An End-to-End Quality of Service Management Architecture for Wireless ATM Networks

Andreas Kassler, Peter Schulthess

Department of Distributed Systems, University of Ulm, Oberer Eselsberg, GER-89069 Ulm, Germany

Tel: *++49 731 502 4238*, **Fax**: *++49 731 502 4142*, **eMail**: *[kassler,schulthess]@informatik.uni-ulm.de*

Abstract

Wireless ATM networks will bridge the gap between mobility and broadband networking. The major challenge to this end consists of guaranteeing adequate end-to-end Quality of Service (QoS) in spite of the unreliable radio channel. In response to this we propose an integrated end-to-end QoS management architecture built around scalable error tolerant codecs, network filters and error modules to compensate for transmission errors and local QoS degradation. A broker concept combines local and distributed resource management and negotiation of QoS at different levels. By introducing QoS intervals both at application and transport level we cope with short term channel fluctuations on a wireless link. User definable degradation paths determine the behavior of the system in case the network QoS changes for a longer period of time. Mapping functions are provided to transform a QoS description from the user perspective to QoS parameters and negotiation at the network level. Incorporating media filters into the overall broker architecture allows different media codecs to be used between the source and the sinks depending on the channel conditions and link utilization while obeying user defined end-to-end QoS strategies. We draft several protocols for the sender and receiver driven multicast scenario and show, how automatic filter propagation can be achieved under given end-to-end QoS constraints.

1. Introduction

Distributed multimedia applications like video conferencing tend to absorb all available communication and computing resources. Improvements in hardware technology are therefore more than welcome. On the network side broadband communication channels will carry multiple media streams of appropriate QoS to individual workstations and set-top boxes. Immediate information access anytime, anywhere will be a key issue in future business and home environments.

Current wireless access networks support low speed and high latency services such as e-mail and give the user the freedom to communicate while he is moving from one location to another. However, to provide a sound basis for broadband applications on mobile broadband networks remains a major challenge due to the inherently unreliable radio connection with varying bit error rates and unpredictable delay characteristics. The concatenation of wireless and fixed networks must provide a pre-negotiated QoS to end users or applications. In the fixed network part Asynchronous Transfer Mode (ATM) networks can guarantee some level of QoS. In the future, wireless ATM networks will offer a unique combination of broadband networking and mobility and extend the QoS concept onto the wireless link.

A second major challenge is to provide adequate QoS abstractions to applications and end-users. QoS aware networks are highly sophisticated but QoS description at network level is cryptic and it is necessary to hide this complexity from the user. Mapping and negotiation functions are needed between different levels of QoS. This layered approach will lead to a consistent system configuration and to a valid set of network level QoS parameters for setting up connections. At the top level different users may have different preferences and understanding of QoS values. Therefore negotiation, mapping and resource allocation must occur between the different partners of a video conference. To cope with large and frequent fluctuations of channel conditions for mobile terminals (MT), a flexible end-to-end QoS management is needed based on soft guarantees. Hard guarantees will more often drop connections because of a single service contract violation at network level, when a mobile user moves into regions with bad radio conditions.

Although wireless ATM networks provide high bandwidth on demand video compression is still required for reducing the amount of data transmitted. In principle, ATM networks can guarantee certain QoS parameters by reservation of switching resources but cell loss ratio at peak packet rate is rarely guaranteed. It is left to higher layer protocols to retransmit lost data. Due to upper bounds for delay retransmission is impractical for real-time applications and packets are simply lost. Unfortunately, most video compression algorithms employ variable length coding and experience severe problems when the compressed data is corrupted. Error concealment techniques *[11]* may hide the introduced artifacts or error tolerant video compression methods like layered coding can be used *[12]*, which only depend on the correctness of the base layer. So called enhancement layers add detail to the overall content. With network filters *[22]* it is possible to tailor media streams to the capabilities of the channel for serving heterogeneous receiver groups. Applying a filtering strategy to video communication different partners may be served simultaneously with a different visual quality without the need for the sender to compress the video frames to more than one representation. State-of-the-art approaches for providing QoS either ignore media filters and rely on a pure client-server approach *[15]* or use filters but do not consider them in an end-to-end QoS framework *[5]*. We suggest to incorporate network filters into the resource allocation strategy and into the QoS ne-

0-7695-0001-3/99 $10.00 (c) 1999 IEEE

gotiation phase because they may consume significant processing power and contribute to the overall delay. Dynamic reallocation of filter resources and dynamic adaptation of filtering strategies may be required when MTs move from one access point (AP) to another.

We have now introduced the problem of providing negotiated end-to-end QoS on wired and wireless broadband networks. Section 2 will discuss related work in wireless broadband networking, QoS provisioning mechanisms, end-to-end QoS guarantees and QoS mapping. Section 3 analyses challenges for wireless multimedia networking. Our overall architecture, the different QoS levels supported, an extended QoS broker model, different QoS negotiation and mapping protocols, QoS adaptation mechanisms for mobile terminals and the role of filters and filter-dealers are presented in section 4. Finally, section 5 draws conclusions and discusses further work.

2. Related Work

Research in the area of wireless broadband networks has led to a number of evolving demonstrator systems like the WAND system [14], MEDIAN [2] or SWAN [2]. QoS aspects are discussed only at the wireless [17] and wired network layer and below without taking advantage of possible interactions between applications and network. Some aspects of co-ordination between lower layers and video coding methods are discussed in [7] and [9].

QoS issues in wired networks have been extensively studied. Sender based as well as receiver based approaches for delivering QoS guarantees for multimedia systems are available and both rely on an end-to-end approach. QoS can only be assured when resource reservation mechanisms are incorporated into the architecture. Krishnamurthy and Little [10] try to guarantee unconditional resource availability for fixed networks. But it appears that when the network is not able to provide hard guarantees only a combined view on application, middleware and network can provide the best QoS trade-offs. Mobile and wireless networks may suffer from temporary channel fading and high bit error rate which makes it very hard to predict the behavior of the wireless transmission channel over a prolonged period. Therefore, resource reservation in the wireless part can never guarantee unconditional and hard availability of negotiated QoS. A close interaction between application and network is proposed by several researchers, where video coding algorithms react upon feedback from the network to adapt their behavior. Yeadon et al. [22] takes care of heterogeneous receivers by separating to a certain extent applications and the network and by introducing network filters for hierarchically encoded video streams.

Several protocols have been designed to support resource reservation between end systems. Typically, transport layer parameters are reserved ignoring media or application specific parameters. Common to most approaches is the assumption, that no processing of the transport data is allowed within the network. Beyond traditional multicast applications RSVP [23] supports many sources and many sinks. Pasquale et al. [18] proposed a dissemination-oriented approach for multimedia multicast channels based on single source application. Both, the RSVP and Pasquales approach consider stream filters to be specified by the end-stations at network level. However, no QoS management at the filters is incorporated.

Several approaches have been developed for mapping between different levels of QoS parameters. In [10] subsequent negotiation and resource reservation within the network is performed, whereas Besse et al. [1] make use of a management information base for mapping application specific QoS to transport level QoS. On the other hand, Schill [20] developed a QoS manager tool which returns a transport level QoS structure based on a given kind of media encoding method and a desired QoS class. Vogt [21] presented a QoS management scheme that maps variable bit-rate streams to a simple and periodic traffic model leading to an estimation of delay bounds. However, all approaches for QoS mapping only cover client/server architectures and fail for symmetric multicast scenarios.

A novel approach based on intelligent agents was developed by Nahrstedt and Smith [15] where a QoS Broker negotiates end-to-end QoS at application level between client and server. Resource reservation at the end-systems and intermediate communication links is combined with a selection of media specific QoS between application components. A few ideas for mapping between different QoS levels are described. Filter operations are not allowed for end-to-end QoS management, which limits the usability for multicast scenarios and heterogeneous receivers. On the other hand, the WaveVideo approach [5] provides a set of mapping and adaptation mechanisms for integrated wired and wireless networks based on a scalable video compression algorithm. However, no resource admission tests are made and the architecture only covers receiver based media scaling. While this architecture is very promising, the protocols for setting up the filters are not flexible and do not cover sender based architectures. No end-to-end QoS negotiation is used, so no end-to-end guarantees can be provided. Furthermore, this architecture is tailored to the WaveVideo coding method and QoS management at the network filters is ignored.

3. Wireless Multimedia Networking

Distributed multimedia applications impose stringent requirements both on resources within end-stations and within the network. Typical wireless networks provide only best effort transmission capabilities, which is not enough for QoS aware multimedia applications because real-time media streams have to be captured, coded, transmitted over the network, decoded and presented to a user within given time and quality constraints. A wireless access sub-network should directly support these real-time multimedia requirements and still be compatible with existing transport infrastructures. Wireless ATM networks try to guarantee QoS at the network level on a per connection basis In the Magic WAND project [14] a so called wireless data link control layer (WDLC) underneath the ATM layer hides the radio link behavior to some extent to the ATM layer to provide a low cell loss ratio. Nevertheless, the trade-off between delay and loss persists and the WDLC offers several options (like forward error correction or ARQ-schemes) to adapt to the requested traffic class and connection priority. Nevertheless it is possible to specify an upper bound on the end-to-end delay of a connection at the expense of possible high cell loss ratio *([13])*.

Distributed multimedia applications are likely to run on heterogeneous end-systems and networks with widely varying resource allocation and control capabilities. Wireless access will lead to mobile networks with fluctuating QoS characteristics due to varying channel conditions and MTs moving around. However, multimedia applications typically negotiate resource allocation with the network and should adapt to changing network conditions using suitable adaptive media compression methods and multimedia filters for tailoring the compressed streams. Our goal is to provide a framework for end-to-end QoS aware transmission of media streams for heterogeneous networks (wireless as well as wired access) and end stations. Our motivation for extending the QoS broker concept to network filters is based on the observation, that end-to-end delay bounds are likely to exist for real-time video conferencing scenarios and if network filters are used for intermediate media processing these possibly expensive operations should also be included into the end-to-end QoS negotiation.

4. QoS Management Architecture

Multimedia applications offer many directions of freedom to adapt to different QoS levels. Delay-bounded transmission is required for precise timing and synchronization between different media streams, but when resource conditions vary, video applications might lower the transmitted frame size, rate or color resolution to save network resources. This adaptation should be dynamic to allow users to specify their cost constraints or preferences in case the original high level QoS settings can no longer be maintained due to deteriorated channel conditions. QoS management is necessary in the end-systems and in the network for maintaining user defined QoS levels over an entire session. Incorporating this management into applications is only an intermediate solution and a more generic framework is needed to deal with filters and active networks of the future. Our architecture incorporates wireless as well as wired networks and comprises a number of functional entities for end-to-end QoS management. A high level description is given in the next section.

4.1 Overall Architecture

Our architecture is based on QoS brokers, network filters, error control modules, source coders and decoders and QoS negotiation protocols. We envisage a combination of wired and wireless sub-networks with resource reservation capabilities to provide soft low-level network QoS guarantees. Interworking with local area networks without QoS guarantees is based on best effort principle utilizing network filters within routers or end-stations.

The WaveVideo *[4]*, *[5]* codec is well suited for flexible and adaptive video compression. Intelligent segmentation of the coded video information into network frames is used and any intermediate node or network filter may drop a set of the tagged frames in order to select the desired quality. This is especially important for a multipoint scenario, where WaveVideo allows to compress the video data to the highest quality resulting in a high bitrate stream. Each partner or intermediate filter can then extract the required quality. Our QoS framework is built around network filters and the concept of brokers and also allows the use of other coding schemes.

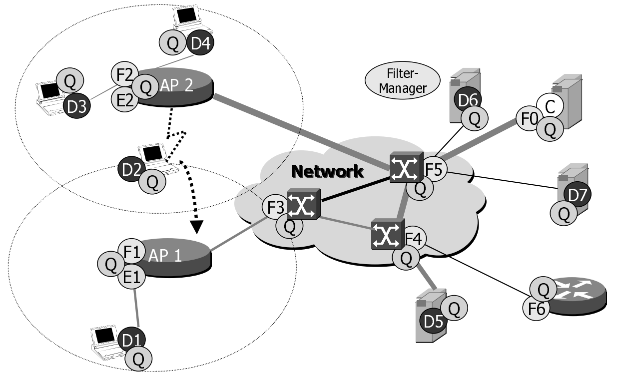

Figure 1: High level system view.

End-systems (fixed and mobile terminals) include a set of media coding and decoding methods and different QoS controlling entities. APs may house filters and error control modules to receive feedback on channel conditions. Error control modules may add robustness to the vital video information[1], because each AP knows about the capacity and channel quality. Filter operation on video streams may range from very fast pure transport level processing (if, e.g. whole video-frames are dropped, a filter for WaveVideo would read the network frame header, analyze the content and decide to drop or forward the frame) to slow and complex application level processing (e.g. transcoding MPEG to WaveVideo for error tolerance). Typically, filters scale the video streams to different requirements (e.g. bandwidth) and serve several MTs when used inside APs with different quality constraints and channel conditions. Fast and simple filters inside switching nodes serve different branches with different quality levels. All filters register with a filter-manager authority to be contacted by a filter-dealer, whenever media scaling is necessary.

Different mobile receivers *(Figure 1)* may have different quality requirements. Based on high-level QoS parameters, a mapping to a set of resource requirements is necessary for reserving resources in the local system (for media processing), intermediate (network filters) or remote systems and within the network for QoS aware end-to-end communication. Consider the video coder C, which generates a high quality video stream. Other terminals attached to the same switch may receive the same (multicast capabilities can be exploited, if the network supports this) or lower quality. For all lower quality streams which are directly attached to the switch filters within end-stations or at the sender are used to select the desired quality. For terminals not directly attached to the switch, filters *(F5*,

[1] For WaveVideo, each network packet contains a high-level description of its content using dedicated tags. An error control module has to look for packets containing vital information for the base layer and add error protection using e.g. forward error correction or distributing the base layer information among several packets. The broker architecture should be extended to the error control modules, too, because the processing needs additional resources and adds delay. However, this is beyond the scope of this paper.

F4, *F3)* at the outgoing links can be used to scale the media stream to be suitable for the quality levels of all subsequent receivers. Filters at APs *(F1*, *F2)* serve all MTs currently registered at the corresponding AP. At the receiving MTs, filters can be used to scale down the video stream to adapt to available processing power. Filters placed at routers *(F6)* may serve as proxies for the connected LAN. At each host we place several QoS management entities using a broker concept to perform QoS negotiation, adaptation and mapping between the communication end-points and intermediate filters. QoS mappers map between different QoS levels, e.g. between user and media specific QoS. Resource managers request, control and reserve local resources to guarantee the requested QoS. Broker-operations performed at filters negotiate filter resources and set-up and control the behavior of filters.

The dynamic behavior of mobiles is our motivation for combining an extended broker concept with network filters. Consider the case, where mobile D*2* moves from AP*2* to AP*1*. Assume D*2* is served with *24* fps C*IF* size true-color high quality video at AP*2* resulting in *4* Mbit/s transmission capacity and overall end-to-end delay of *15*0 ms including intermediate processing at *F*0, *F5* and *F2*. Assume further, that *F1* at AP*1* currently serves mobiles with low-quality video, say *3* fps QC*IF* size, bad colour quality, and that *F4* and the network link between *F4* and *F3* is near the upper capacity limit. Without QoS management at intermediate filters, D*2* will be handed over from AP*2* to AP*1* and filter propagation is triggered so that *F1* will eventually be served with high-quality video. However, in our scenario this is not possible due to the link and filter limitation between *F4* and *F3*. Introducing a QoS management architecture at filters and distributed filter-dealers, D*2* will notice the bad video quality after hand-off and contact *F1* to get higher quality. *F1* will contact *F3* and eventually *F4* to request the desired quality. Since *F4* and the link between *F3* and *F4* is the bottleneck, *F3* will notice that it is not possible to get better quality from *F4*. Therefore, it will contact via filter-dealer *F5* and request higher quality. *F5* already can provide the desired high quality and *F3* will from now on request the video stream from *F5* instead from *F4*. Note, that several other cases can be constructed when end-to-end delay and intermediate filter processing time is taken into account (if, e.g. *F3* can not support overall *15*0 ms delay bound due to possible overload, given D2s high quality requirements). Extending a QoS-broker concept to network filters and filter-dealers will offer more flexibility, dynamics and soft end-to-end guarantees compared to *[5]*, taking into account the dynamic behavior in a mobile environment.

4.2 Levels of QoS

Distributed multimedia applications operate at several levels in order to provide end-to-end guarantees. At least three layers below the human user can be identified according to *[16]:* application level, system level and the network and multimedia device level. For setting up the communication channel a mapping and negotiation procedure is required for the QoS parameters the user has specified. Within our framework we define the following levels:

- User level QoS (UserQoS) is defined by the user and expresses the overall perceived quality using a single value. The higher the value, the better the perceived QoS. The user defines preferences used for the mapping functions between UserQoS and AppQoS. These preferences are also used to calculate a degradation path. The system lowers the least important parameters first, when a re-negotiation is triggered due to either internal (if e.g. the NQoS changes after a hand-off is performed) or external (user driven) events.
- Application and media level QoS (AppQoS) is described in terms of media quality and media relation. These parameters are still expressed in high level terms and presented to the user. For example, the user may specify the visual quality to be good or to use a large frame size. Note, that we define intervals for AppQoS and the system tries to stay within the calculated interval as long as possible.
- System and device level (SysQoS) parameters describe operating system utilization like minimum and maximum CPU usage, memory utilization or media codec availability and capabilities. Parameters at this level are invisible to the user.
- Transport level QoS (TQoS) is described in terms of network load and network performance. The transport service provider does the final mapping to network level QoS (NQoS) parameters. Note, that we also use intervals at this level which helps mobiles to recover from temporal channel fading and changing bandwidth.

The goal for the QoS management system is to set up and maintain end-to-end communication given the user and application level QoS parameters under the constraints imposed by the local and remote system capabilities and resource availability within the network and intermediate filters.

4.3 Broker models

The concept of resource brokerage was first introduced by Nahrstedt *[15]* and resembles the human nature of brokers, which are specialized in particular types of negotiation. Brokers manage resources at application and transport level and negotiate with the network and remote brokers. In *[15]*, buyers are the end-points in a communication process which want to buy resources from other end-points within the distributed multimedia system and the sellers offer their resources for sale. The brokerage process is initiated by the buyer. We extend this model to incorporate intermediate processing at application and transport level by using network filters.

Each broker entity contains a broker-buyer and a broker-seller. The tasks for the buyer are to gather information about available local resources such as CPU utilization, memory allocation or hardware compression capabilities, to gather information about available network resources and to gather resource information from the remote seller. The tasks for the seller at the remote side are to orchestrate the remote resources and to respond to requests initiated by a buyer. In order to cope with two different scenarios (sender and receiver brokerage initiation), two different protocol entities are identified: sender and receiver. The resources necessary for outgoing (incoming) calls and connections are managed by the sender (re-

0-7695-0001-3/99 $10.00 (c) 1999 IEEE

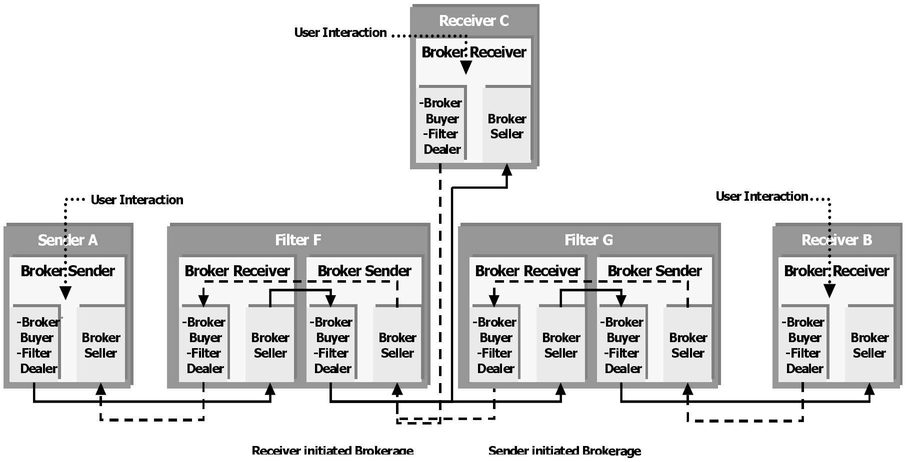

Figure 2: Broker model incorporating network filters

ceiver). To explain the different modes, consider a videoconference, where A wants to transmit his audio and video streams to a remote partner *B*. Here, A is the sender and requests as a buyer resources from the client *B*, which is a seller of his resources. After successful brokerage, the call can be established and the streams are transmitted under certain end-to-end QoS guarantees. Note, that in our case a network filter *F* can play the role of *B* and, if the original intention was to set-up a connection between two endnodes (A and *B*), the same network filter has to react as buyer for the resources of other filters like G in Figure 2 down the sink path or as a buyer for the resources of *B*. For the second mode called receiver initiated brokerage, a receiver acts as buyer for resources from a sender to establish unidirectional connection with QoS guarantees. As an example, consider Video on Demand, where the client (receiver *B*) requests media clips from a server with or without intermediate filters. Another example would be the video conferencing scenario, where the receiver would initiate the call setup. When the brokerage fails due to lack of resources at application or transport level at the receiver side or inside the network, filters are used to lower the media stream quality. In this case we incorporate a so called filter-dealer, which decides for each receiver the filters to be contacted for QoS brokerage. This process is described below. A different example would be, if a multicast tree is already set up between sender A, filter *F*, G and receiver *B*. If receiver C wants to join the multicast tree, its filter-dealer finds a suitable (set of) filters (in our case *F*) and receiver-buyer C contacts sender-seller at filter *F* to contact the multicast group and the protocol starts negotiation at AppQoS level between broker-buyer at C and broker-seller at *F*.

Note, that due to several levels of QoS within the system and due to the fact that network resources like switch output buffers are typically shared and expensive, resource reservation and allocation have to be de-coupled between application and transport subsystem. Consequently, buyer and seller are separated into an application and a transport part. Negotiation and resource management between different partners (e.g. sender and receiver) is carried out at application level and at transport level. Peer-to-peer communication is established between the application and transport sub-parts of buyer and seller. Layer-to-layer communication is used within buyers or sellers between application and transport sub-parts for e.g. QoS parameter mapping.

In summary, the tasks for the broker-sender are to orchestrate local resources for outgoing connections and media input in terms of user, application, system and transport level QoS whereas the broker-receiver is responsible for incoming connections and presentation of media. The buyer and seller protocols reside within the sender, receiver or filter and are activated upon response to sender or receiver initiation. The filter dealer reacts whenever a receiver or the network is not able to provide the necessary resources. Several different state machines are identified (sender-buyer, sender-seller, receiver-buyer, receiver-seller, filter-buyer, filter-seller, filter-dealer) and activated depending on the location (sender, receiver, filter) and the side (buyer or seller). Note, that the sender within a videoconferencing scenario should always generate (captures and compresses) the highest possible media quality. It is up to the receivers to cope with the media quality and, if the receivers do not have enough resources, intermediate filters have to scale the media quality in accordance with the desired end-to-end QoS.

4.4 QoS Management

Typically, an user is not concerned about low level technical parameters like cell loss ratio. Instead, he would like to specify his preferences in order to establish a communication to remote partners. Several proposals exist for suitable user interfaces like Quality Query by Example (QQBE, [6]) and the optimal design is still an open issue. An optimal user interface should balance the complexity of user input with functionality and we presented in [8] an intuitive user interface with the option to specify more precise values. A simple quality slider indicates the overall perceived user level QoS Q_{glob} ranging from low quality to high quality (0...100%). A prioritized list is used for mapping Q_{glob} to AppQos.

We have hidden a number of low level parameters from the user like the used video or audio compression methods, because this can be under control by the QoS management. Only the parameters influencing users perception are presented. For the case of video, the combination of frame rate, size, color and visual quality influences this perception. Given a certain video codec, many different combinations can be supported by a single resource level expressed in transport and system level QoS. The priority settings help to decide, which combination the user prefers. Note, that AppQoS parameters define intervals and the user may define the boundaries (e.g.: frame rate (TV-rate) = [24 ... 30 fps]). Given Q_{glob} and the priority list, we can derive the intervals at AppQoS level. Based on these intervals, we can then calculate intervals at TQoS level for a given media codec. The idea for the broker mechanism is to negotiate end-to-end AppQoS and suitable TQoS intervals. Note, that TQoS is valid only between two peer-to-peer entities (sender-intermediate filter, filter-filter, filter-receiver or sender-receiver if no filters are used) and may change from hop-to-hop due to intermediate filter operations.

4.4.1 QoS Mapping

In our framework several QoS mapping functions are performed by QoS mappers to calculate different QoS parameters at different levels. First, a mapping between the overall desired quality and the AppQoS parameters involves the users preferences and the overall quality parameter Q_{glob}. Each AppQoS category $x \in$ {visual quality, colors, frame size, frame rate, delay, synchronization, cost} is assigned a unique priority $p_x \in \{1,..,7\}$. The higher p_x, the higher the priority. Based on Q_{glob} and

$$\{(x, p_x) | x = \{1,..,7\}\}$$

we choose weights w_x for each p_x with

$$\sum_{i=1}^{7} w_i = 1, \quad w_i > w_j \text{ for } i > j .$$

The higher w_x the higher the priority. Each category x is divided into a variable number $r_x(x)$ of intervals with user defined boundaries.

We calculate the interval number

$$I\left(Q_{glob}, w_x, x\right) = \left\lceil \frac{Q_{glob} \cdot w_x}{Q_{max} \cdot w_{max}} r_x \right\rceil ,$$

where Q_{max} denotes maximum quality (i.e. 100) and $w_{max} = \max\{w_x,\ x = 1,...7\}$. Based on experimentally found weights $\{w_1=0.2,\ w_2=0.18,\ w_3=0.16,\ w_4=0.14,\ w_5=0.12,\ w_6=0.11,\ w_7=0.09\}$ and the defined intervals for frame rate (Table 1) we tabulate the interval number for several exemplary Q_{glob}-values in Table 2. This mapping defines the intervals at AppQoS level together with user definable bounds. A fine granularity is desired for AppQoS intervals and allows for fine-tuning.

Framerate	Interval	Int. nr.
Single frames	[0,...,0.5 fps)	1
Video phone (modem line)	[0.5,...,4 fps)	2
Video phone (ISDN)	[4,...,8 fps)	3
Video phone mode ethernet	[8,...,24 fps)	4
TV rate or higher	>24 fps	5

Table 1: Mapping Q_{glob} to AppQoS intervals

Mapping between AppQoS and TQoS intervals depends on the media codec and the transport system characteristics. For WaveVideo several filter settings are evaluated in [5]. A table based approach is used to perform the mapping to peak bandwidth. Different preferences settings are used to generate the mapping tables but the mapping clearly depends on the activity in the chosen video and no hard values can be derived. The AppQoS to TQoS-mapper either needs a database, where suitable tables are stored for each codec or a QoS-mapping interface provided by a codec. For the second approach the video codec would calculate a set of TQoS intervals based on given AppQoS intervals. For establishing network connections, these TQoS intervals have to be mapped to network QoS based on a given transport service provider. Connection is established using the highest working point within the calculated intervals. If the resources are not sufficient within the network, connection is re-established given the lowest working point within the calculated intervals. If this also fails, re-negotiation is necessary (internal event). Otherwise the quality will be successively increased by re-establishing the connection at increasing working points.

Q_{glob} / w_x; p_x	1	25	50	75	100
0.20;	1	2	3	4	5
0.18;	1	2	3	4	5
0.16;	1	1	2	3	4
0.14;	1	1	2	3	4
0.12;	1	1	2	3	3
0.11;	1	1	2	3	3
0.09;	1	1	2	2	3

Table 2: Derived interval number for frame rate

4.4.2 QoS Controlling Procedure

In principle, the described QoS controlling procedure may be initiated by the sender or the receiver. Sender initiation requires the sender to be the buyer for resources, whereas receiver initiation perceives the sender as the seller of resources. Signalling is used to exchange information among the different entities using the following messages:

- accept: as a response to the possibility of a seller to allocate the resources
- reject: indicating the incapability's of a filter-seller to provide the requested resources, if time-out is signalled, or if no suitable filter is found (in this case filter-dealer sends reject)
- modify: as a response to the situation, when a seller must relax resource requirements, but the sellers upper bound is still within the buyers interval.
- filterrequest: as a response to the situation, where the sellers AppQoS is lower than buyers AppQoS and the seller does not have or does not want to increase resources to support buyers AppQoS.

As a response to the filterrequest message, the filter-dealer is contacted to look for a suitable filter for scaling the streams to the receivers capabilities. The following example explains the signals for the sender initiated approach: The buyer wants to buy resources for sending [20,..24 fps] video. If the sellers resources allow only [8,..12 fps], filterrequest is signalled. If [18,..22 fps] are allowed, modify([20,..22]) is signalled. Accept is signalled for e.g. [20,..24 fps]. A filter would send reject, if it can not support the requested filter operations given his resources and strategies.

For the sender initiated approach we start with the user, who wants to send a media stream to a remote partner given a certain UserQos. UserQoS is mapped to AppQoS as described in 4.4.1. We obtain a set of intervals for AppQoS parameters and the maximum value of each interval is used for mapping to local SysQoS parameters. The resource manager is contacted using the upper bounds for admission tests like a local schedulability test (for example to test, if the system is capable to capture and compress *24* frames per second*)* and an end-to-end delay test to check against upper delay bounds. After successful reservation of local resources, local broker-buyer starts negotiation with remote broker-seller at AppQoS level including the offered and desired compression methods and the AppQoS interval. This information exchange may be performed several times and in the end the sender and receiver know about available compression methods. Of course, other application specific data like user names can be exchanged, too. Note, that at this point we do not start to negotiate at transport level, because we do not know a priory, if an intermediate filter is needed.

The knowledge about offered compression method and media quality helps the broker-seller to decide, if a filter for media scaling or transcoding is needed. Transcoding introduces more delay than decompression but saves transmission bandwidth. If decompression is chosen, the necessary bandwidth between filter and receiver may be too high and the filter may drop every second frame, if the preferences of the receiver suggests this.

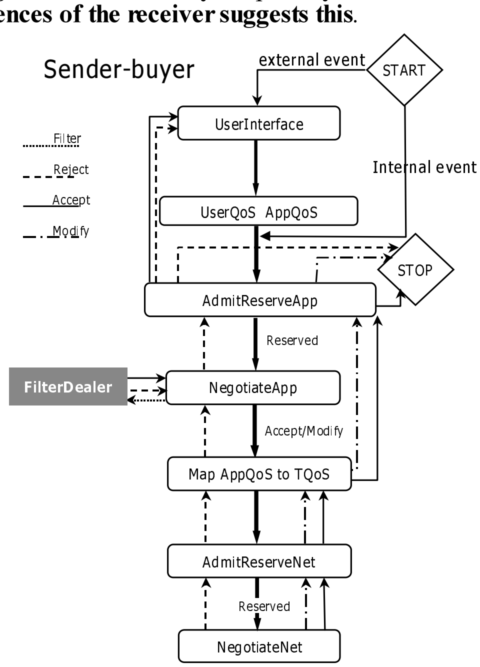

Figure 3: Sender-buyer protocol.

The broker-seller has three possibilities to react upon a request for certain AppQoS values*:* accept, modify, or filterrequest. We now briefly describe how the broker-buyer at a sender reacts upon a accept and modify signal sent back by a broker-seller at a receiver. QoS translation and negotiation continues at transport level by first invoking the QoS mapping functions at the broker-buyer, which translate AppQoS to TQoS intervals depending on a given media codec using the QoS mapper. The broker-buyer invokes admission tests for the transport sub-system like end-to-end delay and bandwidth or schedulability within the endpoints based on upper TQoS bound. After successful admission, the negotiation with the underlying network begins on a per-connection basis. A translation of TQoS NQoS is necessary and negotiation with intermediate network nodes is invoked on upper TQoS bound. If these TQoS parameters can not be supported, the lower bounds are tested and connection is established using lower bounds. Afterwards, the TQoS values can successively be increased towards higher TQoS interval boundary to increase TQoS and therefore AppQoS quality step-by-step. This caters for varying channel and QoS conditions in wireless networks. If not even the lower limit can be supported with the available resources, reject is signalled up to NegotiateApp *(*Figure *3)* for contacting the filter-dealer in order to look for a better compression method, resources at TQoS level are released and negotiation starts again with a filter-seller. Otherwise, the final step for the buyer is to wait for successful response from the resource management of the network and the response from the broker-seller at transport level.

If the broker-buyer receives a filterrequest signal, the filter-dealer is contacted with AppQoS intervals of sender and receiver and user identifications to contact an appropriate *(*set of*)* filter*(*s*)*. Two different architectural concepts exist for network filter placement*:* within an existing multicast tree, where filters are either installed at each branch from the source to the sink, or at selected fixed filter locations. It is the task of the filter-dealer to contact the appropriate filters depending on the sender or receiver driven approach. For the receiver driven multicast scenario, each filter may know the destination of its predecessor on the way to the source. The sender driven approach requires QoS aware routing protocols at AppQoS and at TQoS-level for selecting the best filters. Consider, for example, the case where receiver *4 (*Figure *1)* likes to join the conference and acts as a receiver-buyer. He would then contact the broker-seller within filter *2* with the desired App-QoS interval. Filter *2* already receives a compressed media stream, and if the AppQoS matches, the negotiation at TQoS level proceeds. In the case, that receiver *4* requests higher AppQoS values than filter *2* already receives, this would lead to a request for filter propagation *[19]* and the broker buyer within filter *2* contacts broker seller at filter *5* still at AppQoS level. Finally, this filter propagation may stop at a filter, which already supports the desired AppQoS interval, say e.g. filter *5*, and all intermediate filters are now tuned for scaling the media in a proper way and have reserved their resources at ApPQoS level. As a last step, the negotiation continues on TQoS level after mapping AppQoS to TQoS intervals both for incoming and outgoing streams. If there is a lack of network or system resources detected at an intermediate node, the filter propagation cannot be completed, the reserved resources are released and the receiver-buyer is informed. The original AppQoS values are lowered given the degradation path based on the users preferences and a re-negotiation is triggered at AppQoS level again.

For the receiver driven approach, two possibilities exist*:* either each receiver knows the next filter within the multicast tree towards the sender *(*static approach*)* or the receiver is unaware of the next filter. For the second case, the broker-buyer within the receiver has to contact the filter-dealer to get the best set of possible filters. However, this requires QoS aware routing protocols but gives more flexibility than the first approach. The mechanisms for contacting brokers at filter or receiver are similar to the sender driven approach. One disadvantage of the receiver driven multicast scenario can be seen in the fixed route *(*multicast tree*)* between sender and receivers. It might not be possible for a new receiver with high quality requirements to join the multicast group at a leaf, because this may lead to filter propagation which is not possible if an intermediate filter may get overloaded. For the sender driven approach, a filter-dealer may nevertheless find a different route using other lightly loaded filters.

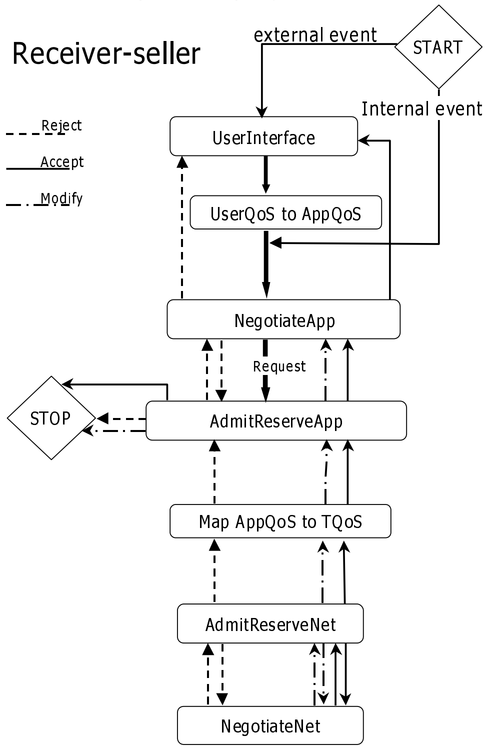

Figure 4: Receiver-seller protocol.

The broker-seller within an end node may receive signals from a broker-buyer within a network filter or a broker-seller within a network filter may receive signals from a broker-buyer at a sender or other network filters. Note, that for guaranteeing end-to-end delay and jitter, all delay and jitter introduced within intermediate filters have to be incorporated in the overall end-to-end delay and jitter calculation. In general, this limits the number of intermediate filters for real-time conferences due to additional processing overhead, where an end-to-end delay bound of 150 ms one way is desirable to guarantee interactivity. Our QoS negotiation and mapping protocol also supports distributional services like delivery of pre-recorded MPEG-2 video sequences. Here, a much longer delay (typical up to one second) can be tolerated which does not prevent the usability of several filters between source and sink.

The first task for the seller (Figure 4) is to wait for a request for certain AppQoS values (including the offered media compression methods) to be received from a sender broker-buyer. A comparison step is invoked between the received AppQoS and its own AppQoS intervals. The admission service is started based on the upper interval bounds and accept is returned to the remote buyer within the application subsystem if a match can be performed. If, however, the broker-sellers AppQoS is lower than the broker-buyers AppQoS and the seller does not have or does not want to increase resources to support the buyers AppQoS or the media codecs are not compatible, a filter-request is sent back to the broker seller with the desired AppQoS to be received and a set of supported media coding formats. It is then the task of the remote buyer to select a (set of) appropriate filter(s) by contacting the filter-dealer. At this stage, the communication between sender-buyer and receiver-seller aborts and the receiver-seller waits for a request from a filter-buyer and the protocol (receiver-seller) starts again.

The second and third phase start with a signal of the network management about network resource availability on behalf of the broker-buyer. Intervals at TQoS level are exchanged with broker-buyer to cope with varying network conditions. As a last step, a global admission service within the transport sub-system of the seller is started after a signal from the network resource management is received.

4.4.3 Network Filters and Filter-Dealers

In a multicasting environment with heterogeneous receivers and different QoS requirements in terms of access bandwidth or processing power, network filters can be used for scaling the media stream. In contrast to the well known approach where a source uses layered video coding methods to generate separate streams representing different quality levels and each receiver subscribes to as many channels as his QoS characteristics and resources allow [12] filters may selectively drop packets and offer more flexibility. The first approach is typically limited to few channels and quality levels due to synchronization and multiplexing overhead. WaveVideo allows to generate one stream based on tagged network frames, which enables filters to extract the desired quality and to chose from more than 1800 different quality levels at 25 fps. Filter operations for WaveVideo are extremely efficient and typically 5 µs are spent per packet [5]. A filter operation typically consists of one or two table lookups and a decision, whether the packet should be forwarded to the receiver or dropped. Note, that the tables for the network filters are prepared depending on AppQoS of sender and receiver and the preferences of the receiver.

In our framework the following types of filters may be used:

- Sender filters are invoked for the outgoing link, whenever an agreement between AppQoS has been reached but no mediation at TQoS level can be achieved. Typically, this situation arises when the network is not capable to support the calculated TQoS at the sender.
- Network filters are invoked, whenever there is a mismatch between requested and offered media type at sender and receiver, or when the receiver cannot cope with the offered AppQoS.
- Receiver filters scale down quality for adapting to mobiles processing power requirements. This is particularly useful, if a network filter at an AP serves several MTs with a superset of quality and each MT may then drop frames before decompression.

Sender filters try to reduce the network load without significantly influencing local CPU usage. Frame dropping for adapting the frame rate, subband dropping for adapting the frame size or color resolution or dropping of quantization layers for adapting the visual quality can be used for multi-resolution based video codecs like WaveVideo. The same mechanisms may be applied in network filters to adapt for heterogeneous receivers but clearly depend on the media codec.

Each filter registers at a dedicated filter-manager with its ID, filter-type, supported compression methods for incoming and outgoing connections. A filter-dealer can then determine, which filter-seller he should contact for performing the filter operations. Compression method specifiers for incoming and outgoing connections and AppQoS ranges are sent as parameters. The filter then chooses a possible filter strategy for media scaling given the AppQoS constraints and performs local admission tests. If no appropriate filter strategy is available or lack of resources is determined, the filter-seller sends reject to the filter-dealer, which continues searching all filters. There is no need for the sender and receiver to use the same compression methods, which leads to a very flexible QoS management architecture especially in the presence of mobile users. If, for example, the sender in the fixed network part uses an MPEG-2 encoder hardware card and sends high quality video to a multicast group, a mobile terminal may still retrieve the same video with a somewhat lower quality, if an intermediate filter transcodes MPEG-2 to a video stream, which the mobile is able to decode. For this case both sender and receiver determined a mismatch in the codecs during the negotiation phase at AppQoS level, so the filter-dealer may eventually find a suitable transcoding filter.

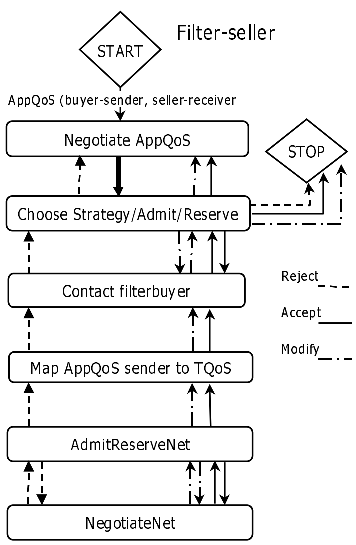

Figure 5: Filter seller

Otherwise, the filter-seller (see Figure 5) contacts the filter-buyer within the same filter to buy resources from the receiver-seller of the destination host or another intermediate filter-seller. In this case, he adopts the role of the original sender-buyer but includes filter processing time in the overall delay calculation. After successful negotiation at AppQoS level between sender and receiver, negotiation at TQoS-level continues starting between the last filter-buyer and the sender-receiver. Then negotiation at TQoS level continues between penultimate filter and last filter and so on. Finally, negotiation ends between sender-buyer and filter-seller of first filter at TQoS level. As a last step, a global admission service is started within the transport sub-system of each filters as a response to an acknowledge sent to all filters by the sender-buyer.

4.4.4 QoS Adaptation

Bad radio conditions are likely to result in a violation of existing traffic contracts. Applications should adapt automatically to hide the behavior of the wireless link to a user. QoS mapping functions based on intervals are used to cope with short term fluctuations. The current settings determine a working point within the given polyhedral set of QoS constraints and the system tries to stay within the bounds. Based on users preferences, the degradation path is calculated for adapting to long term QoS changes, when e.g. the mobile is handed over to the next radio cell with higher population and lower bandwidth available. The protocols may be started based on external events like user interaction (if the user changes his preferences or the interval boundaries) or internal events due to dropped connections or network QoS changes.

A filterrequest is signalled, whenever TQoS falls below the lower bound. If a filter-buyer receives a filterrequest from a receiver-seller, error control modules may be activated at the filter to protect the base layer of the video stream to cope with frequent short term fluctuations. Additionally, the filter may drop more packets in accordance with the users preferences to lower the output bandwidth and help mobiles to receive some video. As soon as the signal quality increases, the error control modules may be deactivated and the original quality can be sent. This process may lead to filter propagation towards the sender for the receiver initiated approach described above. Radio subsystems within access points typically know about the instantaneous air capacity and signal quality and can be incorporated into the QoS framework by sending signals to the brokers at certain time intervals to decide between short term (due to signal fading) and long term changes (due to completed hand-off). Timers may be used to determine, whether the received quality differs from the expected quality and a re-negotiation may be triggered after a certain period of time.

5. Conclusions and Future Work

Within this paper, we designed a unique architecture tailored to soft end-to-end QoS guarantees for heterogeneous networks and receivers. Wireless as well as wired networks and mobile and fixed terminals are integrated with media coders and decoders, error control modules and filters. End-to-end QoS negotiation is split between application and transport level. Mapping functions between different QoS levels are provided and the broker concept is extended to support network filters. The main ideas are based on the fact that due to varying channel conditions no hard end-to-end QoS guarantees can be given at transport level, when mobile networks are incorporated. Soft QoS guarantees are introduced via user definable QoS intervals at application and transport level. The overall system then tries to stay within the negotiated AppQoS interval as long as possible. The extension of the QoS broker concept to network filters allows an efficient tailoring of media streams for heterogeneous receivers under given end-to-end QoS constraints including media processing at intermediate filters. Automatic filter propagation under given AppQoS constraints is described for the receiver driven multicast scenario. The impact of short term network QoS fluctuations common in mobile networks is alleviated by user defined AppQoS intervals, usage of suitable error tolerant media codecs like WaveVideo and error control modules. Impact of long term network QoS changes is reduced by compliance to pre-calculated QoS degradation paths based on users preferences, usage of automatic network filter tuning and propagation for the receiver driven multicast scenario combined with end-to-end QoS regulation mechanisms based on a modified broker concept.

We are currently modeling our broker architecture and different protocols needed for negotiation among senders, filters and receivers using SDL. For a future implementation we are investigating Corba with real-time extensions as a flexible implementation platform for the extended broker architecture. Re-negotiation must be fast for handoffs and short term adaptations but standard Corba does not provide the necessary performance. The audio/video communication subsystem is prototyped and preliminary performance evaluations [8] showed a promising architecture with superior flexibility. Future work on the QoS architecture includes QoS based routing for the sender driven approach. How to find the best filter combination for a given system setup will depend on the overall cost introduced by network filter operations and network and end-system resources once network billing is used. Minimizing a user definable cost must be included into the QoS mapping and adaptation mechanisms. A more generic interface to media codecs is necessary for mapping between AppQoS and TQoS. Dynamic load balancing between all filters and re-routing of existing network connections based on the given end-to-end QoS constraints is another open issue.

Acknowledgement: We gratefully acknowledge the fruitful discussions we have had with colleagues in formulating these ideas, particularly George Fankhauser, Alfred Lupper and Jörg Ritz.

6. References

[1] Agrawal P., Hyden E., Krzyzanowski P., Mishra P., Srivastava M.B., Trotter J.A.: SWAN: A Mobile Multimedia Wireless Network, IEEE Personal Communications, April 1996, pp.18-33

[2] Besse L. et al.: Towards an Architecture for Distributed Multimedia Applications Support, Proc. of Int. Conf. on Multimedia Computing and Systems, May 1994, pp. 164-172

[3] Ciotti C., Borowski J.: The AC006 MEDIAN Project - Overview and State of the Art, Proceedings of 1st ACTS Mobile Communications Summit, Granada, Spain, November 1996, Vol. I, pp.362-367

[4] Fankhauser G., Dasen M.: An Error Tolerant, Scalable Video Stream Encoding and Compression for Mobile Computing, Proc. of 1st ACTS Mobile Comm. Summit, Granada, Spain, November 1996, Vol. II, pp.762-771

[5] Fankhauser G., Dasen M., Weiler N., Plattner B., Stiller B.: WaveVideo - An Integrated Approach to Adaptive Wireless Video, to appear in ACM MONET, Special Issue on Adaptive Mobile Networking and Computing

[6] Kalkbrenner G., Pirkmayer T., van Dornik, A., Hofmann P.: Quality of Service (QoS) in Distributed Hypermedia-Systems, Technical Report Nr. 6, PRZ-Nr.: TUB-PRZ-W-1103, Berlin

[7] Kassler A., Lupper A.: Support for Multimedia Applications in a Wireless ATM System, to appear in: Proceedings of ICATM 98, Colmar, June 1998

[8] Kassler A., Lupper A., Sun D.: A QoS aware Audio/Video Communication Subsystem for Wireless ATM Networks, Proc. of 3rd ACTS Mobile Communications Summit, Rhodes, Greece, June 1998

[9] Kassler A., Schulthess P.: Mobile broadband applications and the Magic WAND system, Proc. of Workshop on Personal Mobile Multimedia, Singapore, September 1997

[10] Krishnamurthy A., Little T. D. C.: Connection-oriented Service Renegotiation for Scalable Video Delivery, Int. Conf. on Multimedia Computing and Systems, Boston, MA, USA, May 1994, pp. 502-507

[11] Luo W., El Zarki M.: Analysis of Error Concealment Techniques for MPEG-2 Video Transmission Over ATM based Networks, Proc. of SPIE Visual Communications and Image Processing'95, Taiwan, May, 1995

[12] McCanne S., Vetterli M., Jacobsen V.: Low-complexity Video Coding for Receiver-driven Layered Multicast, IEEE Journal on Selected Areas in Communications, Vol. 16, No. 6, pp. 983-1001, August 1997

[13] Meierhofer J.: Data Link Control for Indoor Wireless ATM Networks, accepted for Wireless '98, Calgary, Canada, July 1998.

[14] Mikkonen J., Kruys J.: The Magic WAND: a wireless ATM access system, Proceedings of 1st ACTS Mobile Communications Summit, Granada, Spain, November 1996, pp. 535-542

[15] Nahrstedt K., Smith J.: The QoS Broker, IEEE Multimedia, Vol. 2, No. 1, 1995, pp. 53-67

[16] Nahrstedt K., Steinmetz R.: Resource Management in Networked Multimedia Systems, IEEE Computer, Vol. 28, Nr. 5, 1995, pp. 52-63

[17] Paskalis S. et al.: Performance of Traffic Scheduling for Wireless ATM Networks, Proc. of 2nd ACTS Mobile Communications Summit 1997, Aalborg, Denmark, November, 1997, Vol. II, pp. 751-756

[18] Pasquale J., Polyzos G., Anderson E., Kompella V.: The Multimedia Multicast Channel, Proc. of 3rd International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV 92), San Diego, California, November, 1992, pp. 185-192

[19] Pasquale J. et al.: Filter Propagation in Dissemination Trees: Trading Off Bandwidth and Processing in Continuous Media Networks, Proc. of NOSSDAV 93, Lancaster University, Lancaster, UK, 1993, pp. 269-278

[20] Schill A. et al.: A Quality of Service Abstraction Tool for Advanced Distributed Applications, Proc. of International Conference on Open Distributed Processing, February 1995

[21] Vogt C.: Quality-of-service management for multimedia streams with fixed arrival periods and variable frame sizes, Multimedia Systems, 1995, Vol. 3, pp.66-75

[22] Yeadon N., Garcia F., Hutchinson D., Shepherd D.: Filters: QoS Support Mechanisms for Multipeer Communications, IEEE Journal on Selected Areas in Communications, Vol. 14, No. 7, September 1993

[23] Zhang L. et al.: RSVP: A New Resource ReSerVation Protocol, IEEE Network, No. 9, 1993, pp.8-18