ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.1, January 2008, pp.125−135
DOI: 10.3724/SP.J.1001.2008.00125

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

高带宽延时网络中一种协同式拥塞控制协议*

王建新+, 龚 皓, 陈建二

(中南大学 信息科学与工程学院,湖南 长沙 410083)

A Cooperant Congestion Control Protocol in High Bandwidth-Delay Product Networks

WANG Jian-Xin+, GONG Hao, CHEN Jian-Er

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

+ Corresponding author: Phn: +86-731-8830212, Fax: +86-731-8830212, E-mail: jxwang@mail.csu.edu.cn

Wang JX, Gong H, Chen JE. A cooperant congestion control protocol in high bandwidth-delay product networks. *Journal of Software*, 2008,19(1):125−135. http://www.jos.org.cn/1000-9825/19/125.htm

Abstract: This paper presents a Cooperant Congestion Control Protocol (C^3P) that uses 1 bit routers' explicit feedback predicted information and delay signals to adjust the congestion windows appropriately. Simulation results show that C^3P can efficiently improve the bandwidth utilization, TCP (transmission control protocol)-friendliness, RTT (round trip time) fairness and reduce the packet drop rate in High Bandwidth-Delay Product Networks.
Key words: TCP; congestion control; high bandwidth-delay product networks

摘 要: 提出一种协同工作式的 TCP(transmission control protocol)拥塞控制改进协议 C^3P(cooperant congestion control protocol),通过 C^3P 源端检测 RTT(round trip time)延时信息和路由器反馈的 1 bit 显式预测信息来判断网络拥塞状态,自适应地调节拥塞窗口.仿真实验表明,C^3P 协议能够有效地适应这种高带宽延时网络的传输特性,以保证网络获得更优的链路利用率、TCP 友好性以及流与流之间的公平性.
关键词: TCP;拥塞控制;高带宽延时网络
中图法分类号: TP393 **文献标识码**: A

TCP(transmission control protocol) Reno 算法[1]自 1988 年被提出来之后,一直被认为是一种效果很好的 Internet 网络传输控制协议,而被广大科研人员所接受,沿用至今[2].但进入 21 世纪后,随着吉比特网络、无线网络、传感器网络和卫星网络的不断兴起和普及,传统的 TCP 拥塞控制协议在这些网络环境下面临着很大的挑战.特别是随着网络技术的飞速发展和接入性能的不断提高,如今,全世界的互联主干网络呈现出一种高带宽高延时(high bandwidth-delay product networks)的网络特性[3].在这种网络环境下,伴随着网络带宽和往返时延的不断增加,TCP 协议本身反而成为了限制网络性能的瓶颈:

* Supported by the National Natural Science Foundation of China under Grant Nos.90304010, 60673164 (国家自然科学基金); the Excellent Youth Foundation of Hu'nan Scientific Committee of China under Grant No.06JJ10009 (湖南省杰出青年基金); the Program for New Century Excellent Talents in University of the Ministry of Education of China under Grant No.NCET-05-0683 (新世纪优秀人才支持计划); the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No.20060533057 (高等学校博士学科点专项科研基金)

Received 2006-05-11; Accepted 2006-12-06

(1) TCP 效率低下,带宽链路利用率很低,传统“慢启动”和“拥塞避免”机制不适用于高速网络[3−7].举例说,一个包长为 1 500 字节和 100ms 延时的标准 TCP 流,要快速达到 10Gbps 的网络利用率,至少需要 1/50 亿的低丢包率.而一旦发生丢包,由于“拥塞避免”机制的保守性,若从拥塞避免阶段恢复到网络高利用率稳态,则至少需要 1 个多小时的时间,这在实际网络中完全无法让人接受[3];

(2) 在传统 TCP 协议中就一直就存在的流与流之间 RTT(round trip time)不公平性的问题,在高带宽延时网络上表现得更加尖锐,不公平性现象表现得更加显著[3−5];

(3) 文献[8−10]证明,在高带宽延时网络环境下,TCP 流将会不断抖动,造成路由器上的队列长度产生不稳定性.文献[8]还指出,任何一种主动队列管理机制在高带宽延时网络中都不能维持其队列稳定性,而 TCP 的性能都会随着链路带宽或延时的增加而逐渐降低.

为此,各国学者都针对这些问题提出了一系列 TCP 改进协议,虽然在带宽利用率上获得了较好的效果,但仍面临着 TCP 友好性低下、链路丢包率过高以及流之间公平性无法保证等诸多问题.

本文在详细分析高带宽延时网络下拥塞控制协议研究成果的基础上,提出了一种协同工作式的拥塞控制协议 C^3P(cooperant congestion control protocol),通过源端监测 RTT 延时信息和中间路由器反馈的 1 bit 显式预测信息来判断网络拥塞状态,自适应地调节拥塞窗口.

本文第 1 节阐述高带宽延时网络下的拥塞控制协议的研究现状.第 2 节给出协同工作式拥塞控制协议 C^3P 的详细描述.第 3 节是 C^3P 算法与 HSTCP(high speed TCP)[3]和 BIC(binary increase congestion control)-TCP[4]算法的仿真实验对比.第 4 节给出研究结论及下一步的研究工作.

1 研究现状

TCP 拥塞控制协议自 20 世纪 80 年代中期被提出来之后,经过数十年的研究发展,出现了大量的改进和增强版本[11].而在如此众多的 TCP 改进协议中,拥塞窗口管理是 TCP 流量控制的重中之重.TCP 拥塞控制协议大部分改进和增强版本的焦点都集中在窗口管理机制上.而基于这些协议调节窗口所利用的反馈信息,我们可以大致地将其划分为 3 类:

(1) 基于丢包反馈的协议(loss-based congestion avoidance,简称 LCA);

(2) 基于路径延时反馈的协议(delay-based congestion avoidance,简称 DCA);

(3) 基于显式反馈的协议(ECN(explicit congestion notification)-like algorithms).

在基于丢包反馈的 TCP 拥塞控制协议中,最早出现的是 TCP Tahoe 算法,它通过 ACK 所带回来的丢包信息来调整源端的拥塞窗口.其后的 TCP Reno,NewReno 和 SACK 都是针对 ACK 返回的丢包信息来进行的传统 TCP 改进.近几年来,随着高带宽延时网络的普及,针对提高 TCP 带宽利用率这一点上,又涌现出许多新的基于丢包反馈的 TCP 改进协议.比较典型的有 HSTCP[3],BIC-TCP[4]和 STCP(scalable TCP)[5]协议.

在 TCP 拥塞控制协议中,另一种窗口管理的方式是基于路径延时反馈的,首推代表是传统的 TCP Vegas[12]协议.它通过观测 TCP 连接中 RTT 延时的变化来调节拥塞窗口.如果发现 RTT 变大,则认为网络发生拥塞,相应地减小拥塞窗口;如果 RTT 变小,则认为拥塞已经解除,并增加拥塞窗口;如果 RTT 保持不变,则不改变拥塞窗口的大小.因为 RTT 信号相对于丢包信号来说反应更加灵敏,更能及时地反映出一般网络中的拥塞状况,所以,这种通过检测 RTT 的变化进行判断的拥塞控制协议能够较好地预测网络带宽的使用情况,对小缓存的路由器适应性较强,效率也较为理想.在高带宽延时网络上,针对延时反馈的方式,也提出了一系列的改进协议,较为典型的有 FAST TCP[6]和 Astart[7].

除了这两种拥塞控制方式外,TCP 还采用了一种显式反馈形式的窗口管理机制.如 ECN(显式拥塞通告)[13].它利用路由器自己检测本身的拥塞状态,直接发送反馈给 TCP 源端,以调整源端的窗口值及发送速率.在此基础上所提出来的 XCP(explicit congestion protocol)[9],AECN(anti ECN)[14]都是采用类似的显式反馈方式.

然而,虽然这些协议都极大地提高了 TCP 在高带宽延时网络上的吞吐量,但由于种种原因,协议本身的一些弊端在这种网络环境下不但没有减轻,反而表现得更加明显.这些弊端主要包括:

(1) TCP 友好性低下

虽然新的、优秀的拥塞控制协议取代传统的 TCP 协议将会是今后的大势所趋,但在一定时期内,仍会出现新协议与传统协议共存同一网络链路的情况.而在这种情况下,如何有效地提高新协议流的吞吐量,又不过分降低传统 TCP 流的带宽,这是拥塞控制协议所必须具备的要素之一.与前面所提出来的诸多算法相比,HSTCP 协议和 STCP 协议在这一点上存在着非常严重的问题[4].高速流抢占了瓶颈链路的大量可用带宽,而传统的 TCP Reno 流只获得了很少的吞吐量,几乎被饿死,这是协议设计者所不愿意看到的.

(2) RTT 不公平性严重

流与流之间 RTT 不公平性的问题,在文献[4]中已被提出,两个不同 RTT 流之间的吞吐量比值关系为

$$\frac{w_1/RTT_1}{w_2/RTT_2}=\left(\frac{RTT_2}{RTT_1}\right)^{\frac{1}{1-d}}.$$

其中,d 为拥塞控制协议相关参数,传统的 TCP 协议和 AIMD(additional increase multiple decrease)算法的 d 值都为 0.5,而 HSTCP,STCP 以及 BIC-TCP 等协议都因为改变了 TCP 的窗口增长规律,提高了 d 值,使得 RTT 不公平性问题不但没有减轻,反而更加严重.

(3) 触发拥塞丢包事件的概率

HSTCP 等拥塞控制协议使得 TCP 流吞吐量在高带宽延时网络上迅速增加,这一方面极大地提高了网络的带宽利用率;另一方面,对于基于丢包的 TCP 协议来说,大幅度提高网络利用率的同时也意味着接近网络带宽总量的速度越快,而越快地接近网络带宽总量也意味着下一次拥塞丢包事件为期不远了.这些协议在提高网络带宽利用率的同时也间接地加大了网络的丢包率.

(4) 无法准确判断网络拥塞

如何能够有效而快速地反映网络拥塞,一直是拥塞控制研究中一项重要的研究要素.在现实网络中,TCP 协议采用丢包事件作为网络拥塞的信号.但近年来的一些研究[6,12]表明,丢包事件不能及时反映网络中的拥塞.而 RTT 延时信息由于其可变性,相对于丢包事件来说反应更加灵敏,更能及时地反映出一般网络中的拥塞状况.为此,接下来的一些拥塞控制改进协议试图采用 RTT 延时检测来判断网络的拥塞情况,如 FAST TCP 和 Astart.但是近年来,文献[15,16]中表明,高速网络中基于延时的拥塞避免方式仍存在着一些弊端,在某些环境下也不能反映出真实的网络拥塞情况.文献中指出,网络中不断增加的 RTT 延时和丢包事件并没有直接的对应关系,反而网络中随时可能出现的噪声将会对 RTT 检测产生干扰,造成协议无法准确、有效地控制窗口变化.另一方面,显式速率反馈机制也能有效地检测出网络的拥塞,如 XCP 协议.但是,这些协议过于复杂,往往需要在数据包头插入一段很长的控制信息位,这对于数据包内现有空间所剩不多的 TCP/IP 协议来说,在实现上将会是很大的一个问题.

2 C^3P 协议

无论是丢包反馈协议(LCA)还是延时反馈协议(DCA),在高带宽延时网络中都无法及时、准确地反映出网络的真实拥塞状况.DCA 协议虽然在传统网络环境中被证明比 LCA 协议更加有效,但当带宽远大于流速时,DCA 协议也渐渐开始失效.为此,必须采用一种更为有效的方法来反映网络拥塞,比如多种反馈机制相结合的方式.

本文提出了一种结合多种反馈机制的协同工作式拥塞控制协议 C^3P,通过源端监测 RTT 延时信息和路由器反馈的 1 bit 显式预测信息来判断网络拥塞状态,自适应地调节拥塞窗口.C^3P 提供了一种由端节点和中间路由合作反馈网络状态的新型拥塞控制模式.下面我们将从 5 个设计要点来详细描述该协议.

2.1 网络路径拥塞状态相关性

如何真实、有效地反映出网络拥塞,是拥塞控制协议首要考虑的要素之一.在 C^3P 协议中,主要考虑一条传输路径上的拥塞状态,并依据该路径的拥塞状态调节拥塞窗口的大小.下面我们主要分析与网络路径拥塞状态

的相关要素.

我们定义下一τ时隙网络路径上任一节点 i 的负载因子预测值 LF_i 为

$$LF_i = \frac{r_i}{C_i} \tag{1}$$

其中,r_i 表示下一时隙该节点所有入口流到达速率之和的预测值,C_i 表示链路的服务速率(即链路带宽).而网络路径 P={1,2,…,i,…,m}上下一时隙的负载因子 LF_P 为该路径上所有节点负载因子预测值的最大值,即

$$LF_P = \max\{LF_i \mid i \in P\} = \max\left(\frac{r_i}{C_i} \mid i \in P\right).$$

再定义某连接 C 在路径 P 上所有节点缓存队列中排队的数据包数目之和为

$$QL_{CP} = \sum_{i=1}^{m} \tilde{q}_i ,$$

其中,$\tilde{q}_i$ 是连接 C 在路由器 i 上排队数据包数目的估计值.

LF_P 反映的是整个全局网络中瓶颈节点下一时隙的拥塞情况,其值是由共享这条路径的所有连接所决定的.而 QL_{CP} 则考虑的是单个连接在路径中的拥塞情况,它的值由单个连接的发送速率所决定.由于网络的复杂性,我们希望看到拥塞控制协议既能考虑到网络上所有连接所造成的全局负载情况,又要顾及到局部上单个流能以公平的发送速率来共享带宽、处理拥塞.它们之间并不是一个独立的过程,而是一种相互制约、相互作用的联动行为.所以,对于网络路径 P 的拥塞状态(congestion state)CS_P,我们认为,必须将其与全局因素 LF_P 和局部因素 QL_{CP} 相互关联起来,并用以下函数式表示:

$$CS_P = f(LF_P, QL_{CP}) = f\left\{\max\left(\frac{r_i}{C_i} \mid i \in P\right), \sum_{i=1}^{m} \tilde{q}_i\right\} \tag{2}$$

上述二者决定了某一时段内网络路径 P 的真实拥塞状态.二者之间既相互独立又存在着一定的联系.在 C^{3}P 协议中,针对 LF_P 和 QL_{CP} 两种路径拥塞信息,协议将估测出网络路径的拥塞程度,并做出相应的拥塞窗口机制的调整.下面将分别对两种拥塞检测机制的方法给出详细描述.

2.2 延时检测机制

对于 C^{3}P 源端来说,缓存数据包之和 QL_{CP} 所产生的直接影响就是源端所监控到的 RTT 延时的变化量.源端根据排队论中的 Little Formula 定理,可以近似地通过监控 RTT 响应时间变化量来估算该连接的 QL_{CP} 值,如公式(3)所示:

$$QL_{CP} = \sum_{i=1}^{m} \tilde{q}_i \approx \frac{cwnd \times (sRTT - baseRTT)}{sRTT} \tag{3}$$

其中,$sRTT$ 是 C^{3}P 源端所测得的高精度指数平滑往返时间,$baseRTT$ 是每流所测得的最小往返时间.这个最小值近似地等于整个 C^{3}P 连接路径中的传输延时,而($sRTT$-$baseRTT$)则是对当前网络中排队延时的一个估计值.由 C^{3}P 源端每秒发送 $cwnd$/$sRTT$ 个数据包可知,$cwnd$×($sRTT$-$baseRTT$)/$sRTT$ 从物理意义上可以理解为该连接在路由器缓存队列上所排队包数的估计值[12].

在 C^{3}P 源端,发送方通过监控网络中 ACK 所带回来的 RTT 延时情况,采用公式(3)来估计某一连接在路径上路由器缓存队列中所排队包的数目.根据估算出来的路由器缓存队列中排队包数目 QL_{CP} 以及阈值 α,通过下式我们可以得到网络延时状态信息位 S_1.

$$\begin{cases} S_1 = 1, & QL_{CP} \geq \alpha \\ S_1 = 0, & QL_{CP} < \alpha \end{cases},$$

其中,阈值α是一个参数变量,它决定着整个协议对网络延时信号的敏感程度.当 QL_{CP}>α时,意味着该连接当前的发送速率已经过大,引起了路由器上的额外排队行为,使得往返延时加大.这时,C^{3}P 协议将 S_1 状态信息设为 1,表示路径上排队的包数目超过了我们所期望的值,开始临近拥塞.

2.3 1 bit显式拥塞反馈机制

另一方面,下一时隙中路径负载因子最大值 LF_P 是由网络路径中拥塞程度最重的节点所造成的,它决定了网络中最大的瓶颈所在.根据网络流量的自相似性,C^{3}P 协议中每个路由器 i 首先预测分组到达速率(第 2.4 节将给出预测方法),然后利用公式(1)来计算下一个 τ 时隙内自己的负载因子 LF_i,并将自身的拥塞状态划分为低负载和高负载两个等级.根据这个预测所得到的负载因子 LF_i 以及阈值 β,通过下式得到 1 bit 的 S_2 拥塞状态信息位.

$$\begin{cases} S_2 = 1, & LF_i \geq \beta \\ S_2 = 0, & LF_i < \beta \end{cases}.$$

与 ECN 机制[9]一样,C^{3}P 协议在包头中打上这个 1 bit 的负载信息位 S_2,路径上每个路由器利用根据自身负载状态所计算出来的 S_2 负载信息与包头中原有的 S_2 状态信息进行一次或运算,并将运算结果又重新打回到包头中.当携带有 S_2 负载信息的包到达接收方时,C^{3}P 接收方通过 ACK 确认包将该信息返回给 C^{3}P 源端.通过这种方式,最终 C^{3}P 源端以 S_2 状态信息位的形式获取了整个路径 P 的拥塞状态.

在整个 C^{3}P 显式预测拥塞反馈机制中,如何准确、有效地预测下一时段分组的平均到达速率是协议判断 S_2 状态信息位的重要因素,第 2.4 节我们将讨论如何进行预测.

2.4 分组到达速率预测

一方面,分组到达速率的预测精度直接影响拥塞判断的准确性,关系到算法的性能;另一方面,预测必须简单,计算量要小,因为对于流量的测量和预测以及随后的控制都是在线执行的,复杂的计算会导致算法扩展性的下降.

以往的研究表明,网络流量表现出长程相关性(long range dependent,简称 LRD)[17,18],针对网络流量的长程相关性,长记忆模型(long memory model),如 FBM(fractional Brownian motion)[18],FARIMA(fractal ARIMA)[19]在流量预测方面表现出较好的性能.但是这些模型包含了很大的计算量,因此,这些模型不适合进行在线的流量预测,所以必须寻找新的方法进行在线的流量预测.

近来的研究表明,实际网络流量的 Hurst 参数(Hurst 参数是长程相关的一个标志)很少超过 0.85[17,18],这表明,实际的网络流量并没有表现出非常强的长程相关性.在这种情况下,使用 MMSE(minimum mean square error)进行在线流量预测的精度就接近于使用长记忆模型进行预测的精度.以下是对 MMSE 预测方法的描述.

设$\{X_t\}$表示一个线性随机过程,假设$\{X_t\}$的下一个值可以用以前的观测值的线形组合来表示,即

$$X_{t+1}=w_mX_t+w_{m-1}X_{t-1}+\ldots+w_1X_{t-m+1}+\varepsilon_t=WX'+\varepsilon_t,$$

其中,$W=(w_m,w_{m-1},\ldots,w_1)$,$X=(X_t,X_{t-1},\ldots,X_{t-m+1})$,$m$ 表示回归阶数,ε_t 表示误差.

设 X_{t+1} 的估计量记为 $\hat{X}_{t+1}$,$\hat{W}$ 表示 W 的估计量.为了使均方误差最小化,即 $E[e_t^2]=E[(X_{t+1}-\hat{X}_{t+1})^2]$ 取最小值,由此可得

$$\hat{W}=[\rho_m \quad \ldots \quad \rho_1]\times\begin{bmatrix} \rho_0 & \rho_1 & \cdots & \rho_{m-1} \\ \rho_1 & \rho_0 & \cdots & \rho_{m-2} \\ \cdots & \cdots & \cdots & \cdots \\ \rho_{m-1} & \rho_{m-2} & \cdots & \rho_0 \end{bmatrix}^{-1},\text{其中,}\rho_k=\frac{1}{m}\sum_{t=k+1}^{m}X_tX_{t-k}.$$

MMSE 预测器只涉及少量的矩阵操作,运算不是很复杂,而且矩阵运算可以由高速硬件实现,因此,MMSE 完全可以用于在线预测,所以,C^{3}P 协议最终选用了 MMSE 来估测链路拥塞.详细的流量预测算法见文献[20].

2.5 C^{3}P拥塞窗口管理机制

C^{3}P 确定了网络的拥塞状况以后,如何决定自己的发送速率以保证整个网络的有效和正常地运行,是整个协议设计的关键所在.在这一节中,我们将具体介绍 C^{3}P 协议的拥塞窗口管理机制.

当 C^{3}P 源端和接收端三次握手建立起连接以后,协议首先进入到“慢启动”模式,这个阶段的拥塞窗口增长方式与传统 TCP 一样,都是针对每一个回来的 ACK 进行一次窗口加 1.而当 C^{3}P 连接出现了第 1 次丢包时,协议

则进入到“快速恢复(fast-comeback)”模式.在这个模式下,C^{3}P 源端根据 S_1 和 S_2 状态信息相互组合所得到的 4 种网络拥塞状态,调用 4 种不同的窗口调节机制来自适应地控制网络流量,使协议能够快速、有效地恢复到最佳网络利用状态.具体拥塞窗口变化规则见表 1.

Table 1 C^{3}P congestion window change rules during fast-comeback

表 1 C^{3}P“快速恢复”模式拥塞窗口调整机制

S_1	S_2	Network state	Window adjustment
0	0	Low load	$w=w\times k$
1	0	Low load, but this flow sending rate is too high	$w=w+1/w$
0	1	High load, but this problem is not caused by this flow	$w=w+1$
1	1	High load, congestive	$w=w-1$

如表 1 所示,当 S_1 和 S_2 两个网络状态量均为 0 时,表示这时的网络处于低负载状态,还有很多的剩余带宽没有被充分利用,所以,此时 C^{3}P 采取乘性增加的窗口调整机制,迅速提高网络利用率.当 S_1=1,S_2=0 时,此时网络状态有两种可能:一种是网络拥塞程度仍然较轻,但该连接流速相对过大的情况;而另一种则是因为网络噪声对 RTT 延时检测所产生的干扰所致.在这种不确定的状态下,C^{3}P 协议采用了一种较为平缓的窗口增长方式,$w=w+1/w$,以保证网络的可靠运行.而当 S_1=0,S_2=1 时,则表示网络已经开始拥塞,但拥塞并不是由该连接流所造成的情况.为了提高 C^{3}P 算法的公平性,C^{3}P 设定此时为加性增加的窗口调整机制.最后,当 S_1 和 S_2 两个网络状态量均为 1 时,可以肯定这时的网络处于一种重拥塞的状态,C^{3}P 对其拥塞窗口进行逐步递减,以此来缓解网络的负载.同样,当 C^{3}P 源端接收到丢包反馈信息时,也将其拥塞窗口进行减半操作.

为了更完整地描述 C^{3}P 窗口控制算法,我们给出了其算法在“快速恢复”阶段的伪码描述,如图 1 所示.

```
The fast-comeback mode
Receiving acknowledgement packet p
if p return a drop information the congestion window W=W/2;
else {if (S1=1)
        {if (S2=1) W=W–1; else W=W+1/W;}
      else {if (S2=1) W=W+1; else W=W×k;}
}
```

Fig.1 Pseudo code of C^{3}P congestion window updating rules in *fast-comeback* mode

图 1 C^{3}P“快速恢复”阶段窗口控制算法伪代码

C^{3}P 通过结合网络路径的负载因子和 C^{3}P 连接在路径上所有路由器缓存中的数据包数目两种拥塞状态信息,可以将网络划分为 4 种更细的拥塞等级,更能有效地反映出网络的真实状态.并且,将两种拥塞信息机制统一地融合起来,可以尽量避免单一拥塞检测模式在某些环境下所产生的误判信息,如延时噪声所带来的 RTT 抖动情况等等,降低了单方面信息错误对协议可靠性的直接影响.即使某一种检测信息产生了误判,但由于有另一种信息在制约着它,因而其错误并不会直接影响到协议机制,也不会对协议的性能产生较大幅度的影响.

3 模拟场景和结果分析

3.1 模拟环境与参数配置

为了验证所提出算法的有效性,我们在网络仿真软件 NS-2 上实现了 C^{3}P 协议算法,并进行了大量的仿真实验.作为对比,我们选择 BIC-TCP 协议和 HSTCP 协议进行比较,其协议参数均按照 NS-2 中的默认值进行设置.图 2 给出了模拟实验的网络拓扑结构,其中,S_1 到 S_n 为主机端,D_1 到 D_n 为接收端,$N1$ 和 $N2$ 为中间路由器.为了更加有效地提高模拟实验的真实性,我们在所有模拟过程中都引入了峰值约占带宽大小 5%的泊松分布 UDP(user datagram protocol)流,以此来充当真实网络环境下的背景流.整个模拟过程持续 480s.所有协议流的分组大小均设为 1Kbyte.C^{3}P 协议中的相关参数设置如下:α=3,β=0.8,k=1.01,流量预测时间间隔 τ=200ms.

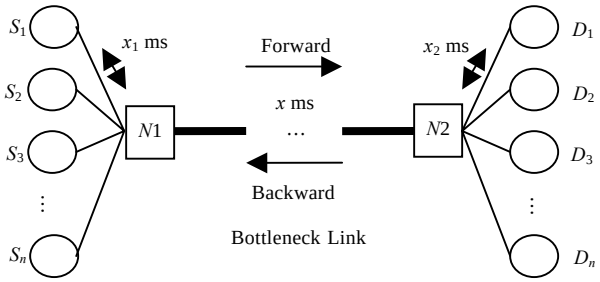


Fig.2 Simulation network topology

图 2 仿真网络拓扑

3.2 链路利用率

首先,我们对 BIC-TCP,HSTCP 和 C^{3}P 协议在高带宽延时网络上的带宽利用率做了一个比较.设定瓶颈链路带宽为 2.5Gbps,2 个相同协议的拥塞控制协议流和 2 个传统 TCP Reno 流共享这一瓶颈链路.每次实验中,我们计算出包括背景流在内的所有数据流总的带宽利用率.同时,在进行 BIC-TCP 协议和 HSTCP 协议仿真时,分别设定了 Drop-tail 和 RED(random early drop)两种队列协议来进行性能比较.

表 2 给出了各协议在带宽利用率上的性能比较.可以看出,在 Drop-tail 队列中,三者的带宽利用率差不多;但在 RED 队列中,BIC-TCP 和 HSTCP 协议的利用率均下降了一些.造成这种现象的主要原因是 Drop-tail 队列允许数据流充满所有的队列缓存,而 RED 队列由于早丢弃的机制,数据流并不能占满瓶颈链路上所有的队列缓存,因而造成整个网络带宽利用率的下降.

Table 2 Link utilization

表 2 带宽利用率

Protocol	Utilization (%)	
C^{3}P	97.1	
	Drop-Tail	RED
BIC-TCP	97.0	95.3
HSTCP	97.2	92.1

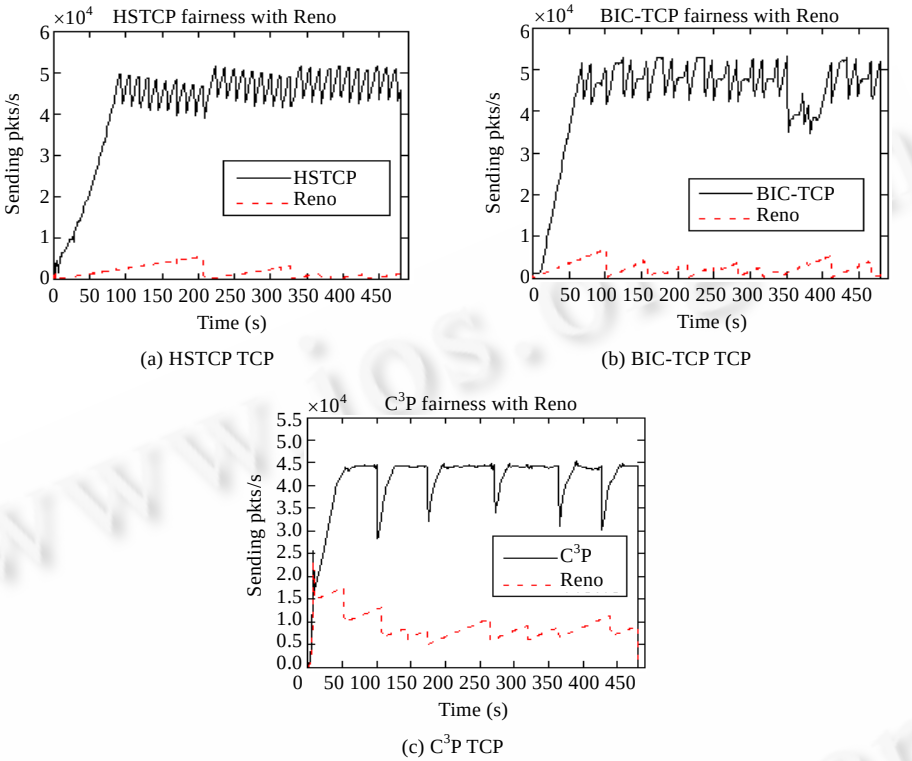
3.3 TCP友好性

随着网络接入性能的不断提高和拥塞控制协议的不断发展,网络中将会出现多种拥塞控制协议共存的情况.在这种情况下,如何保证各协议流之间的正常运行以及提高拥塞控制改进协议流与传统 TCP Reno 协议流的 TCP 友好性,是协议设计中的关键要素之一.这一节中,我们给出了 BIC-TCP,HSTCP 和 C^{3}P 三种协议的 TCP 友好性评价比较.在实验中分别定义两类主机:一类采用拥塞控制改进协议;而另一类则采用传统 TCP Reno 协议.根据图 2 所示的哑铃状拓扑环境,接入链路带宽和延时分别设为 1Gbps 和 1ms,而不同协议流所共享的瓶颈链路为 622Mbps,传输延时为 80ms.在此基础上,我们分别比较不同拥塞控制协议流和 Reno 流共存条件下的发包速率之比.

如图 3(a)、图 3(b)所示,HSTCP 和 BIC-TCP 在大幅度提高自身发包速率的同时,也严重影响了 Reno 流的吞吐率.可以看出,它们抢占了大量的网络剩余带宽资源,导致 TCP-Reno 流几乎被饿死.由此可见,HSTCP 和 BIC-TCP 协议的 TCP 友好性并不能达到一个令人满意的程度.并且由于协议自身的侵略性,同时还给网络带来了大量的丢包,加大了整个网络的抖动.如图 3(c)所示,C^{3}P 协议相对降低了对 Reno 流的影响,C^{3}P 流和 Reno 流均获得了较为理想的带宽分配.C^{3}P 流的瞬时发包速率和 Reno 流的瞬时发包速率之比约为 5:1.很明显,C^{3}P 提高了协议在高带宽延时网络中的 TCP 友好性.同时,C^{3}P 在整个模拟过程中表现得较为稳定,并降低了协议的丢包率.

造成这一结果的主要原因在于 C^{3}P 所采用的延时检测机制降低了协议流抢占瓶颈路由器上过多额外缓存空间事件的概率.像 Vegas 协议一样,C^{3}P 协议认为单个流最合理的发送速率应当保证该流在路由器缓存队列上

的排队包数不超过某一特定阈值.如果路由器缓存队列中排队包大部分均属于同一连接流,那么,很显然,该流的发送速率过大,因为路由器上的拥塞基本上断定是由它所造成的.C^3P 将空出来的缓存空间让给 Reno 协议流,进一步加大了 Reno 流的发送速率,减小了两个流之间的瞬时吞吐量之比,提高了协议的 TCP 友好性.



(a) HSTCP TCP (b) BIC-TCP TCP

(c) C^3P TCP

Fig.3 TCP-Friendliness comparison

图 3 TCP 友好性比较

3.4 RTT公平性

在拥塞控制协议中,传输延时 RTT 差异会带来吞吐量的不公平性,本节进行了相关的实验研究.我们设计了两个传输延时不同的流共享同一瓶颈链路.瓶颈链路的带宽为 622Mbps,延时 5ms.第 1 个传输流的往返传输延时设为 30ms,第 2 个传输流的 RTT 则分别设为 30ms,60ms,90ms 和 180ms 进行实验.通过这种设置来检测,在不同传输延时下协议流之间吞吐量公平性的变化情况.

从表 3 可以看出,如第 1 节所述,HSTCP 和 BIC-TCP 在此环境下均表现出严重的 RTT 不公平性.实验中,往返传输延时的流越小,占据的网络带宽越多.而随着往返延时比率的不断加大,流与流之间的吞吐量比率也越来越大,造成 RTT 较大的数据流得不到有效的带宽分配,活活被饿死.文献[4]中曾经提出,当加入了 RED 路由队列后,整个网络的 RTT 不公平性将得到改善.为此,我们也对各个协议加入 RED 队列后的性能做了一次统计.如表 3 所示,协议在加入了 RED 队列后,在一定程度上也确实降低了 RTT 不公平性,但总的来说,吞吐量比率还是比较大,不公平性问题仍然很严重.

同样,从表 3 中还可以看出,C^3P 在 RTT 公平性方面获得了较为显著的改善.流与流之间吞吐量的比率约等于 RTT 值之间的比率.无论是在 Drop-tail 队列下,还是在 RED 队列下,这个结果甚至好过 TCP Reno 协议在同等网络环境下所得到的结果.C^3P 得到这样较为理想的 RTT 公平性性能的原因主要在于协议接近拥塞时采用了基于 RTT 延时检测的拥塞窗口动态调整机制.RTT 较小的流刚开始的时候增长速率很快,不断地将两流之间吞吐量比值拉大.但随着该流发送速率的不断加大,它对延时变化量也变得更加敏感.于是,它将先于 RTT 较大的流

进入到窗口递减模式.相反地,RTT 较大的流这时仍处于窗口递增的模式,使其瞬时吞吐量不断增加,最终使二流的吞吐量比值维持在一个较小的范围内,保证了 C^3P 协议的 RTT 公平性.

Table 3 The throughput ratio of protocols

表 3 各协议流吞吐量之比

Inverse RTT Ratio	1	2	3	6
C^3P	1.022 341	1.416 702	2.031 433	8.922 543
BIC in drop-tail	1.033 434	4.025 060	9.696 173	30.431 100
BIC in RED	1.164 987	3.017 912	5.674 582	24.454 215
HSTCP in drop-tail	0.942 102	18.273 328	92.844 748	384.833 044
HSTCP in RED	1.155 956	14.192 811	60.578 422	248.157 821
Reno in drop-tail	1.116 590	2.792 160	8.476 595	16.015 658
Reno in RED	1.227 548	2.344 552	5.640 785	22.254 875

3.5 收敛性

当有一个新流加入到网络中时,网络重新恢复到公平带宽分配的收敛时间同样也是衡量一个拥塞控制协议性能的要素之一.在实验中,我们建立了 4 个 RTT 为 100ms 的传输连接.其中两个连接在 0~60s 时间内随机触发,而另外两个则在 100~160s 时间内随机触发.总的模拟实验持续 600s,瓶颈带宽为 622Mbps.为了结果的准确性,我们从第 100s 后每隔 50s 统计一次吞吐量的公平指数 Fairness Index[21].

如图 4 所示,除了 HSTCP 协议在整个过程中有一些轻微抖动外,各协议流的收敛均比较迅速,体现了较好的带宽分配收敛性.

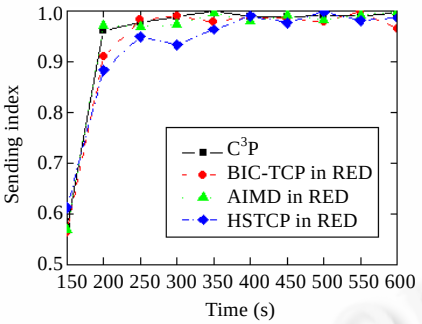


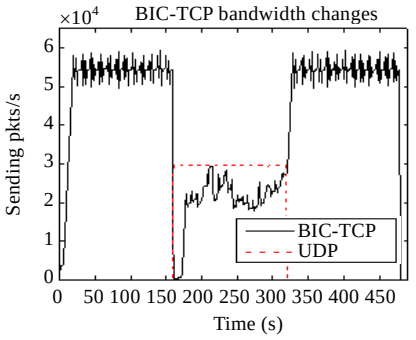
Fig.4 Fairness Index over various time scales

图 4 不同时间段的 Fairness Index

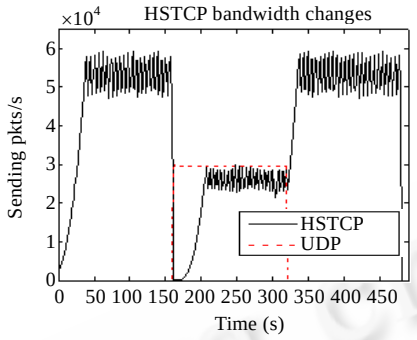
3.6 网络自适应性

在此,我们主要检测 3 种拥塞控制协议对网络变化的自适应性.瓶颈链路设为 622Mbps,整个网络的 RTT 延时为 84ms.在实验进行到 160s 时,引入一个 300Mbps 的 UDP CBR 流到网络中,320s 后停止该流.

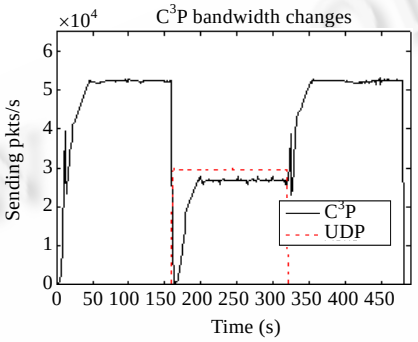
从图 5(a)~图 5(c)分别可以看出,3 种拥塞控制协议均表现出较好的网络自适应性.当 160s UDP 流引入时,各协议都迅速降低了自己的发包速率,紧接着快速占满了剩余的 300M 带宽.而当 UDP 流取消时,协议又能自适应地进入到快速增长模式,重新占用新空出来的剩余带宽.因为在 C^3P 协议的稳定状态采用了类似于 Vegas 协议的拥塞窗口管理机制,所以,相对于其他改进协议而言,C^3P 流在整个模拟过程中都表现得更加稳定.



(a) BIC-TCP



(b) HSTCP



(c) C^3P

Fig.5 Adapting comparison

图 5 协议自适应性比较

4 结 论

本文在详细分析高带宽延时网络下拥塞控制协议研究成果的基础上,提出了一种基于延时信息和网络负载的协同工作式的拥塞控制协议 C^3P.协议结合了网络中的丢包、延时以及显式反馈信息动态调整 C^3P 源端的发送速率.仿真实验表明,在高带宽延时网络环境下,C^3P 算法在吞吐量利用率、TCP 友好性、RTT 公平性以及丢包率等方面都取得了较好的效果.并且像 ECN 机制一样,它较容易实施到现有的网络中.但是在实现 C^3P 算法时,仍需对一些协议配置参数进行设定.在我们今后的工作中,如何对这些参数进行优化选择和根据网络不同状态动态地调整参数配置,将会是进一步的研究目标.

References:

[1] Jacobson V. Congestion avoidance and control. ACM Computer Communication Review, 1988,18(4):314−329.

[2] Luo WM, Lin C, Yan BP. A survey of congestion control in the Internet. Chinese Journal of Computers, 2001,24(1):1−18 (in Chinese with English abstract).

[3] Floyd S. Highspeed TCP for large congestion windows. IETF RFC3649, Experimental, 2003.

[4] Xu L, Harfoush K, Rhee I. Binary increase congestion control for fast long-distance networks. In: Proc. of the IEEE INFOCOM 2004. Piscataway: IEEE Press, 2004. 2514−2524.

[5] Kelly T. Scalable TCP: Improving performance in highspeed wide area networks. ACM SIGCOMM Computer Communication Review, 2003,32(2):83−91.

[6] Jin C, Wei D, Low SH. FAST TCP: Motivation, architecture, algorithms, performance. In: Proc. of the IEEE INFOCOM 2004. Piscataway: IEEE Press, 2004. 2490−2501.

[7] Wang R, Pau G, Yamada K, Sanadidi MY, Gerla M. TCP startup performance in large bandwidth delay networks. In: Proc. of the IEEE INFOCOM 2004. Piscataway: IEEE Press, 2004. 796−805.

[8] Low SH, Paganini F, Wang J, Adlakha S, Doyle JD. Dynamics of TCP/AQM and a scalable control. In: Proc. of the IEEE INFOCOM 2002. Piscataway: IEEE Press, 2002. 239−248.

[9] Handley KM, Rohrs C. Congestion control for high bandwidth-delay product networks. In: Proc. of the SIGCOMM 2002. Pittsburgh: ACM Press, 2002. 89−102.

[10] Ren FY, Lin C, Ren Y, Shan XM. Congestion control algorithm in large-delay networks. Journal of Software, 2003,14(3):503−511 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/503.htm

[11] Zhang M, Wu JP, Lin C. Survey on Internet end-to-end congestion control. Journal of Software, 2002,13(3):354−363 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/13/354.htm

[12] Brakmo L, O'Malley S, Peterson L. TCP vegas: End-to-End congestion avoidance on a global Internet. IEEE Journal on Selected Areas in Communications, 1995,13(8):1465−1480.

[13] Ramakrishnan KK, Floyd S. The addition of explicit congestion notification (ECN) to IP. IETF RFC 3168, 2001.

[14] Kunniyur SS. AntiECN marking: A marking scheme for high bandwidth delay connections. In: Proc. of the ICC 2003. Piscataway: IEEE Press, 2003. 647−651.

[15] Prasad RS, Jain M, Dovrolis C. On the effectiveness of delay-based congestion avoidance. In: Proc. of the PFLDNet 2004. Illinois, 2004. http://www.dsd.lbl.gov/DIDC/PFLDnet2004/papers/Dovrolis.pdf

[16] Martin J, Nilsson A, Rhee I. Delay-Based congestion avoidance for TCP. IEEE/ACM Trans. on Networking, 2003,11(3):356−369.

[17] Crovella ME, Bestavros A. Self-Similarity in World Wide Web traffic: Evidence and possible causes. IEEE/ACM Trans. on Networking, 1997,5(6):835−846.

[18] Song AM, Li SQ. A predictability analysis of network traffic. Computer Networks, 2002,39(2):329−345.

[19] Shu YT, Wang L, Zhang LF, Xue F, Jin ZG, Yang O. Internet traffic modeling and prediction using FARIMA models. Chinese Journal of Computers, 2001,24(1):46−54 (in Chinese with English abstract).

[20] Gao WY, Wang JX, Chen SQ. PFED: A prediction-based fair active queue management algorithm. In: Proc. of the IEEE ICPP 2005. Oslo, 2005. 485−491.

[21] Jain CR. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. Journal of Computer Networks and ISDN, 1989,17(1):1−14.

附中文参考文献:

[2] 罗万明,林闯,阎保平.TCP/IP 拥塞控制研究.计算机学报,2001,24(1):1−18.

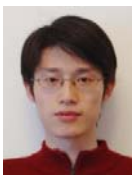
[10] 任丰原,林闯,任勇,山秀明.大时滞网络中的拥塞控制算法.软件学报,2003,14(3):503−511. http://www.jos.org.cn/1000-9825/14/503.htm

[11] 章淼,吴建平,林闯.互联网端到端拥塞控制研究综述.软件学报,2002,13(3):354−363. http://www.jos.org.cn/1000-9825/13/354.htm

[19] 舒炎泰,王雷,张连芳,薛飞,金志刚,Oliver Y.基于 FARIMA 模型的 Internet 网络业务预报.计算机学报,2001,24(1):46−54.

王建新(1969－),男,湖南邵东人,博士,教授,博士生导师,主要研究领域为计算机网络,网络优化理论,计算机优化算法.

陈建二(1954－),男,博士,教授,博士生导师,主要研究领域为计算机理论及优化算法.



龚皓(1982－),男,硕士生,主要研究领域为网络拥塞控制.