The Effects of Packet Delay Variation on the Perceptual Quality of Video

Selim Ickin, Karel De Vogeleer, Markus Fiedler, David Erman

4th IEEE Workshop On User MObility and VEhicular Networks (On-MOVE 2010)

2010 Denver, Colorado, USA

The Effects of Packet Delay Variation on the Perceptual Quality of Video

Selim Ickin, Karel De Vogeleer, Markus Fiedler, David Erman
School of Computing
Blekinge Institute of Technology
371 79 Karlskrona, Sweden
Email: {sic, kdv, mfi, der}@bth.se

***Abstract*—The satisfaction of end-users is important when evaluating services and products. Visualizing the network behavior in mobile streaming as well as modeling the correlation between Quality of Service (QoS) and Quality of Experience (QoE) is expected to improve user satisfaction of services on the Internet. Given that network and human factors are the elements that affect the final output of the measurements, it is important to take the Packet Delay Variation (PDV) into consideration. In this paper we observe the PDV during a series of real-life experiments on a 3rd Generation (3G) network while streaming videos. User Rating (UR) values from user input are recorded accordingly. With this aim, we implemented a QoE assessment tool that measures network metrics in kernel space, while simultaneously logging user ratings with regards to the perception of an ongoing real-time video on an Android phone.**

The primary goal is to identify a clear trend, so that we can assess the satisfaction of a user by deriving the QoE from measurable QoS metrics. We find that PDV degrades user perception. Observations show that, during the experiments, sudden breaks and restarts in the packet flow exist, which we call *on-off flushing* throughout this paper. Even though the use of Exponentially Weighted Moving Average (EWMA) assists in finding a model that shows the relationship between PDV and UR, *on-off flushing* is a major thread affecting this relationship. The results show that correlations with different quality can be reached with various modifications of the proposed matching model.

I. INTRODUCTION

Maximizing the happiness of customers is the goal of all service and product providers, in order to obtain more customers and to reduce the customer churn rate. The customer is not likely interested in knowing QoS metrics as Packet Delay Variation (PDV), wireless signal strength or packet loss, but rather video quality perception. Measuring the satisfaction of a customer is therefore essential, yet difficult to do. Customer feedback then plays an important role. Ideally, providers need a set of measures that can predict the satisfaction of their customers without the need of feedback. QoE models can satisfy this need. However, they are limited to specific cases and can be hard to obtain.

In this paper we present and analyze QoE traces for streaming video over the Internet through a wireless link. From these traces we try to derive a QoE model by statistical inference. The QoE model maps measurable QoS metrics onto the satisfaction of an individual. The models are meant to be used by decision making engines for network selection [10], monitoring applications, etc. In particular, these models are to be created for the decision making engine in PERIMETER, a STREP project granted by EU FP7. We present a QoS analysis of the PDV of a streaming video connection conducted in real-time. PDV is defined in [5] as the difference in one-way delay between packets, while ignoring any lost packets.

PERIMETER's main objective is to establish a new paradigm for user centricity in advanced networking architectures [19]. PERIMETER aims to provide mobility that is transparent to the user (seamless mobility) in heterogeneous networks and tackles the problem from a user-centric perspective. Therefore seamless mobility can be controlled by actual user needs in addition to business considerations. The presented measurement module supplies the PERIMETER framework with QoS statistics about ongoing connections. With the information obtained from all available networks, the framework takes decisions whether to roam to another network to satisfy the end-user perception with regards to the Always Best Connected (ABC) principle [10] and to maximize QoE.

The QoS statistics are ideally measured by low-level packet handling mechanisms in the Operating System (OS). This is of vital importance when statistics must be matched against network expectations [8]. Without providing accurate statistics to decision engines, *e.g.,* handover mechanisms, unnecessary actions may be triggered. This will increase the cost of backbone processes in the Internet infrastructure and delays towards the end-user [2]. Moreover, with the aim of achieving ABC transparently to the end-user, the processing time of the measurements plays a large role in real-time applications. Therefore, the preferred place to conduct network measurements is inside the OS, *i.e.,* in-kernel [16]. While doing QoS measurements the end-user preferably should not notice any degradation in performance. Also, privacy and security must be uncompromised and maintained at all times.

Throughout the paper, related work in Section II, theoratical aspects in Section III, implementation in Section IV and brief explanation of the experimental testbed in Section V will be presented. Then, we will focus on controlled real-life experiments to find out experimental models regarding the correlation of PDV to UR in Section VI-A, the benefit of using the EWMA techniques on human perception statistics in Section VI-B, the definition of Maximal Burst Size (MBS) and observations regarding the effect of MBS on the user perception in Section VI-D.

II. Related Work

Experimental and commercial frameworks have been presented before, trying to map QoS metrics to the end-user's experience for multimedia applications [14], [13], [21]. Respected Mobile Network Operators (MNO) deeply focus on similar research on QoE to reduce the number of rejected communication service products by potential customers in the market [17], [8]. Yet, not all of them can handle frame drops or decoding errors. Early work, however, uses machine-measurable metrics, *e.g.,*Bit Error Rate (BER), Peak Signal-Noise Ratio (PSNR) instead of QoE input from real-life experiments. More recent evolvement in research incorporates individuals in the QoE acquisition. A number of perceptual quality metrics have been developed to be used in technical systems for evaluating the image quality like L_p-norm, Structural Similarity (SSIM) index and Visiual Information Fidelity (VIF) [9].

Various studies have been conducted with the primary focus of understanding how the various levels of network behavior, with respect to Packet Delay Variation (PDV) and packet loss, can affect end-user perception on multimedia streaming services. Generally the contributions regarding this research question is to find out how sensitive are the individuals to the PDV. The User Rating (UR) was initially proposed to describe voice audio quality but can also be applied to video quality, specified by ITU-T recommendation [18]. The QoE of videos is generally given on a scale from 1 to 5. Here, 1 corresponds to *bad quality* whereas 5 is *excellent*. Table I presents the complete scale.

TABLE I
ITU-T SCALE OF MEDIA QUALITY IMPAIRMENT, ALSO REFERRED TO AS THE USER RATING (UR).[18]

Scale	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible, but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

QoS is the ability of a network to provide the assurred service level with the focus on parameters that exist in network and application level [17], [6]. Also, QoE is defined as "subjective and it relates to the actual perceived quality of the overall end-to-end performance of a service" [22]. Analyzing and modeling data is as important as collecting it. Finding a trend between the instantaneous QoS and QoE is a general problem [3].

III. Acquisition of QoS and QoE Metrics

The QoS metric of interest for our QoE model is the PDV. Calculating the PDV can be done in various ways. RFC3393 [5], however, proposes the algorithm outlined in Algorithm 1.

Algorithm 1 Computing Packet Delay Variation (PDV)

Calculates the Packet Delay Variation (PDV) of a packet stream
for each received packet **do**
  while Current time interval **do**
    $T_{S,n} \leftarrow$ departure time of packet n at sender's side
    $T_{R,n} \leftarrow$ arrival time of packet n at receiver's side
    Calculate D_n, update D and N (1)
  end while
  $PDV \leftarrow$ standard deviation of $T_{R,n} - T_{S,n}$ values collected per time interval (2)
end for

PDV can be calculated in many different ways, *e.g.,* as described in [5], [12] and [11]. Before settling for a PDV algorithm, we experimented and compared the output of different PDV algorithms. Ring Buffer (RB) algorithm yields the most accurate results with the fastest execution time. The RB is considered to be efficient in terms of buffer allocation during runtime. Other algorithms are storing packet parameters in a database, calculating the PDV with respect to each of the stored parameters at a later point in time. As a drawback, however, the RB algorithm does not store packet parameters longer than one time interval.

PDV calculations require end-to-end measurements between two peers. The unsynchronized one-way-delay of one packet D_n is calculated by the subtraction of the departure timestamp $T_{S,n}$ from the arrival timestamp $T_{R,n}$ obtained at both ends of a communication channel. The value of these metrics are stored at the reception of the first packets in each interval. The next packets' timestamps are compared with the initial stored values of the metric. Having the first packets as basis, we overcome the time synchronization problem.

$$D_n = T_{R,n} - T_{S,n} \tag{1}$$

We define the Packet Delay Variation (PDV) as the standard deviation of the end-to-end delays between packets that are measured within a given interval. The delays of the packets are stored for one second. The PDV J is then given by

$$J = \sqrt{\frac{1}{N-1}\left[\sum_{n=1}^{N}({D_n}^2) - N\overline{D}^2\right]}, \tag{2}$$

where D_n is the one-way-delay, $\overline{D}^2$ the average delay, and N the number of packets. J is updated each time a packet arrives. We note that the delay variation is being calculated by ignoring lost packets as described in the definition of one-way-delay variation in [5].

As a result of the real-life experiments applied to individuals, the PDV values are directly matched when the UR values are received. The user perception of the previous network directly influences later decisions of the user. Instantaneous matching did not satisfy enough to prove that there is high correlation. The forget factor, *the recency effect* [15] and sudden change of PDV during voting are possible reasons. Including the remaining effects of the previously obtained outputs into the calculation of the current output is possible by the EWMA approach [7]. When obtaining the UR, the EWMA is used for computing the correlation of instantaneous user perception against the QoS metrics. Thus, by using EWMA on PDV values, we tried to imitate the human perception process to a certain extent [7]. The positive impact of the EWMA on the quality of the matching between PDV and the user rating will become apparent in Section VI-B. The EWMA is computed as follows:

$$J_{\mathrm{EWMA}}(i) = (1-\alpha) \cdot J_{\mathrm{EWMA}}(i-1) + \alpha \cdot J(i), \qquad (3)$$

where $J_{\mathrm{EWMA}}(i)$ is the exponential weighted moving average, $J_{\mathrm{EWMA}}(i-1)$ the previous exponential weighted moving average PDV, and $J(i)$ is the current PDV. α is typically set to 0.25 [7].

IV. Implementation

A generic measurement module is used to assess the QoS metric. The measurement data are computed from information located in protocol headers. If the necessary information is not present, the header could be extended. Thus the module can piggyback on any suitable protocol that allows for extensible headers. IPv6 is a candidate for our measurement module as it allows for extensible headers by setting the *next header* field properly. TCP has similar abilities to extend the header. Tunneling protocols are also targeted for our measurement module as they easily allow methods for customizing headers, *e.g.,* Generic Routing Encapsulation (GRE), UDP over IP (UDPIP), or Layer2 Tunneling Protocol (L2TP). In our case we implemented the measurement assessing algorithm on top of a User Datagram Protocol (UDP) tunnel. The measurement module needs the following two fields to work properly; the *sequence number* and the *time stamp*. These fields are preferably 32 bit long.

It is important to send the QoS metrics from the sender to the receiver's side since the actual PDV computation is done at the receiver's side. Storing a history of QoS metrics is not feasible. Upon the reception of each packet, we rather update the average inter-packet arrival and departure times together with the deviation of the instantaneous values from the instantaneous means. PDV is calculated by deploying these updated values at the end of each time interval as depicted in Algorithm 1.

Any interested entities, usually residing in user-space, can obtain the PDV metric by accessing the measurement module in the kernel. Contact with the measurement module can be established through a local UDP socket or another OS specific system, *e.g.,* the `/proc` file system in LINUX.

V. Experimental Testbed

We have set up a testbed to obtain live QoS and QoE measurements. These values allow us to define a QoE model. In our experimental setup, depicted in Fig. 1, we streamed a video over the Internet from a server to an Android phone. The Android phone is connected to the Internet through 3G to the streaming server site. The 3G connection is a regular data subscription from a popular Swedish Internet Service Provider (ISP).

In Fig. 1 we see the streaming server that sends data over the Internet. The Android phone is connected to the Internet via a 3G access point.

We used the Darwin Streaming Server (DSS) framework [1] on a Linux (2.6.27) Ubuntu 9.04 Machine for streaming media to the Internet. The streamed video is MPEG-4 compressed with dimensions 240×180 pixels, has 24 kHz AAC stereo sound, 23.97 fps, and was streamed at a rate of 325 kbps.

The Real Time Streaming Protocol (RTSP) protocol [20] is used for streaming the video from the server to the Android phone. After the RTSP session is initiated, the RTSP client periodically sends RTSP requests to the server as a feedback control mechanism. A session identifier is used to keep track of the sessions as needed, which eliminates the need for permanent TCP connections.

We implemented the QoE assessment tool for Android that consists of measurement module synchronized to our video streaming application. The measurements are recorded by our kernel module, extracted and written to a file by the streaming video application. The files were then retrieved and analyzed with statistics software.

Using this tool, we conducted a set of experiments where a number of individuals rated a streamed video. While watching, they rated the video by pressing one of the five buttons on the touch screen corresponding to the the UR. In total, one hour of data was recorded coming from 15 adult test subjects. We conducted the experiments in diverse environments. The test subjects were selected from various backgrounds.

VI. Results and Observations

A. Packet Delay Variation

In Fig. 2, the relation between the measured PDV and UR values is depicted. The graphs show an excerpt of the time series obtained during our experiments.

$$UR = -0.88\ PDV\mathrm{ms}^{0.27} + 6.38 \qquad (4)$$

The model that best correlates the PDV and the UR is given in (4). When high PDV values appear, the user would see fuzzy images or freezing video from the most recently received video frames until the latest frame arrives. The latest frame would be played for a shorter time than expected in order to compensate the timing for the expected frame [4]. In case of packed loss, the lost frames will not be shown and the image will be skipped.

Causes for PDV are various queues on the way from the streaming server to the video player, but also the fact that our 3G ISP tries to compensate packet losses due to erroneous transmission by retransmissions, which increases PDV.

We also note that TCP is adequate in recovering from lost packets, but on the other hand it amplifies the PDV while doing so. This is one reason why TCP is unpopular among streaming video applications.

B. Exponential Weighted Moving Average

The UR obtained at a time is strongly impacted by the previously measured PDV values. The goodness of the fit, the

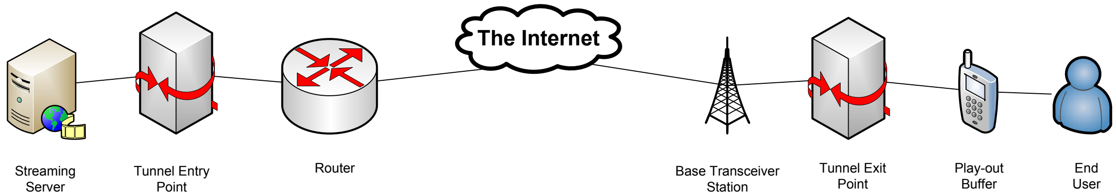
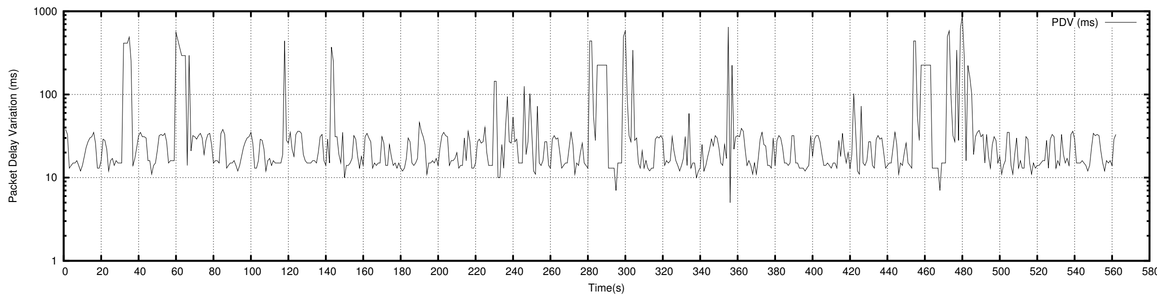
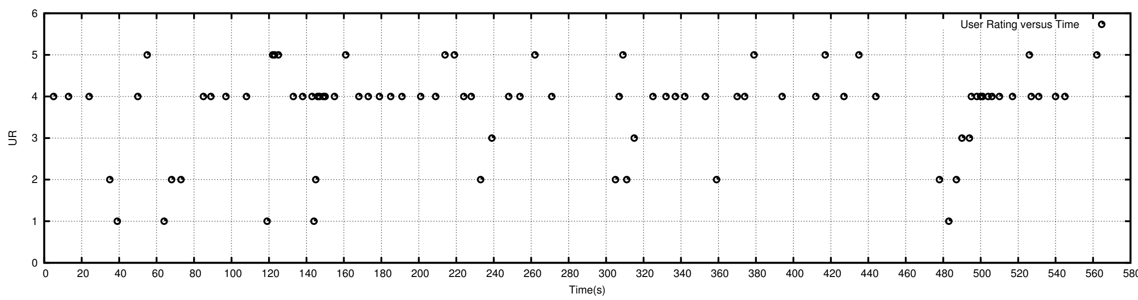


Fig. 1. Schematic overview of the experimental set-up.



(a) Measured PDV values.



(b) Measured UR values.

Fig. 2. Experiment excerpt of the time series of measured Packet Delay Variation (PDV) and User Rating (UR) values. There is a clear negative correlation between peaks in the PDV and negative peaks in the UR.

R^2 value, is improved from 0.25 to 0.51 when the EWMA technique is applied. This is an increase of goodness-of-fit of over $100\ \%$ as stated in Table II. The model evolved to (5) after EWMA is applied.

$$UR = -9.10\ PDV\text{ms}^{0.08} + 16.18 \tag{5}$$

TABLE II
CORRELATION BETWEEN THE USER RATING (UR) AND THE PACKET DELAY VARIATION (PDV) WITH AND WITHOUT EXPONENTIALLY WEIGHTED MOVING AVERAGE (EWMA) IS ILLUSTRATED AMONG DIFFERENT MODELS.

Model	Linear	Log	Exp	Power
Coefficient of determination	R^2	R^2	R^2	R^2
PDV w/ EWMA	0.31	0.42	0.40	0.51
PDV w/o EWMA	0.17	0.19	0.22	0.25

C. On-Off Flushing Behavior

An oddity we have observed during the experiments is the *on-off flushing* behavior. Data transfer is suddenly stopped without any noticeable visual warning signs. After a seemingly random amount of time a burst of data, that was supposed to be sent during the outage, is delivered. After the burst of data, the data transfer continues as before. During the *off* timespan the video on the screen freezes. As a consequence, very low UR values were registered.

In extreme case no packets arrives for about 30 seconds. This *on-off flushing* behavior degrades the correlation between the UR and the PDV results drastically. Fig.3 depicts two datasets where we fit the PDV values against the UR. In Fig.3(a) an excerpt of the unaltered dataset is plotted and fitted against a power function. The power model is chosen since it gives the maximum least square value, thus the best fit among other models as shown in Table II and Table III. The goodness-of-fit, R^2, for the whole dataset is 0.51. Collection of consecutive identical values indicates that the PDV remained unchanged and no packets are received during those intervals. This is the case when data loses occur, and we eliminate these values as described in Algorithm 2. When we ignore the *on-off flushing* behavior from the data set, see Fig.3(b), and compute the R^2 value again we obtain 0.68. This is an increase of goodness-of-fit of about $33\,\%$.

One reason for *on-off flushing* behavior could be that the

Algorithm 2 Tracing *on-off behavior* of a packet stream

```
Initialize variables
PDV_n = 0; ← current PDV
PDV_{n-1} = 0; ← previous PDV
counter = 0; ← counts PDV repetitions
Start tracing for duplicate PDV's.
for each obtained PDV value do
    if PDV_n = PDV_{n-1} then
        counter = counter + 1;
    else
        counter = 0;
    end if
    if counter > 3 then
        discardPDV; counter = 0;
    end if
    PDV_{n-1} = PDV_n;
end for
```

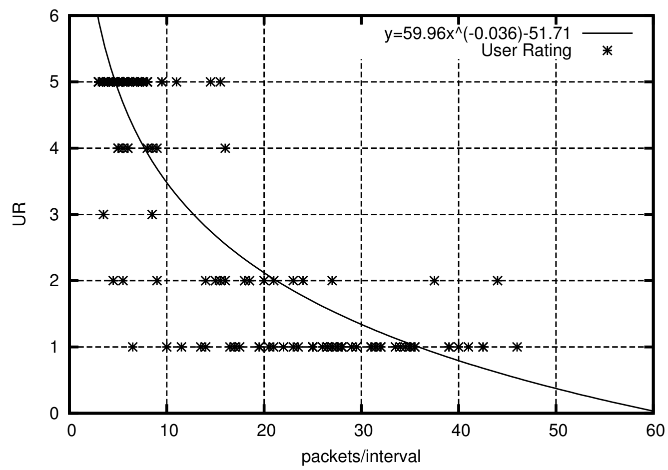
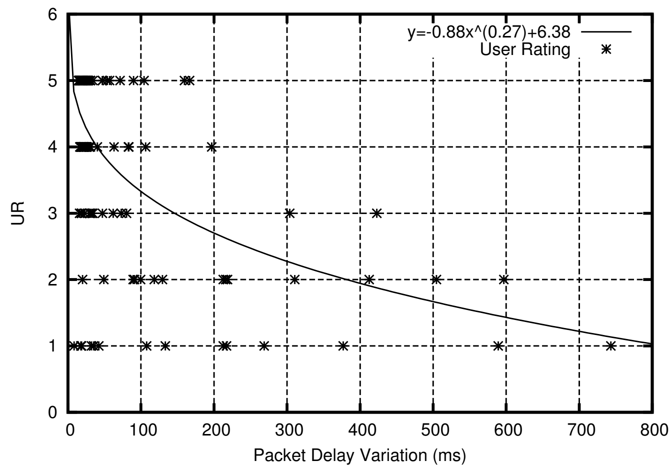
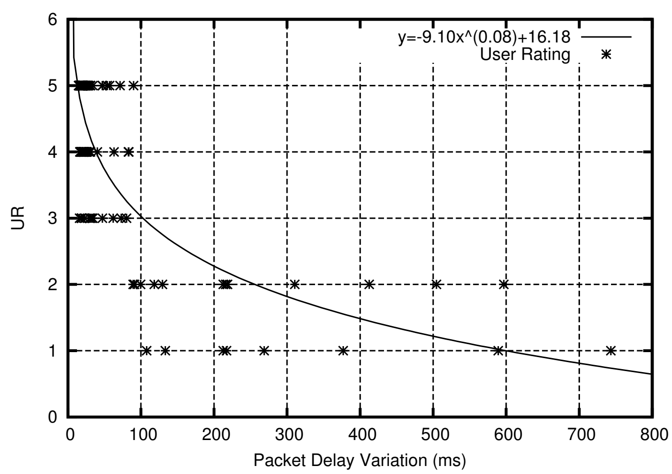


Fig. 4. Fitting of the UR against the Maximal Burst Size (MBS).



(a) Including *on-off flushing* behavior.



(b) Without *on-off flushing* behaviour.

Fig. 3. Fitting of the UR against the Packet Delay Variation (PDV) with the raw data, *i.e.,*with the *on-off flushing* effect, shown in 3(a), is improved to 3(b) when Algorithm 2 is applied to filter out the data affected by *on-off flushing* behavior. With this operation, in 3(b) a better correlation of determination (R^2) value is obtained; improved from 0.51 to 0.68. Corresponding model equations are denoted on the top right corner on both graphs.

3G network tries to recover from packet loss and only releases all packets when the lost packet are recovered.

Referring to the large difference in R^2 values, *on-off flushing* behavior can be seen as unwanted anomalies in the dataset. Another factor that could effect the correlation in a similar way is known as the *recency effect* [15]; user's perception changes faster from *good* to *bad* than from *bad* to *good*. This is reflected in the UR dataset as a skewed reaction to QoS degradation, thus decreasing correlation between the two. We haven't investigated, however, the *recency effect* on our dataset in detail yet.

D. Maximal Burst Size (MBS)

In previous research, measuring the delay variation is based on the assumption that the inter-departing times of the packets are uniform. This assumption does not tell, however, whether QoS degradation originates in the network or if the application layer is responsible. Inter-departure times of packets were not observed to be uniform in our experiments. For this reason and out of curiosity we observed the Maximal Burst Size (MBS). The number of packets departed within the same millisecond is traced. We then refer to MBS as the maximum amount of packets received within this time interval, which can also be considered as the *clumping size*. The MBS is plotted and fit against the UR in Fig.4. The model that we found is given in (6)

$$UR = 59.96MBS^{-0.036} - 51.71 \quad (6)$$

The UR is decreasing as the MBS is growing. The peaks in the number of transmitted packets per given time interval have a negative impact on the UR. This is a somehow surprising effect and can be explained by the fact that large MBS values indicate a flush following an *off* timespan. When we look at *pcap* traces we can identify MBS peaks. We distinguish two cases; short trails of identically-sized and very large sized packets with different content, and long trails of unusual differently-sized large packets. Only in the second case we see a severe video interruption as packet transmission is stopped when the streaming application sends feedback to the streaming server. Our first guess is that the streaming video application, at least on Android, is constrained to process a certain amount of data during a given time. Larger amounts of information, reflected in the MBS, will result in deterioration of the video quality. Thus the application might be the bottleneck. The causes of large MBS values and its effects are currently being under investigation.

TABLE III
THE CORRELATION FOR THE UR IN RELATION TO THE PDV AND MBS AMONG DIFFERENT MODELS

.

Model	Linear	Log	Exp	Power
Coefficient of determination	R^2	R^2	R^2	R^2
PDV w/ on-off flushing	0.31	0.42	0.40	0.51
PDV w/o on-off flushing	0.47	0.63	0.54	0.68
MBS w/o on-off flushing	0.73	0.77	0.75	0.78

VII. Conclusion and Future Work

In this paper, we presented the results and observations from a QoE assessment tool that consists of an in-kernel measurement module and interactive video streaming application. We conducted a set of experiments, focussing on PDV and showed that the system is adequate to predict UR values, related to the QoE of streaming video users. Fitting the QoS on QoE metrics in a mathematical model enables a decision maker, *e.g.,* for vertical handover, to participate in seamless communication with the goal of maximizing end-user satisfaction.

The importance of utilizing EWMA during the analysis of data related to the human perception is presented. From the results of our experiments we concluded that applying EWMA techniques to our collected data increases the goodness-of-fit of the power model by 100 %. In addition, on-off-flushing behavior and its effects on the correlation between the PDV and the UR values are presented. The goodness-of-fit of the model is observed to be increased by 33 % after removing the *on-off flushing* on the network with Algorithm 2 during measurements.

An additional parameter, the MBS, is measured and correlated with UR values. MBS which is measured on the receiving side shows the density of the packets departed from the streaming server within 1 ms time interval. The analysis regarding MBS is under investigation. We have implemented additional assessment algorithms for metrics such as the Packet Drop Rate (PDR) and Packet Reordering Rate (PRR), these are also currently under investigation. Appropriate QoE models for these QoS metrics are being evaluated by statistical inference.

Even though the experiments and observations in this paper focus only on 3G network so far, we have done similar research on other networks *e.g.,*Wireless Local Area Network (WLAN). However, this is under investigation. With the completion of the ongoing research, we will be able to obtain QoS and QoE model from all available wireless networks. As a support for vertical handover mechanisms, the complete information, obtained from within different UDP tunnels which are bound to WLAN and 3G interfaces, will be simultaneously provided to the decision maker to point out the best available network.

Acknowledgment

The authors would like to thank PERIMETER, a STREP project, granted by EU FP7 to fund our research on QoE assessment.

References

[1] Apple. http://developer.apple.com/opensource/server/streaming, 2010.
[2] Patrick Arlos. *On the Quality of Computer Network Measurements*. PhD Dissertation, Blekinge Institute of Technology, 2005.
[3] Zhan Chen and Hidenori Nakazato. A model for time varying quality of speech services. *IEEE Globecom*, pages 240–244, 2005.
[4] Mark Claypool and Jonathan Tanner. The effects of jitter on the perceptual quality of video. In *In Proceedings of the ACM Multimedia Conference*, pages 115–118, 1999.
[5] C. Demichelis and P. Chimento. IP Packet Delay Variation Metric for IP Performance Metrics (IPPM). RFC 3393 (Proposed Standard), November 2002.
[6] Markus Fiedler, Tobias Hossfeld, and Phuoc Tran-Gia. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, 24(2):36–41, March 2010.
[7] Frederic Guyard and Sergio Beker. Towards Real-Time Anomalies Monitoring for QoE Indicators. Annals of Telecommunications. volume 65. No. 1–2, Jan/Feb 2010.
[8] Bjorn Hestnes, Peter Brooks, and Svein Heiestad. QoE (Quality of Experience) - measuring QoE for improving the usage of telecommunication services. Technical Report R 21, Telenor, September 2008.
[9] Muhammad Imran Iqbal, Hans-Jürgen Zepernick, and Ulrich Engelke. Perceptual-based quality assessment of error protection schemes for wireless jpeg2000. In *ISWCS'09: Proceedings of the 6th international conference on Symposium on Wireless Communication Systems*, pages 348–352, Piscataway, NJ, USA, 2009. IEEE Press.
[10] Lennart Isaksson. *Seamless Communications*. PhD thesis, Blekinge Institute of Technology, 2007.
[11] Ren Jiuchun, Maoa Dilin, and Wang ZhiWei. A neural network based model for voip speech quality prediction. In *ICIS '09: Proceedings of the 2nd International Conference on Interaction Sciences*, pages 1244–1248, New York, NY, USA, 2009. ACM.
[12] Juniper Networks Inc. Understanding the Basic Networking Functions, Components, and Signaling Protocols in VoIP Networks. White paper, May 2007.
[13] Jirka Klaue, Berthold Rathke, and Adam Wolisz. Evalvid - a framework for video transmission and quality evaluation. In *In Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 255–272, 2003.
[14] J. Lubin. A human vision system model for objective picture quality measurements. In *International Broadcasting Convention*, number 447, pages 498–503, September 1997.
[15] N Miller and D.T. Campbell. Recency and primacy in persuasion as a function of the timing of speeches and measurements. *Journal of Abnormal and Social Physchology*, 59:1–9, 1959.
[16] Koh Minghao, Khong Yun Chyang, and E.K. Karuppiah. Performance analysis and optimization of user space versus kernel space network application. In *Research and Development, 2007. SCOReD 2007. 5th Student Conference on*, pages 1–6, Dec. 2007.
[17] Nokia Corporation Networks. Quality of Experience (QoE) of mobile services: Can it be measured and improved? White paper, Nokia, 2004.
[18] ITU-T Rec. P.800. *Methods for subjective determination of transmission quality.* Recommendations. International Telecommunications Union, Geneva, Switzerland, 1996.
[19] Perimeter. http://www.ict-perimeter.eu/, 2009.
[20] H. Schulzrinne, A. Rao, and R. Lanphier. Real time streaming protocol (*RTSP*). RFC 2326, April 1998.
[21] Andrew B. Watson, James Hu, and John F Mcgowan Iii. Dvq: A digital video quality metric based on human vision. *Journal of Electronic Imaging*, 10:20–29, 2001.
[22] DSL Forum Technical Work WT-126. Video services quality of experience (QoE) requirements and mechanisms, 2007.