

流媒体中关键技术研究

哈渭涛 著

非外借



科学出版社

流媒体中关键技术研究

哈渭涛 著



科学出版社

内 容 简 介

本书共 6 章,在已有的流媒体技术理论的基础上,较为系统地讨论流媒体传输过程中涉及的关键性问题。主要内容包括流媒体传输过程中的缓存技术、拥塞控制、代理服务器集群技术以及移动环境下流媒体传输等问题,在相关问题的阐述过程中不仅针对具体理论问题展开研究,而且还给出了在实际环境下的测试效果和仿真测试效果。本书的特点是概念准确、论述严谨、图文并茂,既重视基本原理的阐述,又力图反映流媒体技术的一些新发展。

本书可供数字媒体方向的研究生作为参考用书,也可供从事流媒体技术研究的工程技术人员学习参考。

图书在版编目(CIP)数据

流媒体中关键技术研究/哈渭涛著. —北京:科学出版社,2018.10

ISBN 978-7-03-058950-7

I. ①流… II. ①哈… III. ①多媒体技术—研究 IV. ①TP37

中国版本图书馆 CIP 数据核字 (2018) 第 221743 号

责任编辑:宋无汗 张瑞涛/责任校对:郭瑞芝

责任印制:张伟/封面设计:陈敬

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

北京教图印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2018 年 10 月第 一 版 开本:720×1000 1/16

2018 年 10 月第一次印刷 印张:11

字数:220 000

定价:80.00 元

(如有印装质量问题,我社负责调换)

前言

中国互联网络信息中心 (CNNIC) 于 2017 年 8 月发布了第 40 次《中国互联网络发展状况统计报告》(以下简称《报告》)。《报告》显示,截至 2017 年 6 月,我国网民规模达到 7.51 亿,半年新增网民共计 1992 万,半年增长率为 2.7%;互联网普及率为 54.3%,较 2016 年底提升 1.1%;我国手机网民规模达 7.24 亿,较 2016 年底增加 2830 万;网民中使用手机上网的比例由 2016 年底的 95.1% 提升至 96.3%,手机上网比例持续提升。

《报告》同时指出,“网络娱乐类应用用户规模稳步增长,行业不断向正规化发展”,网络视频行业、各大视频网站均布局包括文学、漫画、影视、游戏及其衍生产品的泛娱乐内容新生态,生态化平台的整体协同能力正在逐步凸显,运营正规化和内容精品化是当前发展的主要方向。在网络的各项应用中,网络视频所占比例达到了 75.2%,位居各项应用的第三位。中国网络视频用户规模达 5.65 亿。

随着互联网技术的不断发展和广泛应用,网络中大量的数据以流媒体数据的形式表现出来。流媒体的实时性以及对带宽资源的需求,使得网络资源越来越不能满足要求,为了缓解网络压力和满足用户需求,许多新的技术被提了出来,其中包括流媒体代理缓存技术、流媒体传输拥塞控制技术、流媒体服务器集群调度策略以及移动环境下的流媒体传输技术等。

本书在已有的流媒体技术理论的基础上,主要介绍了代理缓存技术、前缀缓存的管理,以及对于补丁算法的改进及其实现;阐述了网络拥塞和各种拥塞控制方法,并深入研究和探讨了流媒体的各种拥塞控制机制,并在 TCP 友好速度控制机制 (TFRC) 算法的基础上提出了改进的拥塞控制算法;介绍了集群负载均衡技术的发展以及流媒体的相关协议,针对传统动态反馈算法存在的问题作出改进;对移动流媒体特性进行分析和研究,针对移动流媒体传输中的问题给出不同算法,并对给出的算法进行理论分析和实验验证。

本书得到了渭南师范学院学术专著出版基金的资助和科学出版社的大力支持,在此表示感谢。

由于作者水平有限,书中难免存在不足之处,尚请读者批评指正。

目 录

前言

第 1 章 绪论	1
1.1 背景	1
1.2 研究领域和研究现状	1
1.2.1 流媒体技术研究	1
1.2.2 流媒体应用形式研究	3
1.3 本书结构与主要内容	3
参考文献	5
第 2 章 流媒体基础	6
2.1 流媒体的基本原理	6
2.1.1 流媒体简介	6
2.1.2 流式传输的形式	6
2.1.3 流媒体服务器	8
2.2 流媒体技术的相关协议	8
2.2.1 实时传输协议	8
2.2.2 实时传输控制协议	10
2.2.3 实时流协议	12
2.2.4 资源预留协议	16
2.2.5 微软媒体服务器协议	18
2.3 流媒体内容的传播形式	19
2.3.1 单播	19
2.3.2 广播	20
2.3.3 组播	20
2.3.4 点播	21
2.3.5 智能流技术	21
2.4 流媒体技术的发展现状	22
2.5 本章小结	28
参考文献	29
第 3 章 流媒体代理缓存技术	30
3.1 网络缓存机制	30

3.1.1 客户端缓存	30
3.1.2 服务器端缓存	30
3.1.3 代理缓存	31
3.2 流媒体代理缓存	31
3.2.1 流媒体代理缓存的必要性	31
3.2.2 流媒体代理缓存的原理	32
3.3 流媒体代理缓存的设计目标和性能评价	32
3.3.1 流媒体代理缓存的设计目标	32
3.3.2 流媒体代理缓存的性能评价	33
3.4 常见的流媒体代理缓存调度策略	34
3.5 一种流媒体代理缓存系统的实现	35
3.5.1 系统构成	35
3.5.2 所在学院现有校园网的情况	36
3.5.3 硬件设备的选择	38
3.5.4 软件系统的选择	40
3.5.5 前缀缓存	42
3.5.6 前缀缓存的大小	42
3.5.7 前缀缓存的管理	43
3.5.8 前缀缓存的替换	49
3.5.9 补丁算法	52
3.5.10 代理的其他实现	57
3.5.11 系统测试	63
3.6 本章小结	64
参考文献	65
第 4 章 流媒体传输拥塞控制的研究	66
4.1 网络拥塞概述	66
4.1.1 拥塞的定义	66
4.1.2 拥塞产生的原因	66
4.1.3 拥塞控制算法的评价指标	67
4.1.4 因特网传输服务质量	68
4.2 拥塞控制方法	69
4.2.1 慢开始和拥塞避免	69
4.2.2 快重传和快恢复	70
4.3 TCP 友好拥塞控制算法	71
4.3.1 基于探测的拥塞控制	72

4.3.2 基于模型的拥塞控制	73
4.4 基于 TCP 友好速率控制策略 TFRC 算法的改进	78
4.4.1 平均丢失事件间隔计算方式的改进	78
4.4.2 分组数据包权重计算方式的改进	82
4.4.3 仿真实验	85
4.4.4 实验结果分析与总结	87
4.5 本章小结	88
参考文献	89
第 5 章 流媒体服务器集群负载均衡调度策略	90
5.1 集群负载均衡	91
5.1.1 集群负载均衡概述	91
5.1.2 常用的集群负载均衡软件	92
5.2 常用负载均衡算法	93
5.2.1 静态负载均衡算法	93
5.2.2 动态负载均衡算法	94
5.3 动态反馈负载均衡调度策略的优化	97
5.3.1 动态反馈负载均衡调度策略的优化思路	97
5.3.2 负载性能指标及负载权值	100
5.3.3 动态修改负载反馈周期	105
5.3.4 集群节点分类	106
5.3.5 过载节点的负载迁移	108
5.3.6 优化调度策略的整体流程	109
5.4 负载均衡调度策略的实现	111
5.4.1 负载均衡器模块的实现	111
5.4.2 流媒体服务器模块的实现	116
5.4.3 实验平台搭建	119
5.4.4 功能验证实验	119
5.4.5 性能对比实验	124
5.5 本章小结	127
参考文献	127
第 6 章 移动环境下流媒体数据传输	129
6.1 无线网络流媒体数据传输技术的研究现状	130
6.2 保证截止时间的流媒体传输背压算法	137
6.2.1 问题描述	137
6.2.2 模型定义	138

6.2.3 保证截止时间的流控制决策	138
6.2.4 保证截止时间的流媒体传输背压算法	139
6.2.5 仿真与实验结果分析	142
6.3 基于簇的流媒体传输背压算法	144
6.3.1 网络拓扑节点分簇及模型定义	145
6.3.2 基于簇的保证截止时间的背压算法	146
6.3.3 仿真与实验结果分析	147
6.4 基于最短路径的流媒体传输背压算法	149
6.4.1 问题描述	149
6.4.2 模型定义	150
6.4.3 跳约束的背压算法	151
6.4.4 吞吐量最优及跳数最优的路由调度	156
6.4.5 仿真与实验结果分析	159
6.5 本章小结	161
参考文献	162
结束语	164

第1章 绪 论

1.1 背 景

通信技术的飞速发展,改变了传统信息处理、加工和传播的手段,加快了人类文明进步的步伐。多媒体技术则综合了声音、文字、图像、动画和视频等多种交互手段,拓宽了信息的表现形式,为人类的数字生活带来了深刻变革。网络通信技术和多媒体技术相结合,产生了流媒体 (streaming media) 的概念^[1-3]。

流媒体是以流式传输技术通过网络传送的、在时间上具有连续性的媒体文件。与传统的多媒体相比,流媒体具有如下特点^[4,5]: ①流媒体的内容是时间上连续的媒体数据 (如视频、音频、动画等); ②流媒体内容可以不经转换便能通过网络流式传输; ③具有较强的实时性要求以及较好的用户交互性支持; ④支持边下载边观看的用户播放模式,缩短了用户的启动等待时间; ⑤在客户端接收、处理和回放流媒体文件的过程中,文件不在客户端长时间驻留,播放完随即被清除,不占用客户端的存储空间; ⑥由于流媒体文件不在客户端保存,因而在一定程度上解决了媒体文件的版权保护问题。

近年来,随着宽带网络的普及和 4G 研究的深入,流媒体技术获得了广泛关注;对因特网流量的统计表明,流媒体业务流量正成为因特网上流量的主体;权威机构调查发现,流媒体业务已是 4G 网络中的杀手级业务。流媒体的应用系统、国际标准和基础研究正成为目前产业和科研密切关注的热点。

典型的流媒体应用包括视频会议、远程教育、视频监控、协同工作、IPTV、交互式多媒体游戏、数字化多媒体图书馆等。

1.2 研究领域和研究现状

流媒体系统的组成包括四个功能环节:内容制作、发布、传输和播放^[6]。流媒体的研究现状大致分为以下几个部分。

1.2.1 流媒体技术研究

1. 视频压缩及编码

传统的不可扩展性视频编码的目标是将视频压缩成适合一个或者几个固定码率的码流,是面向存储的,因此不适合网络传输。为了适应网络带宽的变化,面向

传输的可扩展性编码的思想应运而生。可扩展性编码就是将多媒体数据压缩编码成多个流,其中一个可以独立解码,产生质量粗糙的视频序列,它适应最低的网络带宽,称为基本层码流;其他的码流可以以层为单位在任何地点截断,称为增强层,用来覆盖网络带宽变化的动态范围,它们不可以单独解码,而只能与基本层和它以前的增强层联合在一起解码,用来提高观看效果。因此,可扩展性码流具有一定的网络带宽适应能力。

可扩展性编码主要分为时域可扩展性编码、空域可扩展性编码和质量可扩展性编码,可以选择在时间、空间和信噪比 (SNR) 中的一个或几个方面实现扩展。考虑到编码效率和复杂性两个方面, MPEG(动态图像专家组) 采纳了精细可扩展性编码 (FGS) 和渐进的精细可扩展性编码 (PFGS), 满足拥有不同网络带宽和不同分辨率接收设备的许多用户的需求,性能得到了更大的提高。结合多种视频编码技术来适应网络上的 QoS (quality of service, 服务质量) 波动是今后可扩展性视频编码的发展方向。例如,可扩展性视频编码可以适应网络带宽的变化;错误弹性编码可以适应丢包; DCVC (delay cognizant video coding, 延迟识别视频编码) 可以适应网络时延。这三种技术的结合可以更好地提供一种应对网络 QoS 波动的解决方案。

2. 应用层 QoS 控制技术

由于目前的因特网只提供 Best-effort 的服务,所以需要通过应用层的机制来实现 QoS 的控制。QoS 控制技术主要集中在对网络带宽的变化进行响应和处理分组丢失的技术上,主要可以分为两类:拥塞控制技术和差错控制技术。

拥塞控制只能减少数据包的丢失,但是网络中不可避免地会存在数据包丢失,而且到达时延过大的分组也会被认为没有用而丢弃,从而降低了视频质量。要改善视频质量,就需要一定的差错控制机制。差错控制机制包括以下几种。

(1) 前向纠错 (FEC)。FEC 是通过在传输的码流中加入用于纠错的冗余信息,在遇到包丢失的情况时,利用冗余信息恢复丢失的信息。它的不足之处是增加了编码时延和传输带宽。

(2) 延迟约束的重传。通常流的播放有时间限制,因此,仅在重传的时间小于正常的播放时间时,重传才是有价值的。

(3) 错误弹性编码 (error-resilient encoding)。在编码中通过适当的控制使得发生数据丢失后能够最大限度地减少对质量的影响。在因特网环境下,最典型的方法是多描述编码 (MDC)。MDC 把原始的视频序列压缩成多位流,每个流对应一种描述,都可以提供可接受的视觉质量。多个描述结合起来可以提供更好的质量。该方法的优点是实现了对数据丢失的鲁棒性和增强的质量,其缺点是相比单描述编码 (SDC),它在压缩的效率上受到影响。而且由于在多描述之间必须加入一定的相关性信息,这进一步降低了压缩的效率。

(4) 错误的取消 (concealment)。错误的取消是指当错误已经发生后, 接收端通过一定的方法尽量削弱对人的视觉影响。主要的方法是时间和空间的插值 (interpolation)。近年来的研究还包括最大平滑恢复、运动补偿时间预测等。

3. 流服务器

视频服务器在流媒体服务中起着非常重要的作用。当视频服务器响应客户的视频流请求以后, 它从存储系统读入一部分视频数据到对应于这个视频流的特定缓存中, 再把缓存的内容通过网络接口发送给相应客户, 保证视频流的连续输出。目前存在三种类型的视频服务器结构。

(1) 通用主机方法。采用计算机主机作为视频服务器。它的主要功能是存储、选择和传送数据, 缺点是系统成本高而且不利于发挥主机功能。

(2) 紧耦合多处理机。把一些可以大量完成某指令或者专门功能的硬件单元组合成的专用系统级联起来, 就构成了由紧耦合多处理机实现的视频服务器。这种服务器费用低、性能高、功能强, 但是扩展性较差。

(3) 调谐视频服务器。这种服务器主板上有一个独特微码的嵌入式仿真器控制。通过在主板中插入更多的服务通路, 可以更方便地进行扩展。

对于流服务器, 如何更有效地支持 VCR 交互控制功能, 如何设计磁盘阵列上多媒体对象高效可靠的存储和检索, 如何设计更好的可伸缩多媒体服务器, 如何设计兼有奇偶和镜像特性的容错存储系统, 这些都是目前研究的重点。

1.2.2 流媒体应用形式研究

传输模式主要是指流媒体传输是点到点的方式还是点到多点的方式。点到点的模式一般用单播 (unicast) 传输来实现, 点到多点的模式一般采用组播 (multicast) 传输来实现, 在网络不支持组播的时候, 也可以用多个单播传输来实现。实时性是指视频内容源是否实时产生、采集和播放, 实时内容主要包括实况 (live) 内容、视频会议节目内容等, 而非实时内容指预先制作并存储好的媒体内容。交互性是指应用是否需要交互, 即流媒体的传输是单向的还是双向的。

视频点播 (VOD) 是目前最常见、最流行的流媒体应用类型。通常视频点播是对存储的非实时性内容以单播传输方式实现, 除了控制信息外, 视频点播通常不具有交互性。在具体实现上, 视频点播可能具有更复杂的功能。例如, 为了节约带宽, 可以将多个相邻的点播要求合并成一个并以组播方式传输。

1.3 本书结构与主要内容

本书内容基于高校的科研项目, 主要任务是搭建一个流式文件的传输环境, 并

且研究流媒体技术中的相关技术。全书分为 6 章。

第 1 章简述了流媒体技术的研究背景和研究现状,并介绍了本书的章节结构。

第 2 章较为系统地介绍了流媒体技术基础知识,包括流媒体的基本原理、发展现状和传输协议。对流式传输技术主要涉及的多种实时传输协议,如 RTSP、RTCP、RTP 等均作了较为细致的研究。

第 3 章主要介绍了代理缓存技术,首先对比网络缓存技术,提出流媒体缓存的概念,并提出了流媒代理缓存的设计目标和性能评价指标;着重阐述了流媒体服务器和流媒体代理服务服务器的搭建过程、前缀缓存的管理,以及对于补丁算法的改进及其实现,并完成了本书流式传输系统。

第 4 章主要阐述了网络拥塞和各种拥塞控制方法,并深入研究和探讨了流媒体的各种拥塞控制机制,并在 TCP 友好速度控制机制 TFRC 算法的基础上提出了改进的拥塞控制算法,利用数学证明验证了其正确性与完备性,并通过仿真软件 NS2 完成了相关的模拟实验和数据采集。通过对实验成果的分析与总结,改良后的 TFRC 算法动态调整和平衡发送端的传输速率,不轻易受网络拥塞的影响而波动,保持平稳的发送速度,对数据流的传输表现出良好的亲和性,从而保证了流媒体实时传输的 QoS。

第 5 章首先介绍了集群负载均衡技术的发展以及流媒体的相关协议,其次深入分析与研究了现有的静态和动态负载均衡算法,明确各种调度算法的适用场景,然后针对传统动态反馈算法存在的问题作出如下改进:①根据集群节点每秒钟任务连接数的变化量动态地修改负载反馈周期,提升集群节点负载反馈的及时性;②将集群节点按其负载状况分为低负载、正常负载和高负载三类,类之间根据总的负载权值进行任务分配,类中采用最小连接数算法分配任务,解决处理大量并发任务请求时存在的负载倾斜问题;③采用流媒体服务器的中继/转发功能实现过载节点的负载迁移,进一步提升集群的负载均衡效果。最后,本书对优化的负载均衡调度策略进行编码实现,并搭建流媒体服务器集群,对优化的调度策略进行验证,并同传统的动态反馈负载均衡算法和最小连接数算法进行比较。实验结果表明,优化的调度策略能够更加及时地反馈集群节点的负载状况,有效解决大量并发请求导致的集群负载倾斜问题,提升集群的负载均衡效果和集群的服务质量。

第 6 章首先对移动流媒体特性进行分析和研究,由于用户的移动会导致以下问题:已经建立的连接中断;数据传输结构效果变差或者失效;无线连接的变化更加频繁;针对通信环境中的监控更难实施。针对这些问题给出保证截止时间的流媒体传输背压算法,并对给出的算法进行理论分析和实验验证;其次,由于背压算法分布式传输的特性,网络的收敛速度可能会比较慢,这在一定程度上会导致网络达到吞吐量最优所需要的时间变长,本章提出了基于簇的背压算法,实验证明,算法能够保证网络吞吐量在达到最优的同时,加快网络的收敛速度,提高用户对流媒体

数据传输的满意度;最后,为了在充分利用网络资源的同时限制流媒体传输对其他传输的影响,减少端对端的传输延迟,本章研究了网络通信能力在达到吞吐量最优的同时最小化数据包传输的平均跳数的问题,并通过理论和实验证明算法在能够保证网络吞吐量达到最优的同时,降低端对端的传输延迟。

参 考 文 献

- [1] 钟玉琢, 向哲, 沈洪. 流媒体和视频服务器 [M]. 北京: 清华大学出版社, 2003.
- [2] 张丽. 流媒体技术大全 [M]. 北京: 中国青年出版社, 2001.
- [3] Wu D, Hou Y T, Zhu W, et al. Streaming video over the Internet: approaches and directions[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2001, 11 (3): 282-300.
- [4] 李向阳, 卞德森. 流媒体及其应用技术 [J]. 现代电视技术, 2002, (4): 18-27.
- [5] 李睿, 曾德贤. 流媒体关键技术与面临的问题 [J]. 现代电视技术, 2005, (5): 92-95.
- [6] 李秋云, 郝建国, 陈鹏. 流媒体业务及技术发展 [J]. 数据通信, 2004, (2): 27-29.

第2章 流媒体基础

2.1 流媒体的基本原理

2.1.1 流媒体简介

流媒体是指在 Internet/Intranet 中使用流式传输技术的连续时间媒体,如音频、视频或多媒体文件。这个词首先出现在美国,英文是“Streaming Media”,中文就直接翻译成“流媒体”^[1-3]。

流媒体把连续的影像和声音信息经过特殊的压缩方式分成一个个压缩包,通过视频/音频服务器向用户计算机连续、实时地传送,用户可以一边下载一边观看、收听,而不需要等整个压缩文件下载到自己的机器后才可以观看。该技术先在用户端的电脑上创建一个缓冲区,于播放前预先下载文件的一小段数据作为缓冲,播放程序时取用这一小段缓冲区内的数据进行播放。在播放的同时,多媒体文件的剩余部分在后台继续下载填充到缓冲区。这样,当网络实际连线速度小于播放所耗用数据的速度时,可以避免播放的中断,也使得播放品质得以维持,所以流媒体最显著的特征是“边下载、边播放”^[4,5]。

2.1.2 流式传输的形式

目前,实现流式传输主要有两种方法:实时流传输和顺序流传输。一般来说,如视频为实时广播,或使用流媒体服务器,或应用如 RTSP 的实时协议,即为实时流传输。如使用 HTTP(hypertext transfer protocol, 超文本传输协议)服务器,文件即通过顺序流发送,即为顺序流传输。当然流文件也支持播放前完全下载到硬盘^[6,7]。

1. 顺序流传输

顺序流传输 (progressive streaming) 其实就是顺序下载,在下载文件的同时,客户端可以欣赏在线流媒体内容。可是,服务器上的传送信息与用户欣赏到的内容并不是协调一致的,客户端机器看到服务器上传过来的内容,需要经过一段延迟。换句话说,客户端看到的都是服务器在一段时间以前传输过来的内容。

在传送过程中,在特定时间,客户端不能跳到前面还未下载的部分,但可以观看自己已经下载的内容。与实时流式传送相比,不能按照用户连接的速率对传送的视频流进行调节,只能被动接受和调整已经下载完成的部分。

对于顺序流式文件,普通的 HTTP 服务器都可以传送,不用其他特殊协议,所以,顺序流式传送也被叫作 HTTP 流式传送。高画质的短片段适宜采用顺序流式传输,尤其是那些质量较高、数据量很小、采用 Modem 推出的短片段,如片头、片尾和广告,因为已经无损下载播放前观赏的那部分文件,因此可以确保流媒体影像播放的最终 QoS。但是,对于这种方式,用户在观看前必须经历一段时间的延迟加载,对于那些较慢的接入方式和低速率的带宽,需要等待很长的时间。

顺序流式传输对于那些采用调制解调器推出的短片段很有效,它创建视频片段,可以容忍数据的速率比调制解调器还高。尽管有延迟,但是可以推出较高画质的视频片段,所以适宜于在网站上推出的客户端点播的音视频节目。

顺序流式文件一般都是存放在标准 HTTP 或 FTP 服务器上,和防火墙无关,方便维护。严格来说,顺序流式传送是一种点播技术,不提供现场广播,同样不适宜有按需随意访问要求的或者长段视频,如讲座、演示与演说。

2. 实时流传输

实时流传输 (realtime streaming) 在确保媒体信号带宽可以满足它连接的网络的前提下,可实时观看流式传输节目。实时流式传送和 HTTP 流式传送不同,必须采用专门的流媒体服务器和传送协议,才能准确无误地将流媒体作品内容传送到用户客户端并实时连续展示出来。

实时流式传送尤其适用于现场事件,因为它总能够保证实时连续传送,还可以随意访问,即客户端可以撤退、快进观赏后面或前面的信息,这是顺序流式传送所不具备的。在理想状态下,实时媒体数据流只要开始播放,就不会暂停,可现实生活中,仍然会发生规律性的停歇现象。

实时流式传送需要满足通信链路的带宽,这会导致图像画质很差,尤其是以调制解调器速率连接的时候。当互联网出现拥塞或延迟时,出错的信息和丢失的信息很容易被忽视,导致客户端显示的影像质量太差,这个时候,采用 HTTP 流式传送效果可能会更好。

要想通过实时流式传送媒体信息,必须具备特定的流媒体服务器,最终才能达成目标,如微软的 Windows Media Server、Real Networks 公司的 Real Server、苹果公司的 QuickTime Streaming Server。这些流媒体传输服务器作用相当强大,提供了多种级别的流式传送控制,因此系统的管理与配置比标准的 HTTP 流媒体服务器复杂太多。

此外,实时流式和 HTTP 流式传送不同的是,必须采用对应的流媒体传输机制,如 MMS(Microsoft media server, 微软媒体服务器) 协议、RTCP(realtime transport control protocol, 实时传输控制协议)、RTP(realtime transport protocol, 实时传输协议)、RSVP(resource reservation protocol, 资源预留协议)、RTSP(realtime streaming

protocol, 实时流协议)。这些协议面对防火墙时也许会被拦截住, 造成客户端看不到实时内容。

显然, 在现实应用中, 按照需求来确定到底采取哪种传送方式, 而且实时流式传送也支持在全部下载到硬盘然后再放映。实时流式传输模式能和流媒体服务器建立联系, 正是因为采用了 RTP/UDP、RTSP/TCP 两种通信机制, 将流媒体服务器的传输定位到目的地址, 也就是执行流媒体播放器应用所在用户机器的 IP 地址。理论上, 在网络带宽通畅无阻的情况下, 比顺序流式传输功能强大许多, 传输效率更高。通常来说, 流式传输系统必须配备一套专门的流媒体服务器和流媒体播放器, 而目前市面上用户电脑上的播放器几乎都支持流媒体播放。

2.1.3 流媒体服务器

服务器软件模型主要有两种, 即循环服务器和并发服务器。循环服务器 (iterative server) 是指在一个时刻只处理一个请求的服务器, 并发服务器 (concurrent server) 是指在一个时刻可以处理多个请求的服务器。事实上, 多数服务器没有用于同时处理多个请求的冗余设备, 而是提供一种表面上的并发性, 方法是依靠执行多个线程, 每个线程处理一个请求, 从客户的角度看, 服务器就像在并发地与多个客户通信^[8-10]。

由于流媒体服务时间的不定性和数据交互实时性的请求, 流媒体服务器一般采用并发服务器算法。

流媒体服务器的主要功能如下:

(1) 响应客户的请求, 把媒体数据传送给客户。流媒体服务器在流媒体传送期间必须与客户的播放器保持双向通信 (这种通信是必需的, 因为客户可能随时暂停或快放一个文件)。

(2) 响应广播的同时能够及时处理新接收的实时广播数据, 并将其编码。

(3) 可提供其他额外功能, 如数字权限管理 (DRM), 插播广告, 分割或镜像其他服务器的流, 以及组播。

2.2 流媒体技术的相关协议

2.2.1 实时传输协议

实时传输协议 (RTP) 是用于因特网上针对多媒体数据流的一种传输协议。RTP 被定义为在一对一或一对多的传输情况下工作, 其目的是提供时间信息和实现流同步。RTP 通常使用 UDP 来传送数据, 但 RTP 也可以在 TCP 或 ATM 等其他协议上工作。当 RTP 工作于一对多的传输情况下时, 依靠底层网络实现组播, 利用 RTP over UDP 模式实现组播的传输就是其典型应用^[11]。

1. RTP 协议工作原理

不可预估数据包到达客户端的时间，同时还要提供数据流的实时播放和回放，是流媒体数据流传输中必须要解决的重要问题。发送端在报文分组中插入一个隐蔽的实时时间标志（时戳），伴随时间的前进而不断增多，时戳把接收到的多媒体信息流按照正确的顺序提交到应用层。接收端拿到数据包后，按照时戳的准确速度恢复成原来的实时数据流。

RTP 协议本身担保数据包的可靠传送，也不负责网络阻塞和网络流量控制，真正负责流媒体传送 QoS 的是 RTCP。RTP 和 RTCP 协议协同工作，完成传输层的协议功能，RTP 只负责数据流的封装，通过 UDP 数据包承载，RTP 协议则控制其传送，提供可靠性保证。

RTP 传输协议有如下一些特点^[12]。

1) 协议灵活性

RTP 协议不具备运输层协议的完整功能，其本身也不提供任何机制来保证实时地传输数据，不支持资源预留，也不保证服务质量。RTP 报文甚至不包括长度和报文边界的描述，而是依靠下层协议提供长度标识和长度限制。另外，RTP 协议将部分运输层协议功能（如流量控制）上移到应用层完成，简化了运输层处理，提高了该层效率。

2) 数据流和控制流分离

RTP 协议的数据报文和控制报文使用相邻的不同端口，这样大大提高了协议的灵活性和处理的简单性。

3) 协议的可扩展性和适用性

RTP 协议通常为一个具体的应用提供服务，通过一个具体的应用进程实现，而不作为 OSI 体系结构中单独的一层来实现，RTP 只提供协议框架，开发者可以根据应用的具体要求对协议进行充分扩展。

一个标准的 RTP 报文是由固定头（fixed header）和用户数据（或称数据负载，payload）两部分组成。RTP 协议的固定头格式如图 2.1 所示。

0			7		8		15		16		31	
V	P	X	CSRC 计数			M	载荷类型			序号		
时间戳												
同步源 (SSRC) 标识符												
作用源 (CSRC) 标识符												

图 2.1 RTP 协议的数据包头

RTP 协议的固定头格式中各个参数的含义分别如下。

(1) V(version): 2 比特。标识 RTP 版本，现在为 2。