Adaptive Video Streaming for Mobile Clients

Jens Brandt   Lars Wolf

IBR
Technische Universität Braunschweig
Mühlenpfordtstrasse 23, 38106 Braunschweig, Germany
{brandt|wolf}@ibr.cs.tu-bs.de

ABSTRACT

Digital video streaming has attracted large interest in research as well as in commercial areas in recent years. The evolution of digital video coding and broadband Internet access enables a large number of users to access high quality video streams with several devices varying from mobile phones to notebooks. However, digital video streaming still has high resource requirements concerning the transmission and decoding of the streams. Especially mobile devices often cannot comply with such resource demands. This paper briefly describes our multimedia gateway implementation which provides video adaptation to mobile clients by using multidimensional compressed domain transcoding mechanisms.

1. INTRODUCTION

Current mobile devices such as PDAs, Smart Phones or pocket multimedia players have a processor speed of about $300 - 600$ MHz, a memory size up to 256 MB and a screen resolution ranging from 128×160 to 640×480 pixels. If no special video decoding hardware is present on such devices, they are often incapable of decoding and displaying high quality video streams. Although the available resources of such devices are constantly increasing, this problem will remain in the near future since quality demands are rising as well. Due to the increasing heterogeneity of mobile devices a fine-grained adaptation to the capabilities of the devices is necessary, instead of statically providing video streams at certain different quality levels.

Because of its compressed nature digital video cannot be manipulated directly, but needs to be decompressed, adapted and compressed again. Obviously, this procedure is quite costly in terms of processing time. However, by the use of compressed domain transcoding techniques the processing time needed for decompressing and compressing can be saved at the expense of lower flexibility and higher complexity compared to tailoring the uncompressed video. Although transcoding can save processing time, the processing power

needed to adapt a stream is still quite high. Therefore, the adaptation should not be performed on the presenting device. Instead, we need some assistance from the network for this purpose, which also reduces the amount of data transmitted over the wireless link to the client. We propose to implement the adaptation engine into gateways which are placed close to the clients in the access networks. The advantage of this approach is its flexibility, because the adaptation service is available for all accessible media servers. Especially if the video stream is not exclusively delivered to one client, i. e., by the use of broadcast or multicast delivery, an individual adaptation of the stream is still possible.

2. SYSTEM OVERVIEW

Our target scenario is depicted in figure 1. The multimedia gateways are placed in an access network (dark-gray) which further contains several access points by which mobile clients can connect to the Internet. We assume that the mobile devices have network layer connectivity via the access points and that seamless handoffs between access points are supported. Media servers are located on the Internet and can be reached by the clients. Additionally, we assume that the network supports multicast communication between clients and gateways as well as between gateways.
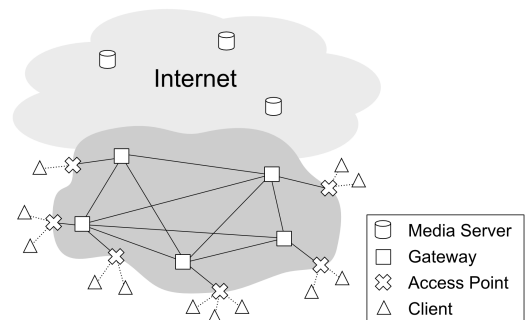

Figure 1: Target scenario

For application layer communication we are using the IETF standard protocols RTSP and RTP. The latter one is used for data transport, whereas RTSP is used for signaling purposes to create and control streaming sessions. Each client which would like to use the adaptation service needs to use one of the gateways as an RTSP/RTP proxy. Now the complete signaling as well as RTP data communication is sent via the gateway, which adapts the stream if possible.

At the time a client initially connects to the access network it is possible to set an initial proxy address. But this kind of static proxy configuration would not be sufficient to support mobility of the clients. If a client moves to a place where another gateway is available it could be reasonable to migrate the current session, as explained below. Moreover, a static configuration neither takes into account the workload of a proxy nor supports proxies with different adaptation capabilities (e. g. the desired target format is not supported by the nearest gateway). Therefore we propose to use a gateway discovery mechanism which enables clients to discover their surrounding gateways by using active service requests as well as passively listen to service announcements. This leads to a hybrid approach where clients can send service requests whenever they need to discover new gateways, and gateways periodically send service announcements. Both, service announcements and service requests, are sent to well-defined multicast addresses. Due to the fact that the usability of a gateway is topologically bound, we limit the range of service announcements by using a low time to live (TTL) for these messages.

To decide which kind of adaptation to use when serving a requesting client, the gateway needs to get the requirements from the client. RTSP, as defined in RFC 2326, does not support any kind of capability exchange regarding the playback capabilities of the client. Therefore, we decided to use the possibility of RTSP to exchange general parameters for this purpose. RTSP defines two methods called `GET/SET_PARAMETER` which can be used to exchange any user-defined data between both parties of an RTSP session.

3. IMPLEMENTATION

We implemented the proposed gateway as an RTSP/RTP proxy, which follows the paradigm of separated control and data paths as presented in [1]. This implementation includes the following features: hybrid gateway discovery, capability exchange between client and gateway, session transfer between gateways, and multidimensional transcoding at the gateway. For data transport within our proxy implementation we use small data processing units called StreamHandlers (SH) which build a data path through which the video stream is pipelined. Each SH can manipulate the processed video stream. By the use of a special SH the proxy is able to load a so-called subgraph into the data path. Thus, different transcoding modules can be loaded at runtime as a subgraph and then tailor the stream to the requirements of the client. If several users are watching the same video stream, our RTSP proxy can use a single control as well as a single data path between the server and the gateway and deliver the stream to each client, instead of establishing a separate session to the server for each client. In contrast to typical multicast delivery, the clients do not necessarily get the same data stream, because the proxy can tailor the stream individually for each client.

For gateway discovery we adapted the Service Location Protocol (SLP) by adding the ability of proactive service announcements from the gateway. Every gateway sends such an announcement periodically to a well defined multicast address to which all clients can listen. A client which receives such an announcement can update its list of available gateways and may chose an appropriate gateway if necessary. If a client does not know any appropriate gateway it may send a service request to actively search for reachable gateways.

With this hybrid approach clients are able to request for reachable gateways and also react to appearing gateways, which may occur due to movements of the client.

Whenever a client choses to use a gateway it has to send its capabilities to this gateway. We implemented the capability exchange between clients and gateways by using the optional RTSP method `SET_PARAMETER` which contains a client's profile. To define such a profile of a client that includes its capabilities as well as the user's preferences we used the Composite Capabilities/Preference Profiles (CC/PP) [3].

Due to the mobility of the clients it may occur that a client would prefer a newly available gateway to its current one. In this case the existing streaming session needs to be transfered from the old to the new gateway. To support such a gateway hand-over we extended our signaling protocol to be able to signal the whole hand-over process in RTSP.

For the video adaptation itself we use our multidimensional transcoding architecture which we presented in [2]. Currently we have implemented a transcoding module which can adapt an MPEG-4 ASP video stream in the following three dimensions: temporal, spatial and detail resolution. The transcoding parameters used during the transcoding process are based on the client's profile.

Our client application uses the same code basis as the gateway for RTSP and RTP handling and the popular multimedia player MPlayer for video decoding and display. Therefore, we developed and integrated a new input module into the mplayer, which uses the same RTSP/RTP implementation as the gateway.

4. DEMONSTRATION

In the demonstration we would like to show the aforementioned adaptation features of our gateway implementation. For the setup we plan to use two separate gateways serving at least two clients and using our multidimensional transcoder implementation on the gateways. Thus, we can show on demand adaptation for different requirements of the client. Further, we would like to provide an insight into the technical details of our implementation.

5. SUMMARY

The presented architecture and implementation of our multimedia gateway system provides individual video adaptation to users with mobile devices. By the use of a hybrid gateway discovery mechanism it can be used automatically by the user without any setup. Further the mobility of the user is supported by the use of a gateway handoff mechanism. The adaptation of MPEG-4 ASP streams is achieved by using multidimensional transcoding mechanisms.

6. REFERENCES

[1] J. Brandt, V. Kahmann, and L. Wolf. A flexible reflector for media streams. In *KIVS*, volume PI-61 of *Lecture Notes in Informatics*, pages 41–48, Mar. 2005.

[2] J. Brandt and L. Wolf. Multidimensional Transcoding for Adaptive Video Streaming. In *Proceedings of the 17th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'07)*, Urbana, Illinois, June 2007.

[3] World Wide Web Consortium (W3C). Composite Capabilities/Preference Profiles (CC/PP). http://www.w3.org/Mobile/CCPP/.