Common Issues in Real-Time and Media Processing

Peter Altenbernd, Ditze
C-LAB
33094 Paderborn, GERMANY
peter.altenbernd@c-lab.de, michael.ditze@c-lab.de

Abstract

In this paper, we outline the possibilities offered by real-time research results in order to handle media streaming based on the Internet Protocol (IP).

On one hand the real-time community has been producing highly advanced methods to treat time-critical processes (like priority-based scheduling). Most of this work focuses on *hard* real-time issues, so it is guaranteed that no deadline will ever be violated.

On the other hand the networking community is dominated by the recent IP technology success. Extended with corresponding *Quality-of-Service (QoS)* models with well-defined standards, IP is even used for media streaming. However, some of these models offer just very soft timing guarantees. Further, scheduling is not as advanced as in the real-time community, and end-system QoS is often neglected.

The main contribution of the paper is that we show how the results of both areas can be combined, in order to provide reliable high-quality media streaming in IP networks. We will outline to what extent available real-time theory can be used, and how it must be combined with the standards given by the networking community.

Keywords: Media Streaming, MPEG, Quality of Service, Scheduling

1 Introduction

The widespread introduction of *Internet Protocol (IP)* technology over the past few years makes it possible to offer professional and private users a host of different services. The supply of continuous data streams (video and audio (V/A)) is an increasingly important part of this. However, these media involve an enormous volume of data and jitter-free display involves time restraints.

Typical examples of time-sensitive applications of this kind are: e-commerce applications (like virtual shopping malls), video-on-demand (VoD), teleconferences, IP telephony, distributed architecture, story board design in film production, TV broadcasting, training courses (CBT). Providing QoS guarantees to these scenarios, increases user acceptance in a drastical way.

A signal with six digital TV channels (typical for satellite technology) requires a transmission capacity (with MPEG-2 compression) of 19.2 Mbit/s, which means that it takes about 3 Mbit/s to transmit a movie of PAL quality. Even if data volumes on this scale for the IP domain appear largely wishful thinking for the moment, they will certainly be achievable in the foreseeable future as network technology brings progressive improvements, and they are therefore not part of the problem area dealt with here.

Whether or not to include time response in this context presents a quite different yet fundamental problem. Traditionally, IP technology only includes strategies that cannot provide any delivery guarantees whatsoever for the media streams transmitted (Quality-of-Service (QoS)). With today's pure IP technology, packet losses and time scatter often lead to disturbances with the transmission of V/A material in the form of losses and jitter. So, the least thing to do is to add a QoS concept.

This work offers a new concept for the transmission of such time-critical data with *Quality-of-Service (QoS)* constraints via IP technology. Special focus is on the efficient delivery of MPEG-2 streams encoded with variable bitrates including both the network and the end systems. The novel approach presented here takes advantage of classic real-time experience which considerably increases resource exploitation compared to classic transmission methods. The resulting benefit (i.e. applications gain either the same performance with less cost, or performance increases on the same cost level) can be exploited by a number of commercial applications.

In the following paper we describe how packet losses and time disturbances can be prevented by reserving bandwidth in dedicated subnetworks (using known QoS models). Particular attention is devoted in this connection to the use of techniques from classic real-time theory. Our techniques will allow resource capacities to be utilized considerably more effectively. While doing this we actually employ hard real-time methods for handling a soft real-time problem.

The rest of the paper is organised as follows. In the next section, we give a brief introduction to the application context and existing QoS models. In Section 3 we show how advanced real-time theory can be combined with existing QoS models, in order to provide reliable high-quality media streaming in IP networks. We will outline to what extent available real-time theory can be used, and how it must be combined with the standards given by the networking community. In Section 4 we give our conclusions.

2 QoS Models and Media Streaming

Two different approaches have been taken in the past in order to be able to provide QoS guarantees in IP networks. Both have been defined by the Internet Engineering Task Force (IETF): *Integrated Services (IntServ)* [3], *Differentiated Services (DiffServ)* [2]. Both approaches look very promising as regards their potential use in practice.

IntServ [3] is based on the principle of reserving bandwidth for each data stream in the system. With an exclusive bandwidth it is possible to provide valid QoS guarantees, since mutual disturbances are ruled out. To achieve this, all network elements (including transmitters (servers), various routers, receivers (clients)) that are involved in the transmission process must support the mechanisms needed for this. Before a data stream is transmitted, it is always preceded by an admission control that checks whether all elements still have sufficient capacity. Transmission is only allowed if this applies everywhere. The advantage of guaranteed QoS with the IntServ approach is offset by the disadvantage of poor scalability, since it would appear to be impossible to administer the status information for several thousand streams (as is possible in the Internet) on every router. IntServ is thus more suitable for intranets or extranets.

The **DiffServ** [2] approach counters the problem of scaling by combining data streams into particular priority classes. Admission control is handled in a heuristic fashion by a central entity (called Bandwidth Broker), as opposed to the accurate path-oriented view of IntServ. Since detailed status information is not transmitted for any of the streams, it is unfortunately not possible to provide any real guarantees with this approach, although transmission quality and efficiency are improved significantly.

Today, the DiffServ approach is the focal point of general research interest for the network community. Only a few players are going for differentiated handling of global (Internet) and local area networks of limited size (intranet/extranet). The work described here is, however, concentrated particularly on these local area networks using IntServ, because they are subject to their own administration and have become increasingly important in recent times. Examples of this are networks in the field of e-commerce, edutainment and networks of companies in multimedia or film production. Networks of this type do not suffer at all from the problem of scaling and can be handled very efficiently, as our work demonstrates.

A suitable signalling protocol, the *Resource Reservation Protocol (RSVP)*, has been developed (likewise by the IETF) as a means of reserving resources for data streams as part of the IntServ concept. RSVP is not responsible for the actual implementation and utilization of the reserved bandwidth. These functions (e.g. control of the time sequence (scheduling) of individual data packets) are performed by corresponding network modules that can be designed relatively flexibly. Unfortunately, RSVP is bandwidth-oriented, which can result in very inefficient utilization of the capacity of resources, as is described in the following.

V/A streams are coded using compression techniques such as *MPEG-2* (from the Motion Pictures Expert Group) in order to enable them to be transmitted digitally. These techniques usually achieve their highest compression rate in conjunction with a *variable bit rate (VBR)*. However, this results in considerable size differences between the largest image and the other images in a video stream. In the examples we looked at, we determined that the average size of images is under 35% of the maximum [1]. Unfortunately, it is this maximum sized image that has to be taken as the basis for the RSVP reservation (peak-rate allocation). In our example, this means that 65% of the reserved resources cannot be utilized.

3 Addressing Media Streaming with Hard Real-Time Methods

We solve the above mentioned efficiency problem by using scheduling and analysis techniques from the domain of classic real-time systems.

In the past, real-time systems were considered predominantly for control applications such as anti-skid systems in vehicles. However, it is now widely accepted that the supply of a video stream with 25 images a second presents fundamentally the same problem as the implementation of a periodic control algorithm that continually calculates new manipulated variables (e.g. braking pressure with anti-skiding) at fixed time intervals. Naturally multimedia systems are not comparable with systems where safety is a critical factor, but service providers are nevertheless under strong pressure to guarantee promised QoS in practice. Generally speaking, it can be said that a good real-time design does not only offer guarantees a priori; it also frees the developer from the need to compensate for time problems by using overdimensioned hardware. The idea presented here could therefore potentially bring about a sustained improvement in the efficiency of IP components.

Novel mechanisms for admission control and packet scheduling on routers and servers hold the key to boosting efficiency, since a bandwidth-oriented procedure does not allow access to over-reservations which are lost for other data streams. The new real-time techniques use detailed information (i.e. other parameters) about the V/A streams. If time constraints permit, data packets are therefore buffered.

There are a couple of individual problems which arise with the realization of the new concept, as discussed in the following sections.

3.1 Parameters in the RSVP Scenario

Real-time theory like Response-Time Analysis, which we use for admission control (see Section 3.2), is based on the knowledge of a set of parameters, such as *period lengths, transmission times, deadlines*, etc.. These parameters completely differ from those used the network community to describe traffic flows.

We therefore elaborated methods that derive the above mentioned parameters, which is fairly different from extracting them from a control programs (see Section 3.3). Furthermore, it was not planned for RSVP to support these parameters, so we designed on a scenario in which RSVP will be used (but not expanded) to transport them [6].

3.2 Scheduling and Admission Control

Scheduling of network traffic is not a direct function of the RSVP. Instead, one possibility is to modify appropriate network modules, such as the widely used Class Based Queuing (CBQ) [5] and server packetizers, in which a typical, priority-based real-time scheduler can be implemented. A scheduler assigns the priorities of individual packets according to their urgency. There are already a large number of different heuristic approaches to assigning priority (e.g. Earliest Deadline First (EDF)), all having the object of providing optimum treatment as far as possible.

New streams that enter the system have to undergo admission control, in order to make sure that the available resource are not overbooked. In its ordinary form (i.e. the approach taken in the RSVP scenario), admission control merely adds up bandwidths. If the total exceeds the available capacity, no further bandwidth can be assigned. According to the properties of the stream (like VBR) the amount of bandwidth needed is determined by using the *Token Bucket Model.*

By contrast, our procedures are based on *Response Time Analysis (RTA)* [6]. In our concept, admission control has to analyse the time response of all streams in the system (while simultaneously taking account of the scheduling method used). This is fully compliant to the RSVP scenario presented in Section 3.1, and we could show that this operation model is more efficient than using the Token Bucket Model. Further, our way of dealing with scheduling deals with both the network and end systems [4].

3.3 Worst-Case Execution Time Analysis of Media Streaming

Scheduling and admission control both require input parameters that need to be specially calculated. To derive these values for the network is relatively easy. However, in view of the particular difficulty of predicting calculation (i.e. for encoding and decoding) and transmission times, we apply techniques used in the analysis of *worst-case execution times (WCET)* [1].

Ordinary WCET tools estimate the worst-case execution time of a given arbitrary control program. In contrast algorithm for decoding or encoding are known in advance, so our tools try to estimate WCET values from the knowledge of the streaming data.

In turn, WCET analysis results and concepts can be used for both end-system HW dimensioning (e.g. amount of memory) and SW configuration (e.g. possible frame-rate). This is particularly useful, for answering questions like *"To what extent is my PC capable of doing SW MPEG encoding?"*.

3.4 End-System QoS

Even though there is a way to allocate sufficient bandwidth on network links, timing on end-systems

has not been considered yet, like in the case of a VoD Server handling more than just one stream at the same time. Consider an example consisting of two streams: *Stream A* (10mbps) and *Stream B* (20mbps) for which network bandwidth reservation has been made. Given that their delivery imposes a load of close to one-hundred percent on the server, this means that a non-weighted CPU/disk scheduling would offer just 15mbps for each stream, so *Stream B* cannot be given the guarantee it needs. Hence, a concept for mapping reservation requests to the end system's task model [4] is needed, which requires employment of an real-time operating systems.

This problem is very similar at streaming clients, even if there is normally just one stream. However, there are other system tasks competing with the access to resources. This even holds of unconnected devices like digital VCRs and settop boxes.

4 Conclusions

The work outlined in this paper describes the use of classic real-time techniques in the field of video/audio (V/A) delivery under Quality-of-Service (QoS) constraints in Internet Protocol (IP) networks. We showed that it is possible to apply these techniques all along the path from the sender to the recipient of a V/A stream, while being conform to commonly used networking standards. The focus is on intra- and extranets, which are becoming more and more important, offering both higher efficiency and a higher degree of predictability than ordinary approaches.

Our future work will also address the use of IP in the context of hard real-time control. For doing that, available QoS models could be used. However, those models were designed for media processing data, which differs a lot from control data.

References

[1] P. Altenbernd, L. Burchard, and F. Stappert. *Worst-Case Execution Times Analysis of MPEG-2 Decoding.* In *12th Euromicro Conference on Real Time Systems*, 2000.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *An Architecture for Differentiated Services.* Technical Report RFC 2475, December 1998.

[3] R. Braden, R. Clark, and S. Shenker. *Integrated Services in the Internet Architecture: an Overview.* Technical Report 1633, 1994.

[4] M. Ditze and P. Altenbernd. *A Method for Real-Time Scheduling and Admission Control of MPEG-2 Streams.* In *The Seventh Australasian Conference on Parallel and Real-Time Systems (PART2000)*, 2000.

[5] S. Floyd. *Notes on CBQ and Guaranteed Service*, 1995.

[6] S. Schneider and P. Altenbernd. *Combining MRTA and RSVP for Efficient Network Scheduling.* In *The Seventh Australasian Conference on Parallel and Real-Time Systems (PART2000)*, 2000.