

## QoE 优先的移动网络 RRTCC 传输改进策略\*

葛志辉, 岑霄, 李陶深, 叶进



(广西大学 计算机与电子信息学院, 广西 南宁 530004)

通讯作者: 葛志辉, E-mail: gezhihui@gxu.edu.cn

**摘要:** 目前 Chrome 浏览器为 WebRTC 适配的 RRTCC(receiver-side real-time congestion control) 拥塞控制算法在移动网络环境的多流竞争情况下表现较差. 在有 TCP 流共存的情况下会过度退让从而导致 WebRTC 流饥饿, 在视频会议这种多 WebRTC 流并发的情况下, 新加入的 WebRTC 流会损害已有流的通信质量. 针对该问题, 提出了 QoE 优先考量的 RRTCC 传输控制改进策略 MTCIS, 该策略根据对当前移动网络性能和 QoE 需求的分析, 采取激进程度适中的 WebRTC 传输控制方法, 避免了与 TCP 流竞争时过度退让的问题. 针对多 WebRTC 流并发的场景, 设计了并发流管理机制, 在合理利用带宽和优先满足用户 QoE 的基础上, 对视频流的传输参数进行动态调整. 实验结果表明, 在移动网络场景多流竞争的情况下, MTCIS 策略能够有效地保证 WebRTC 的合理竞争性、稳定性和健壮性, 提高了用户的 QoE 体验.

**关键词:** WebRTC; 移动网络; QoE; 速率控制; 多流竞争

中文引用格式: 葛志辉, 岑霄, 李陶深, 叶进. QoE 优先的移动网络 RRTCC 传输改进策略. 软件学报, 2016, 27(Suppl. (2)): 91-102. <http://www.jos.org.cn/1000-9825/16024.htm>

英文引用格式: Ge ZZ, Cen X, Li TS, Ye J. QoE preemptive RRTCC transmission improvement strategy for mobile network. Ruan Jian Xue Bao/Journal of Software, 2016, 27(Suppl. (2)): 91-102 (in Chinese). <http://www.jos.org.cn/1000-9825/16024.htm>

## QoE Preemptive RRTCC Transmission Improvement Strategy for Mobile Network

GE Zhi-Hui, CEN Xiao, LI Tao-Shen, YE Jin

(School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China)

**Abstract:** Chrome browser provides a RRTCC (receiver-side real-time congestion control) algorithm for WebRTC, but it exhibits poor performance in the mobile network environment. It cannot effectively compete with other TCP flow applications, causing WebRTC starvation. Furthermore, in the case of multi-WebRTC concurrent flow, new entrants will impact the existing flows. This paper proposes a transmission control strategy MTCIS. The core of MTCIS is the use of an improved RRTCC algorithm in consideration of mobile network scenarios and QoE, and the proper control of WebRTC, which can avoid the problem of excessive concession when competing with TCP flow. While in multi-WebRTC flow concurrent scenarios, the MTCIS can dynamically adjust the transmission parameters of video streaming, based on the analysis of rational utilize bandwidth and priority to satisfy users' QoE. Experimental results show that MTCIS can be used in mobile network environment of flows competition, ensure reasonable competition, stability and robustness of WebRTC, which improves the users' QoE.

**Key words:** WebRTC; mobile network; QoE; rate control; multi-stream competition

近年来快速兴起的 WebRTC(Web real-time communication)技术正鼓励着互联网上的视频通信应用大量增加, 其主要原因在于 WebRTC 技术脱离了传统意义上基于操作系统桌面而独立的 RTP(real-time transport

\* 基金项目: 国家自然科学基金(61363067, 61462007); 广西多媒体通信与网络技术重点实验室(广西大学)资助项目

Foundation item: National Natural Science Foundation of China (61363067, 61462007); Program of Guangxi Key Laboratory of Multimedia Communications and Network Technology (Guangxi University)

收稿时间: 2016-05-01; 采用时间: 2016-10-18

protocol)实时通信应用的范畴,它能够轻易地在支持 RTP 协议的浏览器上提供实时流媒体通信服务.WebRTC 的标准化工作目前正由 W3C(world wide Web consortium)和 IETF(Internet engineering task force)两大互联网组织进行,W3C 负责定义 Javascript APIs,而 IETF 则研究 WebRTC 如何使用现有的多媒体组件.过去,浏览器上的每一个实时流媒体通信应用都需要实现一套自己适用的多媒体协议栈,并制作成插件让用户下载安装才能支持音视频通话.而有了 WebRTC 这个标准以后,多媒体协议栈已作为自带组件内置到浏览器中,开发人员只需适当地使用 HTML5 API 就可以完成一个标准的 WebRTC 实时流媒体通信应用.虽然目前 WebRTC 正在为多媒体通信领域带来一股浪潮<sup>[1]</sup>,但它仍存在一些缺点,比如,目前关于智能移动终端的 WebRTC 应用研究还较少.由于移动设备的处理能力,网络环境和能源相对有限,目前在移动终端上 WebRTC 提供的用户 QoE(quality of experience)还不是很好<sup>[2]</sup>.如何对智能移动设备的资源和 WebRTC 应用的需求进行协调以提高用户 QoE 的体验,是未来需要进一步研究的内容.

目前,虽然 WebRTC 并没有提出一个官方的拥塞控制方法,但 Chrome 浏览器为 WebRTC 适配了一种 RRTCC(receiver-side real-time congestion control)拥塞控制算法<sup>[3]</sup>,这种算法已作为因特网草案(Internet-draft)提交,且已有部分学者对它各方面的性能表现进行了实验分析,给出了中肯的评价:目前 RRTCC 在有线网络环境下表现较好,但在 200ms 延时以上的移动网络环境中表现较差<sup>[2,4]</sup>.

传统的 TCP 协议是为有线网络设计的,由于有线网络和无线网络所固有的差异性,原 TCP 协议并不适用于无线网络.针对无线网络环境中 TCP 协议优化的问题经过了多年的研究,目前已有较多优秀成果,如 802 系列标准和 LTE 技术早已成为业界的标准.可以预计,对于无线网络环境中因延时、带宽不稳定而容易出现的网络抖动等问题,TCP 协议已经解决得较好,而基于 RTP 协议的 WebRTC 在移动网络环境下的应用仍需进一步的优化.

通过性能测试与分析我们发现,目前 RRTCC 没有很好地解决移动网络环境下 WebRTC 多流竞争的问题.在 WebRTC 与 TCP 流竞争的过程中,由于 WebRTC 过度地退让会导致实时视频通信失败;而在 WebRTC 视频会议应用中,人员的加入和退出操作会对其他人员的通信质量造成较大的损伤.因此,本文基于对 RRTCC 算法的添加与修改,提出了 QoE 优先的移动传输控制改进策略 MTCIS.该策略主要包含面向移动网络的传输控制方法和 WebRTC 并发流管理机制两大模块,能够提升移动网络环境下 WebRTC 的性能表现,使其与 TCP 流进行合理竞争,维持一定程度上的平衡;而针对多人视频会议应用,能够有效降低 RTP 流达到平衡的时间,减少视频卡顿现象.实验结果表明,与原 RRTCC 算法相比,在移动网络场景多流竞争的情况下,MTCIS 策略能够有效保证 WebRTC 的合理竞争性、稳定性和健壮性,提高了用户的 QoE 体验.

## 1 相关研究现状

目前在多媒体实时通信应用场景下进行传输速率控制的研究已有较多,它们所采取的主要研究方法都是在保证一定用户体验质量的条件下,根据可用带宽自适应数据传输速率.文献[5]提出了一种基于对 RTCP(RTP control protocol)头部进行拓展,用于实时视频通话的速率自适应算法,根据峰值信噪比、丢包率和带宽占用率等参数对链路质量进行估计.文献[6]对传输速率的控制则是通过对隐式(延时、丢包率等)和显式(ECN 头部)的拥塞指标进行综合考虑,这种控制方法与多种队列管理机制协作得较好.文献[7]提出了一种在没有公共链路存在且同步可用带宽被限制的条件下,通过同步多数据包流来估计网络链路可用带宽的机制.

另一类研究工作则关注 TCP 友好方面,由于 UDP 协议与 TCP 协议的差异性,在发生竞争时 UDP 协议占有强烈的优势,为了保证 RTP 流对 TCP 流的友好性,引入 TCP 友好性的拥塞控制机制是有非常必要的.TFRC(TCP friendly rate control)算法<sup>[8]</sup>就是这部分研究的典型代表.它提出了一个根据丢包率和往返延时计算可用带宽的公式,能够对发送速率实时调整,提高带宽利用率.但是,经过改进后的 TCP 协议会对 UDP 协议产生何种影响,他们竞争的优劣性是否会发生变化,关注这个问题的研究工作则较少.

以上的研究工作大部分都是在稳定的有线网络场景下进行,近年来随着移动互联网的快速发展,越来越多的学者关注到在移动网络环境下的实时流媒体传输控制问题,并且更重视应用服务的 QoE 问题.文献[9]提出了一种适用于异构网络环境,以用户 QoE 为主导的实时音视频通话自适应方案.文献[10]通过对 GCC(Google

congestion control)算法和 Sprout 算法进行分析,提出一种针对 LTE 网络环境的改进算法,该算法能够在保持低延时的基础上进行高效传输,让视频通话更加稳定.文献[11]提出了一种基于 QoE 的视频流自适应方案,视频质量随网络状态自适应调整.可见,在移动网络环境下进行实时流媒体传输控制研究,需要关注带宽、延时和丢包率的变化,解决如何在链路质量不稳定的情况下进行高效传输并保证用户 QoE 的问题.

目前关于 WebRTC 的研究主要是一些初步的性能分析和传输机制优化,只有一小部分的研究使用移动设备作为实验平台.文献[4]分别在星型和网状的网络拓扑环境中,对 WebRTC 的拥塞机制进行了详细的实验分析,结果表明在延时低于 200ms 的情况下,WebRTC 表现良好.文献[12]对 WebRTC 实时视频传输协议的可行性进行了研究,初步结果表明,目前 WebRTC 能够建立一个小规模 P2P 视频网络,但实现一个大规模的视频网络仍存在不小的挑战.文献[13]提出让 WebRTC 通过发送冗余的数据包探测可用带宽的方法,并且当拥塞发生时这些冗余的数据包能够对 RTP 包进行 FEC 向前纠错操作,保证了一定的用户 QoE.文献[14]提出了针对 LTE 网络环境的 WebRTC 拥塞控制机制,通过可用带宽估计和 TCP 友好控制,能够有效降低数据包延时和丢包率.

## 2 问题定义

WebRTC 作为近年来新兴的一种实时流媒体传输技术,受到了工业界和学术界的广泛关注.部分优秀学者已经对应用 RRTCC 算法的 WebRTC 服务在各种受限条件下的性能表现进行了分析与研究,指出目前 RRTCC 算法在有线网络环境下表现较好,但在移动网络中则表现不佳.本文经过大量调研与分析,特别针对移动网络环境下 WebRTC 多流竞争问题展开研究,通过对应用 RRTCC 算法的 WebRTC 应用进行详细的实验对比与分析,发现问题并提出优先保证用户 QoE 的解决思路.WebRTC 的多流竞争实验主要分两部分:WebRTC 流与 TCP 流的竞争以及多 WebRTC 流之间的竞争.

### 2.1 TCP 流竞争问题

为了说明移动网络环境下 WebRTC 流与 TCP 流的竞争问题,本文设置了两组实验,都是在 WIFI 802.11n 网络环境下进行.第 1 组实验有 2 台移动设备,下行链路带宽都为 1M,一台设备运行 WebRTC 的 P2P 视频通话应用,另一台设备运行基于 TCP 长流的下载文件应用,两台测试设备会同步在一定范围内移动,经过障碍物,以制造链路状态变化的情况.实验结果如图 1 所示,可见对于链路状态变化而造成网络轻微抖动的情况下,TCP 长流能够更稳定地保持传输速率,而基于 RTP 协议的 WebRTC 流传输速率波动较大,从用户的角度来,看视频卡顿较多,并有斑点出现.

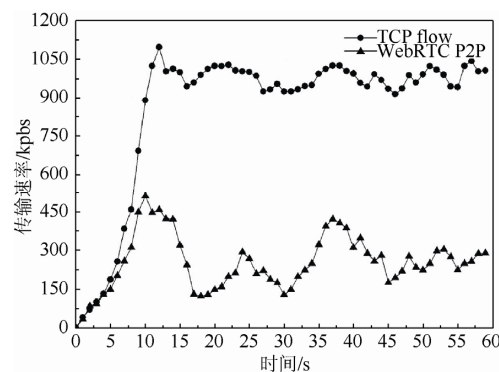


Fig.1 Comparison of transmission rate between TCP flow and WebRTC P2P in mobile network

图 1 移动网络下 WebRTC 应用与 TCP 应用的传输情况对比

第 2 组实验则是在一台设备上同时运行 WebRTC 的 P2P 视频通话和 TCP 长流下载文件应用,上/下行链路带宽分别为 1M.实验如图 2 所示,可见,当总的传输速率到达链路带宽上限时,WebRTC 流和 TCP 流发生了竞争,TCP 流能够稳定地保持传输速率并缓慢上升,而 WebRTC 流由于过度退让,传输速率过低,最终导致视频通话

服务崩溃.

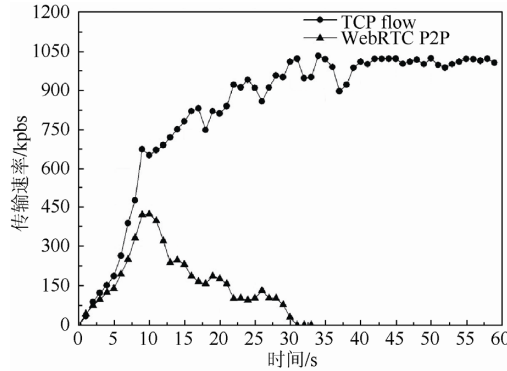


Fig.2 Bandwidth competition between TCP flow and WebRTC P2P in mobile network

图2 移动网络下 WebRTC 应用与 TCP 流带宽竞争情况

通过对以上实现现象结合 RRTCC 算法流程进行分析,我们得出以下结论:(1) 应用 RRTCC 算法的 WebRTC 没有对移动网络环境中出现网络波动的情况与真正发生网络拥塞的情况进行良好的区分,在发生网络波动的情况下,延时的增加和丢包率的增加会直接导致 RRTCC 对链路状态的估计过于消极,认为网络拥塞已发生,对传输速率的衰减控制过于敏感。(2) 虽然 RRTCC 算法集成了基于 TFRC 算法的 TCP 友好机制,解决了 WebRTC 流导致 TCP 流饥饿的问题,但是由于移动网络环境下经过优化的 TCP 协议比有线网络环境下的 TCP 协议更为激进,而 TFRC 过于“友好”,反而导致了 WebRTC 流自身饥饿的问题。

## 2.2 多 WebRTC 流竞争问题

由于 WebRTC 使用的是基于 UDP 协议的 RTP/RTCP 传输协议,因此多 WebRTC 流竞争问题其实是 RTP 流并发问题.实验场景为多 RTP 流的 WebRTC 视频会议应用,网络的上/下行链路带宽分别为 1M,实验设置每 10s 的时间间隔会有新的用户加入视频会议,一共有 3 个用户加入,每加入一个用户就生成一个新的 RTP 流.实验结果如图 3 所示,可见,当第 3 个用户加入视频会议时,网络带宽不足,最早生成的第 1 个 RTP 流传输速率大幅度下降,而最新生成的第 3 个 RTP 流传输速率上升,第 2 个 RTP 流速率正常,状态持续了约 15s 后,3 个 RTP 流速率达到平衡,从用户的角度观察,当第 3 个用户加入时,第 1 个用户的视频发生明显卡顿,状态持续了 15s 后,3 个用户的视频会话恢复正常,但整体画面质量相对有所下降。

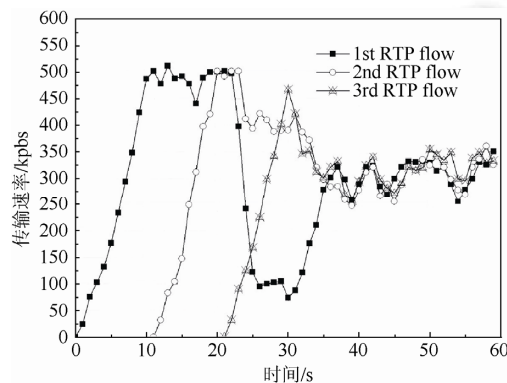


Fig.3 Concurrent transmission of WebRTC RTP flows

图3 WebRTC 多 RTP 流并发的传输情况

通过实验现象可知,WebRTC 对于多 RTP 流并发的场景处理得还不是很好,链路带宽不足时,WebRTC 会优

先保证新加入 RTP 流的传输质量,带宽的不足则会影响到已有流的数据传输,损坏已有流的用户 QoE 体验。

### 2.3 移动 WebRTC 的用户 QoE 问题

近年来,移动服务的 QoE 问题是工业界和学术界关注的一个热点问题,运营商们纷纷意识到在残酷的市场竞争中,如何提高用户的 QoE 体验以保持并扩大用户规模是生存和盈利的关键。QoE 被描述为“在应用或服务的使用过程中,通过主观或客观的心理评价方法来对用户体验质量进行评估”<sup>[15]</sup>。影响用户 QoE 的相关因素有很多方面,比如用户接口的设计、用户使用习惯、设备电量、QoS 指标等。

与有线网络相比,移动网络下的延时通常会更高且容易波动。用户的移动、障碍物的出现和移动设备到 AP 距离的增加都会导致延时恶化。此外,可用带宽不足或网络传输队列过长也会造成延时的增加。低延时能够让用户体验流畅的视频,而高延时则会导致画面偶尔停顿甚至完全冻结。和延时一样,丢包率在移动网络环境中通常更高。由于基于 UDP 协议的 WebRTC 没有数据包重传机制,在发生丢包的情况下,在视频播放过程中可能会出现斑点,更严重的情况下,丢失的视频帧会造成某些画面直接被跳过。尽管少量的丢包不会对用户 QoE 造成很大的影响,但很高的丢包率可能会造成视频整个崩溃。

目前 WebRTC 在有线网络环境下整体性能表现较好,获得了业界的一致好评,但在移动网络环境下的应用仍存在许多问题。虽然也有部分学者提出了在移动网络环境下 WebRTC 传输优化方法,但这些研究往往都是从 QoS(quality of service)的角度进行优化,对用户 QoE 造成的影响并不是特别关注。在当前移动互联网时代的大背景下,如何提升用户的 QoE 才是应该关注的首要问题。因此,本文从优先保证用户 QoE 的角度出发,针对移动网络环境下 WebRTC 多流竞争问题展开研究,以 QoE 为前提进行 WebRTC 服务的性能优化,具有较大的现实意义。

## 3 QoE 优先的传输控制解决方案

本文基于对以上的实验现象和问题分析,通过对原 RRTCC 算法进行改进与补充,提出了一种 QoE 优先考量的 WebRTC 传输控制改进策略 MTCIS,旨在解决移动网络环境下 WebRTC 对多流竞争处理不当而影响用户 QoE 的问题。该策略主要包含两大模块:面向移动网络的传输控制方法和 WebRTC 并发流管理机制。

### 3.1 面向移动网络的传输控制方法

与有线网络相比,移动网络所固有的链路质量不稳定,易受周围环境影响形成网络抖动,进而导致网络延时和丢包率容易波动的特性,使得在有线网络环境下工作稳定的传输控制算法不一定适用于移动网络,目前应用于 WebRTC 的 RRTCC 算法正是如此。本文提出的面向移动网络的传输控制方法基于对移动网络场景与用户 QoE 需求的动态分析,通过对原 RRTCC 算法中的的传输速率控制机制和 TCP 友好机制进行改进和补充,在保证用户 QoE 的前提下,提升 WebRTC 在移动网络环境下对链路质量的容忍度,使传输机制更为激进,进而能够有效区分网络拥塞和网络抖动的情况,解决因 WebRTC 与 TCP 流竞争不平衡而导致自身饥饿的问题。

#### 3.1.1 TCP 友好机制

在发生拥塞的情况下,TCP 协议能够主动减小发送窗口以降低传输速率,保证服务质量,但是尽最大努力交付的 UDP 协议没有拥塞控制机制。因此,为了避免 UDP 流占据链路中所有可用带宽,造成 TCP 流饥饿的情况,RRTCC 算法中引入了 TCP 友好机制,这其实也是保护用户 QoE 的一种方法。虽然该机制在有线网络中能够有效地运作,使 RTP 流与 TCP 流保持相对的平衡,但是在移动网络中,情况有所不同。各大移动网络协议标准如 802.11a/b/g/n 和 3G/LTE 等都或多或少地对 TCP 协议进行了改进,且目前移动网络环境下 TCP 协议的优化仍是一个重要的研究点。这意味着,在移动网络环境下,改进的 TCP 协议已经更适应于链路质量不稳定的特性,本文的性能测试实验结果也说明了这个问题。因此,需要对原 RRTCC 算法中的 TCP 友好机制进行改进,以避免 RTP 流由于过度退让,无法与 TCP 流达到平衡,反而导致自身饥饿的问题。TCP 友好机制应该以保护用户整体 QoE 为前提,友好应该保持一定程度上的平衡,过度地友好或激进都会造成用户 QoE 的损失。

原 RRTCC 算法中 TCP 友好机制的实现主要基于 TFRC 算法<sup>[8]</sup>。该算法主要根据各 QoS 参数计算当前可用

带宽,如式(1)所示.

$$ESA_{bps} = \frac{s}{RTT \sqrt{\frac{2bp}{3}} + RTO \left( 3 \sqrt{\frac{3bp}{8}} \right) p(1+32p^2)} \quad (1)$$

其中, $ESA_{bps}$ 是可用带宽估计值, $s$ 是平均每秒接收到的数据包大小, $p$ 则是丢包率, $RTT$ 是数据包往返时间单位为 $s$ , $RTO$ 是TCP超时重传时间, $b$ 是单个TCP流最大可传输数据包大小.

$ESA_{bps}$ 并不代表最优的发送速率,而是会作为RTP数据发送端的传输速率上限值,因此 $ESA_{bps}$ 直接影响了传输速率的激进程度.由于移动网络中经常发生的网络抖动会使 $RTT$ 和 $p$ 值增大, $ESA_{bps}$ 值减少,进而降低了传输速率的上限值,因此我们计划对TFRC算法进行修改,减弱 $RTT$ 和 $P$ 值对 $ESA_{bps}$ 值的影响,使整体的传输策略更为激进.改进后的TFRC算法如式(2)所示.

$$P\_ESA_{bps} = \frac{s}{\frac{RTT_{av}}{2} \sqrt{\frac{5bp}{3}} + RTO \left( 3 \sqrt{\frac{3bp}{16}} \right) |\ln p| (1+32p^2) + 2p} \quad (2)$$

其中, $RTT_{av}$ 是最近3次 $RTT$ 的平均值,使之变化更平缓,并根据移动网络中 $RTT$ 和 $P$ 值的大致变化范围,调整了部分参数的大小,以降低 $RTT$ 和 $P$ 值在算法中的权重,使可用带宽估计值 $P\_ESA_{bps}$ 更稳定,不易受 $RTT$ 和 $P$ 值变化的影响.此外,为了避免TCP流过度挤压RTP流,导致 $s$ 值过小,进而导致 $P\_ESA_{bps}$ 值过小的问题,应该为 $P\_ESA_{bps}$ 设定一个下限值,这个下限值应该根据当前网络总带宽和保证WebRTC提供用户可接受的最低QoE而所需的网络带宽来设定,如式(3)所示.

$$\min(P\_ESA_{bps}) = \max(0.2 \times S_{bps}, \min(QoE_{bps})) \quad (3)$$

其中, $\min(P\_ESA_{bps})$ 代表可用带宽估计值 $P\_ESA_{bps}$ 的下限值, $S_{bps}$ 代表当前网络总带宽, $QoE_{bps}$ 代表满足用户最低QoE需求的带宽值,它可以由用户在WebRTC应用中设定.

本文研究的一个主要目标是优化用户的QoE体验,但拥塞控制和QoE是矛盾的,QoE体现的是用户对服务整体的感受,所以在保证WebRTC一定QoE的前提下进行TCP友好控制,使RTP流和TCP流在发生竞争时在某个适合的程度上达到平衡,是为了保证用户对移动设备整体服务的QoE体验.在网络环境不好的情况下,优先保证QoE;在网络环境好且QoE能够满足的前提下,通过TCP友好机制保证各种数据流的公平性.

### 3.1.2 传输速率控制机制

原RRTCC算法中已经提出了一套包含链路状态探测,可用带宽估计和发送速率控制功能的拥塞控制机制.但由于它并不能很好地对移动网络中偶尔出现的网络波动和真正的网络拥塞进行良好的区分,在对发送速率控制的过程中过于敏感,速率波动较大,进一步导致了与TCP流竞争过程中处于劣势的问题.因此本文基于对RRTCC算法进行改进,提出面向移动网络的传输速率控制机制,使WebRTC更适应于移动网络的特性,提升用户的QoE体验.本文提出的改进方法将使用原RRTCC算法中处于RTP数据接收端的Arrival-time Filter和Over-use Detector,着重对处于RTP数据发送端的Rate Controller进行改进.

在RTP流传输过程中,首先,由处于RTP数据接收端的Arrival-time Filter对接受到的RTP数据包进行延时和数据量分析,再用高斯白噪声过程和卡尔曼滤波器对数据进行处理,得到能够反映当前可用带宽的偏移量估计值 $m(i)$ ,然后由Over-use Detector对 $m(i)$ 和动态阈值 $\gamma_1(i)$ 进行比较,生成能够反映当前链路状态的信令消息,并发送给处于RTP数据发送端的Rate Controller进行发送速率的控制.

Rate Controller接受到的信令消息有3种:Over-use,Under-use和Normal,分别代表了3种链路状态.Over-use代表网络的传输队列(IP层)过长,认为拥塞发生了;Under-use代表传输队列接近为空,认为可用带宽很多;Normal则是介于它们两者之间的一种状态,认为当前传输队列大小适中,RTP流在稳定传输中.Normal信令每隔一段时间就会发送到RTP数据发送端,而Over-use和Under-use的信令消息则需要等待此状态被检测到连续超过 $\gamma_2$  s,且 $m(i) \geq m(i-1)$ 才会被发送.算法中各参数的定义与作用在原RRTCC算法中有详细说明,这里不再赘述.

网络拥塞的发生一般是根据网络抖动来判断的,而网络抖动往往伴随着延时的增加,丢包率的增加.这样的机制在链路质量相对稳定的有线网络中是适用的,但在链路质量不稳定的移动网络中,偶尔的网络波动不一定代表拥塞发生.因此,需要对 Rate Controller 中对信令消息的反应机制进行改进,使其对链路状态变化更加容忍,更充分地利用移动网络带宽,提高传输效率.

与有线网络不同,移动网络的种类和使用环境更复杂,造成网络抖动的原因有很多种,比如用户在行驶的列车上,或者用户在行走过程中移动设备与基站间存在信号障碍物,还有移动设备进行基站切换,这些情况都有可能造成 Rate Controller 收到 Over-use 信令消息.因此,对改进 Rate Controller 的关键就是其对信令消息的反应机制.

本文改进的 Rate Controller 有 3 种数据发送状态:Fast-increase,Moderate-increase 和 Decrease.这 3 种发送状态的转换关系见表 1.

Table 1 Transformation relation of sending rate states

表 1 发送状态转换关系

	$S_{rate} > P\_ESA_{bps}$	$S_{rate} < P\_ESA_{bps}$
Over-Use	Decrease	Decrease
Under-Use	Decrease	Fast-increase
Normal	Decrease	Moderate-increase

可见,当发送速率  $S_{rate}$  大于由式(2)计算出来的可用带宽估计值  $P\_ESA_{bps}$  时,无论接收到什么类型的信令消息,Rate Controller 都要进入 Decrease 状态,以此来保证发送速率永远不会大于  $P\_ESA_{bps}$  值,避免 WebRTC 对 TCP 流过度挤压,保证与其他 TCP 应用的友好竞争性.

当发送速率  $S_{rate}$  小于  $P\_ESA_{bps}$  时,则需要根据信令类型选择发送状态.在 Fast-increase 状态下,发送速率每秒最多提升 12%,如式(4)所示.

$$\begin{cases} esp = 1.2 \times \min(U_i, 1.0) \\ S_{rate}(i) = esp \times S_{rate}(i-1) \end{cases} \quad (4)$$

在 Fast-increase 状态下该式每秒被执行一次,其中,  $U_i$  代表当前 Fast-increase 状态距离上一次状态更新的时间差.而在 Moderate-increase 状态下,每收到一个 Normal 信令,发送速率就增加一个数据包.

假设  $P\_ESA_{bps}$  值一直大于发送速率,并且在实时流媒体会话刚开始搭建时,信令消息为 Under-use,则 Rate Controller 立即进入 Fast-increase 状态,快速提高发送速率,直到接收到 Normal 或者 Over-use 信令才改变发送状态.通过这种方式来实现流媒体平滑地输出.而当移动网络偶尔发生的网络抖动结束后,链路状态恢复正常,这时进入 Fast-increase 状态能够让发送速率快速增加,以此实现 WebRTC 在移动网络环境下相对有线网络更为激进的传输机制.

另一方面,考虑到移动网络链路不稳定的特点,会导致 Rate Controller 时常接收到 Over-use 信令,因此,Decrease 状态则应该针对不同的情况执行得更为谨慎.

经过实验与分析,我们总结了 3 个级别的移动网络环境下 Over-use 信令出现的特点.(1) 恶劣环境,比如用户坐在行驶的列车上,这种场景下的网络质量很差,Over-use 信令出现频率很高,会出现甚至无法收到信令的情况.(2) 较差环境,比如移动设备与信号基站间有大障碍物或者设备切换基站,这种场景下,Over-use 信令出现频率较高,但状态不会持续很久.(3) 一般环境,比如用户范围地移动或者有其他无线信号轻微地干扰,这种场景下只会零星地出现 Over-use 信令.

根据总结的规律,我们提出 Decrease 状态的发送速率衰减公式,如式(5)所示:

$$\begin{cases} D_{ef} = \begin{cases} 0.75 & R_{over} > 0.4 \\ 0.85 & 0.4 > R_{over} > 0.1 \\ 0.95 & R_{over} < 0.1 \end{cases} \\ S_{rate}(i) = D_{ef} \times S_{rate}(i-1) \end{cases} \quad (5)$$



其中,  $R_{over}$  代表每 50 倍 Normal 信令的时间周期内 Rate Controller 接收到的 Over-use 信令比例,  $D_{cf}$  代表了衰减率. 在 Decrease 状态下, 每接收到一个 Over-use 信令, 式(5)都会被执行, 重新计算一次  $D_{cf}$ .

而针对另一种情况, 由  $S_{rate} > P\_ESA_{bps}$  导致的 Decrease 状态,  $D_{cf}$  将是一个静态的值 0.8. 值得注意的是, 由 Over-use 信令和由  $S_{rate} > P\_ESA_{bps}$  下引发的两种 Decrease 状态能够同时存在,  $D_{cf}$  值选择数值更小的那个, 以达到快速降低发送速率的效果.

在原 RRTCC 算法中, 发送速率进入衰减状态后, 一直保持着 0.85 的衰减率. 相比之下, 我们提出的 Decrease 状态更灵活, 针对 Over-use 信令的出现频率动态选择衰减率, 能够区别对待网络拥塞和移动网络环境中网络波动的情况, 对传输速率的控制更为精准, 提升了用户的 QoE.

### 3.2 WebRTC 并发流管理机制

近年来 WebRTC 技术经过快速的发展, 已经能够提供对视频会议这种多 RTP 流并发服务的支持, 并保证良好的通信效果. 但我们在对 WebRTC 的性能测试过程中还是发现了一些问题: 当链路带宽不足时, WebRTC 会优先保证新加入 RTP 流的传输质量, 但会损坏已有 RTP 流的传输效果, 且处理 RTP 流的平衡所需时间较长, 对用户 QoE 的损害较大. 可见, 虽然 WebRTC 已经内置了码率自适应机制, 但仍需改进. 在研究过程中, 我们发现 RRTCC 并没有提出针对此问题的解决方案. 因此基于对问题的分析, 我们提出了优先保证用户 QoE 的 WebRTC 并发流管理机制.

本文提出的 WebRTC 并发流管理机制将运行在 WebRTC 会话的整个生命过程中, 根据对链路质量和可用带宽的估计, 对 RTP 流的加入, 运行与退出进行合理控制, 优先保证用户 QoE 的最大化. 值得注意的是, 该机制默认每个 RTP 流都是平等的.

在建立 WebRTC 会话前, 该机制需要进行以下的操作步骤.

Begin

Step 1. 根据用户提供的对应其 QoE 范围的视频参数(分辨率、帧率等), 生成各 QoE 等级对应的可用视频参数集(包括码率、帧率、分辨率和编码技术等).

Step 2. 使用原 RRTCC 算法中提出的 Arrival-time Filter, 结合 TCP 友好机制计算出当前链路可用带宽估计值  $A\_hat(i)$ .

Step 3. 在  $S_{rate}(i) < A\_hat(i)$  的条件下, 求优先满足用户最佳 QoE 的视频参数集最优解.

Step 4. 根据最优解建立 WebRTC 会话, 监听 RTP 流.

End

其中, 对用户定制的能够反映其自身所需 QoE 水平的视频参数使用 MOS 评价方法<sup>[16]</sup>, 自动将不同视频参数集划分为 5 个对应的 QoE 等级, MOS 评价方法见表 2. 为了保证 RTP 流与 TCP 流的平衡, 需要保证  $A\_hat(i) < P\_ESA_{bps}$ . 值得注意的是, 由于语音服务相对视频服务所需带宽的比例很小, 这里只考虑对视频参数的调整, 而如何根据可用带宽值  $A\_hat(i)$  求解最优视频参数集的算法研究已有很多, 各有优劣, 限于研究题目, 我们对此问题不做进一步的讨论分析.

Table 2 Meaning of MOS values

表 2 MOS 值含义

MOS 值	含义
5	非常好
4	好
3	一般
2	差
1	非常差

监听 RTP 流的主要任务包括获取 RTP 流的加入、退出和运行状态. 由于 WebRTC 已经内置了码率自适应机制, 在多个 RTP 流正常运行过程中对带宽自适应的操作已较成熟, 这里我们更关注 RTP 流加入和退出的情况, 处理



过程如图 4 所示.

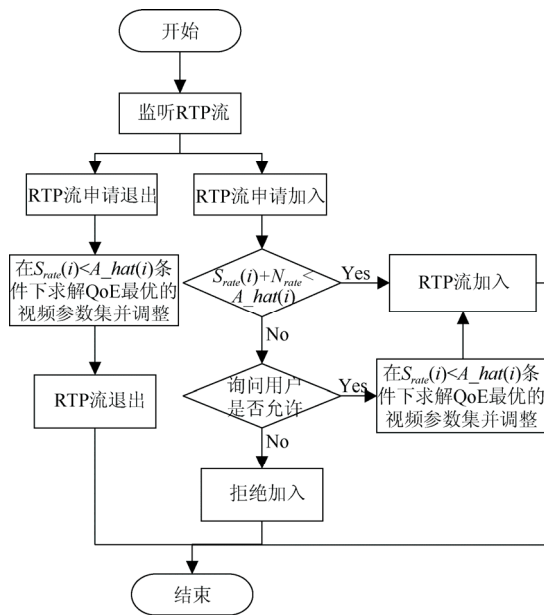


Fig.4 Flowchart of concurrent RTP flow processing

图 4 RTP 并发流处理流程图

图 4 中,  $N_{rate}$  代表加入新 RTP 流所需的带宽值,通过公式  $S_{rate}(i) + N_{rate} < A\_hat(i)$  判断 RTP 流的加入是否在可用带宽允许范围内,如果超出了可用带宽允许范围,则向用户提出询问是否愿意以降低视频质量为代价,允许新 RTP 流的加入,如果用户选择肯定,则通过公式  $S_{rate}(i) < A\_hat(i)$ ,即在所有 RTP 流所需总带宽小于可用带宽的条件下,求解优先满足最佳 QoE 的视频参数集,并根据此参数集对已存在的 RTP 流和即将加入的 RTP 流进行参数调整.

由于该机制默认所有 RTP 流都是平等的,在 RTP 流数量改变时,该机制对所有 RTP 流同步进行调整,避免了由于新 RTP 流的加入,导致某个已有 RTP 流通信质量受损的情况.此外,根据求解的 QoE 最优视频参数集能够立即对 RTP 流的传输速率进行调整,不再需要一段较长的速率自适应等待时间,并且求解过程优先保证了用户的 QoE 体验.

#### 4 实验与结果

Chrome 浏览器的开源版本 Chromium 包含了 WebRTC 模块和 RRTCC 算法的开源代码,我们通过对对其进行增加和修改以实现本文提出传输控制改进策略 MTCIS,并利用 WebRTC 官方提供的项目代码和 API 接口搭建应用服务器作为实验平台,使用的网络协议为 WIFI 802.11n.

实验 1 使用相同配置的 2 台 Android 设置和 2 台 PC,其中 1 台 Android 设备和 1 台 PC 使用 RRTCC 算法进行 P2P 视频通话,而另外的 1 台 Android 设备和 PC 则使用 MTCIS 策略进行 P2P 视频通话,2 台 Android 设备均同时由一位实验人员手持,并在有信号障碍物分布的一定范围内移动.在实验过程中实时采集传输速率、延时和丢包率,最后取平均值.实验中传输速率的变化情况如图 5 所示,实验结果数据见表 3.可见,在相同的网络环境下,与 RRTCC 相比,使用 MTCIS 的 WebRTC 流传输过程更稳定,传输速率更高,延时更低且离散程度更小,丢包率也更小.从实验员的角度进行观察,使用 MTCIS 的 WebRTC 服务在运行过程中更流畅,不易受小型障碍物的干扰.

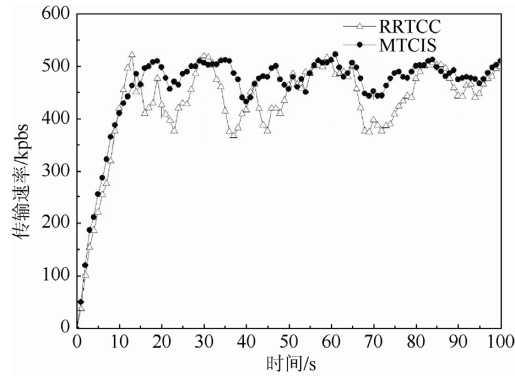


Fig.5 Experiment 1: RRTCC and MTCIS transmission rate changes  
图5 实验1:RRTCC与MTCIS传输速率变化情况

Table 3 Results of Experiment 1

表3 实验1结果数据

	传输速率(kbps)	RTT 延时(ms)	丢包率(%)
RRTCC	437.18±79.57	60.17±24.13	9.2
MTCIS	456.51±44.61	51.55±13.7	4.3

实验2在实验1的基础上,设置网络上/下行带宽分别为1M,并在2台Android设备的系统后台分别运行1个基于TCP长流的数据下载应用,以测试RRTCC和MTCIS对WebRTC与TCP流竞争环境下的性能表现.实验结果中,使用RRTCC算法控制的WebRTC流无法与TCP流合理竞争,过度退让,最终导致WebRTC会话失败,传输情况与图2相似.使用MTCIS策略的传输情况如图6所示.可见,经过改进后由MTCIS控制的WebRTC流更为强势,能够与TCP流良好竞争,能够做到一定程度上的平衡,较为稳定地进行数据传输.从实验员的角度观察,由MTCIS控制的WebRTC服务虽然视频质量有所下降,但仍然能够进行较为流畅的视频通话.

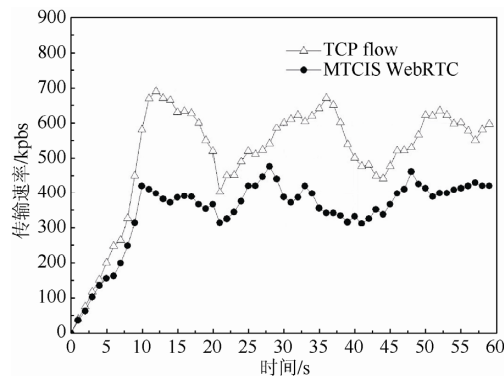


Fig.6 Experiment 2: Competition between TCP flow and MTCIS WebRTC  
图6 实验2:TCP流与MTCIS的WebRTC竞争情况

在实验3中,我们搭建了WebRTC视频会议应用,使用3台Android设备以10s的时间间隔陆续加入到视频会议中,网络上/下行带宽分别为1M,实验分两次进行,第1次3台Android设备全部运行使用RRTCC算法的WebRTC应用;第2次则全部运行MTCIS的WebRTC应用,并由实验员事先根据自己的喜好,在应用中定义了可接受的视频参数.实验过程中,当第3台设备请求加入视频会议时,可用带宽不足,使用RRTCC算法的WebRTC应用没有做出提示,直接允许加入,造成第1台加入的Android设备传输速率大幅下降,等待一段时间后恢复正常,数据传输过程与图3相似.而使用MTCIS的WebRTC应用则分别向已在通信的2台设备提出询问是否允许

加入,2 台设备分别选了拒绝和允许,选择拒绝的设备放弃与第 3 台设备建立连接;选择允许的设备传输速率变化过程如图 7 所示,降低已有流的传输速率后让第 3 台设备加入,从加入到多 RTP 流达到平衡所需时间更短,速率变化更平滑.从用户的角度观察,虽然整体画面质量轻微下降,但没有出现明显的卡顿现象.可见,与 RRTCC 相比,MTCIS 对 WebRTC 多 RTP 流并发的情况处理得更好,优先保证了用户的 QoE 体验.

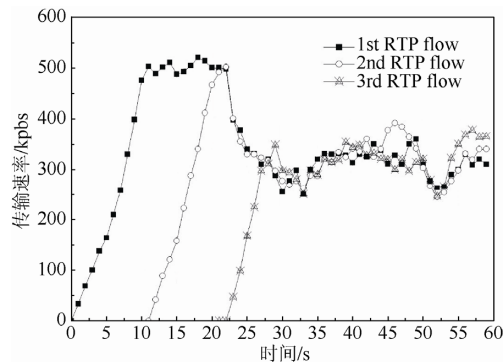


Fig. 7 Experiment 3: MTCIS handle concurrent RTP flows

图 7 实验 3:MTCIS 处理多 RTP 流并发的情况

## 5 结论

本文通过对 RRTCC 算法进行修改与补充,提出了 QoE 优先的移动传输控制改进策略 MTCIS.该策略包含面向移动网络的传输控制方法和 WebRTC 并发流管理机制两大模块.通过对原 RRTCC 算法中的 TCP 友好机制和速率控制器进行改进,使 WebRTC 更适应于链路质量不稳定的移动网络环境,在优先保证用户 QoE 的前提下与 TCP 流进行合理竞争,达到一定程度上的平衡.针对 WebRTC 多 RTP 流并发的问題,提出了用户 QoE 优先的并发流管理机制,在优先保证用户 QoE 的前提下,根据可用带宽值求解最佳 QoE 视频参数集,能够有效缩短平衡 RTP 流的时间,并减少视频卡顿现象.实验结果表明,与原 RRTCC 算法相比,在移动网络场景多流竞争的情况下,MTCIS 策略能够有效保证 WebRTC 的合理竞争性、稳定性和健壮性,提高了用户的 QoE 体验.

## References:

- [1] Zhang XH, Huang JQ, Wu KH, Lei ZB. Survey on real-time video and audio communication based on WebRTC. Computer Science, 2015,42(2):1-6,32 (in Chinese with English abstract).
- [2] Singh V, Lozano AA, Ott J. Performance analysis of receive-side real-time congestion control for WebRTC. In: Proc. of the 20th IEEE Packet Video Workshop. 2013. 1-8.
- [3] Holmer S, Lundin H, Carlucci G, De CL, Mascolo S. A Google congestion control algorithm for real-time communication. Internet-Draft, 2015. <https://tools.ietf.org/html/draft-alvestrand-rmcat-congestion-03>
- [4] Nurminen JK, Meyn AJR, Jalonen E, Raivio Y. P2P media streaming with HTML5 and WebRTC. In: Proc. of the 27th IEEE Computer Communications Workshops. 2013. 63-64.
- [5] Singh V, Ott J, Curcio IDD. Rate-Control for conversational video communication in heterogeneous networks. In: Proc. of the 6th IEEE World of Wireless, Mobile and Multimedia Networks Workshop on Autonomic and Opportunistic Communications. 2012. 1-7.
- [6] Zhu X, Pan R. NADA: A unified congestion control scheme for low-latency interactive video. In: Proc. of the 20th IEEE Packet Video Workshop. 2013. 1-8.
- [7] Satoh Y, Oki E. A scheme for available bandwidth estimation in simultaneous multiple-pair communications. In: Proc. of the 17th IEEE Asia-Pacific Conf. on Communications. 2011. 68-72.
- [8] Floyd S, Handley M, Padhye J, Widmer J. TCP Friendly Rate Control (TFRC): Protocol specification. RFC 5348, 2008.

- [9] Jammeh E, Mkwawa I, Khan A, Goudarzi M, Sun L, Ifeakor E. Quality of experience (QoE) driven adaptation scheme for voice/video over IP. *Telecommunication Systems*, 2012,49(1):99–111.
- [10] Hermanns N, Hamm L, Sarker Z. A framework and evaluation of rate adaptive video telephony in 4G LTE. In: *Proc. of the World Telecommunications Congress*. 2014. 1–6.
- [11] Qian C, Wang Y, Zhu X. A QoE-based sender bit rate adaptation scheme for real-time video transmission in wireless networks. In: *Proc. of the Int'l Conf. on Image and Signal Processing*. 2013. 6–10.
- [12] Rhinow F, Veloso PP, Puyelo C, Barrett S, Nuallain EO. P2P live video streaming in WebRTC. In: *Proc. of the World Congress on Computer Applications and Information Systems*. 2014. 1–6.
- [13] Nagy M, Singh V, Eggert L. Congestion control using FEC for conversational multimedia communication. In: *Proc. of the 5th ACM Multimedia Systems Conf.* New York: ACM Press, 2013. 191–202.
- [14] Kilinc C, Andersson K. A congestion avoidance mechanism for WebRTC interactive video sessions in LTE networks. *Int'l Journal of Wireless Personal Communications*, 2014,77(4):2417–2443.
- [15] Brooks P, Hestnes B. User measures of quality of experience: Why being objective and quantitative is important. *IEEE Network*, 2010,24(2):8–13.
- [16] Fiedler M, Hossfeld T, Tran-Gia P. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, 2010,24(2):36–41.

#### 附中文参考文献:

- [1] 张向辉,黄佳庆,吴康恒,雷志斌.基于 WebRTC 的实时音视频通信研究综述. *计算机科学*,2015,42(2):1–6,32.



葛志辉(1978—),男,河北唐山人,博士,教授,CCF 专业会员,主要研究领域为无线网络,移动计算.



岑霄(1991—),男,硕士生,主要研究领域为无线网络,移动计算.



李陶深(1957—),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为无线 mesh 网络,分布式数据库,遗传优化设计,云计算.



叶进(1970—),女,博士,教授,主要研究领域为无线传感器网络,移动计算.