

兰州理工大学

硕士学位论文

大数据量实时流媒体传输拥塞控制机制研究

姓名：王伟

申请学位级别：硕士

专业：通信与信息系统

指导教师：宋健

20070510

摘 要

随着网络技术的发展,单纯的文本形式已经不能满足人们对网络应用的要求。多媒体技术的应用成为网络发展的方向。流式传输作为最为经济的传输方式,逐渐受到人们的青睐。网络正不断改变着人们的交流方式,远程教育、远程医疗等实时多媒体应用不断增加,人们在享受多媒体应用的同时,对实时多媒体应用的传输质量提出了更高的要求。网络带宽瓶颈不断被突破,一些高带宽要求的视频流传输成为可能。为了加速实时多媒体应用的发展,IETF A/V工作组于1996年制定了实时传输协议(Real-time Transport Protocol,RTP),并以RFC1889文档的形式进行了具体的规范,从而提出了一种基于RTP协议,利用RTP传输控制协议(Real-time Transport Control Protocol,RTCP)进行传输质量反馈,实现多媒体实时传输的解决方案。

高带宽要求的视频流传输数据量大,视频流传输中存在许多问题。网络的负载均衡、视频的回放质量都将严重影响整个网络的安全以及传输系统的应用。由于TCP协议本身的三次握手、以及拥塞退避机制不利于实时应用,所以实时流媒体传输普遍采用UDP作为传输层协议,这在一定程度上满足了实时多媒体应用的时延敏感性,但是UDP协议本身不提供拥塞控制,而网络上的应用大多采用具有拥塞控制机制的TCP协议,大量的RTP/UDP流传输,必将产生与传统TCP流之间的带宽竞争,缺少拥塞控制的UDP流无法适时减少数据发送量,缩小带宽要求,将严重侵占TCP流的带宽,严重时会导致网络拥塞崩溃;UDP本身的尽力传输性还会导致拥塞发生时,将有大量的数据包被抛弃,从而严重影响实时应用的服务质量,同时也不能最大限度提高视频回放的质量,所以说针对高带宽要求的大数据量实时流媒体应用必须结合媒体特点有相应的传输拥塞控制机制。

TCP协议提供了拥塞控制的机制,为了实现网络的负载均衡,人们提出了TCP友好TCP-Friendly的拥塞控制机制,并应用于实时流媒体的应用,研究表明目前提出的TCP友好拥塞控制不能很好的满足实时流媒体应用的需求,本文在对TCP协议的拥塞控制机制、TCP友好的拥塞控制机制研究的基础上,通过对RTP协议的深入分析,结合高带宽要求视频格式特点,通过对丢包率进行卡尔曼过滤(Kalman Filter)分析预测网络负载状况、结合接收缓冲区大小设置来平滑发送速率,提出了一种基于RTCP反馈的TCP友好拥塞控制机制,并通过模拟试验对该拥塞控制机制的TCP友好性,以及视频流回放质量进行了测试。

关键词:实时传输协议;流媒体;大数据量;TCP友好;拥塞控制

Abstract

With the development of network technology, a simple text format can no longer meet the requirements of network applications. The application of multimedia technology into network development direction. Streaming transmission, as have most economic means of transmission, people have gradually been favored. The Internet is changing the way of people, tele-education, tele-medicine and other real-time multimedia applications increasing. People enjoy multimedia applications, the transmission of real-time multimedia applications require a higher quality. Network bandwidth bottleneck continuously be broken and some of the high bandwidth requirements of video transmission be possible. In order to accelerate the development of real-time multimedia applications. IETF A/V Working Group in 1996 enacted the Real-time Transport Protocol (Real Transport Protocol, RTP), and RFC1889 document in the form of a concrete standards, thus proposed based on RTP. RTP using transmission control protocol (Real-time Transport Control Protocol, RTCP) transmission quality feedback, and real-time transmission of multimedia solutions.

Video bandwidth requirements of large-volume data transmission, video transmission, there are many problems. Network Load Balancing, the intervals Video quality will seriously affect the entire transmission system and network security applications. As the three TCP handshake and congestion backoff mechanism is not conducive to real-time applications. Therefore, real-time transmission streaming media commonly used as a transmission protocol for UDP. This to a certain extent, to meet the real-time delay-sensitive multimedia applications. But UDP fails to provide congestion control. Most of the application and network congestion control mechanism with the TCP protocol used. lots of RTP /UDP transmission will have the traditional TCP bandwidth competition. UDP is not a lack of time to reduce congestion control the amount of data sent and narrow bandwidth requirements, TCP will be a serious invasion of bandwidth, network congestion will lead to a serious collapse. UDP will lead to the best serious transmission congestion occurs when a large number of data packets to be discarded. This will seriously affect the quality of service for real-time applications, but also can maximize the quality of video intervals. Therefore the volume of data required for high bandwidth real-time streaming media applications must be integrated with corresponding characteristics of the transmission media congestion control mechanism.

TCP provides congestion control mechanisms to achieve network load balancing. It proposed a friendly TCP (TCP-Friendly congestion control mechanism and the application of real-time streaming media applications. The present study shows that the TCP Congestion

Control unable to meet the demand for real-time streaming media applications. Based on the TCP's congestion control mechanisms, TCP-friendly congestion control mechanism, RTP-depth analysis of the agreement, combined with high bandwidth requirements of video format features through Kalman Filter to forecast the packet loss rate for analysis of network load conditions. Smoothing the size of the buffer zone set up to receiver light transmission rate. RTCP feedback on the proposed a TCP-friendly congestion control mechanism and congestion control mechanism through simulation tests on the TCP-friendly and the quality of the test video streaming intervals.

Key Words: real-time transport protocol;stream; large volume;TCP-friendly; congestion control

兰州理工大学

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名： 王伟

日期：2007年5月10日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权兰州理工大学可以将本学位论文的全部内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

1、保密口，在_____年解密后适用本授权书。

2、不保密☒。

(请在以上相应方框内打“√”)

作者签名： 王伟

日期：2007年5月10日

导师签名：

陈健

日期：07年5月15日

第1章 绪 论

1.1 课题研究的背景

我们正处在网络多彩精致的时代。快速链接和多媒体计算机的出现使网络上的信息不再只是文本、图像和简单的声音文件。越来越多的公司和个人正在利用音频、视频剪辑的多媒体技术使自己的站点更具吸引力、拥有更大的信息量和更高的经济效益。这些多媒体内容包括电影、新闻剪辑电台、电视广播、网络游戏甚至虚拟宠物。想要使用这些多媒体信息,就必须实现视频、音频信息在网络上的流式传输。在过去,音视频文件需要从服务器上下载到本地后才能播放,采用下载方式虽然使 Internet 上多媒体信息的传输成为可能,但它也带来了两个突出的问题。首先,由于必须下载多媒体信息,而多媒体信息的数据量通常都很大,在普通用户接入速率较低的情况下,一个很短的视频片段可能都需要很长的下载时间;其次,由于必须将节目下载到本地计算机后才能播放,这必然占用本地计算机的存储资源。一个 1 分钟的 MPEG-1 视频节目所需的存储空约为 12M,如果用户使用 56Kbps 的 Modem 接入,那么下载这个节目至少需要 2 分钟。如此长的下载时间对普通用户来说是难以忍受的,这些都限制了人们在 Internet 上大量使用音频和视频信息进行交流。针对这些亟待解决的问题,流式传输技术开始崭露头角,并立即受到重视而且逐渐得到了应用。不同于下载方式,流式传输把声音、影像或动画等信息经过特殊的压缩方式分成一个个压缩包,由音/视频服务器向用户计算机连续、实时地传送。让用户一边下载一边观看。与单纯的下载方式相比,这种流式传输方式的优势如下^[1]:

1. 启动延时大幅度地缩短。在采用流式传输的系统中,用户不必等到整个文件全部下载完毕,而只需经过几秒或十几秒的启动延时即可在用户的计算机上利用相应的播放器进行观看。当音频、视频等媒体文件在客户机上播放时,文件的剩余部分将在后台从服务器继续传送。因此使启动延时成十倍、百倍地缩短。
2. 对系统存储容量的需求大大降低。由于不需要把所有的视/音频内容都下载到本地磁盘,仅在内存中缓存,并且通过丢弃已播放的内容,“流”可以重新利用空余缓存来存储后续未播放内容。

Internet 的迅猛发展和普及为流式传输技术的发展提供了强大的市场动力。流媒体业务正变得日益流行。根据统计,2000 年 Internet 上的视频流媒体技术应用增长幅度达到了史无前例的 215%,网络使用者在 2000 年所点播观看的流媒体节目多达 9 亿条,比起 1999 年成长超过两倍。流式传输技术广泛用于多媒体新闻发布、在线直播、网络广告、

电子商务、视频点播、远程教育、远程医疗等网络信息服务的方方面面。作为新一代 Internet 的标志,流媒体彻底改变了传统 Internet 只能表现文字和图片的缺陷,集音频、视频及图文于一体。它的应用为网络信息交流带来革命性的变化,对人们的工作和生活产生了深远的影响。流媒体必将成为 Internet 应用的主流,并将推动 Internet 整体架构的革新。

随着网络带宽瓶颈不断被突破,人们在分享流媒体业务所带来快乐的同时,又提出了新的要求,实时性好、传输数据量大、回放质量高成为人们对流式传输应用的更高要求。通过网络进行大数据量的流媒体传输,并且能最大限度的满足实时性,达到高质量的回放效果,这对流媒体数据的采集、传输、回放都是一个挑战。与传统的数据通信应用相比,多媒体应用具有连接时间长,传输数据多,涉及图形(graphics)、图像(image)、音频(audio)、视频(video)等不同媒体数据等性质,因而对连接管理、差错控制、流量控制的要求也各不同,同时一些应用还需要网络提供多点投递(multicast)和用户组管理的能力,所以对网络系统提出更新、更复杂的要求。目前,面向差错控制的传统网络协议(如 TCP 和 OSI 的 TP4)^[2],至少在如下几个方面已经不适应需求:第一,传统运输层协议过分依赖数据流发送方和接收方之间的反馈机制,而随着技术和应用的发展,一方生成的反馈信息在返回另一方的时间间隔中,已有大量新的数据注入网络,控制信息往往大大滞后于网络的动态变化;第二,传统网络协议过分依赖数据重发进行差错恢复,但在高速网络及应用中,差错率的升高意味着网络将处于拥塞状况,而大量的数据重发会加重网络拥塞,甚至导致网络运行瘫痪;因此,新型流媒体应用在功能和性能上都对传统协议提出了挑战。针对高带宽要求的实时流媒体传输系统,传输数据量大,视频回放质量高,有重要的应用价值,但达到这些目标的同时,伴随引起网络传输的种种不安全因素。目前, IETF 已经制订出关于 DV/HDV 等高带宽要求数据的实时传输封装格式,能够实时传输 DV/HDV 格式的流式传输系统也在高速网络中实现跨洋传输,例如: DVTS (Digital Video Transport System)^[2]。

与E-mail、WWW、FTP等TCP传输不同,视频的传输有以下特点^[3]:

1. 数据量大。以每帧 $512 \times 512 \times 8$ 位的视频来计算,一帧图像就占用大约2Mbits的空间,若以PAL制的每秒25帧计算,一秒中就要处理50Mbits的数据,这是文本、图形等媒体不可比拟的;
2. 延时敏感。视频传输要求连续,即每两个相邻数据,每两帧之间都有一定的延迟极限,超过了就会引起图像的抖动,也就是说如果有的视频数据帧发生延迟,可能造成在同一幅画面上各点发生的时间不同步,从视觉效果上看物体存在变形。
3. 对数据丢失率的要求。数据的丢失率会导致同一画面的各个区域视觉效果不同,有的区域清晰,有的区域模糊。
4. 具有最低接收质量的要求。就象看电视一样,观众对画面的视觉效果有一定的要求,效果太差,观众会拒绝收看。

5. 需要较高的质量要求。在传输视频数据时, 由于数据量较大, 即使出错, 一般也不要求重传, 因为对于微小的错误, 用户不易察觉, 但较高的图像质量则可弥补网络较低的传输率。

流式传输的发展面临着机遇, 更面临着挑战。为促进流式传输, 这种最为经济的传输方式的发展, 迎接高质量流式传输应用时代的到来, 相关的研究工作必须走在应用的前列。

1.2 课题研究的意义

视频传输的上述特点其实就表现了对网络带宽和延迟保证的要求, 将视频传输的特点和TCP(Transport Control Protocol)协议对照起来看, TCP协议是采用反复重传来保证传输的可靠性, 它虽然保证了传输的可靠性, 但是与视频传输的实时性相抵触(造成延迟、导致接收质量的降低), 因此采用单纯的TCP协议进行视频传输是不合适的。实时传输系统在传输层普遍采用UDP(User Data Protocol)协议对数据包进行封装, 但网络中传输的大量数据在传输层是采用TCP封装的形式, 大量UDP报文长时间占用带宽, 势必造成与TCP数据流之间产生带宽竞争。TCP协议本身提供拥塞退避的机制, 能根据传输过程中网络负载情况及时进行调整, 但UDP协议本身却不提供拥塞退避机制; 当前的Internet又是一个“尽力而为”的网络, 竞争的结果将会产生网络传输过程中带宽分配的不公平性发生。所以对于实时流媒体的应用, 必须要根据网络的现状找到一条折衷的办法, 充分加大流式传输数据量的同时, 与网络中的其他应用和平相处。本课题的研究作为WIDE Project项目的一部分, 通过流式传输协议、网络拥塞控制机制研究, 结合DVTS系统的成功经验和遇到的问题, 着重研究高质量实时视频流传输过程的拥塞控制机制, 研究成果将为设计符合实际应用的大数据量实时传输系统提供理论依据, 并在加速大数据量实时流媒体传输应用的同时, 确保网络的安全。在流式传输应用激增的今天, 研究工作具有重大意义。

1.3 课题研究的基础和条件

兰州理工大学现代网络教育学院网络实验室已经加入了日本Keio大学WIDE Project项目组, 着重进行DVTS项目应用的深入研究。双方有良好的沟通渠道, 能同步获得最新的研究成果; 现代网络教育学院于2003年10月顺利完成了教育部科学技术研究重点项目“IPv6网移动接入的实现”, 几年来的研究工作积累了许多成功的经验; IPv6(Internet Protocol Version 6)^[4]做为下一代互联网协议具有相对当前IPv4协议许多的优点, 针对网络拥塞控制提出了根本性的解决方案, 高质量实时流媒体应用在下一代网络中有广泛的应用前景, DVTS的研究也是面向下一代网络应用的研究课题; 网络实验室现有能进行网络状况分析、数据包分解的Fluke链路分析仪, 各种配置的试验用机, 可以构建不同的实验环境; 网络实验室现已成功接入了中国最大的纯IPv6实验网, 在

网络状况分析、协议分析方面积累了丰富的经验；大数据量流式传输的应用研究正是建立在现有的各项研究基础上，才使研究工作得以顺利完成。

1.4 课题研究的成果

在课题的进展过程中，研究了流式传输方式，实时流媒体传输协议；对 TCP 协议的拥塞控制机制进行深入研究的同时探讨了 TCP 友好(TCP-Friendly)的拥塞控制机制；结合网络状况分析的方法，对影响实时传输视频回放效果的因素进行了讨论，最终提出了基于 RTCP 反馈的大数据量流媒体实时传输的拥塞控制机制，并在全国核心期刊《计算机应用》发表相关论文，顺利完成研究课题。

1.5 论文的组织结构

第一章. 阐述了课题研究的背景、意义、基础和条件以及研究的成果。

第二章. 流媒体与流式传输协议研究

第三章. 拥塞以及拥塞控制机制研究

第四章. DVTS 传输系统的研究

第五章. 大数据量实时流媒体传输拥塞控制机制

最后. 总结和展望。

第 2 章 流式传输协议研究

因特网的普及以及多媒体技术在因特网上的广泛应用,迫切要求解决实时传送视频、音频、图像、动画等多媒体文件的技术。在这种背景下,流式传输技术及流媒体应运而生。

2.1 流媒体与流式传输

2.1.1 流媒体

流媒体是指在 Internet/Intranet 中使用流式传输技术的连续时基媒体,与传统的单纯下载相比较,流媒体具有明显的优点:由于不需要将全部数据下载后再播放,等待时间大大缩短;流文件往往小于原始文件,用户也不需要将全部流文件下载到硬盘,从而节省了大量的磁盘空间;采用了 RTP^[5] (Realtime Transport Protocol)等实时传输协议,更加适合动画、视频音频在网上的实时传输。流媒体(Streaming Media) 技术涉及流媒体数据采集、视/音频编解码、存储、传输、同步、播放等领域。

2.1.2 流式传输

流式传输是通过网络进行多媒体数据传输的方式。实现流式传输有两种方法:渐进流式传输(Progressive streaming)、实时流式传输(Real-time streaming)。渐进流式传输采用顺序下载方式,用户可以观看在线媒体节目,但在给定时刻,用户只能观看已下载的部分,不能跳到未下载的后序部分,并且不能根据用户的连接速度做调整;渐进流式传输方式适合高质量的短片段,如片头、片尾和广告。渐进流式文件要放在标准 HTTP 或 FTP 服务器上,易于管理,基本上与防火墙无关。但渐进流式传输不适合长片段和有随机访问要求的视频、讲座、演说与演示等,更不支持现场广播;实时流式传输,是在用户方收到视频数据的同时并对视频进行播放,采用实时流式传输,用户不用等待所有的视频数据完全传输完毕,再进行播放。由于视频数据是一边下载一边播放,将占用较小的磁盘空间,并可对客户端设备不作过高的要求,实时流式传输具有重要的应用价值,可以广泛应用于视频会议、赛事直播、现场广播等多个方面。随着网络技术的发展,实时流式传输成为流式传输应用的主流,各项研究均围绕实时流式传输展开。

2.2 流式传输协议

流式传输媒体必然涉及到网络传输协议, IP 协议之上的传输层有两种通信协议:TCP 和 UDP。TCP 协议是一种面向连接的传输协议,提供端到端可靠、有序的信息传输机制。但 TCP 协议速率减半的拥塞退避机制容易引起多媒体流过大的速率波动,其每包确认、超时重发机制会引入较大的时延,同时由于 TCP 是面向连接的可靠传输协议,存在建立和关闭连接的开销,所以 TCP 协议不适合用于多媒体传输。UDP 协议提供无连接、不可靠的数据报服务,UDP 协议不提供拥塞退避机制,UDP 数据流在堵塞的网络环境中将大量抢占具有拥塞退避机制协议流的带宽,同时自身的丢包率也迅速增加,将带来系统拥塞崩溃(Congestion Collapse)的潜在危险,因此没有拥塞控制的 UDP 协议也不适合用于多媒体流的传输,只有采用合适的协议才能发挥流式传输的效用,保证传输系统的服务质量 QoS(Quality of Service)。Internet 工程任务组 IETF (Internet Engineering Task Force)是 Internet 规划与发展的主要标准化组织,已经设计出几种支持流媒体传输的协议,其中实时传输协议 RTP (Real-time Transport Protocol) 以及 与 RTP 一起提供流量控制和拥塞控制服务的实时传输控制协议 RTCP(Real-time Transport Control Protocol)定义了一个实时数据端到端传输的合理框架,以 RTP 框架为基础来实现语音、视频实时传输是比较合理的应用方案^[6]。作为研究的重要要内容,我们首先研究一下实时传输协议- RTP。

2.2.1 RTP 协议

RTP 由 IETF 的 A/V 工作组于 1996 年公布作为 RFC1889 正式文档,是专门为交互式语音、视频、仿真等实时媒体数据的端到端传输服务而设计的传输协议。RTP 包含两个紧密联系的部分:RTP 用于携带具有实时特性的数据;RTCP 用于监视服务质量并传送正在进行的会话中参与者的信息。RTP 以报文形式携带着媒体数据,而 RTCP 则利用 RTP 报文的相关传输质量信息构成 RTCP 报文,以监视服务质量并传送正在进行的会话中参与者的信息,它们能提供的服务类型则包括有效载荷类型标识、顺序号、时间戳、传送监测。RTCP 和 RTP 一起提供流量控制和拥塞控制服务。以建立在 UDP 上的应用为例,通常在程序启动时,会打开两个连续的端口,偶数端口给 RTP,奇数端口给 RTCP。在 RTP 会话期间,各参与者周期性地传送 RTCP 包。RTCP 包中含有已发送的数据包的数量、丢失的数据包的数量等统计资料,因此,发送端服务器可以利用这些信息动态地改变传输速率,甚至改变有效载荷类型。RTP 和 RTCP 配合使用,能以有效的反馈和最小的开销使传输效率最佳化,因而特别适合传输网上的实时数据。RTP 协议被 IESG(Internet Engineering Steering Group)采纳为 Internet 标准后,

包括 Microsoft 和 Intel 等公司在内的全球逾百家厂商联合签署协议，共同致力于建立一个基于 RTP/RTCP 协议的开放性平台，致力打造出“使基于 Internet 的音频、视频及数据通信变得象打电话一样简单”的环境。

在 RTP 分组格式中，与实时传输密切相关的字段是：序列号 (Sequence Number)、时间戳 (Timestamp) 和同步源标识 (Synchronization Source (SSRC) identifier)。RTP 提供的时间戳 (Timestamp) 信息是多媒体实时数据传输的重要信息，它既可用于数据的流内同步，也可用于流间同步。时间戳记录了发送方数据块 (如视频帧) 第一个字节的创建时间。接收方利用时间戳可维持数据接收的实时性，实现数据流的同步，完成 RTP 包的重组，并按照正常的速率回放数据。对于一些大的数据块，如视频帧，一个数据块被分成多个包，它们使用相同的时间戳，但仅靠时间戳不足以恢复数据包的顺序，并且 UDP 等传输方式也不保证数据包传输的顺序性。因此，RTP 提供了序列号 (Sequence Number) 以恢复数据包的顺序，实现丢包检测，为实时传输提供网络拥塞等信息。为了确保实时数据的接收方能够识别数据负载格式、回放数据，RTP 提供了负载类型 (payload type) 标识。负载类型标识指定了 RTP 数据包的负载格式及编码压缩方法。RFC1890^[7] 中定义了一些常见的负载类型 (如: MPEG-1/2、JPEG 视频、H.261 视频流等)，用户可以根据需要定义更多的负载类型。在实时数据传输过程中，根据网络的拥塞情况或 RTP 数据接收方的网络信道可及时调整负载类型，避免实时数据传输的中断，增强数据传输的交互性，RTP 还提供了信源标识 (source identification)，使接收方有可能获得有关发送方的信息。RTP 固定头信息如下图所示。

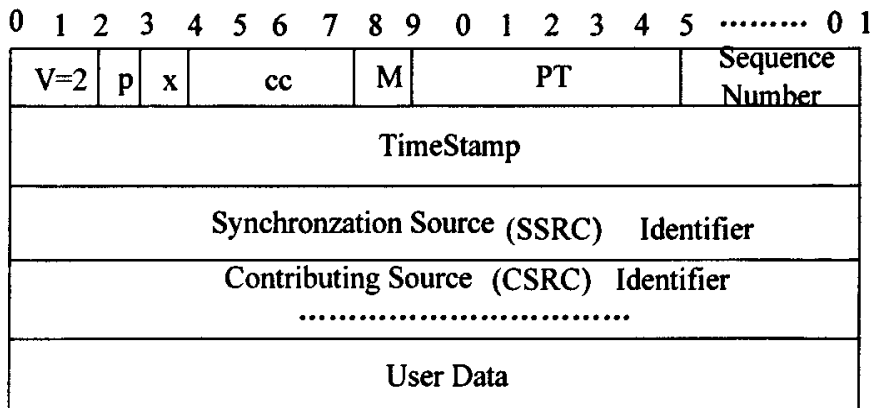


图 2.1 RTP 固定头信息

图2.1中，前三行是RTP包头，大小固定为12字节，其中每个字段有特定的意义。其字段代表的意义如下：

V: Version 版本号

- P: Padding 是/否还有补丁数据的标志位
- X: Extension 扩展头信息是否还需扩展
- CC: CSRC Count, CSRC计数, 表示固定头后有多少个CSRC标识符
- M: Marker, 标记
- PT: payload type 有效载荷数据的类型
- Sequence Number: 发送包的序列号
- Timestamp: 数据包发送的时间戳
- SSRC: 同步源标记, 用以识别同步源
- CSRC: 对载荷数据有作用的源标记

实时数据传输对于传输实时性的要求远高于传输可靠性, 因此 RTP 数据通常采用 UDP/IP 封装, 利用 UDP 提供的复用和校验功能, 共同完成网络传输层功能。UDP/IP 封装的 RTP 数据如图 2.2 所示。

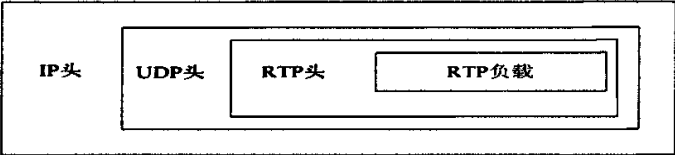


图 2.2 UDP/IP 封装的 RTP 数据

RTP并不作为一个独立的网络层来实现, 通常位于TCP和UDP协议层之上, 利用底层的UDP协议实现实时音/视频数据进行组播(Multicast)或单播(Unicast)。图2.3描述了这种体系结构。

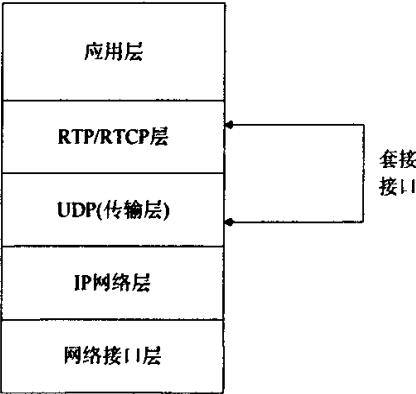


图 2.3 RTP/RTCP/UDP

RTP/RTCP 层从应用层接收多媒体码流, 组装成 RTP 数据包, 然后发送给下层 (UDP 协议) 来进行传输。媒体数据传输过程中, RTP 协议需要下层协议的支持, 依靠资源预留(RSVP:resource reservation protocol)、信令协议等提供通信路径上的资源预留, 实现为通信流提供可靠的服务质量保证。RTP 设计之初就是建立在传

输层协议之上的,可以充分利用传输层协议的复用和校验机制,由于不考虑底层网络结构,可以运用到多种网络环境中,可以运行在 IPX、TCP、IPv6 和 ATM AAL5 等多种网络协议之上。应当注意的是 RTP 本身并不提供任何机制来保证数据传输的时间和质量,也就是说:它所提供的实时服务既没有资源预约,也没有 QoS(Quality of Service)保证。但由于 RTP 提供扩展报头的功能,使其具有很强的应用扩展性^[6]。

2.2.2 RTCP 协议

RTCP 是作为配合 RTP 协议进行传输控制而设计的。实际标准也是以 RFC 文档的形式发布的(RFC 3550)。RTCP 的原理是使用与 RTP 传送报文,每隔一段时间就传送控制报文(control packet)。RTCP 主要的功能是了解数据分发的质量,配合 RTP 完成数据分发机制。如果流媒体是组播形式,可以让流媒体服务器提供这类 RTCP 例程(session),去诊断网络上的问题;RTCP 包含一个不变的 RTP 源(source)传输等级身份,叫做 canonical name 或是 CNAME,客户端依靠 CNAME 来追踪每一个参与者;RTP 也使用 CNAME 在一组相关的 RTP sessions 中,将来自同一个参与者的多个数据流进行联系。为适应大量的参与者,RFC3550 定义了 RR、SR、SDS、BYE、APP 等五类 RTCP 控制报文:

1. RTCP RR(Receiver Report)报文:由实时数据接收方发送,RR 报文针对每个信源都提供报文丢失数、已收报文的最大序列号、到达时间抖动、接收最后一个 SR 的时间、接收最后一个 SR 的延迟等信息。
2. RTCP SR (Sender Report)报文:由处于活跃状态的信源发送方发送,SR 报文不仅提供该端系统作为接收方的数据接收质量反馈信息(与 RTCP RR 报文相同),而且还提供 SSRC(同步源)标识符、NTP 时间戳、RTP 时间戳、发送包数以及发送字节数等与发送有关的信息。
3. RTCP SDS (Source Description Items) 报文:提供信源的描述信息。包括 CNAME(信源端系统标识)、NAME(用户名)、EMAIL(电子邮件地址)、PHONE(电话号码)、LOC(地理位置)、TOOL(应用程序或工具名)、NOTE(通知状态)等 SDS 报文项。
4. RTCP BYE 报文:将某参与者退出信息通知会话。
5. RTCP APP 报文:用户自己定义报文。

利用上述五类 RTCP 控制报文,可以实现如下服务:

(1) 媒体同步:RTCP SR 报文包含与 RTP 时间戳相对应的实时信息,可以象视频帧同步一样实现媒体同步。

(2) 信源标识: 在RTP数据包中, 信源采用不直观的32位SSRC进行标识, 而RTCP SDES包可提供具有文本信息的CNAME项来标识信源端系统、NAME项来标识用户名、EMAIL项来标识电子邮件地址、PHONE项来标识电话号码、LOC项来标识信源的地理位置, 方便实时数据传输的接收方获得发送信源的有关信息。

(3) 拥塞控制和QoS监控: 这是RTCP的一个重要功能。封装质量信息的RTCP报文可以利用RR和SR报文中一些相关字段来形成质量消息, 达到控制回放质量的目的。这些字段包括:

- 1) 丢失率: 从上一个SR或RR报文至今为止, RTP数据报文的丢失率。
- 2) 累计的丢失包数: 从上一个SR或RR报文至今为止, RTP数据报文的丢失计数。
- 3) 最高接收到的包顺序号: 从接收者接收到的RTP数据报文最高顺序号。
- 4) 到达时间间隔的抖动: 两个连续的包之间不同的到达时间间隔。
- 5) 最后到达SR时间戳: 最后到达的SR报文的时间戳。
- 6) 从最后到达SR至今发送RR的延迟: 接收到最后一个SR报文之后, 到发送本RR 报 文之间的时间间隔。

上面这些字段可以实时反馈网络的状况, 无论对发送方、接收方还是网络管理员, RTCP提供的数据传输反馈信息都是非常有用的。发送方可根据RTCP RR报文调整数据实时传输方式, 保障视频流的稳定性和健壮性, 使端系统能够正常接收; 接收方可确定网络拥塞的范围是在本地、本区域还是全局, 有的放矢地采取对策; 网络管理员可及时监视网络实时传输的性能。由于RTCP报文是周期发送的, 随着实时传输的参与者不断增加, RTCP报文的频繁发送对网络传输非常不利, 为防止RTCP报文占用过多的网络资源, 协议规定报文最多占用实时传输带宽的5%, 根据参与者数量可自动调节RTCP报文的发送频率, 及时避免报文过多占用网络资源。RTCP报文中的接收者报文如下图2.4所示, 其中每一字段也都有不同的意义。

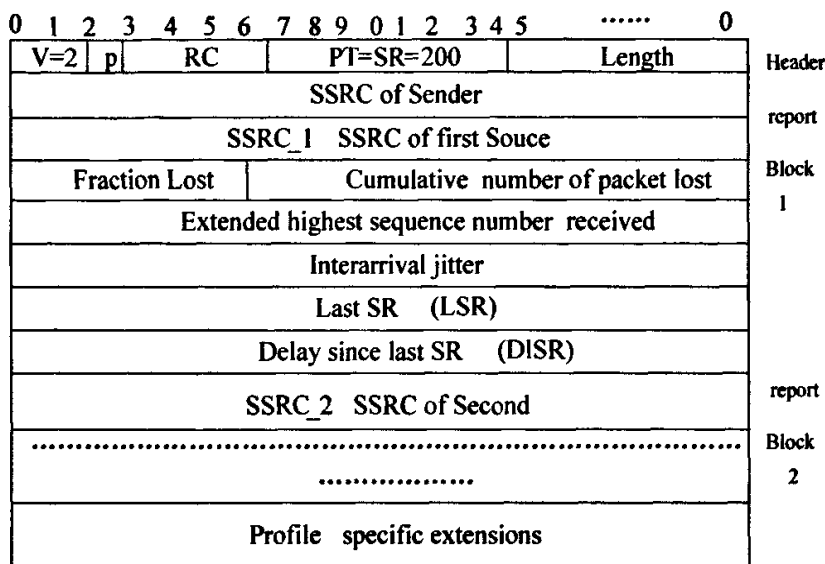


图2.4 RTCP报文中的接收者报文

V: Version 版本号

P: Padding 是/否还有补丁数据的标志位

RC: reception report count 本报文中的接收报告数量

PT: payload type 有效载荷数据的类型

Length: 报文长度

SSRC: 发送者的同步源标记

SSRC: 本报文所保持的同步源标记

fraction lost: 丢失率

cumulative number of packets lost: 累计的丢失包数

extended highest sequence number received: 最高接受到的包顺序号

interarrival jitter: 到达时间间隔的抖动

last SR timestamp(LSR): 最后到达SR时间戳

delay since last SR(DLSR): 从最后到达SR至今发送RR的延迟

2.3 小结

从 RTP最初的设计思想来看,体现了如下特点^[8]:

1. 轻型的传输协议。RTP 提供端到端的实时媒体传输功能,但并不提供机制来确保实时传输和服务质量保证,协议本身相对轻型,快捷。由于 RTP 没有像 TCP/IP 那样完整的体系框架,只是一个轻型的传输协议,主要与具体应用结合在一起来实现特定的目标。

2. 灵活性。体现在把协议机制与控制策略的具体算法分开, 协议本身只提供完成实时传输的机制, 对控制策略的有关算法实现不作具体规定。开发者可用根据不同的应用环境, 选择实现效率较高的算法与合适的控制策略。
3. 协议独立性。RTP 协议与下层协议无关, 可以在 UDP/IP, IPX, ATM 的 AAL(ATM Adaptation Layer)层上实现。
4. 扩展性。既支持传统的单播(Unicast)使用, 又支持组播(Multicast)使用。
5. 控制信息与媒体数据分离。RTP 控制分组采用与 RTP 媒体数据分组不同的“带外”方式发送。
6. 安全性。RTP 协议在设计上考虑到安全功能支持对数据加密和身份鉴别认证功能。

人们通过对 RTP 协议的研究, 已经在多方面增强了实时流媒体的应用。进行流媒体传输, 拥塞控制的研究也应该是建立在 RTP/RTCP 的基础上, 由于传输过程中的传输策略与网络状况是密切相关的, 所以说对于网络拥塞控制机制的研究也是研究工作的基础。

第3章 拥塞及拥塞控制机制研究

作为一个新兴领域,实时视频传输无论在科学研究还是在工业领域都是一个很有潜力的研究课题。然而,现存的Internet及其通信协议本身并不是为实时通信而设计的,在其上传输实时视频信号时会有几个主要的性能问题,如带宽的波动、传输性能不可预测、QoS无法保证等。所以在实现视频传输时,我们必须依据网络的动态状况来实时调节信号的传输速率,采取有效的速率控制算法,以避免由网络拥塞而导致的分组丢失以及传输延迟而造成播放质量的剧烈波动,提高视频传输的服务质量。从1986年10月开始,Internet出现了一系列的拥塞崩溃现象,网络吞吐量急剧降低,许多分散在各地的网点被迫长时间停止服务。随着计算机网络规模的不断扩大,用户急剧增长,拥塞状况越来越严重,这已经成为制约网络发展和应用的一个瓶颈,拥塞以及拥塞控制机制已经成为当前研究的一个热点问题。

3.1 拥塞概述

当主机转储到通信子网中的分组数量在其传输容量之内时,它们将全部发送达目的地,且传送到数量与发送的数量成比例。然而,当通信量增加太快时路由器不再能够应付,开始丢失分组,导致数据包的送达率下降。在通信量非常高的情况下,几乎没有分组能够送达。即:当(一部分)通信子网中有太多的分组时,其性能下降,这种情况就产生了拥塞(congestion)。

3.1.1 拥塞产生原因

网络产生拥塞的根本原因在于用户(端系统)给网络提供的负载大于网络资源容量和处理能力,表现为数据包时延增加、丢弃概率增大,上层应用系统性能下降等。拥塞产生的直接原因有以下三点^[9]:

1. 存储空间不足。如果几个输入数据流共用同一个输出端口,那么在这个端口就会建立排队。如果没有足够的存储空间,数据包则会被丢弃。对突发数据流更是如此。增加存储空间在某种程度上可以缓解这一矛盾,但当路由器有无限存储量,拥塞只会变得更坏,而不是更好,因为在网络里数据包经过长时间排队完成转发时,它们早已超时,源端认为它们已经被丢弃,而这些数据包还会继续向下一个路由器转发,从而浪费网络资源,加重网络拥塞。
2. 带宽容量不足。低速链路对高速数据流的输入也会产生拥塞,所有信源发送的速率 R 必须小于或等于信道容量 C 。如果 $R > C$,在理论上无差错传输则是不可能的。所以在网络低速链路处就会形成带宽瓶颈,当其满足不了通过

它的所有源端带宽的要求时,网络就会发生拥塞。

3. 处理器处理能力弱、速度慢。如果路由器的CPU在处于排队缓存、更新路由表等功能时,处理速度跟不上高速链路,也会产生拥塞。

简言之,拥塞是网络系统各部分数据传输和处理能力不匹配的结果。而随着网络规模的膨胀和复杂度的增加,要求一个数据流传输过程中涉及的各个部分达到完全匹配是几乎不可能实现的,拥塞是不可完全避免的,而拥塞一旦发生往往会形成一个不断加重过程。例如,如果路由器没有空余的缓存,它就必须丢弃新到的数据包,当数据包丢弃时,源节点端就会超时从而重传该包,由于没有得到确认,源节点端只能保留数据包,结果缓存会进一步消耗,加重拥塞。所以,采用合理的算法与机制确保网络不因传输数据流过大耗尽网络资源节点而导致崩溃就成为极为重要的工作,这就是一般意义上的拥塞控制。

3.1.2 拥塞控制策略的目的与任务

拥塞控制的主要任务是保证子网不被用户发送的数据所淹没,如图3.1所示,当负载比较小时,网络的吞吐量随负载逐渐增加,当负载达到网络最大容量时,吞吐量停止增加,若负载继续增长,则开始排队、丢包,当负载超过 *Cliff* 这点时,吞吐量急剧下降,网络发生拥塞。拥塞控制策略的目的就是在网络接近于拥塞崩溃点时能尽快检测到,并采取措施减小网络负载,使网络恢复到正常状态,保持在 *Cliff* 点左侧运行。拥塞控制又可分为拥塞避免和拥塞控制两个阶段^[10]。在 *Knee* 点处吞吐量的增长速度减小,而网络时延加速增长,在该处采取的策略成为拥塞避免策略,其目的是在没有显著的时延时,仍可以让用户适当增加通信量,当时延明显增加时,再要求其减少通信量,从而使网络负载在 *Knee* 处振荡;但是当突发的通信量使网络负载接近 *Cliff* 点时,必须采取拥塞控制策略来减少用户的通信量。

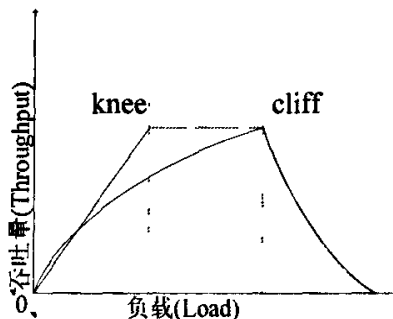


图3.1网络负载与吞吐量之间的关系^[11]

设 $power$ 为网络传输能力, $powerMax$ 为 *Knee* 点网络的传输能力, $Throughput$ 为网络传输吞吐量, $Delay$ 为传输延时,一般地采用如下公式计算资源利用率

$$Efficiency = \frac{power}{powerMax} \quad (3.1)$$

在 *Knee* 处 *power* 和 *powerMax* 相等, 故网络资源利用率达到100%。其中 *power* 可由下式计算

$$power = \frac{Throughput}{Delay} \quad (3.2)$$

进行拥塞控制的最终目的是网络资源的使用效率最大, 整个网络的资源利用率控制在 *Knee* 和 *Cliff* 之间。

3.1.3 拥塞控制的基本策略

针对拥塞控制的两个阶段, 制定拥塞控制的机制, 拥塞控制的算法一般分为两类^[12], 即开环与闭环。开环算法致力于通过良好的设计来避免问题的出现, 确保问题在一开始就不会发生, 一旦系统安装并运行该算法, 就不再作任何中间阶段的更正, 但网络状况是不断变化的复杂系统, 所以开环控制不是理想的选择; 闭环的解决方案是建立在反馈环路概念基础上的, 其中按反馈方式可分为显式反馈和隐式反馈两种。在显式反馈算法中, 由拥塞点将拥塞信号反馈给源端, 而在隐式算法中, 源端通过局部观察来推断是否存在拥塞。拥塞控制策略主要包含三个部分:

1. 监视系统, 检测何时何地发生了拥塞。
2. 将此信息传送到可能采取行动的地方。
3. 调整系统操作以更正问题。

基于RTP实时流媒体应用, 在传输层普遍采用UDP作为传输层协议。作为传输层的协议TCP和UDP, TCP协议是具有拥塞控制机制的, 通过传输过程中的三次握手, 以及滑动窗口技术进行发送速率的调整, 而UDP本身没有拥塞控制机制。现实Internet 网络中大多数的数据传输是通过TCP方式来进行的, 多媒体业务由于自身的特点, 普遍采用UDP作为传输层协议, 随着实时流媒体应用的发展, 特别是高带宽要求实时流媒体应用的开展, 大量的流媒体数据采用UDP的方式传输, 必将加大网络拥塞发生的可能, 还会引不同协议流之间带宽的竞争, 这些都将给网络安全造成潜在的威胁。所以, 在扩大实时流媒体应用的同时, 必须对多媒体应用对网络各方面造成的影响有全盘的考虑, 制定出有效的应对策略。针对多媒体应用的拥塞控制策略的制定, 可以从对TCP的拥塞控制机制的研究中获得启发。

3.2 TCP的拥塞控制机制

TCP协议是在RFC793^[14]中正式定义的，TCP是一项从实践中诞生，并在实践中不断得到发展和完善的网络技术，因特网中广泛采用的端节点拥塞控制协议就是TCP。TCP的拥塞控制是采用基于窗口的闭环拥塞控制协议。

1988年Jacobson指出了TCP在控制网络拥塞方面的不足，并提出了“慢启动”(Slow Start)、“拥塞避免”(Congestion Avoidance)算法^[15]。1990年出现的TCP Reno版本增加了“快速重传”(Fast Retransmit)，“快速恢复”(Fast Recovery)算法，避免了网络拥塞不够严重时，采用“慢启动”算法而造成大量减小发送窗口尺寸的现象，这样TCP的拥塞控制就由四个核心部分组成，最近几年又出现TCP的改进版本如New-Reno、SACK等。TCP拥塞控制是通过控制一些重要参数的改变实现的。TCP用于拥塞控制的参数主要有：

1. 拥塞窗口(cwnd)，是拥塞控制的关键参数，它描述源端在拥塞控制情况下一次最多能发送数据包的数量。
2. 通告窗口(awin)，是接收端给源端预设的发送窗口大小，它只在TCP连接建立的初始阶段起作用。
3. 发送窗口(win)，是源端每次实际发送数据的窗口大小。
4. 慢启动阈值(ssthresh)，是拥塞控制中慢启动阶段和拥塞避免阶段的分界点。初始值通常设为65535 bytes。
5. 回路响应时间(RTT)，是一个TCP数据包从源端发送到接收端，源端收到接收端确认的时间间隔。
6. 超时重传计数器(RTO)，描述数据包从发送到失效的时间间隔，是判断数据包丢失与否，网络是否拥塞的重要参数。通常设为2RT 或4RT。
7. 快速重传阈值(tcpexmtthresh)，是能触发快速重传的源端收到重复确认包ACK的个数。当此个数超过tcpexmtthresh时，网络就进入快速重传阶段，tcpexmtthresh缺省值为3。

TCP拥塞控制的三个阶段：

1. 慢启动阶段：旧的TCP在启动一个连接时会向网络中发送许多数据包，由于一些路由器必须对数据包排队，所以这样就有可能耗尽存储空间，从而导致TCP连接的吞吐量(throughput)急剧下降。避免这种情况发生的算法就是慢启动，当建立新的TCP连接时，将拥塞窗口(cwnd)初始化为一个数据包大小(一个数据包缺省为536或512bytes)，源端按cwnd大小发送数据，每收到一个ACK确认，cwnd就增加一个数据包发送量。很显然，cwnd的增长将随RTT呈指数级

(exponential)增长:1个、2个、4个、8……。源端向网络中发送的数据量将急剧增加。

2. 拥塞避免阶段:当发现超时或收到3个相同ACK确认帧时,网络即将发生拥塞,TCP假定由传输引起的数据包损坏和丢失的概率很小(小于1%),则进入拥塞避免阶段。慢启动阈值(sssthresh)被设置为当前cwnd的一半,如果还是超时,cwnd要被置1;如果此时cwnd小于或等于sssthresh,TCP就重新进入慢启动过程;如果cwnd大于sssthresh,TCP就执行拥塞避免算法,cwnd在每次收到一个ACK时只增加 $1/cwnd$ 个数据包(这里数据包大小segsz假定为1),所以在拥塞避免算法中cwnd的增长不是指数的,而是线性的。

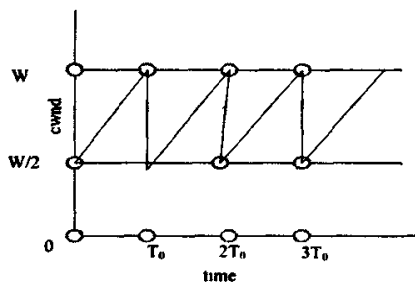
3. 快速重传和恢复阶段:当数据包超时,cwnd要被置为1,重新进入慢启动,这会导致过大地减小发送窗口尺寸,降低TCP连接的吞吐量。所以快速重传和恢复就是在源端收到3个或3个以上重复ACK时,就断定数据包已经丢失,重传数据包,同时将sssthresh置为当前cwnd的一半,而不必等到RTO超时。

3.3 TCP流量控制

TCP有着明确的协议状态,不同的网络配置和状态(调度、缓存)都会影响TCP的流量特性。TCP拥塞控制机制的核心是加法增加乘法减少(AIMD: Additive Increase Multiplicative Decrease)算法。它可以用数学式表示为

$$\begin{aligned} I: W_{i+r} &\leftarrow W_i + \alpha; \alpha > 0 \\ D: W_{i,s} &\leftarrow \beta * W_i; 0 < \beta < 1 \end{aligned} \quad (3.7)$$

其中 I 表示因为在RTT内接收到ACK包而引起的窗口增加, D 表示遇到拥塞后窗口减少, W_i 是 t 时刻窗口的大小, R 是RTT, α, β 是常数。Floyd最早分析了加法增加/乘法减少的拥塞避免算法的流量特性。她得出了稳态下分组丢弃率、连接往返时间、拥塞的网关数目和窗口增加/减少算法之间关系的一些解析公式^[16], 近期研究表明, 稳态时TCP连接的吞吐量是链路数据包丢失率和往返时延的函数^[17]。下面我们利用一个“稳态”模型简单地分析一下TCP的吞吐量特性。考虑一个有固定往返时延 R 和固定分组大小 L 的TCP连接, 设网络是稳态的, 即排队时延与往返时延相比可忽略, 平均分组丢弃率 p 较小, 因此可以不考虑慢启动和快速恢复过程的影响。根据加法增加/乘法减少的控制规则, TCP将工作在一个以 T_0 为周期的稳定周期状态, 拥塞控制窗口 $cwnd$ 的变化如图3.2所示


 图3.2 稳态下的 $cwnd$ 性能评估^[18]

在每个周期中, $cwnd$ 一直增大到最大值 W , 这时一个数据包丢失使得 $cwnd$ 下降到 $W/2$, 根据稳态假设, 每经过一个往返时延 R , TCP 都会发送全部拥塞窗口内的数据包且 $cwnd$ 会增加1, 所以:

$$T_0 = \frac{W}{2} R \quad (3.8)$$

假设, 发送速率近似为 W/T 。所以在每个周期 T 内, 发送的数据包数目 N 为:

$$N = \int_0^{T_0} \frac{W(t)}{R} dt = \frac{3}{8} W^2 \quad (3.9)$$

因为每个周期只有一个数据分组丢弃, 所以: $p = 1/N$, 将式(3.8)带入可解得

$$W = 2\sqrt{\frac{2}{3}} \frac{1}{\sqrt{P}} \quad (3.10)$$

因此平均吞吐量 T 为

$$T = \frac{(N-1)L}{T_0} = \frac{(\frac{1}{P}-1)L}{\sqrt{\frac{2}{3}} \frac{1}{\sqrt{P}} R} \quad (3.11)$$

当 p 非常小时, 上式可近似为:

$$T \approx \frac{LC}{R\sqrt{P}}, C = \sqrt{\frac{3}{2}} \quad (3.12)$$

可见, TCP的吞吐量是与丢包率的平方根的倒数成正比的。

不少学者进一步研究了TCP流量特性。Lakshman等推导了在考虑反向拥塞情况下的TCP吞吐量公式, 公式中吞吐量与分组丢弃率平方根的倒数成正比, 但比例系数为数据分组的传输时间除以ACK分组的传输时间。此外, Padhye等建立了一个比较完整的稳态下TCP的吞吐量模型, 他推导出稳态时TCP吞吐量 T 可用下式表示

$$T = \min(W_{\max} / R, s / (R\sqrt{2bP/3} + t_{RTO} \min(1, 3\sqrt{3bP/8})P(1 + 32P^2)) \quad (3.13)$$

其中, W_{\max} 为拥塞控制窗口的最大值, R 为往返时延, S 为分组大小, t_{RTO} 为重发定时器的值, b 为每一个ACK分组确认的分组数, P 为分组丢失率。这一模型考虑了快速重传和超时重发的情况, 因此得到了广泛的引用。

TCP的拥塞控制机制适合于传输成块的数据(bulk data)业务, 由于采取慢启动, 超时重传, 以及AIMD的速率控制机制, 在进行多媒体业务传输时, 速率减半会引起多媒体数据传输速率的过大抖动, 将明显降低用户端可察觉的质量, 所以对于多媒体应用是很不利的。在Internet这样的分享网络中, 所有的端系统都必须在出现(消失)网络拥塞时减少(增加)自己的发送速率, 只有这样才能避免网络崩溃和提高网络利用率。所以一种理想的多媒体传输方案必须要具有速率调节功能, 由于目前Internet上主要的业务都是基于TCP协议的, 所以传输层协议之间的公平性(inter-protocol fairness)也成为网络不安全的引发因素。为了有利于多媒体应用的发展, 使基于UDP的多媒体应用与基于TCP的应用最大可能的共享带宽, 研究人员提出了TCP友好(TCP-Friendly)的概念以及基于TCP-friendly的拥塞控制机制^[19]。

3.4 TCP 友好拥塞控制机制

因特网逐步进入商业应用后, 网络中各协议流之间广泛合作进行拥塞控制的假设将不能完全成立^[20]。不少网络应用, 尤其是交互式多媒体应用, 很少采用或根本不用拥塞控制机制, 从而造成了网络带宽的不公平竞争。Floyd对不公平竞争带宽的流做了如下定义: 非响应流, 这种流对路由器不断增加的包丢失率, 没有反应, 即并不降低其发送速率; 非TCP友好流: 这种流的长期平均速率超过了同等情况下符合TCP协议流的速率。为了扩大多媒体应用而设计新的拥塞控制机制时, 既要适应应用的需求, 又要使多媒体流与TCP业务流具有友好性, 这就是当前流量控制与优化研究领域的一个热点—TCP友好的拥塞控制。

TCP友好性的定义是: 连接在长时间范围的平均吞吐量, 不超过相同条件下TCP连接的平均吞吐量, 这样的协议流称为是TCP友好的。Rejaie最早提出了一种在长时间范围内符合TCP友好性的、基于速率的拥塞控制机制。Padhye等也比较早地提出了一种TCP友好的、基于速率的流量调节算法, 该算法每轮经过 m 个时间单位, 由发送方估计往返时间, 若有分组在本轮丢失, 则发送方将把速率设里成为TCP在同等丢失率下的速率; 若本轮没有分组丢失, 则发送方将加倍发送速率。在此基础上, Floyd等于2000年提出了一种完整的符合TCP友好性的拥塞控制协议。此外, Vicisano将TCP友好性思想应用到组播拥塞控制机制的设计中, 为分层的组播业务提出了一种拥塞控制算法^[21]。

目前提出的TCP-friendly协拥塞控制策主要有以下三种^[22]:

- 1) 基于AIMD的控制机制, 它与TCP拥塞控制机制相似, 也通过调整发送端窗口的大小对网络拥塞作出反应, 在窗口增加阶段

$$\begin{cases} w_{i+r} = \min(Rcvr_Adv, 2w_i; w \leq \frac{1}{2}w \\ w_{i+r} = \min(Rcvr_Adv, w_i + \alpha; \alpha > 0 \end{cases} \quad (3.14)$$

在窗口减小阶段

$$w_{i+r} = (1 - \beta)w_i; 0 < \beta < 1 \quad (3.15)$$

其中 $Rcvr_Adv$ 指接收端提供的窗口通告值, r 指往返时间, w 代表拥塞前的窗口大小。

- 2) 基于模型参数的控制机制: 它是根据TCP流控模型中发送端速率 T 和分组丢失率 P 之间的关系式(3.11)或(3.12)对发送速率进行调控, 使它们不超过相应条件下TCP流的传输速率, 以满足TCP-friendly的要求。
- 3) 基于AIMD和基于模型参数的结合: 如果基于AIMD控制机制不利于多媒体业务对平滑传输速率的要求, 或如果只采用基于模型参数的控制方式丢失率 $P > 16\%$, 公式(3.11)或(3.12)都不能准确反映TCP-friendly特性时, 将以上两种方式结合起来。

3.5 TCP友好的多媒体通信流控制机制

针对多媒体应用的流控机制都是建立在多媒体通信协议基础上的。近年来, 一系列适合多媒体应用的TCP-friendly拥塞控制协议被制定出来。

Rejaie等人也提出了一种适应性方案称为速率适应协议RAP(Rate Adaptation Protocol)。就像TCP协议, 发送端的分组得到接收端的确认, 并通过确认分组的序列号的间隙或分组是否超时来指示分组的丢失。通过确认分组, 发送端可以估计分组发送的往返时间RTT, 如果没有检测到分组丢失, 发送端可以周期性的增长它的发送速率, 增长率是分组往返时间的函数; 当检测到分组丢失时, 发送的速率就立即减少为当前的一半。

Padhye等人提出了一个TCP连接平均分享带宽的解析模型:

$$T_{tcp} = \frac{M}{t_{RTT} \sqrt{\frac{2Dl}{3} + t_{out} \min(1, 3\sqrt{\frac{3Dl}{8}})(1 + 32l^2)}} \quad (3.16)$$

其中 M 是分组大小, l 是丢失率, t_{out} 是TCP重传值, t_{RTT} 是往返延时, D 是被确认的TCP分组数目。通过这个模型, Padhye等人提出了一个方案, 发送端根据接收

端的确认估计传输往返延时和丢失率。在丢失的时候，发送端限制其发送速率在3.16所估计的值，否则他的发送速率可以加倍。

IETF A/V工作组在RFC3448^[23]中，详细描述了基于模型参数的TCP-friendly控制协议TFRC (TCP-Friendly Rate Control)。TFRC是一种基于公式的拥塞控制机制，它通过对TCP的控制机制进行分析，建立了TCP的吞吐量模型，推导出TCP的吞吐量公式(即与丢包率和往返时延的函数关系式)。发送端根据通过接收端定期的反馈，计算往返时间和丢包率，根据吞吐量公式计算出的值调整发送速率，保证与TCP流的友好性。虽然很多研究证明TFRC有很好的TCP友好性和平滑性^[24]，但是它只考虑了网络变化缓慢或静态的环境下的性能，并没有考虑网络剧烈变化时的拥塞控制机制的性能。已经有许多研究发现TFRC在网络状态变化频繁、剧烈时或丢包率很高的情况下，计算出的吞吐量并不精确，明显小于可用值，无法保证与TCP的友好性。而且基于公式的拥塞控制对网络状态变化的响应太慢，虽然保证了发送速率的平滑性，但在网络拥塞严重时很容易造成拥塞阶段的延长，造成丢包的加剧，反而使流的播放效果不佳。

3.6 小结

TCP协议提的拥塞控制显然是不利于多媒体传输的。对于实时多媒体应用，现有的TCP-friendly拥塞控制机制不能很好的满足人们对传输质量与时延的要求；对于高带要求的多媒体数据，接收端反馈的数据如果不能较好地接收，传输系统的服务质量将是不能忍受的。所以，多媒体数据进行实时传输要根据多媒体数据的特点制定相应的拥塞控制机制，通过多方面状况的综合分析进行合适的速率，只有这样才能充分保证传输的服务质量。

第 4 章 DVTS 流式传输系统的研究

多媒体传输可以通过许多的途径,流式传输方式作为最为经济的传输手段备受人们的青睐。由于流式传输技术本身所具有的特点,对于传输高带宽要求格式视频数据,例如: MPEG、DV/HDV 等,必须在传输系统的设计与实现过程中根据高带宽多媒体数据的特点采用恰当的编码方案;流式传输过程中,网络设备对数据包的丢弃是难免的,为了弥补传输过程中传输延迟和视频质量损失,更要有相应的拥塞控制策略与视频补偿措施。当前能够传输高带宽数据的传输系统取得了一定的成绩,作为本课题研究的出发点,我们将对开源的高带宽数据传输系统 DVTS 进行深入研究。

4.1 DVTS 流式传输系统

随着人们生活水平的提高, DV(Digital Video)/HDV(High Definition Video)等数码摄/录设备进入了普通人的生活。DV 格式采用帧内(intra-frame) 压缩,离散余弦变换 DCT(Discrete Cosine Transform)和变长编码 VLC(Variable Length Coding), DV 帧的音频数据采用 12bit 或是 16 bit 脉冲编码(PCM),一个 DV 视频帧能经过 DCT 变换被分成几个 DCT 超级块(Super block),其中每一个超级块又包含 27 个宏块(Macro block),每一个宏块又由 4 到 6 个数据块组成;一个 DV 视频帧在进行传输时可被分成多个数据接口格式(Digital Interface Format: DIF)序列,一个 DIF 序列包含 150 个 80 字节(byte)的 DIF 块,原始 DIF 块中包含有 3 字节(3byte)的标识位,用来标识 DIF 块的类型。从 DV 帧的格式组成方式,可以知道:部分 DV 数据的丢失对整体的 DV 流质量的影响是可以控制的^[24]。IETF A/V 工作组制订了关于 DV 格式数据的确的 RTP 封装格式相关草案,可以搭建 DV 数据的 RTP 封装实时传输平台, DVTS 系统的实现不但是一个具体的应用实例,也促成了 DV 的 RTP 封装成为正式的标准文档^[25]。

DV 的数据特点决定了传输 DV 数据必将占用大量的网络带宽,进行流式传输的多媒体格式主要有 Real Media、Quick Time、Windows Media、MPEG 等,研究表明进行多媒体流式传输的带宽需求如下图 4.1 所示。试验表明传输带宽需求在 25Mbps 以上的媒体数据全速传输的情况下能够得到较好的回放质量^[27],在研究过程中,我们将传输带宽需求在 25Mbps 以上的媒体称为高带宽媒体格式,进行这样媒体格式流式传输称为大数据量流媒体传输。

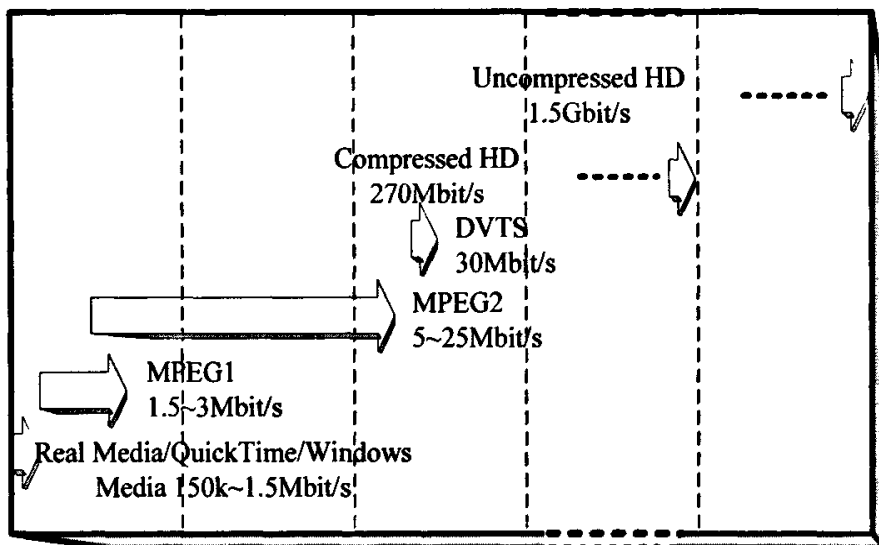


图 4.1 流媒体数据带宽需求

DVTs(Digital Video Transport System)是 WIDE project^[26]项目的一个研究成果。DVTs 使用过程中对传输带宽的要求是 35Mbps, 如果网络带宽充足, 利用 DVTs 可以提供电视质量的视频享受, 是大数据量、高带宽要求视频流实时传输系统的成功实例。DVTs 系统的应用效果图如下图所示。

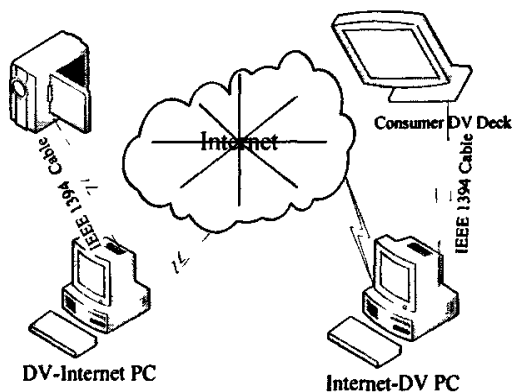


图 4.2 DVTs 系统效果图

DVTs 的研究开创了低成本、高质量的视频交流解决方案, 为高效、移动的视频传输奠定了基础。作为一个开放源代码的系统又为其他研究工作提供了良好平台。在 DVTs 系统中采用 RTP 协议作为实时传输协议, 整个传输系统的工作流程可以从发送端和接收端两方面体现。发送端通过 IEEE1394 接口, 将用 DV 设备采集的实时 DV 视频流进行编码, 封装成 IP 数据包, IP 数据包通过网络进行传输; 在接收端客户端程序解开接收到的 IP 数据包, 将 RTP/DV 数据包还原成 DV 视频

流，可以在普通 PC 或是通过 IEEE1394 接口连接到放像设备进行重放。整个系统延长了 IEEE1394 网络，采用 DV/RTP 封装方式，可以支持 IPv4 和 IPv6 等多种网络协议，并可以运用到组播环境中。

DVTS 系统已经成功运用到了远程监控，远程医疗、以及远程教育等多个方面^[28]。对于一个实时流式传输系统，在系统的设计过程中主要涉及到视频采集、RTP 封装、传输、回放等方面，现从这几个方面对 DVTS 系统进行研究，由于流式传输系统的服务质量是影响其应用的关键，当前的网络是一个异构、时变、缺乏 QoS 控制的网络状况，作为大数据量，高带要求的实时传输系统，在传输系统设计中也有相应的传输控制策略，这也是我们研究的关键所在。

4.2 DV 数据的 RTP 封装

IETF RFC 3189 对 DV 格式的 RTP 封装进行了相关的规范。DVTS 系统在发送端将采集到的 IEEE1394 数据包，整合成 DIF 序列进行 RTP 的封装。在 DV 视频流的 RTP 封装中，同一个视频帧的所有 DIF 块封装在同一个 RTP 数据包中。

图 4.3 为 DV/RTP 封装格式。

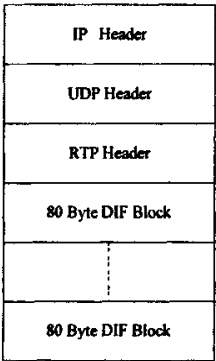


图 4.3(a)DV/RTP 封装格式

对于一个视频帧所有的DIF块都必须封装在一个RTP包中；传送不同的视频帧是通过RTP报头中的序列号来实现的，音频和视频数据的传送采用的是不同的策略：首先DV音频被转换成标准的PCM音频格式，然后进行封装、传输；音频流与相应的视频流是两个不同的RTP数据包，对于音频流和视频流数据的区分是通过RTP报头中不同的时间戳值来进行的。采用音频数据和频数据分别进行编码和封装，为传输控制策略的实施提供了保障^[29]。我们在局域网环境下进行了抓包分析，对DV/RTP的封装有了深入理解。

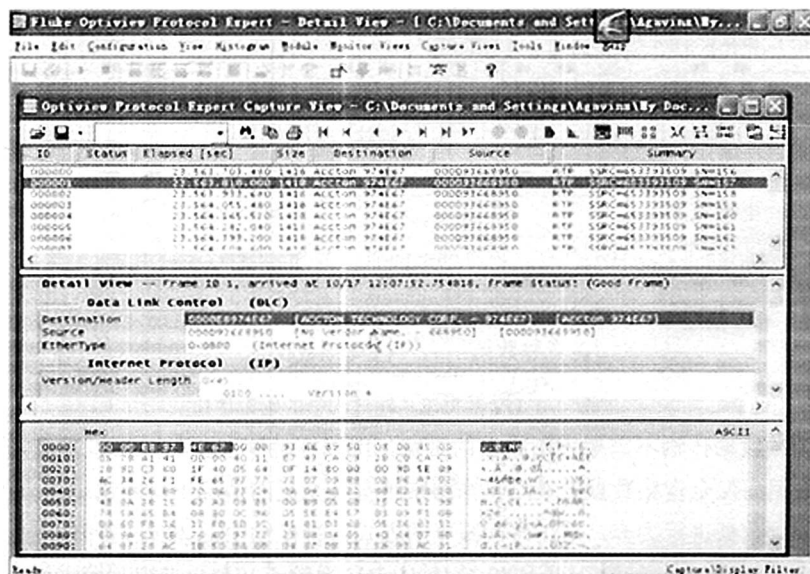


图 4.3(b) DV/RTP 封装格式

4.3 DVTS 中的传输控制策略

DVTS 也可以在带宽变换的网络环境下进行视频流传输。为了减少传输时延并能有很好的服务质量,就要对结合 RTP 协议制定相应的传输策略和视频补偿策略。

4.3.1 发送端与接收端的控制策略

网络拥塞的发生是伴随着流式传输进行的,发送速率的调整是拥塞控制的基本方法,码率切换是常用的方式。DVTS 系统就是采用码率切换的方式,对发送速率进行调整。在大多数情况下,用户并不需要视频数据的全速率传输,这样就可以降低帧速率。与此相反,音频数据不像视频数据需要占用那么大的带宽,但需要相对稳定和持续的传输。所以在一定程度上丢掉部分视频帧而保存所有的音频数据就可以有效地压缩 DV 流,而不会产生大的通信错误。如果有足够的带宽,在全速率传输的情况下,发送端程序只是简单地转发每一个 DIF 块给接收端,如果没有足够的带宽,发送端程序就将通过丢失适当的 DV 视频数据来降低输出速率。图 4.4 是 DV 流数据的传输时的码率切换情况。

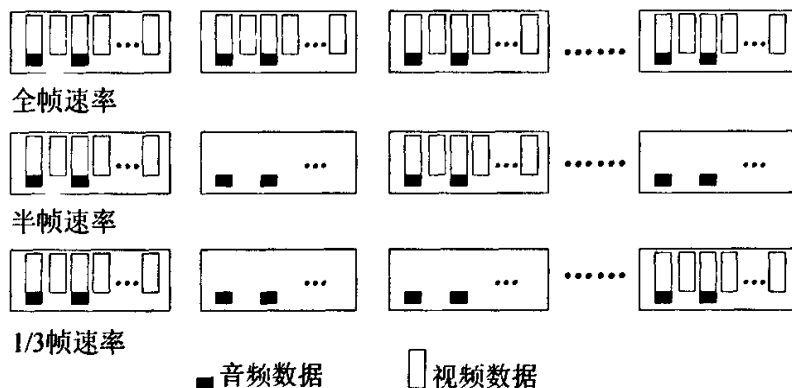


图 4.4 DV 数据流传输过程中的码率切换

由于音频数据传输不需要占用太多的带宽，所以在拥塞发生时，可以丢弃一定的视频数据而保全音频数据的策略。当网络带宽不充足的情况下，发送端将 DV 流中的视频数据进行丢弃，采用逐次减半的方式，将视频由 30 帧/秒降低到 15 帧/秒，以至于更少的帧数。对于 30 帧/秒的 DV 视频流减少一半视频帧进行发送，对于图像的质量影响是不明显的，但如果网络发生拥塞并且随着时间推移，拥塞情况却没有缓解的迹象，那么接收到的视频帧将是极其有限的。

由于采用丢弃视频数据的方式，在接收端也要采取相应的措施以弥补丢失视频帧所造成的空白。一般可以采用两种方式：(1) 将前一个时刻收到的视频数据进行重复解码、播放。(2) 将前一个时刻收到的视频数据的一部分重复进行回放。DVTS 系统是采用第二种方式，由于 DV 的格式特点，可以在以前的数据中很容易地找到特定位置的 8×8 的 DCT 数据。当一个包丢失的时候，接收方使用了一个以前的相关 DIF 块来填补丢失的内容。在 DV 格式中，DCT 块是分布式的，这样少量的包丢失不会导致整个图像出现严重的质量下降，为了实现视频补偿的过程，接收端程序应该能够将接收到的数据包进行缓存，也就是说，在接收端采用缓存区来进行视频补偿。

4.3.2 传输延迟、抖动平衡策略

由于 RTP 协议没有 QoS 保证，不能确保数据包完全到达接收端；在实时视频流传输过程中，为了适应带宽的变化要采取速率调整策略；这将给接收端收到的数据包带来一定的延迟，传输网络的负载状况也将会加大延迟的时间，数据包的非顺序性到达也将造成回放时视频流的抖动。DVTS 系统设计中，为了减少时延和抖动采用了共享缓冲区的策略来进行平衡延迟和抖动。共享缓冲区的方式如图 4.5 示：

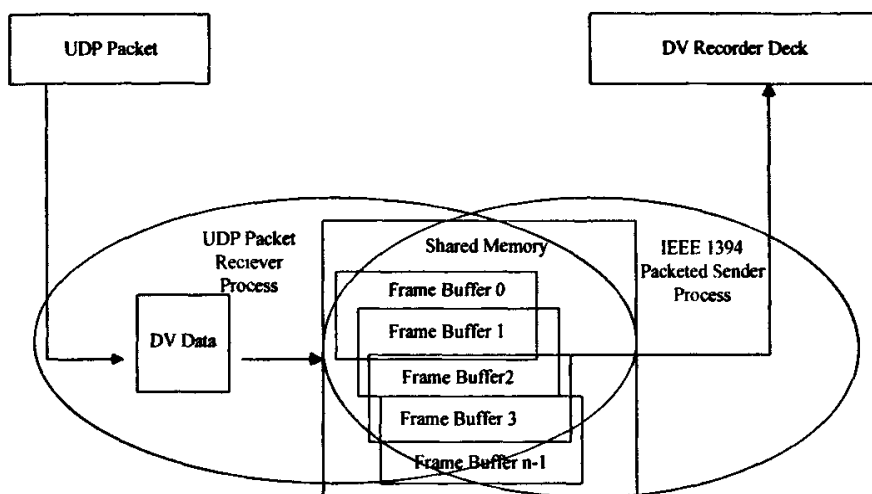


图 4.5 共享缓冲区

接收端程序有两个过程来完成对共享缓冲区的处理，一个过程用来将接收到的 IP 收据包进行解包，还原成 DV 数据放入缓冲区，另一个过程将缓冲区的 DV 数据包送到 IEEE1394 网络进行传送，还原成 DV 视频流，分别称这两个过程为解码过程和回放过程。解码过程只当接收到完整的 IP 数据包才对其进行解码，对共享缓冲区进行更新，并通知回放过程，当收到的数据包是零散不完整的，解码过程并不对缓冲区进行更新；回放过程收到解码过程的信息后读取缓冲区的数据，进行回放处理。正是采用共享缓冲区的方法，才使 DVTS 系统发送端速率的调整策略得以实现，同时在接收端对传输时延和抖动进行了有效的平衡。但缓冲区的数量和大小的设置对传输时延是有影响的。

4.4 小结

通过对 DVTS 系统的研究，可以得出这样的结论：首先随着高速网络技术的发展规律，高质量的实时流式传输服务已逐步走向了现实。DVTS 系统在成功的同时，又暴露新的问题，大量的视频流长时间占用高额带宽，将会造成 RTP/UDP 数据流与传统 TCP 数据流之间带宽的竞争，进而对网络的安全造成威胁，基于 RTP 结构的流式传输控制策略对服务质量的提高有重要的作用。当前的 DVTS 系统主要面临如下的问题^[30]：

1. 对于传输过程中，由于音频数据某一位(bit)的丢失，将使接收端的音频回放产生巨大的噪声。所以对 DV 数据中音频数据传输安全，成为应用的瓶颈。
2. 针对 MPEG-TS 等高带宽要求数据的传输要求，亟待提高传输系统的服务质量。

3. 大数据量实时传输系统可以实现不同事件的同步播放,不同地点的传输视频流时钟同步也是面临的问题。

DVTS 系统的实现,为大数据量实时传输系统的研究提供了良好的平台。影响高带宽要求视频实时传输服务质量的关键问题仍然是网络的拥塞,RTP/UDP 结构自身拥塞调节机制的缺乏,发送/接收速率调整策略、延迟和抖动消除方法,解决这些问题,提高传输服务质量的关键在于有切实有效的大数据量实时视频流传输拥塞控制机制。

第 5 章 大数据量实时流媒体传输拥塞控制机制

通过前面的研究,我们知道要进行大数据量流媒体数据的实时传输,主要从以下几个方面进行传输过程中拥塞控制。首先要根据传输媒体的具体情况来设计合适的面向质量的速率调整策略,其次由于传输数据量大势必加重网络的传输负担,在实时传输应用中,网络负载情况的了解与预测将对传输时延和传输质量产生重要的影响;传输过程中,拥塞发生时网络设备对传输数据包的丢弃是难免的,为了保证传输质量的最大化,在流式传输系统的接收端必须要有合适的视频补偿策略;多媒体应用的发展在增强应用与网络安全之间要达到平衡,也就是说要使建立于 UDP 协议之上的实时多媒体应用要达到 TCP 友好,只有满足这样的条件下,大数据量、高带宽要求的多媒体流式传输才能加速发展。通过课题的研究,我们提出了针对大数据量实时视频流传输的拥塞控制机制,并通过模拟试验对该机制进行了性能分析。

5.1 TCP 友好拥塞控制机制在实时视频流传输中的缺陷

目前提出的用于实时视频流传输的 TCP-friendly 拥塞控制可以分为两类:一种是基于 AIMD(Additive Increase Multiplicative Decrease)^[31]的控制机制;另外一类是以 TFRC(TCP-Friendly Rate Control)^[32]为代表的基于数学模型控制机制。

基于 AIMD 拥塞退避算法的主要特点是:加性增加和乘性减少的速率调节机制, RAP(Rate Adaptation Protocol)协议采用了 AIMD 的速率控制,并对速率改变进行平滑处理,采用每包确认机制,可以实现与 TCP 流公平分享带宽,但 AIMD 速率调整可能会带来突然、大幅的速率改变,引起视频流传输发生剧烈振荡,这对于实时流媒体应用是很不利的。

基于数学模型的控制机制来源于 Padhye 等提出的 TCP 稳态反应函数,这一反应函数是根据给定丢失事件率和链路回环时间得出平均吞吐量。

$$R_{mean} = \frac{S_p}{t_{RTT} \sqrt{\frac{2p}{3}} + 3p(1 + 32p^2)t_{RTO} \sqrt{\frac{3p}{8}}} \quad (5.1)$$

(5.1)式中 s_p 为数据包大小(单位为 Byte); t_{RTT} 为发送端与接收端之间的回环时间(单位为 s), p 为包丢率, t_{RTO} 为 TCP 传输时限(单位为秒 s)。为了达到 TCP-friendly,着重对高带宽流的发送速率进行控制,但速率调整较为平滑,一定程度有利于实时流媒体传输,但对于带宽要求较大的流媒体数据,采用这种方式在速率调整的同时,大量数据包将会被丢弃以平衡带宽,从而导致回放的视频质量

大幅下降,所以在速率调整的同时也必须根据视频帧对回放质量的影响程度,有选择的进行码率切换,并且根据网络负载状况对发送速率进行平滑处理。

5.2 原始视频帧对图像回放质量的影响

视频一般都是压缩后在网络中传输,网络拥塞控制使丢包现象的发生极为普遍,经过网络传输,有些数据包在接收端也可能无法解码,导致接收端回放视频质量较低,对于大数据量的实时视频流,高额的带宽要求加剧了丢包发生的可能,所以对原始视频帧对图像回放质量影响的研究是必要的^[11]。

通过网络传输后的视频数据进行解码,解码的图像质量通常采用峰值信噪比来表示其与原图的差别。峰值信噪比定义如下:

$$PSNR = 20 \times \lg \frac{255}{\left\{ \frac{1}{N_1 N_2} \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} [f(x,y) - f'(x,y)]^2 \right\}^{1/2}} \quad (5.2)$$

$N_1 \times N_2$ 为图像的大小, f, f' 分别是原始图像和通过网络接收到的数据的解码图像。根据式 (5.2) 当网络出现丢包时,视频数据受到影响, $PSNR$ 将随之变小,而且丢包越多, f, f' 差别越大, $PSNR$ 也越小。

当某一视频帧的 $PSNR$ 小于一定值时,就认为其质量是不满足要求的,而把该帧丢弃,假设 α 为丢帧率,那么

$$\alpha = \sum_{i \in \{I, P, B\}} p(f_i) \cdot p(K | f_i), \quad (5.3)$$

其中 i 表示 I, P, B 三种类型的视频帧, K 表示当某帧的 $PSNR$ 值小于某一设定值后,要把该帧丢弃, $P(f_i)$ 表示帧为 I, P, B 帧的概率, $P(f_i)$ 可预先从视频流分析得到。对于视频帧为 I 帧的情况,因为它可以独立解码,所以计算比较简单,可以表示为:

$$p(K | I) = 1 - (1 - \varepsilon)^{S_I} \quad (5.4),$$

其中 S_I 为在 I 帧中平均包含的数据包数, ε 为丢包率。对于 P 和 B 帧而言,每个 P 帧的解码要与它在同一个 GOV(Group of Vop (video of plane)) 中的在它之前的 I 帧或 P 帧,每个 B 帧要用到与它相邻的那两个 I 帧或 P 帧,因此可得:

$$p(K | P) = \frac{1}{N_P} \sum_{k=1}^{N_P} [1 - (1 - \varepsilon)^{S_I + S_P}] \quad (5.5)$$

$$p(K | B) = \frac{1}{N_B} \sum_{k=1}^{N_B} [1 - (1 - \varepsilon)^{S_I(K+1) + S_B}] \quad (5.6)$$

其中 S_P 和 S_B 分别是在一个 P 帧或是 B 帧中平均包含的数据包数目; N_P 是一个 GOV 中平均包含的 P 帧的数目。通过 (5.4)、(5.5) 和 (5.6) 可以求出要求 $PSNR$

大于某一设定值时的丢弃率。对于多条标准 MPEG-4 视频流进行传输试验, 计算 *PSNR*, 可以得出, 通常 *PSNR* 大于 20dB 时, 丢包率小于 1/250 时, 视频质量才是可以接受的^[34], 并且 *P*、*B*、*I* 帧对视频质量的影响是不同的, 视频流中 *I* 帧比 *P*、*B* 帧重要, *P* 帧又比 *B* 帧对视频质量的影响大, 所以在对视频数据进行封装时, 可以对不同的视频帧设置不同的优先级, 这可以通过 RTP 报头扩展来实现。

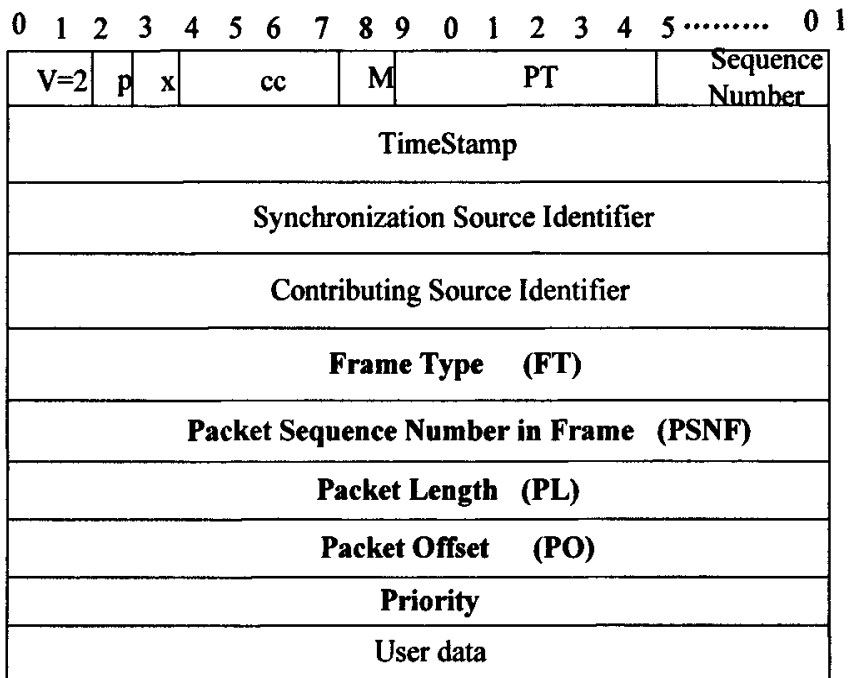


图 5.1 扩展 RTP 报头

图 5.1 是对 RTP 报头进行了扩展, 因一个视频帧的数据可能有多个包组成, 域 *Frame Type* 表示出帧的类型, *Packet Sequence Number in Frame (PSNF)* 表示该数据包在该帧序号, *PL* 表示该数据包的长度, *PO* 表示该数据包在该帧的具体位置, *Priority* 表示该包的优先级。在网络传输发生拥塞时, 对数据包按优先级别进行有选择的丢弃, 这将会减小回放视频质量的损失。

5.3 卡尔曼过滤器 (Kalman filter) 在网络状况分析中应用

流量预测是拥塞控制和网络管理的核心问题, 网络流量预测的精确性、实时性直接关系到业务管理的效率和性能, 现有的网络流量预测分为线性预测和非线性预测。对于线性预测, 它的理论依据在于网络流量具有线性特征, 但是网络流量具有多构性、自相似性、突发性, 且具有明显的非线性变化特征, 因此线性预测并不能保证预测的精确性^[45]。

卡尔曼滤波在数学上是一种统计估算方法, 通过处理一系列带有误差的实

际测量数据而得到物理参数的最佳估算。卡尔曼过滤器通过建立一系列数学方程,运用递归的方式对离散随机事件状态进行预测^[36, 37, 38]。

设通过 Kalman Filter 预测的状态为 $x \in \zeta^n$ 为一个由线性可微方程控制的随机事件,

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (5.7)$$

(5.7) 式中的系数 A 是一个 $n \times n$ 的矩阵, 描述了步长从 $k-1$ 改变为 k 时的参数状态; 参数 B 为一个 $n \times l$ 的矩阵, 它描述了当前状态参数为 x 时地最优控制参数 $u \in \zeta^l$ 。权值向量 $z \in \zeta^m$ 由下式确定:

$$z_k = Hx_k + v_k \quad (5.8)$$

其中 H 是一个 $m \times n$ 的状态矩阵, 描述了度量值的状态变化过程, H 的数值是随着步长的变化而变化的。(5.7) 和 (5.8) 中 w_k 和 v_k 分别为预测和测量噪声值, 两者是独立同分布的, 且符合正态分布。

$$p(w) \sim N(0, Q) \quad (5.9)$$

$$P(v) \sim N(0, R) \quad (5.10)$$

在实际运算中, 相关系数矩阵 Q 和 R 是随着步长进行变化的。通过以上步骤, 我们就可以建立预测方程, 然后对当前状态的下一个状态值进行递归计算, 设 $x_k \in \zeta^n$ 为当前工作状态为 k 的测量值, $x_k^{\Delta-} \in \zeta^n$ 为前一个状态为 $k-1$ 时的测量值, $x_k^{\Delta} \in \zeta^n$ 为后一个状态为 $k+1$ 时的预测值, 可以得出方程:

$$x_k^{\Delta} = x_k^{\Delta-} + K(z_k - Hx_k^{\Delta-}) \quad (5.11)$$

其中 K 为卡尔曼增益,

$$K = \frac{P_k^- H^T}{HP_k^- H^T + R} \quad (5.12)$$

其中 $P_k^- = E[e_k^- e_k^{-T}]$ 。为了校正误差, 定义出前向和后向误差校正方法。则前向校验为

$$e_k^- = x_k^{\Delta-} - x_k^{\Delta-} \quad (5.13)$$

后向校验为:

$$e_k = x_k^{\Delta} - x_k^{\Delta} \quad (5.14)$$

拥塞控制策略实施的关键在于网络状况的分析, 在实时流式传输系统中, 对网络流量的预测分析, 对传输服务质量有重大的影响。现有的许多控制机制都运用了不同的测量方法, 根据网络流量分析, 将网络状况分为^[39]: 轻载(unload)、重载(load)、拥塞(congested)。为了更好的测量非线性变化网络的流量, 可以将卡尔曼滤波器运用到网络状况分析来进行流量预测, 这样使传输系统的应用更具普遍性。

5.4 基于 RTCP 反馈的拥塞控制

通过第二章中对 RTP 协议分析, 我们知道 RTP 协议本身提供了很强的扩展能

力,可以利用视频接收端的 RTCP 反馈数据包来为发送端发送速率的变化提供调控的依据。带宽要求高的大数据量视频数据实时传输,接收端的反馈信息结合网络状况的分析构成了拥塞控制速率调整的基础。

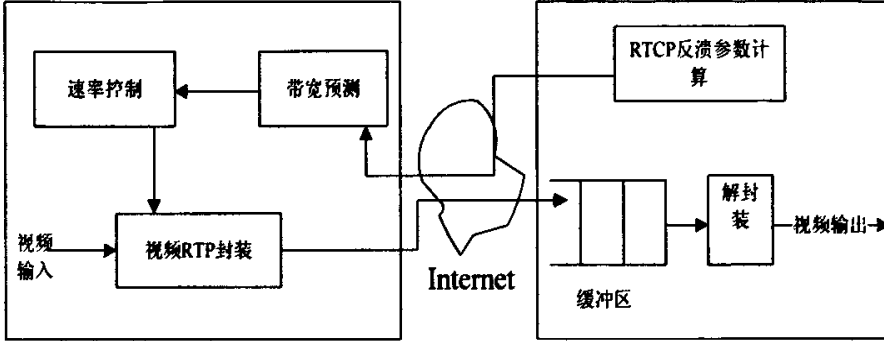


图 5.2 基于 RTCP 反馈的拥塞控制模型 RCC

如图 5.2 所示,我们提出的拥塞控制机制 RCC (Based on Real-time transport control protocol Congestion Control, RCC) 是基于 RTCP 反馈数据包基础上,基于 TCP 稳态反应函数确保 TCP 友好性。根据传输媒体特点,合理运用码率切换,采用设置接收端缓冲区的方式,并且动态调整缓冲区的大小。所有这些方面的组合构成了能够用于大数据量传输的拥塞控制机制,这其中涉及到许多具体的步骤,现分别进行说明。

5.4.1.1 回环时间 RTT 的测定

通过在 RTP 扩展报头设置特定的发出时间域,在 RTCP 的报头中也增加接收到数据包的时间域,在每次发出数据包时都进行时间域的更新,发送端通过接收端的反馈报告计算出回环时间 RTT。

$$t_{RTT} = t_{NOW} - t_{LSR} - t_{DLSR} \quad (5.15)$$

(5.15) 式中 t_{NOW} 为发送端收到 RR (Receiver Report) 的当前时间, t_{LSR} 为接收端接收到的来自源端的 SR (Sender Report) 包的时间, t_{DLSR} 为从接收端接到 SR 和发出 RR 之间的延迟。每个 RTCP 报告周期计算出来的 t_{RTT} 有较大的差异,做如下的平滑处理^[40],得到平滑后的回环时间值:

$$t_{RTT(i+1)} = \alpha_{RTT(i)} + (1 - \alpha) t_{RTT} \quad (5.16)$$

(5.16) 式中, $0 < \alpha < 1$, 取 $\alpha = 0.9$ 。

5.4.1.2 传输时限的设定

对于大数据量的多媒体实时传输,不考虑超时重传的问题,所以设定数据包的传输时限为

$$t_{RTO} = 4 t_{RTT} \quad (5.17)$$

5.4.1.3 包丢失率的计算

RTP 应用并不直接反馈链路的丢包事件发生率, 而只给出 RTCP 报告间隔内丢失包的数目以及丢失包所占的比率。而反馈的包丢失率是网络状况的及时反应, 发送端仅利用当前的包丢失比率来改变发送速率将会导致速率的急剧振荡, 高带宽的视频流数据将会出现瞬间大量数据包丢失的现象, 为了能得到较好的视频回放质量, 必须减小发送速率的振荡, 同时能够对下一个时间段的丢包率进行预测, 有效利用网络负载。通过统计历史包丢失率, 我们运用卡尔曼滤波算法通过迭代过程对 RTCP 间隔内的包丢失率进行预测。取传输过程任意时间段长为 M_{loss} , 设 RTCP 报告间隔 n 时刻时包丢失率为 \hat{p}_n , 这段时间内的包丢失率向量为 $p_n(n)$,

$$p_n(n) = [p_n, p_{n-1}, p_{n-2}, \dots, p_{n-M_{loss}+1}]^T, \quad (5.18)$$

取定权值向量 $W_p(n)$, 采用前向校正的误差校正策略, 则 $\alpha_n = p_n - \hat{p}_n$, 所以

$$\hat{p}_{n+1} = W_p^T(n) p_n(n) \quad (5.19)$$

$$W_p(n) = W_p(n-1) + K_p(n) \alpha_n \quad (5.20)$$

取 $\theta = 0.6$, 卡尔曼增益 $K_p(n)$ 为 $K_p(n) = \frac{\theta^{-1} \hat{p}_{n-1} \hat{p}_n}{1 + \theta^{-1} p_n(n)^T \hat{p}_{n-1} \hat{p}_n}$; $0 < \theta < 1$, 取 $\theta = 0.6$,

$$M_{loss} \approx \frac{1}{(1-\theta)}.$$

5.4.1.4 发送端速率调整

根据丢包率的反馈, 发送端探测到网络的状态和可用带宽。在接收端采用设置缓冲区的方式来进行回放视频的平滑处理, 接收端缓冲区的大小, 至少是一个 RTT 内传输的数据包的大小, 但接收端缓冲区设置的太大, 将严重增加视频的回放时延; 但接收端缓冲区设置的大小, 发送来的数据包将会被丢弃, 同样影响视频的回放质量; 同时当网络负载大时, 高带宽要求的视频流可以采用码率切换的方式来调整发送速率, 发送端有选择的丢弃一定的量的数据包, 但丢弃数据包的数量也是影响回放质量的重要因素, 所以说发送端速率的调整应该结合网络状况、接收缓冲区的大小、以及丢弃数据包的数量综合考虑。将这这三者有效结合起来, 通过预测的丢包率分析网络状态、结合网络状态确定丢包数量、缓冲区大小, 最后计算出发送端速率的调整值, 进行速率变换, 根据视频帧的特点码率切换时采用“保全 I 帧, 较少丢弃 P 帧, 丢弃 B 帧”的策略。

设接收端视频的平均回放速率为 μ , 接收端期望的发送速率为 δ_n , 发送端的实际发送速率为 λ_n , q_n 为第 n 个 RTCP 间隔 t_n 时接收到的数据包的数量, q_n^v 为第 n 个 RTCP 间隔 t_n 时刻接收端缓冲区的实际大小, q_n^v 为第 n 个 RTCP 间隔 t_n 时接收端缓

冲区的需求大小, $\bar{\zeta}_n$ 为拥塞发生时丢弃的数据包数目, 传输过程的包丢失率为 p_n 。根据公式 (5.1) 我们可以计算出 TCP-friendly 的发送速率为 R_{mean} , 为了达到传输的 TCP-friendly, 我们设 $\delta = R_{mean}$, 当传输过程中带宽充足, 那么 $\bar{\zeta}_n = 0$; 传输中发生拥塞时, 发送端丢弃的数据包的数目为, $\bar{\zeta}_n = (\lambda - \delta)x_n$; 所以 $q_n^v = q_n^v + \bar{\zeta}_{n-1}$, 即: 缓冲区大小应为当前的长度加上丢弃的数据包数目; 也可以计算出接收端在第 $n+1$ 个 RTCP 反馈时间间隔 t_{n+1} 时缓冲区应设置大小为 $q_{n+1}^v = q_n^v - \mu_n + \delta_n(1 - p_n)x_n + \bar{\zeta}_n$, 综合上面的分析, 我们可以计算出发送端的发送速率应为:

$$\lambda_n = \frac{3\mu_n - q_n^v - \bar{\zeta}_n}{(1 - p_n)x_n} - \delta_{n-1} \quad (5.21)$$

当 $\lambda_n \leq R_{mean}$ 时, 表示网络还有多余的链路带宽可以使用, 可以按这个速率发送; 当 $\lambda_n > R_{mean}$ 时, 发送端速率调整为 R_{mean} 。

此后, 每当发送端接收到来自接收端的反馈报告, 首先根据 (5.1) 式计算出 R_{mean} , 然后对下一个时间间隔应设置缓冲区的大小、丢弃数据包的数量进行分析, 最后由 (5.21) 式计算出发送端的发送速率, 进行速率的调整。

5.5 模拟实验

5.5.1 实验平台

为了评价协议和网络方案的正确性和性能, 需要做很多测试。传统的方法是为协议和网络方案的运行搭建实际的网络环境, 试验其在真实网络环境下的运行情况。但是搭建实际网络环境往往代价昂贵, 而且不易比较不同网络环境之间协议的运行结果, 真实网络中有很多随机网络事件也无法再现。因此, 搭建真实网络环境对于研究和评价协议设计与实现并不合适。相对于真实网络, 网络仿真是一种廉价、有效的协议测试办法。NS(Network Simulator)^[41]就是一种使用最广泛的网络仿真软件。NS 是美国 DARPA 支持的 VINT 项目的核心部分, 由 UC Berkeley, USC/ISI, LBL 和 Xerox PARC 等大学和实验室联合开发。其目的是构造虚拟的网络平台, 提供一系列仿真工具, 实现新的网络协议的设计和开发。它的前身是 S.Keshav 研制的 REAL 仿真器。这是一个可扩展的、容易配置的、可编程的事件驱动仿真引擎(simulation engine), 其源代码全部公开, 提供开放的用户接口。NS 以它对有线和无线(本地或卫星)网络、局域网和广域网、网络分层模型各协议的丰富支持、强大的二次开发能力以及可扩展、易配置和编程的事件驱动特性, 在国际网络研究界得到广泛的应用。我们也使用使用 NS 作网络模拟、评估、比较基

于 AIMD 的速率控制机制 RAP、基于 TCP 稳态反应函数的 TFRC 以及本文提出的 RCC。

5.5.2 NS 的组成

5.5.2.1 NS 软件构成

NS是一个软件包,包括Tcl/Tk、OTcl、NS仿真器。其中Tcl是一个开放脚本语言,用来对NS进行编程;Tk是Tcl的图形界面开发工具,可帮助用户在图形环境下开发图形界面;Otcl是基于Tcl/Tk的面向对象扩展,有自己的类层次结构;NS仿真器为本软件包的核心,是用C++编写、以Otcl解释器作为前端的面向对象的仿真器;Tclcl则提供NS和Otcl的接口,使对象和变量出现在两种语言中。为了直观地观察和分析仿真结果,NS提供了可选件Xgraph(显示静态的图形曲线)、可选件Nam(动态观察仿真的进行过程)。NS在Unix下开发,除了可用于Unix系统、Linux系统外,也可用于Windows系统,后者需要Cygwin进行安装^[42]。

5.5.2.2 NS 现有的仿真元素

目前,NS提供了大量仿真环境的元素,如仿真器、节点和分组转发、链路和延迟、队列管理与分组调度、代理、时钟、分组头及其格式、错误模型、局域网、地址结构(平面型和层次型)、移动网络、卫星网络、无线传播模型、能量模型等;提供了丰富的数学支持,如随机数产生、积分等;提供方便的追踪和监视方法;提供完整的路由支持如单播/组播路由、动态/静态路由、层次路由。下面从网络拓扑仿真、协议仿真和通信量仿真等方面介绍NS的相应元素。

1. 网络拓扑主要包括节点、链路。NS的节点由一系列的分类器(Classifier,如地址分类器等)组成,而链路由一系列的连接器(Connector)组成。
2. 在节点上,配置不同的代理可以实现相应的协议或其它模型仿真。如NS的TCP代理,发送代理有:TCP、TCP/Reno, TCP/Newreno, TCP/Negas, TCP/Sack, TCP/Fack, TCP/FullTcp等,接收代理有:TCPSink, TCPSink/DelAck, TCPSink/Sackl, TCPSink/Sackl/DelAck等。还提供有UDP代理及接收代理Null(负责通信量接收)、LossMonitor(通信量接收并维护一些接收数据的统计)。
3. 网络的路由配置通过对节点附加路由协议而实现。NS中有三种单播路由策略:静态、会话、动态。前两种使用Dijkstra算法,后一种使用分布式Bellman-Ford算法。在动态路由中,每一个节点可以执行一个以上的路由协议。NS还提供了组播路由策略:集中式、密集模型和共享树模型。
4. 在链路上可以配置带宽、时延和丢弃模型。NS支持:Drop-tail(FIFO)队列、

RED缓冲管理、CBQ(包括优先权和Round-robin调度)。各种公平队列包括:FQ、SFQ、DRR等。

5. 通信量仿真方面, NS提供了许多通信应用, 如FTP, 它产生较大的峰值数据传输; Telnet则根据相应文件随机选取传输数据的大小。此外, NS提供了四种类型的通信量产生器:

① EXPOO 根据指数分布(On/Off)产生通信量, 在On阶段分组以固定速率发送, Off阶段不发送分组, On/Off的分布符合指数分布, 分组尺寸固定。

② POO根据Paret分布产生通信量, 它能用来产生长范围相关的急剧通信量;

③ CBR 以确定的速率产生通信量, 分组尺寸固定, 可在分组间隔之间产生随机抖动; Traffic Trace根据追踪文件产生通信量。

5.5.3 NS 的功能

NS鼓励用户对其进行扩展, 它是一个开放、免费使用的软件。NS软件体系提供的基本抽象是“可编程组合”。仿真程序动态地把对象组合到任意的仿真配置中。

(1) NS使用的仿真语言

针对不同的仿真功能, NS提供了两种编程语言模型, 从而在不限仿真性能的前提下表现了充分的灵活性。对于细节的协议仿真, 涉及处理字节、分组以及大量数据集的算法等不需频繁修改的任务, NS采用编译型语言C++, 这样有利于提高仿真效率; 对于网络仿真中常常需要修改的参数配置、场景构造, 对于协议对象和规范的动态配置, 通信量的反复重定义任务, NS使用灵活、交互式的脚本TCL/OTCL, 这种方法的好处在于: 通过提供易用、重配置、可编程的仿真环境, 降低了仿真器设计、维护、扩展、调试的负担; 而且, 它鼓励将机制和策略分离的编程风格, 有利于代码重用。仿真器支持C++中的类层次(也叫编译层次)和OTCL解释器中对应的类层次(也叫解释层次), 这两个层次彼此紧密相关。从用户的角度看, 在编译层次和解释层次的类之间是一一对应的, 这些层次的基础是类TclObject。用户通过解释器建立新的仿真对象, 这些对象在解释器中实例化, 并且一个对应的对象紧密地镜像在编译层次。

(2) NS面向对象分析

NS使用了两种面向对象的语言来实现仿真目的, 这决定NS面向对象的性质。众所周知, C++是一种完善的面向对象编程语言; Otc;是 MIT Object Tcl的缩写, 以Tcl的语言和概念为基础进行了面向对象编程的扩展。因此无论进

行细节的协议仿真，还是进行仿真参数的配置、仿真场景的构造，都必须建立面向对象编程的思想。即实现仿真要对仿真元素进行认真的分析，抽象出相应元素的共性和派生关系；然后在仿真具体实现时进行实例化来生成仿真对象，在此基础上进行对象间关系的连接和处理。

5.5.4 NS 实现仿真过程

用NS来进行网络协议的仿真实验，可以分为以下的步骤：

1. 创立仿真器并配置或构造仿真网络拓扑(包括链路和节点)。仿真之前首先要构造一个基本的网络拓扑平台。此时，可以确定链路的基本特性，如延迟、带宽和丢失策略等。
2. 建立协议代理，包括端设备的协议绑定和通信量模型的建立。
3. 给节点进行特性化配置。根据仿真的具体要求对节点进行代理、路由协议等的初始化。
4. 编写必要的Tcl/Otcl过程或构造可能需要的Otcl类，如记录过程等。
5. 进行仿真结果的追踪。NS通过建立追踪文件来保存仿真期间网络性能的参数变化。仿真完成后，对追踪文件进行分析、处理得出所需性能指标或调用相应观察器(如Xgraph, Nam)对结果文件进行分析、研究。
6. 建立了上述代码后，设定通信量应用和时间相关过程的发送/结束时间，然后运行仿真。

5.6 实验及结果分析

5.6.1 模拟实验网络拓扑及参数配制

我们的实验是在 Windows xp 环境下，安装 Cygwin+ns2.2.29 来进行实验平台的搭建。网络实验拓扑如下图所示：

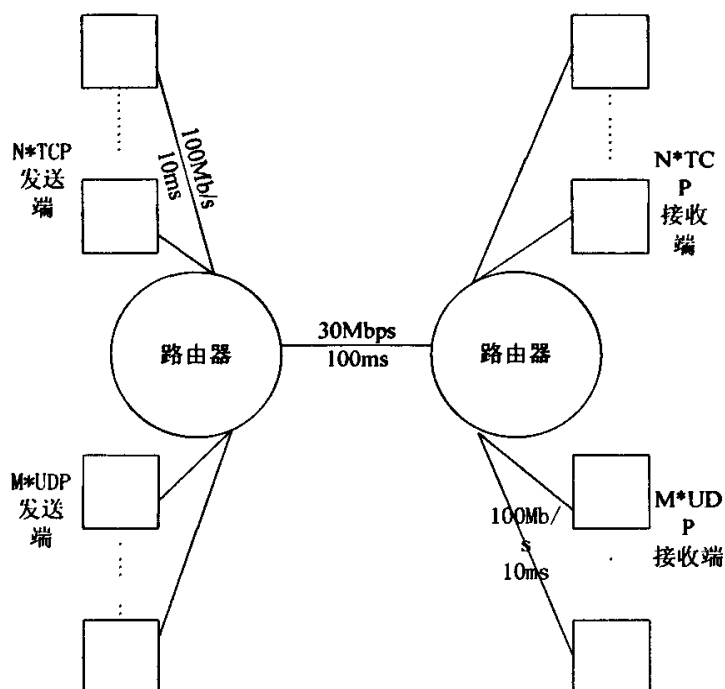


图 5.2 网络模拟拓扑

由于目前Internet中的路由器大都采用droptail的丢包方式，而且目前Internet中所用的TCP为RENO版本，因此在模拟试验中，所有链路都采用Droptail队列；TCP流都采用RENO版本。路由器间的瓶颈带宽为30Mb/s，时延为100ms，端点与路由器之间的瓶颈链路带宽100Mb/s,时延为10ms。

5.6.2 对 TCP 协议的友好性

RAP 和 TFRC 作为 NS 自身模块，我们将其加入到系统模块中，然后对 NS 进行重新编译。模拟实验参数如表 5.1 所示

表 5.1 模拟实验参数

模拟时间	500s	
路由队列管理策略	DropTail	
瓶颈以外的链路	延迟	10ms
	带宽	100Mb
瓶颈链路	延迟	100ms
	带宽	30Mb
	缓冲区大小	1000packets
RCC/TFRC/RAP/TCP 流的个数	50%	
RCC/TFRC/RAP/TCP 分组尺寸	10000 Byte	
反馈分组尺寸	50Byte	
R_{\min}	1000Kb/s	
T_{ON}	50s	
T_{OFF}	2s	

定义协议间的公平指数 F_p^{inter} 的作为两种数据流长时(long duration)公平性的度量。 F_p^{inter} 的定义为:

$$F_p^{inter} = \frac{R_{1/P}}{R_{1/P} + R_p} \quad (5.22)$$

其中, R_p 为待评估的数据流 P 的归一化平均吞吐量, $R_{1/P}$ 为其他与之竞争带宽的数据流的归一化平均吞吐量, 当 $F_p^{inter} = 0.5$ 时, 我们称它是一种理想的公平。通过调用 Xgraph 对实验结果进行分析。

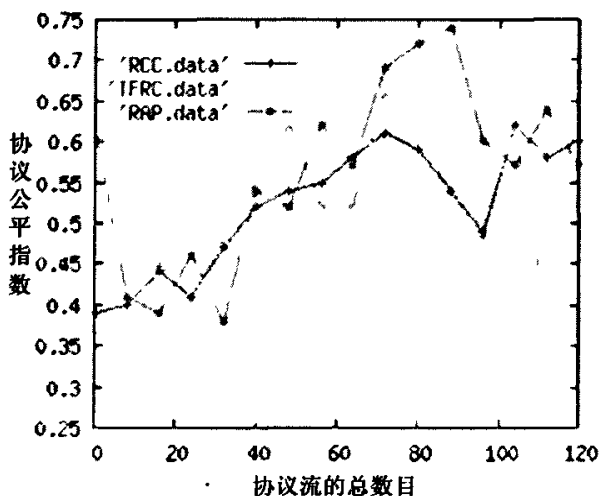


图 5.3RCC/TFRC/RAP 对 TCP 流公平性比较

结果如图 5.3 所示, 其中 x 轴表示 TCP-Friendly 流与 TCP 流的总数, y 轴表示协

议流的公平指数。可以得出，UDP 流采用 RCC 拥塞机制在传输过程中，协议数据流的吞吐量波动平滑，并且随着协议流总数的变化，比 TFRC 控制的协议流更趋近于理想的公平状态^[8]。

5.6.3 视频回放质量分析

我们通过模拟实际的视频流来进行分析，模拟参数的设置如下所示。

表 5.2 模拟实验参数

模拟时间	600s	
视频编码方案	DV(DCT)	
路由队列管理策略	DropTail	
瓶颈链路	带宽	30Mb
	延迟	20ms
	缓冲区大小	1000packets
瓶颈以外的链路	带宽	100Mb
	延迟	10ms
RCC/TFRC/TCP 流的个数	10	
RCC/TFRC/TCP 流的尺寸	2000Byte	
视频帧速率	30 帧/秒	
帧模式	IBBPBBPBBBPBB.....	

采用 TFRC 和 RCC 两种拥塞控制，得出接收端接收到的数据包的百分比，如图 5.3 所示，x 轴为网络的归一化负载率，定义为

$$W_n^{inter} = \frac{\Delta P_n}{P_{n-1} + P_n} \quad (5.23),$$

其中 Δ 为丢包率；y 轴为接收端所收到数据包的百分比，定义为

$$A_d^{inter} = \frac{q_n^v}{S_p} \quad (5.24)$$

其中， S_p 为发送数据包的大小， q_n^v 为接收缓冲区的实际大小。

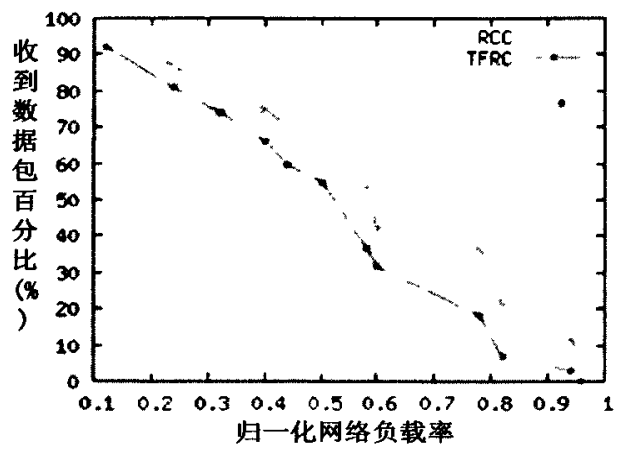


图 5.4 TFRC/RCC 丢包率比较

从图 5.4 可以得出，整个网络负载较小的时候，数据包可以较好地接收，视频回放质量自然高；随着网络负载率的变化，TFRC 机制首先对高带宽的数据流进行控制，丢包率将会增大，所以接收的数据包也将减少，而 RCC 控制机制有效进行速率调整，从而使接收的数据包的数量的减少比较平滑。

5.7 小结

本文提出的基于 RTCP 反馈的拥塞控制机制，充分考虑了高带宽视频帧的特点，结合网络状况分析与变长的缓冲区设置对发送速率进行平滑处理，弥补了基于数学模型的 TCP-Friendly 拥塞控制机制在大数据量视频流应用中的不足。通过网络模拟实验表明，该拥塞控制机制具有较好的 TCP 友好性，并能得到良好的视频回放质量，由于采用 RTCP 反馈机制，该控制可以灵活运用到单播或组播的环境。

总结和展望

1 论文的总结

随着网络技术的发展,基于 IP 的应用越来越广泛。高速交换技术的发展,网络带宽瓶颈正逐步得到解决,远程教育、远程医疗等基于 MPEG-1/MPEG-2 等高带宽要求的视频传输应用得到了发展。大数据量的视频流实时在网络上传输,对网络设备、网络带宽、以及整个网络的负载均衡都带来了极大的挑战。大数据量流媒体传输对网络安全影响的关键问题是发生拥塞时传输协议缺乏速率调节机制以及传输层协议之间带宽竞争的不公平性。针对这两个关键问题,本文进行了如下的研究工作:

- 1) 流式传输协议的研究。流式传输的发展依赖着传输协议,本文着重对实时传输协议 RTP/RTCP 进行了研究,仔细分析了 RTP/RTCP 的报文格式,对 RTP 协议特点有了较深的理解。
- 2) TCP 以及 TCP 友好的拥塞控制机制研究。TCP 协议是网络上端到端传输的通用网络技术。TCP 是具有拥塞退避机制的传输层协议,本文对 TCP 拥塞控制的原理、拥塞控制参数、拥塞控制的阶段进行了详细阐述。针对 TCP 拥塞控制机制的理解对制定拥塞控制策略是有帮助的。TCP 友好是为了解决传输层协议公平性而提出的协议规范,多媒体数据普遍采用 UDP 作为传输层协议,但只有符合 TCP 友好的吞吐量模式,才能最大限度的保障传输质量和网络安全,本文对当前提出的针对多媒体应用 TCP 友好速率调节机制进行了分析。
- 3) DVTS 传输系统的研究。作为 WIDE Project 课题的组成部分,本文对数字视频传输系统 DVTS 进行了深入研究,吸取了大数据量传输系统设计所必须的补偿和延迟、抖动消除策略。
- 4) 提出了大数据量流媒体传输的拥塞控制机制。结合以上所有的研究工作,本文提出了大数据量实时流媒体单播传输的拥塞控制机制,并通过 NS-2 对拥塞控制机制进行了实验分析。

2 未来的工作

课题的研究的过程,带给我的不仅是对网络协议的理解,更是一种思维方式

的改变。加深了协议的工作过程以及它在互联网技术和通信技术中的重要作用的理解,也感受到了研究协议的乐趣。虽然这是一件很枯燥乏味的事,但是当我深入到协议内部去研究时,我发现协议的世界是非常奇妙的。随着网络技术的发展,流式传输技术将深入的人们工作、学习的方方面面,流式传输在改变人们交流方式的同时对目前的网络提出了严峻的考验。由于个人能力有限,本文仅仅提出了一种拥塞控制的模型,只是通过模拟实验对其进行了实验分析,这些都是远远不够的。针对下一步的研究工作,主要可以从以下几个方面进行:

- 1) 由于本文提出的拥塞控制机制是基于端到端单播条件下的,但组播应用越来越广泛,目前针对实时流媒体组播也已经提出了相应的拥塞控制机制,但针对大数据量实时流媒体组播应用也需要有适合的拥塞控制机制。
- 2) 本文对 RTP 协议进行了扩展,对 DVTS 系统进行了深入研究,但没有将所提出的拥塞控制机制付诸于具体的实施,没有开发出改进的传输系统,从而缺乏在真实网络环境下的测试。所以开发一个具体的传输系统,在真实的网络环境中进行试验分析,对流式传输的应用更为重要。
- 3) 下一代互联网协议 IPv6 的提出,根本上克服了当前 IPv4 网络所具有的缺陷。IPv6 环境下实时流式传输的应用研究可以结合网络协议的特点制定相应的控制策略。

大数据量实时流媒体传输涉及到很多方面,本文仅仅对传输过程的拥塞控制机制进行了研究,由于时间和条件的关系,以及本人水平有限,在论文中难免有不足和缺陷,恳请各位老师批评指正。

参 考 文 献

- [1] D.Wu,Y.Thou,Y.Q.Zhang:Streaming over the Internet:approaches and directions .IEEE Trans on Circuit and Systems for Video Technology “.2001.11(1): pp 20
- [2] WIDE Project,Digital Video Transport System (DVTS)
<http://www.sft.wide.ad.jp/DVTS/>.
- [3] Dapeng Wu, Y Thomas Hon, Jason Yao, H Jonathan Chao, Real-Time Video over the Internet: A Big Picture. <http://www.cs.cmu.edu/~dpwu/publications.htm>
- [4] S.Deering,R.Hinden:Internet Protocol, Version 6 (IPv6)Specification, RFC2460 December 1998. <http://www.ietf.org/rfc/rfc2460.txt>.
- [5] Schulzrinne H ,Casner S,Fredrickson R et.RTP: A Transport Protocol for Real-time Applications .RFC 3550,2003.12.<http://www.ietf.org/rfc/rfc3550.txt>.
- [6] Perkins C RTP:Audio and Video for the Internet .MA :Addison.Wesley.2003
- [7] H. Schulzrinne,GMD Fokus:RTP Profile for Audio and Video Conferences with Minimal Control.<http://www.ietf.org/rfc/rfc1890.txt>.1996.1
- [8] Youn,J.Xin and M-T. Sun: Fast video transcoding architectures for networked multimedia application IEEE international symposium on Circuits and systems, pp25-28, Geneva, 2000.05
- [9] Brown Patrick .Resource sharing of TCP connections with different round trip times.Proc of IEEE Infocom 2000.
- [10] Andrew.S.Tanenbaum.著。熊桂喜,王小虎译。计算机网络(第3版)。北京:清华大学出版社。2002.7 pp 286-300
- [11] Reuven Cohen, Srinivas Ramanathan.TCP for High Performance in Hybrid Fiber Coaxial Broad-Band Access Networks.IEEE/ACM Transactions on Networking.2003.2
- [12] 林闯.多媒体信息网络QoS的控制.软件学报. 2001.10: pp1016-102
- [13] W. Richard Stevens著, 范建华, W光辉, 张涛等译. TCP/IP详解卷2: 协议. 北京: 机械工业出版社2000: pp 170-190
- [14] Paul Mockapetris,Domain System Changes and Observations. RFC973.
<http://www.ietf.org/rfc/rfc973.txt> 1986.1
- [15] Sandeep Bajaj, Breslau, L, Deborah Estrin, et al. Improve Simulation for Network Research .Technical Report 99-702,University of Southern California,1999.3
- [16] Douglas E.Comer,David L.Stevens 著。张娟,王海,黄树真译。用TCP/IP进

行网际互联-第二卷:设计、实现与内核.北京电子工业出版社.2004.11

[17] 张占军,韩承德,杨学良:多媒体实时传输协议 RTP.计算机工程与应用 2001.4: pp145-170

[18] 宋刚, 杨显富:实时流媒体传输及其协议.电子科技大学学报(自然科学版).2005.3

[19] Rao K R, Zoran S B, Dragorad A.多媒体通信系统技术、标准及网络(影印版) 北京:清华大学出版社,2003.

[20] 刘郁恒,陈广文,胡严,张光昭:一种在接收端实现的 TCP-Friendly 拥塞控制机制.电子学报.2005.5

[21] 易学能,唐卫清:面向 TCP 友好的音视频自适应实时传输.计算机工程与应用.2005.(6) pp 148-151

[22] 沈兰荪,卓力.小波变换与网络视频传输.北京:科学技术出版社.2005.pp359-380

[23] S. Floyd, J. Padhye, J. Widmer. TCP Friendly Rate Control (TFRC): Protocol Specification. 2003.1 <http://www.ietf.org/rfc/rfc3448.txt>

[24] K. Kobayashi, A. Ogawa, S. Casner, C. Bormann: RTP Payload Format for DV (IEC 61834) Video. 2000.1. <http://www.ietf.org/rfc/rfc3189.txt>

[25] K. Kobayashi, A. Ogawa, S. Casner, C. Bormann: RTP Payload Format for 12-bit DAT Audio and 20- and 24-bit Linear Sampled Audio. 2002.1 <http://www.ietf.org/rfc/rfc3190.txt>

[26] The WIDE Project. <http://www.wide.ad.jp>

[27] Sangwoo Han, Jong Won Kim: High-Quality Video Services for the Access Grid. ICAT2004.

[28] Toyokazu Akiyama, Shinji Shimojo, Shojiro Nishio, Yoshinori Kitatsuji, et al: Telecontrol of Ultra-High Voltage Electron Microscope Over Global IPv6 Network. In proceedings of SC97, UCSD Technical Report No. TR-CS97-540.

[29] Hiroshi Esaki, Akira Kato, Jun Murai: R&D Activities and Tested Operation in WIDE Project. Journal of Electron Microscopy 51(Supplement): S253-S257(2002)

[30] Akimichi Ogawa, Katsushi Kobayashi, Kazunori Sugiura: Design and Implementation of DV based Video over RTP. Packet video 2000..5

[31] YANG Y R, LAM S, S. General AIMD Congestion Control. In: Proceedings of the ICNP'00, 2000.

[32] M. HANDLEY, J. PADHYE. TCP Friendly Rate Control (TFRC) Protocol Specification. RFC3448. 2003.1 <http://www.ietf.org/rfc/rfc3448.txt>

- [33] SUX,WAH.BW: A Survey of Error-concealment Schemes for Real-time Audio and Video Transmission over the Internet.In Proc.Int'l Symposium on Multimedia Software Engineer .IEEE Taipei.TaiWan..2000,12 .pp 17-24;
- [34] 张拥军, 张怡, 彭宇行, 曾友春: 实时流媒体传输中的差错控制和恢复技术。国防科技大学学报。2003.3
- [35] REJAIE, HANDLEY, ESTRIN: An End-to-End Rate-Based Congestion Control Mechanism for Realtime Streams in the Internet. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. 1999, 4 1337-1345.
- [36] 李捷, 刘瑞新, 刘先省, 韩志杰: 一种基于混合模型的实时网络流量预测算法. 计算机研究与发展. 43 (5): pp 806~812 . 2006
- [37] A. Kolarov , A. Atai , J. Hui1 : Application of Kalman filter in High speed networks. GLOBECOM'94 , San Francisco , USA , 1994
- [38] Greg Welch, Gary Bishop : An introduction to the Kalman Filter. <http://www.cs.unc.edu/~welch>.
- [39] Ingo Busse, Bernd Deffner, Henning Schulzrinne: Dynamic QoS Control of Multimedia Application Based on RTP .IEEE. 0-768-02500499. 1999
- [40] Warren Toomey, S. Keshav, Alex Dupuy: The Network Simulator ns-2]. Dec. 2003. SA <http://www.isi.edu/nsnam/ns/index.html> Dec. 2003
- [41] 徐雷鸣, 庞博, 赵耀: NS 与网络模拟。北京: 人民邮电出版社。2003.2
- [42] Kevin Fall, Kannan Varadhan: The ns Manual (formerly ns Notes and Documentation) .A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC. 2006.10
- [43] 宋健, 王伟. 大数据量实时视频流单播传输拥塞控制机制研究。计算机应用。2006.11
- [44] 董振亚. MPEG-4 视频流式传输拥塞控制研究与实现。[国防科技大学硕士学位论文]。国防科技大学。2003.4 pp 32-39
- [45] 张敬轅. 互联网端到端拥塞控制研究。[上海交通大学博士学位论文]。上海交通大学。2002.6 pp 43-50

致 谢

本论文是在导师精心指导之下完成的。在此，我衷心的向他表示深深的谢意和崇高的敬意！

在整个研究生学习阶段，我的导师给予我详尽而细微的指导，是他的开阔的胸怀、渊博的学识、敏锐的洞察力、独到的见解和献身事业的忘我精神给我有力的鞭策和激励。从他身上我不仅学到了知识，更学到了科研为人的态度和方法。他事无巨细、严于要求，对我三年的学习和生活给予了无微不至的关怀和指导。在学习中，他要求我们严于律己。正是他的关心和帮助，使我三年的学习能够顺利进行，使本论文能够顺利完成。这一切都将令我今生难忘，受益终身！在此再次真心对他表示感谢。

感谢现代网络教育学院孙为老师对我的无私关心和帮助，正是他的敬业精神和谆谆教诲，使我受益非浅；正是由于他的大力支持和帮助，才使得我的课题研究和论文撰写得以顺利完成；感谢现代网络教育学院的各位老师，在实验环境和实验设备上给予的大力支持，我能够顺利完成论文是与他们所提供的良好的学习环境、细致和有效的帮助分不开的。

感谢 NS 与网络模拟讨论组，感谢互连网上亲密的朋友。

在此，向多年来始终支持、和鼓励我学习的家人表示由衷的谢意！是我父母的辛勤劳动和无私奉让我在学习中不感懈怠，努力进取，能顺利完成硕士阶段的学习。

最后，衷心感谢在百忙之中抽出时间来评阅论文和参加论文答辩的各位专家。

附录 A 攻读学位期间所发表的学术论文目录

- [1] 宋健, 王伟. 大数据量实时视频流单播传输拥塞控制机制研究. 计算机应用. 2006. 11
- [2] 孙为, 王伟 应用 Netfilter/iptables 构建穿越 NAT 的 IPv6 隧道网关系统. 计算机工程与设计. 2007. 8
- [3] 孙为, 王伟 大数据量 TCP 友好视频流传输拥塞控制研究. 兰州理工大学学报. 2007. 4
- [4] 孙为, 王伟 6to4 隧道穿越 NAT 方案研究与设计. 兰州理工大学学报. 2007. 6