文章编号:1001-2486(2006)05-0068-05

I-TCP:一个高速长距离网络的AIMD算法*

杨 征,吴玲达
(国防科技大学 信息系统与管理学院,湖南 长沙 410073)

摘 要:针对互联网中高速长距离网络上的拥塞控制问题,基于同步AIMD源在Drop-Tail网络环境下竞争带宽的网络模型,提出了一个新的AIMD拥塞控制算法,I-TCP。通过合理地设计高速和低速模式以及模式切换规则,使得该算法适合于配置到高速长距离网络,同时在传统低速网络中也具有很好的性能。仿真试验表明,I-TCP中的AIMD算法在高速长距离网络中表现出了很好的效率、公平性和网络响应性。

关键词:拥塞控制; AIMD; 高速长距离网络
中图分类号:TP393 **文献标识码**:A

I-TCP: An AIMD Algorithm for High-speed Long Distance Networks

YANG Zheng, WU Ling-da
(College of Information System and Management, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: To investigate the problem of the congestion control over high-speed long distance networks, this paper presents a new AIMD congestion control algorithm, I-TCP, which is based on a model of network of synchronous AIMD sources competing for a shared bandwidth under drop-tail queuing. With a reasonable design of a high-speed mode, a low-speed mode and the mode switch rule, I-TCP is suitable for deployment in the high-speed long distance networks as well as the traditional low speed networks. Simulation experiments show that the AIMD algorithm of I-TCP performs better than current TCP implementations in terms of efficiency, fairness and network responsiveness over high-speed long distance networks.

Key words: congestion control; AIMD; high-speed long distance networks

高速长距离网络上的拥塞控制是目前互联网研究的热点问题之一[1-5]。这是因为人们认识到当前的TCP实现在这些网络上性能非常差,由此导致了多个新的拥塞控制算法的出现,如High-Speed TCP[1]、FAST-TCP[2]、Scalable TCP[4]和BIC-TCP[5]。然而,尽管这些算法都表现出比标准TCP算法更好的性能,但是它们中还没有一个有足够的证据表明能够普遍配置到这些高速长距离网络上。

本文基于同步AIMD源在Drop-Tail网络环境下竞争带宽的网络模型[10],设计了一个高速长距离网络中的AIMD算法I-TCP。

1 相关工作

多个相关研究表明,当前的TCP拥塞控制算法在高速和长距离路径上性能非常差[1-5]。在这样的路径上,带宽延时积(BDP, Bandwidth-Delay Product)通常都非常大,因此,在发生拥塞事件后,数据流可能需要花费非常长的时间来恢复其拥塞窗口大小。解决这个问题一个显而易见的方案是设计一个高速TCP拥塞避免模式,其目标是使得拥塞周期尽可能小。

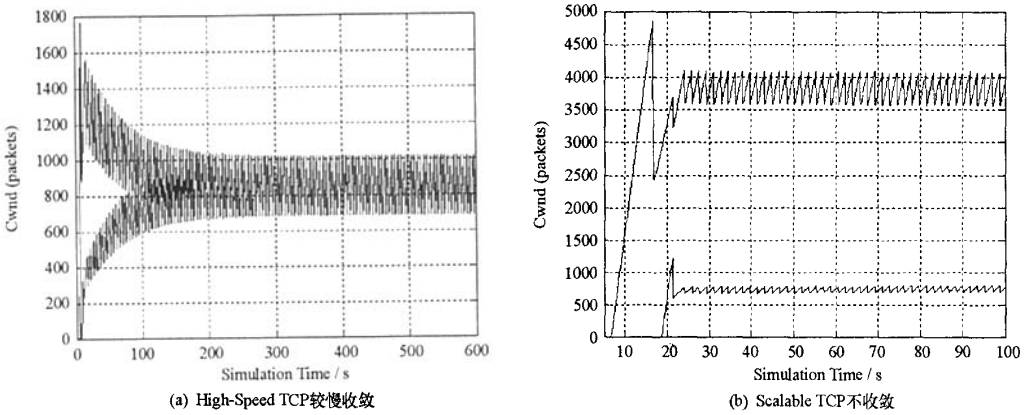
高速模式逻辑设计可以通过多个途径来实现。其中一个方案就是采用当前的拥塞窗口值cwnd作为路径BDP的指示器。也就是说,AIMD的加性增长参数α随着拥塞窗口cwnd的增长而增长,从而导致一个直接随BDP伸缩的加性增长算法。这正是为高BDP网络而设计的High-Speed TCP[1]和Scalable TCP[4]协议所采用的方法。除了调整AIMD增长参数α(cwnd的函数)以外,这两个协议还增大了乘性减

* 收稿日期:2006-03-10
基金项目:国家"863"计划青年基金资助项目(2002AA717019)
作者简介:杨征(1978—),男,博士生。

少因子 β,从而进一步增大了流的侵略性。

然而,基于某个特定流的状态 cwnd 来调整其 AIMD 参数,带来了网络中本质上的不对称性,从而导致意料之外的网络行为。图 1(a)给出了 High-speed TCP 在启动一个新流之后的行为。从图中可以看出,在网络状况改变之后,cwnd 的收敛速度非常地慢。这是因为 cwnd 大的流释放带宽比那些 cwnd 较小的流慢得多(由于有较大的 β),而获取带宽的速度却很快(由于较大的 α)。因此,相对已经建立的流,新启动的流总是处于一个劣势位置。

在 Scalable TCP 中,AIMD 增长参数被设计成与瞬时 cwnd 值成比例。这就使得拥塞周期的持续时间不随 cwnd 变化。图 1(b)给出了这一改变对 Scalable TCP 在启动一个新流之后的影响。从图中可以看出,两个流没有收敛到公平。事实上,Scalable 是一个 MIMD 算法,而 MIMD 算法在 drop-tail 队列下是不可能收敛到公平的[8]。



(a) High-Speed TCP较慢收敛　(b) Scalable TCP不收敛

图 1　两个 TCP 流收敛到公平的速度

Fig.1　Example of two TCP flows illustrating convergence to fairness

显然,设计高速网络 TCP 拥塞控制算法时必须考虑多个目标。也就是说,TCP 拥塞控制算法设计是一个多目标规划问题。特别地,充分地解决下面所有的问题似乎是设计任何适合配置到高 BDP 网络中的 TCP 协议必须满足的最基本的需求[6-7]。

(1)对传统 TCP 的友好性(Friendliness)。在低速环境中,新的 TCP 变种应该有与传统 TCP 流相似的行为。在高速环境中,传统 TCP 流应该不会完全被"饿死"。

(2)效率(Efficiency)。TCP 应该确保有效地利用网络资源。特别地,TCP 流应该尽可能有效地利用可利用的带宽。

(3)公平性(Fairness)。这里所指的公平性是指采用相同 TCP 拥塞控制算法的竞争流之间的公平性。我们希望新的 TCP 变种的不公平性应该不会比传统 TCP 的差(例如,RTT 不公平性应该不会比当前 TCP 算法的差)。

(4)响应性(Responsiveness)。响应性指的是新的 TCP 变种流响应网络状况改变(如启动或停止一个网络流)的速度。

2　算法设计

在前面的工作中[10],我们基于非负矩阵的特性提出了一个多源竞争共享带宽的网络模型,研究发现,满足我们模型假设的网络通常都有一个唯一的、全局指数收敛的稳态点。利用这个模型,分析了端系统采用 AIMD 拥塞控制算法且路由器采用 Drop-Tail 丢包策略的同步网络收敛到稳态点的收敛速度和稳定性。

在文献[10]的基础上,设计了一个高速长距离网络 AIMD 算法,称作 I-TCP,它包括一个高速和一个低速模式。在高、低速模式中,源 i 的增长参数分别为 α_i^H、α_i^L,一旦拥塞,窗口回退到 $\beta w_i(k)-\delta_i$。在低

速模式中，$\delta_i=0$，在高速模式中，$\beta(\alpha_i^H-\alpha_i^L)=\delta_i$，模式切换为：

$$\alpha_i=\begin{cases}\alpha_i^L & cwnd_i-[\beta w_i(k)-\delta_i]\leqslant\Delta^L\\ \alpha_i^H & cwnd_i-[\beta w_i(k)-\delta_i]>\Delta^L\end{cases}\tag{1}$$

其中 $cwnd_i$ 为第 i 个 I-TCP 源的当前拥塞窗口大小，$w_i(k)$表示源 i 在探测到第 k 个拥塞事件之前瞬间的拥塞窗口大小，Δ^L 为模式切换阈值。算法设计时一个重要的考虑就是向后兼容性，也就是说，当配置改进后的协议应用到低速网络上，源端应该能与采用标准 TCP 的源端公平并存，因为标准 TCP 算法中 $\alpha=1,\beta=0.5$，因此，通过选择 $\alpha_i^L=1,\beta=0.5$ 可以满足这一要求。

考虑一个典型的拥塞周期，其中，$t_a(k)$是当 $w_i=\beta w_i(k)$时的时间，$t_b(k)$是模式切换触发的时间，$t_c(k)$是网络发生拥塞的时间，$t_d(k)$为网络源端感知拥塞的时间。因此，在假设源同步的情况下，可以获得一个 I-TCP 的动态性模型。考虑 n 个流，当 $\alpha_i^H=\alpha_i^L$ 时，第 i 个流恢复到了标准 AIMD 算法，窗口大小根据以下公式演变：

$$w_i(k+1)=\beta w_i(k)-\delta_i+\alpha_i^L[t_b(k)-t_a(k)]+\alpha_i^H[t_d(k)-t_b(k)]\tag{2}$$

其中

$$t_b(k)-t_a(k)=\min[\Delta^L,1/(\sum_{i=1}^{n}\alpha_i^L)(1-\beta)w_i(k)-\delta_i]$$

且

$$t_d(k)-t_b(k)=\begin{cases}0 & t_b(k)-t_a(k)\leqslant\Delta^L\\ \dfrac{1}{\sum_{i=1}^{n}\alpha_i^H}\sum_{i=1}^{n}[(1-\beta)w_i(k)-\delta_i-\alpha_i^L\Delta^L] & t_b(k)-t_a(k)>\Delta^L\end{cases}$$

初始条件服从于$\sum_{i=1}^{n}w_i(0)=P+\sum_{i=1}^{n}\alpha_i^H$。

因此，可以得到网络的动态性，描述如下：

$$\boldsymbol{W}(k+1)=A\boldsymbol{W}(k)+\boldsymbol{V}\tag{3}$$

其中 $\boldsymbol{A}=\beta\boldsymbol{I}+(1/\sum_{i=1}^{n}\alpha_i^H)g\boldsymbol{H}^{\mathrm{T}},\boldsymbol{g}^{\mathrm{T}}=[\alpha_1^H,\cdots,\alpha_n^H],\boldsymbol{H}^{\mathrm{T}}=[1-\beta,\cdots,1-\beta]$，$\boldsymbol{V}$ 是一个 n 维向量，它的第 j 个元素由下面的式子给出：

$$V_j=\alpha_j^L\Delta^L-\delta_j-(\alpha_j^H/\sum_{i=1}^{n}\alpha_i^H)\sum_{i=1}^{n}[\delta_i+\alpha_i^L\Delta^L]$$

为方便起见，将方程(3)重写成

$$\boldsymbol{W}(k)=(\boldsymbol{A}-\boldsymbol{x}_p\boldsymbol{y}_p^{\mathrm{T}})\boldsymbol{W}(k)+\boldsymbol{V}+\boldsymbol{W}_{ss}=\bar{A}\boldsymbol{W}(k)+\boldsymbol{V}+\boldsymbol{W}_{ss}$$

其中 $\boldsymbol{x}_p,\boldsymbol{y}_p$ 和 $\boldsymbol{W}_{ss}$按文献[10]中定理3.2和定理3.3所定义。矩阵 $\boldsymbol{A}$ 是一个 Schur 矩阵，因此可以直接应用文献[10]中相关结论得到关于网络收敛和稳定性的结论。

3 仿真试验

3.1 仿真实验配置

为了评价 I-TCP 算法的性能，在网络仿真器 NS2[9](版本为 NS2.26)平台上进行了一系列的仿真试验，对 High-Speed TCP、Scalable TCP 、FAST-TCP、BIC-TCP 和 I-TCP 之间的性能进行了比较和分析。

(1)网络拓扑。试验中采用一个单瓶颈链路的哑铃型拓扑。拓扑参数包括瓶颈链路带宽从 1Mb/s ~250Mb/s 变化，具体选择 1Mb/s，10Mb/s，100Mb/s 和 250Mb/s，链路往返延时(RTT)在 16ms ~ 320ms 变化，具体选择了 16ms，40ms，80ms，160ms 和 320ms。采用 Drop Tail 队列管理策略。

(2)TCP 流设置。分组大小为 1500 比特；最大窗口足够大，不至于成为瓶颈；随机设置各流发送时间以避免相位影响；TCP 流采用修改版的大拥塞窗口有限慢启动算法。发送端和接收端采用的 TCP 代

理为 TCP SACK。采用 FTP 作为通过 TCP 连接传输数据的应用。所有试验的仿真时间均为 600s。

I-TCP 参数:$\alpha^L = 1$, $\alpha^H = 20$, $\beta = 0.5$, $\Delta^L = 19$ 对应窗口大小阈值为 38 个分组。其他算法参数均取其缺省值(这些协议的 NS 实现都可以在 Internet 上下载)。

3.2 仿真结果分析

受篇幅所限,本文只给出了一些有代表性的仿真结果。

(1)效率

图 2 给出了两个 TCP 流的聚合吞吐量随瓶颈队列大小的变化图,两个流有相同链路传播延时(80ms),瓶颈链路带宽为 100Mb/s,队列大小为 BDP 的函数。从图中可以看出,I-TCP 在不同队列大小情况下,都获得了相对较好的聚合吞吐量性能。

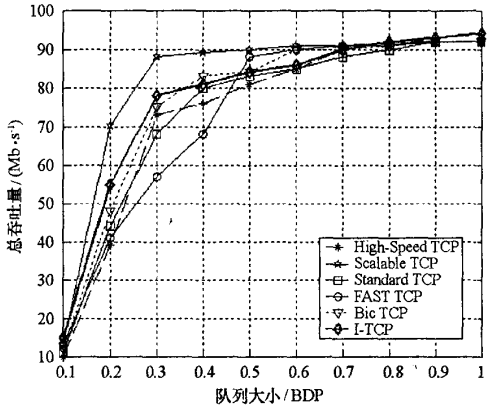


图 2　两个竞争 TCP 流的聚合吞吐量

Fig.2　Aggregate throughput of two competing TCP flows

(2)公平性

图 3 给出了两个共享瓶颈链路 TCP 流的吞吐量之比,两个流有相同的链路传播延时,且传播延时为 16 ~ 320ms。图中给出了瓶颈链路带宽为 10Mb/s 和 250Mb/s 的仿真结果,分别对应低速和高速网络条件。图中给出了队列大小为 20% BDP 的结果,当队列大小为 100%BDP 时获得了类似的结果。

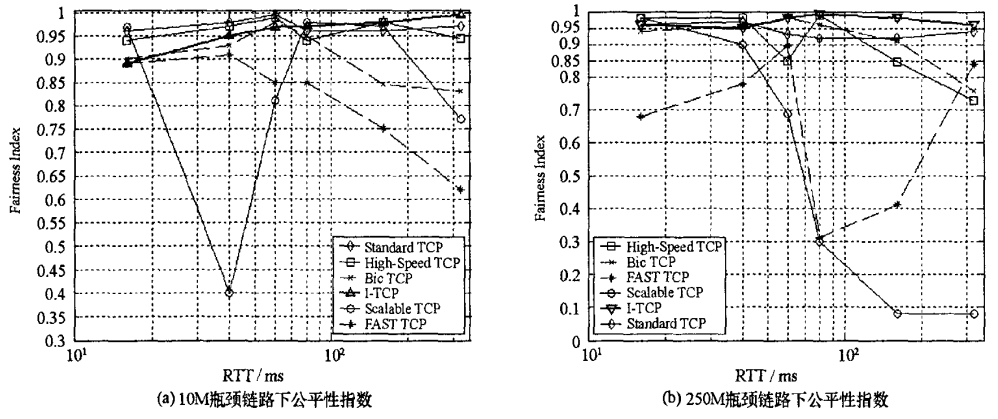


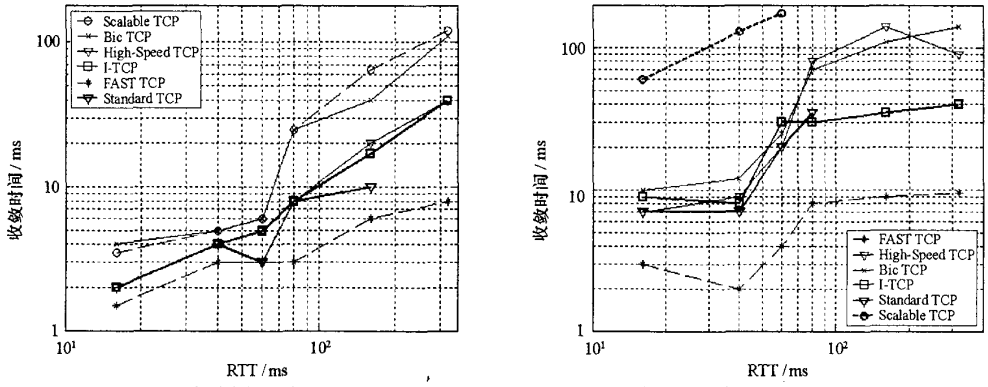
图 3　相同网络条件下两个流吞吐量之比

Fig.3　Ratio of throughput of two flows under same network condition

从图中可以看出,标准 TCP 和 I-TCP 本质上是公平的(竞争流吞吐量差异在 5% 以内),而 Scalable TCP 和 FAST TCP 在低速和高速网络中都非常地不公平,High-Speed TCP 和 Bic TCP 相对要好一些。

(3)响应性

图4给出了测量到的在第二个流启动之后,第一个流的收敛时间。图中给出的值是多次仿真的平均值,每次仿真随机设定第二个流的启动时间。图中给出了随链路传播延时变化的收敛到80%的时间曲线(在本组仿真中,两个流有相同的链路传播延时)。同样地,图中给出了瓶颈链路带宽为10Mb/s和250Mb/s的结果。从图中可以很明显地看出,I-TCP收敛时间相对较快,具有很好的网络响应性。在图中,某些RTT情况无对应的收敛时间,表示在此RTT下已经无法收敛,如右图中的Scalable TCP。



(a) 10Mb/s瓶颈链路下收敛到80%的时间 (b) 250Mb/s瓶颈链路下收敛到80%的时间

图4 在启动一个新流之后两个流收敛到80%的时间

Fig.4 80% convergence time following startup of a second flow

4 结论

基于同步AIMD源在Drop-Tail网络环境下竞争带宽的网络模型,提出了一个高速长距离网络中AIMD算法设计,仿真试验表明我们提出的算法在效率、公平性和网络响应性方面均具有很好的性能。进一步的研究工作包括将同步网络模型拓展到非同步情况,即源端经历异步丢包和不同的RTT等。

参考文献:

[1] Floyd S. HighSpeed TCP for Large Congestion Windows[R]. RFC 3649, Dec. 2003.

[2] Jin C, Wei D X, Low S H. Fast TCP: Motivation, Architecture, Algorithms, Performance[A]. Proc. Ieee Infocom 2004[C], Hong Kong, China, 2004.

[3] Leith D J, Shorten R N. H-TCP Protocol for High-Speed Long-Distance Networks[A]. Proc. 2nd Workshop on Protocols for Fast Long Distance Networks[C]. Argonne, Canada, 2004.

[4] Kelly T. On Engineering a Stable and Scalable TCP Variant[R]. Cambridge University Engineering Department Technical Report, June 2002.

[5] Xu L, Harfoush K, Rhee I. Binary Increase Congestion Control for Fast Long-Distance Networks[A]. Proc. Iee Infocom 2004[C]. HongKong, China, 2004.

[6] Shorten R N, Leith D J, Foy J, et al. Analysis and Design of Congestion Control in Synchronized Communication Networks[J]. Automatica, 2004.

[7] Floyd S. Metrics for the Evaluation of Congestion Control Mechanisms[R]. Draft-irtf-tmrg-metrics-01.txt, October 2005.

[8] Chiu D, Jain R. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks[J]. Computer Networks and ISDN Systems, 17, 1989.

[9] NS2: Network Simulator[CP]. Available at http://www.isi.edu/nsnam/ns/.

[10] 杨征, 吴玲达. 基于非负矩阵理论的同步网络AIMD算法分析[J].数学的实践与认识,2006,36(6):275-278.