Coupling QoE with dependability through models with failures

Ana Paula Couto da Silva
Inria/Irisa
Campus de Beaulieu, 35042
Rennes, France
Email: apaula@irisa.fr

Pablo Rodríguez-Bocca
InCo – Universidad de la República
Julio Herrera y Reissig 565, 11300
Montevideo, Uruguay
Email: prbocca@fing.edu.uy

Gerardo Rubino
Inria/Irisa
Campus de Beaulieu, 35042
Rennes, France
Email: rubino@irisa.fr

Abstract

Consider the problem of sending real-time video streams over the Internet, using a P2P network. The main difficulty is to deal with the high mobility of the peers entering and leaving the network (the latter can be seen as *failures*). This paper studies the impact of the peers' dynamics on the QoE (Quality of Experience) as seen by the clients. Instead of looking at standard metrics, like loss rates, delays, reliability, etc., we focus on the perceived video quality and we use the PSQA (Pseudo Subjective Quality Assesment) technology to evaluate it. The main contribution of this paper is to show how to couple PSQA with dependability aspects of the system in order to design a robust P2P architecture against the peers' disconnections.

I. Introduction

There is nowadays an increasing growth of multimedia systems present in the Internet. This is a consequence of the development of broadband accesses in residential users, together with the opening of content producers to new business models. These systems have many different architectures, depending, among other factors, on their sizes and on the popularity of their contents. The majority of them have a traditional Content Delivery Network (CDN) structure (for instance, YouTube [1]), while new proposals try to share the distribution of the multimedia with the servers through the present mature Peer to Peer (P2P) systems. In this work, we are concerned with the last case, and, as a part of a larger project, we will have a look at some aspects of the associated performability.

P2P are virtual networks developed at the application level over the Internet infrastructure. The nodes in the network, called peers, offer their resources (bandwidth, processing power, storing capacity) to the other nodes, basically because they all share common interests. The main reason explaining their success is that they allow to share the resources spread over the population of nodes in a scalable way. As the number of nodes increases, here aiming at receiving the stream, the global request for bandwidth also increases, but so does the global amount of available resources. This is a nice property. The price to pay is the high dynamics of the network, due to the fact that peers join and leave the system arbitrarily, and frequently.

Let us consider a P2P architecture to distribute live streaming, where a peer P_1 sends a stream to a peer P_2. If P_1 leaves the network, P_2 will not be able to play the stream until a new peer P_3, already receiving the stream, can replace P_1. In the P2P context, when P_1 leaves the network we say we have a *failure* event. Associated with it, a *repair* occurs when P_3 starts sending data to P_2.

We are interested here in the impact of these failures on the quality of the stream as seen by the clients. The traditional way of analyzing this is to study performance measures of the systems, or dependability ones. Here, we will follow a performability-like approach, considering at the same time the failures (as defined before) and the resulting performance. However, instead of looking at standard metrics which we qualify here of *indirect* (loss rates, delays, or reliability, availability...), we will address the *ultimate target*, the quality of the stream, as perceived by the end user.

Quantifying the perceived quality, by definition a *subjective* concept, can be done using a panel of real human observers, on a set of sequences, following some appropriate norm (see, for instance, [2], [3]). The corresponding area is called *subjective testing*. This provides a numerical value for the perceived quality, but it is very costly and can not be used in a modeling work, for instance for designing purposes. In this paper we will use the PSQA (Pseudo-Subjective Quality Assesment, [4], [5]) technology, which allows to build an accurate estimation of the value given by human observers to the streams.

The paper is organized as follows. Section II introduces the PSQA methodology, as well as the quality evaluation function. Section III describes the model we propose in order to study the way quality is perceived in the described system. Section IV gives some results that describe how the quality perceived by the end user is influenced by the P2P network dynamics. The main contributions of this work are then summarized in Section V.

II. Quality of Experience Measurement

As stated before, there are different ways to attack the problem of evaluating the perceived quality of a video flow, that is, of quantifying the quality as perceived by the end customers (sometimes called *Quality of Experience (QoE)*). The most accurate way is to use a panel of human observers, which following a specific norm (for instance, the ITU-R BT.500-11 [2]) and under controlled experimental conditions, provide a precise numerical quality value of the flows. The technical area is called *subjective testing*. These tests are expensive, time-consuming, and, by definition, they are not automatic.

The *Pseudo Subjective Quality Assessment (PSQA)* methodology provides a way to automatically building an accurate estimation of the value that can (could) be obtained from a subjective test, but automatically, and in real-time if useful (see [6], [7] for the original papers and [8] for a more recent and global presentation). PSQA consists of learning the way humans react to quality, by performing a set of subjective tests, and making a learning tool behave like them in a real networking environment. It operates by measuring the instantaneous value of specific metrics in the streams (for instance, the frame loss rate, or the effective bandwidth of the connection), and then, by building the evaluation of the QoE using the function defined during the learning phase. PSQA provides a value at time t for each t (in practice, it will be at every Δt), which can be used as the *instantaneous* perceived quality.

In this analysis we focus on two specific parameters concerning losses, because we know from previous work on PSQA [4] that the loss process is the most important network factor for video quality. We consider the loss rates of video frames, denoted by LR, and the mean size of loss bursts, $MLBS$, that is, the average length of a sequence of consecutive lost frames not contained in a longer such sequence. For the purpose of this work, it must be just said that the PSQA approach, once the learning (and validation) phases correctly done, provides a mathematical function $Q()$ of the chosen variables (here, LR and $MLBS$), mapping them into a MOS-like quality value (MOS stands for Mean Opinion Score). This means that we are approximating the perceived quality at t by $Q(LR(t), MLBS(t))$ where $LR(t)$ and $MLBS(t)$ are instantaneous values obtained by measuring. In a nutshell, $Q()$ is built by learning the way users behave from the quality point of view, face to the streams. The learning tool used is a specific class of Neural Network (a Random Neural Network) having a feedforward structure. A consequence of this is that the $Q()$ function has nice mathematical properties. In particular, it is a rational function of its input variables; the degrees of the numerator and the denominator in $Q()$ depend on the number of hidden layers and the number of neurons in each. We used a 3-layer architecture first, and also a 2-layer one, that is, a Neural Network without hidden neurons (just two input neurons and one neuron at the output layer). Using a 3-layer architecture, the validation phase led to a Mean Square Error (average squared difference between the values of $Q()$ and those given by real human observers)

$MSE_{\text{3-layer}} = 0.023$. For the 2-layer case, we got $MSE_{\text{2-layer}} = 0.041$. Such a good validation allows us to approximate the perceived quality with a very simple function:

$$Q(LR, MLBS) = \frac{a\, LR + b\, MLBS}{c\, LR + d\, MLBS + \nu}, \qquad (1)$$

where $a = 0.00142128$, $b = 0.47242076$, $c = 0.00142128$, $d = 0.47242076$ and $\nu = 0.01$. This is valid in the specific considered range of the input variables (that is, in the range corresponding to the values of those variables in the sequences that were used to learn from real human behaviors [8]). In Figure 1 we can see the obtained function. The interval where the chosen variables were considered are the ones indicated in the axis. Observe that quality is monotone in the two variables, and particularly increasing with the $MLBS$, meaning that, in this losses range, humans prefer sequences where losses are concentrated over those where losses are spread through the flow.

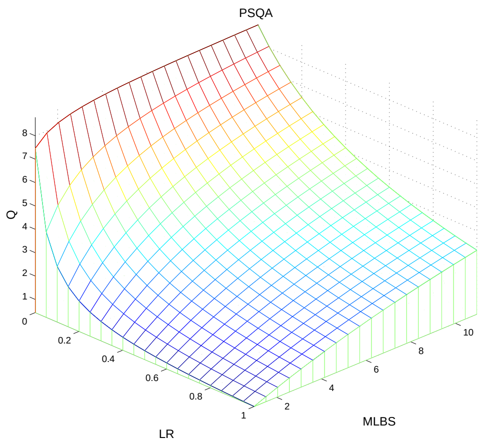

Fig. 1. The PSQA curve in our setting. Based on a two-layer RNN architecture.

III. Model

In this section, we will use a simple model in order to study the way quality would be perceived in such a system. Consider the following simplifying assumptions. The time to process a data unit (a frame) is exponentially distributed with parameter μ. The arrival rate of frames to the client, when the node sending them to it is working, is obviously also μ. The client starts with a buffer containing N frames. When everything goes correctly, he receives frames at the same rate he plays them, so (we assume) the buffer level will remain constant. The node sending the frames can fail (typically, it will leave the network), which happens with some failure rate ϕ (that is, we assume an exponentially distributed sojourn time of a peer in the network). The network reconfigurates itself at points in time distributed as a Poisson process with rate r. After a reconfiguration (an instantaneous action, in our model), the clients that had lost their stream because of a failure are connected again (to some peer/source). Let us call *cycle* the period between two consecutive network reconfigurations.

With the usual independence assumptions, we can build the following Markov process X describing the evolution of such a client, during a cycle. The state space is the set of non-negative integers. State 0 means that the node sending the frames to the observed client (from now, the client's *provider*) is active, and everything goes normally. The client's buffer level is N. If the provider fails, X moves to state 1 (with rate ϕ). State n, for $1 \leq n \leq N+1$ means that the provider is down (left the system) and that the client has already played $n-1$ frames in that situation; the number of frames remaining in the buffer is $N-n+1$. State $N+m$, $m \geq 1$ means that the provider is down and that the client has already lost $m-1$ frames (because their playtimes arrived and there was nothing to play). Figure 2 shows the transition diagram of process X. The loop at state 0 represents the fact that there is a regeneration of the whole network with rate r. Of course, if the client started with N frames in its buffer, after the next reconfiguration it will have a random number Y of frames, $Y \leq N$. We will look at that later. For the moment, we will use X to compute the loss rate at the client side. We call this a *performability* model because it takes into account failures, repairs, and services.

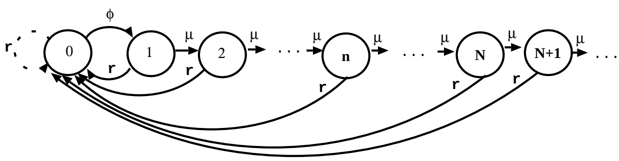

Fig. 2. Evolution of a client during a cycle

Remember that we focus on losses due to starvation, because in the case of a P2P system, these are by far the most important ones (compared with congestion losses). Let π_n denote the probability that X is at state n in steady state. Then, the loss rate in the cycle will be defined as the probability of being at states $N+1$, $N+2$, etc. (which is, due to the PASTA property, the probability that at the moment we must play a frame, there is no frame in the buffer). In formal terms,

$$LR = \sum_{n \geq N+1} \pi_n.$$

Process X is always stable, and we have

$$\pi_0 = \frac{1}{1+\varepsilon}, \quad \varepsilon = \frac{\phi}{r},$$
$$\pi_1 = \frac{\pi_0}{1+\delta}, \quad \delta = \frac{r}{\mu}$$

and for any $n \geq 2$,

$$\pi_n = \frac{\varepsilon}{1+\varepsilon}\frac{\delta}{(1+\delta)^n}.$$

After some algebra, this leads to

$$LR = \frac{\varepsilon}{1+\varepsilon}\frac{1}{(1+\delta)^N}.$$

The computation of the $MLBS$ is easier. First, we are now counting, so, we work on the canonical discrete time Markov chain embedded into X. In Figure 3 we see its transition probabilities.

The parameters of this model are

$$u = \frac{\varepsilon}{1+\varepsilon}, \;\; v = 1-u, \;\; p = \frac{1}{1+\delta}, \;\; q = 1-p.$$

The probability that a burst of losses has size j being $p^{j-1}q$, we have:

$$MLBS = \frac{1}{q} = 1 + \frac{\mu}{r}.$$

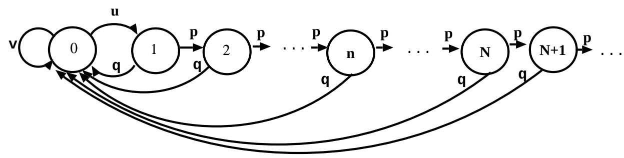

Fig. 3. The canonical discrete time Markov chain embedded in X.

Observe that if the cycle starts with N frames in the buffer, at the beginning of next cycle the buffer can contain N, $N-1$, ..., frames. We denoted by Y the number of frames present in the buffer at the beginning of next cycle. We have

$$\Pr(Y = N) = v + uq,$$

then, for $j = N-1, N-2, \cdots, 1$,

$$\Pr(Y = j) = up^{N-j}q,$$

and, finally,

$$\Pr(Y = 0) = up^{N}.$$

IV. Tests and First Results

In this section we present the first results about the impact of the peers' dynamics on the quality perceived by the end user. The main objective is to analyze how different sojourn times of a peer in the network, as well as different reconfiguration strategies, affect the quality of the video played by a client.

We consider here two strategies of reconfiguration. In the first case, every $1/r$ secs, on the average, the network is reconfigurated. By means on message exchanges, the system discovers which nodes left and, if they were serving other peers, they replace them by other nodes to allow the former to continue receiving the signal. In the second case, we consider using a supplementary server peer that delivers the information to the client having lost its source until the network can be reconfigurated.

Let us analyze the first scenario. We will consider two cases here: when a cycle begins the client has either 0 or 100 frames in the buffer. For the failure rate, we consider a broad range: $\phi = 0.1, 0.01, 0.001$. Figure 4 shows the values for the loss rate metric in these scenarios as a function of the reconfiguration rate r, in a range from 1 sec to 20 sec (larger reconfiguration intervals decrease significantly the network performance). As expected, if the server spends more time in the network, the client observes less losses. A similar correlation can be observed with respect to the initial number of frames in the buffer: the larger this number, the smaller the loss rate.

Now, let us consider the $MLBS$ measure. For the set of values of ϕ and r, the $MLBS$ reaches high values (600 to 30 frames). In these cases, we are out of the input domain of $Q()$. What simply happens is that such high values of $MLBS$ mean that quality will be very low (only 30 frames correspond to a second of signal). Moreover, observe that those very long bursts are rare. The probability of observing at least m consecutive losses is up^{N+m} in the case of a buffer having at the beginning of the cycle N frames, that is, $\varepsilon(1+\varepsilon)^{-1}(1+\delta)^{-N-m}$. This means that a very few number of peers will observe those bursts, leading to a poor quality. In other words, the situation where $MLBS = 600$ corresponds to a case of very rare losses, but when happening, arriving in long bursts.

In any case, a reconfiguration rate $r = 0.1$ means a mean reconfiguration period of 10 sec, and $r = 1$ means a reconfiguration every sec, on the average. This can be too low. In order to improve

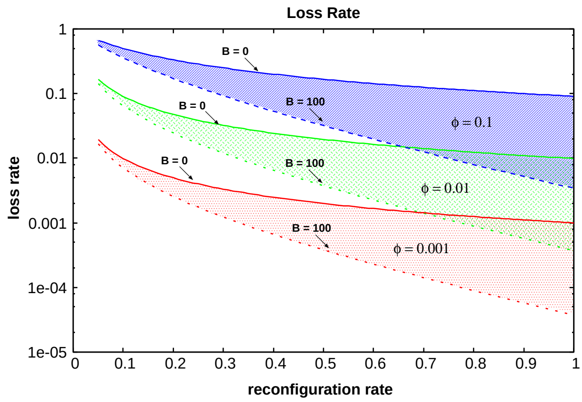

Fig. 4. Loss rate measure for the first strategy of reconfiguration.

the performance of the system, let us consider the use of a special server peer that sends the stream to nodes having lost their providers, before the next reconfiguration point arrives. Let us consider that the mean time to such a special server starts transmitting information is 250 msec. In this case, as expected, the loss rate and the $MLBS$ are lower than in the previous one. Again we consider two different initial buffer sizes in the cycle: either 0 or 100 frames, and the same range for the failure rate ϕ. The largest loss probability is equal to 0.025 and the $MLBS$ varies between 7 and 8.4, as shown in Figure 5. The LR and $MLBS$ values were mapped into a perceived quality level using the $Q()$ function. The resulting quality Q had a small variation in this context: the maximum quality is equal to 0.87 and the minimum quality is equal to 0.82. This means that in the considered situation, quality is good enough.

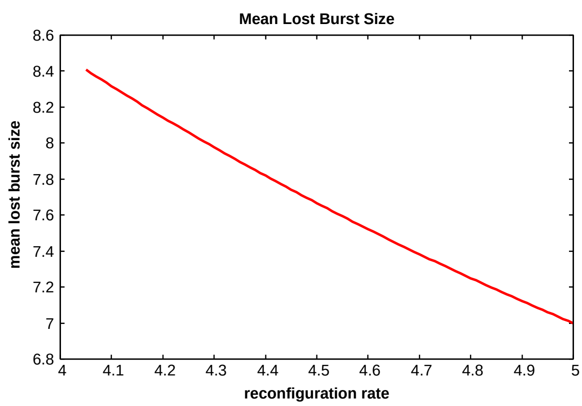

Fig. 5. MLBS measure for the second strategy of reconfiguration.

A. *Assuring Video Quality*

Using our model and the simple $Q()$ function introduced before, we can provide an analytical expression of the perceived quality as a function of the system's parameters ϕ, r, N and μ. Just replace LR and $MLBS$ in (1) by the corresponding expressions provided by the model.

This can allow to easily obtain rough answers to interesting questions. For instance, which is the minimal buffer size necessary to assure a given quality level? That is, what is the minimal value of N such that $Q(\phi, r, N, \mu) \geq Q_0$? After some algebra, this leads to:

$$N \geq \frac{\log \dfrac{cQ_0 - a}{\left(1 + \dfrac{r}{\phi}\right)\left[(bQ_0 - d)\left(1 + \dfrac{\mu}{r}\right) - Q_0\nu\right]}}{\log\left(1 + \dfrac{r}{\mu}\right)}.$$

For $Q_0 = 0.87$, it is possible to see that the necessary buffer size is very small, with a maximum of 28 frames (1 second) when we have $\phi = 0.1$ and the highest reconfiguration rate $r = 5$.

V. Conclusion

In this paper, we present a new approach to evaluating P2P network QoE for multimedia streaming. We do this by integrating a simple Markov performability model with a pseudo-subjective quality assessment (PSQA) approach. The proposed model allows us to describe the evolution of a peer client during a *cycle*, i.e, the period between two consecutive network reconfigurations. Using PSQA we are able to work with the perceived quality as a function of the design parameters and thus to efficiently perform a quantitative analysis of our system.

We consider two network design parameters in our analysis: the reconfiguration rate r of the network and the number N of frames in the buffer just after a reconfiguration. The coupling between PSQA and the performability model allows to evaluate the impact of r and N on the QoE. Current work is being done now to extend these results to other parameters and to use transient views of the system.

References

[1] youTube website, http://www.youtube.com/, 2007.

[2] ITU-R Recommendation BT.500-11, "Methodology for the subjective assessment of the quality of television pictures," Jun. 2002.

[3] ITU-R Document 6Q/57-E, "Draft new recommendation for subjective assessment of streaming multimedia images by non-expert viewers," Apr. 2004. [Online]. Available: http://www.itu.int/

[4] S. Mohamed and G. Rubino, "A Study of Real–time Packet Video Quality Using Random Neural Networks," *IEEE Transactions On Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1071 –1083, Dec. 2002.

[5] G. Rubino and M. Varela, "A New Approach for the Prediction of End-to-End Performance of Multimedia Streams," in *Proceedings of the First International Conference on Quantitative Evaluation of Systems (QEST'04)*, Enschede, The Netherlands, Sep. 2004.

[6] S. Mohamed, G. Rubino, H. Afifi, and F. Cervantes, "Real–time Video Quality Assessment in Packet Networks: A Neural Network Model," in *Proceedings of International Conference on Parallel and Distibuted Processing Techniques and Applications (PDPTA'01)*, Las Vegas, Nevada, USA, Jun. 2001.

[7] S. Mohamed, F. Cervantes, and H. Afifi, "Audio Quality Assessment in Packet Networks: an Inter–Subjective Neural Network Model," in *Proc. of IEEE International Conference on Information Networking (ICOIN-15)*, Beppu City, Oita, Japan, Jan. 2001, pp. 579 –586.

[8] G. Rubino, "Quantifying the Quality of Audio and Video Transmissions over the Internet: the PSQA Approach," in *Design and Operations of Communication Networks: A Review of Wired and Wireless Modelling and Management Challenges*, ser. Edited by J. Barria. Imperial College Press, 2005.