

Effects of P2P Streaming on Video Quality

Csaba Kiraly, Luca Abeni, Renato Lo Cigno
DISI - University of Trento, 38100 Trento, Italy
{csaba.kiraly,luca.abeni,renato.locigno}@disi.unitn.it

Abstract—P2P TV distribution is going commercial, and the video quality delivered to users becomes of the utmost importance. However, the impact of P2P distribution on the video quality is not completely understood yet, especially in live streaming situations. This work addresses the impact of P2P distribution when the delay of the playout is limited, as it must be in any true live TV service. A methodology for the evaluation (using standard objective video quality metrics) is proposed, showing that different ways of grouping frames in chunks for the distribution can lead to very different quality when the system is overloaded.

I. INTRODUCTION

P2PTV systems are distributing popular channels all over the world¹. The experience of users, however, can be compromised by the large delays introduced by some P2P systems, and by the loss of some parts of the audio or video stream (chunks) that can happen when the system is overloaded, or when the various peers do not distribute the media streams properly. Such delays and chunk loss can be controlled by reducing the system load (hence, by reducing the encoding bitrate of the audio and video), but this requires some quality/bitrate trade-offs that can further compromise the quality perceived by end users.

Recently, some work has been done to analyse the diffusion delay and the amount of lost chunks in P2P streaming systems, and to optimise the streaming performance through proper *scheduling* algorithms [1], [2], [3]. The goal of such work is to reduce the traffic between different peers by sending the right part of the video to the right peer at the right time, so that duplicated chunks are avoided and all the bandwidth available in the overlay can be correctly exploited. In other words, this approach is based on optimising the network performance so that all the parts of the encoded media are correctly received, in the hope that this is sufficient to improve the quality perceived by the end user. However, the quality perceived by the user also depends on the encoding bitrate, on the audio and video codecs used, etc.

In this work, a more direct measure of the received media quality is used to analyse the relationship between playout delay, chunk loss, and the final media quality. First of all, a methodology is proposed for interfacing standard P2P streaming simulators (which are able to evaluate streaming delays

and chunk loss ratios) with a quality evaluation tool (which is able to provide an estimation of the quality of the media received by each P2P client). Then such a methodology is used to evaluate the relationship between the encoding bitrate and the chunk loss rate, and to find proper trade-offs that maximise the user-perceived quality.

II. VIDEO ENCODING AND CHUNKISATION

Traditional P2P distribution systems generally broadcast a file to multiple peers by splitting it in fixed-size pieces named *chunks*. Many P2P streaming systems adapt this approach to media streaming, by dividing the encoded media in chunks. As a result, chunks generally have a fixed size in bytes, independently from the properties and characteristics of the encoded media; this way of splitting a stream in chunks is referred in this paper as “media unaware”. In contrast to this approach, a *media aware* distribution strategy uses some knowledge about the encoded media stream to optimise its distribution and improve the streaming performance. From now on, the process of splitting a media stream in chunks will be referred in this paper as “chunkisation” (hence, this paper will talk about “media aware chunkisation” and “media unaware chunkisation”).

Without focusing on any particular encoding algorithm (which is out of the scope of this paper), it is possible to say that a coded audio/video stream is composed of *frames*, each of which starts with one or more headers and then contain the compressed data. If such headers are lost, it is not possible to decode the rest of the frame.

Another important observation is that most of the modern video encoding algorithms use three different kinds of frames: *I frames* (which do not depend on other frames), *P frames* (Predicted frames, which are coded as deltas with respect to a prediction—including motion compensation—from a previous frame), and *B frames* (Bi-predicted frames, which are coded as deltas respect to a previous and a following frames). If the frame on which a P or a B frame depends is not correctly decoded, then also this frame cannot be decoded correctly, even if it has been correctly received. For this reason, losing parts of an I frame is more problematic than losing parts of a P or B frame, and if an I frame is lost, all the P and B frames directly or indirectly depending on it are damaged too. This suggests that a predicted frame should be transmitted in the same chunk containing the frames on which it depends.

Summing up, the previous discussion suggests that the tolerance to chunk loss can be increased by inserting an integer number of frames in a chunk (and not splitting frames between

This work is supported by the European Commission through the NAPA-WINE Project (Network-Aware P2P-TV Application over Wise Network – www.napa-wine.eu), ICT Call 1 FP7-ICT-2007-1, 1.5 Networked Media, grant No. 214412

¹For example, see <http://www.pplive.com>, <http://www.soapcast.com>, <http://www.tvants.com>, or <http://www.zattoo.com>.

different chunks) and by grouping predicted frames in the same chunk as their reference frames. Such a grouping can be easily performed by exploiting the regular structure of many modern video encoding algorithms² which periodically repeat the same sequence of frame types (called GOP - Group Of Pictures) with a period named *GOP size*. Chunks containing an integer number of frames which is a multiple of the GOP size have the properties required in the discussion above. Moreover, this kind of chunkisation can reduce the variability in the chunks sizes, helping some P2P distribution algorithms that have problems in coping with variable-size chunks and making the media aware chunkisation more efficient.

Finally, note that the term “quality” has been used very informally until now: it has been argued that a media-aware chunkisation can improve the quality when some chunks are lost, but it has not been provided any way to measure or quantify such quality improvements. Obviously, to verify that a chunkisation strategy can actually perform better than another one, the quality improvements need to be measured. The approach proposed in this paper is based on quantitative evaluation techniques, based on some well-defined quality metrics. At the network level such metrics are well defined as the measure of loss ratios, delays, and jitters, but it is not clear how such values map to the user experience. Hence, the next section introduces some metrics to be used as a quantitative measure of the distortion introduced in the media (the PSNR), or of the quality perceived by the end user (SSIM) [4].

III. QUANTITATIVE QUALITY EVALUATION

In this paper, objective and quantitative metrics are used to measure the quality of experience P2P-TV users can have. Similar measurements have already been performed in the past for evaluating the amount of artefacts and the distortion introduced by network streaming (when some packets are lost) in encoded digital media. For example, the EvalVid [5] tool is able to evaluate the video quality loss introduced by packet loss on RTP streams, by using the PSNR or/and the SSIM as a quality metric. This paper uses an approach similar to the EvalVid’s one, extending it to P2P streaming systems (instead of RTP).

Focusing on video, the colour of each pixel can be seen as a point in a three-dimensional space (the so called *colourspace*). Hence, it can be described by three components which depend on the used coordinate system (for example, they can be the Red, Green, and Blue components if the RGB colourspace is used, or the luminance, red chrominance, and blue chrominance components if the YCrCb or YUV colourspace is used). Each component can be coded on Nb bits, and its values can be signed or unsigned. Since the PSNR is generally computed on single components, there can be many flavours of it. In this paper, the PSNR computed on component c will be indicated as c -PSNR (for example, Y-PSNR is the PSNR for the luminance component in the YUV colourspace). In

general, given an “original” and an “encoded/reconstructed” video frame with $m \times n$ resolution, for any component c the c -PSNR is computed as

$$c - \text{PSNR} = 10 \log_{10} \left(\frac{\max_I^2}{\text{MSE}} \right) \quad (1)$$

where

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2 \quad (2)$$

and $I(i, j)$ is the value of the c component for pixel (i, j) in the original video, $K(i, j)$ is the value of the c component for pixel (i, j) in the encoded/reconstructed video; $\max_I = 2^{Nb} - 1$ is the absolute maximum value for component c .

In this paper, the Y-PSNR (i.e., the PSNR computed on the luminance component) will be used as a measure of the distortion introduced by the P2P streaming system. This choice has been made for the sake of simplicity, because the luminance component is perceptively dominant, and because some preliminary tests showed that the PSNR computed on the other components (Cr and Cb) has a similar behaviour than Y-PSNR (i.e., it is affected by the chunk loss in the same way).

The Structural Similarity Index SSIM[4] is more complex than PSNR and we refer to the original paper for details. The idea is to represent better the errors that are perceptible and annoying to users. The index is not based on the direct difference of the components amplitude, but it is a weighted composition of the luminance difference, the contrast difference and the difference in the structure (i.e., general organization) of the image. SSIM varies between -1 and 1 , but acceptable degradation normally implies $\text{SSIM} > 0.8$.

Note that when evaluating the distortion introduced by chunk loss in the received media, two problems have to be addressed:

- 1) A chunk generally contains multiple encoded frames. Hence, when a chunk is lost the number of received audio or video frames is different from the original number of frames (this is the “missing frames” problem). Hence, comparing the received frames with the source can be problematic;
- 2) Some metrics used for evaluating the quality (for example, the PSNR) cannot cope with reconstructed frames which are identical to the original ones (in particular, if the reconstructed frame is identical to the original one the MSE is 0 and the PSNR is undefined).

For video streams, problem 1 (missing frames) can be addressed by “filling” the holes in the stream with a copy of the latest correctly decoded frame (in other words, if the n^{th} frame is missing, it is substituted with the last frame that has been correctly decoded). This is consistent with what many set-top-boxes do: when the stream cannot be correctly decoded, the video is not updated and a still frame is displayed. If more than one frame in a row are missing (as for instance when a large chunk is lost) the same frame is repeated over and over until the next correct frame is decoded.

²Note that audio is generally not considered because it often has a much smaller bitrate than the video

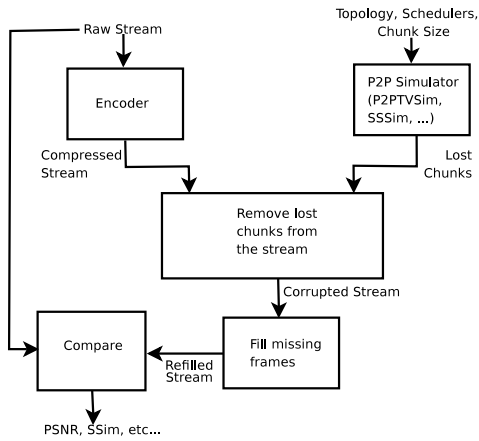


Fig. 1. The quality evaluation tool.

Problem 2 can be addressed by evaluating the PSNR between the received media and the original uncompressed (raw) media (not the encoded media before the transmission). In this way, the MSE will never be 0, because of the effects of the encoding. This approach has the advantage that it accounts for both the effects of lost chunks (which increase when the bitrate increases) and the artefacts introduced by the encoding (which increase when the bitrate decreases). In fact, the number of lost chunks depend on the media bitrate (if the media bitrate is too high, the P2P streaming system can be overloaded and too many chunks are lost); on the other hand, reducing the bitrate to avoid lost chunks can have a bad effect too, because it introduces encoding artefacts that reduce the PSNR. Hence, some kind of trade-off is needed. The proposed way to compute the PSNR allows to find such a trade-off. Finally, this approach is consistent with the one used by EvalVid. Given a video stream \mathcal{V} , both the PSNR and SSIM of the stream are computed as the average over all the frames of the relative quality metric.

Summing up, the evaluation of the quality provided by a P2P streaming system is performed as shown in Figure 1: first, the raw (non compressed) media stream is encoded at a specified bitrate, and the average size of a chunk is computed (note that media aware chunkisation uses 1 GOP = 1 chunk, hence there is not much variability in the chunk sizes. For this reason, the effects of chunk size variation are neglected during the simulation of chunks diffusion, and the average value is used). The chunk size value is then used to simulate the chunks diffusion by using a P2P streaming simulator such as P2PTVSim³ or SSSim [6]. As a result, a list of lost chunks is obtained (this depends on some parameters such as the used schedulers, the playout delay, etc.). Such list is used to remove the lost chunks from the encoded stream (this is where the chunkisation algorithm is used) obtaining a corrupted stream. The missing frames are replaced with duplicates of the previous frame, and the PSNR and SSIM are finally computed by comparing the “refilled stream” with

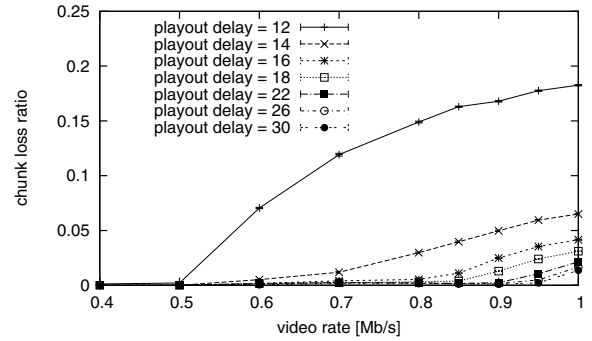


Fig. 2. Chunk loss rate as a function of video bitrate and playout delay.

the original raw (uncompressed) stream.

IV. EXPERIMENTAL RESULTS

The quality evaluation techniques and tools proposed in Section III can be used to check the effects of P2P diffusion on the video quality perceived by the end user.

To focus the evaluation on the effects of chunk loss on the video quality, and to compare different chunkisation strategies, the following reference scenario has been used:

- The simple LUC/RUP (Latest Useful Chunk; Random Useful Peer—see [3] for schedulers definition) scheduler has been used, simulating an overlay of 1000 peers, connected according to a random n-regular graph of degree 20;
- Each peer as well as the source has an upload bandwidth limit of 1 Mbit/s, the download bandwidth is much larger as in ADSL, and it is not a bottleneck;
- The simulations are based on the dissemination of 2000 chunks, but only the middle 100 are used for video quality evaluation;
- For the evaluation of the received video quality, the 25fps CIF “foreman” sequence has been used. The sequence has been looped 4 times, to obtain 1200 frames and 100 GOPs;
- The video is encoded with `ffmpeg`⁴ at the given target bitrate (in the first experiments, the H.264 video encoding algorithm has been used, through the `x264` codec⁵). Details about the scripts invoking `ffmpeg`, and the video sequences used for these experiments are available at <http://imedia.disi.unitn.it/QoE>;
- Each simulation has been repeated 50 times (on different random topologies) and an average of the results has been computed.

Before looking at the PSNR results, it is interesting to understand how the chunk loss depends on the bitrate. For this purpose, Figure 2 shows the chunk loss rate as a function of the target video bitrate, with various target playout delays. A chunk is “lost” for a peer when it is not received within its

³<http://www.napa-wine.eu/cgi-bin/twiki/view/Public/P2PTVSim>

⁴<http://www.ffmpeg.org>

⁵<http://www.videolan.org/developers/x264.html>

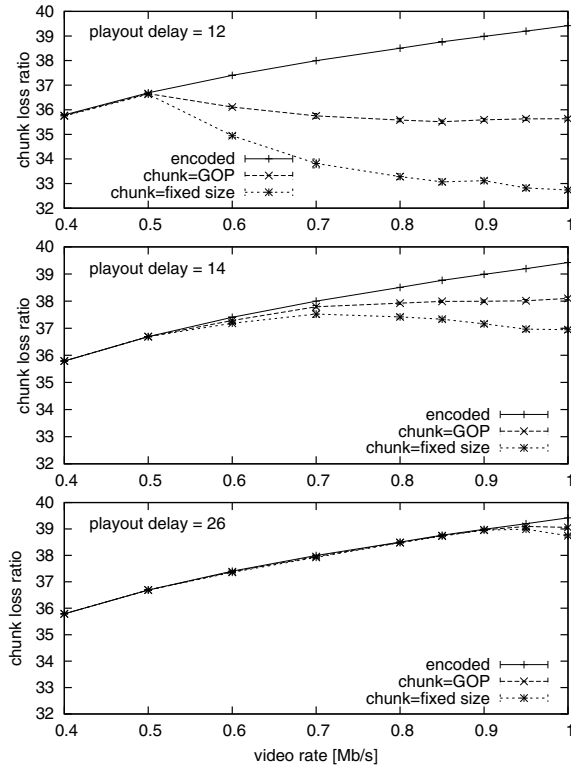


Fig. 3. PSNR with media-aware and unaware chunkizers, as a function of video coding rate, at various target playout delays.

playout time. The playout delay is expressed in “chunk times” (T_s), in our case: $T_s = \frac{12\text{Frames}/\text{GOP}}{25\text{Frames}/s} = 0.48s$

As the video coding rate increases, the chunk size increases as well, and therefore the diffusion of a given chunk takes longer. As a consequence, the chunk loss ratio also increases with coding rate at a given target playout delay.

As previously mentioned, many P2P streaming systems split the encoded stream into fixed size chunks without considering the underlying structure of the media stream. In Section II it was argued that a smarted, media-aware, chunkisation can provide better streaming performance in case of chunk loss. Now that a quality evaluation strategy has been presented and a reference scenario for the simulations has been set up, it is possible to confirm such intuition through a set of simulations and measurements of the video quality. In particular, a media-aware chunkisation strategy, inserting a GOP in each chunk, is compared with fixed-size chunkisation (using the average GOP size as a chunk size to achieve a fair comparison).

Figures 3 and 4 report the PSNR and SSIM at a given target playout delay, as a function of the media bitrate and the applied chunkisation scheme; all the figures show the quality (PSNR or SSIM) with no chunk loss (the quality of the encoded stream at the source, which is a function of the encoder and serves only for reference), the quality achieved with GOP sized chunks (media aware chunkisation), and the quality achieved when using fixed size chunks (media unaware chunkisation). This allows understanding the impact of loss ratio on the video

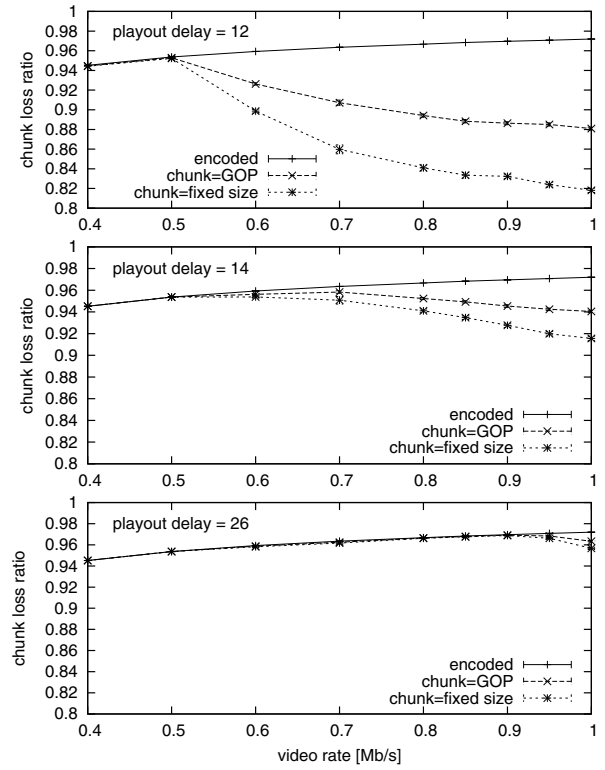


Fig. 4. SSIM with media-aware and unaware chunkizers, as a function of video coding rate, at various target playout delays.

quality. For example, the top plot of Fig. 3 shows what happens if the playout delay is set relatively low ($12T_s = 5.76s$): in this case, as soon as the chunk loss ratio starts to increase (at bitrate 0.5 Mbit/s, according to Fig. 2) the quality of the received stream becomes much smaller than the quality of the original encoded stream. In case of media aware chunkisation, the PSNR is always much higher than for media-unaware chunkisation.

Increasing the playout delay to $14T_s$ creates an interesting situation, which shows that for media aware chunkisation the PSNR can be increased by increasing the bitrate up to about 0.8 Mbit/s (in case of media unaware chunkisation, increasing the bitrate over 0.6 Mbit/s does not increase the PSNR, and increasing it over 0.8 Mbit/s even decreases the PSNR). Finally, the bottom part of the figures show the quality achieved with a large playout delay ($26T_s$), which reduces the chunk loss (see Fig. 2) to a minimum and allows to achieve high video quality by increasing the bitrate to around 0.95 Mbit/s.

In any case, by comparing all the figures it is possible to see that media aware chunkisation provides significantly higher quality (either in PSNR and SSIM) than fixed size chunks. Moreover, while in case of no chunk loss the PSNR increases with the bitrate, the distribution system cannot keep up with the increased amount of traffic maintaining the playback delay constraint.

Figure 5 analyses the effects of chunk and peer scheduling

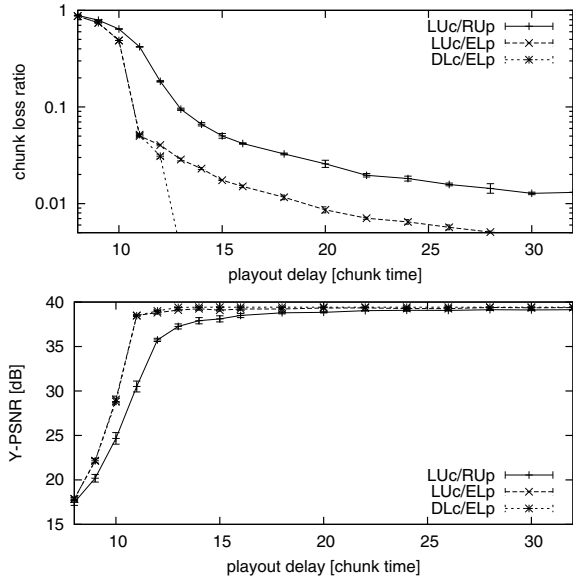


Fig. 5. Chunk loss rate and PSNR with different schedulers and GOP sized chunks, as a function of playout delay.

on the video quality, showing the chunk loss rate and PSNR values obtained with different schedulers (namely LUC/RUp, LUC/ELp, and DLc/ELp⁶). The analysis of the two plots immediately shows the impact of the loss rate on the PSNR, but also shows that the loss rate is a more sensitive measure: the PSNR rises within 2 dB from the encoded sequence as soon as the loss rate falls below 10%. Quality at higher loss rates degrades sharply. Thus a fast feedback from the network level identifying potentially dangerous situations where the loss rate starts increasing say, above 1% can prevent quality degradation that will be difficult to recover. The second observation is the impact of the scheduler: a good scheduler (in this case DLc/ELp) ensures lossless delivery with low delay (in this example 13 T_s which is less than 7 s), and guarantees unaffected PSNR (and quality) to the users. As a result of this analysis, it is possible to state that combining a media aware chunkisation (in this case mapping full GOPs to chunks) with a good, network-aware scheduler is the win-win approach to users' satisfaction.

Finally, some experiments have been performed to evaluate the impact of the video codec on the final video quality. To this purpose, the original video has been encoded by using x264 (an H.264 encoder) and some of the other video codecs provided by ffmpeg (mpeg4, mpeg2, and H.263). The results are shown in Fig. 6 and confirm the expectations: H.264 provides better quality than mpeg4, which provides better quality than mpeg2. The fact that mpeg2 provides better quality than H.263 was initially considered surprising, and is probably due to the fact that H.263 does not use B frames.

⁶ELp is a peer scheduler that selects the peer with the oldest most recent chunk (Earliest-Latest); DLc is an optimal chunk scheduler based on deadlines, that are embedded in chunks and incremented upon chunk transmission [3]

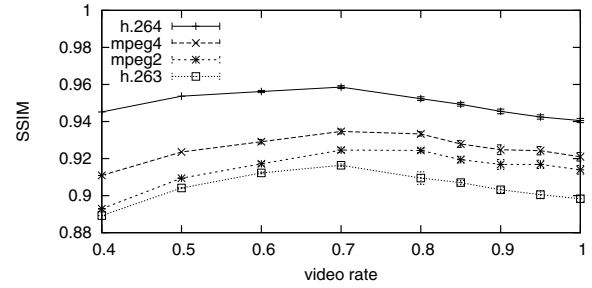


Fig. 6. SSIM achieved using different codecs with GOP chunkization, LUC/RUp scheduling and $T_s = 14$.

The other result to note is that all the codecs have a similar behaviour against lost chunks, which increase as the bitrate increases.

V. CONCLUSIONS AND FUTURE WORK

This paper describes a methodology for evaluating the video quality in P2P streaming systems, complementing “network-oriented” performance metrics such as the chunk diffusion delay or the chunk loss rate. The proposed approach is based on using a P2P system simulator for identifying the chunks lost during the stream diffusion. Such simulator is then interfaced to a tool that can actually remove the lost chunks from an encoded media stream to obtain a “received stream” which can be compared with the original one. The comparison is based on quantitative quality metrics such as PSNR and SSIM.

The proposed methodology allows to evaluate the effects of media encoding, chunkisation, scheduling, and network condition on the quality perceived by the end user. As an example, the quality improvement due to media aware chunkisation has been evaluated.

As a future work, more advanced media aware chunkisation strategies will be studied, and MDC or SVC will be considered. Moreover, the proposed tools and methodology will be used to better evaluate the effects of chunk and peer scheduling.

REFERENCES

- [1] Y. Liu, “On the minimum delay peer-to-peer video streaming: how realtime can it be?” in *Proc. of ACM MULTIMEDIA '07*, Augsburg, DE, Sept. 2007.
- [2] A. Couto da Silva, E. Leonardi, M. Mellia, and M. Meo, “A bandwidth-aware scheduling strategy for p2p-tv systems,” in *Proc. of P2P'08*, Aachen, DE, Sept. 2008.
- [3] L. Abeni, C. Kiraly, and R. Lo Cigno, “On the Optimal Scheduling of Streaming Applications in Unstructured Meshes,” in *Proc. of IFIP Networking 2009*, Aachen, DE, May 11–15 2009.
- [4] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, April 2004.
- [5] J. Klaue, B. Rathke, and A. Wolisz, “Evalvid - a framework for video transmission and quality evaluation,” in *Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, 2003, pp. 255–272.
- [6] L. Abeni, C. Kiraly, and R. Lo Cigno, “SSSim: a Simple and Scalable Simulator for P2P Streaming Systems,” in *Proc. of IEEE CAMAD'09*, Pisa, IT, June 12 2009.