High Quality Mobile Communication

H. Hartenstein*, A. Schrader*, A. Kassler•, M. Krautgärtner†, C. Niedermeier†

* Computer & Communication Research Laboratories Heidelberg, NEC Europe Ltd
email: {Hannes.Hartenstein|Andreas.Schrader}@ccrle.nec.de

• University of Ulm
email: {Andreas.Kassler}@informatik.uni-ulm.de

† Siemens AG Munich
email: {Michael.Krautgaertner|Christoph.Niedermeier}@mchp.siemens.de

Abstract. Future communication environments have to support mobility at various levels ranging from device and personal to session and service mobility. Much effort is currently beeing spent in the areas of cellular access technology, wireless LAN technology and mobility support in IP (Mobile IP). There is a clear trend that the IP protocol is becoming the dominant networking protocol. Since standard IP networks do not provide any guarantees for the transmission quality parameters, there is a clear demand for a comprehensive QoS mechanism, which allows for adaptation in a mobile environment using heterogeneous devices with heterogeneous access networks.

In this paper we present a project on defining and implementing a comprehensive QoS framework for 'Mobility and Service Adaptation in Heterogeneous Mobile Communication Networks' (MASA). Our thesis is that, in order to provide high quality communication for mobile users, media processing facilities as well as mobility handling and handoff decision mechanisms should be closely integrated into a QoS framework. This allows, e.g., to base handoff decisions on *all* available QoS elements such as availability of transcoding units or local resource management. The *MASA* framework is able to release applications of QoS-related work as much as possible and, in addition, hides the complexity of network QoS mechanisms from the applications. The MASA QoS framework is able to support users with the ability to continue ongoing sessions even during handoffs and device changes (session mobility).

We present an outline of the general *MASA* architecture, consisting of distributed autonomous QoS Brokers that can be placed on the (potentially mobile) end-system, on intermediate network nodes (e.g. router, switches) and on transcoding units (gateways). The Brokers are supported by Managers and Controllers responsible for different tasks like resource, network, media, policy and mobility management. As an example, we describe the internal structure of the MASA *Mobility* Manager.

Keywords: *QoS, Mobility Management, Seamless Handover, Wireless Networks*

1 Introduction

The Internet offers challenging new opportunities for the development of communication applications such as video conferencing, multimedia distance learning, virtual reality group chatting, secure online shopping and banking, distributed games and many more. Some applications like IP telephony (or sometimes simplified VoIP) [7] have already become one of the most challenging business areas in the communication industry. In order to develop the Internet to be a widely deployed, commonly accepted communication medium allowing everywhere usage by everybody anytime, three major problems have to be solved.

First of all, the most well-known problem lies in the fact that today's Internet does not support any kind of guarantees for real-time multimedia streaming with adequate qualities. Therefore, effective methods for providing *Quality of Service* (QoS) are needed. Secondly, a less recognized problem is that among all of the above mentioned applications capturing, processing, transmitting, receiving and presenting media information are the crucial elements and have to be re-implemented within every application again. This results in an enormous waste of implementation resources and in a heterogeneous collection of incompatible applications. Thus, from our perspective there is a need for a *media architecture* that supports applications with multimedia streaming and processing facilities. Such an architecture eases the development of new challenging applications which can be implemented merely as a collection of graphical user interfaces using the architecture interfaces to realize the media processing. Last but not least, since more and more devices will be mobile, many multimedia applications should also be able to run in mobile environments. However, mobile environments are characterized by low bandwidth (compared to wired networks), quickly changing link qualities, as well as a wide variety of different access technologies. Therefore, taking also the mobility-specific needs into account we propose a comprehensive *media-centric QoS architecture with mobility support*. With this architecture a mobile user will be supported to get the highest possible quality according to some user criterion wherever she is located. The decision of the actual transmission parameters should be based on monitoring information about the status of the network as well as the local end-system resources and has to provide a tradeoff between the user's QoS policy settings and the network operators service level agreements.

The rest of the paper is organized as follows. In section 2 we review some related works. Section 3 introduces the *MASA* QoS framework. Whereas subsection 3.1 outlines the general design of the *MASA* system, subsection 3.2 describes the typical structure of *MASA* on end-systems. It is out of the scope of this paper to give detailed descriptions of all the broker's manager components. Instead we will restrict ourselves to a discussion on mobility management aspects in section 4. Section 5 summarizes the concept and gives some ideas for future extensions.

2 Related Works

In recent years, a number of approaches have been proposed to enhance the Internet with different Classes of Services (CoS). Most of them concentrate only on certain aspects of the overall QoS problem, like media filtering [16] or resource reservation [5]. Others only operate on certain layers of the communication model, like the network layer (e.g. DiffServ [2], IntServ [12]) or the application layer [11]. Some architectures only cover certain entities of the end-to-end transmission path, like the end-system [10]. Some architectures are integral, but some key features are missing, like inter-session relationships [3], or load-balancing and fast handover [9]. None of the approaches provide separation of the actual media processing activities from the application and the combination with QoS issues. For further references and comparisons of QoS architectures, see [1, 9]. On the other hand, frameworks have been developed to provide applications with media processing facilities, for example the Java Media Framework (JMF)[14]. However, these frameworks do not provide any kind of QoS mechanisms.

Although all mentioned approaches provide important mechanisms for parts of the QoS problem, an overall optimal solution can only be achieved, if all mechanisms are handled within an integrated comprehensive end-to-end management system. In the following we introduce the *MASA* QoS framework as such an integrated architecture.

3 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. The *MASA* framework was designed to fulfill the requirements of a *comprehensive integrated end-to-end QoS multimedia management system*, invoking all entities on the transmission path, like sender, network nodes, transcoding nodes, switches, routers, filter nodes, gateways and receivers. *MASA* allows the usage of underlying network layer QoS technologies (e.g. DiffServ, IntServ, MPLS, etc.), and also hides the complexity of these mechanisms 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.

The framework follows an object-oriented design and most of the components (except some operating system specific tasks) are implemented in Java. This allows for downloading plugable components from different parties. Through the usage of open interfaces we can reach a high flexibility. The design also fulfils other important rules like separation of media processing and control or separation of tasks with different timing constraints etc (see also [3]). Since not all of the possible components have to be used on all locations, our design also provides scalability which is extremely important if we consider the usage of *MASA* on terminals with limited facilities, e.g., future UMTS devices.

3.1 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 is responsible for the brokerage between managers with very different tasks, like resource, network, media, monitoring, policy and mobility management.

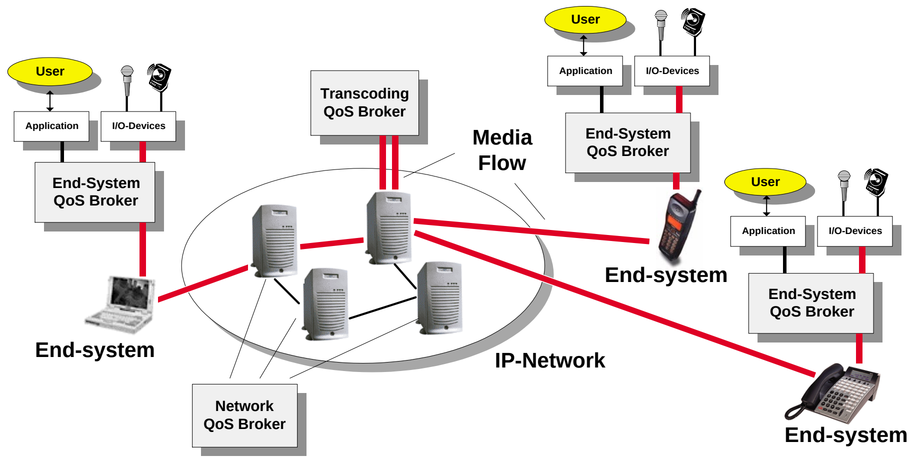

Fig. 1. Distributed set of autonomous QoS Brokers.

The main task of the *End-System QoS Broker* is to coordinate, orchestrate and manage local and remote resources for multimedia streaming and service quality, 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, Wireless-LAN, UMTS, etc.) We will describe the *MASA* end-system QoS Broker in detail in section 3.2.

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 the 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* can be instantiated if heterogeneous clients must be supported in a multi-party conference scenario or if special network characteristics on certain network links have to be supported with mechanisms like adaptive Forward Error Correction (FEC), appropriate transmission protocols (e.g. wireless-TCP), etc. Transcoding QoS Brokers can also realize simple filter mechanisms. In our framework we use filter routers to realize priority filtering based on priority settings within the RTP header. This allows for very fast and fine-grained adaptation schemes which are combined with coarse-grained sender rate adaptation schemes in the end-systems.

Communication between this 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.

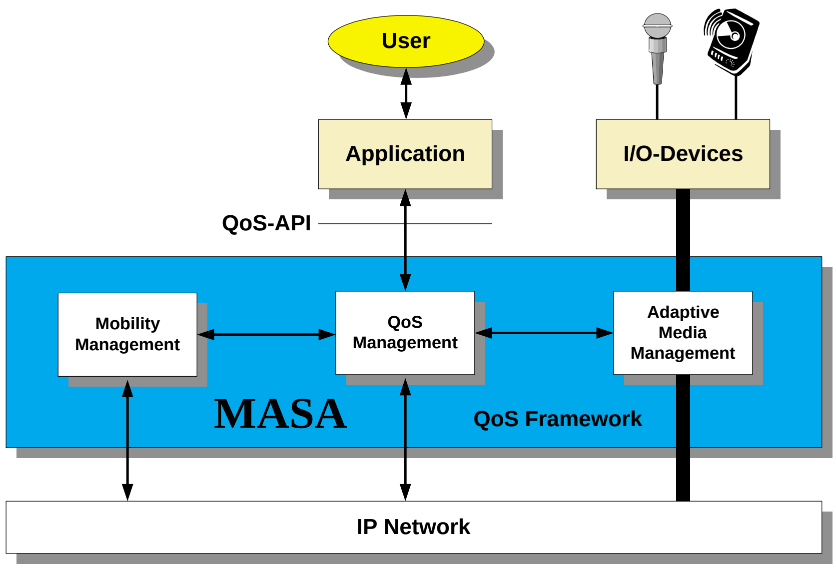

Fig. 2. General Structure of *MASA* on end-systems.

Fig. 2 outlines the general concept of *MASA* on end-systems. As a major feature, *MASA* separates the adaptive media management from the applications. The media management is controlled by the QoS management which is closely coupled with the integrated mobility management.

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. Applications subscribe to the system and use the provided facilities via a respective QoS-API. This API can be used to instantiate multimedia sessions. Within the *MASA* QoS management, the hierarchical concept of participants, session, streams and flows is used. Each participant can have several media sessions with other participants (at remote machines) at the same time. The sessions consists of an arbitrary number of streams with one or more actual media flows. This allows for the usage of layered coding.

The set-up of a complete chain of media processors, consisting of capture devices, codecs, effect processors, etc., can be controlled via this API. Appropriate graphical user interfaces are provided to present media information (e.g. video panel). Since *MASA* provides a flexible mechanism to *plug-in* arbitrary components to capture, process, code, transmit, receive, decode and display any kind of media, the applications are shielded from that low-level complexity. Adaptive and layered coding can be used seamlessly by any application to allow for scaleable media transmission.

With respect to mobility, efficient handoff algorithms are a cost-effective way of enhancing the capacity and QoS of cellular systems [15]. *MASA* supports mobile devices by integrating mobility management into the framework and using fast QoS re-negotiation and adaptation mechanisms to allow seamless intra- and inter-technology handoff (see section 4).

Adaptive media processing is controlled via *trading policies*. *MASA* offers appropriate intuitive graphical user interfaces to capture user QoS preferences (policy GUI) allowing the selection of a certain CoS as well as appropriate degradation paths for situations with changing resources or transmission link characteristics. On network nodes, preference-controlled management of QoS capabilities for network administration is also supported.

The *MASA* QoS Management provides *mapping functions* between QoS parameters at the various levels (user, application, framework, operating system, network sub-system and network layer), to hide underlying QoS parameters from users and applications and support applications with mechanisms to allow *soft* and *hard QoS negotiations*.

Group conferencing is enabled with and without the use of multicast mechanisms. This is particularly challenging for heterogeneous devices with varying capabilities, like resources, media processing mechanisms, etc. Clients are supported with *capability exchange* mechanisms in order to agree on a certain service quality and to dynamically join and leave ongoing sessions. To support different service levels for group communication *audio and video filtering* is used on network level (packet-based) as well as on application level (content-based). To support heterogeneous devices with incompatible communication mechanisms, appropriately placed *transcoding units* can be used. These units allow for downloading codecs on demand. The placement and optimization of transcoding and filter units can follow intelligent algorithms to optimize network load and processing requirements.

The provision of differentiated CoS demands for the introduction of charging and billing mechanisms. The *MASA* architecture supports control of media quality in relation to a cost-over-quality function which is part of the user's QoS preferences GUI.

In the following subsection we outline one possible structure of *MASA* components on end-systems.

3.2 The End-System QoS Broker

Fig. 3 outlines a typical structure of *MASA* on end-systems, like multimedia terminals. The *End-System 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.

Fig. 3. Hierarchical structure of *MASA* on end-systems.

The *Policy Manager*, e.g., is responsible for the storage and retrieval of QoS preferences within a user profile and for presenting an appropriate policy GUI to the user. The Policy Controller enables the access to a policy database (in accordance with the IETF QoS Policy Framework [6]) for storing the profiles as well as QoS trading policies used inside the QoS Brokers. The *Resource Managers* are responsible for controlling the available resources (like CPU, memory, network, etc.) via the respective Resource Controllers. This is of special importance for real-time communication with limited tolerance against delay and jitter. 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 statistic information to the Broker. The *Intercom Manager* is used to allow inter-Broker communication. The *Application Manager* provides applications with their specific needs. This allows to support different categories of applications, like VoD or IP-Telephony. The Application Manager maps typical requests from applications (e.g. SIP protocol sequences) to the Broker QoS API. With this mechanism we achieve a better flexibility and also support legacy code. The *Mobility Management* is responsible for the support of device mobility and enables the usage of different access devices. The Mobility Manager will be explained in detail in the next section.

With this structure, the *MASA* framework is perfectly able to support all kinds of multimedia applications. Regarding local and remote resources, available media mechanisms, user preferences and network admission policies, the QoS Broker is able to choose an appropriate compromise and to provide a certain CoS by using underlying network layer QoS technology. The QoS Broker can produce an appropriate compromise between resource reservations (via RSVP; mainly in corporate managed LANs), appropriate CoS (via DiffServ Codepoints; mainly in the backbone) and adaptive scaling mechanisms. The Broker regularly requests monitoring information from the connected managers. The aggregated monitoring information together with the user's QoS policy is used as input for the included 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 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.

Through the use of open interfaces between the QoS Brokers and Managers, the system can be easily extended with 3^{rd} party provided Manager/Controller pairs and, therefore, provides a future proof-concept for the integration of further mechanisms (e.g. Managers for location awareness using GPS Controllers). We use a combination of a generic interface description which must be implemented by all involved managers and more specific interfaces for each manager. The generic interface allows for basic control of participants, sessions, streams, and flows, whereas the specific interfaces can offer methods which are only important for the specific manager, e.g. defining a thread priority via the CPU Resource Manager. This plug-in mechanism can be explicitly used to support any (future) underlying network technology (e.g. wired or wireless access networks). For example, within the *MASA* project, different realisations of the Media Manager are provided by the project partners. The NEC *AQUARIUS* (Adaptive Quality of Service Architecture of Intelligent Universal Services) Media Manager is built in Java. Through the usage of JMF [14] as a Media Controller *AQUARIUS* is able to download, install and maintain a broad range of audio/video codecs to filter and transmit media, ranging from low bandwidth speech codecs (e.g. GSM) to high quality video streaming (e.g. H.261/H.263, MPEG-1/2/4).

4 The MASA Mobility Manager

In order to allow a mobile user to roam between various operators, between public and private networks, between wired and wireless networks, and between different access technologies, the mobility management within the *MASA* framework is based on Mobile IP (for an introduction cf., e.g., [13]). As a layer 3 approach, Mobile IP provides a more generic mobility management than any L2 mechanism and at the same time releases applications of any mobility concerns. In our current implementation Mobile IPv4 is used but could be substituted by a Mobile IPv6 implementation because of the modularity of the architecture.

Mobile IP can be conceptually split up into the following three parts: *i)* signalling and configuration, *ii)* data transport, and *iii)* movement detection plus handoff decision. The movement detection mechanisms specified in Mobile IP are based on whether a mobile node is able to hear some agent advertisements that are periodically sent by home or foreign agents. Since these advertisements are usually sent only once per several seconds, no fast handoffs can be achieved with these mechanisms. Therefore, we use (downlink) signal quality measurements to achieve faster movement detection and focus on mobile node initiated handoffs.

The mobility management part on a *MASA* end-system has been structured as illustrated in figure 3. The *Mobility Controller* is a standard Mobile IP daemon with the difference that it does not implement any handoff decision mechanism. Thus, the Mobility Controller is able to send/receive and process Mobile IP signalling messages like registration requests/replies, but performs any action only when told so by the Mobility Manager. For each network interface an *Access Network Monitor Controller* (ANMC) measures link quality parameters and reports them to the Mobility Manager. The quality parameter might be a boolean value like link integrity in the case of Ethernet or some real value for example measuring signal-to-noise ratio in the case of a wireless LAN. The *Mobility Manager* processes the information of the ANMCs and either directly forces a handoff or informs the QoS Broker of available access options. In the case when no foreign agents are present in a access network, the mobile node has to operate in co-located care-of-address mode and, therefore, has to be able to acquire a topologically correct IP address by means of DHCP. For this purpose, an *IP Address Controller* might be needed in addition.

The purpose of the tight integration of mobility management and handoff control with the QoS architecture is to support seamless handoffs for realtime communications. The QoS Broker has a more complete knowledge -compared to the Mobility Manager- a handoff decision can be based on. While the Mobility Manager deals with IP connectivity, the QoS Broker can also take into account issues related to service mobility and admission control. For example, when the Mobility Manager advertises a new access network, the QoS Broker (with the help of all managers) can base the decision whether to handoff to the new network on cost policies, available codecs in the Media Manager, required resources at the processor, the current location, and so on. Thus, a handoff is only initiated when a successful completion of the handoff can be expected. This aspect is particularly important for future scenarios with a wide range of wireless access alternatives.

The interface between Mobility Manager and QoS Broker essentially consists of methods *new_network* and *removed_network* that announce the addition/removal of an access option to the QoS Broker as well as methods *request_handoff* (QoS Broker requests a handoff to a specific network) and *get_parameters* (QoS Broker requests some quality parameters). In addition, a method *set_threshold* allows the QoS Broker to register an event filter at the Mobility Manager in order to avoid excessive polling of quality parameters.

We achieve seamless handoffs by following the *make before break* philosophy, i.e., all negotiation for registering in a new network are done while the mobile node is still able to send/receive packets over the old network. In the case of an inter-technology (vertical) handoff this can be easily achieved: when for each access technology the mobile node is equipped with a specific interface, the mobile node is able to listen to one interface while registering with a new network over a different interface. Thus, the handoff problem is basically a *multi-homing* problem. The new registration is done via the new network (called a *forward handoff*). In the case of an intra-technology (horizontal) handoff the interface can usually only communicate on IP level with one base station at a time (for example, in IEEE 802.11 a mobile node is associated with a single access point when using infrastructure mode). Thus, a typical handoff procedure will first break the old *mobility binding* before establishing the new one via the new access network. In order to be able to benefit from the "make before break" approach even in the case of an intra-technology handoff, we have developed a method where the new mobility binding is established via the old network (i.e. a *backward handoff*) and *simultaneous bindings* are used [8]. Thus, packets destined for the mobile node will be delivered to the old and new base stations, and the mobile node only has to reconfigure its interface in order to complete the handoff.

In our current testbed, the mobile node is equipped with Ethernet, WaveLAN, and GSM cards. The Mobility Controller is running Dynamics Mobile IP [4] in co-located care-of-address mode. The interface between Mobility Controller and Mobility Manager is based on the Dynamics API. The interface for the WLAN ANMC is based on the Linux Wireless Extension. The Ethernet ANMC currently checks only the link integrity flag. The 'make-before-break' approach leads to times for physical interruption of about 5 msec, i.e., seamless handoffs can be performed.

5 Conclusion and Future Work

We have presented the *MASA* Framework as a comprehensive integrated end-to-end QoS multimedia management system with mobility support for heterogeneous wireless environments. A distributed set of autonomous, flexible QoS Brokers, located on end-systems, network nodes and transcoding entities can be easily extended with new Manager/Controller pairs through the usage of open interfaces. By using exchangeable trading policies, all aspects of mobile multimedia communication can be organised and managed by a de-centralised intelligence mechanism. As an example, we demonstrated the interworking of the QoS and mobility management systems for seamless handoffs of mobile devices. The framework has been implemented and the interworking of the various components of the different project partners has been tested. As example applications, we have realized small prototypes for Video on Demand, Video Conferencing, Audio Broadcasting and IP-Telephony.

Currently we are integrating DiffServ routers in our testbed and develop mapping functions between QoS policies and DiffServ classes for the above mentioned application scenarios. Future work will concentrate on appropriate trading policies and the provision of new management components, for example location awareness for improved handoff detection.

References

1. C. Aurrecoechea, A. Campbell, and L. Hauw. A Survey of QoS Architectures. *Multimedia Systems Journal*, 6(3):138–151, May 1998.
2. A. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *RFC2475: An Architecture for Differentiated Services.* IETF.
3. A. Campbell, G. Coulson, and D. Hutchinson. A Quality of Service Architecture. *ACM Computer Communication Review*, April 1994.
4. Dynamics. *Mobile IPv4 implementation.* http://www.cs.hut.fi/Research/Dynamics/.
5. L. Zhang et. al. RSVP: A new Resource ReSerVation Protocol. *IEEE Network*, (9):8–18, 1993.
6. S. Gai, J. Strassner, D. Durham, S. Herzog, H. Mahon, and F. Reichmeyer. *Internet Draft: QoS Policy Framework Architecture.* IETF, February 1999.
7. S. Gessler, O. Haase, and A. Schrader. A Service Platform for Intelligent Internet Telephony. In *The 1st IP Telephony Workshop (GI)*, April 2000.
8. H. Hartenstein, K. Jonas, and R. Schmitz. Seamless Inter-Domain Handoffs via Simultaneous Bindings. *Proc. European Wireless 2000, Dresden, Germany*, September 2000.
9. A. Kassler and P. Schulthess. An End-to-End Quality of Service Management Architecture for Wireless ATM Networks. In *Proc. HICSS'32*, Hawaii, Jan 1999.
10. K. Lakshman and R. Yavatkar. AQUA: An Adaptive End-System Quality of Service Architecture. In *High Speed for Multimedia Applications*, pages 155–177. Kluwer Academic Publishers, 1996.
11. K. Nahrstedt. *An Architecture for End-to-End Quality of Service Provision and its Experimental Validation.* PhD thesis, Department of Computer and Information Science, University of Pennsylvania, August 1995.
12. D. Clark R. Braden and S. Shenker. *RFC1633: Integrated Services in the Internet Architecture: An Overview.* IETF, June 1994.
13. J. D. Solomon. *Mobile IP.* Prentice Hall, 1998.
14. Sun. *The Java Media Framework Version 2.0 API.* http://java.sun.com/products-/java-media/jmf.
15. N. Tripathi, J. Reed, and H. VanLandingham. Handoff in cellular systems. *IEEE Personal Communications*, Dec. 1998.
16. N. Yeadon, F. Garcia, D. Hutchinson, and D. Shepherd. Filters: QoS Support Mechanisms for Multipeer Communications. *IEEE Journal on Selected Areas in Communications*, 14(7), September 1993.