小时间尺度网络拥塞研究 *

邵立松, 戴华东, 孔金珠, 张 菁
(国防科学技术大学 计算机学院, 长沙 410073)

摘 要: 基于窗口的拥塞控制算法通常忽略了比往返时间级更小的时间尺度上的网络拥塞,因此不能达到设计的预期效果。重点从理论上研究了窗口更新算法与小时间尺度网络拥塞之间的相互关系,提出了窗口递增算法的小时间尺度突发性度量方法,并分析了小缓冲高速网络环境下传统 TCP 窗口递增算法的效率。

关键词: 小时间尺度; 网络拥塞; 拥塞控制
中图分类号: TP393 **文献标志码**: A **文章编号**: 1001-3695(2012)01-0278-04
doi:10.3969/j.issn.1001-3695.2012.01.077

Network congestion on small scale-time

SHAO Li-song, DAI Hua-dong, KONG Jin-zhu, ZHANG Jing
(*School of Computer*, *National University of Defense Technology*, *Changsha* 410073, *China*)

Abstract: The window-based congestion control algorithms, without considering the network congestion on the smaller time-scale than RTT, cannot reach the designing object. This paper researched the relationship between the window-updating algorithms and network congestion on the small time-scale, proposed a method measuring the connection traffic burst on the small time-scale, and analyzed the TCP efficiency in the high speed network with small buffer.

Key words: small time-scale; network congestion; congestion control

0 引言

端到端拥塞控制的基本设计目标是:如果网络负载很轻或链路利用率不足 100%,应该提高连接的吞吐量;如果网络负载很重以至链路已经发生拥塞,应该降低连接吞吐量。基于窗口的端到端拥塞控制的基本设计目标是:如果拥塞窗口低于效率点,应该增长连接的拥塞窗口值;如果拥塞窗口超过效率点而造成网络拥塞,应该减小连接的拥塞窗口值。设计目标建立在一个重要假设的基础上:拥塞窗口最大值必定超过效率点,或者说,当拥塞发生时连接吞吐量必定不小于有效带宽。

不幸的是,上述假设并不总是成立的[1]:在特定的网络条件下,某些窗口更新算法可能永远无法将拥塞窗口值递增到效率点之上。在理论研究过程中,连接的平均吞吐量常常被近似等于拥塞窗口与往返时间的比值,从而忽略了比往返时间级更小的时间尺度上的网络拥塞。尽管连接的拥塞窗口值小于效率点,小时间尺度网络拥塞仍然会导致拥塞窗口的减小,以至于链路利用率相当低。

为了说明这个问题,本文在单瓶颈链路的网络上对单条传统 TCP 连接进行 NS 仿真实验。传统 TCP 连接的分组长度固定为 1 KB,发送端与瓶颈路由器之间的链路容量为 800 Mbps,连接的往返链路传输总延迟为 24 ms。瓶颈链路的输出缓冲长度为 5 个分组长度,瓶颈链路容量为 15 Mbps。平均吞吐量的采样间隔为 20 ms;拥塞窗口和瓶颈链路队列长度的采样间隔为 1 ms。根据图 1,瓶颈链路分别在 0.2 和 1.6 s 发生两次拥塞现象,这两次拥塞现象显然是有区别的:在 0.2 ms 之前的慢启动阶段采用的是指数递增算法,丢弃的分组数量多,平均吞吐量远远小于有效带宽,拥塞窗口也远远低于效率点;而在 1.9 ms 之前的拥塞避免阶段采用的是线性递增算法,丢弃的分组数量相当少,平均吞吐量到达甚至超过有效带宽,拥塞窗口也超过了效率点。从时间尺度的角度出发,前者是队列长度在小时间尺度上突发的结果;后者是队列长度在大时间尺度上累积的结果。

本文将给出在单瓶颈链路和多级瓶颈链路环境下小时间尺度网络拥塞的理论分析;拥塞窗口超过效率点的一个前提条件是瓶颈链路必须提供足够长的输出缓冲来存储在一个往返时间内的拥塞窗口递增量。讨论了在小时间尺度上的窗口递增算法突发性度量;在往返时间级的拥塞窗口递增量直接度量了窗口更新算法的突发程度,以及对瓶颈路由器缓冲的需求量。讨论了在小缓冲高速网络环境下小时间尺度网络拥塞对传统 TCP 慢启动和拥塞避免两种窗口递增算法效率的影响;小缓冲高速链路很容易发生小时间尺度的网络拥塞,这不仅降低了 TCP 慢启动过程的网络效率,还会进一步损害 TCP 拥塞避免过程的网络效率。

1 单瓶颈链路

先考虑在连接的往返路径上仅存在一条瓶颈链路的情况。假设某条连接发送端输出链路的发送速率为 c_s,瓶颈链路分配给该连接的有效带宽为 c_r,其中 $c_s > c_r$,那么该连接的往返路径上所有其他链路分配的带宽均大于 c_s 和 c_r 的最大值。因此单瓶颈链路的往返链路模型可以简化为图 2 所示的结构。

收稿日期: 2011-05-02; **修回日期**: 2011-06-14 **基金项目**: 核高基重大专项(20102x01037,2010ZX01045-001-002-5,2010ZX01045-001-007-2)

作者简介:邵立松(1977-),男,助理研究员,博士,主要研究方向为计算机体系结构、操作系统和计算机网络;戴华东(1975-),男,副教授,博士,主要研究方向为计算机体系结构、操作系统;孔金珠(1974-),男,副研究员,硕士,主要研究方向为计算机体系结构、操作系统;张菁(1987-),男,博士研究生,主要研究方向为分布式系统、存储系统、云计算(zhangjing110108198@126.com).

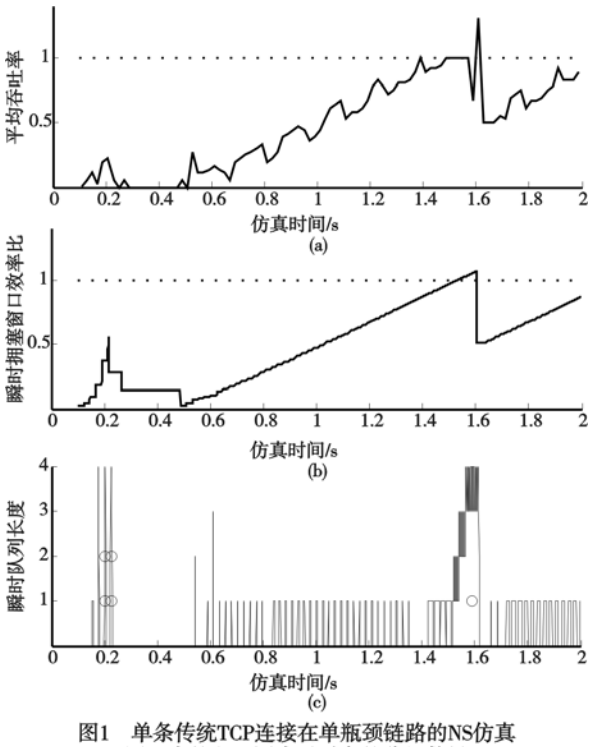

图1　单条传统TCP连接在单瓶颈链路的NS仿真
(图(c)中的空心圆表示丢弃的分组数量)

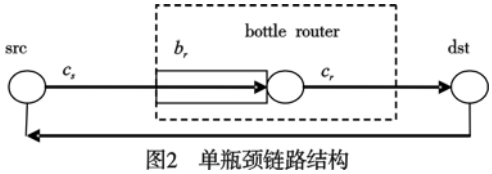

图2　单瓶颈链路结构

定理 1　如果瓶颈链路的缓冲长度相当大,而且连接的拥塞窗口不超过效率点,那么在单瓶颈链路输出缓冲中连接排队等待的最大分组数 $q_{\max}$ 为

$$q_{\max} = \mathrm{MAX}\{A(w_i) + 1, w_1\} \tag{1}$$

其中:$i = 1,2,\cdots,m-1$;$A(w_i)$ 是第 i 个往返时间内拥塞窗口的递增量;w_1 是连接的拥塞窗口初始值;w_m 是效率点 $c_r T$。

证明　不妨设 $x = c_r/c_s$,$0 < x < 1$。在第一个往返时间内,发送端连续发送 w_1 个背靠背的分组。这 w_1 个背靠背分组完全进入瓶颈链路缓冲需要的时间为 w_1/c_s,瓶颈链路发送 w_1 个背靠背分组需要的时间为 w_1/c_r,那么从第一个分组到达缓冲的 w_1/c_s 时间内,连接在缓冲中等待的分组数一直递增,并在第 w_1 个分组到达缓冲时达到最大值。

$$q_{\max}^{(1)} = w_1 \times (1-x) \tag{2}$$

其中:$q(i)_{\max}$ 表示第 i 个往返时间内连接在缓冲排队等待的最大分组数。

如果在第二个往返时间内,连接的拥塞窗口保持不变,即 $w_2 = w_1$。发送端仅仅需要在接收到一个确认分组之后立即发送一个新分组。假设接收端每接收到一个分组就立即返回一个确认分组,那么从第一个确认分组返回到发送端到第 w_1 个确认分组返回到发送端之间的时间间隔为 w_1/c_r,发送端在第二个往返时间内发送分组的速率就降低为 c_r,因此缓冲长度仅仅需要能容纳一个分组就足够了。

如果在第二个往返时间内,连接窗口递增 $A(w_1)$,即 $w_2 = w_1 + A(w_1)$。发送端每接收到一个确认分组之后不仅立即发送一个分组,而且还要将窗口增大 $A(w_1)/w_1$;如果接收到某个确认分组之后,累计增大的窗口值足够一个分组长度,发送端就要发送两个背靠背的分组。在 w_1 个确认分组返回发送端的时间间隔 w_1/c_r 内,发送端发送的分组数为 $w_1 + A(w_1)$。

如果发送链路带宽足够大,以至于在时间间隔 w_1/c_r 内能够完成 $w_1 + A(w_1)$ 个分组的发送任务,那么

$$\frac{w_1 + A(w_1)}{c_s} \leqslant \frac{w_1}{c_r}$$

$$x \leqslant \frac{1}{1 + a(w_1)}$$

其中:$a(w)$ 为分组级拥塞窗口递增函数。根据式(2)的分析结论,排队等待的分组数达到最大值为

$$q_{\max}^{(2)} = (w_1 + A(w_1)) \times \left(1 - \frac{c_r}{(1 + a(w_1))c_r}\right) = A(w_1)$$

如果发送链路带宽比较小,以至于时间间隔 w_1/c_r 内不能够完成 $w_1 + A(w_1)$ 个分组的发送任务,那么 $w_1 + A(w_1)$ 个分组将会以背靠背的形式发送,发送速率为 c_s。又因为 $q_{\max}^{(1)} = w_1 \times (1-x)$,所以排队等待的分组数达到最大值为

$$q_{\max}^{(2)} = w_2 \times (1-x) \tag{3}$$

不等式 $0 < x < 1$ 能够保证确认分组返回发送端的速率为 c_r,而且在拥塞窗口到达效率点之前,往返时间能够维持在连接的往返链路延迟附近 T,因此利用数学归纳法,可以将上面的情况推广到第 i 个往返时间内。又考虑到当 $A(w) = 0$ 的情况,缓冲必须能容纳一个分组,因此

$$q_{\max}^{(i)} \leqslant \begin{cases} A(w_{i-1}) + 1 & x \in (0, \dfrac{1}{1 + a(w_{i-1})}] \\ w_i(1-x) & x \in (\dfrac{1}{1 + a(w_{i-1})}, 1) \end{cases} \tag{4}$$

因为最大排队分组数量函数是关于 x 的非递增函数,所以定理成立。证毕。

一般来说拥塞窗口的初始值都非常小,即使文献[2]建议拥塞窗口的初始值从原来的一个分组长度增长到四个分组长度,拥塞窗口的初始值都可以忽略不计。因此根据上述定理,拥塞窗口到达效率点之前在瓶颈链路缓冲中排队等待的最大分组数取决于往返时间级拥塞窗口递增量。

2　多级瓶颈链路

考虑在连接的往返路径上存在不止一条瓶颈链路的情况。假设某连接的发送端输出链路 j_0 的最大发送速率为 c_0,瓶颈链路 j_n 分配给该连接的有效带宽为 c_n,那么瓶颈链路 j_{n-1} 与 j_n 分配给连接的有效带宽满足 $c_{n-1} > c_n$;瓶颈链路 j_{n-1} 和 j_n 之间的任意非瓶颈链路 j 分配给连接的有效带宽一定满足 $c_j \geqslant c_{n-1}$。假设连接的往返路径上存在 n 条瓶颈链路,它们分配给连接的有效带宽满足 $c_0 > c_1 > \cdots > c_n$,根据前一章的分析,确认分组返回到发送端的速率为 c_n,发送端输出链路的最大发送速率为 c_0 与 $(1+a)c_n$ 之间的最小值,其中 a 为拥塞窗口分组级递增量;进一步假设 $c_0 > (1+a)c_n$,且连接的往返路径上的瓶颈链路 j_i 和 j_{i+1} 满足 $c_i > (1+a)c_n > c_{i+1}$,那么连接的往返路径上的瓶颈链路数量就减少为 r 条,即链路 $j_{i+1}, j_{i+2}, \cdots, j_n$。因此连接的往返链路模型可以简化为图 3 所示的结构。

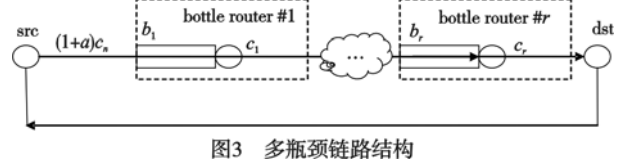

图3　多瓶颈链路结构

定理 2　如果瓶颈链路的缓冲长度相当大,而且连接的拥

塞窗口不超过效率点,那么在多瓶颈链路的输出缓冲中连接排队等待的最大分组数 $q_{\max}$ 的平均值为

$$\bar{q}_{\max}=\frac{\mathrm{MAX}(A(w_i))}{n} \tag{5}$$

其中:$i=1,2,\cdots,m-1$;w_1 是连接的拥塞窗口初始值;$w_m=c_rT$。

证明 根据前面的结论,在第 i 个往返时间内,当发送端向瓶颈链路 1 发送 w_i 背靠背个分组时,瓶颈链路 1 也向瓶颈链路 2 发送 w_i 背靠背个分组,而且在输出缓冲中排队等待的最大分组数为

$$q_1^{(i)}=w_i\left(1-\frac{c_1}{(1+a(w_{i-1}))\times c_n}\right)$$

其中:$q_j(i)$ 表示第 i 个往返时间内瓶颈链路 j 的输出缓冲中排队等待的最大分组数。那么在瓶颈链路 j 的输出缓冲中排队等待的最大分组数为

$$q_j^{(i)}=w_i\left(1-\frac{c_j}{c_{j-1}}\right) \tag{6}$$

那么就有 $\prod_{j=1}^{j=n}\left(1-\frac{q_j^{(i)}}{w_i}\right)=\frac{1}{1+a(w_{i-1})}$。

又因为

$$\prod_{j=1}^{j=n}\left(1-\frac{q_j^{(i)}}{w_i}\right)^{q_j^{(i)}<<w_i}\approx 1-\frac{1}{w_i}\sum_{j=1}^{j=n}q_j^{(i)}$$

所以第 i 个往返时间内,瓶颈链路中排队等待的最大分组数的平均值为

$$\bar{q}^{(i)}=\frac{w_i}{n}\times\frac{a(w_{i-1})}{1+a(w_{i-1})}=\frac{w_{i-1}\times a(w_{i-1})}{n}=\frac{A(w_{i-1})}{n}$$

在拥塞窗口到达效率点之前,一个窗口内的分组可以在一个往返链路延迟 T 内返回到发送端,而且返回到发送端的瞬时速率为 c_n。定理成立。证毕。

文献[3]建议,在慢启动阶段,每接收到一个确认分组,拥塞窗口最多递增一个分组长度。即使输出链路容量有冗余,发送端发送分组的速率不会超过 c_n 的 2 倍。因此在连接的往返链路上,瓶颈链路分配给该连接容量必须满足 $2c_n<c_1<c_2<\cdots<c_n$。

定理 3 如果链路分配给连接的有效带宽服从均匀分布:

$$f(x)=\frac{1}{c_{\max}}\quad 0<x<c_{\max} \tag{7}$$

那么往返路径上存在 n 条链路都成为瓶颈链路的概率不会超过 0.5^{n-1}。

证明 当 $n=1$ 时,往返路径上总存在某个最小的有效带宽链路,因此定理成立。

当 $n=2$ 时,假设往返路径上最小的有效带宽为 c_2,其中,$0<c_2<c_{\max}$,那么在该链路之前必定存在一条瓶颈链路,其有效带宽 c_1 满足 $c_2<c_1<\min(c_{\max},2c_2)$。如果 $2c_2<c_{\max}$,那么 c_1 成为瓶颈带宽的概率为

$$P(c_2<c_1<2c_2)=\frac{1}{2}\times\frac{2c_2-c_2}{c_{\max}}$$

如果 $2c_2>c_{\max}$,那么 c_1 成为瓶颈带宽的概率为

$$P(c_2<c_1<c_{\max})=\frac{1}{2}\times\frac{c_{\max}-c_2}{c_{\max}}$$

因此 c_1 成为瓶颈带宽的概率为 0.5。定理成立。

假设 $n=k$,定理成立。当 $n=k+1$ 时,往返路径上最小的有效带宽为 c_{k+1},其中 $0<c_{k+1}<c_{\max}$。在该链路之前的瓶颈链路有效带宽分别为 $c_2,\cdots,c_k$。根据 $n=2$ 的情况,有效带宽为 $c_2,\cdots,c_k$ 的链路成为瓶颈链路的概率为 0.5^k,那么 $c_1,c_2,\cdots,c_k$ 同时成为瓶颈链路的概率为

$$P\{n=k+1\}=P\{n=k\}\times\frac{1}{2}\left(\frac{2c_{k+1}-c_2}{c_{\max}}+\frac{c_{\max}-c_2}{c_{\max}}\right)\leqslant\left(\frac{1}{2}\right)^{k+1}\left(1-\frac{c_2-c_{k+1}}{c_{\max}}\right)$$

又因为 $c_2>c_{k+1}$,所以定理成立。因此对于任意 $n>0$ 的正整数,定理都成立。证毕。

定理 3 说明了往返路径上的瓶颈链路数量 n 一般都很小。另外,大量的研究表明[4~8],互联网具有无尺度网络结构:大部分网络节点具有很少的邻接节点,而少量网络节点却拥有大量的邻接节点。互联网具有“小世界”特性:任何两个网络节点可以通过少量的其他节点联系起来。因此在互联网中,拥塞窗口超过效率点的一个条件是瓶颈链路必须提供足够长的输出缓冲来存储在一个往返时间内的拥塞窗口递增量。

3 递增算法的突发性分析

首先讨论限制背靠背发送分组数和主动队列管理算法对瓶颈路由器最大队列长度的影响。不妨假设最大背靠背发送的分组数为 m,背靠背发送的时间间隔为 τ,其中 $m=1,2,\cdots$,且 $\tau>0$。那么在第一个往返时间内,发送端发送 w_1 个分组需要的时间近似为 $w_1(1/c_s+\tau/m)$,发送 w_1 个分组的发送速率就减小为 $c_s/(1+c_s\times\tau/m)$。如果该发送速率仍然大于瓶颈链路分配的有效带宽,即 $\tau/m<1/c_r-1/c_s$,那么参数 $x=c_r/c_s+\tau/m$ 满足 $0<x<1$,所以定理仍然成立;否则,发送端的输出链路就成为新的瓶颈链路,而在网络缓冲中排队等待的分组数为 0。因此在发送端输出链路容量 c_s 相当大的条件下,即使限制最大背靠背发送分组数,定理 1 的结论仍然成立的一个充分条件是 $\tau\times c_r<1$。

参数 τ 的数量级由主机的指令周期决定。当前主机的指令周期不会超过 ns,那么参数 τ 的数量级为 10^{-9}。当分组长度为 64 B~1.5 KB 时,只要瓶颈链路容量 $c_r<64$ Gbps,定理 1 仍然成立。因此限制背靠背发送分组数对排队等待的最大分组数几乎没有什么影响。仿真实验也验证这个结论。

现在分析主动队列管理算法对瓶颈路由器最大队列长度的影响。排队等待的最大分组数量是队列长度在瞬间达到的最大值,这反映了小时间尺度上队列长度的变化规律;而主动队列管理算法使用平均队列长度,这反映了大时间尺度上队列长度的变化规律,平均队列长度将小时间尺度上出现的瞬态最大队列长度过滤掉了。小时间尺度上的最大队列长度造成缓冲溢出的直接后果是瓶颈链路的网络效率会进一步降低;因此主动队列管理算法不可能检测到最大队列长度,更不可能预防或处理小时间尺度上的缓冲溢出。

总之,小时间尺度上路由器的最大队列长度直接取决于拥塞窗口在往返时间级的递增量。一方面,拥塞窗口在往返时间级的递增量直接反映了相应的窗口更新算法的突发程度;另一方面,拥塞窗口在往返时间级的递增量也直接反映了相应的窗口更新算法对路由器缓冲的需求量。

传统 TCP 协议的发送机制具有小时间尺度上的突发性:只要发送窗口足够组成一个完整的分组,TCP 连接的发送端就立即发送一个分组。这是造成小时间尺度上网络拥塞的根本原因。往返时间级拥塞窗口递增量直接决定了小时间尺度上

的突发程度:递增量越大,突发程度越高,小时间尺度的网络拥塞越严重;递增量越小,突发程度越低,小时间尺度的网络拥塞越轻微。图 4 给出了拥塞窗口从 5 KP 递增到 10 KP 的过程中各种递增算法在往返时间级递增量。TCP 慢启动、HSTCP[9] 和 STCP[10] 算法都属于凸递增算法[11],在往返时间级拥塞窗口的递增量是当前拥塞窗口的增函数;TCP 拥塞避免算法属于线性递增算法,在往返时间级拥塞窗口的递增量与当前用拥塞窗口无关。有趣的是,在上述所有算法中,TCP 慢启动算法的突发程度和对路由器缓冲的需求量最大,而 TCP 拥塞避免算法的突发程度和对路由器缓冲的需求量最小。

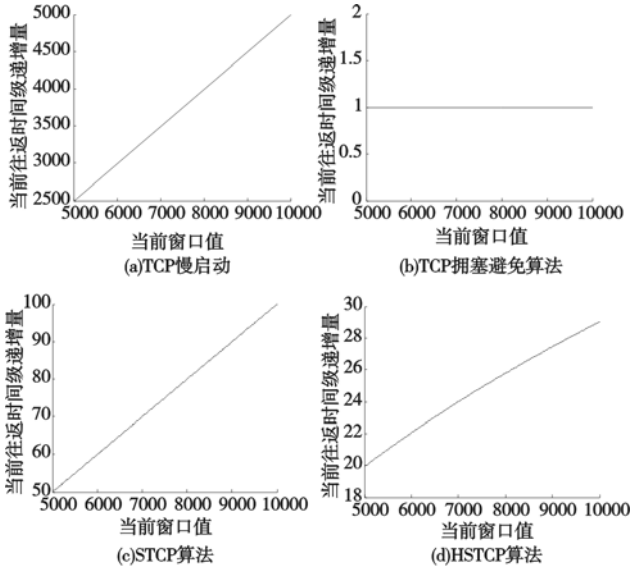

图4　各种窗口递增算法的往返时间级递增量

4　小缓冲高速网络环境

互联网路由器缓冲设计的一个目标就是保证链路利用率为 100%,即输出缓冲永远不会为空闲。当前路由器厂商在设计缓冲时遵循 rule-of-thumb 原则[12]:每条链路需要的输出缓冲长度为链路容量与往返延迟积 c,T,文献[13]建议往返延迟参数采用 0.25 s[13,14]。该原则的设计目标是,即使在单 TCP 连接的情况下输出缓冲也不会空闲,以至于网络效率仍然能够达到 100%。根据该原则,在链路带宽为 40 Gbps 的主干网中,路由器应该采用长度为 10 Gb 的缓冲。大缓冲设计方案不仅带来硬件设计的问题,而且也导致了相当长的排队延迟和剧烈的延迟抖动;更重要的是,由于 TCP 的超时重传机制,过大的缓冲也是互联网的一个不稳定因素。

最近研究发现,在大量 TCP 连接统计复用瓶颈链路情况下,主干网路由器采用相当小的缓冲长度也可以保证获得 100% 链路利用率[15]。文献[16]认为,如果所有 TCP 连接都是同步时,高利用率就意味着大缓冲;但当它们都是异步统计复用的时候,小缓冲设计就成为可能。小缓冲设计方案是基于硬件考虑:a)路由器缓冲的访问速率必须能够匹配链路速率,因此主干网路由器的设计相当复杂;b)交换速率每 18 个月翻一倍,缓冲访问速率每 10 年翻一倍,因此缓冲越来越成为设计高速路由器的瓶颈;c)全光路由器可能仅仅缓存 100 个分组左右,如果 100 个分组大小的缓冲能够获得足够好的性能,那么全光分组交换的核心网络就能够实现。虽然小缓冲设计方案能够在统计复用的基础上保证链路利用率,但小缓冲可能导致相当高的分组丢弃概率(5% ~15%),这么高的分组丢弃概率不仅使得网络应用难以接受,而且会导致频繁的 TCP 重传和超时[17]。

小缓冲高速链路很容易发生小时间尺度的网络拥塞,这不仅降低慢启动过程的 TCP 连接的网络效率,还会进一步损害拥塞避免过程的 TCP 连接的网络效率。图 5 描述了 TCP 连接在不同缓冲长度条件下的吞吐量变化曲线。当缓冲比较小时,不仅在慢启动过程而且在拥塞避免过程,TCP 连接的吞吐量都远远低于有效带宽;当缓冲比较大时,无论是在慢启动过程还是在拥塞避免过程,TCP 连接的吞吐量都有所增大。当缓冲长度为 30 和 300 分组时,慢启动阈值对应的吞吐量几乎为零;增大缓冲长度,慢启动算法能够快速地增大吞吐量,在小时间尺度上甚至超过有效带宽,即使导致大量的分组丢弃并引发超时重传,慢启动阈值仍然相当大;当缓冲增大到 3 000 个分组长度时,慢启动阈值对应的吞吐量为 150 Mbps;当缓冲增大到 10 KP 时,慢启动阈值对应的吞吐量就近似等于有效带宽了。

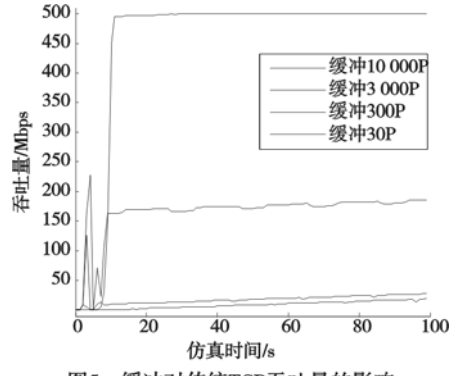

图5　缓冲对传统TCP吞吐量的影响
(有效带宽为500 Mbps，往返延迟为200 ms)

5　结束语

传统 TCP 拥塞控制算法在当前互联网上已经获得显著成功;但是在大窗口高速网络条件下,相当低的网络效率是传统 TCP 面临的最大问题。为了提高大窗口条件下的网络效率,已经提出了许多新的拥塞控制算法,如 HSTCP、STCP 和 ACC 等。然而这些算法还存在两个问题:它们都属于凸性递增算法,与线性和凹性递增算法相比较,它们会导致更低的网络效率和更高的分组丢弃概率;它们都忽略了小时间尺度上的网络拥塞现象。因此在小缓冲高速网络环境中,这些凸性递增的窗口算法并不能达到预期效果。

参考文献:

[1] SHAO Li-song, ZHANG He-ying, DOU Wen-hua. General window-based congestion control: buffer occupancy, network efficiency and packet loss[C]//Proc of the 4th Annual Communication Networks and Services Research Conference. Washington DC: IEEE Computer Society, 2006: 181-195.

[2] ALLMAN M. RFC 2914, Increasing TCP's initial window[S]. 1998.

[3] ALLMAN M, PAXSON V, STEVENS W. RFC2581, TCP congestion control[S]. 1999.

[4] ADAMIC L. The small world Web[C]//Proc of the 3rd European Conference on Research and Advanced Technology for Digital Libraries. London: Spring-Verlag, 1999: 443-452.

[5] ADAMIC L A, HUBERMAN B A. Technical comments: power-law distribution of the world-wide Web[J]. Science, 2000, 287(5461): 2115.

(下转第 285 页)

清新,且消除了相位旋转。

3.2　时变信道仿真

为了构造一条时变信道,在此采用将两条信道进行组合的方法来模拟一条时变信道。在迭代起始阶段,将其通过信道 c_1,信道参数为

$$c_1 = [0.3132 \quad -0.1040 \quad 0.8908 \quad 0.3134]$$

当迭代到 10 000 次时,信道发生突变,信道改为 c_2, 信道参数变为

$$c_2 = [1 \quad 0 \quad 0.498 + i\,0.0435 \quad 0 \quad 0.2955 + i\,0.0522]$$

其他仿真参数设置如表 3 所示。200 次蒙特卡诺仿真结果如图 6 所示。

表 3　仿真参数

算法	仿真步长		初始化权值	
	前向滤波器	反馈滤波器	前向滤波器	反馈滤波器
CMA-DFE	0.000 05	0.000 05	第 1 个抽头初始化值为 1	第 16 个抽头初始化值为 1
MWT-CMA-DFE	0.000 1	0.000 1		
FMWT-CMA-DFE	0.000 1	0.000 4		
DMWT-CMA-DFE	0.000 7	0.000 8		

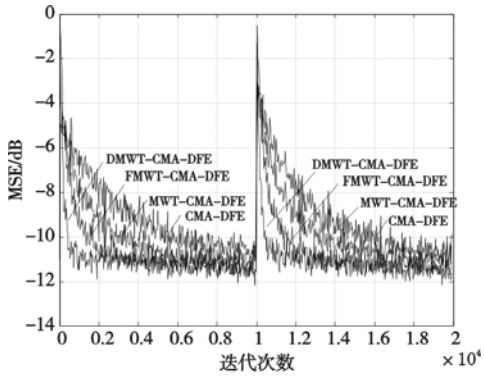

图6　均方误差曲线

由图 6 可知,当信道发生突变时,由于信道的突变,MSE 在迭代到 10 000 次时突然变大,并使均衡器重新收敛,这表明算法对信道的突变能力能够及时跟踪。而且与 FMWT-CMA-DFE、MWT-CMA-DFE 和 CMA-DFE 相比,DMWT-CMA-DFE 具有更快的跟踪时变信道的能力。

4　结束语

现有的基于正交小波变换自适应均衡器在设计时均是将正交小波变换放置在均衡器(前向滤波器)之前,以起到加快算法收敛的目的。针对此问题,本文以常模判决反馈均衡器为例,根据小波变换位置不同,提出了三个正交多小波变换常模判决反馈盲均衡器,即 MWT-CMA-DFE、FMWT-CMA-DFE 和 DMWT-CAM-DFE,分析了各种均衡器的计算复杂度。最后,用水声信道对三个均衡器性能进行了仿真,并从收敛速度、眼图和跟踪时变信道的能力三个方面对各种均衡器的性能做了比较,结果表明:与 CMA-DFE、MWT-CMA-DFE 和 FMWT-CMA-DFE 相比,DMWT-CAM-DFE 具有更快的收敛速度和跟踪时变信道的能力,且均衡后的眼图没有相位旋转。

参考文献:

[1] HOSUR S, TEWFIK A H. Wavelet transform domain LMS algorithm [C]//Proc of IEEE International Conference on Acoustics, Speech and Signal Processing. 1993:508-511.

[2] 黄奎,吕锐. 基于小波包变换的自适应均衡算法[J]. 电子学报,2003,31(8):1205-1208.

[3] 曾召华,刘贵忠,马社祥. 基于正交小波变换的瞬变步长 LMS 自适应滤波算法[J]. 通信学报,2001,22(4):123-128.

[4] 杨超,郭业才. 基于正交小波变换的判决引导联合盲均衡算法[J]. 兵工学报,2010,31(2):199-203.

[5] 韩迎鸽,郭业才,李保坤,等. 引入动量项的正交小波变换盲均衡算法[J]. 系统仿真学报,2008,20(6):1559-1562.

[6] 刘振兴,郭业才,高敏,等. 多小波模糊神经网络盲均衡算法[J]. 兵工学报,2010,31(9):1137-1144.

[7] 郭业才,刘振兴. 基于平衡正交多小波变换的盲均衡算法[J]. 兵工学报,2010,31(3):199-204.

[8] ERDOL N, BASBUG F. Wavelet transform based adaptive filters: analysis and new results [J]. IEEE Trans on Signal Processing, 1996,44(9):2163-2171.

[9] GODARD D. Self-recovering equalization and carrier tracking in two dimensional data communication systems[J]. IEEE Trans on Communication,1980,28(11):1867-1875.

(上接第 281 页)

[6] BU Tian, TOWSLEY D. On distinguishing between Internet power law topology generators[C]//Proc of the 21st Annual Joint Conference of IEEE Computer and Communications Societies. 2002:638-647.

[7] WILLINGER W, GOVINDAN R, JAMIN S, *et al*. Scaling phenomena in the Internet: critically examining criticality[J]. Proceedings of the National Academy of Sciences of the United States of America,2002,99(1):2573-2580.

[8] JIN Shu-dong, BESTAVROS A. Small-world Internet topologies, Technical Report BUCS-TR-2002-004[R]. 2002.

[9] FLOYD S. RFC 3649, Highspeed TCP for large congestion windows [S]. 2003.

[10] KELLY T. Scalable TCP: improving performance in highspeed wide area networks[J]. ACM SIGCOMM Computer Communication Review,2003,33(2):83-91.

[11] 邵立松,张鹤颖,窦文华. 基于窗口的端到端拥塞控制:网络稳定性与效率[J]. 计算机学报,2006,29(3):353-360.

[12] VILLAMIZAR C, SONG Cheng. High performance TCP in ansnet [J]. ACM SIGCOMM Computer Communications Review, 1994,24(5):45-60.

[13] BUSH R, MEYER D. RFC3439, Some Internet architectural guidelines and philosophy[S]. 2002.

[14] Cisco line cards[R/OL]. http://www.sisco.com/en/US/products/hw/modules/ ps2710/ products_data_sheets_list.html.

[15] APPENZELLER G, KESLASSY I, McKEOWN N. Sizing router buffers[J]. ACM SIGCOMM Computer Communication Review, 2004,34(4):281-292.

[16] WISCHIK D, McKEOWN N. Part I: buffer sizes for core routers[J]. ACM SIGCOMM Computer Communication Review, 2005, 35(3):75-78.

[17] DHAMDHERE A, DOVROLIS C. Open issues in router buffer sizing [J]. ACM SIGCOMM Computer Communication Review,2006, 36(1):87-92.