第 35 卷 第 4 期 计 算 机 工 程 2009 年 2 月
Vol.35 No.4 Computer Engineering February 2009
·网络与通信· 文章编号：1000—3428(2009)04—0110—03 文献标识码：A 中图分类号：TP393

一种基于 RTT 公平性的 TCP 慢启动算法

王建峰[1]，黄国策[1]，陈才强[2]，朱 蒙[1]
(1. 空军工程大学电讯工程学院，西安 710077；2. 空军通信技术网络管理中心，北京 100843)

摘 要：分析标准慢启动算法应用于包含 GEO 卫星链路的网络时存在的问题，提出一种基于 RTT 公平性的 TCP 慢启动改进算法。改进算法采用大初始窗口机制，慢启动初期窗口保持指数增长，慢启动后期引入窗口增长控制因子，使 RTT 较大的窗口增加较快，反之增加较慢。性能分析和仿真结果表明，改进算法可以在慢启动后期减缓拥塞窗口的增长速度，削弱 RTT 较小的 TCP 流竞争带宽的侵略性，在一定程度上保证不同 RTT 数据流共享带宽的公平性。

关键词：GEO 卫星网；拥塞控制；RTT 公平性

TCP Slow Start Algorithm Based on RTT Fairness

WANG Jian-feng[1], HUANG Guo-ce[1], CHEN Cai-qiang[2], ZHU Meng[1]
(1. Institute of Telecommunication Engineering, Air Force Engineering University, Xi'an 710077;
2. Air Force Communications Technology Network Management Center, Beijing 100843)

【Abstract】 Thi paper analyzes the problems in standard TCP slow start algorithm when it is applied to the network environment including GEO satellite links. To solve these problems, a new improved TCP slow start algorithm based on RTT fairness is proposed, termed RFB-SS. RFB-SS algorithm uses the larger initial window mechanism. And in the early slow start phase, the congestion window remains exponential growth, in the late phase, by using window control growth factor, the congestion window of larger RTT increases faster, otherwise increases slower. Performance analysis and simulation results show that RFB-SS algorithm slows down the congestion window growth rate, and weakens the TCP flow of smaller RTT in the aggressive bandwidth competition in the late slow start phase. To a certain extent, RFB-SS algorithm obtains a fair sharing of available bandwidth among different RTT flows.

【Key words】 GEO satellite networks; congestion control; RTT fairness

1 概述

GEO 卫星链路大带宽长时延的特性给 TCP 应用带来了一定的局限性，而慢启动策略的优劣对 TCP 拥塞控制算法的性能有着明显的影响。

在标准慢启动阶段，拥塞窗口 *cwnd* 的初始值设置为 1 个分组，源端每收到 1 个确认信号 ACK，*cwnd* 的值就增加 1 个分组，与 ACK 的到达速度呈指数增长。假设 $ssthresh=2^{N+1}$，所有 RTT 均相等，经过前 N 个 RTT，*cwnd* 从 1 增加到 *ssthresh*/2，而在第 N+1 个 RTT 时，*cwnd* 将从 *ssthresh*/2 增加到 *ssthresh*。在慢启动初期，由于拥塞窗口初始值为 1，窗口增长缓慢，在 GEO 卫星网长时延的特点下，其所需的若干个 RTT 时间很长；而在慢启动后期，窗口增长速度过快，尤其最后一个 RTT 的突发流量将对网络的稳定性造成很大的影响，易造成全局同步网络性能下降，这与慢启动逐渐探测网络带宽的设计初衷有冲突。

拥塞窗口 *cwnd* 的增长速度与 ACK 的到达速度呈指数关系，而 ACK 的到达速度与 RTT 呈反比，也就是拥塞窗口 *cwnd* 的增长速度与 RTT 的大小呈负指数关系。因此，RTT 较小的 TCP 流对网络资源的占用具有较大的侵略性，RTT 较大的链接只能获得部分应得资源，由此带来相同瓶颈链路上不同 RTT 流之间竞争带宽的不公平性，这种现象在包含 GEO 卫星链路的网络环境中尤其明显。

针对标准慢启动算法存在的问题，研究者提出了一系列改进方法。文献[1]提出 Smooth Start，当 *cwnd* 接近 *ssthresh* 时，逐步减慢 *cwnd* 的增长速度。文献[2]提出了 Limited Slow Start，引入了一个新的慢启动临界值 *max-ssthresh*，当拥塞窗口 *cwnd* 大于 *max-ssthresh* 时，改变 *cwnd* 的增长方式，问题在于 *max-ssthresh* 难以确定。当慢启动增长到 *ssthresh*/2 之后，引入一个比例因子，随着 *cwnd* 的增大，逐渐减慢其增长速度[3]。文献[4]提出分段慢启动改进算法 P-start，当 *cwnd* 大于 *ssthresh*/2 时，每个 *RTT* 增加(*ssthresh*-*cwnd*)/2。这些改进方法的共同思想是，当发送速率接近网络带宽时，逐渐减少发送窗口增大幅度，确保平滑逼近网络带宽过渡到拥塞避免阶段，有效地减少了一个窗口内出现多个分组丢失的现象，避免了粗粒度的重传超时问题。此外，文献[5]提出了使用估计的 *ssthresh* 设定一个安全的 *ssthresh*，但由于网络动态变化，因此难以保证估计的门限阈值与网络可用带宽的完全匹配。

但上述改进算法几乎没有涉及相同瓶颈链路上不同 RTT 间竞争带宽不公平性的改进。本文提出一种能改善不同 RTT 竞争带宽不公平问题的慢启动机制，称为基于 RTT 公平性的慢启动算法(RTT-Fairness-Based Slow Start algorithm, RFB-SS)。

基金项目：国家部委科研基金资助项目
作者简介：王建峰(1978－)，男，助教、硕士，主研方向：短波通信，卫星 TCP 拥塞控制；黄国策，教授、博士生导师；陈才强，高级工程师；朱 蒙，硕士研究生
收稿日期：2008-07-22 **E-mail：**wjf4206@126.com

2 改进的 TCP 慢启动算法 RFB-SS

RFB-SS 算法主要从以下 3 个方面考虑：(1)针对不同 RTT 流之间竞争带宽的不公平性问题，为了减小对长时延链接的歧视，需要提高长时延链接打开窗口的速度，同时抑制短时延链接的窗口打开速度。(2)针对初始拥塞窗口值过小问题，把初始窗口值设置为 2，可节省 1 个 RTT 时间。(3)针对慢启动后期窗口增长过快的问题，在窗口接近慢启动阈值时逐渐减缓窗口增加速度。

基于以上考虑，改进算法的基本思想是：在连接建立初期，拥塞窗口 *cwnd* 的增长方式与标准慢启动机制相同，呈指数增长。当 *cwnd* 增至 *ssthresh*/2 时，引入拥塞窗口增长控制因子，使每个 RTT 内增加(*ssthresh-cwnd*)×(*RTT*/(*RTT*$_{max}$+*RTT*))逐步逼近门限值，直到(*ssthresh-cwnd*)小于设定的调节因子δ(*ssthresh*/≥δε2)，同时将拥塞窗口设为门限值，从而进入拥塞避免阶段。拥塞窗口 *cwnd* 的变化可表示为

$$cwnd(t+T)=\begin{cases}2\times cwnd(t) & \text{if } cwnd < ssthresh/2\\ cwnd(t)+\dfrac{RTT}{RTT+RTT_{max}}\times(ssthresh-cwnd(t)) & \text{if } ssthresh/2\le cwnd(t)<ssthresh\\ & \text{and } ssthresh-cwnd\ge\delta\\ ssthresh & \text{otherwise}\end{cases}$$

RFB-SS 算法削弱了 RTT 较小的 TCP 流竞争带宽时的侵略性，缓解了不同 RTT 流间竞争带宽的不公平性，并且减缓了慢启动后期拥塞窗口的增长速度。具体描述如下：

Step 1 初始化，即 *cwnd* = 2；

ssthresh=65 535 Byte；

$\delta=\delta_0$。

Step 2 发送 *cwnd* 个分组。

Step 3 当正确的 ACK 到达时，每个 RTT 内窗口大小调整如下

```
If (cwnd < ssthresh)
  {If (cwnd < ssthresh/2)
      {cwnd = 2*cwnd ;}
    Else if (cwnd≥ssthresh/2 && ssthresh – cwnd≥δ)
  {cwnd = cwnd + (ssthresh-cwnd)* (RTT/(RTTmax+ RTT)) ;}     }
Else
  {cwnd=ssthresh;
go to Step 5;}
```

Step 4 若接收到 3 个重复 ACK，进入快速重传和快速恢复阶段；若应答超时，重新进入慢启动阶段，转 Step 1，并修改 Step 1 中的窗口初始值为 1。

Step 5 进入拥塞避免阶段。

其中，δ_0为设定的调节因子。

3 RFB-SS 算法性能分析

假设 *ssthresh*=2^{N+1}，改进算法与标准慢启动算法相比，具有以下特点：

(1)在拥塞窗口增长到 *ssthresh*/2 之后，*cwnd* 的增长速度随 RTT 的不同而不同，每个 RTT 内增加(*ssthresh-cwnd*)的(*RTT*/(*RTT*$_{max}$+*RTT*)) 倍。令 *k*=(*RTT*/(*RTT*$_{max}$+*RTT*))，则当 *RTT*＝*RTT*$_{max}$ 时，*k* =1/2；当 *RTT*＝*RTT*$_{max}$/2 时，*k*=1/3；当 *RTT*＝*RTT*$_{max}$/3 时，*k*=1/4；依此类推，当 *RTT*＝*RTT*$_{max}$/γ时，*k*=ι/(γ+1)，其中，γε1。可以看出，随着 RTT 的减小，*cwnd* 增长的速度逐渐减慢，这样，RTT 较大的 TCP 流获得较多的增加带宽，而 RTT 较小的 TCP 流获得较少，缓解了不同 RTT 流之间竞争带宽的不公平性。

(2)在慢启动初期，初始窗口值 W_1=2，慢启动所需时间将节省 lbW_1 个 RTT，即经过前(*N*-1)个 RTT，*cwnd* 即可从 2 增加到 *ssthresh*/2。

(3)在拥塞窗口增长到 *sthresh*/2 之后，随着 *cwnd* 的增大，*cwnd* 逐渐减缓增长速度，在一个 RTT 内，最多将 *cwnd* 增加(*ssthresh-cwnd*)/2。当 *RTT*=*RTT*$_{max}$，δ=*ssthresh*/8，拥塞窗口 *cwnd* 在第 *N* 个 RTT 时从 *ssthresh*/2 增加到 3×*ssthresh*/4；在第(*N*+1)个 RTT 时，*cwnd* 从 3×*ssthresh*/4 增到 7×*ssthresh*/8；此时，(*ssthresh-cwnd*)≤δ=*ssthresh*/8，因此，在第(*N*+2)个 *RTT* 时，*cwnd* 从 7×*ssthresh*/8 增加到 *ssthresh*；其所需 *RTT* 次数只比标准慢启动算法多 1 次，但改进算法在慢启动后期，*cwnd* 增长趋于缓慢，能较平滑地进入拥塞避免阶段，延缓拥塞的发生，减少网络的突发流量和分组丢失的数量，进而减轻瓶颈路由的瞬时负载，降低路由器缓存溢出的风险；同时在一定程度上减少了慢启动阶段同一个窗口多个分组丢失的现象，避免了慢启动阶段的重传超时。

随着 RTT 值的增大，到达慢启动阈值 *ssthresh* 所需的 *RTT* 次数减少，由于慢启动所需要的时间等于 *RTT* 与所需的 *RTT* 次数的乘积，因此采用改进算法在一定程度上缓解了不同 RTT 数据流竞争带宽的不公平性。

(4)改进算法实现简单、开销小，适用于任意 TCP 源端算法。即针对大的初始窗口值，只需要在发送端改变 TCP 堆栈，且只增加一个比较运算和一个算术运算，对于当前的高速计算机来说，性能上几乎没有影响。

4 仿真结果与分析

利用 NS2 对改进算法的性能进一步验证，仿真拓扑结构及相关参数取值如图 1 所示。2 个网关之间的链路为瓶颈链路，发送端 *S*1~*S*3 以 Reno 方式分别向接收端 *D*1~*D*3 传输 FTP 流，其中接收端 *D*1 通过 GEO 卫星链路接收源端 *S*1 发送的数据。所有链路均为双向链路，为便于比较，假设各收发端对的 *RTT* 呈比例关系，如 *RTT*1=560 ms, *RTT*2=280 ms, *RTT*3=140 ms。瓶颈链路采用 Drop-tail 算法，网关采用 FIFO 的分组调度算法，网关路由缓存容量设为 50 个分组，δ=2，分组大小设为 1 KB，*ssthresh* 为 64 个分组。仿真运行 100 s。

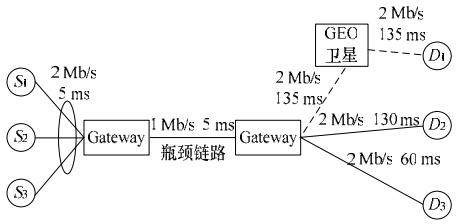

图 1 包含 GEO 卫星链路的网络拓扑结构

图 2 给出了改进慢启动算法 RFB-SS 对 3 个不同 RTT 值的拥塞窗口增长与所需 *RTT* 次数的对比情况，同时引入标准慢启动算法进行比较。由图可见，在慢启动初期，改进算法由于采用了初始窗口为 2 的大窗口机制，因此比标准 TCP 慢启动拥塞窗口达到 *ssthresh*/2 的时间节省了 1 个 *RTT*，加快了慢启动初期启动的速度。当 *cwnd*>*ssthresh*/2 时，达到 *ssthresh*，标准慢启动只用 1 个 *RTT*，而在改进算法中，当 *RTT*=560 ms 时，需要 5 个 *RTT*；当 *RTT*=280 ms 时，需要 8 个 *RTT*；当 *RTT*=140 ms 时，则需要 14 个 *RTT*。由此可看出，改进算法在慢启动后期减缓拥塞窗口增长速度的同时，也削弱了 *RTT* 较小的 TCP 流在竞争带宽时的侵略性，因为从 *ssthresh*/2 到 *ssthresh* 所用的时间增加了，占用带宽的增长速度减缓了。

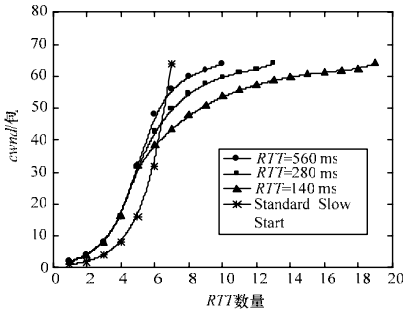

图 2　改进算法拥塞窗口大小与 RTT 次数关系

图 3 和图 4 分别给出了标准慢启动算法和改进慢启动算法中不同 *RTT* 连接的窗口随时间的变化情况，说明改进算法改善了不同 *RTT* 的连接竞争资源时的不公平性。

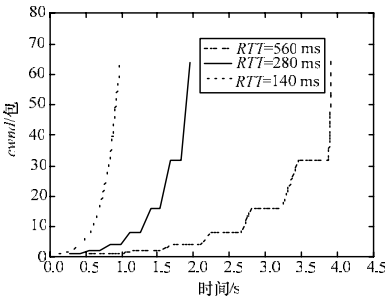

图 3　标准算法不同 RTT 连接的拥塞窗口随时间的变化

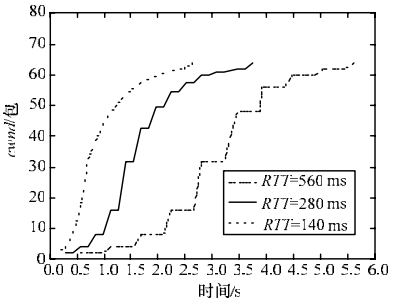

图 4　改进算法不同 RTT 连接的拥塞窗口随时间的变化

由图可见拥塞窗口从 *ssthresh*/2 到 *ssthresh* 所耗用的时间，当 *RTT*=560 ms 时，标准慢启动耗时 0.56 s，改进算法耗时 2.8 s；当 *RTT*= 280 ms 时，标准慢启动耗时 0.28 s，改进算法耗时 2.24 s；当 *RTT*=140 ms 时，标准慢启动耗时 0.14 s，改进算法耗时 1.96 s。即当 *RTT* 分别为 560 ms, 280 ms, 140 ms 时，标准慢启动算法和改进算法的耗时比值分别为 4:2:1 与 1.43:1.14:1，可见改进算法在一定程度上保证了不同 RTT 数据流享用带宽的公平性。

5　结束语

本文针对包含 GEO 卫星链路的网络环境中相同瓶颈链路上存在不同 RTT 流之间竞争带宽的不公平性问题，同时考虑初始拥塞窗口值过小和慢启动后期拥塞窗口增长速度过快问题，对标准 TCP 慢启动算法进行改进。改进算法在连接建立初期，拥塞窗口大小仍保持指数增长，当拥塞窗口增加至 *ssthresh*/2 时，拥塞窗口的增加随 RTT 的不同而不同，RTT 较大的窗口增加较快，反之，窗口增加较慢。性能分析和仿真结果表明，改进算法在慢启动后期不仅减缓了拥塞窗口的增长速度，与慢启动算法的设计初衷相吻合，而且削弱了 RTT 较小的 TCP 流在竞争带宽时的侵略性，缓解了不同 RTT 的 TCP 流竞争资源时的不公平性，在一定程度上保证了不同 RTT 数据流享用带宽的公平性。

参考文献

[1] Wang Haining, Xin Hongiie, Douglas S, et a1. A Simple Refinement of Slow-start of TCP Congestion Control[C]//Proc. of ISCC'00. Los Alamitos, CA, USA: [s. n.], 2000: 98-105.

[2] Floyd S. Limited Slow Start for TCP with Large Congestion Window[S]. RFC 3742, 2004-03.

[3] 刘文远, 冯　波, 龙承念, 等. 一种新的 TCP 拥塞控制慢启动策略[J]. 小型微型计算机系统, 2005, 26(1): 23-25.

[4] Chen Zhigang, Deng Xiaoheng, Zhang Lianming, et al. A New Parameter-config-based Slow-start Mechanism[J]. Journal of Communication and Computer, 2005, 12(5): 56-62.

[5] Pau G, Yamada K. TCP Startup Performance in Large Bandwidth Delay Networks[C]//Proc. of the IEEE INFOCOM'04. Piscataway, USA: [s. n], 2004: 796-805.

(上接第 109 页)

在当前网络环境下，链路带宽非常丰富，而且核心链路带宽一般是冗余的，高处理速率的缓存比较珍贵，尤其是光存储器。因此，需要在丰富的核心网络带宽资源和稀有的光缓存之间作一个平衡，从而推动核心路由器性能大幅度提升。基于仿真和分析的最新成果表明，小缓存(可以用光缓存实现)即可满足性能需求。本文对采用 CIOQ 缓存队列的路由器的仿真显示，小缓存是能够满足需要的，但是对小缓存在实际网络中的适应性还需要做进一步的研究。

参考文献

[1] Villamizar C, Cheng Song. High Performance TCP in ANSNET[J]. ACM Computer Communication Review, 1994, 24(5): 45-60.

[2] Park H, Burmeister E F, Bjorlin S, et al. 40-Gb/s Optical Buffer Design and Simulations[C]//Proc. of the 4th International Conference on Numerical Simulation of Optoelectronic Devices. Santa Barbara, USA: IEEE Press, 2004: 19-20.

[3] Appenzeller G, Keslassy I, McKeown N. Sizing Router Buffers[C]// Proc. of the SIGCOMM'04. New York, USA: ACM Press, 2004: 281-292.

[4] Enachescu M, Ganjali Y, Goel A, et al. Router with Very Small Buffers[J]. ACM/SIGCOMM Computer Communication Review, 2005, 35(3): 83-90.

[5] Raina G, Towsley D, Wischik D. Control Theory for Buffer Sizing[J]. ACM/SIGCOMM Computer Communication Review, 2000, 35(3): 79-82.

[6] Chuang Shang-Tse, Goel A, McKeown N, et al. Matching Output Queuring with a Combined Input Output Queureed Switch[C]// Proceedings of IEEE INFOCOM'99. [S. l.]: IEEE Press, 1999: 1169-1178.

[7] 徐雷鸣, 庞　博, 赵　耀. NS 与网络模拟[M]. 北京: 人民邮电出版社, 2003.