

面向移动网络的视频初始缓冲队列长度测量方法

程光^{1,2}, 房敏^{2,3}, 吴桦^{1,2}

(1. 东南大学网络空间安全学院, 江苏 南京 211189;

2. 东南大学计算机网络和信息集成教育部重点实验室, 江苏 南京 211189;

3. 东南大学计算机科学与工程学院, 江苏 南京 211189)

摘要: 针对视频初始缓冲队列长度难以准确测量的问题, 对非加密的优酷和加密的 YouTube 两类视频平台进行研究, 提出了视频初始缓冲队列长度测量方法。通过识别分析视频流量特征, 关联流量行为与播放状态, 构建视频指纹库, 实现了队列长度的准确测量。实验结果表明, 两类视频平台的测量结果均能够精确到帧, 完全满足准确评估视频体验质量的需要。

关键词: 视频 QoE 参数; 初始缓冲队列; 加密流量分析

中图分类号: TP391.41

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2019179

Measurement of video initial buffer size for mobile network

CHENG Guang^{1,2}, FANG Min^{2,3}, WU Hua^{1,2}

1. School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China

2. Key Laboratory of Computer Network and Information Integration of Ministry of Education of China, Southeast University, Nanjing 211189, China

3. School of Computer Science and Engineering, Southeast University, Nanjing 211189, China

Abstract: To resolve the difficulty in accurately measure the length of video initial buffering queue, two video platforms, non-encrypted Youku and encrypted YouTube, were selected to research, and the video initial buffer queue length measurement method was proposed. By identifying and analyzing the characteristics of video traffic, correlating the traffic behavior with the playing state, constructing video fingerprint database, accurate measurement of queue length was realized. The experimental results show that the measurement results of the two types could be accurate to the frame, fully meeting the need to accurately evaluate the quality of the video experience.

Key words: video QoE parameter, initial buffer queue, encrypted traffic analysis

1 引言

近年来,随着网络基础设施的不断建设及用户数量的不断增加,互联网业务和应用飞速发展,尤其是视频点播服务成为主流,网络中的视频数据规

模急剧膨胀^[1]。2017 年全球视频流量占互联网总流量的 75%, 到 2022 年预计达到 82%^[2]。以视频平台 YouTube 为例, 其拥有超过 10 亿用户, 占互联网总人数的近三分之一, 视频观看总次数中有超一半来自移动设备。可以预见的是, 对于移动网络

收稿日期: 2019-01-19; 修回日期: 2019-04-20

基金项目: 江苏省研究生科研与实践创新计划基金资助项目 (No.SJCX17_0021); 国家重点研发计划基金资助项目 (No.2017YFB0801703); 国家自然科学基金资助项目 (No.61602114, No.61872080, No.61502098); 江苏省自然科学基金资助项目 (No.BK20151416); 赛尔网络下一代互联网技术创新基金资助项目 (No.NGII20170406)

Foundation Items: The Jiangsu Graduate Research and Practice Innovation Program (No.SJCX17_0021), The National Key Research and Development Program of China (No.2017YFB0801703), The National Natural Science Foundation of China (No.61602114, No.61872080, No.61502098), The Natural Science Foundation of Jiangsu Province (No.BK20151416), CERNET Innovation Project (No.NGII20170406)

运营商 (MNO, mobile network operator) 而言, 亟需解决在有限的移动网络带宽上承载大量视频数据的挑战。除了加大网络基础设施的投入外, 要提高网络的利用率, 使硬件资源得到充分利用, 就需要了解用户的视频体验质量 (QoE, quality of experience), 通过准确地监测和评估用户视频 QoE, 动态地调整网络资源和相关参数, 提高移动网络的综合利用率。

研究表明, 与用户视频 QoE 关系最密切的是观看视频过程中出现的缓冲次数和缓冲总时长。文献[3-5]对 YouTube 底层数据传输特征、应用层播放状态和用户的 QoE 体验做了深入分析研究, 发现视频的播放状态 (播放、缓冲等) 与缓冲区状态密切相关, 两者的关系如图 1 所示。

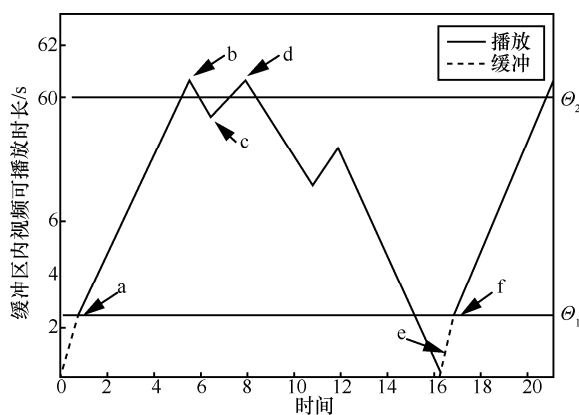


图 1 视频播放状态与缓冲区对应关系

- 1) a 点前, 缓冲区内视频可播放时长小于 θ_1 , 视频处于缓冲状态, 缓冲区内数据只增加不消耗。
- 2) a 到 b 点, 缓冲区内视频可播放时长大于 θ_1 , 视频处于播放状态, 缓冲区内数据增加大于消耗。
- 3) 缓冲区内数据不会无限制增加, b 到 d 点, 缓冲区内视频可播放时长维持在 θ_2 上下。
- 4) 外部原因导致缓冲区内数据被耗尽, e 到 f 点, 视频再次缓冲, 缓冲区需重新缓存至 θ_1 。

综上, 视频发生缓冲与否由缓冲区的状态决定, 而下载速率、码率等相关参数只能预测发生缓冲的趋势, 并不是决定性因素^[6]。 θ_1 的值即代表初始缓冲队列长度, 可以看出, 初始缓冲队列的存在是导致视频缓冲的根本原因, 利用其准确值可以测量视频缓冲的发生和总时长, 对于 MNO 评估用户视频 QoE 有重要价值。

目前, 优酷、爱奇艺、腾讯视频等国内主流视频平台在移动端还未采用加密协议传输数据, 而

YouTube、Netflix 等国外主流视频平台已全面采用加密协议承载数据。本文面向移动网络, 研究了非加密视频平台优酷和加密视频平台 YouTube 的初始缓冲队列长度测量方法, 给出了相关实验方案和测量结果。对于非加密的优酷平台, 建立视频指纹库, 分析视频流与播放状态的关联关系, 通过控制视频终端接收的视频数据量, 实现精确测量初始缓冲队列长度值。对于加密的 YouTube 平台, 首先关联加密视频流与播放状态的对应关系, 建立流量行为模型; 其次构建加密视频指纹库, 支撑加密数据的解析; 再次分析误差, 排除干扰, 设计实验环境; 最后基于上述支撑, 测量计算初始缓冲队列长度值。实验结果表明, 2 个平台的测量结果均能精确到帧, 完全满足 MNO 评估用户视频 QoE 的需要。

本文的主要贡献有以下三点。

- 1) 针对非加密和加密两类视频平台的初始缓冲队列长度展开研究, 测量结果精确到帧, 测量方法可推广或借鉴至其他视频平台。
- 2) 基于数据层面流量特征与应用层面播放状态的关联关系, 建立流量行为模型, 打通了底层数据传输与上层用户感知。
- 3) 针对加密视频数据的分析, 利用中间人技术, 设计了新的加密视频指纹库构建方案, 改进了传统指纹库存在的缺陷。

2 相关研究

视频业务的飞速发展, 推动了视频数据传输机制和视频 QoE 评估的相关研究。Hoßfeld 等^[7-9]提出, 目前视频平台在传输层采用可靠的 TCP 协议, 用户不会观察到视频图像的失真, 与文献[3-4]的结论一致, 认为用户的视频 QoE 主要与观看视频时产生的缓冲有关, 其中, 缓冲又分为视频开始播放之前的初始缓冲和视频播放过程中的卡顿缓冲。为了量化用户的视频观看体验, 华为公司联合牛津大学建立了一个用户视频观看体验的量化模型 vMOS (video mean opinion score)^[10], 模型模仿语音业务, 利用 MOS 量化语音质量^[11]的思想, 根据在线视频初始缓冲时延 (sStartDelay)、视频源质量 (sQuality)、视频卡顿因素 (sStalling) 等客观因素来量化用户观看视频的体验。综合已有研究可以发现, 缓冲决定了用户视频 QoE, 造成缓冲的初始缓冲队列长度是视频 QoE 评估的重要参数。

目前还没有针对国内视频平台初始缓冲队列长度的相关研究，国外视频平台的相关研究主要聚焦在 YouTube 上。文献[3]给出了初始缓冲队列长度的定义，并利用初始缓冲队列长度描述了 YouTube 的流量模型。但是文献[3]的模型中并未给出测量方法，而是直接采用了 Ameigeiras 等^[12]和 Schatz 等^[13]的研究结果，他们给出的初始缓冲数据可播放时长分别为 2.2 s 和 1.9 s。Ameigeiras 等^[12]利用接收速率、视频码率、带宽等参数估算初始缓冲数据时长，Schatz 等^[13]则利用的数据分组的时间特征，如分组到达时间、分组间隔时长等参数来估算。但这 2 个研究结果是在 YouTube 未采用加密协议，且应用层采用渐进式视频下载技术（HPD，HTTP progressive download）时得到的，而当前 YouTube 已全面采用加密协议承载数据，在应用层已弃用 HPD，改用 DASH（dynamic adaptive streaming over HTTP）视频传输技术，两者在传输机制和流量特征上有很大区别。

Burger 等^[14]基于离散时间分析对视频缓冲区状态进行建模，利用视频的分片到达时间、分片传输时长、分片平均下载速率、分片平均码率等分片在传输层面的参数，计算出描述缓冲区模型的 3 个关键阈值：初始缓冲阈值（initial buffering threshold）、暂停缓冲阈值（pause buffering threshold）和继续缓冲阈值（continue buffering threshold）。其中初始缓冲阈值的结果约为 0 s。Burger 等^[14]的研究重点是描述视频平台流量模型，并未关联播放状态，实验中的网络状况相对较好，只有初始缓冲，且缓冲时延很短，因此 0 s 的初始缓冲阈值对文献[14]中 QoE 评估的准确性影响有限。但当网络状况较差时，初始缓冲的时延相当明显，0 s 显然不能再用来描述初始缓冲队列的长度。Krishnamoorthi 等^[15]设计了测量检测系统 BUFFEST，可以从 HTTP 和 HTTPS 的流量中识别出视频流量，并评估当前的缓冲区状态，BUFFEST 利用的是 TCP 和 IP 层分组的相关特征，如到达时间、分组大小、分组间隔时长等参数。与 Krishnamoorthi 等^[15]的研究类似的有 Tsilimantos 等^[16]的研究，仅通过 IP 层视频分组的相关特征值，利用机器学习方法训练模型，实时地分析检测缓冲区的状态。但是，Krishnamoorthi 等^[15]和 Tsilimantos 等^[16]的研究也都没有实现对初始缓冲队列的精确测量。

综合已有关于视频初始缓冲队列长度的研

究，发现主要是利用的网络侧分组数据的特征参数，来测量评估缓冲区的大小，进而估算初始缓冲队列的大小。同时，已有的估算值大多精确到秒，而视频数据的基本单位为帧，视频缓冲区实际排列的是解码后的原始视频帧队列，因此能够精确描述初始缓冲队列长度的单位是视频帧个数，而非可播放几秒钟的时长。本文针对非加密优酷和加密 YouTube 这 2 个视频平台的初始缓冲队列长度展开研究，精确测量队列长度值，测量结果精确到帧。

3 非加密平台优酷的测量方法

对于采用非加密协议传输视频的平台，研究初始缓冲队列长度先要确定满足播放所需的最小视频数据量，由于数据未加密，容易算出数据量对应的帧数。本节基于优酷视频传输机制和数据特征的研究分析，建立优酷视频帧偏移信息指纹库；通过控制优酷终端应用接收的视频数据量，建立视频流量与播放行为的关联关系，分析计算优酷终端的初始缓冲队列长度值。

3.1 非加密平台测量方法总体思路

优酷移动端平台采用非加密协议传输视频数据，因此通过采集分析中间端数据，可以实时计算当前终端应用已经接收到视频数据量。反之，在中间端通过阻断视频数据的传输，可以定量控制终端应用接收到的视频数据量。

优酷视频的 MP4 文件格式如图 2 所示，解析 MP4 文件可以定位“stco”结构体，其中描述了视频每一帧相对于视频文件头的偏移信息，以此建立优酷视频的视频帧偏移信息指纹库。理论上，逐帧控制优酷终端应用接收的视频数据量，可以找到恰好满足播放条件的帧数，此值即为优酷移动终端的初始缓冲队列长度。

3.2 非加密平台测量方法实施步骤

步骤 1 建立视频的帧偏移信息指纹库

确定实验视频集，预先抓取对应分辨率的完整视频，利用算法 1 解析每个优酷视频 MP4 文件，建立帧偏移信息指纹库。

算法 1 优酷视频的视频帧偏移信息指纹库建立算法

输入 优酷视频 MP4 文件 file

输出 优酷视频帧偏移信息指纹库

1) 查找 file 中“stco”字符串位置 pos，读取帧总个数 N

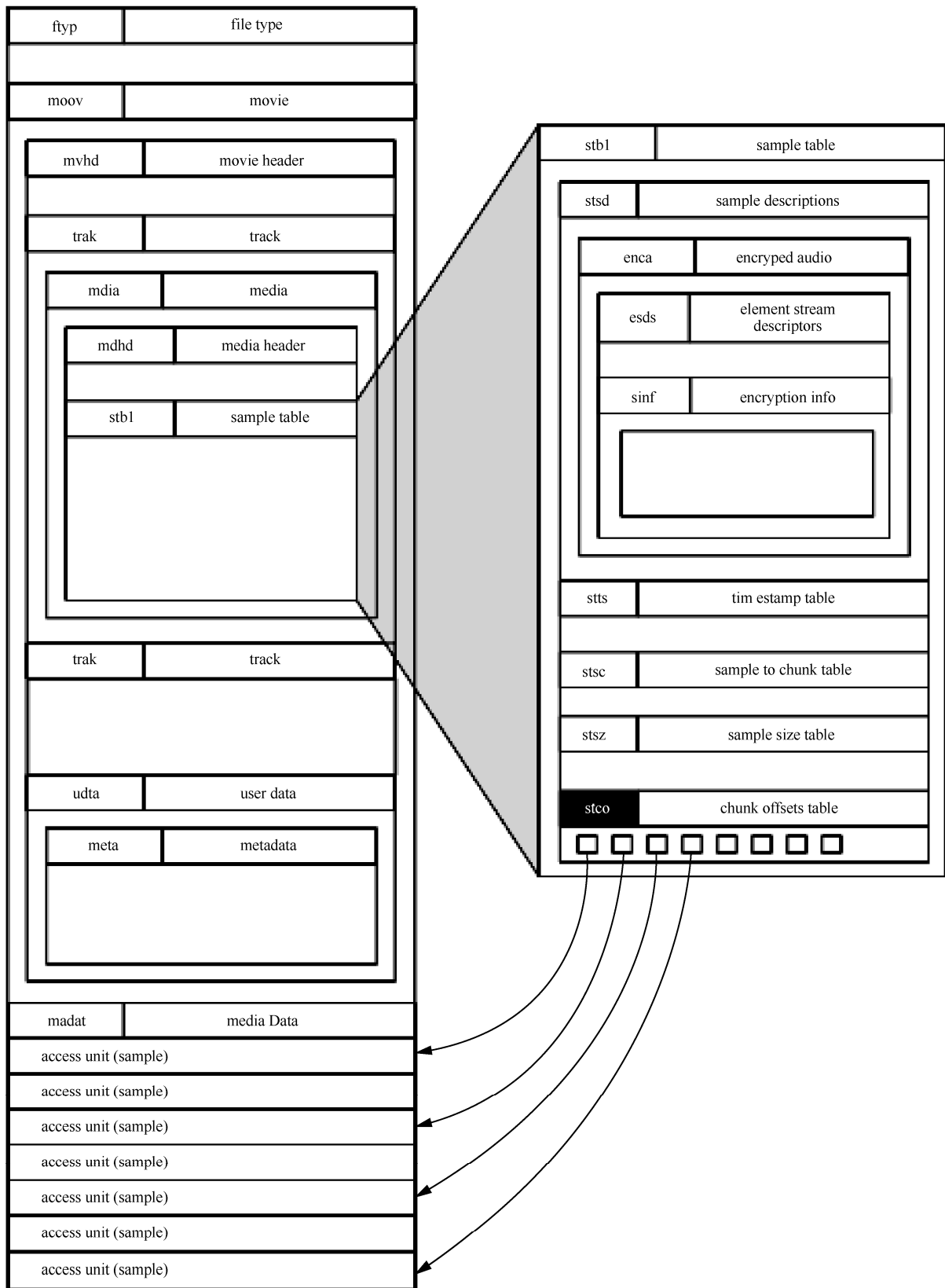


图 2 优酷视频 MP4 文件格式

```

pos, N=SearchFile(file,“stco”)
2) for i=1 to N do
① offset=ReadFile(file,pos+12(i-1),4)
② index=i
③ 帧编号 i, 偏移 offset 入库
end for

```

步骤 2 中间端视频数据控制

移动终端连接 PC 共享的 Wi-Fi, PC 作为中间端,部署基于 winpcap 开发的优酷视频流测量程序。根据算法 2,中间端程序实时分析数据分组,若有优酷视频请求,就跟踪该请求的 TCP 流,当此流的视频响应数据总量大于预先设定的断网条件,就在程序中调用程序断网模块,切断 Wi-Fi。此时,移动终端优酷应用接收到的视频数据量约等于预先设定的断网条件,若大于初始缓冲队列长度,终端优酷视频可以播放;若小于初始缓冲队列长度,终端优酷视频无法播放。

算法 2 控制终端接收定量视频数据算法

输入 断网条件 condition

输出 终端优酷视频播放状态,“1”表示播放,“0”表示未播放

1) 实时监听网络,利用 winpcap 接口读取分组 packet

```
packet=PcapLoop()
```

2) 判断 packet 是否是优酷视频请求分组

```
if packet is YoukuResquest do
```

```
① 跟踪 TCP 流,嗅探优酷响应分组 yPacket
```

```
② totalSize=0
```

```
③ while totalSize<condition do
```

```
④ totalSize+=PayloadSize(yPacket)
```

```
end while
```

```
⑤ CloseWi-Fi()
```

```
end if
```

3) 记录终端优酷视频是否播放,“1”表示播放,“0”未播放

4) 数据综合分析

每次测量过程中,在移动终端侧使用 tcpdump 采集 pcap 数据分组,用来确定终端优酷应用实际接收到的数据量,配合帧偏移信息指纹库可以确定该数据量对应的帧数和可播放时长。若当次测量优酷视频可以播放,实际接收到的帧数大于或等于初始缓冲队列长度;若当次测量优酷视频无法播放,实际接收到的帧数小于初始缓冲队列长度。

3.3 非加密平台测量方法误差分析

实验发现中间端 PC 的断网模块存在时延(约 0.5 s),由于该时延无法克服,终端优酷应用实际接收到的视频数据量一定会大于中间端的预设断网条件。

为了尽量减小与预设值的偏差,需要对网络进行限速,使断网时延阶段终端接收到尽量少的分组,以减小偏差。同时限速要在一定范围内,因为限速过大,会导致分组丢失增加,造成无法准确计算实际接收的视频数据量。通过实验验证,200 KB/s 的限速标准可以满足测量需求,此时从调用断网模块到实际终端停止接收分组,优酷终端应用还会接收到约 40~100 KB 的视频数据。

4 加密平台 YouTube 的测量方法

YouTube 等视频服务商为了保护用户隐私和网络安全,相继采用加密协议传输视频数据。加密协议导致基于 DPI (deep packet inspection) 的传统方法已无法分析。本节首先介绍了移动网络下 YouTube 传输机制,基于传输机制的分析,同时借鉴优酷平台的测量方法,给出了测量方法的总体思路,并详细描述了测量方法的实施步骤,设计实验方案来精确测量 YouTube 的初始缓冲队列长度值。

4.1 YouTube 传输机制和文件格式

优酷的传输机制虽对视频进行了分片,但分片相对简单,按照固定大小将视频文件剪裁成几块。而 YouTube 采用 DASH 协议传输视频,同时又采用了加密协议,测量初始缓冲队列长度首先需要了解 YouTube 的 DASH 传输机制。

DASH 视频传输流程如图 3 所示,首先在视频服务器部署同一个视频的不同分辨率视频分片组及 MPD (media presentation description) 分片目录文件。当用户点击播放视频后,首先请求下载 MPD 文件,YouTube 客户端的 MPD 解析模块对 MPD 文件进行处理,获取该视频所有分辨率、码率视频资源的 URL 信息,然后根据目前终端网络状况向视频服务器请求特定分辨率、码率的视频片段。同时,视频播放过程中,YouTube 客户端可以根据自身网络状况切换请求不同分辨率、码率的视频分片,实现视频的自适应播放。

借鉴非加密平台优酷的初始缓冲队列长度测量方法,控制 YouTube 终端仅接收到一个视频分片后切断终端的网络,发现视频都会正常播放,因此

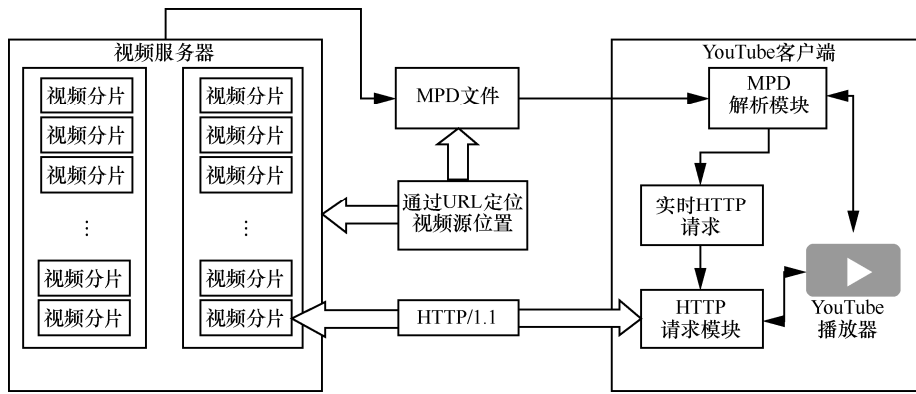


图 3 DASH 视频传输流程

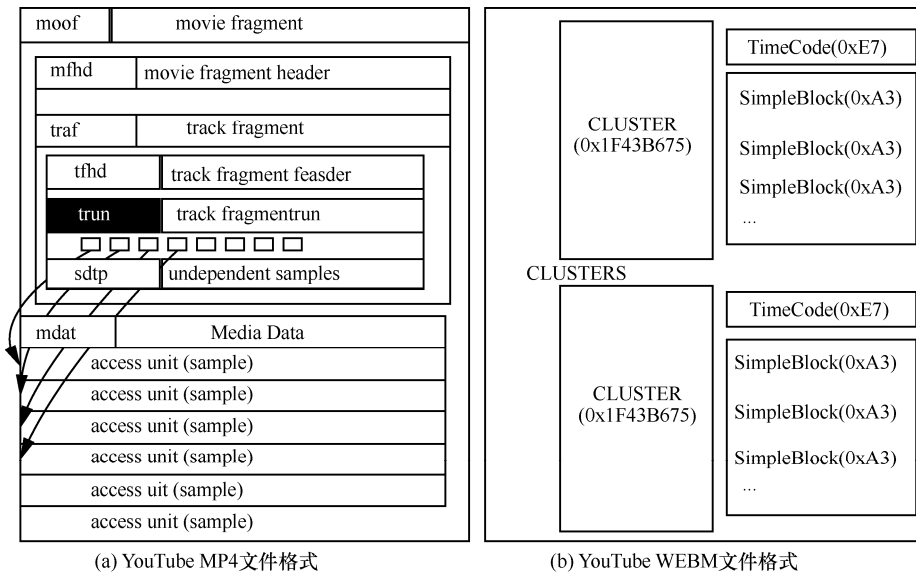


图 4 YouTube 的 2 种文件格式

YouTube 的初始缓冲队列长度小于一个分片的时长。目前，YouTube DASH 的分片时长一般为 5 s 和 10 s 这 2 种规格。

YouTube 的音频和视频是分离的，音频、视频分片的文件格式主要有 MP4 和 WEBM 这 2 种文件格式，如图 4 所示。解析 MP4 文件中的“trun”结构体，或者读取 WEBM 中每一个“SimpleBlock”的位置，可以计算出每一帧相对于分片文件头的偏移。但是由于 YouTube 采用加密协议，无法直接从加密流量中解析这 2 种文件格式，后文介绍的构建加密视频指纹库用以解决此问题。

4.2 加密平台测量方法总体思路

优酷初始缓冲队列长度的测量方法是通过控制终端应用接收到的视频数据量，建立播放行为与流量的对应关系，是一种相对简单的流量行为模型。参考这一思路，针对采用加密协议传输视频数据的

YouTube 平台，仍要首先建立 YouTube 的加密流量行为模型，关联分析加密流量与视频播放状态，从而在加密流量中识别出初始缓冲阶段的视频数据。进一步分析，由于 YouTube 采用了加密协议，识别出的初始缓冲数据仍无法直接计算出初始缓冲队列长度值，这一难点可以通过构建加密视频指纹库的方法得以解决。在建立流量行为模型和构建加密视频指纹库的基础上，本文分析 YouTube 传输机制和加密协议对测量过程可能造成的误差，排除相关影响因素，设计合理的实验环境。最后，针对 YouTube 实验视频集，利用实验环境采集数据，综合分析计算初始缓冲队列长度值。算法 3 给出了测量方法的总体思路和步骤，下一节将针对算法中每一步骤进行详细介绍。

算法 3 YouTube 初始缓冲队列长度测量算法
步骤 1 建立流量行为模型。分析 YouTube 加

密视频流与播放状态的关联关系，从而从加密流中识别出初始缓冲数据。

步骤 2 构建加密视频指纹库。改进传统指纹库构建方案，辅助加密视频流数据的分析。

步骤 3 影响因素分析排除。分析测量误差，排除干扰因素，设计实验环境。

步骤 4 综合计算分析。利用前三步的流量行为模型、加密视频指纹库和实验环境，采集实验视频集流量数据，分析计算初始缓冲队列长度值。

4.3 加密平台测量方法实施步骤

1) 建立流量行为模型

图 5 给出了视频播放过程中的所有状态，初始缓冲数据就是“初始缓冲”至“达到播放条件”这一阶段接收到的视频数据。

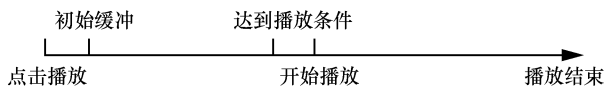


图 5 YouTube 视频的播放状态

识别初缓数据，需要建立视频流与播放状态的对应关系，而 YouTube 原生 APP 无法实现这一目标。解决方案是借助 YouTube 官方提供的 YouTube Android API 库^[18]，开发基于原生 APP 之上的 Android 应用，该应用可以依赖原生 APP 播放 YouTube 视频。同时，开发库提供 onVideoStarted、onBuffering、onPlaying、onVideoEnded 这 4 个回调函数，可以在终端获取准确的视频播放状态。

表 1 列出了播放状态与 YouTube API 回调函数的对应关系，onBuffering 在初始缓冲阶段会执行 2 次，第二个 onBuffering 时间点与 onPlaying 时间点几乎一致，2 个 onBuffering 回调函数之间下载的视频数据即为初始缓冲队列长度对应的初缓数据。

表 1 YouTube 播放状态与 API 回调函数关系

播放状态	对应 YouTube API 的回调函数
点击播放	onVideoStarted
初始缓冲开始	onBuffering
达到播放条件	onBuffering
开始播放	onPlaying
播放结束	onVideoEnded

利用 API 可以在终端得到准确的播放状态，下一步需要将播放状态和加密的视频流建立关联。解决方案是在回调函数处主动构造状态标识分组，则视频流中混入播放状态的“时间点”。如图 6 所示，

通过此对应关系可以在加密的视频流中剥离出缓冲阶段的流量数据，使视频流量与播放状态建立关联关系，构建流量行为模型。

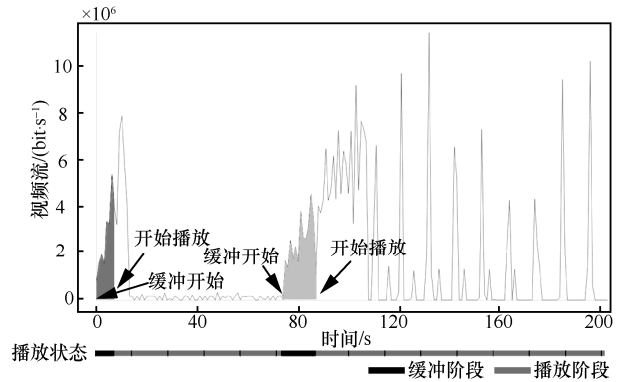


图 6 建立视频流与播放状态关联的流量行为模型

2) 构建加密视频指纹库

建立流量行为模型，识别出初始缓冲数据之后，需要确定初始缓存数据对应的帧数，但由于视频加密，无法直接得到这一数据。为实现初始缓冲队列长度测量结果精确到帧的目标，需要构建包括视频帧偏移信息库在内的一系列加密视频指纹库。

构建指纹库是分析加密流量数据的一般做法，传统 YouTube 加密视频指纹库构建方法，是对热点视频在测试终端进行点播，同时在网络接入点采集数据流量。对视频请求分组后服务器响应的一簇密集响应分组进行累加，认定此簇分组对应一个视频片段，由此构建视频的指纹库。传统方法主要存在以下 3 个问题。

① DASH 传输机制中音频、视频分离在 2 条流中，传统方法忽视了数据量较小的音频片段被累加所造成的误差。

② YouTube 采用自适应的传输机制，每次传输时分辨率随网络状况改变而改变，传统方法无法确定所采指纹对应的视频分辨率。

③ 仅对加密数据传输进行统计，实际上获得的只是当次的传输指纹，并不一定能获取视频所有可能的指纹信息。

针对上述问题，本文提出的构建加密视频指纹库改进方案如图 7 所示，该方案在中间采集设备上使用中间人技术，获得目标视频的明文数据，解析采集的明文数据文件，获得视频的基本信息指纹，并基于 YouTube 视频分发机制和 DASH 传输机制进行组合分析，进一步获得与传输状态相关的视频传

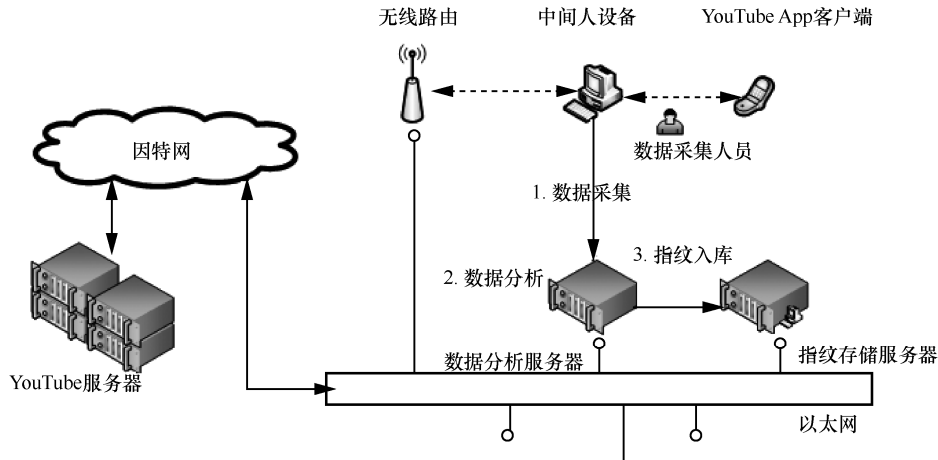


图 7 构建加密视频指纹库

输指纹，将这些指纹存入数据库中。表 2 给出了指纹库中包含的各类信息库及其说明。

表 2 加密视频指纹库的组成及其说明

信息库名称	说明
视频基本信息库	包含视频 ID、帧率、时长、视频所有可用分辨率等信息
分片信息库	包含音频分片和视频分片的格式、编码方式、各分辨率下分片大小等信息
帧偏移信息库	包含各分辨率下，分片内每一帧相对分片头的偏移信息（单位为 B）
分片组合信息库	包含所有分辨率、码率的视频分片和音频分片的组合信息

与传统指纹库构建方案相比，本文提出的方案主要有以下 3 个改进。

① 改进方案基于中间人技术，解析所有分辨率下的明文数据，读取各种码率的音频、视频分片信息，避免了传统方法无法区分音频、视频分片的问题。

② 从明文中读取分片的长度等信息码，构成了加密视频的基本信息指纹。基于传输机制，对基本信息指纹进一步分析，计算出在不同的网络条件下所有音、视频可能的传输组合，获得视频在不同的网络服务质量下所有可能的传输指纹。

③ 视频指纹是从中间人设备采集的明文数据中获得，与具体的传输环境无关，因此可以应用于网络性能不断变化的真实网络，具有较好的适用性。

3) 影响因素分析排除

YouTube 在传输层之上采用了 TLS 加密协议，TLS 协议会对应用层视频数据进行分块，一个数据块大小一般为 16 KB。在终端为了解密和验证每一个

TLS 数据块，必须保证 16 KB 加密数据都已被完整接收。

图 8 给出不限速下 YouTube 视频数据到达终端的示意图，图中长方形表示一个 16 KB 的 TLS 块。由于一般视频帧的大小不会大于 16 KB，故初始缓冲队列长度对应的视频帧大概率位于 TLS 块的内部，第四个 16 KB TLS 块中的黑色小方块给出示意。在 t_1 时间点时，对接收到的数据而言，已经达到了播放条件，但由于上层应用需要经历解协议、解封装、解码、判断当前缓冲队列长度等一系列过程，到视频真正开始播放已经在 t_1 之后的 t_2 时间点（ t_2 时间点对应 YouTube API 回调函数中的 onPlaying）。图 8 中， t_1 至 t_2 时间段内，又接收到第五个完整 16 KB 的 TLS 块。测量中会判定 onPlaying 前的完整 TLS 块均为初缓数据，因此第五块会被当作初始缓冲的数据，在计算初始缓冲队列长度时造成误差。

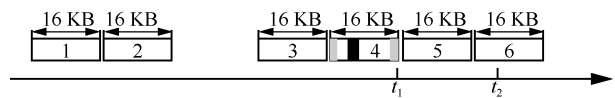


图 8 不限速下接收到 TLS 块示意

图 9 是限速后的 YouTube 视频数据到达终端的示意图，此时 TLS 块之间的时间间隔被拉长，其中， t_1' 时间点达到了播放条件，同样是经过一系列的过程，到 t_2' 时间点开始播放（ t_2' 时间点对应 YouTube API 回调函数中的 onPlaying）。至 t_2' 时间点第五个 TLS 块仍未接收完整，不会被算作初始缓冲的数据。此时初缓队列长度就是由初始缓冲队列长度对应的视频帧（第四个 TLS 块中的黑色小方块给出示意）所在 TLS 块算出，误差得以排除。

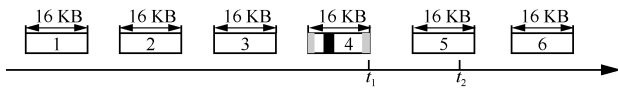


图 9 限速下接收到 TLS 块示意

综上，为了保证初始缓冲队列长度测量结果的准确性，需要对实验环境进行设计，排除由于终端播放器正式播放时间点滞后于真实达到播放条件的时间点，导致会将多余的完整 TLS 块被误判成初缓数据带来的误差。因此，在采集加密视频流的实验环境中设计了限速模块，限速的标准是既要保证视频不会因为带宽太低而无法播放，也要保证排除“多余”TLS 完整块的干扰。实验结果表明，限速为 1 Mbit/s，可以满足上述 2 个要求。

4) 综合计算分析

至此已经确定出初缓数据，并能够确定初缓队列长度对应的视频帧所在的 TLS 块，但仍无法给出这一视频帧的具体位置。如图 10 所示，虚线框出的视频帧即为初始缓冲队列长度的准确值。

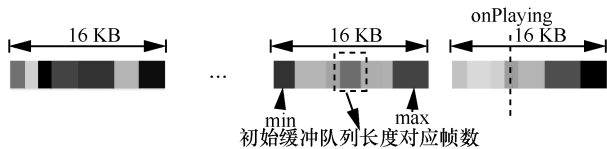


图 10 初始缓冲队列长度值所在区间

对于一个视频而言，无法直接确定初始缓冲队列长度具体对应哪一帧，但如图 10 所示，可以参照指纹库中的帧偏移信息，确定 min 和 max 的具体值，则初始缓冲队列长度位于 [min, max] 区间内部。

对于实验视频集而言，每一个视频都可以计算出一个初始缓冲队列长度所在的区间，如图 11 所示，对所有视频计算出的区间取交集，且满足结果为整数的条件，即可得到一个精确到视频帧的初始缓冲队列长度值，具体步骤如算法 4 所示。

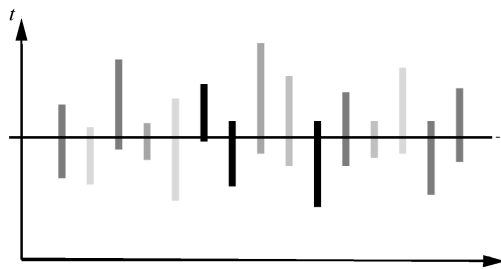


图 11 取交集计算初始缓冲队列长度值示意

算法 4 实验视频集处理方法

输入 实验视频集 $\{V_i | i=1, \dots, N\}$

输出 初始缓冲队列长度 L 取值区间, $L \in [\min, \max]$, 且 L 为整数

- ① 取实验视频集中一个视频 V_i 。
- ② 限速 1 Mbit/s 采集实验数据, 计算 onPlaying 前完整 TLS 块数 B_i 。
- ③ 查帧偏移信息库, 计算 $16(B_i-1)$ KB 和 $16 B_i$ KB 对应帧数 \min_i 和 \max_i , 以及 V_i 帧率。
- ④ 对所有视频重复①到③, 得到 N 个区间 $[\min_i, \max_i], i=1, \dots, N$ 。
- ⑤ 相同帧率视频的 $[\min_i, \max_i]$ 区间取交集, 得到 $[\min, \max]$ 。

5 实验分析

5.1 实验环境

实验共使用了 4 部测试手机, 测试手机的参数信息如表 3 所示。

表 3 测试手机参数信息

设备型号	系统版本	分辨率	内存/GB
三星 Galasy S6	Android 5.1	2 560 像素×1 440 像素	3
三星 Galasy Note5	Android 6.0	2 560 像素×1 440 像素	4
华为 Mate8	Android 7.0	1 920 像素×1 080 像素	3
华为 Mate2	Android 4.2	1 280 像素×720 像素	2

为满足 YouTube 加密视频平台的分析, 全部测试手机均安装了基于 Android YouTube API 开发的应用, 应用中集成了 tcpdump 分组抓取模块, 可以在 YouTube 视频播放过程中自动抓取流量数据。

图 12 给出了非加密优酷平台的实验数据采集环境, 手机终端连接中间端 PC 共享的 Wi-Fi, 中间端 PC 开发部署优酷视频流测量程序。

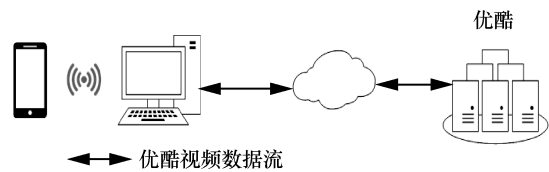


图 12 优酷平台实验数据采集环境

YouTube 视频流量数据采集自蜂窝网络 (3G/4G) 和 Wi-Fi 这 2 种网络环境, 采集环境如图 13 所示。

5.2 数据采集

1) 优酷平台数据采集

确定优酷实验视频集, 选择起始画面变化相

对剧烈的视频，保证每一帧的数据量较大，使数据传输时，一帧需要多个分组，可以减小测量误差。同时覆盖优酷视频目前主流帧率：15 帧/秒、23 帧/秒和 25 帧/秒。最终一共确定 10 个测试视频。实验前后共采集了 10 组数据，每组数据约 50 例。

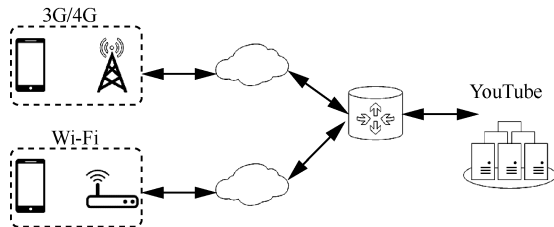


图 13 YouTube 平台实验数据采集环境

2) YouTube 平台数据采集

确定 YouTube 实验视频集，选取标准有 3 个条件：① 覆盖 0~2 min、2~5 min 和大于 5 min 这 3 种时长要求；② 视频的开始几秒钟画面变化比较剧烈，目的是使起始阶段视频帧的平均大小不至于太小，导致每个 16 KB 的 TLS 块可能包含大量的帧，增大偏差；③ 覆盖了目前 YouTube 上最普遍的帧率，如 4 帧/秒、25 帧/秒和 30 帧/秒等。最终确定了 61 个测试视频。

实验前后共采集了 4 批数据，具体的采集时间等信息如表 4 所示。

表 4 YouTube 数据采集信息

编号	采集日期	采集地点	接入网络	采集数量/次
1	2017-12-23	香港	3G/4G	1 327
2	2017-12-24	香港	Wi-Fi	618
3	2018-03-08	南京	Wi-Fi	560
4	2018-04-06	南京	Wi-Fi	550

5.3 优酷实验结果分析

表 5 给出了 10 组实验数据其中一组的部分结果，此视频的帧率为 23 帧/秒。

表 5 优酷一组实验数据的部分结果

编号	是否播放	实际接收帧数/帧	换算成时长/s
1	1	237	10.304
2	0	232	10.087
3	0	228	9.913
4	1	234	10.174
5	1	237	10.304

表 5 的“是否播放”列中，“1”代表此次实验优酷终端应用最终能够播放，表明接收到的视频数据量满足了播放条件，即实际接收的帧数 \geq 初始缓冲队列长度；“0”代表此次实验优酷终端应用未播放，即实际接收的帧数 $<$ 初始缓冲队列长度。

图 14 给出了表 5 对应数据所有结果的汇总。当实际接收到帧数 \geq 234 时，此次测量能够播放；当实际接收到帧数 $<$ 233 时，此次测量未播放。故此组实验结果表明，对于 23 帧/秒的视频而言，初始缓冲队列长度位于区间[233, 234]内。

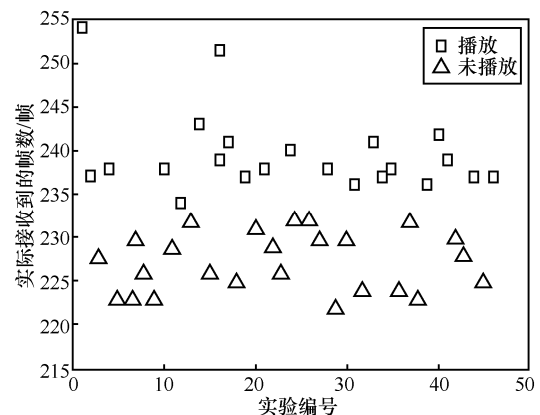


图 14 帧率为 23 帧/秒优酷视频的一组实验结果

综合所有 10 组实验数据，其结果如表 6 所示，3 种帧率下初始缓冲队列长度值不同，但是换算成时长，均略大于 10 s。

表 6 优酷平台初缓队列长度测量结果

帧率/(帧·秒 ⁻¹)	初缓队列长度/帧	换算成时长/s
15	[155, 157]	[10.267, 10.467]
23	[233, 234]	[10.130, 10.174]
25	253	10.120

由于实验本身无法克服的误差（断网时延），无法精确控制终端应用实际接收到的帧数，因此 15 帧/秒帧率下没有得到实际接收 155 个或 156 个视频帧的情况，23 帧/秒帧率下没有得到实际接收 233 个视频帧的情况，所以这 2 种帧率下初始缓冲队列长度无法精确到具体的帧数。

5.4 YouTube 实验结果分析

如表 7 所示，给出了 YouTube 实验视频集中的部分结果，表中所有数值均为视频帧的个数。

实验视频集中所有 24 帧/秒视频的测量结果如图 15 所示，可以看出，所有结果的区间存在重合，重合区域为[60, 61]。

表 7 部分 YouTube 视频的计算结果

视频帧率/(帧·秒 ⁻¹)	视频 ID	min	max	计算结果/帧
24	e5N_Tq1EtRQ	59	61	[60, 61]
	nx1R-eHSkFM	60	61	
	9OeZYhjyweQ	51	64	
	Mqaobr6w6_I	55	61	
	prmQgSpV3fA	60	61	
25	QAwL0O5nXe0	62	64	[62, 63]
	msUKf_cEKo0	55	66	
	OsSKFlk8oEo	59	73	
	p-yIGS9eW-o	54	63	
	_9YMpuLDnwo	60	67	
30	IWwhYPjyE1U	68	76	[75, 76]
	h_67XloYXTc	55	84	
	JVHFMBfG-cM	75	83	
	iMg9ibfcBnM	74	85	
	TZSVf-yD7N8	75	76	

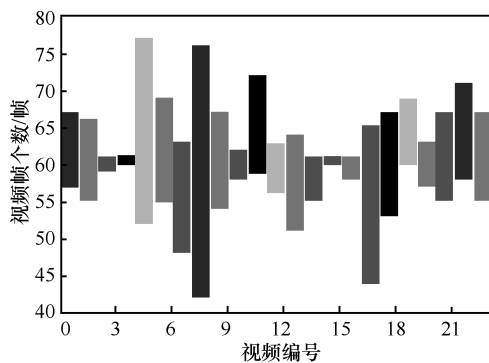


图 15 24 帧/秒视频集测量结果

统计所有帧率的视频，24 帧/秒视频的初始缓冲队列长度区间为[60, 61]，换算成时长为[2.5, 2.542]；25 帧/秒的区间为[62, 63]，换算成时长为[2.48, 2.52]；30 帧/秒的区间为[75, 76]，换算成时长为[2.5, 2.533]。因此，可以认定 YouTube 的初始缓冲队列长度以时长来描述应为 2.5 s。

进一步分析实验结果，初始缓冲队列长度在 YouTube 内部并不是一个不变的值，对于不同帧率的视频，有一个不同的满足播放条件所需的帧数。

- 1) 对于 24 帧/秒的视频而言，当已缓冲帧数 ≥ 60 即满足播放条件。
- 2) 对于 25 帧/秒的视频而言，当已缓冲帧数 ≥ 63 即满足播放条件。
- 3) 对于 30 帧/秒的视频而言，当已缓冲帧数 ≥ 75 即满足播放条件。

在计算初始缓冲队列长度时，进行了限速的处理，为了排除限速可能影响初始缓冲队列长度本身，又做了 2 组对照试验，实验结果汇总如图 16 所示。图 16 中的 \circ 为图 8 中 $t_1 \sim t_2$ 的时间间隔， \triangle 为图 9 中 $t'_1 \sim t'_2$ 的时间间隔。 $t_1 \sim t_2$ 和 $t'_1 \sim t'_2$ 都是上层应用经历的解协议、解封装、解码、判断当前缓冲队列长度等一系列过程的时长，对于同一部硬件终端而言，这 2 个时间间隔无论限速与否，应大致相同。

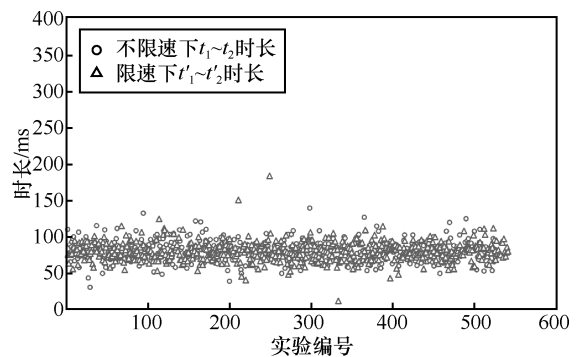


图 16 不限速下 $t_1 \sim t_2$ 与限速下 $t'_1 \sim t'_2$ 对比

从图 16 的实验结果可以看出， $t_1 \sim t_2$ 和 $t'_1 \sim t'_2$ 的时间间隔大致相同，可以论证限速并不会影响初缓队列长度的值，其值就是播放器内部设置的一个确定阈值。

6 结束语

视频终端初始缓冲队列长度是评估用户视频 QoE 的重要参数，但现有关于队列长度的研究并未给出其准确值。本文针对非加密优酷和加密 YouTube 这 2 个视频平台，分别给出了面向移动网络的视频初始缓冲队列长度的测量方法，以建立流量行为模型和辅助指纹库为基础，分析误差，排除干扰因素，设计实验环境，采集特定网络环境下实验视频集的数据，分析计算初始缓冲队列长度，测量结果精确到视频帧。

优酷的测量方法核心思想是 2 条：一是数据非加密，解构视频文件可确定每一帧的偏移；二是控制终端应用接收到的视频数据量，找到恰好达到播放条件的帧个数。这对于其他非加密平台，如爱奇艺、腾讯视频等同样可以实现，因此优酷平台的测量方法对于其他非加密视频平台同样适用。

YouTube 的测量方法关键步骤也是 2 条：一是建立流量行为模型，识别初始缓冲数据；二是构建

加密视频指纹库, 辅助加密数据分析。对于其他加密视频平台, 加密视频指纹库构建方案同样适用; 流量行为模型依赖于官方 API 库, 目前 top 级视频平台为扩展平台应用, 均会提供开发者 API 接口。因此其他加密视频平台可依据自身开放接口情况, 借鉴 YouTube 的测量方法, 实现初始缓冲队列长度的测量。

下一步的工作主要有两部分: 一是将两类测量方法推广至其他类似平台, 验证其通用性; 二是结合视频码率识别技术, 利用初始缓冲队列长度值计算初始时延和卡顿时延, 完善视频 QoE 评估模型。

参考文献:

- [1] SANDVINE. 2018 Global internet phenomena report[R]. [S.l. : s.n.], 2018.
- [2] CISCO. Cisco visual networking index: forecast and methodology 2017—2022 [R]. San Francisco: Cisco, 2018.
- [3] WAMSER F, CASAS P, SEUFERT M, et al. Modeling the YouTube stack: from packets to quality of experience[J]. Computer Networks, 2016(109): 211-224.
- [4] SEUFERT M, WEHNER N, WAMSER F, et al. Unsupervised QoE field study for mobile YouTube video streaming with YoMoApp[C]//The 9th International Conference on Quality of Multimedia Experience. IEEE, 2017: 1-6.
- [5] WAMSER F, HÖFNER S, SEUFERT M, et al. Server and content selection for MPEG DASH video streaming with client information[C]//The Workshop on QoE-based Analysis and Management of Data Communication Networks. ACM, 2017: 19-24.
- [6] ORSOLIC I, SUZnjeVIC M, SKORIN-KAPOV L. YouTube QoE estimation from encrypted traffic: comparison of test methodologies and machine learning based models[C]//The 10th International Conference on Quality of Multimedia Experience. IEEE, 2018: 1-6.
- [7] HÖBFELD T, EGGER S, SCHATZ R, et al. Initial delay vs. interruptions: between the devil and the deep blue sea[C]// The 4th Fourth International Workshop on Quality of Multimedia Experience. IEEE, 2012: 1-6.
- [8] HÖBFELD T, SCHATZ R, KRIEGER U R. QoE of YouTube video streaming for current Internet transport protocols[M]//Heidelberg: Springer, 2014: 136-150.
- [9] HÖBFELD T, VARELA M, HEEGAARD P E, et al. QoE analysis of the setup of different internet services for FIFO server systems[C]//International Conference on Measurement, Modelling and Evaluation of Computing Systems. Springer, 2018: 234-248.
- [10] 华为技术有限公司. 固定视频指标体系白皮书[R]. (2017-07-10) [2019-01-09]. Huawei Technologies Co., Ltd. Fixed video indicators system white paper[R]. (2017-07-10) [2019-01-09].
- [11] NUNES R D, PEREIRA C H, ROSA R L, et al. Real-time evaluation of speech quality in mobile communication services[C]// IEEE International Conference on Consumer Electronics. IEEE, 2016: 389-390.
- [12] AMEIGEIRAS P, RAMOS - MUNOZ J J, NAVARRO - ORTIZ J, et al. Analysis and modelling of YouTube traffic[J]. Transactions on Emerging Telecommunications Technologies, 2012, 23(4): 360-377.
- [13] SCHATZ R, HÖBFELD T, CASAS P. Passive youtube QoE monitoring for ISPs[C]//Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. IEEE, 2012: 358-364.
- [14] BURGER V, ZINNER T, DINH-XUAN L, et al. A generic approach to video buffer modeling using discrete-time analysis[J]. ACM Transactions on Multimedia Computing, Communications, and Applications, 2018, 14(2s): 33.
- [15] KRISHNAMOORTHY V, CARLSSON N, HALEPOVIC E, et al. BUFFEST: predicting buffer conditions and real-time requirements of HTTP (S) adaptive streaming clients[C]// The 8th ACM on Multimedia Systems Conference. ACM, 2017: 76-87.
- [16] TSILIMANTOS D, KARAGKIOULES T, VALENTIN S. Classifying flows and buffer state for YouTube's HTTP adaptive streaming service in mobile networks[C]// The 9th ACM Multimedia Systems Conference. ACM, 2018: 138-149.

[作者简介]



程光 (1973-), 男, 安徽黄山人, 博士, 东南大学教授、博士生导师, 主要研究方向为网络空间安全监测和防护、网络大数据分析。



房敏 (1989-), 男, 山东泰安人, 东南大学硕士生, 主要研究方向为网络测量和网络安全。



吴梓 (1973-), 女, 江苏南京人, 博士, 东南大学副教授, 主要研究方向为网络测量、网络安全和网络管理。