

# 移动网络下流媒体传输拥塞控制机制的研究

范浩 严军 郑新元 朱静 王臣

(上海大学通信与信息工程学院 上海 200072, China)

**摘要:** 在移动网络环境下,实现流媒体数据的传输已经得到非常广泛的应用。由于经典的 TCP 友好速率控制机制,在网络发生拥塞时,将无线误码丢包误认为拥塞丢包,会降低网络传输速率,导致网络利用率过度降低。如何保证移动网络环境下流媒体发送端正确区分拥塞丢包和无线链路丢包,实现网络传输速率的准确调整是个很重要的研究课题。本文分析了 TFRC 协议以及 AIMD 等拥塞控制机制在该数据传输问题中的适用性,提出了在 TFRC 协议基础上的一种改进的 M-TFRC 协议,添加了丢包区分算法,可以准确区分丢包类型。通过 NS2 分析仿真,验证了 M-TFRC 相比 TFRC 提高了网络吞吐量,维持较高的网络利用率,从而保证移动网络环境下流媒体服务质量。

**关键词:** 移动网络; 拥塞控制; 丢包区分; 流媒体传输

**中图分类号:** TP393 **文献标识码:** A **国家标准学科分类代码:** 510.5025

## Research on congestion control mechanism of streaming media transmission over wireless network

Fan Hao Yan Jun Zheng Xinyuan Zhu Jing Wang Chen

(Communication &amp; Information Department of Shanghai University, Shanghai 200072, China)

**Abstract:** Streaming media transmission has widely used over wireless network. Because of the problem of conventional TFRC mechanism concerning incorrect packet loss rate, when it is applied to mobile congestion networks, the streaming media sever can not get the accurate information of the network condition and thus leading to excessive degrade the sending rate of data. Therefore, an important research issue arises, how to realize loss differentiation between congestion losses and wireless losses and accurate adjustment of network sending rate. Some major congress control protocols including TFRC, AIMD are briefly introduced. This paper propose a mechanism called M-TFRC in the base of TFRC, which appends loss differentiation algorithm. The simulation result shows that M-TFRC improves network throughput and the network utilization and guarantee service quality of wireless network streaming media.

**Keywords:** mobile network; congress control; loss differentiation; streaming media transmission

### 1 引言

随着移动互联网的快速发展<sup>[1]</sup>,人们在无线终端上对基于语音和视频的流媒体服务有很大的需求<sup>[2]</sup>。近年来,无线网络技术快速发展,覆盖范围广,终端产品丰富使得无线流媒体传输系统得到了广泛的应用,如无线视频监控、无线可视电话、视频点播、手机电视直播、移动会议等<sup>[3]</sup>。

流媒体传输具有高网络带宽、低传输延迟、低延迟抖动以及对传输可靠性要求相对较低的特点。流媒体应用对于服务质量有较高要求,而 UDP 协议在与 TCP 竞争中占有优势<sup>[4]</sup>,要保证流媒体对 TCP 流的友好性,必须要引入 TCP 友好性条件下的拥塞控制机制<sup>[5]</sup>。此外,丢包率是判

断网络拥塞的重要指标,而在移动网络环境下,丢包有拥塞丢包和无线链路丢包两种类型<sup>[6]</sup>。因此无线网络环境下如何设计一种传输机制,保证流媒体数据传输对 TCP 的友好性,又可以区分丢包类型,准确判断出当前网络状况,是一个十分重要的研究课题<sup>[7]</sup>

目前主流的 TCP 友好拥塞控制机制中,一类是基于窗口的控制方法,采用加性增加乘性减少(AIMD)的方法<sup>[8]</sup>,这种方法会导致传输速率突变,严重影响流媒体应用的服务质量;另一类是基于速率公式的控制方法,其中 TFRC 就是该方法的典型代表,TFRC 协议根据网络丢包率判断网络状况,调整数据发送速率,来提高网络利用率<sup>[9]</sup>,但 TFRC 主要用于有线网络中,而在混合网络中,TFRC 把无

线链路丢包和拥塞丢包都认为是拥塞丢包,从而在发送端错误的减小数据流的发送速率,降低了网络利用率。所以本文根据实际应用需求,设计一种流媒体传输拥塞控制方法,实现移动网络环境下,能够正确区分丢包类型,动态调整数据发送速率,从而提高网络利用率。

## 2 混合网络下流媒体传输系统

混合网络下流媒体系统如图 1 所示,流媒体系统是实现多媒体数据的采集、传输、播放,系统主要有编码及采集工具、数字音视频记录及发送端<sup>[10]</sup>、客户端播放器三部分组成。信号源采集音频、视频等多媒体数据,流媒体发送端负责响应客户端请求将流媒体数据按照流媒体协议 RTP/RTSP/RTCP 发送,客户端收到数据后,解码播放。

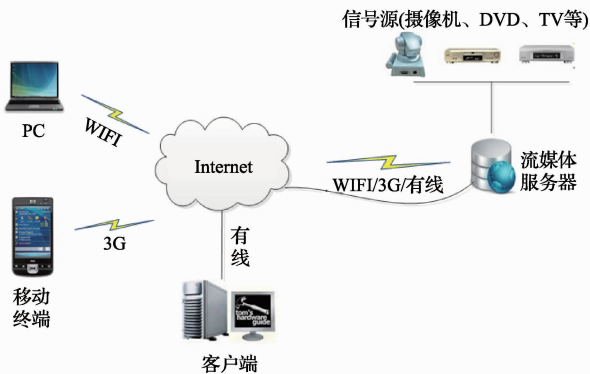


图 1 混合网络下流媒体传输系统

流媒体传输协议 RTCP (real-time transport control protocol) 是一个实时传输控制协议,负责管理网络传输质量,RTCP 发送的 RTCP 包,含有已发送的数据包数量、丢失的数据包等统计数据,但并没有提供拥塞控制机制,并且 RTCP 判读出的丢包包括拥塞丢包和无线丢包。由于网络环境不稳定,有时会出现网络拥塞引起的拥塞丢包,也会出现无线链路引起的无线丢包,这给流媒体数据的传输带来了挑战。

## 3 TFRC 协议及其改进

本文针对无线网络环境下流媒体传输系统中的数据传输需求,设计了引入丢包区分算法的改进 TFRC 协议,实现了网络丢包类型的准确区分及拥塞控制,提高了网络利用率。本协议的关键技术是丢包区分和拥塞控制。

### 3.1 TFRC 协议原理

TFRC 是一种基于速率公式的 TCP 友好的拥塞控制算法。TFRC 根据 RTCP 发送的数据包,判断当前网络状态,实时调整 RTP (real-time transport protocol) 包的数据发送速率<sup>[11]</sup>。在一个 RTP 会话期间,接收端会周期性地

发送 RTCP 包,RTCP 不提供网络状态检测,它依靠 RTCP 进行网络状态检测,由接收端提供流量控制和拥塞控制服务,从而克服 UDP 传输的不可靠性以及 TCP 不友好性,使得传输效率最佳化<sup>[12]</sup>。TFRC 协议采用如下吞吐量计算公式<sup>[13]</sup>:

$$Send\_rate = \frac{Q_{size}}{t_{RTT} \times \sqrt{\frac{2p}{3}} + 3 \times t_{RTO} \times \sqrt{\frac{3p}{8}} \times (1 + 32p^2)} \quad (1)$$

式中:  $Send\_rate$  是计算得到的数据发送速率,  $Q_{size}$  为发送的数据包大小,  $t_{RTT}$  为往返时间,  $t_{RTO}$  为重传超时时间,一般取值为  $RTO = 4 \times t_{RTT}$ ,  $p$  为丢包事件率。数据包大小以及重传时间是稳态的,发送端速率的调整主要是根据丢包率  $p$  和  $t_{RTT}$  决定。

TFRC 实现过程如下:

- 1) 接收端将丢包事件率  $p$ , 作为网络状态参数与单向传输时延一起通过 RTCP 包发送给发送端;
- 2) 发送端在 RTCP 反馈包中, 提取时间戳和往返时延 RTT;
- 3) 将上述得到的  $p$  和 RTT, 带入吞吐量计算公式(1), 计算出数据包的发送速率;
- 4) 发送端计算吞吐量, 调整数据发送速率。

### 3.2 TFRC 协议局限性

TFRC 将无线丢包和拥塞丢包都归咎为拥塞丢包, 由于 TFRC 的数据发送速率计算公式主要根据丢包率和往返时延来计算, 在有线网络中丢包事件是由于网络拥塞引起的拥塞丢包, 因此在传统的有线网络环境下表现良好<sup>[5]</sup>。

但是在有线和无线混合网络环境下, 丢包类型有拥塞丢包和无线链路丢包, 接收端错误的将无线链路丢包也归结为拥塞丢包, 这使得判断出的丢包率变大, 导致数据以较低的速率进行传输, 使得网络带宽利用率下降<sup>[14]</sup>。

因此, 要对 TFRC 协议进行改进, 必须在数据发送端, 将拥塞丢包和无线链路丢包区分开, 仅统计拥塞丢包, 准确判断当前网络状况, 提高网络利用率。

### 3.3 TFRC 协议改进

在流媒体发送端, TFRC 协议根据接收端反馈回来的 RTCP 包, 计算双向传输时延  $RTT$  和丢包率  $p$ , 同时假设丢包都是拥塞丢包, 很显然它并没有使用丢包区分算法。为使 TFRC 适用于移动网络环境下, 本文在 TFRC 的基础上添加能够进行丢包区分的 M-Spike 算法, M-Spike 算法相对于 Spike<sup>[15]</sup> 算法增加了丢包状态的待定区及对  $RTT$  的平滑处理。当检测到  $RTT$  处于拥塞丢包阈值和无线丢包阈值之间时, 为准确判断丢包类型, 需要检测  $RTT$  的趋势, 如果是增加的趋势则判断为拥塞丢包, 反之判断为无线链路丢包。如图 2 M-Spike 算法示意图, 下面简述 M-Spike 算法的流程。

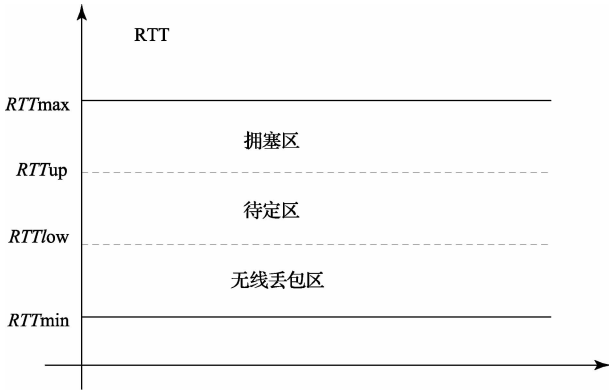


图2 M-Spike算法示意图

在流媒体发送端,根据收到的 RTCP 包,提取 RTP 包时间戳,按照公式(2)计算第  $i$  个包的双向传输时延  $RTT(i)$ 。

$$RTT(i) = T_{arrive}(i) - T_{send}(i) \quad (2)$$

式中:  $T_{arrive}(i)$  为第  $i$  个包的到达时间,  $T_{send}(i)$  为第  $i$  个包的发送时间。为了避免  $RTT$  的波动过大,导致阈值过高需要对  $RTT(i)$  做平滑处理,如下式:

$$RTT(i) = (1 - \gamma) * RTT(i) + \gamma * RTT(i - 1) \quad (3)$$

式中:  $\gamma$  为平滑因子,根据实验  $\gamma$  取 0.8 时实验效果最好,即每个新估计的  $RTT(i)$ ,80% 来自于前一个估计值而 20% 依赖于新的测量值,再根据如下式计算判为拥塞丢包和无线丢包的阈值:

$$RTT_{up} = RTT_{min} + \alpha * (RTT_{max} - RTT_{min}) \quad (4)$$

$$RTT_{low} = RTT_{min} + \beta * (RTT_{max} - RTT_{min}) \quad (5)$$

式中:  $RTT_{min}$  和  $RTT_{max}$  分别为统计时间内  $RTT$  最大值和最小值,  $\alpha$  和  $\beta$  为调节因子,对两个参数的赋值将会影响到此算法的精确度,通过实验发现  $\alpha$  和  $\beta$  最佳值分别是 0.8 和 0.3;当检测到丢包事件时,判断  $RTT(i)$  与阈值的关系,如果大于  $RTT_{up}$  则判断为拥塞丢包,如果小于  $RTT_{low}$  则判为无线链路丢包,如果介于  $RTT_{up}$ 、 $RTT_{low}$  之间则处于待定区;当检测到当前  $RTT(i)$  为待定区时,需要进一步做  $RTT$  变化趋势判断,计算  $RTT(i)$  与  $RTT(i-1)$  大小关系,如果  $RTT(i) > RTT(i-1)$ ,判为拥塞丢包,反之判为无线链路丢包。

如图 3 改进的 TFRC 协议 M-TFRC 流程图所示,发送端收到接收端反馈回的 RTCP 包后,进行数据分析处理,如果检测到丢包事件发生,则应用上述 M-Spike 算法判断

丢包类型,区分出拥塞丢包和无线链路丢包,统计拥塞丢包率,再利用公式(1)计算数据发送速率,实时调整数据发送速率。

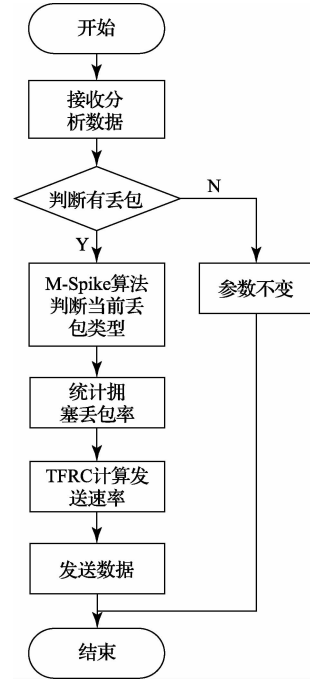


图3 M-TFRC协议流程图

#### 4 仿真与分析

采用网络模拟器 NS2<sup>[14]</sup>,在 Ubuntu 12.04 环境下进行仿真,在相同参数情况下,分别对 TFRC 及加入丢包区分算法 Spike 的 S-TFRC,以及加入丢包区分算法 M-Spike 的 M-TFRC 进行仿真。

在仿真实验中,采用如图 4 网络仿真拓扑结构,  $S_0$ 、 $S_1$  为发送端节点,  $R_0$ 、 $R_1$  为接收端节点,  $S_2$  为家庭路由器,  $R_2$  为基站。节点  $S_0$ 、 $R_0$  之间分别绑定 TFRC 协议代理、S-TFRC 协议代理、M-TFRC 协议代理。有线网络部分设置是无差错的,而无线网络设置是 IEEE802.11。其中  $S_2$  和  $R_2$  之间是整个网络的瓶颈带宽,设置为 1.7 Mbit/s,时延是 5 ms。其他链路带宽都大于瓶颈带宽,设置为 2.1 Mbit/s,时延 5 ms。仿真过程所需的模拟时间为 50 s,并且是在不同无线链路误码率情况下网络的吞吐量分析。

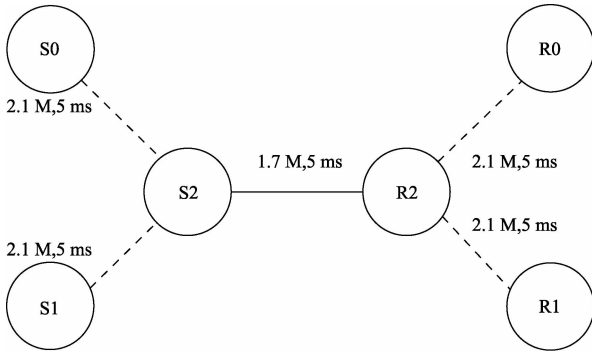


图4 网络仿真拓扑结构

分别在无线链路丢包率是0%和2%的情况下,前后三次网络仿真配置相同情况下,在0 s时 TCP 流开始发送,10 s时 TFRC 流、S-TFRC 流、M-TFRC 流开始发送。网络吞吐量的仿真图分别如图5、图6所示。

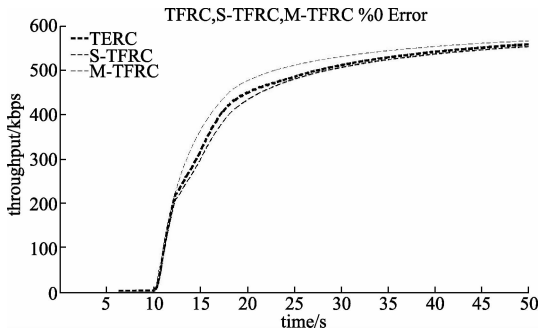


图5 无线链路误码率为0%吞吐量比较图

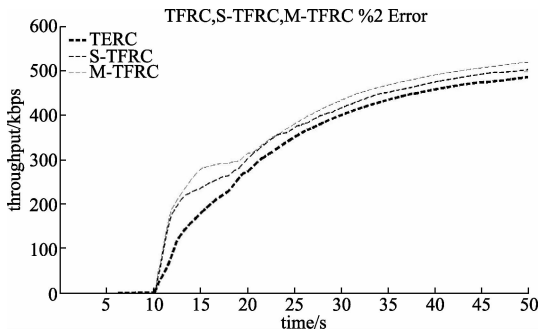


图6 无线链路误码率为2%吞吐量比较图

如图5所示,在无线链路误码率是0%情况下,TFRC、S-TFRC、M-TFRC协议的吞吐量基本相同。因为这种情况下,丢包主要是拥塞丢包,不存在丢包错误判断问题。如图6所示,无线链路丢包率是2%时,使用了丢包区分算法的M-TFRC和S-TFRC的吞吐量高于未使用丢包区分算法的TFRC。其中M-TFRC趋于平稳的吞吐量比TFRC提高6.76%,S-TFRC趋于平稳的吞吐量比TFRC提高3.28%,M-TFRC趋于平稳的吞吐量比S-TFRC提高3.48%。

这表明,使用了更优良丢包区分算法 M-Spike 的 M-TFRC 准确的判断了网络状态,及时调整发送速率,更大限度的发挥了网络的利用率。因此,对添加丢包区分算法的 M-TFRC,提高了网络吞吐量,在无线网络的流媒体传输中是切实可行的。

## 5 结 论

本文针对移动网络环境下传统的 TFRC 协议不能区分丢包类型、过度降低网络传输速率问题,提出了一种适用于移动网络环境下改进的 TFRC 协议 M-TFRC 协议,可以满足在流媒体传输时,发送端利用丢包区分算法 M-Spike 来真实反映网络的情况,并利用吞吐量公式实时调整发送速率。通过仿真实验证明 M-TFRC 使得网络吞吐量性能得到提高,有效的提高了网络带宽利用率。

## 参考文献

- [1] 严中毅,李凯. 测量仪器与现代微电子、计算机和软件技术的融合[J]. 电子测量与仪器学报,2015,29(5): 631-637.
- [2] 袁浩,杜茂康,李昌兵. 一种适用于无线网络流媒体传输的改进 TFRC 协议[J]. 哈尔滨理工大学学报,2011,16(2): 40-43.
- [3] 翟旭平,王琦,聂勇. WCDMA 智能终端无线视频传输系统设计与实现[J]. 电子测量技术,2014,37(7): 101-105.
- [4] 李校林,刘海波,张杰,等. RTP/RTCP, RTSP 在无线视频监控系统的设计与实现[J]. 电视技术,2011,35(19): 89-92.
- [5] 瞿佳俊,严军,朱渊婧. 无线传感网低开销型数据可靠传输方法的研究[J]. 电子测量技术,2014,37(1): 92-95.
- [6] 王万良,张小玮,姚信威,等. 无线网络中基于流媒体传输的自适应 TFRC 机制[J]. 计算机系统应用,2013,32(7): 161-167.
- [7] TAPPAYUTHPIJARN K, LIEBL G, STOCKHAMMER T, et al. Adaptive video streaming over a mobile network with TCP-friendly rate control [C]//Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly. ACM, 2009: 1325-1329.
- [8] SONG Z, ZHANG Y, ZHOU M. Enhanced TFRC congestion control mechanism based on refined network states division [C]//2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP). IEEE, 2015: 938-942.

(下转第132页)