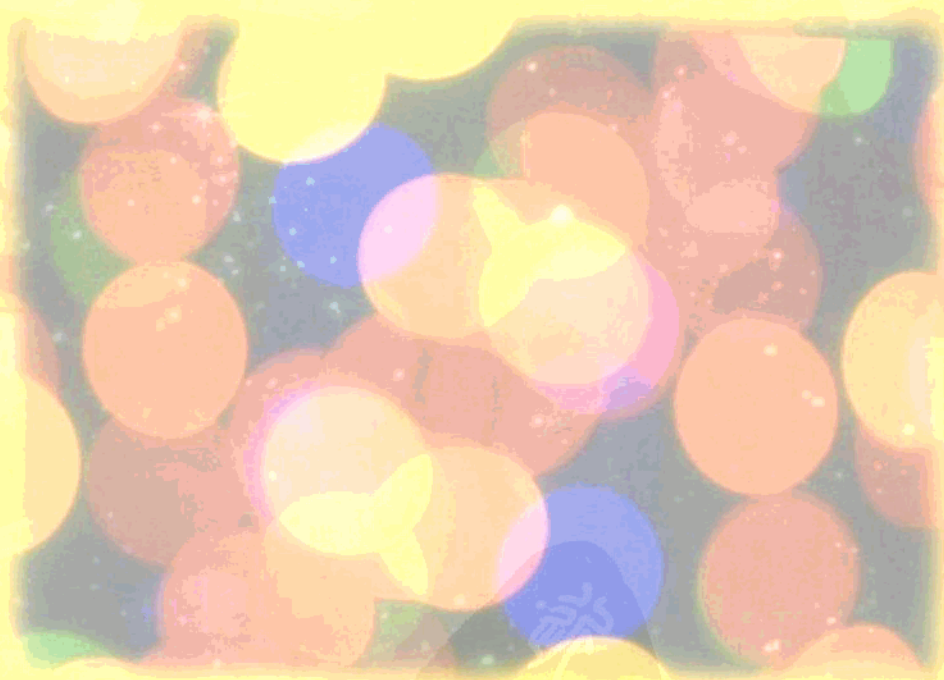


哈尔滨工程大学出版社电脑丛书

Internet 程序设计

JAVA & HTML

杨文新 编著



哈尔滨工程大学出版社

序 言

互联网(Internet)是最近最热门的话题之一,由于它的影响范围并不局限在计算机的领域,而已逐渐融入我们的生活空间内。因此设计更多适合的程序及系统于互联网上,就显得格外的重要。

本书以互联网程序设计(Internet Programming)为依归,希望能让读者有全盘性的了解,由常规的网络程序设计(Network Programming)概念,带入当前的互联网的环境里。本书的章节安排想法是希望能在很少的篇幅内,带给读者由最基础的计算机网络概念至 TCP/IP 网络程序设计,进而到当前互联网上的首页(Home Page)制作及 Java 程序两种设计方法,全书共计五个章节及附录。

第一章计算机网络简介,介绍要进行互联网程序设计所需要的计算机网络的一些概念。

第二章 TCP/IP 简介,介绍有关于互联网里最重要的 TCP/IP 网络结构,并且对此种网络的通信协议(Protocol)作一番说明。

第三章 Socket 程序设计,介绍 TCP/IP 网络的程序设计方法,也就是 Socket 应用程序接口(API)的使用方式,这也是进行互联网程序设计的最基本的方法。

第四章全球信息网(WWW),介绍当前互联网上所流行的最重要的信息网络,并且对于在全球信息网上进行的一种互联网程序设计方式——首页制作,加以说明。

第五章 Java 程序设计,介绍另外一种配合全球信息网的互联网程序设计方式——Java 程序设计,这也是当前互联网程序设计最重要的一种趋势,本章详尽的对 Java 语言的各个部分加以说明。

附录部分,则收录了全球信息网相关资料的 URL(网址),以作为补充本书之不足部分。

本书匆匆完稿,如有疏漏错误之处,欢迎指教,谢谢!

杨文新

目 录

1. 计算机网络简介	1
1.1 什么是计算机网络	1
1.2 为什么需要计算机网络	1
1.3 计算机网络的连接方式	2
1.3.1 星形连接方式	2
1.3.2 环形连接方式	2
1.3.3 总线连接方式	3
1.4 计算机网络结构	4
1.4.1 计算机网络结构的设计	4
1.4.2 层次式计算机网络结构	4
1.4.3 通信协议	5
1.5 三层式计算机网络结构	5
1.5.1 数据的加封与传递过程	6
1.6 OSI 网络模式	7
1.7 本章摘要	8
1.8 本章习题	8
2. TCP/IP 简介	10
2.1 什么是 TCP/IP	10
2.2 TCP/IP 的网络结构	10
2.2.1 网络间结构的传递过程	11
2.3 应用层	12
2.4 传输层	12
2.4.1 传输协议端口	13
2.4.2 TCP 数据格式	13
2.4.3 UDP 数据格式	15
2.5 网络层	15
2.5.1 网络地址	16
2.5.2 网络地址的分类	16
2.5.3 IP 数据格式	17
2.5.4 ICMP 通信协议	18
2.5.5 网络路径选择	20
2.6 本章摘要	22
2.7 本章习题	22
3. Socket 程序设计	23

3.1 Socket 应用程序接口	23
3.1.1 使用 Socket 应用程序接口目的	23
3.1.2 什么是一个 Socket	23
3.1.3 主从结构	24
3.1.4 Socket 的使用类型	25
3.2 WinSock 应用程序接口	28
3.2.1 WinSock 程序设计模型	28
3.2.2 CSocket 程序设计模型	28
3.2.3 CAsyncSocket 程序设计模型	30
3.3 Socket 程序设计例子	31
3.4 本章摘要	38
3.5 本章习题	38
4. 全球信息网(WWW)	39
4.1 全球信息网介绍	39
4.2 WWW 浏览器	42
4.2.1 如何由浏览器连上互联网	44
4.2.2 浏览器内部结构	46
4.3 HTML 语言	47
4.4 编写 HTML 文件——首页制作	49
4.4.1 什么是 HTML 文件	49
4.4.2 如何设计 HTML 文件	49
4.5 HomePage 例子	51
4.6 本章摘要	61
4.7 本章习题	62
5. Java 程序设计	63
5.1 Java 语言介绍	63
5.1.1 对象(Object)的概念	63
5.1.2 Java 语言是什么	65
5.2 Java 语言语法	67
5.2.1 Comments	67
5.2.2 Identifiers	67
5.2.3 Keywords	67
5.2.4 Basic Data Types	68
5.2.5 Arrays	69
5.2.6 Operators	69
5.2.7 Expressions	69
5.2.8 Control-Flow Statements	70
5.2.9 Classes	73
5.2.10 Interfaces	80

5.2.11 Native Methods	82
5.2.12 Packages	84
5.2.13 Exceptions	85
5.2.14 Application vs. Applet Program	89
5.2.15 Input/Output	91
5.2.16 Threads	100
5.2.17 Networks	105
5.3 Java 程序例子	109
5.4 本章摘要	121
5.5 本章习题	122
附录	124

1. 计算机网络简介

正式进入互联网(Internet)的缤纷世界以前,本章先对计算机网络做个简单的介绍。从计算机网络的起源谈起,循序介绍其连接方式(拓扑)、层次式结构和通信协议,并以一个简单的三层式结构和国际标准组织的 OSI(Open System Interconnection)网络模式为例,说明数据的加封、拆封和传递过程。以此概念性的介绍做为研读后续章节的理论基础。

1.1 什么是计算机网络

早先时候,一般企业都只购置一部大型计算机(mainframe)来处理所需的计算机化业务。时至今日,为了提高计算机的作业能力而将几部计算机或相关的设备以通信线路互相连接,彼此互相支持的情形,早已是司空见惯了,计算机网络(computer networks)一词也因应而生。随着各项作业需求的增加,连接的计算机跟着增加,连接的方式更为复杂,计算机和网络的关系也就更密不可分。

一个好的计算机网络必须让用户几乎察觉不到网络的存在,使用上就好像所有资料都垂手可得,至于实际的访问路径和访问方式便交由网络上的软件来负责。然而,当网络成长到一定程度以后,便会引发维护管理和运行效率上的瓶颈,为了便于维护管理,并且确保运行效率,有关计算机网络的种种课题也就越来越热门了。

1.2 为什么需要计算机网络

从早期单机计算机各司其职的运作类型,至今遍布全球信息网络的热络情形,计算机结构的演进之快是大家有目共睹的,其所带来的效益,也早已不是三言两语所能道尽的。

资源共享一直是计算机网络让大家颇为津津乐道的效益。让通信线路上的各个系统经由此一连接管道来共享彼此的软、硬件资源,进而分担彼此的工作负荷,不但可以避免重复投资的浪费,各个系统之间互相支持的措施,更可以提高整体作业的稳定度。

近来有越来越多的厂商,为了适时掌握商机,而纷纷将其行销系统搬上网络。同时,各行各业也依照其不同的需求来传递各项情报资料,举行学术性讨论、书信往来、商业行为、军事情报……等等,都可借助计算机网络的实时反应能力,而达到事半功倍的效果。电子商店、网络线上交易、电子银行等新型的往来交易,都伴随着计算机网络的日益普及而逐渐盛行。这种行动力及便利性的提高,往往是今日工商社会致胜成功的关键,也是计算机网络最可贵的一面。

1.3 计算机网络的连接方式

计算机与计算机之间的连接方式俗称为网络拓扑(Network Topology),最简单的连接方式是以点对点的方式(peer-to-peer),将两部计算机的串行或并行通信端口(serial/parallel port)对接,互传数据。然而,这种接法在某些情况下是很不经济的。因为当两部计算机相距很远的时候,便需要很长的传输线来配合,而且连接计算机增多以后,要在每两部计算机之间都架设一条传输线是相当昂贵的。所以只有将各个计算机以较有效的方式连接起来,形成一个计算机网络,才是根本解决之道。

连接的计算机越多,其连接的方式也越复杂多变。然而随着各项研究的发展和实际经验的累积,还是有了三种最典型的连接方式。分别是星形连接(Star Topology)、环形连接(Ring Topology)、总线连接(Bus Topology)。以下分三个小节加以说明。

1.3.1 星形连接方式

星形连接方式如图 1-1 所示,所有连接的周边计算机都连接到一部中控计算机,所有的数据传输路径也都是通过此一中控计算机以电路交换(circuit switching)的方式做转接,并在等候通信的周边计算机之间创建专线。任何两部周边计算机之间都不直接连接。

星形连接方式的好处是,任何一部周边计算机与中控计算机之间断线的时候,并不会影响到其余周边计算机间的通信,只要中控计算机持续的运作,则其余周边计算机之间仍可以经由中控计算机做转接,互传数据。然而过度依赖中控计算机的结果也有其缺点,一旦中控计算机出了问题时,整个网络便跟着瘫痪了。

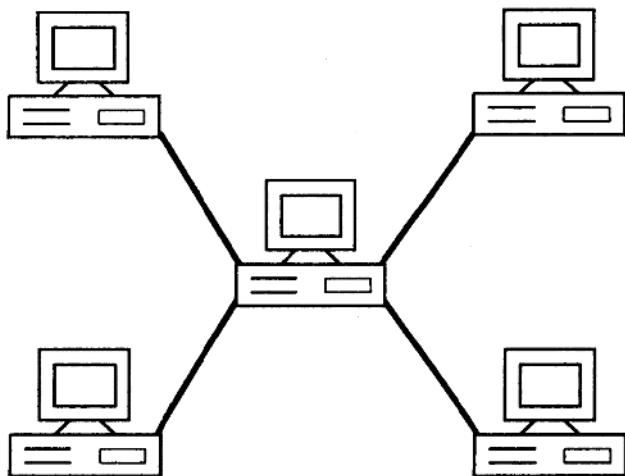


图 1-1 星形连接方式

1.3.2 环形连接方式

环形连接方式如图 1-2 所示,所有连接计算机都串接在一个环形线路上,信息的传输也是沿着此一环形线路绕经各计算机,最后又回到原发送计算机。

环上各计算机之间是以轮流的方式来传送数据,希望发送数据的计算机必须等轮到

它发送的时候,才能将数据封包(data packet)送出,因为封包的格式含有发送端和接收端的地址,所以当封包绕行网络时,环上各站计算机都可以检查其地址数据,由接收端计算机将该封包拷贝一份保留下来,而原封包则继续绕行各计算机,最后由原发送计算机吸收回去,做为回应(Reply)数据,完成该数据的通信以后,再将发送机会让给下一部计算机。

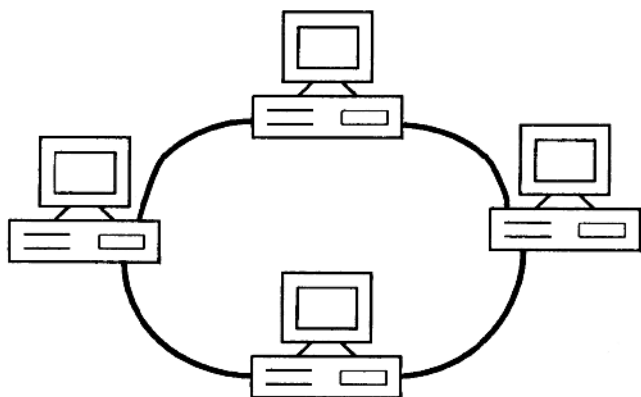


图 1-2 环形连接方式

环形连接方式必须在整个环形线路通畅的情况下才能传输数据,任何两部计算机间断线都将拖垮整个系统。此外,由于其传输数据必须绕行环上各个计算机,也比较容易遭到窃取或受到非法的监控。

1.3.3 总线连接方式

总线连接方式如图 1-3 所示,所有连接的计算机都连接挂在一条主要的传输线上,该传输线的两端则各接上一个终端设备(Terminator)。由特殊的通信协议(Protocol)来控制各站的送发机会,数据可在线上双向传输,送达目的地。但因为只有这么一条共同的传输线,所以同一时间就只能有两部计算机互传数据。

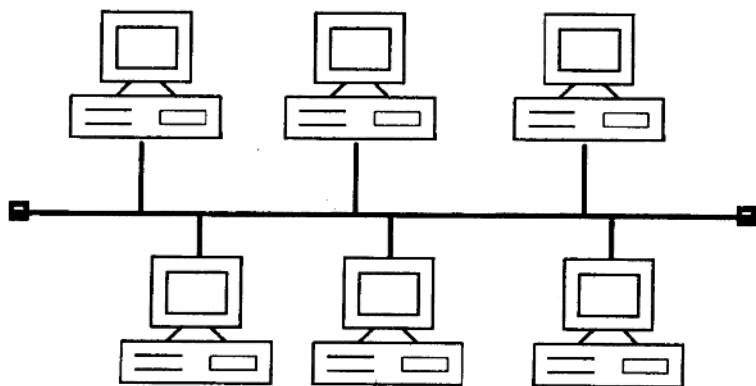


图 1-3 总线连接方式

和环形连接方式一样,其封包的格式也含有发送端和接收端地址,各站计算机也同样

会检查过往的封包。同样由接收端计算机将该封包拷贝一份保留下来,而原封包则由终端设备负责吸收,完成该数据的通信以后,再依照该通信协议,将发送机会让给其它计算机。

总线连接方式也是必须在全线通畅的情况下才能传输数据,而且同样有数据容易遭窃和受监控的缺点。

1.4 计算机网络结构

计算机网络不断发展的结果,必然是有越来越多各式各样的通信装备需要连上网络。这时候有一个潜在问题需要解决,便是各个装备之间的软、硬件必须兼容,否则便需要改善接口的设计,才能确保通信的顺畅。有鉴于此,计算机网络结构(Network Architecture)的发展便是为了提高网络上各个装备之间的兼容性,使网络适用于更多样的连接装备。

计算机网络结构的涵义包括了通信协议、信息格式,以及通信软、硬件必须遵守的各项设计规格。所以定义完善的计算机网络结构对用户和厂商都是很重要的。就用户而言,可以在实际配置网络之前做好装备的选择,在经济和技术之间取得一个平衡点,以节省不必要的开支;就厂商而言,可以根据现行网络结构来开发具兼容性的新产品,来确保行销市场。

1.4.1 计算机网络结构的设计

计算机网络结构的设计是一项繁复的工作,但是在现行的各种网络架之间,却不难发现一些共同的特性,例如:

- * 连接性

不同的软、硬件产品之间,只要采用相同的通信协议和数据格式便可以连上网络,互传数据。

- * 模块性

以功能模块的方式来发展网络设备。

- * 容易架设

为计算机网络的架设提供一个一般性的方案,允许用户以不同的装备来架设符合其实际需求的网络。

- * 容易使用

为用户提供一个容易使用的通信环境。

- * 可靠性

让网络具有适当的调试和回复的能力。

- * 容易修改

让网络结构成为一项持续性的发展工作,当新的软、硬件或相关技术发展出来时,可以配合修改。

1.4.2 层次式计算机网络结构

为了设计上的方便,通常是将计算机网络结构细分成几个不同的功能项目来考虑,再将性质相近的项目归类到同一个功能层次来设计,也就是设计成所谓的层次式结构(Layered Architecture)。而划分出来的各个功能层次必须是互相独立,使得局部层次的

修改并不会影响到相邻的层次,让整体的设计更有弹性,便于日后的各种改进措施。

1.4.3 通信协议

通信协议为一套通信双方所接受的通信规则和一般约定俗成的通信行为。在层次式的计算机网络结构下,各个层次都必须有一套规划完善的通信协议来规范所辖的通信行为,如数据表示法、数据格式、传送顺序、路径选择、错误检查、错误回复、信号规格等等,都是必须规范的事项。

通信协议可以说是计算机之间沟通的礼节,就如同人跟人间谈话的礼貌一样。两人之间的谈话,最好是使用相同的语言,否则便要有第三者帮忙翻译。计算机之间的传输也是一样,最好是使用相同的数据表示法,否则便要通过其它装置来做转换;而谈话当中如果有一方没听清楚对方的话,通常是请对方再说一次,这又好像接收端计算机检查出传输错误时会要求发送端计算机重送该组数据一样。只要试着将人和计算机的角色互换,便不难了解其在通信行为上所扮演的角色。

1.5 三层式计算机网络结构

三层式计算机网络结构是一种较简单的层次式结构,其层次是依照三个最基本的计算机网络实体设计出来的。因为计算机之间要互相传输数据之前,首先接收、传送双方的计算机都要连上网络,再经由实体网络创建连接,并在双方的计算机上运行相关的应用软件,才能开始其间的通信。三层式计算机网络结构便是依照这三个基本元件设计出来的,其结构如图 1-4 所示,自底向上的三个层次分别是网络层、传输层、和应用层。自底向上各个层次的功能,分别描述如下:

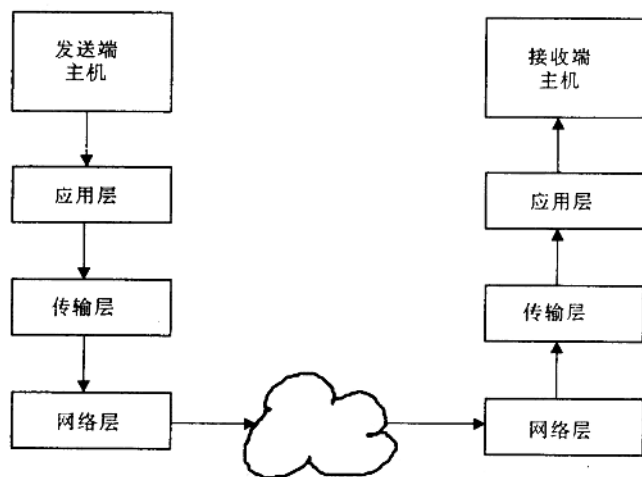


图 1-4 三层式网络结构

* 网络层

网络层负责引导数据进出网络,通常发送端计算机必须提供接收端计算机的地址,好让网络层据以引导该数据到达目的地。这一层次的软件和它所连接的网络有非常密切的

关系,不同类型的网络有不同的软件来配合运作。例如分封交换网络、电路交换网络和区域网络便各有其所属的软件来配合。

将引导数据进出网络的工作规划成一个独立的层次以后,上层的软件便可以独立设计,即使将来网络层作业方式有所更动,也不必重新设计上层的软件。

* 传输层

传输层的工作关系着整体传输的可靠度和正确性,它负责依照顺序将数据正确无误的送达目的地,其做法通常是在接收端做检查,并要求重新发送错误和脱序的数据。

* 应用层

应用层是一个范围比较广泛的层次,通常是根据不同的作业需要来发展所需的应用软件,文件传输(FTP, File Transfer Protocol)便是一个常见的例子。

1.5.1 数据的加封与传递过程

数据在三层式计算机网络结构环境下的加封与传递过程如图 1-5 所示。发送端的数据由上层往下层传递,经过各个层次的分段(Fragmentation)和加封(Encapsulation)处理,最后经由实体网络送达接收端。接收端则以相反的处理过程,将数据由下层往上传递,由各个层次解读相对层次的封包,最后重组(Reassembly)为原来发送的数据。

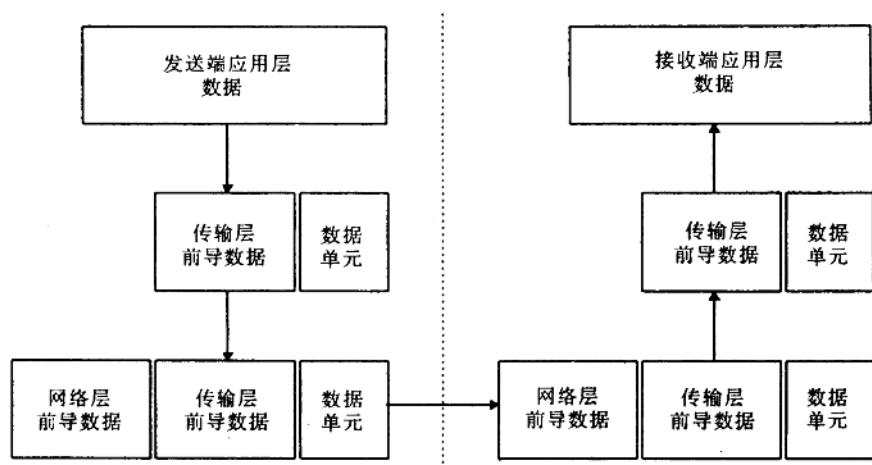


图 1-5 数据的传递过程

当发送端计算机要传递数据的时候,首先由发送端的应用程序将数据往下传给传输层处理。传输层在收到应用层的数据以后,将视数据的长度做适当的分段,以较小的数据单元(Data Unit 简称为 DU)来处理。然后依照数据顺序在各个数据单元的前面加上传输层前导数据(Transport Header),而成为传输协议数据单元(Transport Protocol Data Unit 简称为 TPDU),再往下传给网络层。

前导数据通常是一些控制信息(Control Information),借以引导传输动作。以传输层的前导数据而言,通常包括数据的顺序码(Sequence Number)和错误检测码(Error Detection Code),借以核对数据顺序并确定数据的正确性。

网络层收到传输层传来的传输协议数据单元(TPDU)以后,并不理会传输层的前导

数据,而将整个传输协议数据单元(TPDU)看成一个数据单元(DU)来处理,并在各个数据单元的前面加上网络层的前导数据(Network Header),而成为网络协议数据单元(Network Protocol Data Unit 简称为 NPDU),再经由实体网络送达接收端的网络层。

网络层的前导数据通常包括发送端和接收端的计算机地址(Source & Destination Computer Address),借以引导各个网络协议数据单元(NPDU)到达目的地。

接收端计算机的网络层收到发送端传来的网络协议数据单元(NPDU)以后,先检查网络层的前导数据,确定自己为接收端计算机的时候,便拿掉网络层的前导数据,将数据还原成传输协议数据单元(TPDU),再往上传给传输层做处理。

传输层收到网络层传来的传输协议数据单元(TPDU)以后,先依照传输层前导数据的内容检查数据传递顺序,并且在确定该数据正确无误以后,拿掉传输层的前导数据,将数据还原成数据单元(DU),再往上传给应用层做处理。

应用层收到传输层送来的数据单元(DU)以后,便将一连串的数据单元(DU)重组为原来应用层的传送数据。至此整个传递过程,算是告一段落。

1.6 OSI 网络模式

OSI 网络模式是国际标准组织(International Standard Organization,简称为 ISO)为了使各个系统之间互通,而制定的一套七层式计算机网络结构参考模式,称为开放式系统连接模式(Open System Interconnect Model,简称为 OSI 模式),后来被 CCITT 所采用,是当前用来发展各项通信标准的基础。

由于 OSI 模式设计的着眼点是在于系统与系统之间的互相连接,而不在于单一系统内部的功能,所以适用的范围特别广泛,从终端机与主机间的连接,到网络与网络间的连接,都可以应用这个模式来发展。

OSI 模式将计算机网络结构分成七个层次,自底向上分别是实体层(Physical Layer)、链接层(Link Layer)、网络层(Network Layer)、传输层(Transport Layer)、会话层(Session Layer)、表示层(Presentation Layer)、和应用层(Application Layer)。

自底向上各个层次的功能分述如下:

* 实体层

负责数据位串(Bit Stream)在传输媒体上的传送,包括实体线路的机械规格、电气信号、电路功能、和传送过程四项规则。

* 链接层

负责在通信节点(计算机主机或其他通信装备)之间传输的可靠性,包括数据块的同步、调试、和流量控制。

* 网络层

负责在收、送的两个终端节点(两端点主机)之间提供一个透明的(Transparent)传送服务,包括网络连接的创建、维护、结束,以创建路径的选择、流量控制、拥塞控制、错误控制等功能。

* 传输层

负责在收、送的两个用户之间提供一个可靠和透明的传送服务,包括流量控制、错误

检测与回复等功能。

* 会话层

负责在接收、传送的两个用户之间创建一个谈话管道,包括会话的创建、维护和结束,并规定用户的会话方式,其会话方式有单向、双向、或是双向轮换三种。

* 表示层

负责将传输数据以有意义的方式显示给用户,包括所使用的语法、格式、数据表示法和数据的压缩、还原、加密、解密等功能。

* 应用层

负责让用户进入 OSI 通信环境,并提供分布式数据处理的方法。

OSI 模式的第一~第三层(实体层、链接层、和网络层)通常称为较低层次,因为这些层次必须涉及通信节点的内部运作方式。而第四层(传输层)以上通常称为较高层次,因为这些层次只涉及接收、传送两端用户之间的通信行为。

数据在 OSI 模式下的加封与传递过程与三层次计算机结构类似。发送端的数据由上层往下层传递,经过各个层次的分段和加封处理,再由实体网络送达接收端,接收端也以相反的处理过程,将数据由下层往上传递,由各个层次解读相对层次的封包,最后重组为原来发送的数据。

1.7 本章摘要

从早期的单机作业方式到现今广泛使用信息网络的热络情形,计算机连上网络早已是形势所趋。连接的计算机越多其连接的方式也越复杂多变,三种最普遍的计算机网络连接方式是星形连接、环形连接、和总线连接。

为了设计上的方便通常是采用层次式结构来设计,先将繁复的结构划分成各个互相独立的层次,再分别设计,任何局部层次的修改并不影响相邻层次的修改。层次式计算机网络结构的数据传递过程都类似,数据由发送端计算机的最上层往下传,行经各层时都经过适当的分段,并加上前导数据成为该层的数据单元,往下层传送。到达最底层时,再经由实体线路送达目的地。各层次的通信规则都隐含在该层次的前导数据中。接收端收到数据以后,则以相反方向,由最下层往上传。行经各层时都先对该层的前导数据做检查,确定传输过程正确无误以后,再拿掉该层的前导数据,传给上层。数据单元到达最上层时经过重组便可得原来传送的数据。

1.8 本章习题

1. 请扼要的说明计算机网络是如何形成? 并且为什么需要计算机网络?
2. 试比较三种计算机网络的连接方式——星形连接方式、环形连接方式、以及总线连接方式,彼此之间的异同。
3. 请说明当前你所使用的计算机网络是什么? 又是那一种连接方式?
4. 当前常见的 Novell 区域网络(LAN, Local Area Network),服务器使用 Netware 网络操作系统,将网络上所有的用户连接在一起。请问此种网络采用那一种连接方

式,并请说明此种网络的操作方式。

5. 试说明层次式计算机网络结构的作用。

6. OSI 网络模式为何? 又其中每个层次部分所要处理的功能又是什么?

2.TCP/IP 简介

有了前一章关于计算机网络的概念以后,本章将继续介绍学习 Internet 程序设计的一项重要课题—— TCP/IP 网络通信协议。

2.1 什么是 TCP/IP

TCP/IP 是一套网络和网络之间的通信协议,正确的名称是「互联网通信协议组(Internet Protocol Suit)」。而 TCP(Transportation Control Protocol)和 IP(Internet Protocol)只是这一组通信协议中的两个通信协议而已,只因为 TCP 和 IP 是其中最具代表性的两个,所以一般都直接以 TCP/IP 来称呼这一组通信协议。

其实如何称呼不是什么大问题,重要的是不要误以为所有互联网的应用程序都使用了 TCP 和 IP,例如网络文件系统(Network File System, 简称 NFS)便是采用 UDP(User Datagram Protocol)和 IP 而没有采用 TCP。这是学习 TCP/IP 之前必须澄清的一个概念。

2.2 TCP/IP 的网络结构

TCP/IP 是以层次式网络结构所设计出来的,总共由五个层次所构成,自顶向下分别说明如下:

* 应用层

应用层是依照网络间实际操作需要而设计的通信协议,例如 FTP、E-Mail、Telnet、NFS……等,都是这一层次的通信协议。

* 传输层

传输层负责让数据在网络间稳定的传输,以程序设计的眼光来看是将数据由发送端的应用程序送到接收端的应用程序。所以,除了做好流量的控制(Flow Control)以外,还必须确保数据传递的正确性,并且重送遗失和错误的数据。

这一个层次共有 TCP(Transport Control Protocol)和 UDP(User Datagram Protocol)两个通信协议。

* 网络层

网络层负责网络与网络间数据传输路径的选择,并引导数据穿越各个中间网络到达目的地,以程序设计的眼光来看是将数据由发送端的主机送到接收端主机。这一个层次共有 ICMP、IP 和 IGMP 三个通信协议。

* 链接层

链接层为 TCP/IP 和下层各个网络之间的接口,比较常见的结构是将 TCP/IP 架在 Ethernet、Token Ring、X.25……等网络之上运作。所以,除了硬件接口以外,这一个层次还必须处理 TCP/IP 和下层网络之间不同的地址表示法。

这一个层次共有二个通信协议,ARP(Address Resolution Protocol)负责将下层网络的地址转换为 Ethernet 的地址,而 RARP(Reverse Address Resolution Protocol)则负责将下层网络的地址转换为 Internet 的地址(IP)。

* 实体层

实体层指的是实际的传输线路,例如双绞线(Twisted-Pair)、同轴电缆(Coaxial Cable)和光纤(Optical Fiber)等。

各个层次间的关系如图 2-1 所示。必须注意的是并不是所有数据的传输都经过传输控制层,有些时候数据是由应用层直接送到网络层,再经由下层的网络传出去。

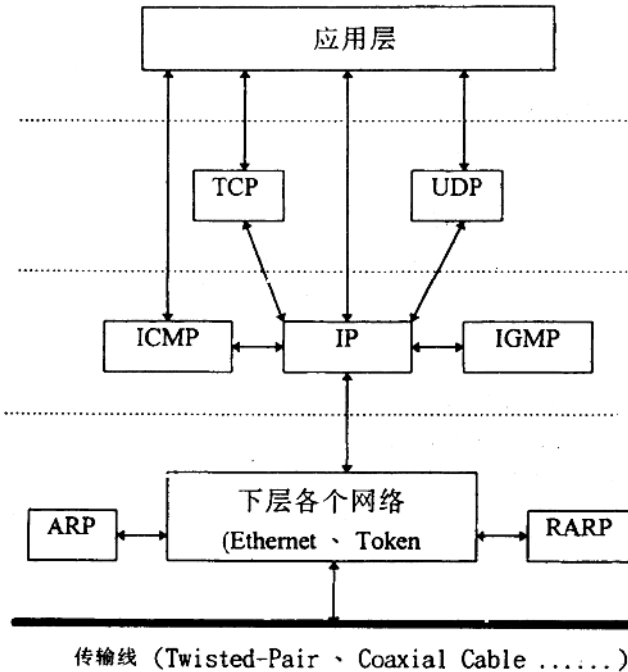


图 2-1 TCP/IP 的层次结构

2.2.1 网络/层间数据的传递过程

如前所述,TCP/IP 是属于层次式的网络结构,所以数据的传递方式也是和一般层次式网络结构的传递情形类似。通常数据都是由发送端的应用程序往下传递,依序经过各个层次的加封处理,到达下层网络,下层网络必须先看懂目的地地址,再决定这些数据该往何处送。到达目的地以后,以相反的顺序由下往上传递,经过各个层次的解封包处理,最后由接收端的应用程序接收。

接下来的几个小节将依序介绍 TCP/IP 的应用层、传输层、以及网络层等三个层次,使读者更容易了解其运作方式。

2.3 应用层

如前所述,应用层是依照网络间实际操作需要而设计的通信协议。由于早期的计算机系统都备有完整的磁盘存储系统,所以,常规的互联网应用层通信协议也都是因应这种环境而发展的。比较典型的几个应用程序如下:

* 文件传输

文件传输是指在不同的计算机系统间访问文件,包括从别的计算机系统上读取文件,或将文件写入别的计算机系统。其访问操作是由文件传输协议(File Transfer Protocol,简称 FTP)来规范。

然而,并不是所有文件都可以在计算机间传输的,基于系统安全的考虑,某些文件必须有限措施。所以,针对外来的用户,通常系统会先核对该用户的姓名和口令,并限定其访问范围,以确保数据的安全性。

* 远端登录

远端登录(Remote Login)指的是,用户可以登录到网络上的其它计算机系统来操作,其操作方式是由网络终端机协议(Network Terminal Protocol,简称 Telnet)来规范。登录到其它系统以后,所有键盘的输入都会送到该系统做处理,所有操作也都在远端主机上运行,直到结束登录(logout)以后,才会再回到原来的系统上操作。

* 电子邮件

电子邮件(Electronic Mail,简称 E-Mail),可以让用户在不同的计算机系统间接收、发送信件,几乎是当前最流行的互联网应用程序。

以程序设计的眼光来看,寄信指的是把自己的信件放入别人的信件文件内,而收信则是把别人的信件放入自己的信件文件中。由于,送信之前必须先接收、发送计算机间创建一条连接,所以,信件的往来通常是由大计算机来处理,以免因为经常关机而导致连接无法创建,而延误信件的传递。

2.4 传输层

如前所述的数据传递方式,应用层将数据送给传输层,传输层则将该数据分割成较容易处理的数据段,加上前导数据处理以后形成各个信息单元(Message Unit),送达接收端以后再依照顺序重组回原数据,并要求重送遗失和错误的信息单元。

这个层次共有 TCP 和 UDP 两个通信协议。其主要的差异是 TCP 为可靠的通信协议(Reliable Protocol),而 UDP 为不可靠的通信协议(Unreliable Protocol)。

顾名思义,可靠的通信协议指的是一个能确保数据正确无误的送达目的地的通信协议。其做法比较复杂,通常发送端每送出一组数据以后,都会等待接收端回应一个认可(Acknowledgment)信息,确定该数据正确无误的到达目的地以后,再继续发送下一组数据。所以接收端每收到一组数据以后都必须检查该数据的顺序码(Sequence Order)和总和检查码(Checksum),如果一切正确无误,则回应一个认可的信息给发送端,若发现任何传输错误或数据遗失,则要求重送该组数据。相反的,不可靠的通信协议只负责将数据送