AI论文分析：Lightweight, general inference of streaming video quality from encrypted traffic

部门：数据通信AI使能技术部
作者：龚旭g00482242
日期：2019年7月21日

Security Level:

HUAWEI

目录

作者介绍

一作：Francesco Bronzino

Dec. 2018 – now: Research Scientist @ Network Protocol and Systems Research department, Bell Labs Paris-Saclay, France
Dec. 2016 – Nov. 2018: Post-Doctoral research fellow @ Inria, Paris, France
Sep. 2012 – Nov. 2016: PhD @ Rutgers University/The State University of New Jersey, Greater New York, USA
Oct. 2015 – Dec. 2015: Research Intern @ Ericsson, Stockholm, Sweden
May 2014 – Aug. 2014: Research Intern @ Huawei, California, USA

其他作者

- Sara Ayoubi, Renata Teixeira, Sarah Wassermann @ Inria, Paris
- Nick Feamster @ University of Chicago, Google citations: 17590, h-index: 64, i10-index: 174
- Paul Schmitt, Srikanth Sundaresan @ Princeton University

文章概述

ABSTRACT

Accurately monitoring application performance is becoming more important for Internet Service Providers (ISPs), as users increasingly expect their networks to consistently deliver acceptable application quality. At the same time, the rise of end-to-end encryption makes it difficult for network operators to determine video stream quality—including metrics such as startup delay, resolution, rebuffering, and resolution changes—directly from the traffic stream. This paper develops general methods to infer streaming video quality metrics from encrypted traffic using lightweight features. Our evaluation shows that our models are not only as accurate as previous approaches, but they also generalize across multiple popular video services, including Netflix, YouTube, Amazon Instant Video, and Twitch. The ability of our models to rely on lightweight features points to promising future possibilities for implementing such models at a variety of network locations along the end-to-end network path, from the edge to the core.

摘要

互联网用户日益期望网络能够持续提供可以接受的应用质量，精确度量网络应用的性能对运营商因此变得越来越重要。然而，作为网络流量大户的视频流，日趋端到端加密。这使得运营商难以基于传统的深度报文解析技术（DPI）从视频流直接度量视频用户侧的质量（KQI），包括起播时延、分辨率、卡顿和分辨率的切换。这篇文章提出了一套通用的、基于轻量特征的度量方法，以从加密视频流推断上述视频KQI指标。实验评估显示，该方法训练出的度量模型不但能达到传统方法的精度，而且能够在多个主流视频服务间保持较好的泛化能力，包括Netflix、YouTube、Amazon Instant Video和Twitch。这种基于轻量特征的模型能力进一步指明了将视频KQI度量模型部署在现网不同位置的可行性。

贡献点

- 视频KQI（Key Quality Indicator）度量方法跨视频应用
- 轻量特征
- 多指标度量

背景介绍

1 INTRODUCTION

Video streaming traffic is by far the dominant application traffic on today's Internet, with some projections stating that video streaming will comprise 82% of all Internet traffic in just three years [13]. Video content providers and Internet Service Providers (ISPs) both manage network routing and content delivery to ensure that users experience good video streaming performance; content providers use distributed content delivery [2] to serve video streams, adapting the video bitrate to changing conditions [19]. ISPs deploy proxies, transcode and compress video, and re-route traffic to deliver good video quality with the existing network resources.

Optimizing video delivery depends on the ability to determine the quality of the video stream that a user receives. In contrast to video content providers, who have direct access to video quality from client software, network operators such as ISPs must typically infer video quality from traffic as it passes through the network. End-to-end encryption, which is becoming increasingly common with video streaming over HTTPS and QUIC [28, 32], prevents an ISP from directly observing precise application metrics such as startup delay, rebuffering events, and video resolution from the video streaming protocol [5, 17]. Without access to such metrics, ISPs cannot determine the quality of video sessions or the underlying causes of quality degradation; therefore, they cannot detect when the network may be inducing poor application performance or perform steps to improve the delivery of the application to users.

关键点

- 未来三年，视频流量将占据82%的互联网流量
- 视频内容供应商：使用分布式方法来供应视频流，使视频码率实时匹配变化的网络环境和客户端buffer
- 运营商：通过部署网关、转码和压缩视频、路由选路来确保利用现有网络资源提供好的视频质量
- 视频质量在网络侧可视是视频质量调优的前提
- 视频流日趋端到端加密，如HTTPS和QUIC --> 问题：运营商不能看到应用层指标，进而很难独立视频流的用户侧质量（KQI）、不能判定网络在什么时候会引起质差、不能基于质差调优

互联网视频流基础技术

2.1 DASH Protocol

Internet video streaming services typically use Dynamic Adaptive Streaming over HTTP (DASH) [35] to deliver a video stream. DASH divides each video into time slices known as *segments* or chunks (of possibly equal duration), which are then encoded at multiple bitrates and resolutions. These segments are typically stored on multiple web servers so that a client can download segments from a nearby server.

At the beginning of a video session, the client downloads a DASH Media Presentation Description (MPD) file from the server. The MPD file contains all of the information the client needs to retrieve the video (*i.e.*, the audio and video segment file information). The client sequentially issues HTTP requests to retrieve segments at a particular bitrate. An *application-layer Adaptive Bitrate (ABR) algorithm* determines the quality of the next segment that the client should request. ABR algorithms are proprietary, but most video services rely on recently experienced bandwidth [36], current buffer size [19], or a hybrid of the two [42] to guide the selection. The downloaded video segments are stored in a client-side application buffer. The video buffer is meant to ensure continuous playback during a video session. Once the size of the buffer exceeds a predefined threshold, the video starts playing. A video session can typically be classified into two distinct phases: (1) the buffering phase, where the client fetches video segments as quickly as possible to fill the buffer; and (2) the steady-state phase, where a client downloads new video segments at roughly the same rate as playback to maintain a stable buffer level.

关键协议

- DASH（Dynamic Adaptive Streaming over HTTP）协议
  - 每个视频会话切分成一定时长的片段，每个片段用不同码率和分辨率编码，被存储在多个服务器上
  - 用户通过串行的HTTP请求，从距离最近的一个服务器上下载特定码率的视频分片
  - ABR（application-layer Adaptive Bitrate）算法动态决定用户需要请求的下一个分片的质量，定制化，基于网络带宽、buffer水线、或两者
  - 一个视频会话分为两个阶段：（1）缓冲阶段，（2）稳态阶段

互联网视频流基础技术

2.2 Streaming Video Quality Metrics

Startup delay. Startup delay is the time elapsed from the moment the player initiates a connection to a video server to the time it starts rendering video frames. Prior work has shown that higher startup delays cause more users to abandon video sessions [23]. High startup delays also reduce the total time that users spend on a video service [15].

Video bitrate. The bitrate of video segments is one metric of the quality of the displayed content. The average bitrate over a video session (computed as the average of the bitrates played weighted by the duration each bitrate is played) is the key quality metric for video content [15]. The relationship between bitrate and quality is complex because the bitrate depends on the resolution, the encoding, and the content type [1].

Bitrate switches. Video players switch the bitrate to adapt to changes in network conditions with the goal to select the best possible bitrate for any given condition. Nevertheless, bitrate switching has a negative effect on quality of experience [6, 18]. Both the frequency of bitrate switches (or switches per second) and the amplitude of switches—the difference between the bitrate before and after the change—affect quality of experience [18].

Rebuffering events. This metric captures the periods of time video playback stalls because the buffer is empty. We can measure the impact of rebuffering with two metrics: the rebuffering time, which is the total time spent rebuffering in a session, or the rate of rebuffering, which is the frequency of rebuffering events. High rebuffering rate and time increase the rate of abandonment and reduce the likelihood that a user will return to the service [15, 23].

关键指标

- 视频体验和许多因素相关，包括用户的期望、观看背景、网络性能、设备性能等
- 赋能运营商网络运维的视频质量指标
  - 起播时延
  - 码率：均值
  - 码率切换：频率、幅度
  - 卡顿事件：卡顿时长，卡顿率

业界视频QoE度量方法

业界方法缺陷

- 不具备跨视频应用的能力：业界方法主要集中研究YouTube，由于不同视频应用通常使用定制化的视频推流算法、并有不同的视频内容特征，经作者实验证实，这些差异影响了YouTube质差度量模型的跨应用泛化
- 不能做在线处理：业界方法要么使用DPI，要么设计状态丰富的特征，导致计算复杂、消耗较多算力
- 较少做精细化KQI度量：基于整个视频会话，粗略地判断该会话是好还是坏，不能在更短时长范围内和更多KQI指标上进行推理度量

业界方法

- [14][26][22][25]

  - Dimopoulos et al. [14] developed a method that runs on a web proxy in the network to identify YouTube sessions over TCP, collects traffic statistics and HTTP requests to infer rebuffering, bitrate switches, and the average quality of the entire video session.
  - Mazhar and Shafiq [26] infer video bitrate, startup delay, and rebuffering events for YouTube (over both TCP and QUIC) using network and transport layer statistics collected per ten-second time slots.
  - BUFFEST [22] identifies individual video segments within video traffic to then infer YouTube rebuffering events using a buffer-state emulator.
  - eMIMIC [25] relies on the same method to identify video segments and then models the video session as a sequence of segments to infer average bitrate, bitrate switches, rebuffering ratio, and startup delay of a video session. eMIMIC's model must be parameterized for each video service.

YouTube模型直接度量其它视频应用的起播时延

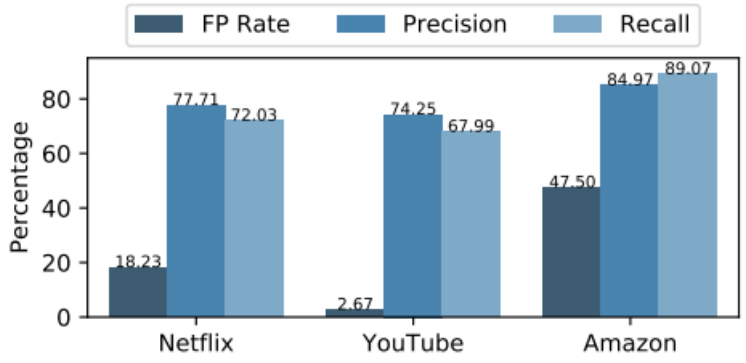


Figure 1: Using a per-service model to detect high (bigger than 5) startup delay using method from Mazhar and Shafiq [26].

[26]方法跨应用的性能不稳定

目录

问题描述-输出：视频流KQI

起播时延

- 分类问题：二分类[26]不利于跨应用泛化
- 回归问题：达到更精细的度量粒度，有利于跨应用泛化 --> 本文使用

分辨率

- 业界通常使用平均分辨率
  - 基于视频会话
  - 基于时间窗口 --> 本文使用，10秒切片，对每个分片的分辨率进行分类识别（240p，360p，480p，720p，1080p）
- 其它方法
  - 二分类（好/坏）、三分类（高/标准/低） -->此类方法容易误导，不利用各类终端通用，如480p手机 vs. 4K 电视
  - 平均码率 --> KQI与QoE映射关系不清晰，因为视频质量与码率、编码、内容均有关系

分辨率切换和卡顿

- 分类问题：二分类（是否发生），三分类（无/轻微/严重）
- 回归问题：事件发生的时间比例
  - 基于视频会话
  - 基于时间窗口 --> 本文使用

问题描述-输入：轻量特征

轻量特征设计的思想

- 在特征数据采集的工作量、处理复杂度、检测精度上做最优折中

网络层

- 只依靠四元组信息识别网络流量、并从中提取特征，比如流量吞吐、包/字节数、包到达时间等 --> 多数路由都能够采集到这类信息

传输层

- 特征包括端到端时延、包重传等，表征可能出现的网络问题
- 但是，这些特征有两方面先天缺陷
  - 由于流量端到端加密，只能从非加密TCP流中提取特征，QUIC一类的加密流量则不能提取特征
  - 很多特征需要长时间维持每条流的状态，不利用方案的规模化

应用层

- 应用层的特征可以提供最大化的视频会话的性能信息，但是端到端加密让DPI失效
- 本文采用BUFFEST[22]方法，来从加密流中提取一些应用层的信息，如视频流分片，作者验证了此方法能够同时适用于TCP和QUIC视频流量，并且轻量

问题描述-目标：通用预测模型

目标

- 跨应用通用模型，而不是基于应用的独立模型

基于应用的独立模型

- 缺点
  - 一个应用的模型直接度量其它应用的性能不佳
  - 需要运营商基于每个应用采集现网数据、调试每个模型、并基于每种视频的播放逻辑的更新来调整对应模型，成本过高

通用模型

- 弥补独立模型的缺陷：使能运营商周期性地训练一个通用模型
- 建立通用模型的可能性：多数视频服务基于DASH算法，流量具有不同但是相似的动态特性
- 通用模型性能也许不及独立模型，但只要保持达标的性能，就能有效弥补独立模型的缺陷

目录

方法：视频流量产生、标签数据采集

目的

- 构造视频会话的网络流量标签数据集

方法

- 11个便携，5种场景，采集现网数据
- 采集时长：连续7个月
- 播测4种主流视频的5种会话，包括Netflix、YouTube TCP、YouTube QUIC、Twitch、Amazon Instant Video
- 每个视频会话播测8-12分钟，具体时长取决于是否有广告
- 对于长视频会话，随机变更播放起始点
- 尽量播测不同类型的视频内容
- 基于Chrome WebRequest APIs和Chrome browser API来获取视频质量的时间序列信息

We instrumented 11 machines to generate video traffic and collect packet traces together with the data from the Chrome extension: six laptops in residences connected to the home WiFi network (three homes in a European metropolitan area with download speeds of 100 Mbps, 18 Mbps, and 6 Mbps, respectively; one room in an European student residence; one apartment in a campus town in the US; and one home from a rural area in the US), four laptops located in our lab connected via the local WiFi network, and one desktop connected via Ethernet to our lab network.

In the lab environment, we manually varied the network conditions in the experiments using `tc` [9] to ensure that our datasets capture a wide range of network conditions. These conditions can either be stable for the entire video session or vary at random time intervals. We varied capacity from 50kbps to 30mbps, and introduce loss rates between 0% and 1% and additional latency between 0 ms and 30 ms. All experiments within homes ran with no other modifications of network conditions, to emulate realistic home network conditions.

采集到的数据

- 11767条视频会话数据

Service	Total Runs	% Home	% Lab
Netflix	3,498	59	41
YouTube TCP	4,232	18	72
YouTube QUIC	1,101	68	32
Twitch	2,231	17	83
Amazon	1,852	10	90

Table 1: Summary of the dataset.

方法：特征设计

Network Layer		Transport Layer		Application Layer	
throughput up/down (total, video, non-video)	○	# flags up/down (ack/syn/rst/push/urgent)	○	segment sizes (all previous, last-10, cumulative)	◐
throughput down difference	○	receive window size up/down	○	segment requests inter arrivals	◐
packet count up/down	○	idle time up/down	○	segment completions inter arrivals	◐
byte count up/down	○	goodput up/down	○	# of pending request	◐
packet inter arrivals up/down	○	bytes per packet up/down	○	# of downloaded segments	◐
# of parallel flows	○	round trip time	●	# of requested segments	◐
		bytes in flight up/down	●		
		# retransmissions up/down	●		
		# packets out of order up/down	●		

Table 2: Summary of the extracted features from traffic. Amount of state required: Simple state (○), Simple application-layer state (◐), Complex state (●).

TCP独有特征

特征设计的轻量化思路

- 网络层：只需硬件设备使用一个寄存器来使能一个特征的计算
- 传输层
  - 分上下行流量，从TCP/UDP报文头提取概括统计特征
  - 复杂度取决于是否需要整条流的状态
- 应用层
  - 基于推断的视频分片，进行特征设计，如上表
  - 对上表中所有特征进行统计特性计算，得出细分特征，如最小、最大、均值、百分位数、标准差

方法：模型训练

算法选型

- 从Adaboost、logistic regression、decision trees、random forest中优选出random forest

训练与测试数据集

- 6种特征数据集：Net，Tran，App，Net+Tran，Net+App，All
- 5种视频会话数据集：Netflix, YouTube, Amazon, Twitch, 4种应用混合
- 每个KQI指标训练30种模型： 6种特征数据集 × 5种视频会话数据集
- 注：随后的实验没有展示基于Tran和App的性能测试，因为对应模型性能较差、且数据采集难度比Net大

目录

性能评估：起播时延度量

总体性能：好

性能分析

- 图1、图2
  - 基于Net+App训练的模型性能最优，但Net性能也足够好

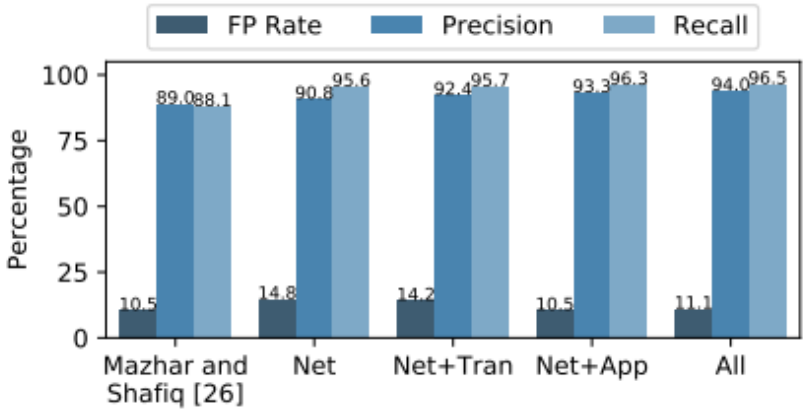


图1 起播时延度量模型在不同特征集上的性能

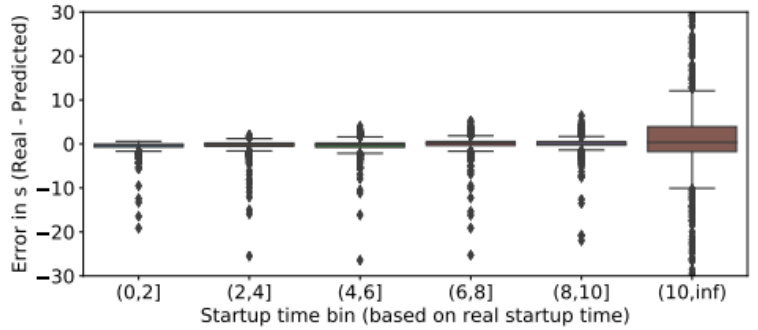


Figure 4: Relative error in startup delay inference (by levels) using Net+App features cut at [-30, 30] on the y axis (For all four video services).

图2 起播时延度量误差测试

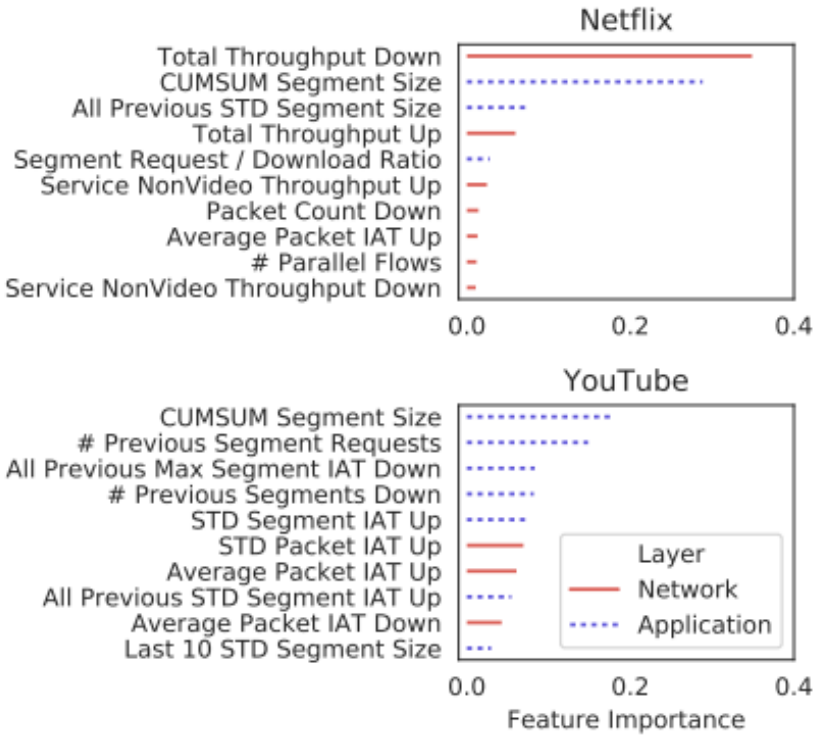


图3 起播时延度量重要特征

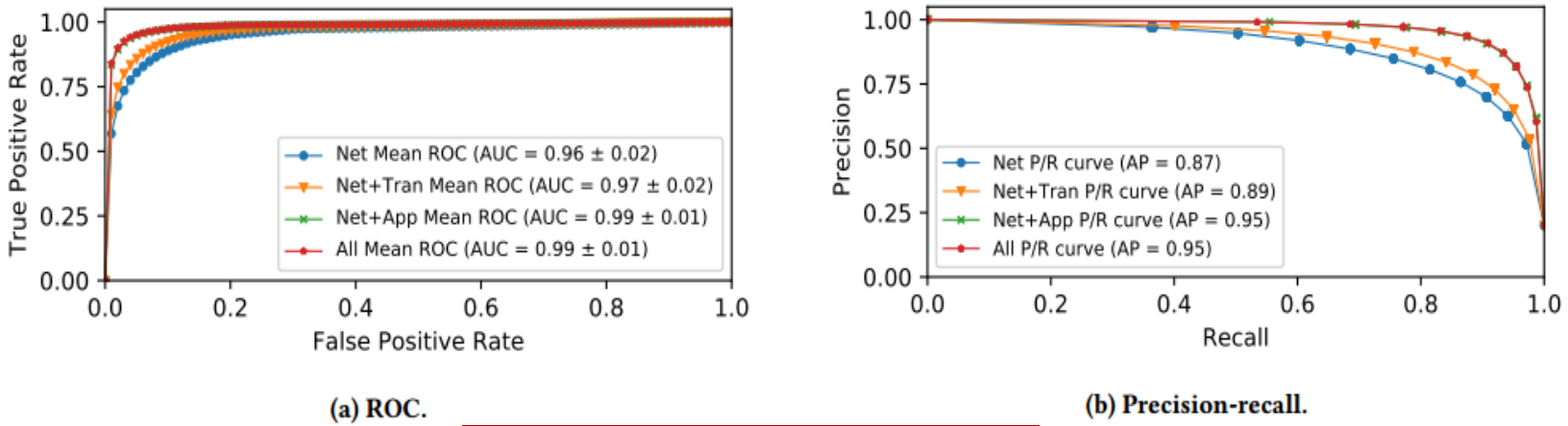
- 图3
  - YouTube、Amazon和Twitch具有相似的特征重要性排序，应用层特征占主导地位，这些重要特征反应客户端下载视频分片的速率
  - Netflix具有不同于其它三种视频的特征重要性排序，这些重要特征反应下行流量大小
  - 这四种视频的重要特征都反应客户端视频起播的速度，也和DASH视频流行为相符：客户端buffer达到一定门限才开始播放视频，若越快达到门限，则起播时延越低

性能评估：分辨率度量

总体性能：好

性能分析

- 图1、图2
  - 包含App特征的模型能达到最好性能：91%准确率和召回率、4%误报率
  - 无App特征的模型至少降低8%准确率、并双倍升高误报率
  - 83.9%的误分类只与真实分辨率偏差1个档位
- 图3
  - 4种视频类型的重要特征多数与分片大小有关-->高分辨率意味着每英寸有更多像素，因此相同的视频分片需要更多数据来表达



(a) ROC. (b) Precision-recall.

图1 不同特征集、混合视频会话集下的分辨率度量性能

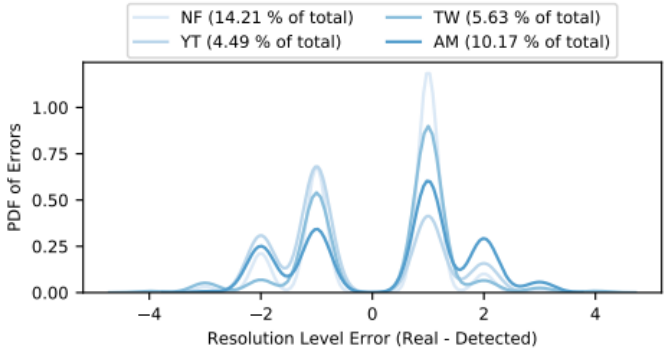


图2 度量误差评估

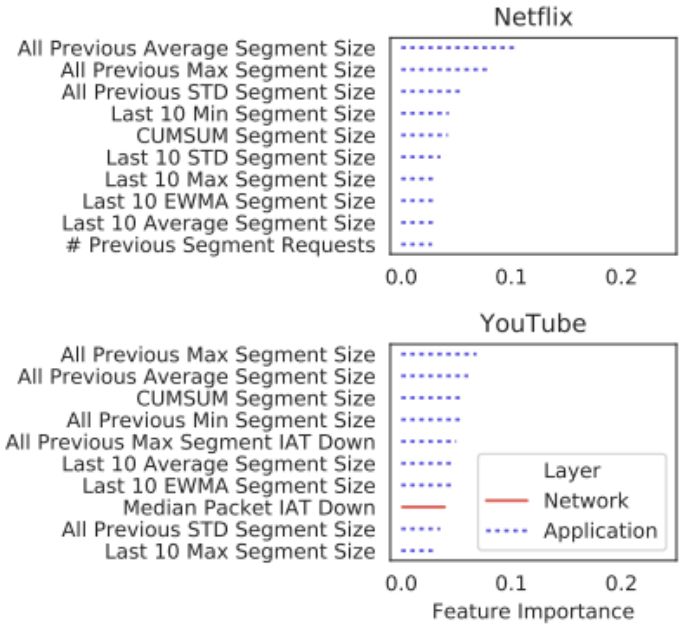


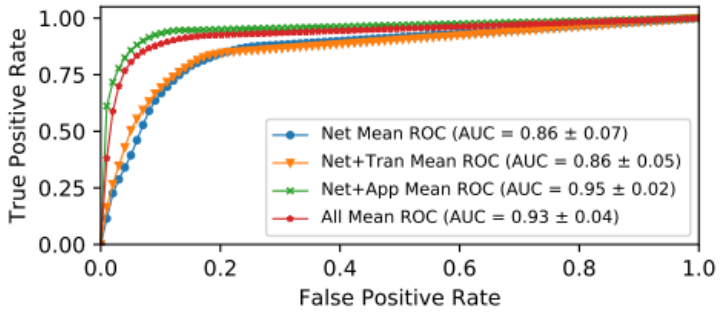
图3 分辨率度量重要特征

性能评估：卡顿度量

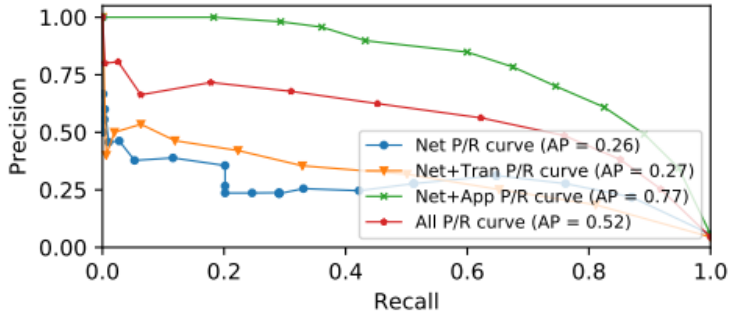
总体性能：好坏视需求而有所波动

性能分析

- Amazon、Twitch和Netflix似乎使用更大的客户端应用buffer来避免卡顿，因而只有YouTube视频数据有卡顿样本
- 图1
  - Net+App特征集性能最佳：YouTube的卡顿度量准确率84.9%、召回率60%，有待提高
  - 26%和41.6%漏报的时间与真实卡顿事件的时间分别相差在10sec和30sec以内，对这些误差的容忍将把召回率分别提升至78.9%和91.9%
- 图2
  - 最重要的特征与视频分片大小、分片到达间隔时间及下载总数有关
  - 这些特征刻画了客户端buffer的消耗行为，当到达时间间隔增加、下行分片的数量和大小减少时，客户端buffer的消耗速率快于填充速率、进而引起卡顿



(a) ROC.



(b) Precision-recall.

图1 YouTube卡顿度量性能测试

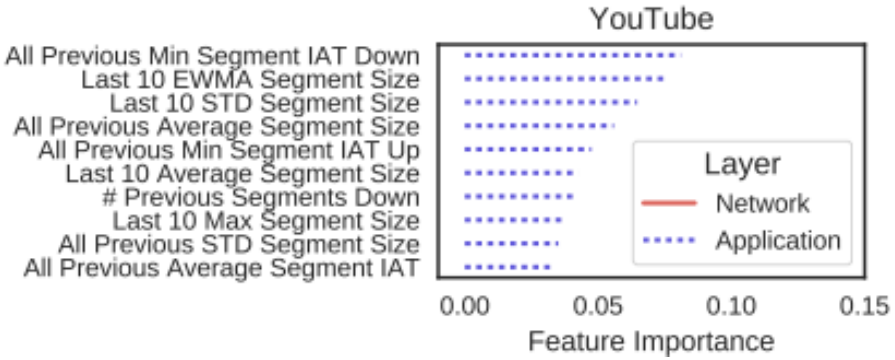


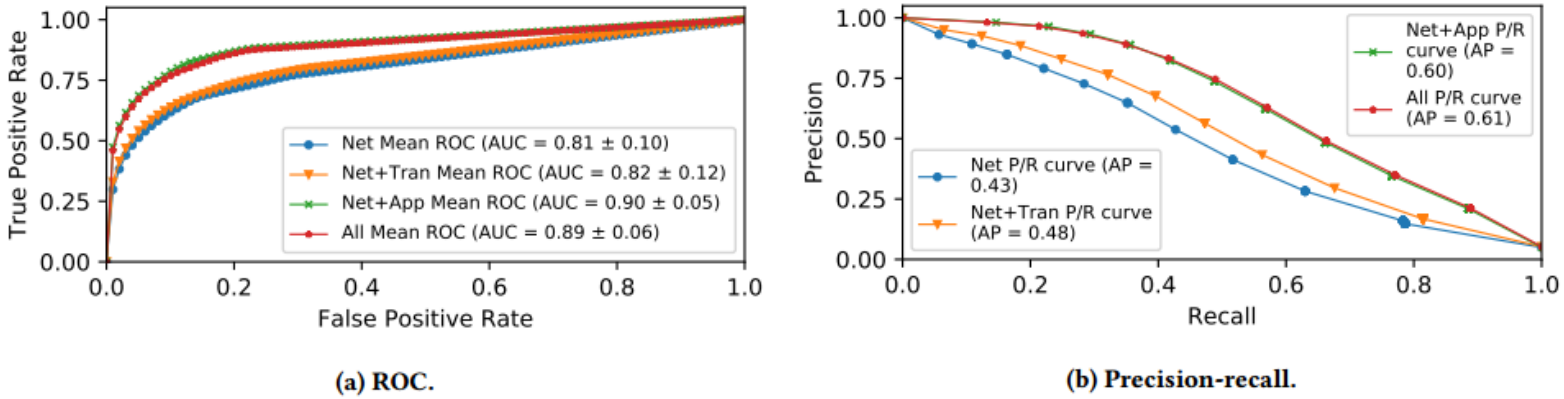
图2 卡顿度量模型的重要特征

性能评估：分辨率切换度量

总体性能：不好

性能分析

- 图1：准确率最高只有61%
- 图2：没有明显重要的特征反应分辨率的切换
- 图1和图2所展示的现象应该与ABR算法的复杂性有关，取决于多个因素，包括吞吐率的估算、buffer的占有率、参数的多样性和网络环境发生变化引起的多变性等
- 结论：相对其它视频QoE指标，也许更适合使用其它方法来度量分辨率的切换



(a) ROC. (b) Precision-recall.

图1 所有视频流的分辨率切换度量性能测试

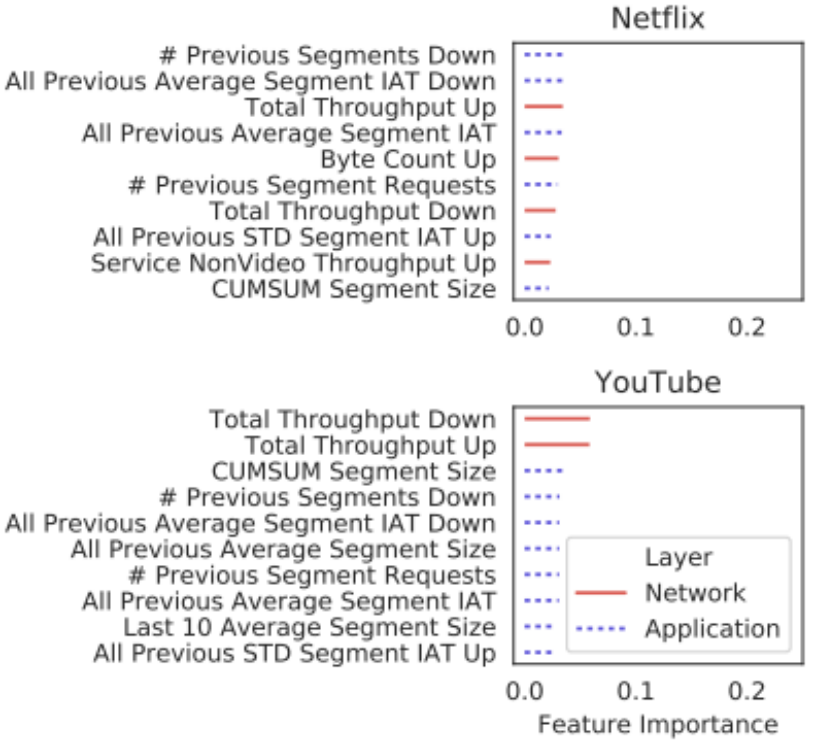


图2 分辨率切换度量重要特征

性能评估：跨应用泛化能力

评估结论：可以使用跨应用通用模型来度量起播时延、分辨率、分辨率切换（卡顿由于数据量少，不包含在此结论中）

性能分析

- 起播时延
  - 图1：通用模型（Combined-NF）和独立模型（Same-NF）性能相似，且在[2, 10]均能将度量误差保持在1sec以内
  - 通用模型重要特征：cumulative segment sizes和size distribution
- 分辨率
  - 图2：通用模型几乎可以达到独立模型的性能
  - 通用模型重要特征：4个与segment size相关的特征
- 分辨率切换
  - 图3：通用模型与独立模型的性能相差在2.5%以内

图1 通用模型和独立模型的起播时延度量性能对比

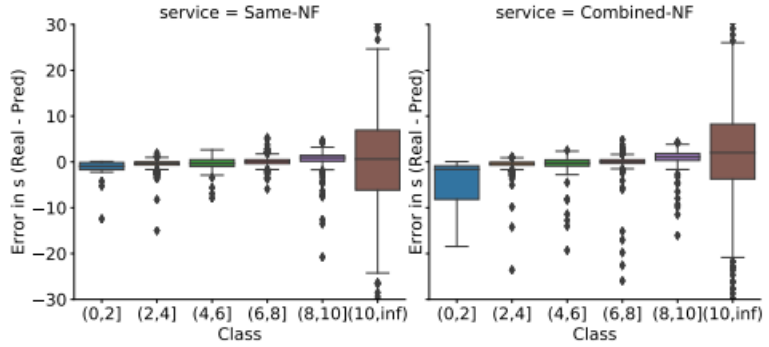
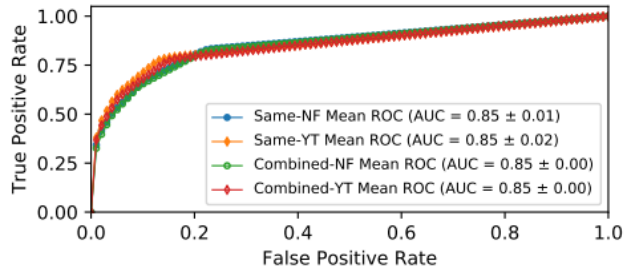
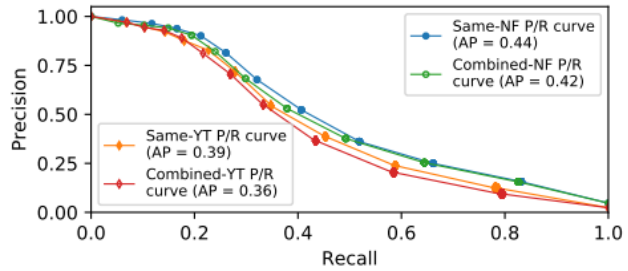


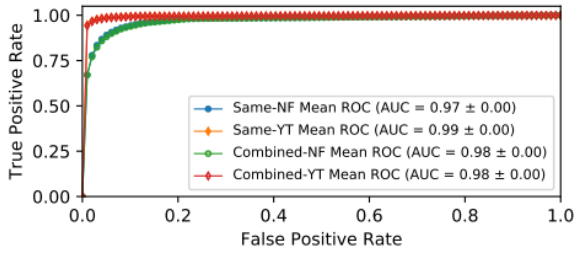
图3 通用模型和独立模型的分辨率切换度量性能对比



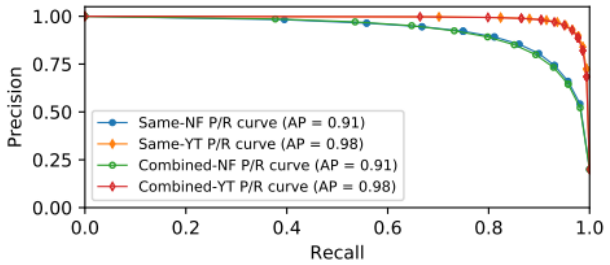
(a) ROC.



(b) Precision-recall.



(a) ROC.



(b) Precision-recall.

图2 通用模型和独立模型的分辨率度量性能对比

本文最新版本下载：https://arxiv.org/abs/1901.05800

Lightweight, General Inference of Streaming Video Quality from Encrypted Traffic

Francesco Bronzino[†], Paul Schmitt[°], Sara Ayoubi[†], Nick Feamster[°],
Renata Teixeira[†], Sarah Wassermann[†], and Srikanth Sundaresan[°]
[†]Inria, Paris [°]Princeton University

ABSTRACT

Accurately monitoring application performance is becoming more important for Internet Service Providers (ISPs), as users increasingly expect their networks to consistently deliver acceptable application quality. At the same time, the rise of end-to-end encryption makes it difficult for network operators to determine video stream quality—including metrics such as startup delay, resolution, rebuffering, and resolution changes—directly from the traffic stream. This paper develops general methods to infer streaming video quality metrics from encrypted traffic using lightweight features. Our evaluation shows that our models are not only as accurate as previous approaches, but they also generalize across multiple popular video services, including Netflix, YouTube, Amazon Instant Video, and Twitch. The ability of our models to rely on lightweight features points to promising future possibilities for implementing such models at a variety of network locations along the end-to-end network path, from the edge to the core.

video streaming algorithms and content characteristics can vary significantly across video services (*e.g.*, buffer-based [19] versus throughput-based [36] rate adaption algorithm or fixed-size [19] versus variable-size video segments [27]). Thus, a model that is tuned for YouTube is unlikely to work more generally across the growing set of video streaming applications. We find, in fact, that these previous methods do not work well for other video streaming services, and thus that more general models are needed.

Second, existing models cannot perform on-path processing; these existing approaches are typically computationally complex, and they often require either deep packet inspection or the computation of stateful features that can be difficult to extract and track for a large number of flows. Third, existing work predicts the quality of experience at a coarse granularity, such as whether quality is "good" or "bad", as opposed to more precise metrics such as resolution or startup delay. Finally, most previous approaches typically classify application granularity for an entire session and do not track these metrics over shorter time intervals within a single session.

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home and organization for a fully connected, intelligent world.

HUAWEI