，在第二种方式下记述网络地址时可以省略后面的“0”。例如 172. 20. 0. 0/16 跟 172. 20/16 其实是一个意思。

图4.17
子网掩码可以灵活指定网络标识的长度




## 4.3.7　CIDR与VLSM

▼0、10、127等开头的A类地址都是具有特殊意义的保留地址。

直到20世纪90年代中期，向各种组织分配IP地址都以A类、B类、C类等分类为单位进行。对于架构大规模网络的组织，一般会分配一个A类地址。反之，在架构小规模网络时，则分配C类地址。然而A类地址的派发在全世界最多也无法超过128个▼，加上C类地址的主机标识最多只允许254台计算机相连，导致众多组织开始申请B类地址。其结果是B类地址也开始严重缺乏，无法满足需求。

▼即使申请了B类地址的组织，如果发现根本没必要选用B类标准长度作为网络地址，那么可以将原申请的地址返还，再重新申请一个长度合适的IP地址及其网络标识。

于是，人们开始放弃IP地址的分类▼，采用任意长度分割IP地址的网络标识和主机标识。这种方式叫做CIDR▼，意为“无类型域间选路”。由于BGP（Border Gateway Protocol，边界网关协议，参考7.6节）对应了CIDR，所以不受IP地址分类的限制自由分配▼。

▼Classless Inter-Domain Routing

▼迁移到CIDR的初期，由于A类和B类地址个数严重不足，常常把那些以2的幂次（4，8，16，32，……）划分的C类IP地址组合起来再进行分配。当时这种方式也叫做“超网”。

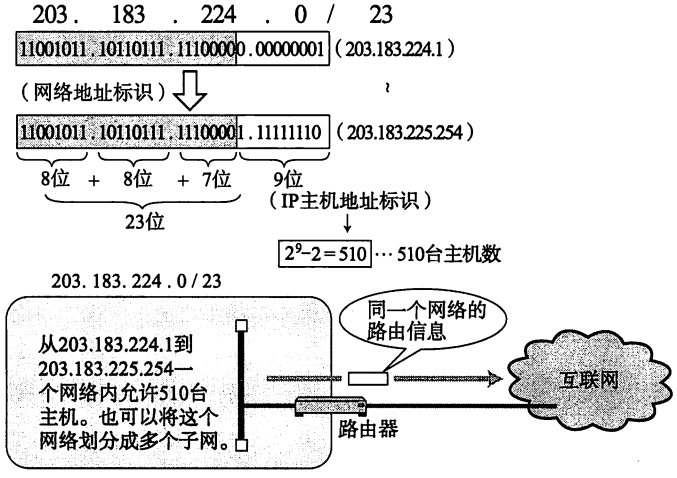
根据CIDR，连续多个C类地址▼就可以划分到一个较大的网络内。CIDR更有效地利用了当前IPv4地址，同时通过路由集中▼降低了路由器的负担。

▼CIDR汇总的C类地址以2的幂次（4，8，16，32，……）划分，因此必须有一个能够按位分割的边界。

例如，以图4.18为例，应用CIDR技术将203.183.224.1到203.183.225.254的地址合为同一个网络（它们本来是2个C类地址）。

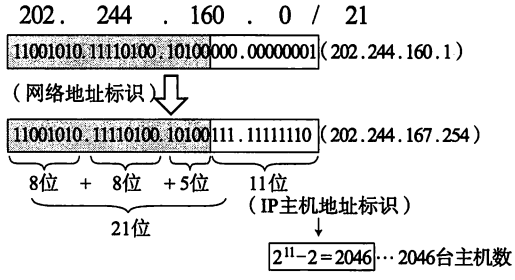
▼关于路由集合的更多细节请参考4.4.2节。

图 4.18
CIDR 应用举例（1）




类似地，图 4.19 展示了将 202.244.160.1 到 202.244.167.254 的地址合并为一个网络的情形。该例子中实际上是将 8 个 C 类地址合并为一个网络。

图 4.19
CIDR 应用举例（2）




在 CIDR 被应用到互联网的初期，网络内部采用固定长度的子网掩码机制。也就是说，当子网掩码的长度被设置为/25 以后，域内所有的子网掩码都得使用同样的长度。然而，有些部门可能有 500 台主机，另一些部门可能只有 50 台主机。如果全部采用统一标准，就难以架构一个高效的网络结构。为此人们提出组织内要使用可变长度的、高效的 IP 地址分配方式。

于是产生了一种可以随机修改组织内各个部门的子网掩码长度的机制——VLSM（可变长子网掩码）▼。它可以通过域间路由协议转换为 RIP2（7.4.5 节）以及 OSPF（7.5 节）实现。根据 VLSM 可以将网络地址划分为主机数为 500 个时子网掩码长度为/23，主机数为 50 个时子网掩码长度为/26。从而在理论上可以将 IP 地址的利用率提高至 50%。

▼Variable Length Subnet Mask

有了 CIDR 和 VLSM 技术，确实相对缓解了全局 IP 地址▼不够用的问题。但是 IP 地址的绝对数本身有限的事实无法改变。因此才会出现本章 4.6 节中将要介绍的 IPv6 等 IPv4 以外的方法。

▼为了对应全局 IP 地址不足的问题，除了 CIDR 和 VLSM 之外还有 NAT（5.6 节）、代理服务器（1.9.7 节）等技术。

## 4.3.8 全局地址与私有地址

起初，互联网中的任何一台主机或路由器必须配有一个唯一的 IP 地址。一旦出现 IP 地址冲突，就会使发送端无法判断究竟应该发给哪个地址。而接收端收到

数据包以后发送回执时，由于地址重复，发送端也无从得知究竟是哪个主机返回的信息，影响通信的正常进行。

然而，随着互联网的迅速普及，IP地址不足的问题日趋显著。如果一直按照现行的方法采用唯一地址的话，会有IP地址耗尽的危险。

于是就出现了一种新技术。它不要求为每一台主机或路由器分配一个固定的IP地址，而是在必要的时候只为相应数量的设备分配唯一的IP地址。

尤其对于那些没有连接互联网的独立网络中的主机，只要保证在这个网络内地址唯一，可以不用考虑互联网即可配置相应的IP地址。不过，即使让每个独立的网络各自随意地设置IP地址，也可能会有问题▼。于是又出现了私有网络的IP地址。它的地址范围如下所示：

▼例如因运维方案发生变化该网络需要连接到互联网时，或者不小心误被连接到了互联网时，再例如连接两个本来就各自独立的网络时，都容易发生地址冲突。

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 10. | 0. | 0. | 0 | ~ | 10. | 255. | 255. | 255 | (10/8) | A类 |
| 172. | 16. | 0. | 0 | ~ | 172. | 31. | 255. | 255 | (172.16/12) | B类 |
| 192. | 168. | 0. | 0 | ~ | 192. | 168. | 255. | 255 | (192.168/16) | C类 |

包含在这个范围内的IP地址都属于私有IP，而在此之外▼的IP地址称为全局IP▼。

▼A类~C类范围中除去0/8、127/8。

▼也叫公网IP。

私有IP最早没有计划连接互联网，而只用于互联网之外的独立网络。然而，当一种能够互换私有IP与全局IP的NAT▼技术诞生以后，配有私有地址的主机与配有全局地址的互联网主机实现了通信。

▼更多细节请参考5.6节。

现在有很多学校、家庭、公司内部正采用在每个终端设置私有IP，而在路由器（宽带路由器）或在必要的服务器上设置全局IP地址的方法。而如果配有私有IP的地址主机连网时，则通过NAT进行通信。

全局IP地址基本上要在整个互联网范围内保持唯一▼，但私有地址不需要。只要在同一个域里保证唯一即可。在不同的域里出现相同的私有IP不会影响使用。

▼在使用任播（5.8.2节）的情况下，多台主机或路由器可以配置同一个IP。

由此，私有IP地址结合NAT技术已成为现在解决IP地址分配问题的主流方案。它与使用全局IP地址相比有各种限制▼。为了解决这些问题IPv6出现了。然而由于现在IPv6还没有得到普及，IPv4地址又即将耗尽，人们正在努力使用IPv4和NAT技术解决现有的问题。这也是互联网的现状之一。

▼例如在应用的首部或数据部分传递IP地址和端口号的应用程序来说，直接使用私有地址会导致无法通信。

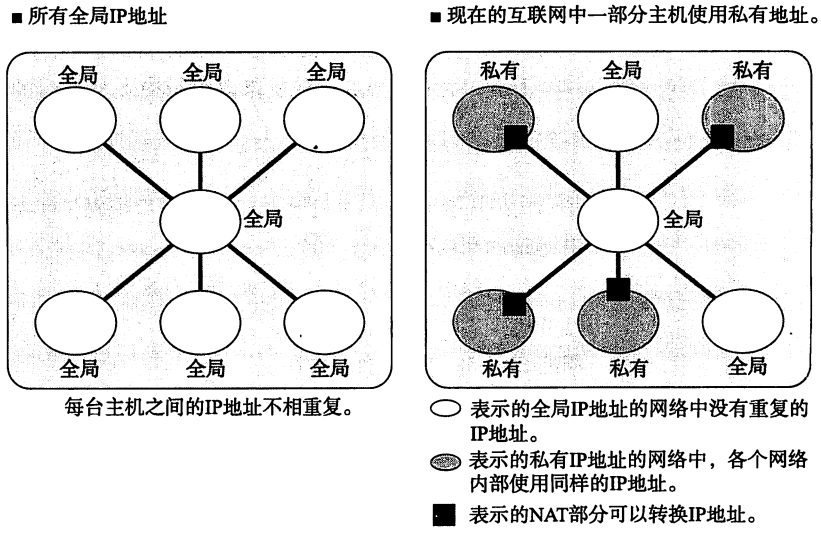



图4.20 全局IP与私有IP

## 4.3.9 全局地址由谁决定

到此，读者可能会问这个所谓的全局地址究竟是由谁管理，又是由谁制定的呢？在世界范围内，全局 IP 由 ICANN▼ 进行管理。在日本则由一个叫做 JPNIC▼ 的机构进行管理，它是日本国内唯一指定的全局 IP 地址管理的组织。

▼ Internet Corporation for Assigned Names and Numbers，中文叫“互联网名称与数字地址分配机构”，负责管理全世界的 IP 地址和域名。

▼ Japan Network Information Center，负责日本国内 IP 地址与 AS 编号的管理。

在互联网被广泛商用之前，用户只有直接向 JPNIC 申请全局 IP 地址才能接入互联网。然而，随着 ISP 的出现，人们在向 ISP 申请接入互联网的同时往往还会申请全局 IP 地址。在这种情况下，实际上是 ISP 代替用户向 JPNIC 申请了一个全局 IP 地址。而连接某个区域网络时，一般不需要联系提供商，只要联系该区域网络的运营商即可。

对于 FTTH 和 ADSL 的服务，网络提供商直接给用户分配全局 IP 地址，并且用户每次重连该 IP 地址都可能会发生变化。这时的 IP 地址由提供商维护，不需要用户亲自申请全局 IP 地址。

一般只有在需要固定 IP 的情况下才会申请全局 IP 地址。例如，如果要让多台主机接入互联网，就需要为每一台主机申请一个 IP 地址。

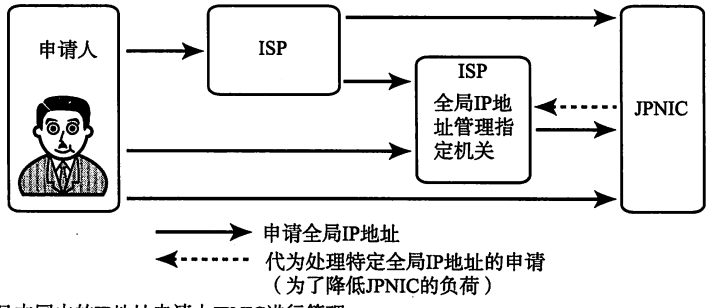



图 4.21 IP 地址的申请流程

日本国内的IP地址申请由JPNIC进行管理。
也有指定的代理全局IP地址分配及管理的机构。
一般的用户，申请IP地址可以联系ISP。如果直接向JPNIC申请有时可能会遭到拒绝。

不过现在，普遍采用的一种方式是，在 LAN 中按照 4.3.8 节所介绍的那样设置私有地址，通过少数设置全局 IP 地址的代理服务器（1.9.7 节）结合 NAT（5.6 节）的设置进行互联网通信。这时 IP 地址个数就不限于 LAN 中主机个数而是由代理服务器和 NAT 的个数决定。

如果完全使用公司内网，今后不会接入互联网，只要使用私有地址即可。

### WHOIS

互联网其实是由各种各样的域组合而成的。分组数据像包中继那样经过众多域才能被发送出去。也就是说，即使是在相互认识的人与人之间进行通信，包在传输过程中所经过的线路或设备也无从得知。而且通常为了实现正常通信，也不需要了解这些信息。

然而，有时在包的传输过程中可能会遇到一些意外▼。如果这些异常仅仅是跟自己或对端有关，那么直接联系对端或许就能够很容易地解决问题。但是如果这些异常是由途中其他设备所造成的，那该如何是好呢？

▼例如，设备上的错误配置或设备本身的故障、缺陷导致线路频繁切换以及网络不稳定，路由错误甚至会导致无法与子网主机进行通信、丢包等问题。

此时，网络技术人员可以通过检查ICMP包▼、利用traceroute▼等命令定位发生异常的设备或线路最近的IP地址。一旦明确了IP地址，就可以跟管理这个IP地址的域管理员取得联系，提出问题并找到解决问题的突破口▼。

不过，这里也有一个问题。那就是即使知道了发生问题的IP地址，该如何了解该IP隶属于哪个域哪个机构？对此，又该如何定位呢？尤其在近来网络病毒的入侵愈加迅猛，受感染的主机很有可能在不知情的情况下又将非法的数据包继续转发出去。管理员在处理此类问题时，必须通过IP地址和主机名定位出具体管理人。

为了解决这个问题，互联网中从很早开始就可以通过网络信息查询机构和管理人联系方式。这种方法就叫做WHOIS。WHOIS提供查询IP地址、AS编号以及搜索域名分配登记和管理人信息的服务。

例如，查找在日本国内使用的特定IP可以在Unix下输入如下命令：

```
whois-h whois.nic.ad.jp <IP地址>
```

使用域名▼的情况下，可以输入如下命令：

```
whois-h whois.jprs.jp <域名>
```

最近，亦可使用面向浏览器的web服务。

- IP地址、AS编号：
  http://www.nic.ad.jp/ja/whois/ja-gateway.html
- 域名：http://whois.jprs.jp/

▼ICMP是诊断IP时必须的信息。更多细节请参考5.4节。

▼利用ICMP呈现线路上路由的一种命令。更多细节请参考5.4.2节。

▼在互联网上即使遇到问题也没有受理问题的服务窗口。所用用户包括互联网提供商的相互合作解决所遇到的问题。网管需要做的就是当发生问题时，跟发生问题的那个域管理员取得联系。当域管理员发现是本域的设备出现故障时应提供应对办法。

▼类似于ohmsha.co.jp的互联网地址。更多细节请参考5.2.3节。①另外，中国也有众多提供whois查询的网站。

① 例如：查找域名可参考http://ewhois.cnnic.net.cn/，查找IP地址和AS编号可参考http://ipwhois.cnnic.net.cn/ip-whois.php。——译者注

# 4.4 路由控制

发送数据包时所使用的地址是网络层的地址，即 IP 地址。然而仅仅有 IP 地址还不足以实现将数据包发送到对端目标地址，在数据发送过程中还需要类似于“指明路由器或主机”的信息，以便真正发往目标地址。保存这种信息的就是路由控制表（Routing Table）。实现 IP 通信的主机和路由器都必须持有一张这样的表。它们也正是在这个表格的基础上才得以进行数据包发送的。

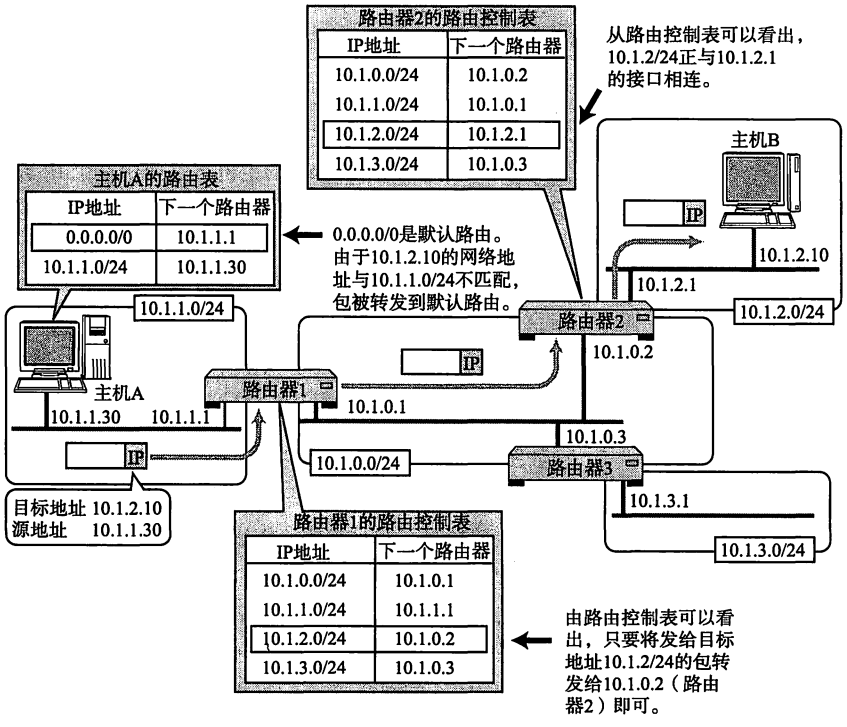
该路由控制表的形成方式有两种：一种是管理员手动设置，另一种是路由器与其他路由器相互交换信息时自动刷新。前者也叫静态路由控制，而后者叫做动态路由控制。为了让动态路由及时刷新路由表，在网络上互连的路由器之间必须设置好路由协议，保证正常读取路由控制信息。

IP 协议始终认为路由表是正确的。然而，IP 本身并没有定义制作路由控制表的协议。即 IP 没有制作路由控制表的机制。该表是由一个叫做“路由协议”（这个协议有别于 IP）的协议制作而成。关于路由协议的更多细节将在后续的第 7 章详细介绍。

## 4.4.1 IP 地址与路由控制

IP 地址的网络地址部分用于进行路由控制。图 4.22 即发送 IP 包的示例。

图 4.22
路由控制表与 IP 包发送




路由控制表中记录着网络地址与下一步应该发送至路由器的地址▼。在发送 IP 包时，首先要确定 IP 包首部中的目标地址，再从路由控制表中找到与该地址具有相同网络地址的记录，根据该记录将 IP 包转发给相应的下一个路由器。如果路

▼在 Windows 或 Unix 上表示路由表的方法分别为 netstat-r 或 netstat-rn。

由控制表中存在多条相同网络地址的记录，就选择一个最为吻合的网络地址。所谓最为吻合是指相同位数最多的意思▼。

▼也叫最长匹配。

例如172.20.100.52的网络地址与172.20/16和172.20.100/24两项都匹配。此时，应该选择匹配度最长的172.20.100/24。此外，如果路由表中下一个路由器的位置记录着某个主机或路由器网卡的IP地址，那就意味着“发送的目标地址属于同一个链路”▼。

▼目标地址在同一个链路中的情况下，路由表的记录格式可能会根据操作系统和路由器种类的不同而有所区别。

### 默认路由

如果一张路由表中包含所有的网络及其子网的信息，将会造成无端的浪费。这时，默认路由（Default Route）是不错的选择。默认路由是指路由表中任何一个地址都能与之匹配的记录。

默认路由一般标记为0.0.0.0/0或default▼。这里的0.0.0.0/0并不是指IP地址是0.0.0.0。由于后面是“/0”，所以并没有标识IP地址▼。它只是为了避免人们误以为0.0.0.0是IP地址。有时默认路由也被标记为default，但是在计算机内部和路由协议的发送过程中还是以0.0.0.0/0进行处理。

▼表示子网掩码时，IP地址为0.0.0.0，子网掩码也是0.0.0.0。

▼0.0.0.0的IP地址应该记述为0.0.0.0/32。

### 主机路由

“IP地址/32”也被称为主机路由（Host Route）。例如，192.168.153.15/32▼就是一种主机路由。它的意思是整个IP地址的所有位都将参与路由。进行主机路由，意味着要基于主机上网卡上配置的IP地址本身，而不是基于该地址的网络地址部分进行路由。

▼表示子网掩码时，若IP地址为192.168.153.15，其对应的子网掩码为255.255.255.255。

主机路由多被用于不希望通过网络地址路由的情况▼。

▼不过，请读者注意，使用主机路由会导致路由表膨大，路由负荷增加，进而造成网络性能下降。

### 环回地址

环回地址是在同一台计算机上的程序之间进行网络通信时所使用的一个默认地址。计算机使用一个特殊的IP地址127.0.0.1作为环回地址。与该地址具有相同意义的是一个叫做localhost的主机名。使用这个IP或主机名时，数据包不会流向网络。

## 4.4.2　路由控制表的聚合

利用网络地址的比特分布可以有效地进行分层配置。对内即使有多个子网掩码，对外呈现出的也是同一个网络地址。这样可以更好地构建网络，通过路由信息的聚合可以有效地减少路由表的条目▼。

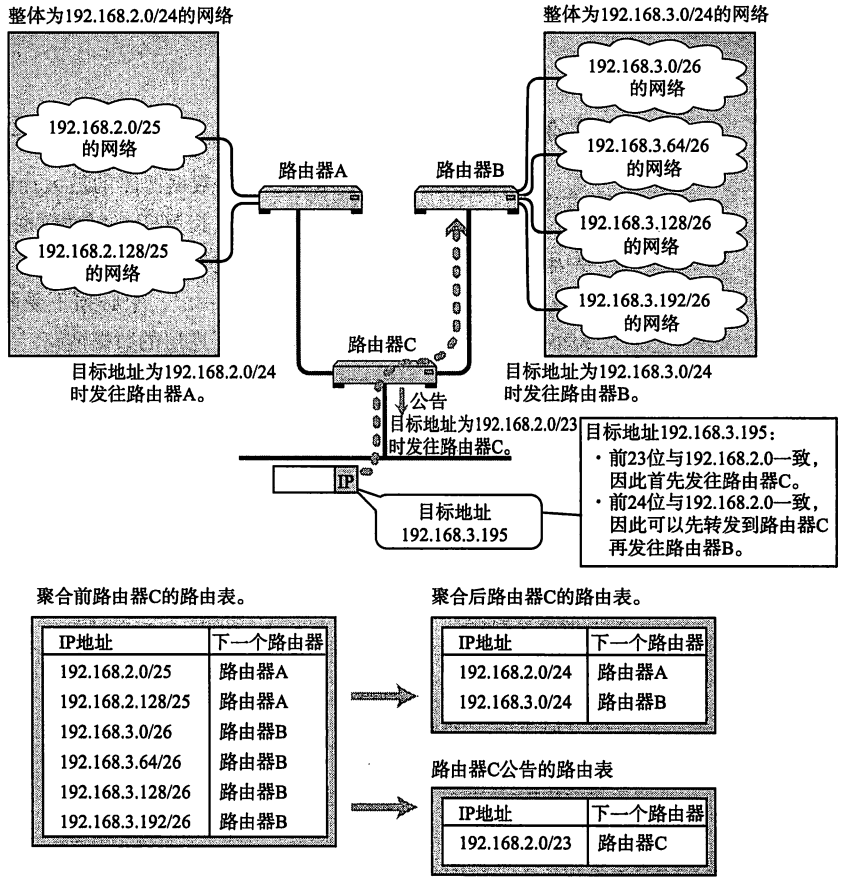
▼路由表的聚合也叫路由汇总（Aggregation）。

如图4.23所示，在聚合之前需要6条路由记录，聚合之后只需要2条记录。

能够缩小路由表的大小是它最大的优势。路由表越大，管理它所需要的内存和CPU也就越多。并且查找路由表的时间也会越长，导致转发IP数据包的性能下降。如果想要构建大规模、高性能网络，则需要尽可能削减路由表的大小。

而且路由聚合可以将已知的路由信息传送给周围其他的路由器，以达到控制路由信息的目的。图4.23的例子中路由器C正是将已知192.168.2.0/24与192.168.3.0/24的网络这一信息聚合成为对“192.168.2.0/23的网络也已知”，从而进行公示。

图 4.23
路由控制表聚合的例子




聚合前路由器C的路由表。

| IP地址 | 下一个路由器 |
|---|---|
| 192.168.2.0/25 | 路由器A |
| 192.168.2.128/25 | 路由器A |
| 192.168.3.0/26 | 路由器B |
| 192.168.3.64/26 | 路由器B |
| 192.168.3.128/26 | 路由器B |
| 192.168.3.192/26 | 路由器B |

聚合后路由器C的路由表。

| IP地址 | 下一个路由器 |
|---|---|
| 192.168.2.0/24 | 路由器A |
| 192.168.3.0/24 | 路由器B |

路由器C公告的路由表

| IP地址 | 下一个路由器 |
|---|---|
| 192.168.2.0/23 | 路由器C |

# 4.5 IP 分割处理与再构成处理

## 4.5.1 数据链路不同，MTU 则相异

如前面 4.2.3 节中所介绍，每种数据链路的最大传输单元（MTU）都不尽相同。表 4.2 列出了很多不同的链路及其 MTU。每种数据链路的 MTU 之所以不同，是因为每个不同类型的数据链路的使用目的不同。使用目的不同，可承载的 MTU 也就不同。鉴于 IP 属于数据链路上一层，它必须不受限于不同数据链路的 MTU 大小。如 4.2.3 节所述，IP 抽象化了底层的数据链路。

表 4.2
各种数据链路及其 MTU

| 数据链路 | MTU（字节） | 总长度（单位为字节，包含 FCS） |
|---|---|---|
| IP 的最大 MTU | 65535 | - |
| Hyperchannel | 65535 | - |
| IP over HIPPI | 65280 | 65320 |
| 16Mbps IBM Token Ring | 17914 | 17958 |
| IP over ATM | 9180 | - |
| IEEE 802.4 Token Bus | 8166 | 8191 |
| IEEE 802.5 Token Ring | 4464 | 4508 |
| FDDI | 4352 | 4500 |
| 以太网 | 1500▼ | 1518 |
| PPP（Default） | 1500 | - |
| IEEE 802.3 Ethernet | 1492 | 1518 |
| PPPoE | 1492 | - |
| X.25 | 576 | - |
| IP 的最小 MTU | 68 | - |

▼最近以太网也可以使用大于 1500 字节的 MTU。这种方式叫做 Jumbo Frame，是指超长帧格式。为了提高服务器主机的通信速度，采用 9000 字节左右 MTU 的情况更多一些。使用 Jumbo Frame 不仅要对应网段的主机，还需要路由器、交换机和网桥（交换集线器）的支持。即使在不使用 Jumbo Frame 的情况下，经由 IP 隧道也能通过途中的路由器或网桥实现 1500 字节以上 MTU 的通信。因此，如果想避免过多的 IP 碎片，可以适当地扩大路由器或网桥上的 MTU 值。

## 4.5.2 IP 报文的分片与重组

任何一台主机都有必要对 IP 分片（IP Fragmentation）进行相应的处理。分片往往在网络上遇到比较大的报文无法一下子发送出去时才会进行处理。

图 4.24 展示了网络传输过程中进行分片处理的一个例子。由于以太网的默认 MTU 是 1500 字节，因此 4342 字节的 IP 数据报无法在一个帧当中发送完成。这时，路由器将此 IP 数据报划分成了 3 个分片进行发送。而这种分片处理只要路由器认为有必要，会周而复始地进行▼。

▼分片以 8 个字节的倍数为单位进行。

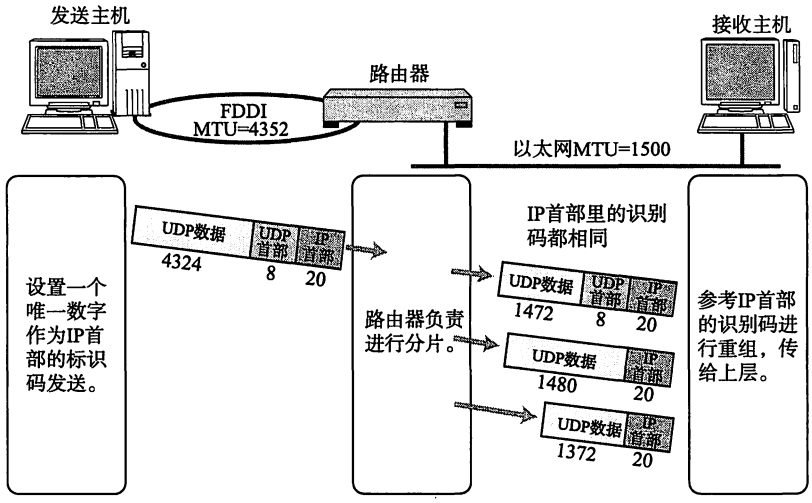
经过分片之后的 IP 数据报在被重组的时候，只能由目标主机进行。路由器虽然做分片但不会进行重组。

这样的处理是由诸多方面的因素造成的。例如，现实当中无法保证 IP 数据报是否经由同一个路径传送。因此，途中即使等待片刻，数据包也有可能无法到达目的地。此外，拆分之后的每个分片也有可能会在途中丢失▼。即使在途中某一

▼在目标主机上进行分片的重组时，可能有一部分包会延迟到达。因此，一般会从第一个数据报的分片到达的那一刻起等待约 30 秒再进行处理。

处被重新组装，但如果下一站再经过其他路由时还会面临被分片的可能。这会给路由器带来多余的负担，也会降低网络传送效率。出于这些原因，在终结点（目标主机）端重组分片了的 IP 数据报成为现行的规范。

图 4.24
IP 报文的分片与重组




IP首部中的"片偏移"字段表示分片之后每个分片在用户数据中的相对位置和该分片之后是否还有后续其他分片。
根据这个字段可以判断一个IP数据报是否分片以及当前分片为整个数据报的起始、中段还是末尾。
（数字表示数据长度。单位为字节）

## 4.5.3 路径 MTU 发现

分片机制也有它的不足。首先，路由器的处理负荷加重。随着时代的变迁，计算机网络的物理传输速度不断上升。这些高速的链路，对路由器和计算机网络提出了更高的要求。另一方面，随着人们对网络安全的要求提高，路由器需要做的其他处理也越来越多，如网络过滤▼等。因此，只要允许，是不希望由路由器进行 IP 数据包的分片处理的。

▼过滤是指只有带有一定特殊参数的 IP 数据报才能通过路由器。这里的参数可以是发送端主机、接收端主机、TCP 或 UDP 端口号或者 TCP 的 SYN 标志或 ACK 标志等。

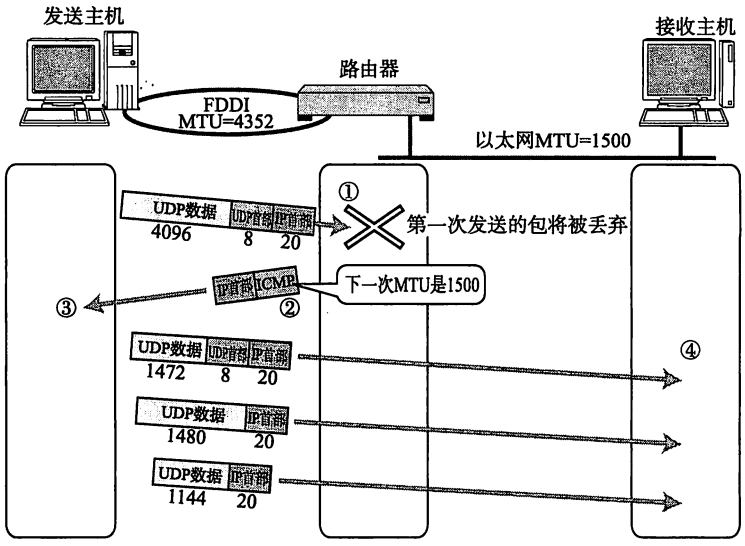
其次，在分片处理中，一旦某个分片丢失，则会造成整个 IP 数据报作废。为了避免此类问题，TCP 的初期设计还曾使用过更小▼的分片进行传输。其结果是网路的利用率明显下降。

▼包含 TCP 的数据限制在 536 字节或 512 字节。

为了应对以上问题，产生了一种新的技术"路径 MTU 发现"（Path MTU Discovery▼）。所谓路径 MTU（Path MTU）是指从发送端主机到接收端主机之间不需要分片时最大 MTU 的大小。即路径中存在的所有数据链路中最小的 MTU。而路径 MTU 发现从发送主机按照路径 MTU 的大小将数据报分片后进行发送。进行路径 MTU 发现，就可以避免在中途的路由器上进行分片处理，也可以在 TCP 中发送更大的包。现在，很多操作系统都已经实现了路径 MTU 发现的功能。

▼也可以缩写为 PMTUD。

图4.25
路径MTU发现的机制(UDP的情况下)




① 发送时IP首部的分片标志位设置为不分片。路由器丢包。
② 由ICMP通知下一次MTU的大小。
③ UDP中没有重发处理。应用在发送下一个消息时会被分片。具体来说，就是指UDP层传过来的"UDP首部+UDP数据"在IP层被分片。对于IP，它并不区分UDP首部和应用的数据。
④ 所有的分片到达目标主机后被重组，再传给UDP层。
（数字表示数据长度，单位为字节）

路径MTU发现的工作原理如下：

首先在发送端主机发送IP数据报时将其首部的分片禁止标志位设置为1。根据这个标志位，途中的路由器即使遇到需要分片才能处理的大包，也不会去分片，而是将包丢弃。随后，通过一个ICMP的不可达消息将数据链路上MTU的值给发送主机▼。

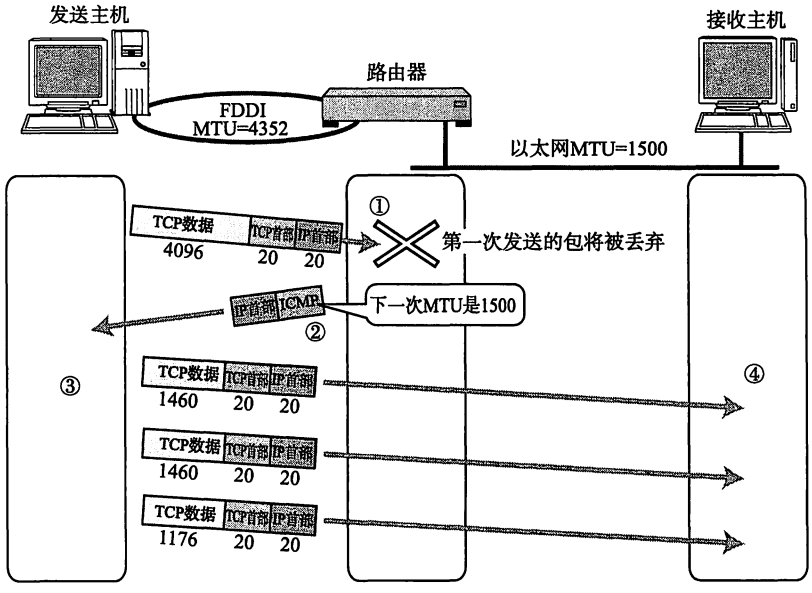
▼具体来说，以ICMP不可达消息中的分片需求（代码4）进行通知。然而，在有些老式的路由器中，ICMP可能不包含下一个MTU值。这时，发送主机端必须不断增减包的大小，以此来定位一个合适的MTU值。

下一次，从发送给同一个目标主机的IP数据报获得ICMP所通知的MTU值以后，将它设置为当前MTU。发送主机根据这个MTU对数据报进行分片处理。如此反复，直到数据报被发送到目标主机为止没有再收到任何ICMP，就认为最后一次ICMP所通知的MTU即是一个合适的MTU值。那么，当MTU的值比较多时，最少可以缓存▼约10分钟。在这10分钟内使用刚刚求得的MTU，但过了这10分钟以后则重新根据链路上的MTU做一次路径MTU发现。

▼缓存是指将反复使用的信息暂时保存到一个可以即刻获取的位置。

前面是UDP的例子。那么在TCP的情况下，根据路径MTU的大小计算出最大段长度（MSS），然后再根据这些信息进行数据报的发送。因此，在TCP中如果采用路径MTU发现，IP层则不会再进行分片处理。关于TCP的最大段长度，请参考6.4.5节。

图 4.26
路径 MTU 发现的机制（TCP 的情况下）




① 发送时IP首部的分片标志位设置为不分片。路由器丢包。
② 由ICMP通知下一次MTU的大小▼。
③ 根据TCP的重发处理，数据报会被重新发送。TCP负责将数据分成IP层不会再被分片的粒度以后传给IP层　IP层不再做分片处理。
④ 不需要重组。数据被原样发送给接收端主机的TCP层。

（数字表示数据长度，单位为字节）

▼出于网络安全的考虑，有些域会限制 ICMP 消息的接收。然而实际上这也有问题。因为这时路径 MTU 发现的功能无法正常运行，会造成最终用户不明，导致连接不稳定。

# 4.6 IPv6

## 4.6.1 IPv6的必要性

IPv6（IP version 6）是为了根本解决IPv4地址耗尽的问题而被标准化的网际协议。IPv4的地址长度为4个8位字节，即32比特。而IPv6的地址长度则是原来的4倍，即128比特▼，一般写成8个16位字节。

▼因此IPv6的地址空间是IPv4的$2^{96}=7.923\times10^{28}$倍。

从IPv4切换到IPv6极其耗时，需要将网络中所有主机和路由器的IP地址进行重新设置。当互联网广泛普及后，替换所有IP地址会是更为艰巨的任务。

也是出于上述原因，IPv6不仅仅能解决IPv4地址耗尽的问题，它甚至试图弥补IPv4中的绝大多数缺陷。目前，人们正着力于进行IPv4与IPv6之间的相互通信与兼容性方面的测试▼。

▼即IP隧道（5.7节）和协议转换（5.6.3节）等。

## 4.6.2 IPv6的特点

IPv6具有以下几个特点。这些功能中的一部分在IPv4中已经得以实现。然而，即便是那些实现IPv4的操作系统，也并非实现了所有的IPv4功能。这中间不乏存在根本无法使用或需要管理员介入才能实现的部分。而IPv6则将这些通通作为必要的功能，减轻了管理员的负担▼。

▼这些只能在IPv6的情况下使用。如果想要在IPv4和IPv6都投入使用，工作量恐怕会是原来的两倍不止。

- IP地址的扩大与路由控制表的聚合

  IP地址依然适应互联网分层构造。分配与其地址结构相适应的IP地址，尽可能避免路由表膨大。

- 性能提升

  包首部长度采用固定的值（40字节），不再采用首部检验码。简化首部结构，减轻路由器负荷。路由器不再做分片处理（通过路径MTU发现只由发送端主机进行分片处理）。

- 支持即插即用功能

  即使没有DHCP服务器也可以实现自动分配IP地址。

- 采用认证与加密功能

  应对伪造IP地址的网络安全功能以及防止线路窃听的功能（IPsec）。

- 多播、Mobile IP成为扩展功能

  多播和Mobile IP被定义为IPv6的扩展功能。由此可以预期，曾在IPv4中难于应用的这两个功能在IPv6中能够顺利使用。

## 4.6.3 IPv6中IP地址的标记方法

IPv6的IP地址长度为128位。它所能表示的数字高达38位数（$2^{128}$ = 约$3.40\times10^{38}$）。这可谓是天文数字，足以为人们所能想象到的所有主机和路由器分配地址。

如果将IPv6的地址像IPv4的地址一样用十进制数据表示的话，是16个数字的序列（IPv4是4个数字的序列）。由于用16个数字序列表示显得有些麻烦，因

此，将 IPv6 和 IPv4 在标记方法上进行区分。一般人们将 128 比特 IP 地址以每 16 比特为一组，每组用冒号（“:”）隔开进行标记。而且如果出现连续的 0 时还可以将这些 0 省略，并用两个冒号（“::”）隔开。但是，一个 IP 地址中只允许出现一次两个连续的冒号。

在 IPv6 当中，人们正在努力使用最简单的方法标记 IP 地址，以便易于记忆。

- IPv6 的 IP 地址标记举例
  - 用二进制数表示

    1111111011011100：1011101010011000：0111011001010100：
    0011001000010000：1111111011011100：1011101010011000：
    0111011001010100：0011001000010000
  - 用十六进制数表示

    FEDC：BA98：7654：3210：FEDC：BA98：7654：3210
- IPv6 的 IP 地址省略举例
  - 用二进制数表示

    0001000010000000：0000000000000000：0000000000000000：
    0000000000000000：0000000000000000：0000100000000000：
    0010000000001100：0100000101111010
  - 用十六进制数表示

    1080：0：0：0：8：800：200C：417A

    ↓

    1080::8：800：200C：417A（省略后）

## 4.6.4 IPv6 地址的结构

IPv6 类似 IPv4，也是通过 IP 地址的前几位标识 IP 地址的种类。

在互联网通信中，使用一种全局的单播地址。它是互联网中唯一的一个地址，不需要正式分配 IP 地址。

限制型网络，即那些不与互联网直接接入的私有网络，可以使用唯一本地地址。该地址根据一定的算法生成随机数并融合到地址当中，可以像 IPv4 的私有地址一样自由使用。

在不使用路由器或者在同一个以太网网段内进行通信时，可以使用链路本地单播地址。

而在构建允许多种类型 IP 地址的网络时，在同一个链路上也可以使用全局单播地址以及唯一本地地址进行通信。

在 IPv6 的环境下，可以同时将这些 IP 地址全都配置在同 1 个 NIC 上，按需灵活使用。

图4.27
IPv6中的通信

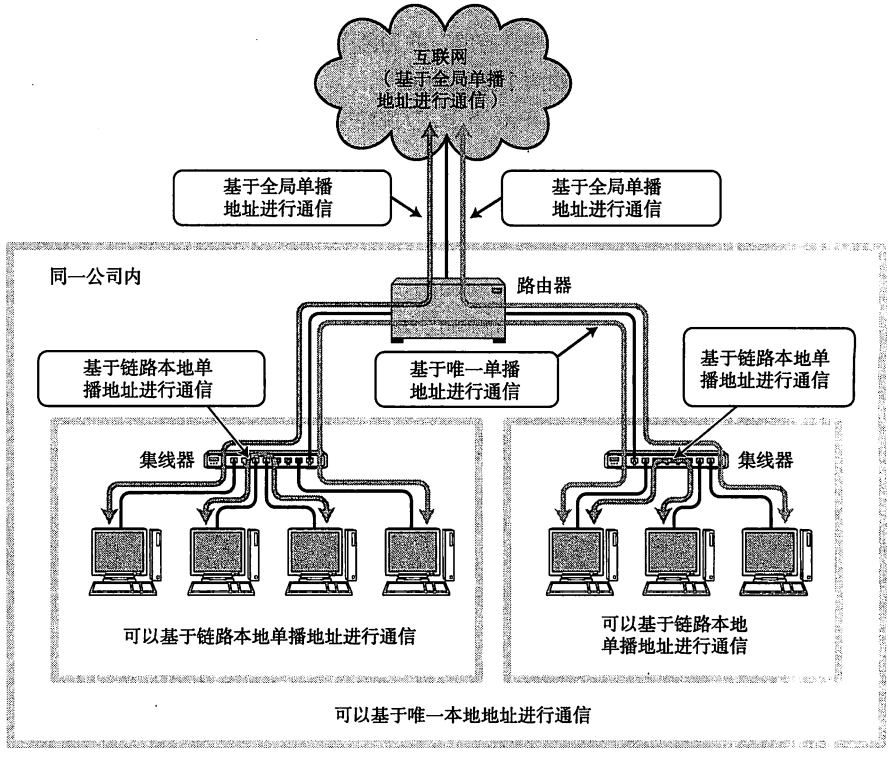



表4.3
IPv6地址结构

| | | |
|---|---|---|
| 未定义 | 0000 … 0000（128比特） | :: /128 |
| 环回地址 | 0000 … 0001（128比特） | :: 1/128 |
| 唯一本地地址 | 1111 110 | FC00:: /7 |
| 链路本地单播地址 | 1111 1110 10 | FE80:: /10 |
| 多播地址 | 1111 1111 | FF00:: /8 |
| 全局单播地址 | （其他） | |

## 4.6.5　全局单播地址

全局单播地址是指世界上唯一的一个地址。它是互联网通信以及各个域内部通信中最为常用的一个IPv6地址。

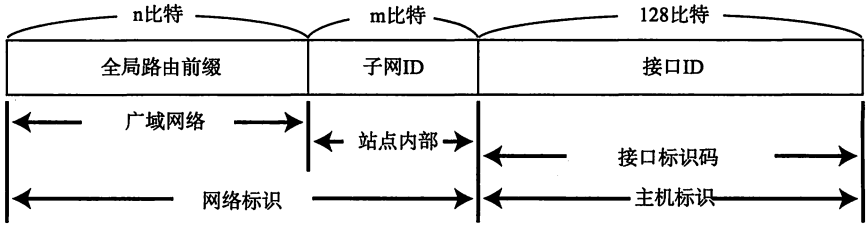
全局单播地址的格式如图4.28所示。现在IPv6的网络中所使用的格式为，n=48，m=16以及128-n-m=64。即前64比特为网络标识，后64比特为主机标识。

通常，接口ID中保存64比特版的MAC地址的值。不过由于MAC地址▼属于设备固有的信息，有时不希望让对端知道。这时的接口ID可设置为一个与MAC地址没有关系的“临时地址”。这种临时地址通常随机产生，并会定期更新。因此，从IPv6地址中查看定位设备变得没那么简单。究竟会是哪种信息，全由操作系统的具体装置决定▼。

▼称为IEEE EUI-64识别码。

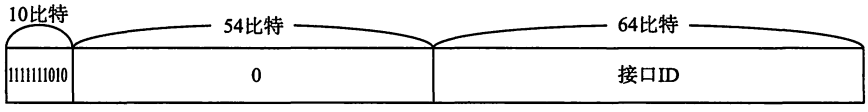
▼常被用作客户端的个人电脑中分配这种临时地址的情况多一些。

图 4.28
全局单播地址




## 4.6.6 链路本地单播地址

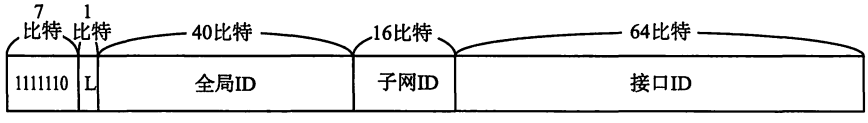
图 4.29
链路本地单播地址




链路本地单播地址是指在同一个数据链路内唯一的地址。它用于不经过路由器，在同一个链路中的通信。通常接口 ID 保存 64 比特版的 MAC 地址。

## 4.6.7 唯一本地地址

图 4.30
唯一本地地址




※ L通常被置为1.
※ 全局ID的值随机决定
※ 子网ID是指该域子网地址
※ 接口ID即为接口的ID

唯一本地地址是不进行互联网通信时所使用的地址。

设备控制的限制型网络以及金融机关的核心网等会与互联网隔离。为了提高安全性，企业内部的网络与互联网通信时通常会通过 NAT 或网关（代理）进行。而唯一本地地址正是在这种不联网或通过 NAT 以及代理联网的环境下使用的。

唯一本地地址虽然不会与互联网连接，但是也会尽可能地随机生成一个唯一的全局 ID。由于企业兼并、业务统一、效率提高等原因，很有可能会需要用到唯一本地地址进行网络之间的连接。在这种情况下，人们希望可以在不改动 IP 地址的情况下即可实现网络的统一▼。

▼全局 ID 不一定必须是全世界唯一的，但是完全一致的可能性也不高。

## 4.6.8 IPv6 分段处理

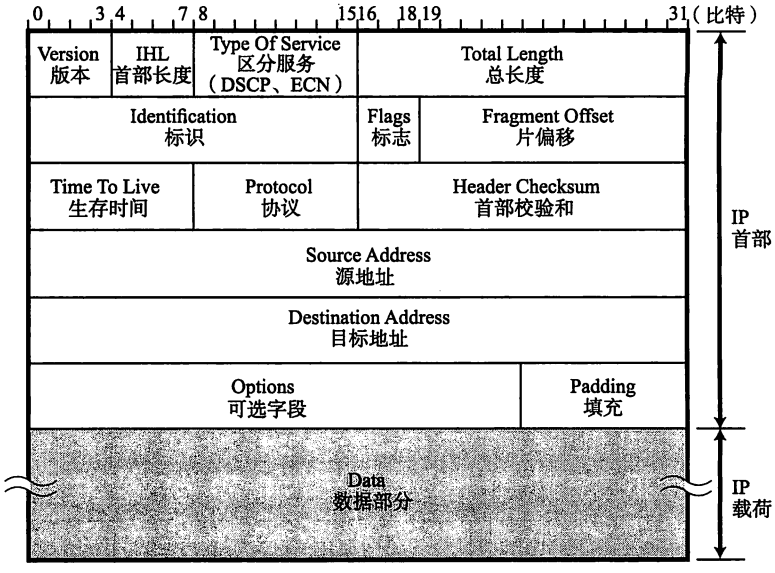
IPv6 的分片处理只在作为起点的发送端主机上进行，路由器不参与分片。这也是为了减少路由器的负荷，提高网速。因此，IPv6 中的“路径 MTU 发现”功能必不可少。不过 IPv6 中最小 MTU 为 1280 字节。因此，在嵌入式系统中对于那些有一定系统资源限制▼的设备来说，不需要进行“路径 MTU 发现”，而是在发送 IP 包时直接以 1280 字节为单位分片送出。

▼ CPU 处理能力或内存限制等。

# 4.7 IPv4 首部

通过IP进行通信时，需要在数据的前面加入IP首部信息。IP首部中包含着用于IP协议进行发包控制时所有的必要信息。了解IP首部的结构，也就能够对IP所提供的功能有一个详细的把握。

图4.31 IP数据报格式（IPv4）




## 版本（Version）

由4比特构成，表示标识IP首部的版本号。IPv4的版本号即为4，因此在这个字段上的值也是“4”。此外，关于IP的所有版本在以下表4.4中列出。关于IP版本的最新情况，读者也可以在以下网址发布的信息中查看：

```
http://www.iana.org/assignments/version-numbers
```

表4.4 IP首部的版本号

| 版本 | 简　称 | 协　议 |
|---|---|---|
| 4 | IP | Internet Protocol |
| 5 | ST | ST Datagram Mode |
| 6 | IPv6 | Internet Protocol version 6 |
| 7 | TP/IX | TP/IX：The Next Internet |
| 8 | PIP | The P Internet Protocol |
| 9 | TUBA | TUBA |

**关于 IP 版本号**

IPv4 的下一个版本是 IPv6。那么为什么要从版本 4 直接跳到版本 6 呢?

这里需要提到的是，IP 版本号的含义与普通软件的版本号有所区别。普通的软件产品，版本号会随着更新逐渐增大，最新版本号即为最大号码。这是基于每款软件都由特定的软件公司或团体进行开发才能实现的。

而在互联网中，为了让 IP 协议更为完善，有众多机构致力于它的规范化。为了让这些机构能够验证相应的 IP 协议，它们会按照顺序分配具体的版本。

一向重视实践的互联网，在遇到好的提案时，不能只纸上谈兵，还需要反复实验。为此，对于那些还未正式被广泛使用的版本就会像表 4.4 所示那样标上几个号码，从而在实验的过程中，选择一个最佳的产物进行标准化。IP version 6（IPv6）正是经历了这些过程后才成为 IPv4 下一代的 IP 协议的。因此，IP 协议版本号的大小本身没有什么太大的意义。

### 首部长度（IHL：Internet Header Length）

由 4 比特构成，表明 IP 首部的大小，单位为 4 字节（32 比特）。对于没有可选项的 IP 包，首部长度则设置为“5”。也就是说，当没有可选项时，IP 首部的长度为 20 字节（4×5=20）。

### 区分服务（TOS：Type Of Service）

由 8 比特构成，用来表明服务质量。每一位的具体含义如表 4.5 所示。

表 4.5 服务类型中各比特的含义

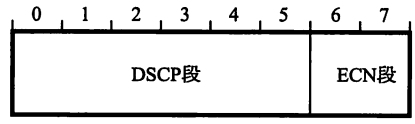
| 比　特 | 含　义 |
|---|---|
| 0 1 2 | 优先度▼ |
| 3 | 最低延迟 |
| 4 | 最大吞吐 |
| 5 | 最大可靠性 |
| 6 | 最小代价 |
| (3~6) | 最大安全 |
| 7 | 未定义 |

▼用 0、1、2 这三位表示 0~7 的优先度。从 0 到 7 表示优先度从低到高。

这个值通常由应用指定。而且现在也鼓励这种结合应用的特性设定 TOS 的方法。然而在目前，几乎所有的网络都无视这些字段。这不仅仅是因为在符合质量要求的情况下按其要求发送本身的功能实现起来十分困难，还因为若不符合质量要求就可能会产生不公平的现象。因此，实现 TOS 控制变得极其复杂。这也导致 TOS 整个互联网几乎就没有被投入使用。不过已有人提出将 TOS 字段本身再划分为 DSCP 和 ECN 两个字段的建议。

### DSCP 段与 ECN 段

图 4.32
DSCP 段与 ECN 段




DSCP（Differential Services Codepoint，差分服务代码点）是 TOS（Type Of Service）的一部分。现在统称为 DiffServ▼，用来进行质量控制。

▼关于 DiffServ 的更多细节请参考5.8.3节。

如果3~5位的值为0，0~2位则被称作类别选择代码点。这样就可以像 TOS 的优先度那样提供8种类型的质量控级别。对于每一种级别所采取的措施则由提供 DiffServ 的运营管理者制定。为了与 TOS 保持一致，值越大优先度也越高。如果第5位为1，表示实验或本地使用的意思。

ECN（Explicit Congestion Notification，显式拥塞通告）用来报告网络拥堵情况，由两个比特构成。

表 4.6
ECN 域

| 比特 | 简称 | 含　义 |
|---|---|---|
| 6 | ECT | ECN-Capable Transport |
| 7 | CE | Congenstion Experienced |

第6位的 ECT 用以通告上层 TCP 层协议是否处理 ECN。当路由器在转发 ECN 为1的包的过程中，如果出现网络拥堵的情况，就将 CE 位设置为1▼。

▼关于 ECN 的更多细节请参考5.8.4节。

### 总长度（Total Length）

表示 IP 首部与数据部分合起来的总字节数。该字段长16比特。因此 IP 包的最大长度为65535（$=2^{16}$）字节。

如表4.2所示，目前还不存在能够传输最大长度为65535字节的 IP 包的数据链路。不过，由于有 IP 分片处理，从 IP 的上一层的角度看，不论底层采用何种数据链路，都可以认为能够以 IP 的最大包长传输数据。

### 标识（ID：Identification）

由16比特构成，用于分片重组。同一个分片的标识值相同，不同分片的标识值不同。通常，每发送一个 IP 包，它的值也逐渐递增。此外，即使 ID 相同，如果目标地址、源地址或协议不同的话，也会被认为是不同的分片。

### 标志（Flags）

由3比特构成，表示包被分片的相关信息。每一位的具体含义请参考下表。

表 4.7
标志段各位含义

| 比　特 | 含　义 |
|---|---|
| 0 | 未使用。现在必须是0。 |
| 1 | 指示是否进行分片（don't fragment）<br>0- 可以分片<br>1- 不能分片 |
| 2 | 包被分片的情况下，表示是否为最后一个包（more fragment）。<br>0- 最后一个分片的包<br>1- 分片中段的包 |

### 片偏移（FO：Fragment Offset）

由 13 比特构成，用来标识被分片的每一个分段相对于原始数据的位置。第一个分片对应的值为 0。由于 FO 域占 13 位，因此最多可以表示 8192（$=2^{13}$）个相对位置。单位为 8 字节，因此最大可表示原始数据 8×8192=65536 字节的位置。

### 生存时间（TTL：Time To Live）

由 8 比特构成，它最初的意思是以秒为单位记录当前包在网络上应该生存的期限。然而，在实际中它是指可以中转多少个路由器的意思。每经过一个路由器，TTL 会减少 1，直到变成 0 则丢弃该包▼。

▼ TTL 占 8 位，因此可以表示 0~255 的数字。因此一个包的中转路由的次数不会超过 $2^8$ = 256 个。由此可以避免 IP 包在网络内无限传递的问题。

### 协议（Protocol）

由 8 比特构成，表示 IP 首部的下一个首部隶属于哪个协议。目前常使用的协议如表 4.8 所示已经分配相应的协议编号。

关于协议编号一览表的更新情况可以从以下网站获取：

```
http://www.iana.org/assignments/protocol-numbers
```

表 4.8 上层协议编号

| 分配编号 | 简　称 | 协　议 |
|---|---|---|
| 0 | HOPOPT | IPv6 Hop-by-Hop Option |
| 1 | ICMP | Internet Control Message |
| 2 | IGMP | Internet Group Management |
| 4 | IP | IP in IP（encapsulation） |
| 6 | TCP | Transmission Control |
| 8 | EGP | Exterior Gateway Protocol |
| 9 | IGP | any private interior gateway（Cisco IGRP） |
| 17 | UDP | User Datagram |
| 33 | DCCP | Datagram Congestion Control Protocol |
| 41 | IPv6 | IPv6 |
| 43 | IPv6-Route | Routing Header for IPv6 |
| 44 | IPv6-Frag | Fragment Header for IPv6 |
| 46 | RSVP | Reservation Protocol |
| 50 | ESP | Encap Security Payload |
| 51 | AH | Authentication Header |
| 58 | IPv6-ICMP | ICMP for IPv6 |
| 59 | IPv6-NoNxt | No Next Header for IPv6 |
| 60 | IPv6-Opts | Destination Options for IPv6 |
| 88 | EIGRP | EIGRP |
| 89 | OSPFIGP | OSPF |
| 97 | ETHERIP | Ethernet-within-IP Encapsulation |
| 103 | PIM | Protocol Independent Multicast |

| 分配编号 | 简　称 | 协　议 |
|---|---|---|
| 108 | IPComp | IP Payload Compression Protocol |
| 112 | VRRP | Virtual Router Redundancy Protocol |
| 115 | L2TP | Layer Two Tunneling Protocol |
| 124 | ISIS over IPv4 | ISIS over IPv4 |
| 132 | SCTP | Stream Control Transmission Protocol |
| 133 | FC | Fibre Channel |
| 134 | RSVP-E2E-IGNORE | RSVP-E2E-IGNORE |
| 135 | Mobility Header（IPv6） | Mobility Header（IPv6） |
| 136 | UDPLite | UDP-Lite |
| 137 | MPLS-in-IP | MPLS-in-IP |

### 首部校验和（Header Checksum）

由16比特（2个字节）构成，也叫IP首部校验和。该字段只校验数据报的首部，不校验数据部分。它主要用来确保IP数据报不被破坏。校验和的计算过程，首先要将该校验和的所有位置设置为0，然后以16比特为单位划分IP首部，并用1补数▼计算所有16位字的和。最后将所得到这个和的1补数赋给首部校验和字段。

▼1补数
通常计算机中对整数运算采用2补数的方式。但在校验和的计算中采用1补数运算方法。这样做的优点在于即使产生进位也可以回到第1位，可以防止信息缺失并且可以用2个0区分使用。

### 源地址（Source Address）

由32比特（4个字节）构成，表示发送端IP地址。

### 目标地址（Destination Address）

由32比特（4个字节）构成，表示接收端IP地址。

### 可选项（Options）

长度可变，通常只在进行实验或诊断时使用。该字段包含如下几点信息：

- 安全级别
- 源路径
- 路径记录
- 时间戳

### 填充（Padding）

也称作填补物。在有可选项的情况下，首部长度可能不是32比特的整数倍。为此，通过向字段填充0，调整为32比特的整数倍。

### 数据（Data）

存入数据。将IP上层协议的首部也作为数据进行处理。

# 4.8 IPv6 首部格式

IPv6 的 IP 数据首部格式如图 4.33。相比 IPv4，已经发生了巨大变化。

IPv6 中为了减轻路由器的负担，省略了首部校验和字段▼。因此路由器不再需要计算校验和，从而也提高了包的转发效率。

▼因为 TCP 和 UDP 在做校验和计算的时候使用伪首部，所以可以验证 IP 地址或协议是否正确。因此，即使在 IP 层无法提供可靠传输，在 TCP 或 UDP 层也可以提供可靠传输的服务。关于这一点可以参考 TCP 或 UDP 的详解。

此外，分片处理所用的识别码成为可选项。为了让 64 位 CPU 的计算机处理起来更方便，IPv6 的首部及可选项都由 8 字节构成。

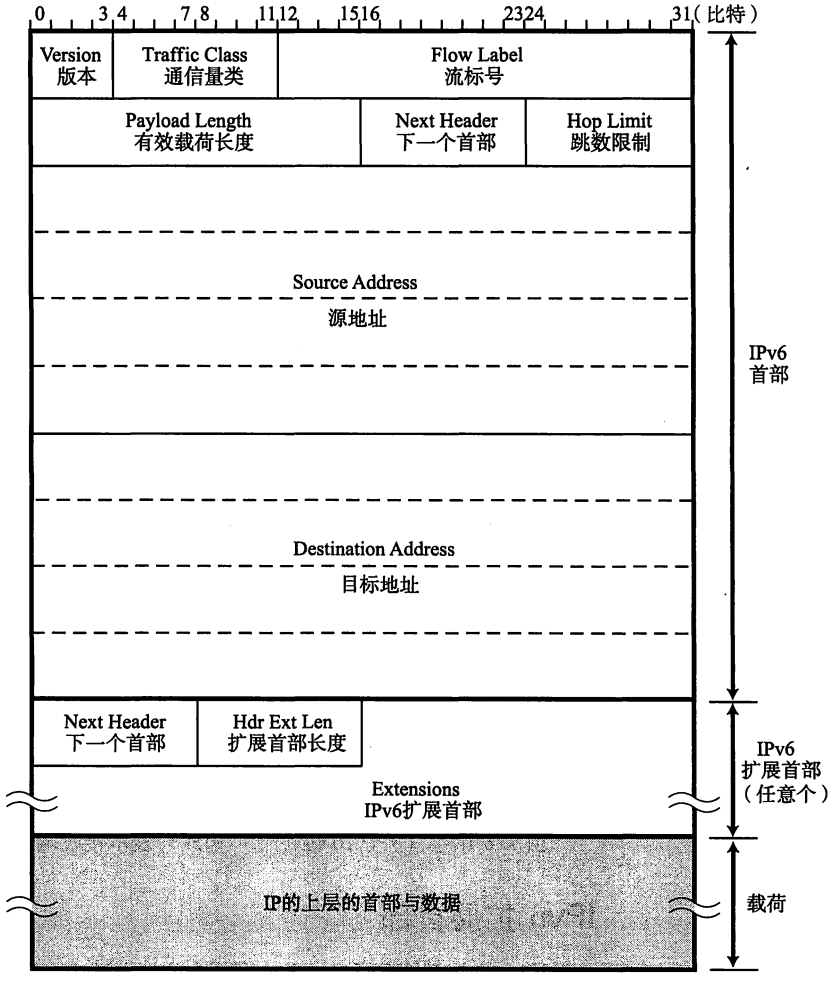



图 4.33 IPv6 数据报格式

### 版本（Version）

与 IPv4 一样，由 4 比特构成。IPv6 其版本号为 6，因此在这个字段上的值为“6”。

### 通信量类（Traffic Class）

相当于 IPv4 的 TOS（Type Of Service）字段，也由 8 比特构成。由于 TOS 在 IPv4 中几乎没有什么建树，未能成为卓有成效的技术，本来计划在 IPv6 中删掉这个字段。不过，出于今后研究的考虑还是保留了该字段。具体可以参考 5.8.3 节

对 DiffServ 的说明，以及 5.8.4 节对 ECN 的详解。

■ 流标号（Flow Label）

▼详见 5.8.3 节。

由 20 比特构成，准备用于服务质量（QoS：Quality Of Service）▼ 控制。使用这个字段提供怎样的服务已经成为未来研究的课题。不使用 QoS 时每一位可以全部设置为 0。

▼ RSVP 相关的更多细节，请参考 5.8.3 节中的 IntServ。

在进行服务质量控制时，将流标号设置为一个随机数，然后利用一种可以设置流的协议 RSVP（Resource Reservation Protocol）▼ 在路由器上进行 QoS 设置。当某个包在发送途中需要 QoS 时，需要附上 RSVP 预想的流标号。路由器接收到这样的 IP 包后先将流标号作为查找关键字，迅速从服务质量控制信息中查找并做相应处理▼。

▼采用 QoS 的路由器必须尽早转发所接受的包。但是由于以何种质量发送包才合适还需要检索相应的质量控制信息，因此有时可能会反而影响发送质量。而流标号正是为“高速检索”而是用的一种索引（Index）。它的值本身没有什么具体含义。

此外，只有流标号、源地址以及目标地址三项完全一致时，才被认为是一个流。

■ 有效载荷长度（Payload Length）

有效载荷是指包的数据部分。IPv4 的 TL（Total Length）是指包括首部在内的所有长度。然而 IPv6 中的这个 Playload Length 不包括首部，只表示数据部分的长度。由于 IPv6 的可选项是指连接 IPv6 首部的数据，因此当有可选项时，此处包含可选项数据的所有长度就是 Playload Length▼。

▼该字段长度为 16 比特，因此数据最大长度可达 65535 字节。不过，为了让更大的数据也能通过一个 IP 包发送出去，便增加了大型有效载荷选项（Jumbo Payload Option）。该选项长度为 32 比特。有了它 IPv6 一次可以发送最大 4G 字节的包。

■ 下一个首部（Next Header）

相当于 IPv4 中的协议字段。由 8 比特构成。通常表示 IP 的上一层协议是 TCP 或 UDP。不过在有 IPv6 扩展首部的情况下，该字段表示后面第一个扩展首部的协议类型。

■ 跳数限制（Hop Limit）

由 8 比特构成。与 IPv4 中的 TTL 意思相同。为了强调“可通过路由器个数”这个概念，才将名字改成了“Hop Limit”。数据每经过一次路由器就减 1，减到 0 则丢弃数据。

■ 源地址（Source Address）

由 128 比特（8 个 16 位字节）构成。表示发送端 IP 地址。

■ 目标地址（Destination Address）

由 128 比特（8 个 16 位字节）构成。表示接收端 IP 地址。

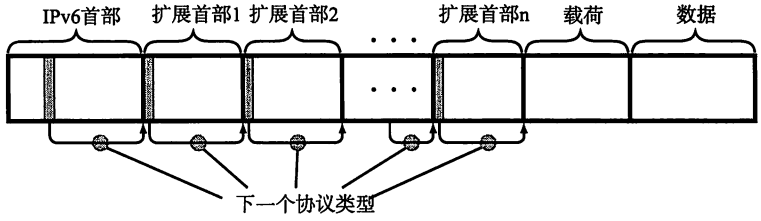
## IPv6 扩展首部

IPv6 的首部长度固定，无法将可选项加入其中。取而代之的是通过扩展首部对功能进行了有效扩展。

扩展首部通常介于 IPv6 首部与 TCP/UDP 首部中间。在 IPv4 中可选项长度固定为 40 字节，但是在 IPv6 中没有这样的限制。也就是说，IPv6 的扩展首部可以是任意长度。扩展首部当中还可以包含扩展首部协议以及下一个扩展首部字段。

IPv6 首部中没有标识以及标志字段，在需要对 IP 数据报进行分片时，可以使用扩展首部。

图 4.34
IPv6 扩展首部




具体的扩展首部如表 4.9 所示。当需要对 IPv6 的数据报进行分片时，可以设置为扩展域为 44（Fragemant Header）。使用 IPsec 时，可以使用 50、51 的 ESP、AH。Mobile IPv6 的情况下可以采用 60 与 135 的目标地址选项与移动首部。

表 4.9
IPv6 扩展首部与协议号

| 扩展首部 | 协议号 |
|---|---|
| IPv6 逐跳选项（HOPOPT） | 0 |
| IPv6 路由标头（IPv6-Route） | 43 |
| IPv6 片首部（IPv6-Frag） | 44 |
| 载荷加密（ESP） | 50 |
| 认证首部（AH） | 51 |
| 首部终止（IPv6-NoNxt） | 59 |
| 目标地址选项（IPv6-Opts） | 60 |
| 移动首部（Mobility Header） | 135 |