

# 基于自适应加权平均的 TCP 友好拥塞控制机制\*

黄奎<sup>1,2+</sup>, 吴亦川<sup>1</sup>, 郑健平<sup>1</sup>, 吴志美<sup>1</sup>

<sup>1</sup>(中国科学院 软件研究所,北京 100080)

<sup>2</sup>(中国传媒大学 数字化工程中心,北京 100024)

## TCP-Friendly Congestion Control Mechanism Based on Adaptive Weighted Average

HUANG Kui<sup>1,2+</sup>, WU Yi-Chuan<sup>1</sup>, ZHENG Jian-Ping<sup>1</sup>, WU Zhi-Mei<sup>1</sup>

<sup>1</sup>(Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

<sup>2</sup>(Digital Engineering Center, Communication University of China, Beijing 100024, China)

+ Corresponding author: Phn: +86-10-65783391, E-mail: huangkui@ios.cn, http://www.iscas.ac.cn

Received 2004-07-01; Accepted 2005-01-07

Huang K, Wu YC, Zheng JP, Wu ZM. TCP-Friendly congestion control mechanism based on adaptive weighted average. *Journal of Software*, 2005,16(12):2124–2131. DOI: 10.1360/jos162124

**Abstract:** TCP-friendly congestion control is a key technique to guarantee the large-scale deployment of real-time streaming and multicast service in the Internet. On the basis of TCP emulation at receivers (TEAR) scheme, a new congestion control mechanism, called adaptive TCP-friendly congestion control (ATFCC), is proposed. In this scheme, the parameters of weighted average are dynamically adapted according to the type of packet loss and the duration of current congestion epoch. Simulation results show that ATFCC scheme outperforms TCP-Friendly rate control (TFRC) protocol in terms of rate smoothness and fairness to TCP flows.

**Key words:** congestion control; TCP-friendly; adaptive weighted average

**摘要:** TCP友好拥塞控制是保证实时媒体流和组播业务在 Internet 广泛应用的关键技术.基于收端 TCP 模拟方案 TEAR(TCP emulation at receivers),提出了一个根据丢包类型和当前拥塞周期的持续时间动态调整加权平均参数的拥塞控制机制,称为自适应 TCP 友好拥塞控制方案 ATFCC(adaptive TCP-Friendly congestion control).仿真结果表明,ATFCC 方案在速率平滑程度和 TCP 友好性方面的性能优于 TCP 友好速率控制协议 TFRC(TCP-Friendly rate control).

**关键词:** 拥塞控制;TCP 友好;自适应加权平均

中图法分类号: TP393 文献标识码: A

传输层控制协议 TCP<sup>[1]</sup>由源端根据接收端反馈调整发送速率,进行闭环控制,提高了 IP 网络的稳定性和健

\* Supported by the National Natural Science Foundation of China under Grant No.60272078 (国家自然科学基金)

作者简介: 黄奎(1977 - ),男,河南信阳人,博士,主要研究领域为 IPv6 组播,拥塞控制;吴亦川(1977 - ),男,博士,主要研究领域为无线网络和 IP 报头压缩;郑健平(1976 - ),男,博士,主要研究领域为 IPv6 组播和泛播;吴志美(1942 - ),男,研究员,博士生导师,CCF 高级会员,主要研究领域为宽带网络,多媒体通信.

壮性,但随着 IP 网络的发展和多种业务的融合,出现了越来越多的实时媒体业务和组播业务,为保证业务的实时性、简单性和接收质量的稳定性,它们通常不使用 TCP 协议,因此缺乏有效的拥塞控制机制。

近年来提出了多种用于这些业务的拥塞控制方案,这些方案一方面能够维持较为平滑的速率,响应网络拥塞;另一方面能够满足 TCP 友好要求.TCP 友好(TCP-friendly)是指在较长的一段时间内,平均吞吐量不大于相同条件下的 TCP 流<sup>[2]</sup>,以免影响 Internet 上的原有应用。

本文在 TEAR(TCP emulation at receivers)方案<sup>[3]</sup>的基础上,提出了自适应 TCP 友好拥塞控制方案 ATFCC (adaptive TCP-friendly congestion control).该方案根据丢包类型和拥塞周期的持续时间动态调整各个拥塞周期速率采样的加权平均参数.本文通过仿真分析了 ATFCC 方案的性能。

## 1 相关工作

已有的 TCP 友好拥塞控制方案主要包括 TFRC(TCP-Friendly rate control)<sup>[4]</sup>,TEAR,GAIMD(general additive increase multiplicative decrease)<sup>[5]</sup>和 SIMD(square increase multiplicative decrease)算法<sup>[6]</sup>。

在 TFRC 中,接收端采用加权平均的方法估计包丢失率,发送端根据包丢失率和 RTT(round trip time)采用修改后的 TCP 吞吐量公式<sup>[7]</sup>估计并调整会话速率.TFRC 的 TCP 友好程度受 TCP 吞吐量公式影响,在包丢失率较高时速率小于 TCP<sup>[3,8]</sup>.另外,文献[9]将 TCP 吞吐量作为丢包率的函数,分析了该函数在不同丢包率下的特性.由该文的分析可知,即使 RTT 不变,TCP 吞吐量公式也无法在各种丢包率下准确描述 TCP 的平均吞吐量。

TEAR 在收端模拟 TCP 协议,采用窗口/速率混合模式调整速率.为提高速率平滑性,首先计算拥塞周期的平均速率,并在多个拥塞周期上进行加权平均,以此调整发送速率,速率抖动较小.但由于 TEAR 在固定数目的拥塞周期内进行速率平均,在网络拥塞状况发生变化时,无法快速适应网络环境的变化.另外,TEAR 在包丢失率较高时吞吐量高于 TCP 流<sup>[8]</sup>。

GAIMD 基于文献[7]中的 TCP 吞吐量推导方式,给出了在考虑 RTO(retransmission timeout)时,为保证 TCP 友好,加性和乘性因子应满足的条件.SIMD 算法基于时间的平方增加拥塞窗口,提高了收敛到公平分配的速度.GAIMD 和 SIMD 是基于窗口的拥塞控制方案,数据包的突发传输容易造成网络节点的队列溢出,进而导致速率的抖动.同时,接收端的每包反馈可能导致反向路径的拥塞。

收端速率自适应方案 RAAR<sup>[10]</sup>结合 GAIMD 与 TEAR 的思想,在收端采用 GAIMD 算法和加权平均机制估计速率,进一步提高了速率平滑性.但由于 RAAR 方案对最近几个 round 而非拥塞周期的速率进行加权平均,引入了一个远小于 1 的修正因子,由文献[8]的分析可知,RAAR 的拥塞响应和可用带宽探测速度小于 TFRC 和 TEAR。

## 2 自适应 TCP 友好拥塞控制方案 ATFCC

### 2.1 基本思想

这里首先介绍 TEAR 的速率加权平均算法.该算法是对最近一定数量的拥塞周期的速率采样进行加权平均,过滤随机噪声,获得更为准确和平滑的速率估计.从本质上讲,加权平均算法是根据最近  $n$  个拥塞周期的速率信息来预测和估计下一时间的会话速率的.加权平均算法可表示为

$$\bar{T} = \frac{\sum_{i=1}^n T(i) w_n(i)}{\sum_{i=1}^n w_n(i)} \quad (1)$$

$n$  是进行加权平均的拥塞周期个数, $T(i)$ 表示最近第  $i$  个拥塞周期的平均速率.权重系数  $w_n(i)$ 如下式所示:

$$w_n(i) = \begin{cases} 1, & 1 \leq i \leq n/2 \\ \frac{n+1-i}{n/2+1}, & n/2 < i \leq n \end{cases} \quad (2)$$

文献[8]基于协方差 CoV(coefficient of variation)分别描述了速率平滑性和不同时间尺度上的流间公平性,

并分析了不同类型的协方差参数之间的对应关系.下面采用文献[8]的分析方法,给出  $n$  取不同值时,TEAR 方案的时间协方差.

根据文献[8]中的附录 D 可知,当参与加权平均的拥塞周期个数为  $n$  时,TEAR 方案平均每 RTT 的发送速率为  $T = \sum_{i=1}^n \frac{4w_n(i)}{3n} \times \left( \frac{W_i}{2} + \frac{W_{i-1}}{4} \right)$ , 则  $E\{T\} = \frac{3}{4} E\{W\}$ . 经过推导,可以得到发送速率的方差为

$$\text{Var}\{T\} = \left\{ \frac{1}{\left[ 3n \left( \frac{n}{2} + 1 \right) \right]^2} \left[ \sum_{k=0}^{\frac{n}{2}} (1+3k)^2 + \left( \frac{n}{2} - 1 \right) \left( \frac{3n}{2} + 3 \right)^2 + (n+2)^2 \right] \right\} \text{Var}\{W\} \quad (3)$$

由协方差的表达式,我们可以分别得到  $n=16,8,4,2$  时对应的时间协方差为 0.15,0.21,0.29 和 0.37.

从以上的理论分析可以看出,  $n$  值越大,时间协方差越小,速率平滑性越好.通常  $n$  的取值是快速响应网络状态变化和减小速率抖动之间的折中:  $n$  越大,速率变化更为平滑,但响应状态变化的能力下降.当网络拥塞状况发生明显变化时,以前拥塞周期的速率采样对于估计当前速率的意义不大,可以减少  $n$ ,提高响应网络环境变化的能力和 TCP 友好性;当网络处于较为稳定的状态时,增加  $n$ ,减小速率抖动.

因此,ATFCC 方案的核心就是自适应加权平均算法,即如何检测网络拥塞状况的变化,以及如何根据检测结果动态设置用于加权平均的拥塞周期个数  $n$ ,在 TCP 友好程度和速率平滑程度方面获得更好的性能.

## 2.2 自适应加权平均算法

为了在网络较为稳定时增加速率平滑性,ATFCC 方案缺省在 16 个拥塞周期进行加权平均,即  $n=16$ .

TCP 协议有两种类型的包丢失:一种为 RTO 类型的包丢失,另一种是 TDA (triple duplicate ACK) 对应的包丢失.发生 RTO 类型的包丢失通常表明网络拥塞较为严重,因此首先需要处理 RTO 类型的包丢失:在 ATFCC 方案中,当接收者检测到 RTO 类型的丢包时,立刻令  $n=2$ ,进行速率估计并反馈给源端. $n=2$  可以降低以前拥塞周期的速率信息对速率估计的影响,同时避免速率的过度抖动.

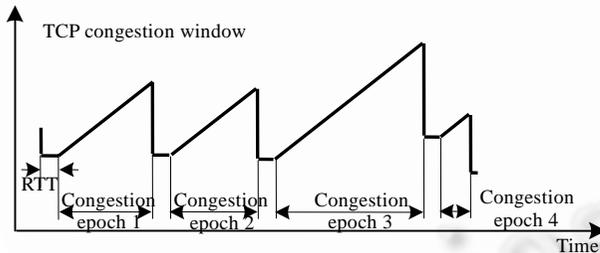


Fig.1 Evolution of TCP window with time during congestion avoidance phase

图 1 拥塞避免阶段 TCP 窗口随时间的变化

TDA 类型的包丢失说明仍有数据包到达接收端,网络拥塞程度较轻.图 1 描述了 TCP 拥塞避免阶段拥塞窗口随时间的变化情况.图 1 给出了本文的一个重要概念——拥塞周期,它是指前一次 TDA 类型的包丢失事件结束时到后一次包丢失事件开始时的时间间隔.

从图 1 可以看出,当网络环境较为稳定时,相邻两次丢包事件的间隔变化不大,如图 1 所示中的拥塞周期 2 和周期 1 之间的差别不大.当网络拥塞减轻时,下一次发生包丢失的时间将随之延后,对应拥塞周期的持续时间也将增加,如图 1 中的拥塞周期 3.反之,则拥塞周期减小,如拥塞周期 4.

与 TFRC 方案测量和计算包丢失率相比,测量拥塞周期的持续时间更为方便,实现更简单.另外,在理想的稳态环境下,拥塞周期与 TCP 平均速率存在相对应的关系.

在理想的稳态环境下,TCP 处于拥塞避免阶段,且每个拥塞周期中 TCP 窗口的最大值相同,设为  $W$ ,因此拥塞周期的持续时间保持不变,设为  $P$ ,RTT 值设为  $R$ .由图 1 可知,拥塞周期和丢包事件的持续时间之和为  $P+R$ ,这段时间内 TCP 源发出的数据包个数为  $\sum_{k=0}^{W/2} \left( \frac{W}{2} + k \right) = \left( \frac{W}{2} + 1 \right) \times \frac{3W}{4}$ .因此,TCP 平均速率为  $\frac{3W \times (W+2)}{8(P+R)}$ .根据 TCP

协议可知  $\frac{W}{2} = \frac{P}{R}$ ,通过推导,可以得到 TCP 平均速率  $T$  与拥塞周期持续时间和 RTT 的关系为  $T = \frac{3P}{2R^2}$  packets/s.

因此,ATFCC 方案使用拥塞周期持续时间的变化来推测网络环境的变化.接收端维护过去  $n$  个拥塞周期的持续时间,并对当前拥塞周期的持续时间(如果当前拥塞周期尚未结束,则将本次拥塞周期的起始时间到当前时

间的间隔作为拥塞周期持续时间)和最近  $n$  个周期持续时间的加权平均进行比较.如果当前拥塞周期没有结束,且持续时间大于以前  $n$  个拥塞周期持续时间的加权平均的 1.5 倍,这表示网络拥塞程度减轻,令  $n=8$ ,如果持续时间大于最近  $n$  个拥塞周期时间的加权平均的 2 倍,表明网络拥塞程度继续减轻,则将  $n$  设为 4.当持续时间大于加权平均的 4 倍时,设置  $n=2$ .

需要指出的是,最近  $n$  个拥塞周期持续时间的加权平均是在前一个拥塞周期结束时计算的,在当前拥塞周期结束之前,不再重新计算拥塞周期持续时间的加权平均,即  $n$  的变化在当前拥塞周期结束之前不会影响加权平均,避免了  $n$  的变化反向作用于比较结果,引起  $n$  的频繁变化.

当前拥塞周期结束时,如果持续时间小于以前  $n$  个周期持续时间的加权平均的 0.5 倍,说明拥塞程度加重,则将  $n$  设为 8.由于一个 RTT 内的包丢失通常是同一个拥塞事件引起的,在当前拥塞周期结束后,接收端过滤一个 RTT 内发生的包丢失.同时将 RTT 结束的时间作为下一个新的拥塞周期的起始时间.如果拥塞周期的持续时间小于 RTT,这说明出现了连续的丢包事件,网络拥塞较为严重,因此,设  $n=4$ ,以快速响应网络拥塞.

对于接收端,还可以通过监测相邻包的到达时间间隔来推测网络的拥塞程度,但由于包到达时间间隔受流量突发性、竞争流个数、随机包丢失等多种因素影响,难以正确反映网络的真实状态,因此,ATFCC 方案仅根据丢包类型和拥塞周期的持续时间来判断网络的拥塞程度,进而动态调整加权平均参数.

### 2.3 ATFCC方案的实现

与 TEAR 相同,ATFCC 方案在收端采用窗口/速率混合方式和加权平均机制估计会话速率,并通知源端调整发送速率.与 TEAR 不同的是,ATFCC 方案通过检测丢包类型和测量拥塞周期的持续时间来判断网络拥塞程度及其变化情况,动态调整参与加权平均的拥塞周期个数.下面详细介绍自适应加权平均算法的实现.

调整  $n$  值时,若仅保留最近  $n$  个周期的速率采样的历史记录,丢弃其他历史速率采样,可能产生边界效应,同时难以处理随网络状态的稳定  $n$  逐步增加的情况.因此引入一个权重系数的修正因子  $cf(i)$ ,并调整式(1)为

$$\bar{T} = \frac{\sum_{i=1}^n T(i) \cdot cf(i)w_n(i)}{\sum_{i=1}^n cf(i)w_n(i)} \quad (4)$$

在整个会话持续阶段, $n$  一直保持为 16,通过调整修正因子来间接达到调整  $n$  值的目的.会话初始建立时,令修正因子为 1,即  $cf(i)=1, 1 \leq i \leq 16$ .当收端由拥塞信息确定出合适的  $n$  值后,根据  $n$  值的不同更新修正因子,对应的修正因子为

$$cf(i) = \begin{cases} 1, & \text{when } i = 1 \\ 4x \text{ if } (cf(i) > 4x), & \text{when } i = 2 \\ 3x \text{ if } (cf(i) > 3x), & \text{when } 3 \leq i \leq 4 \\ 2x \text{ if } (cf(i) > 2x), & \text{when } 5 \leq i \leq 8 \\ x \text{ if } (cf(i) > x), & \text{when } 9 \leq i \leq 16 \end{cases} \quad (5)$$

之所以按照式(5)更新  $cf(i)$ ,是基于不同拥塞周期权重不同的思想,越近的周期对于估计下一时间的速率的重要性越大.本文将最近的 16 个拥塞周期分为最前面的 8 个、接着的 4 个、2 个周期以及最近的第 2 个拥塞周期,令它们的重要性之比为 1:2:3:4,同时保证当前拥塞周期的权重为 1.在更新修正因子时,为避免以前由于拥塞程度变化而设置的较小修正因子错误地增加,式(5)在更新修正因子时增加了一个约束条件,即仅在当前修正因子大于更新值时才进行更新.下面讨论如何设置变量  $x$ .

由式(2)可知,如果  $cf(i)=1, 1 \leq i \leq n$ ,所有权重系数之和为

$$\sum_{i=1}^n cf(i)w_n(i) = \sum_{i=1}^{n/2} w_n(i) + \sum_{i=n/2+1}^n w_n(i) = \frac{n}{2} + \frac{n}{4} = \frac{3n}{4} \quad (6)$$

因此在计算速率时,当前拥塞周期的归一化权重为  $4/3n$ .设当前应该设置的  $n$  值为  $k$ ,并设修正因子更新前为  $cf(i)=1, 1 \leq i \leq n$ ,为保证当前拥塞周期的权重在对  $cf(i)$  按式(5)更新后等于  $4/3k$ ,式(5)中的  $x$  需要满足下式:

$$\sum_{i=1}^{16} cf(i)w_{16}(i) = \sum_{i=9}^{16} x \times w_{16}(i) + \sum_{i=5}^8 2x \times w_{16}(i) + \sum_{i=3}^4 3x \times w_{16}(i) + 4x \times w_{16}(2) + w_{16}(1) = \frac{3k}{4} \quad (7)$$

根据式(2)和式(7),可以得出

$$x = \left( \frac{3}{4}k - 1 \right) / 22$$

在以后的每个拥塞周期结束时,对修正因子进行移位:

$$cf(1) = 1; cf(i+1) = cf(i), 1 \leq i \leq 15 \quad (8)$$

修正因子的移位保证在  $n$  值减少为  $k$  后,随网络状态的稳定、新的拥塞周期的权重保持为较大的值.自适应加权平均算法的具体实现可总结如下:

- (1) 初始化时令  $n=16$ ,修正因子为 1.
- (2) 如果出现 RTO 类型的包丢失,令  $x=1/44$ ,按式(5)更新  $cf(i)$ .
- (3) 如果当前拥塞周期尚未结束:
- (4) 若当前拥塞周期的持续时间大于最近 16 个拥塞周期的持续时间加权平均的 1.5 倍,令  $x=5/22$ ,按式(5)更新  $cf(i)$ ;
- (5) 如果当前拥塞周期的持续时间大于加权平均的 2.0 倍,令  $x=1/11$ ,按式(5)更新  $cf(i)$ ;
- (6) 如果当前拥塞周期的持续时间大于加权平均的 4.0 倍,令  $x=1/44$ ,按式(5)更新  $cf(i)$ ;
- (7) 如果包丢失类型为 TDA:
- (8) 如果本次拥塞周期的持续时间小于最近 16 个拥塞周期的持续时间加权平均的 0.5 倍,令  $x=5/22$ ,按式(5)更新  $cf(i)$ ;
- (9) 如果当前拥塞周期的持续时间小于 RTT,令  $x=1/11$ ,按式(5)更新  $cf(i)$ ;
- (10) 如果本次拥塞周期的持续时间大于加权平均的 1.5 倍以上,对修正因子进行移位.
- (11) 在当前拥塞周期结束时,更新拥塞周期持续时间的记录,计算拥塞周期持续时间的加权平均.

### 3 性能分析

本节分析 ATFCC 方案在不同网络环境中的性能,使用仿真软件 ns2<sup>[11]</sup>对 ATFCC,TFRC 和 TEAR 三种方案进行了以下几个方面的性能评估:(1) 协议间公平,即 TCP 友好;(2) 速率平滑性;(3) 协议内公平,即执行相同拥塞控制机制的多个会话公平占用带宽.另外,简单分析了 ATFCC 方案在动态网络环境中的性能.

#### 3.1 仿真设置

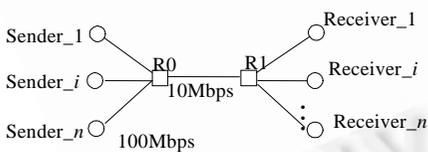


Fig.2 Simulation network topology

图 2 网络仿真拓扑

仿真使用如图 2 所示的单瓶颈链路拓扑评估不同 TCP 友好拥塞控制方案的性能,其中路由器 R0 与 R1 之间的瓶颈链路带宽为 10Mbps,其他链路的带宽为 100Mbps.各个链路的延时在不同的实验中取值不同.

与文献[3,4]相同,仿真中使用的 TCP 版本为 TCP SACK.仿真实验分别模拟了低复用和高复用网络环境,低复用网络环境用多个流竞争瓶颈链路带宽来模拟,高复用环境通过调整服从 Pareto

分布的 ON/OFF 背景流的个数,产生不同程度的网络拥塞.表 1 给出了 ns 仿真中使用的缺省参数.

Table 1 Default parameters used in simulation

表 1 仿真中的缺省参数

Packet size	Buffer size	ON time	OFF time	ON rate	Shape	Buffer management	TCP tick
1000 bytes	200 packets	1s	2s	0.5Mbps	1.5	Drop-tail	0.1

#### 3.2 稳态网络环境中的性能比较

本节主要分析稳态网络环境下不同方案在速率平滑性、协议内公平和 TCP 友好性方面的性能.我们采用文献[4]的性能评估参数及其仿真脚本进行仿真实验,文献[4]的性能评估参数有以下两个:

- (1) 速率采样的协方差,用于评估速率随时间的变化程度.首先在仿真中以一定的时间尺度进行速率采样,

并计算这些速率采样的协方差  $CoV = \frac{\sqrt{E\{rate^2(t)\} - E\{rate(t)\}^2}}{E\{rate(t)\}}$ , 式中  $E\{rate(t)\}$  表示一系列速率采样的平均,  $\sqrt{E\{rate^2(t)\} - E\{rate(t)\}^2}$  表示速率采样的标准偏差.

(2) 等价比,用于评估流间公平性.取两个流在一定时间尺度上速率采样的比值及其倒数的较小值,然后在整个仿真时间段中进行平均  $EqR = E\left\{\min\left(\frac{rate_1(t)}{rate_2(t)}, \frac{rate_2(t)}{rate_1(t)}\right)\right\}$ . 由于等价比只是衡量两个流之间的公平性,它可以同时用来评估协议内公平性和 TCP 友好性.

根据以上两个评估参数的定义,可知速率采样的协方差与文献[8]中的时间协方差的含义一致;而不同时间尺度上的等价比则可以用来衡量短时和长时公平协方差.仿真实验中,速率协方差的时间尺度为 0.2s,近似等于一个 RTT;而等价比的尺度为 10s(低复用网络环境)和 20s(高复用网络环境),以评估一段较长时间内的公平性.

在本节的仿真实验中,瓶颈链路延时为 50ms,其他链路的延时为 0ms~20ms 之间的随机值.在低复用网络环境中,选取不同的流数,且 TCP 流的个数与非 TCP 拥塞控制方案的流个数相同,分别令非 TCP 拥塞控制方案为 ATFCC,TFRC 和 TEAR,统计它们的速率协方差和等价比.在高复用的网络环境中,通过调整 ON/OFF 背景流的个数,得到不同的拥塞程度和包丢失率.除了背景流,网络中还有一个 TCP 流和一个 TCP 友好拥塞控制方案的流,TCP 友好拥塞控制方案分别为 ATFCC,TFRC 和 TEAR,统计速率协方差和等价比.

图 3 分别给出了不同方案在不同竞争流个数下的速率平滑性和等价比.从图 3(a)可以看出,3 个方案的速率协方差远小于 TCP 协议.在竞争流个数较小时,由于拥塞程度和包丢失率较低,大多数时间 ATFCC 的  $n$  为 16,因此 ATFCC 的速率协方差小于 TEAR 和 TFRC,即速率平滑性更好.当流个数增加时,拥塞程度和包丢失率随之增加,ATFCC 的  $n$  也会随之下降,当流总数达到 128 时,其速率协方差大于 TEAR,但仍小于 TFRC.图 3(b)的仿真结果证明速率平滑性和协议内公平之间存在正相关性<sup>[8]</sup>.图 3(c)说明,ATFCC 的 TCP 友好性好于 TFRC,而在流总数增加到 128 时,由于包丢失率的上升,ATFCC 的 TCP 友好性好于 TEAR.图 3 还说明 3 个拥塞控制方案和 TCP 协议的各项性能指标都随着竞争流个数,即包丢失率的增加而下降.

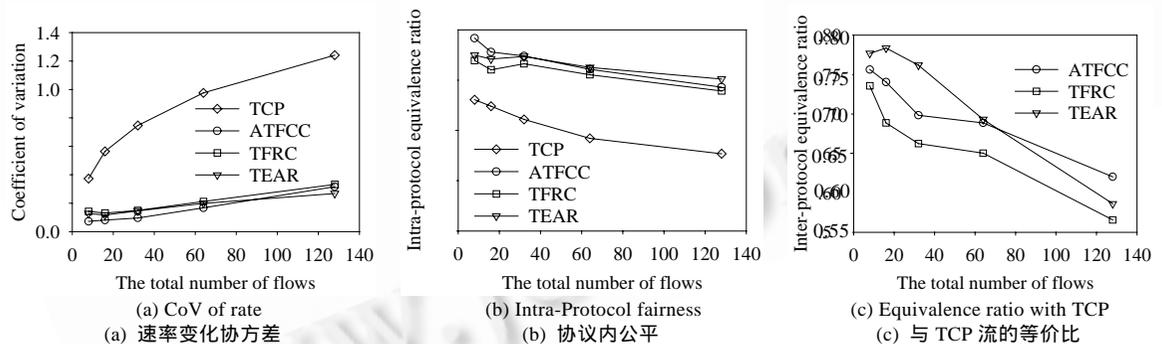


Fig.3 Performance of different TCP-friendly congestion control schemes in low multiplexing environments

图 3 低复用环境下不同 TCP 友好拥塞控制方案的性能

图 4 给出了高复用网络环境下的仿真结果,该图说明 ATFCC 的公平性和速率平滑性在大多数情况下优于 TFRC,而 ATFCC 在包丢失率较高时,速率平滑性差于 TEAR,但是 TCP 友好性好于 TEAR.为便于说明不同拥塞控制方案在不同包丢失率下的性能,图 4(a)显示了在不同的 ON/OFF 流数下的平均丢包率.

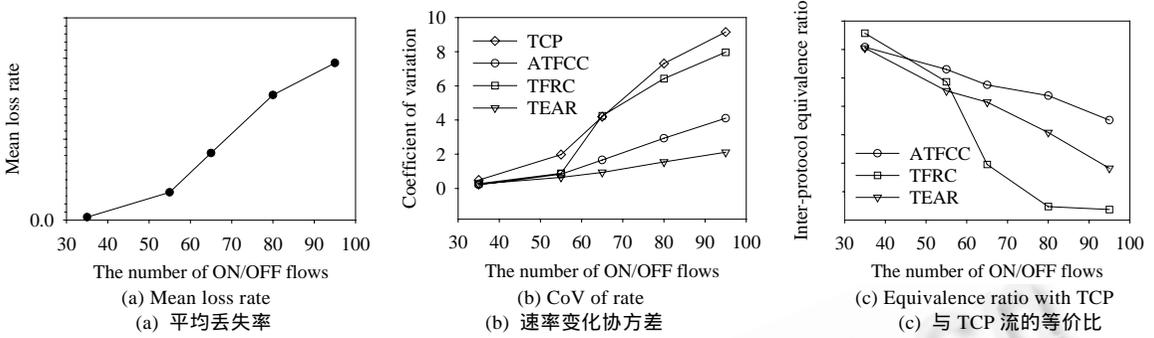


Fig.4 Performance of different TCP-friendly congestion control schemes in high multiplexing environments

图 4 高复用环境下不同 TCP 友好拥塞控制方案的性能

3.3 动态网络环境中的性能分析

网络环境的动态变化包括两种情况:(1) 网络拥塞减轻,可用带宽增加;(2) 网络拥塞加重,包丢失率增加.文献[4,8]使用每 RTT 速率增加的幅度描述 TCP 友好拥塞控制方案在网络拥塞减轻时的性能;使用出现持续拥塞(持续拥塞是指每个 RTT 至少丢失一个包)时速率减半所需的 RTT 个数来描述网络拥塞加重时的性能.

根据文献[4,8]的分析,TFRC 和 TEAR 速率增加的上限分别为每 RTT 增加 0.22packets/RTT 和 0.08packets / RTT.TFRC 以及 TEAR 在网络出现持续拥塞时,经过 5~6 个 RTT 后,速率减半.

由文献[8]的分析可知,ATFCC 当前拥塞周期的平均速率每 RTT 最多增加 0.5packets/RTT.由第 2.3 节可知,当前拥塞周期的权重为  $\frac{4}{3k}$ ,则 ATFCC 每 RTT 速率增加的上限为  $\frac{2}{3k}$  packets/RTT.因此,ATFCC 方案每 RTT 速率增加的上限为 0.33packets/RTT.当网络出现持续拥塞时,经过 4 个 RTT 后,历史记录变为 {W/16,W/8,W/4,W/2,3W/4,...,3W/4},此时计算的速率减半.因此理想情况下,ATFCC 方案速率减半只需 4~5 个 RTT.

仿真实验使用周期丢失模型比较不同方案的性能.仿真中只有一个流,分别为 ATFCC,TFRC 和 TEAR 流,瓶颈链路延时为 20ms,其他链路的延时为 10ms.包丢失率在第 20s 发生变化.图 5 显示了不同的 TCP 友好拥塞控制方案在包丢失率变化时的速率变化情况,其中纵坐标表示平均速率,用每 RTT 发送的包个数描述.

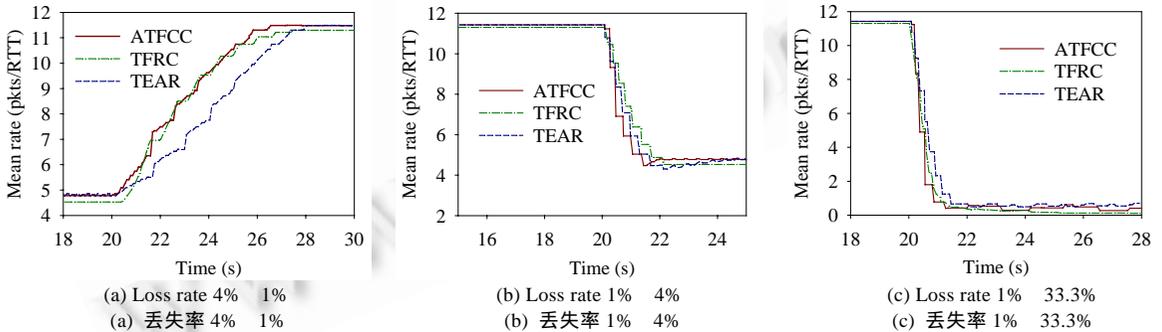


Fig.5 Performance of different TCP-friendly congestion control schemes in dynamic network environments

图 5 动态网络环境下不同 TCP 友好拥塞控制方案的性能

图 5(a)为包丢失率由 4%变为 1%时各个方案的速率增加速度.从图中可以看出,ATFCC 的速率增加速度与 TFRC 相近,高于 TEAR.图 5(b)描述了包丢失率由 1%变为 4%时各个方案的速率下降速度.可以看出,3 个方案的速率下降速度基本相同.图 5(c)显示了包丢失率由 1%变为 33.3%时各方案的速率下降速度,该图说明 ATFCC 和 TFRC 略快于 TEAR.一个值得注意的现象是在包丢失率增加到 33.3%时,TFRC 的平均速率几乎下降为 0,TEAR 的平均速率偏高,约为 0.7packets/RTT.而 ATFCC 的平均速率介于两者之间,约为 0.5packets/RTT.

## 4 结束语

本文提出了一个 TCP 友好拥塞控制方案 ATFCC. 该方案根据接收端的网络拥塞信息即丢包类型和拥塞周期的持续时间, 动态调整进行加权平均的拥塞周期个数. ns2 仿真结果表明, ATFCC 方案能够根据网络拥塞程度的变化而动态调整  $n$  值: 当包丢失率较高时, 采用较小的  $n$  值, 提高了 TCP 友好性; 而在包丢失率较低时, 采用较大的  $n$  值, 提高了速率平滑性. 对 ATFCC 方案在动态网络环境下的性能分析说明 ATFCC 具有较快的可用带宽探测速度和较短的拥塞响应时间.

### References:

- [1] Allman M, Paxson V, Stevens W. TCP congestion control. RFC2581, Internet Engineering Task Force, 1999.
- [2] Floyd S, Fall K. Promoting the use of end-to-end congestion control in the Internet. IEEE/ACM Trans. on Networking, 1999, 7(4):458-472.
- [3] Rhee I, Ozdemir V, Yi Y. TEAR: TCP emulation at receivers-flow control for multimedia streaming. Technical Report, North Carolina State University, 2000.
- [4] Floyd S, Handley M, Padhye J, Widmer J. Equation-Based congestion control for unicast applications. In: Proc. of the SIGCOMM 2000. Stockholm: ACM Press, 2000.43-56.
- [5] Yang YR, Lam SS. General AIMD congestion control. In: Proc. of the ICNP 2000. Osaka: IEEE Computer Society, 2000.187-198.
- [6] Jin S, Guo L, Matta I, Bestavros A. TCP-Friendly SIMD congestion control and its convergence behavior. In: Proc. of the ICNP 2001. Riverside: IEEE Computer Society, 2001. 156-164.
- [7] Padhye J, Firoiu V, Towsley D, Kurose J. Modeling TCP reno performance: A simple model and its empirical validation. IEEE/ACM Trans. on Networking, 2000,8(2):133-145.
- [8] Yang YR, Kim MS, Lam SS. Transient behaviors of TCP-friendly congestion control protocols. Computer Networks, 2003, 41(2):193-210.
- [9] Vojnovic M, Boudec J. On the long-run behavior of equation-based rate control. In: Proc. of the SIGCOMM 2002. Pittsburgh: ACM Press, 2002. 103-116.
- [10] Hu Y, Zhang GZ, Zhang GQ. RAAR: A TCP-friendly congestion control mechanism of transporting multimedia traffic in Internet. Chinese Journal of Computers, 2003,26(4):427-437 (in Chinese with English abstract).
- [11] UCN/LBL/VINT. The Network Simulator - ns2. <http://www.isi.edu/nsnam/ns>

### 附中文参考文献:

- [10] 胡严,张光昭,张国清.RAAR:多媒体流在 Internet 上传输的一种 TCP-friendly 拥塞控制机制.计算机学报,2003,26(4):427-437.