EXPLORING QUALITY ISSUES IN MULTI-SOURCE VIDEO STREAMING

Pablo Rodríguez-Bocca

IRISA/INRIA,
Campus de Beaulieu, 35042 Rennes, France

ABSTRACT

This paper explores a key aspect of the problem of sending real-time video over the Internet using a P2P architecture. The main difficulty with such a system is the high dynamic in the nodes connections of the P2P network. We consider a multi-source approach where the stream is decomposed into several flows sent by different peers to each client. Using the recently proposed PSQA technology for evaluating automatically and accurately the perceived quality, the paper focuses on the consequences of the way the stream is decomposed on the resulting quality. We provide a global methodology that can be used to design such a system, illustrated by looking at three extreme cases. We also propose an improved version of PSQA obtained considering the video sequences at the frame-level, instead of the packet-level one of previous works.

1. 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 on their sizes and on the popularity of their contents. The majority of them have a traditional Content Delivery Network (CDN) structure (for instance, the case of msnTV, YouTube, Jumptv, etc.), while new proposals try to share the distribution of the video with the servers through the present mature Peer to Peer (P2P) systems (for instance, the case of Joost, iMP, etc.).

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 technics, 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 2, and in particular PSQA. Section 3 explain multi-source streaming techniques and develops some models needed for the construction of the PSQA measuring module. In Section 4 our first experimental results are introduced. The main contributions of this work are then summarized in Section 5.

2. QUALITY MEASUREMENTS

This section discusses different ways of dealing with the perceived quality in a video delivering system.

Perceived video quality is, by definition, a subjective concept. The mechanism used for assessing it is called *subjective testing*. It consists of building a panel with real human subjects, which will evaluate a series of short video sequences according to their own personal notion of quality. The output of these tests is typically given as a Mean Opinion Score (MOS). Obviously, these tests are very time-consuming and expensive in manpower, which makes them hard to repeat often. And, of course, they cannot be a part of an automatic process (for example, for analyzing a live streaming in real time, for controlling purposes). There exist standard methods for conducting subjective video quality evaluations, such as the ITU-R BT.500-11 [3]. Some variants included in the standard are DSIS, DSCQS, SS, SSCQE, SCACJ and SDSCE.

Other solutions, called *objective tests*, have been proposed. Objective tests are algorithms and formulas that measure, in a certain way, the quality of a stream. The most commonly used objective measures for video are PSNR, ITS' VQM, EPFL's MPQM, CMPQM, and NVFM. With some exceptions, the objetive metrics propose different ways of comparing the received sample with the original one, so, it is not possible to use them in an real-time passive test enviroment. Besides, these quality metrics often provide assessments that do not correlate well with human perception.

The *Pseudo Subjective Quality Assessment (PSQA)* [1] is a technique allowing to approximate the value obtained from a subjective test but automatically. To apply PSQA, we start by choosing the parameters we think will have an impact on quality. This depends on the application considered, the type of network, etc. Then, we must build a testbed with several distorted samples (by changing the values of the different parameter values), and evaluate this distorted samples by a panel of human observers (i.e. a subjective test). Finally, PSQA uses the results of this evaluation to train a specific learning tool (in PSQA the best results come from the Random Neural Networks one [4]) in order to capture the relation between the parameters that cause the distortion and the perceived quality. The output of this learning process is then a function able to build a quality value from the values of the parameters.

After training, using PSQA is very easy: we need to evaluate the values of the chosen parameters at time t, and then we use the obtained function which gives the *instantaneous* perceived quality at t.

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.

3. MULTI-SOURCE STREAMING MODELS

The main architecture considered involves some 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.

An important aspect is the degree of redundancy being employed; in the case of multiple servers, the extreme cases are 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.

Although in this work we concentrate on these extreme policies (either zero redundancy or full replication of the information sent by each server), it is clear that the degree of redundancy admits many other possibilities in-between.

In this section we develop stochastic (Markovian) models for the frame loss process in single source and multi-source (split and copy cases) streaming. We do not differentiate among losses due to the server node itself and losses due to the underlying Internet connection between server and client.

3.1. The simplified Gilbert model

To model the loss process on a end-to-end communication we build a discrete time stochastic process $(X_1, X_2, \ldots)$ where $X_n = 1$ if the nth frame is correctly transmitted, 0 if it is lost. The simplest i.i.d. case (a Bernoulli process) is obviously too simple because in general losses are correlated. To keep the model as simple as possible we used the so-called simplified Gilbert model. It consists of using a 2-state Markov chain for controlling which frames are lost in the flow (so, with 2 parameters; the original Gilbert model has 3 parameters [5]). Let us denote by 1 and 0 the states, with the following semantics: after a correct transmission, we will always be at state 1, and after a loss, at state 0. The two parameters are then $p = \Pr(\text{a loss after a correct transmission})$ and $q = \Pr(\text{a correct transmission after a loss})$.

The steady-state distribution of this model is given by $\pi_1 = q(p+q)^{-1}$, $\pi_0 = p(p+q)^{-1}$. The distribution of the length S of a generic burst of losses, considering the system in equilibrium, is geometric: $\Pr(S = n) = (1-q)^{n-1}q$, $n \geq 1$, with mean $\mathrm{E}(S) = q^{-1}$.

The Loss Rate LR of the flow, according to this model, and the Mean Loss Burst Size $MLBS$ of the stream, are

$$LR = \frac{p}{p+q}, \quad MLBS = \mathrm{E}(S) = \frac{1}{q}.$$

3.2. 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 model previously described, with parameters p and q. It is clear that the receiver will observe the loss of a frame if all the copies of the frames are lost. If LR_K^{copy} denotes this global Loss Rate, we then have

$$LR_K^{copy} = LR^K = \left(\frac{p}{p+q}\right)^K.$$

If S_K denotes the size of a generic burst of losses, we have $\Pr(S_K = n) = \left[(1-q)^K\right]^{n-1}\left[1-(1-q)^K\right]$, giving a global Mean Loss Burst Size $MLBS_K^{copy} = \mathrm{E}(S_K)$ as follows:

$$MLBS_K^{copy} = \frac{1}{1-(1-q)^K} = [1-(1-MLBS^{-1})^K]^{-1}.$$

3.3. Complete split of the stream into $K \geq 2$ 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 $K+1$ through substream 1, etc. In general, frame n is sent by substream $((n-1) \mod K) + 1$.

We obviously have here, for the Loss Rate of this scheme, the same value as for the single source case: $LR_K^{split} = LR = p(p+q)^{-1}$. The evaluation of the Mean Loss Burst Size is more involved. After some algebra, we get

$$MLBS_K^{split} = 1 + \frac{p}{q},$$

4. 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 single server and multiple server (copy and split) streaming policies.

The first step was to apply the PSQA technique, as explained in Section 2. 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 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 1 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 vari-

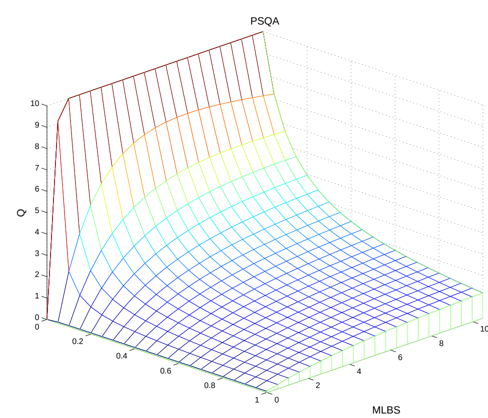

Fig. 1. The PSQA curve in our setting

ables, 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.

With this information, we can now compare the effect of the different streaming policies on the quality perceived at the client node. To compare we use the same frame loss process going on at the server side. In the three cases, the perceived quality deteriorates quickly with increasing values of LR; but the quality values are very different for every policy. In order to make it easier to compare the different policies, we show in Figures 2 and 3 the quality values for the three policies as a function of LR, with $MLBS = 1.0$ and $MLBS = 4.0$. These figures shows very clearly how the copy policy with two servers, which transmits every frame twice, profits from this redundancy and copes much better with quite high values of LR, maintaining better quality results, for both low and high $MLBS$ scenarios. In the case where $MLBS = 1$, the single and split policy give the same results (this results directly from our analytical models). When $MLBS = 4$, the split policy fares worse than the single policy. This results from the facts that in this case, the split policy results in a lower value for the $MLBS$ perceived by the user in relation to the $MLBS$ at the server side, and that the MOS computed from the experts opinions show that lower $MLBS$ result in higher quality values.

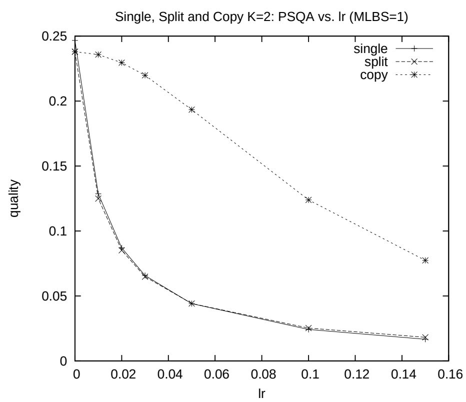

Fig. 2. Multi-source streamings without bursts, MLBS=1

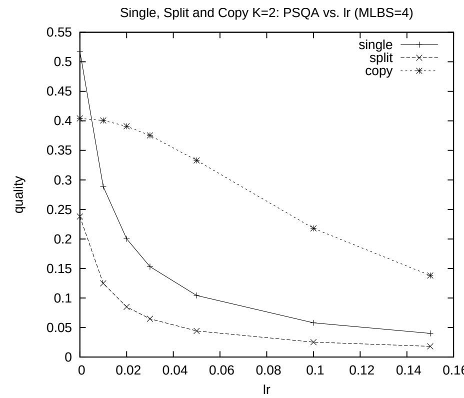

Fig. 3. Multi-source streamings with big bursts, MLBS=4

5. 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 from one node to the other (for reference purposes), sending two complete copies of the stream through different paths, and sending two disjoint substreams whose union recomposes the original one.

We are able to evaluate the resulting perceived quality associated with these extreme architectures. The main conclusions are: (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 the signal twice obviously leads to a much better quality, even if, as expected, losses are less concentrated in this case; (iii) sending disjoint pieces of the stream is not useful (under our simplifying assumptions).

This study suggests that some intermediate point between the extreme cases we considered, perhaps with a higher number of flows per stream, can be an interesting solution. The paper proposes a global methodology that allows to look for such a scheme. It also strongly suggest to look for more realistic models, perhaps including cost aspects, to allow for a deeper analysis of different splitting cases.

6. 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, June 2004.

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

[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, Sept. 1960.