

A Measurement Study on Skype Voice and Video Calls in LTE Networks on High Speed Rails

Li Li^{*†}, Ke Xu^{*†}, Dan Wang^{‡§}, Chunyi Peng[¶], Kai Zheng^{||}, Haiyang Wang^{**}, Rashid Mijumbi^{††} and Xiangxiang Wang^{*†}

^{*}Tsinghua National Laboratory for Information Science and Technology, Beijing, China

[†]Department of Computer Science and Technology, Tsinghua University, Beijing, China

[‡]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China

[§]The Hong Kong Polytechnic University Shenzhen Research Institute, Shenzhen, China

[¶]Department of Computer Science and Engineering, The Ohio State University, Columbus, USA

^{||}Central Software Institute, Huawei Technologies, Beijing, China.

^{**}University of Minnesota, Duluth, USA

^{††}Nokia Bell Labs, Dublin, Ireland

Email: ll-12@mails.tsinghua.edu.cn, xuke@tsinghua.edu.cn, csdwang@comp.polyu.edu.hk, chunyi@cse.ohio-state.edu, kai.zheng@huawei.com, haiyang@d.umn.edu, rashid.mijumbi@nokia.com and wx15@mails.tsinghua.edu.cn

Abstract—Recent advances in high speed rails (HSRs), coupled with user demands for communication on the move, are propelling the need for acceptable quality of communication services in high speed mobility scenarios. This calls for an evaluation of how well popular voice/video call applications, such as Skype, can perform in such scenarios. This paper presents the first comprehensive measurement study on Skype voice/video calls in LTE networks on HSRs with a peak speed of 310 km/h in China. We collected 50 GB of performance data, covering a total HSR distance of 39,900 km. We study various objective performance metrics (such as RTT, sending rate, call drop rate, etc.), as well as subjective metrics such as quality of experience of the calls. We also evaluate the efficiency of Skype's algorithms regarding the level of utilization of network resources. We observed that the quality of Skype calls degrades significantly on HSRs. Moreover, it was discovered that Skype significantly under-utilizes the network resources, such as available bandwidth. We discovered that the root of these inefficiencies is the poor adaptability of Skype in many aspects, including overlay routing, rate control, state update and call termination. These findings highlight the need to develop more adaptive voice/video call services for high speed mobility scenarios.

I. INTRODUCTION

In the last few years, there has been a significant worldwide progress in the development of high speed rail (HSR), reaching about 32,000 km at end of 2015. In this regard, China has played a more active role, contributing more than 60% of the world's HSR network in length. The current peak operating speed is 310 km/h. The passenger transport volume on HSRs in China was at least 1 Billion in 2016, and is growing at an annual rate of over 30%. Therefore, like never before, there is an increasing need for acceptable quality of communication services in high speed mobility scenarios.

However, there are questions on whether the current voice/video applications would be able to retain acceptable quality of experience (QoE) in such high speed environments. This is because, compared to low speed mobility, there are some particular inherent network characteristics (such as se-

rious fast fading due to Doppler frequency shift, extremely high handoff frequency, and repeated disconnections) in high speed mobility scenarios which could impact on the quality of voice/video calls, and pose severe challenges to key operations of online communication applications. It is therefore important to examine how well popular voice/video call applications can work on HSRs. If such applications cannot work well even with current HSRs, it would necessitate timely action before we face more severe challenges resulting from much faster trains in the foreseeable future.

This is the main motivation of our measurement study. Needless to remark, Skype is arguably the most popular audio/video communication platform for end users, with over 700 million worldwide users. Skype uses two transport layer protocols: TCP for control messages and UDP for voice/video transmission. To this end, we conduct the first comprehensive measurement study on Skype voice/video calls in LTE networks in high speed mobility scenarios. We use two large commercial cellular carriers in China, with a combined number of users of about 1 Billion nationwide. We refer to these as Carrier A and Carrier B.

The objective of this paper is to answer three key questions:

Q1: How well does Skype perform in high speed mobility environments? That is, does high speed motion have a significant impact on the quality of calls?

Q2: What is the efficiency of Skype's algorithms in such scenarios? That is, can it perform at a quality close to the best quality that the network would permit?

Q3: If Skype's algorithms are not efficient enough, what are the root causes?

Finding answers to these questions is technically challenging due to the following reasons:

1) For Q1, there are many factors (such as terrain, train speed, handoff, disconnection, etc.) in high speed mobility environments, which influence the performance of Skype. This makes it challenging to collect information for so many

factors, and analyze the independent impact of each of them. To overcome this, we have designed and developed a measurement tool, *MobiNet* [1], to collect information regarding the various factors, and an analysis tool, *Clearthink*, to quantitatively study the isolated impact of each factor.

TABLE I: List of measurement devices

Carrier	Device	Version (model)	RAT
A	Smartphone	Samsung Galaxy S4	FDD-LTE (4G)
	Smartphone	Huawei Honor 6	WCDMA (3G)
	USB Cellular Modem	Huawei EC3372-871	GSM (2G)
B	Smartphone	Samsung Galaxy j7	TD-LTE (4G)
	Smartphone	Huawei Mate 7	TD-SCDMA (3G)
	USB Cellular Modem	Huawei EC3276S-861	GSM (2G)

TABLE II: HSR lines

Route	Length (km)	Duration (h)	Number of trips	Static (h)	0-150 km/h (h)	150-280 km/h (h)	280-310 km/h (h)
B-G	2,298	9.7	10	16.9	24.5	19.5	46.3
B-S	1,318	5.8	8	11.2	10	8.4	24.4
B-T 1	115	0.6	20	17.1	2.6	2.0	6.4
B-T 2	515	3.0	8	9.6	5.2	4.0	13.3

2) With regard to Q2 and Q3, Skype is a proprietary software, and there is very limited public information about its encoding and transmission algorithms. The common practice is to treat it as a black-box and observe its behaviors under different conditions in a controlled testbed, where network bandwidth, packet losses and delay can be set to required values. However, our measurement is in a real, highly varying high speed mobile environment, making it extremely difficult to study Skype's algorithms. To overcome this, we employed two reference traffic patterns, the ping test and TCP transmission traffic, to probe the network capability. This way, we consider network parameters (connectivity, delay, packet loss rate, bandwidth) measured by ping and TCP as the baselines. We compare the metrics measured for Skype with the baselines to evaluate the efficiency of Skype's algorithms.

With these methods, we measure 3200 voice sessions and 3200 video sessions between a static user (User S) and a mobile user (User M). User S is at Tsinghua university in Beijing and User M is on a high speed train. We collected 50 GB of performance data, and covered a total HSR distance of 39,900 km. We make three main contributions:

1) We study the impact of high speed mobility environments on the quality of Skype calls. We study various objective performance metrics (such as RTT, sending rate, call setup time, call setup failure rate and call drop rate). We also study the subjective QoE in terms of Mean Opinion Score (MOS). We find that various objective performance metrics and subjective QoE all decline significantly, and vary in wide ranges in such scenarios. We also find that the QoE of User S and User M are quite different.

2) We evaluate the efficiency of Skype's algorithms. It is natural to attribute Skype's performance degradation to highly varying and poor network conditions in high speed mobility environments. However, our measurements show that the low efficiency of Skype's algorithms is also to blame. We have discovered various inefficiencies: 1) Long RTT of Skype (5 to 200 times that of ping 68% of the time), which indicates low efficiency of delay control. 2) Low utilization rate of available

bandwidth (below 20% in 75% of time), which implies low efficiency of rate control. 3) Even when network connectivity is good, call setup failures and call drops still occur frequently, which also indicates serious waste of network resources.

3) We find that the root cause of the various inefficiencies is poor adaptability of Skype's algorithms in such scenarios. First, overlay routing is highly dynamic when User M is moving fast, which leads to significant RTT increase and wide variation. Secondly, rate control is too conservative, refraining from fully utilizing all available bandwidth. Thirdly, Skype cannot update the state of User M timely, which makes it difficult for User S to set up a session with User M. Finally, Skype terminates sessions automatically when the sending rate is too low, which results in high call drop rates.

To summarize, we discover that the low efficiency of Skype's algorithms is the main cause of quality degradation of calls most of the time, rather than the highly varying and poor network conditions on HSRs. This means that there is still a gap between Skype's quality and the best quality that the network would permit.

The rest of this paper is structured as follows. Section II covers related work. Section III describes the measurement setup and Section IV presents our analysis methods. The quality of voice/video calls and efficiency of Skype algorithms are analyzed in Section V. We study the root causes of Skype's low efficiency in its key operations in Section VI before concluding the paper in Section VII.

II. RELATED WORK

A. Measurement Work on Voice Calls in Low Speed Mobility Environments

The impact of mobility on the quality of VoIP has been studied in a number of works. Salangam [2], and Phusamchot et al. [3] studied the impact of vehicular mobility on the quality of Skype calls in 3G networks, and found that, compared to stationary cases, mobility leads to QoE degradation. Tu et al. [4] performed a mobility test on the quality of VoIP for Hangouts in 4G LTE networks, and observed that call drop rates increase significantly when the user suffers handoffs. Wuttidittachotti et al. [5] studied the quality of Skype and Line in both 3G and 4G networks on vehicles, and found that while 4G does not provide better VoIP quality than 3G in static cases, it has a significant advantage for mobile users.

The above measurement studies have four main limitations: i) They only perform tests in low speed (< 100 km/h) mobility scenarios, without consideration for high speed (> 280 km/h) mobility cases. ii) They do not quantitatively analyze the independent impact of the factors (such as speed and handoff) which influence the quality of VoIP. iii) They do not evaluate the efficiency of VoIP applications regarding the level of utilization of network resources. iv) Although video calls are becoming very popular, previous studies do not study the performance of video calls in mobility cases.

TABLE III: Data set

Route	Data set: A-L-P			Data set: A-L-L			Data set: B-L-P			Data set: B-L-L		
	Number of voice calls	Number of video calls	Size (GB)	Number of voice calls	Number of video calls	Size (GB)	Number of voice calls	Number of video calls	Size (GB)	Number of voice calls	Number of video calls	Size (GB)
B-G	532	545	9.5	239	224	3.8	558	569	9.2	221	232	3.6
B-S	275	268	5.0	109	115	1.8	250	272	4.9	121	98	2.0
B-T 1	146	156	2.5	65	48	0.7	138	129	2.6	56	58	0.6
B-T 2	168	151	3.3	78	65	1.0	149	160	3.2	65	87	1.0

B. Measurement Work on TCP in High Speed Mobility Environments

With the deployment of HSRs, a critical demand is to understand network performance under extremely high-speed mobility. A number of studies [1], [6]–[8], have reported on the performance and behavior of TCP in such scenarios. However, many real-time applications (such as live streaming video, voice and video calls) use UDP as the transport layer protocol. These applications are expected to show different performance and behavior from TCP. It is therefore of great significance to explore how well real-time applications primarily based on UDP can perform in such high speed mobility environments.

III. MEASUREMENT SETUP AND DATA SET

A. Measurement Setup

We measure Skype voice and video calls between a static user (User S) and a mobile user (User M). User M is in a high speed train. User S uses Skype through a laptop running Windows 8.1, which connects to the Internet via a wired network at Tsinghua University in Beijing, with sufficient upload/download bandwidth for the Skype service. User M uses Skype through a laptop that has the same configuration as User S's laptop. User M's laptop accesses Internet through a USB Cellular Modem. In addition, User M also uses Skype through various smartphones. The laptops run Skype Version 7.17.99.105, while smartphones run Skype Version 6.15.99.1162 for Android. Table I shows all the mobile devices used in our experiments in detail. These devices are compatible with Radio Access Technology (RAT) ranging from 2G to 4G.

To emulate a voice call, we use a reference speech material from ITU (International Telecommunication Union) recommendations, and inject it into Skype using a virtual microphone e2esoft [9]. Similarly, to emulate a video call, we use a standard TV news video sequence "Akiyo" from JVT (Joint Video Team) test sequence pool. The sequence is mostly made up of head and shoulder movements, very similar to a video call scenario. We inject the video sequence into Skype using a virtual video camera Vcam [10]. This ensures the transmitted audio and video contents are consistent and repeatable. Most of the voice/video calls (99.9 %) are 2 minutes long. To observe the performance variation over a long time, we also measure 60 voice/video calls with longer durations ranging from 10 minutes to 30 minutes.

Data was collected in three ways: 1) Packet capture. We captured all the packets at the two ends of the communication using TCPdump for packet level analysis. 2) Skype technical information. Although Skype employs proprietary protocols, it reports some technical information (such as RTT, packet loss rate, sending rate, etc.) through its user interface for voice/video calls. We used a screen text capture tool textgrab

[11] installed on laptops to capture this information periodically from the Skype window. For the phones, screenshots were periodically taken and the necessary information was obtained from them using an OCR (Optical Character Recognition) tool. The sampling interval is 1 second. 3) Screen recording. We also recorded Skype calls using Camtasia Studio [12], and SCR Screen Recorder [13] on laptops and phones respectively, so that by replaying the videos, we can study events such as user login, call setup, call drop, etc., and analyze metrics including call setup time, call drop rate, and so on.

Finally, we also developed a measurement tool, *MobiNet*. The tool runs on the smartphones, and determines the geographical location and speed of the train via GPS, reads signal strength, network type, Location Area Code (LAC) and Cell ID (CID) of base stations from the Android OS. All this information is recorded in log files.

MobiNet can only run on Android (for smartphones), and it is impossible to develop a similar tool for Windows (for laptops) to collect various network parameters, because the USB cellular modems do not provide an interface to access the data. Therefore, the laptop-to-phone Skype calls are used to analyze the impact of the various network parameters. Besides, since virtual microphone and virtual camera tools that inject the source audio/video material to Skype can only run on Windows laptops, laptop-to-laptop calls are used to compare User S's and User M's QoE, where the two sides send the same audio/video materials repeatedly to each other.

B. Data Set

Experiments were conducted from December 2015 to July 2016 on 4 HSR lines: Beijing-Guangzhou (B-G), Beijing-Shanghai (B-S), Beijing-Tianjin (B-T 1) and Beijing-Taiyuan (B-T 2). Table II shows these HSR lines in detail. As an example, the length of the B-G line is 2,298 km. The duration of each one-way trip is 9.7 hours (excluding the parking time at the originating and terminal stations). We accumulate a mileage of 22,980 km in 10 one-way trips (2,298 km for each trip) on the line. Trains experience 4 phases of motion: parking at stations, acceleration, running at full speed and deceleration. In the 10 trips, the train parks at stations along the line for 16.9 hours, runs at speeds between 0 and 150 km/h, 150 and 280 km/h, and 280 and 310 km/h for 24.5, 19.5 and 46.3 hours respectively. In each one-way trip, we get on the train and start the measurement about 30 minutes before the train leaves the originating station, and stay on board for about 30 minutes after it arrives at the terminal to continue the measurement. Therefore, we have enough time to perform static measurements.

Table III shows details of the data set. According to the cellular carrier, and measurement devices, we divide data set

into four sub-data sets: A-L-P, A-L-L, B-L-P, B-L-L. As an example, sub-data set A-L-P contains data of laptop-to-phone Skype calls between User S and User M in networks of Carrier A, while A-L-L contains data of laptop-to-laptop Skype calls between the two sides in networks of Carrier A. For instance, we measure 532 laptop-to-phone voice calls and 545 laptop-to-phone video calls in networks of Carrier A on the B-G line, and the size of data collected is 9.5 GB, including captured packets, Skype technical information logs, screen recordings and log files of *MobiNet*.

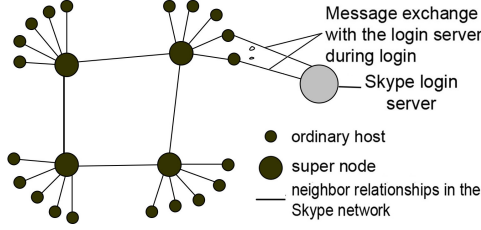


Fig. 1: Skype network architecture [14]

IV. ANALYTICAL METHOD

A. Analyzing the Impact of Each Influence Factor

There are four inherent influence factors in high speed mobility scenarios which have a significant impact on the performance of Skype. 1) Diverse terrain. Terrain along HSR routes is diverse, including plains, hills and tunnels, which highly affects characteristics of signal fading [15]. 2) Train speed. Due to Doppler frequency shift and fast multi-path fading, high speed can cause serious fast signal fading [16]. 3) Frequent handoff. High speed motion causes more frequent handoffs, resulting in sharp delays and consecutive packet losses. 4) Repeated disconnection. Mobile devices suffer repeated network disconnections, resulting in voice/video call drops.

Due to the above factors, providing acceptable quality of voice/video call services over IP in high speed mobility environments is very challenging. Since the four factors are intertwined, it is nearly infeasible to study the independent effect of each factor at the same time. This was one of the main challenges faced during our measurement study.

To overcome this, we develop an analysis tool, *Clearthink*. It uses the following algorithms:

To make the analysis feasible, *Clearthink* isolates the impact of terrain and repeated disconnections. This is achieved by using the information on geographical location (longitude and latitude) collected by *MobiNet*, from which we can know the terrain of areas crossed by the train through Google Earth. To avoid the interference from complicated signal fading due to diverse terrain on the analysis, *Clearthink* discards data collected when the train passes through hills, valleys, and tunnels, and hence use only the data collected in large areas of open plains. In addition, repeated disconnections can lead to call setup failures, call drops, and frequent online/offline state switches of clients, so we study the impact of disconnection alone when we discuss call setup, call drop, and online/offline state switches. However, even without the effects of diverse

terrain and repeated disconnections, it is still challenging to simultaneously analyze the impact of both speed and handoff, since they also depend on each other.

To overcome this challenge, *Clearthink* uses the total number of handoffs that a 2-minute call experiences to quantify the handoff frequency that a call suffers. Due to the variation of base station deployment density along the railway lines, *Clearthink* is able to achieve variation in handoff frequency suffered by a call. Besides, *Clearthink* makes use of variation in train speed (parking, acceleration, running at full speed, and deceleration) as shown in Table II to achieve variation in train speeds. Finally, *Clearthink* analyzes the impact of train speed and handoff as follows: i) Impact of train speed. *Clearthink* compares the performance among 2-minute calls that suffer no handoff when the train runs at various speed to study the effects of speed change alone. ii) Impact of Handoff. *Clearthink* makes a comparison among 2-minute calls that suffer different handoff frequency when the train runs at a relatively stable high speed (between 280 and 300 km/h) to quantitatively analyze the impact of handoff alone.

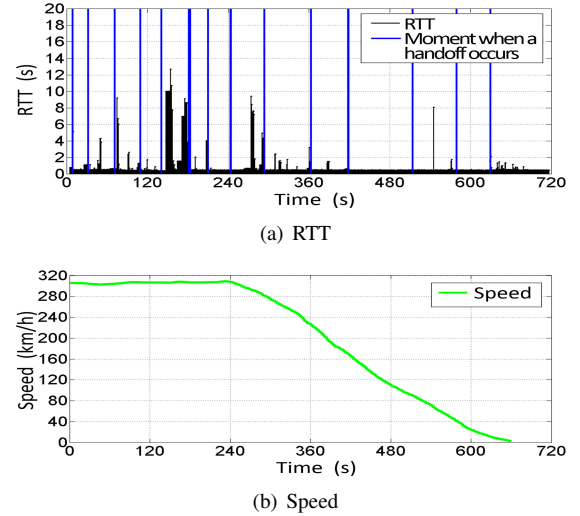


Fig. 2: RTT variation over time (B-L-P)

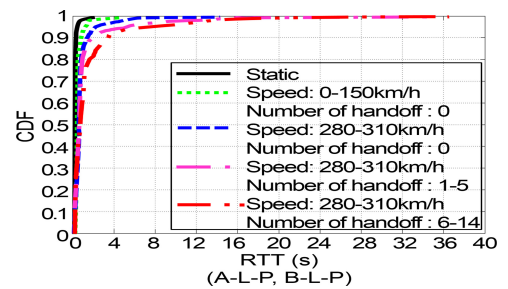


Fig. 3: CDF of Skype's RTT

B. Evaluating the Efficiency of Skype Algorithms

As shown in Figure 1 [14], a Skype network consists of three main components: a login server, ordinary nodes (Skype clients), and super nodes. The login server is the only central component in the network. It stores Skype user names, passwords, and contact lists. During the login process, a Skype

client authenticates the user name and password with the login server, advertises its presence to other peers and its contacts.

Excluding the login server, Skype uses a super node-based hierarchical peer-to-peer network, with two layers: super nodes, and ordinary nodes. This network architecture allows Skype clients to use the network of super nodes to find a specific Skype user. Super nodes maintain an overlay network among themselves. They also function as ordinary nodes and are elected from amongst them. Ordinary nodes must pick one (or a small number of) super nodes to associate with, and they issue queries, send control traffic including availability information, requests for voice and video sessions through the super node(s) they are associated with. Skype uses two transport layer protocols: TCP for control messages and UDP for voice/video transmission. The TCP connection acts as a feedback channel through which the receiver periodically reports current network conditions to the sender. Then, the sender adapts its UDP sending rate to network conditions.

Skype uses proprietary protocols and encrypts data and signaling messages. Therefore, it is difficult to analyze its key operations. The common practice is to treat it as a black-box and observe its behavior under different conditions. For example, Zhang et al. [17] observed how Skype adjusts its rates, FEC redundancy and video quality, by varying packet loss rate, propagation delay and bandwidth in a controlled network testbed. However, our experiment is based on a real high speed mobility environment, where network conditions are highly diverse and time-varying. It is impossible to come up with a set of scenarios in a controlled testbed, which are representative of the complicated network conditions on HSRs.

To overcome this, we use two types of reference traffic: (1) ping test, and (2) TCP transmissions. User S's laptop has a public IP address, and we ping User S from User M to probe network connectivity, packet loss rate and delay, and transmit TCP traffic in a client/server mode by *iperf* [18] to estimate available bandwidth. We use the network parameters measured by ping and TCP as baselines. We compare the baselines with parameters obtained while using Skype. In this way, we are able to evaluate the efficiency of Skype's algorithms regarding the level of utilization of network resources.

It is worth remarking that due to Skype's overlay routing in the super node-based P2P networks, there are big differences between the paths of ping and Skype transmissions, and between TCP and Skype transmissions. However, the network bottleneck in their paths (which is the last hop from the base station to User M) is the same. The network conditions of other parts of their paths are expected to be reliable, but the last hop suffers serious fast fading, frequent handoff and repeated disconnections, due to high speed mobility. Hence, even with different paths, for the same bottleneck, the comparison between ping and Skype, and between TCP and Skype is meaningful. Besides, since TCP performance also degrades in high speed mobility scenarios [1], [6]–[8], we cannot use it to estimate the accurate bandwidth. However, we are able to conclude that the network bandwidth is at least the sending rate of TCP.

Since ping test traffic is light, it brings minimal competition to Skype traffic. Therefore, we measure Skype and ping tests simultaneously. However, the impact of TCP cannot be ignored. To avoid competition between TCP and Skype traffic, we do not measure them simultaneously. For sake of fairness, we measure Skype and TCP independently at the same time of the same weekday along the same HSR routes repeatedly, because volumes of other network traffic is comparable in this way. We evaluate the efficiency of Skype's algorithms in the following ways:

i) We compare the RTT of a Skype call with that measured by the ping test simultaneously. If the former is comparable with the latter in static and low speed mobility cases, but is much longer in high speed mobility cases, we can decide that delay management in Skype becomes inefficient with increase in mobility speeds.

ii) We compare the sending rate of Skype calls with that of TCP. If the former is much lower than the latter, we can attribute the low sending rate to inefficient Skype rate control rather than insufficient network bandwidth.

iii) During call setup, we monitor the connectivity and packet loss rate of the network through ping tests, so that we can decide whether to attribute a call setup failure to network disconnection, heavy losses of handshakes or inner problems of Skype protocols.

iv) We compare Skype's call drop rate and network disconnection rate measured by ping tests. If the former is higher than the latter, we can attribute call drops to inner problems of the Skype protocols rather than network disconnection.

V. SKYPE PERFORMANCE AND EFFICIENCY ANALYSIS

In this Section, we first analyze various objective quality metrics (including RTT, sending rate, call setup time, call setup failure rate, and call drop rate) of voice/video calls. We then analyze the subjective QoE of voice/video calls.

We make extensive vertical and horizontal comparisons: i) Vertical comparison is based on the methods presented in Section IV-A, and are used to investigate the effect of high speed mobility environments on Skype calls. Comparisons are made for calls in static, low speed and high speed mobility situations under various handoff frequencies. ii) Horizontal comparison is based on the methods presented in Section IV-B, and is used to evaluate the efficiency of Skype's algorithms. Comparisons are made for the same performance metric between Skype calls and ping test, and between Skype calls and TCP flows, under the same train speed and handoff frequency. We make these vertical and horizontal comparisons mainly based on laptop-to-phone calls because, this way, we are able to get the handoff frequency of phones using *MobiNet*.

A. RTT

Figure 2(a) shows the variation of RTT over time of a 12-minute video call during train deceleration (from 300km/h to 0km/h) on the B-G line. We mark the moments when handoffs occur in Figure 2(a). Figure 2(b) depicts the variation of speed in the period. We find that when the speed is below 150 km/h

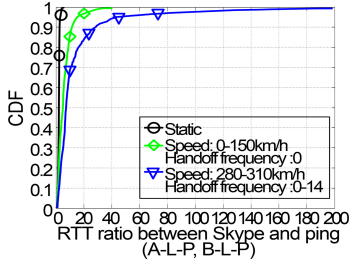


Fig. 4: CDF of the RTT ratio between skype and ping

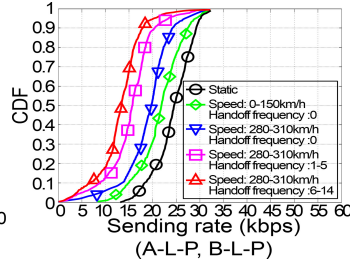


Fig. 5: CDF of User S's sending rate (voice)

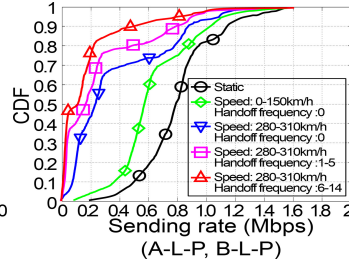


Fig. 6: CDF of User S's sending rate (video)

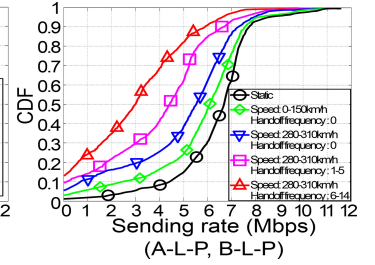


Fig. 7: CDF of sending rate of TCP (downlink)

(i.e. from 450 to 720 s on the horizontal axis), the RTT is short and stable with light variation. However, when the speed is over 280km/h (i.e. from 0 to 290 s), the RTT is very long and varies in a very wide range. We also find that RTT spikes often appear at times around handoff events. This is an example of how high speed mobility environments impact Skype's RTT. Figure 3 shows the CDF of RTT of 2-minute voice and video calls under various speeds and handoff frequencies. It can be observed that both speed and handoff have a significant impact on RTT. Moreover, compared with speed, handoff contributes more to RTT increases.

Although the overlay routing used by Skype is different from the IP routing used by ping, the two paths share the same bottleneck, the last hop (from the base station to User M). The network conditions of other parts of their paths are expected to be good, but the last hop suffers serious fast fading and frequent handoff, due to high speed mobility. Therefore, the increase in RTT can be mainly attributed to the last hop. This means that the increasing range of RTT for both ping and Skype should be comparable.

However, the result of the measurements is very unexpected. In Figure 4, we present the ratio of the RTT measured by 2-minute Skype calls to that measured by 2-minute ping tests at the same time. We observe that in static cases, the ratio is always below 2. This indicates that the latency of Skype's overlay routing is at most twice that of IP network routing used by ping. However, when the speed is over 280 km/h, the RTT of Skype is between 5 and 200 times as long as that of ping, with a probability of 68%. This reveals that the efficiency of Skype's overlay routing decreases in high speed mobility cases. We study the root cause for this issue in Section VI-A.

B. Sending Rate

We find that the impact of high speed mobility environments on downlink (User S to User M) and uplink (User M to User S) transmission is qualitatively consistent. For brevity, we only show the results of the downlink sending rate of 2-minute voice and video calls in Figures 5 and 6 respectively. It can be observed that both speed and handoff have a significant impact on the sending rate of Skype calls. As train speed and handoff frequency increase, the sending rate decreases sharply. Moreover, handoff contributes more to the decrease of sending rate than speed. We also observe that sending rate degradation for video calls is more significant than that for voice calls.

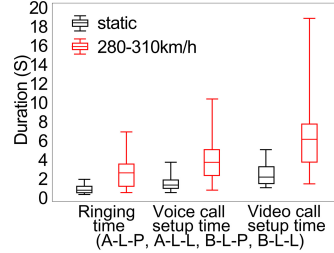


Fig. 8: Call setup time

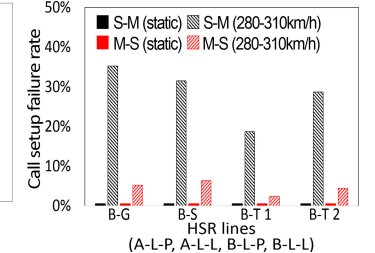


Fig. 9: Call setup failure rate

The natural question is on whether the low sending rate is due to low bandwidth in high speed environments, or due to poor rate control where Skype fails to make good use of available bandwidth. To answer this question, we measure TCP's sending rate to probe the bandwidth of the network.

Figure 7 shows downlink sending rate of 2-minute TCP flows. We observe that the sending rate drops to 0 with a considerable probability. This is due to TCP's congestion control and exponential back-offs during serial retransmissions [8]. In spite of that, the sending rate is generally much higher than that for Skype calls. For example, as shown in Figures 6 and 7, when the train speed is over 280km/h, and the handoff frequency is above 6 (every 2 minutes), the sending rate of TCP is over 1 Mbps 75% of time (implying that available bandwidth is at least 1 Mbps in 75% of time), but the sending rate of Skype video calls is below 200 kbps 75% of time. This represents a utilization rate no more than 20% most of the time.

To achieve the best quality of service, the ideal sending rate of Skype voice and video calls are around 30 kbps and 1 Mbps respectively. The sending rate of TCP implies that the available network bandwidth is sufficient for voice/video calls in most time, but Skype cannot make good use of it. It means that Skype performs at a quality much lower than the best that the network would permit. We conclude that the efficiency of Skype's rate control is seriously low in high speed mobility environments. We analyze the root cause for this problem in Section VI-B.

C. Call Setup and Drop

We use three metrics to analyze call setup time: ringing time, voice call setup time, and video call setup time. Ringing time is the duration from the time the caller dials to when the receiver's Skype client rings. We define voice call setup time as the duration from the time the caller dials to when he/she

hears the receiver's voice. Video call setup time is the duration from the time the caller dials to when he/she sees the receiver's video. We have developed a program to automatically accept a session request timely, such that the duration from the time the receiver's Skype rings to when the receiver accepts the request is negligible.

Figure 8 depicts the three call setup time metrics. We analyze two scenarios: i) when the train is parking (denoted by static) and ii) when the train is running at full speed (denoted by 280-310km/h). We make following findings: Compared with static cases, the three metrics all rise significantly and vary in a wide range in high speed mobility scenarios. Among the three metrics, increase in video call setup time is the most significant. When the speed is over 280 km/h, the average video call setup time is 6 s, almost 3 times longer than that in static cases. The increase of the three metrics can be explained by the rise in RTT and drop in sending rate.

We also study call setup failure rate and call drop rate. The former is defined as the probability that a call cannot be established even after 10 times of redialing. The latter is the probability that a call is interrupted before it is finished. Figures 9 and 10 show call setup failure rate and drop rate respectively. S-M in Figure 9 denotes that User S is the caller and M the receiver, while M-S denotes that User M is the caller and S the receiver. We make following observations:

1) In static cases, the call setup failure rate is almost 0 on various HSR lines. When the train is running at full speed, we make an unexpected observation: the rate rises slightly when User M is the caller, but rises sharply when User S is the caller. For example, when User S is the caller, setup failure rate is 34% on the B-G line, almost 6 times as high as that when User M is the caller.

2) Call drop rate is almost 0 in static cases, but rises significantly when the train runs at full speed. As shown in Figure 10, call drop rate is 8.3%, 9.1%, 7.2% and 10.6% on the 4 HSR lines respectively.

To explore the causes of the extremely high failure rate for calls initiated by User S, we probe network connectivity and packet loss rate by a ping test when User S is dialing User M. We check network connectivity at the moment when a call drop occurs using a ping test. The results are unexpected:

1) We observe that network disconnections for User M contribute to 53.8% of call setup failures for User S. This is quite strange since S only dials M when it finds User M online through the Skype window. However, we find that even when User M is offline due to network disconnections, this is not immediately reflected on S's Skype window. Therefore, S dials User M who appears to be online, leading to a call setup failure.

2) We also observe that 46.2% of setup failures happen when the network connectivity is good and packet loss rate is below 5%. That is to say, even though User M does not suffer network disconnection, and he/she is actually online, User S may still not establish a session with M after many times of redialing. This is not very clear what the cause for this behavior is.

3) We find that 71.5% of call drops are due to network disconnections, while 28.5% occur even when the network connectivity is good, which is also requires further investigation.

All these observations indicate that network disconnections are not the only cause of call setup failures and call drops. Even when network connectivity is good, call setup failures and call drops still occur frequently, which also indicates serious waste of network resources. It is possible that some bugs in Skype algorithms are responsible for these problems. We study the root cause for these issues in Section VI-C and VI-D.

D. Quality of Experience

We use the WF-Regression model [19] to analyze the subjective QoE of Skype voice calls. The model can be used for super wide band calls, which is a characteristic of Skype's codec, SILK, and outperforms the widely-used PESQ model [20] in modelling the quality of Skype voice calls. The videophone subjective quality model, also known as the opinion model for video-telephony applications, has been standardized as ITUT Recommendation G.1070 [21]. We use a subjective quality model [22] to estimate the effect of coding distortion and frame reduction on video quality. The various coefficients of the two models were set according to their recommendations for Skype voice/video calls. Both models use 5-point MOS to quantify subjective QoE, 5 for excellent, 4 for good, 3 for fair, 2 for poor and 1 for bad QoE. We evaluate the QoE of Users S and M independently according to the quality of voice/video contents received by them through Skype.

We found that the impact of high speed mobility environments is qualitatively consistent on User S's and User M's QoE. Therefore, for brevity, we only show User S's QoE under various speed and handoff frequency during 2-minute calls in Figure 11.

We observe that the MOS of both voice and video calls degrades significantly and varies in a much wider range, as the train's speed and handoff frequency increase. Video calls suffer more significant degradation than voice calls. For example, when the speed is over 280km/h, and the handoff frequency is over 6, the average MOS of voice calls is 3.8 (i.e. fair), which is 15.6% lower than that in static cases. On the other hand, the average MOS of video calls is 2.8 (i.e. poor) under the same train speed and handoff frequency, which is 33.3% lower than that in static cases.

Finally, we also compare User S's and User M's QoE. As explained in Section III-A, for fairness, we choose to use laptop-to-laptop Skype calls to make the comparison, where the two sides send the same source audio/video materials repeatedly to each other.

Figure 12 shows the CDF of the ratio of User M's MOS to User S's MOS at the same time. We find that the CDF of the MOS for both users is roughly consistent, with a slight quantitative difference in a long time duration. However, the difference between the two sides at the same time rises sharply

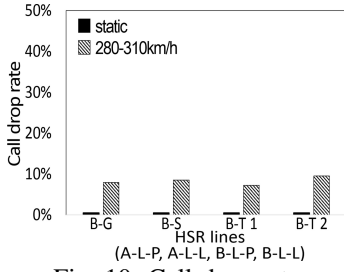


Fig. 10: Call drop rate

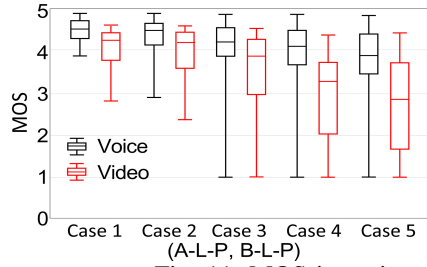


Fig. 11: MOS in various scenarios

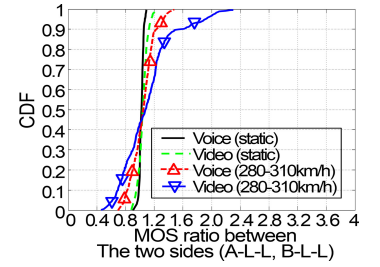


Fig. 12: CDF of the MOS ratio between the two sides

in high speed mobility cases. As shown in Figure 12, in static cases, the ratio is around 1 most of the time, which indicates that the QoE of the two sides is comparable. However, when the train is running at full speed, the ratio varies within a wider range, especially for video calls. As an example, for video calls, the ratio drops below 0.6 with a probability of 7%, and surpasses 1.6 with a probability of 10%. That is to say, the QoE of the two sides are quite different even during the same call. For example, although User S sees that the quality of M's video transmitted by Skype is good ($MOS > 4$), User M may see that the quality of User S's video is poor ($MOS < 3$) at the same time. This can be explained by the highly dynamic and different uplink and downlink conditions in high speed mobility environments.

VI. ROOT CAUSES ANALYSIS

In the previous Section, we observed that Skype's performance is much lower than the best quality that the network would permit. In this Section, we study the root causes for its low efficiency from four aspects: overlay routing, rate control, state update and call termination.

A. Highly Varying Overlay Routing

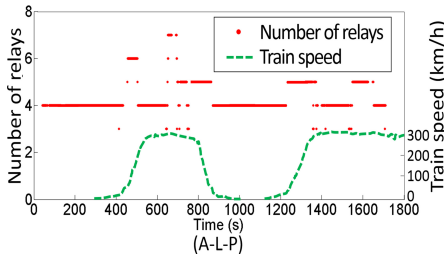


Fig. 13: Change in the number of relays over time

Skype uses application layer routing to transmit voice/video data through the super node-based P2P overlay networks. In Section V-A, we found that Skype's RTT is, at most, twice that of ping in static cases, but 5 to 200 times that of ping when speed is over 280km/h 68% of the time. This indicates that the adaptability of Skype's overlay routing is much worse than the IP network routing used by ping in high speed mobility environments.

We use the number of relays to study the issue. Skype often uses relay nodes to establish an indirect voice/video call connection [14]. We collected real-time information on the number of relays through Skype's technical reports.

We observed that high speed mobility has a significant impact on Skype's relay scheme. Figure 13 shows change in the number of relays of a 30-minute video call observed in Carrier A's network on B-S line. When the train is parking between 0 and 300 s, and between 1005 and 1125 s, the number of relays is constant at 4. However, the number shows significant variation during acceleration. When the speed is over 280 km/h (between 650 and 750 s), the number of relays increases significantly and varies rapidly ranging from 3 to 7. This reveals that the path between the two ends through the super node-based P2P networks is highly dynamic during the high speed movement of User M. The increase in number of relays can lead to sharp RTT rise, while the rapid and wide-range variation in the number of relays can lead to rapid and wide-ranged RTT variation. This can explain why the RTT of Skype rises more significantly and shows much severer variation than that of ping in high speed mobility cases.

B. Poor Rate Control

In Section V-B, we found that the bandwidth probed by TCP is over 1 Mbps most of the time when the train is running at full speed, which is sufficient for acceptable voice/video call services. However, the sending rate of Skype is very low with a bandwidth utilization rate below 20% most time, which means that the rate control is poor in high speed mobility scenarios.

Since Skype uses UDP to transmit voice/video data, there is no transport layer congestion control. Skype conducts a rate control in the application layer [17]. In static and low speed mobility cases, network bandwidth is adequate and relatively stable most of the time. Skype's rate control works well in this case. However, network bandwidth is highly varying, and packet loss bursts occur repeatedly in high speed mobility environments. The rate control algorithm cannot work well in following ways:

1) Skype is more sluggish than TCP to keep pace with the rapid bandwidth variation in high speed mobility environments. Cicco et al. [23] also make consistent findings in a controlled testbed. They measured the responsiveness of Skype video calls to bandwidth variations, and found that Skype's response time to bandwidth increase is long. Therefore, it cannot adapt fast enough with the highly varying bandwidth, leading to under-utilization of bandwidth.

2) Skype refrains from fully utilizing all available bandwidth, which means that a Skype call is not performed at the best quality that a network would permit. Cicco et al. [24] also draw a similar conclusion in a controlled testbed. If

the bandwidth is adequate, for example, 5 Mbps, 20% of the bandwidth is enough to provide acceptable video call services. If the bandwidth is only 1 Mbps, it would be wise to use most of the bandwidth to achieve good quality of service. However, Skype always limits its sending rate conservatively no matter how wide the bandwidth is. By contrast, TCP is more ambitious to occupy all the bandwidth: if no packet drop occurs, it will increase its sending window as big as possible. Therefore, Skype cannot make use of bandwidth as good as TCP.

3) Skype cannot work well under heavy packet losses. We find that the probability that the sending rate drops below 10 kbps is 70% when packet loss rate is between 10% and 20%, and reaches as high as 90% when the packet loss rate is over 20%. Such a low sending rate is not enough for voice calls, not to mention video calls. We conclude that Skype cannot work under heavy packet losses. Zhang et al. [17] also make consistent findings in a controlled testbed. They find that when the packet loss rate is larger than 10%, even though there is abundant bandwidth available on the path, Skype still drops its sending rate to the lowest rate. Unfortunately, users suffer packet loss bursts from time to time on HSRs. Therefore, Skype's simple and aggressive rate control under heavy packet losses is definitely fatal to its performance.

C. Sluggish State Update

We observed two types of abnormal call setup failures in Section V-C:

1) When User M (in a high speed train) is offline due to network disconnections, he/she still appears to be online on User S's Skype window. Therefore, User S dials User M, leading to a call setup failure.

2) Even though User M does not suffer network disconnection, and he is really online, User S may still not establish a session with M after many times of redialing.

The abnormal call setup failures are caused by problems in state update. Although it is easy for a caller to find a receiver and establish a session in static and low speed mobility cases, it may be challenging to establish a call in high speed mobility cases. When the caller is static, and the receiver is in high speed mobility, the receiver's location in the super node-based P2P networks changes rapidly. Moreover, the receiver may suffer frequent online/offline state switches due to repeated network disconnections. This requires Skype to update the receiver's location and state accurately and rapidly. However, the two types of abnormal call setup failures implies that the state update is too sluggish. That is to say, the static caller sometimes cannot obtain the latest location and online/offline state of the fast moving receiver timely. We explain this in detail as following:

1) Spurious online/offline states. Due to repeated disconnections, User M is abnormally forced to be offline for a network disconnection and logs in again when he/she reconnects to the network. Hence, User M suffers frequent online/offline state switches on HSRs. However, User S cannot update the latest state of User M timely, so that User M is spuriously

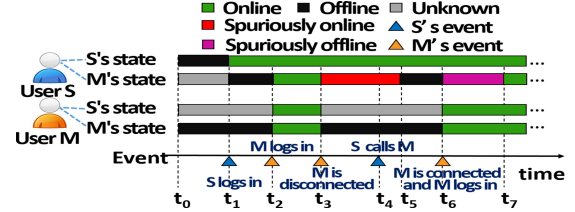


Fig. 14: Spurious online/offline states

online/offline for a considerable time in the view of User S. Figure 14 gives an example. There are three states of a User: online, offline and unknown. For instance, when User M is online, he/she can see that he/she is online, and he/she also can see the online/offline state of User S from his/her Skype window. When User M is offline, then he/she can know that he/she is offline, but does not know the state of User S, hence, the state of User S is unknown in his/her view. We show the state of the two sides in the view of User S and User M respectively. As shown in the figure, User S logs in at t_1 . When User M logs in at t_2 , User S notices that the state of M turns from offline to online immediately. Similarly, M notices that S is online at t_2 . User M keeps online until t_3 , when he/she suffers a network disconnection. Unfortunately, S cannot notice that User M is forced to be offline immediately, and the state of M shown in S's Skype window is still online. After multiple times of redialing between t_4 and t_5 , the state of M is updated, turning from online to offline in S's Skype window at t_5 . User M reconnects with the network at t_6 and logs in again. However, S cannot update the state of M timely and M is spuriously offline in the view of S until t_7 . Through measurement, we find that if S does not dial M, User M can keep spuriously online in the view of S for a time ranging from several seconds to dozens of minutes. Similarly M can keep spuriously offline in the view of S for a period ranging from a few seconds to several minutes.

2) Difficulty in call setup. When the Skype call is initiated by User S, it may be very hard to send a session request to User M. This is because even though User M is offline, it is spuriously online for a considerable time in the view of User S. When User S sends a session request in this case, User M cannot receive the request. In addition, even when User M is actually online, S may not locate him/her, and a session request may not reach M. By contrast, S is static, and always keeps online, so it is much easier for User M to locate him/her, and send a session request successfully. As a result, when the train is running at full speed, the call setup failure rate rises slightly when User M is the caller, but rises sharply when User S is the caller.

D. Hasty Call Termination

In Section V-C, we found that even though the network is not disconnected, Skype still suffers call drops. In this case, Users S and M can ping each other successfully, but cannot continue the Skype session.

To seek the cause, we analyze what happens before these unexpected call drops. We analyze Skype's packet loss rate and sending rate in the 30 s duration just before the call drop

occurs. We find that most of these calls suffer heavy packet losses and sharp sending rate decreases before the call drop happens. This reveals that Skype ends a call when the sending rate is too low, which is consistent with findings in [25].

This call termination scheme may not be suitable for high speed mobility scenarios. Skype suffers packet loss bursts from time to time on HSRs, and reduces the sending rate to the lowest level under heavy packet drops. Actually, the network can recover quickly from packet loss bursts most of the time. It is therefore not proper to terminate the session each time the sending rate is too low. Users may be more tolerant of temporary performance degradation than a call drop.

VII. CONCLUSION

In this paper, we have presented a comprehensive measurement study on Skype voice/video calls in LTE networks on HSRs with a peak speed of 310km/h in China. Over an 8-month period, we collected 50 GB of performance data along various HSR routes totaling 39,900 km.

We developed a measurement tool, *MobiNet*, to collect information of various influence factors, and an analysis tool, *Clearthink*, to study how each factor impacts Skype. Moreover, since Skype is a proprietary software, we turned to a black-box testing approach in which we probed its functionality and network parameters using two reference flows (ping and TCP).

We have found that high speed mobility environments have a significantly negative impact on Skype. Various objective performance metrics (such as RTT, sending rate, call setup time, etc.) and subjective parameters such as QoE all decline significantly and vary in a wide range in high speed mobility scenarios. We also discovered various inefficiencies in Skype. There is quite a big gap between Skype's quality and the best quality that the network would permit.

While analyzing the cause of such inefficiency, we have found that Skype encounters serious problems in high speed mobility, including highly varying overlay routing, poor rate control, sluggish state update and hasty call termination. As a result, Skype performs much worse than the best quality that the network would permit.

As the need for high speed travel continues to increase, our findings lead to a call for action to develop more adaptive voice and video calling applications in high speed mobile environments. It is our hope that our research would be of good reference value for application developers to enhance their high speed mobility support.

ACKNOWLEDGMENT

This work was supported in part by the Huawei Innovation Research Program (HIRP), the National Natural Science Foundation of China under Grants 61170292 and 61472212, the National Science and Technology Major Project of China under Grant 2015ZX03003004, the National High Technology Research and Development Program of China (863 Program) under Grants 2013AA013302 and 2015AA015601, the EU Marie Curie Actions CROWN under Grant FP7-PEOPLE-2013-IRSES-610524, Grant PolyU 1-BBYC, the funding support from the Innovation and Technology Commission of the

HKSAR to the Hong Kong Branch of National Rail Transit Electrification and Automation Engineering Technology Research Center, the National Science Foundation under Grants CNS-1421440 and CNS-1526456, Chancellor's Small Grant and Grant-in-aid program from the University of Minnesota.

REFERENCES

- [1] Q. Xiao, K. Xu, D. Wang, L. Li, and Y. Zhong, "TCP Performance over Mobile Networks in High-speed Mobility Scenarios," in *Proceedings of IEEE ICNP*, 2014.
- [2] W. Salangam, "Comparison of VoIP Quality between Skype and LINE over 3G and 4G Networks: A Case Study of BTS - Sukhumvit Line." in *Master Thesis, KMUTNB, Bangkok, Thailand*, Dec. 2014.
- [3] P. Phusamchot and P. Wuttidittachotti, "Comparison of Skype VoIP Quality over 3G with Mobility: A Case Study of Fair Usage Policy Effects." in *Proceedings of IEEE xyz*, 2015.
- [4] G.-H. Tu, C.-Y. Li, C. Peng, Z. Yuan, Y. Li, X. Zhao, and S. Lu, "Volte*: A lightweight voice solution to 4g lte networks," in *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*. ACM, 2016, pp. 3-8.
- [5] P. Wuttidittachotti and T. Daengsi, "QoE of Social Network Applications: A Study of VoIP Quality from Skype vs LINE over 3G and 4G," in *Proceedings of IEEE ICUFN*, 2015.
- [6] Q. Liu, K. Xu, H. Wang, M. Shen, L. Li, and Q. Xiao, "Measurement, Modeling, and Analysis of TCP in High-Speed Mobility Scenarios," in *Proceedings of IEEE ICDCS*, 2016.
- [7] L. Li, K. Xu, D. Wang, C. Peng, K. Zheng, R. Mijumbi, and Q. Xiao, "A Longitudinal Measurement Study of TCP Performance and Behavior in 3G/4G Networks over High Speed Rails," *IEEE/ACM Transactions on Networking*, 2017.
- [8] L. Li, K. Xu, D. Wang, C. Peng, Q. Xiao, and R. Mijumbi, "A Measurement Study on TCP Behaviors in HSPA+ Networks on High-speed Rails," in *Proceedings of IEEE INFOCOM*, 2015.
- [9] "E2esoft," <http://e2esoft-vsc.en.softonic.com>.
- [10] "Vcam," <http://vcam.en.softonic.com/>.
- [11] "Textgrab," <http://textgrab-sdk.renovation-software.qarchive.org/>.
- [12] "Camtasia studio," <http://camtasia-studio.en.softonic.com/>.
- [13] "Scr screen recorder," <http://www.scr-screen-recorder.com/>.
- [14] S. A. Baset and H. G. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol," in *Proceedings of IEEE INFOCOM*, 2006.
- [15] F. Luan, Y. Zhang, L. Xiao, C. Zhou, and S. Zhou, "Fading characteristics of wireless channel on high-speed railway in hilly terrain scenario," *International Journal of Antennas and Propagation*, vol. 2013, 2013.
- [16] Y. YANG and P. FAN, "Doppler frequency offset estimation and diversity reception scheme of high-speed railway with multiple antennas on separated carriage," *Journal of Modern Transportation*, vol. 20, no. 4, pp. 227-233, 2012.
- [17] X. Zhang, Y. Xu, H. Hu, Y. Liu, Z. Guo, and Y. Wang, "Profiling Skype video calls: Rate control and video quality," in *Proceedings of IEEE INFOCOM*, 2012.
- [18] "Iperf," <https://iperf.fr/>.
- [19] C. Chen, C. Chu, S. Yeh, H. Chu, and P. Huang, "Measuring the Perceptual Quality of Skype Sources," in *ACM SIGCOMM Computer Communication Review*, 2012.
- [20] ITU, "ITU-T Recommendation P.862: Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs." in *ITU, Tech. Rep.*, 2001.
- [21] ITU-M, "ITU-T Recommendation G.1070: Opinion model for video-telephony applications," in *ITU, Tech. Rep.*, 2007.
- [22] K. Yamagishi and T. Hayashi, "Qrp08-1: Opinion model for estimating video quality of videophone services," in *IEEE Global Telecommunications Conference*, 2006.
- [23] L. De Cicco, S. Mascolo, and V. Palmisano, "Skype Video congestion control: An experimental investigation," *Computer Networks*, vol. 55, no. 3, pp. 558-571, 2011.
- [24] —, "Skype Video Responsiveness to Bandwidth Variations," in *Proceedings of ACM NOSSDAV*, 2008.
- [25] Y. Xu, C. Yu, J. Li, and Y. Liu, "Video telephony for end-consumers: measurement study of Google+, iChat, and Skype," in *Proceedings of ACM IMC*, 2012.