Modeling and Analysis of Skype Video Calls: Rate Control and Video Quality

Xinggong Zhang, *Member, IEEE*, Yang Xu, *Student Member, IEEE*, Hao Hu, Yong Liu, *Member, IEEE*, Zongming Guo, *Member, IEEE*, and Yao Wang, *Fellow, IEEE*

***Abstract*—Video-conferencing has recently gained its momentum and is widely adopted by end-consumers. But there have been very few studies on the network impacts of video calls and the user Quality-of-Experience (QoE) under different network conditions. In this paper, we study the rate control and video quality of Skype video call, and analyze the network impacts in large-scale networks. We first measure the behaviors of Skype video call on a controlled network testbed. By varying packet loss rate, propagation delay and available network bandwidth, we observe how Skype adjusts its sending rate, FEC redundancy, video rate and frame rate. It is found that Skype is robust against mild packet losses and propagation delays, and can efficiently utilize the available network bandwidth. We also find that it employs an overly aggressive FEC protection strategy. Based on the measurement results, we develop rate control model, FEC model, and video quality model for Skype video calls. Extrapolating from the models, we conduct numerical analysis to study the network impacts. We demonstrate that user back-offs upon quality degradation serve as an effective user-level rate control scheme. We also show that Skype video calls are indeed *TCP-friendly* and respond to congestion quickly when the network is overloaded. Through a case study of a 4G wireless network, we demonstrate that the proposed models can be used in user-QoE-aware network provisioning.**

***Index Terms*—Rate control, Skype, user QoE aware, video conferencing, video quality.**

I. Introduction

THE Internet has fundamentally changed the way people communicate. While emails, text-messages, and Voice-over-IP (VoIP) calls will remain to be important communication means, we are at the threshold to embrace the next big change: *Video Conferencing*. Most online chatting software, such as MSN messenger, GTalk and Facetime, support peer-to-peer video chat. Skype [1], arguably the most popular audio/video communication platform for end-consumers, provides two-way video calls and multi-party video conferencing to its 500 million users on PCs or mobile devices. The fact that many distributed video calls are offered free-of-charge further fueled the wide adoption of video conferencing among them.

Video calls are much more bandwidth-demanding than VoIP calls. While the data rate of a typical Skype VoIP call is around 30 kbps [2], a good quality Skype video call can easily consume bandwidth up to 950 kbps, representing an increase by a factor of more than thirty. The sheer traffic volumes and dynamic overlay topology make it imperative for network providers and network researchers to understand the impact of this new "killer application" on the performance and stability of the existing network protocols and infrastructures. Meanwhile, due to the real-time interaction between users, the quality of a video call is more sensitive to packet losses and delays than one-way video streaming. It is therefore of great interest for end users and application developers to assess the quality of video calls under different network conditions. However, up to now, there have been very few study on the popular video telephony applications.

In this paper, we present our recent effort on profiling Skype video calls' rate control and video quality through empirical measurement and analytical modeling. We attempt to answer three key questions:

Q1: How does a video call adapt its sending rate, video rate and quality under different network conditions?

Q2: Are the video calls friendly to TCP flows when they compete for network resources?

Q3: How should network providers provision their networks to support multiple video calls with satisfactory quality?

It is challenging to come up with comprehensive answers for those questions. First of all, Skype is a proprietary software. There is very limited public information about its video encoding and transmission algorithms. The common practice is to treat it as a black-box and observe its behaviors under different conditions. Secondly, It employs a distributed peer-to-peer overlay for communications [3], and all messages are encrypted. It is hard to reverse-engineer its protocols. Thirdly, real network conditions are highly diverse, often time-varying and traffic-varying. It is impossible to come up with a set of scenarios that are representative for all networks. Measuring a video call in a fast changing network environment often gives misleading conclusions.

To address those challenges, we answer *Q1* using an extensive measurement study of Skype in a controlled environment. We set up a network testbed with configurable packet loss, propagation delay, and available link bandwidth. We systematically

Manuscript received May 27, 2012; revised September 12, 2012; accepted November 07, 2012. Date of publication February 20, 2013; date of current version September 13, 2013. This work was supported by Natural Science Foundation of China under contract No. 61071082, National Basic Research Program (973 Program) of China under contract No.2009CB320907 and National Key Technology R&D Program of China under Grant 2012BAH18B03. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Pal Halvorsen.

X. Zhang and Z. Guo are with the Institute of Computer Science & Technology, Peking University, Beijing 100871, China (e-mail: zhangxg@pku.edu.cn; guozongming@pku.edu.cn).

Y. Xu, H. Hu, Y. Liu, and Y. Wang are with the Department of Electrical and Computer Engineering, Polytechnic Institute of NYU, Brooklyn, NY 11201 USA (e-mail: yxu10@poly.edu; hhu01@poly.edu; yongliu@poly.edu; yao@poly.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2013.2247988

generate different network settings. In each setting, we measure Skype's *stationary* behaviors, the behaviors in steady state, such as sending rate, video bit rate, and frame rate after it adapts to a network condition given enough time. Based on the measurement data, we propose rate control model, FEC model, and video quality model for video calls. Those models enable us answer *Q2* quantitatively. Specifically, we study the effectiveness of user back-offs upon quality degradation as a user-level rate control scheme. We study the *TCP-friendliness* of Skype by comparing its responsiveness to packet losses and delays with TCP. We also conduct a numerical case study on a 4G type of wireless network where multiple video call flows and TCP flows compete for bandwidth on uplink. We investigate the bandwidth requirement of Skype video calls, and present a numerical tool to provision a network that can support video calls with a high level of user QoE.

The contributions of our study are three-fold:

1) We are the first to measure the *stationary* behaviors of Skype video calls under different network conditions. We have the following findings: a) Skype's sending rate is insensitive to packet losses when the packet loss rate is below 10%; It significantly reduces its sending rate when loss rate goes above 10%; b) It keeps close track of available network bandwidth, and can maintain bandwidth utilization around 80% under a wide range of network conditions; c) It's sending rate is insensitive to propagation delays; d) It utilizes an overly aggressive forwarding error correction (FEC) coding scheme. Its FEC redundancy ratio is about 4.5 times the actual packet loss rate.
2) We develop models for Skype video call's rate control, FEC redundancy, and video quality. Using the models, we demonstrate that user back-offs react fast to the onset of congestion, and serve as an efficient user-level rate control mechanism. We are the first to include user behaviors in studying the rate control of video calls. We further show that, with the built-in rate control scheme and quality-driven user back-offs, Skype video calls are indeed *TCP-friendly*, and are very responsive to congestion when the network is overloaded.
3) Through a case study on a 4G wireless network, we demonstrate that our models can be used as numerical tools in user QoE-aware network planning and dimensioning.

The rest of the paper is organized as follows. Section II describes the related work. We present our measurement methodology and testbed configurations in Section III. In Section IV, we present measurement results on the stationary behaviors of Skype. In Section V, we propose a rate control model, FEC model and video quality model for Skype video calls. In Section VI, we study the video call's TCP-friendliness by taking into account user back-offs. We investigate the competition between Skype video calls and TCP flows in a 4G type of wireless network. We then study network provisioning to support large-scale video calls with a high level of QoE. Finally, we conclude the paper with summary and future work in Section VII.

II. Related Work

Measurement work on Skype can be classified into two categories: characterizing network protocols and exploring VoIP details. Baset *et al.* [3] are the first to analyze Skype's peer-to-peer (P2P) overlay topology, call establishment protocols, and NAT traversal mechanism. By analyzing Skype call traffic, they reverse engineered the communication mechanisms of Skype. Since then, a lot of papers have been published on Skype's overlay architecture, P2P protocol, and VoIP traffic [4], [5]. In the second category, these studies [2], [6], [7] focused on the quality of Skype's VoIP calls. Huang *et al.* investigated Skype's FEC mechanism and its efficiency for voice streaming [2], [7]. In [6], the authors proposed a USER Satisfaction Index model to quantify VoIP user satisfaction. Cicco *et al.* [8] proposed a congestion control model for Skype VoIP traffic. All of these studies only focused on the voice service of Skype, did not consider its video service.

There are few works emphasizing on video calls. In a closely related paper, Cicco *et al.* [9] measured the responsiveness of Skype video calls to bandwidth variations. They conclude that Skype's response time to bandwidth increase is long. However, this paper only considers the transient behaviors of Skype, and did not systematically measure its stationary behaviors when it reaches the steady state. We conducted extensive measurement of Skype under different network settings of packet losses, packet delays and available network bandwidth. Based on the measurements, we also propose the models for Skype video calls' rate control, FEC redundancy, and video quality.

There have been some other related studies on investigating the impact of user behaviors on network stability [10], [11]. In [11], Tay *et al.* studied how TCP user aborts enable a network sustain a higher demand without causing congestion collapse. Bu *et al.* proposed in [10] that the user back-offs in VoIP will help maintain the network stability. They assumed that VoIP flows do not adapt their sending rates to congestion. Instead, they showed that user back-off is an efficient user-level congestion control mechanism for VoIP. We study rate control of a real video calls, Skype. In our experiments, we observe that Skype video call has a built-in rate control mechanism, which adapts its video rates to network conditions in a wide range. On top of that, we show that user back-offs can further enhance video calls' responsiveness to network congestion.

III. Measurement Test-Bed Setup

Firstly, to study Skype video calls under various network conditions, we set up a controlled testbed on which Skype is observed as a black-box as in Fig. 1. Two hosts with Skype client (ver. 5.2) are connected by a router. Each host has a private IP address and connects to the Internet through the router. To emulate a wireline or wireless network, all packets pass through a software-based network emulator, NEWT [12]. It emulates a variety of network attributes, such as propagation delay, random packet loss, and available bandwidth. We also inject some UDP flows as background traffics using iPerf tools [13].

To emulate a video call, we choose a standard TV news video sequence "Akiyo" from JVT (Joint Video Team) test sequence pool. The sequence has mostly head and shoulder movements. It is very similar to a video-call scenario. We inject the video sequence into Skype using a virtual video camera tool [14]. This ensures the transmitted video content are consistent and repeatable.

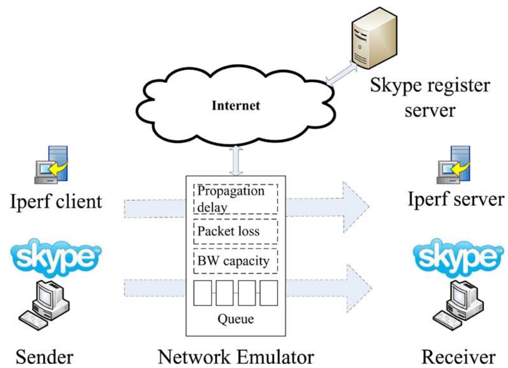


Fig. 1. Network testbed for measurement.

The measurement data are collected by two methods: TCP-dump for packet level information, and Skype reports for video level information. Since Skype employs proprietary protocols and encrypts all control messages, it is hard to reverse-engineer its protocols and measure video information externally. Fortunately, Skype reports some technical information through its user interface, such as video rates, frame rates, RTT, *et al.* We use a screen text capture tool [15] to capture these information periodically from Skype window. The sampling interval is one second.

Unlike previous works, we are more interested in Skype's *stationary* behaviors. By giving enough time for Skype to adapt to a given network scenario, we are able to analyze its built-in rate control mechanisms. In each scenario, after Skype enters steady state, we capture data from ten minutes of a video call. To cope with random effects, we report the mean and 95% confidence interval for each data point of interest. Followed, we will introduce some preliminary observations in the measurements.

A. Rate Control Mechanism

There exists a close-loop control between Skype clients to adapt video rate and video quality. It uses two transport layer protocols: TCP protocol for control messages and UDP protocol for video transmission. The TCP connection acts as a feedback channel through which a receiver periodically reports current network conditions to a sender. Then the sender adapts its UDP sending rate to network conditions. According to our observations, a typical signaling protocol for a video call is as follows: 1) clients connect to a register server on the Internet to log into the system. 2) To start a video call, a client first sends a message to the register servers and looks up the IP address of the callee. 3) The caller attempts to establish a call session with the callee either directly or through a relay. In our experiments, even though the two hosts are behind a NAT router and are assigned with private IP addresses, the caller is always able to find the correct private address of the callee and connect to the private address directly without going through a relay. The monitored video call only traverses the network emulator and NAT router before it reaches the callee. We have full control on the network setting along the path.

Skype employs a video codec provided by On2 (Google) [16]. It is a non-scalable motion-estimation based video codec. In our experiments, the video encoder version of VP6, VP7, and VP8 have ever been detected. It demonstrates Skype is embedded with multiple versions of video codec, and is compatible with earlier versions. This codec supports real-time video coding to generate variable video bit rates. It adapts its rate to network conditions by adjusting video quantization step, video resolution, and the number of frames per seconds (FPS). The observed video codec parameters are listed in Table I.

TABLE I
OBSERVED PARAMETERS OF SKYPE VIDEO CODEC

Video resolution	640*480, 320*240, 160*120
Frame rate per second	5fps - 30fps
Video bits rate	5kbps - 950kbps

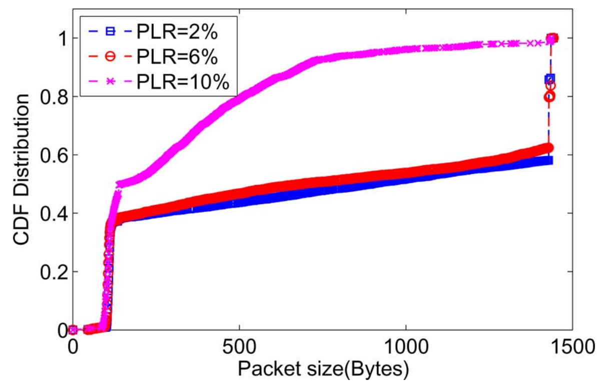


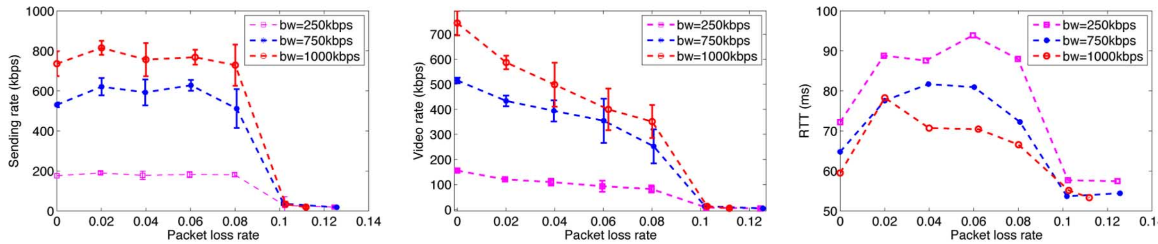
Fig. 2. The CDF distribution of packet size.

Skype adapts its coding parameters and bit rate periodically to network conditions. We find that Skype responds to degrading network conditions faster than to improving network conditions. It shows that Skype employs a conservative adaptation strategy and tries to keep stable the video quality perceived by users. These observations will also be verified in Sections IV–VII.

B. Packet Size Distribution

Generally, the packet sizes of media data are larger than those of control messages. To demonstrate the differences of UDP packets, we also measure the distribution of packet size. The CDF distributions of packet sizes under three Packet Loss Rates (PLR), $\mathrm{PLR}=2\%$, $\mathrm{PLR}=6\%$ and $\mathrm{PLR}=10\%$, are plotted in Fig. 2. From the figure, we notice that the sizes of UDP packets are distributed around two modes: 113 Bytes and 1429 Bytes.

In the two cases of 2% and 6% losses, the percentage of larger packets, with the size of 1429 bytes, is more than about 45%. While in the case of 10% losses, the percentage of larger packets decreases to less than 5%. From packet header analysis, we know that the larger packets are video packets and the small packets are for audio. This measurement results show that audio data will preempt video data in Skype when there are serious packet losses. This is also verified by some other observations. When PLR equals 10%, the video rate drops to 13.3 kbps that only a few larger size packets are generated. At this time, video quality is very poor, but Skype still keeps audio transmission clearly. It means that when Skype detects network condition being worse, it will allocate more resources to maintain audio quality.



(a) (b) (c)

Fig. 3. Impact of packet losses under different available bandwidth. (a) Sending Rate. (b) Video Rate. (c) Round-trip Time.

TABLE II
SYMBOL DEFINITIONS

p	Packet loss rate (PLR)
ρ	FEC redundancy ratio
t	Round-trip time
C_w	Available bandwidth capacity for Skype
R_S	Sending rate for Skype
R_v	Video rate for Skype
Q_S	Video quality in MOS score for Skype video calls
D_S	Video calls' drop-off probability upon quality degradation
$\hat{R}_S$	Expected sending rate considering user back-off behaviors
p_q	Congestion loss probability in a *M/M/1/K* queue
p_c	Links' physical packet loss probability
t_q	Queueing delay in a *M/M/1/K* queue
t_c	Links' propagation delay
C	Total link capacity
R_T	Sending rate for TCP flows
N_T	Number of TCP flows
N_S	Number of Skype flows
N	Total number of Skype video call subscribers
P_b	Call blocking probability in a wireless cell

IV. VIDEO CALL MEASUREMENTS

In this section, we report measured Skype performances under various network conditions. By isolating various network impairments, we investigate how Skype adapts its sending rate and video rate to different packet loss, delay and available bandwidth. Here, sending rate is the total rate at which a source sends out packets. These packets consists of video data and redundant packets. The rate of video data is defined as video rate.

We show some of the important symbols used in the paper and their explanations in Table II.

A. Impact of Packet Loss

We first introduce random packet losses using the network emulator. The sending rate, RTT, and video rate of Skype are measured when Packet Loss Rate (PLR) varies from 0% to 12%. The two-way propagation delay is fixed at 50 ms. To examine whether Skype's responses to packet loss is consistent, we also carry out the above experiments under three different available bandwidth settings: 250 kbps, 750 kbps, and 1000 kbps. The measurement results are illustrated in Fig. 3.

Fig. 3(a) and (b) plots the mean and 95% confidence interval of sending rate and video rate. When PLR is larger than 10%, both sending rate and video rate drop dramatically. Even though there is still abundant bandwidth available on the path, Skype still drops its sending rate to the lowest rate. It indicates that these drops are not due to network congestion. *Thus, we infer that Skype operates in two states: Normal-state (NORM) and Conservative-state (CONS)*. Whenever it detects that the network condition becomes very bad, it would switch to the CONS state. In this state, Skype only sends out necessary data at a low constant rate. From Fig. 3, we know $PLR \geq 10\%$ is one trigger for Skype to switch to CONS state.

We observe that in NORM state, the sending rate oscillates around a constant value of 500 kbps for available bandwidth of 750 kbps. The trend is the same for the other two bandwidth settings. This fact indicates that in NORM state, the sending rate of Skype is ignorant to packet losses. This is totally different from the widely used TCP congestion control scheme, that reduces the sending rate by half upon each perceived packet loss.

From this study, we observe that when the packet loss rate is below 10%, Skype works in normal (NORM) state, in which its sending rate is loss-ignorant; Skype switches to conservative (CONS) state whenever the packet loss rate goes over 10%. Both its sending rate and video rate will be significantly reduced to a low value.

B. Redundant Packets and FEC Ratio

From Fig. 3(a) and (b), we noticed that when PLR is less than 10%, Skype's sending rate almost remains constant while its video rate drops linearly with PLR. The gap between sending rate and video rate increases with PLR. *We conjecture that this gap is due to redundant packets by Forward Error Correction (FEC) coding*. It has been reported in [7] that Skype implements FEC to combat packet losses for voice calls. It is reasonable to assume it protects video calls against packet losses in a similar way. As packet losses increase, Skype decrease its video rate linearly and uses more bandwidth to transmit FEC redundant packets.

To verify our assumption, we measured the gap between sending rate and video rate under a wide range PLR and bandwidth settings. The percentage of gap out of sending rate, also named FEC redundancy ratio, are plotted as a function of packet loss rate in Fig. 4.

Obviously, the FEC ratio is nearly a linear function of PLR, and is independent from both sending rate and video rate. Thus we are able to model the FEC ratio function $\rho(p)$ through a linear regression:

$$\rho(p) = \psi + \omega p \tag{1}$$

where p denotes packet loss rates, ψ and ω are constants determined by Skype's FEC mechanism. Through curve fitting, we

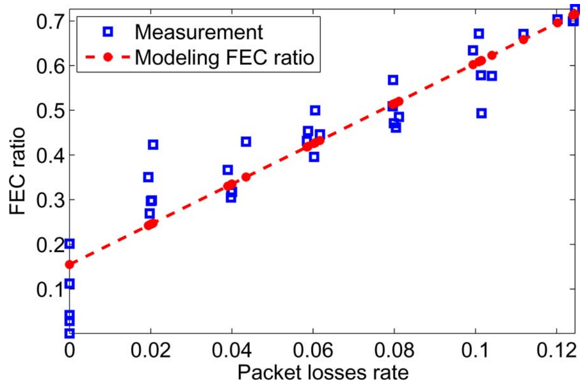


Fig. 4. FEC redundance ratio as a function of PLR.

TABLE III
MEAN AND STANDARD DEVIATION OF RTT WITH $bw = 750$ kbps

Loss Ratios	0%	2%	4%	6%	8%	10%	12%
Median	64.8	77.6	81.7	80.9	72.21	53.7	54.4
Std. Dev.	16.8	30.0	39.4	33.2	28.9	8.61	9.9

found that $\psi = 0.15$, $\omega = 4.5$. The model curve is also illustrated in Fig. 4. It demonstrates that the model curve fits the measurement points very well.

In (1), ω can be explained as the ratio between FEC redundancy and packet loss rate. From the curve fitting results, we know that the redundancy ratio of Skype is about 4.5 times of loss rate. This is a very aggressive FEC protection. This result echos the conclusion in [7] that there is still a significant space to improve Skype's FEC efficiency.

C. Round-Trip Time

The results of round-trip time (RTT) measurement are shown in Fig. 3(c). Since we fixed the two-way propagation delay at 50 ms, the difference between the measured RTT and 50 ms is the queueing delay experienced by Skype due to congestion along the path.

Table III lists the mean and standard deviation of RTT under link bandwidth of 750 kbps. We observe that the deviations of RTT are very large. The largest ratio of deviation to mean is about 48%, while that for the sending rate is less than 10%. This indicates that there is much larger variations in RTT than in sending rate.

Fig. 3(c) plots the mean RTT. There is no strong correlation between RTT and PLR in the NORM state. RTT decreases significantly when Skype switches to the CONS state, due to the largely reduced sending rate. But by comparing Fig. 3(a) and (c), we observe that RTT is correlated with sending rate. When Skype's sending rate increases, RTT also increases, but more steeply. We also notice that the RTT for link bandwidth of 1000 kbps is less than that of 250 kbps, even though the former's sending rate is larger. So RTT is not only affected by sending rate, but also by bandwidth capacity.

To verify our assumption, we measured the sending rate and RTT under a wide range PLR and bandwidth capacity settings. It is interesting to observe how RTT relates to sending rate R_S and available bandwidth capacity C_w. Fig. 5 is the scatter plot of Skype RTT versus the surplus bandwidth $C_w - R_S$. Each point corresponds to one Skype run under some combination of PLR and available bandwidth. It can be seen that the RTT drops as the surplus bandwidth increases.

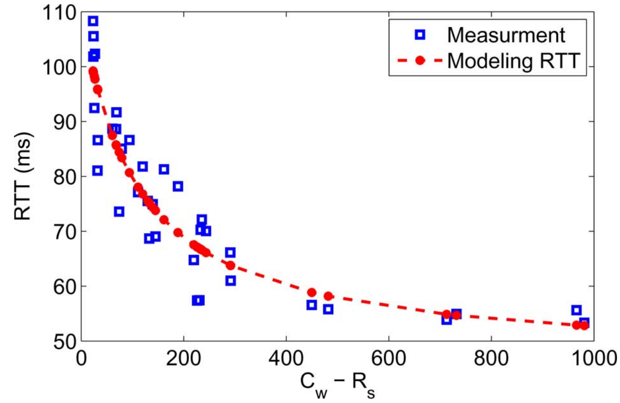


Fig. 5. Relation between Skype RTT and surplus bandwidth. Each point is for one combination of bandwidth and packet loss rate.

We can model RTT as a function of the surplus bandwidth:

$$t = \frac{\alpha}{C_w - R_S + \beta} + \sigma, \qquad (2)$$

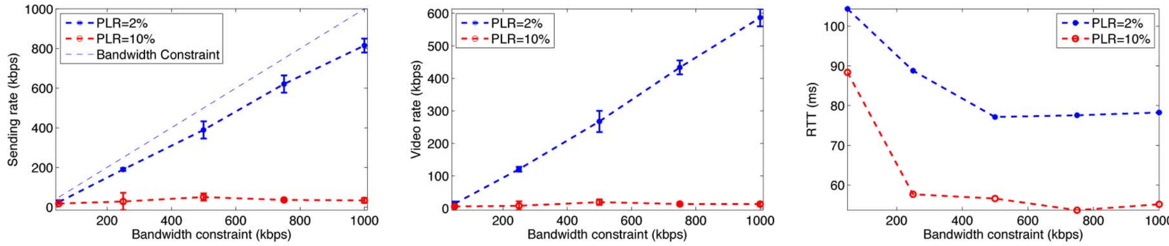
where α, β and σ are model parameters. The intuition behind this formula is that the system can be generally modeled as a $M/M/1$ queue with some corrections on the traffic pattern. From the classic queueing theory, the per-packet queueing delay in a $M/M/1$ queue with capacity C and rate R can be calculated as $B/C - R$, where B is the average packet size. In other words, queueing delay is inversely proportional to the surplus bandwidth $C - R$ on a link.

In (2), we can treat σ as the static propagation delay component in RTT. α is packet size. β controls the RTT increasing speed at low surplus bandwidth region when Skype tends to send a fairly large number of small packets. By curve fitting over the measured data, we find the parameters to be $\sigma = 54.63$ and $\alpha = 6906$, which match our system settings that the propagation delay is about 50 ms, and the actual measured average packet size is about 6364 bits. Therefore, with the sending rate model, it is possible to calculate RTT using a $M/M/1$ queue.

D. Impact of Available Bandwidth

To investigate how Skype responds to bandwidth available in the network, we vary the bandwidth capacity in network emulator from 50 kbps to 1000 kbps while fixing the PLR and propagation delay. In order to cover both NORM state and CONS state, two PLRs are considered, 2% and 10%. In both cases, the two-way propagation delay is set to 50 ms.

The measured results are shown in Fig. 6. We again observe distinct behaviors of Skype in NORM state and CONS state. In Fig. 6(a), when $PLR = 2\%$, Skype is in NORM state and increases its sending rate proportionally as the bandwidth capacity increases. On the other hand, when $PLR = 10\%$, Skype is in CONS state and its sending rate almost remains unchanged even when the bandwidth capacity increases. This verifies our conjecture in Section IV-A that Skype has two states. When it



(a) (b) (c)

Fig. 6. Impact of available network bandwidth. The two-way propagation delay is fixed at 50 ms. (a) Sending Rate. (b) Video Rate. (c) Round-trip time.

detects PLR is larger than 10%, it will switch to the CONS state and sends data at the lowest rates.

Similar trend can be detected in video rate as illustrated in Fig. 6(b): video rate in NORM state changes linearly with bandwidth capacity, and remains unchanged in CONS state. It is also noticed that the video rate increases proportionally with the sending rate. This verifies that there exists a linear function between sending rate and video rate when PLR is fixed, as showed in Section IV-B.

The changes of RTT with available bandwidth are shown in Fig. 6(c). We observe that RTT decreases in general as bandwidth capacity increases. The slight increasing trend towards the end of the curves is again due to the large variations of RTT and the curves only report the mean. Again, with the fixed two-way propagation delay, the RTT reduction is due to decreasing queueing delay resulted from bandwidth capacity increase.

From these results, we find that Skype can closely keep track of the available network bandwidth and adjust its sending rate and video rate to efficiently utilize available bandwidth without causing excessive congestion. While the exact algorithm employed by Skype to track available bandwidth is unknown, due to its loss-ignorance in NORM state as studied in Section IV-A, we conjecture that the algorithm is most likely driven by packet delays, as commonly done in available network bandwidth measurement [17], [18].

E. Impact of Propagation Delay

Network delay perceived by a packet consists of two components: congestion delay and propagation delay. While congestion delay is highly variable, propagation delay is static and is determined by paths taken by the packet. In this section, we measure Skype's performance by varying the two-way propagation delay in the network emulator. The bandwidth capacity is set to 500 kbps and the PLR is 0. We vary the propagation delay from 50 ms to 2000 ms.

The measurement results are listed in Table IV. We observe in this scenario Skype keeps sending rate stable around 360 kbps for all propagation delay settings. We conclude that typically Skype does not adapt its sending rate when the propagation delay changes. The RTT observed by Skype increases linearly with propagation delay. The differences between the two are very small which is due to the queueing delay occurred in the networks.

TABLE IV
MEAN PERFORMANCES OF SKYPE UNDER VARIOUS PROPAGATION DELAY

Propagation delay	Sending rate	Video rate	RTT
50	354.4	343.6	64.9
200	358.7	342.3	220.5
400	352.7	335.9	416.1
1000	355.8	338.9	1017.7
2000	368.8	350.9	2019

From the results, we conclude that Skype rate control is insensitive to propagation delay.[1] Combined with the findings in Sections IV-A and IV-D, we infer that Skype's rate control algorithms are driven by congestion delay, instead of propagation delay.

V. MODELING VIDEO CALLS BEHAVIORS

In this section, we present more extensive measurement results and propose analytical models for Skype's sending rate, video rate, and video quality. Those models will allow us to extrapolate the measurement results to quantitatively answer important questions regarding the effectiveness of user back-offs as a rate control mechanism and the competition between Skype video calls and TCP flows in Section VI.

A. Sending Rate

In Section IV, we showed that Skype increases its sending rate linearly with the available bandwidth. In addition, Skype's rate control is insensitive to packet losses in NORM state. We propose to model Skype's sending rate R_S as a piecewise linear function as follows.

$$R_S(C_w, p) = \begin{cases} \gamma_w^C + \mu & p < 10\% \\ \delta & p \geq 10\% \end{cases} \tag{3}$$

where C_w is bandwidth capacity, and p is packet losses rate. δ denotes the constant sending rate at CONS state, γ and μ are two model parameters in NORM state.

To verify the model and derive the parameters, we measured Skype sending rates under a wide range of PLR and available bandwidth settings. Totally 25 scenarios are set up with the PLR varying from 0%–12% and the available bandwidth varying from 50 kbps–1000 kbps. Fig. 7 shows the measured data versus our proposed model. As illustrated in Fig. 7(a), in NORM state, data points at each available bandwidth are overlapped or closely located to each other. A linear curve

[1]Due to the realtime interaction requirement, when the propagation delay is excessively long, e.g., over 2 seconds, users would have dropped the video call.

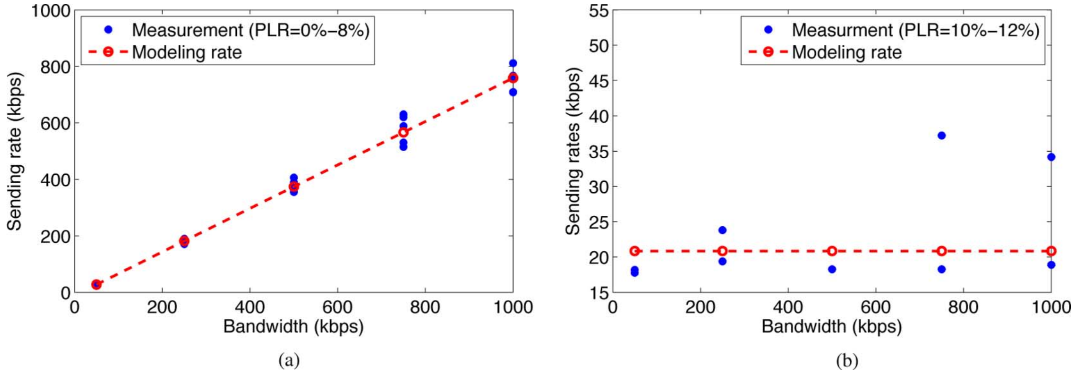


Fig. 7. Modeling sending rate with bandwidth capacity. PLR is from 0% to 12%. The two-way propagation delay is fixed as 50 ms. (a) Skype in NORM State. (b) Skype in CONS State.

is sufficient to fit all the data points. In CONS state, most of the sending rates are around 21 kbps as shown in Fig. 7(b). With curve fitting over the measured data, the parameters are obtained as $\gamma = 0.77$, $\mu = -10.8$ and $\delta = 21$. Here, γ can be explained as the bandwidth utilization. From the model, we know the bandwidth utilization of Skype is about 77%.

From this sending rate model, we summarize that Skype has two built-in rate control mechanisms: when PLR is less than 10%, it controls its sending rate to maintain around 77% utilization of the available bandwidth; when PLR is greater than 10%, it sends out data at a conservative constant rate.

B. Video Rate

Skype uses a video codec that adapts its video bit rate to network conditions. This is verified in our measurements in Section IV-D. We also observe that Skype decreases its video rate as packet losses increase, but keeps its sending rate unchanged. So we conclude that Skype uses more bandwidth to transmit FEC bits when PLR increases as in Section IV-B.

Let R_v be the actual video rate. From the definition of FEC ratio ρ, we have a video rate model as

$$R_v = (1-\rho)R_S.$$

When ρ becomes larger, there are more redundancy bits.

Combining the FEC model in (1) and the sending rate model in (3), the video rate of Skype can be formulated as a function of bandwidth capacity and PLR:

$$R_v(C_w, p) = \begin{cases} (1-\psi-\omega p)(\gamma C_w + \mu) & p < 10\% \\ (1-\psi-\omega p)\delta & p \geq 10\% \end{cases} \tag{4}$$

C. Frame Rate

To assess the video quality of Skype, we need to measure Skype's frame rate under various network conditions. Since most video codec changes its frame rate based on video rate, we investigate the relation between video rate and frame rate. We collect the frame rate and video rate information from Skype window. The measured data are plotted in Fig. 8. Again, each point corresponds to a measurement result under a combination of PLR and available bandwidth.

In Fig. 8, we observe that most video frame rates are distributed around 5, 10, 15, and 28 frame-per-second (fps). As

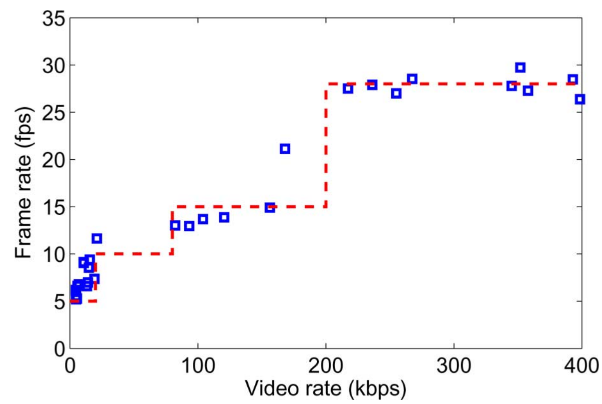


Fig. 8. Video frame rate as a function of video bit rate.

video rate increases, the frame rate increases. We deduce that Skype codec encodes video into a pre-selected set of frame rates. It selects a frame rate according to the current video rate. Thus, we propose the frame rate model of Skype as a piecewise-constant function:

$$f(R_v) = \begin{cases} 5 & R_v \leq 20 \\ 10 & 20 < R_v \leq 80 \\ 15 & 80 < R_v \leq 200 \\ 28 & 200 < R_v \end{cases}, \tag{5}$$

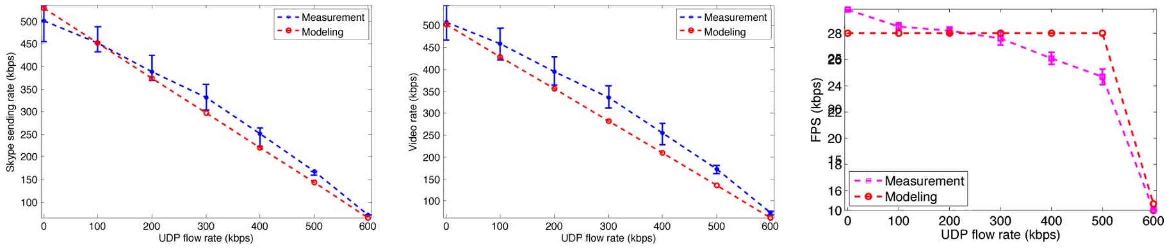
where the video rate is measured in kbps.

D. Video Quality

The ultimate QoE measure of video call service is the video quality perceived by users. We use a subjective quality model [19] to assess Skype's video quality. The videophone subjective quality model, also known as the opinion model for video-telephony applications, has been standardized as ITU-T Recommendation G.1070 [20].

The perceived video quality is measured by the Mean Opinion Score (MOS), a subjective quality score that ranges from unacceptable to excellent. This model estimates the video quality affected by coding distortion and frame reduction. For the same video codec, the video quality is a function of video rate and frame rate. We rewrite the model as:

$$Q = 1 + G(f, R_v), \tag{6}$$



(a) (b) (c)

Fig. 9. Model validation. The capacity of bottleneck link is 700 kps, and the two-way propagation delay is fixed at 50 ms. (a) Sending Rate. (b) Video Rate. (c) Frame Rate.

TABLE V
CALL DROP RATIO AT DIFFERENT MOS SCORES

MOS score	1.5	2	2.5	3	3.5	4	4.5
Drop ratio	0.98	0.85	0.7	0.5	0.3	0.1	0

where f is the frame rate measured in frames-per-second (fps), and R_v is the video rate.

Along with the Skype's frame rate model and the video rate model in the previous section, the Skype's subjective quality is as (7). The coefficients a, b, c, d, e, h, g are model parameters as defined in ITU-T Recommendation G.1070 [20]. $a = 1.431$, $b = 0.02228$, $c = 3.759$, $d = 184.1$, $e = 1.161$, $h = 1.446$, $g = 0.03881$.

$$\begin{aligned} Q_S(C_w,p) &= 1 + G(f(R_v), R_v(C_w,p)) \\ &= 1 + \left(c - \frac{c}{1 + \left(\frac{R_v(C_w,p)}{d}\right)^e} \right) * \\ & exp\left(- \frac{(ln(f(C_w,p)) - ln(a + bR_v(C_w,p)))^2}{2(h + gR_v(C_w,p))^2} \right). \end{aligned} \quad (7)$$

E. User Behaviors Upon Quality Degradation

Unlike elastic data transmission, video call users are more sensitive to video quality. Once users' quality-of-experience are not comfortable, they would drop the call. We call it *user back-off* behaviors. In the current Skype implementation, it would pop up a window to recommend users to turn off their video when network conditions degrade. Noted here, we consider a user dropping the video call if she turns off her video. The Skype call might still continue with just voice.

Previous studies [10], [11] have shown that user back-offs serve as an effective rate control scheme at the *user-level*. It can significantly reduce network traffic when user QoE degrades as network congestion builds up. Thus, we use the obtained quality model in (7) to study the effectiveness of user back-offs as a user-level rate control scheme for Skype video calls.

Specifically, we consider that a user drops her video call probabilistically when the MOS score degrades. From the MOS score definition and subject quality assessment in G.1070 [20], we have video calls' drop-off probabilities as in enumerated in Table V.

The drop-off probability is a function of MOS score. Given available network bandwidth and packet loss rate, we can then calculate video drop-off probability as follows:

$$D_S(C_w,p) = D_S(Q_S(C_w,p)) \quad (8)$$

Since Skype users may adapt to network congestion through video drop-offs, given a group of N_S Skype users under a network condition characterized by $\{C_w, p\}$, the number of active users can be represented as $N_S(1 - D_S(C_w,p))$. The total *effective* traffic rate generated by all users can be approximated as:

$$\hat{R}(C_w,p,N_S) = N_S(1 - D_S(C_w,p))R_S(C_w,p).$$

Considering the audio traffic is very small, and the average traffic contribution of each user is simply:

$$\hat{R_S}(C_w,p) = (1 - D_S(C_w,p))R_S(C_w,p) \quad (9)$$

We call $\hat{R_S}(C_w,p)$ the *expected* Skype sending rate. It represents the expected traffic generated by a Skype video call considering user drop-offs as a user-level rate control. In the analysis part of the paper, we will use $\hat{R_S}(C_w,p)$ in place of Skype's sending rate.

F. Model Validation

To validate the above models, we conduct experiments where Skype flows compete with UDP flows in one bottleneck link. The link's capacity is set to 700 kbps, and the propagation delay is fixed at 50 ms. The UDP flow with constant rate is injected into the test-bed using "iPerf" tool. After Skype enters into steady state, the sending rate, video rate, and frame rate are sampled. We carry out experiments under various scenarios with UDP flow rates varying from 0 kbps to 600 kbps. Each experiment is run for 20 minutes and the sampling interval is one second. All measured points are illustrated in Fig. 9.

We also use the proposed models to predict Skype's rates. When Skype enters into steady state, the equilibrium of bandwidth allocation is reached. All flows detect the same packet loss ratio p and round-trip-time t. Assuming that the bandwidth of link is C and the UDP rate is fixed at R_U, the available bandwidth for Skype can be expressed as:

$$C_w = C - R_U \quad (10)$$

Substituting (10) into the sending rate model $R_S(C_w, p)$ in (3), the video rate model $R_v(C_w, p)$ in (4), and the frame rate model $f(C_w, p)$ in (5), Skype's modeling rates are calculated.

We compare the modeling results with the measurements as in Fig. 9. The figures show that our modeling curves match the measurement results pretty well. The Pearson correlation coefficients between the measurement and modeling results are 0.9898, 0.9831, and 0.9545 for sending rate, video rate, and frame rate respectively.

VI. Analysis of Video Calls Performance Under Competition

Our measurement and modeling study characterize a Skype video call's behaviors in a controlled environment. In a real network, a Skype video call competes with other network flows, including other Skype calls and TCP flows, for network resources. In this section, we extrapolate from the obtained the rate control and quality models and quantitatively answer the following questions through numerical analysis:

1) How video call users respond to quality degradation resulted from network impairments? How effective user back-offs are as a user-level rate control scheme?
2) What is the performance of a video call when it competes with other Skype calls and TCP flows? Is Skype video call TCP-friendly?
3) For a given network, how many video calls it can support? For a given video call population, how to provision network bandwidth to guarantee a certain level of user QoE?

A. Network Model

We consider a wireless cellphone network where multiple TCP and Skype users compete for the access to the base-station. Both the uplink and downlink are the potential bandwidth bottlenecks. In our numerical analysis, we consider a LTE wireless network [21], with 100 Mbps downlink capacity and 50 Mbps uplink capacity. Since video call is bi-directional communication, all users within a cell compete for the 50 Mbps uplink bandwidth. Due to the lossy nature of wireless transmission, we further assume that there are random channel losses on access links other than congestion losses.

The base station takes traffic from a large number of users. It has been shown in [22] that when the link multiplexing degree is high, the packet arrival process can be well modeled as a Poisson process. Thus, we consider an access link with finite buffer and model it as a $M/M/1/K$ queue, which is a drop-tail queue with a total buffer size of K. Let C denote the total link capacity. Given the total traffic arrival rate of R, the average congestion loss probability p_q and queueing delay t_q can be approximately calculated as [23]:

$$p_q = \frac{\frac{1-R}{C}}{1-(\frac{R}{C})^{K+1}}\left(\frac{R}{C}\right)^K \tag{11}$$

$$t_q = \frac{B(\frac{1-R}{C})}{R(1-(\frac{R}{C})^K)}\sum_{i=0}^{K} i\left(\frac{R}{C}\right)^i \tag{12}$$

where B is the average packet size.

TCP users react to congestion by adjusting their sending rates according to packet losses and round trip time. We assume all TCP users are long-lived flows whose congestion control phase is significant longer than the slow start phase. According to [24], for a TCP flow with end-to-end loss probability p and RTT t, the TCP sending rate can be characterized by:

$$R_T = \frac{1.5\sqrt{\frac{2}{3}}B_T}{t\sqrt{p}} \tag{13}$$

where t, q and B_T represent RTT, packet loss rate and TCP packet size respectively. The delay t is the sum of queueing delay t_q and propagation delay t_c. The end-to-end packet loss probability consists of two parts: $p = p_q + p_c$, where p_q is the queueing loss due to congestion, and p_c is the random channel loss. Our analysis can also be extended to incorporate short-lived TCP sessions following a different model [25]. We skip it here for the clarity of presentation.

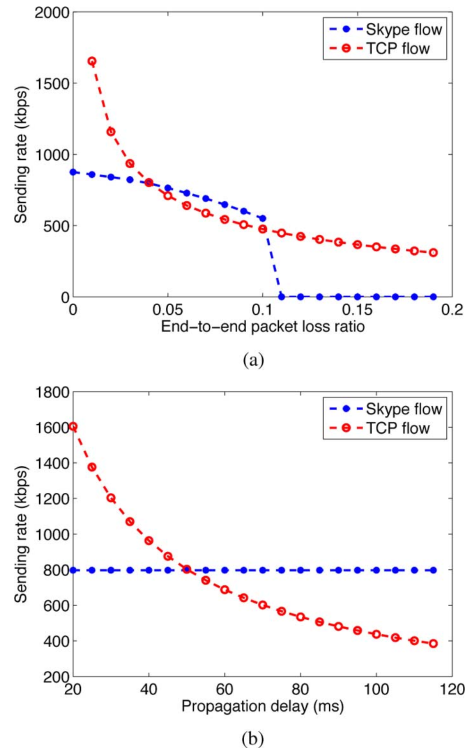
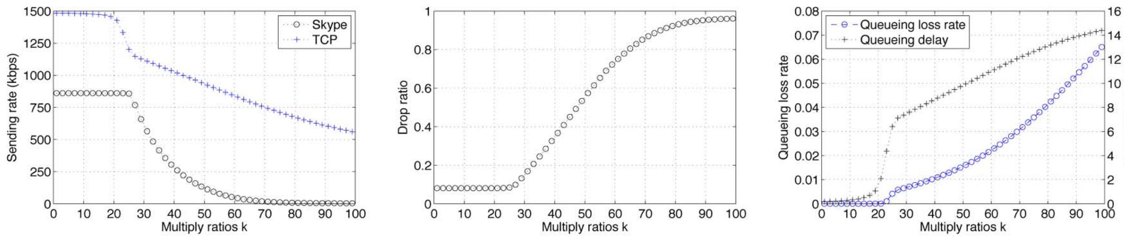


Fig. 10. Responsiveness to loss and delay. (a) Varying link loss ratio, the propagation delay is fixed at 60 ms. (b) Varying propagation delay, the packet loss ratio is fixed at 4%.

B. TCP-Friendliness of Skype Video Call

Since Skype uses UDP as transport protocol, it has no congestion control scheme at the transport layer. To maintain the Internet stability, it is important for applications developed over UDP protocols to be *TCP-friendly* [24]. As demonstrated in the previous sections, Skype has its built-in rate control scheme at the application layer. Additionally, user back-offs can be considered as a rate control scheme at user-level. We now study the TCP-friendliness of Skype video calls.

1) Responsiveness to Loss and Delay: We first compare a Skype video call's responsiveness to packet losses and delays with the responsiveness of a TCP flow. We not only compare the sending rates of Skype and TCP, but also compare the slope



(a) (b) (c)

Fig. 11. Bandwidth allocation between Skype and TCP users as we scale up their population proportionally. (a) Sending rate. (b) Video drop-off ratio. (c) Queue drops and delay.

of their rate curves, which indicates how fast each flow reduces its rate under various network impairments. If Skype curve has a steeper slope than TCP's, it means Skype is more responsive and reacts faster to network condition changes than TCP; Otherwise, it means Skype is more sluggish and reacts slower than TCP.

First, we introduce random packet losses at different PLR while keeping the end-to-end delay and available bandwidth fixed. The numerical results are illustrated in Fig. 10.

In Fig. 10(a), we plot the sending rate as end-to-end PLR increases from 0% to 20%. The end-to-end delay is fixed at 60 ms. TCP starts with a much higher rate than Skype. But TCP flow is more responsive to packet losses than Skype flow. When the end-to-end PLR increases from 1% to 3%, the sending rate of TCP drops quickly. In contrast, when PLR is less than 10%, Skype only slowly decreases its expected sending rate as defined in (9). Skype's rate decreases in this region is due to user backoffs. The slope of the Skype curve is flatter than that of TCP. It shows that Skype is more sluggish in reducing its rate than TCP when the random PLR is below 10%. When PLR goes over 10%, Skype switches to the CONS state; both its sending rate and quality decrease significantly. In this region, Skype is much more conservative than TCP.

We also investigate how Skype and TCP adapt their rates to propagation delay. In Fig. 10(b), we plot the sending rate of Skype and TCP as we increase the two-way propagation delay from 20 ms to 120 ms. Again, TCP has a much higher rate than Skype when the propagation delay is small. But TCP is more responsive to propagation delay increase than Skype. As propagation delay increases, TCP reduces its rate while Skype keeps its rate unchanged. However, if we keep increasing the propagation delay till it become too large for realtime user interaction, most users will drop off, which will also cause Skype sending rate drop much faster than TCP.

2) Head-Head Competition With TCP: In this section, we numerically study the behaviors of Skype video calls when they share a wireless bottleneck link with TCP flows. The random channel loss rate is set at $p_c = 2\%$; the propagation delay is $t_c = 50$ ms. There are totally N_T TCP flows and N_S Skype flows. Then the aggregate traffic rate is:

$$R = N_T R_T(t,p) + N_S(1 - D_S(C_w,p))R_S(C_w,p), \quad (14)$$

where R_T is the sending rate calculated by the TCP rate model in (13).

The available bandwidth seen by each Skype user can be determined by

$$C_w = C - R + R_S(C_w,p) \quad (15)$$

Equation (15) states that the available bandwidth seen by one Skype user equals the total bandwidth minus the aggregated TCP rates and Skype rates except the current one. By merging (3), (11) and (12), (13), (8), (14), and (15), we reach at three equations for three unknowns $\{C_w, p_q, t_q\}$. We can numerically solve them for the rate allocation between TCP and Skype users and the video quality of Skype.

We calculate the bandwidth allocation among TCP flows and Skype flows as the number of flows in each group grows proportionally. We set the initial number of TCP and Skype users to one. Then we scale up their population by the same factor k. Fig. 11 plots the bandwidth shares of Skype and TCP flows as k increases.

In Fig. 11(a), when the scale up factor $k \leq 20$, there is abundant bandwidth in the cell, and almost no congestion loss and delay. Skype operates in its NORM state and video quality is good. TCP only slightly drops its rate as k increases. Congestion starts to build up as k goes over 20, both queueing delay and packet loss increase as illustrated in Fig. 11(c). TCP reacts faster to congestion and quickly reduces its sending rate. Meanwhile, as k keeps increases, the available bandwidth for each Skype call decreases, and the quality perceived by Skype users start to degrade. User video drop-offs kick in quickly as k goes over 25, as shown in Fig. 11(b). The expected sending rate of Skype decreases at a faster pace than TCP in Fig. 11(a). TCP users take advantage of Skype user drop-offs and slow down their rate decreases when $k \geq 25$. Towards the end, all Skpy users turn off video and only the "persistent" TCP users prevail in the head-head bandwidth competition.

To summarize, when network congestion level is low, Skype is more sluggish than TCP and manages to provide smooth video transmission in face of mild random link losses and delays. When the network congestion level is high, Skype is more conservative and react faster to congestion than TCP, due to its built-in rate control scheme and user back-offs. This indicates that Skype is indeed TCP-Friendly when the network is heavily loaded and the congestion level is high.

C. QoE-Aware Network Capacity Planning

In this section, we demonstrate how to apply our models to network capacity planning and dimensioning. It is an interesting problem that how much capacity a mobile network needs to support users with satisfiable video call quality. Following the planning practice in the traditional telephone network, we assume the video call attempts arrive following a Poisson process, and the call duration is exponentially distributed.

We assume there are totally N subscribers in a base-station cell. Each user makes in average k calls in one time unit T. Then the aggregate call arrival rate is $\lambda = (kN/T)$. Let h denote the average call duration by Skype users. Then the offered traffic load measured in Erlang is:

$$A(N) = \frac{kNh}{T} \tag{16}$$

Given a network that can handle N_S concurrent video calls with satisfactory user QoE, we can use the Erlang-B formula [23] to calculate the call blocking probability for a group of N video subscribers as:

$$P_b(N_S, N) = \frac{\left(\frac{A(N)^{N_S}}{N_S!}\right)}{\sum_{i=0}^{N_S} \frac{A(N)^i}{i!}} \tag{17}$$

To make the call blocking probability less than a target value P_b with N subscribers, based on (16) and (17), we can get the target network capacity $N_S(N, P_b)$ measured in the number of supportable concurrent calls.

Ultimately, we want to know how much bandwidth $C(C_w, p, N_S)$ is needed to support N_S concurrent Skype video calls with satisfactory QoE. As shown in Table V, the MOS score should be larger than a constant threshold to get a satisfactory user rating. Assume the quality constraint is Q_0, the available bandwidth seen by each Skype user can be determined by

$$C_w = C(C_w, p, N_S) - (N_S - 1)(1 - D_S(Q_0))R_S(C_w, p). \tag{18}$$

Also, we know the video quality is a function of C_w and p from the Skype subject quality model in (7). To maintain a MOS score above Q_0, we should have

$$Q_S(C_w, p) \geq Q_0, \tag{19}$$

Then, to find the minimum required bandwidth to support N_S users, we formulate it as a constraint optimization problem:

$$\begin{aligned} C^* = \underset{\{C_w, p, N_S\}}{\arg\min}\ & C\left(C_w, p, N_S\right) \\ \text{subject to}\ & \\ & Equ.\ (18) \\ & Equ.\ (19) \\ & Equ.\ (11) \end{aligned} \tag{20}$$

In the above equations, there are three constraints: the available bandwidth as in (18), the quality model as in (19) and the tail-drop queue model as in (11). By solving this optimization problem, we are able to find the minimum bandwidth C^* needed to support $N_S(N, P_b)$ concurrent Skype video calls at MOS score of Q_0. Based on the definition of $N_S(N, P_b)$, C^* is the bandwidth needed to support a population of N Skype video call subscribers with blocking probability under P_b.

For numerical analysis, we adopt the average Skype call duration as $h = 10.42$ minutes from the measurements results in [6]. We also assume the average number of video calls per-user per-day is $k = 2$. We calculate the network capacity requirement under different user populations and different target call block probabilities: $P_b = 1\%$, 5%, or 10%. To maintain a satisfactory user QoE, the required MOS score is set to $Q_0 = 4$. The results are summarized in Table VI.

TABLE VI
NETWORK PROVISIONING WITH QOE CONSTRAINT

Subscribers number	1000	2000	3000	4000	5000
$P_b < 1\%$	20	34	51	64	78
$P_b < 5\%$	17	29	43	57	69
$P_b < 10\%$	14	26	39	52	63

In Table VI, we notice that for a network with 3000 subscribers, the minimal network capacity requirement is 51 Mbps when the target call block probability is less than 1%. This results indicates that even with the LTE standard which archives 50 Mbps upload capacity, it is still not able to support more than 3000 Skype video registered users in a cell.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we characterized the rate control schemes and video quality of Skype video calls. Through extensive measurement, we showed that Skype is robust against mild packet losses and propagation delays and can efficiently utilize available network bandwidth. Skype significantly reduces both sending rate and video rate when the network losses become severe. Based on measurement results, we developed rate control model, FEC model, and video quality model for Skype. Through numerical analysis, we showed that user back-offs serve as an effective user-level rate control scheme, and Skype is indeed "TCP-friendly".

As a future work, we will extend our framework to include audio quality and its impact on user behaviors. We showed that Skype rate control is mostly driven by congestion delay and available bandwidth. We will explore more on delay-driven rate control algorithms for video telephony in wireline and wireless networks. We are also interested in improving the efficiency of Skype's FEC strategy. In this paper, we focused on the stationary behaviors of Skype. The transient behaviors of Skype under competition with other flows is also an interesting topic for future study. Finally, the rate control and video quality of Skype's multi-party video conferencing is also on our future research agenda.

REFERENCES

[1] Skype Inc., "Skype features". [Online]. Available: http://www.skype.com.
[2] T. Huang, K. Chen, and P. Huang, "Tuning Skype redundancy control algorithm for user satisfaction," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1179–1185.
[3] S. A. Baset and H. G. Schulzrinne, "An analysis of the Skype peer-to-peer internet telephony protocol," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–11.
[4] D. Bonfiglio, M. Mellia, M. Meo, N. Ritacca, and D. Rossi, "Tracking down Skype traffic," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 261–265.
[5] S. Guha, N. Daswani, and R. Jain, "An experimental study of the Skype peer-to-peer VoIP system," in *Proc. 5th Int. Workshop Peer-to-Peer Systems*, Santa Barbara, CA, USA, Feb. 2006, pp. 1–6.
[6] K. Chen, T. Huang, P. Huang, and C. Lei, "Quantifying Skype user satisfaction," in *Proc. ACM SIGCOMM*, Oct. 2006, vol. 36.
[7] T. Y. Huang, K. T. Chen, and P. Huang, "Could Skype be more satisfying? a QoE-Centric study of the FEC mechanism in an internet-scale VoIP system," *IEEE Netw.*, vol. 24, no. 2, pp. 42–42, Mar. 2010.

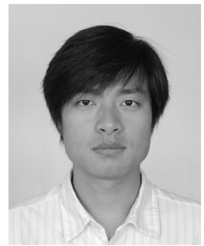
[8] L. D. Cicco, S. Mascolo, and V. Palmisano, “A mathematical model of the Skype VoIP congestion control algorithm,” in *Proc. IEEE Conf. Decision and Control*, Dec. 2008.
[9] L. D. Cicco, S. Mascolo, and V. Palmisano, “Skype video congestion control: An experimental investigation,” *Comput. Netw.*, vol. 55, no. 3, pp. 558–571, Feb. 2011.
[10] T. Bu, Y. Liu, and D. Towsley, “On the TCP-friendliness of VoIP traffic,” in *Proc. IEEE INFOCOM*, Dec. 2006.
[11] R. Morris and Y. Tay, A Model for Analyzing the Roles of Network and User Behavior in Congestion Control Massschusetts Institute of Technology,, 2004, Tech. Rep.
[12] Microsoft Research Asia, “Network Emulator for Windows Toolkit (NEWT)”. [Online]. Available: http://blogs.msdn.com/b/lkruger.
[13] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, “Iperf”. [Online]. Available: http://dast.nlanr.net/Projects/Iperf/.
[14] e2eSoft, “Vcam: Webcam emulator”. [Online]. Available: http://www.e2esoft.cn/vcam/.
[15] Renovation Software, “Text grab for windows”. [Online]. Available: http://www.renovation-software.com/en/text-grab-sdk/textgrab-sdk.html.
[16] Google Inc., “On2 video codec”. [Online]. Available: http://www.on2.com/on2-video/.
[17] M. Jain and C. Dovrolis, “Pathload: A measurement tool for end-to-end available bandwidth,” in *Proc. Passive and Active Measurements (PAM) Workshop*, 2002, pp. 14–25.
[18] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang, “Locating internet bottlenecks: Algorithms, measurements, and implications,” in *Proc. 2004 Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2004.
[19] K. Yamagishi and T. Hayashi, “Qrp08–1: Opinion model for estimating video quality of videophone services,” in *Proc. IEEE Global Telecommunications Conf.*, Dec. 2006, pp. 1–5.
[20] G.1070: Opinion Model for Video-Telephony Applications, ITU-M, 2007, ITU, Tech. Rep.
[21] Recommendation m.1645: Framework and Overall Objectives of the Future Development of imt-2000 and Systems Beyond imt-2000, ITU-R, 2003, ITU, Tech. Rep.
[22] J. Cao and K. Ramanan, “A Possion limit for buffer overflow probabilities,” in *Proc. IEEE INFOCOM*, Mar. 2002.
[23] L. Kleinrock, *Theory, Volume 1, Queueing Systems*. New York, NY, USA: Wiley-Interscience, 1975.
[24] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the Internet,” *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 458–472, Aug. 1999.
[25] N. Cardwell, S. Savage, and T. Anderson, Modelling the Performance of Short TCP Connections, Univ. Washington, 1998, Tech. Rep.

Xinggong Zhang (M’11) received the B.S. degree from Harbin Institute of Technology and the M.S. degree from Zhejiang University in 1995 and 1998 respectively, and the Ph.D. degree from Peking University, Beijing, China, in 2011.

He is currently an Associate Professor with the Institute of Computer Science and Technology, Peking University. He has been a senior Research Staff with Founder Research China, and was the recipient of the First Prize of the Ministry of Education Science and Technology Progress Award in 2006, and the Second Prize of the National Science and Technology Award in 2007. His general research interests lie in multimedia networking and video communications. His current research directions include video conferencing, dynamic HTTP streaming, and content-centric networking.

Yang Xu (S’12) received his bachelor degree in the field of automatic control from University of Science and Technology of China in July 2009. He is currently a Ph.D. candidate at the Department of Electrical and Computer Engineering, Polytechnic Institute of New York University, Brooklyn, New York.

His current research focuses on network measurement and multimedia networking, such as dynamic video streaming and multi-party video conferencing. He is the winner of the ACM Internet Measurement Conference (IMC) Best Paper Award in 2012.



Hao Hu received the B.S. degree from Nankai University and the M.S. degree from Tianjin University in 2005 and 2007 respectively, and the Ph.D. degree from Polytechnic Institute of New York University in January 2012.

He is currently with the Advanced Architecture and Research group, Cisco Systems, San Jose, CA. He interned in the Corporate Research, Thomson Inc., NJ in 2008 and Cisco Systems, CA in 2011. His research interests include video QoE, video streaming and adaptation.

Yong Liu (M’02) is an associate professor at the Electrical and Computer Engineering department of the Polytechnic Institute of New York University (NYU-Poly). He joined NYU-Poly as an assistant professor in March, 2005. He received his Ph.D. degree from Electrical and Computer Engineering department at the University of Massachusetts, Amherst, in May 2002. He received his master and bachelor degrees in the field of automatic control from the University of Science and Technology of China, in July 1997 and 1994 respectively.

His general research interests lie in modeling, design and analysis of communication networks. His current research directions include Peer-to-Peer systems, overlay networks, network measurement, online social networks, and recommender systems. He is a member of ACM. He is currently serving as an associate editor for IEEE/ACM Transactions on Networking, and Elsevier Computer Networks Journal.

Zongming Guo (M’09) received the B.S. degree in mathematics, and the M.S. and Ph.D. degrees in computer science from Peking University, Beijing, China, in 1987, 1990, and 1994, respectively.

He is currently a Professor with the Institute of Computer Science and Technology, Peking University. His current research interests include video coding and processing, watermarking, and communication.

Dr. Guo is the executive member of China-Society of Motion Picture and Television Engineers. He received the First Prize of the State Administration of Radio Film and Television Award, the First Prize of the Ministry of Education Science and Technology Progress Award, and the Second Prize of the National Science and Technology Award, in 2004, 2006, and 2007, respectively. He received the Wang Xuan News Technology Award and the Chia Tai Teaching Award, in 2008, and received Government Allowance granted by the State Council, in 2009.

Yao Wang (M’90–SM’98–F’04) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1983 and 1985, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara, in 1990.

Since 1990, she has been with the Electrical and Computer Engineering Faculty, Polytechnic University, Brooklyn, NY (now Polytechnic Institute of New York University). She is the leading author of the textbook *Video Processing and Communications* (Englewood Cliffs, NJ: Prentice-Hall, 2001). Her current research interests include video coding and networked video applications, medical imaging, and pattern recognition.

Dr. Wang has served as an Associate Editor for the IEEE Transactions on Multimedia and the IEEE Transactions on Circuits and Systems for Video Technology. She received the New York City Mayor Award for Excellence in Science and Technology in the Young Investigator Category in 2000. She was a co-winner of the IEEE Communications Society Leonard G. Abraham Prize Paper Award in the Field of Communications Systems in 2004. She received the Overseas Outstanding Young Investigator Award from the National Natural Science Foundation of China in 2005 and was named the Yangtze River Lecture Scholar by the Ministry of Education of China in 2007.