|  |  |
| --- | --- |
|  | 学校代码： 10246 |
|  | 学 号： 093053024 |



|  |
| --- |
| 硕 士 学 位 论 文 |

（专 业 学 位）

|  |
| --- |
| **组播与单播相结合的下载系统的设计和实现** |

|  |  |
| --- | --- |
| 院 系： | 软件学院 |
| 专 业： | 软件工程 |
| 姓 名： | 张 晓 欧 |
| 指 导 教 师： | 徐 迎 晓 |
| 完 成 日 期： | 2011年2月22日 |

目录

[摘要 1](#_Toc292873140)

[Abstract 2](#_Toc292873141)

[第一章 引言 4](#_Toc292873142)

[1.1. 现有解决方案和系统的分析 4](#_Toc292873143)

[1.1.1. 拷贝机方案及其缺陷 4](#_Toc292873144)

[1.1.2. 纯单播下载系统及其缺陷 5](#_Toc292873145)

[1.2. 新系统的解决方案及意义 6](#_Toc292873146)

[1.3. 本论文的篇章结构 7](#_Toc292873147)

[第二章 网络模型和理论 9](#_Toc292873148)

[2.1 OSI参考模型 9](#_Toc292873149)

[2.2 TCP/IP参考模型 9](#_Toc292873150)

[2.2.1. TCP通信协议 10](#_Toc292873151)

[2.2.2. UDP通信协议 10](#_Toc292873152)

[2.3 网络组播 11](#_Toc292873153)

[2.3.1 基于IP的组播模型 11](#_Toc292873154)

[2.3.2 组播转发 12](#_Toc292873155)

[2.4 套接字概述 12](#_Toc292873156)

[第三章 系统需求分析 13](#_Toc292873157)

[3.1. 业务流程描述 13](#_Toc292873158)

[3.1.1 笔记本电脑生产流程概要 13](#_Toc292873159)

[3.1.2 出厂软件下载及安装工序 14](#_Toc292873160)

[3.2. 功能性需求 15](#_Toc292873161)

[3.2.1. 业务数据的日常维护 16](#_Toc292873162)

[3.2.2. 生产数据的查询 16](#_Toc292873163)

[3.2.3. 组播下载 17](#_Toc292873164)

[3.2.4. 单播下载 18](#_Toc292873165)

[3.2.5. 负载均衡 18](#_Toc292873166)

[3.3. 系统角色 19](#_Toc292873167)

[3.3.1. 用户角色 19](#_Toc292873168)

[3.3.2. 硬件角色 20](#_Toc292873169)

[3.4. 非功能性需求 21](#_Toc292873170)

[3.4.1. 低成本、低能耗的需求 21](#_Toc292873171)

[3.4.2. 可靠性和茁壮度的需求 21](#_Toc292873172)

[3.4.3. 方便重构、易于升级的需求 22](#_Toc292873173)

[3.4.4. 安全的需求 22](#_Toc292873174)

[第四章 系统架构设计 23](#_Toc292873175)

[4.1. 系统架构设计所面临的问题 23](#_Toc292873176)

[4.2. 系统设计的思路 23](#_Toc292873177)

[4.3. 系统软件架构的设计 24](#_Toc292873178)

[4.4. 系统内服务器角色的设计 25](#_Toc292873179)

[4.5. 系统模块的功能设计 26](#_Toc292873180)

[4.5.1. 前台模块群 26](#_Toc292873181)

[4.5.2. 中间层业务逻辑模块群 27](#_Toc292873182)

[4.5.3. 后台数据库 28](#_Toc292873183)

[4.6. 各主要模块间的协作关系 29](#_Toc292873184)

[4.6.1. 安全认证模块与相关模块的关系 29](#_Toc292873185)

[4.6.2. 服务器控制台和相关模块的关系 29](#_Toc292873186)

[4.6.3. 业务管理控制台和相关模块的关系 30](#_Toc292873187)

[4.6.4. 日志类各模块的关系 30](#_Toc292873188)

[4.6.5. 负载均衡模块和相关模块的关系 30](#_Toc292873189)

[4.7. 系统网络架构的设计 31](#_Toc292873190)

[4.7.1. 网络层的组播实现 31](#_Toc292873191)

[4.7.2. 网络层的单播实现 31](#_Toc292873192)

[第五章 核心模块的详细设计 32](#_Toc292873193)

[5.1 文件分块模块的详细设计 32](#_Toc292873194)

[5.1.1 模块详细功能 32](#_Toc292873195)

[5.1.2 流程设计 32](#_Toc292873196)

[5.1.3 接口设计 33](#_Toc292873197)

[5.1.4 类的设计 33](#_Toc292873198)

[5.1.5 接口或者类的实现 34](#_Toc292873199)

[5.2 组播下载服务器模块的详细设计 35](#_Toc292873200)

[5.2.1. 模块详细功能 35](#_Toc292873201)

[5.2.2. 流程设计 35](#_Toc292873202)

[5.2.3. 接口的设计 36](#_Toc292873203)

[5.2.4. 类的设计 36](#_Toc292873204)

[5.2.5. 接口或者类的实现 37](#_Toc292873205)

[5.3 数据修补模块的详细设计 38](#_Toc292873206)

[5.3.1. 模块详细功能 38](#_Toc292873207)

[5.3.2. 流程的设计 39](#_Toc292873208)

[5.3.3. 接口的设计 39](#_Toc292873209)

[5.3.4. 类的设计 40](#_Toc292873210)

[5.3.5. 接口或者类的实现 41](#_Toc292873211)

[5.4 下载客户端模块的详细设计 42](#_Toc292873212)

[5.4.1. 模块详细功能 42](#_Toc292873213)

[5.4.2. 流程的设计 42](#_Toc292873214)

[5.4.3. 接口的设计 43](#_Toc292873215)

[5.4.4. 类的设计 43](#_Toc292873216)

[5.4.5. 接口或者类的实现 45](#_Toc292873217)

[5.5 负载均衡模块的详细设计 46](#_Toc292873218)

[5.5.1. 模块详细功能 46](#_Toc292873219)

[5.5.2. 流程的设计 47](#_Toc292873220)

[5.5.3. 接口的设计 47](#_Toc292873221)

[5.5.4. 类的设计 48](#_Toc292873222)

[5.5.5. 接口或者类的实现 49](#_Toc292873223)

[第六章 系统的效能分析及主要特点 50](#_Toc292873224)

[6.1. 本系统与拷贝机模式的比较 50](#_Toc292873225)

[6.2. 本系统与纯单播下载系统的比较 50](#_Toc292873226)

[6.3. 本系统的特点 52](#_Toc292873227)

[6.3.1. 文件分块机制 52](#_Toc292873228)

[6.3.2. 循环组播机制 53](#_Toc292873229)

[6.3.3. 负载平衡机制 53](#_Toc292873230)

[6.3.4. 组播和单播相结合 53](#_Toc292873231)

[第七章 结论与展望 55](#_Toc292873232)

[参考文献 56](#_Toc292873233)

[致谢 58](#_Toc292873234)

摘要

信息技术的发展正在改变现代企业的生产经营模式，日益成为企业得以生存、发展的最重要的因素之一。企业的现代化必定以信息化为基础，生产效率的提高必须以信息化为前提。

在企业信息化的步伐中，计算机网络的应用又是信息化中必将面对的领域。在日新月异的计算机科学理论和应用技术中，网络组播占有重要的地位。一方面组播理论不断发展，新的组播模型层出不穷；另一方面组播技术也被更加广泛地应用到社会生产、生活中去。

单播是与组播相对应的范畴，它同样是被广泛应用的技术。单播可以提供可靠，稳定的传输品质，被人们大量运用在日常的生产和生活中。

本论文研究网络数据传输领域的新成果，设计网络组播和单播相结合的模式，以现代笔记本电脑生产企业为现实应用背景，先后从现况分析、相关理论和技术的简介、需求分析、架构设计、详细设计和效能分析等主要方面，设计与实现一个结合组播与单播的高效数据下载系统。论文采用IP组播组模型，选用.NET开发平台，运用面向对象的C#语言实现论文所设计的系统。系统运行环境为Windows 2003 或以上版本和Windows Vista 或以上版本。

本论文所设计的新型数据下载系统通过联合使用网络组播和单播，以达到提高生产效率，减少设备投资，降低能源消耗的目的。该混合模式下的数据下载系统不但能体现比传统拷贝机生产模式自动化程度更高、性能更可靠的优点；而且具有比单一的单播数据下载系统更高效、更稳定、更可靠的优势。

本论文所涉及的理论和应用的技术，除了在笔记本电脑生产企业有实际的应用价值之外，在网络音频及视频点播、网络视频会议、多媒体远程教育、网络虚拟社区和网络游戏等其它网络数据传输领域也有广阔的应用和发展前景。

**关键字：**组播、单播、结合、高效、可靠

Abstract

The development of the information technology leads the change of the production in the modern companies and factories. The IT technology has been one of the main factors for surviving and developing the enterprises. The modernization of a company must be on the foundation of the informatization. The enhancement of the production efficiency must be with premise of the informatization too.

During the process of the informatization in the enterprise, the networking technology is the field which nobody can avoid. In the fast-developing networking theories and applications, the networking multicast owns its most important condition. The new theory and protocol of the multicast continue to appear. On the other hand multicast technology has been used more widely and popularly in the social production and life.

Unicast is the other important field in opposition to the multicast, it also spreads widely. It can support robust and stable data transmission, so it is also used popularly in the human production and life.

This thesis studies the new achievements in the field of network data transmission. It designs and constructs a new data transmission system with high performance on the basement of the mixed technology of the multicast and unicast through the procedure of present condition analysis, demand analysis, infrastructure design, module detailed design and performance evaluation. The thesis use IP multicast mode to develop the system by the C# language on the .NET platform. The system runs on the windows 2003 or above and the windows vista version or above.

The system which the thesis developed uses a mixed solution of the multicast and unicast in order to advance the performance and reduce the energy consumption. It not only has the advantage in the automation and reliability compared to the copy machine, but also has the superiority of the high performance and energy saving to the simple unicast data downloading system.

Both the system and technology in the thesis has the realistic value for the enterprise of the notebook computers production. Furthermore the theory and technology of the thesis can be applied widely and developed well in the applications of network audio-on-demand and the video-on-demand applications, network meeting, distant multimedia education, network virtual community, network games and so on.

**Key words:** multicast, unicast, mixed, performance, reliability

1. 引言

计算机信息科技的发展大力地推动现代制造业不断进步。尤其在笔记本电脑产品制造领域，以计算机网络为代表的信息科技更是持续推动企业向前发展的最主要、最持久的动力。

如今在笔记本电脑和相关产品的设计和制造产业中，企业对信息技术的依赖比已往任何时候都强烈。信息科技渗透在产品设计、硬件组装、功能测试、软件安装、品质检测等诸多环节内，并持续改善企业运营的各个环节。

以笔记本电脑制造企业为例，预安装出厂操作系统和应用软件是制造流程中最复杂的生产环节之一。在保护知识产权的意识不断提高、拒绝盗版的声浪不断高涨的时代背景下，笔记本电脑产品在出厂时均按照国家有关的法律法规，预安装正版的操作系统和应用软件。实体产品的软件含量在不断增加，也由于软件配置种类的千变万化，导致这道生产工序非常复杂。

在行业内，制造厂商需要采取计算机信息技术应对这一崭新的挑战。本论文结合工厂的实际应用，研究并运用计算机组播和单播的理论和技术，设计比传统生产模式或已有的单播下载系统更优的解决方案，即组播与单播相结合的下载系统，以提升生产制造的效率，降低硬件的投入成本，集约使用宝贵的硬件资源，减少因后续运行维护大量设备而造成的高能耗成本和人力资源成本。

* 1. 现有解决方案和系统的分析
     1. 拷贝机方案及其缺陷

硬盘拷贝机的生产模式是企业中传统的生产方式。它是指工人离线手动操作一种被称为硬盘复制机的设备，将带有出厂的操作系统的母盘复制到子盘中，再将子盘安装入笔记本电脑，实现产品价值在软件方面的增值。

这套模式的具体步骤是：技术人员最先把一套成品裸机所对应的操作系统、驱动程序、语言包和应用软件等安装到相同配置的物理主机上，将此台机器的硬盘作为母源发布到生产部门。生产线的操作工在后续生产中把组装完毕的笔记本电脑的硬盘从机器本体卸下，同母源盘一起放置到具有硬盘复制能力的硬盘拷贝机上进行从母盘到子盘的硬盘克隆。企业利用拷贝机将母源盘上的数据连同启动扇区、文件目录结构等信息一并复制到子盘上。

因为拷贝机生产模式是纯手动的操作，所以操作简单，对信息系统几乎没有要求。但是从生产自动化的角度分析，此流程具有大量的无效动作，比如拆、装硬盘，重复锁螺丝等，会造成工时的浪费。无效的工序非但不创造价值，而且造成产品机构零件因反复拆装而出现的损坏和外观的刮伤，产生作业性不良，甚至报废。

其次，如今操作系统的版本复杂，可能仅在同一个时间段内，生产线上就有需要灌装不同操作系统的机器（例如：Window XP、Window Vista和Window7中的一种或者两种系统）。Window Vista或是Window7又可以再往下细分成Home Version、Premium Version和Ultimate Version。驱动程序、应用软件更是品种、规格繁多。每一台出厂机器预安装的软件是操作系统、驱动程序、应用软件和语言包的组合，要把这些千变万化的组合固化在不同的母盘内，准确地复制到对应的产品上，仅依靠人为的管控，错误难以避免。

再次拷贝机在复制过程中容易产生数据错误。因为复制的数据对准确度要求高，如果在复制过程中遇到数据的丢失或者错误，就可能产生操作系统或者应用软件无法正常使用。以上情况一旦发生，该制程内无有效的手法进行检测，往往是产品到终端消费者手中后才被发现，造成消费者的投诉、退货。

总之，拷贝机流程作为传统的制造模式，工厂内虽然制订了一套标准作业流程以应对它的风险，但无显著成效。拷贝机基于人的手动操作，错误率高，检测率低，不良品反馈周期长的缺陷，在软件不断多样、复杂的趋势下，无法满足现实的需求。

* + 1. 纯单播下载系统及其缺陷

除拷贝机流程之外，某些企业开始探索使用计算机及其网络技术，解决软件预装流程中存在的难题。开发数据下载系统，使用网络下载的方式生产，正逐渐被企业接受。

现有的下载系统多是基于网络单播技术的解决方案。单播下载系统的基本架构是建立若干组服务器群，将软件包资源共享在这些服务器群内。客户端通过网络启动的方式，通过主控程序获得服务器上的资源，数据经由网络链路复制到产品机器的硬盘内。技术员无需维护大量的母源硬盘，只需将软件包按照一定的规则放置在服务器群中。生产线工人只需要将成品机器连上网络线，开机即可。后续的流程完全由程序控制，不需要人为的干预。

下载系统的生产模式没有拷贝机模式下的机构件卸装工序。机器的机构件和外观得到了最大程度的品质保证，避免了作业性不良的产生。当客户端数据下载完毕后，主控程序可以进行文件的正确性检查，如果发现文件有损坏，立刻重新自动建立连接，再一次下载。

单播技术成熟，而且纯单播模式的下载程序实现的复杂度低，开发难度小，易于维护。TCP单播模型是面向连接的网络模型，从客户端发起的每一个连接都作为一个平等请求，分享服务器和网络的资源。单播的稳定性高，数据在源节点和端节点间的网络传输过程中，文件不易发生损坏或者丢失。TCP/IP协议自身设计有校验机制和拥塞控制机制，网络设备间也有丰富的路由协议可以选择，可快捷地构建适应现实应用的网络平台。

但是纯单播系统的优点需要大规模硬件投入作为支撑。为了满足产能的需要，企业建设此类型的数据下载系统时需要购置大量的服务器，配套大面积的机房，配备大功率的空调以保证系统的运行。系统的其它服务设施，如消防和监控设备也必须和机房面积、服务器数量相匹配。在系统建成之后，运营系统及外围设施将会长期大量消耗电能，持续产生高昂的运营成本。总之，建设制造业级别的单播下载系统往往使企业背负沉重的成本压力。在提倡精益制造产品，集约使用资源，推崇可持续发展理念的今天，基于单播下载系统的生产模式的弊端暴露得很明显。

* 1. 新系统的解决方案及意义

传统的拷贝机模式无法满足制造企业日益复杂的生产需求。单播下载系统投资巨大，后期运行昂贵，不符合当今高效、绿色的可持续发展理念。想要突破现有模式的局限，实现短时间内大量数据的集中下载，又保证下载数据的完整和正确，使用网络组播和单播相结合的方式是最佳的选择，也是本论文研究和应用的方向。

本论文设计的基于组播和单播的新型数据下载系统，兼具网络层组播的快速和单播的可靠。系统辅助使用文件分块，负载均衡等机制，实现高效率，少投入，低能耗的适合笔记本电脑制造及更广泛应用的新型数据下载方案。

论文设计的思路是，首先在服务器端先将大容量的软件包分成小块，并建立索引。当组装完毕的笔记本电脑进入出厂软件安装环节时，首次运用网络组播模型，以文件块为单位，运用UDP协议下载数据。文件服务器和网络设备秉承“最大速率，极力转发”的原则，先将以块为单位的软件包用最大速度下载到机器本地硬盘中。客户机内运行组播下载客户端模块，接收并且恢复数据块为完整的大文件。

该方案考虑到UDP协议本身是一种非面向连接的传输协议，所以在下载大文件时会存在一定的概率的数据损坏，丢失的现象。所以客户端首次下载完毕后，需要进行文件的完整性检查。假如系统发现文件完整性或准确性已被破坏，客户端再次建立连接，通过单播的方式进行数据修补。

数据修补阶段采用单播传输模型，通过建立客户机与数据修补服务器建立的单播链路，采用单播模式进行数据传输。数据以TCP协议支持的数据流的形式，被送达到目的客户机。设计的数据修补进程不竞争实时生产的文件组播服务器的资源，保证首次下载业务依然以“最大速率，极力转发”的方式运作。

此混合模式的新数据传输系统，设计大文件首次传送用文件下载服务器对客户端机器的一对多的组播传输。即使有多台客户机请求，数据仍在文件服务器中被读取一次，服务器网卡也只发送一份数据，这样就可以减少服务器的数量。而在文件下载完毕之后，运用另一部分服务器进行文件的修补。新系统避免了大量的服务器作为硬件支持，也就降低了与服务器配套的机房面积，空调设备、消防和安全设备的需求。系统运营所需的电能消耗也大量降低。

将一个大文件划分成很多小的文件块，文件在应用层按照可定义的数据块为单位逐份发送，可以提高在UDP协议下网络传输的成功率。即使发生了文件块的损坏，客户端只需要重新下载对应的文件块，而不是整个文件全新下载，提高了生产效率。文件分块同时为客户端实现断点续传创造了条件，机器在传输过程中即使失去网络连接，客户端只要读取文件块的索引信息，在后续下载中就可以从最后一个已下载的文件块开始往后下载，而非自文件头处重新下载。

新系统中的负载均衡机制合理调度服务器资源，将相对轻负荷的服务器指派给客户机，实现系统资源的充分使用。

综合以上，本论文以组播与单播相结合的数据传输系统的设计及其性能优化为课题，实现既具备单播的稳定，又具备组播的高效的新型数据下载系统，在现实领域有很高的应用价值。

* 1. 本论文的篇章结构

本论文主要分为如下章节。

最先部分是论文的中英文摘要。

第一章为引言，主要介绍企业在预装操作系统和应用软件等软件资源时所面临的难点和问题。本章节内简介拷贝机生产方式和纯单播下载系统，并分析它们的不足。继而论文提出开发基于组播和单播相结合的新数据下载系统，满足企业的生产需要，以达到精益制造，节能减排的目的。

第二章为网络模型和理论，主要内容是本论文涉及的理论和技术的前导性知识。本章节在概述OSI模型的基础上引入TCP/IP模型，介绍TCP通信协议和UDP通信协议和它们各自的特点。阐述网络组播技术的现状及发展成果，重点介绍网络层组播概念、IP组播的常用模型和组播转发的相关知识。简介网络套接字编程的基础知识。

第三章为系统需求分析。结合笔记本电脑制造企业的生产流程，分析该行业的制造企业对于软件下载的功能性需求和非功能性需求。解析新型软件下载系统的用户构成，使用UML的用例图举例论述主要角色的系统需求。

第四章为系统架构设计，在前一章节的需求分析的基础上进行系统体系设计。它包括设计系统的总体架构与分解得出系统的各个模块。逐一论述系统各模块的功能，使用UML的协作图阐明系统各模块之间的相互联系。

第五章为核心模块的详细设计，选取系统关键模块，利用UML的方法细化流程，分解、整理、合并以得到各个模块的接口。在接口的基础上设计关键类。本章节运用UML的方法，采用类图表示核心类，并列举部分有代表性的编程代码。

第六章为系统的效能分析及主要特点。结合实验所得的数据，比较论文设计的新系统和拷贝机生产模式、单播下载系统之间的性能差异，彰显新系统的优势。论述本系统的主要特点。

第七章为结论与展望。总结论文，并阐述论文设计的系统的应用价值，展望本论文所涉及的理论和技术在将来的应用前景。

参考文献章节列举了论文形成过程中参考的中外文献资料。

致谢章节。

独创性声明。

1. 网络模型和理论

本论文设计的系统遵循OSI网络模型分层原则，在TCP/IP协议上使用.NET开发平台的C#语言实现编程。论文采用TCP和UDP的套接字编程技术和基于IP组播组模型以实现所设计的系统。

针对以上论文所牵涉的理论或者技术，本章节简介OSI模型，重点介绍它的网络层和传输层。通过介绍TCP和UDP不同特性，论述不同的下载方式选用不同协议的理由。最后介绍论文中所使用的IP组播组模型和网络套接字编程理论和技术。

* 1. OSI参考模型

计算机网络快速发展的首要前提是各设备、系统间使用标准的通信模式，遵循共同的通信协议。于是网络的OSI模型被提出，它规范了网络的通信标准，使得只要遵循OSI的标准，系统就可以与位于世界上任何地方、同样遵循同一标准的其它任何系统进行通信OSI参考模型共分为以下七层，各层次之间详细定义了各自所应提供的服务。自下而上分为物理层，数据链路层，网络层，传输层，会话层，表示层和应用层[18]。本论文的内容主要涉及网络层、传输层。

网络层是OSI参考模型中自下而上的第三层。网络层的主要功能为以分组为传送单位，通过路由选择算法为分组到达目的地址选择适当的路径，由此实现各子网乃至各网络间的互联。它还通过路由算法实现网络中拥塞控制。网络层使用一定逻辑地址集进行通讯，本论文设计的系统使用IPV4地址集。

传输层是网络层的上一层。传输层的主要功能是为用户提供可靠的端到端服务。传输层使用端口号的机制，实现同一台网络设备上可以多个应用程序进行网络间通讯。它是计算机通讯中关键的一层，向更高层屏蔽了下层数据通信的细节。传输层控制协议（TCP）和用户数据报文协议（UDP）都是属于传输层的协议，它们都使用端口号来标识应用程序。本论文设计的系统在网络传输层也使用UDP和TCP协议。

* 1. TCP/IP参考模型

OSI的模型相当于一个理论模型，在现实的网络技术发展状况中TCP/IP协议则是被广泛使用的网络模型。TCP/IP参考模型可以分为四个层次：应用层、传输层、互联层、主机-网络层。从实现功能的角度来看，TCP/IP参考模型的应用层于OSI的参考模型的应用层、表示层、会话层相对应；TCP/IP参考模型的传输层对应OSI参考模型的传输层对应；TCP/IP参考模型的主机-网络层则与OSI参考模型的数据链路层和物理层对应[18]。在TCP/IP协议族中，为传输层设计了两个协议：TCP和UDP。TCP是一种面向连接的传输层协议，而UDP则是一种无连接的传输层协议。

* + 1. TCP通信协议

TCP协议在网络层控制协议的基础上，向应用层用户进程提供可靠性、全双工的数据流传输。它允许两个应用进程之间建立一条传输连接，应用程序之间通过传输连接可以实现顺序，无差错、不重复和无报文丢失的流传输。

TCP协议是面向连接的服务，在进行通信之前必须在源进程和目的进程之间建立传输连接。只有在连接建立成功之后，两个通信才可以在该连接上发送和接受数据流。它将数据分割成一定长度的数据块，利用确认和超时重传的方式确保数据的可靠性。当两个应用进程使用TCP连接建立之后，双方可以同时发送和接收数据，接收的数据被缓存在接收缓存中，等待接收进程的读取，从而实现全双工的通讯。TCP协议提供一个流的接口，应用进程可以通过它发送连续的数据流，TCP协议对流的内容不做任何解释。同时TCP协议利用三次握手方法，在传输连接建立阶段，防止出现因失效的连接请求而造成的失败。在释放连接时，保证在关闭连接时已经发送的数据报可以正确地达到目的端口。它的滑动窗口机制进行流量控制，防止拥塞的形成。综合以上TCP协议的特点，论文中采用TCP作为数据修补模块的使用协议，使用C#语言在微软视窗平台上实现网络套接字编程。

* + 1. UDP通信协议

UDP协议为无连接的协议，它提供了一种不必建立连接就可将封装过的IP数据报发送到目的地的方法。UDP协议的特点是它面向无连接的协议，UDP协议是不可靠的传输协议，它在整个传输过程中提供有限的差错校验功能。程序在使用UDP协议通信时不必要使用连接。由于UDP协议的此特性可以很大地节省系统的开销，所以从数据传送的速度方面衡量，它比TCP协议要更快速，效率会更高。UDP协议不仅支持一对一的通信，而且支持对多的通信，在网络的通信中，组播技术的就是一对多通信的技术。综合以上特点，论文设计UDP协议作为组播下载模块的协议，使用C#语言实现微软操作系统平台上的网络套接字编程。

* 1. 网络组播

论文设计的系统应用于传输数据量巨大，速度和准确度要求高、硬件投资要求少的环境下。所以广播或者仅靠单播的数据传输方式都无法有效实现。在下载系统中设计和使用组播机制是能够解决以上问题一个优选方案。

组播是指一种允许一台主机（组播源或者发送端）一次同时发送单一份数据到多台主机（接收端）的技术[1]。现有的网络组播体系主要有两种模式，基于IP的组播和基于应用层的组播。在IP组播中又分为主机组模型和单源组播模型。论文设计的系统选用的组播模型属于网络层基于IP的主机组组播模型。

* + 1. 基于IP的组播模型

基于IP的组播模型[1]是成熟和被广泛使用的模型。其中基于主机组的组播模型利用网络层机制完成组播数据的复制或转发。主机组播模型是应用广泛的组播模型。

在网络体系结构中，网络层提供位于不同网段的主机之间的分组转发功能。在沿用此种经典的体系结构的框架下，IP组播体系结构首先被提出，即实现组播功能的机制设定在网络层。网络层设备（路由器、三层交换机或者其它具备三层转发功能的设备）通过一定的路由算法构建一颗组播转发树。分组数据沿着转发树进行转发的时，只有在树的分叉处，由三层网络设备进行复制。主机组模型还定义了主机和网络层设备在IP层的功能机制和上层所看到的组播业务的形式。该模型定义了组播组的概念，即凡是参加同一个组播组会话的所有的主机都加入一个以特定的IP地址为标识的组播组。这个特定的IP地址就是组播地址。数据以IP数据分组的方式被转发到组播地址，从而使得加入组播组的各台主机得到数据。主机组模型的组播兼容IP服务，采用UDP方式进行组播包的发送。组播组是动态、开放的集合，即任何的主机业务可以随时加入或者撤离组播组。理论上一个组播组可以拥有任意数量的源节点。在主机组模型中网络层设备的主要任务就是组播数据的路由和转发功能。通过特定的组播路由协议制定的转发树，在树的分叉结点上接收、复制和转发数据和使用组播协议知晓本组的新成员加入或者老成员退出的情况。

单源组播模型是另外一种组播模型[4,5]。采用单元组播模型可以很好的支持大量的、可以预见的组播应用。在单源组播模型中没有组播组的概念，而是通过发送者的单播地址与目的组的组播地址来定义组播信道[11,12,13]。主机通过向网络发送明确的订阅请求来申请接收信道的数据。在此模型中接收主机为订阅者，源节点通过向目的地址发送数据实现向信道内发送数据，且只有源节点可以向组播信道发送数据。

* + 1. 组播转发

组播在进行转发时要将同一信息发送到不同的接收端，为了节约网络带宽和减少复制信息的次数，应尽可能使得同一链路上的信息只复制一次。组播转发结构通常采用树型结构。组播包的复制在树的分叉处进行，使得分组复制数量达到最小。当某个分支上的所有活动都不再请求接收发的组播组的数据时，转发树将某个分支剪去。路由协议将重新计算转发树。常见的树型转发树包括：源树和共享树。

* + - 1. 源树

源树的根是组播组的源节点，各个枝干形成一颗覆盖网络中所有组播组成员的生成树。此树采用网络中的最短路径进行数据转发。基于源树的转发结构的好处在于不同的组播源发出的数据包被分散到各自分离的组播树上，有利于网络的负载均衡。还有从源节点到各组播组成员之间的路径是最优的，端到端的延时较好。源树的缺点在于它的扩展性能差。如果一个组播组内有多个发送端，那么就大系统的开销。

* + - 1. 共享树

共享树使用一个共同的根，位于网络的某个地点，成为汇集点。当组播源发送信息时，它先将信息发送到汇集点，然后由汇集点发送到各个组播成员。组播组内的所有发送源都使用这颗组播树。

* 1. 套接字概述

套接字是不同的计算机为了满足各自进程间的通信假设的一条数据通道，它是一个通讯链的句柄[22]。当应用层通过传输层进行数据传送时，为了区别不同应用程序的进程和连接，操作系统为应用程序与TCP/IP协议交互提供了称为套接字的接口，区分不同应用程序进程间的网络通讯和连接。套接字一般有单个主要参数：通讯目的地IP地址、使用的传输层协议（TCP/UDP）和使用的端口号。

本论文开发用的.NET平台有丰富的套接字资源。主要分为流式套接字、数据报文套接字和原始套接字。论文的第五章核心模块的详细设计采用流式套接字和数据报文套接字。

1. 系统需求分析

本论文设计的数据下载系统应用于笔记本电脑的出厂操作系统和软件下载环节。这道工序和前后的多个制程相关联，系统需求涉及多个业务部门。本章节以业务流程为脉络，使用UML的用例图，角色图等工具，梳理和分析不同用户对系统的需求。

* 1. 业务流程描述
     1. 笔记本电脑生产流程概要

笔记本电脑生产流程复杂，从主板上的芯片贴装到整机组装，再经整机测试到出厂软件下载，最后到外箱包装，总共经历几十道工序，制造流程繁复。

生产一台成品笔记本电脑从主板贴面工序开始。在此阶段工厂使用专用的大型贴片机将芯片组、电阻、电容等电子元器件，安装或者熔接到主板上。成品主板通过半制的测试后，进入成制阶段。成制有多道组装工序把笔记本电脑的主板、外壳、屏幕和其它组件（CPU、Memory、显卡、硬盘、DVD、无线网卡、蓝牙、调至解调器等等）装配成整体，锁合牢固。

组装完毕之后，笔记本电脑物理成型。此时厂内MES系统赋予机器一个序列号，该号码是机器在工厂内唯一的身份标识。之后笔记本电脑进入成制的测试阶段，测试分为前段测试、压力测试和后段测试。此阶段内有大量的程序模拟终端用户的使用行为，对机器进行大压力测试。续数小时的测试后，只有检测通过的整机才能进入下一道外观检查工序。

外观检查完毕之后，机器进入出厂软件的下载流程。厂内的软件下载系统首先对笔记本电脑的硬盘进行全盘清空，清除硬盘中原先存储的测试操作系统和测试程序。之后系统自动将操作系统，驱动程序、语言包，应用软件和补丁程序逐一下载入硬盘内。本论文所设计的新型软件下载系统即应用于这段制造流程。

下载流程完毕后，机器进入预安装阶段。机器重启后，本身的操作系统获得管理员权限，开始安装操作系统的各个程序，包括系统文件、驱动软件、应用程序、语言包。最后系统清除工厂制造使用的软件，进行自我封装。

当出厂软件安装工序完成之后，产品进入包装阶段。机器被黏贴上产品商标、各种标志。合格的产品在包装完毕后通过物流中心被分发到品牌商在世界各地的专卖店或者销售点。

因为品牌电脑出厂时必须安装完整的软件系统，所以以上的出厂软件下载和安装成为笔记本电脑产品工艺的必经工序。而且由于操作系统、语言包、应用软件和补丁程序的组合千变万化，这道工序发展到现今已经变得异常繁复，无法再靠人为的方式管控，只有借助信息系统的帮助才能完成。

* + 1. 出厂软件下载及安装工序

出厂软件下载及安装环节可被详细分解为如下图3-1所示。



图3-1 出厂软件下载及安装流程图

工人将机器被放入软件下载区域，并连上网线。机器网络启动后，申请得到动态IP地址。系统检查之前的各项制造流程，以确定前每一步流程都已正常完成。自查通过之后，机器重新启动进入微软的预安装环境(WINPE)。微软预安装环境是微软公司为各品牌电脑商和各大笔记本电脑制造企业开发的类操作系统。WINPE可以由网络或者硬盘启动，并将自身全部驻留内存中。此时系统开始格式化硬盘，彻底抹除之前硬盘内所有数据，包括分区信息，测试程序等。WINPE控制机器连接服务器单播下载一个最单纯微软视窗系统到本地硬盘内。这个操作系统即为笔记本电脑产品的操作系统的基底。此系统为视窗系统的核心，除了最必要的驱动程序（如网卡驱动等）和类库之外，无其它任何内容（如应用软件，非生产必须的驱动程序，语言包，补丁包，副操作系统等）。最核心操作系统之外的一切系统文件、软件都用组播模式下载，或有需要再采用单播下载修补数据。

当完整的出厂操作系统及所有附带软件下载完毕之后，机器重新启动，进入视窗操作系统的审核模式。在审核模式下，系统以管理员的身份自动安装的驱动程序、应用软件、语言包甚至是副操作系统等，安装结束后执行清理程序自动删除工厂内应用程序，清理注册表，实现OOBE（Out Of Box Experience）的终端用户首次开机体验。然后机器返回软件下载系统和工厂MES系统各一条PASS记录，并显示“PASS”界面提示生产线工人拔除网络线，将机器移转进入后续的包装流程。

* 1. 功能性需求

以生产流程为依据，分析系统功能性需求。主要的功能性需求可以主要分为业务数据维护、生产数据的查询、排序等使用和客户端组播下载、客户端单播下载等部分。

* + 1. 业务数据的日常维护

软件下载部门的工作人员要在系统内维护生产需要的软件包资源、软件包资源的网络路径、各类服务器地址、网络各网段等等信息。举例助理的用例图如图3-2所示：



图3-2助理用例图

* + 1. 生产数据的查询

生产的信息需要共享，系统将可以提供相关的业务单位所需的查询功能。包括产销部门、生产线部门、品质检验部门等等外部单位可以根据自身的业务需要从系统得到各自的工作所需的信息。系统对各外部单位查询的内容做区分和限制，使得使用者的行为受到合理约束，对系统的操作在一定时间内可以被记录与追溯。

* + 1. 组播下载

系统需要以最大的效能，实现大流量的数据下载。如图3-3所示：



图 3-3 软件下载用例图

当基本的最小操作系统下载完成后，机器重新启动，基本操作系统获得管理员权限。客户端通过提交自身的客户序列号（下简称SN）得到由MES系统反馈的PN码（机型码）。组播下载客户端根据该PN码，得到需要下载的各个软件包的网络相对路径（相对路径只包含文件夹及文件名信息，不包含服务器IP地址），再向负载均衡模组申请最合适的文件下载服务器的IP地址。

客户端把服务器IP地址和网络相对路径拼接成完整的网络路径，并开始从组播服务器下载数据。首次组播下载完毕之后，客户端开始进行自检下载的文件，以判断是否存在数据损坏或者丢失。客户机如果发现数据的完整性已破坏，则启动数据修补流程，修补数据。

* + 1. 单播下载

系统在WINPE的初始环境下，先要单播下载一个最小化的windows的操作系统。最小化的操作系统是指它本身除了windows运行的最基本系统文件、生产中需使用的驱动程序和系统下载客户端软件之外，不含其它内容。现有微软视窗的win7或者vista的操作系统，一般出厂的操作系统平均达到35G，但是去除非生产必要的内容后，最小化的基础操作系统的容量可以控制在7G左右。

WINPE下载完成后，机器重新启动，基本操作获得控制权。系统在最小操作系统运行之后，客户机首先开始组播下载，组播下载完成后大部分的数据已经存在于本地硬盘中。此时机器检查数据完整性，如果需要补包，则使用单播下载的方式把UDP协议丢失的或者损坏的数据块逐一修补完毕。

* + 1. 负载均衡

该系统内的服务器角色众多，特别是组播下载服务器和单播下载服务器的数量也非常庞大。大量的文件下载服务器运行一个网络平台上，就需要有负载均衡的机制，合理调度、调配各台服务器资源，用来杜绝系统中某些服务器由于高负载被拖垮，而同时又存在其它服务器空闲的情况发生。系统需要将大网络分为小网段，文件下载服务器按子网段配置，各个网段内的服务器优先承担和它相同网段的客户机的下载请求。在同一个网段内，仍需要有负载均衡机制，将同一网段内的组播下载服务器和单播下载服务器均衡调配，实现在子网段层级的负载均衡。无论客户机组播下载还是单播补包，在此之前都要由负载均衡机制告诉它们确定的服务器资源地址。

* 1. 系统角色
     1. 用户角色

下载系统用户分为内部用户、外部用户和下载用户三类。如图3-4所示：



图3-4用户角色分类

内部用户，包括计算机信息部门的经理、工程师、技术员和助理。如图3-5所示：



图3-5内部用户分类

部门经理、工程师为管理员，拥有对系统的最高权限。可以进行包括增加、删除用户，定义用户权限，更改系统配置，修改网络配置等一切操作。

技术员为高授权用户，拥有较高的系统操作权限。技术员可以对各角色服务器进行配置、修改服务器的参数、对数据库进行备份、配置网段等较高权限的操作。

助理为一般用户，可以使用界面实现对软件包资源的维护。实现数据的增加、修改、删除，和对历史数据的进行查询、排序等功能。

下载用户，是指需要下载软件包的笔记本电脑。下载用户属于访客用户，该类用户拥有合法的账号，可以申请系统的软件下载服务。它们被分配属于特定网段的IP地址，由负载均衡模块指定从特定的组播文件服务器或单播下载服务器下载数据。下载用户只拥有文件的读权限，它们下载数据的详细历史被记录在系统的数据库内。

外部用户，包括有该软件下载流程有关的其它业务部门。他们包括产销部门、生产部门、品质管理部门等。所有的外部用户人员均为访客用户，只能对系统进行和与自身业务相关的查询操作，只具备读权限。例如产销人员能查看软件包资源的准备状态，生产部门可以查看各条生产线的下载客户端数量等。

* + 1. 硬件角色

此系统的硬件角色按大项分为服务器、网络设备和客户端。其中服务器类由如下角色组成：

业务管理类服务器。此类服务器包Portal Server, Business Server, Load Balance Server, Migration Server。这类服务器用于业务管理。包括系统维护，业务逻辑的实现，负载均衡的实现，软件包在服务器之间的相互调度等。

文件下载类服务器。此类服务器包括组播下载服务器和单播下载服务器，它们提供软件资源的下载。

数据库类服务器。这类服务器运行数据库软件。将online生产的数据或者日志数据以数据库的方式进行管理。为工厂端的数据分析，生产日志查询等等应用提供数据库的平台。

网域类服务器。这里服务器包括了域服务器、DHCP服务器、病毒更新服务器等。主要承担域环境下的服务，使得下载系统的域的环境正常运行，提供动态IP地址的发布服务，并实时地向更新补丁、病毒防护库保证网络的安全。

网络类设备主要是具备三层路由转发功能的网络交换机。客户端机器，即申请文件下载服务的客户端笔记本电脑。

* 1. 非功能性需求

系统除了功能性的需求，系统另有非功能性方面的要求。非功能的需求主要体现在降低能耗，稳定可靠，维护简单，易于升级、安全茁壮等几个方面。

* + 1. 低成本、低能耗的需求

软件下载的制造流程产生的数据流量巨大且持续，但是现实中企业由于成本的考虑，不允许将系统建设成为一个由硬件堆砌的粗放型系统。工厂端受限于机房面积、资金压力、人力资源等诸多方面的限制，设备的数量得到合理控制，各台设备的负载必须均衡，各项资源必须合理、集约的使用。

硬件资源的合理使用，不仅是硬件型号和数量的合理选择，减少系统在建设之初的投入。它还体现在后续的运维成本上，即系统对电能的消耗，系统外围设备（空调设备、监控设备）的能源消耗和维护该系统的人力成本等方面。总之系统对硬件的需求应合理。系统针对特定业务进行设计，将服务器等硬件资源合理规划成不同的角色，实现不同的任务，尽力提高各台设备的使用率。例如，针对软件下载的业务，新系统设计将文件服务器设计成组播服务器和单播服务器两种种类，采用组播实现首次文件下载，后再进行单播数据修补，最大程度地减少服务器的使用数量。同时针对不同角色的服务器在选型上差异对待，控制设备的硬件规格，做到够用又预留一定的性能空间，方便后续的系统升级，但是不过度投资、粗放型设计。

硬件资源要实现负载的均衡，不能出现某些设备被高负荷拖垮，某些设备使用率过低的冷热不均的现象。所以因考虑引入负载均衡机制，在它的调度下完成生产业务。例如，在系统中客户端的任何资源下载请求，先向负载均衡模块注册，等待分配资源，避免因系统负载不均而导致的性能降低，乃至崩溃的风险。

* + 1. 可靠性和茁壮度的需求

系统的稳定性在制造业的需求中至关重要。因为系统运行出现的微小抖动，可能严重影响正常的生产顺序，甚至会造成产品质量出现瑕疵、拖延产品交期等公司业务层面的影响。

系统的关键节点需要冗余设计。当故障出现时，故障点可以在最短的时间被切换到备份资源，无缝提供服务，不干扰企业正常生产。稳定性的另一层更高要求是一旦出现问题，系统有能力将故障限制一个尽可能小的范围内，在工作人员排错并修复故障之前，系统的其它部分不受到干扰，还能正常发挥功能。

所以系统内的关键资源需要有多份备份。网络平台需要具备一定的容灾能力。

* + 1. 方便重构、易于升级的需求

制造企业中由于订单的改变，系统的调整或者二次开发很普遍。优秀的制造用系统模块必须科学合理，模块之间松散耦合，模块内部强内聚。

本论文设计的新系统，模块按照业务划分而非按照使用者划分。因为使用者角色经常变动，而业务通常稳定，模块参照业务最大程度地提供了系统弹性。按照面向对象的程序开发模型，模块间定义清晰的接口。使用接口机制，即使模块被更新或者调换，如果对外接口不变，系统依然正常运行。模块的功能需紧凑，各模块对应特定的某一功能，由类实现模块的功能。类的属性或者方法按照公共的，私有的，受保护的等加以区分。

同时系统需要将不同的业务分层体现。用户界面、商业逻辑、数据库需在系统中使用分层设计。系统需要保持对用户界面的风格的统一。即使后续升级界面风格需要延续。业务逻辑部分能随着产品的更新而弹性调整，数据库需要方便迁移、备份等其它应用。

* + 1. 安全的需求

对用户账号分层，区分用户权限的等级。系统赋予用户的权限够用即可。对用户的行为进行记录，当出现系统警讯时，用户的在系统内的操作可以被追溯。

在数据资料方面，需要将数据的安全性分等级。用于生产的资料的安全性要求最高，历史数据资料的安全性次之。

在系统防护方面，系统需要有专属的进行病毒防范，系统补丁更新的服务器。该类服务器可以和系统内负载业务的服务分开建制。系统内各类角色服务器都需要实时打补丁和定时更新病毒库文件。

1. 系统架构设计

参照需求分析的结论，本章节从系统架构层面提出解决问题的思路，设计系统总体架构并且论述主要模块的功能和阐述它们之间的协作关系。

* 1. 系统架构设计所面临的问题

前章节的需求分析表明此项数据下载系统属于生产用的系统，牵涉的部门众多，系统的伸展面宽广。要设计出符合实际生产需求的、稳定高效的数据下载系统面临不少难题。首先将跨部门的用户界面，业务逻辑、后台数据整合在一个系统中，同时又需要把各模块间的耦合程度降到最低，最大限度地方便后续因为产品的更新而产生的系统升级。

其次，将庞大的数据在短时间内下载入客户机，同时又要保证数据的准确无误。而且此系统下载的不仅是普通数据，更多是包含了对准确度要求极高的操作系统文件、设备驱动程序、应用软件安装包等等。如果发生了数据的损坏或者丢失，轻则直接导致产品机器工作不正常，重则产品进入终端用户手中后开机即出现蓝底白字无法进入系统，造成客户投诉和赔偿。

再次，在需要保证效率的同时，又要达到系统的茁壮。因为本系统属于企业的生产用系统，如果发生系统运行不稳定或者出现更严重的系统层级的故障，结果将是企业无法正常运营。产生无法按时完成产品交期，甚至影响到企业和它的客户在商业层面合作的失败，危及公司的生存。

* 1. 系统设计的思路

针对以上几点困难，本论文设计的系统设计思路为：采用三层的模型将系统的设计为表示层，业务逻辑层，数据库分开而又联系的分层结构。这样的设计既保证了系统具有最大的包容度和弹性，又可以满足模块的重构和组件的扩充等后续的升级

因为TCP/IP协议族是被广泛应用的网络协议，也是对OSI开发标准支持的比较好的协议族，其普遍性和成熟度都很高。所以系统采用网络层组播作为组播的模型，采用UDP协议作为网络层组播下载的协议。设计文件分块，在应用层用文件块作为UDP的传送单位；选用单播进行数据修补，用TCP协议作为单播的协议。网络采用三层架构，构建系统运行平台。使用OSPF协议作为动态路由协议，用PIM模型作为网络层设备的组播协议。各个网段之间独立而又有联系，某个网段的故障不会对其它网段产生不利的影响。

* 1. 系统软件架构的设计

在需求分析成果的基础上，设计系统软件架构如下图所示：



图4-1系统软件架构图

* 1. 系统内服务器角色的设计

Portal Server，它属于表示层服务器。用户可以通过基于浏览器的界面，使用或者维护整套系统。日常工作的内容包括维护各项软件包的名称、相对路径、软件包的状态信息、版本信息、系统的网段信息及各个网段对应的文件组播下载服务器的名称和IP地址。Portal Server可以开放权限给其它部门，例如质量检测部门、、产销部门、各生产单位等等，使得这些部门可以通过访问网页的方式，获取他们所需要的信息。

Business Server，该服务器上部署所有系统业务管理相关的业务层模块。它通过和Portal Server之间的service的连接，获取用户对系统的操作命令或者其它信息。相对于Portal Server的门户或者显示的功能，Business Server以非前台的方式运行。

Database Server，该类服务器部署业务数据库。设计中将业务层的服务器和数据库服务器分开，目的是为了增加系统的可维护性。又考虑到业务数据库和日志数据库存储的数据的差别，将两种不同性质的数据库分别部署在不同的服务器上，方便后续的数据库的备份和迁移，使得系统更加易于维护和更增加茁壮度。

Migration Server，该服务上部署对软件包同步的有关模块。服务器通过计划任务或者特定策略同步每台文件服务器上的软件包资源。依据第三章的业务流程，客户机在进入软件下载流程之前需要历经多项工序，耗时很长。Migration 服务可以利用这段时间，检查客户机所对应的各台文件下载服务器是否拥有客户端在下载所需要的资源。假使部分服务器上没有客户机所需要的软件包，Migration 服务可以利用前期工序的时间，将之后会被使用的软件包同步到文件服务器上。它还可以定期轮询各台文件下载服务器，删除长期不用的文件或者已被新版本替代的软件包文件，以保留足够的服务器硬盘空间。

Load Balance Server，该服务器上运行有负载均衡模块，它的功能非常重要。负载均衡服务器负责将负荷最轻的文件组播服务器或者数据修补服务器指派给客户端。负载均衡策略综合处理文件服务器网段信息、客户机所处的网段信息、每台服务器上正在运行的组播通道数量、服务器可用的网络带宽等数据，将最合适的服务器提供给客户机。该模块保证了在系统中各种服务器资源能被均衡利用。

File Multicast Download Server，组播下载文件服务器储存各类软件包资源，运行网络组播下载服务器端模块。它提供客户机的首次下载，完成大部分的数据下载任务，自始自终秉承“最大速率，极力转发”的策略向下派发软件包文件。在工作中它保持和Load Balance Server的实时交互，把自身的状态信息定期提供给负载均衡服务器。

Data Fixed Server，数据修补服务器上运行数据修补服务器端模块，在接到客户端的请求后，其内部形成服务队列。按照队列顺序向客户端重新发送特定的数据块文件。数据修补服务器需要配置高规格的RAID卡和快转速的硬盘和充足的网卡带宽，它需要保证数据的稳定和高品质的下载。

Log System Server，日志服务器上运行下载历史数据库，记录客户机下载产生的各项数据。此类数据量庞大且增长飞快，所以设计日志数据库独立运行在专门的服务器上，更有利于数据存档、数据挖掘等深入的应用，同时也更方便日常维护。

Domain Servers，域功能服务器类，此类服务器与业务无关。它们负责整个下载系统网域的维护和管理。DHCP Server用于动态分派、管理包括客户端在内的整个网络的IP地址。补丁服务器负责系统补丁的自动派发和安装。病毒防御服务器负责防病毒软件的及时更新。

* 1. 系统模块的功能设计

从软件模块的视角设计系统的各个模块，按功能的分类分为前台模块、中间层模块、后台数据库。前台模块负责对用户提供友好的显示界面，方便各个层面的用户使用系统。中间层业务模块负责业务逻辑的处理和流程的处理，关注在业务逻辑的合理高效。后台数据库存储数据，使用数据库的方式管理生产的实时数据和历史数据。

* + 1. 前台模块群

服务器控制台

由于在该系统中，可以使用多台属于不同的网段的文件下载服务器。该模块提供文件服务器的注册功能，将文件服务器配置到每个网段中，便于负载均衡模块的管理和调度。文件服务器的IP地址和软件包文件的相对路径一起形成一条完整的路径。

业务管理控制台

文件管理模块负责提供软件包的可视化的管理。模块主要包括了软件包管理模块、日志模块等。该模块的主要功能是提供下载内容的可视化管理。主要为工作人员提供可视化的操作，进行软件包资源的建立、删除、查询、排序等。

当工厂端得到软件包之后，工作人员把软件包同步到文件服务器的特定目录中。部门内的助理按照EDI或BOM将笔记本电脑的PN（产品型号）和软件包在服务器上的目录一一对应建立关联，使得特定的PN可以下载对应的软件包。

软件包的维护关系是一对多的关系，一个PN号码可以对应特定的多条软件包下载路径。在生产线的实际作业中，特定PN的笔记本电脑按照软件包管理模块中所指定的路径逐一下载软件包。

电子看板模块

电子看板模块将正在下载的客户端的状态集中显示到界面中。软件下载流程重开始到结束时长2至4个小时，整个过程细化为很多小的阶段。电子看板模块显示机器所处的任何一个阶段的状态。客户机下载Pass或者Fail将显示在屏幕上。如果下载失败，则将机器下载失败产生的概要信息显示在屏幕中，使生产人员对整个软件下载情况一目了然。

日志显示模块

日志操作模块由众多用户使用的界面组成。操作人员可以通过日志模块通过中间的业务层模块，显示后台的日志数据库的日志数据。模块向操作者提供友好的人机界面。操作包括简单查询或者符合条件的查询、排序、新增记录或者删除记录等等。

* + 1. 中间层业务逻辑模块群

安全认证模块。安全认证模块负责用户的增加、删除、权限、行为审计等功能。当有客户端连接入系统请求下载服务时，需要先通过用户名和密码的认证。同时该模块除了认证之外，还对不同的用户的行为做出不同权限限制。以及对用户的行为进行记录和审计。

业务管理模块。业务管理模块是一系列处理业务流程的单元的总和。此模块以业务逻辑层的形式承担业务流程的处理。它需要接收显示层转达的用户的操作请求，经自身的业务逻辑处理后和业务数据库或者日志数据库交互信息。模块内部根据不同的业务再做细分，形成颗粒度更小的单元。

日志操作模块。该模块负责提供日志显示模块所提供的一切操作的后台作业。它以业务逻辑层的形式，运行在后台，真正负责向日志数据库的读操作，如查询，检索、排序数据全部操作。它和日志数据库交换产生的结果，被反馈到日志显示界面。

资源同步模块。资源同步模块主要根据厂内的MES系统的排程，按照一定的优先顺序将必要的软件包同步到各台文件组播服务器和数据修补服务器上。它也可以通过工作人员设定的同步计划同步软件包文件。

文件派发模块。文件派发模块使用UDP协议向客户端发送数据。

数据修补模块。数据修补模块在UDP下载之后，在接收到客户端的文件块修补请求后，通过TCP协议向客户端发送数据。

负载均衡模块。由于整个网络内包含多个网段、每个网段内又存在多台文件服务器可以提供上万种软件包的下载服务。负载均衡模块就显的尤为重要。该模块主要负责服务器资源的合理分配和调度。负载均衡模块采用负载均衡算法，将拥有目标软件包的服务器归为一组，综合考虑某一时间内的服务器组播组数量，下载软件包的种类等信息，为客户端指定最优的网络路径，获取下载资源。如果无任何一台服务器正在提供客户端所需要的下载软件包，优先考虑组播组少的服务器作为候选者。如果有多台服务器正在提供该软件下载，并且软件组播数量等各项指标均相同，又都在同网段内，则随机挑选一台服务器作为下载源服务器。该模块还负责向客户端指派最优的数据修补服务器。当得到客户机的文件块修补请求时，根据负载均衡算法得出最佳的数据修补服务器，将数据修补服务器的路径传送到客户端。

下载客户端模块。客户端下载模块负责将软件包资源下载到本地。并在文件完整性检查后，将可能丢失或者损坏的文件块额外请求并下载完整。

日志上传模块。日志上传模块将客户端信息上传到日志数据库服务器。

* + 1. 后台数据库

业务数据库。业务数据库保存和产生业务相关的数据库。该数据是现时间段内的业务数据。现时间段内所用到业务数据包含了各类软件包的状态信息，实时每条生产线的下载机器的机型及对应的数量和状态。业务数据库要求高稳定性和高速度。特别在出现数据库故障时需要最快速的恢复。但是它对容量的要求没有日志数据库那样庞大。

日志数据库。日志数据库保存每一台机器的生产的全过程产生的日志。日志数据库模块和业务数据分开。使得日志数据库中庞大的历史记录可以被独立操作、备份或者移转。此设计目的在于排除在操作大量数据时对于业务数据库的可能的干扰，保证企业正常的生产秩序，增加系统整体的茁壮性。

* 1. 各主要模块间的协作关系

设计系统各模块之间协作关系如图4-2，各模块的主要协作关系说明如下。



图4-2系统模块主要协作关系图

* + 1. 安全认证模块与相关模块的关系

用户登入首先需要经过安全认证模块的认证，安全认证模块认证通过之后，系统才根据用户的输入转向到电子看板模块、日志模块、业务管理模块或者服务器控制台等模块。

* + 1. 服务器控制台和相关模块的关系

服务器控制台和文件组播下载模块

服务器控制台模块将对文件组播下载模块的管理和设定信息发送到文件组播下载模块。管理和设定的信息包含了，每个文件组播下载模块的最大下载速度、最大组播组数目，单组播组等待时间，客户端最大等待时间等等。文件组播下载模块收到此类指令后，进行对应的参数调整。

服务器控制台和数据修补模块

服务器控制台传递参数设定指令到数据修补模块。该设定参数包括客户服务队列的长度、最大单播连接数量等等。

* + 1. 业务管理控制台和相关模块的关系

业务管理控制台、业务管理模块和业务数据库的关系。业务管理模块传达操作人员的操作命令（写入、查询、排序等）到业务数据库。业务管理模块作为中间层负责将用户的数据库命令作用到业务数据库。业务数据库和业务管理控制台之间无直接关联。

业务管理控制台和业务管理模块和软件包同步模块的关系。业务管理控制台将对软件包的同步策略通过业务管理管理模块传递到软件包同步模块。业务管理控制台和软件包同步模块没有直接关联，业务管理模块作为它们的之间层。

* + 1. 日志类各模块的关系

日志模块、日志查询模块和日志数据库的协作关系。日志模块传递接收操作人员的操作命令，并将指令传递到日志查询模块，由日志查询模块完成对日志数据库的操作。日志数据库和日志模块之间无关联，日志查询模块为它们的中间层。

日志上传模块与日志监听的协作关系。下载客户机将日志信息，通过日志上传模块上传到日志监听模块，由日志监听模块写入日志数据库。日志监听模块为日志上传模块和日志数据库的中间层。

* + 1. 负载均衡模块和相关模块的关系

负载均衡模块和客户端模块、文件组播下载服务器端模块和数据修补服务端模块之间有密切的关系。首先客户端下载模块将PN发送到业务管理模块，随后得到对方反馈的软件包的相对路径（不含服务器IP地址）。之后客户端将相对路径传递到负载均衡模块，负载均衡模块取得文件下载服务模块的状态，通过计算或者遵照一定的策略返回给下载客户端模块一条同时带有文件组播服务器IP地址和文件相对路径信息的完整路径。客户端在得到该路径后向对应的文件组播服务器请求文件下载服务。

当文件下载完毕，需要数据修补操作时，客户端将所需要的数据块信息发送到负载均衡模块，负载均衡模块取得各数据修补服务器的状态后反馈一台特定的数据修补服务器的IP地址到下载客户端。客户端向特定的数据修补服务器请求数据块文件。

* 1. 系统网络架构的设计

在整个下载系统中，系统的网络设计举足轻重。由于该系统需要支持大型企业的大规模的生产，网络平台的平稳运行是基础。该新系统的网络设计采用三层结构。将一个大生产型的网络分成很多小的网段。每个小网段为二百个左右的接入点。

设计以OSPF协议为基础的三层架构，增强网络的易维护性和茁壮性。如果某网络出现异常，也将异常控制在单独的网段内，方便工作人员的故障排除。单个的网段出现异常不会拖垮整个网络，不会对在其它网段下载的客户机造成影响。

服务器在网络拓扑结构上直接连接网络交换机。每个独立网段内拥有各自的文件下载服务器。采用这样的设计可以最大限度地利用交换机的背板带宽，避免在网段间的链路上出现大流量数据形成网络拥堵。同时为了适应大数据量传输的应用，该网络架构可以在最大程度地支持二层数据组播的前提下，同时支持三层组播，效率最高。

* + 1. 网络层的组播实现

论文设计采用IP组播主机组模型。网络中的三层设备利用路由算法构造组播转发树。系统使用IGMP协议及时侦听是否存在某组播组的任意成员。IGMP协议使路由器或者三层交换机可以及时加入特定的组播组，并将数据报文转发给加入该组的客户端。由于现实的应用环境是客户端数量多且集中，服务器数量相对少，达到近60：1的比率。所以系统在数据传输中优先使用二层组播转发，然后再使用三层的组播转发，以提高数据传送的效率。

* + 1. 网络层的单播实现

根据系统需求分析的成果，设计网络层使用OSPF协议实现单播的应用。将整体网络划分为多个区域，彼此相邻。OSPF作为支持动态路由的协议，可以短时间内计算出最佳路由。它可以在链路故障时自动启用备份链路，并将新路由的收敛时间大幅缩短。网络内路由的变动也可以通过毗邻的节点及时更新，使整个网络运行得更加平稳。

1. 核心模块的详细设计

本章节详细论述系统的核心模块。详细论述各模块的功能，设计模块的工作流程，并在此基础上设计接口。再由接口设计出模块需要的各主要类，并用UML类图清晰描述主要类。使用C#丰富的套接字类[22,23,24,25]和自此基础上的网络编程技术，实现并列举使用C#在.NET开发平台上的核心模块的部分代码。举例系统在Windows2003平台上和Windows7环境中运行的部分人机界面。

* 1. 文件分块模块的详细设计
     1. 模块详细功能

设计文件分块模块，该模块进行文件处理，负责将大的软件包分割成小块的，有索引的数据块文件。文件分割模块打开大文件，按照定义的大小，将整体大文件分解，利用文件流的方式逐一写到多个小文件中，并对每一个小文件标记索引号。

模块工作时，首先读出大文件的容量，通过总的容量除以单位容量，计算出小文件块的文件数量。在系统中预生成这些数量的小文件，然后用流的方式读取大文件的内容并且往小文件内写入。在写入完毕后，该模块生成CRC校验码，并标记文件块的序号。在文件分解结束之后，将原先的大文件删除，或可以通过调用其它的进程把源文件备份到别处。

* + 1. 流程设计

图5-1 文件分块流程图

* + 1. 接口设计

按流程图整理、设计接口如下：

IOperateFile接口，接口定义的方法有：

GetFileSize方法得到目标大文件的大小，单位为byte。

ReadFile方法按照设计的标准单位，逐步读取大文件的内容，并缓存在Buffer数组中。

WriteFile方法将Buffer的内容写入小文件，形成小数据块文件。

ITag接口用于给每个数据块文件打标签，用于后续索引或者后续的文件完整性检查。

ICreateCRC接口对于每个数据块文件产生CRC校验码，用于后续的校验。

IFail接口处理过程所生产的错误或者异常。

* + 1. 类的设计

设计SplitFile类

类的属性

Buffer：单个数据块的大小。

类的方法

GetFileSize：得到文件的容量大小。

ReadFile：按照Buffer大小读取目标文件。

WriteFile：将Buffer的内容，写入进新的文件，形成数据块文件。

CreateCRC：产生CRC校验码

Tag:对数据文件块标记索引号。

模块类图：



图5-2 文件分块类类图

* + 1. 接口或者类的实现

文件分块类的部分主要代码如下，用C#语言实现将预先定义大小的数据写入某个小文件，成为一个数据块文件：

try

{

aSplitFile = new SplitFile(filepath,

FileMode.Open,FileAccess.Read,FileShare.Read);

data = aSplitFile.Length;

length = aSplitFile.Read(buffer, 0);

aSplitFile.OutputStream.Write(buffer, 0, length);

}

catch (Exception ex)

{

Response.Write("Error : " + ex.Message);

}

finally

{

aSplitFile.close;

}

}

* 1. 组播下载服务器模块的详细设计
     1. 模块详细功能

该模块为文件组播发送服务器端程序，运行在文件组播下载服务器端。该模块设计采用UDP协议作为传输协议，将预处理后的数据块文件，使用网络层组播的方式在UDP协议的基础上进行文件传输。主要遵循“快速尽力发送”的原则，最大可能地提高传送效率。

由于传送的软件包容量很大，平均可以达到25G，包括了语言包、应用程序的安装包、驱动程序、高清视频等内容。在UDP协议的非面向连接的特性下，数据并非以流的形式传送，而是用报文的形式传输存在一定概率的数据丢失或者损坏。而在下载过程中出现任何的错误都会造成整个出货操作系统的报废，需要重新下载。为了降低传输失败导致重新下载的概率，设计该模块用数据文件块为传输单位。第一通过对文件的分割，使得大文件化整为零，单个小文件出现数据报文传输损坏的概率更小。其次即使出现文件块的损坏或者丢失，也可以通过后续的修补流程，修补对应的文件块，而不是整个文件全新下载一次。

* + 1. 流程设计



图5-3组播下载服务器端流程图

* + 1. 接口的设计

分析流程图，整理并设计接口如下：

IReceiveImagePath接口用来侦听并接收客户端发送到该模块的软件包申请信息。资源信息的形式是一条包含服务器IP地址的绝对路径，以字符串的形式传递到文件组播发送模块。

ICreateMultiCastGroup接口提供由文件组播服务器模块的生成组播组。如果客户端申请的软件包在已有的组播组内，模块将不重复开启组播组。客户端下载模块会加入进已有的组播组。

IOperate接口处理文件，其中定义的ReadFile方法读取需处理的文件内容。

IUdpSend接口将读取的数据内容通过文件组播服务器端模块用UDP协议往目的组播组地址中发送数据。

ICloseMultiCastGroup接口当组播组内所有客户端退出后，负责关闭组播组，释放系统资源。

IFail接口被设计为一个异常处理模块，当发生报错时，统一由该接口处理。

* + 1. 类的设计

在以上设计接口的基础上，设计模块包含的类，和实现接口的方法。

类：UdpSend

该类负责网络组播传输。它的属性包含有：

EnableBoardCast 设定是否可以泛洪或者广播

Speed:设定默认下载速度

MaxClient:设定最大单组播组内客户端的数量，限制过多的机器拥挤在一个组播组内，影响组播组的数据传输质量。通过设定此数值限定单个组播组的客户端数量上限。当单个组播组内客户端数量的达到上限时，此组播组拒绝新客户端的加入。

WaittingTime：等待时间的设计，目的是为了客户端等待的时间过久。因为存在某个组播组的客户端数量一直未满，导致组播通道无法开启服务，造成客户端机器长时间等待而无法下载数据。设计此属性后，机器的等待时间会有一个最上限，一旦到达这个上限，无论通道的机器是否上满，立刻开始下载。

类的方法含有：

CreateMultiGroup：新建立组播组。当开启一个新的软件包下载时，创立组播组。

UdpSend方法，使用UDP协议发送数据块。

CloseMultiGroup:关闭组播组。当组播组内所有的机器都完成下载之后，关闭组播组。

类：OperateFile：

该类的功能是处理文件，它的方法包含有：

FileExist方法判断文件是否存在。

OpenFile方法打开文件，获得文件的操作句柄。

ReadFile方法获取文件的内容。

CloseFile方法关闭文件。

模块类图



图5-4 UdpServer模块类图

* + 1. 接口或者类的实现

UDP Server类的实现代码如下，用C#语言实现UDP Server模块绑定一个组播地址和端口号，进行UDP协议的数据传输。

public partial class UdpServer

{

protected bool EnableBroadcast;

protected int WaittingTime;

}

protected int UdpSend (IPAddress ipAdd, int port)

{

if (DataByte == null)

{

MessageBox.Show"SendData Error";

return (-1);

}

udpSend = new UdpServer();

udpSend.EnableBroadcast = true;

IPEndPoint iep = new IPEndPoint(ipAdd,port);

try

{

udpSend.Send(byte,bytes.length,iep);

}

catch (Expception e)

{

MessageBox.Show(Send Data Error);

}

finally

{

udpSend.close();

}

}

* 1. 数据修补模块的详细设计
     1. 模块详细功能

由于UDP协议的非面向连接性，所以客户端存在出现文件块丢失或者损坏的可能。当出现文件丢失或损坏需要重新传送时，客户端先向负载均衡组播模块提出文件块的重传申请。负载均衡模块收到申请后，为客户端提供一台合适的数据修补服务器。客户端获得数据修补服务器的地址后，向该台数据修补服务器提出文件块修补请求。

数据修补模块在收到数据修补的请求后和客户端建立通信，开启等待进程。这样的目的在于监听是否有其它客户端申请同样的文件块。如果在特定时间范围内，有其它客户端申请同一文件块，数据修补模块将目的文件块复制对应的若干份，最后一并通过TCP协议以文件流的形式发送往目的地客户端。通过这样的设计可以使得数据修补服务器不必反复读取硬盘，可以最大限度的使用内存；也利用了数据在内存中复制的速度远远高于反复对硬盘的读写，从而提高系统的效率。

数据修补模块中设计队列机制，当出现大量的客户端请求，而同时被请求的数据块文件又比较分散时，可以利用先到先服务的原则，分批次提供服务，避免数据修补服务器被大量的客户端拖垮。

设计数据修补模块和文件组播下载模块部署在不同的物理服务器上，避免UDP协议和TCP协议相互竞争服务器硬件资源的情况，所以在数据块文件的修补过程中不会影响到UDP方式的大文件下载，整体上保证了系统的效率。

设计数据修补模块使用TCP协议连接服务器端和客户端，利用TCP协议自身的面向连接的特性，可以大幅提高数据块文件传送的准确率，达到无差错传送的数据传送。

* + 1. 流程的设计

图5-5 数据修补流程图

* + 1. 接口的设计

根据以上泳道图分解、整理得到如下接口：

ITcpListen接口，用以监听Tcp的连接请求，当出现网络中对应的TCP连接请求时，调用TcpConnect方法建立连接。

ITcpLink接口，接口包含：

TcpConnect方法建立并保持TCP连接。CloseConnecting方法在转发块结束后，关闭TCP连接，释放资源。

ITcpSend接口负责使用TCP协议传输数据块。

复用IOperateFile接口处理文件。

IFail接口处理整个过程中的任何报错信息。

* + 1. 类的设计

参照接口的设计，设计如下类

TcpServer类

该类负载网络单播传输，它的属性有：

Speed属性定义TcpServerSend的文件传输速率。由于服务器可以将多块网卡绑定在一个IP地址增加带宽或者服务器自带万兆的网卡，此项属性可以设定服务器网卡使用特定的速度传输。

MaxConnecting属性定义同一时间最大连接的客户端数量。

DataPort属性定义端口号。

方法：

SetUpConnecting方法建立TCP连接。

TcpListen方法进行侦听，确定是否有客户端需要连接。

TcpSend方法使用TCP协议进行数据的传输。

TcpReceive方法使用TCP协议进行信息的接收。

CloseConnecting方法在文件块传送结束后关闭连接

OperateFile类

模块需要对文件操作，所以设计复用之前设计的OperateFile类。

模块类图

图5-6 TcpServer模块类图

* + 1. 接口或者类的实现

TcpServer模块代码的实现，使用C#语言实现网络套接字编程。代码实现绑定IP4地址和端口，使用TCP协议进行数据传输。

protected int TcpSend(IPAddress ipAdd,int remotePort, int port)

{

byte[]sendByte=null;

IPEndPoint remoteIpEndPoint = new IPEndPoint(ipAdd,remotePort);

tcpSend = new TcpServer(remoteIpEndPoint);

IPEndPoint iep = new IPEndPoint(ipAdd,port);

while(true)

{

try

{

sendByte = TcpSend.Send(ref iep);

}

catch (Exception e)

{

MessageBox.Show("Send Data Error");

log.info("Response uut: " + errorInfo + ". Error Information " + TcpServerIP);

return (-1);

}

finally

{

tcpSend.Close();

}

}

* 1. 下载客户端模块的详细设计
     1. 模块详细功能

客户端下载模块负责通过UDP协议向文件服务器下载软件包，之后通过文件检查机制判断是否存在数据块文件的丢失或者损坏。当出现文件块丢失或者损坏时，客户机向负载均衡模块申请数据修补模块资源进行数据修复。

UDP下载子模块在申请文件组播资源时，首先将需要下载的软件包信息提交给负载均衡模块。负载均衡模块反馈它文件组播服务器地址。同样在进行数据修补流程时，也需要先连接负载均衡模块，由负载均衡模块返回最佳的数据修补服务器的路径。在该客户端模块中设计校验子模块，在首次下载后对文件进行CRC校验和数据块完整性的检查。如果发现某些文件块的损坏或者丢失，模块调用数据修补模块对数据进行修补，重新下载损坏或丢失的数据。

* + 1. 流程的设计



图5-7 Udp组播下载客户端流程

图5-8 Tcp单播下载客户端流程

* + 1. 接口的设计

通过以上两张泳道图，整理得出如下接口：

IJoinMulitCastGroup接口负责加入对应的组播组。

IUdpReceive接口，通过该接口从服务器端使用UDP协议接收数据。

复用IOperateFile接口处理文件。

IExitMulitCastGroup接口负责现在完毕之后退出组播组。

IIntergrityCheck接口负责文件块的完整性检查，查看是否出现文件块的丢失。

ICrcCheck接口负责进行文件块的CRC检查。

复用ITcpLink接口，接口方法TcpConnect负责建立TCP连接。接口方法CloseConnecting负责文件下载完毕后，关闭TCP连接，释放资源。

ITcpReceive接口负责以TCP协议的流的方式接收数据。

IFail接口负责处理过程中的任何报错。

* + 1. 类的设计

UdpClient类

该类的功能是在组播模式下接收数据，它的属性有：

DataPort：应用层的端口

类的方法：

JoinMultiCastGroup方法将客户端加入进某个组播组。

UdpRecevie方法，客户端使用UDP协议接收组播数据报文。

ExitMultiCastGroup方法，客户端UDP下载完毕后，退出组播组。

TcpClient类

该类的功能是在单播模式下接收数据，它的方法有：

TcpConnect方法建立TCP连接。

TcpRecevie方法以TCP协议接收数据流。

CloseConnecting方法当客户端下载完毕之后，终止TCP连接，释放系统资源。

CheckFile类

根据之前的需求分析，模块设计FileCheck类，用于对文件进行检查。检查的方式为首先检查文件块是否存在遗失，再者可以对每个文件块进行CRC校验。

设计该类的方法：

IntegrityCheck方法，检查文件块的完整性。

CrcCheck方法对文件进行CRC检查。

模块类图



图5-9 Udp客户端模块类图



图5-10 Tcp客户端模块类图

* + 1. 接口或者类的实现

UdpClient模块的实现，使用C#语言实现UDP接收端的功能。图5-11为客户端下载数据时的界面。由于在流程自动化的要求下，下载程序是被控制脚本自动调用运行，不需要人为干预，所以设计为控制台程序。

protected int udpReceive (IPAddress ipAdd,int remotePort, int port)

{

byte[]receiveByte=null;

IPEndPoint remoteIpEndPoint = new IPEndPoint(ipAdd,remotePort);

udpReceive = new UdpClient(remoteIpEndPoint);

IPEndPoint iep = new IPEndPoint(ipAdd,port);

while(true)

{

try

{

receiveByte = udpReceive.Receive(ref iep);

}

catch (Exception e)

{

MessageBox.Show("Receive Data Error");

log.info("Response uut: " + errorInfo + ". Error Information " + UdpServerIP+UdpClientIP);

return (-1);

}

finally

{

udpReceive.Close();

}

}

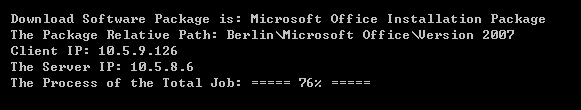


图5-11 软件运行界面

* 1. 负载均衡模块的详细设计
     1. 模块详细功能

负载均衡模块在本论文设计的系统中有至关重要的作用。当客户机开始首次组播下载时，负载均衡模块通过检查软件包的分布情况，各组播服务器的负载情况，通过负载均衡算法或者策略，指定一台最佳服务器给下载客户端。之后当客户端申请文件块修补的时候，负载均衡模块和数据修补模块其交互，得到各台数据修补服务器的负载情况，并根据对应的算法或者策略，提供客户端一台最佳的数据修补服务器作为下载服务器。

* + 1. 流程的设计

图5-12 负载均衡流程图

* + 1. 接口的设计

根据以上的分解、整理得到接口如下：

IReceiveInfo接口用来使负载均衡模块，得到客户端或者数据修补服务器的资源请求信息。经由客户端传送入的是软件包的请求信息，而数据修补服务器传送入的是文件块的请求信息。

IMethod接口通过对传送如的请求信息的判断，决定使用何种负载均衡策略。

IGetServerStatus接口负责从服务器得到服务器的即时信息，如文件服务器上的组播通道数、软件包资源的分布情况。数据修补服务器上连接的客户端的数量等。

IUdpLoadBalance接口负责负载均衡选择一台最空闲的文件服务器，提供客户端软件包下载。

ITcpLoadBalance接口负责选择一台合适的文件服务器，提供数据修补服务器获所需要的得文件块。

IFail接口负责处理报错信息。

* + 1. 类的设计

AcceptInfo类

该类的功能是负责和服务器进行交互，得到服务器状态。

AcceptInfo方法定时和Server交换，得到最新的server负载情况和server状态。

Mathod类

Mathod类，定义负载均衡算法的模式，即第一阶段使用UDP服务器的负载均衡算法，第二阶段使用TCP服务器的复制均衡算法。

Algorithm类

用特定的负载均衡算法，计算出特定的服务器，提供客户端数据下载服务。

UdpCalculate方法，当接收到请求下载软件包的请求时，负载均衡模块使用该方法选择某台文件服务器作为资源服务器。

TcpCalculate方法，当模块接收到文件块的下载请求时，负载均衡模块使用该方法选择某台文件服务器作为资源服务器。

GetNeedFile方法获得所需文件在各台服务器的分布状态。

SendInfo类

该类功能是和客户端进行信息交互。

SendInfoToClient方法向下载客户端提交服务器资源的信息。

GetClientIP方法获得客户端的IP，判断其所在网段。

模块类图:

图5-13 load balance模块类图

* + 1. 接口或者类的实现

使用C#语言实现负载均衡模块，通过负载均衡类内定义的方法，实现服务器资源的调度。图5-4为负载均衡模块内显示的服务器列表。

public void run()

{

log.info("run response uut thread begin.");

for (int i = 0; i < responseUUT.size(); i++) {

SendInfo responseOne = responseUUT.get(i);

responseOne.setStatus(SendInfo.STATUS\_INVALID\_UNKNOWN);

if (!responseOne.getKey().equals(key)) {

this.responseUUT(responseOne, UdpServerIP);

}

}

log.info("run response uut thread end.");

}

public int responseUUT(SendInfo uutInfo, String UdpServerIP) {

requestResource.UdpCalculate(uutInfo.getClientIp(),

uutInfo.getSn()

new Date(uutInfo.getFirstRequestTime())

uutInfo.getNeedImageFile()

uutInfo.getFileCode()

UdpServerIP);

log.info("Response uut: " + uutInfo + ". You can connect UdpServer: " + UdpServerIP);

Socket socket = null;

OutputStream os = null;

InputStream is = null;

}

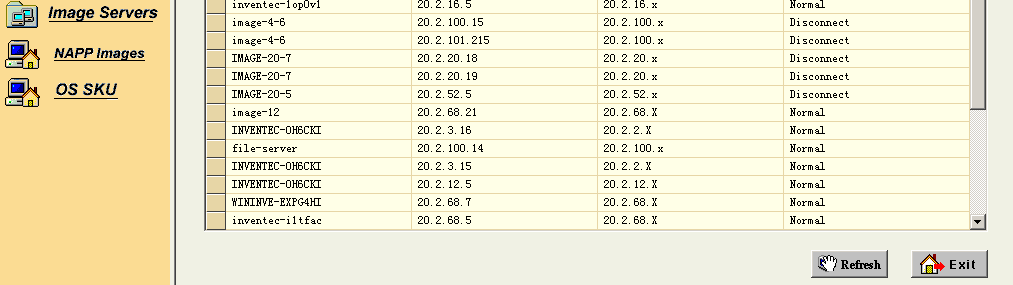


图5-14 系统运行界面

1. 系统的效能分析及主要特点

本章节分析系统产生的效能，论述系统采用的计算机控制生产的方式比手动操作拷贝机的生产模式，更自动化，更节省人力。结合试验环境下的测试数据，说明本系统比较传统生产模式或者其它一般单播下载系统的优势。

论文设计的文件分块，负载均衡，组播功能、单播功能等各模块是本系统的主要特点。

* 1. 本系统与拷贝机模式的比较

对比传统硬盘拷贝机的生产模式，本下载系统的生产方式在生产自动化率和产品品质方面都有明显的优势。以使用的人力资源为例，原先管理母盘的作业员按照班次计算总共为30人。如果使用本系统管理软件下载流程，仅仅需要2至3名助理就可以完成原先的工作。

在品质方面，拷贝机方式耗时长，并且没有品质的保障。生产定量的机器，还需要事先准备足量的硬盘，生产流程上无任何的弹性。如果导入下载系统的方案，笔记本电脑产品可以一次组装完毕，无需重新拆机，在流程上更加精简，同时可以避免由于拆机所导致的零件和外观的损坏等问题

* 1. 本系统与纯单播下载系统的比较

组播结合单播的下载机制和纯单播下载方式比较，组播技术可以减少系统的硬件投资。因为单播是一对一的传输，每一份数据（即使是相同的数据）都需从存贮器中读取，并且每一份数据都是单独从网卡发出，容易造成硬件的高负载。假设按照百兆网卡计算，10台笔记本电脑的网卡如果都需要达到线速接收，就可将一台千兆网卡的服务器的网卡带宽消耗殆尽。在服务器硬盘IO资源的使用方面也类似。按照现在工厂大规模制造的产能测算，建立生产用纯单播下载系统的硬件开销会造成企业很大的包袱。同时大量的服务器也会造成很大的电能消耗，还需要有更多专业人员进行日常的维护和保养，这些间接发生的成本，日积月累是一笔巨大的开销。

如果系统中拥有组播机制，服务器和客户机之间可以是一对多的传输，一份数据在服务器端的硬盘和网卡中只被执行一次，数据在网络节点中进行复制。还是用百兆网卡的客户端下载为例，服务器资源可以下降一个数量级。而且在后续的能源消耗、人力配置方面也会呈数量级的减少，成本大幅度降低。

比较本系统实际测得的数据和纯单播下载系统的效能，本下载系统具有明显的优势。主要体现在如下几个方面。首先是硬盘IO方面，由于硬盘的IO读写速度存在上限值，而每一个单播连接都需要直接消耗硬盘的IO资源。据模拟生产线的环境测试，一台主流配置的服务器，最多只能支持百兆笔记本电脑15台，如果连接数量大于15台，则服务器出现响应缓慢，客户端下载速度降为5Mb/s以下，后出现进程死亡。

图6-1所示单播下载随客户端增加而性能降低。当客户端数量从1台增加到5台时，系统效能已经降到大约55Mb/s。当客户端数量增加到15台或以上时，从实测的数据中看，已经没有任何流量流入客户端。此时观察服务器的硬盘IO占用率接近100%，客户机网卡占用率均小于10%。说明客户端已经无法从服务器端接收任何数据。此时服务器硬盘IO资源已经被消耗殆尽，成为系统的瓶颈之一。

其次是服务器网卡资源方面。现有主流服务器的网卡为千兆，按照单播协议的设计，每一个单播连接都机会均等向服务器发出请求，而服务器也将一视同仁地均分带宽给每台客户端。现有的笔记本电脑的有线网卡最低配置为百兆。一台主流服务器的单网卡能力理论值上最多支持10台笔记本电脑进行全线速下载。如果客户端的数量再增加，即使TCP协议的内部机制能保证在一定的时间内连接不会断开，数据下载的效率也会大幅降低。

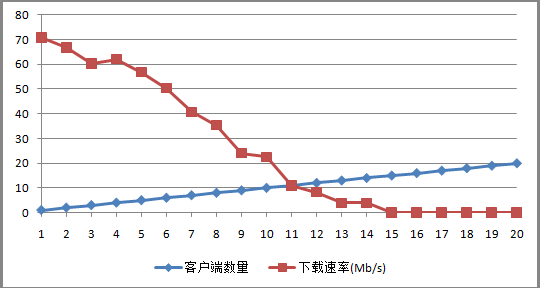


图6-1纯单播系统的下载效能

新系统中设计有组播和单播相互结合的下载机制。组播下载完成大部分的数据传输任务，单播只是负责流量较小的数据修补。并且组播服务器和单播服务器分别由不同的物理服务器担任，当下载的机器数量增加时，系统的整体效能还能保持平稳。图6-2说明本系统在客户端数量增加情况下的性能表现。下载客户端的数量逐渐由1台增加20台时，从客户端百兆网卡监测的数据表明下载速度稳定在60Mb/S左右。充分说明了组播在一对多网络传输方面的优势，正因为系统的组播机制，服务器端向代表一个接收群的组播地址发送数据。数据的复制工作交由网络设备完成，对服务器本身不产生额外的负载压力，系统的效能没有发生下降。

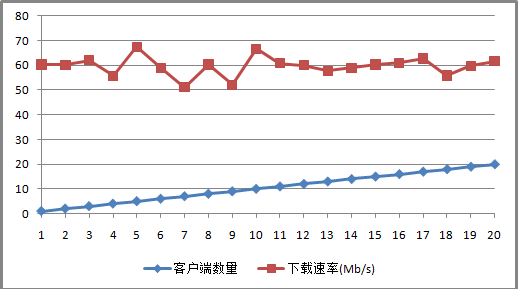


图6-2本系统的下载效能

* 1. 本系统的特点

组播和单播混合模式的下载系统之所以具有比其它解决方案更优秀的性能，是因为它结合了组播和单播的优势于一身。在系统中如下几个关键因素是系统达到设计预期的关键。

* + 1. 文件分块机制

本系统中设计有文件处理模块，将文件按照标准大小分割成小块，并且对每一个数据块进行编号，建立索引。将大文件分成小数据块，分别传输，可以提高组播传输的准确率。当机器首次组播下载完毕之后，大部分的数据都已经存在硬盘中。客户端的主控模块开始检查、拼接各数据块文件，如果存在数据块数量的减少或者数据的损坏。客户端随后开始数据修补进程，从数据修补服务器得到需要修补的数据块文件，而非全新下载。另外，在引入文件分块机制之后，才有基础实现即时加入、断点续传等其它功能。

* + 1. 循环组播机制

传统的组播下载系统，往往是按照通道分配系统资源。当组播组开始启动时，如果有后续的客户机申请加入，必须等待本组内当前批次的机器下载完毕。因为客户机器请求的数据的前面部分已经被服务器端发送完成。当文件组播下载进行到中间阶段，服务器无法在同一个组播组内，既发送现有序列的数据又同时发送之前已经送出的数据。

为了解决以上的问题，新系统的一种做法是先让客户机等待，当累计的一定数量的客户机时才激活组播组，可以让单个组播组承担更多的客户端数据下载。但是在实际生产环境中，机器是源源不断的流入下载制程，不能让客户端等待太久，以免浪费宝贵的生产时间和操作空间，降低了系统效率。所以系统还设计循环组播的方式下载数据。循环组播是指在存在客户端下载需求的情况下，组播服务器端周而复始地组播发送数据。客户端可以随时加入对应的组播组，从任何一个数据块文件开始下载数据。客户端组播下载软件利用数据块的索引信息，可以判断出自己是否已经完成一个完整周期的下载。只有在完成一个完整的下载周期后，客户端才开始执行后续的进程。

* + 1. 负载平衡机制

比较传统的下载系统，该系统设计了负载模块去主动平均各台文件服务器的负载。现实的工业级大规模生产，厂内有大量的服务器同时工作，服务器群的规模往往达到100台到150台左右。而订单的情况往往又不平均，如果没有负载均衡机制，服务器的负载会极不均衡。

在加入了负载均衡机制之后，将服务器资源统一纳入系统负载均衡模块的管理中，实现资源择优配置和调度，明显改善服务器资源的利用率。各台组播下载服务器上运行的组播组数量基本相同。各台文件修补服务器上的客户端连接数也大致相仿。客户端在首次下载时就必须和负载均衡模组交互，由负载均衡模块提供它组播文件服务器的IP地址，在数据修补时再次和负载均衡模组交互得到最适合的数据修补服务器IP地址。这种设计使得数据流负载最大程度的分散，使各台服务器负载平均，又互相备份。负载均衡机制比传统下载系统通常采用的热机备份的做法更科学合理。

* + 1. 组播和单播相结合

由于TCP协议本身面向连接的特性，传输数据的质量高。所以现有的主流下载软件在大流量的数据下载应用中，即使需要消耗大量的硬件资源也采用单播方式工作。

UDP传输可以实现一对多传输，建立连接的速度快，可以节省硬件的投资，降低成本。但是由于它是非面向连接的传输，数据在传输过程中会几率性出现丢失、损坏等现象，用它传输大流量的数据，需要有额外的辅助设计。所以要开发完美的数据下载系统就需要同时在系统中融合TCP和UDP两种下载机制，发挥组播和单播各自的优点，弥补双方各自的不足。

本论文设计的系统采用组播和单播结合的方式。系统中设计组播下载服务器端模块、组播下载客户端模块和数据修补模块（单播下载服务器端模块）将组播和单模相结合，实现了将组播和单播应用在同一个系统中。

引入组播和单播相结合的机制，将文件组播下载服务和数据修补服务分开，利用组播“最大限度，一对多转发”的特点，完成大部分数据的下载。系统把需要修补的文件块交给单播处理。同时组播服务器和单播服务器分开，使得客户端修补文件块时，不竞争正在以UDP协议快速发送数据的组播文件服务器的资源，而是向专门的数据修补服务器申请下载，使得组播所需的资源得到保证。在此前提下，也提升了首次数据下载的传输品质，又从另一方面减少了数据修补的需求，形成良性循环。

数据修补服务器中本身设计的队列机制，使得它可以缓存客户端的请求，逐步分批次地向客户端下载数据。此设计最大限度的减低了数据修补服务器上TCP同时传输的数量。避免了TCP协议的点对点下载在客户端大量出现的情况下，数据修补服务器被拖垮。当存在大量数据修补请求的客户端时，数据修补服务器缓存需求，客户端会存在一定时间的等待，但是相对整个下载所需要的数小时的耗时而言，这点时间可以忽略不计。

数据修补服务选用TCP协议，该协议族内部的保障机制保证了数据传输品质，可以确保修补的数据准确无误地被客户端接收。而且文件块本身被划分得很小，TCP协议以流的方式传输的文件块，单次发送的量很小。即使在网络繁忙状态下，笔记本电脑和数据修补服务器之间的数据传输质量依然可以得到保证。

因为有了以上的各机制，才使论文设计的组播和单播相结合的数据下载系统在实际应用中优势明显。

1. 结论与展望

本论文以笔记本电脑制造企业为应用背景，分析企业在出厂软件下载工序中的需求，结合计算机组播和单播技术，设计并实现了低成本、高效能的数据下载系统。新系统既具备组播的高效，又拥有单播的可靠，解决了企业在生产中面临的难题。论文详细设计了文件分块、组播下载、单播下载和负载均衡等核心模块，用理论分析、数据对比的方法证明了新系统比传统生产模式或者单播下载系统在效率、可靠度和成本方面具有明显的优势。

中国拥有大量的数码产品制造企业。企业要生存、发展必须以信息化和网络化为基础，这是蕴藏着庞大商机和无限机会的领域。而且数码产品被不断地推陈出新，在产品更新换代的浪潮中，产品中软件的附加价值在不断提高。数码产品离开软件就无法使用。出厂前安装正版操作系统和应用软件成为企业必不可少的工序，所以本论文所设计的系统的现实应用价值很高。

除此之外，本论文所涉及的理论、技术在视屏会议、数字资料传输等等其它商业领域也有广阔的应用空间。除了在企业中的应用，这些理论和技术在个人应用中也是前景无限。从视屏聊天，视屏音频点播，到网络游戏，虚拟社区等都可以结合运用计算机网络的组播和单播的各项理论和技术，研发出更好的系统为人们服务。

计算机科技的发展持续改变我们的生活。它使得整个世界变得更快，更小，更扁平，也成为我们每个平凡人都参与其中的生活方式。展望未来，现在基于有线媒介的系统会被移植到无线传输的物理平台上。计算机的组播和单播的理论和技术会在无线的物理平台上继续发展。物理层的实现方式不同也会对系统的设计和构建提出新的要求。未来的网络数据下载或传输系统能够在无线平台上提供可靠、稳定、低能耗的服务。

此外基于IPV4的互联网地址资源不久也会被人类耗尽，未来是属于IPV6的时代。在IPV6的新规范下，组播和单播的理论会有新的发展和突破。新的理论和模型将会应运而生，使用IPV6的系统将会愈来愈普遍。

总之，计算机信息科技将持续发展；网络的组播和单播理论、技术将持续发展，它们渗透进人类社会的各个领域，服务于人们生产、生活的各个方面，带给我们更便捷、更高效、更环保、更美好的新生活。

参考文献

1. S Deering,D Cheriton.Multicast Routing in Datagram Internetworks and Extended LANs. ACM Transactions on Computer Systems,1990,8(2): 85-110.
2. D Thaler, M Handley, D Estrin. The Internet Multicast Address Allocation Architecture. RFC2908
3. S Hanna, B Patel, M Shah. Multicast Address Dynamic Client Allocation Protocol. RFC2730, December 1999
4. H Holbrook, B Cain. An Overview of Source-Specific Multicast (SSM).RFC 3569, July 2003.
5. D Meyer, R Rockell, G.Shepherd. Source-Specific Protocol Independent Multicast in 232/8. IETF Internet Draft, <draft-ietf-mboned-ssm232-08.txt>, work in progress, March 2004
6. J Moy.Multicast Extensions to OSPF.RFC1584,1994.
7. RFC896 Congestion Control in IP/TCP Internetworks,J.Nagle. January,1984
8. RFC2581 Known TCP Congestion Control,M.Alman,V.Panon,W.Stevens March,1999
9. RFC768 User Datagram Protocol, J.Postel.August 1980
10. RFC1075 Distance Vector Multicast Routing Protocol, D.Waitzman,C.Partridge,S.E.Deering. November,1988
11. B Cain, S Deering, B Fenner, A Thyagarajan. Internet Group Management Protocol, Version 3.RFC 3376, October 2002.
12. B Haberman, J Martin. IGMPv3/MLDv2 and Multicast Routing Protocol Interaction. IETF Internet Draft. draft-ietf-magma- igmpv3-and-routing-05.txt, work in progress, October 2003.
13. H Holbrook, B Cain, B. Haberman. IGMPv3 and MLDv2 for SSM. IETF Internet Draft, draft-holbrook-idmr-igmpv3-ssm-06.txt, work in progress, August 2004.
14. D Estrin et al Protocol Independent Multicast-Sparse Mode(PIM-SM):Protocol Specification. RFC2362,June 1998
15. （美）Martin L.Shoemaker著，高猛 朱洁梅译 UML实战教程——面向.NET开发人员 清华大学出版社
16. 袁涛 孔蕾蕾 编著 统一建模语言UML 清华大学出版社
17. 徐恪，吴建平，徐伟明 高等计算机网络——体系结构、协议机制、算法设计与路由技术 北京：机械工业出版社
18. 吴功宜 编著 计算机网络（第二版） 清华大学出版社
19. 戴伟辉 孙海 黄丽华 编著 信息系统分析与设计 高等教育出版社
20. （美）Karli Watson Christian Nagel著，齐力波 翻译 黄静 审校 C#入门经典 清华大学出版社
21. 王寅永 李降宇 李广歌 编著 C#深入详解 电子工业出版社
22. 梅晓冬 颜烨华 编著 Visual C#网络编程技术与实践 清华大学出版社
23. 马俊 郑逢斌 沈夏炯 编著 C#网络应用高级编程 人民邮电出版社
24. （美）Wei-Meng Lee 著 田国法 吴兰涉 译 C#与VB.NET网络通信开发实战 人民邮电出版
25. 吴晨 陈建孝 编著C# 网络与通信程序设计案例精讲 清华大学出版社

致谢

在整个复旦大学专业硕士的学习期间，我学习了许多计算机信息领域的新知识，不断接受来自计算机科学领域的新观念和新成果。特别是论文的撰写，更使我有机会系统地整理全部的学习成果，起到了提纲挈领，加深理解的作用。掌握知识的最佳途径是实践，我一定将我的所学所得运用到将来的工作和生活中去，在实践中学习，在学习中实践，与时俱进，不断自我提高。

我在此感谢复旦大学的老师们对我的谆谆教导和悉心培育。特别感谢指导我论文写作的徐迎晓老师。徐老师治学严谨，知识渊博，他的辅导对我完成我的硕士论文帮助良多。

我还要感谢我的妻子，感谢她对我学习的支持和在我遇到困难时对我的一贯鼓励。还要感谢我的女儿，她成长过程中的每一点进步都带给我极大的快乐和向前的动力。

更要感谢我的父母，谢谢他们对我的养育之恩，祝愿他们身心舒畅，健康长寿。

论文独创性声明

本论文是我个人在导师指导下进行的研究工作及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其它机构已经发表或撰写过的研究成果。其他同志对本研究的启发和所做的贡献均已在论文中作了明确的声明并表示了谢意。

作者签名： 张晓欧 日期：2011年2月22日

**论文使用授权声明**

本人完全了解复旦大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存论文。保密的论文在解密后遵守此规定。

作者签名： 张晓欧 导师签名： 徐迎晓 日期：2011年2月22日