

# A P2P Platform based on Rate-Controlled FGS encoding for Broadcast Video Transmission to Heterogeneous Terminals with Heterogeneous Network Access

Diego Reforgiato, Alfio Lombardo, Giovanni Schembra

Dipartimento di Ingegneria Informatica e delle Telecomunicazioni

University of Catania

V.le A. Doria 6, 95125 Catania - ITALY

{diegoref, lombardo, schembra}@diit.unict.it

## ABSTRACT

<sup>1</sup> P2P technology gives new opportunities to define an efficient multimedia streaming application, and, at the same time, it involves a set of technical challenges and issues due to the best-effort service offered by the Internet and its dynamic and heterogeneous nature.

The idea of this paper is to propose a multipoint video broadcast framework over a heterogeneous content distribution P2P network. In the proposed system the source generates the video flow by using an MPEG-4/FGS encoder, in such a way that no losses occur at the Base-layer stream even in the presence of short-term bandwidth fluctuations.

The FGS layer is sent together with the Base layer, but with a lower priority. The source uses a rate controller to regulate the encoding rate of the Base layer. To this aim, a protocol is defined in order to provide the source with information related to the most stringent bottleneck link on the overlay network. A technique to reorganize the content distribution tree is proposed and discussed. To evaluate the performance of the proposed framework a case study is introduced; improvements obtained with respect to several reference cases where FGS is not applied are also shown.

## Keywords

Adaptive video broadcast; P2P; Multipoint Video Communication; Fine Granularity Scalability (FGS); PSNR.

## 1. INTRODUCTION

In the last few years broadcast video transmission over the Internet has become increasingly popular [15]. The main problem of such a kind of systems is to realize multipoint communication on the current Internet that does not natively support it. In the first generation of

<sup>1</sup>This work is partially supported by the Italian MIUR PRIN 2007 project "Sorpasso". Moreover, the work leading to this invention has benefited from a fellowship of the Seventh Framework Programme of the European Community [7° PQ/2007-2013] regarding the Grant Agreement n. PIRG03-GA-2008-231021.

such systems the most widely used approach has been the employment of multiple parallel unicast streaming. According to this approach, multimedia data are transmitted by the source to each receiver in a point-to-point fashion: whenever a new client accesses the service, a dedicated stream is allocated until the end of its connection. However this approach requires a separate streaming bandwidth for each client, thus the multimedia source and the consumed network bandwidth resources inevitably grow linearly with the user population. Therefore this approach is not scalable with the number of users, and thus unfeasible in video broadcast scenarios, where the number of users is very high.

The first solution to the scalability problem was the application of IP network multicasting. In this way, unlike multiple parallel unicast transmission, a multicast multimedia stream can be shared by more than one receiver. The network switches/routers automatically replicate the multicast video for multiple receivers without adding any extra streaming workload on the multimedia server. However, IP multicast is not widely deployed, mainly due to practical and political issues. For example, multicast is not available in many low-cost Internet access points, like domestic ADSL.

In the meanwhile, Peer-to-Peer (P2P) systems have gained ground for content sharing between communities, determining a real revolution on the Internet. Unlike traditional distributed systems, P2P networks are self-organizing networks that aggregate large numbers of heterogeneous computers called nodes or peers (nodes and peers will be used interchangeably in this paper). In P2P systems, peers can communicate directly with each other for data sharing and data exchanging. Peers also share their communication and storage resources.

The characteristics of P2P systems make them a very good choice for video broadcast over IP networks [19]. According to the P2P approach, peers interested in the same video transmission organize themselves into an application-layer multicast tree. A peer in the application layer multicast tree receives video packets from

its parent, then duplicates and forwards them to its children [3, 6]. Peers in the overlay network are cooperative in the sense that they share data and exchange group control information. A multitude of protocols have been proposed in previous literature to manage tree-structured overlay networks for video broadcast [11, 4, 7, 16].

However, although P2P technology gives new opportunities to define an efficient multimedia streaming application, at the same time, it involves a set of technical challenges and issues due to the best-effort and dynamic nature of the service offered by the Internet, and the heterogeneity of terminals used by clients to access the service. First of all, although peers access the network through different access links with totally different bandwidth characteristics, often dominated by the uplink connections, P2P communication infrastructure is often built upon an overlay network whose topology does not depend on the underlying physical network. Second, network bandwidth, delay and loss behaviors rapidly change in time. Last, but not least, a large number of different terminals are becoming very popular, ranging from powerful high-performance home receivers to mobile handheld video devices; they present different capabilities and requirements in receiving and decoding video content.

Keeping all this in mind, the idea of this paper is to apply fine granularity scalability (FGS) encoding [13, 10, 14, 5, 20, 18] for video transmission. FGS is an evolution of the scalable hierarchical video encoding; it was defined some years ago to deliver multimedia applications in heterogeneous network environments with different bandwidth and loss behaviors.

An FGS stream has only two layers: a Base layer that must be received to make possible video decoding, and an enhancement layer, henceforward indicated as the FGS layer which can be delivered optionally where bandwidth is available.

FGS allows the source to adjust the relative sizes of both Base and FGS layers, therefore allowing the FGS layer to be broken up and allowing the decoder to decode any portion of the received FGS layer. The source or any intermediate node are responsible to do that.

With such a process, peers connected to other peers through bottleneck links guarantee an efficient transmission of the Base layer, discarding only portions of the FGS layer.

The authors of this paper think that, although not widely applied in the past due to its encoding complexity, today, thanks to the evolution in hardware and software technologies, the FGS appears a good and feasible solution for multipoint multimedia broadcast systems for the immediate future. Besides, [17] is an invention of 2006 that discloses methods, devices and systems for effective fine granularity scalability coding and decod-

ing of video data.

In fact, the MPEG-4 FGS encoder developed by the authors is able to encode and decode video in real-time. The encoder has been implemented in Visual C++ using the Intel<sup>®</sup> Integrated Performance Primitives (Intel<sup>®</sup> IPP) [2], an extensive library of multi-core-ready, highly optimized software functions for multimedia data processing, and communications applications. Running on a start-level dual-core personal computer equipped with 2GB of RAM, it is able to encode about 100 fps and decode about 400 fps for CIF video streams.

In the proposed platform, the video source uses a rate controller to regulate the encoding rate of the Base layer. A protocol is defined in order to provide the source with the necessary information related to the bandwidth of the most stringent bottleneck link. The framework works on a tree-structured P2P network, and any tree construction and management protocols [11, 4, 7, 16] can be adopted.

A case study is introduced to evaluate both the performance of the proposed framework and the improvements obtained with respect to a reference case in which FGS is not applied.

The paper is structured as follows. Section 2 provides a brief description of FGS. Section 3 describes the system overview. Section 4 analyzes a case study. Specifically, performance are calculated analyzing video at the receiving side, accounting frame loss and encoding PSRN simultaneously. Performance comparison with other platforms is made in order to evaluate the improvements introduced by that proposed in this paper. Finally, Section 5 concludes the paper.

## 2. FGS: OVERVIEW AND SOME STATISTICS

In this section we will provide a brief overview of the main characteristics of the FGS encoding technique in order to facilitate the understanding of the remainder sections of the paper. For a more detailed description of FGS, the reader is referred to [13, 10, 14].

FGS is a scalable encoding technique. Generally, there are three types of scalability, i.e., temporal scalability, spatial scalability and SNR scalability. In all the three cases, the Base-layer pictures are encoded based on sub-sampling with either less frame rate (for temporal scalability), or smaller picture size (for spatial scalability), or coarser picture quality (for SNR scalability). Full-quality video is obtained by the combination of both Base and FGS layers.

FGS is an evolution of the scalable hierarchical video encoding. It emerged to deliver multimedia applications in heterogeneous network environments with different bandwidth and loss behavior. It was defined in [10] with the main target of achieving a good balance between

coding efficiency and scalability.

Its encoding is designed to cover any desired bandwidth range while maintaining a very simple scalability structure. The basic idea of FGS is to encode a video sequence into two layers only, a Base layer and an enhancement layer, in the following called the FGS layer. A MPEG-4 encoding is used: the Base layer is obtained with a classical MPEG-4 encoder using non-scalable coding, whereas the FGS layer is coded using a fine-granular scheme. The latter encodes the difference between the original picture and the reconstructed one with the use of bit-plane coding of the DCT coefficients.

The idea of this paper is to apply MPEG-4 FGS encoding for video transmission, which allows intermediate nodes to truncate the bit stream of each frame at any point, thus only degrading the quality proportionally to the current available bandwidth. The Base layer of the MPEG-4 stream is generated in such a way that its encoding bandwidth is a little lower than the current minimum bandwidth in the tree. Therefore losses are negligible at the Base-layer stream even in the presence of short-term bandwidth fluctuations (specifically, in our system the Base layer is generated frame-by-frame at 90% of the minimum link bandwidth in the whole network). The FGS layer is sent together with the Base layer, but with a lower priority. By so doing, peers connected to other peers through bottleneck links guarantee the transmission of the Base layer, discarding only portions of the FGS layer.

### 3. SYSTEM DESCRIPTION

The system we propose in this paper is a live video broadcast platform where a video source distributes a video stream to a number of clients in a multipoint fashion. Multipoint communication is achieved by applying a P2P approach, configuring a tree-structured overlay network where the root is the video source, while the other clients, henceforward referred to as peers, are internal nodes or leaves.

In real scenarios bandwidth apparently available to a peer (e.g. the ADSL access of its domestic connection) might be not true, and the peer actually can make use only of a small fraction of it, for example due to the intensive use of file sharing programs or bandwidth sharing with other users in the same LAN. For this reason, in order to model the bandwidth variations in a realistic manner, we realized a bandwidth generation simulator at the overlay network level. In order to avoid excessively strong oscillations of both the encoding quality and the system behavior, the bandwidth values are first smoothed with an exponentially-weighted moving average (EWMA) filter with parameter  $\beta^{(RC)}$ , defined as follows:

$$\hat{B}_n = \beta^{(RC)} \cdot \hat{B}_{n-1} + (1 - \beta^{(RC)}) \cdot B_n \quad (1)$$

An important component in the video source architecture is the *Topology Manager*, which decides and maintains the tree network topology by deciding the position of each peer within the tree.

In order to have our protocol working efficiently by applying FGS, it is obvious that a node with a low uplink bandwidth behaves as a bottleneck; if it were located in the upper part of the tree it would penalize all its descendants. Thus, the best peers should be located at the top of the tree. For this reason, and taking into account that link bandwidth varies in time, the Topology Manager has to optimize the tree topology runtime, in order to avoid configurations with bottlenecks at the highest levels of the tree.

We will assume that the downlink bandwidth of each peer is higher than the corresponding uplink bandwidth. In fact:

1. there are many scenarios (e.g. ADSL) where the uplink bandwidth is lower than downlink bandwidth;
2. the uplink bandwidth is shared with tree nodes whereas the downlink bandwidth of each peer is used as a whole to connect them with their parent only.

The Topology Manager obtains the bandwidth estimations every  $T_P$  seconds ( $T_P = 10$  sec in our implemented system). Then, every  $T_T$  seconds ( $T_T = 30$  in our implementation) it updates the tree topology as follows: it chooses the  $F$  peers with the highest uplink bandwidth, and connects them as children of the source. Then, for each of these peers, the same operation is repeated, choosing the next  $F$  peers with the highest uplink bandwidths. This algorithm is run recursively until all the peers get a position in the tree. To optimize the bandwidth performance and to further balance the tree, the nodes in the last level (the leaves) are evenly distributed among the peers in the upper level of the tree (the parents nodes) in a round robin fashion; this avoids the scenario where some peer is overloaded with its uplink bandwidth whereas others are not.

Let us note that changes in the topology structure can be deleterious for video decoding of peers; changing their position in the tree can cause sudden changes in delays, thus increasing the delay jitter as well. In addition, let us observe that this does not happen only when a peer changes level in the tree, but also if either it changes its parent or one of its predecessors changes level or predecessor. However, two important observations can be made in favor of such a strategy:

1. since the bandwidth values are passed under a EWMA filter this strategy is not affected by occasional bandwidth changes. Changes in the tree topology occur in the event of serious and lasting

bandwidth changes; in this case changes are likely to optimize performance;

2. the delay jitter caused by topology changes can be recovered at destination through the application of intelligent compensation buffers and adaptive media play-out techniques (see for example [12, 9]).

## 4. CASE STUDY

In this section we define a case study to analyze the performance of the proposed system. The target is to demonstrate the gain achieved by applying FGS encoding against classical video streaming over P2P, in terms of the peak signal-to-noise ratio (PSNR) evaluated on the video flow received at destination from each peer. More specifically, Section 4.1 introduces our case study; Section 4.2 analyzes some statistics of the considered sequence; Section 4.3 contains both the results of the overlay network behavior analysis and the performance analysis at the application level. Finally, Section 4.4 shows the comparison of the platform we propose in this paper against other approaches currently used.

### 4.1 Case Study Description

According to what we said in the previous section, we consider a video distribution platform with centralized control by the Topology Manager.

We will assume that all the peers, included the source, have set the same fan-out parameter,  $F$ , representing the maximum number of peers that can be attached as children in the distribution tree.

We will carry out a steady-state analysis, assuming that the number of peers in the network, hereafter referred to as  $N$ , remains constant for the whole duration of the simulation. In this way our study does not depend on the particular algorithm used to manage the topology structure when peer arrivals or departures occur. Therefore, the only job of the Topology Manager is to re-arrange the tree according to bandwidth variations.

Peers are grouped in  $C = 5$  different classes, each characterized by different Internet access link performances, and different average values of the uplink bandwidth. As previously explained, the downlink bandwidth of each peer is assumed to be much higher than the corresponding uplink bandwidth. Therefore, along this section we will refer to the uplink bandwidth as the *bandwidth*, unless explicitly mentioned. Classes are organized with decreasing average bandwidth values: e.g., peers in the third class have higher bandwidth values than peers in the fourth class. We have assumed that the source is a high bandwidth server with an uplink of 5 Mbit/s.

## 4.2 Statistical Analysis of the considered sequence

For all the experiments along the paper we have considered a sequence of the video “BBC Planet Earth documentary”. The sequence has a duration of 20 minutes and is encoded with a  $176 \times 144$  QCIF format, at a frame rate of 25 frame/sec, and using as Group of Pictures (GoP) structure with the pattern IBBPBB. It is misleading considering the peak signal-to-noise ratio (PSNR) parameter at the source only, i.e., for all the frames encoded by the source, since the quality perceived at destination is strongly degraded if network losses occur. On the other hand, it is misleading as well averaging PSNR over only the non-corrupted received frames because in this case the estimation would result not fair (for example because low-quality frames are small, and therefore more likely to be received). For this reason, in order to take into account the encoding quality and the frame rate reduction simultaneously, we have used a *Quality parameter*, defined in [8] through a heuristic formula, which models the overall video quality at destination as a function of both the mean PSNR in dB of the received frames and the frame rate. More specifically, the *Quality parameter* is defined as follows:

$$Q = 0.45 \cdot psnr + (fr - 5)/10 - 16.9 \quad (2)$$

## 4.3 Numerical Results

This section contains numerical results concerning the overlay network behavior analysis (Section 4.3.1), and the performance analysis at the application level (Section 4.3.2).

### 4.3.1 Overlay Network Behavior Analysis

Here we will show results on some peers randomly chosen as representative of their corresponding class. Their uplink bandwidth have been generated using the tool in [1].

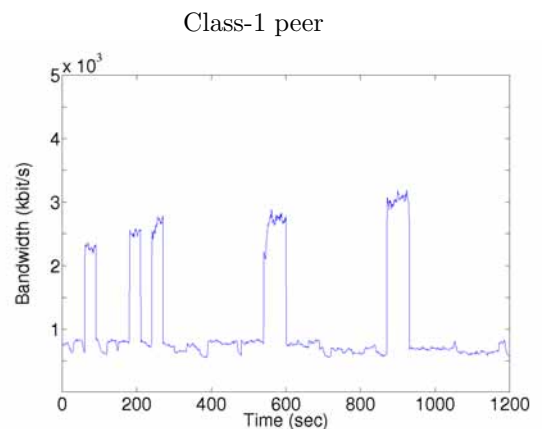
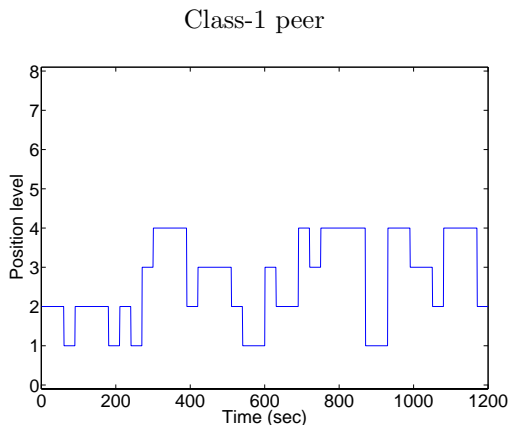


Figure 1: Generated uplink bandwidth for a peer of class 1 of the tree.

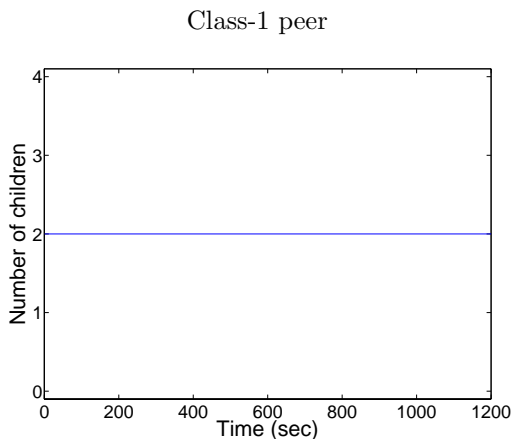
Class	PSNR			Q		
	F=2	F=3	F=4	F=2	F=3	F=4
1	36.2427	36.3263	34.5218	0.409232	0.446816	-0.365208
2	35.9045	36.0584	34.1065	0.257003	0.326285	-0.552064
3	35.8657	36.0403	33.9856	0.239575	0.318134	-0.60648
4	35.8563	36.0343	33.9228	0.235342	0.315454	-0.634722
5	34.4134	34.4915	32.3582	-0.413992	-0.378809	-1.33883

**Table 1: Average PSNR and  $Q$  for representative peers in each class.**

We have fixed the number of peers to 85, including the source, and analyzed the performance for three different values of the fan-out parameter,  $F \in \{2, 3, 4\}$ . The 84 peers (the source has been considered as a high bandwidth server with 5 Mbit/s uplink) have been randomly chosen from five different classes of peers (peers in different classes have different ranges of bandwidth values according to a uniform distribution).



**Figure 2: Time behavior of peer position in the tree of a peer of class 1 for  $F = 2$ .**



**Figure 3: Time behavior of the number of children in the tree of a peer of class 1 for  $F = 2$ .**

We have generated traces for a time interval of 20 minutes, equal to the length of the considered movie trace. The last level of the obtained trees is partially filled for  $F \in \{2, 3\}$ , whereas it is completely filled for  $F = 4$ .

We first show in Figs. 1, 2 and 3 how the system works, analyzing the behavior of one representative peer (for space constraint we will show the behavior of just one peer of class 1 for  $F = 2$ ) taken as representative of its corresponding class.

More specifically, Fig. 1 shows its generated bandwidth over time; Fig. 2 shows how the peer changes level during its lifetime according to the behavior of the instantaneous uplink bandwidth from the source, and therefore it is strongly influenced by the class it belongs to.

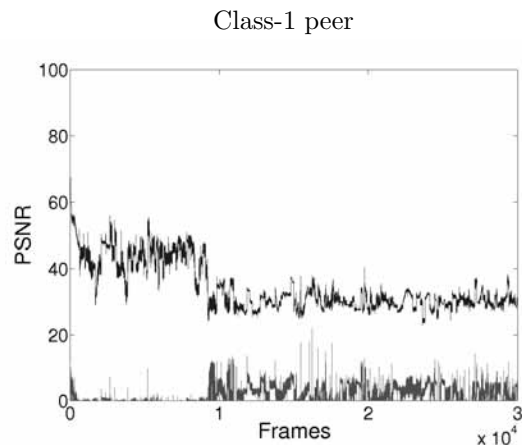
Fig. 3 shows the time behavior of the number of children for the considered peer. The peer of class 1 has always 2 children.

The reader notes that Fig. 1, 2 and 3 are strictly correlated. Peers with low bandwidth values get lower positions in the tree, whereas peers with higher bandwidth values lie in the upper levels. For example, around the time instant 200 secs, the peer of class 1 has a bandwidth which rapidly increases (see Fig. 1) and, consequently, it is immediately moved from level 2 to level 1.

#### 4.3.2 Performance Analysis at the Application Level

In this section we will show some statistics concerning the quality of the video received from the source by the peers we are considering as representative of each class.

Table 1 shows the average PSNR and the average Quality parameter  $Q$  defined in Section 4.2 (Equation 2), measured at destination on the overall sequence by peers of each class for different values of  $F$ . Figure 4 shows



**Figure 4: PSNR for the Base layer (black) and gain achieved using FGS (gray) for a peer of class 1 with  $F = 2$ .**

the PSNR per frame for the videos received by the representative peer of class 1 when both Base and FGS layers are transmitted (in black), and the PSNR difference between the video transmitted with both Base and FGS layers and the video transmitted with just the Base layer (in gray). When both Base and FGS layers are used, the PSNR is always higher, or, at the most, equal (difference equal to 0) to each other.

For each frame of the used GoP (see Section 4.2), we have also computed its loss percentage (shown in Figure 5 only for  $F = 2$  for space constraint). The reader notes that I-frames have the priority to be transmitted with respect to the other GOP frames because an I-frame loss causes 7 more frame losses. It is for this reason that, in our system, it is more difficult to incur in I-frame losses (i.e. charts in Figure 5 do not show any loss of I-frames). The displayed chart has the peer class as x-axis, and the frame loss percentage as y-axis. For each value in the x-axis there are five columns related to the six GOP frames (since the one related to I-frame is always 0). Moreover, one more column ( $T$ ) for each value shows the average frame loss percentage.

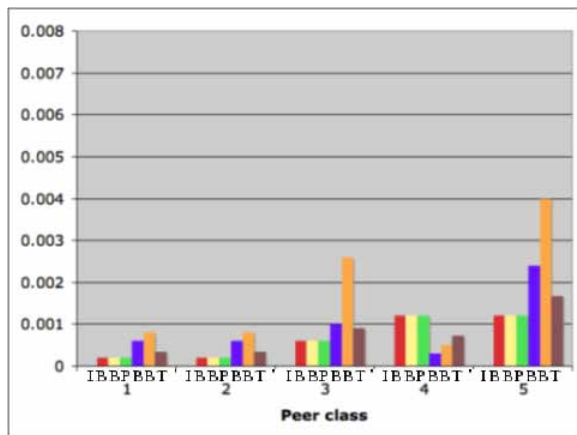


Figure 5: Frame loss percentage for five representative peers of each class for  $F = 2$ .

#### 4.4 FGS Performance Assessment

In this section we will show how our video transmission system greatly improves on scenarios where either the tree is created but is not updated according to peer uplink bandwidth variations (and therefore it is static), or the enhancement layer for video transmission is not sent at all (i.e. FGS encoding is not applied). More specifically, in the following we will consider for comparison the four combinations of the above situations that can be classified and called as follows:

1. *Dynamic + Base/FGS* (DF): a peer-to-peer network (with  $F \in \{2, 3, 4\}$ ) where peers are organized with a tree topology which changes dynamically using both Base and FGS layers for video

transmission; this is the novel method we propose in this paper;

2. *Dynamic + Base* (DB): a peer-to-peer network (with  $F \in \{2, 3, 4\}$ ) where peers are organized with a tree topology which changes dynamically; in such a case each peer receives the Base layer only of the whole movie;
3. *Static + Base/FGS* (SF): a peer-to-peer network where peers are organized in a static tree (peer position is fixed in the tree for the whole transmission), and the source uses both Base and FGS layers for video transmission;
4. *Static + Base* (SB): a peer-to-peer network where peers are organized in a static tree (peer position is fixed in the tree for the whole transmission); in such a case each peer receives the Base layer only of the whole movie.

For such scenarios, for each class of peers and values of  $F \in \{2, 3, 4\}$ , we show the PSNR, the  $Q$  value and the frame loss percentage.

We have already shown the above parameters for the novel method we propose in this paper (i.e. *Dynamic + Base/FGS*) in Table 1 and Figure 5. As far as the frame loss percentage is concerned, let us notice that only the Base-layer frames contribute to its computation; therefore the methods *Dynamic + Base/FGS* and *Dynamic + Base* produce the same results (shown in Figure 5) for them. However, the PSNR and  $Q$  values for *Dynamic + Base* method are different since the FGS layer gives more details in terms of video quality. Table 2 shows such values. As expected, the PSNR and  $Q$  values for each combination of  $F$  and peer class are worse than the corresponding PSNR and  $Q$  values achieved by applying the *Dynamic + Base/FGS* method, and shown in Table 1. The PSNR and  $Q$  values produced by *Static*

Class	PSNR			$Q$		
	F=2	F=3	F=4	F=2	F=3	F=4
1-5	34.186	34.108	32.3634	-0.411642	-0.371378	-1.33648

Table 2: Average PSNR and  $Q$  for representative peers in each class - *Dynamic + Base* method.

+ *Base/FGS* and *Static + Base* methods are shown in Table 3 and Table 4, respectively. Similar considerations taken for the Dynamics methods hold: as shown in Figure 6 they produce the same results for the frame losses.

As seen above for the Dynamics methods, the PSNR and the  $Q$  values for the *Static + Base/FGS* method are better than those produced by the *Static + Base* method.

Class	PSNR			Q		
	F=2	F=3	F=4	F=2	F=3	F=4
1	35.3316	35.4134	34.1984	0.3764	0.4189	-0.3744
2	35.2096	35.3491	34.1605	0.21363	0.30123	-0.4546
3	34.8912	34.773	33.6131	0.2111	0.2918	-0.5101
4	34.1513	34.0378	33.0019	0.2075	0.2744	-0.5321
5	33.6781	33.7644	32.5137	0.1012	0.1791	-0.559

**Table 3: Average PSNR and  $Q$  for peers in each class for *Static + Base/FGS* method with  $F \in \{2, 3, 4\}$ .**

Class	PSNR			Q		
	F=2	F=3	F=4	F=2	F=3	F=4
1-5	30.6046	30.675	30.2024	-2.12792	-2.09624	-2.23894

**Table 4: Average PSNR and  $Q$  for representative peers in each class - *Static + Base* method.**

Finally, we have produced the Fig. 7 showing global results of each of the four methods above discussed for each of the variables of interest (PSNR, frame losses and  $Q$  value), and for  $F = 2$ . Each value is calculated as the average of each performance parameter over all the peers in the network. It is evident how the PSNR and the  $Q$  values are much better for the proposed method (i.e. *Dynamic + Base/FGS*).

## 5. CONCLUSIONS AND FUTURE WORK

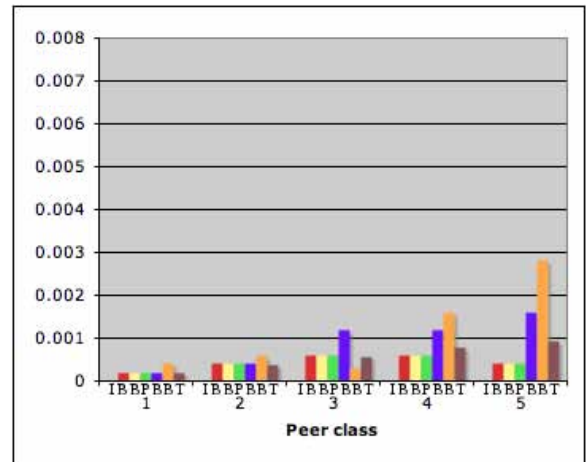
This paper proposes a multipoint video transmission framework over a heterogeneous content distribution P2P network. The source generates the video flow by using an MPEG-4/FGS encoder, in such a way that the number of losses occurring at the Base-layer stream are minimized, even in the presence of short-term bandwidth fluctuations.

The FGS layer is sent together with the Base layer, but with a lower priority. The source uses a rate controller to regulate the encoding rate of the Base layer, according to an estimation of the bandwidth of the overlay network bottleneck link. A protocol is defined in order to provide the source with this information.

A case study is introduced to evaluate the performance of the proposed framework and the improvements obtained with respect to some reference cases in which FGS is not applied and/or the overlay network topology is not dynamically re-organized.

The problem of organizing the tree in real-time is discussed and some techniques are proposed and compared.

As future work we plan to investigate the possibility to apply the proposed scheme for 3D-video streaming on several application scenarios: *Three-Dimensional Television (3D-TV)*, *Free ViewPoint Television (FTV)*, and *Multi-View Video Coding (MVC)*. Last but not least, in



**Figure 6: Frame loss percentage for the five representative peers - *Static + Base/FGS* and *Static + Base* methods for  $F = 2$ .**

presence of high churn rate, the management of transitory peers will be investigated in order to maintain the architecture as more robust as possible.

## 6. REFERENCES

- [1] Bandwidth generator tool. Available at <http://www.diit.unict.it/arti/Tools/BandwidthSim.zip>.
- [2] Intel(r) integrated performance primitives (intel(r) ipp), available at <http://www.intel.com/cd/software/products/asm-na/eng/302910.htm>.
- [3] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. August 2001.
- [4] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and Atul Singh. Splitstream: High bandwidth multicast in cooperative environments. 2003. Bolton Landing, NY, USA.
- [5] Kihwan Choi, Kwanho Kim, and Massoud Pedram. Energy-aware mpeg-4 fgs streaming. In *DAC '03: Proceedings of the 40th conference on Design automation*, pages 912–915, New York, NY, USA, 2003. ACM.
- [6] Y.H. Chu, S.G. Rao, and H. Zhang. A case for end system multicast. June 2000.
- [7] T.T Do, K.A. Hua, and Tantaoui M.A. P2vod: providing fault tolerant video-on-demand streaming in peer-to-peer environment. 3:1467–1472, June 2004.
- [8] G. Hauske et al. Subjective image quality of low-rate and low-resolution video sequences. October 5-8 2003.
- [9] M. Kalman, E. Steinbach, and B. Girod. Adaptive media playout for low-delay video streaming over error-prone channels. 14(6):841–851, June 2004.

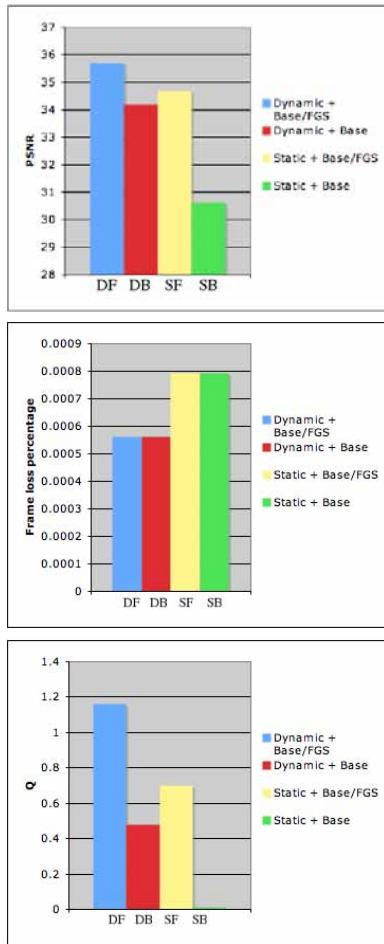


Figure 7: Global results for each method for  $F = 2$ .

[10] W. Li. Overview of fine granularity scalability in mpeg-4 video standard. 11(3):301–317, March 2001.

[11] V.N. Padmanaban, H. J. Wang, and P.A. Chou. Resilient peer-to-peer streaming. November 2003. Atlanta, GS, USA.

[12] G. Platt and J.Y. Khan. Adaptive jitter compensation on the downlink of future mobile multimedia communications systems. May 2003.

[13] H. Radha and Y. Chen. Fine-granular-scalable video for packet networks. April 1999. New York, NY.

[14] H. Radha, M. van der Schaar, and Y. Chen. The mpeg-4 fine-grained scalable video coding method for multimedia streaming over ip. 3(1), March 2001.

[15] S.D. Sevetto and K. Nahrstedt. Broadcast quality video over ip. 3(1), March 2001.

[16] D.A. Tran, K.A. Hua, and T.T. Do. A peer-to-peer architecture for media streaming. pages 121–133, January 2004.

[17] Wang and Ye-Kui. Method, device and system for effective fine granularity scalability (fgs) coding and decoding of video data. In *International Application N. PCT/IB2006/000631, International Filing Data 22.03.2006*, 2006.

[18] F. Wu, S. Li, and Y. Zhang. A framework for efficient fine granularity scalable video coding. 11(3):332–344, March 2001.

[19] Dongyan Xu, Mohamed Hefeeda, Susanne Hambrusch, and Bharat Bhargava. On peer-to-peer media streaming. July 2002. Vienna, Austria.

[20] Jian Zhou, Jian Zhou, Huai rong Shao, Huai rong Shao, Chia Shen, Chia Shen, Ming ting Sun, and Ming ting Sun. Multi-path transport of fgs video. In *in Proc. Packet Video Workshop*, 2003.