

Research on Congestion Control of Network Video Based on RTP/RTCP

Juan Du

School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications,
Nanjing Jiangsu
Email: 18252062906@163.com

Received: May 22nd, 2016; accepted: Jun. 5th, 2016; published: Jun. 13th, 2016

Copyright © 2016 by author and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

With the rapid development of real-time transmission technology and the application of network video of the streaming media, video traffic has become the main contributor of network traffic. However, due to the limited bandwidth of the network, more and more people pay attention to the problem of network video congestion in the process of real-time transmission of streaming media. In this paper, through the in-depth study of RTP video in the framework of the streaming media transmission, based on its theory, a set of RTCP congestion strategies for RTP video stream is studied, and the feasibility of the strategy is verified in the test environment.

Keywords

RTP/PTCP, Streaming Media, Video Transmission, Congestion Control

基于RTP/RTCP的网络视频拥塞控制研究

杜娟

南京邮电大学通信与信息工程学院, 江苏 南京
Email: 18252062906@163.com

收稿日期: 2016年5月22日; 录用日期: 2016年6月5日; 发布日期: 2016年6月13日

摘要

随着流媒体实时传输技术和网络视频应用的迅猛发展，视频流量已成为网络流量的主要贡献者。然而有限的网络带宽导致流媒体实时传输过程中网络视频拥塞问题越来越受到人们的关注。本文通过对流媒体传输框架下RTP视频的深入研究，在其理论上研究出了一套针对RTP视频流的RTCP拥塞策略，并在测试环境下验证了该策略的可行性。

关键词

RTP/PTCP, 流媒体, 视频传输, 拥塞控制

1. 引言

随着多媒体和网络技术的飞速发展，视频服务也越来越盛行，网络上当前视频数据的传输大都是采用流媒体的形式，流媒体的传输需要合适的传输协议，比较成熟的流媒体传输一般采用 RTP/UDP 来传输实时信息。

为何要在 UDP 协议上进行实时数据的传输呢？这是因为 UDP 协议是非面向连接的，它本身并不能做任何校验，比较注重数据的传输速度。因此，对于对传输质量要求不是很高，而对传输速度则有很高要求的视音频流媒体文件来说，采用 UDP 协议则更适合 RTP 报文的传输。

然而随着视频流媒体文件传输量的增大，势必容易引起网络达到饱和，而一旦网络发生拥塞、延迟和丢包等情况，视频等实时流量对这些参数的变动又十分敏感，则会对实时视频传输造成严重影响。在网络传输领域中经常采用的拥塞控制方法有两种。一种是基于窗口的，它以数据包个数为单位，使用缓慢增加拥塞窗口来获取与可用网络带宽的匹配，当检测到网络拥塞时，就迅速减少拥塞窗口的大小，以减少和避免网络冲突，TCP 传输协议就是采用了这种方法。另一种是基于速率的，它以每秒发送的比特数为单位，先估计网络的可用带宽，然后调整发送数据的速率，试图使视频传输的网络带宽需求与该连接链路的可用带宽相匹配，来减少或避免拥塞的发生。显然，由于基于窗口的控制方法重传延时太长及抖动问题难以适应于视频实时传输，因此基于速率的拥塞控制在视频的实时传输中得到广泛的应用。

本文讨论的拥塞控制方法就是基于速率的，实时流传输协议 RTP 详细说明了在互联网上传递音频和视频的标准数据包格式，它与传输控制协议 RTCP 配合使用，成为流媒体技术最普遍采用的协议之一[1]。因此有必要研究基于 RTP/RTCP 的实时流媒体传输的拥塞控制策略。

2. 网络拥塞的产生

在因特网上，当一个子网或者其中的一部分出现太多分组时，网络性能开始下降，诸如网络延时加大、丢包率上升、吞吐量下降等，这种情况即成为拥塞。导致拥塞产生的原因通常是当前的负载(临时性地)超过了(网络中的某一部分)资源的处理能力，或者说用户提供给网络的负载大于网络资源的容量和处理能力[2]。

网络拥塞控制研究一直都是 Internet 的一个核心研究问题。当用户注入网络的流量低于网络传输用量时，网络资源得不到充分利用；在用户业务流量等于或者稍稍大于网络传输流量时，网络资源得到了充分利用，但在这种情况下，如果持续处于过载状态，则会使网络处于崩溃边缘[2]。

网络拥塞往往是由许多因素引起的，而拥塞控制所要做的都有一个前提，就是网络能够承受现有的

网络负荷。拥塞控制是一个全局性的过程，涉及到所有的主机、路由器，以及与降低网络传输性能有关的所有因素。进行拥塞控制需要付出代价，首先需要获得网络内部流量分布的信息。在实施拥塞控制时，还需要在节点之间交换信息和各种命令以便选择控制的策略和实施控制，这样就产生了额外的开销[2] [3]。拥塞控制有时需要将一些资源(如缓冲器、带宽等)分配给个别的用户单独使用，这样就使得网络资源不能实现更好地实现共享。因此，在设计拥塞控制策略时，必须全面衡量得失。

3. UDP 协议

UDP 协议在 IP 数据报服务的基础上增加了复用和分用的功能以及差错检测的功能。UDP 是面向无连接的，因此，在发送开始和结束时都不需要建立连接，这样一来，网络传输的开销和传输时延都会降低，有利于流媒体数据的传输[4]。由于 UDP 没有拥塞控制机制，因此当网络出现拥堵时，流媒体数据发送服务器并不会降低发送速率，这非常有利于需要稳定速率发送数据的应用。

4. RTP/RTCP 协议分析

RTP(Realtime Transfer Protocol)和 RTCP(Realtime Transfer Control Protocol)协议是端到端基于组播的应用层协议。其中 RTP 用于传输数据，RTCP 用于控制和管理 RTP 传输，两者协同工作，能有效地提高网络的实时数据传输效率。

RTP 由 IETF 的多媒体工作小组于 1996 年在 RFC1889 中公布，之后于 2003 年在 RFC3550 中更新。RTP 协议规定了互联网上传输音频和视频的标准数据包格式，它可以提供单播或组播形式的实时流媒体数据传输服务。由于 RTP 没有 Qos 保证，它不能保证数据的可靠传输和拥塞控制，而必须依靠 RTCP 来提供这些服务。RTP 通常工作在 UDP 之上，当应用程序开始一个 RTP 会话时，应用程序将确定一对目的传输地址[4]。目的传输地址由一个网络地址和一对端口组成，两个端口分别和 RTP、RTCP 关联，这样 RTP/RTCP 数据就能够正确传输。RTP 数据发向 1025 到 65,535 之间一个未使用的偶数的 UDP 端口，而对应的控制信号 RTCP 数据发向相邻的奇数 UDP 端口(偶数的 UDP 端口+1)，这样就构成一个 UDP 端口对。RTP 的发送过程如下所示，接收过程则相反：

- 1) RTP 协议从上层接收流媒体信息码流(如 H.263)，将其封装成 RTP 数据包；RTCP 从上层接收控制信息，将其封装成 RTCP 控制包。

- 2) RTP 将 RTP 数据包发往 UDP 端口对中偶数端口；RTCP 将 RTCP 控制包发往 UDP 端口对中的奇数端口。

5. 基于 RTP/RTCP 拥塞控制机制

5.1. RTP/RTCP 拥塞控制原理

RTCP 是和 RTP 一起工作的控制协议，单独运行在低层协议上，由低层协议提供数据与控制包的复用。在 RTP 会话期间，每个会话参与者向所有其他参与者周期性地发送 RTCP 控制包[5]。RTCP 的功能主要有以下四点：

1. 为应用程序提供会话质量或者广播性能质量的信息

每个 RTCP 信息包封装着发送端或接收端的统计报表。这些信息包括发送的信息包数目、丢失的信息包数目和信息包的抖动等情况，这些反馈信息反映了当前的网络状况，对发送端、接收端或者网络管理员都非常有用。

2. 确定 RTP 用户源

RTCP 为每个 RTP 用户提供了一个全局唯一的称为规范名称(Canonical Name)的标志符 CNAME，接

收者使用它来追踪一个 RTP 进程的参加者，同时，当 RTP 需要进行音视频同步的时候，接收者就需要使用 CNAME 来关联同一发送者的音视频数据。

3. 控制 RTCP 传输间隔

由于每个对话成员定期发送 RTCP 信息包，随着参加者不断增加，RTCP 信息包频繁发送将占用过多的网络资源，为了防止拥塞，必须限制 RTCP 信息包的流量，控制信息所占带宽一般不超过可用带宽的 5%，因此需要调整 RTCP 包的发送速率。由于任意两个 RTP 终端之间都互发 RTCP 包，因此终端的总数很容易估计出来，应用程序根据参加者总数就可以适当调整 RTCP 包的发送速率。

4. 最小进程传输控制信息

这项功能使参加者可以自由进入或离开较为松散的会话进程，没有成员控制或参数协调。发送端报告 SR 含有不同媒体流间的同步信息，以及已经发送的数据报和字节的计数，接收端根据这些信息可以估计出实际的数据传输速率。另一方面，接收端会向所有已知的发送端发送接收端报告 RR，该报告含有已接收数据报的最大序列号、丢失的数据报数目、延时抖动和时间戳等重要信息，发送端应用根据这些信息可以估计出往返时延，并且可以根据数据报丢失概率和时延抖动情况动态调整发送速率，以改善网络拥塞状况，或者根据网络状况平滑地调整应用程序的服务质量[6]。

下面以 SR 报告为例给出 SR 分组的主要内容，如图 1。

从图 1 中可以看到，SR 分组主要包括三部分内容：头部、发送端信息、接收报告块。包类型通过头部中的 PT 字段进行说明，除包类型代码外，SR 与 RR 间唯一的差别是源报告包含有一个 20 字节发送者信息段。

5.2. 本文提出的网络拥塞控制方法

网络拥塞控制的一般过程为：1) 第一步是得到当前网络的性能参数，如网络时延、分组丢失率、传输速率、到达抖动等；2) 第二步是判断网络是否达到拥塞态，涉及网络参数阈值的配置问题；3) 第三步则是根据所处状态实施对应的控制策略。RTP 视频流拥塞控制流程如图 2 所示。

在网络处于拥塞状态时，可以采用基于发送端速率控制和接收端会话控制策略。一方面是经计算处理后向数据源发送控制信息，动态调整数据发送速率，网络拥塞时降低发送速率，网络良好时则增大发送速率，这种策略需要数据源端进行配合方可有效；另一方面是利用访问控制工具 iptables，借助其 connlimit 模块对 udp 会话数进行限制，或者根据视频流量分布直接限制某些 IP 的收发速率，由于系统是并联在原有网络中，因此需要将定制的 iptables 规则发送到与其连接的网关上进行间接控制。connlimit 模块主要用来限制内网用户的网络使用，能够限制每个客户端 IP 的并发连接数，即每个 IP 同时连接到一个服务器的个数。UDP 不同于 TCP，其中没有 SYN 之类的连接标志，但 iptables 对此提供一种连接状态机制，能够指定并记住为发送或接收信息包所建立的连接的状态，并且定义了四种有效状态。

ESTABLISHED, INVALID, NEW, RELATED，其中 ESTABLISHED 指出该信息包属于已建立的连接，该连接一直用于发送和接收信息包并且完全有效；INVALID 指出该信息包与任何已知的流或连接都不相关联，它可能包含错误的头或数据；NEW 意味着该信息包已经或将启动新的连接；RELATED 表示该信息包正在启动新连接，以及它与已建立的连接相关联信息。其参数和用法如图 3 所示。

5.2.1. 计算网络性能相关参数

SR 源报告包和 RR 接收者报告包用于提供接收质量反馈，提供的有效信息如表 1 所示。

RR 针对每个信源都提供信息包丢失数、已收信息包最大序列号、到达时抖动、接收最后一个 SR 的时间、接收最后一个 SR 的延迟等信息。SR 不仅提供接收质量反馈信息(与 RR 相同)，而且提供 SSRC 标识符、NTP 时间戳、RTP 时间戳、发送包数以及发送字节数等[6]。

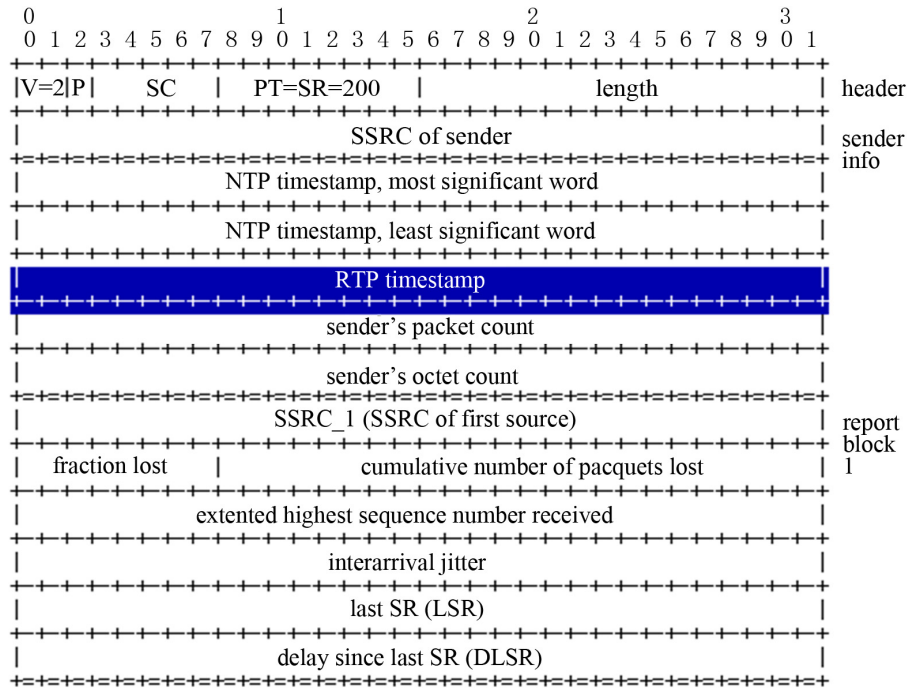


Figure 1. The sender report-SR grouping package

图 1. 发送端报告-SR 分组包

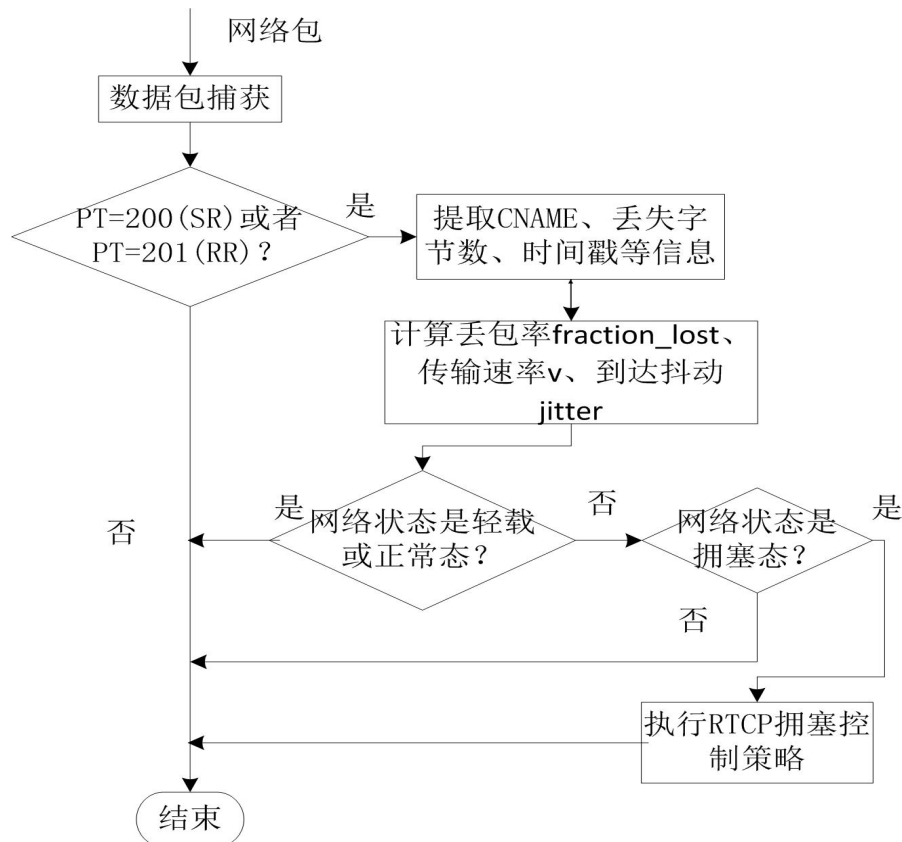


Figure 2. Congestion control flow of RTP video stream

图 2. RTP 视频流拥塞控制流程


```

--connlimit-aboven      # 连接数限制为 n 个
--connlimit-mask m     # 主机的掩码，默认是 m=32，即每个 IP
//限制 192.168.1.2 和 192.168.1.3 的 udp 连接数不超过 30
iptables-N UDPLIMIT
iptables-I FORWARD -s 192.168.1.2 -p! tcp -m connlimit--connlimit-above 30 -j
UDPLIMIT
iptables-I FORWARD -s 192.168.1.3 -p! tcp -m connlimit--connlimit-above 30 -j
UDPLIMIT
iptables-A UDPLIMIT -m conntrack --ctstate NEW --ctprotoudp -j DROP
iptables-I FORWARD -p udp --dport 53 -j ACCEPT
//第 1 条：是在 iptables 的 filter 链中创建一条自定义子链 UDPLIMIT，以便
存放超过连接数后的 udp 数据包
//第 2-3 条：当 192.168.1.2 和 192.168.1.3 的连接数超过 30 的时候，把后来
的数据包放进 UDPLIMIT 中
//第 4 条：丢弃状态为 NEW 的 UDP 数据包。（因为是新建 UDP 数据包）
//第 5 条：在 FORWARD 链所有规则前面放一条允许 DNS 数据包通过的规则
（防止 UDP 连接满后无法浏览网页）
    
```

Figure 3. Parameters and usage
图 3. 参数和用法

Table 1. The Feedback information of SR and RR quality
表 1. SR 和 RR 质量反馈信息

	包丢失数	最大序列号	到达抖动	NTP 时间戳	RTP 时间戳	发送包数	发送字节数
SR	√	√	√	√	√	√	√
RR	√	√	√	N/A	N/A	N/A	N/A

本文提供三个参数来评估当前网络传输性能，分别是包丢失率、载荷平均传和达抖动。丢失包累计数给出间隔期间丢包的数量，而所收到扩展的最后一个序列号给出间隔期间希望发送的包数量，两者之比等于间隔期间包丢失百分比，计算过程如下[7]：

$$\begin{aligned}
 \text{丢包率} &= \text{包丢失累计数} / \text{期望发送的包数} \\
 \text{丢失包累计数} &= \text{期望发送的包数} - \text{收到的包数} \\
 \text{期望发送的包数} &= \text{最后一个序列} - \text{第一个序列} + 1
 \end{aligned}$$

发送字节数是发送者从发送 RTP 数据包开始，直到产生 SR 包，传输的载荷字节总数，可用于估计载荷平均传输速率：

$$\text{载荷平均传输速率} = \text{传输的载荷字节总数} / \text{传输载荷所用的时间}$$

RTP 视频业务传输过程中的抖动示意图如图 4 所示。

设 Rec_i 是包 i 到达时的 RTP 时间戳， $Sent_i$ 是包 i 发出时的 RTP 时间戳，则到达抖动的计算公式为：

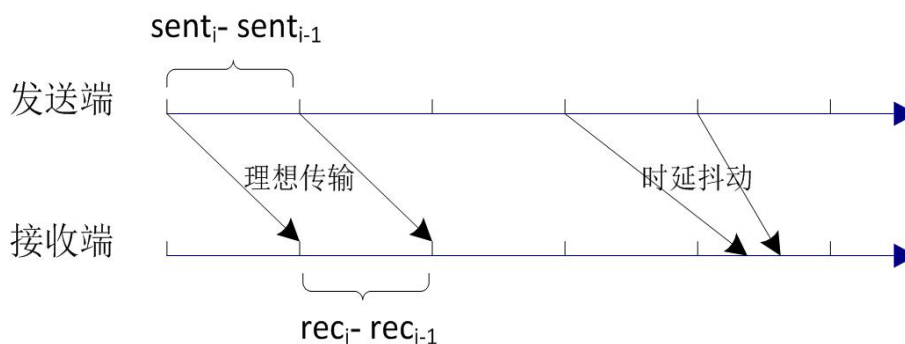


Figure 4. The schematic diagram of RTP transmit jitter
图 4. RTP 传输抖动示意图

$$\text{新到达抖动} = \text{旧到达抖动} + (\text{瞬时抖动} - \text{旧到达抖动})/16$$

瞬时抖动 = |(接收第 i 个数据包的时间戳 - 接收第 $i-1$ 个数据包的时间戳) - (发送第 i 个数据包的时间戳 - 发送第 $i-1$ 个数据包的时间戳)|

5.2.2. RTP 视频流拥塞控制策略

可以对丢包率、载荷平均传输速率、到达抖动分别设置相应的阈值，然后借助这些阈值将网络状态分为轻载、正常、拥塞三种状态。设定丢包率阈值 $L1$ 、 $L2$ ，且 $0 < L1 < L2 < 1$ ；载荷平均传输速率阈值 $V1$ 、 $V2$ ，且 $0 < V1 < V2$ ；到达抖动阈值 $J1$ 、 $J2$ ，具体控制策略为：

当丢包率 $< L1$ 或载荷平均传输速率 $< V1$ 或到达抖动 $< J1$ 时，轻载状态，可以逐步增大传输数据，直到达到正常状态或达到数据源的最大发送能力；

当丢包率 $\geq L2$ 或载荷平均传输速率 $< V2$ 或到达抖动 $< J2$ 时，拥塞状态，必须迅速减小数据传输量，降低源端的发送速率；或减少视频会话数量，保证用户的最低需求，直到回到正常状态；

当 $L1 \leq \text{丢包率} < L2$ 或 $V1 \leq \text{载荷平均传输速率} < V2$ 或 $J1 \leq \text{到达抖动} \ll J2$ 时，正常状态，表示当前网络可以正常接收视频数据，网络性能良好。

为避免 QoS 振荡的影响， $L1$ 应取较低值， $L2$ 的取值标准是丢失的包对解码后的视频播放的影响是用户可接受的。这里取 $L1 = 2\%$ ， $L2 = 4\%$ 。到达抖动比较均方根值 RMS (Root Mean Square) 更有意义，这里取 $J1 = 5.71 \text{ ps RMS}$ 、 $J2 = 50 \text{ ps RMS}$ 。由于不同环境的网络带宽各自的差异性，相应的载荷的平均传输速率也各有不同，因此，这里的两个阈值 $V1$ ， $V2$ 由用户根据当前网络性能或者经验值自行设定。

6. RTP 视频流拥塞控制策略的测试

6.1. 测试环境的部署

本文提出的拥塞控制策略的测试环境主要包括以下四个部分，分别是：

1. 硬件环境

服务器采用 IBM System x3650 M5(5462I05)系列机架式服务器，其单核 CPU 型号为 Xeon E5-2603v3，8GB 的 DDR4 内存容量，1TB 的硬盘容量，搭载 $4 \times 1 \text{ GbE}$ 千兆以太网接口。

2. 软件环境

在本次测试实验中，该服务器使用 RedHat 服务器版本的 Linux 操作系统即 rhel-server-5.5-i386；底层抓包需要分别顺序安装 Libpcap、Libnids、Libnet 开发包；在数据库选择方面使用 MySQL 关系型数据库，版本号为 mysql-5.0.56；为提供 Web 访问服务，还在服务器端安装了最流行的 Web 服务器软件 Apache 1.3.37 版本；最后为支持前端 Web 开发，系统使用 PHP 通用开源脚本语言，安装版本为 php-5.6.0。

3. 网络环境

主要涉及服务器两张网卡的相关网络配置,如设置IP地址(服务器通过直接修改Linux下的配置文件,交换机可以通过物理端口或VLAN逻辑端口设置,要确保相互连接的两端属于同一个网段中),子网掩码,广播地址,网关等重要参数。

4. 系统安装

本系统将前端与后台以文件夹的形式打包在一起,解压之后即可使用。系统服务分为前端Web服务和后台业务服务。通过开启Linux下的Apache服务即可使用Web相关功能,后台服务与普通的Linux软件安装过程相类似,通过在src目录下执行Linux命令make和make install完成后台服务的安装,然后执行bin目录下的可执行文件即可启动视频拥塞控制功能。

6.2. 测试结果分析

在测试环境下根据当前网络的三个状态轻载、正常、拥塞,分别对视频会话做出相应处理,测试过程首先需要获得当前网络的性能参数,根据第5节的内容,计算的实时参数如图5所示。

图5是根据丢包率做出的实时动态图,其标注了在这段时间内的最大和最小丢包率的点,图中绿色虚线代表轻载状态的阈值,红色虚线代表拥塞状态的阈值,这样便于我们清晰直观地查看特殊状态值并及时做出判断。本次测试条件是在丢包率达到拥塞阈值4.0%时启动后台拥塞控制策略,可以看到由于历史数据包的存在以及服务响应时延,在8:23:31之后丢包率呈现明显下降趋势,证实了该策略的有效性。

第二个性能参数是包到达抖动,动态如图6所示。

到达抖动是决定视频能否正常播放的重要指标,图中标注了在这段时间内的最大和最小包到达抖动的点,同时也设置了轻载和拥塞两种状态的阈值5.7和20,分别用绿色虚线和红色虚线来表示。图中可以看到在2:58:48时刻抖动值达到系统设置的拥塞阈值,后台控制策略执行之后从2:58:52开始生效,有效地降低了包到达抖动。

最后一个性能参数是载荷传输速率,详情见图7。

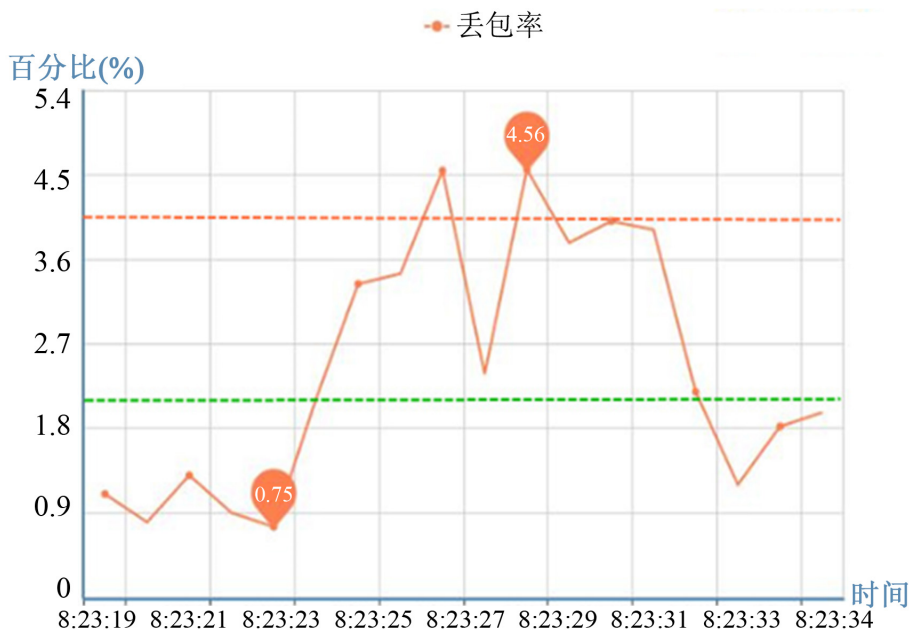


Figure 5. Packet loss rate

图5. 丢包率

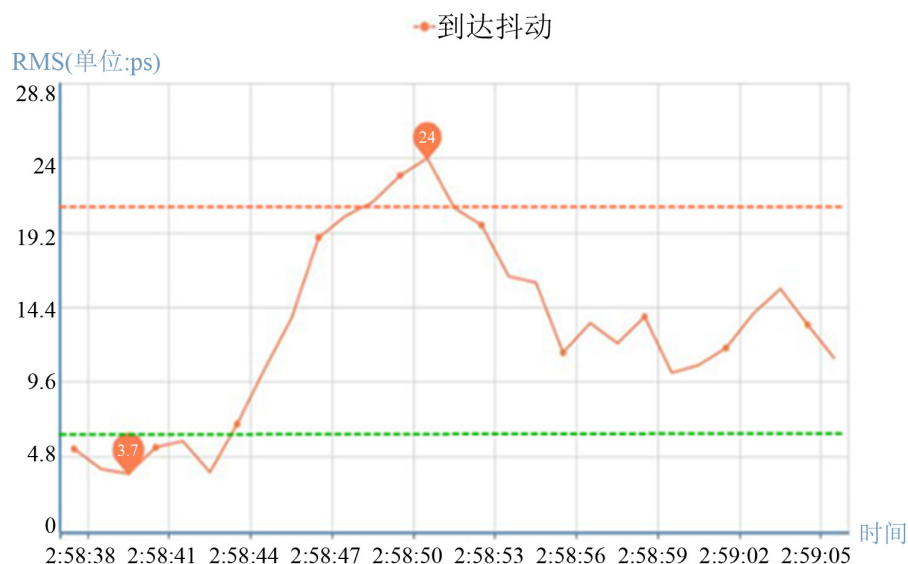


Figure 6. Arrival jitter

图 6. 到达抖动



Figure 7. Load transmit rate

图 7. 载荷传输速率

载荷传输速率动态图记录了视频载荷在网络上的实时传输速率，是描述网络带宽的重要指标，在本次的测试实验中，该网络的平均传输速率在 260 Kbps 左右，拥塞阈值设置为 120 Kbps，在执行策略之后，可以看到从 10:41:41 时刻开始传输速率有明显提高且在均值附近波动。

7. 结语

本文提出的基于 RTP/RTCP 的网络视频拥塞方法在测试环境下证明了其可行性。在实际应用中将首先通过数据包捕获得到 RTCP 的 SR 报文和 RR 报文，然后通过字段解析获得对应值，并且根据本文提出的计算公式分别求得丢包率、载荷平均传输速率和到达抖动三个参数的值，与设置的阈值进行比较，准确定位当前网络性能区间，根据不同的状态执行对应的策略，这一方法可应用于企业局域网的网络视频监控系统中。

参考文献 (References)

- [1] 万正道, 杨永超. 基于 RTP 的实时多媒体数据网络传输拥塞控制策略[J]. 自动化技术与应用, 2010, 29(7): 44-46.
- [2] 贺秋燕. 一种网络视频监控系统的研究与实现[D]: [硕士学位论文]. 广州: 华南理工大学, 2012.
- [3] 张彧也. 基于无线网络的流媒体传输拥塞控制的研究[D]: [硕士学位论文]. 重庆: 重庆大学, 2014.
- [4] 李浩炜. 基于 RTP 的实时 H.264 视频监控系统的流传输研究[D]: [硕士学位论文]. 长春: 长春理工大学, 2014.
- [5] 谢希仁. 计算机网络(第五版) [M]. 北京: 电子工业出版社, 2011: 310-312.
- [6] 褚典, 江春华, 郝宗波, 江维. 基于 SIP、RTP/RTCP 和 RTSP 协议的视频监控系统[J]. 计算机与现代化, 2013(11): 139-142.
- [7] 刘国柱, 王洪林. RTCP 反馈和缓冲区的传输控制算法研究[J]. 计算机工程与设计, 2010, 31(12): 2704-2706.