

基于自适应流媒体传输的直播系统建设

陈自强¹
CHEN Ziqiang

摘要

为了解决互联网上流媒体直播所面临的网络容量不足和用户访问变化的问题,首先通过讨论控制 IP 网络中流媒体传输的控制技术,实时引入了自适应媒体传播策略,以适应网络带宽动态变化,通过认证信件和 RTP / RTCP,成功实现了流媒体的实时传输,采用这种策略有效地解决了网络容量不足和用户访问变化所带来的挑战,实现了流媒体的成功实时传输。然后,对自适应传输算法进行了详细的分析和设计,并利用多线程技术实现了网络状况监测和实时流切换控制功能。最后,通过对研究和实验数据的比较分析,验证了所提出解决方案在提高并发流传输效率和提供优秀用户体验方面的有效性。

关键词

自适应; 实时传输协议; 流媒体; 实时传输控制协议

doi: 10.3969/j.issn.1672-9528.2024.02.032

0 引言

随着网络技术的迅猛发展和三网融合进程的加速,互联网协议电视(IPTV)业务得到了快速普及。在 IPTV 技术中,流媒体技术是一个关键核心部分,随着时间的推移,它也变得越来越成熟,但多媒体流具有大量的数据和对实时性的高要求,这导致在 IP 网络中传输时很难保证服务质量(QoS)。然而,由于 IP 网络的不断变化和网络异构性等特征,流媒体传输过程中可能会遇到网络拥堵、传输延迟和数据波动等挑战。

本文对于推动 IPTV 业务的发展以及提升流媒体传输质量具有重要意义。随着流媒体技术的不断演进和网络环境的改善,人们可以更好地享受到高质量的流媒体直播服务。未来,可以进一步探索和改进流媒体传输控制技术,以满足不断增长的用户需求,并推动流媒体领域的创新发展。

1 流媒体传输协议

随着互联网上实时数据传输需求的增加,IETF(互联网工程任务组)制定了一套标准协议,即实时传输控制协议/实时传输协议(RTCP/RTP),用于在互联网上传输实时数据。RTP/RTCP 协议在支持多媒体流传输环境方面提供了广泛的灵活性,为多媒体流提供了多样化的控制空间和方法。该协议在用户数据报协议(UDP)上运行,由 RTP 数据部分和 RTCP 控制部分组成。

RTP 提供实时数据传输,可用于通信或网络传输。该协议实现了端到端的运输服务,并确保能够通过传送必要信息

和序列信息来恢复实时信号。它可以通过信息的时间戳实现同步流动。包头结构如图 1 所示。

版本号 2 bit	补齐位 1 bit	扩展位 1 bit	CSRC 个数 1 bit	标记 1 bit	负载类型 7 bit	顺序号 16 bit
时间戳 32 bit						
SSRC 标识符 32 bit						
CSRC 标识符 32 bit						
CSRC 标识符 1 bit						
用户数据						

图 1 RTP 包头结构

RTCP 协议是 RTP 协议的控制部分,主要功能是允许接收端点向发送端点发送接收者报告,提供关于接收到的媒体数据质量、网络延迟等信息。RTCP 提供了五种类型的控制包。

- (1) 发送方报告包(SR): 发送方发送的统计信息。
- (2) 接收方报告包(RR): 接收方接收的统计信息。
- (3) 源描述项包(SDES): 描述与会话源相关的信息。
- (4) 应用包(APP): 用于开发新应用。
- (5) 离开包(BYE): 在离开会话时发送的报告包。

RTCP 控制协议必须与 RTP 协议结合使用。当应用程序启动 RTP 时,同时使用两个端口:一个用于 RTP,另一个用于 RTCP。为了区分 RTP 包和 RTCP 包,通常使用不同的端口号码。RTCP 包使用与 RTP 相同的传播机制,并定期向会议所有参与者发送控制信息。通过接收这些信息,应用程序可以获得会议参与者、网络状态、包丢失速度、提供 QoS(服务质量)或诊断网络健康等相关信息。

这些 RTP/RTCP 协议的定义和应用为实时数据传输提供了可靠的基础,为多媒体流的传输和控制提供了有效的解决

1. 新华网(北京)科技有限公司 北京 100000

方案。通过合理使用这些协议,可以更好地管理流媒体数据的传输,并提供良好的用户体验。随着技术的不断发展,可以进一步优化和改进这些协议,以满足日益增长的实时数据传输需求。

2 自适应流媒体直播系统设计

2.1 结构形式

自适应流媒体直播系统通常可分为视频源捕获、编码压缩、传输传控和终端接收四个主要部分,如图2所示。



图2 流媒体直播系统

收集视频来源是实时流媒体系统的关键步骤之一,包括捕获和用视频卡记录视频的来源。视频的来源可能是标准或高纯度的来源,而常见的来源包括DVD播放器、有线电视游戏机和摄像机。可以通过PCI、AGP或PCI-E或通过外部USB接口连接到视频源,重要的是确保视频源的稳定捕获和高质量录制,以提供优质的流媒体直播体验。

编码压缩部分实现可分为纯软件和嵌入式两种方式。在纯软件方式中,利用操作系统提供的第三方编码软件(例如Windows media encoder)进行编码压缩,其处理过程完全依赖计算机的CPU。在嵌入式方案中,编码部分将直接固化在单片机或嵌入式芯片中,可以与视频采集卡配合或独立使用,实现编码压缩功能。选择适合需求的实现方式对于确保高效地编码压缩和流媒体直播体验至关重要。

终端输出部分主要涉及视频解码和视频播放。

控制设备的移动在媒体流通系统、稳定、拥挤和延误网络方面发挥着重要作用,即使在目前的网络条件下,媒体流通和战略控制的转移仍然是一个问题。通过这样的策略,可以确保流媒体传输的稳定性和流畅性,从而为用户提供更优质的体验。

通过对视频源的采集、编码压缩、终端输出以及传输控制等关键环节的优化和整合,流媒体实时直播系统能够提供高质量的视频传输和播放服务。未来,随着技术的不断进步,可以进一步改进流媒体系统的各个组成部分,以满足用户对实时直播不断增长的需求。

2.2 系统设计

在设计直播系统时,视频来源于购买蓝光播放器部分,提供高清晰度或可持续发展视频来源,以及使用高清晰度c727元视频卡。未来,MPEG-2(传输流)是一种通过标准MPEG-2在开放源平台上捕获的视频编码和压缩生成的形式,RTP/rthock被用于实时传输控制。图3显示了该系统的总体结构。

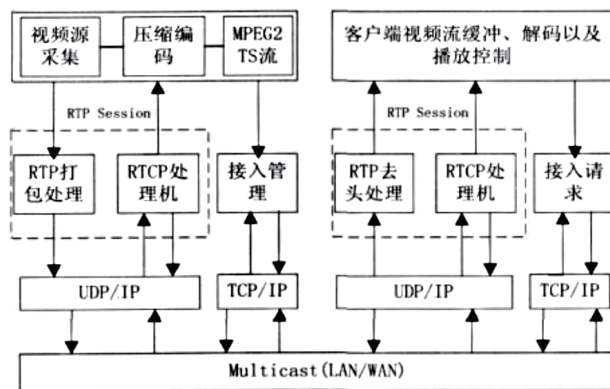


图3 系统结构图

课程安排首先在服务器上创建,需要现场直播。会议期间安排启动服务器,建立两个基于UDP的插头,一个连接广播电视台进行数据传输,另一个获得RTCP数据。服务器可以读取临时存储器里的捕获卡捕捉到的媒体流。视频流量是一个MPEG-2型压缩,按最大传输器的大小分解。将头部排名,然后添加到每个破碎的包件,并将其归类为包件。为了确保数据传输的准确性和完整性,有必要在制造头级时将标识符、有效载荷类型和时间等单位储存。

在自适应媒体流2上,可以选择动态组中合适的装载类型。当标志位设置为1时,它表明当前的RTP是最后一个框架。每次RTP包发送时,最初数值是随机数字,并增加1。时间戳反映了从24h取样服务器获得的包类第一个字节取样时间,用以前创建的RTP发送数据到目的地地址。当目标是通信地址时,节目广播实现。

客户端还需要创建RTP会话基于UDP的插头。通过加入通信组,客户可以从相应的软件获得数据流量。RTP在现有数据分级中所载的束,为实现节目编排,采用了电解标记来区分收件包、RTP流分析、解密等。通过以上步骤,服务器和客户端之间建立了RTP会话,并通过UDP进行数据传输,实现了直播系统中的节目直播和观看功能。这种流程保证了节目的实时性和传输的稳定性,为用户提供了良好的观看体验。未来,可以进一步优化传输过程,提升直播系统的性能和可靠性。

3 自适应传输的算法设计

3.1 网络状况测量与调节算法

为了实时监测当前网络状态,本文提出了一个网络状况测量模块,用于计算可用的带宽资源和网络的运行状态。

根据RTCP协议的要求,每个客户端都将RTCP包发送到定期的媒体服务器流量中。自上次报告以来,RR(收到者报告)有8个领域是SR(发件人报告),记录包损失率。按客户端RTCP包中RR字段计算的定期媒体流量估计网络状况损失率。

假设在本文所设计的流媒体系统中,服务器同时生成了 n 个(例如 $n=4$)并发的视频流,每个视频流有 m 个(例如

$m=3$) 不同码率的选项。最初, 二级服务器会接收这 n 个视频流中各自 m 个不同码率视频流中最高质量 (即码率最高) 的节目流, 并将其转发到网络中。

系统可以根据用户的需求和网络状况进行智能调整, 以提供最佳的播放质量。设 n 路节目的质量优先级分别为 f_1 、 f_2 、 $\dots$ 、 f_n , 设 $f_1 > f_2 > f_3 > \dots > f_n$ 。假设第一路节目流的 m 种视频流码率分别为 R_{11} 、 R_{12} 、 $\dots$ 、 R_{1m} , 那么, 第 n 路节目的 m 种视频流码率分别为 R_{n1} 、 R_{n2} 、 $\dots$ 、 R_{nm} 。用以下矩阵来表示一级服务器产生的所有视频流码率:

$$\text{码率矩阵 } \mathbf{M} = \begin{bmatrix} R_{11} & \dots & R_{1m} \\ \vdots & \ddots & \vdots \\ R_{n1} & \dots & R_{nm} \end{bmatrix} \quad (1)$$

设 $\mathbf{N} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, 假设初始时每路节目有 m 个码率选项, 按照

从高到低的顺序排列。那么二级服务器初始时产生的 m 路节目流可以表示为 $\mathbf{Q} = \mathbf{M} \times \mathbf{N}$, 即 $\begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix} = \begin{bmatrix} R_{11} \\ R_{21} \\ \vdots \\ R_{n1} \end{bmatrix}$, 在统计丢失率时,

测量模块可以计算其平均值, 并将其表示为 f , 代表测得的丢包率。

通过码率矩阵 $\mathbf{M}$, 系统可以根据用户需求和网络状况选择合适的视频流码率组合, 以提供最佳的播放质量和流畅性。通过矩阵 $\mathbf{N}$, 系统可以根据网络状况和用户需求, 在不同的码率选项中进行选择, 以提供最佳的播放质量和流畅性。

通过灵活地选择合适的码率组合, 流媒体系统可以在不同的网络环境下提供高质量的视频传输服务。这种设计可以满足用户对不同清晰度的节目需求, 并兼顾网络的实际状况, 提供良好的用户体验。在未来的发展中, 可以进一步优化该矩阵的设计和码率调整算法, 以适应不断变化的网络条件和用户需求。

假设在某个时间片内, 服务器获取了 a 个 RTCP 包, 并且这些包中 RR 字段反馈的丢失率分别为 h_1 、 h_2 、 $\dots$ 、 h_a 。为了描述该时间片的网络状况, 用丢失率 p 来表示它们的均值。

根据获取的 a 个 RTCP 包的丢失率, 可以计算平均丢失率 p , 公式如下:

$$p = \frac{h_1 + h_2 + \dots + h_a}{a} \quad (2)$$

则有 $p_t = (1-a) \times p(t-1) + a \times p$, 式中 p_t 表示到第 t 个时间片上的平均丢包率, a 通常取 0.25 或者 0.125。

网络状况监测模块通过实时监测丢包率等指标来评估网络的稳定性和质量。一旦获得了网络丢包率的信息, 这些数据会被传输到服务器端。服务器端根据这些反馈信息, 使用特定的算法来判断当前网络的状况, 并相应地调整流媒体传输的带宽。

通过网络带宽的自适应分发控制, 服务器端可以根据实际网络情况动态地调整传输的带宽。当网络状况良好时, 可以提高传输的带宽, 以获得更高的视频质量和流畅度。而当网络状况不稳定或带宽有限时, 服务器端会相应地降低传输的带宽, 以避免出现视频的卡顿和缓冲等问题。

通过这种网络带宽的自适应分发控制机制, 流媒体系统可以更好地适应网络环境的变化, 并提供更好的用户体验。未来, 可以进一步改进算法和控制策略, 更精确地根据网络状况进行带宽调整, 进一步提高用户体验的质量。

丢失率阈值为 e (如 $\frac{1}{16}$), 丢失率增量为 δ , 则调节方法如下。

(1) $p_t = [e-\delta, e+\delta]$, 则 Q 保持不变。

(2) $p_t < e-\delta$, $b = (e-p)/\delta$ 。

①若 $b < t$, 则 $R_s = R_{s(t-b)}$ 。

②若 $b \geq t$, 则 $R_s = R_{s1}$ 。

(3) $p_t > e+\delta$, $b = (p-e)/\delta$ 。

①若 $b \leq m-t$, 则 $R_s = R_{s(t+b)}$ 。

②若 $b > m-t$, 令 $i = s-c$, $c = [(b-(m-t))/m] + 1$,

式中: i 为新的质量等级区分流的序号, 则 $R_i = R_{iq}$, $q = b-(m-t)-(m-1)c+1$, while $(i++ \leq s) \{R_i = R_{im}\}$ 。

为了保持节目播放的同步, 同一程序流程的不同版本被调成同步的二级数据。通过在相应的通信组之间切换码流组播组, 码流切换可以很容易完成。

根据不同的网络状况状态, 二级服务器中的码流切换模块可以相应地选择合适的码率版本进行传输。在轻载状态下, 可以选择高质量的码率版本以提供最佳的播放体验。在正常状态下, 可以继续保持当前的码率版本, 因为传输质量良好, 不需要切换。而在满载状态下, 为了保证播放的连续性, 可以选择降低码率版本, 以减少包丢失率, 并提供相对较好的播放质量。

在这一制度中, 有两个阈值分别设定的阈值和上阈值, 其中 $0 < \text{最低阈值} < 1$ 。最低限度是设定为最小值, 而上限相对较高, 但尚未达到拥挤程度。与此同时, e 值是在下端和下端之间的选择, 这表明在正常程序中正常质量的损失率。 e 值被确定当前下网络是否接近装载光或完整装载光的临界点。

为了设定这个临界点, 可以引入一个松弛因子 δ , 表示丢包率变化的容忍范围。该松弛因子可以根据 e 、 λ_c 的值和并发流的数量来设定。可以计算 $\delta = (\lambda_c - e) / (n \times m)$, 式中 n 表示并发流的数量, m 表示每路节目流的码率选项数目。

通过设定阈值和临界点, 系统可以根据实际网络状况判断当前状态是轻载、正常还是满载。当丢包率低于 λ_u 时, 表示网络负载较轻, 可以选择高质量的码率版本进行传输。当丢包率处于 λ_u 和 λ_c 之间, 表示网络状态正常, 可以继续保持

当前的码率版本。而当丢包率超过 λ_c 时,表示网络负载较重,可以选择降低码率版本以减少丢包率。

通过实时监测网络状况,包括丢包率、带宽利用率等指标,系统可以根据当前的网络情况进行相应调整。当网络状况出现异常,如丢包率过高或带宽利用率过低,算法会根据设定的阈值和临界点来判断当前状态,并采取相应的措施来使网络状况趋近于正常状态。

为了保证多路并发流中的节目播放质量良好,算法会根据网络状况的变化进行码率切换。当网络负载较轻时,可以选择高质量的码率版本进行传输,以提供更好的播放体验。而当网络负载较重时,可以降低码率版本,以减少丢包率,并保持较好的播放质量。

3.2 多线程模型设计

网络状况测量线程负责实时监测网络的状况,包括丢包率、带宽利用率等指标。它周期性地从网络中接收并分析相应的数据包,并根据设定的算法计算出当前的网络状况。这些测量结果会被传递给自适应调节线程进行进一步处理。

自适应调节线程根据网络状况测量线程提供的数据,判断当前的网络状态,并根据设定的阈值和临界点来进行相应的码流切换控制。它负责根据实时的网络状况来调整传输的码率版本,以保证在不同网络环境下的最佳播放体验。

主线程是整个系统的控制中心,它协调和管理网络状况测量线程和自适应调节线程的工作。主线程负责启动和终止各个线程,并实时监控系统的运行状态。它还与用户交互,接收用户的输入和指令,并根据用户的需求进行相应的处理。

通过多线程的设计和协作,整个系统能够并行地进行网络状况测量和自适应调节,以实现网络状况的实时监测和动态调整。这样可以提高系统的响应速度和效率,并提供更好的用户体验。图4展示了各个线程之间的关系和交互过程。

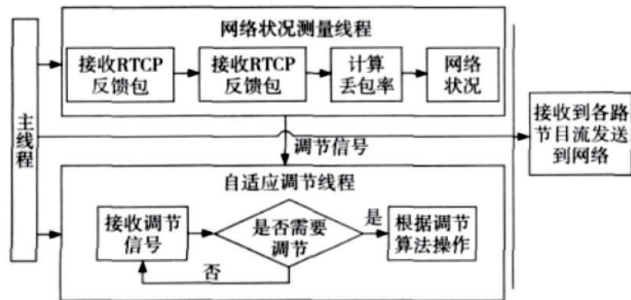


图4 多线程模型

经过对比实验,如图5所示,可以看到在未采用任何传输控制策略时,丢包率呈现一种近似线性增加的趋势。当网络状况变差,丢包率达到阈值 e (例如1/16)后,丢包率会不断上升,导致客户端的播放经常出现停顿的情况。这说明在缺乏传输控制策略的情况下,网络负载较重时,无法有效地保证播放的连续性和流畅性。一旦引入了上述自适应传输

控制算法,情况就完全不同了。通过算法的调节和优化,丢包率在阈值上下波动,而不是呈现线性增加的趋势。这意味着算法能够更好地适应网络状况的变化,提高带宽的使用率,并提供更好的客户端播放质量。

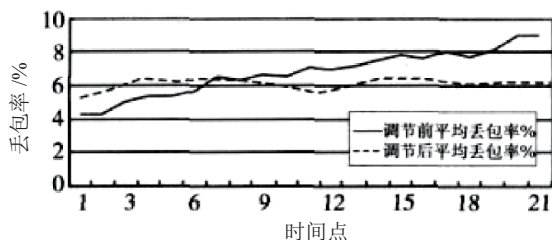


图5 使用自适应调节算法前后对比

4 总结

根据局域网测试结果,在正常网络环境下,本文提出的算法能够满足用户对画面质量和实时性的需求。然而,当网络业务增加,网络状况恶化时,本文中的自适应算法将智能调整视频流编码速率,以适应网络目前的带宽。这种转换算法既简单又有效,可以大大提高客户的用户体验。特别是在服务器上播放的程序数量增加时,这种模式的影响更加明显,不会对服务器造成太大压力。本文提出的自适应算法结合了IPv6环境下的多流复合传输,进一步提高了应用效果。未来将继续探索这两个方面,进行更深入的研究和优化。与此同时,在实时切换的过程中,目前还存在画面停机的问

参考文献:

- [1] 沈以楼,朱艳琴.流媒体直播系统的多路并发流自适应传输控制[J].计算机工程与科学,2012,34(2):35-40.
- [2] 王如恒.流媒体直播传输码率自适应的方法研究[D].上海:上海师范大学,2020.
- [3] 李清.自适应流媒体传输协议研究[J].中国新通信,2023,25(4):19-21.
- [4] 刘韶星.HTTP自适应流媒体点播云平台缓存管理研究[D].桂林:桂林电子科技大学,2022.
- [5] 钱洋.基于IPv6的高伸缩性SVC视频传输技术研究[D].南京:东南大学,2021.
- [6] 王瀚.基于动态自适应流媒体传输技术的全景视频传输系统研究与实现[D].重庆:重庆邮电大学,2021.
- [7] 张莒璇.基于流媒体自适应策略的全景内容播放平台的研究与设计[D].北京:北京交通大学,2020.

【作者简介】

陈自强(1988—),男,宁夏固原人,硕士,中级工程师,研究方向:多场景下网络流媒体直播系统建设及技术优化。

(收稿日期:2023-06-15)