摘 要

以太网技术的广泛应用，带来的是用户数量的迅速增长。如今以太网已经垄断了局域网领域，有超过95%的用户使用以太网连接其内部网络。同时，越来越多的新业务的出现使得以太网承载的业务类型也不断增加，不仅要为传统的数据传输业务提供服务，还要传送实时的话音或视频业务。不同的用户和不同的应用对服务质量有着不同的需求，这些都是靠传统以太网技术和增加网络带宽所不能解决的问题。本文关注的 Residential Ethernet 技术就是以改善在以太网上传输话音、视频流等实时业务服务质量的一种技术，它通过在传统以太网技术上增加了各个交换设备之间的时间同步，从而产生同步的超帧起始信号，然后按照超帧结构传输各种业务的数据；特定的帧结构设计有利于更好地更有效地控制实时业务的时延与时延抖动；量化一点的性能目标是：经过一个设备给实时业务流带来的时延与时延抖动分别不超过两个超帧与一个超帧的时间长度（目前超帧的时间长度为125us[1]）。本文以实现同步以太网交换机为目标进行大量研发工作，对多播注册与资源预留、以太网网桥之间的资源协同调度、自适应的多跳时分复用的资源调度、时间同步及其算法改进这几个方面开展了研究。本文的主要工作如下：

同步以太网交换机的研发：首先跟踪学习 IEEE802.3 工作组关于同步以太网技术的技术提案,；然后与韩国总部讨论开发内容,确定研发内容分为三个阶段:第一步阶段的任务是在单片 FPGA 内实现同步

以太网交换机，完成五路千兆同步/异步混合报文的分析、交换和发送，并完成主从设备的选择与设备间的时间同步；第二阶段的任务是在单片 FPGA 内完成同步交换与异步交换的集成，实现单片 FPGA 支持同步与异步两种交换的功能，经过集成交换机处理后的同步业务服务质量不受异步业务流的影响；第三阶段的任务是优化与改进设备间时间同步的算法并实现，尤其要改进多跳设备互连情况下设备之间同步后的精度，克服与解决传统同步算法产生的级联误差随级联级数呈指数分布的缺陷，提高同步精度。

多播注册与资源预留的研究：首先研究了传统以太网中的多播技术不能有效保证服务质量的问题；然后提出一种新的综合多播和资源预留的解决方案；该方案中，接收者可以通过发送单个消息来达到多播注册和资源预留的目的，从而能够快速，高效地提供多播服务质量的支持。

以太网交换机资源的调度的研究：首先研究了以太网交换机在调度低带宽高时延性能需求的应用与高带宽性能需求的应用时存在的问题，然后提出两个专利。这两个专利的提出可以使得以太网有效地支持低带宽高延迟性能需求的应用以及在保证端到端延迟需求的同时兼顾高带宽需求应用的接入请求。

时间同步算法的研究：首先研究了传统的利用精确时间协议实现时间同步的算法；并在硬件平台上实现并测试了传统算法在单跳与多跳环境下的时间同步精度，测试结果发现传统算法的单跳同步精度有待提高；而且验证了传统同步算法产生的级联误差随级联级数呈指数

分布的缺陷；在此基础上提出一种改进算法。改进算法能有效提高单跳时的时间同步精度，并在多跳环境下能有效克服级联误差随级联级数呈指数分布的缺陷，显著提高了级联后的时间同步的性能。

关键词：服务质量、资源预留协议、频率补偿算法、超帧、精确时间协议、在场可编程门阵列、千兆位介质无关接口

Abstract

With the wide range of use of Ethernet network, the amount of users grows rapidly. Now Ethernet is playing a key important role in local area network. It is reported that 95% network is connected by Ethernet. And at the same time, more and more new services appear on Ethernet and put great challenge on traditional Ethernet network to support conventional best-effort service as well as real-time service such as Video and Audio services. It is impossible for Ethernet to solve this problem just by adding network bandwidth. Under this situation, Residential Ethernet is brought forward to address such problems. The aim of Residential Ethernet is to improve and assure the quality of real-time service transmitted on it. By adding time synchronization function on each traditional switch and using special super-frame architecture, Residential Ethernet switch can assure maximum delay when synchronous packet passes it. The quantitative delay object that a real-time packet needed to pass through a Residential Ethernet switch is less than 2 super-frame length (One super-frame length is 125us). We do a great amount of FPGA and software development related of work to implement Residential Ethernet switch. And at the same time, we do much research on such aspects

such as Multicast Registering and Resource Reservation, Resource Coordinated Scheduling between bridges, Adaptive Multi-hop TDMA resource Scheduling and algorithm improvement on conventional time synchronization based on frequency-only compensation correction. The main work of this paper is stated in what follows.

Research and development of Residential Ethernet switch: Firstly, we follow the technical proposal about Residential Ethernet technology which is under control 802.3 Residential Ethernet study group. Secondly, discuss with Korea headquarter about the work contents that we need to develop. Our work contents can be divided into three relatively independent three phases. The work content for first phase is to: develop Residential Ethernet switch in one PFPGA with embedded CPU and implement time synchronization function in switch. The work content for second phase is to integrate Residential switch and asynchronous switch into one PFGA chip and do resource analysis required for PFGA. The work content for third phase is to optimize and improve the algorithm used for synchronizing each device by using the process of precise time protocol. The aim is to decrease synchronization error, especially in cascaded environment.

Research for broadcast registering and resource reservation: Firstly, we study the problem that conventional broadcast technology can' t assure the quality on Ethernet. Secondly, we propose one scheme that integrates broadcast and resource reservation. By using this scheme, receiver can finish broadcasting registering and resource reservation by sending out single message, so as to provide support to quality of fast and efficient broadcasting.

Research for resource scheduling of Ethernet bridges: Firstly, we investigate the problems existing to support two kinds of applications. These applications are low-bandwidth but high-delay requirement application and high-bandwidth requirement application. Then we propose two patents to address the problem so as to effectively support low-bandwidth but high-delay requirement application and still increase the admission probability of high-bandwidth requirement application

Research and improvement of time synchronization: Firstly, we first study conventional scheme for time synchronization by using precise time protocol. Then we implement conventional scheme in our FPGA. The experimental result shows that conventional scheme has the problem that synchronization error

increase exponentially with the hop number of cascaded bridges. This defect isn' t acceptable. Based on conventional scheme, we propose one improved scheme. Our improved scheme can overcome the defects of traditional scheme and increases synchronization precision markedly.

Key words:

Quality of service, Resource Reservation protocol, frequency compensation correction, super-frame, precise time protocol (PTP), field programmable gate array (FPGA), Gigabit Media Independent Interface (GMII)

第一章 绪论

1.1 研究背景

过去的以太网技术主要应用于局域网环境中，连接的设备数量较少。在以太网传输速率由十兆发展到百兆、千兆的同时，以太网设备的成本也在不断降低，这导致网络中的设备数量迅速增加；交换技术在以太网中的应用，通过提供优先级的区分、全双工通信等功能，极大地提高了以太网的性能。而且，以太网的统计复用方式更适于传输数据，而且成本也低得多。随着千兆以太网的普及和 10G 以太网技术的发展，以太网技术已经完全能够满足驻地网与城域网所需的速率要求。如今以太网已经垄断了 LAN 领域，有超过 95%的用户使用以太网连接其内部网络。与此此时，传统的 Ethernet 只提供尽力而为类型的服务，这种类型的服务非常适合一些传统的应用，如电子邮件、HTTP 与 FTP 业务等。但是随着网络技术的快速发展与 Internet 的迅速普及，Ethernet 上承载的业务类型越来越多，尤其是像会议电视、VoIP、远程教学、远程医疗和网络电视等宽带多媒体业务更是有着巨大的市场需求。这些业务对 Ethernet 提出了服务质量要求：它们不仅要求一定的传输带宽，而且对单向时延、时延抖动以及报文丢失率也提出了要求。ITU-T G.1010 建议[2]从单向时延、时延抖动以及报文丢失率这三个方面对话音业务与视频业务所要达到的目标进行了规定。G.1010 规定交互式话音业务的单向时延不超过 150ms，时延抖动应小于 1ms，报文丢失率应小于 3%；而 G.1010 规定交互式视频业务的单向时延的短期目标是不超过 400ms，其长期目标是不超过 150ms，报文丢失率应小于 1%，它还要求视频与音频必须同步在 80ms 内。这种服务质量的要求给传统 Ethernet 网络带来了巨大的挑战。

同步以太网就是在这种背景下提出了，它的提出，致力于解决家庭网络、驻地网络甚至城域范围网络在传输实时业务如话音、视频通信等业务时存在的问题。为了解决这些问题，同步以太网新提出了几项核心技术，包括设备间时间同步、资源预留策略、流量接入调度策略、超帧传输结构等，这些技术旨在传统以太网技术上增加了各个交换设备之间的时间同步，从而产生同步的超帧起始信号，然后按照超帧结构传输各种业务的数据，特定的帧结构设计有利于更好地更

有效地控制实时业务的时延与时延抖动；量化一点的性能目标是：经过一个设备给实时业务流带来的最大时延不超过两个超帧的时间长度（目前超帧的长度为125us）；而经过一个设备给实时业务流带来的最大时延抖动不超过一个超帧的时间长度。

鉴于以上的挑战与新技术的提出，本文以实现同步以太网交换机为目标进行大量研发工作，对多播注册与资源预留、以太网网桥之间的协同调度、自适应的多跳时分复用调度、时间同步算法改进这几个方面开展了研究。

本章的后续部分概述了本文在这一领域的创新和主要工作、以及论文后续部分的组织结构和内容。

1.2 博士后工作成果和研究创新点

我们项目组以实现同步以太网交换机为目标进行了大量的研发工作，对多播注册与资源预留、以太网网桥之间的协同调度、自适应的多跳时分复用调度、时间同步算法改进这几个方面开展了研究。主要的创新与工作如下：

(1) **同步以太网交换机的研发**：首先跟踪学习 IEEE802.3 工作组关于同步以太网技术的技术提案，；然后与韩国总部讨论开发内容，确定研发内容分为三个阶段。第一步阶段的任务是研发同步以太网交换机：完成混合报文的分析、交换和发送，以及完成主时间设备的选择与设备间的时间同步；第二阶段的任务是在FPGA 内完成同步交换阵与异步交换阵的集成，实现单片 FPGA 支持同步与异步流量两种交换的功能，经过集成交换机处理后的同步业务的服务质量不受异步业务的影响；第三阶段的任务是优化与改进设备间时间同步的算法并实现，尤其要改进多跳环境下设备之间时间同步的同步精度，克服与解决传统同步算法产生的级联误差随级联级数呈指数分布的缺陷，提高同步精度。

(2) **一种单播方式的以太网多播控制信息传递方法的研究**：首先研究了传统的多播方式传送多播控制信息的缺点，在此基础上提出了使用单播方式传递以太网的多播控制信息。使用单播方式有两个好处：第一、由于消息是单播方式传送，因此它不会被传递到全网；第二、网桥可以很容易地区分多播控制信息和数据流，这样就使得网桥能够高效地对两者进行分别处理。

(3) **多播注册与资源预留的研究**：首先研究了传统以太网中的多播技术不能

有效保证服务质量的问题；然后提出一种新的综合多播和资源预留的解决方案；该方案中，接收者可以通过发送单个消息来达到多播注册和资源预留的目的，从而能够快速、高效地支持多播服务的质量。

(4)以太网交换机资源调度的研究：首先研究了以太网交换机在调度低带宽高时延性能需求的应用与高带宽性能需求的应用时存在的问题；然后提出两个专利。这两个专利的提出可以使得以太网有效地支持低带宽高延迟性能需求的应用以及在保证端到端延迟需求的同时兼顾高带宽需求应用的接入请求。

(5)时间同步算法的研究：首先研究了传统的时间同步算法；然后在硬件平台上实现并测试了传统算法在单跳与多跳环境下的时间同步精度，测试结果发现传统算法的单跳同步精度有待提高；而且验证了传统同步算法产生的级联误差随级联级数呈指数分布的缺陷；在此基础上提出一种改进算法。改进算法能有效提高单跳时的时间同步精度，并在多跳环境下能有效克服级联误差随级联级数呈指数分布的缺陷，显著提高了级联后时间同步的性能。

1.3 博士后出站报告结构

本文后续部分的内容安排如下：

第二章 开发工作的总结：首先对博士后课题的开发工作进行概述，然后对开发平台、开发环境、使用的操作系统、研发工具与测试设备进行了介绍；然后对三个阶段的研发工作分别从研发需求、系统模块设计与测试结果这几个方面进行了描述。最后，对本章进行了总结。

第三章 研究工作的总结：首先对提出的五个专利进行了概述；然后对每个专利从专利提出的技术背景、专利细节描述与仿真或测试结果这几个方面进行了详细描述。最后，对本章进行了总结。

第四章 本文总结与展望：对本文所做的工作进行了简要的总结，并对未来的研究作了展望。

在文章的最后，给出了本文所引用的参考文献、致谢等内容。

第二章 开发工作的总结

2.1 概述

RSE 相关课题的研发工作是两年工作量，经过我们项目组的共同努力，仅用了一年半的时间就提前完成了。它的执行过程可以划分为三个阶段，在每一个阶段结束时都通过了韩国三星总部严格的验收测试,每一阶段的工作划分参见后续章节。课题主要的开发内容围绕 FPGA 逻辑与嵌入在 FPGA 芯片内部的 CPU 核进行同步以太网相关内容的开发，主要开发涉及到 FPGA 代码编写与调试、CPU 核的外围平台的搭建与调试、嵌入式 Linux 内核移植、驱动程序代码的编写与调试、以及应用层软件代码的编写与调试。

第一阶段的主要工作是开发同步以太网交换机与时间同步系统；第二阶段的主要工作是在单片 FPGA 内集成同步以太网交换机与传统以太网交换机，评估 FPGA 资源占用情况；第三阶段的工作集中在时间同步协议的验证与改进方面。至于这三个阶段具体的工作情况，请参见 2.2 章节。

在随后的半年时间里，我们将承接新的课题项目，以扩展博士后课题的研究范围与研究内容。下面我将就开发平台与开发环境、开发工作等方面进行介绍。

2.1.1 开发平台与开发环境介绍

开发板介绍

RSE 开发板由韩国总部开发完成，整个开发板对外提供四路千兆以太网 RJ45 接口与一路光接口，板子上面集成了千兆以太网物理层芯片、32MB Flash 存储器、8MB SDRAM 与 XILINX FPGA 芯片。FPGA 芯片的型号是 VirtexII Pro 系列的 xc2vp40-ff1152-6，芯片门数高达 4 百万门，它内部内嵌了两个 PPC405 CPU 核，且能够提供 3456KB 的块存储器供开发者使用。由于这款 FPGA 的强大功能，我们可以在单片 FPGA 上实现片上系统的开发，而且使得项目的开发具有很大的灵活性，而不需要额外的 CPU 控制板。当然，这种片上系统项目的设计涉及到 FPGA 代码的开发、CPU 核外围平台的搭建、嵌入式 Linux 的移植、驱动程序的开发以应用代码的编写，给项目带来了巨大的难度与挑战性。

操作系统介绍

我们选择了 MontaVista Linux 嵌入式操作系统来开发我们的项目，MontaVista Linux 操作系统是在实时嵌入式领域里广泛应用的系统软件，它同时也是一个开发环境。它可以对软硬件资源进行有效的管理，并提供人机接口，它提供了基于标准化开放系统的完整的多任务环境，我们使用的 MontaVista Linux 版本是 Montavista Linux 专业版 PE 3.1，它充分利用 Linux 开放源代码的稳定性、高性能和可扩展性的优点，增强 Linux 2.4.20 内核以适应嵌入式应用，包括优化嵌入式应用的高级工具。专业版还包含的 Montavista DevRocket 交叉开发平台，它具有综合开发和分析工具，可降低项目开发风险。该版本 Linux 包含 C/C++编辑浏览器，源码级调试器，编译器和图形运行分析工具。用户可以在 GNU 环境中轻松地完成编辑、编译、调试等工作。

为满足嵌入式设计的实时性能需求，MontaVista Linux 采用具有战略性的方案，既显著提高实时响应而又保留健壮的 Linux 用户编程模式和标准 Linux API。MontaVista 可抢占内核技术明显降低了内核延迟和抖动，保证了毫秒级的系统最坏实时响应。为了优化 Linux 进程/线程调度，MontaVista 在内核里集成了固定调度开销的实时调度器，提供了可配置实时优先级和具有 CPU 亲和性（affinity）。MontaVista 在 PE 3.1 中引入了高分辨率定时器（High Resolution Timers），给开发人员提供了增加实时程序的控制。使用 HRT，编程人员能实现微秒级精度的基于时间、事件驱动的新算法，减少作 CPU 时钟周期的轮询和空循环时间开销。这些技术在不影响 Linux 优势的基础上为嵌入式应用开发提供了一个实时开发平台。专业版以 MontaVista DevRocket 为主要特色，提供了交叉和本地工具链的集成开发环境，具有基于图形化界面的调试能力、软件和性能分析能力和用于提高开发效率强大的工程向导。专业版也包含超过 250 种 Linux 系统应用包，可被用于快速构建平台映像。依靠 MontaVista Software 的技术领先性和与流行微处理器和板级供应商的伙伴关系，以及先进的特色、高级的 OS、完整的开发环境和 MontaVista 产品订阅模式所提供的可继承优势，MontaVista Linux 专业版在世界范围内被成百上千的用户使用。

研发工具与设备介绍

在开发过程中，我们主要使用的研发工具有：

1. Modelsim SE 5.8d，它主要用来编译/仿真我们的 VHDL 代码，Modelsim 卓越而强大的功能给我们项目的开展提供了很大的便利与支持；
2. ISE6.3i，它主要用来对 FPGA 代码进行综合、布局和布线；
3. EDK6.3i，它主要用来生成 CPU 核外围设备相关的工程文件与网表文件，并生成与嵌入式 Linux 相关的驱动与参数定义文件；
4. BDI2000，它主要用来控制 CPU、并通过它下载软件与调试操作；
5. Smartbits 流量产生仪，它能够产生 1000Mbps 的千兆线速流量的报文输入我们设计的 FPGA 里，然后通过它接收报文来观察从我们 FPGA 里输出的报文是否有错，是否有丢包以及时延特性，并根据它的分析结果寻找代码的缺陷，这种测试仪对于我们项目的调试来说是不可缺少的。
6. Chipscope6.3i，片内逻辑分析仪，它的主要功能是通过 JTAG 编程接口，在线、实时地读出 FPGA 内部的信号。其基本原理是利用 FPGA 中未使用的块存储器，根据用户设定的触发条件，将信号实时地保存到这些存储块中，然后通过 JTAG 接口传送到计算机，并通过计算机的用户界面 GUI 显示出所采集的时序波形，大大方便了对 FPGA 内部信号的跟踪与调试。

2.1.2 开发工作的阶段划分

RSE 相关课题的执行过程可以划分为三个阶段。第一阶段的主要工作是开发同步以太网交换机与时间同步系统；第二阶段的主要工作是在单片 FPGA 内集成同步以太网交换机与传统以太网交换机，评估 FPGA 资源占用情况；第三阶段的工作集中在时间同步协议的验证与改进方面。下面我们将仔细介绍这三个阶段具体的工作情况。

2.2 第一阶段工作总结

第一阶段工作的持续时间是从2004.11到2005.7。这期间项目组共五名成员，我担任项目经理，其他四名成员是郑剑锋博士、谭兴晔博士、吴起博士与毕务刚。

2.2.1 系统设计需求

功能需求：

1. 开发 FPGA 代码实现从千兆以太网接口上接收报文，并识别与插入超帧，超帧长度为 125us；
2. 开发 FPGA 代码实现对同步报文、异步报文与协议控制报文的分析，并提取出源/目的 MAC 地址、协议类型与报文长度；完成对报文的 CRC 检验；完成对报文的存储与控制；将同步报文发送到同步交换阵；将异步报文发送到外部 GMII 接口；将协议控制报文存储到内部存储器，供 CPU 读取；
3. 开发 FPGA 代码实现对同步报文、异步报文与协议控制报文这三种报文的发送控制，并复用到千兆以太网端口上；复用时需要遵守超帧的格式。三种报文中同步报文的优先级最高，协议控制报文次之，异步报文的优先级最低；为了保证同步报文的服务质量，需要对异步报文进行 HOLD 控制；
4. 软件与 FPGA 协同开发实现时间同步模块；
   a) FPGA 提供时间计时功能，并根据软件的设置进行时间调整；与此同时提供时间信息给报文分析模块；并根据报文复用模块提供的指示信号锁存时间信息供应用层软件使用；
   b) 软件需要完成的功能分为两个部分：
      i. 从报文分析模块里读取协议控制报文；
      ii. 将协议控制报文发送到报文复用模块；
      iii. 根据主同步设备的选择原则，从多个设备选择出主同步设备，其它设备为从设备；
      iv. 根据 IEEE1588 时间同步协议，完成主从同步设备之间的时间信息的交互，并实现设备间的时间同步；
      v. 设计并实现算法实现对时钟漂移的补偿，实现主从设备之间的精确同步；
      vi. 设计人机界面与一些控制接口，并通过人机界面控制是否进行同步以及各种参数的调整。
5. 开发 FPGA 代码实现 5X5 千兆交换阵，需要完成：
   a) 完成同步报文 MAC 地址的自学习功能；

b) 完成同步报文 MAC 地址超时自动删除功能；

c) 对同步报文交换的性能需要达到 5 路千兆输入内部交换无阻塞；

6. 需要完成以上模块的功能仿真、单元测试、集成测试与系统测试。

性能需求：

1. 千兆线速情况下异步报文通过率要求达到 100%，测试报文数目需要达到 10^4 以上；
2. 千兆线速情况下同步报文通过率要求达到 95%以上，测试报文数目需要达到 10^8 以上；
3. 异步报文与同步报文混合输出时（以下称混合流量或混合报文）：

a) 70%同步报文+20%异步报文，要求两种报文均不能出现报文丢失；

b) 70%同步报文+30%异步报文，要求同步报文不能出现报文丢失；

c) 70%同步报文+50%异步报文，要求同步报文不能出现报文丢失；

d) 70%同步报文+100%异步报文，要求同步报文不能出现报文丢失；

4. 两个设备之间时间同步的误差要求在[-500ns, 500ns]范围以内。

2.2.2 系统设计描述

系统设计的总体模块框图如下图所示。

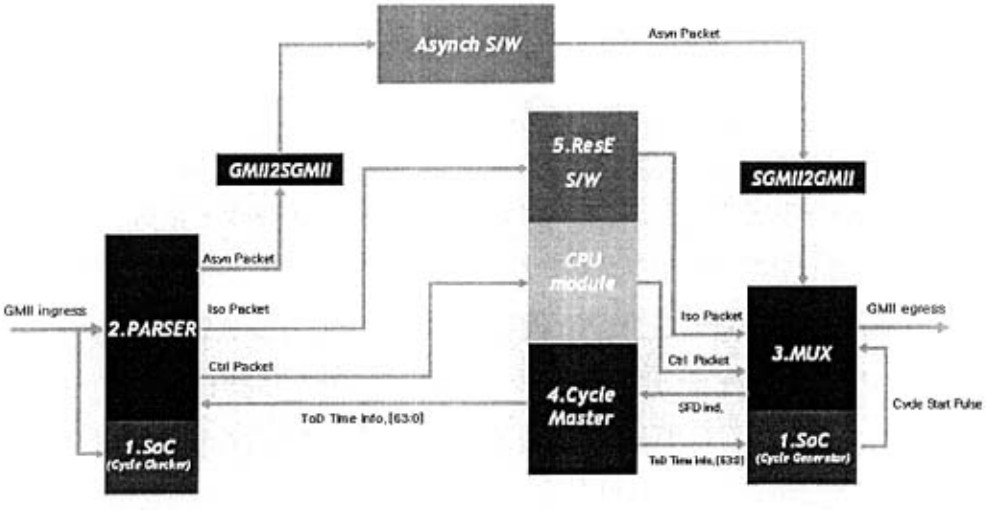


图 1 总体模块框图

FPGA 设计包括以下模块：

- 报文分析模块—Parser 模块
- 报文复用模块—MUX 模块
- 同步交换阵模块—ResE S/W 模块
- 时间同步模块—Cycle Master 模块
- CPU 接口模块—CPU interface 模块
- 超帧脉冲产生模块—SoC 模块
- GMII to SGMII 模块，由韩国总部人员开发

从上图可以看出，FPGA 系统主要包括 7 个模块。下面我们将描述我们为何需要设计以上几个模块。

报文从物理线路进行 RJ45 接口后，经物理层芯片处理后变成标准的 GMII 接口送到 FPGA 芯片。这些报文首先需要被 Parser 模块与 SoC 模块处理。Parser 模块根据报文中携带的 Ethernet 类型域携带的类型关键字将以太网报文划分成三种类型：同步报文、异步报文与协议控制报文。然后，同步报文将根据目的 MAC 地址查表结果发送到同步交换阵模块；异步报文将通过 GMIItoSGMII 转换模块发送到 FPGA 外部的异步交换阵模块进行相应处理；协议控制报文，主要包括时间同步相关的协议报文与设备能力广播报文，将被 Parser 模块发送到内部存储器内；如果该协议控制报文是时间同步相关的协议报文，将打上该报文到达时刻的系统时间标签供应用层软件使用。Parser 模块还将对所有的报文进行 CRC 检查，如果发现错误，将报告错误结果。

同步交换阵模块将根据同步报文的目的 MAC 地址的查表结果将报文交换到相应的输出端口，如果查表结果没有匹配项，将该报文广播其它端口。而且同步交换阵需要支持 MAC 地址自动学习、自动超时更新与静态路由配置功能。

CPU 接口模块用于适配 PLB 总线信号，并对地址信号进行译码，产生相应的读写控制信号来访问报文复用模块、同步交换阵模块以及时间同步模块。

报文复用模块有三个数据来源，它们是：来自与 FPGA 芯片外部的异步交换阵的异步报文；来自同步交换阵的同步报文以及来自 CPU 接口模块的协议处理报文。报文复用模块将根据超帧的要求将这三种报文复用到 GMII 接口上。

SoC 模块将要检测超帧的抖动是否在系统允许的范围内，如果超限，将产生超帧抖动超限指示信号指导随后处理。此外，SoC 模块还将根据时间同步模块输

出的时间信息产生周期为 125us 的超帧起始信号，并将该超帧起始信号发送到报文复用模块，报文复用模块将根据超帧起始信号按照超帧格式向 GMII 接口上发送报文。

时间同步模块将维护时间计数器，并根据 CPU 设定的控制命令进行时间调整，以达到设备间时间的同步。时间同步模块将给报文分析模块提供时间信息。此外，时间同步模块将根据报文复用模块产生的 SFD 指示信号锁存当前系统的时戳，以供应用层软件读取使用。

2.2.3 系统测试结果

测试结论：第一阶段的工作通过了韩国总部要求的所有测试。

测试连接图如下图所示，其中 GRE 板子就是指韩国总部开发的硬件平台。下图描述了两种配置连接图，配置(a)用于测试单板的性能，配置(b)用于测试两个板子级联后的性能。

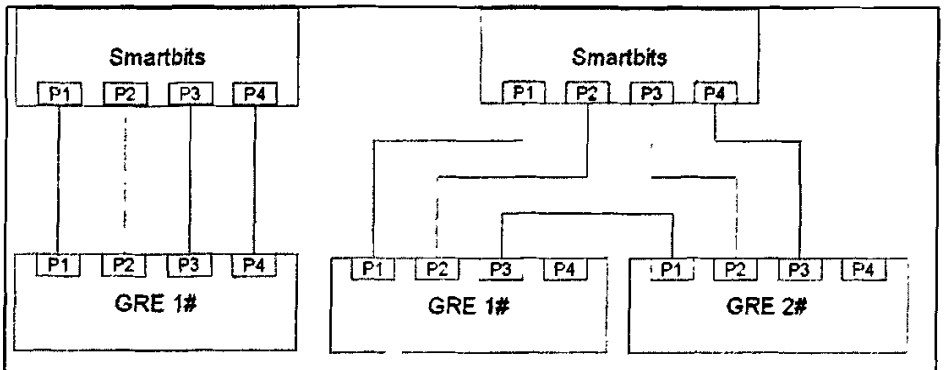


Fig 2 Test Configuration

下面两个表分别总结了同步以太网交换机性能与异步交换机性能的测试结果，在每个表的后面，给出了部分测试场景下的详细测试配置与测试结果。

表 1： 同步交换机的测试结果

1. 测试同步交换机功能

单跳或级联测试		期望的结果	测试结果：是否通过
异步报文测试	单跳 (Fig.2.a)	100%异步报文通过	通过

	级联 (Fig.2.b)		100%异步报文通过	通过
同步报文测试	级联 (Fig.2.b)		同步报文通过率为95%以上	通过率为 97.9% 。通过
同步异步混合流量测试(Fig.2.b)	70%同步流量	20%异步流量	同步、异步报文均无报文丢失	通过
		24%异步流量	同步、异步报文均无报文丢失	通过
		50%异步流量	没有同步报文丢失	通过
		80%异步流量	没有同步报文丢失	通过
		100%异步流量	没有同步报文丢失	通过
MAC 地址自学习 (Fig.2.a)	80%链路负载		第一步，报文被广播，没有报文丢失	通过
			第二步，报文被单播，没有报文丢失	通过
MAC 地址超时删除 (Fig.2.a)	80%链路负载		1 分钟后 MAC 地址老化删除	通过
时间同步 (Fig.2.b)	无链路负载		时间同步误差需小于500ns	测试后误差小于 100ns，通过

两个板子级联时同步报文测试时的数据：

1. 级联时通过千兆流量线速测试

测试配置：随机报文，随机大小的同步报文。测试仪的端口1与端口3之间进行千兆线速双向流量测试，测试报文在1.3亿个报文以上。

测试结果：报文通过率为97.94%>95%，具体测试数据见下图。

两个板子级联时混合流量测试时的数据：

1. 70% 同步流量 + 20% 异步流量

测试配置: 测试仪从端口2发送70%的同步报文流量到端口4进行测试，与此同时，测试仪从端口1发送20%的异步报文流量到端口3进行测试，测试的同步报文数目在1亿个报文以上

测试结果：同步报文与异步报文均无报文丢失，具体测试数据见下图。

2. 70% 同步流量 + 100% 异步流量

测试配置: 测试仪从端口2发送70%的同步报文流量到端口4进行测试，与此同时，测试仪从端口1发送100%的异步报文流量到端口3进行测试，测试的同步报文数目在1亿个报文以上。

测试结果：同步报文无报文丢失，具体测试数据见下图。

表2总结了异步交换机性能的测试结果，在表2的后面，给出了部分测试场景下的详细测试配置与测试结果

表 2： 片外异步交换机的测试结果

单跳或级联测试		期望的结果	测试结果：是否通过
异步报文通过率的测试	单跳(Fig.2.a)	100%异步报文通过	通过
	级联(Fig.2.b)	100%异步报文通过	通过
MAC 地址自学习 (Fig.2.a)	80%链路负载	第一步，报文被广播，没有报文丢失	通过
		第二步，报文被单播，没有报文丢失	通过
MAC 地址超时删除 (Fig.2.a)	80%链路负载	1 分钟后 MAC 地址老化删除	通过

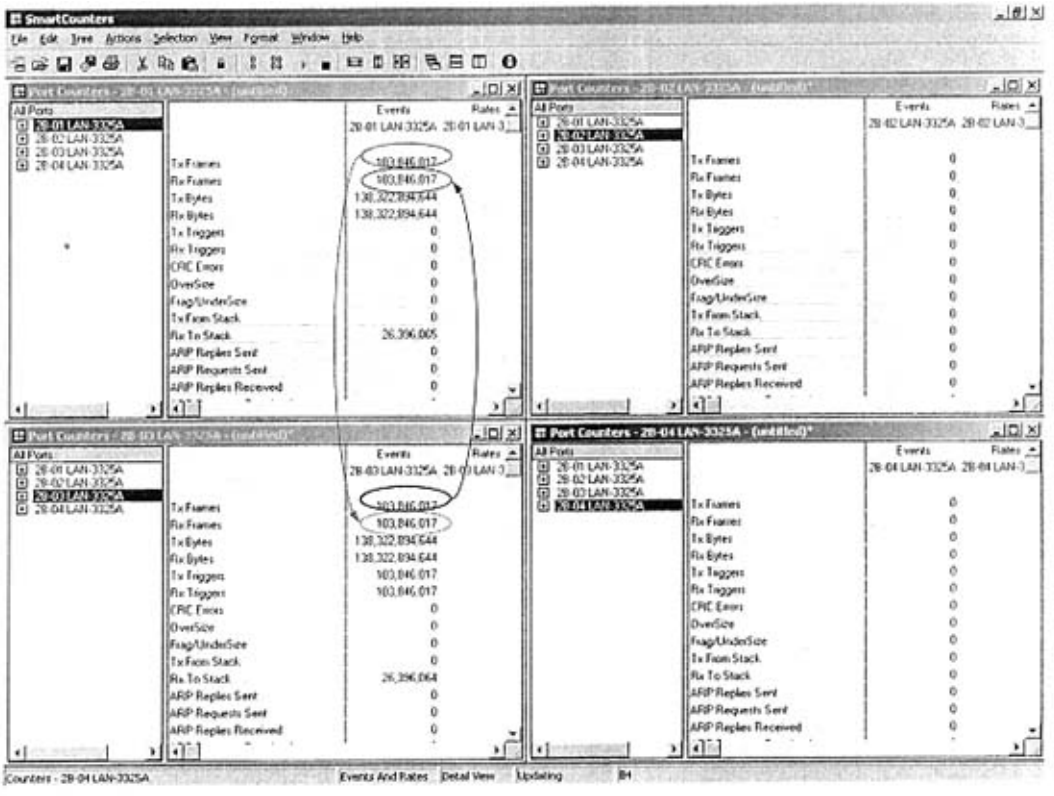
下面是给出了异步交换机测试时部分测试场景下的详细测试配置与测试结果。

两个板子级联时异步流量测试数据：

1. 100% 异步流量

测试配置：随机报文，随机大小的同步报文。测试仪的端口1与端口3之间进行千兆线速双向异步流量测试，测试报文在1亿个报文以上。

测试结果：异步报文无报文丢失，具体测试数据见下图。



2.3 第二阶段工作总结

第二阶段工作的持续时间是从 2005.8 到 2005.12。这期间项目组共五名成员，我担任项目经理，其他四名成员是郑剑锋博士、谭兴晔博士、吴起博士与毕务刚。

2.3.1 系统设计需求

功能需求：

1. 修改报文分析模块，将同步报文发送到同步交换阵；将异步报文发送到

异步交换阵；将协议控制报文存储到内部存储器，供 CPU 读取；

2. 修改报文复用模块，将异步报文由以前的从外部 GMII 接口输入修改成从异步交换阵输出；其它两路报文输入部分的接口不变；同步报文依然由同步交换阵输出；协议控制报文依然由内部存储器输出；
3. 优化软件平台，包括以下方面：
   a) 优化驱动与应用程序的设计框架；
   b) 将 CPU 轮询机制的设计优化成中断机制；
4. 修改 FPGA 代码增加一套 5X5 千兆交换阵，新增加的交换阵需要完成：
   a) 完成异步报文 MAC 地址的自学习功能；
   b) 完成异步报文 MAC 地址超时自动删除功能；
   c) 对异步报文交换的性能需要达到 5 路千兆输入内部交换无阻塞；
5. 需要完成以上模块的功能仿真、单元测试、集成测试。

性能需求：

1. 千兆线速情况下异步报文通过率要求达到 100%；
2. 千兆线速情况下同步报文通过率要求达到 95%以上；
3. 异步报文与同步报文混合情况下：
   a) 70%同步报文＋20%异步报文，要求两种报文均不能出现报文丢失；
   b) 70%同步报文＋30%异步报文，要求同步报文不能出现报文丢失；
   c) 70%同步报文＋50%异步报文，要求同步报文不能出现报文丢失；
   d) 70%同步报文＋100%异步报文，要求同步报文不能出现报文丢失；

2.3.2 系统设计描述

系统设计的总体模块框图如下图所示。

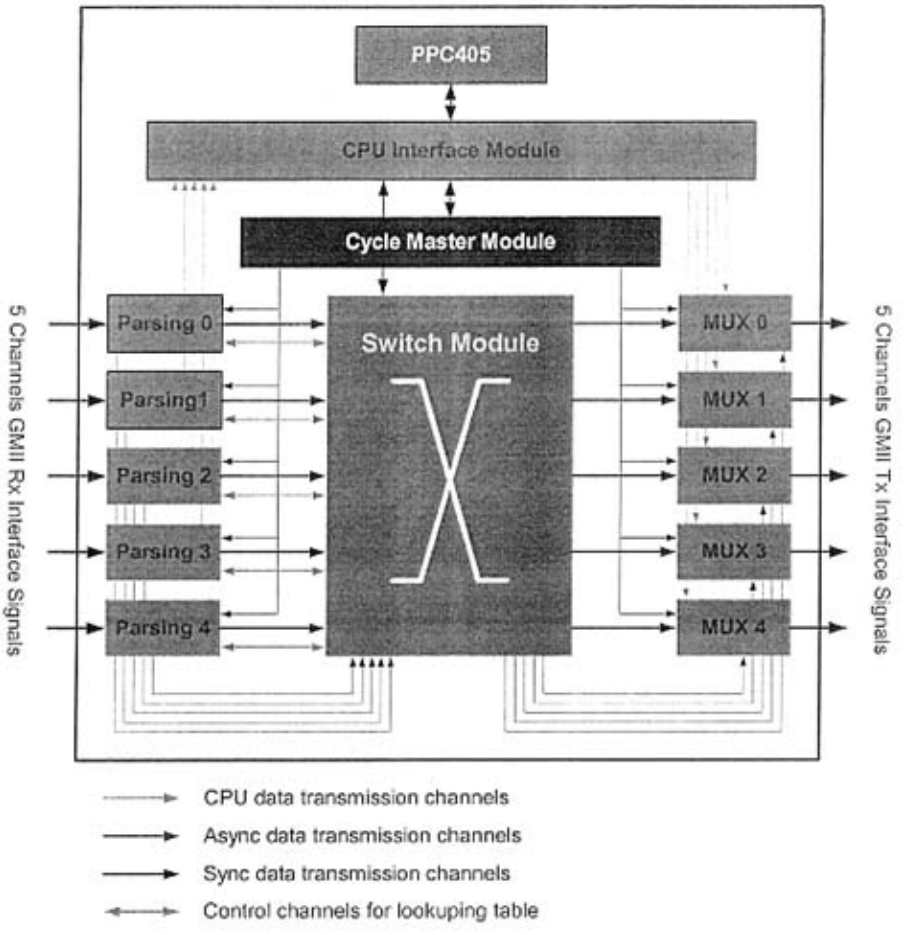


图 3 总体模块框图

从上图可以看出，FPGA 设计包括主要以下模块：

- 报文分析模块—Parsing 模块
- 报文复用模块—MUX 模块
- 交换阵模块—Switch 模块，它包括同步交换阵模块与异步交换阵模块
- 时间同步模块—Cycle Master 模块
- CPU 接口模块—CPU interface 模块
- CPU core 处理模块—PPC405 模块

从上图可以看出，FPGA 系统主要包括 6 个模块。下面我们将描述我们为何需要设计以上几个模块。

报文从物理线路进行 RJ45 接口后，经物理层芯片处理后变成标准的 GMII 接口送到 FPGA 芯片。这些报文首先需要被 Parsing 模块处理。Parsing 模块根据

报文中携带的 Ethernet 类型域携带的类型关键字将以太网报文划分成三种类型：同步报文、异步报文与协议控制报文。然后，同步报文将根据目的 MAC 地址查表结果发送到同步交换阵模块；异步报文将根据目的 MAC 地址查表结果发送到异步交换阵模块；协议控制报文，主要包括时间同步协议报文与设备能力广播报文，将被 Parsing 模块发送到内部存储器内；如果该协议控制报文是时间同步协议报文，将打上该报文到达时刻的系统时间标签供应用层软件使用。Parsing 模块还将对所有的报文进行 CRC 检查，如果发现错误，将报告错误结果。

交换阵模块将区别同步报文与异步报文进行处理。对于同步报文，同步交换阵模块将根据同步报文的目的 MAC 地址的查表结果将报文交换到相应的输出端口，如果查表结果没有匹配项，将该报文广播其它端口。对于异步报文，异步交换阵模块将根据异步报文的目的 MAC 地址的查表结果将报文交换到相应的输出端口，如果查表结果没有匹配项，将该报文广播其它端口。交换阵需要支持 MAC 地址自动学习、自动超时更新与静态路由配置功能，交换阵维护的转发表将提供给同步报文与异步报文一起使用。

CPU 接口模块用于适配 PLB 总线信号，并对地址信号进行译码，产生相应的读写控制信号来访问报文复用模块、交换阵模块以及时间同步模块。

报文复用模块有三个数据来源，它们是：来自异步交换阵的异步报文；来自同步交换阵的同步报文以及来自 CPU 接口模块的协议处理报文。报文复用模块将根据超帧的要求将这三种报文复用到 GMII 接口上。

SoC 模块将要检测超帧的抖动是否在系统允许的范围内，如果超限，将产生超帧抖动超限指示信号指导随后处理。此外，SoC 模块还将根据时间同步模块输出的时间信息产生周期为 125us 的超帧起始信号，并将该超帧起始信号发送到报文复用模块，报文复用模块将根据超帧起始信号按照超帧格式向 GMII 接口上发送报文。至于超帧脉冲产生模块，由于在重用它，所以在框图中没有给出。

时间同步模块将维护时间计数器，并根据 CPU 设定的控制寄存器进行时间调整，以达到时间时间同步。时间同步模块将给报文分析模块提供时间信息。此外，时间同步模块将根据报文复用模块产生的 SFD 指示信号锁存当前系统的时戳，以供应用层软件读取使用。它的功能基本没有变化。

2.3.3 FPGA 仿真结果

仿真测试结论：通过韩国总部要求的所有测试样例。

下表给出了各种场景下 FPGA 仿真的结果。

测试配置单跳或级联测试			期望的结果	是否通过测试
异步报文测试			100%异步报文通过	通过
同步报文测试			同步报文通过率为 95%以上	通过率为 97.9% 通过
同步异步混合流量测试	70%同步流量	20%异步流量	同步、异步报文均无报文丢失	通过
		50 % 异步流量	没有同步报文丢失	通过
		80%异步流量	没有同步报文丢失	通过
		100%异步流量	没有同步报文丢失	通过
MAC 地址自学习(Fig.2.a)	80%链路负载		第一步，报文被广播，没有报文丢失	通过
			第二步，报文被单播，没有报文丢失	通过

下面我们将给出部分场景下仿真配置与仿真结果。

异步流量测试数据：

1. 100% 异步流量的仿真

仿真持续时间： 6400 us

仿真流量产生时间： 6000 us

输入激励：

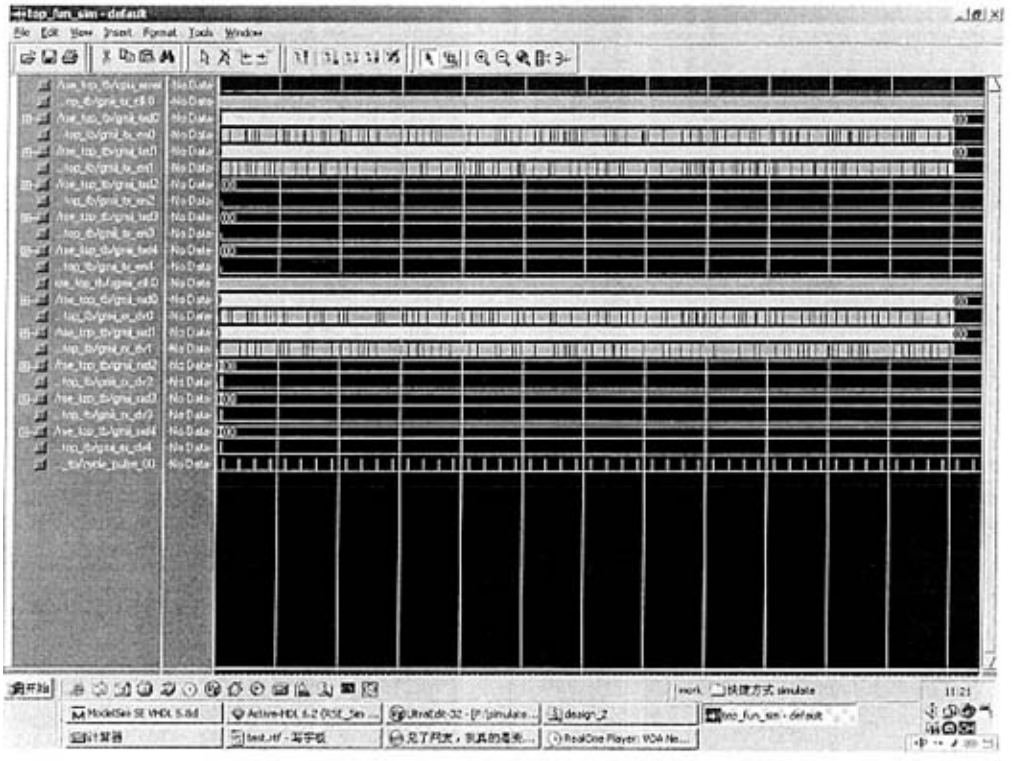
源端口号	报文类型	流量特征	目的端口	报文发送个数
端口-0	异步	随机变长内容	端口-1	925
端口-1	异步	随机变长内容	端口-0	932
端口-2	同步	随机变长内容	端口-1	0

端口-3	异步	随机变长内容	端口-4	0
端口-4	异步	随机变长内容	端口-3	0

仿真结果：

端口号	收到的报文总数	收到的同步报文数	收到的异步报文数
端口-0	932	0	932
端口-1	925	0	925
端口-2	2	0	2
端口-3	2	0	2
端口-4	2	0	2

仿真结论：异步报文无报文丢失，通过。仿真波形图见下图。



2. 100% 同步流量的仿真

仿真持续时间： 15400 us

仿真流量产生时间： 15000 us

输入激励：

源端口号	报文类型	流量特征	目的端口	报文发送个数
端口-0	同步	随机变长内容	端口-1	2312
端口-1	同步	随机变长内容	端口-0	2316
端口-2	同步	随机变长内容	端口-1	0
端口-3	异步	随机变长内容	端口-4	0
端口-4	同步	随机变长内容	端口-3	0

仿真结果：

端口号	收到的报文总数	收到的同步报文数	收到的异步报文数
端口-0	2305	2305	0
端口-1	2303	2303	0
端口-2	2	2	0
端口-3	2	2	0
端口-4	2	2	0

仿真结论：同步报文通过率为99.5%>95%，通过。

3. 70% 同步流量＋20%异步流量混合仿真

仿真持续时间： 5400 us

仿真流量产生时间： 5000 us

输入激励：

源端口号	报文类型	流量特征	目的端口	报文发送个数
端口-0	同步	随机变长内容	端口-1	614
端口-1	异步	随机变长内容	端口-2	776
端口-2	异步	随机变长内容	端口-1	306
端口-3	异步	随机变长内容	端口-4	0
端口-4	同步	随机变长内容	端口-3	0

仿真结果：

端口号	收到的报文总数	收到的同步报文数	收到的异步报文数
端口-0	3	0	3

端口-1	920	614	306
端口-2	776	0	776
端口-3	3	0	3
端口-4	3	0	3

仿真结论：同步报文与异步报文均无报文丢失，通过。

4. 70% 同步流量＋100%异步流量混合仿真

仿真持续时间： 5400 us

仿真流量产生时间： 5000 us

输入激励：

源端口号	报文类型	流量特征	目的端口	报文发送个数
端口-0	同步	随机变长内容	端口-1	614
端口-1	异步	随机变长内容	端口-2	776
端口-2	异步	随机变长内容	端口-1	782
端口-3	异步	随机变长内容	端口-4	0
端口-4	同步	随机变长内容	端口-3	0

仿真结果：

端口号	收到的报文总数	收到的同步报文数	收到的异步报文数
端口-0	3	0	3
端口-1	946	614	332
端口-2	776	0	776
端口-3	3	0	3
端口-4	3	0	3

仿真结论：同步报文无报文丢失，通过。

2.4 第三阶段工作总结

第三阶段工作的持续时间是从2006.1到2006.5。这期间项目组共两名成员，由我与郑剑锋博士组成。

2.4.1 系统设计需求

功能需求：

1. 修改报文分析模块，去掉同交换阵的接口，将协议控制报文存储到内部存储器，供 CPU 读取；
2. 修改报文复用模块，去掉同交换阵的接口；协议控制报文依然由内部存储器输出；
3. 实现韩国总部关于改进时间同步算法的专利思想，若该专利实现后性能达不到预期的性能，则需要提出新的专利思想；实现需要软件与 FPGA 协同设计，主要包括：
   a) 软件方面：软件需要移植并测试第二阶段的代码；修改应用层代码，实现改进时间同步的专利思想；
   b) FPGA 方面：需要修改时间同步模块，使用新的方法实现时间计数与同步调整；
4. 需要完成以上模块的功能仿真、单元测试、系统测试；
5. 跟踪中国标准，若有重要发现，报告韩国总部。

性能需求：

1. 两个设备之间的时间同步的误差要求在从[-500ns, 500ns]范围修改成[-150ns, 150ns]以内；
2. n 个设备级联后的时间同步误差要求：
   a) 具体指标测试时再定；
   b) 不能与级联数目 n 成指数增长的关系；

2.4.2 系统设计描述

系统设计的总体模块框图如下图所示。

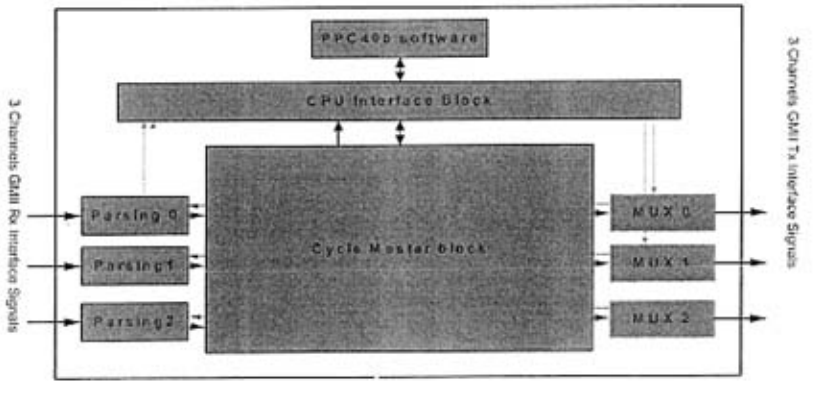


图 4 总体模块框图

从上图可以看出，FPGA 设计包括以下模块：

- 报文分析模块—Parsing 模块
- 报文复用模块—MUX 模块
- 时间同步模块—Cycle Master 模块
- CPU 接口模块—CPU interface 模块
- CPU core 处理模块—PPC405 模块

从上图可以看出，FPGA 系统主要包括 5 个模块。下面我们将描述我们为何需要设计以上几个模块。

报文从物理线路进行 RJ45 接口后，经物理层芯片处理后变成标准的 GMII 接口送到 FPGA 芯片。这些报文首先需要被 Parsing 模块处理。Parsing 模块根据报文中携带的 Ethernet 类型域携带的类型关键字将以太网报文划分成两种类型：时间同步协议报文和设备能力广播报文。Parsing 模块对两种报文均进行 CRC 检查，并将检查结果附在报文尾部提交给应用层软件。对于这两种报文处理的唯一区别在于：对于时间同步相关的协议报文，Parsing 模块还将该报文到达时刻的系统时戳附在报文尾部提交给应用层软件分析处理。

CPU 接口模块用于适配 PLB 总线信号，并对地址信号进行译码，产生相应的读写控制信号来访问报文复用模块、以及时间同步模块。

报文复用模块只有一个数据来源，它们是：自 CPU 接口模块的时间同步协议报文和设备能力广播报文。报文复用模块直接将这两种报文发送到 GMII 接口上，不再遵从超帧结构。

时间同步模块将维护时间计数器，并根据 CPU 设定的控制寄存器进行时间调整，以达到时间时间同步。时间同步模块将给报文分析模块提供时间信息。此外，时间同步模块将根据报文复用模块产生的 SFD 指示信号锁存当前系统的时戳，以供应用层软件读取使用。它的功能基本没有变化，但需要重新设计实现的架构。

2.4.3 系统测试结果

系统测试的网络连接图如图 5 所示。其中，设备 1 是主同步设备，其它六个设备均同步与它。同步的周期是 100ms。同步过程如下：设备 1（主同步设备）

每隔 100ms 向设备 2 的从端口发送 PTP 协议报文，设备 2 运行改进算法，实现与主同步设备之间的时间同步；在设备 2 完成与主同步设备同步之后，也以 100ms 时间间隔向设备 3 的从端口发送 PTP 协议报文，设备 3 运行传统算法或改进算法，实现与设备 2 的同步。依此类推，设备 7 也实现与设备 6 之间的同步。

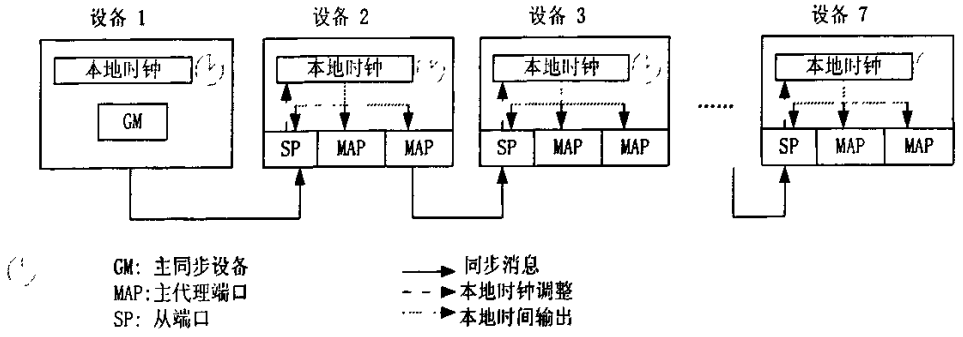


图 5 系统测试的网络连接图

测试过程中我们记录了时间同步误差的峰峰值（Pk-Pk），以 ns 为单位。我们对传统 FCC 算法、韩国总部提出的算法（简称为 KHQ 算法）与我们提出的算法（简称为 BST 算法）分别进行了测试。

传统算法的测试结果：

(Hop 0 是主同步设备)	同步误差的测试结果 (ns)
Hop 1	[-45, 50]
Hop 2	[-120, 160]
Hop 3	[-340, 370]
Hop 4	[-1000, 900]
Hop 5	[-2800, 2700]
Hop 6	[-8000, 8800]
Hop 7	[-20000,20000]

图 6 中通道 2、3、4 分别给出了 hop 1，hop 2 与 hop 3 的测试波形；图 7 中通道 2、3、4 分别给出了 hop 4，hop 5 与 hop 6 的测试波形；图 8 中通道 2 给出了 hop 7 的测试波形。

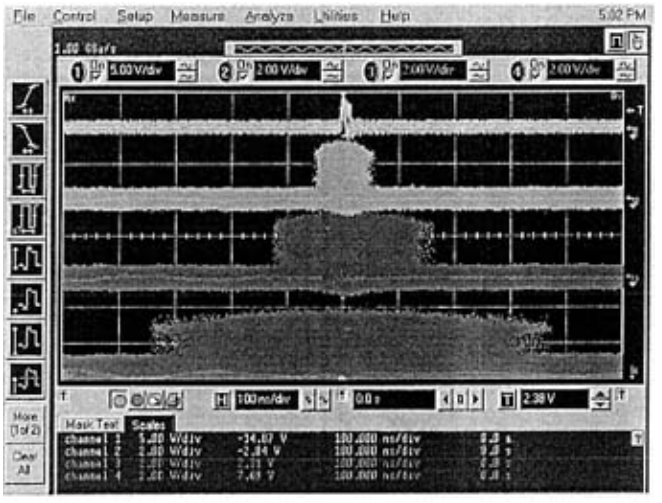


图 6 hop 1、2、3 的测试波形

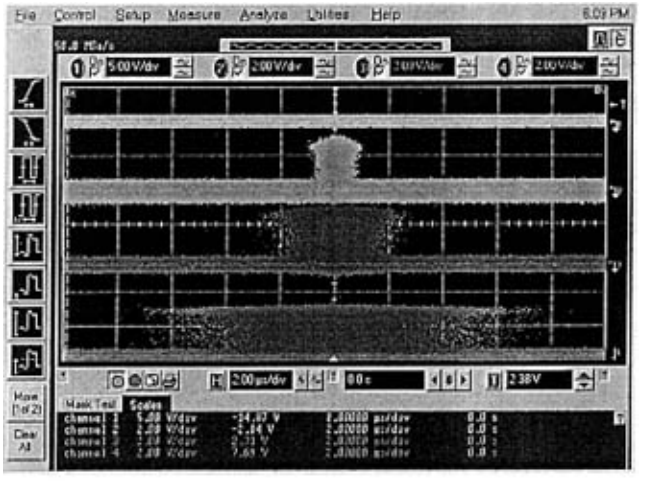


图 7 hop 4、5、6 的测试波形

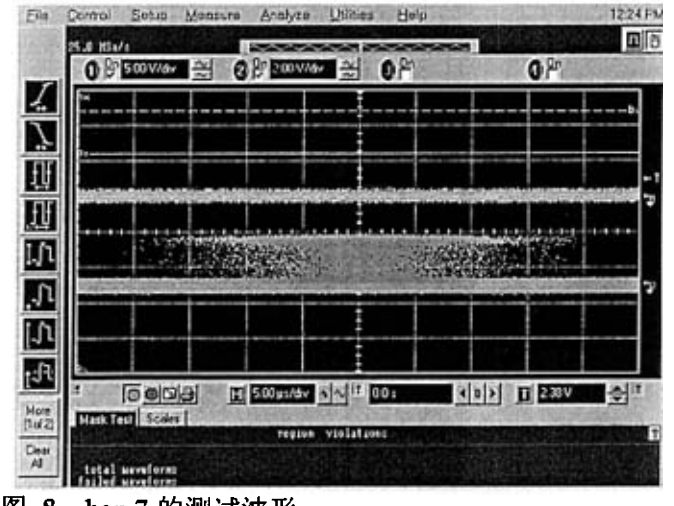


图 8 hop 7 的测试波形

KHQ 算法的测试结果：

(Hop 0 是主同步设备)	同步误差的测试结果 (ns)
Hop 1	[-35, 48]
Hop 2	[-65, 100]
Hop 3	[-85, 120]
Hop 4	[-75, 120]
Hop 5	[-115, 130]
Hop 6	[-135, 175]
Hop 7	[-210, 240]

图 9 中通道 2、3、4 分别给出了 hop 1，hop 2 与 hop 3 的测试波形；图 10 中通道 2、3、4 分别给出了 hop 4，hop 5 与 hop 6 的测试波形；图 11 中通道 2 给出了 hop 7 的测试波形。

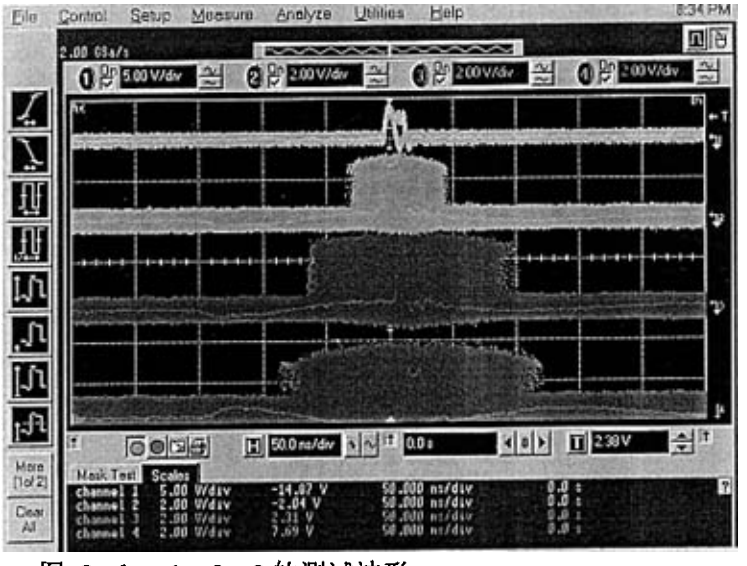


图 9 hop 1、2、3 的测试波形

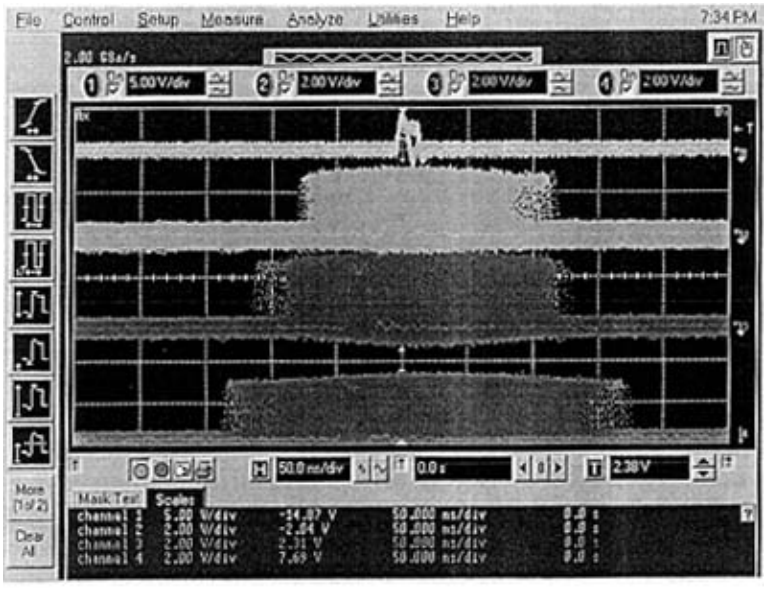


图 10 hop 4、5、6 的测试波形

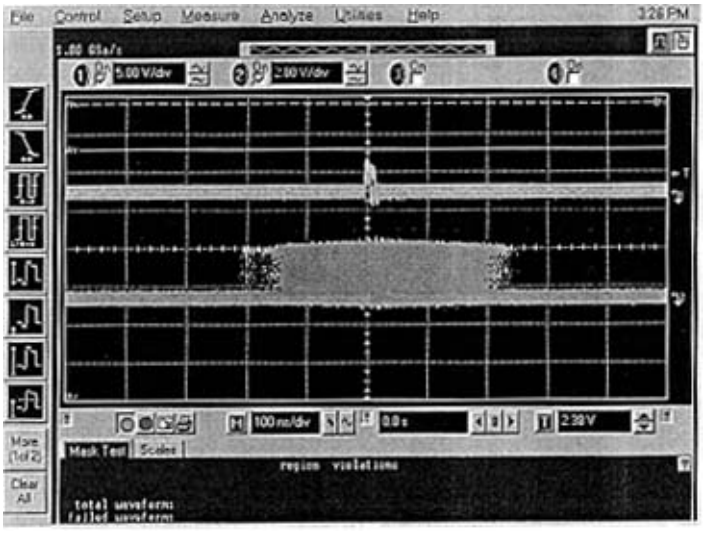


图 11 hop 7 的测试波形

BST 算法的测试结果：

(Hop 0 是主同步设备)	同步误差的测试结果 (ns)
Hop 1	[-20, 35]
Hop 2	[-25, 50]
Hop 3	[-30,50]

Hop 4	[-40, 60]
Hop 5	[-45, 70]
Hop 6	[-60, 75]
Hop 7	[-30, 120]

图 12 中通道 2、3、4 分别给出了 hop 1，hop 2 与 hop 3 的测试波形；图 13 中通道 2、3、4 分别给出了 hop 4，hop 5 与 hop 6 的测试波形；图 14 中通道 2 给出了 hop 7 的测试波形。

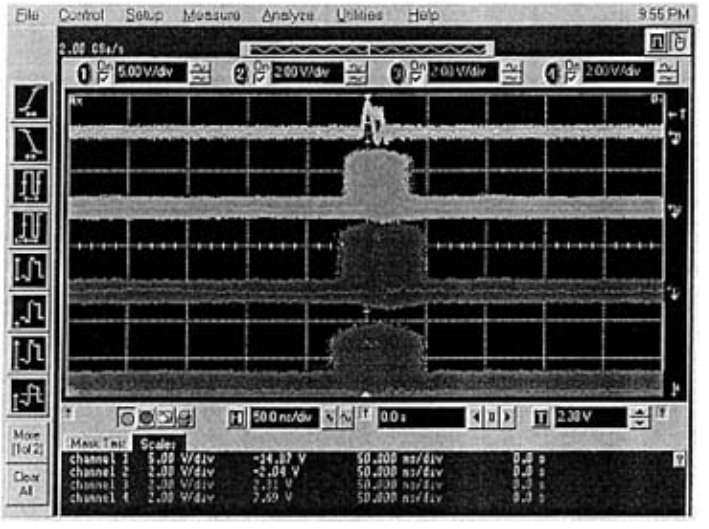


图 12 hop 1、2、3 的测试波形

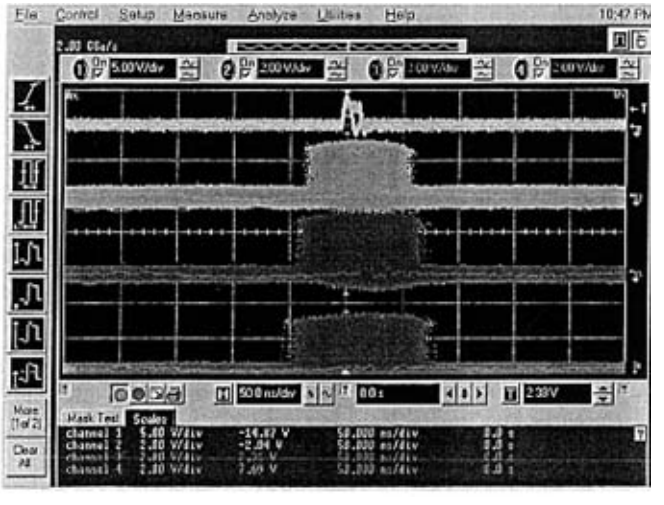


图 13 hop 4、5、6 的测试波形

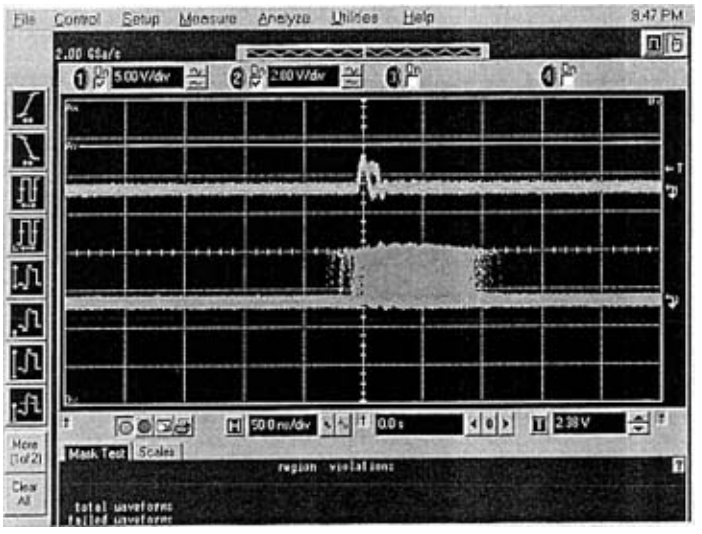


图 14 hop 7 的测试波形

图 15 给出了三种算法结果的比较曲线；图 16 给出了其中两种算法结果的比较曲线。

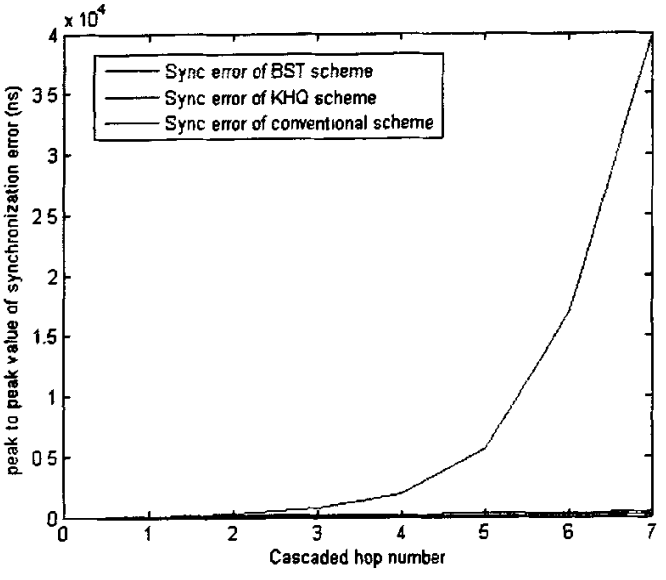


图 15 三种算法结果的比较曲线

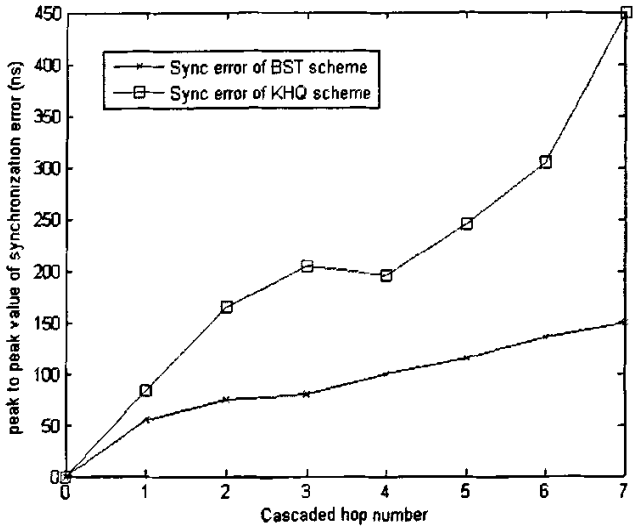


图 16 KHQ 算法与 BST 算法结果的比较曲线

从以上测试结果可以看出，KHQ 算法与 BST 算法明显优于传统 FCC 算法的性能，他们有效解决了多个设备级联时同步误差随级联数目成指数分布的问题；而 BST 算法又稍优于 KHQ 算法。

2.5 本章小结

本章首先对博士后课题的开发工作进行了概述，然后对开发平台、开发环境、使用的操作系统、研发工具与测试设备进行了介绍；然后对三个阶段的研发工作分别从研发需求、系统模块设计与测试结果这几个方面进行了描述。这三个阶段的开发工作全部通过了韩国总部要求的验收测试，得到了非常高的评价。本文作者也因为项目组的卓越表现并评为 2005 年度优秀模范员工。

第三章 研究工作的总结

3.1 概述

2004.11 至 2006.5 月份这段工作期间项目组共发明五项专利，前四项专利是本文作者作为第二发明人提出，第一发明人是吴起博士；第五项专利是本文作者作为第一发明人提出。这五项专利发明是：

1. 一种单播方式的以太网多播控制信息传递方法
2. 综合多播注册和资源预留方法
3. 以太网网桥之间的协同调度方法
4. 自适应的多跳时分复用调度算法
5. 多级设备之间时间同步 FCC 补偿方法的改进算法

以下章节将分别描述上述的 5 项专利发明的主要内容。

3.2 专利一：一种单播方式的以太网多播控制信息传递方法

3.2.1 专利提出的背景

发明背景介绍

a. 发明的技术领域

本发明涉及计算机和通信网络中的多播问题，特别是介质访问子层的多播。如 IEEE 802.3 以太网的多播注册。

b. 现有技术的说明

现有的数据链路层多播解决方案主要有 IEEE 802.1D[3]中规定的多播注册协议 GMRP（GARP Multicast Management Protocol）和 Cisco 公司的 CGMP(Cisco Group Management Protocol)，其中 GMRP 基于 IEEE 802.1D 中定义的通用属性注册协议 GARP(Generic Attribute Registration Protocol)，它的运行过程如下：

想要注册成为某个多播组 G 成员的设备 D 生成一个 GMRP 报文，并把该报文发到它所在的局域网上，其中

- 报文的目的地址为GMRP多播地址01-80-C2-00-00-20.
- 报文包含多播组G的介质访问子层(Medium Access Control, MAC)地址。

- 当设备D所在局域网的网桥收到该GMRP报文之后，把收到该报文的端口标记为多播组G的转发端口，并把该报文向其他所有处于活动状态的端口转发。

其它网桥也作类似处理，这样，该 GMRP 报文就类似广播报文那样被发到整个局域网上。

CGMP 是 Cisco 公司提出的，用于 Cisco 网桥和路由器之间交换二层和三层多播注册信息的协议。当 Cisco 路由器收到三层的多播注册消息后，用 CGMP 协议通知 Cisco 的交换机，这样建立三层多播组的同时也在二层建立了多播组。

c. 现有技术问题或要改善的地方

目前介质访问子层的多播技术，无论是 GMRP 还是 CGMP，都使用多播的方式来传送多播的控制信息，这种方式主要有下面两个缺陷：

该方式有可能造成控制信息无法到达某些设备，或者是造成在全网范围内广播。具体哪种效果依赖于具体的实现。对于 GMRP 和 CGMP 来说，其控制信息实质上就是在全网范围内进行广播，从而造成资源的浪费。

该方式不利于设备的高效处理。众所周知，网络中间设备，无论是二层的网桥还是三层的路由器，都在设计时采用控制信息和数据转发分开处理的方式，其中数据转发部分做了大量优化，以使得设备具有较好的转发性能。使用多播作为控制信息的载体将使得设备难以区分多播控制信息和数据流，从而造成处理上的低效。

3.2.2 专利思想的描述

本发明提出了使用单播方式传递以太网的多播控制信息。和传统的多播方式传送的多播控制信息相比，本发明具有两大好处：第一、由于消息是单播方式传送，因此它不会被传递到全网。同时，本发明的控制消息只到达那些必须知道有新加入者的设备，从而避免了资源的浪费。第二、网桥可以很容易地区分多播控制信息和数据流，这样就使得网桥能够高效地对两者进行分别处理。该方法包括以下几个部分。

目的 MAC 地址的选择

对于一对多的多播（即只有一个发送者，多个接受者），很自然的选择该发送者的地址作为目的 MAC 地址。本发明不提供对多个发送者，多个接受者多播的支持。

和其它单播流量的区分

为了使网桥能够快速识别并处理单播方式传送的多播控制信息，我们定义了一种特殊的以太网帧类型，0x7498。对于那些认识该类型的网桥来说，该类型的帧被送到 CPU 中进行多播注册或离开操作，对于不认识该类型的网桥来说，该帧被当成一个普通的单播帧进行转发处理，从而避免了使用多播方式引起的广播。

多播注册信息（可以给出消息的详细结构）

目前主要有两种多播控制信息。多播注册信息 join 和多播离开信息 leave，这两个消息中均包含想要加入或离开多播组的地址。其中多播消息的接收者或网桥使用 join 消息来通知上游网桥或多播消息的发送者自己想要加入消息中包含的多播组；多播消息的接收者或网桥使用 leave 消息来通知上游网桥或多播消息的发送者自己想要离开消息中包含的多播组。

网桥处理单播方式的多播控制信息

处理多播注册信息 join

当一个网桥收到 join 消息以后，它首先检测自己有没有和 join 消息中的多播组对应的多播（也可以给出多播表项的详细定义）。如果表项不存在，则该网桥为该多播组创建一个表项。表项中的入口端口为该 join 消息的目的 MAC 地址所对应的端口，表项的出口地址列表包含且只包含接收到 join 消息的端口。由于该网桥在接收到 join 消息之前没有对应的表项，因此它不是多播树上的节点。也就是说，只是该网桥创建表项不足以使得接收者连接到多播树上，也就是说，多播数据流无法顺利地到达接收者。为了使得接收者能够连接到多播树上，该网桥必须把 join 消息继续向发送者转发，以使得该消息能够到达多播树上的某个节点，从而为该接收者创建一条新的分支。

如果网桥有对应的表项，则说明它是多播树上的成员。该网桥把新接收到 join 消息的端口添加到对应表项的出口端口列表中。由于添加了该端口相当于在多播树上创建了一条到达新加入接收者的分支，即接收者已经连接到多播树上。因此该网桥无需转发 join 消息。

处理多播离开信息 leave

当一个网桥收到 leave 消息以后，它首先检检测自己有没有和 join 消息中的多播组对应的多播表项。如果表项不存在，则说明该网桥目前不在多播树上。也就是说，该 leave 消息和该网桥没有什么关系，因此丢弃该 leave 消息。

如果该网桥有对应的表项，则把接收到 leave 消息的端口从该表项的转发端口列表中删除。删除后，如果转发端口列表为空，则说明没有设备需要该网桥为其转发多播报文了，因此，该网桥把该表项删除，并把 leave 消息继续向多播发送者的方向转发。

发送者处理单播方式的多播控制信息

处理多播注册信息 join

多播的发送者处理 join 消息和网桥处理 join 消息基本相同。由于发送者是多播树的根节点，因此它总是多播树的成员。在接收到 join 消息后，发送者总是把新接收到 join 消息的端口添加到对应表项的出口端口列表中，并丢弃该消息。

处理多播离开信息 leave

多播的发送者处理 leave 消息和网桥处理 leave 消息基本相同。由于发送者是多播树的根节点，因此它总是有对应的表项。在接收到 leave 消息后，把接收到 leave 消息的端口从该表项的转发端口列表中删除。

注意发送者和网桥处理 leave 消息的最大区别在于：当发送者发现转发端口列表为空时，并不删除表项。这是由于发送者必须总是处于能够接收新的设备加入的状态。另外，由于发送者即是 leave 消息的目的节点，因此发送者无需转发 leave 消息。

3.3 专利二：综合多播注册和资源预留方法

3.3.1 专利提出的背景

1. 发明背景介绍

a. 发明的技术领域

本发明涉及计算机和通信网络中的多播注册和资源预留问题，特别是介质访问子层的多播。如 IEEE 802.3 以太网的多播注册和资源预留。

b. 现有技术的说明

现有的以太网多播协议主要有 IEEE 802.1D 中规定的多播注册协议 GMRP(GARP Multicast Management Protocol)和 Cisco 公司的 CGMP(Cisco Group Management Protocol)，其中 GMRP 基于 IEEE 802.1D 中定义的通用属性注册协议 GARP(Generic Attribute Registration Protocol)，它的运行过程如下：

想要注册成为某个多播组 G 成员的设备 D 生成一个 GMRP 报文，并把该报文发到它所在的局域网上，其中

- 报文的目的地址为GMRP多播地址01-80-C2-00-00-20.
- 报文包含多播组G的介质访问子层(Medium Access Control, MAC)地址。
- 当设备D所在局域网的网桥收到该GMRP报文之后，把收到该报文的端口标记为多播组G的转发端口，并把该报文向其他所有处于活动状态的端口转发。

其它网桥也作类似处理，这样，该 GMRP 报文就类似广播报文那样被发到整个局域网上。

CGMP 是 Cisco 公司提出的，用于 Cisco 网桥和路由器之间交换二层和三层多播注册信息的协议。当 Cisco 路由器收到三层的多播注册消息后，用 CGMP 协议通知 Cisco 的交换机，这样建立三层多播组的同时也在二层建立了多播组。

GMRP 和 CGMP 只提供多播注册，不提供资源预留。

IEEE 的驻地以太网 RSE(Residential Ethernet)研究小组最近提出了简单预留协议 SRP (Simple Reservation Protocol)，该协议的主要目的是为同步流提供资源预留。本专利和 SRP 的主要不同之处在于，SRP 在多播注册的时候不进行资源是否满足需求的检查，只有发送者开始发送数据之后才知道带宽是否不足；本专利在多播注册的时候同时进行资源检查，只有当资源满足需求时才进行多播注册。

c. 现有技术问题或要改善的地方

只包含多播注册的技术，如 GMRP 和 CGMP 等，均不能提供服务质量的保证，由于目前越来越多的应用如视频播放、VoIP 有着严格的带宽或延迟要求，因此在以太网中提供服务质量保证势在必行。

简单预留协议 SRP 在多播注册的时候不进行资源是否满足需求的检查，只有发送者开始发送数据之后才知道带宽是否不足，这样会造成网络状态的不一致性。对于有服务质量需求的流来说，一旦被接纳，则说明该流的服务质量一定能够得到保证。而 SRP 协议不具有这种特征，即使多播请求被接纳，也不一定能够提供相应的质量保证，这使得 SRP 协议不适合为以太网提供服务质量保证。

3.3.2 专利思想的描述

为了解决以太网中多播服务质量的保证问题，我们提出了一种新的综合多播和资源预留的解决方案。该方案中，接收者可以通过发送单个消息来达到多播注

册和资源预留的目的，从而能够快速，高效地提供多播服务质量的支持。

a. 发明的构成

为了在以太网中提供服务质量保证，我们提出了新的综合多播和资源预留的方法，该方法包括以下几个部分。

以太网网桥的多播变量定义

网桥使用多播表项来记录多播组的注册和资源预留信息。每一个表项所包含的变量如下：

- ✓ 接收端口：即该多播组的上游网桥所对应的端口。上游的控制信息和多播数据都应该来自接收端口。
- ✓ 转发端口列表：一组端口。当网桥接收到来自上游的多播数据时，应该把该数据向转发端口列表中的所有端口进行转发。
- ✓ 刷新定时器(RTimer)：控制着网桥向上游定期发送刷新消息的定时器。当RTimer超时后，网桥向上游发送刷新消息。
- ✓ 状态：表明该表项当前所处的状态。其中
  - ■ “等待状态”表明网桥已经为该多播组建立了多播表项，但它还没有收到来自上游网桥的加入确认信息；
  - ■ “完成状态”表明网桥已经为该多播组建立了多播表项，且它已经收到来自上游网桥的加入确认信息。
- ✓ 转发端口列表中的每个端口也对应着一个定时器，即端口超时定时器(PTimer)。当PTimer超时后，网桥则把该端口从转发端口列表中删除。

网络中传递的消息定义

多播加入消息

接收者使用多播加入消息来通知网桥和发送者它想要接收某类多播数据。除此外，接收者和网桥还定期发送多播消息，从而保持上游网桥和发送者的状态。

除了要加入的多播组的信息外，多播加入消息中还包括资源请求信息，以便网桥或发送者能够根据该信息进行资源预留。

多播离开消息

接收者和网桥使用多播离开消息通知上游网桥和发送者它们不想再接收某类多播数据。

肯定应答消息

发送者或网桥使用肯定应答消息通知接收者已经成功加入多播组

否定应答消息

发送者或网桥使用否定应答消息通知接收者加入多播组失败

以太网网桥的行为定义

处理多播加入消息

当网桥接收到多播加入消息后，首先检测自己是否有和该消息对应的表项。如果没有找到对应的表项，或者接收到多播加入消息的端口不在该表项的转发端口列表中，则网桥进一步检测是否有足够的资源满足多播加入消息的资源请求。根据检测结果，网桥做如下操作：

如果网桥没有足够的资源满足多播加入消息的资源请求，则该网桥生成一个否定应答消息，把该消息发往多播加入消息的发送者，并丢弃加入消息。

如果网桥有足够的资源，则该网桥的行为取决于网桥是否有对应表项，以及该表项的状态，具体的来说：

- 如果网桥没有对应表项，则创建一个表项，其中表项的状态为“等待”，表项的接收端口为多播加入消息的目的 MAC 地址对应端口，表项的转发端口列表中包含且只包含接收多播加入消息的端口。表项创建以后，网桥为转发端口列表中的每个端口启动一个定时器(PTimer)，并在端口为该多播组保留资源。最后，网桥把该多播加入消息向多播加入消息的目的 MAC 地址对应端口进行转发。
- 如果网桥包含对应表项，且该表项的状态为“等待”，则该网桥检查在该表项的转发端口列表是否已经包含接收多播加入消息的端口。如果不包含，则网桥把该端口加入到转发端口列表中，启动该端口的定时器，并在端口为该多播组保留资源；如果包含，则该网桥重启该端口的定时器。最后，网桥该多播加入消息向多播加入消息的目的 MAC 地址对应端口进行转发。
- 如果网桥包含对应表项，且该表项的状态为“完成”，则该网桥检查在该表项的转发端口列表是否已经包含接收多播加入消息的端口。
  - 如果不包含，则网桥把该端口加入到转发端口列表中，启动该端口的定时器，并在端口为该多播组保留资源。接着，该网桥生成一个肯定应答消息，并把该消息发往多播加入消息的发送者；
  - 如果包含，则该网桥重启该端口的定时器。

处理多播离开消息

当网桥接收到多播离开消息后，它首先检测自己是否有和该消息对应的表项。并根据结果做以下操作。

- 如果网桥没有对应表项，或者接收多播离开消息的端口不在对应表项的转发端口列表中，则丢弃该多播离开消息；
- 如果网桥包含对应表项，且接收多播离开消息的端口在对应表项的转发端口列表中，则该网桥从转发端口列表中删除该端口，并释放在该端口上为该多播组保留的资源。如果删除后转发端口列表为空，则意味着目前已经没有设备需要网桥保留该多播组的注册信息，因此网桥删除该表项，并把多播离开消息向该消息的目的 MAC 地址对应端口进行转发。

处理肯定应答消息

当网桥接收到肯定应答消息后，它首先检测自己是否有和该消息对应的表项。并根据结果做以下操作。

- 如果网桥没有对应表项，或者接收多播离开消息的端口和对应表项的接收端口不一致，则丢弃该多播离开消息；
- 如果网桥对应表项的状态为“等待”，且接收多播离开消息的端口和对应表项的接收端口相同，则网桥把该表项的状态改为“完成”，并启动刷新定时

器 RTimer。最后，该网桥把肯定应答消息向该消息的目的 MAC 地址对应端口进行转发。

- ✧ 如果网桥对应表项的状态为“完成”，且接收多播离开消息的端口和对应表项的接收端口相同，则网桥检查该消息的目的 MAC 地址是否是自己的 MAC 地址，如果不是，该网桥把肯定应答消息向该消息的目的 MAC 地址对应端口进行转发。

处理否定应答消息

当网桥接收到肯定应答消息后，它首先检测自己是否有和该消息对应的表项。并根据结果做以下操作。

- ✧ 如果网桥没有对应表项，或者接收多播离开消息的端口和对应表项的接收端口不一致，则丢弃该多播离开消息；
- ✧ 如果网桥包含对应表项，且接收多播离开消息的端口和对应表项的接收端口相同，则该网桥把该消息向对应表项的转发端口列表中的所有端口转发，并删除该表项，释放所有相关的资源。

处理时钟超时

当定时器 PTimer 超时后，网桥的行为和从该定时器对应端口收到定时器对应表项的多播离开消息相同。

当定时器 RTimer 超时后，网桥重新启动 RTimer，并向 RTimer 对应表项的多播组的发送者发送多播加入消息。该消息的目的地址为多播组发送者的 MAC 地址，源地址为网桥的 MAC 地址。

发送者的行为定义

处理多播加入消息

当发送者接收到多播加入消息后，首先检测自己是否有和该消息对应的表项。如果没有找到对应的表项，则发送者丢弃该消息。不做任何操作。

如果发送者有对应表项，且接收到多播加入消息的端口在该表项的转发端口列表中，则发送者重启该端口对应的 PTimer。

如果发送者有对应表项，且接收到多播加入消息的端口不在该表项的转发端口列表中，则进一步检测是否有足够的资源满足多播加入消息的资源请求。根据检测结果，做如下操作：

- ✓ 如果发送者没有足够的资源满足多播加入消息的资源请求，则生成一个否定应答消息，把该消息发往多播加入消息的发送者，并丢弃加入消息。
- ✓ 如果发送者有足够的资源，则网桥把该端口加入到转发端口列表中，启动该端口的定时器 PTimer，并在端口为该多播组保留资源。接着，该发送者生成一个肯定应答消息，并把该消息发往接收者。

处理多播离开消息

当发送者接收到多播离开消息后，首先检测自己是否有和该消息对应的表项。如果没有找到对应的表项，则发送者丢弃该消息。不做任何操作。

如果发送者有对应表项，且接收到多播加入消息的端口在该表项的转发端口列表中，则该网桥从转发端口列表中删除该端口，并释放在该端口上为该多播组保留的资源。

如果发送者有对应表项，且接收到多播加入消息的端口不在该表项的转发端口列表中，则发送者丢弃该消息。不做任何操作。

处理肯定应答消息

当发送者接收到肯定应答消息后，丢弃该消息，不做任何操作。

处理否定应答消息

当发送者接收到否定应答消息后，丢弃该消息，不做任何操作。

接收者的行为定义

处理多播加入消息

当接收者接收到多播加入消息后，丢弃该消息，不做任何操作。

处理多播离开消息

当接收者接收到多播离开消息后，丢弃该消息，不做任何操作。

处理肯定应答消息

当接收者接收到肯定应答消息后，启动 RTimer，并准备接收多播数据。

处理否定应答消息

当接收者接收到否定应答消息后，删除相应的多播表项。

3.4 专利三：以太网网桥之间的协同调度方法

3.4.1 专利提出的背景

1. 发明背景介绍

a. 发明的技术领域

本发明涉及计算机和通信网络中的服务质量保证和调度问题，特别是介质访问子层的服务质量保证和调度。如 IEEE 802.3 以太网的服务质量保证和调度。

b. 现有技术的说明

为了保证以太网的服务质量，IEEE 成立了驻地以太网研究组。在该组的最新研究报告中，引入了 125 微秒为单位的调度周期。在每个周期内部，代表多媒体应用的同步流量比代表传统以太网应用的异步流量有更高的优先权进行发送。为了防止同步流量过多时异步流量被饿死，每个周期的同步流量的最大利用率限制在 75%。选用 125 微秒周期的主要原因是参考已有的 IEEE 1394 标准[4]，该标准目前被广泛用来连接音频/视频设备；另一个原因是短的调度周期比较容易提供低延迟和低抖动。

在 125 微秒周期的基础上，研究报告中给出了如何对同步流量进行调度，从而达到较低的延迟和抖动的机制——踱步（Pacing）：

“通过保持每个同步帧直到该帧相应的发送时间，踱步机制在整个流的路径上维护着流量的模式，从而保证了低的抖动边界和网络中缓存空间的分布化。具体的来说：”

“同步 A 类帧被控制以防它们（比预计）较早的离开交换机，其中控制指阻塞从第 n 个周期来的同步 A 类帧，直到第 n+p 个周期的开始。第 n+p 个周期开始，且第 n+p-1 个周期的非 A 类帧处理完毕之后，传输器开始发送这些来自第 n 个周期的 A 类帧，这些帧对于下个网桥而言就成了从 n+p 个周期来的。”

“每个网桥的延迟，抖动和缓存需求由踱步中的参数 p 所决定”

c. 现有技术问题或要改善的地方

从本质上来说，踱步属于非工作守恒的调度机制。踱步机制在延迟和带宽分配粒度之间存在平衡。也就是说，如果想要较小的延迟，则带宽分配粒度必定比较大，反之亦然。踱步机制通过调节调度周期的长短来在延迟和带宽分配粒度之间进行平衡。调度周期越短，则延迟越小，带宽分配粒度越大；调度周期越长，则延迟越大，带宽分配粒度越小。为了获得低延迟，驻地以太网研究组选择了非常短的调度周期（125 微秒），这就造成了非常粗糙的带宽分配。即使应用选择使用 128 字节的小型帧且不考虑帧间隙，最低的可分配带宽也达到了 8.192 兆位每秒(Mbps)。对于目前的重要应用 IP 电话而言，所占用的带宽一般只有 3~12 千位每秒(Kbps)，带宽的浪费达到了 99.8%。

为了解决该问题，驻地以太网研究组在报告中采用了基于速率的优先权调度机制。具体的来说，A 类流量又被细分成四个子类，分别是 CLASS_A0, CLASS_A1, CLASS_A2 和 CLASS_A3。对于这四个子类，其调度周期从原来的 125 微秒分别变成 125 微秒，500 微秒，2 毫秒和 8 毫秒。CLASS_A0 子类代表了最高速率的流量，这些流量以 8 KHz 的频率周期性的进行发送，而 CLASS_A1 和 CLASS_A3 则依次代表了更低速率的流量，如 IP 电话等。

更多流量子类的引入给用户提供了流量优先级方面更多的选择，然而，这种方案仍然没有解决延迟和带宽分配粒度之间的问题。也就是说，我们仍然需要牺牲延迟来达到更高的网络利用率。对于低带宽，低延迟需求的 IP 电话类业务而言，仍然意味着很大的带宽浪费。

3.4.2 专利思想的描述

本发明提出了一种以太网中网桥之间的协同的调度方法，其目的是为了使得以太网能够有效地支持低带宽高延迟需求的应用，例如 IP 电话或者交互式游戏。目前以太网中的方法很难有效地支持此类应用。具体的来说，本发明提出了一种降低低速率流延迟的方法。对于 8 毫秒才发送一次数据的低速率流而言，由于为该流保留资源的时隙和该流产生数据的不同步性，因此最大 8 毫秒的等待时间是不可避免的。然而，我们希望这种不同步性所造成的大延迟在整个端到端传输的过程中只出现一次。也就是说，通过网桥之间进行协作调度，使得网桥之间的不同步性所导致的延迟尽量降低。

a. 发明的构成

为了能够更好在网桥之间进行协作调度，本发明定义了基于超帧的调度框架，以及网桥之间的协作算法。超帧从编号为 64 的倍数的调度周期开始，其长度等于 64 个调度周期，即 8 毫秒。图 3 给出了超帧的结构。在跳步方案中，如果没有流的加入和离开，每个周期的调度表应该是不变的。对于本发明而言，超帧中的每个调度周期的调度表都可能不同。这里，调度表包括每个流的特征描述，以及一定时期内每个流被允许通过的流量大小。基于超帧的调度算法根据当前网桥的调度表，以及邻居网桥的调度表，来判断一个新的流是否能够被接纳。同时，如果是的话，对这个新的流进行调度安排，并分配资源。

网桥之间的协作算法在邻居网桥之间交换调度信息，使得调度算法能够借助此类信息尽量降低端到端的延迟。当调度算法根据当前网桥和邻居网桥的调度表对新的流进行调度后，协作算法把更新后的调度表发送给邻居网桥。

a.1 协作算法

协作算法主要有两个功能：第一个功能是超帧起始位置的同步，简称超帧同步；第二个功能是在网桥之间交换调度表。对于第一个功能，目前的驻地以太网已经引入了全网同步的概念，在此基础上很容易进行超帧同步。目前驻地以太网中，设备之间交换信息来选择一个时钟精度高的设备作为首席主设备（Grand Master），所有的其它设备直接或间接跟该设备进行时间同步。当选定了首席主设备后，由该设备决定超帧起始位置，并向全网广播。其余设备把首席主设备发布的超帧起始位置也作为自己的超帧起始位置，从而达到了全网超帧起始位置的同步。

对于协作算法的第二个功能，为了减少不必要的更新和交互，只对变化的流调度信息进行交互。具体来说，在接纳控制阶段，上游的网桥把新流的调度表，即在哪些周期内允许该流进行传输，以及这些周期内允许传输多少的信息，发送给下游的网桥。这里下游和上游以流的发送者为参照物，对于两个网桥而言，靠近发送者的称为上游，远离发送者的称为下游。

a.2 接纳控制算法

在给出接纳控制算法之前，需要先引入空余能力的概念。空余能力指网桥的某个端口上，一定时间范围内，还能传送的 A 类流量。考虑到 A 类流量被细分为四个子类，空余能力需要在这四个子类的不同周期长度上分别进行统计（即 125 微秒，500 微秒，2 毫秒和 8 毫秒）。根据空余能力的定义，如果在某个子类的周期长度上，空余能力小于零，则说明当前的流大于网桥的处理能力，即网桥无法满足当前所有流的需求。因此，控制接纳算法需要检查加入新的流后，所有四个子类的周期上的空余能力是否大于等于零。如果是，则新的流可以被接纳；否则，需要拒绝新的流。

a.3 调度算法

在前面定义的协作算法的前提下，网桥只知道端到端的延迟需求以及从发送者到自己为止共需要多少时间，而不清楚从自己到接收者的状况。为了使得流不超过端到端的延迟需求，对每个网桥来说，需要尽可能的减少流排队所需的延迟。另外，为了和踱步(Pacing)机制兼容，在周期 N 收到的流最早只能在 N+1 转发。由此，本发明给出的调度算法如下：

第一步，发送者把新流的相关信息向接受者发送；

第二步，当网桥捕获到该流的信息后，寻找从预计接收到该流的周期开始，能够满足流带宽需求的首个周期作为该流的发送周期；

第三步，网桥更新流的描述信息，包括累计延迟和周期编号。

第四步，如果更新后的累计延迟小于端到端延迟需求，则把该信息继续向接受者转发；否则，拒绝该流。

第五步，当接收者收到新流的描述信息后，检查累计延迟是否小于端到端延迟需求，如果是，则说明成功地进行了调度，否则，拒绝该流。

3.4.3 专利的效果

目前同步以太网很难有效的支持低带宽，低延迟要求的应用，例如 IP 电话和在线游戏等交互式应用。本发明给同步以太网提供了支持该类交互式应用的能力。我们以下面的简单例子来说明该问题：

假设 VoIP 的流使用 256 字节的帧，两个帧之间的最小间距为 16 字节，同步以太网的带宽为 100Mb/1Gb，周期长度为 125 微秒，可以保留的带宽占总带宽的 75%。

100Mbps 的快速以太网交换机每个周期能够保留的字节数为：

$$\frac{100\times10^{6}}{8}\times125\times10^{-6}\times75\%\approx1171\text{ 字节}$$

类似，1Gbps 的快速以太网交换机每个周期能够保留的字节数为：

$$\frac{1\times10^{9}}{8}\times125\times10^{-6}\times75\%\approx11718\text{ 字节}$$

如果资源预留策略选择每个周期都保留，则

100Mbps 的快速以太网交换机能够支持的 VoIP 流数目为：

$$\frac{1171}{256+16}\approx4\text{个流}$$

1Gbps 的快速以太网交换机能够支持的 VoIP 流数目为：

$$\frac{11718}{256+16}\approx43\text{个流}$$

在这种状态下，每个流实际占用的带宽为：

$$\frac{1\times10^{9}\times75\%}{43}\approx17{,}441{,}860\text{bps}\approx17.4\text{Mbps}$$

交换机所带来的延迟（统计平均）为 125 微秒

实际上，VoIP 所需要的带宽一般只有十几到几十 Kbps，每个周期都保留的预留策略带宽浪费超过 99%。

如果资源预留策略采用基于速率的优先权调度机制，并选用 CLASS_A1 子类，即 4 个周期预留一次，则

100Mbps 的快速以太网交换机能够支持的 VoIP 流数目为：$4\times4=16$个流

1Gbps 的快速以太网交换机能够支持的 VoIP 流数目为：$43\times4=172$个流

在这种状态下，每个流实际占用的带宽为：$\frac{1\times10^9\times75\%}{172}\approx4.4\text{Mbps}$

交换机所带来的延迟（统计平均）为$\frac{125*4}{2}=250$微秒

如果资源预留策略采用基于速率的优先权调度机制，并选用 CLASS_A2 子类，即 16 个周期预留一次，则

100Mbps 的快速以太网交换机能够支持的 VoIP 流数目为：$4\times16=64$个流

1Gbps 的快速以太网交换机能够支持的 VoIP 流数目为：$43\times16=688$个流

在这种状态下，每个流实际占用的带宽为：$\frac{1\times10^9\times75\%}{688}\approx1.1\text{Mbps}$

交换机所带来的延迟（统计平均）为$\frac{125*16}{2}=1000$微秒$=1$毫秒

如果资源预留策略采用基于速率的优先权调度机制，并选用 CLASS_A3 子类，即 64 个周期预留一次，则

100Mbps 的快速以太网交换机能够支持的 VoIP 流数目为：$4\times64=256$个流

1Gbps 的快速以太网交换机能够支持的 VoIP 流数目为：$43\times64=2{,}752$个流

在这种状态下，每个流实际占用的带宽为：$\frac{1\times10^9\times75\%}{2752}\approx272\text{Kbps}$

交换机所带来的延迟（统计平均）为$\frac{125*64}{2}=4000$微秒$=4$毫秒

选用 CLASS_A3 子类时，交换机可以支持的流数目比起 CLASS_A1（即每个周期都保留）子类要大大增加，同时带宽浪费也大大减少，但 CLASS_A3 子类每经过一个交换机的延迟统计平均约为 4 毫秒，对于 VoIP 业务来说是难以接受的。

如果选用本专利所定义的协作式调度算法，交换机能接纳的流数，以及延迟依赖于网络拓扑、流分布、端到端延迟需求、当前的资源预留状况，上游交换机

选定的周期等一系列参数，难以像上面的调度方法那样直接给出确定性的数值。但每个流实际占用的带宽比较容易计算出来，即 $\frac{(256+16)\times 8}{125\times 10^6\times 64}\approx 272\text{Kbps}$

3.5 专利四：自适应的多跳时分复用调度算法

3.5.1 专利提出的背景

1. 发明背景介绍

a. 发明的技术领域

本发明涉及计算机和通信网络中的服务质量保证和调度问题，特别是介质访问子层的服务质量保证和调度。如 IEEE 802.3 以太网的服务质量保证和调度。

b. 现有技术的说明

为了保证以太网的服务质量，IEEE 成立了驻地以太网研究组。在该组的最新研究报告中，引入了 125 微秒为单位的调度周期。在每个周期内部，代表多媒体应用的同步流量比代表传统以太网应用的异步流量有更高的优先权进行发送。为了防止同步流量过多时异步流量被饿死，每个周期的同步流量的最大利用率限制在 75%。选用 125 微秒周期的主要原因是参考已有的 IEEE 1394 标准，该标准目前被广泛用来连接音频/视频设备；另一个原因是短的调度周期比较容易提供低延迟和低抖动。

在 125 微秒周期的基础上，研究报告中给出了如何对同步流量进行调度，从而达到较低的延迟和抖动的机制——踱步（Pacing）。踱步机制在延迟和带宽分配粒度之间存在平衡。也就是说，如果想要较小的延迟，则带宽分配粒度必定比较大，反之亦然。踱步机制通过调节调度周期的长短来在延迟和带宽分配粒度之间进行平衡。调度周期越短，则延迟越小，带宽分配粒度越大；调度周期越短，则延迟越大，带宽分配粒度越小。为了获得低延迟，驻地以太网研究组选择了非常短的调度周期（125 微秒），这就造成了非常粗糙的带宽分配。

为了解决踱步机制带来的问题，专利“以太网网桥之间的协同调度方法”利用网桥之间的协同调度降低端到端延迟。该发明提出了一种降低低速率流延迟的方法。对于 8 毫秒才发送一次数据的低速率流而言，由于为该流保留资源的时隙和该流产生数据的不同步性，因此最大 8 毫秒的等待时间是不可避免的。然而，该专利希望这种不同步性所造成的大延迟在整个端到端传输的过程中只出现一

次。也就是说，通过网桥之间进行协作调度，使得网桥之间的不同步性所导致的延迟尽量降低。

具体的来说，专利“以太网网桥之间的协同调度方法”首先引入了超帧的概念，即把 64 个调度周期合并在一起称为一个超帧。对于低速率的流，网桥根据邻居网桥发来的该流在超帧中的预计到达周期，选择转发该流的周期。在选择转发周期的时候，该专利使用了贪婪算法，即选择能够满足带宽需求且延迟最小的周期。模拟试验表明，和踱步方法相比，该专利大大提高了可以接纳的流数目，从而能够更有效地支持低带宽低延迟要求的流，如 VoIP 和网络游戏等。

c. 现有技术问题或要改善的地方

专利“以太网网桥之间的协同调度方法”虽然很好的解决了支持低带宽低延迟要求流，如 VoIP 和网络游戏等的效率问题，但并没有考虑到这种流和传统高速流的共存问题。驻地以太网所采用的 125 微秒周期长度，很大程度上是为了和目前在音视频方面占有很大市场的 IEEE 1394 兼容；同时，驻地以太网的研究报告中也花很大篇幅研究了如何在驻地以太网上传送 IEEE 1394 流。因此，一个好的调度方案在高效支持 VoIP 和网络游戏等流的同时，也应该能够很好地支持传统的高速流（即需要在每个周期内做资源预留的流）。

专利“以太网网桥之间的协同调度方法” 使用了贪婪算法作为选择转发周期的算法，虽然该算法能够在最大程度上保证延迟不超过需求，但同时，该算法也容易造成在一些周期很空的时候，某些周期过早饱和，从而使得需要在每周期内都发送数据的高速流被拒绝。也就是说，贪婪算法很有可能对传统的高速流不够友好。为了避免不合适的调度方案降低高速流的接纳概率，需要研究一种充分考虑高速率特性的调度方案。

3.5.2 专利思想的描述

为了在保证端到端延迟需求的同时兼顾高速流的接入请求，本发明提出了一种自适应的多跳时分复用调度算法。传统的贪婪算法尽可能选择能够满足带宽需求且延迟最小的周期，虽然贪婪算法能够在最大程度上保证延迟不超过需求，但同时，贪婪算法也容易造成在一些周期很空的时候，某些周期过早饱和，从而使得需要在每周期内都发送数据的高速流被拒绝。本专利提出的自适应的多跳时分复用调度算法克服了贪婪算法的不足之处，使得高速流有更大的可能性被网络接纳。

a. 发明的构成

为了在保证端到端延迟需求的同时兼顾高速流的接入请求，本发明提出了一种自适应的多跳时分复用调度算法。为了达到尽量满足高速流接入请求的目的，首先需要知道什么情况下流的接入请求有可能被拒绝。如果有些周期的空余能力接近零，则那些需要在每个周期内都传送的高速流将会有很大概率被拒绝。为了使得这类事情发生的概率尽量降低，调度算法应该使得每个周期内的流分布尽量均匀。同时，调度算法还要使得流的端到端延迟满足需求。在这两个原则的共同作用下，本发明给出的调度算法如 a.1 小节所示

a.1 自适应的多跳时分复用调度算法

第一步，根据端到端的延迟需求和从发送者到接收者的路径长度，估算出从发送者到每个网桥的时候，累积的延迟大小。该延迟被称为预期延迟，预期延迟的估算方法参见下面 a.2 小节的说明；

第二步，对于从发送者到接收者的路径上的每个网桥，在计算新流的调度表时，参考从发送者开始到自己为止，实际累计的延迟，并使得该延迟尽量不要超过预期延迟。具体的方法如下：

如果有多于一个可以满足实际累计的延迟不超过预期延迟的调度方案，选择使得流分布尽量均匀的那个调度方案。

如果不存在实际累计的延迟不超过预期延迟的调度方案，则选择使得延迟最小的调度方案。

如果累计的延迟超过了端到端延迟需求，则拒绝接纳该流。

第三步，如果该流被接纳，则网桥把该流的调度信息，以及累积的延迟传递到下游网桥。

a.2 估算预期延迟

把端到端的延迟平均分布到每个网桥上是比较直观的估算预期延迟的方法。虽然这种方法看起来比较公平，但实际上它并不公平。对于接近发送者的网桥来说，即使预期延迟之前的所有周期都不可用，由于预期延迟和端到端延迟需求相差较大，网桥仍然可以把流安排在预期延迟之后的周期中；对于靠近接收者的网桥，如果预期延迟之前的所有周期都不可用，由于预期延迟和端到端延迟需求相差较小，因此网桥的可选择余地（即预期延迟之后的周期数目）很小，即新的流会有很大概率被拒绝。

为了减低该问题的影响，需要给靠近接收者的网桥留更多的可选择空间。具体的来说，假设端到端的延迟需求为 D，从发送者到接收者的路径上共有 N 个网桥，我们把延迟需求分为 N+2 份，给最接近接收者的网桥留 3 份，给所有其它的网桥留一份。即这 N 个网桥的预期延迟分别为 $\frac{D}{N+2}, \frac{2\cdot D}{N+2}, \mathrm{K}, \frac{(N-1)\cdot D}{N+2}, D$。

3.6 专利五：多级设备之间时间同步 FCC 补偿方法的改进算法

3.6.1 专利提出的背景

目前使用 PTP 协议实现时间同步有三种补偿方法，它们是 FCC，OFCC 和 OCC 方法[5][6][7][8]。本文所描述的发明是对 FCC(Frequency Compensation Correction)补偿方法的改进算法，所以下文将描述 FCC 补偿方法的现有技术以及存在的问题。图 17 给出了多跳环境下逐跳同步的示意图。

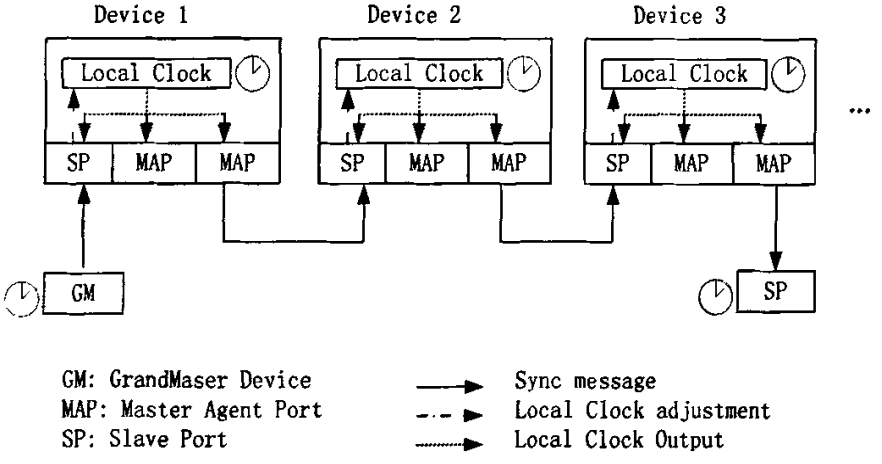


图 17　逐跳同步的示意图与应用场景

现有的技术

FCC 补偿是指以频率调整方式来补偿不同设备时间的频率漂移与时间偏移量。频率漂移是指不同设备时间计时所用晶振的频率差值；时间偏移量是指不同设备时间的差值。图 18 是一种时间同步的实现原理框图，而 FCC 补偿方法主要是通过计算得到频率补偿值 FreqCompValue，从而补偿频率漂移与时间偏移量对时间的影响，动态达到时间的同步。

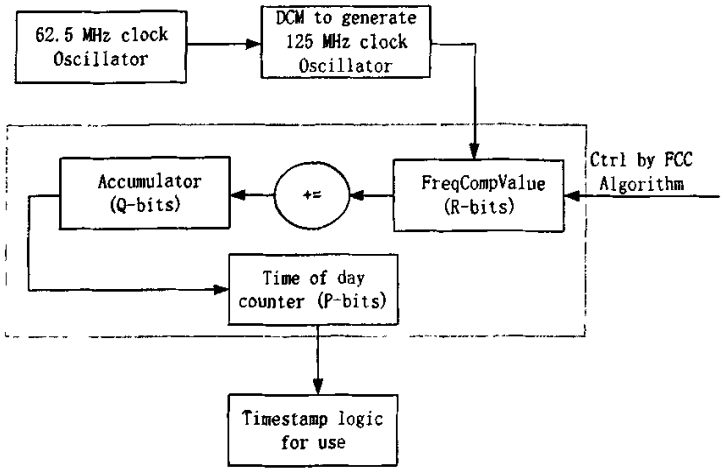


图 18 时间同步原理框图

图 19 描述了 IEEE1588[9]时间协议交互时间的基本流程。FCC 算法就是利用 IEEE1588 来交互时间信息的。

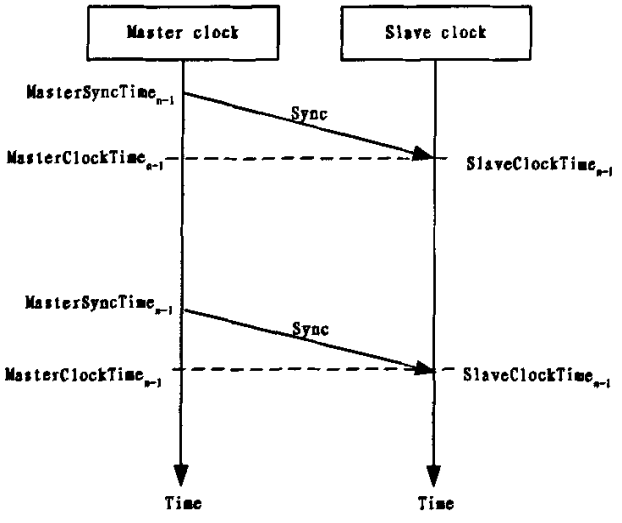


图 19： IEEE1588 的基本流程

下面将介绍现有技术（本文称为传统算法）是如何补偿FreqCompValue的。在时间点MasterSyncTime$_n$ 主设备发送一个Sync消息给从设备。从设备在时间点SlaveClockTime$_n$接收到 Sync消息；通过分析Sync消息中携带的时间信息，它可以计算主设备当前的时间MasterClockTime$_n$：

$$MasterClockTime_n = MasterSyncTime_n + MasterToSlaveDelay$$

在当前 Sync 周期里，主设备的时间计数 MasterClockCount$_n$ 由下式给出：

$MasterClockCount_n = MasterClockTime_n - MasterClockTime_{n-1}$

与此同时从设备计算出在当前Sync周期里从设备的时间计数$SlaveClockCount_n$，由下式给出：

$SlaveClockCount_n = SlaveClockTime_n - SlaveClockTime_{n-1}$

在当前Sync周期里，主从设备之间的时间计数的差值$ClockDiffCount_n$由下式给出：

$ClockDiffCount_n = MasterClockTime_n - SlaveClockTime_n$

从设备的频率调整因子 $FreqScaleFactor_n$ 由下式给出：

$$FreqScaleFactor_n = (MasterClockCount_n + ClockDiffCount_n)/SlaveClockCount_n \quad (0)$$

从设备可以根据下式更新频率补偿值 $FreqCompValue_n$：

$FreqCompValue_n = FreqScaleFactor_n * FreqCompValue_{n-1}$

在以上算法中，频率漂移与时间偏移量在每一个同步周期里都得到补偿，式(0)是上述算法的核心表达式。

存在的问题

从式(0)可以看出，频率漂移与时间偏移量在每一个同步周期里都得到补偿，这种补偿方法对于两个设备之间的同步能取得比较好的同步精度。但对于多级设备环境下的同步，它的同步精度或误差是随着级联数目呈指数分布增长的[4]。传统算法的确存在级联时误差与级联数目呈指数分布的问题。此外，根据传统算法调整后的精度也有待提高。

3.6.2 专利思想的描述

本专利对传统算法进行了改进，具体改进算法如下(称为改进算法)[10]：

在当前 sync 周期里，一旦从设备接收到 Sync 消息，它将根据以下公式计算频率调整因子 $FreqScaleFactor_n$。首先计算出 $ClockDiffCount_n$，然后执行如下计算：

1. 若$ClockDiffCount_n>0$；然后根据
   a) 如果$MasterClockCount_n$ 大于 $SlaveClockCount_n$，在式(0)的基础上对 $ClockDiffCount_n$ 加权 α 倍后进行$FreqScaleFactor_n$的调整；
   b) 如果$MasterClockCount_n$ 小于或等于 $SlaveClockCount_n$，在式(0)的基础上对$ClockDiffCount_n$ 加权 β 倍后进行$FreqScaleFactor_n$的调整；

2. 若$ClockDiffCount_n<=0$；然后

a) 如果$MasterClockCount_n$ 小于 $SlaveClockCount_n$，在式(0)的基础上对 $ClockDiffCount_n$ 加权 α 倍后进行$FreqScaleFactor_n$的调整；

b) 如果$MasterClockCount_n$ 大于或等于 $SlaveClockCount_n$，在式(0)的基础上对$ClockDiffCount_n$ 加权 β 倍后进行$FreqScaleFactor_n$的调整；

改进算法中新定义了有 α 、β 这两个参数，它们的取值范围在[0,1]之间。它默认取值分别为：0.5, 和 0.75。

专利创新点：

- ✓ 调整$FreqScaleFactor_n$时，判断$ClockDiffCount_n$的正值/负值，并判断$MasterClockCount_n$与$SlaveClockCount_n$的大小比较结果，分四种情况采取四种不同的策略进行调整；四种情况是：
  - $ClockDiffCount_n$大于0而$MasterClockCount_n$大于$SlaveClockCount_n$；
  - $ClockDiffCount_n$大于0而$MasterClockCount_n$不大于$SlaveClockCount_n$；
  - $ClockDiffCount_n$不大于0而$MasterClockCount_n$大于$SlaveClockCount_n$；
  - $ClockDiffCount_n$不大于0而$MasterClockCount_n$小于$SlaveClockCount_n$；
- ✓ 采取四种不同的策略进行调整是通过给$ClockDiffCount_n$分配不同的权重来实现的，且要求权重之间的关系满足$0 \leq \alpha \leq \beta \leq 1$； 具体调整方法见公式(1)~(4)；通过控制权重来调整，从而避免对时间进行过度地调整。

3.6.3 专利的测试结果

我们对传统算法与改进算法分别在两个设备级联与四个设备级联后的同步性能进行了测试与比较。。测试时的参数设置与测试结果说明：通道 2 的绿色波形是从示波器观察到的由 Grandmaster 输出的时间脉冲信号；通道 1 的黄色波形是从示波器观察到的由从设备输出的时间脉冲信号，由于是通道 2 触发，所以通道 1 的时间脉冲宽度反映了同步后的时间误差；每 100ms 同步一次。图 20 给出了两个设备运行传统算法后的同步结果；图 21 给出了两个设备运行改进算法后的同步结果；图 22 给出了 4 个设备运行传统算法后主设备与经历三跳后从设备之间的同步结果；图 23 给出了 4 个设备运行改进算法后主设备与经历三跳后从设备之间的同步结果。

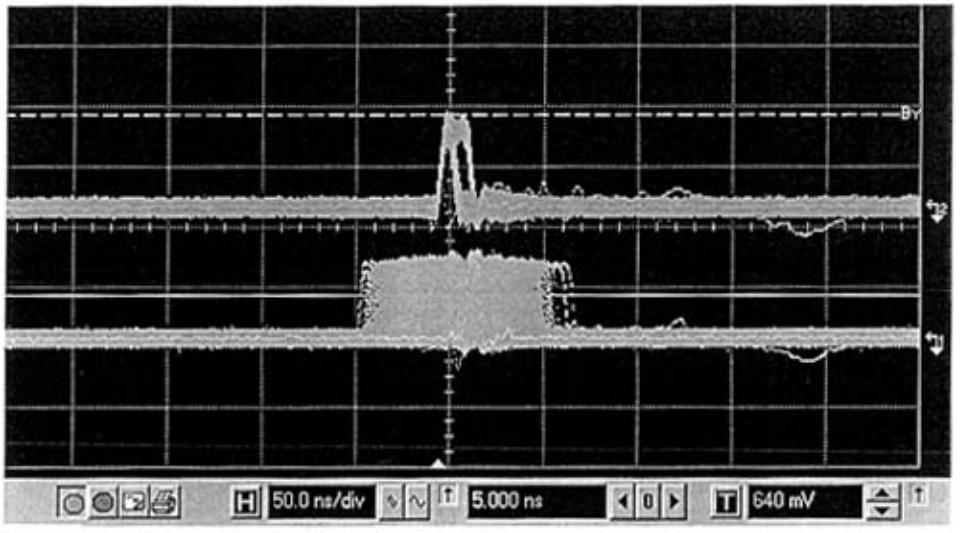


图 20　两个设备运行传统算法后的同步结果（1 跳）

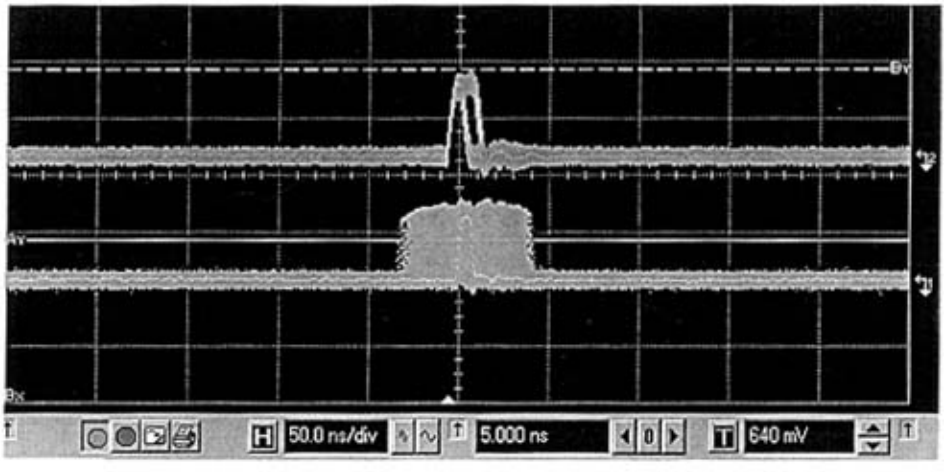


图 21　两个设备运行改进的同步算法后的同步结果（1 跳）

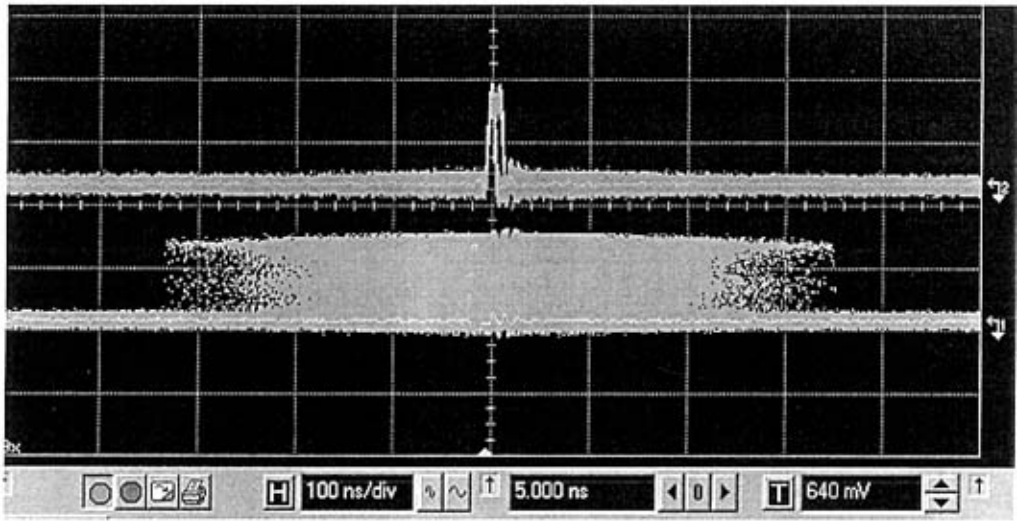


图 22 四个设备运行传统同步算法后的同步结果（3 跳）

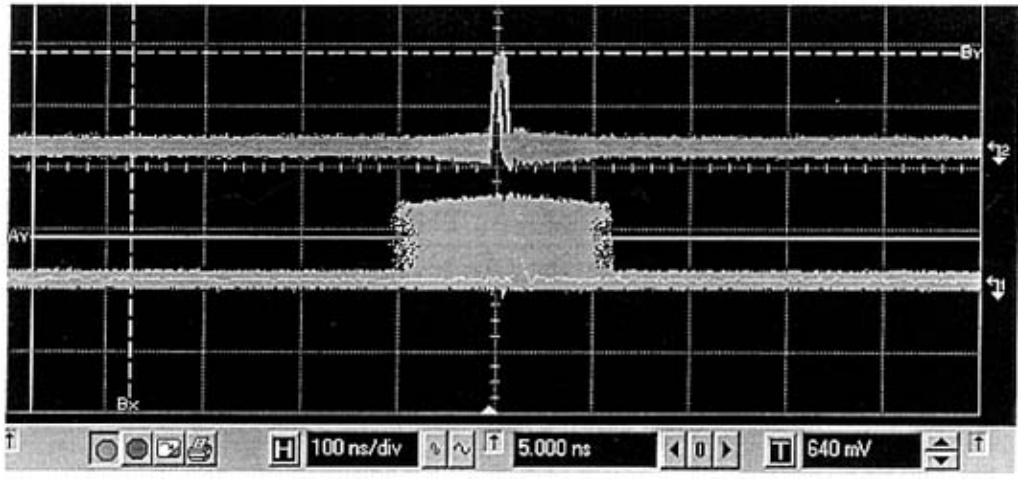


图 23 四个设备运行改进的同步算法后的同步结果（3 跳）

测试结果分析与结论：

1. 两个设备之间的同步结果：传统算法的误差范围约为[-50ns, 60ns]；改进算法的误差范围约为[-30ns,40ns]，其误差的峰峰值是传统算法的64%。
2. 四个设备之间3跳的同步结果：传统算法的误差范围约为[-340ns, 350ns]；改进算法的误差范围约为[-100ns,120ns]，其误差的峰峰值是传统算法的32%。
3. 改进算法显著改善了传统算法的性能，尤其是极大地改进了级联多

跳后时间同步的性能。

3.7 本章小结

本章首先对提出的五个专利进行了概述；然后对每个专利进行从专利提出的技术背景、专利细节描述与仿真或测试结果这几个方面进行了详细描述。最后，对本章进行了总结。本章所提的专利都是在项目开发进度非常紧张的情况下提出的，这些专利的思路或灵感也都来自于开发过程中遇到的问题，具有实用性与创造性，是项目组全体成员的集体智慧的结晶，是专利编写人巨大付出和努力的结果。

第四章 本文总结与展望

4.1 本文的工作总结

写到此处，行将搁笔，心中也是无限感慨。在本文最后，从开发成果、研究成果与经验总结这三个方面进行总结，并对进一步的研究工作进行了展望。

4.1.1 开发成果

RSE 相关课题的执行过程可以划分为三个阶段，在每一个阶段结束时韩国三星总部都要进行严格的验收测试。下面将总结出各个阶段的主要研发成果：

第一阶段的主要开发成果：

FPGA 逻辑与嵌入式软件协同开发实现同步以太网交换机：

1. 主要功能
   a) 实现 5X5 千兆输入输出交换阵，实现 5 路同步报文的交换；
   b) 支持 MAC 地址自学习；
   c) 支持 MAC 地址超时更新；
   d) 支持静态路由项；
   e) 根据同步以太网制定的超帧结构发送报文，保证同步报文的时延与时延抖动；
   f) 根据各个设备的能力，选择出主从同步设备；
   g) 实现精确时间协议，运行时间同步算法，实现设备间的时间同步。
2. 主要性能
   a) 千兆线速测试时，异步报文通过率达到 100%，测试的报文数达到 10^8 以上；
   b) 千兆线速测试时，同步报文通过率达到 95%以上，测试的报文数达到 10^8 以上；
   c) 混合流量测试时，要求 70%同步流量与 20%异步流量复用到一路时均不丢包；
   d) 混合流量测试时，要求 70%同步流量与 100%异步流量复用到

一路时要求同步报文不丢包；

e) 设备间时间同步误差在[-100ns, 100ns]之间；

第二阶段的主要开发成果：

单片 FPGA 实现混合流量的交换：

1. 主要功能：

a) 单片实现两套 5X5 千兆输入输出交换阵，实现 5 路同步报文的交换；并且实现 5 路异步报文的交换；

b) 支持 MAC 地址自学习；

c) 支持 MAC 地址超时更新；

d) 支持静态路由项；

e) 根据同步以太网制定的超帧结构发送报文，保证同步报文的时延与时延抖动。

2. 主要性能

a) 千兆线速测试时，异步报文通过率达到 100%；

b) 千兆线速测试时，同步报文通过率达到 95%以上；

c) 混合流量测试时，要求 70%同步流量与 20%异步流量复用到一路时均不丢包；

d) 混合流量测试时，要求 70%同步流量与 100%异步流量复用到一路时要求同步报文不丢包；

第三阶段的开发成果：

1. FPGA 逻辑与嵌入式软件协同开发时间同步系统：主要功能：

a) 实现精确时间协议，运行时间同步算法，实现设备间的时间同步；

b) 相邻设备之间的时间同步的误差在[-50ns, 50ns]范围之内；

c) 多跳设备之间的时间同步的精度需要克服传统算法的缺陷（级联后的时间同步的误差与级联数呈指数分布），需要对算法进行改进，显著改进同步精度。

4.1.2 研究成果

研究成果的主要输出是五个专利的发明，其中前四项在国内申请专利，第五

项准备申请国际专利。五项专利依次如下。

1. 一种单播方式的以太网多播控制信息传递方法
2. 综合多播注册和资源预留方法
3. 以太网网桥之间的协同调度方法
4. 自适应的多跳时分复用调度算法
5. 多级设备之间时间同步 FCC 补偿方法的改进算法

4.1.3 经验总结

经验一：

在执行项目计划时或衡量项目中各个活动难易时，对于能掌控的工作任务，应制定非常详细的工作计划与工程成果输出；对于暂时不能完全掌控的工作任务，可以制定比较粗的工作计划，并尽量做到有一定的灵活性；随着经验与知识的积累，逐渐去掌握它，然后才可能制定出详细地切实可行的计划。这类经验来自与金文雄总监的点拨，在项目开展过程中非常有用，受益匪浅。

经验二：

对于以开发为主体的博士后课题中，首先需要良好的开发平台，以平台为依托，才可以更好测试验证开发的成果；对于专利思想，通过仿真获知其性能数据固然重要，但是只有在开发平台上验证后在能清楚地知道专利思想是否可行、实现的难易程度、具体的性能结果。

经验三：

现在的项目开发越来越复杂，而且开发的需求是不断变化的，某一个人是不可能完成所有工作的，因此需要有良好的团队合作精神与能力。此外，组织团队时也需要多种不同方面特长的人员组成，才能更好地应对各种风险。

4.2 进一步的研究工作

有关在 Ethernet 上开展实时业务的研究已经持续多年了，但目前依然是活跃的研究领域，充分说明其复杂性和艰巨性。作者与项目组成员的这一点点工作，相对于目前已有的研究和待续的研究来说，仅仅是冰山之一角。在下一步，作者将加强下述几个方面的研究工作：

1. 进一步跟踪并研究设备间时间同步算法，以开发成果与平台为依托，加强理论研究，并将研究结果推向标准化组织；
2. 研究资源预留与接纳控制策略，以切实可行地支持已预留业务的服务质量；
3. 研究设备发现协议与内容发现协议，为视频内容的点播与设备的利用提供指导信息。

参考文献

[1] Residential Ethernet (RE) (a DVJ working paper), REsE interest group, May 2005.

[2] ITU-T Recommendation G.1010(2001), End-user multimedia QoS categories.

[3] IEEE Standard 802.1D. Media Access Control (MAC) Bridges. IEEE Standard for Local and Metropolitan Area Networks, 1998

[4] Johansson, P., "IPv4 over IEEE 1394", RFC 2734, December 1999.

[5] J. Jasperneite, K. Shehab, and K. Weber, "Enhancements to the time synchronization standard IEEE-1588 for a system of cascaded bridges", in 5th IEEE International Workshop on Factory Communicatioin Systems (WFCS'2004), 2004, pp. 239-244.

[6] S. Wang, J. Cho, Y. Joo, S. Hwang, Y. Oh, "Improvement to boundary clock based time synchronization through cascaded switches" , IEEE Trans. Consumer Electron., submitted for publication.

[7] S. Wang, J. Cho, Y. Joo, S. Hwang, Y. Oh , "Performance Analysis of Boundary Clock Based Time Synchronization through Cascaded Switches", IEEE Trans. Consumer Electron., submitted for publication

[8] S. Balasubramanian, K. R. Harris, and A. Moldovansky, "A frequency compensated clock for precision synchronization using IEEE 1588 protocol and its application to Ethernet", presented at Proc. of the Workshop on IEEE 1588, Gaithersburg, U.S., 2003.

[9] IEEE, "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems", ANSI/IEEE Std 1588-2002.

[10] Zhousong Huang, Sihai wang and etc., "Performance Analysis of Boundary Clock Based Time Synchronization through Cascaded Switches", IEEE/ACM DS-RT 2006 meeting, submitted for publication

本文作者在博士后工作期间的论文

[1] Zhousong Huang, Sihai Wang and etc., "Analyze and improve Frequency-only Compensation Correction Based Time Synchronization through Cascaded Bridges", IEEE/ACM DS-RT 2006 meeting, submitted for publication.

本文作者在博士后工作期间的专利

[1] 黄周松，郑剑锋，多级设备之间时间同步频率补偿方法的改进算法， 中国，发明专利

[2] 黄周松，郑剑锋，多级设备之间时间同步 FCC 补偿方法的快速收敛算法， 正在申请中

[3] 吴起,黄周松， 一种单播方式的以太网多播控制信息传递方法，专利申请号：200510120072.0

[4] 吴起,黄周松,综合多播注册和资源预留方法，专利申请号：200510135915.4

[5] 吴起,黄周松，以太网网桥之间的协同调度方法， 中国， 发明专利

[6] 吴起,黄周松，自适应的多跳时分复用调度算法， 中国， 发明专利

致谢

论文搁笔之际心中有着无限的感慨。在 BST 攻读博士后的经历是我成长过程中最重要的阶段，其间泛着学业与项目管理的风风雨雨与成功的喜悦。

衷心感谢我的导师林金桐教授与伍剑老师。他们的敏锐渊博的学识、认真严谨的治学作风、以及对学生的提携及照顾，令我获益良多且深受感动。

感谢 BST 的所有领导，是他们的多方指导和关心，使本人的研究工作和总结报告能顺利完成。尤其感谢尹洪烈副院长、金文雄总监和李小强 GL，是你们直接指导我完成了所有项目，也传授给我许多经验和技能。感谢王彤院长、沈勇男部长与郑喜勋课长，在你们的管理下我能有序地开展工作，感谢你们的帮助。

感谢 KHQ 与我们合作部门的所有领导与同事。

我要特别感谢我们组的同事们。郑剑锋、谭兴晔、吴起与毕务刚为我们项目付出了艰苦努力与不悔。我永远感激你们，也永远怀念我们的团队！

感谢我的爷爷奶奶的抚育与培养之恩，没有他们就没有我的今天。

感谢我的妻子、女儿与其他亲友，你们的每一个笑容给了我无限的鼓励。

最后，谨将此文献给我远去的爷爷、奶奶与父亲！