Predicting Freezing of WebRTC Videos in WiFi Networks

Suying Yan[1], Yuchun Guo[1(✉)], Yishuai Chen[1], and Feng Xie[2]

[1] Beijing Jiaotong University, Beijing, China
[2] ZTE Inc., Shenzhen, China
{13120210,ychguo,yschen}@bjtu.edu.cn, xie.feng@zte.com.cn

Abstract. WebRTC is an open source project which enables real-time communication within web browsers. It facilitates web-based multimedia applications, e.g. video conferencing and receives great interest from the academia. Nevertheless understanding of quality of experience (QoE) for the WebRTC video applications in wireless environment is still desired. For the QoE metric, we focus on the widely accepted video freezing event. We propose to identify a freezing event by comparing the interval of receiving time between two successive video frames, named *F-Gap*, with a threshold. To enable automatically tracking of video freezing, we modify the original WebRtc protocol to punch receiving timestamp on the frame overhead. Furthermore, we evaluate the correlation between video freezing and quality of service (QoS) in WiFi network based on experiments in typical indoor environment. We build a machine learning model to infer whether QoE is unacceptable or not in the next time window based on current QoS metrics. Experiments verify that the model has good accuracy and the QoE state is mainly relevant to quality metrics of *Round-Trip Time*, *Link Quality* and *RSSI*. This model is helpful to highlight the providers in system design and improve user experience via avoiding bad QoE in advance.

Keywords: WiFi · WebRTC · QoS · Freezing · Machine learning

1 Introduction

Wireless video real-time communication (RTC) is becoming a killer application on mobile devices, such as Apple Facetime, Google Hangout, and Microsoft Skype, etc. Evaluation results of these applications are reported in [1]. Recently, the open source project WebRTC which enables RTC within webpages, has received great interest from both academic and industry. Most popular web browsers support WebRTC without the needs of installing extra software or plugin.

Supported by National Science Foundation of China under grant No. 61572071, 61271199 and 61301082.

J. Zheng et al. (Eds.): ADHOCNETS 2018, LNICST 258, pp. 292–301, 2019.
https://doi.org/10.1007/978-3-030-05888-3_27

WebRTC also attracts academic interest, such as implementation schemes of WebRTC [2], the congestion control mechanism for WebRTC [3], and video conferencing system design [4,5] based on WebRTC for general realtime communication or specialized purpose like tele-health. However, understandings of QoE for WebRTC video or RTC video is still limited [6] due to tediousness of the traditional methods of measuring user experience (e.g., MOS) corresponding to video quality (e.g., Peak Signal-to-Noise Ratio). Nowadays with the previllance of online service, it is widely accepted to characterize QoE with objective quality metrics, e.g., buffering rate or bitrate [7], which are easily to obtained in a large scale. Authors of [8] analyzed performance of WebRTC video in terms of throughput, jitter, and packet loss under different LTE scenarios. Authors of [9] focused on the comparison of smartphone configurations (e.g., CPU) on quality ratings under WiFi network. A recent study reported that the freezing event is an indicator of QoE that users care most [10]. Thus, in this paper, we focus on the occurence possibility of freezing event as a metric of WebRTC video QoE.

To predict WebRTC video freezing in WiFi networks, we need to answer the following three questions:

(1) *How to identify and track WebRTC video freezing?* Answer to this question is the first step for the prediction. We find that the time interval between two successively received frames, named *F-Gap*, can serve as a proper metric to identify a freezing event. However, it is non-trivial to obtain the value of F-Gap as WebRtc provides sending timestamps instead of receiving ones, but the sending time cannot be used due to the delay variance. Authors of [11] proposed to camera video playing screen with a stopwatch setting aside as timestamps and recover the timing text of each frame from the camera records with OCR (optical character recognition) tool afterwards. Thanks to the openness of WebRTC, we modify the original WebRtc protocol to insert receiving timestamps at each frame to enable the metric *F-Gap* to be obtained directly and the video freezing event to be identified in realtime.

(2) *How to build comprehensive measurements to evaluate the correlation of video QoE state with wireless quality?* To make this evaluation effective, we systematically design and conduct extensive measurement experiments in a typical indoor WiFi environment. During the experiments, we collect the values of F-Gap and two types of network QoS metrics: (a) wireless signal/link quality metrics, including Signal Quality, received signal strength indicator (RSSI), etc.; (b) network data transfer quality metrics, including packet loss rate and Round Trip Time. QoE state can be further inferred based on setting a proper threshold for the F-Gap.

(3) *How to predict QoE state of whether WebRTC's video freezing is unacceptable from wireless network's QoS metrics?* Based on the observation that wireless network quality correlates with the QoE state of whether the freezing ratio is unacceptable, we propose a practical model predicating video freezing event in the next time window based on the quality in current time window. This model can be used for the system to adjust service strategy in real time during a video call or for the user to avoid to access to the service if a freezing is predicted.

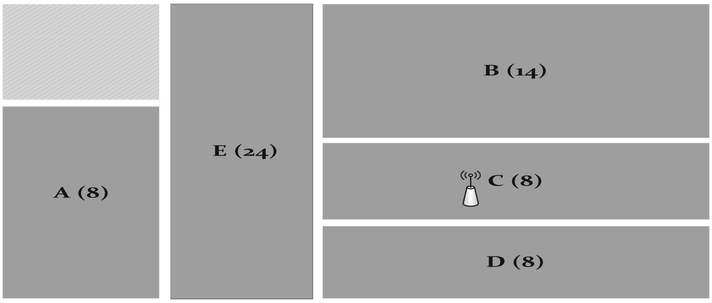

Fig. 1. Indoor measurement environment

In a word, our freezing evaluation method, measurement observation, and prediction model provide valuable insights for improving performance of wireless WebRTC-based video communication system. The remain of this paper is organized as follows. Section 2 describes our experimental methodology. Section 3 introduces our measurement results and the correlation analysis of the wireless network quality metrics and the proposed QoE metric in terms of video freezing state. Section 4 presents the freezing prediction model. Section 5 concludes this paper.

2 Measurement Methodology and Metrics

2.1 Testbed and Experiment Datasets

In this paper, we focus on the typical two-party WebRTC video chat widely used by users in WiFi environment. We set up a testbed consisting of laptops and a 802.11n wireless LAN AP. We modify the official open-source reference protocol of WebRTC to enable monitoring of video freezing events and network quality. We design another program to collect wireless quality. To ensure that the transmitted video contents are consistent and repeatable, we choose a high-definition (HD) video sequence *Big Buck Bunny*, widely used in video-related research, as the video sourceas. We inject this video sequence into WebRTC clients with a virtual video camera tool[1].

As most RTC communication takes place indoors with WiFi access, we consider the typical office usage environment and multi-room home environment as shown in Fig. 1. The AP is placed in room C. The white thick lines are the walls between rooms, and the gray blocks are our experiment spaces. Room A to D are typical office rooms with desks, chairs, computers, and other office supplies. Besides, each room is covered by several other WiFi APs which work in channels that different from our AP. We conducted independent experiments at each seat in these rooms within the AP's signal coverage range. We also divided the space of corridor (i.e., area E) into 62 blocks with of similar size and conducted 10

[1] e2eSoft. http://www.e2esoft.cn/vcam/.

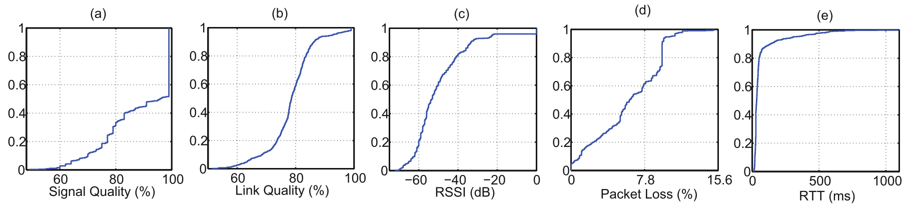

Fig. 2. Distribution of measured wireless network quality metrics.

experiments in each block. Totally we have 620 groups of basic experimental data.

Moreover, we invited 10 volunteers to conduct extra experiments with the following changes compared to the above basic experiments: (1) Use another AP of different type; (2) Play another video sequence with plentiful facial expression change; (3) Each student conducted 20 groups of experiments in random positions and accessed to the video sequence with his/her own laptop. Finally, we choose 50 groups of data with freezing events among these 200 groups of experiments as the extra dataset to verify our model.

2.2 Wireless Network Quality Metrics

To characterize the wireless network quality, we use the following metrics.

- Wireless signal/link quality metrics: We use all wireless physical layer metrics reported by Microsoft Windows 7 OS through its API, including *received signal strength indicator* (RSSI), *Signal Quality* (SQ), and *Link Quality* (LQ).
- UDP transportation quality metrics. As video transportation in WebRTC uses RTP over UDP, we measure *Packet Loss Rate* (Loss) and RTT.

The *cumulative distribution function* (CDF) for each wireless network quality metric is shown in Fig. 2. Our measurement covers a wide range of wireless network conditions. For instance, Fig. 2(c) shows the RSSI ranges from −70 dB to 0 dB, which is the general working range of WiFi network. Likewise, each of other metrics covers working range respectively as shown. Such a result verifies the effectiveness and generality of our measurement methodology.

2.3 QoE Metrics in Terms of Video Freezing

It is widely accepted that users of video services mainly care about the perceped fluency and clarity of video. Video's Structural SIMilarity (SSIM) index of a received frame with the transmitted frame is newly accepted metric of video clarity. However it is impossible to measure SSIM at a receiver client in a real-time scenario. On the other hand, video fluency in terms of freezing ratio is feasible to be measured with our modification of the WebRtc protocol by adding a timestamp of receiving time at the receiver side.

To identify a video freezing event, we first propose a metric *F-Gap*. We define *F-Gap* to be the *time interval or gap between two consecutively received video frames*. Then we compare it with the visual quality metrics. We find that the *F-Gap* is a good video freezing indicator. For a WebRTC video with frame rate of 30 frames/second, the regular interval of two successive frames is 33 ms. When the F-Gap is longer than 33 ms, there are some frames delayed or lost. Due to the limited visual sensitivity of human, short pause between two consecutive frames cannot be sensed by human. Therefore, the detection of freezing event is equivalent to find when the F-GAP is larger than a threshold. Specifically, to determine this threshold value, we ask volunteers to label freezing events they felt, and find that F-Gap of 1 s can be felt visually by human. Thus, We say it is "Freezing" when F-Gap >1 s, otherwise, we say it is "No Freezing".

We find that the F-Gap is correlated to video's Structural SIMilarity (SSIM) index.the freezing time approaches 20% of a session, SSIM would degrade about 0.172. This is because when the network condition is worsen, the video sender will decrease its video encoding rate to ensure the communication smooth. Besides, we change the threshold for F-Gap to 0.5 s, 2 s, 3 s, ..., 10 s, and find that such correlation between F-Gap and SSIM remains the same. This reveals that the F-Gap metric reflects the visual quality partially. Hence it is proper to choose the F-Gap to as the metric to identifying freezing.

Furthermore, the duration of freezing events in a time window above a fraction, say 10% or 30%, of the window is often considered unacceptable QoE. For the prediction of QoE, it is not feasible to make an realtime estimation of the exact time when a freezing occurs. Instead, we will show that it is feasible to make an prediction about whether the QoE is unacceptable or not in the next time window of some length, say 10 s.

3 Correlation Between Wireless Network QoS and Video Freezing

3.1 Statistical Perspective

In this section, we intend to find proper perspective to evaluate the relationship between WebRTC's user freezing and wireless network's quality metrics. Figure 3 plots the temporal variance of the five QoS metrics for an experiment conducted at a position in room B for 5 min. For clearness, we show parts of the result from 210 s to 260 s. The Freezing and No Freezing events are marked with blue '*' and black 'o', respectively. As shown in Fig. 3, the occurrence of Freezing event seems correlated with wireless network QoS. For instance, the Freezing seems correlated with wireless network QoS degradations, e.g. low RSSI and link quality (LQ). However, such a perspective on a single experiment cannot support drawing significant observation.

We then evaluate the correlation of wireless network quality metrics with video freezing statistically in all experiments. To obtain a macroscopic analysis of all experiments, we define a session (or a time window) of a video is of *unacceptable* QoE if the ratio of freezing time is greater than 30% of the whole session (or window).

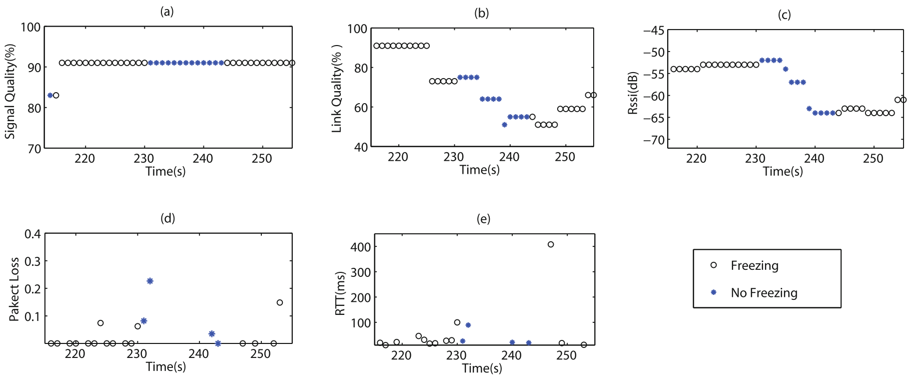

Fig. 3. Variances of wireless network QoS metrics in one experiment.

3.2 Feature Importance: Relative Information Gain

We then calculate the relative information gain [10] of the mean and variance of the five wireless network QoS metrics to F-Gap unacceptable indicator of all experiments, respectively. More specifically, Y denotes the random variable of QoE state (unacceptable, acceptable), X denotes the random variable of a QoS metric. For each random variable X for a QoS metric, we calculate the relative information gain (RIG) of Y against X as

$$RIG(Y|X) = \frac{H(Y) - H(Y|X)}{H(Y)},$$

where $H(Y)$ is the entropy of random variable Y and $H(Y|X)$ is the conditional entropy of Y given random variable X. The relative information gain quantifies how much uncertainty of knowing the F-Gap is unacceptable or not is reduced by wireless network QoS metrics. The higher the information gain, the more correlated the QoS metric is to the QoE state. Table 1 shows the result.

As shown in Table 1, the relative information gain of QoE state against the variance of RTT, the mean of RTT and variance of link quality are 0.136, 0.087 and 0.082 respectively. Thus, we conclude that *the video freezing relates to the wireless network quality metrics, in particular the variance of RTT*. Such a result suggests that the current WebRTC's video freezing problem is mainly due to the volatility of RTT. This finding is reasonable. Although WebRTC congestion control algorithm adjusts the video streaming rate for fluency partially based on variance of network latency, it cannot remedy excessive churns. However, none single QoS metric is strong enough to predicet QoE state so that we will choose to use these metrics integratedly to predict QoE state.

Table 1. RIG of network QoS metrics vs. QoE state.

Feature	RIG
RTT$_{-Variance}$	0.136
RTT$_{-Mean}$	0.087
Link Quality$_{-Variance}$	0.082
RSSI$_{-Mean}$	0.040
Link Quality$_{-Mean}$	0.035
Signal Quality$_{-Variance}$	0.031
RSSI$_{-Variance}$	0.026
Signal Quality$_{-Mean}$	0.017
Packet Loss$_{-Mean}$	0.012
Packet Loss$_{-Variance}$	0.009

Table 2. Feature importance of QoE models.

Feature	Importance
RTT$_{-Variance}$	0.23
RTT$_{-Mean}$	0.22
RSSI$_{-Mean}$	0.15
Link Quality$_{-Mean}$	0.13
Link Quality$_{-Variance}$	0.07
Signal Quality$_{-Mean}$	0.07
RSSI$_{-Variance}$	0.05
Packet Loss$_{-Variance}$	0.04
Packet Loss$_{-Mean}$	0.03
Signal Quality$_{-Variance}$	0.01

4 WebRTC Video Freezing Prediction Model

4.1 Model

We intend to build a machine learning model to predict the video freezing of a user's WebRTC video communication session from the wireless network quality metrics. An intuitive idea is to map the QoS metrics into the QoE state in same time window via training a classifier. However, such mapping is not effective in practice as it leave no time for making a scheduling decision accordingly and further deploying it. Hence, to make the prediction feasible and helpful in the network scheduling in practice, we intend to design a model to predict the QoE state in the future with present QoS condition considering the self-correlation of each metric to itself.

We propose a *video freezing prediction model* as follows. We use the measured wireless network QoS metrics in a current time window (say window A) to predict the video F-Gap *unacceptable* event in the next time window (say window B), as shown in Fig. 4. In Window A the wireless network quality metrics is collected historically for predicting the QoE in the next window, i.e. the Window B. As WebRTC use a 10-s video jitter buffer at the receiver side, we use 10 s as the size of window B. We can investigate the size of window A to obtain best prediction performance in our model training. The training and prediction can be done online. During a user's video communication process, we can keep collecting wireless network quality metrics, predicting freezing extent in the next time window, which can be used to in WebRTC's rate control algorithm to improve video playback continuity. Moreover, a WebRTC video call is started up at the magnitude of seconds for establishing connection. Therefore we can use the short window in the maganitude of seconds to estimate the QoE state for users and even make a space for making scheduling decision. This will be verified with the experiment results shown in Fig. 5.

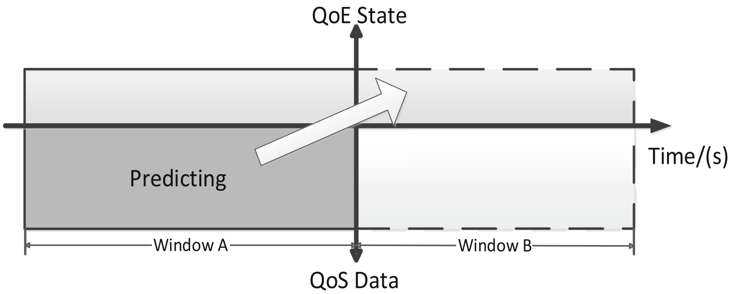

Fig. 4. Prediction window mechanism.

4.2 Performance

We use the wireless network quality features listed in Table 2 to train our video freezing model. For each window, we calculate the mean and variance of each metric, and then use them as the features. Thus, we totally have 10 features. Besides, we mix the basic dataset and the extra-dataset into an integrated dataset, 80% data are randomly selected as the training set and the remaining are used as testing set.

We use *Decision Trees (DTs), Random Forests (RandF), Support Vector Machines (SVM)* and *Extra-Trees classifier (ExtraT)* to train our models and compare their performance. We evaluate the effectiveness of classification methods in terms of the following indexs: *Precision*, *Recall* and F_1 *score* [12]. Among them, we use F_1 score as the main metric, as it is a comprehensive index which includes precision and recall. Moreover, the prediction accuracy of QoE bad is more important to avoid users' frustration of wrong prediction. Thus, we mainly compare the algorithms' F_1 score of QoE bad prediction results, and our results show the Random Forests method has the highest F_1 score for QoE bad prediction. After extensive experiments, we find that F_1 score returned by SVM is always below 0.3 and the performance of Extra-Trees and Decision Tree fluctuates widely with the size of sliding window A. Based on comparison, Random Forests method performs well and stably. Such a result is reasonable as Random Forests is ensembles of a number of decision trees and is the most successful general-purpose algorithm [13]. Thus, we finally select Random Forests model.

Figure 5 plots the performance of the Random Forests model against the size of sliding window. As shown in Fig. 5, as the size of sliding window A increases from 5 s to 120 s, the F_1 score of the prediction model gradually increases, meaning the model performs better when using more historical data. When the window size is of 5 s, the precision, recall and F_1 score are 87.3%, 60.8% and 7.21 respectively. When the widow size is larger than 17 s, the precision, recall and F_1 score keep relatively stable and larger than 90%, 80% and 0.8, which means the model is of high accuracy. For instance, when the window size is 20 s, the precision, recall and F_1 score of the model are 99.6%, 74.4%, and 0.84, respectively.

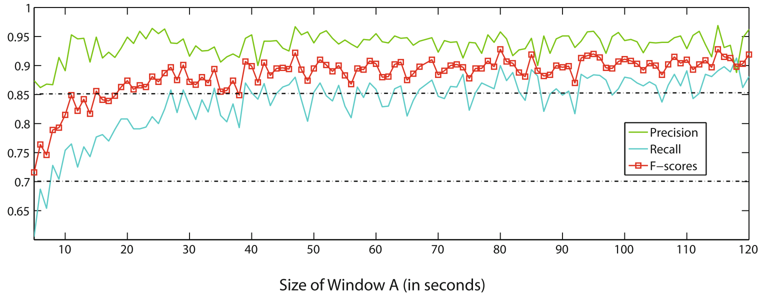

Fig. 5. Performance of QoE prediction model versus the size of Window A.

Thus, in practice, we suggest that the size of sliding window can be selected in the range from 20 to 30 s.

We list the features' importance of the Random Forests model in Table 2. As shown in Table 2, the RTT mean and variance are of top importance. Link quality and RSSI are also important metrics which represent quality on network level and physical level respectively.

5 Conclusions

In this paper, we studied the problem of accurate prediction of user video QoE of WebRTC in WiFi networks. First, we proposed a new, simple, and efficient QoE metric which is based on the time interval between two successive video frames. Second, we conducted 620 basic experiments and some extra experiments in an indoor WiFi environment and showed the strong correlation of WebRTC user QoE with wireless network QoS metrics. Finally, we built a machine learning models to predict a user's WebRTC video communication QoE state based on the current wireless network measurement results. The model can be used by a system to adjust its servicing strategy in real-time during a video call. Experimental result demonstrated that the model is accurate, with F_1 scores above 0.7 with 5 s of measurements and .84 with 20 s of measurements. Moreover, our analysis results and models clearly show that the current WebRTC implementation's QoE problem is mainly due to volatility of RTT. Our QoE evaluation method, analysis results, and prediction models provide valuable insights for wireless WebRTC video communication system design.

For more parameter settings, such as values of several thresholds, and the model targetted for multi-party meeting senario, we plan to make more investigation in the future work.

References

1. Sun, W., Qin, X.: End-to-end delay analysis of wechat video call service in live DC-HSPA+ network. In: Proceedings of International Conference on 6th Wireless Communications and Signal Processing (WCSP), Heifei, China, pp. 1–5. IEEE (2014)
2. Taheri, S., et al.: WebRTCbench: a benchmark for performance assessment of webRTC implementations. In: Proceedings of 13th Embedded Systems For Real-time Multimedia (ESTIMedia), Amsterdam, Netherlands, pp. 1–7. IEEE (2015)
3. De Cicco, L., Carlucci, G., Mascolo, S.: Experimental investigation of the Google congestion control for real-time flows. In: 1st Proceedings of the ACM SIGCOMM Workshop on Future Human-Centric Multimedia Networking, Hong Kong, China, pp. 21–26. ACM (2013)
4. Rodríguez, P., Cerviño, J., Trajkovska, I., Salvachúa, J.: Advanced videoconferencing based on WebRTC. In: Proceedings of 9th IADIS International Conferences Web Based Communities and Social Media and Collaborative Technologies, Lisbon, Portugal, pp. 180–184. IADIS (2012)
5. Jang-Jaccard, J., Nepal, S., Celler, B., Yan, B.: WebRTC-based video conferencing service for telehealth. Computing **98**(1–2), 169–193 (2016)
6. Balachandran, A., Sekar, V., Akella, A., Seshan, S., Stoica, I., Zhang, H.: Developing a predictive model of quality of experience for internet video. ACM SIGCOMM Comput. Commun. Rev. **43**(4), 339–350 (2013)
7. Balachandran, A., Sekar, V., Akella, A., Seshan, S., Stoica, I., Zhang, H.: A quest for an internet video quality-of-experience metric. In: 11th Proceedings of the ACM Workshop on Hot Topics in Networks, Seattle, WA, USA, pp. 97–102. ACM (2012)
8. Carullo, G., Tambasco, M., Di Mauro, M., Longo, M.: A performance evaluation of WebRTC over LTE. In: 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS), Cortina d'Ampezzo, Italy, pp. 1–6. IEEE (2016)
9. Vucic, D., Skorin-Kapov, L.: The impact of mobile device factors on QoE for multi-party video conferencing via WebRTC. In: Proceedings of 13th International Conference on Telecommunications (ConTEL), Graz, Australia, pp. 1–8. IEEE (2015)
10. Dobrian, F., et al.: Understanding the impact of video quality on user engagement. ACM **41**(4), 362–373 (2011)
11. Yu, C., Xu, Y., Liu, B., Liu, Y.: Can you see me now? A measurement study of mobile video calls. In: Proceedings IEEE 33rd INFOCOM, Toronto, Canada, pp. 1456–1464. IEEE (2014)
12. Joumblatt, D., Chandrashekar, J., Kveton, B., Taft, N., Teixeira, R.: Predicting user dissatisfaction with internet application performance at end-hosts. In: 32nd Proceedings IEEE INFOCOM, Turin, Italy, pp. 235–239. IEEE (2013)
13. STRATACONF Homepage. http://strataconf.com/strata2012/public/schedule/detail/22658. Accessed 4 Feb 2012