

一种类 TCP 流媒体拥塞控制算法

赵晓焱, 杨育捷, 毛文涛

(河南师范大学 计算机与信息工程学院, 河南 新乡 453007)

摘 要: 为了提高网络运行效率, 提出了一种采用 RTP/RTCP 协议基于模型的类 TCP 拥塞控制算法. 算法采用 TCP 确认机制, 对流量参考模型、往返时间及超时时钟值等参数进行了定义; 使用改进的 AIMD 机制进行窗口控制, 调整 RTP 数据帧的传输速率; 同时彻底摒弃 TCP 的重传机制以保证流媒体业务的连续性和实时性. 实验结果表明, 提出的算法与其他 RTP 流控算法相比, 具有较好的拥塞控制能力和公平性.

关键词: 拥塞控制; 实时传输协议; 往返时间; 速率控制

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2014)04-0108-04

TCP-Like Congestion Control Algorithm for Stream Media Transmission

ZHAO Xiao-yan, YANG Yu-jie, MAO Wen-tao

(Department of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China)

Abstract: In order to improve the network efficiency, this paper present a kind of TCP-Like congestion control algorithm used in the transmission based on RTP/RTCP. Algorithm uses the TCP validation mechanism and the flow model and the parameter definition of RRT and overtime clock value is presented. AIMD is used to improve window control mechanisms and adjust the transmission rate of RTP data frame. At the same time the TCP retransmission mechanism is completely abandoned to ensure the continuity of streaming and real-time. the experimental results show that the algorithm enhances the network bandwidth utilization and improves the congestion control capabilities and fairness.

Key words: congestion control; reliable transport protocol (RTP); round-trip time; rate control

1 引言

多媒体技术出现后, 网络中出现了实时性、连续性强、数据量大的流媒体业务. 与一般的数据业务不同, 流媒体业务对服务质量 QoS 有特殊的需求^[1], 如果使用传统的 TCP 协议传输, 会引起传输速率在短时间内大幅度下降, 导致流媒体业务质量严重降低, 因此需要选择合适的方法进行拥塞控制, 提高网络的性能.

历次 TCP 算法的改进基本上都是针对拥塞控制, 例如针对 TCP 协议的 AIMD 窗口控制机制^[2]

及改进算法^[3]、TCP Reno 慢启动算法及其改进策略^[4-5]、基于 TCP 友好拥塞控制^[6]及改进算法^[7]等. 不断改进的 TCP 协议机制兼容传统 TCP 业务的同时使得流媒体传输质量有所提高, 然而, TCP 协议的快速恢复、超时重传机制导致流媒体的连续性、实时性降低, 这些极大地限制了 TCP 协议在流媒体传输中的应用. 相对于 TCP 协议, 位于 UDP 协议之上的 RTP/RTCP 协议提供带有实时特性的端对端数据传输服务, 不要求重传丢包, 更加适用于流媒体业务. 然而, 目前的 RTP 流控算法^[8]反馈速率低, 对网络有效带宽的利用不够充分, 因此无法完全实现

收稿日期: 2013-05-30; 修回日期: 2013-07-22

基金项目: 国家自然科学基金(U1204609); 河南省重点科技攻关项目(102102210179); 河南师范大学青年科学基金项目(2011QK23)

流控自同步,而且由于过分抢占业务带宽,严重影响了传统的 TCP 业务.为了控制网络拥塞,更好地提高流媒体业务的传输质量,增强媒体流公平性,本文提出了一种基于 RTP/RTCP 协议的类 TCP 拥塞控制算法,将 TCP 的拥塞控制机制适当改进应用于 RTP/RTCP 协议之上.命名该算法为 TCP-RT.

2 TCP-RT 算法原理

TCP-RT 算法在标准 TCP 拥塞控制算法的基础上,主要进行了以下改进来适应基于 RTP/RTCP 的流媒体传输:

(1)流媒体数据往往是以一“帧”为处理单位,为了适于流媒体的处理、简化算法以加快处理速度,窗口控制对象、确认对象采用带有全局序号的 RTP 数据帧,窗口大小的单位也采用帧而不是字节.

(2)根据流媒体的特点,发送端不重传丢失的帧,防止重传造成进一步拥塞以及由于等待重传造成的流媒体的长时间停顿;

(3)接收端对没有收到的不发送重复确认帧,即不以任何方式通知发送端某一具体的帧丢失,以提高对拥塞的反应;

(4)发送端在收到 RTCP 确认帧后,根据确认收到帧的序号和尚未得到确认的帧数,判断是保持在拥塞避免状态还是进入快速恢复状态,不必等待重复确认,充分发挥网络层的 RED 机制的作用,以便更早期对拥塞做出反应;

(5)如果发生超时,为了降低传输速率下降的速度,滑动窗口值减半,进入快速恢复状态,而不是改为 1 进入慢启动状态,从而使传输速率降低的缓慢一些.

TCP-RT 算法由窗口控制、发送控制、接收确认、时钟控制等主要模块组成,实现拥塞控制机制.窗口控制模块根据接收端的确认帧和超时时钟控制模块的超时消息调整窗口大小,传输控制模块根据窗口大小来发送编码器送来的流媒体数据,时钟控制模块计算确认超时时钟的大小、进行计时提供超时消息,接收确认模块根据收到的数据帧向发送端发送确认帧.算法中各模块的相互关系如图 1 所示.

3 TCP-RT 参数的描述与计算

发送端根据接收端的反馈信息调整发送速率,保证流媒体业务的质量.算法采用 TCP 中的术语和参数描述,其中包括几个重要控制参数,pipe 为当前发送端已经发送但尚未收到确认的数据帧的数

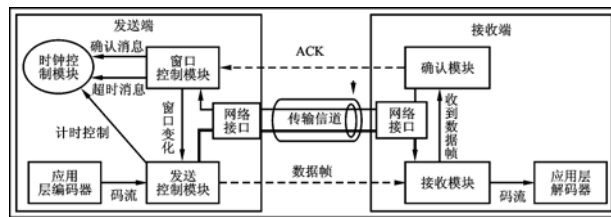


图1 类 TCP 算法组成模型

量,认为这些帧仍在发送端和接收端之间的信道里; $cwnd$ 为当前窗口大小指当前最多可以连续发送的帧数; $ssthresh$ 窗口调整阈值,控制窗口 $cwnd$ 增长的速度; RTT 为往返时间(round trip time); $Timeout$ 为确认超时时钟值,指一个数据帧从发送到收到接收端确认的时间限制,如果超过了这个时间仍然没有收到确认,则认为发生了丢帧,说明网络拥塞, $cwnd$ 、 $ssthresh$ 都是正整数, $cwnd$ 的最小值为 1, $ssthresh$ 的最小值为 2 且可以初始化为任何正整数; $pipe$ 初始化为 0.

3.1 往返时间 RTT

接收端反馈发送时间间隔是基于 RTT 的,RTT 的测量采用时间戳法.第一次的 RTT 计算直接使用 2 倍的 RTT_{sample} ,其中 $RTT_{sample} = Trcv - Tlsr - Tdlsr$,是实际 RTT 的采样值, $Trcv$ 是收到 RR 包的时间, $Tlsr$ 是上次发送 SR 包的时间, $Tdlsr$ 是上次发送 SR 包后的延迟.根据接收端的反馈包,发送方能够根据式 1 进行数据包的 RTT 估计:

$$RTT = \alpha \times \overline{RTT} + (1 - \alpha) \times RTT_{sample} \quad (1)$$

这里 $\overline{RTT}$ 是上次数据包的来回时间,RTT 是估计的数据包来回时间, α 是权值,起平滑作用,为了增强算法对延迟的反应,本文选择 0.87.

3.2 超时时钟值

确认超时时钟值的计算主要由时钟控制模块负责.在没有发生超时的情况下,超时时钟值 $Timeout$ 由式(2)计算所得.

$$Timeout = RTT + 4 * D \quad (2)$$

$$D = (1 - \beta) D + \beta * |RTT - M| \quad (3)$$

式中, α 、 β 为常数,一般 $\alpha = 1/8$, $\beta = 1/4$,RTT 为往返时间, M 为最近测量的传输过程中没有超时的往返时间, D 为一种偏移量,使 RTT 的变化平滑一些,同时规定 $Timeout$ 最小值为 1 s,避免不必要的超时.

3.3 参考模型

由于 TCP-RT 采用与 TCP 相似的拥塞控制机制,可以对 TCP 模型进行相应的修改计算有关业务

质量的参数. 算法由于保留了拥塞避免和快速恢复机制, 因而具备了一定的 TCP 友好性, 因此采用一个稳定状态下近似的 TCP 流量模型如式(4)所示.

$$B(p) = \min \left[B_{\max}, \frac{1}{RTT} \sqrt{\frac{3}{2bp}} \right] \quad (4)$$

式中, B 指传输速率, 表示单位时间发送的帧数, 也可以与平均帧长度相乘, 得出用比特数表示的传输速率; $B_{\max}$ 为发送端的最大编码速率或接收端允许的最大发送速率; p 指丢包率.

4 TCP-RT 算法的具体实现

4.1 接收端的确认机制

确认机制由接收端的确认模块实现. 由于流媒体业务对实时性和连续性的要求较高, TCP-RT 算法不处理发生乱序的流媒体数据包, 不要求重传丢包. 接收端对收到的数据帧进行确认, 每收到 b 帧 (不包括乱序的包) 向发送端发送一个 RTCP 确认帧, 通知发送端收到最后一帧的序号, 其中 b 为反馈速率 (其值由发送端通过 DES 帧通知接收端). 接收端每次发送一个确认帧后开始计时, 如果在规定的时间内没有收到下面 b 个数据帧, 则发送一个确认帧, 确认收到的最后一个 RTP 数据帧, 并将 b 值改为 1. 反馈速率 b 是正整数, 随窗口值的增长而增长, 这样在拥塞避免状态下可以使速率更稳定; 为了防止丢包后发生死锁而引起的超时, b 的最大值为 $\text{cwnd}/2$, 最小值为 1.

为了减少确认帧的数量从而降低网络负载和增强发送速率的稳定性, 发送端根据窗口值对接收端的传输速率进行控制.

4.2 传输控制机制

窗口模块根据反馈情况进行窗口控制, 调节传输速率, 是实现拥塞控制的核心. 窗口调整算法如下:

当确认超时时钟超时而没有收到确认帧时, 认为拥塞引起了丢包, 则令 $\text{pipe} = 0$, $\text{ssthreshold} = 3/4 * \text{cwnd}$, $\text{cwnd} = \text{cwnd}/2$, 使传输速率较平稳地下降, 或较快进入拥塞避免状态.

如在确认时钟超时之前发送端收到确认帧, 根据接收端所反馈的收到的最后一帧的序号, 利用式(5)计算实际接收和丢失帧的总数 r :

$$r = \text{pipe} - (\text{currentSentPacketNum} - \text{currentAckPacketNum}) \quad (5)$$

式中, $\text{currentSentPacketNum}$ 为当前发送的最后一个数据帧的序号, $\text{currentAckPacketNum}$ 为确认帧

中反馈的接收端收到的最后一帧的序号. 如果 $r \neq b$ 说明有丢包, 则调整窗口大小, 进入“快速恢复”状态, 但不重传丢失的帧, 直接发送新的数据帧, 与网络层的 RED 机制相适应, 增强了避免拥塞的能力. 如果 $r = b$ 说明没有丢包, 根据 cwnd 与 ssthreshold 关系确定传输进入慢启动或拥塞避免状态.

5 实验分析

为了考察算法在实际应用中的性能, 本文使用集成了 TCP-RT 算法的 Openphone 视频会议软件终端进行实验. 实验过程中, 设置两个终端向对方发送最高速率为 300 kb/s, 30 帧/秒的 H261CIF 格式的视频码流, 同时为了计算方便, 规定算法中接收端最高的反馈速率为 2.

5.1 拥塞控制能力

为了测试 TCP-RT 的带宽的利用能力和拥塞控制能力, 采用测试带宽程序发送干扰码流 (下文称为干扰程序) 占用带宽、制造拥塞, 同时与文献[8]中的 RTP/RTCP 流控算法, 进行了对比试验.

两种算法发送的视频在传输过程中, 每隔 15 分钟左右, 调整一次干扰程序的平均码率. 干扰程序平均码率的调整过程为: 64 kb/s, 128 kb/s, 256 kb/s, 128 kb/s, 64 kb/s, 实验结果如图 2、3 所示.

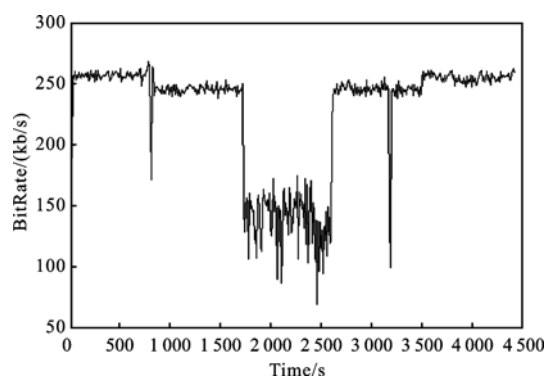


图 2 TCP-RT 算法传输速率随干扰程序平均码率变化的过程

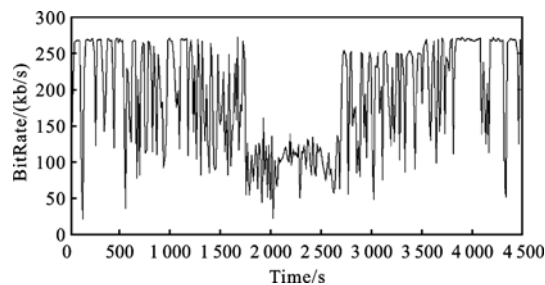


图 3 文献[8]算法传输速率随干扰程序平均码率变化的过程

实验结果表明, 在 75 分钟内, 在每个干扰码率不同的各个时间段内, TCP-RT 算法的传输速率基

本平稳;在干扰程序平均码率变化的时候,能较快的调整传输速率,防止在网络负载增加时造成拥塞,在网络负载降低时也能及时提高传输速率,提高流媒体质量.而文献[8]的算法在传输过程中,当可用带宽发生变化时,传输速率下降过大,而且传输速率不稳定、震动较大.当干扰程序平均码率增加导致可用带宽下降时,TCP-RT 程序的传输速率在 20~40 s 左右的时间内将发送速率降低,码率降低的过程比较平稳,变化曲线为阶梯状.这是由于算法中超时和快速恢复时对窗口的处理造成的效果,相对于标准的 TCP 算法,TCP-RT 在由于窗口下降速度较慢,保持了媒体流较高的连续性,提高了业务质量.

5.2 媒体流间的公平性

为测试 TCP-RT 媒体流之间分配带宽情况,实验过程中两对终端同时传输视频约 1 小时.实验结果如图 4 所示,两路视频可以较平均的分配链路的可用带宽,保证音频、视频码流的同步,从而提高流媒体业务质量.

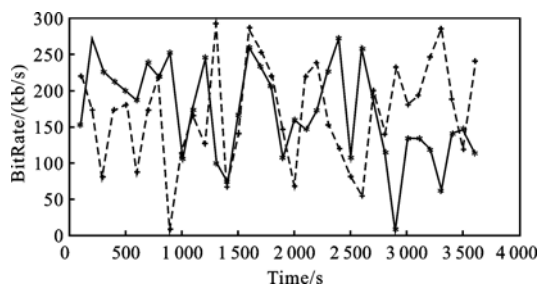


图4 两路视频公平性试验结果图

6 结束语

本文提出了一种基于 RTP/RTCP 协议的类 TCP 拥塞控制算法,将 TCP 拥塞控制机制合理地融合到基于 RTP/RTCP 协议的流媒体传输算法中,使两种协议的拥塞控制算法的优势互补,提高了基于 RTP/RTCP 协议流媒体业务的质量.但是当可用带宽降低较大时,码率的变化较快且没有一般 RTP 算法平稳,下一步的工作中可以考虑进一步改进参数的计算模型,更精确的计算吞吐量和延迟抖

动,使速率的变化曲线更平滑,更好地提高流媒体业务质量.

参考文献:

- [1] 麻海圆,孟相如,马志强,等.基于种群生态的组播拥塞控制机制[J].计算机应用,2011,31(12):3195-3199.
- [2] Hayder Natiq Jasem, Zuriati Ahmad Zukamain, Mohamed Othman, et al. The TCP-based new AIM D congestion control algorithm[J]. International Journal of Computer Science and Network Security, 2008, 8(10): 331-338.
- [3] Wu Yuan-ban, Liu Zhen-sheng, Zhang Wen-liang. Congestion control mechanism of study based expanded AIMD [J]. Computer Engineering, 2008, 34 (9): 232-234.
- [4] 王强,普杰信,刘伟. TCP 拥塞控制慢启动策略的研究[J].微电子学与计算机,2007,24(12):210-212.
- [5] Attiya, Gamal. New strategy for congestion control based on dynamic adjustment of congestion window [J]. International Journal of Computer Science Issues, 2012, 9(2):368-377.
- [6] 肖甫,王汝传,孙力娟,等.基于 TCP 友好的无线网络拥塞控制机制研究[J].计算机科学,2010,37(7):50-53.
- [7] Shiang Hsien-Pol, Van Der Schaar Mihaelal. A quality-centric tcp-friendly congestion control for multimedia transmission[J]. IEEE Transactions on Multimedia, 2012, 14(3): 896-909.
- [8] 匡全,李晓勇,白英彩.基于 RTP/RTCP 的分层多播拥塞控制算法的研究[J].计算机应用与软件,2006,23(2):80-82.

作者简介:



赵晓焱 女,(1981—),硕士,讲师.研究方向为多媒体网络通信.