文章编号:1000 - 582X(2006)11 - 0092 - 04

TCP 带宽估计算法*

汪纪锋,胡 晗,王春辉
(重庆邮电大学 通信学院,重庆 400065)

摘 要:在有线网络或者误比特率较低的环境中,分组丢失往往是由于网络拥塞造成的,因此传输控制协议(TCP)能够良好运行;然而当 TCP 运行在高误码环境中,并且在遭遇误码丢包时,TCP 拥塞窗口依旧盲目减半,没有能够充分利用可用带宽,从而导致其性能大幅度下降. 近年来,提出了不少对 TCP 拥塞控制机中带宽估计算法进行改进的方案,以用来改进 TCP 在随机丢包链路中的性能. 文中首先分析了在 TCP 连接中发送端实现的带宽估计算法所面临的问题,然后重点分析了几种带宽估计算法的准确性及其性能,同时讨论了带宽估计算法的准确性对协议的公平性产生的影响.

关键词:拥塞控制,随机丢包,带宽估计
中图分类号:TN915 **文献标识码**:A

传输控制协议(TCP)已经被证明在传统的有线网络中能够高效地运行,即使在原先并不是为其设计的高速有线网络及其它的拓扑结构中也能够正常运行. 然而随着无线通信技术的发展,在诸如蜂窝网、无线局域网以及新的移动计算环境下,TCP 拥塞控制机制却面临着新的挑战.

在高误码的环境中,现有的 TCP 版本(如:Reno 和 NewReno)的使用都会使得网络吞吐量急剧下降. 造成这种现象的主要原因是 TCP 不能够区分数据包的丢失是误码丢包,还是网络拥塞丢包. 因此 TCP 拥塞控制机制会不必要地降低传输速率[1-2].

TCP 保持了 2 个状态变量来控制传输速率:拥塞窗口(cwnd)和慢启动门限(ssthresh). 后者用来把慢启动和拥塞避免 2 个阶段区分开来. 在建立连接的最初阶段,源端采用指数增加 cwnd 的值的方法直到出现数据包丢失. 当出现这种现象的时候,TCP - Reno 将 ssthresh 数值减半,而当 cwnd 达到新的 ssthresh 时,TCP 进入拥塞避免阶段,随后 cwnd 以线性的方式增加.

这种机制是用 ssthresh 对可用带宽进行估计,并利用拥塞避免的方法来探测额外的带宽. 然而这种方式只有当发送速率达到了可用速率,并且此时出现第一个数据包丢失的时候,估计出来的带宽才是准确的. 如果丢包是因为误码或者传输错误而引起的,那么 ssthresh 就会错误的被设定为一个较小的值,进而又会使得发送速率和网络吞吐量下降.

为了避免这种现象,文献[3]提出了几种机制并进行了分类. 在端到端的协议中,通过发送端恢复丢失的包;分割连接的协议是把端和端的连接在基站处分割成两部分,而链路层的协议是基于把 ARQ 和 FEC 结合起来的,这些均不同程度地提升了 TCP 的性能,然而它们或者失去了端对端的语义,或者存在公平性方面的问题,或者链路层的重传与 TCP 层的重传存在冲突,这些问题都有待于进一步研究. 而另外一种方法,即如果在 TCP 拥塞控制机制中使用更加精确的带宽估计技术,同样能够提升 TCP 的性能. 一个可能的方案就是把源端的带宽估计算法与具有 ECN(Explicit Congestion Notification)功能的中间路由器节点结合起来,这样就有可能把网络拥塞丢包和非拥塞丢包区分开来,从而提高 TCP 在无线环境中的性能.

文中首先讨论可能影响 TCP 带宽估计准确性的原因,同时还指出这种不准确性对 TCP 可调节参数的精度产生的影响;然后对几种估计算法进行分析.

* 收稿日期:2006 - 06 - 17
基金项目:重庆市科技攻关资助项目(CSTC,2004BB2165)
作者简介:汪纪锋(1944 -),男,湖北武汉人,重庆邮电大学教授,博士生导师,主要从事通信网及其智能化、控制理论及应用研究.

1　带宽估计

1.1　对带宽估计产生影响的原因

对于一个连接，如果获得的可用信息越多，则估计的结果也就越准确，从而能够更加有效和公平地利用网络资源，这是所有带宽估计技术必须要遵循的原则.

在文献[4-5]中提出了一种通过在发送端计算接收到的 ACK 之间的时间间隔来估计带宽的方法，即用已经得到确认的 2 个连续 ACK 之间传输的字节数除以 2 个连续 ACK 之间的时间间隔来得到一个带宽样本，同时还使用滤波技术处理样本序列以用来消除估计带宽值的快速变化和减小随机丢包对带宽估计造成的影响. 在文献[6]中指出：

$$\text{ssthresh} = Bwe * RTT_{\min} \quad (1)$$

这里 $RTT_{\min}$ 是 TCP 连接所探测到的往返时间最小值，这个值可以被看作是网络没有发生拥塞时对 RTT 的一个估计.

由于传输分组时的特殊定时方式，以及 TCP 源端测量时间间隔和 $RTT_{\min}$ 的不准确，一些问题也就随之而来. 下面是产生部分问题的一些原因.

1) ACK 压缩

当在反向路径上路由器发生拥塞而改变接收到 ACKs 的时间间隔时，就会产生 ACK 压缩的现象. 事实上，一簇分组到达目的地时，一簇 ACKs 也会同时产生. 如果这些 ACKs 碰到发生拥塞的节点，在转发它们的时候就会使它们的间隔时间缩短，这就造成了 ACK 压缩. 这种现象同时造成过高地估计正在使用的带宽. 而产生的误差依赖于 TCP 分组的长度与 ACK 分组之间的长度的比值. 在典型的环境中，ACK 分组的长度是 40 字节，数据分组则是 1 500 字节，那么对可用带宽的估计就会比真实值高出近 37.5 倍. 绝大多数的 TCP 实现采用了延迟 ACK 技术，带宽估计值相当于 2 倍的真实值，因而网络运行时的 ACK 压缩是不能够忽略的[7].

2) TCP 粗粒度时钟

TCP 必须把带宽估计值转换成在拥塞控制机制中使用的参数. 在文献[8]里可以看到，当 TCP 的发送速率等于可用带宽时，优化后的慢启动门限 ssthresh 等于“管道”中传输的分组，例如，传输窗口等于带宽时延积，如公式(1)所示. 然而，TCP 是用粗粒度时钟测量 RTT 的，因此 RTTmin 的精度依赖于 TCP 的定时时钟粒度 G. 例如，在一个传播迟延为 G/10 的局域网中，如果 RTTmin 的值等于 G，则测量出的 RTT 会比真实值大 10 倍. 因而即使带宽估计值是正确的，ssthresh 也会被设置得比正确值高 10 倍，从而使连接表现出更为激进地争夺带宽的行为.

3) 路由变更

在连接期间路径发生改变，此时主机并不能意识到，如果新的路径具有更短的传播迟延，$RTT_{\min}$ 可以正确地更新；但是如果新的路径的传播迟延变得更长时，连接就不能够分辨出迟延的增加究竟是因为拥塞造成的还是路径发生变化造成的.

1.2　带宽估计算法

一些文献为 TCP 拥塞控制提出了带宽估计算法. 它们各有各的特点，同时也面临上述问题. 应该指出的是，仅在发送端测量出的带宽只是 TCP 连接的使用带宽，而不是可用带宽. 根据文献[6]，可用带宽指的是 TCP 连接在能够正确进行拥塞控制的基础上的最大的、最理想的传输速率. 而使用带宽是指源端实际发送数据的速率.

1) 包对算法(Packet Pair Algorithm)[5]

在开始建立连接时使用 Packet Pair 算法，其目的是为 ssthresh 设置一个初始值以减少由于过高的默认值而导致多包丢失现象的发生[8]. 尽管 ssthresh 与可用带宽等价，但是通过对接收到的 ACKs 之间的时间变化分析，可以很容易获得对瓶颈带宽的估计. Packet Pair 算法原理：如果以背靠背(back - to - back)方式发送 2 个分组，则它们到达接收端时的时间间隔直接反映了瓶颈带宽；同样，如果反向路径没有发生拥塞，则在发送端应该以同样的时间间隔收到对应的 ACKs. 这样，源端就可以用发送的分组长度除以对应 ACKs 的时间间隔来估计可用带宽.

2) TC PVegas 算法(TC PVegas Estimation Algorithm)[9]

这个机制是通过计算期望速率 cwnd/$RTT_{\min}$ 和实际速率 cwnd/RTT 之间的差来估计可用带宽. 通过对网络没有发生拥塞时的观测来调整拥塞窗口的大小，使得实际发送速率接近于期望速率；而当网络发生拥塞的时候，实际速率低于期望速率. 当接收到一个 ACK 时计算如下：

$$\text{diff} = (expected\_Rate - actual\_Rate) * RTT_{\min}$$

当 diff < 1 时，cwnd + 1；

当 diff > 3 时，cwnd - 1；

当 $1 \leqslant \text{diff} \leqslant 3$ 时，cwnd 不变.

尽管 TC PVegas 算法可以使 cwnd 达到一个平衡点[10]，但是在有其他竞争流的情况下却无法满足公平性要求. 另外，为了估算连接路径上的传播时延，TC

PVegas 需要检测 RTT 的最小值,而这恰恰会遇到指出的因路由改变或持续拥塞引起的问题.

3) TC PWestwood 算法(TC PWestwood Estimation Algorithms)

文献[6,10,11]都提到通过测量 ACK 的速率来估计带宽. 这种方法是在发生拥塞之后(例如接收到 DUPACKs 或重传定时器超时)用于设置 ssthresh 和 cwnd 的,以避免象 TC PReno 一样在发生丢包时盲目将发送速率减半,因此 TC PWestwood 在面临随机丢包的情况下可以达到较高的链路利用率. 在文献[11]中,TC PWestwood 算法考虑了通过 ACKs 的到达而得到的带宽样本序列 sample_BWE[k]和通过低通滤波技术获得的 BWE[k],算法伪码如下:

Algorithm WESTWOOD1:

if(ACK is received)

sample_BWE[k] = (acked * pkt_size * 8) / (now - last_ACK_time) ;

BWE[k] = (1 - beta) * (sample_BWE[k] + sample_BWE[k-1]) / 2 + beta * BWE[k-1] ;

endif

这里 acked 是已经确认的报文段的数量,pkt_size 是以字节为单位的报文段尺寸,now 是当前时刻,last_ACK_time 是收到前一个 ACK 的时刻;beta 是极点([11]建议 beta = 19/21). 在文献[4]中指出,Algorithm WESTWOOD1 算法产生较大误差的原因是该算法在对样本带宽处理时采用了固定极点的滤波器,也就是说对带宽样本的算术平均值并不能代表平均带宽. 文献[6]提出了改善方法,Algorithm WESTWOOD2 算法如下:

Algorithm WESTWOOD2:

if (ACK is received)

ACK_interval = now - last_ACK_time ;

sample_BWE[k] = (acked * pkt_size * 8) / ACK_interval ;

pole = (2 * tau - ACK_interval)/(2 * tau + ACK_interval) ;

BWE[k] = (1 - pole) * (sample_BWE[k] + sample_BWE[k-1]) / 2 + pole * BWE[k-1] ;

endif

但这个算法在遇到 ACK 压缩以及与 TC PReno 共享一个链路的时候,会过高地估计可用带宽,因而无法达到公平性.

4) CSFQ 算法(Core Stateless Fair Queueing Estimation Algorithm)

在文献[12-13]中提出了直接从带宽样本估计带宽的非线性技术,即 CSFQ 算法,它最初是为 IP 路由器设计的. 该算法对返回 ACK 的速率进行过滤,并且在发生拥塞后根据公式(1)来优化 ssthresh 的值. 带宽估计算法由下述方程描述:

$$Bwe[k] = (1 - e^{\frac{-T[k]}{K}}) * \frac{L[k]}{T[k]} + e^{\frac{-T[k]}{K}} * Bwe[k-1] \quad (2)$$

这里 *Bwe* 是低通滤波后的估计带宽,$L[k]$是最近收到的 ACK 所确认的字节数,$T[k]$是最近两个 ACK 的时间间隔,$L[k]/T[k]$是 ACKs 流的瞬时速率,K 是时间常数(在文献中它的取值范围被推荐为 0.1-0.5). 这个算法也存在对带宽的过高估计,同时当 K 较小时所估计的带宽振荡加剧;而 K 较大时,震荡虽然减小了,但是推迟了对带宽做出估计的时间.

5) TC PJersy 算法[14]

该算法与 TC PWestwood 的思想类似,即在发送端通过观察返回 ACKs 的速率来估计带宽. 算法如下:

$$R_n = \frac{RTT * R_{n-1} + L_n}{(t_n - t_{n-1}) + RTT} \quad (3)$$

这里 Rn 是在 tn 时刻第 n 个 ACK 到达时的估计带宽,tn 是接受到第 n 个 ACKs 的时刻,Ln 是被确认的分组 n 的尺寸,RTT 是往返时间. 可以看出,当遇到 ACK 压缩或路由变更时会过高.

2 可用带宽算法研究的意义

文章对 TCP 拥塞控制中增强带宽估计算法相关的一些相关问题进行了讨论和分析,这些算法与 TC PReno 中使用的不同,因此它们有可能通过考虑丢包事件发生时带宽的情况来判断出丢包是拥塞造成的还是误码造成的,进而优化 ssthresh 和 cwnd 来提高网络吞吐量. 但是无论什么样的算法都必须满足公平性要求,而达到公平性要求的一个关键因素是带宽估计的准确性. 因此如何使带宽估计更加准确,是亟待解决的问题.

参考文献:

[1] MATHIS m,SEMKE J, MAHDAVI J. The Macroscopic Behavior of the TC PCongestion Avoidance Algorithm [J]. ACM Computer Comm. Rev, 1997, 27(3):21-24.

[2] LAKSHMAN T V ,MAKHOW U. The Performance of TCP/IPfor Networks with High Bandwidth - Delay Products and Random Loss [J]. IEEE/ACM Trans. Networking, 1997, 5(3):336-350.

[3] BALAKRISHNAN H,PADMANABHAN V N, S. Seshan, et

al. A Comparison of Mechanisms for Improving TC PPerformance over Wireless Links [J]. IEEE/ACM Trans. Networking, , Dec. 1997, 5(6): 759 - 769.

[4] KESHAV S. A Control - Theoretic Approach to Flow Control [J]. Proc. ACM SIGCOMM, 1991,(9):3 - 15.

[5] ALLMAN M, PAXSON V. On Estimating End - to - End Network Path Properties [J]. SIGCOMM Computer Comm Rev,2001,31(2):124 - 151.

[6] MASCOLO S, SANADIDI M Y, CASEGTI C, et al. TC PWestwood: End - to - End Congestion Control for Wired/ Wireless Networks [J]. Wireless Networks , 2002,(8): 467 - 479.

[7] MOGUL J C. Observing TC PDynamics in Real Networks. Proc. ACM SIGCOMM Symp[J]. Comm Architectures and Protocols, 1992, (5):305 - 317.

[8] HOE J C. Improving the Start - U PBehavior of a Congestion Control Scheme for TCP[J]. ACM SIGCOMM Computer Comm Rev, 1996, 26(4): 270 - 280.

[9] BRAKMO L S, PETERSON L L. TC PVegas: End - to - End Congestion Avoidance on a Global Internet[J]. IEEE. Selected Areas in Comm,1995, 13(8): 1 465 - 1 480.

[10] WANG R, VALLA M, SANADIDI M Y, et al. Adaptive Bandwidth Share Estimation in TCP Westwood[J]. Globecom,2002,(3):2 604 - 2 608.

[11] ZHANG L, SHENKER S, CLARK D, Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two - Way Traffic, Proc. SIGCOMM Symp[J]. Comm. Architectures and Protocols, 1991, (9):133 - 147.

[12] MASCOLO S, CASETTI C, GERLA M. Sanadidi, TCP Westwood: Congestion Control with Faster Recovery[R]. UCLA CS:Technical Report 200017, 2000.

[13] STOICA I, SHENKER S, ZHANG H. Core - Stateless Fair Queueing:Achieving Approximately Fair Bandwidth Allocations in High Speed Networks [Z]. Proc. ACM SIGCOMM, Sept. 1998.

[14] XU KAI, YE TIAN, ANSARI, N.; TCP - Jersey for Wireless Ipcommunications[J]. IEEE Selected Areas in Communications, 2004,22(4):747 - 756.

Survey of Bandwidth Estimation Scheme in TCP

WANG Ji-feng, HU Han, WANG Chun-hui

(Chongqing University of Posts and Telecommunications, Chongqin 400065, China)

Abstract: Transmission Control Protocol (TCP) is tuned to perform well in wired networks or environment with low bit errors where packet losses occur mostly because of congestion. However, when TC Pis operated in high BER environment and suffers from significant losses due to bit errors, it does not make the best use of the available bandwidth of links because TC Phavles its current congestion window blindly. As a result, the performance of TC Pwould be deteriorated very much. Recently, the use of enhanced bandwidth estimation procedures within the congestion control scheme of TC Pwas proposed as a way of improving TC Pperformance over links affected by random loss. This paper first analyzes the problems faced by every bandwidth estimation algorithm implemented at the sender side of a TCP connection. Some proposed estimation algorithms, accuracy and performance are estimated. At the same time, the authors discuss the relationshi Pbetweent the bandwidth estimation accuracy and the fairness of protocol. Finally, this paper point out the further work should be done in future.

Key words: congestion control; random loss; bandwidth estimation

(编辑　陈移峰)