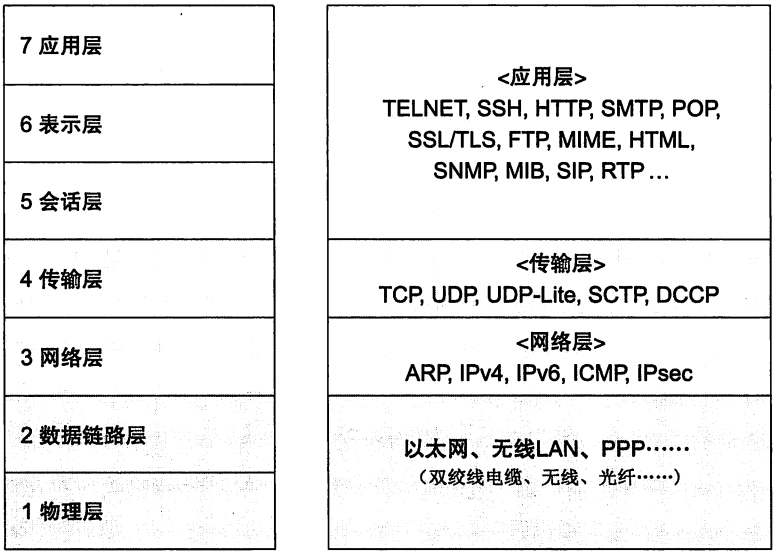
第2章

TCP/IP基础知识

TCP (Transmission Control Protocol) 和IP (Internet Protocol) 是互联网的众多通信协议中最为著名的。本章旨在介绍TCP/IP的发展历程及其相关协议的概况。



2.1 TCP/IP出现的背景及其历史

目前，在计算机网络领域中，TCP/IP协议可谓名气最大、使用范围最广。那么TCP/IP是如何在短时间内获得如此广泛普及的呢？有人认为是个人电脑的操作系统如Windows和Mac OS支持了TCP/IP所致。虽然这么说有一定的道理，但还不能算作TCP/IP普及的根本原因。其实，在当时围绕着整个计算机产业，全社会形成了一股支持TCP/IP的流行趋势，使得各家计算机厂商也不得不适应这种变化，不断生产支持TCP/IP的产品。现在，你在市面上几乎找不到一款不支持TCP/IP的操作系统。

那么，当时的计算机厂商又为何跟随潮流支持TCP/IP呢？要了解这个问题，我们不妨追溯一下互联网的发展历史。

2.1.1 从军用技术的应用谈起

20世纪60年代，很多大学和研究机构都开始着力于新的通信技术。其中有一家以美国国防部（DoD，The Department of Defense）为中心的组织也展开了类似的研究。

DoD认为研发新的通信技术对于国防军事有着举足轻重的作用。该组织希望在通信传输的过程中，即使遭到了敌方的攻击和破坏，也可以经过迂回线路实现最终通信，保证通信不中断。如图2.1所示，倘若在中心位置的中央节点遇到攻击，就会影响整个网络的通信传输。然而，图2.2中网络呈现出由众多迂回线路组成的分布式通信，使其即便在某一处受到通信攻击，也会在迂回线路的极限范围内始终保持通信无阻▼。为了实现这种类型的网络，分组交换技术便应运而生。

▼分布式网络的概念于1960年由美国RAND研究所的Paul Baran提出。

人们之所以开始关注分组交换技术，不仅是因为它在军工防卫方面的应用，还在于这种技术本身的一些特征。它可以使多个用户同一时间共享一条通信线路进行通信，从而提高了线路的使用效率，也降低了搭建线路的成本。▼

▼通过分组交换技术实现的分组通信，是在1965年由英国NPL（英国国家物理实验室）的Donald Davies提出。

到了20世纪60年代后半叶，已有大量研究人员投身于分组交换技术和分组通信的研究。

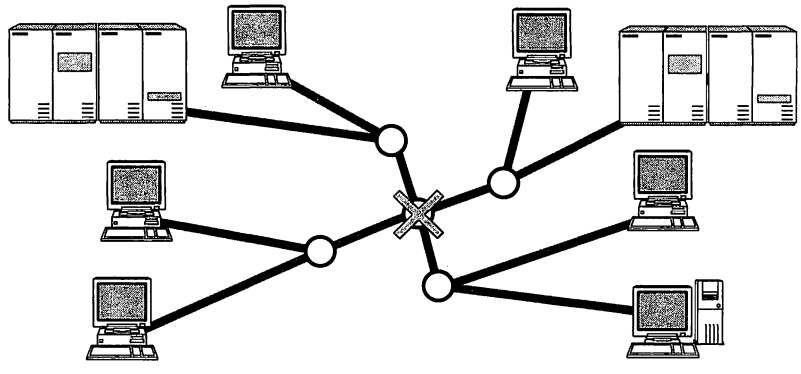


图2.1 容灾性较弱的中央集中式网络

当中心节点发生故障时，绝大多数通信都会受到影响。

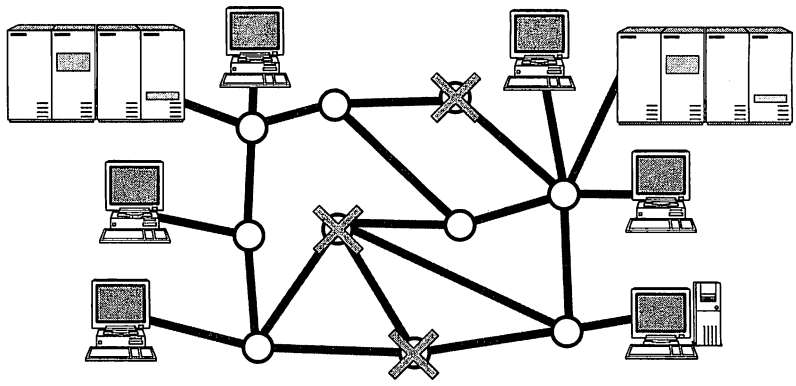


图 2.2
容灾性较强的分组网络

即使在几个节点上发生故障，通过迂回线路仍然能保持分组数据的传输。

2.1.2 ARPANET 的诞生

1969 年，为验证分组交换技术的实用性，研究人员搭建了一套网络。起初，该网络只连接了美国西海岸的大学和研究所等 4 个节点▼。之后，随着美国国防部的重点开发和相关技术的飞速发展，普通用户也逐渐加入其中，发展成了后来巨大规模的网络。

▼这四个节点分别是 UCLA（加州大学洛杉矶分校）、UCSB（加州大学圣巴巴拉分校）、SRI（斯坦福研究所）和犹他州大学。

该网络被人们称作 ARPANET▼，也是全球互联网的鼻祖。在短短 3 年内，ARPANET 从曾经的 4 个节点迅速发展成为 34 个节点的超大网络。研究人员的实验也获得了前所未有的成功▼，并以此充分证明了基于分组交换技术的通信方法是可行性。

▼ Advanced Research Projects Agency Network，阿帕网。

▼阿帕网的实验及其协议的开发，是由美国一个叫做 DARPA（Defense Advanced Research Projects Agency：国防部高级研究计划署）的政府机构资助的。

2.1.3 TCP/IP 的诞生

ARPANET 的实验，不仅仅是利用几所大学与研究机构组成的主干网络进行分组交换的实验，还会进行在互连计算机之间提供可靠传输的综合性通信协议的实验。于是在 20 世纪 70 年代前半叶，ARPANET 中的一个研究机构研发出了 TCP/IP。在这之后，直到 1982 年，TCP/IP 的具体规范才被最终定下来，并于 1983 年成为 ARPANET 网络唯一指定的协议。

表 2.1
TCP/IP 的发展

年　份	事　件
20 世纪 60 年代后半叶	应 DoD 要求，美国开始进行通信技术相关的研发。
1969 年	ARPANET 诞生。开发分组交换技术。
1972 年	ARPANET 取得初步成功。扩展到 50 个节点。
1975 年	TCP/IP 诞生。
1982 年	TCP/IP 规范出炉。UNIX 是最早开始实现 TCP/IP 的协议。
1983 年	ARPANET 决定正式启用 TCP/IP 为通信协议。
1989 年左右	局域网上的 TCP/IP 应用迅速扩大。
1990 年左右	不论是局域网还是广域网，都开始倾向于使用 TCP/IP。
1995 年左右	互联网开始商用，互联网服务供应商的数量剧增。
1996 年	IPv6 规范出炉，载入 RFC。（后于 1998 年修订）

2.1.4　UNIX 系统的普及与互联网的扩张

TCP/IP 的产生，ARPANET 起到了举足轻重的作用。然而，ARPANET 网络组成之初，由于其节点个数的限制，TCP/IP 的应用范围也受到一定的限制。那么，TCP/IP 后来又是如何在计算机网络中得到如此广泛普及的呢?

1980 年左右，ARPANET 中的很多大学与研究机构开始使用一种叫做 BSD UNIX 的操作系统。由于 BSD UNIX▼ 实现了 TCP/IP 协议，所以很快在 1983 年，TCP/IP 便被 ARPANET 正式采用。同年，前 SUN 公司也开始向一般用户提供实现了 TCP/IP 的产品。

▼ BSD UNIX：由美国加州大学伯克利分校开发的免费的 UNIX 系统。

20 世纪 80 年代不仅是局域网快速发展的时代，还是 UNIX 工作站迅速普及的时代，同时也是通过 TCP/IP 构建网络最为盛行的时代。基于这些趋势，那些大学和研究机构也逐渐开始将 ARPANET 连接到了 NSFnet 网络。此后，基于 TCP/IP 而形成的世界性范围的网络——互联网（The Internet）便诞生了。

以连接 UNIX 主机的形式连接各个终端节点，这一主要方式使互联网得到了迅速的普及。而作为计算机网络主流协议的 TCP/IP，它的发展也与 UNIX 密不可分。到了 80 年代后半叶，那些“各自为政”开发自己通信协议的网路设备供应商们，也陆续开始“顺从”于 TCP/IP 的规范，制造兼容性更好的产品以便用户使用。

2.1.5　商用互联网服务的启蒙

研发互联网最初的目的是用于实验和研究，到了 1990 年逐渐被引入公司企业及一般家庭。也出现了专门提供互联网接入服务的公司（称作 ISP▼），这些都使互联网得到了更为广泛的普及。同时，基于互联网技术的新型应用，如在线游戏、SNS、视频通信等商用服务也如雨后春笋般不断涌现出来。

▼ Internet Service Provider，为个人、公司或教育机构等提供互联网接入服务的供应商。

于是，人们对拨号（当时个人电脑通信▼通过拨号实现）上网的要求越来越高，希望每两个人之间也都能够通过计算机实现通信。然而，个人电脑通信只能为有限的用户提供服务，而且多台电脑加入通信时操作方法又不相同，这给人们带来了一定的不便。

▼ 1980 年后半叶广为普及的一种网络服务。在这种通信中个人电脑通过电话线和调制解调器（Modem）与主机连接，可以使用电子邮件、公告板等服务。

于是，面向公司企业和一般家庭提供专门互联网接入服务的具有商用许可▼的提供商（ISP）便出现了。这时，由于 TCP/IP 已长期应用于研究领域，使人们积累了丰富的经验，因此，面对这样一种成熟的技术，人们对于它的商用价值充满期待。

▼ NSFnet 层被禁止商用。

连接到互联网，人们可以从 WWW 获取世界各处的信息，可以通过电子邮件进行交流，还可以向全世界发布自己的消息。互联网中没有所谓会员的限制，它是一个连接全世界的公共网络。互联网使人们的生活变得更加多姿多彩，人们不仅可以享受多姿多彩的服务，还可以通过互联网自己开创新的服务。

互联网作为一种商用服务迅速发展起来。这使得到 90 年代为止一直占据主导地位的个人电脑通信也开始加入到互联网的行列中来，自由的、开放的互联网就这样以极快的速度为大众所认可，得到更为广泛的普及。

2.2 TCP/IP 的标准化

20 世纪 90 年代，ISO 开展了 OSI 这一国际标准协议的标准化进程。然而，OSI 协议并没有得到普及，真正被广泛使用的是 TCP/IP 协议。

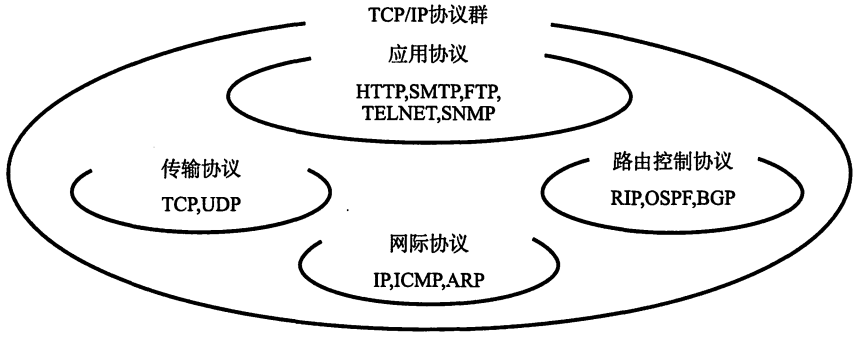
究其原因，是由 TCP/IP 的标准化所致。TCP/IP 的标准化中有其他协议的标准化没有的要求。这一点就是让 TCP/IP 更迅速地实现和普及的原动力。本节将介绍 TCP/IP 的标准化过程。

2.2.1 TCP/IP 的具体含义

从字面意义上讲，有人可能会认为 TCP/IP 是指 TCP 与 IP 两种协议。实际生活当中有时也确实就是指这两种协议。然而在很多情况下，它只是利用 IP 进行通信时所必须用到的协议群的统称。具体来说，IP 或 ICMP、TCP 或 UDP、TELNET 或 FTP、以及 HTTP 等都属于 TCP/IP 的协议。它们与 TCP 或 IP 的关系紧密，是互联网必不可少的组成部分。TCP/IP 一词泛指这些协议，因此，有时也称 TCP/IP 为网际协议族▼。

▼网际协议族（Internet Protocol Suite）：组成网际协议的一组协议。

图 2.3
TCP/IP 协议群



2.2.2 TCP/IP 标准化精髓

TCP/IP 的协议的标准化过程与其他的标准化过程有所不同，具有两大特点：一是具有开放性，二是注重实用性，即被标准化的协议能否被实际运用。

首先，开放性是由于 TCP/IP 的协议是由 IETF 讨论制定的，而 IETF 本身就是一个允许任何人加入进行讨论的组织。在这里人们通常采用电子邮件组的形式进行日常讨论，而邮件组可以由任何人随时订阅。

其次，在 TCP/IP 的标准化过程中，制订某一协议的规范本身已不再那么重要，而首要任务是实现真正能够实现通信的技术。难怪有人打趣到“TCP/IP 简直就是先开发程序，后写规格标准”。

虽然这么说有点夸张，不过 TCP/IP 在制定某个协议规范的过程中确实会考虑到这个协议实现▼的可行性。而且在某个协议的最终详细规范出炉的同时，其中一些协议已在某些设备中存在，并且能够进行通信。

▼实现：指开发那些能够让计算机设备按照协议预期产生某些动作或行为的程序和硬件。

为此，TCP/IP 中只要某个协议的大致规范决定下来，人们就会在多个已实现该协议的设备之间进行通信实验，一旦发现有什么问题，可以继续在 IEFT 中讨论，及时修改程序、协议或相应的文档。经过这样一次又一次的讨论、实验和研

究，一款协议的规范才会最终诞生。因此，TCP/IP协议始终具有很强的实用性。

然而，对于那些由于实验环境的限制没有发现问题的协议，将会在后期继续进行改进。相比TCP/IP，OSI之所以未能达到普及，主要原因在于未能尽早地制定可行性较强的协议、未能提出应对技术快速革新的协议以及没有能及时进行后期改良的方案这几点。

2.2.3 TCP/IP规范——RFC

前面提到TCP/IP的协议由IETF讨论制定。那些需要标准化的协议，被人们列入RFC（Request For Comment）▼文档并在互联网上公布。RFC不仅记录了协议规范内容，还包含了协议的实现和运用的相关信息▼，以及实验方面的信息▼。

▼RFC从字面意义上看就是指征求意见表，属于一种征求协议相关意见的文档。

▼协议实现或运用相关的信息叫做FYI（For Your Information）。

▼实验阶段的协议称作Experimental。

RFC文档通过编号组织每个协议的标准化请求。例如IP协议的规范由RFC279制定，TCP协议的规范由RFC793号文档决定。RFC的编码是既定的，一旦成为某一RFC的内容，就不能再对其进行随意修改。若要扩展已有某个协议规范的内容，一定要有一个全新编号的RFC文档对其进行记录。若要修改已有某个协议规范内容，则需要重新发行一个新的RFC文档，同时，老的那份RFC作废。新的RFC文档会明确规定是扩展了哪个已有的RFC以及要作废哪个已有RFC。

此时，有人提出每当对RFC进行修改时都要产生新的RFC编号太麻烦。为此，人们采用STD（Standard）方式管理编号。STD用来记载哪个编号制定哪个协议。因此，同一个协议的规范内容即便发生了变化也不会导致STD编号发生变化。

▼例如STD5表示包含ICMP的IP协议标准。因此，STD5由RFC791、RFC919、RFC922、RFC792、RFC950以及RFC1112 6个RFC组成。

今后，即使协议规范的内容改变也不会改变STD编号，但是有可能导致某个STD下的RFC编号视情况有所增减。

此外，为了向互联网用户和管理者提供更有益的信息，与STD类似，FYI（For Your Information）也开始标注编号组织。FYI为了人们方便检索，也在其每个编号里涵盖了所涉及的RFC编号。即使更新内容，编号也不会发生变化。

STD1记录着所有要求协议标准化的RFC状态。到2012年1月为止，STD1相当于RFC5000（很多情况下会采用比较容易记忆的编号）。

表2.2　具有代表性的RFC（2012年1月为止）

协　议	STD	RFC	状　态
IP（v4）	STD5	RFC 791、RFC919、RFC922	标准
IP（v6）		RFC2460	草案标准
ICMP	STD5	RFC792、RFC950	标准
ICMPv6		RFC4443	草案标准
ND for IPv6▼		RFC4861	草案标准
ARP	STD37	RFC826	标准
RARP	STD38	RFC903	标准
TCP	STD7	RFC793、RFC3168	标准
UDP	STD6	RFC768	标准
IGMP（v3）		RFC3376	提议标准
DNS	STD13	RFC1034、RFC1035	标准

▼Neighbor Discovery Protocol for Internet Protocol Version 6

（续）

协　议	STD	RFC	状　态
DHCP		RFC2131、RFC2132、RFC3315	草案标准
HTTP（v1.1）		RFC2616	草案标准
SMTP		RFC5321	草案标准
	STD10	RFC821、RFC1869、RFC1870	标准
POP（v3）	STD53	RFC1939	标准
FTP	STD9	RFC959、RFC2228	标准
TELNET	STD8	RFC854、RFC855	标准
SNMP	STD15	RFC1157	历史性
SNMP（v3）	STD62	RFC3411、RFC3418	标准
MIB-II	STD17	RFC1213	标准
RMON	STD59	RFC2819	标准
RIP（v2）	STD34	RFC1058	历史性
RIP（v2）	STD56	RFC2453	标准
OSPF（v2）	STD54	RFC2328	标准
EGP	STD18	RFC904	历史性
BGP（v4）		RFC4271	草案标准
PPP	STD51	RFC1661、RFC1662	标准
PPPoE		RFC2516	信息性
MPLS		RFC3031	提议标准
RTP	STD64	RFC3550	标准
主机实现要求	STD3	RFC1122、RFC1123	标准
路由器实现要求		RFC1812、RFC2644	提议标准

每个 RFC 的最新信息请参考 http：//www. rfc-editor. org/rfc/rfc××××. txt（其中××××为 RFC 编号）。

新的 RFC 与旧的 RFC

下面，以第 4 章要介绍的 ICMP 为例来介绍一下 RFC 的变迁过程。

ICMP 是由 RFC792 定义、由 RFC950 扩展的。也就是说，ICMP 是由这两个 RFC 文档组合起来构成其详细的规范内容。RFC792 本身废除了以前的 RFC777。而 RFC1256 虽然还未正式成为标准，但目前（到 2012 年 2 月为止）已处于提议标准阶段。

主机和路由器处理 ICMP 时所涉及的要求细节也写入了 RFC，分别为 RFC1122 和 RFC1812▼。

▼ RFC1122 与 RFC1812 中不仅记载了对 ICMP 的处理要求，还记载了主机和路由器对 IP、TCP 以及 ARP 等众多协议在实现上的要求。

2.2.4　TCP/IP的标准化流程

一个协议的标准化一定要经过IETF讨论。IETF虽然每年只组织3次会议，但是日常都会通过邮件组的形式进行讨论，并且该邮件组不限制订阅。

TCP/IP协议的标准化流程大致分为以下几个阶段：首先是互联网草案阶段；其次，如果认为可以进行标准化，就记入RFC进入提议标准阶段；第三，是草案标准阶段；最后，才进入真正的标准阶段。

如果仔细分析这些阶段，不难发现在协议真正被标准化之前会有一个提议阶段。正是在这一阶段，那些想要对协议提出建议和意见的个人或组织会撰写文档，将内容作为草案发布在互联网上，而讨论也将基于这些文档内容通过邮件进行，从而也可以进行相应的设备实现、模拟以及应用实验。

互联网草案的有效期通常为6个月。也就是说，只要进入讨论流程，就必须在6个月内将所讨论的结果反映到的草案，否则将以长时间无任何进展为由自动消除。这也是为了防止一些没有实质意义和实际讨论内容的草案出现。在这个全世界信息泛滥的时代，TCP/IP的草案也是漫天横飞。因此，去伪存真是非常重要的。

经过充分的讨论，如果得到IESG（IETF Engineering Steering Group，由IETF的主要成员组成）的批准，就能被编入RFC文档。这个文档叫做提议标准（Proposed Standard）。

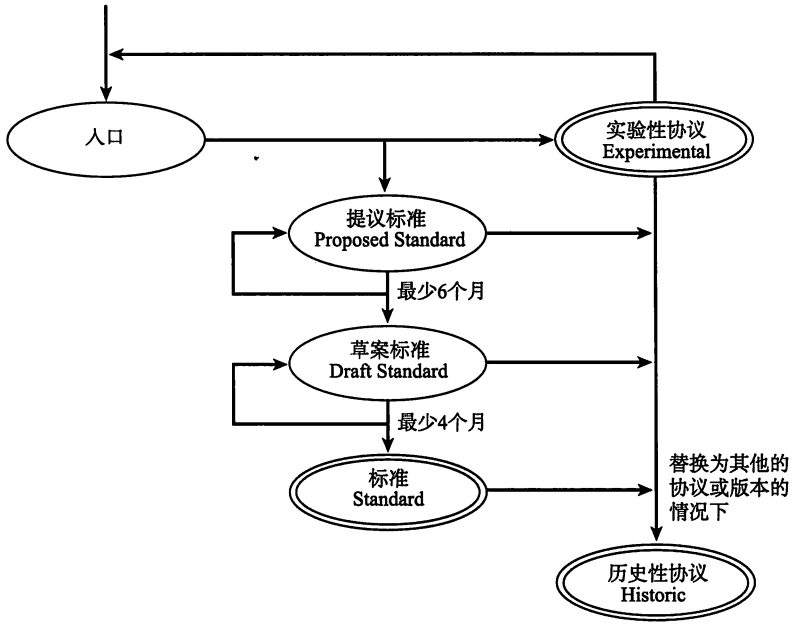
提议标准中所提出的协议将被众多设备应用。如果能够得到IESG的认可，就可以成为草案标准（Draft Standard）。而如果在实际应用当中遇到问题，则可在成为草案标准前进行修订。当然，这种修订也是通过互联网草案的形式发布的。

要从草案标准达到真正的标准，还需要更多的设备实现并应用这个特定的协议。若所有参与该协议制定的人都觉得它“实用性强，没有什么问题”，并得到IESG的最终批准，那么这个草案标准就可以成为标准。

因此标准化的过程是漫长而有风险的。如果未在互联网上被广泛使用，就无法最终成为一个提案标准。TCP/IP的标准化过程与一般的标准化过程不同。它不是由标准化组织制定为标准以后才开始投入应用，而是到其成为标准的那一刻为止，已经被较为充分地试验并得到了较广的普及▼。那些已经成为标准的TCP/IP协议其实早已被人们广泛应用，因此，具有很强的实用性。

▼有些协议不是以标准化为目的，而只是实验性质的。这种协议在RFC中被称作实验性协议（Experimental）。

图 2.4
协议的标准化流程



提议标准与草案标准的实现

很多情况下，向市场推广一些只实现了 RFC 中标准协议的产品显然不够，因为只有被广泛使用之后才能成为标准。

因此从前瞻性考虑，应该实现那些草案协议和提议协议，这样才可能有机会抢先市场。并且，当规范经过修订以后，设备厂商也应该提供升级等方式将其迅速反映到产品当中。

2.2.5 RFC 的获取方法

获取 RFC 可以有几种方法。最直接的方法就是利用互联网查询“RFC Editor”（所有的 RFC 都在“RFC Editor”中管理）。具体网址为：

```
http: //www.rfc-editor.org/rfc/
ftp: //ftp.rfc-editor.org/in-notes/
```

上面两个网址保存着所有 RFC 文件，网站中有一个名为 rfc-index.txt 的文件包含了所有 RFC 的概览。RFC 网站除了发布 RFC 的相关信息，还提供 RFC 检索功能。此外，在日本国内的某些 anonymous ftp 服务器▼（如 JPNIC 的 ftp 服务器，ftp: //ftp.nic.ad.jp/rfc/）上也存有 RFC 信息。

▼互联网上有很多这样的 ftp 服务器。它的特点是可以由任何人用匿名用户访问。

■ 如何获取 STD 或 FYI 以及 ID

STD、FYI、ID（I-D：Internet Draft）号可以从以下网站获取。关于它们的概览也分别记录在 std-index. txt、fyi-index. txt 等文件中。因此可以先从这些网站搜索对应的编号。

- STD 获取网址
  http：//www. rfc-editor. org/in-notes/std/
- FYI 获取网址
  http：//www. rfc-editor. org/in-notes/fyi/
- ID 获取网址
  http：//www. rfc-editor. org/internet-drafts/

JPNIC 的 ftp 服务器中的目录：

- STD 获取网址
  ftp：//ftp. nic. ad. jp/rfc/std/
- FYI 获取网址
  ftp：//ftp. nic. ad. jp/rfc/fyi/
- ID 获取网址
  ftp：//ftp. nic. ad. jp/internet-drafts/

2.3 互联网基础知识

“互联网”一词家喻户晓，本书也曾多次提到过。那么互联网究竟是什么？它与 TCP/IP 之间又有什么关系？本节就互联网以及互联网与 TCP/IP 之间不可分割的关系做一些简单介绍。

2.3.1 互联网定义

“互联网”，英文单词为“Internet”。从字面上理解，internet 指的是将多个网络连接使其构成一个更大的网络，所以 internet 一词本意为网际网。将两个以太网网段用路由器相连是互联网，将企业内部各部门的网络或公司的内网与其他企业相连接，并实现相互通信的网络也是互联网，甚至一个区域的网络与另一个区域的网络相互连接形成全世界规模的网络也可以称作互联网。然而，现在“互联网”这个词的意思却有所变化。当专门指代网络之间的连接时，可以使用“网际网”这个词。

“互联网”是指由 ARPANET 发展而来、互连全世界的计算机网络。现在，“互联网”已经是一个专有名词了，其对应的英文单词“The Internet”也早已成为固有名词（Internet 指网际网，The Internet 指互联网，首字母大写）▼。

▼与 Internet 对应的另一种网络叫做 Intranet。该网络是指使用 Internet 技术将企业内部的组织机构连接起来形成一个企业范围内的封闭网络，提供面向企业内部的通信服务。

2.3.2 互联网与 TCP/IP 的关系

互联网进行通信时，需要相应的网络协议，TCP/IP 原本就是为使用互联网而开发制定的协议族。因此，互联网的协议就是 TCP/IP，TCP/IP 就是互联网的协议。

2.3.3 互联网的结构

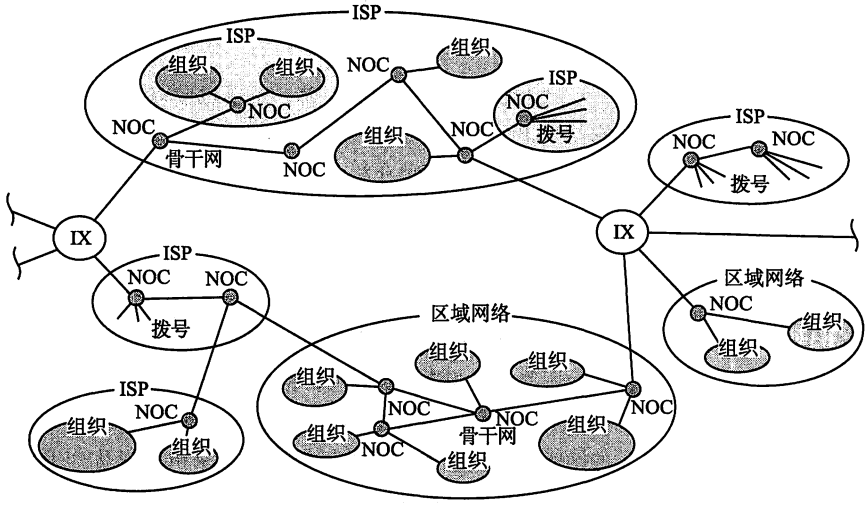
如 2.3.1 节中提到，互联网一词原意是网际网，意指连接一个又一个网络。那么连接全世界的互联网也是如此。较小范围的网络之间相连组成机构内部的网络，机构内部的网络之间相连再形成区域网络，而各个区域网络之间再互连，最终就形成了连接全世界的互联网。互联网就是按照这样的形式构成了一个有层次的网络。

互联网中的每个网络都是由骨干网（BackBone）和末端网（Stub）组成的。每个网络之间通过 NOC▼ 相连。如果网络的运营商不同，它的网络连接方式和使用方法也会不同。连接这种异构网络需要有 IX▼ 的支持。总之，互联网就是众多异构的网络通过 IX 互连的一个巨型网络。

▼ Network Operation Center，网络操作中心。

▼ Internet Exchange，网络交换中心。

图2.5
互联网的结构



ISP: Internet Service Provider
IX: Internet Exchange
NOC: Network Operation Center

2.3.4 ISP和区域网

连接互联网需要向ISP或区域网提出申请。公司企业或一般家庭申请入网只要联系ISP签约即可。

不同的ISP所提供的互联网接入服务的项目也不同。例如，不限流量包月、限定上网时限以及有线/无线网络连接等各种各样的服务。

区域网指的是在特定区域内由团体或志愿者所运营的网络。这种方式通常价格比较便宜，但是有时可能会出现连接方式复杂或使用上有限制等情况。

所以人们在实际申请连网前，最好先确认一下ISP或区域网所对应的具体服务条目、所提供服务的细则（如接入方式、条件、费用等）等，然后再结合自己的使用目的做决定。

互联网内外

当公司的网络与家里的个人电脑都能连网时，一方面可以认为它们都是互联网的一部分（如图2.6），另一方面，从公司的局域网或家里个人电脑的角度出发，可以认为它们连接的目标网络都是互联网。这种透视方法其实就是在将提供网络的ISP看作是外在、将内外明确划分的一种方法（如图2.7）▼。

▼实际上有些公司会将互联网看作外在，并对与其连接的设备或协议进行限制。

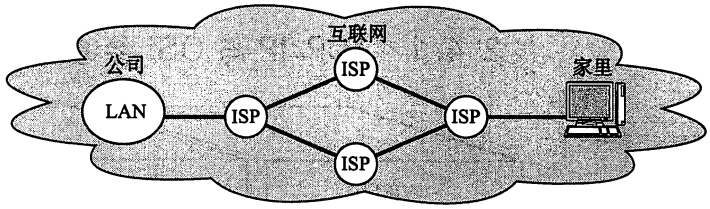


图2.6 将公司网络与家里个人电脑看作互联网一部分的方法

图2.7 将互连的对端看作互联网的方法

2.4 TCP/IP 协议分层模型

TCP/IP 是当今计算机网络界使用最为广泛的协议。TCP/IP 的知识对于那些想构筑网络、搭建网络以及管理网络、设计和制造网络设备甚至是做网络设备编程的人来说都是至关重要的。那么，TCP/IP 究竟是什么呢？本节就 TCP/IP 协议做一个简单地介绍。

2.4.1 TCP/IP 与 OSI 参考模型

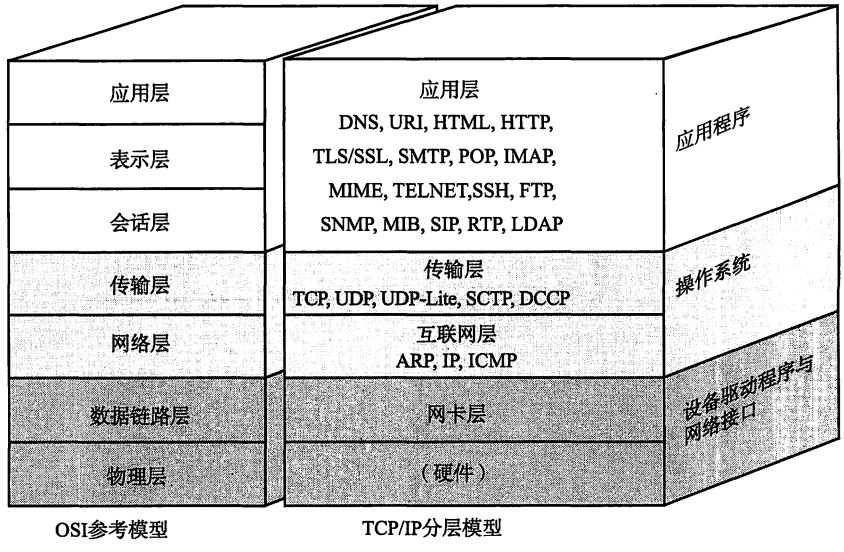


图2.8 OSI 参考模型与 TCP/IP 的关系

第 1 章我们介绍了 OSI 参考模型中各个分层的作用。TCP/IP 诞生以来的各种协议其实也能对应到 OSI 参考模型当中。如果了解了这些协议分属 OSI 的哪一层，就能对该协议的目的有所了解。然后对于每个协议的具体技术要求就可以参考相应的规范了。在此，暂时略过协议本身的细节（第 4 章以后详解），先介绍一下各个协议与 OSI 参考模型中各个分层之间的对应关系。

图 2.8 列出了 TCP/IP 与 OSI 分层之间的大致关系。不难看出，TCP/IP 与 OSI 在分层模块上稍有区别。OSI 参考模型注重“通信协议必要的功能是什么”，而 TCP/IP 则更强调“在计算机上实现协议应该开发哪种程序”。

2.4.2 硬件（物理层）

TCP/IP 的最底层是负责数据传输的硬件。这种硬件就相当于以太网或电话线路等物理层的设备。关于它的内容一直无法统一定义。因为只要人们在物理层面上所使用的传输媒介不同（如使用网线或无线），网络的带宽、可靠性、安全性、延迟等都会有所不同，而在这些方面又没有一个既定的指标。总之，TCP/IP 是在网络互连的设备之间能够通信的前提下才被提出的协议。

2.4.3 网络接口层（数据链路层）

▼有时人们也将网络接口层与硬件层合并起来称作网络通信层。

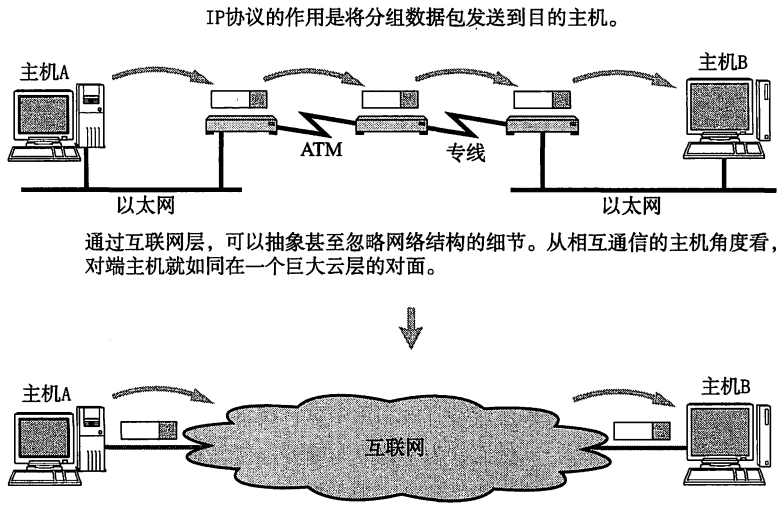
网络接口层▼利用以太网中的数据链路层进行通信，因此属于接口层。也就是说，把它当做让 NIC 起作用的“驱动程序”也无妨。驱动程序是在操作系统与硬件之间起桥梁作用的软件。计算机的外围附加设备或扩展卡，不是直接插到电脑上或电脑的扩展槽上就能马上使用的，还需要有相应驱动程序的支持。例如换了一个新的 NIC 网卡，不仅需要硬件，还需要软件才能真正投入使用。因此，人们常常还需要在操作系统的基础上安装一些驱动软件以便使用这些附加硬件▼。

▼现在也有很多是即插即拔的设备，那是因为计算机的操作系统中早已经内置安装好了对应网卡的驱动程序，而并非不需驱动。

2.4.4 互联网层（网络层）

互联网层使用 IP 协议，它相当于 OSI 模型中的第 3 层网络层。IP 协议基于 IP 地址转发分包数据。

图 2.9 互联网层



TCP/IP 分层中的互联网层与传输层的功能通常由操作系统提供。尤其是路由器，它必须得实现通过互联网层转发分组数据包的功能。

此外，连接互联网的所有主机跟路由器必须都实现 IP 的功能。其他连接互联网的网络设备（如网桥、中继器或集线器）就没必要一定实现 IP 或 TCP 的功能▼。

▼有时为了监控和管理网桥、中继器、集线器等设备，也需要让它们具备 IP、TCP 的功能。

IP

IP 是跨越网络传送数据包，使整个互联网都能收到数据的协议。IP 协议使数据能够发送到地球的另一端，这期间它使用 IP 地址作为主机的标识▼。

▼连接 IP 网络的所有设备必须有自己唯一的识别号以便识别具体的设备。分组数据在 IP 地址的基础上被发送到对端。

IP 还隐含着数据链路层的功能。通过 IP，相互通信的主机之间不论经过怎样的底层数据链路都能够实现通信。

虽然 IP 也是分组交换的一种协议，但是它不具有重发机制。即使分组数据包未能到达对端主机也不会重发。因此，属于非可靠性传输协议。

ICMP

IP 数据包在发送途中一旦发生异常导致无法到达对端目标地址时，需要给发

送端发送一个发生异常的通知。ICMP就是为这一功能而制定的。它有时也被用来诊断网络的健康状况。

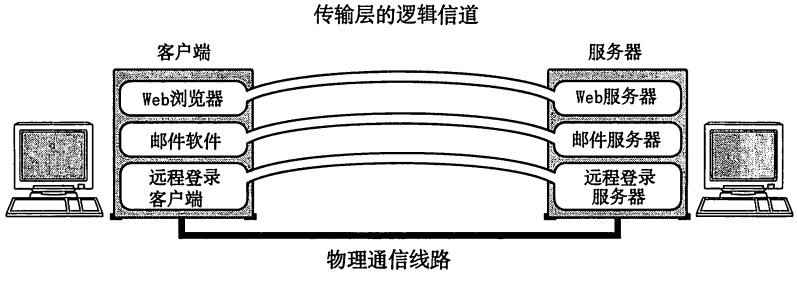
ARP

从分组数据包的IP地址中解析出物理地址（MAC地址）的一种协议。

2.4.5　传输层

TCP/IP的传输层有两个具有代表性的协议。该层的功能本身与OSI参考模型中的传输层类似。

图2.10
传输层



传输层最主要的功能就是能够让应用程序之间实现通信。计算机内部，通常同一时间运行着多个程序。为此，必须分清是哪些程序与哪些程序在进行通信。识别这些应用程序的是端口号。

TCP

TCP是一种面向有连接的传输层协议。它可以保证两端通信主机之间的通信可达。TCP能够正确处理在传输过程中丢包、传输顺序乱掉等异常情况。此外，TCP还能够有效利用带宽，缓解网络拥堵。

然而，为了建立与断开连接，有时它需要至少7次的发包收包，导致网络流量的浪费。此外，为了提高网络的利用率，TCP协议中定义了各种各样复杂的规范，因此不利于视频会议（音频、视频的数据量既定）等场合使用。

UDP

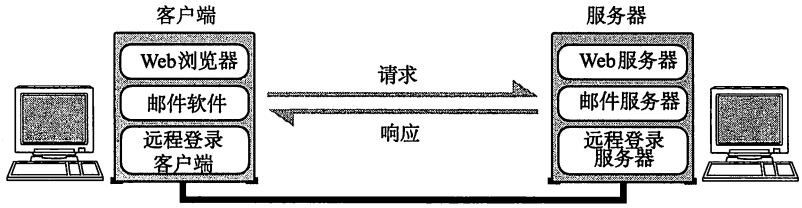
UDP有别于TCP，它是一种面向无连接的传输层协议。UDP不会关注对端是否真的收到了传送过去的数据，如果需要检查对端是否收到分组数据包，或者对端是否连接到网络，则需要在应用程序中实现。

UDP常用于分组数据较少或多播、广播通信以及视频通信等多媒体领域。

2.4.6　应用层（会话层以上的分层）

TCP/IP的分层中，将OSI参考模型中的会话层、表示层和应用层的功能都集中到了应用程序中实现。这些功能有时由一个单一的程序实现，有时也可能会由多个程序实现。因此，细看TCP/IP的应用程序功能会发现，它不仅实现OSI模型中应用层的内容，还要实现会话层与表示层的功能。

图 2.11
客户端/服务端模型

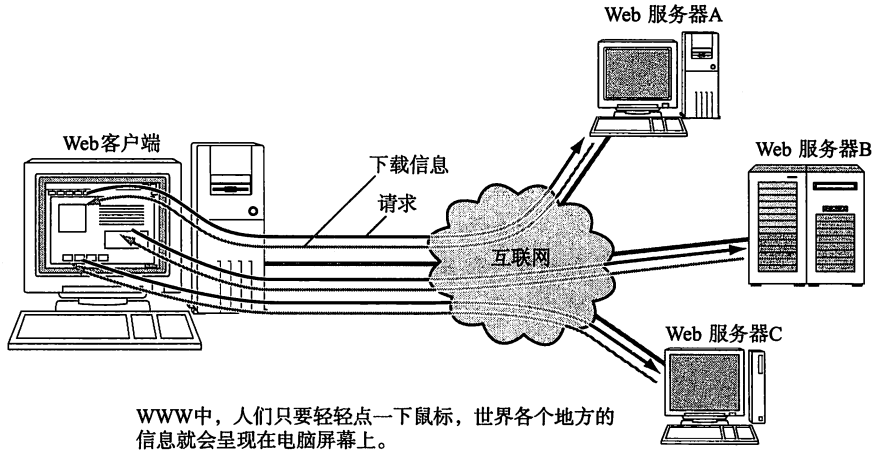


TCP/IP 应用的架构绝大多数属于客户端/服务端模型。提供服务的程序叫服务端，接受服务的程序叫客户端。在这种通信模式中，提供服务的程序会预先被部署到主机上，等待接收任何时刻客户可能发送的请求。

客户端可以随时发送请求给服务端。有时服务端可能会有处理异常[①]、超出负载等情况，这时客户端可以在等待片刻后重发一次请求。

WWW

图 2.12
WWW



WWW中，人们只要轻轻点一下鼠标，世界各个地方的信息就会呈现在电脑屏幕上。

WWW▼ 可以说是互联网能够如此普及的一个重要原动力。用户在一种叫 Web 浏览器▼的软件上借助鼠标和键盘就可以轻轻松松地在网上自由地冲浪。也就是说轻按一下鼠标架设在远端服务器上的各种信息就会呈现到浏览器上。浏览器中既可以显示文字、图片、动画等信息，还能播放声音以及运行程序。

浏览器与服务端之间通信所用的协议是 HTTP（HyperText Transfer Protocol）。所传输数据的主要格式是 HTML（HyperText Markup Language）。WWW 中的 HTTP 属于 OSI 应用层的协议，而 HTML 属于表示层的协议。

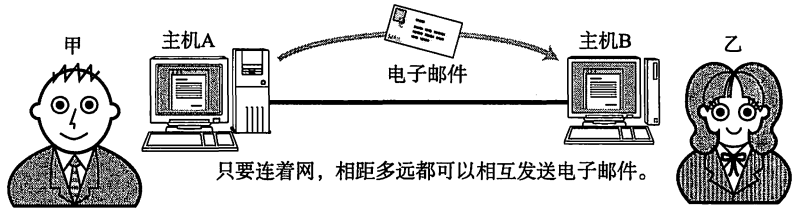
▼中文叫万维网，是一种互联网上数据读取的规范。有时也叫做 Web、WWW 或 W3。

▼通常可以简化称作浏览器。微软公司的 Internet Explore 以及 Mozilla Foundation 的 Firefox 等都属于浏览器。它们已被人们广泛使用。

① 当然，如果是整个服务器宕掉，或者服务端容器宕掉，那就只有等待充分恢复之后才能继续处理客户端请求。——译者注

电子邮件（E-Mail）

图 2.13
电子邮件



电子邮件其实就是指在网络上发送信件。有了电子邮件，不管距离多远的人，只要连着互联网就可以相互发送邮件。发送电子邮件时用到的协议叫做 SMTP（Simple Mail Tranfer Protocol）。

最初，人们只能发送文本格式▼的电子邮件。然而现在，电子邮件的格式由 MIME▼ 协议扩展以后，就可以发送声音、图像等各式各样的信息。甚至还可以修改邮件文字的大小、颜色▼。这里提到的 MIME 属于 OSI 参考模型的第 6 层——表示层。

▼只由文字组成的信息。日语最初只能发送 7bit JIS 编码的文字。

▼在互联网上广泛使用的、用来定义邮件数据格式一种规范。在 WWW 与网络论坛中也可以使用。关于这一点的更多细节请参考 8.4.3 节。

▼有时某些机能可能会因为邮件接收端软件的限制不能充分展现。

电子邮件与 TCP/IP 的发展

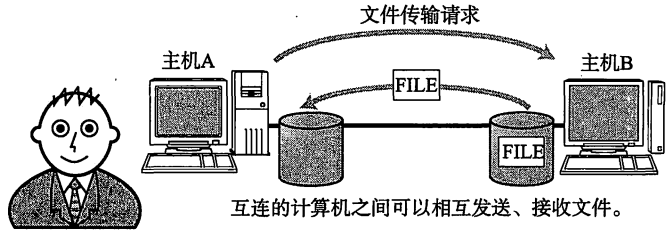
有人可能会说“TCP/IP 的发展离不开电子邮件！”这句话可能有两方面的含义。

一方面，电子邮件使用起来非常方便，便于讨论 TCP/IP 协议的进度和细节。而另一方面，为了正常使用电子邮件，需要具备完善的网络环境并对某些协议进行。

总之，电子邮件与 TCP/IP 的发展相辅相成。电子邮件协助改善协议，更加完善的协议又可以令电子邮件的形式多样化。

文件传输（FTP）

图 2.14
FTP



文件传输是指将保存在其他计算机硬盘上的文件转移到本地的硬盘上，或将本地硬盘的文件传送到其他机器硬盘上的意思。

该过程使用的协议叫做 FTP（File Transfer Prototol）。FTP 很早就已经投入使用▼，传输过程中可以选择用二进制方式还是文本方式▼。

在 FTP 中进行文件传输时会建立两个 TCP 连接，分别是发出传输请求时所要用到的控制连接与实际传输数据时所要用到的数据连接▼。

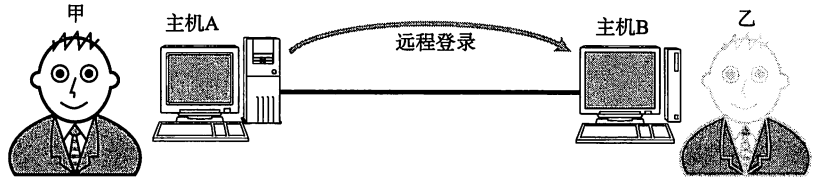
▼最近在文件传输中使用 WWW 的 HTTP 的情况也在增加。

▼用文本方式在 Windows、MacOS 或 Unix 等系统之间进行文件传输时，会自动修改换行符。这也属于表示层的功能。

▼这两种连接的控制管理属于会话层的功能。

远程登录（TELNET 与 SSH）

图 2.15
TELNET



坐在主机A前面的甲远程登录到主机B以后，就和乙一样，可以自由地操作主机B了。

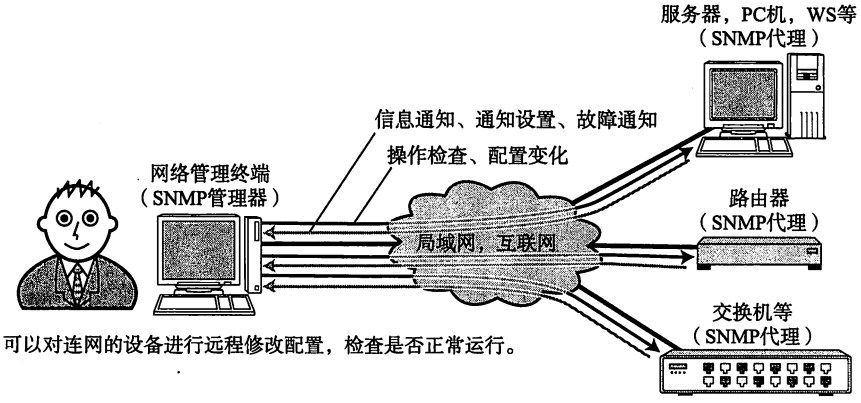
远程登录是指登录到远程的计算机上，使那台计算机上的程序得以运行的一种功能。TCP/IP 网络中远程登录常用 TELNET▼ 和 SSH▼ 两种协议。其实还有很多其他可以实现远程登录的协议，如 BSD UNIX 系中 rlogin 的 r 命令协议以及 X Window System 中的 X 协议。

▼ TELetypewriter NETwork 的缩写。有时也称作默认协议。

▼ SSH 是 Secure SHell 的缩写。

网络管理（SNMP）

图 2.16
网络管理



可以对连网的设备进行远程修改配置，检查是否正常运行。

在 TCP/IP 中进行网络管理时，采用 SNMP（Simple Network Management Protocol）协议。使用 SNMP 管理的主机、网桥、路由器等称作 SNMP 代理（Agent），而进行管理的那一段叫做管理器（Manager）。SNMP 正是这个 Manager 与 Agent 所要用到的协议。

在 SNMP 的代理端，保存着网络接口的信息、通信数据量、异常数据量以及设备温度等信息。这些信息可以通过 MIB（Management Information Base）▼ 访问。因此，在 TCP/IP 的网络管理中，SNMP 属于应用协议，MIB 属于表示层协议。

▼ MIB 也被称为是一种可透过网络的结构变量。

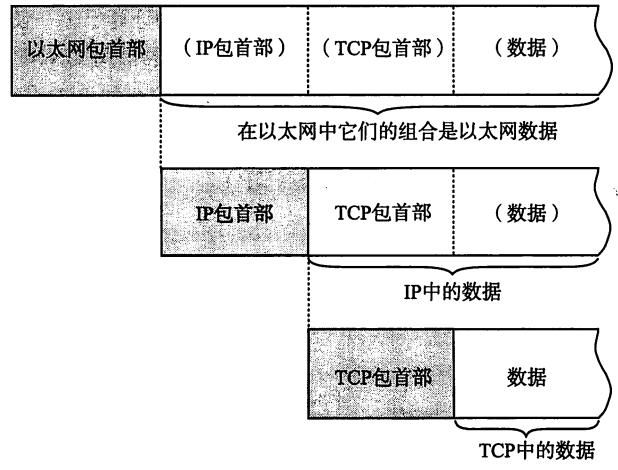
一个网络范围越大，结构越复杂，就越需要对其进行有效的管理。而 SNMP 可以让管理员及时检查网络拥堵情况，及早发现故障，也可以为以后扩大网络收集必要的信息。

2.5 TCP/IP 分层模型与通信示例

TCP/IP 是如何在媒介上进行传输的呢？本节将介绍使用 TCP/IP 时，从应用层到物理媒介为止数据处理的流程。

2.5.1 数据包首部

图 2.17
数据包首部的层次化



每个分层中，都会对所发送的数据附加一个首部，在这个首部中包含了该层必要的信息，如发送的目标地址以及协议相关信息。通常，为协议提供的信息为包首部，所要发送的内容为数据。如图 2.17，在下一层的角度看，从上一分层收到的包全部都被认为是本层的数据。

包、帧、数据报、段、消息

以上五个述语都用来表述数据的单位，大致区分如下：

包可以说是全能性述语。帧用于表示数据链路层中包的单位。而数据包是 IP 和 UDP 等网络层以上的分层中包的单位。段则表示 TCP 数据流中的信息。最后，消息是指应用协议中数据的单位。

包首部就像是协议的脸

网络中传输的数据包由两部分组成：一部分是协议所要用到的首部，另一部分是上层传过来的数据。首部的结构由协议的具体规范详细定义。例如，识别上一层协议的域应该从包的哪一位开始取多少个比特、如何计算校验和并插入包的哪一位等。相互通信的两端计算机如果在识别协议的序号以及校验和的计算方法上不一样，就根本无法实现通信。

因此，在数据包的首部，明确标明了协议应该如何读取数据。反过来说，看到首部，也就能够了解该协议必要的信息以及所要处理内容。因此，看到包首部就如同看到协议的规范。难怪有人会说首部就像是协议的脸了。

2.5.2 发送数据包

假设甲给乙发送电子邮件，内容为：“早上好”。而从 TCP/IP 通信上看，是从一台计算机 A 向另一台计算机 B 发送电子邮件。我们就通过这个例子来讲解一下 TCP/IP 通信的过程。

① 应用程序处理

启动应用程序新建邮件，将收件人邮箱填好，再由键盘输入邮件内容“早上好”，鼠标点击“发送”按钮就可以开始 TCP/IP 的通信了。

首先，应用程序中会进行编码处理。例如，日文电子邮件使用 ISO-2022-JP 或 UTF-8 进行编码。这些编码相当于 OSI 的表示层功能。

编码转化后，实际邮件不一定会马上被发送出去，因为有些邮件的软件有一次同时发送多个邮件的功能，也可能会有用户点击“收信”按钮以后才一并接收新邮件的功能。像这种何时建立通信连接何时发送数据的管理功能，从某种宽泛的意义上看属于 OSI 参考模型中会话层的功能。

应用在发送邮件的那一刻建立 TCP 连接，从而利用这个 TCP 连接发送数据。它的过程首先是将应用的数据发送给下一层的 TCP，再做实际的转发处理。

② TCP 模块的处理

TCP 根据应用的指示[▼]，负责建立连接、发送数据以及断开连接。TCP 提供将应用层发来的数据顺利发送至对端的可靠传输。

▼这种关于连接的指示相当于 OSI 参考模型中的会话层。

为了实现 TCP 的这一功能，需要在应用层数据的前端附加一个 TCP 首部。TCP 首部中包括源端口号和目标端口号（用以识别发送主机跟接收主机上的应用）、序号（用以发送的包中哪部分是数据）以及校验和[▼]（用以判断数据是否被损坏）。随后将附加了 TCP 首部的包再发送给 IP。

▼ Check Sum，用来检验数据的读取是否正常进行的方法。

③ IP 模块的处理

IP 将 TCP 传过来的 TCP 首部和 TCP 数据合起来当做自己的数据，并在 TCP 首部的前端在加上自己的 IP 首部。因此，IP 数据包中 IP 首部后面紧跟着 TCP 首部，然后才是应用的数据首部和数据本身。IP 首部中包含接收端 IP 地址以及发送端 IP 地址。紧随 IP 首部的还有用来判断其后面数据是 TCP 还是 UDP 的信息。

IP 包生成后，参考路由控制表决定接受此 IP 包的路由或主机。随后，IP 包将被发送给连接这些路由器或主机网络接口的驱动程序，以实现真正发送数据。

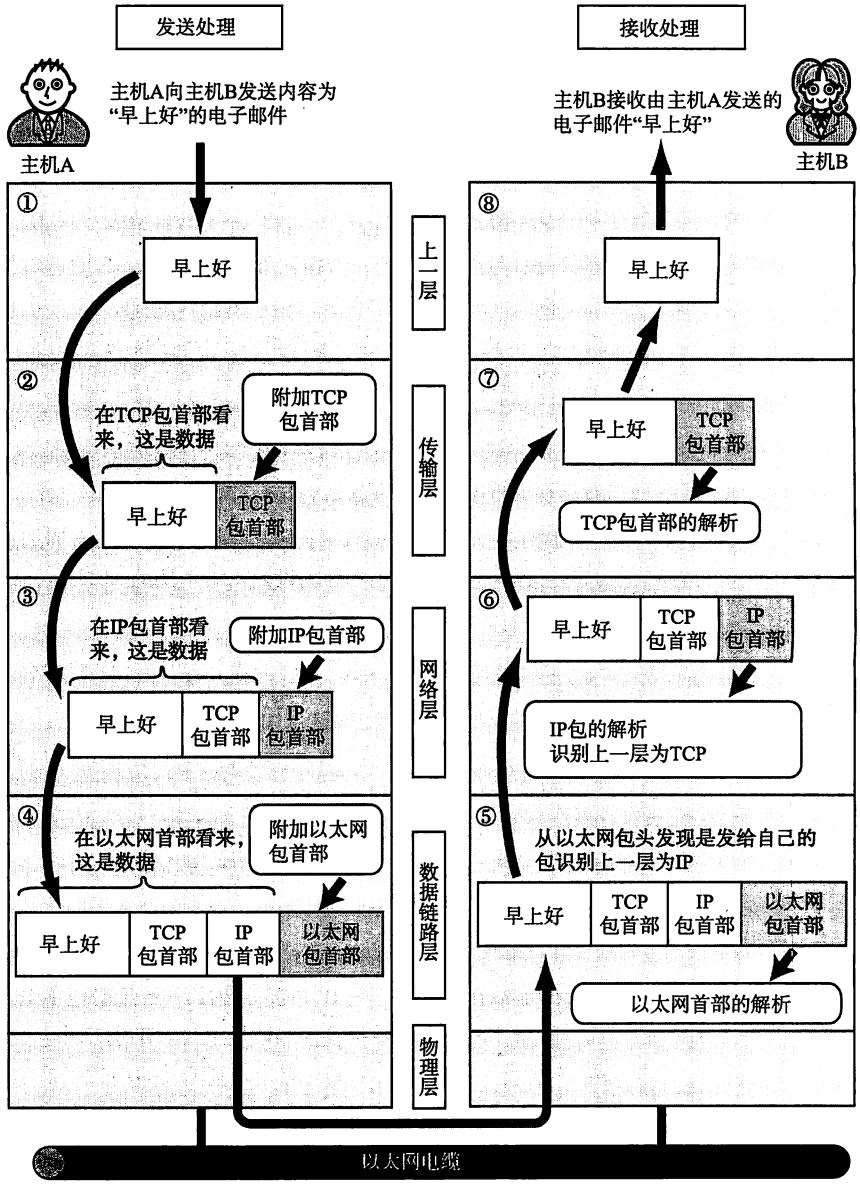
如果尚不知道接收端的 MAC 地址，可以利用 ARP（Address Resolution Protocol）查找。只要知道了对端的 MAC 地址，就可以将 MAC 地址和 IP 地址交给以太网的驱动程序，实现数据传输。

④ 网络接口（以太网驱动）的处理

从 IP 传过来的 IP 包，对于以太网驱动来说不过就是数据。给这数据附加上以太网首部并进行发送处理。以太网首部中包含接收端 MAC 地址、发送端 MAC 地址以及标志以太网类型的以太网数据的协议。根据上述信息产生的以太网数据包将通过物理层传输给接收端。发送处理中的 FCS[▼] 由硬件计算，添加到包的最后。设置 FCS 的目的是为了判断数据包是否由于噪声而被破坏。

▼ Frame Check Sequence

图2.18
TCP/IP各层对邮件的收发处理



2.5.3　经过数据链路的包

分组数据包（以下简称包）经过以太网的数据链路时的大致流程如图2.19所示。不过请注意，该图对各个包首部做了简化。

包流动时，从前往后依此被附加了以太网包首部、IP包首部、TCP包首部（或者UDP包首部）以及应用自己的包首部和数据。而包的最后则追加了以太网包尾▼（Ethernet Trailer）。

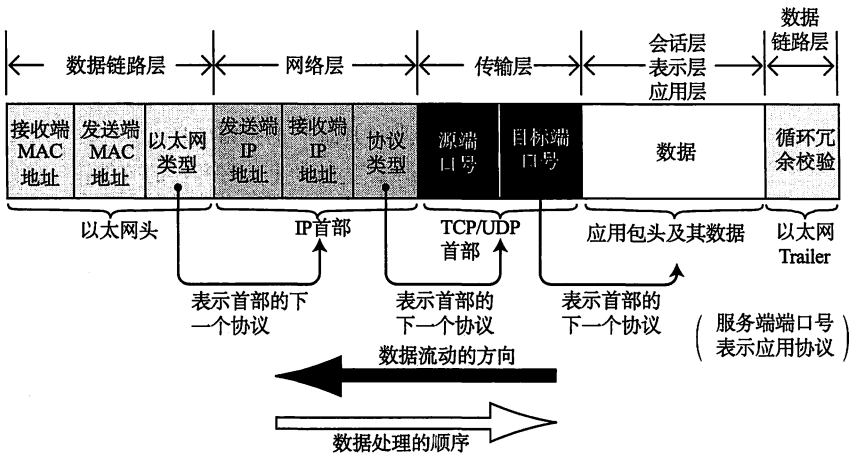
▼包首部附加于包的前端，而包尾则指追加到包的后端的部分。

每个包首部中至少都会包含两个信息：一个是发送端和接收端地址，另一个是上一层的协议类型。

经过每个协议分层时，都必须有识别包发送端和接收端的信息。以太网会用

MAC 地址，IP 会用 IP 地址，而 TCP/UDP 则会用端口号作为识别两端主机的地址。即使是在应用程序中，像电子邮件地址这样的信息也是一种地址标识。这些地址信息都在每个包经由各个分层时，附加到协议对应的包首部里边。

图 2.19
分层中包的结构



此外，每个分层的包首部中还包含一个识别位，它是用来标识上一层协议的种类信息。例如以太网的包首部中的以太网类型，IP 中的协议类型以及 TCP/UDP 中两个端口的端口号等都起着识别协议类型的作用。就是在应用的首部信息中，有时也会包含一个用来识别其数据类型的标签。

2.5.4 数据包接收处理

包的接收流程是发送流程的逆序过程。

⑤ 网络接口（以太网驱动）的处理

主机收到以太网包以后，首先从以太网的包首部找到 MAC 地址判断是否为发给自己的包。如果不是发给自己的包则丢弃数据▼。

▼很多 NIC 产品可以设置为即使不是发给自己的包也不丢弃数据。这可以用于监控网络流量。

而如果接收到了恰好是发给自己的包，就查找以太网包首部中的类型域从而确定以太网协议所传送过来的数据类型。在这个例子中数据类型显然是 IP 包，因此再将数据传给处理 IP 的子程序，如果这时不是 IP 而是其他诸如 ARP 的协议，就把数据传给 ARP 处理。总之，如果以太网包首部的类型域包含了一个无法识别的协议类型，则丢弃数据。

⑥ IP 模块的处理

IP 模块收到 IP 包首部及后面的数据部分以后，也做类似的处理。如果判断得出包首部中的 IP 地址与自己的 IP 地址匹配，则可接收数据并从中查找上一层的协议。如果上一层是 TCP 就将 IP 包首部之后的部分传给 TCP 处理；如果是 UDP 则将 IP 包首部后面的部分传给 UDP 处理。对于有路由器的情况下，接收端地址往往不是自己的地址，此时，需要借助路由控制表，在调查应该送达的主机或路由器以后再转发数据。

⑦ TCP 模块的处理

在 TCP 模块中，首先会计算一下校验和，判断数据是否被破坏。然后检查是否在按照序号接收数据。最后检查端口号，确定具体的应用程序。

数据接收完毕后，接收端则发送一个“确认回执”给发送端。如果这个回执信息未能达到发送端，那么发送端会认为接收端没有接收到数据而一直反复发送。

数据被完整地接收以后，会传给由端口号识别的应用程序。

⑧ 应用程序的处理

接收端应用程序会直接接收发送端发送的数据。通过解析数据可以获知邮件的收件人地址是乙的地址。如果主机B上没有乙的邮件信箱，那么主机B返回给发送端一个“无此收件地址”的报错信息。

但在这个例子中，主机B上恰好有乙的收件箱，所以主机B和收件人乙能够收到电子邮件的正文。邮件会被保存到本机的硬盘上。如果保存也能正常进行，那么接收端会返回一个“处理正常”的回执给发送端。反之，一旦出现磁盘满、邮件未能成功保存等问题，就会发送一个“处理异常”的回执给发送端。

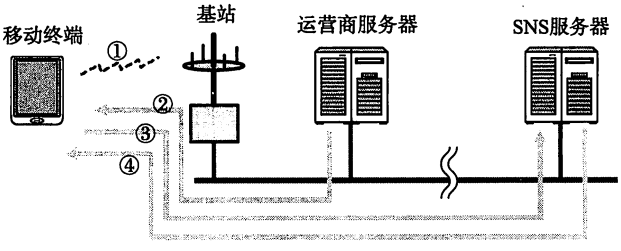
由此，用户乙就可以利用主机B上的邮件客户端，接收并阅读由主机A上的用户甲所发送过来的电子邮件——“早上好”。

SNS中的通信示例

SNS（Social Network Service），中文叫社交网络，是一种即时共享，即时发布消息给圈内特定联系人的一种服务。如前面电子邮件中通信过程的描述一样，也可以分析用移动终端发送或接收SNS消息的过程。

首先，由于移动电话、智能手机、平板电脑等在进行分组数据的通信，因此在它们装入电池开机的那一刻，已经由通信运营商设定了具体的IP地址。

启动移动电话中的应用程序时，会连接指定的服务器，经过用户名、密码验证以后服务器上积累的信息就会发送到手机终端上，并由该终端显示具体内容。



① 终端初始设置
② 由运营商设定终端的IP地址
③ 向SNS服务器发送信息进行用户认证
④ SNS服务器转发数据给终端

图2.20
TCP/IP中的网络分层

类似地，通过SNS轻轻一点就能够运行各种工具、发送文本动画等，这都基于互联网的TCP/IP应用。因此，在排查这些应用的问题时，TCP/IP的知识是必不可少的。