Network Characterization and Perceptual Evaluation of Skype Mobile Videos

Shraboni Jana, Amit Pande, An Jack Chan and Prasant Mohapatra
University of California, Davis, CA 95616
Email: {sjana, pande, anch, pmohapatra}@ucdavis.edu

***Abstract*—We characterize the performance of both video and network layer properties of Skype, the most popular video telephony application. The performance in both mobile and stationary scenarios is investigated; considering network characteristics such as packet loss, propagation delay, available bandwidth and their effects on the perceptual video quality, measured using spatial and temporal no-reference video metrics. Based on 200+ live traces, we study the performance of this mobile video telephony application. We model video quality as a function of input network parameters and derive a feed-forward Artificial-Neural-Network that accurately predicts video quality given network conditions (0.0206 $\leq$ MSE $\leq$ 0.570). The accuracy of this model improves significantly by incorporating end-user mobility as an input to the model.**

I. Introduction

The use of mobile video telephony is on a rapid rise in both enterprise and consumer worlds. Many video telephony applications for mobile phones have been rolled-out in the market. Fring, Tango, Skype, FaceTime, Vtok, ooVoo are just a few examples of them. With improvement in network bandwidths (LTE 4G), video coding (HEVC or H.265 codec), device technology (quad-core processors) and other optimizations, we expect a big leap in the user expectations of the quality of experience (QoE) for video telephony applications in coming years.

It is an obvious fact, that the state of the network impacts the quality of video transmitted through it. If the network path suffers from significant jitter or loss, the perceived video quality at the smartphone gets deteriorated. Being delay-intolerant, video telephony apps have stringent network Quality of Service (QoS) requirements. It becomes quintessential from network operators perspective to examine the impact of underlying network on the performance of such video apps. In doing so, network operators can foresee the impact of QoS on end-users QoE and optimize the networking conditions accordingly.

Skype, with 663 Million subscribers as of 2011 [1], is the most popular video telephony app. Although some works have characterized the performance of Skype [2], [3], [4], [5], [6] based on its congestion-control, rate-control and VoIP traffic etc., to the best of our knowledge, very limited work has been done in terms of its perceptual video quality.

Smartphones make the treatment of video quality of video telephony apps different then what has been perceived conventionally. Unlike other end-devices, smartphone end-users have the flexibility to hold the end-device and move around during video conversation. Capturing packet level information is therefore insufficient in this context. This leads to the following questions:

1) What are the impacts of given networking conditions on the perceptual video quality of video telephony apps?
2) Does end-user mobility impact end-user perceptual video quality in interactive apps? Should we use different QoE prediction models for mobile and stationary end-users?

Although some findings we draw are specific to Skype, most of them are generic to all video telephony applications. The main contributions of this work can be summarized as below:

- With extensive experiments, we evaluate the perceptual quality of Skype video telephony in different network conditions;
- We measure the impact of user mobility on perceptual video quality;
- We present a feed-forward back-propagation Artificial Neural Network based non-linear model to effectively estimate Skype perceived video quality as a function of network conditions and user mobility. It gives high accuracy with low Mean-Square Error (MSE) for our QoE prediction models based on network QoS.

We substantiate our claims using 200+ live traces, each trace lasting for more than 4 minutes of Skype video conversations over smartphones.

Section II discusses the related work followed by description of performance metrics and experiment setup in Section III and Section IV respectively. In Section V, we present our experimental results and discuss its implications. In Section VI, we discuss prediction of perceptual video quality based on networking conditions and in Section VII, we conclude our findings. In this paper, smartphone video telephony, mobile video telephony and Skype have been used interchangeably and imply the same thing.

II. Related Work

There has been some work ([2], [3], [4], [5], [6]) studying Skype. In [2], authors evaluate the QoS level provided by Skype voice calls. The authors in [4] used ITU-T Recommendation G.1070 standardized opinion model for stationary Skype to study its video quality. The work uses a synthetic video with only head to shoulder newsreader movements for profiling Skype's behavior. The head-to-shoulder movements of end-users may not be the case with smartphones where the end-devices and end-users both may be in motion causing frequent background (video content) changes. Authors in [6] investigate QoS parameters and measure the QoE in terms of subjective assessment. But it considers only the voice-part of the Skype application. Another work [5] investigates Skype video in order to study its rate control against the unpredictable Internet bandwidth. The performance of four popular Instant Messenger (IM) clients - Skype, Windows Live Messenger, Eyebeam and X-Lite is analyzed in [3]. The paper reports Skype being superior to other three IMs. The authors in [7], compare perceptual voice quality of Skype and Google Talk.

Also, there have been attempts ([8], [9]) on estimating an application's video quality using deep inspection of video packets. In [8], authors proposed to monitor the quality of video transmitted over a packet network from the perspective of a network operator by detailed parsing of video stream. This is not feasible for video telephony apps as the video packets are encrypted specific to application provider. Real-time video quality in IP networks studied in [9] introduces a new video quality metric, which is evaluated using only network statistics and basic codec configuration parameters obtained offline. The work mainly relies on underlying full-reference assessment, PSNR. However, its impossible to have the original video for assessment at the receiver's side for smartphone video telephony apps.

To the best of our knowledge, none of these works, study perceptual video quality of Skype mobile videos based on networking conditions.

III. Performance Metrics

Skype being closed source, their compression algorithms and implementations are unavailable. The only application-layer information available is Skype's sending rate. The perceptual video quality metrics used are following -

a) **Blocking***:* Existing video compression standards such as MPEG-x and H.26x, have adopted block-based methods. In block-based method, Discrete Cosine Transform (DCT) is applied to the pixels in each block and then each block is independently quantized prior to encoding. During compression, each block being quantized independently may cause blocking artifacts. Also, packet losses may lead to full or partial loss of the block information. In such cases, the reconstruction of the video at the decoder will be erroneous, in turn causing visually apparent discontinuities across block boundaries. We implement the technique proposed in [10] for measuring the amount of blocking artifact in video traces captured during video telephony.

b) **Blurring***:* Blurring happens due to loss of high frequency information. Natural images have typically much lower energy at high frequencies. We implement the model proposed in [11] to evaluate the amount of blurring in the captured videos. Based on the histogram computations of the DCT coefficients of the the entire image, and filtering out the ones with zero DCT values, blurring metric is calculated. Finally, the metric is normalized to remove dependency on the image size.

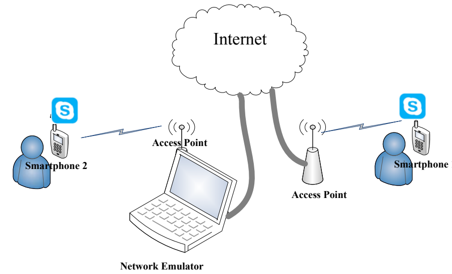


Fig. 1. Setup and layout of experimental environment

c) **Temporal Smoothness***:* Blocking and blurring evaluate the video quality in the spatial dimension. But, in addition of the spatial dimension, video also has the temporal dimension. Temporal information is the measure of the motion of objects in a video or movement of background including scene changes. We use the recently proposed metric TVM [12], to measure the temporal information of the video conversations. Numerically, TVM is calculated as log of mean square value of difference between two consecutive frames (F_{p-1} and F_p) of the video (measured in dB).

$$TVM_p = 10\log_{10}\left(\frac{k^2}{d}\right) \tag{1}$$

where k is the color depth of a video. It depends on the number of bits used to present a pixel in a video frame. $k = 255$ if 8 bits are used. d is the mean square difference of the corresponding pixels in two frames, F_{p-1} and F_p.

$$d = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(F_p(i,j) - F_{p-1}(i,j)\right)^2$$

Logarithm is used to compensate the non-linearity of human visual system.

High TVM value is indicative of low video motion. Packet loss events in network may lead to frame loss or partial loss. Partial loss can cause jerkiness which is captured by TVM and leads to lowering of TVM values. Number of full frame losses (F_{fl}) leads to video freezing. We propose to modify TVM to incorporate video freezing artifact, as follows:

$$\widehat{TVM} = TVM_p - \varepsilon \times \frac{F_{fl}}{F} \tag{2}$$

where F is total number of frames in video and ε is a constant which weights the negative impact caused by video freezing. It changes for videos with different content and length. In our video chat clips, we find that setting ε to 20 can effectively reflect the video freezing artifact in different scenarios.

IV. Experiment Setup & Methodology

We use an experimental approach to examine the relationship between the video quality of mobile Skype and the network-layer parameters. To examine these relationships, we set up a controlled testbed between two Android SAMSUNG Exhibit II smartphones. Smartphone 1 and Smartphone 2 are in different network domains.

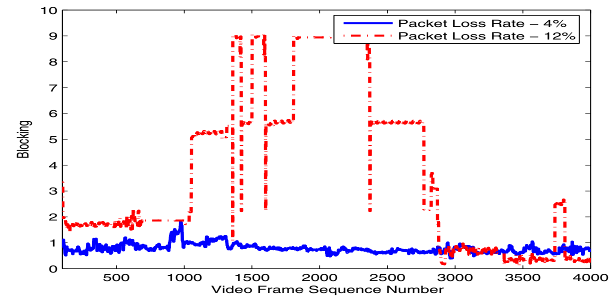
Figure 1 shows the experiment setup and experiment environment layout. The data collection is done for following two scenarios:

1) $Stationary-$ The video telephony can be conducted when end-devices are static as considered conventionally. We refer to this scenario as $Stationary$. In $Stationary$ scenario, end-users are free to do any body movements. In this scenario, both Smartphone 1 and Smartphone 2 are stationary during video telephony. But the end-users are free to have body movements.
2) $Movement-$ With the mobile video telephony, end-users have the option to hold the device and move around during video conversation. This highly likely scenario of video telephony with end-user movement will be referred as $Mobile$ scenario. Similar to $Stationary$ scenario, here end-users are also free to do any body movements. In this scenario, Smartphone 2 is recording Smartphone 1. End-user does brisk movements in the surrounding location within the range of their respective access point.

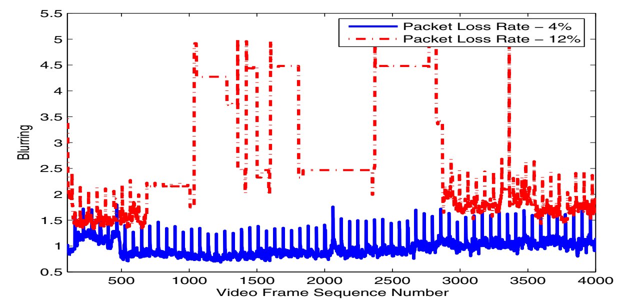
In both the scenarios, smartphone cameras are focused to capture facial part body movements only. Smartphone 2 is connected to a Campus Access Point. A HP compaq nc6000 laptop is also configured as an access point to which Smartphone 1 is connected. The laptop is equipped with Athores 802.11abg wireless cards operating using Madwifi driver. Each nc6000 has two Wifi antennas. The packets from and to the laptop access point are forwarded using Ethernet. The network emulator is also installed on the laptop. We use NETEM [13] for network emulation functionality. NETEM is used for testing protocols by emulating the properties of wide area networks. The network emulator is connected to the Internet via Ethernet. It emulates variable delay, loss, incoming and outgoing bandwidth by the command line tool 'tc' which is a part of the iproute2 package of tools. The packets are captured at both the smartphones using tcpdump.

We capture the video of the video telephony session using Screencast Video Recorder. Screencast captures smartphone screen at 21-22 fps and saves it into a high quality MPEG4 video. We use FFMPEG [14] to convert the captured video into sequence of bitmap image files loaded later into Matlab for further analysis. Video quality metrics - blocking, blurring and TVM were implemented in Matlab.

We did our experiments in a dedicated environment and at dedicated frequency channels. We did repeated experiments to measure the last access-link losses due to the end-user brisk movements with smartphones at the same location and found it to be negligible. This indicates that end-user mobility i.e. brisk walk within the range of access point doesn't induce packet losses or delays in the network and ensures that all 'effective' packet losses are occurring at NETEM and not at wireless hop.



(a) Blocking artifacts in videos



(b) Blurring artifacts in videos

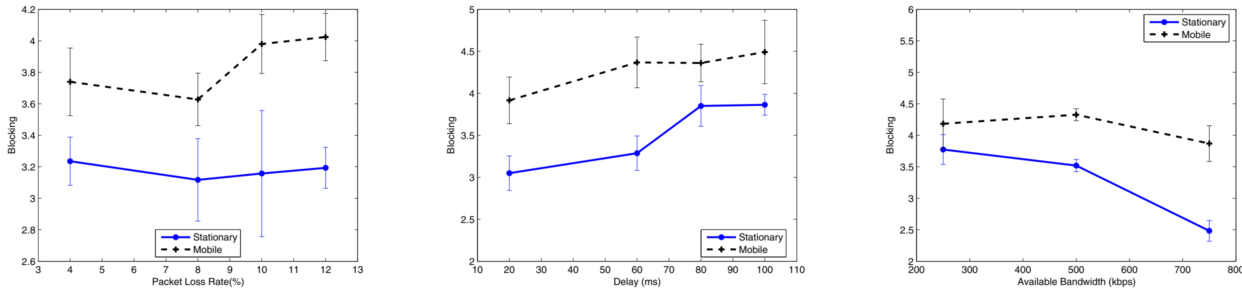
Fig. 2. Blocking & Blurring in video frames

V. Experimental Results

Video quality suffers due to the loss, delay and link constraints experienced by the packets in the network. Being delay intolerant, Skype primarily uses UDP protocol for data transmission. Hence, it is not possible to deduce actual loss or delay incurred by the video telephony packets from packet traces. We therefore, vary these network parameters at the network emulator, and take multiple traces in controlled experiments to study the impact on video quality in the $Stationary$ and $Mobile$ cases. The *Delay*, *Packet Loss* and *Available Bandwidth* mentioned in the figures hereon, are the settings of network emulator.

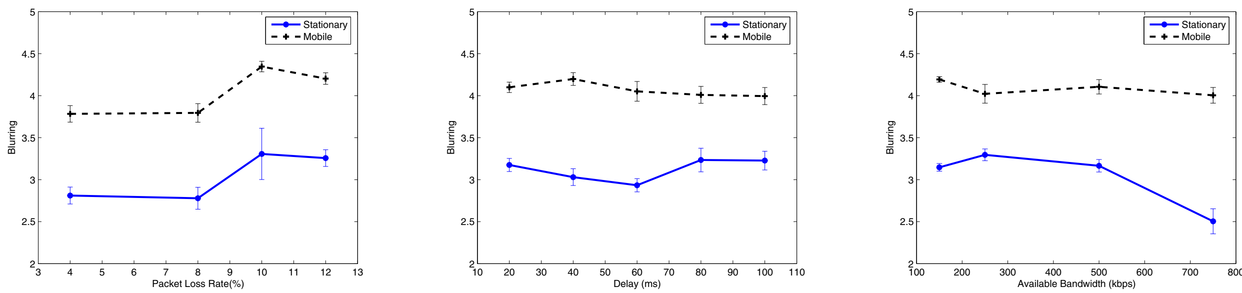
A. Spatial Metrics

Figure 2 shows the blocking and blurring measurements for two different packet loss cases. The traces were taken from two different instances of Skype video telephony under controlled network conditions. Uniform packet loss was inserted. Here, a single representable measure of blocking and blurring value for each frame of video received at the end device is reported. The initial images for both videos have somewhat similar blocking and blurring artifacts. As the video conversation session proceeds further, one of the videos shows more blocking and blurring compared to the other. This loss leads to lower value of blocking and blurring artifacts in the video both globally and also over the specific time frames of packet losses in a session. The objective measurements for the frame sequence corroborate with our subjective observations over the video frames, also reported in [15]. Thus, we can characterize the



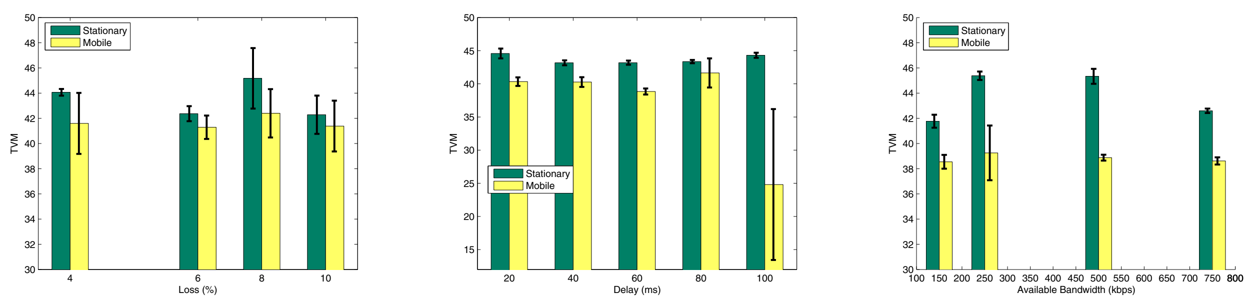
(a) Blocking vs Packet Loss (b) Blocking vs Delay (c) Blocking vs Available Bandwidth

Fig. 3. Blocking & Network Impairments



(a) Blurring vs Loss (b) Blurring vs Delay (c) Blurring vs Available Bandwidth

Fig. 4. Blurring & Network Impairments



(a) TVM vs Loss (b) TVM vs Delay (c) TVM vs Available Bandwidth

Fig. 5. Impact on Perceptual Video Quality vs Network Impairments.

quality of a video using-

$$Y^{block}_{avg} = \frac{\sum_{i=1}^{N} \frac{\sum_{k=1}^{K} Y_i^{block}(k)}{K}}{N}, \quad (3)$$

$$Y^{blur}_{avg} = \frac{\sum_{i=1}^{N} \frac{\sum_{k=1}^{K} Y_i^{blur}(k)}{K}}{N} \quad (4)$$

which are evaluated across the sequence of video frames. N is the number of video conversation traces. $Y_i^{block}(k)$ and $Y_i^{blur}(k)$, are the blocking and blurring experienced by the k^{th} frame of the i^{th} video conversation trace respectively. For our results, we consider K=1000 frames in descending order of their blocking and blurring values in a video conversation to consider worst-case scenarios. Y^{block}_{avg} and Y^{blur}_{avg}, the average blocking and average blurring respectively, are referred simply as *Blocking* and *Blurring* in the subsequent sections.

All the figures, henceforth, plot mean values of the y-axis metrics with 90% confidence interval.

a) Network Packet Loss: We vary the packet loss at the emulator, from 4% to 12%. We observe from our experiments that with higher packet loss rates ($\geq$ 12%), Skype refuses to establish end-to-end video conversation session. The two-way delay in the network emulator is fixed to 20ms. The link bandwidth of the emulator is $80Mbps$, way above than the required bandwidth for the smooth functioning of mobile video telephony application. The effect of network losses on spatial video quality are shown in Figure 3(a) and Figure 4(a).

The average blocking in Skype video remains unaffected in $Stationary$ scenario where as for $Mobile$ scenario, the average blocking artifacts increase with the loss rate of more than 8%. Unlike average blocking, average blurring increases at loss rate of more than 8% for both $Mobile$ and $Stationary$ scenarios. Thus, spatial impairments increase when the loss-rate is 8% or more. The blocking and blurring experienced by the $Mobile$ cases is $18\% - 25\%$ more than $Stationary$ cases.

b) Network Delay: Video telephony packets need to adhere to strict delay constraints of less than 150ms [16]. Increasing the bandwidth usage can be a straightforward solution for the applications that requires low network delay jitter for their video data. But this can be very costly and inefficient in terms of network resource-provisioning and usage. In addition, if the delay is more than the required limitation, the video telephony application fails to initiate video conversation or the video chat application freezes.

The two-way propagation delay at the emulator is varied from $20ms$ to $100ms$. The impact of network delay on spatial metrics is depicted in Figure 3(b) & 4(b). The network emulator link bandwidth is unconstrained and the packet loss rate is set to 1%.

Skype blocking increases with the delay of $60ms$ and higher whereas blurring is not effected by the increase in delay at the network emulator. It is worth noting that $Mobile$ and $Stationary$ scenarios greatly differ in perceptual video quality. Blocking and blurring artifacts are 23% and 33% higher respectively, in $Mobile$ scenario compared to $Stationary$ scenario for the same network delay.

c) Network Bandwidth Constraints: We change the two-way link bandwidth at the network emulator to $150kbps$, $250kbps$, $500kbps$ and $750kbps$. The packet loss rate is 1% and delay is $0ms$ at the network emulator. For link bandwidths lower than $150kbps$, Skype fails to establish connection. Intuitively, packet loss will occur due to link constraints.

As the available bandwidth is increased at the emulator, blocking artifacts for $Stationary$ scenario reduces by $\approx 17\%$ although there is only a marginal decrease in blurring impairments (Figure 3(c) & 4(c)). For $Mobile$ scenario, the decrease in blocking and blurring are very less with increasing available bandwidth. The $Stationary$ scenario has better perceptual video quality both in terms of blocking and blurring as the bandwidth constraints are relieved. For a network bandwidth of $750kbps$, $Mobile$ Skype spatial metrics perform $\approx 38\%$ poorly compared to $Stationary$ Skype.

B. Temporal Metrics

TVM values, as mentioned in the Section III decreases when the temporal smoothness of a video deteriorates. It is worth noting that temporal quality of a video is measured from the loss of temporal information. Although TVM of a single video clip does not exactly measure the temporal quality of this video telephony, its variation in different experimental scenarios can give us an indication of how the temporal smoothness is impacted by the application and network metrics. We observe from Figure 5, that TVM consistently gives larger values in $Stationary$ scenarios than for $Mobile$ scenarios.

Although all the network QoS metrics have impacts on the video smoothness, a TVM value of larger than 40 is generally regarded as a good smoothness. Among the network QoS metrics, delay in the network has comparatively major impact on temporal video quality. Network delay results in video stalling or freezing which creates very bad experience to end-user. We observe from Figure 5(b), that the TVM values decrease substantially ($\approx 45\%$) for $Mobile$ scenarios when the network delay is as large as $100ms$. In interactive video application, usually the end-to-end delay budget is 100ms as any delay longer than 100ms will be obvious to end users. Skype receiver may have dropped late packets in such a large network delay scenario, causing whole frame loss and leading to freezing and stalling.

TABLE I
CORRELATION COEFFICIENT QUALITY METRICS AND MOVEMENT.

	Cor_m	p-values	95% confidence interval
Blocking	0.3756	0.0000	[0.2491 0.4877]
Blurring	0.7624	0.0000	[0.6976 0.8148]
TVM	-0.3091	0.0000	[-0.4291 -0.1783]

Table I, gives correlation coefficients (Cor_m) of different video quality metrics with end-user movements. We consider, $Movement = 1$, for $Mobile$ scenarios and $Movement = 0$ for $Stationary$ scenarios. Blocking and Blurring are positively correlated to end-user movement whereas TVM is negatively correlated to the end-user movement. The small p-values in Table I indicate that this correlations observed from the experiments are very reliable. In summary, from the Figures 3, 4 and 5 we can conclude that spatial and temporal video quality degrades in $Mobile$ cases in comparison to $Stationary$ cases.

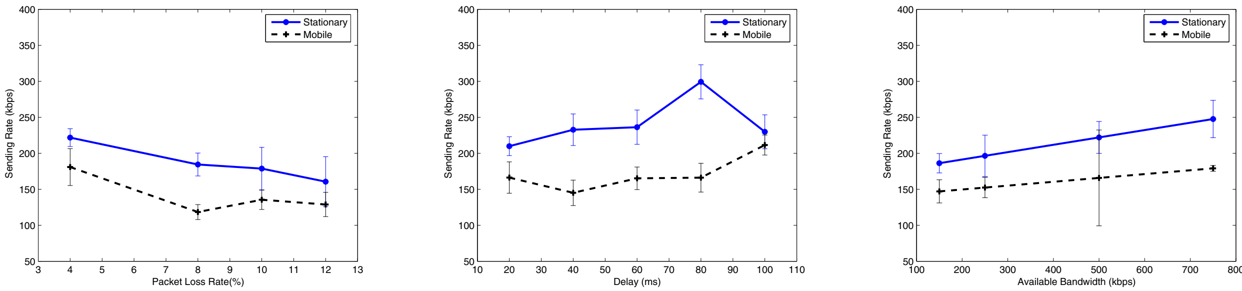
C. Sending Rate

Skype is a propriety application with undisclosed implementation. We therefore, study Skype's behavior by measuring its data sending rate to the network. Skype employs rate control algorithms depending on the congestion in the network. The sending rate of Skype under $Stationary$ scenario has been widely studied in the literature ([5], [4]) but to the best of our knowledge, none of the work studies Skype's sending rate under $Mobile$ scenario. We divide our discussion between these two scenarios -

a) Stationary: We observe from Figure 6(a), that Skype changes its sending rate as the loss rate in the network increases. When the loss rate is $\geq 8\%$ and less than $\leq 12\%$, Skype's sending rate is more or less constant around $120kbps$, the minimal data-rate which is required to carry out a smartphone video telephony conversation. Intuitively, we conclude, that Skype adapts its sending rate not only depending upon the network losses but also depending on the end-device requirements.

We observe from Figure 6(b) & 6(c), that Skype does not adapt its sending rate with increasing network delay whereas, Skype increases its sending rate with increased network bandwidth. The inconsistent result at $80ms$ delay, can be due to Skype or other unknown factors.

b) Mobile: We find the rate-control behavior of Skype under $Mobile$ conditions have similar trends as in $Stationary$

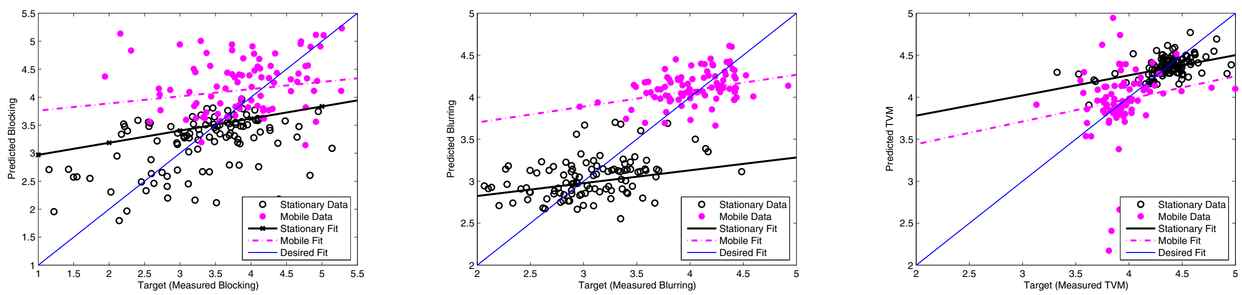


(a) Loss vs Sending Rate(kbps)

(b) Delay vs Sending Rate(kbps)

(c) Available Bandwidth vs Sending Rate(kbps)

Fig. 6. Impact of Sending Rate



(a) Blocking Prediction Model

(b) Blurring Prediction Model

(c) TVM Prediction Model

Fig. 7. Evaluation of Predicted Models

conditions (Figure 6),i.e., Skype's sending rate decreases with increased network losses and with decreased network bandwidth, but the average sending rate in $Stationary$ cases is always 50% more than the $Mobile$ cases for same networking conditions. This can happen provided losses, other than network emulator losses, incurred by $Mobile$ scenario is more than $Stationary$ scenario.

The data losses can be due to the congestion losses, random losses and video coding losses in the network.

- **Congestion losses** are due to the packets lost in end-to-end transit over the Internet. Congestion losses will impact both the scenarios equally.
- **Random losses** are caused by the wireless access-link. We did repeated measurements of the access-link losses with an end-user doing brisk movements in an area and an end-user being static. In both the cases, we found that the access-link packet losses in both scenarios differ negligibly. Moreover, all the experiments were carried out in a dedicated, unsaturated indoor wireless environment.
- **Video coding losses** are induced by loss of high frequency information (such as content motion) by the video codec. When user walks, there is a relative motion between camera and background and some jerky motion between camera and user. This leads to high frequency temporal content being created which will require higher bitrates to satisfy the application requirements. However, we observe a peculiar phenomenon - Skype uses lower bitrates for $Mobile$ scenarios. One possible explanation for this phenomenon is the Skype's rate control algorithm. However, this in turn will lead to truncation of video codec bit-rate leading to loss of high frequency information. This leads to probable increase in Blocking and Blurring in case of $Mobile$ scenarios.

VI. Modelling using ANNs

In this section, we analyze the prediction models for perceptual video quality. Our experiments show that influence of the network QoS and application sending rate is not linearly related, hence we use Artificial Neural Network (ANN), a non-linear, non-parametric and data driven modeling approach.

We use feed-forward back-propagation neural networks with one hidden layer to model the system. The first layer has 4 nodes corresponding to inputs v.i.z. network delay, loss rate, available bandwidth and application sending rate. The output layer has 1 neuron corresponding to the output being predicted. Each hidden layer is composed of simple elements (called neurons) and each neuron uses a non-linear transfer function to map inputs into outputs[17], [18]. The connections between neurons largely determine the network function. One can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. The number of hidden layer neurons used in our prediction models are mentioned in Table II.

The final layer produces the network's output. Such networks can be used for any kind of input to output mapping. The output of a feed-forward neural network with one hidden layer and one output neural network is given by

$$Y = \Gamma\left[\sum_{j=1}^{N_{hidden}} \omega_{j,o} \times \Gamma\left(\sum_{i=1}^{N_{input}} \omega_{i,j} \times X_i + b_j\right) + b_o\right]$$

where, $\omega_{a,b}$ denotes weight between link i and j; all the inputs to a node are summed up and passed through transfer function Γ. Input layer neurons uses $tansig$ (Tan-Sigmoid) transfer function. Thus, output (y) -input (x) relationship is given by:

$$y = \Gamma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

It introduces non-linearity into the system and has been observed as most pertinent transfer function for a number of applications [19].

Levenberg-Marquardt (LM) back-propagation optimization algorithm is used to update weights in the network while individual neuron uses gradient descent with momentum weight and bias ($learngdm$) learning function. It uses an approximate Hessian matrix and often a more efficient alternative to steepest ascent and its variations [18]. Here, the new weight ($w + \delta w_*$) is updated from previous step weight change (δw) based on momentum constant m and learning rate l

$$\delta w_* = m \times \delta w + (1 - m) \times l \times w$$

These functions are available for implementation as standard routines in Neural Network toolbox in Matlab.

Figure 7 shows the scatter-plot and regression curve for estimated values (based on ANN) of perceptual quality metrics for both $Stationary$ and $Mobile$ scenarios. It is evident that the $Stationary$ and $Mobile$ points on the scatter plots are quite distinct. Therefore, one can't use $Stationary$ model to estimate perceptual quality for $Mobile$ scenario and vice-versa. We can also infer from Figure 7 how blocking values are quite scattered around $y = x$ line while blurring values and stationary TVM values are close to $y = x$ line. Thus, blocking model has a higher MSE than others.

For brevity, we do not mention the weights and bias obtained in our prediction models. The performance of our ANN model is detailed in Table II. A very high accuracy was achieved. Typically, the more complex the relationship between inputs and outputs, the more epochs are required to fit an ANN. The prediction models for blocking, blurring and TVM require very few ($\leq$ 13) epochs to converge with low Mean Square Errors (MSE).

TABLE II
NEURAL NETWORK SETTING & PERFORMANCE

	Stationary	Mobile
Blocking	Neurons = 20, MSE = 0.351, Epoch = 10	Neurons = 20, MSE = 0.347, Epoch = 7
Blurring	Neurons = 10, MSE = 0.118, Epoch = 10	Neurons = 10, MSE=0.0377, Epoch = 6
TVM	Neurons = 20, MSE = 0.0206, Epoch = 9	Neurons = 20, MSE = 0.570, Epoch = 13

VII. CONCLUSION & FUTURE WORK

This work provides network characterization and perceptual evaluation of video telephony application Skype in $Mobile$ and $Stationary$ conditions. From extensive experiments, we find that delay impacts temporal video quality only when the network delay is as large as $100ms$ whereas, network loss has more impact on spatial video quality. Overall, we find that mobility plays an important role in determining end-user perceived video quality. Therefore, for accurate predictions of Skype's perceptual video quality, QoE prediction models must consider end-user mobility.

As a future work, it may be possible to include exact motion information using smartphone accelerometer sensor and build models for network bandwidth and delay under such mobility. This may enable network and cellular operators to get accurate feedback of end-user perceived video quality. It will also be interesting to study other quality issues in video telephony such as no-reference audio quality assessment and audio-video sync.

REFERENCES

[1] Skype grows FY revenues 20%, reaches 663 mln users. [Online]. Available: www.telecompaper.com/news/skype-grows-fy-revenues-20-reaches-663-mln-users

[2] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei, "Quantifying Skype user satisfaction," in *Proc. ACM SIGCOMM*, 2006.

[3] O. Boyaci, A. G. Forte, and H. Schulzrinne, "Performance of Video-Chat Applications Under Congestion," in *Proc. IEEE ISM*, 2009.

[4] X. Zhang, Y. Xu, H. Hu, Y. Liu, Z. Guo, and Y. Wang, "Profiling Skype video calls: Rate control and video quality," in *Proc. IEEE INFOCOM*, 2012.

[5] L. D. Cicco, S. Mascolo, and V. Palmisano, "Skype Video Responsiveness to Bandwidth Variations," in *Proc. NOSSDAV*, 2008.

[6] T. Hofeld and A. Binzenhofer, "Analysis of Skype VoIP traffic in UMTS: End-to-end QoS and QoE measurements," *Computer Networks*, Nov 2007.

[7] B. Sat and B. Wah, "Analysis and Evaluation of the Skype and Google-Talk VoIP Systems," in *Proc. IEEE ICME*, 2006.

[8] A. Reibman, V. Vaishampayan, and Y. Sermadevi, "Quality monitoring of video over a packet network," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 327 – 334, april 2004.

[9] S. Tao, J. Apostolopoulos, and R. Guérin, "Real-time monitoring of video quality in IP networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 5, pp. 1052–1065, Oct. 2008.

[10] Z. Wang, A. Bovik, and B. Evan, "Blind measurement of blocking artifacts in images," in *Proc. ICIP*, vol. 3, 2000.

[11] X. Marichal, W.-Y. Ma, and H. Zhang, "Blur determination in the compressed domain using DCT information," in *Proc. ICIP*, vol. 2, 1999.

[12] A. J. Chan, A. Pande, E. Baik, and P. Mohapatra, "Temporal quality assessment for mobile videos," in *Proc. ACM MOBICOM*, 2012.

[13] Netem. [Online]. Available: www.linuxfoundation.org

[14] FFMPEG. [Online]. Available: http://ffmpeg.org/

[15] Z. Wang and A. C. Bovik, *Modern Image Quality Assessment*, ser. Synthesis Lectures on Image, Video, and Multimedia Processing. Morgan & Claypool Publishers, 2006.

[16] U. It, "ITU-T recommendation G.114," International Telecommunication Union, Tech. Rep., 1993.

[17] H. Demuth and M. Beale, "Neural network toolbox for use with matlab," 1993.

[18] M. Hagan, H. Demuth, M. Beale *et al.*, *Neural network design*. Thomson Learning Stamford, CT, 1996.

[19] H. Yonaba, F. Anctil, and V. Fortin, "Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting," *Journal of Hydrologic Engineering*, vol. 15, no. 4, pp. 275–283, 2010.