The Journal of Systems and Software 83 (2010) 1489–1499

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Quality adaptive end-to-end packet scheduling to avoid playout interruptions in Internet video streaming systems ☆

Rossella Fortuna, Luigi Alfredo Grieco *, Gennaro Boggia, Pietro Camarda

"Dip. di Elettrotecnica ed Elettronica (DEE)" at "Politecnico di Bari", V. Orabona 4, 70125 Bari, Italy

ARTICLE INFO

Article history:
Received 31 July 2009
Received in revised form 11 March 2010
Accepted 11 March 2010
Available online 31 March 2010

Keywords:
Bandwidth allocation
Video streaming
Layered video

ABSTRACT

In Internet multimedia streaming, the quality of the delivered media can be adapted to the Quality of Service provided by the underlying network, thanks to encoding algorithms. These allow a fine grained enhancement of a low quality base layer at streaming time. The main objective that should be satisfied in such systems is to avoid the starvation of the decoding process and consequent playout interruptions. In this work, we tackle the problem using a control theoretic approach. In particular, we design and implement the novel end-to-end Quality Adaptive Scheduler for properly distributing the network available bandwidth among base and enhancement layers. The developed solution can be adopted in many contexts given that it has been designed without assumptions on the delivered media nor on the protocol stack. Anyway, to test its effectiveness, we have casted it in a H.264/AVC SVC based video streaming architecture for unicast Internet applications. The performance of the scheduler has been experimentally evaluated in both a controlled testbed and several "wild" Internet scenarios, including also UMTS and satellite radio links. Results have clearly demonstrated that our Quality Adaptive Scheduler is able to significantly improve the performance of the video streaming system in all operative conditions.

1. Introduction

Multimedia streaming services are nowadays broadly deployed in different scenarios of social life, thanks to technological drivers such as the ever increasing availability of bandwidth, the evolution of advanced quality adaptive encoding systems, and the wide diffusion of mobile networking devices with high computational and storage capabilities (Cisco, 2008; Wu et al., 2001; Chang and Vetro, 2005; Schwarz et al., 2007; Radha et al., 2001).

Adaptation is the enabling key of such services (Rejaie et al., 2000). In fact, unpredictable packet losses, delays, and bandwidth fluctuations due to the intrinsic behavior of packet switching networks can be compensated by adapting the quality of the delivered media (Dai et al., 2006; Zhu et al., 2007; Djama et al., 2008). In these systems, the media is usually encoded in a base layer stream, representing a low quality version of the content, and one or several enhancement layers that can be progressively added or striped with fine-grained resolution, according to the Quality of Service (QoS) provided by the network (Schwarz et al., 2007; Radha et al., 2001). The main objective that should be satisfied in these systems is to avoid the starvation of the decoding process and the consequent playout interruptions (Saparilla and Ross, 2000). In this work, we tackle this problem using a control theoretic approach. In particular, we design and implement a new algorithm, which will be referred to as Quality Adaptive Scheduler (QAS). It runs at the application layer, properly distributing the network available bandwidth among base and enhancement layers. With QAS, the client feedbacks to the server the amount of buffered base layer, $Q(t)$, that has not yet been played out. By exploiting these feedbacks, the server regulates the base layer transmission rate, $R_b(t)$, in order to keep $Q(t)$ at a constant target layer Q_0, thus reducing the risk of playout interruptions. The remaining quota of available bandwidth, if present, is used to transmit the enhancement layer. In this way, when the available bandwidth is scarce, QAS is able to lower the risk of playout interruptions at cost of a reduced quality of the delivered media.

For sake of generality, QAS has been designed without making assumptions on the delivered contents nor on the protocol stack. Its effectiveness has been tested casting it in a H.264/AVC SVC (Advanced Video Coding, Scalable Video Coding) based video streaming architecture for unicast Internet applications (Schwarz et al., 2007). This choice is motivated by the fact that the H.264/AVC standard (Wiegand et al., 2003) holds a leading position due to its very high compression efficiency. As example, for the same

☆ This work was funded by the Projects PS-121, PS-092, and PS-025 (Apulia Region, Italy). A preliminary version of this work appeared as: G. Boggia, P. Camarda, R. Fortuna, L.A. Grieco, A scheduling strategy to avoid playout interruptions in video streaming systems, Proc. of Int. Conf. on Computer Commun. and Networks, ICCCN, San Francisco, August 2009.

* Corresponding author.
E-mail addresses: r.fortuna@poliba.it (R. Fortuna), a.grieco@poliba.it (L.A. Grieco), g.boggia@poliba.it (G. Boggia), camarda@poliba.it (P. Camarda).

0164-1212/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2010.03.037

video quality, H264 coding shows an average bit-saving of 30% with respect to the classical MPEG-2 compression standard (Haskell et al., 1997). The Joint Video Team (JVT) working group is now developing the scalable extension of H.264 (Schwarz et al., 2007) (i.e., the H.264/AVC SVC) to add the flexibility of a scalable encoding strategy to the efficiency of H.264. This scalable coding also provides a great flexibility in partitioning the video stream (Choi et al., 2007; Wenger et al., 2005, 2008). In fact, as in the not-scalable H.264 version, the stream is fragmented in elementary Network Abstraction Layer (NAL) units that semantically differ from each other, so that portions of a stream containing different information types can be easily identified and treated, e.g., timing informations, parameter set, enhancement layer data or base layer data (Wenger et al., 2005, 2008). To this aim, the Fine Granularity Scalability (FGS) is used, which allows the server to truncate enhancement layer at any point (Schwarz et al., 2007). Using FGS, the server can finely regulate the enhancement layer transmission rate for a full exploitation of available bandwidth. The client/server architecture, that has been implemented, employs TCP (Postel, 1981) protocol at transport layer. Even if UDP is generally preferred for transmitting multimedia flows (Papadimitriou et al., 2008), this choice is not unusual, as testified by many analogous examples, either in literature (Gurses et al., 2005; Mehra et al., 2005; Argyriou, 2006), or in commercial applications (for instance the well-known *Youtube*, as asserted in Gill et al., 2007). One of the reasons motivating TCP choice is that many networks filtrate UDP flows allowing only TCP connections, as reported in RFC 4571 (Lazzaro, 2006). Furthermore, implementing QAS on top of TCP represents a challenging objective. In fact, since TCP uses a window-based congestion control (Postel, 1981), QAS has not an explicit estimation of the available bandwidth, and, as a consequence, NAL scheduling has to take into account cross-layer information coming from the transport layer. QAS performance has been experimentally tested using both a controlled testbed and several Internet scenarios, also including UMTS and Satellite links. Results have clearly demonstrated the effectiveness of the proposed approach in all considered operative conditions.

The rest of the paper is organized as follows: in Section 2, related works are summarized. In Section 3, the proposed video streaming architecture is described. In Section 4, QAS is designed. In Section 5, experimental results are presented. Finally, the last section draws the conclusions and forecasts future research.

2. Related works

Research on quality adaptive video streaming systems has been very active in the last years. Herein, we summarize the main contributions that we consider relevant for our discussion, which is mainly focused on unicast video streaming applications in the Internet.

In Saparilla and Ross (2000), authors present a theoretical discussion on the optimal repartition of the bandwidth among base and enhancement layers. The results of the study can be only applied to scenarios having a negligible Round Trip Time (*RTT*). Furthermore, no validation in realistic settings has been reported.

Many works on quality adaptive video streaming, such as Dai et al. (2006) and Kim and Ammar (2005), consider the problem of minimizing quality variations, in order to maximize perceived video quality. They only focus on enhancement layers delivery, under the assumption that base layer rate is smaller than the available bandwidth. This assumption could be unsafe because Internet bandwidth is variable and can even snap to zero during transients. Moreover, assuming that base layer rate is smaller than the available bandwidth, implicitly drives to encode the video using a very small base rate and a very high enhancement rate. The subtle effect of this choice consists in a reduction of the compression efficiency, which increases proportionally to the ratio between base rate and enhancement rate (Amon et al., 2008).

With our approach, assumptions on the instantaneous available bandwidth can be relaxed: QAS is able to avoid playout interruptions if the average available bandwidth is greater than the base rate, thus compensating transient bandwidth reductions.

In Zhu et al. (2007), an algorithm for the joint design of encoding rate control and congestion control has been proposed. At transport layer, a modification of the TCP Friendly Rate Control (TFRC) algorithm (Handley et al., 2003) has been designed, ensuring TCP friendliness in the long term. Constraints on both encoding rate and sending rate of the transport layer are derived using a Virtual Buffer approach (Xie and Zeng, 2004). The performance of the algorithm has been tested using ns-2 simulations in a four hop topology with short and long lived TCP connections as well as multimedia traffic sources.

A real-time TCP-based video streaming system is proposed in Gurses et al. (2005). Due to the real-time nature of the system, a discarding frame algorithm provides the sending of only frames that would arrive in-time at the client, according to an estimated end-to-end delay. Each frame is associated with a transmission deadline, after which it is discarded by the source. To assign a higher priority to base frames, the deadlines of enhancement frames are lowered when the available bandwidth is scarce. This mechanism is particularly suited if temporal scalability is used. It is not useful with FGS, because a mechanism to dynamically truncate FGS frames is not provided. As a consequence, it cannot exploit FGS capability to totally adapt enhancement rate to available bandwidth. Another limit of the study presented in Gurses et al. (2005) is that system performances have been evaluated only by ns-2 simulations.

To conclude, the main lacks of the aforementioned works are: (i) restrictive or simplistic hypotheses on the underlying network and/or on the allowed codecs; and (ii) the absence of a validation through real experiments on Internet.

In the present work, our objective is to design a robust and general mechanism for avoiding base layer buffer underflows, which can heavily degrade the quality of the media due to abrupt playout interruptions. QAS algorithm has been implemented and tested in many networking conditions to demonstrate its general validity. We believe that this proposal can be though as an important building block of a generic streaming architecture. In fact, it has been designed without making assumption on the delivered media or on the protocol stack. Thus, it could be integrated in the future to enforce recent proposals available in literature, such as those previously reported in this section.

3. Streaming architecture

The H.264/AVC SVC based video streaming system implemented is based on RTP/RTCP protocols (Real Time Protocol/Real Time Control Protocol) (Schulzrinne et al., 2003). TCP has been used at the transport layer. Working in an Unix Kubuntu-based environment, the default Kubuntu TCP implementation has been used, i.e., CUBIC TCP (Rhee and Xu, 2005). Live555 libraries (Live555, 2009) and the JVT Joint Scalable Video Model (JSVM) module (JSVM, 2009) have been properly modified to allow the implementation of QAS.

The client is made of three modules: the *network entity*, the *decoder*, and the *viewer* (see Fig. 1). The *network entity* receives RTP packets, reorders NALs in decoding order, and stores them in a decoding buffer. To enable the adaptation of the base layer transmission rate ($R_b(t)$), every Round Trip Time (*RTT*), using RTCP-APPlication-defined (RTCP-APP) packets (Schulzrinne et al., 2003), it also sends a feedback message to the server containing the actual base layer buffer level $Q(t)$. If $RTT > 200$ ms, the feedback is sent