

摘要

网络控制系统是指控制回路通过实时网络进行闭合的反馈控制系统。控制系统的网络化程度是工业自动化的关键，是衡量一个国家工业水平的重要指标。网络控制是控制科学与飞速发展的计算机网络及通讯技术相结合的产物，信息的网络化和智能化使网络控制系统中存在一些亟待解决的问题，如网络传输延迟、数据包丢失、数据包在传输过程中发生排列顺序混乱等。控制网络中存在的上述问题将导致控制信号或反馈信号的时延，该时延将降低控制系统的性能，甚至使控制系统不稳定。因此，控制系统的网络化是对已有控制理论和控制模式的巨大挑战，又极可能衍生出新的控制模式和控制方法。

本文提出了一种解决基于 Internet 的网络控制系统相关问题的创新控制模式，以多中心化的网络节点架构为背景，以系统集成思想为依据来解决上述控制网络存在的问题，将平面网络提升到层次网络的概念。本文提出的多路径数据传输模式为具有功能自治特性的对等网络节点设置了“路由管理”和“数据传输”两类功能，由网络中的路由管理器节点为传输数据的节点提供明确的路径指示，实现快速选择数据传输路径和以多条指定路径同时向目标节点并发数据的功能，同时建立存放网络链路属性的路径信息数据库，从而形成一个新式的控制网络数据传输模式。

本文对相关的控制科学与网络通信等多项技术进行了认真的研究，并用 JAVA 语言编写了一个描述多路径数据传输模式的软件模拟原型以实现其工作过程，为在实际网络中推行该模式提供了依据。

关键词：网络控制系统，控制模式，传输延迟，多路径

Abstract

Networked Control System (NCS) connects the controllers, the sensors and the actuators through one shared networking to form feedback control loops. The development level of NCS plays a decisive part in industrial automatic process of a country. NCS is a product with combination of automation, computer network communication technology and innovation of information, but the communication network inserted in NCS has brought some issues such as data transmission delay, lost of packets, packets in disorder and etc. These problems make the NCS's control quality worse, and sometime make NCS become instability. The major challenges can lead to a progressive innovation on methodologies and control modes.

In this thesis, one innovative mode called multi-route data transmission control mode is proposed to solve the related issues of NCS based on Internet. The mode has integrated the peer-to-peer network structure and system integration theory. The mode set the net peers to play the roles of router manager or data transform protocol agent. The route database is also set to record the proper routes while sending data packets. A progressive is presented while the net is divided to layers.

The important related control science and network technologies are detailedly studied. There is also a software simulation system based on JAVA to demonstrate how the mode actually works. This system provides a good basic for running the mode in a true networked control system environment.

Key Words: Networked Control System, Control Mode, Data Transmission Delay, Multi-Route

学位论文版权使用授权书

本人完全了解同济大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版；学校有权保留学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版；在不以赢利为目的的前提下，学校可以适当复制论文的部分或全部内容用于学术活动。

学位论文作者签名：仇鹤凤

2007年3月16日

经指导教师同意，本学位论文属于保密，在 年解密后适用本授权书。

指导教师签名：		学位论文作者签名：	
	年 月 日		年 月 日

同济大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律 responsibility 由本人承担。

签名：仇鹤凤

2007年3月16日

第1章 绪论

1.1 网络控制系统概述

随着计算机被用于控制生产过程，取代了传统的电动、气动式模拟控制装置。计算机控制系统经历了从直接数字控制（DDC）到集散控制系统（DCS）到网络控制系统（Networked Control Systems, NCS）的不断发展。

网络控制系统(NCS)是通过网络形成闭环控制回路的控制系统，最早于1998年出现在马里兰大学G.C.Walsh的论著中^[1]，也被称为“通讯和控制综合系统”(Integrated Communication and Control Systems, ICCS)^[2]、“基于网络的控制系统”(Network-based Control Systems)等^[3]。这种分布式控制系统是计算机技术、通信技术与控制技术发展和融合的产物，是计算机控制系统更高级的发展，以实现生产过程控制的彻底分散化、网络化，使系统配置达到最优，实现完整的控制、管理网络架构。

网络控制系统强调了分布式和控制系统的概念，是将控制系统的传感器、执行器和控制器等单元通过网络连接起来的分布式闭环控制系统^[4]。其定义主要突出了以下两个方面：系统节点的分布化和控制回路的网络化，典型的网络控制系统结构如图1.1所示：

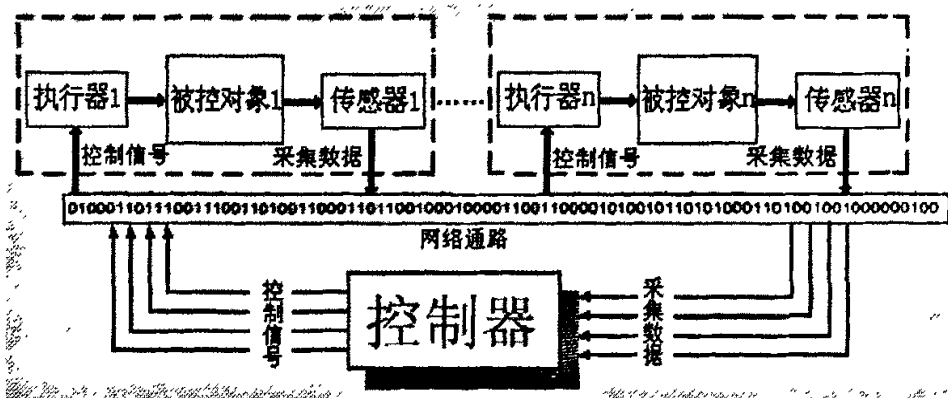


图 1.1 典型的网络控制系统结构图

目前，网络控制系统已经成为国际控制领域的一个研究热点，自动控制杂

志(Automatica)、美国控制会议(American Control Conference, ACC)、国际自动控制联合会的世界大会(IFAC World Congress)及 IEEE 的控制与决策会议(Conference on Control and Decision, CDC)等各种国际控制界最重要的学术期刊和会议上都有有关 NCS 的专题及各种研讨会。在控制系统领域相当具有权威性的 IEEE 控制系统分会(Control Systems Society, CSS)的自动控制杂志(IEEE Transactions on Automatic Control)也设立了 NCS 专刊。

由于控制系统的控制回路建立在网络上,极大地提高了工业生产的信息集成度,同时也使得整个控制系统的稳定性以及性能除了依赖控制算法外,还与网络的传输能力密不可分。NCS 中的网络系统与控制系统相互依赖、相互渗透,增加了整个网络控制系统的复杂度。目前网络控制系统的实际应用系统多数采用能够严格控制信息传输延时的现场总线网络,但是随着计算机网络的发展和控制系统规模的日益扩大,从企业发展和控制系统的应用前景来看,需要将现场控制与上层管理相联系,将孤立的自动化孤岛连接起来形成统一的企业信息化系统。同时,由于网络技术特别是互联网技术的不断发展,以公共数据网络(例如:Internet)为网络通路的 NCS 被很多学者公认为未来控制技术的发展趋势。

1.2 网络控制系统的特点和研究分析

网络控制系统是控制科学与飞速发展的计算机网络及通讯技术相结合的产物与常规的集中型计算机控制系统相比具有十分显著的特点:

- 结构网络化:支持如总线型、星型、树型等拓扑结构,比分层控制系统的递阶结构具有更好的平稳性;
- 节点智能化:NCS 的每个节点具有各自相对独立的功能,带有 CPU 的智能化节点之间可通过网络实现信息传输和功能协调;
- 控制现场化:网络化结构可以将任务下放到智能化现场设备上执行,使控制的风险因素得到分散,从而提高了系统的可靠性和安全性;
- 系统开放化:由于系统模块之间通过网络来连接,所以系统的开放程度高,可以灵活分解系统模块,便于实现先进的控制技术,通用性强。而且系统的构成也更加灵活,便于扩展。

网络控制系统的上述特点决定了其安装维护简单,网络管理灵活,故障诊

断容易等优势。因此近几年来以现场总线为代表的网络控制系统得到了前所未有的快速发展，在遥控操作、远程医疗、远程教学实验、局域控制（如网络家电、蓝牙技术、办公室自动化系统和计算机集成制造系统）等场合得到了广泛的应用。

由于网络控制系统是通过一系列的通信信道构成一个或多个控制闭环。因此研究网络控制系统需要综合考虑网络与控制两大问题。除了物理设备即被控对象的性能和稳定性，网络通信系统的性能和稳定性也是网络控制系统课题的研究重点。

因此整个网络控制系统中控制部分的研究重点集中在设计更加优化的反馈控制算法来消除传输延迟对控制性能的影响^[6]；网络通信部分的研究则主要关注如何减少网络延迟、抖动等问题，采用的方法也很多，如增加网络带宽^[8]、优化网络拓扑结构^[7]、调整信息调度策略、选用特殊传输协议^[4]等。因此可以归纳出网络控制系统的研究分析关联图，如图 1.2 所示。

根据定义将网络控制系统划分成传输网络、网络化设备和控制系统三个主要部分。传输网络的特性（拓扑、协议、调度算法以及数据包长度等）、网络化设备的属性（产生数据的频率、单位时间里产生的数据量的大小等）以及网络上传输的数据流特性决定了数据信号在网络通路中的延时。研究网络控制系统的相关问题时，可以根据网络传输能力和网络化设备属性假定一个网络延时模型，然后将其作用于控制系统部分，并结合控制系统的可控性、可观性、控制算法的收敛性等研究内容，进行综合分析、设计建模，最终实现整个网络控制系统的实时要求和稳定要求。

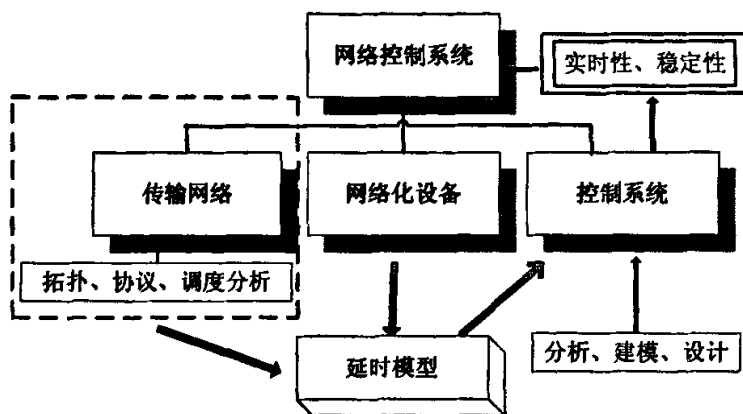


图 1.2 网络控制系统研究分析结构图

1.3 网络控制系统的传输网络问题

网络控制系统中的传输网络是一个很大的范畴,包括了现场总线网络,但又不局限于现场总线网络,更可以推广到以太网、无线网络、基于 IP 的 Ethernet 或 Internet 等多种形式的传输网络。

控制网络的类型和协议有很多,常见的是 8 种 IEC61158 的标准现场总线: FF, ControlNet, Profibus, P-NET, FF(HSE), SwiftNet, WorldFIP 和 Interbus; 没有被 IEC61158 接纳但是具有相当影响力的现场总线还有 CAN, Lon, HART 等。

各种网络协议的结构模型、技术细节都有所不同。但各种网络都兼顾了国际标准化组织 (ISO) 的开放系统互联 (OSI, Open System Interconnection) 标准,又充分考虑自己的技术特点和应用需求,实现了具有自己特点的控制系统专用网络。

按照控制网络的实时特性对其进行分类,常见的控制网络分类方法有两种,一是按照实时性和传输延时的确定性分类;二是按照 MAC 层协议分类。

按照对实时性和传输延时确定性能力的支持,现场总线网络协议主要分为两大类:一类是不直接支持实时性的协议,采用事件触发的方式;另一类则是直接支持实时性的协议,采用时间触发方式。事件触发方式的网络协议多数采用随机载波监听 (CSMA, Carrier Sense Multiple Access) 协议,代表性的控制网络有 CAN (CSMA/AMP)、LonWorks、Ethernet (CSMA/CD) 等;时间触发方式的网络协议通常采用令牌 (Token) 协议,这种协议又进一步分为: 1)、集中式令牌,具有代表性的协议有 WorldFIP 和 FF 等; 2)、分布式令牌,具有代表性的协议有 ProfiBus 等; 3)、虚拟令牌,具有代表性的协议有 P-NET 等;按照控制网络 MAC 层协议来看主要分成三种类型,即 CSMA 方式、Token Bus 方式和主从 Polling 方式。每种类型都产生了许多具有代表性的控制网络协议:以太网、CAN 和 Lon 等都是基于 CSMA 方式; Profibus, P-NET 及 Control Net 等则是基于 Token Bus 方式; FIP 以及某些专用主从式 RS-422/485 网络可以归为主从 Polling 方式控制网络协议。

然而随着计算机网络的不断发展和控制系统规模的日益扩大,网络控制系统的应用越来越广泛。NCS 的传输网络已不仅仅是能够严格控制信息传输延时的专用网络 (Control Networks),可以扩展到提供“尽力而为” (Best Effort) 效益的公共数据网络,以公共数据网络为网络通路的 NCS 被很多学者公认为未来

控制技术的发展趋势，具有广阔的应用前景。随着网络技术与控制系统的结合不断深入，工业以太网(Ethernet)，无线网络甚至互联网 Internet 等公共数据网络也可以用来传输控制信号和决策信号，促使新的网络控制协议、算法等成为研究热点。

由于网络控制系统通过网络进行信息传输，因此与传统的控制系统相比，具有以下优点：系统搜索覆盖面广、扩展灵活、故障诊断容易、安装维护简单等。随着互联网技术的不断发展，基于 Internet 的网络控制技术已成为控制领域的一项研究热点。Internet 可以将以往相互独立的，散落在各个地方的单个计算机或者相对独立的计算机局域网，借助已经发展得有相当规模的电信网络，通过一定的通讯协议实现更高层次的互联。同时互联网本身通信模式也在不断地进步与创新，这又增加了基于 Internet 的网络控制系统的发展空间，并且极有可能推动衍生出控制理论的新范例和新模式。目前，国内外学者针对基于 Internet 的网络控制系统展开的研究包括各种新的网络控制理论与方法、基于新一代互联网模式的开放式控制系统、拥塞控制与调度、网络控制系统的分析与建模等。此类研究的根本方向即在于扩展控制理论的新范例及模式。

从本质上来说，目前应用较多的现场总线网络控制系统(Fieldbus Control System, FCS)侧重的是节点之间实时或非实时数据的传输和共享，而以公共数据网络为网络通路的 NCS 强调的是将控制回路建立在通信网络上。因此从系统结构的角度来看可以认为 FCS 是网络控制系统的初级阶段。

当传感器以异步方式将信息包传输到通信网络上，控制器根据信息包进行数据处理，然后将控制信息发送到执行器。在这一过程中，数据包的传输可能有时变延迟，甚至丢失(无限长延时)，通讯连接可能发生断路或被阻塞，传感器和执行器可能被停止使用，新的传感器、执行器和处理器可能被加入到该系统，需要自动地进行重新配置以使用新的资源。如果 NCS 的传输网络并不能保证各节点之间的信号以一个已知的固定延迟或有最差上界的延迟来传输，就会影响整个系统的性能。

如同现实世界的交通情况一样，一辆汽车从目的地出发，在路上行驶经过多个站点，最后到达目的地所需要的时间，与道路的宽度、是否平坦等性能有关，也与此时段道路上的汽车数目及其分布地点密切相关。网络的传输能力和在网络中传输的数据流的特性决定了信号的传输延时及其模型。网络在控制系统中是信息传输的通道，网络特性将影响控制信息的传输，继而影响整个系统

的工作特性。因此，研究网络控制系统的传输网络问题，就是集中研究如何改善网络的数据传输能力，如何更加合理地设置在网络上传输的数据流的特性。

1.4 主要工作和论文结构

在互联网通信模式不断发展的基础上，本课题着重研究基于 Internet 的网络控制系统中信息传输问题的解决方案，提出了能够快速选择信息传输路径，并且能够沿多条路径并行发送数据的一种新型数据传输模式。在本文最后还给出了控制网络多路径传输模式的软件模拟原型。

本文主要的创新点在于：

创新性地提出了一种基于 Internet 的网络控制系统数据传输模式，以多中心化的网络节点架构为背景，以系统集成思想为依据，将硬件设备、网络基础设施、网络基础服务系统、应用软件等组织成为一体，用以解决控制网络存在的问题。将平面网络提升到层次网络的概念。具体描述为：

- 由网络中的路由管理器节点为传输数据的节点提供明确的路径指示，实现快速选择数据传输路径的效应；
- 不同于目前网络中实行的单一路径路由策略，本设计以指定的多条路径同时向目标节点传输数据，确保信息传输的可靠性；
- 在网络的应用层设置面向用户的路由管理进程，不使用 IP 地址，而是用实名来代表网络节点；

本章为论文的绪论部分。介绍了网络控制系统的概念，网络控制系统与传统的集中型控制系统相比的所具有的特点，着重强调了网络控制系统中传输网络部分的研究意义和概况，以及论文的主要内容和安排。

第二章系统性地介绍了网络控制系统中传输网络部分的相关问题，重点分析了以 Internet 为传输网络的控制系统结构及其存在的问题，并针对其中的网络延迟问题进行论述。

第三章阐述了与本设计相关的知识内容，重点介绍了新一代多中心化的对等网络通信模式，基于 TCP/IP 的互联网数据传输技术，网络通信中的 Socket 套接字工作原理、数据传输的路由技术，以及网络管理与系统集成的原理。

第四章首先对控制网络多路径传输模式的设计思想进行了具体描述，然后详细地解析了该模式下路由管理器节点、数据传输节点和路径信息库的功能原

理，最后用软件模拟原型来具体说明了多路径传输模式的工作流程。

第五章对全文作了总结，对进一步的研究作了展望，希望本设计能推动网络控制系统的研究领域衍生出控制理论的新范例和新模式。

第2章 网络控制系统的传输网络

2.1 控制网络的结构

根据不同的应用环境，控制网络的结构可以分为两种类型，直接控制结构(Direct structure)和分层控制结构(Hierarchical structure)^[9]。

在直接控制结构中，由执行元件、被控对象和传感器组成的远程系统(Remote System)与控制器分别位于网络的不同位置，两者在物理上是异地的，如图 2.1 所示。这两部分通过网络进行数据交换，控制器发出的控制信号以包或帧的形式通过网络传送给远程系统，而远程系统中则将传感器的输出数据也以包或帧的形式通过网络反馈传送给控制器。这样，控制器与远程系统通过网络形成了闭环控制系统。在实际应用中，一个控制器可以通过网络形成多个环路，同时控制多个远程对象。直接控制结构适合要求具有快速响应的简单对象的控制，目前出现的采用这种控制方式的应用实例有远程实验室，远程直流电机速度控制系统等。

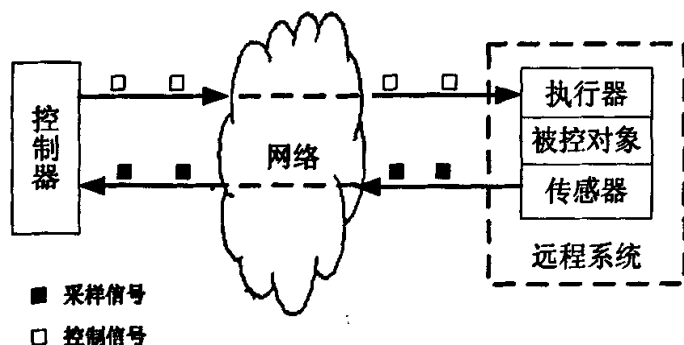


图 2.1 直接控制结构

除了直接控制结构以外，在网络控制系统的传输网络中，分层控制也是比较常见的结构，如图 2.2 所示，与直接控制不同的是，分层控制结构中的远程系统模块本身也是一个闭环反馈控制系统，包含一个远程控制器，而与直接控制结构中相对应的控制器被称为主控制器，主控制器通过网络向远程系统以包或帧的形式发送参考信号，远程系统通过自身的闭环完成对被控对象的控制，

传感器信息除传送给本地的远程控制器外，还通过网络以包或帧的形式传送到主控制器中，即分层控制结构由两个控制环路构成。同样，分层结构下的主控制器也可以同时控制多个远程闭环子系统，外环的网络控制环路同本地的控制环路相比，一般具有更长的采样周期。如果远程系统包含复杂的被控对象，则适合采用层次控制结构。采用这种控制方式的例子主要有机器人和遥操作等。

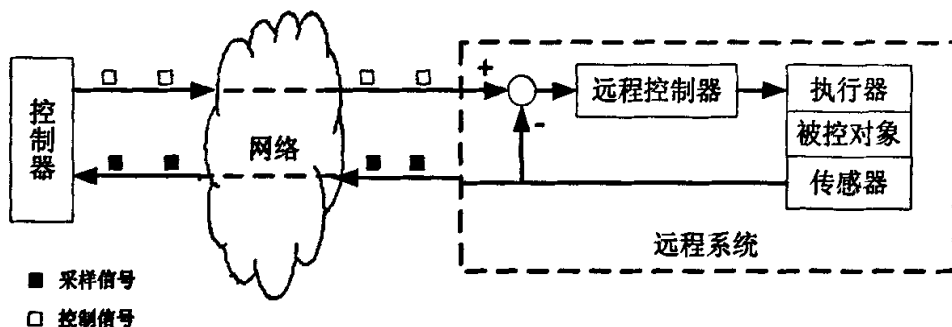


图 2.2 分层间接控制结构

表面上，两种控制网络的结构存在一定差别，其实本质上两者是一致的，在分层控制结构中只要将远程闭环系统看作整体，接收主控制信号回送采样信号。形式上的不同之处在于系统对采样周期和控制周期的要求上，直接结构的控制网络对实时性的要求比较高，采样周期相对较小。而在系统设计过程中究竟采用哪一种控制网络结构，取决于被控对象的要求和特性。例如，被控对象是一个远程控制的机械手，需要控制几个马达同步且平滑地工作，此时采用含有远程控制器的层次结构更加合适；如果被控对象是远程实验室，则采用直接控制结构更为合理。

2.2 基于 Internet 的网络控制系统存在的问题

在基于 Internet 的网络控制系统中，由于信息通过多节点、多用户共享传输介质通道的互联网传输，且网络流量变化不规则。所以在网络控制系统的传感器、控制器和执行器通过网络交换数据的过程中必然会产生如：丢包、乱序、延时等各种亟待解决的问题。这些问题的存在将降低控制系统的性能，甚至使

控制系统不稳定。

2.2.1 网络延迟问题

对用户来说,网络延迟是指从用户发出请求到远端系统对该请求做出响应传回给用户的这一段时间。对基于 TCP/IP 协议通信的 Internet 来说,对用户发出的每一个请求都要做出如下处理:路由处理、ADU(用户数据单元)在网络上传输以及服务器对用户请求的处理,而这些处理过程都会引起延迟^[10]。

➤ 路由延迟^[11]

路由延迟包括域名请求延迟、TCP 连接建立延迟、TCP 连接释放延迟和 IP 在各个网关的寻径延迟。

若用户在网络应用中使用的是对方主机域名而不是 IP 地址,在应用程序通信之前应解析出对方的 IP 地址,域名解析过程引起的延迟就称作域名请求延迟。应用程序将域名交给本地解析器软件,该软件首先在本机缓存区中查找相应域名,称之为地址联编;如果找不到相应域名,本地解析器构造一个询问报文,发往初始域名服务器(本地服务器),域名服务器根据解析情况回答一个响应报文。域名服务器解析采用两步法:当初始服务器找不到该域名时,则将询问报文发往根服务器,进行自顶向下的搜索(域名服务器组织成树形层次结构)。本地缓存区对域名请求的响应延迟是由 CPU、内存、外存速度决定的,该延迟相对来说很小。域名服务器的响应与网络负载、服务器的速度和负载有关。当需要从根服务器开始查找域名时,传输延迟则成为局域网(LAN)上的延迟,此延迟相对较大。

TCP 连接建立延迟是指从传输服务用户要求建立连接到收到连接证实之间的时间,它包括远端传输层的处理工作时延。

TCP 连接释放延迟是指从一端的传输层用户发起释放连接请求到另一端的释放实际发生为止之间的时延。连接建立和释放延迟与网络负载及服务器负载有关。

TCP/IP 协议中,每个 IP 数据包是独立寻径的。IP 寻径延迟是指 IP 数据包从源端到目的端的各个网关上的寻径延迟,包括网关寻径表处理延迟和地址解析延迟。网关地址解析由 TCP/IP 提供的 ARP(Address Resolution Protocol,地址解析协议)来完成。由于网关寻径在本机内完成,而地址解析也在本网内进

行, 所以 IP 寻径延迟也相对较小。

► ADU 在网络上的传输延迟^[11]

位于应用层的用户数据单元(ADU)在应用层协议调用下层协议提供的服务之后, 形成协议数据单元(PDU), PDU 作为独立的数据单元在网络上传输。PDU 在网络上的传输延迟由以下几部分组成: 打包延迟、传送延迟、传播延迟、排队延迟及处理延迟。

打包延迟是实时数据流应用独有的延迟。实时数据流应用是指对基于时间的信息(比如视频、音频和动画等)进行实时传送的应用。实时信息流是有时间性的, 按时间顺序有规律地产生, 要等待信息流达到一定的数量以满足协议数据单元有效负载才打包成 PDU, 这段等待时间称作打包延迟。打包延迟为 S_p/K , 其中 S_p 为 PDU 有效负载长度, K 为应用的编码速度。因此采用低速编码器的通信过程, 打包延迟会比较大。

传送延迟是指 PDU 数据全部传送到线路上的这段时延, 或者说是从 PDU 的第一个比特沿端点传送到线路上到最后一个比特离开端点所需的时间间隔。传送延迟与 PDU 大小以及局部线路上的传送速率有关, 对于低速链接, 传送延迟将会比较大。

传播延迟是指 PDU 在介质中传播所用的时间, 取决于信号穿过介质的速度。也就是说, 这个延迟由链路的物理特性决定, 与链路的业务量无关。传播延迟是随着通信距离而线性增加的。通常局域网 LAN 和 LAN 中的传播延迟只有 50 至 200 μ s, 则传播延迟并不算延迟的重要组成部分。而广域网 WAN 中的传播延迟就变得比较重要了, 因为远距离传送使得传播延迟可以达到几十毫秒。

排队延迟是分组交换网中的主要延迟, 它指的是 PDU 在传输路径上每交换一次所引起的缓冲延迟的集合。若分组交换临时过载, 每一个 PDU 的目的输出端口上可能有许多分组排队。队列中位于 PDU 前的每一个分组都会产生一个等于传送延迟的附加延迟。在“先入先出”(FIFO)队列机制的交换中, 新到达的 PDU 的排队延迟等于己在该输出端口上排队的所有 PDU 的传送延迟总和。所以排队延迟不仅与队列前面的 PDU 数量有关, 也与输出端口的传送速度有关。作为主要延迟的排队延迟受当前网络负载影响, 它也是分组交换网中延迟变化的主要因素。由于 Internet 主干网上每个路由器都有大量的数据包排队, 排队延迟也成为 Internet 上的主要延迟。若假设一项数据传送时经过 10 个路由器, 每个路由器平均有 10 个 IP 数据包排队, 则这条路径的排队延迟能够达到上百毫

秒。

► 服务器对用户请求的处理延迟

在现行的 Internet 通信模式下，信息可以在客户与服务器之间沿任意方向相互传递。在某些情况下，客户向服务器发送一系列请求，服务器返回一系列应答，比如，一个数据库客户程序可能允许用户同时查询多条记录。服务器既能接收信息，又能发送信息。大多数文件服务器都被设置成向客户发送一组文件。就是说客户发出一个包含文件名的请求，而服务器通过发送这个文件来应答。然而，文件服务器也可被设置成输入文件，它允许客户发送一个文件，服务器接收并储存于磁盘。

虽然一台计算机上可以运行多种服务，但它与互联网只有一个物理连接。同时提供多种服务如 FTP 服务、WWW 服务的计算机系统，虽然允许动态创建多个应用程序并发执行，但这些需要高级的操作系统和强大的硬件基础。所以服务器处理用户请求的延迟在随着互联网规模的不断扩大。

上述网络传输延时的存在大大增加了网络控制系统研究的复杂度，延时的大小和确定性程度受到多方面因素的影响。下面再通过分析典型的网络控制系统中信息的流程来获得整个 NCS 延时的基本组成：

网络控制系统中，一个完整的控制周期包括数据采集、传输、计算和决策反馈四个过程。信息经过每一个处理过程都会产生不同的延时，宏观上，延时随着信息的流动会逐渐增大，这其中包括处理数据的延时（如控制器的计算延时）和传输延时（主要是数据经过网络需要的时间）^[17]，如图 2.3 所示。

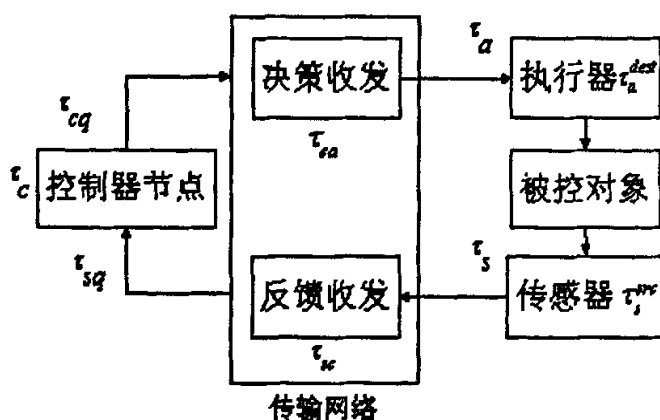


图 2.3 NCS 数据流程模型

其中, τ_s^{src} 为传感器开始采集信号直至信息被放入就绪队列, 等待网络传输的处理延时, τ_s 为传感器信号在队列中等待网络传输的延时; τ_{sc} 为反馈信息在网络上传输所产生的延时; τ_{sq} 为反馈信息等待被控制器使用的延时; τ_c 为控制器节点执行控制算法的延时; τ_{cq} 为决策信息等待网络媒体空闲的延时; τ_{cs} 为决策信息在网络上传输所产生的延时; τ_a 为决策信息在等待被执行器采用的延时; τ_a^{dest} 为执行器处理决策信息的延时。

由此可以得出从反馈信号的产生, 到相应的决策命令到达被控对象节点, 如果中间没有数据丢失的话, 整个过程的延时 τ 的组成如式 2.1 所示

$$\tau = \tau_s + \tau_{sc} + \tau_{sq} + \tau_c + \tau_{cq} + \tau_{cs} + \tau_a \quad (2.1)$$

由于 τ_{cq} 和 τ_s 都是等待网络传输而引起的延迟, 因此可以分别将其归入 τ_k^{ca} 和 τ_k^{cc} 。当执行器和控制器采用事件驱动时, 有 $\tau_a = \tau_{sq} = 0$ 。一般情况下, τ_c 可以忽略不计, 但在某些情况下, τ_c 也会对控制系统的品质产生不可忽略的影响^[14]。

一般而言, 式(2.1)只是一个平均意义下的延时度量, 因为一个远程决策一般都是根据多个反馈推出, 不是由单独的反馈而定。通常情况下, 网络平均延时的 2 倍就是 τ 的一个很好且容易获得的近似值^[15]。

式(2.1)只是从控制系统的角度, 给出了各种延时的产生原因, 但是如果从数据使用的角度分析延时的组成, 就需要从网络传输的角度进一步分析上述每一个延时环节。

信源(Source)产生的延时: 信源节点从开始产生数据到信息完全被送出的延时 T_{pro}^{src} 包括了数据在源端的处理时间 T_{pro}^{src} 和等待网络空闲的时间 T_{wait}^{src} , 即:

$$T_{pro}^{src} = T_{pro}^{src} + T_{wait}^{src} \quad (2.2)$$

例如, 当信源为传感器时, 结合图 2.3 有 $T_{pro}^{src} = \tau_s^{src}$, $T_{wait}^{src} = \tau_s$; 当信源为控制器时有 $T_{pro}^{src} = \tau_c$, $T_{wait}^{src} = \tau_{cq}$ 。

而网络传输延时, 即数据在网络中的传输延时 T_{tr} 是本文的主要关注对象。对延时要求较高的网络控制系统来说, T_{tr} 指的是数据在网络中的传输时间; 对于远程实验室和遥操作等应用而言, T_{tr} 指的是数据在广域网络中的传输延时, 此时的延时就可以通过网络服务质量调节和控制。与图 2.3 对应, $T_{tr} = \tau_{sc}$ 或 $T_{tr} = \tau_{cs}$ 。网络传输延时与信源、网络特性等密切相关。

除了上述两个主要延时 (T_{pro}^{src} 和 T_{tr}) 外, 信息到达目的节点后需要等待节点使用该信息, 另外还有处理该信息的时间, 这里记作 T_{post} ; 包括在目的节点等待

被处理的时间 T_{wait}^{dest} 和处理时间 T_{pro}^{dest} 。当目的节点为事件驱动时 T_{wait}^{dest} 等于零。

$$T_{post} = T_{wait}^{dest} + T_{pro}^{dest} \quad (2.3)$$

当节点为执行节点时, 结合图 2.3 有 $T_{wait}^{dest} = \tau_a$, $T_{pro}^{dest} = \tau_a^{dest}$ 。

T_{pro} 和 T_{post} 通常都是时变的, 并且与 T_{α} 相比, 它们的数值和变化都非常微小, 通过对网络控制系统硬件和驱动方式的调整可以使二者延时产生的影响减少到可以忽略的地步。因而在分析设计 NCS 时, 可以把 T_{pro} 和 T_{post} 包括在 T_{α} 内一起考虑。^[16]

从上述分析可以看出, 对于控制系统的传输网络来说, 最主要的描述参数有带宽、差错率、延迟等。而从控制系统的角度来看, 由于要传输的控制信息数据量相对比较小, 对带宽没有特别的要求; 信元差错率能最终反映在延迟的特性上 (如引起数据包超时到达甚至丢失), 所以不用单独考虑; 而传输过程中的时变延迟会大大降低控制系统的性能, 甚至引起系统不稳定, 因此时变延迟对于控制系统是一个巨大的挑战。

2.2.2 数据包丢失和乱序问题

由于网络中信息以数据包的形式传输, 当 NCS 中采样周期大小确定的不恰当, 或者网络的传输能力有限, 不可避免会产生存在网络阻塞或连接中断的现象, 按照某些网络协议, 将会丢弃网络中的数据包, 这会直接导致数据包的丢失。虽然 Internet 可以设有重传机制, 但重传的数据延时相对比较大, 属于过时的系统状态信息, 对系统控制的影响往往是负面的。一般而言, 反馈控制的被控对象能够忍受一定比例的数据丢失。特别的, 对于一个原本没有发生数据包丢失的稳定系统, 当数据丢失率达到某一定值时, 系统将变得不再稳定^[12]。因此, 数据包的丢失将会影响整个控制系统的实时性, 甚至影响系统的稳定性。

网络中的数据传输可以分为单包传输和多包传输^[13]。其中单包传输是指 NCS 中的传感器、控制器的一个待发数据被捆绑在一个数据包中一起发送。而多包传输是指 NCS 中的传感器、控制器的一个待发的数据被分成多个数据包进行传输。在包交换网络中要求进行多包传输, 一方面是因为单包传输字节大小的限制, 一个数据包只能装载有限的信息; 另一方面是因为 NCS 的传感器和控制器常常分布在一个较大的物理空间, 要将这些数据放在一个数据包中传输往往是不现实的。

无论采用单包传输还是多包传输，在网络传输过程中由于不可避免的存在网络阻塞和连接中断，必然会导致数据包的丢失。虽然一些网络具有重新传输数据的机制，但是只能在一个有限的时间内重新传输，当超过这个时限后，就很难再利用此数据包。因而可以将这种情况也可以归结为 NCS 中的数据丢包的一种特殊情况。传统的点对点结构的控制系统基本上都是同步和定时的系统，未建立动态模型，可以使得系统中的参数具有较强的鲁棒性，但可能完全不能容忍数据网络的结构和参数的改变。因此在 NCS 的设计中，对数据包的丢失问题必须寻找相应的解决方法。

另一方面，网络中被传输的数据会经过众多计算机和通讯设备，而且路径不是唯一的，这必然会导致数据包的时序错乱。

数据包在传输过程中的时序错乱也可以分为单包传输和多包传输两种情况。在单包传输情况下，每一包数据便是一个完整的数据，此时的数据包的时序错乱是指从源节点按一定先后次序发送的多个完整的数据，当到达目标节点时，其时序与原来的时序不同；由于数据包长度受网络传输协议的限制，不能无限增长，而且控制系统的传感器和执行器常常分布在一个很大的物理空间里，需要经过不同的网络，在网关处需要拆分数据包，因此数据发送往往采用多包传输的形式。在多包传输情况下，一个数据被分成多个数据包进行传输，则数据包到达目标节点的时序可能与在源节点处拆分的时序不同。

传统的采样控制系统都是假设被控对象的输出和控制器的输入是同时进行的，而该同步性质在 NCS 中不再成立。因为一个数据要分成多个数据包传输，因此不可能同时传递，也不可能同时到达。除了数据分包外，被传输的数据沿着不是唯一的路径，经过多个交换设备转发，也会导致数据包的时序错乱。等待最后一个包的到达以及重现原始数据都会影响整个数据传输的延时。主要表现为，虽然已有多个数据包到达，但是在最后一个数据包到达前，所有数据包均需等待，这种情况也会造成网络延时的增大，导致系统控制的性能下降。

2.3 网络传输服务质量分析

网络控制系统是一个交叉性很强的研究领域，信息传输的延迟、抖动等都是单纯的网络性能参数，但同时也会影响控制算法、控制策略的分析和设计。网络控制系统的可控性、稳定性以及性能的提高都与控制回路中的服务质量密

切相关。

在网络控制系统中，对系统的性能和稳定性产生重要影响的因素是点到点传输延时^[9]，但是在网络系统中，延时的抖动、网络链路的带宽以及通路的差错率等网络服务质量(QoS, Quality of Service)因素与传输延时相互影响，相互制约^[9]，需要深入研究网络的服务质量才能准确地描述传输网络的延时情况。

网络服务质量是一个完全网络的概念^[22]，是指按照一定的控制协议，合理地分配网络中的资源，使信息能够按照要求准确地到达目的节点的技术。不同的网络结构、数据包长度、负载大小、数据包产生频率（传感器采样周期）等都会影响网络的服务质量。

无论控制网络采用的是现场总线网络(CAN、LonWorks等)，还是基于IP的Ethernet或Internet，网络模型都可以抽象为有权图（对称或非对称），应用图论知识进行分析，研究和描述。

在普通计算机网络中，将网络的交换节点、数据源和数据终点等抽象为有权图的节点，而将网络链路抽象为连接点与点的弧。由于控制网络与普通计算机网络的差别仅仅在于对实时性的支持程度上，因此将NCS的各节点包括控制器、传感器、执行器还有网络的交换设备，均视作抽象图中的节点，网络连线视作抽象图的弧，关于网络的图论分析方法仍然适用。

定义2.1 有向图 G ：由一个非空有限集合 $V(G)$ 和其中某些元素的有序对集合 $E(G)$ 构成的二元组，记为 $G=(V(G),E(G))$ 。其中， $V(G)$ 称为图 G 的节点集，元素 $v \in V$ 称为图 G 的顶点或节点； $E(G)$ 称为图 G 的边集，元素 $e_j \in E$ 记为 $e(v_i, v_j)$ 或 $e_j = v_i v_j$ ，为 V 中元素的有序对，称为图 G 的一条从 v_i 到 v_j 的边。

定义2.2 有权图 \tilde{G} ：在有向图 $G(V, E)$ 中，设元素 e 具有一组有序数列 (w_1, w_2, \dots, w_m) 作为它的属性或权，则称 \tilde{G} 为有权图，其中 (w_1, w_2, \dots, w_m) 称为弧 e 的度量（权值）。

节点集 V 可以表示网络中的任意节点； E 为连接 V 中两个节点的链路，且 $\forall e_i \in E$ 都有相应的属性集 (w_1, w_2, \dots, w_m) ，这些属性包括带宽、延迟、抖动、吞吐量、队列长度、花费等参数。

定义2.3 对称有权图 \tilde{G} ：对于有权图 $\tilde{G}(V, E)$ ，当 $\forall e_j, e_\mu \in E$ ，其中 $e_j = v_i v_j$ ， $e_\mu = v_j v_i$ ，时总有 $e_j(w_1, w_2, \dots, w_m) = e_\mu(w_1, w_2, \dots, w_m)$ 成立，则称 $\tilde{G}(V, E)$ 为对称有权图，记为 $\tilde{G}(V, E)$ 。

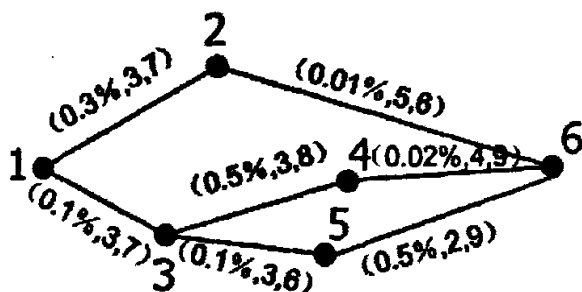


图 2.4 对称有权图模型 链路状态属性依次为 (丢包率, 延迟, 吞吐量)

定义2.4 数据通道 $P_{(i,j)}$: 在图 $G(V,E)$ 中, 不失一般性, 设 $i < j$, 称 $P_{(i,j)}$ 为 $v_i \in V$ 到 $v_j \in V$ 的数据通道或路径。其中 $P_{(i,j)} = P(v_i, v_{i+1}, \dots, v_j) = \{e_{k,k+1} \mid k = i, i+1, \dots, j-1\}$ 。

服务质量 (QoS) 是网络传输业务流时, 业务流对网络服务的需求集合。QoS 保证应用业务对网络传输服务所提出的一组可度量的要求, 主要包括带宽、端到端延迟、分组丢失率、抖动、花费的代价等, 网络传输相应的数据业务时, 必须满足这组要求。这组要求可分别用相应的网络代价进行数学描述^[23]。

衡量网络服务质量的主要属性包括^[24]:

- 传输延迟: 指两个参照点之间发送和接收数据包的时间间隔。
- 可变延时: 也称为延时抖动, 指在同一条路由上发送的一组数据流中数据包之间的延迟差异。
- 吞吐量: 网络中发送数据包的速率, 可用平均速率或峰值速率表示。
- 丢包率: 在网络中传输数据包时丢弃数据包的最高比率。数据包丢失一般是由网络拥塞引起的。

点对点 QoS 往往都是从单个的网络节点和连通弧的性质着手, 进而分析整条路径的 QoS。因此路径 $P_{(i,j)}$ 的 QoS 抽象值是路径上的所有节点和弧的 QoS 度量的函数, 这里假设此函数为 h , 且对于图 G 中的路径 $P_{(i,j)}$ 和各链路 $e_{i,i+1}$ 的延迟分别用 $D(P_{i,j})$ 和 $D(e_{i,i+1})$ 表示, 则整条路径 $P_{i,j}$ 的延迟表示为:

$$D(P_{i,j}) = h\{D(e_{i,i+1}), D(e_{i+1,i+2}), \dots, D(e_{j-1,j})\} \quad (2.1)$$

NCS 中各节点需要网络资源的确切时刻、占用网络资源的时间都不能提前预测。特别是当需要占用网络资源时, 节点会尽最大努力竞争, 希望能及时得到资源, 而在得到资源使用权后的大部分时间里却并不使用它们。因此, 用户对网络资源的需求是一种突发性的异步需求, 而等待信道空闲、信道竞争、数据

帧的冲突等均可能带来不确定型传输延时，这需要在控制网络的 QoS 控制机制中加以考虑。

业务流从源节点经过若干中间节点转发后，到达目的节点，如图 2.4 中路径 P(1, 3, 4, 6) 数据包从 1 至 6 的过程中，经过 3、4 两个中间交换设备的转发，这使得网络的 QoS 参数难以预测。因为数据包在经过这些中间节点时需要某些特殊的处理过程，这些过程对数据的不同 QoS 参数表现出不同的敏感性。

当网络中的一个节点与另一节点通信时，源节点的应用层软件首先将需要传输的原始数据按照应用层协议的规定打包，并将数据包传输给邻近的下层协议进程处理，直到数据包经过网络到达目的节点，以 Internet 中的 OSI 协议为例的数据包处理过程及延时如图 2.5 所示。

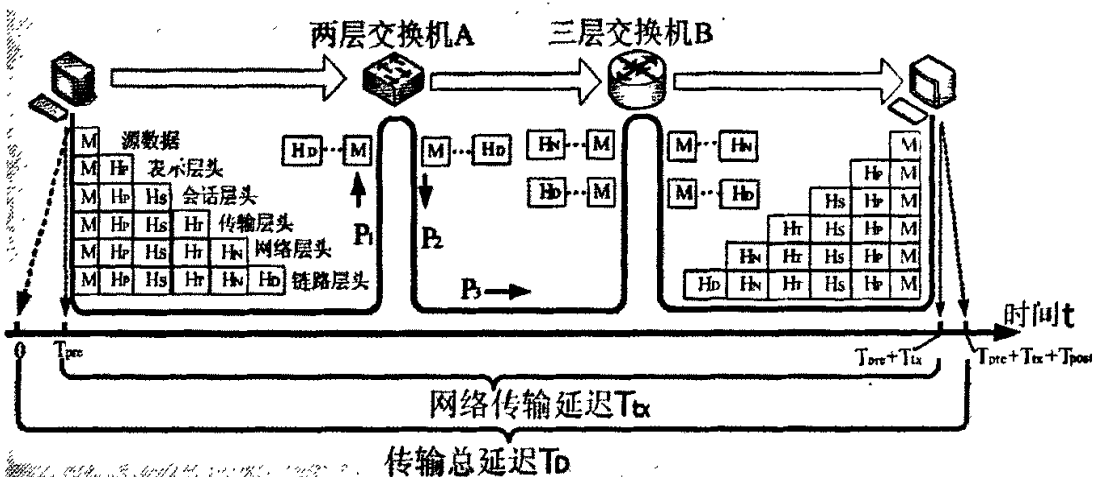


图 2.5 数据包在网络链路及节点中的协议处理模型

信息传输延迟在这一过程中可表示成三部分，即 $T_D = T_{pre} + T_{\alpha} + T_{pos}$ ，其中 T_D 为信息的总延迟； T_{pre} 为信息收集、打包并完全离开源节点进入链路的延迟； T_{α} 为信息完全进入网络链路在网络上传输直到信息完全传入目的节点为止的延迟； T_{pos} 为信息在目的节点上解包直到完全获得信息为止的延迟。其中 T_{α} 对 T_D 的影响最大，而 T_{α} 对所传输的数据包长度和密集度的敏感程度不同。

数据信息离开源节点，在到达目的节点前处于网络通道中。网络有暂存和处理数据包的能力，而网络 QoS 的研究目标是能够有效地控制网络缓存和转发特定业务流的信息，使之满足一定的 QoS 要求。如图 2.4 所示，业务流从数据源 1 出发经由 3, 4 两个网络转换节点到达目的节点 6，在路径 P(1, 3, 4, 6) 上存

储了 $n-m+1$ (假设 $0 < m < n$ 为整数) 个数据包, 如果不考虑网络上其他数据流的交叉传输, 那么网络为此业务流提供的 QoS 能力与交换节点 3, 4 的处理能力和路径 $P(1, 3, 4, 6)$ 的带宽紧密相关。可以将节点 3, 4 之间的网络链路看作处理节点 3 的出口缓存, 即将节点 3 和边 $e_{3,4}$ 视为一体。

从数据包自 $e_{1,3}$ 完全转交给 3, 到经过 3 和 $e_{3,4}$ 并完全转交给 4 之前, 数据包所经历的处理过程为: 处理分析过程、排队过程、传输过程和传播过程, 如图 2.6 所示。

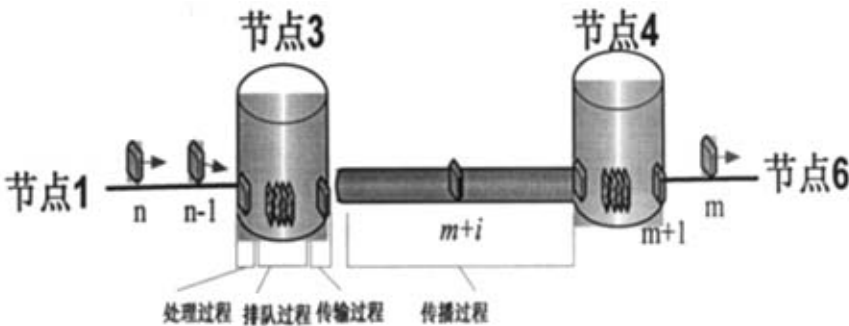


图 2.6 节点间影响数据包 QoS 的各处理过程

处理分析过程: 数据包 k 进入 3 后需要检查数据包的头部信息, 以确定数据包的正确出口, 除此之外还包括数据包的差错校验过程等, 这一过程对应于图 2.5 中的处理过程 P_1 , 这一过程一般在微秒级, 主要影响数据的延迟, 与数据包长度 l 和 3 的处理器速度有关。

排队过程: 无论是交换机还是其他网络转发设备, 对应每个出口都有一个专用内存区, 用于数据排队, 这一过程的延时长短取决于网络拥塞状况, 也是与网络 QoS 联系最为密切的一个过程。排队时间的长短会直接影响数据包的传输延迟, 队列长度的变化会引起业务流的延迟抖动。如果队列过长, 还会导致丢包。当队列中所有排在第 k 个包之前的数据包都已经传输完毕后, 那么从此时开始数据包 k 从排队过程进入传输过程。

传输过程: 传输过程与边 $e_{1,3}$ 的吞吐量 R 和数据包 k 的长度 l 有关, 这一过程主要影响数据的延迟, 需要的时间为 l/R 。数据包 k 从 3 被完全送到边 $e_{1,3}$ 上后进入传播过程。

传播过程: 在此过程中, 数据包 k 从节点 3 经由物理链路传输到节点 4。与

传输过程不同,这一过程主要影响业务流的延迟和丢包率。数据信息在物理链路上的表现形式为 0 和 1 的组合,更进一步表现为各种物理信号(如电压的高低交替或光信号)组合,因此这一过程对延时的影响与数据包的大小无关,而与物理链路的长度有关。如在光纤中传播的分组每 200 千米延时大约会增加 1 毫秒,每个分组经过一个高速路由器延时大约会增加 10 微妙^[22]。

对控制网络 QoS 产生影响的主要因素包括以下三方面:网络模型(包括拓扑结构、链路带宽、链路长度等)、传输模型(调度策略、协议、媒介访问策略等)、负载模型(包括数据包长度、发送间隔、校验复杂度、突发数据流量和负载循环规律等)。

2.4 本章小结

本章内容介绍了本课题的研究重点——NCS 控制网络的相关知识。综述了主要的控制网络结构,分析了基于 Internet 的网络控制系统中存在的延迟、丢包、数据包排列顺序混乱等问题,研究了网络传输服务质量特性 QoS 以及相关的计算机网络图论分析方法,描述了网络控制系统中信息传输的延迟的组成。

第3章 控制网络多路径传输模式的相关原理

3.1 多中心化的对等网络模式

当前，互联网的发展正在经历一场影响力可与引入图形界面浏览器相提并论的全新应用模式的革命。这种应用模式将网络中不同的计算机连接在一起，通过在系统之间的直接交换来共享计算机资源和服务，称为对等网络 (Peer to Peer 简称 P2P) 模式。

第一代互联网依靠全局共享的通信媒介来动态地访问远程信息；随着图形界面浏览器的诞生，第二代互联网模型采用了中心化结构的客户机/服务器 (C/S) 通信模式；然而由于互联网规模的扩大，网络节点的增多，驻存了大量通信内容的服务器越来越难以承载负担，于是“资源在哪里创建，就到哪里访问”的愿望促进了互联网进入第三个发展阶段——分布式计算。比如，当 100 万个客户要同时访问一个站点，WWW 模型必须构建一个高配置的服务器来满足所有需求；而分布式计算模型可以将此站点分解到 100 万台普通 PC 上，通过 Internet 上的协作来进行。这一方案隐含着两个显著的改变：搜索分布在网络上进行，内容驻留在边缘（而非中心服务器）。（如图 3.1 所示）

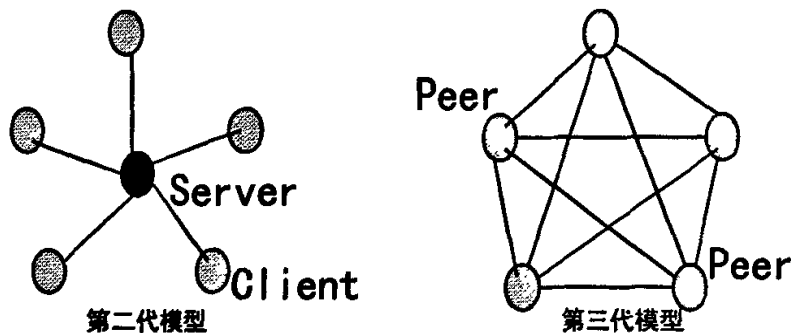


图 3.1 网络模型示意图

Intel 公司将 P2P 系统定义为“通过系统间的直接交换所达成的计算机资源与信息的共享”，这些资源与服务包括信息交换、处理时钟、缓存和磁盘空间等。

Roku Technologies 公司将 P2P 系统定义成“使个人与个人之间直接通信成为可能且更便捷的网络结构”。IBM 公司则给 P2P 赋予了更广阔的定义，把它看成是由若干互联协作的计算机构成的系统并具备若干特性。

P2P 引导网络计算模式从集中式向分布式偏移，也就是说网络应用的核心从中央服务器向网络边缘的终端设备扩散：服务器到服务器、服务器到 PC 机、PC 机到 PC 机，PC 机到 WAP 手机……所有网络节点上的设备都可以建立 P2P 对话。这使得在 Internet 上的群体通信行为提到了一个更高的层次。

在对等网络中，每个节点的地位都是相同的，可以同时作为服务使用者和服务提供者，因此对等 P2P 网络的节点具有很高的功能自治性和结构随意性。

3.1.1 对等网络结构特征

客户/服务器(Client/Server 简称 C/S)模式兴起于 20 世纪 90 年代初，到了 90 年代的中期开始成为流行的使用的模式。它的基本思想是把集中在一起的应用划分成功能不同的两个部分，分别在不同的计算机上运行，通过它们的分工合作来实现一个完整的功能。

其中，服务器程序的功能是用来响应和提供因定的服务；客户机程序的功能是用来向服务器提出请求和要求某种服务。“服务器”和“客户机”是软件的概念，是相对的概念。比如一台计算机上可以同时运行着一个客户机程序和一个服务器程序，它同时充当着客户机和服务器。一般服务器程序总是运行在性能较强、配置较高的计算机上，而且服务器通常可以同时响应几个客户机的请求。

从服务器所提供的服务类型角度，可分为文件服务器、打印服务器、数据库、服务器和通信服务器等。网络控制系统最主要是通信服务器。服务器通过以下步骤和客户进行交互：

第一步，监听等待一个客户的请求的到来。

第二步，启动一个新的服务进程来处理客户的请求，而原来的服务器进程则继续监听是否有新的客户请求的到来。客户的请求处理结束后，这个新的进程被终止。

第三步，返回到第一步。

采用 C/S 模式结构通信具有如下特点：

(1) 信息和数据都保存在服务器端。只有服务器端具有控制能力，客户端基本上只是一个高性能的终端设备；

(2) 服务器及网络的带宽决定了网络的性能。每台服务器所能提供的信息数量受到自身存储空间的限制，而且任意时刻它所能支持的客户端访问数量则既受到自身处理能力的限制，也受到服务器所在网络吞吐能力的限制；

(3) 用 URL 来表示信息资源的地址。但是 URL 很少能直接体现所定位的信息的内容甚至不能直接链接到具体的内容上；

(4) 待发布信息的分布与生存期由服务器控制。服务器只发布机器所有者想公之于众的信息，这些信息将会在该服务器上稳定地保存一段时间，并且该服务器通常也不间断地运行在网络上；

(5) 待发布信息的存贮与管理是集中制的。互联网上所有可以公开访问的信息基本上都保存在服务器上，服务器根据适当的算法和规则管理本地信息应答客户端的访问请求或进行管理；

对等网络模式的非中心结构使它与 C/S 模式有着明显的差别，其特点如下：

(1) 每一个对等点具有相同的地位，既可以请求服务也可以提供服务。同时扮演着对等模式中的服务器和客户端两个角色。还可以具有路由器和高速缓冲存储器的功能，从而弱化了服务器的功能甚至取消了服务器；

(2) 对等网络技术可以使得非互连网络用户很容易地加入到系统中。在对等网络计算环境中，任何设备从大型机到移动电话甚至是传呼机均可以在任何地点方便地加入进来，P2P 技术不仅可以应用于目前有线的互联网络，同时该技术还可以应用于无线计算技术；

(3) 在对等模式的网络中每一个对等体可以充分利用网络上其他对等体的信息资源、处理器周期、高速缓存和磁盘空间等；

(4) 对等网络是基于内容的寻址方式。这里的内容不仅包括信息的内容，还包括空闲机时存储空间等。对等网络中用户直接输入要索取的信息的内容而不是信息的地址，P2P 软件将会把用户的请求翻译成包含此信息的节点的实际地址，而这个地址对用户来说是透明的；

(5) 对等网络中的每一个对等体通常没有固定的 IP 地址，并且有自由加入退出网络的特性；

对等网络模式的目的是充分利用互联网和 Web 站点中的资源，将网络中不同的节点连接在一起。对等网络模式在结构和功能上体现了多中心化的特点，

在网络中每个节点的地位都是对等的，既充当网络资源的提供者 (Server)，又是资源的获取者 (Client)。在多中心化的模式下，用户通过分布式的对等计算，协同工作，实现高效率的搜索发现和文件交换等工作，使互联网上信息的价值得到极大提升，真正实现了互联网资源共享和状态自由的理想。

3.1.2 对等网络节点路由机制

在对等应用程序中，每个对等点都可以进入或退出，在他们活动状态期间，尝试完成分配到的任务，大多数任务都要涉及与其它对等点交互。节点查找是进行信息存储和查询的基础，所有可行的对等网络体系结构都提供一种针对发现问题的解决方案，不需要依赖传统的服务器。

1 静态机制

按照术语点到点(point-to-point)的本义，系统网络中每个对等节点都知道其它对等点，并与之相连。每个对等点必须预先配置其它所有对等点的地址。通过对网络中的每个对等点预先设置其它所知道并且将要与之交互的对等点列表进行通信，使得网络在外部攻击面前表现得很稳固。这种路由机制的主要优点是简单，它的主要缺点是缺乏灵活性并且缺少扩展到具有较多节点对等点的大型网络的能力。

2 目录式机制

与显式点到点配置的静态特性相反，具有动态特性的目录服务模型利用一个或多个有特殊用途的服务节点为其他对等点提供目录服务(图 3.1)。对等节点向目录服务节点注册自身的信息(如名称、地址、资源和元数据等)，并根据服务节点目录中信息来定位其它对等点。由于服务节点保存着网络中所有活动对等端点地址信息及其共享资源的目录信息，当需要查询某个文件时，该对等节点会向一个服务节点发出文件查询请求。经过在服务节点之间进行的相应检索和查询后，返回符合查询要求的对等端点信息列表。发起查询的节点接收到应答后，再根据网络流量和延迟等信息进行选择，与适合的端点直接建立连接，开始文件传输。

这种网络节点路由机制具有客户机/服务器模式的中心化结构特点，但是它与传统意义上的 C/S 模式不同。C/S 模式的通信内容都存放在服务器上，客户机只能被动地从服务器上读取信息，而且客户机之间不具有交互能力；目录式机

制的数据资源都存放在提供该数据的客户机上，服务节点只保留索引信息，相应的对等节点之间具有交互能力。而且为了均衡对等网络的负载，服务节点可以只担当目录而不作它用。对等网络应用领域中目录模型的最佳示例是 Napster。当 Napster 的使用者登录 Napster 的中央服务器时，Napster 客户端程序会送出共享文件的清单到 Napster 的服务器里。服务器软件会把这个清单整合到一个存储歌名和乐手姓名的巨大数据库，没有真正的 MP3 文件存储在服务器中，只有 Napster 使用者所提供的下载目录。另外，使用者以歌名或是乐手姓名为搜寻条件，Napster 的服务器会搜寻数据库并传回一个可以下载到这些资料的电脑清单。一旦点选了想要下载的档案，Napster 使用者的计算机便会直接建立和远端计算机连接以进行档案传输，不需要通过中央服务器通讯。这种路由机制的主要优点是发现效率高，易于维护和扩展，最大的问题是容易造成单点故障，产生访问瓶颈。^[10]

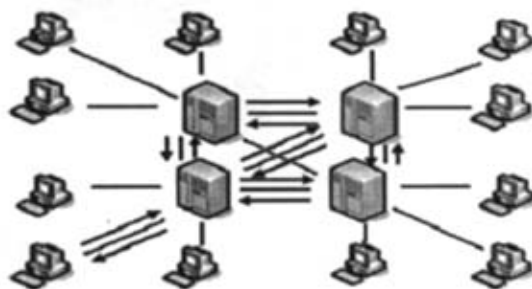


图3.1 目录式节点路由机制示意图

3 泛洪式路由机制

该机制集中体现了 P2P 网络的“对等”概念，每个节点在功能上都是相似的，并没有专门的服务器，而对等端点必须依靠它们所在的分布网络来查找文件和定位其它端点并与之相连。没有一个对等节点知道整个网络的结构或者组成网络的所有对等点的身份，只是知道直接与它们通信的对等节点。当需要查询某个文件时，对等节点首先网络中所有活动的向与之相邻的对等节点发送一个查询请求包 (Query)。这些对等节点端点在接收到该查询请求包后，先检查本地是否有符合查询请求的文件内容，如果有，则按查询请求包的发送路径返回一个查询响应包。无论本地是否存在符合查询请求的文件内容，其它对等端点都会将该查询包通过广播方式继续在网络中传递，直至查询包中 TTL (Time of Life) 属性值递减为 0 时停止继续转发 (图 3.2)。对等网络应用领域中泛洪式机

制的最佳示例是 Gnutella^[10]，它可以让使用者不必透过中央服务器的介入，就直接在两台独立的个人计算机上共享文件，真正做到纯粹的对等式传输。由于网络采用了确定性拓扑结构，可以提供精确的发现，也使得网络在外部攻击面前表现得很稳固，增强了安全性。但这种全分布式的拓扑结构不能很好地扩展到具有较多节点的大型网络，传播查询耗费了大量的网络和处理能力。对等点频繁地加入退出会造成网络波动，极大地增加维护代价。

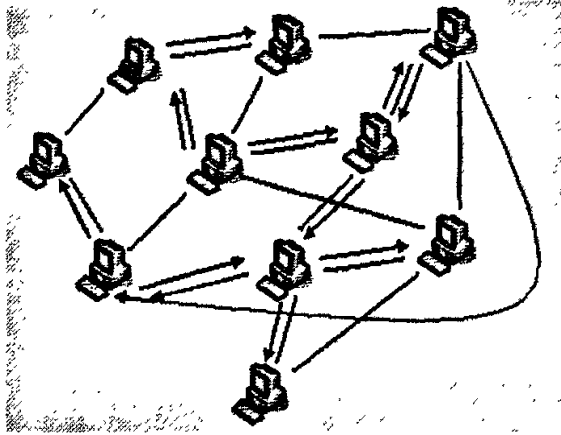


图 3.2 泛洪式节点路由机制示意图

4 混合式路由机制

混合式路由机制结合了集中式和分布式的优点，在设计思想和处理能力上都得到了进一步的优化。它首先将网络中的多个物理节点组合成一个虚拟节点，大大减小了网络的规模层次。然后在分布式模式的基础上，将用户节点按能力进行分类，使某些节点担任特殊的任务。多数可能频繁地加入和退出网络系统的作为普通的用户节点，不具有任何特殊的功能；那些连接速度快、内存充足的节点作为索引节点，被赋予处理搜索请求、保存可以利用的搜索节点信息、以及搜集和维护网络的状态和结构信息的任务。

索引节点不会直接连接到相关的资料上，它就像搜索引擎一样，只是搜索和所需资料相关的地址，至于用户到底连接下载了什么内容则和它无关；索引节点同时管理着所属用户的文件列表。当用户节点发出搜索请求后，在较低的层次的网络中为了取得较快的查找速度，采用泛洪路由机制；在较高的层次的网络中，通过让索引节点以 IP 多播方式定期在对等网络中发布消息来查找目标节点。多播是在网络中实现一对多、多对多的技术。多播数据只向那些需要数

据包的主机和网络发送数据包(以多播地址寻址);多播数据在网络中是沿着多播生成树分发数据的,在共享的链路上,相同的信息只需要一个多播流。从而能够很好地控制流量,减少了主机和网络的负担,提高了网络应用服务的效率和能力。

索引节点一方面要接受别的索引节点的服务,另一方面还要向别的索引聚集节点和用户节点提供服务;用户节点则不转发来自别的节点的请求,仅仅在本地检查、请求并回复。

这是中心化和分布式模式的结合,避开两者的缺点,集取他们的优点。目前的第三代对等网络系统中,一般都采用混合式 P2P。

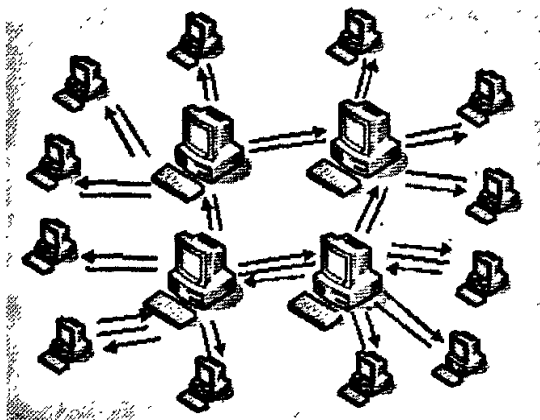


图 3.3 混合式节点路由机制示意图

本设计提出的多路径传输模式,就是基于互联网的分布式对等模型发展趋势,借鉴了对等网络中多中心化的节点分布架构所提出的新型控制模式。本设计提出的路由管理器节点具有提供路由指引,维护路径信息库的职能,是参照了对等网络节点目录式路由机制的架构和工作原理。因此,多中心化的对等网络模式是对本设计起到关键技术指导作用的理论依据之一。

3.2 网络传输原理及互联技术

3.2.1 网络的体系结构

作为网络控制系统的信号传输介质,计算机网络本身也是一个涉及计算机

技术、通讯技术等多项研究领域的复杂系统。网络中的各部分都必须遵照合理而严谨的结构化管理规则。计算机网络是以资源共享、信息交换为根本目的,通过传输介质将物理上广为分散的计算机系统、智能终端、外围设备、网络通信设备等互联成的网络系统。为了完成计算机之间的通信合作,把每台计算机几乎连的功能划分成有明确定义的层次,并固定了同层次的进程通信的协议及相邻层次之间的接口及服务,这些层次进程通信的协议及相邻层的接口就统称为网络体系结构。

在计算机网络中实现通信,必须依靠网络通信协议。在 20 世纪 70 年代,各大计算机生产商的产品都拥有自己的网络通信协议。但是不同的厂家生产的计算机系统就难以连接,为了实现异构计算机系统和异构网络的网络互连,以达到相互交换信息、共享资源的目的,推动计算机网络向更统一、更大规模发展,1979 年国际标准化组织 ISO 提出了开放系统互连参考模型(即 OSI 参考模型)。OSI (Open System Interconnection) 采用分层的结构化技术将计算机网络的体系结构划分为七层:由低到高依次为物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。各层具体功能如下:

物理层:提供二进制传输,为数据链路层提供物理连接,确定如何在通信信道上传输比特流;

数据链路层:提供介质访问,加强物理层的传输功能,提供在两个相邻节点间无差错地传送以“帧”为单位的数据的线路;

网络层:提供 IP 寻址和一般意义的路由,通过互连网络来路由和中继数据,负责寻找最佳传输线路;

传输层:为源端主机到目的端主机提供可靠的数据传输服务,隔离网络的上下层协议,使得网络应用与下层协议无关,提供终端到终端的可靠连接;

会话层:在两个相互通信的应用进程之间,为两个主机上的用户进程建立会话链接,并使用这个链接进行通信,使双方操作相协调,处理被传送数据的表示问题,即信息的语法和语意,在多种数据表示格式之间进行一种转换,还有数据的加解密,压缩与解压缩等;

应用层:为用户的应用程序提供网络通信服务;(如图 3.4 所示)

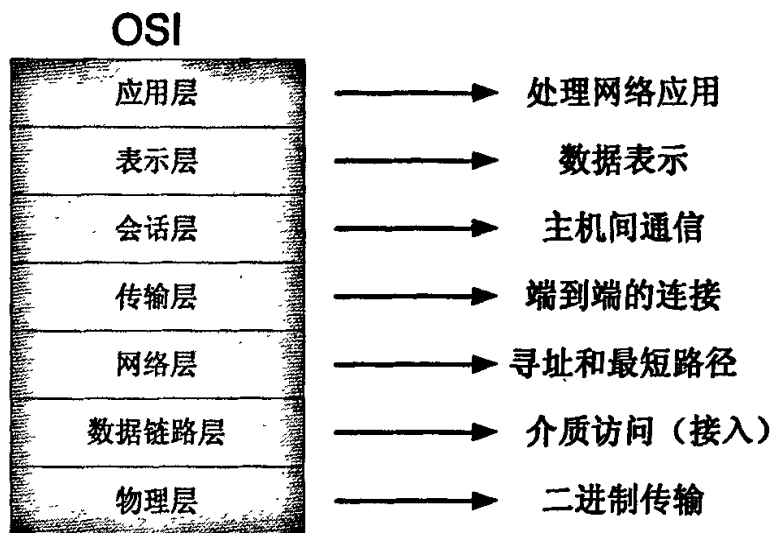


图3.4 OSI参考模型层次功能图

OSI 模型用途相当广泛。比如交换机、集线器、路由器等很多网络设备的设计都是参照 OSI 模型设计的。在数据发送到另一层时，都要分成数据包。数据包是一个信息单位，作为一个整体，从网络中的一个设备传送给另一个设备。

数据包的创建过程是从 OSI 模型的应用层开始的。要发送的信息从应用层开始，往下依次穿过各层。每层都对数据包进行重新组装，以增加自己的信息（信头）；要接受的信息从物理层开始，往上依次穿过各层。每层对数据包进行重新拆分，除去信头，最终得到来自发送方应用层的原始信息。

3.2.2 TCP/IP 模型

网络设计者在解决网络体系结构时经常使用 ISO/OSI 七层模型，但 TCP(Transmission Control Protocol)/IP(Internet Protocol)是目前网络中应用最广泛的一种基本的通信协议。同 ISO/OSI 比较，TCP/IP 更加简单、实用、高效和成熟（如图 3.2），以 TCP/IP 为网络协议建立起来的 Internet 发展十分迅猛，TCP/IP 已经成为当今网络协议事实上的标准。^[20]

通常说 TCP/IP 是 Internet 协议族，除了传输控制协议(TCP)和网际协议(IP)外，TCP/IP 较低层的协议有 IP 地址解析协议 ARP、反向地址解析协议 RARP，IP 差错与控制协议 ICMP，路由协议、用户数据报协议 UDP 等。TCP/IP 应用层的协

议有文件传输协议 FTP, 远程登录协议 Telnet, 简单邮件专用协议 SMTP, 简单网络管理协议 SNMP、网络文件系统 NFS 系统。TCP/IP 各层的结构如图 3.2 所示:

OSI	TCP/IP	协议
应用层	应用层	FTP, Telnet, SMTP, DNS, SNMP等
表示层		
会话层		
传输层	传输层	TCP, UDP
网络层	网际层	IP, 路由协议, ICMP, ARP, RARP
数据链路层	网络接口层	网络接口卡 (以太网, 令牌环网等)
物理层		传输介质 (双绞线, 同轴电缆, 光纤等)

图 3.2 TCP/IP 模型及协议框架

在 Internet 上传输控制协议和网际协议是配合进行工作的。网际协议 (IP) 负责将信息从一个主机传送到另一个主机。信息在传送的过程中被分割成一个一个的数据包。传输控制协议 (TCP) 负责收集这些信息包, 并将其按适当的次序传送, 在信息接收端收到信息包后再将其正确地还原。尽管计算机通过安装 IP 软件, 保证了计算机之间可以发送和接收数据, 但 IP 协议还不能解决数据分组在传输过程中可能出现的问题。因此, 连上 Internet 的计算机还需要安装 TCP 协议来提供可靠的并且无差错的通信服务。Internet 是一个庞大的国际性网络, 网路上的拥挤和空闲时间总是交替不定的, 加上传送的距离也远近不同, 所以传输数据所用时间也会变化不定。TCP 协议具有自动调整“超时值”的功能, 以适应 Internet 上各种各样的变化, 确保传输数据的正确。

网络层及其下层仅仅提供了两个相邻节点之间的点到点的传输, 网络上的每台计算机或网络设备都分配到一个 32 位的数字标识, 称为 IP 地址。IP 地址分 A, B, C, D, E 五类, 地址分两字段, 前为网络号, 后为主机号。为减轻网络地址的巨大开销和网关寻径的负担, 采用子网掩码技术, 实现 IP 网络地址的多重复用。IP 的这种点到点特性没有应用程序的概念, 在多任务操作系统如 WINDOWS 中, 数据传到信宿机时该由哪个进程来处理到达的数据, IP 是无能为力的。IP 之上的传输层有传输控制协议 TCP 能够解决这个问题, 它提供了信源机和信宿机之间端到端的通信, 即提供了应用程序之间的通信, 为上层协议建立了一个完善的服务接口。

传输控制协议 (即 TCP) 建立在 IP 的基础之上。TCP 是一个面向连接的协议,

它指定数据格式和对传输数据的确认。TCP 允许在一个系统上有多个应用并行通信，因为它能处理应用程序间传输的信号分解。TCP/IP 协议的基本通信模式如图 3.3 所示。

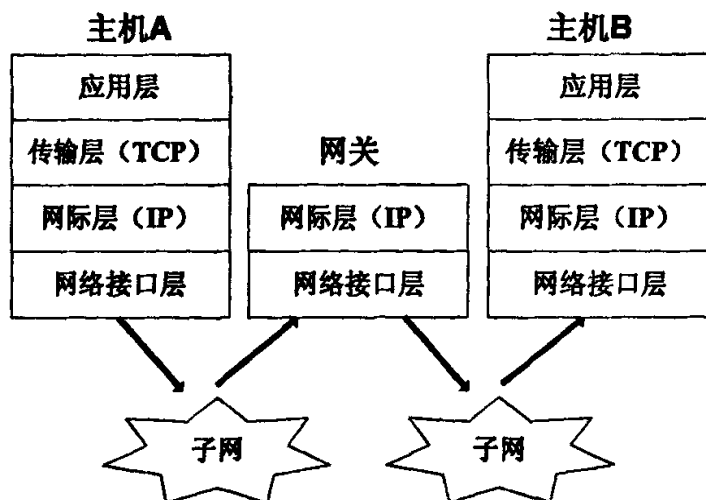


图3.3 TCP/IP及协议框架

本设计提出的多路径传输模式，是工作在开放系统的应用层之上的一种路由服务，通过让系统的多个应用进程实现相互通信，来完成一系列指定的业务处理；同时，如上文所述，本设计使用的 TCP/IP 的通信模型是一种有连接的、可靠的协议，它将输入的字节流分成报文段，还能处理流量控制，以避免快速发送方向低速接收方发送过多的报文而使接收方无法处理。与没有排序和流量处理能力的 UDP (User Datagram Protocol) 协议相比，能够使本设计的性能更加可靠完善。

值得一提的是，目前 Internet 使用的 IP 协议是 IPv4 (Internet Protocol version 4)。然而互联网规模的不断扩大和迅速发展使得 IPv4 定义的有限地址空间将被耗尽，于是为了能够扩大地址空间和提高网络的整体吞吐量，IPv6 作为下一版本的互联网协议被提出。本设计提供的路由服务是用实名来表示网络节点，后台的路径信息库来保存节点实名与 IP 地址间的映射关系，直接面向应用层进行的。随着互连网络 IP 技术的不断发展，本设计也可以进行相应的技术升级更新。

3.2.3 网络通信原理及 Socket 套接字

网络通信实际上就是网络中的主机之间进行的信息交换过程。如果从 TCP/IP 协议栈的传输层上来观察网络中的通信，网络中的通信可以分成两类，一种是 TCP 协议类、另一种是 UDP 协议类。

TCP 提供一种面向连接的、可靠的字节流服务。在一个 TCP 类的通信中，通信双方首先必须经过协商建立起一个逻辑上的连接，在通信的全过程中，该连接可能包含有非常多的报文，这些报文都具有相同的源 IP 地址、目的 IP 地址、源端口号、目的端口号、协议类型等属性。而 UDP 提供一种无连接的、不可靠的服务。尽管一次网络通信活动中的各个 IP 数据包都是单独路由的，但是如果从传输层的角度来看，网络通信其实可以看作是由节点及节点之间建立的网络连接构成的。其中：

(1) 节点就是指网络中的通信主机。它能够发送或接收数据，具有 IP 地址、系统类型、MAC 地址等各种属性。

(2) 网络连接就是指数据源节点和目的节点之间进行的一次通信活动，表达了每次通信过程中的所有网络流量的统计信息，具有源节点、目的节点、源端口、目的端口、开始时间、结束时间等属性。

对于 TCP 协议而言，一次网络连接是指从连接建立(三次握手)开始，到连接结束(四次握手)为止的一段通信过程，同时虽然在 TCP 类的连接中，数据流动是双向的，即通信双方都可以发送或接收报文，但是我们把网络连接的方向定义为从最初发起连接建立请求的一方指向另一方。

这种传输层的网络连接屏蔽了 TCP/IP 协议栈中传输层以下的各层在通信实现上的具体细节，使得报文传送的过程(如路由转发、分片重组等)对于通信双方而言是透明的，从发送方到接收方的数据包传送路径实际上可以看作是一条简单的直通链路。这样，就可以在网络连接这个层次上在对网络通信进行分析。

[21]

为了在不同计算机之间使用 TCP/IP 协议进行网络通信编程，Berkeley 大学最早开发了针对 UNIX 操作系统的套接字 Sockets 规范。随着计算机操作系统的发展，Windows 操作系统也引入了这种接口，它不仅包含了 Berkeley sockets 风格的库函数，也包含了一组针对 Windows 的扩展库函数，以便能利用 Windows 消息驱动机制进行编程应用程序调用 Sockets 的 API 实现相互之间的通讯，而

Sockets 又利用下层的网络通讯协议功能和操作系统调用实现实际的通讯工作。它们之间的关系如图 3.4 所示。

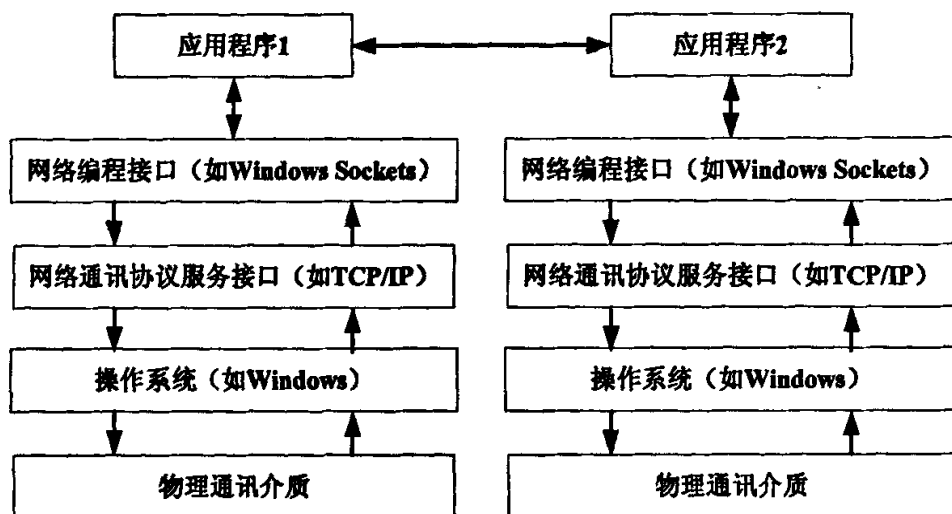


图 3.4 应用程序和 Socket 的关系

套接字 Socket 是网络通信的基本构件，一个 Socket 对应于网络通信的一端。网络通信的 Socket 接口模型将通信主机或进程当作端点，每次网络对话包括两个端点：本地主机（或进程）和远程主机（或进程）。Socket 接口将网络对话的每个端点称为一个 Socket。Socket 是可以被命名和寻址的通信端点，每一个正在被使用的 Socket 都有它的类型和与其相关的进程。

套接口存在于通讯域中。通讯域是为了处理一般的线程通过套接口通讯而引进的一种抽象概念。套接口通常和同一个域中的套接口交换数据。Windows Sockets 规范支持单一的通讯域，即 Internet 域。各种进程使用这个域互相之间用 TCP/IP 协议族来进行通讯。套接口可以根据通讯性质分类；这种性质对于用户是可见的，应用程序一般仅在同一类的套接口间通讯。不过只要底层的通讯协议允许，不同类型的套接口间也照样可以通讯。用户目前可以使用两种套接口，即流套接口和数据报套接口。流套接口提供了双向有序的、无重复并且无记录边界的数据流服务；数据报套接口支持双向的数据流，但并不保证是可靠、有序、无重复的。

TCP/IP 的套接字有三种类型：数据报套接字 (SOCK_DGRAM)、流式套接字 (SOCK_STREAM) 和原始套接字 (SOCK_RAW)。

数据报套接字就是在传输层利用了不可靠的 UDP 传输协议提供一种不可靠的、非连接的数据包(Packet)通信方式,这种不同向连接的编程相对比较简单。发送一个数据包不能获得担保,也不能保证数据包按照发送的顺序到达目的地。数据报文以独立数据包的形式被发送,不提供无错保证,数据包可能在传送过程中丢失或重复,并且接收顺序混乱。

如果想让发送出去的数据按照顺序无重复地到达目的地,那么就需要使用流式套接字。流式套接字提供了一种可靠的面向连接的数据传输方法,数据无差错、无重复地发送,且按发送的顺序进行接收。不管对单个的数据报,还是对整个的数据报,流式套接字都提供了一种流式数据传输,流式套接字使用传输控制协议 TCP。此外,在数据传输是,如果连接断开,应用程序会被通知。流式套接字内设流量控制,避免数据流超限,数据被看作是字节流,无长度限制。

原始套接字允许对较低层协议,如 IP、ICMP 的直接访问,它常用于检验新的协议实现或访问现有服务中配置的新设备。原始套接字提供对网络下层通信协议(如 IP 协议)的直接访问,它一般不是提供给普通用户的,超级用户才能创建原始套接字。原始套接字主要用于开发新的协议或者用于提取协议较隐蔽的功能。

套接字编程均采用客户机服务器的协作模式,即由客户进程向服务器进程发出请求,服务器进程执行被请求的任务并将响应结果返回给客户进程。TCP 协议提供了一种可靠的面向连接服务,数据无差错、无重复地发送,且按发送的顺序进行接收。客户端进程与服务器进程交互的过程如图 3.5 所示。

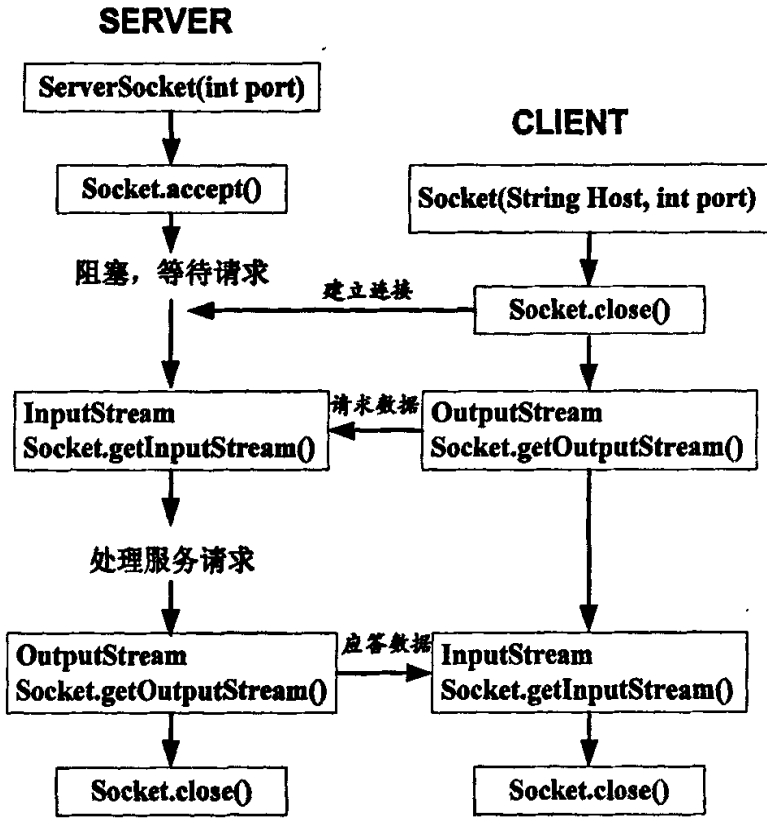


图3.5 面向连接的Socket通信流程

面向连接的 Socket 系统，服务器端必须先启动。系统调用次序为：

- 1) 服务器要创建一个用于监听的套接字，为该套接字分配地址之后，调用监听函数使它处于监听状态；
- 2) 客户机在创建套接字完毕后，为套接字分配地址，然后调用连接函数，请求与服务器套接字连接；
- 3) 服务器套接字在收到客户机的连接请求后，调用接收函数来创建一个用连接的套接字；
- 4) 应用服务器的连接套接字和客户机的套接字，就可以在服务器和客户机之间进行数据的传输；
- 5) 在结束传输之后，客户机调用关闭函数来关闭套接字，服务器也调用关闭函数来关闭连接套接字以及用于监听的套接字。

面向连接服务处理的请求响应比较复杂，一般是并发服务的，而不像无连接的服务器一个请求一个应答就完成了客户程序与服务程序之间的相互作用。

本设计中的软件模拟原型就利用了 JAVA 的 .net 包提供的基于 TCP 协议 Socket 套接字编程工具来模拟网络节点间的通信过程，实现数据包的可靠传输。下一章中将更加详细地解析如何用 JAVA 语言编程方式实现 Socket 通信的过程。

3.2.4 路由技术原理

所谓路由就是指通过相互连接的网络把信息从源地点移动到目标地点的活动。一般来说，在路由过程中，信息至少会经过一个或多个中间节点。早在 40 多年之间就已经出现了对路由技术的讨论，但是第一代互联网依靠全局共享的通信媒介来动态地访问远程信息，网络结构都非常简单，路由技术没有用武之地。直到最近十几年，大规模的互联网络才逐渐流行起来，为路由技术的发展提供了良好的基础和平台^[34]。

路由技术其实是由两项最基本的活动组成，即决定最优路径和传输信息单元（也被称为数据包）。其中，数据包的传输和交换相对较为简单和直接，而路由的确定则更加复杂一些。

大型网络如因特网，会被分解成为多个自治系统 AS (Autonomous System)，如图 3.6 所示。每个自治系统是一个自我管理的网络，一个自治系统只负责管理自己内部的路由。内部网关路由协议是指运行在一个自治系统内部的路由选择协议。常见的例子有路由信息协议 RIP (Routing Information Protocol 路由信息协议)，开放式最短路径优先 OSPF (Open shortest Path First, OSPF) 等。

在自治系统之间的路由选择协议被称为外部网关路由协议。如 BGP 边界网关协议 BGP (Border Gateway Protocol) 等。这些协议工作在自治系统之间，用以确保自治系统之间的通信^[36]。

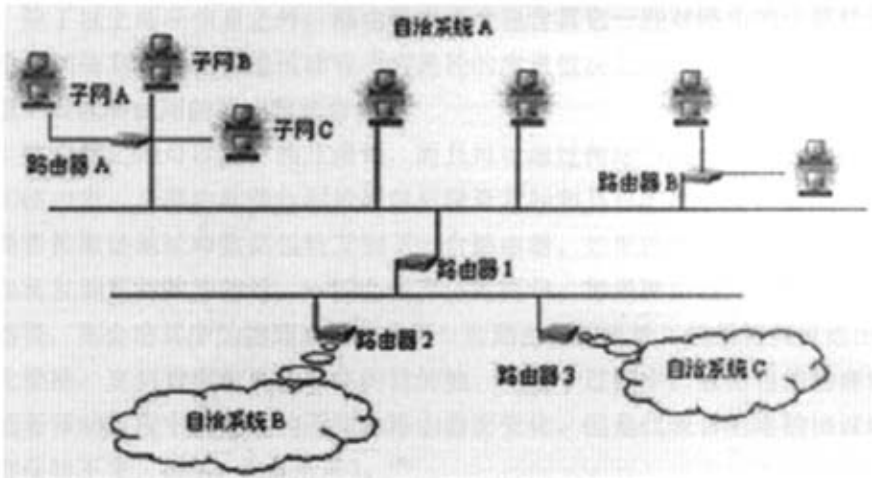


图3.6 内部和外部网关路由协议的使用示意图

度量标准 (metric)，例如路径长度等，是被路由算法用来计算和确定到达目的地的最优路径的标准。为了帮助确定数据传输的路径，路由算法可以建立和维护路由表。路由表中包含了各种路由信息。路由信息根据所使用的路由算法的不同而各异。

路由算法在路由表中写入各种不同的信息，路由器会根据数据包所要到达的目的地选择最佳路径把数据包发送到可以到达该目的地的下一台路由器处。当下一台路由器接收到该数据包时，也会查看其目标地址，并使用合适的路径继续传送给后面的路由器。依次类推，直到数据包到达最终目的地。表 3.1 所示的就是这种通过目的地和路由器地址决定最佳传输路径的路由表。

表 3.1 基本路由表

To Reach Network	Send To:
27	Node A
57	Node B
17	Node C
26	Node A
.....

除了以上两项信息之外，路由表中还会包含其它一些对路由的计算和选择有价值的信息。路由器通过比较不同路径的度量值决定最优路径，而具体的度量值则要视所使用的路由算法而定。

路由器之间可以进行相互通讯，而且可以通过传送不同类型的信息维护各自的路由表。当路由器接收到数据包后将查看标明其目的地的协议地址，并决定是否按照该地址将数据包转发到下一台路由器。如果路由器不知道如何把数据包转发到其目的地的话，一般会丢弃该数据包。如果路由器知道数据包的转发路径，则会将其中的物理地址改为下一台路由器的地址，然后将其发送出去。以此类推，直到数据包到达最终的目的地。在整个过程中，数据包的物理地址会随着移动过程中所经过的不同的路由器而变化，但是代表目的地的协议地址一直保持不变（如图：3.7 所示）。^[36]

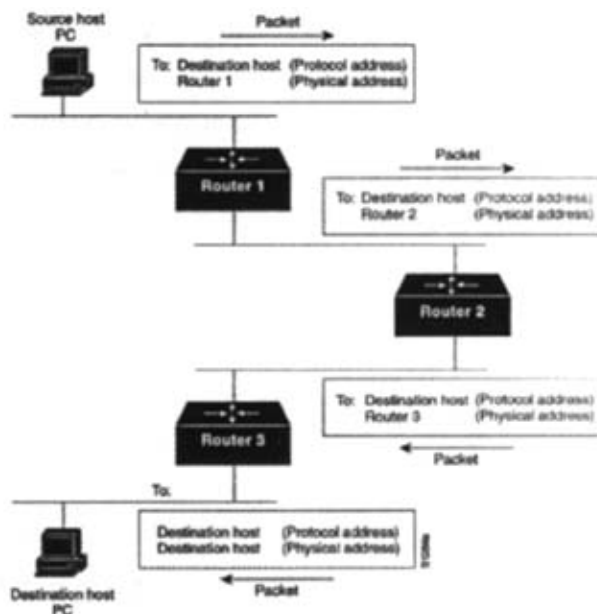


图3.7 路由器间数据包的交换

路由算法主要由几个关键因素决定。首先，算法的设计意图对路由协议的实际运作具有很大的影响。其次，目前存在许多不同类型的路由算法，每一种算法对网络和路由器资源都有不同的要求和影响。最后，路由算法使用不同的度量标准，从而使最优路径的计算结果不同。通常一种路由算法应该体现出最优性、简单，低开销、健壮，稳定、快速聚敛、适用性强等方面的特性。

路由算法使用很多不同的度量标准来确定最佳路径。一些复杂的路由算法更是可以把几个独立的度量标准综合起来，得出更为全面和准确的路由决定。常用的路由度量标准包括：路径长度、可靠性、时延、带宽、负载、通讯成本等等。

路径长度是最为常用的一种路由度量标准。有些路由协议可以允许网络管理人员为每一条网络连接指定路由成本。在这种情况下，路径长度就是所有有关连接的路由成本的总和。其它一些路由协议还可以定义跳数，即数据包从源地址到目的地所必须经过的网络互连设备（如路由器等）的个数。

在路由算法的范畴内，可靠性主要是指每一条网络连接的可用性（通常使用误码率表示）。一些网络连接可能比其它连接更容易出现问题。在网络故障修复之后，有些网络连接可能比其它连接的恢复速度更快或更方便。网络管理人员可以把任何可靠性因素考虑在内，并据此为每一条网络连接指定相应的可靠值。

路由时延是指通过网络把数据包从源地址移动到目的地所需要的时间总和。有许多因素可以造成路由时延，其中包括网络连接的带宽，每一台途经路由器的负载，网络拥挤状况以及数据包所需要经过的物理距离等。因为路由时延是多项重要变量的综合反映，所以被普遍的采用。

带宽是指一条网络连接所能提供的流量吞吐能力。很明显，10-Mbps 以太网的带宽要大大高出 64-kbps 专线的带宽。虽然带宽反映了一条网络连接所能够提供的最大速率，但是有时使用宽带连接的路由并不一定是最优路径。例如，如果一条高速连接非常繁忙，那么实际等待发送数据包的时间可能会更长。

负载是指像路由器这样的网络资源和设备的繁忙程度。我们可以通过多种方式计算负载，例如 CPU 的使用率以及每秒钟可以处理的数据包的数目等。对路由负载进行长期的持续监控可以更加有效的管理和配置网络资源。

通讯成本是另外一种非常重要的路由度量标准，尤其是对那些相对于网络性能更加关注运行成本的企业来说，其重要性就更加明显。举例来说，有时企业会为了节省公用线路的使用成本而改用延迟更大的私人线路，这就是通讯成本的具体体现。

本设计从本质上可看作是一种路由服务，这种路由服务借鉴了上述的传统路由服务的基本技术，本设计也将制定一个类似路由表的路径信息数据库，各项路由度量标准都可以作为数据传输路径的属性，现有的用来确定最佳路径的

路由算法也值得借鉴，但是不同于传统概念上的以单一路径进行的自治路由，本设计提出的模式让网络节点从数据源节点开始发送数据时就明确将要传输数据的路径，实时地知道传输数据的下一跳节点。

3.3 网络管理与系统集成

随着连接 Internet 的网络节点不断增多，网络系统中被控对象的数目也日益增多^[39]。如何对高速、大容量、被控对象数量呈指数级增长的网络系统进行综合集成已经成为目前互联网络技术的热点。基于网络系统集成的网络管理技术从 20 世纪 80 年代中期就成为研究热点，经历了从集中到分布、从平台/应用模式到基于 web 的管理等发展历程，围绕 ISO 定义的五大功能(故障、配置、计费、性能和安全)管理，人们展开多方面研究，并由此诞生许多新的软件技术和系统技术^[40]。即通过将多种测量、控制、管理功能集中在路由器等网络互连设备上，依靠增加网络节点自身的功能来改善网络系统的状态，但是在目前的高速网络环境下，通信能力每三至六个月翻一番，这一新摩尔定律远远高于计算能力增长的旧摩尔定律。如果加强了管控功能的网络互连设备仅仅是各自为政，缺乏有效的集成机制，就将难以再承受多种负载。

因此，一些企业开始基于低层的网络管理变量、集成管理信息库 MIB (Management Information Base) 变量，开发新的网络系统集成技术。例如：HP 公司的 OpenView、SUN 公司的 SUN Solstice Enterprise Manager 和 IBM 公司的 NetView 网管系统等，通过提出基于开放系统互联参考模型(OSI)的系统管理模型和公共管理信息协议(Common Management Information Protocol)^[41]或者基于 TCP/IP 模型的简单网络管理协议(Simple Network Management Protocol)来解决 Internet 发展带来的各种问题。

网络管理就是指监督、组织和控制网络通信服务以及信息处理所必需的各种活动的总称。其目标是确保计算机网络的持续正常运行，并在计算机网络运行出现异常时能及时响应和排除故障。网络管理的目的很明确，就是使网络中的资源得到更加有效的利用。它应维护网络的正常运行，当网络出现故障时能及时报告和处理，并协调、保持网络系统的高效运行等。

网络管理技术是伴随着计算机、网络和通信技术的发展而发展的，二者相辅相成。从网络管理范畴来分类，可分为对网络中“路”的管理。即针对交换

机、路由器等主干网络进行管理；对接入设备的管理，即对内部 PC、服务器、交换机等进行管理；对行为的管理，即针对用户的使用进行管理；对资产的管理，即统计 IT 软硬件的信息等。根据网管软件的发展历史，可以将网管软件划分为三代：

第一代网管软件就是最常用的命令行方式，并结合一些简单的网络监测工具，它不仅要求使用者精通网络的原理及概念，还要求使用者了解不同厂商的不同网络设备的配置方法。

第二代网管软件有着良好的图形化界面。用户无须过多了解设备的配置方法，就能图形化地对多台设备同时进行配置和监控。大大提高了工作效率，但仍然存在由于人为因素造成的设备功能使用不全面或不正确的问题数增大，容易引发误操作。

第三代网管软件相对来说比较智能，是真正将网络和管理进行有机结合的软件系统，具有“自动配置”和“自动调整”功能。对网管人员来说，只要把用户情况、设备情况以及用户与网络资源之间的分配关系输入网管系统，系统就能自动地建立图形化的人员与网络的配置关系，并自动鉴别用户身份，分配用户所需的资源(如电子邮件、Web、文档服务等)。

根据国际标准化组织定义网络管理有五大功能：故障管理、配置管理、性能管理、安全管理、计费管理。对网络管理软件产品功能的不同，又可细分为五类，即网络故障管理软件，网络配置管理软件，网络性能管理软件，网络服务/安全管理软件，网络计费管理软件：

► 网络故障管理(fault management)

在大型计算机网络中，当发生失效故障时，往往不能轻易、具体地确定故障所在的准确位置，而需要相关技术上的支持。因此需要故障管理系统，科学地管理网络发生的所有故障，并记录每个故障的产生及相关信息，最后确定并改正那些故障，保证网络能提供连续可靠的服务。网络故障管理包括故障检测、隔离和纠正三方面的典型功能，例如：维护并检查错误日志、接受错误检测报告并做出响应、跟踪、辨认错误、执行诊断测试、纠正错误。对网络故障的检测依据对网络组成部件状态的监测。不严重的简单故障通常被记录在 错误日志中，并不作特别处理；而严重一些的故障则需要通知网络管理器，即所谓的“警报”。一般网络管理器应根据有关信息对警报进行处理，排除故障。当故障比较复杂时，网络管理器应能执行一些诊断测试来辨别故障原因。

► 网络配置管理(Configuration management)

它初始化网络、并配置网络,以使其提供网络服务。配置管理是一组对辨别、定义、控制和监视组成一个通信网络的对象所必要的相关功能,目的是为了 实现某个特定功能或使网络性能达到最优。包括监视和控制网络设备的配置; 识别被管网络的拓扑结构,标识网络中的各个对象;自动修改指定设备的配置; 动态维护网络配置数据库等;

► 网络性能管理(performance management)

性能管理估价系统资源的运行状况及通信效率等系统性能。其能力包括监视和分析被管网络及其所提供服务的性能机制。性能分析的结果可能会触发某个诊断测试过程或重新配置网络以维持网络的性能。性能管理收集分析有关被管网络当前状况的数据信息,并维持和分析性能日志。一些典型的功能包括:收集统计信息、维护并检查系统状态日志、确定自然和人工状况下系统的性能、改变系统操作模式以进行系统性能管理的操作等;

► 网络计费管理(accounting management)

当计算机网络系统中的信息资源是有偿使用的情况下,需要能够记录和统计哪些用户利用哪条通信线路传输了多少信息,以及做的是什工作等。在非商业化的网络上,仍然需要统计各条线路工作的忙、闲状态和不同资源的利用情况,以供决策参考。计费管理记录网络资源的使用,控制和监测网络操作的费用和代价。估算出用户使用网络资源可能需要的费用和代价,以及已经使用的资源。网络管理员还可规定用户可使用的最大费用,从而控制用户过多占用和使用网络资源。这也从另一方面提高了网络的效率。

► 网络安全管理(security management)

计算机网络系统的特点决定了网络本身安全的固有脆弱性,因此要确保网络资源不被非法使用,确保网络管理系统本身不被未经授权的访问,以及网络管理信息的机密性和完整性。

网络安全管理应包括对授权机制、访问控制、加密和加密关键字的管理,另外还要维护和检查安全日志。包括:创建、删除、控制安全服务和机制、与安全相关信息的分布、与安全相关事件的报告等;

网络管理是网络系统集成中应用基础平台的要素之一。

网络系统集成是指:根据应用的需要,将硬件设备、网络基础设施、网络设备、网络系统软件、网络基础服务系统、应用软件等组织成为一体,使之成

为能够满足设计目标、具有优良性能价格比的计算机网络系统的全过程。计算机网络系统集成有 3 个主要层面：技术集成、软硬件产品集成和应用集成（如图 3.8）。在基于 Internet 的网络控制系统中，相关的互联技术，网络通信协议，现有的设备设施以及数据传输模式等均为网络系统集成的元素。

系统集成绝不是指各种硬件和软件的堆积，系统集成是一种在系统整合、系统再生产过程中为满足客户需求的增值服务业务，是一种价值再创造过程。^[33]

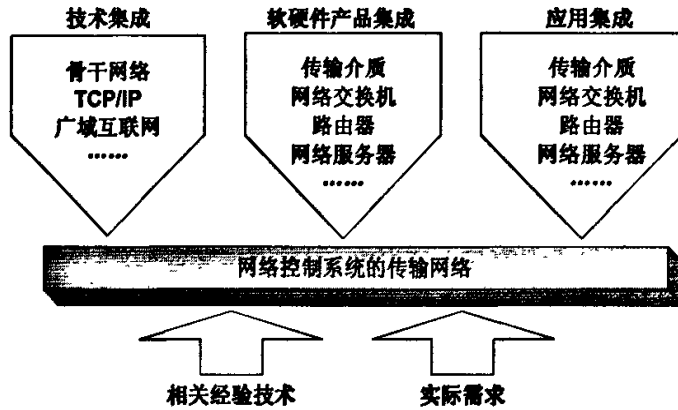


图 3.8 NCS 的传输网络系统集成

网络系统集成的体系架构从多个技术层面体现了综合管理控制 NCS 传输网络的思想，具体分类如图 3.9 所示。

环境支持平台是指为了保障网络安全、可靠、正常运行所必须采取的环境保障措施；

计算机网络平台包括网络传输基础设施，即以网络连通为目的铺设的信息通道；

网络通信设备指通过网络基础设施连接网络节点的各类设备；

网络服务器硬件和操作系统，服务器是组织网络共享核心资源的宿主设备，网络操作系统则是网络资源的管理者和调度员。二者又是构成网络基础应用平台的基础；

网络协议是协调不同网络设备间信息交换的规则，以网络层次划分，约定数据传输的速率、顺序、数据格式及差错控制等方面；

外部信息基础设施的互联和互通，突破物理联通和应用相容的建设问题；

应用基础平台包括数据库平台；

Internet 基础服务，即指建立在 TCP/IP 协议基础和 Internet/Intranet 体系基础之上，以信息沟通、信息发布、数据交换、信息服务为目的的一组服务程序；

网络管理平台，大多支持 SNMP 协议，是建立在 HP OpenView 网管平台基础上的网络管理产品；

而开发工具指为建造具体网络应用系统所采用的软件通用开发工具，包括数据库开发工具、Web 平台开发应用工具和 Visual C++ 等标准开发工具；

网络应用系统指以网络基础应用平台为基础，为满足建网单位要求，由系统集成商为建网单位开发，或由建网单位自行开发的通用或专用系统；

用户界面包括 C/S 平台界面、Web 平台界面、图形用户界面（GUI）

网络安全平台贯穿了网络系统集成体系架构的各个层次。网络安全的主要内容是防信息泄漏和防黑客入侵。在应用层通过用户身份认证来授予他对资源的访问权；在网络层，使用防火墙技术，分割内外网，使用包过滤技术，跟踪和隔离不良侵入；在数据链路层，使用信道或数据加密传输技术来传送主要信息；在物理层，实施内外网物理隔离。

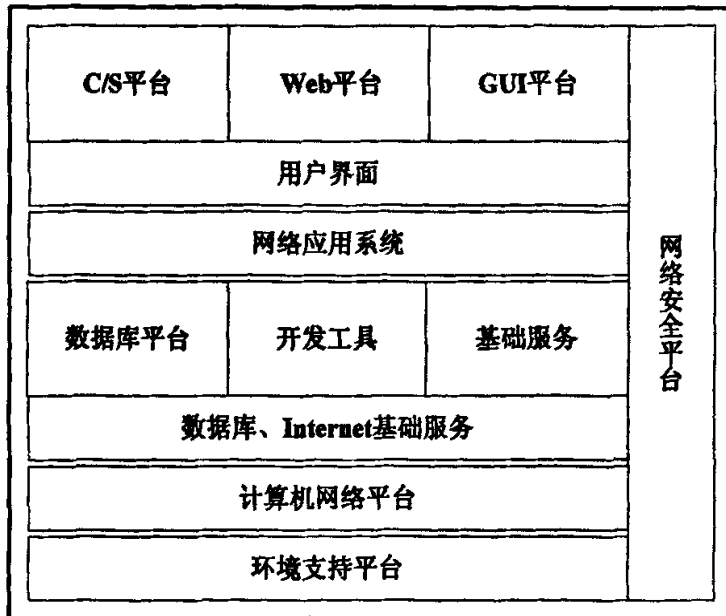


图 3.9 网络系统集成体系框架

本文所提出的控制网络多路径传输模式正是在网络系统集成思想的指导

下，受到了网络管理技术的启发进行的设计。与 NetView 等网络管理系统相似，本设计也是一项工作在应用层的服务，但此项应用层服务并非如网络管理系统那样机械地以 get、set 等命令提交和设置管理参数，而是可以根据网络情况主动实时地更新路径信息记录，体现系统的自适应特点。

3.4 本章小结

本章内容介绍了与本设计相关的多项技术知识。其中多中心化的对等网络模式是当今互联网发展的全新应用模式，对等网络模式的非中心化服务器思想和节点功能自治的特性为第 4 章中多路径传播模式的路径管理器提供了设计依据；而网络传输原理及互联技术中 OSI 七层参考模型各层功能进一步说明了本课题所设计的应用层路由管理进程的意义、网络通信和路由技术等在第 4 章的软件模拟原型设计中将得到实际应用；网络系统集成理论是本课题设计的主要思想依据。

第4章 控制网络多路径数据传输模式的软件模拟原型设计

4.1 多路径数据传输模式的提出

4.1.1 与现有 NCS 技术不同的方案

基于 Internet 的网络控制系统存在着网络传输延时、数据包的丢失和排列顺序混乱等问题,影响了网络控制系统的实时性,可靠性,甚至稳定性。同时,互联网规模的迅速扩展对 IP 地址模式的威胁并不是它不能保证主机地址的唯一性,而是会带来两方面的负担:第一,巨大的网络地址管理开销;第二,网关寻径表的急剧膨胀。其中第二点尤为突出,寻径表的膨胀不仅会降低网关寻径效率(甚至可能使寻径表溢出,从而造成寻径故障),更重要的是将增加内外部路径刷新时的开销,从而加重网络负担。因此,迫切需要寻求新的技术,研究新的网络控制理论与方法,以解决互联技术和网络规模增长所带来的问题。

现有的 NCS 技术包括:

增广状态法——通过将经过延迟的输入输出变量作为增广状态并建立新的状态空间方程,与原始状态方程联立共同来描述系统。例如 Halevi 和 Ray 针对具有周期性时延特性的网络提出了增广状态确定性离散时间网络控制系统模型。

状态估计方法——通常是利用对象经过延迟的部分状态信息来获取当前甚至将来的完全状态,再利用获得的状态实施各种控制算法。在获取了当前或者将来的完全状态后,据此计算当前的控制量甚至将来的控制量,最终实现了控制系统的时延补偿。例如 Luck 和 Ray 提出了基于确定性预报器的时延补偿方法^[32],并将其应用于有随机时延特性的网络控制系统。它的主要思想是利用过去的测量信号,采用观测器重构系统过去的状态,再利用预报器提前预报系统状态,进而根据得到的预报状态生成控制信号,以此补偿时延造成的影响。

还有“最优控制方法”:用一个线性随机系统模型来描述带随机时延特性的网络控制系统,并将网络控制系统中通信时延的影响转化为一个 LQG 问题^[26];“摄动方法”:将网络控制系统中的延迟效应看作连续系统的摄动^[27];“增益自适应法”^[28]:提出了增益适应的方法来补偿网络服务质量变动和恶化造成的影响

[28][29][30]；另外还有鲁棒控制方法、采用时间调度法等等。

目前对网络控制系统的研究大多分为两个方向。一类是对网络延迟模型的研究，如假设延迟为常量、延迟为符合某种分布的随机数、延迟有界且延迟向量符合某种随机过程等等；另一类是运用现代控制理论，在网络控制系统的控制器中改进参数，增加补偿环节。这些已有的技术一般仅限于理论或仿真研究，应用于实际比较困难。

而本文的设计思想与上述 NCS 技术不同。考虑到现有的网络体系纷繁复杂，各种网络通信技术又层出不穷，本文立足于控制网络的现有技术，同时充分考虑了 Internet 技术本身在不断发展的实际情况，提出了一种利用现有的各项网络设施、协议等资源，解决基于 Internet 的网络控制系统存在的传输延时和数据包丢失、排列顺序混乱等问题，以提升系统快速性和可靠性的设计方案。

目前的 Internet 自身发展进入了网络节点对等互联模式的阶段，相应的网络节点具有功能自治和拓扑结构多中心化的特点。NCS 的规模不断发展，使得其传输网络不再是单纯能够严格控制信息传输延时的现场总线网络，以 Internet 为代表的公共数据通路也成为了 NCS 的传输网络。在基于 TCP/IP 协议通信的 Internet 中，路由处理、ADU（用户数据单元）在网络上传输以及服务器对用户请求的处理，是造成传输网络延迟的主要原因；同时，如第三章中所分析，现有的网络路由技术会产生数据信息的丢包、排列顺序混乱等传输问题。

在现有的技术背景下，本课题不着力于改进控制器单元的算法，而是研究合理利用现存的硬件设备、网络基础设施设备、网络系统软件、网络基础服务系统、应用软件等，创新性地提出一种创新的数据传输模式，使 Internet 成为能够满足设计目标、具有优良性能价格比的 NCS 传输网络系统。

4.1.2 基于网络系统集成的研究目标

从网络系统集成的角度，对基于 Internet 的 NCS 传输网络部分的控制，除了需要考虑技术因素外，还需要考虑管理因素。从系统工程的角度，本节提出课题系统集成的初步体系架构。

网络互联部件是网络互联的关键，它既可以是专门的设备，也可以利用各子网原有的节点。网络互联部件在内部执行各子网的协议，成为子网一部分；同时实现不同子网协议之间的转换，保证执行两种不同协议的网络之间可以进

行互联通信。协议转换包括协议数据格式的转换、地址映射、速率匹配和网间流量控制等。

进行网络互联的中间设备，按照 ISO 的术语，也称为中继 (Relay) 系统。常用的网络互联部件有：转发器（负责物理层的数据中继）、网桥（负责数据链路层的数据中继）、路由器（负责网络层的数据中继）、网关（负责网络层以上的数据中继）。中继系统协议转换的过程可以发生在任何层次，如网桥的协议转换发生在第二层（数据链路层）以及下层，路由器的协议转换发生在第三层（网络层）及其以下各层。

多路径传输模式的设计原理类似专用集成电路 (ASIC) 中的 FPGA (现场可编程门阵列) 工作原理。用户可对 FPGA 内部的逻辑模块和 I/O 模块重新配置，以实现用户的逻辑；也可在线对其编程，实现系统在线重构。但用户并不需要重新设计 FPGA 的逻辑模块和 I/O 模块，也不需要关注这些模块的底层电路和内部结构，只需要将这些模块合理地利用以实现用户的需求。对系统的评价是基于对各模块整体组合的特性，而与模块内部特性无关。同样的，多路径传输模式并不关注如何改进网络中个别的硬件设备、基础设施或者应用软件的特性，而是将这些现有的资源都作为“模块”来进行合理利用，最终实现提升整个系统性能的目的。

互联网规模的增长主要表现为网络节点的增加。现有的互联网配置中，路由器都应用了某种自适应路由技术。采用自适应路由，则随着互联网络中条件的改变，用于转发数据报的路由也会改变。当网络或路由器传输信息失败时，就不再用作路由的一部分。当网络中特定的部分严重阻塞时，自适应路由就让数据报绕开拥塞区域，而不是通过拥塞区域。但使用自适应路由，也存在一些缺点：

- ▶ 路由决策更为复杂；因此，对路由器的处理的速度要求更高；
- ▶ 在大多数情况下，用于某路由节点的自适应策略取决于别处收集的状态信息。在信息质量和开销量之间存在着一定的折衷。交换的信息越多越频繁，各节点才能更好地做出路由决策。可以想象，此类信息的频繁交换也会造成网络的负载，从而会导致网络性能下降；
- ▶ 自适应策略的控制效果不同：可能反应过快，由此导致拥塞生成 (congestion-producing) 摆动，也可能反应过慢，以至于路由滞后；
- ▶ 可能会带来颤动 (fluttering) 和循环 (looping) 等问题；

颤动指由于路由器试图分割或平衡负载而形成的路由快速摆动。路由器在下一跳选择过程的最后，仍可能保留多个路由。路由器可以保持多条路由，并采用一种负载分割机制，将业务流划分至这些路由中，这种负载分割可能导致异常行为：如果只在一个方向上出现颤动，那么两个方向上的路由特征（包括定时和差错特征）就会有所区别。这样就会使一些力图度量路由特征的管理和排错应用造成混淆：如果源和目标之间采用两条不同的路由，往返时间和可用容量的估计就会变得很困难；若两条路由的传播时间不同，TCP 数据包会以不正确的顺序到达，这样将生成重复确认而造成欺骗性的快速重传。

而循环是一种更严重的路由问题，即一个路由器所转发的数据包最终仍回到该路由器。当互联网络连通性有所变化，而且这种变化并未即时传播至所有其他路由器时，就有可能出现循环。

以上所述的路由问题导致的信息传输的延迟、抖动等虽然都是单纯的网络性能参数，但同时也会影响控制算法、控制策略的分析和设计。网络控制系统的可控性、稳定性以及性能的提高都与控制回路的服务质量密切相关。如图 4.1 所示，“北京”的一个路由器将要发至目标“上海”的负载划分至路径 1（实线：3 跳）和路径 2（虚线：6 跳）。其结果是同样指向“上海”的连续数据包却会通过完全不同的路由到达。

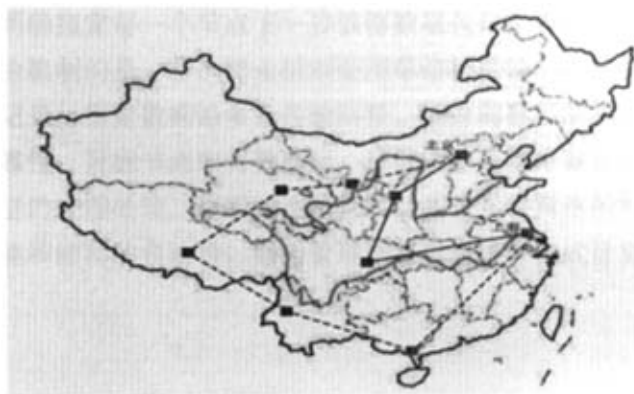


图 4.1 数据包从“北京”传到“上海”的路由示例

因此解决问题的思路集中在：如何利用现有的互联网络的对等通信模式，减少网络数据传输时延迟，减少数据包的丢包和排列顺序混乱问题，提高数据包的传输成功率，提升网络控制系统的实时性和可靠性。于是提出控制网络多路径传输模式。

4.1.3 多路径传输模式的设计概要

多路径数据传输模式创新性地设计了一种新的网络数据传输协议。不采用自适应路由方式，而是通过路由管理器指定网络中数据传输的路径，并且可按指定的多条路径传输同一组数据。

多路径数据传输模式体系结构分为被管理的网络设备(Managed Device)、路由管理器(Router Manager)、数据传输协议代理(Data Transform Protocol Agent)和传输路径信息库(Route Database)四个部分。其中被管理的设备是网络中的一个节点，有时被称为网络单元(Network Elements)，它可以是路由器、网管服务器、交换机、网桥、集线器等。每个支持此协议的网络设备中都运行着一个协议代理，它负责随时收集路由管理器指定的传输路径信息和反馈即时的路由信息，然后通过应用层的程序进程查询或修改代理所记录的信息，并将数据传输路径信息集中存储在路径信息库中。

整个多路径数据传输模式结构体系的工作过程如下。当一台主机需要发送数据时，先主动向路由管理器发送信息，告知要将数据传输给哪个目标节点，路由管理器得到此信息后，立即根据路径信息库中的记录，指定数据的传输路径，并同时向此路径中从数据源到传输目标节点所需经过的每一个网络节点发送路由指令，明确指定每一个节点下一步该将数据包传至哪一个节点。并且不同于现有的路由机制的是，新的路由机制采用单路径传输一组数据，可以根据路径信息库的记录，指定数据沿多条传输路径，同时向目标节点发送，保证了数据传输的可靠性。目标节点取得数据后，可以向路径管理器反馈路由信息，通过网络应用层的进程处理，更新路由信息库。基于互联网络的对等结构，网络节点具有功能和结构的自治性，路由管理器可以通过分布式计算的方式由多节点协作完成。

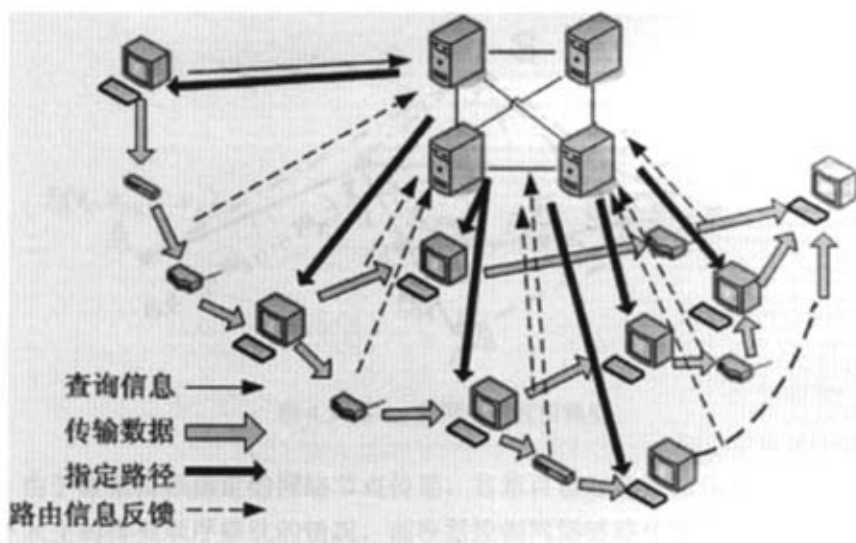


图 4.2 多路径传输模式简单示意图

多路径数据传输模式的创新思想主要体现在让网络中的数据传输不再依照传统的单路径自适应路由方式进行，而是变成按指定的多条路径发送；并且传输数据的网络节点以主动查询路径信息的方式来发送数据，区别于现有的 BT 等 P2P 模式中数据节点被动等待收发的形式。

从网络控制系统的网络传输服务质量角度考虑，以多路径传输模式进行数据传输，可以大大改善传输延迟、丢包率高、网络吞吐量大等问题。按照图 4.2 的结构得到如图 4.3 所示的网络对称有权图模型，其中描述各网络节点的有序数列 P 和描述各路径的有序数列 e 都表示了它们的属性或权，这些属性即可反馈给路由管理器，作为更新路由信息库的有效参数。此类参数的获得方式可参照宽带网络测试系统等工具的技术原理，例如系统自带的 PING 命令可以确定本地主机是否能与另一台主机(服务器或各地机器)成功地交换数据包，再根据返回的信息可以推断 TCP/IP 参数是否设置、网络是否通畅等来简单评价网络的 QoS。假如在命令提示符中输入“ping 网址”后回车，就会有类似这样的结果：

Ping statistics for 220.181.29.154:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 9ms, Maximum = 15ms, Average = 13ms

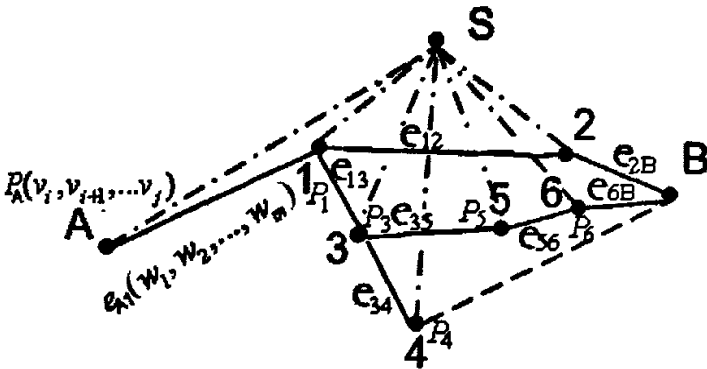


图 4.3 图 4.2 的网络有权图模型

由于数据按照指定的网络节点传递，自然可以避免数据包在自适应路由情况下发生的排列顺序错乱的情况，而根据控制网络链路传输特性及时更新的路径信息库保证了数据能够以实时最优的路径传输，加上可以设置数据源节点沿多条传输路径共同向目标节点发送一项数据，杜绝了数据老化和丢包现象。

4.2 多路径数据传输模式的描述

本模式在设计过程中主要是针对对等网络的多中心化模型、带有发现和查找节点服务器模型和利用应用层实现路由服务进程的特点，在综合它们的优点的同时加以改进，组合而成一种综合模型。系统分为三个模块——路由管理器、网络传输节点、路由信息数据库接下来分别讨论每个模块的功能以及它们之间的关系。

4.2.1 系统的体系结构模型

网络中数据的传输是以发送和接收数据包的形式传输的，理想状态下是发送了多少数据包就能接收到多少数据包，但是由于信号衰减、网络质量等等诸多因素的影响，将造成网络时间延迟和丢包等情况，对网络控制系统不利。

多路径数据传输模式的提出是基于网络系统集成的思想，应用了当前的几种成熟的网络通信技术，又借鉴了互联网发展中不断创新的先进技术思想。综合考虑用户实际使用情况，致力于解决控制网络的实时性、信息交换的准确性

等方面的问题。

本模式是设置在不同网络或网络通信域之间的一系列部件的组合。计算机网络平台包括网络传输基础设施、网络通信设备、网络服务器硬件和操作系统、网络协议 (TCP/IP 等)、外部信息基础设施的互联和互通。

应用基础平台包括：路径数据库平台、Internet 基础服务、网络管理平台、JAVA 语言开发工具 JBuilder 9.0；而网络应用系统，用户界面，网络安全平台则皆应用现有网络部分。

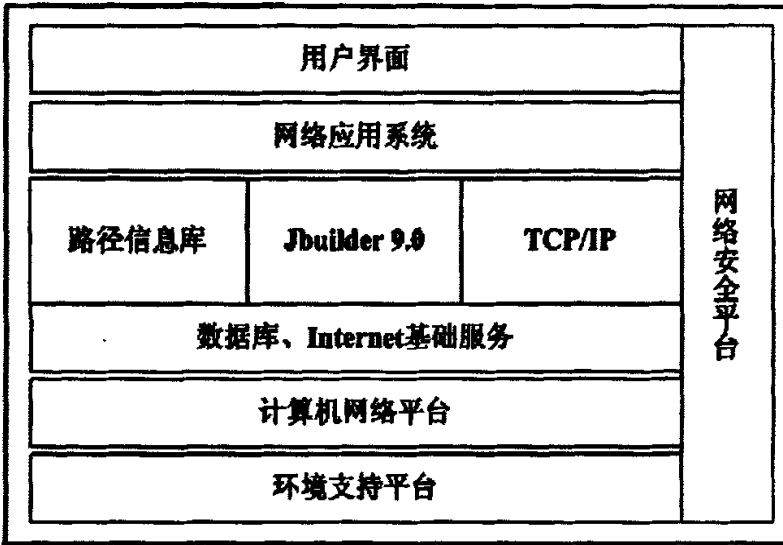


图 4.4 多路径传输模式体系框架

多路径数据传输模式是建在基于 TCP/IP 协议的 Internet 应用层上的一个路由管理系统架构，比类似架构的多采用 UDP 协议的 SNMP 网络管理协议更加可靠。主要通过一组 Internet 协议及其所依附资源提供网络数据传输服务，它提供了一个基本框架用来实现对路由管理、信息库管理、访问控制等的高层管理。以上几种功能部分组合在一起就构成了对等式控制网络多路径传输模式的整个体系结构。每一部分将在下文详细描述。

4.2.2 路由管理器

路由管理器主要是提供路径服务，引导数据沿多条指定路径从数据源节点传送到目标节点，并及时获取各路径的反馈信息，维护路径信息库。路由管理

器的主要作用是发送路径指示，目的是为了克服基于 Internet 的 NCS 出现的网络传输延迟和由于现有的互联网络自治路由机制造成的数据包丢失和排列顺序错误等问题。路由管理器的设计主要体现了对等网络节点路由机制中目录式路由机制的思想。

在基于 Internet 的网络控制系统中，路径管理器本身也是互联网中的节点。其具体的工作方式是，当某节点需要发送数据时，就自动连接到路径管理器上，提交自己将要发送数据的目标节点信息，所以路径管理器，就保存了此数据源节点的一个索引。同时路径管理器将查询后台的路径信息库，选出记录中最新的且 QoS 指标好的传输路径（可以为多条），并向此路径所经过的所有网络节点发送路径指示，明确这些节点下一步应该将数据包传至哪个节点，直到目标节点收到数据包后，采用相应的机制，拒绝其他路径传来的同样的数据。

如同对等网络文件共享系统 Napster 提供服务允许音乐迷们交流 MP3 文件那样，虽然路径管理器向从数据源到目标节点之间的路径上的数据传输节点提供路径指示，但它没有保存任何传输的数据。这种路由机制虽然具有客户机/服务器模式的中心化结构特点，但是它与传统意义上的 C/S 模式不同。所有的数据资源都存放在传输该数据的用户机上，路径服务节点只保留索引信息，相应的对等节点之间具有交互能力，体现了均衡对等网络负载的优点。而且考虑到互联网络的分布式多中心化结构发展趋势，可以有成百上千这样的路由管理器超级节点，由单独的用户来维护。

路由管理器的创新之处还体现在“应用级路由”的设计思想，与微软公司使用 Windows 操作系统自带的程序 alg.exe 实现应用级网关 (Application Layer Gateway Service) 的原理类似。传统的路由器体系结构中，硬件部分就是一个通用计算机系统，有关路由转发和协议的管理部分主要由路由器内核的软件体系结构体现。根据 ISO 提出 OSI 七层参考模型，互联网络通信中常用的网络互联部件如转发器、网桥、路由器等都分别在物理层、数据链路层、网络层等工作。本设计中的路由管理器是工作在应用层之上的。

在逻辑上，应用层数据传输路由模式就是一个位于计算机和它所连接的网络之间的软件。它可通过监测、控制跨越各个网络节点的数据流，在网络应用层上对网络内部的信息、结构和运行状况进行分析、登记和统计。提供上层的路由管理，有效地实现了通过多条传输路径有序传递数据包的功能。

4.2.3 数据传输节点

数据传输节点就是基于数据传输协议代理的普通网络用户节点，主要性能是发送路径查询请求和接收路径指示，输入/输出需要传输的数据。

在传统的传输模式下，还存在着“数据老化”的问题^[42]。网络通信机制中的“生存时间”(TTL, Time To Live)是IP协议包中的一个值，它告诉网络路由器数据包在网络中的时间是否太长而应被丢弃，如图4.5所示。现实中，有很多原因使数据包在一定时间内不能被传递到目的地。例如，不正确的路由表将可能导致数据包发送过程的无限循环。解决此问题的一个方法就是在一段时间后丢弃这个包，然后给发送者一个报文，由发送者决定是否要重发。TTL的初值通常是系统缺省值，是包头中的8位域值。TTL的最初设想是确定一个时间范围，超过此时间就把包丢弃。由于每个路由器都至少要把TTL域减1，TTL通常表示包在被丢弃前最多能经过的路由器个数。当记数到0或小于下一网段许可通过的TTL值时，路由器决定丢弃该包，并发送一个ICMP报文给最初的发送者。

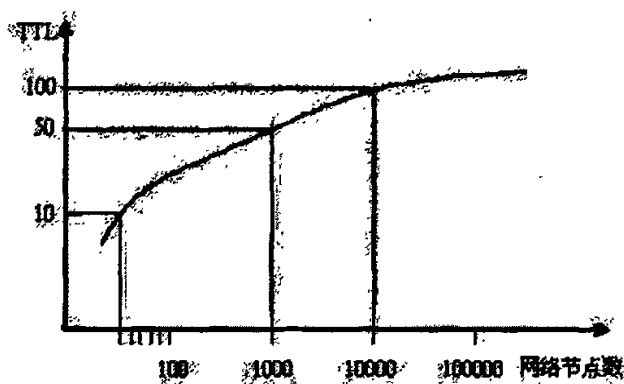


图 4.5 TTL 值与网络节点数的关系^[42]

采用多路径传输模式，则不但可以减少因自适应路由而造成的“迷宫式”数据传输形式，即网络中某节点在得到数据前都未确定下一跳的目标节点，而且可以解决网络层的 TTL 问题。现今互连网上最流行的 P2P 软件 BitTorrent(简称 BT) 先将一个扩展名为 .torrent 的文件放在一个普通的 WEB 服务器上，Tracker 是主要负责帮助下载者互相寻找对方的服务器，在占用文件发布者极少的带宽资源的情况下快速发布文件；本文所设计的模式下，数据传输节点发送

路径查询信息,也只占用极少的带宽和机器资源。更为重要的是,BT的用户节点若是只有远程连接,没有本地连接,就不能成功的主动发起连接,只能被动的等别的客户端来连接。与BT在每个客户端在启动以后监听某个端口被动等待其它节点的连接请求的情况不同,数据传输节点是收发行为的发起者,强调了网络控制系统的实时性,这种PUSH方式所体现的主动性将现行的平面网络上升到层次网络的概念,实现了整个NCS中“控制”的理念无处不在。

4.2.4 路径信息库

路径信息库的服务目标就是帮助网络节点正确迅速地定位数据的传输路径。

依据网络图论的相关知识,可以定义网络的有权图模型,定义非空有限集合 $V(G)$ 和其中某些元素的有序对集合 $E(G)$ 构成的二元组,记为 $G=(V(G),E(G))$ 。其中, $V(G)$ 是图 G 的节点集,元素 $v \in V$ 表示网络中的节点; $E(G)$ 是图 G 的边集,元素 $e_{ij} \in E$ 为 V 中元素的有序对,表示图 G 的一条从节点 v_i 到节点 v_j 的边。网络有权图中的 E 即为连接节点集合 V 中两个节点的链路,且 $\forall e_i \in E$ 都有相应的属性集 (w_1, w_2, \dots, w_m) ,这些属性包括可以评价网络传输服务质量QoS的各项属性,包括网络带宽、延迟、抖动、吞吐量、队列长度、花费等参数。

从网络系统集成的角度分析,对NCS的传输网络部分的控制,除了技术因素外,还有管理因素。任何现有的网络资源,包括硬件设备、网络基础设施、系统软件、网络基础服务系统、应用协议等,只要能满足设计目标,就可以被整合利用。路径信息库的建设可以参照目前各类技术成熟的搜索引擎的索引库排序原理。从第一代的“词频和位置统计排序”到第二代的“链接分析排序”,发展到第三代的“智能化排序”(即考虑网页的质量,并且惩罚在链接量上作弊的网站),都是从不同的角度选择网页的相关参数作为排序权值的。专业搜索引擎的排序算法是基于概率模型,贝叶斯模型、随机漫游模型或者Hub和Authority相互加强模型等多种数学模型的设计,例如Google的PageRank技术和百度的超链分析技术等^[43]。

路径信息库的建设即可以采用适用于结构化数据的加权排序算法技术,以网络有权图中的数据传输链路的状态属性作为权值,实现智能化的维护技术,

及时为路由管理器节点提供数据传输路径记录。同时，由于多路径传输模式是在网络的应用层上设置的面向用户的路由管理进程，所以虽然网络通信是基于TCP/IP的传输控制协议，但用户在设置数据发送的目标节点时可以不必要以IP地址来表示网络节点，而每个节点可以拥有实名来表示，路径信息库可以保存节点实名与IP地址间的映射关系。

4.3 多路径数据传输模式的软件模拟原型

4.3.1 模拟原型概述

本文提出的多路径传输模式是为了解决NCS控制网络存在的对控制系统不利的网络延迟、数据包丢失和排列顺序混乱等问题，应用网络系统集成思想提出的一种创新性的数据传输方案。在本文的研究设计过程中，应用JAVA语言编写了一个依照此模式来传输数据的软件模拟原型，以进一步说明该模式的工作过程。

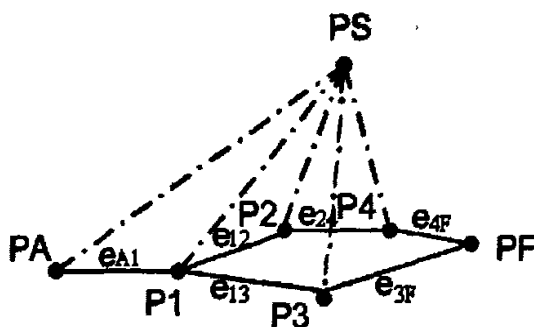


图 4.6 模拟原型的网络有权图

软件模拟原型使用操作系统的七个端口表示网络中的对等节点。当节点 A 要发送数据到节点 F 时，先主动向路由管理器节点 S 发送信息，告知节点 S 要将数据传输给哪个目标节点，路由管理器得到此信息后，立即查询后台的路径信息库中的记录，发现信息库中现存的两条数据传输路径记录为 A-1-2-4-F 和 A-1-3-F，于是向这两条路径中的每一个网络节点发送路由指令，明确指定每一个节点下一步该将数据包传至哪一个节点。例如，节点 S 向节点 1 发送指令——若收到来自节点 A 的数据则将其传至节点 2 和节点 3，节点 2、3、4 也收到相

应指令，节点 F 在从某一路径收到发自节点 A 的数据之后，经过校验无误就拒绝再接收相同的数据。

同时，用 JAVA 语言的 sleep() 函数将节点的通信进程暂停一段时间来模拟网络中的延迟情况，Sleep() 函数的时间参数由随机数发生器 Random() 产生。两条路径的延迟表示即为：

$$D(P_{A,F}) = h\{D(e_{1,2}), D(e_{2,4}), D(e_{4,F})\} \quad (4.1)$$

$$D(P_{1,F}) = h\{D(e_{1,3}), D(e_{3,F})\} \quad (4.2)$$

路径管理器得到各数据传输路径反馈的网络服务质量 QoS 信息，通过应用层的进程处理，维护更新路由信息库。

4.3.2 模拟原型的设计

► 开发环境

1 硬件系统

Pentium (R) 4 CPU 主频 450MHz，内存 512MB

2 软件环境

操作系统：WINDOWS XP

图形平台：X WINDOWS

开发平台：JAVA version 1.4.1_02-b06；J2ME Wireless Toolkit 1.0.4_01

3 编程语言

主要采用 J2SE 版本的 JAVA 语言作为主要程序开发编程语言。

J2SE 是 Java 的标准版本 (Java2 Standard Edition)，定位在客户端，特别在 Internet 和 多媒体 (Multimedia) 相关产品类方面广泛应用。Java 应用程序 (Applications) 能在网路上开启及连结使用物件，就如同透过 URLs 连结使用一个本地文件系统 (Local File System)。一般而言，网络是由很多不同机型的机器所组合而成的，CPU 和作业系统体系结构均有所不同，而 Java 的编译器产生一种结构中立的文件格式 (Object File Format)；这使得编译码得以在很多种处理器中执行。Java 语言具有多线程的功能，这对于交互回应能力及即时执行行为是非常重要的。Java 比 C 或 C++ 语言更具有动态性，更能适应时刻在变的环境，Java 不会因程序库的更新，而必须重新编译程序。综上所述，Java 语言是一种具有简单的 (Simple)、面向对象的 (Object Oriented)、分布

式的(Distributed)、解释的(Interpreted)、健壮的(Robust)、安全的(Secure)、多线程的(Multithreaded)、动态的(Dynamic)的等特性的程序语言。

在 JAVA 开发工具中, JBuilder 一直是最受开发者推崇的开发工具之一。它是 Borland 公司推出的一款优秀的 Java 集成开发工具, 支持 JDK 和 J2EE1. X 平台, 集成了工程管理、单元测试、代码审查、撰写 Javadoc 说明文档等功能。本文中程序编写所用到的正是 JBuilder 9.0 版本。

➤ 总体规划

本软件模拟原型系统的架构结构见图 4.7。本系统建立在工作站小型机系统上, 基于 Windows 操作系统, 在指定的软件开发和运行平台环境之上构建由数据平台和图形平台组成的管理应用平台, 在此基础上实现控制网络的多路径传播模式。

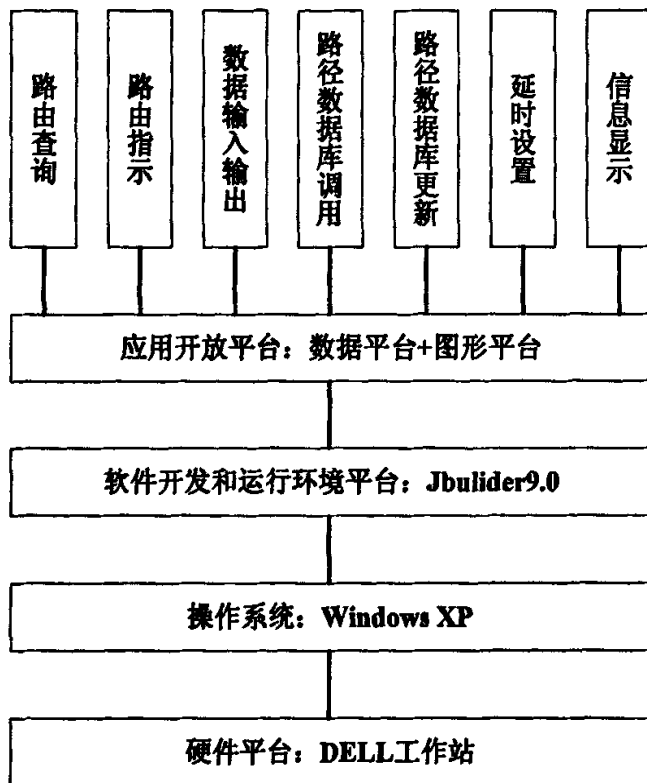


图 4.7 软件系统架构

根据系统架构框图可知：本软件系统由几项模拟功能组成，功能的划分及其之间的关系如图 4.7 所示。包括：路由查询、路由指示、数据输入输出、路

径数据库调用、路径数据库更新、延时设置和信息显示。

► JAVA 设计规范

1 进程和线程

进程和线程是操作系统中非常重要的两个概念。

简单地说，进程是程序和资源的结合。进程中不仅包括所要执行的指令，还包括执行指令所需的系统资源，如数据空间、I/O 端口等。在现代操作系统中，都能够同时执行多个任务，也就是多进程。

而线程是比进程更小的执行单位。线程通常的定义是“一个计算机程序的一个执行路径”。一个进程内可以同时容纳多个线程，一个进程内的多个线程共享进程资源。多线程是指在同一个应用程序中，有多个具体的工作同时进行。例如本系统中包含的七个操作系统端口，它们之间的通信分别作为一个独立的线程，保证了数据传输的连贯。

无论是在哪种平台，一个线程都必然包括创建、可执行、不可执行和结束 4 个基本的生命状态，称为线程的生命周期。

首先要创建一个线程，但创建以后线程并没有直接启动，而是要等待调用某线程的某些方法来启动它。线程结束以后便出于不可运行状态，直到该线程由于某些原因变为不可执行状态或者线程的工作结束。不可执行的原因可能是睡眠、被暂停或者等待某些资源。当线程被唤醒、被恢复或者得到资源时，它就马上回到可运行状态，再次等待被调度执行。

线程的生命周期的四个基本环节如图 4.8 所示。在实际的操作系统中，线程的生命状态远比这要复杂，而线程处于可运行状态和线程实际在 CPU 上运行是不同的。因为线程要实际运行，不但线程要处于可运行状态，而且还要调度器为该线程分配 CPU 的使用权限。

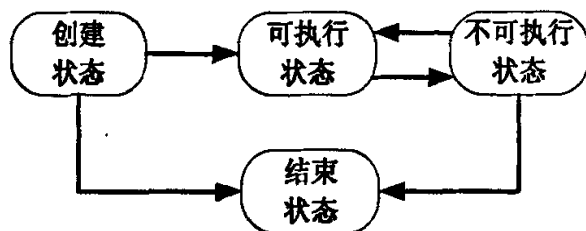


图 4.8 线程的基本生命周期

JAVA 的线程结构由 3 部分组成：虚拟 CPU、代码和数据，如图 4.9 所示。

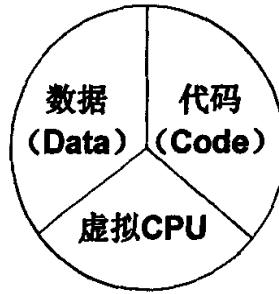


图 4.9 线程的基本结构

虚拟 CPU 封装在 `java.lang.Thread` 类中，无须编程人员处理。因此，当一个线程被构造时，只需要使用构造方法的参数、执行代码和操作数据来进行初始化即可。由于 JAVA 是面向对象的，因此代码只能写在类的方法中，数据只能作为方法中的变量或类的成员变量，所以为线程提供代码和数据的工作都是以类的实例形式完成的。

本次设计中为了实现多个端口（代表网络节点）的相互通信，构造了具有 Socket 通信收发数据功能的线程 `ReSe`：

```
class ReSe extends Thread    //创建一个线程，用以监听端口收发数据
{ ... ..
  new ReSe(s,ip,port).start();           //启动线程
  ... ..
}
```

2 输入输出流

JAVA 中所有基本的输入输出操作都是以流（Stream）为基础的。流是程序内数据流动的路径。大的来说，流可以按流向分为输入流和输出流两种。输入流是指数据流向程序，而输出流是指数据从程序流出。

如果按照流的数据类型来分，又可以分为字节流和字符流。字节的范围是从 0 到 255，但是许多其他类型的数据可以用字节格式来表示，包括十进制数据，可执行程序、Internet 通信和 JAVA 虚拟机运行的类文件等。字符流实际上是一种特殊的字节流，它擅长于处理 Unicode 表示的文本数据，包括文本、网页和其他常见的文本类型。

无论是字节流，还是字符流，其使用的方式都是类似的。

对于输入流，首先创建和数据来源相关的一个输入流类的对象，然后使用

该对象的一个方法从该流中写入信息，当完成信息的读入以后，调用 `close()` 方法结束对流的使用。

对于输出流，相应地，首先创建和数据输出目的相关的一个输出流类的对象，然后使用该对象的一个方法来向该流中写入信息，当完成信息的写入以后调用 `close()` 方法结束对流的使用。

抽象类 `InputStream` 和 `OutputStream` 是所有字节流类的父类，它们分别提供了输入和输出处理的基本接口，并且实现了其中的一些常用方法。

本次设计中端口间的数据收发利用了抽象类 `InputStream` 和 `OutputStream` 以及缓冲流类 `BufferedReader` 所定义的读写方法。以下代码表示将数据流 `pw` 发出，读入的数据放入字符串变量 `get`：

```
private Socket s;
... ..
    OutputStream os=s.getOutputStream();
    PrintWriter pw= new PrintWriter(os);
    InputStream is=s.getInputStream();
    BufferedReader br=new BufferedReader(new InputStreamReader(is));
    String get;
    get=br.readLine();
... ..
```

3 随机网络延时数的产生与实现

在计算机中并没有一个真正的随机数发生器，但是可以做到使产生的数字重复率很低，得到的结果类似于真正的随机数，实现这一功能的程序叫伪随机数发生器^[37]。实际操作中必须给伪随机数发生器提供一个名为“种子”的初始值，而且这个初始值是随机的。“种子”的值通常是用快速计数寄存器或移位寄存器来生成的。根据数学推导^[38]，利用公式 4.3，就可以生成一个伪随机数了。

$$\text{Rand_Number} = (\text{Rand_Seed} * X + Y) \bmod Z \quad (4.3)$$

其中 `Rand_Seed` 表示随机数种子，`X` 和 `Y` 必须至少有一个为素数，即除了 1 和它本身，不能被其他数整除的数。而 `Z` 也应该是一个素数，这样才能保证产生的随机数能得到上限的值，尽量使生成的数字均匀分布在指定的范围内。在 Java 中，可以使用 `java.util.Random` 类来产生一个随机数发生器。

它有两种形式的构造函数, 分别是 `Random()` 和 `Random(long seed)`。`Random()` 使用当前时间即 `System.currentTimeMillis()` 作为发生器的种子, `Random(long seed)` 使用指定的 `seed` 作为发生器的种子。

随机数发生器即 `Random` 对象产生以后, 可以通过对象调用不同的函数: `nextInt()`、`nextLong()`、`nextFloat()`、`nextDouble()` 等来获得不同类型的随机数。本设计中调用 `Random` 类产生随机数的代码如下:

```
import java.util.*;
class TestRandom {
    public static void main(String[] args) {
        Random random = new Random();
        for(int i = 0; i < 10; i++) {
            float f=Math.abs(random.nextInt()) % 10;
            ... ..
        }
        ... ..
    }
}
```

本次软件模拟设计中用 `java.util.Random` 类产生范围在 0~9 的随机浮点数, 保留小数点后两位, 以此作为暂停进程命令 `Sleep()` 的时间参数, 来模拟网络延迟的情况, 并作为路径的属性反馈给路由管理器, 供其更新维护路径信息库。

4.3.3 模拟节点之间的数据收发流程

TIP/IP 网络协议提供了在不同硬件体系结构和操作系统的计算机组成的网络上进行通信的能力。一台 PC 机通过 TCP/IP 网络协议可以和多个远程计算机 (即远程节点) 进行实时通讯。Jbuilder9.0 中提供了各种网络开发的组件运行在基于 TCP/IP 网络协议的网上, 可以快速生成简捷安全的网络应用程序, 使用户方便地进行联网。

模拟原型的设计运用了 `JAVA.net` 包中定义的两个类: `ServerSocket` 和 `Socket`, 来实现节点间的收发数据过程。其中数据源节点 A 与路径管理器 S 的通信流程如图 4.10 所示:

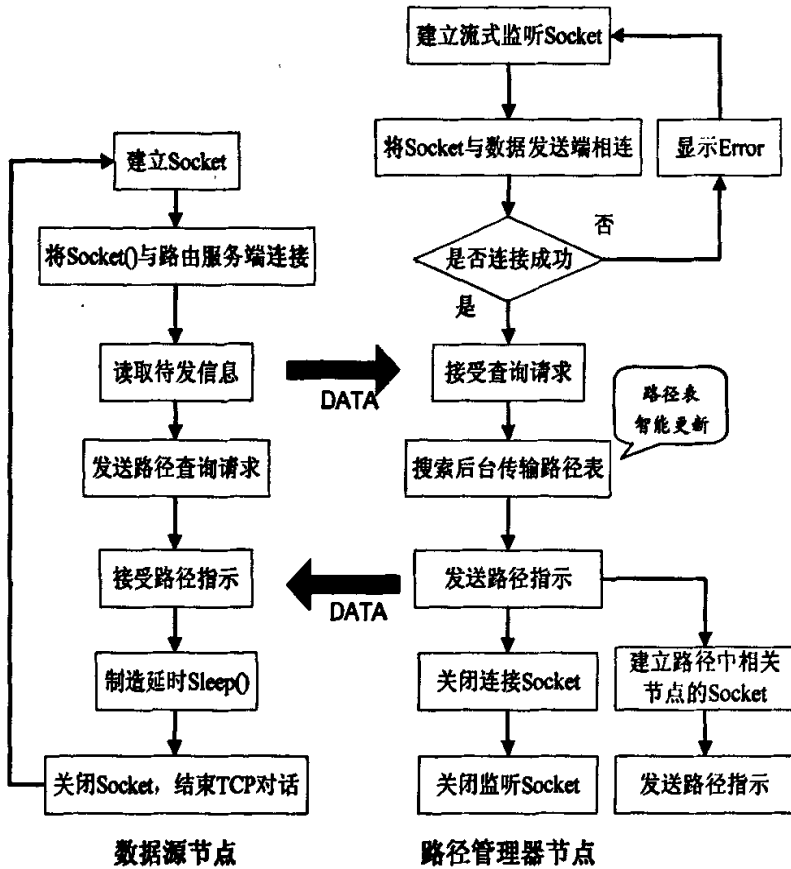


图 4.10 数据源节点与路径管理器节点通信流程图

路由管理器节点：

- (1) 创建 ServerSocket 对象；
- (2) 执行 accept() 方法，等待数据节点的连接请求；
- (3) 响应数据节点的连接请求，建立连接；
- (4) 创建输入流获取请求内容；
- (5) 查询路径信息库得到路径指示；
- (6) 创建输出流发送路径指示；
- (7) 关闭此次 Socket 对象，继续监听；

数据源节点：

- (1) 创建 Socket 对象，发出连接请求；

- (2) 建立连接;
- (3) 创建输出流发送路径查询请求;
- (4) 创建输入流获取路径指示;
- (5) 关闭此次 Socket 对象, 等待下一次的数据发送;

数据传输节点 1、2、3、4 与路径管理器 S 之间同样用 Socket 套接字来实现通信, 但此时路径管理器节点要向数据传输节点发送路径指示, 所以是建立连接端, 而数据传输节点执行 accept() 方法, 等待路径管理器的连接。连接成功后, 数据传输节点得到路径指示, 明确了应该将数据传至哪个节点, 同时将此通路的 QoS 信息发送给路径管理器节点, 使其及时更新维护后台的路径信息库。它们的通信流程如图 4.11 所示:

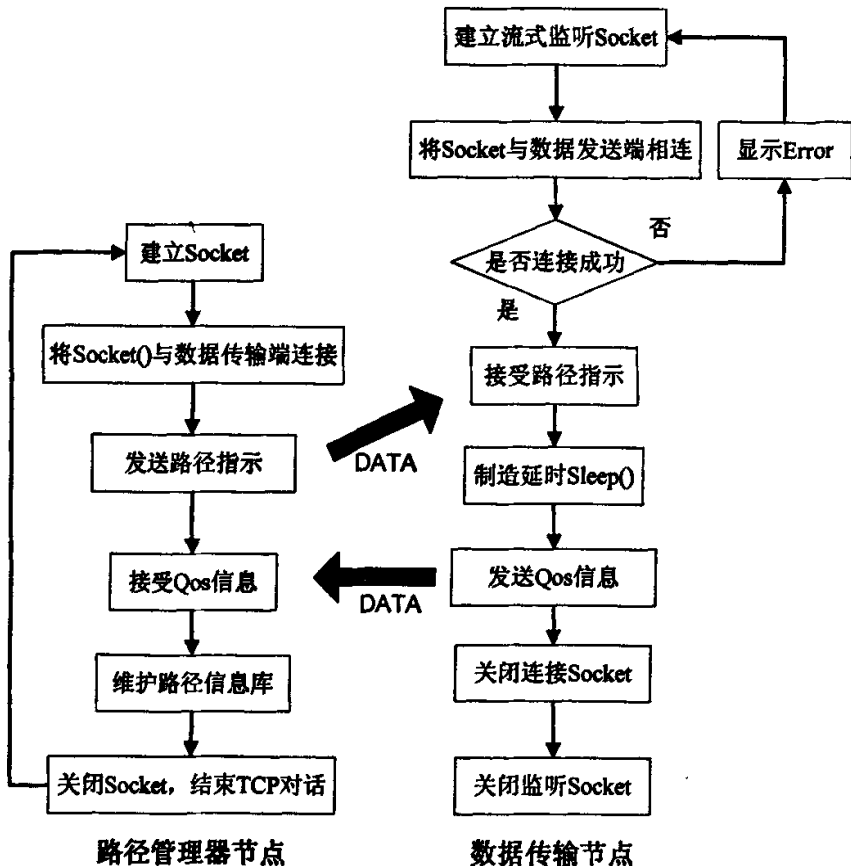


图 4.11 路径管理器节点与数据传输节点通信流程图

虽然采用有连接的基于 TCP/IP 协议的 Socket 通信方式不如 UDP 通信方式迅速，但是 Socket 通信更加可靠安全，特别是对于控制信息的传输，可靠收发是相当重要的。

4.3.4 系统模拟实现图

首先根据面向对象的软件分析与设计原则，为了使系统简单、直观，易于用面向对象语言编程实现，从软件静态建模的角度，用表 4.1 描述本软件模拟原型中的网络节点。

表 4.1 模拟节点分类

节点类型	节点实名	功能属性	操作
路由管理器	S	发送路径指示； 维护路径信息库；	Socket 套接字 加权排序算法
数据传输节点	A,1,2,3,4,F	发送和接收数据； 发送链路属性；	Socket 套接字 Sleep(Random());

然后用标准的图形化建模语言 UML 的动态模型序列图来描述软件模拟原型中的消息如何在节点对象中发送和接收。序列图中的消息分为简单消息、同步消息和异步消息三类，其中“简单消息”是指普通的控制流。它只表示控制是如何从一个对象传给另一个对象，而不描述通信的任何细节；“同步消息”是指一个嵌套控制流，表示一个操作调用。处理消息的操作在调用者恢复执行之前完成（包括任何在本次处理中发送的其他消息）；“异步消息”是指异步控制流中，没有直接的返回给调用者，发送者发送完消息后不需要等待消息处理完成而是继续执行，也就是实时系统中的并发执行。

如图 4.12 所示，路径管理器节点向数据传输节点发送的路径指示都是并发执行的异步消息；而数据传输节点间的通信一般情况下为同步消息，但当某节点为通信路径的公共节点时（如 P1），则它会在得到路由指示后并行向两条路径上的后续节点发送数据；并且各数据传输节点还将向路径管理器节点发送相应的传输链路的状态属性，作为维护路径信息库的算法参数。

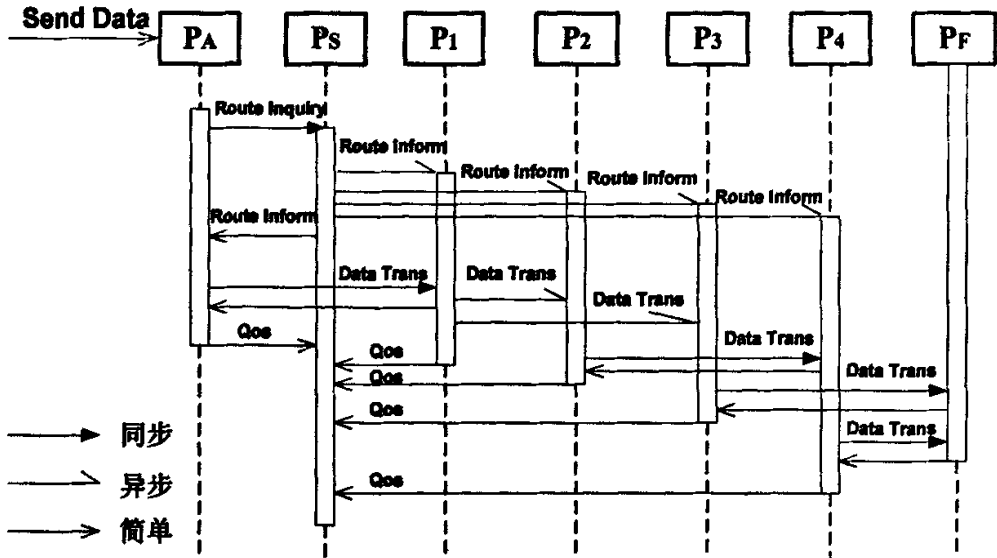


图 4.12 系统通信序列图

为了便于了解节点的实际工作情况，为本软件模拟原型中的各个节点设计了具有一致或者相似风格的面向用户的界面。图 4.13、图 4.14、图 4.15 和图 4.16 显示了本软件模拟原型执行从把数据从节点 A 传输到节点 F 这一数据传输过程的各相关窗口显示情况。

图 4.13 显示了节点 A 的窗口界面以及开始发送数据的连接情况。窗口中的“Requirement”状态栏可以输入路径查询请求，在按下“Send”按钮后，该状态栏中的请求就被发送到担任路径管理器的节点 S 处，用以模拟网络中的数据源节点向路径管理器节点发送路径查询请求的行为。当请求发送成功时，系统就弹出“连接成功”对话框，表示节点 A 和节点 S 之间已经建立了 Socket 通信机制；“Route”状态栏可以显示路径管理器节点 S 向节点 A 发出的路径指示命令，节点 A 就按照此指示将“Data”状态栏中的数据传输至节点 1；而窗口中的“Delay”状态栏用来显示用随机数发送器模拟的数据从节点 A 传至节点 1 所经历的网络延迟时间。

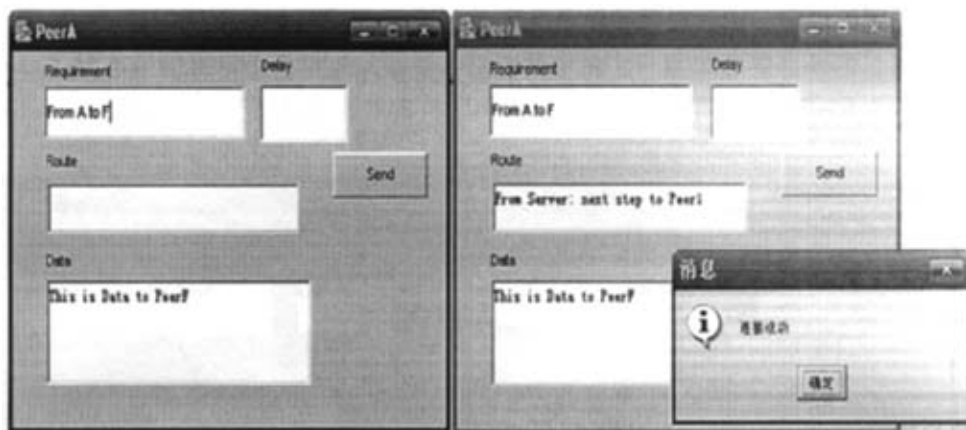


图 4.13 节点 A 开始发送数据窗口

图 4.14 显示了节点 A 和节点 1 在数据传输过程中的窗口界面。节点 A 的“Delay”状态栏显示了数据从节点 A 传至节点 1 的模拟网络延迟时间为 5.41ms；节点 1 的窗口与节点 A 风格一致，“Route, Delay, Data”各状态栏的功能也与节点 A 相同。由于节点 1 是两条传输路径的公共节点，因此得到路径管理器的指示是将从节点 A 处接收到的数据分别发送至节点 1 和节点 2。并且显示沿此两条路径发送产生的两个模拟网络延迟时间，分别为节点 1 至节点 2 之间的 2.38ms 和节点 1 与节点 3 之间的 6.07ms。

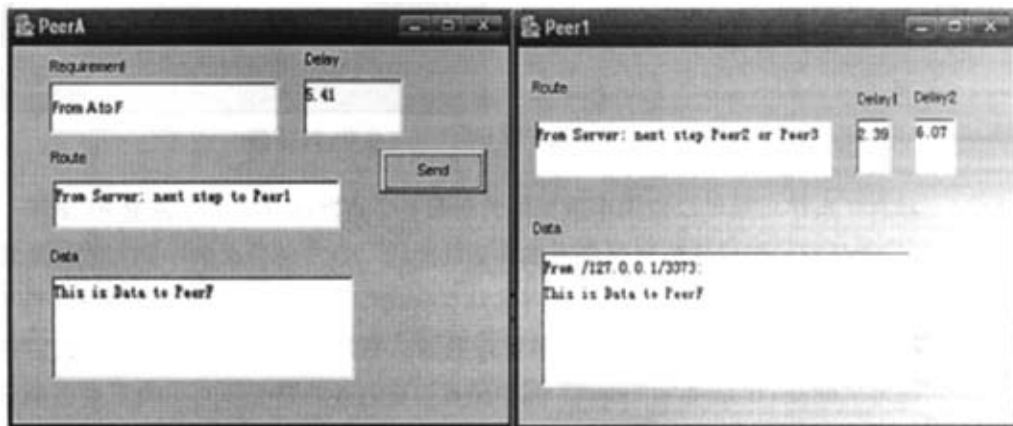


图 4.14 节点 A、节点 1 显示延时窗口

图 4.15 显示了节点 1、节点 2 和节点 3 在数据传输过程中的窗口界面。“Data” 状态栏则显示了各节点从数据传输路径中的上一跳节点接收到的数据。“Delay” 状态栏显示了这段接收数据过程的模拟网络延迟时间。

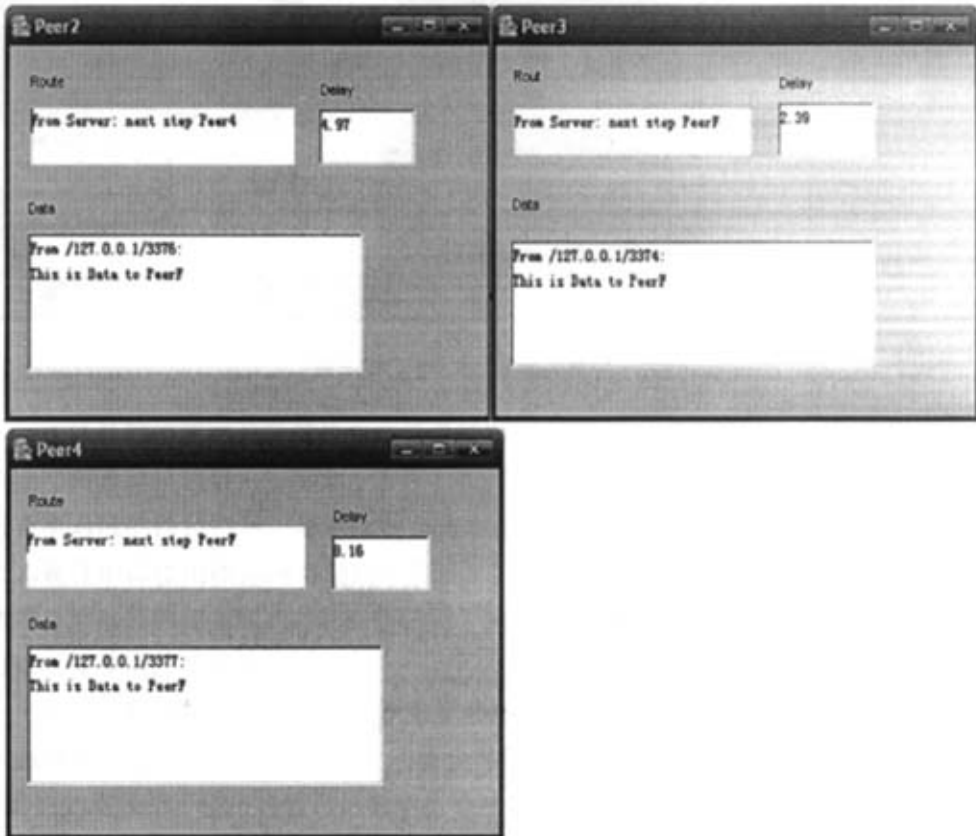


图 4.15 节点 2、节点 3 和节点 4 窗口

图 4.16 显示了节点 F 和节点 S 在数据传输过程中的窗口界面。节点 F 在这次数据传输过程中作为目标节点，因此并没有得到路径管理器发送的路径指示，也不存在模拟的网络延迟时间。“Data” 状态栏则显示了节点 F 从两条路径先后接收到的相同数据，其先后顺序与程序设置的模拟网络延时有关。而节点 S 承担路由由管理器节点的工作任务，窗口界面中的“Peer’s Requirement” 状态栏将显示数据源节点的数据传输目标。节点 S 的后台还存在着模拟的路径信息数据库。

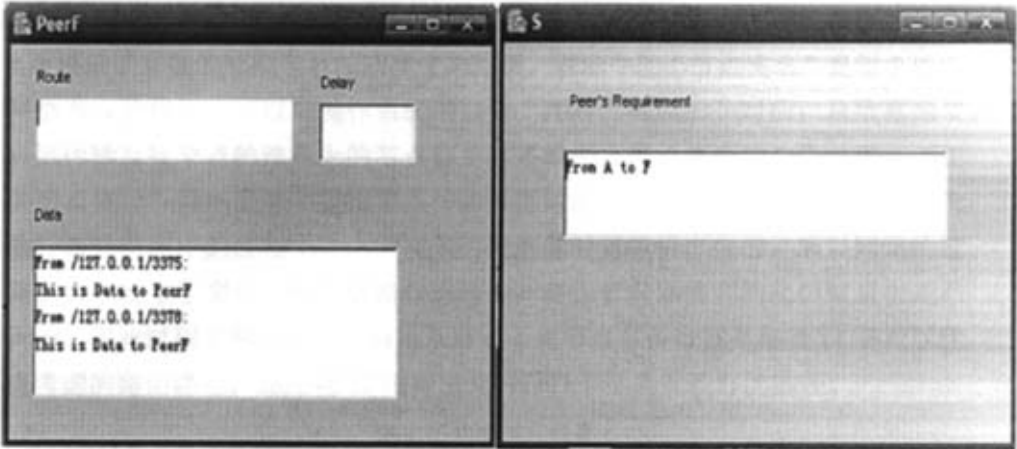


图 4.16 节点 F、节点 S 窗口

本软件模拟原型的路径信息库简单设置为表 4.2 所示的形式，该数据库中列出了两条路径记录，分别为 A-1-2-4-F 和 A-1-3-F，以及这两条路径中包含的节点所对应的路径指示，还有这些节点本身与下一跳节点之间的链路状态属性（本设计中即使用模拟网络延迟来表示）。并且用“1”和“0”来代表这两条数据传输路径的优劣性，“QoS 评估值”为“1”的路径就是下一次传输数据时可以优先使用的质优路径。

在本软件模拟原型每执行一次数据传输任务后，该信息数据库中的“链路状态属性参数”项和“QoS 评估”项将会得到一次更新，以保证路由管理器节点 S 给出的路径指示适应传输网络的动态特性。

表 4.2 路径信息数据库的设置

路径记录	节点	路径指示 (Send To:)	链路状态属性参数 (ms)	QoS 评估值
A-1-2-4-F	A	1	5.41	0
A-1-2-4-F	1	2	2.39	0
A-1-3-F	1	3	6.07	1
A-1-2-4-F	2	4	4.97	0
A-1-3-F	3	F	2.39	1
A-1-2-4-F	4	F	8.16	0

经过程序编写编译、运行和反复调试，本软件模拟原型系统能够正常运行，实现了简单模拟控制网络多路径传输模式的各功能节点工作过程的目标。如表 4.3 显示了系统的模拟运行三次产生的结果，表项中的“节点实名”是每个模拟网络节点的代号，可以与操作系统端口号“PORT”可以一一对应；从数据源节点到传输目标节点的路径上的节点都是这条路径上节点集合的一个元素，“相关节点集合”列出了每个模拟节点传输数据时的上一跳和下一跳节点；在三次执行该系统程序的过程中，由于模拟的网络延时是随机产生的，所以网络的链路状态也发生了变化，根据链路状态属性，路由管理器得到的反馈信息也是不同的，因此过程 I 和过程 III 得出从节点 A 向节点 F 传数据是路径 II 是可以优先考虑的路由路径，而过程 II 则得出此时路径 I 是属性较好的路径。

表 4.3 模拟运行结果

节点实名	相关节点集合	PORT	路径	过程 I 延迟 (ms)	过程 II 延迟 (ms)	过程 III 延迟 (ms)
1	{A, 1, 2, 3}	6002	A-1	5.41	8.61	2.32
2	{1, 2, 4}	6004	1-2	2.39	6.89	4.15
3	{1, 3, F}	6006	1-3	6.07	3.36	5.86
4	{2, 4, F}	6008	2-4	4.97	2.97	7.58
F	{3, 4, F}	6010	4-F	8.16	0.27	7.09
			3-F	2.39	7.73	3.15
路径 I: A-1-2-4-F 传输总延迟 (ms)				20.93	18.74	21.14
路径 II: A-1-3-F 传输总延迟 (ms)				13.87	19.7	11.33
链路属性好的路径				II	I	II

在以上三次数据传输过程中，随着模拟网络延迟时间的变化，链路状态参数也有所变化，不同过程中属性好的适时最优路径也各不相同，体现了网络控制系统中传输网络复杂、时变的动态特性。本软件模拟原型系统是利用操作系统的多线程协作完成。图 4.16 显示了运行此程序时工作站主机的 CPU 使用记录和页面文件使用记录。

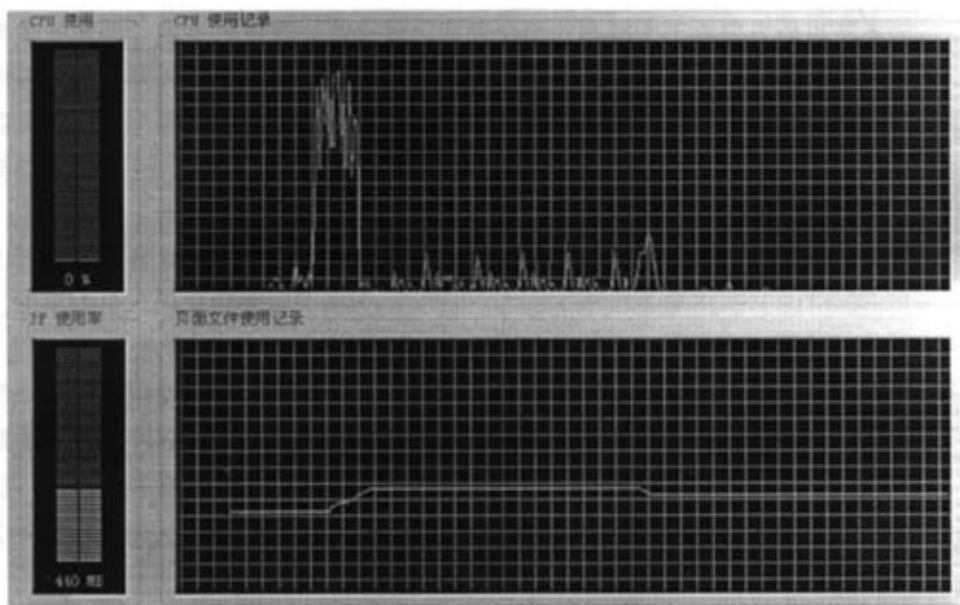


图 4.15 CPU 和页面文件的使用记录

可以看出，在模拟节点 A 向模拟路径管理器 S 发送路径查询请求后，路径管理器立即调用路径信息库中的记录，向各节点发送路由指示，此时的工作站主机 CPU 的占用率是最高的。此后的数据传输就按照路由管理器的指示开始执行，占用的系统资源就较少。

4.4 本设计与相关技术的联系

本课题研究的根本方向在于扩展网络控制系统针对传输网络部分的相关控制理论的新范例及模式。在设计过程中综合了多项控制工程技术和网络技术思想精髓。

本模式的设计思想中“以数据为中心”和“面向应用的网络”的特点在传感器网络工程中已经有所应用。在传感器网络中，传感器节点不需要地址之类的标识，脱离传统网络的寻址过程，快速有效地组织起各个节点的信息并融合提取出有用信息直接传送给用户，可类比于本设计中以实名表示网络节点的通信方式；而且传感器网络的各个节点随机分布于某一区域，以自组织形式构成网络。网络中部分或全部节点可以移动，并根据变化的任务和网络拓扑需求，自动进

行聚集和重组，动态地适应节点失效与性能恶化，可类比于本模式中以节点间通信链路属性状态为参数更新数据传输路径的设计思想；可以说，当多路径传输模式在实际的基于 Internet 的 NCS 中得到推广，就形成了一个“控制器网络”的新概念；

本模式是工作于互连网络应用层之上的进程服务，以便能够理解应用层上的协议，做一些复杂的访问控制，并实现精细的注册和稽核工作。这一设计思想与防火墙技术的应用级网关相近；

其他方面，本模式中路径管理器的设计借鉴了互联网中搜索引擎数据库的加权排序算法；网络节点的功能自治性和目录式管理特性则体现了对等网络技术的节点路由机制原理；对于网络中节点通信链路中被管理的网络设备（如：路由器、网管服务器、交换机、网桥、集线器等）的控制方案则借鉴了网络管理的相关技术思想。

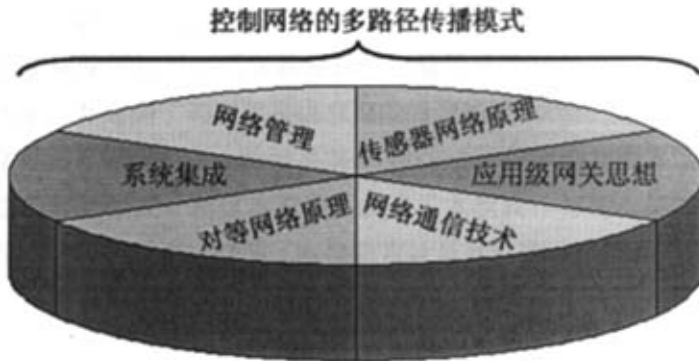


图 4.16 与相关技术的联系

需要指出的是，本文提出的多路径传输模式的实际通信过程依然是建立在 OSI 参考模型上的，并没有在硬件上改变互连网络的结构，而是利用控制的思想和网络系统集成的理论合理利用网络设备、协议、系统软件、应用软件等资源，建设一个能解决基于 Internet 的网络控制系统所存在的相关问题的创新控制模式。

4.5 本章小结

本章主要对控制网络多路径数据传输模式的设计过程进行了详细的解析。首先通过将此模式与现有的 NCS 技术作比较,说明了本课题设计并不是像大多数现有技术那样依靠假设延迟模型,改进网络控制系统的控制器参数,增加补偿环节等方法进行的多限于理论或仿真的研究。控制网络多路径数据传输模式的设计更着重于实际应用的角度,基于网络系统集成的思想,把如何利用现有的互联网络的对等通信模式,减少网络数据传输时延迟,减少数据包的丢失和排列顺序混乱问题,提高数据包的传输成功率,提升网络控制系统的实时性和可靠性作为研究的目的。在全局性地叙述了本模式的设计概要后,再针对模式中的不同功能部分的设计思想和工作原理进行详细描述。

然后介绍了用 JAVA 语言编写了一个模拟控制网络多路径数据传输模式的软件原型。解释了该软件原型系统的设计思路,运用 JAVA 语言的编程规范说明了程序编写中各项功能的实现原理,又介绍了模拟节点之间数据接收和发送的流程。最后给出了软件模拟原型用多路径数据传输模式执行数据传输任务的窗口界面和运行结果,并介绍了模拟路径信息库的设置情况。

由于本文创新性地提出了一种解决基于 Internet 的网络控制系统所存在的相关问题的创新控制模式,所以本章最后叙述了本次设计与现有相关控制工程技术和网络技术之间的联系,强调了本课题设计是具有实现可能和应用意义的。

第5章 结论与展望

5.1 全文工作总结

网络控制系统的发展代表了下一阶段信息革命的方向。在理论研究方面，可以促进控制、计算机、微电子和通讯技术等多门学科的交叉渗透，使控制理论和技术进入一个崭新的阶段；在实践方面，能够解决远程实时控制遇到的各种困难。NCS的发展对于控制系统的进步是一个巨大的机遇和挑战，需要对各种专业知识（包括控制理论、计算机科学、网络通信技术等进行融通与应用。

本文主要针对基于 Internet 的网络控制系统存在的数据传输延迟、数据包丢失和排列顺序混乱问题，提出了一种创新的解决方案。现有的网络路由方式传输数据时是单向的、不可靠的（消息可能丢失）、异步的（多个消息不按次序到达），传输数据的网络节点在得到数据包前并未确定将要发送该数据下一个节点；而本文提出的多路径数据传输模式利用网络系统集成原理为具有功能自治特性的对等网络节点设置了路由管理器和数据传输两类功能，同时建立了存放节点路由指令的路径信息数据库，从而形成了一个新式的控制网络数据传输模式。

本文的主要工作包括：

- (1) 论述了优化网络控制系统传输网络的重要性，列举分析了传输网络存在的问题，用计算机图论的方法表示网络链路状态属性和网络节点的拓扑结构；
- (2) 叙述了与本设计相关的多项理论技术和相关原理，并将这些控制工程、网络通信方面的关键技术综合集成起来，组织成为一体，使之成为能够满足设计目标的理论基础；
- (3) 编写了一个多路径传输模式的软件模拟原型系统，实现了该模式的基本工作过程。该软件模拟原型设立了模拟的网络节点，用进程的随机暂停产生了模拟的网络传输延迟，并附有相关的路径信息数据库原型。该软件模拟原型系统包含面向用户的窗口界面，能够正常运行，实现了简单模拟控制网络多路径传输模式的各功能节点工作过程的目标。

本文的设计思想创新点是在 NCS 传输网络的建设中引入了“控制”的理念，使其由“平面网络”提升到“层次网络”的概念，

5.2 进一步工作的方向

本设计的主要任务是提出一种解决基于 Internet 的网络控制系统所存在的相关问题的创新控制模式。本文虽然初步解析了 NCS 控制网络多路径数据传输模式的原理与实现过程，但本设计涉及的网络建设规模宏大，限于研究时间，有些内容还有待进一步深入在以后的工作中进一步验证和完善：

(1) 多路径数据传输模式目前只是初步的理论研讨阶段。

本文虽然用软件原型演示了该模式的基本工作过程，但还可以作一些更深入的研究工作。例如：分析多路径数据传输的成功率，即研究数据发送的成功率与路径数量的关系等；可以引入人工神经网络、蚁群算法等现代控制理论的相关技术来更好地维护路径信息库；

(2) 把多路径数据传输模式应用到实际的网络控制系统加以分析验证。

真实的互连网络有节点加入、退出网络、多样化的网络安全技术等更加复杂的情况。必将对多路径数据传输模式的可行性提出更高的考验，而整个网络控制系统的其他模块也存在着状态的不确定性，网络控制系统的服务性能与控制性能是一个相互矛盾而又相互联系的整体，能否找到两者间性能的优化点。这是一个值得研究的课题。

致谢

经过两年多的学习，终于完成了毕业设计工作。欣慰之余，真诚的感谢为论文写作直接、间接提供帮助、支持和关心的老师、同学、朋友和家人。

衷心的感谢我的导师许维胜教授。该论文是在导师的悉心指导下完成的。许老师对事业的不懈追求、一丝不苟的敬业精神、严谨的治学态度、饱满的工作热情，深深地影响着我，鼓舞我在求学路上不断前进，是我人生道路中学习的楷模。

感谢许老师在我攻读硕士学位的过程中给予的各种有形无形的支持和关怀，尤其是那些无形的思想渗透和折射，以及对我的鼓励和支持。这两年半的珍贵时光将成为我人生中最重要阶段并将使我受益终生。

感谢余有灵副教授在我课题的完成过程中给予的帮助和指导。并感谢半导体信息技术研究所的王翠霞老师、郭丽娜博士、彭俊同学、陆莹同学、刘晓明同学、江兆根同学以及其他同学们在我完成毕业设计的过程中给予我的帮助和鼓励。

我从刚开始对课题的难以把握到最后顺利完成的整个过程里都体现了半导体信息技术研究所实验室的各位老师同学间积极融洽的学术研究氛围和互相协助的工作态度。让我的整个读研期间都非常充实愉快，学到了许多书本上没有的知识，磨练了刻苦钻研的意志，增长了个人能力，得到了各方面的锻炼。

特别要感谢我的父母家人，始终默默地为我付出，关心着我的学习和生活情况，在我遇到困难的时候给予我最大的鼓励支持，是你们一路的关爱和理解帮助我实现了学习目标，顺利完成了学业。真心地谢谢！

最后，感谢所有参加论文评审的专家，感谢你们在百忙之中对我的论文提出批评指正。

2007年3月

参考文献

- [1] Halevi Y, Ray A. Integrated Communication and Control System: Part I —Analysis. ASME Journal of Dynamic Systems, Measurement and Control, 1988A, Vol. 110(4), 367-373.
- [2] Liou L.W., and Ray A., An integrated communication and control systems: Part II-nonidentical sensor and controller sampling, Journal of Dynamic Systems, Measurement, and Control, 1990, 112(4):357~364
- [3] Park H.S., Kim Y. H., and Kim D. S., A scheduling method for network-based control systems, IEEE Transactions on Control Systems Technology, 2002, 10(3):318~330
- [4] Walsh G. C., Hong Y. & Bushnell L.G, Stability Analysis of Networked Control Systems. IEEE Transactions on Control Systems Technology, 2002, 10(3):438~446
- [5] Nilsson J., Bernhardsson B., Analysis of real time control systems with time delays. Proceeding of the 35th IEEE CDC, Kobe, Japan, 1996: 3173-3178
- [6] Otanez P.G, Parrott J.T., Moyne J.R. and Tilbury D.M., The Implications of Ethernet as a Control Network.Global Power rain Conference, Ann Arbor, MI, 2002, pp. 1-9.
- [7] Krommenacker N., Rondeau E. and Divoux T., Genetic Algorithms for Industrial Ethernet Network Design.4th IEEE International Workshop on Factory Communication Systems, Vasteras, Sweden, 2002, 28-30.
- [8] Tanenbaum A.S., Computer networks.Prentice-Hall International Editions.Prentice-Hall, Upper Saddle River NJ, 3 editions, 2004.
- [9] Chow M.Y., & Tipsuwan Y., Network-based control adaptation for network QoS variation, IEEE MILCOM 2001, Vienna, VA, 2001, 257-261
- [10] 王连明, 邓玉芬, 陈海霞. 基于网络的控制技术研究[J]. 东北师大学报. 2004. 6: 3438
- [11] 蒋婕. Internet 网络延迟分析[J]. 广东通信技术. 2003. 2. : 20-2
- [12] Zhang W., Stability Analysis of Networked Control Systems.PhD dissertation, Case Western Reserve University, 2001A.
- [13] Babak Azimi-Sadjadi. Stability of networked control systems in the presence of packet losses. Proceedings of the 42nd IEEE Conference on Decision and Control Maui, Hawaii USA, December 2003
- [14] Anton C., Improved scheduling of control tasks, Proceedings of the 11th Euromicro Conference on Real-Time Systems, York, England, June 1999.
- [15] 王飞跃. 基于网络控制的若干基本问题的思考和分析. 自动化学报. 2002, Vol128, 171-176.
- [16] Nilsson J., Bernhardsson B., Wittenmark B., Stochastic analysis and control of real-time systems with random time delays. Automatica, 1998A, 34(1):57-64
- [17] 朱其新, 吴建国, 陆国平, 堵俊, 周晖, 姜平. 网络控制系统中的基本问题与时延分析. 南

- 通工学院学报(自然科学版)[J]. 2003 .09:40-43
- [18] 张婷, 王建国, 赵锦东. 一种新型的基于 P2P 网络的发现机制的设计[J]. 计算机工程与设计. 2005. 9. :2509-2511
- [19] 董芳, 费新元, 肖敏. 对等网络 Chord 分布式查找服务的研究[J]. 计算机应用. 2003. 11 (23) :25-28
- [20] Ramadas Shanmugam,R.Padmini,S.Nivedita. TCP/IP 详解.电子工业出版社 第二版.2003 年 8 月
- [21] Stevens, WR. TCP/IP Illustrated, Vol.I Addison-Wesley, 1994
- [22] James F.K., Keith W.R., Computer Networking A top-Down Approach Feachturing the Internet (Third Edition), Pearson Education, Inc., 2005.
- [23] 韩琳娜, 马正新, 胡捍英. QoS 网络最优节点状态搜索算法研究.西安电子科技大学学报(自然科学版). 2005 年 12 月: 974-976
- [24] Jonathan H. C. & Guo X. L., Quality of Service Control in High-Speed Networks, John Wiley & Sons, Inc., 2002.
- [25] Luen-Woei Liou, A.Ray. Integrated communication and control systems: part III-nonidentical sensor and controller sampling[J].
- [26] O.V.Beldiman. Networked control system[D]. U.S.A.: Duke University, 2001
- [27] GC.Walsh, H.Ye, L.GBushnel. Stability analysis of networked control system[J]. IEEE Trans. On Control Systems Tech, 2002, Vol-10(3):438-446
- [28] Y.Tipsuwan, M.Y.Chow. Network-based controler adaptation based on QoS negotiation and deterioration[A]. Proceedings of IECON'01[C]. U.S.A.: IEEE, 2001, 1794-1799.
- [29] Y.Tipsuwan, M.Y.Chow. Network-based controller adaptation based on QoS negotiation and deterioration. Proc. Of IECON'01, 2001. 1794-1799
- [30] M.Y.Chow, Y.Tipsuwan. Network-based control adaptation for network QoS variation. Proc. Of MILCOM2001, McLean, Va, 2001: 257-261
- [31] Y.Tipsuwan, M.Y.Chow. Gain adaptation of networked mobile robot to compensate QoS deterioration. 2002. 3146-3151
- [32] R.Lock, A.Ray. An observer-based compensator for distributed delays[J]. Automatica, 1990, Vol.26: 903-908
- [33] 杨卫东. 网络系统集成与工程设计. 北京科海电子出版社页数 第二版. 2005 年 10 月
- [34] M.Garey, D.John son. Computer and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco, 1979
- [35] Salama H.F. Multicast Routing for Real-Time Communication on High-Speed Networks. [Ph.D dissertation], Dept of Electronic and Computer Engineering, North Cralina State University, 1996
- [36] Li Layuan. The routing protocol for dynamic and large computer networks. Journal of computers, 1998, 11(2):137 - 144
- [37] Donald E.Knuth. The Art of Computer Programming, Vol.2, Seminumerical Algorithms 3rd Edition published by Addison Wesley Longman

参考文献

- [38] Andrew Rukin, Juan Soto, James Nechvatal, etc. "A Statistical Test Suite for Random and Pseudorandom Number generators for Cryptographic Applications".NIST Special Publication 800-22, 2001.3.15
- [39] J.Patrick Thompson.Web-Based Enterprise Management Architecture.IEEE Communication Magazine, 1998, 36(3):80-86
- [40] K.Mccloghrie, M.Rose.Management Information Base for Network Management of TCP/IP-Based Intenets, RFC 1156,1990
- [41] William Stallings, SNMP and SNMPv2.The Infrastructure for Network Management.IEEE Communications Magazine, 1998, 36(3):37-43
- [42] Braden B, Clark D, Crowcroft J, et al. Recommendations on Queue Management and Congestion Avoidance in the Internet [DB/OL]. RFC2309, 1998.
- [43] 杨思洛.搜索引擎的排序技术研究.《现代图书情报技术》2005年第1期:43-47
- [44] Korkmaz T. & Krunz M., Multi-Constrained optimal path selection, Proceedings of the IEEE INFOCOM, 2001, 834-843

个人简历 在读期间发表的学术论文与研究成果

个人简历:

仇鹤凤, 女, 1981年10月生。

2004年6月毕业于同济大学 自动化专业 获学士学位。

2004年9月入同济大学读硕士研究生, 控制理论与控制工程专业。

已发表论文:

[1] 仇鹤凤, 余有灵. 基于P2P网络的发现机制. 电脑知识与技术. 学术交流, 2006, No. 10

[2] 仇鹤凤. 可靠性测试技术 MEOST 及其应用实例. 新技术新工艺, 2006, No. 10

待发表论文:

[1] 仇鹤凤. 基于模糊控制的液体滴速测控系统. 计算机应用研究, 2007.4 (已录用)