

第3章 IP：网际协议

3.1 引言

IP是TCP/IP协议族中最为核心的协议。所有的TCP、UDP、ICMP及IGMP数据都以IP数据报格式传输（见图1-4）。许多刚开始接触TCP/IP的人对IP提供不可靠、无连接的数据报传送服务感到很奇怪，特别是那些具有X.25或SNA背景知识的人。

不可靠（unreliable）的意思是它不能保证IP数据报能成功地到达目的地。IP仅提供最好的传输服务。如果发生某种错误时，如某个路由器暂时用完了缓冲区，IP有一个简单的错误处理算法：丢弃该数据报，然后发送ICMP消息报给信源端。任何要求的可靠性必须由上层来提供（如TCP）。

无连接（connectionless）这个术语的意思是IP并不维护任何关于后续数据报的状态信息。每个数据报的处理是相互独立的。这也说明，IP数据报可以不按发送顺序接收。如果一信源向相同的信宿发送两个连续的数据报（先是A，然后是B），每个数据报都是独立地进行路由选择，可能选择不同的路线，因此B可能在A到达之前先到达。

在本章，我们将简要介绍IP首部中的各个字段，讨论IP路由选择和子网的有关内容。还要介绍两个有用的命令：ifconfig和netstat。关于IP首部中一些字段的细节，将留在以后使用这些字段的时候再进行讨论。RFC 791[Postel 1981a]是IP的正式规范文件。

3.2 IP首部

IP数据报的格式如图3-1所示。普通的IP首部长为20个字节，除非含有选项字段。

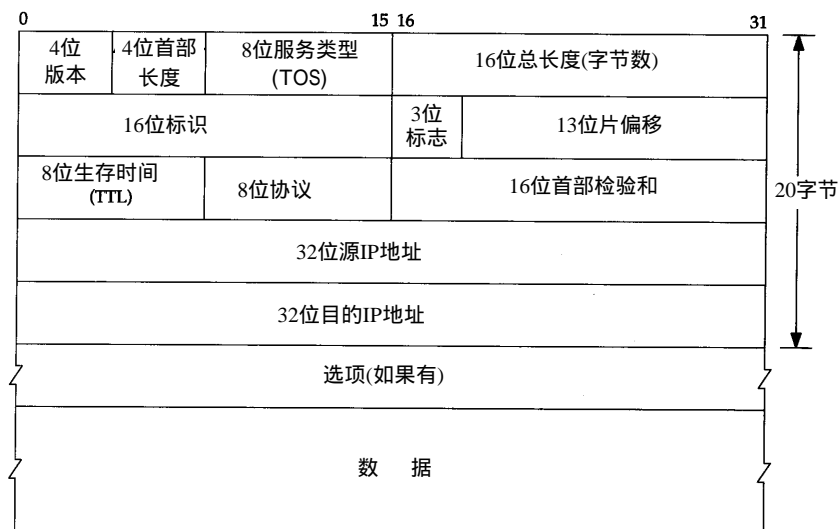


图3-1 IP数据报格式及首部中的各字段

分析图3-1中的首部。最高位在左边，记为0 bit；最低位在右边，记为31 bit。

4个字节的32 bit值以下的次序传输：首先是0~7 bit，其次8~15 bit，然后16~23 bit，最后是24~31 bit。这种传输次序称作big endian字节序。由于TCP/IP首部中所有的二进制整数在网络中传输时都要求以这种次序，因此它又称作网络字节序。以其他形式存储二进制整数的机器，如little endian格式，则必须在传输数据之前把首部转换成网络字节序。

目前的协议版本号是4，因此IP有时也称作IPv4。3.10节将对一种新版的IP协议进行讨论。

首部长度指的是首部占32 bit字的数目，包括任何选项。由于它是一个4比特字段，因此首部最长为60个字节。在第8章中，我们将看到这种限制使某些选项如路由记录选项在当今已没有什么用处。普通IP数据报（没有任何选择项）字段的值是5。

服务类型（TOS）字段包括一个3 bit的优先权子字段（现在已被忽略），4 bit的TOS子字段和1 bit未用位但必须置0。4 bit的TOS分别代表：最小时延、最大吞吐量、最高可靠性和最小费用。4 bit中只能置其中1 bit。如果所有4 bit均为0，那么就意味着是一般服务。RFC 1340 [Reynolds and Postel 1992]描述了所有的标准应用如何设置这些服务类型。RFC 1349 [Almquist 1992]对该RFC进行了修正，更为详细地描述了TOS的特性。

图3-2列出了对不同应用建议的TOS值。在最后一列中给出的是十六进制值，因为这就是在后面将要看到的tcpdump命令输出。

| 应用程序 | 最小时延 | 最大吞吐量 | 最高可靠性 | 最小费用 | 16进制值 |
|---------------|------|-------|-------|------|-------|
| Telnet/Rlogin | 1 | 0 | 0 | 0 | 0x10 |
| FTP | | | | | |
| 控制 | 1 | 0 | 0 | 0 | 0x10 |
| 数据 | 0 | 1 | 0 | 0 | 0x08 |
| 任意块数据 | 0 | 1 | 0 | 0 | 0x08 |
| TFTP | 1 | 0 | 0 | 0 | 0x10 |
| SMTP | | | | | |
| 命令阶段 | 1 | 0 | 0 | 0 | 0x10 |
| 数据阶段 | 0 | 1 | 0 | 0 | 0x08 |
| DNS | | | | | |
| UDP查询 | 1 | 0 | 0 | 0 | 0x10 |
| TCP查询 | 0 | 0 | 0 | 0 | 0x00 |
| 区域传输 | 0 | 1 | 0 | 0 | 0x08 |
| ICMP | | | | | |
| 差错 | 0 | 0 | 0 | 0 | 0x00 |
| 查询 | 0 | 0 | 0 | 0 | 0x00 |
| 任何IGP | 0 | 0 | 1 | 0 | 0x04 |
| SNMP | 0 | 0 | 1 | 0 | 0x04 |
| BOOTP | 0 | 0 | 0 | 0 | 0x00 |
| NNTP | 0 | 0 | 0 | 1 | 0x02 |

图3-2 服务类型字段推荐值

Telnet和Rlogin这两个交互应用要求最小的传输时延，因为人们主要用它们来传输少量的交互数据。另一方面，FTP文件传输则要求有最大的吞吐量。最高可靠性被指明给网络管理（SNMP）和路由选择协议。用户网络新闻（Usenet news, NNTP）是唯一要求最小费用的应用。

现在大多数的TCP/IP实现都不支持TOS特性，但是自4.3BSD Reno以后的新版系统都对它进行了设置。另外，新的路由协议如OSPF和IS-IS都能根据这些字段的值进行路由决策。

在2.10节中，我们提到SLIP一般提供基于服务类型的排队方法，允许对交互通信

数据在处理大块数据之前进行处理。由于大多数的实现都不使用 TOS 字段, 因此这种排队机制由 SLIP 自己来判断和处理, 驱动程序先查看协议字段 (确定是否是一个 TCP 段), 然后检查 TCP 信源和信宿的端口号, 以判断是否是一个交互服务。一个驱动程序的注释这样认为, 这种“令人厌恶的处理方法”是必需的, 因为大多数实现都不允许应用程序设置 TOS 字段。

总长度字段是指整个 IP 数据报的长度, 以字节为单位。利用首部长度字段和总长度字段, 就可以知道 IP 数据报中数据内容的起始位置和长度。由于该字段长 16 比特, 所以 IP 数据报最长可达 65535 字节 (回忆图 2-5, 超级通道的 MTU 为 65535。它的意思其实不是一个真正的 MTU——它使用了最长的 IP 数据报)。当数据报被分片时, 该字段的值也随着变化, 这一点将在 11.5 节中进一步描述。

尽管可以传送一个长达 65535 字节的 IP 数据报, 但是大多数的链路层都会对它进行分片。而且, 主机也要求不能接收超过 576 字节的数据报。由于 TCP 把用户数据分成若干片, 因此一般来说这个限制不会影响 TCP。在后面的章节中将遇到大量使用 UDP 的应用 (RIP, TFTP, BOOTP, DNS, 以及 SNMP), 它们都限制用户数据报长度为 512 字节, 小于 576 字节。但是, 事实上现在大多数的实现 (特别是那些支持网络文件系统 NFS 的实现) 允许超过 8192 字节的 IP 数据报。

总长度字段是 IP 首部中必要的内容, 因为一些数据链路 (如以太网) 需要填充一些数据以达到最小长度。尽管以太网的最小帧长为 46 字节 (见图 2-1), 但是 IP 数据可能会更短。如果没有总长度字段, 那么 IP 层就不知道 46 字节中有多少是 IP 数据报的内容。

标识字段唯一地标识主机发送的每一份数据报。通常每发送一份报文它的值就会加 1。在 11.5 节介绍分片和重组时再详细讨论它。同样, 在讨论分片时再来分析标志字段和片偏移字段。

RFC 791 [Postel 1981a] 认为标识字段应该由让 IP 发送数据报的上层来选择。假设有两个连续的 IP 数据报, 其中一个是由 TCP 生成的, 而另一个是由 UDP 生成的, 那么它们可能具有相同的标识字段。尽管这也可以照常工作 (由重组算法来处理), 但是在大多数从伯克利派生出来的系统中, 每发送一个 IP 数据报, IP 层都要把一个内核变量的值加 1, 不管交给 IP 的数据来自哪一层。内核变量的初始值根据系统引导时的时间来设置。

TTL (time-to-live) 生存时间字段设置了数据报可以经过的最多路由器数。它指定了数据报的生存时间。TTL 的初始值由源主机设置 (通常为 32 或 64), 一旦经过一个处理它的路由器, 它的值就减去 1。当该字段的值为 0 时, 数据报就被丢弃, 并发送 ICMP 报文通知源主机。第 8 章我们讨论 Traceroute 程序时将再回来讨论该字段。

我们已经在第 1 章讨论了协议字段, 并在图 1-8 中示出了它如何被 IP 用来对数据报进行分用。根据它可以识别是哪个协议向 IP 传送数据。

首部检验和字段是根据 IP 首部计算的检验和码。它不对首部后面的数据进行计算。ICMP、IGMP、UDP 和 TCP 在它们各自的首部中均含有同时覆盖首部和数据检验和码。

为了计算一份数据报的 IP 检验和, 首先把检验和字段置为 0。然后, 对首部中每个 16 bit 进行二进制反码求和 (整个首部看成是由一串 16 bit 的字组成), 结果存在检验和字段中。当收到一份 IP 数据报后, 同样对首部中每个 16 bit 进行二进制反码的求和。由于接收方在计算过

程中包含了发送方存在首部中的检验和，因此，如果首部在传输过程中没有发生任何差错，那么接收方计算的结果应该为全 1。如果结果不是全 1（即检验和错误），那么 IP 就丢弃收到的数据报。但是不生成差错报文，由上层去发现丢失的数据报并进行重传。

ICMP、IGMP、UDP 和 TCP 都采用相同的检验和算法，尽管 TCP 和 UDP 除了本身的首部和数据外，在 IP 首部中还包含不同的字段。在 RFC 1071 [Braden, Borman and Patridge 1988] 中有关于如何计算 Internet 检验和的实现技术。由于路由器经常只修改 TTL 字段（减 1），因此当路由器转发一份报文时可以增加它的检验和，而不需要对 IP 整个首部进行重新计算。RFC 1141 [Mallory and Kullberg 1990] 为此给出了一个很有效的方法。

但是，标准的 BSD 实现在转发数据报时并不是采用这种增加的办法。

每一份 IP 数据报都包含源 IP 地址和目的 IP 地址。我们在 1.4 节中说过，它们都是 32 bit 的值。

最后一个字段是任选项，是数据报中的一个可变长的可选信息。目前，这些任选项定义如下：

- 安全和处理限制（用于军事领域，详细内容参见 RFC 1108 [Kent 1991]）
- 记录路径（让每个路由器都记下它的 IP 地址，见 7.3 节）
- 时间戳（让每个路由器都记下它的 IP 地址和时间，见 7.4 节）
- 宽松的源站选路（为数据报指定一系列必须经过的 IP 地址，见 8.5 节）
- 严格的源站选路（与宽松的源站选路类似，但是要求只能经过指定的这些地址，不能经过其他的地址）。

这些选项很少被使用，并非所有的主机和路由器都支持这些选项。

选项字段一直都是以 32 bit 作为界限，在必要的时候插入值为 0 的填充字节。这样就保证 IP 首部始终是 32 bit 的整数倍（这是首部长度的要求）。

3.3 IP 路由选择

从概念上说，IP 路由选择是简单的，特别对于主机来说。如果目的主机与源主机直接相连（如点对点链路）或都在一个共享网络上（以太网或令牌环网），那么 IP 数据报就直接送到目的主机上。否则，主机把数据报发往一默认的路由器上，由路由器来转发该数据报。大多数的主机都是采用这种简单机制。

在本节和第 9 章中，我们将讨论更一般的情况，即 IP 层既可以配置成路由器的功能，也可以配置成主机的功能。当今的大多数多用户系统，包括几乎所有的 Unix 系统，都可以配置成一个路由器。我们可以为它指定主机和路由器都可以使用的简单路由算法。本质上的区别在于主机从不把数据报从一个接口转发到另一个接口，而路由器则要转发数据报。内含路由器功能的主机应该从不转发数据报，除非它被设置成那样。在 9.4 小节中，我们将进一步讨论配置的有关问题。

在一般的体制中，IP 可以从 TCP、UDP、ICMP 和 IGMP 接收数据报（即在本地生成的数据报）并进行发送，或者从一个网络接口接收数据报（待转发的数据报）并进行发送。IP 层在内存中有一个路由表。当收到一份数据报并进行发送时，它都要对该表搜索一次。当数据报来自某个网络接口时，IP 首先检查目的 IP 地址是否为本机的 IP 地址之一或者 IP 广播地址。如果确实是这样，数据报就被送到由 IP 首部协议字段所指定的协议模块进行处理。如果数据报的

目的不是这些地址, 那么 (1) 如果IP层被设置为路由器的功能, 那么就对数据报进行转发 (也就是说, 像下面对待发出的数据报一样处理) ; 否则 (2) 数据报被丢弃。

路由表中的每一项都包含下面这些信息:

- 目的IP地址。它既可以是一个完整的主机地址, 也可以是一个网络地址, 由该表目中的标志字段来指定 (如下所述)。主机地址有一个非0的主机号 (见图1-5), 以指定某一特定的主机, 而网络地址中的主机号为0, 以指定网络中的所有主机 (如以太网, 令牌环网)。
- 下一站 (或下一跳) 路由器 (next-hop router) 的IP地址, 或者有直接连接的网络IP地址。下一站路由器是指一个在直接相连网络上的路由器, 通过它可以转发数据报。下一站路由器不是最终的目的, 但是它可以把传送给它的数据报转发到最终目的。
- 标志。其中一个标志指明目的IP地址是网络地址还是主机地址, 另一个标志指明下一站路由器是否为真正的下一站路由器, 还是一个直接相连的接口 (我们将在 9.2节中详细介绍这些标志)。
- 为数据报的传输指定一个网络接口。

IP路由选择是逐跳地 (hop-by-hop) 进行的。从这个路由表信息可以看出, IP并不知道到达任何目的的完整路径 (当然, 除了那些与主机直接相连的目的)。所有的IP路由选择只为数据报传输提供下一站路由器的IP地址。它假定下一站路由器比发送数据报的主机更接近目的, 而且下一站路由器与该主机是直接相连的。

IP路由选择主要完成以下这些功能:

- 1) 搜索路由表, 寻找能与目的IP地址完全匹配的表目 (网络号和主机号都要匹配)。如果找到, 则把报文发送给该表目指定的下一站路由器或直接连接的网络接口 (取决于标志字段的值)。
- 2) 搜索路由表, 寻找能与目的网络号相匹配的表目。如果找到, 则把报文发送给该表目指定的下一站路由器或直接连接的网络接口 (取决于标志字段的值)。目的网络上的所有主机都可以通过这个表目来处置。例如, 一个以太网上的所有主机都是通过这种表目进行寻径的。
这种搜索网络的匹配方法必须考虑可能的子网掩码。关于这一点我们在下一节中进行讨论。
- 3) 搜索路由表, 寻找标为“默认 (default)”的表目。如果找到, 则把报文发送给该表目指定的下一站路由器。

如果上面这些步骤都没有成功, 那么该数据报就不能被传送。如果不能传送的数据报来自本机, 那么一般会向生成数据报的应用程序返回一个“主机不可达”或“网络不可达”的错误。

完整主机地址匹配在网络号匹配之前执行。只有当它们都失败后才选择默认路由。默认路由, 以及下一站路由器发送的ICMP间接报文 (如果我们为数据报选择了错误的默认路由), 是IP路由选择机制中功能强大的特性。我们在第9章对它们进行讨论。

为一个网络指定一个路由器, 而不必为每个主机指定一个路由器, 这是IP路由选择机制的另一个基本特性。这样做可以极大地缩小路由表的规模, 比如Internet上的路由器有只有几千个表目, 而不会是超过100万个表目。

举例

首先考虑一个简单的例子: 我们的主机bsdi有一个IP数据报要发送给主机sun。双方都在

同一个以太网上（参见扉页前图）。数据报的传输过程如图 3-3 所示。

当 IP 从某个上层收到这份数据报后，它搜索路由表，发现目的 IP 地址（140.252.13.33）在一个直接相连的网络上（以太网 140.252.13.0）。于是，在表中找到匹配网络地址（在下一节中，我们将看到，由于以太网的子网掩码的存在，实际的网络地址是 140.252.13.32，但是这并不影响这里所讨论的路由选择）。

数据报被送到以太网驱动程序，然后作为一个以太网数据帧被送到 sun 主机上（见图 2-1）。IP 数据报中的目的地址是 sun 的 IP 地址（140.252.13.33），而在链路层首部中的目的地址是 48 bit 的 sun 主机的以太网接口地址。这个 48 bit 的以太网地址是用 ARP 协议获得的，我们将在下一章对此进行描述。

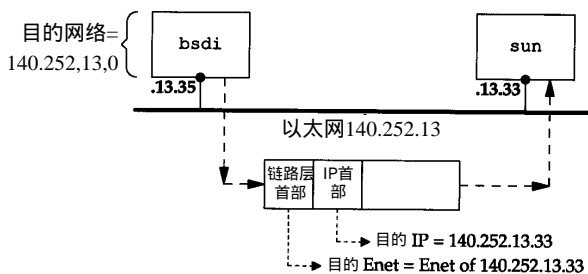


图3-3 数据报从主机bsdi到sun的传送过程

现在来看另一个例子：主机 bsdi 有一份 IP 数据报要传到 ftp.uu.net 主机上，它的 IP 地址是 192.48.96.9。经过的前三个路由器如图 3-4 所示。首先，主机 bsdi 搜索路由表，但是没有找到与主机地址或网络地址相匹配的表目，因此只能用默认的表目，把数据报传给下一站路由器，即主机 sun。当数据报从 bsdi 被传到 sun 主机上以后，目的 IP 地址是最终的信宿机地址（192.48.96.9），但是链路层地址却是 sun 主机的以太网接口地址。这与图 3-3 不同，在那里数据报中的目的 IP 地址和目的链路层地址都指的是相同的主机（sun）。

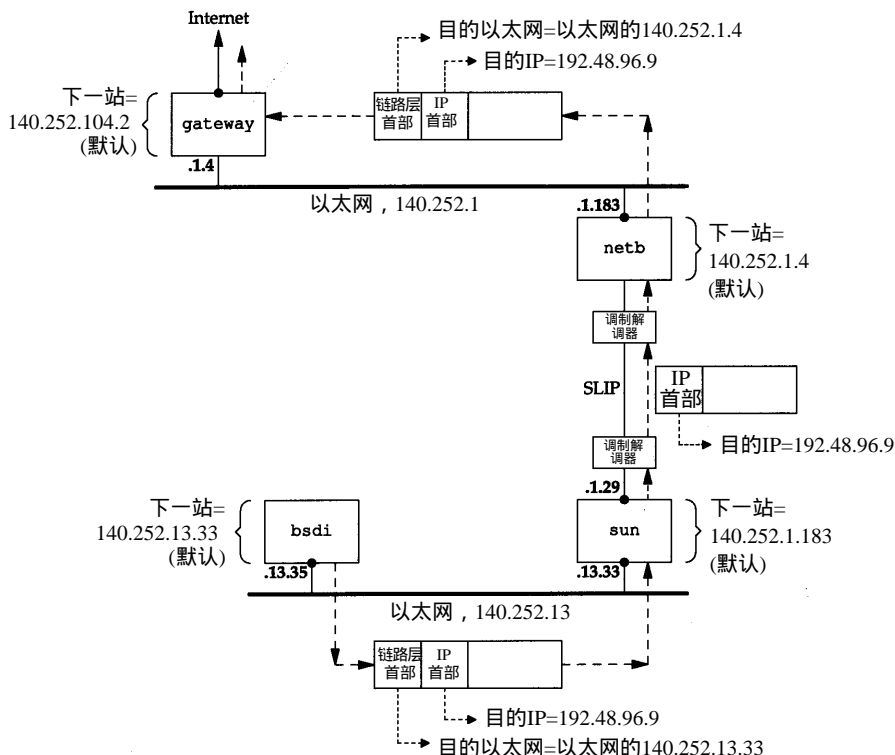


图3-4 从bsdi到ftp.uu.net (192.48.96.9)的初始路径

当sun收到数据报后, 它发现数据报的目的IP地址并不是本机的任一地址, 而sun已被设置成具有路由器的功能, 因此它把数据报进行转发。经过搜索路由表, 选用了默认表目。根据sun的默认表目, 它把数据报转发到下一站路由器netb, 该路由器的地址是140.252.1.183。数据报是经过点对点SLIP链路被传送的, 采用了图2-2所示的最小封装格式。这里, 我们没有给出像以太网链路层数据帧那样的首部, 因为在SLIP链路中没有那样的首部。

当netb收到数据报后, 它执行与sun主机相同的步骤: 数据报的目的地址不是本机地址, 而netb也被设置成具有路由器的功能, 于是它也对数据报进行转发。采用的也是默认路由表目, 把数据报送到下一站路由器gateway (140.252.1.4)。位于以太网140.252.1上的主机netb用ARP获得对应于140.252.1.4的48 bit以太网地址。这个以太网地址就是链路层数据帧头上的目的地址。

路由器gateway也执行与前面两个路由器相同的步骤。它的默认路由表目所指定的下一站路由器IP地址是140.252.104.2 (我们将在图8-4中证实, 使用Traceroute程序时, 它就是gateway使用的下一站路由器)。

对于这个例子需要指出一些关键点:

1) 该例子中的所有主机和路由器都使用了默认路由。事实上, 大多数主机和一些路由器可以用默认路由来处理任何目的, 除非它在本地局域网上。

2) 数据报中的目的IP地址始终不发生任何变化 (在8.5节中, 我们将看到, 只有使用源路由选项时, 目的IP地址才有可能被修改, 但这种情况很少出现)。所有的路由选择决策都是基于这个目的IP地址。

3) 每个链路层可能具有不同的数据帧首部, 而且链路层的目的地址 (如果有的话) 始终指的是下一站的链路层地址。在例子中, 两个以太网封装了含有下一站以太网地址的链路层首部, 但是SLIP链路没有这样做。以太网地址一般通过ARP获得。

在第9章, 我们在描述了ICMP之后将再次讨论IP路由选择问题。我们将看到一些路由表的例子, 以及如何用它们来进行路由决策的。

3.4 子网寻址

现在所有的主机都要求支持子网编址 (RFC 950 [Mogul and Postel 1985])。不是把IP地址看成由单纯的一个网络号和一个主机号组成, 而是把主机号再分成一个子网号和一个主机号。

这样做的原因是因为A类和B类地址为主机号分配了太多的空间, 可分别容纳的主机数为 $2^{24}-2$ 和 $2^{16}-2$ 。事实上, 在一个网络中人们并不安排这么多的主机 (各类IP地址的格式如图1-5所示)。由于全0或全1的主机号都是无效的, 因此我们把总数减去2。

在InterNIC获得某类IP网络号后, 就由当地的系统管理员来进行分配, 由他 (或她) 来决定是否建立子网, 以及分配多少比特给子网号和主机号。例如, 这里有一个B类网络地址 (140.252), 在剩下的16 bit中, 8 bit用于子网号, 8 bit用于主机号, 格式如图3-5所示。这样就允许有254个子网, 每个子网可以有254台主机。



图3-5 B类地址的一种子网编址

许多管理员采用自然的划分方法，即把 B 类地址中留给主机的 16 bit 中的前 8 bit 作为子网地址，后 8 bit 作为主机号。这样用点分十进制方法表示的 IP 地址就可以比较容易确定子网号。但是，并不要求 A 类或 B 类地址的子网划分都要以字节为划分界限。

大多数的子网例子都是 B 类地址。其实，子网还可用于 C 类地址，只是它可用的比特数较少而已。很少出现 A 类地址的子网例子是因为 A 类地址本身就很少（但是，大多数 A 类地址都是进行子网划分的）。

子网对外部路由器来说隐藏了内部网络组织（一个校园或公司内部）的细节。在我们的网络例子中，所有的 IP 地址都有一个 B 类网络号 140.252。但是其中有超过 30 个子网，多于 400 台主机分布在这些子网中。由一台路由器提供了 Internet 的接入，如图 3-6 所示。

在这个图中，我们把大多数的路由器编号为 R_n ， n 是子网号。我们给出了连接这些子网的路由器，同时还包括了扉页前图中的九个系统。在图中，以太网用粗线表示，点对点链路用虚线表示。我们没有画出不同子网中的所有主机。例如，在子网 140.252.3 上，就超过 50 台主机，而在子网 140.252.1 上则超过 100 台主机。

与 30 个 C 类地址相比，用一个包含 30 个子网的 B 类地址的好处是，它可以缩小 Internet 路由表的规模。B 类地址 140.252 被划分为若干子网的事实对于所有子网以外的 Internet 路由器都是透明的。为了到达 IP 地址开始部分为 140.252 的主机，外部路由器只需要知道通往 IP 地址 140.252.104.1 的路径。这就是说，对于网络 140.252 只需一个路由表目，而如果采用 30 个 C 类地址，则需要 30 个路由表目。因此，子网划分缩减了路由表的规模（在 10.8 小节中，我们将介绍一种新技术，即使用 C 类地址也可以缩减路由表的规模）。

子网对于子网内部的路由器是不透明的。如图 3-6 所示，一份来自 Internet 的数据报到达 gateway，它的目的地址是 140.252.57.1。路由器 gateway 需要知道子网号是 57，然后把它送到 kpno。同样，kpno 必须把数据报送到 R55，最后由 R55 把它送到 R57。

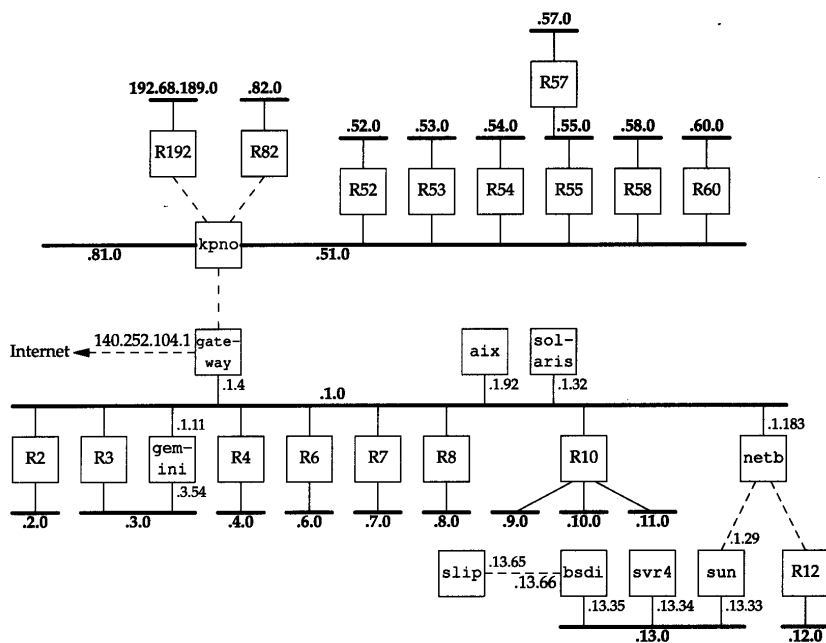


图3-6 网络noao.edu (140.252) 中的大多数子网安排

3.5 子网掩码

任何主机在引导时进行的部分配置是指定主机 IP 地址。大多数系统把 IP 地址存在一个磁盘文件里供引导时读用。在第 5 章我们将讨论一个无盘系统如何在引导时获得 IP 地址。

除了 IP 地址以外, 主机还需要知道有多少比特用于子网号及多少比特用于主机号。这是在引导过程中通过子网掩码来确定的。这个掩码是一个 32 bit 的值, 其中值为 1 的比特留给网络号和子网号, 为 0 的比特留给主机号。图 3-7 是一个 B 类地址的两种不同的子网掩码格式。第一个例子是 noao.edu 网络采用的子网划分方法, 如图 3-5 所示, 子网号和主机号都是 8 bit 宽。第二个例子是一个 B 类地址划分成 10 bit 的子网号和 6 bit 的主机号。

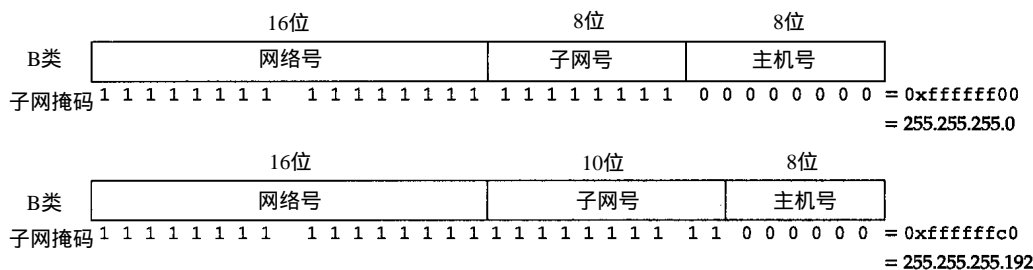


图 3-7 两种不同的 B 类地址子网掩码的例子

尽管 IP 地址一般以点分十进制方法表示, 但是子网掩码却经常用十六进制来表示, 特别是当界限不是一个字节时, 因为子网掩码是一个比特掩码。

给定 IP 地址和子网掩码以后, 主机就可以确定 IP 数据报的目的是: (1) 本子网上的主机; (2) 本网络中其他子网中的主机; (3) 其他网络上的主机。如果知道本机的 IP 地址, 那么就知道它是否为 A 类、B 类或 C 类地址 (从 IP 地址的高位可以得知), 也就知道网络号和子网号之间的分界线。而根据子网掩码就可知道子网号与主机号之间的分界线。

举例

假设我们的主机地址是 140.252.1.1 (一个 B 类地址), 而子网掩码为 255.255.255.0 (其中 8 bit 为子网号, 8 bit 为主机号)。

- 如果目的 IP 地址是 140.252.4.5, 那么我们就知道 B 类网络号是相同的 (140.252), 但是子网号是不同的 (1 和 4)。用子网掩码在两个 IP 地址之间的比较如图 3-8 所示。
- 如果目的 IP 地址是 140.252.1.22, 那么 B 类网络号还是一样的 (140.252), 而且子网号也是一样的 (1), 但是主机号是不同的。
- 如果目的 IP 地址是 192.43.235.6 (一个 C 类地址), 那么网络号是不同的, 因而进一步的比较就不用再进行了。



图 3-8 使用子网掩码的两个 B 类地址之间的比较

给定两个IP地址和子网掩码后，IP路由选择功能一直进行这样的比较。

3.6 特殊情况的IP地址

经过子网划分的描述，现在介绍 7 个特殊的IP地址，如图 3-9 所示。在这个图中，0 表示所有的比特位全为 0；-1 表示所有的比特位全为 1；netid、subnetid 和 hostid 分别表示不为全 0 或全 1 的对应字段。子网号栏为空表示该地址没有进行子网划分。

| IP 地 址 | | | 可 以 为 | | 描 述 |
|--------|----------|-----|-------|-----|---------------------------|
| 网络号 | 子网号 | 主机号 | 源 端 | 目的端 | |
| 0 | | 0 | OK | 不可能 | 网络上的主机（参见下面的限制） |
| 0 | | 主机号 | OK | 不可能 | 网络上的特定主机（参见下面的限制） |
| 127 | | 任何值 | OK | OK | 环回地址（2.7 节） |
| -1 | | -1 | 不可能 | OK | 受限的广播（永远不被转发） |
| netid | | -1 | 不可能 | OK | 以网络为目的向 netid 广播 |
| netid | subnetid | -1 | 不可能 | OK | 以子网为目的向 netid、subnetid 广播 |
| netid | -1 | -1 | 不可能 | OK | 以所有子网为目的向 netid 广播 |

图3-9 特殊情况的IP地址

我们把这个表分成三个部分。表的头两项是特殊的源地址，中间项是特殊的环回地址，最后四项是广播地址。

表中的头两项，网络号为 0，如主机使用 BOOTP 协议确定本机 IP 地址时只能作为初始化过程中的源地址出现。

在 12.2 节中，我们将进一步分析四类广播地址。

3.7 一个子网的例子

这个例子是本文中采用的子网，以及如何使用两个不同的子网掩码。具体安排如图 3-10 所示。

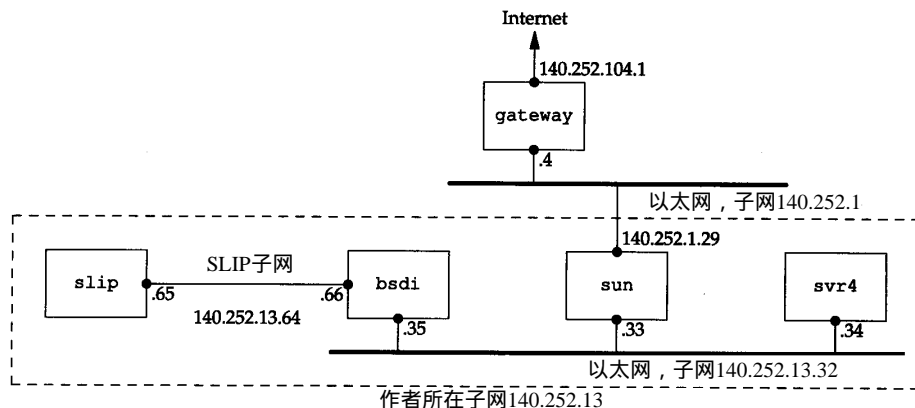


图3-10 作者所在子网中的主机和网络安排

如果把该图与扉页前图相比，就会发现在图 3-10 中省略了从路由器 sun 到上面的以太网之间的连接细节，实际上它们之间的连接是拨号 SLIP。这个细节不影响本节中讨论的子网划分

问题。我们在4.6节讨论ARP代理时将再回头讨论这个细节。

问题是在子网13中有两个分离的网络：一个以太网和一个点对点链路（硬件连接的SLIP链路）（点对点链接始终会带来问题，因为它一般在两端都需要IP地址）。将来或许会有更多的主机和网络，但是为了不让主机跨越不同的网络就得使用不同的子网号。我们的解决方法是把子网号从8 bit扩充到11bit，把主机号从8 bit减为5 bit。这就叫作变长子网，因为140.252网络中的大多数子网都采用8 bit子网掩码，而我们的子网却采用11 bit的子网掩码。

RFC 1009[Braden and Postel 1987]允许一个含有子网的网络使用多个子网掩码。新的路由器需求RFC[Almquist 1993]则要求支持这一功能。

但是，问题在于并不是所有的路由选择协议在交换目的网络时也交换子网掩码。在第10章中，我们将看到RIP不支持变长子网，RIP第2版和OSPF则支持变长子网。在我们的例子中不存在这种问题，因为在我的子网中不要求使用RIP协议。

作者子网中的IP地址结构如图3-11所示，11位子网号中的前8 bit始终是13。在剩下的3 bit中，我们用二进制001表示以太网，010表示点对点SLIP链路。这个变长子网掩码在140.252网络中不会给其他主机和路由器带来问题——只要目的是子网140.252.13的所有数据报都传给路由器sun（IP地址是140.252.1.29），如图3-11所示。如果sun知道子网13中的主机有11 bit子网号，那么一切都好办了。

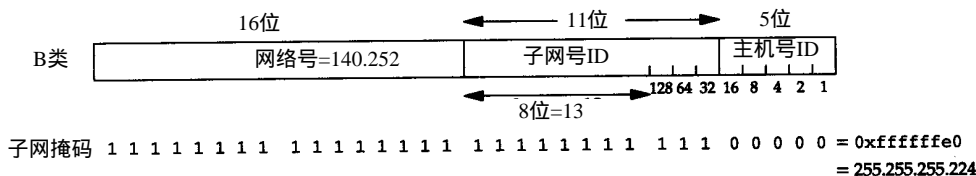


图3-11 变长子网

140.252.13子网中的所有接口的子网掩码是255.255.255.224，或0xffffffe0。这表明最右边的5 bit留给主机号，左边的27 bit留给网络号和子网号。

图3-10中所有接口的IP地址和子网掩码的分配情况如图3-12所示。

| 主机 | IP地址 | 子网掩码 | 网络号/子网号 | 主机号 | 注 释 |
|------|---------------|-----------------|---------------|-----|-------------------|
| sun | 140.252.1.29 | 255.255.255.0 | 140.252.1 | 29 | 在子网1上 在作者所在子网上 |
| | 140.252.13.33 | 255.255.255.224 | 140.252.13.32 | 1 | |
| svr4 | 140.252.13.34 | 255.255.255.224 | 140.252.13.32 | 2 | |
| bsdi | 140.252.13.35 | 255.255.255.224 | 140.252.13.32 | 3 | 在以太网上 点对点 |
| | 140.252.13.66 | 255.255.255.224 | 140.252.13.64 | 2 | |
| slip | 140.252.13.65 | 255.255.255.224 | 140.252.13.64 | 1 | 点对点 |
| | 140.252.13.63 | 255.255.255.224 | 140.252.13.32 | 31 | 以太网上的广播地址 |

图3-12 作者子网的IP地址

第1栏标为是“主机”，但是sun和bsdi也具有路由器的功能，因为它们是多接口的，可以把分组数据从一个接口转发到另一个接口。

这个表中的最后一行是图3-10中的广播地址140.252.13.63：它是根据以太网子网号（140.252.13.32）和图3-11中的低5位置1（16 + 8 + 4 + 2 + 1 = 31）得来的（我们在第12章中将看到，这个地址被称作以子网为目的的广播地址（subnet-directed broadcast address））。

3.8 ifconfig命令

到目前为止，我们已经讨论了链路层和 IP 层，现在可以介绍 TCP/IP 对网络接口进行配置和查询的命令了。ifconfig(8)命令一般在引导时运行，以配置主机上的每个接口。

由于拨号接口可能会经常接通和挂断（如 SLIP 链路），每次线路接通和挂断时，ifconfig 都必须（以某种方法）运行。这个过程如何完成取决于使用的 SLIP 软件。

下面是作者子网接口的有关参数。请把它们与图 3-12 的值进行比较。

```
sun % /usr/etc/ifconfig -a          在所有接口报告的选项
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
      inet 140.252.13.33 netmask fffffffe0 broadcast 140.252.13.63
sl0: flags=1051<UP,POINTOPOINT,RUNNING,LINK0>
      inet 140.252.1.29 --> 140.252.1.183 netmask fffffff00
lo0: flags=49<UP,LOOPBACK,RUNNING>
      inet 127.0.0.1 netmask ff000000
```

环回接口（2.7节）被认为是一个网络接口。它是一个 A 类地址，没有进行子网划分。

需要注意的是以太网没有采用尾部封装（2.3节），而且可以进行广播，而 SLIP 链路是一个点对点的链接。

SLIP 接口的标志 LINK0 是一个允许压缩 slip 的数据（CSLIP，参见 2.5 节）的配置选项。其他的选项有 LINK1（如果从另一端收到一份压缩报文，就允许采用 CSLIP）和 LINK2（所有外出的 ICMP 报文都被丢弃）。我们在 4.6 节中将讨论 SLIP 链接的目的地址。

安装指南中的注释对最后这个选项进行了解释：“一般它不应设置，但是由于一些不当的 ping 操作，可能会导致吞吐量降到 0。”

bsdi 是另一台路由器。由于 -a 参数是 SunOS 操作系统具有的功能，因此我们必须多次执行 ifconfig，并指定接口名字参数：

```
bsdi % /sbin/ifconfig we0
we0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX>
      inet 140.252.13.35 netmask fffffffe0 broadcast 140.252.13.63
bsdi % /sbin/ifconfig sl0
sl0: flags=1011<UP,POINTOPOINT,LINK0>
      inet 140.252.13.66 --> 140.252.13.65 netmask fffffffe0
```

这里，我们看到以太网接口（we0）的一个新选项：SIMPLEX。这个 4.4BSD 标志表明接口不能收到本机传送的数据。在 BSD/386 中所有的以太网都这样设置。一旦这样设置后，如果接口发送一帧数据到广播地址，那么就会为本机拷贝一份数据送到环回地址（在 6.3 小节我们将举例说明这一点）。

在主机 slip 中，SLIP 接口的设置基本上与上面的 bsdi 一致，只是两端的 IP 地址进行了互换：

```
slip % /sbin/ifconfig sl0
sl0: flags=1011<UP,POINTOPOINT,LINK0>
      inet 140.252.13.65 --> 140.252.13.66 netmask fffffffe0
```

最后一个接口是主机 svr4 上的以太网接口。它与前面的以太网接口类似，只是 SVR4 版的 ifconfig 没有打印 RUNNING 标志：

```
svr4 % /usr/sbin/ifconfig emd0
emd0: flags=23<UP,BROADCAST,NOTRAILERS>
      inet 140.252.13.34 netmask fffffffe0 broadcast 140.252.13.63
```

ifconfig命令一般支持TCP/IP以外的其他协议族,而且有很多参数。关于这些细节可以查看系统说明书。

3.9 netstat命令

netstat(1)命令也提供系统上的接口信息。-i参数将打印出接口信息, -n参数则打印出IP地址,而不是主机名字。

```
sun % netstat -in
Name Mtu Net/Dest Address IpKts Ierrs OpKts Oerrs Collis Queue
le0 1500 140.252.13.32 140.252.13.33 67719 0 92133 0 1 0
sl0 552 140.252.1.183 140.252.1.29 48035 0 54963 0 0 0
lo0 1536 127.0.0.0 127.0.0.1 15548 0 15548 0 0 0
```

这个命令打印出每个接口的MTU、输入分组数、输入错误、输出分组数、输出错误、冲突以及当前的输出队列长度。

在第9章将用netstat命令检查路由表,那时再回头讨论该命令。另外,在第13章将用它的一个改进版本来查看活动的广播组。

3.10 IP的未来

IP主要存在三个方面的问题。这是Internet在过去几年快速增长所造成的结果(参见习题1.2)。

- 1) 超过半数的B类地址已被分配。根据估计,它们大约在1995年耗尽。
- 2) 32 bit的IP地址从长期的Internet增长角度来看,一般是不够用的。
- 3) 当前的路由结构没有层次结构,属于平面型(flat)结构,每个网络都需要一个路由表目。随着网络数目的增长,一个具有多个网络的网站就必须分配多个C类地址,而不是一个B类地址,因此路由表的规模会不断增长。

无类别的域间路由选择CIDR(Classless Interdomain Routing)提出了一个可以解决第三个问题的建议,对当前版本的IP(IP版本4)进行扩充,以适应21世纪Internet的发展。对此我们将在10.8节进一步详细介绍。

对新版的IP,即下一代IP,经常称作IPng,主要有四个方面的建议。1993年5月发行的IEEE Network(vol.7, no.3)对前三个建议进行了综述,同时有一篇关于CIDR的论文。RFC 1454[Dixon 1993]对前三个建议进行了比较。

1) SIP,简单Internet协议。它针对当前的IP提出了一个最小幅度的修改建议,采用64位地址和一个不同的首部格式(首部的前4比特仍然包含协议的版本号,其值不再是4)。

2) PIP。这个建议也采用了更大的、可变长度的和有层次结构的地址,而且首部格式也不相同。

3) TUBA,代表“TCP and UDP with Bigger Address”,它基于OSI的CLNP(Connectionless Network Protocol,无连接网络协议),一个与IP类似的OSI协议。它提供大得多的地址空间:可变长度,可达20个字节。由于CLNP是一个现有的协议,而SIP和PIP只是建议,因此关于CLNP的文档已经出现。RFC 1347[Callon 1992]提供了TUBA的有关细节。文献[Perlman 1992]的第7章对IPv4和CLNP进行了比较。许多路由器已经支持CLNP,但是很少有主机也提供支持。

4) TP/IX，由RFC 1475 [Ullmann 1993]对它进行了描述。虽然SIP采用了64 bit的址址，但是它还改变了TCP和UDP的格式：两个协议均为32 bit的端口号，64 bit的序列号，64 bit的确认号，以及TCP的32 bit窗口。

前三个建议基本上采用了相同版本的TCP和UDP作为传输层协议。

由于四个建议只能有一个被选为IPv4的替换者，而且在你读到此书时可能已经做出选择，因此我们对它们不进行过多评论。虽然CIDR即将实现以解决目前的短期问题，但是IPv4后继者的实现则需要经过许多年。

3.11 小结

本章开始描述了IP首部的格式，并简要讨论了首部中的各个字段。我们还介绍了IP路由选择，并指出主机的路由选择可以非常简单：如果目的主机在直接相连的网络上，那么就把数据报直接传给目的主机，否则传给默认路由器。

在进行路由选择决策时，主机和路由器都使用路由表。在表中有三种类型的路由：特定主机型、特定网络型和默认路由型。路由表中的表目具有一定的优先级。在选择路由时，主机路由优先于网络路由，最后在没有其他可选路由存在时才选择默认路由。

IP路由选择是通过逐跳来实现的。数据报在各站的传输过程中目的IP地址始终不变，但是封装和目的链路层地址在每一站都可以改变。大多数的主机和许多路由器对于非本地网络的数据报都使用默认的下一站路由器。

A类和B类地址一般都要进行子网划分。用于子网号的比特数通过子网掩码来指定。我们为此举了一个实例来详细说明，即作者所在的子网，并介绍了变长子网的概念。子网的划分缩小了Internet路由表的规模，因为许多网络经常可以通过单个表目就可以访问了。接口和网络的有关信息通过ifconfig和netstat命令可以获得，包括接口的IP地址、子网掩码、广播地址以及MTU等。

在本章的最后，我们对Internet协议族潜在的改进建议——下一代IP进行了讨论。

习题

- 3.1 环回地址必须是127.0.0.1吗？
- 3.2 在图3-6中指出有两个网络接口的路由器。
- 3.3 子网号为16 bit的A类地址与子网号为8 bit的B类地址的子网掩码有什么不同？
- 3.4 阅读RFC 1219 [Tsuchiya 1991]，学习分配子网号和主机号的有关推荐技术。
- 3.5 子网掩码255.255.0.255是否对A类地址有效？
- 3.6 你认为为什么3.9小节中打印出来的环回接口的MTU要设置为1536？
- 3.7 TCP/IP协议族是基于一种数据报的网络技术，即IP层，其他的协议族则基于面向连接的网络技术。阅读文献[Clark 1988]，找出数据报网络层提供的三个优点。