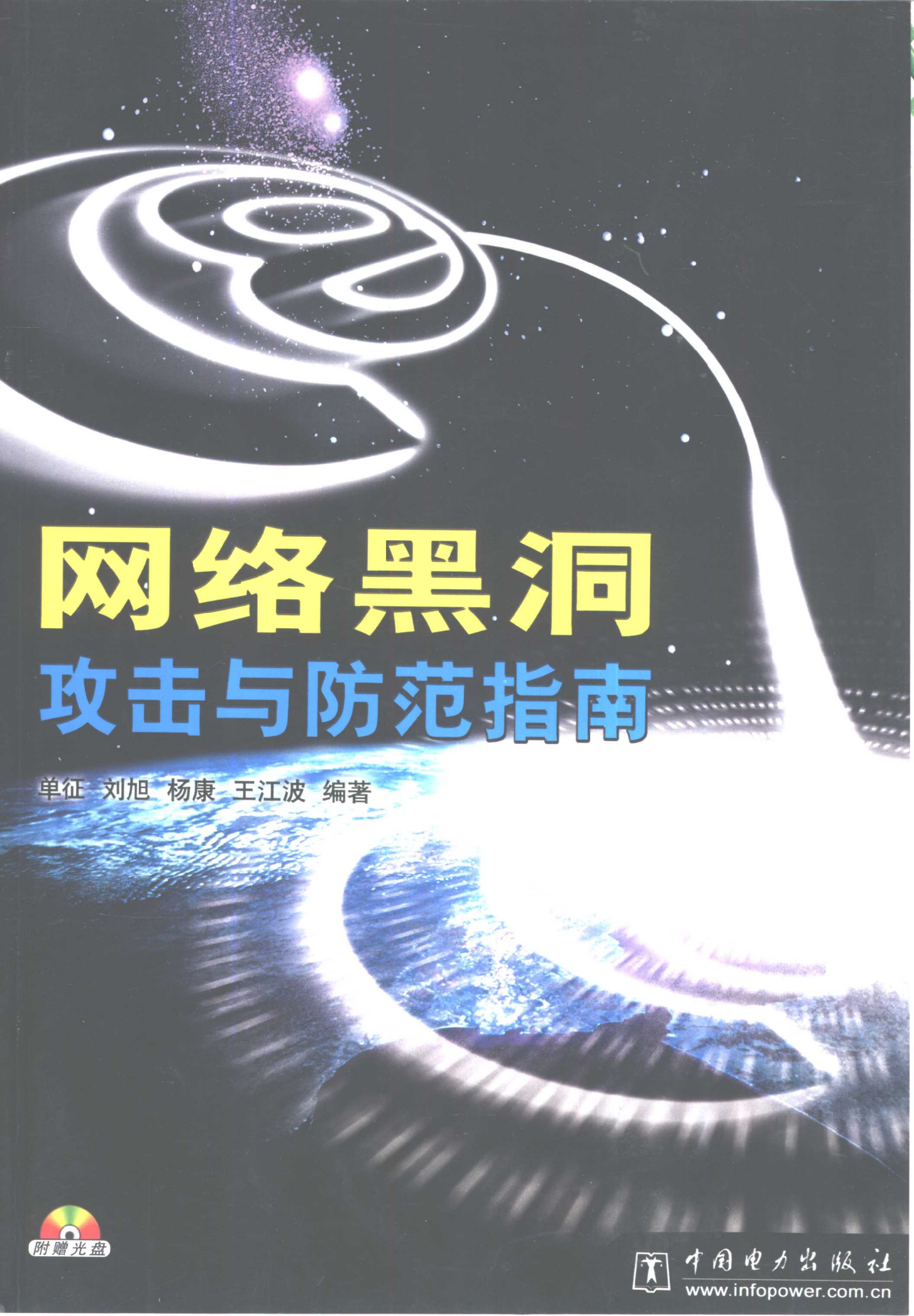

# 网络黑洞

## 攻击与防范指南

单征 刘旭 杨康 王江波 编著

附赠光盘

中国电力出版社
www.infopower.com.cn

# 网络黑洞
## 攻击与防范指南

单征 刘旭 杨康 王江波 编著

中国电力出版社

## 内 容 提 要

本书从网络安全的基础开始讲起，将重点放在目前常用的 Windows 和 NT 系列操作系统上，从攻击（黑客）和防御（网络管理员）两个角度论述了攻击与防御的有关知识体系。全书共分两个部分十三章，包括网络程序设计概述、套接字程序设计、MFC 套接字程序设计、MFC WinInet 程序设计、ISAPI 程序设计、URL 和 Moniker 的使用、网络开发工具介绍、黑客简史、攻击分析与防范、后门与木马、数据安全、安全测试程序设计以及入侵检测系统的设计与分析、网络资源、防火墙和入侵检测产品的介绍与选择等翔实的内容。

本书适合研究网络安全的初中级读者和安全技术人员参考学习使用。



中国电力出版社出版、发行
（北京三里河路 6 号　100044　http://www.infopower.com.cn）
三河市实验小学印刷厂印刷
各地新华书店经售
*
ISBN 7-900038-58-2/TP·45
2002 年 2 月第一版　　2002 年 2 月北京第一次印刷
787 毫米×1092 毫米　16 开本　24.5 印张　546 千字
定价 **39.00** 元

# 前　言

进入21世纪，因特网的发展已经影响到人们的日常生活。同时随着网络应用飞速发展，安全问题显得非同寻常，随着见诸报端的一幕幕网络攻击和犯罪事件，人们开始对网络的安全性感到担忧。

从1988年11月2日因特网（Internet）上发生的第一起影响广泛的“网络攻击”事件——蠕虫（WORM）起，这类事件不断发生，到最近的Microsoft的开发源代码被盗，“我爱你”、“梅莉莎”病毒的泛滥，Yahoo、AOL等著名的公司都不同程度地受到了来自网络的攻击，造成的损失不可估计。短短十几年内，来自网络的攻击和入侵无论是在数量上还是在技术上都有了质的“飞跃”，特别是2001年中美网络黑客大战，更是开创了“不见硝烟的战斗”的先河，网络安全的紧迫性不仅仅局限在几个密码破译、口令丢失的方面，而是涉及到整个国家民族的信息系统的安全。

黑客的攻击使许多公司甚至政府面临着他们从未想像到的威胁，网络安全产品的销售量也随之猛增。如何辨别防御攻击、增强网络安全性等已成为网络运营者和技术管理人员，甚至是普通的网络用户非常关心的问题。

本书结合当前比较流行的强大的编程语言Visual C++ 6.0所具有的网络方面能力，从一般的网络程序设计讲起，由浅入深地讲述了如何增强应用程序的安全性、如何编写安全测试程序、如何辨别攻击和防御，最后从黑客和网络管理员两个角度讲述了黑客程序的设计与入侵监测系统的设计。由于网络安全涉及范围广、技术复杂，许多核心软硬件技术秘密掌握在少数国家的几家大公司手里，因此本书涉及内容难免挂一漏万，在此表示遗憾。

本书有以下几个特点：

（1）由浅入深。从网络编程的基础讲到网络安全程序设计，从简单的检测网络主机是否活动的Ping程序设计到复杂的入侵监测系统设计。本书包含了许多作者的原创网络攻击和安全防范程序。

（2）由于本书讲的主要内容为设计，所以每章中不仅仅举几个实例，而且更侧重讲述的是程序设计的思想。同时给出更完整、更详细的编程设计方案，每个实例都包括设计、分析和源代码，让您更容易理解，从中能学到更多的东西。

（3）从Socket到ISAPI，每个类和函数都有详细讲解，而且书中含有许多其他中文书籍中很少出现的内容，比如“网络开发包libpcap、URL Moniker”等，对许多读者来说本书也不失为一本好的参考资料。

（4）本书讲述了Visual C++的安全方面的性能，读后可以举一反三，用Visual C++设计出安全测试程序和简单的入侵检测程序。

（5）尽可能地在描述程序设计的同时，介绍大量的系统和网络安全和管理知识，让您

读了本书后不仅可以成为一个优秀的网络程序设计员，也能成为一个优秀的网络或系统管理员。

（6）从攻击者（黑客）和防御者（网络管理员）两个角度论述了攻击和防御。

（7）本书介绍了目前大量的国际和国内的安全产品，让您在选择产品、构建企业或个人的安全系统时有据可依。

（8）本书附有光盘，光盘上的部分程序为黑客程序，主要供读者研究学习，以防范他人攻击。请不要利用这些程序攻击他人，否则一切后果自负。黑客程序对计算机系统及其安全可能会有危害。若无把握，请在专业人士指导下安装使用。对于可能由此造成的损害或损失，读者自行承担。

本书的第 1~6 章由刘旭创作，第 7~12 章由单征完成，王江波独立完成了第 13 章。由于时间仓促，书中错误与遗漏之处在所难免，希望广大读者予以批评指正。除了封面署名的作者之外，参加本书编写工作的还有陈亮、姜斌、赵雪峰、杨康、王勇、夏少强、常万明等，对他们的辛勤工作表示衷心的感谢。

感谢电力出版社的编辑为本书出版所作的大量工作，感谢吴斌、程显锋在本书撰写过程中给予编者的大力协助。

编　者

# 目　　录

# 第 1 章

# 网络程序设计概述

“千里之行，始于足下”，任何程序设计的开始，首先应当对自己将要使用的开发环境有所了解，其次应当对自己将要拓展的知识领域有所掌握。本章从网络协议概述开始，向读者逐一介绍在网络程序设计中经常使用到的一些协议的有关概念和细节，同时还介绍Visual C++的开发环境以及应用该环境进行网络编程的一些基本思想和方法。

## 1.1 网络协议概述

### 1.1.1 OSI 网络分层参考模型

所有的网络在传输协议上都是分层（layer）的，层的集合通常称为堆栈（stack），应用程序同最高层进行通话，最底层则与网络进行通话。网络协议设计者并不设计一个单一的、巨大的协议来为所有形式的通信规定完整的细节，而是把通信问题划分成多个小问题，然后为每一个小问题设计一个单独的协议。这样做使得每个协议的设计、分析、实现和测试都比较容易。协议划分的一个主要原则是确保目标系统有效且效率高。为了提高效率，每个协议只注意没有被其他协议处理过的那一部分通信问题。为了让协议的实现更加有效，协议之间应该能够共享特定的数据结构，同时这些协议的组合应该能够处理所有可能的硬件错误以及其他异常情况。为了保证这些协议工作的协同性，应当将协议设计和开发成完整的、协作的协议系列即协议族，而不是孤立地看待每个协议。

国际标准化组织（ISO）和国际电报电话咨询委员会（CCITT）共同出版了开放系统互联（OSI）的七层参考模型。一台计算机操作系统中的网络过程包括从应用请求（在协议栈顶部）到网络介质（协议栈底部），OSI参考模型把功能分成七个分立的层次，如图1-1所示。

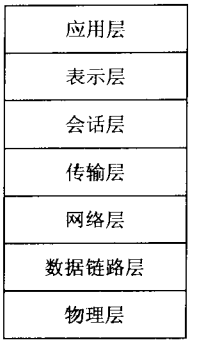


图 1-1　OSI 网络模型

### 第一层——物理层

物理层负责最后将信息编码成电流脉冲和其他信号用于网上传输。它由计算机和网络介质之间的实际界面组成，可定义电气型号、符号、线的状态和时钟要求、数据编码和数据传输用的连接器。如最常用的RS-232规范、10Base-T曼彻斯特编码以及RJ-45就属于第一层。所有比物理层高的层都通过事先定义好的接口与之对话。如以太网的辅助单元接口（AUI），一个DB-15连接器可被用来连接层1和层2。

### 第二层——数据链路层

数据链路层通过物理网络链路提供可靠的数据传输。不同的数据链路层定义了不同的网络和协议特征，其中包括物理编址、网络拓扑结构、错误校验、帧序列以及流控。物理编制定义了设备在数据链路层上的编址方式；网络拓扑结构定义了设备的物理连接方式（如总线拓扑结构或者环形拓扑结构）；错误校验向发生传输错误的上层协议告警；数据帧序列重新整理并传输除序列以外的帧；流控可以延缓数据的传输，以使接收设备不会因为在某一时刻接收到超过其处理能力的信息流而崩溃。数据链路层实际上由两个独立的部分组成，介质存取控制（MAC）和逻辑链路控制层（LLC）。MAC描述在共享介质环境中如何进行站的调度、发生和接收数据。逻辑链路控制确保信息跨链路的可靠传输，对数据传输进行同步、识别错误和控制数据的流向。一般来讲，介质存取控制只在共享介质环境中才是重要的，只有在共享介质环境中多个节点才能同时连接到一个传输介质上。IEEE MAC规则定义了“地址”，以标识数据链路层中的多个设备。逻辑链路控制层管理单一网络链路上的设备间的通信，IEEE 802.2标准定义了LLC。LLC支持无连接服务和面向连接的服务。在数据链路层的信息中定义了许多域，这些域使得多种高层协议可以共享一个物理数据链路。

### 第三层——网络层

网络层负责在源和终点之间建立连接。它一般包括网络寻址，还可能包括流量控制、错误检查等。相同MAC标准的不同网段之间的数据传输一般只涉及到数据链路层，而不同

的MAC标准之间的数据传输都要涉及到网络层。例如，IP路由器工作在网络层，因而可以实现多种网络间的互连。

### 第四层——传输层

传输层向高层提供可靠的端到端的网络数据流服务。传输层的功能一般包括流控、多路传输、虚电路管理以及差错校验和恢复。流控管理设备之间的数据传输，确保传输设备不发送比接收设备处理能力大的数据；多路传输使得多个应用程序的数据可以传输到一个物理链路上；虚电路由传输层建立、维护和终止；差错校验包括为检测传输错误而建立的各种不同的结构；而差错恢复包括所采取的行为，以便解决发生的任何错误。传输控制协议是提供可靠数据传输的TCP/IP族中的传输层协议。

### 第五层——会话层

会话层建立、管理和终止表示层与实体之间的通信会话，通信会话包括发生在不同网络设备的应用层之间的服务请求和服务应答，这些请求与应答通过会话层的协议实现。它还包括创建检查点，使通信发生中断的时候可以返回到此前的某一个状态。

### 第六层——表示层

表示层提供多种功能用于应用层数据编码和转化，以确保从一个系统应用层发送的信息可以被另一个系统应用层识别。表示层的编码和转化模式包括公用数据表示格式、性能转化表示格式、公用数据压缩模式和公用数据加密格式。

标准的图像、声音和视频格式都属于公用数据表示格式。通过使用这些标准格式，不同类型的计算机系统可以相互交换数据；转化模式通过使用不同的文本和数据表示，在系统间交换信息，例如ASCII(American Standard Code For Information Interchange，美国标准信息交换码)；标准数据压缩50确保原始设备上被压缩的数据可以在单目标设备上正确地给予；加密模式确保原始设备上加密的数据在可预测目标设备上的正确加密。

### 第七层——应用层

应用层是最接近终端用户的OSI层，它面向用户应用，包含用户界面，提供了对用户应用程序的直接支持和管理，这些用户应用程序包括数据库访问、电子邮件以及文件传输等。同时允许应用程序与其他计算机上的应用程序进行通信。*应当注意的是*：应用层并非由计算机上运行的实际应用软件组成，而是由向应用程序提供访问网络资源的API（Application Program Interface，应用程序接口）组成，这类应用软件程序超出了OSI模型的范畴。

## 1.1.2　TCP/IP 分层模型

在因特网上的计算机通过互相发送信息分组进行通信。这些信息分组由成块的数据、

特定的控制和寻址信息组成。控制和寻址信息用来确保信息包到达目的地，并能对目的主机重新组成可以使用的计算机数据。这种信息传输方式是由TCP/IP规定的。因此，TCP/IP也被称为是Internet上的通用语言。它是一组计算机通信协议的集合，其目的是允许互相合作的计算机系统通过网络共享彼此的信息，并应用到异构的网络系统中。现在，TCP/IP已成为Internet的标准协议和WWW(万维网)的基础。

TCP/IP也可以看做是一个协议族，该族包含了上百个协议，TCP/IP只是其中最著名的两个。但由于人们最熟悉它们，所以通常用TCP/IP称呼这个协议族。

下面来介绍TCP/IP与OSI对应关系。

OSI参考模型本身并不是一种规范，该模型只起到一个指导作用，在模型的具体实现上可以变得很灵活。TCP/IP模型是实际中Internet的标准模型，TCP/IP网络只使用了OSI参考模型中的四层，如图1-2所示。


**图 1-2　TCP/IP 模型**

TCP/IP是一个开放式网络协议，非常适用于网间互连。与OSI参考模型不同，TCP/IP不是人为制定的标准，而是产生在网络研究和实践应用中。TCP/IP模型分为4层，分别为应用层、传输层、网络层和接口层，同时各层完成的功能与OSI参考模型也有所不同。由于TCP/IP的目的在于解决异构网络互连问题，同时隐藏网络细节，以便为用户提供通用的、一致的通信服务，因此从一开始TCP/IP就考虑了对底层网络技术的包容问题，故在TCP/IP中并不包括物理层。同时在TCP/IP中，也不包括标准的表示层，应用程序必须完成表示层的功能。TCP/IP也不包括会话层，但会话层的功能，要么是由传输层协议要么是由用户应用程序来实现的。

TCP/IP族中的协议共同工作，来完成因特网上的数据传输任务。这些协议集合提供了今天因特网上的所有实用型服务。TCP/IP分为两大类，一类是网络协议，另一类是应用协议，网络协议是应用协议的基础。网络协议管理数据传输的具体结构，包括传输层以下的所有协议，主要的网络协议有：地址解析协议（ARP）、因特网控制报文协议（ICMP）、网络协议（IP）和传输控制协议（TCP）等。这些协议进程对用户来说一般是不可见的，除非用户利用应用程序监视系统进程，它们运行在系统一级。应用协议是所有运行在应用层上的协议。与网络协议相反，应用协议的有些部分对于用户来说是可见的。比如，文件传输协议（FTP），用户请求与另一台机器建立连接，传输数据，在传输过程中，用户可以看到用户计算机和远端计算机之间的部分交换信息。

### 第一层——应用层

应用层面向用户提供一组常用的应用程序，如电子邮件、文件传输等。严格来讲，TCP/IP只包含下3层，应用程序不能算TCP/IP的一部分。就上面提到的常用应用程序，TCP/IP制定了相应的协议标准，所以一般也把它们作为TCP/IP的内容，常用的协议有：FTP、HTTP、Telnet、Gopher、SMTP等。事实上，用户完全可以根据自己需要建立网络与网络之间的专用应用程序，这些程序要用到TCP/IP，但不属于TCP/IP。

### 第二层——传输层

提供应用程序间的连接和任务服务，它可以在进程之间提供可靠有效的传输服务，其功能主要包括：

（1）提供可靠传输。传输层协议规定接收端必须发回确认信息，并且如果分组丢失，那么必须重新发送。

（2）报文分组和重组。如果发送的报文大小超过目的主机的接收能力，则要对报文进行分组。

（3）流量控制。

传输层还要解决不同应用程序间的识别问题，因为在一般的通用计算机中，常常是多个应用程序同时运行。为了区别各种应用程序，传输层在每一个分组中增加识别信源和信宿的应用程序信息。另外，传输层中每一个分组均附带校验和是专用名词，接收端以此来校验收到的分组的正确性。

### 第三层——网络层

负责相邻计算机之间的通信问题。其功能主要有以下三个方面：

（1）处理来自主传输层的分组发送请求。收到请求后，将分组装入IP数据包，填充报头，选择路径，然后将数据包发往适当的网络接口。处理接收数据报，首先检查其合法性，然后寻址。假如该数据包已到达目的地，则去掉IP报头，将剩下的部分交给适当的传输协议进行处理；假如该数据包尚未到达目的地，则转发该数据包。

（2）提供数据报的分组和重组，以支持最大传输单元（MTU）不同的数据链路。

（3）处理ICMP报文、路径、流控和拥塞等问题。

### 第四层——接口层

这是TCP/IP的最底层，负责接收IP数据包并通过网络发送出去或者从网络上接收物理帧，抽出IP数据报，交给IP层。

网络接口有两种类型，一种是设备驱动程序，如局域网的网络接口；另一种是包含自身数据链路协议的复杂子系统。

## 1.1.3 网络协议概述

### 网间协议（IP）

因特网的核心层是网络层和传输层。IP 是建立 TCP/IP 网络的最基本的协议，它定义了在整个 TCP/IP 网络上传输数据所用的基本单元。网络层协议为上层协议提供了无连接的、不可靠的数据报传送服务，其他协议作为 IP 数据报的数据被承载。

IP 只负责数据的传递，并不对数据的到达以及到达的连续性和顺序性做任何的保证，也不提供任何的纠错功能。

#### IP数据报格式

TCP / IP使用的IP数据报和物理网络上传输数据单元用的硬件帧有相同的格式，它包含IP报头和IP数据区，如图1-3所示。

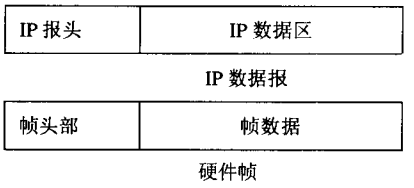


**图 1-3 IP 数据报的一般格式**

IP数据报包含有源、目的信息，在穿越因特网时作寻址作用，并且指明承载负载的协议类型，如TCP、ICMP、UDP等。数据报所携带的数据量不固定，发送方根据特定的用途选择合适的数据量。在IPv4版本中，一个数据报的数据量可以小到一个字节，而如果包括报头的话，则数据包可以大到64KB。图1-4给出了IP数据报更详细的格式。

| 版本 | 报头长度 | 服务类型 | 总长度 | |
|---|---|---|---|---|
| 标识符 | | | 标志 | 分片偏移量 |
| 存活时间 | 协议 | | 报头校验和 | |
| 源地址 | | | | |
| 目的地址 | | | | |
| IP 选项 | | | | 填充 |
| 数据区..... | | | | |

**图 1-4 IP 数据报格式**

为了方便编程，这里同时给出了IP数据报的C语言结构。应当注意的是：它并没有包括可选项和填充域。

```
/* 利用 VC++定义的 IP 包头 */
typedef struct tagIPHEADER
{
    unsigned char h_len : 4;           // 包头长度
    unsigned char version : 4;         // IP 版本号
    unsigned char tos;                 // 服务类型
    unsigned short total_len;          // 包长

    unsigned short ident;              // 序列号
    unsigned short frag_and_flags;     // 标识和偏移
    unsigned char ttl;                 // 存活时间
    unsigned char proto;               // 头部校验和

    unsigned int sourceIP;             // IP 源地址
    unsigned int destIP;               // IP 目标地址
}IPHEADER;
```

## 传输控制协议（TCP）

### 1. 概述

TCP提供IP环境下的数据可靠传输。提供的服务包括数据流传输、可靠性、有效流控、全双工操作和多路复用。使用数据流传送，TCP发送用序列号标识的非结构化的比特流。这种服务给应用带来了方便，因为它们不必再将数据传给TCP以前对数据进行分块。TCP将它们分组，并传输给IP层。

TCP 的上层是应用程序，上层接口包括了一系列类似于操作系统的中断调用。对于上层应用程序来说，TCP 应该能够实现异步传送数据。我们假定下层接口为 IP 接口。为了在并不可靠的网络上实现面向连接的可靠的传送数据，TCP 必须解决的问题有可靠性、流量控制等。同时它还必须能够为上层应用程序提供多个接口，并同时为多个应用程序提供数据。此外，TCP 还必须解决连接问题，这样 TCP 才能称得上是面向连接的，最后，TCP 也必须能够解决通信安全性的问题。

网络环境包括由网关、路由器、交换机等（或其他设备）连接的网络，网络可以是局域网也可以是一些城域网或广域网，但无论它们是什么，它们都必须是基于包交换的。主机上不同的协议有不同的端口号，一对进程通过这个端口号进行通信。这个通信不包括计算机内的 I/O 操作，只包括在网络上进行的操作。网络上的计算机被看做包传送的源节点和目的节点。应该注意的是：计算机的不同进程可以同时进行通信，此时它们会用不同的端口号加以区别，而不会把发向 A 进程的数据由 B 进程进行接收。

进程为了传送数据会调用 TCP，将数据和相应的参数传送给 TCP，于是 TCP 会将数据传送到目的 TCP 那里，当然这是通过将 TCP 包打包在 IP 包内在网络上传送达到的。接收方 TCP 在接收到数据后会和上层应用程序进行通信，TCP 会保证接收数据顺序的正确性以及数据包的完整性，虽然下层协议可能不会保证顺序是正确的。需要说明的是，网关在接收到这个数据包后，会将包解开，看看是不是已经到目的地了，如果没有到，则判断应该走什么路由达到目的地，在决定后，网关会根据下一个网络内的协议情况再次将 TCP 包打包传送出去，如果需要，还要把这个包再次分成几段再传送。这就是 TCP 传送的基本过程，当然具体过程可能还要复杂得多。

在实现 TCP 的主机上，TCP 可以被看成是一个模块，和文件系统区别不大。TCP 也可以调用一些操作系统的功能，TCP 不直接和网络打交道，控制网络的任务由专门的设备驱动模块完成。TCP 只是调用 IP 接口，IP 向 TCP 提供所有 TCP 需要的服务。

### 2. 数据格式

通过面向连接的、端到端的可靠数据包发送，TCP提供数据传输的可靠性。TCP通过使用转发确认号对字节排序来实现可靠性。确认号将源节点希望接收的下一个字节指示给目的主机。在一段时间内，没有确认的字节将被重新发送。TCP的可靠性机制允许设备处理丢失、延迟、重复或错读数据包。这种机制允许设备检测丢失的数据包并重新请求传输。

TCP段的基本格式如图1-5所示。

<table>
<tr><td colspan="8">源端口</td><td colspan="2">目的端口</td></tr>
<tr><td colspan="10">序号</td></tr>
<tr><td colspan="10">确认号</td></tr>
<tr><td>数据偏移</td><td>保留</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FTN</td><td colspan="2">窗口</td></tr>
<tr><td colspan="8">紧急指针</td><td colspan="2">校验和</td></tr>
<tr><td colspan="9">IP 选项</td><td>填充</td></tr>
<tr><td colspan="10">数据区.....</td></tr>
</table>

**图 1-5　TCP 数据报格式**

各个域的具体含义如下：

（1）源端口（Source Port）。

该域的长度为16 bits，用来表示源端口号。

（2）目的端口（Destination Port）。

该域长度为16 bits，用来识别目的端口号。目的端口和源端口加上源和目的的IP地址可以确定一个惟一的TCP连接。

（3）发送序列号（Sequence Number）。

该域长度为32 bits，用来指出TCP 5段中数据区本次发送数据的起始位置，TCP以字节

为单位来确认当前段在全部信息中的位置，因此可以用序号来确定数据流中的每一个字节。TCP为每一个连接选择一个初始序列号（ISN）。为了防止被延误或重复的数据包混淆信息，ISN不能随便选择，不同的系统采用了不同的算法。比如发送序列号为“1a 4a 00 c3”，表示本次发送信息是从第4410557等475B开始的。

（4）应答序列号（Acknowledgment Number）。

该域长度为32 bits，与ACK标志结合使用，用来指出本机已经确认的字节数，并希望从数据流中接收下一个字节的序列号。比如应答序列号为“5d f5 96 5d”，ACK标志为“1”，表示已经正确接收了1576375900－ISN个字节，并希望从1576375901B开始接收，具体能接收的字节数从窗口域中可以得到。

（5）TCP头标长（Header Length）。

该域长度为4 bits，用来指定TCP头标的长度，以32 bits为单位来确认TCP头标长度。

（6）保留（Reserved）。

该域长度为6 bits，目前保留，用于将来扩展使用，必须为0。

（7）控制标志（Control Flags）。

该域长度为6 bits，用于指示各种控制信息，包括6个1 bits的标志位。

（8）窗口（Window）。

该域长度为16 bits，用于通知接收端，发送者之前能够接受的字节数，以防止接收缓存区溢出而丢失大量数据。比如发送段窗口大小为“1f cb”，表示发送方能够目前接受的最大信息量为8139B。

（9）校验和（Checksum）。

该域长度为16 bits，用于检验TCP头标传输过程中的正确性。

（10）紧急指针（Urgent Pointer）。

该域长度为16 bits，它与该TCP段的发送序号相加，形成紧急指针。紧急指针指定情急数据的最后一个字节的序列号。

（11）选项（Option）。

该域长度可变，与IP头的选项相似，用于具体TCP选项，每个选项由一个选择号、选项中字节个数和选项值组成。通常TCP的值定义三个选项：

0——选项表结束

1——没有操作和

2——最大段大小

（12）填充（Padding）。

Option域的长度是可变的，插入一个填充域以达到一个32 bits的界限。

### 3. TCP的序列号

在 TCP 发送的消息格式中，发送的每个数据包都有一个序列号。根据这个编号，可以确认它们是否被接收到。对序列号的确认是累积性的，也就是说，如果用户收到对 X 的确认信息，就表示在 X 以前的数据（不包括 X）都接收到了。