第3章 应用层

目前，计算技术的发展使得现代社会进入了以 Internet 为运行环境，以网络计算和普适计算为特征，正积极向移动计算和云计算推进的信息化时代。Internet 上提供了丰富的计算、存储资源和信息资源。那么，如何去使用和访问这些资源？Internet 为人们使用和访问网络资源提供了哪些服务和手段？计算机网络如何接受和响应用户的请求并为用户提供服务？这些都是应用层所要完成的功能。应用层为用户提供的具体服务包括远程登录、电子邮件、文件传输、WWW 服务等。每种服务均有对应的应用层协议来支持，这些协议规定了网络应用所应遵守的规范和准则。下面就详细介绍 Internet 的应用层所提供的各种协议及其功能，具体包括远程终端登录、电子邮件、文件传输、超文本传输等协议，以及为这些协议软件运行提供支持的域名解析、动态主机配置等协议。

3.1 概述

应用层是网络协议层次模型的最高层，也是用户与网络的接口。通过使用 Internet，知道计算机网络提供多种类型的应用服务，如 WWW 服务、电子邮件、文件传输等。对于每种应用服务均涉及网络上不同节点间的通信，以及人机间的交互过程。显然，既然涉及通信和交互过程，就需要有相关的标准、规则和约定（即协议）来支撑。在 Internet 环境下，各种应用功能存在一定的差异。所以，分别为每种服务定义了相应的应用层协议，如完成 WWW 服务的 HTTP、完成电子邮件功能的 SMTP 和完成文件传输功能的 FTP 等等。每种服务均由相应的协议软件来实现，这些协议软件分别运行在用户端和服务器端。运行在用户端的软件称为用户软件；运行在服务器端的软件称为服务器软件。不同的服务器软件可以安装、运行在同一台计算机上，也可以分别安装、运行在不同的计算机上。装有服务器软件的系统启动后，便创建了一个称为守护进程（Daemon）的服务器进程，该进程一直处于运行状态，等待用户的服务请求。用户使用服务时，需通过用户软件（如 WWW 浏览器等）发出请求，本地网络操作系统接受该请求，并创建一个用户进程（也称为客户进程），该进程调用网络协议软件，通过网络向对应的服务器进程发出请求；服务器进程接受请求，并按照协议来分析和响应请求，将处理结果按照通信协议通过网络返回给用户进程。协议定义了用户进程与服务器进程间交换信息的格式和顺序，以及发送、接收和响应请求时所采取的操作。客户/服务器模式的通信过程如图 3.1 所示。显然，用户进程是一个请求进程，而服务器进程是一个响应进程。这种服务模式称为客户/服务器模式，网络上大部分服务均工作于这种模式。

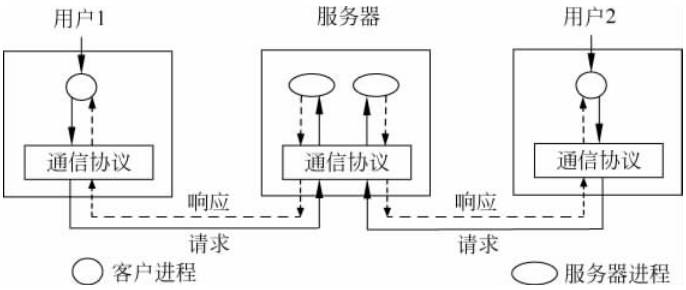

图 3.1 客户/服务器模式的通信过程

目前,计算机网络所使用的客户/服务器模式应该严格地称为基于 Web 的客户/服务器模式,简称为 B/S(Browser/Server)模式。而传统的客户/服务器模式简称为 C/S(Client/Server)模式。基于 Web 的客户/服务器模式与传统的客户/服务器模式间存在一定的区别。传统的客户/服务器模式下,客户端程序承担较多的功能,是基于不同的操作系统和数据库管理系统开发的。当客户端的操作系统等运行环境发生变化时,可能需要重新开发应用软件,这样,势必造成很大的浪费和不便,并且对于不同的操作系统环境,应用软件的移植也存在一定困难。能否把应用系统都放在服务器端,而简化用户端的功能,让应用软件在服务器端运行;在这种情况下客户使用时,只需要访问服务器即可。这样,无论用户端的系统软件环境如何变化,只要能访问服务器,就可以得到相应的服务,这就是所谓的 B/S 模式。显然,这种工作方式下,服务器端的负担较重,而客户机所承担的工作较少,所以也称之为瘦客户机。目前的云计算正是贯彻这种理念,将服务集中在云端(即服务器端),而客户端可以做得非常简单(如智能手机等各种移动终端),只要它能通过网络访问云端,就可以得到云端提供的各种服务。

现在计算机所使用的都是多任务、多用户操作系统,无论是客户机,还是服务器,均可能同时运行多个进程(即多个应用程序)。对于计算机网络而言,它可能同时提供多种类型的服务,而每种服务都将创建相应的应用层进程,这些应用层进程可能要用到相同的运输层协议。为了区分这些并发的应用层进程,在应用层的下一协议层:运输层引入了端口号的概念。每个端口号是一个 16 位的二进制整数,它唯一地标识了某台主机内运行的一个应用层进程。应用层进程在调用运输层协议时,由运输层进程为其分配一个端口号,并保证在一台计算机内运行的应用层进程与端口号是一一对应的。

对于服务器来讲,它可能同时提供多种类型的网络服务,即运行多个服务器进程。当客户端想要得到某种类型的网络服务时,它必须事先知道向哪台服务器上的哪个服务器进程发送请求,即必须事先知道提供该服务的服务器地址和服务器进程的端口号。所以,与各种网络服务相对应的服务器进程的端口号必须是公布于众的,这些端口称为熟知端口。Internet 规定熟知端口号从 0 到 1023,表 3.1 给出了常用 Internet 服务所对应的熟知端口号以及与运输层协议间的关系。而客户端进程的端口号是由它调用的运输层协议进程随机产生的,称之为临时端口,其值要大于 1023,一般在 49 152～65 535 之间。

当用户想要得到某种网络服务时,它首先调用相应的应用软件(如浏览器),创建一个客户进程,客户进程将调用通信协议把所获得的端口号和客户机的 IP 地址作为源端口号和源 IP 地址,而将提供服务的服务器进程的熟知端口号和服务器的 IP 地址作为目的端口号和

目的 IP 地址，连同请求信息一起封装成请求数据包，发送给服务器。网络层的 IP 根据请求数据包中的目的 IP 地址，控制该数据包传输到相应的服务器；服务器接收到该数据包后，由运输层协议根据其中的目的端口号将该请求送给相应的服务器进程（即应用层进程，对应某种网络服务）。服务器进程处理客户端的请求，然后，将请求数据包中的源端口号和源 IP 地址作为目的端口号和目的 IP 地址与响应信息一起封装成响应数据包，通过网络协议发送给请求服务的客户进程。

各种网络服务与端口间的关系如图 3.2 所示。显然，客户机的 IP 地址和客户进程的端口号在全网范围内唯一地标识了客户进程，而服务器的 IP 地址和服务器进程的端口号也在全网范围内唯一地标识了服务器进程。因此把主机的 IP 地址以及主机上标识进程的端口号组合在一起，称之为套接字（socket）。这样，一对套接字唯一地标识了网络上的一条通信连接，这一点在运输层介绍 TCP 时将详细讲解。

表 3.1 常用 Internet 服务所对应的熟知端口号以及与运输层协议间的关系

服务类型	文件传输		远程终端	SMTP	POP3	WWW	域名服务	简单文件传输	简单网络管理
	数据连接	控制连接							
端口号	20	21	23	25	110	80	53	69	161
运输层协议	TCP						UDP		

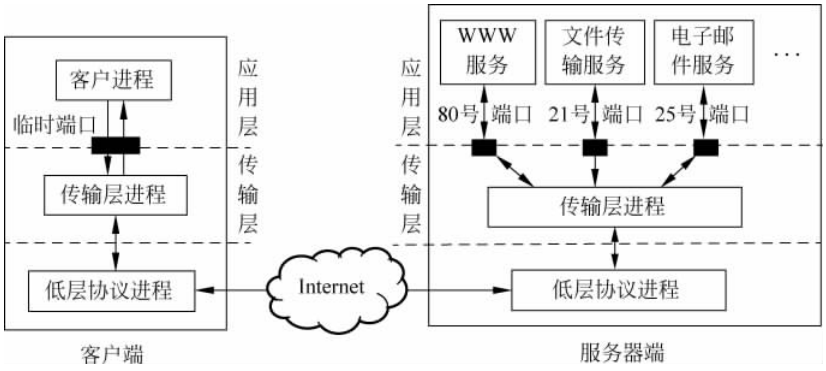

图 3.2 各种网络服务与端口间的关系

3.2 域名和域名解析

3.2.1 域名及域名的组成结构

在日常生活中，当人们想要寻找某个地点时，应事先知道该地点的地址。同样，当要访问某种网络资源时，必须知道该资源的网络地址。在 Internet 上，用 IP 地址就可以唯一地标识出网络节点的地址。第 1 章简单介绍了 IP 地址的概念，在网络层还将详细介绍。实际上，IP 地址是一个 32 位的二进制数字串，它尽管可以采用四位点分十进制来表示（如 202.156.122.210），但这样一串枯燥的数字仍不利于记忆，而人们习惯于记忆名字。所以，为了

便于记忆，常采用形象、直观的字符串作为网络上各个节点的地址，如搜狐的 WWW 服务器的网络地址为：www. sohu. com。这种唯一地标识网络节点地址的符号名就称之为域名(domain name)，也称为主机名(host name)。域名是一个逻辑概念，与主机所在的地理位置没有必然联系，它们由专门的组织(不同级别的网络信息中心)进行管理和分配，用户使用域名需要向该组织申请和注册。

由于 Internet 上用户数量的快速膨胀，域名急剧增多。为了便于管理、记忆和查找，常采用树形层次结构组织域名。其中树根节点为空，以下的每级节点依次分别称为：顶级域、二级域、三级域……所表示的名称分别称为：顶级域名、二级域名、三级域名……如图 3.3 所示。整个域名树组成了 Internet 的域名空间，而以每个非根节点为根的子树组成了 Internet 的一个域名子空间。域名树上的每个节点(除根节点)都定义了一个标签(label)，它是该节点所对应级别域名的一个实例，它由字母、数字和连字符“-”组成，其最大长度为 63 个字符，而且不区分大小写。网络上每个节点的域名是从该节点开始，按照域名树的层次结构自底向上，最终结束到根节点，而形成的一个标签序列。书写时从左到右排列，中间用圆点分开，最右边的域名为顶级域名，具体形式为：

……三级域名. 二级域名. 顶级域名

如：北京大学图书馆的域名表示为 lib. pku. edu. cn。

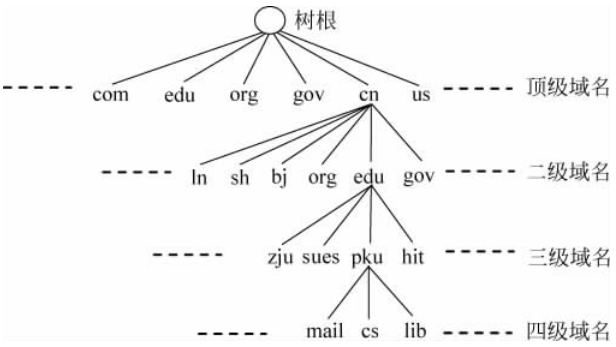

图 3.3 Internet 域名空间

注意，一个域名的最大长度为 255 个字符。各级域名由上一级域名机构进行管理，顶级域名由因特网名字与号码指派公司(The Internet Corporation for Assigned Names and Numbers，ICANN)进行管理。顶级域名分成 3 大类：

- 国家顶级域名。采用了 ISO 3166 的规定，例如：. cn 表示中国，. us 表示美国，. jp 表示日本等。
- 国际顶级域名。采用. int，国际性组织可在. int 下面注册域名。
- 通用顶级域名。这些域名表示公司、政府部门、军事部门、教育机构等，共 13 个，见表 3.2。

顶级域名下面是二级域名。中国将二级域名分成“类别域名”和“行政区域名”两大类。其中，类别域名 6 个，见表 3.3。行政区域名 34 个，用于表示全国的省、自治区、直辖市和特区，如. bj 表示北京，. sh 表示上海，等等。若在中国二级类别域名. edu 下申请注册三级域名，则需要向中国教育科研计算机网络中心申请；若在其他二级域名下申请注册三级域名，则需要向中国互联网网络信息中心 CNNIC 申请。图 3.3 展示了 Internet 域名空间的大致

情况，从表中可以看到，一旦某个单位拥有了三级以下级别的域名，下一级域名完全由它自己决定如何去命名和管理。如北京大学拥有三级域名：pk.edu.cn，其下属单位的域名就由它自己分配和管理。

表 3.2 常用的通用顶级域名

序号	域名	代表含义	序号	域名	代表含义	序号	域名	代表含义
1	.com	公司企业	6	.edu	教育机构	11	.museum	博物馆
2	.net	网络服务机构	7	.aero	航空部门	12	.name	个人
3	.org	非营利性组织	8	.biz	商业	13	.pro	自由职业者
4	.gov	政府部门	9	.coop	合作团体			
5	.mil	军事部门	10	.info	网络信息服务组织			

表 3.3 中国二级域名中的类别域名

序号	域名	代表含义	序号	域名	代表含义
1	.com	工、商、金融等企业	4	.edu	教育机构
2	.net	网络信息中心和运行中心	5	.ac	科研机构
3	.org	非营利性组织	6	.gov	政府部门

3.2.2 域名解析

上面已经说明，用域名标识某个网络节点的地址只是为了方便记忆，但随数据包一起传输的还是 32 位二进制数组成的 IP 地址，而上网所使用的却是域名，所以就需要一种方法自动地将域名转换为 IP 地址。这种将域名转换为 IP 地址的过程称为域名解析。

域名解析的过程与打电话的过程颇为相似。显然，平时人们大脑中记忆的是人的名字。当要给某个人打电话时，首先，想到的是这个人的姓名，然后，根据姓名查找电话号码簿，找到其电话号码，再使用电话号码进行拨号和通话。此时通过查找电话号码簿就将受话人的姓名转换为其电话号码。

在 Internet 发展初期，由于当时的网络规模比较小，采用 Hosts.txt 文件来记录和管理网络上各个节点的域名和对应的 IP 地址，进而实现域名解析功能。该 Hosts.txt 文件存储在网络系统的中心管理服务器上，每个网络节点在启动时，均需要从中心服务器上下载该文件。但是，随着 Internet 主机数量的急剧增加，除了 Hosts.txt 文件的大小不断增加外，每次 Hosts.txt 文件的下载及其更新过程产生的流量也在不断增加，进而给网络通信造成了很大负担。所以，迫切需要一种采用分布式管理方式、可扩展性好、支持多种数据格式的软件系统来代替 Hosts.txt 文件实现域名解析功能，这就是下面将要介绍的域名服务器软件系统。

这种称为域名服务器的软件系统诞生于 1984 年，它代替了基于 Hosts.txt 文件的域名解析方式来完成域名解析功能。该软件也称为 DNS 服务器，它执行的协议是一个应用层协议，称之为域名解析协议。DNS 服务器上设有 DNS 数据库，用于记录和存储网络节点的域名、IP 地址等信息。当用户使用域名访问某个网络节点时，它创建的客户程序首先发送请求给域名服务器，要求域名服务器将要访问的域名转换成对应的 IP 地址，然后，使用 IP 地

址来与目标服务器进行交互。

Internet 上，通常设有多个域名服务器，分别负责不同域名子空间的域名解析工作。这些服务器按着域名的层次结构进行组织，分布在级别不同的各个域中，组成了一个高效、可靠、协同工作的分布式系统。域名服务器具体分为以下 3 类：

- 本地域名服务器。每一个独立的网络系统（称之为自治系统（Autonomous System，AS））均设有自己的域名服务器，它负责本区域内所有网络节点域名的管理和解析，该域名服务器称为本地域名服务器，也称为默认域名服务器。该区域的每个网络节点必须将其域名和对应的 IP 地址等信息登记在一个本地域名服务器的 DNS 数据库中。
- 授权域名服务器。Internet 上，登记有某台主机域名信息的本地域名服务器也称为这台主机的授权域名服务器。为了可靠起见，要求每台主机至少有两台授权域名服务器。
- 根域名服务器。Internet 上，共有 13 个根域名服务器，它们是负责顶级域名管理的授权域名服务器，其中有 10 个位于北美洲，其他 3 个位于欧洲和亚洲。根域名服务器的作用非常重要，它是架构因特网所必需的基础设施。若出现故障将严重影响网络的正常运行，所以，每个根域名服务器一般是由多个服务器组成的分布式系统，它们协同、可靠地工作，共同完成域名的解析任务。

域名解析系统工作于客户/服务器模式，它使用运输层中的 UDP 进行传输，对应的端口号为 53。为了提高域名解析的速度，每个网络节点常在本地存储器中开辟一块存储区域（称为 DNS 缓存），用于存储已经获得的域名和对应的 IP 地址等信息。当 WWW、E-mail、FPT 等服务采用域名访问网络时，应用进程首先创建一个称为域名解析器（resolver）的本地客户端进程。该进程接收应用进程的域名解析请求，并在本地 DNS 缓存中进行查找。若找到对应域名的 IP 地址，则直接将查询结果返回给应用进程；若域名解析器在本地缓存中没有找到匹配结果，则它将要进行解析的域名等信息封装成 DNS 请求报文，调用运输层的 UDP，向本地域名服务器的 53 号端口发送请求，要求本地域名服务器协助查找；若本地域名服务器仍没有查找到结果，则它作为新的客户端，再向根域名服务器发送 DNS 请求，根域名服务器没有找到，则再向下一级域名服务器发送 DNS 请求，直到在某级域名服务器上查询到待解析的结果。然后，形成 DNS 响应报文将查询结果返回给请求 DNS 服务的主机。该主机首先将域名和得到的 IP 地址存入自己的 DNS 缓存，然后，采用该 IP 地址封装数据包，并通过网络发给相应的目的节点。

因此，为了实现上述的域名解析功能，必须做如下假设：

- 每个域名服务器必须知道所有根域名服务器的 IP 地址。
- 每个域名服务器必须知道其下一级域名服务器的 IP 地址。

显然，只有满足上述条件，各个域名服务器才能知道其他相关域名服务器的地址信息，这样它们才能相互协同，以共同完成域名解析工作。实践证明：DNS 是在 Internet 上实现的最成功的分布式系统。

通常，域名解析有两种不同的实现方式：递归解析（recursive resolution）和反复解析（iterative resolution）。

1. 递归解析

当主机需要进行域名解析，但在本地查找 DNS 缓存而没有成功时，它调用解析器向本地域名服务器发送一个 DNS 请求报文，其中含有要解析的域名信息。若本地域名服务器找到了指定的域名，则形成 DNS 响应报文将对应的 IP 地址信息返回给主机；若本地域名服务器没有找到指定的域名，则本地域名服务器将请求授权的根域名服务器协助查找；然后，根域名服务器根据要查找域名的二级域名请求相应的二级域名服务器协助查找；重复上述过程，直到某一级的域名服务器找到了相应的域名信息，然后，它形成 DNS 响应报文，按照刚才的请求路径逆向传递，如图 3.4 中的虚线所示，最终传递给请求域名解析的主机。上述域名解析过程如图 3.4 所示，其中的序号标明了 DNS 请求和响应的顺序。用户访问新浪网 WWW 服务器域名的递归解析过程如图 3.5 所示。此时在本地域名服务器和本地 DNS 缓存中，均没有找到新浪 WWW 服务器的 IP 地址，只好调用域名解析系统进行解析。

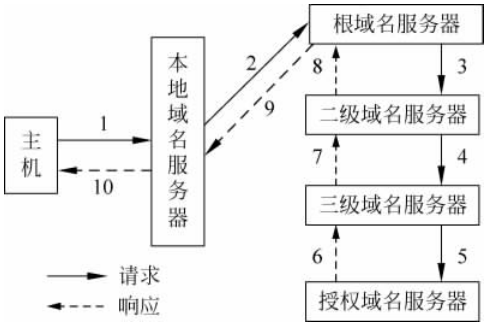

图 3.4 域名的递归解析过程

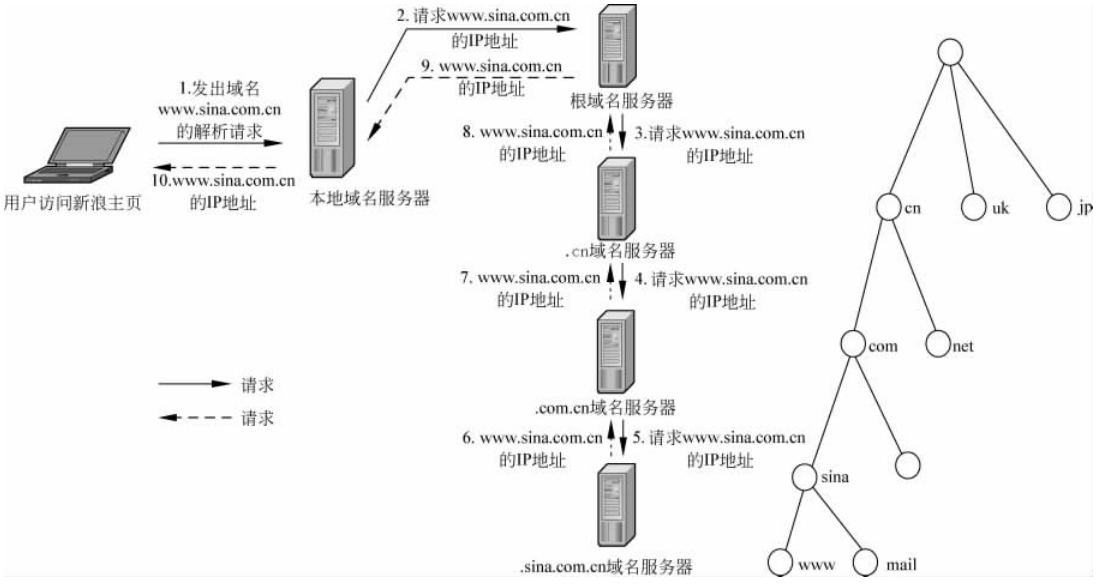

图 3.5 用户访问新浪网 www 服务器域名的递归解析过程

2. 反复解析

反复解析也称为迭代解析。当主机需要进行域名解析，但在本地没有成功时，它也是首先调用解析器向本地域名服务器发送一个 DNS 请求报文。若本地域名服务器找到了指定的域名，则形成 DNS 响应报文将对应的 IP 地址信息直接返回给主机；若本地域名服务器没有找到指定的域名，则它要向授权的根域名服务器发送 DNS 请求报文要求协助查找。根

域名服务器若找到相应的域名，则形成 DNS 响应报文返回结果给本地域名服务器；否则根域名服务器将根据待查找域名的二级域名确定下一个域名服务器，并把该域名服务器的地址通知给本地域名服务器；然后，本地域名服务器再向该二级域名服务器发出 DNS 请求，要求其协助查找；若该域名服务器找到了相应的结果，则形成 DNS 响应报文，将查到的 IP 地址信息通知给本地域名服务器；若没有找到，则再根据待查找域名的三级域名确定下一级域名服务器，并把它的地址通知给本地域名服务器。重复上述过程，直到某一级域名服务器找到了结果，并将结果通知给本地域名服务器，本地域名服务器将查找结果形成 DNS 响应报文通知给主机，从而完成了一次域名解析工作。域名反复解析的过程如图 3.6 所示，其中的序号标明了 DNS 请求和响应的顺序。用户访问新浪网 WWW 服务器的域名解析过程如图 3.7 所示，此时在本地域名服务器和本地 DNS 缓存中均没有找到新浪 WWW 服务器的 IP 地址，只好调用域名解析系统进行解析。

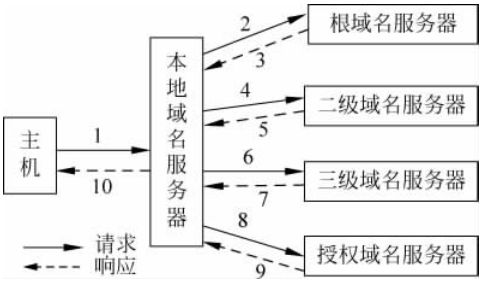

图 3.6 域名反复解析的过程

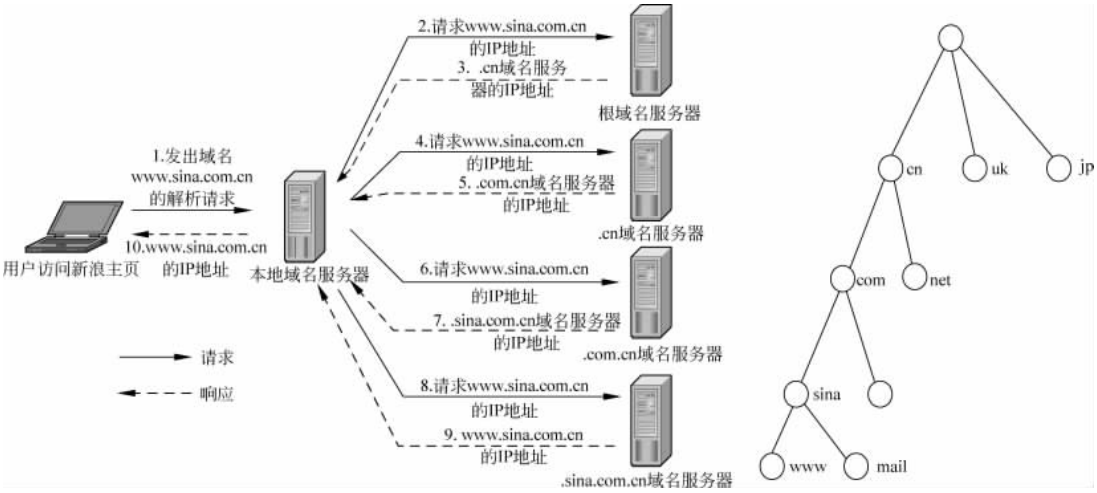

图 3.7 用户访问新浪网 WWW 服务器的域名解析过程

为了提高域名解析的速度，还采取了如下 3 种措施：

(1) 为了减轻根域名服务器的负担，它常工作于反复解析模式。即根域名服务器一旦接收到某个本地域名服务器的 DNS 协助请求，它将根据待解析的域名确定负责解析该域名的下一级域名服务器的 IP 地址，并将之通知给发出请求的本地域名服务器，让它们合作完成该次域名的解析工作，以后不再干预，从而减轻其工作负担。

(2) 在本地域名服务器内设置了一块高速缓存。当本地域名服务器从其他域名服务器得到了不在其管辖范围内的域名和对应的 IP 地址信息后，就把该域名和对应的 IP 地址等信息存入高速缓存。以后在域名解析过程中，一旦确定所解析的域名不在其管辖区域内，下一步就查找高速缓存。若高速缓存内仍查不到，则再向根域名服务器发出请求，要求协助查找。

(3) 许多主机在启动时,从本地域名服务器中下载全部域名解析信息。同时,在内存中设置了 DNS 高速缓存(在前面介绍域名查找算法时已经提到),用于记录新得到的域名解析信息。这样,每次进行域名解析时,首先查找本地 DNS 信息,若找不到才求助于本地域名服务器,这样可以显著加快域名解析的速度。

由于网络是动态变化的,高速缓存中的内容可能因为时间过长而失效。所以,为了保证高速缓存中域名信息的有效性,要设置时钟限制并定时更新。

此外,为了保证域名解析系统工作的可靠性,域名服务器一般均成对存在,以便一个出现故障时,另一个能接替工作。

3.3 远程终端协议

在计算机发展初期,个人计算机的功能相对简单,而更多的计算机软、硬件资源和信息资源都集中在小型机或者更高档次的计算机系统上,这些计算机系统运行分时操作系统,同时可以为多个本地和远程用户提供服务。而功能有限的个人计算机(或者终端)为了完成复杂的功能或者为了获取更多的信息资源,常常作为一个仿真终端或者智能终端登录到远程计算机系统,以使用远程计算机系统上的软、硬件资源。此时,对于用户而言,它的显示器和键盘就好像直接连接到远程计算机上,远程计算机系统就像它的本地主机一样供其使用,这就是远程登录服务。但随着个人计算机功能的逐渐增强,远程登录服务已很少使用;但在网络管理领域,目前仍在使用这种方式来管理和配置路由器、交换机等网络设备。

Internet 使用远程终端协议(Telnet)来提供远程登录服务。这种服务工作在客户/服务器模式下,它由两部分组成:Telnet 客户端和 Telnet 服务器。远程 Telnet 服务器上运行服务器进程,这个进程是一个守护进程,一直运行在 23 号端口,等待用户的 Telnet 请求。当用户想要申请 Telnet 服务时,它在本地计算机上启动一个 Telnet 客户进程,向 Telnet 服务器的 23 号端口发送连接请求。服务器进程接受该请求,建立与客户机的连接,并启动一个子进程响应该 Telnet 请求,然后,返回到等待状态继续等待其他 Telnet 请求。远程登录过程示意图如图 3.8 所示。

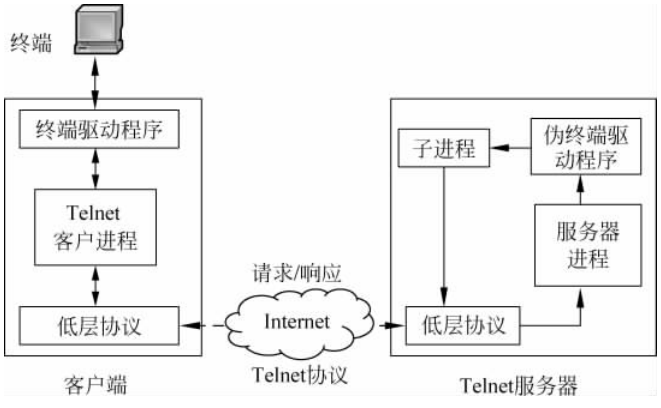

图 3.8 远程登录过程示意图

远程登录服务所涉及的一个关键问题是客户端和远程服务器可能是异构的，它们的计算机硬件、所使用的操作系统以及键盘输入字符的格式等方面可能并不相同。如有些系统使用 ASCII 码的回车(CR)来表示文本行的结束，有些系统使用换行符(LF)来表示，而有些系统使用回车加换行符(CR+LF)的组合来表示。为了使 Telnet 服务器能够识别远程用户所输入的命令，必须消除这些差异。Internet 采用网络虚拟终端(Network Virtual Terminal，NVT)来适应不同系统间的差异。即用户输入的命令首先由本地格式转换为 NVT 格式，然后，发送给 Telnet 服务器；Telnet 服务器接收这种 NVT 格式的命令，再转换为服务器端的命令格式，然后执行，并将响应结果转换为 NVT 格式后返回给用户；用户接收后再将之转换为本地格式并显示在显示器上。远程登录的 NVT 转换示意图如图 3.9 所示。

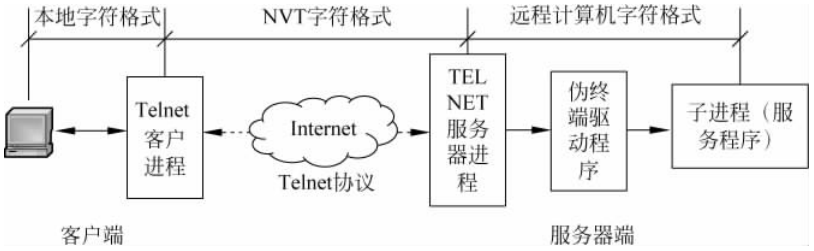

图 3.9　远程登录的 NVT 转换示意图

NVT 使用 7 位标准的 ASCII 码来表示数据，每个 7 位字符以字节为单位进行发送，字节的最高位规定为 1。7 位标准的 ASCII 码字符集包括可打印/显示的普通字符和不可显示的控制字符。对于用户输入的普通字符，NVT 将其按照原始含义进行传送；而当输入的是控制字符或者组合字符时，NVT 将它们转换为特殊的编码，然后在网络上传输。

当本地用户想要获得远程登录服务时，首先，在本地计算机上启动一个 Telnet 客户进程，向 TELNET 服务器申请进行登录和注册，建立与 Telnet 服务器间的双向连接(该连接通过运输层的 TCP 来建立)，然后就通过该条连接进行交互，传输命令、响应等信息。

Telnet 协议支持多种命令，这些命令用于控制客户端与服务器间的交互。每条 Telnet 命令一般由两个以上字节组成，第一个字节各位全为 1(即 0xFF)，称为 IAC(Interpret As Command)。它是一个转义字符，用于表示后续的字节为命令代码。对于某些命令代码，其后跟有选项代码，用于客户端与服务器间协商数据的传输方式、传输速度等，以增强 Telnet 协议的灵活性和对异构环境的适应能力。常用的 Telnet 命令见表 3.4。

表 3.4　常用的 Telnet 命令

命　　令	命令编码	命 令 含 义
IAC	255	表示之后的数据为命令
DON'T	254	表示选项参数所指定的请求被拒绝
DO	253	表示选项参数所指定的请求被接受
EL	248	通知服务器删除当前行
EC	247	通知服务器删除前一个字符
IP	244	中断、终止或结束某个进程
BRK	243	表示数据传输的中断
EOF	236	表示文件中的数据已全部传输出去

3.4 文件传输协议

计算机网络最基本的功能就是实现数据通信和资源共享。位于一台计算机上的文件可以通过网络传输到另一台计算机，通过网络也可以读取其他计算机上的文件。在计算机网络中，上述功能是通过文件传输协议来实现的。这种文件传输协议最初是 ARPANET 的一个组成部分，后来发展成 TCP/IP 中的一个应用层协议。这种协议实现了文件的远程传输功能，是计算机网络所提供的基本功能之一，也是计算机网络产生初期使用最多的功能，目前仍在广泛使用。

文件传输服务通过网络将文件从本地计算机复制到另一台称为文件服务器的计算机，该过程称为文件的上传。相反，从文件服务器传输文件到本地计算机的过程称为文件的下载。这种文件的传输过程看起来似乎很简单，但是，实现起来却相当复杂。众所周知，计算机网络常常由多台异构的计算机所组成。这些计算机的操作系统、所使用的字符集、采用的文件结构及格式、目录的组织方式等均可能存在一定差异。所以，文件传输必须考虑到这些差异，并能屏蔽掉这些差异，为用户提供透明的文件传输服务。Internet 采用文件传输协议(File Transfer Protocol，FTP)实现上述功能，所以，文件传输服务也称为 FTP 服务。

3.4.1 FTP 的工作原理

文件传输服务工作于客户/服务器模式，它由客户端和服务器端两部分组成，如图 3.10 所示。用户操作的本地计算机即为客户端，存储用户想要访问文件的计算机系统为服务器端，通常称该计算机系统为 FTP 服务器，用户通过 FTP 访问服务器端的文件。FTP 工作于运输层的 TCP 之上，它采用 21 号端口，以面向连接的方式工作，通过客户端和服务器端间的交互会话过程，实现它们之间的数据传输。

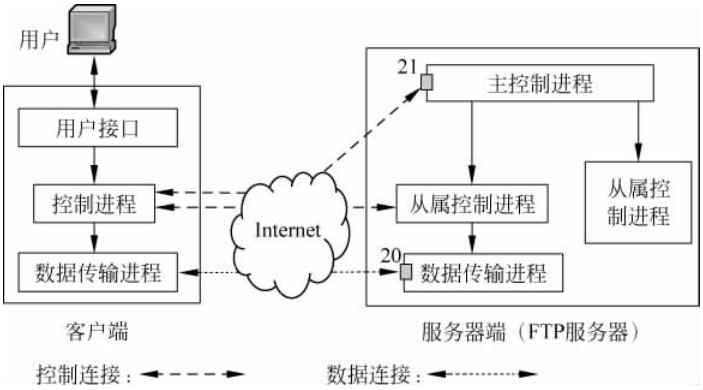

图 3.10 文件传输过程示意图

图 3.10 给出了 FTP 服务的基本模型。客户端由用户接口、控制进程和数据传输进程组成。而服务器端由主控制进程、从属控制进程和数据传输进程组成。其中，主控制进程为守护进程，FTP 服务器启动后，它就一直处于运行状态，该进程一直监测 21 号端口，等待用户的 FTP 请求。当客户端要访问 FTP 服务器时，首先，调用运输层的 TCP 建立客户端与

FTP 服务器端的两条双向连接：控制连接和数据连接。其中，控制连接用于传输命令和响应信息；而数据连接用于传输要存取的数据文件。建立连接时，控制连接先建立，然后，客户端与 FTP 服务器端使用这条控制连接进行交互协商，以建立数据连接，之后使用这条数据连接传输数据文件。

FTP 规定有两种不同的连接模式：PORT 和 PASV，分别称为：主动模式(active mode)和被动模式(passive mode)。对于不同的连接模式，FTP 服务器和客户端仅在建立数据连接时的作用有所不同。

1. PORT 模式

该模式下，数据连接的建立由 FTP 服务器发起。当用户想要访问 FTP 服务器时，具体的工作过程如下：

(1) 用户在 FTP 客户端启动一个控制进程，通过该控制进程向 FTP 服务器的 21 号熟知端口发送文件传输请求。

(2) 服务器端的主控制进程接收到客户端的 FTP 请求，创建一个从属控制进程，由该从属控制进程建立与客户端控制进程间的连接(即控制连接)，然后，主控制进程返回并继续等待响应其他 FTP 用户的并发请求。

(3) FTP 客户端通过其客户端控制进程创建一个客户端的数据传输进程，并通过控制连接和 PORT 命令，将该数据传输进程所使用的端口号通知给服务器端的从属控制进程，进而发出一次数据传输请求。

(4) 服务器端接收到客户端的数据传输请求后，从属控制进程采用 20 号熟知端口，启动一个服务器端的数据传输进程，并向客户端数据传输进程的端口发送请求，进而建立与客户端数据传输进程间的连接(即数据连接)。

(5) 以后的文件传输过程就在客户端的控制进程与服务器端的从属控制进程的控制下，由客户端与服务器端的数据传输进程来完成。

2. PASV 模式

该模式下，数据连接的建立由客户端发起，其控制连接的建立过程与 PORT 模式相同，但数据连接的建立过程有所区别。此时，客户端要求建立数据连接时，使用的是 PASV 命令。客户端通过 PASV 命令告诉 FTP 服务器：它希望连接到服务器的某个端口(非熟知端口)。若 FTP 服务器上该端口可用，服务器就返回 ACK，作为确认信息，然后，建立数据连接；若 FTP 服务器上该端口不可用，服务器就返回 UNACK 信息，表示该端口已经被占用，客户端接到端口不可用信息后，再次发送 PASV 命令，继续要求与服务器上的其他端口建立数据连接。

对于上述两种连接模式，最初客户端通常默认使用 PORT 模式建立数据连接；但是，因为 PORT 模式存在安全隐患，现在，许多客户端默认使用 PASV 模式。

显然，无论对于哪一种连接模式，均可能存在多个用户同时访问同一个 FTP 服务器。此时，主控制进程可以创建多个从属控制进程，以响应来自多个不同客户端的 FTP 请求。此外，还需注意的是，在文件传输期间，控制连接是一直存在的，而数据连接是在每次数据传输请求时创建，数据传输结束后撤销。随着数据连接的撤销，相应的数据传输进程也随之消

亡。当有新的数据传输请求时，再由控制进程创建新的数据传输进程和建立新的数据连接，以再一次完成数据传输任务，直到所有数据传输完毕，控制进程才随之消亡。

3.4.2 FTP的命令和响应

FTP通过建立两条双向连接来分别传输命令/响应信息以及数据文件。为了实现异构网络平台上的信息传输，与Telnet协议一样，FTP也采用NVT格式传输命令和响应信息。FTP定义了许多命令和响应信息，分别用于登录FTP服务器、设置传输参数、浏览文件服务器上的文件和目录、读取文件服务器上的文件、存储文件到服务器以及管理FTP服务器与客户端的文件传输过程。当应用进程调用FTP进行某种操作时，FTP将对应的命令或者响应信息封装成应用层协议数据单元，然后，调用运输层的TCP来完成进一步的数据传输工作。

1. FTP命令

FTP定义了3类FTP命令：存取控制命令、传输参数命令和FTP服务命令。这些命令通过控制连接由客户端传输到FTP服务器，要求服务器完成指定的操作。

1）存取控制命令

该类命令主要用于实现用户身份验证、切换目录、关闭连接等功能。如用户登录FTP服务器时，客户端进程必须连续使用USER命令和PASS命令，将用户名和口令传递给FTP服务器进行认证。常用的FTP存取控制命令见表3.5。

表3.5 常用的FTP存取控制命令

命令与格式	命令功能
USER username	为FTP服务器提供用户名，用于身份验证
PASS password	为FTP服务器提供用户口令，用于身份验证
CWD pathname	改变当前工作目录
CDUP	返回到上一级目录
QUIT	退出FTP登录，关闭控制连接

2）传输参数命令

FTP实现文件传输时，事先要通过控制连接协商某些参数，如数据端口号、数据连接建立方式、传输模式等。实际上，许多传输参数设有默认值，当这些默认值不能满足文件传输要求时，就要使用传输参数命令对传输参数进行协商设定。常用的FTP传输参数命令见表3.6。

表3.6 常用的FTP传输参数命令

命令与格式	命令功能
PORT host-port	使用主动模式传输文件，并将客户端的端口号通知给FTP服务器
PASV	使用被动模式传输文件
TYPE type-code	设置文件的数据类型(type-code=A/E/I/L)
STRU structure-code	设置文件的结构类型(structure-code=F/R/P)
MODE mode-code	设置传输模式(mode-code=S/B/C)

对于异构的网络环境，客户端和 FTP 服务器的系统环境可能存在一定差异，如字长不同，NVT ASCII 码字符在不同系统中的存储表示也不一样。为了保证文件能够被正确地存取和传输，FTP 规定了文件传输和存储的数据表示规范以及传输模式。FTP 的数据表示包括：数据类型和文件结构类型两个方面。

(1) FTP 的数据类型。共包括四种，具体描述如下：

- ASCII 码类型。为默认的数据类型，用于传输文本文件。发送方将本地文件转换为标准的 8 位 NVT ASCII 码形式，然后，在数据连接上传输。
- EBCDIC 类型，即扩充的二进制编码的十进制交换码，是一种类似 ASCII 码规范的编码方式。主要使用在 IBM 计算机上，可用于传输文本文件。
- IMAGE 类型，即通常所说的二进制文件类型，数据打包成 8 位的传输字节，以连续比特流形式进行传输，通常用于传输二进制文件。以二进制文件类型传输数据时，收发双方均不需要进行数据格式的转换，所以传输速度比较快。
- LOCAL 类型，即本地文件类型，用于在具有不同字长的主机间传输二进制文件。

上述 4 种类型中，ASCII 码类型和 EBCDIC 类型最为常用。

(2) FTP 的文件结构类型。共包括 3 种类型，具体描述如下：

- 文件类型。默认的数据结构。此时，文件由连续的字节流组成，不存在内部结构。
- 记录类型。文件由一系列记录组成。该类型适用于表示文本文件，文本文件的每一行就是一条记录。
- 页类型。文件由一组独立的带有编号的页组成，每个页发送时，都带有一个页号，以便接收方能够随机地存储各页。

除了数据表示以外，FTP 还规定了文件在数据连接上如何进行传输。FTP 共规定了 3 种传输模式，具体如下：

① STREAM 模式。即流模式，是默认的文件传输模式。文件以字节流的形式传输。这种传输模式对要传输数据的表示类型没有限制。

② BLOCK 模式。即块模式。文件以一系列数据块的方式进行传输。每个块都带有一个由 3 个字节组成的报头。其中，2 个低位字节为计数字段，用于表示数据块所包含的字节总数；高位字节是位标志的描述符，用于表示数据块的结束标志(EOF 或者 EOR)等信息。

③ COMPRESSED 模式。即压缩模式。对传输的信息进行压缩后再发送，这样有利于节省网络的传输带宽。

3) FTP 服务命令

FTP 服务命令为用户提供了有关文件传输和文件系统操作的一系列功能，这些命令的参数通常是一个路径名(pathname)。除个别命令外，大部分命令的使用顺序不受限制。常用的 FTP 服务命令及功能见表 3.7。

表 3.7 常用的 FTP 服务命令及功能

命令与格式	命 令 功 能
RETR pathname	从文件服务器上下载一个文件
STOR pathname	上传一个文件到文件服务器上
DELE pathname	删除文件服务器上的一个指定文件

续表

命令与格式	命 令 功 能
MKD pathname	在文件服务器上建立一个目录或文件夹
RMD pathname	在文件服务器上删除一个目录或文件夹
PWD	显示当前工作目录
STAT	返回状态信息(如文件上传或下载的字节数)
ABOR	终止前一个命令,并中断数据传输

2. FTP 响应

当 FTP 服务器接收到来自客户端的 FTP 命令并按照命令的要求完成指定操作后,要通过控制连接给客户端返回 FTP 命令的响应信息,这种响应信息也称为 FTP 应答。与 FTP 命令一样,FTP 应答也是采用 NVT 编码方式进行传输,它反映了 FTP 服务器对 FTP 命令的执行情况和服务器的当前工作状态,这些执行情况和状态信息通过 FTP 应答能够及时地反馈给客户端和用户。每条 FTP 命令可以产生一个或者多个 FTP 应答。每个 FTP 应答包括一个 3 位的数字编码和附在其后的一串文本信息。3 位的数字编码为 FTP 应答码,文本信息仅提供给用户阅读并使用户了解命令的执行情况或者服务器的当前状态。部分 FTP 应答码及其含义见表 3.8。

表 3.8 部分 FTP 应答码及其含义

应答码	应答码的含义
125	表示数据连接已建立,传输开始
200	表示命令已被成功执行
220	表示 FTP 服务准备就绪,用户可以登录
226	表示关闭数据连接
331	表示用户名有效,可以输入用户密码
425	表示 FTP 服务器无法建立数据连接
426	连接关闭,传输终止
452	表示 FTP 请求没有被响应,系统没有足够的存储空间
500	表示语法错误(无法识别命令)
503	表示命令顺序错

3.4.3 FTP 的使用

FTP 提供交互式的访问方式完成文件操作。在使用 FTP 操作时,首先,用户需要启动 FTP 客户端程序,通过传统的 FTP 命令、浏览器或者专用的下载工具向 FTP 服务器发送连接请求,通过输入自己的用户名和口令以登录 FTP 服务器。登录成功后就可以访问 FTP 服务器,进行各种文件操作。通常 FTP 提供两种类型的服务。

1. 特许 FTP 服务

特许 FTP 服务指用户在 FTP 服务器上建立自己的账号,在访问 FTP 服务器时,要求

用户输入自己的用户名和口令。该种服务方式下，用户拥有较高的使用权限，但对于没有开设账号的用户来讲，显然无法访问 FTP 服务器。所以，Internet 提供了另一种 FTP 服务方式，即匿名 FTP 服务。

2. 匿名 FTP 服务

该种服务方式为用户建立了一个公开账户，用户名一般为 anonymous，口令一般为 guest。普通用户都可以使用上述用户名和口令登录 FTP 服务器，访问 FTP 服务器上的信息资源。但为了安全起见，对公众开放的信息资源是有限的，因而对用户的使用权限进行了限制，不允许用户上传文件或者修改 FTP 服务器中的文件。

3.4.4 简单文件传输协议

相对于 FTP 而言，还有另一个更为简单的文件传输协议，即简单文件传输协议(Trivial File Transfer Protocol，TFTP)。它是一个简化的文件传输协议，其代码非常简短，可以放在无盘工作站的只读存储器中运行。TFTP 使用 69 号熟知端口，采用无连接的方式(即使用传输层中的 UDP)进行文件传输；而 FTP 是采用面向连接的方式(即使用传输层中的 TCP)。TFTP 不支持交互，因而非常适合多个文件同时传输的情况。但因为在传输层采用了不可靠的 UDP，所以，TFTP 需要有自己的差错保证措施。

3.5 电子邮件服务

电子邮件服务是计算机网络所提供的基本服务方式，是目前使用最为广泛的 Internet 服务之一。电子邮件也称 E-mail，它代替了传统的电报业务，为通信双方提供了简单、廉价、快捷的信息传递方式，深受广大用户的欢迎。

3.5.1 电子邮件系统的组成及工作过程

电子邮件系统一般由 3 部分组成：用户代理(User Agent，UA)、邮件服务器和邮件传输协议，如图 3.11 所示；邮件传输协议又包括简单邮件传输协议(Simple Mail Transfer Protocol，SMTP)、邮局协议(Post Office Protocol，POP3)等。

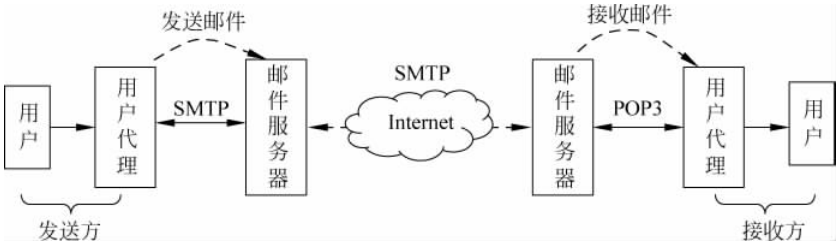

图 3.11 电子邮件系统的组成

用户代理是用户使用电子邮件系统的接口，它运行在用户的本地计算机上，负责为用户生成和管理电子邮件，具体实现电子邮件的编辑、显示、删除等，并负责将邮件发送到邮件服

务器以及从邮件服务器中读取邮件。为了确保邮件能够有效地传输给邮件服务器,用户代理创建了到邮件服务器间的双向连接通道,分别用于发送邮件和接收来自邮件服务器的响应信息。

邮件服务器是为用户提供邮件传输服务的软件系统,也称之为邮件传输代理(Mail Transfer Agent,MTA),它负责接收用户传输邮件的请求,并把要传输的邮件发送给目的邮件服务器。邮件服务器为每个申请邮件服务的用户提供一定大小、称之为邮箱的磁盘空间,用来存储用户收发的邮件。邮箱通过名字进行标识,该名字由用户名和邮件服务器名组成,邮件服务器名即为邮件服务器的域名。此外,邮件服务器还设有邮件缓存以临时存储等待发送的邮件和已接收但尚没有转存到用户邮箱的邮件。

邮件传输协议用于控制用户代理与邮件服务器以及发送邮件服务器与接收邮件服务器间的信息交换。其中,最基本的两个邮件传输协议是 SMTP 和 POP3,它们分别用于控制邮件的发送和接收。

电子邮件系统传输的邮件包括信封、邮件首部和邮件正文 3 个部分。信封上记录有发件人和收件人的地址,收件人的地址可以有多个。邮件首部包括第一收件人的电子邮件地址、发件人、发件人的电子邮件地址、主题(邮件的简要说明)、抄送的邮件地址(可能有多个)、回信应送达的电子邮件地址等信息。正文是邮件的具体内容,最初仅限于 NVT ASCII 文本,后来,经 MIME 协议的扩展可以传输非 ASCII 码字符(如图像、声音等信息)。

邮件服务器兼有发送邮件服务器和接收邮件服务器的双重功能,它们通过 Internet 分别完成邮件的发送和接收。如果邮件服务器在一定时间内没有将用户的邮件成功地发送出去,它要将发送失败的相关信息通知给相关的用户代理。因为邮件服务器独立于客户端,它一直处于运行状态,所以,无论用户是否在线,邮件服务器均能完成邮件的接收工作。

为了有效地控制用户代理与邮件服务器以及邮件服务器与邮件服务器之间的信息交换,Internet 定义了两个协议:邮件发送协议和邮件读取协议,分别用于控制邮件的发送和读取。邮件发送协议使用的是简单邮件传输协议,即 SMTP,它是发送方的用户代理向发送邮件服务器以及发送邮件服务器向接收邮件服务器发送邮件时所使用的协议。邮件读取协议使用的是邮局协议,即 POP,目前采用的是协议的第 3 个版本,即 POP3,它是接收方的用户代理从接收邮件服务器中读取邮件时所使用的协议。上述两种协议的作用范围如图 3.11 所示。

电子邮件的发送和接收均工作于客户/服务器模式,邮件发送和接收的具体过程如下。

(1) 发送方在本地创建一个 SMTP 客户进程,即用户代理,实现对邮件的编辑,并将编辑好的邮件通过 SMTP 发送到发送邮件服务器的邮件缓存中等待发送。

(2) 发送邮件服务器定期扫描邮件缓存,如发现有邮件就一一取出进行发送。对于每个要发送的邮件,首先,通过解析邮件中的收件人邮箱地址中的邮件服务器域名来判断是否为本地邮件;若是本地邮件则直接发送到收件人的邮箱。若待发送的邮件不是本地邮件,发送邮件服务器马上启动低层协议(下一章将要介绍的 TCP),建立与接收邮件服务器 25 号熟知端口的双向连接;若连接成功,则将邮件发送给接收邮件服务器;若连接失败,或者因其他原因导致发送失败,则每隔一定时间再重复发送一次;若发送规定次数后仍没有发送成功,则将失败信息通知给发送方(即用户代理)。

(3) 接收邮件服务器接收到电子邮件后,将它们存放到收件人的邮箱中等待用户读取。

(4) 当读取邮件时,收件方在本地创建一个 POP 客户进程,即用户代理,通过 POP3 与接收邮件服务器的 110 号熟知端口建立连接,连接成功后就可以从接收邮件服务器中读取邮件,然后进行阅读和处理。

3.5.2 SMTP

如前所述,Internet 上无论是客户端到邮件服务器间的邮件发送,还是邮件服务器间的邮件传输均采用 SMTP 进行控制。SMTP 采用客户/服务器工作模式。当用户要发送邮件时,首先启动 SMTP 客户进程,并使用 25 号熟知端口建立与邮件服务器端 SMTP 服务器进程间的双向连接,然后通过这条连接传送邮件以及之间交换的命令和应答信息。

SMTP 规定了 14 条 SMTP 命令和 21 种应答信息。这些命令和应答信息采用 NVT ASCII 文本格式,并以明文方式进行传输。下面简单介绍一下主要命令的功能及格式,其中,< SP >表示空格,< CR-LF >表示行结束符。

1. HELO

命令格式: HELO < SP ><域名>< CR - LF >

功能: 用于启动邮件传输过程。发送方以自身域名作为参数来标识身份,接收方通过返回自己的域名进行确认。

2. MAIL

命令格式: MAIL < SP > FROM: < reverse - path >< CR - LF >

功能: 用于初始化邮件传输。< reverse-path >为逆向路径,用于指出到达邮件发送方的路径,一般为发件人的电子邮件地址; 当邮件在传输过程中出现问题时,错误信息会通过这些路径传输到发件人。

3. RCPT

命令格式: RCPT < SP > TO: < forward - path >< CR - LF >

功能: 用于标识单个收件人。< forward-path >为前向路径,一般为收件人的电子邮件地址。当有多个收件人时,该条命令要重复使用。

4. DATA

命令格式: DATA < CR - LF >

功能: 用于将邮件报文发送给邮件服务器。该命令在 RCPT 命令后面使用,声明开始传输邮件数据。邮件数据要求以仅有一个“点”的行表示结束。当传输到仅有一个“点”的行时就表示邮件数据已经传输完毕。

5. QUIT

命令格式: QUIT < CR - LF >

功能: 表示结束邮件传输。该命令用于终止客户端与服务器间的连接,服务器会返回

一个代码为 221 的应答信息。

6. RSET

命令格式：RSET < CR - LF >

功能：使客户端与服务器端的连接复位。该命令使当前的邮件事务异常终止，所存储的关于发件人和收件人的所有信息被删除，复位客户端与服务器端的连接。

应答信息是从邮件服务器发送给客户端的响应信息，每个应答信息是一行文本，包括 3 位十进制数组成的应答码和附加的描述信息。SMTP 命令的主要应答代码及代码含义见表 3.9。

表 3.9 SMTP 的主要应答代码及代码含义

应答代码	代 码 含 义	应答代码	代 码 含 义
220	服务准备就绪	500	语法错，不能识别的命令
221	关闭连接	501	参数语法错
250	请求操作就绪	503	命令顺序错
354	开始邮件传输	550	操作未执行：邮箱不可用
421	服务不可用	552	操作中止：存储空间不足
450	操作未执行：邮箱忙	553	操作未执行：邮箱名不正确
451	操作中止：本地出错	554	传输失败

下面使用 SMTP 命令和应答信息对邮件发送过程进行详细描述。该过程包括 3 个阶段：建立连接、传输邮件和终止连接。具体过程如下：

(1) 建立连接。客户端启动客户进程向邮件服务器的 25 号端口发出请求，要求建立 TCP 连接。当连接成功后，服务器返回代码为 220 的应答信息，表示服务准备就绪。然后，客户端向服务器发送 HELO 命令以标识发件人自己的身份。如果服务器允许接收邮件，则返回代码为 250 的应答信息；若服务器不可用，则返回代码为 421 的应答信息，表示暂时不能提供服务。

(2) 传输邮件。当连接建立成功后，客户端首先发送 MAIL 命令以标识发件人的信息。此时，如果服务器已准备接收邮件，则返回代码为 250 的应答信息；否则，根据错误返回相应代码的应答信息，并说明出错原因。

当邮件服务器已准备接收邮件并返回正确的应答信息后，客户端发送 RCPT 命令标识接收方；若有多个收件人，则需要多次发送 RCPT 命令。如果邮件服务器同意为收件人接收邮件，则返回代码为 250 的应答信息；如果不同意，则返回代码为 550 的失败信息。

当接收到邮件服务器同意为收件人接收邮件的应答信息后，客户端发送 DATA 命令给服务器，服务器端返回代码为 354 的应答信息进行确认，通知客户端可以开始发送邮件数据。客户端将邮件数据按行发送，当服务器检测到接收的一行数据仅有一个“点”时表示邮件数据传输结束，然后，返回代码为 250 的应答信息。

(3) 终止连接。当邮件数据传输结束后，客户端向邮件服务器发送 QUIT 命令，邮件服务器返回代码为 221 的应答信息表示关闭服务，同意释放之间的 TCP 连接，结束邮件传输过程。

注意，上述过程可以是用户代理向发送邮件服务器发送邮件，也可以是发送邮件服务器向接收邮件服务器发送邮件。

3.5.3 POP3

与 SMTP 一样，POP3 也采用客户/服务器工作模式以及命令-响应方式进行信息交换，其命令和响应信息也采用 NVT ASCII 文本格式。每个 POP3 响应信息包括：状态码和附加说明两部分。其中，状态码非常简单，只有两个："＋OK"和"－ERR"，分别表示操作正确和操作失败。POP3 的主要命令及功能见表 3.10。

表 3.10 POP3 的主要命令及功能

命令及格式	功　能	命令及格式	功　能
USER username	指定用户名	DELE［Msg＃］	删除指定的邮件
PASS password	指定密码	NOOP	空操作
STAT	询问邮箱状态（如邮件总数和总字节数等）	RSET	重置所有标记为删除的邮件，用于撤销 DELE 命令
LIST［Msg＃］	列出邮件索引（如邮件数量和每个邮件的大小等）	QUIT	提交修改，并断开连接
RETR［Msg＃］	取回指定的邮件		

当用户使用 POP3 接收邮件时，用户首先在本地计算机上启动用户代理进程，即 POP3 客户程序，使用 110 号熟知端口建立与邮件服务器上 POP3 服务器进程间的双向连接，然后，通过这条连接进行邮件读取。注意，POP3 服务器同时还是邮件接收服务器，以通过 SMTP 接收其他邮件服务器发送来的邮件。当用户启动 POP3 客户程序并使用 USER 和 PASS 命令正确输入用户名和口令后就登录到 POP3 服务器。客户端通过发送 POP3 命令进行相应操作，服务器接收命令并做出响应，使客户端读取其邮箱中的邮件。当邮件读取结束后，客户端发出 QUIT 命令，服务器确认用户的操作，关闭客户端与服务器端间的连接。邮件读取过程结束。

POP3 属于离线式协议，不能对邮件实现在线操作，要读取的邮件必须下载到本地计算机上。而且，一旦邮件被读取它就从 POP3 服务器中被删除，用户就不能在其他计算机上再读取这个邮件，显然这给用户造成了许多不便。为此对 POP3 进行了功能扩充和改进，可以设定邮件被读取后仍保留在 POP3 服务器中的时间，从而方便用户的使用。

3.5.4 电子邮件协议的扩充

SMTP 和 POP3 是最初使用的电子邮件传输协议，它们或多或少存在一定的缺陷；随着 Internet 应用的普及，电子邮件传输协议逐渐得到了完善而走向成熟。

1. 邮件发送协议的扩充

Internet 中最初采用的邮件发送协议为 SMTP。该协议为了消除系统间的异构性，邮件正文采用 NVT 编码进行传输。而且 SMTP 只能传输可打印的 7 位 ASCII 码数据，这就限制了邮件的使用范围。为了能够传输更多类型的电子邮件，1993 年提出了一个辅助协

议：通用因特网邮件扩充协议(Multipurpose Internet Mail Extensions，MIME)。MIME 协议对 Internet 所能传输邮件的种类进行了扩充，它可以传输文本、图像、音频、视频等多种类型的数据。注意，MIME 协议并不是一个独立工作的邮件传输协议，它不是取代 SMTP 和 POP3，而是与它们一起工作，它位于用户代理和 SMTP/POP3 之间。在发送端，MIME 协议接收到非 ASCII 码数据后，将之转换成 7 位的 NVT ASCII 码数据，然后，交给 SMTP 邮件服务器进行传输；在接收端，邮件服务器接收到 NVT ASCII 码数据，然后，经 POP3 交付给 MIME 协议进行转换，恢复成原来的数据后再交给用户。MIME 协议的作用如图 3.12 所示。

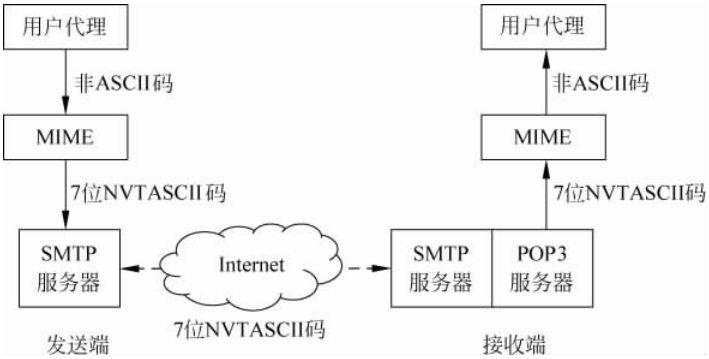

图 3.12 MIME 协议的作用

2. 邮件读取协议的扩充

目前，用户从接收邮件服务器中读取邮件时，除了使用邮局协议(POP3)外，还有因特网报文存取协议(Internet Message Access Protocol，IMAP)。

IMAP 是一种比 POP3 更为复杂、功能更强的邮件读取协议。它也是工作在客户/服务器模式。该协议是一种联机协议，运行在用户主机上的 IMAP 客户程序要一直保持与 IMAP 服务器的连接，被读取的邮件也一直保存在 IMAP 服务器上，直到被用户主动删除。这样用户就可以随时使用其他计算机通过网络连接到 IMAP 服务器读取信箱中的邮件。但该协议因一直要保持与 IMAP 服务器的连接而占用过多的服务器时间。

3.6 WWW 服务

WWW 是 World Wide Web 的简称，也称为万维网或环球信息网，是由欧洲原子核研究委员会于 1989 年提出的。像文件传输、电子邮件等各种网络服务一样，WWW 虽然被称为万维网，但它并不是一种计算机网络，而只是 Internet 所提供的一种应用。通过 WWW 服务(也称为 Web 服务)，用户可以浏览 Internet 上的各种信息。众所周知，因特网规模庞大，它由许多异构的计算机系统所组成，存储着海量的信息资源。这些资源包括文本、图像、声音、视频、动画等各种信息。那么，在因特网环境下，如何表示如此丰富多样的信息资源？如何在茫茫的信息海洋中寻找我们想要的信息？Internet 上如何传输这些信息等等。在开始介绍这些知识前，先介绍超文本(hypertext)、超媒体(hypermedia)、WWW 浏览器等几个重

要概念。

首先，Internet 上以超文本文件的形式表示和存储各种信息资源。这种超文本文件不但含有文本信息，而且还含有链接信息；通过单击链接信息就可以跳转到文件的另一处，或者跳转到另一个超文本文件，而这些超文本文件可能位于网络中的不同计算机上，常把这种链接称为超链接。显然，超文本文件具有非线性的信息组织形式。

当超文本文件中的链接信息为多媒体信息时，就称这种超文本为超媒体。所链接的多媒体信息可以是文字、图形、图像、动画、声音、视频、表格等。通过灵活地设置超链接的内容，就可以非常方便地访问和浏览 Internet 上的各种信息。这些被访问的信息可能位于本地，也可能位于异地，只要它们连接到 Internet 上就可以被访问到。

超文本文件采用超文本标记语言（Hyper Text Markup Language，HTML）进行编写。HTML 语言是一种制作网页的标准语言，万维网上不同种类的计算机间通过这种语言可以非常方便地进行信息交流。

Internet 上的各种信息资源采用统一资源定位符（Uniform Resource Locator，URL）来唯一地进行标识和定位。URL 给出了信息资源在网络上的具体位置，即指出了该信息在哪台服务器的哪个文件夹下，同时，还指出了访问这种资源的方式。

WWW 浏览器是用户访问网络信息资源的工具软件，如 IE 浏览器、360 浏览器、Google 浏览器等。这些浏览器具有友好的图形界面，用户只需输入所要访问资源的 URL，浏览器就可以完成在 Internet 上的信息搜索并将结果显示给用户。

3.6.1 WWW 服务的工作原理

WWW 服务采用客户/服务器工作模式，它包括：客户机、WWW 服务器和 HTTP 等 3 个组成部分，如图 3.13 所示。其中，客户机是用户的本地计算机，WWW 服务器是用户访问的信息资源所在的服务器；HTTP 控制客户机访问 WWW 服务器的整个交互过程。

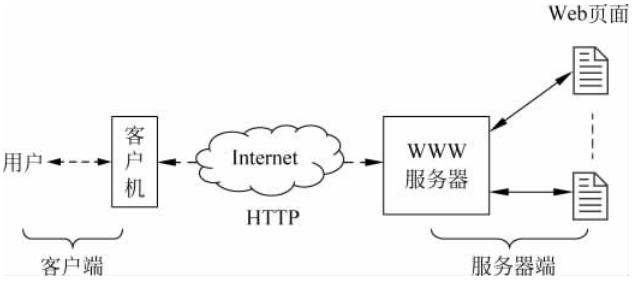

图 3.13 WWW 服务的工作原理示意图

用户通过 HTTP 访问因特网上信息资源，信息资源的具体位置由 URL 给出，具体过程如下：

（1）用户在本地浏览器上输入想要访问信息资源的 URL。

（2）浏览器接收该 URL 后，向 DNS 服务器发送请求要求进行域名解析。

（3）DNS 服务器接受浏览器的请求对 URL 中的域名进行解析，并把查找到的 IP 地址发送给浏览器。

(4) 浏览器向该 IP 地址所标识的服务器的 80 号熟知端口发出一个 Web 请求。

(5) 服务器接受来自用户的 Web 请求,并建立与客户机间的 TCP 连接。

(6) 浏览器将所要访问信息资源在服务器上的具体位置通过 HTTP 中的 GET 方法通知给服务器。

(7) 服务器根据请求寻找到相关的 Web 信息,然后,通过 HTTP 将查找的结果信息返回给客户端。

(8) 服务器和客户机释放 TCP 连接。

(9) 客户机将接收到的返回信息以 Web 页面的形式呈现在显示窗口上,供用户浏览。

客户机与 WWW 服务器交换信息的基本单位为 Web 页面(Web page)。Web 页面一般是通过超文本标记语言(HTML)来进行组织和编写的。HTML 和 HTTP 构成了 WWW 服务的基础。

3.6.2 超文本标记语言

超文本标记语言(HTML)是编写 Web 程序和建立超文本文件的工具,是一种标准化的页面描述语言,它使用文本标志来说明结构元素、输出格式、显示图像和超链接等,使得同一页面能够以相同的格式在不同的计算机系统上显示出来,从而对用户屏蔽了网络系统的异构特性,方便了用户的使用。

采用 HTML 编写的程序通常由普通文字和标签组成,其中的标签用来定义 Web 网页的显示方式、链接方式等。HTML 的标签可分成基本结构标签、连接标签、列表标签、字属性标签等。HTML 程序由 HTML 解释程序解释执行。当 WWW 服务器将请求的页面返回浏览器,浏览器根据其显示器的分辨率重新组织和显示页面。

HTML 的发明和成功应用对 Internet 的普及和推广起到了积极的促进作用。但随着 Web 信息的增加和所涉及内容的复杂化,HTML 也逐渐暴露出许多不足,具体为:

(1) 可扩展性较差。由于 HTML 不允许用户定义自己的标签,从而限制了用户对数据语义的准确描述。

(2) 结构描述能力弱。HTML 不支持深层次的嵌套表达,无法充分描述结构复杂的文档数据,特别是一些关系数据或者面向对象的层次结构数据。

(3) 缺乏有效的确认机制。HTML 没有提供对所描述数据结构的正确性进行确认的机制,造成了 HTML 文档本身的不规范性。

为了更好地适应 Web 技术的发展,在继承了 HTML 优点的基础上对其进行了改进,提出了可扩展标记语言(Extensible Markup Language,XML)。XML 具有允许用户自由定义标签、支持元素任意层次的嵌套等优点,很好地克服了 HTML 的不足。

3.6.3 统一资源定位符

众所周知,要在个人计算机上访问一个文件,必须给出相应的路径和文件名。显然,要在计算机网络上访问一个文件,不但要给出文件名及其路径,而且还要给出文件所在服务器的地址以及访问文件的方式。Internet 中使用统一资源定位符(URL)来唯一地标识网络资源的位置,可以把 URL 看成是单机环境下的文件名概念在网络环境上的扩展和延伸。

URL 的一般格式如下：

<URL 访问方式>: //<主机地址或域名>: <端口号>/<路径名和文件名>

其中“主机地址或域名”指出所要访问的目的主机或者服务器在网络上的具体位置；“端口号”指出需要得到的服务类型(如 80 号端口对应 WWW 服务，25 号端口对应电子邮件服务)，该项可以省略；“路径名和文件名”指出所访问信息资源在目的主机上的具体位置，该项可以省略；几种常见的“URL 访问方式”有 ftp、http 和 news，分别对应文件传输服务、超文本信息浏览和 USENET 新闻。

各种 URL 访问方式中 ftp 和 http 的简化格式如下：

ftp: //<ftp 服务器的地址或域名>
http: //<WWW 服务器的地址或域名>

如：麻省理工学院的匿名文件服务器和北京大学的 WWW 服务器的 URL 分别为：

ftp: //rtfm.mit.edu 和 http: //www.pku.edu.cn

当要通过浏览器访问本地的信息资源时，URL 的格式如下：

file: ///<路径名和文件名>或者 file: ///localhost/<路径名和文件名>

其中 localhost 为本地主机名。

前面已经讲过，浏览器给用户访问各种网络信息资源提供了一种有效手段。用户若想访问某种 Web 信息资源，则只需在浏览器的打开文件一栏输入相应的 URL 即可。

3.6.4 HTTP

HTTP 为支撑 WWW 服务的网络传输协议，是一个面向对象的应用层协议，由于其简捷、快速，故非常适用于分布式超媒体信息系统。该协议于 1990 年提出，经过多年的使用与发展，不断得到扩展和完善，陆续出现了 HTTP 1.0、HTTP 1.1 等版本，使得 WWW 服务器一次仅能接收一个 Web 请求发展到可以同时响应多个 Web 请求，而且目前新的协议版本：HTTP-NG(HTTP-Next Generation)也已经出现。

在 Internet 提供 Web 服务的过程中，运行在不同端系统上的客户程序和服务器程序通过 HTTP 交换 Web 消息。HTTP 定义了这些消息的结构以及客户和服务器间交换这些消息的方法和步骤。HTTP 具体包括两部分：资源定位和消息内容格式。资源定位采用 URL 指明所要访问的信息资源的位置。HTTP 采用电子邮件的 MIME(Multipurpose Internet Mail Extensions)协议定义所传输数据的格式，以使 HTTP 可以传输包括多媒体在内的多种类型数据。HTTP 定义了两类数据报文：请求报文和响应报文，并详细规定了每种报文的格式。当客户进程接收到客户的请求并按照请求报文的格式封装成报文，通过相应端口传输给下一协议层(即传输层)，然后，通过网络层等低层协议和通信线路将请求报文发送给 WWW 服务器。WWW 服务器接收到客户的请求，然后按照客户的请求进行相应的操作并把结果封装成响应报文，再经过下面的运输层、网络层等低层协议和通信线路发送给客户。

HTTP 的主要特点可概括如下：

(1) 支持客户/服务器模式。

(2) 简单快速。客户和服务器交互请求服务时,只需传送请求方法和应答状态。请求方法封装在请求报文中,由客户端发出,要求服务器完成指定的操作。请求报文有固定的格式,其中第一行是请求行,该行包括:请求方法、请求资源的 URL 和 HTTP 的版本号 3 部分。常用的请求方法有:

- GET:请求读取 URL 所标识的页面。
- HEAD:与 GET 类似,但只请求读取页面的首部。
- PUT:与 GET 的功能相反,PUT 操作要求存入一个页面,用于新增或者修改页面。
- POST:与 PUT 的功能类似,但仅是把数据附加在原来页面的后面。
- DELETE:请求服务器删除 URL 所标识的资源。

应答状态以编码的形式封装在响应报文中,响应报文的第一行为状态行,该行包括 HTTP 的版本号、状态编码以及解释状态编码的短语等 3 个字段。服务器通过响应报文将自身的状态等信息通知给客户端。每个状态编码由 3 位数字组成,共分成 5 类:

- 1xx:表示服务器发给客户机的通知信息。如要求客户机继续发送剩余的请求。
- 2xx:表示客户机的请求是否被服务器成功地接收、理解和接受。
- 3xx:指明为了完成 Web 请求需要客户进程所要进一步完成的操作。
- 4xx:表示客户机出现错误的各种情况,如客户请求超时、客户请求未授权等。
- 5xx:表示服务器出现错误的各种情况。如服务器不可用、HTTP 版本不支持等。

每类编码的后两位数字表示更为具体的详细信息。

(3) 灵活。虽然被称为超文本传输协议,但 HTTP 允许传输任意类型的数据对象(包括多媒体数据等),具体由请求报文和响应报文中的消息首部(message header)来说明所传输信息的具体类型。

(4) 无连接。HTTP 建立在可靠的、面向连接的传输控制协议(即 TCP)基础之上。无连接的含义是指 HTTP1.0 限制 WWW 服务器每次连接只处理一个请求。服务器处理完客户的请求,在收到客户的应答后,立即断开连接。采用这种方式使得 WWW 服务器实现起来非常简单,而且可以避免服务器为了保持过多的 TCP 连接而浪费系统资源。但这种短连接的频繁建立和拆除增加了网络传输数据包的数量而容易引起网络拥塞,故 HTTP1.1 对其进行了改进,允许浏览器发出一次 Web 连接请求可以从服务器上下载多个文件。

(5) 无状态。HTTP 是无状态协议。无状态是指协议对于事务处理没有记忆能力,即服务器不保留与每个客户交互时的各种状态信息,从而大大减轻了服务器的记忆负担,使其实现简单,程序规模小,响应速度快。但没有记忆能力意味着如果后续处理需要前面的信息,则服务器必须重传,这样可能导致增大每次连接传送的数据量。

HTTP 完成一次 WWW 服务需要经历 4 个阶段:建立连接、请求服务、应答和关闭连接,如图 3.14 所示。

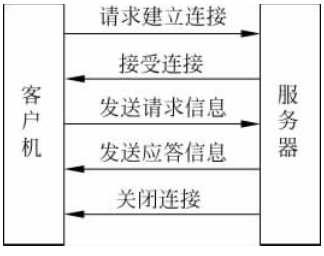

图 3.14 HTTP 完成一次 WWW 服务的过程

① 建立连接:用户输入链接地址后,浏览器查找相应的 WWW 服务器的主机名并与之建立连接,其中包括调用 DNS 服务器进行域名解析以获得 WWW 服务器的 IP 地址。

② 请求服务：一旦建立了与 WWW 服务器的连接，浏览器向服务器发送一个请求，指明所请求的 Web 页面、所用的协议及版本、语言及版本、所能接收的 MIME 类型、编码类型等信息。

③ 应答：WWW 服务器响应客户进程的请求，把用户要求的数据文件按 MIME 格式传输给客户机。

④ 关闭连接：客户端接收服务器所返回的信息，并通过浏览器显示在用户的显示屏上，然后，客户机与服务器断开连接。

HTTP 支持的连接方式主要有两种。第一种是直联方式，即客户进程直接与服务器进程建立连接。此时，客户进程想要访问某个 WWW 服务器时，便向其发出连接申请；WWW 服务器监听网上的连接请求，当请求到达时，就创建新的进程来处理请求，并将应答信息传输给客户进程。

第二种连接方式在客户端和 WWW 服务器之间增加了一个中间环节：代理服务器。代理服务器通常位于用户网络的出口。在这种方式下，客户机将请求先发给代理服务器，代理服务器再作为客户机转发该请求给 WWW 服务器并与之建立连接。WWW 服务器的响应信息先发给代理服务器，然后再经代理服务器转发给客户机。显然，分别从客户机和 WWW 服务器的角度看，代理服务器有时作为服务器，有时又作为客户机，它扮演着客户机和服务器的双重身份，所以称之为代理服务器。

为什么在客户端和服务器之间增加一个代理服务器呢？因为代理服务器在计算机网络中有着非常重要的作用，主要体现在两方面：安全代理和缓存代理。

安全代理是为了将内部计算机网络与外部计算机网络相隔离，使得外部网络用户只能看到代理服务器，而且只能通过代理服务器来访问内部网络。从而，通过代理服务器实现了内网对外部用户的屏蔽，使他们无法知道内部网络的具体情况，从而增强了内部网络的安全性。防火墙等安全软件常安装在代理服务器上。

缓存代理的设置主要是为了提高 Web 信息的访问速度。在代理服务器上设置了高速缓存，用于存储用户刚访问过的 Web 页面。当代理服务器接收到客户机的请求，它首先检查其高速缓存，看是否存在所请求的 Web 页面。若有，则直接从高速缓存中取出，送给客户机；若无，则向 WWW 服务器转发请求，并将得到的响应信息发送给客户机，同时也存储到高速缓存中。注意，为了保证高速缓存内信息的有效性，此时要为高速缓存中的信息设置一个有效时间，以定期清除或更新。显然，缓存代理的存在，不但提高了 Web 信息的访问速度，而且也降低了网络通信量，从而减轻了网络的通信负荷。

上面所介绍的两种代理可以同时工作于同一台代理服务器上。代理服务器除具有上面所介绍的两种功能外，它还具有控制流量、减少用户上网费用以及采用地址转换技术、解决 IP 地址不足等作用。

3.7 动态主机配置协议

通常网络中计算机的 IP 地址有两种配置方式：手动配置和自动配置。对于手动配置方式，IP 地址、默认网关等信息需要通过手工输入；当需要配置的计算机数目较多时，不但配置起来麻烦，需要时间长，而且往往发生输入错误，导致不能正常地进行配置，也可能配置

的IP地址因重复而导致冲突。所以,为了增强网络管理的灵活性和使用的方便性,常常不给临时上网的用户分配固定的IP地址,而是在这些用户上网时,由系统软件自动地给他们分配临时的IP地址,对于移动用户更是如此。这样,不但降低了IP地址管理的复杂度和工作量,用户使用起来特别方便,而且避免了由于手工配置易于引起的IP地址冲突等问题。

网络中负责管理和为用户分配IP地址的系统软件称为动态主机配置协议(Dynamic Host Configuration Protocol,DHCP)服务器,每个网络必须具有至少一台DHCP服务器,它们负责本网络内IP地址的分配,而且每个DHCP服务器所能分配的IP地址范围是不同的,该范围称为DHCP服务器的作用域。DHCP服务器工作于客户/服务器模式,它执行的协议称为动态主机配置协议,简称DHCP。该协议是应用层的一个协议,它通过租借过程为客户端动态地分配或释放IP地址。客户端租借到的IP地址及相关信息并不是永久有效的,而往往有一个使用期限,该期限通常称为租约周期。当超过租约周期后,申请到的IP地址等信息失效,需要重新申请。客户端和DHCP服务器间通过报文交换来实现IP地址的动态分配和释放。DHCP定义的报文有如下几种。

- DHCPDISCOVER报文:即DHCP发现报文,客户端使用该报文定位DHCP服务器。该报文是DHCP客户端在首次尝试登录网络并向DHCP服务器请求分配IP地址时发出的消息。
- DHCPOFFER报文:即DHCP提供报文,由DHCP服务器发给客户端,是对DHCP发现报文的响应,其中,包括临时分配给客户端的IP地址及相关信息;而且DHCP服务器保证在没有接收到客户端的响应信息前不会把该IP地址分配给其他客户。
- DHCPREQUEST报文:即DHCP请求报文,是客户端以广播方式发送给DHCP服务器的报文。当客户端以广播方式发送DHCP发现报文后,可能有多个DHCP服务器均接到请求,它们均给客户端发送DHCPOFFER报文。但是,客户端仅采用第一个收到的DHCPOFFER报文,然后,以广播方式发出一个DHCPREQUEST报文作为响应,表明接收该DHCP服务器提供的响应数据。DHCPREQUEST报文中含有为该客户端提供IP地址的DHCP服务器的标识信息,所有其他的DHCP服务器收到该DHCPREQUEST报文后,知道客户端没有采用它们提供的IP地址,然后,都将刚分配出去的IP地址收回,留给其他IP租约请求使用。
- DHCPACK报文:即DHCP确认报文,是DHCP服务器发送给客户端的报文,用来确认并完成IP地址的配置过程。DHCP服务器通过该报文通知客户端分配给它的IP地址及相关信息有效。当客户端接收到该确认报文后,会用DHCP服务器提供的IP配置信息初始化TCP/IP,并将TCP/IP绑定在网络服务和网络适配器上。
- DHCPNAK报文:即DHCP否定报文,也是DHCP服务器发送给客户端的报文。如果DHCP服务器所提供的IP地址不再有效或者已被其他计算机所使用,它就发送该报文给客户端;客户端接收到该报文后将开始一个新的IP地址租约过程。

在实际运行过程中,需要申请IP地址的主机因为不知道DHCP服务器的IP地址,所以,在启动时以客户端的身份采用广播方式向网络上的所有结点发送DHCP发现报文,要求给它分配一个IP地址。显然,网络上的所有结点均能接收该DHCP发现报文,但只有DHCP服务器响应该报文,因为它根据接收的报文头部得知该报文是向它(们)发出请求的

DHCP 发现报文；然后，DHCP 服务器首先查找其数据库，判定是否有该主机的配置信息（事先已配置并存储好的，如地址绑定）；若有，则取出其 IP 地址并连同子网掩码、默认网关等信息一起形成 DHCP 提供报文返回给发出申请的客户端；若无，则在其 IP 地址池中，取出一个空闲的 IP 地址，同样，也要形成 DHCP 提供报文，将分配的 IP 地址等信息通知给客户端。发出请求的主机可能接收到多个 DHCP 服务器发出的 DHCP 提供报文，但是，它仅选择第一个接收到的 DHCP 提供报文所对应的 DHCP 服务器；然后，采用该 DHCP 服务器的标识信息形成 DHCPREQUEST 报文，并以广播方式发送给所有 DHCP 服务器，被选中的 DHCP 服务器接收到该 DHCPREQUEST 报文后，再形成 DHCPACK 报文发送给客户端，表示该次 IP 地址租约过程成功，客户端可以使用分配给它的 IP 地址进行通信。而没有被选中的 DHCP 服务器，则自动收回刚分配出去的 IP 地址。IP 地址的动态分配过程如图 3.15 所示。主机（客户端）在应用层发出的 DHCP 发现报文采用 68 号熟知端口进行标识，并经过运输层的 UDP 协议发送给下层协议，而 DHCP 服务器使用的是 67 号熟知端口来处理 DHCP 请求。

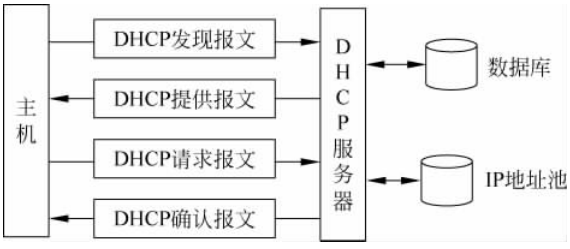

图 3.15　IP 地址的动态配置过程

当客户端动态申请到一个 IP 地址并使用一段时间后，可能超出了租约周期，此时，客户端必须开始新的 IP 地址租约过程。DHCP 提供了自动续订租约功能。当申请到的 IP 地址使用时间超过了 50％的租约周期，DHCP 客户端自动尝试续订租约。此时，它直接向对应的 DHCP 服务器发送一个 DHCPREQUEST 报文来续订租约。如果该 DHCP 服务器可用，则它将续订该租约并向该客户端发送一个 DHCPACK 报文；该报文中含有新租约的持续时间和需要更新的参数，客户端收到该确认报文后更新其配置。如果 DHCP 客户端首次续订失败，则一旦超过了 87.5％的租约时间，该客户端会广播一个 DHCP 发现报文以开始新的地址租约过程。此时，该客户端会接受任一台 DHCP 服务器所发布的租约。

如果客户端在发出 4 次请求后，均没有得到任何 DHCP 服务器的响应，则在 169.254.0.1～169.254.255.254 之间的保留地址范围内选择一个 IP 地址；客户端使用这些 IP 地址只能在本子网中进行通信。此后，每隔 5 分钟，客户端将持续尝试寻找可用的 DHCP 服务器。当 DHCP 服务器可用后，客户端将申请到有效的 IP 地址，然后，就可以与子网外的主机进行通信了。

同样，在租约周期内的任何时候，客户端也可以向 DHCP 服务器发送 DHCPRELEASE 报文来释放 IP 地址等配置信息，以取消剩余的租约。

在实现动态 IP 地址分配的过程中，为了确保 DHCP 服务的可靠性，根据微软的建议，常采用 80/20 规则在网络中设置两台 DHCP 服务器；这两台 DHCP 服务器位于同一子网，但具有不同的作用域，分别分配 80％和 20％的 IP 地址。两个 DHCP 服务器中 IP 地址的分

配比例可以灵活调整，但是绝对不允许存在相同的 IP 地址。

DHCP 非常适合配置位置经常变动的计算机。对于运行在 Windows 环境下的计算机，若想动态获取临时的 IP 地址，则需要在控制面板的“网络连接”图标中选中“本地连接”的“属性”选项，双击“TCP/IP”，然后，选择“自动获取 IP 地址”选项。这样该计算机在启动时通过 DHCP 自动获取一个动态的 IP 地址。

3.8 其他 Internet 服务

TCP/IP 的应用层除提供上面介绍的各种服务外，还提供网络新闻组(UseNet)、电子公告板(BBS)、菜单式信息查询系统(Gopher)、广域信息服务(WAIS)等。

UseNet 也称为新闻论坛，是人们利用 Internet 发表看法、交换创意、收集信息以及回答问题的网络新闻系统软件。它利用网络新闻传输协议(Network News Transfer Protocol，NNTP)在 Internet 上发布网络新闻，并对新闻提供分类管理功能，用户根据兴趣自由选择新闻讨论小组，检索、阅读新闻或发表自己的见解。

电子公告板(Bulletin Board Service，BBS)是 Internet 上的一种电子信息服务系统，是一种强有力的信息交流工具。它提供一块公共电子白板，各个用户可以在上面发布信息，提出看法，也可以方便、迅速地获取公告信息。目前，许多大专院校的校园网上都建立了各具特色的 BBS 系统。

菜单式信息查询系统 Gopher 是一个综合性的网上文件查询工具。Gopher 工作于客户/服务器方式。Gopher 服务器维护一个供用户访问的菜单集，每个菜单指向一个特定的信息。Internet 上存在多个 Gopher 服务器。Gopher 客户一般被指定访问一个默认的 Gopher 服务器。当 Gopher 客户机启动后，它就自动地与默认的 Gopher 服务器建立连接。Gopher 服务器将主菜单发送给 Gopher 客户机，Gopher 客户选择菜单中的某一项后，便向 Gopher 服务器发送请求，Gopher 服务器根据请求将相应的菜单或信息返回给 Gopher 客户机。这样 Gopher 客户根据 Gopher 服务器提供的菜单就可以检索和浏览 Gopher 服务器提供的各种信息。

WAIS 是一个网络环境下的数据库查询工具，它利用关键词来搜索信息。WAIS 也是工作于客户/服务器方式。WAIS 客户发送包括检索关键字的请求给 WAIS 服务器，WAIS 服务器根据客户的请求检索相应的信息，并将结果返回给客户机。

习题

(1) Internet 为用户提供哪些服务？这些服务使用了应用层的哪些协议？

(2) 什么是客户/服务器工作模式？

(3) 什么是守护进程？

(4) 什么是端口和套接字？它们的作用分别是什么？

(5) 什么是域名？采用域名的意义是什么？域名是如何进行组织和管理的？

(6) 什么是域名解析和域名服务器？请解释域名解析的具体过程。

(7)(　　)一定可以将其管辖的主机名转换为 IP 地址。

A. 本地域名服务器　　B. 根域名服务器

C. 授权域名服务器　　D. 代理域名服务器

(8) 对于递归域名解析,由(　　)返回 IP 地址。

A. 最开始连接的域名服务器　　B. 最后连接的域名服务器

C. 目的地址所在的域名服务器　　D. 不确定

(9) 解释域名解析过程中设置高速缓存的作用。

(10) 请解释 NVT 格式在 Telnet 服务中的作用。

(11) 请说明 FTP 中控制连接和数据连接的建立过程以及作用?

(12) 一次 FTP 会话中存在(　　)个 TCP 连接。

A. 0　　B. 1　　C. 2　　D. 3

(13) 在一次 FTP 会话期间,协议打开数据连接(　　)。

A. 一次　　B. 两次　　C. 按需要定　　D. 以上均不正确

(14) 请说明电子邮件系统的组成以及各个部分的作用。

(15) SMTP 和 POP3 以及 POP3 和 IMAP 的主要区别是什么?

(16) POP3 采用(　　)模式,当客户机需要服务时,客户端软件与 POP3 服务器建立(　　)连接。

A. B/S　　B. C/S　　C. TCP　　D. UDP

(17) MIME 允许(　　)数据通过 SMTP 进行传输。

A. 音频　　B. 非 ASCII 码数据

C. 图像　　D. 以上所有

(18) 什么是 WWW 服务? 什么是超文本和超媒体?

(19) 什么是超文本标记语言? 其特点是什么?

(20) 什么是 URL? 请写出其一般格式,并说明其作用。

(21) 什么是 HTTP? 其特点是什么?

(22) 说明两个不同版本 HTTP 的主要区别。

(23) 说明 HTTP 完成 WWW 服务需要经历的四个阶段。

(24) 请说明在 WWW 服务中代理服务器的作用。

(25) 从协议分析的角度看,WWW 服务的第一步一般是 WWW 浏览器(　　)。

A. 请求地址解析　B. 建立传输连接　C. 请求域名解析　D. 建立会话连接

(26) 什么是动态主机配置协议? 它的作用是什么?

(27) 对于运行在 Windows 环境下的计算机若想启用 DHCP,应如何进行配置?

(28) 请解释 DHCP 发现报文为什么不采用 TCP 而是采用 UDP 进行传输?

(29) DHCP 提供了哪些报文? 它们的作用是什么?

(30) 请详细论述 DHCP 为主机动态分配 IP 地址的过程。

(31) 主机如何续订 IP 地址?

(32) DHCP 客户不能发送(　　)报文。

A. DHCPDISCOVER　　B. DHCPOFFER

C. DHCPREQUEST　　D. DHCPRELEASE

(33) DHCP 服务器不能发送(　　)报文。

A. DHCPDISCOVER　　B. DHCPACK

C. DHCPNACK　　D. DHCPRELEASE

(34) (　　)报文向客户提供 IP 地址。

A. DHCPNACK　　B. DHCPOFFER

C. DHCPREQUEST　　D. DHCPRELEASE

(35) 下列协议中,(　　)调用 UDP。

A. SMTP　　B. TFTP　　C. HTTP　　D. DHCP