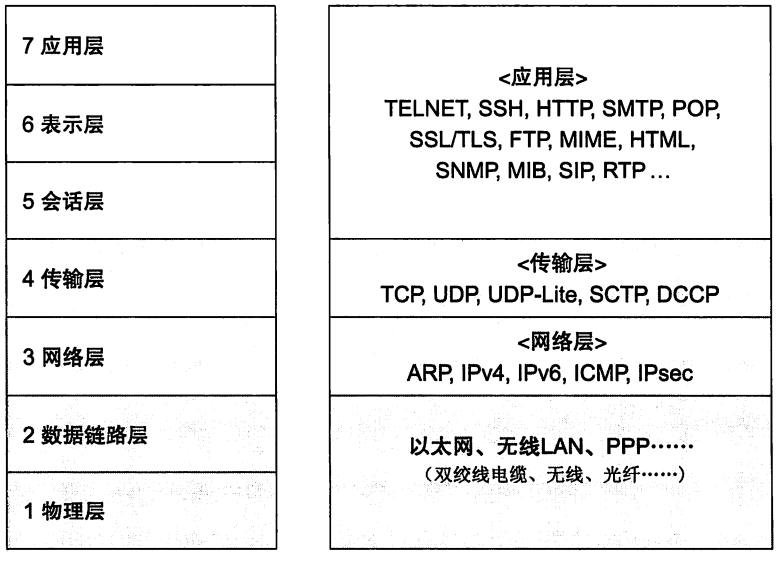
第8章

应用协议

一般情况下，人们不会太在意网络应用程序实际上是按照何种机制正常运行的。本章则旨在介绍TCP/IP中所使用的几个主要应用协议，它们多处于OSI模型的第5层以上。



8.1 应用层协议概要

到此为止所介绍的 IP 协议、TCP 协议以及 UDP 协议是通信最基本的部分，它们属于 OSI 参考模型中的下半部分。

从本章开始所要介绍的应用协议主要是指 OSI 参考模型中第 5 层、第 6 层、第 7 层上半部分的协议。

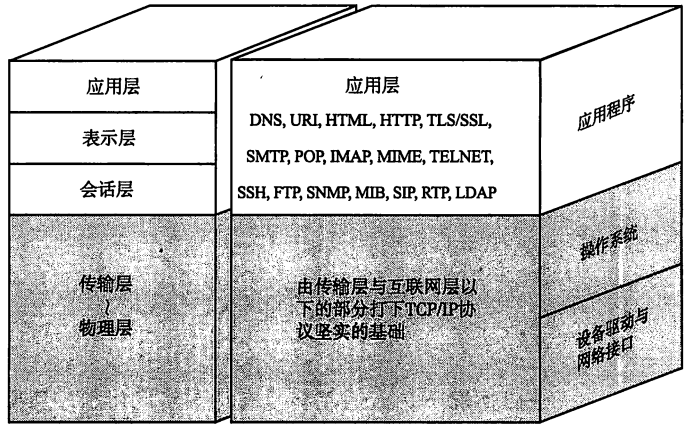


图 8.1 OSI 参考模型与 TCP/IP 的应用层

应用协议的定义

利用网络的应用程序有很多，包括 Web 浏览器、电子邮件、远程登录、文件传输、网络管理等。能够让这些应用进行特定通信处理的正是应用协议。

TCP 和 IP 等下层协议是不依赖于上层应用类型、适用性非常广的协议。而应用协议则是为了实现某种应用而设计和创造的协议。

例如，远程登录等应用经常使用的 TELNET 协议，它的支持基于文字的命令与应答，通过命令可以执行各种各样的其他应用。

应用协议与协议的分层

网络应用由不同的用户和软件供应商开发而成。为了实现网络应用的功能，在应用之间进行通信时将其连接的网络协议是非常重要的▼。设计师和开发人员根据所开发模块的功能和目的，可以利用现有的应用协议，也可以自己定义一个新的应用协议。

▼应用之间交互的信息叫消息。应用协议定义这些消息的格式以及使用这些消息进行控制或操作的规则。

应用可以直接享用传输层以下的基础部分。因为开发者只要关心选用哪种应用协议、如何开发即可，而不必担心应用中的数据该以何种方式发送到目标主机等问题。这也是得益于网络层的功劳。

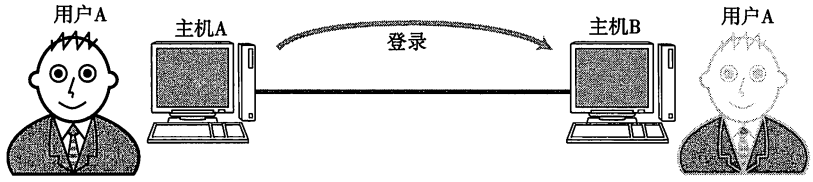
相当于 OSI 中第 5、第 6、第 7 层的协议

TCP/IP 的应用层涵盖了 OSI 参考模型中第 5、第 6、第 7 层的所有功能，不仅包含了管理通信连接的会话层功能、转换数据格式的表示层功能，还包括与对端主机交互的应用层功能在内的所有功能。

从下一节开始我们将逐一介绍几款经典的应用协议。

8.2 远程登录

图 8.2
远程登录



主机A的用户A通过远程登录到主机B，就好像坐在主机B跟前一样，可以利用主机B上各种功能。

远程登录是为了实现 TSS▼（曾在第 1 章介绍）环境，是将主机和终端的关系应用到计算机网络上的一个结果。TSS 中通常有一个处理能力非常强的主机，围绕着这台主机的是处理能力没有那么强的多个终端机器。这些终端通过专线与主机相连。

▼ TSS（Time Sharing System）分时系统。参考第 1 章。

类似地，实现从自己的本地计算机登录到网络另一端计算功能的应用就叫做远程登录。通过远程登录到通用计算机或 UNIX 工作站以后，不仅可以直接使用这些主机上的应用，还可以对这些计算机进行参数设置。远程登录主要使用 TELNET 和 SSH▼ 两种协议。

▼ Secure SHell。

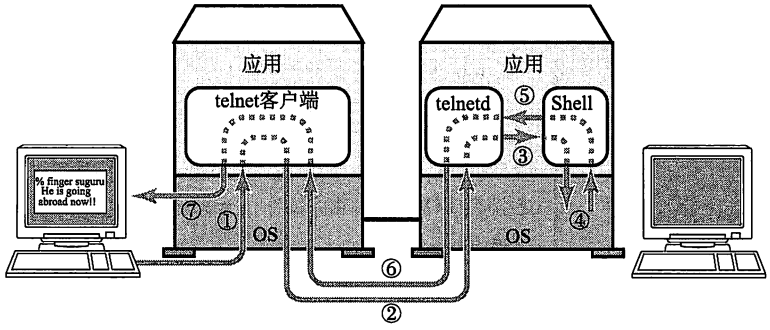
8.2.1 TELNET

TELNET 利用 TCP 的一条连接，通过这一条连接向主机发送文字命令并在主机上执行。本地用户好像直接与远端主机内部的 Shell▼ 相连着似的，直接在本地进行操作。

▼ Shell 是操作系统提供给用户的、便于使用该系统中各种功能的一种用户接口。它可以解释用户从键盘或鼠标输入的内容，并让操作系统执行。UNIX 中的 sh、csh、bash 和 Windows 中的 Expolorer 以及 MAC OS 的 Finder 等都属于同一范畴。

TELNET 可以分为两类基本服务。一是仿真终端功能，二是协商选项机制。

图 8.3
TELNET 中输入命令、运行、展示结果的过程



① 键入文字命令
② 进行行模式或透明模式处理后将前一步中的命令传送给telnetd守护进程。
③ 向Shell发送文字命令（严格来说这一步要经过操作系统内部）
④ 解释从Shell收到的命令、执行程序、获取结果
⑤ 获取从Shell返回的结果（严格来说这一步要经过操作系统内部）
⑥ 进行行模式或透明模式等处理后将结果返回给TELNET客户端。
⑦ 根据NVT的设置回显在屏幕上。

TELNET 经常用于登录路由器或高性能交换机等网络设备进行相应的设置▼。

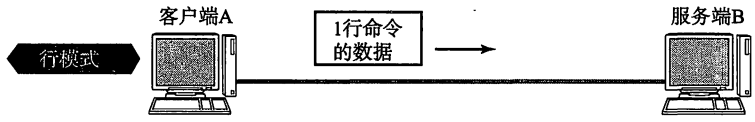
▼由于路由器和交换机一般都不配备键盘和显示器，因此对它们进行设置时可以通过串行线连接计算机，也可以通过使用 TELNET、HTTP、SNMP 等方法连接网络。

通过 TELNET 登录主机或路由器等设备时需要将自己的登录用户名和密码注册到服务端。

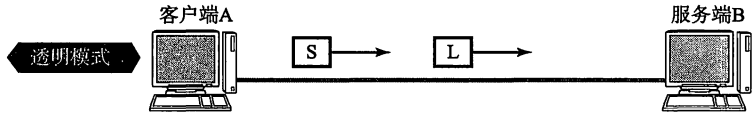
选项

TELNET 中除了处理用户所输入的文字外，还提供选项的交互和协商功能。例如，为实现仿真终端（NVT，Network Virtual Terminal）所用到的界面控制信息就是通过选项功能发送出去的。而且，如图 8.4 所示 TELNET 中的行模式或透明模式两种模式的设置，也是通过 TELNET 客户端与 TELNET 服务端之间的选项功能进行设置的。

图 8.4
行模式与透明模式



每从键盘输入一个换行，就将该行的数据作为1整行发送给服务端B。



客户端A每输入一个字符都要发送给服务端B。

TELNET 客户端

所谓 TELNET 客户端是指利用 TELNET 协议实现远程登录的客户端程序。很多情况下，它的程序名就是 telnet 命令。

TELNET 客户端通常与目标主机的 23 号端口建立连接，并与监听这个端口的服务端程序 telnetd 进行交互。当然，也可以与其他的 TCP 端口号连接，只要在该端口上有监听程序能够处理 telnet 请求即可。在一般的 telnet 命令▼中可以按照如下格式指定端口号▼：

telnet 主机名　TCP 端口号

TCP 端口号为 21 时可以连接到 FTP（8.3 节）应用，为 25 时可以连接到 SMTP（8.4.4 节），为 80 时可连接到 HTTP（8.5 节），为 110 时可连接到 POP3（8.4.5 节）。如此看来，每个服务器都有相应的端口号在等待连接。

因此，以下两个命令可以视为相同：

ftp 主机名
telnet 主机名 21

鉴于 FTP、SMTP、HTTP、POP3 等协议的命令和应答都是字符串，因此通过 TELNET 客户端连接以后可以直接输入这些协议的具体命令。TELNET 客户端也可用于跟踪 TCP/IP 应用开发阶段的问题诊断。

▼在 Windows 的命令行里输入 telnet 命令执行的操作方法并未在本节列出。用户可以通过输入 telnet 以后，在 telnet 的命令行里再输入 "open 主机名　端口号" 的方式进行连接。但是，从 Windows Vista 系统以后命令行的 telnet 功能默认是关闭的，需要单独安装才能使用。

▼在使用 GUI 类型客户端的情况下可以通过设置菜单等命令修改所要连接的端口号。

8.2.2 SSH

SSH 是加密的远程登录系统。TELNET 中登录时无需输入密码就可以发送，容易造成通信窃听和非法入侵的危险。使用 SSH 后可以加密通信内容。即使信息被窃听也无法破解所发送的密码、具体命令以及命令返回的结果是什么。

SSH 还包括很多非常方便的功能：

- 可以使用更强的认证机制。
- 可以转发文件▼。
- 可以使用端口转发功能▼。

▼ UNIX 中可以使用 scp、sftp 等命令。

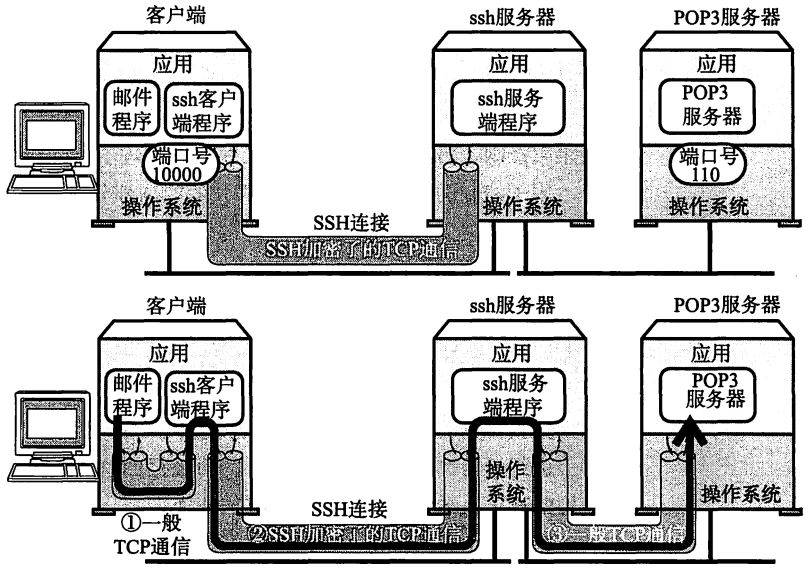
▼可以通过 X Window System 串口展现。

端口转发是指将特定端口号所收到的消息转发到特定的 IP 地址和端口号码的一种机制。由于经过 SSH 连接的那部分内容被加密，确保了信息安全，提供了更为灵活的通信▼。

▼可以实现虚拟专用网（VPN，Virtual Private Network）。

使用端口转发的情况下，SSH客户端程序、SSH服务端程序都起着一个网关的作用。下图设置了当连接到客户端的TCP端口10000时，设定连接到POP3服务器的端口110的情况。

图 8.5 SSH 的端口转发



邮件程序使用“一般TCP通信”连接ssh客户端。
SSH客户端则通过“SSH加密的通信”转发给SSH服务端程序。
SSH服务端程序使用“一般TCP通信”连接POP3服务端程序。

就这样，通过建立3个TCP连接进行整个通信。

8.3 文件传输

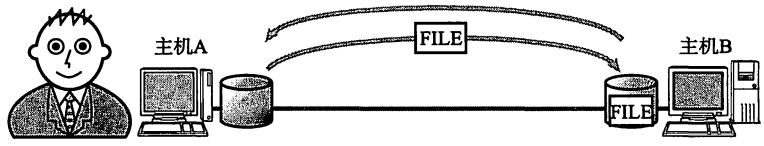


图 8.6
文件传输 FTP

网络上相连的两台主机之间可以进行文件传输。

FTP 是在两个相连的计算机之间进行文件传输时使用的协议。在 8.2 节中，我们已经讲过“远程登录”的概念，FTP 中也需要在登录到对方的计算机后才能进行相应的操作。

互联网上有一种 FTP 服务器是允许任何人进行访问的，这种服务器叫做匿名服务器（anonymous ftp）。登录这些服务器时使用匿名（anonymous）或 ftp 都可以▼。

▼习惯上该用户的密码为电子邮件地址的情况居多。

FTP 的工作机制概要

FTP 是通过怎样的机制才得以实现文件传输的呢？它使用两条 TCP 连接：一条用来控制，另一条用于数据（文件）的传输。

用于控制的 TCP 连接主要在 FTP 的控制部分使用。例如登录用户名和密码的验证、发送文件的名称、发送方式的设置。利用这个连接，可以通过 ASCII 码字符串发送请求和接收应答（如表 8.1、表 8.2 所示）。在这个连接上无法发送数据，数据需要一个专门的 TCP 进行连接。

FTP 控制用的连接使用的是 TCP21 号端口。在 TCP21 号端口上进行文件 GET（RETR）、PUT（STOR）、以及文件一览（LIST）等操作时，每次都会建立一个用于数据传输的 TCP 连接。数据的传输和文件一览表的传输正是在这个新建的连接上进行。当数据传送完毕之后，传输数据的这条连接也会被断开，然后会在控制用的连接上继续进行命令或应答的处理。

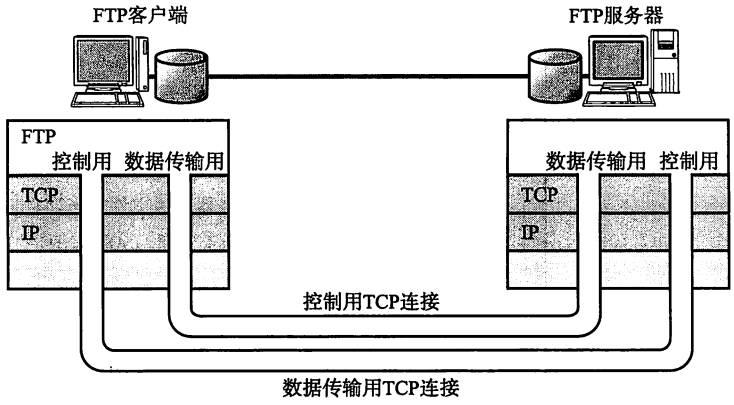
通常，用于数据传输的 TCP 连接是按照与控制用的连接相反的方向建立的。因此，在通过 NAT 连接外部 FTP 服务器的时候，无法直接建立传输数据时使用的 TCP 连接。此时，必须使用 PASV 命令修改建立连接的方向才行。

控制用的连接，在用户要求断开之前会一直保持连接状态。不过，绝大多数 FTP 服务器都会对长时间没有任何新命令输入的用户的连接强制断开▼。

▼在文件传输过程中不会断开连接，而是在文件已经传输完成以后，一段时间没有任何其他命令输入，则断开连接。

数据传输用的 TCP 连接通常使用端口 20。不过可以用 PORT 命令修改为其他的值。最近，出于安全的考虑，普遍在数据传输用的端口号中使用随机数进行分配。

图 8.7
FTP 通信中使用两条 TCP 连接



通过 ASCII 码字符串进行的交互处理

▼ American Standard Code for Information Interchange 的省略。

FTP 中请求命令中使用着“RETR”等 ASCII▼ 码字符串。而针对这些命令的应答则使用如“200”等 3 位数字的 ASCII 码字符串。TCP/IP 的应用协议中有很多使用这种 ASCII 码字符串的协议。

对于 ASCII 码字符串型的协议来说换行具有重要意义。很多情况下，一行字符串表示一个命令或一个应答，而空白则用来标识与参数之间的分割符。即，命令和应答的消息通过换行区分、参数用空格区分。换行由“CR”（ASCII 码的十进制数为 13）和“LF”（ASCII 码的十进制数为 10）两个控制符号组成。

表 8.1 列出了 FTP 主要的命令、表 8.2 汇总了 FTP 的应答信息。

表 8.1
FTP 主要命令

访问控制命令	
USER 用户名	输入用户名
PASS 密码	输入密码（PASSWORD）
CWD 目录名	修改工作目录（CHANGE WORKING DIRECTORY）
QUIT	正常结束

设置传输参数的命令	
PORT h1，h2，h3，h4，p1，p2	指定数据传输时使用的 IP 地址和端口号
PASV	不是从服务器端向客户端建立连接，而是由客户端开始向服务器端建立数据传输用的连接（PASSIVE）
TYPE 类型名	设置发送和接收的数据类型
STRU	指定文件结构（FILE STRUCTURE）

FTP 服务命令	
RETR 文件名	从 FTP 服务器下载文件（RETRIEVE）
STOR 文件名	向服务器上传文件（STORE）
STOU 文件名	向服务器发送文件。当存在同名文件时，为了避免冲突，适当地修改当前文件名后再上传（STORE UNIQUE）
APPE 文件名	向服务器发送文件。当存在同名文件时，将当前文件内容追加到已有文件（APPEND）
RNFR 文件名	指定 RNTO 之前要修改名称的文件（RENAME FROM）
RNTO 文件名	修改由 RNFR 指定文件的文件名（RENAME TO）
ABOR	处理中断，异常退出（ABORT）
DELE 文件名	从服务器上删除指定文件（DELETE）
RMD 目录名	删除目录（REMOVE DIRECTORY）
MKD 目录名	创建目录（MAKE DIRECTORY）
PWD	列出当前目录位置（PRINT WORKING DIRECTORY）
LIST	文件列表的请求（包括文件名，大小，更新日期等信息）
NLST	文件名一览表请求（NAME LIST）
SITE 字符串	执行服务器提供的特殊命令
SYST	获取服务器操作系统的信息（SYSTEM）
STAT	显示服务器 FTP 的状态（STATUS）
HELP	命令帮助（HELP）
NOOP	无操作（NO OPERATION）

表 8.2
FTP 的主要应答消息

提供信息	
120	Service ready in *nnn* minutes.
125	Data connection already open; transfer starting.
150	File status okay; about to open data connection.

连接管理相关应答	
200	Command okay.
202	Command not implemented, superfluous at this site.
211	System status, or system help reply.
212	Directory status.
213	File status.
214	Help message.
215	NAME system type. Where NAME is an official system name from the list in the Assigned Numbers document.

（续）

连接管理相关应答	
220	Service ready for new user.
221	Service closing control connection. Logged out if appropriate.
225	Data connection open; no transfer in progress.
226	Closing data connection. Requested file action successful.
227	Entering Passive Mode (h1, h2, h3, h4, p1, p2).
230	User logged in, proceed.
250	Requested file action okay, completed.
257	"PATHNAME" created.

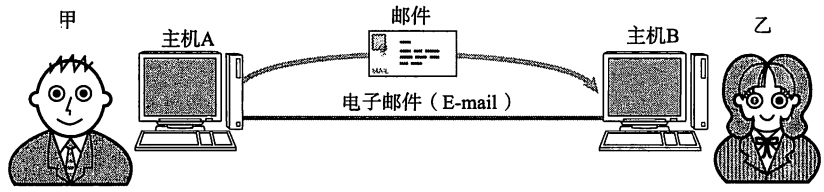
验证与用户相关应答	
331	User name okay, need password.
332	Need account for login.
350	Requested file action pending further information.

不固定的错误	
421	Service not available, closing control connection. This may be a reply to any command if the service knows it must shut down.
425	Can't open data connection.
426	Connection closed; transfer aborted.
450	Requested file action not taken. File unavailable.
451	Requested action aborted: local error in processing.
452	Requested action not taken. Insufficient storage space in system.

文件系统相关应答	
500	Syntax error, command unrecognized.
501	Syntax error in parameters or arguments.
502	Command not implemented.
503	Bad sequence of commands.
504	Command not implemented for that parameter.
530	Not logged in.
532	Need account for storing files.
550	Requested action not taken. File unavailable.
551	Requested action aborted: page type unknown.
552	Requested file action aborted. Exceeded storage allocation.
553	Requested action not taken. File name not allowed.

8.4 电子邮件

图8.8
电子邮件（E-mail）



只要连着网，即使相隔很远，也可以发送邮件。

电子邮件，顾名思义，就是指网络上的邮政。通过电子邮件人们可以发送编写的文字内容、数码相片，还可以发送各种报表计算得出的数据等所有计算机可以存储的信息。

电子邮件的发送距离不受限，可以在全世界互联网中的任何两方之间进行收发。如果没有电子邮件，出差时也就无法接收最新的邮件信息。电子邮件还可以提供邮件组的服务。它是指向邮件组中的所有用户同时发送邮件的功能。邮件组现在被广泛用于公司或学校下达通知、不同国度的人们讨论共同的话题等场景。出于以上这些优点，电子邮件已经成为当前人们普遍使用的一种服务。

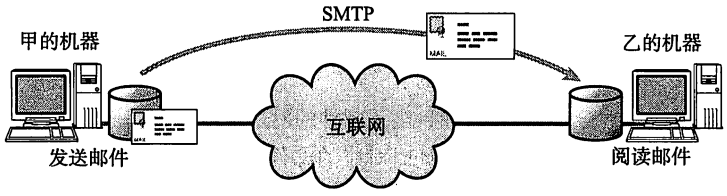
8.4.1 电子邮件的工作机制

提供电子邮件服务的协议叫做 SMTP（Simple Mail Transfer Protocol）。SMTP 为了实现高效发送邮件内容，在其传输层使用了 TCP 协议。

早期电子邮件是在发送端主机与接收端主机之间直接建立 TCP 连接进行邮件传输。发送人编写好邮件以后，其内容会保存在发送端主机的硬盘中。然后与对端主机建立 TCP 连接，将邮件发送到对端主机的硬盘。当发送正常结束后，再从本地硬盘中删除邮件。而在发送过程中一旦发现对端计算机因没有插电等原因没有收到邮件时，发送端将等待一定时间后重发。

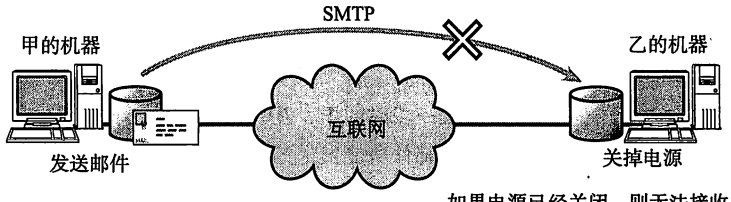
这种方法，在提高电子邮件的可靠性传输上非常有效。但是，互联网应用逐渐变得越发复杂，这种机制也将无法正常工作。例如，使用者的计算机时而关机时而开机的情况下，只有发送端和接收端都处于插电并且开机的状态时才可能实现电子邮件的收发。由于日本属于东九时区，和美国之间存在时差。日本的白天相当于美国的夜晚。如果大家都是只在白天开机，那么日本跟美国之间就根本无法实现收发邮件。由于互联网是一个连接全世界所有人进行通信的网络，所以这种时差问题就不得不考虑在内。

图 8.9
早期的电子邮件发送过程



早期的电子邮件，发送端主机与接收端主机之间会建立一个直接的TCP连接，再进行邮件的收发。

然而，这种方法要求两端主机都必须插电，且一直处于连网的状态才行，否则可能会收不到邮件。

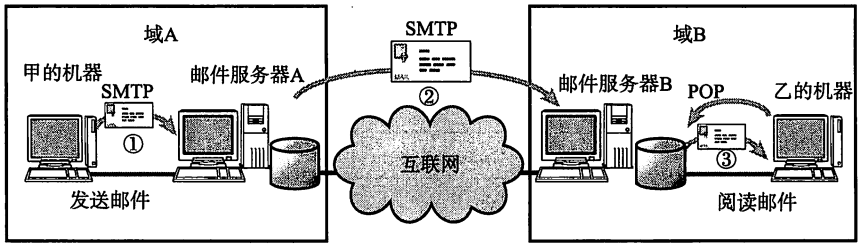


当无法与对端主机进行通信时，发送端会稍等一段时间后尝试重发。然而，如果发送端主机已经拔掉电源，那么在它再次插电之前邮件将无法发送出去。

如果电源已经关闭，则无法接收邮件。
主机如果没有连接互联网，也就无法接收邮件。

图 8.10
现在互联网中电子邮件的发送过程

经过邮件服务器发送邮件。每个域根据需要在不同阶段设置邮件服务器。



① 根据邮件软件的设置，发送邮件给邮件服务器A。
② 参考DNS的MX记录，发送邮件给邮件服务器B。
③ 根据邮件软件的设置，从邮件服务器B接收邮件。

为此，在技术上改变了以往直接在发送端与接收端主机之间建立 TCP 连接的机制，而引进了一种一直会连接电源的邮件服务器▼。发送和接收端通过邮件服务器进行收发邮件。接收端从邮件服务器接收邮件时使用 POP3（Post Office Protocol）协议。

▼由于在传输层以上的网络中对通信进行转播，因此邮件服务器相当于 1.9.7 节中介绍的网关。

电子邮件的机制由 3 部分组成，它们分别是邮件地址，数据格式以及发送协议。

8.4.2 邮件地址

使用电子邮件时需要拥有的地址叫做邮件地址。它就相当于通信地址和姓名。互联网中电子邮件地址的格式如下：

名称@通信地址

例如，master@tcpip.kusa.ac.jp 中的 master 为名称，tcpip.kusa.ac.jp

为地址。电子邮件的地址和域名的构造相同。此处，kusa. ac. jp 表示域名，tcpip 则表示 master 接收邮件的主机名称或为发送邮件所用的子网名称。现在个人邮件地址和邮件组的格式完全相同，因此，光从地址上是无法区分个人电子邮件地址和邮件组的。

现在，电子邮件的发送地址由 DNS 进行管理。DNS 中注册有邮件地址及其作为发送地址时对应的邮件服务器的域名。这些映射信息被称作 MX 记录。例如，kusa. ac. jp 的 MX▼ 记录中指定了 mailserver. kusa. ac. jp。于是任何发给以 kusa. ac. jp 结尾的地址的邮件都将被发送到 mailserver. kusa. ac. jp 服务器。就这样，根据 MX 记录中指定的邮件服务器，可以管理不同邮件地址与特定邮件服务器之间的映射关系。

▼ Mail Exchange

8.4.3 MIME

很长一段时间里，互联网中的电子邮件只能处理文本格式的▼邮件。不过现在，电子邮件所能发送的数据类型已被扩展到 MIME▼，可以发送静态图像、动画、声音、程序等各种形式的数据。鉴于 MIME 规定了应用消息的格式，因此在 OSI 参考模型中它相当于第 6 层表示层。

▼由文字组成的信息。过去的电子邮件，就日本来说人们只能发送 7 比特 JIS 编码的信息。

▼ Multipurpose Internet Mail Extensions，广泛用于互联网并极大地扩展了数据格式，还可以用于 WWW 和 NetNews 中。

MIME 基本上由首部和正文（数据）两部分组成。首部不能是空行，因为一旦出现空行，其后的部分将被视为正文（数据）。如果 MIME 首部的“Content-Type”中指定“Multipart/Mixed”，并以“boundary=”后面字符作为分隔符▼，那么可以将多个 MIME 消息组合成为一个 MIME 消息。这就叫做 multipart。即，各个部分都由 MIME 首部和正文（数据）组成。

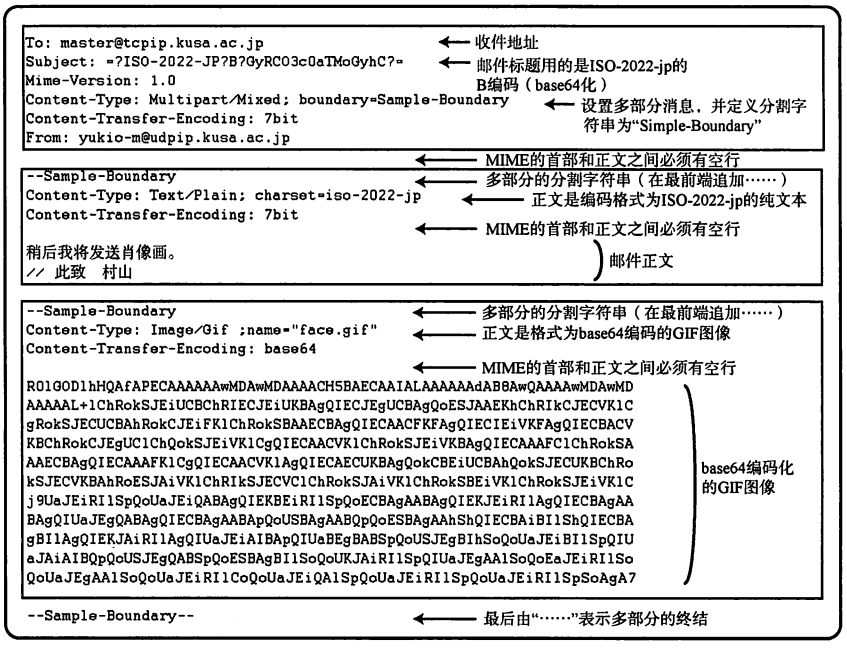
▼ boundary = 后面的字符串，开头一定要写 - -。而且，间隔符后面也一定要写 - -。

“Content-Type”定义了紧随首部信息的数据类型。以 IP 首部为例，它就相当于协议字段。表 8.3 列出了具有代表性的“Content-Type”。

表 8.3　MIME 具有代表性的 Content-Type

Content-Type	内　容
text/plain	纯文本
message/rfc822	MIME 与正文
multipart/mixed	多部分消息
application/postscript	PostScript
application/octet-stream	二进制数据
image/gif	GIF 图像
image/jpeg	JPEG 图像
audio/basic	AU 格式的音频文件
video/mpeg	MPEG 动画
message/external-body	包含外部消息

图 8.11
MIME 举例



8.4.4 SMTP

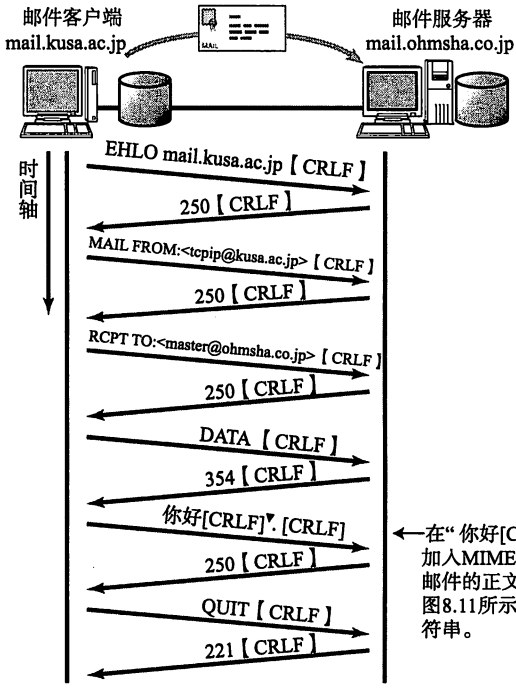
SMTP 是发送电子邮件的协议。它使用的是 TCP 的 25 号端口。SMTP 建立一个 TCP 连接以后，在这个连接上进行控制和应答以及数据的发送。客户端以文本的形式发出请求，服务端返回一个 3 位数字的应答。

每个指令和应答的最后都必须追加换行指令（CR、LF）。

表 8.4
SMTP 主要的命令

HELO <domain>	开始通信
EHLO <domain>	开始通信（扩展 HELO）
MAIL FROM：<reverse-path>	发送人
RCPT TO：<forward-path>	接收人（Receipt to）
DATA	发送电子邮件的正文
RSET	初始化
VRFY <string>	确认用户名
EXPN<string>	将邮件组扩展为邮件地址列表
NOOP	请求应答（NO Operation）
QUIT	关闭

图 8.12
SMTP



▼SMTP以“.”作为邮件正文的结束符，即使正文本身含有这个字符，也能做出识别。具体处理方法为，如果邮件正文的行首有“.”字符时，会在其后面紧接着再追加一个“.”字符。接收邮件时如果行首出现两个“.”字符，则删除其中一个。

随着电子邮件使用的普及，那些漫天的广告邮件和包含钓鱼连接的垃圾邮件成为了日益严重的问题。由于SMTP本身没有验证发送者的功能，因此人们无法避免这类邮件到达自己的邮件服务器。不过现在，通过“POP before SMTP”或“SMTP认证”（SMTP Authentication）等功能进行认证，以此防止冒充发送者的人也越来越多。

▼这样叫OP25B（Outbound Port 25 Blocking）。如果出差地的酒店也进行OP25B的话，可能会导致无法发送邮件，此时一般会使用587端口（Submission Port）。（RFC6409）

并且很多除了自己本域的邮件服务器以外，很多供应商已将网络设置为不与其他网络的25号端口进行通信▼。

表 8.5
SMTP 应答

针对请求进行肯定确认应答	
211	系统状态或求助回答
214	求助信息
220 <domain>	服务就绪
221 <domain>	服务结束
250	完成请求命令
251	非本地用户，报文将被转发

数据输入	
354	开始邮件输入。以“.”结束一行

发送错误消息	
421 〈domain〉	服务不可用，关闭连接
450	邮箱不可用
451	命令异常终止：本地差错
452	命令异常终止：存储容量不足

无法继续处理的错误应答	
500	语法错误，不能识别的命令
501	语法错误，不能识别参数或变量
502	命令未实现
503	命令序列不正确
504	命令参数暂时未实现
550	邮箱不可用，请求未实现
551	非本地用户，不接受请求
552	存储容量不足，请求异常终止
553	邮箱不可用，请求异常终止
554	其他错误

试用 SMTP 命令

当允许使用 TELNET 登录 SMPT 服务器时，可如表 8.5 的形式在登录▼ SMTP 服务器后输入命令。

telnet 服务器名或其 IP 地址 25

假定自己是 SMTP 客户端，那么在执行 SMTP 相关命令以后可以收到如表 8.5 所示的应答信息。通过这样的尝试可以加深对 SMTP 协议中各个动作的理解。

▼关于 telnet 命令的使用方式可以参考 8.2.1 节的最后部分。

8.4.5 POP

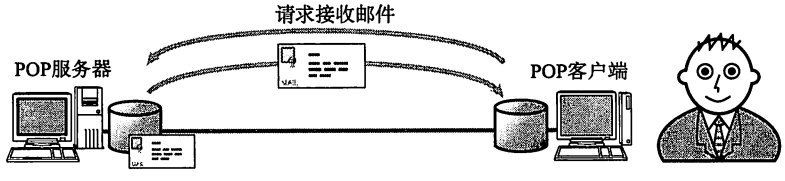


图 8.13 POP

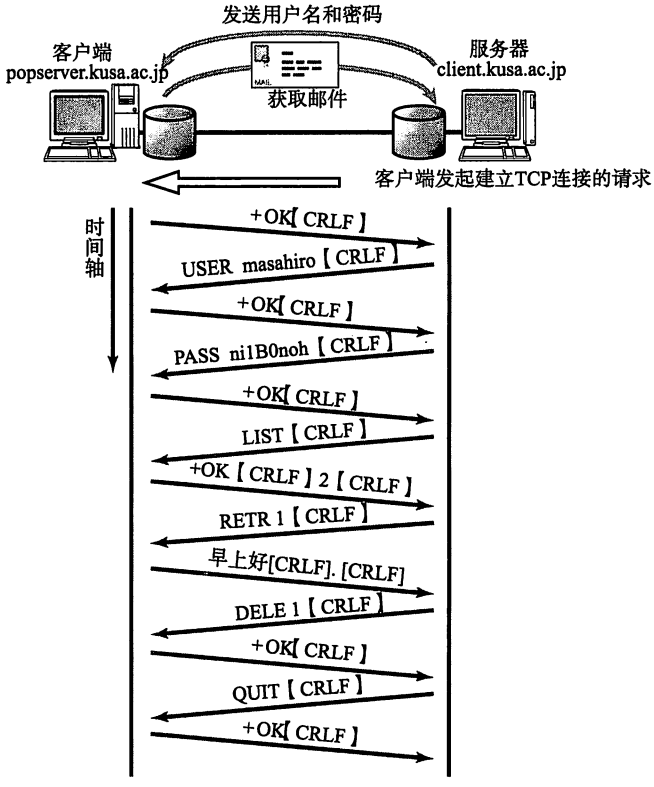
前一节提到的 SMTP 是发送邮件的协议，即，SMTP 是想要发送邮件的计算机向接收邮件的计算机发送电子邮件的一种协议。在以 UNIX 工作站为主的互联网初期，这种机制没有什么问题，但是后来用个人电脑连接互联网的环境中就出现

很多不便之处。

个人电脑不可能长时间处于开机状态。只有用户在使用时才会开机。在这种情况下，人们希望一开机就能接收到邮件。然而 SMTP 没有这种处理机制。SMTP 的一个不利之处就在于它支持的是发送端主机的行为，而不是根据接收端的请求发送邮件。

为了解决这个问题，就引入了 POP 协议。如图 8.14 所示，该协议是一种用于接收电子邮件的协议。发送端的邮件根据 SMTP 协议将被转发给一直处于插电状态的 POP 服务器。客户端再根据 POP 协议从 POP 服务器接收对方发来的邮件。在这个过程中，为了防止他人盗窃邮件内容，还要进行用户验证。

图 8.14
POP 的工作机制



POP 与 SMTP 一样，也是在其客户端与服务器之间通过建立一个 TCP 连接完成相应操作。POP 的具体命令和相关应答代码如表 8.6 所示。它的命令都是较短的 ASCII 码字符串，应答更是极其简单，只有两种。正常的情况下为“+OK”，发生错误或异常的情况下为“-ERR”。

表 8.6
POP 主要命令

认证时的有效命令	
USER 用户名	发送用户名
PASS 密码	发送密码
QUIT	通信结束
APOP name digest	认证

应答	
+OK	正常时
-ERR	发生错误时

事务状态命令	
STAT	状态通知
LIST [msg]	确认指定邮件大小（获取一览表）
RETR [msg]	取得邮件信息
DELE [msg]	删除服务器中保存的邮件（QUIT 命令执行时才真正删除）
RSET	撤销所有的 DELE 命令，通信结束
QUIT	执行 DELE 命令，终止通信
TOP msg *n*	只要邮件的前 *n* 行内容
UIDL [msg]	获得该邮件的唯一标识

试用 POP 命令

当允许使用 TELNET 登录 POP 服务器时，在以如下形式登录▼ POP 服务器后，可以手工执行表 8.6 所列的命令。

telnet 服务器名或其 IP 地址 110

与前一节的 SMTP 一样，假定自己是 POP 客户端，在执行 POP 相关命令以后可以收到相应的应答信息。

▼关于 telnet 命令的使用方式可以参考 8.2.1 节的最后部分。

8.4.6 IMAP

▼ Internet Message Access Protocol

IMAP▼ 与 POP 类似，也是接收电子邮件的协议。在 POP 中邮件由客户端进行管理，而在 IMAP 中邮件则由服务器进行管理。

使用 IMAP 时，可以不必从服务器上下载所有的邮件也可以阅读。由于 IMAP 是在服务器端处理 MIME 信息，所以它可以实现当某一封邮件含有 10 个附件时“只下载其中的第 7 个附件”的功能▼。这在带宽较窄的线路上起着非常重要的作用。而且 IMAP 在服务器上对“已读/未读”信息和邮件分类进行管理，因此，即使在不同的计算机上打开邮箱，也能保持同步，使用起来非常方便▼。如此一来，使用 IMAP，在服务器上保存和管理邮件信息，就如同在自己本地客户端的某个闪存中管理自己的信息一样简单。

▼在 POP 中无法下载某个特定的附件。因此想要确认附件时就不得不下载邮件中所有的附件。

▼ POP 虽然也可以支持在多台计算机中下载邮件内容，但是未读信息和邮箱分组只能在每台计算机的软件中各自进行管理。

有了 IMAP 人们就可以通过个人电脑、公司的电脑、笔记本电脑以及智能手机等连接到 IMAP 服务器以后进行收发邮件。由此，在公司下载的电子邮件就不必在笔记本电脑和智能手机上转来转去▼。IMAP 确实为使用多种异构终端的人们提供了非常便利的环境。

▼不过笔记本电脑和智能手机必须能够连上 IMAP 服务器才行。

8.5 WWW

8.5.1 互联网的蓬勃发展

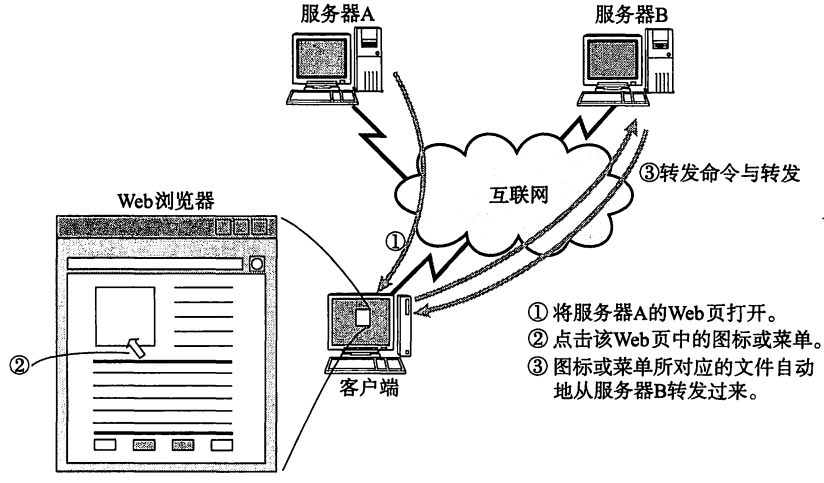
▼超文本用以显示文本及与文本相关的内容。

▼Web 浏览器（Web Browser），有时也简称为浏览器。

万维网（WWW，World Wide Web）是将互联网中的信息以超文本▼形式展现的系统。也叫做 Web。可以显示 WWW 信息的客户端软件叫做 Web 浏览器▼。目前人们常用的 Web 浏览器包括微软的 Internet Explorer、Mozilla 基金会的 Firefox、Google 公司的 Google Chrome、Opera 软件公司的 Opera 以及 Apple 公司的 Safari 等。

借助浏览器，人们不需要考虑该信息保存在哪个服务器，只需要轻轻点击鼠标就可以访问页面上的链接并打开相关信息。

图 8.15 WWW



通过浏览器进行访问后回显在浏览器中的内容叫做“Web 页”（或 WWW 页）。公司或学校等组织以及个人的 Web 页被称作主页。在日本，很多公司的主页地址形式如下：

```
http: //www. 公司名称. co. jp/
```

这一类主页当中通常会发布公司概况、产品信息、招贤纳士等内容。人们可以通过点击这些标题的图标或链接就可以跳到对应的页面上。而这些页面上所提供的信息不仅仅是文字内容，还有图片或动画乃至声音或其他程序等各式各样的信息。此外，通过 Web 页不仅可以获取信息，还可以通过自己制作 Web 页来向全世界发布信息。

8.5.2 WWW 基本概念

WWW 定义了 3 个重要的概念，它们分别是访问信息的手段与位置（URI，Uniform Resource Identifier）、信息的表现形式（HTML，HyperText Markup Language）以及信息转发（HTTP，HyperText Transfer Protocol）等操作。

8.5.3 URI

URI 是 Uniform Resource Identifier 的缩写，用于标识资源。URI 是一种可以用于 WWW 之外的高效的识别码，它被用于主页地址、电子邮件、电话号码等各种组合中。如下所示：

http：//www. rfc-editor. org/rfc/rfc4395. txt

http：//www. ietf. org：80/index. html

http：//localhost：631/

这些例子属于一般主页地址，也被叫做 URL（Uniform Resource Locator）。URL 常被人们用来表示互联网中资源（文件）的具体位置。但是 URI 不局限于标识互联网资源，它可以作为所有资源的识别码。现在，在有效的 RFC 文档中，已经不再使用 URL，转而在使用 URI▼。相比 URL 狭义的概念，URI 则是一个广义的概念。因此，URI 可以用于除了 WWW 之外的其他应用协议中。

▼它们之间好比比特跟字节的关系。协议定义中经常使用字节，但是在日常生活中却用比特较多。

URI 所表示的组合叫方案（Scheme）▼。在众多 URI 的 Scheme 中 WWW 主要用其中的 http 和 https 表示 Web 页的位置和访问 Web 页的方法。关于 URI Schema 一览表，请参考下面的文档。

▼ schema 是指具有体系的计划或方案。

http：//www. iana. org/assignments/uri-schemes. html

URI 的 http 方案的具体格式如下：

http：//主机名/路径

http：//主机名：端口号/路径

http：//主机名：端口号/路径？访问内容#部分信息

其中主机名表示域名或 IP 地址，端口号表示传输端口号。关于端口号的更多细节，读者可以参考 6.2 节。省略端口号时，则表示采用 http 的默认端口 80。路径是指主机上该信息的位置，访问内容表示要传给 CGI▼ 的信息，部分信息表示页面当中的位置等。

▼关于 CGI 请参考 8.5.6 节。

这种表示方法可以唯一地标识互联网中特定的数据。不过，由于用 http 方案展现的数据随时都有可能发生变化，所以即使将自己喜欢的页面的 URI（URL）记住，也不能保证下次是否还能够访问到该页。

表 8.7 列出了 URI 的主要方案。

表 8.7 主要的 URI 方案

方 案 名	内 容
acap	Application Configuration Access Protocol
cid	Content Identifier
dav	WebDAV
fax	Fax
file	Host-specific File Names
ftp	File Transfer Protocol
gopher	The Gopher Protocol
http	Hypertext Transfer Protocol

（续）

方 案 名	内　容
https	Hypertext Transfer Protocol Security
im	Instant Messaging
imap	Internet Message Access Protocol
ipp	Internet Printing Protocol
ldap	Lightweight Directory Access Protocol
mailto	Electronic Mail Address
mid	Message Identifier
news	USENET news
nfs	Network File System Protocol
nntp	USENET news using NNTP access
rtsp	Real Time Streaming Protocol
service	Service Location
sip	Session Initiation Protocol
sips	Secure Session Initiation Protocol
snmp	Simple Network Management Protocol
tel	Telephone
telnet	The Network Virtual Terminal Emulation Protocol
tftp	Trivial File Transfer Protocol
urn	Uniform Resource Names
z39.50r	Z39.50 Retrieval
z39.50s	Z39.50 Session

8.5.4　HTML

HTMP 是记述 Web 页的一种语言（数据格式）。它可以指定浏览器中显示的文字、文字的大小和颜色。此外，不仅可以对图像或动画进行相关设置，还可以设置音频内容。

HTML 具有纯文本的功能。在页面中不仅可以为文字或图像附加链接，当用户点击那些链接时还可以呈现该链接所指示的内容，因此它可以将整个互联网中任何一个 WWW 服务器中的信息以链接的方式展现。绝大多数互联网中的 Web 页，都以链接的形式指向关联的其他信息。逐一点开这些链接就可以了解全世界的信息。

HTML 也可以说是 WWW 通用的数据表现协议。即使是在异构的计算机上，只要是可以用 HTML 展现的数据，那么效果基本上是一致的。如果把它对应到 OSI 参考模型，那么可以认为 HTML 属于 WWW 的表示层▼。不过，鉴于现代计算机网络的表示层尚未完全准备就绪，根据操作系统和所用软件的不同，最终表现出来的效果也可能会出现细微差别。

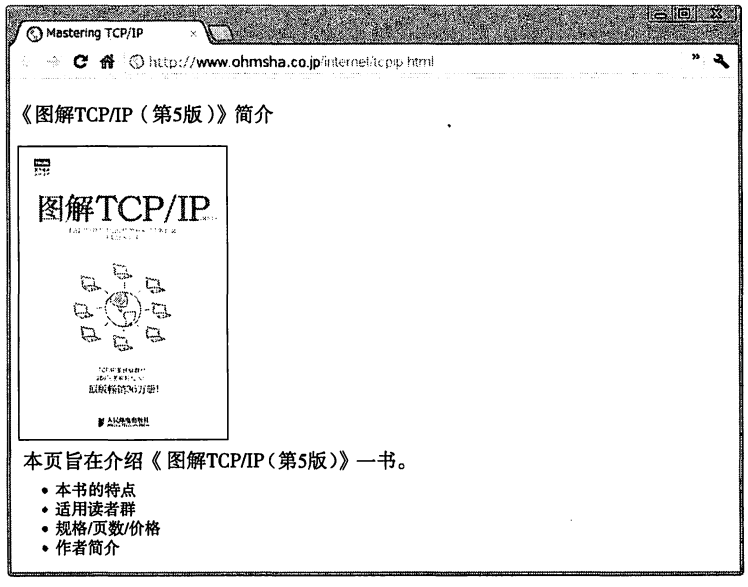
▼ HTML 不仅用于 WWW，有时还用于电子邮件。

图 8.16 展示了一个通过 HTML 表现数据样本的例子。如果将其用浏览器（例如 Firefox）打开的话，效果如图 8.17 所示。

图 8.16
HTML 举例

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html lang="ja">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Mastering TCP/IP</title>
</head>
<body>
<h1>《图解TCP/IP（第5版）》简介</h1>
<img src="cover.jpg" alt=图解TCP/IP（第5版）封面图片>
<p>本页旨在介绍《图解TCP/IP（第5版）》一书。</p>
<ul>
  <li><a href="feature.html">本书的特点</a></li>
  <li><a href="feature.html">适用读者群</a></li>
  <li><a href="feature.html">规格/页数/价格</a></li>
  <li><a href="feature.html">作者简介</a></li>
</ul>
</body>
</html>
```

图 8.17
用浏览器读取并显示图 8.16 的内容



XML 与 Java

WWW 中将数据存入文件或在应用之间进行交互时会经常使用 XML（Extensible Markup Language）。XML 是从 SGML▼ 衍生出来的一种语言，与 HTML 类似，也需要在每个项目的前后加入标签以表达其具体含义。一般，从<标签名>到</标签名>为止表示一个数据。

最近，开发人员经常结合 Java 与 XML 进行程序开发。原 SUN Microsystems 公司发明的 Java 是一种与平台无关的开发语言。而 XML 又是不依赖于任何软件供应商的数据格式。

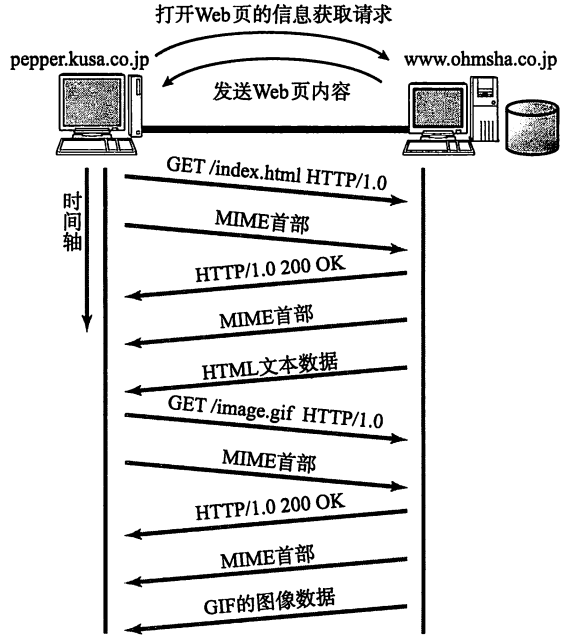
可以认为 Java 和 XML 都相当于 OSI 参考模型中的第 6 层表示层。这两者一结合，不论连接的是何种类型的网络，其应用上的动作效果能够保持一致。

▼ Standard Generalized Markup Language

8.5.5 HTTP

当用户在浏览器的地址栏里输入所要访问 Web 页的 URI 以后，HTTP 的处理即会开始。HTTP 中默认使用 80 端口。它的工作机制，首先是客户端向服务器的 80 端口建立一个 TCP 连接，然后在这个 TCP 连接上进行请求和应答以及数据报文的发送。

图 8.18
HTTP 的工作机制



HTTP 中常用的有两个版本，一个 HTTP1.0，另一个是 HTTP1.1。在 HTTP1.0 中每一个命令和应答都会触发一次 TCP 连接的建立和断开。而从 HTTP1.1 开始，允许在一个 TCP 连接上发送多个命令和应答▼。由此，大量地减少了 TCP 连接的建立和断开操作，从而也提高了效率。

▼这种方式也叫保持连接(keep-alive)。

表 8.8
HTTP 的主要命令以及应答报文

HTTP 的主要命令	
OPTIONS	设置选项
GET	获取指定 URL 的数据
HEAD	仅获取文档首部
POST	请求服务器接收 URI 指定文档作为可执行的信息
PUT	请求服务器保存客户端传送的数据到 URI 指定文档
DELETE	请求服务器删除 URI 指定页面
TRACE	请求消息返回客户端

信息提供	
100	Continue
101	Switching Protocols

肯定应答	
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content

重定向请求	
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified
305	Use Proxy

客户端请求内容出现错误	
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Time-out
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Large
415	Unsupported Media Type

服务器错误	
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Time-out
505	HTTP Version not supported

试用 HTTP 命令

当允许 HTTP 服务器和 TELNET 连接时，可以以如下形式登录▼HTTP 服务器后，再以手动形式执行表 8.8 所列的命令。

telnet 服务器名或其 IP 地址 80

假定自己是 HTTP 客户端，输入 ASCII 码字符串的命令，并确认表 8.8 中的应答结果。

▼关于 telnet 命令的使用方式可以参考 8.2.1 节的最后部分。

8.5.6 JavaScript、CGI、Cookie

JavaScript

Web 的基本要素为 URI、HTML 和 HTTP。然而仅有这些还无法更改与条件相符的动态内容。为此，通过在浏览器端和服务器端执行特定的程序可以实现更加精彩、多样的内容。例如实现网络购物或搜索功能。

我们称 Web 浏览器端执行的程序为客户端程序，在服务器端执行的程序为服务器端程序。

JavaScript 是一种嵌入在 HTML 中的编程语言，作为客户端程序可以运行于多种类型的浏览器中。这些浏览器将嵌入 JavaScript 的 HTML 下载后，其对应的 JavaScript 程序就可以在客户端得到执行。这种 JavaScript 程序用于验证客户端输入字符串是否过长、是否填写或选择了页面中的必须选项等功能▼。JavaScript 还可以用于操作 HTML 或 XML 的逻辑结构（DOM，Document Object Model）以及动态显示 Web 页的内容和页面风格上。最近，更是盛行服务器端不需要读取整个页面而是通过 JavaScript 操作 DOM 来实现更为生动的 Web 页面的技术。这就是 Ajax（Asynchronous JavaScrip and XML）技术。

▼如果将用户输入正确与否的验证都放在服务端执行的话，给服务器带来的负荷太大。因此只要能在客户端进行检查，就在客户端执行这样也可以保证效率。

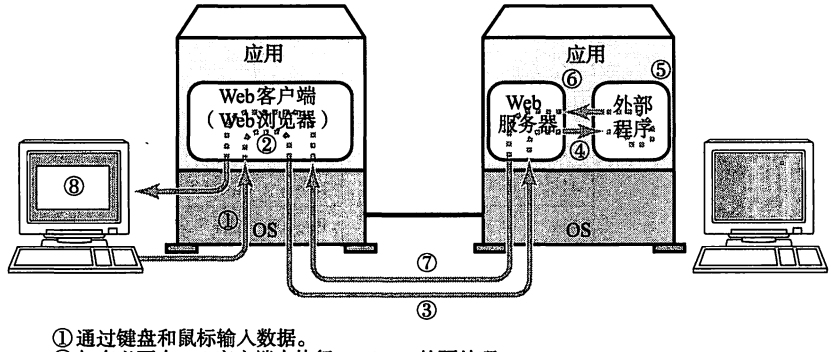


图 8.19　JavaScript、CGI 中的处理流程

①通过键盘和鼠标输入数据。
②如有必要在Web客户端内执行JavaScript的预处理。
　遇到错误时可以不进行HTTP的请求。
③发出HTTP请求将数据从Web客户端发送给Web服务器。
④使用CGI调用外部程序。
⑤程序执行。
⑥将外部程序应答结果返回给Web服务器。
⑦根据HTTP，再将应答报文从Web服务器发给Web客户端。
⑧显示处理结果。

CGI

▼ Common Gateway Interface

CGI▼ 是 Web 服务器调用外部程序时所使用的一种服务端应用的规范。

一般的 Web 通信中，只是按照客户端请求将保存在 Web 服务器硬盘中的数据转发而已。这种情况下客户端每次收获的信息也是同样（静态）的内容。而引入 CGI 以后客户端请求会触发 Web 服务器端运行另一个程序，客户端所输入的数据也会传给这个外部程序。该程序运行结束后会将生成的 HTML 和其他数据再返回给客户端▼。

▼外部程序并不仅局限于使用 CGI 启动，它也有可能被包含在 Web 服务器内部的程序里，或是嵌入了解释器的 Web 服务器程序里。

利用 CGI 可以针对用户的操作返回给客户端有各种各样变化（动态）的信息。论坛和网上购物系统中就经常使用 CGI 调用外部程序或访问数据库。

Cookie

Web 应用中为了获取用户信息使用一个叫做 Cookie 的机制。Web 服务器用 Cookie 在客户端保存信息▼（多为“用户名”和“登录名”等信息）。Cookie 常被用于保存登录信息或网络购物中放入购物车的商品信息。

▼还可以设置 Cookie 的有效期。

从 Web 服务器检查 Cookie 可以确认是否为同一对端的通信。从而存放于购物车里的商品信息就不必要在保存到服务器了。

博客与 RSS

博客（blog）是 weblog 的缩写。它是一种在使用者完全不懂 HTML、也不需要使用 FTP 的情况下，轻松建立 Web 页并更新内容的网络服务应用。常用于网络日记、报表等。

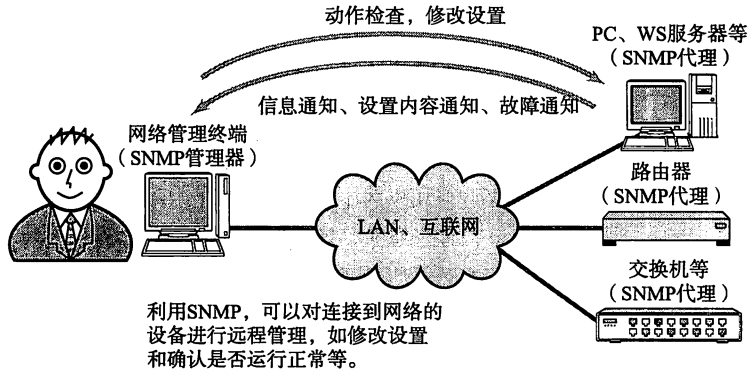
RSS 是用来交互与 Web 站点内容更新相关的摘要信息的一种数据格式，也叫做 Really Simple Syndication 或 RDF（Resource Description Framework）Site Summary。Web 上的数据看起来虽然比 HTML 等顺眼些。但是，通过这些数据，若要立即抽取该页面的概要信息或根据关键字自动集合显示那些自己感兴趣的页面，还是一件比较困难的事情。然而，如果使用 RSS，则可以将页面的标题、内容中的章节标题和概要、分类、关键字等信息记述下来，只显示页面的概要，提高关键字搜索的精度。作为发布消息为主的 Web 站点如果支持 RSS，那么用户可以轻松地通过 RSS 获取该站点的最新消息。

通过博客公开信息已经成为现代信息通信中不可阻挡的趋势。而 RSS 也将会成为人们从日益增多的互联网海量信息中收集自己感兴趣内容的必不可少的工具。

8.6 网络管理

8.6.1 SNMP

图8.20
网络管理



以前，网络管理都是凭借管理员的记忆和直觉进行。然而随着网络规模变得越来越大，个人的记忆、经验或直觉已经无法与之匹配，需要一个严密的管理工具或方法显得格外重要。在 TCP/IP 的网络管理中可以使用 SNMP（Simple Network Management Protocol）收集必要的信息。它是一款基于 UDP/IP 的协议。

SNMP 中管理端叫做管理器（Manager，网络监控终端），被管理端叫做代理（路由器、交换机等）▼。决定管理器与代理之间的通信中所要交互信息的正是 SNMP。SNMP 中如果将 MIB▼ 看做代理所管理的信息在数据库中的值，那么它可以新增一个值。

▼SNMPv3 中管理器和代理都叫做实体（Entity）。

▼关于 MIB（Management Information Base），请参考8.6.2节。

起初 SNMP 的安全机制并不完备。虽然在 SNMPv2 中有人提出过安全方面的建议，但是由于最终意见未能达成一致，所以支持基于团体认证方式的 SNMPv2c 成为了当时的标准。不过，该标准并没有采用安全机制。

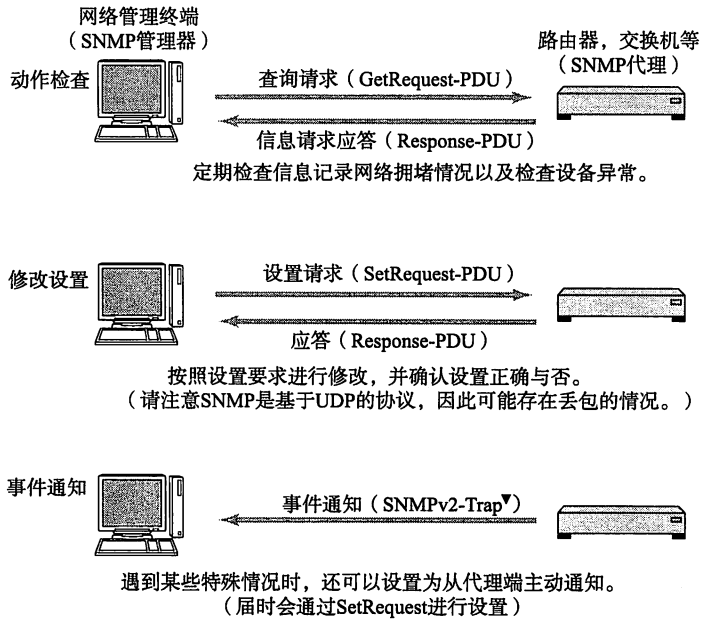
后来的 SNMPv3，不仅集合了所有 SNMP 的功能于同一个版本，定义了个别的功能模块（Component），并可以结合各种不同版本进行通信。

SNMPv3 中将“消息处理”、“用户安全”和“访问控制”三部分分开考虑，可以为每一个部选择各自必要的机制。

例如，在消息处理中除了有 SNMPv3 中所定义的处理模型以外，还有 SNMPv1 和 SNMPv2 的处理模型可供选择。实际上，在 SNMPv3 中选用 SNMPv2 的消息处理模型进行通信的情况居多。

消息处理中如果选择了 SNMPv2 的模型，那么会进行以下 8 种操作。它们分别是：查询请求，上次要求的下一个信息的查询请求（GetNextRequest-PDU）、应答、设置请求、批量查询请求（GetBulkRequest-PDU）、向其他管理器发送信息通知（InformRequest-PDU）、事件通知、用管理系统定义的命令（Report-PDU）等操作。

图 8.21
SNMP 工作机制



▼ SNMP 的 Trap 有类似于陷阱的意思。

通常，根据查询请求和应答可以定期检查设备的运行动作，根据设置请求可以修改设备的参数。SNMP 的处理可以分为从设备读取数据和向设备写入数据两种。它们采用 Fetch 和 Store 模式。这些操作类似于计算中的输入输出等基本操作▼。

▼计算机中可以向内存中特定的地址写入信息，也可以读取内存中特定地址中的内容，据此进行键盘输入、屏幕显示、磁盘存取等操作。这些过程叫做内存映射 I/O，是 Fetch/Store 模式的典型代表。SNMP 正是将这些操作应用到了网络上。

如果出于某种原因网络设备的状况发生变化，将这个变化通知给 SNMP 管理器时就可以使用 Trap。有了 Trap，即使没有管理器到代理的请求，也能在设备发生变化时收到从代理发来的通知。

8.6.2 MIB

SNMP 中交互的信息是 MIB（Management Information Base）。MIB 是在树形结构的数据库中为每个项目附加编号的一种信息结构。

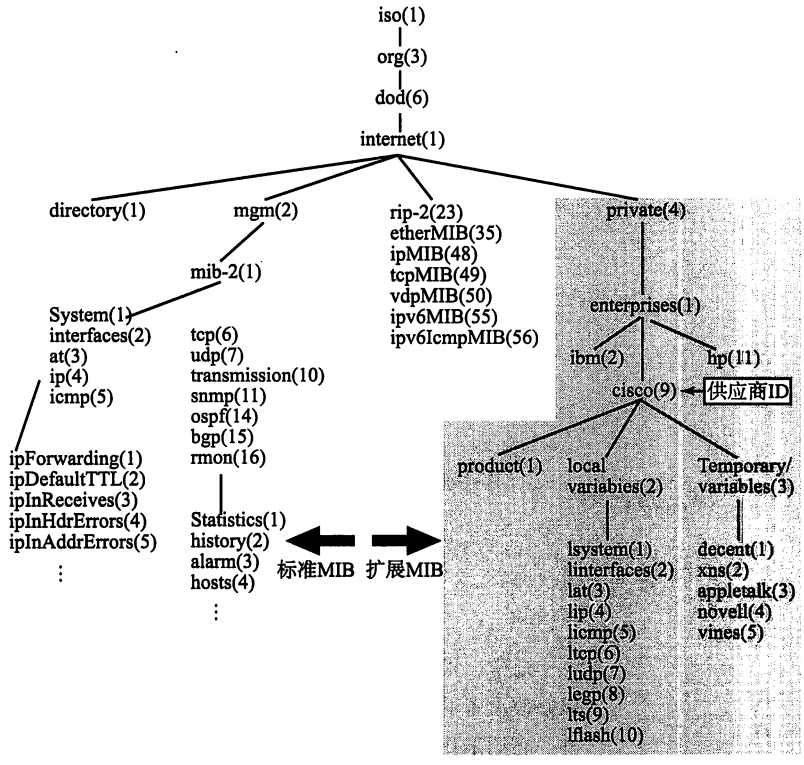
SNMP 访问 MIB 信息时使用数字序列。这些数字序列各自都有其易于理解的名字。MIB 分为标准 MIB▼（MIB、MIB-II、FDDI-MIB 等）和各个提供商提供的扩展 MIB。不论是哪种类型的 MIB 都通过 SMI（Structure of Management Information）定义，其中 SMI 使用 ISO 提出的 ASN.1▼方法。

▼有时也叫私有 MIB。

▼ ASN.1（Abstract Syntax Notation 1）是指抽象语法标记法。为标记 OSI 参考模型中表示层协议而被开发的一种语言。用 ASN.1 标记的数据可以在网络上传输。

MIB 相当于 SNMP 的表示层，它是一种能够在网络上传输的结构。SNMP 中可以将 MIB 值写入代理，也可以从代理中读取 MIB 值。通过这些操作可以收集冲突的次数和流量统计等信息，可以修改接口的 IP 地址，还可以进行路由器的启停、设备的启动和关闭等处理。

图 8.22
MIB 树举例（Cisco Systems 相关）



8.6.3　RMON

RMON 是 Remote Monitoring MIB 的缩写。MIB 由监控网络中某个设备接口（某个点）的众多参数构成。相比之下，RMON 则由监控网络上线路的众多参数构成。

RMON 中可监控的信息从原来的一个点扩展到了一条线上。这样可以更高效率地监控网络。可监控的内容上也增加了很多从用户角度看极为有意义的信息，如网络流量统计等。

通过 RMON 可以监控某个特定的主机在哪里通过什么样的协议正在与谁进行通信的统计信息，从而可以更加详细地了解网络上成为负荷的主体并进行后续分析。

RMON 中从当前使用状况到通信方向性为止，可以以终端为单位也可以以协议为单位进行监控。此外，它不仅可以用于网络监控，以后还可以用于收集网络扩展和变更时期更为有意义的数据。尤其是通过 WAN 线路或服务器段部分的网络流量信息，可以统计网络利用率，还可以定位负载较大的主机及其协议相关信息。因此，RMON 是判断当前网络是否被充分利用的重要资料。

8.6.4　SNMP 应用举例

下面举一个使用 SNMP 的例子。

MRTG（Multi Router Traffic GRAPHER）是利用 RMON 定期收集网络中路由

器的网络流量信息的工具。该用具可以从以下网站获取：

http：// oss. oetiker. ch/ mrtg/

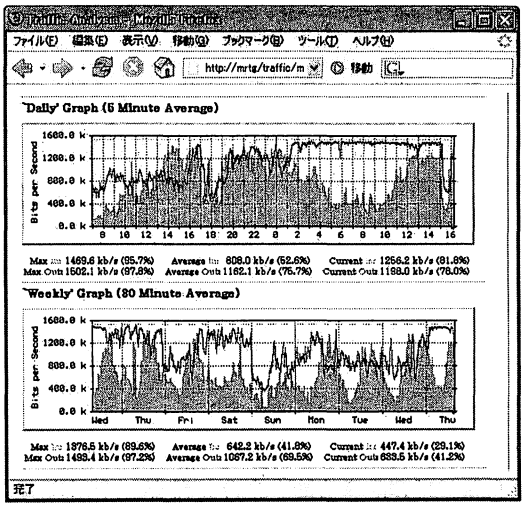


图 8.23 MRTG 可以图像化显示网络流量

8.7 其他应用层协议

互联网一直以来作为数据通信网络得到了蓬勃的发展。最近它的利用范围有了更进一步的扩大。不仅用于实时收发音频、图像、视频等多媒体数据领域，还被用于电视电话会议、现场转播等即时性、双向性的领域。

8.7.1 多媒体通信实现技术

由于TCP具有流控制、拥塞控制、重发机制等功能，有时应用所发出去的数据可能无法迅速到达对端目标主机。然而在互联网电话（使用的VoIP▼）和电视会议当中，即使有少许丢包，也希望系统延时少一点，非常注重系统的即时性。因此，在实时多媒体通信当中采用UDP。

▼ Voice Over IP 的缩写。

然而，只使用UDP还不足以达到进行实时多媒体通信的目的。例如，在互联网电视电话议会中需要提供查询对方号码、模拟电话机的拨号以及以什么形式交互数据等功能。为此，需要一个叫做“呼叫控制”的支持。呼叫控制主要采用H.323与SIP协议。此外，还需要RTP协议（结合多媒体数据本身的特性进行传输的一种协议）和压缩技术（在网络上传输音频、视频等大型多媒体数据时进行压缩）的支持。

结合上述众多技术才能够真正实现实时多媒体通信。此外，互联网电视电话会议对实时性的要求远远高于到目前为止的任何一个数据通信领域。因此在搭建网络环境时有必要考虑QoS、线路容量和线路质量等方面的要求。

H.323

H.323是由ITU开发用于在IP网上传输音频、视频的一种协议。起初，它主要是作为接入ISDN网和IP网之上的电话网为目的的一种规范而被提出的。

H.323定义了4个主要组件。它们分别是终端（用户终端）、网关（吸收用户数据压缩顺序的不一致性）、网闸（电话本管理、呼叫管理）以及多点控制单元（允许多个终端同时使用）。

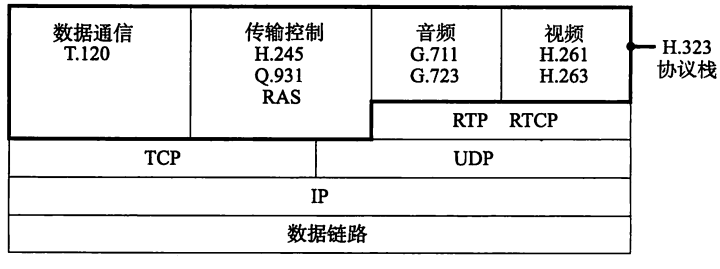


图8.24 H.323的基本构成

SIP

与H.323相对的TCP/IP协议即是SIP（Session Initiation Protocol）协议。SIP的提出要晚于H.323，但是被普遍认为更适用于互联网。H.323的规范内容较多、对应起来比较复杂，而相比之下SIP的构成则简单了许多。

终端之间进行多媒体通信时，需要具备事先解析对方地址、呼出对方号码并对所要传输的媒体信息进行处理等功能。此外，还需要具备中断会话和数据转发

的功能。这些功能（呼叫控制与信令）都被统一于 SIP 协议中。它相当于 OSI 参考模型中的会话层。

通过终端之间收发消息，可以令 SIP 进行呼叫控制并做一些多媒体通信中必要的准备。不过仅凭 SIP 对数据收发的准备工作还不足以进行多媒体数据的传输。SIP 消息通常都由终端进行直接处理，但是也有在服务器上进行处理的情况。由于 SIP 非常相似于 HTTP 的工作机制▼，不仅在 VoIP，在其他应用当中也已经被广泛使用。

▼ HTTP 中进行 Web 页的获取与发送依赖于 ASCII 码字符串的请求命令和数字序列的应答报文。SIP 在这一点是与 HTTP 一样采用 ASCII 码字符串。

图 8.25
SIP 基本组成

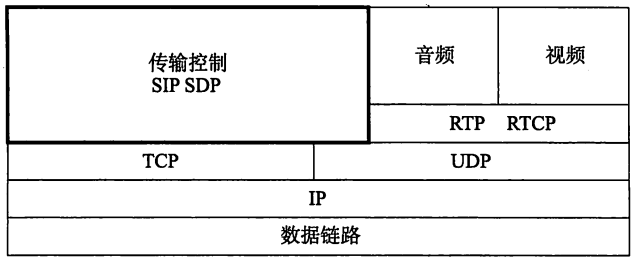
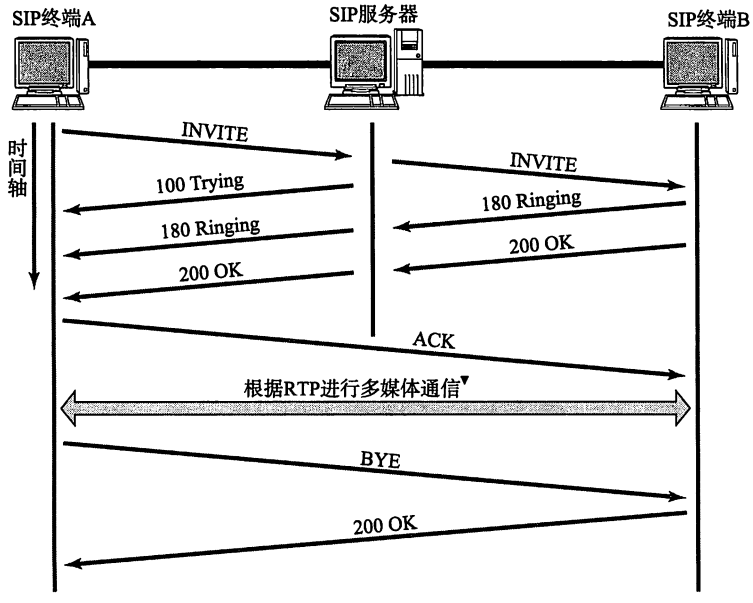


图 8.26
通过 SIP 服务器的呼叫控制的顺序



▼根据 RTP 通信可以不必经过 SIP 服务器，可直接在 SIP 终端之间进行。

表 8.9
主要 SIP 命令

报　文	内　容
INVITE	开始会话
ACK	针对 INVITE 的确认应答
BYE	结束会话
CANCEL	取消会话
REGISTER	注册用户 URI

表 8.10
主要 SIP 响应消息

报　文	内　容
100 系列	临时应答
100	Trying 正在处理中
180	Ringing 振铃
200 系列	会话成功
200	OK 会话成功
300 系列	重定向
400 系列	客户端错误
500 系列	服务器错误
600 系列	其他错误

RTP

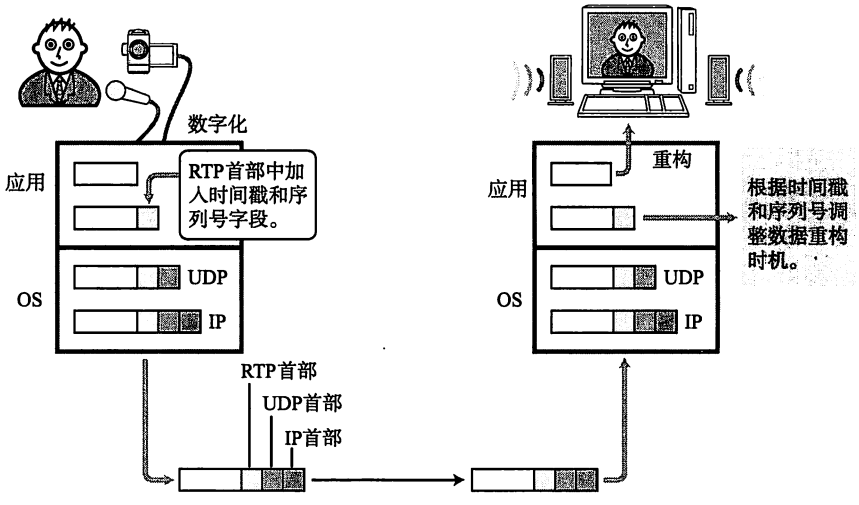
UDP 不是一种可靠性传输协议。因此有可能发生丢包或乱序等现象。因此采用 UDP 实现实时的多媒体通信需要附加一个表示报文顺序的序列号字段，还需要对报文发送时间进行管理。这些正是 RTP（Real-Time Protocol）的主要职责。

RTP 为每个报文附加时间戳和序列号。接收报文的应用，根据时间戳决定数据重构的时机。序列号则根据每发出一次报文加一的原则进行累加。RTP 使用这个序列号对同一时间戳的数据▼进行排序，掌握是否有丢包的情况发生。

▼尤其是对于视频的数据。视频中一个帧的数据往往要超过一个包，然而它们发送的时间戳一致。此时就可以使用同一时间戳内不同的序列号加以区分。

RTCP（RTP Control Protocol）是辅助 RTP 的一种协议。通过丢包率等线路质量的管理，对 RTP 的数据传送率进行控制。

图 8.27
RTP 通信



数字压缩技术

通过有效的压缩可以大量减少音频和视频数据的大小。在有限的网络资源中进行多媒体数据的传输，压缩技术成为一个必要的手段。

MPEG（Moving Picture Experts Group）是决定数字压缩规范的 ISO/IEC 工作组。在这里所制定的规范叫做 MPEG。在 MPEG 的众多规范当中，MPEG1 主要用于 VideoCD，而 MPEG2 主要用于 DVD 和数字电视播放领域。此外，还有 MPEG4

和 MPEG7 等规范。连音乐压缩的 MP3▼ 也属于 MPEG 的规范。

▼正式的名称为 MPEG1 Audio Layer III。

另一方面，由 ITU-T 的 H.323 所规定 H.261、H.263 与 MPEG 共同协作的产生了 H.264。除此之外，还有微软公司自己的规范。

这些都属于数字压缩技术的范畴。由于它们着重于数据格式上的处理，可以认为它们相当于 OSI 的表示层。

8.7.2 P2P

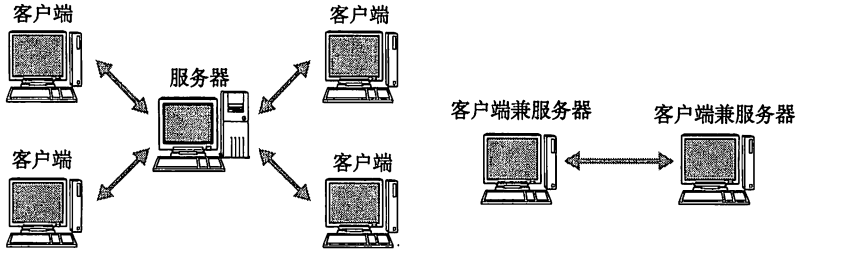
互联网上电子邮件的通信，普遍属于一台服务器对应多个客户端的 C/S 模式，即 1 对 N 的通信形态。

与之不同，网络上的终端或主机不经服务器直接 1 对 1 相互通信的情况叫做 P2P（Peer To Peer）。这就好比使用无线收发器进行一对一通话。P2P 中主机具备客户端和服务端两方面的功能，以对等的关系相互提供服务。

IP 电话中也有使用 P2P 的例子。使用 P2P 以后，可以分散音频数据给网络带来的负荷，实现更高效的应用。例如互联网电话 Skype 就采用了 P2P 的功能。

除了 IP 电话外，其他实现互联网的文件传输应用如 BitTorrent 协议或一部分群组软件等，也是用到了 P2P 的技术。

图 8.28
集中型与 P2P 型



1台服务器连接N台客户端的集中型　　每个主机兼客户端与服务器进行1对1相连的P2P型

不过，也有不支持 P2P 的环境。例如在服务器与客户端分离型的环境中，服务器要在一个可以由互联网直接访问的地方，而客户端即使是在 NAT 内侧也不会有问题。然而在 P2P 中这个结构却行不通。它必须具备从互联网越过 NAT 令双方终端能够访问的功能。

8.7.3 LDAP

LDAP（Lightweight Directory Access Protocol）是访问目录服务的一种协议，也叫轻量级目录访问协议。所谓“目录服务”是指网络上存在的一种提供相关资源的数据库的服务。这里的目录也有地址簿的意思。可以认为目录服务就是管理网络上资源的一种服务。

▼ ISO 于 1988 年制定的标准目录访问协议（DAP，Directory Access Protocol）。X.500 是它在 ITU-T 中的编号。

LDAP 用于访问这种目录服务。目录服务的规范作为 X.500▼ 于 1988 年由 ISO（国际标准化组织）制定。而 LDAP 在 TCP/IP 上实现了 X.500 中的一部分功能。

就像 DNS 为了更简单地对网络上的各个主机进行管理一样，LDAP 是为了更简单地管理网络上的各种资源。

▼ LDIF（LDAP Interchange Format：LDAP 数据交换格式）

LDAP 定义了目录树的结构、数据格式、命名规则、目录访问顺序和安全认证。图 8.29 列出了 LDAP 设置的一般结构▼。图 8.30 则为单纯目录树的例子。

图 8.29
LDIF 文件

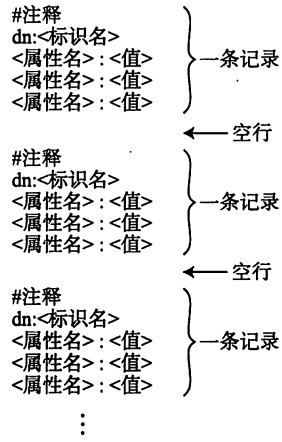
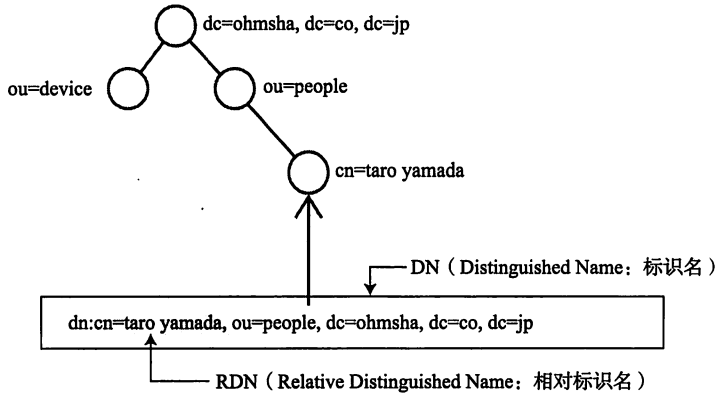


图 8.30
LDAP 目录树（DIT）



在大规模的公司或教育机关中，所要管理的对象如使用者（用户）和设备的数量往往非常庞大。那么为了让这些用户能够使用计算机或某个应用，有必要事先进行可否使用计算机或应用的设置。此时如果这些设备和应用应对了 LDAP，并在一个可以进行统一管理的 LDAP 服务器中注册了所有用户，那么就可以对这些用户是否有效进行判断。LDAP 常被用于这一类的认证管理和资源管理中▼。

▼同一类型同样功能的产品还有微软公司的 Active Directory、Novell 公司的 eDirectory 等。它们都在支持 LDAP 的同时还提供自身扩展的功能，所以每个产品所能提供的服务也都不相同。因此，很多公司会根据自己的需求选择合适的产品。