Audiovisual network service optimization by quality of experience estimation

Johanson, Mathias; Jalminger, Jonas; Laulajainen, Jukka-Pekka; Bür, Kaan

Published in:
Proceedings of NEM – Networked and Electronic Media Summit

2012

Document Version:
Peer reviewed version (aka post-print)

Audiovisual Network Service Optimization by Quality of Experience Estimation

Mathias Johanson[1], Jonas Jalminger[1], Jukka-Pekka Laulajainen[2], Kaan Bür[3]

[1]Alkit Communications, Mölndal, Sweden; [2]VTT Technical Research Centre of Finland, Oulu, Finland; [3]Lund University, Lund, Sweden

E-mail: [1]{mathias, jonas}@alkit.se, [2]jukka-pekka.laulajainen@vtt.fi, [3]kaan.bur@eit.lth.se

Abstract: **With the growing popularity of audio and video communication services on the Internet, network operators, service providers and application developers are becoming increasingly interested in assuring that their services give the best possible experience to the users. Since real-time audio and video services are very sensitive to packet loss, latency and bandwidth variations, the performance of the network must be monitored in real time so that the service can be adapted to varying network conditions by mechanisms such as rate control, forward error correction and jitter buffer adaptation. However, in order to optimize a service in terms of the user's experience, the subjective effect that various network perturbations have on the user should be taken into consideration in the service adaptation mechanism. In this paper we present a novel approach to performance optimization based on rate adaptation driven by real-time estimation of the subjective Quality of Experience of a videoconferencing service. A proof-of-concept service optimization framework consisting of network monitoring, quality estimation, rate adaptation and service optimization mechanisms is presented and a testbed configuration based on network emulation is described and used for evaluation. Our initial experiments show that the approach is viable in practice and can substantially improve the Quality of Experience of real-time audio-visual services.**

Keywords: Quality of Experience, video communication, congestion control, service optimization.

1 INTRODUCTION

Audio and video communication services are typically very sensitive to variations in bandwidth, packet loss and latency. Consequently, in order for such services to work well in IP-based networks without a guaranteed Quality of Service (QoS), sophisticated end-to-end mechanisms are needed to monitor the network for perturbations and predict how a perturbation will affect the users of the service. Based on the monitoring of the network and the predicted subjective Quality of Experience (QoE), the service can be adapted in real time to maximize the quality experienced by the users.

Whereas adaptive multimedia communication services have been studied for a long time [1, 2, 3], the introduction of a QoE model in the adaptation process, in order to capture the effect of network perturbations on the user, is a novel concept. The rationale is that by considering not only the monitored network parameters (e.g. loss rate, latency), but also the way these parameters affect the user, the service optimization can be performed in a way that is perceptually preferable. In this paper we describe one such effort, wherein a videoconferencing service is extended with QoE prediction mechanisms for conversational audio and video, which is used to optimize the performance of the service based on the observed network conditions.

Different methods and metrics are available for QoE estimation and prediction. One approach explored in this work is called Pseudo-Subjective Quality Assessment (PSQA) [4], which is based on a neural network that has been trained through subjective QoE tests. With this approach, the application (i.e. the videoconferencing tool) measures the network parameters and calls a function that applies the parameters to the neural network, which in response returns an value between 1 and 5, representing the Mean Opinion Score (MOS) of the test panel that rated similar media streams with similar network conditions when the neural network was trained. The knowledge about how human subjects will experience media streams of different quality is thus encoded in the neural network. The advantage of this approach is that it gives a good correlation between subjective experience and network conditions and that it can be used in real time (which of course is necessary for the application considered here). The disadvantage is that it is time consuming and costly to conduct the subjective tests to train the neural network.

As the QoE of media streams is being estimated, the service optimization algorithm decides what can be done to improve the current situation, i.e. to maximize the QoE. If the currently estimated QoE is above a certain level, the proper action might be to do nothing (i.e. good enough quality). If the QoE is below some other (lower) threshold, the only reasonable thing to do might be to recommend the user to try again later (i.e. too bad quality to be even feasible). Between the extremes, the application should optimize the performance by adapting to the network conditions experienced. The most common service optimization mechanisms include codec rate adaptation (i.e. adjusting the sending rate or transcoding rate), adaptive Forward Error Correction (FEC) and stream shaping mechanisms.

Corresponding author: Mathias Johanson, Alkit Communications AB, Sallarängsbacken 2,SE-431 37 Mölndal, Sweden, +4631675543,mathias@alkit.se

To study the opportunities with QoE-driven service optimization, we have developed an experimentation and demonstration testbed, wherein Real-time Transport Protocol (RTP) audio and video streams originating from a videoconferencing system are sent through a network emulator to introduce perturbations in a controlled way. The prototype algorithms for QoE estimation and adaptation integrated in the videoconferencing tool and in the RTP reflector used for multipoint conferencing are then studied for different network conditions, and the performance of the optimization mechanisms is evaluated. Our testbed configuration also includes an external measurement system for verification of the network conditions, i.e. to make sure the network performance measured by the built-in probes, driving the QoE estimation and service optimization mechanisms, are correct and have the desired effect on the network.

Initial experiments show that the mechanisms developed have a considerable positive impact on the perceived quality of videoconferencing sessions in networks with large fluctuations in bandwidth, latency and loss rate.

2 MEASURING QUALITY OF EXPERIENCE

Recently, the term Quality of Experience (QoE) has emerged to complement the traditional concept of Quality of Service (QoS) for assessing the quality of networked services. Whereas the notion of QoS only takes technical parameters into account, such as packet loss rate or latency, the concept of QoE also includes the effect these performance metrics have on the user of the service [5, 6].

When assessing the experienced quality of a service, the available mechanisms can be broadly classified as *objective, subjective* or *hybrid* approaches. Objective methods are based on statistical or mathematical models for calculating how well a signal that is distorted (e.g. by transmission over a noisy communication channel) corresponds to the original. For instance, the Peak Signal to Noise Ratio (PSNR) is a very common objective quality metric for image and video communication services. Subjective methods, on the other hand, rely on having a panel of test subjects rate the perceived quality of media sequences in a controlled environment. Subjective quality is usually quantified by a value between one and five, representing the Mean Opinion Score (MOS) of the test panel. Hybrid methods, incorporating both objective and subjective elements, are typically based on utilizing some form of Machine Learning (ML) technique that is trained using subjective tests.

Quality assessment methods can also be classified according to what kind of *reference* is available when doing the assessment. In Full Reference (FR) models, the original, undistorted media is available for comparison with the transmitted, distorted media. This allows for a detailed, offline analysis of the objective or subjective difference between the original (reference) signal and the recreated far-end signal. In No Reference (NR) models, there is no reference signal available to compare the recreated signal to. Finally, in Reduced Reference (RR) models, partial information about the original signal is available for comparison against the recreated signal.

Although FR metrics typically give the most reliable results, they are inherently incompatible with the application of interest here, since the original media is not available. For such real-time control purposes as we are focusing on in this paper, a NR hybrid approach is the most suitable.

3 SERVICE OPTIMIZATION FRAMEWORK

As discussed above, the adaptive service optimization concept is based on continuously monitoring the network state to predict the QoE, i.e. the quality of the media experienced by the user, and then adapt the application to optimize the QoE. The framework we have developed to test the concept is based on five main components:

- A network monitoring framework, based on probes measuring parameters like throughput, loss rate, latency and jitter
- A QoE estimation framework for audio and video streams
- A service adaptation mechanism, for maximizing the perceived QoE of the application for each state of the network
- A video communication platform based on the software products Alkit Confero (videoconferencing end system) and Alkit Reflex (RTP reflector), wherein the monitoring and service optimization components are integrated
- A network emulator used to introduce network perturbations to observe the performance of the service optimization and QoE estimation mechanisms.

3.1 QoE Estimation and Prediction

For adaptation purposes, an accurate no-reference QoE model is needed in order to obtain estimations in real time. Our QoE estimation framework is based on the use of PSQA, which is a parametric methodology for estimating perceived quality. It works by mapping network- and application-layer parameters having an effect on quality to subjective scores. The parameters used as input to the estimator are:

- Codec, bitrate
- FEC
- Packet loss rate
- Temporal distribution of losses (mean loss burst size)
- One-way delay
- Jitter

The mapping from the parameters to the subjective scores is done by training a Random Neural Network (RNN) to learn the relationship between the input parameters and the subjective quality. A subset of input parameter

combinations is carefully selected, a subjective assessment campaign conducted for each of them in an emulated network, and the MOS values recorded for each combination. Once the RNN is trained with the results from the subjective assessment, it can give good estimations not only for the parameter combinations used in the subjective test, but also for other parameter combinations within the range of parameters used in the training. The use of a trained RNN for MOS estimation is computationally trivial and gives very high correlation with subjective scores.

It should be noted that, because the quality value resulting from the RNN corresponds to the actual user experience, the quality of the original signal (i.e. codec, bitrate) has an effect on the MOS values. In practice, MOS values of 5 cannot be achieved and MOS above 4 is considered toll-quality.

3.2 Audiovisual Service Optimization

The service optimization algorithm is implemented in the videoconferencing end system. It works by continuously feeding data from the network monitoring component into the PSQA algorithm, which returns a MOS-like score of the estimated quality. Currently, this is done only for the audio streams received. The estimated audio quality is then used as a trigger for the actual optimization algorithm, which is implemented both in the sender side of the end system and in the RTP packet reflector. This is to allow the same mechanism to be employed both when a reflector is used and when there is no reflector (i.e. point-to-point or multicast sessions.) Many different adaptation events can be envisioned in response to QoE changes. The currently implemented actions are to trigger rate adaptation for the video streams received by the end system and to trigger a modality change from audio/video to audio only (and vice versa). The events are triggered when the score returned by the PSQA algorithm passes two threshold levels.

When the calculated MOS decreases below 3, the bandwidth adaptation mechanism in the reflector (or in the sender if no reflector is used) is triggered. This is done using RTP Control Protocol (RTCP) packets containing the monitored loss rate and throughput values, which makes the bandwidth adaptation algorithm reduce the rate of the video stream. The fact that the QoE estimation of the audio stream is used to trigger an adaptation event related to the video streams might seem strange at first, but the rationale for this is that in most videoconferencing situations, the cause of audio quality degradation is in fact the video consuming too much of the available bandwidth, leading to packet loss both in the audio and video streams. In other words, degradation in audio quality usually indicates degradation in video quality as well. By having the video bandwidth reduced, the audio stream quality is improved. Since the bandwidth of video streams is usually much higher than the audio bandwidth, only a slight reduction in video bandwidth can make a big difference in terms of audio quality. Moreover, video codecs typically provide greater opportunity to trade bandwidth for quality, compared to audio codecs. Video streams are hence more suited for rate adaptation.

If the available bandwidth in a videoconferencing session gets below a certain level, the only viable approach in order to be able to continue the conference at all, is to simply drop the video and continue using audio only. This quite radical measure is triggered when the MOS calculated by the PSQA algorithm goes down to 1 (the lowest level, corresponding to more or less unusable quality). This signals to the sender of the media streams (or reflector) to stop sending video. When the estimated quality increases to the value 3 again, the video is re-enabled, in rate-controlled mode. When the MOS reaches 4, the rate adaptation is disabled after a configurable hold-down time.

3.3 Rate Adaptation

The rate adaptation mechanism is implemented both in the end system and in the RTP reflector. By having the adaptation mechanism in the reflector, the downstream rates from the reflector can be adapted to different bandwidth levels for heterogeneous network configurations. The upstream rate (to the reflector, or when no reflector is used) is controlled by the sender. The rate adaptation algorithm is basically the same in both cases: the frame rate of the video is adjusted to match a bandwidth limit which is determined by RTCP Receiver Reports (RR) of actual throughput, as measured by each receiver. At the reflector, this is done by intelligent frame dropping. At the sender, it is done by configuring the codec's target bitrate.

The components of the rate control mechanism is illustrated schematically in Figure 1 for a hypothetical scenario with one sender (S) and three heterogeneous receivers (R_i) interconnected by a reflector (R). (In a real scenario, all end systems would typically be both senders and receivers.) The rate control components are illustrated by circles in the figure, whereas the RTP stream monitoring components (measuring loss rate and throughput) are represented by triangles.

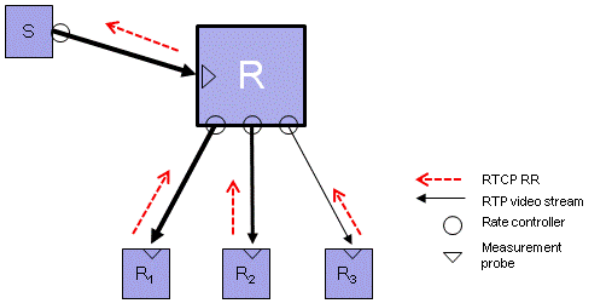

Figure 1: Schematical illustration of the rate control components implemented in the sender (S) and reflector (R). The downstream rates from the reflector are determined from the RTCP Receiver Reports from the receivers (R_i).

The advantage of adjusting the frame rate instead of some other video codec parameter, such as the quantization

level which is commonly used for codec rate adaptation, is that it can be done at the reflector without the need for transcoding. For the sender side rate controller, more elaborate codec parameter adjustments could be preferable. The main disadvantage of frame rate adaptation is that the burstiness of the resultant stream increases, which can cause problems in networks with low bandwidth links and small router buffers. In this case, a stream shaping mechanism (token bucket or similar) can be applied after the rate adaptation to smooth out the frames.

The RTP streams received by the end systems are monitored to measure throughput and packet loss rate. These performance metrics are reported back to the reflector in RTCP RR packets. The reflector monitors the reports, and as soon as packet loss is detected, the bandwidth limit of the corresponding RTP stream is adjusted to the actual throughput reported in the RTCP packet. To effectuate the rate control, the reflector selectively drops packets in order to keep the downstream rate below the bandwidth limit that the receiver reported when it first experienced loss. By selectively we mean that the reflector drops full frames, dropping P-frames before I-frames, instead of dropping random packets and the current bandwidth usage is re-evaluated on I-frame boundaries.

The reflector keeps the bandwidth limit until it receives a new RTCP RR packet with updated loss rate and throughput values. If there is still loss, the bandwidth is set to what is reported by the receiver, if there is no loss the bandwidth is kept at the current rate.

When there has been a lossless period for about 10 seconds following a loss event, the bandwidth limit is increased by 5 percent to probe for available bandwidth. After yet another period with no loss, the bandwidth is increased another 5 percent. Hence, the algorithm is driven by the receiver reports and typically the algorithm adapts rapidly to worsening conditions and rather slowly to improved conditions. We have so far seen that a too optimistic approach to increase the rate usually ends up with an overestimation of the available bandwidth, resulting in congestion and a poor performance with respect to perceived user experience.

The upstream adaptation algorithm, i.e. limiting the stream from the sender to the reflector, is more or less done in the same way, but with the reflector measuring throughput and loss rate and sending RTCP RR packets to the rate controller in the sender.

Given the rate control algorithm described above, there is of course a set of parameters that are of interest to experiment with. Firstly, how often should the receiver reports ideally be sent? At least for the case when the receiver detects nonzero packet loss it should be reported as quickly as possible. However, too often would be a waste of bandwidth, without improving responsiveness. Secondly, should the reflector use the reported effective throughput as the limit for the sending rate or should it possibly use a bandwidth slightly below the reported bandwidth? Thirdly, how often should the probing for available bandwidth take place and at what rate should it be increased? We are currently investigating these issues in order to optimize the performance.

Another open research issue is how the available bandwidth should be shared between multiple streams. This question is very interesting since there are many possible ways to allocate the available bandwidth to the streams. One of the more obvious solutions would be to use a bandwidth “fairness” algorithm where each stream's allocated bandwidth is in direct proportion to the original stream's bandwidth. Another approach would be to add information about who is currently speaking and allocate more bandwidth to that stream, since it could somehow be considered more important.

4 TESTBED CONFIGURATION

A testbed for experimentation and demonstration with the QoE-driven rate adaptation has been established, as depicted in Figure 2. In this set-up, one computer, labeled *Demo System 1* in the figure, sends real-time audio and video streams through an RTP reflector to three video receivers labeled *Demo System 2*, *Demo System 3* and *Demo System 4* respectively. A network emulator is positioned between the RTP reflector and two of the video receivers to simulate different network conditions, while one of the receivers (*Demo System 2*) is unaffected by the perturbations introduced. This gives the opportunity to study how the QoE optimization can be performed in the RTP reflector independently for a set of heterogeneous receivers, enabling different quality levels for the receivers based on their downstream bandwidths. The external monitoring system and QoE estimation visualization, shown in the top part of Figure 2, are used to verify and visualize the actual network conditions and the QoE as estimated based on the monitoring. The external monitoring systems tap into the network at any place of interest, typically before and after the network emulator, as indicated in the figure.

Service optimizations mechanisms implemented in the end systems (demo systems) and in the RTP reflector can be demonstrated by showing the performance with and without the mechanisms, for the emulated network conditions.

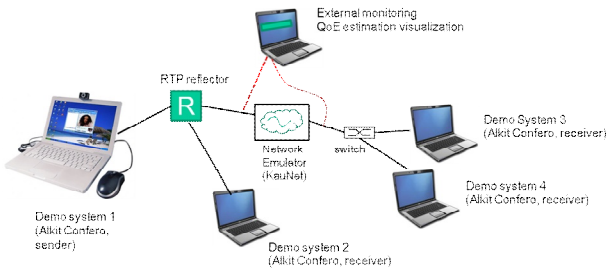

Figure 2: Testbed for QoE-driven service optimization

4.1 Network Emulation

The purpose of using a network emulator in the testbed is to create a variety of network conditions in a controlled manner. Our intention is to analyse the performance of our service optimization framework under controlled conditions. Once the framework is fully deployed, testing the system in a real network is going to be an important next step. However, the operating point of a network at any given time is a non-trivial combination of a number of different factors, making the overall behaviour non-deterministic. In order to fine-tune our video bandwidth adaptation algorithm, we need to observe its behaviour as we change essential network parameters like bandwidth, delay, and packet loss rate in a repeatable way. In other words, a network emulator provides us with the fully deterministic network we need for our tests.

In order to have repeatable network emulation in our testbed, we use KauNet 2.0 [7], a software tool developed at Karlstad University. KauNet actually extends another network emulation software, Dummynet [8], which is a standard tool on FreeBSD and Mac OS X. KauNet provides its users with many configuration possibilities, such as bandwidth, delay, packet loss, packet reordering, bit errors, or any combination of these. Pattern files are generated in advance to define the desired network behaviour on a per-packet or per-millisecond basis. KauNet processes these pattern files to emulate the network conditions. Using the pattern files, KauNet is even capable of emulating temporal changes in the network conditions, such as an increase or decrease in bandwidth or delay over time.

In our testbed, KauNet runs on Linux on a desktop PC with two network interfaces. In the network topology depicted in Figure 2, this node corresponds to a network link between the RTP reflector and the switch connected to two of the receivers in the demo system. This configuration enables KauNet to act as a transparent node between two nodes in the network and to introduce delay, loss, and change of bandwidth as defined by the user.

5 PERFORMANCE EVALUATION

The testbed described in section 4 has been used to verify how our novel QoE-driven audiovisual service optimization framework can improve the quality of video communication and conferencing sessions. For the results presented here, a configuration with a single sender and a single receiver of audio and video streams is chosen for simplicity. In the general case, multiple senders and receivers can be present.

In the performance plot shown in Figure 3, the MOS calculated by the receiving end system is plotted together with the packet loss rate for the audio stream. Figure 4 shows the bandwidth of the audio and video streams as measured by the same receiver. As can be seen in the figures, the media streams are initially received without loss, at their full transmitted rate (about 1.1 Mbps for video and 32 kbps for audio). The estimated audio quality is 4 (good quality). After about 3 seconds into the experiment, a bandwidth limitation of 500 kbps is introduced by the network emulator. This can be seen to drastically reduce the throughput of video (as expected), while the loss rate increases to about 5%, and in response the estimated audio quality drops to 3. This triggers the video rate adaptation in the reflector, which initially reduces the video bandwidth to around 200 kbps and then stabilizes around 400 kbps, leaving enough bandwidth for the audio stream to recover.

After about 10 seconds, the network emulator is reconfigured with a bandwidth restriction of 50 kbps. This can be clearly seen to increase the packet loss rate dramatically, to over 80% loss, since the rate adaptation mechanism is unable to reduce the video bandwidth enough, resulting in an estimated audio quality measure of 1. This triggers the modality change event, whereby the video stream is dropped altogether by the reflector, which makes the audio quality recover to the highest level attainable in practice (4), as the packet loss rate vanishes.

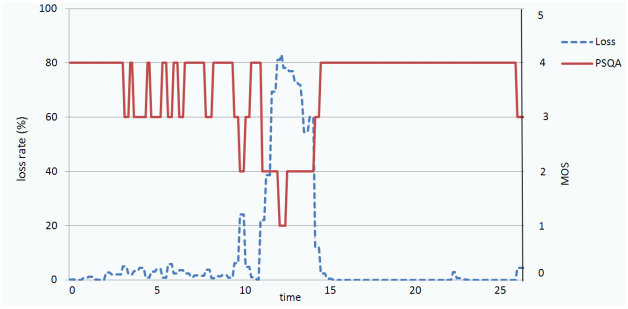

Figure 3: Packet loss rate and MOS

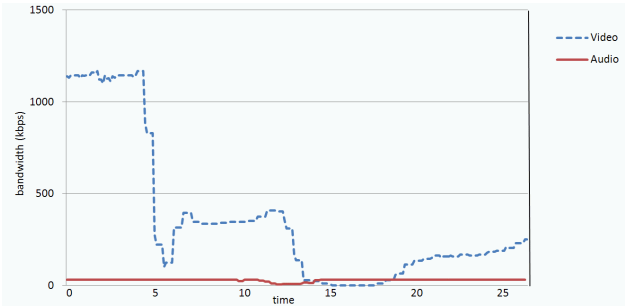

Figure 4: Audio and video bandwidth

Finally, after about 20 seconds, the bandwidth restriction is removed in the emulator, which in combination with the expiration of a 5 second hold-down timer causes the video to be re-enabled and the rate adaptation can be seen to start increasing the video bandwidth. The rate increase is done in small increments, to avoid driving the network to congestion when probing for available bandwidth. This will lead to a slow convergence to the optimal bandwidth level, when recovering from a low level (i.e. when going from bad network conditions to good). This part of the rate adaptation algorithm needs to be further developed with a more aggressive decision mechanism, although still

careful enough not to congest the network immediately and not cause oscillation between on-off states for video.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have presented a proof-of-concept implementation of QoE-driven audiovisual service optimization. The concept is based on conducting subjective tests with a video communication tool in a controlled network environment, where network perturbations are introduced and user response recorded to train the neural network. The subjective QoE can then be estimated in real time by the video communication service by feeding network monitoring data into the neural network, resulting in a MOS-like score quantifying the QoE.

The QoE estimation of our current prototype uses audio quality estimations to drive the video rate adaptation algorithm of the service and to trigger modality changes from audio/video to audio only. Our initial experiments show that the novel approach with QoE-driven real-time adaptation and quality optimization of audiovisual communication services is feasible in practice and can improve the subjective experience of future systems and services.

As future work, we are going to incorporate video quality estimations into the video rate adaptation system. We also intend to improve the adaptation algorithm according to performance evaluation results that we obtain from discrete event simulations where we can experiment further with various network scenarios complementing the testbed we designed.

Acknowledgment

This work is supported by IPNQSIS, a Celtic-Plus project. The work of the Swedish partners is partially funded by VINNOVA, the Swedish Governmental Agency for Innovation Systems. The work of the Finnish partner is partially funded by Tekes, the Finnish Funding Agency for Technology and Innovation.

References

[1] X. Wang, H. Schulzrinne, "Comparison of adaptive Internet multimedia applications," IEICE Transactions on Communication, Special issue on distributed processing for controlling tele-communications systems, vol. E82-B, no. 6, June 1999.

[2] J. C. Bolot, T. Turletti, "A rate control mechanism for packet video in the Internet," Proceedings of IEEE INFOCOM'94, June 1994.

[3] M. Johanson, "Supporting video-mediated communication over the Internet," PhD Thesis, Chalmers University of Technology, Department of Computer Engineering, ISBN 91-7291-282-0, May 2003.

[4] M. Varela, "Pseudo-Subjective Quality Assessment of Multimedia Streams and its Applications in Control," Ph.D. Thesis, INRIA/IRISA, univ. Rennes I, Rennes, France, Nov. 2005.

[5] K. Kilkki , "Quality of Experience in Communications Ecosystem ," Journal of Universal Computer Science, vol. 14, no. 5 (2008), pp. 615-624, 2008.

[6] P. Reichl, "From Quality-of-Service and Quality-of-Design to Quality-of- Experience: A Holistic View on Future Interactive Telecommunication Services," 15th International Conference on Software, Telecommunications and Computer Networks, Sept. 2007.

[7] J.Garcia, P. Hurtig, A. Brunstrom, "KauNet: A Versatile and Flexible Emulation System," Proceedings of SNCNW 2008 – the 5th Swedish National Computer Networking Workshop, Karlskrona, Sweden, April 2008.

[8] M. Carbone, L. Rizzo, "Dummynet Revisited," SIGCOMM CCR, Vol. 40, No. 2, April 2010.