Modeling Quality of Experience in Multi-source Video Streaming

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

Héctor Cancela
InCo – Universidad de la República
Julio Herrera y Reissig 565, 11300
Montevideo, Uruguay
Email: cancela@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. In this paper, we present some results exploring a multi-source approach where the flow is decomposed into different substreams transporting the original sequence plus some redundancy. Using a recently proposed technique called PSQA which allows to automatically and accurately approximate the value of the quality at the terminals, as perceived by the end user, we study the design of such a multi-source transmission scheme, with the goal of optimizing the quality as seen by the final users. We explore the problem using simple models of the transmission scheme and of the peers dynamics.**

I. Introduction

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. As a consequence, as the number of customers increases, the same happens with the global resources of the P2P network. P2P networks are becoming more and more popular today and are very used for file sharing and distribution; some known examples are Bittorrent, KaZaA, eMule, etc. The problem is that peers connect and disconnect with high frequencies, in an autonomous and completely asynchronous way. This means that the resources of the network as a whole are also highly dynamic, and thus, that the network must be robust face to these fluctuations.

In this paper, we are interested in some aspects related to the use of a P2P architecture to distribute live video. The main problem is how to provide good quality levels in a context where this quality depends on other clients that are delivering the stream, and given the fact that users connect and disconnect very frequently. This can be addressed by using some redundancy in the signals. Multi-source streaming is one of these techniques, where the live video stream is received by the client from flows sent by many sources simultaneously. This approach allows flexibility modulated by the dynamics of the network. In particular, it is possible to increase the number of sources and/or the amount of redundant information. This flexibility must be carefully tuned in order to get a satisfactory quality level with a minimal cost. The usual approach here is to use a well chosen metric, that we know plays an important role in quality, such as the loss rate of packets, or of frames. In this paper we instead address the problem of measuring *perceived* quality by means of the PSQA technology [1], [2]. PSQA is a general procedure that allows the automatic measure of the perceived quality, accurately and in real-time. We extend the technique to the case of multi-source streaming for live video, and improve its efficiency for video analysis by studying the flows at the frame level, instead of the packet level previously considered in the literature.

In order to face the high dynamics of such a system, we explore a multi-path approach where (i) the stream is decomposed in some way into several flows, (ii) each client receives several flows following different paths and sent from different other clients, (iii) the client is able to reconstruct the stream from the whole set of received flows and possibly from part of them; moreover, (iv) the system measures automatically the perceived quality at the client continuously, and takes its decisions (basically, periodically rebuilds the architecture of the network) using these values. The paper focuses then on the analysis of the impact on the perceived quality, as captured by the PSQA metric, of the fact that the stream is received from several nodes decomposed into different flows (explaining the term *multi-sourcing*). Our main goal is the description of a global methodology that can be used to design such a P2P distribution algorithm.

The paper is organized as follows. Different video quality measurements are presented in Section II, and in particular PSQA. Section III explain multi-source streaming techniques and develops some models needed for the construction of the PSQA measuring module. Section IV models the P2P environment with server nodes fails. In Section V our first experimental results are introduced. The main contributions of this work are then summarized in Section VI.

II. Quality of Experience Measurements

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 [3]) 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. Some variants included in the standards are DSIS, DSCQS, SS, SSCQE, SCACJ and SDSCE.

There have been attempts to do this quality assessment automatically. The technical area is called *objective testing*. Most of these techniques consist of comparing the original and the received sequences, which precludes using them in real-time, a needed feature for our purpose of designing P2P transport architectures. But the main drawback of objective tests, when using them to evaluate perceived quality (or QoE) is that they usually (that is, very often) correlate poorly with the values coming from subjective assessments.

The *Pseudo Subjective Quality Assessment (PSQA)* [1] is a technique allowing to approximate the value obtained from a subjective test but automatically, and in real-time if useful. 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. PSQA 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 building the approximation of the QoE using the function defined during the learning phase. The statistical learning tool used is the Random Neural Networks one [4], used to map the chosen parameters into quality. Observe that 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 our work, we focused on two specific parameters concerning losses, because we know from previous work on PSQA that the loss process is the most important global factor for 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. It is important to observe that in previous work using the PSQA technology the analysis was done at the packet level. Here, we are looking at a finer scale, the frame one, because quality is more directly influenced by loss frames than by loss packets. Packet-level parameters are easier to handle (in the testbed and from the measuring point of view in the network), but frame-level ones provide a more accurate view of perceived quality.

III. Multi-Source Streaming Models

Our P2P architecture consist of a main server producing a live video stream, splitting its stream into several flows with some amount of redundancy in them, and sending these flows to a specific set of clients. Each client sends the received flows to other nodes. The architecture must ensure that each client receives the different flows from different nodes, so from the client's point of view, we have a multi-source delivering system. The main goal is to better resist to the losses due to the frequent disconnection of nodes.

A first important aspect of our scheme is the degree of redundancy being employed; in the case of multiple servers, the extreme cases being to apply no redundancy at all, or to completely replicate all the information. In the first case, we have a "split" policy: each server can send a fraction of the streaming information, without any redundancy, and the loss of information at any of these flows will imply also losses at the client. In the second case, the policy being applied is "copy": each of the server nodes sends the full streaming to the client, which will then be less prone to quality problems caused by frames lost by communication problems or server disconnections. We will consider a third case, called "redundant-split" working as in the first case but adding some redundancy to the streams.

A. Sending K copies of the stream

Assume K copies of the same stream travel following independent and stochastically equivalent paths to the same terminal. The loss process at any of the K streams is represented by the server failure model described in next section. It is clear that the receiver will observe the loss of a frame only if all the K copies of the frames are lost. Let $LR^{copy}_{K,i}$ denote this global Loss Rate with K servers multi-source, and i connected among them, and $MLBS^{copy}_{K,i}$ the corresponding Mean Loss Burst Size. If there is no server connected, then everything is lost: $LR^{copy}_{K,0} = 1$ and $MLBS^{copy}_{K,0} = \infty$. If $1 \le i \le K$, then $LR^{copy}_{K,i} = 0$ and $MLBS^{copy}_{K,i}$ is undefined (or set to some arbitrary value).

B. Simple split of the stream into $K \ge 2$ substreams

In the other extreme case considered in this paper, we have K substreams transporting each a frame over K in the following way: frame 1 goes through substream 1, frame 2 through substream 2, until frame K going through substream K; then frame $K+1$ is sent through substream 1, frame $K+2$ through substream 2, etc. In general, frame n is sent by substream $((n-1) \mod K)+1$.

Assuming independence in server failures again, the global Loss Rate of this scheme is obviously proportional to the number of faulty servers; when i of them are still connected, we have

$$LR^{split}_{K,i} = \frac{K-i}{K}, \quad K \ge i \ge 0,\ K \ge 1.$$

The evaluation of the Mean Loss Burst Size is much more involved than the previous one, but since our goal is to guarantee some quality level, we only use trivial lower bound and an upper bound, by observing that, by definition,

$$1 \le MLBS^{split}_{K,i} \le K-i, \quad K \ge i \ge 0,\ K \ge 1.$$

C. Split of the stream into $K \ge 2$ substreams, adding complete redundancy

Between these two extreme policies (copy and split cases), we can for example split the stream in K sub-streams adding some redundancy to each in order to diminish the effect of losses at least when only one server disconnects (fails). If the original stream needs some bandwidth B Kbps, then we assume that each substream will use B/K Kbps plus some bandwidth needed to transport redundant data. Substream j is completely sent by server j, and its content is also sent by the remaining $K-1$ servers, each of them sending exactly a $(K-1)$th of it. Let us look now at the losses when there are only i active servers, among the initial connected K. In this case, without any redundancy we will loose a fraction $(K-i)/K$ of the stream. But with the adopted redundancy scheme, this is diminished by the fraction of this information that is transported, as redundant data, by the remaining connected servers. We have:

$$LR_{K,i}^{split-red} = \frac{K-i}{K} - \frac{(K-i)i}{K(K-1)} = \frac{(K-i)(K-1-i)}{K(K-1)}.$$

For the evaluation of the Mean Loss Burst Size we can use the same trivial lower and uppper bounds than in the "split" case.

IV. P2P Network and Model

In this section we describe a simple Markovian model used to represent the server connection/disconnection process in a multi-source streaming context. We adopt the following simplifying assumptions. The connection-time of any node acting as a server (that is, most of the nodes in the network) is exponentially distributed with some parameter λ. That is, $1/\lambda$ is the expected time a customer remains connected. Thus, it can be estimated from network statistics (strictly speaking, we refer here to the servers' connection time, which means that, to estimate λ, we must sample on the population of clients acting as servers; this usually happen after a minimal amount of connection time). Since we further assume that the servers leave the network independently of each other's behavior, the number of connected servers sending the stream to a fixed but arbitrary customer, at time t, considering that the network was re-built at time 0 and that no other re-building process is done in $[0,t]$, is a Markov process with states $K, K-1, \ldots, 1, 0$. The corresponding transition graph is shown in Figure 1. Since the failures of the components are assumed to behave

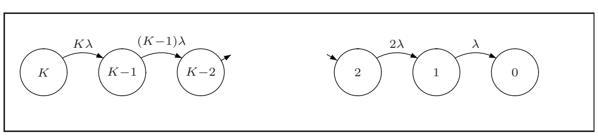

Fig. 1. The Markovian model used to represent the evolution of the number of connected servers sending the stream to the same (arbitrary) client.

independently, the probability that any of them is operating at time t is $e^{-\lambda t}$, and thus, the number of active servers at time t is Binomial with parameters K and $e^{-\lambda t}$. In other words, if $p_{K,i}(t)$ is the probability that i servers among the initial K are still operating at time t, then we have

$$p_{K,i}(t) = \binom{K}{i} e^{-i\lambda t}(1-e^{-\lambda t})^{K-i}, \quad K \geq i \geq 0,\ \lambda, t \in \Re.$$

To decide which client will serve another one, some degree of intelligence and knowledge about the peers and the network state is needed. That is, considering a client receiving the stream from K independent servers, when one of these servers leaves the network, whatever the assignment algorithm used (for instance: decentralized or centralized), it will need some time to operate, time denoted in the sequel by T (a convergence time). Specifically, we used $1/\lambda = 900$ sec. and $T = 10$ sec. To compute the value of λ, we employed logs of user behavior (specifically connection times) from the live-video service offered by a medium-sized ISP, which gave us access to this information.

V. Testing and First Results

In this section we study how the frame loss rate and frame mean loss burst size parameters affect the quality (as measured by the PSQA technique) for the three (copy, split and redundant split) streaming policies.

A. QoE Estimation in a Frame Losses Context

The first step was to apply the PSQA technique, as explained in Section II. This involved choosing four MPEG2 video sequences, of about 10 seconds duration each, with sizes between 1.5 MB and 2.8 MB). For each sequence, we generated 25 different evaluation points, where each evaluation point is defined by a loss rate value chosen at random with an uniform distribution between 0.0 and 0.2, and a mean loss burst size value chosen at random with an uniform distribution between 0.0 and 10.0. For each of the evaluation points, we used a simplified Gilbert model[1] [1] to simulate a frame drop history which was applied to the original video sequences; in this way, we obtained 100 modified video sequences with variable quality levels. A group of five experts evaluated the sequences and the MOS for each of the copies was computed, following the ITU-R BT.500-11 [3] norm. These MOS were scaled into a quality metric in the range $[0,1]$. Finally, we employed the MOS value for each of the design points as inputs in order to calibrate a Random Neural Network (RNN). After trained and validated, the RNN provides a function of two variables, LR and $MLBS$, mapping them into perceived quality (on a $[0,1]$ range). In Figure 2 we can see the obtained function. For ease of reading, we extrapolated the curve to the borders, but observe that the data are accurate and used on an internal region ($[1\%, 15\%]$ for LR, and $[1, 4]$ for the $MLBS$). We can see that quality is monotone in the two variables, and in particular increasing with the $MLBS$, meaning that humans prefer sequences where losses are concentrated over those where losses are more isolated. We have no room here to describe in detail the learning procedure; see the references given before for similar processes.

B. QoE Evaluation in our Multi-Source Streaming Techniques

Having a function mapping the two chosen parameters (frame loss rate and mean frame loss burst size) into perceived

[1]The simplified Gilbert model consists of a 2-state Markov process that controls which frames are lost in the flow (so, with 2 parameters; the original Gilbert model has 3 parameters [5]).

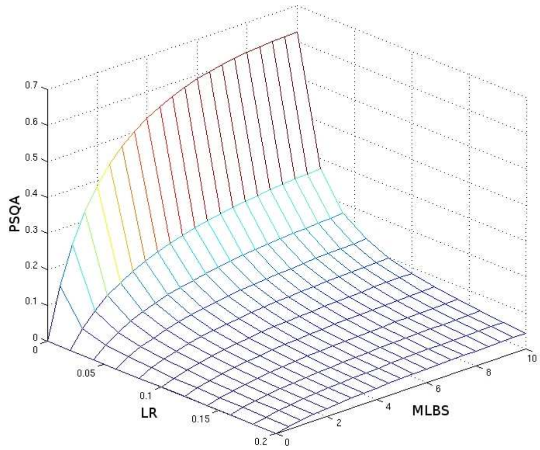

Fig. 2. The PSQA curve in our setting

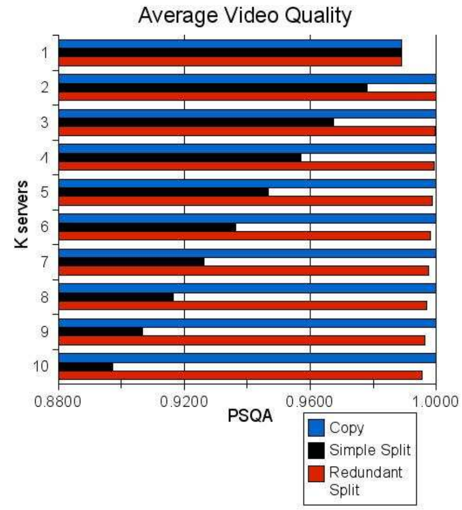

Fig. 3. Average Video Quality. In the "split" and "redundant split" cases, the data was computed using the lower bound for the perceived quality, the difference with the upper bound is completely negligible.

quality, it only remains to evaluate it on the values generated by the servers' disconnections in each streaming algorithm discussed in Section III. In the "split" and "redundant split" cases we have a lower quality bound (based in lower bound $MLBS_{K,i}^{split} = 1$) and an upper quality bound (based in upper bound $MLBS_{K,i}^{split} = K - i$); the data was computed using the lower bound for the perceived quality because the difference with the upper bound is completely negligible.

Knowing the subjective quality associated with every state of the network (i.e, with any combination of working and failed servers), makes it possible to answer different interesting and relevant questions. For example, we can observe that the worst quality level must occur just before a network configuration, that is, at time T if we consider that the last re-configuration happened at time 0. The mean quality (considering the whole population of clients) is, with our assumptions, given by

$$\mathrm{E}(Q_K) = \sum_{i=1}^{N} Q_{K,i}(LR_{K,i}, MLBS_{K,i}) p_{K,i}(T).$$

Figure 3 compare the average video quality for the three policies. It is possible to see that, in the case where there is no redundancy ("simple split"), the subjective quality degenerates rapidly with the growth of servers K. Also it is possible to compare the "copy" and "redundant split" cases, where for the frequency of disconnection of our real scenario it does not seem to be much gain in sending K copies of the streaming ("copy"), as sending a single copy ("redundant split") only loses a little percentage of the quality (remember that the "copy" method consumes KB Kbps of bandwidth and the "redundant split" method just $2B$ Kbps).

VI. Conclusion

This paper proposes some general principles for the design of a live-video P2P distribution system following a multi-source procedure where the video stream is decomposed into different flows that travel independently through the network. We use the PSQA technique that allows automatically measuring the perceived quality as seen by the final users. The paper focuses on the impact on quality (as measured by PSQA) on three extreme cases: sending the stream split into several disjoint substreams, that is, with no redundancy, sending many complete copies of the stream through different paths, and splitting again the stream into several substreams but adding some redundancy to the original signal.

The main conclusions are the following: (i) thanks to an improved version of PSQA we see that quality increases as losses concentrate in the stream (for a fixed loss rate); (ii) sending many copies of the signal is obviously the best option from the quality point of view, but the split approach with some redundancy provides very good results.

References

[1] 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.

[2] S. Mohamed, G. Rubino, and M. Varela, "A method for quantitative evaluation of audio quality over packet networks and its comparison with existing techniques," in *Proceedings of the Measurement of Speech and Audio Quality in Networks workshop (MESAQIN'04)*, Prague, Czech Republic, Jun. 2004. [Online]. Available: http://www.irisa.fr/armor/lesmembres/Varela/research/pdf/varela-mesaqin%04-1.pdf

[3] ITU-R Recommendation BT.500-11, "Methodology for the subjective assessment of the quality of television pictures," Jun. 2002. [Online]. Available: http://www.itu.int/

[4] E. Gelenbe, "Random Neural Networks with Negative and Positive Signals and Product Form Solution," *Neural Computation*, vol. 1, no. 4, pp. 502–511, 1989.

[5] E. Gilbert, "Capacity of a burst–loss channel," *Bell Systems Technical Journal*, vol. 5, no. 39, Sep. 1960.