

# Streaming MPEG-4 AudioVisual Objects Using TCP-Friendly Rate Control and Unequal Error Protection

Toufik Ahmed<sup>1</sup>, Ahmed Mehaoua<sup>1</sup> and Vincent Lecuire<sup>2</sup>

<sup>1</sup> CNRS-PRISM Lab. University of Versailles  
45 av. des Etats-Unis,  
F-78035, Versailles, France

<sup>2</sup> CRAN lab. – CNRS UMR 7039. University Henri Poincaré of Nancy 1,  
Faculté des Sciences – BP 239,  
F-54506 Vandoeuvre-lès-Nancy Cedex, France

**Abstract** — This article describes a fair and robust video streaming framework over IP networks. It is based on an MPEG4 Audio-Visual Object (AVOs) classification, TCP-Friendly transport and out-of-band unequal forward error protection. According to network congestion feedback, video source servers dynamically adjust their bit rates by adding and dropping MPEG-4 AVO to conform to the TCP-Friendly Rate Control (TFRC) algorithm and by taking into consideration media semantic relevancy. Thus, an accurate MPEG-4 Access Unit (AU) partitioning and packetization can be performed to cope with decoding error propagation and network bandwidth fluctuation. Finally, AVOs requiring similar network QoS level are automatically classified, packetized and mapped to one of the available IP DiffServ PHB (Per Hop Behaviors). Simulation results show a significant improvement regarding to user-perceived video quality, packet loss recovery and bandwidth share fairness.

**Keywords:** Adaptive video streaming, IP, TCP-Friendly, QoS, FEC.

## I. INTRODUCTION

Video streaming over the Internet is becoming very popular and it is competing with traditional TCP based applications for bandwidth utilization. As a result, Network stability and traffic fairness become critical issues. On the other hand, next generation Internet will be characterized by Quality of Service (QoS) capabilities. It is commonly accepted that IP Differentiated Services (DiffServ) will be highly deployed in the next generation IP networks. Recent researches on transmitting video over IP demonstrate that DiffServ is a strong candidate for supporting real-time video communications. Works presented in [1], [2], [3], and [4] clearly state that IP Diffserv is the most suitable model for delivering interactive and streamed video content over Internet at a large scale. The QoS-capability cannot be achieved efficiently without mechanisms to ensure a fair share of network resource between real-time (UDP-based) and non-real time (TCP-based) IP services. Those mechanisms are known as TCP-friendly transport protocols. In this paper, we propose an integrated transmission architecture that efficiently combines FEC and TCP-Friendly mechanisms to guarantee both, a high visual quality level of the played-out MPEG-4 Audio-Visual streams and a fair share of the bandwidth.

The control of the quality of the video service is performed at the video streaming sources through three schemes First, an adaptive MPEG-4 Access Unit (AU) partitioning and packetization protocol that classify MPEG4 AudioVisual Objects (AVO) according to their importance for the video scene. Second, video servers perform an unequal and out-of-band forward error protection to sensitive Access Units to deal with IP packet loss and error propagation. Third, servers adjust their transmission rate based on network congestion control information, accordingly to the TCP-Friendly Rate Control scheme (TFRC). This source bit rate adjustment leads to a fair share of network resources with other TCP/IP connections. The video servers tag and stream the most relevant IP packets embedding AVO data according to their relevancy to the service (i.e., low or high drop precedence). Less important AVO are transmitted only if bandwidth availability. Consequently, this integrated video streaming architecture provides a significant improvement for the control of the end-to-end QoS. Performance evaluation is carried out through simulation.

The remainder of this paper is as follows: Section II presents an overview of MPEG-4 framework and the IP video streaming framework. Section III describes the proposed IP video streaming architecture. Section IV focuses on the AVO protection scheme using Unequal Error Protection based on audiovisual relevancy. Performance evaluation and analysis is presented in section V. We finally conclude in Section VI.

## II. MPEG-4 AVO STREAMING OVER IP

### A. MPEG-4 Object based Coding

Basically, an MPEG-4 scene consists of one or more audio visual objects called AVOs. Each of them is characterized by temporal and spatial information. Each Video Object (VO) may be encoded in a scalable (multi-layer) or non scalable (single layer) form. A layer is composed of a sequence of a Group of Video-Object-Plane (GOV). A Video Object Plane (VOP) is similar to the MPEG-2 frame. VOP supports intra coded (I-VOP) temporally predicted (P-VOP) and bi directionally predicted (B-VOP) [5].

To take benefits from the object-based compression, we have proposed in [6] an intelligent adaptation to cope with network congestion and end terminal heterogeneity. We proposed to classify MPEG-4 AVOs at the video server from most important AVO to least important ones. Several methods have been used for objects classification. During scene creation, one can affect the adequate priorities to each object in the scene. For scenes with no assigned object priorities, MPEG-4 object descriptors or MPEG-7 QoS descriptors metadata have been used to provide the relevant information needed to handle object priority.

The classification process is very important in order to apply adaptive video streaming and unequal error protection to different video streams. AVO requiring different level of QoS and handling traffic prioritization in the network level are automatically classified and mapped to one of the IP DiffServ PHB supported by the IP network. Details on the classification process and model are out of the scope of this article. Readers can refer to [6] for more information.

### B. MPEG-4 System Layer

The MPEG 4 video coding provides an object-based representation of video by allowing the coding of audio-visual object. Each audio-visual object is coded separately. Texture and shape coding in MPEG-4 are very similar to the coding of frames in MPEG-2. In a particular AVO, the different parts of the video data stream have not the same importance for the quality of the decoded video. The damages caused by some data loss in a reference picture (I-VOP or P-VOP) will affect subsequent picture(s) due to inter-frame predictions. Subsequently, I-Frame must be protected more than P-Frame and P-Frame more than B-Frame. Let us consider now the example of video object coded with layered wavelet transform techniques. The most important layer contains the low frequency sub-band of the picture, called Base Layer (BL). Other layers, which represent a hierarchical level of resolution of the wavelet transform, are less important. These layers are called Enhancement Layers (EL).

This is the second step of preparing AU to be transmitted over the network. It operates within a single audio-visual object. The first step is handled by the classification layer which classifies the object among them. As we said, the result of the classification is a set of AVOs sorted according to their importance in the scene. We affect a final priority to each AU to apply an unequal error protection. This priority reflects both the priority of particular AVO in the scene and the priority of a single frame type (I, P, B or hierarchical stream if any BL or EL). We chose a scale of 100 levels of priority ( $p \in [0,99]$ ), the higher value being associated to the most important AU) that can be applicable to a large number of media encoding scheme.

### C. IP Video Streaming Control Schemes

By classifying MPEG-4 AVO, we provide a first level of scalability called object scalability. It gives the server the ability to add and drop video objects dynamically and deal with network

congestion intelligently. In this paper we propose an integrated architecture for unicast video streaming using the following mechanisms: (1) An AVO classification mechanism to generate MPEG4 access units (2) A mechanism for adding and dropping AVO according to network congestion and TCP-friendly rate control, that is performed by the server to adjust transmission rate while being fair to other network traffic. (3) An out-of-band and unequal FEC signaling to minimize packet loss impact on video quality.

These mechanisms are integrated and collaborate with each others to guarantee a high level of protection during video transmission and network congestion periods.

### III. ADAPTIVE AVO STREAMING USING TCP-FRIENDLY

The video quality adaptation mechanism is based on TFRC [7]. It operates as follows: The receiver measures the loss rate and feeds this information back to the sender. This is achieved by a modified version of RTP (Real-time Transport Protocol) and RTCP (Real-time Transport Control Protocol). Each RTP packet has a timestamp and a sequence number that allow the receiver to compute the packet loss and the sender to compute the RTT. The loss rate and the RTT are then fed into the TFRC module to get the appropriate transmission rate (cf. Eq. 1 below). The sender then adds or drops audio-visual objects and the associated layers if any, to adjust its transmission rate to match the target rate (i.e., allowed rate). The calculated rate is obtained by using the TFRC equation [8]:

$$R_{TCP} \cong \frac{s}{RTT \sqrt{\frac{2bp}{3}} + t_{RTO} (3 \sqrt{\frac{3bp}{8}}) p (1 + 32p^2)} \quad (\text{Eq.1})$$

Where RTCP is the target transmission rate or the allowed transmission rate,  $s$  is the packet size, RTT is the round trip time,  $p$  is the loss rate,  $t_{RTO}$  is the TCP retransmission timeout value and  $b$  is the number of packets acknowledged by a single TCP acknowledgement.

#### A. Add / Drop of Audio-Visual Objects

Let  $S$  be a set of MPEG-4 AVOs containing  $n$  AVOs  $O_j$ , with  $j \in \{1, 2, \dots, n\}$ . Without loss of generality, we assume that these objects are sorted in a decreasing order of priority using our Classification process. Each object  $O_j$  may consist of  $m_j$  layers ( $m_j \geq 1$ ). Note that lower layers within an object have higher priorities than higher layers.

Let  $P$  be the function that returns the priority of a particular object or layer. Without loss of generality, we assume that:

$$\begin{aligned} \forall j, 1 \leq j < n: P(O_{j+1}) \leq P(O_j) \\ \forall j, 1 \leq j < n, \forall l, 1 \leq l < m_j: P(L_{j,l+1}) < P(L_{j,l}) \end{aligned} \quad (\text{Eq.2})$$

$L_{j,l}$  is the Layer number  $l$  of the Object  $O_j$

By using formula (2) we can construct an Audio-Visual Entity set called  $E$  composed of all object layers ordered by their priorities.  $E = \{L_{1,1}, L_{1,2}, \dots, L_{1,m_1}, L_{2,1}, L_{2,2}, \dots, L_{2,m_2}, \dots, L_{n,1}, L_{n,2}, \dots, L_{n,m_n}\}$ . We will

note  $E$  as follows:  $E = \{e_1, e_2, \dots, e_w\}$  with  $w = |E| = \sum_{j=1}^n m_j$ .

The **server adds** a new audio-visual entity as soon as the target rate exceeds the current sending rate of current entities plus the new entity. Assume that the server is streaming  $k$  entities at time  $t_i$ . We assume also that the client has sufficient resources to play all the entities being sent by the server. Therefore, at time  $t_{i+1}$  the server can add a new entity while the following condition (3) is satisfied:

$$\sum_{j=1}^{k+1} R_{i+1}(e_j) \leq R_{TCP} \quad (\text{Eq.3})$$

With the same manner, when the estimated throughput of the TCP session indicates that the video server is transmitting more data than it should, then the video server must reduce its sending rate by dropping one or more audio-visual entities. Therefore, the **server drops** entities while the following condition (4) is satisfied:

$$\sum_{j=1}^k R_{i+1}(e_j) > R_{TCP} \quad (\text{Eq.4})$$

### IV. AVO PROTECTION USING FEC

Error resilience of each *Elementary Stream* associated to one AVO can be enhanced when the sensitive data is protected whereas the less important data is none or less protected, as shown in [9], [10]. Paper [11] and [12] specify how error protection is unequally applied to different part of the video stream. We extend this idea in case of object based coding (i.e., MPEG-4 AVO). In this case, the classification process specifies how assigning priority levels to each *Access Units* within an AVO. From such classification, an Unequal Error Protection (UEP) mechanism can be performed through forward error correction. It is quite obvious that the most important AVO data must be protected as strongly as possible against packet loss during transmission. This section presents first Reed Solomon codes and then our proposal for protecting MPEG-4 AVO.

#### A. Reed-Solomon FEC Codes

The aim of Reed-Solomon (RS) codes is to produce at the sender  $n$  blocks of encoded data from  $k$  blocks of source data in such a way that any subset of  $k$  encoded blocks suffices at the receiver to reconstruct the source data [13]. RS code is called an  $(n, k)$  code. RS code  $(n, k)$  is defined over the Galois Field  $GF(2^q)$  where each block contains  $q$  bits. The codeword length  $n$  is restricted by  $n \leq 2^q - 1$ . We choose  $q$  to be 8 bits and therefore  $n \leq 255$ . With this value for  $q$ , encoding and decoding are processed easier.

Let  $x = x_0 \dots x_{k-1}$  be the source data,  $G$  an  $(k \times n)$  generator matrix of the  $(n, k)$  RS code, and  $y$  the encoded data. Then,  $y$  is given by:

$$y = G \cdot x \quad (\text{Eq.5})$$

$G$  consists of two parts. The first part is the  $(k \times k)$  identity matrix  $I_k$ . The second part is an  $(h \times h)$  matrix, with  $h = n - k$ .  $G$  is given by (6).

$$\begin{aligned} G: GF(2^8)^k \rightarrow GF(2^8)^n \\ x_i \mapsto y_i = x_i \cdot G, \quad G = \begin{bmatrix} I_k & F_{k,h} \end{bmatrix} \\ \text{with } F_{k,h} = \{f_{r,c}\} \text{ a matrix } (k \times h) \\ \text{and } f_{r,c} = c^{r-1}, \quad r \in [1, k], c \in [1, h] \end{aligned} \quad (\text{Eq.6})$$

When  $G$  is used as generator matrix, the blocks of encoded data include a verbatim copy of the source. It simplifies the reconstruction of source data when few losses are expected.

#### B. Unequal Error protection (UEP) using adaptive RS codes

We assign priority to each AU to apply an unequal error protection to the data. Priority reflects both the semantic importance of the AVO within the scene and importance of particular video coding structures such as frame type (I, P, B or hierarchical stream if any BL or EL). Of course, UEP increases the traffic load due to control overhead. To correctly control the volume of transmitted data, we propose to control the proportion of FEC overhead with a ratio parameter called  $r$ , for each level of priority. We assume that moving from one level priority to other increases by a 1 percent this ratio. So, we chose a scale of 100 levels of priority ( $p \in [0..99]$ ). Then, the ratio  $r$  can be defined by:  $r = 0.01 \times p$  (Eq. 7)

Therefore, the traffic overhead is limited to 1 percent (i.e.,  $r=0.01$ ) for the data flow of priority 1, to 2 percent (i.e.,  $r=0.02$ ) for the data flow of priority 2, to 10 percent for the data flow of priority 10, and so on. Such fine scale provides a greater granularity for control of the traffic overhead that can be applicable to a large number of media encoding scheme.

Let us consider  $U_i$ , the  $i^{th}$  Access Unit in the flow of priority  $p$ . The main block of the proposed UEP is to determine the values  $n_i$  and  $k_i$  in such a way that the  $(n_i, k_i)$  RS code is efficient. The value  $k_i$  is defined as the number of packets in which  $U_i$  is broken when no error protection is performed. The value  $n_i$  depends on the priority  $p$ . It depends also on the length  $m_i$  of  $U_i$  because the traffic overhead introduced by redundant data does not become excessive.

Once the efficient  $(n_i, k_i)$  RS code is found, the coding step begins. We also investigate packetization process known as block of packets that was introduced in [14] and adapt it to MPEG-4 Object Plan. Data of  $U_i$  is placed in  $k_i$  horizontal packets ( $S_1, S_2, \dots, S_{k_i}$ ). Each packet has the same size of  $t_i$  bytes. Padding is added to the last

packet if  $m_i$  is not a multiple of  $k_i$ . Then the  $(n_i, k_i)$  RS code is applied across these packets, vertically. We generate  $h_i = n_i - k_i$  redundant packets ( $R_1, R_2, \dots, R_{h_i}$ ). After appending the RS codes, result packets are transmitted horizontally with a FEC header. Finally, the packet can be transmitted over RTP.

FEC header contains both the  $U_i$  sequence number and the values  $n_i$  and  $k_i$  of the RS code. In case of packet losses, the decoder needs this information to decode correctly the received packets. If the number of lost packets is not more than  $h_i$ , then the decoder will be able to recover  $U_i$ . Otherwise,  $U_i$  is completely lost. Figure 1 shows the format of packets sent on the IP network.

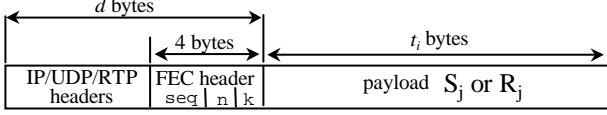


Figure 1: Header information and video packet format

In order to find the efficient value  $n_i$  for the  $(n_i, k_i)$  RS code, we proceed as follow: Let  $\zeta_{U_i}^*$  be the reserved byte-budget for error protection. It depends on the number of bytes used to send  $U_i$  when no error protection is performed. It is given by:  $\zeta_{U_i}^* = r \cdot (k_i \cdot d + m_i)$  (Eq. 8). Where  $d$  is the packet header size (i.e., when RTP is used with the proposed UEP,  $d = (20+8+12+4) = 44$  bytes). The relation between the real byte-budget spent on error protection,  $\zeta_{U_i}$ , and the RS code to be used can be stated as follows:  $\zeta_{U_i} = n_i \cdot t_i - m_i + d \cdot (n_i - k_i)$  (Eq. 9)

The error margin between  $\zeta_{U_i}$  and  $\zeta_{U_i}^*$  is  $\hat{\zeta} = \zeta_{U_i} - \zeta_{U_i}^*$ , that can be positive or negative. It cumulates along the data access unit arrivals. Using the formula (8) and (9), the fluctuation of the error margin can be written as:

$$\hat{\zeta}(n_i) = \begin{cases} 0 & , i=1 \\ (r+1) \cdot m_i - n_i \cdot t_i + (r \cdot k_i - n_i + k_i) \cdot d + \hat{\zeta}(n_{i-1}), & i > 1 \end{cases} \quad (\text{Eq. 10})$$

To respect the constraint given on traffic overhead, the best value  $n_i$  is the one that provides the smallest error margin in formula (10). Then,  $n_i$  is obtained by:  $\min_{n_i} |\hat{\zeta}(n_i)|$  such that  $n_i \in \mathbb{N}$ ,  $n_i \geq k_i, \forall m_i, k_i, t_i$

With the proposed UEP, the RS code evolves dynamically so the network bandwidth is correctly controlled according to video application requirement.

## V. PERFORMANCE EVALUATION

### A. IP Video Streaming System architecture

Figure 2 depicts the proposed MPEG-4 video streaming system. It is composed of a video server and client. The server streams MPEG4 video to the end terminal through an IP network using RTP. Two RTP sessions are created, the first session handles **AVO stream** and the second one handles **FEC stream**. The client decodes and composes the original MPEG-4 scene based on received AVO streams and FEC data. Each AVO is coded separately so the decoding process decodes also each AVO separately and then the composition module composes the original scene. The target transmission rate of the video server is calculated by the TFRC module. This information is sent to the "add/drop module" which adapts the video transmission rate using add/drop algorithms. Diffserv Marker module handles the marking of the different RTP packet with Diffserv Code Point before entering the Diffserv network. It implements an algorithm that better takes into account the characteristics and the relevance of MPEG-4 AVOs. Sensitive data streams will undergo a processing privilege from the Diffserv routers. Most important AVOs in the scene are marked with low drop precedence. Least important AVOs are marked with high drop precedence, and so on.

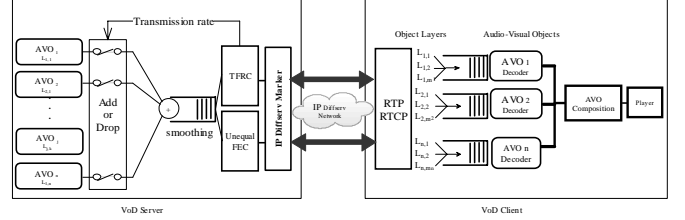


Figure 2: The MPEG-4 Video Streaming System Model

### B. Simulation Model

Intensive simulations are conducted to evaluate our MPEG-4 streaming with TCP-friendly transport mechanism and error protection scheme. We have used NS2 and we have developed an MPEG-4 Server (NS2 Agent) and an MPEG-4 client (NS2 Agent). The server reads and sends the different MPEG-4 AVOs found in video trace files to the client through the IP Diffserv network.

We used the network architecture shown in Figure 3 to simulate a unicast service provided by the MPEG-4 server attached to the node "S". The server sends data to the client attached to the node "C". R1 uses A Two Rate Three Color Marker (**TR3CM**) to mark the background traffic evenly among the different Assured Forwarding class. Recall that the MPEG-4 video packets are marked at the source (i.e., the video server) that knows well the characteristic of each video stream.

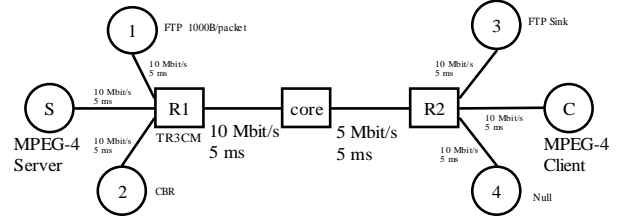


Figure 3: Network Simulation Topology

### C. MPEG-4 Video Traffic Characterization

The MPEG-4 traffic is obtained from the MPEG-4 trace file presented in [16]. In our simulation, the MPEG-4 presentation was obtained by using a set of AVOs components. We simulate an MPEG-4 scene composed of the following AVO: (1) AO (audio speech), (2) VO1 (background), (3) VO1 (speaker) and (4) VO3 (logo). These objects are sorted as follows: AO has the priority 1, it is the most important object in this scene. It is marked with Diffserv PHB AF11 (low drop precedence). VO1 and VO2 have the priority 2. They are marked with Diffserv PHB AF12 (medium drop precedence). Each Object is composed of 3 layers (one base layer and 2 enhancement layers) VO3 has the priority 3, it is the least important object in this scene. It is marked with Diffserv PHB AF13 (high drop precedence). Figure 4 shows the bit-rate of the MPEG-4 video objects that can be sent from the MPEG-4 server to the client during a period of 120 seconds.

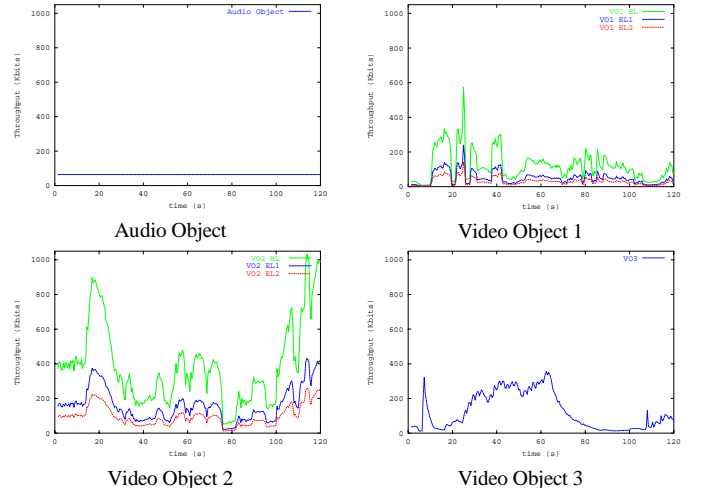
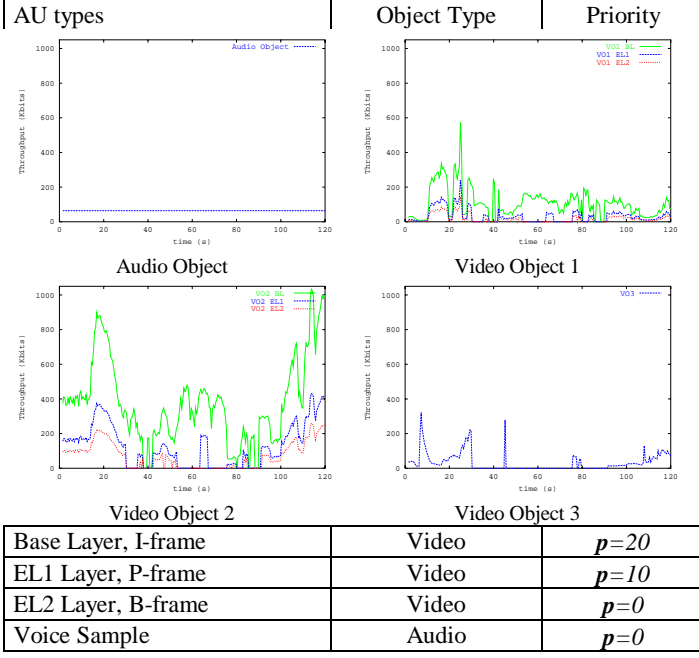


Figure 4: Instantaneous throughput of MPEG-4 AVOs layers



The table above shows the priority level of each AU. With respect to this error protection policy, the insertion of redundant data generates a global network traffic overhead equal to 7.0 % in this scene.

#### D. Simulation Scenarios

We perform simulations with different parameters according to these scenarios:

**Scenario 1:** our proposed combined error protection scheme (i.e., AVO protected with Unequal Error Protection and transmitted over IP Diffserv Network using TCP-friendly mechanism). Both mechanisms are based on AVOs priorities.

**Scenario 2:** AVO (without error protection) transmission over IP Diffserv and without TCP-friendly mechanism.

**Scenario 3:** AVO transmission using Unequal Error Protection (based on the described priority) using TCP-friendly Transport mechanism (TFRC).

**Scenario 4:** AVO transmission without any error protection using TCP-friendly Transport mechanism (TFRC).

In each scenario, we vary gradually the network load by CBR traffic and by using  $n$  FTP traffic each time in order to get more information on the behavior of the different mechanisms.

#### E. Results Analysis

In order to highlight the efficiency of our combined error protection scheme, we perform three measurements. The first measurement concerns the worthiness of FEC-based unequal error protection (UEP). We study the contribution of this mechanism on error resilience associated to each AVO. Figure 5 shows the results of the comparison between the decoded object ratio for **scenario 4** (no error protection) and **scenario 3** (UEP). The X-axis represents the throughput of the background traffic. As expected, the quantity of the AVOs decoded at the receiver side decreases when the network load increases because it entails more packet losses.

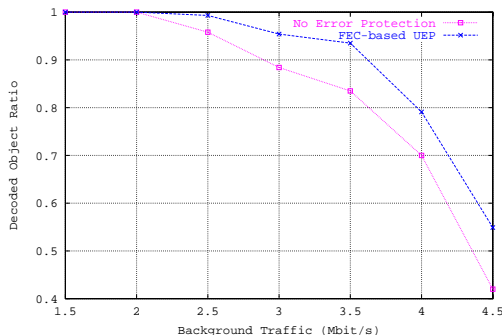


Figure 5: Correctly decoded AVO ratio vs. Background traffic throughput

Packet loss rises when using our FEC-based UEP because UEP increases the MPEG-4 packet-stream throughput by 7 %. For this reason, there is more packet losses in scenario 3 compared to scenario 4, for a given network load. However, the redundant UEP information better recovers lost packet at the receiver. Consequently, a particular *Access Unit* can be restored. Failures in the decoding process are rather distributed toward the less important objects, and then UEP reduces the effects of spatial and temporal errors propagation. Figure 5 shows that the decoded object ratio of scenario 3 is always better than scenario 4.

Second measurement concerns only the adaptation mechanism in case of TCP-friendly transport mechanism. Figure 6 show a scenario of a network session shared with eight FTP stream and one video streaming session. FTP starts streaming at time  $t=30s$  and stops at time  $t=90s$ . We can see in this Figure that the network resources are fairly shared among the different connections. The important AVO is always present. Video Object 3 is present when there are sufficient resources in the network.

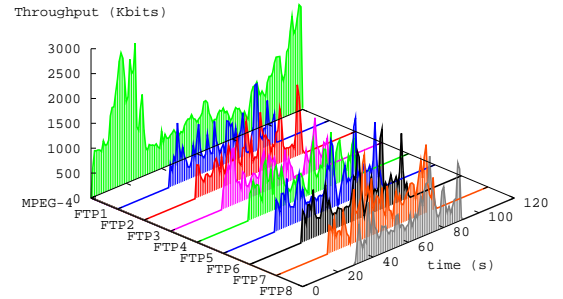


Figure 6: Effect of the Source Rate adaptation mechanism

The contribution of TCP-friendly mechanism on error protection associated to each AVO is also considered. In “scenario 1” 100% of packets loss were recovered by our UEP scheme. In this case, DiffServ network provides a differentiated level of QoS for each video stream depending to its priority class. It is visible for these scenarios that the losses happen mainly on the lower priority streams (i.e., O1 and some background traffics which are marked with high drop precedence). When best effort service is used, loss approximately follows a uniform distribution. This is due to the IP best Effort routers that use Drop Tail queue management policy. When the queue is full, all the incoming packets are dropped with the same probability. In contrast, DiffServ network provides a differentiated level of QoS for each stream depending to its priority class.

#### VI. CONCLUSION

We described a new adaptive video streaming framework for delivering MPEG-4 content over next generation IP networks with differentiated services support. MPEG-4 video streams based on Audio Visual Objects (AVOs) are automatically classified, packetized and streamed over the network according to data semantic relevancy and network resource availability. Combined with a FEC-based Unequal Error Protection and a TCP-friendly transports mechanism, the proposed video streaming system shows a significant improvement regarding to user-perceived quality, packet loss recovery and bandwidth share fairness.

#### REFERENCES

- [1] T. Ahmed, G. Buridant, A. Mehaoua, “Delivering of MPEG-4 Multimedia Content Over Next Generation Internet”, IFIP/IEEE MMNS, October 2001.
- [2] J. Shin, J. Won Kim and C.-C. Jay Kuo, “Content-Based Packet Video Forwarding Mechanism in Differentiated Service Networks”, IEEE PV, May 2000.
- [3] M. Albrecht, M. Köster, P. Martini, M. Frank, “End-to-end QoS Management for Delay-sensitive Scalable Multimedia Streams over DiffServ”, in Proc. LCN00, Tampa-FL, pp 314-323, November 2000.

- [4] Huai-Rong, Wenwu, and Ya-Qin Zhang, "Scalable Object-Based Video Multicasting Over The Internet", ICIP 2000, Vancouver, Canada, September 2000.
- [5] ISO/IEC 14496-1 "Coding of audio-visual objects - Part 1, 2, 3: Systems", final committee draft, May 1998.
- [6] T. Ahmed, A. Nafaa and A. Mehaoua "An Object-Based MPEG-4 Multimedia Content Classification Model for IP QoS Differentiation" IEEE Symposium on Computers and Communications - ISCC'2003. July 2003.
- [7] S.Floyd et al. "Promoting the Use of End-to-End Congestion Control in the Internet" Proc. of IEEE/ACM Transaction on Networking, 7(4), pp.458-472, Aug. 1999.
- [8] M.Handley, J.Padhye, S.Floyd, and J.Widmer "TCP Friendly Rate Control (TFRC): Protocol Specification" Internet draft, (expires April 2003), work in progress. 2002.
- [9] I. Rhee and S. Joshi, "Error recovery for interactive video transmission over the Internet," IEEE J. Selected Area Comm., vol. 18, no. 6, pp. 1033-49, June 2000
- [10] B. Girod, K. Stuhlmuller, M. Link, and U. Horn, "Packet Loss Resilient Internet Video Streaming", in Proceedings of SPIE Visual Communications and Image Processing, San Jose, CA, January 1999, pp. 833-844.
- [11] G. Liebl, M. Wagner, J. Pandel and W. Weng "An RTP Payload Format for Erasure-Resilient Transmission of Progressive Multimedia Streams" draft-ietf-avt-uxp-04.txt, work in progress, expires: May 2003
- [12] Adam H. Li, et al. "An RTP Payload Format for Generic FEC with Uneven Level Protection" Internet draft, work in progress, expires: May 4, 2003
- [13] S. Lin, D.J. Costello, "Error Control Coding: Fundamentals and Applications", Prentice Hall, 1983.
- [14] A. Albanese, J. Bloemer, J. Edmonds, M. Luby, and M. Sudan, "Priority Encoding Transmission", IEEE Trans. on Inf. Theory 42, pp. 1737-1747, Nov. 1996.
- [15] J. Rosenberg and H. Schulzrinne "RFC 2733 An RTP Payload Format for Generic Forward Error Correction" IETF Editio, December 1999
- [16] Frank H.P. Fitzek, and Martin Reisslein "MPEG-4 and H.263 Video Traces for Network Performance Evaluation" IEEE Network, vol 5, no 6, pp 40-54, November 2001.