第 3 章　TCP/IP 与网络互连

TCP/IP 协议栈可以分为应用层、传输层、网际层和网络接口层 4 层。其核心是 TCP/IP。本章从应用的角度，介绍它们的细节，并将之分为 TCP/UDP、IP、ICMP、路由协议和网络接口 5 部分。图 3.1 为对于图 1.55 的细化。

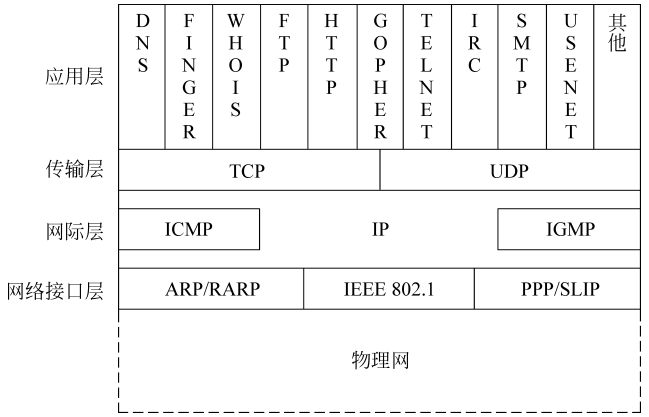


图 3.1　细化的 TCP/IP 参考模型

传输层和 IP 层是 TCP/IP 体系的核心，也是本章的主要内容。图 3.2 为它们在 TCP/IP 体系中的位置和基本职责：传输层提供应用进程之间的逻辑通信，IP 层提供互联网络中两台主机之间的通信，尽最大努力为传输层提供服务。

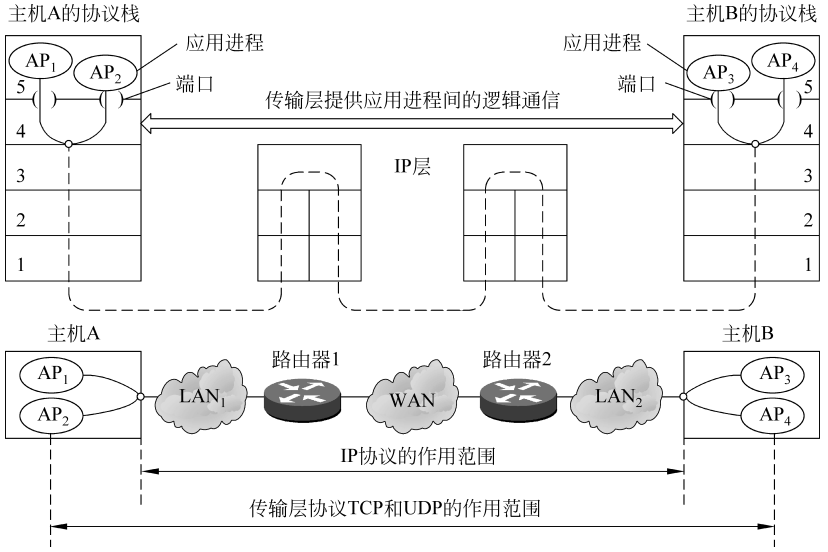


图 3.2　TCP/IP 体系中的传输层与 IP 层模型

3.1 TCP/UDP 协议

在 TCP/IP 网络体系中，TCP/UDP 层是应用层与通信子网之间的接口。它将应用层的数据提交到网络上，并抽象为进程之间的端到端的传送。这种端到端的传输具有 3 个重要特征：

（1）应用层的应用进程很多，为了区分不同的应用进程，在传输层使用了端口（port）的概念。用端口号区分和识别不同的应用协议。

（2）为了在网络上有效地传输，在传输层将对应用层报文进行分段，并形成两种传输模式：TCP（Transmission Control Protocol，运输控制协议）和 UDP（User Datagram Protocol，用户数据报协议）。TCP 是建立在虚电路基础上的有连接端对端可靠传输，UDP 是建立数据报基础上的无连接端对端传输。

（3）如图 3.3 所示，在传输层具有多路复用功能。

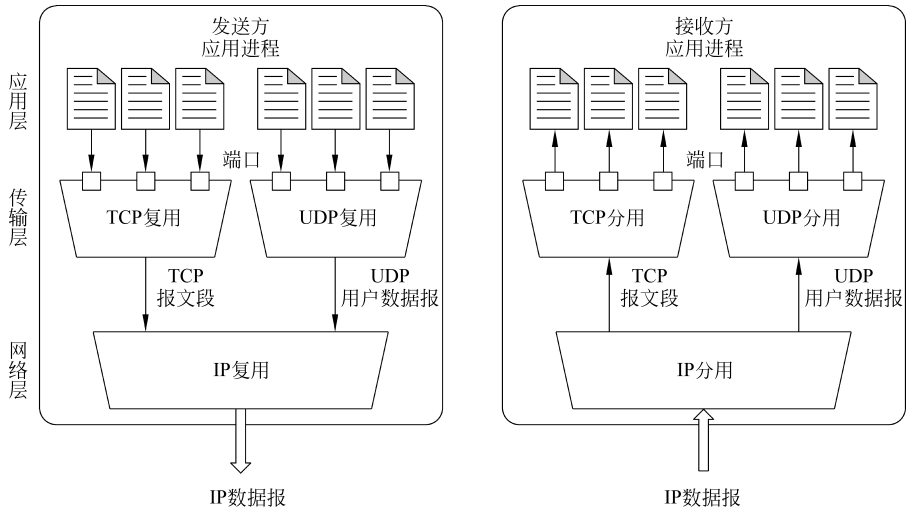


图 3.3　TCP 与 UDP

3.1.1 协议端口

传输层是网络中非常关键的一层。传输层的下面是网络级通信，传输层上面是主机级通信，即进程级通信。由于每个进程都与一个相应的应用协议相联系，因此 TCP/UDP 协议簇中使用 2B 协议端口号（通常也简称端口号）来标识一台机器上的多个进程。端口可以分为 3 大类。

1. 公认端口

公认端口（well known ports）也称为统一分配（universal assignment）端口、众所周知端口和保留端口，是由中央管理机构用静态方式分配的端口号。这些端口号是固定的、全局性的，所有采用 TCP/IP 协议的标准服务器都必须遵从。TCP 与 UDP 的标准端口号是各

自独立编号的，范围在 0～1023 之间。

2. 注册端口

注册端口（registered ports）号在 1024～49 151。它们松散地绑定于一些服务上，多数没有明确地定义服务对象，不同程序可以根据需要自己定义，向 IANA 注册。

3. 动态和/或私有端口

这类端口（dynamic and/or private ports）在 49 152～65 535 之间，可以临时顶替某些端口，以免与别的同协议进程冲突。

表 3.1 列出了一些当前分配的 TCP 和 UDP 的端口号。

表 3.1　一些当前分配的 TCP 和 UDP 的协议端口号

端口号	关　键　字	UNIX 关键字	说　　明	UDP	TCP
7	ECHO	echo	回显		Y
13	DAYTIME	daytime		Y	Y
19	CHARACTER GENERATOR	Character generator		Y	Y
20	FTP_DATA	ftp_data	文件传输协议（数据）		Y
21	FTP_CONTRAL	ftp	文件传输协议（命令）		Y
22	SSH	ssh	安全命令解释程序		Y
23	TELNET	telnet	远程连接		Y
25	SMTP	smtp	简单邮件传输协议		Y
37	TIME	time	时间	Y	Y
42	NAMESERVER	name	主机名服务器	Y	Y
43	NICNAME	whois	找人	Y	Y
53	DOMAIN	nameserver	DNS（域名服务器）	Y	Y
69	TFTP	tftp	简单文件传输协议	Y	
70	GOPHER	gopher	Gopher		Y
79	FINGER	finger	Finger		Y
80	WWW	www	WWW 服务器		Y
101	HOSTNAME	hostname	NIC 主机名服务器		Y
103	X400	x400	X.400 邮件服务		Y
104	X400_SND	x400_snd	X.400 邮件发送		Y
110	POP3	pop3	邮局协议版本 3		Y
111	RPC	rpc	远程过程调用	Y	Y
119	NNTP	nntp	USENET 新闻传输协议	Y	Y
123	NTP	ntp	网络时间协议	Y	Y
161	SNMP	snmp	简单网络管理协议	Y	
179	BGP		边界网关协议		Y
520	RIP		路由信息协议	Y	

3.1.2 TCP 的特征

TCP 是传输层的一个主要协议，从应用程序的角度看，TCP 提供的服务具有如下特征。

1．端对端的通信

TCP 是在网络层提供的服务基础上，提供一个直接从一台计算机上的应用到另一台远程计算机上的应用的连接。由于每一个 TCP 连接都有两个端点，所以是一种端对端的协议。

2．虚电路连接

TCP 提供面向连接的服务，其传输过程由 3 个过程组成：建立连接、传输数据和释放连接。即一个应用程序必须首先请求一个到目的地的连接，然后才能使用该连接传输数据。由于该连接是通过软件实现的，所以是虚连接（virtual connection）。

3．全双工通信

一个 TCP 连接允许任何一个应用程序在任何时刻发送数据，使数据在该 TCP 的任何一个方向上流动，并可以在传输数据包时搭载应答信息。

4．可靠传输

（1）可靠连接。指防止在连接过程一方要发送数据而另一方还没有做好准备。

（2）确认和超时重传机制。TCP 在发送一段报文时，要同时在自己一侧存放该报文的一个副本。若收到确认，则删除该副本；若在超时之前没有收到确认，则重传该报文段。

（3）字节编号。TCP 协议是面向字节的，它将所要传送的报文看成字节流。为了对字节的确认，需要为每个字节提供一个编号（在数据链路层中是为帧进行编号）。另外，字节序号并不是从 0 或 1 开始的。初始的序号是通信开始时双方商定的。并且，TCP 接收方送给发送方的确认是接收到的最后一个序号+1——期待接收的数据的编号，即前面的字节都已经正确收到。例如，接收端已经正确地接收到了 201~300 号字节的数据，则发给对方的确认号为 301，表示等待 301 号字节到来。

（4）校验和。TCP 采用校验和进行检错。这种方法检错能力不强，但效率比较高，符合 TCP/IP 的设计原则。同时，随着低层网络质量的改善，这种方法也能达到要求。

（5）从容关闭。从容关闭是指确保关闭连接之前传递完所有的数据。

5．基于滑动窗口的流量控制和拥塞控制

当建立一个 TCP 连接时，连接的每一端分配一个缓冲区来保存输入的数据。通常把缓冲区中的空闲部分称为窗口。TCP 采用可变滑动窗口协议，并且当交付的数据不够填满一个缓冲区时，应用程序可以用流服务提供的“推（push）”机制进行强迫传送。

3.1.3 TCP 报文格式

TCP 的报文段由首部和数据部两部分组成。图 3.4 为 TCP 报文段的首部格式。TCP 的数据部为应用层报文。

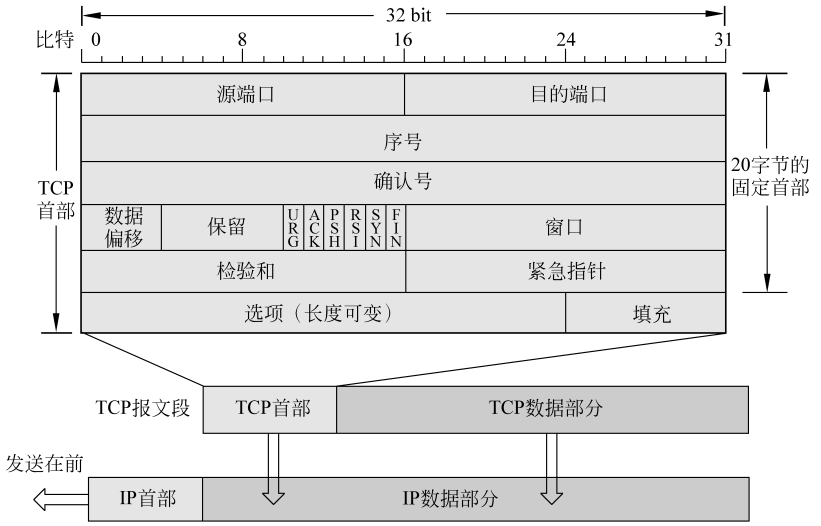


图 3.4　TCP 报文段格式

下面分别介绍各字段的含义。

1）源端口（Source Port）

源端口即本地通信端口，支持 TCP 的多路复用机制。

2）目的端口（Destination Port）

目的端口即远地通信端口，支持 TCP 的多路复用机制。

3）序号（Sequence Number）

序号是数据段的第 1 个数据字节（除含有 SYN 的段外）的序号。在 SYN 段中，该域是 SYN 的序号，即建立本次连接的初始序号，在该连接上发送的第 1 个数据字节的序号为初始序号+1。

4）数据偏移（Data Offset）

数据偏移指出该段中数据的起始位置，以 4 字节为单位（TCP 头总以 32 位边界对齐）。

5）6 个控制位（Control Bit）

URG——紧急指针域有效。URG=1 表示该段中携带有紧急数据。

ACK——ACK=1，确认序号有效；ACK=0，确认序号无效。

RST——连接复位，RST=1 表示 TCP 连接中出现严重差错，必须释放连接。

SYN——建立连接时用来同步序号：SYN=1，ACK=0，表明这是一个连接请求报文段；SYN=1，ACK=1，表明这是一个连接请求或连接请求接受报文段。

FIN——发送方字节流结束。FIN=1 表明本端数据已经发送完，请求释放连接。

PSH——本报文段请求“推（push）”操作。即认定该段为“推进”段，段中数据是发送方当时发送缓冲区中的全部数据。对于收到这种数据的接收方来讲，应当把“推进”段

中的数据尽快交给用户，并结束一次用户接收请求。

6）确认号（Acknowledgment Number）

当 TCP 段头控制位中的 ACK 置位时，确认号才有效。它表示本地希望接收的下一个数据字节的序号。对于收到有效确认号的发送者来说，其值表示接收者已经正确接收到了该序号以前的数据。

7）窗口（Window）

表明该段的接收方当前能够接收的从确认号开始的最大数据长度，该值主要向对方通告本地接收缓冲区的使用情况。

8）校验和（Checksum）

校验对象包括协议伪头、TCP 报头和数据。

9）选项（Option）

选项位于 TCP 头的尾端，有单字节和多字节两种格式。单字节格式只有选项类型；多字节格式由一个字节的选项类型、多字节的实际选项数据和一个字节的选项长度（三部分的长度）组成。下面说明 TCP 协议必须实现的选项。

（1）选项表尾选项：KIND=0。表示 TCP 头中由全部选项组成的选项表结束。其格式为：

0	0	0	0	0	0	0	0

（2）无操作选项：KIND=1。该选项可能出现在两个选项之间，作为一个选项分隔符，或提供一种选项字边界对齐的手段，本身无任何意义。其格式为：

0	0	0	0	0	0	0	1

（3）最大段长选项：KIND=2，LENGTH=4。该选项主要用于通知通信连接的对方本地能够接收的最大段长。它只出现在 TCP 的初始建链请求中（SYN 段）。如果在 TCP 的 SYN 段中没有给出该选项，就意味着有能力接收任何长度的段。其格式为：

0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	最大段长

10）填充（Padding）

当 TCP 头由于含有了选项而无法以 32 位边界对齐时，将会在 TCP 头的尾部出现若干字节的全 0 填充。

11）URG 位和紧急指针（Urgent Pointer）

传输层协议使用带外数据（Out-Of-Band，OOB）机制来传输一些重要数据，如通信的一方有重要的事情通知对方，需要加速传送这些通知数据。TCP 支持一个字节的带外数据，并提供了一种紧急模式：在数据分组中设置 URG = 1，表示进入紧急模式，同时用紧急指针表明从该段序号开始的一个正向位移，指向紧急数据的最后一个字节。

3.1.4 TCP 的可靠连接与从容关闭

TCP 是一种面向连接的协议，所以其传输过程由 3 个过程组成：建立连接、传输数据和释放连接。

1．TCP 的可靠连接

TCP 的建立应当是可靠的。TCP 建立可靠连接的方法是采用三次握手（three-way handshaking）方法。握手也称联络，是在两个或多个网络设备之间通过交换报文序列以保证传输同步的过程。如图 3.5 所示为用三次握手方式建立 TCP 连接的过程。

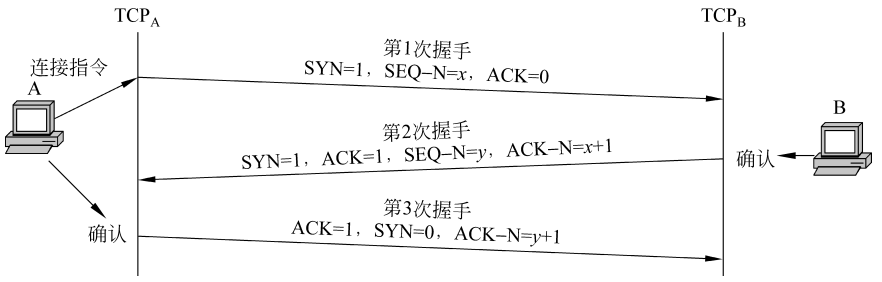


图 3.5　三次握手建立可靠 TCP 连接的过程

第 1 次握手：主机 A 发出主动打开（active open）命令，TCP_A 向 TCP_B 发出请求报文，内容如下。

- SYN=1，ACK=0：表明该报文是请求报文，不携带应答。
- SEQ–N= x：自己的序号为 x，后面要发送的数据序号为 x+1。

第 2 次握手：TCP_B 收到连接请求后，如同意连接，则发回一个确认报文，内容如下。

- SYN=1，ACK=1：该报文为接收连接确认报文，并携带有应答。
- ACK–N = x + 1：确认了序号为 x 的报文，期待接收序号以 x + 1 为第一字节的报文。
- SEQ–N = y：自己的序号为 y，后面要发送的数据序号为 y + 1。

这时，TCP_A 和 TCP_B 会分别通知主机 A 和主机 B，连接已经建立。

到此为止，似乎就可以正式传输数据报文了。但是，问题没有这么简单。因为虽然 B 端同意了接收由 TCP_A 发起的连接，准备好了接收由 TCP_A 发来的数据，而 A 端还没有同意由 TCP_B 发起的连接。所以这时的连接仅仅是全双工通信中的半连接——TCP_A 到 TCP_B 的连接，TCP_B 到 TCP_A 连接并没有建立起来。

所以，只有两次握手的连接是不可靠的。为了避免这种情况，必须再来一次握手。

第 3 次握手：TCP_A 收到含两次初始序号的应答后，再向 TCP_B 发一个带两次连接序号的确认报文，内容如下。

- ACK=1，SYN=0：该报文是单纯的确认报文，但不携带要传输数据的序号。
- ACK–N = y+1：确认了序号为 y 的报文，期待第 1 字节序号为 y + 1 的数据字段。

这样，双方才可以开始传输数据，并且不会出现前面的问题了。

经过三次握手，已经实现了可靠连接的双方就可以传输数据了。下面介绍数据传输过程中的一些关键技术。

2．TCP 连接的从容关闭

TCP 连接是在硬件连接的基础上通过软件实现的，所以称为软连接。软连接后就要占用硬连接的资源。连接释放就是释放一个 TCP 连接所占用的资源。

正常的释放连接是通过断连请求及断连确认实现的。但是，在某些情况下，没有经过断连确认，也可以释放连接，但断连不当就有可能造成数据丢失。如图 3.6 所示为一种断连不当引起数据丢失的情形：A 方连续发送两个数据后，发送了断连请求；B 方在收到第一个数据后，先发出了断连请求，结果第二个数据丢失。

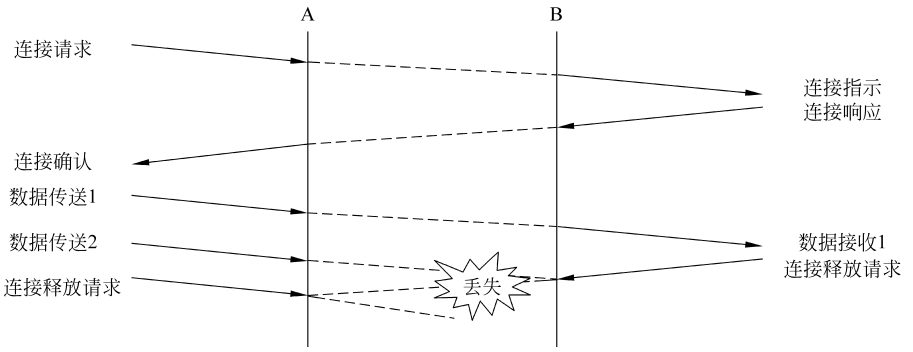


图 3.6　断连不当引起数据丢失

为了防止因断连不当引起的数据丢失，断连应选择在确信对方已经收到自己发送的数据并且自己和对方不再发送数据时。进行由于 TCP 连接是双工的，它包含了两个方向的数据流传送，形成两个“半连接”。在撤销时，一方发起撤销连接但连接依然存在，要在征得对方同意之后，才能执行断连操作。

下面分两种情况考虑连接释放问题：传输正常结束释放和传输非正常结束释放。

1）传输正常结束释放

数据传输正常结束后，就应当立即释放这次 TCP 连接所占用的资源。所以连接的双方都可以发起释放连接。如图 3.7 所示为一个由 A 方先发起的连接可靠释放过程。一般它是一个 4 次握手过程。

图 3.7　A 方先发起的连接可靠释放过程

第 1 次握手：主机 A 先向 TCP_A 发出连接释放指令 FIN，并不再向传输层发送数据；T_{CPA} 向 T_{CPB} 发送释放通知报文，内容如下。

- FIN=1：A 已经没有数据发送，要求释放从 A 到 B 的连接。
- SEQ–N=*x*：本次连接的初始序列号（即已经传送过的数据的最后一个字节的序号加 1）为 *x*。

第 2 次握手：TCP_B 收到 TCP_A 的连接释放通知 FIN 后，向 TCP_A 发确认报文，内容如下。

- ACK=1：确认报文。
- ACK–N = x +1：确认了序号为 x 的报文。
- SEQ–N = y：自己的序号为 y。

这时，从 TCP_A 到 TCP_B 的半连接就被释放。而从 TCP_B 到 TCP_A 的半连接还没有释放，从 TCP_B 还可以向 TCP_A 传送数据，连接处于半关闭（half-close）状态。如果要释放从 TCP_B 到 TCP_A 的连接，还需要进行类似的释放过程。这一过程可以第 1 次握手后开始，即选择另一种第 2 次握手。

另一种第 2 次握手：TCP_B 收到 TCP_A 的连接释放通知后，即向主机 B 中的高层应用进程报告，若主机 B 也没有数据了，主机 B 就向 TCP_B 发出释放连接指令，并携带对于 TCP_A 释放连接通知的确认。报文内容如下。

- FIN=1，ACK=1：释放连接通知报文，携带了确认。
- SEQ–N=y，ACK-N = x+1：确认了序号为 x 的报文，自己的序号为 y。

第 3 次握手：TCP_A 对 TCP_B 的释放报文进行确认。报文内容如下。

- ACK=1：确认报文。
- SEQ–N = x+1，ACK-N = y+1：本报文序列号为 x + 1；确认了 TCP_B 传送来的序号为 y 的报文。

这时，从 TCP_B 到 TCP_A 的连接也被释放。

2）传输非正常结束释放

在有些情况下，希望 TCP 传输立即结束。为了提供这种服务，当一方突然关闭时，TCP 会立即停止发送和接收，清除发送和接收缓冲区，同时向对方发送一个 RST=1 的报文，要求重新建立连接。

3.1.5 TCP 数据传输

1．确认和超时重传机制

TCP 传输的可靠性在于其使用了序号和确认：发送方通知发送序号，接收方在此基础上确认（用期望序号表示）。同时，为了防止发送后接收方收不到的情况，TCP 每发送一个报文，就在自己的重发队列中存放一个该报文的副本，并对此报文设置一个计时器，为超时重发做准备。如果一个 TCP 段在规定的时间片内收不到确认（接收端没有收到报文或报文错误，都不发确认），就重传该报文。

2．流量与拥塞控制

TCP 采用滑动窗口进行流量和拥塞控制。TCP 的流量控制“窗口”是一种可变窗口。当接收方用户没有及时取走滞留在 TCP 缓冲区的数据时，会占用系统资源，窗口将变小；当接收方取走在 TCP 缓冲区中的数据时，释放了系统资源，窗口将变大。也就是说，TCP 允许随时改变窗口大小，这样不仅可以提供可靠传输，还可以提供很好

的流量控制。

与可变窗口相配套的是窗口通告（window advertisement），即每个确认中，除了要指出已经收到的 8 位组序号外，还包括一个窗口通告，用于说明接收方窗口（接收缓冲区）还有多大——能接收多少个 8 位组数据。发送方要以当前记录的接收方最新窗口大小为依据决定发送多少 8 位组。所以通常也把接收端窗口（receiver window，rwnd）称为通告窗口（advertised window）。

3. 拥塞控制的慢开始与拥塞避免算法

TCP 通过通告窗口，使得发送端的发送能力不大于接收端的接收能力。但是，这样并不能完全避免网络拥塞。因为网络是一个多结点的系统，其拥塞状况并不完全取决于某个接收方的接收能力。在这种情况下，为了避免网络拥塞状况恶化，发送端还需要根据网络的拥塞程度调整自己的发送能力。为此，除了要设置一个按照接收方接收能力决定的通告窗口外，还要设置一个按照网络拥塞程度决定的一个发送窗口限制——拥塞窗口（congestion window，简写为 cwnd）。显然，实际的发送窗口的上限应当取通告窗口与拥塞窗口中的小者。

慢开始和拥塞避免算法是早期使用的决定拥塞窗口大小的两个算法。

（1）慢开始算法：首先设置 cwnd 为 1 个 MSS（Maximum Segment Size，最大报文段中的数据字节数），以后每收到其 ACK 后，将 cwnd 增加至多一个 MSS 值，再发送相应数量的报文段。这样，在不出现拥塞的情况下：

第 1 次发送后，cwnd 将增加为 2 个 MSS，即一次具有 2 个 MSS 的发送能力；

第 2 次发送后，cwnd 将增加为 4 个 MSS，即一次具有 4 个 MSS 的发送能力；

……

拥塞窗口呈指数规律增长。

（2）拥塞避免算法不是按照收到的 ACK 数量增加 cwnd，而是按照时间，即每经过一个往返时延 RTT，增加一个 MSS 大小，使 cwnd 呈线性增长。

为了控制拥塞状况，要设置一个门限 ssthesh（通常设置为 65 535 字节，即 16 个报文段），形成如下拥塞控制算法：

（1）比较 cwnd 与 ssthesh。

- cwnd < ssthesh，继续执行慢开始算法；
- cwnd > ssthesh，停止慢开始算法，改用拥塞避免算法；
- cwnd = ssthesh，可执行慢开始算法。也可执行拥塞避免算法。

（2）将发送窗口设置为通告窗口与拥塞窗口中的小者。

（3）网络每出现一次拥塞（出现某个报文段的超时），就执行一次 ssthesh=0.5×ssthesh 的计算（但不能小于 2MSS）。这个计算被称为“乘法减小”（multiplicative decrease）原则。

（4）若执行拥塞避免算法后，发送端能够收到所有发出报文的确认，便要执行 cwnd=cwnd+MSS 的操作。这称为“加法增大”（odditive increase）原则。

按照上述算法，可以得到如图 3.8 所示的 TCP 拥塞窗口变化规律。

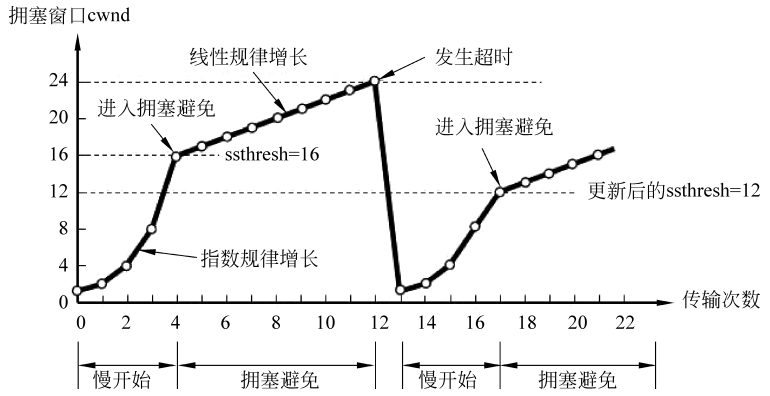


图 3.8　采用慢开始和拥塞避免算法的 TCP 拥塞窗口变化规律

4. 拥塞控制的快重传和快恢复

慢开始和拥塞避免算法在有些情况下会让 TCP 等待某个数据段的超时后才重新开始发送丢失的报文段。例如，发送端连续发送报文段 M_1~M_6。则接收方在收到 M_1 和 M_2 后，将发出 ACK_2 和 ACK_3。若所发送的报文 M_3 丢失，则收到 M_4 后发出的仍然是 ACK_3；收到 M_5 后发出的仍然是 ACK_3，收到 M_6 后发出的仍然是 ACK_3。尽管收到这么多 ACK_3，但是发送段还是要等到 M_3 的重传计时器超时后，才重传 M_3。

快重传的基本思想是早些重传丢失报文段，它规定只要收到某个报文段的 3 个重复的 ACK，就要立即重发该报文段，而不必等待其重传计时器超时。

在尽早开始重传的同时，还可以使用快恢复算法使网络出现拥塞后尽快使网络恢复到正常工作状态。将快重传和快恢复结合起来，形成下面的算法：

（1）当发送端连续收到三个重复的 ACK 时，则按照“乘法减小”的原则，重新设置慢开始门限 ssthresh。

（2）设置拥塞窗口 cwnd 为 ssthresh+ n×MSS。n（≥3）为收到的 ACK 的数量。因为收到的 n 个重复的 ACK，是接收端对已经到达的 3 个报文的应答。这 n 个报文段已经保存在接收端的缓存中。所以网络中不是堆积了报文，而是减少了报文。这比慢开始将拥塞窗口设置为 1，要恢复得快。

（3）若发送窗口还允许发送报文段，就按拥塞避免算法继续发送报文段。

（4）若收到了确认新的报文段的 ACK，就将 cwnd 缩小到 ssthresh。

这是一种可以明显改进 TCP 性能的算法。

5. TCP 的紧急数据传输

用户有时要发送一些紧急的数据，例如在一个要运行很长时间的程序执行过程中取消该程序的运行等，需要发出由 Control+C 两个字符组成的中断命令。这时，就需要将紧急发送的报文段中的 URG 位置 1，以告诉 TCP 这个报文段中有紧急数据，同时将这两个字符插入到报文段的数据字段的前面，并用一个紧急指针（Urgent Pointer）指出紧急数据的最后一个字节的序号，以通知接收方紧急数据的大小。接收方即使窗口为 0，发送方也可以发送紧

急数据。

接收方收到紧急数据并处理完后，TCP 即通知应用程序恢复到正常工作。

3.1.6 UDP

UDP 是一个无连接的协议，在发送时无须建立连接，仅仅向应用程序提供了一种发送封装的原始 IP 数据报的方法。如图 3.9 所示为 UDP 数据报格式。其中校验和是可选的，当不进行校验时，这个域为 0。

UDP源端口	UDP目的端口
UDP数据报长度	UDP校验和
UDP数据区	

图 3.9 UDP 数据报格式

UDP 是一个基于不可靠通信子网的不可靠传输层协议。因此，基于 UDP 的应用程序必须自己解决可靠性问题，如报文丢失、报文重复、报文失序、流量控制等。

应用程序可以使用 UDP 进行通信。不同的进程用不同的端口号进行标识。端口号分为公认（众所周知）端口号和自由端口号两种。

在 UNIX 系统中，一个 UDP 端口是一个可读和可写的软件结构，UDP 为每个端口维护一个接收缓冲区。发送数据时，UDP 将数据内容生成一个 UDP 数据报，然后交给网络层的 IP 发送。接收数据时，UDP 从网络层 IP 接收到 UDP，然后根据目的端口号将其放在相应的接收缓冲区中。如果没有匹配的端口号，UDP 将丢弃该数据报，并向发送主机返回一个“不可到达”的 ICMP 消息；如果匹配端口号已满，UDP 也丢弃该数据报，但不回送错误消息，该数据报要靠超时重发。

UDP 的优点在于高效率，通常用于交易型应用，一次交易只有一来一往两次报文交换。

3.2 IPv4

IP 层是基于网络互联的 TCP/IP 计算机网络体系中关键的一层。在传输层虽然解决了进程之间的通信问题。但由于进程是运行在主机上的，因此只有解决了主机之间的通信问题，才能有进程之间的通信。因此需要解决如下 3 个关键问题。

（1）主机运行在不同网络之中，为此需要解决对于源主机和目标主机的识别问题。因此 IP 提出了一套对于网络和主机的编号规则——IP 地址协议。

（2）主机到主机需要经过一系列的网络，为此需要解决数据的传输路径问题，因此提出了一套路径选择的算法——路由协议。

（3）在传输层，报文分段是根据端主机之间的接收能力进行分段的。这些分段在 IP 层，则要按照网络之间的路由能力进一步切分封装成分组传输。

目前广为应用的 IP 是其第 4 个版本——IPv4。

3.2.1 IP 分组格式

1. IP 分组的形成

在 IP 层要对传输层的数据进行分片，并进一步加上 IP 头，封装 IP 分组。如图 3.10 所示，每个分组都是一个两层封装：外封装是 IP 头，内封装的从传输层传来的 TCP 或 UDP 头。

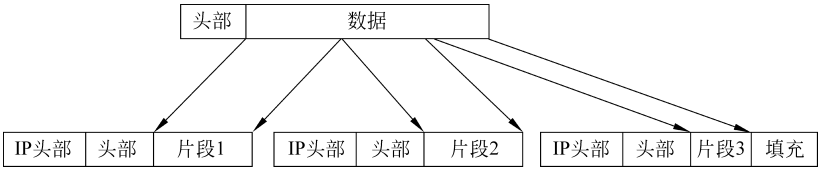


图 3.10　IP 数据分组的形成

在 IP 分组中，对于数据部分规定了一定的长度。最后一个分组中的数据是前面分片剩余的数据，长度往往不正好是规定的长度，不足部分需要进行填充。

2. IPv4 分组格式

图 3.11 为 IPv4 分组格式。

图 3.11　IPv4 分组格式

下面介绍 IPv4 分组首部各字段的意义。

1）IPv4 分组首部的固定部分各字段含义

（1）版本。占 4b，指 IP 协议的版本。通信双方使用的 IP 协议版本必须一致。目前广泛使用的 IP 协议版本号为 4（即 IPv4）。关于 IPv6，目前还处于试用阶段。

（2）首部长度。占 4b，可表示的最大十进制数值是 15。请注意，这个字段所表示数的单位是 32 位（4B）字长，因此，当 IP 的首部长度为 1111 时（即十进制的 15），首部长度就达到 60B。若 IP 分组的首部长度不是 4B 的整数倍时，就必须利用最后的填充字段加以填充。因此数据部分永远在 4 字节的整数倍开始，以便于实现 IP 协议。首部长度限制为 60B 的缺点是有时可能不够用，但能使用户尽量减少开销。最常用的首部长度就是 20B（即首部长度为 0101），这时不使用任何选项。

（3）区分服务。占 8b，用来获得更好的服务。这个字段在旧标准中叫作服务类型，被分为 3 个部分：

- 优先级子字段（3b），即前 3b，用 0～7 表示优先级别高低。但现在已经不用。
- 第 8 位保留未用。
- TOS，中间 4b，分别代表：最小时延、最大吞吐量、最高可靠性和最小费用。每种服务类型中只能置其中 1 比特为 1。可以全为 0，若全为 0 则表示一般服务。

表 3.2 列出了对不同应用建议的 TOS 值。例如，TELNET 协议可能要求有最小的延迟，FTP 协议（数据）可能要求有最大吞吐量，SNMP 协议可能要求有最高可靠性，NNTP（Network News Transfer Protocol，网络新闻传输协议）可能要求最小费用，而 ICMP 协议可能无非凡

要求（4 比特全为 0）。实际上，大部分主机会忽略这个字段，但一些动态路由协议如 OSPF（Open Shortest Path First Protocol）、IS-IS（Intermediate System to Intermediate System Protocol）可以根据这些字段的值进行路由决策。

表 3.2　对不同应用建议的 TOS 值

应用程序	最小时延	最大吞吐量	最高可靠性	最小费用	十六进制值
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

（4）总长度。总长度指首部和数据之和的长度，单位为字节。总长度字段为 16b，因此分组的最大长度为 2^{16}–1=65 535B。

在 IP 层下面的每一种数据链路层都有自己的帧格式，其中包括帧格式中的数据字段的最大长度，这称为最大传送单元 MTU（Maximum Transfer Unit）。当一个分组封装成链路层的帧时，此分组的总长度（即首部加上数据部分）一定不能超过下面的数据链路层的 MTU 值。

（5）标识（identification）。占 16b。IP 软件在存储器中维持一个计数器，每产生一个分组，计数器就加 1，并将此值赋给标识字段。但这个“标识”并不是序号，因为 IP 是无连接服务，分组不存在按序接收的问题。当分组由于长度超过网络的 MTU 而必须分片时，这个标识字段的值就被复制到所有分组的标识字段中。相同的标识字段的值使分片后的各分组片最后能正确地重装成为原来的分组。

（6）标志（flag）。占 3b，但目前只有 2b 有意义。

- 标志字段中的最低位记为 MF（More Fragment）。MF=1 表示后面“还有分片”的

分组。MF=0 表示这已是若干分组片中的最后一个。

- 标志字段中间的一位记为 DF（Don't Fragment），意思是“不能分片”。只有当 DF=0 时才允许分片。

（7）片偏移。占 13b。片偏移指出：较长的分组在分片后，某片在原分组中的相对位置。也就是说，相对用户数据字段的起点，该片从何处开始。片偏移以 8B 为偏移单位。这就是说，每个分片的长度一定是 8B（64b）的整数倍。

（8）生存时间。占 8b，常用的 TTL（Time To Live）表示，由发送数据的源主机设置，限制分组最多可以经过的路由器数。通常为 32、64、128 等。每经过一个路由器，其值减 1，直到 0 时该数据报被丢弃。防止分组进入死循环。

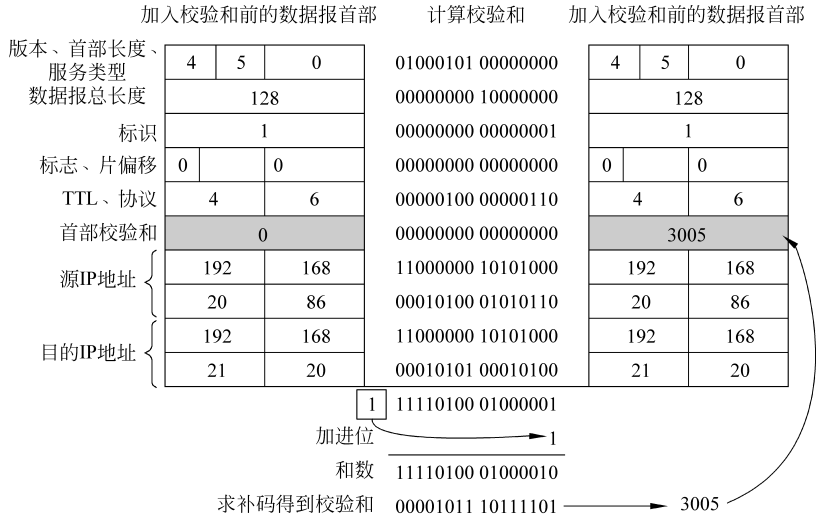
（9）协议。占 8b，协议字段指出此分组携带的数据是使用何种协议，以便使目的主机的 IP 层知道应将数据部分上交给哪个处理过程。表 3.3 为常用协议的协议字段值。

表 3.3　常用协议的协议字段值

协议名	ICMP	IGMP	特殊 IP	TCP	EGP	IGP	UDP	IPv6	ESP	OSPE
协议字段值	1	2	4	6	8	9	17	41	50	89

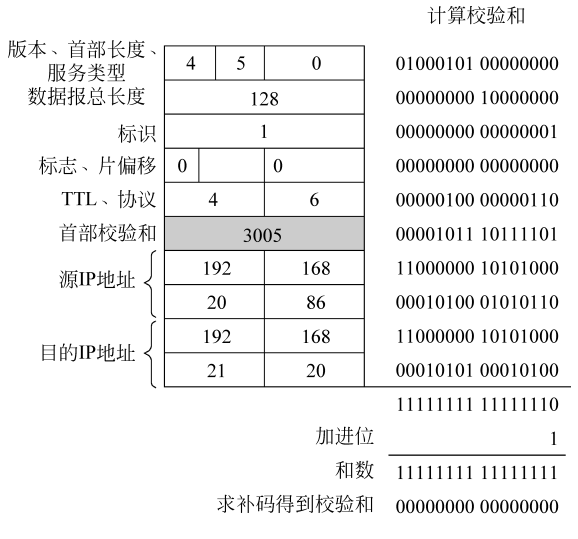
表中的“特殊 IP”指被封装到 IP 分组中的 IP 分组。

（10）首部检验和。占 16b。这个字段只检验分组的首部，不包括数据部分。这是因为分组每经过一个路由器，路由器都要重新计算一下首部检验和（一些字段，如生存时间、标志、片偏移等都可能发生变化）。不检验数据部分可减少计算的工作量。为了使读者对于首部校验和有进一步了解，图 3.12（a）给出了发送端计算校验和的过程，图 3.10（b）给出了接收端检验校验和的过程。



(a) 发送端计算校验和

图 3.12　IP 分组中的校验和使用实例



(b) 接收端检验校验和

图 3.12（续）

（11）源 IP 地址、目标 IP 地址字段：各占 32b，用来标明发送 IP 数据报文的源主机地址和接收 IP 报文的目标主机地址。

2）IP 分组首部的可变部分

IP 首部的可变部分就是一个可选字段。选项字段用来支持排错、测试以及安全等措施，内容很丰富。此字段的长度可变，从 1 个字节到 40 个字节不等，取决于所选择的项目。某些选项项目只需要 1 个字节，它只包括 1 个字节的选项代码。但还有些选项需要多个字节，这些选项一个个拼接起来，中间不需要有分隔符，最后用全 0 的填充字段补齐成为 4 字节的整数倍。

增加首部的可变部分是为了增加 IP 分组的功能，但这同时也使得 IP 分组的首部长度成为可变的。这就增加了每一个路由器处理分组的开销。实际上这些选项很少被使用。新的 IP 版本 IPv6 就将 IP 分组的首部长度做成固定的。

目前，这些任选项定义如下：

（1）安全和处理限制（用于军事领域）。

（2）记录路径（让每个路由器都记下它的 IP 地址）。

（3）时间戳（让每个路由器都记下它的 IP 地址和时间）。

（4）宽松的源站路由（为分组指定一系列必须经过的 IP 地址）。

（5）严格的源站路由（与宽松的源站路由类似，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，并非所有主机和路由器都支持这些选项。

3.2.2 IP 地址格式

1. IP 地址分类

TCP/IP 是面向网络互连而建立的一种网络体系。为了进行网络互联，就要考虑不同的

网络互联之后，一个网络中的某台主机与另一个网络中的某台主机之间的通信问题。也就需要对所有连接起来的主机进行编码。为此提出了一套编码标准，这个编码标准称为 IP 地址。所有想要连接在 Internet 中的网络或计算机必须遵循这个规则，申请到合法的 IP 地址，才可以被识别，进行通信。

IPv4 规定的 IP 地址是 32b 编码，即 4B。为了便于理解和记忆，通常将每个字节用一个十进制数表示，并用小数点分隔，称为点分十进制（dotted decimal notation）码。例如某 IP 地址为

10000000　00001010　00000010　00011110

可以写为

128.10.2.30

如图 3.13 所示，IP 地址可分为三部分：

- 类型标志，分为 A、B、C、D、E 五类，分别用 0、10、110、1110 和 11110 标识。
- 网络标识符（netID）。
- 主机编号（hostID）。

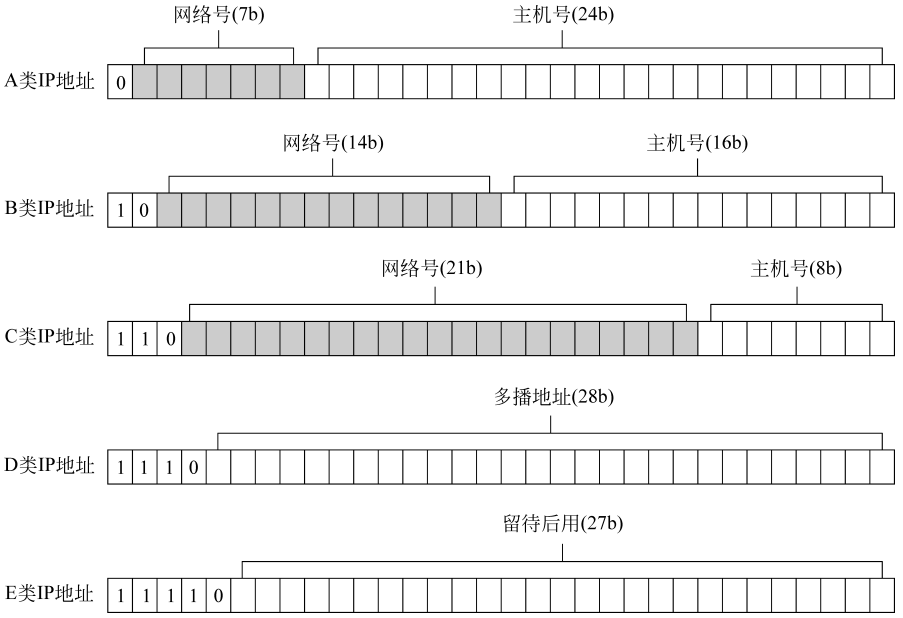


图 3.13　五类 IPv4 地址结构

A、B、C 是 3 类基本地址类型，区别仅在于网络大小不同，如表 3.4 所示。

表 3.4　A、B、C 三类基本 IP 地址类型与规模

类别	类型标识	网络规模	netID 位数	可编址网络数	hostID 位数	每个网络最多主机数	IP 地址范围
A	0	巨型网络	7	126	24	16777214	1.0.0.0～127.255.255.255
B	10	大型网络	14	16383	16	65534	128.0.0.0～191.255.255.255
C	110	中小型网络	21	2097151	8	254	192.0.0.0～255.255.255.255

D 类地址是一种多址广播地址格式，用 1110 作为标志，netID 的范围为 224～239。

E 类地址是为研究和实验保留的地址，用 11110 作为标志，netID 的范围为 240 及以上。例如，上述 128.10.2.30，显然是一个 B 类地址。由于 B 类地址网络地址占 2B，所以，其网络地址为 128.10，网络内主机地址为 2.30。

2. IP 地址的重要特点

（1）在 Internet 网络中，只用 IP 地址中的 netID 来标识主机所在的网络，即 netID 相同的主机就认定是网络中的主机，不管它们物理上是如何连接的。

（2）不同 netID 的局域网或主机之间必须使用路由器连接。

（3）路由器一个端口用来连接一个网络。当一台路由器连接多个网络时，它的多个端口都会有一个 netID 不同的 IP 地址。一台路由器最少连接两个网络，即它最少有两个 netID 不同的 IP 地址。

（4）路由器仅根据 netID 转发分组，即路由表中只需存储网络号，无须存储主机号。这样可以减少路由表的存储量，提高路由器的处理速度。

（5）两台路由器可以直接相连。这时，在这条连线就被看成一个逻辑上的网络，其连接的两个端口可以分配一个 IP 地址，也可以不分配。若不分配 IP 地址，就被称为无编号网络（unnumbered network）或匿名网络（anonymous network）。

（6）一台主机也可以拥有两个 netID 不同的 IP 地址。这样的主机称为多属主机（multihomed host）。

3. IP 地址的使用约束

1）私有 IP 地址

在规划的时候，将 IP 地址分为私有地址和公有地址，私有地址只能在内部网络使用，不能在互联网使用，认为这样的地址是互联网的不合法地址。在 A、B、C 三类地址中都选择一部分地址作为私有地址：

A 类地址中，netID 为 10 的所有 IP 地址。

B 类地址中，netID 为 172.16~172.31 的所有 IP 地址。

C 类地址中，netID 为 192.168 的所有 IP 地址。

2）受限 IP 地址

netID 全 0，只用作源地址，表示本网，不路由；若 hostID 全 0，在 DHCP 协议中指本机。

netID 为 1（全 255），仅用作目的地址，代表所有网络；若若 hostID 全 0，仅作子网掩码。

hostID 全 0，仅表示网络地址，不可用作源地址。

hostID 全 1，不可作源地址，仅向指定网络广播。

netID 为 127，仅可以用作本地软件环回测试。

4. IP 地址的分配与申请

为了确保一个 IP 地址对应一台主机，IP 地址由 IANA（Internet Assigned Numbers Authonty，互联网名称与数字地址分配机构）统一编号，并交由 Internet 名字与号码指派公

司 ICANN（Internet Corporation for Assigned Names and Numbers）进行统一分配。ICANN 将地址的分配授权给 RIR（Regional Internet Registry，区域性互联网注册机构）负责地区的登记注册申请。现在全球有 5 个 RIR：

（1）RIPE（Reseaux IP Europeans）欧洲 IP 地址注册中心——服务于欧洲、中东地区和中亚地区；

（2）LACNIC（Lation American and Caribbean Internet Address Registry）拉丁美洲和加勒比海 Internet 地址注册中心——服务于中美、南美以及加勒比海地区；

（3）ARIN（American Registry for Internet Numvers）美国 Internet 编号注册中心——服务于北美地区和部分加勒比海地区；

（4）AFRINIC（Africa Network Information Centre）非洲网络信息中心——服务于非洲地区；

（5）APNIC（Asia Pacific Network Information Centre）亚太地址网络信息中心——服务于亚洲和太平洋地区的国家。

另外，许多国家和地区都成立了自己的域名系统管理机构，负责从前述 3 个机构获取 IP 地址资源后在本国或本地区的分配与管理事务。这些国家和地区的域名系统管理机构大多属于半官方或准官方机构。但在实际运作过程中，相关国家或地区的政府至少在业务上对其不加干预，使其成为前述 3 个机构之一在各该国家或地区的附属机构。如日本的 JPNIC 和中国的 CNNIC 均属此种机构。

获得公有 IP 地址的方法：向 InterNIC 申请，也可以向 ISP 申请。

3.2.3 子网划分与子网掩码

1. 子网与子网编码

任何一个 A、B、C 类 IP 地址都对应着一定规模（主机数目）的网络，是一种二级地址。当实际的网络规模接近 IP 地址的主机 ID 上限时，该 IP 地址才能得到充分利用。例如，一个实际网络中的主机数为 250 台左右时，申请一个 C 类地址最为合理。若实际的网络规模较小，如只有 30 台左右的主机时，独自占用一个 C 类地址会造成地址资源的浪费，较合理的做法是将一个 C 类地址分给若干个小的网络共同使用。具体办法是从 hostID 域中借用某几位高位作为子网的 subnetID 域，形成如图 3.14 所示的三级地址。

这样，网关（连接物理网络的路由器）的路由表就被分成两级：先识别由 NetID 标识的路由，以确定一个逻辑的网络；再在该逻辑的网络内部用 SubnetID 来确定具体的子网。

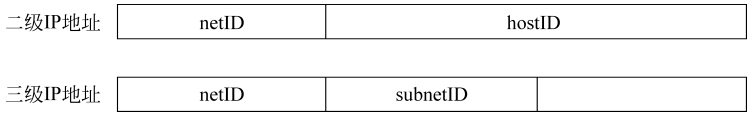


图 3.14　三级 IP 地址结构

2. 子网掩码与子网划分

三级 IP 地址提供了有效的、灵活的使用 IP 地址资源的手段。但是，带来的问题是：路由器如何判断到底一个 IP 地址中有多少位用于子网编码呢？在二级地址中，可以用类型标

识来判断这个 IP 地址是哪类地址，得到 netID 有多少位的信息。在三级 IP 地址中，虽然还有这个信息，但没有 subnetID 是多少位的信息了。为此必须另外提供一个关于网络码有多少位的信息，这个信息用子网掩码（sunnet mask）表示。

子网掩码是一种 32 位的位模式。RFC 文档中规定：每个使用子网的结点都选择一个 32 位的位模式，用这个位模式的 1 位来对应 IP 地址中网络地址（包括 netID 和 subnetID）中的一位，用位模式中的 0 位来对应 IP 地址中主机地址中的一位。虽然，RFC 文档中没有强求子网掩码中的 1 和 0 必须连续，但这种连续的 1 和 0，可以清晰地表明哪一段是网络地址，哪一段是主机地址。

子网掩码的作用是进行子网划分。划分的方法是对 IP 地址与子网掩码进行“与”运算，得到的就是子网的网络地址。图 3.15 为一个子网划分的实例。

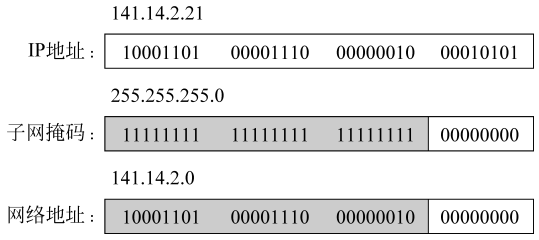


图 3.15　子网划分实例

例 3.1　将一个有 256 台主机、网络号为 200.15.192 的 C 类型网络分为两个相同的各拥有 128 台主机的子网。

在网络号为 200.15.192 的 C 类网络中，主机的编号为 200.15.211.0～200.15.211.256。由于要将一个 C 类网络分为两个子网，因而要在其 HostID 域中借用最高一位，子网掩码由 C 类的默认掩码 255.255.255.0 变为 255.255.255.128，即

11111111.11111111.11111111.10000000

以此类推，要划分为 4 个子网，应借用 2 位；要划分为 8 个子网，应借用 3 位……

例 3.2　对于一个 C 类网络 202.113.240，可以使用子网掩码 255.255.255.224 划分为 8 个子网，每个子网有 32 个 IP 地址：

202.113.240.0 ～202.113.240.31
202.113.240.32～202.113.240.63
⋮
202.113.240.224～202.113.240.255

应当注意，不管如何划分，一个网络中可容纳的主机总数不会增多。

RFC 950 成为 Internet 的正式标准后，子网掩码随之成为网络或子网的重要信息，所有的网络配置必须确定子网掩码，路由器在与相邻路由器交换路由信息时，必须同时传递子网掩码，即一个路由器连接几个网络，就有几个 IP 地址和相应的子网掩码。

3. 默认子网掩码

路由器要求子网掩码，因此当网络中没有子网时，A、B、C 三类网络就要使用如下默认子网掩码：