Optimal Quality-of-Experience Design for a P2P Multi-source Video Streaming

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*—We consider the design of a P2P network for the distribution of real-time video streams through the Internet. We follow a multi-source approach where the stream is decomposed into several flows sent by different peers to each client. The goal is to resist to the frequent moves of the peers entering and leaving the network. We analyze our approach using the recently proposed PSQA technology which allows to obtain an accurate (and automatic) numerical evaluation of the quality as perceived by each client. Our transmission technique includes the use of an arbitrary amount of redundancy in the signal, whose specification is a part of the dimensioning process, and it works with *very low signaling overhead*. We illustrate with real data how the overall system allows to compensate efficiently the possible losses of frames due to peers leaving the network.**

I. Introduction

Recent advances in networking allows today to share and to store all types of information, in particular in the multimedia field, with a corresponding considerable increase in the demand for distributing and accessing video streams. This brings, at the same time, a new set of hard research issues, including how to provide enough bandwidth, QoS guarantees and copyright protection.

One of the main solutions to this set of problems is based on Peer-to-Peer (P2P) systems. This architecture has been extensively used for decentralizing the delivery process among the connected nodes, diminishing the server load or avoiding any server at all. In this paper we are interested in applying the P2P infrastructure to real-time video distribution (live TV). Real-time video streaming has strong constraints that lead to a series of specific technical problems. The most important one is that video streaming quality, as *perceived* by the user (also known as *Quality of Experience* (QoE)), is very sensitive to frame losses [1]. Therefore, using P2P for delivering real-time video means to deal with one of the most important characteristics of those systems: peers connect and disconnect very frequently, in an autonomous and completely asynchronous way. The resources of the network as a whole are then also highly dynamic. For providing a minimum QoE, the design of the P2P architecture must mitigate the impact of losses coming from peers disconnections. This is the main challenge in the P2P network design: to offer the quality needed by the clients in a highly varying environment.

In order to diminish the effect of losses caused by the P2P's dynamics, we proposed in [2] a *multi-source streaming* technique where the flow is transported decomposed into a set of K subflows. The whole set of subflows allows for an arbitrary amount of redundancy in the signal, and the architecture is such that each peer receives each subflow from a different one. In [1], [3] some aspects of this technique are also studied. To be complete, in [4], we discuss some optimization aspects related to the design of the structure of a P2P network.

The specific contributions of the present paper can be summarized as follows. First of all, we analyze the impact of frame losses on the *perceived* quality by the end-user (previous works considered losses at the packet level). We address the problem of measuring the *perceived* quality by means of the PSQA (Pseudo Subjective Quality Assessment) technology, improving from [5]. Second, we analyze the behavior of the system when peers leave. We decided to focus on an important "conditional" aspect, the impact of the peers' dynamics on the perceived quality when at least one of the nodes sending the signal at the customer site left. Using real data of the connection/disconnection peers' process in a real-time video service coming from a medium-size ISP, we analyze how to maximize the delivered QoE of this *multi-source technique*. The main objective is to provide a methodology that can be used in other P2P systems with similar constraints and users' behavior, in order to provide the best possible quality.

A few studies consider the implementation needed to split and to merge stream contents, especially when it is real-time video streaming. Some keywords associated with this approach are Multiple Description Coding (MDC) [6], [7] and Network Coding [8]–[10]. If we consider the P2P architecture and not only the streaming mechanism, some related work are SpreadIt [11], CoopNet [12] and SplitStream [13]. Besides the academic studies, some commercial networks for video distribution are available. Focusing on the P2P world the most successful ones are PPlive (`www.pplive.com`) and TVUnetwork (`tvunetworks.com`).

The remainder of this paper is organized as follows. Section II introduces the different methodologies that can be used for measuring video quality. Section III studies the peers' dynamics and presents the *multi-source streaming* technique, considering the different frame types of a video stream (coded into MPEG-4). Some numerical results are presented in Section IV. Section V contains our main conclusions.

II. Video Quality Measurements

In what follows, we discuss the different methods for assessing the perceived quality in a video delivering system.

A. Subjective and Objective Quality Assessment Methods

Subjective quality assessment methods (*subjective testing*) measure the perceived quality from the user's perspective, by means of a panel of human subjects, which will evaluate a series of short video sequences according to their own personal notion of quality. For video content, the ITU-R BT.500-11 [14] recommendation provides guidelines on how this should be performed. Like other subjective assessment techniques, the result of these test types is a Mean Opinion Score (MOS), that is, a numerical expression of perceived quality. The chosen grade is a 11-point scale, which spans from worst to best quality. Subjective tests are very time-consuming and expensive in manpower, which makes them hard to repeat often. Furthermore, given their nature, they are obviously not suitable for real-time operation.

Objective tests consist in techniques allowing to compare the sent signal to the receiver. They do not need to use human subjects, but the correlation with values coming from subjective tests are often quite poor. Moreover, they need to access the original stream, which precludes their use in real time (there are a few exceptions to this, but correlation with subjective values is still too bad). Some of the most well-known objective assessment methods are Peek signal to noise ratio, ITS' Video Quality Metric, EPFL's Moving Picture Quality Metric and Color Moving Picture Quality Metric.

B. Pseudo Subjective Quality Assessment (PSQA)

PSQA [5] is a technique based on merging subjective assessments with a statistical learning tool (a Random Neural Network, or RNN, which allows to produce subjective-like quality estimations. Its main advantage is that it provides results very close to actual MOS values while being cheap and suitable for real-time applications. To implement PSQA, three main steps must be followed: (a) a set of quality-affecting parameters must be selected; (b) a (set of) subjective tests session(s) must be performed, and (c) an RNN must be chosen, trained and validated.

We must start by choosing the parameters that are likely to have the highest impact on quality. We focused our study on a specific one: the loss rate of video frames. Actually, we used one loss rate per frame type (see below), but to simplify the description, let's assume in this paragraph that we consider the global loss rate, LR. For step (b), we chose some representative video sequences. Many copies of each original sequence were then built, each one associated with a value of LR. These copies were produced using a testbed and the human subjects evaluated their quality following a subjective test, the result being a numerical (MOS) value associated with each sequence (then with each value of LR). After that, a statistical analysis was performed, in order to detect (and eliminate, if necessary) the bad observers (defined as being in strong disagreement with the majority). Finally, the average of the scores (MOS) given by the remaining subjects to each video sequence was computed.

After step (b) we have a table associating the MOS with each copy of the original sequence, thus with each value for LR. In step (c), we follow a training process, where an RNN learns the mapping from the values of the selected parameters into quality. As usual with learning, a validation phase closes the process. The output of it is a function able to build a MOS value from *any* set of values of the selected parameters (see [5] for details). Therefore, we can calculate the *instantaneous* perceived quality at any time t using this function, from measures of the chosen parameters.

C. Pseudo Subjective Quality Assessment with Frame Losses

In the most adopted standard specifications for video (we consider a MPEG-4 codec), the transmission units are the *frames*, which belong to three main types: Intra (I), Predicted (P) and Bidirectional (B). Besides their frequency and mean size, the frames also differ in how the loss of each of them influences the QoE.

Let us now look at the impact on the QoE of the combination of the loss rates per frame type, denoted by LR_I, LR_P and LR_B. From a set of video sequences of about 10 seconds each, we generated 204 different randomly (uniformly) chosen. Each evaluation point is defined by a set of values for the tuple $\tau = (LR_I, LR_P, LR_B)$. Using a simplified Gilbert model as in [15] to simulate a frame drop history which was applied to the original video sequences, we obtained the set of modified ones. Each modified sequence was evaluated by 10 subjects and the MOS was computed following [14]. The set of MOS values was then employed in order to calibrate a 3-layer RNN, with 9 neurons. The RNN was trained with 160 sequences, resulting in a small *Mean Squared Error* (≈ 0.025), and then validated with the remaining 44 sequences.

The 3-layer RNN provides a function $Q()$ mapping τ into perceived quality, illustrated in Figures 1 and 2. In Figure 1, quality degrades quickly with an increment in the loss rate of frames I and P. For example, for $LR_P \geq 10\%$ the quality is less than 6 (between good-fair) and the impact of P-frames' losses is a bit higher than for I-frames. Figure 2 shows that the quality degrades slowly with an increment in the loss rate of B-frames, as expected.

III. Multi-source Streaming Technique Considering Heterogeneous Peers' Lifetimes

In this paper, a server producing a live video stream splits it into several flows, with some amount of redundancy in them (that is, together they send more data than contained in the original signal). This server sends each of these flows to a specific set of peers, which, in turn, send the received flows to other peers. Therefore, from the client's point of view, we have a *multi-source delivering system*. In our approach, the P2P system has the necessary knowledge about the peers and the network state, and can thus decide which peer will serve

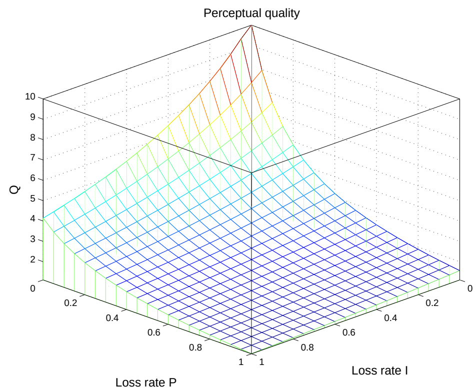

Fig. 1. The quality degrades quickly with an increment in the loss rates of frames I and P.

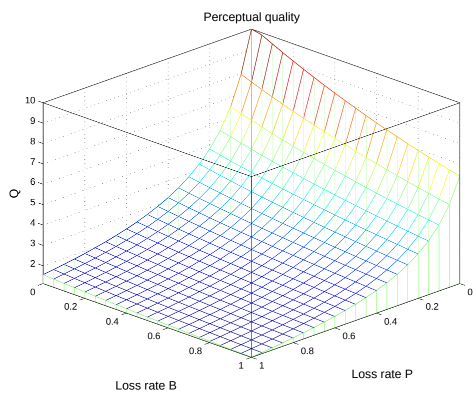

Fig. 2. The quality degrades slowly with an increment in the loss rate of frames B.

which other node, and how flows must be split and merged (see [4] for the structural P2P design problem). The network topology is then re-built periodically by the central controlling system, every T seconds.

We analyze here what happens between a network re-built operation (let's say at time 0), and next one happening at time T. A generic peer starts the cycle by receiving the video stream from K heterogeneous and independent peers (also refereed as *server peers*). The peers' heterogeneity is based on the their connection-time in the network (peer's lifetime).

A. Peer's Probability Connection and Multi-source Streaming

Assume that the connection times of the servers are independent and exponentially distributed, with parameter λ_k for server k. Let us denote by $N_k(t)$ the binary r.v. equal to 1 iff server k is connected at t, and by $\vec{N}(t)$ the vector $\vec{N}(t) = (N_1(t), \cdots, N_K(t))$. Then, for any *configuration* $\vec{n} \in \{0,1\}^K$,

$$\Pr(\vec{N}(t) = \vec{n}) = \prod_{j:n_j=1} e^{-\lambda_j t} \prod_{j:n_j=0} (1 - e^{-\lambda_j t}). \quad (1)$$

Assume we order the servers according to their mean connection time, with the best at the end, that is, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_K$. If B Kbps is the total bandwidth needed by the stream, server k sends the fraction $y_k B$ Kbps. We call *weight* of server k the fraction y_k. As we want that the best servers send most of the data, we set $y_{k+1} = \gamma y_k$, with $\gamma > 1$. This means that $y_k = \gamma^{k-1} y_1$, with $y_1 = (\gamma - 1)/(\gamma^K - 1)$.

We also add some redundancy $r \in [0,1]$ to the global flow; $r = 0$ means no redundancy, $r = 1$ means that any frame is sent twice. The total bandwidth employed is thus $BW^{red} = (1+r)\,B$ Kbps. The redundancy is distributed in the following way. A fraction r of the original data sent by server k is also sent by the other servers, proportionally to their weights. The total bandwidth BW_k^{red} used by server k is then

$$BW_k^{red} = y_k B + \sum_{j=1, j\neq k}^{K} r y_k B \frac{y_j}{1 - y_k} = (1+r)\, y_k B,$$

where $r y_k B y_j/(1 - y_k)$ is the bandwidth used by server k to stream the redundancy of substream j. The technique implies that each frame is sent either once or twice, but no frame is sent more than twice (see [2] for the implementation issues).

Consider $\vec{n} \neq \vec{1}$, where $\vec{1} = (1, \cdots, 1)$. So, if the *configuration* is $\vec{n} \neq \vec{1}$, *at least one server left the network*. Let j be such that $n_j = 0$. The fraction of lost data due to the lack of the jth substream is:

$$LR_j^{red} = y_j - \frac{r y_j}{1 - y_j} \sum_{i:n_i=1} y_i. \quad (2)$$

The total loss rate at configuration $\vec{n}$ is then

$$LR_{\vec{n}}^{red} = \sum_{j:n_j=0} LR_j^{red} = 1 - \Big(\sum_{i:n_i=1} y_i\Big)\Big(1 + r \sum_{j:n_j=0} \frac{y_j}{1 - y_j}\Big). \quad (3)$$

B. I/P/B-Frames Analysis

As stated before, this paper analyzes the impact on the QoE of losses of each video frame type. In our implementation (see [2] for details), we can control both the distribution of each type of frame separately (that is, the weights, or directly the γ parameter) and the redundancy factor (the r parameter) per frame type (I, P or B). The loss rate of each frame type is given by Equation (3), where we use the weights and redundancy for that frame type. We denote by $LR_{x,\vec{n}}$ the loss rate of x frames, where x is I, P or B, when the configuration is $\vec{n}$. The QoE in that configuration is thus $Q(LR_{I,\vec{n}}, LR_{P,\vec{n}}, LR_{B,\vec{n}})$. The main objective of Section IV is to maximize the mean QoE value, considered at the end of the cycle (worst case analysis), when at least one server is down, defined by:

$$\mathbf{E}(QoE) = \sum_{\vec{n}\neq\vec{1}} Pr(\vec{N}(T) = \vec{n}) Q(LR_{I,\vec{n}}, LR_{P,\vec{n}}, LR_{B,\vec{n}}),$$

where the peer's probability connection for each configuration is defined by Equation (1).

IV. Optimization Results

In what follows, we analyze the behavior of the system when at least one peer leaves the network. This scenario is particularly interesting, due to the fact that the system should be robust in order to provide the best possible quality. The presented results focus on the evaluation of all concerned parameters for maximizing $\mathbf{E}(QoE)$. The needed bandwidth for delivering the original video is $B = 512$ Kbps. We will consider the two cases where the total bandwidth (including the redundancy r) is equal to 640 and to 768 Kbps, with $T = 10$ secs. The results are based on the connection/disconnection peers' process measured at the real-time video service of a medium-size ISP. Figure 3 shows the peers' lifetime in this ISP.

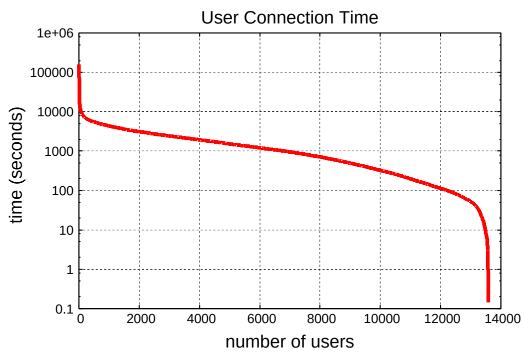

Fig. 3. Real data of a live-video service from a medium-size ISP. For a given number of users x, the curve gives $F(x) = y$ meaning that x users have connection time $\geq y$.

The optimization results were obtained using the *fminsearch* function of Matlab[1], which finds the minimum of a scalar function of several variables, starting at an initial estimation. For all the results, we randomly chose 10 starting points, with $2 \leq K \leq 10$, and the maximum mean quality among all provided results was selected.

For the scenario we are considering, first observe that if we have only one *server peer* and it leaves the network, the QoE is obviously zero. Now, in real systems, peers have sometimes bandwidth restrictions. We will consider two cases: either the peers send information without bandwidth restrictions or the bandwidth is restricted to BW^{red}/K. The two scenarios result in different restrictions on the γ and r parameter values.

Limitation on the Total Bandwidth. We start by the case where the bandwidth is limited by its total amount, without caring about the bandwidth of each *server peer*. We report our results on Tables I and II (for $r = 25\%$ and $r = 50\%$).

Limitation on the Individual Bandwidth. Now, consider that each *server peer* has a bandwidth limitation. Tables III and IV show the results.

The following conclusions can be listed. (1) As expected, more bandwidth results in a better quality for the delivered video. (2) We can see that, for P-frames, the largest quality is achieved when γ_P is around 2. This means that *server peers* that stay longer into the system will be responsible for delivering the most important information. These results match with practical experiences, where P-frames have a crucial role on the quality perceived by the end-user. (3) After the P-frame in importance order with respect to the quality, we have I-frames and then B-frames, confirming the results of the *subjective testing* sessions. (4) The redundancy value is counterbalanced by the γ value: for greater γ we can set a smaller value to the redundancy factor, with the following interpretation: if the largest part of the information is delivered by the most stable peers, it is not necessary to use a high redundancy factor.

TABLE I

γ AND REDUNDANCY FACTORS OPTIMIZATION (25% REDUNDANCY)

# Servers	$\mathbf{E}(QoE)$	γ_I	γ_P	γ_B
1	0	-	-	-
2	3.493	1.189e+00	1.999e+00	1.146e+00
3	5.593	1.000e+00	1.999e+00	1.712e+00
4	7.016	1.535e+00	1.992e+00	1.999e+00
5	7.857	1.000e+00	1.999e+00	1.589e+00
6	8.606	1.524e+00	1.995e+00	1.744e+00
7	9.076	1.109e+00	1.999e+00	1.921e+00
8	9.332	1.000e+00	1.999e+00	1.993e+00
9	9.479	1.890e+00	1.999e+00	1.999e+00
10	9.591	1.999e+00	1.999e+00	1.898e+00
# Servers	$\mathbf{E}(QoE)$	r_I	r_P	r_B
1	0	-	-	-
2	3.493	9.797e-01	2.547e-01	2.171e-01
3	5.593	9.999e-01	2.942e-01	8.694e-02
4	7.016	9.999e-01	2.252e-01	2.941e-01
5	7.857	9.223e-01	3.228e-01	1.491e-06
6	8.606	9.378e-01	2.789e-01	6.931e-06
7	9.076	9.949e-01	2.583e-01	4.065e-04
8	9.332	9.991e-01	2.154e-01	8.857e-02
9	9.479	9.999e-01	1.366e-01	9.171e-02
10	9.591	9.999e-01	1.829e-01	6.647e-02

TABLE II

γ AND REDUNDANCY FACTORS OPTIMIZATION (50% REDUNDANCY)

# Servers	$\mathbf{E}(QoE)$	γ_I	γ_P	γ_B
1	0	-	-	-
2	5.011	1.258e+00	1.874e+00	1.615e+00
3	6.636	1.147e+00	1.863e+00	1.999e+00
4	7.865	1.999e+00	1.999e+00	1.999e+00
5	8.711	1.420e+00	1.993e+00	1.993e+00
6	9.024	1.644e+00	1.999e+00	1.601e+00
7	9.324	1.809e+00	1.999e+00	1.999e+00
8	9.610	1.999e+00	1.999e+00	1.998e+00
9	9.669	1.822e+00	1.995e+00	1.999e+00
10	9.766	1.910e+00	1.999e+00	1.914e+00
# Servers	$\mathbf{E}(QoE)$	r_I	r_P	r_B
1	0	-	-	-
2	5.011	9.999e-01	6.629e-01	4.926e-06
3	6.636	9.999e-01	5.739e-01	3.093e-01
4	7.865	7.043e-01	6.323e-01	1.456e-01
5	8.711	9.999e-01	5.662e-01	3.887e-01
6	9.024	9.999e-01	4.340e-01	8.044e-01
7	9.324	5.996e-01	5.650e-01	5.259e-01
8	9.610	7.697e-01	6.520e-01	1.957e-01
9	9.669	4.824e-01	7.285e-01	4.523e-04
10	9.766	8.409e-01	7.082e-01	6.855e-06

To explore the robustness of the optimization procedure, we selected some cases with more starting points in order to obtain the best $\mathbf{E}(QoE)$. Just for illustration, let us consider

[1] © 1984-2007 The MathWorks, Inc.

TABLE III
γ AND REDUNDANCY FACTORS OPTIMIZATION (25% REDUNDANCY)

# Servers	$\mathbf{E}(QoE)$	γ_I	γ_P	γ_B
1	0	-	-	-
2	3.557	1.000e+00	1.960e+00	1.607e+00
3	5.442	1.296e+00	1.999e+00	1.771e+00
4	7.020	1.108e+00	1.999e+00	1.999e+00
5	7.972	1.220e+00	1.999e+00	1.797e+00
6	8.719	1.999e+00	1.997e+00	1.999e+00
7	9.064	1.409e+00	1.999e+00	1.661e+00
8	9.319	1.999e+00	1.999e+00	1.692e+00
9	9.510	1.995e+00	1.932e+00	1.974e+00
10	9.586	1.999e+00	1.996e+00	1.949e+00
# Servers	$\mathbf{E}(QoE)$	r_I	r_P	r_B
1	0	-	-	-
2	3.557	9.913e-01	2.953e-01	5.410e-04
3	5.442	9.999e-01	1.705e-01	4.744e-01
4	7.020	9.999e-01	2.279e-01	2.610e-01
5	7.972	9.999e-01	1.988e-01	3.856e-01
6	8.719	8.286e-01	3.305e-01	9.977e-06
7	9.064	9.999e-01	2.462e-01	2.373e-01
8	9.319	8.124e-01	2.946e-01	6.276e-06
9	9.510	7.986e-01	3.428e-01	5.984e-05
10	9.586	1.550e-01	3.959e-01	4.255e-02

TABLE IV
γ AND REDUNDANCY FACTORS OPTIMIZATION (50% REDUNDANCY)

# Servers	$\mathbf{E}(QoE)$	γ_I	γ_P	γ_B
1	0	-	-	-
2	4.787	1.000e+00	1.909e+00	1.000e+00
3	6.999	1.088e+00	1.991e+00	1.850e+00
4	7.865	1.999e+00	1.999e+00	1.999e+00
5	8.711	1.420e+00	1.993e+00	1.993e+00
6	9.188	1.273e+00	1.999e+00	1.761e+00
7	9.470	1.495e+00	1.998e+00	1.896e+00
8	9.610	1.999e+00	1.999e+00	1.998e+00
9	9.669	1.822e+00	1.995e+00	1.999e+00
10	9.752	1.999e+00	1.967e+00	1.655e+00
# Servers	$\mathbf{E}(QoE)$	r_I	r_P	r_B
1	0	-	-	-
2	4.787	9.929e-01	6.576e-01	4.480e-03
3	6.999	9.999e-01	6.552e-01	1.138e-01
4	7.865	7.043e-01	6.323e-01	1.456e-01
5	8.711	9.999e-01	5.662e-01	3.887e-01
6	9.188	9.810e-01	6.599e-01	1.057e-01
7	9.470	9.999e-01	6.222e-01	1.825e-01
8	9.610	7.697e-01	6.520e-01	1.957e-01
9	9.669	4.824e-01	7.285e-01	4.523e-04
10	9.752	9.574e-01	6.826e-01	1.589e-05

the case with $r = 25\%$ and $K = 2$ (Table I). We randomly chose 70 starting points, with $\mathbf{E}(QoE)$ values in $[2.571, 3.716]$ and mean 3.250. Coming back to Table I, we can see that this does not basically change our pseudo-optimal points.

Based on these results, the maximum $\mathbf{E}(QoE)$ improves when the number of *server peers* increases. Our methodology suggests the development of systems with many *server peers* in order to provide the best possible $\mathbf{E}(QoE)$. However, increasing the number of *server peers* can result, for instance, in hard problem related to the management of message delivering or too much overhead when merging the final video itself. Our results show that, with a minimum number of 7 *server peers*, the $\mathbf{E}(QoE)$ is ≥ 9.0, which means an excellent quality. In [2], we discuss about the main problems of implementing the *multi-source streaming technique*. Subjective experiments show that a variation of plus or minus one point of QoE is not perceived by human eyes. Therefore, similar P2P systems can provide an excellent quality level using, say, between 7 and 10 *server peers* (that is, substreams).

V. CONCLUSIONS

In this work, we considered a *multi-source streaming technique*, based on the heterogeneous peers' lifetimes, for designing a P2P structure for delivering real-time video. This technique provides good quality to the end-users, mitigating the impact of losses coming from peers disconnections. The main results of this paper concern the joint impact of different frame type losses on the QoE, using the PSQA methodology, and how to identify an optimal parameter setting, to obtain the best possible quality level for a given peers' dynamics.

REFERENCES

[1] P. Rodríguez-Bocca, H. Cancela, and G. Rubino, "Perceptual quality in P2P video streaming policies," in *50th IEEE Global Telecommunications Conference (GLOBECOM'07)*, Washington DC, United States, 26-30 November 2007.

[2] P. Rodríguez-Bocca, G. Rubino, and L. Stábile, "Multi-Source Video Streaming Suite," in *7th IEEE International Workshop on IP Operations and Management (IPOM'07)*, San José, California, United States, October 31 - November 2 2007.

[3] P. Rodríguez-Bocca, H. Cancela, and G. Rubino, "Video Quality Assurance in Multi-Source Streaming Techniques," in *IFIP/ACM Latin America Networking Conference (LANC'07). Best paper award by ACM-SIGCOMM*, San José, Costa Rica, October 10-11 2007.

[4] H. Cancela, F. R. Amoza, P. Rodríguez-Bocca, G. Rubino, and A. Sabiguero, "A robust P2P streaming architecture and its application to a high quality live-video service," in *4th Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS'07)*, Puerto Varas, Chile, November 25-29 2007.

[5] 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.

[6] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Processing Magazine*, September 2001.

[7] J. Apostolopoulos and S. Wee, "Unbalanced multiple description video communication using path diversity." in *IEEE International Conference on Image Processing*, Oct. 2001, pp. 966–969.

[8] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *41st Conference on Communication Control and Computing.*, Oct 2003.

[9] P. Chou, Y. Wu, and K. Jain, "Network coding for the internet," *IEEE Communication Theory Workshop, Capri, IEEE.*, 2004.

[10] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory.*, 2000.

[11] H. Deshpande, M. Bawa, and H. Garcia-Molina, "Streaming live media over peer-to-peer network," Stanford University, 2001., Tech. Rep., 2001.

[12] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking," in *NOSSDAV '02*. New York, NY, USA: ACM Press, 2002, pp. 177–186.

[13] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream: High-bandwidth content distribution in a cooperative environment," in *IPTPS '03: Proceedings of the Second International Workshop on Peer-to-Peer Systems*, 2003, p. 6.

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

[15] S. Mohamed, G. Rubino, and M. Varela, "Performance evaluation of real-time speech through a packet network: a Random Neural Networks-based approach," *Performance Evaluation*, vol. 57, no. 2, pp. 141–162, May 2004.