Delivering of MPEG-4 Multimedia Content over Next Generation Internet

Toufik Ahmed, Guillaume Buridant, and Ahmed Mehaoua

University of Versailles - CNRS-PRiSM Laboratory
45, av. des Etats Unis 78035 – Versailles - France
Tel: +33 1 39 25 40 56
Fax: +33 1 39 25 40 57
{tad, bug, mea}@prism.uvsq.fr

Abstract. IP and MPEG-4 digital video are two key technologies that will be increasingly combined in the very near future for the deployment of next generation Internet multimedia services. This article discusses technical issues related to the transport and the QoS control of MPEG-4 multimedia applications, such as interactive video-on-demand, over Internet. In particular, it describes the design, implementation and performance evaluation of an experimental Java-based MPEG-4 interactive VOD system over IP DiffServ networks. We demonstrate that Assured Forwarding Per Hop Behavior is a performing candidate for conveying real-time video communications in conjunction with video-aware packet marking and scheduling mechanisms and advanced MPEG-4 DMIF signaling protocol. In this issue, we also propose a Video packet Marking Algorithm for IP Diffserv routers, called DVMA that reduces video packet loss probability and end-to-end transfer delay during network congestion.

1   Introduction

The rollout of Internet opens up a new frontier for the digital video broadcasting industry. Two worlds that were barely connected are on the verge of being merged: namely, real-time video and Internet. Thanks to the transition of the video medium from analogue to digital, and the rise of powerful video compression standards, such as MPEG-2 and MPEG-4, it is now possible to combine video, audio, and data within the same signal and transmit it over an integrated packet switching network infrastructure, such as Internet.

This combination leads to powerful new multimedia applications, with limitless possibilities of great commercial potential. For example, computers can be turned into traditional TV receivers and the digital set-top boxes can host applications such as interactive TV, e-commerce, and customized programming.

This article discusses technical issues related to the delivery of MPEG-4 multimedia content, such as interactive video-on-demand, over next generation Internet. The article is organized as follows. Section 2 is devoted to the description of

IP QoS networking and MPEG-4 video technologies that contribute to this issue. Section 3 discusses the design and the implementation of the proposed interactive video distribution service. Interaction between MPEG-4 framework and IP Diffserv architecture is also investigated. In section 4, a QoS control mechanism named DVMA, for *IP DiffServ video packet Marking Algorithm*, is presented and integrated in our architecture. Performance evaluation and analysis is carried out in section 5 and 6 respectively. Finally, we conclude and discuss future work in Section 7.

2   Internet and Digital Video Convergence

2.1   Internet Multimedia Protocols: RTP, RTCP, and RTSP

Unlike ordinary data services, multimedia services are sensitive to delay, jitter, and loss, which in turn are affected by the volume and pattern of traffic load in the network. A Transport Control Protocol (TCP)-based stream will respond to losses or excessive delays by triggering a retransmission and exerting window flow control, both of which will have negative effects on the quality of the multimedia service.

The IETF *Audio/Video Transport* (AVT) working group has partially addressed this issue by proposing the *Real Time Transport Protocol* (RTP), the associated *Real Time Transport Control Protocol* (RTCP) and the *Real Time Streaming Protocol*. RTP is a lightweight transport protocol based on the concept of application-level framing [1]. Since the reliable transmission mechanism offered by TCP incurs considerable overhead by retransmission, RTP does not rely on TCP. Instead, applications are put on top of UDP. How to cope with the lost packets is up to the applications. Following the application-level framing principle, RTP functionality is usually integrated into the application rather than being implemented as a separate protocol entity. RTP provides basic packet format definitions to support real-time communication but does not define control mechanisms or algorithms. The packet formats provide information required for audio and video data transfer, such as the incoming video data packet sequence. Continuous media such as non-compressed PCM audio can be synchronized by the use of sequence numbers. Non-continuous data such as MPEG can be resynchronised by using the time stamp fields [17].

RTCP is the control protocol for RTP, and provides mechanisms for data distribution monitoring, cross media synchronization, and sender identification [1]. The sender transmits a multimedia data stream by using RTP packets. The receiver periodically sends RTCP packets that contain information about the received RTP packets. The information includes statistics such as the highest sequence number received, inter-arrival jitter, or packet loss rate.

RTSP [28] is used for initiating and controlling delivery of stored and live multimedia content to both unicast and multicast destinations. RTSP borrows time concept from MPEG-2 DSM-CC, but unlike DSM-CC, it does not depend on an entire set of supporting protocols [29]. RTSP is transport-independent, and can use either TCP or UDP as the transport protocol. The state machine makes RTSP suitable for

remote digital editing and retrieval. RTSP is also text-based protocol, and therefore easy to parse. RTSP reuses the HTTP concept, but unlike HTTP, RTSP is a stateful protocol.

2.2   Internet QoS Architectures: INTSERV and DIFFSERV

The provision of quality of service within the Internet has been the subject of significant research and deployment efforts recently. Two approaches have drained the attention of the IETF: namely the *IntServ* and the *DiffServ* architectures.

The *Integrated Service* (IntServ) model is motivated by the ability for applications to choose among multiple, controlled levels of delivery service for their data packet [5]. In the integrated service framework, many functions are used to provide QoS: the first function is control services such as Controlled-Load [6] and Guaranteed [7]. The second function may be provided in a number of ways, but is frequently implemented by a resource reservation setup protocol such as RSVP [8].

The *Differentiated Services* (DiffServ) model [9] uses a small, well-defined set of building blocks from which a variety of aggregate router behaviours may be designed to provide quality of service [10]. IP packets are tagged with distinct labels before entering an IP Diffserv domain and will receive particular forwarding treatment at each network node along the path. This set of routing function is called a *Per-Hop Behavior* (PHB). The PHB is characterized and invoked according to the Differentiated Service Code point (DSCP) value located in the packet's header. A small number of PHBs has been currently standardized by the IETF Diffserv working group. The most well known are Expedited Forwarding (EF) [11] and Assured Forwarding (AF) [12].

The key difference between Intserv and Diffserv is that while Intserv provides end-to-end QoS service on a per-flow basis, Diffserv is intended to provide better scalability through flow aggregation and class differentiation over a long timescale.

Therefore, we believe that IP Diffserv is the most suitable model for delivering interactive video content over Internet and we present, in the following lines, some related works in this field.

In [2], The authors present a content-based packet video forwarding mechanism where the QoS interaction between video applications and Diffserv network is taken into account. The interaction is performed through a dynamic mapping between the RPS (relative priority score) of each video packet and the differentiated forwarding mechanism.

In [3], the authors introduce a QoS management system for multimedia servers that benefits from the scaling properties of layered media streams. The presented system enable to map application QoS demands to available streams according to inter-stream QoS dependencies.

Authors in [4] present a bit-stream classification, prioritization, and transmission scheme designed for MPEG-4 video. Different type of data are re-assembled and assigned to different priority using IP Diffserv model.

2.3   MPEG-4 Multimedia Framework: BIFS and DMIF

MPEG-4 is an emerging digital multimedia standard with associated protocols for representing, manipulating and transporting natural and synthetic multimedia content (i.e. audio, video, data) over a broad range of communication infrastructures. The original characteristic of MPEG-4 is to provide an integrated object-oriented representation of multimedia content for the support of new ways of communication, access, and interaction with digital audiovisual data, and offering a common technical solution to various telecommunications, broadcast, and interactive services. MPEG-4 addressed a broad range of existing and emerging multimedia applications such as video on the Internet, multimedia broadcasting, content-based audiovisual database access, games, audiovisual home editing, advanced audiovisual communications and video over mobile networks.

This is achieved by a set of tools defined in several recommendations of the standard. First, the media compression schemes are defined in the Visual [14] and Audio [15] parts of the MPEG-4 framework. Every multimedia element to be compressed are presented as individual audiovisual objects. The combination of these elements is assured by the scene description language called *Binary Format for Scenes* (BIFS) defined in MPEG-4 Systems document [13]. The delivery of the media is defined in the *Delivery Multimedia Integrated Framework* (DMIF) [16], which is the part 6 of MPEG-4 specification. DMIF is actually the control plane of MPEG-4 *Delivery layer*, which allows applications to transparently access, retrieve and view multimedia streams whether the source of the stream is located on a remote or local end-system.

MPEG-4 terminal architecture is composed of three layers (see Figure 1): the *Compression Layer*, the *Sync Layer* and the *Delivery Layer* [13].
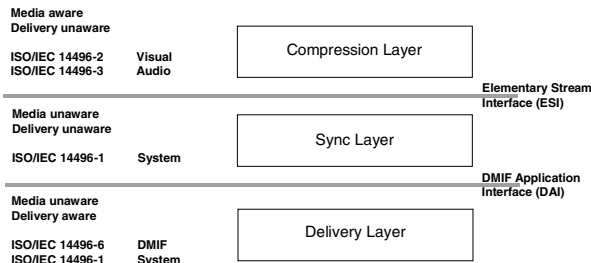

Fig. 1. MPEG-4 Framework

Compression layer generates representation of content data called *Elementary Streams* (ES), the hierarchical relations; locations and properties of ESs in a representation are described by dynamic set of *Object Descriptors* (OD). ESs are the basic abstraction for any data source. ODs are themselves conveyed through one or more ESs. MPEG-4 scene is described by *Scene Description* (SD) information that addresses the organization of audiovisual objects within a scene, in terms of both spatial and

temporal location. SD is performed using BIFS, which is a VRML-based language. ESs are partitioned into *Access Units* (AU) which are defined by MPEG-4 general framework for the framing of semantic entities in ES. For examples, a valid MPEG-4 ES could be an MPEG-1 video, labeled with MPEG-4 system information in its headers. An AU would then be one video frame I, B or P. Those AUs will be labeled with priority information, timing information, and others. An AU is the smallest data entity to which timing information can be assigned.

On the *Sync layer*, AUs are conveyed into a sequence of packets called SL-PDUs. A SL-PDU consists of SL packet header and SL packet payload. The header provides means of continuity checking in case of data loss, carries the time stamps, and associated control information.

The SL-PDUs are subsequently transmitted to the *Delivery Layer* for multiplexing and generating a FlexMux stream. The FlexMux tool is one of the components of the MPEG-4 DMIF. FlexMux is a flexible and simple data multiplexer that accommodates interleaving of SL-PDUs. Two different modes of operation for FlexMux are defined. The first mode is called "Simple Mode" in which one SL-PDU is encapsulated into one FlexMux packet, and the second mode is called "MuxCode Mode" in which one or more SL-PDU are encapsulated into a single "FlexMux" packet. The interface between the *Sync layer* and the *Delivery layer* is referred to DAI (*DMIF Application Interface*) [16]. It offers content location independent procedures for establishing MPEG-4 sessions and access to transport channels such as RTP/UDP/IP. DMIF is primarily an integration framework. It provides default DMIF signalling protocol which corresponding to DNI (*DMIF Network Interface*).

3   An Experimental Real-Time and Interactive Video on Demand Service over IP

3.1   Proposal of a Video Packet Marking Algorithm for IP DiffServ Routers

MPEG-4 uses hierarchical coding for resolving scalability and heterogeneity issues. Different coding modes exist: *Spacial scalability* allows the decoder to treat a subset of streams produced by the coder to rebuild and display textures, images and objects video with a reduced space resolution. *Temporal scalability* allows the decoder to treat a subset of streams produced by the coder to rebuild and display a video with reduced temporal resolution. With *SNR Scalability* (Signal to Noise Ratio), the coder transmits the difference between the original image and the preceding one. This allows improving subjective quality of the final image to get maximum quality [21].

We use the IP packet marker Algorithm described in [21] to tag the different MPEG-4 ESs (see Figure 2). The MPEG-4 video stream is composed of tree hierarchical layer: *Base layer stream* that offers a **minQoS** (Minimum QoS), *enhanced layer 1 stream* is used to improve minQoS and to offer a **MedQoS** (Medium QoS) and *enhanced layer 2 stream* is used to improve medQoS and to offer a **MaxQoS** (Maximum QoS).

In the Best Effort service, when network congestion occurs, packets are discarded with no distinction. Therefore, essential audiovisual data (i.e. control and decoding information, audio, …) can encounter important drop probability while less important information . The result of this situation is that the decoder cannot reproduce properly the video sequences. The solution of this problem is to distinguish between important data and less relevant ones. This can be achieved by taking into account MPEG-4 video coding properties and marking the outgoing video packet at the video server side. In this situation, IP video packets are dropped according to their semantic

```
if stream is "audio stream" then (application of property 2)
        if coder rate is "low rate"
                then DSCP=AF Low Drop Prec
                //example AF11
        if coder rate is "medium rate"
                then DSCP=AF Medium Drop Prec
                //example AF12
        if coder rate is "high rate"
                then DSCP=AF High Drop Prec
                //example AF13

if stream is "video stream" (application of property 3)
        if "base layer video stream"              (level 1 = mimimum QoS)
                then DSCP = AF low Drop Prec
                //example AF21
        if " enhanced layer video stream 1"       (level 2 = medium QoS)
                then DSCP = AF Medium Drop Prec
                //example AF22
        if "enhanced layer video stream 2"        (level 3 = maximum QoS)
                then DSCP = AF Medium Drop Prec
                //example AF23

if stream is "objects descriptor, scene descriptor" (application of property 4 )
                then DSCP = EF
                //descriptions streams are significant, no loss
                //it's necessary that these streams will be available
                //as soon as possible in MPEG-4 player to interpret
                //correctly the scene
```

Fig. 2. DVMA – An IP DiffServ packet Video Marker Algorithm

relevance. If the packet is marked as low priority, then it will be dropped first in situation of congestion. In addition, when the video sequence is structured as a multi-layered video stream, multiple heterogeneous clients can receive simultaneously the video sequence by selecting the number of layer that they can handle.

3.2   Design and Implementation of the IP Diffserv Experimental Network

The latest Linux kernel (2.4) offers a wide variety of network traffic control features [18], [17], such as IP Diffserv support. Unfortunately, basic classification and IP packet DS-field manipulation required by the Diffserv model are not correctly addressed by Linux.

When a Linux-based IP gateway receives IP packets from it input device that must be forward to the output device, the traffic control determines if packets have to be queued or dropped. For example, if the queue length reaches a threshold or if the traffic exceeds some rate limit, the scheduling module decides which packet should be served first according to given priority. Finally, it can delay the sending of packets, for example to limit the rate of outbound traffic. Linux Box can operate as Edge or Core router. The Edge router can classify, police, mark aggregates and reshape traffic. The Core routers provide transit packet forwarding service between other core and edge routers. It manages traffic to avoid and cope with congestion.

The traffic control code in the Linux kernel consists of the following major conceptual components: (1) queuing disciplines, (2) classes (within a queuing discipline), (3) filters and (4) traffic policing. Figure 3. depicts the Linux traffic control components.

The queue determines the order in which packet will be served and send to the outgoing ports. Traffic can be divided into different classes according to specific rules. Each class maintains a queuing discipline to serve the packets, and it is associated with a priority. Actually there are 11 queuing disciplines implements in the Linux Kernel [18].
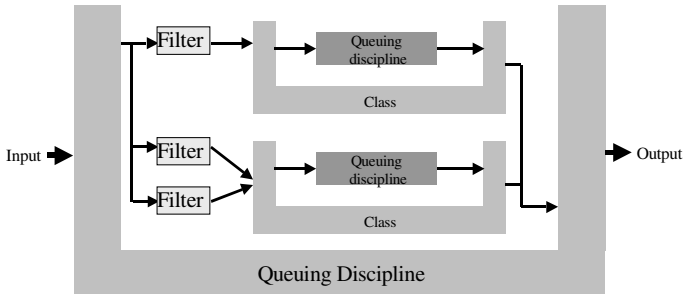
Fig. 3. Linux Traffic Control Components

Filters are used to discriminate the packet and to associate them with the appropriate class of traffic. The classification is based on certain properties of the packet. In the differentiated service the classification is based on the IP DSCP (DiffServ Code Point) field.

Traffic policing is used to control the amount of entering traffic in each class. It controls that a specific traffic does not exceed a predefined bound.

3.3   Design and Implementation of the Proposed MPEG-4 Interactive VOD System

We developed an MPEG-4 interactive video on demand system consisting of a video server and a multimedia client communicating over the previously described IP Diffserv network testbed as illustrated in Figure 4. The client and server code is based on *Java Media Framework* developed by Sun Microsystems [19]. We extended the client and the server functionalities by integrating DMIF signalling. Thus, the client can establishes a session with the server using DMIF primitives to select, negotiate and retrieve MPEG-4 video sequences. The DMIF implementation is out of the scope of this paper but additional information can be found in [20].

The MPEG-4 Server consists of MPEG-4 pump, DMIF Instance for IP network, and tools for RTP/UDP/IP stack.

The server delivers ES packetized Stream to FluxMux tools witch encapsulates ES packet in RTP packets. For our testbed we have used a video only presentation with three hierarchically stream, which are carried in three separated RTP session. Document [21] explains in more details the video encapsulation protocol used.

The MPEG-4 pump talks to a clients via DMIF and delivers RTP stream during the time of the session. The DMIF Instance is responsible for session setup request and release, it also provides QoS requirement for the application. This option is not yet supported in our implementation. The signaling channel for DMIF uses TCP protocol for reliability.

When a client requests a presentation from the server, he queries its local DMIF daemon to initiate a session with the remote video server. The local DMIF contact the remote DMIF to establish a signalling channel. The server answers client using the same channel with response code. When the session is setup, the client requests one or several stream channels for presentation, afterwards, the server pushes the audiovisual data in the RTP channel and marks the stream with the appropriate PHB using DVMA.
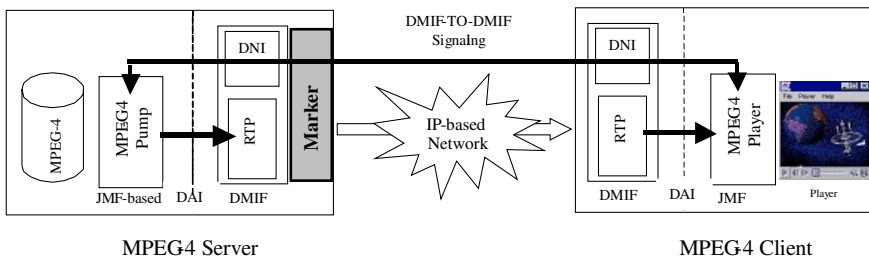
Fig. 4. A JAVA-based MPEG-4 VOD system with DMIF Session Signaling Implementation

The example below resumes the steps for activating a VoD session by the client:
1. *The client application initiates the services by calling in it local DMIF a DA_ServicesAttach primitive.*
2. *The DAI determines weather a new Network Session is needed. If it is, it calls a DN_SessionSetup.*
3. *The server DMIF replays the client DMIF by attaching creating a new session.*
4. *The DMIF server activates the session in the video server.*
5. *Upon these steps, media channels are opened and media flows can be sent..*

4 Performance Evaluation

4.1 Network and Traffic Models

Let is consider the IP network testbed depicted in Figure 5., where a server delivers MPEG-4 video content to several heterogeneous receivers. The receiver can be a simple wired PC or any wireless terminals such as a mobile GSM/GPRS/UMTS phone capable of rendering MPEG-4 video sequences. The network nodes are IP Diffserv compliant routers. We use Linux-based IP routers with Diffserv implementation.
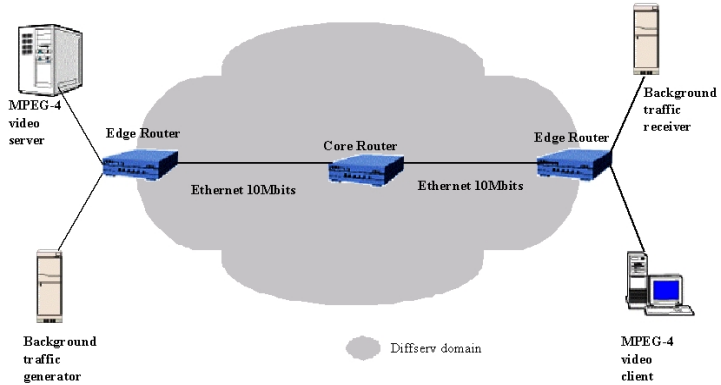
Fig. 5. IP DiffServ Network testbed

Our IP Diffserv network testbed is composed of a video server application that plays MPEG-4 video stream for many heterogeneous clients. The server and clients are connected through TCP/IP network. We exploit some test-scenario on the transmission process. We quantify QoS measurement and we compare between transmission process in Best Effort service and in Differentiated Service.

We've implemented an MPEG-4 video streaming system to test and evaluate the performance of the proposed marker algorithm. The testbed is illustrated in Figure 5. The testbed is composed of tow edge routers and a core router running Linux with IP Diffserv components. All Ethernet links are 10 Mb/s.

4.2   Edge Router Configuration

Edge routers accept traffic into the network. They can characterize, police, and/or mark customer traffic between other edge or core routers.

Within the Diffserv domain, service association is performed according to the DSCP value in each packet's IP header. Therefore, the video application must mark the packets correctly to obtain a particular level of service within the Diffserv region.

It is not desirable to let the network (edge router) marking the incoming traffic as several algorithms do. The marking algorithms such as *Time Sliding Window Three Colour Marker* (TSWTCM) [19] and a *Two Rate Three Color Marker* (TRTCM) [23] cannot be used in the case of multimedia traffic. These algorithms make no distinction between essential audiovisual data and less important ones and thus cannot mark the packet correctly for future control within the network.

The configuration of the edge router is simple in our testbed. Our edge router limits the amount of EF traffic to 15% of the bandwidth capacity rate i.e. 1.5Mbit. We used a policing mechanism to limit the EF traffic, because EF is more require in term of latency time and losses. Furthermore, the router must make sure that the departure rate configured is greater than the arrival rate and the queuing delay is minimized. This is extensively sufficient since we use EF only for sensitive information such as OD and

BIFS signalling, that should be transported to the destination as early as possible, with no loss and with a minimum jitter. The EF flow is bounded and isolated.

For the Edge Router we used a simple *Class-Based Queuing* (CBQ) discipline to classify the incoming traffic.

4.3 Core Router Configuration

Core routers are configured to perform (1) packet classification based on DSCP, (2) packet scheduling, (3) queue management, (4) policing and (5) packet dropping.

We used CBQ as the packet scheduler, which is a classical assumption as proposed in [24]. For CBQ a single set of mechanisms is proposed to implement link-sharing and real-time services. In our implementation, CBQ is used to classify EF, AF, and BE traffic classes so each connection can get appropriate resources based on packet marking.

Our CBQ mechanisms include:

* a *classifier* to classify arriving packets to the appropriate class. This classification is based on DSCP field in the IP header,
* a *scheduler* to determine the order in which packets from the various classes will be sent. Linux Kernel implements several queuing disciplines (e.g. **RED** "Random Early Detection" or **GRED** "generalized RED"). The GRED queuing discipline is used to support multiple drop priorities as required by AF PHB. One physical GRED queue is composed of multiple **VQ** (Virtual Queue). GRED can operate in **RIO** *(*RED with In/Out bit) *mode* [25], with coupled average queue estimates from the virtual queues, or in *standard mode* where each virtual queue has its own independent average queue estimate as required by RED [26]. In our testbed, we used GRED as queuing discipline for the AF classes, since our marker algorithm takes into account these properties to give different level of QoS: *minQoS*, *MedQoS* and *MaxQoS*.

For the AF classes we allocated 1.5Mbit/s for each AF sub-classes namely AF1, AF2, AF3 and AF4, all of which are bounded. For the best effort traffic, we allocated a minimum of 3.5Mbit but this traffic is allowed to borrow any available bandwidth.

5 Performance Analysis

Figures 6 and 7 give statistical properties of the MPEG-4 video traffic generated by the video server to the client. The network is loaded by TCP and UDP background traffic. In our testbed, the background traffic is marked as best effort traffic.

The first set of performance measurements are on packet loss probability for each video elementary streams, in both IP best effort and Diffserv models. The second set of measures concern the end-to-end one-way delay encountered by video packet
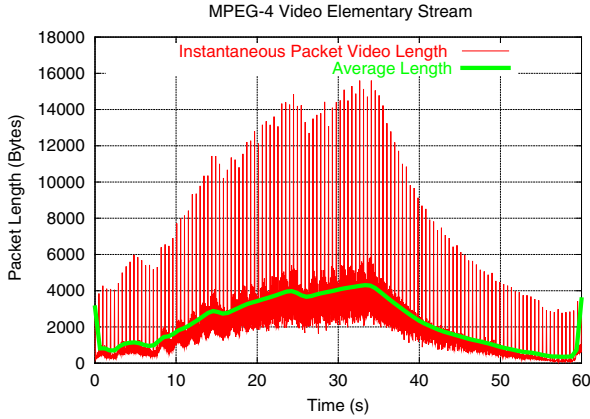
MPEG-4 Video Elementary Stream

Fig. 6. Instantaneous packet length in MPEG-4 Video Elementary Stream

between the server and the destination. Two different network loads have been tested, i.e. 80% and 95% of the available bandwidth.
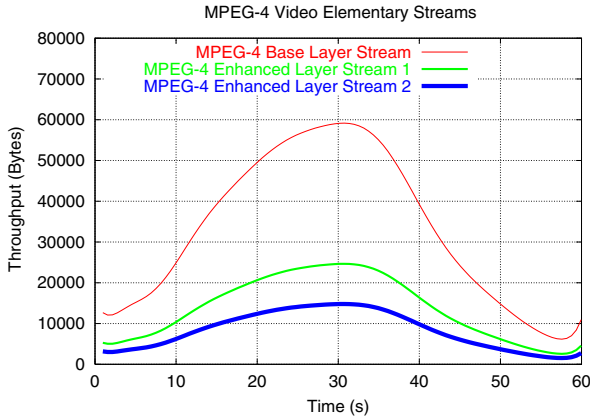
MPEG-4 Video Elementary Streams

Fig. 7. MPEG-4 Video Layered Streams

5.1   IP Packet Losses

Figures 8 and 9 depict the variation of the video packet loss versus network load for IP Best Effort and IP Diffserv respectively. Individual MPEG-4 video layers encounter different packet loss probability. With IP Best Effort, the most essential video layer (i.e. base layer) obtains the highest loss with 60 % for a network load of about 80%. Using IP Diffserv and DVMA, the base layer faces the lowest packet drop probability with a maximum of about 0.2 %.
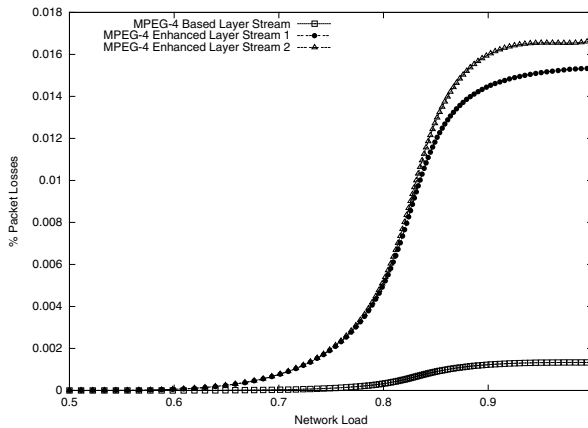
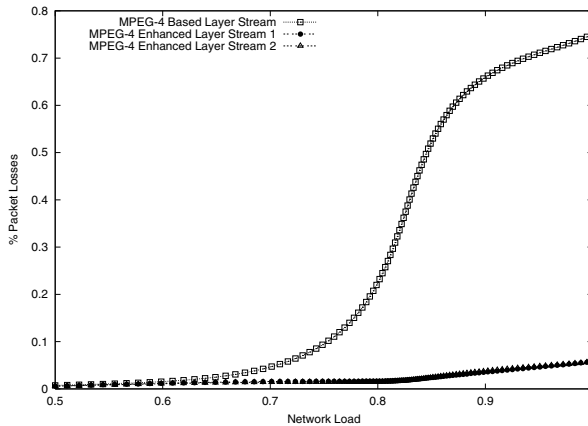**Fig. 8.** MPEG-4 video packet Loss ratio vs Network load with IP Diffserv

Fig. 9. MPEG-4 video packet Loss ratio vs Network load with IP Best Effort

Figure 10 shows the percentage of packets losses when the amount of background traffic is about 80% (6.5 Mbit/s) of the bandwidth. This leads to some losses essentially at time t=30s i.e. when the video server sends at its peak rate (figure 9). The majority of the losses are concentrated within the base layer stream. This provides a degradation of the video quality at the client. Moreover, the receiver can't decode properly the other elementary video streams without the good reception of the base layer. Loss increases dramatically when the network load increases (Fig. 11).

The high losses of the base layer are due to his highest requirement of bandwidth. We can compare it with a MPEG-2 video stream where I pictures are bigger than P and B pictures. However, they are much more important. In this case, I packet's losses must be lower than the other packets types losses. When talking about MPEG-4, the

base layer stream must have a low packet loss when the enhanced layers streams 1 and 2 must have a respectively increasing drop probability.
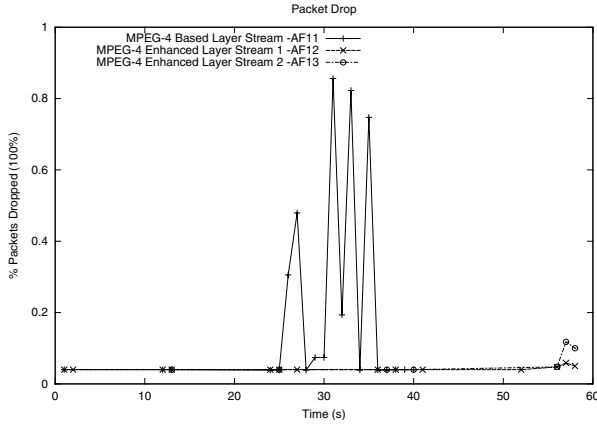

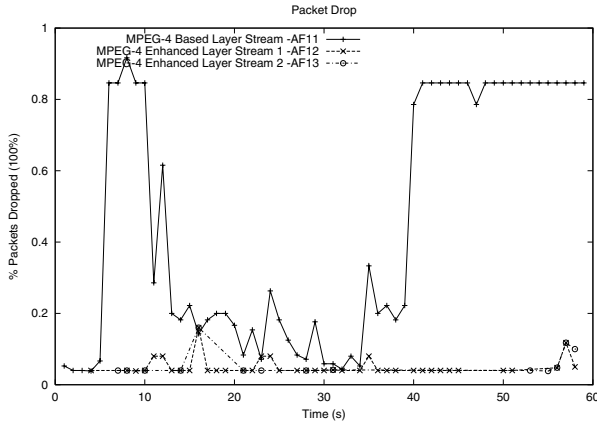**Fig. 10.** Packet drops in Best Effort scenario. - Network Load 80% -

Fig. 11. Packet drops in Best Effort scenario. - Network Load > 95%

With IP Diffserv, we can see that the video packet losses are almost equals to 0, and we have ensured that the base layer has no losses. Figures 12 and 13 illustrate this fact.

5.2 End-to-End Transmission Delay

Figures 14 and 15 illustrate the end-to-end delay, which is an important component of the user-perceived QoS. Since, we are dealing with real time information, two much delayed audiovisual IP packets are simply discarded by the destination.
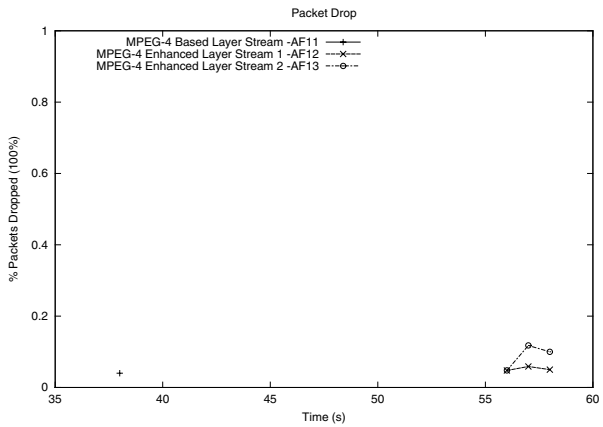
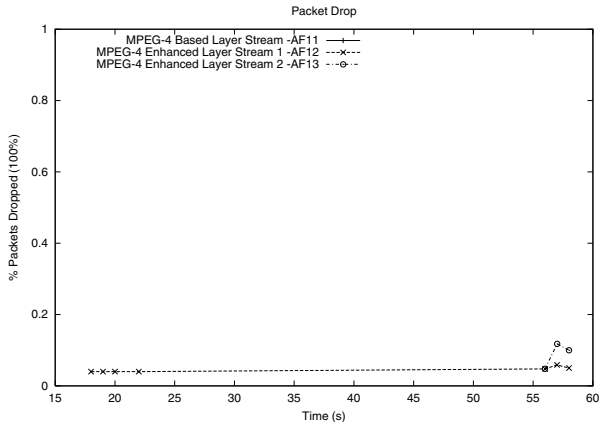**Fig. 12.** Packet drops in IP Diffserv.- Network Load 80% -

Fig. 13. Packet drops with IP Diffserv- Network Load >95% -

We can notice that the video packet's losses are the similar regardless the network load, i.e. 80% or 95% of the bandwidth. It indicated that traffic load variation have no deep effect upon the video traffic. This is simply due to the static priority-based packet scheduling mechanism performed by the gateways. We also note that best effort traffic class (background traffic) can dynamically use any available bandwidth.

Figure 14 shows that during the peak rate, the delay dramatically increases. When the network load is about 95%, the one-way delay measurement is the highest, about 150 ms (Figure 15).

When using the Diffserv scenario, the packet delay is decreasing and doesn't really increase when the network load reaches 95%. In both Figures 16 and 17 we can also see that the highest delay is during the peak rate at the middle of the transmission process; i.e. 116 ms.
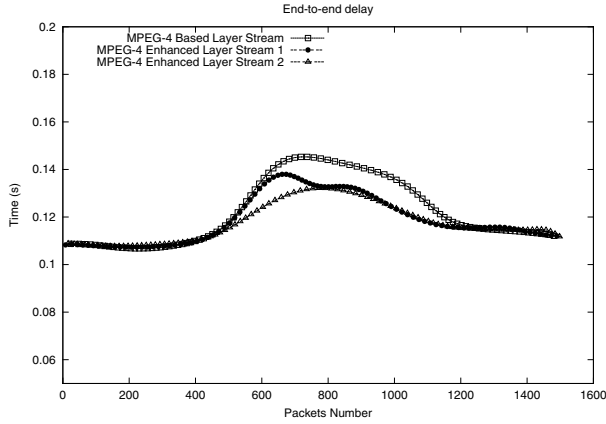
Fig. 14. End-to-end transfer delay with IP Best Effort- Network Load of 80% -
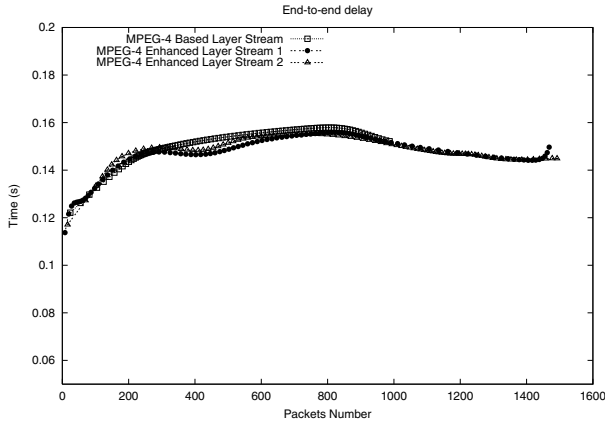

Fig. 15. End-to-end Transfer Delay with IP Best Effort - Network Load > 95% -

6   Conclusion and Future Work

In this article, a hierarchically encoded MPEG-4 interactive video on demand (VOD) service over IP Diffserv networks is proposed and evaluated using an experimental testbed. An IP packet marker, called DVMA, has been also designed and integrated in this QoS effective IP video delivery framework. Performance results have shown that two sensitive QoS parameters have been sensibly improved during network overload: video packet loss and end-to-end video packet transmission delay.
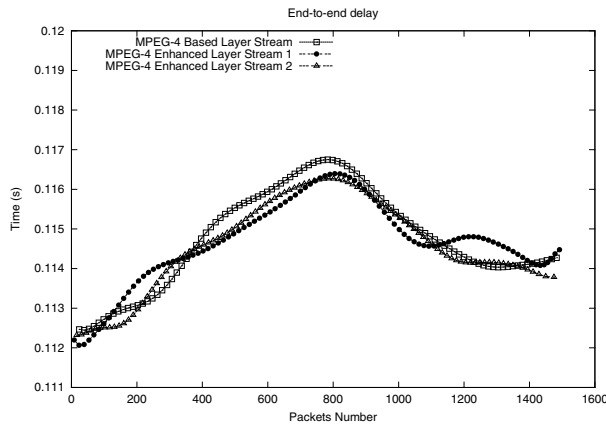
Fig. 16. End-to-end Transfer Delay with IP Diffserv - Network Load of 80% -
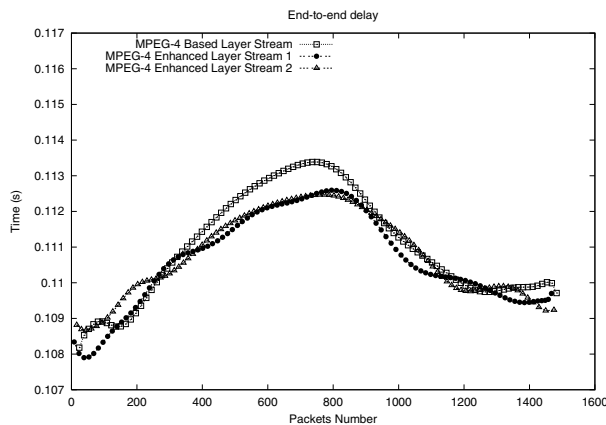

Fig. 17. End-to-end Transfer Delay with IP Diffserv - Network Load > 95% -

The proposed marking algorithm better takes into account the characteristics and relevance of MPEG-4 sub streams (audio, video, BIFS, OD signalling…) and performs well with Assured Forwarding IP Diffserv PHB. Consequently, sensitive video streams will undergo a privileged processing by the routers using our proposed transport delivery service.

Further work will examine the interaction between video stream and background traffic. As mentioned, the background traffic in our testbed is marked as best effort traffic. This is not realistic in next generation Internet services. Additionally, domain's administrators should configure edges and cores routers according to predefined high-level resource management policies and Service Level Agreements (SLA). It is

commonly accepted that cooperative Bandwidth Brokers communicating with the Common Open Policy Service Protocol (COPS) will probably assist them [27]. In this perspective, we will investigate the performance of such architecture and will propose solutions for dynamic configurations of multimedia gateways.

References

1. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson: RFC1889 RTP - A Transport Protocol for Real-Time Applications. (January 1996)
2. Jitae Shin, JongWon Kim and C.-C. Jay Kuo: Content-Based Packet Video Forwarding Mechanism in Differentiated Service Networks. IEEE Packet Video Workshop 00 (May 2000)
3. M. Albrecht, M. Köster, P. Martini, M. Frank: End-to-end QoS Management for Delay-sensitive Scalable Multimedia Streams over DiffServ. Proc. of the 25th Annual Conference on Local Computer Networks, LCN'00, Tampa-FL. (November 2000) 314-323
4. Huai-Rong, Wenwu, and Ya-Qin Zhang: Scalable Object-Based Video Multicasting Over The Internet. International Conference on Image Processing, Vancouver, Canada. (September 2000)
5. J. Wroclawski: RFC2210 – The Use of RSVP with IETF Integrated Services. (September 1997)
6. J. Wroclawski MIT: RFC2211 – Specification of the Controlled-Load Network Element Service. (September 1997)
7. S. Shenker, C. Partridge, R. Guerin: RFC 2212- Specification of Guaranteed Quality of Service. (September 1997)
8. R. Braden, L. Zhang, Berson, S. Herzog,S. Jamin: RFC 2205- Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification. (September 1997)
9. S. Blake, D. Black M. Carlson,E. Davies, Z. Wang, W. Weiss: RFC 2475- An Architecture for Differentiated Services (December 1998)
10. K. Nichols, S. Blake, F. Baker, D. Black: RFC 2474- Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (December 1998)
11. V. Jacobson, K. Nichols, K.Poduri: RFC-2598 An Expedited Forwarding PHB. (June 1999)
12. J.Heinanen, , F.Baker , W. Weiss, J. Wroclawski : RFC 2597- Assured Forwarding PHB Group. (June 1999)
13. ISO/IEC 14496-1: Coding of audio-visual objects – Part 1: Systems -final committee draft. (May 1998)
14. ISO/IEC 14496-2: Coding of audio-visual objects – Part 2: Visual -final committee draft. (May 1998)
15. ISO/IEC 14496-3: Coding of audio-visual objects – Part 3: Audio -final committee draft. (May 1998)
16. ISO/IEC 14496-3: Coding of audio-visual objects – Part 6: Delivery Multimedia Integration Framework (DMIF) final committee draft (May 1998)
17. Werner Almesberger: Differentiated Services on Linux Home Page http://icawww1.epfl.ch/linux-diffserv/
18. Werner Almesberger, Jamal Hadi Salim, Alexey Kuznetsov: Differentiated Services on Linux. Work in progress. (June 1999)
19. Sun Microsystems: Java Media Framework API http://java.sun.com/products/java-media/jmf/index.html (2000)

20. T. Ahmed. MPEG-4 DMIF Implementation in Java Technical report, Master of CS, University of Versailles, France. Available at : http://www.prism.uvsq.fr/~tad/publication (2000)

21. T. Ahmed, G. Buridant, A. Mehaoua: Encapsulation and Marking of MPEG-4 Video over IP Differentiated Services. In proceeding of Sixth IEEE ISCC 01. Hammamet Tunisia (July 2001) 346-352

22. W. Fang, Seddigh, B. Nandy: RFC2859 - A Time Sliding Window Three Colour Marker (TSWTCM). (June 2000)

23. J. Heinanen, R. Guerin: RFC2698 – A Two Rate Three Color Marker (TRTCM). (September 1999)

24. Floyd, S., Jacobson, V.: Link-sharing and Resource Management Models for Packet Networks. IEEE/ACM Transactions on Networking, Vol. 3, No. 4, pp., (August 1995) 365-386

25. David D. Clark and Wenjia Fang: Explicit Allocation of Best Effort Packet Delivery Service. ACM Transactions on Networking (August 1998) 362-373

26. Sally Floyd and Van Jacobson: Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking. Vol. 1, no. 4, (August 1993) 397-413

27. D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry: RFC 2748 - The COPS (Common Open Policy Service) Protocol. (January 2000)

28. H. Schulzrinne, A. Rao, and R. Lanphier:): IETF RFC 2326 RTSP - Real-Time Streaming Protocol (April 1998)

29. H. Schulzrinne: A Comprehensive Multimedia Control Architecture for the Internet. In Proc. IEEE 7th Workshop on Network and Operating System Support for Digital Audio and Video (May 1997) 65–76.