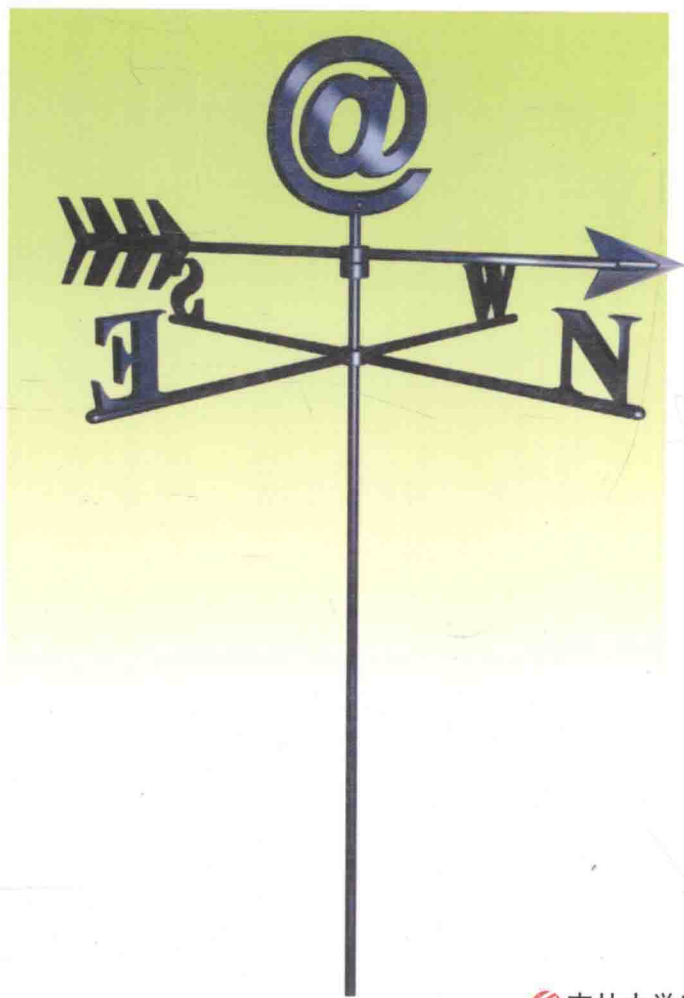


INTERNET 

网络程序设计

胡亮 李强 康健 / 编著



 吉林大学出版社

网络程序设计

胡亮 李强 康健 编著

2003年9月

网络程序设计

胡 亮 李 强 康 健 编著

责任编辑：杨 枫

封面设计：李立嗣

吉林大学出版社出版

吉林大学出版社发行

(长春市明德路3号)

吉林大学教育印刷厂印刷

开本：787mm×1092mm 1/16

2003年9月第1版

印张：22.375 字数：300千字

2003年9月第1次印刷

ISBN 7-5601-2662-6/TP·119

定价：31.20元

前 言

网络程序设计是编写通过计算机网络与其它程序进行通信的程序。本书在介绍了计算机网络的基本原理和技术以及 TCP/IP 协议之后,重点针对套接口(socket)网络程序设计方法进行了论述。主要内容有:计算机网络的基本原理, TCP/IP 协议, 顾客服务员程序设计, 名字与地址转换, 机内进程间通信, 广播和多播, 带外数据, 原始套接口, 线程程序设计, RPC 远程过程调用, 流程程序设计。

本书是在参考国内外许多优秀教材的基础上编写而成,内容由浅入深,适合于不同层次的学生使用,可作为高等院校本科生、研究生教材使用。本书在出版前作为教材在吉林大学使用过,在教学过程中经过了不断的完善。本教材具有丰富的图解,力求做到既通俗易懂,又具有一定的理论深度。

由于作者水平所限,书中难免还存在一些缺点和错误,敬请读者批评指正。

目 录

第一章 网络基础.....	1
1.1 概述.....	1
1.1.1 计算机网络定义和应用.....	1
1.1.2 计算机网络组成和结构.....	2
1.2 OSI 参考模型.....	3
1.2.1 网络分层体系结构.....	3
1.2.2 分层的原则和目标.....	3
1.2.3 ISO OSI/RM.....	4
1.3 通信方式与服务.....	6
1.3.1 层间通信与对等层间通信.....	6
1.3.2 服务和数据单元.....	7
1.3.3 服务原语.....	8
1.3.4 面向连接的和无连接的服务.....	10
1.4 传输层原理.....	12
1.4.1 传输层概述.....	12
1.4.2 传输层地址.....	13
1.4.3 传输层复用功能.....	13
1.4.4 可靠传输.....	14
1.4.5 传输层流量控制.....	17
1.4.6 传输连接.....	19
1.4.7 传输层提供的服务质量.....	22
1.4.8 传输服务用户间的通信连接.....	22
1.5 TCP/IP 协议族.....	24
1.5.1 TCP/IP 层次.....	24
1.5.2 RFC 文档.....	26
1.6.1 Unix 标准.....	26
1.6.2 体系结构.....	27
1.6.3 协议无关性.....	28
第二章 TCP 协议和 UDP 协议.....	29
2.1 概述.....	29
2.2 UDP: 用户数据报协议.....	29
2.2.1 UDP 报头.....	30
2.2.2 UDP 校验和.....	30

2.3	TCP: 传输控制协议	31
2.3.1	TCP 提供的服务	31
2.3.2	TCP 的首部	32
2.3.3	TCP 连接的建立和释放	35
2.3.4	最大报文段长度	36
2.3.5	TCP 的半关闭	37
2.3.6	同时关闭	38
2.3.7	TCP 选项	38
2.3.8	TCP 的性能	39
2.3.9	T/TCP: 为事务用的 TCP 扩展	42
2.4	传输层端口号	44
2.4.1	寻址	44
2.4.2	套接字接口	46
2.5	面向连接编程模型	48
2.5.1	面向连接编程模型使用的主要系统调用	48
2.5.2	面向连接服务流程	50
2.5.3	面向连接服务程序例子	52
2.6	面向无连接编程模型	55
2.6.1	发送和接收数据	55
2.6.2	面向无连接服务流程	55
2.6.3	面向无连接服务程序例子	56
2.6.4	两种模型的区别	60
2.7	进程通信原语	60
2.7.1	报文传递	61
2.7.2	远程过程调用	62
2.7.3	进程通信原语的实现问题	65
第三章	IP 协议	66
3.1	IP 地址	66
3.2	IP 报文格式	67
3.3	IP 路由	70
3.3.1	IP 路由选择	70
3.3.2	子网编址与子网掩码	72
3.3.3	IP 的未来发展	73
3.4	ARP 协议和 RARP 协议	74
3.4.1	ARP(地址解析协议)	74
3.4.2	RARP: 反向地址解析协议	77
3.5	BOOTP: 引导协议	80
3.5.1	BOOTP 重发策略	80
3.5.2	BOOTP 的分组格式	80

3.5.3	BOOTP 端口号	83
3.5.4	BOOTP 穿越路由器	84
3.6	ICMP: Internet 控制报文协议	84
3.6.1	ICMP 报文格式和类型	84
3.6.2	ICMP 地址掩码请求与应答	87
3.6.3	ICMP 时间戳请求与应答	87
3.6.4	ICMP 报文的 BSD4.4 实现	88
3.7	路由套接口编程	89
3.7.1	数据链路套接口地址结构	90
3.7.2	读和写	91
3.7.3	sysctl 操作	92
3.7.4	接口名和索引函数	96
第四章	顾客服务员程序设计	104
4.1	网络服务模式	104
4.1.1	顾客/服务器应用	105
4.1.2	顾客/服务器应用程序分类	105
4.1.3	三层顾客/服务器结构	108
4.1.4	中间件	108
4.1.5	文件 cache 的一致性	109
4.1.6	服务员类型	111
4.2	标准 Internet 服务和常见的应用	111
4.2.1	标准 Internet 服务	111
4.2.2	常见的 Internet 应用	112
4.3	并发服务器	112
4.3.1	套接口对	112
4.3.2	并发服务器	113
4.3.3	缓冲区大小及其限制	115
4.3.4	TCP 发送	116
4.3.5	UDP 发送	117
4.4	网络服务员工作模式	118
4.4.1	TCP 迭代服务器程序	118
4.4.2	TCP 并发服务器程序	119
4.4.3	BSD 上的实现	122
4.5	守护进程	123
4.5.1	守护进程的原理	123
4.5.2	syslog 守护进程	124
4.5.3	syslog 守护函数	125
4.5.4	daemon_init 函数	127
4.5.5	避免守护进程本身的多个副本互相干扰	129

4.6 Inetd 超级服务员	129
第五章 名字与地址转换	134
5.1 域名系统	134
5.1.1 资源记录	134
5.1.2 解析器和名字服务器	135
5.1.3 DNS 替代方法	136
5.2 gethostbyname 函数	136
5.3 RES_USE_INET6 解析器选项	139
5.4 与名字和地址有关的常用函数	141
5.4.1 gethostbyaddr 函数	141
5.4.2 uname 函数	142
5.4.3 gethostname 函数	143
5.4.4 getservbyname 和 getservbyport 函数	143
5.4.5 其它网络相关信息	146
5.5 套接字选项	146
5.5.1 获取和设置套接口选项	147
5.5.2 通用套接口选项	150
5.5.3 IPv4 套接口选项	155
5.5.4 ICMPv6 套接口选项	157
5.5.5 IPv6 套接口选项	157
5.5.6 TCP 套接口选项	159
5.6 fcntl 函数	160
第六章 机内进程间通信	162
6.1 引言	162
6.2 信号	162
6.2.1 Linux 系统支持的信号	163
6.2.2 信号的捕获和处理	164
6.2.3 信号的处理	165
6.2.4 信号和系统调用	166
6.2.5 sigsetjmp 和 siglongjmp 函数	166
6.2.6 信号阻塞	168
6.2.7 发送信号	168
6.2.8 异步信号安全函数	170
6.3 管道	170
6.3.1 管道分类	171
6.3.2 管道技术的实现	171
6.3.3 管道程序设计	172
6.3.4 管道的容量	174
6.3.5 关闭管道	175

6.3.6 非阻塞读和写	176
6.3.7 使用 select 调用处理多路管道	178
6.3.8 FIFO 或命名管道	180
6.4 共享内存	181
6.4.1 共享内存原理	181
6.4.2 共享内存技术的实现	182
6.4.3 共享内存的操作函数	182
6.5 套接字	183
6.5.1 套接字通信原理	183
6.5.2 实现过程	184
6.5.3 socketpair 函数	184
6.5.4 性能测试	186
6.5.5 结论	187
第七章 广播和多播	188
7.1 概述	188
7.2 广播	189
7.2.1 广播的用途	189
7.2.2 受限广播	190
7.2.3 指向网络的广播	190
7.2.4 指向子网的广播	190
7.2.5 指向所有子网的广播	191
7.2.6 广播的例子	191
7.2.7 IP 分片和广播	192
7.3 多播	192
7.3.1 多播组地址	192
7.3.2 多播组地址到以太网地址的转换	193
7.4 单播与广播的比较	194
7.5 局域网上的多播与广播比较	196
7.6 广域网上的多播	198
7.7 多播套接口选项	200
7.8 虚拟网络	204
7.8.1 概述	204
7.8.2 Mbone	204
7.8.3 6bone	206
第八章 带外数据	208
8.1 概述	208
8.2 TCP 带外数据	208
8.2.1 TCP 带外数据的发送和接收	208
8.2.2 TCP 带外数据处理例子	210

8.2.3 TCP 带外数据总结	213
8.3 socketmark 函数	214
8.4 顾客服务器心博函数	215
第九章 原始套接口和数据链路访问	220
9.1 概述	220
9.2 原始套接口创建	220
9.3 原始套接口输出	221
9.3.1 原始套接口在 IPv4 和 IPv6 上的差异	222
9.3.2 IPV6_CHECKSUM 套接口选项	222
9.4 原始套接口输入	223
9.5 ping 程序	225
9.6 ICMP 消息守护进程	234
9.7 数据链路访问	236
9.7.1 BPF: BSD 分组过滤器	236
9.7.2 DLPI: 数据链路提供者接口	238
9.7.3 Linux: SOCK_PACKET	238
9.7.4 libpcap: 分组捕获函数库	239
9.7.5 小结	239
第十章 线程程序设计	240
10.1 概述	240
10.1.1 线程	240
10.1.2 线程状态	242
10.1.3 线程的运行	242
10.1.4 线程的分类	243
10.2 线程间的同步: 互斥锁	246
10.3 Solaris 的线程	247
10.3.1 多线程体系结构	247
10.3.2 进程结构	248
10.3.3 线程的执行	249
10.4 Linux 系统的进程和线程管理	250
10.4.1 Linux 进程	250
10.4.2 Linux 线程	251
10.5 使用线程技术进行网络程序设计	251
10.5.1 基本线程函数	251
10.5.2 线程安全函数	253
10.5.3 互斥锁与条件变量	254
10.5.4 TCP 并发服务器程序, 每个顾客一个线程	256
10.5.5 Web 客户与同时连接	257
第十一章 RPC 远程过程调用	262

11.1 XDR 标准.....	262
11.1.1 数据结构传输的问题.....	262
11.1.2 XDR 标准.....	263
11.2 远程过程调用 (RPC) 的原理	269
11.2.1 分布式数据处理方法.....	269
11.2.2 RPC 系统组成及特点	270
11.2.3 实现 RPC 要解决的问题	272
11.3 RPC 的实现.....	273
11.3.1 本地函数调用的过程.....	273
11.3.2 远程过程的标识.....	274
11.3.3 端口的动态映射.....	275
11.3.4 RPC 报文.....	276
11.3.5 RPC 开发工具.....	277
11.3.6 RPC 设计的原则	278
11.4 SUN RPC	279
11.5 加密技术	282
11.5.1 传统加密方法	283
11.5.2 公开密钥加密方法.....	286
第十二章 流.....	291
12.1 流 (STREAMS) 概述.....	291
12.1.1 流.....	291
12.1.2 流的基本操作	293
12.1.3 STREAMS 的组成.....	294
12.1.4 多路复用	297
12.1.5 STREAMS 的优点	298
12.2 STREAMS 机制.....	299
12.2.1 STREAMS 系统调用	299
12.2.2 流的构造.....	300
12.2.3 添加与删除模块	301
12.2.4 关闭流.....	302
12.2.5 STREAMS 处理例程	302
12.3 消息.....	303
12.3.1 消息类型.....	304
12.3.2 消息结构	305
12.3.3 发送和接收消息	307
12.3.4 流首处理的控制	309
12.3.5 消息队列和消息优先级	310
12.4 服务界面	317
12.4.1 服务界面的优点	318

12.4.2 服务界面库	318
附录 A 头文件	320
附录 B 英文缩写词汇	332
参考文献	338

第一章 网络基础

1.1 概述

1.1.1 计算机网络定义和应用

计算机网络是计算机技术和通信技术相结合的产物。自从 1946 年第一台电子计算机 ENIVAC 问世以来,由于计算机网络技术和软件技术的不断发展,人们使用计算机的方式有了根本的改变,由多人通过终端使用一台计算机到现在一人通过计算机网络使用多台计算机。

计算机网络是由多个独立的计算机通过通信线路和通信设备互连起来的系统,以实现彼此交换信息和共享资源的目的。

计算机网络具有以下功能:

数据通信,网络系统中各相连的计算机能够相互传送数据信息,使相距很远的用户之间能够直接交换数据。

资源共享,网络中的软件,硬件资源如外部设备,文件系统和数据等可为多个用户所共享。

并行和分布式处理,在计算机网络中用户可根据问题的性质和要求选择网内最合适的资源来处理。对于综合性的大问题,可以采用合适的算法,将任务分散到不同的计算机上进行分布和并行处理。

提高可靠性,由于控制、数据、软件和硬件的分散型(不存在集中环节),资源冗余以及结构上可动态重组提高了可靠性。

好的可扩充性,随着用户需求的增长,包括性能方面和功能方面的增长,只需增加新节点数,而不必替换整个系统。由于可扩充性,可以避免较大的初始投资,以及用多个微小型机可以代替一个大型主架机,获得很好的性能价格比。

计算机的应用领域:

随着计算机网络的日益普及,它已经应用在各个领域中,我们在日常生活中常见的采用计算机的服务项目,如银行的提款机,销售点的终端,支票和发票的核实等都依赖于计算机网络。下面是计算机网络应用的一些典型领域:

服务业:通过计算机网络系统进行酒店和航空公司的在线订票、订房,远程购物等等。

金融服务业:现在的金融服务都依赖于计算机网络,如外汇汇兑和投资服务,电子资金转账服务等。

企业管理:通过网络信息系统对企业生产、销售、财务等方面进行管理。

制造业:计算机网络在制造业的多个方面包括制造过程本身,都有应用。如计算机辅助

设计(CAD)和计算机辅助制造(CAM),这两种业务都允许同时有许多用户在同一个项目上工作。

电子消息传递:最广泛的应用如电子邮件。

信息服务:如电子公告板和WWW站点。

实时信息传递:如音频和视频会议,视频点播,远程教学等。

总之,计算机网络的应用已经深入到社会和经济生活的各个方面。随着计算机网络应用的不断推广和普及,网络软件的设计和开发越来越流行,目前绝大多数系统软件(例如:操作系统 UNIX、Linux、Windows 系列等)和应用软件是网络版的。掌握网络程序设计的原理和方法对于设计和开发网络应用程序是十分重要的。

1.1.2 计算机网络组成和结构

计算机网络要完成数据处理和数据通信两大功能,同这两大功能对应,它的结构可以分成两大部分:负责数据处理的计算机和终端;负责数据通信的通信控制处理机 CPP(Communication Control Processor)、通信线路。计算机网络从逻辑功能上可以分为两个子网:资源子网和通信子网,其结构如图 1-1。

资源子网由计算机系统、网络终端、外部设备(如打印机等)、各种软件资源与数据资源组成。资源子网负责全网的数据处理业务,向网络用户提供各种网络资源与网络服务。

通信子网由网络通信控制处理机(如存储转发处理机、集中器、网络协议变换器、报文分组装配/拆卸设备等),通信线路与其他通信设备组成。通信子网完成全网数据传输、转发等通信处理工作。

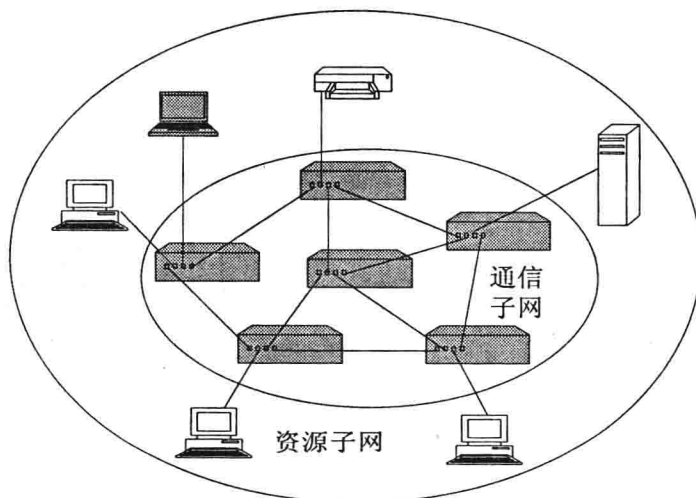


图 1-1 资源子网和通信子网

1.2 OSI参考模型

1.2.1 网络分层体系结构

网络协议。计算机网络由若干个相互连接的节点组成，在这些节点之间要不断地进行数据交换。要进行正确的数据传输，每个节点就必须遵守一些事先约定好的规则，这些规则就是网络协议。网络协议是在主机与主机之间、主机与通信子网之间或子网中各通信节点之间的通信而使用的，是通信双方必须遵守的，事先约定好的规则、标准或约定。

一个网络协议主要由以下三个要素组成；

(1) 语法。即数据与控制信息的结构或格式。如：数据格式、信号电平等规定。

(2) 语义。即需要发出何种控制信息，完成何种动作，以及做出何种应答。包括用于调整和进行差错处理的控制信息。

(3) 时序(同步)。即事件实现顺序的详细说明，包括速度匹配和顺序。

对于复杂的网络协议，其结构最好是采用层次式的。大体来讲，计算机网络中的协议采用层次结构，分层有以下几个好处：

(1) 各层之间是独立的。一个层次并不需要知道它下面的一层是如何实现的，而仅需知道该层通过层间的接口所提供的服务，及调用此服务所需要的格式和参数。

(2) 灵活性好。当任何一层发生变化时，只要接口关系保持不变，则其他层次均不受影响。

(3) 结构上可分隔开。各层可以采用最合适的技术来实现。

(4) 易于实现和维护。这种结构使得一个复杂系统的实现和调试变得简单。因为整个系统已被分解为若干个小的易于处理的部分。

(5) 有利于标准化工作。每一层的功能以及向其他层所提供的服务都有了精确的说明。因此对于标准化工作是十分方便的。

我们将计算机网络的各个层次及其相关协议的集合，称为网络体系结构。网络体系结构是计算机网络所应完成的功能的精确定义。体系结构与其具体实现是不同的，网络体系结构中的功能无论是用硬件实现，还是用软件实现，则是一个遵循这种体系结构的实现问题。由此可见，网络体系结构是抽象的，是对功能的精确描述。而实现是具体的，是真正运行的硬件和软件。

1.2.2 分层的原则和目标

分层虽然是一个处理复杂问题的有效方法，但分层本身并不是一项简单的工作。目前还不存在一个最佳的层次划分方法。下面介绍分层的一些主要原则：

(1) 当需要有一个不同等级的抽象时，就应当有一个相应的层次；

- (2) 每层的功能应当是十分明确的;
- (3) 层与层的边界应当选择合适,使通过这些边界的信息量尽量少些;
- (4) 层数太少会使每一层的协议太复杂,但层数太多则在描述和综合各层的系统任务时会有较大的困难。

现代的计算机网络是围绕着分层协议或分层功能的概念来设计的。这些技术的发展是为了实现如下目标:

- (1) 把一个复杂的网络合乎逻辑地分为若干个较小的,比较容易理解的部分(层);
- (2) 在各个网络功能之间提供标准接口,例如软件之间的标准接口;
- (3) 网络中各节点执行功能的对称性,网络中各个节点的相同层执行相同的功能;
- (4) 为预测和控制网络逻辑(软件或微码)的修改提供手段;
- (5) 为网络设计者、开发者讨论网络功能时提供一种标准的语言。

世界上第一个网络体系结构是美国 IBM 公司在 1974 年提出的,取名为系统网络体系结构(System Network Architecture—SNA)。凡是遵循 SNA 标准的设备就称为 SNA 设备,这些 SNA 设备可以很方便的进行互连。此后许多计算机公司纷纷建立了自己的网络结构。这些体系结构大同小异,均采用分层技术,但各有特点以适合本公司生产的计算机网络,比较具名的网络体系结构还有:

DNA(Digital Network Architecture)—DEC 公司;

BNA(Burroughs Network Architecture)—宝莱公司;

DSA(Distributed Systems Architecture)—Honeywell 公司。

为了使不同厂家的网络产品能够相互通信,需要制定一个国际通用的标准的网络体系结构。

1.2.3 ISO OSI/RM

国际标准化组织(International Standards Organization—ISO)经过反复地研究,在已有的网络体系结构(如 DNA、SNA 等)的基础上,制定了开放系统互连参考模型(Open Systems Interconnection Reference Model—OSI/RM)。供设计、实现和应用各种计算机网络参考。一个系统是开放的,是指它可与世界上任何地方的遵守相同标准的其他任何系统通信。在 OSI/RM 中采用了如图 1-2 所示的 7 个层次的体系结构:

图中的各层协议是通信双方在通信过程中的约定,规定有关部件在通信过程中的操作以保证正确地进行通信。各层的主要功能如下:

1. 物理层:规定在一个节点内如何把计算机连接到通信介质上,规定了机械的、电气的功能;该层负责建立、保持和拆除物理链路;规定如何在此链路上传送原始比特流;比特如何编码,使用的电平,极性,连接插头插座的插脚如何分配等。所以在物理层数据的传送单位是比特(bit)。
2. 数据链路层:它把相邻两个节点间不可靠的物理链路变成可靠的无差错的逻辑链路,包括把原始比特流分帧、排序、设置检错、确认、重发、流控等功能;数据链路层传输信息的单位是帧(frame),每帧包括一定数量的数据和一些必要的控制信息,在每

帧的控制信息中,包括同步信息、地址信息、差错控制信息、流量控制信息等;同物理层相似,数据链路层负责建立、维护和释放数据链路。

3. 网络层:它连接网络中任何两个计算机节点,从一个节点上接收数据,正确的传送到另一个节点;在网络层,传送的信息单位是分组或包(packet)。网络层的主要任务是要选择合适的路由和交换节点,透明地向目的站交付发送站所发的分组或包,这里的透明表示收发两端好像是直接连通的。另外网络层还要解决网络互连、拥挤控制和记账等问题。

上述三层组成了所谓的通信子网,用户计算机连接到此子网上。通信子网负责把一个地方的数据可靠地传送到另一个地方,但并未实现两个地方主机上进程之间的通信。通信子网的主要功能是面向通信的。

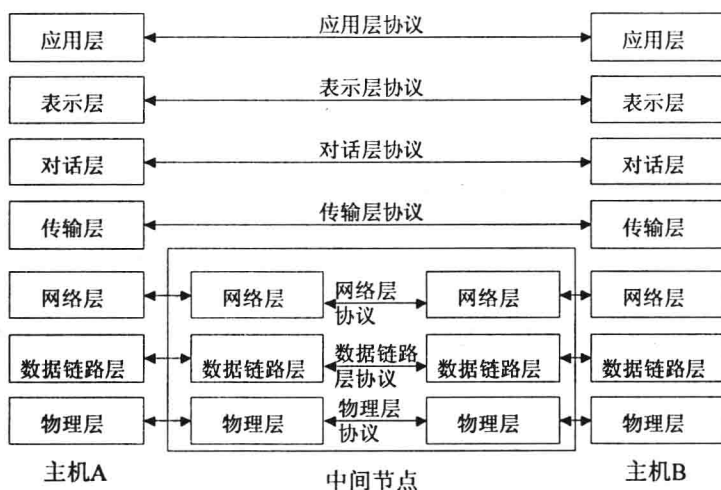


图 1-2 ISO OSI/RM

4. 传输层:真正地实现了端—端间通信,把数据可靠地从一方的用户进程或程序送到另一方的用户进程或程序。这一层的控制通常由通信两端的计算机完成,中间节点一般不提供这一层的服务,这一层的通信与通信子网无关。从这一层开始的以上各层全部是针对通信的最终源端—目的端计算机的进程之间的。传输层传送的信息单位是报文(message)。

传输层向上一层提供一个可靠的端—端的服务,使上一层看不见下面几层的通信细节。正因为如此,传输层成为网络体系结构中最关键的一层。对于传输层的功能,主要在主机内实现。而对于物理层、数据链路层以及网络层的功能均在报文接口机中实现。对于传输层以上的各个层次的功能通常在主机中实现。

5. 对话层:又称会话层。它允许两个计算机上的用户进程建立对话连接,双方相互确认身份,协商对话连接的细节;它可管理对话是双向同时进行的,还是任何时刻只能一个方向进行。在后一种情况下,对话层控制哪一方有权发送数据;对话层还提供同步机制,在数据流中插入同步点机制,在每次网络出现故障后可以仅重传最近一个同步点以后的数据,而不必从头开始。