

监控系统中实时视频流码率动态调整机制的研究

巴洪涛 陈耀武 周 蓉

(浙江大学数字技术及仪器研究所 杭州 310027)

摘 要 为了达到监控系统视频的实时性和流畅性要求,针对现有拥塞控制算法对实时视频流控制的不足,提出了一种新的基于 RTCP 反馈的拥塞控制算法。该算法通过客户端反馈的 RTCP 包得到关键参数,估计回环时间和丢包率,预测网络带宽,进而调整服务器端视频流的编码码率。实验证明,该算法能根据网络状况动态调整实时视频流的编码码率,使客户端观察到实时流畅的画面。

关键词 监控系统,RTCP,拥塞控制,回环时间,丢包率

Studies on Dynamic Adjustment Mechanism of Real-time Video Streaming on Surveillance System

BA Hong-tao CHEN Yao-wu ZHOU Rong

(Institute of Advanced Digital Technologies and Instrumentation, Zhejiang University, Hangzhou 310027, China)

Abstract In order to design a real-time and fluent video surveillance system, this paper presented a new RTCP-based feedback congestion control algorithms to make up for the deficiencies of current way using the congestion control algorithms. In the algorithm,utilizing the key parameters in the feedback TCP packet received from the client,we can estimate the circle time and packet loss rate,forecast the network bandwidth to adjust the encoding bit rate of server-side video stream. Experiments show that the algorithm can adjust the video stream encoding bit rate dynamically according to network conditions so that the client can get the smooth real-time video stream.

Keywords Surveillance system,RTCP,Congestion control,RTT,Fraction lost

1 引言

视频监控系统发展到今天,已经进入了网络视频监控的阶段^[1],网络视频监控系统是一种以计算机技术、图像视频压缩技术和视频数据实时传输技术为核心的新型数字视频监控系统,在教育、军事、银行、邮电、交通、电力、医疗等各行业有着广泛的应用前景。本网络视频监控系统可充分利用计算机网络,网络带宽可复用,信号不易被干扰,图像品质及稳定性高,且便于集中控制;其次采用数字化存储,把监控录像保存在大容量硬盘上,数字信号长期存储信息永不丢失,图像质量不下降,同时附属设备少,造价低,操作维护方便。

网络视频监控系统要保证监控端所获得视频的实时性和可靠性,必须解决网络中出现的拥塞情况。现有的拥塞控制算法可以分为两类。

第一类是基于 AIMD(Additive Increase Multiplicative Decrease)的控制算法。该算法的主要思想是设置一个丢包率阈值,当 RTCP(Real-time Transport Control Protocol)数据包反馈的丢包率低于阈值时,使用加性增长传输来逐渐适应网络带宽;当丢包率超过阈值时,使用乘性减少码流量传输。但该算法容易引起 RTP 流的抖动,导致丢包和时延的剧烈变化,采用 AIMD 算法所带来的反馈控制的超调量比较大,直接的影响就是网络中数据传输率的波动过大,不适合 Internet 上发展迅速的影音等流媒体文件的传输^[2]。

第二类是基于 TCP 吞吐量模型的控制算法,以 TFRC(TCP Friendly Rate Control)为代表。TFRC 是基于速率的拥塞控制机制,不同于 TCP 通过拥塞窗口进行控制,是根据以公式计算而得的速率对吞吐量进行控制^[3]。其原理是通过一个以丢包率等为参数的公式来计算发送方的发送速率上限,发送方根据此公式的计算结果对自身的发送速率进行调整。TFRC 使用的是在 TCP 协议中定义的吞吐率计算公式,如式(1)所示^[4-6]:

$$R_{sp} = \frac{S}{RTT \sqrt{\frac{2bp}{3}} + t_{RTO} (3\sqrt{\frac{3bp}{8}}) p(1+32p^2)} \quad (1)$$

其中, R_{sp} 为吞吐率, S 为数据包的大小, RTT 为回环时间, P 为丢包率,值在 $[0,1]$ 内, t_{RTO} 为重传时钟,一般设置为 4 RTT , b 为 TCP 接收端一次确认的包数,一般设置为 1。该算法在大部分网络情况下具有稳定的传输性能,但开始发送数据时,接收端因没有收到发送端的报文而无法计算出丢包率,这时发送端便无法根据公式计算出应采用的发送速率。这类似 TCP 的慢启动阶段,使得实时流媒体接收端在开始有较大的延迟和抖动。

2 动态调整机制

2.1 系统相关模块设计

本监控系统传输音视频采用了基于 RTSP(Real-time Streaming Protocol)^[7]协议的流媒体服务器和相应的客户端。

到稿日期:2009-06-26 返修日期:2009-08-05

巴洪涛(1985—),男,硕士生,E-mail:huodian@gmail.com;陈耀武(1963—),男,教授,博士生导师;周 蓉(1985—),女,硕士生。

建立了连接以后,服务器开始向客户端发送 RTP(Real-time Transport Protocol)^[8] 视频数据,同时,服务器和客户端每隔一定时间都分别向对方发送 RTCP^[8] 反馈包,其中,服务器向客户端发送 SR 包(发送者报告),客户端向服务器发送 RR 包(接收者报告)。服务器端通过客户端反馈的 RR 包获取相应信息,分析网络状况,并根据网络状况进行视频编码码率的动态调整。系统具体相关的模块如图 1 所示。

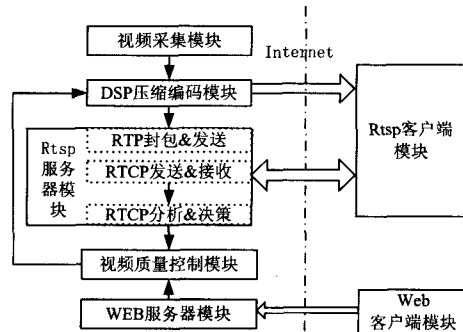


图 1 视频流码率动态调整相关模块

引入动态调整机制之前,本系统视频流质量调整过程如下:

用户通过远程 Web 客户端即浏览器访问本地 Web 服务器,设置实时视频流的编码参数,包括帧率和码率。

本地 Web 服务器模块将相应的信息发送给视频质量控制模块。

视频质量控制模块立即发送相应的命令给 DSP 编码模块,DSP 进行视频流编码的帧码率调整。

为了使客户端看到流畅的实时视频,引入的动态调整机制仅仅调整监控系统视频流的码率,而不调整帧率。过程如下:

RTSP 服务器发送 RTP 数据包到 RTSP 客户端,并自动记录与 RTP 数据包发送有关的信息。

RTSP 客户端自动记录与 RTP 数据包接收有关的信息并发送给服务器。

RTSP 服务器根据已记录的信息和接收到的 RTCP 分析计算得到当前网络带宽,根据带宽预测发送下一帧时的网络带宽,并根据预测调整实时视频流的编码码率,将相应信息传送给视频质量控制模块。

视频质量控制模块立即发送相应的命令给 DSP 编码模块,DSP 进行视频流编码码率调整。

2.2 参数估计

2.2.1 丢包率

RTCP 的 RR 包中丢包率字段并不直接反馈链路中包丢失事件的发生率,而只给出 RTCP 报告间隔内丢失包的数目及丢失包所占的比率。因此,发送端若直接利用该丢包率计算发送速率,将会导致网络数据传输速率的振荡。为了保证接收端多媒体码流的质量,应该尽可能减少传输速率振荡。对特定时间段内的包丢失报告进行平滑处理是缩减振荡的有效方法,根据平均算法^[9]计算丢包率:

$$p_i' = (1 - \delta) \times p_i + \delta \times p_{i-1}$$

其中, p_i 和 p_{i-1} 分别是第 i 和 $i-1$ 个 RR 包反馈的丢包率, p_i' 为估计的现在的丢包率, δ 为权重系数,取值范围为 $0 < \delta < 1$,这里可取值 0.3。

2.2.2 回环时间 RTT

发送端通过接收端反馈的接收报告块中的相关信息计算出回环时间,公式如下:

$$RTT = T_{now} - T_{start} - T_{delay}$$

其中, T_{now} 为服务器收到 RR 的当前时间, T_{start} 为客户端最近接收到的 SR 包中的 NTP 时间戳, T_{delay} 为客户端接收到来自服务器的 SR 包到发送 RR 包之间的时延。为了消除网络中的随机抖动, RTT 的估计值要采用 EWMA (Exponentially Weighted Moving Average) 方法进行加权平均,得到平滑回环时间即 SRTT:

$$RTT'_{i+1} = RTT'_i + (1 - \lambda) \times RTT_i$$

其中, RTT'_{i+1} 是第 $i+1$ 次估计值, RTT_i 是第 i 次的测量值, RTT'_i 是第 i 次的估计值。对于第一个 RTT 值,不作加权平均。 λ 是平滑因子,取 0.9,时间戳间隙规定转换成秒。

2.3 速率调整算法

设该系统服务器端视频流的最大码率为 R_{max} , 最小码率为 R_{min} 。

视频流在网络上传输分为两个阶段:起始阶段和拥塞控制阶段,拥塞控制阶段根据丢包率的不同又可以分为 3 个阶段。它们之间的状态转换如图 2 所示。

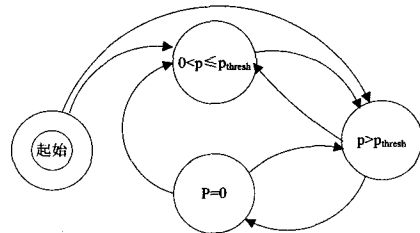


图 2 视频流状态转换图

其中, p_{thresh} 为客户端根据视频质量设置的丢包率阈值,当丢包率低于此值时,视频质量在可接受的范围内;高于此值时,视频质量严重下降。具体分析如下。

2.3.1 起始阶段

服务器端的媒体流在此阶段类似于 TCP 流的慢启动阶段。希望在稳定状态下,视频发送速率以 R_{min} 启动,在低于网络可用带宽阶段快速上升,之后则缓慢上升。采用 AIMD 算法的加增长策略每次都恒定的速率增长,容易引起 RTP 流的抖动,不能满足这一要求。因此,将加性因子改为一个变量,该变量随 R_i 的增长而减小,可以得到此阶段的算法:

$$R_{i+1} = R_i + \alpha_1 \times (R_{max} - R_i) \quad (p=0)$$

其中, $0 < \alpha_1 \leq 1$ 为常量。

2.3.2 拥塞控制阶段

此阶段丢包率 $p > 0$,可根据式(1)得出对 TCP 友好的 R_{cp} 。

随着视频编码码率的增大,服务器端发送速率也增大。当达到一定程度时,出现丢包, $p > 0$ 。此阶段有以下 3 种情况。

情况 1: $0 < p \leq p_{thresh}$ 。若此时 $R_i < R_{cp}$ 说明网络还有多余带宽可供使用,服务器端可以继续增加编码码率。但为发送速率平稳,防止网络出现较大抖动,要缓慢增加编码码率,丢包率越大增加得越慢,调整:

$$R_{i+1} = \min\{R_{max}, R_i + (\sqrt{p_{thresh} - p}) \times (R_{cp} - R_i)\}$$

若此时 $R_i \geq R_{cp}$,则服务器端需降低码率,调整 $R_{i+1} =$

$\max\{R_{kp}, R_{\min}\}$ 。

情况 2: $p > p_{thresh}$ 。此时视频质量已严重受影响,需降低码率,同时又要考虑平滑性,调整:

$$R_{i+1} = \max\{\min\{R_{kp}, (1 - \sqrt{p - p_{thresh}}) \times R_i\}, R_{\min}\}$$

并记下此刻的 R_i 为 R_{con} 。

情况 3: $p = 0$ 。经过拥塞调整后,视频流码率降低,丢包率减小。丢包率又为 0 时,服务器端又将增大编码码率。调整:

$$R_{i+1} = \begin{cases} \frac{R_{con} + R_i}{2} & (R_i < 0.9R_{con}) \\ \min\{R_i + \alpha_2 \times (R_{\max} - R_{\min}), R_{\max}\} & (R_i \geq 0.9R_{con}) \end{cases}$$

其中, $0 < \alpha_2 \leq 1$ 为常量。算法思想是先将码率增长至接近 R_{con} ,若此时丢包率仍然为 0,则保持一定速度缓慢增长。若又出现丢包率大于 0 的情况,则采用前两种情况处理。

3 系统测试结果与分析

本流媒体服务器在嵌入式 Linux 环境下开发,客户端可使用 VLC 播放器来进行测试,在实验室局域网内建立连接,网络状况采用开源软件 nistnet^[10] 进行模拟。本系统中, $R_{\max}$ 为 2Mbps, $R_{\min}$ 为 64kbps, α_1 和 α_2 分别取值 0.1 和 0.01, p_{thresh} 为 0.04。客户端与服务器建立连接后,不断接收到视频流并解码播放,并且服务器端每隔 2 秒收到一个来自客户端的 RR 包。通过 nistnet 模拟网络状况,同时观察客户端视频,发现画面流畅,且画面质量随网络带宽变化及时变化。服务器端通过串口打印出每次改变。取其中一段数据加以分析,如图 3 所示。

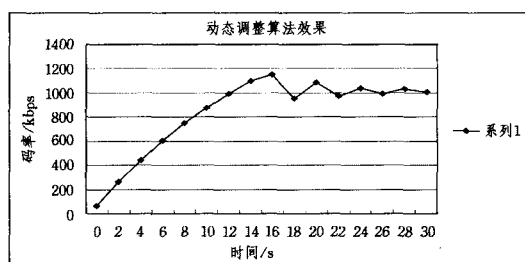


图 3 动态调整算法效果图

从图 3 可以看出,本系统能够实时地监测网络的状况,进而有效调整视频流的码率,提高客户端观看视频的质量。

结束语 本文针对网络视频监控系统,在现有算法 AIMD 和 TFRC 的基础上,提出了适用于实时流媒体的基于 RTCP 反馈信息的自适应动态 QoS 机制,实现了流媒体服务器根据网络状况自动调整编码码率进行传输,使客户端观看到实时流畅的画面。

参考文献

- [1] Tsai Chengfa, Tsai Chunwei. A new approach for solving large traveling salesman problem using evolutionary ant rules[C]// Proc. of International Joint Conference on Neural Network. New Jersey: IEEE Press, 2002: 1540-1545
- [2] 梁丰, 翁格奇. AIMD 拥塞控制算法分析[J]. 计算机工程与应用, 2004, 1: 173-174
- [3] 王光阳, 徐昌彪, 陈前斌. TFRC 与 TCP 流数之比对协议间公平性影响的研究[J]. 计算机应用研究, 2005, 22(11): 16-18
- [4] Handley M, Floyd S, Padhye J, et al. TCP Friendly Rate Control (TFRC); Protocol Specification[S]. IETF RFC 3448. 2003
- [5] Floyd S, Handley M, et al. Equation-based Congestion Control for Unicast Applications; The Extended Version[C]. 2000
- [6] Padhye J, Firoiu V, Towsley D, et al. Modeling TCP Throughput: A Simple Model and Its Empirical Validation[J]. IEEE/ACM Transactions on Networking, 2000, 8(2): 133-145
- [7] Schulzrinne H, Rao A, Lanphier R. Real Time Streaming Protocol[S]. RFC 2326. 1998
- [8] RTP: A Transport Protocol for Real-Time Application[S]. RFC 3550-2003
- [9] Sisalem D, Schulzrinne H, Emanuel F. The Direct Adjustment Algorithm: A TCP-Friendly Adaptation Scheme[R]. Berlin: Fraunhofer, GMD-FOKUS, 1997
- [10] Carson M, Santay D. Nistnet, a Linux-based network emulation tool[J]. Computer Communication Review (ACM SIGCOMM), 2003, 33(3): 111-126
- [11] 肖卫初, 陈伟宏. 监控系统中的多摄像头协同算法[J]. 重庆工学院学报: 自然科学版, 2008, 22(4): 117-123

(上接第 288 页)

- [4] Fisher S P B, Walker A, Wolfart E. Adaptive thresholding[M]. Department of Artificial Intelligence, University of Edinburgh, 1994
- [5] 苏金明, 王永利. MATLAB 图形图像(上)[M]. 北京: 电子工业出版社, 2005
- [6] Weszka J S. A survey of threshold selection techniques[J]. Computer Graphics and Image Processing, 1978, 7(2)
- [7] Haralick R M. Digital step edges from zero crossing of second

directional derivatives[M]. Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, 1987

- [8] Nalwa V S, Binford T O. On detecting edges[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, 8(6)
- [9] Otsu N. A Threshold Selection Method from Gray-Level Histograms[J]. Automatica, 1975(11)
- [10] 罗南超, 向昌成. 基于低频边缘特征和能量的多聚焦图像融合方法[J]. 重庆工学院学报: 自然科学版, 2009, 23(8): 168-171

(上接第 295 页)

参考文献

- [1] 叶俊勇. 人脸检测与识别方法研究[D]. 重庆: 重庆大学, 2002
- [2] Sobottka K, Pitas I. A novel method for automatic face segmentation, facial feature extraction and tracking[J]. Signal Processing: Image Communication, 2008, 12(3): 263-281
- [3] Craw I, Ellis H, Lehman J R. Automatic extraction of face-features[J]. Pattern Recognition Letters, 1987, 5(2): 183-187
- [4] Govindaraju V. Locating human faces in photographs[J]. International Journal of Computer Vision, 2006, 19(2): 129-146
- [5] Wang J, Tan T. A new face detection method based on shape information[J]. Pattern Recognition Letters, 21(6/7): 463-471
- [6] 闫文秀, 裴建岗, 孙颖, 等. 基于 Gabor 滤波器和改进 BP 神经网络的人脸检测方法[J]. 重庆工学院学报: 自然科学版, 2009, 23(4): 98-102