

文章编号:1671-5497(2006)03-0433-05

一种改进的 TCP 拥塞控制算法

杨晓萍, 史 帅, 陈 虹

(吉林大学 通信工程学院, 长春 130022)

摘 要:传统的 TCP 拥塞控制算法在遇到丢包时发送速率会有剧烈抖动, 不适合传输实时多媒体数据流。本文在 TCP 拥塞控制算法加增乘减(AIMD)基础上提出一种改进算法慢增慢减(SISD)算法。仿真结果表明:该算法同传统 TCP 拥塞控制算法相比, 有较好的发送速率平滑性, 减小了数据延时抖动, 加入了多媒体流的拥塞调整机制。本算法可作为实时多媒体数据传输研究的基础, 同时也改善了 TCP 流的传输质量。

关键词:通信技术; TCP 拥塞控制; 加增乘减(AIMD); 慢增慢减(SISD); 服务质量(QoS)

中图分类号: TN915 **文献标识码:** A

Improved algorithm for TCP congestion control

Yang Xiao-ping, Shi Shuai, Chen Hong

(College of Communication Engineering, Jilin University, Changchun 130022, China)

Abstract: The traditional TCP congestion control algorithm is unsuitable for the multimedia traffics because of the fact that when a packet is discarded, the sender halves its rate, resulting in sending rate wobbling greatly. An enhanced congestion control algorithm, slow increase slow decrease(SISD), was proposed based on the algorithm additive increase multiplicative decrease(AIMD). The simulation results show that SISD algorithm has better smoothness of sending rate compared with the congestion control algorithm of TCP, which reduces the fluctuation of sending data and adds the congestion control to multimedia traffics. This algorithm can act as development basis of real-time multimedia traffics, and improve the transmission quality of TCP traffics.

Key words: communication technology; TCP congestion control; additive increase multiplicative decrease(AIMD); slow increase slow decrease(SISD); quality of service(QoS)

0 引 言

随着网络技术的发展和普及, 网络中实时多媒体应用比例迅速加大, 如 IP 电话、视频会议、远程教育等。这些多媒体流量与普通的文件数据传

输有着不同的服务质量(QoS, Quality of Service)要求: 视频、音频等要求数据流的平滑性, 即发送方的发送速率不能过于抖动; 而多媒体流不要求可靠传输, 对超时或失序的数据包不需要重传, 但对数据延时比较敏感。由于 TCP 传输控制的大抖动不适合传送多媒体数据流, 这些新的应用中

收稿日期: 2005-07-25.

基金项目: 国家自然科学基金资助项目(60234030); “863”国家高技术研究发展计划项目(2003AA412210).

作者简介: 杨晓萍(1963-), 女, 副教授, 博士研究生. 研究方向: 多媒体流控制及网络拥塞控制.

E-mail: yangxiaop2003@yahoo.com.cn

通讯联系人: 陈虹(1963-), 女, 教授, 博士生导师. 研究方向: 网络优化控制和预测控制. E-mail: chen@jlu.edu.cn

很多都使用 UDP 协议。UDP 协议本身不响应拥塞,这损害了网络的鲁棒性,可能引发网络拥塞并最终导致网络崩溃^[1,2]。为了解决这个问题,有人建议采取资源预留协议^[3](RSVP)或区分服务^[4](DiffServ)的方式来传输多媒体流。但多媒体流实时性的特点导致采用这些方式的传输成本很高,大多数用户需要的是用尽可能低廉的价格传输多媒体流,这种低廉价格的服务仍是尽力而为(Best-effort)服务。可见,研究 Best-effort 服务模式下载传输多媒体数据是非常有意义的。

TCP 协议不适合传送多媒体数据流的根本原因在于网络发生拥塞时它的发送速率剧烈震荡^[5],针对 TCP 拥塞控制,作者在加增乘减(Additive Increase Multiplicative Decrease, AIMD)算法思想的基础上提出一种改进的发送速率调节的慢增慢减(Slow Increase Slow Decrease, SIRD)算法,以解决 TCP 发送速率震荡大的问题。该算法具有拥塞响应机制,并且带宽利用率较高。

1 TCP 拥塞控制算法

TCP 的拥塞控制算法是由 Jacobson^[6]提出的,使用 AIMD 的窗口调节机制。算法用数学公式表示为:

$$I: W_{t+R} \leftarrow W_t + \alpha \quad (\alpha > 0)$$

$$D: W_{t+R} \leftarrow \beta \times W_t \quad (0 < \beta < 1) \quad (1)$$

式中: I 为在一个回路响应时间(round-trip time, RTT)内接收到 ACK 确认包而引起窗口(速率)增加的算法; W_t 为 t 时刻窗口的大小; R 为回路响应时间 RTT; α 是加性增加因子; D 表示遇到拥塞后窗口(速率)减小的算法; β 是乘性减少因子。

Chiu 和 Jain^[7]研究了 AIMD 算法的稳定性和公平性,指出了 α 、 β 应满足的条件,如式(1)所示。在传统 TCP 中 $\alpha = 1, \beta = 0.5$ 。

TCP 的发送速率控制公式如式(2)所示^[8]

$$S \approx \frac{1}{R \sqrt{\frac{2bp}{3}} + T_0 \min\left(1, 3 \sqrt{\frac{3bp}{8}}\right) p(1 + 32p^2)} \quad (2)$$

该式将发送速率 S 表示为 R 、 b 、 p 的函数。其中 p 值是在假定该轮(round)的第一个包或随后的包没有丢失的前提下,一个包发生丢失的概率; T_0 代表超时值; b 代表每个 ACK 确认的数据包个数。

基于上述的 TCP 控制算法和速率公式,推导出慢增慢减算法 SIRD。

2 SIRD 拥塞控制算法

SIRD 算法仍使用 AIMD 模型,如式(1)所示。SIRD 从慢启动开始,在慢启动阶段,每收到一个包确认后,窗口大小增加一个,即每个 RTT 时间窗口大小加倍。当收到拥塞指示后,窗口大小减半进入拥塞避免阶段,在这个阶段,每确认收到一个新的 ACK,拥塞窗口大小增加 α/W_t 。源端如果收到三个重复的 ACK,则将窗口减少到 βW_t ,如果发生超时则窗口降为 1。SIRD 算法的关键是确定合适的 α 、 β 值。

2.1 推广发送速率公式

将增加因子 α 和减少因子 β 作为变量加入到式(2),得到式(3):

$$S(\alpha, \beta) \approx \frac{1}{R \sqrt{\frac{2bp(1-\beta)}{\alpha(1+\beta)}} + T_0 \min\left(1, 3 \sqrt{\frac{bp(1-\beta^2)}{2\alpha}}\right) p(1 + 32p^2)} \quad (3)$$

显然式(2)是式(3)取 $\alpha = 1, \beta = 0.5$ 时的特殊情况。为方便讨论,将式(3)分母的两部分分别定义为 SD 和 SO :

$$SD(\alpha, \beta) = R \sqrt{\frac{2bp(1-\beta)}{\alpha(1+\beta)}} \quad (4)$$

$$SO(\alpha, \beta) = T_0 \min\left(1, 3 \sqrt{\frac{bp(1-\beta^2)}{2\alpha}}\right) p(1 + 32p^2) \quad (5)$$

如果所有的拥塞都不发生定时超时,则分母就只剩下 SD ,说明 SD 可以不依赖于 T_0 。当拥塞有超时发生时,就需要加入 SO , SO 也不直接依赖于 R 。比较这两部分可知,当 p 很小时, SD 比 SO 更占主导地位,此时 S 主要由 SD 决定;而当 p 增加时, SO 逐渐变大,最后增加到 1。

Chiu 和 Jain 在理论上指出了应用 AIMD 能以最快的速度达到最大限度利用带宽,且能保证最大的公平性。根据当时网络发展的实际状况,其研究没有考虑到多媒体应用。作者考虑到多媒体应用的要求,对算法做了改进研究,下面主要讨论 (α, β) 的选取方法。

2.2 SIRD 算法的参数选取

TCP 流发送速率剧烈抖动的原因在于它的增

加因子 α 为 1, 减少因子 β 为 0.5, 这对实时多媒体流来说太过激进。SISD 算法的设计目标就是通过减小拥塞调节幅度来减小发送速率的振荡幅度, 使发送速率变化平缓, 可见 β 应在区间 (0.5, 1) 内选取。目前的网络多媒体应用中未对平滑性给出确切指标, 故根据仿真研究, 取 $\beta = 0.8$, 此时可以兼顾发送速率平稳和算法响应速度的性能要求。

目前网络中大部分数据流仍是 TCP 流, TCP 友好性是一个源端控制算法必须考虑的重要性能指标。TCP 友好是指在相同条件下传输速率不大于 TCP 流的数据流^[9]。从式 (3) 可知, 通过控制 (α, β) 的取值可得到期望的发送速率 $S(\alpha, \beta)$, 即选择参数 (α, β) 的值, 使 SISD 流等于 TCP 流的发送速率 $S(\alpha = 1, \beta = 0.5)$:

$$S(\alpha, \beta) = S(\alpha = 1, \beta = 0.5) \quad (6)$$

当 $\beta = 0.8$ 时, 通过选取 α 的值, 使在较大的丢包率范围内 SISD 和 TCP 发送速率的差值最小。下面定义误差方程为:

$$E_{\beta}(\alpha) = \int_0^1 w(p) \left| \frac{S(\alpha, \beta)}{S(\alpha = 1, \beta = 0.5)} - 1 \right| dp \quad (7)$$

式中: $w(p)$ 为关于 p 的函数, 它允许丢包率在优化中占较大的比重权值。这里只考虑对丢包率小于一个阈值 ($p = 0.1$) 时权值取 1、大于它时取 0 的简单情况, 图 1 画出了权值函数曲线。图 2 显示了当 $\beta = 0.8, T_0 = 4$ RTT 时 $E_{\beta}(\alpha)$ 的曲线, 由该曲线图可见, α 在 (0.2, 0.4) 范围内取值时, 差值较小。经过仿真实验比较, 选定 $\alpha = 0.28$ 。可见, 通过计算误差函数 $E_{\beta}(\alpha)$ 的方法, 对给定的 β 可以找到合适的 α 取值范围。

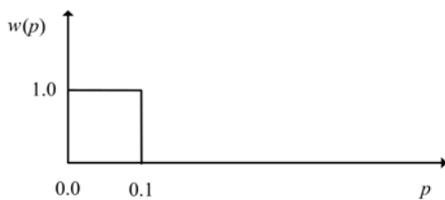


图 1 权值函数曲线

Fig. 1 Curve of weighted function

一般在网络丢包率较低时, SD 占主导地位, SO 差别不大, 由式 (5) 近似估计有:

$$\sqrt{\frac{bp(1-\beta^2)}{2\alpha}} = \sqrt{\frac{bp(1-\beta^2)}{2\alpha}} \Big|_{\alpha=1, \beta=0.5}$$

$$\alpha = \frac{4}{3}(1-\beta^2) \quad (8)$$

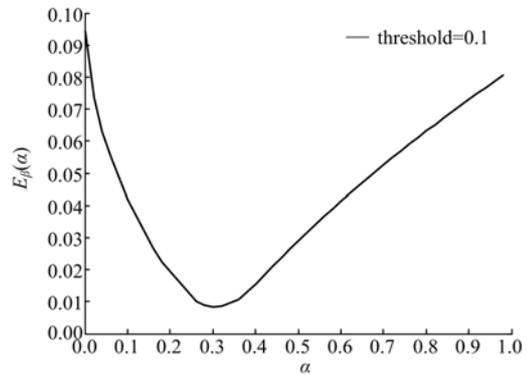


图 2 误差曲线

Fig. 2 Curve of errors

3 算法仿真

评价源端拥塞控制算法, 应该考虑它的友好性、平滑性、响应性及侵略性。友好性是指竞争流间的发送速率差别不大, 考察方法是看这个算法的长期平均发送速率是否与 TCP 流相当。平滑性是指在一个静态环境下, 某一数据流在一时间段内发送速率变化较小。响应性是指当网络拥塞变化时, 源端的发送速率能响应拥塞而跟随其变化。侵略性是指当可用带宽增加时, 算法能快速增加源端的发送速率, 以提高网络的利用率。

使用 NS2^[10] 做仿真平台, 拓扑结构如图 3 所示。n0、n1 为发送端, n2、n3 为转发路由器, n4、n5 为接收端。n0 发送 SISD 数据流至 n4; n1 发送 TCP 数据流至 n5, 它们共同竞争瓶颈链路的带宽。应用层均为 ftp 协议。

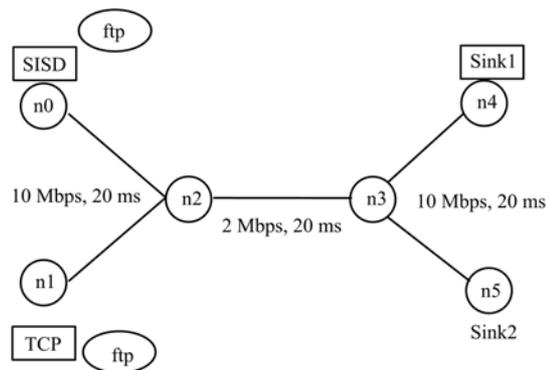


图 3 网络拓扑结构

Fig. 3 Network topology

3.1 友好性与平滑性

n0 和 n1 同时发送数据, 持续到规定时间后

同时关闭。首先统计了在 3 个不同长度时间段内,使用 RED 和 Droptail 两种队列管理算法(使用典型参数配置,见文献[11])的情况下 TCP 和 SISD 的平均发送速率,结果见表 1。

表 1 TCP 和 SISD 的平均发送速率

Table 1 Average sending rates of TCP and SISD flows

仿真 时间/s	队列管 理算法	平均发送速率/(Mb·s ⁻¹)	
		TCP	SISD
10	RED	0.84	0.915
10	Droptail	0.88	0.816
100	RED	0.808	1.02
100	Droptail	0.79	1.04
200	RED	0.812	1.03
200	Droptail	0.788	1.05

由表 1 可知,队列管理无论使用 Droptail 还是 RED,SISD 源的平均发送速率基本上略高于 TCP 源,但相差不大。说明 SISD 流竞争带宽的能力比 TCP 流略强,但仍为友好流。

为了考察平滑性和响应性,该仿真记录了平均发送速率的瞬时值,如图 4、图 5 所示。无论是使用 RED 还是 Droptail 队列管理算法,SISD 源的发送速率振荡幅度都远小于 TCP 源,即 SISD 的平滑性远好于 TCP。当拥塞发生时 SISD 会对拥塞做出响应,说明 SISD 是响应流。

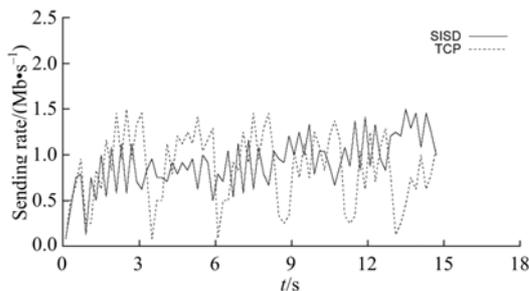


图 4 SISD 与 TCP 在 Droptail 下的发送速率比较
Fig. 4 Comparison of sending rates between SISD and TCP under Droptail gateway

3.2 侵略性

拓扑结构不变,n0 和 n1 同时开始发送数据,8 s 后其中一个数据源停止发送数据,记录发送速率如图 6、图 7 所示。比较两图可知,当可用带宽出现时,SISD 响应变化的速度较慢,本实验中,SISD 约花 5 s 时间,而 TCP 在 2 s 内就完成了,即 SISD 重新调整到稳定状态的时间长于 TCP。这是因为 SISD 窗口调整时的增加和减小幅度都较小,因此会得到比 TCP 小的震荡幅度。但作为

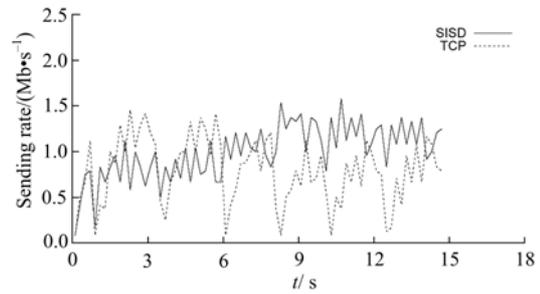


图 5 SISD 与 TCP 在 RED 下的发送速率比较
Fig. 5 Comparison of sending rates between SISD and TCP under RED gateway

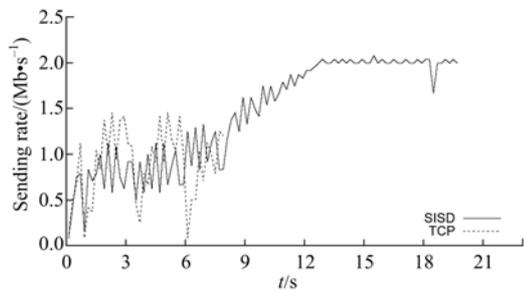


图 6 SISD 的侵略性

Fig. 6 Aggressiveness of SISD flow

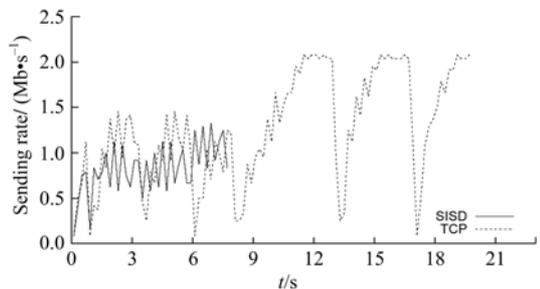


图 7 TCP 的侵略性

Fig. 7 Aggressiveness of TCP flow

代价 SISD 在网络没有发生拥塞时,它的窗口增加速度要小于 TCP,导致它在探测可用带宽时比 TCP 慢。

4 结束语

针对 TCP 拥塞控制算法不适合传输实时多媒体流的问题,对其核心算法 AIMD 进行了改进。推广了 TCP 发送速率公式,重点给出了 SISD 算法中(α, β)参数的选取方法。在首先选定 β 值时,当 β 取值较大时, SD 和 SO 的值较小,计算出的 $S(\alpha, \beta)$ 较大。通过前述方法计算出的 α 会减小。此时的 SISD 算法调整拥塞时比较保守,发送

源端拥塞窗口的增加和减小的速度都较慢,具有更好的发送速率平滑性,但在收到拥塞指示后发送速率的调整速度慢。相反地,如果 β 取值较小,计算出的 α 较大,SISD调整拥塞比较激进,使发送源端的速率平滑性变差,但调整速度较快。对于不同的网络环境, (α, β) 的最优取值将不同。如果将此算法的参数实现自适应配置,从而使网络数据传输层协议的UDP被TCP所取代,而TCP中的核心拥塞控制算法AIMD被SISD所取代,则可以避免多媒体数据传输不响应拥塞控制及其带来的一系列后果。故这方面的研究将是下一步工作的重点。

参考文献:

- [1] 肖萍萍,杨晓萍,陈虹. 基于RED分组丢失历史方法的非TCP流鉴别[J]. 计算机工程与应用, 2005, 41(7): 153-156.
Xiao Ping-ping, Yang Xiao-ping, Chen Hong. Identification of no-TCP flows based on the history of RED packet drops[J]. Computer Engineering and Application, 2005, 41(7): 153-156.
- [2] 杨晓萍,陈虹,肖萍萍. UDP流量对网络性能影响及对策[C]//第24届中国控制会议论文集. 广州:华南理工大学出版社, 2005: 245-248.
Yang Xiao-ping, Chen Hong, Xiao Ping-ping. Effects and schemes of UDP flows for network performance[C] //Proceedings of the 24th Chinese Control Conference. Guangzhou: South China University of Technology Press, 2005: 245-248.
- [3] David D C, Scott S, Lixia Z. Supporting real-time applications in an integrated services packet network: architecture and mechanism[C] // Proceedings of ACM SIGCOMM'92. New York(USA): ACM Press, 1992: 14-26.
- [4] Blake S, Black D, Carlson M. An architecture for differentiated services[EB/OL]. [1998-12-05] <http://www.rfc.net/rfc2475.html>.
- [5] Kevin F, Floyd S. Simulation-based comparisons of Tahoe, Reno, and SACK TCP[J]. ACM Computer Communication Review, 1996, 26(3): 5-21.
- [6] Jacobson V. Congestion avoidance and control[J]. ACM Computer Communication Review, 1988, 18(4): 314-329.
- [7] Chiu D M, Jain R. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks[J]. Computer Networks and ISDN Systems, 1989, 17(1): 1-14.
- [8] Padhye J, Firoiu V, Towsley D, Kurose J. Modeling TCP throughput: a simple model and its empirical validation[C] // Proceedings of ACM SIGCOMM'98, Vancouver, B. C. New York(USA): ACM Press, 1998: 303-314.
- [9] Braden B, Clark D, Crowcroft J. Recommendations on queue management and congestion avoidance in the Internet[EB/OL]. [1998-04-08] <http://www.rfc.net/rfc2309.html>.
- [10] UCN/LBL/VINT. Network Simulator-NS2[DB/OL]. [2003-09-04] <http://www.mash.cs.berkeley.edu/ns>.
- [11] 杨晓萍,陈虹,翟双. 基于路由器的RED和Droptail算法比较[J]. 吉林大学学报:信息科学版, 2005, 23(1): 69-74.
Yang Xiao-ping, Chen Hong, Zhai Shuang. Comparison of RED and Droptail algorithms based on router[J]. Journal of Jilin University (Information Science Edition), 2005, 23(1): 69-74.

(责任编辑 程仲基)