IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ENERGY MANAGEMENT IN BUILDINGS

Received January 31, 2018, accepted February 27, 2018, date of publication March 9, 2018, date of current version May 24, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2811416

QoE Assessment of Encrypted YouTube Adaptive Streaming for Energy Saving in Smart Cities

WUBIN PAN AND GUANG CHENG, (Senior Member, IEEE)
School of Cybersecurity, Southeast University, Nanjing 210096, China
School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
Key Laboratory of Computer Network and Information Integration, Ministry of Education, Nanjing 210096, China

Corresponding author: Wubin Pan (wbpan@njnet.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities, in part by the National High Technology Research and Development Program (863 Program) of China under Grant 2015AA015603, in part by Jiangsu Future Networks Innovation Institute: Prospective Research Project on Future Networks under Grant BY2013095-5-03, and in part by the Six talent peaks of high level Talents Project of Jiangsu province under Grant 2011-DZ024.

ABSTRACT Video streaming has become one of the most prevalent mobile applications and uses a substantial portion of the traffic on mobile networks today. With the limited bandwidth of mobile networks, understanding the user perception of the quality (i.e., Quality of Experience or QoE) of video streaming services is thus paramount for content providers and content-delivery network providers to flexibly configure network bandwidth, video servers, routing devices, and other network resources to save energy in smart cities. Although various video QoE assessment approaches have been proposed using different key performance indicators (KPIs), they all essentially relate to a common parameter: bitrate. However, because YouTube has adopted hyper text transfer protocol over secure socket layer (HTTPS) as its adaptive video streaming method to better protect user privacy and network security, bitrate can no longer be obtained from encrypted video traffic via typical deep packet inspection. In this paper, we address this challenge by proposing a machine-learning-based bitrate estimation (MBE) approach to parse bitrate information from IP packet level measurements. First, we filter HTTPS YouTube traffic based on the previously established video server IP according to the data packet googlevideo field. Then, we identify the transmission mode according to the traffic characteristics of several previous packets. Next, we identify the bitrates and resolutions of HTTP Live Streaming and Dynamic Adaptive Streaming over HTTP modes according to the characteristics of video chunks. Finally, for evaluating the effectiveness of MBE, we have chosen the video Mean Opinion Score (vMOS) proposed by a leading telecom vendor as the QoE assessment framework, and have conducted comprehensive experiments to study the impact of bitrate estimation accuracy on its KPIs for the HTTPS YouTube video streaming service. Experimental results show that MBE is a feasible and highly effective QoE evaluation approach to flexibly configure network resources in smart cities.

INDEX TERMS Hyper text transfer protocol over secure socket layer (HTTPS) YouTube, QoE assessment, adaptive streaming, machine learning, smart city.

I. INTRODUCTION

Network as an essential resource in life, excessive deployment of network resources may waste energy, and too low deployment may affect QoS in smart cities. Currently, video traffic occupies a large amount of network resources. Therefore, the video QoE can be used to evaluate the network status effectively. Network providers and content providers can flexibly configure network resources such as network bandwidth, video servers, and routing devices to achieve energy saving in smart cities [1], [2].

Though mobile network technology has seen continuous development, the increasing processing power of mobile devices and the quality upgrades of YouTube videos from 720P/1080P to 2K and up to 4K/8K, presents the network with a substantial challenge. The current limited mobile network bandwidth needs to carry large amounts of video data. Video service providers and network service providers need to collaborate to improve network utilization and transmission efficiency. To ensure the quality of the user's video experience, YouTube has adjusted its video transmission mode and coding. In addition, ISP service providers perform monitoring and assessment for the video quality of experience, and dynamically adjust network parameters to save energy based on the assessment results in smart cities.

Recently, hyper text transfer protocol over secure socket layer (HTTPS) was adopted by major video content providers including YouTube and Netflix to provide video services to mobile users [3] with better protection of user privacy. Adaptive streaming is also commonly used as an effective means to enhance user QoE by dynamically adjusting the bitrate to adapt to current network conditions.

Previous video quality assessment methods compute the bitrate from the video bytes and playback duration based on deep packet inspection or the YouTube API. In an unencrypted scenario, a DPI-based method can acquire these parameters to accurately assess the video source quality, initial buffering latency, stalling and other QoE metrics. However, these parameters cannot be obtained from encrypted traffic. Therefore, encrypted traffic requires a deep flow inspection (DFI) technology.

The problem most concerning ISPs is the huge amount of encrypted traffic of YouTube videos. On the one hand, heavy traffic should not affect other network users by causing network congestion. On the other hand, YouTube video users should have good video viewing experiences. Therefore, the focus of encrypted video traffic research is how to get a good user viewing experience from encrypted traffic in smart cities.

QoE assessment in HTTP video streaming is a heavily investigated topic. Most research has focused on identifying the most relevant Key Performance Indicators (KPIs) and studying their impact on user-perceived video quality.

Video quality-of-experience assessment includes both objective and subjective assessment. Due to its feasibility and cost effectiveness, objective video QoE assessment is commonly used to estimate the user perception of the quality of video streaming services [4], [5]. For objective assessments, active measurements can be conducted from a client to probe and evaluate the network. However, active probing can provide only instant samples and cannot accurately determine network conditions over an entire period of video streaming service. In contrast, passive measurements can be performed either at the client device or in-network. However, client-side measurements (e.g., YoMoApp [6]–[8] and YouSlow [9]) are more intrusive; end users are directly involved, and can provide accurate views on several objective key performance indicators (KPIs) from an individual's perspective. Those KPIs provide more detailed information about the perceived video service quality and can jointly contribute to QoE estimation under different QoE assessment frameworks [10], [11].

However, client-side measurements need the customers' cooperation to install an assessment application. Also, they cannot demonstrate the impact on QoE caused by specific network links. In-network measurements (e.g., YOUQMON [12,] [13]) have better coverage of end users. However, the resulting QoE must be estimated from traffic characteristics, typically via deep-packet inspection, which becomes infeasible when HTTP video streaming is replaced by HTTPS.

Although it is clear that all these KPI factors indeed have an impact on a resulting QoE assessment, there have been very few QoE assessment frameworks proposed to systematically combine multiple KPIs into a resulting QoE.

Recent literatures [14]–[16] have shown that stalls (i.e., stopping of the video playback) and initial delays are the most relevant KPIs for QoE in HTTP video streaming. In the adaptive streaming case, literatures [17]–[19] have shown that increasing or decreasing the video quality during the playback have an important impact on QoE. We hypothesize that there is a learnable relationship between these KPIs and video QoE [4]. Literatures [20], [21] propose a function F: KPIs $\rightarrow$ QoE. With such a function, a video content provider can estimate the video QoE over all its subscribers simply by monitoring the selected KPIs.

Therefore, we have cooperated with a human-factors engineering lab, which maps subjective feelings to objective KPIs to establish a video MOS (vMOS) evaluation framework to synthetically assess a video QoE [4], which is composed of video source quality, initial buffering latency, and stall ratio. Hence we can directly compare the user's subjective feelings during video viewing with scores of KPIs, and objectively derive the video QoE from the quality of the video source and the network transmission conditions.

We first propose a method for identifying the transmission modes based on statistical features. Based on the analysis of the HLS (HTTP Live Streaming) and DASH (Dynamic Adaptive Streaming over HTTP) adaptive streaming modes, and in accordance with a model built on statistical features of the video chunk to identify the traffic, a video chunk feature is extracted from the network layer without parsing the application layer to obtain the plaintext data. We then propose a machine-learning-based bitrate estimation (MBE) method using only the traffic characteristics from the network layer. A decision tree is applied as a quick base classifier to identify the bitrate of video chunks. We have also extensively studied the impact of deviations of the bitrate estimation on the KPI parameters used in the vMOS, a systematic objective of a QoE assessment framework, so as to solve the problem of QoE assessment from the intermediate nodes for users watching YouTube. Experimental results show that the method has better performance on bitrate and resolution identification, which can be effectively applied to the encrypted video QoE evaluation to flexibly configure the network parameters in smart cities. The contributions of our work can be summarized as follows:

(1) We propose an MBE method to compute the QoE-relevant KPIs for HTTPS YouTube network video, relying only on traffic characteristics of video chunks from the network layer without using DPI technology. In addition, we introduce four distinguishing features for identifying the transmission mode; the method can identify the encrypted video transmission mode from the first several packets of the flow.

(2) We have extensively analysed the impact of deviations from estimated bitrates on the KPIs and vMOS scores. Experimental results show that the impact of bitrate deviations on QoE assessment using MBE is negligible.
(3) Our assessment system can monitor and evaluate the QoE of each video chunk, rather than requiring a holistic assessment of the whole video. Moreover, several QoE assessment points in the network can cooperate to locate network failures.

The remainder of the paper is structured as follows. Section II summarizes related work on QoE for HTTP video streaming to mobile users. Section III presents the design of MBE, the ML-based bitrate estimation method. Section IV presents the experimental data set, a brief description of the experimental environment, and the evaluation results of MBE. A conclusion is drawn in Section V.

II. RELATED WORK

QoE in HTTP video streaming is a well-known and heavily investigated topic. Literatures [14], [15] show that stalling events and initial buffering latency are the most QoE-relevant KPIs for HTTP video streaming; Hoßfeld *et al.* [16] show that initial buffering latency is preferred to stalling by approximately 90% of users. Nam *et al.* [22] study YouTube and Netflix video traffic over mobile networks and found that network bandwidth and CPU computing power are important factors affecting video QoE. YouTube video QoE was evaluated by packet loss rates for different devices and network conditions. Qi and Dai [23] show that increased stalling duration decreases the users' QoE, and that one long stall is preferred to frequent short ones. Similar to us, Mok *et al.* [24] propose KPIs of initial buffering latency, mean buffering duration and buffering frequency to establish a utility function for HTTP video streaming. Literatures [18], [19] show that adaptive streaming has a large influence on QoE; adaptive streaming can dramatically reduce stalling events when the network performance decreases [17]. Particularly when the network performance is poor, adaptive streaming is clearly better than the HTTP progressive download (HPD) transmission mode. Seufert *et al.* [25] compare the QoE of classical and adaptive streaming; they show that adaptive streaming is excellent during the poorest network performance. Similarly, Zinner *et al.* [26] show that the impacts on controlled video quality is preferred to uncontrolled impacts like stalling events. Seufert *et al.* [10] describes the QoE of adaptive streaming in detail.

To measure the video QoE, Aggarwal *et al.* [4] introduces an approach they call Prometheus to estimate the QoE of mobile applications based on a combination of passive in-network measurements and client-side measurements; machine learning (ML) techniques were adopted to map the video QoS to QoE. Wamser *et al.* [13] present a real-time QoE assessment framework called QoM for video services. The influence factors include KPIs of QoE, network conditions and application-level parameters. Nam *et al.*[9] presents a web browser plug-in named YouSlow to detect real-time stalling events. Similarly, literatures [6], [7] present a client-side application named YoMoApp to monitor YouTube video streaming on Android devices. It monitors KPIs of QoE via the YouTube API. Gómez *et al.* [27] present an application to measure objective parameters of QoS, and then map the QoS to subjective QoE through a utility function. However, these client-side approaches require installing an application on mobile devices and reporting measurements to a server, which are annoying to users. Similar to our own work, Casa *et al.* [12] presents an in-network measurement approach to measure the QoE of YouTube, relying on network-layer passive probing only. Hoßfeld *et al.* [14] monitor application-level stalling events for the YouTube QoE from a mobile ISP's perspective, relying on network-level measurements only.

Research on identifying encrypted traffic is based mainly on ML approaches [28]–[31] and traffic behaviour [32], [33]. The machine learning method mainly establishes models based on flow statistics, such as flow durations [34], [35], the numbers of packets [36], the minimum, maximum, mean and variance of inter-arrival times [37], payload sizes [37], [38], bit rates [38], [39], round trip times [39], and packet directions or bit rates sent from server [40]. Then, most of the features do not apply to the QoE parameter identification of the video stream. TCP parameters (such as bit rate sent from server, arrival time interval, round trip time, and packet direction) are weakly correlated features for identifying video QoE parameters. Korczynski and Duda [41] propose an identification method for application flows conveyed in SSL/TLS sessions based on stochastic fingerprints. The fingerprints are observed from training application flows based on first-order homogeneous Markov chains. Khakpour and Liu [42] propose a real-time identification framework called Iustitia to identify the nature of flows for the first time. The basic idea of Iustitia is based on the different values of entropy for text flows, binary flows and encrypted flows to classify flows using machine learning techniques. Distinct from the work above, the identification object for encrypted YouTube QoE assessment is finer-grained video chunks, not TCP flows. The identification class is the content type of YouTube traffic (e.g., video bitrate), not the class of traffic (e.g., YouTube, Netflix).

In this work, we first propose an ML-based QoE assessment for HTTPS YouTube video streaming on video chunks without a users' subjective ranking or installation of an assessment application. Then, we analyse the feasibility of the ML-based method on QoE assessment. Additionally, we map the KPIs of QoE to subjective feelings based on human-factors engineering to establish an objective assessment framework that contains the influence factors of video source quality, initial buffering latency and stalling ratio. Finally, our assessment can monitor and evaluate the QoE of each chunk rather than a holistic assessment of the whole video.

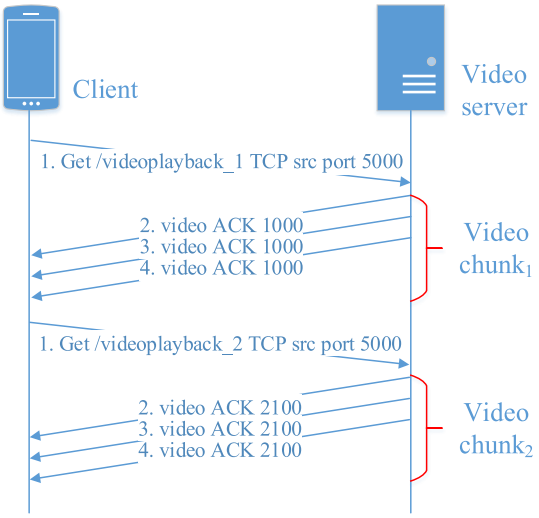


FIGURE 1. Video chunk transmission.

III. MACHINE-LEARNING-BASED BITRATE ESTIMATION

A. YOUTUBE TRAFFIC PRE-PROCESSING

1) OVERVIEW OF ADAPTIVE STREAMING MODE

The current YouTube client's adaptive streaming protocols are Apple HLS and MPEG DASH. HLS transmission mode is a manner of multiple request multiple downloads. The stream is divided into video and audio resources. Each video chunk is a separate unit to request, but the video and audio resources are not transmitted alternately. Most of the video chunks are equally divided in time with durations of approximately 5 seconds. There are also a few video chunks with durations of 2-10 seconds; each chunk corresponds to a URL. Before requesting the first video chunk, the chunk index file m3u8 must be requested first. Video transmission will begin a fast transfer of approximately 2 seconds of video sources to help reduce the initial buffering latency and stalling events. For the same video, different resolutions of video chunk vary greatly in their amount of data; audio chunks all have substantially the same amount of data, and the numbers of video and audio chunks are the same. Since the video is transferred in chunks, a video chunk (video and audio) is split into many TCP packets because the MTU is limited. The ACKs are the same for all TCP packets in the same video chunk except in abnormal situations. One ACK can be integrated to respond to a series of video chunks, as shown in FIG.1.

DASH transmission mode is different from HLS in that video and audio chunks are alternately requested in turn; most of the videos are divided into time durations of approximately 10 seconds. Video media are organized by Fragment MP4 (FMP4); FMP4 divides the media file into chunks, each of which can be separately decoded and played back.

2) VIDEO CHUNK PRE-PROCESSING

The video server and the client transmit not only the audio and video data but also the directory files and other interactive information; the volume of other information is far less than that of audio and video data. Therefore, a threshold L (the default is 20 KB) can be used to filter non-audio and video chunks. When distinguishing between audio and video chunks, the audio bitrate is relatively fixed compared with the video bitrate, and remains unchanged during playback. The volume of audio chunk data with different resolution is concentrated in a fixed interval.

Under ideal network conditions, abnormal audio and video chunks are transmitted in the same TCP stream, and all TCP packets in the chunk are ACKed at once. However, due to the uncontrollability of the network, interruptions and retransmissions of the TCP stream occur, which directly leads to the audio and video chunks being transmitted through more than one TCP stream. Through the use of Fiddler man-in-the-middle attacks, every HTTPS connection carrying encrypted audio and video data is reported with an SSL Alert message when aborting the transmission. The last audio and video stream transmitted by the SSL Alert message is terminated without transmission completion. DASH interruption processing re-uses a new TCP stream to resume transmission of data from a breakpoint of the last audio and video chunks that have not been transmitted; HLS interruption processing re-uses a new TCP stream to re-transmit the last non-acknowledged audio or video chunks. Therefore, audio and video chunks with SSL Alert messages can be concatenated (DASH) or de-duplicated (HLS).

3) TRAFFIC INTEGRATION FOR ABORT CONNECTION

Due to the complexity and uncontrollability of the network environment, chunks may be interrupted during transmission. Some key problems in identifying YouTube audio and video in encrypted traffic include:

Network instability can cause packet loss, retransmission, and reordering during the transmission; the integration will be disturbed based on the same ACK Number.

User behaviour operations (pause, playback, fast forward, etc.) will lead to complexity of packets that interferes with segmentation.

Aborts of chunks are of three types: (i) the data chunk is discarded, (ii) the subsequent data of the chunk resumes a broken download, (iii) the chunk is not complete but is not resumed. Identifying these situations when the traffic is encrypted requires the combination of some of the key features of TCP and SSL and the network transmission to set a few thresholds to assist the judgment.

The following modules were designed to solve the problems above. An integration module is responsible for integrating data packets into chunks; a useless-chunk filtering module is responsible for filtering data smaller than 20K bytes; a chunk location-identification module is responsible for labelling the locations of the chunks in a TCP stream; an audio chunk analysis module is responsible for estimating the range of the amount of data in an audio chunk; a disconnection-identification module is responsible for judging the conditions of disconnection flows.

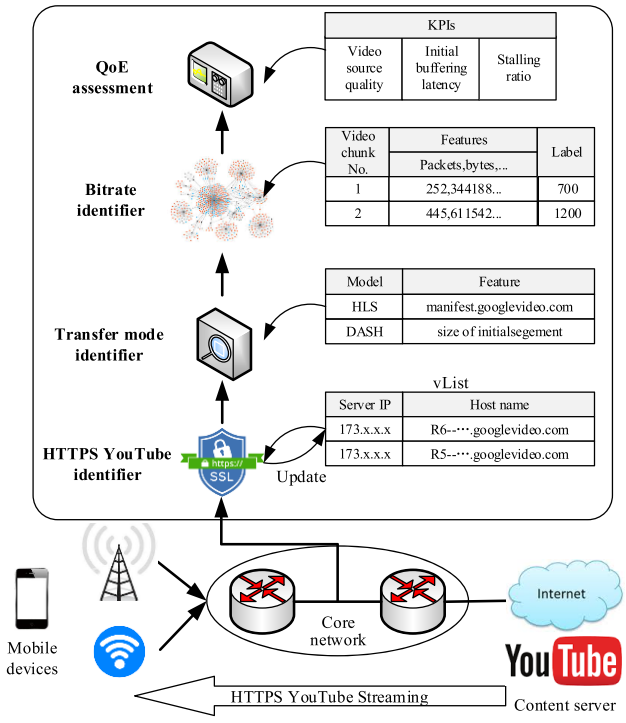


FIGURE 2. Architecture of QoE assessment system.

B. QoE ACCESSMENT SYSTEM

The MBE system works with packet data, passively captured at the vantage point of interest. Fig. 2 shows the architecture of the QoE assessment system. The MBE consists of four steps: (i) identify the HTTPS YouTube traffic, (ii) identify the transfer mode of the traffic, (iii) identify the bitrate of video chunks, and (iv) compute the KPIs and assess the video QoE.

All of the packets between a YouTube server and mobile devices pass through the YouTube traffic identifier to filter the HTTPS YouTube traffic based on a white list (vList) of video server IPs or ''googlevideo'' in the clienthello packets. The transfer mode identifier is used to identify HLS and DASH adaptive streaming based on the previous several packets of flow. Then, the packets from the YouTube server and mobile device are paired into video chunks based on the same ACK number. The bitrate identifier is used to extract the feature of video chunks from the network layer, and then identify the bitrate of video chunks based on the trained classifier. Finally, the bitrate is used to compute the KPIs (e.g., video source quality, initial buffering latency and stalling ratio) and QoE score. First, the HTTPS YouTube ClientHello traffic identification module filters YouTube encrypted traffic based on the previously established video server IP according to the data packet ''googlevideo'' field. Then, the transmission mode identification module identifies the transmission mode according to the traffic characteristics of several previous packets of YouTube video. Next, the video identification identifies the bitrates and resolutions of HLS and DASH modes according to the characteristics of video chunks. Finally, a video QoE evaluation module calculates KPIs and vMOS scores based on the video QoE parameters and video traffic transmission parameters.

C. HTTPS YOUTUBE IDENTIFIER

To assess the video QoE of HTTPS YouTube streaming, we must first identify the HTTPS YouTube video traffic. Under the adaptive streaming mechanism, users will connect to a YouTube video server before interacting with the media profile; based on this, users can select videos with appropriate bitrates and download them from the description file BaseURL address profile according to current network status. As the videos of adaptive bitrate streaming have several backups for different resolutions or codecs, BaseURL corresponds to the specific video sources; e.g., parameter 'itag' in BaseURL represents a specific resolution and codec, as shown in [43]. In the case of non-encrypted video, chunks can be linked into a single video based on the IP addresses of the client and video server. However, the IP address of the media is described in encrypted IP packets, which cannot be parsed. To quickly identify HTTPS YouTube traffic, we prebuilt a video server IP list, or vList, which can be automatically updated later. We can effectively extract a YouTube video server IP address from either the DNS response packets or TLS handshake ''ClientHello'' message. More specifically, we identify a video server's IP by searching for the specific string 'r $*$. googlevideo.com' in TLS-handshake packets or DNS-response, which will also be used to update vList. The records will be removed if not hit within a week. Then, a video stream can be associated with the IP pair, video server, and client using a bloom filter.

D. IDENTIFICATION OF TRANSMISSION MODE

Three transmission modes are used in the current YouTube video service: Apple HLS (HTTP Live Streaming), MPEG DASH (Dynamic Adaptive Streaming over HTTP) and HPD. Distinguishing the video modes is necessary to further estimate the bitrate of the received video streaming.

We adopt the feature-based identification method in our system for efficiency and adaptability. Four identifiable features are currently used in our implementation to distinguish the transmission modes: the number of ACK Number segments, SYN-ACK inter-arrival time, the version of the SSL/TLS protocol, and the bytes of the SSL/TLS protocol handshake packets, which are used to identify the transmission mode by machine learning algorithms; the method is advantageous in that the transmission mode can be recognized based on the previous several packets of flow.

In DASH transmission mode, the server first needs to send the Initial Segment to the client. The Initial Segment contains the initialization information needed by the video decoder before starting to transmit the video data. The first P Application Data packets transmitted after the SSL/TLS handshake appear with S types of ACK numbers, which is important information for distinguishing video transmission modes. Taking the DASH, HLS and HPD transmission modes as an example, it is found that the ACK Number type of the

first three data packets in DASH is 2 or 3, and the types of HLS and HPD are both 1. This feature is very different and occurs early in the video data transmission; it can greatly avoid the adverse effects of retransmission.

In comparing the flow levels of different transmission modes, we found that HPD uses single-flow transmission (video and audio are not separated), while DASH and HLS streaming use several flows. We further found that the transmission always begins and finishes with two flows. Significantly different from DASH, HLS frequently replaces the flow to complete the transfer of the entire video. In these three transmission modes, the statistics of the front two SYN-ACK inter-arrival times show that DASH has the shortest inter-arrival time of the first two flows, HLS is second, and HPD is much longer than HLS and DASH, a feature of HPD that differs greatly from the other transmission modes.

Comparing the flow-level features of the different transmission modes, HPD uses single stream transmission (video and audio are not separated), while DASH and HLS use multiple stream transmission. DASH video transmission always starts with two streams and ends with two streams (for network reasons, they are replaced by another two streams to continue transmission). Clearly different from DASH, HLS frequently changes the stream (in particular, the port) to complete the entire video transmission. Taking these three transmission modes as an example, statistics are collected of the SYN-ACK arrival time intervals of the first two streams. It was found that the first two stream intervals of DASH are the shortest, the HLS is slightly longer, and the SYN-ACK inter-arrival time of HPD is much longer than HLS and DASH; this is the most noticeable difference between HPD and other transmission modes.

Currently, these four adopted features provide sufficient information for our system to quickly and effectively identify the video transmission mode (i.e., HLS or DASH) in HTTPS video streaming. It is worth noting that this video mode identification function is a pluggable module in the system for update if necessary (e.g., due to a change of the vendor's implementation).

E. BITRATE AND RESOLUTION IDENTIFIER

A decision tree is very discriminative in classifying Internet traffic [44]–[46]. In our system, the C4.5 and RandomForest (RF) algorithms are applied to train the traffic classifiers, and Bayes Networks and Adaboost are used for validation and comparison. Fig. 3 diagrams the process of the ML-based bitrate identification, which is composed of three modules: chunk statistics computation, model training, and classification. The chunk statistics computation extracts the network-layer features from captured video traffic. The model training module finds a subset of stable features to build the ML-based classification model. Finally, the classification module identifies the bitrate of HTTPS YouTube video streaming. In the following, we describe each module in detail.

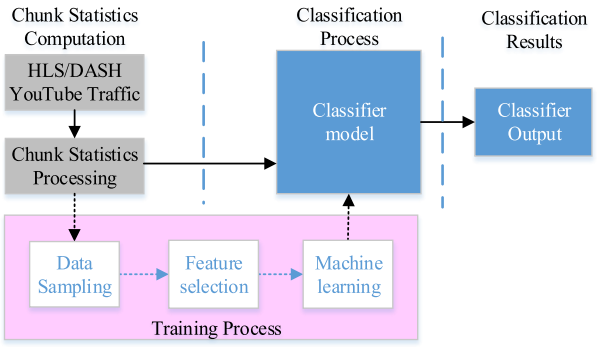


FIGURE 3. The process of bitrate identification based on machine learning.

1) CHUNK STATISTICS AND FEATURE SELECTION

In YouTube adaptive video streaming, a video from a server consists of multiple paired video and audio chunks. Each video and its paired audio chunks are transferred to a client in multiple TCP packets. The client uses the same ACK number for all received TCP packets from the same video and audio chunks. Therefore, we can use the ACK numbers from the client as an index to recover the original structure of video and video chunks from the server. We further count the number of packets from the same chunk C (denoted P_C), and multiply it by the estimated packet size of a chunk (denoted N_C) to get an estimated byte count of each received video chunk, namely $P\_C * N\_C$. The bitrate for a video chunk C (denoted B_C) is the video chunk size divided by playback duration T_C, namely $B\_C = (P\_C * N\_C) / T\_C$.

Through analysis of HTTPS YouTube traffic [47], we found that the YouTube streaming servers start sending an initial burst with a size depending on the current video bitrate. Therefore, the video initial burst bytes can be regarded as a bitrate indicator of the following streaming video. Since the related audio chunk is transmitted at a constant bitrate, dividing the audio chunk size by the audio bitrate yields the playback duration of the corresponding video chunk.

Many chunk statistics are derived directly by counting packets, and packet header. A significant number of features are derived from the TCP headers - we used tcptrace [48] for this information.

Feature selection: One of the key challenges is to identify the best and the most relevant features to properly identify the bitrate of YouTube video streaming. Feature selection (FS) helps to identify the best features to improve the accuracy and reduce the computational complexity related to the construction of the classifier. We selected Correlation-based Feature Selection (FCBF) [49]techniques implemented by weka[50]. Table 1 lists the chosen packet level features that describe video chunks in two directions between a video server and a client.

Table 1 describes the feature subset of bitrate identification and resolution identification. Bitrate identification includes bpackets, bbytes, burst_bytes, audio_bytes and duration.

TABLE 1. Feature subsets of bitrate and resolution identification.

Abbreviation	Feature description	Bitrate	Resolution
bpackets	Packets of video chunks	Yes	Yes
bbytes	Bytes of video chunks	Yes	Yes
burst_bytes	The initial fast-transfer bytes	Yes	-
audio_bytes	Audio bytes	Yes	-
duration	Video chunk duration time	Yes	Yes
resolution	Previous video chunk resolution	-	Yes
brate	Video chunk Bitrate	-	Yes

Bpackets and bbytes represent the number of packets and the number of bytes of a video chunk. Since the playing durations of video chunks are basically the same, the larger the bitrate is, the larger are the number of packets and the number of bytes. Burst_bytes means that when a video starts playing, it will quickly transmit a small piece of video for delivery as soon as possible. The bytes of this quickly-transmitted video chunk is rate-dependent. Audio_bytes indicates the bytes of the audio chunk; the video chunks correspond to the audio chunks one by one. Therefore, since the audio chunk bitrate is fixed, it can be used to estimate the playing duration of the video chunk and prevent some videos with indefinite playing duration from being misidentified, such as a video chunk at the end of playback or when switching the resolution. Duration indicates the duration of the video chunk transmission. The longer the duration, the lower the transmission speed. The adaptive streaming mechanism selects a suitable bitrate according to the transmission speed to prevent the occurrence of a stall. Resolution identification features include bpackets, bbytes, duration, resolution and brate. Resolution represents the resolution of the previous video chunk, brate represents its bitrate, and the adaptive streaming mechanism selects the appropriate resolution based on the bitrate so that the stall matches the watching experience to strike a balance. This subset of features is highly distinctive, and there is no redundancy between features, which can be effectively applied to identification of video QoE parameters.

2) FEATURE DISCRETIZATION

When there are many continuous attributes and the value of any attribute is large, the complexity of the decision tree will increase greatly. Table 2 shows that all the features have continuous data. Continuous data, as a decision tree node, will have many branches, which will affect the generation and classification efficiency of the decision tree. This article uses the minimum description length method MDL for data discretization.

MDL uses the length of the description language to represent the complexity of the model with the goal of achieving low complexity and high accuracy. The longer the description language is, the higher the accuracy; the shorter the description language, the lower the model complexity. According to the mathematical description, the goal of the MDL model is to minimize the description language length M_{mdl}.

$$M_{mdl} = \arg\min_{M_i \in M} \{|L_m(M_i)| + |L_c(D|M_i)|\} \quad (1)$$

$|L_m(M_i)|$ represents the number of bits needed by the model, $L_m(M_i)$ represents the description language of the model, $L_c(D|M_i)$ represents the language of the model M_i for description object D, and $|L_c(D|M_i)|$ indicates the length required for the corresponding description language. Each object can be regarded as consisting of a deterministic sequence and a random sequence. M_i is the deterministic sequence; the random sequence represents the error between the determinate sequence and the object; the determinate sequence can be expressed by an autoregressive model or a polynomial model; the length of the description language $|L_m(M_i)|$ is the number of parameters in the model; the random sequence can be described by a probability-distribution model. According to the Shannon theorem, when the stochastic process is represented by a probability-distribution model, the length of the description $|L_c(D|M_i)|$ is a negative logarithm of base 2.

F. QoE ACCESSMENT MODEL

Due to their high cost and time-intensiveness, subjective measurements are typically replaced by objective measurements to assess video QoE. To establish an objective QoE assessment framework, a leading telecom vendor (Huawei) uses eye tracking and physiography [21] to measure human perception of video to quantify the impacting factors [20]. Following a physiological index, an inflection point reflects a mood change of an experimental subject. Then, according to the existing definition of an emotional criterion, we define a key-indicator point (1 to 5). Mapping between the experimental subject's subjective feelings and the objective assessments, it turns out that video source quality, initial buffering latency, and stall duration have the greatest effects on consumers' experiences. We can then establish a video MOS (vMOS) assessment framework to synthetically assess the impact of multiple KPIs on the user-perceived video quality. We can objectively present the video experience as affected by the video source quality and the network transmission performance.

The main factors that affect the video source quality are the video codec (VC) (e.g., H.264, H.265, VP9), the codec profile (CP) (e.g., Main Profile, High Profile), video resolution, and video bitrate. The maximum score under a particular video resolution is denoted Qualitymax. For example, on a scale of 1 to 5 (5 is the best), for video resolutions 4K, 2K, 1080p, 720p, 480p, and 360p, the corresponding Qualitymax values are 4.9, 4.8, 4.5, 4, 3.6, and 2.8, respectively. The score of video source quality (sQuality) [21] is defined as follows [21]:

$$sQuality = Qualitymax * (1 - 1/((1 + VB * VC * CP/VR)^2)) \quad (2)$$

When assessing the impact of buffering, both the initial buffering latency (IBL) and the total duration of stalls during

TABLE 2. Distribution of video samples (number of chunk samples / number of videos).

Modes	Duration	Resolution				
		360P	480P	720P	1080P	Adaptive streaming
HLS	short	5042/120 (20,10,10,80)	4217/100 (20,10,10,60)	3274/80 (10,10,10,80)	2561/60 (10,10,10,30)	6138/100 (10,10,10,70)
	medium	8490/100 (20,10,10,60)	6518/80 (20,10,10,40)	4857/60 (10,10,10,30)	3526/40 (10,10,10,10)	7916/80 (10,10,10,50)
	long	15371/100 (20,10,10,60)	12036/80 (20,10,10,40)	9858/60 (10,10,10,30)	6295/40 (10,10,10,10)	11324/60 (10,10,10,30)
DASH	short	3714/150 (20,10,10,110)	3287/120 (20,10,10,80)	3019/100 (10,10,10,70)	2871/80 (10,10,10,50)	3502/100 (10,10,10,70)
	medium	5241/130 (20,10,10,90)	4902/100 (20,10,10,60)	4583/80 (10,10,10,50)	4225/80 (10,10,10,50)	5161/80 (10,10,10,50)
	long	8582/120 (20,10,10,80)	7391/90 (20,10,10,50)	6752/70 (10,10,10,40)	6382/70 (10,10,10,40)	7092/60 (10,10,10,30)

a video playout should be considered. Initial buffering latency is the time from clicking ''Play'' to video playback. Insufficient bandwidth in mobile communication networks often results in a long video buffering time. The score of initial buffering latency (sLatency) is defined as follows [21]:

$$\text{sLatency} = \begin{cases} 5 & IBL \le 0.1 \\ 1 & IBL > 10 \\ 0.25 + 4.66 * e^{-IBL/5.37} \end{cases} \quad (3)$$

Another KPI is the incidence of stalling events during playback, including the stalling durations and frequency. These determine the stalling ratio (SR). The stalling ratio score (sStalling) is defined as follows [21]

$$\text{sStalling} = \begin{cases} 5 - 20 * SR & SR \le 0.15 \\ 0 & SR > 0.45 \\ 2 - 20 * (SR - 0.15)/3 \end{cases} \quad (4)$$

Considering the impact of these three KPIs, video source quality, initial buffering latency, and stalling ratio, we can objectively assess the video QoE using the vMOS score as defined below [21]:

$$\begin{aligned} vMOS = {} & (1 - 0.092 * (1 + 2e^{(-sLatency)})) * (5 - sLatency) \\ & * sQuality - 0.018 * (1 + 2e^{(-sStalling)}) \\ & * (5 - sStalling) \end{aligned} \quad (5)$$

IV. EXPERIMENTAL EVALUATION

A. EXPERIMENTAL SETUP

We have designed experiments for evaluating the transmission quality of YouTube video streaming over different mobile user access networks, including WiFi and 4G, as shown in Fig. 4. Traffic was passively collected from different YouTube servers at selected user sites in four different countries and areas: including South Korea, Brazil, Hong Kong, and Shanghai. To verify the robustness of the identification methods, we sampled with different transmission modes, different durations, and different resolutions. The videos of HLS transmission mode were obtained by an iPhone client; the videos of the DASH transmission mode were obtained by Android browsers and clients. Videos with

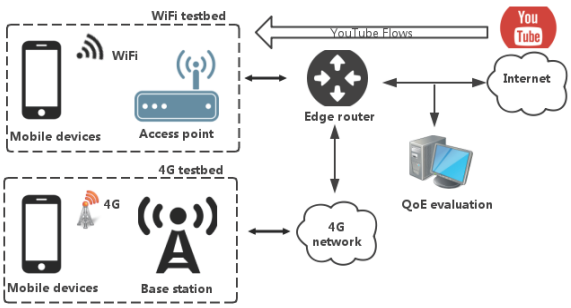


FIGURE 4. The QoE assessment framework for YouTube video.

TABLE 3. Mobile devices.

Devices	OS version	Screen resolution	Screen size	Memory
iPhone 6s	iOS9.1	1334×750	4.7	2G
iPhone 6	iOS8.3	1334×750	4.7	1G
iPhone 5	iOS8.3	1136×640	4	1G
HTC M7	Android 4.4.3	1920×1080	4.7	2G
Samsung S4	Android 4.4.2	1920×1080	5.0	2G
Xiaomi MI 2	Android 4.1	1280×720	4.3	2G

different durations were captured. Short videos were not longer than 5 minutes; medium videos were 5 to 10 minutes; long videos were more than 10 minutes. Fixed resolutions (360 P, 480 P, 720 P, 1080 P) and adaptive bitrate video were captured, as shown in Table 2.

We used a Huawei 4G base-station mirror to capture traffic from a 4G cellular network, and used a laptop in WiFi hotspots to capture traffic to and from connected Android and iOS terminals. Table 3 lists the specific mobile devices used in our experiments.

Since video data are encrypted and involve privacy protections, currently no authoritative public data set was used to evaluate the performance of QoE parameter identification of encrypted video. To make the identification method comparable, the article evaluates the Top 10 YouTube videos of 2016; we collected five types of code streams for each video: 360 p, 480 p, 720 p, 1080 p and adaptive. The specific videos are shown in Table 4.

TABLE 4. Top 10 YouTube videos of 2016.

No.	Video	Duration	DASH	HLS
1	Adele Carpool Karaoke	14:52	84	167
2	PPAP（Pen Pineapple Apple Pen）	1:09	7	13
3	What's inside a Rattlesnake Rattle?	6:06	35	69
4	Nike Football Presents: The Switch ft. Cristiano Ronaldo, Harry Kane, Anthony Martial & More	5:58	34	67
5	Grace VanderWaal: 12-Year-Old Ukulele Player Gets Golden Buzzer - America's Got Talent 2016	5:25	31	61
6	Water Bottle Flip Edition | Dude Perfect	7:28	42	84
7	Channing Tatum & Beyonce's "Run The World (Girls)" vs. Jenna Dewan-Tatum's "Pony" | Lip Sync Battle	4:43	27	53
8	Donald Trump: Last Week Tonight with John Oliver (HBO)	21:54	123	246
9	The $21,000 First Class Airplane Seat	9:05	51	102
10	Brothers Convince Little Sister of Zombie Apocalypse	3:41	21	42

B. GROUND TRUTH

To validate the performance of the proposed ML-based identification method, we developed a Fiddler-based tool to label video samples. The Fiddler [51] functions as a middle agent between a client and an HTTPS video streaming server to decrypt the traffic [52]. First, the Fiddler establishes a TLS connection with the server on behalf of the client using the server certificate to handle requests and responses. Next, the Fiddler establishes a TLS connection with the client on behalf of the server, using the Fiddler's certificate to handle the related requests and responses. Through decrypting HTTPS traffic, we can extract the size and playback duration of video chunks, and further calculate the corresponding video bitrates. Therefore, the bitrates of video chunk samples can be labelled using the selected features. Our experimental results show that different resolutions (240 p, 360 p, 480 p, 720 p, 1080 p, 2K) on a mobile device correspond to average bitrates of 250 Kbps, 450 Kbps, 700 Kbps, 1.5 Mbps, 3 Mbps, and 6 Mbps, respectively. Accordingly, we label these samples based on their bitrates as 250, 450, 700, 1500, 3000, 6000 Kbps.

C. ACCURACY

To evaluate the performance of the MBE approach, the most commonly used video transmission modes HLS and DASH were used to validate the accuracy of MBE with four machine learning methods: RandomForest (RF-based), Bayes Networks, C4.5, and AdaBoost. We define:

$$accuracy = \sum_{i=1}^{m} (TP_i) / \sum_{i=1}^{m} (TP_i + FN_i) \tag{6}$$

Here, m is the number of bitrate types, TP represents the number of labelled samples in which the actual bitrate type is i, and FN is the number of samples in which the actual type i is mis-identified as another type.

TABLE 5. Accuracy of transmission mode identification.

Modes \ Approach	Random Forest	Bayes networks	C4.5	Adaboost
DASH	97.5	99.5	97.5	97
HLS	96	96.1	97.3	95.9
HPD	99.3	99.3	98	98

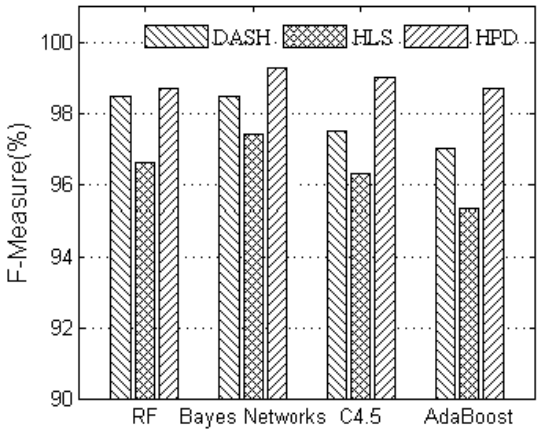


FIGURE 5. F-Measure on transmission mode.

1) ACCURACY OF TRANSMISSION MODE IDENTIFICATION

To verify the effectiveness of the method of identifying the transmission mode based on machine learning, videos in HLS, DASH and HPD transmission modes were adopted to verify and compare the identification performance of four commonly used machine learning methods, as shown in Table 5 and Fig. 5.

As can be observed in Fig. 5, the accuracy of identifying the encrypted transmission mode based on machine learning was higher than 95.9%. Because the number of ACK Number segments, the SYN-ACK arrival interval, the SSL/TLS protocol version, and the bytes of the SSL/TLS protocol handshake are strongly correlated features, machine learning methods can better use their relevance to identify the transmission mode for encrypted YouTube videos. Table 5 and Fig. 5 show that the Bayes Networks' average accuracy and F-Measure are higher than that of other algorithms because Bayesian networks, unlike decision trees, are effective with probabilistic inference models that imply the relevance between network nodes. The accuracy of HLS identification is lower than that of DASH because the ACK Number of HLS and HPD have similar features and HLS is easily misidentified as HPD mode.

2) ACCURACY OF BITRATE IDENTIFICATION

HPD is a progressive download, different from the streaming media server for which each transmission is approximately 5-10 s of video data; the HPD video server will continue to transmit data until the video data download is complete. The HPD transmission mode is not identified in the paper. The

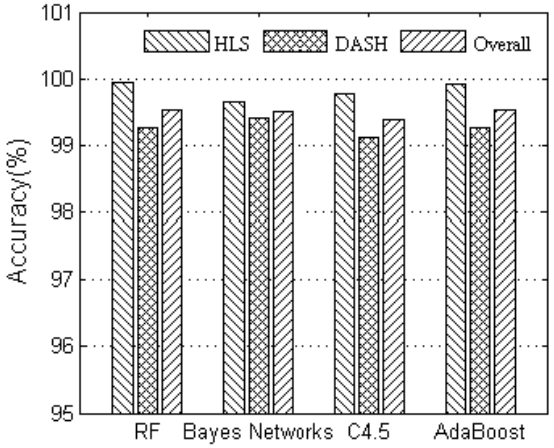


FIGURE 6. Accuracy of bitrate identification on HLS, DASH and overall videos.

TABLE 6. Number of errors in video chunk bitrate identification.

No.	1	2	3	4	5	6	7	8	9	10
DASH	1	0	0	0	0	1	0	1	0	0
HLS	0	0	1	1	1	0	0	3	0	0

number of HLS video chunks was 144012 and the number of DASH video chunks was 109624. The results are shown in Fig. 6.

As shown in Fig.6, the minimum accuracy of MBE with the four adopted algorithms is higher than 99.1%. It is worth nothing that the bitrate estimation accuracy for DASH is generally lower than that for HLS; this is due to the different playback durations between DASH and HLS modes. The average playback duration of DASH video chunks is approximately 10 seconds, but only 5 seconds for HLSs. Generally speaking, the longer the playback duration of video chunks, the higher the fluctuation of byte count. Thus, longer playback duration implies a greater chance of errors on byte counts and hence on the estimation of the corresponding bitrates.

To make the bitrate identification method comparable, using Top 10 YouTube videos of 2016 for verification, the numbers of errors in video chunk bitrate identification are shown in Table 6.

It can be observed from Table 6 that the number of errors in DASH and HLS bitrates are 3 and 6, respectively, indicating that the identification method can effectively identify the video bitrate.

3) ACCURACY OF IDENTIFYING RESOLUTION

Table 7 describes the identification accuracy of 4 machine-learning algorithms. Identification accuracy is based on a comprehensive assessment of the entire data set. A good algorithm has not only high accuracy, but also good identification performance for each class, particularly when the samples of each class are unevenly distributed. The F-Measure can effectively describe the identification performance of various algorithms; the results are shown in Fig. 7.

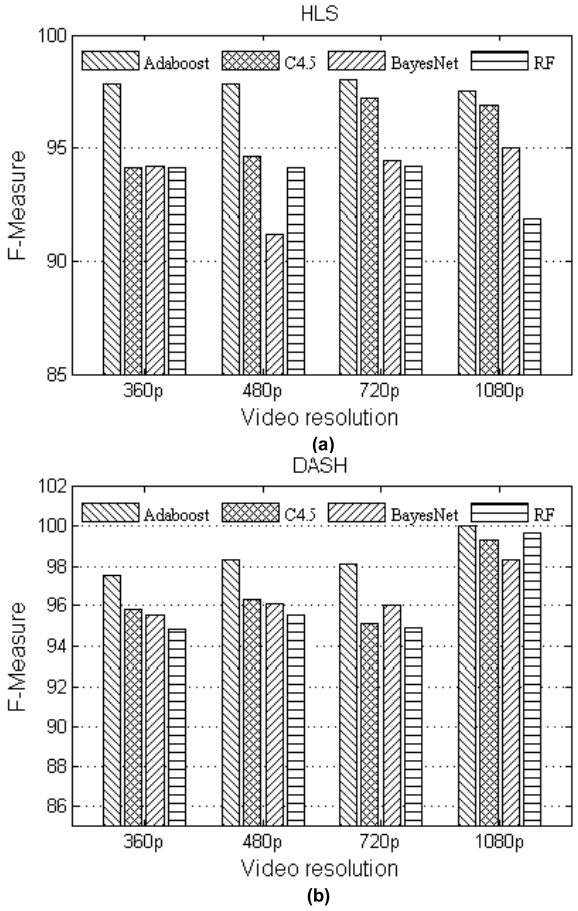


FIGURE 7. F-Measure of resolution identification. (a) F-Measure of transmission mode HLS. (b) F-Measure of transmission mode DASH.

TABLE 7. Identification accuracy.

Approach	Adaboost	C4.5	Bayes networks	Random Forest
HLS	97.8	95.78	93.7	93.7
DASH	98.18	96.09	96.09	95.49
Overall accuracy	97.99	95.94	94.9	94.6

As can be observed in Table 7, machine learning methods have achieved a good accuracy; this is because the numbers of bytes of video blocks have a strong correlation with the resolution of the previous video block feature. The AdaBoost method has the highest accuracy; the accuracy of HLS was 97.8%, and DASH was 98.18%. Because the Adaboost classifier trained different weak classifiers on the same data set, the weak classifiers were integrated to obtain a relatively good classification performance. While the RandomForest classifier was built according to the different feature subsets, a classifier with weak classification performance affects the final result. As can be observed in Fig. 7, AdaBoost has better identification results for every resolution. AdaBoost can improve performance on resolution identification through the advantages of ensemble learning.

TABLE 8. Numbers of resolution mistakes.

No.	1	2	3	4	5	6	7	8	9	10
DASH	2	0	0	1	0	0	1	3	1	1
HLS	3	1	2	0	2	0	3	5	2	4

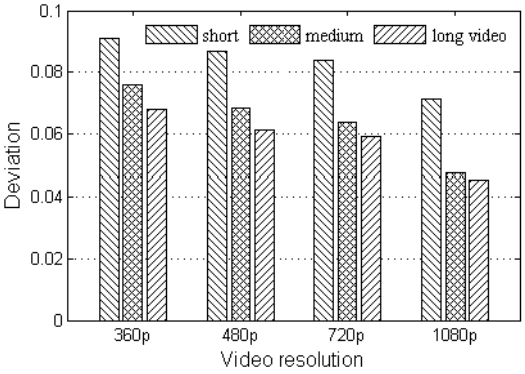


FIGURE 8. The impact of playback duration and resolution on bitrate estimation.

A comparison of the methods, using top 10 YouTube videos of 2016 to verify the identification result, is shown in Table 8.

From Table 8, the numbers of HLS and DASH mistakes identified were 22 and 9, respectively. The accuracies were 98.02% and 97.57%, respectively, indicating that the identification method can be identified effectively.

4) IMPACT OF BITRATE DEVIATION

In the following, we study the impact of playback duration and resolution on bitrate estimation using the Deviation Ratio. Playback duration and resolution are two critical video characteristics that are closely related to user perception of video quality. We define: Deviation Ratio = |actual bitrate − bitrate estimation|/actual bitrate. In our study, we analysed three lengths of streaming videos: short (≤5 minutes), medium (5 ∼ 10 minutes), and long (>10 minutes). We chose 100 videos of each length with different resolutions, including 360 p, 480 p, 720 p and 1080 p.

As shown in Fig. 8, for videos with same playback duration, the lower the video resolution, the higher the bitrate deviation ratio, which ranged up to 9.1%. Because the bitrate of low-resolution video is relatively low, the absolute deviation will be relatively high at the same deviation ratio. In addition, the bitrate deviation of medium and long video is less because the number of video chunks is small, resulting in an uneven distribution of bitrate estimation. For example, although the deviation of 1080 p video is relatively small, the deviation value between actual bitrate and estimated bitrate is larger.

Fig. 9(a, b, c, d) describes the bitrate of MBE and real bitrate of the video ''Cristiano Ronaldo - A Great Person''. The average bitrates of the resolutions of 360 p, 480 p, 720 p, and 1080 p are approximately 400 kbps, 800 kbps, 1500 kbps, and 3000 kbps, respectively. It can be observed that MBE has achieved good identification effectiveness for videos with different resolutions. Additionally, it is able to maintain good identification performance with the video playback for a long time, and can be applied to large bitrate variations effectively. Thus, MBE can be effectively used to identify bitrates of encrypted video streaming.

TABLE 9. Rate identification error (%).

No.	1	2	3	4	5	6	7	8	9	10
DASH	0.6	1	0.7	1.2	0.8	0.3	1.1	0.9	1.7	1.6
HLS	1.3	1.7	1.6	1.4	1.2	0.5	0.4	1.0	1.3	1.3

5) IMPACT OF BITRATE DEVIATION ON KPIS

Video bitrate is the key parameter of KPIs. However, bitrate deviation affects not only the video source quality, but also the stalling ratio, initial buffering latency, and final video QoE assessment, as shown below.

To make bitrate identification methods comparable, error of bitrate identification is shown in Table 9 for the top 10 YouTube videos of 2016.

Table 9 shows that the error rate of the bitrate identification method is 0.4-1.7%; the minimum error of the DASH mode is 0.3%, and the minimum error of HLS mode is 0.4%, indicating that the identification method can be effectively used for encrypted video bit rate identification

6) THE IMPACT OF BITRATE DEVIATION ON VIDEO SOURCE QUALITY

The score of video source quality is presented in Equation (1). The parameters, Qualitymax, VC, CP and VR are intrinsic video properties and are not affected by bitrate. Therefore, the impact of bitrate deviation on video source quality simply needs to consider the factor of bitrate. The video source quality of machine learning and real bitrate of the video ''Cristiano Ronaldo - A Great Person'' are shown in Fig. 10.

As shown in Fig.10, the maximum scores of the video source quality for video resolutions 1080 p, 720 p, 480 p, and 360 p were 4.5, 4, 3.6, and 2.8, respectively, which matches our analysis result: the low-resolution results of low scores of video source quality. The average deviations of video source quality with different resolutions were also counted; with 100 videos of each resolution. The average deviation of video source quality decreases with increasing resolution; the deviations for video resolutions 360 p, 480 p, 720 p, and 1080 p were 1.32%, 0.81%, 0.64% and 0.43% respectively, which can be negligible. As the resolution falls, so does the video source quality score; the same score variation results in a relatively large deviation for a video with low resolution.

7) THE IMPACT OF BITRATE DEVIATION ON VIDEO STALLING RATIO

Stalling is the interruption of video playback due to an empty playout buffer, typically triggered when network throughput

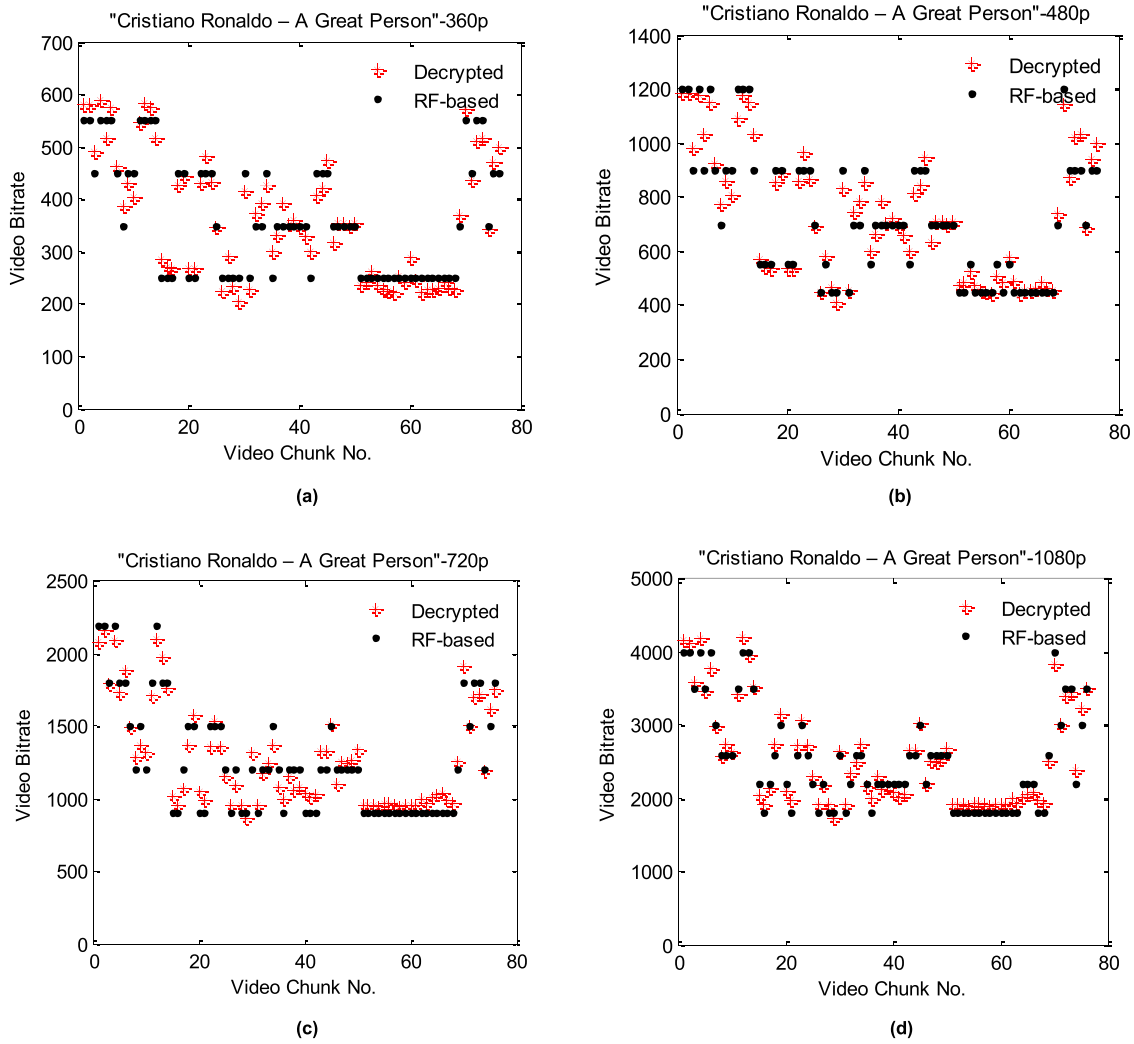


FIGURE 9. Bitrate deviations on varying resolutions of video "Cristiano Ronaldo - A Great Person". (a) Bitrate deviation between MBE and real bitrate on 360 p video "Cristiano Ronaldo - A Great Person". (b) Bitrate deviation between MBE and real bitrate on 480 p video "Cristiano Ronaldo - A Great Person". (c) Bitrate deviation between MBE and real bitrate on 720 p video "Cristiano Ronaldo - A Great Person". (d) Bitrate deviation between MBE and real bitrate on 1080 p video "Cristiano Ronaldo - A Great Person".

is lower than video bitrate. The Stalling Ratio (SR) score is defined in Equation (2). SR represents the ratio of stalling duration; SR values of 0, 5%, 10%, 15%, and 30% correspond to SR scores of 5, 4, 3, 2, and 1, respectively. The stalling ratio reflects the proportion of stalling in the total video watching duration. With Stalling Ratio = stalling duration / viewing duration, and viewing duration = video size / bitrate + stalling duration, it can be seen that the stalling ratio will be affected by the bitrate deviation. Reference [14] found an exponential relationship between stalling parameters and MOS and that users may tolerate at most one stalling event of up to three-seconds. We set multiple stalling durations as 0.5 s, 1 s, 1.5 s and 3 s for comparison. The results are shown in Fig. 11.

As shown in Fig. 11, a longer stalling duration results in a greater average deviation of stalling ratio. When the stalling duration is 0.5 s, the minimum deviation of the stalling factor is 1%; when it is 3 s, the deviation of the stalling ratio is between 4.4% and 6.7%.

8) THE IMPACT OF BITRATE DEVIATION ON INITIAL BUFFERING LATENCY

Initial buffering latency reflects the waiting time from clicking on the "Play" button to the start of the video when users watch YouTube videos. The score of initial buffering latency (IBL) is presented in Equation (3); IBL $\leq$ 0.1, 1, 3, 5, and 10 s correspond to buffer perceived scores of 5, 4, 3, 2, and 1, respectively. However, initial buffering latency = rate $*$ 2 s / initial buffer average rate, and initial buffering latency is also associated with the bitrate. The deviations of initial buffering latency with initial buffering rates of 2 Mbps, 4 Mbps, 8 Mbps, 16 Mbps, and 32 Mbps were counted; the results are shown in Fig. 12. The scores of the initial loading latency of the video "Cristiano

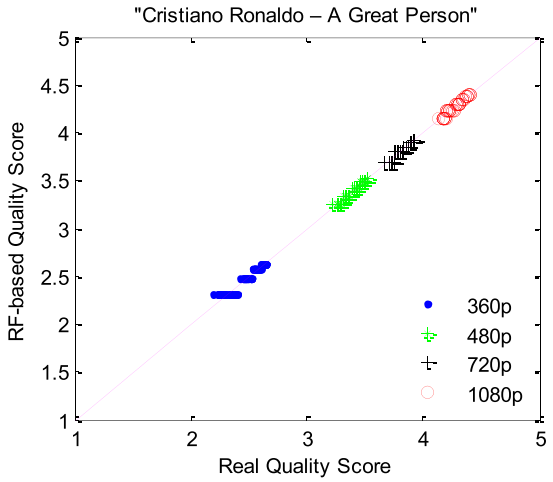


FIGURE 10. Score of video source quality of MBE and real bitrate on varying resolutions.

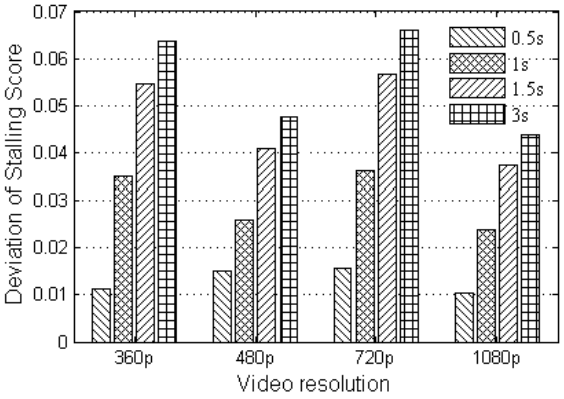


FIGURE 11. Bitrate deviation impact on video stalling ratio of varying resolutions.

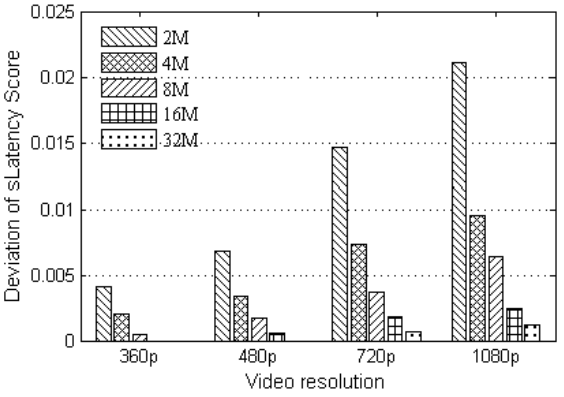


FIGURE 12. Bitrate deviation impact on video initial buffering latency of varying resolutions.

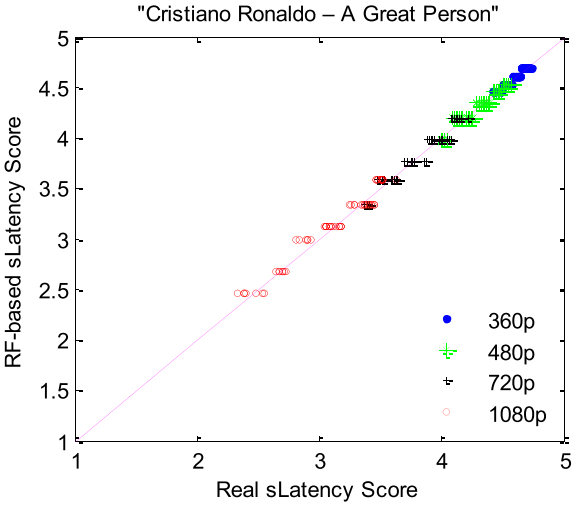


FIGURE 13. Scores of video initial buffering latency of MBE and real bitrate on varying resolutions.

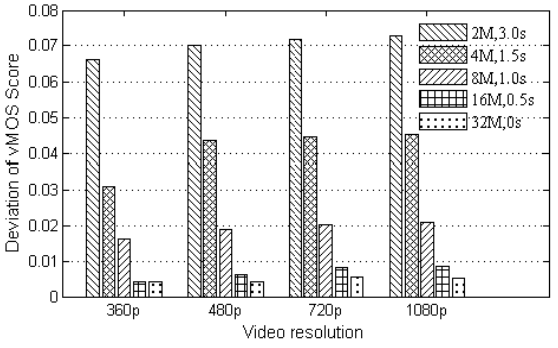


FIGURE 14. The impact on vMOS of average initial buffering rate and stalling duration caused by bitrate deviation.

Ronaldo - A Great Person'' on varying resolutions are shown in Fig. 13.

As shown in Figs. 12 and 13, a higher average initial buffering rate results in a small average deviation of initial buffering latency. When the average initial buffering rate comes to 2 Mbps, the deviation of different resolutions is between 0.4% and 2.1%. The average deviation increases with resolution; when the average initial buffer rate is 32 Mbps, the deviation of different resolutions is approximately 0.1%, and the impact can almost be ignored.

9) THE IMPACT OF BITRATE DEVIATION ON VMOS

The impact of bitrate deviation on video source quality, stalling ratio and initial buffering latency will ultimately affect the video QoE assessment under the framework of vMOS. As shown in Equation (4), the impacts of bitrate deviation on vMOS are counted based on varying initial buffering rates and stalling durations. The corresponding experimental results are shown in Fig. 14.

As shown in Fig. 14, a smaller average initial buffering rate and longer stalling duration result in great deviations of vMOS assessment; the maximum average deviation is 7.2%. With optimized network conditions, the average deviation decreases; the average deviation of vMOS on varying resolutions is minimized to 0.4%. A vMOS score of 4 represents that the QoE is good (vMOS ranges between 1-5), so the variation value caused by maximum deviation is

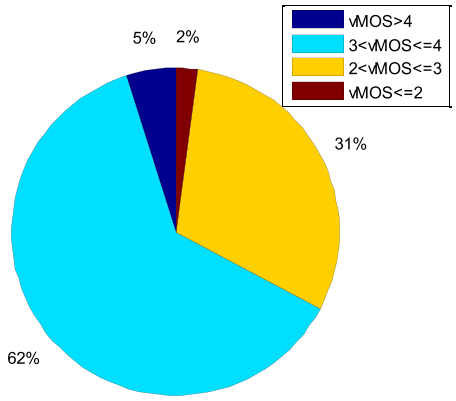


FIGURE 15. VMOS distribution on some mobile networks.

between 0.22-0.29. When the vMOS is generally between 3 and4, the change value of the minimum deviation is between 0.01-0.02, which can be negligible.

On the whole, bitrate identification based on machine learning has less impact on obtaining the KPIs of the QoE assessment framework. The ultimate impact on vMOS is small, and as a result, it is well suited for encrypted video bitrate identification for evaluation of video QoE.

10) VMOS DISTRIBUTION

Based on the QoE analysis, we use vMOS framework to assess some mobile networks, as shown in Fig. 15. Base on the proportion of vMOS scores, we can evaluate the video QoE of mobile network. VMOS scores of 1-5 represent bad, poor, fair, good and excellent respectively.

As reported in Fig. 15, about 5% of the total videos in test network of YouTube QoE is good (i.e., MOS $>= 4$). Among 62% of the total videos, their corresponding QoE was fair (i.e. $3 <$ MOS $<= 4$); for about 31% of the total videos, their related QoE was poor (i.e. $2 <$ MOS $<= 3$); for the rest of 2% of the total videos, their QoE was bad (i.e. MOS $<= 2$). When the QoE assessment returns as bad and poor quality, the assessment system will alert the mobile operators for possible network performance improvement.

V. CONCLUSIONS

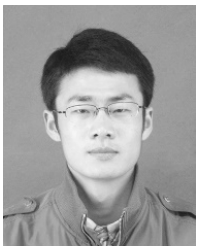
To protect user privacy and to prevent the interference of ISPs during video transmissions, more and more online video services use HTTPS encrypted transmission, and the traditional DPI methods cannot obtain the video size, play duration, etc. This leads to the failure of the original video QoE assessment methods. To solve this problem, we have proposed a machine-learning-based approach, MBE, to effectively estimate the bitrates of HTTPS YouTube video streaming for QoE evaluation. MBE relies exclusively on readily available IP packet level measurements to obtain the bitrate information of encrypted video streaming, the most critical information for video QoE assessment in smart cities. We have also extensively studied the impact of deviations in bitrate estimation on the KPI parameters used in vMOS, a systematic objective QoE assessment framework. Our experiments show that our proposed MBE based QoE assessment framework can effectively and accurately estimate the user perception of the quality of YouTube adaptive streaming service in real time, which is of special importance to network operators and video content providers to flexibly configure network resources to save energy in smart cities.

In future work, we will develop a QoE validation tool based on MBE for assessing YouTube HTTPS video streaming services under different network conditions in smart cities. We will additionally consider incorporating user subjective feelings to improve the objective vMOS QoE framework under different network conditions and application scenarios. Another interesting research direction is to apply QoE assessment along with QoS monitoring to enhance fault diagnosis in smart cities, which is also included in our future research agenda. In addition, taking into account the superiority of the QUIC UDP protocol compared to the TCP protocol, we will strive to resolve the QUIC UDP traffic YouTube video bit rate and identify resolution of encrypted traffic in smart cities.

REFERENCES

[1] K. Wang *et al.*, "A survey on energy Internet: Architecture, approach, and emerging technologies," *IEEE Syst. J.*, to be published.
[2] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green industrial Internet of Things architecture: an energy-efficient perspective," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 48–54, Dec. 2016.
[3] YouTube Press. (Jan. 2016). *Statistics*. [Online]. Available: http://YouTube.com/yt/press/statistics.html
[4] V. Aggarwal, "Prometheus: Toward quality-of-experience estimation for mobile apps from passive network measurements," in *Proc. HotMobile*, Santa Barbara, CA, USA, 2014, p. 18.
[5] K. Wang and Y. Yu, "A query-matching mechanism over out-of-order event stream in IOT," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 13, nos. 3–4, pp. 197–208, Jul. 2013.
[6] F. Wamser, "YoMoApp: A tool for analyzing QoE of YouTube HTTP adaptive streaming in mobile networks," in *Proc. EuCNC*, Paris, France, 2015, pp. 239–243
[7] M. Seufert, "On the monitoring of YouTube QoE in cellular networks from end-devices," in *Proc. S*, Paris, France, 2015, p. 23.
[8] F. Wamser *et al.*, "Poster: Understanding YouTube QoE in cellular networks with YoMoApp: A QoE monitoring tool for YouTube mobile," in *Proc. MobiCom*, Paris, France, 2015, pp. 263–265.
[9] H. Nam *et al.*, "Youslow: A performance analysis tool for adaptive bitrate video streaming," in *Proc. SIGCOMM*, 2014, pp. 111–112.
[10] M. Seufert *et al.*, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 469–492, 1st Quart., 2014.
[11] X. He, K. Wang, H. Huang, and B. Liu, "QoE-driven big data architecture for smart city," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 14–18, Feb. 2018.
[12] P. Casas, M. Seufert, and R. Schatz, "YOUQMON: A system for online monitoring of YouTube QoE in operational 3G networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 2, pp. 44–46, Sep. 2013.
[13] F. Wamser *et al.*, "Demonstrating the prospects of dynamic application-aware networking in a home environment," in *Proc. SIGCOMM*, Chicago, IL, USA, 2014, pp. 149–150.
[14] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of YouTube QoE via crowdsourcing," in *Proc. ISM*, Dana, CA, USA, 2011, pp. 494–499.
[15] R. K. P. Mok *et al.*, "Inferring the QoE of HTTP video streaming from user-viewing activities," in *Proc. W-MUST*, Toronto, ON, Canada, 2011, pp. 31–36
[16] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial delay vs. interruptions: Between the devil and the deep blue sea," in *Proc. QoMEX*, Yarra Valley, VIC, Australia, 2012, pp. 1–6.

[17] J. Yao, S. S. Kanhere, M. Hassan, and I. Hossai, "Empirical evaluation of HTTP adaptive streaming under vehicular mobility," in *Proc. Netw*, Valencia, Spain, 2011, pp. 92–105
[18] B. Lewcio, B. Belmudez, A. Mehmood, M. Wältermann, and S. Möller, "Video quality in next generation mobile networks—Perception of time-varying transmission," in *Proc. CQR*, Naples, FL, USA, 2011, pp. 1–6.
[19] T. Hoßfeld, M. Seufert, C. Sieber, and T. Zinner, "Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming," in *Proc. QoMEX*, Singapore, 2014, pp. 111–116
[20] Huawei. (Jan. 2016). *Mobile Video Service Performance Study*. [Online]. Available: http://www.huawei.com/minisite/hwmbbf15/img/video_coverage_whitepaper_en.pdf
[21] Huawei. (Jan. 2016). *Mobile MOS*. [Online]. Available: Jan. 2016http://www.mbblab.com:9090/mobilemos/index.php!r=site/index
[22] H. Nam, B. H. Kim, D. Calin, and H. G. Schulzrinne, "Mobile video is inefficient: A traffic analysis," *Nexus*, vol. 1, pp. 1–5, Mar. 2013.
[23] Y. Qi and M. Dai, "The effect of frame freezing and frame skipping on video quality," in *Proc. IIH-MSP*, Pasadena, CA, USA, 2006, pp. 423–426.
[24] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of HTTP video streaming," in *Proc. IM*, Dublin, Ireland, 2011, pp. 485–492
[25] M. Seufert, F. Wamser, P. Casas, R. Irmer, P. Tran-Gia, and R. Schatz, "YouTube QoE on mobile devices: Subjective analysis of classical vs. adaptive video streaming," in *Proc. IWCMC*, Dubrovnik, Croatia, 2015, pp. 43–48.
[26] T. Zinner *et al.*, "Controlled vs. uncontrolled degradations of QoE: The provisioning-delivery hysteresis in case of video," in *Proc. EuroITV*, 2010, Tampere, 2010.
[27] G. Gómez *et al.*, "YouTube QoE evaluation tool for Android wireless terminals," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, p. 164.
[28] T. T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.
[29] D. Bonfiglio, M. Mellia, P. Tofanelli, D. Rossi, and M. Meo, "Revealing skype traffic: When randomness plays with you," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 37–48, Oct. 2007.
[30] K. Wang, H. Li, Y. Feng, and G. Tian, "Big data analytics for system stability evaluation strategy in the energy Internet," *IEEE Trans. Ind. Inf.*, vol. 13, no. 4, pp. 1969–1978, Aug. 2017.
[31] K. Wang *et al.*, "Wireless big data computing in smart grid," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 58–64, Apr. 2017.
[32] L. Bernaille, I. Akodkenou, K. Salamatian, A. Soule, R. Teixeira, "Traffic classification on the fly," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 2, pp. 23–26, Apr. 2006.
[33] P. Bermolen *et al.*, "Abacus: Accurate behavioral classification of P2P-TV traffic," *Comput. Netw.*, vol. 55, no. 6, pp. 1394–1411, Apr. 2011.
[34] V. Paxson, "Empirically derived analytic models of wide-area TCP connections," *IEEE/ACM Trans. Netw.*, vol. 2, no. 4, pp. 316–336, Aug. 1994.
[35] S. Zander, T. Nguyen, and G. Armitage, "Self-learning IP traffic classification based on statistical flow characteristics," in *Proc. PAM*, Boston, MA, USA, 2005, pp. 325–328.
[36] I. Paredes-Oliva, I. Castell-Uroz, P. Barlet-Ros, X. Dimitropoulos, and J. Solé-Pareta, "Practical anomaly detection based on classifying frequent traffic patterns," in *Proc. INFOCOM WKSHPS*, Orlando, FL, USA, 2012, pp. 49–54.
[37] R. Alshammari and A. N. Zincir-Heywood, "Unveiling Skype encrypted tunnels using GP," in *Proc. CEC*, Barcelona, Spain, 2010, pp. 1–8.
[38] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi, "Detailed analysis of Skype traffic," *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 117–127, Jan. 2009.
[39] K.-T. Chen, C. Y. Huang, P. Huang, and C. L. Lei, "Quantifying Skype user satisfaction," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 399–410, 2006.
[40] R. Bar-Yanai, M. Langberg, D. Peleg, and L. Roditty, "Realtime classification for encrypted traffic," in *Proc. Experim. Algorithms*, Naples, Italy, 2010, pp. 373–385.
[41] M. Korczynski and A. Duda, "Markov chain fingerprinting to classify encrypted traffic," in *Proc. INFOCOM*, Toronto, ON, Canada, 2014, pp. 781–789.
[42] A. R. Khakpour and A. X. Liu, "An information-theoretical approach to high-speed flow nature identification," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1076–1089, Aug. 2013.
[43] Wikipedia. (Jan. 2016). *YouTube, Quality and Foramt*. [Online]. Available: https://en.wikipedia.org/wiki/YouTube#Quality_and_codecs
[44] Y. Lim *et al.*, "Internet traffic classification demystified: On the sources of the discriminative power," in *Proc. Co-NEXT*, Philadelphia, PA, USA, 2010, p. 9.
[45] H. Kim *et al.*, "Internet traffic classification demystified: myths, caveats, and the best practices," in *Proc. CoNEXT*, Madrid, Spain, 2008, Art. no. 11.
[46] K. Wang, Y. Shao, L. Shu, Y. Zhang, and C. Zhu, "Mobile big data fault-tolerant processing for eHealth networks," *IEEE Netw.*, vol. 30, no. 1, pp. 1–7, Jan. 2016.
[47] J. J. Ramos-Munoz, J. Prados-Garzon, P. Ameigeiras, J. Navarro-Ortiz, and J. M. Lopez-Soler, "Characteristics of mobile youtube traffic," *IEEE Wireless Commun.*, vol. 21, no. 1, pp. 18–25, Feb. 2014.
[48] A. Moore, D. Zuev, and M. Crogan, "Discriminators for use in flow-based classification," M.S. thesis, Dept. Comput. Sci., Queen Mary Univ. London, London, U.K., 2005.
[49] M. A. Hall and L. A. Smith, "Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper," in *Proc. FLAIRS*, 1999, pp. 235–239.
[50] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.
[51] Telerik. (Jan. 2016). *Fiffler, the Free Web Debugging Proxy*. [Online]. Available: http://www.telerik.com/fiddler
[52] K. Wang, J. Yu, X. Liu, and S. Guo, "A pre-authentication approach to proxy re-encryption in big data context," *IEEE Trans. Big Data*, to be published.



WUBIN PAN is currently pursuing the Ph.D. degree with the Computer Science and Engineering School, Southeast University. His research interests include network security, network measurement, and traffic classification.

GUANG CHENG (SM'10) served as the Director of Computer Network Committee, Micro Computer Application Association, the Senior Member of Chinese Computer Federation, the Standing Committee Member of CCF TCI and Nanjing branch of China Computer Federation, and the Vice President of Jiangsu Software Talent Training Association and Computer Ethics and Occupation Training Committee. He is currently a Professor with the Computer Science and Engineering School, Southeast University; a Doctoral Tutor, a Doctor of Engineering, and the Director of the Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education; and the Secretary with the School of Computer Science and Engineering, School of Software, Southeast University.