Adaptive Packet Video Streaming Over IP Networks: A Cross-Layer Approach

Toufik Ahmed, Ahmed Mehaoua, Raouf Boutaba, *Member, IEEE*, and Youssef Iraqi

Abstract—There is an increasing demand for supporting real-time audiovisual services over next-generation wired and wireless networks. Various link/network characteristics make the deployment of such demanding services more challenging than traditional data applications like e-mail and the Web. These audiovisual applications are bandwidth adaptive but have stringent delay, jitter, and packet loss requirements. Consequently, one of the major requirements for the successful and wide deployment of such services is the efficient transmission of sensitive content (audio, video, image) over a broad range of bandwidth-constrained access networks. These media will be typically compressed according to the emerging ISO/IEC MPEG-4 standard to achieve high bandwidth efficiency and content-based interactivity. MPEG-4 provides an integrated object-oriented representation and coding of natural and synthetic audiovisual content for its manipulation and transport over a broad range of communication infrastructures. In this paper, we leverage the characteristics of MPEG-4 and Internet protocol (IP) differentiated service frameworks, to propose an innovative cross-layer content delivery architecture that is capable of receiving information from the network and adaptively tune transport parameters, bit rates, and QoS mechanisms according to the underlying network conditions. This service-aware IP transport architecture is composed of: 1) an automatic content-level audiovisual object classification model; 2) a reliable application level framing protocol with fine-grained TCP-Friendly rate control and adaptive unequal error protection; and 3) a service-level QoS matching/packet tagging algorithm for seamless IP differentiated service delivery. The obtained results demonstrate, that breaking the OSI protocol layer isolation paradigm and injecting content-level semantic and service-level requirements within the transport and traffic control protocols, lead to intelligent and efficient support of multimedia services over complex network architectures.

Index Terms—Content-based rate adaptation, Internet protocol (IP) quality-of-service (QoS), MPEG-4/7, service-aware transport protocols, unequal error protection.

I. INTRODUCTION

RAPID ADVANCES in digital video coding and networking is leading to the development of a wide range of new audiovisual services and applications. Such applications include wired/wireless videoconferencing, interactive digital TV, remote diagnosis/surgery, distance/remote sensing, process monitoring, and tele-education. Many of these applications involve the use of enabling media coding and content analysis techniques for media compression, indexing, search and retrieval.

One of the major requirements for the successful and wide deployment of such applications is the efficient transmission of multimedia content (audio, video, text, images) over a broad range of bandwidth-constrained access network infrastructures, in particular Internet protocol (IP)-based next-generation networks.[1]

Audiovisual applications have adaptive bandwidth capability, but stringent delay, jitter, and packet loss constraints, which are not supported by current IP networks. Today, IP quality-of-service (QoS) mechanisms and architectures [differentiated service (DiffServ), integrated services (IntServ)] are expected to address these requirements and enable a wide spread use of real-time IP services. However, these QoS control models are not sufficient since they operate on per-IP domain, and not on an end-to-end basis. Service Level agreements (SLAs) are expected to address the end-to-end service provisioning, but in the context of mobile IP multimedia services, SLAs are hard to implement since there may not be enough resources available in some areas of the network as the terminal is moving to.

It is, therefore, important to design multimedia applications capable to adapt to system and network resource constraints while ensuring that end-user requirements are met. Our work differs from others by integrating digital information, including the content itself, semantic descriptions, QoS information, and to integrate them in one layer (cognitive layer) to enable efficient, scalable, and reliable content distribution to the end-user.

The key contribution of this paper is a combination of media content analysis techniques and network control mechanisms for adaptive video streaming over IP networks. We have designed, implemented, and evaluated a cross-layer video streaming system that includes the three following components.

- A content-based video classification model for automatic translation from video application level QoS (e.g., MPEG-4 object descriptor and/or MPEG-7 metadata framework) to network system level QoS [e.g., IP DiffServ per-hop-behaviors (PHBs)].
- A robust and adaptive application level framing protocol with video stream multiplexing and unequal forward error protection.
- A fine grained TCP-Friendly video rate adaptation algorithm.

The proposed mechanisms work together in order to achieve seamless quality of service. Their performance evaluation is carried out using network simulator 2 (ns2).

[1]Note that we do not address the entire family of next-generation networks as our scheme is targeted to wired networks mainly.
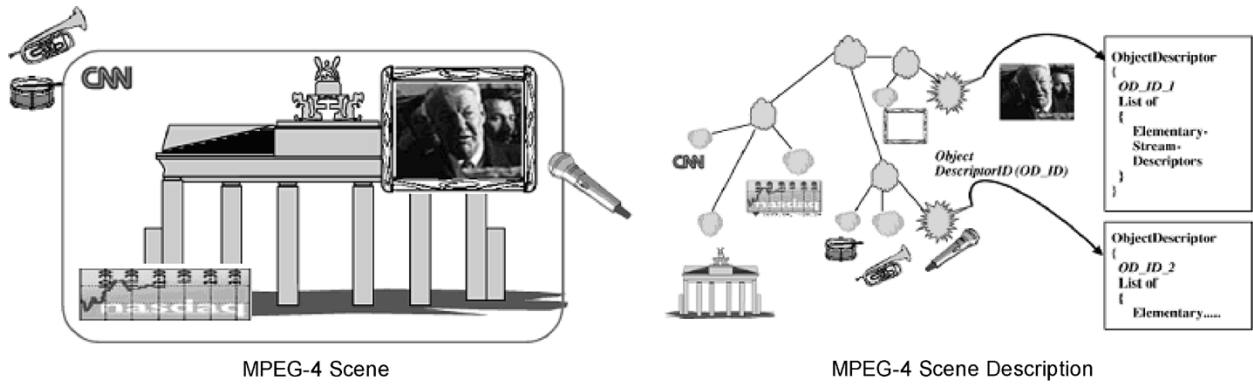
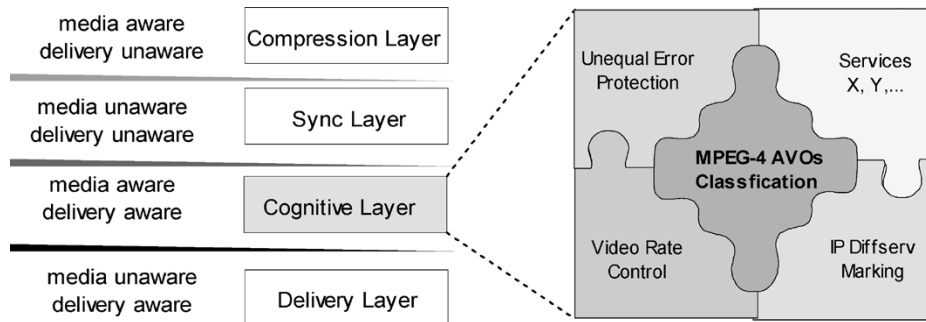Fig. 1. MPEG-4 AVOs with corresponding object descriptors.

Fig. 2. Cognitive middleware for MPEG-4 content delivery and control.

The reminder of the article is as follows: Section II presents an intelligent packet video transport mechanism that is aware of application level requirement and network QoS level. This transport mechanism implements several functionalities such as an audiovisual objects classification, an unequal error protection and a TCP-Friendly video rate adaptation. Section III presents an automatic video-object classification model. Sections IV and V describe video content protection through an unequal error protection and a fine grained TCP-Friendly video rate adaptation respectively. The overall cross-layer video streaming system architecture is analyzed in Section VI. Section VII are devoted to performance evaluation and analysis. We conclude in Section VIII.

II. INTELLIGENT PACKET VIDEO TRANSPORT FRAMEWORK

Video sequences are typically compressed according to the emerging MPEG-4 multimedia framework to achieve bandwidth efficiency and content-based interactivity. The original characteristic of MPEG-4 is that it provides an integrated object-oriented representation and coding of natural and synthetic audiovisual content for their manipulation and transport over a broad range of communication infrastructures.

Fig. 1 shows a collection of MPEG-4 objects with the associated object descriptors. These audiovisual objects (AVOs) can be classified according to some attributes or metadata in the corresponding object descriptors or provided by MPEG-7 tools. These attributes can refer to the Quality of Service parameters associated with every AVO (i.e., mean/peak bandwidth, loss rate, and jitter tolerance) or related to the structural and/or

semantic description of the content like proposed by MPEG-7 or TV-anytime.

The scene description is done at two levels: the **structure level** and the **semantic level**. The structure level indicates how the scene is composed and how the AVOs are arranged in the scene in terms of both spatial and temporal locations. The semantic level is concerned with how the various streams are configured and how they have to be delivered to the user. In particular, it describes the expected QoS requirements from the network.

To implement an efficient system for object-based MPEG-4 video transmission over IP networks with QoS management capabilities, it is necessary to distinguish between important AVO and less important one. The MPEG-4 AVOs are classified based on application-level QoS criteria and AVO's semantic descriptors according to AVO's descriptors and MPEG-7 framework [1]. The classification model leads to a relative priority score (RPS) for each AVO. The MPEG-4 AVOs requiring the same QoS performance (i.e., having the same RPS) from the network are automatically grouped and associated within one of the IP DiffServ PHB. Object data-packets within the same class are then multiplexed and transmitted over the selected transport layer with the corresponding bearer capability and priority levels. Thus, we propose to extend the MPEG-4 system architecture with a novel "cognitive layer." This layer implements the automatic content-based video object classification model. In our implementation, the "cognitive layer" makes use of a neural network classification model that is transparent to the video application and the network layers.

Fig. 2 depicts the enhanced layered MPEG-4 architecture. The classical MPEG-4 architecture is composed of three layers.

The "compression layer" is media aware but delivery unaware. The "sync layer" is media unaware and delivery unaware, and finally, the "delivery layer" which is media unaware but delivery aware. By media unaware, we mean that media characteristics are not used in the functionalities provided by this particular layer. We propose to include a "cognitive layer" which is media aware and delivery aware. This layer is introduced between the "sync layer" and the "delivery layer." The "cognitive layer" is aware of the transported media and highly aware of network QoS capabilities. The word "cognitive" refers to being aware of and making judgments about something. A "cognitive transport protocol" will be able to sense its surroundings (i.e., service requirements and underlying network conditions) and to identify the media type being delivered. Using this information, it will then be able to adapt, without user intervention, to its service's communication constraints.

Essentially, cognitive transport protocols will be able to create their own adaptable, on-the-fly traffic control level, based on the QoS requirements of the service at that moment. This fit well in the concept proposed in [24] "to build a fundamentally different sort of network—services and applications—that can assemble itself given high level instructions, reassemble itself as requirements change, automatically discover when something goes wrong, and automatically fix a detected problem or explain why we cannot do so." The interface between "sync layer" and "cognitive layer" is called "MPEG-4 AVO interface," and it performs a logical MPEG-4 object identification and retrieval.

The interface between "cognitive layer" and "delivery layer" is called "MPEG-4 AVO mapping interface," it is a logical interface at which the classified MPEG-4 AVO are mapped into various network QoS models such as IP DiffServ, IntServ, MPLS, etc., and where various functionalities can be performed such as video rate control and error protection.

As shown in Fig. 2, the cognitive middleware for MPEG-4 content delivery is build around several mechanisms and is composed of the following

- *A system-level audiovisual object classification model:* That implements an efficient transmission of object-based MPEG-4 video over IP networks with QoS management capabilities. The MPEG-4 AVOs are classified based on application-level QoS criteria and AVO's semantic descriptors according to MPEG-4 objects descriptors and MPEG-7 meta-data. This leads to a RPS for each video packet. The MPEG-4 AVO's (e.g., video elementary streams, video layers, video plans, audio, …) requiring the same QoS performance (same RPS) from the network are automatically classified and multiplexed within one of the available IP DiffServ PHBs. This classification mechanism provides automatic and accurate mapping between application-level QoS metrics and content relevancy to underlying QoS-capable network services.
- *A robust and adaptive application level framing protocol with data multiplexing and unequal error protection:* The application level framing (ALF) prepares the video stream to be transmitted over the selected transport layer. In case of IP network, RTP protocol is the most suitable protocol for ALF. The Audio/Video Transport Working

Group (AVT) was formed to specify a protocol for unicast and multicast real-time transmission over IP. RTP has to be customized to the transported media. Therefore, we propose a new RTP profile for MPEG-4 media that provides video object-based unequal error protection.
- *A fine grained TCP-Friendly video rate adaptation algorithm:* Video servers perform video source rate adaptation to tackle the varying network resource conditions. Based on end-to-end feedback measurements conveyed by real-time transport control protocol (RTCP) reports, the video source can estimate the allowed transmission rate and conform to it. The originality of this contribution is to finely adjust the source video rate according to the RPS of the AVOs calculated by the classification model.

The proposed MPEG-4 cognitive layer satisfies the following features.

- *Object-based abstraction:* the cognitive layer deals with MPEG-4 AVOs as a fundamental calculation entity. This allows a flexible, extensible, scalable and simple manipulation of the MPEG-4 scene.
- *Flexibility:* The flexibility of the proposed layer is achieved by: 1) allowing different functionalities and services to plug-in it (not only the one proposed in this section) and 2) enabling or disabling on the fly some of its functionalities.
- *Extensibility:* The cognitive layer can be used to derive new elements for different domains. As an example, a content examination can be used in video surveillance system to detect and track objects within the scene.
- *Scalability:* The cognitive layer augments the scalability of the system by the classifying media object into classes according to some criteria such as visual features (e.g., size and color), semantic relevance (e.g., relevance to user interest profile, background objects, and foreground objects), service quality (e.g., media features, bit-rate, and loss ratio), and/or temporal features. For example, a terminal in the MPEG-4 scene of Fig. 1 with limited capacity can choose to view a limited video sequence without logo, background, and other banners.
- *Simplicity:* The "cognitive layer" implements a simple classification algorithm based on media QoS requirements. Additional objects and features can be easily added in a modular and flexible way.
- *Application domain:* The Cognitive layer is generic and supports a broad range of applications and media transport mechanisms.

III. VIDEO OBJECT-BASED CLASSIFICATION MODEL

A. Background

Classification has been the subject of frequent and profound investigations. It has proved a useful tool in real-world applications. In networking, packet classification can be used in network elements such as edge routers for packet forwarding and filtering [2]. In [3], the authors present algorithms that can be used for packet classification and can be categorized as basic search algorithms, geometric algorithms, heuristic algorithms,

or hardware-specific search algorithms. These algorithms are used in IP framework such as firewalls and quality of service mechanisms.

In the machine-learning area, classification methods are largely used. One well-established approach is Bayesian classification, a technique that has become increasingly popular in the recent years in part due to recent developments in learning with Bayesian belief networks [4]. Another classification method based on similarity such as K-nearest neighbors (K-NN), Naïve–Bayes and hierarchical clustering are very used in machine learning.

In the area of classification, neural networks solve many problems that conventional above methods cannot, or at least not within acceptable cost or performance constraints. In this paper, we have used neural network algorithms for automatic AVO classification.

In order to benefit from object-based compression, we propose to classify the MPEG-4 AVOs at the video server from most important AVO to least important AVO. We deal with the AVO as an independent calculation primitive. Several methods can be used for AVO's 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 objects descriptors and/or MPEG-7 [1] can provide the relevant information needed to compute the RPS for each objects.

The MPEG-7 standard describes generic description schemes (DSs) for image, video, multimedia, home media, and archive content. MPEG-7 aims to create a multimedia content description standard in order to facilitate various multimedia searching and filtering applications. We can use this content description to perform an intelligent classification of AVOs. The main components of the image, video, and multimedia DSs are objects, feature classification, object hierarchy, entity-relation graph, code downloading, multiabstraction levels, and modality transcoding. Each AVO may have one or more associated features, which are grouped in the following categories: media features, visual features, temporal features, and semantic features. Each feature is described by a set of descriptors.

The user can interact with the MPEG-4 server and decide at any time to choose some AVOs among several others available in the scene. This is the basic type of classification. Automatic classification is performed at the server by a prioritization mechanism which assigns a relative priority score (RPS) to each AVO. High RPS values (high priorities) are assigned to the relevant AVOs in the scene (e.g., base layer stream in hierarchical coding or foreground objects of the scene) and low RPS value are assigned to the less important AVOs (e.g., enhancement layers stream or background objects).

B. Audio Visual Object (AVO) Classification Model

We suppose that we have a collection of MPEG-4 AVOs that must be grouped into classes. The number of classes is predefined and each class has well-known characteristics. A class is viewed as a class of service of a network layer. It can be an IP DiffServ class (best effort, EF, or AFx class), an ATM class of service (ABR, VBR, UBR, etc.), or any others network service abstraction that delivers QoS.

The MPEG 4 video coding standard provides an object-based representation of the video scene by allowing separate coding of AVOs. Texture and shape coding in MPEG-4 is very similar to the coding of frames in MPEG-2. Temporal instance of a video object is called video object plane (VOP). VOP is divided into macro, luminance, and chrominance blocks. VOP supports intracoded (I-VOP) temporally predicted (P-VOP) and bidirectionally predicted (B-VOP) frames. The different components contained in the video data stream do not have the same impact on the quality of the decoded video. The damage caused by some data loss in a reference picture (I-VOP or P-VOP) will affect subsequent picture(s) due to interframe predictions. Subsequently, I-frames must be protected more than P-frames and P-frames more than B-frames. Let us consider the example of video object coded using layered wavelet transform techniques. The most important layer contains the low-frequency subband 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* (ELs).

This is a basic classification model for preparing the MPEG-4 access unit (AU) to be transmitted over the network. It operates within a single audiovisual object. A more complete classification is handled by the cognitive layer which implements a classification model. The result of the classification is a set of AVOs sorted according to their importance in the scene. The classification model assign a final RPS to each AU to apply different mechanisms such as service differentiation, video rate control or unequal error protection. This priority score reflects both the priority of a particular AVO in the scene and the priority of a single frame type (I, P, B, or hierarchical stream if any BL or EL). In the rest of this section, we will focus on the prototype implementation using a neural network algorithm.

Let $T = \{(X, c)\}$ be a training set of N labeled vectors, where $X \in \Re^n$ is a feature vector $X = (x_1, x_2, \ldots, x_n)^t$ and $c \in T$ is its class label from an index set T (X is an AVO, and T is the set of available classes, e.g., DiffServ PHB). The variables x_i are referred to as attributes (e.g., QoS features of each AVO). A class is modeled by one or more prototype, which has $\{U_1, U_2, \ldots U_m\}$ as features.

A classifier is a mapping function called C defined as $C: R^n \to T$, which assigns a class label in T to each vector in R^n (a vector is an MPEG-4 AVO features). Typically, the classifier is represented by a set of model parameters $\Lambda = \{\Delta_k\}$. The classifier specifies a partitioning of the feature space into regions $R_j = \{X \in R^n : C(X) = j\}$, where $\bigcup_j R_j \equiv R^n$ and $\bigcap_j R_j \equiv \Phi$.

It also induces a corresponding partitioning of the training set into subset T.

There are different methods based on this definition, which allow an automatic classification of vectors. The classification model used in our cognitive layer is based on radial basis function (RBF) classification.

Fig. 3 shows the RBF classifier. A vector to be classified is passed to a set of basis functions, each returning one scale value φ_j $j = 1, 2, \ldots, l$. The concrete choice of the basis function is not critical; the common implementations prefer the radial basis
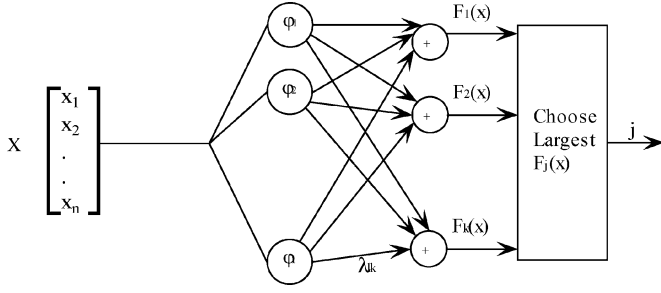
Fig. 3. RBF classification model.

function (the Gaussian "bell" functions). Hence, the name: RBF network, with: $\varphi_j(X) = e^{-(D^2/2\sigma_j^2)}$

In RBF, $D = \|X - U_k\|^2$ is the distance between X and U_k. $\varphi j(x)$ measures similarity, i.e., the mapping function. λjk in the Fig. 3. is a set of scalar weights that connect each of the receptive fields to the class outputs of the network. The general equation of an output of the neuron j is given by: $F_k(X) = \sum_{j=1}^{n} \lambda_{jk}\varphi_j(X)$

The classifier maps the vector X (i.e., an AVO) to the class with the largest output: $R_j \equiv \{x \in R^n / F_j(x) \geq F_k(x) \forall k\}$

A neural network takes as input the feature vector X, produces computing class outputs $\{F_j(x)\}$, and then, classification decisions are made based on the largest output. The classification cannot be accomplished without knowing the characteristics of the underlying transport mechanisms.

IV. AVO PROTECTION THROUGH UNEQUAL ERROR PROTECTION

The proposed "cognitive layer" integrates an unequal error protection that is based on the above AVO classification model [25]. Error resilience of each *Elementary Stream* associated to one AVO can be enhanced when sensitive data are protected, whereas the subjectively less relevant audiovisual information are none or less protected during transmission, as investigated in, [5]–[9]. The IETF work in progress described in [10] and [11] specify how error protection is unequally applied to different parts of the video stream. We adopt this approach with adaptation to object based video coding streams (i.e., MPEG-4). Consequently, the classification model specifies how to assign priority score to each AU within an AVO.

A. Reed–Solomon (RS) Codes

Reed–Solomon (RS) codes aims 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 [12]. RS code is called an (n, k) code. RS code (n, k) is defined over the Galois Field $\mathbf{GF}(2^q)$, 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 $x = x_0 \ldots x_{k-1}$ be the source data, G an $(n \times k)$ generator matrix of the (n, k) RS code, and y the encoded data. Then, y is given by

$$y = G \cdot x. \qquad (1)$$

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 (2).

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

$$G = \begin{bmatrix} 1 & 0 & \cdots & 1 \\ 0 & 1 & \ddots & \vdots \\ \vdots & & & \vdots \\ \vdots & & \ddots & 1 & 0 \\ 0 & \cdots & 0 & 1 \\ (n-1)^1 & (n-2)^1 & \cdots & (n-h)^1 \\ (n-1)^2 & (n-2)^2 & \cdots & (n-h)^2 \\ \vdots & \cdots & \cdots & \vdots \\ (n-1)^h & (n-2)^h & \cdots & (n-h)^h \end{bmatrix}. \qquad (2)$$

B. Object-Based Unequal Error Protection Using Adaptive RS Codes

We proposed an object-based unequal error protection (UEP) for handling MPEG-4 audiovisual stream according to its importance. Let us consider U_i, the ith Access Unit in the AVO of priority score p [25]. The main challenge of the proposed object-based UEP is to determine the values n_i and k_i of RS code 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 an access unit U_i is divided when no error protection is performed. The value n_i depends on the priority score of a particular AVO. It depends also on the length m_i of U_i so that the traffic overhead introduced by redundant data does not become excessive.

Once the efficient (n_i, k_i) RS code is found the transmission process starts. Data of U_i is placed in k_i horizontal packets $(S_1, S_2 \ldots S_k)$. Each of them has the same packet size. Padding is added to the last packet if necessary. Then, the (n_i, k_i) RS code is applied across these packets, vertically as introduced in [13]. So, redundant data for each AVO is calculated by $h_i = n_i - k_i$. The RS code for whole AVO's objects is represented as $(n_1, k_1), (n_2, k_2), \ldots, (n_L, k_L)$, where $(n_1 - k_1) > (n_2, k_2) > \cdots > (n_L, k_L)$. This means that $h_1 > h_2 > \cdots > h_L$, where L is the number of AVO's object present in the scene or to be transmitted in the network when the TCP-Friendly adaptation is performed.

It is clear that the proposed object-based UEP increases the amount of the traffic sent in the network. Traffic overhead is defined by the accumulated number of data-bytes in each AVO. Each AVO j with a priority score p generates traffic overhead R_{jp} given by (3) during the streaming interval (the streaming interval for the particular object is defined in the next
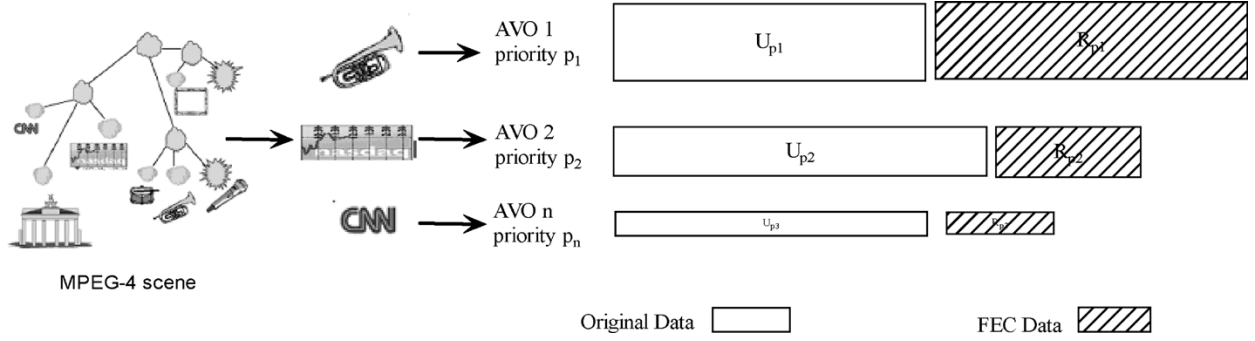
Fig. 4.   RS codes for object-based unequal error protection.

section in order to comply with the TCP-Friendly adaptation requirements)

$$R_{jp} = \sum_{i=k_p+1}^{n_p} \frac{U_{ijp}}{k_{ijp}}. \qquad (3)$$

Then, the accumulated overhead of the MPEG-4 scene can be denoted as

$$R(\text{scene}) = \sum_{j=1}^{L} \sum_{i=k_p+1}^{n_p} \frac{U_{ijp}}{k_{ijp}}. \qquad (4)$$

Fig. 4 shows how RS code could be applied to MPEG-4 scene composed of different AVOs with different priority.

We assume that for each AVO j with a priority score $\boldsymbol{p}$, error protection byte-budget B_{jp} is reserved for error protection. This byte-budget is fixed with a ratio r among the useful data U_{jp}. Then, B_{jp} can be defined as

$$B_{jp} = U_{jp} * r \qquad (5)$$

and the accumulated error protection byte-budget for the MPEG-4 scene is given by (6)

$$B(\text{scene}) = \sum_{j=1}^{L} U_{jp} * r. \qquad (6)$$

For example, with an MPEG-4 AVO of priority score $p = 1$, we can choose the ratio $r = 0.1$ then the error protection byte-budget will be limited to 10% of the useful data of this object.

In order to find the efficient value of $\boldsymbol{n_i}$ and $\boldsymbol{k_i}$, we must satisfy two conditions. First, the traffic overhead R_{jp} is upper-bounded by the error protection byte-budget B_{jp}. The second is that k_i cannot be greater that n_i for a particular AVO. Combining (4) with the above two conditions result in a constrained optimization problem given as follows:

$$\textbf{Minimize}: \sum_{j=1}^{L} \left| U_{jp} * r - \sum_{i=k_p+1}^{n_p} \frac{U_{ijp}}{k_{ijp}} \right|$$

for each AVO j of a priority score p and with $k_i \le n_i$.

$$\qquad (7)$$

With the proposed object-based unequal error protection mechanism, the RS code evolves dynamically so that the network bandwidth is correctly controlled according to the network conditions.

V. FINE-GRAINED TCP-FRIENDLY VIDEO RATE ADAPTATION ALGORITHM

The majority of multimedia applications perform over an RTP/UDP/IP stack. However, UDP offers no congestion control mechanism and, therefore, is unaware of network condition and unfair toward other competing traffic. Today's Internet traffic is dominated by transmission control protocol (TCP). TCP uses several mechanisms to handle network congestion such as: additive increase and multiplicative decrease (AIMD), slow start, congestion avoidance, fast retransmit, and fast recovery. Thus, it is crucial that user datagram protocol (UDP) traffic performs also TCP-Friendly congestion control [15].

Congestion control helps preventing the application entering congestion collapse in which traffic demand is high but little useful throughput is available. Such situation can be prevented if all applications implement a TCP-like congestion control mechanism. Traffic that does not perform in a TCP-Friendly manner can be dropped by the router [16].

In our congestion control [26], the server performs rate adaptation through the adjustment of the number of streamed objects based on network state and RPS of each objects. We use a TCP-Friendly approach to adapt the server rate to network condition. The server attempts to deliver the maximum number of AVO that can fit in the current available bandwidth slot. This flow control mechanism is combined with a DiffServ marker. The server must be aware of each audiovisual object in the scene so as to classify these objects in a hierarchical manner, from less important object to more important one. As presented in Section III, it allows the server to: 1) deal with network congestion by stopping the streaming of less important objects when congestion is detected and 2) prioritize the transport of important objects through a content-aware IP packet marking scheme in IP DiffServ networks. When network congestion occurs less important AVOs will be dropped automatically by network elements. Lost packets notify the server to reduce its transmission rate by stopping streaming less important AVOs.

The idea of TCP-Friendly transport protocol is to emulate TCP behavior without replicating the TCP mechanism itself. By definition, a flow is said to be TCP-Friendly or TCP-compatible, if its arrival rate does not exceed the arrival rate of a conformant TCP implementation in the same circumstances [17]. Many TCP-Friendly congestion control mechanisms were developed recently that are either window-based or equation-based including: rate adaptation protocol (RAP) [19], loss-delay

based adaptation algorithm (LDP) [18], a spectrum of TCP-Friendly algorithm [20] and TCP-Friendly rate control protocol (TFRC) [21]. While these protocols and others are comparable in their features of simulating TCP behavior, TFRC seems to be more robust and was recently adopted by the IETF. TFRC provides sufficient responsiveness by taking into consideration all the parameters that affect the TCP rate such as loss, round-trip time (RTT) and retransmission timeout value. The key advantage of TFRC is that it has a more stable rate during the session lifetime.

Our video quality adaptation mechanism is based on TFRC. 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 and RTCP protocols [14]. Each RTP packet has a timestamp and a sequence number that allow the receiver to compute the packet loss rate and the sender to compute the RTT.
- The loss rate and the RTT are then fed into the TFRC module to determine the appropriate transmission rate [see (8)].
- The sender then adds or drops audiovisual objects and the associated layers if any, to adjust its transmission rate to match the target rate (i.e., allowed rate).

The target transmission rate is computed by using the following TFRC equation [21]:

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

Where R_{TCP} 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 acknowledgment.

A. Video Object-Based Rate Adaptation Algorithm

Let S be a set of MPEG-4 AVOs containing n AVO's O_j, with $j \in \{1, 2 \ldots n\}$. Without loss of generality, we assume that these objects are sorted in a decreasing order of priority score carried-out through classification. Each object O_j may consist of m_j layers $(m_j = 1)$. Note that lower layers within an object have higher priorities than higher layers.

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

$$\forall j, 1 \leq j < n : P(O_{j+1}) \leq P(O_j)$$
$$\forall j, 1 \leq j < n, \forall l, 1 \leq l < m_j : P(L_{j,l+1}) < P(L_{j,l})$$
$$L_{j,l} \text{ is the Layer number } l \text{ of the Object } O_j. \quad (9)$$

Using (9), we can construct an audioVisual Entity set called E composed of all object layers ordered according to their respective priorities

$$E = \{L_{1,1}, L_{1,2} \ldots L_{1,m1}, L_{2,1}, L_{2,2}$$
$$\ldots L_{2,m2}, \ldots, L_{n,1}, L_{n,2} \ldots L_{n,mn}\}.$$

We will note E as follows:

$$E = \{e_1, e_2, \ldots, e_W\} \text{ with } w = |E| = \sum_{j=1}^{n} m_j.$$

Our object-based video rate 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 R_{TCP} using (8). The server tries to send as many audio visual entities as possible without exceeding R_{TCP} taking into consideration entity's priorities. Details of the adding and the dropping processes will be presented in Sections V.B and V.C respectively.

B. Adding Audiovisual Objects

The server adds a new audiovisual 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. Assume also that the client has sufficient resources to play all the entities being sent by the server. At time t_{i+1} the server can add a new entity while the following condition remains satisfied

$$\sum_{j=1}^{k+1} R_{i+1}(e_j) \leq R_{\text{TCP}}. \quad (10)$$

At the client side, the new audiovisual entity must be buffered and synchronized to the current playback time.

C. Dropping Audiovisual Objects

When the estimated throughput of the TCP session indicates that the video server is transmitting more data than it should, the later must reduce its sending rate by dropping one or more audiovisual entities. The server drops entities as long as the following condition is satisfied:

$$\sum_{j=1}^{k} R_{i+1}(e_j) > R_{\text{TCP}}. \quad (11)$$

D. GOV-Driven Stability

Since the TFRC compute the new target rate each RTT, adding and dropping audiovisual entities can lead to undesired bit rate oscillation and consequently video quality degradation at the receiver. To prevent from such behavior, several measures are taken.

First, the TFRC module reduces bit rate oscillations through exponentially weighted moving average (EWMA) to detect out-of-control states. EWMA statistics are used to respond dynamically to the changing values in the measured RTT and loss and regulate these values to reflect as much as possible the real network conditions. In TFRC, the loss rate is measured in terms of loss interval which represents the number between two consecutive loss events [20]. The mechanism reacts too strongly to single loss events and ensures that allowed sending rate do not change aggressively.

Second, we propose to synchronize server transmission rate changes with beginning of the group of VOP (GOV). Thus, the
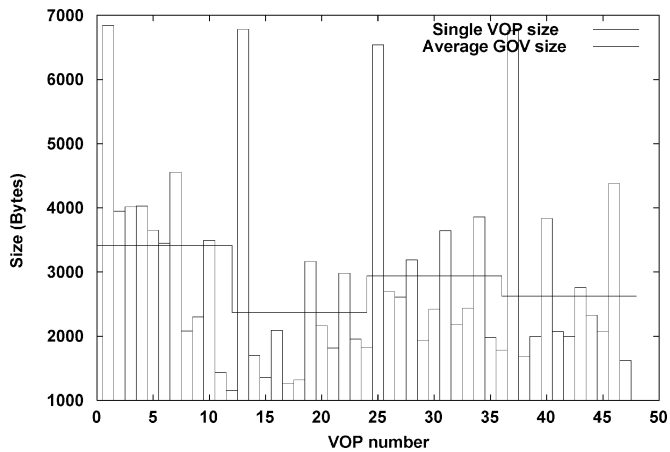
Fig. 5.   Handling stability through VOP control.

new transmission rate obtained from TFRC module is used to intelligently adapt video sending rate to conform to media object encoding dependencies. Fig. 5 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.

E. Rate Adaptation and Admission Control (AC) Issues

Admission control (AC) is used to control the network load by restricting access to the network and, hence improving the level of QoS guarantee. Admission control approaches can be categorized in a number of ways such as parameter-based approaches versus measurement-based approaches and edge/end-point admission control versus hop-by-hop admission control.

Parameter-based approaches (as in [33]) assume some traffic pattern and try to maintain the aggregated resource consumption below the total capacity. They often lead to conservative resource allocation and low network utilization especially in the case of bursty traffic. In turn, measurement-based admission control relies on the measurement of current network load and, therefore, responds faster to the network status and consequently improves the network utilization [28]–[33].

As being done in the IntServ/RSVP architecture, admission control is traditionally performed on a hop-by-hop basis. Each intermediate network element along the path has to decide whether the new request can be accommodated or not and reserves resources accordingly. However, adding admission control functionality to all the core elements violates the DiffServ principle of leaving the core simple. End-point/edge admission control that pushes the admission control functionality to the edge of the network seems more suitable in a DiffServ environment and has a number of advantages over the hop-by-hop approach such as faster response time and less implementation overhead. Studies also show that simple admission control algorithms based on estimated or measured network status are generally robust [27], [28]. Most of the approaches in this category use probing [29], [32], [33] or explicit congestion notification (ECN) [27], [29] to convey the network status back to the end points. Admission decision is then made based on this feedback information. For example, in [30], state

information is embedded inside the packet header (dynamic packet state) so that admission control decisions can be made based on this information. Our proposed scheme can be used in conjunction with most of the admission control approaches proposed for DiffServ.

VI. ADAPTIVE CROSS-LAYER VIDEO STREAMING SYSTEM

Fig. 6 depicts the general block diagram of the proposed adapative cross layer video streaming system. It is composed of a video server and a video client. The server streams the audiovisual object to the client via an IP DiffServ network using the RTP protocol. The client decodes and composes the original MPEG-4 scene. As shown in Fig. 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 end-to-end feedback algorithm which is based on RTP and RTCP. This feedback information is sent to the "add/drop module," which adapts the video transmission rate to conform to the available bandwidth. The IP DiffServ marker module handles the marking of the different RTP packet with DiffServ code point before entering the DiffServ network. The higher is the packet importance, the lower is the DiffServ drop precedence assigned by marker to this packet.

DiffServ object prioritization aims to favorite the transport of some AVOs compared with others. When network congestion occurs, less important AVO's streams are dropped automatically by the active queue implemented in the DiffServ router. Recall that an MPEG-4 scene contains several MPEG-4 AVOs grouped according to their importance in the presentation. 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, and so on.

VII. PERFORMANCE EVALUATION

A. System and Network Models

Simulations are conduced using the network simulator *ns2*. We used the network architecture shown in Fig. 7 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" using TFRC module to adapt the number of transmitted AVOs. The client is also an ns2 agent which extends the capabilities of the RTP sink by reporting statistics 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 congested differently. FTP sources always have a packet to send and always send a maximum-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. We configure two scenarios:1) Scenario A: four FTP sources and (2) Scenario B: eight FTP sources. FTP sources send data from time $t = 30$ s until $t = 90$ s. 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
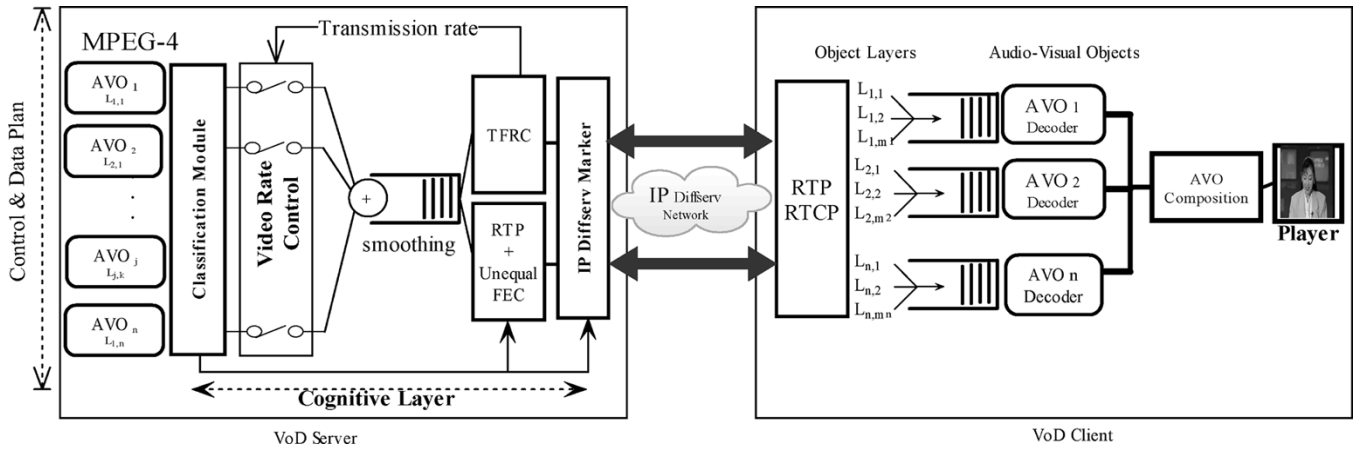
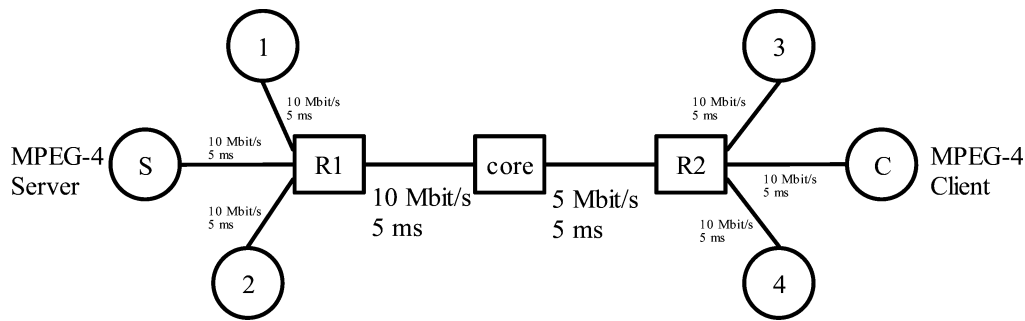Fig. 6. General block diagram of the proposed cross-layer video streaming system.

Fig. 7. Network topology for congestion control.

on incoming packets. R1 uses a two-rate three-color marker (TR3CM) [22] to mark the background. Therefore, background traffic is evenly distributed among the different DiffServ classes. Recall that the video traffic is marked at the MPEG-4 server according to AVO's priorities. The bottleneck link between the core router and R2 has a capacity of 5 Mb/s.

Some of the metrics used to setup the DiffServ domain are taken from [23] and are listed in Table I. These metrics are: 1) Priority: our configuration of the priority score of the DiffServ classes (higher value corresponds to higher priority); 2) IPTD: IP packet transfer delay; 3) IPDV: IP packet delay variation; 4) IPLR: IP packet loss ratio; 5) IPER: IP packet error ratio; and 6) BW: Bandwidth used. These values are computed statistically among the simulated DiffServ domain.

In our simulation, the MPEG-4 presentation was obtained using a set of AVO's components. We simulate the "Akiyo" video sequence shown in Fig. 8 by using four multimedia objects: AO (audio speech), VO1 (background), VO1 (speaker), and VO3 (logo). These objects are sorted as follows.

Fig. 9 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 s. The complete scene is shown in Fig. 9(a). The audio object is an adaptive audio coding (AAC) MPEG-4 audio with a constant bit rate throughput of 64 Kb/s. An audio packet is sent each 125 ms. Video object 1 (the background object) has an average throughput of 200 Kb/s and a peak rate of 956 Kb/s. This object is composed of three Layers: base layer (BL), enhancement layer 1 (EL1), and enhancement layer 2 (EL2). The throughputs of the different layers are shown in

TABLE I
IP DiffServ QoS Class Definitions

Network Performance Parameter	QoS Classes				
	EF	AF11	AF12	AF13	Best Effort
Priority	4	3	2	1	0
IPTD	100ms	150ms	200ms	300ms	Unspecified
IPDV	50ms	50ms	Unspecified	Unspecified	Unspecified
IPLR	$1*10^{-3}$	$1*10^{-3}$	$2*10^{-3}$	$3*10^{-3}$	Unspecified
IPER	$1*10^{-4}$				Unspecified
BW	1,5Mbits	1,5Mbits	1,5Mbits	1,5Mbits	4Mbits

Fig. 9(b). Video object 2 (the speaker object) has an average throughput of 650 Kb/s and a peak rate of 1722 Kb/s. This object is composed also of three layers: BL, EL1, and EL2. The throughputs of the different layers are shown in Fig. 9(c). Video object 3 (the logo object) has an average throughput of 124 Kb/s and a peak rate of 356 Kb/s. It is composed of one single layer [see Fig. 9(c)]. Table II summarizes the application level requirement for each MPEG-4 AVO.

B. Results Analysis

Intensive simulations are performed, each time with different parameters, to see the behavior of our cross-layer video streaming system.

First, we run our classification algorithm with the parameters described in the previous section, and we compute the output of each neuron. The system chooses the class of service with the largest neuron output which reflects the high similarity. We also choose the value of $\lambda_{1k} = \lambda_{2k} = \lambda_{3k} = 1$
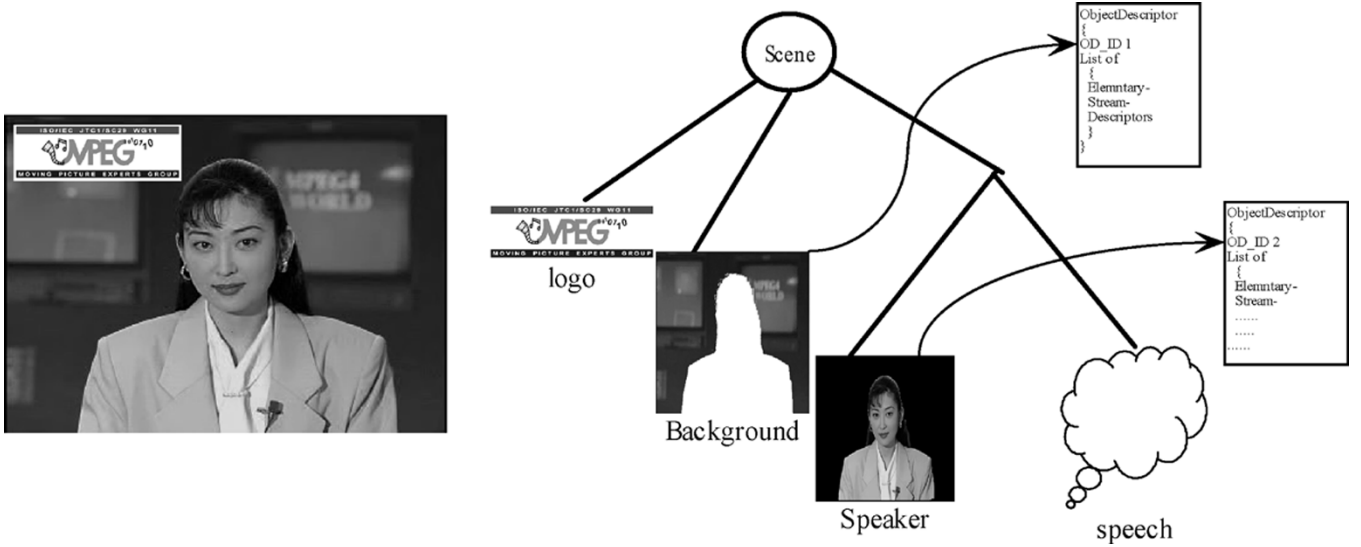
Fig. 8.   Simple composite MPEG-4 scene based on "Akiyo" video sequence.

which connects each feature from the vector to the network. The feature vector of each MPEG-4 AVO is described as follows: $X = (MAX\_DELAY, MAX\_JITTER, LOSS\_PROB)^T$ with $X \in \Re^3$, and we have taken the value of

$$MAX\_JITTER =$$
$$MAX\_DELAY - PREF\_MAX\_DELAY.$$

Output result of the RBF network (i.e., $F_k(X) = \sum_{j=1}^{n} \lambda_{jk} \varphi_j(X)$) is summarized in Table III. As previously explained, we select the largest output from the network. According to these results, we can see that O1 is marked with AF11 PHB and that O2, O3 are both marked with AF12 PHB. We have also measured the execution time of the algorithm and found to be about 30 ms of the mapping of 100 AVOs is about 30. The complexity of this algorithm is $O(n)^2$.

According to these results, the audio will be marked with EF PHB and has the priority score of 4. The background, and the speaker will be marked with AF11 PHB and have the priority score of 3. Finally, the logo will be marked with AF13 PHB and has the priority score of 2.

The rest of the section presents some QoS measurement namely, the video server throughput as a function of network state, packet loss, and user perceived quality.

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

Scenario A (i.e., MPEG-4 scene with four FTP streams) is interesting since we see the effect of our adaptation mechanism.

In Fig. 10, 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 most important audiovisual entity to least important one. We can see that all the streams (FTP and video) fairly share the bandwidth. The logo object is not transmitted every time, because it is the less important object. When there is a lack of bandwidth in the link, the server stop streaming lower priority objects.

Scenario B (i.e., MPEG-4 scene with eight FTP streams) confirms the previous result and shows the effect of our adaptation mechanism as presented in Fig. 11. A minimum of QoS is guaranteed by our adaptation mechanism. The network does not enter in congestion since the video server is aware of network condition. Some snapshots of the real video are presented later.

2) Packet Loss and Unequal FEC Performance: Fig. 12 shows packet loss for scenarios A and B, using our congestion control scheme. In scenario A, we observe some loss of lower priority packets but in scenario B high priority packet are also dropped. This is due to: 1) lower priority packets are not transmitted because our congestion control mechanism regulates the server transmission rate by stopping the streaming of lower priority packets and 2) AVO1 and AVO2 require more bandwidth in our scene and cause congestion.

FTP packet loss observed in the same scenarios is also presented in Fig. 12. FTP packets encounter more loss than the video packets due to two factors. First factor is that FTP traffic is marked using TR3CM marker which distributes the marked traffic among the different classes of the DiffServ network. It is worth noting that the majority of dropped packets are those marked with high drop precedence. Second factor is that FTP source does not regulate the traffic by computing the allowed transmission rate rather it uses window-based congestion control mechanism. In case of video traffic, the damage caused by some data loss in some reference picture such as I-VOP or P-VOP will affect subsequent picture(s) due to inter-frame predictions. For example, when the I-VOP is lost, the whole
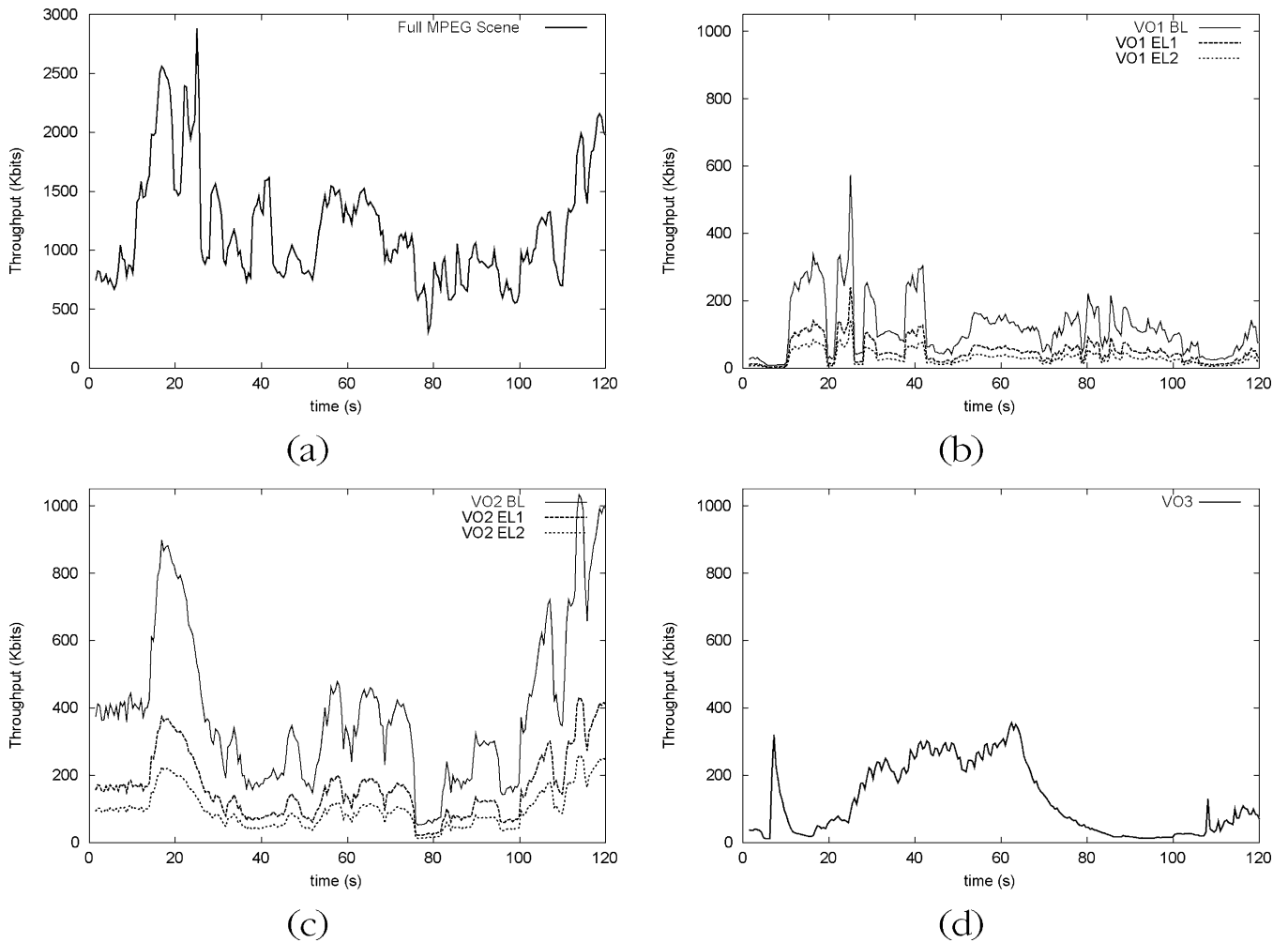
Fig. 9.   Instantaneous throughput of the different MPEG-4 video objects.

TABLE II
APPLICATION-LEVEL QoS REQUIREMENTS SPECIFIED
IN MPEG-4 QoS DESCRIPTORS

QoS Metric	Synthetic Description	Audio Object	Video Object-1	Video Object-2	Video Object-3
MAX_DELAY	Absolute maximum end to end delay for the stream	150ms	250ms	250ms	350ms
PREF_MAX_DELAY	Preferred maximum end to end delay for the stream	100ms	200ms	200ms	300ms
LOSS_PROB	Allowable loss probability of any single AU.	$1*10^{-3}$	$2*10^{-3}$	$2*10^{-3}$	$3*10^{-3}$
MAX_GAP_LOSS	Maximum allowable number of consecutively lost AUs	Unspecified	Unspecified	Unspecified	Unspecified
MAX_AU_SIZE	Maximum size of an AU	342 bytes	1024bytes	1024bytes	1024bytes
AVG_AU_SIZE	Average size of an AU	342 bytes	1024bytes	1024bytes	1024bytes
MAX_AU_RATE	Maximum arrival rate of AUs	Unspecified	Unspecified	Unspecified	Unspecified
PRIORITY	Priority for the stream	Unspecified	Unspecified	Unspecified	Unspecified

TABLE III
OUTPUTS FROM THE RBF CLASSIFICATION MODEL

	EF	AF11	AF12	AF13	Best
audio	3.0	2.9962	2.9984	2.9943	2.8143
background	2.9962	2.9999	2.9995	2.9998	2.8248
speaker	2.9962	2.9999	2.9995	2.9998	2.8248
Logo	2.9927	2.9994	2.9978	2.9999	2.8273

TABLE IV
TRANSMISSION BIT-RATE RATIO PER MPEG-4 OBJECTS

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

dependant P-VOP and B-VOP cannot be decoded. The same conclusion is valid for hierarchical streams. Hence, enhancement layer 1 cannot be decoded without the reception of base layer, and so on. When using our congestion control mechanism, lower priority audiovisual entities (those marked with high drop precedence) are not transmitted by the server when the allowed

rate decreases. This helps to prevent a future drop by the router. So, the regulation is done at the server and demonstrates clearly the advantage of our mechanism.

In order to highlight the efficiency of the unequal error protection (UEP) scheme, we compute the number of AUs
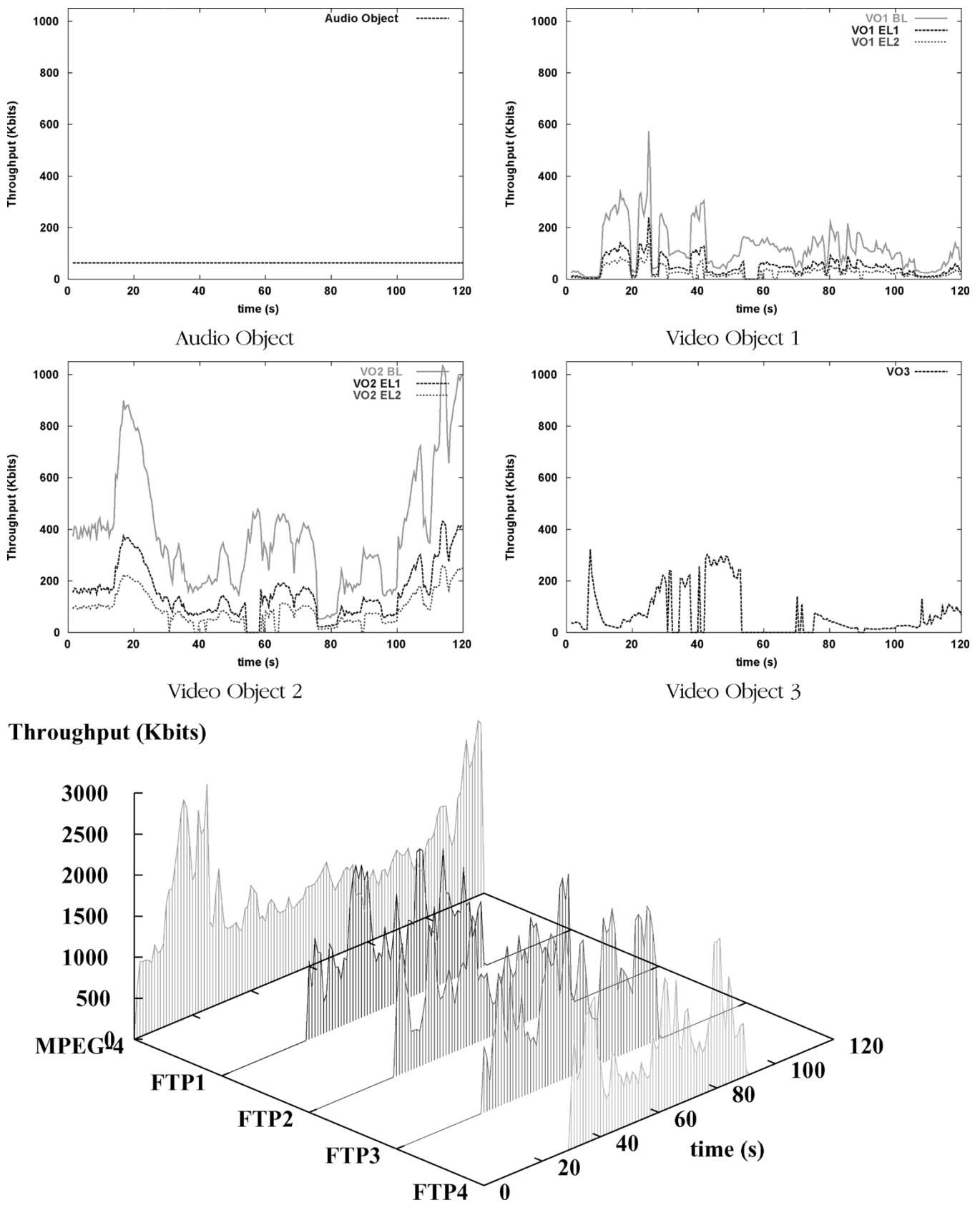
Fig. 10.   Traffic throughput and fairness in scenario A.

that can be decoded at the client. Fig. 13 shows the comparison between the decoded object ratios. The X axis represents the throughput of the background traffic, and Y axis represent decoded object ratio. As expected, the quantity of AVOs decoded at the receiver side decreases when the network load increases because it entails more packet losses.
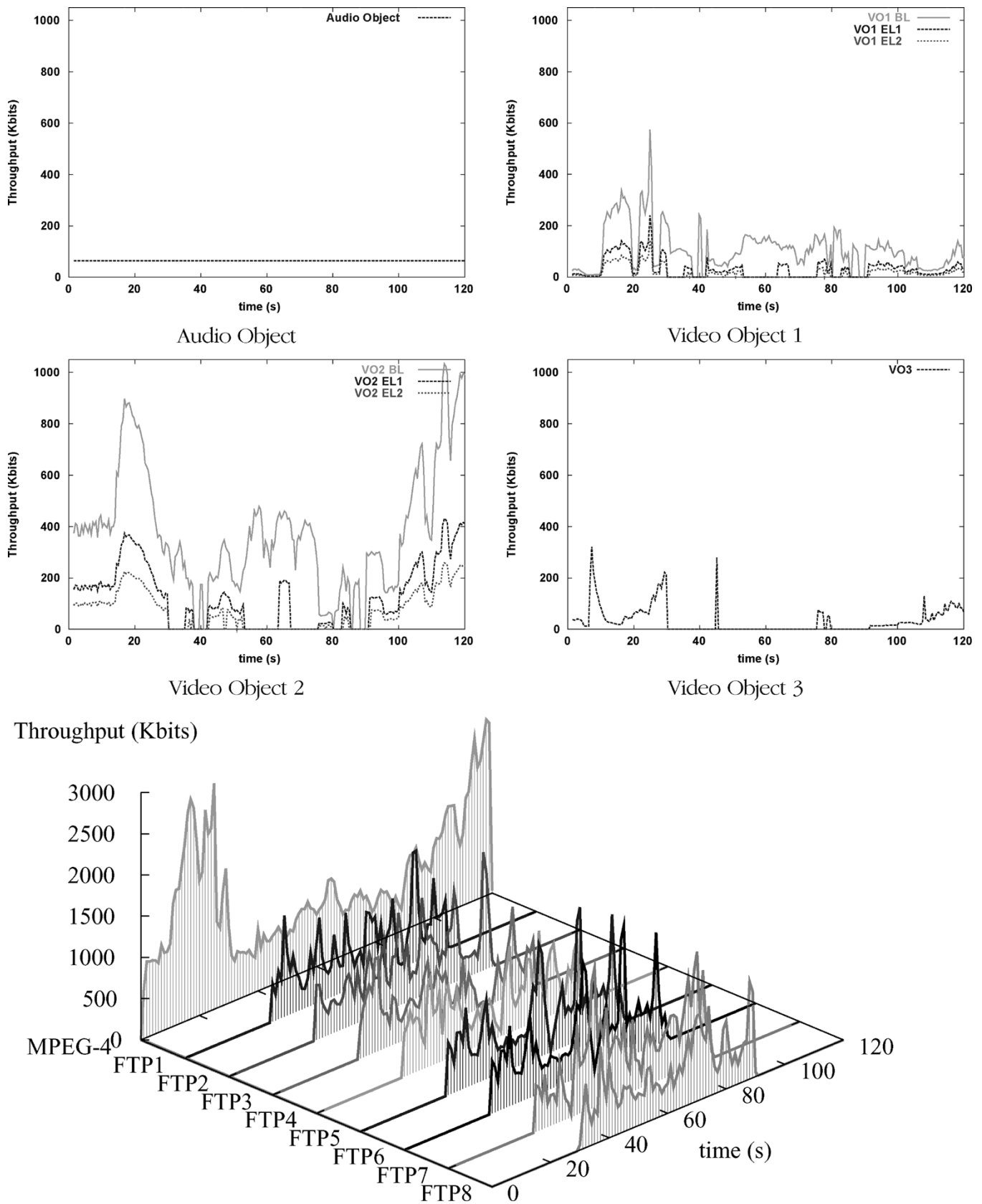
Fig. 11.   Traffic throughput and fairness in scenario B.

By the use of the FEC-based UEP, packet loss rises due to an increase of the MPEG-4 packet-stream throughput by 7%. Consequently, we notice a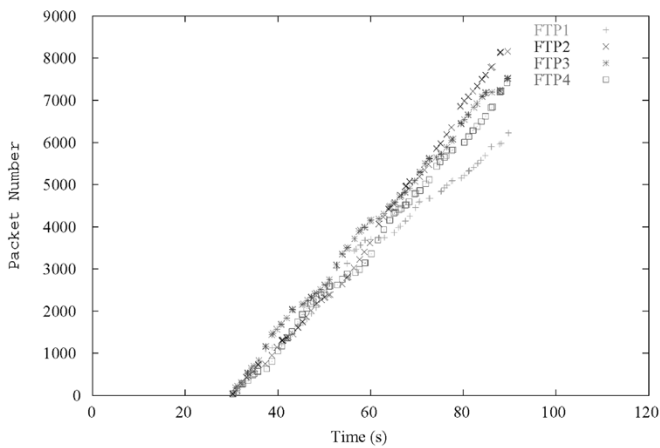 slight higher packet losses with UEP configurations, for a given network load. However, the redundant UEP information better recovers lost packet at the receiver. Consequently, some *access unit* can be restored correctly.
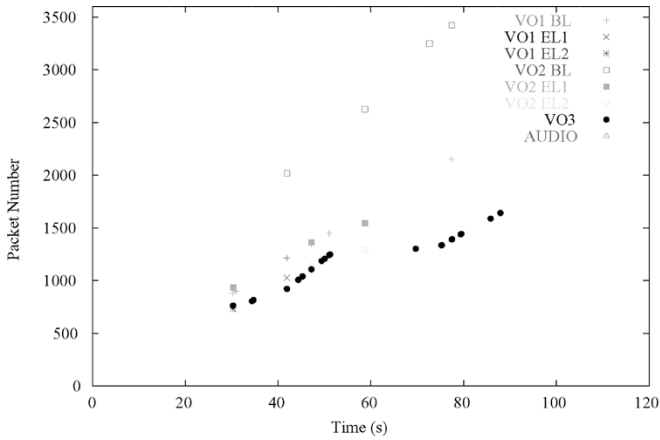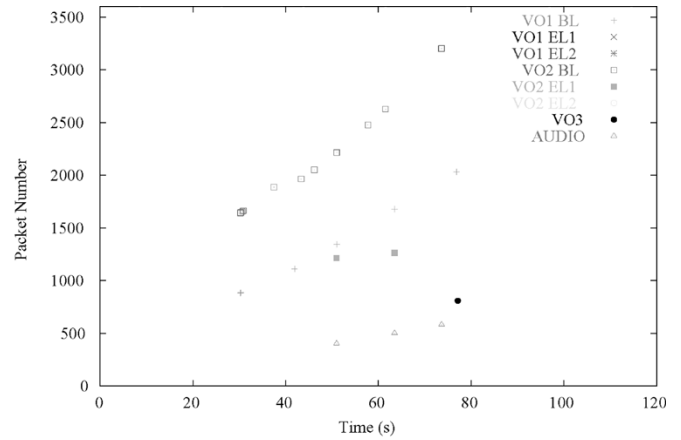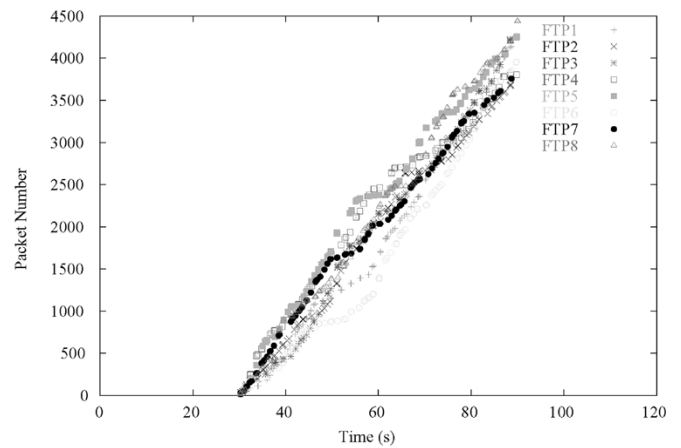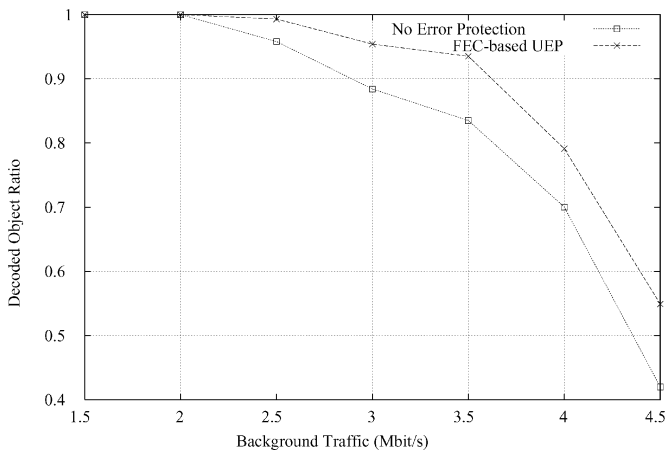
Fig. 12.　Instantaneous packet loss.

Fig. 13.　Correctly decoded AV object ratio versus background traffic throughput.

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.

3) User Perceived Quality: Peak signal-to-noise ratio (PSNR) is a commonly used for measuring picture quality degradation. It is derived from the root mean squared error. The PSNR for a degraded $N1 \times N2$ 8-bit image f' from the original image f is computed according to the formula in (12)

$$\text{PSNR} = 20 \log_{10} \frac{255}{\sqrt{\frac{1}{N_1 N_2} \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} [f(x,y) - f'(x,y)]^2}}. \quad (12)$$

In order to compute the PSNR value, we should rebuild the received scene. By using ns2 trace file, we create the received MPEG-4 scene. The measured PSNR indicates the difference between the original and the received video sequences. Fig. 14 shows comparison between the original and the received scene quality for scenarios (A and B). In scenarios A and B, the degradation of the received quality is due to our adaptive delivery system which only sends pertinent audiovisual objects to the client. This PSNR measurement does not reflect the semantic of what the client received. Because in our case, the logo object is basically not sent. This affects the PSNR value but has no importance for the end user (in our assumption). Fig. 15 presents snapshots of the received video in scenario B.
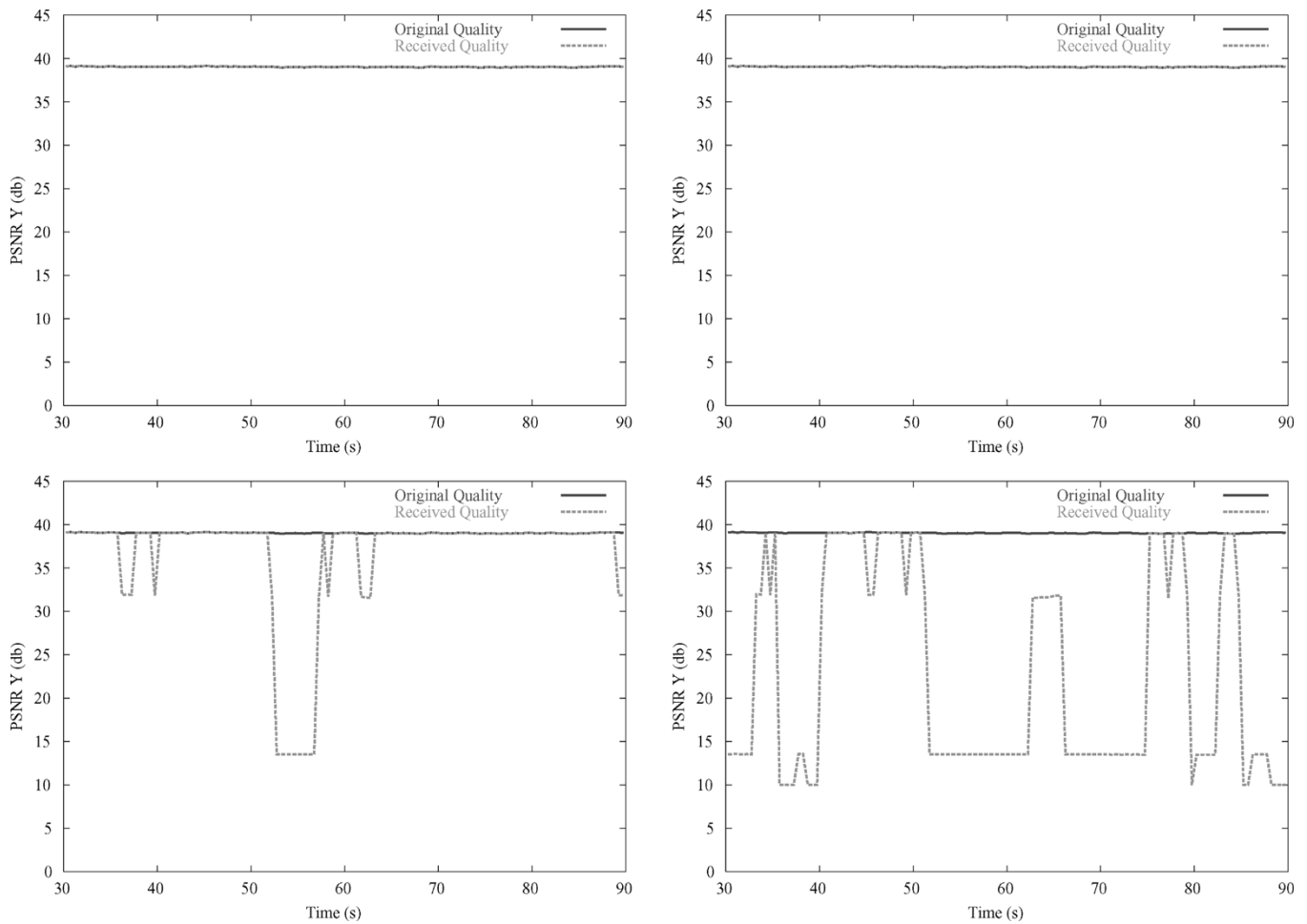
Fig. 14.   Perceived video quality (PSNR) measurements for the different scenario.

Fig. 15.   Received video presentation.

VIII. CONCLUSION

In this paper, we proposed a cross-layer video streaming system that integrates an innovative "cognitive layer." This "cognitive layer" implements intelligent services and is capable of interfacing the underlying network technology to provide: 1) a system-level automatic audiovisual object classification model; 2) a robust and adaptive ALF protocol with fine-grained TCP-Friendly rate control and unequal error protection; and 3) an application-level video packet marking algorithm to be deployed on DiffServ-enabled networks.

This "cognitive layer" is an extension to the MPEG-4 system architecture that makes the use of a neural network classification model to dynamically and accurately group audiovisual objects of a scene with same QoS requirements to create elementary video streams that are subsequently mapped to IP DiffServ PHBs. These MPEG-4 AVOs are classified based on application-level QoS descriptors and MPEG-7 content-descriptive metadata. Thus, MPEG-4 AVOs requiring same QoS from the network are automatically classified and multiplexed within one of the IP DiffServ PHB. Object data-packets within the same class are then transmitted over the selected transport layer with the corresponding bearer capability and RPS.

The transmitted MPEG-4 streams take also benefit from the proposed "cognitive layer" by applying an unequal error protection according to the priority score of each object. The amount of recovered data is related to audiovisual objects priority score in the MPEG-4 scene. The more the object is important, the more the recovered data is valuable and better.

For fair share of bandwidth and higher user perceived quality, we have designed a content-based rate adaptation mechanism

for MPEG-4 video streams that uses a TCP-Friendly Rate Control. The novelty of our mechanism is to perform video rate adaptation by adding and dropping MPEG-4 AVOs according to their subjective relevancy to the service (i.e., through MPEG-4/7 object descriptors/metadata analysis) and instant network congestion estimations.

We have evaluated the performance of the proposed IP video streaming system by simulation using ns2. The MPEG-4 server implemented in ns2 uses the TFRC module as an equation-based congestion control mechanism. We coupled end-to-end congestion control with a DiffServ network that guarantees objects prioritization within the network. The simulation results show that subjectively important multimedia streams (audio, video in the foreground) are preserved from loss by the routers in situation of network congestion. These results show clearly the gain obtained in term of visual quality compared with a classical video streaming system without a "cognitive layer."

Combining these control mechanisms into a coherent and cognitive middleware demonstrates the usefulness and gains of interfacing content-level semantic and service-level QoS requirements to the underlying next-generation network technologies.

REFERENCES

[1] "MPEG-7 Overview," ISO/IEC JTC1/SC29/WG11, N4980, 2002.

[2] S. Iyer, R. R. Kompella, and A. Shelat, "ClassiPI: An architecture for fast and flexible packet classification," *IEEE Netw. (Special Issue on Fast IP Packet Forwarding and Classification for Next-Generation Internet Services)*, vol. 15, no. 2, pp. 33–41, Mar./Apr. 2001.

[3] P. Gupta and N. McKeown, "Algorithms for packet classification," *IEEE Netw.*, vol. 15, no. 2, pp. 24–32, Mar./Apr. 2001.

[4] D. Heckerman *et al.*, "Real-world applications of Bayesian networks," in *Commun. ACM*, vol. 38, Mar. 1995, pp. 24–26.

[5] J. Rosenberg and H. Schulzrinne, "RTP payload format for generic forward error correction," Request for Comments, IETF, RFC 2733, 1999.

[6] J.-C. Bolot and T. Turletti, "Adaptive error control for packet video in the internet," in *Proc. IEEE ICIP*, Lausanne, Sep. 1996, pp. 232–239.

[7] G. Carle and E. W. Biersack, "Survey of error recovery techniques for IP based audiovisual multicast applications," *IEEE Netw.*, vol. 11, no. 6, pp. 24–36, Jun. 1997.

[8] I. Rhee and S. Joshi, "Error recovery for interactive video transmission over the internet," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1033–49, Jun. 2000.

[9] B. Girod, K. Stuhlmuller, M. Link, and U. Horn, "Packet loss resilient internet video streaming," in *Proc. SPIE Visual Commun. Image Processing*, San Jose, CA, Jan. 1999, pp. 833–844.

[10] G. Liebl, M. Wagner, J. Pandel, and W. Weng. (2003) An RTP payload format for erasure-resilient transmission of progressive multimedia streams. [Online]. Available: draft-ietf-avt-uxp-06.txt

[11] A. H. Li *et al.*. (2003) An RTP payload format for generic FEC with uneven level protection. Internet Draft. [Online]. Available: draft-ietf-avt-ulp-07.txt

[12] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[13] A. Albanese, J. Bloemer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inf. Theory*, vol. 42, no. 11, pp. 1737–1747, Nov. 1996.

[14] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," Request for Comments, IETF, RFC3550, 2003.

[15] M. Mahdavi and S. Floyd, "TCP-Friendly unicast rate-based flow control," Tech. Note sent to the end2end-interest mailing list, 1997.

[16] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. ACM SIGCOMM*, 2000, pp. 43–56.

[17] S. Floyd and K. Fall, "Promoting the use of end-to-End congestion control in the internet," *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 458–472, Aug. 1999.

[18] 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," in *Proc. NOSSDAV*, Cambridge, U.K., Jul. 8–10, 1998, pp. 215–226.

[19] R. Rejaie, M. Handley, and D. Estrin, "RAP an end-to-end congestion control mechanism for real-time streams in the Internet," in *Proc. IEEE INFOCOM*, vol. 3, New York, Mar. 1999, pp. 1337–1345.

[20] S. jin, L. Guo, I. Matta, and A. Bestavros, "A spectrum of TCP-Friendly window-based congestion control algorithms," *IEEE/ACM Trans. Netw.*, vol. 11, no. 3, pp. 341–355, Jun. 2003.

[21] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP-Friendly rate control (TFRC): Protocol specification," Request for Comments, IETF, RFC 3448, 2003.

[22] J. Heinanen and R. Guerin, "A two rate three color marker (TRTCM)," RFC 2698, 1999.

[23] "Internet protocol communication service—IP performance and availability objectives and allocations," ITU-T, ITU-T Recommendation Y.1541, 2001.

[24] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the Internet," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 3–10.

[25] T. Ahmed, A. Mehaoua, and V. Lecuire, "Streaming MPEG-4 audiovisual objects using TCP-Friendly rate control and unequal error protection," in *Proc. IEEE Int. Conf. Multimedia Expo*, vol. 2, Baltimore, Jul. 2003, pp. 317–320.

[26] T. Ahmed, A. Mehaoua, R. Boutaba, and Y. Iraqi, "IP video streaming with fine-grained TCP-friendly rate adaptation," in *Lecture Notes in Computer Science*. New York: Springer-Verlag, 2003, vol. 2839, IFIP/IEEE Management of Multimedia Networks and Services, (MMNS'03), pp. 18–31.

[27] A. Nafaa, T. Ahmed, and A. Mehaoua, "Unequal and interleaved FEC for wireless MPEG-4 video multicast," in *Proc. IEEE Int. Conf. Commun.*, vol. 27, Paris, France, Jun. 2004, pp. 1431–1435.

[28] F. P. Kelly, P. B. Key, and S. Zachary, "Distributed admission control," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 12, pp. 2617–2628, Dec. 2000.

[29] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint admission control: Architectural issues and performance," in *Proc. SIGCOMM 2000*, Stockholm, Sweden, 2000, pp. 57–69.

[30] I. Stoica and H. Zhang, "Providing guaranteed services without per flow management," in *Proc. ACM SIGCOMM*, Cambridge, MA, Sep. 1999, pp. 81–94.

[31] R. Mortier, I. Pratt, C. Clark, and S. Crosby, "Implicit admission control," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 12, pp. 2629–2639, Dec. 2000.

[32] B. Pang, H. Shao, W. Zhu, and W. Gao, ""An admission control scheme to provide end-to-end statistical QoS provision in IP networks," in *Proc. 21st IEEE Int. Conf. Perform., Comput., Commun.*, 2002, pp. 399–403.

[33] M. Fidler and V. Sander, "A parameter based admission control for differentiated services networks," *J. Comput. Netw.*, vol. 44, no. 4, pp. 463–479, 2004.

Toufik Ahmed received the M.S. and Ph.D. degrees in computer science from the University of Versailles, Versailles, France, in 2000 and 2003, respectively.

He was a Visiting Scientist at the School of Computer Science, University of Waterloo, Waterloo, ON, Canada, in 2002. He is currently an Associate Research Fellow at PRiSM Laboratory, University of Versailles. His main research activities concern quality-of-service for multimedia wired and wireless networks.

Ahmed Mehaoua received the Ph.D. degree in computer science from the University of Versailles, Versailles, France, in 1997.

He is an Associate Professor in the Computer Science Department, University of Versailles. He is currently the Leader of the Multimedia and Mobile Networking Group, CNRS PRiSM Laboratory. Prior to joining the University of Versailles, he was with the University of Cambridge, Cambridge, U.K., as an Associate Research Fellow in the Center for Communications Systems Research and with the Hitachi Telecom R&D Laboratory. From 1995 to 1998, he was with the Computer Science Research Institute, Montreal, Canada, and was involved in various R&D projects in telecommunication system management and quality-of-service management in broadband ATM networks. His main research interests include digital video communications over wired and wireless ATM and IP networks. His research has been supported by grants from government and corporate sources (IST, CNRS, RNRT, THALES).

Raouf Boutaba (M'01) received the M.S. and Ph.D. degrees in computer science from the University Pierre Marie Curie, Paris, France, in 1990 and 1994, respectively.

He is currently an Associate Professor in the School of Computer Science, University of Waterloo, Waterloo, ON, Canada. He conducts research in the areas of network and distributed systems management and resource management in multimedia wired and wireless networks. In these areas, he has published more than 120 papers in refereed journals and conference proceedings.

Dr. Boutaba is the recipient of the Premier's Research Excellence Award, a fellow of the faculty of mathematics of the University of Waterloo, and a distinguished lecturer of the IEEE Computer Society. He is the Chairman of the IFIP Working Group on Networks and Distributed Systems, the Vice Chair of the IEEE Communications Society Technical Committee on Information Infrastructure, and the Chair of the IEEE Communications Society Committee on Standards. He is on the Advisory Editorial Board of the *Journal of Network and Systems Management*, the Editorial Board of the *KIKS/IEEE Journal of Communications and Networks*, and the *Journal of Computer Networks*.

Youssef Iraqi received the B.Sc. degree in computer engineering, with high honors, from Mohamed V University, Morocco, in 1995, and the M.S. and Ph.D. degrees in computer science from the University of Montreal, Montreal, Canada, in 2000 and 2003, respectively.

He is currently a Research Assistant Professor at the School of Computer Science, University of Waterloo, Waterloo, ON, Canada. From 1996 to 1998, he was a Research Assistant at the Computer Science Research Institute, Montreal, Canada. His research interests include network and distributed systems management, resource management in multimedia wired and wireless networks, and wireless ad hoc networking.