Applying Quality of Service Extensions to an Open Source Media Streaming Platform

S. ZOI, P. STATHOPOULOS, P. PAPAGEORGIOU, D. KOUIS, N. MITROU
Electrical & Computer Engineering Department, Telecommunications Lab
National Technical University of Athens, Polytechniopolis Zographou
GR15753, Athens, Greece,

Abstract: - In this paper, application extensions for enhancing the MPEG4IP media-streaming platform, with Quality of Service (QoS) mechanisms are presented. These mechanisms concern both best effort and differentiated services networks. In the first case, rate adaptation schemes are implemented, together with error resilience mechanisms. In the second case, an appropriate layering is adopted in order to enable the definition of QoS policies based on stream prioritization and content aware packet marking.

Keywords: - MPEG-4, rate adaptation, error resilience, packet marking

1. Introduction

State-of-the-art multimedia technology gives the potential to author complex networked multimedia applications, composed of multiple media streams (e.g. audio, video, virtual reality, images) [1] [2]. Although several application level mechanisms that enhance perceived quality have already been specified and experimentally evaluated, they are not yet widely deployed in real application scenarios.

On the other hand, network level (*Quality of Service*) QoS frameworks do exist, such as the *Differentiated Services* (DiffServ) [3] and the *Integrated Services* frameworks (IntServ) [4]. However, the deployment of QoS mechanisms to multimedia applications is still very limited, and relevant studies are usually restricted to stream traces and simulated networks [5] [6]. This is among others, because most existing multimedia applications development platforms or Application Programming Interfaces (APIs) don't incorporate QoS extensions.

In this paper, the experience from implementing application level extensions in order to enhance an open media-streaming platform, namely the MPEG4IP platform, with Quality of Service (QoS) mechanisms is presented. These extensions include rate adaptation schemes, error resilience mechanisms, as well as content aware packet marking capabilities targeting at DiffServ networks. Towards evaluating perceived quality, QoS violations are captured both at the sender and at the receiver side by a quality monitoring module and suitable metrics are extracted.

The rest of the paper is organized as follows. Section 2 provides related work concerning QoS mechanisms suited for multimedia streams, as well as existing QoS enabled platforms. In sections 3 and 4, existing features, as well as implemented extensions on the MPEG4IP platform are presented, respectively. Finally, section 5 gives conclusions and directions for further work.

2. QoS mechanisms for multimedia streams

2.1. Error resilience

In MPEG-4 encoding, high compression is achieved at the cost of low error resistance and therefore, several optional error resilience mechanisms are foreseen by the standard [7], such as *resynchronization* and *data partitioning*. Resynchronization is based on the incorporation of special markers into the bitstream for localizing errors and regaining synchronization between the encoder and the decoder. Data partitioning is based on the separation of shape and motion data from texture data, for each macroblock, thus allowing one to be recovered even if the other is lost [8].

2.2. Rate adaptation

The issue of making media streaming applications rate adaptive has raised a significant research interest, with the focus being on achieving graceful quality degradation during congestion. In this direction many mechanisms have been proposed mainly suited for homogeneous IP networks, both for unicast, and multicast transmission.

Among them the *Loss-Delay Adaptation Algorithm* (LDA) [9] has been proposed and extensively tested.

LDA is a continuous rate, sender-based *Additive Increase Multiplicative Decrease* (AIMD) adaptation scheme. It relies on information like packet loss rate, calculated *Round Trip Time* (RTT) and an estimation of the available bandwidth. The transmission rate of the sender is adjusted in conformance with the TCP throughput equation. The responsiveness of LDA is determined by the frequency of the receiver's RTCP feedback messages. The maximum frequency of RTCP feedback can be one per 5 seconds [10]. The sender target rate is estimated by taking into account the receiver loss rate, the *round trip time* (RTT) and the bottleneck bandwidth (b). Whenever the receiver reports packet losses, the sender reacts by decreasing the target rate in a rather TCP similar fashion. In particular, if the current target rate is $`r$, and l is the fraction of the lost packets, the new target rate, r_{LDA} is given by the equation:

$$r_{LDA} = `r * (1 - l * R_f) \quad (1),$$

where R_f is a reduction factor (usually between 2 and 5). The R_f factor determines the degree of the reaction of the sender to losses. On the other hand, during periods where no packet loss is observed, the sender increases the target rate by a quantity equal to the *Additive Increase Rate* (AIR). The new AIR_i value is estimated through the following equation:

$$AIR_i = AIR * (1 - `r/b) \quad (2),$$

where `r is the current transmission rate and b is the estimated bottleneck bandwidth. AIR is set usually to a small value, around 10kbps. Further, AIR_i is limited to the average rate increase r_{incr}, similar to the increase rate that a TCP connection would utilize during a period time without packet loss, equal to the interim between two RTCP messages reception. If M is the packet size, T is the reporting period between two RTCP packets then:

$$r_{incr} = M * \frac{(T/RTT + 1)}{2} \quad (3)$$

Consequently, the new transmission rate is given by the following equation:

$$r_i = r + AIR_i \quad (4)$$

2.3. Differentiated Services

The Differentiated Services framework supports the differentiation of packets, not only belonging to different competitive streams, but also within the same stream, therefore defining several priority levels. This is achieved by marking the Type of Service (ToS) byte [11] of each packet header. This feature can be exploited in applying packet differentiation policies to media streams based on their semantics, in order to achieve graceful quality degradation, as described among others in [12], [13], [14], [15], [16],[17], and [18].

The problem of defining open QoS frameworks for multimedia streaming platforms has not been exhausted yet, although several studies have appeared in the literature ([19], [20]). Another step towards this direction is the Delivery Multimedia Integration Framework – DMIF [21], defined as a part of the MPEG-4 standard.

The framework proposed in [19] relies upon a dynamic QoS monitoring scheme. This scheme is based on ATM specific network feedback mechanisms, in order to guarantee synchronization requirements of SMIL presentations. In [20] the "QCompiler" programming framework is presented and experimentally evaluated for quality aware ubiquitous multimedia applications. Four layers are defined: (1) a *high-level application specification* layer allowing the user to specify application quality requirements, (2) a *metadata compilation* layer which compiles the quality requirements of layer 1 to a quality specification, (3) a *binding* layer which prepares a quality specification to be executed in a specific environment and (4) a *run-time metadata execution* layer, which uses the bound quality specification, to manage and control a quality aware multimedia application.

3. The extended MPEG4IP platform

3.1. Existing features

The MPEG4IP project [22] is an open-source (C/C++ based) platform, mainly developed by Cisco Inc. incorporating additional open source tools from other parties. It provides a standards-based, end-to-end platform for encoding, decoding, and streaming, over the UDP/RTP protocol stack, MPEG-4 audio/video streams. The client side mainly comprises the player and the content decoders, while the server side comprises the following components:

- A toolkit for off-line encoding of MPEG-4 compatible streams (Xvid or Divx [23][24] or ISO MPEG-4 simple profile video [7] and AAC audio [25]). In this package a utility for incorporating hint information inside the mp4 file metadata description is also included (*mp4creator*).
- An application (*mp4live*) for capturing, real-time encoding, and streaming video and audio content.
- A streaming server, in this case, the open-source

Apple's *Darwin Streaming Server (DSS)* [26]. However, in its present form MPEG4IP does not provide any mechanism for QoS control, neither at the application level nor through interfacing with a networking middleware.

Implemented extensions concern both *Best Effort* (BE), and Differentiated Services networks. Each one of these cases is examined in the subsections that follow.

3.2. Extending MPEG4IP with the LDA scheme

The MPEG4IP platform was extended in order to provide rate adaptation according to the LDA scheme. The elements that had to be added included a set of modules on the sender side, namely a *rate adaptation*, a *fidelity*, a *feedback receiver*, and an *adaptive encode*. On the receiver side a packet loss monitor and a feedback sender were required. These were implemented on the m4player and the mplive, applications of the MPEG4IP suite. In greater detail:

- *Rate Adaptation*. This module implements the rate adaptation scheme using as inputs the parameters mentioned in the previous subsection. Specifically, it collects feedback reports from the receiver application and outputs a target rate.
- *Fidelity*. The Fidelity module configures appropriately the encoder in order to achieve the optimal perceived quality corresponding at the target rate provided by the rate adaptation module. This target rate can be achieved using alternative encoder configurations. The configuration policy depends on the application kind. Candidate parameters for achieving the target bit rate can be the video picture size, the frame rate and the encoding quality.
- *Adaptive Encoder*. The Xvid encoder software of the *mp4live* application has been enhanced in order to be able to produce, in real time, content under different configurations, as these are provided by the Fidelity Module.
- *Feedback receiver*. The sender collects the receiver's reports, which contain information about packet losses. Moreover it calculates the RTT and bottleneck bandwidth.

On the receiver side the modules that implemented were:

- *Packet loss monitor*. Losses are indicated by missing sequence numbers on the received rtp bytestream The loss monitor module detects packet losses on the RTP byte stream level. These losses are collected and the fraction of packets lost (l) in the latest measurement period, is calculated. A Receiver Report feedback mechanism is implemented in order to report these losses every 5 seconds, in conformance with the RTP specifications.
- *Feedback sender*. By using this module, feedback is sent to the stream's sender. RTCP receiver reports together with RTCP application specific extensions can be used. However, any other feedback mechanism is also applicable.

Furthermore, this sender is interoperable with receivers that do not implement the LDA receiver side extensions. The upcoming subsection presents the implementation details concerning of this scheme.

The xvid encoder has been modified, allowing the generation of streams with variable target bitrate. The optimal encoder configuration that maximizes user perceived quality, while producing the desired target rate is decided by the fidelity module. In order to optimize the perceived quality of the received video, the target bitrate is achieved by adapting both the encoding and the frame rate. As the target rate drops, the frame rate also drops, in order to keep the per frame quality almost constant (this configuration is mainly suitable for a video with limited image motion). The frame rate and the target bitrate affect the number of the coefficients produced during the encoding process, and thus the number of bytes allocated for each encoded frame. Consequently, by suitably altering the frame rate and encoding bitrate parameters, the quality of each encoded frame is implicitly modified. For example, for a CIF (Common Intermediate Format) video stream which is used in the relevant experiments, the frame rate f is calculated as follows:

$$f = 0.0267 * bitrate + 1.667 \quad (5)$$

Equation 5 was determined through subjective quality tests. The minimum achievable encoder rate for the current implementation is 50kbps and the maximum 500kbps, thus the applications' frame rate ranges from f = 2fps to f = 15fps. Nevertheless these parameters are quite open to modification.

The RTP packetization of the video stream conforms to RFC3016 [27]. While encoding parameters vary throughout the session, the RTP session remains the same, therefore achieving a smooth play out at the receiver.

3.3. QoS extensions implemented for Differentiated services networks

The layering described in subsection 2.3 is adopted in order to enable the translation of stream semantics information to QoS metadata specifications and their mapping to DiffServ class information during transmission. In this way, QoS policies based on stream prioritization [19] and content-aware packet marking (see subsection 2.1) are possible.

3.3.1. The high-level application specification layer

Towards defining class-based information within a media stream, the role of each semantic entity produced during encoding needs to be identified. Therefore, a tool for studying the effects of quality violations (e.g. losses, delays) has been developed within the MPEG4IP platform. With this tool, the user can impose fully controllable violations on a media stream (or groups of streams) and preview their effects before transmission. This is achieved on a packet basis, by delaying or preventing a packet from entering the output network interface, based on the payload of the packet (e.g. flags, motion vectors, coefficients). These deliberate QoS violations can be configured to correspond to realistic conditions, such as bursty losses and delays, which may cause several consecutive frames to be discarded.

3.3.2. The Metadata compilation layer

At the ***Metadata Compilation*** layer, high-level stream quality requirements defined at the previous level are incorporated into lower-level metadata descriptions, produced during the encoding process. Two cases are examined, *live* and *pre-encoded* streams.

In the first case, the *mp4live* application was enhanced with packet differentiation extensions. A dedicated packet marking component was implemented which is capable of differently labelling parts of the produced bitstream with arithmetic values corresponding to different quality levels, according to the contribution to quality. These values will be assigned to the corresponding packets as ToS values by the binding layer, during packetization. A packet marking policy can be defined for unequal error protection of the stream by assigning, for example, different service priorities to motion and texture information in a data-partitioned stream [7].

In the case of pre-encoded streams, the DSS server is used to packetize and transmit stream packets over the network. Because the server is unaware of the MPEG-4 payload, the packetization process is realized based on *metadata* descriptions, which are incorporated into the media file during encoding and describe media data *by reference*. In MPEG4IP this is done with the *mp4creator* utility. The metadata format is according to [1].

Metadata structuring is tree-like and is based on the concept of ***atoms***. The general structure of the MP4 file is illustrated in figure 1.

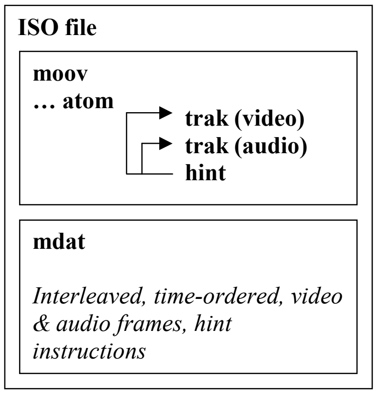

Figure 1 - The mp4 file structure

The ***mdat*** atom abstracts the structure holding the actual media data. The ***moov*** is an atom whose sub-atoms define the metadata for a presentation. Every media track has its own timeline, samples (e.g. frames) and properties. One or more sample descriptions can be defined based on how the sample must be decoded. The mp4 file, also, describes how to synchronize the timelines of the tracks and the aggregate properties of the tracks. *Hint tracks* contain instructions for the streaming server on how to packetize media track data for transmission, e.g. based on RFC 3016 as it is in the MPEG4IP case.

The MPEG4IP platform provides an API, referred to as *MP4 library*, for creating and modifying MP4 files as defined by [1]. To accommodate applications that need access to information not otherwise available via the API there are file and track level generic *get* and *set* property routines that use arbitrary string property names. Also, a set of utilities is provided to inspect the metadata information created with every encoded MP4 file, such as the MP4Dump application.

This API is exploited for incorporating stream QoS specifications, which are based on stream semantics, into the metadata descriptions. Specifically, within the hint track information every RTP packet is described by a set of fields, giving for example, its size and its offset from the VOP start. A new field needs to be inserted corresponding to the quality level each packet is assigned at, depending on its payload (e.g. motion or texture data in the data partitioning case). Then the DSS packet-marking component reads from the hint track the QoS specifications and maps them to DiffServ compliant packet TOS values. This procedure is illustrated in figure 2.

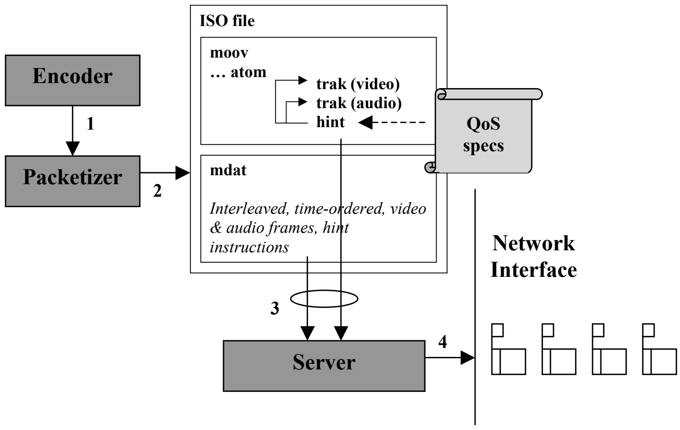

Figure 2 – Mapping of QoS specifications to ToS values

The binding layer

The ***Binding*** layer takes as input the QoS specification and prepares it to be executable in a specific environment (e.g. by mapping a network service interface to a QoS enhanced socket system call). For example, in the case of a DiffServ network, packet differentiation and assignment to network classes is achieved through appropriate packet marking of the ToS byte of the packet header, with a value corresponding to a Per Hop Behavior (PHB) [28], which in turn corresponds to a QoS level.

The run-time metadata execution layer

In the ***Run-time Metadata Execution*** layer the metadata descriptions are parsed and different QoS levels are assigned to different network classes, e.g. through DiffServ compliant ToS marking performed by the streaming server during the packetization process.

ToS marking is realized by exploiting the *setsockopt()* function of the socket networking API. DiffServ compliant packet marking is performed by setting the six bits of the Differentiated Service (DS) byte to the appropriate DS CodePoint value, corresponding to Assured Forwarding (AF) [29] classes AF11 (0x28), AF12 (0x30) and AF13 (0x38).

4. Conclusions and further work

In this paper, application extensions for enhancing the MPEG4IP media streaming platform with Quality of Service (QoS) capabilities were presented. Based on these extensions, a set of supported QoS mechanisms for Best Effort and Differentiated Services networks were experimentally evaluated under different configurations. Quality impairments were captured by a quality monitoring module and suitable metrics were extracted in each case.

There are several directions for further research built on the work presented. The QoS aware MPEG4IP platform can be exploited as an experimental environment. Within this it is possible to simulate realistic application scenarios with many streams [19] and highly variable traffic patterns. In this way, complex traffic profiles will be thoroughly studied and they will be related with quantitative parameters. Therefore, the behaviour of different QoS policies can be examined, together with the compilation of portable metadata QoS descriptions. Quality metrics that parameterize perceived quality will be further explored.

The existing QoS mechanisms can be enhanced and others can be incorporated into the platform, such as rate adaptation mechanisms for multicast transmission, or signalling for resource reservation.

5. References

[1] ISO/IEC 14496-6, "Information technology – Coding of audio-visual objects", Part 1: Systems

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998

[4] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", IETF RFC 1633, 1994

[5] V. Marques, R. Cadime, A. de Sousa, A.M. Oliveira Duarte, "DMIF based QoS Management for MPEG-4 Multimedia Streaming: ATM and RSVP/IP Case Studies", ConfTele 2001

[6] P. Bocheck, Y. Nakajima and S. F. Chang, "Real-time Estimation of Subjective Utility Functions for MPEG-4 Video Objects", Proceedings of the Packet Video'99 (PV'99), New York, USA, April 26-27, 1999

[7] ISO/IEC 14496-2, "Information technology – Coding of audio-visual objects, Part 2: Visual".

[8] S.T. Worrall, S. Fabri, A.H. Sadka, A.M. Kondoz, "Prioritisation of Data Partitioned MPEG-4 Video over Mobile Networks", ETT-European Transactions on Telecommunications, Vol. 12, No. 3, May/June 2001

[9] Dorgham Sisalem, Henning Schulzrinne "The Loss-Delay Based Adjustment Algorithm: A TCP-Friendly Adaptation Scheme", NOSSDAV'98, Cambridge, UK.

[10] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC3550, July 2003.

[11] W. R. Stevens, "UNIX Network Programming", Volume 1, second edition, ISBN 0-13-490012-X

[12] E. Masala, D. Quaglia, J.C. De Martin, - "Adaptive Picture Slicing for Distortion-Based Classification of

Video Packets", 2001 IEEE Workshop on Multimedia Signal Processing, Cannes, Francia, pp. 111-116

[13] M. Miska, Hannuksela, Ye-Kui Wang, and M. Gabbouj, "Sub-picture: ROI coding and unequal error protection", IEEE 2002 International Conference on Image Processing (ICIP'2002), Sept. 2002, Rochester, New York, USA

[14] W. Tan and A. Zakhor, "Packet classification schemes for streaming MPEG video over delay and loss differentiated networks", Proceedings of Packet Video Workshop, Kyongju, Korea, April 2001

[15] T. Tian, A. Li, J. Wen, and J. Villasenor, "Priority Dropping in Network Transmission of Scalable Video", *IEEE International Conference on Image Processing*, Volume III, pp. 400-403, September 2000

[16] J. G. Kim, J. W. Kim, and C.C. Jay Kuo, "Internet video packet categorization with enhanced end-to-end QoS performance", in *Proc. SPIE Visual Communications and Image Processing `2002*, San Jose, CA, Jan. 2002

[17] F. Wang, P. Mohapatra, "Using Differentiated Services to support Internet Telephony", Elsevier Journal Computer Communications 24 (2001) 1846-1854

[18] N. Cranley, P. Perry and L. Murphy, "User-Perceived Quality-Aware Adaptive Delivery of MPEG-4 Content", Proc. NOSSDAV 2003: 13th International Workshop on Networking and Operating Systems Support for Digital Audio and Video, Monterey, California, June 1-3, 2003

[19] C. Chang and S.W. Hsieh, "An Adaptive QoS Guarantee Framework for SMIL Multimedia Presentations with ATM ABR Service", IEEE Globecom 2002

[20] D. Wichadakul, X. Gu, K. Nahrstedt, "A Programming Framework for Quality-Aware Ubiquitous Multimedia Applications", ACM Multimedia 2002

[21] V. Marques, R. Cadime, A. de Sousa, A.M. Oliveira Duarte, "DMIF based QoS Management for MPEG-4 Multimedia Streaming: ATM and RSVP/IP Case Studies", ConfTele 2001

[25] ISO/IEC 14496-3, "Information technology – Coding of audio-visual objects", Part 3: Audio"

[27] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, H. Kimata, "RTP Payload Format for MPEG-4 Audio/Visual Streams", RFC3016, November 2000.

[28] D. Black S. Brim B. Carpenter F. Le Faucheur, "Per Hop Behavior Identification Codes", RFC 3140, June 2001

[29] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999