

# 一种嵌入视频流的丢包自测量方法\*

胡治国<sup>1,2</sup>, 张大陆<sup>1</sup>, 谷丽丽<sup>1</sup>, 张起强<sup>1</sup>, 陈志伟<sup>1</sup>, 周华磊<sup>1</sup>, 曹孝晶<sup>1</sup>

<sup>1</sup>(同济大学 计算机科学与技术系, 上海 201804)

<sup>2</sup>(中国人民解放军 65583 部队, 辽宁 辽阳 11100)

通讯作者: 张大陆, E-mail: daluz@acm.org

**摘要:** 现有的丢包主动测量方法是通过探测流的丢包信息去推测网络的丢包特性,进而推测特定应用流的丢包,测量结果不能准确获知某一给定应用流的丢包情况.由于丢包通常属于短时间、小概率事件,要更加准确地测量丢包就意味着需延长测量时间,或者提高探测流的发送速率以及时发现丢包,这将不可避免地增加网络的额外负载.分析了不同类型帧损伤的影响,并以 MPEG-4, H264 视频为研究对象,通过对其码流结构特点及 RTP 封装策略的分析,提出一种将测量信息嵌入到视频用户数据域(User\_Data)的丢包测量方法 PLBU(packet loss measurement based on User\_Data).该方法利用视频码流信息完成对丢包的探测,不影响视频的正常播放,不需要注入新的探测流,极大地降低了因测量而引入的额外负载.NIST Net 及 Planetlab 等实验结果表明,该算法不仅丢包测量准确性高,且可提供丢包所属视频帧类型等信息,如视频中 I, P, B 帧的数据包丢失的情况.借助该测量方法,服务提供商可评测网络视频流丢包,并分析视频体验质量(QoE)变化情况,且不受视频流在网络传输中的优先级影响.

**关键词:** 主动测量; 视频; 用户数据域; 丢包; 视频帧

**中图法分类号:** TP393      **文献标识码:** A

中文引用格式: 胡治国, 张大陆, 谷丽丽, 张起强, 陈志伟, 周华磊, 曹孝晶. 一种嵌入视频流的丢包自测量方法. 软件学报, 2013, 24(9): 2182-2195. <http://www.jos.org.cn/1000-9825/4338.htm>

英文引用格式: Hu ZG, Zhang DL, Gu LL, Zhang QQ, Chen ZW, Zhou HL, Cao XJ. Embedding method for packet loss self-measurement of video streaming. Ruan Jian Xue Bao/Journal of Software, 2013, 24(9): 2182-2195 (in Chinese). <http://www.jos.org.cn/1000-9825/4338.htm>

## Embedding Method for Packet Loss Self-Measurement of Video Streaming

HU Zhi-Guo<sup>1,2</sup>, ZHANG Da-Lu<sup>1</sup>, GU Li-Li<sup>1</sup>, ZHANG Qi-Qiang<sup>1</sup>, CHEN Zhi-Wei<sup>1</sup>, ZHOU Hua-Lei<sup>1</sup>, CAO Xiao-Jing<sup>1</sup>

<sup>1</sup>(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

<sup>2</sup>(The Chinese People's Liberation Army, 65583 Troops, Liaoyang 111000, China)

Corresponding author: ZHANG Da-Lu, E-mail: daluz@acm.org

**Abstract:** The accurate packet loss measurement of application traffic has been challenging, since the frequently-used active monitoring approaches that use probe packets cannot capture the packet loss experienced by the traffic of individual user applications. Moreover, measurement and estimation of packet loss characteristics are challenging due to the relatively rare occurrence and typically short duration of packet loss episodes. Because it is often impossible to measure the entire data passing across a network, the most widely used method of measurement works by injecting probe packets. In order to overcome these shortcomings, this paper based on the analysis of MPEG4 video structural features and RTP encapsulation strategy, a novel packet loss probing technique, packet loss measurement based on User\_Data (PLBU), is proposed. The PLBU complete packets loss measurement by using User\_Data field of MPEG-4 video. This does not affect the normal video playback and dose not need to inject new probing stream, which makes it, less intrusive. NIST Net and

\* 基金项目: 国家自然科学基金(61073154)

收稿时间: 2011-09-19; 定稿时间: 2012-10-24

Planetlab experiments verify the effectiveness of this method, and show that PLBU estimates packet loss more accurately, and with less measurement overhead than other existing technique such as BADABING.

**Key words:** active measurement; video; User\_Data; packet loss; video frame

丢包测量在网络拥塞控制、服务质量保障及视音频质量评价等领域有着广泛的应用,包丢失率常被作为包丢失的度量指标.由于被动测量通常无法完成对端到端的评测,现有丢包测量方法多为主动测量,其方法是通过探测流的丢包信息去推测网络的丢包特性,进而推测特定应用流的丢包.通常在丢包测量过程中,为了更好地分析数据包丢失情况且不失一般性,认为探测流(probe traffic)的丢包率就是背景流(cross traffic)的丢包率,且假设注入的测量数据包不引起背景流额外的丢包<sup>[1]</sup>.

IETF 组织的 IP 网络性能指标工作组(IP performance metrics working group)提出了针对丢包测量的建议标准,其主要关注的是网络丢包测量<sup>[2]</sup>.但用户和服务提供商所关心的是其能够得到或提供怎样的业务服务质量,显然,用户与服务提供商的需求与应用流的 QoS 相关,并不会关心网络的丢包情况.尽管网络丢包与应用流的丢包有着直接的关联,但并不能准确代表应用流的丢包.另外,由于丢包通常是小概率、短时间的事件,丢包的测量与探测数据包长度以及包发送频率相关,因此,准确测量丢包非常困难<sup>[1]</sup>.此外,对流媒体服务而言,其传输质量的好坏不仅与丢包数量,还与丢包位置、内容紧密相关,如 MPEG 视频中,I,P,B 帧的数据包丢失对视频质量损伤程度明显不同,而目前传统的丢包测量方法并不能提供除丢包数量之外的信息.

本文通过对 MPEG4,H264 视频编码格式及 RTP 封装策略的研究,提出了一种新的丢包测量算法 PLBU (packet loss measurement based on User\_Data),该算法在视频文件 User\_Data 域中嵌入特定的测量信息.与传统的被动、主动测量方法不同,PLBU 算法利用视频码流完成对丢包的探测,不生成任何新的探测流,降低了因测量而引入的额外负载,且能直接提供视频数据丢失的详细内容.借助于此算法,服务提供商可精确获得视频流的丢包情况,监测自身的系统状况,判断当前运营服务水平和用户体验质量(QoE),并为 SLA 服务参数、阈值的界定、QoS 保证机制等提供有效依据.

## 1 相关研究

当网络发生拥塞或 IP 数据包传输过程出现比特差错,都可能造成 IP 数据包丢失.PING<sup>[3]</sup>,ZING<sup>[4]</sup>,BADABING<sup>[1]</sup>是目前丢包测量中的代表算法.PING 程序发送多个 ICMP 数据包,其根据 ICMP 数据包应答信息来判断是否发生丢包,显然,其应答包的丢弃直接影响测量精度.此外,如果传输过程中防火墙或路由器启用了 ICMP 过滤功能,则无法使用 PING 方法测量丢包.ZING 算法<sup>[4]</sup>是基于泊松模型的丢包测量方法,其采用了依概率发包的方式,对网络的入侵度较小.但是 Sommers 等人指出,由于丢包是属于小概率事件,要准确地测量丢包就意味着增长测量时间,或者提高探测流的发送速率来及时发现丢包,这将不可避免地增加网络的额外负载<sup>[1]</sup>.文献[1]运用排队理论设计了独特的网络探测包,提出 BADABING 丢包测量方法,其主要思想是发送一系列的探测包列,每个包列以一定概率决定是否发送,每个包列中包含一对或多对背对背的探测包.该种探测包列较好地平衡了测量准确性和测量入侵性这对矛盾.文献[1]中的实验表明,在相同发送速率下,BADABING 比标准泊松测量包有着更高的准确性,且可推测丢包持续时间.基于端到端路径数据包丢失率测量结果及已知的网络拓扑结构,文献[5-7]分别利用最大似然估计算子、因子图和精确估计算子来推断网络内部各链路上数据包的丢失情况.通常在丢包测量过程中,为了更好地分析数据包丢失情况且不失一般性,认为探测流(probe traffic)的丢包率就是背景流(cross traffic)的丢包率,并反映当前应用流的丢包情况<sup>[1]</sup>,且假设注入的测量数据包不引起背景流额外的丢包.即使 BADABING 通过设置包列中探测包数量的方法可以较好地测量一定持续时间的丢包情况,也不可能准确捕获测量过程中所有的丢包情况,因此,其测量结果并不一定与背景流的丢包情况相一致.此外,这种不断注入探测流的测量方法不可避免地对网络造成较大的额外负载.

除了以上测量算法以外,基于 RTP/RTCP 协议传播系统中,可通过分析 SR(sender report)类型 RTCP 包中丢包字段(fraction lost (8bit)),可以获取当前 RTP 会话的丢包数目.该方法虽然简单,但 RTCP 包的丢失将直接导致

测量失败,而且无法获取丢包的类型信息.

文献[8]提出可以利用 MPEG 文件中的 User Data 域携带额外信息,从而使得视频流的传输更加智能和高效;文献[9]通过在 User Data 域插入帧的整体运动参数,重构受损的视频对象平面(VOP)边界;文献[10]提出将帧同步信息及特殊水印等嵌入到 User Data 域中去克服视频传输损伤.但目前还没有利用 User Data 域去测量视频传输中丢包情况的研究.

## 2 视频编码格式及 RTP 分包策略

IETF 针对不同的视频类型分别给出实时传输协议(RTP)分包的封装策略,如文献[11]针对 MPEG-1,2,文献[12]针对 MPEG4,文献[13]针对 H264.本文主要以目前常用的 MPEG-4 视频编码及文献[12]定义的 RTP 分包规则为例进行详细分析.

### 2.1 MPEG视频编码格式分析

MPEG 视频由若干个 GOP 组成,每个 GOP 又由 1 个 I 帧和数量不等的 P 帧、B 帧组成.图 1 给出的 GOP 中 I,P,B 帧的典型排列方式.在 MPEG-4 中,提出视频对象平面 VOP(video object plant)的概念.在编码器中,通常将整个图像作为一个对象处理,VOP 与 MPEG-4 中的帧相对应,即 I VOP 对应 I 帧,P VOP 对应 P 帧<sup>[14]</sup>.I 帧,又称 Intra-Coded 帧,解码时仅用 I 帧的数据就可重构完整图像;P 帧,又称为 predictive 帧,P 帧根据前一个 I 帧,或者 P 帧的变化编码;B 帧,又称为 bidirectional predictive-coded 帧,B 帧根据前后的 I,P 帧完成编解码.

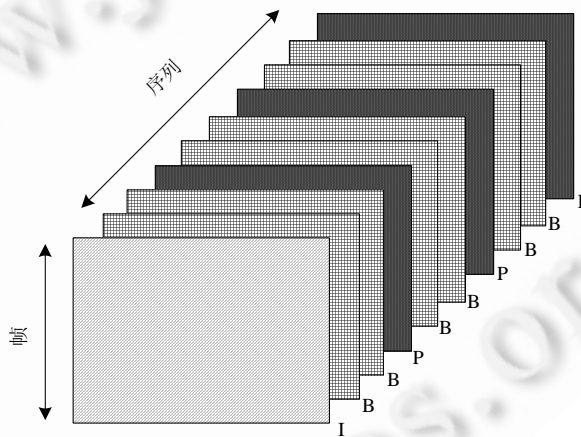


Fig.1 A sample of MPEG4 GOP<sup>[15]</sup>

图 1 MPEG-4 视频帧序列片段<sup>[15]</sup>

MPEG4 视频编码,其码流起始标识(保留字段)见表 1.

Table 1 Start code values in MPEG-4<sup>[16]</sup>

表 1 MPEG-4 码流起始标识<sup>[16]</sup>

序列起始码	0x000001B0
对象起始码	0x000001B5
对象层起始码	0x00000120
图片组起始码	0x000001B3
图片起始码	0x000001B6
用户数据起始码	0x000001B2

紧跟在图片起始(0x000001B6)标记后的 2 位二进制数是帧类型标识信息,00 为 I 帧,01 为 P 帧,10 为 B 帧<sup>[16]</sup>. MPEG-4 视频中留有用户数据域(User\_Data),起始码为 0x000001B2,起始码之后数据部分可由用户自己定

义并用于特定的应用.但是在用户数据域内不能出现与已定义的保留字段相同二进制信息,以避免解码器在检索起始码时出现差错.User\_Data (用户数据域)在视频序列中具体位置如图 2 所示.



Fig.2 User\_Data field position in Mpeg-4 video

图 2 User Data 域在视频中的位置

**2.2 RTP 分包策略**

RFC3016 描述了 MPEG-4 视频流与 RTP 包的映射关系,其分包规则以 VOP 作为基本封装单位<sup>[12]</sup>.主要特征如下:

- (1) MPEG-4 编码信息头部仅出现在第 1 个 RTP 载荷的开始部分;
- (2) 一个 VOP 包单独放入单个 RTP 包中,但是一个 VOP 在一个 RTP 包中放不下的情况下,此时将 VOP 按 RTP 包限长分段,分别放入多个 RTP 包;
- (3) 如果包含视频的 IP 包太大,超过网络最大传输单元(maximum transmission unit),将其按 MTU 为单位分割;
- (4) RTP 时间戳按常数递增,属于同一个视频帧的 RTP 包,将有相同的时间戳.

**2.3 不同类型帧损伤影响分析**

在 MPEG 视频中,I 帧描述了图像背景和运动主体的详情,I 帧是 P 帧和 B 帧的参考帧.P 帧采用运动补偿的方法传送它与前面的 I 帧或 P 帧的差值及运动矢量(预测误差);P 帧编解码参考前面最靠近它的 I 帧或 P 帧;B 帧传送的是它与前面的 I 帧或 P 帧和后面的 P 帧之间的预测误差及运动矢量;B 帧编解码以前面的 I 帧或 P 帧和后面的 P 帧为参考帧.由此可知:I 帧,P 帧,B 帧对视频质量的影响相差很大.其中,I 帧最为重要,若 I 帧丢失,则其后到下一个 I 帧之前的所有 P 帧 B 帧的质量都会受到影响;若 P 帧丢失,则其后到下一个 P 帧之前的所有 B 帧解码都会受到影响;而 B 帧丢失的错误不发生传播.

不失一般性,假设 1 个 GOP 中包含 I,P,B 这 3 种类型的帧,有  $m$  个 P 帧, $n$  个 B 帧,B 帧均衡分布在 I 帧或 P 帧之间,由于 1 个 GOP 仅有一个 I 帧,则 GOP 共包含  $m+n+1$  个帧;

1) GOP 中不同帧受损影响分析

设:

- 由于 I 帧类型的数据包丢失造成的 I 帧自身损伤记为  $D_I$ ;
- 由于 P 帧类型的数据包丢失造成的 P 帧自身损伤记为  $D_P$ ;
- 由于 B 帧类型的数据包丢失造成的 B 帧自身损伤记为  $D_B$ ;
- GOP 中 I 帧损伤而使相邻 P 帧损伤记为  $D_{IP}$ ;
- 由于前一 P 帧受损而使相邻 P 帧损伤记为  $D_{PP}$ ;
- 由于 I 帧损伤而使 I 帧与第 1 个 P 帧之间的 B 帧损伤为  $D_{IBP_1}$ ;
- 由于 P 帧损伤使 GOP 中此 P 帧之前的 B 帧解码损伤为  $D_{PB}$ ;
- 由于 P 帧损伤而使 B 帧损伤为  $D_{PBP}$ .

由 I,P,B 在编解码所起作用可知:

情况 1. 当数据包丢失而使 I 帧受损,GOP 中受到影响损伤总和为

$$D_I + D_{IP_0} + \sum_{i=1}^{m-2} D_{P_i P_{i+1}} + \sum_{i=0}^{n/m} D_{IB_i P_1} + \sum_{i=1}^{m-2} \sum_{j=0}^{n/m-1} D_{P_i B_j P_{i+1}} \tag{1}$$

情况 2. 当数据包丢失而使 P 帧受损,GOP 中受到影响损伤总和为:

记该 P 帧标号为  $k$ ,则公式如下:

$$D_{P_k} + \sum_{i=k}^{m-2} D_{P_k P_{k+1}} + \sum_{i=k}^{m-2} \sum_{j=0}^{n/m-1} D_{P_k B_j P_{k+1}} + \sum_{i=k}^{m-2} \sum_{j=0}^{n/m-1} D_{P_k B_j} \quad (2)$$

情况 3. 当数据包丢失而使某一 B 帧受损,由于 B 帧损伤并不影响除自身以外的其他帧,故 GOP 中受到影响损伤总和为

$$D_B \quad (3)$$

2) 丢包对视频帧的损伤影响分析

设  $N_I, N_P, N_B$  为 I,P,B 单个帧分包数量, $p$  为包丢失率, $p_I, p_P, p_B$  为 GOP 中 I,P,B 帧损伤概率.由于不同编码的包错误隐藏策略存在差异,所以丢包对视频帧损伤的影响也不相同,但本文此处仅对丢包对视频帧损伤影响的作一般性的分析讨论,设视频帧发生一个丢包,即认为该帧受损.

丢包对帧损伤的概率为

$$p_I = 1 - (1 - p)^{N_I} \quad (4)$$

$$p_P = \frac{1}{m} \times \sum_{k=1}^m (1 - (1 - p)^{N_I + k \times N_P}) \quad (5)$$

$$p_B = \frac{1}{m+n} \times \sum_{k=1}^m \sum_{l=1}^n (1 - (1 - p)^{N_I + k \times N_P + l \times N_B}) \quad (6)$$

GOP 中帧受损数量的期望值  $E(\rho)$ 为

$$E(\rho) = p_I \times (n+m+1) + p_P \times \theta + p_B \times 1 \quad (7)$$

$\theta$ 为 P 帧受损后使 GOP 中帧受损数量的期望.

从 GOP 中 P 帧受损(从第 1 个 P 帧开始)而造成 GOP 中帧受损数量分别是

$$m \times \frac{n}{m} + m, (m-1) \times \frac{n}{m} + m - 1, \dots, 2 \times \frac{n}{m} + 2, \frac{n}{m} + 1.$$

由此可知  $\theta = \frac{m^2 + m + 2n}{2m}$ ,则公式(7)可写为

$$E(\rho) = p_I \times (n+m+1) + p_P \times \frac{m^2 + m + 2n}{2m} + p_B \times 1 \quad (8)$$

3) 丢包对视频质量影响分析

设  $p_I, p_P, p_B$  为 GOP 中 I,P,B 帧所对应的丢包率,则丢包对视频质量的损伤还可表示为:

不同帧丢包率对视频帧损伤的概率为

$$I'_I = 1 - (1 - p_I)^{N_I} \quad (9)$$

$$I'_P = \frac{1}{m} \times \sum_{k=1}^m (1 - (1 - p_P)^{k \times N_P}) \quad (10)$$

$$I'_B = \frac{1}{n} \times \sum_{l=1}^n (1 - (1 - p_B)^{l \times N_B}) \quad (11)$$

文献[17]认为,由于解码器一般会对帧损伤进行空间过滤,则第  $k$  个帧损伤后对第  $(k+i)$ 个帧的造成传播损伤可近似用如下公式表示为<sup>[17]</sup>

$$\sigma^2[k+i] = \sigma^2[k] \gamma^i \quad (12)$$

$\sigma^2[k]$ 表示第  $k$  个帧损伤, $\gamma$ 为递减因子.即第  $k$  个帧损伤为  $\sigma^2[k]$ ,则对其他紧跟它的帧造成的损伤为  $\sigma^2[k] \gamma$ ,  $\sigma^2[k] \gamma^2$ ,  $\sigma^2[k] \gamma^3, \dots$ ,且假设损伤具有可加性<sup>[17]</sup>.

因此,设 I,P,B 帧受损对视频质量的影响为  $\sigma_I^2, \sigma_P^2, \sigma_B^2$ ,其递减因子分别为  $\gamma_I, \gamma_P, \gamma_B$ .

则不同 I,P,B 帧丢包率情况下对视频质量的损伤为

$$I'_i \times \sum_{k=1}^{n+m+1} \sigma_i^2 \gamma_i^{k-1} + I'_p \times \sum_{k=1}^{(m^2+m+2n)/2m} \sigma_p^2 \gamma_p^{k-1} + I'_B \times \sigma_B^2 \tag{13}$$

上式易于推广到含多个 GOP 的视频文件.

从公式(1)~公式(13)的分析可知,I,P,B 帧损伤波及影响不同,且不同帧类型丢包对视频质量的影响也不同.从公式(13)可以看出,准确测量视频流的丢包和获得丢包的内容信息是准确评测某一丢包情况下视频质量变化的重要基础.

在 Evalvid 仿真视频传输平台上<sup>[18]</sup>,本文初步研究了 I,P,B 帧数据包丢失对视频质量造成的影响.视频质量评价采用平均意见分 MOS<sup>[19]</sup>,MOS 评分法是一种广为使用的主观评价方法,它以平均意见分来衡量视频质量,用 5 个等级来表示视频质量:优(5 分)、良(4 分)、一般(3 分)、差(2 分)、坏(1 分).图 3 的实验结果也证明 I 帧数据包丢失对视频质量的影响最为重要.

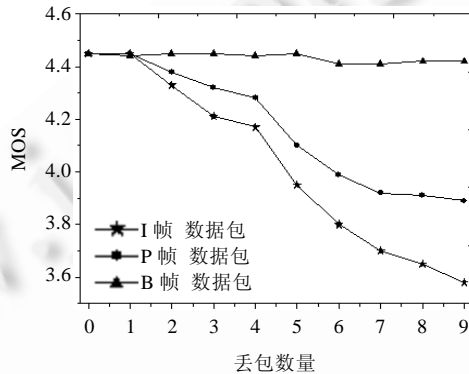


Fig.3 Different types of packet loss on video quality  
图 3 不同类型丢包对视频质量影响

### 3 算法设计

本文提出一种新的丢包测量算法:基于用户数据域的丢包测量算法 PLBU(packet loss measurement based on User\_Data),该算法在视频文件 User\_Data 域中嵌入特定的测量信息,在不影响视频播放的同时,通过在接收端检测和分析视频 User\_Data 域内的信息,完成对视频传输过程视频流的丢包测量,整个算法过程可分为 User\_Data 域测量信息插入与提取两部分.

#### 3.1 视频User\_Data域中测量信息的插入

计算一个 MPEG4 视频文件在 IP 网络传输中的分包数量,首先需要知道网络传输时的分包规律,即视频服务器如何将较大的 MPEG-4 视频文件数据封装成 IP 数据包发送到客户端.IP 数据包能够装载的最大数据量 MSS(max segment size,最大传输长度)与 MTU(最大传输单元)的关系是

$$MSS=MTU-IP 头-TCP/UDP/RTP 头.$$

由第 2.2 节的分析可知:若一个 MPEG-4 视频帧长度大于 MSS,则视频帧被装入多个 IP 数据包中进行传输.每个帧单独分包传输,即 RTP 包中仅包含单一帧类型的数据包.由此,根据帧起始码之间的字节数可确定每帧长度,进而根据路径 MTU 值确定每帧分包个数.另外,由于视频序列头和图片组头等信息较小,其与 I 帧数据共同封装成第 1 个 RTP 包.

文献[20]描述了路径 MTU 的发现机制,主要思想是通过设置数据包不分片(DF)的标志比特、严格路由选项及 ICMP 不可达等信息,找到指定路径中 MTU 值.依据测得的路径 MTU 值,本文使用 MP4box<sup>[21]</sup>设置 MPEG4 视频 RTP 分包长度.借助帧的起始标志位,可计算出视频中不同帧的大小,并由 RTP 分包大小,计算出帧的具体分包数量.

服务器端在视频发送之前,将视频文件帧数量、分包数量等写入每个帧末尾的 User\_Data 域.接收端程序在测量过程中提取 User\_Data 内信息,结合实际的收包情况判断视频流丢包情况.

具体内容包括:

- User\_Data(用户数据)域的起始标志 0x000001B2;
- 0x00ff(自定义)为标识 PLBU 算法测量标记;
- I 帧、P 帧、B 帧标志信息,本文定义:I 帧为 01,P 帧为 02,B 帧为 03;
- I,P,B 帧分包数量.

举例:设某一 I 帧分包为 5,则此 I 帧的 User\_Data 测量包设计如图 4 所示.

0x000001B2	0x00ff	0x01	0x05
------------	--------	------	------

Fig.4 Structure of the measurement information in User\_Data field

图 4 User\_Data 内测量信息设计

User\_Data 测量信息丢失将会引起测量误差,因此,本文通过加入备份冗余数据的方式来确保至少有 1 处以上的 User\_Data 测量信息被接收到.为此,假设 User\_Data 测量信息至少有 1 处被接收的概率为  $P$ ,包含测量信息的数据包在网络中丢失的概率为  $PLR$ ,针对每个帧的 User\_Data 测量信息的总数为  $N$  个,则测量信息全部丢失的概率等于  $PLR^N$ ,有一处以上测量信息被正确的接收的概率等于  $1-PLR^N$ ,因此若要保证 User\_Data 测量信息有一处以上被正确的接收,需满足  $1-PLR^N > P$ ,即  $N > \log_{PLR}(1-P)$ .

假设至少有一处 User\_Data 测量信息被接收的概率需  $P > 95\%$ ,且每个测量信息丢失概率小于 50%;当  $N \geq 5$  时,即可满足条件.因此需要有 4 个备份冗余信息来确保实现这一目标.冗余信息可插入在其他帧之后的 User\_Data 中.

含冗余信息的 User\_Data 设计如下:

- 0xff(自定义)为标识 PLBU 算法冗余数据标记;
- I 帧、P 帧、B 帧冗余标志信息,本文定义:I 帧为 0x11,P 帧为 0x12,B 帧为 0x13;
- 针对某个 I 帧、P 帧、B 帧的序号标志信息(用 3 个字节表示,可提供 16 777 216 个数字,保证有足够的位数来标记帧个数);
- 被冗余帧的分包数量(1 个字节).

假设第 10 个帧(序号标志信息:0x00000a),分包为 3 个(0x03),则针对该 I 帧的冗余 User\_Data 设计如图 5 所示.

0x000001B2	0x00ff	0x01	0x05	0xff	0x11	0x00000a	0x03
------------	--------	------	------	------	------	----------	------

Fig.5 Structure of redundant measurement information in User\_Data field

图 5 User\_Data 内冗余信息设计

### 3.2 视频 User\_Data 域中测量信息的提取

在视频传输过程中,接收端为了统计指定视频流的数据包丢失情况,需要将该视频数据包与链路上其他数据包区分开.具体算法如下:

- (1) 根据视频服务器地址、视频序列头标记等信息抓取第 1 个视频数据包,提取该数据包源 IP 地址、目的 IP 地址、源端口、目的端口、协议类型的五元组信息;
- (2) 获得上述五元组符合的视频包个数,检查收到的视频包是否包含本文定义的 User\_Data 标志信息,如有则提取 User\_Data 域的内容信息;

- (3) 根据 User\_Data 数据域中已写入的标记信息,获得每个帧的分包数,并判断是何种帧类型;
- (4) 结合标记信息及时间戳信息,计算每个 I,B,P 帧所对应的实际收到包的个数(同一个帧分出的包 RTP 时间戳相同);
- (5) 计算出 I 帧、B 帧、P 帧丢包数;
- (6) 计算出总的丢包数.

接收端通过统计收到视频数据包数量,并从 User\_Data 域内提取出视频总的发包数量及帧类型信息就可以准确的算出本次视频播放中数据包中 I,P,B 帧的丢包率及总的丢包率.

PLBU 接收端算法如图 6 所示.

---

**PLBU algorithm at RCV.**

---

**Notations**

Identify\_video\_stream: Identify the video stream or not  
 packets[]: The received packets number of frames  
 userdata[][]: Frame kind and packets number the frame packeted  
 Ilossrate: The packets loss rate of I frame  
 lossrate: The packet lossrate of the videoBegin  
 polling interval: Sampling on a specified time

**Begin**

1. identify\_video\_stream=0;
2. **while** identify\_video\_stream is 0
3.     **if** original\_ip is video\_server\_ip && sequence\_start\_code  
        &&transport\_layer\_protocol is RTP
4.         catch\_first\_video\_packet();
5.         identify\_video\_stream=1;
6.         get\_five\_tuples\_info();
7.     **end if**
8. **end while**
9. **while** video is not over || polling interval
10.     catch\_packet();
11.     get\_five\_tuples\_info();
12.     **if** satisfy\_five\_tuples\_info
13.         **if** contain\_userdata
15.             **if** userdata[i][] is I
16.                 isendnum+=packet[i];
17.                 ircvnum+=packets[i];
18.             **else if** userdata[i][0] is P
19.                 psendnum+=packet[i];
20.                 prcvnum+=packets[i];
21.             **else if** userdata[i][] is B
22.                 bsendnum+=packet[i];
23.                 brcvnum+=packets[i];
24.             **end if**
25.         **end if**
26.     allsendnum=isendnum+psendnum+bsendnum;
27.     allrcvnum=ircvnum+prcvnum+brcvnum;
28.     ilossrate=(isendnum-ircvnum)/allsendnum;
29.     plossrate=(isendnum-ircvnum)/allsendnum;
30.     blossrate=(isendnum-ircvnum)/allsendnum;
31.     return measurement results;
32.     **end if**
33. **end while**

**End**

---

Fig.6 PLBU receiver-end algorithm

图 6 PLBU 接收端算法

如果数据包在传输过程中,由于传输路径发生改变而使路径中的 MTU 值随之也产生变化,从而导致视频包发生分片,理论上在接收端通过分析 IP 数据包标识(identification,当数据包长度超过网络 MTU 而必须分片时,



标识字段的值复制到所有的数据包片的标识字段中)、分片标志 flag 字段 MF(分片时除最后一块外 MF 均置为 1,最后一块为 0)及片偏移字段(指出某片在原分组中的相对位置)即可推测出接收到的完整数据包数量.另外,本文通过实验分析表明:使用 UDP 传输协议,如数据包中某个分片丢失则整个数据包丢失,使用 Wireshark 包分析软件及 PLBU 算法均不能获得有分片发生丢失的数据包,即数据包中某一分片丢失,则该数据完全丢失.因此,不会因传输过程中视频数据包产生分片而影响 PLBU 测量精度.

#### 4 算法性能对比分析与验证

文献[1]对 BADABING 算法与 ZING 算法进行了对比实验,指出 BADABING 在同样测量负载的情况下,比 ZING 算法的准确度更高.因此,本文选择 BADABING 算法作为对比对象.

为了验证本方法准确性,在服务器端和客户端都安装网络嗅探工具 Wireshark 来分析统计视频传输过程的数据包丢包情况.本实验中视频服务器选用 Helix Server<sup>[22]</sup>.为避免实验中 BADABING 算法与本文提出的算法因网络环境产生差异而造成统计偏差,所有实验中 BADABING 与本文提出的算法均同时运行及结束.

##### 4.1 仿真实验

###### 4.1.1 Nistnet 仿真平台

采用网络模拟软件 Nistnet<sup>[23]</sup>模拟网络路径丢包情况.客户端点播服务器端视频并计算视频在网络传输中发生的丢包.具体的实验拓扑图如图 7 所示.

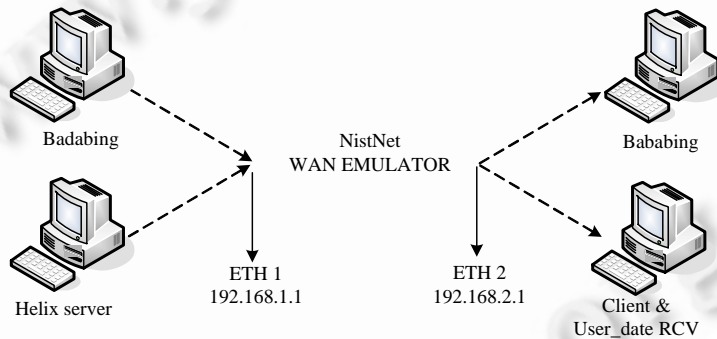


Fig.7 Simulation experiment topology

图 7 仿真实验拓扑图

进行两组实验对比,标记为实验(1)和实验(2);Nistnet 中分别设置丢包率为 15%和 5%.

实验(1)中测得的 I,P,B 帧丢包率及总的丢包率分别为 0.8038%,7.8778%,4.1800%,12.8617%,此时,BADABING 算法发送 2 568 个探测包,收到探测包数量为 2 213,丢包率为 13.823%;

实验(2)中,I,P,B 帧丢包率及总的丢包率分别为 0.3215%,3.8585%,0.9646%,5.1446%,此时,BADABING 发送探测包数量为 2 292 个,收到探测包数量为 2 190 个,丢包率为 4.45%.

此实验中,PLBU 测量结果与 Wireshark 抓取的视频包丢失情况相一致.实验结果如图 8(a)和图 8(b)所示.

实验结果表明,视频流的实际丢包率与 Nist Net 设定丢包率之间出现偏差,其主要原因是由于网络丢包与应用流丢包发生情况并不总是一致.

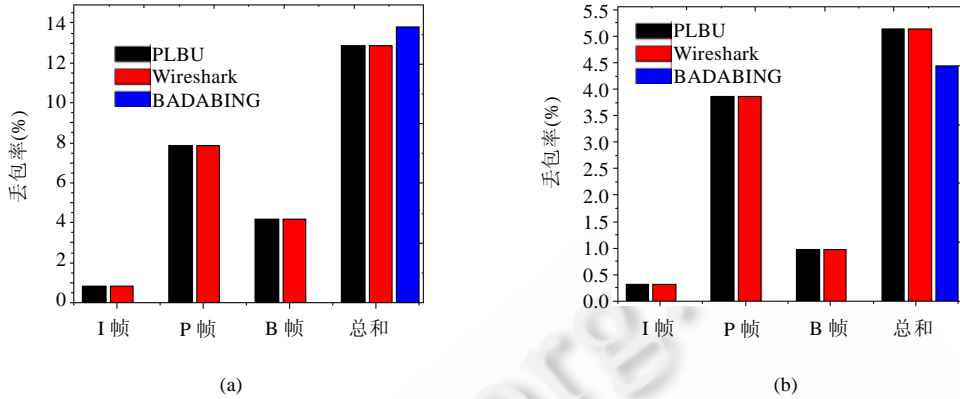


Fig.8 Estimation obtained by PLBU and BADABING, compared with real value

图 8 PLBU 与 BADABING 测量结果比较结果

4.1.2 实验网仿真平台

为验证本文提出的视频丢包测量算法,本文搭建一个实验网络,中间节点均为锐捷路由器,拓扑结构如图 9 所示.

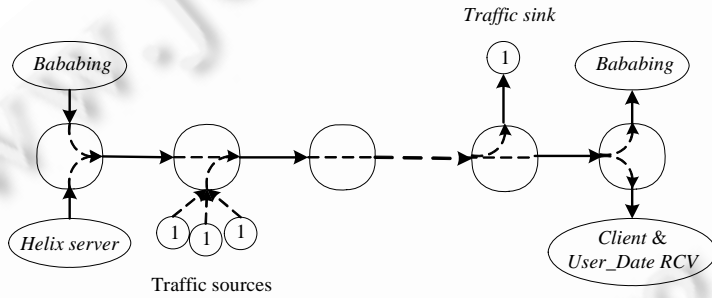


Fig.9 Experiment network topology

图 9 实验网拓扑图

实验中所用交叉背景流为来自 WIDE 主干网的一条连接日本和美国的网络路径上实际采集的流<sup>[24]</sup>,流重放工具采用 Tcpreplay<sup>[25]</sup>.

进行两组实验对比,标记为实验(1)、实验(2).

实验(1)中测得的 I,P,B 帧丢包率及总的丢包率分别 1.9293%,2.7331%,7.7170%,12.3794%;此时, BADABING 发送探测包数量为 2 481 个,接收到的探测包数量为 1 911 个,丢包率为 22.974%;

实验(2)中,I,P,B 帧丢包率及总的丢包率分别为 1.6077%,4.3408%,2.2508%,8.1994%,BADABING 发送探测包数量为 2 316 个,收到探测包数量为 1 968 个,丢包率为 15.361%.

此实验中,PLBU 算法测量结果与 Wireshark 获得的视频丢包情况相一致.实验结果如图 10(a)、图 10(b)所示.实验中发现,视频流丢包与 BADABING 测量到的丢包相差较大.除探测流与应用流丢包存在差异的原因之外,本文认为,由于 BADABING 背对背的发包方式与背景流本身具有的突发性作用相互迭加,加剧了网络拥塞,而使 BADABING 的丢包现象更加严重.

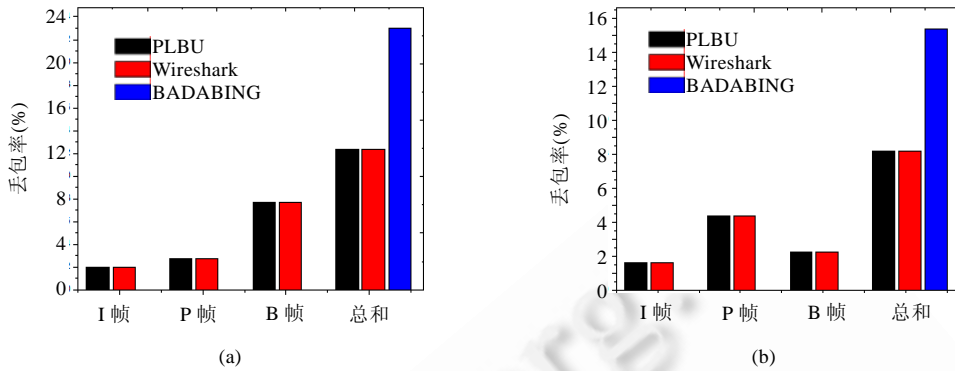


Fig.10 Estimation obtained by PLBU and BADABING, compared with real value  
图 10 PLBU 与 BADABING 测量结果比较结果

4.2 Planetlab平台

本文在互联网技术测试平台 PlanetLab<sup>[26]</sup>上对算法性能进行了验证,节点分别选择清华大学(THU, pl1.6test.edu.cn,219.243.208.60)、加州大学洛杉矶分校(UCLA,planetlab1.cs.ucla.edu,131.179.150.70)。由于 PlanetLab 无法使用图形界面的客户端,因此本文使用 OpenRSTP<sup>[27]</sup>作为客户端,包分析软件使用 TcpDump。因在实际网络上很少出现丢包,且因 Planetlab 对每个 slice 流量和分配容量的限制,无法长时间地运行算法去检验丢包,因此引入 Iperf<sup>[28]</sup>软件来增大测量过程网络背景流量,造成丢包,来进行丢包测量精度分析。

进行两组实验对比,标记为实验(1)、实验(2)。实验(1)中测量得的 P,B 帧丢包率及总的丢包率分别 1.6077%, 0.9646%,2.5723%;BADABING 发包 2 469 个,收到 2 424 个,丢包率为 1.8226%。实验(2)中,P,B 帧丢包率及总的丢包率分别为 2.0900%,1.2862%,3.3762%。BADABING 发送探测包数量为 2 426 个,收到探测包数量为 2 364 个,丢包率为 2.5556%,PLBU 与 TcpDump 抓取视频包丢失情况相一致。在此实验中,没有发现含 I 帧的数据包丢失。实验结果如图 11(a)、图 11(b)所示。

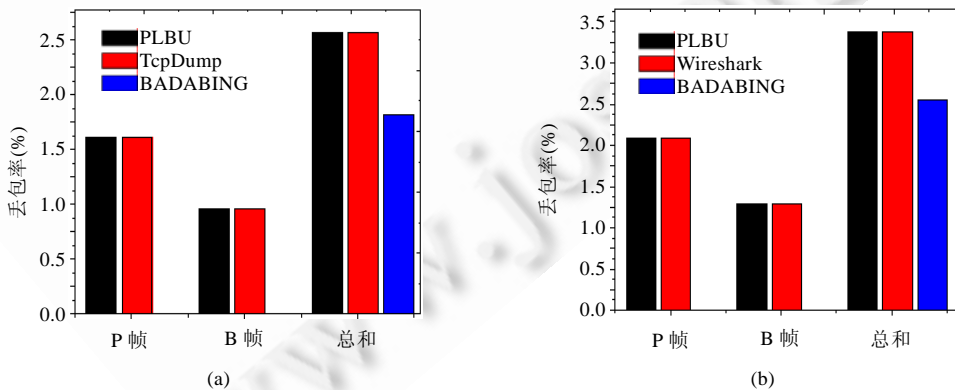


Fig.11 Estimation obtained by PLBU and BADABING, compared with real value  
图 11 PLBU 与 BADABING 测量结果比较结果

从上述实验可以看出,使用 BABABING 算法所测量出来的丢包情况去推测具体应用的丢包情况还存在较大的偏差。在实际网络,通常使用设置优先级队列的方法来保证视频等应用的服务质量。这种情况下,探测包与应用数据包的优先等级将不一致,此时若再使用 BABABING 等算法的测量结果去判断视频的丢包情况,则测量的误差还会更大,而 PLBU 算法则不会出现这种情况。

虽然某个帧的部分数据包发生丢失,该帧能否被解码还与具体的解码器有着很大关系;但如果帧的第 1 个包丢失,该帧几乎不可能被解码<sup>[29]</sup>.依据帧标识(帧标识出现在该帧的第 1 个 RTP 包内)及图 5 中定义的帧序号标志信息,PLBU 算法还可以判断出包含帧头部信息的数据包是否发生丢包及所属帧序号.表 2 为第 4.1.1 节实验(1)中,通过 PLBU 算法测量得到的帧头部发生丢包时,丢包所属帧序号信息.

Table 2 Partial output of PLBU  
表 2 PLBU 算法的部分输出结果

	丢包总数	I 帧/P 帧第 1 个丢失时所属帧的序号
I 帧第 1 个丢失	4	3,7,8,14
P 帧第 1 个丢失	26	14, 16, 33, 42, 43, 44, 64, 66, 91, 92, 94, 95, 98, 101, 106, 109, 126, 131, 144, 152, 157, 167, 185, 187, 193, 196

上述结果与 Wireshark 得到结果相一致.另外,本文实验所用视频中的 B 帧较小,传输过程中仅需单个 RTP 包封装.

### 4.3 入侵度分析

本文定义测量负载为获得一次完整测量结果所需写入视频文件(注入网络)的字节数.实验中均采用 foreman.qcif 视频文件,经编码后该视频长约 13s,PLBU 算法写入视频文件中的总数据为 10 259 字节,即测量负载为 10 259 字节.使用 BADABING 算法在视频开始传输时也同时发送测量流,在视频播放结束的同时也结束测量发包,其算法参取其默认值.BADABING 算法每次依事先设定的概率来发包,测量包大小为 600B,在第 4.1 节和第 4.2 节的 6 组实验中,最小发包数量为 2 292 个,最大发包数量为 2 568,故 BADABING 在本文实验中最小测量负载为 1 375 200 字节.

### 4.4 算法思想在 H.264 视频文件中应用

经分析与实验,上述算法思想同样也可在 H.264 视频的传输过程中加以应用.

H.264 中,用户数据域信息格式与 Mpeg-4 略有不同,其名称为 User Unregistered SEI(用户未注册 SEI);数据格式也有所不同,具体如图 12 所示.User unregistered SEI(用户未注册 SEI)域的起始标志 0x0605,紧跟着为 size 字段和 16 字节的 UUID,最后是 User\_Data 域.

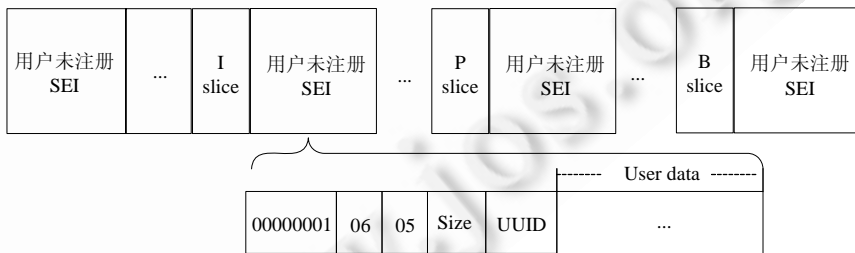


Fig.12 Place of User\_Data field of H.264

图 12 H.264 视频用户数据域位置

User\_Data 域内测量信息设计、接收端丢包率计算、冗余信息设计等均与 MPEG-4 用户数据域中所述方法基本一致,此处不再赘述.

## 5 结论及下一步工作

本文将流媒体数据帧转化为测试序列,通过在视频文件 User\_Data 域中嵌入特定的测量信息完成对丢包的测量.这种测量方法不生成任何新的测试流,不影响视频正常播放,降低了因测量而引入的额外负载.经过分析和实验验证表明:使用 PLBU 测量视频丢包,具有准确度高、入侵度小等优点,且可区分出丢包所属视频帧类型.

实际使用中,可通过在 PLBU 算法中设置丢包报警阈值和测量结果定时输出间隔,实现对网络性能及视频质量实时监测.

进一步分析不同帧丢包对视频质量的影响、借助视频文件 User\_Data 域实现对其他 QoS 参数的测量及 PLBU 算法在视频质量监测系统中的应用,是我们下一步的研究重点.

**致谢** 感谢清华大学计算机科学与技术系刘斌教授和网络中心为本文 Planetlab 实验提供的相关支持.

## References:

- [1] Sommers J, Barford P, Duffield NG, Ron A. A geometric approach to improving active packet loss measurement. *IEEE/ACM Trans. on Networking*, 2008,16(2):307–320. [doi: 10.1109/TNET.2007.900412]
- [2] Almes G, Kalidindi S, Zekauskas M. A one way packet loss metric for IPPM. IETF. RFC 2680. 1999.
- [3] <http://www.ping127001.com/pingpage.htm>
- [4] Adams A, Mahdavi J, Mathis M, Paxson V. Creating a scalable architecture for Internet measurement. In: *Proc. of the INET'98*. Geneva, 1998.
- [5] Duffield NG, Horowitz J, Towsley D, Wei W, Friedman T. Multicast-Based loss inference with missing data. *IEEE Journal on Selected Areas in Communications*, 2002,20(4):700–713.
- [6] Mao YY, Kschischang FR, Li BC, Pasupathy S. A factor graph approach to link loss monitoring in wire-less sensor networks. *IEEE Journal on Selected Areas in Communications*, 2005,23(4):820–829. [doi: 10.1109/JSAC.2005.843557]
- [7] Duffield NG, Horowitz J, Lo Presti F, Towsley D. Explicit loss inference in multicast tomography. *IEEE Trans. on Information Theory*, 2006,52(8):700–713. [doi: 10.1109/TIT.2006.878228]
- [8] Dutta S, Dimitrova N. Smart video streams: 101 uses of the user data field in MPEG. In: *Proc. of the Conf. Record of the 29th Asilomar Conf. on Signals, Systems and Computers*. 1995.
- [9] Huang C, Salama P. Error concealment for shape in MPEG-4 object-based video coding. *IEEE Trans. on Image Processing*, 2005, 14(4):389–396. [doi: 10.1109/TIP.2004.841197]
- [10] Wu M, Yu H, Liu B. Data hiding in image and video: Part II—Designs and applications. *IEEE Trans. on Image Processing*, 2003, 12(6):696–705 [doi: 10.1109/TIP.2003.810589]
- [11] Hoffman D, Fernando G, Goyal V, Civanlar M. Rtp payload format for mpeg1/mpeg2 video. RFC 2250, 1998.
- [12] Kikuchi Y, Nomura T, Fukunaga S, Matsui Y, Kimata H. Rtp payload format for mpeg-4 audio/visual streams, RFC 3016, 2000.
- [13] Wenger S, Hannuksela MM, Stockhammer T, Westerlund M, Singer D. Rtp payload format for h.264 video. RFC 3984, 2005.
- [14] Blanch C, de Groot H, van der Stok P. Inventory of MPEG-4 codecs. Information Society Technologies. 2005.
- [15] Khan A, Sun LF, Ifeachor E. Content-Based video quality prediction for MPEG4 video streaming over wireless networks. *Journal of Multimedia*, 2009,4(4):228–239.
- [16] International Standard ISO/IEC 14496-2. 2nd ed., Printed in Switzerland, 2001.
- [17] Tao S, Apostolopoulos JG, Guérin R. Real-Time monitoring of video quality in IP networks. *IEEE/ACM Trans. on Networking*, 2008,16(5):1052–1065. [doi: 10.1109/TNET.2007.910617]
- [18] Ke CH, Shieh CK, Hwang WS, Ziviani A. An evaluation framework for more realistic simulations of mpeg video transmission. *Journal of Information Science and Engineering*, 2008,24(2):425–440.
- [19] ITU-T. Recommendation P.800-methods for subjective determination of transmission quality. Int'l Telecommunication Union-Telecommunication Standardisation Sector (ITU-T), 1996.
- [20] Mogul J, Deering S. Path mtu discovery. RFC 1191, 1990.
- [21] Mp4box.<http://www.videohelp.com/tools/mp4box>
- [22] Helix-Server. <http://www.realnetworks.com/products-services/helix-server-proxy.aspx>
- [23] Nist net. <http://www-x.antd.nist.gov/nistnet/>
- [24] MAWI. <http://tracer.csl.sony.co.jp/mawi/>
- [25] Tcpreplay. <http://tcpreplay.synfin.net/trac/wiki/>
- [26] Planetlab. <http://www.planet-lab.org/>

- [27] OpenRTSP. <http://www.live555.com/openRTSP/>
- [28] Iperf. <http://iperf.sourceforge.net/>
- [29] Liu H, Xu D. Research on video quality of video transmission over IP. Journal of UEST of China, 2005,34(6):1005-1008 (in Chinese with English abstract).

附中文参考文献:

- [29] 刘昊,许都.视频在 IP 网络中传输的质量研究.电子科技大学学报,2005,34(6):1005-1008.



胡治国(1977-),男,山西灵石人,博士,主要研究领域为计算机网络,分布式系统.

E-mail: huzhiguotj@163.com



张大陆(1949-),男,教授,博士生导师,主要研究领域为计算机网络,分布式系统.

E-mail: daluz@acm.org



谷丽丽(1985-),女,硕士,主要研究领域为网络测量.

E-mail: gll8086@126.com



张起强(1986-),男,硕士,主要研究领域为网络性能测量,视频质量评价.

E-mail: cofol1986@126.com



陈志伟(1986-),男,硕士,主要研究领域为网络性能测量,语音质量评价.

E-mail: sealove\_you@163.com



周华磊(1988-),男,硕士,主要研究领域为网络性能测量,高速流分类.

E-mail: zhouhualei@gmail.com



曹孝晶(1988-),男,硕士,主要研究领域为网络性能测量.

E-mail: 360505227@qq.com