

一种在接收端实现的TCP-Friendly 拥塞控制机制

刘郁恒^{1,2}, 陈广文², 胡 严³, 张光昭²

(1. 广东省电信规划设计院, 广东广州 510630; 2. 中山大学电子与通信工程系, 广东广州 510275;
3. 中国科学院计算技术研究所, 北京 100080)

摘 要: 本文提出了一种基于速率的单播TCP Friendly 拥塞控制算法——RAAR(Rate Adaptation at Receivers) 控制机制。RAAR 是一种接收端的速率自适应算法, 它抛弃了每包反馈机制, 采用 GAIMD(General Additive Increase Multiplicative Decrease) 策略进行拥塞控制, 其主要控制操作由接收方完成。本文建立了简化的数学模型对其进行吞吐量的分析, 得到在 RAAR 中用于 TCP Friendly 的 GAIMD 拥塞控制中 α 与 β 的关系。通过与 TFRC 及 TEAR 这两种重要的 TCP Friendly 协议进行对比研究发现, RAAR 协议在对 TCP 协议的友好性, 协议内的公平性以及速率的平滑性等方面具有更好的综合性能。由于 RAAR 不需进行每包反馈, 且主要功能在接收方实现, 因此可方便地将该机制引入多媒体组播传输系统中。

关键词: RAAR; TCP Friendly; GAIMD; 拥塞控制

中图分类号: TP393 文献标识码: A 文章编号: 0372-2112 (2005) 05-0835-07

A TCP-Friendly Congestion Control Mechanism Implemented by Receivers

LIU Yurheng^{1,2}, CHEN Guangwen², HU Yan³, ZHANG Guangzhao²

(1. Guangdong Planning and Designing Institute of Telecommunications, Guangzhou, Guangdong 510630, China;
2. Dept. of Electronics & Communication Engineering, SUN Yat-sen University, Guangzhou, Guangdong 510275, China;
3. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: A new unicast rate based TCP-Friendly protocol called RAAR (Rate Adaptation at Receivers) is proposed. RAAR is a rate adaptive algorithm at receivers. It applies GAIMD (General Additive Increase Multiplicative Decrease) policy for congestion control and doesn't need to feed back per packet received. In RAAR, almost all complex operations of traffic control are carried out at receivers. A simple mathematical model is constructed to analyze the throughput of RAAR and get the relationship of α and β in RAAR for AIMD (Additive Increase Multiplicative Decrease) TCP Friendly congestion control. Comparing with two important TCP Friendly protocols, TFRC and TEAR, results are that RAAR has better performance in TCP Friendliness, intraprotocol fairness and smoothness. RAAR is also a promising avenue of development for congestion control of multicast traffic, since it is not a per packet acknowledgment and is mainly implemented at receivers.

Key words: RAAR; TCP-Friendly; GAIMD; congestion control

1 引言

众所周知, TCP 与 UDP 协议都不能很好地满足连续媒体流传输的需要。TCP 速率减半的拥塞退避机制容易引起多媒体流过大的速率波动, 其每包确认机制也是多媒体传输中所不希望的。而不具备拥塞退避机制的 UDP 流(被视为违规流)在拥塞的网络环境中将大量抢占具有拥塞退避机制的协议流的带宽, 同时自身的丢包也迅速增加, 并带来系统拥塞崩溃(Congestion Collapse)的潜在危险^[1]。为保证系统的稳定性和服务的公平性, 业界正在研究各种在拥塞条件下能对违规流进行严厉惩罚的队列管理机制(如 SFB^[2], RED PD^[3], CHOKe^[4],

SCHOKe^[5]等), 在这些系统中, 使用 UDP 的多媒体流在遭遇拥塞时, 将会遇到更为严重的性能衰减。

目前 Internet 中的传输业务主要是基于 TCP 的^[6], 随着多媒体实时业务在 Internet 的迅速增长, 研究一个适合于多媒体传输, 并具有拥塞退避机制, 能够与 TCP 协议公平分享带宽的传输协议, 成为了 Internet 传输的一个重要课题。如果一种通信协议与在同等条件下的 TCP 流具有近似相同的吞吐量, 则称这种协议是 TCP-Friendly 的。TCP-Friendly 协议的研究得到了业界的广泛重视, 目前提出的 TCP-Friendly 协议可以分为 2 类: 一种是基于 AIMD (Additive Increase Multiplicative Decrease) 的控制机制, 如 TEAR^[7], RAP^[8], LDA + ^[9] 等, 另一类是以

TFRC 为代表的基于数学模型的控制机制^[10]. TFRC 是现在比较成熟的由发送方控制的 TCP Friendly 控制协议, 它已成为了 IETF 工作组的正式草案^[11].

本文提出一种新的 TCP Friendly 算法, 称为 RAAR 算法 (Rate Adaptation at Receivers). RAAR 算法的主要设计目标包括: (1) 能与竞争的 TCP 流公平地分享带宽; (2) 流的发送速率平滑性好, 适合于传输多媒体流; (3) 考虑到在多媒体传输系统中, 发送方一般为多媒体服务器, 工作负荷比较重. 因此, 在 RAAR 中我们希望把复杂的拥塞检测与速率控制工作转移到接收端完成, 以减轻服务器的工作负担, 提高系统的扩展能力, 便于这个控制机制将来向组播系统扩展; (4) 针对多媒体流对丢失具有一定容忍能力的特点, 抛弃传统的每包反馈 (确认) 机制, 减少 ACK 分组的数量, 以适应反向通道带宽受限的非对称网络环境 (如 ADSL, HFC, 卫星通信网络等), 同时也可避免引入组播系统时产生反馈内暴 (feedback implosion) 现象.

2 RAAR 算法的描述

RAAR 本质上是一种 GAIMD (General Additive Increase Multiplicative Decrease)^[12] 的速率控制机制, 但速率控制操作主要由接收方完成. 接收方通过检测数据分组通过网络时的丢失状况, 以 GAIMD 的原则评估合适的发送速率, 并通过 ACK 分组控制发送方的发送速率, 以实现与 TCP 流公平分享带宽.

2.1 RAAR 的速率调节周期

在 RAAR 中, 其速率调整是按轮 (round) 进行的, 如图 1 所示. 当发送端接收到接收端反馈的 ACK 分组后, 则从中取出速率字

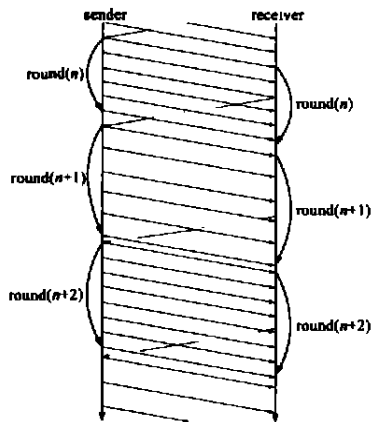


图 1 Rounds in RAAR

段中的速率, 马上按这个速率发送数据分组, 这时表明一个新的 round 开始; 直到发送端接收到新的 ACK 分组, 这个 round 才结束, 同时开始下一个 round. RAAR 协议就是由不断更迭的连续的 round 所构成的. 在以下两种情况之一出现时, 接收端需要立即发送 ACK 分组, 通知发送端更新发送速率: (1) 接收端检测到有分组丢失事件发生; (2) 接收端的 RTT 计时器超时 (RTT 计时器在收到第一个属于新的 round 的分组后启动).

在 ACK 分组无丢失的情况下, 发送方每隔一个时间段 $t < = 2RTT$ 就应该接收到一个 ACK 分组; 当网络高度拥塞时, 就会造成发送方在 $2RTT$ 内收不到正常的 ACK 分组. 为了保证在这种环境下发送方仍然能够及时地调整速率, 发送端需要具有超时减速功能. 当发送端在发送了一个新 round 的第一个分组后, 启动一个长度为 $2RTO$ 的定时器 T_{SND_RTO} , 如果发送端在 $2RTO$ 内仍然没有收到接收端返回的 ACK 分组, 则

在 T_{SND_RTO} 超时的时侯主动把自己的发送速率调整为原来的 β 倍 ($0 < \beta < 1$).

如果仅仅是由于 ACK 分组的偶然丢失而减少系统的发送速率则往往是不恰当的, 因此我们在接收端也使用了超时机制以补偿这种错误的速率调节. 当接收端发送了一个 ACK 分组后, 启动一个长度为 RTO 的定时器 T_{REV_RTO} . 如果接收端在 RTO 内能够接收到发送方在 T_{SND_RTO} 超时前发送的分组, 但既检测不到属于新 round 的分组, 也检测不到分组丢失事件, 则接收端认为它所发送的 ACK 分组发生丢失, 接收端在 T_{REV_RTO} 超时的时侯将重传上次所发送的 ACK 分组. 因此, 当 ACK 分组发生丢失时, 尽管发送端可能因为 T_{SND_RTO} 的超时而不恰当地临时降低发送速率. 但当发送端收到重传的 ACK 分组后, 其发送速率可以获得重新修正. 当接收端检测到属于新 round 的分组或者检测到分组丢失事件, 则取消马上取消 T_{SND_RTO} 定时器.

2.2 RAAR 的速率控制算法

2.2.1 慢启动阶段

与 TCP 类似, 当 RAAR 连接建立起来, 接收端接收到第一个数据分组后, 系统对必要的状态参数进行初始化, 并进入慢启动 (SLOW-START) 状态. 在此阶段中, 接收端每按序收到一个分组就把目前的速率 $rate_{-}$ 修改为: $rate_{-} + pktSize_{-} / RTT$, 并立即向发送方发送含有新发送速率信息的 ACK 分组, 以尽快地发现可合理利用的带宽. 如果在接收端探测到了一个丢分组事件, 则 RAAR 进入拥塞退避状态.

2.2.2 拥塞避免阶段

在拥塞退避阶段中, RAAR 通过 GAIMD 速率控制算法使系统的发送速率维持在一个在可合理利用带宽附近抖动的动态平衡状态.

RAAR 的核心速率调控机制就是在接收端模拟 TCP 所采用的 GAIMD 速率控制过程. 为了保持与 TCP 具有相同的速率增量, 在没有检测到分组丢失事件时, RAAR 每个 round 的速率增量为: $\frac{2\alpha \times pktSize_{-}}{RTT}$; 当 RAAR 的接收端检测到分组丢失事件后, 则把发送速率减少为原来的 β 倍, 其速率控制方程如下所示:

$$\begin{cases} rate_{-} \leftarrow rate_{-} + 2 * \alpha * \frac{pktSize_{-}}{RTT}, & \text{如果没有发生丢包事件} \\ rate_{-} \leftarrow rate_{-} * \beta, & \text{如果发生了丢包事件} \end{cases} \quad (1)$$

其中 α 与 β 的取值原则在第 3 节中进行论述.

2.3 分组丢失事件的判断

端系统对一个拥塞现象从探测到作出反应最少需要大约一个 RTT 的时间长度, 在此期间系统可能会出现多个分组丢失, 但这些分组丢失实际上是由同一个网络拥塞引起的. 因此以分组丢失事件来代替个别分组的丢失作为速率调整的依据显得更为合理.

当 RAAR 的接收端成功收到一个数据分组后, 它首先根据其序列号判断本次收到的数据分组与上次收到的数据分组之间是否出现分组的丢失. 如果出现了个别分组的丢失, 则需

要进一步判断这些分组的丢失是否属于一个新的丢失事件。如果同时满足下面的条件, 则认为是发生了新的分组丢失事件。

(1) 当前的 round 应该大于因上次分组丢失事件而进行了速率调整的 Round Number, 以表明这是经过系统的速率调整后所出现的新的分组丢失。

(2) 当前收到的分组不是本 round 中的第一个分组, 以表明本 round 确实发生了分组的丢失; 或者在前一个 round 没有检测到分组丢失事件, 而增加速率后在本 round 检测到分组丢失, 丢失的分组可能属于前一个 round(但在收到属于新 round 的分组后才检测到)。

当在上一个成功接收的分组和最新成功接收的分组之间发生了新的分组丢失事件时则马上通过 ACK 分组通知发送方减少发送速率; 如果一直没有检测到分组丢失事件, 则在长度为 RIT 的定时器 T_{fd} 超时的时候, 通知发送方增加发送速率。

3 RAAR 稳态的吞吐量近似模型

为简化对 RAAR 的稳态性能分析, 假设在稳态环境下 RIT 是常数且系统可用带宽不随时间变化, 所有分组大小相等, 设为 MTU 。RAAR 在稳态条件下的传输速率调节简化模型如图 2 所示。当系统没有分组丢失时, RAAR 周期性增加发送速率, 以探测系统的可用带宽; 当出现分组丢失事件后, 则把发送速率减少为原来的 β 倍。我们把相邻 2 次分组丢失事件发生的时间间隔称为分组丢失周期(Packet Loss Period), RAAR 的稳态过程可近似地简单看作是由首尾相连的一系列分组丢失周期所组成。设分组丢失周期为 T_{PLP} , 目标是求 RAAR 在一个分组丢失周期内单位时间的平均吞吐量表达式。

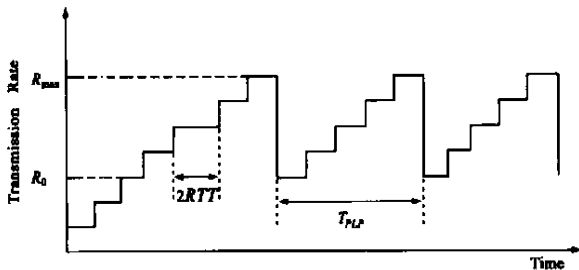


图 2 单个 RAAR 流的传输速率

如 2.1 节所述, 在无分组丢失的情况下, 每个 round 发送速率的增幅 Δ 为:

$$\Delta = 2^* \alpha * \frac{MTU}{RIT} \quad (2)$$

在一个分组丢失周期 (T_{PLP}) 中, 设 RAAR 最小发送速率为 R_0 , 最大发送速率为 R_{max} , 在稳态条件下有: $R_0 = \beta * R_{max}$, 则在一个 T_{PLP} 中所经历的 round 的个数 N 为:

$$N = \frac{R_{max} - R_0}{\Delta} = \frac{(1 - \beta) R_{max}}{\Delta} \quad (3)$$

在第 i 个 round 内, RAAR 所发送的分组数目 M_i 为:

$$M_i = \frac{2^* RIT^* R_i}{MTU} = \frac{2RIT^* (R_0 + i^* \Delta)}{MTU} \quad (4)$$

则 RAAR 在相邻 2 个分组丢失事件发生的时间间隔内 (即一

个 T_{PLP} 内) 可连续发送的分组数目 M 为:

$$M = \sum_{i=0}^N M_i = \frac{RIT}{\Delta^* MTU}^* R_{max}^2 * (1 + \beta)(1 - \beta) \quad (5)$$

设每个分组丢失的概率均为 p , 则连续 k 个分组无丢失的概率为:

$$P[M = k] = (1 - p)^{k-1} p, \quad k = 1, 2, \dots$$

$$E[M] = \sum_{k=1}^{\infty} (1 - p)^{k-1} p k = \frac{1}{p} \quad (6)$$

将式(3)与(6)代入式(5)得:

$$R_{max} = \sqrt{\frac{\Delta^* MTU}{RIT^* p^* (1 + \beta)(1 - \beta)}} = \frac{MTU}{RIT} * \sqrt{\frac{2\alpha}{p(1 + \beta)(1 - \beta)}} \quad (7)$$

在一个 T_{PLP} 期间, RAAR 的速率是从 R_0 等间隔地均匀增加到 R_{max} , 故在此期间 RAAR 在单位时间的平均吞吐量(即平均发送速率) T 为:

$$T = \frac{R_0 + R_{max}}{2} = \frac{(1 + \beta)^* R_{max}}{2} = \frac{MTU}{RIT \sqrt{p}} \sqrt{\frac{\alpha(1 + \beta)}{2(1 - \beta)}} \quad (8)$$

若按照当前 TCP 的算法, α 与 β 的取值分别为 $\alpha = 1, \beta = 0.5$ 。根据这组取值代入式(8)得:

$$T = \sqrt{\frac{3}{2}} \frac{MTU}{RIT \sqrt{p}} \sim 1.22 \frac{MTU}{\sqrt{p} * RIT} \quad (9)$$

式(9)与 Floyd 等人^[1]所得到的 TCP 的吞吐量公式完全相同。

为了在相同的网络条件下获得与 TCP 近似相同的吞吐量(即 TCP Friendly 的条件), (α, β) 这一常数对需要满足下式:

$$T_{\alpha, \beta}(MTU, RIT, p) = T_{1, \frac{1}{2}}(MTU, RIT, p) \quad (10)$$

由式(8)与(10)可得 α 与 β 满足 TCP Friendly 条件时的关系为:

$$\alpha = \frac{3(1 - \beta)}{1 + \beta} \quad (11)$$

由于 TCP 在拥塞时速率减半的策略将引起多媒体数据传输速率过大的抖动, 会明显降低用户端可察觉的质量(user perceived quality)^[9], 在本文中取 $\beta = 0.875$, 这是经过仿真试验后发现的一个整体性能较优的一个取值。而 α 则是满足式(11)的函数值。

我们通过仿真实验对式(8)的理论推导进行验证。仿真实验的拓扑结构如图 4 所示。N1 与 N2 为 2 个, 端系统节点, R1 为路由节点。链路 $N1 \rightarrow R1$ 的带宽为 100Mbps, 传播时延为 1ms; 链路 $R1 \rightarrow N2$ 为系统的瓶颈链路, 带宽为 10Mbps, 传播时延为 49ms。R1 采用 RED 的队列机制, 其 min-thresh 与 max-thresh 分别取 20 与 100 个分组, 在仿真实验中, 分组的大小固定为 1kbyte。在不考虑队列时延的情况

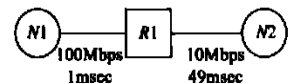


图 3 仿真实验的拓扑结构

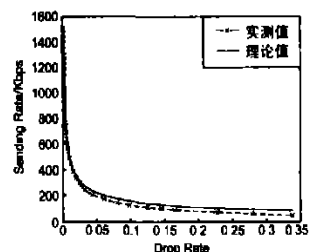


图 4 RAAR 的发送速率的理论值与实测值

下, 该系统的 RTT 约为 100ms. 系统中有 2 个协议流, 均是从 $N1$ 经过 $R1$ 流向 $N2$. 其中一个为 CBR 流, 其占用的带宽在 0Mbps 与 10Mbps 之间变动, 以控制系统的分组丢失率; 另一个为待观测的 RAAR 协议流, 我们测量在稳态条件下 RAAR 协议流的平均发送速率与分组丢失率之间的关系.

仿真结果表明 RAAR 发送速率的实测值与理论曲线基本一致, 说明上述的 RAAR 稳态模型基本正确. 在丢失率较高的网络环境中, RAAR 实际的发送速率要略低于理论值, 其原因主要有 2 个: (1) 在计算图 4 的理论值时, RTT 的取值并没有考虑队列时延的影响, 在拥塞程度较高的环境中, 队列时延变得不可忽略. (2) 在理论计算中没有考虑在发送端的超时减速机制, 在拥塞的环境中该机制在一定程度上造成实际的发送速率低于理论值.

式(9)实际上是各种 TCP 版本平均发送速率的上限^[13], 随着拥塞程度的加剧, TCP 的平均发送速率要低于式(9)的理论值. 为了改善在高拥塞环境中 RAAR 与 TCP 协议之间的公平性, 我们在式(1)中引入一个在 0~1 之间的调整系数 k_0 , 在无分组丢失的情况下, 每个 round 发送速率的调整方程为:

$$rate_{-} \leftarrow rate_{-} + 2 * k_0 * \alpha * \frac{pktSize_{-}}{RTT} \quad (12)$$

k_0 的取值范围一般在 [0.6, 0.8] 之间, 在仿真实验中, 我们取 $k_0 = 0.7$. 仿真实验表明, 引入 k_0 后, 对 RAAR 在低拥塞环境中对 TCP 协议的公平性影响不大, 但却大大改善了在高拥塞环境下对 TCP 协议的公平性.

4 RTT 与 RTO 的估算以及发送速率的平滑

在 RAAR 中, RTT 是一个十分重要的时间参量, 对于 RAAR 的性能有重要的影响, RTT 值的估算在接收端进行. 利用分组头部所携带的时间戳可以比较容易获得 RTT 的取样值. 我们采用 EWMA (Exponential Weighted Moving Average) 过滤器对 RTT 样值进行平滑得到 $SRIT$, 即: $SRIT(K+1) \leftarrow SRIT(K) + g * SERR(K+1)$, 其中, $SERR(K+1) \leftarrow sampleRTT - SRIT(K)$. 在接收端和发送端分别需要使用长度为 RTO 与 $2RTO$ 的定时器, 为了减少 RTT 的抖动误差造成定时器的错误超时触发, RTO 要取得比 $SRIT$ 的评估值大, 其计算公式可参照 Jacobson 算法^[13], 取: $RTO(K+1) \leftarrow SRIT(K+1) + f * SDEV(K+1)$, 其中, $SDEV(K+1) \leftarrow SDEV(K) + h * (|SERR(K+1)| - SDEV(K))$. RTO 的评估与计算也在接收端进行, 并通过 ACK 分组把 RTO 的值传递给发送方.

按照 Jacobson 的建议, 取 $g = 1/8$, $h = 1/4$, $f = 4$, 这样的取值也有利于操作系统进行快速移位运算.

为了减少突发噪声对系统的干扰, 使 RAAR 获得更为平滑的性能. 我们对 RAAR 的发送速率进行了平滑的处理. 接收端在发送 ACK 包之前, RAAR 首先对由 GAIMD 机制评估的发送速率 $rate_{-}$ 进行平滑, 它使用一个数组 $r(i)$ 记录最近的 n 个 $rate_{-}$ 历史值, 然后计算一个加权的平均速率 $Rate_{-}$:

$$rate_{-} = \frac{\sum_{i=1}^n w(i) \cdot r(i)}{\sum_{i=1}^n w(i)}$$

其中, $w(i)$ 是一个加权系数数组, 它的取值方法是:

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

平滑后的速率 $Rate_{-}$ 通过 ACK 包通知发送端, 需要注意的是: $Rate_{-}$ 是发送端实际的发送速率, 而 $rate_{-}$ 是由 GAIMD 计算后的速率, 在进行新一轮 GAIMD 计算时不能用 $Rate_{-}$ 来替代 $rate_{-}$, 即平滑后的速率 $Rate_{-}$ 用于实际的传输, 而不影响用于 GAIMD 计算的速率 $rate_{-}$. 如果不这样处理, 就会严重影响 RAAR 的 TCP Friendly 性能. 在仿真实验中, 取 $n = 8$.

5 RAAR 的性能仿真测试

我们以 NS2.1b8a^[16] 作为仿真平台对 RAAR 协议进行比较全面的性能仿真测试, 并且与 TFRC 和 TEAR 协议这两种重要的 TCP Friendly 协议的性能进行了对比. TFRC 协议是一种由发送方实现的基于数学模型的 TCP Friendly 控制机制, 它相对比较成熟, 它已被接纳为 IETF 的协议草案, 并向 IESG 推荐为 TCP Friendly 控制机制的正式标准. TEAR 是另外一种比较重要的在接收端实现的 TCP Friendly 拥塞控制算法, 它在接收端模拟了 TCP 协议的基于窗口的速率控制机制. 这一节将详细地介绍我们仿真实验的主要结果, 这些结果显示: 在对 TCP 的友好性, 协议内的公平性以及速率波动的平滑性等方面, RAAR 都具有比较好的性能.

5.1 仿真环境的配置

实验仿真的拓扑结构如图 5 所示. $R1$ 和 $R2$ 两个路由器之间的链路为瓶颈链路, 带宽为 15Mbps, 其他链路的带宽均为 100Mbps. 瓶颈链路的固定传输时延为 50ms, 两侧接入链路的传输时延之和为 5ms. $R1$ 是唯一可能拥塞的节点, 分组在 $R1$ 的排队时延的变动是引起 RTT 变化的决定因素, 因此 RTT 可以作为衡量系统拥塞程度的度量.

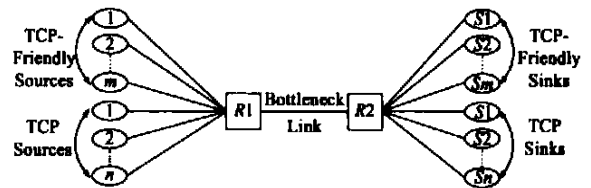


图 5 实验仿真的网络拓扑

系统中同时存在两种类型的数据流, 分别是 TCP 协议流与 TCP Friendly 协议流. TCP 协议有多个版本, 其中 SACK TCP^[14] 是 IETF 大力推荐的 TCP 版本, 它是在 Tahoe TCP 与 Reno TCP 的基础上发展起来的, 并且还在对其进行不断的扩展^[15]. 因此在本文中取 SACK TCP 作为 TCP 协议流的代表. 对于 TCP Friendly 协议流, 则分别取 RAAR, TFRC 与 TEAR 在同等网络条件下进行协议性能仿真测试.

本实验测量的是协议的稳态性能, 实验结果为仿真实验后 2/3 时间段内的统计数据. 所有协议流的分组大小都设为 1kbyte, 其开始发送数据时间都从 5s 到 10s 之间均匀分布的随机变量中提取. 一次仿真实验的时间长度为 6000s, 取 10 次实验的均值作为最后的实验结果.

5.2 对 TCP 协议的友好性(TCP Friendliness)

由于 RAAR 作为一种 TCP-Friendly 协议使用,这里我们首先对 RAAR 与 TCP(SACK-TCP) 协议公平分享带宽的能力进行研究,并与 TFRC 以及 TEAR 协议进行性能对比。

图(6)以协议间公平指数 F_P^{inter} 作为两种协议流吞吐量长时(long duration)公平性的度量,对比了 RAAR、TFRC、TEAR 这 3 种协议对 TCP 协议流的公平性。图 6(a)中, R1 与 R2 的瓶颈链路带宽为 15Mbps。X 轴表示 TCP-Friendly 与

SACK-TCP 流的总数,两者各占 50%, Y 轴为 TCP-Friendly 协议流与 TCP 流的协议间公平指数, F_P^{inter} 定义为^[17]:

$$F_P^{inter} = \frac{R_{\Pi \setminus P}}{R_P + R_{\Pi \setminus P}} = 1 - \frac{R_P}{R_P + R_{\Pi \setminus P}}$$

其中 R_P 为待评估的协议流 P 的归一化平均吞吐量, $R_{\Pi \setminus P}$ 为其他与之竞争带宽的协议流的归一化平均吞吐量,当 $F_P^{inter} = 0.5$ 时,我们称它是一种理想的公平(ideal fairness)。

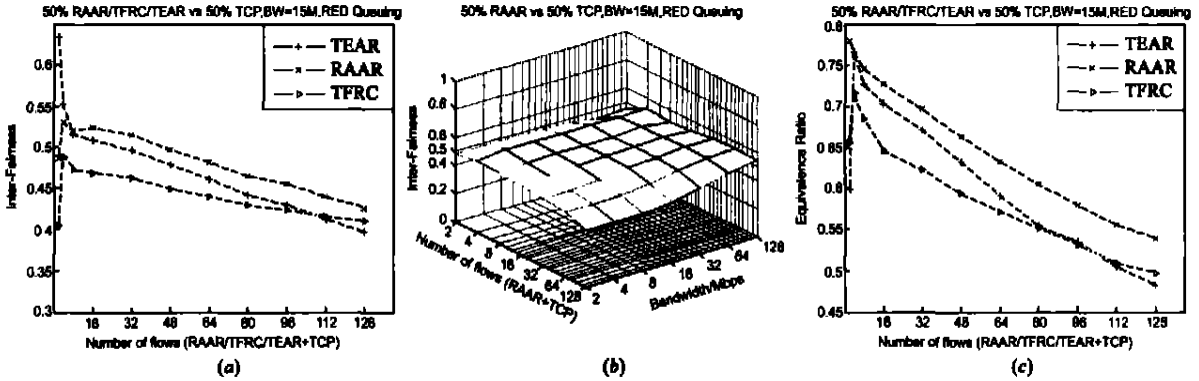


图 6 RAAR/TFRC/TEAR 对 TCP 协议的公平性

从协议间公平指数来看, RAAR 在一个相当大的参数范围内都可以与 TCP 协议较好地公平分享带宽。在竞争流的数目不多,网络拥塞较低时,RAAR 协议与 TFRC 以及 TEAR 协议在以协议间公平指数衡量的 TCP 友好性方面具有相当的性能;在竞争流数目增多,网络拥塞程度加剧的环境中,RAAR 对 TCP 的公平性略优于其他两种 TCP-Friendly 协议。

图 6(b) 示出了在不同的瓶颈链路带宽,不同数目竞争流的环境下,RAAR 对 TCP 的协议间公平指数 F_P^{inter} 。由图 6(b) 可以看出,相当大的测量范围内 RAAR 对 TCP 的公平指数都基本上落在 $z = 0.5$ 的理想公平平面附近,因此 RAAR 具有良好的 TCP 友好性。

F_P^{inter} 是从一个长时性的宏观角度对协议间的公平性进行评估,图 6(c) 则是利用两个协议流在较小测量时间粒度中的吞吐量等价比(Equivalence Ratio)值来衡量两个协议流(既可以是相同协议,也可以是不同协议)短时的吞吐量公平性。两个协议流 a 与 b 在给定时间尺度 δ 的吞吐量的等价函数定义为^[18]:

$$e_{\delta, a, b}(t) = \min \left(\frac{R_a, \delta(t)}{R_b, \delta(t)}, \frac{R_b, \delta(t)}{R_a, \delta(t)} \right)$$

其中 $R_a, \delta(t)$ 与 $R_b, \delta(t)$ 分别表示两个协议流 a 与 b 在时间间隔 δ 内的吞吐量。时间序列 $\{e_{\delta, a, b}(t_0 + i \cdot \delta)\}_{i=0}^n$ 中各元素的均值 $e_{\delta, a, b}$ 即为 a, b 两个协议流的吞吐量等价比, $e_{\delta, a, b} \in [0, 1]$ 。 $e_{\delta, a, b}$ 越接近 1, 表明两个协议流在时间尺度 δ 下越公平。图 6(c) 中的等价比值是 1 条 TCP-Friendly 协议流与各 TCP 竞争流的吞吐量等价比值取平均的结果,测量时间粒度 $\delta = 1s$, 从中我们可以看出 RAAR 协议在较小时间尺度(1s)下对 TCP 的公平性同样好于其他两种 TCP-Friendly 协议。

5.3 协议内的公平性(Intra protocol Fairness)

使用同种协议的各协议流之间的公平性(即 Intra protocol Fairness)也是评估一种传输协议性能的一个重要指标。图 7(a) 与图 7(b) 使用 $F_P^{equality}$ 评估在大时间尺度下的协议内公平性, $F_P^{equality}$ 定义为^[17]:

$$F_P^{equality} = \frac{\left(\sum_{f \in P} R_f \right)^2}{|P| \cdot \sum_{f \in P} R_f^2}$$

其中, R_f 为协议流 f 的归一化吞吐量, $|P|$ 为协议流的数目。 $F_P^{equality} \in [1/|P|, 1]$, 当各个流的吞吐量相等时, $F_P^{equality} = 1$ 。

图 7(c) 与图 7(d) 中使用 5.2 节的吞吐量等价比值

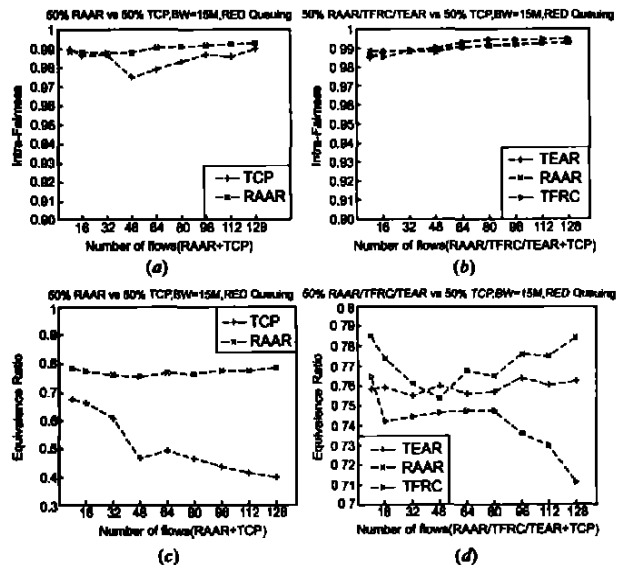


图 7 同种协议内的公平性

(Equivalence Ratio) 对在小时间尺度上的协议内公平性进行评估, 取测量的时间尺度 $\delta = 1s$. 图 7(c) 与图 7(d) 的等价比值是其中一条协议流与其余的同种协议流的等价比值取平均的结果.

图 7(a) 与图 7(c) 对比了 RAAR 协议流以及与之竞争带宽的 TCP 协议流的协议内公平性. 对于大时间尺度上的性能指标 ($F_{\beta}^{equidist}$), RAAR 协议的性能略优于对应的 TCP 协议; 对于在小时间尺度上的性能指标 (吞吐量等价值), RAAR 的性能明显优于 TCP 协议. 图 7(b) 与 7(d) 对比了 3 种 TCP-Friendly 协议的协议内公平性. 对于在大时间尺度上衡量的协议内公平指标, 三种协议都体现出良好性能, $F_{\beta}^{equidist}$ 都在 0.9 左右; 对于在小时间尺度 (1s) 下测量的协议内公平指标, RAAR 则具有较好的性能.

5.4 平滑性 (Smoothness)

由于 TCP-Friendly 协议主要应用于多媒体业务中, 接收端通常希望能够获得比较平滑的吞吐量特性. 我们用协议流吞吐量的方差系数 (Coefficient of Variation, CoV) 对协议流的平滑性进行评估. 协议流 f 吞吐量的 $CoV^{[17]}$ 定义为吞吐量标准方差 (standard deviation) 与其均值的比值, 即, $CoV_{f, \delta} =$

$$\frac{\sqrt{\frac{1}{T-t_0} \sum_{i=1}^{(T-t_0)/\delta} (R_f, \delta(t_0 + \delta \cdot i) - R_f)^2}}{R_f}. \text{ CoV 的值越小, 表示速率越平滑.}$$

在仿真实验中, 取测量的时间尺度 $\delta = 1s$.

从图 8(a) 对比了 RAAR 协议流以及与之竞争带宽的 TCP 协议流的速率平滑性. 从仿真结果看, RAAR 的平滑性远远优于 TCP 协议, 因此, RAAR 协议更适合于多媒体业务流的传输. 图 8b 对比了 3 种 TCP-Friendly 协议的速率平滑性. 在测量的参数范围内, RAAR 的速率平滑性优于 TFRC, 在带宽竞争激烈, 系统比较拥塞的环境下表现得更为明显. 而在测量的参数范围内, RAAR 的速率平滑性与 TEAR 大致相当.

6 结论

本文提出了一种基于速率的单播 TCP-Friendly 拥塞控制算法——RAAR 控制机制. RAAR 是一种接收端的速率自适应算法, 它抛弃了每包反馈机制, 采用 GAIMD 的策略进行拥塞控制, 其主要的控制操作由接收方来完成. 我们建立了简化的数学模型对其进行吞吐量的分析, 得到在 RAAR 用于 TCP-Friendly 的 AIMD 拥塞控制中的 α 与 β 的关系. 通过与其他两种主要的 TCP-Friendly 协议进行对比研究发现, RAAR 协议在对 TCP 协议的友好性, 协议内的公平性以及速率的平滑性等方面具有更好的综合性能.

考虑到在多媒体应用中, 发送端一般为负荷较重的多媒体服务器, 因此 RAAR 中复杂的拥塞控制机制都在接收端完成, 大大减轻发送服务器的负担, 因而与 TFRC 相比较, 更有利于提高系统的扩展性能. 虽然 TEAR 的主要流控功能也是主要在接收方完成, 但是它的接收方需要维护一个具有 7 个状态的有限状态机, 实现比较复杂, 影响了它的进一步推广. RAAR 的实现机制比 TFRC 与 TEAR 简单, 并且抛弃了每包确认的机制, 从而大大减少了反馈包发送的数量, 特别适用于象

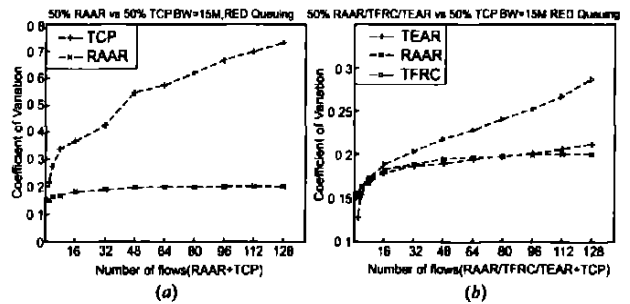


图 8 协议的平滑性

ADSL, HFC, 卫星系统, 无线网络等非对称的通信信道. 这种反馈机制有效地防止了反馈内爆 (feedback implosion) 的发生, 便于将来把这种机制应用到组播系统中.

使用 RAAR 协议构造一个实际的应用系统, 并在真实的网络环境下进行性能测试, 以及把 RAAR 控制机制扩展到组播应用领域是我们下一步的主要工作.

参考文献:

- [1] Floyd S, et al. Promoting the use of end-to-end congestion control [J]. IEEE/ACM Transactions on Networking, 1999, 17(4): 458-472.
- [2] Feng W, et al. Stochastic fair blue: A queue management algorithm for enforcing fairness [A]. Proceedings of IEEE INFOCOM 2001 [C]. Anchorage, Alaska, USA, 2001. 1520-1529.
- [3] Mahajan R, et al. Controlling high bandwidth flows at the congested router [A]. Proceedings of ACM 9th International Conference on Network Protocols (ICNP) [C]. Riverside, California, USA, 2001. 192-201.
- [4] Pan R, et al. A stateless active queue management scheme for approximating fair bandwidth allocation [A]. Proceedings of IEEE INFOCOM 2000 [C]. Tel Aviv, Israel, 2000. 942-951.
- [5] 胡严. 因特网中多媒体流传输的拥塞控制协议研究 [D]. 广州: 中山大学博士学位论文, 2001. 10.
- [6] Yang Y R, et al. Transient behaviors of TCP-Friendly congestion control protocols [R]. Technical Report TR 2000-14, Networking Research Laboratory, Department of Computer Sciences, the University of Texas at Austin, 2000.
- [7] Rhee I, et al. TEAR: TCP emulation at receivers - flow control for multimedia streaming [R]. North Carolina State University, 2000.
- [8] Rejaie R, et al. RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet [A]. Proceedings of IEEE INFOCOM 1999 [C]. New York, USA, 1999. 1337-1345.
- [9] Sisalan D, et al. LDA+: A TCP-Friendly adaptation scheme for multimedia communication [A]. Proceedings of IEEE International Conference on Multimedia and Expo (ICME 2000) [C]. New York, USA, 2000. 1619-1622.
- [10] Floyd S, et al. Equation-based congestion control for unicast applications [J]. Computer Communication Review, 2000, 30(4): 43-56.
- [11] Handley M, et al. TCP-Friendly Rate Control (TFRC): Protocol Specification [S]. IETF RFC 3448, 2003.
- [12] Yang Y R, et al. General AIMD congestion control [A]. Proceedings of ICNP 2000 [C]. Osaka, Japan, 2000. 187-198.
- [13] Jacobson V. Congestion avoidance and control [A]. Proceedings of the

Sigcomm' 88 Symposium in Stanford [C]. CA, August, 1988. 314 – 329.

- [14] Mathis M, et al. TCP selective acknowledgement options [S]. RFC 2018, 1996.
- [15] Floyd S, et al. An extension to the selective acknowledgement (SACK) option for TCP [S]. RFC 2883, 2000.
- [16] UC Berkeley, LBL, USC/ ISI, Xerox PARC. The ns Manual (formerly ns Notes and Documentation) [O B / O L]. <http://www.isi.edu/nsnan/ns/ns-documentation.html>, 2001.
- [17] Widmer J. Equation based congestion control [D]. Dept. of Mathematics and Computer Science, University of Mannheim, 2000.
- [18] Padhye J. Towards a comprehensive congestion control framework for continuous media flows in best effort networks [D]. University of Massachusetts Amherst, 2000.

作者简介:



刘郁恒 男, 1975 年生, 博士, 研究方向为多媒体网络通信技术与流量控制技术, E-mail: liuyh@gdi.com.

陈广文 男, 1977 年生, 硕士, 研究方向为多媒体网络通信, E-mail: unique@21cn.com.

胡 严 男, 1970 年生, 博士, 研究方向为网络拥塞控制技术, E-mail: hu_yan@263.net.

张光昭 男, 1938 年生, 博士生导师, 研究方向为计算机通信网络, isszgz@zsu.edu.cn.