Signaling Architectures for the Interaction of the Session Initiation Protocol and Quality of Service for Internet Multimedia Applications

A Dissertation
Presented to
The Academic Faculty

by

Ana Elisa P. Goulart

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
April 2005

Signaling Architectures for the Interaction of the Session Initiation Protocol and Quality of Service for Internet Multimedia Applications

Approved by:

Professor Randal Abler, Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Ashraf Saad
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Henry Owen
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Jun Xu
College of Computing
Georgia Institute of Technology

Professor Douglas Williams
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: April 14, 2005

To Augusto and Eric

ACKNOWLEDGEMENTS

I would like to thank all the faculty and staff of the Electrical and Computer Engineering Department at Georgia Tech. I have learned so much from so many good teachers in this department, and could always count on the support of an efficient and friendly administrative staff.

In special, I owe much to my advisor, Dr. Randal Abler, whose support, advice and confidence made this work possible. Although we worked in different locations, he was able to combine the use of multimedia applications and close meetings to help me throughout my research.

Also, I am really grateful for the friendship of Ms. Gail Palmer, lecturer of the Professional Communications Skills program. Since taking her class, she has supported my work by revising so many documents that I wrote, and by being a very close friend whom I could always count on.

Finally, I have no words to thank my whole family for their support. My parents, brother and sister who, although faraway, followed my work and always cheered for me. In special, Augusto, my husband, and Eric, my little one. Augusto's hard work, countless feedbacks, and patience have been essential to the completion of this thesis. Eric has brought a lot of happiness to my studies. And more happiness is coming with a little girl, who will soon arrive to this world...

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Although there have been many attempts to provide quality of service (QoS) in the Internet, users may experience confusion on how and when to apply QoS. As a result, the communication in the Internet remains best-effort. This thesis addresses interactive multimedia sessions (e.g., video-conference), which combine requirements of traditional telephony services and Internet applications. This requires call setup, call signaling, negotiation, routing, security, and network resources. Therefore, an interaction between session control signaling and resource management mechanisms is needed to facilitate the use of QoS mechanisms to users of such applications.

In this thesis, new signaling architectures for the interaction of session control signaling and resource management are proposed. The architectures are based on the interaction of existing protocols: the Session Initiation Protocol as the session control protocol, and current QoS architectures. The Differentiated Services (DiffServ) architecture is used as the primary example. The interaction of SIP and resource management addresses resource negotiation, call authorization, and end-to-end QoS in heterogeneous networks. In order to evaluate if the proposed signaling architectures are indeed improving the use of QoS in heterogeneous networks, testbed and simulation experiments are used to evaluate the architectures' signaling efficiency, the performance improvements they can bring, and how easy and available they are to end users no matter their access device or type of network they are connected to.

CHAPTER I

INTRODUCTION

The Internet has revolutionized the way we communicate. In special, interactive multimedia applications, which are usually referred to as Internet telephony applications, have impacted our lives in many ways. For instance, in the schools video-conferencing has allowed new means for distance learning [59], video-conference systems for telemedicine applications has enabled remote patient monitoring and consultation [106], and at home or in the work environment voice-over-IP has become an alternative way of making phone calls [96]. These applications combine requirements of traditional telephony services and Internet applications. This requires call setup, call signaling, negotiation, routing, and network resource management. However, in dynamic network environments, such as the Internet, network resources are not protected unless Quality of Service (QoS) mechanisms are deployed and available to the applications.

Although the Internet Engineering Task Force (IETF) has been working on the development of QoS mechanisms [7, 116], they are not widely deployed and there is no unique common solution to provide QoS in the Internet [102]. Typically, the Internet provides best-effort services. Furthermore, the variety of access devices and network infrastructures requires heterogeneous QoS mechanisms to support an end-to-end communication path. For instance, some access devices have limited capacity and may not directly support QoS; also, different networks that employ different QoS mechanisms require means to manage both the policies and the QoS control protocols in order to provide end-to-end QoS. Thus, in addition to the fundamental need for QoS support for real-time communication, there is a need for compatibility across network equipment and protocols.

One way to facilitate the use of QoS mechanisms and to protect network resources to Internet multimedia applications is to develop efficient ways that will allow these applications to take advantage of the underlying resource management mechanisms. This is the main

goal of the proposed work presented in this thesis, which seeks to improve the interaction of session control signaling (at the application level) and network resource management mechanisms (at the network level). From the viewpoint of protocol layers, this interaction is illustrated in Figure 1, with the Session Initiation Protocol (SIP) [86, 92, 96, 112] as the focus of this signaling work.

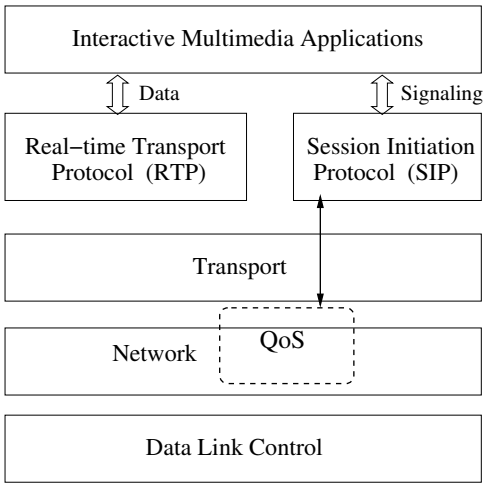


Figure 1: The interaction of session control signaling and QoS.

The Internet's network resource management and session control signaling mechanisms are independent and have their roles well defined. In general, the role of network resource management includes differentiated treatment for packet forwarding (i.e., quality of service), route decision and dissemination, and signaling involved in admission control and QoS negotiation [83]. On the other hand, session control signaling protocols focus on connection and call control, for establishing, modifying or terminating multimedia sessions. Moreover, both network resource management signaling and session control signaling are separate from the actual data transmission, which typically uses the Real-time Transport Protocol (RTP) [97] to send multimedia content end-to-end.

On the interaction of network resource management and session control signaling mechanisms, there have been different approaches proposed in the literature [19, 48, 66, 94, 99, 102]. They focus mostly on the call setup phase and resource-reservation-based QoS mechanisms, in an effort to provide a signaling scheme that minimizes call defects which are caused by the lack of available resources in the network. Their approaches vary in terms of

architectural complexity, level of integration and type of interaction between session control and resource reservation protocols. In general, they address more strict-type call setup schemes in order to provide assured services, as in traditional telephone networks. Furthermore, most of those studies lack either performance analysis or experimental results on the proposed schemes.

The interaction of SIP and resource management requires additional signaling exchange between end users. Additional control signaling impacts the delay users will experience and the signaling load in the network during the call setup transaction. In order to evaluate this impact, and motivated by the need to investigate other interesting approaches to the integration of application-layer and resource management protocols, the main research goal is to create new signaling models that can provide more flexibility to interactive multimedia applications (e.g., telephony-style vs. Internet-style reservations). Next, the research objectives and the proposed solutions are briefly described.

1.1 Research Objectives and Solutions

This thesis addresses the need for end-to-end call signaling in heterogeneous IP networks to protect resources for real-time multimedia streams. New signaling architectures that contribute to the interaction of SIP and current resource management frameworks are presented considering the role of SIP agents and servers and their interface with the network layer entities. First, the role of QoS-enhanced or QoS-enabled SIP proxies is discussed. In particular, their role in requesting resources on behalf of users who may not have access to QoS reservation protocols is emphasized. Based on the Differentiated Services architecture, a lightweight signaling scheme is proposed for the interaction of QoS-enhanced SIP proxies and DiffServ edge routers. This scheme, which can provide loose QoS guarantees to users, is an example of a one-way SIP and resource management interaction in a QoS-enhanced SIP proxy-based architecture

The integration of SIP and resource management proposed by the IETF (RFC3312) [19] defines new ways to communicate QoS information at the application layer and addresses a resource-reservation-based approach to QoS. The impact of the signaling needed for this

integration and of the reservation delays during the call management phase are evaluated through testbed experiments. Then, motivated by the idea of efficiently using the idle times (e.g., reservation and user answering delays) during the call setup transaction, an experimental signaling work on the Resource management Overlapped with Answering Delay (ROAD) call setup architecture is proposed. Basically, this signaling architecture applies a different and more flexible call setup model than the traditional telephony-style model.

The idea of QoS-enhanced SIP proxies can be further applied in a two-way SIP and resource management interaction (Figure 2), which uses the concepts of the integration of SIP and resource management to authorize network resources for the call, considering the current network load (i.e., two-way interaction) and allowing an integrated and more granular call authorization model between origin and destination domains. In addition, the increased processing load that this model may impose on SIP proxies is analyzed using queuing models, which can help on evaluating the scalability of the proposed architecture.

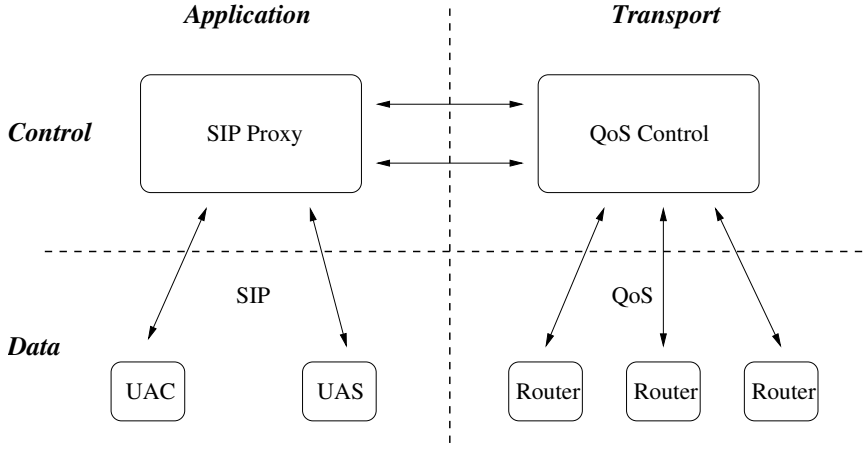


Figure 2: Two-way SIP and resource management interaction.

To achieve end-to-end QoS, the role of application-layer signaling has growing importance, especially in heterogeneous network environments, each with their own policies and QoS schemes. The heterogeneity happens mostly at the transport level. Thus, this work envisions that the application-layer can add some type of control to harmonize this heterogeneous environment. One scenario that will be considered is that of mobile networks with different infrastructures and signaling protocols.

1.2 Thesis Outline

This thesis is organized as follows. In Chapter 2, a survey of current state-of-the-art work in the area of cooperation between session control signaling and resource management mechanisms is presented. Chapter 3 presents an architecture with a QoS-enhanced SIP proxy and its signaling scheme based on the DiffServ architecture. Then, Chapter 4 presents the Resource management Overlapped with Answering Delay (ROAD) call setup scheme. Chapter 5 is about the role of QoS-enhanced proxies in the interaction of call authorization and QoS-enabled media authorization. In Chapter 6 the application of the above mentioned architectures are considered in a heterogeneous network with the objective of achieving end-to-end QoS. Finally, this thesis concludes in Chapter 7 where the main research results are summarized and a number of problems for future investigation are suggested.

CHAPTER II

PREVIOUS WORK

To develop an understanding of the issues of the interaction of application-layer signaling and network-layer QoS mechanisms, first this chapter provides an overview of session control signaling, including existing protocols. Among these protocols, emphasis is given to the Session Initiation Protocol (SIP), where SIP's history and applications, its basic signaling flow, and the role of SIP proxies are discussed. Then, a review of current models proposed in the literature to provide the interaction of session control and resource management is presented.

2.1 Session Control Signaling

Session control signaling consists of the set of messages and procedures to set up, modify, or terminate multimedia sessions. For example, signaling protocols are required to coordinate connections of voice-over-IP and video-conference sessions. In general, session control protocols focus on connection and call control, being separated from the actual transmission of multimedia content between session participants.

To perform connection and call control functions, basic signaling protocols must provide the following services [69, 96]: user name translation and location services, capability negotiation, call participants management, and gateway services. In addition to the need to identify the location of the called party (i.e., *name translation and location service*), users that want to establish a multimedia session must all agree on media formats (encoding and bandwidth) that they support, and on network addresses and port numbers to receive the media flows. The term *capability negotiation* is usually used to refer to this type of media negotiation that occurs between users at call setup. Moreover, capability changes may occur during an active session. Also, active sessions can have call participants added or removed; thus, signaling protocols must provide *management of call participants* during an on-going

call. Finally, *gateway services* are needed to provide an interface with telecommunication networks, such as the Public Switched Telephone Network (PSTN).

Two protocols provide call control services to Internet telephony applications: the Session Initiation Protocol (SIP) [92] and the H.323 protocol suite [56]. Comparisons between these protocols can be found in the literature [29, 31, 44, 98]. In summary, their differences are originated in the nature of their designs: the architectural philosophy, the standards process, and details such as ASN.1 encoding (H.323) versus text encoding (SIP). To be more specific, the H.323 suite has strong roots on traditional telephony/Integrated Services Digital Network (ISDN), being part of the H.3xx family of protocols of the International Telecommunications Union (ITU-T) (e.g., H.320 for ISDN and H.321 for Broadband-ISDN (B-ISDN)). SIP uses a different model, however. Its IETF's model is transaction-oriented, with similarities to the HyperText Transfer Protocol (HTTP) that is used to transfer and display files on the Internet.

2.2 The Session Initiation Protocol (SIP)

2.2.1 Why SIP?

This thesis is about the interaction of SIP and resource management mechanisms. There are several reasons why SIP is of special interest. First, its design is based on the Internet architecture, which allows an easy integration with other Internet protocols (e.g., links to SIP services can be integrated in web pages). Also, it operates in a client-server mode, and new services can be implemented either at the end user or intermediate servers. Second, scalability and reliability are some advantages of SIP, since SIP servers do not keep state for the whole session, but usually for only a single transaction. Thus, they can handle a large number of subscribers, and calls will not be affected if they become out of service. Finally, SIP's design has considered application-layer mobility [100]. Application-layer mobility is allowed in different ways: personal mobility, in which SIP requests can be directed to several of the user's possible locations (that is, users' identification is independent of their actual physical address in the network); or session mobility, which allows users to transfer an on-going session from one access device (e.g., a cellular phone) to another (e.g., a desktop

computer); or service mobility, that provides the same services to the users independent of their location or access device. The use of SIP for mobility management and its signaling performance has been studied in [8, 63].

After an initial version of SIP - SIPv1 - that was developed in the mid 1990's, the second version became an IETF standard in 1999 (RFC2543) [52]. After several revisions that added security measures and more detailed descriptions of the protocol's functional behavior, this standard has been outdated by a newer one (RFC3261) [92]. From the beginning, the main goal of SIP designers was to create a generic session signaling protocol that could be used not only for multimedia sessions but also multi-player games, bank transactions, instant messaging, and many other applications. For instance, [71] proposes the use of SIP for communication among networked home appliances. On the field of voice-over-IP, SIP phones, servers and gateways are being implemented by several companies. Furthermore, one important application is the use of SIP as the signaling protocol in the core network (i.e., IP backbone) of third-generation (3G) wireless systems [4], where IP-based protocols facilitate the integration between fixed and mobile networks [13, 113].

2.2.2 Network Architecture

SIP's design tries to keep its functionality to only session control functions, but it allows cooperation with other protocols. SIP is an application-layer protocol that typically uses the services of either the reliable connection-oriented Transport Control Protocol (TCP) or the unreliable connectionless User Datagram transport Protocol (UDP). In the latter case, message retransmissions at the application level are specified in the SIP standard (RFC3261) in order to provide a reliable signaling exchange. Other transport protocols can also be used to carry SIP messages, such as the Stream Control Transmission Protocol (SCTP) [18, 91] which is a connection-oriented transport protocol originally intended to transport telephony signaling over IP networks. SCTP features include protection against Denial of Service (DoS) attacks, efficient use of the available IP interfaces in a multi-homed host, and message-oriented delivery (i.e., delivers complete signaling messages). According to [18], the use of UDP is recommended in the case of light signaling loads. However, the

application-layer retransmissions used with UDP are not appropriate for heavy traffic loads. In this case, TCP or SCTP are more appropriate.

As in a typical client-server transaction, SIP messages can be in the form of requests or responses. When transported over UDP, a SIP message is self-contained in one datagram, which carries text information. The first text line informs the type of request (e.g., INVITE, REGISTER) or the type of response (e.g., provisional responses such as "180 RINGING", or final responses such as "200 OK"). The message also has a header, which resembles an email header, containing addresses, session identification, signaling path, routing directives, type of message bodies, among other fields.

Some of the requests and responses have a body attached to the message. As SIP works with other application protocols, the Session Description Protocol (SDP) [51] is usually adopted to format the body of the messages. They describe the media parameters within the SIP messages. To illustrate the media flow descriptions in SIP messages using SDP, Figure 3 shows a typical body of a SIP message that has two flow descriptions (audio PCM μ-law and MPEG video), each one bi-directional. Note that the flow descriptions include dynamic information about the address to which send the media (e.g., port 49170 for audio).

SDP Example with Two Flows

General Session Information	v = 0
	o = ana 2890844526 2890842807 IN IP4 130.207.230.189
	s = ECE8000 Class
	c = IN IP4 130.207.230.189
	b = CT:256
	t = 3034423619 3038023619
Audio Flow	m = audio 49170 RTP/AVP 0
	a = rtpmap:0 PCMU/8000
Video Flow	m = video 51374 RTP/AVP 32
	a = rtpmap:32 MPV/90000

Figure 3: The body of SIP messages, in SDP.

In the SIP architecture, the exchange of SIP requests and responses defines a transaction, which can be for instance a transaction to set up a new call, or a transaction to modify an existing call. A typical SIP transaction to establish a call involves user agent clients (UACs)

and user agent servers (UASs), eventually redirect servers, and most often proxy servers. In normal operation, SIP transactions comprise a handshake of two or three messages between SIP agents or servers. Thus, the protocol is very straightforward. A few number of messages usually accomplish a certain task, as in the basic SIP call setup of Figure 4, with proxies P1 and P2.

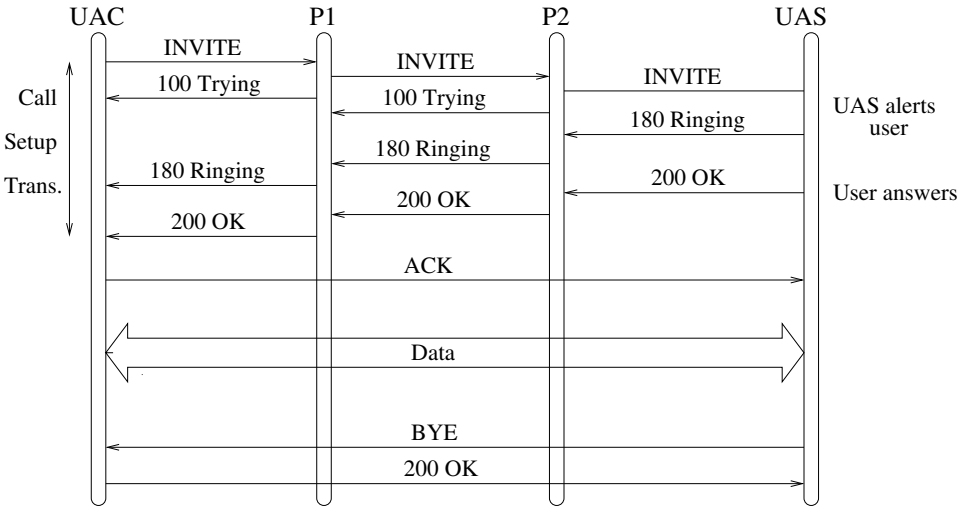


Figure 4: Basic SIP call setup signaling scheme.

Furthermore, this simple call setup flow of messages allows end users to negotiate their media capabilities and flows that will be in the multimedia session. Following an SDP-based offer/answer model for capability negotiation [89], the media capability negotiation can happen in two forms, as shown in Figure 5.

One form of offer/answer is to provide the SDP offer in the initial *invite* request, and

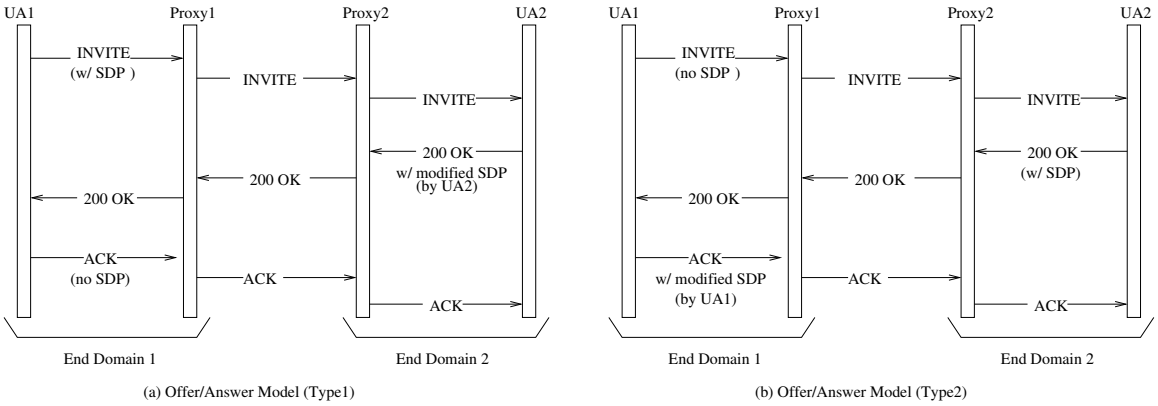


Figure 5: Offer/answer models for capability negotiation.

the SDP answer is provided in the first reliably transmitted response (i.e., a final response or a provisional response, given that it is properly acknowledged [90]). The second form is to send the initial SDP offer in the response message for an empty-bodied *invite* request, and the SDP answer comes in the final acknowledgement message.

Although user agents can send signaling messages directly to each other, it is important to note that in medium to large networks, SIP proxies always participate in the SIP call control transactions [29, 42]. For instance, SIP proxy servers perform the role of authentication and authorization since UAs must be authenticated and authorized for every call or session request they make. Moreover, SIP proxies act as application-layer routers that are responsible for routing the SIP requests to other proxies towards their final destination.

2.3 Interaction of Application-Layer Signaling and Resource Management

As SIP provides call control services to real-time conversational applications, resource management schemes can protect network resources for the call. The need for coordination or interaction between session control signaling and resource management in the Internet was first raised by Goyal *et al.* in [48]. The approach taken to make this coordination happen is a resource-reservation-based new signaling architecture - the Distributed Open Signaling Architecture (DOSA), which defines a two-phase call setup scheme, where resources are first reserved, the called party is alerted, and then the resources are committed to the users. In addition, DOSA architecture defines the role of "gates" and "gate controllers". Gates represent the network-layer (e.g., edge routers) while gate controllers represent the application-layer (e.g., an application server such as a proxy server). When users initiate signaling to set up a session, the gate controller performs service-specific admission control, verifies if the network can admit the call prior to the resource reservation phase, and sets up the session policy information at the network-layer (i.e., at the gates).

The IETF has addressed the interaction of session control signaling and resource management in RFC3312 [19]. The RFC3312 standard defines new ways to communicate media flow and QoS information end-to-end and to integrate the resource management phase with

the SIP call setup signaling (which will be reviewed in more detail in the next subsection). Although the resource-reservation-based approach is used and RSVP is usually referenced in its examples, the RFC3312-based signaling scheme is independent of a specific QoS signaling protocol. Other approaches for the interaction of SIP and resource management have also considered the Differentiated Services (DiffServ) QoS architecture [94].

Independently of the QoS management procedure (i.e., RSVP or DiffServ) and the type of users, the role of SIP proxies can be extended to the role of *QoS-enabled SIP proxies*. QoS-enabled SIP proxies not only support the typical session control functionality but also understand the session's QoS requirements and translate them to the network level. For instance, the interaction of QoS-enabled SIP proxies and QoS admission control is considered in RFC3313 [66]. Moreover, a more centralized SIP architecture, based on the use of SIP proxies at the end domains, is discussed in [79] as a way to provide assured services to multimedia sessions.

Even more practical examples of the interaction of session control signaling and resource management and the extended role of application-layer servers can be found in the QoS frameworks of the Universal Mobile Telecommunications Systems (UMTS) [4] and PacketCable [2]. In both architectures, the application-layer signaling triggers QoS procedures in the network layer. These procedures include policy-based admission control through the interaction of an application-layer server and policy control servers to authorize resources for the call, and resource reservation either initiated by the end user or by the application server on behalf of the end user. All these occur during the call establishment phase.

Especially, the UMTS QoS architecture adopts SIP as the session control signaling protocol. At the transport-level, the UMTS gateway to external networks is the GPRS gateway support node (GGSN). It is a DiffServ-capable edge router that embeds the functions of a policy enforcement point (PEP) [118, 119]. At the application-level, the UMTS QoS framework defines an *application function* that exchanges with the GGSN service-based policy set up information during session establishment. An example of the *application function* is the Proxy Call State Control Function (P-CSCF) in the IP multimedia system (IMS) (which serves IP multimedia sessions over the UMTS domain). In its role to obtain an

authorization token for the session, the *application function* behaves in a similar way to a QoS-enabled SIP proxy. If authorization is granted, resources are reserved following the coordination of SIP and resource reservation (RFC3312), as described next.

2.4 Overview of the Integration of SIP and Resource Management

In order to address in more detail the current work on the integration of SIP and resource management, including security mechanisms to protect the signaling messages, an overview of the different phases of a SIP call setup procedure integrated with resource management is presented. To illustrate these different phases in the call setup process of a QoS-enabled multimedia session, Figure 6 shows the authentication, call authorization, media authorization, and resource reservation phases that have been currently proposed for SIP by the IETF.

2.4.1 Call Authentication

In the SIP architecture, user agents (UAs) must be authenticated and authorized for every call or session request they make. For security regarding user-to-user communication or end user-to-proxy communication, usually SIP proxy servers perform the role of authentication and authorization. However, for hop-by-hop security, such as proxy-to-proxy authentication, SIP relies on other mechanisms such as IPSec which adds security to the packet at the network layer, and the Transport Layer Security (TLS) [32, 95] protocol which is used with connection-oriented transport protocols (e.g., TCP).

For authentication, SIP uses a digest authentication mechanism [92] derived from HTTP digest authentication [39]. It is a challenge-based mechanism in which the proxy or user that receives the call request challenges the user who sent the request. The user then issues a new request with the appropriate response (e.g., a function of the challenge, the realm, username, and password). After its identity is confirmed, the proxy verifies account information, and if the user is authorized to make the call, the proxy forwards the request.

For end-to-end authentication, user agents at the destination domain can use a similar challenge-based mechanism to authenticate the caller. However, the issue of transferring

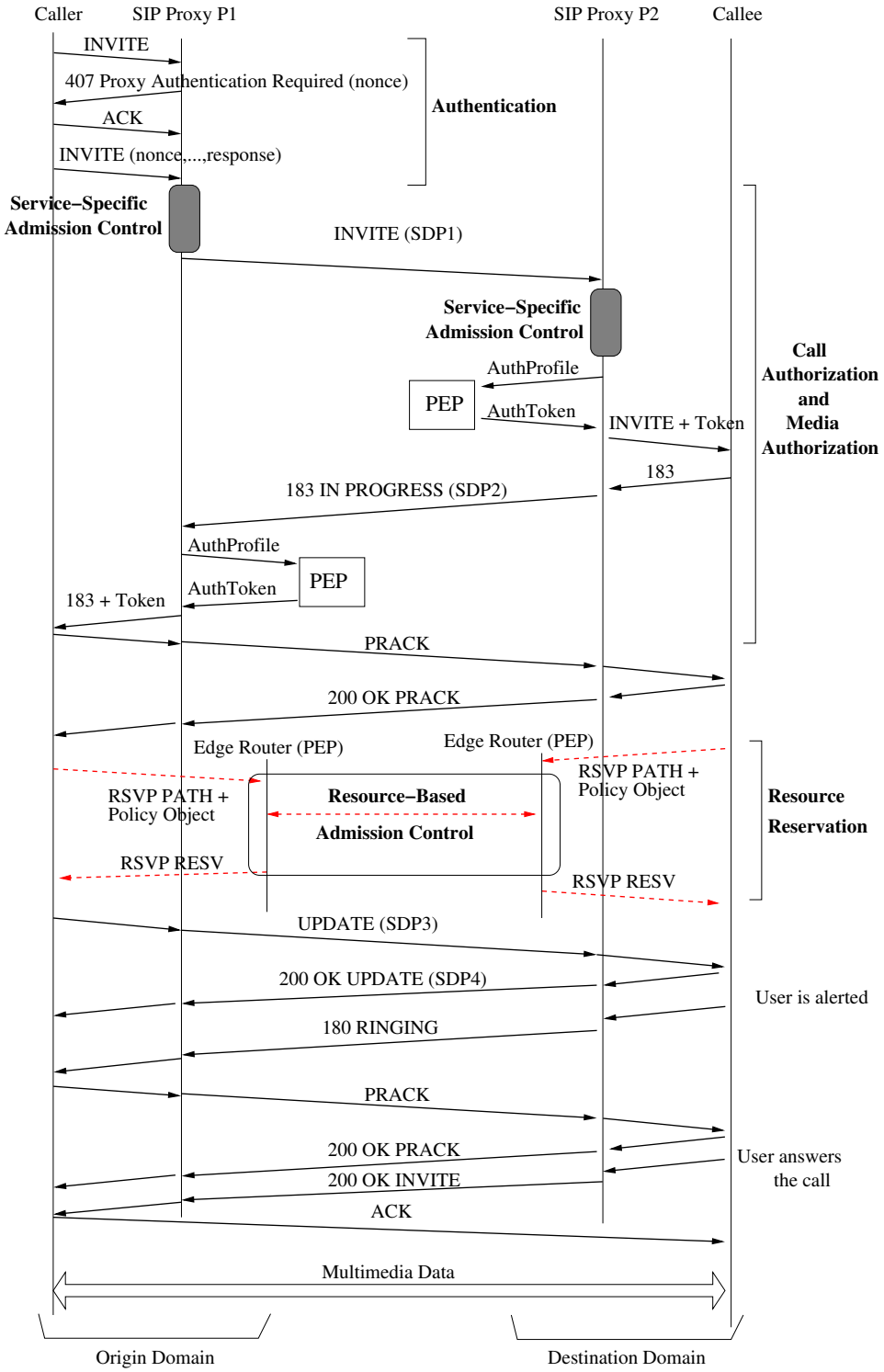


Figure 6: Phases of a SIP call setup signaling integrated with resource management.

authenticated user's identity across different domain has been the subject of different proposals.

First, in RFC3325 [58] the authenticated identity information is sent in a new SIP header (the *From* header may have an anonymous identity), which is called *P-Identity* header (the prefix P meaning that this is a private header). However, this information is transferred with no added security since it assumes a network of trusted servers. A second solution is proposed in an IETF draft [78] in which an authentication service (usually a proxy server) adds a signature in the header of the SIP message. This signature is obtained by combining the user's identity information, and information in other headers, in addition to the information in the SDP body of the messages. When the destination domain receives a message with this signature, it can verify its authenticity based on the origin domain's certificate, and it can assume that the user has been authenticated and that the message has been safely transmitted. However, this solution requires added complexity in the SIP proxies and its operation has some restrictions on the way proxies redirect requests.

A variation of the approach of sending authenticated identity or a signature information in the header of SIP messages is to send sensitive information in the body of SIP messages using encryption mechanisms. An example of such approach is adopted in the secure web-conferencing model of [101], which uses MIME bodies to send identity information encoded in XML. The advantage of this latter scheme is the ability to send more information about the user (i.e., more granularity in the authentication). However, the proxy would need to attach this message to the body of the message and either send it to the network or send it back to the UA to re-issue a new request with this information. The first way is not recommended since proxies are not allowed to modify the content of messages. The second way requires a new transaction between proxies and UA. This could be done by using a back-to-back user agent (B2BUA) [79] instead of a simple proxy; however, additional transactions between SIP proxy and UA would be needed, which may lead to a very long call setup signaling.

2.4.2 Call Authorization

For call authorization, no authorization system is proposed in the basic standard for SIP (RFC3261). In [95], this process is considered completely transparent to the SIP procedure. However, the application-layer signaling must ensure some sort of service-specific admission control [48]. This means for instance to block some requests depending on the target user or on the volume of call requests, and to give priority to certain call requests (emergency calls, priority users). It is true that the decision on whether to authorize the call is ultimately made by the receiver of the call invitation, at the destination domain. But proxies at both the origin and destination domains also have call authorization responsibilities when deciding whether to forward the call requests or not.

2.4.3 Media Authorization

The role of proxy authorization has been linked to media authorization in an informational IETF standard (RFC3313) [66]. According to RFC3313, it is assumed that users must present an authorization token to reserve resources in the network, using RSVP. It also assumes that SIP proxies have access to the content of the SIP messages (thus, messages cannot be encrypted). Based on these assumptions, it proposes the interaction of SIP proxies and local policy enforcement points (PEPs) at both the origin and destination domains of a call. If the PEP authorizes the media flows the user has requested, it returns tokens to the proxy, which sends them to the UA. The proxy uses a header in the SIP message - *P-Media-Authorization* - to send this information to the UA. Then, the UA uses the token to reserve resources in the network (e.g., by converting the token information to policy elements in an RSVP message [55]). Given this token, the network verifies if the user is authorized to reserve the requested resources, and if the amount of resources matches the allowable amount. Then, the network admits the call based on the availability of resources (resource-based admission control).

It is interesting to note the similarities of the concepts of QoS-enabled SIP proxies and the idea of gate controllers [48] previously discussed. QoS-enabled SIP proxies interact with PEPs in a similar way as gate controllers interact with gates to authorize the service and

set up the network for the session.

After call authentication and call authorization, including media authorization, user agents take control of the call setup procedure, and request resources from the network using the authorization token. Only after resources are reserved, the receiver of the call invitation is alerted, and answers the call. Finally, QoS-enabled media exchange begins. The integration of SIP call setup signaling and resource reservation signaling based on RFC3312 [19] is described next.

2.4.4 SIP and Resource Reservation (RFC3312)

- *Communicating Media Flow and QoS Requirements at the Application Layer*

SIP messages in an offer/answer process communicate and negotiate media flow information end-to-end. In QoS-enabled networks, information about the media streams of a multimedia session allow users to invoke their resource reservation schemes and map the flows to resource reservation flows. Per-flow or per-session reservation may be used, and ways of mapping media streams to resource reservation flows is presented in [20]. Information about media flows is particularly important when the user being called needs to be involved in the resource reservation. For instance, conversational multimedia sessions usually have two-way flows and the receiver of the call will only know what resources to reserve through the application-layer messages of a SIP call setup. Thus, at the application-level, SIP provides media information that is communicated end-to-end and can be used to bi-directionally reserve resources in QoS-enabled networks.

In addition to the information concerning flow identification, bandwidth, and media types, it is clear that more information about the media flows and their QoS requirements would be helpful to the application to reserve resources. Also, users need to communicate the status of the reservation end-to-end (i.e., in order to update each other on the results of the reservations triggered at both ends of a call). Hence, the IETF proposes the concept of "preconditions" in RFC3312. It defines additional attributes to the media flows with respect to QoS (direction, strength, status type) [19]. These attributes are used to communicate the desired and the current QoS reservation status. Moreover, the preconditions concept has

the purpose to hold the call setup transaction until all preconditions are met (i.e., desired status is equal to current status). Then, the called user is alerted of the incoming call.

- *Signaling Flow*

In RFC3312's interaction of SIP and QoS, at the application-level UAs exchange SIP messages to establish a new session, and at the network-level UAs send RSVP PATH messages to trigger resource reservation throughout a QoS-enabled network. In its signaling flow (Figure 6), we have identified five important steps in the interaction of SIP and resource management:

1. Perform initial offer/answer
2. Reserve resources
3. Update users on the results of resource reservation (offer/answer)
4. Resume call setup (alert user)
5. Exchange QoS-enabled media

The first step includes a basic offer/answer SDP negotiation, with QoS preconditions included in the SDP content of the messages. In the example shown in Figure 6, the offer (e.g., SDP1) is sent in the initial *invite* request, and the answer (e.g., SDP2) is sent in the response. Note that in this step the receiver of the call invitation is not alerted of the incoming call. The reservation phase (step 2) must be successful and its results updated to end users in a new offer/answer exchange which includes SDP3 and SDP4 (step 3) before the callee is notified of the call (step 4). Finally, when call setup resumes and the callee answers the call, QoS-enabled media exchange begins (step 5).

The signaling flow proposed in RFC3312 achieves the interaction of SIP and resource management smoothly, through the use of preconditions in the SIP messages to hold the call setup transaction until resources are reserved for the call.

2.5 End-to-End QoS in Heterogeneous Networks

In a network with heterogeneous domains, each with their own policies, addressing schemes, and QoS mechanisms, the issue of end-to-end QoS has growing importance. The work in [102] addresses this issue (in addition to the problems of firewall traversing and address translation) in the proposal of a generic control framework, which is also standardized by the European Telecommunication Standards Institute (ETSI) in its Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) project [35].

Since the heterogeneity occurs mostly at the transport domain (which is the part of the network that actually handles the data packets, i.e., the transport and network layers), the framework uses a model in which the application layer (represented as the service domains in Figure 7) controls the transport domains. This control architecture translates end-to-end requirements to transport domain semantics for QoS, addressing, billing, and accounting purposes.

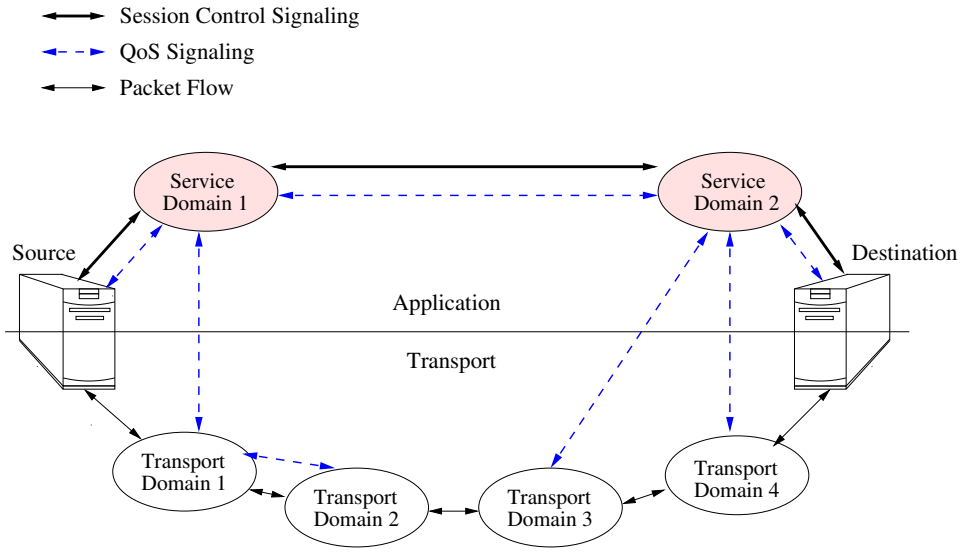


Figure 7: Application-level control of transport domains.

In addition, the authors specify in the framework interfaces between application level and data transport level, and suggest the implementation with current signaling protocols (e.g., SIP, RSVP). Further work is required to adapt existing protocols to this new architecture, which adds a certain level of complexity to the signaling needed to establish a session.

The interfaces between application and transport levels show that there is a high degree of signaling interaction between session control and QoS schemes. Moreover, QoS is fully integrated at the application level since the QoS requirements are handled at this upper layer. Concerning the implementation of this framework using existing Internet protocols, it is necessary that the service domain have knowledge of the data path. In this case, application layer entities (such as a SIP proxy server) should be aware of the data path. In the Internet this is usually true at the access domains.

Considering the heterogeneity of the transport domains, the access networks may be as well wireless access networks that can generate dynamic changes for an on-going session (e.g., changes in user location, abrupt changes in the availability of network resources). Thus, adding some type of control at the application-layer has the potential to harmonize this heterogeneous network environment.

2.6 Conclusion of Literature Review

After reviewing state-of-the-art works on the interaction of application-layer and resource management, the conclusion is that they mainly addressed the addition of a resource reservation procedure during the call setup phase. Although different levels of integration between session control and QoS have been proposed, current studies are mostly influenced by the resource-reservation-based QoS mechanism. Therefore, they address more strict-type call setup schemes in order to provide assured services, as in traditional telephone networks. Furthermore, those studies lack either performance analysis or experimental results on the proposed schemes.

More specifically, looking from the perspective of the proxy-user interaction, Figure 8 identifies three types of interaction that have been addressed in previous works. First, in a SIP call setup integrated with resource management, as proposed in RFC3312, SIP's architecture allows users to reserve resources directly using their own resource management protocols. This is similar to the procedure in the generic DOSA signaling architecture. On the other hand, in heterogeneous networks there are users who may not be able to request resources directly. They may take advantage of the interaction of QoS-enabled SIP proxies

where the proxy server requests resources on their behalf and thus controls the reservation process. An example of work in this area is the signaling architecture proposed by Salsano *el al.* [94] where a proposal for a new Common Open Policy Service (COPS) [34] client type has been applied in the interaction of proxy servers and DiffServ entities.

The third area that has addressed the role of QoS-enabled SIP proxies is on admission control in policy-based QoS architectures, such as the idea of gate controllers in the pioneer work of DOSA architecture, and the RFC3313 proposal for the interaction of SIP proxies and QoS admission control for obtaining the authorization information to reserve resources in the network. Note that this role of proxy servers applies to both the cases when users request resources directly or when proxies request resources on behalf of the users. Also, different QoS mechanisms may be applied to either case.

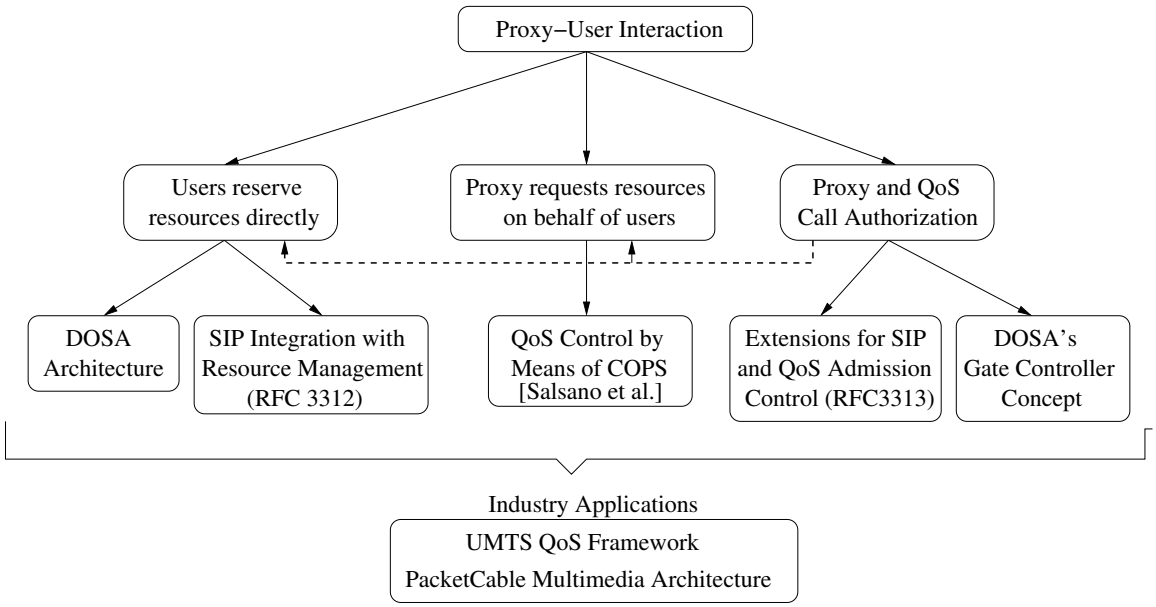


Figure 8: Overview of previous research work.

Finally, the industry's approach to the application-layer and resource management interaction presented in the frameworks of UMTS and PacketCable covers the different cases, i.e., either users requesting resources directly, or application servers requesting resources on behalf of users. In both cases, the role of QoS-enabled application services in the call authorization process is adopted.

CHAPTER III

A LIGHTWEIGHT CALL SETUP SCHEME USING QOS-ENHANCED SIP PROXIES

Based on SIP and the Differentiated Services architecture, the role of QoS-enhanced SIP proxies[1] is investigated in this chapter. The cooperation between resource management and SIP occurs in a lightweight scheme that can provide coarse-grain QoS guarantees to users who may not have direct access to QoS reservation protocols.

By implementing the proposed scheme in a testbed, performance results show improvements in the quality of high-quality video applications with minimum impact to end users, especially in terms of call setup overhead.

3.1 Problem and Solution

In traditional circuit-switched telecommunication networks, session control signaling and resource management are all integrated (i.e., signaling for call control also selects the path that can provide resources for the call). In addition, resource reservation is characterized by hard-state reservations, where resources are explicitly allocated and released for a call.

In IP networks, however, the hop-by-hop principle implies no fixed route and stateless routers. As a result, services are best-effort. To provide QoS guarantees to different types of applications, two complementary approaches have been proposed: *(i)* the resource-reservation-based approach which performs admission control and reserve resources for flows or connections on an end-to-end basis, and *(ii)* the differentiated services approach which provides individual packets with different per-hop behaviors at a given node depending on the packets' service markings. Given these QoS schemes, the fact is that they are totally independent from session control signaling. Hence, in the IP world there is no integration

[1]The concepts of QoS-enabled SIP proxies and QoS-enhanced SIP proxies are similar. But this work introduced the term QoS-enhanced SIP proxies prior to the definition of QoS-enabled SIP proxies as described in Chapter 2.

between session control and QoS signaling.

This separation between session control and QoS signaling is in agreement with the Internet architecture, and the independence from the network and application layers allowed the separate development of protocols and frameworks for session control and QoS signaling. Thus, it suits most of the applications in the Internet, such as traditional Web and email traffic. However, this lack of integration between session control and QoS signaling becomes an issue for Internet multimedia applications which offer real-time conversational services. Such applications combine requirements of traditional telephony applications and Internet applications. Therefore, there is a need for some level of interaction between session control and resource management, something in between the totally integrated model of telecommunication networks and the decentralized model of IP networks.

As discussed in Chapter 2, to solve this problem there have been several different approaches proposed in the literature. However, their approaches address a telephony-style call setup model to provide assured services for interactive multimedia applications over the Internet. In this chapter, the approach taken is to provide loose QoS guarantees to the applications, through the interaction of QoS-enhanced SIP proxies and Differentiated Services edge routers at the access network (Figure 9). The target users are access devices that have limited capacity and may not support resource management mechanisms.

In [94], the interaction of SIP proxy servers and Differentiated Services has also been addressed. This interaction occurred through a modified version of the COPS protocol to request resources. However, in order to guarantee the compatibility among network equipment and protocols, an architecture that uses a SIP-based interface for the interaction of proxies and DiffServ routers is here adopted. The proposed SIP-based architecture to communicate QoS related information has the advantage of being independent of a specific QoS signaling protocol. In addition, it is based on a lightweight communication scheme, which is independent of a centralized QoS entity such as a bandwidth broker [23, 73].

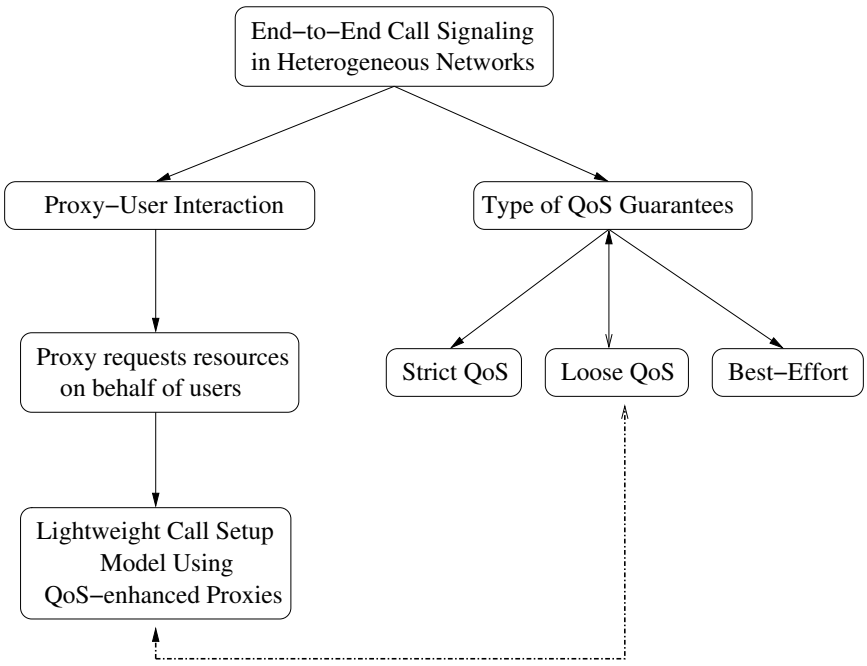


Figure 9: Lightweight call setup scheme's approach.

3.2 Architecture with QoS-Enhanced SIP Proxies

As shown in Figure 10, the application-level element that works as the liaison between end users and QoS services is a stateful SIP proxy, i.e., proxies that can save the state of a transaction. Concerning the state of the multimedia session, stateful proxies can request through the *record-route* functionality [92] that user agents route the SIP messages through them for all mid-call transactions. Thus, the stateful proxy can trigger resource allocation depending on the context of the SIP messages received and the status of the current SIP transaction (including the results of the capability negotiation process).

The stateful proxy is a *QoS-enhanced SIP proxy*, which not only supports the typical session control functionality but also understands the users' QoS requirements and provides an interface with the network routers. According to [102], this interface between application (SIP proxy) and network (router) layers communicates transport-related characteristics of the user flows. Basically, it is a one-way interaction between SIP proxies and the network layer (Figure 11).

The interface between the QoS-enhanced SIP proxy and DiffServ-capable edge routers occurs through the exchange of SIP messages. For this purpose, the edge routers must

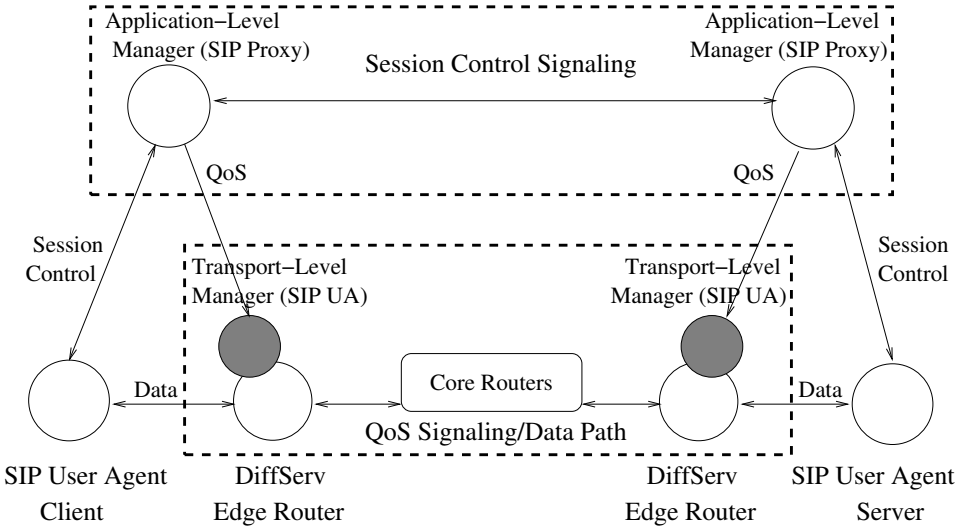


Figure 10: Basic architecture with QoS-enhanced SIP proxies.

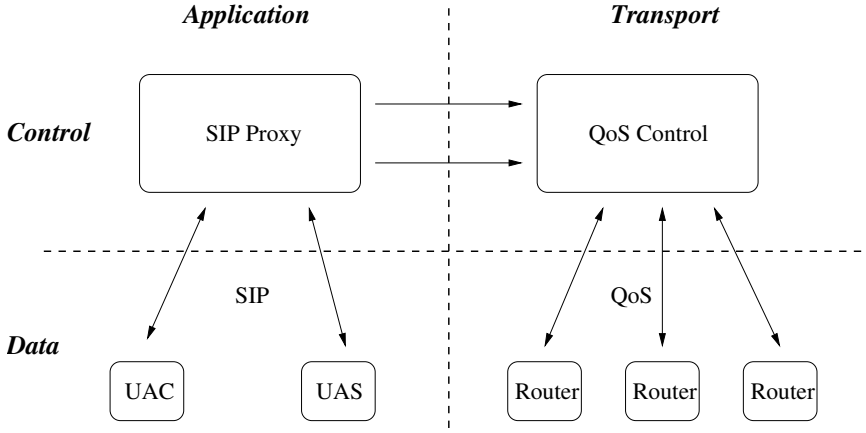


Figure 11: One-way SIP and resource management interaction.

implement a basic SIP user agent in order to communicate with the proxy. Due to the simplicity of the SIP protocol, and since these services are used on a per-session basis, this interface has minimum impact in the packet forwarding process of the routers.

In this one-way SIP and resource management interaction, the information from the QoS-enhanced SIP proxy is used by the router for these purposes: *(i)* to classify packets by updating the filters that perform multi-field classification, *(ii)* to meter the traffic according to the bandwidth indicated in the session and manage exceeding traffic, *(iii)* to mark packets to receive appropriate per-hop behavior at the backbone. For simplicity, it is assumed that the network can provide the required bandwidth, and the edge routers are configured with

classes with adequate capacity. The edge routers can then classify and enqueue the packets of the new flow according to the QoS requirements transmitted by the proxy. Additionally, the routers do not need to keep state for the SIP session. We assume that the filters and meters are periodically reset back to the basic router configuration.

- *Call Setup Signaling Flow*

The signaling messages exchanged between source and destination end domains to set up a multimedia call are shown in Figure 12. This scheme assumes that the edge router of each domain has previously subscribed with the proxy server (1). The SUBSCRIBE method [84] opens a communication channel in which the router can be remotely notified by the proxy of the incoming media flows and their QoS requirements. Next, to place a call the UAC sends an INVITE message (2) to the other user. In the basic SIP protocol, without any additions, this INVITE message with an SDP body naturally includes the following information:

- the media level attributes of the types of flows that will be exchanged in the session, such as the media types (audio, video) and the transport port to which the media flows will be sent;
- the required bandwidth for the session;
- end user information, represented in SIP URLs that are independent of the physical location of the user.

In addition, the proposed call setup model adds a parameter (*class*) that indicates the type of traffic in terms of QoS requirements, for instance, if the traffic requires low latency or high bandwidth.

The call setup signaling proceeds as normal (3), and when the final response from the called user, with the final parameters of capability negotiation, is issued (4), the proxies at the end domains notify the router of the QoS requirements (5). The SIP messages

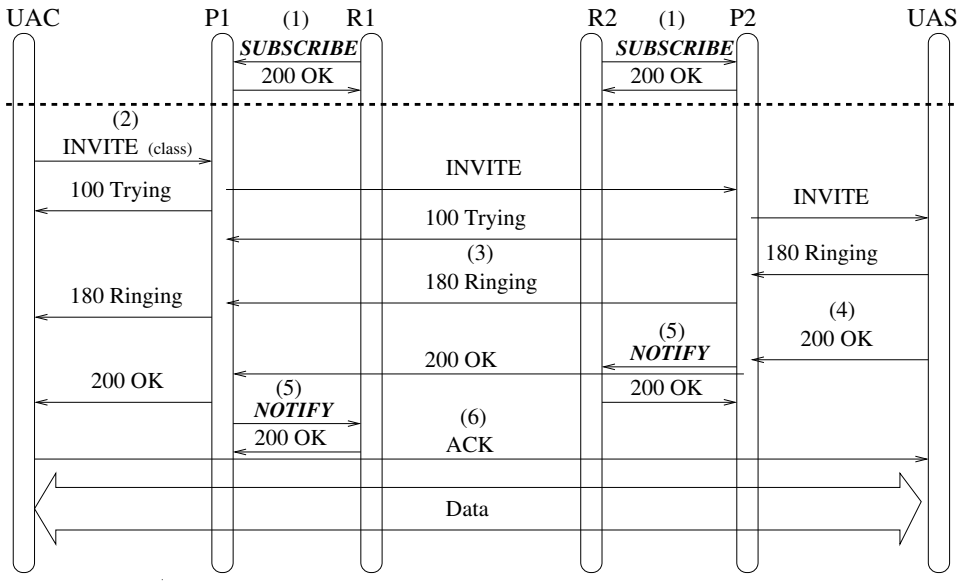


Figure 12: SIP call setup signaling with interaction with DiffServ routers.

that can be used are the NOTIFY [84] requests which allow agents to send notification of asynchronous events to remote nodes. Finally, the three-way handshake of the call setup is completed through the final acknowledgement (6) from the caller, and then the data streams are transmitted.

In summary, the proposed architecture keeps the session control signaling and QoS signaling independent. At the end domains, their interaction is provided by the QoS-enhanced SIP proxy which interacts with the edge router on behalf of the end users of the domain. Moreover, compared with the basic SIP call setup [92] the proposed call setup flow represents a small call setup overhead.

3.3 Testbed Experiments

3.3.1 The Testbed

A testbed (Figure 13) was implemented in order to study the interaction of session control signaling and QoS. The testbed comprises PCs running the Linux operating system for the SIP entities and network routers. Also, it has a bottleneck architecture, where the main backbone allows only 10 Mbps of bandwidth, being the end users connected to 100 Mbps Ethernet hubs.

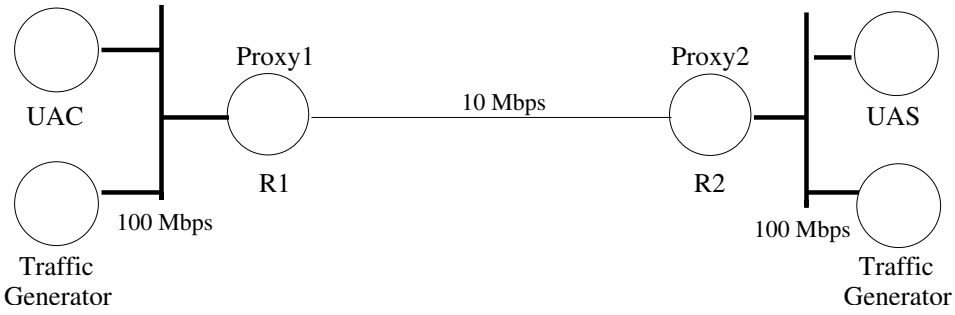


Figure 13: Testbed with Linux routers in a bottleneck topology.

At the access networks, SIP user agents and proxies implement a basic version of the SIP protocol (Section 3.3.2). In the experiments, a SIP user agent client initiates a call setup with the SIP user agent server on the other end. After call setup signaling is completed, data transmission starts between them, simulating the transmission of multimedia content. This traffic competes with background traffic generated by the *Smartbits* traffic generator. The SIP proxies in each domain are implemented in the same PCs as the routers, and they have the added functionality of communicating the QoS requirements to the router (QoS-enhanced proxies). In this implementation, the proxy issues the traffic control commands directly to the router.

The router's software consists of the Linux 2.4.7 kernel compiled with support for IP forwarding. The routers are configured using the traffic control engine, which is part of the *iproute2* package, with Differentiated Services support [5, 1]. It has methods that can classify, prioritize, share, and limit inbound and outbound traffic.

3.3.2 Protocol Implementation

One of the advantages of SIP, especially in comparison with the H.323 protocol suite, is the accessible implementation of the protocol and related services. As shown in Figure 14, client-server socket programming [25, 103] is used to implement SIP and its interface with the UDP transport protocol. Two groups of programs are defined: the core stack layer and the transaction user layer. The core stack layer carries out the interface with the network kernel, translates SIP text messages to objects [41], and implements the client and server transaction state machines (which are used by both proxies and user agents). On

top of the core stack layer, the transaction user layer implements the core programs for the user agents and proxies. Among them, the call state machines at the user agents, and the message routing and forwarding functions at the proxy. Additionally, the implementation of a QoS-enhanced SIP proxy includes a QoS management functionality, which captures the QoS requirements of SIP transactions and notifies the router of the session's media flows.

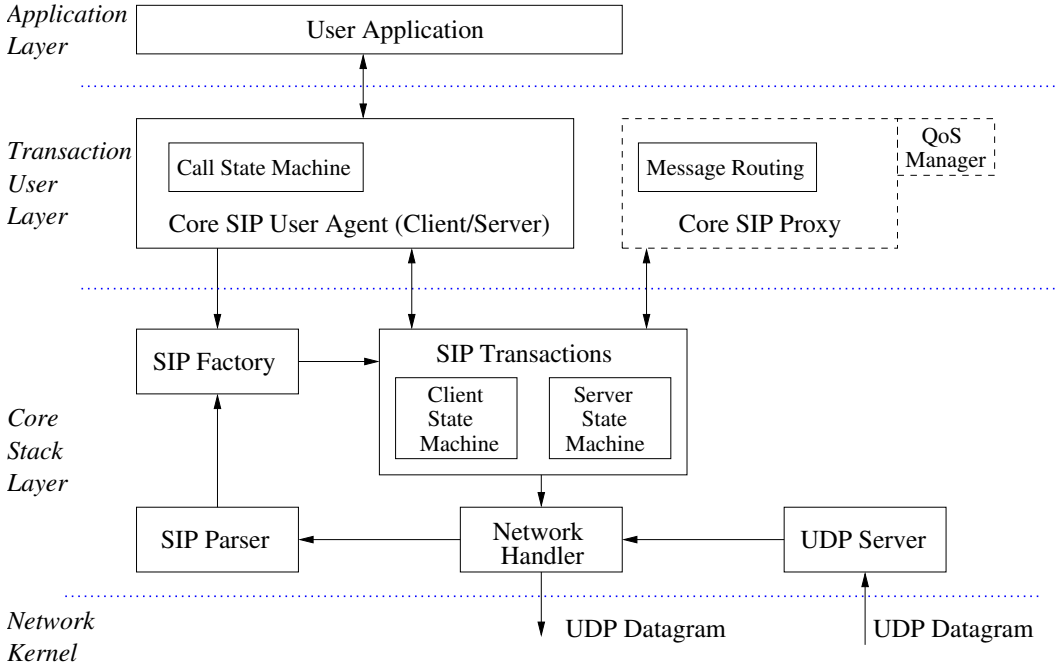


Figure 14: SIP protocol implementation.

3.3.3 Experiments Configuration

In the experiments, after the call setup is completed, the user agents exchange video traffic. It is assumed that the video traffic is of a class that requires very high quality and does not tolerate excessive packet losses. The main objective is to measure the quality of the video perceived by the end user, hence the focus is on packet losses. Because it is not possible to measure the quality of the multimedia content perceived by end users in the testbed experiments, a simple statistical analysis of the packet flow is performed. It measures the number of consecutive packets successfully received which can indicate good quality of the reception. In other words, the metric used in the experiments is *back-to-back received packets*, which gives the average, variance, and probability distribution function (pdf) of

the number of packets that were delivered and played back on time, with no packet drops in between [9].

On the routers' configuration, the differentiated services marking (*dsmark*) queuing discipline is used to mark the packets, and class-based queuing (CBQ) is used to manage the bandwidth. It is configured with three traffic classes: an expedite forwarding (EF) class with bandwidth of 1.5 Mbps, an assured forwarding class (AF) with bandwidth of 5 Mbps, and a best effort (BE) class with the remaining bandwidth. The EF class has the highest priority, and its small FIFO queue (5 packets long) is served as long as it has packets. The AF class traffic is directed to a random early detection (RED) queue. Filters are used to classify the packets to the appropriate class, while non-matching traffic is treated as best-effort. In these experiments, filters are configured to classify the flows according to the destination port. Also, meters are attached to the filters in order to allow the router to react to exceeding traffic.

In sum, this configuration of experiments is used to allow the high quality video traffic to be forwarded as expedite forward; hence, having priority over other types of traffic. The router's filters and meters are configured based on the information provided by the QoS-enhanced SIP proxy.

Two sets of experiments were performed:

- In the first set, video traffic is sent at a constant bit rate, at 128 kbps. Then, two measurements are performed. First, statistics are collected of back-to-back received packets (or burst lengths) for video traffic that is transmitted as AF (regular service) in a congested network (i.e., background traffic is in the range of 5 Mbps, which is the capacity of the AF class). Then, statistics are collected of back-to-back received packets for video traffic that is transmitted as EF (premium service), in the same congested network. In this last situation, the QoS-enhanced proxy communicates the requirements of this high-quality video stream to the DiffServ edge router. They indicate a required bandwidth of 128 kbps (committed rate) and that packet losses should be minimized.

- In the second set, the rate of video traffic is 128kbps, but it may vary to 156 kbps during the transmission. The purpose of this set of experiments is to test the shaping effects on the quality of the video transmission. As in the previous experiments, the video traffic is first transmitted as AF (regular service) in a congested network, then as EF (premium service) after the interaction of the QoS-enhanced SIP proxy and routers.

3.3.4 Results

For the first set of experiments, the percentage of lost/out-of-sequence packets, and the mean and variance of the size of the received video streams are presented in Table 1. The results show that video traffic suffers no packet losses when transmitted as EF, in the committed transmit rate. Also, the perceived quality improves considerably since the mean size of the streams is just 11 packets as regular service. To better illustrate the improvement in the perceived quality by end users, the pdf of the back-to-back received packets are shown in Figures 15 and 16, for regular and premium service respectively. Note that dropped or out-of-sequence packets result in smaller burst length values of back-to-back received packets, which means less quality perceived by the end user. In the first histogram, video traffic receives regular service in a congested network and most of the size of the back-to-back received packets are small. On the other hand, no packet losses in the premium service is shown as one long stream of the size of the total transmission, i.e, in the range of 10000 packets. (The plot groups all bursts longer than 200 packets in the x-axis value of 200.) The occurrence of longer, uninterrupted streams, as shown in the second plot, provides a better quality to the end user.

In the second set of experiments, the number of packet losses increases and the average size of the stream is 24 packets for premium service (Table 2). Note that the large variance in this case shows that there are streams that can be quite longer than the average, which happens when incoming traffic is within the committed rate. However, the lower average size of the streams in comparison with the first experiment reflects the effects of shaping when the traffic rate exceeds the committed rate. The back-to-back received packet distributions

Table 1: Packet statistics (experiment 1).

Service	Losses	Mean Burst Size	Var Burst Size
Regular	8.9%	11.1 pkts	19.7 pkts
Premium	0%	10000 pkts	0 pkts

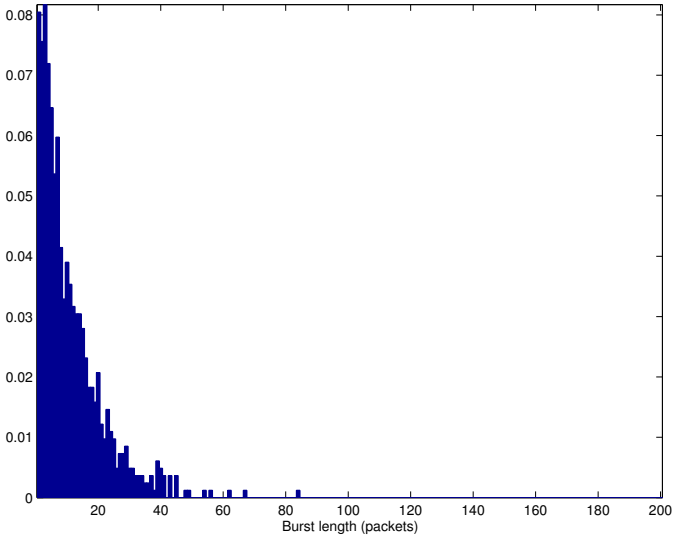


Figure 15: Back-to-back packets distribution, with regular service (experiment 1).

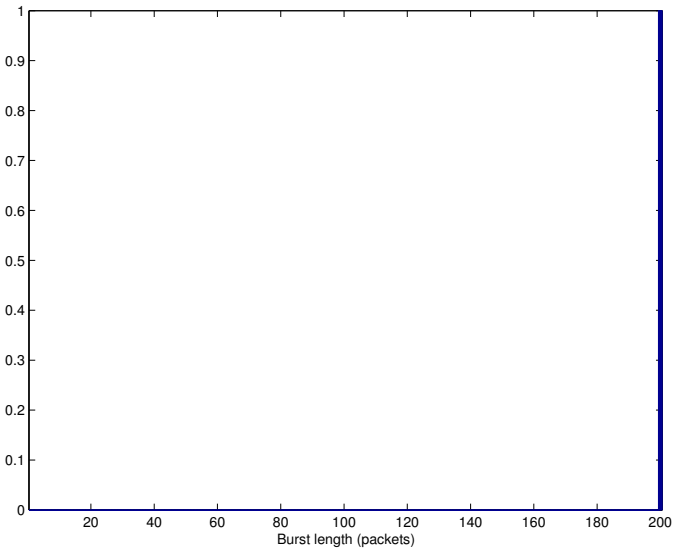


Figure 16: Back-to-back packets distribution, with premium service (experiment 1).

are given in Figures 17 and 18 for regular and premium services. In the histograms it can be seen that when the end user exceeds the committed rate, as in Figure 18, the number of out-of-sequence packets increases. Thus, the histogram shows a higher frequency of small streams, with packet drops in between. The reason is the shaping done by the network diverts the exceeding traffic to the regular service queue, which does not offer the same on-time delay guarantees as the premium service queue. Therefore, packets may arrive at the receiving end out of sequence and the quality perceived by the end user will be affected.

3.4 Conclusion

This chapter introduced a signaling architecture with QoS-enhanced SIP proxies. In this architecture, QoS-enhanced SIP proxies interact with DiffServ-capable edge routers in a lightweight scheme that can provide coarse-grain QoS guarantees to users who may not have direct access to QoS reservation protocols. In addition, the proposed lightweight call setup scheme uses a new approach (as opposed to the traditional telephony approach of assured services) to the interaction of session control signaling and resource management.

Based on experimental comparisons obtained in a multimedia call with high-quality video streams, the proposed approach improves the quality of the received media perceived by end users, with minimum impact in terms of additional signaling load or call setup delays.

Table 2: Packet statistics (experiment 2).

Service	Losses	Mean Burst Size	Var Burst Size
Regular	10.3%	9.7 pkts	16.7 pkts
Premium	6.8%	24.4 pkts	4154.3 pkts

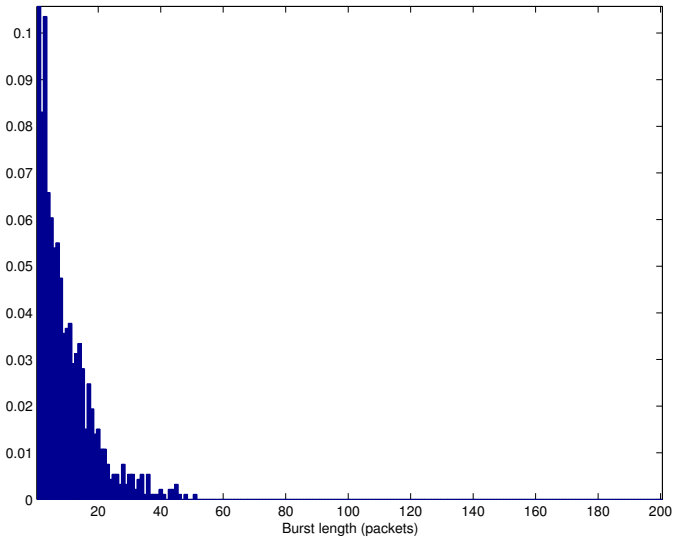


Figure 17: Back-to-back packets distribution, with regular service (experiment 2).

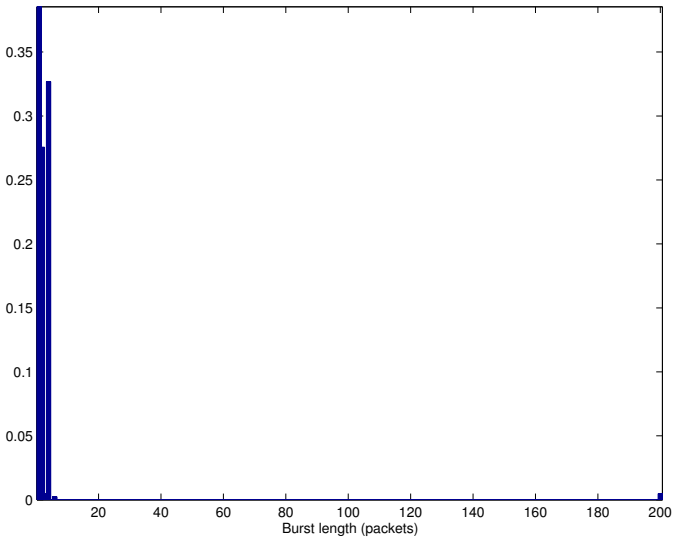


Figure 18: Back-to-back packets distribution, with premium service and shaping (experiment 2).

CHAPTER IV

ON OVERLAPPING RESOURCE MANAGEMENT AND CALL SETUP SIGNALING: A NEW SIGNALING APPROACH FOR INTERNET MULTIMEDIA APPLICATIONS

In order to provide users with a practical and fast way to reserve resources during session setup, this chapter proposes a different call setup model than that of traditional telephony. The new model allows users to be alerted while the resource reservation takes place, thus, taking advantage of parallel user answering delays and reservation delays. For this to occur efficiently, existing elements of the SIP architecture provide means for independent call setup and resource management transactions and flow negotiation. Experimental comparisons obtained in a multimedia call demonstrate that the proposed approach reduces the impact of the reservation delays, with maximum call setup delay improvements for approximate values of reservation and answering delays.

4.1 Problem and Solution

To provide quality of service control, the subject of the interaction of call signaling and resource management has addressed resource reservation during the call setup phase. Usually, the rule is to use the traditional telephony-style call setup model (i.e., users are only alerted after all resources have been successfully reserved for the call). In the Internet, however, this may result in long call setup delays and a high call blocking rate ("no QoS/no call"). For instance, the works on the DOSA architecture [48] and the IETF's proposal for the integration of SIP and resource management (RFC3312) [19] adopt the traditional telephony call setup model in which the end user is notified of the incoming call request only after resources have been successfully reserved for the call.

In particular, the signaling flow proposed in RFC3312 achieves the interaction of SIP and resource management smoothly, through the use of preconditions [19] in the SIP messages to hold the call setup transaction until resources are reserved for the call. However, the additional signaling needed for this interaction and the reservation delays of the resource management phase certainly impact the delay experienced by users and the signaling load on the network during the call setup transaction. In order to evaluate this impact and also motivated by the idea of efficiently using idle times during call setup (e.g., reservation and user answering delays), this chapter proposes an experimental signaling work on the *Resource management Overlapped with Answering Delay (ROAD)* call setup architecture. The ROAD call setup architecture addresses the subject of users capable of requesting their own resources to the network, as shown in Figure 19.

The ROAD call setup model has the objective to provide more flexibility to interactive multimedia applications: first, motivated by the idea of efficiently allocating the normal waiting times in a call setup transaction integrated with resource management, existing

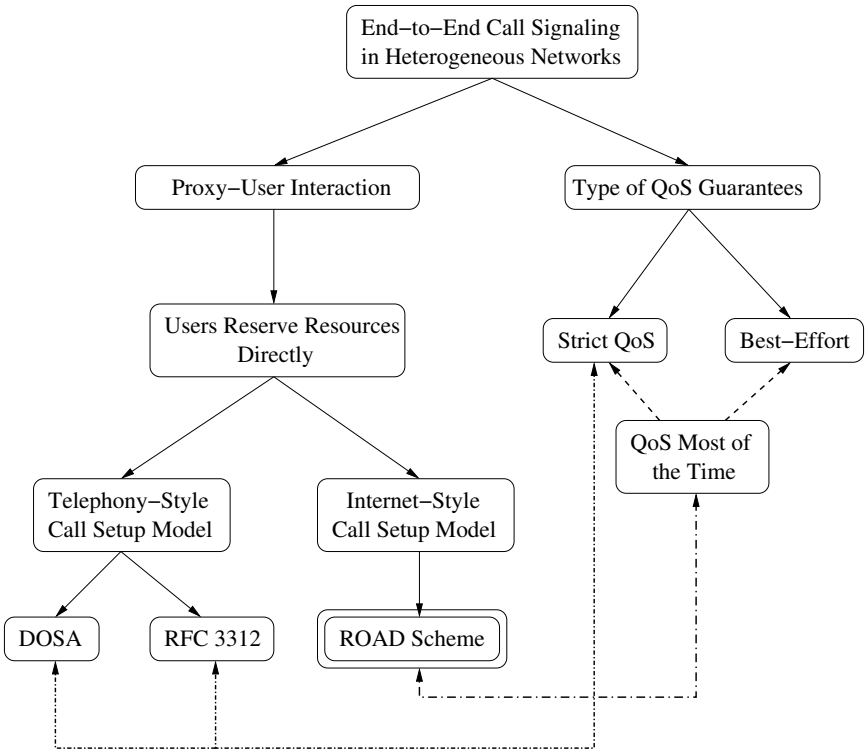


Figure 19: ROAD scheme's approach.

elements of the SIP architecture are applied to allow the call setup and resource management phases to occur concurrently. In simple words, the network is in charge to reserve resources for the call and catches-up with the call setup transaction when the reservation is completed. Second, the ROAD call setup model takes into consideration the different types of Internet applications. It includes a flow negotiation approach to adaptive and non-adaptive multimedia applications. Basically, this flow negotiation is a best-effort/QoS flow negotiation that occurs during the call setup phase. It has the advantage of avoiding the "no QoS/no call" problem to adaptive applications. As a consequence, the type of QoS guarantees it offers is strict QoS guarantees with a short period of best-effort flow at the beginning of the session. Therefore, it can be said that it offers "QoS most of the time" (Figure 19). For non-adaptive applications, the trade-offs of the proposed schemes are also considered.

4.2 ROAD - Resource management Overlapped with Answering Delay

The rationale behind the ROAD architecture is to take advantage of normal waiting times during the call setup transaction integrated with resource management. On the network level signaling, there is the delay between the resource reservation request and response. On the application level signaling, there is the delay between alerting the callee and receiving his/her response. Thus, is it possible to overlap the resource reservation delays with the user answering delays? This can potentially reduce the call setup delay, save bandwidth used for signaling, and bring light to a call setup mode that is different than the traditional telephony mode. Current architectures for SIP and resource management arrange resource reservation and user answering delays in a sequential manner. The ROAD call setup architecture proposes an implementation to these delays in a parallel manner, and experimentally evaluates its benefits.

The ROAD architecture defines a simple call setup scheme in which there is interaction between SIP UAs and QoS agents. It uses existing elements of the SIP architecture, such as SIP requests and responses and basic SIP transactions and dialogs. Thus, it can be

easily implemented in current SIP UAs that support communication of QoS information in the form of preconditions as proposed in RFC3312. Concerning the QoS signaling, it is assumed a generic QoS agent and signaling protocol. For instance, the QoS agent can be a gate controller, an edge router, a GGSN, a cable modem termination system (CMTS), or a bandwidth broker, employing their own signaling protocol to perform the resource management functionality.

Furthermore, this current implementation is based on the assumption that the application layer (e.g., a QoS-enhanced SIP proxy) has authorized the call and verified if the network can admit the new call. This usually occurs when the first call request (invite) is issued [48, 66]. In other words, an envelope is opened for the call, and if the user requests resources within that envelope, the final results of the resource reservation will most probably be successful. Based on this assumption, the ROAD signaling flow investigates independent call setup and resource management transactions. Also, it proposes the use of preconditions in a new flow negotiation approach that takes in consideration the existence of adaptive and non-adaptive applications in the Internet.

4.2.1 Signaling Flow: Independent Call Setup and Resource Management Transactions

Te ROAD call setup scheme includes the five main steps described in Chapter 2 (Section 2.4.4), but its sequence of events closely resembles steps 2 and 4 occurring in parallel. Thus, the initial call setup transaction between SIP UAs (i.e., an *invite* transaction) follows the basic SIP call setup model, in which the called user is alerted of the incoming call as soon as the *invite* request is received. The resource management transaction between call participants and QoS agents is triggered when the initial offer/answer transaction is completed. Having both transactions - invite and resource management - carried out independently, the ROAD scheme takes advantage of the normal delays of the invite transaction (i.e., user answering delays (D_{ans})) and the reservation delays (D_{res}) of the resource management transaction to reduce the impact of the interaction with resource management in the call setup signaling delay.

As a result, the ROAD signaling flow is implemented as shown in Figure 20. Looking at

this signaling flow from a higher level, the signaling flow includes two *invite* transactions. The first *invite* request initiates the call setup transaction between caller (SIP UA1) and callee (SIP UA2). This transaction, which follows a basic SIP call setup, can establish a call between SIP UAs. Then, within an active call and after the resource management transaction is completed, the second *invite* request - a re-invite - updates the session's parameters. This re-invite transaction updates users on the results of the resource management transaction. In summary, two offer/answer exchanges occur: in the first invite, the offer (SDP1) is carried in the *invite* request, and the answer (SDP2) is carried in the provisional response "ringing"; in the re-invite, the offer (SDP3), which has updated parameters of the resource reservation, is carried in the *invite* request, and the answer (SDP4), with also updated parameters, is carried in the final response "ok". After both invite transactions, QoS-enabled data exchange begins.

Within the first invite transaction, when the caller issues the initial offer QoS-enabled SIP proxies authorize the service and verify if the network can admit the call. This interaction between QoS-enabled proxies and the network layer's policy control entities may occur in different ways. For brevity, the details of this interaction have been omitted in the presented signaling flow, but the idea is that the *invite* request does not reach the callee if the service has not been properly authorized.

Then, as soon as the offer/answer exchange is completed, caller and callee trigger the resource management transaction. As it is proposed that the SIP invite transaction and the resource management transaction occur in parallel, the callee may answer the call during the resource management phase. If this happens, his/her answer is acknowledged and the session may be established before the reservation is completed. For this reason, there is a time in the ROAD call setup scheme that call participants may start transmitting the multimedia flows, even though the resource reservation process is not completed and/or the re-invite transaction has not updated users of the results of the reservation. This time is considered as a best-effort (BE) interval, since the multimedia flows may be transferred as best-effort with no QoS guarantees before the completion of the ROAD call setup. Thus, the BE interval comes as a trade-off: the simple ROAD signaling scheme overlaps reservation

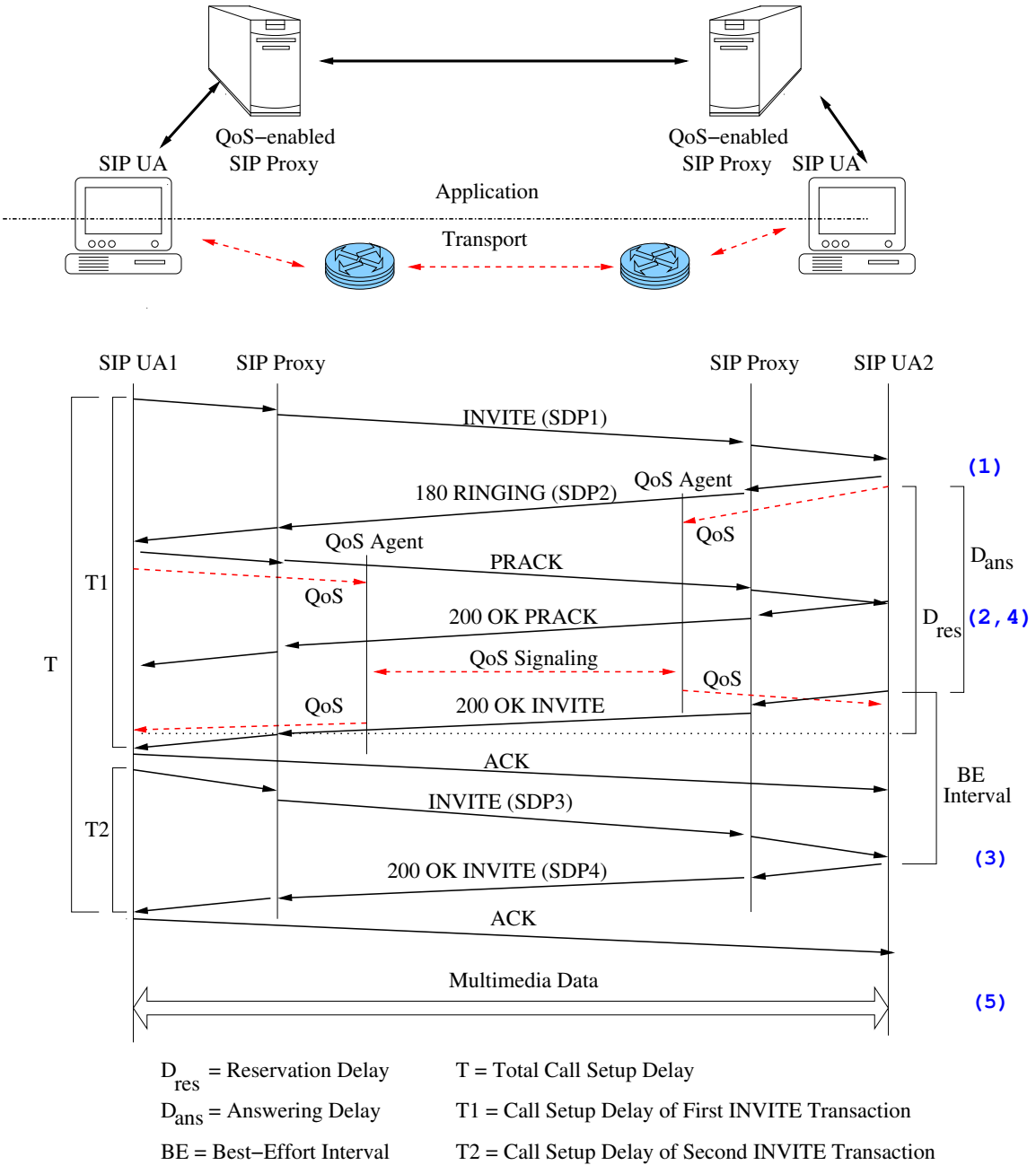


Figure 20: ROAD signaling flow.

and user answering delays at the expense of potentially less strict QoS guarantees at the beginning of the session.

4.2.2 Call Setup Delay and BE Interval

In the ROAD call setup scheme, the total call setup delay T includes the delay of the first invite transaction ($T1$) and the delay of the re-invite transaction ($T2$). In the first

invite transaction, the delay $T1$ is a sum of the overall signaling delay (d) (which includes transmission, propagation, queuing, and processing delays), and the reservation delay (D_{res}) or answering delay (D_{ans}), whichever is larger. In the re-invite transaction, the delay $T2$ is only due to the signaling messages' delays and is not affected by any reservation or answering delays. Therefore, the delays for the ROAD call setup scheme can be expressed as in Equations 1 and 2.

$$T1 = d + max(D_{res}, D_{ans}) \tag{1}$$

$$T = T1 + T2 \tag{2}$$

Now focusing only on the reservation and answering delays, we can take a look at the total call setup delay and the BE interval from two perspectives:

1. $D_{res} \leq D_{ans}$

   When the resource management transaction finishes before the initial invite transaction, users do not notice the reservation delay since it has no impact on the call setup delay:

   $$T = d + D_{ans} + T2 \tag{3}$$

   Also, the BE interval is in its lowest level, i.e., it is approximately the duration of the re-invite transaction which updates users on the results of the reservation:

   $$BE \approx T2 \tag{4}$$

   In this case, the BE interval represents the time the network needs to catch-up with the call setup transaction.

2. $D_{res} > D_{ans}$

   When the resource management phase takes longer and continues even after the callee answers the call, the reservation delay directly affects the call setup transaction:

$$T = d + D_{res} + T2 \tag{5}$$

   And the BE interval increases as the difference between reservation and answering delays becomes more evident:

$$BE \approx (D_{res} - D_{ans}) + T2 \tag{6}$$

In summary, the first case ($D_{res} \leq D_{ans}$) shows the natural advantages of the ROAD signaling scheme, with a small BE interval and no impact of the reservation delay on the user's call setup delay. In contrast, the latter case ($D_{res} > D_{ans}$) shows the greater impact of the reservation delays on the BE interval and call setup delay. Note that in both cases the total call setup delay (T) includes the BE interval (BE).

4.2.3 BE/QoS Flow Negotiation

Since the BE interval is affected by the difference between the reservation and answering delays, and being both delays variables difficult to control from the point of view of the applications, the ROAD scheme includes a flow negotiation approach in which the applications can take advantage of the simple signaling of the ROAD scheme to overcome the impact of the BE interval in the beginning of the multimedia session. This flow negotiation allows the applications to consider two options during the BE interval: (1) to hold the data transfer until the completion of the call setup, or (2) to allow the data transfer in a lower quality mode (i.e., best-effort mode) as a preliminary multimedia session during the call setup. This will depend on whether the application supports the difference in the network resources availability during the BE interval.

To allow the two alternative ways of operation, the proposed flow negotiation approach applies the flow id concept [17] which allows applications to specify different flow formats as

a unique flow. By adding extensions to the SDP content of the SIP messages following the flow-id concept, a best-effort flow and a QoS-enabled flow can be defined under a unique flow id. Thus, it is basically one flow, but that can be sent in different forms, only one active flow at a time. This concept is applied to a video flow in the SDP shown in Figure 21.

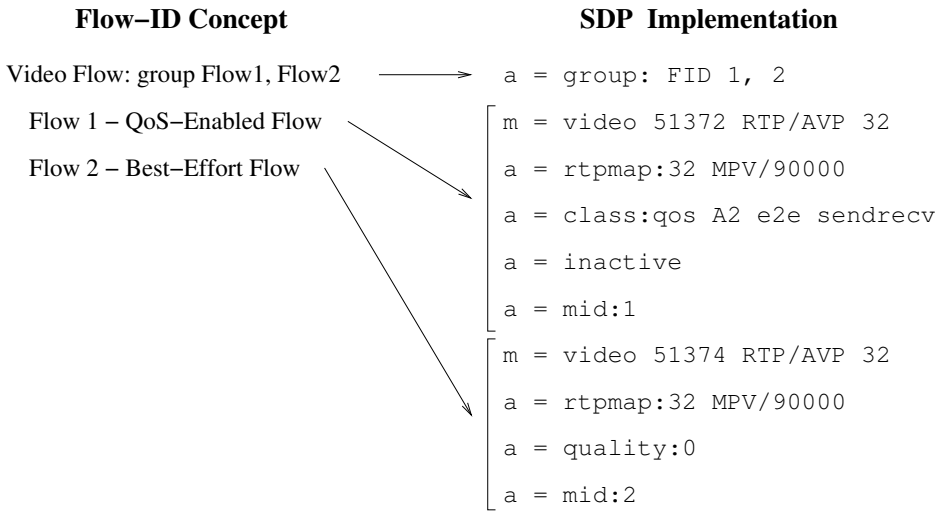


Figure 21: Best-effort and QoS-enabled flows are grouped as a unique flow.

To differentiate a BE flow from a QoS-enabled flow, there are two options: using the same codec, or using different codecs if supported by the applications. By using the same codec, an SDP flow attribute of quality can give a suggestion for the quality of the encoding. As in the video flow example of Figure 21, a "m=quality:0" attribute to the video flow tells the codec to use "the worst still image quality the codec designer thinks is still usable" [51]. The second alternative (i.e., by using different codecs) is to define the BE flow as a lower quality, lower bandwidth video flow from a different encoding type than the QoS flow.

In addition, to define the flow's required service class, the "class" media attribute line is adopted in the QoS-enabled flow's definition. Using some concepts of the RFC3312's preconditions approach, and using a classification similar to the one proposed in [22], this attribute includes information on delay/packet loss tolerance (e.g., in Figure 21, class A2 defines a real-time traffic class that allows moderate levels of packet loss), type of QoS (end-to-end), and direction of QoS-enabled flow (send/receive).

Now, from the viewpoint of the applications, some may be intelligent and adaptive in

a way that they can accommodate differences in the BE/QoS-enabled performance. Thus, they may tolerate the additional packet losses and delays that may occur at an earlier start of the multimedia session, during the BE interval. In this case, the media exchange should be allowed to start in a BE mode, to be transitioned later to a QoS-enabled media exchange. That is the case shown in Figure 22.a, where the user datagram protocol (UDP) is used to carry the multimedia packets (this means that UA2 may start transferring data as soon as it sends its final response to the first invite transaction). This example of flow negotiation assumes a successful resource reservation initiated by both call participants (UA1 and UA2). Thus, in the re-invite transaction the QoS-enabled flow is activated in both directions (*send/receive*).

Other types of applications include those that may be overwhelmed by discrepancies in the resource availability in the network. Either the required bandwidth is available for the application and the application operates normally, or the application does not operate in an acceptable mode that allows communication between end users. In this way, the best option for the application is to put the media flows on hold during the BE interval, as shown in Figure 22.b. Basically, there are no BE flows and the QoS flows are set to inactive operation until the user agents update each other on the status of the reservation. In the case of a successful reservation, we can say that for non-adaptive multimedia applications the BE interval represents the post-pickup delay (i.e., the delay after the call is answered) that the callee experiences before QoS-enabled multimedia exchange begins.

Now considering the case of an unsuccessful reservation, then the call may continue as BE for adaptive applications but must be cancelled for non-adaptive applications. The latter means a greater impact to users who have been alerted in vain. However, based on the previous assumption that an envelope has been set for the call during the call authorization process, the chances of an unsuccessful reservation are small. Thus, the impact to call participants lies mainly on the BE interval.

To sum up, the main point of the BE/QoS flow negotiation approach is to properly handle the users' ability of an early media transmission that comes as a result of the ROAD

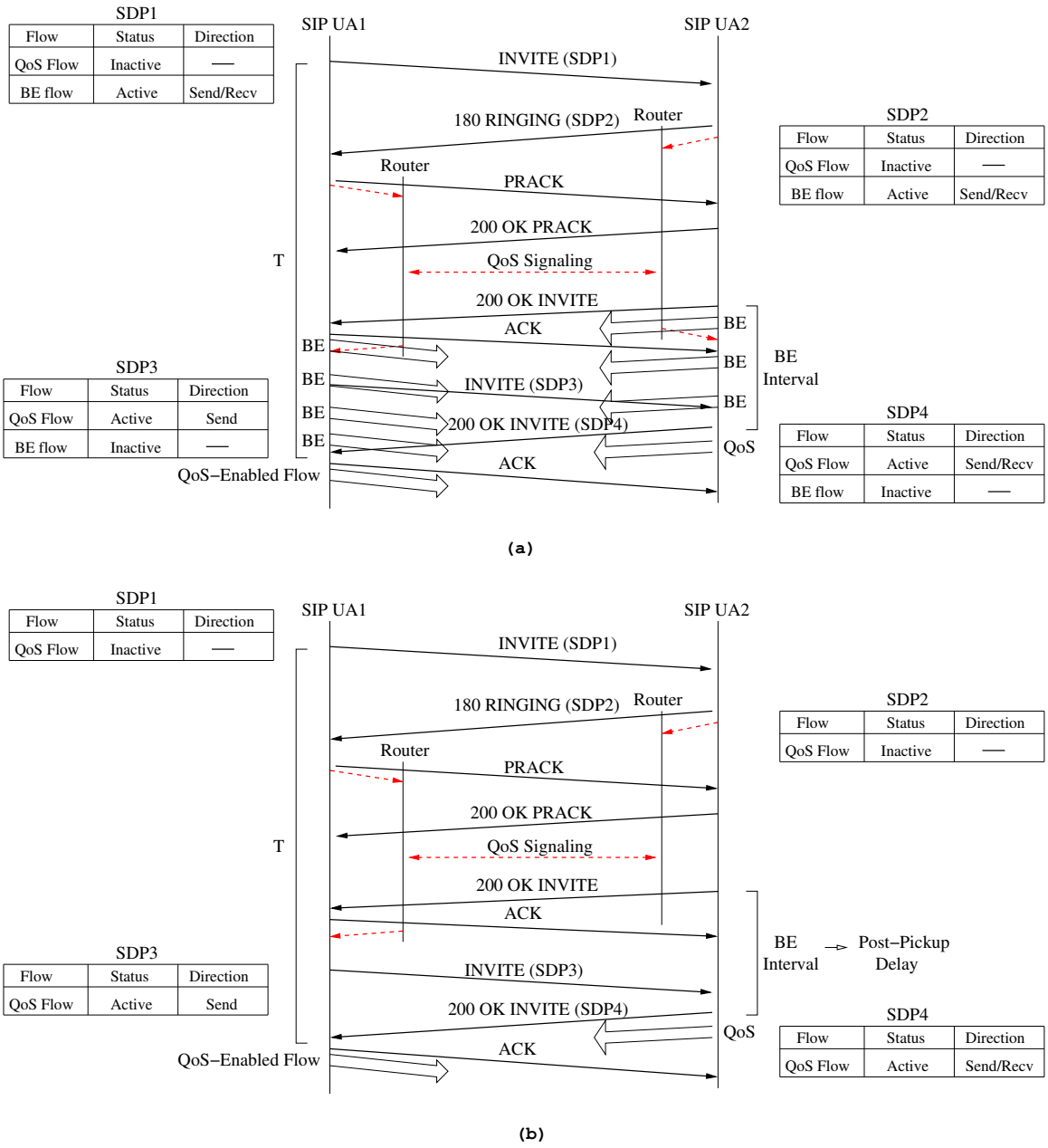


Figure 22: ROAD flow negotiation: (a) for adaptive applications, (b) for non-adaptive applications.

call setup way. To adaptive multimedia applications, it allows an earlier start of the multimedia session and reduces the idle time during call setup. To non-adaptive multimedia applications, it holds the multimedia transmission until the full completion of the ROAD call setup signaling.

4.3 Testbed Experiments

In order to study the interaction of session control signaling and resource management proposed in the ROAD call setup architecture, the SIP testbed (Figure 23) is configured in the same bottleneck architecture as described in the previous chapter. The main backbone allows only 10 Mbps of bandwidth, being the end users connected to 100 Mbps Ethernet hubs.

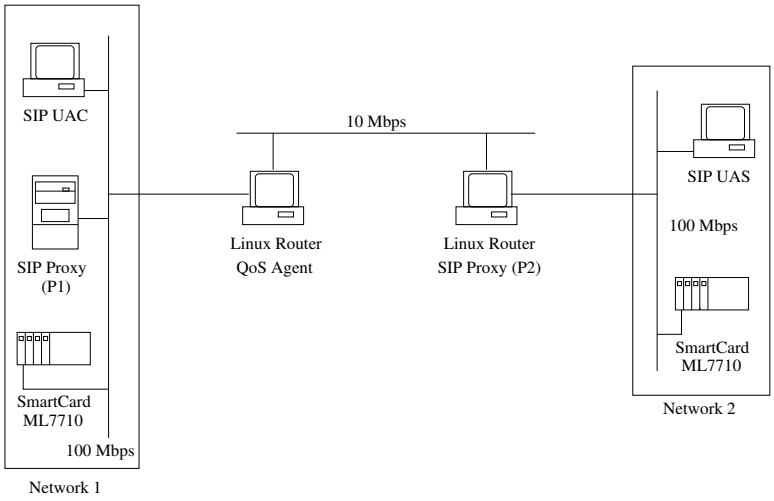


Figure 23: SIP testbed's topology and hardware.

At the access networks, a SIP user agent client (UAC) in Network 1 initiates a call setup with the SIP user agent server (UAS) in Network 2. A generic QoS agent is implemented in the same machine as the edge router of Network 1. When the SIP user agent sends a resource reservation request to the QoS agent, a reservation delay timer is triggered in order to emulate the propagation of QoS signaling requests throughout the network. The QoS agent configures the router's traffic control engine, which has Differentiated Services capabilities.

4.3.1 Metrics

The experiments have the goal to determine the delay, number of messages, and total number of bytes transferred in the set up of a multimedia call. The call setup delay (T) is computed from the time the UAC transmits the initial invite request to the time the UAC receives the final "ok" (as in Figure 20). Also, to exemplify the signaling overhead

measurements, Figure 24 illustrates the total number of bytes exchanged in one of the tests of the RFC3312-based call setup scheme. Note that the total number of bytes transferred is a sum of the size of all messages exchanged between SIP UAC and proxy P1.

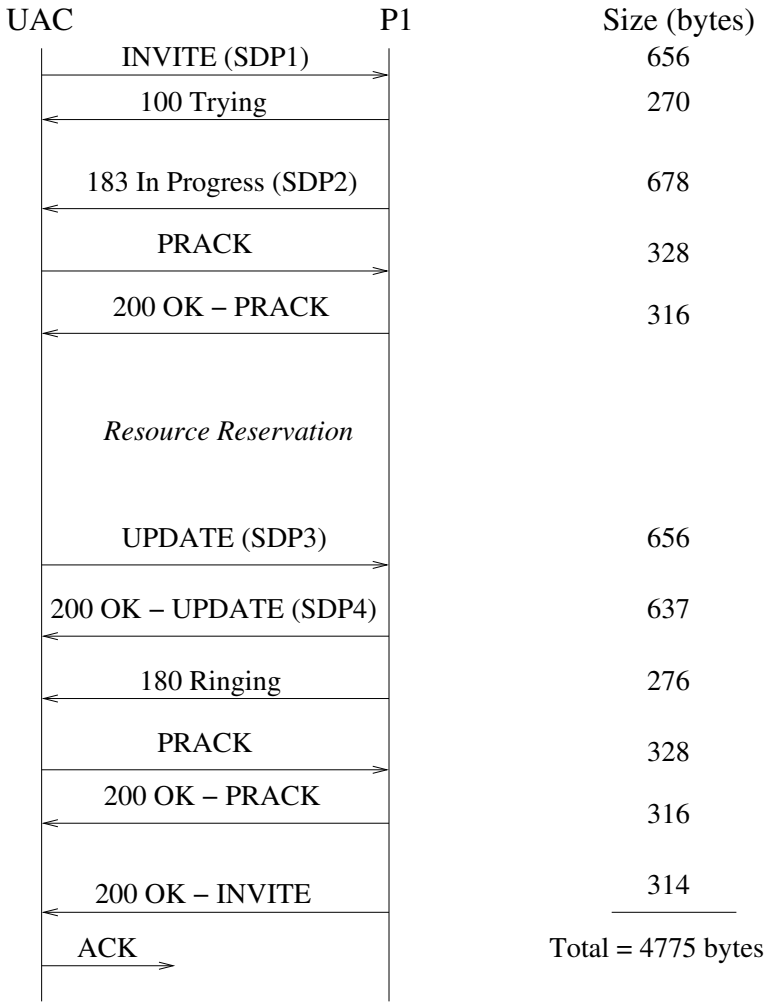


Figure 24: Example of how the number of bytes was obtained.

4.3.2 Results

In the call setup delay experiments, the focus is on the main differences between the ROAD signaling flow and the RFC3312-based signaling. Hence, two things are compared: how the sequence of events, and how the amount of signaling (messages, bytes/message) affect the call setup delay. Thus, it is assumed in the experiments that the SIP messages of both schemes experience the same processing delays and same network-related delays.

Experiments were performed in which the answering delay is set as a constant variable

(D_{ans} = 0, 1, 5, 10 and 20 sec) and the reservation delay (D_{res}) varies from 0 to 2 seconds (D_{res} = 0, 100 msec, 300 msec, 500 msec, 1 sec, and 2 sec). The values of reservation delays are in a range close to the one used in [53] (i.e., 1.5 sec for RSVP). The values of user answering delay tend to be higher since the alerting duration in telephony may be significantly greater than 1 second. Also, the user answering delay of 0 seconds is to test a boundary condition that can show the difference in the call setup delay of both schemes as a result of the overall delay of the signaling messages (d).

The call setup delay results are shown in Figure 25. In the plots of the RFC3312-based call setup delays, the delay increases linearly as the reservation delay increases and the answering delay is kept constant. In the plots of the ROAD scheme, however, we can identify these two different regions (as in Section 4.2.2):

1. When $D_{res} \leq D_{ans}$, a near flat line shows that the call setup delay can be approximated to the user answering delay ($T \approx D_{ans}$). In this region, the call setup delay savings of the ROAD scheme over the RFC3312-based call setup scheme increases with the value of the reservation delay (D_{res}). In addition, the results within this region for the experiment in which $D_{ans} = 1sec$ (Table 3) confirm that the BE interval's duration is very close to the duration of the re-invite transaction (T2).

2. When $D_{res} > D_{ans}$, the call setup delay increases linearly as the reservation delay D_{res} increases ($T \approx D_{res}$). This can be seen in the plots of D_{ans}=0 and 1 second. As the reservation delay increases for both schemes being compared, the difference between them becomes stable and approximately equal to the user's answering delay D_{ans}. Also in this region, the BE interval's duration in Table 3 jumps to a higher level, which is approximately the difference between the reservation and answering delays.

Note that in the ROAD call setup delay results of Figure 25, even when the answering delay is equal to zero, the ROAD call setup scheme is still advantageous in terms of reducing the total call setup delay, mainly because of its reduced signaling load. Our results show an average delay difference of 426 msec, when D_{ans}=0 seconds.

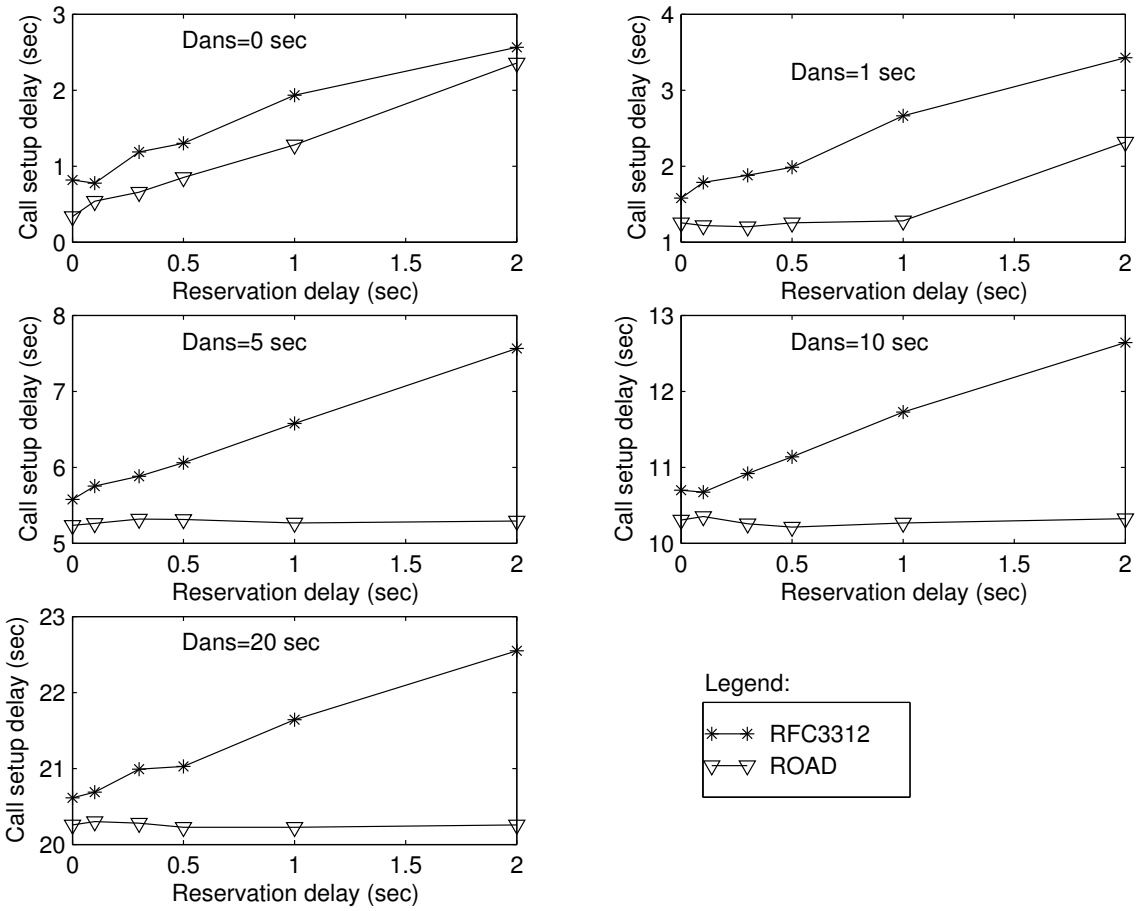


Figure 25: The ROAD scheme's call setup delay in comparison with the RFC3312-based signaling.

Table 3: The best-effort interval (D_{ans} = 1 sec).

D_{res} (msec)	T2 (msec)	BE (msec)	T (msec)
100	60	68	1218
300	59	62	1202
500	91	74	1255
1000	76	122	1281
2000	96	1136	2317

Proportionally to the call setup delay values obtained in the testbed for the RFC3312-based signaling scheme, the call setup delay reduction of the proposed ROAD scheme is depicted in Figure 26. As the answering delay increases to 20 sec, the call setup delay values of both schemes get closer, and the call setup delay reduction shows a phase of diminishing returns, being only around 10%. Basically, the delay of the reservation transaction is being

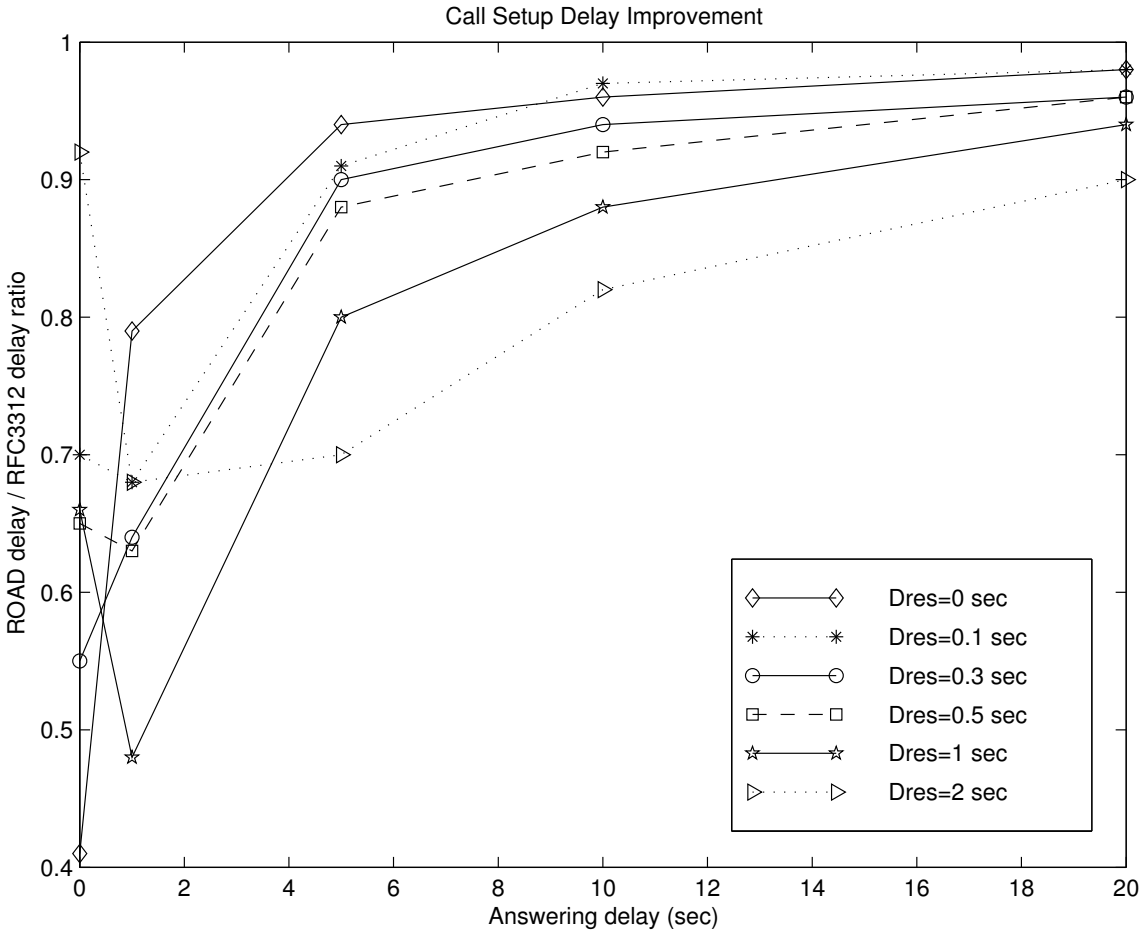


Figure 26: Delay improvement evaluation between ROAD and RFC3312's delays.

saved when the call setup and reservation are done in parallel. Thus, the real advantage of the proposed scheme was achieved for values of reservation delay close to the answering delay. For instance, the call setup delay reduction of the ROAD scheme was in the range of 30% improvement when $D_{res} = 2sec$ and D_{ans} varied from 1 to 5 seconds. Even better results (up to 50%) were obtained for $D_{res} \leq 1sec$ and $D_{ans} = 1sec$.

Besides offering call setup delay savings, the ROAD scheme requires less messages and comparable number of bytes per call setup than the RFC3312-based scheme: 9 messages *vs.* 11 messages, and in average 4625 bytes *vs.* 4787 bytes, respectively. However, both schemes add considerable overhead to a basic SIP call setup transaction: the average number of bytes of a basic SIP call setup being 1630 bytes, for 4 messages (without acknowledgement messages to provisional responses (PRACKs)). A comparison of the signaling load of both schemes can be observed in Figure 27, which shows the number of bytes per call setup when QoS-related information is added to the SDP content of SIP messages, i.e., the BE/QoS flow negotiation for the ROAD scheme, and the basic preconditions approach for the RFC3312-based call setup scheme. The leftmost bars compare both schemes using a plain SDP content, i.e., without QoS-related information in the body of the SIP messages. Then, the rightmost bars compare both schemes using the flow negotiation and preconditions approach. The results show that the ROAD scheme with the proposed BE/QoS flow negotiation adds 26% more bytes than the same ROAD signaling with plain SDP, while the RFC3312-based signaling scheme with preconditions creates around 11% increase in the RFC3312-based signaling without preconditions (i.e., for the conditions we assumed in the experiments: end-to-end QoS in both directions for two flows). In conclusion, the ROAD flow negotiation with a QoS-enabled flow and a BE flow creates proportionally more overhead than the preconditions approach of RFC3312, given the conditions tested. But for similar SDP contents, the ROAD scheme always uses fewer messages and bytes per call setup than the RFC3312-based scheme.

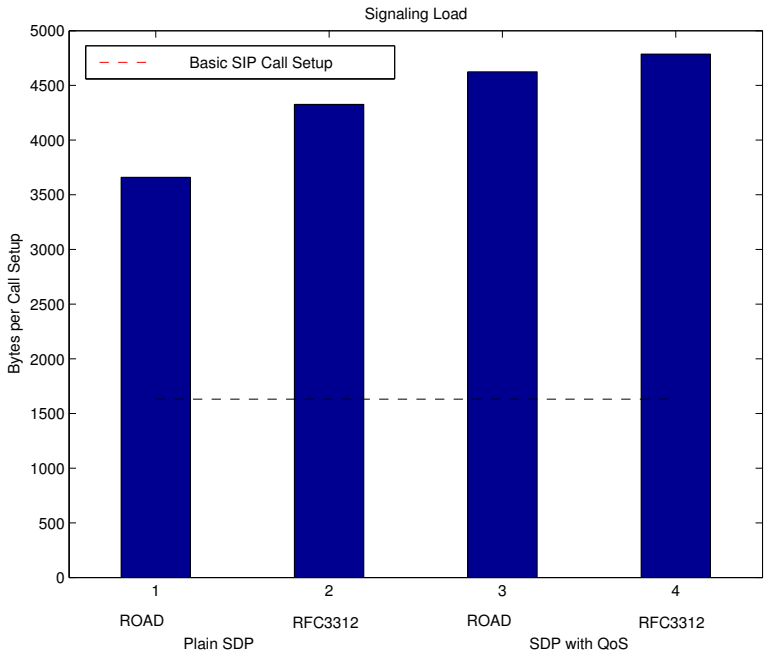


Figure 27: Comparison of number of bytes increase.

4.4 Conclusion

Motivated by the idea of efficiently allocating the idle times (i.e., reservation and user answering delays) during a call setup transaction integrated with resource management, this chapter proposed the ROAD (*Resource management Overlapped with Answering Delay*) signaling scheme. It can be easily implemented in current SIP UAs that support communication of QoS information in the form of preconditions, as proposed for the integration of SIP and resource management (RFC3312). Basically, the ROAD scheme is envisioned as a new version of the RFC3312-based signaling scheme: one that applies a different and more flexible model than the traditional telephony-style call setup model; one that may suit better users and network operators that favor shorter call setup delays and less signaling load, at the expense of possibly less strict QoS guarantees.

By allowing a best-effort/QoS flow negotiation during call setup, the ROAD scheme avoids the "no QoS/no call" problem for adaptive applications. As for non-adaptive multimedia applications, users may experience a brief post-pickup-delay or, in the worst case of an unsuccessful reservation they may be alerted in vain. In this last issue, as future work some specific applications will be chosen to evaluate the probability of unsuccessful

reservations *vs.* the risk of alerting users prior to the completion of the reservation.

With the SIP testbed, it has been demonstrated the feasibility of having a SIP call setup scheme integrated with resource management, where the call setup transaction and the resource management transaction occur in parallel. On the evaluation of the call setup delay savings and signaling load of the ROAD scheme, the conclusion is that in most cases the ROAD scheme offsets the reservation delays. In addition, proportionally to the call setup delays obtained with the RFC3312-based signaling scheme, the most gains from the ROAD scheme are achieved when the user answering delays and the reservation delays are in a close range.

Moreover, the ROAD scheme is based on a new call setup paradigm and offers an experimental view of the interaction of SIP and resource management.

CHAPTER V

ON THE INTERACTION OF SIP AND ADMISSION CONTROL: AN INTER-DOMAIN CALL AUTHORIZATION MODEL WITH QOS ENHANCED SIP PROXIES

In heterogeneous network environments such as the Internet, assuming that the networks support service differentiation, each in their own way, it is very important to manage how interactive multimedia applications use the networks' enhanced services. As a step in this direction, ways to ensure that these services are properly authorized and accounted for are needed. Therefore, this chapter addresses the role of QoS-enhanced SIP proxies to authorize QoS-enabled multimedia sessions, based on the session's policy information and the network resources' availability.

5.1 Problem and Solution

In the SIP architecture, user agents (UAs) must be authenticated and authorized for every call or session request they make. Usually, SIP proxy servers perform the role of authentication and authorization [95]. For call authentication, SIP uses a digest authentication mechanism [92] derived from HTTP digest authentication [39]. However, no call authorization system is recommended in the SIP standard. For this reason, the way the proxy performs the call authorization is still generic and open for discussion.

The proxy's role of call authentication and authorization can be combined to the role of an *application-layer admission control*. The latter meaning that the application layer interacts with the network layer entities to verify in a higher level if the network can admit the call. This is done in the initial steps of the call setup signaling, and has the potential to avoid the large number of signaling transactions that are transferred between end domains

when for instance the requested resources cannot be committed to the call. Basically, what has just been described is very similar to the idea of *gate controllers*. The concept of gate controllers was introduced in the DOSA network architecture for IP telephony [48]. The *gates* are the ones that control and provide access to network resources, such as edge routers that act as policy enforcement points (PEPs). *Gate controllers* make the decision on whether the *gates* should be opened to admit the new call and they set up the *gates* for the session.

The architecture in which the gate controller concept was originally proposed is a generic signaling architecture and does not address the details of the SIP architecture. A more recent work by the IETF (RFC3313) [66] defines a SIP extension that can be used to integrate QoS admission control with call signaling. This extension consists of a new header field called *P-Media-Authorization* header, which the proxy attaches to a SIP message to inform the UA of the results of QoS media authorization (Figure 28). The *P-Media-Authorization* header is only applicable in administrative domains, or among federations of administrative domains with common policies.

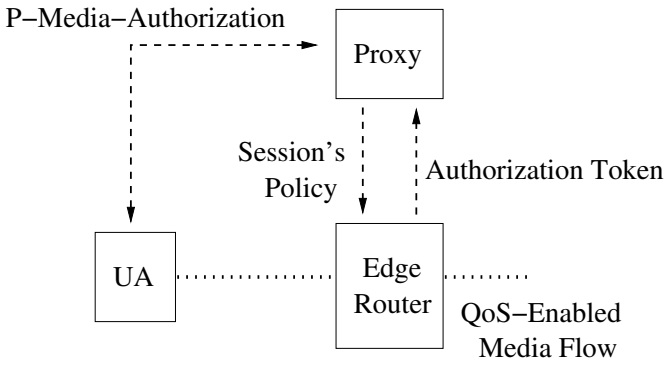


Figure 28: Basic architecture for media authorization (RFC3313).

It is important to note that the motivation for the SIP extension proposed in RFC3313 is to provide policy control over whether a media flow should have access to QoS (i.e., media authorization) or not. Therefore, the interaction between application layer and network layer is mainly in terms of transferring session's policy information and the results of call admission on a preliminary policy-based admission control in the interface between SIP proxies and PEPs. The results of media authorization are in the form of an authorization

token, which RFC3313 describes as "sufficient information for the user agent to get the authorized QoS for the media streams".

In the interaction of SIP and admission control, there are additional services that can be provided to enhance the media authorization model proposed in RFC3313. Here are some aspects that are considered in this chapter:

- *Call authorization status at the destination domain:* There are some open issues on how to ensure that the call request that arrives at the destination has been properly authorized at the origin domain. Media authorization is the main issue here, since in the scheme proposed in RFC3313 the media authorization token is transferred only inside a domain, from proxy to user agent through the *P-Media-Authorization* header. Therefore, the destination domain has no information on the call authorization status at the origin domain. Typically, the destination domain assumes that if the request has been successfully forwarded then it has been properly authorized at the origin domain. However, considering that the signaling path is usually independent from the data path, potential problems occur when the signaling path does not include the proxy that would interact with the network for media authorization. Another potential problem is when the *invite* request is forwarded independently of the positive or negative results of the media authorization process. In both cases, there is no way that the destination domain can verify whether the call has received QoS authorization at the origin domain or not.

- *Caller's account information:* Additional granularity to the call authorization process is desired in an open environment of untrusted domains and/or untrusted users (e.g., mobile users), according to [101]. Also, the destination domain requires additional information about the caller's identity in order to authorize the call completion or not. However, if the caller's additional attributes are needed for call authorization at the destination domain, in general its proxy would have to retrieve it from a remote server and this would originate additional delays to the call setup process.

- *Signaling impact on the call setup process:* The additional signaling needed in the

interaction of SIP proxies and the network layer to obtain media authorization for the session requires an analysis of its impact on the call setup process.

In addition, by combining the concepts behind the idea of SIP proxies as gate controllers to the media authorization model proposed in RFC3313, the interaction between SIP proxies and the network for call admission is extended to include a verification from the network layer on the resource availability for the session. Thus, an envelope is opened for the session and if the user reserves resources within that envelope, then the resource reservation will complete successfully. This can be seen as a two-way interaction between application and network layers, where the proxy informs the network about the session's parameters and QoS requirements and gets information on the network's ability to accept and handle the call. This completes the proxy's media authorization process.

Targeting in this direction, an implementation of a call authorization model (Figure 29) that combines the concept of gate controllers to the media authorization model proposed in RFC3313 is presented in this chapter. In this model, QoS-enhanced SIP proxies function as gate controllers to admit new calls to the network and to set up the session's policy at the edge routers (PEPs). This interaction of SIP proxies and edge routers provides similar

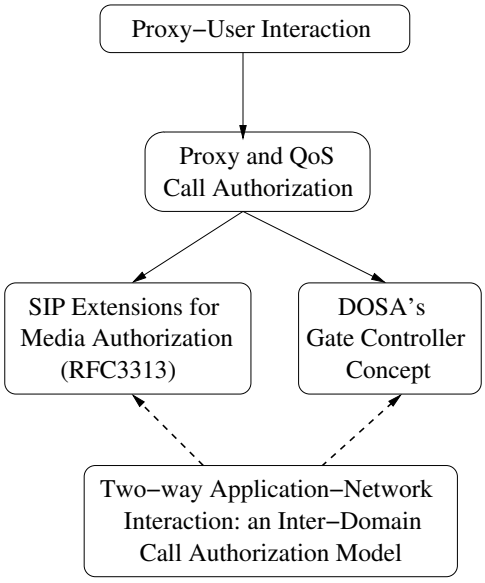


Figure 29: Inter-domain call authorization model's approach.

functionality to the interaction proposed in RFC3313. However, in the proposed implementation of SIP proxies as gate controllers, the goals are to overcome the aforementioned issues of communicating call authorization status to destination domains and adding more granularity to the call authorization process. In addition, the model is tested in a SIP testbed to evaluate the impact of this two-way proxy interaction with the network on the basic call setup signaling delays.

5.2 Network Scenario

The proposed work addresses the need of integration of origin and destination domains in the call authorization process; thus, call authorization in each remote access network is part of a larger call authorization model: an *inter-domain call authorization model*. Since QoS-enabled multimedia sessions are considered, this call authorization model is part of a call setup architecture that integrates SIP and resource management [19].

In this implementation of an inter-domain media authorization for Internet multimedia applications, the network scenario shown in Figure 30 is used as an example. This scenario assumes a set of common policies among end domains, as it occurs in a federation of trusted domains. However, there may be untrusted domains in the path between origin and destination domains.

The network scenario is adopted from a university (Georgia Tech) that has remote and independent campus locations, where multimedia sessions may be established between different campuses. Students, professors, or administrative personnel may move temporarily from one campus to another and still be able to access the multimedia services they usually access from their home campus. An example of user mobility across different campuses and a review of the main steps to establish a multimedia session using SIP are given next.

In this example, end user Alice (Alice@GT-Savannah) moves from her home campus to GT-Atlanta campus (step 1). She logs into a machine of the new domain and registers herself. Her registration request (2) is proxied to her home registrar server. The registrar server stores Alice's current location information in a location service database, which will be accessed when someone tries to call Alice. Based on the information on its current

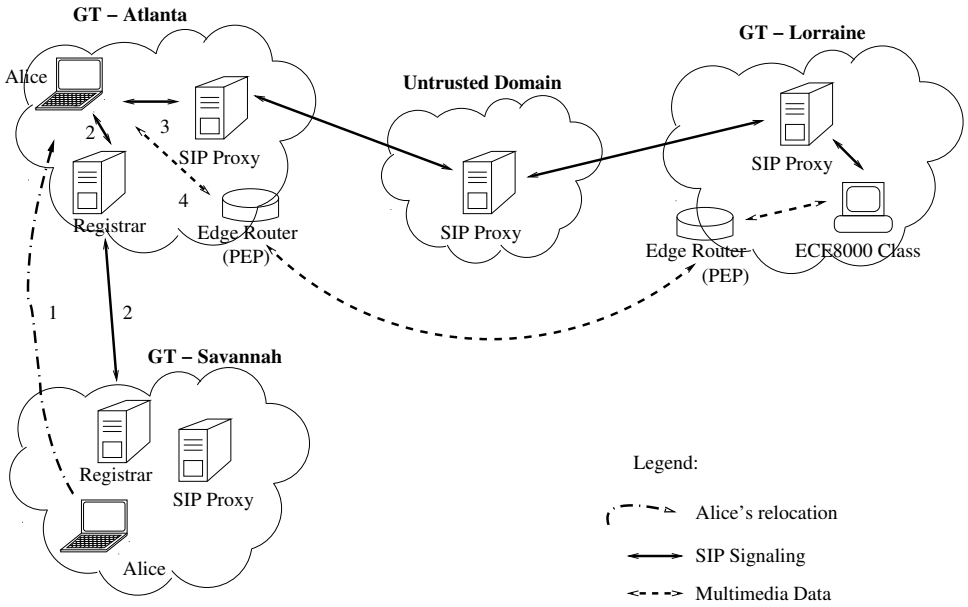


Figure 30: Network scenario.

location, the incoming call will be properly routed to Alice at the other domain. In other words, the registration process binds a user-level identifier (sip:alice@GT-Savannah) to a temporary IP address or host name [100].

At the new domain, Alice wants to set up a video conference with a user at another domain, e.g., GT-Lorraine. Alice can be for instance a student who wants to attend a remote class taught at GT-Lorraine, or a professor who wants to attend via video-conference a dissertation defense that is taking place at GT-Lorraine. So, Alice sends a SIP request to establish the new session (step 3).

Besides the fact that Alice is a new user in her new location (GT-Atlanta), there may be untrusted SIP proxies in the signaling path. Thus, during the session establishment Alice's identity must be authenticated in the origin and destination domains. Also, call authorization and QoS-enabled media authorization must be performed. For this, the signaling information, including Alice's authenticated identity [58] and its attributes (e.g., its authority level inside the university) must be safely conveyed from one domain to another.

At last, media flows are exchanged (step 4) in a data path that may be different from the SIP signaling path. Edge routers at each domain are considered as policy enforcement points to control the outgoing and incoming flows of a domain.

5.3 Inter-Domain Call Authorization Model

Based on RFC3313, there is really no way to inform the destination domain that the call has been authorized (in special media authorization). Therefore, an end-to-end approach to media authorization is to have a SIP proxy in each domain that strictly performs the role of a gate controller - a *QoS-enhanced SIP Proxy-GC* - and exchanges policy control information and call authorization status to trusted domains. An overview of the call authorization model here proposed is presented in Figure 31, which highlights the functionality of the proxy layers for call authorization. The relationship of the proxy with other signaling and transport-level entities and the steps needed in a call setup with inter-domain call authorization are also shown.

Concerning the structure of the QoS-enhanced SIP Proxy-GC for call authorization, the higher layers depend on decisions and information from the lower layers. As shown in the block representing the QoS-enhanced SIP proxy-GC in Figure 31, the top of the structure includes service-specific admission control and media authorization. They both rely on an authentication service, which in turn relies on user's account information that is stored in a local database. This local information can be obtained during the user's registration process, where in addition to informing its current location, the user can inform its media capabilities [88] and user's attributes for a more granular authentication process as discussed in [101].

In the proposed implementation, policy information and call authorization status are transferred between domains so that end users and proxies work together in the role of inter-domain call authorization. Policy information (e.g., user's authority level) and call authorization status are carried in a new header *P-Auth-Profile* added by the SIP proxy and transported over the network to inform the destination domain if the media flows have been authorized at the origin domain, and vice-versa. In Figure 32, the new header's fields are described, according to ABNF syntax [27].

As of the current implementation, the *P-Auth-Profile* header carries two parameters: the user's authority which is here defined based on our university campus network scenario (e.g., STUDENT, FACULTY, STAFF), and the call authorization status that explains whether

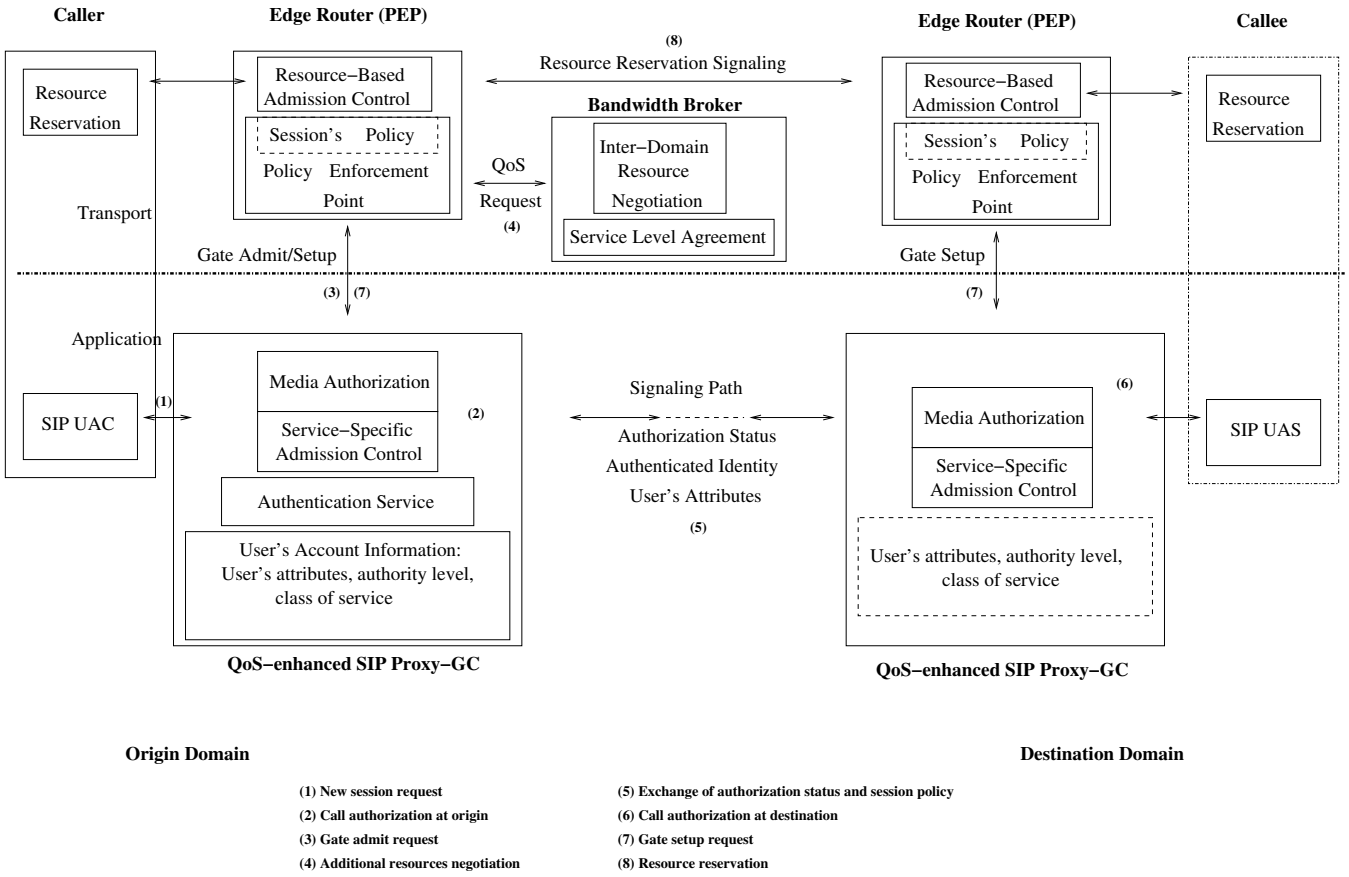


Figure 31: Inter-domain call authorization model.

P–Auth–Profile Header Fields

```
P-Auth-Profile = "P-Auth-Profile" HCOLON
                 user-authority SEMI call-authorization-status

user-authority = ("faculty" | "student" | "staff")
call-authorization-status = ("SRC" | "DEST" | "SRCDEST")
```

Figure 32: New header's *P-Auth-Profile* description.

the request has been authorized in the source, destination, or both domains (e.g., SRC, DEST, SRCDEST). Concerning the new header's application in SIP requests and responses, it can be used only in those requests and responses that can carry a SIP offer/answer (in a similar way as the *P-Media-Authorization* header [66]).

After the QoS-enhanced SIP proxy-GC defines that the request can be serviced (i.e., service-specific admission), it interfaces with transport-level entities (e.g., an edge router) to verify if the network can admit the call (*Gate Admit*) and to inform the network about the session's attributes (*Gate Setup*). In the first step of verifying if the network can admit the call, just a basic verification if the user can be serviced in its class of service is performed. The network verifies the amount of aggregated resources already allocated for the class. In case it is close to the maximum provisioned amount, the edge router can request more resources to a bandwidth broker, for instance using the inter-domain resource management model proposed in [105]. A second step of informing the network about the session's attributes and policies is needed to verify the resource reservation's requests (per-flow or per-session requests) that the user agent performs for the new session. This second step must be performed at both the origin and destination domains. SIP proxies-GC, thus, are in charge of transferring policy information between domains, by augmenting the header of the SIP message being forwarded with the new header *P-Auth-Profile* previously described. As this is a private header, the SIP proxy-GC may encrypt this information. Since in our network scenario the origin and destination domains share a mutual trust relationship, they can have a mutual agreement to decrypt the information in the header. Thus, at the destination domain, this policy information is used by the QoS-enhanced SIP proxy-GC to perform service-specific admission control and media authorization. This last step

means interfacing with the edge router to set up the session information that will be used to authorize resource reservation requests.

5.3.1 Call Signaling Flow

In summary, Figure 33 illustrates the call signaling flow at the origin and destination domains. Following the succession of events, at the origin domain these are the functions the QoS-enhanced SIP proxy-GC performs to authorize a new call request:

- User authentication (1)
  - After challenging the user, the proxy verifies its identity by querying a database that has the user's account information. This information may be limited only to the authority level and class of service of the user. For a new user in the domain, this information can be loaded during registration and its authenticity then verified with the user's home domain.
- Service-specific admission control (2)
  - Based on the user's information, the type and priority of the request, the proxy verifies the domain's local forwarding policies to verify, for instance, if this type of call can be serviced in this domain, if the number of calls the proxy can serve has not reached the maximum allowable limit, or if there are restrictions on forwarding the request to the destination.
- Media authorization - Gate Admit Request (3)
  - In this step, the proxy interfaces with a resource manager or policy control point at the network level to verify if the network can admit this call, in terms of required resources and the type of user. The proxy gives information to the network such as the bandwidth required for the call, the class of service, the call priority, and the authority level of the user.

If these steps complete successfully, the proxy forwards the request with additional policy control information in the *P-Auth-Profile* header. Upon receiving the request, the

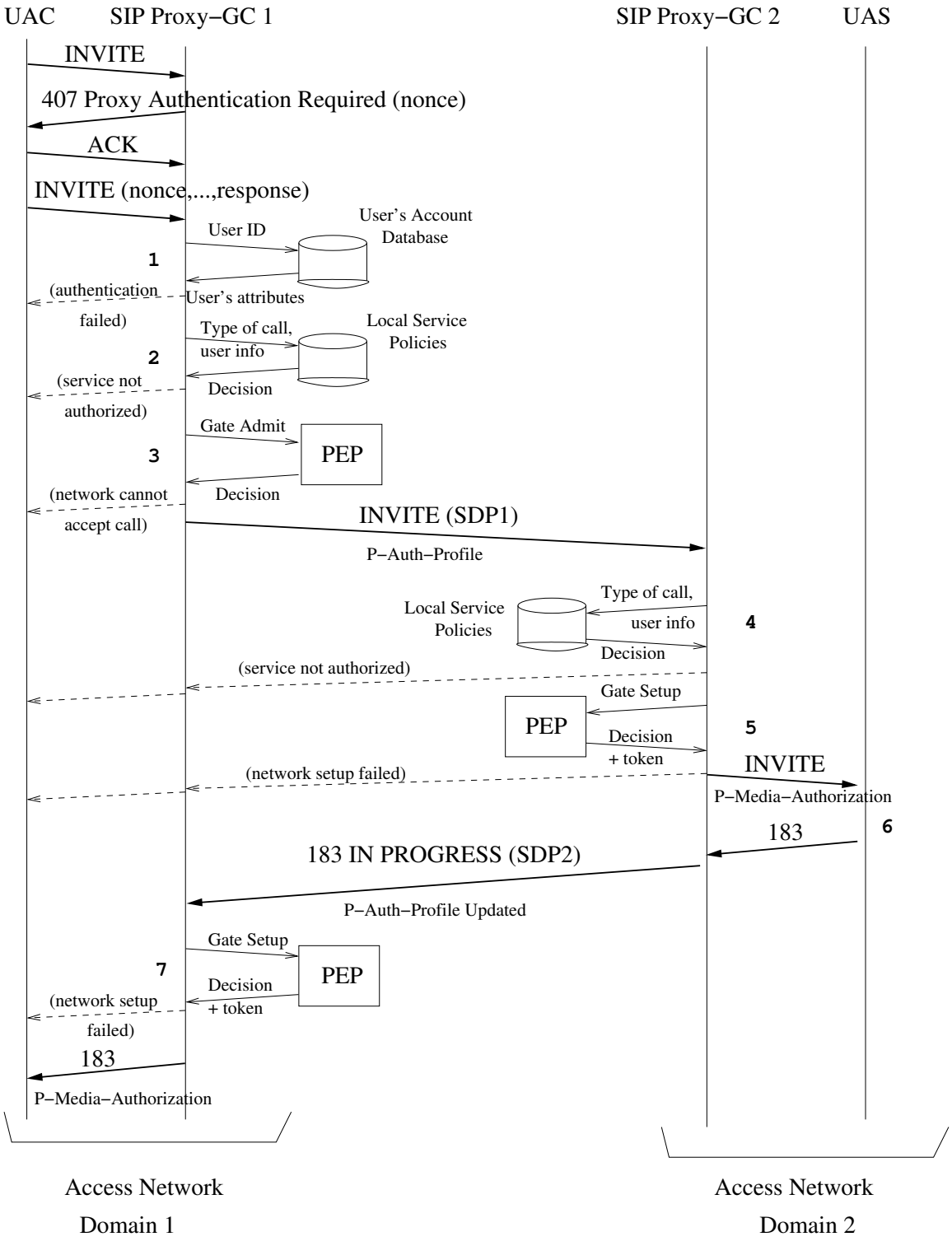


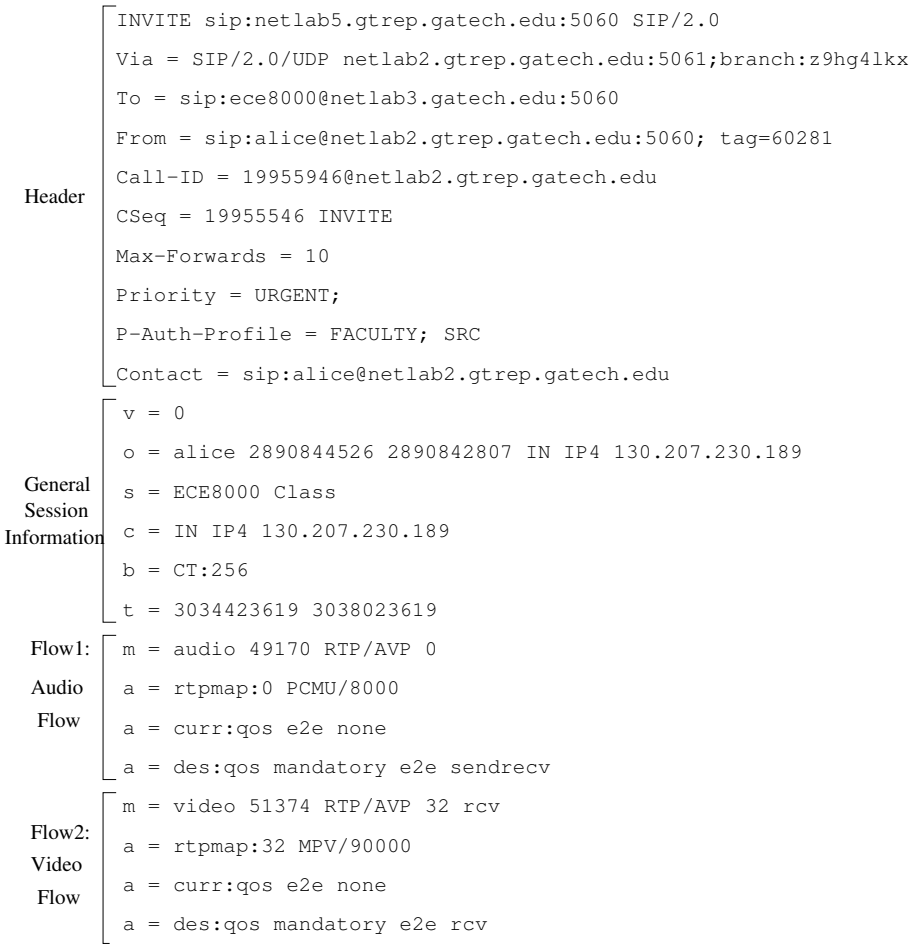
Figure 33: Call signaling flow with the inter-domain call authorization model.

QoS-enhanced SIP proxy-GC at the destination domain performs the following functions:

- Service specific admission control (4)
  - Being informed by the origin of the request of the user's attributes and that the request has been admitted to the network at the origin domain, the proxy at the destination domain makes the decision on whether to forward the request to the final destination. Local policies may define a maximum number of users in a given multimedia session, or only users at a given authority can make this type of call.
- Media Authorization - Gate Setup (5)
  - The proxy sends a gate setup request to the edge router with the policy information associated with the session information (e.g., call id) that arrived in the request. The edge router or policy control agent saves this information for use when it receives a resource reservation request for the flows associated with this call. Therefore, from this point on, the edge router saves a state for this session with this policy control information. (This state will then be maintained by the session reservation requests. If no reservation requests are received, it is automatically released.) The proxy updates the information on call authorization and media authorization in the header of the SIP messages so that states do not need to be saved at the application level.

If these steps complete successfully, the proxy then forwards the request to the final recipient of the call request. This user then performs the media capability negotiation to decide which media flows it can communicate (6). Then it issues a response with a new session description (*183 Session In Progress* response, with SDP2 in the body of the message, and an updated *P-Auth-Profile* header). When this response arrives at the origin domain, the proxy sets up the gate in a similar way as it is done at the destination domain (step 5), with the updated information on the media capabilities and the call authorization status (7).

When the resource management phase begins, reservation requests are triggered at the user agents at both the origin and destination domains. The edge routers then verify if the media has been authorized and if its gates have been set up for the call. To do this, it verifies the policy information contained in the resource reservation request with the policy information it has saved for that user. To illustrate the mapping of application-layer information to policy information for the session, Figure 34 gives an example of the contents of a SIP message (header and body) and the session's policy information that is sent to the routers and associated with the authorization token.



Session Policy Information:

	User ID	Authority	Service Type	Call–ID	BW	Flow Description
Token	Alice	FACULTY	Urgent	19955946	256	flow1 (send)
						flow1 (recv)
						flow2

Figure 34: Mapping of SIP messages content to policy control information.

Finally, based on the call setup signaling flow here presented (Figure 33) and the use of the header *P-Auth-Profile*, the QoS-enhanced SIP proxy-GC uses the algorithm illustrated in Figure 35 to forward the *invite* request at the origin and destination domains (i.e., caller's and callee's domains). In this algorithm, first the call's priority (defined in the `PriorityHeader`) and user's attributes are gathered at the caller's domain. Then, service admission is performed based on the caller's and callee's addresses, call priority, and caller's authority within the domain. This is followed by the interaction with the QoS agent (`sendRequest(GATEADMIT)`). Finally, the new header *P-Auth-Profile* is added (`addAuthorizationHeader`) to inform the destination domain about the call authorization status and caller's authority. At the destination domain, similar procedure occurs, with the difference that information about both caller and callee are gathered, and if the `GATESETUP` request is successful, the *P-Auth-Profile* is updated (`updateAuthorizationHeader`).

5.3.2 Unauthorized Calls

The QoS-enhanced SIP proxy-GC is a logical role of a SIP element. According to the SIP standard [92], when a request arrives an element that can play the role of a proxy first decides if it needs to respond to the request on its own. In this case of responding directly to a request, the element is playing the role of a user agent server (UAS). Thus, in the call signaling flow of the QoS-enhanced SIP proxy-GC (Figure 33) it is assumed that the proxy behaves like a UAS and thus can issue error responses back to the initiator of the request. The proposal here is to assign a specific code reason to the error response and to identify the element that issued the response. In the former case, a specific code reason explains the reason for the rejection of the request, such as if the type of service requested cannot be performed, or if the network does not have enough resources to admit the new call request; in the latter case, identification of the SIP entity in the error responses allows the initiator of the request to ask it for additional information about the error if needed.

Several categories of error responses are defined in the SIP architecture. Among them, an error response with a code *5xx* is issued when a server fails to fulfill a valid request. For instance, in RFC3312 when a user agent is not willing to meet the preconditions in the offer,

```
GateController::rcvRequest
if (request == INVITE) {
 priority = PriorityHeader->priority;
 if (CALLER's domain) {
    callerAuthority = getAttributes(caller);
    if (serviceAdmission(priority, caller,
         callerAuthority, callee) == TRUE) {
       sendRequest(GATEADMIT, qosHost, qosPort);
       if (qosResponse->gateAdmit == TRUE) {
          authorizationStatus = SRC;
          addAuthorizationHeader(authorizationStatus,
             callerAuthority);
         forward(request, callee);
        }
       else sendResponse("Network_Cannot_Admit_Call", caller);'
     }
    else sendResponse("Service_Not_Authorized", caller);'
  }

 if (CALLEE's domain) {
   calleeAuthority = getAttributes(callee);
   if (authorizationHeader->status == SRC) {
      callerAuthority = AuthorizationHeader->authority;
      if (serviceAdmission(priority, caller,
           callerAuthority, callee, calleeAuthority) == TRUE) {
        sendRequest(GATESETUP, qosHost, qosPort);
        if (qosResponse->gateSetup == TRUE) {
           authorizationStatus = DEST;
           updateAuthorizationHeader(authorizationStatus,
                   callerAuthority);
          addMediaAuthorization(qosResponse->token);
          forward(request, callee);
          }
        else sendResponse("Network_Setup_Failed", caller);'
        }
      else sendResponse("Service_Not_Authorized", caller);'
    }
  }
 }
```

Figure 35: QoS-enhanced SIP proxy-GC's algorithm.

it issues a *580* response to indicate *precondition failure*. Adopting this same class of error codes, in this current implementation of the inter-domain call authorization model with SIP proxies as gate controllers, *500* responses are sent back to the user in the following cases:

- Service admission control fails - Error Code: *Service_not_Authorized*
- Gate admission request fails - Error Code: *Network_Cannot_Accept_Call*
- Gate setup request fails - Error Code: *Network_Setup_Failed*

Although specific code numbers have not been formally assigned in our implementation, the error reason description in addition to the contact header information identifying who issued the error message represent valuable information to the initiator of the call about the reason for his unauthorized call request.

Next, the advantages of sending error messages in the proposed call authorization model is explained in terms of the number of signaling messages per unsuccessful call setup transaction.

- *Signaling Reduction*

At the origin domain, the QoS-enhanced SIP proxy-GC verifies if the network can admit the call when the initial *invite* request is received. Through the *Gate Admit* request, the proxy checks with the network if the aggregate resources being used do not exceed a provisioned limit. This preliminary, possibly high-level resource verification request triggered by the QoS-enhanced SIP proxy-GC helps to avoid additional signaling messages that would otherwise occur when no *Gate Admit* request is issued.

In addition, the *Gate Setup* request issued by the QoS-enhanced SIP proxy-GC at both origin and destination domains allows the network to verify the session's policy information with the network's access policies, and if access is granted to the new session a media authorization token is assigned to the end users. This token is used when the actual reservation of resources occurs.

Considering negative replies from the network in one of the above cases (e.g., no admit request is granted or no media authorization token is assigned), the actions of the QoS-enhanced SIP proxy-GC (i.e., its requests and coded negative responses) avoid the additional signaling messages resulting from all the transactions needed to reach the resource reservation phase.

Moreover, according to RFC3312 additional signaling is needed after the resource reservation phase. Based on the call setup signaling integrated with resource management and the use of QoS preconditions, the results of the resource reservation phase are communicated between end users through the *update* transaction. In case of rejected or unauthorized media flows, end users can decide to reject the updated offer by issuing a *580 - Precondition Failure* response (i.e., assuming the current status does not meet the desired status for mandatory preconditions). This cancels the offer in the update transaction. However, as per RFC3311 [87] if a user agent receives a non-2xx final response to an update request, the session's parameters remain unchanged, as if no update request had been issued. Hence, the initial offer in the invite transaction is still pending, and the end user can issue a new *580 - Precondition Failure* response to cancel the initial offer. Also, the *580* responses should contain an SDP description to indicate which precondition status failed. Finally, the responses are acknowledged and then the call setup transaction ends.

Looking back at Figure 33 in Section 5.3.1, some of the dotted lines show the end of the call setup transaction in case of unauthorized calls due to failed *Gate Admit* and *Gate Setup* requests. For instance, in the case of a *Network_Cannot_Accept_Call* error, the number of SIP messages exchanged between user agent and proxy in Domain 1 is just 3 messages (INVITE, 500 response, ACK), i.e., not counting the messages to authenticate the user. Then, in the case of a failed *Gate Setup* request at Domain 2 (*Network_Setup_Failed*), the number of SIP messages exchanged between user agent and proxy in Domain 1 is also 3 messages, with the difference that the response takes longer to arrive, as the *invite* request and the response indicating the failure is transmitted throughout the network between origin and destination domains.

Now, considering the basic call setup flow integrated with resource management (and

without QoS-enhanced SIP proxies acting as gate controllers) shown in Figure 36, the number of SIP messages exchanged between user agent and proxy at the access network of Domain 1 is 8 messages, considering all the transactions prior to the reservation phase and the necessary closure of the offers.

Therefore, the interaction of QoS-enhanced SIP proxies-GC and the network layer prior to the actual resource reservation reduces significantly the signaling load in the access and backbone networks in the case of unauthorized call requests.

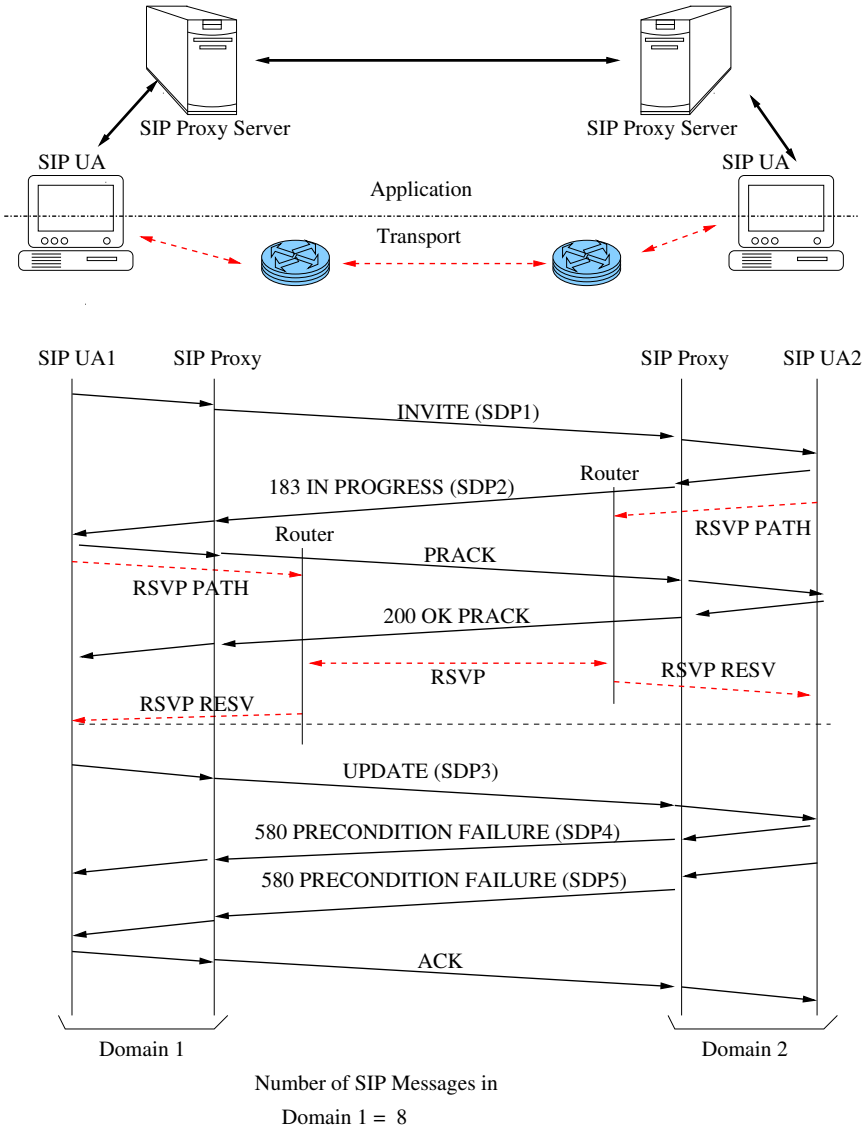


Figure 36: Signaling flow assuming preconditions failure.

5.4 Call Setup Delay Analysis and Scalability Issues

In the previous section, a simple comparison of the number of messages exchanged in the specific case of rejected calls due to failed media authorization was presented. However, in order to evaluate the real impact on the call setup delays of the interaction of QoS-enhanced SIP proxies-GC and the network layer this section addresses the additional service time imposed on the proxies, the effects on SIP messages retransmissions, and scalability issues in terms of the number of additional proxy servers needed to meet the service load in the network.

In the analysis that follows, the objective is to show that the inter-domain call authorization model proposed in this chapter is scalable, given delay restrictions on the processing time of the SIP transactions and application-layer retransmissions.

The additional load on the proxies that perform the functionality of gate controllers is highlighted in the oval areas of Figure 37. Basically, all the functionality of the proxy servers proposed in the inter-domain call authorization model (Figure 31), which includes service admission and media authorization at both the origin and destination domains, are now considered as increased proxy service time.

In addition, this implementation of the inter-domain call authorization model assumes UDP as the transport protocol. This has two important consequences:

1. a message-based transmission is assumed, i.e., the SIP messages can be transmitted in UDP datagrams without the need for application-layer framing;

2. the reliability of SIP messages is handled at the application layer, i.e., application-layer retransmission [92].

5.4.1 Restrictions on Application-Layer Retransmissions

On the subject of application-layer retransmissions, the impact of the additional service load on the proxies is mainly on the initial *invite* request and the arrival of the first reliable provisional response: *183 - Session In Progress*. In other words, the increased service time at the QoS-enhanced SIP proxies-GC delays the delivery of those messages.

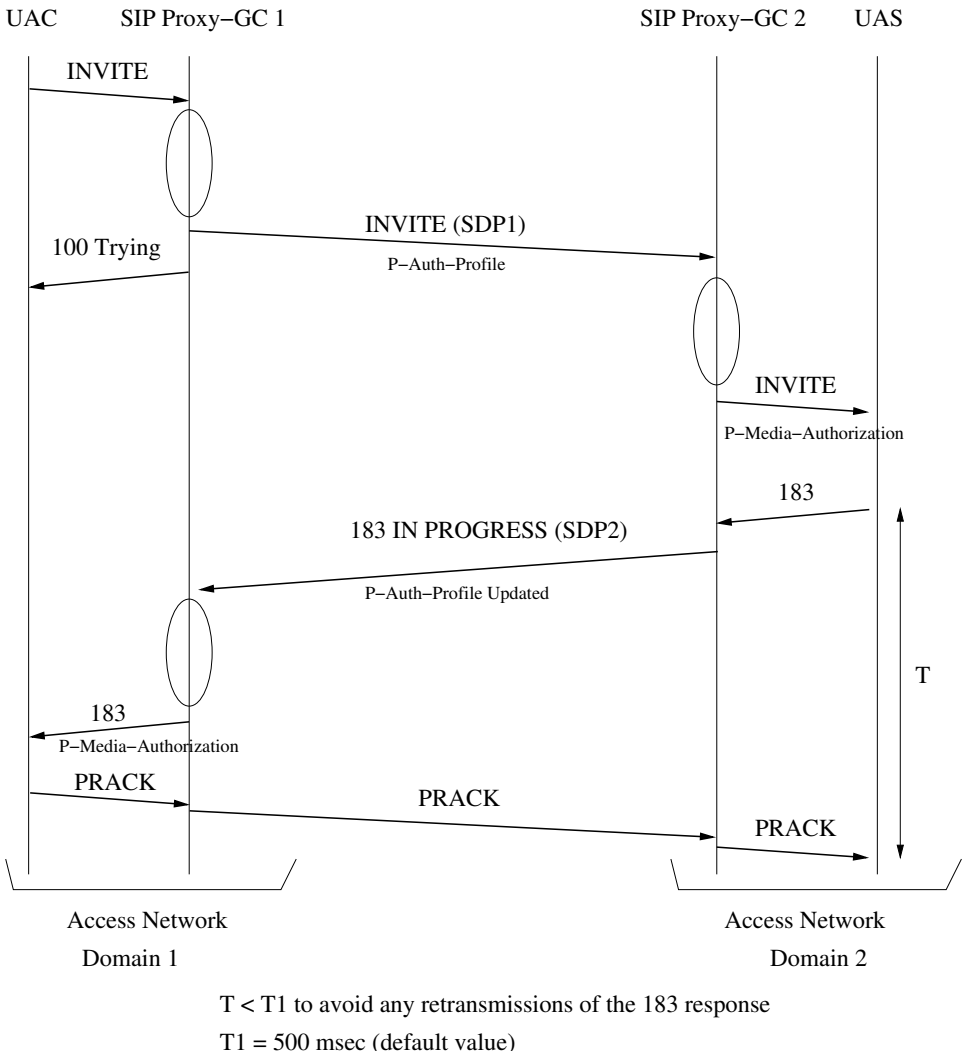


Figure 37: Increase in service time at proxy servers.

On the transaction layer of the user agent client (UAC) [92], the *invite* request is retransmitted after a certain timeout expires in case no provisional response is received (the UAC remains in the "calling" state). The timeout rule is based on the usual round trip delays, referenced as $T1$ in the standard, with a default value defined as 500 $msec$. A maximum of 7 retransmissions of the *invite* request is allowed, with intervals equal to $2^n * T1$, $n = 0, 1, 2, ..6$.

But after the UAC receives a provisional response, there is no more need to retransmit the request. The UAC keeps waiting for a final response from the network (in the "proceeding" state). The transmission of a provisional response (usually a *100 Trying* response)

is done by the server transaction state machines (either in a proxy or a user agent server). The server transaction state machine dictates that if no provisional response arrives within 200 *msec*, then it must send a *100* provisional response. Thus, if the *183* response from the UAS takes longer than that interval, the server transaction of the proxy at the origin domain will communicate with the UAC with a generic *100* provisional response to indicate that the new session request is being taken care of.

On the transaction layer of the user agent server (UAS), the *183 - Session In Progress* response must be acknowledged in order to be considered a reliable provisional response [90]. This requirement comes from the fact that it carries an SDP body with an answer from the initial offer; thus, the UAS waits for an acknowledgement from the UAC. A provisional response acknowledgement transaction (PRACK) then follows. In case this PRACK request does not arrive within a certain time, the UAS retransmits the *183* response. The maximum waiting time of the UAS is also a function of the $T1$ parameter, in a similar way as the retransmission of the *invite* request: the UAS passes the provisional response to the transaction layer periodically at an interval that starts at $T1$ and doubles for each retransmission until it reaches a value of $64 * T1$. If this maximum interval is reached, the UAS sends a final *500* response indicating that an error occurred in that transaction.

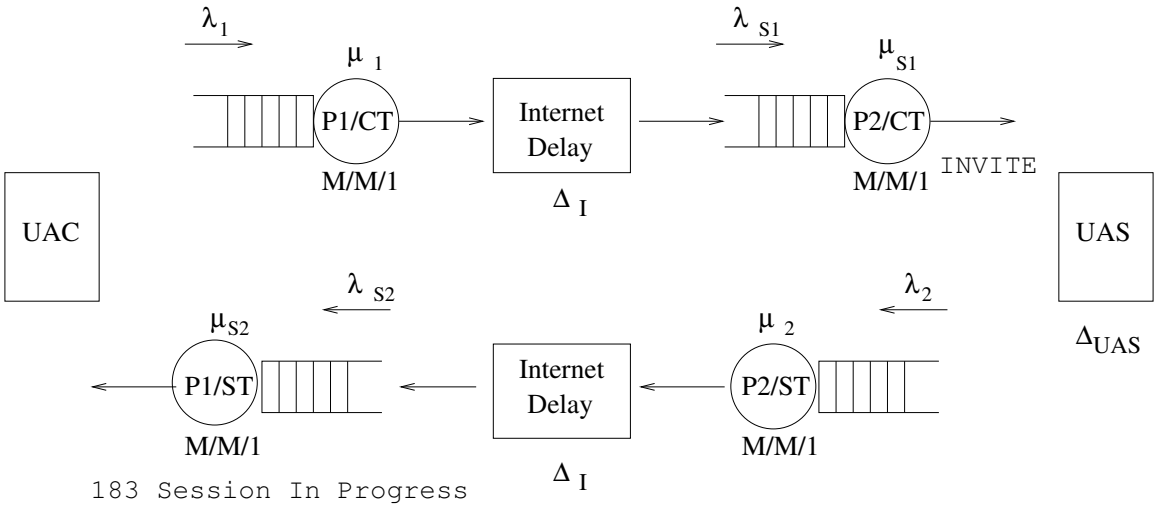
Based on the retransmission procedures defined for the user agent client and server transactions, it can be concluded that the interaction between QoS-enhanced SIP proxies-GC and edge routers during the call authorization process can be very critical and may cause the retransmissions of *invite* requests and *183 - Session In Progress* responses. In this analysis, the retransmission of the *183 - Session In Progress* response is considered more critical than the case of the *invite* request. The *invite* request receives a generic *100* response within 200 *msec* which will avoid its retransmissions, while the *183* will be retransmitted by the UAS if the processing of the QoS-enhanced SIP proxy-GC delays the response and the receipt of the provisional acknowledgement for more than the limit interval $T1$. These retransmissions of the response sent by the UAS will increase the load on the proxy servers, causing more delays, and consequently more retransmissions.

Therefore, the delay analysis for the proposed inter-domain call authorization model

addresses the timeouts for application-layer retransmissions of the provisional response *183 - Session In Progress* as the main constraint. Concerning scalability issues, the focus is on the service load of the proxy servers at the access networks (i.e., origin and destination networks), the impact on the number of users they can service and the need for additional resources given the delay constraints to avoid application-layer retransmissions.

5.4.2 Queuing Analysis

Consider the queuing model illustrated in Figure 38. This model is based on the assumption that the processing of SIP messages takes considerable time due to queuing delays at the SIP agents and servers. This approach has been taken by Banerjee *et al.* [8] to study the handoff performance of SIP for handling mobility. Using a similar approach, the proposed queuing model can be used to provide rough estimates of the delays due to the queuing of SIP requests and responses as a consequence of the increased service load at the QoS-enhanced SIP proxies-GC at the origin and destination domains.



$$\rho_1 = \lambda_1 / \mu_1 \qquad \rho_{S1} = \lambda_{S1} / \mu_{S1}$$

$$\rho_{S2} = \lambda_{S2} / \mu_{S2} \qquad \rho_2 = \lambda_2 / \mu_2$$

Figure 38: Queuing model.

The queuing delays are computed based on the assumption that the proxy servers perform dedicated jobs (i.e., to process SIP messages) and that they can be modeled as an

M/M/1 queue [11]: messages arrive according to a Poisson process with rate λ, and the probability distribution of the service time is exponential with mean $1/\mu$ seconds. Poisson random variables are used to model the arrival of new session requests (e.g., λ_1) because it is the most widely used arrival model in computer network theory, whereas exponential random variables are the most widely used service time model in queuing theory [109]. Also, the Internet delays are assumed to be constant as the main goal of this analysis is to evaluate the delays in the origin and destination domains due to increased service load on the proxy servers.

It is also assumed that proxies can accept SIP requests and responses from several other domains. For instance, the SIP message arrival rate at proxy P2 in the case of the *invite* request and P1 in the case of the *183* response result from requests and responses from other domains as well. Therefore, their arrival rates are shown as λ_{S1} and λ_{S2}, respectively. Additional parameters used in the analysis are described in Table 4.

The model shown in Figure 38 assumes separate queues within the same proxy server in the processing of requests and responses. This comes from the fact that in the implementation of the client-server-based SIP protocol, client transactions (CT) are defined to process the request, and server transactions (ST) are defined to process the responses. As an example, attached to the core of SIP proxy servers there may be one client transaction and several server transactions (as proxies can fork requests to several places). Therefore, it is reasonable to consider separate queues to process requests and responses.

The round-trip delay of the initial SIP *invite* request to the arrival of the *183* response ($T_{Inv-183}$), can be computed as

$$T_{Inv-183} = T_{P1/CT} + \Delta_I + T_{P2/CT} + \Delta_{UAS} + T_{P2/ST} + \Delta_I + T_{P1/ST} \tag{7}$$

Based on M/M/1 queuing systems, in steady state, the average number of customers in the system (in this case, messages) (N) is

$$N = \frac{\lambda}{\mu - \lambda} \tag{8}$$

Table 4: List of system parameters.

Parameter	Description
λ_1	SIP request arrival rate at proxy at origin domain (P1/CT)
λ_{S1}	SIP request arrival rate at proxy at destination domain (P2/CT)
λ_2	SIP response arrival rate at proxy at destination domain (P2/ST)
λ_{S2}	SIP response arrival rate at proxy at origin domain (P1/ST)
μ_1	Service rate at proxy at origin domain (P1/CT)
μ_{S1}	Service rate at proxy at destination domain (P2/CT)
μ_2	Service rate at proxy at destination domain (P2/ST)
μ_{S2}	Service rate at proxy at origin domain (P1/ST)
T_{SIP}	Delay of a complete call setup transaction (from sending the *invite* to receiving the final *200-OK* response)
$T_{Inv-183}$	Delay from sending the *invite* request and receiving the *183* response
$T_{P1/CT}$	Queuing delay at proxy at origin domain (P1/CT)
$T_{P2/CT}$	Queuing delay at proxy at destination domain (P2/CT)
$T_{P2/ST}$	Queuing delay at proxy at destination domain (P2/ST)
$T_{P1/ST}$	Queuing delay at proxy at origin domain (P1/ST)
Δ_I	Internet delay in transmitting of SIP messages
Δ_{UAS}	Processing delay at user agent server (callee)
K	Service rate multiplication factor ($K \in (0, 1]$)

Using the service load or utilization factor $\rho = \lambda/\mu$, the average delay per message (waiting time in queue plus service time) is

$$T = \frac{N}{\lambda} = \frac{\rho}{\lambda(1-\rho)} \tag{9}$$

$$T = \frac{1}{\mu - \lambda} \tag{10}$$

Using the parameters in Table 4 for equations 8 to 10, the following are the delay components of $T_{Inv-183}$:

$$T_{P1/CT} = \frac{1}{\mu_1 - \lambda_1} \tag{11}$$

$$T_{P2/CT} = \frac{1}{\mu_{S1} - \lambda_{S1}} = \frac{\rho_{S1}}{\lambda_{S1}(1-\rho_{S1})} \tag{12}$$

$$T_{P2/ST} = \frac{1}{\mu_2 - \lambda_2} = \frac{\rho_2}{\lambda_2(1-\rho_2)} \tag{13}$$

$$T_{P2/ST} = \frac{1}{\mu_{S2} - \lambda_{S2}} = \frac{\rho_{S2}}{\lambda_{S2}(1-\rho_{S2})} \tag{14}$$

- *Decreasing the Service Rate*

Consider a proxy server's processing time whose service rate (in messages/sec) is decreased from μ to $K\mu$, where $K \in (0, 1]$. It follows that the utilization factor ρ also decreases by a factor of K and the new queuing delay (T_{new}) increases by

$$\frac{T_{new}}{T} = \frac{1-\rho}{K-\rho} \tag{15}$$

It is important to notice that the steady state equations being used assume that $\rho < 1$, i.e., $\lambda < \mu$. Therefore, if the new service rate μ decreases considerably, the utilization factor $\rho = \lambda/\mu$ tends to 1. When the arrival rate λ becomes bigger than the service rate μ, then

a new server will be needed to avoid excessive queuing delays (i.e., tending to ∞). In this case, the M/M/1 queue becomes an M/M/m queue, with utilization factor $\rho = \rho/m$.

The M/M/m system is very similar to the M/M/1 queue, with the difference that there are m servers in the system and the messages at the head of the queue are routed to any server that is available. Although no particular routing scheme has been considered, there may be different approaches to choosing the next available server (e.g., a resource reservation approach or heuristic dispatching rules) and this issue has been considered in [93]. The derivations for an M/M/m queue can be found in [11], but here are the main formulas that are used in this analysis:

- The probability that a message finds all servers busy ($P_Q = P_{Queuing}$) and is forced to wait is

$$P_Q = \frac{p_0(m\rho)^m}{m!(1-\rho)} \tag{16}$$

  where

$$p_0 = \left[\sum_{n=0}^{m-1} \frac{(m\rho)^n}{n!} + \frac{(m\rho)^m}{m!(1-\rho)}\right]^{-1} \tag{17}$$

- The average delay per message is

$$T = \frac{1}{\mu} + \frac{P_Q}{m\mu - \lambda} \tag{18}$$

- And the average number of messages in the system is

$$N = \lambda * T = \frac{\lambda}{\mu} + \frac{\lambda P_Q}{m\mu - \lambda} \tag{19}$$

In the next section, the numerical results for this analysis are presented. Focusing on increased service times at the proxy servers because of the introduction of the new call authorization procedures, like the new service admission checks and the interaction with the router (*gate admit* and *gate setup* requests), the increased delays in the call setup

transaction are measured. The transmission times of the *183 response* and the limits to avoid retransmission are addressed using the queuing model. Finally, the delay analysis considers the introduction of new servers in the system (using the M/M/m queuing model). The increase rate of the number of servers *vs.* the increased network load can show whether the proposed model is scalable or not.

5.4.3 Numerical Results

5.4.3.1 Testbed Results

In the experiments, the QoS-enhanced proxy performing the functionality of the gate controller is tested in the SIP testbed, using a configuration similar to the one shown in Figure 23 (Chapter 4). The call setup procedure follows the signaling proposed in RFC3312. The algorithm described in Figure 35 has been implemented at the proxy servers at the origin and destination domains. The goal of the experiments is mainly to test the impact of the new authorization model in the overall call setup delay. For this, it is assumed a typical reservation delay of 500 *msec* (D_{res}) and a user delay to answer the call of 1 *sec* (D_{ans}).

Table 5 shows the results of the average overall call setup delay T_{SIP} and the round-trip delay of the initial *invite* request to the arrival of the *183* response $T_{Inv-183}$ with and without the implementation of the proposed inter-domain call authorization model. The delay increase with the implementation of the inter-domain call authorization model is a result of the delay increase at proxies P1 and P2 in the initial signaling exchange of the SIP call setup transaction (Figure 37). After that, the remainder of the call setup signaling is exactly the same for both cases. However, the remainder of the call setup signaling also experiences delay differences in both cases due to changes in the size of the messages, the processing of the new header, and normal delay variations in the testbed.

The delay component $T_{Inv-183}$ with the implementation of QoS-enhanced SIP proxies-GC is approximately 23% higher than the corresponding delay with the plain implementation of the signaling proposed in RFC3312. This increase is mainly due to the new functions implemented at the proxy's core and the proxy-router interactions such as the transmission of the *gate admit* and *gate setup* requests. However, the processing delay of those

Table 5: Call setup delays measured in the testbed.

	RFC3312	RFC3312 with QoS-enhanced proxies-GC
T_{SIP}	2132 $msec$	2358 $msec$
Ratio	1.00	1.11
$T_{Inv-183}$	128 $msec$	157 $msec$
Ratio	1.00	1.23

requests at the edge routers was not considered in the experiments, neither was the queuing delays. Note that in this testbed implementation, the proxies process only one call setup request at a time (up to the receipt of the *200 - OK* response or the sending of the final acknowledgement (ACK)).

5.4.3.2 Simulation Results

In a rough approximation of the increase service times at the proxy servers, and based on the experimental results obtained through the testbed, the simulations of the queuing model (Figure 38) in steady state initially adopt the increase delay ratio as 1.2 and get its inverse (0.833) as the factor K that decreases the service rate at the proxies. Then, different values of K are considered, up to a 50% delay increase in the service time ($1/K = 1.5$).

Also, in the next experiments, the queuing delays will only be evaluated for the initial signaling exchange of the *invite* request and the *183* response. This implies that the proxy servers may process the requests and responses waiting in the queue, one at a time, but not necessarily have to wait for the whole duration of the call setup transaction (up to the final *200* response) to process a new request. This means that they can keep the state of several call setup transactions.

Figure 39 shows the delays of the two-way signaling exchange ($T_{Inv-183}$) of the initial *invite* request and the receipt of the *183* response, using the parameter values listed in Table 6. In this experiment, to simplify the analysis it is assumed that $\lambda_{S1} = \lambda_2 = \lambda_{S2} = \lambda$, and that the service rate at the proxy servers is μ which is affected by the factor K at queues P1/CT, P2/CT, and P1/ST. The results show how the queuing delays increase

much faster as the number of new session requests increases when the service rate at the proxies is reduced by a factor K. For instance, when the utilization factor ρ becomes higher than 0.6 (at a SIP arrival rate of 30 messages/sec), the delay difference with and without the implementation of the new call authorization model becomes significant and increases exponentially. This means that less call requests will be serviced if limits are imposed on the delays.

Table 6: List of system parameter values.

Parameter	Value
μ	50 sec^{-1}
$1/\mu$	20 $msec$
K	0.833
ρ	λ/μ $(\lambda < \mu)$
Δ_I	150 $msec$
Δ_{UAC}	10 $msec$

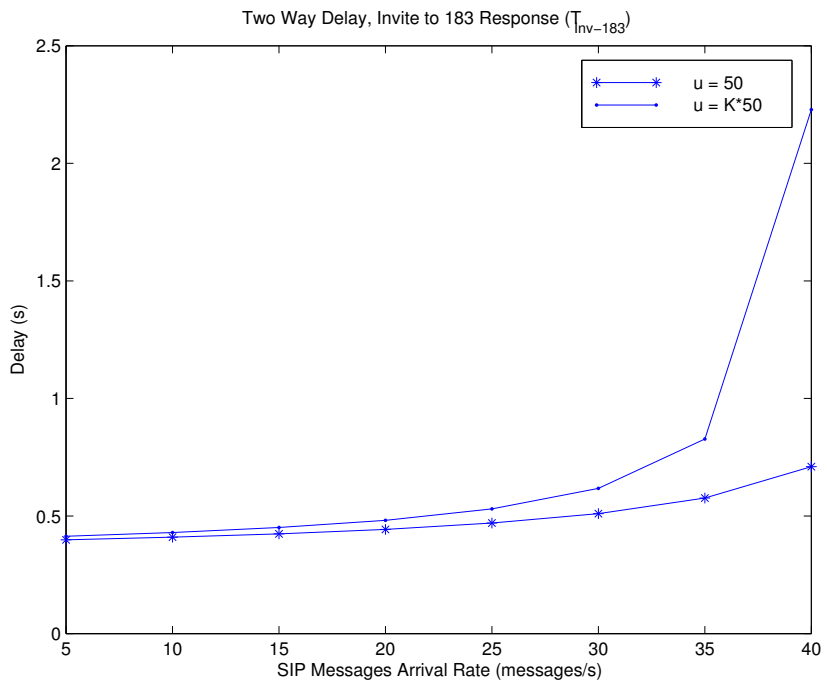


Figure 39: Delays between the *invite* request and the receipt of the *183* response.

In particular, there are time constraints on the receipt of the *183* response to avoid retransmissions. In order to avoid any retransmissions, the UAS waits $T1$ seconds between

the sending of the *183* response and the arrival of the provisional response acknowledgement message (PRACK). Hence, an approximation can be made that the one-way delay for the transmission of the *183* response between the UAS and UAC must be less than $T1/2$ (i.e., 250 $msec$, based on the standard's default value) to avoid any retransmissions. In Figure 40, the one-way delay from UAS to UAC of the *183* response is depicted for a service rate of μ at proxies P1 and P2, and then for a reduced service rate $K\mu$ at proxy P1. In the former case, the incoming rate of SIP messages must be limited to 30 requests/sec in order to avoid delays greater than $T1/2$. In the case of reduced service rate at proxy P1, the maximum number of incoming call requests is reduced to 25 requests/sec. In both cases, the utilization factor ρ is 0.6 and the number of messages both waiting in the queue and being serviced (N) is 1.5. Note that the number of SIP messages serviced that is reduced from 30 to 25 decreases by the same factor $K = 0.833$ as the service rate in the case of $\mu = 50$.

Now, considering a higher service rate ($\mu = 100$) in Figure 41 the number of incoming call requests reduces from 80 to 68 as μ decreases by a factor K. The utilization factor ρ in both cases is 0.8 and the number of messages in the system is $N = 4$. The number of call requests that is reduced from 80 to 68 decreases by a factor of 0.85 in the case of $\mu = 100$.

- *Adding More Servers*

Back to the case when $\mu = 50$, simulations of an M/M/m queue system with m=1,2,3 and 4 servers are performed. The number of servers varies only for proxy P1 in the path of the *183* response. First, the number of incoming SIP messages increases from 5 to 40 messages/sec in both proxies P2/ST and P1/ST; however, only proxy P1/ST is affected by the factor K. The delay results for 1 and 2 servers are plotted in Figure 42. Our previous simulation of the M/M/1 queue showed that the arrival rate of incoming SIP messages had to be reduced to 25 messages/sec to conform to delay restrictions ($T < T1/2$). Now comparing this result with the results of $m = 2$ servers, the arrival rate of SIP messages that can be serviced within delay constraints becomes 35 messages/sec, with an utilization factor of $\rho = 0.4$.

However, the previous result is also affected by the increasing utilization factor ρ at

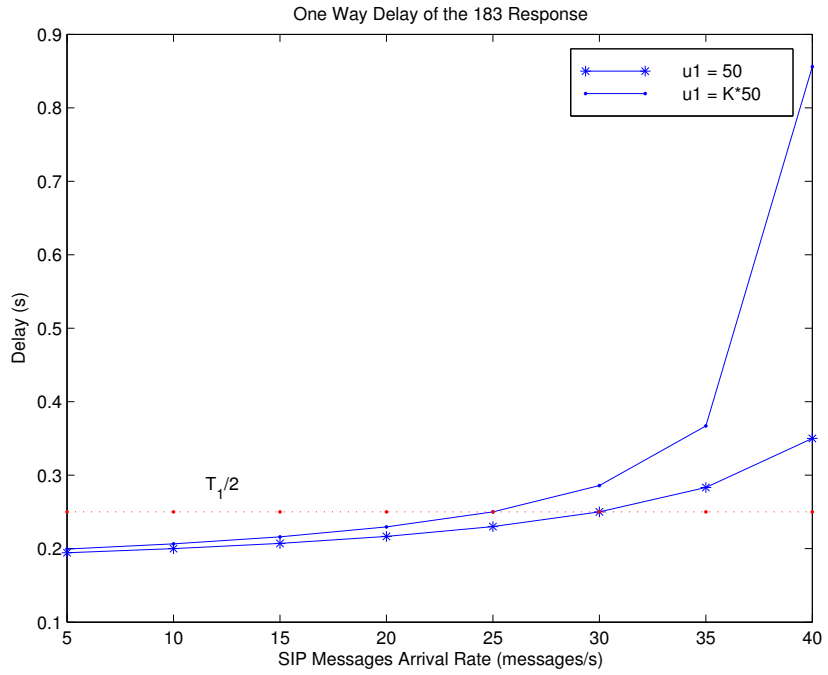


Figure 40: One-way delay of the *183* response (μ=50).

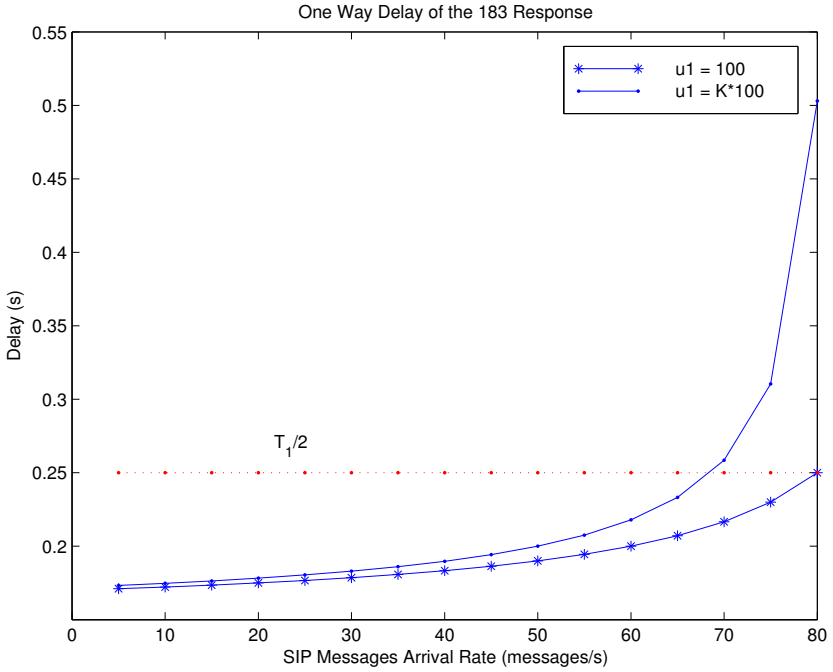


Figure 41: One-way delay of the *183* response (μ=100).

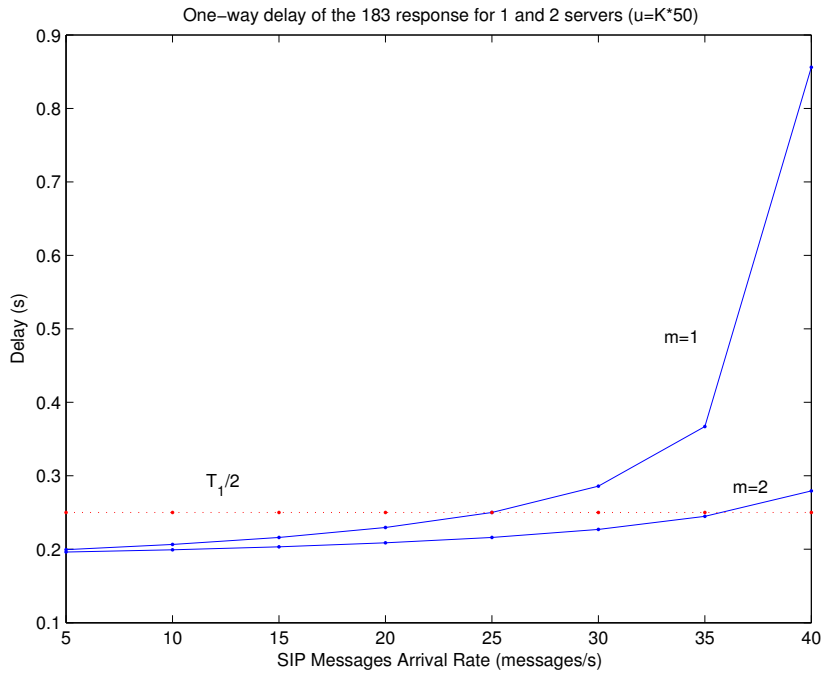


Figure 42: One-way delay of the *183* response for 1 and 2 servers at P1.

proxy P2/ST. Now, in order to better understand the effects of adding more servers to P1, the utilization factor at proxy P2/ST has been fixed to $\rho = 0.4$. This means that $\rho_{P2/ST}$ will remain the same at proxy P2/ST even for varying SIP messages arrival rate at proxy P1/ST. The reason is that proxy P2 in processing the responses does not perform additional tasks as a consequence of the implementation of the new call authorization model. Since only the number of servers at P1 is altered, it is better to isolate the effects of increasing messages arrival rate to proxy P1 only. For instance, proxy P1/ST may process messages from other Internet domains in addition to the destination domain shown in the model.

The new delay results for the *183* response transmission given a fixed utilization factor at proxy P2 are shown in Figure 43. In this case, when $m = 1$ server the maximum arrival rate changes slightly to 26 messages/sec. However, when $m = 2$ servers simulations of higher values of SIP messages arrival rate had to be performed to detect the maximum arrival rate given $T1/2$ delay restrictions. As shown in Figure 44 the maximum number of SIP messages becomes 72 messages/sec, more than doubling the maximum arrival rate for $m = 1$ server. Also, the utilization factor with $m = 2$ raises to $\rho = 0.86$, with $N = 4$

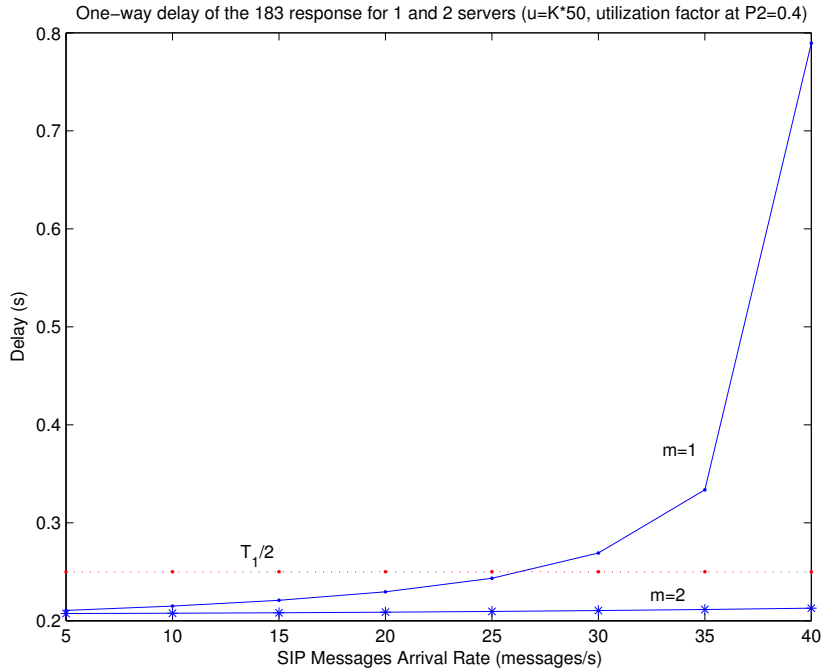


Figure 43: One-way delay of the *183* response for 1 and 2 servers at P1, and fixed utilization factor at P2.

messages in the system, either waiting or being serviced.

Using the same procedures for $m = 3$ servers (Figure 45), the maximum rate of arrival for SIP messages raises to 117 messages/sec, with a utilization factor $\rho = 0.94$ and $N = 7.5$ messages in the system.

Finally, the results of maximum SIP messages arrival rate and utilization factor *vs.* the number of servers (m=1,2,3,4) are summarized in Figures 46 and 47.

These results lead us to the conclusion that the inter-domain call authorization model is a scalable model regarding the need for more resources (i.e., more servers) at the access networks. In other words, the model imposes a higher processing load at the proxies (in this case proxy P1/ST queue is being analyzed), which may lead to the need to add more servers in the network to cope with an increasing number of incoming SIP messages to be processed. However, as the number of servers increases, the number of SIP messages that will be serviced within the delay constraints to avoid retransmissions increases in a linear rate with an average factor of 45 more messages per new server in the case of μ=50 and

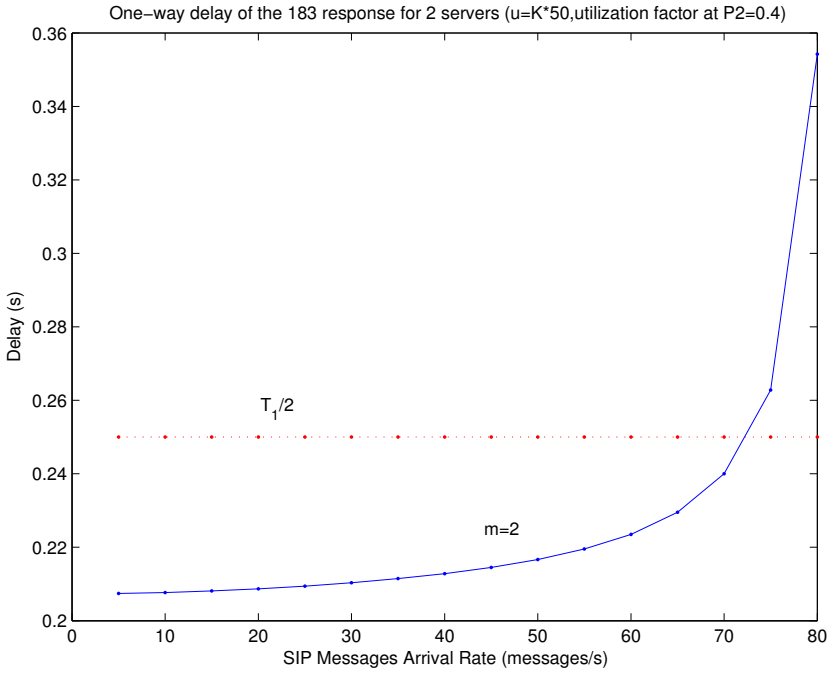


Figure 44: One-way delay of the *183* response for m=2 servers.

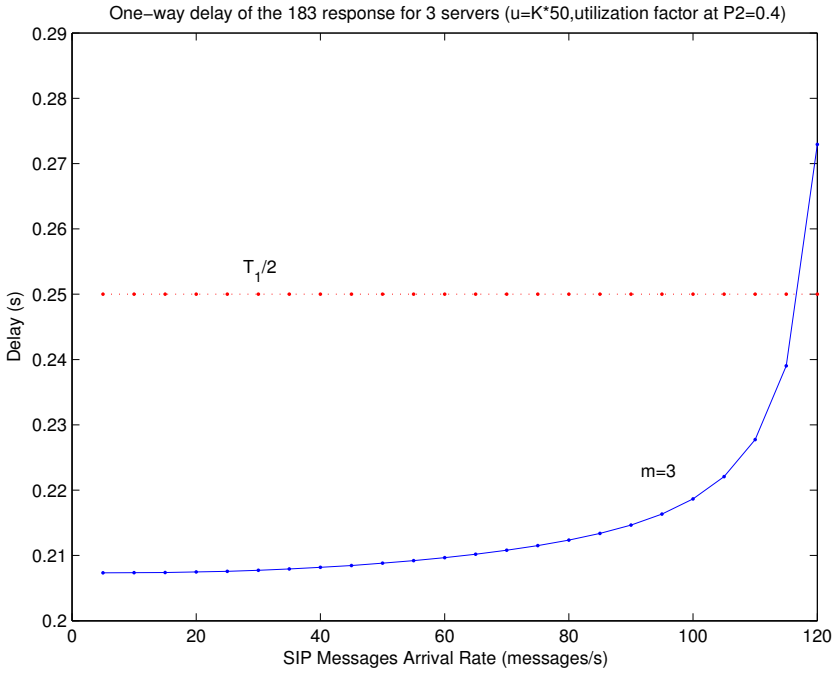


Figure 45: One-way delay of the *183* response for m=3 servers.

K=0.833 ($1/K = 1.2$). Moreover, the utilization factor $\rho = \lambda/m\mu$ increases asymptotically as new servers are added, tending to $\rho = 1$ (Figure 47). Even though this means more messages waiting in the queue, the delays in the signaling path of the *183* response were still kept under the threshold to avoid retransmissions.

To complement the previous results, additional results for varying values of K and μ also lead to the above conclusions. Figure 48 shows how the parameter K impacts the maximum SIP message arrival rate as new servers are added. For instance, when the increase in service rate is 50% ($1/K = 1.5$) the number of SIP messages that will be serviced within the delay constraints is reduced and increases at an average rate of about 36 new messages per new server added. On the other hand, Figure 49 shows the effect of a higher service rate at the proxy for a fixed parameter $1/K = 1.2$. The higher service rate μ=100 that was tested increased the number of SIP messages arrival rate in comparison with μ=50, and the increase rate was about 86 new messages that can be serviced for a new server added. Note these results were obtained for m=1 to 4 servers.

Of course, this delay and scalability analysis has some limitations, since the results can be dependent of the specific characteristics of the implemented tasks, and the network's parameter values used in the current model. However, the previous experiments to test the impact of the inter-domain call authorization model in the processing load of the servers and call setup delays can be useful to provide a first estimate on the performance aspects of the proposed implementation.

5.5 Conclusion

This chapter presented a new inter-domain call authorization model that explores the two-way interaction between application and network layers in the process of call admission control with QoS guarantees. This new model incorporates the concept of a gate controller [48] in which QoS-enhanced SIP proxies interact with network entities that control the access to network resources (e.g., PEPs) to verify firsthand if the network can admit the new call and to set up the network for the call. Also, the new model incorporates the idea of media authorization [66] and the need to assign a media authorization token to the

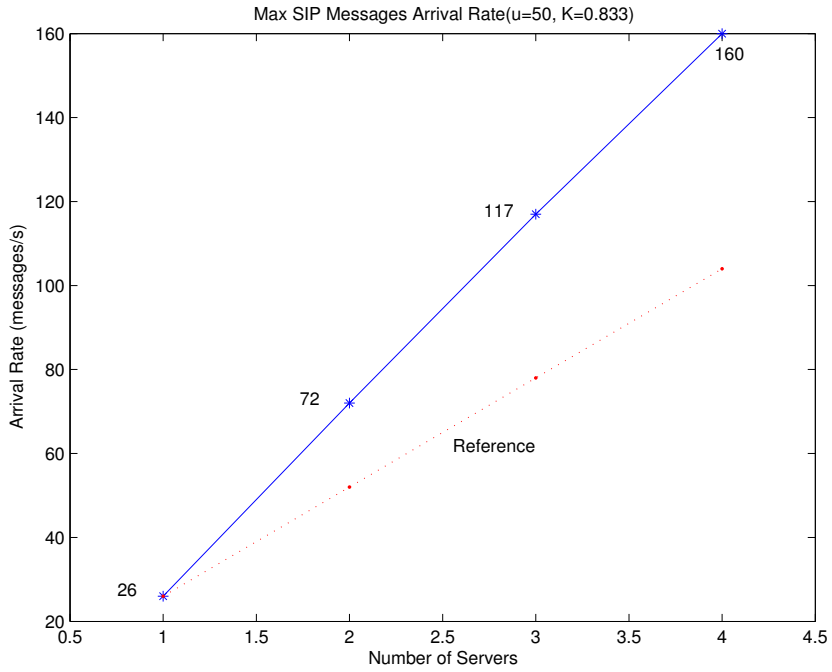


Figure 46: Maximum SIP messages arrival rate for a varying number of servers at proxy P1.

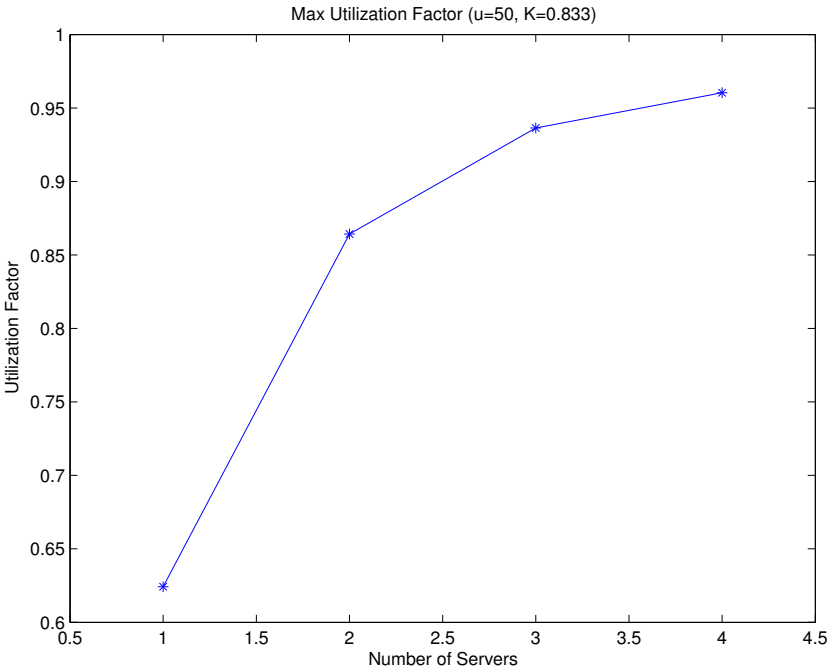


Figure 47: Maximum utilization factor for a varying number of servers at proxy P1.

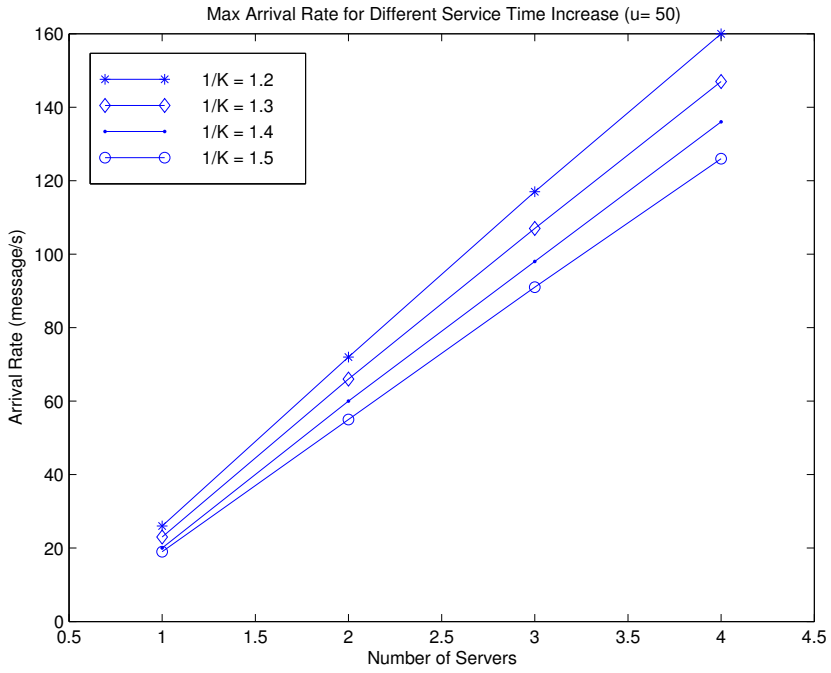


Figure 48: Maximum SIP messages arrival rate for different values of service time increase (1/K) at proxy P1.

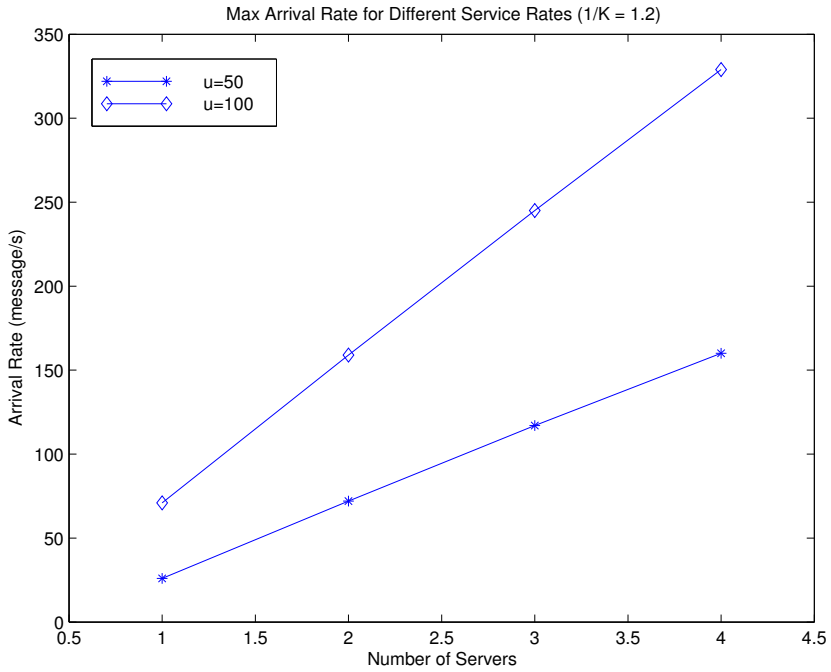


Figure 49: Maximum SIP messages arrival rate for different values of service rate (μ) at proxy P1.

session prior to the actual resource reservation. Based on those two concepts, and adding features to ensure that the call authorization status is transferred between domains and to increase the granularity of the authorization process, the new model has been implemented in a SIP testbed and its signaling impact on the whole call setup signaling transaction has been evaluated.

First, the case of unauthorized calls has been considered to evaluate the benefits of the new call authorization model. The introduction of additional error messages that precisely inform the reason for unauthorized calls helps to reduce the number of signaling messages in the access and backbone networks.

In addition, a delay analysis has been presented which addressed scalability issues in terms of the need to add more resources (i.e., servers) due to the increasing processing load at the proxies as a consequence of the inter-domain call authorization model. Deriving some parameters values from testbed experiments, simulation results showed that the model is scalable at the end domains. As the number of servers increases, the number of SIP messages that will be serviced increases in a linear rate. Although the numbers presented in the analysis are dependent on the specific characteristics of the implemented tasks, they provided an insight on the impact of the proposed call authorization model on the call setup delays that users will experience.

CHAPTER VI

A FRAMEWORK FOR END-TO-END CALL SIGNALING IN HETEROGENEOUS NETWORKS

Following the concept of an application-level control of IP networks and the interfaces between application and transport domains proposed in [102], a case study of a network with heterogeneous transport-level domains (which supports heterogeneous QoS schemes) is performed to evaluate the flexibility and benefits of the proposed signaling architectures for SIP and QoS interaction. Basically, the goal is to verify the applicability of the signaling architectures previously proposed to a real network, with different types of infrastructure, QoS schemes, and network access devices.

6.1 Problem and Solution

When mobile users access the Internet through wireless access networks, which can be for instance a General Packet Radio Services (GPRS) network, a 3G Universal Mobile Telecommunications System (UMTS), a wireless local area network (WLAN), or a satellite network, a common element is that most user applications are based on the IP network layer protocol for routing and packet forwarding. The benefits of an all-IP network include the capability of IP for carrying all types of data, and the fact that using the same technology (i.e., IP) in fixed and mobile networks facilitates their integration.

Following this trend, fourth-generation (4G) systems propose to integrate all systems (3G, WLAN, legacy systems, fixed IP networks), offering all services, all the time [36, 76]. It is envisioned as a system that will be able to offer personalized services over the most efficient/preferred network, depending on the user profile and the type of data to transmit. 4G wireless networks will generally be characterized by heterogeneity in architectures, protocols, and air interfaces [108].

However, the idea of seamless connection of heterogeneous wireless access networks and

fixed IP networks raises some questions from the point of view of resource management and quality of service provision. There is no unique common solution to provide QoS in the Internet [102]. Therefore, the variety of access devices and network infrastructures requires heterogeneous QoS mechanisms to support an end-to-end communication path. For instance, some access devices have limited capacity and may not directly support QoS; also, different networks that employ different QoS mechanisms require means to manage both the policies and the QoS control protocols in order to provide end-to-end QoS. To the best of our knowledge, [120] is the only reference that addresses end-to-end QoS support in heterogeneous networks. The authors in [120] propose a policy-based QoS management architecture considering the interconnection between UMTS and WLAN system as a multi-domain system. They present different UMTS-WLAN interworking scenarios that minimize session setup delay and policy exchange load while maximizing network scalability.

The application of current resource management schemes and Internet protocols to heterogeneous networks to facilitate the achievement of end-to-end QoS is here investigated for interactive multimedia applications that use SIP as the session control signaling protocol. SIP can provide a common ground for users to exchange QoS requirements and policies. At the network level, resource negotiation between neighboring domains is based on the two-tier resource management model [105] and the concept of an application-level control of transport domains [102]. To illustrate an approach to end-to-end QoS in heterogeneous networks, a control architecture assuming the interworking of the UMTS QoS architecture for 3G systems [4], and the INSIGNIA QoS architecture for mobile ad-hoc networks [64] is presented. In summary, this chapter addresses the need for end-to-end call signaling in heterogeneous IP networks to protect resources for real-time multimedia streams.

6.2 End-to-End QoS Support

End-to-end QoS in heterogeneous networks requires service negotiation either before a session begins or during an on-going session, and ways to manage the various QoS signaling protocols of autonomous wireless access networks and network domains to achieve the service users agreed upon for a given session.

6.2.1 Service Negotiation

Assuming interactive multimedia sessions that require session control signaling to set up the sessions, SIP is the Internet protocol to be used for service negotiation at the application level. First, SIP provides capability negotiation, i.e., flow information is carried by SIP messages so that users can agree on the types of media encoding to use. Second, signaling for session setup can be integrated with resource reservation protocols to reserve resources for a new session. In this integration, QoS requirements of the media flows are used to communicate the desired and current QoS reservation status, in the form of *preconditions* for the session establishment. Therefore, SIP messages transport flow information and its QoS attributes end-to-end. This allows a seamless negotiation of media capabilities and QoS status to users no matter the type of their access network.

Further on the interaction of SIP and QoS, the idea of QoS-enhanced SIP proxies has been proposed to provide an interaction of SIP and QoS admission control [66], prior to the actual resource reservation. In Chapter 5, QoS-enhanced SIP proxies functioning as *gate controllers* [48] have been implemented on an inter-domain call authorization model that allows the interaction of application and network layer entities that control and provide access to network resources.

Assuming the use of SIP and QoS-enhanced proxies at the access network, end users can agree on the multimedia session's parameters and the required QoS. Such information allow users to invoke their resource reservation schemes and map the flows to resource reservation flows. (Ways of mapping media flows to resource reservation flows are presented in [20]). But, in such heterogeneous network layer, the issue of how the resource reservation requests can provide the required QoS along the data path relies on the interworking of QoS signaling protocols.

6.2.2 Heterogeneous QoS Signaling Protocols

Managing the heterogeneous QoS control protocols of different domains requires *inter-domain* service negotiation. The concept of *intra-domain* and *inter-domain* resource management is defined in the two-tier resource management model [105], which is based on the

concept of Internet routing where each autonomous system has its own routing protocol and an inter-system routing protocol (such as the border gateway protocol (BGP)) routes packets between different autonomous systems.

At the wireless access networks, intra-domain schemes have their own policies and procedures to manage internal resources to deal with the individual characteristics of the radio network. The way signaling information is exchanged within the domain varies, and they can define either to separate control from the data transport (such as the UMTS QoS architecture [4] or to send signaling information in the same packet as data (such as INSIGNIA's in-band signaling [64]).

Inter-domain negotiation depends on the operation of edge routers. They are gateways in wireless access networks, such as the GPRS gateway support node (GGSN) in UMTS, or a fixed access point in WLANs. They provide the interface between wireless access networks and the IP backbone.

From a domain perspective, QoS parameters are mainly derived from end users' expectations and service level agreements (SLAs). Network operators in neighboring domains can establish an initial SLA, which can then be adjusted automatically through inter-domain service negotiation. Inter-domain schemes work on a flow aggregate level. For instance, when a QoS-enhanced proxy interacts with the edge router to admit the new session for a user in a certain class of service, this may trigger the edge router to request additional network resources for that service class. Additional network resources are requested to a higher-level entity such as a bandwidth broker [23] or a service domain that oversees the domain [102]. In [105], bandwidth brokers from adjacent domains negotiate resources, then intra-domain schemes are invoked to verify if internal resources can be allocated for the new request.

In summary, the application of inter-domain and intra-domain resource management concepts to a heterogeneous network is illustrated in Figure 50. At an application level, source and destination domains communicate QoS requirements and media flow information during session initiation and eventual session re-negotiations. QoS-enhanced SIP proxies perform the role of gate controllers by interacting with edge routers to verify if the session

can be admitted to the network and to set up session policy information at the routers. Each access network (e.g., a 3G network and a mobile ad-hoc network) invoke its own intra-domain scheme to allocate resources for the new call. Resource allocation may require inter-domain resource negotiations between adjacent domains.

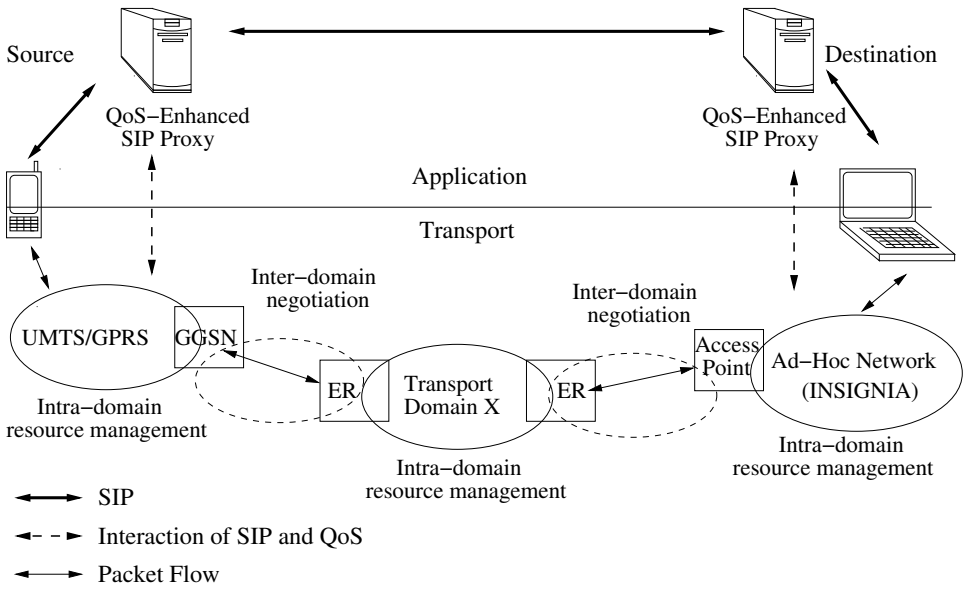


Figure 50: End-to-end QoS support in heterogeneous networks.

6.3 Framework for Managing Heterogeneous Networks

Figure 51 illustrates in more detail the building blocks of a framework for managing the QoS signaling protocols of heterogeneous access networks. The transport path is represented by the gateway/bandwidth broker modules. Horizontal arrows represent the exchange of inter-domain control messages, whereas the vertical arrows represent communication with intra-domain protocols. Based on this control architecture, next an example with UMTS and mobile ad-hoc network at the access domains is presented. Also, details of the application and network layer interface is given to demonstrate the use of SIP signaling schemes integrated with resource management during call setup and call management phases.

6.3.1 An Example of Heterogeneous Access Networks

In this section, the application of the proposed control architecture is studied for two different access networks: a GPRS network [3] and a wireless ad-hoc network that uses an

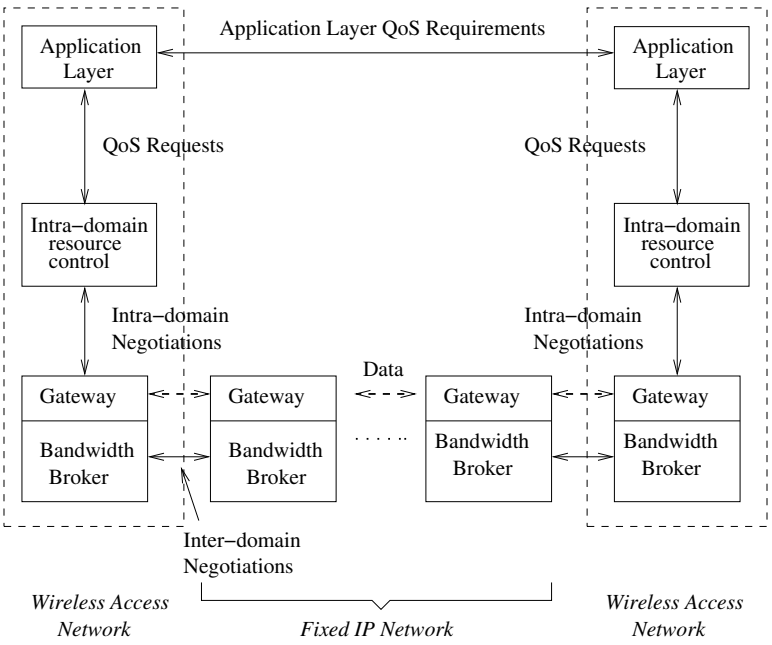


Figure 51: Control architecture based on two-tier resource management model.

in-band QoS signaling framework named INSIGNIA [64].

This example is illustrated in Figure 52. A mobile terminal (MT) of a GPRS network sets up communication with a mobile user of an ad-hoc wireless network. At the application layer, they use SIP to communicate the types of media flow they want to exchange, their source and destination addresses and port numbers, the bandwidth required for the session, and the type of QoS requirements desired for the session.

At the UMTS/GPRS network, the MT can use the resource reservation protocol (RSVP) to request resources to the network intra-domain's resource management model. In the UMTS QoS architecture, QoS support is provided through a layered QoS architecture [4]. The layers represent *bearer services*. For instance, on the top-layer, an end-to-end service is realized through the terminal equipment/mobile terminal (TE/MT) local bearer service, the UMTS bearer service, and the external bearer service. The focus here is on the UMTS bearer service and its use as an intra-domain resource scheme. In GPRS's network infrastructure, the GPRS support node (GGSN) provides access to external data networks. As the GGSN provides gateway functionality to external networks and implements DiffServ edge functions (e.g., packet marking, policing, scheduling), it acts as a policy enforcement point (PEP). It

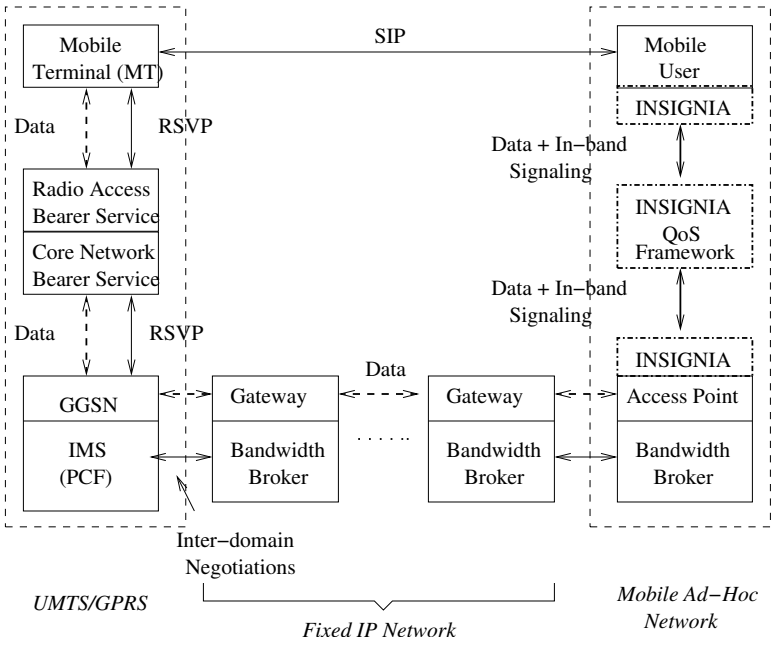


Figure 52: Example of control architecture with different QoS frameworks.

communicates with a policy control function (PCF) at the IP Multimedia System of UMTS networks (IMS) which stores the local policies, service level agreements (SLA), and can have the role of a bandwidth broker to communicate with neighboring bandwidth brokers when the maximum threshold of a certain class is reached (inter-domain negotiation).

At the wireless ad-hoc network, the intra-domain resource management model is adapted to the INSIGNIA distributed QoS framework. INSIGNIA aims to provide QoS to adaptive multimedia applications, by taking into consideration the dynamic changes in network topology and link quality which are typical in ad hoc environments. For each mobile node, the main components of the framework include a generic routing protocol, in-band signaling, an admission control module, and a packet scheduler. Among the above components, in-band signaling offers flexibility to establish flows and to adapt them to changes in the data path. Thus, a mobile user applies in-band signaling to request internal network resources. When the mobile user communicates with the MT at the GPRS network, a fixed access node provides the interface between the ad hoc network and a fixed IP domain. This access node receives packets with in-band signaling requiring reservation of resources. Acting as a gateway to other networks, it implements the following functions:

- translation of the control information contained in the reservation packet,
- monitoring of outgoing traffic,
- bandwidth broker capability to interface with peer bandwidth brokers in case outgoing aggregate flows reach a limit threshold,
- reporting back to source to give status of the reservation process.

When actual communication occurs, in-band signaling refreshes the reservations, notifies eventual bottleneck nodes, and efficiently transmits control information when the network topology changes. In the case of communication with external networks, the access point constantly receives in-band signaling which may indicate changes in the initial reservation process. The access point, in turn, periodically performs the gateway functions of translation, traffic monitoring, reporting back to source, and negotiation with adjacent bandwidth brokers.

6.3.2 Application/Network Interface

The interface between application layer and intra-domain resource control of the control architecture shown in Figure 51 can be achieved through the different signaling architectures proposed in this thesis. Figure 53 illustrates the framework for this interface. At the application/user space, there are different types of access devices and QoS-enhanced SIP proxy servers. At the network space, two planes can be identified: the data plane, and the QoS and network management control plane. At the data plane, data flow through DiffServ-capable routers. At the QoS and network management control plane, control is provided by an entity such as a bandwidth broker.

At the application/user space, the user equipment may exchange QoS signaling directly with the network space, such as reservation requests directly to the edge router. In this case, the ROAD scheme proposed in Chapter 4 can be applied to coordinate this interaction and to provide QoS "most of the time". In other cases, such as the case of access devices with limited QoS functionality, reservations can occur through the QoS-enhanced SIP proxy using the lightweight signaling architecture proposed in Chapter 3. The inter-domain call

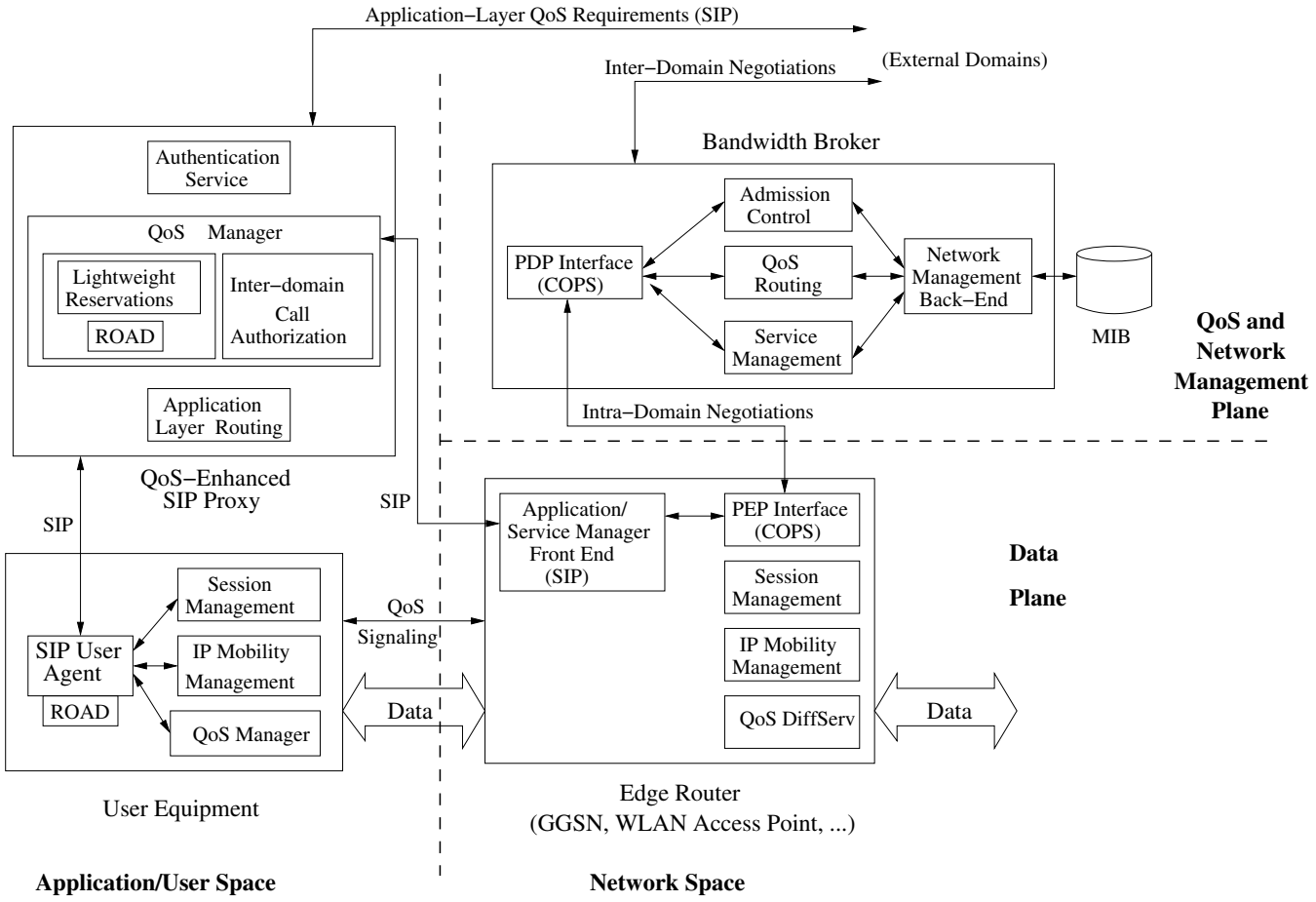


Figure 53: Framework for the interaction of application and network layers using the proposed signaling architectures.

authorization model can be applied for every new session request, independent of the QoS reservation scheme being used. The inter-domain call authorization model interfaces with the edge router, which in turn may invoke intra-domain negotiations with the bandwidth broker.

At the network space, the functionality of the bandwidth broker includes admission control, QoS routing, service management, and a network management interface with the management information base (MIB). A detailed description of this architecture can be found in [62]. The edge router functionality includes for instance some functions derived from the UMTS QoS architecture [4] (such as IP mobility management and session management for packet data protocol (PDP) signaling support). The architectures of the bandwidth broker and edge router include other functionality as well; however, those are the ones most relevant to this work. Especially, the application/service manager interface with the application layer is here shown through a SIP-based interface, as was used in the implementations of the proposed signaling architectures.

6.4 Conclusion

In this chapter, the roles of QoS signaling and session control signaling are illustrated in a control architecture that has the potential to offer end-to-end QoS between heterogeneous access networks. The UMTS/GPRS network and a mobile ad-hoc wireless LAN are used as examples of access networks. The interaction between application and network layers is presented in a framework that incorporates the signaling schemes proposed in the previous chapters. This is an example framework that shows the applicability of the proposed schemes in the problem of end-to-end call signaling among heterogeneous networks.

CHAPTER VII

CONCLUSION

Seeking to improve the interaction of session control signaling and network resource management, this thesis presented new signaling architectures that addressed the interaction of SIP as the session control signaling protocol and current resource management frameworks. The Differentiated Services (DiffServ) architecture is used as the primary example. The new architectures addressed the roles of SIP agents and proxy servers in subjects such as resource negotiation, call authorization, and end-to-end QoS in heterogeneous networks.

Since interactive multimedia applications combine the requirements of traditional telephony services and Internet applications, the approach taken on the design of the signaling architectures was to provide users with practical and fast ways to reserve resources during session establishment and to avoid the issue of "no QoS/no call". By considering the different types of applications in the Internet, the different levels of QoS guarantees, and the different users' ability to access resource management schemes, the proposed architectures investigated a new call setup paradigm - an Internet-style call setup model, instead of a telephony-style call setup model.

As an initial work in this direction, an architecture based on the use of QoS-enhanced SIP proxies and a SIP-based interface between the application and network layers was developed, implemented in a testbed, and performance enhancements demonstrated. QoS-enhanced SIP proxies not only support the typical session control functionality but also understand the users' QoS requirements and provide an interface with the network routers. Due to its simple mechanism of a one-way interaction between QoS-enhanced SIP proxies and DiffServ-capable edge routers and its small overhead in terms of signaling and coarse-grain QoS guarantees, this scheme was named as a lightweight scheme. On behalf of users who may not have access to QoS reservation protocols, the QoS-enhanced SIP proxies inform the network the types of media flows and the required bandwidth for the session, in addition

to end-user information, in parallel with the on-going call setup signaling.

Further studying of the IETF's proposed integration of SIP and resource management led to the development of the improved alternative scheme, Resource management Overlapped with Answering Delays (ROAD). It explores the SIP user agent interaction with the network in a way that takes advantage of parallel user answering delays and reservation delays. Also, a new capability negotiation approach was taken to allow a more flexible QoS negotiation, in terms of allowing a best-effort flow to transition to a QoS-enabled multimedia flow as part of the extended call setup. This introduces flexibility in the setup of adaptive multimedia sessions, and reduces the impact of additional signaling on the call setup delays. An experimental evaluation of the call setup delay savings and signaling load of the ROAD signaling scheme showed that in most cases the ROAD schemes offsets the reservation delays. Also, the most gains in terms of delay improvement are achieved when the user answering delays and reservations delays are in a close range.

On the issue of the interaction of SIP and call admission control, the proxy-network interaction was implemented in an inter-domain call authorization model that incorporates the concepts of proxies as gate controllers (QoS-enhanced SIP proxies-GC) and the need to properly authorize the media flows prior to the actual resource reservation. The new model provides call authorization status and adds more granularity to the authorization process at the destination domain. A delay analysis has been presented that addressed scalability issues in terms of the need to add more resources (i.e., proxy servers) to compensate for the increasing load on the servers, given delay constraints on the SIP transactions such as timeout constraints for application-layer retransmissions in UDP-based SIP transport. This analysis showed that the new call authorization model is scalable at the end domains.

All the above signaling architectures were evaluated experimentally, in a testbed with Linux PCs and routers where a SIP protocol stack was implemented. With the SIP testbed, the feasibility of having a SIP call setup scheme integrated with resource management was demonstrated, through the different signaling architectures discussed above.

Finally, the last chapter presented an example framework that applies the new signaling architectures to achieve end-to-end QoS in heterogeneous networks. This framework showed

the functionalities of the network layer and application layer entities and how they interact to manage resources for the call. From the viewpoint of intra-domain and inter-domain resource management schemes, examples of the signaling needed to achieve QoS are given, and especially two types of wireless access networks are considered: UMTS and mobile ad-hoc networks.

The example framework is an open and flexible framework that can accommodate other signaling schemes at the proxy-user interaction to the network. It is envisioned that in heterogeneous network environments, each with their own policies and QoS schemes, the application-layer adds some type of control to harmonize their differences in terms of QoS signaling. Thus, the interaction between application and network layers to support QoS-enabled multimedia sessions is very important. In the course of this research work, the growing importance of the interaction of session control signaling and resource management has been observed in the Internet community and in the industry (such as in the UMTS QoS framework and the PacketCable multimedia framework).

The main focus of the current work and related work has been on the call setup phase. Most of the signaling schemes that are used to establish a session can be used to update an active session's parameters (i.e., for call management). This update, which usually occurs through a *re-invite* request, is triggered for instance by changes in the network conditions due to session mobility. However, in this topic as well as other areas in SIP, there are a number of problems for future investigation:

- analyze the efficiency of the proposed signaling architectures for managing active calls;
- investigate how feedback packets at the network level (in-band signaling) interact with session control signaling (out-of-band signaling) to trigger new intra-domain and eventually inter-domain negotiations to manage resources for an on-going call;
- improve the capability negotiation process in SIP, by supporting more flexible interactions in the offer/answer negotiation process to allow not only the negotiation of media parameters but also of QoS requirements at the application layer;

- study the effects of new transport-layer protocols (such as SCTP) and security mechanisms to the performance of SIP signaling; and
- on the role of application-layer signaling to achieve end-to-end QoS in heterogeneous wired/wireless networks, research resource management techniques and their integration in the call setup/call management of multimedia sessions.

REFERENCES

[1] *Differentiated Services on Linux.* http://diffserv.sourceforege.net.

[2] "PacketCable Multimedia Architecture Framework," tech. rep., Cable Labs, http://www.packetcable.com, June 2003.

[3] 3G TS 23.060 V3.4.0 (2000-07), "Universal Mobile Telecommunications Systems (UMTS); General Packet Radio Service (GPRS); Service Description; Stage 2 (Release 99)," tech. rep., 3GPP/ETSI, http://www.3gpp.org, 2000.

[4] 3G TS 23.207 V6.3.0 (2004-06), "End-to-end Quality of Service (QoS) concept and architecture (Release 6)," tech. rep., 3GPP/ETSI, http://www.3gpp.org, June 2004.

[5] Almesberger, W., Salim, J., and Kuznetsov, A., *Differentiated Services on Linux.* IETF Internet Draft, draft-almesberger-wajhak-diffserv-linux-01, June 1999.

[6] Andrews, M., Borst, S. C., Dominique, F., Jelenkovic, P. R., Kumaran, K., Ramakrishnan, K. G., and Whiting, P. A., "Dynamic Bandwidth Allocation Algorithms for High-Speed Data Wireless Networks," *Bell Labs Technical Journal*, pp. 30–49, July-September 1998.

[7] Armitage, G., *Quality of Service in IP Networks.* Macmillan Technical Publishing, ISBN 1-57870-189-9, 2000.

[8] Banerjee, N., Wu, W., Basu, K., and Das, S. K., "Analysis of SIP-based Mobility Management in 4G Wireless Networks," *Computer Communications*, vol. 27, pp. 697–707, 2004.

[9] Barberis, A., Casetti, C., Martin, J. D., and Meo, M., "A Simulation Study of Adaptive Voice Communications on IP Networks," *Computer Communications*, vol. 24, pp. 757–767, 2001.

[10] Bernet, Y., Yavatkar, R., Ford, P., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J., and Felstaine, E., *A Framework for Integrated Services Operation over DiffServ Networks.* IETF RFC 2998, http://www.ietf.org/rfc/rfc2998, November 2000.

[11] Bertsekas, D. and Gallager, R., *Data Networks.* Prentice Hall, Inc., ISBN 0-13-200916-1, 1992.

[12] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W., *An Architecture for Differentiated Services.* IETF RFC 2475, http://www.ietf.org/rfc/rfc2475, December 1998.

[13] Bos, L. and Leroy, S., "Toward an All-IP-Based UMTS System Architecture," *IEEE Network*, vol. 15, pp. 36–45, January/February 2001.

[14] Braden, R., Zhang, L., Berson, S., Herzog, S., and Jamin, S., *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification.* IETF RFC 2205, http://www.ietf.org/rfc/rfc2205, September 1997.

[15] Braun, T., Castelluccia, C., Stattenberger, G., and Aad, I., "An Analysis of the DiffServ Approach in Mobile Environments," in *Proc. Workshop on IP Quality of Service for Wireless and Mobile Networks (IQWiM99)*, Germany 1999.

[16] Busse, I., Deffner, B., and Schulzrinne, H., "Dynamic QoS Control of Multimedia Applications Based on RTP," *Computer Communications*, vol. 19, pp. 49–58, 1996.

[17] Camarillo, G., Eriksson, G., Holler, J., and Schulzrinne, H., *Grouping of Media Lines in the Session Description Protocol (SDP).* IETF RFC 3388, http://www.ietf.org/rfc/rfc3388, December 2002.

[18] Camarillo, G., Kantola, R., and Schulzrinne, H., "Evaluation of Transport Protocols for the Session Initiation Protocol," *IEEE Network*, vol. 17, pp. 40–46, September 2003.

[19] Camarillo, G., Marshall, W., and Rosenberg, J., *Integration of Resource Management and Session Initiation Protocol (SIP).* IETF RFC 3312, http://www.ietf.org/rfc/rfc3312, October 2002.

[20] Camarillo, G. and Monrad, A., *Mapping of Media Streams to Resource Reservation Flows.* IETF RFC 3524, http://www.ietf.org/rfc/rfc3524, April 2003.

[21] Chalmers, D. and Sloman, M., "A Survey of Quality of Service in Mobile Computing Environments," *IEEE Communications Surveys*, Second Quarter 1999.

[22] Chen, J.-C., McAuley, A., Caro, A., Baba, S., Ohba, Y., and Ramanathan, P., *QoS Architecture Based on Differentiated Services for Next Generation Wireless IP Networks.* IETF Internet Draft, draft-itsumo-wireless-diffserv-00, July 2000.

[23] Chimento, P. and Teitelbaum, B., *Qbone Bandwidth Broker Architecture.* http://qbone.internet2.edu/bb/bboutline2.html, 2000.

[24] Cho, S., Goulart, A., and Akyildiz, I. F., "Adaptive FEC for Real-Time Traffic in LEO Satellite Networks," in *Proc. IEEE ICC'2001*, (Helsinki, Finland), June 2001.

[25] Comer, D. E. and Stevens, D. L., *Internetworking with TCP/IP, Vol.III, Client-Server Programming and Applications.* Prentice Hall, ISBN 0-13-260969-X, 1996.

[26] Costello, D. J., Hagenauer, J., Imai, H., and Wicker, S., "Applications of Error Control Coding," *IEEE Transactions on Information Theory*, vol. 44, pp. 2531–2560, October 1998.

[27] Crocker, D. and Overell, P., *Augmented BNF for Syntax Specifications: ABNF.* IETF RFC 2234, http://www.ietf.org/rfc/rfc2234, November 1997.

[28] Crow, B. P., Widjaja, I., Kim, J. G., and Sakai, P. T., "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, pp. 116–126, September 1997.

[29] DALGIC, I. and FANG, H., "Comparison of H.323 and SIP for IP Telephony Signaling," in *Proc. of Photonics East*, September 1999.

[30] DAS, S. K., JAYARAM, R., KAHANI, N. K., and SEN, S. K., "A Call Admission and Control Scheme for Quality-of-Service (QoS) Provisioning in Next Generation Wireless Networks," *Wireless Networks*, vol. 6, pp. 17–29, February 2000.

[31] DE CARMO, L., "Internet Telephony Protocols," *Dr. Dobb's Journal*, pp. 30–39, July 1999.

[32] DIERKS, T. and ALLEN, C., *The TLS Protocol.* IETF RFC 2246, http://www.ietf.org/rfc/rfc2246, January 1999.

[33] DIXIT, S., GUO, Y., and ANTONIOU, Z., "Resource Management and Quality of Service in Third-Generation Wireless Networks," *IEEE Communications Magazine*, vol. 39, pp. 125–133, February 2001.

[34] DURHAM, D., BOYLE, J., COHEN, R., HERZOG, S., RAJAN, R., and SASTRY, A., *The COPS (Common Open Policy Service) Protocol.* IETF RFC 2748, http://www.ietf.org/rfc/rfc2748, January 2000.

[35] ETSI TS 101 329-3 v1.1.1, "End-to-End Quality of Service in TIPHON Systems; Part 3: Signaling and Control of End-to-End Quality of Service," tech. rep., ETSI Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON), 2001.

[36] EVANS, B. G. and BAUGHAN, K., "Visions of 4G," *Electronics and Communication Engineering Journal*, vol. 12, pp. 293–303, December 2000.

[37] FACCIN, S. M., LALWANEY, P., and PATIL, B., "IP Multimedia Services: Analysis of Mobile IP and SIP Interactions in 3G Networks," *IEEE Communications Magazine*, vol. 42, pp. 113–120, January 2004.

[38] FINEBERG, V., "A Practical Architecture for Implementing End-to-End QoS in an IP Network," *IEEE Communications Magazine*, vol. 40, pp. 122–130, January 2002.

[39] FRANKS, J., HALLAM-BAKER, P., HOSTETLER, J., LAWRENCE, S., LEACH, P., LUOTONEN, A., and STEWART, L., *HTTP Authentication: Basic and Digest Access Authentication.* IETF RFC 2617, http://www.ietf.org/rfc/rfc2617, June 1999.

[40] FROST, V. S. and MELAMED, B., "Traffic Modelling for Telecommunications Networks," *IEEE Communications Magazine*, pp. 70–81, March 1994.

[41] GAMMA, E., HELM, R., JOHNSON, R., and VLISSIDES, J., *Design Patterns - Elements of Reusable Object-Oriented Software.* Addison Wesley Publishing Company, Inc., ISBN 0-201-63361-2, 1994.

[42] GAYNOR, M., "Linking Market Uncertainty to VoIP Service Architectures," *IEEE Internet Computing*, vol. 7, pp. 16–22, July 2003.

[43] GIBSON, M. and CROWCROFT, J., *Use of SIP for the Reservation of QoS Guaranteed Paths.* IETF Internet Draft, draft-gibson-sip-qos-resv-00, October 1999.

[44] Glitho, R. H., "Advanced Services Architectures for Internet Telephony: A Critical Overview," *IEEE Network*, pp. 38–44, July/August 2000.

[45] Goulart, A., "Achieving End-to-End QoS Through the Application of an Internet Resource Management Model to 4G Systems," in *Proc. The Path to 4G Mobile*, (IIR - U.K.), September 2001.

[46] Goulart, A. and Abler, R. T., "Interaction of Session Initiation Protocol (SIP) and Quality of Service (QoS) for Internet Multimedia Sessions," in *Proc. International Conference on Computer, Communications and Control (CCCT'03)*, (Orlando, FL), 2003.

[47] Goulart, A. and Abler, R. T., "On Overlapping Resource Management and Call Setup Signaling: a New Signaling Approach for Internet Multimedia Applications," *Computer Communications*, 2005. In press.

[48] Goyal, P., Greenberg, A., Kalmanek, C., Marshall, W., Mishra, P., Nortz, D., and Ramakrishnan, K., "Integration of Call Signaling and Resource Management for IP Telephony," *IEEE Network*, pp. 24–32, May/June 1999.

[49] Guardini, I., D'Urso, P., and Fasano, P., "The Role of Internet Technology in Future Mobile Data Systems," *IEEE Communications Magazine*, vol. 38, pp. 68–72, November 2000.

[50] Haartsen, J. C., "The Bluetooth Radio System," *IEEE Personal Communications*, pp. 28–36, February 2000.

[51] Handley, M. and Jacobson, V., *SDP: Session Description Protocol.* IETF RFC 2327, http://www.ietf.org/rfc/rfc2327, April 1998.

[52] Handley, M., Schulzrinne, H., Schooler, E., and Rosenberg, J., *SIP: Session Initiation Protocol.* IETF RFC 2543, http://www.ietf.org/rfc/rfc2543, March 1999.

[53] Hannu, H., *Signaling Compression (SigComp) Requirements and Assumptions.* IETF RFC 3322, http://www.ietf.org/rfc/rfc3322, January 2003.

[54] Heinanen, J., Baker, F., Weiss, W., and Wroclawski, J., *Assured Forwarding PHB Group.* IETF RFC 2597, http://www.ietf.org/rfc/rfc2597, June 1999.

[55] Herzog, S., *RSVP Extensions for Policy Control.* IETF RFC 2750, http://www.ietf.org/rfc/rfc2750, January 2000.

[56] ITU-T Rec. H.323, "Packet-Based Multimedia Communications Systems," *ITU-T*, vol. 4, November 2000.

[57] Jacobson, V., Nichols, K., and Poduri, K., *An Expedite Forwarding PHB.* IETF RFC 2598, http://www.ietf.org/rfc/rfc2598, June 1999.

[58] Jennings, C., Peterson, J., and Watson, M., *Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks.* IETF RFC 3325, http://www.ietf.org/rfc/rfc3325, November 2002.

[59] Jiang, G., Lan, J., and Zhuang, X., "Distance Learning Technologies and an Interactive Multimedia Educational System," in *Proc. IEEE International Conference on Advanced Learning Technologies*, pp. 405–408, 2001.

[60] Knight, R. R., Norreys, S. E., and Harrison, J. R., "Bearer-Independent Call Control," *BT Technology Journal*, vol. 19, pp. 77–88, April 2001.

[61] Koodli, R. and Puuskari, M., "Supporting Packet-Data QoS in Next-Generation Cellular Networks," *IEEE Communications Magazine*, vol. 39, pp. 180–188, February 2001.

[62] Kusmierek, E., Choi, B., Duan, Z., and Zhang, Z., "An Integrated Network Resource and QoS Management Framework," in *Proc. IEEE Workshop on IP Operations and Management (IPOM'02)*, (Dallas, Texas), 2002.

[63] Kwon, T. T., Gerla, M., and Das, S., "Mobility Management for VoIP Service: Mobile IP vs. SIP," *IEEE Wireless Communications*, vol. 9, pp. 66–75, October 2002.

[64] Lee, S.-B., Ahn, G.-S., Zhang, X., and Campbell, A., "INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks," *Journal of Parallel and Distributed Computing*, vol. 60, pp. 374–406, April 2000.

[65] Lu, W. W., "Compact Multidimensional Broadband Wireless: The Convergence of Wireless Mobile and Access," *IEEE Communications Magazine*, vol. 38, pp. 119–123, November 2000.

[66] Marshall, W., *Private Session Initiation Protocol (SIP) Extensions for Media Authorization*. IETF RFC 3313, http://www.ietf.org/rfc/rfc3313, January 2003.

[67] Mathy, L., Edwards, C., and Hutchison, D., "The Internet: A Global Telecommunications Solution?," *IEEE Network*, pp. 46–57, July/August 2000.

[68] Mikkonen, J. and Turunen, M., "An Integrated QoS Architecture for GSM Networks," in *Proc. IEEE International Conference on Universal Personal Communications (ICUPC'98)*, (Florence, Italy), October 1998.

[69] Mortada, I. and Probst, W., "Internet Telephony Signaling," *Telematics and Informatics*, vol. 18, pp. 159–194, 2001.

[70] Mortier, R., Pratt, I., Clark, C., and Crosby, S., "Implicit Admission Control," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 2629–2639, December 2000.

[71] Moyer, S., Marples, D., and Tsang, S., "A Protocol for Wide-Area Secure Networked Appliance Communication," *IEEE Communications Magazine*, vol. 39, pp. 52–59, October 2001.

[72] Negus, K. J., Stephens, A. P., and Lansford, J., "HomeRF: Wireless Networking for the Connected Home," *IEEE Personal Communications*, pp. 20–27, February 2000.

[73] Nichols, K., Jacobson, V., and Zhang, L., *A Two-bit Differentiated Services Architecture for the Internet*. IETF RFC 2638, http://www.ietf.org/rfc/rfc2638, July 1999.

[74] Oliveira, C., Kim, J. B., and Suda, T., "An Adaptive Bandwidth Reservation Scheme for High-Speed Multimedia Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 858–874, August 1998.

[75] Patel, G. and Dennett, S., "The 3GPP and 3GPP2 Movements Toward an All-IP Mobile Network," *IEEE Personal Communications Magazine*, vol. 7, pp. 62–64, August 2000.

[76] Pereira, J. M., "Fourth Generation: Now, it is Personal!," in *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'2000)*, vol. 2, pp. 1009–1016, 2000.

[77] Perkins, C., "Mobile IP," *IEEE Communications Magazine*, vol. 35, pp. 84–99, May 1997.

[78] Peterson, J. and Jennings, C., *Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP).* IETF Internet Draft, draft-ietf-sip-identity-02, May 2004.

[79] Pierce, M. and Choi, D., *Architecture for Assured Service Capabilities in Voice over IP.* IETF Internet Draft, draft-pierce-tsvwg-assured-service-arch-00, April 2004.

[80] Priggouris, G., Hadjiefthymiades, S., and Merakos, L., "Supporting IP QoS in the General Packet Radio Service," *IEEE Network*, pp. 8–17, September/October 2000.

[81] Puuskari, M., "Quality of Service Framework in GPRS and Evolutions Towards UMTS," in *Proc. 3rd European Personal Mobile Communications Conference*, (Paris, France), March 1999.

[82] Rahman, M., Akinlar, C., and Kamel, I., "On Secured End-to-End Appliance Control Using SIP," in *Proc. IEEE Workshop on Networked Appliances*, pp. 24–28, October 2002.

[83] Ramakrishnan, K., Hjalmtysson, G., and der Merwe, J. V., "The Role of Signaling in Quality of Service Enabled Networks," *IEEE Communications Magazine*, pp. 124–132, June 1999.

[84] Roach, A., *Session Initiation Protocol (SIP): Specific Event Notification.* IETF RFC 3265, http://www.ietf.org/rfc/rfc3265, June 2002.

[85] Rosen, E., Viswanathan, A., and Callon, R., *Multiprotocol Label Switching Architecture.* IETF RFC 3031, http://www.ietf.org/rfc/rfc3031, January 2001.

[86] Rosenberg, J., *Distributed Algorithms and Protocols for Scalable Internet Telephony.* PhD thesis, Columbia University, May 2001.

[87] Rosenberg, J., *The Session Initiation Protocol (SIP) UPDATE Method.* IETF RFC 3311, http://www.ietf.org/rfc/rfc3311, September 2002.

[88] Rosenberg, J. and Kyzivat, P., *Guidelines for Usage of the Session Initiation Protocol (SIP) Caller Preferences Extension.* IETF Internet Draft, draft-ietf-sipping-callerprefs-usecases-02, July 2004.

[89] Rosenberg, J. and Schulzrinne, H., *An Offer/Answer Model with the Session Description Protocol (SDP)*. IETF RFC 3264, http://www.ietf.org/rfc/rfc3264, June 2002.

[90] Rosenberg, J. and Schulzrinne, H., *Reliability of Provisional Responses in the Session Initiation Protocol (SIP)*. IETF RFC 3262, http://www.ietf.org/rfc/rfc3262, June 2002.

[91] Rosenberg, J., Schulzrinne, H., and Camarillo, G., *The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP)*. IETF Internet Draft, draft-ietf-sip-sctp-06.txt, January 2005.

[92] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., R.Sparks, Handley, M., and Schooler, E., *SIP: Session Initiation Protocol*. IETF RFC 3261, http://www.ietf.org/rfc/rfc3261, June 2002.

[93] Saad, A., Kawamura, K., and Biswas, G., "Performance Evaluation of Contract Net-Based Heterarchical Scheduling for Flexible Manufacturing Systems," *International Journal of Intelligent Automation and Soft Computing*, vol. 3, pp. 233–252, January 1997.

[94] Salsano, S. and Veltri, L., "QoS Control by Means of COPS to Support SIP-Based Applications," *IEEE Network*, vol. 16, pp. 27–33, March/April 2002.

[95] Salsano, S., Veltri, L., and Papalilo, D., "SIP Security Issues: The SIP Authentication Procedure and its Processing Load," *IEEE Network*, vol. 16, pp. 38–44, November/December 2002.

[96] Schulzrinne, H. and Rosenberg, J., "The Session Initiation Protocol: Providing Advanced Telephony Services Across the Internet," *Bell Labs Technical Journal*, pp. 144–160, October–December 1998.

[97] Schulzrinne, H. and Rosenberg, J., "Internet Telephony: Architecture and Protocols - an IETF Perspective," *Computer Networks*, vol. 31, pp. 237–255, February 1999.

[98] Schulzrinne, H. and Rosenberg, J., "Comparison between SIP and H.323," in *Proc. of Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, July 1998.

[99] Schulzrinne, H., Rosenberg, J., and Lennox, J., "Interaction of Call Setup and Resource Reservation Protocols in Internet Telephony," tech. rep., Columbia University, Computer Science Dept., http://www.cs.columbia.edu/sip/drafts/resource.pdf, June 1999.

[100] Schulzrinne, H. and Wedlund, E., "Application-layer Mobility Using SIP," *Mobile Computing and Communications Review*, vol. 4, July 2000.

[101] Sicker, D. C., Kulkarni, A., Chavali, A., and Fajandar, M., "A Federated Model for Secure Web-Based Videoconferencing," in *Proc. IEEE Conference on Information Technology: Computers and Communications (ITCC'03)*, pp. 396–400, April 2003.

[102] Sijben, P., Buckley, M., Segers, J., and Spergel, L., "Application-Level Control of IP Networks: IP Beyond the Internet," *Bell Labs Technical Journal*, pp. 98–115, January–June 2001.

[103] Stevens, W. R., *TCP/IP Illustrated, Vol.I, The Protocols.* Addison Wesley Publishing Company, Inc., ISBN 0-201-633346-9, 1994.

[104] Terzis, A., Srivastava, M., and Zhang, L., "A Simple QoS Signaling Protocol for Mobile Hosts in the Integrated Services Internet," in *Proc. INFOCOM'99*, 1999.

[105] Terzis, A., Wang, L., Ogawa, J., and Zhang, L., "A Two-Tier Resource Management Model for the Internet," in *Proc. GLOBECOM'99*, (Rio de Janeiro, Brazil), 1999.

[106] Tulu, B., Chatterjee, S., Abhichandani, T., and Li, H., "Secured Video Conferencing Desktop Client for Telemedicine," in *Proc. IEEE Workshop on Enterprise Networking and Computing in Healthcare Industry (HealthCom)*, June 2003.

[107] v. Nee, R., Awater, G., Morikura, M., Takanashi, H., Webster, M., and Halford, K. W., "New High-Rate Wireless LAN Standards," *IEEE Communications Magazine*, pp. 82–88, December 1999.

[108] Varshney, U. and Jain, R., "Issues in Emerging 4G Wireless Networks," *IEEE Computer*, vol. 34, no. 6, pp. 94–96, 2001.

[109] Viniotis, Y., *Probability and Random Processes for Electrical Engineers.* WCB/McGraw-Hill , ISBN 0-07-067491-4, 1998.

[110] Wang, H., Joseph, A., and Katz, R., "A Signaling System Using Lightweight Call Sessions," in *Proc. of IEEE INFOCOM 2000*, pp. 697–706, 2000.

[111] White, P. P. and Crowcroft, J., "The Integrated Services in the Internet: State of the Art," *Proceedings of the IEEE*, vol. 85, pp. 1934–1946, December 1997.

[112] Wisely, D. R., "SIP and Conversational Internet Applications," *BT Technology Journal*, vol. 19, pp. 107–118, April 2001.

[113] Wisely, D. R., "The Challenges of an All IP Fixed and Mobile Telecommunications Network," in *Proc. of IEEE Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, vol. 1, (London, UK), pp. 13–18, September 2000.

[114] Wroclawski, J., *The Use of RSVP with IETF Integrated Services.* IETF RFC 2210, http://www.ietf.org/rfc/rfc2210, September 1997.

[115] Xiao, X., Hannan, A., Bailey, B., and Ni, L. M., "Traffic Engineering with MPLS in the Internet," *IEEE Network*, vol. 14, pp. 28–33, March/April 2000.

[116] Xiao, X. and Ni, L. M., "Internet QoS: A Big Picture," *IEEE Network*, vol. 13, pp. 8–18, March/April 1999.

[117] Xin, W. and Schulzrinne, H., "An Integrated Resource Negotiation, Pricing and QoS Adaptation Framework for Multimedia Applications," *IEEE Journal on Selected Areas on Communications*, vol. 18, pp. 2514–2525, December 2000.

[118] Yavatkar, R., Pendarakis, D., and Guerin, R., *A Framework for Policy-Based Admission Control.* IETF RFC 2753, http://www.ietf.org/rfc/rfc2753, June 2002.

[119] Zhuang, W., Gan, Y., Loh, K., and Chua, K., "Policy-Based QoS Architecture in the IP Multimedia Subsystem of UMTS," *IEEE Network*, vol. 17, pp. 51–57, May 2003.

[120] Zhuang, W., Gan, Y., Loh, K., and Chua, K., "Policy-Based QoS Management Architecture in an Integrated UMTS and WLAN Environment," *IEEE Communications Magazine*, vol. 41, no. 11, pp. 118–125, 2003.

[121] Zorzi, M. and Rao, R. R., "Perspectives on the Impact of Error Statistics on Protocols for Wireless Networks," *IEEE Personal Communications*, pp. 32–40, October 1999.

VITA

Ana Elisa P. Goulart received the B.S. degree in Electrical Engineering from the Federal School of Engineering at Itajuba- MG (Brazil). She has a M.S. degree in Computer Engineering from North Carolina State University.

She has held positions as a Hardware Development Engineer and as a Communications Analyst at IBM Brasil and at Compaq Computer, respectively. At IBM, she was a member of the ASIC design group, where she worked in the development of network products. At Compaq, she was responsible for data and voice communications. While working in the industry, she completed her Masters in Information Science at the Pontificial Catholic University at Campinas- SP (Brazil).

Ana's research has focused on communication networks and protocols, including wireless networks and signaling for Internet telephony. In special, the research on the interaction of call signaling and resource management was motivated by the need to improve the quality of interactive multimedia applications in the area of distance learning. In this area, she has worked with the Session Initiation Protocol (SIP) and implemented a testbed to experimentally evaluate the call signaling procedures involved with QoS.

Her research interests include Quality of Service in heterogeneous wired/wireless IP networks. The efficient use of the available communication networks infrastructure, given limitations such as user location, type of access device, and mobility are the main goals of her work.