

# IP Video Streaming With Fine-Grained TCP-Friendly Rate Adaptation

Toufik Ahmed<sup>1,2</sup>, Ahmed Mehaoua<sup>2</sup>, Raouf Boutaba<sup>1</sup> and Youssef Iraqi<sup>1</sup>

<sup>1</sup> University of Waterloo, Dept. of Computer Science  
200 University Avenue West, Waterloo,  
Ont. N2L 3G1, Canada

{tad, rboutaba, yiraqi}@bbr.uwaterloo.ca

<sup>2</sup> University of Versailles, CNRS-PRISM Lab.

45 av. des Etats-Unis, 78035, Versailles, France  
{tad, mea}@prism.uvsq.fr

**Abstract:** Video streaming over the Internet is a challenging task since the Internet is a shared environment offering only best effort service. That is, it offers no quality of service and no guarantee of resources in term of (1) bandwidth, (2) transfer delay, (3) delay variation (jitter), and (4) packet losses. Then, network stability and traffic fairness become critical issues. To solve these problems some source rate control and adaptation should be introduced for UDP traffic as well, in such a way that this traffic becomes TCP-compatible “TCP-friendly”. In this article we propose an adaptive streaming framework for unicast MPEG-4 streams over TCP/IP networks. Based on Audio-Visual Content (AVOs) classification and network congestion feedback, video sources dynamically adds and drops MPEG-4 AVO to the streamed multiplex to conform to the TCP-Friendly Rate Control (TFRC) mechanism. Using a content classification model, TFRC automatically adjusts the number of AVOs to be streamed to adapt to network congestion while given much attention to the quality of the service perceived by the end-user. To achieve such a dynamic output rate and video quality adjustment, MPEG-4 AVOs are classified and multiplexed according to both application-level QoS parameters and AVOs semantic descriptors. AVOs requiring same QoS from the network are automatically classified and mapped to one of the available IP DiffServ PHB (Per Hop Behaviors). Performance evaluation shows that transmitted video gracefully adapts to network bandwidth variations while optimizing user perceived quality.

**Keywords:** Video Streaming, TCP-Friendly Rate Control, QoS, Object Classification Model.

## 1 Introduction

Streaming audio and video on the Internet is becoming more popular. This rapid expansion underlies a new challenge for efficient handling of Internet traffic. The majority of multimedia applications perform over an RTP stack that is implemented on top of UDP/IP. However, UDP offers no congestion control mechanism and therefore is unaware of network condition and unfair towards other competing traffic. Today’s Internet traffic is dominated by TCP. TCP uses several mechanisms to handle network congestion such as: AIMD (Additive Increase and Multiplicative Decrease), slow start,

congestion avoidance, fast retransmit and fast recovery. Thus, it is crucial that UDP traffic performs also TCP-friendly congestion control [1].

In this article, we consider the application scenario of streaming object oriented video over IP with congestion control mechanism. We investigate Quality of Service (QoS) interaction provisioning between an MPEG-4 video application and the IP Diffserv network. To achieve the best possible QoS, all the components involved in the transmission process must collaborate together. In this regards, we propose the following mechanism. First, the streaming server classifies the MPEG-4 Audio Visual Objects (AVOs) based on application-level QoS criteria and/or AVOs semantic descriptors according to MPEG-7 framework [2]. Second, the server performs rate adaptation through the adjustment of the number of streamed AVO based on network state. We use a TCP-friendly to adapt the server rate to network condition. The server tries to deliver the maximum number of objects that can fit in the current allowed transmission bandwidth. It begins streaming the important AVO after what it streams the less important AVO. This mechanism allows the server to deal with network congestion by stopping streaming less important AVO when congestion is detected. Finally, the server affects one of the IP diffserv classes of service that guarantee a good quality of service, using an intelligent marking scheme. The more the AVO is important is the scene, the more the server affects low drop precedence class. When network congestion occurs less important AVOs will be dropped automatically by the diffserv router. Lost packets notify the streaming server to reduce its transmission rate.

The idea of adaptive streaming using TCP-friendly helps to prevent the server entering congestion collapse in which the network link is heavily used and little useful work is being done. So to prevent such situation, all applications must perform TCP-like congestion control mechanisms. Traffic that does not perform in TCP-friendly manner is dropped by the router [3].

The paper is organized as follows: Section 2 reviews some background and related work. The proposed adaptation mechanism is presented in Section 3. Simulation model and performance analysis are presented in Section 4 and 5 respectively. Finally, Section 6 concludes the paper.

## **2 Background and Related Work**

Adaptive video streaming is not a new topic. Several researches have been conducted in this area and various approaches have been proposed. While all the work done until now concerns essentially layered video streaming, our work is different from others by using the new concept of object scalability introduced in the MPEG-4 standard.

Video scalability is the key approach for achieving rate adaptation. Many works exploit the video scalability to implement the rate control via TCP-Friendly congestion management. McCanne et al. [4], propose a method that enables different receivers to adapt to bandwidth fluctuations by adjusting the number of layers to which multicast users subscribe. The video stream is divided into a number of multi-resolution layers. Each layer is transmitted to a separate multicast group.

Another type of server rate adaptation using the video scalability is to adjust the codec quantization parameters. In [5] and [6] the video server continually negotiates the available bandwidth and modifies the codec quantization values accordingly. We recall that adapting codec quantization values is a CPU-intensive task which can affects the

performance of the video server. The idea of quantizer scale was also used in the context of MPEG-4 in the work presented in [7]. This later employs Fine Granular Scalability which uses a layered coding algorithm. A TCP-friendly rate control algorithm adjusts the rate of each video layer by regulating the level of quantization.

In order to apply a video rate control, it is necessary to have a scheme that returns the transmission rate. The Loss-Delay Based Adaptation Algorithm (LDP) [8] is an end-to-end rate adaptation scheme for adjusting the transmission rate of multimedia applications to the congestion level of the network. Another class of congestion control schemes applies the additive increase, multiplicative decrease (AIMD) algorithm in some form [9],[10]. The Rate Adaptation Protocol (RAP) proposed by Rejaie et al. [9] is an example of an AIMD-based scheme, where the source performs rate adaptation based on acknowledgments sent by the receivers (feedback information). The acknowledgments are used to detect losses and estimate round trip time (RTT). Rate adaptations are performed once per RTT, with transmission rate being increased linearly in the absence of loss, and transmission rate being decreased multiplicatively when congestion is detected. RAP uses the ratio of short-term to long-term averages of RTT to fine tune the sending rate. The RAP protocol was applied in the context of unicast video delivery [11]. The video is a layered constant-bit rate. All the layers have the same throughput. The rate control algorithm used by the server adapts the video quality to network state by adding and dropping layers to efficiently use the available bandwidth. The algorithm takes into consideration the status of the receiver buffer, making sure that base layer packets are always available for playback.

The TCP-friendly congestion control mechanism that was developed recently is TCP-friendly Rate Control Protocol (TFRC) [12]. TFRC provides sufficient responsiveness by taking into consideration all the parameters that affect the TCP rate such as packet size ( $s$ ), loss rate ( $p$ ), Round-Trip Time (RTT), retransmission timeout value ( $t_{RTO}$ ), and number of packets acknowledged by a single packet ( $b$ ). The key advantage of TFRC is that it has a more stable rate during the session lifetime. The calculated rate is obtained by using the TFRC is [12]:

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

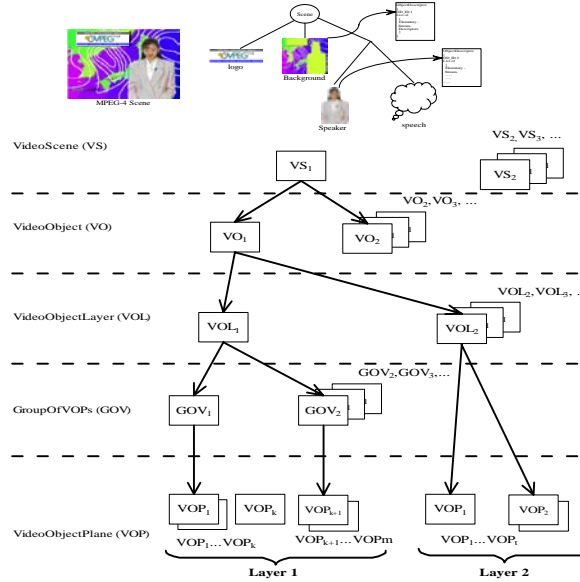
In contrast to previous adaptive video streaming mechanisms, the proposed approach in this paper uses the concept of object scalability introduced in MPEG-4. It adapts the video quality to network state by adding or dropping objects and their associated layers according to the network state. This solves the problems of heterogeneity of receivers and redundancy of data. Objects are encoded separately which does not prevent one object from being decoded if another one is not received.

### 3 Object-Based Adaptive Video Streaming using TCP-Friendly Mechanism

#### 3.1 MPEG-4 Framework

The MPEG-4 standard [13] introduces a new technique of coding multimedia scenes called “object-based compression”. This technique allows the encoding of different audio-visual objects in the scene independently.

An MPEG-4 scene consists of one or more AVOs, each of them is characterized by temporal and spatial information. The hierarchical composition of an MPEG-4 scene is depicted in Figure 1. 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)



**Figure 1: Hierarchical composition of an MPEG-4 scene**

To take benefits from the object-based compression, we propose to use an intelligent adaptation to cope with network congestion and client terminal heterogeneity. We proposed in [14] a mechanism to classify the MPEG-4 AVOs at the video server from most important AVO to least important AVO. Several methods can be 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 [2] can provide the relevant information needed to handle object priority. By classifying these objects, 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. This technique is presented in next subsections.

### 3.2 Notations and Parameters

Let  $\mathbf{S}$  be a set of MPEG-4 AVOs containing  $n$  AVOs  $\mathbf{O}_j$ , 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 [14]. Each object  $\mathbf{O}_j$  may consist of  $m_j$  layers ( $m_j \geq 1$ ). Note that lower layers within an object have higher priorities than higher layers.

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

$$\begin{aligned} \forall j, 1 \leq j < n : P(\mathbf{O}_{j+1}) &\leq P(\mathbf{O}_j) \\ \forall j, 1 \leq j < n, \forall l, 1 \leq l < m_j : P(L_{j,l+1}) &< P(L_{j,l}) \end{aligned} \quad \mathbf{L}_{j,l} \text{ is the Layer number } l \text{ of the Object } \mathbf{O}_j \quad (2)$$

By using Eq. 2 we can construct an Audio-Visual Entity set called  $\mathbf{E}$  composed of all object layers ordered by their priorities.

$\mathbf{E} = \{L_{1,1}, L_{1,2}, \dots, L_{1,m_1}, L_{2,1}, L_{2,2}, \dots, L_{2,m_2}, \dots, L_{n,1}, L_{n,2}, \dots, L_{n,m_n}\}$ . We will note  $\mathbf{E}$  as follows:

$$\mathbf{E} = \{e_1, e_2, \dots, e_w\} \text{ with } w = |\mathbf{E}| = \sum_{j=1}^n m_j$$

Note that if two objects have the same priority, then the associated layers of an object have the same priority as the object (in relation to other objects) with the lower layers having higher priorities than higher layers.

At time  $t_i$ , the function  $\mathbf{R}_i$  gives the instantaneous transmission rate of an audio-visual entity. For example, the audio-visual entity  $\mathbf{e}_p$  has an instantaneous transmission rate equal to  $\mathbf{R}_i(\mathbf{e}_p)$ , and the object  $\mathbf{O}_j$  has the instantaneous transmission rate equal to  $\mathbf{R}_i(\mathbf{O}_j)$ .

The adaptation mechanism operates as follows: The server evaluates the network state from the information gathered (i.e. RTT and loss rate) at time  $t_i$ , then computes the allowed sending rate  $\mathbf{R}_{TCP}$  using Eq. 1. The server tries to send as much as possible of the audio-visual entities without exceeding  $\mathbf{R}_{TCP}$  taking into consideration entities priorities. Details of the adding and the dropping process will be presented in section 3.4 and 3.5 respectively.

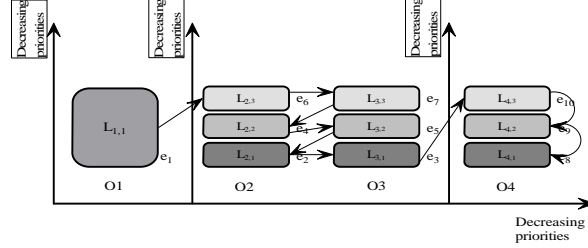
### 3.3 Example

Assume that we have an MPEG-4 scene composed of four audio-visual objects:  $\mathbf{O}_1$ ,  $\mathbf{O}_2$ ,  $\mathbf{O}_3$  and  $\mathbf{O}_4$ . Assume that  $\mathbf{O}_1$  is composed of a single layer, and that each of  $\mathbf{O}_2$ ,  $\mathbf{O}_3$  and  $\mathbf{O}_4$  is composed of three layers (one base layer and two enhancement layers). Also assume that the classification layer associates AVO priorities as follows (see Figure 2):

- $\mathbf{O}_1$  is the most important
- $\mathbf{O}_2$  and  $\mathbf{O}_3$  have the same priority
- $\mathbf{O}_4$  is the less important

Then,  $\mathbf{E} = \{L_{1,1}, L_{2,1}, L_{3,1}, L_{2,2}, L_{3,2}, L_{2,3}, L_{3,3}, L_{4,1}, L_{4,2}, L_{4,3}\} = \{e_1, e_2, \dots, e_{10}\}$ . Here  $w=10$ .

The video server adds audio-visual entities in the order of their importance (i.e. from left to right in the set  $\mathbf{E}$ ) as shown in see Figure 2. Entities are dropped in reverse order (i.e. from right to left) until matching the target sending rate.



**Figure 2: Transmission Order of different video objects/layers**

### 3.4 ADD of Audio-Visual Objects

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  $t$ . 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 is satisfied:

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

At the client side, the new audio-visual entity must be buffered and synchronized to the current playback time. For this fact, the server must start streaming with a synchronization marker of a synchronization frame such as I-VOP. The estimated throughput of the object being streamed is also considered. This allows assuring that the object being streamed has allows sufficient short-time resources. Section 3.6 gives more details about stability management.

### 3.5 DROP of Audio-Visual Objects

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 is satisfied:

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

The server can achieve a minimum throughput by assuring that the most important object or the base layer is always streamed a least. Assuring a minim throughput gives a minimum quality at the receiver side.

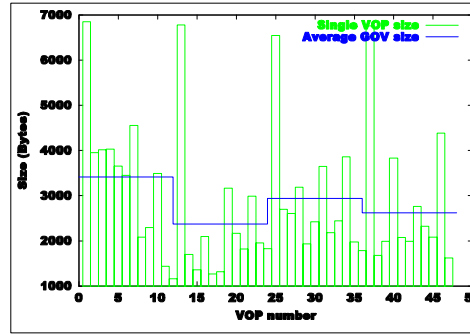
### 3.6 Handling Stability

Since the TFRC compute the new target rate each RTT, adding and dropping audio-visual entities can lead to undesired oscillation and poor video quality at the receiver. To prevent from such behavior, several measures are taken into consideration.

First, the TFRC module copes with oscillation behavior by using EWMA (Exponentially Weighted Moving Average) to detect out-of-control situations quickly. EWMA statistics

are used to attempt to respond dynamically to the changing value in the measured RTT and loss and attempt to regulate this value to reflect as much as possible the reality. In TFRC, the loss rate is measured in terms of loss interval which represents the number between two consecutive loss events. The mechanism reacts too strongly to single loss events and ensures that allowed sending rate do not change aggressively.

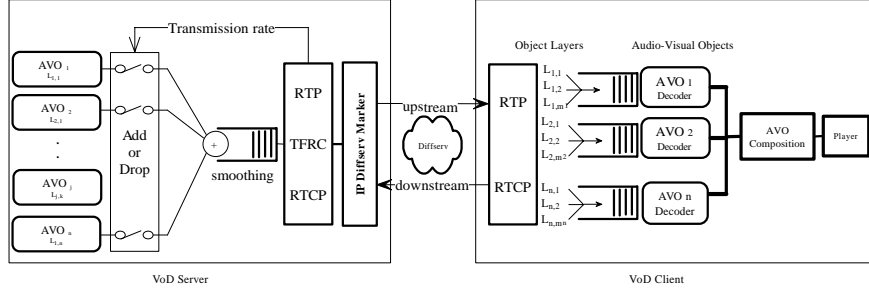
Second, we propose to adjust the server transmission rate at the beginning of each GOV (Group of video object plane). So the synchronization is assured by starting the transmission with I-VOP. Thus, the new transmission rate obtained from TFRC module is used to adapt video sending rate. Figure 3 shows four GOVs (each group has twelve VOPs). The average line in the Figure shows the server transmitting rate at the beginning of each GOV of a current Video Object (VO). If this value does not fit in the current available bandwidth then the server does not stream the object. In other words, the server smoothes its sending data to reduce its aggressivity.



**Figure 3: Stability management**

### 3.7 System architecture

Figure 4 depicts the general block diagram of our MPEG-4 Video on Demand system. It is composed of a video server and a video client. The server streams the audio-visual object to the client via an IP network using the RTP protocol [15]. The client decodes and composes the original MPEG-4 scene. As shown in Figure 1 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. IP Diffserv Marker module handles the marking of the different RTP packet with Diffserv Code Point before entering the Diffserv network.



**Figure 4: General Block Diagram of the MPEG-4 VoD System**

Diffserv object prioritization aims to privilege the transport of some AVOs compared to others. When network congestion occurs, less important AVOs streams are dropped automatically by the active queue implemented in the Diffserv router. The work detailed in [16] presents a method to handle a layered MPEG-4 stream over IP Diffserv. We extend this approach by handling MPEG-4 AVOs streams prioritization over IP Diffserv network. Recall that the MPEG-4 scene contains many MPEG-4 AVOs sorted according to their importance in the presentation. Therefore, the IP Diffserv Marker tags each video data packet belonging to one AVO with one of the supported Diffserv class of service to reflect object priority. Hence, important objects will be marked with a low drop precedence to guarantee a minimum loss.

It is worth noting that the choice of TCP or TFRC-based congestion control mechanisms is completely orthogonal to whether the traffic is best-effort or not. If the transport protocol is using conformant end-to-end congestion control, then the transport protocol does not have to know whether the traffic is being treated as best-effort or as part of a Diffserv class.

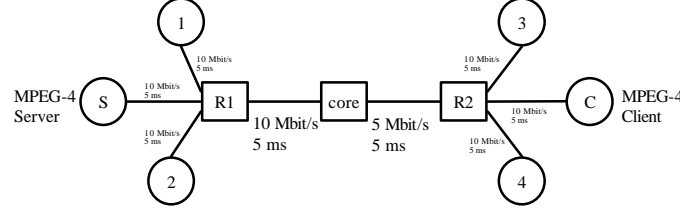
## 4 Simulation model

### 4.1 Network architecture

Simulations are conducted using the network simulator *ns2*. We used the network architecture shown in Figure 5 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”. We developed an MPEG-4 server in *ns2* with TFRC capability. The client is also developed in *ns2* and which extends the capabilities of the RTP sink by reporting statistic information to the server. The network is loaded by  $n$  FTP streams carried over TCP ( $n$  ranges from 0 to 8). This allows the link between the routers “R1” and “R2” to be on congestion differently. FTP sources always have a packet to send and always send a maximal-sized (1000-bytes) packet as soon as the congestion control window allows them to do so. FTP sink immediately sends an ACK packet when it receives a data packet. The queue in the routers has a size of 50 packets. The core IP Diffserv router examines incoming packets and reacts according to the marking, whereas “R1” is an edge router that implements Marking/Classification policy on incoming packets. R1 uses A Two Rate Three Color Marker (**TR3CM**) [17] to mark the background. Therefore, background traffic is evenly distributed among the different Diffserv classes. We recall



that the video traffic is marked at the MPEG-4 server according to AVOs priorities. The bottleneck link between the core router and R2 has a 5 Mbit/s of bandwidth.



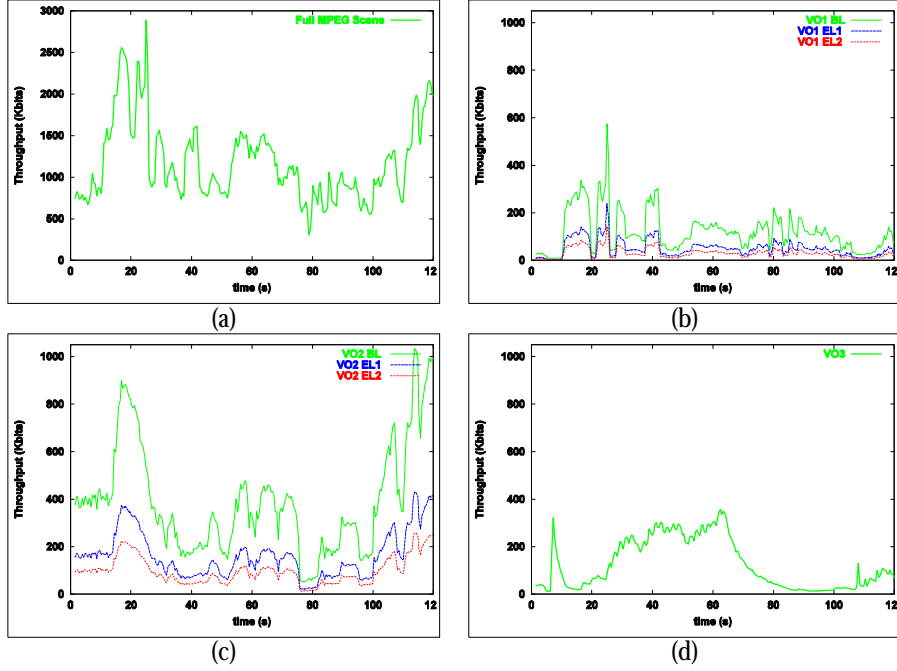
**Figure 5: Network topology**

## 4.2 MPEG-4 Traffic model

The MPEG-4 traffic is obtained from the MPEG-4 trace file presented in [18]. In our simulation, the MPEG-4 presentation was obtained by using a set of AVOs components. We simulate the weather presentation shown in Figure 1 by using four multimedia objects: AO (audio speech), VO1 (background), VO1 (speaker) and 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 6 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. The complete scene is shown in Figure 6 (a). The Audio Object is a constant bit rate at 64Kbits/s. An Audio packet is sent each 125ms. Video object 1 has an average throughput of 200 Kbit/s and a peak rate of 956 Kbit/s. This object is composed of three Layers: BL (Base Layer), EL1 (Enhancement Layer 1) and EL2 (Enhancement Layer 2). The throughputs of the different layers are shown in Figure 6 (b). Video object 2 has an average throughput of 650 Kbit/s and a peak rate of 1722 Kbit/s. This object is composed of three Layers: BL, EL1 and EL2. The throughputs of the different layers are shown in Figure 6 (c). Video object 3 has an average throughput of 124 Kbit/s and a peak rate of 356 Kbit/s. It is composed of one single layer (see Figure 6 (c)).



**Figure 6: Instantaneous throughput of the different MPEG-4 Video Object**

## 5 Simulation Analysis

We perform an intensive simulation, each time with different parameters to see the behavior of our video on demand system. We vary the number  $n$  of FTP source according to following scenarios: (1) **Scenario A**: one FTP source; (2) **Scenario B**: two FTP sources; (3) **Scenario C**: four FTP sources; (4) **Scenario D**: eight FTP sources. FTP sources send data from time  $t=30s$  until time  $t=90s$ . We present only results of scenario D due to the lack of spaces.

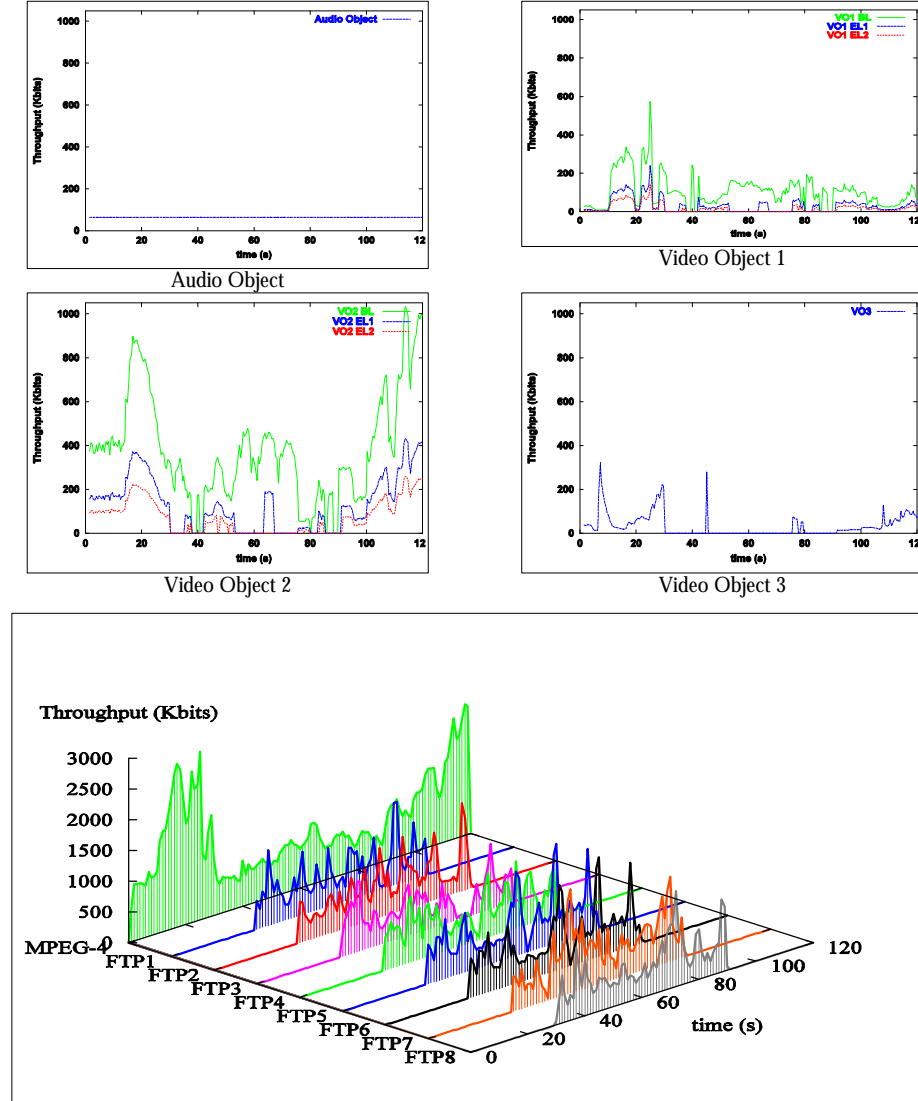
This section presents some QoS measurement such as, the video server throughput as a function of network state, packet loss and end-to-end packet transmission delay.

### 5.1 Video Server Throughput

The video server regulates its transmission rate to reflect the allowed rate by adding or dropping audio-visual entities. Results obtained of the different scenarios are shown in Figures below. Also, to simplify the interpretation of the results, Table 1 summarizes the transmission ratio per AVO stream observed during the period of the simulations (120s). Note that the FTP sources begin data transmission at time  $t=30s$ , and stop at time  $t=90s$ . VO3 has the low ratio since it has the lowest priority is the scene. VO1 and VO2 have the same priority, so the corresponding layers have more or less the same transmission ratio.

Scenario D is interesting since we see the effect of our adaptation mechanism. We can see that the audio object is always present and that less important objects (respectively

object layers) are not transmitted when the shared bandwidth is not sufficient. Our adaptation mechanism begins transmitting data from important audio-visual entity to less important. We can see that all the streams (FTP and video) fairly share the bandwidth.



**Figure 7: Scenario D**

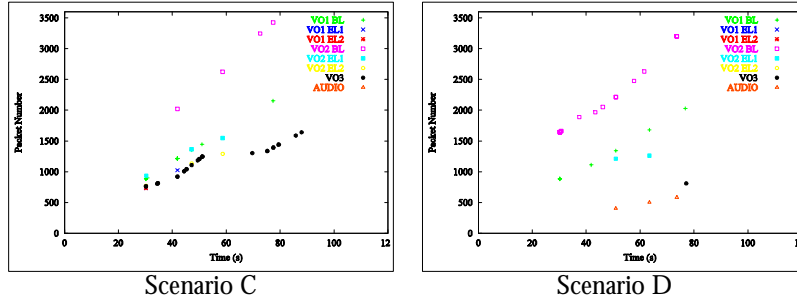
**Table 1: Transmission ratio per MPEG-4 objects**

AVO Scenario	Audio	VO1			VO2			VO3
		BL	EL1	EL2	BL	EL1	EL2	
A	100%	100%	100%	100%	100%	100%	100%	100%
B	100%	100%	100%	100%	100%	100%	100%	100%
C	100%	100%	94%	87%	100%	96%	92%	55%
D	100%	89%	60%	53%	97%	77%	71%	26%

Without using our adaptation mechanism, the server transmits the audio-visual entities without any regulation as shown in Figure 6. The loss may increase and the network may enter in congestion collapse.

## 5.2 Packet Loss

Figure 8 shows lost packets for scenarios C and D using our adaptation mechanism. Scenario A does not experience any loss. In scenario B, some lost packets are observed on VO3. This is due to the active queue of the DiffServ router which drops lower priority packets when a predefined threshold is reached to prevent congestion. In scenario C, we observe also some loss on lower priority packets but in scenario D high priority packets are also dropped. This is due to: (1) lower priority packets are not transmitted because our adaptation mechanism regulates the server transmission rate by stopping streaming lower priority packets and (2) AVO1 and AVO2 request more bandwidth in our scene and cause some congestion.

**Figure 8: MPEG-4 Packet Loss Using Our Adaptation Mechanism**

## 6 Conclusion

Most of the work done on “TCP-friendly” mechanisms deals with the problems of network stability and traffic fairness. Few attentions have been paid to the applications using them, and in particular, to the quality of service perceived by end users. Therefore, in this article we proposed an adaptive video streaming framework that provides both output rate and visual quality control for MPEG-4 based multimedia applications. The proposed framework relies on two cooperative components, (1) a TCP-Friendly Rate Control mechanism that estimates the targeted output rates for video sources; (2) and media object classification model that automatically adds and drops MPEG-4 Audio-Visual Objects to the multiplex stream to match the allowed source bit rate. These

components are combined to form a new “Media Classification Layer” that improve the MPEG-4 system architecture over QoS-capable IP networks. Performance evaluation through intensive simulations and various IP Diffserv configurations, shown a significant reduction of the video packets loss during network congestion, a smooth video source rate adaptation, a fair bandwidth sharing among UDP and TCP streams, and an increase in the Quality of the service with integration of Object descriptors (OD) during the media classification and the multiplexing process.

## References

1. Mahdavi, and S.Floyd “TCP-Friendly Unicast Rate-Based Flow Control”, Technical note sent to the end2end-interest mailing list, Jan., 1997.
2. ISO/IEC JTC1/SC29/WG11, “MPEG-7 overview,” N4980, Jul. 2002.
3. S.Floyd, M. Handley, J. Padhye, and J. Widmer “Equation-based congestion control for unicast applications” Proc. of ACM SIGCOMM, pages 43–56, Aug. 2000.
4. S. McCanne “Scalable compression and transmission over Internet multicast video” Ph.D thesis University of California, Berkeley, Dec. 1996.
5. T.V.Lakshman, P.P.Mishra, and K.K.Ramakrishnan “Transporting compressed video over ATM networks with explicit rate feedback control,” in *Proc. IEEE Infocom 97*, pp. 38–47, Apr. 1997.
6. N.G.Duffield, K.K.Ramakrishnan, and A.R. Reibman “SAVE: An Algorithm for Smoothed Adaptive Video Over Explicit Rate Networks” *IEEE/ACM transactions on networking*, vol. 6, no. 6, Dec. 1998.
7. N.Wakamiya, M.Miyabayashi, M.Murata, and H.Miyahara “MPEG-4 Video Transfer with TCP-friendly Rate Control” in *IFIP/IEEE MMNS 2001*, pp 29-42, Oct. 2001.
8. D.Sisalem, and H.Schulzrinne “The Loss-Delay Adjustment Algorithm: A TCP-friendly Adaptation Scheme, Network and Operating System Support for Digital Audio and Video” (NOSSDAV), Cambridge, UK, 8-10, Jul. 1998.
9. R.Rejaie, M.Handley, and D.Estrin “RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet” *Proc. IEEE Infocom’99*, pp. 1337-1345, Mar. 1999.
10. D. Bansal and H. Balakrishnan. Binomial Congestion Control Algorithms. In *Proc. IEEE Infocom ’01*, Anchorage, AK, Apr. 2001.
11. R.Rejaie, M.Handley, and D. Estrin “Layered quality adaptation for Internet video streaming” *IEEE Journal of selected areas in communications*, vol. 18, No. 12, Dec. 2000.
12. M.Handley, S.Floyd, J.Padhye, and J.Widmer “RFC 3448, TCP Friendly Rate Control (TFRC): Protocol Specification”, Request for Comments, IETF, Jan. 2003.
13. ISO/IEC 14496-1 “Coding of audio-visual objects, Part 1,2, 3”, final committee draft, May 1998.
14. T. Ahmed, A. Nafaa , A. Mehaoua “An Object-Based MPEG-4 Multimedia Content Classification Model for IP QoS Differentiation”, *Proc. of 8<sup>th</sup> IEEE Symposium on computers and communications ISCC’03*, pp. 1091-1096, Jul. 2003
15. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson “RFC1889 RTP: A Transport Protocol for Real-Time Applications”, Request for Comments, IETF, Jan. 1996.
16. T. Ahmed, A. Mehaoua, and G. Buridant “Implementing MPEG-4 video on demand over IP differentiated services” *Proc. IEEE Globecom’01*, pp. 2489 - 2493, Nov. 2001.
17. J. Heinanen and, R. Guerin “RFC2698: A Two Rate Three Color Marker TRTCM”, Request for Comments, IETF, Sept. 1999
18. 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, Nov. 2001.