

# 无线网络中基于流媒体传输的自适应 TFRC 机制<sup>①</sup>

王万良, 张小玮, 姚信威, 岑跃峰

(浙江工业大学 信息工程学院, 杭州 310023)

**摘要:** 在无线网络高误码率的环境下, 经典 TFRC 机制会将无线误码丢包误认为拥塞丢包, 导致吞吐量过度降低. 针对无线网络实时流媒体业务的传输控制问题, 提出了一种改进型动态自适应 TFRC 机制(Adaptive-TFRC). 它在接收端利用丢包区分参数来真实反映网络的状态(即拥塞或者误码), 然后反馈至发送端, 同时对经典 TFRC 机制的吞吐量模型公式进行改进, 最终能够根据实时网络条件动态自适应地调节传输速率. 仿真结果表明, Adaptive-TFRC 机制能够有效地提高网络吞吐量, 降低实时业务流的延时抖动, 同时能够进一步改善 TCP 业务的友好性传输, 从而保证无线网络实时流媒体的服务质量.

**关键词:** TFRC 机制; 单向传输时延; Adaptive-TFRC; 无线网络; 流媒体

## Adaptive Mechanism of TFRC for Streaming Media Service's Transmission in Wireless Network

WANG Wan-Liang, ZHANG Xiao-Wei, YAO Xin-Wei, CEN Yue-Feng

(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

**Abstract:** In the high error rate wireless network environment, packet loss caused by wireless error will be misunderstood as congestion packet loss by the classic mechanism of TFRC, leading to lower throughput. To solve the transmission control problem of the wireless network for real-time streaming media service, the paper proposed a mechanism called Adaptive-TFRC which was an improved mechanism of TFRC, which can reflect the true state of the network by using loss differentiation parameter of the receiving end, then the value of parameter would be sent back to the sending end, also we have improved the throughput model formula of the classic mechanism of TFRC in this paper, finally it can realize the dynamic adaptation of transmission rate. The simulation result shows that Adaptive-TFRC can greatly improve the throughput, reduce the delay jitter of real-time traffic, and it possesses TCP friendly characteristic very well, thus guaranteeing service quality of the wireless network real-time streaming media.

**Key words:** mechanism of TFRC; one-way delay; Adaptive-TFRC; wireless network; streaming media

随着无线网络技术的发展和多媒体技术的不断成熟, 无线网络中涌入了大量的流媒体应用, 如 IP 电话、视频点播、视频会议、远程教育等. 流媒体数据流占据着 Internet 整体流量中相当大的部分, 并且容易引起网络过载, 造成网络拥塞甚至网络崩溃. 因此, 流媒体数据传输给网络带来了巨大挑战, 已经成为近年来网络研究热点之一<sup>[1,2]</sup>.

流媒体业务对于传输数据的 QoS 具有一定要求, 同时还要保持对 TCP 数据流的友好性. 目前, 端对端

的控制协议主要是 TCP 和 UDP 两种. TCP 由于具有网络拥塞控制和重传机制使得传输时延比较大而不适合于多媒体流的传输. 当网络处于拥塞状态时, TCP 的拥塞退避机制会引起多媒体流的过大的速率波动. 因此大部分的流媒体网络传输系统都采用 UDP 作为传输协议. UDP 是一种无连接的非可靠传输协议, 比 TCP 更适合传输流媒体, 然而 UDP 是一种“尽力而为”的服务, 并不提供流量控制和拥塞控制机制, 它并不会根据网络的状况来调整发送端的发送速率, 如果直

<sup>①</sup> 基金项目:国家自然科学基金(61070043)

收稿时间:2012-12-24;收到修改稿时间:2013-01-28

接使用 UDP 作为传输协议,可能会给整个无线网络带来网络资源分配严重不公平的问题<sup>[3,4]</sup>.例如当大量的 UDP 流和 TCP 流共享网络带宽时,UDP 流将会不公平的获得大量带宽,会加重网络拥塞程度,甚至导致网络拥塞崩溃.所以传统的 TCP 和 UDP 都不完全适用于流媒体的传输.

目前流媒体传输大多数采用 TCP 友好速率控制协议(TFRC: TCP-friendly rate control protocol)<sup>[5]</sup>.TFRC 最早是应用于有线网络,但在无线网络中的性能比较差,因为在无线网络中的丢包是由于链路误码和网络拥塞造成的,而 TFRC 的接收端把所有的丢包归咎为拥塞丢包,从而 TFRC 发送端盲目地减小数据流的发送速率.因此,本文重点研究了 TFRC 控制机制,针对无线网络环境,提出了一种改进的 TFRC 机制 Adaptive-TFRC,使其适用于无线网络环境下的流媒体的传输,并且具有动态自适应的速率调节功能.

## 1 相关工作

TFRC 是一种基于公式控制发送速率的端到端单播 TCP 友好协议,IETF 于 2003 年正式颁布了 TFRC 协议的 RFC 文档,近年来也受到了国内外研究学者的广泛关注与深入研究.

朱晓亮<sup>[6]</sup>等人采用 IEEE802.11 构建 WSN,推导满足重传时延最小的传输层包长调整方程和结合 TFRC 建立 WSN 实时媒体传输的速率控制机制(WSNRC),该机制能够改善降低丢包率、平均传输时延和具备节点间的公平性特点,但是整个网络的吞吐量性能并未提高.姜明<sup>[7]</sup>等人通过对 TFRC 速率计算公式中丢包率的不同幂级项引入权重系数,降低网络拥塞严重时的发送速率,增加网络拥塞较轻时的发送速率,从而降低了网络拥塞程度对 TFRC 流传输速率的影响.王东<sup>[8]</sup>等人对 TFRC 进行改进,提出一种基于多路复用的 TCP 友好速率控制算法-MTFRC.该算法在保持 TFRC 较好速率平滑性的同时,增加了对多媒体流最低速率阈值限定特性及多路复用链路环境的考虑,但是同样整个网络的吞吐量性能并未得到改善.

Song Cen<sup>[9]</sup>等人提出了一种端到端的丢包区分算法 ZigZag 以及由 ZigZag、Biaz、Spike 三种丢包区分算法组成的丢包区分算法 ZBS, Spike、Biaz、ZigZag 在不同的网络环境下性能各不相同,而 ZBS 算法则根据不同的网络环境在这三种算法中进行切换,然

而 ZBS 的高误分类比率使得 ZBS 并不能高效地利用无线带宽.Chen<sup>[10]</sup>等人提出了 MULTFRC 模型,该模型是通过增加 TFRC 流的数量来加大无线信道的利用率.Shen<sup>[11]</sup>等人根据不同的情况下 TFRC 的性能,制定一个在无线衰减信道上的离散时间马尔可夫评估模型.

国内外的学者对于 TFRC 机制的研究最终大多数能够实现协议间的公平性,并且使 TFRC 业务流保持较好的速率平滑性,但是对于无线网络中的丢包是由于网络拥塞还是链路误码引起的未能准确有效地区分.本文提出的一种改进的自适应 TFRC 机制(Adaptive-TFRC),该机制可以利用丢包区分参数来有效准确地反映网络的状态(即拥塞或者误码),从而根据实时网络条件动态自适应地调节传输速率,最终保证无线网络流媒体业务传输数据的服务质量.

## 2 TFRC分析

### 2.1 TFRC 控制算法工作原理

TFRC 是由 S.Floyd 等人提出的一种主要用于流媒体传输的 TCP-Friendly 拥塞控制协议.为了使 TFRC 流与 TCP 流在相同的网络环境下实现带宽的公平,达到与 TCP 流的友好性,同时也使 TFRC 的发送速率具有较小的波动,TFRC 的吞吐量计算公式采用的是 TCP Reno 吞吐量模型计算公式<sup>[12]</sup>,如式(1)所示:

$$X = \frac{S}{R \sqrt{\frac{2bp}{3} + t_{RTO}} \cdot \left( 3 \sqrt{\frac{3bp}{8}} \right) p (1 + 32p^2)} \quad (1)$$

式(1)中  $X$  为发送端发送速率,  $S$  是数据包的大小,  $R$  是往返时延 RTT,  $t_{RTO}$  是重传超时时间,一般取  $t_{RTO} = 4 * RTT$ ;在这个公式中,参数  $R$  和  $b$  起着决定性的作用,  $b$  在 TCP 没有确认机制的情况下取值为 1;  $p$  是丢包事件率,即丢失事件的数量与所传输的分组数量的比值.

TFRC 控制算法的关键在于如何获取往返时延 RTT 以及丢包事件率  $p$ ,其中  $p$  在控制端通过计算而得到,为一定时间内平均丢包事件间隔的倒数,  $p$  的计算公式如下:

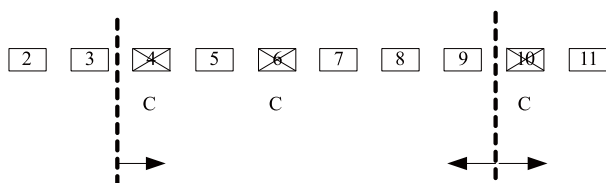
$$p = \sum_{i=0}^n \omega_i / \sum_{i=0}^n C_i \omega_i \quad (2)$$

式(2)中,  $C_i$  为最近的第  $i$  个丢包间隔,两个连续分组丢包事件的第一个丢包分组序列号之差被定义

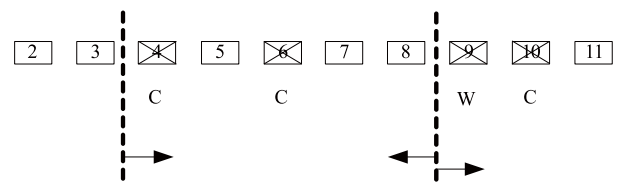
为丢包事件间隔,它指的是两次丢包事件之间数据包的数量.平均丢失间隔为对最近的  $n$  个丢失事件间隔进行加权平均所获得, $n$  值的大小决定了 TFRC 对拥塞程度变化时的响应速度.为了与 TCP 进行公平的竞争,TFRC 推荐采用, $n=8$  作为最好的一个平衡值. $\omega_i$  为对应的权重, $\omega_0 \sim \omega_7$  的取值分别为 1、1、1、1、0.8、0.6、0.4、0.2.从式(2)中可以看出,丢包事件间隔的计算直接影响着丢包事件率的计算,而丢包事件间隔通过计算两个连续分组丢包事件的第一个丢包分组序号之差来获得,与分组自身的序号大小并无直接的关系.

### 2.2 TFRC 在无线网络中应用的不足

传统的 TFRC 是将丢包作为拥塞信号的,在稳定的网络环境中表现良好,并且可被广泛接受但是在网络环境不稳定的情况下,由于 TFRC 的吞吐量计算公式仅仅根据丢包率和往返时延来更新吞吐量的估值,TFRC 并不能很好地适应网络的变化.在有线网络中丢包事件主要由于网络拥塞引起的,因此 TFRC 可以得到很有效的应用,其丢包事件间隔示意图如图 1(a)所示.但是当直接应用于存在无线链路的网络中时,由于存在无线误码,接收端误将无线误码丢包当作拥塞丢包,使得丢包事件的判断受到影响,从而盲目地降低速率,如图 1(b)所示.图中 C 代表拥塞丢包,W 代表无线误码丢包,由于存在无线误码丢包,丢包事件间隔由原来的 6(10-4)减小为 5(9-4),由式(1)和(2)可知这样会造成发送速率突然减小,不能保证实时数据传输的平滑性要求,网络吞吐量也会下降,同时当计算出的发送速率小于最低发送速率时,会以最低发送速率的速率来发送数据包,造成无效传输,一定程度上使得网络带宽利用率降低,从而导致流媒体的传输播放质量下降.因此,需要对现有的传输协议进行改进,改进目标有两个,包括可以有效地区分拥塞丢包和链路误码丢包以及能够有效地保障流媒体传输应用的服务质量,这是当前拥塞控制领域的一个热点.



(a) 不存在无线丢包的情况



(b) 存在无线丢包的情况

图 1 丢包事件间隔示意图

## 3 TFRC算法的改进

### 3.1 适用于无线网络环境的改进的 TFRC 吞吐量模型公式

TFRC 机制利用 TCP Reno 的吞吐量模型计算公式来调整发送速率,所以 TFRC 机制具有较好的 TCP 友好特性.但是由于无线网络具有传输信号容易受到外界干扰、时延抖动比较大等缺点,使得无线链路具有很高的误码率,这就使得无线信道中的丢包原因分为无线误码丢包和拥塞丢包,而 TFRC 无法区分无线误码丢包和拥塞丢包,当网络发生丢包时,发送端会盲目地过度降低发送速率,从而使得网络吞吐量大大降低.并且当网络环境处于较高网络拥塞的情况时,整个网络的吞吐量会严重下降,为了改进在高网络拥塞状态时的 TFRC 吞吐量下降的问题,我们以 TFRC 的吞吐量模型公式为原型,在它的基础上作修改,在公式(1)的后半部分除以常数  $m$ ,在丢包率  $p$  较低的情况时,即网络处于非拥塞状态时,公式(1)后半部分几乎可以忽略不计,从而提高了发送端的发送速率,在丢包率  $p$  比较高的情况时,即网络处于拥塞或者高拥塞状态时,后半部分加入了常数  $m$  可以使得 TFRC 机制能够在更广的拥塞度范围内保持较好的 TCP 友好性,这是一种适合于无线环境传输的吞吐量模型公式,式(3)给出改进的 TFRC 吞吐量模型公式:

$$X' = \frac{S}{R \sqrt{\frac{2bp}{3} + \frac{1}{m} t_{RTO} \cdot \left( 3 \sqrt{\frac{3bp}{8}} \right) p (1 + 32p^2)}} \quad (3)$$

其中  $X'$  为发送端发送速率, $S$  是数据包的大小, $R$  是往返时延 RTT,  $t_{RTO}$  是重传超时时间,一般取  $t_{RTO} = 4 * RTT$ ;  $b$  取 1;  $p$  是无线环境下更新之后的丢包事件率;  $m$  为常数,实验证明  $m$  取 1.5 时,TFRC 可以能够很好地保持对 TCP 的友好性.

### 3.2 改进型 Adaptive-TFRC 的算法思想

网络状况的不同可以通过时延和时延抖动来判别,

很多针对无线网络的 TCP 友好拥塞控制算法在判断网络状态时, 都是利用单项时延  $ROTT$  来作为网络拥塞的信号指示, 通过  $ROTT$  的值来确定网络到底处于什么状态. 如果网络处于拥塞的状况, 由于网络的整体容量具有局限性, 并且不断有数据进入网络中, 导致网络中拥塞程度会变得更加严重, 从而使得网络分组到达接收端所花费的时间比较长, 单向时延  $ROTT$  值也会相应地增大. 但是如果网络丢包是由无线链路误码所引起时, 此时  $ROTT$  值会较小, 因此, 端对端传输中  $ROTT$  值的大小变化可以真实地反映出网络所处的状态. 在  $ROTT$  值比较小的情况下, 如果检测到网络丢包是由无线链路误码所引起的, 此时发送端不需要降低发送速率. 反之, 当判断出网络中丢包是由网络拥塞所引起的, 此时发送端必须降低其发送速率.

$ROTT$  在消除链路层差错控制重传时会产生抖动, 从而影响到区分丢包原因的准确性, 本文提出的 Adaptive-TFRC 是对相邻的 8 个数据包的  $ROTT$  值进行方差运算, 实验表明采用相邻的 8 个数据包进行方差运算可以更好地反映当前的网络状况, 根据方差值符合的判定条件进行相应的计算, 长期样本加权平均值  $ROTTMEAN$  将会进行相应的计算. 但是差错控制对无线链路误码的反应会产生乱序包, 对于乱序包, Adaptive-TFRC 将不会用来计算  $ROTT$  的平均值以及统计长期样本均值  $ROTTMEAN$  的样本, 从而有效地提高算法的准确性.

Adaptive-TFRC 算法中, 首先要计算出数据包的单向时延  $ROTT_i$ , 在公式(4)中,  $S_i$  为发送第  $i$  个数据包时所产生的时间戳,  $R_i$  为当第  $i$  个数据包到达接收端所记录的时间戳.  $ROTT_i$  的值是由数据包  $i$  到达接收端的时间戳减去数据包  $i$  发送端的发送时间戳所得.

$$ROTT_i = R_i - S_i \quad (4)$$

然后在接收端得到  $ROTT$  的一个长期样本加权平均值  $ROTTMEAN$ , 可以避免由于  $ROTT$  值的较大变化造成发送速率很大的波动,  $ROTTMEAN$  采用式(5)、(6)来进行计算:

$$ROTTMEAN = \mu * ROTTMEAN + (1 - \mu)ROTT \quad (5)$$

$$\mu = \begin{cases} 1 & \left( |ROTT_n - ROTTMEAN_n| > \sqrt{\eta S_{n-1}^2} \right) \\ 0.95 & \left( |ROTT_n - ROTTMEAN_n| > \sqrt{\eta S_{n-1}^2} \right) \end{cases} \quad (6)$$

公式(5)、(6)中  $ROTT$  是实测的单向时延值,  $S_{n-1}^2$  是通过最近的 8 个数据包的  $ROTT$  进行加权得到的样

本方差.  $\mu$  的取值如公式(6)所示, 判定系数  $\eta$  的取值极其重要,  $\eta$  选择地过小导致数据包丢失被区分为乱序丢包的可能性增大,  $\eta$  选择地过大导致数据包丢失被区分为拥塞或者链路误码造成的丢包可能性增大. 实验证明,  $\eta$  取值在 1.55~1.75 比较合适, 本文中取  $\eta=1.65$ .

在公式(7)中提出了丢包区分参数  $\tau$ , 定义为当  $ROTT$  前值和前一个长期样本加权平均值的比值.

$$\tau = ROTT_n / ROTTMEAN_{n-1} \quad (7)$$

其中,  $ROTTMEAN_{n-1}$  是前一个长期样本加权平均值,  $ROTT_n$  是当前的单向时延值. 当  $\tau > 1$  时, 可认为网络处于拥塞状态, 对应丢包将被认为拥塞丢包, 发送端需要降低发送速率. 当  $\tau \leq 1$  时, 可认为此时网络处于非拥塞状态, 当前丢包被认为是误码丢包, 此时应按照一定的比例因子扩大当前的丢包间隔, 来增大当前发送端的发送速率, 最终提高网络吞吐量.

考虑到网络存在瞬时的抖动, 我们采用公式(8)对公式(7)进行修正, 其主要思想用一个基于加权移动平滑滤波器来平滑网络测量过程中所产生的噪音, 即

$$\tau'_n = \gamma * \tau'_{n-1} + (1 - \gamma)\tau_n \quad (8)$$

其中,  $\gamma$  取值为 0.125, 当检测到网络产生丢包时, 当  $\tau'_n > 1$  时, 表明对应的丢包是拥塞丢包, 发送端应该降低发送速率, 当  $\tau'_n \leq 1$  时, 表明对应的丢包是无线误码丢包, 此时应该按照一定的比例因子扩大当前的丢包间隔, 来增大当前发送端的发送速率.

根据 TCP Veno 的公式可知丢包间隔  $C_i$  标准的调整因子应取 3, 因此设定的调整因子分段函数应是以 3 作为分段点. 当  $\tau'_n > 1$  时, 网络处于拥塞状态, 调整因子分段函数的值应该处于 [0,1] 之间, 从而降低丢包间隔, 最终降低发送速率; 当时, 网络处于非拥塞状态, 调整因子分段函数的值应该处于 [1,3] 之间, 从而扩大丢包间隔, 最终提高发送速率.

因此本文提出调整因子分段函数  $C(t)$  的计算方法, 为了使得其能够取 3 作为函数的分段点, 并且根据网络中不同状态, 当  $\tau'_n > 1$  时, 网络处于拥塞状态, 那么此时发送速率的降低应该比较缓和的, 那么就针对此状态下速率要求将此段的函数取为指数函数; 反之, 当  $\tau'_n \leq 1$  时, 那么网络处于非拥塞状态, 那么就应该提高发送速率, 但是此时速率的提高并不能像拥塞状态下的那么缓慢, 因此将此状态下的分段函数定义为线性函数.

调整因子分段函数  $C(t)$  在公式(9)定义为:

$$C(t) = \begin{cases} e^{-(\tau'_n-1)^2} / (2\sigma^2), & \tau'_n > 1 \\ -2\tau'_n + 3, & \tau'_n \leq 1 \end{cases} \quad (9)$$

其中  $\sigma$  是极小值, 当  $\tau'_n > 1$ ,  $C(t) \in [0, 1]$ , 减小当前丢包间隔; 当  $\tau'_n \leq 1$ ,  $C(t) \in [1, 3]$ , 增大当前丢包间隔。

此时丢包事件率  $p$  的计算公式(2)可以更新为:

$$p' = \sum_{i=0}^n \omega_i / \sum_{i=0}^n C(t) C_i \omega_i \quad (10)$$

最终改进的 TFRC 吞吐量模型公式(3)可更新为:

$$X' = \frac{S}{R \sqrt{\frac{2bp'}{3} + \frac{1}{m} t_{RTO} \cdot \left( 3 \sqrt{\frac{3bp'}{8}} \right) p' (1 + 32p'^2)}} \quad (11)$$

因此改进型 Adaptive-TFRC 的算法实现最终由以下几个步骤组成:

第 1 步: 在数据接收端计算改进公式中的丢包区分参数值;

第 2 步: 确定网络中数据包的丢包原因以及网络的拥塞程度;

第 3 步: 更新网络中拥塞丢包事件率;

第 4 步: 更新吞吐量模型公式中发送端的发送速率, 能够根据实时网络条件动态自适应地调节传输速率。

## 4 仿真实验结果与分析

### 4.1 仿真拓扑结构

采用 NS2 网络仿真工具作为网络性能测试平台, 仿真实验采用单瓶颈网络拓扑结构, 并分别对 TFRC 以及 Adaptive-TFRC 在无线环境中的 TCP 友好性和网络整体性能进行分析。

在仿真实验中, 采用著名的哑铃型拓扑结构, 设定无线误码丢包率为 1%。如图 2 所示, R0、R1 为发送端节点, R2、R3 是路由节点, R4、R5 为接收端节点。节点 R0、R4 之间分别绑定 TFRC 协议代理和 Adaptive-TFRC 协议代理, 此外节点 R1、R5 之间绑定 TCP 背景流, TCP 流与 TFRC 流以及 Adaptive-TFRC 流形成竞争, 中间路由节点 R2 和 R3 之间构成整个网络的瓶颈链路, 我们设置此瓶颈链路的带宽为 1.6Mbit/s, 瓶颈链路中的时延值为 10ms, 并且对于中间路由节点 R2 和 R3 我们采用目前已经

被广泛应用的 DropTail 队列管理。根据网络拓扑结构的特点, 其他链路的带宽都大于瓶颈链路, 我们设定为 2Mbit/s, 并且延时值均小于瓶颈链路的延时值, 我们设定为 5ms。TCP 流、TFRC 流以及 Adaptive-TFRC 流的数据分组大小设为 1000B, 接收端发送的反馈分组大小设为 40B。整个仿真所需的模拟时间为 100s。

仿真主要从不同无线链路误码率下的平均吞吐量、公平性来分析 Adaptive-TFRC 的性能。

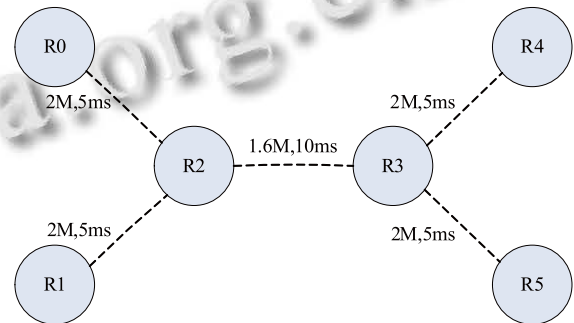


图 2 网络仿真拓扑结构

### 4.2 吞吐量性能分析

在节点 R0、R4 之间分别绑定 TFRC 协议代理和 Adaptive-TFRC 协议代理, 节点 R1、R5 之间绑定 TCP 背景流, 其中在 0 秒时 TCP 流开始发送, 而在 5 秒时 TFRC 流和 Adaptive-TFRC 流开始发送, 前后两次网络配置完全相同, 且设定仿真模拟时间为 100 秒。网络吞吐量如图 3、图 4 所示。

如图 3 所示, 当 TCP 流和 TFRC 流共存的情况下, TCP 流的平均吞吐率为 934kb/s, TFRC 流的平均吞吐率为 615kb/s; 在图 4 中, 节点 R0、R4 之间绑定 Adaptive-TFRC 协议代理, 网络中 TCP 流的平均吞吐率为 900kb/s, 而 Adaptive-TFRC 的平均吞吐率为 690kb/s。对比图 3、图 4 可以发现, Adaptive-TFRC 流相对于 TFRC 流的平均吞吐率提高了 75kb/s, 而对应的 TCP 流却仅降低了 34kb/s, 因此在引入了 Adaptive-TFRC 协议后网络整体吞吐量性能得到提高, 同时相对于 TFRC 协议来说, 能够很好保持对 TCP 流的友好性。

在相同的网络配置下, 节点 R0、R4 之间通过分别绑定 TFRC 协议代理和 Adaptive-TFRC 协议代理, 多次进行仿真, 表 1 为得到统计后网络吞吐量性能仿真结果对表。



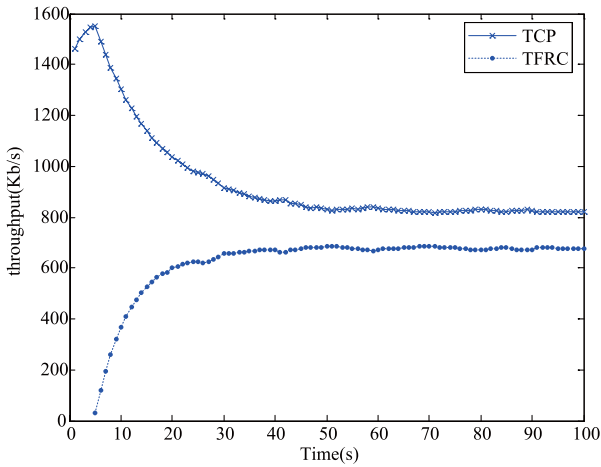


图 3 TCP/TFRC 代理下网络吞吐量性能比较

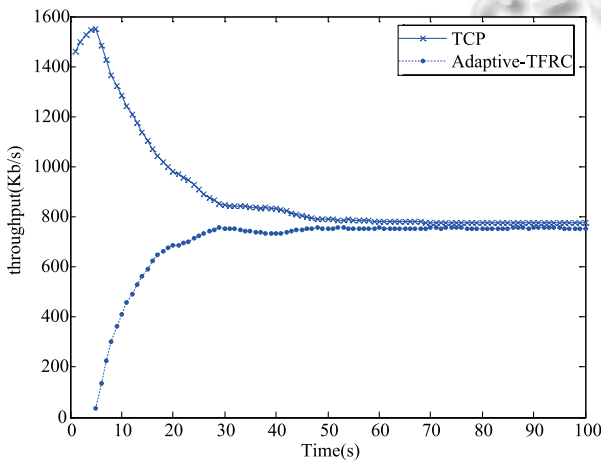


图 4 TCP/Adaptive-TFRC 代理下网络吞吐量性能比较

表 1 网络吞吐量性能仿真结果对比表

算法	TFRC 平均吞吐率(kb/s)	TCP 平均吞吐率(kb/s)	网络总吞吐率(kb/s)
TFRC	615.1	934.4	1549.6
Adaptive-TFRC	690.2	900.6	1590.8

通过分析表 1 发现, 在应用 TFRC 算法时, 网络中 TCP 流和 TFRC 流的吞吐率比值为 0.658; 而应用 Adaptive-TFRC 算法时, 网络中 TCP 流和 Adaptive-TFRC 流的吞吐率比值达到了 0.767. 进一步验证了 Adaptive-TFRC 能够很好地保持对 TCP 的友好性. 在应用了 Adaptive-TFRC 算法后, 网络的总吞吐率要比应用 TFRC 算法时提高 41.3kb/s, 这是由于改进的 TFRC 协议能够精确地区分网络的丢包情况, 并且及时调整发送速率, 使得网络整体吞吐量性能得到提高.

### 4.3 延迟抖动变化分析

基于相同的网络配置和环境, 得到不同的延迟抖动变化趋势图, 图 5、图 6 分别为 Adaptive-TFRC 以及 TFRC 流的延迟抖动变化图. TFRC 流的延迟抖动平均值为 0.006s, 而 Adaptive-TFRC 流的延迟抖动平均值降低为 0.003s, 这体现了 Adaptive-TFRC 算法具有更好的实时性和传输稳定性, 对丢包控制更有效, 同时也更适用于无线网络流媒体的传输.

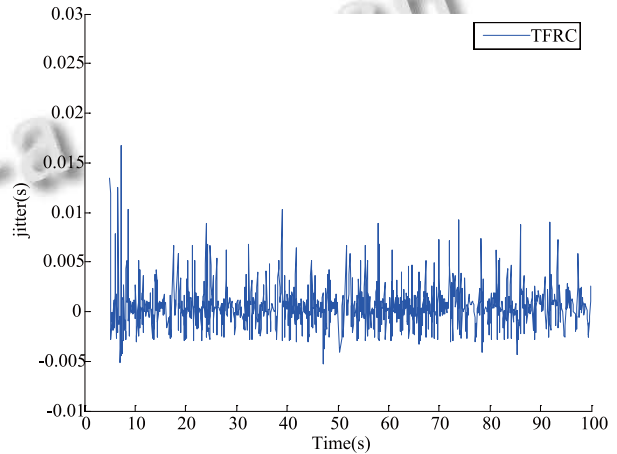


图 5 TFRC 延迟抖动变化图

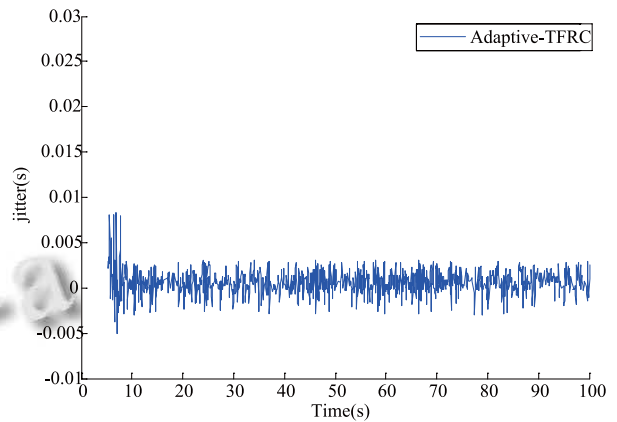


图 6 Adaptive-TFRC 延迟抖动变化图

### 4.4 无线带宽利用率分析

在相同的网络条件下, 仿真中设置无线链路误码率为 1%~5%, 通过 AWK 脚本计算得到此网络中的链路误码率的变化导致网络瓶颈链路带宽利用率变化的一种关系图, 如图 7 所示.

从图 7 可知, 当应用 Adaptive-TFRC 算法后, 网络的瓶颈带宽利用率一直保持在 85%以上. 但是在应用了 TFRC 算法后, 当无线链路误码率增大时, 网络瓶颈链路带宽利用率是随之加速下滑的. 由图 7 可以看

出, 瓶颈链路带宽利用率在误码率变化范围为 1%~2% 时都能保持在 90% 以上, 这是整个瓶颈链路带宽利用率的最高水平. 由于经典的 TFRC 算法不具有针对无线网络的丢包原因进行区分的机制, 因而在整个网络链路误码率达到 3% 时, 网络两端将会呈现越来越多的丢包, 从而导致瓶颈链路的带宽利用率直线下降. 但是此时两端绑定了 Adaptive-TFRC 算法的带宽利用率仍然维持在较高的水平. 当整个网络链路误码率达到 5% 时, 绑定了 TFRC 算法的网络瓶颈带宽利用率以较快的速度下跌至 53%, 而绑定了 Adaptive-TFRC 的网络瓶颈带宽利用率却为 86%. 以上数据表明, 应用了 Adaptive-TFRC 后的网络瓶颈带宽利用率要高很多. 这是因为 Adaptive-TFRC 是基于丢包区分机制, 能够有效地区分链路误码丢包和网络拥塞丢包, 并且具有动态自适应的传输速率调节机制, 提高网络效用.

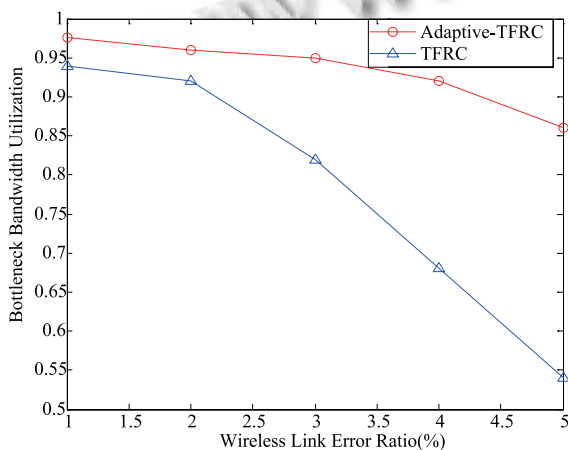


图 7 无线带宽利用率与无线链路误码率关系图

## 5 结束语

本文提出了一种适用于无线网络环境的 Adaptive-TFRC 机制, 当其应用于流媒体传输时, 该机制在接收端利用丢包区分参数来真实反映网络的状态, 然后反馈至发送端, 同时本文对经典 TFRC 机制的吞吐量模型公式进行改进, 最终能够根据实时网络条件动态自适应地调节传输速率. 实验结果表明, Adaptive-TFRC 机制使得网络整体吞吐率性能得到提高, 在保持对 TCP 业务友好性的同时实现了链路的高效使用, 使得网络瓶颈带宽利用率一直保持在较高水平, 并且降低了实时业务流的延时抖动, 因而其具有更好的实时性和传输稳定性, 从而能够保证多协议共存的无线

网络实时流媒体的服务质量.

## 参考文献

- 1 Al-Majeed S, Fleury M. Multi-connection TFRC Video Streaming in a Concatenated Network: Latency and Video Quality. *Recent Trends in Wireless and Mobile Networks*, 2010: 34-45.
- 2 Yao XW, Wang WL, Yang SH, et al. PABM-EDCF: parameter adaptive bi-directional mapping mechanism for video transmission over WSNs. *Multimedia Tools and Applications*, 2011, doi:10.1007/s11042-011-0934-7.
- 3 Tobe Y, Tamura Y, Molano A, et al. Achieving Moderate Fairness for UDP Flows by Path Status Classification. *Proc. 25th Annual IEEE Conf. on Local Computer Networks(LCN 2000)*, Tampa, FL, 2000: 252.
- 4 Floyd S, Handley M, Padhye J, Widmer J. Equation-based congestion control for unicast applications. *ACM SIGCOMM Computer Communication Review*, 2000, 30(4): 43-56.
- 5 Floyd S, Handley M, Padhye J, Widmer J. RFC 5348: TCP Friendly Rate Control (TFRC): Protocol Specification, RFC 5348, 2008.
- 6 朱晓亮, 杜旭, 杨宗凯, 等. 无线传感器网络中实时媒体传输速率控制机制. *小型微型计算机系统*, 2007, 28(2): 199-204.
- 7 姜明, 吴春明, 张旻, 等. TFRC 协议友好性与平稳性改进算法研究. *电子学报*, 2009, 37(8): 1723-1727.
- 8 王东, 陈明, 张大方. 一种基于多路复用的多媒体流 TCP 友好拥塞控制机制. *电子学报*, 2006, 34(3): 567-572.
- 9 Cen S, Cosman PC, Voelker GM. End-to-end differentiation of congestion and wireless losses. USA: *IEEE/ACM Trans on Networks*, 2003.
- 10 Chen M, Zakhor A. Multiple TFRC connections based rate control for wireless Networks. *IEEE Trans. on Multimedia*, 2006, 8(5): 1045-1062.
- 11 Shen H, Cai L, Shen X. Performance analysis of TFRC over wireless link with truncated link-level ARQ. *IEEE Trans. on Wireless Communications*, 2006, 5(6): 1479-1487.
- 12 Zhou B, et al. An enhancement of TFRC over wireless networks. *IEEE Trans. on Multimedia*, 2007, 8(5): 1045-1062.