

# 网络抖动实时测量方法的实现与对比

王萌, 王翊

(中国传媒大学 广播电视数字化教育部工程研究中心 北京 100024)

**摘要:** 带有时间戳的延迟因子算法(TS\_DF)是用来衡量视频流和音频流在应用过程中的网络抖动及其累积效应的工具。该算法适用于 MPEG 传输流、音频传输流以及未压缩的视频传输流,解决了视频传输质量指标中对于可变比特媒体流(VBR)的延迟因素测量。MDI:DF 是 MDI 标准中对于视频传输质量中延迟因素的测量方法。本文对两种算法进行了实现和对比分析。实验证明 TS\_DF 算法适用于带有时间戳的媒体流,突破了传统测量方法中需要区分恒定比特率流(CBR)和可变比特率流的局限性,同时具有更高的精确度和稳定性。

**关键词:** 网络抖动; RTP 流; MDI:DF; 时间戳延迟因子

**中图分类号:** TP391 **文献标识码:** A **文章编号:** 1673-4793(2016)01-0027-05

DOI:10.16196/j.cnki.issn.1673-4793.2016.01.006

## Implementation and Comparison of Two Standardized Jitter Measurement Methods

WANG Meng, WANG Xuan

(ECDAV, Communication University of China, Beijing 10024)

**Abstract:** Time-stamped delay factor(TS-DF) algorithm can be used as a tool to measure IP network jitter and its cumulative effect for applications such as video and audio streaming. This algorithm is suitable for measuring IP network jitter in MPEG Transport streams over IP, voice over IP, as well as uncompressed video over IP. The delay factor of media delivery index raised in RFC4445 proposes another method. It defines the Delay Factor as the maximum difference, observed at the end of each media stream packet, between the arrival of media data and the drain of media data. In this paper, we try to conduct the two algorithms and prove that the TS\_DF algorithm is more advanced with high precision.

**Keywords:** jitter; RTP; MDI:DF; time-stamped delay factor

### 1 引言

随着网络协议电视(IPTV)和网络电话(Voice Over IP)的广泛应用,越来越多的视频和语音流量进入了 IP 网络。为了有效利用和高效管理这些服务,网络运营商和服务提供商对于网络质量的了解具有重要的意义。

目前,主流网络传输质量的衡量指标有时延、抖

动和丢包率等。为了准确的了解网络传输质量,需要对这些衡量指标进行测量。现有的视频传输质量测量及评估算法中比较通用的方法分别是由 RFC3550<sup>[1]</sup>、RFC4445<sup>[2]</sup>、EBU-TECH3337 提出的。

在实际应用中,2006 年发布的 RFC4445 标准中提出的 MDI:DF,局限于恒定比特流的网络传输。而现实中,越来越多的网络传输媒体,尤其是视频传输为了节约带宽提高其利用率,多采用可变比特流模式进行传输。2010 年在 EBU-TECH3337 中提出

收稿日期:2015-08-03

作者简介:王萌(1990-),女(汉族),内蒙古赤峰人,中国传媒大学硕士生。E-mail:358934973@qq.com,

的  $TS\_DF^{[3]}$ , 相对于传统的 UDP 测量该测量方法同时适用于 CBR 和 VBR, 具有测量结果更加准确的显著优势。

本文首先分析了网络传输对 IPTV 视频质量的影响, 然后对比分析现有的网络抖动测量方法<sup>[4]</sup>, 详细介绍了 MDI:DF 和  $TS\_DF$  计算原理。通过广域网模拟器 (WAN\_emulator 简称 WANem) 配置网络环境对两种方法进行了仿真分析, 证明  $TS\_DF$  算法适用于带有时间戳的媒体流, 突破了传统测量方法中需要区分恒定比特率流和可变比特率流的局限性, 同时具有更高的精确度和稳定性。

## 2 网络传输对视频质量的影响

网络传输过程中的时延、抖动、数据包丢失、发送和接收到的数据包的顺序不一致而引起的混乱等都会对视频造成损伤<sup>[5]</sup>。数据包丢失和顺序混乱所引起的视频质量下降, 在解码器端引入相应的补偿机制就可以解决。而网络传输过程中的时延、抖动对视频所造成的损伤是需要提前测量有关的时延、抖动幅度进而设置适应大小的缓冲区来解决。若在网络传输过程中存在时延而时延又是恒定的话, 那么这将会导致视频播放的推迟。但若在传输过程中存在抖动, 这将会影响到视频的质量, 使画面停滞或者丢失。所以, 为了消除因网络抖动所带来的视频质量的下降, 视频播放时解码器需要缓存一定的数据, 即解码器要有一定的缓冲区。

抖动越大, 需要消除抖动的缓冲器就越大。但在采用较大缓冲器解决抖动的同时, 也带来了更大的延迟。相反, 若缓冲器过小, 那么较大的抖动就会导致缓冲器上溢或下溢, 从而导致 IP 包的丢失, 用户的体验就会变差。所以需要寻找一种准确稳定的测量方法来对网络抖动进行测量, 解决上述问题。

## 3 网络抖动测量方法综述分析

### 3.1 RFC3550 抖动计算

在 IP 网络的实时传输媒体 (通常包括多播、单播和广播形式) 中, 一种传统的抖动测量方法是由 RFC3550 定义的。该方法是基于 RTP 报头中的时间戳时间和 IP 包到达时间的关系实现的。

实时传输协议 (RTP)<sup>[6]</sup> 通常与实时传输控制协议 (RTSP) 配套使用。在实时传输协议 (RTP) 中, 发送数据包的源设备完成对每个包的编号和时间戳。在 RFC3550 中定义的抖动测量算法正是基于 RTP 协议, 应用时间戳字段, 通过连续数据包间相对传输时间 (Relative Transmit Time) 的概念实现的。

如果用  $D(i-1, i)$  表示连续两个包的相对传输时间, 则抖动的计算方法可定义如下。

$$J(i) = J(i-1) + (|D(i-1, i) - J(i-1)|) / 16 \quad (3-1)$$

值得注意的是, 式中对于  $D(i-1, i)$  的运算使用的是绝对值, 原因在于两个包的延迟到达和提前到达都会造成媒体传输的抖动。在实际应用中, 由于数据包的延迟到达和提前到达具有相反的影响, 因此相对传输时间对于缓冲区分析具有重要的意义。所以该算法取绝对值的运算具有一定的缺陷。

另外, 式 (3-1) 中增益系数 1/16 有效的实现了低通滤波功能, 消除了高频抖动变化所产生的高峰值。在测量网络抖动的累积效应及其缓冲影响时高峰值是同等重要的, 因此这种算法不能有效监测到解码器遇到的缓冲区上溢或下溢的情况。

### 3.2 MDI:DF

流媒体应用具有实时性的特点。在流媒体通过 IP 网络传输的同时, 终端解码器在消耗已接收到的媒体流信息。IP 网络传输媒体流出现抖动表现为同一媒体流的 IP 封包传输的间隔不均匀。RFC4445 提出了视频传输质量指标中的延迟因素的测量方法, 即 MDI:DF。该方法具有独立协议的优势。它是基于测量名义媒体速率和到达速率之间的流动不平衡性这一参数实现的, 这个参数是对累积抖动的估计。

RFC4445 提出的延迟因素 (DF) 算法是在每个媒体流封包结束处观察到的, 到达的媒体数据和流出的媒体数据之间的最大差值。即计算时间间隔期间流动速率不平衡的最大观测值。

DF 计算方法如下:

$$VB(i, pre) = \text{sum}(S_j) - MR * T_i (j=1, 2 \dots i) \quad (3-2)$$

$$VB(i, post) = VB(i, pre) + S_i \quad (3-3)$$

$$DF = [VB(max) - VB(min)] / MR \quad (3-4)$$

式中定义 VB (Virtual Buffer) 为一个虚拟缓冲区,用来缓冲一个流的接收封包。在一个计算间隔期间,当一个封包  $P(i)$  到达时,计算两个 VB 的值,即  $VB(i,pre)$  和  $VB(i,post)$ 。分别代表  $P(i)$  包到达前和到达后的缓冲区大小。 $S_j$  是第  $j$  个封包的媒体负载大小, $T_i$  是在此计算间隔中封包  $i$  到达的相对时间; $MR$  是媒体码率。

MDI:DF 描述的延迟因素计算方法,可以测量瞬时和长期行为网络流媒体的视频传输质量,即被测视频流的延迟和抖动状况。该算法仅适用于恒定比特流的网络传输,默认为所有的 IP 传输包缓冲区是相同的。而在实际中,越来越多的网络媒体,尤其是视频传输为了节约带宽提高其利用率,多采用可变比特流模式进行传输。因此 MDI:DF 算法具有一定的局限性。

### 3.3 TS DF

2010 年提出了带有时间戳的延迟因子即 TS\_DF 算法。时间戳延迟因子的计算是基于网络数据包的实际到达时间与 RTP 包头中的时间戳字段间的关系实现的。对于典型的 1Mbit/s 的媒体流,一秒时间可以接收足够的数据包进行抖动的测量,因此通常选用测量周期为 1 秒。在该算法中,测量周期内第一个到达的数据包设定为参考包,假设该数据包没有延迟。对于测量周期内随后到达的数据包  $i$ ,其相对于参考数据包的相对传输时间计算如下:

$$D(i,0) = (R(i) - R(0)) - (S(i) - S(0)) \quad (3-5)$$

式中  $R(i)$ 、 $S(i)$  分别代表封装包  $i$  的 RTP 字段时间戳时间和实际到达时间。

最后取相对传输时间  $D$  的最大值和最小值进行如式(3-6)计算,求得  $TS_{DF}$ 。

$$\text{TS\_DF} = D(\text{Max}) - D(\text{Min}) \quad (3-6)$$

该算法突破了传统方法需要区分 VBR 和 CBR 的局限性,仅需要计量数据包的相对传输时间,适用于所有包含时间戳字段的数据包传输。

### 3.4 算法流程图

经过对 MDI\_DF 和 TS\_DF 两种测量算法的分析,在 Python 平台对两种算法进行实现,算法流程图如图 1、图 2 所示。

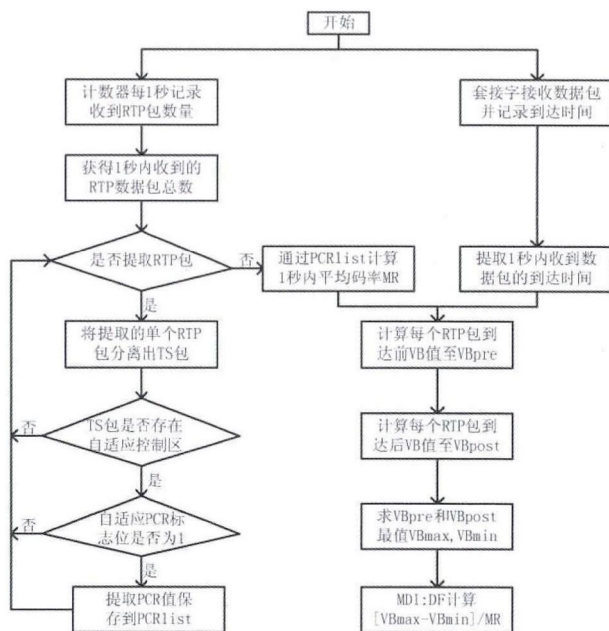


图 1 MDI:DF 算法流程图

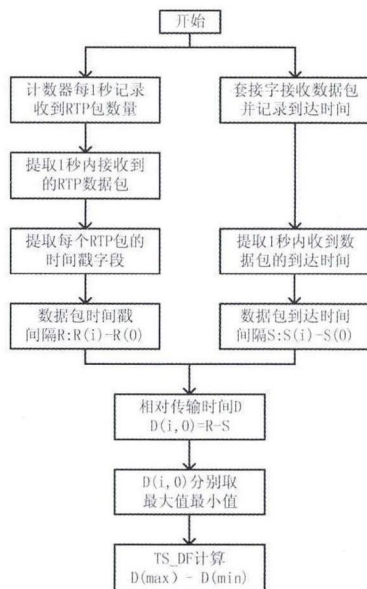


图 2 TS\_DF 算法流程图

## 4 实验环境配置

广域网模拟器<sup>[7]</sup> (WAN\_emulator 简称 WANem) 可以模拟各种网络状况, 如带宽、延迟、丢包等。该模拟器能够完成开发者的单元测试、性能测试与服务器端的链接仿真、多个地点的用户性能测试等功能。为了测试两种算法的准确性, 需要通过 WANem 预先设定网络传输过程的延迟和抖动, 应用不同算法进行测量, 进而分析测试结果的准确度。

同时由于视频传输质量的延迟测量过程对网络环境具有较高要求,网络的拥塞、不稳定等诸多因素均会导致传输数据包的延迟或抖动,因此实验过程需要搭建单独的网络环境。

#### 4.1 网络环境搭建

实验过程中,为了排除其他网络因素的干扰,需要搭建单独局域网,模拟同一网段内的网络状况。将三台主机连接于同一交换机,搭建网络环境如图 3 所示。

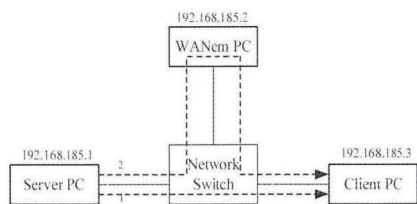


图 3 局域网搭建环境

如图所示, ServePC、WANemPC、ClientPC 通过交换机连接形成一个局域网。其 IP 分别配置为 192.168.185.1、192.168.185.2、192.168.185.3。定义 ServerPC 为数据发送端, ClientPC 为测量端。ServerPC 完成向 ClientPC 的数据传输, WANemPC 在两台主机通信过程中加入延迟和抖动, ClientPC 接收数据并进行视频质量测量。

#### 4.2 WANem 网络仿真

ServePC 完成向 ClientPC 的数据包传送。在正常情况下,两主机间的通信数据流通过图 3 所示虚线 1 传输。为了对发送的数据包加入延迟和抖动, ServePC 和 ClientPC 间的通信需要经过 WANemPC, 即实现数据流经过虚线 2 进行传输。为了实现这一性能,需要对两台通信主机进行路由配置。路由配置方法如下:

发送端: route add 192.168.185.3 mask 255.255.255 192.168.185.2

测量端: route add 192.168.185.3 mask 255.255.255 192.168.185.2

路由配置完成后,在 WANemPC 端高级模式添加实验控制量 Jitter 值的大小。在 WANem 高级模式界面下 Delay time 和 Jitter 栏设置相应数值后单击 Apply setting 即可对两主机间的数据通信加入设定的延迟和抖动。Jitter 值即为算法测量值。

### 5 实验过程与结果分析

VLC 多媒体播放器支持众多音频与视频解码器及文件格式。本文应用 VLC 的串流解码功能,将解码的 TS 流重新封装打包发送到指定网络。在 ServePC 端,通过 VLC 向 ClientPC 端发送 RTP 流。在 VLC 上添加实验视频后模拟出 TS 流的发送过程,将 TS 流以单播的形式发送到 ClientPC。通过 VLC 播放器在串流目标位置中输入目标 IP 后,即可完成 ServerPC 端向 ClientPC 端的串流传输。

应用 Python 语言对两种算法进行实现。为了对算法测量结果进行对比,在 WANemPC 端设置不同的传输环境完成对传输数据包抖动和延迟的添加。实验中,在 WANemPC 端分别设置抖动(Jitter)值为 1 到 20 毫秒。设置 ServerPC 端的流媒体服务器连续进行数据发送。在每种 Jitter 值的网络情况下,数据包在传送过程中通过 WANemPC 加入延迟和抖动处理,在 ClientPC 端分别应用 MDI:DF 和 TS\_DF 两种算法接收数据并进行视频质量测试。每个 Jitter 值传输过程分别取三百组测量数据。

测试完成后,对两种测量算法的测量结果取均值,计算结果数据如图 4 所示。

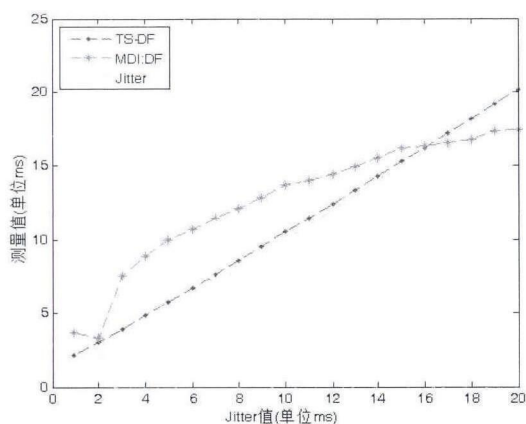


图 4 测量平均值

通过测量值与 WANem 预先设定了抖动值大小相对比,观察到测量值具有上下波动性,则对两组测量结果的累加误差和方差进行计算,结果如图 5、图 6 所示。

理论上,测量值应等于 WANem 设置的 Jitter 值。图 5“--”线为设置 Jitter 值大小。测量结果图 5 表明 MDI:DF 算法测量值具有较大的波动性,测量结果与 WANem 设定的抖动大小相比较具有较

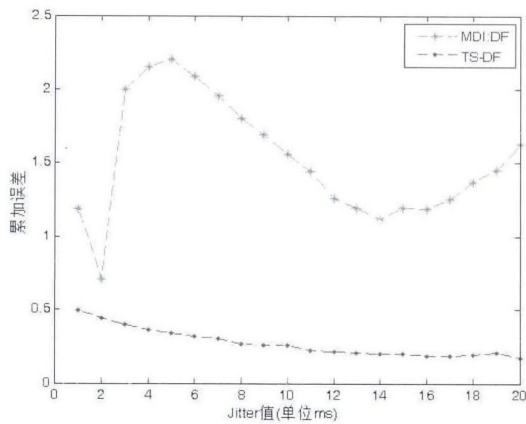


图5 累加测量误差

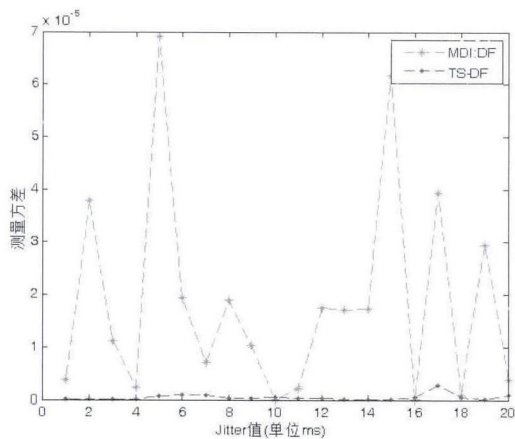


图6 测量值方差

大误差。TS\_DF 算法的测量结果与 WANem 设置的抖动大小基本一致。两种算法测量数据均存在一定的波动性,图5所示 TS\_DF 算法在多次测量后的累加误差明显小于 MDI\_DF 算法,同时随着设置抖动值的增大,TS\_DF 算法累加误差逐渐减小并趋于平稳。图6所示的测量值方差表明对于不同的 Jitter 值网络情况的多次测量中,MDI\_DF 算法的测量结果波动性较大,因此 TS\_DF 算法的测量结果具有更好的稳定性。

## 6 结论

本文分析了 MDI\_DF 测量方法和 TS\_DF 测量方法两种延迟测量算法并通过 Python 语言进行了实现。应用广域网模拟器模拟不同抖动情况下的网络环境,对两台主机间的网络通信人为加入已知大小的延迟和抖动。对不同网络环境下的抖动值大小应用两种测量方法进行测试。测试结果分析显示 MDI\_DF 算法具有较大的波动性,TS\_DF 算法具有较高的精确度和较稳定的测量结果。

## 参考文献

- [1] RFC 3550 - RTP: A Transport Protocol for Real - Time Applications[S].
- [2] RFC 4445 - A Proposed Media Delivery Index (MDI).
- [3] Geneva. A Proposed Time - Stamped Delay Factor (TS - DF) algorithm for measuring Network Jitter on RTP Streams[S]. Jan, 2010.
- [4] 王飞超. 网络抖动的测量技术的分析与实现[J]. 2012.
- [5] 黎致斌. 视频质量测试指标选择 IneoQuest 公司[OL].
- [6] Colin Perkins. RTP - Audio and Video for the Internet[S]. ISBN 0 - 672 - 32249 - 8.
- [7] WANem 2.0 Wide Area Network Emulator Performance Engineering Research Centre[R]. 11<sup>th</sup> Nov, 2008.
- [8] Guo CX, Zheng SR. Analysis and evaluation of the TCP/IP protocol stack of LINUX[J]. In Proc of the IEEE ICCT 2000.

(责任编辑:马玉凤)