

高等学校数字媒体技术系列教材

网络游戏 开发技术

Online Game
Programing

王方石 吴炜 编著

高等教育出版社

高等学校数字媒体技术系列教材

网络游戏 开发技术

Wangluo Youxi

Kaifa Jishu

王方石 吴 炜 编著

高等教育出版社·北京

内容提要

本书主要介绍网络游戏开发中难度较大的服务器端开发技术，服务器端开发人员不但应掌握基本的编程知识和计算机网络技术，还应掌握众多相关算法。为了便于读者学习，本书首先介绍网络游戏的发展历史、计算机网络的基础知识和基本协议，然后介绍网络游戏客户端和服务端开发设计的基础知识，并分析服务器端开发过程中涉及各个模块，同时提供大量实例，详细讲解网络游戏服务器端编程技术。通过本书的学习，读者可以独立实现简单的网络游戏服务器端功能。

本书可供高等学校本科计算机科学与技术、数字媒体技术等专业开设网络游戏相关课程时使用，也可作为相关技术人员的参考用书。

图书在版编目（CIP）数据

网络游戏开发技术 / 王方石, 吴炜编著. —北京: 高等教育出版社, 2016. 2

ISBN 978 - 7 - 04 - 044723 - 1

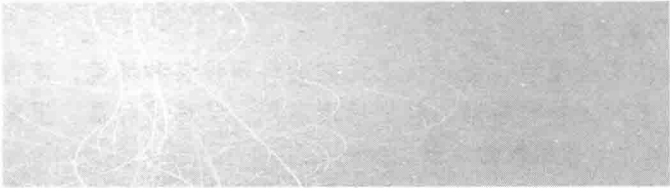
I. ①网… II. ①王… ②吴… III. ①互联网络-游戏-游戏程序-程序设计-教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2016) 第 021148 号

策划编辑 韩 飞	责任编辑 韩 飞	封面设计 王 琰	版式设计 杜微言
插图绘制 杜晓丹	责任校对 李大鹏	责任印制 韩 刚	

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街 4 号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.hepmall.com.cn
印 刷	河北新华第一印刷有限责任公司		http://www.hepmall.com
开 本	787mm × 1092mm 1/16		http://www.hepmall.cn
印 张	10.75		
字 数	240 千字	版 次	2016 年 2 月第 1 版
购书热线	010 - 58581118	印 次	2016 年 2 月第 1 次印刷
咨询电话	400 - 810 - 0598	定 价	21.00 元

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换
版权所有 侵权必究
物 料 号 44723 - 00



前言

网络游戏是具有辉煌前景的朝阳产业，存在着巨大的发展潜力和广阔的市场空间。据统计，2013 年我国网络游戏市场规模突破 890 亿元，同比增长 32.9%¹，其中网页游戏和手机游戏成为发展的重点，且其 3D 化趋势越来越明显。我国网络游戏市场的高速发展让人们对这一产业的价值刮目相看。但是，随着技术的发展，网络游戏公司对人才的要求也逐渐提高，真正符合需求的人才非常匮乏。

为培养产业所需的应用型、实践型人才，教育部 2004 年正式批准设置了数字媒体技术专业。2011 年，教育部示范性软件学院建设工作办公室组织编制了《高等学校数字媒体技术专业规范》，并以其为指导，成立数字媒体技术系列教材编委会，组织多所高校共同编写了“高等学校数字媒体技术系列教材”，本书即为其中的一本。

网络游戏开发分为客户端开发和服务器端开发。服务器端开发相对比较容易掌握，除需要掌握基本的编程知识和相关算法外，还需要对计算机网络有深入的了解。本书较为全面地介绍了网络游戏服务器端开发所需要的专业知识和基本方法，系统地介绍了服务器端的协议及各个模块实现的具体步骤。为便于读者学习，书中首先介绍了计算机网络基础知识，包括基本协议、Socket 编程和 Winsock 编程方法；然后介绍网络游戏客户端和服务端设计开发的基础知识，分析服务器端的各个模块，并以大量的实例详细讲解网络游戏服务器端编程技术。

本书适用于 48 学时的教学。学时分配具体如下：第 1 章 网络游戏概述（2 学时），第 2 章 与网络编程有关的协议简介（4 学时），第 3 章 网络游戏服务器端开发及优化（6 学时），第 4 章 网络游戏客户端开发（4 学时），第 5 章 网络游戏通信模块开发（6 学时），第 6 章 网络游戏规则模块开发（6 学时），第 7 章 网络游戏多线程技术（6 学时），第 8 章 网络游戏世界管理模块（6 学时），第 9 章 网络游戏开发实例（8 学时）。

本书第 1~4 章由王方石撰写，第 5~9 章由吴炜撰写，王方石负责全书的统稿与校对工作。浙江工业大学计算机科学与技术学院（软件学院）院长王万良教授审阅了全书，并提出很

¹ 数据来源：艾瑞咨询。

多中肯的建议与意见，在此向他表示衷心的感谢。

感谢北京交通大学软件学院数字媒体实验室为作者提供了优良的科研设备和环境，使得本书得以顺利完成。感谢实验室各位同学的支持和帮助，特别感谢温阳、吴琮、师磊、王震军、朱雪阳、郑帅、吴霏宸、李倩等同学为本书所做的贡献。

网络游戏开发所涉及的专业领域知识十分广泛，与之相关的学科也很多，由于作者的学识和水平有限，书中的不妥之处在所难免，敬请读者批评指正。

作者

2015年4月

目录

第1章 网络游戏概述 1

- 1.1 Internet 简介 1
- 1.2 网络游戏简介及其发展 2
- 1.3 网络游戏服务器端介绍 3
- 1.4 主流网络游戏技术简介 4
- 习题 1 5

第2章 与网络编程有关的协议简介 6

- 2.1 OSI 七层协议 6
- 2.2 TCP/IP 8
- 2.3 UDP 10
- 2.4 IP 11
- 2.5 ARP 11
- 2.6 ICMP 12
- 习题 2 13

第3章 网络游戏服务器端开发及优化 14

- 3.1 UNIX 套接字介绍 14
 - 3.1.1 Socket 套接字介绍 14
 - 3.1.2 套接字的创建和关闭 16
 - 3.1.3 连接处理 17
 - 3.1.4 select()函数 18
- 3.2 Windows Socket 介绍 21

- 3.3 TCP 实例 24

- 3.4 UDP 实例 29

- 3.5 广播实例 32

- 3.6 服务器端基本的优化方法 33

- 习题 3 34

第4章 网络游戏客户端开发 35

- 4.1 初始化客户端 35

- 4.2 TCP 客户端与服务器端通信 38

- 4.3 UDP 客户端与服务器端通信 42

- 4.4 客户端协议的定制 44

- 4.5 客户端协议的架构 44

- 4.6 客户端数据的接收和发送 56

- 4.7 客户端错误信息 59

- 习题 4 60

第5章 网络游戏通信模块开发 61

- 5.1 Socket 的封装 61

- 5.2 客户端数据读入 62

- 5.3 多路复用技术 63

- 5.4 通信模块的消息机制 68

- 5.5 通信模块的架构与实现 69

- 5.5.1 字节流 70

5.5.2 协议基类.....	74	第 8 章 网络游戏世界管理模块.....	121
5.5.3 Socket 封装类.....	79	8.1 服务器搭建.....	121
5.5.4 会话基类.....	82	8.2 数据的压缩与加密.....	124
5.5.5 NetworkFacade.....	87	8.3 世界管理模块构建.....	126
习题 5.....	90	8.4 世界管理模块实现.....	128
第 6 章 网络游戏规则模块开发.....	91	习题 8.....	131
6.1 业务逻辑的消息定义.....	91	第 9 章 网络游戏开发实例.....	132
6.2 业务逻辑的消息处理.....	92	9.1 ACE 架构介绍.....	132
6.3 业务逻辑的消息管理.....	93	9.2 ACE Socket Wrapper Facade.....	133
6.4 网络游戏规则模块的架构.....	95	9.3 ACE 进程 Wrapper Facade.....	135
6.5 网络游戏规则模块的实现.....	96	9.4 ACE 线程 Wrapper Facade.....	137
习题 6.....	99	9.5 ACE 同步 Wrapper Facade.....	139
第 7 章 网络游戏多线程技术.....	100	9.6 服务器搭建.....	142
7.1 进程间通信.....	100	9.7 服务器优化.....	150
7.2 多线程技术.....	102	9.8 客户端实现.....	154
7.3 同步控制机制.....	105	9.9 大型网络游戏实现.....	156
7.4 多线程技术应用.....	118	习题 9.....	162
习题 7.....	120	参考文献.....	163

第1章 网络游戏概述

为了帮助读者更好地了解网络游戏开发，本章从 Internet 简介开始，介绍网络游戏的发展史、网络游戏客户端和服务端的基本知识及其使用的主流技术。

1.1 Internet 简介

Internet（因特网，又名国际互联网）是由世界范围内各种大大小小的计算机网络相互连接而成的全球计算机网络。只要连接到它的任何一个节点上，就意味着用户的计算机已经连入 Internet 网络了。人们在 Internet 上开发了许多应用系统供入网的用户使用，用户还可以在 Internet 上方便地交换信息，共享资源。Internet 使用 TCP/IP（传输控制协议/网间协议）互相通信，它是一个无级网络，不专门为某个人或某个组织所拥有和控制，人人都可以参与其中。

Internet 起源于美国，现已成为连通全世界的一个超级计算机互联网络。Internet 在美国分为三层：底层为大学校园网或企业网，中层为地区网，上层为全国主干网。如国家自然科学基金网 NSFNET（National Science Foundation Network）等主干网连通了美国东、西海岸，并通过海底电缆或卫星通信等手段连接到世界各国。

Internet 之所以能发展如此迅猛，主要归功于其以下特点。

（1）Internet 是一个全球计算机互联的网络，用户可以在世界范围内的任意地点、任何时刻访问到连在互联网上的任何节点。

（2）Internet 对用户的计算机配置和网络操作技能要求低，费用也低，但带宽要求越来越高。

（3）它是一个巨大的信息资源，参与人数众多，共同享用着人类自己创造的财富（即资源）。

Internet 是一个网络,凡是采用 TCP/IP 并且能够与 Internet 中的任何一台主机进行通信的计算机,都可以看成是 Internet 的一部分。Internet 采用分布式的客户端/服务器架构,大大增强了网络信息服务的灵活性。

Internet 最初的宗旨是为大学和科研单位服务,由于其信息丰富、收费低廉,不但成为服务于全社会的通用信息网络,而且早已出现商业化趋势。美国在 Internet 骨干网的经营方面已经商业化,例如,美国国家科学基金会把 NSFNET 分成 SPRINTNET、MCINET 和 ANSNET 三部分,并将其管理权和经营权分别交给美国最大的三家电信公司,即 SPRINT、MCI 和 ANS。在 NSFNET 中建立一系列的网络存取点(Network Access Point)用于集中存放路由器,可为客户提供入网服务。

1.2 网络游戏简介及其发展

MMOGAME (Massive Multiplayer Online Game, 又称“大型多人在线游戏”,简称“网游”)是必须依托于互联网进行、可多人同时参与的游戏,通过人与人之间的互动达到交流、娱乐和休闲的目的。

第一款真正意义上的网络游戏可追溯到 1969 年,当时瑞克·布罗米为 PLATO (Programmed Logic for Automatic Teaching Operations) 系统编写了一款名为《太空大战》(SpaceWar)的游戏,它以 8 年前诞生于美国麻省理工学院的第一款计算机游戏《太空大战》为蓝本,不同之处在于它可支持两人远程连线。

网络游戏市场的迅速膨胀刺激了网络服务业的发展,网络游戏开始进入收费时代,许多消费者都愿意为网络游戏支付高昂的费用。从《凯斯迈之岛》(The Island of Kesmai)的每小时 12 美元到 GEnie 的每小时 6 美元,第二代网络游戏的主流计费方式是按小时计费,尽管也有过包月计费的特例,但未能形成规模。

1978 年,英国埃塞克斯大学的罗伊·特鲁布肖用 DEC-10 编写了世界上第一款 MUD 游戏——MUD1,这是一个纯文字的多人世界,拥有 20 个相互连接的房间和 10 条指令,用户登录后可以通过数据库进行人机交互,或通过聊天系统与其他玩家交流。

这是第一款真正意义上的实时多人交互网络游戏,它可以保证整个虚拟世界的持续发展。尽管这套系统每天都会重启若干次,但重启后游戏中的场景、怪物和谜题仍保持不变,这使得玩家所扮演的角色可获得持续的发展。

1991 年,Sierra 公司架设了世界上第一个专门用于网络游戏的服务平台——The Sierra Network (后改名为 ImagiNation Network,1996 年被 AOL 收购),这个平台类似于国内的联众游戏,它的第一个版本主要用于运行棋牌游戏,第二个版本加入了《叶塞伯斯的阴影》(The Shadow of Yserbius)、《红色伯爵》(Red Baron)和《幻想空间》(Leisure Suit Larry Vegas)等

功能更为复杂的网络游戏。随后几年内,MPG-Net、TEN、Engage 和 Mplayer 等一批网络游戏专用平台相继出现。

1996 年开始,随着网络游戏的发展,引入了“大型网络游戏”(MMOG)的概念,网络游戏不再依托于单一的服务商和服务平台而存在,而是直接接入互联网,在全球范围内形成了一个大一统的市场。其中代表性的作品有《无尽的任务》、《天堂》、《魔兽世界》等。

2006 年之后,随着 Web 技术的发展,网站技术在各个层面得到提升,“无端网游”,即不用客户端也能玩的游戏,亦称网页游戏或 WebGame 逐渐兴起。其实,这种特殊的游戏类型早在网络游戏盛行之前的 Mud 时代就已经存在,只是因其玩法单调、模式固化和交互简单而没有引起玩家过多的关注,但延续至今却也拥有了一批忠实的玩家。网页游戏以其无需游戏客户端、在浏览器中点击即可操作的方便快捷的游戏方式,受到上班族的青睐和追捧。随着宽带的发展,网页游戏将会成为网络游戏的重要补充模式。

WebGame 一般是采用服务器端脚本编写的,但其运行还需要一定的客户端技术支持,如网页浏览器或浏览器上常用的一些插件(Java 或 Flash)。由于技术限制,早期在浏览器端采用 JavaScript、VBscript 等技术开发的 WebGame 多为策略型和简单图片型。随着 Flash 10 技术的支持,在浏览器端采用 Flash 或 Flex 开发的 WebGame 可获得类似客户端网络游戏的画面效果,但受限于 Flash 技术本身,WebGame 在处理大规模场景的地图、即时战斗、同屏角色效率问题上有很大的局限性。

另外,还有极少数基于 Shockwave、ActiveX 插件的 WebGame,由于难度较高,且限制较多、效果一般,所以使用者更少。

国内的 WebGame 也在 2006 年以全新面貌兴起,发展至今已进入成熟稳定期。随着国内网络带宽升级和技术的发展,目前利用 Unity3D 等引擎制作的 Web3DGame 也逐步发展起来。随着智能手机的发展和 4G 手机的普及,移动互联平台上的网络游戏将是下一个重要的增长点。

1.3 网络游戏服务器端介绍

网络游戏服务器端开发是网络游戏开发过程中非常重要的部分。玩家越多,游戏厂商越容易赚到钱;同时上线人数越多,则意味着服务器端压力就越大。网络游戏运营商迫切需要运行效率更高、运行状态更稳定、运行环境更安全的网络游戏服务器。

当前服务器端开发所使用的操作系统有 Windows Server 系统和免费开源的 Linux 操作系统。使用 Windows 操作系统开发服务器端的好处在于:Windows 有微软强有力的技术支持,操作界面交互较友好,操作方法容易掌握,在后期维护和周边支持方面也强于其他操作系统。使用 Linux 操作系统的好处在于:它是开源、免费的操作系统,故采用 Linux 作为服务器端操作系统的运营成本比采用 Windows 操作系统低,而且由于设计方式不同,Linux 操作系统运行

效率比 Windows 操作系统高，安全性也强于 Windows 操作系统。但是 Linux 操作系统不够人性化，无论是操作方式还是交互界面都比 Windows 操作系统稍差。此外，使用 Linux 操作系统开发服务器应用的编程人员也较少，这在某种程度上限制了 Linux 操作系统编程在程序员中的普及。

由于这两种操作系统各有优缺点，所以越来越多的网络游戏服务器开发都开始使用跨平台开发技术，使服务器端可运行于不同的操作系统上，这也可满足不同网络游戏运营商的需要。

网络游戏服务器端对性能要求很高，所以一般采用 C 或者 C++ 编程语言进行开发。一般网络游戏服务器端的开发流程分为三个阶段。

(1) 开发网络库，在操作系统提供的网络接口上封装自己的网络库（例如，在 Windows 平台下的 IOCP 是一个异步 I/O 的 API，可以作为大型网络架构的基础）或利用成熟的第三方开源的网络库（利用 ACE）。此为最重要的一步，因为后续逻辑实现的开发都要在此基础上进行。

(2) 制定通信协议，定义客户端与服务器端、服务器端与服务器端的通信规则。

(3) 实现具体的逻辑功能，按照不同的系统功能需求开发相应的模块，如登录游戏服务器、背包、商城和聊天系统等模块。

1.4 主流网络游戏技术简介

目前网络游戏技术大体分为客户端技术和服务器端技术，3D 游戏的客户端技术一般都是以 DirectX 或 OpenGL 作为底层的图形 API，封装后形成游戏引擎，如 Ogre、Unreal、BigWorld 等。由于各个公司对服务器端的要求不一样，大部分公司均采用 C++ 语言编写自主知识产权的网络游戏服务器端程序。

目前主流网络游戏服务器的架构通常采用客户端/服务器（C/S）结构。一般分为以下三种类型。

第一种类型是对等的客户端和服务端，在这种游戏服务器结构中，其中一个玩家的机器既充当服务器又充当客户端的角色，通常情况下服务器是由主导创建游戏的玩家担任的，称之为主机，其他的玩家通过网络搜索到相应的游戏并加入其中。《魔兽争霸》、《星际争霸》、《CS》等游戏都属于这种类型。

第二种类型是基于大厅（Lobby）的网络游戏，在这种游戏服务器结构中，大厅只负责把不同的客户端组织到一起，各个单独的客户端可以自行开始游戏。在进行游戏时，玩家通常只和自己加入的服务器发生交互。在数据库里只保存玩家的账户信息以及分数等数据。这种体系结构中的服务器端只是独立于游戏之外的一个辅助性软件，作用只是提供多个游戏服务房间的连接，供玩家选择。

第三种类型是真正意义上的 C/S 结构。客户端不再对数据进行逻辑处理，只作为一个收发装置从玩家那里接收操作信息，然后反馈给服务器，再由服务器处理后发回给客户端，在客户端通过图形化处理给玩家呈现出一个缤纷的游戏世界。如图 1-1 所示，数据库服务器上存有整个游戏世界的数据库，包括地图信息、人物信息等。玩家通过客户端连接到登录服务器，账号验证成功后得到游戏世界的所有信息。在这种体系结构中，客户端之间无法直接通信，所有客户端消息均必须先发送给服务器。客户端发送的消息经服务器分析、验证和处理后，才能被服务器转发给其他客户端。与单机游戏不同的是：在网络游戏中，所有玩家信息如人物等级、装备等都存放于服务器上。这些信息由服务器发送给客户端，客户端只用于显示信息。游戏逻辑也统一由服务器处理，然后把处理的结果发送给客户端。此做法的目的是保证游戏数据不会被恶意破坏，预防不法用户篡改游戏数据的恶劣行为。但是这种做法的问题是：随着用户数量的增加，服务器所承受的压力将越来越大，成为整个系统的瓶颈。最终，服务器的瘫痪将会造成整个网络游戏系统的瘫痪，给运营商带来巨大的损失。

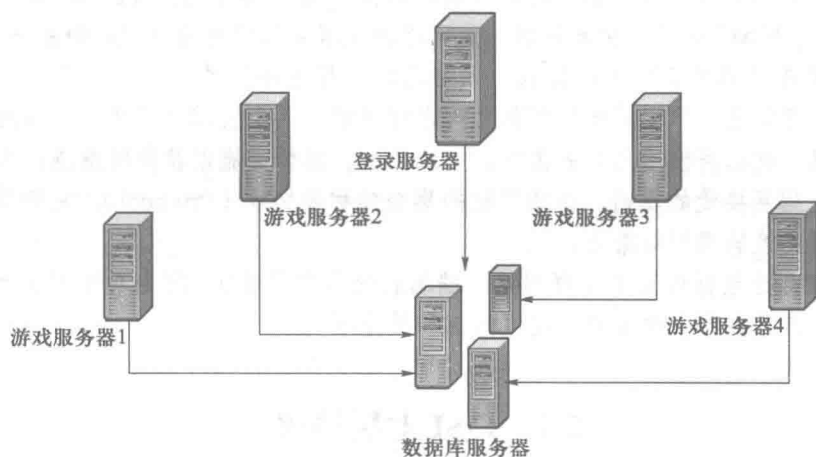


图 1-1 服务器端信息交互图

【小结】 本章主要介绍了网络游戏发展的历史、网络游戏服务端开发的基本知识和主流网络游戏技术，详细的内容将在后续章节中进行介绍。

习题 1

1. 简述网络游戏发展的历史。
2. 主流的网络游戏技术有哪些？

第 2 章 与网络编程有关的协议简介

本章将详细介绍网络编程的各个基础协议，包括 OSI 七层模型、TCP/IP 的分层模型以及 UDP、IP、ARP、ICMP 等几个主流协议，为网络游戏服务器端开发奠定理论基础。

网络编程需要了解计算机网络协议，网络游戏编程也不例外。共享计算机的资源以及在网上传换信息，都需要在不同系统的实体之间进行通信，两个实体或实体间对等的应用程序要成功地交换信息，就必须使用同样的语言。交流什么、怎样交流以及何时交流，都必须遵从有关实体间某种能相互接受的规则。这些规则的集合被称为协议（Protocol），它可以定义为两个实体间控制数据交换的规则的组合。

协议的关键部分包括语法（数据格式、编码、信号电平等）、语义（包括用于协调和进行差错处理的控制信息）和定时（包括速度匹配和排序等）。

2.1 OSI 七层协议

国际标准化组织（ISO）于 1983 年提出了著名的 ISO/IEC 7498 标准，它是一个试图使各种计算机在世界范围内互连为网络的标准框架，简称为 OSI 模型（Open System Interconnection Reference Model），即开放式系统互联参考模型。

OSI 将计算机网络划分为 7 层，如表 2-1 所示。

表 2-1 OSI 七层模型与网络协议对应表

OSI 七层网络模型	对应网络协议
应用层（Application）	TFTP、FTP、NFS、WAIS
表示层（Presentation）	Telnet、Rlogin、SNMP、Gopher

续表

OSI 七层网络模型	对应网络协议
会话层 (Session)	SMTP、DNS
传输层 (Transport)	TCP、UDP
网络层 (Network)	IP、ICMP、ARP、RARP、AKP、UUCP
数据链路层 (Data Link)	FDDI、Ethernet、Arpanet、PDN、SLIP、PPP
物理层 (Physical)	IEEE 802.1A、IEEE 802.2~IEEE 802.11

1. 应用层

应用层 (Application Layer) 是最靠近用户的 OSI 层, 主要负责为软件提供接口, 使程序能使用网络服务。简单地说, 就是能与应用程序进行沟通, 以达到将其展示给用户的目的。术语“应用层”并不是指运行在网络上的某个特别的应用程序。应用层提供的服务包括文件传输、文件管理及电子邮件的信息处理。常见的有 HTTP、HTTPS、FTP、Telnet、SSH、SMTP 和 POP3 等协议。

2. 表示层

表示层 (Presentation Layer) 主要为不同用户提供数据和信息的语法转换和格式化, 同时, 也能提供数据的解密与加密、压缩与解压等功能, 如常见的系统口令处理, 如果在发送端对用户的账户数据进行了加密, 则网络接收端的表示层将对所接收的数据进行解密。此外, 表示层还需要对图片和文件格式信息进行解码和编码。

3. 会话层

会话层 (Session Layer) 负责为通信双方制定通信方式。它在网络的两个节点之间建立和维持通信, 并使会话保持同步, 还决定通信是否被中断及通信中断后从何处重新发送。

4. 传输层

传输层 (Transport Layer) 是 OSI 模型中最重要的一层, 当两台计算机通过网络进行数据通信时, 它是第一个端到端的层次, 可起到缓冲作用。当网络层的服务质量不能满足要求时, 它将提高服务以满足高层的要求; 当网络层服务质量较好时, 它所需要做的工作很少。此外, 它还处理端到端的差错控制和流量控制等问题, 最终为会话提供可靠、无误的数据传输。传输层包括 TCP、UDP、SPX 等协议。

5. 网络层

网络层 (Network Layer) 的主要功能是将网络地址翻译成对应的物理地址, 它通过综合考虑发送优先权、网络拥塞程度、服务质量及可选路由的花费, 确定网络中两个节点之间的最佳路径。此外, 它还可实现拥塞控制、网际互连等功能。网络层包括 IP、IPX、RIP、OSPF 等协议。

6. 数据链路层

数据链路层 (Data Link Layer) 是 OSI 模型的第二层, 它控制网络层与物理层之间的通信, 其主要功能是确保在不可靠的物理线路上进行可靠的数据传递。为了保证传输的可靠性, 从网络层接收到的数据被分割成特定的可被物理层传输的帧。它的主要作用包括: 物理地址寻址、数据的成帧、流量控制、数据的检错、重发等。数据链路层独立于网络层和物理层, 工作时无须关心计算机是否正在运行软件或执行其他操作。有些连接设备, 如交换机, 要对帧解码并用帧信息将数据发送到正确的接收方, 所以它们在数据链路层工作。数据链路层包括 SDLC、HDLC、PPP、STP、帧中继等协议。

7. 物理层

物理层 (Physical Layer) 作为 OSI 模型的第一层, 其主要功能是为数据端设备提供传送数据的通路, 数据通路可以是一个物理媒体, 也可以由多个物理媒体连接而成。一次完整的数据传输过程包括激活物理连接、传送数据和终止物理连接等步骤。物理层规定了激活、维持、关闭通信端点之间的机械特性、电气特性、功能特性以及过程特性。尽管物理层不提供纠错服务, 但它能够设定数据传输速率并监测数据出错率。若网络出现物理问题, 如电线断开, 将影响物理层。

OSI 七层模型有效地解决了不同网络体系互连时所遇到的兼容性问题, 其最大优点是将服务、接口和协议这三个概念明确地区分开来。服务说明某一层为上一层提供一些什么功能, 接口说明上一层如何使用下层的服, 而协议涉及如何实现本层的服务。各层之间具有很强的独立性, 互连网络中各实体采用什么样的协议是有限制的, 只需向上提供相同的服务且不改变相邻层接口即可。OSI 七层模型减轻了网络的复杂程度, 一旦网络发生故障, 可迅速定位故障所处层次, 便于查找和纠正错误; 通过在各层上定义标准接口, 使同属一层的不同网络设备间能实现互操作; 还保证了各层之间的相对独立; 高层协议可运行在多种低层协议上, 提高了网络效率, 因为每次更新只需在一个层次上进行, 不受整体网络的制约, 所以 OSI 七层模型的出现有效刺激了网络技术革新, 它是网络技术发展的原动力。

2.2 TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) 即传输控制协议/因特网互连协议, 又叫网络通信协议, 是 Internet 最基本的协议, 也是 Internet 国际互联网络的基础, 由网络层的 IP 和传输层的 TCP 组成。TCP/IP 定义了电子设备连入因特网、数据在设备间传输的标准。协议采用了 4 层的层级结构, 称为 DARPA 模型, 这 4 层从上到下依次是: 应用层 (Application)、传输层 (Transport)、网络层 (Network)、链路层 (又叫网络访问层, Network Access)。每一层都呼叫其下层所提供的网络来完成自己的需求。TCP 负责发现传输的问题,

一旦出现问题就发出信号，要求重新传输，直到所有数据安全正确地传输到目的地。IP 则为因特网中每台计算机规定一个地址。

TCP/IP 的 4 个层次如表 2-2 所示。

表 2-2 TCP/IP 协议的四个层次

TCP/IP 协议层次	对应网络协议
应用层	Telnet、FTP 和 E-mail 等
传输层	TCP 和 UDP
网络层	IP、ICMP 和 IGMP
链路层	设备驱动程序及接口卡

1. 应用层

应用层协议专门为用户提供应用服务，它建立在网络层协议之上，包含所有与应用程序协同工作、利用基础网络交换数据的专用数据协议。该层协议用来处理网络相关程序与其他程序间的网络通信。通过该层协议处理，数据被编码成标准协议的格式。

一些特定的程序运行在这个层上，它们为用户应用直接提供服务支持。这些程序与其对应的协议包括 HTTP（万维网服务）、FTP（文件传输）、SMTP（电子邮件）、SSH（安全远程登录）、DNS（域名系统）及许多其他协议。

从应用程序传来的数据一旦被编码成一个标准的应用层协议，它将被传送到 IP 栈的下一层。

2. 传输层

传输层主要提供应用程序间通信的协议，它负责两部分工作：格式化信息流和提供可靠传输。它能够解决诸如端到端可靠性和确保数据按照正确顺序到达之类的问题。在 TCP/IP 协议簇中，传输协议也包含数据应送达的目的地站点信息。

传输层协议主要有 TCP（Transmission Control Protocol，传输控制协议）和 UDP（User Datagram Protocol，用户数据报协议）。

3. 网络层

网络层主要负责单一网络上的数据通信，其主要功能包括三方面：第一，处理来自传输层的分组发送请求，收到请求后，将分组装入 IP 数据报，填充报头，选择去往信宿机的路径，然后将数据报发往适当的网络接口；第二，处理输入数据报，首先检查其合法性，然后进行寻径——若该数据报已到达信宿机，则去掉报头，将其余部分交给适当的传输协议，若该数据报尚未到达信宿，则转发该数据报；第三，处理路径、流控、拥塞等问题。

网络层包括 IP（Internet Protocol，因特网互连协议）、ICMP（Internet Control Message

Protocol, 因特网控制报文协议)、ARP (Address Resolution Protocol, 地址转换协议)、RARP (Reverse ARP, 反向地址转换协议)。

IP 是网络层的核心, 通过路由选择将下一站路由器 IP 封装后交给接口层。IP 数据报是无连接服务。

一些 IP 承载的协议, 如 ICMP (用来发送关于 IP 发送的诊断信息) 和 IGMP (用来管理多播数据), 位于 IP 层之上但是却可以完成网络层的功能, 这表明了因特网和 OSI 模型之间的不兼容性。

4. 链路层

链路层又称为网络访问层, 主要负责接收 IP 数据报并通过网络发送它, 或从网络上接收物理帧, 抽出 IP 数据报, 交给 IP 层。实际上它并不是因特网协议簇中的一部分, 但它是数据包从一个设备的网络层传输到另外一个设备的网络层的方法, 可在网卡的软件驱动程序中控制该过程。

常见的链路层协议有 Ethernet 802.3、Token Ring 802.5、X.25、Frame Relay、HDLC、PPP、ATM 等。

2.3 UDP

UDP (User Datagram Protocol) 即用户数据报协议, 其主要作用是将网络数据流量压缩成数据报形式, 提供面向事务的简单信息传送服务。UDP 提供面向无连接的、不可靠的数据报投递服务。当使用 UDP 传输信息流时, 用户应用程序必须负责解决数据报丢失、重复、排序及差错确认等问题。

UDP 的主要特征如下。

(1) UDP 传送数据前并不与对方建立连接, 即 UDP 是无连接的。在传输数据前, 发送方和接收方相互交换信息使双方同步。

(2) UDP 不对收到的数据进行排序, UDP 报文首部并没有关于数据顺序的信息, 而且报文不一定按顺序到达, 所以接收端无法排序。

(3) UDP 对接收到的数据报不发送确认信号, 发送端不知道数据是否被正确接收, 也不会重发数据。

(4) UDP 传送数据比 TCP 速度快, 系统开销也小。

UDP 比较简单, UDP 头包含了源端口、目的端口、消息长度和校验和等很少的字节。由于 UDP 比 TCP 简单、灵活, 常用于少量数据的传输, 如域名系统 (DNS) 及简单文件传输系统 (TFTP) 等。TCP 则适用于对可靠性要求高, 但对实时性要求不高的应用, 如文件传输协议 FTP、超文本传输协议 HTTP、简单邮件传输协议 SMTP 等。