# Adaptive Wavelet Video Filtering

Andreas Kassler[1], Christian Kücherer[2], and Andreas Schrader[2]

[1] University of Ulm, Distributed Systems Department, Ulm, Germany
Andreas.Kassler@informatik.uni-ulm.de,
[2] NEC Europe Ltd., Network Laboratories, Heidelberg, Germany
Christian.Kuecherer|Andreas.Schrader@ccrle.nec.de

**Abstract.** In this paper we propose a flexible adaptive media streaming mechanism based on wavelet encoded video streams. By using a combination of sender rate adaptation and filtering inside the transmission path, an appropriate tradeoff between flexibility, efficiency and security can be achieved. The media adaptation mechanisms are one part of the MASA Quality-of-Service framework, which provides end-to-end QoS enhanced multimedia communication and includes local resource management. We provide examples for syntactial and semantical filter algorithms, show the interaction of media adaptation and QoS management and present experimental measurement results.

## 1 Introduction

The Internet offers challenging new opportunities for the development of communication applications such as real time video conferencing, multimedia distance learning and many more. The growing number of different fixed line (e.g. Ethernet, ATM, xDSL, etc.) and wireless (e.g. GSM/GPRS, Wireless LAN, UMTS, etc.) access network technologies in combination with the growing variety of user devices (e.g. mobile phone, PDA, Multimedia Workstation, etc.) will lead to a high level of heterogenity. Another source of heterogenity is imposed by the fact, that users should have the possibility of defining their own QoS policies with parameters like color resolution, frame size or frame rate.

In each of these categories, static and dynamic variations can occur. Transmission links do have a maximum available bandwidth, but the actually available bandwidth will vary over time depending on the overall usage. Especially, when mobile terminals perform intra- and inter-technology handoffs, transmission link characteristics will show sudden significant variations. Similar dynamic variations could also appear for terminal capabilities and user QoS policies.

Therefore, adaptivity is necessary to provide a high flexibility. Adaptation of media streams is the most important task as it influences user perceived quality as well as resource requirements both for the endsystems and for the network. Sender rate adaptation offers the highest flexibility, since the full semantical information of the data source is available. In addition, network node based media adaptation can adapt the stream to the aggregated demands of the following subtree, which bridges the heterogeneity gap for multicast scenarios. Mid-term, slow adaptation to the static demands of the session can be achieved by using appropriate feedback mechanisms in a distributed set of media adaptation nodes. To deal with highly dynamic changes during the transmission, simple filtering mechanisms inside routers should be used in addition to allow for short-term, very fast, locally optimized reactions and seamless handoff performance on mobile clients (e.g. in case of queue buffer overflows). Finally, user perceived QoS needs to be managed and maintained end-to-end. This requires local, peer and network resource reservation, adaptive media magement to react properly to QoS violations and a coordination of the activities to comply to users requirements. As it is too complex to manage these tasks by an application itself, a QoS framework (like the MASA QoS framework [5]) is necessary that applications can use to provide QoS enhanced adaptive multimedia services towards the user.

Several adaptive codecs have been proposed (e.g. layered DCT [2], *lH.261* [3]), but they often suffer from the flexibility for the support of different receiver requirements. Recently, Wavelet based coding mechanisms have become very popular [1]. In this paper we are using the WaveVideo codec developed at the ETH Zürich [4], which offers a good compromise between compression ratio, error tolerance, implementation efficiency, and the ability to support various filter operations. By producing a layered hierarchical coding scheme, WaveVideo is able to support prioritization of packets.

There are different possibilities to realize scaled media streaming. In the receiver driven layered multicast (RLM) approach, proposed in [11], receivers can decide about the quality of the stream by joining or leaving the respective multicast groups, which contain different layers of information. However, the most important problem of fixed layer subscription schemes is the unability to support different receiver QoS policies at the same time without inserting redundant layers. Therefore, we propose to use a single (multicast) stream which contains the aggregated sum of all requested data parts for the respective subtrees.

In this paper we propose two different video adaptation strategies. A simple, but efficient filter is based on pure syntactical information. A set of semantical filters allow for highest flexibility. We will demonstrate the use of both filter types in a multicast scenario with a single QoS policy and in a unicast scenario supporting multiple QoS policies.

This paper is organized as follows. In section 2, we give an overview of the MASA QoS framework. A set of syntactical and semantical filter algorithms for WaveVideo streams is presented in section 3. In section 4 we describe two example implementations of our adaptive media management system. Section 5 reports on some first measurement performed for two application scenarios in a wireless environment. Finally, we conclude our paper and present future work.

## 2 The MASA QoS Framework

The *MASA* QoS framework (Mobility and Service Adaptation in Heterogeneous Mobile Networks) is a joint project of NEC Europe Ltd. Heidelberg, Siemens AG Munich and the University of Ulm [5]. The *MASA* framework was designed to fulfill the requirements of a *comprehensive integrated end-to-end QoS multimedia management system*, allows the usage of underlying network layer QoS technologies, and hides the complexity of QoS and adaptive media management from the applications. By controlling the complete communication infrastructure *MASA* is able to support QoS in a way which can neither be realized inside the applications nor with the underlying network QoS technologies alone.

### 2.1 Architectural Overview

The *MASA* QoS architecture consists of a distributed set of autonomous QoS Brokers (fig. 1) which can be placed on the (potentially mobile) end-system, on intermediate network nodes (e.g. router, switches) and on transcoding units (gateways). Each Broker delegates and coordinates different Managers which in turn are responsible for specific tasks, like resource, network, media, monitoring, policy and mobility management.

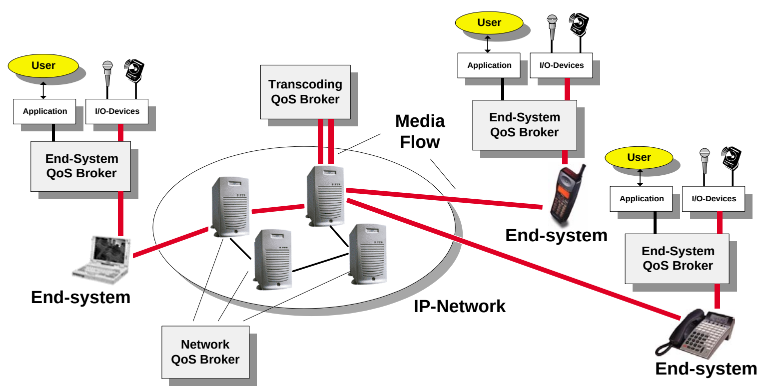


**Fig. 1.** Distributed set of autonomous QoS Brokers.

The main task of the *End-System QoS Broker* is to coordinate, orchestrate and manage local and remote resources for adaptive multimedia streaming, to map the user's QoS wishes to appropriate QoS parameters and to support mobility with the use of different access networks (e.g. GSM, WirelessLAN, UMTS, etc.). *Network QoS Brokers* can be regarded as centralised QoS management units which support the End-System QoS Brokers and organise the orchestration of all streams in the respective network domain according to a given network management policy. The Network QoS Broker monitors network resources, decides admission in cooperation with other admission controlling entities (e.g. H.323 Gatekeepers) and realises load balancing and fairness concepts for all participating systems. The actual behaviour of the Network QoS Broker depends on the location of the respective network node, e.g. core network router, access network router, media gateway, LAN switch etc. The *Transcoding QoS Broker* coordinates its Managers to provide services for media adaptation which may include changing the codec to support heterogeneous clients. In addition, adaptive Forward Error Correction (FEC), appropriate transmission protocols (e.g. wireless-TCP), etc. may be invoked. Transcoding QoS Brokers allow for automatic downloading of codecs and media adaptation objects on demand. A detailed description of the network QoS Broker and the transcoding QoS Broker is outside the scope of this paper.

Communication between the distributed collection of QoS Brokers is realized via appropriate interfaces. Main communication issues are capability exchange methods, QoS routing mechanisms, admission and authorization requests, and the management of media channels.

*MASA* presents applications with mechanisms for the processing and transmission of 'high quality' multimedia streams, i.e. adapted to the user's QoS wishes and the available infrastructure and resource availability. Applications subscribe to the system and use the provided multimedia facilities to control the set-up of a complete chain of media processors, consisting of capture devices, codecs, effect processors, etc. Appropriate graphical user interfaces are provided to present media information (e.g. video panel).

Efficient handoff algorithms are a cost-effective way of enhancing the capacity and QoS of cellular systems. *MASA* supports mobile devices by integrating mobility management into the framework, using fast QoS re-negotiation and adaptation mechanisms to allow seamless intra- and inter-technology handoff.

### 2.2 Hierarchical QoS Broker Structure

Fig. 2 outlines the typical structure of a *MASA* QoS module. The *QoS Broker* is the central intelligence unit which is supported by a set of *QoS Managers* which in turn are supported by appropriate *QoS Controllers*. With this hierarchy of Manager/Controller structures, the QoS Broker can delegate separate tasks for controlling and processing media streams and therefore provides a clear separation of tasks with different time constraints.

The *Policy Manager* is responsible for the storage and retrieval of QoS preferences within a user profile. It presents an appropriate policy GUI to the user. The Policy Controller enables the access to a policy database. The *Resource Managers* are responsible for controlling the available resources (like CPU, memory, network, etc.) via the respective Resource Controllers. The *Media Manager* is responsible for the provision and orchestration of actual media processing entities, like codecs, packetizers, etc., inside the Media Controller. It also monitors the transmission parameters and reports aggregated statistical information to the Broker. The *Intercom Manager* is used to support inter-Broker communication. The *Application Manager* provides mapping functionality between different categories of applications, like VoD or IP-Telephony and the Broker QoS API. The *Mobility Management* is responsible for the support of device mobility and enables the usage of different access devices. We have implemented a Mobility Manager using Mobile IP [6]. The Mobility Controller is a standard

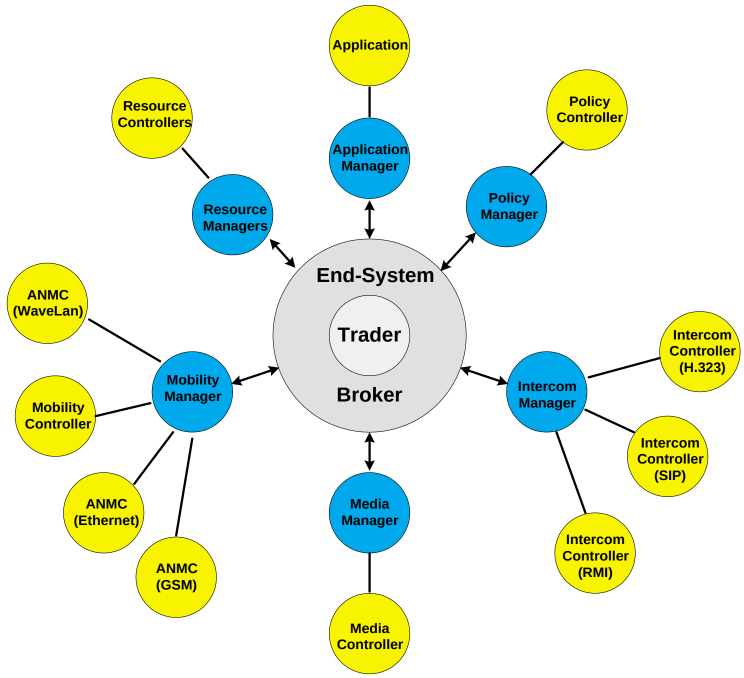


**Fig. 2.** Hierarchical structure of *MASA* on end-systems.

Mobile IP daemon. For each network interface an Access Network Monitoring Controller (ANMC) measures link quality parameters and reports them to the Mobility Manager, which either directly forces a handoff or informs the QoS Broker about available network options. This tight integration of mobility management and handoff control within the QoS architecture allows for supporting seamless handoffs, fast adaptation (e.g. by changing filter settings, see section 4) as well as QoS-controlled handoff decisions.

The Broker regularly requests monitoring information from its Managers. The aggregated monitoring information together with the user's QoS policy is used as input for a trader mechanims which analysis the current situation and decides for possible adaptation of the current active sessions. On the end-system, the algorithm of the trader is controlled by a local trading policy which can be easily exchanged even during runtime. The Broker parses the result and informs the respective Managers about the necessary actions that have to be performed to realize the adaptation. Examples are codec changes inside the Media Manager or handoffs inside the Mobility Manager.

Since not all Managers have to be used for each Broker type, our design provides scalability. For example, at transcoding/filtering nodes, the Broker does not need Application or Mobility Managers. Through the use of open interfaces between the QoS Brokers and Managers, the system can be easily extended with new Manager/Controller pairs. For example, a Manager/Controller pair for power management would allow to support a trading policy like *when the battery power is low, switch to codec X*. The major parts of the architecture have been implemented in Java to allow downloadability of new plug-ins even during runtime. The interfaces between Broker and Managers have been defined by using two request/response queues for each pair, allowing for asynchronous non-blocking processing of requests with different time constraints. A standard interface consisting of methods like *adapt(), get_monitor_results(), etc.* can be extended for each type of Manager.

## 3 Adaptive Media Management

As basis for our adaptive media management we use an adaptive media coder called WaveVideo [4], which is based on the wavelet transform and offers a high degree of error tolerance as well as a high compression ratio at a good image quality. Media adaptation is achieved by inspecting a 32-bit-tag inserted by the coder which describes the content of each network packet. As an example, the tag allows the decoder or any media filter (see section 4) to associate each WaveVideo packet to its frame and for each frame to determine the correct colour/luminance plane. It also determines the associated leave (combination or recursion depth and direction) of the wavelet tree and the quantization layer to which the coefficients contained in the packets contributes to. Video adaptation is achieved by dropping network packets that do not contribute to a desired quality. For more information about the WaveVideo coding method see also [4].

The Adaptive Media Management is implemented by the Media Controller, which is configured by the Media Manager on requests from the Broker. The QoS Broker derives a QoS contract for a given stream based on user QoS parameters maintained by the Policy Manager/Controller. The mapping is given in [10]. It is up to the Media Controller to enforce the QoS contract. Typical parameters are frame rate, frame size, visual quality and end-to-end delay/jitter. The Media Controller at the receiver maintains playout buffers to control the delay, jitter and losses and may request re-transmission of lost packets from the sender Media Controller. If the QoS contract is violated, the receiver Media Controller informs the Broker, which requests media stream adaptation (see section 4) to react to resource availability.

### 3.1 Java Media Framework Plugin

The NEC *AQUARIUS* Media Manager is built in Java. The Media Controller is implemented as an extension of the Java Media Framework (JMF) [12]. JMF is an optional Java package, providing APIs for using, manipulating and managing audio, video and other time-based media. It can be expected that JMF will be supported in the near future by small handheld devices. Therefore, JMF is well suited for mobile multimedia communication.

Through the usage of JMF *AQUARIUS* is able to download, install and maintain a broad range of audio/video codecs to filter and transmit media (via RTP), ranging from low bandwidth speech codecs (e.g. GSM) to high quality video streaming (e.g. H.261/H.263, MPEG-1/2/4). Additional custom coding schemes can be integrated into the architecture using multiplexer, demultiplexer, codec, effect filters or renderer plug-ins.

To support WaveVideo RTP streaming, we have implemented packetizer and depacketizer JMF plug-ins. The packetizer splits WaveVideo encoded frames into network packets based on the tag information and sends them over the network by using the JMF RTP Manager. The depacketizer collects the packets, reconstructs the video frame and gives it to a codec and video renderer for presentation. We have also implemented a JMF filter plug-in, which allows to integrate filter algorithms into the media transmission chain of JMF. The filter plug-in can be parametrized by the supported quality interface, allowing to specify a quality value in $[0, 1]$ (see section 5).

### 3.2 C++ Implementation

Typically, distributed multimedia applications should be able to operate in real time. This was motivation for the University of Ulm to implement a Media Controller in native C++ which realizes high performance media processing and adaptation. Since the whole MASA framework is implemented in Java, interoperability has been achieved by implementing the Media Manager in Java and providing a simple wrapper to the functionality provided by the native Media Controller as well as a data converter. The Media Manager is therefore a stub, which interacts at the Java side with the Broker's requests, which are propagated to the native Media Controller using customized socket communication. The Media Controller itself provides mechanisms to open, close and (re-)configure a local or remote media channel. Monitoring data is provided on request and forwarded to the Broker via the Media Manager. In the case that the requested quality can no longer be maintained, the Media Manager informs the Broker, which starts a new trading cycle based on the monitoring information and users QoS wishes to decide, if new filter settings have to be applied or the codec needs to be changed.

## 4 Filtering Algorithms

In this section we will describe some example filter algorithms for WaveVideo codec video streams. Simple priority-based filtering based on syntactical information can be used to efficiently support a single adaptation policy in a multicast scenario. More complex structure-based filtering based on the semantical information of the WaveVideo tag can be used for a unicast scenario, allowing for different receiver adaptation policies.

All filter algorithms are implemented inside the Media Controller under the control of the MASA QoS framework. Access to the filter services is provided at the end-system using the End-System QoS Broker or at the transcoders using the Transcoding QoS Broker. Filter services can be instantiated on any video stream maintained by the Media Controller. (Re-)configuration of the filter algorithms is achieved by the QoS Broker via the Media Manager based on decisions of the trading policy or user input (e.g. user changes its requirements).

Note, that the filter algorithms do not depend on the MASA architecture. In principle, they work on any WaveVideo compressed stream. However, the algorithms were implemented under the MASA framework.

### 4.1 Syntactical Filtering

The term syntactical filtering is used for algorithms, which base their packet dropping decisions on simple structural information, e.g. priority fields. This scheme can only support one single adaptation policy, since the priority fields can not reflect the relative importance of the packet for the respective users.

One simple filter algorithm is the following. A given input frame consists of $n$ network packets, each carrying one quantization layer of every subband of every color/luminance plane. The number of packets $m$ to be sent is calculated with the quality function $f_q$ defined as follows:

$$f_q : m = trunc(n * q)$$

whereas $q$ is the quality factor in $[0, 1]$, $n$ is the number of packets for this frame at a 100% quality and $m$ the number of packets (starting at packet 0) to be sent with $m \leq n$. Thus, the first $m$ packets of a frame ordered by priority will be passed and the least important packets ($packet_m \ldots packet_{n-1}$) will be discarded. The two extremes of this function are:

- $q = 1$:
  every packet in the input frame which can pass the filter is copied to the output frame.
- $q = 0$:
  none of the packets in the input frame can pass the filter. No output frame is created.

### 4.2 Semantical Filtering

In this section we classify semantical filter algorithms and show how they can be applied to WaveVideo streams. Transcoder filters have been presented and evaluated in [9]. In addition, a dynamic rate shaping filter (DRS-P) that includes user preferences has been presented in [7].

Filters adapting the frame rate operate in the temporal domain and reduce the frame rate. The frame number and the frame type (I- or Delta-frame) are coded within the WaveVideo tag. The frame rate generated by the source is included in an I-frame header [4]. Let us suppose that the sender sends at $r_{src}$ and the receiver wants to be served with $r_{dest}$. The frame rate filter has to decide, if the frame $k$ has to be dropped based on $r_{src}$ and $r_{dest}$ and the dropping history (i.e. what frames have already been discarded).

Filters adapting visual quality of single video frames operate in the frequency domain, degrade visual quality of the luminance and/or chrominance planes and reduce bandwidth without affecting frame rate. A filter adapting colour quality reduces the visual quality of the chrominance planes only. The WaveVideo frame quality filter [9] first extracts frequency information and then determines which coefficients to discard. Filtering is achieved by dropping whole subbands (or certain quantization layers).

The combined filter offers filter services to adapt frame rate, frame size, luminance quality and chrominance quality independently and is controlled by:

- resolution $r_l$ (0 denotes best Quality), i.e. quality of the luminance channel
- quality of the chrominance channels $r_c$ (10 denotes black/white)
- frame size $r_s$.

Variation of $r_l$ and $r_c$ influences the visible quality by implementing subband dropping. Subbands are not dropped if $r_l = r_c = 0$. All subbands $i$ (i.e. all packets contributing to subband $i$) referring to the luminance plane are dropped with $i \leq r_l$. For the colour resolution similar rules apply [9]. The combi-filter allows to reduce frame width and height by a factor of two by setting $r_l = r_c = 2$. The new frame size is calculated to $width_{new} = \frac{width_{old}}{2^{r_s}}$ and $height_{new} = \frac{height_{old}}{2^{r_s}}$. Finally, an I-frame header is updated to notify the decoder about the new dimension of the frames. Frame-rate filter services are provided in the combi-filter by including a new parameter *target frame rate* $r_{dest}$ ($0 < r_{dest} \leq r_{src}$). The combi-filter first performs frame rate filtering as described above. All remaining frames are then filtered according to the settings of $r_l$ and $r_c$.

## 5 Measurement Results

In this section we present initial experimental results for the proposed filter algorithms in a real environment using the MASA QoS framework on a testbed consisting of mobile clients connected via various access network technologies, like e.g. WaveLan and Ethernet.

The performance of the semantical filters have already been evaluated in [9], [7] and [8]. In this work we present additional performance values for the syntactical filter and focus on the interaction between QoS Broker and filter algorithms implemented inside the Media Controller. In particular we are interested in adapting the media stream at the sender side based on various feedback information. To this extend, we focus on the *trading policy* inside the Broker and its interaction with the media management. The decision to reconfigure filters is made by the Trader based on measurement results, which have been obtained from all Managers, QoS profiles and preferences provided by the Policy Manager and resource availability.

We implemented two video application scenarios for unicast as well as multicast transmission using the Application Manager to request adaptive media streaming and filtering services provided by the MASA QoS framework.

### 5.1 Internet Television

The first application is used to demonstrate a scenario, where multicast can be used to stream the same video simultaneously from one sender to many receivers. The quality of the stream and the policy for performing adaptation in the case of congestion as well as to support receivers connected via heterogeneous transmission links is determined at the sender site.

We used the JMF filter plug-in described in section 3.1 to realize a very simple implementation of the syntactical filter described in section 4.1. Using the RTP protocol for transmission, the sequence numbers of the packets start with zero and increase for subsequent packets for each frame. The filter assumes, that the encoder sends a priority-ordered sequence of packets, where the high priority base layer packets are sent first, followed by packets with lower priority. If we further assume, that reordering does not occur or could be corrected by appropriate mechanisms, the filter algorithm just forwards the first part of the packet sequence for each frame, depending on the quality factor.

For the measurements a video with 1460 frames and a resolution of 384 x 288 pixels with 15 frames/s has been used. The complete video was filtered with different quality factors ranging from 0.1 to 0.95. Figure 3 shows the results for the average Peak Signal to Noise Ratio (PSNR) (averaged over $YC_bC_r$) compared to the non-filtered WaveVideo encoded video and the average data rate per frame. Obviously, the data rate tends to decrease much faster with smaller quality factors then the PSNR does. Thus, even relatively high filter ratios only lead to a moderate reduction in the image quality.

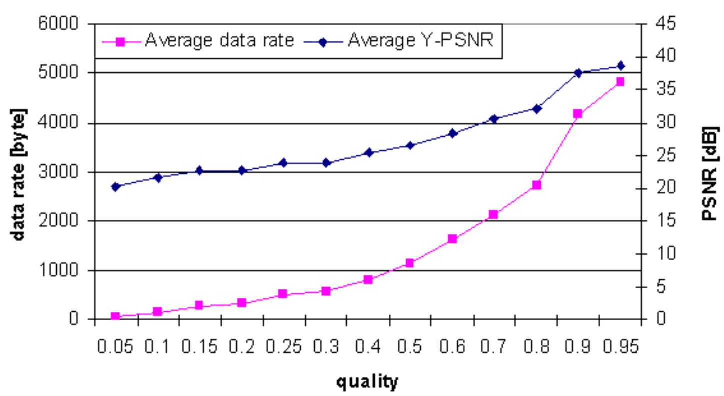


**Fig. 3.** Averaged PSNR values and data rates for different filter quality settings.

These measurements demonstrate, that even with this simple filter implementation a reasonable filter behaviour can be achieved. The JAVA implementation was able to filter the test videos in real time on a standard PC.

### 5.2 Video-on-Demand

In the Video-on-Demand scenario, each receiver is applying for a dedicated video stream defined by specific QoS adaptation policies.

We used a very simple implementation of the trading algorithm which is based on user preferences (to build an adaptation path in case QoS violations are detected) and monitoring data from the Mobility Manager (to know when a hand-over occurred and what access network was used). We used the mapping from a single quality parameter and user preferences to derive a set of intervals at the MASA QoS framework level for several categories (frame rate, frame size, visual quality), which we presented in [10]. Based on information about the access network (available from the Mobility Manager) the number of streams to send, the relative importance of the streams and receiver capabilities, we derive a target bandwidth for the sender. This information is available at the trading algorithm, which calculates a set of filter parameters based on local monitoring information (like instantaneous bandwidth), user requirements mapped to intervals and adaptation paths according to [10] and resource availability.

As video source file we used a digitized and WaveVideo compressed scene from the Beverly Hills Cop movie at 352x288 pixel resolution. The original frame rate was 25 fps. In our scenario, the receiver was not interested in receiving more than 10 fps and could not manage more than 1 MBit/s. As frame rate intervals, the receiver specified ]8,...,10] fps as desirable, ]5,...,8] fps as tolerable, and [2,...,5] fps as minimum acceptable. The Media Controller uses a combi-filter that allows to adapt frame rate (to scale down from 25 to at most 10 fps) and frame quality to match available bandwidth and the preferences.

We conducted the following scenario. First, the sender was connected to the fixed network. After sending 500 frames, the fixed network connection was terminated, which resulted in a hand-over to a WaveLAN base station. After another 500 frames, a hand-over to a new base station was performed that shows better signal quality. Frame rate was judged more important than visual quality. As a consequence, the trader tried to maintain the 10 fps while maintaining the 1 MBit/s during the first 500 frames (phase one), see figure 4. The Media Controller derived the following settings: $r_{src} = 25$, $r_{dest} = 10$ to drop the HH, HL and LH subband of the highest layer ($r_l = 1$). Note that the original stream was based on I- and Delta frames so the instantaneous frame rate varied between 5 and 12 fps. The average frame rate of 10 fps was maintained during the first period. The data rate varied between 400 kBit/s and 1 MBit/s. Without invoking the combi filter, the data rate would have been more than 1 MBit/s.

At frame 500, the Mobility Manager detected a handoff to WaveLAN with a low signal quality. As a result, the QoS Broker instructed the Media Controller to reduce the data rate to at most 200 kBit/s and comply to the user preferences. However, this was only possible if the frame rate was also reduced. The Media Controller therefore applied the following settings: $r_{src} = 25$, $r_{dest} = 4$, $r_l = 5$ thus reducing the average frame rate to 4 fps (minimum acceptable interval) and dropping more subbands of all those frames that are to be forwarded. The data rate was thus reduced to values between 180 kBit/s and 90 kBit/s in phase two.

The receiver noticed that the visual quality was bad as a result of its preferences, so he signaled to the sender that visual quality is as important as frame rate and that a new setting should be applied if resource situation allows this. After frame 1000, a hand-over to a better access point occurred. The QoS Broker instructed the Media Manager/Controller to use the following settings: $r_{src} = 25$, $r_{dest} = 6$ thus increasing frame rate to 6 fps (tolerable interval). As upper data rate limit the trader calculated 500 kBit/s, which the Media Controller used to invoke the combi filter with $r_l = 3$. More subbands were added to increase the visual quality and the data rate varied between 200 kBit/s and 500 kBit/s in phase three.

We can draw the following conclusions from this example. If the original stream is encoded at a variable bitrate, applying a frame rate, luminance quality, chrominance quality filter will not change the variable nature of the stream. Thus the post filter stream also shows a variable data rate (see figure 4). If a constant data rate is desired the dynamic rate shaping filter with priorities should be used [7]. An integration of media management and mobility management enables wireless devices to re-act and adapt to QoS fluctuations on the wireless link by observing user preferences. Finally, adaptation should be co-ordinated among the different entities: resource management, media management, profile management and the coordination unit itself. The adaptation has to be coordinated with the peer because a change in the media stream will influence remote resource consumption. Integration of local, network and remote resources is necessary to provide a full end-to-end QoS aware service.

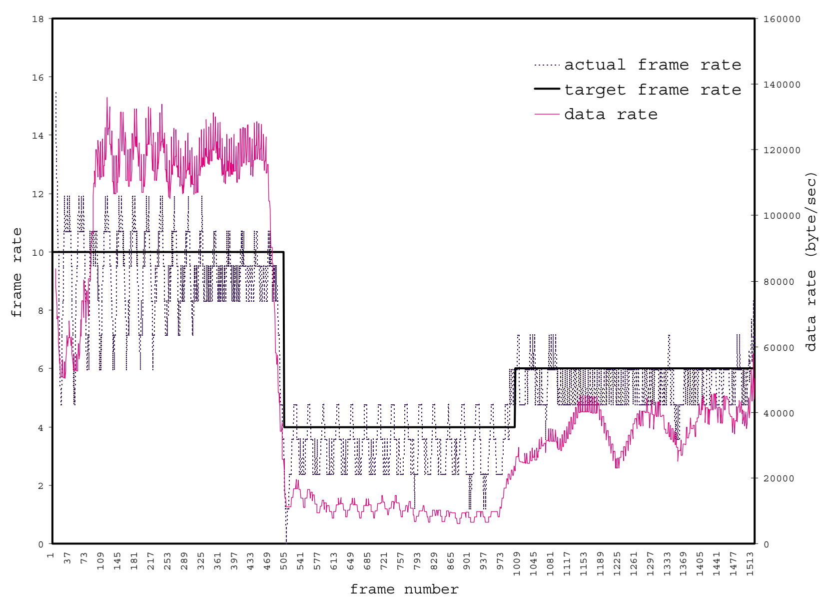


**Fig. 4.** Frame rate and data rate for the Beverly Hills Cop movie based on combi filter.

## 6 Conclusions

In this paper we have proposed a flexible filtering strategy supporting adaptive transmission of video streams in a heterogeneous mobile environment. By a combination of syntactical and semantical filter algorithms at filter nodes inside the transmission path, adaptation to varying network situations concerning QoS policies could be achieved. The filter network was implemented within the MASA QoS framework, which allows for a comprehensive management of multimedia streams. The functionality and efficiency of the filtering approach have been demonstrated by first measurements in two example scenarios supporting unicast and multicast applications.

We are currently investigating mechanisms to support multiple different QoS policies in a multicast scenario at the same time. Therefore, we are working on the definition of algorithms to map aggregated subtree requirements to packet drop decisions. Also, enhanced feedback mechanisms are under development, which allow for an optimized parametrization of the codec itself.

## References


1. C. Schremmer and C. Kuhmünch and W. Effelsberg. Layered Wavelet Coding for Video. In *Workshop on Packet Video*, 2001.
2. E. Amir and S. McCanne and M. Verterli. A layered DCT coder for Internet Video. In *Proceedings of the IEEE Conference on Image Processing ICIP'96*, September 2001.
3. F. Raspall and C. Kuhmünch and A. Banchs and F. Pelizza and S. Sallent. Study of packet dropping policies on layered video. In *Workshop on Packet Video*, 2001.
4. G. Fankhauser, M. Dasen, N. Weiler, B. Plattner, and B. Stiller. WaveVideo - An Integrated Approach to Adaptive Wireless Video. *ACM Monet, Special Issue on Adaptive Mobile Networking and Computing*, 4(4), 1999.
5. H. Hartenstein, A. Kassler, M. Krautgärtner, A. Schrader. High Quality Mobile Communication. In *Proceedings of the KIVS'2001 Conference*, 2001.
6. J.D. Solomon. *Mobile IP.* Prentice Hall, 1998.
7. A. Kassler and A. Neubeck. Self Learning Video Filters for Wavelet Coded Videostreams. *Proceedings of the Int. Conf. of Image Processing (ICIP2000)*, September 2000.
8. A. Kassler, A. Neubeck, and P. Schulthess. Filtering Wavelet based Video Streams for Wireless Interworking. *Proc. of the ICME*, July 2000.
9. A. Kassler, A. Neubeck, and P. Schulthess. Classification and Evaluation of Filters for Wavelet Coded Videostreams. *Signal Processing: Image Communication*, 16(8):795–807, May 2001.
10. A. Kassler and P. Schulthess. An End-to-End Quality of Service Management Architecture for Wireless ATM networks. *Proc. of the HICSS 32*, January 1999.
11. Steven MacCanne and Van Jacobson. Receiver-driven layered multicast. In *Proceedings of ACM SIGCOMM'96*, August 2001.
12. Sun. *The Java Media Framework Version 2.0 API.* http://java.sun.com/products-/java-media/jmf.